Epi Info

VERSION 6

A Word Processing, Database, and Statistics System

for

Epidemiology on Microcomputers

U.S. DEPARTMENT OF HEALTH & HUMAN SERVICES
Public Health Service

CDC

Epi Info, Version 6

**A Word-Processing, Database, and Statistics Program
for Public Health on
IBM-compatible Microcomputers**

**Program design by Andrew G. Dean, Jeffrey A. Dean,
Denis Coulombier, Anthony H. Burton, Karl A. Brendel,
Donald C. Smith, Richard C. Dicker, Kevin M. Sullivan,
Robert F. Fagan,**

**Programming by
Jeffrey A. Dean, Denis Coulombier, Donald C. Smith,
Karl A. Brendel, Thomas G. Arner, and Andrew G. Dean**

Manual by Andrew G. Dean

Revised for Version 6.03, January 1996

Produced by:
**The Division of Surveillance and Epidemiology
Epidemiology Program Office
Centers for Disease Control and Prevention (CDC)
Atlanta, Georgia 30333**

in collaboration with
**The Global Programme on AIDS
World Health Organization (WHO)
Geneva, Switzerland**

**Epi Info Hotline for Technical Assistance
(404) 728-0545
FAX (404) 315-6440
EpiInfo@CDC1.CDC.GOV**

Notes

These programs were produced by the Epidemiology Program Office, Centers for Disease Control and Prevention, and the Global Programme on AIDS, World Health Organization, and are provided for use by the public health community. You are encouraged to give copies of the programs and the manual to your friends and colleagues. No warranty is made or implied for use of the software for any particular purpose.

Epi Info is written in Turbo Pascal from Borland International and uses GRAF/DRIVE PLUS ((C) 1989, 90 M.K.Fleming) printer drivers. The manual was written and indexed in Microsoft Word on a Macintosh computer. The cover art is by Mr. Marlon Wolcott. The master disks were tested for computer viruses with the McAfee virus-detection software.(McAfee Associates, (408) 988-3832)

Trade names are used for identification only or for examples; no endorsement of particular products is intended or implied. The use of trade names or trademarks in this manual does not imply that such names, as understood by the Trade Marks and Merchandise Marks Act, may be used freely by anyone.

Telephone Assistance Line

Telephone assistance will be provided if a problem is not solved by reference to the manual. Information for obtaining Epi Info technical assistance is given on the preceding page.

Please send **comments and suggestions** for future versions to:

Andrew G. Dean, M.D., M.P.H.
Epidemiology Program Office, Mailstop C08
Centers for Disease Control and Prevention
Atlanta, GA 30333
Telephone (404) 639-2693
Home Telephone (770) 458-2271
(Emergencies only please)
Fax (404) 639-1546
Internet AGD1@EPO.EM.CDC.GOV

Contents

Using Epi Info in Special Environments

Reference Section

Functional Index

Chapter **1**

How To Use This Manual

> **Tutorial**
> **For a quick introduction to Epi Info, read Chapter**
> **2, install the system as described in Chapter 4, and**
> **follow the tutorial instructions in bold type (like**
> **this one) at the beginning of each chapter.**

Epi Info can be used at several levels depending on your level of experience and needs. The elementary level I allows word processing and entry and analysis of questionnaire data. Classes of students usually reach this level with a few hours of practice.

Learning aids of several kinds are provided. The menu and major programs have help sections, some of which contain major portions of reference Chapters 33, 35, and 36 in this manual. Major programs have interactive tutorials that lead you through major concepts.

Extensive examples are provided in the chapters headed "Example" and as actual programs on the Epi Info disks. These include:

- An Investigation of Paralytic Shellfish Poisoning in Central America
- A Program To Do Nutritional Anthropometric Calculations within a Survey Questionnaire
- A Multi-file Automated Disease Surveillance System with Customized Reporting Formats and Relational File Handling
- Numerous smaller examples represented in the EXAMPLE and TUTORIAL sections of the EPI6 menu

Each of these programs can be modified for use in your own setting and will provide a "flying start" to persons responsible for setting up such systems.

Shorter examples are provided throughout the manual. When a topic is discussed, software examples elsewhere in the manual are cited so that you can see actual applications.

This manual provides several different approaches for learning Epi Info. There are tutorial sections at the beginning of most chapters that provide quick summaries of the material in the chapter. Chapters in the book are arranged in experience levels I through III. Levels I and II are useful for both epidemic investigation and disease surveillance. Level III offers features designed for permanent or research-oriented systems and some aspects require programming skill. Those that require more than simple persistence and experimentation are clearly marked.

If you intend to use Epi Info mainly for processing data rather than for word processing, you can skip Chapter 6 temporarily and proceed to make a questionnaire as described in Chapter 7. Users of all levels who have not used Epi Info before should begin by reading Chapters 6 or 7-9 and doing the tutorials. More advanced features can be learned as needed by approaching the appropriate chapters through the table of contents, the topical index, or the two functional indices at the end of the manual. Some experienced users will make heaviest use of the reference chapters 33-36, where essential facts about commands are presented in somewhat technical but complete form.

Users of Epi Info, Version 5, can begin using Version 6 by reading Chapter 3, "What's New in Version 6." This chapter lists new features and gives summaries of those for which descriptions are provided in other chapters. To learn about all features of Version 6, an experienced Version 5 user should read Chapters 3 and 19-28, and refer to Chapters 33-36 or the help files in the programs for reference. These chapters contain most of the new material added to the Version 5 manual, although many of the other chapters have been reorganized and, we hope, improved.

Advanced techniques for entering and analyzing data are illustrated in application programs on the disks, as described in Chapters 12, 14, 23, and 26. Chapter 29 gives suggestions for those using Epi Info with languages other than English. Chapter 30 discusses equipment and procedures for using Epi Info on a portable computer in the investigation of outbreaks. Technical information on memory size and processing speed are given in Chapter 31. Formulas or references for the statistics in Epi Info are given in Chapter 32.

Notation

A few simple rules apply to reading the manual. Keys on the computer's keyboard are indicated by "< >." Thus, <Enter> is the key labeled "Enter." Material that you type is shown in boldface. If you are instructed to type:

A> **COPY A:*.* B:**

the computer will have supplied "A>". You should type "COPY A:*.* B:" and press the <Enter> key. The <Enter> key is almost always needed to mark the end of an entry, although this will not be shown in the manual each time.

Names of programs in Epi Info, like ANALYSIS, will be capitalized. In the reference sections, the format (syntax) for commands is represented as follows:

> <File name> or <Variable name> means the name of a file or variable, such as:
>
> > **TEST1.REC** or **AGE**
>
> { <File name>} means that entering a file name at this location is optional.

```
 ┌──────┐  Tutorials  Examples  Manual  File  Edit  Setup
 └──────┘
 ┌─────────────────────────────┐
 │██████████████████████████████│
 │ ENTER data              ♫     │  ┌──────────────────────────────┐
 │ ANALYSIS of data        ♫     │  │                              │
 │                               │  │                              │
 │ CHECK customize entry   ♫     │  │                              │
 │ IMPORT files            ♫     │  │   ┌──┐      ┌─┐      ┌──┐     │
 │ EXPORT files            ♫     │  │   │  │ ┌─┐  │ │  ┌─┐ │  │     │
 │ MERGE files             ♫     │  │   Info 6                      │
 │                               │  │                              │
 │ STATCALC calculator           │  │                              │
 │ CSAMPLE analyze surveys       │  │                              │
 │ EPITABLE calculator           │  │                              │
 │ EPINUT anthropometry    ♫     │  │ m 6.03 - June 1995           │
 │                               │  │                              │
 │ VALIDATE duplicate entry      │  │                              │
 │ QUIT                          │  │                              │
 └─────────────────────────────┘  └──────────────────────────────┘

        ┌─────────────────────────────────────────────┐
        │ ♫ = <Ctrl-Enter> for Command-line Parameters │
        └─────────────────────────────────────────────┘
 F1-Help  F10-Quit │ « Create questionnaires or write and edit text
```

<div align="right">

Chapter **2**

</div>

What Is Epi Info, Version 6?

Epi Info is a series of microcomputer programs for handling epidemiologic data in questionnaire format and for organizing study designs and results into text that may form part of written reports. A questionnaire can be set up and processed in a few minutes, but Epi Info can also form the basis for a powerful disease surveillance system database with many files and record types. It includes features used by epidemiologists in statistical programs such as SAS or SPSS and database programs like dBASE. Unlike commercial programs, however, Epi Info may be freely copied and given to friends and colleagues.

There are three levels of facilities in Epi Info for processing questionnaire or other structured data. Using Epi Info on the simplest level, you can computerize a question naire or investigation form in a few minutes by:

- Running the main menu.
- Typing the questionnaire or form with EPED, the word processor, or another word-processing program.
- Entering data in the questionnaire using the ENTER program.
- Analyzing the data using the ANALYSIS program to produce lists, frequencies, cross tabulations, means, graphs, and accompanying statistics.

As your knowledge of the program grows, you will use additional features to shape data entry and analysis to your needs. You will probably want to:
- Insert error checking, skip patterns, and automatic coding in the questionnaire using the CHECK program.
- Select records, create new variables, recode data, manipulate dates, and carry out conditional operations with IF statements during ANALYSIS.
- Incorporate these operations into program files so that they can be performed repeatedly, or by other persons unfamiliar with programming.
- Import and export files from other systems like SAS, SPSS, dBASE, and Lotus 1-2-3.
- Change the names of variables in the data file using the CHECK program.

The third level of features is important if you are setting up a permanent database system, a large study, or want to customize Epi Info's operations to suit special

needs. For such purposes, you can:

- Program the data entry process to include mathematical operations, logical checks, color changes, pop-up boxes, and custom routines written in other languages.
- Specify the format of reports from ANALYSIS to produce customized tables.
- Enter data into more than one file during the same session, moving automatically among several questionnaires within ENTER.
- Link several different types of files together in ANALYSIS so that questions can be answered that require data from more than one file.
- Compare duplicate files entered by different operators to detect data-entry errors.

EPED, the editor included with Version 6, contains a programming system for text called EPIAID that parallels Epi Info's facilities for structured data. Programs are included to guide you through the creation of questionnaires and the design of an epidemic investigation. The text produced can be used as part of an investigation report. EPIAID programs can be written to assist in letter writing or production of reports for other purposes.

Version 6 of Epi Info contains many new features, such as a configurable pull-down menu system, new commands for programming data entry and analysis, capacity to sort and relate very large files, a program to analyze data from complex sample surveys, a new epidemiologic calculator, and a batch processing program for nutritional anthropometry. Files created by previous versions of Epi Info can be used in Version 6. With a few revisions, programs written for Version 5 can be run in Version 6. The programs included in Epi Info are:

EPI6 -- The main menu that allows access to other programs. This program is also a fully configurable menu for use with other software. It includes a built-in editor/viewer, programmable functions for menu choices, and a facility for displaying help files with "hypertext" features.

EPED -- A word processor for creating questionnaires or for general use. A special facility called EPIAID in EPED runs programs that assist in text preparation and in the design of epidemiologic investigations.

ENTER -- Produces a data file automatically from a questionnaire created in EPED or another word processor. Manages entry of data from a questionnaire into a disk file. Allows revision of file format even after records have been entered.

ANALYSIS -- Produces lists, frequencies, cross tabulations, and a variety of other results from Epi Info or dBASE files. Appropriate epidemiologic statistics, such as

odds ratios, relative risk, exact confidence limits, and p values accompany the tables. Stratified analyses with the Mantel-Haenszel technique, one-way ANOVA and Kruskal-Wallis tests, linear regression, and analysis of matched case-control studies are also offered. Records may be selected or sorted using defined variables, "if" statements, and mathematical and logical operations. Graphing, complex report formatting, generation of new data sets for testing, and a programming language are included in ANALYSIS. Data items spread over several files can be linked and analyzed as a single unit.

CHECK -- Sets up ranges, legal values, automatic coding, and skip patterns for the ENTER program, if desired. Mathematical and logical operations between fields, complex jump patterns, access to several files in the same entry process, and use of custom programs provided by the user are also supported by CHECK.

CSAMPLE -- Performs analyses of data from surveys of complex design, taking into account cluster sampling, stratification, and weights.

STATCALC -- Calculates statistics from table values entered from the keyboard. STATCALC performs calculations for single and stratified 2-by-2 tables, sample size, and single and stratified trend analysis.

EPITABLE -- An epidemiologic calculator, containing many useful statistics.

EPINUT -- A program for nutritional anthropometry providing for batch calculation of anthropometric indices.

EXPORT -- Exports data files from Epi Info into 12 other formats used by a variety of database and statistics programs.

IMPORT -- Brings in files from other programs so that they can be used in Epi Info.

MERGE -- Merges files produced from questionnaires with the same or different formats. This allows combining data files entered on different computers, and updating of previous records using data from new entries.

VALIDATE -- Compares two Epi Info files entered by different operators and reports any differences.

Help files -- The entire manual for Epi Info, Version 6, is provided in electronic form with the programs and is accessible from the main menu. Much of the material in the reference section of this manual is also available by pressing <F1> from within Epi Info programs.

Sample Programs -- A complete reportable disease surveillance system, files from two epidemic investigations, and a system for anthropometric calculations in nutritional surveys, are provided as examples of Epi Info use. Three data files from complex sample surveys are provided for use with CSAMPLE. There are numerous other small data files, and many programming examples are provided in the manual.

Tutorials -- Complete interactive tutorials are provided to introduce the features of EPED, ENTER, and ANALYSIS.

Utility -- A program called REC2QES is provided to create a questionnaire from an existing Epi Info .REC file.

Epi Info, Version 6, requires an IBM-compatible microcomputer running the PC-DOS or MS-DOS operating system (Version 2.0 or higher), 512 K (524,288) bytes of random-access memory (RAM), and at least one floppy disk drive. A graphics adapter (board) is necessary to produce graphs. Adapters using the Hercules or IBM monochrome, EGA, VGA, IBM3270, IBM8514, and AT&T standards are supported. Files are provided to produce graphic output for 30 different printers and file formats For most effective use of Epi Info with large data sets, a hard-disk drive, 640 K of RAM, and a color graphics board and monitor are recommended.

Epi Info can make limited use of RAM above 640 K if you have a memory manager such as EMM386 or QEMM loaded. Programs with overlay (.OVR) files will load the overlays from higher memory, meaning that they will run faster, but not necessarily have more RAM capacity.

Data files can consist of as many records as DOS and your disk storage can handle (up to two billion). A questionnaire can have up to 500 lines or approximately 20 screens. The number of variables is not limited, except that they must fit in the 500 lines. The maximum length for a text variable is 80 characters. The total length of variables in one file must not exceed 2048 characters. By linking several related files, you can increase this limit, since the same limits apply to related files. The programs in Epi Info require about 6 megabytes of disk space, but only ANALYSIS, the largest program, exceeds the capacity of one 360 K disk.

Several programs in Epi Info now can be operated with a mouse if your computer is so equipped. A mouse driver, supplied by the mouse manufacturer, must be run prior to running Epi Info if a mouse is used. As in previous versions, however, all of the programs can be operated easily from the keyboard without the use of a mouse. Future versions of Epi Info will be totally "mouse adapted," although EPED, ENTER, ANALYSIS, and CHECK do not support the use of a mouse in version 6.

Versions 1-3 of Epi Info were produced in the Epidemiology Program Office of the Centers for Disease Control and Prevention (CDC). Version 5 was produced by the CDC and the Surveillance, Forecasting, and Impact Assessment Unit, Global Programme on AIDS, World Health Organization (WHO), Geneva, Switzerland. Version 6 has been largely the responsibility of the CDC authors.

The programs are made available by WHO and CDC and are not copyrighted. Making copies for others is permitted and encouraged. If you plan to distribute the programs on a regular basis, the authors would appreciate being notified of the approximate numbers distributed, as a tally is kept of copies known to have been distributed. Testing for computer viruses is advised before distribution and on receipt of programs from any but official sources.
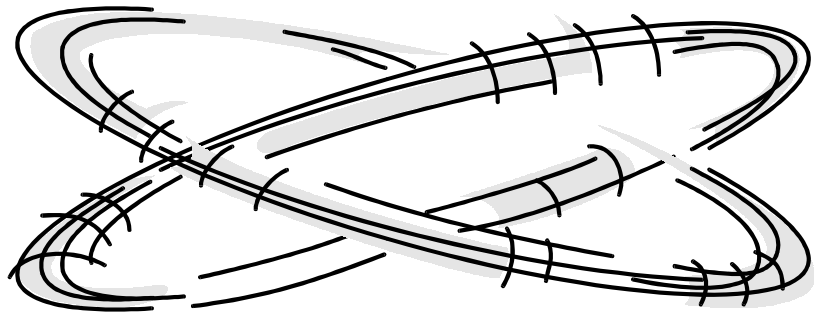
Chapter 3

What's New in Version 6?

New Features in Versions 6.03 and Later

The README.603 (or subsequent version number) file, accessible from EPED, or from the MANUAL menu on the EPI6 menu always contains information about the latest updates, bug-fixes, and improvements. Important changes in Version 6.03 include:

- A program to install an icon to run Epi Info under Microsoft Windows (in DOS mode)
- Many new features in the EPI6 menuing and hypertext system (also known as "EPIGLUE")
- A version of ENTER (ENTERX) that uses higher memory and allows larger .CHK files and more complex data-entry systems
- Improvements in local-area-network compatibility,
- Many minor and some medium-sized "bug fixes"
- Completely rewritten manual chapters 20 and 36-39 and a new index

It is recommended that you read these chapters for details and consult the index of the manual for other features that may have changed. The rest of this chapter describes features common to Versions 6.00 through 6.03.

Compatibility with Version 5

The data (.REC) file formats in Versions 5 and 6 are the same. Data files created in Version 6 are compatible with those from Version 5.

Word processing files made in EPED, Version 5, are compatible with Version 6 of EPED. Those made in EPED, Version 6, may have extra formatting information at the end of the file, but can still be used in EPED, Version 5.

No changes to Version 5 .PGM files should be necessary to make them run properly in Version 6. In Version 5 .CHK files, the BEFORE block, if any, must be changed to BEFORE FILE to run properly in Version 6. Otherwise, you should find that Version 5 .CHK and .PGM files can be run in Version 6 directly, although you will probably want to revise them to take advantage of new features in Version 6.

Features for Local Area Networks

EPED, Version 6, saves its settings to a separate file called EPED.CFG in the currently logged directory so that users on a local area network can each have different settings while running EPED from the network.

ENTER allows more than one operator on a local area network (LAN) to access the same file. Individual records are automatically locked and unlocked at the right time, to prevent problems with data corruption that might occur without this feature. ANALYSIS has a similar feature to prevent problems when two users try to UPDATE the same file simultaneously.

See Chapter 38 for more information on local area networks.

A Completely Configurable Pulldown Menu and Information System

A Configurable Pulldown Menu

The main menu, EPI6.EXE, is also a completely configurable program for developing your own menus. The associated files, EPI6.MNU, EPI6.SCR, and EPI6.HLP, and the configuration file, EPI6.CFG, created through the menu, determine menu choices, submenus, dialog boxes, background appearance ("wallpaper"), and activities of the menu. The .EXE program or the .MNU program may be renamed to create multiple menus in one system, merely by running another .MNU file.

An Information System

The menu has built-in editing and viewing features, that allow it to show text files in multiple windows. The ability to call other programs allows the menu to be used for viewing many types of reports, tables, maps, or graphics from a single control point, merely by designating in the .MNU file the files to be displayed and the programs capable of displaying them. The help facility in EPI6 allows "active words" or "hypertext" to be set up in a text file so that clicking a mouse button or pressing <Enter> will navigate through large text files to select exactly the topics wanted.

The menuing and information system is described in Chapters 5 and 20. The setup features attached to the Version 5 EPI menu are now reached through the ENTER program, and are described in Chapter 8.

New Features in EPED, the Word Processor

Chapters 6, 7, 33, and 34 describe the features of EPED in detail. Functions that are new in Version 6 include:

Configuration Settings

EPED now offers a choice of three MODES--Word Wrap(WW), Text(TXT), and Questionnaire(QES). The abbreviation for the current mode is shown in the lower right corner of the screen. To change modes, press the F6 key to show the SET menu, and use the space bar to change the current setting. <Ctrl-F9> or <Ctrl O-W> will toggle between TXT and WW modes rapidly, a useful feature if you are working in word-wrap mode and want to turn off the word wrap to type in columnar format for a list or series of one-line items, for example.

Each mode comes with a built-in series of settings on the <F6> menu. WW mode is initially set with a right margin of 65, for example, and TXT and QES with margins of 80 (full screen). For any of the modes, you can change the settings; just switch to the desired mode and change the settings. The settings are saved automatically when you exit from the SET menu or change modes after altering a setting.

Reading of WordPerfect Files

Version 6 of EPED reads WordPerfect, Version 5.xx, files directly. They are converted automatically to as close an approximation of the original as possible within EPED, although embedded graphics and WordPerfect tables will not be converted. When you save a file that was originally in WordPerfect format, it will be saved as an EPED file in the ASCII or plain DOS format. You will be prompted to give it another name before saving.

Automatic Reformatting of Text

When Word Wrap is ON, paragraphs will be reformatted or justified automatically after insertion or deletion of text, as soon as a space is inserted or a deletion is made. This feature can be turned off by setting AUTO WORD WRAP to OFF in the F6-SET menu.

Saving of Word Wrap and Other Formatting Information

When creating a file in EPED with Word Wrap ON, saving a file offers a chance to save the current settings in the F6 SET menu in the file. The information will be saved at the end of the file, after the "End-of-File marker" where most other programs will never see it.

If the file is a computer program or data file, the extra characters could interfere with its use in some programs. If you edit such files with Word Wrap off (as should always be the case), EPED will not insert any formatting information. If, by

accident, you have Word Wrap on when saving a file without such information, a prompt will appear to give you a choice so that the plain-character ("ASCII") nature of such files can be preserved.

Option to RELOAD the Last File Accessed

Whenever you press <F10> to leave EPED, the current settings and the name of the current file are saved automatically. If you would like EPED to reload the last file worked with when it starts up again, set the option RELOAD LAST FILE on the F6 menu to ON. If not, set this option to OFF and save the settings. If RELOAD is on, the last mode used (WW, TXT, or QES) will also return when you run EPED, even if no file was active when you quit the program.

Improved Page Markers

In the new EPED, markers appear on the first line of every page on both sides of the screen. The vertical bar for the page markers on the left side of the screen has been eliminated, giving a full 80 columns on the screen.

New Function Keys

Several frequently used functions are accessible using the <Ctrl> key with a function key. To see these functions, hold down the <Ctrl> key and examine the prompt line at the top of the screen. The functions are:

> <Ctrl F1> Begin Block (Repeat twice to begin a Column Block)
> <Ctrl F2> End Block
> <Ctrl F3> Copy Block
> <Ctrl F4> Move Block
> <Ctrl F5> Delete Block
> <Ctrl F6> Find Text
> <Ctrl F7> Replace Text
> <Ctrl F8> Justify--Reformat a paragraph
> <Ctrl F9> WW Change Word Wrap mode to Text (TXT) mode or vice
>   versa

Boxes

Boxes and shading are useful for emphasis in documents, for drawing diagrams, or for constructing menus. EPED, Version 6, allows drawing single or double horizontal and vertical lines or boxes on the screen with the Arrow (Cursor) Keys. Boxes and shading are produced with the arrow keys after adjusting the type of character with the <Home> and <End> keys. Instructions will appear at the top of the screen after you press the<Scroll Lock> key to turn this feature on.

Non-English-Language Characters

To produce a non-English character, hold down the <Alt> key and then use the following keys to produce vowels and other characters with special marks:

Use <Alt> with:	a	e	i	o	u	c	n	y
to produce:	áàâäåæ	éèêë	îîï	óòôöø	úùûü	ç	ñ	ÿ
or								
with <Alt> and <Shift>								
to produce:	Å Ä Æ	É				Ç	Ñ	

The first time the key is pressed, the first of the special characters appears. To produce the others, continue to hold down the <Alt> key and press the same letter key repeatedly until the correct character is obtained. Thus, holding the <Alt> key down and pressing "a" three times will produce "â ". After this, pressing the "e" key will first produce "ê ," with repeated presses of <Alt e> cycling through the various types of "e's." To produce the capital letters, hold the <Alt> key and the <Shift> key. Thus <Alt Shift A> produces Ä and then Å .

Greek and Mathematical Symbols
The IBM character set for microcomputers contains a number of characters that are not part of the ASCII standard, but are quite useful for drawing boxes and typing mathematical expressions and non-English languages. These characters may be produced in EPED after pressing the Scroll-Lock key by typing on the main keyboard.

Choice of Font Styles
Font styles are selected by holding down the <Alt> key and pressing a number (1-7).

> <Alt-1> **Bold**
> <Alt-2> `Double`
> <Alt-3> <u>Underscore</u>
> <Alt-4> Superscript
> <Alt-5> Subscript
> <Alt-6> Compressed
> <Alt-7> *Italic*

To use a particular style, press the corresponding Alt key combination. The color of the text on the screen will change to indicate the new style. Type the text to be printed in the new style. Press the <Right arrow> key to return to the normal text style. (This jumps over the control character that is inserted to the right of the special-style text.)

Output to Various Printer Types, Including Postscript
Printer Definition Files (.PDF) are provided for Epson, IBM Proprinter, Okidata, Hewlett Packard Deskjet and Laserjet, and Postscript printers.. To choose the

correct file, press <F5> to bring down the print menu, and then press <Enter> under "Which Printer" to choose a .PDF file. Postscript printing requires that the file POSTSCPT.INI, supplied with Epi Info, be present in a directory in the current path.

Column Block Operations

In editing tables, data files or screens, it is often important to perform operations on or in vertical rectangles corresponding to columns on the screen. In EPED, Version 6, you can mark a vertical block in the display by using the BEGIN BLOCK command in the same cursor location rather than once as in marking a standard block. After marking a column block, you can use the FIND, FIND/REPLACE, MOVE, COPY, and DELETE commands so that they apply only to the marked block. This is particularly useful for correcting errors in a data file in a particular column, for example, or for moving boxes around on the screen.

Running EPIAID Programs from the DOS Command Line

EPIAID programs, with names ending in .EPA, may be run from the DOS command line by giving the command:

>   **EPED** <File name>.**EPA**

<File name> is the name of the program in 8 or fewer characters. What happens after the program has run to completion can be determined from within the .EPA program. If there is a QUIT command, the program will exit to DOS or, if not, to EPED.

EXISTFILE in EPIAID

To check for the existence of a file in EPIAID, the function EXISTFILE may be used as follows:

IF EXISTFILE <File name> THEN...

where <File name> is the file name including the extension, possibly preceded by the path, for example, D:\ EPI6\ WP\ STUDY.DOC. This feature is useful for writing EPIAID programs that behave differently depending on the existence of a file. If a file does not exist, for example, you may want to display a message, or have the program search in another directory for the file.

•More Convenient FIND and REPLACE Functions

The following options for the FIND/REPLACE functions have been changed:

C=Case Specific	Match upper/lower case of your entry and ignore those that fail to match. The normal default is to match regardless of case.
U=Upward	Look upward in the file from the current cursor

	postition
B=Block	Find "color" only within a selected block of text. This allows confining choices to a selected horizontal or vertical block.

New Features in ENTER and CHECK

New Menu for ENTER
The ENTER program opens with a request for the data file name and a menu giving the following choices:

> 1. Enter/Edit--existing data file
> 2. Create new .REC file--from .QES file
> 3. Revise structure of data file using revised .QES
> 4. Reenter and verify records in existing data file
> 5. Rebuild index (es) specified in .CHK file

For each choice, appropriate file names and other information will be requested if needed. Option 4 is a new feature described below; the other choices were available in Version 5, but are now more accessible through the new menu.

Improved Search and AUTOSEARCH Functions
Searching for the first match is now done with the <F3> key rather than with <F1>. Search results are displayed in a spreadsheet format that allows several records to be displayed. To see more than one screen of such matches--all the SEX = "M" records, for example--press <PgDn> to see the screens after the first. When matches have been displayed, the bar cursor may be moved up and down with the up and down arrow keys, and a particular record selected by pressing <Enter>. To avoid editing any of the matches, press <Esc>.

While a search is occurring, the numbers of the records being searched are displayed. A search may be terminated at any time by pressing <Esc>. To resume the search, use the <F4> and <Shift-F4> keys to go to the next or previous match.

Improved Error Messages
Error messages now include the file name and line number in the .CHK file where the problem was encountered. If the .CHK file is edited in EPED, the line number is that displayed at the bottom of the screen (in TXT mode).

SOUNDEX Searches
SOUNDEX is a method of coding names so that differences in spelling are minimized. "Smith" and "Smyth" are both converted to the same SOUNDEX code, for example, so that subsequent searches will find both names.

If the command, SOUNDEX, is inserted in the .CHK file block for any field, searches on that field will be done by the SOUNDEX method. The original string will be stored in the field, however, not the SOUNDEX string. An index for the field containing the SOUNDEX command will automatically be created in ENTER using the SOUNDEX keys so that a search can proceed using these keys. The search key entered on the screen will be SOUNDEXed before the index is searched.

Specifying Fields for AUTOSEARCH

The AUTOSEARCH command in a .CHK file can be followed by a list of fields to match on. If no fields are specified, AUTOSEARCH acts as in Version 5, searching on all non-blank fields except <TODAY> and NOENTER fields.

AUTOSEARCH can specify whether or not to use available SOUNDEX keys. AUTOSEARCH SOUNDEX will use the SOUNDEX keys on all search fields which have them; AUTOSEARCH alone will use only the normal keys.

INCLUDE Command in .CHK Files

Portions of a check file can be kept in separate files and referenced using the INCLUDE command. The statement INCLUDE <File name> in a block of a .CHK file will cause the named file to be included in the check file as though its contents were part of the original check file.

The INCLUDEd file is processed exactly as if it appeared in place of the INCLUDE command. When its processing is completed, processing of the previous file will continue exactly as if it had been the only file, and the INCLUDEd file had appeared within it.

HELP File Compiler

A new program, HELPPREP, is supplied to compile help files and make possible the display of text blocks with the HELP command.

Syntax in .CHK file:

      HELP <File name> <Topicno> <Col> <Row> CONTEXT

If HELP is used with a file name, the file referred to must be in the Epi Info Help File format, consisting of text, optional highlight commands, and markers that divide the file into topic sections. <Topicno> is an integer that identifies the Topic block within the file. The help file must have been indexed by using the HELPPREP program, as described in Chapter 37. CONTEXT-specific help may be supplied for any questionnaire by constructing a help file and inserting appropriate HELP commands in the .CHK file.

DEFINEd Variables (Passing Variables Between Records or to Related Files)

DEFINE allows new variables to be defined in a .CHK file as they are in ANALYSIS. These variables may be used to hold intermediate values in calculations, to carry over items from one record to the next, or to transfer the contents of a field from one file to a field in a related file. This is illustrated in the Relational Hierarchy EXAMPLE accessible from the main EPI6 menu.

Variables declared as GLOBAL can be accessed by related files; otherwise, variables can be accessed only by the file which DEFINEs them. Variables declared as GLOBAL or CUMULATIVE keep their values when new records are read, and should be initialized either in the DEFINE statement or by a LET statement before their first use. Other variables are set to missing each time a record is read.

For more information, see Chapters 22 and 35.

USEing the Same Codes and Legal Values in More Than One Field (to Save Random Access Memory)

CODES, LEGAL, COMMENT LEGAL, JUMPS and RANGEs can be listed in a single field but referenced in others with the USE command. Since the list is then loaded into memory only once rather than multiple times, a larger and more complex questionnaire can be accommodated than would otherwise be possible.

This feature is further described in Chapters 22 and 35.

Codes or Legal Values in Files Rather Than in RAM

Large code or legal-value lists can be put into external (.REC) files rather than in the .CHK file. To do this, use the syntax CODES <File name>.REC,where <File name>.REC is the name of the .REC file. Each .REC file containing codes must have only two variables, both of which must be KEY. The first variable is for the first value (the one the user enters). The second variable is for the coded value. (Usually, the first variable is for the field that declares CODES, the second is for its CODEFIELD.)

The EXIT Command

EXIT is used to terminate processing of a sequence of commands, as when one error condition has been determined in a sequence of several checks. EXIT can be used to move the user's focus from an error message to the appropriate field for error correction. EXIT can also be used to avoid an endless line of ELSE IF ... ELSE IF ... ENDIF ENDIF ENDIF ....

Commands for Actions at a Particular Time

Sometimes is is desirable to perform an action such as a calculation or presenting a help message at a specific time during data entry that is not the standard time for

activation of a block of .CHK commands--i.e., after an entry has been made in a field and the <Enter> key has been pressed.

The following commands control actions so that they occur before or after a file, record, or field is accessed:

BEFORE FILE
AFTER FILE
BEFORE RECORD
AFTER RECORD
BEFORE ENTRY
AFTER ENTRY

These commands are used in blocks, starting with the command itself and ending with END. BEFORE ENTRY and AFTER ENTRY are always contained within a block for a field within the questionnaire.

The TYPE COMMENT Command
This command allows confirmation of abbreviated entries on the screen. If the user enters "M" for example, TYPE COMMENT may add, from the list of LEGAL or CODE values, "ale," so that the word "Male" appears to tell the user the meaning of the code just entered. Only "M" will be stored in the data file, however, conserving disk space.

The syntax is:

TYPE COMMENT <Horizontal location> <Vertical location> <Color>

HIDE and UNHIDE
HIDE hides a field and its prompt from view and makes it NOENTER. UNHIDE returns it to its status before the HIDE. These commands are useful to remove entire questions from the screen in response to a particular input, perhaps removing questions on symptoms if the subject was not ILL.

Double Entry Verification
ENTER offers a rigorous method of doing data validation. After records have been entered and saved in a file, option 4 on the ENTER menu, "Reenter and verify records in existing data file," allows a data-entry operator to reenter each record and verify the data entered the first time.

The operator enters data exactly as for new records. When a field entry matches the data in the file, the cursor moves exactly as for new entries. When an entry does not match, however, a message appears, and the verify operator is given a chance to reenter the field or to compare the original entry with the new one and make a

choice.

This and other methods for data verification are described in Chapter 19.

New Features in ANALYSIS

An Editor for .PGM Files Accessible From Within ANALYSIS
Typing EDIT in ANALYSIS brings up an editor with the last program run displayed for editing. This provides a convenient way of editing programs without leaving ANALYSIS.

SORT and RELATE No Longer Limited by RAM Space
In Version 5, the size of files that can be sorted or related is limited by available RAM space. In Version 6, the sort and relate functions use hard disk space automatically instead of RAM if this is necessary, and the size of the files that can be handled is limited only by available disk space.

We have experimentally related and sorted files containing 1.5 million records (although it took about an hour per million records on a 486 computer). A similar computer did a frequency on 560,000 records in about 2.5 minutes.

Use of @Variable Values in Expressions
In Version 5, the "@" character followed by a variable name can be used in the TYPE and TITLE commands to insert the value of a variable rather than a literal value. This has been extended in Version 6, so that you can use expressions like ROUTE @FILENAME after getting user input for FILENAME, for example. You might also set a variable COREFIELDS equal to "AGE SEX RACE COUNTY DISEASE" and then save typing by giving the command LIST @COREFIELDS rather than LIST AGE SEX RACE COUNTY DISEASE. Different files may be read with READ GEPI@YEAR. This feature is not available in .CHK files.

TO in Lists of Variables
 In the LIST, FREQ, WRITE, and TABLES commands, you may use the word TO to indicate a series of variables in the order in which they appear in the questionnaire and in the Variables display (<F3>). LIST AGE TO COUNTY will list the variables AGE, COUNTY, and all those in between, perhaps AGE SEX RACE OCCUPATION COUNTY.

Change in Function Keys
Function keys will no longer cause the text commands "QUIT" or "BROWSE" to appear in the output of ANALYSIS.

Unlimited Line Length of Output with WRITE

The maximum size of lines output to the printer by the WRITE command is now unlimited, although individual printers will have limits for printing long lines.

PROCESS Command Passes Through File Without Producing a Table or List

This command causes ANALYSIS to pass through the current file and to perform all calculations, but no table is produced.   Output, if any, must be programmed with TYPE or similar commands.

Continue-line character "\ " may be used both on the screen and in program files.

In Version 5, the line continuation character was allowed only in program files. For convenience, it can be used during interactive programming on the screen as well.

User defined or "Add-in" commands

Version 6 of ANALYSIS allows the user to define additional commands, thus extending the language. The commands added may take the form of DOS executable programs, DOS batch files or a user-defined set of ANALYSIS commands. These added commands may be used in any place where the built in commands are used, including BREAKCMD and IF statements, thus effectively allowing for multiline statements. The CMD command can also be used to implement functions or subroutines that can be called from within other commands, such as IF statements.

The CMD command may take one of two forms, depending upon whether the command being added is an "external" (DOS) command or a set of ANALYSIS commands. The first, form is

```
  CMD <COMMANDNAME>
     <COMMAND>
     <COMMAND>
     <COMMAND>
   END
```

In the second form, a subroutine called ENDOFPAGE is being defined to be called later in the BREAK command.

```
Example:
      SORT COUNTY
      CMD ENDOFPAGE
        TYPE "Subtotal = @SUM(COST)"
        NEWPAGE
      END
      SET BREAKCMD = ENDOFPAGE
```

BREAK COUNTY
PROCESS

Improved Programmer's Interface

The CMD command can also be used to run an external program from within ANALYSIS and to return the results to ANALYSIS. Programmers can use this form to add an entirely new program to ANALYSIS and define a new command through which the program can be accessed as though it were an integral part of ANALYSIS. Further details are given in Chapter 39.

CMD can also be used to produce a subroutine or procedure in ANALYSIS, that can be called from another part of the .PGM file, either for every record (as happens when it is in an IF 1=1 statement, for example), or as needed during the program. It can be used to implement multi-line consequences of IF statements or to set up a command of more than one line that can be called from the BREAK command by using SET BREAKCMD = <Name of command created with CMD>.

Improvements in UPDATE

NoEnter, IDNUM, and TODAY fields and fields that are defined as KEY in a .CHK file cannot be modified in UPDATE. The TodayType field is set to today's date when an UPDATEd record is written to disk. Data may be entered in any field when entering a new record. When leaving a record that has been changed, either by positioning to a new record, or exiting UPDATE, the user is asked if the updated record should be written to disk.

When BROWSEing or UPDATEing records that have been sorted, the records are displayed in sorted order. If a change is made that would affect the sorted order of the records, the change in order would not be noticed until the records are again sorted.

Fields in related files may also be modified in UPDATE. If a record is added to the main file, a record with the appropriate relational field is added to the related file, in addition to the related fields that have been entered. The files are not re-RELATED automatically, however; the related records that are added would not appear until the user RELATEs the files.

Error Messages Containing Line Numbers in .PGM File

Error messages now include the file name and line number where the problem was encountered. If the .PGM file is edited in EPED, the line number is that displayed at the bottom of the screen (in TXT mode).

Improved Output for Graphic Images

Graphs in ANALYSIS require "drivers" for printing with various kinds of printers. The drivers (.BGI files) in Version 6 of Epi Info are different from those used with

Version 5, and have names beginning with "$", for example, $OKI92.BGI for Okidata 92 printers.

If output is routed to a file, the drivers can be used to produce graphs in various formats for transfer to other graphics programs where they can be edited and enhanced. To use any graphics driver, use SET PRINTER = <Driver name>, where <Driver name> is the name of the driver minus the initial $ and the .BGI extension. SET PRINTER = OKI92 will use $OKI92.BGI, for example. The available drivers are the same ones supplied with Epi Map, including:

```
File name       Size   Format
$BMP      BGI   17768 BMP (Windows 3.x) Format
$CGM      BGI   11818 Computer Graphics Metafile
$FX       BGI   20894 Epson FX80 Printer
$PS       BGI   15518 Postscript
$TIF      BGI   20862 Compressed TIF Format
$UTIF     BGI   20862 Uncompressed TIF Format
$WPG      BGI    9642 WordPerfect Graphics
$CFX      BGI   21128 Epson 9-pin Color Printer
$CLQ      BGI   23726 Epson 24-pin Color Printer
$DJ       BGI   21792 Hewlett-Packard Deskjet
$DJC      BGI   22252 Hewlett-Packard Color Deskjet
$DXF      BGI   16084 AutoCad DXF Format
$HP7090   BGI   16264 Hewlett-Packard Plotter 7090
$HP7470   BGI   16232 HP Plotter 7470
$HP7475   BGI   16336 HP Plotter 7475
$HP7550   BGI   16336 HP Plotter 7550
$HP7585   BGI   16368 HP Plotter 7585
$HP7595   BGI   16368 HP Plotter 7595
$IBMQ     BGI   20220 IBM Quietwriter Printer
$IMG      BGI   20402 GEM IMG Format
$LJ       BGI   23146 Hewlett-Packard Laserjet
$LJ3R     BGI   23350 Hewlett-Packard Laserjet III
$LQ       BGI   23492 Epson 24-Pin Printer
$OKI92    BGI   18064 Okidata 92 Printer
$PCX      BGI   17616 PCX (640 x 480) Format
$PJ       BGI   24940 Hewlett Packard Paintjet
$PP24     BGI   23562 IBM Proprinter, 24 Pin
$TSH      BGI   22306 Toshiba 24-Pin Printer
$VSHO     BGI   13112 ANSI NAPLPS/Video Show Format
```

COMBINE Command Combines or "Normalizes" Data in Multiple Fields in Same File

Syntax:

COMBINE { NAME=<Namefield>} VALUE=<ValueField> <Var name><Var name> {|<Var name>|}

COMBINE { NAME=<Namefield>} VALUE=<ValueField> <Var name> TO <Var name>

COMBINE causes the contents of several fields to be treated as though they were in the same field. Suppose, for example, that you have several fields in a case questionnaire as follows:

      Diagnosis1 _____

```
                    Diagnosis2 _____
                    Diagnosis3 _____
                    Diagnosis4 _____
```

You would like to know how many times "pneumonia" appears as a diagnosis among all the cases. Since it may appear in any of the four fields, you would have to write an IF statement to count pneumonia if it appears in any of the four fields, or even a program to create new records for each diagnosis rather than for each "case." In database parlance, this is called "normalizing" the database. It may be thought of in simplistic terms as "putting all the data you want to count or sum into the same field."

In a foodborne outbreak, the following fields might appear:
```
              Chicken          <Y>
              Beef             <Y>
              Salad            <Y>
```

Using combine, you can achieve the effect of having one field called FOOD containing the names of the foods eaten.

Further details are given in Chapter 13 and 36.

FORM Prints Questionnaires or Forms Containing Data
Many users have requested a method for printing forms containing data. The FORM command provides a way of doing this within ANALYSIS. This command itself causes a questionnaire to be printed that contains data from the current record. It may be combined with the IF and PROCESS commands to print any or all of the records in the file in questionnaire format.

Features for Making Menus in .PGM Files
The PICKLIST command causes a list of items to be displayed on the screen so that the user may move the cursor to the desired item and make a choice by pressing <Enter>, allowing the construction of interactive menus with popup choice boxes in .PGM programs.

International Language Support
Enhancements have been made to ANALYSIS in order to better support users in countries that use other than the default DOS character set. Specifically, if an alternate code page is installed in DOS (refer to your DOS manual for instructions, "Customizing for International Use"), ANALYSIS will use the uppercase conversions, and collating sequences appropriate for the country for which the code page is designed. String comparisons will also use that country's collating order. International currency symbols, thousands separators, Japanese dates, and multi-character character sets are not directly supported by ANALYSIS.

CLOSE Command
To delete a file safely, the file must be closed. The CLOSE command has been added to allow the user to safely close the file before deleting it.

The CLOSE command can also be used to remove the effect of a RELATE command ("unrelate" a file), so that a new relation can be set up.

New Statistical Features

- A program for analysis of complex (cluster) sample surveys is included. See chapter 14 for a complete description.
- The FREQuency command displays confidence limits for binomial proportions for numeric data.
- The MEANS command can be used with a single variable.
- A t-test statistic has been added to the MEANS command.
- Maximum-likelihood estimates of the odds ratio and exact confidence limits are now produced routinely in ANALYSIS.
- The mid-p statistic produces a less conservative version of the Fisher exact test for small numbers.
- A t-test to test difference from zero has been added to the FREQ command so that differences between matched pairs can be tested.
- A new statistical calculator called EPITABLE offers many new statistics, including vaccine efficacy, screening statistics, random numbers, and probability statistics

New Features in Other Programs

- IMPORT, EXPORT (formerly CONVERT), MERGE, and VALIDATE have been completely rewritten to provide better and faster function and more modern control screens.
- EXPORT produces files for MULTLR (a logistic regression program) and for SYSTAT and EGRET, two commercial programs for epidemiologic statistics.
- IMPORT and EXPORT are 50-700% faster, depending on the file format.
- EpiNut provides batch processing and other facilities for anthropometric statistics.

New Help and Sample Programs

- The entire Epi Info Manual is included on the disks.
- NETSS, a new relational public health surveillance system for state health departments illustrates many aspects of Version 6 programming, including producing hypertext files as output from ANALYSIS See Chapter 26.
- There are new tutorials on making a questionnaire, entering data, and processing data from an epidemic using the case-control design.
- Numerous sample programs and files are accessible from the main menu.

```
              Welcome to Epi Info

      This program will INSTALL or COPY the Epi Info System
 ╔════════════════════════════════╦════════════════════════════════╗
 ║ Which drive is the SOURCE drive--║ Which drive is the DESTINATION ║
 ║ the one where the original files ║ drive for installing or copying--║
 ║ are to be found?                 ║ the one to receive the files?  ║
 ║                                  ║                                ║
 ║    A: (Must be a letter)         ║    C: (Must be a letter)       ║
 ║                                  ║                                ║
 ║ (The files must be in the current║ Does this drive have REMOVABLE ║
 ║  directory of the SOURCE drive.) ║ (USUALLY FLOPPY) disks (Y/N)? N║
 ╚════════════════════════════════╩════════════════════════════════╝
      Would you like to INSTALL the system for use, or COPY it for further
                    distribution (enter "I" or "C")? █




                        Press <F10> to Quit
```

34

Chapter 4

Installing Epi Info

> **Tutorial**
> **Summary of Installation**
>
> **Place Disk 1 of the Epi Info System in drive A (or other diskette drive). Type the appropriate drive letter and INSTALL, for example:**
> **A:INSTALL**
> **and follow the directions given by the program, indicating that you wish to *install* rather than *copy* Epi Info. If you are installing Epi Info on floppy disks, you will need to format the disks for your computer before running INSTALL, as described below in the section on Floppy Disk Installation.**

Overview

The programs on the Epi Info distribution disks are in compressed format and they must be expanded before the programs are run. A program called INSTALL is provided on Disk 1 and this program should always be used for installing the system. INSTALL will perform either of two functions:

1. It will *install* Epi Info for use on either hard or floppy disks.
2. It will make *copies* for distribution in the compressed format on any size disk.

If you have a computer with a hard (fixed) disk, installation of Epi Info is as simple as the instructions given in the Tutorial section above. Epi Info can also be run from 720 K or 1.44 megabyte 3 1/2-inch disks often used with laptop computers. If you have a computer equipped with two 360 K floppy disks and no hard drive, you should probably consider buying a hard drive or using Epi Info, Version 3 or 5. Although individual programs can be used from floppy disks in Version 6, the convenience of a hard drive is well worth the price.

Installation

To install Epi Info, start your computer in the usual way until the operating system prompt is seen.

C> (or D>, etc.)

(If you will be installing onto floppy diskettes, see the section below on formatting diskettes.) Place Disk 1 of the Epi Info system in drive A (or other floppy-disk drive). Now type the drive letter and a colon, **A:**, for example, press <Enter> and then type **INSTALL**:

C>**A:**   (And press <Enter>)

A>**INSTALL**   (And press <Enter>. We will assume this for DOS commands in the rest of the manual rather than mentioning it repeatedly.)

The program will ask for the following information:

What drive is the SOURCE drive--the one where the original files are to be found?

Usually this will be Drive **A**, where you have inserted the first Epi Info distribution disk.

Which drive is the DESTINATION drive for installing or copying--the one to receive the files?

If you are installing the system on a hard disk, this will be the correct letter for the hard drive, usually **C** or **D**. To install on floppy disks for use in a computer without a hard drive, such as a laptop computer for field use, give the letter of another floppy disk drive, usually Drive **B**.

Does this drive have floppy (or other removable) disks (Y/N)?

You should answer "Y" if the destination disk is removable, particularly if you intend to use the installed system on a computer other than the one you are using for installation. You may, for example, be making a set of installable disks for a laptop computer or for distribution to others.

For a normal Epi Info installation, press <Enter>.

If you press <Enter> at this point, the installation directories will be \EPI6 and \NETSS on whatever drive you specified. If you prefer to install Epi Info in directories that are not directly off the root directory, such as \APPS\EPI6 and \APPS\NETSS on a local area network, press escape and follow the directions to enter the "parent" directory under which the two Epi Info directories will be created. For most purposes, the simple option obtained by pressing <Enter> is preferable. If you choose the other option, see Chapter 38 for instructions to create a DOS environment variable called EIDIR that contains the "parent" directory name.

Would you like to INSTALL the system for use, or COPY it for further distribution (enter "I" or "C")?

You should answer "I" unless you are making disks for further distribution. If you answer "C", the resulting disks will contain compressed files that cannot be run until they are installed.

Select Video Drivers.

Choices are presented for **CGA or MCGA, EGA or VGA, IBM8514, IBM 3270 PC, AT&T 6300, Hercules Monochrome, and "Install all."** INSTALL will attempt to detect the type you have and indicate it by the position of the bar cursor, but if there is any doubt, press <F8> to select all. The only problem with this is that the entire set of graphics drivers, contained in files labelled .BGI, takes up more disk space (about 40 K) than a single driver. If the computer is a laptop, it probably has either a CGA or EGA/VGA graphics adapter. You should be able to find this information in the manual for the computer.

Select Print Drivers

Printer drivers are files that assist Epi Info in printing graphics on a variety of printers or in outputting graphics to files formatted for other systems. You may load one or more drivers depending on anticipated future needs. If you plan to print graphs on a Hewlett Packard Laserjet printer, for example, load the driver for that printer. You might also choose the drivers for .CGM and WordPerfect Graphics files if you will be exporting graphics to Harvard Graphics or including them in WordPerfect documents. In ANALYSIS, you will use the SET PRINTER command to specify the driver used for a particular task.

After selecting printer drivers with the arrow keys and <spacebar>, press <F4> to continue.

Select Groups to Install

The Epi Info files are divided into *groups* on the disks. The files in a group function best if they are on the same disk, but you can choose to leave out particular groups, such as the sample surveillance system, NETSS, during installation. To select all groups, press <F8>.

If you have previously installed part of Version 6 and wish to install an additional part of the system or were interrupted during a previous installation, choose the groups that you wish to install.

After choosing the groups, press <F4> to continue.

The INSTALL program will prompt you to insert a disk containing each file group as it is needed. When prompted, insert the necessary disk and then press <Enter>. At any time except during file copying, you may press <F10> to stop the installation and leave the INSTALL program.

Creation or Editing of AUTOEXEC.BAT and CONFIG.SYS Files

Your computer system may already contain special files called AUTOEXEC.BAT and CONFIG.SYS that tell DOS to perform operations automatically whenever the computer is first turned on ("booted.up"). Several statements need to be added to these files for proper operation of Epi Info, particularly if you have a hard disk.

The necessary statements are:

In the AUTOEXEC.BAT file: A PATH statement that includes a path to **\EPI6** on the drive where the Epi Info system resides. Previous paths should remain.

In the CONFIG.SYS file: The statements **FILES=20** and **BUFFERS=20**. Numbers greater than 20 are fine.

During installation on a hard disk, INSTALL may ask if these files should be changed automatically. If you type "Y" to signal your agreement, the previous files, if any, will be changed to "AUTOEXEC.OLD" and "CONFIG.OLD." New AUTOEXEC.BAT and CONFIG.SYS files will be created containing the necessary additions as described above. If you are installing on diskettes, a CONFIG.SYS file with the necessary statements is placed on the first diskette if it does not have one.

Registration for Program Updates and Newsletters

During installation, information is provided for registering your copy of Epi Info. If you obtained the system directly from the Epi Info supplier mentioned on the screen, there is no need to register, and you can press <Enter> and begin to run Epi Info as described in the next chapter.

If you obtained a copy of Epi Info from another source, registering your name and address gives you the opportunity to be on the mailing list for future updates and occasional newsletters. It also tells us how well the informal copy distribution system has been working and how many copies have actually been received. In contrast to commercial programs, Epi Info is designed to be distributed by copying from person to person. Please register, and then encourage others to copy your original disks and the manual. Before distributing copies, however, it is a good idea to use a commercial program to check the disks for viruses that may have crept in as a result of passage through other computers.

Installing Epi Info to Run Under Microsoft Windows

After DOS installation is complete, a message will appear providing information on installing•icons and other items to make it easy to run Epi Info from the Microsoft Windows desktop. Even if you do not choose this option, you can always run Epi Info 6 from Windows using Window's MSDOS feature, but it is convenient to have an "Epi 6" icon on the desktop and be able to run Epi Info merely by clicking twice on the icon.

If you have Windows and wish to install the Epi 6 icon and create an Epi 6 program group, run Windows as usual. From the FILE menu, choose RUN. Indicate the drive and directory where your Epi Info system has been installed and the SETUP program's name, e.g. **C:\EPI6\SETUP.EXE**. Press <Enter> or click OK and you should soon see a new program group window containing the Epi 6 icon.

To run the Epi 6 menu, click twice on the icon. From now on you have the choice of running Epi Info from Windows (as a DOS program), or directly from DOS by typing EPI6.

If you are running Windows from a local area network rather than from your own hard disk and have difficulty installing the icon feature, ask your network administrator for advice, as the network may not allow changes to Windows except by the network administrator.

Completing Installation

The INSTALL program will offer brief advice on running Epi Info, and you are ready to run the main menu, although you may wish to set up printer options as described in the next section at this point.

Setting Printer and Standard Disk Directory Options

Most Epi Info results can be printed on any standard IBM- or Epson-compatible printer. If yours is a dot-matrix printer, no further setup is necessary to print graphs produced by ANALYSIS. If you have a laser printer or plotter, however, you should create a file called CONFIG.EPI that contains one of the following commands:

SET PRINTER = EPSON	(for Epson/IBM-compatible dot-matrix printers)
SET PRINTER = LQ1500	(for Epson LQ1500-compatible printers)
SET PRINTER = HP	(For Hewlett Packard-compatible laser printers)
SET PRINTER = PLOTTER	(For Hewlett Packard-compatible plotters)

Laser printers require output in pages of 60 lines rather than the 66 lines usually used by dot-matrix printers. If you have a dot-matrix printer, you should add to your CONFIG.EPI file the command **PLINES = 66.** Printers connected to a serial, rather than parallel port, also require commands for PORT, BAUD rate, and PMODE as described in Chapter 36 in the section on the SET command.

In the EPED program, you should also set the page length in the SET menu to 60 for laser printers, or 66 for many dot-matrix printers. The page length can be set for each of the three modes in EPED--TXT, WW, and QES. Each time you exit from the <F6> SET menu, be sure to choose SAVE SETTINGS to if you have made changes.

Formatting Floppy Disks

Before running the INSTALL program to copy or install Epi Info onto floppy disks, you should first *format* the required number of disks. This is not necessary if you are installing Epi Info onto a hard (fixed) disk.

Usually formatting is done with a DOS program called FORMAT.COM, but you may have other methods if your computer has been customized for easier use. Instructions for using the standard DOS FORMAT command are given here in case you do not have convenient access to the DOS manual or other information.

To format a blank disk in a system without a hard disk, using the FORMAT command in DOS, place the DOS "boot" disk for your computer in drive A. Start your computer in the usual way until the operating system prompt is seen:

A>

If the prompt is not A>, enter A: and press the <Enter> key. Place a new, unformatted disk in drive B. Use the DOS command FORMAT to format the disk and copy the operating system:

A>**FORMAT B:**

For use on an all-floppy computer, it may be helpful to copy COMMAND.COM to each diskette:

A>**COPY A:COMMAND.COM B:**

After typing the command, press the <Enter> key on the right-hand side of the keyboard to tell the computer that you have finished entering a command.

Format enough disks for the Epi Info system to hold all the files that you intend to install. The entire installed system takes about 6 megabytes of disk space. The executable programs occupy about 3.2 megabytes.

A CONFIG.SYS file containing necessary statements is included on DISK 1 of the Epi Info system. It will be placed automatically on disk 1 of your installed floppy system.

Special Considerations for Floppy Disk Installation

As noted above, the Epi Info distribution disks cannot be used directly, and working disks must be prepared on disks of appropriate type for your computer by running the INSTALL program. You can install executable files for Epi Info on floppy disks, but the ANALYSIS program requires more than the 360 K of storage space available on disks of low capacity. Disks of 720 K capacity are adequate. The computer used for installation must have drives capable of reading the SOURCE and DESTINATION disks.

In a computer having floppy disk drives only, the DOS file COMMAND.COM must be present at the time the computer is turned on and at other times when part of DOS must be loaded into memory. To prevent having to insert the DOS disk at these times, it is desirable to have COMMAND.COM present at all times, although DOS will ask you to insert the correct disk if disk space prevents putting COMMAND.COM on each program disk.

We recommend that you set up the disks as follows for a floppy-disk computer:

> Drive A: A disk containing the DOS operating system that is otherwise empty and is used to hold data or word processing files.

> Drive B: One of the Epi Info disks produced by the INSTALL program. Depending on the capacity of this disk, you may have to insert other Epi Info disks to access all the programs.

To use this configuration, you should first run the main menu, press <F9>, and set the standard disk directory or default path, as described in the next chapter, to "A:\". This will cause the Epi Info programs to assume that all data files are on Drive A and you will not have to specify this each time a file name is chosen.

The statement "FILES=20" in the CONFIG.SYS file allows up to 20 files to be open at one time, rather than the 4 files that most versions of DOS allow automatically. This is important if you will be operating a surveillance system or using the relational file features of Epi Info with several files open at one time. The statement "BUFFERS = 20" helps to increase the speed of disk access. These statements should be inserted in a CONFIG.SYS file that is copied onto each disk containing the DOS operating system. Creating these disks is discussed below.

It is very important that all floppy disks be labelled precisely. Use a felt-tip marker to avoid damaging the disk and include information on groups, files, dates, versions, and other details on each label.

Making Copies of the Epi Info Disks for Further Distribution

As mentioned above, the INSTALL program can also make copies of Epi Info disks in compressed format. This is particularly useful if you want to produce a distribution copy on disks that differ in size from the original. INSTALL can even produce distribution disks from a copy on a hard disk, if that copy was in turn copied in compressed format from the distribution disks.

If you will be copying to floppy disks, prepare enough formatted disks to hold the entire system. To produce a copy in compressed format on floppy or hard disks, run the INSTALL program, give the SOURCE and DESTINATION drive letters, and indicate that you wish to *Copy*, not *Install*, the system. No information about graphics adapters is necessary. The program will prompt you to insert diskettes containing the appropriate groups.

Copies can also be produced by the DISKCOPY command in DOS if you are copying onto disks of the same size and density as the source disks.

Solving Installation Problems

If problems occur during installation, be sure that there is room on your hard disk for the new programs and that any floppy DESTINATION disks have been properly formatted for the drive in which they are placed.

The installation program checks each file as it is expanded from the compressed format, and reports any files that have been corrupted. If corrupted files are reported, try the installation again, and if the same thing occurs, call or write to the source of your disks for a replacement or contact the Epi Info Hotline listed on the title page of this manual. If you obtained your programs from a bulletin board or network, be sure to check for computer viruses using one of the available commercial programs.

Epi Info 6

Version 6.03 – June 1995

Tutorials   Examples   Manual   File   Edit   Setup

♫ = <Ctrl-Enter> for Command-line Parameters

F1-Help   F2-New   F3-Open   F4-Close   F5-Print   F8-Menu   F9-Save   F10-Quit

Chapter **5**

Running Epi Info

<div style="border:1px solid black; padding:10px;">

Tutorial

Start your computer in the usual way until you see the DOS prompt:

C:\ >

If you have a hard disk, type:

CD \EPI6

and press <Enter>.

Type:

EPI6

and press <Enter>. You should see the main Epi Info menu.

If you have a mouse, use it to place the cursor on various menu items and press the left mouse button to select them. If not, use the arrow keys to move around and press <Enter> to select an item. Be sure that <Numlock> is off.

Press the <Enter> key to run EPED, or select another program using the down-arrow key and then press <Enter>.

If you are using floppy disks and a program is not present, insert the appropriate program disk and try again.

To return to the EPI menu from another program, press <F10>. To return to DOS from the EPI menu, press <F10>.

</div>

Turn on your computer and "boot" the DOS system in the usual way. If you have a hard disk, this should happen automatically. If the computer has only floppy disk drives, you should insert a disk containing the DOS system in Drive A and the first Epi Info disk in Drive B as recommended in the chapter on Installation.

If your computer has been provided with custom software, you may see a menu with Epi Info as one of the choices. In most computers, however, you should see the DOS prompt as:

C:\>

If you are using a hard disk, type **CD \EPI6** to change to the Epi Info directory. (If you chose another directory for installation or have a custom menu, this step may be slightly different. If you have a custom menu, for example, you may need to exit to DOS before the next step.)

Now type EPI6

C:\ >**EPI6**

and you will see Epi Info's main menu, shown at the beginning of this chapter.

At the top of the menu are the choices:

- PROGRAMS--The main programs of the Epi Info system
- TUTORIALS--Interactive tutorials to introduce the main concepts of Epi Info
- EXAMPLES--Sample files that illustrate particular aspects
- MANUAL--The entire manual for Epi Info in easily accessible form
- FILE--Functions to open, create, view, or print text files from the EPI6 menu
- EDIT--Functions to edit, copy, cut, and paste text files opened with the FILE menu
- SETUP--Functions for setting up the menu and the making of questionnaires in ENTER

To select an item from the menu:

1. If you do not have a mouse, move the highlighted bar with the arrow keys. When the correct item is highlighted, press the <Enter> key to run the program. If the cursor leaves the menu bar, press the <Alt> key to make it reappear.

2. If you have a mouse, you may move the cursor with the mouse and press the left mouse button to select an item.

3. You may also type the highlighted capital letter of the selection, for example, **E** for the **E**PED editor, or **N** for E**N**TER, and then press <Enter>.

Pressing the <F10> key will exit from the menu or any other Epi Info program. Leaving the menu will bring back the DOS prompt unless your computer has a customized menu system.

Command-Line Parameters

Any of the Epi Info programs may be run directly from DOS rather than from the Epi menu. Some users find this faster than using the menu; it also saves the small amount of memory space (about 12 K) that is used by the menu.

Each of the programs has a set of "command-line parameters" that gives information normally obtained through interactive screens in the program. For example, to run EPED, the word processor, and work on the file called MEMO1, you would type

C>**EPED MEMO1**

from the DOS prompt. EPED will then automatically display the file called MEMO1 on the screen ready for editing or additions.

From the Epi6 menu, you can also enter command-line parameters by pressing <Ctrl-Enter> while the bar cursor is on the desired program choice in the menu. A dialog box will appear so that you can enter the parameters, for example, JONES.DOC to edit a file by that name if the bar cursor is on EPED in the PROGRAM menu. The available parameters for each program are indicated in reference chapters 33-36 of this manual.

Viewing and Editing Files from the Menu

The EPI6 menu has a built-in editor that offers basic editing of text files. To create a new file, choose NEW from the FILE menu. To open an existing file, choose OPEN from the same menu.

See Chapter 20 for further details on commands within the menu Editor.

Related Topics in Other Chapters

Chapter 20 contains more detailed directions for the editing functions of EPI6, and describes how to reconfigure the menu, add new entries and functions, and construct an entirely different menu for use with other software. You may wish to add items so that your favorite programs are accessible from the EPI6 menu.

Functions to set up a default data file directory and colors and other features of questionnaires have been included in the ENTER program, and are described in Chapter 8.

To: The Director

From: John Smith, M.D.

Subject: Gastroenteritis in Covington County

On July 6, 1988, Dr. Robert Downs, Director of the Covington
County Health Department, called to say that 44 cases of
gastroenteritis had occurred in attendees at a wrestling team
picnic at Coriander State Park. Approximately 54 adults and
children had attended the picnic on July 4, giving a preliminary
attack rate of 81%.

Dr. Downs requested assistance in interviewing the attendees and
performing cultures of case and food specimens. Ten patients
interviewed by his staff had eaten the following foods:

       Chicken   4
       Steak   6
       Baked beans   7
       Potato salad 10
       Green salad   6

Chapter **6**

Using EPED, the Epidemiologist's Editor, as a General Word Processor

---

Tutorial

Run EPED from the EPI menu as described in the previous chapter.

When you see a blank screen with function key commands listed at the top, press and hold the <Ctrl> key. Another menu, listing the <Ctrl-Function> keys appears at the top of the screen. If the "mode" listed in the lower right corner of the screen is QES or TXT rather than WW (word wrap), press <Ctrl-F9> one or more times until the mode is WW.

Now type a paragraph on screen, using the <Backspace>, <Delete>, and arrow keys to move around and correct errors if necessary. When you have finished, press <F9> to save the text in a file. A request for a file name appears. Enter eight or fewer characters, perhaps all or part of your first name, and an optional suffix, such as ".TXT". To leave EPED, press <F10>. You have just created a document.

To bring back the same document for editing, run EPED again, and press <F2>. Choose OPEN FILE THIS WINDOW by pressing <Enter>. Now enter the name of the file you created, including the drive letter and directory if necessary, and press <Enter>. The text will appear, and can be for further edited and saved as needed.

A more detailed, interactive tutorial is available. To use it, press the <F3> function key to run the EPIAID function. A menu will appear. Choose the "Word Processing" tutorial from this menu by moving the cursor with the arrow keys until the bar is over the right choice. Then press the <Enter> key to run the tutorial. Follow the instructions for a tour of EPED's functions.

Overview

This chapter concentrates on processing text documents such as letters, memos, and scientific articles, using EPED, the word processor provided with Epi Info. The next chapter tells how to make a questionnaire for use in entering data into Epi Info. If you are familiar with word processing and are eager to enter and analyze data, you may want to skim this chapter quickly and go on to the next, where the process of constructing questionnaires in EPED is described.

We will introduce the basic concepts of word processing, and demonstrate the functions and commands of EPED. Most of the same concepts are presented in the interactive tutorial that accompanies EPED. Instructions for running the tutorial are given above.

The features of EPED include:

- Full screen entry and editing of text
- Help screens and pop-up menus available at any time
- A complete interactive tutorial and programmable word processing system
- Automatic formatting of text during entry (Word Wrap)
- Cutting and pasting of blocks of text via block commands
- FIND and FIND-and-REPLACE commands
- Flexible Tab settings, with automatic setting of tabs from column headings within a table
- Top, Bottom, Right, and Left margin settings
- Automatic conversion of WordPerfect files to ASCII text files
- Easy drawing and shading of boxes with the cursor keys
- Bold, italic, compressed, underlined, superscript, and subscript text
- Easy use of non-English characters, such as accented vowels
- Screen colors that can be set by the user
- Access to DOS commands and other programs from within EPED
- Response to most Wordstar commands. These are not easy to remember, but once learned, greatly speed up the process of editing.
- The ability to have up to four files open at one time and to switch rapidly from file to file

A programmable text processing system called EPIAID is included in EPED. It is used not only for tutorials but also for guidance in producing reports of epidemic investigations. By interacting with EPIAID, you will generate text for the

investigation report at the same time you plan the investigation. EPIAID programs are provided to create a preliminary report in the CDC Epi I format, to make a questionnaire, and to develop a design for an epidemiologic study. The use of EPIAID for producing questionnaires is described in the next chapter and for other purposes in Chapters 20 and 21.

Running EPED

Run EPED from the EPI menu as described in the previous chapter. From the PROGRAM menu, choose EPED, and you should see the EPED banner and then the word-processing screen.

The EPED Screen

Across the top of the screen is a command prompt line listing the 10 function keys that perform major functions in EPED. Each one either performs a command directly or causes a menu to pop up in the middle of the screen. Pressing <F2>, for example, shows the menu of commands affecting files. Holding down the control (<Ctrl>) key displays a second menu of commands that are activated by holding down the <Ctrl> key while pressing a function key. Many of these commonly used functions also have entries in the pulldown menu, but appear on the <Ctrl-Function> menu for easier access.

At the bottom of the screen is the STATUS LINE. From left to right, it shows:

- The number of the currently active window if more than one file is open
- The name and directory of the current file, if any
- The amount of memory (RAM) available for holding the document
- (EPED keeps the document temporarily in Random Access Memory for greater speed in processing, but this will generally accommodate about 70,000 words in a computer having 640 K of RAM).
- The function you are carrying out: EPED for general word processing; EPIAID for programmed assistance with word processing.
- The current page number in the document
- The line and column numbers occupied by the cursor
- A series of letters that indicate the status of important settings: WW, TXT, or QES for the current mode--Word Wrap for word processing, TXT for line-oriented documents such as computer programs, and QES to make an Epi Info

questionnaire. After this may appear: INS for Insert mode ON; OVR for Overwrite mode ON; IND for AutoIndent mode ON; JST for Right/Left Justification ON.

Creating a Document

Creating a document in EPED is easy. Simply type the document on the screen as you would like it to appear. At any point, you may press the "save" or <F9> key. A window will appear, asking for a file name, and after this is given, the file will be saved on the current or the designated disk. At the end of the document, press <F10> to leave EPED. The program will remind you if any portion of the file has not been saved, and then exit to the EPI menu or to DOS. The completed document will be on the disk, waiting to be called up for viewing, editing, or printing.

Setting Word Wrap, Text, and Questionnaire Modes

EPED offers a choice of three *modes*--Word Wrap(WW), Text(TXT), and Questionnaire(QES). The abbreviation for the current mode is shown in the lower right corner of the screen. To change modes, press the <F6> key to show the SET menu, and use the space bar to change the current setting. <Ctrl-F9> or <Ctrl O-W> will toggle between TXT and WW modes rapidly.

Each mode comes with a built-in series of settings on the <F6> menu. WW mode is initially set with a right margin of 65, for example, and TXT and QES with margins of 80 (full screen). For any of the modes, you can change the settings while in that mode. The settings are saved in a file called EPED.CFG when you choose SAVE SETTINGS THIS MODE in the <F6> menu. You must save the settings for each mode separately if you want them to take effect every time you run EPED.

When you exit from EPED, the current settings and file name are saved. If RELOAD is set to ON in the SET menu, the same file (if any) and mode will be reloaded the next time EPED is run.

Since EPED, Version 6, saves its settings to a separate file called EPED.CFG in the currently logged directory, users on a local area network (LAN) can each have different settings while running EPED from the network. Each user should run EPED from a directory on his/her own computer, using a PATH statement in the AUTOEXEC.BAT file to find the EPED program on the LAN. Alternatively, you can copy EPED.* from the LAN onto the local computer and run it from there, the only drawback being the use of a small amount of hard disk space.

Automatic Reformatting of Text

When Word Wrap is ON, paragraphs will be reformatted or justified automatically after insertion or deletion of text, as soon as a space is inserted or a deletion is made. This feature can be turned off by setting AUTO WORD WRAP to OFF in the F6-SET and WW, JUSTIFY menus.

Automatic formatting does not apply to existing documents, except in paragraphs in which you make a change and press the space bar. The FORMAT PARAGRAPH command, activated from the TEXT menu, or by pressing <Ctrl F8> or <Ctrl B> with the cursor in the paragraph, will format an existing paragraph.

Page Markers

When PAGINATION (in the <F6> SET menu) is ON, markers appear on the first line of every page on both sides of the screen as solid rectangles.

Editing a Document

To edit a document, use the "Open file this window" command in the <F2> (File) menu. When the file name window appears, give the name of the file name you would like to edit. If you have spelled it correctly, and the disk letter and directory are correct, the file will appear on the screen ready for editing.

If you are not sure of the spelling or location of the file, use a "wildcard" file name, like "*.LET" for all file names ending in ".LET" or "jones.*" for all files beginning with JONES, or "*.*" for all files. If you wish to see files on another disk, the name might be "A:*.*" or "D:\ DATA\ *.LET", for example. The names of files matching the description will be displayed. Move the cursor bar to the one desired and press <Enter> to bring the file into EPED for editing.

Now use the cursor keys to move to a point where you would like to insert or delete text. If the letters INS are not at the lower right corner of the screen, press the <Ins> key to turn INSERT on. Inserting is easy; just type the material for insertion and it will be inserted at the cursor location.

Deleting a few letters is done with the backspace or delete keys. An entire line can be deleted by pressing <Ctrl-Y> when the cursor is on that line.

You can turn off the insert feature so that characters typed "write over" the ones on the screen rather than pushing them to the right. Pressing the <Ins> or Insert key will turn off the Insert mode and turn on Overwrite mode; pressing it again will reverse the process.

When you have finished editing, press the <F9> (Save) key and then <F10>(Quit) to return to the menu. You may prefer to use <F10> alone and be prompted to save any new material.

The Cursor and Some Important Keys (for Those Not Familiar with DOS Keyboard Commands)

The cursor is the small bar or box that moves when you type on the EPED screen. It indicates the location where the next character typed will be inserted, and also serves as a marker for the location where many commands will take effect. For example, the command to mark the beginning of a block of text places the mark at the current cursor location. As you move the cursor, the numbers indicating line and column at the bottom of the screen change to reflect the new position.

Several keys on the keyboard in addition to the letters and numbers are important in EPED.

Holding down the <Shift> key and pressing any letter produces the capital or upper-case letter. Pressing <CapsLock> causes all letters to be in upper case until <CapsLock> is pressed again.

The <Del> or Delete key removes the character at the cursor position; <Backspace> removes the one immediately to the left.

The <Enter> key is the equivalent of the Carriage Return key on a typewriter. In all the Epi Info programs, it is used to finalize and execute a command, as well as to terminate a line of text in EPED. When Word Wrap is on, each paragraph, but not each line of text, ends in <Enter>. In the TXT and QES modes, each line ends with <Enter>.

The <Ctrl> or Control key, like the shift key, is used in combination with other keys. The command <Ctrl-J> means hold down the control key and press "J" once. Many com mands require two keys after pressing the <Ctrl> key. To give the <Ctrl-K B> command (to mark the beginning of a block), for example, you would

56

hold down the <Ctrl> key while pressing <K> and then <B>. You can also release the <Ctrl> key before pressing <B>; the result will be the same in EPED>.

The <Alt> key is another way of giving different meanings to the letter keys. <Alt-1> to begin typing bold text is produced by holding down the <Alt> key and then pressing the digit <1>. The function keys used by Epi Info are those labelled <F1> through <F10>. They have special meanings for important commands or menus as indicated at the top of the EPED screen. Other important keys used for moving the cursor around are described in the following section.

Entering Text and Moving the Cursor Around on the Screen

To see the effect of the keys that move the cursor, type a few paragraphs of text on the screen. This is done exactly as on a typewriter, except that with Word Wrap ON (WW displayed at the bottom of the screen), you should **avoid pressing the <Enter> key** except at the end of a **paragraph**. Do **not** press <Enter> after every line, unless you want the line to be considered a separate paragraph.

The cursor may be moved to any part of the text with the following commands:

- The **arrow** keys, sometimes called the cursor keys, move the cursor right, left, up, or down.
- The <PgUp> and <PgDn> move up and down one entire screen (22 or 23 lines).
- The <Home> key moves to the beginning of the current line.
- The <End> key moves to the end of the current line.

Note that the cursor will not move past the end of the document unless you insert more blank lines with the <Enter> key.

Pop-up Menus, the <Esc> Key, and Giving Commands

Pressing a function key indicated on the command prompt line at the top of the screen will often cause a menu to pop up in the middle of the screen. To choose one of the commands on the menu, move the cursor bar to the choice with the down or up arrows and press <Enter>. Alternatively you can enter the highlighted letter in the command to execute the command directly.

To close a menu on the screen, press the <Esc> (Escape) key. This key will remove any of the menus or help windows and is a good one to press experimentally if you have trouble deciding how to move around the menu system.

Note as you move the cursor bar up and down that a series of command keys is shown in the top left corner of the screen. These keystrokes are alternative commands for the current menu choice. Pressing these keys will call the command when you are in the middle of text processing without using the pop-up menu. To center a line, for example, you can choose Center Current Line from the <F4> TXT menu with the cursor bar, press C to choose this command from the menu, or press <Ctrl-O C> without popping up the menu. At first, you will probably use the menus, but later you may find the <Ctrl> key sequences faster for some often-repeated operations.

Handling Files

The documents in EPED are stored as files on the computer's floppy or hard disk. The File menu that pops up when you press <F2> provides commands for opening new or existing files and for saving material on the screen in files. It also gives access to file directories, information on the current file, and a command for moving from window to window when more than one file is open.

The format of file names is important in EPED, as in any program that runs on IBM-compatible microcomputers. A file name contains a **name** of eight or fewer letters or numbers, followed by an optional **period** and **extension** name of three or fewer letters or numbers.

Examples of correct file names are:

      DOC1
      DOC1.TXT
      JONES.LET
      jones.let
      TUCSON.QES   (An Epi Info questionnaire file)
      TUCSON.REC   (An Epi Info data file)
      TUCSON.CHK   (An Epi Info CHECK file)
      TUCSON.PGM   (An Epi Info ANALYSIS program)

Punctuation (other than the single period), spaces, and other unusual characters are not allowed. Note that upper and lower case are the same, as DOS converts all file names to upper case.

A file name may be preceded by a drive letter and a colon and may also have a "directory" indicated. Thus the more complete file names:

A:Jones.let
C:\Epi6\Gonzalez.let

These more complete names containing drive letters and directories are called "path" names. The file called JONES.LET is on drive A. The file called GONZALEZ.LET is on drive C in the directory called EPI6.

When you open a file or save one that has not been named, you will be asked to type the file name in a window that pops up on the screen. If you try to leave EPED without saving the current file, you will be reminded to save it if desired before the program exits to the EPI menu or to DOS.

Since the document under construction is held in Random Access Memory (RAM), recent changes can be lost if there is a power failure or if someone turns off your computer. To minimize the frustration from such events, you should save the current document approximately every 20 minutes as you work, so that only bearable amounts of text will be lost in the event of such disasters.

If you work on a hard disk, it is important to copy ("back up") newly created or edited files to a floppy disk periodically, since hard disks, given enough time, can be counted upon to "crash" and make files unavailable. Backup files on a floppy disk can be made using the "Save File to .." command in the <F2> File menu, or can be copied with the copy command in DOS.

Text that has been "saved" is safely on the disk, but subsequent changes are only in RAM and will be removed by an electrical power failure. During a thunderstorm or when someone is working on the electrical system, it is usually best to save your file, turn off the computer, and wait until later, but if you continue working, frequent saves (<F9>) are in order!

Configuration Settings

EPED offers a choice of three *modes*--Word Wrap(WW), Text(TXT), and Questionnaire (QES). The abbreviation for the current mode is shown in the lower right corner of the screen. To change modes, press the <F6> key to show the SET menu, and use the space bar to change the current setting. <Ctrl-F9> or <Ctrl O-W> will toggle between TXT and WW modes rapidly.

Each mode comes with a built-in series of settings on the <F6> menu. WW mode is initially set with a right margin of 65, for example, and TXT and QES with margins of 80 (full screen). For any of the modes, you can change the settings while in that mode. The settings are saved in a file called EPED.CFG when you

choose SAVE SETTINGS THIS MODE in the <F6> menu. You must save the settings for each mode separately if you want them to take effect every time you run EPED.

When you exit from EPED, the current settings and file name are saved. If RELOAD is set to ON in the SET menu, the same file (if any) and mode will be reloaded the next time EPED is run.

Since EPED, Version 6, saves its settings to a separate file called EPED.CFG in the currently logged directory, users on a local area network (LAN) can each have different settings while running EPED from the network. Each user should run EPED from a directory on his/her own computer, using a PATH statement in the AUTOEXEC.BAT file to find the EPED program on the LAN. Alternatively, you can copy EPED.* from the LAN onto the local computer and run it from there, the only drawback being the use of a small amount of hard disk space.

Automatic Reformatting of Text

When Word Wrap is ON, paragraphs will be reformatted or justified automatically after insertion or deletion of text, as soon as a space is inserted or a deletion is made. This feature can be turned off by setting AUTO WORD WRAP to OFF in the F6-SET and WW, JUSTIFY menus.

The FORMAT PARAGRAPH command, activated from the TEXT menu, or by pressing <Ctrl-B> will accomplish the same function at any time.

Finding and Replacing Text

Word processors are good at finding a particular word or phrase and replacing it with another combination of characters. Suppose, for example, that you have a document in which the word "color" appears many times. You wish to change the spelling to "colour" for a European colleague.

Press the <Ctrl> key and note the two functions FIND and REPL on the menu at the top of the screen. To find the word "color" without changing it, you would use <Ctrl-F6> FIND, but in this case <Ctrl-F7> REPL is needed to replace the word with another. A dialog box appears asking for the characters to be found. You type "color," and respond "colour" to REPLACE WITH. A series of options appears:

C=Case Specific	Match upper/lower case of your entry and ignore those that fail to match. Normal default is to match regardless of case. Note: This is a change from Version 5 of EPED.
U=Upward	Look upwards in the file from the current cursor position
G=Global	Find all instances of "color," not just the first one
W=Word	Find "color" only if surrounded by spaces. Do not find "colorize," for example.
B=Block	Find "color" only within a block of text that has been selected with the block commands (see next section). This is particularly handy for confining choices to a vertical column.
N=No ask	Do not ask for confirmation before changing "color" to "colour"
1..n=How many	Change only the specified number of instances before stopping

You may combine options. For example, the options GN will change all instances of "color," regardless of case, to "colour," without asking for confirmation for each one. U10N would change the first ten instances that match exactly the upper/lower-case combinations that you entered. Entering no options will change only the first instance of "color," "Color," etc., regardless of case, pausing for confirmation before doing so.

Block Commands

So far we have described elementary methods for dealing with files, characters, and lines. EPED also offers commands for marking, copying, moving, and deleting **blocks** of text. A block is a body of text that has been selected for further manipulation with the block commands. The common commands for block operations--BEGIN, END, COPY, MOVE, DELETE--are shown on the menu that appears when the <Ctrl> key is held down. These commands are also in the Blk or Block menu, reached by pressing <F8>.

To use the block commands, you must first mark the beginning and end of the block. To mark the beginning of a block, place the cursor at the place desired and then choose BEGIN by pressing <Ctrl-F1>. Alternatively, you may choose BEGIN BLOCK from the <F8> BLK menu, or press <Ctrl-K B>. No visible results occur, but you have marked the block nevertheless.

To mark the end of the block, move the cursor to the end, and choose END by pressing <Ctrl-F2> (or <Ctrl-K K>). The block will now be highlighted. Any subsequent block commands will act on this block as long as it remains selected.

To copy the block to another location, leaving the original marked text intact, first

place the cursor at the point where the copied text should be inserted, then select COPY with <Ctrl-F3>. A copy of the selected block will be inserted at the new location, leaving the original block where it was.

To move the block to another location ("cut and paste"), put the cursor at the point where the block should be inserted and choose the MOVE command, <Ctrl-F4>. The block will appear at the cursor location and disappear from the old location.

To delete a block use DELETE, <Ctrl-F5>. The text will disappear.

Note that when you place the cursor bar on a menu choice, the alternative command, e.g., "<Ctrl-K><Ctrl-B>", appears on the first line of the screen. This is the "Wordstar" command that may be used without having to pop up the menu. There are Wordstar commands for each of the block operations.

To "unmark" a block and remove the highlighting, you may choose one of several methods. Pressing <Ctrl-K B> after the end of the block, so that the beginning comes after the end, effectively removes the block designation altogether, as does marking another block. The command "Hide Block Display" or (<Ctrl-K H>) leaves the block still defined but removes the highlighting from the screen, making an "invisible" block.

Block operations may seem a little complicated at first, but the benefits over the old typewriter system and scissors and paste are enormous. The scissors and paste method also requires outlining the beginning and end of the block, opening a hole for the new block, and closing up the hole where the old one left, but with visible seams and sticky fingers as side effects.

Column Block Operations

In editing tables, data files or screens, it is often important to perform operations on or in vertical rectangles corresponding to columns on the screen. EPED offers a method of marking COLUMN BLOCKS so that you can designate the upper left and lower right corners of a rectangular area to be highlighted.

To select a COLUMN BLOCK, place the cursor at the upper left corner of the area to be selected, and press <Ctrl-F1> (or <Ctrl K B>) *twice,* without moving the cursor (once will select the block in the usual, line-oriented way). Then move the cursor to the lower right corner of the intended rectangle and press <Ctrl-KK> to mark the end of the block, just as with a line-oriented block. The columns selected will now be highlighted, and further block commands will operate on the block automatically.

The following functions apply to BLOCKS and also to COLUMN BLOCKS:

Function	Command
Copy Block to Here	<Ctrl-F3> or <Ctrl-K C>
Move Block to Here	<Ctrl-F4> or <Ctrl-K V>
Delete Block	<Ctrl-F5> or <Ctrl-K Y>
Find	<Ctrl-F6> or <Ctrl-Q F> (With the bLock option)
Find/Replace	<Ctrl-F7> or <Ctrl-Q A> (With the bLock option)

The Copy and Move commands move or copy the marked block so that its upper left corner is located at the current position of the cursor.

To practice using the column block functions, first mark the upper left corner of a block of text with <Ctrl-F1><Ctrl-F1> and the lower right with <Ctrl-F2>. Now move the cursor elsewhere on the screen and press <Ctrl-F3> to copy or <Ctrl-F4> to move the block.

Try the FIND/REPLACE functions, replacing every "e" with "E", for example, by using the "gnl" (global/no ask/BLOCK) options. Note that the replacement only occurs within the marked block. When you have finished, press <Ctrl-F5> to delete the block.

Note that by learning only one new command, the twice-repeated <Ctrl-F1>, you have the power of all the column block commands at your disposal.

Printing

The final product of most word processing is a printed document. To print a document from EPED, you must first save the file. Then bring down the **Print** menu by pressing <F5>. If the file currently on the screen is the one you wish to print, just select "Print file now" and press <Enter>. Otherwise, move to "Name of File" and type the name of the desired file and press <Enter>. Then return to "Print file now" and press <Enter>.   If the document is long and you would like to print only a portion, designate the first and last pages to be printed in the appropriate menu locations before printing.

If this is the first time you have used EPED with a particular printer, you may need to set up margins in the <F6> (Set) menu. Laser printers often require a page length of 60 lines rather than 66, for example. To customize output for your printer, you may choose another printer file from "Which printer" in the Print menu and insert special instructions to the printer for different type styles, ends of pages, etc., by

editing the printer codes. For instructions on how to do this, see Chapter 33, "EPED Commands," in combination with the manual for your printer. Advice from an expert and some trial and error are usually required before a given printer works well with any word processor.

Printer Definition Files (.PDF), providing information about particular printers, are provided for Epson, IBM Proprinter, Okidata, Hewlett Packard Deskjet and Laserjet, and Postscript printers. To choose the correct file, press <F5> to bring down the print menu, and then press <Enter> under "Which Printer" to choose a .PDF file. Postscript printing requires that the file, POSTSCPT.INI, supplied with Epi Info, be present in a directory in the current path.

To create a new .PDF file or edit an existing one, use the Edit Printer Codes and Save Printer Setup commands on the <F5> screen. Special sequences of characters must be sent to the printer to indicate the beginning and end of the various text styles. Since these vary for each brand of printer, you may have to consult the printer manual and set them up for your printer using these commands.

To do this, press <F5> to get the PRINT menu, and then choose EDIT PRINTER CODES. From the list of "Printer Strings" that appears, choose one, such as "Initialization." To insert an Escape character into the file so that it will be sent automatically during initialization of the printer, press the <Scroll Lock> key once so that the message at the bottom of the window indicates the "Macro" mode. Now press the <Esc> key. "<Esc>" will appear in the window. Other keys are treated similarly, those that are not printable having a "macro" representation in Printer Strings. When you have finished installing the appropriate commands for your printer, use the SAVE PRINTER SETUP command on the PRINT menu to save the file.

Reading of WordPerfect Files

Version 6 of EPED reads WordPerfect, Version 5.xx, files directly. They are converted automatically to as close an approximation of the original as possible within EPED, although embedded graphics and WordPerfect tables will not be converted. When you save a file that was originally in WordPerfect format, it will be saved as an EPED file in the ASCII or plain DOS format. You should therefore give it another name before saving. This version of EPED does not read WordPerfect, Version 6 files.

Importing and Exporting Files from Other Word Processors

Epi Info produces ASCII files, that is, the characters are all of the standard character code of 128 characters. Many other word processors will also produce ASCII files through conversion of their own non-ASCII files to the ASCII form. One way to import other files is to save the other file in an ASCII or plain text format and then open the file in the usual way, just as though it were an EPED file.

EPED has another feature that helps in importing files containing non-ASCII characters. In the <F6> (Set) menu is a submenu called LOAD FILE OPTIONS. If you set STRIP HIGH BIT in this menu to ON, the non-ASCII characters will be converted to ASCII when a file is opened and loaded into memory. If you then save the file in EPED, it will become an ASCII file.

The DOS Character Set

The DOS character set for microcomputers contains a number of characters that are not part of the standard ASCII standard, but are quite useful for drawing boxes and typing mathematical expressions and non-English languages. These characters may be produced in EPED after pressing the Scroll-Lock key.

Alternate character sets are supplied with DOS in some countries. Not all of the following instructions will apply to these non-English character sets, but most keys in the Scroll-lock mode will function as described.

Not all printers will print the full DOS character set. If your printer does not, you can still use the special characters for producing menu screens and other files, but printing requires a fully DOS-compatible printer.

Boxes

Boxes are useful for emphasizing text (as in the tutorials at the beginning of chapters in this manual). They can also be used to make professional-looking menus on the screen, or for organizational diagrams. Single lines are also useful for emphasis and decoration in documents.

EPED, Version 6, allows drawing single or double horizontal and vertical lines or boxes on the screen with the Arrow (Cursor) Keys. Boxes and shading are produced with the arrow keys after adjusting the type of character with the <Home> and <End> keys. Instructions will appear at the top of the screen after you press the<Scroll Lock> key.

To draw a box on the screen using single lines, press the <Scroll Lock> key, and then simply draw the box with the up, down, right, and left arrow keys. The corners will be taken care of automatically. If your last character is on a corner, be sure to "turn" the corner with another arrow key so the program knows that you want to complete the box.

To draw boxes with double lines, press the <Home> key. The information line at the top of the screen will change to "DOUBLE." Any lines drawn in this mode will be double. Single and double lines may be mixed in the same box, and the program will take care of putting the correct joining characters at the intersections. You can easily make the following boxes:
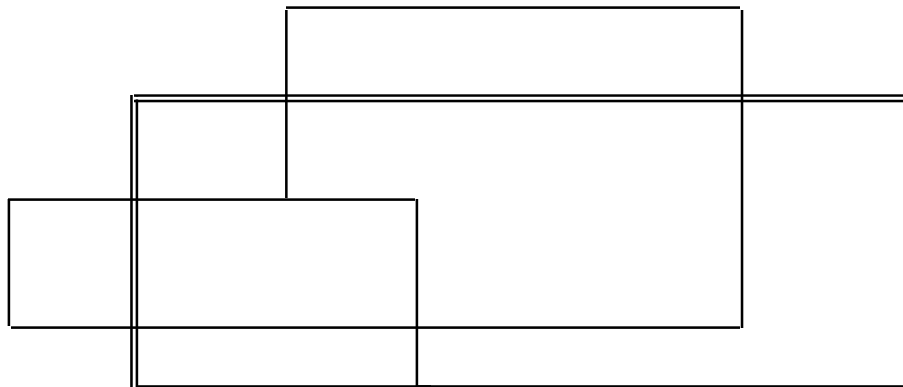
To erase a box, press the <End> key and use the arrow keys to trace the part you want to erase. To start drawing boxes again, press the <Home> key.

There are a number of shading characters in the IBM character set. To choose a character, press the <End> key repeatedly, cycling through the available characters. After the character is chosen, use the arrow keys to do the shading, just as in drawing boxes.

Non-English-Language Characters

To produce a non-English character, hold down the <Alt> key and then use the following keys to produce vowels and other characters with special marks:

Use <Alt> with: a   e   i   o   u   c   n   y
to produce: áàâäåæ éèêë   îíï   óòôöø   úùûü   ç   ñ   ÿ

or with
<Alt-Shift>

to produce: Å Ä Æ É   Ç   Ñ

The first time the key is pressed, the first of the special characters appears. To
produce the others, continue to hold down the <Alt> key and press the same letter
key repeatedly until the correct character is obtained. Holding the <Alt> key down
and pressing "a" three times will produce "â ". Pressing the "e" key will first
produce "ê ," with repeated presses of <Alt e> cycling through the various types of
"e's." To produce the capital letters, hold the <Alt> key and the <Shift> key. <Alt
Shift A> produces Ä , Å , and Æ.

Greek and Mathematical Symbols

To type selected Greek letters and mathematical symbols, turn <Scroll Lock> on.
The following Greek and mathematical symbols are available by typing on the main
keyboard:

Scroll lock on--No Shift Key

```
Type:          `  1 2 3 4 5 6 7 8 9 0 - =
To obtain:         1/2   1/4

Type:          q w e r t y u i o p [ ] \
To obtain:           Ã           _ ½ ¹     _

Type:          a s d f g h j k l ; '
To obtain:     a o _   ¨γ

Type:          z x c v b n m , . /
To obtain:                     µ       /
```

Scroll lock on--Shift Key Pressed

```
Type:          ~ ! @ # $ % ^ & * ( ) _ +
To obtain:     Å   2       _   • •       ±
```

```
Type:          Q W E R T Y U I O P {  }
To obtain:

Type:          A S D F G H J K L : "
To obtain:     _       ¬           ÷

Type:          Z X C V B N M < > ?
To obtain:             ß       ²  ³
```

Font Styles

Font styles are selected by holding down the <Alt> key and pressing a number (1-7).

<Alt-1> **Bold**
<Alt-2> `Double`
<Alt-3> Underscore
<Alt-4> Superscript
<Alt-5> Subscript
<Alt-6> Compressed
<Alt-7> *Italic*

To use a particular style, press the corresponding Alt key combination. The color of the text on the screen will change to indicate the new style. Type the text to be printed in the new style. Press the <Right arrow> key to return to the normal text style. (This jumps over the control character that is inserted to the right of the special-style text.)

To print text in the styles selected, a printer configuration file (.PDF) file must be loaded to handle the conversion of style indicators into directions for particular printers. Further details on .PDF files are given in the previous section on "Printing."

Macros

A macro is a "recording" of a number of keystrokes that can be played back under your direction to repeat a complex operation . Suppose, for example, that you want to insert two blank lines after every line in a document. You could make a macro in which you press the <End> key to go the end of the current line, then press <Enter> three times.

The macro can be played back once by choosing the <F4> TEXT menu and then MACRO USE and PLAYBACK. The real power of macros comes, however, from using them in combination with the FIND/REPLACE function. To do this, choose the <F7> FIND menu, then USE MACRO WITH FIND/REPL. Depending on what you enter for FIND, you can have the macro operate each time a particular character, word, or phrase is encountered in the file, or, by FINDing "^L", at the end of each line in the file. The latter would be especially useful with the macro mentioned above, to insert one or more lines after each line in the file. It could also be used to indent or remove indentations from a file.

To record a macro, choose the <F4> TXT menu and then MACRO USE. A menu appears, offering several choices, as follows:

LOAD MACROS--allows you to load a file containing up to 10 macros already recorded. A file called EPED.MAC is loaded automatically, if one is present, but this option allows you to load other files of macros that you may create. This file is supplied with Epi Info, but may be altered or replaced with macros of your own choosing.

STORE MACROS--saves the 10 or fewer current macros into a .MAC file. You will be asked for the name of the file in which to store the macros.

PLAY BACK MACRO--Plays back a macro from the current cursor position. You will be asked to choose the macro from a list of 10 in the current macro file.

RECORD--ON OR OFF--The first time this choice is made, EPED starts recording macro keystrokes. Up to 255 keystrokes can be recorded. When you have completed the operation(s), make this choice again to turn off recording, and you will be asked to choose one of the ten "slots" in the current macro file and give a name to the macro. To make the recording permanent, you should also choose STORE MACROS before leaving EPED to save the recording in a file. The keystrokes <Ctrl-J T> are a shortcut method of choosing RECORD-ON or OFF. Whichever the method of recording, only the material between the two choices will be recorded--not the menu choices that lead to the recording.

EDIT--Allows editing or creation of a macro. You are asked to choose a macro. An editing window appears. If there are already keystrokes in the macro, these are represented in symbolic form--"<Enter>" for the <Enter> key, for example. The window has two modes, controlled by the <Scroll lock> key. In Literal Mode, the keystrokes are entered directly in the window as in an editor. In Macro Command Mode, pressing <Enter> makes the symbol "<Enter>" appear on the screen, and the <Backspace> and <Delete> keys are represented by "<Ctrl-H>" and "<Del>." In literal mode, <Backspace> and <Delete> may be used to remove symbols so that the correct commands can be included in the macro in the desired sequence. The

Wordstar commands listed in Chapter 33, the EPED Reference Chapter, are particularly useful for macros, as they are easier to visualize that the keystrokes necessary to go through the menuing system.

A sample macro file called EPED.MAC comes with EPED. It is loaded automatically when EPED is run. To see how macros work, press <F4> and then choose MACRO USE from the TXT menu, and then PLAYBACK. Choose a macro from the 10 choices presented and you should see the results on the screen.

Two shortcut methods of playing back macros are offered. The first allows you to play back one of macros 1 through 9 in the current macro file, by pressing <Ctrl O> (that's "oh," not "zero") and then the number of the macro (since <Ctrl 1> is not recognized by the keyboard, you must release the <Ctrl> key before pressing the number.)

The second series of shortcuts allows you to play back macro number zero from 1 to 9 times by pressing <Ctrl J> and then a digit 1 through 9 (after releasing the <Ctrl> key).

Macros are useful not only for text items such as addresses, but they can be used to automate repetitive operations. A little experimentation will suggest new uses.

Related Topics in Other Chapters

Many more features of EPED are explained in Chapters 33 and 34 on EPED and EPIAID commands. When time allows, you should try each of the menu commands while consulting these chapters to see the full extent of the facilities available. The setting of margins, tabs, and screen colors, the use of multiple windows, conversion of files to ASCII (in "Load File Options"), and other features can all be helpful in particular situations.

The making of questionnaires is described in the next chapter, and the use of the EPIAID system, a complete facility for interactive text generation, is described in chapters 20, 21, 28, and 34. The tutorial that accompanies EPED is an example of one kind of EPIAID program.

```
City or town      _____ State <A > ZIP <A    >


ID <IDNUM>      Age ###        Sex <A> (M,F,U)

Disease <A              >   {Dis}ease {Code} ####


Date <mm/dd/yy>

This is the date of:
   1. Onset
   2. Diagnosis
   3. Laboratory Report
   4. Report to Health Dept.
   Date Type #
```

72

Chapter 7

Creating Questionnaires Using EPED

Tutorial
Run the main EPI6 menu. From the TUTORIAL menu, choose MAKING A QUESTIONNAIRE and follow the instructions on the screen.

For a more substantial foodborne-outbreak questionnaire, also produced in an interactive tutorial, choose EPED from the PROGRAMS menu. In EPED, press <F3> to see the EPIAID menu, and choose "Make Epi Info Questionnaire" by moving the cursor bar with the down arrow key and pressing <Enter> to make the choice. Follow the instructions in the tutorial carefully and make a questionnaire. Note that many items, such as specific foods consumed, are only outlined in the sample questionnaire, and that you must supply appropriate questions for your own epidemic before the questionnaire can be used.

Whichever method you have used to make a questionnaire, you should save it with a file name ending in .QES. You might use up to eight letters of your name, followed by a period and .QES, as in MARY.QES

Overview

Before beginning to enter data, you must provide Epi Info with a questionnaire that describes the data to be entered. The ENTER program (Chapter 8) then uses the questionnaire to construct a data file, called a .REC file, automatically.

ENTER reads a questionnaire that has been produced on a word processor in ordinary text format and uses it as a template to define a data file. It constructs variable names from the text in your questionnaire and derives data types and lengths of variables from the blanks you have inserted in the questionnaire. After creating the data file, ENTER presents the questionnaire on the computer screen and allows entry of data within a few seconds.

The word processor, EPED, is provided for convenience in constructing questionnaires. EPED can also be used for correspondence, reports, and other purposes, as described in the preceding chapter. This chapter tells how to make a questionnaire in EPED. If you have another favorite word processor that can produce a simple text (ASCII) file or a non-document Wordstar file, the instructions that follow will tell how to produce the questionnaire, but you must rely on other sources for information about the word processor itself.

Setting Up EPED To Make a Questionnaire

Run EPED from the EPI6 menu. Press <F6> to see the SETUP menu. The first choice is WW/TXT/QES mode. Press the space bar one or more times until QES mode is selected, and then press <Esc>. This establishes page size and other settings with convenient values to make a questionnaire.

Indicating Data Entry Locations in the Questionnaire

When a questionnaire is being developed for use in ENTER, a few simple conventions are necessary to tell the program where to create data entry **fields** or **variables** and what kind of data to accept at these locations. ENTER will give names to the fields. In the ANALYSIS program, the same names are called variable names and are used to refer to particular data items. The field formats are described below and on the back cover of this manual.

There is a special command in EPED to make it easy to insert fields in your questionnaire. To see it in action, type <Ctrl-Q Q> (hold down the control key, type **Q** once, and then let go of the control key and type a single **Q**). It can also be accessed from the Text menu, but <Ctrl-Q Q> is easy to remember. A menu of field types like the ones on the back of the manual will appear on the screen. To insert one in your questionnaire, move the bar cursor to the choice and press <Enter>. For some field types, EPED may ask about the length of the field or the number of digits. After you have provided this information, the field will be inserted in the questionnaire at the current position of the cursor.

If you are using a word processor other than EPED or find it easier, simply type the proper characters for the field into your questionnaire without using the <Ctrl-Q Q> command. The field types are shown below, with information about their meaning.

Field Types in Epi Info

_
_____
_____

Text or "underline" fields, indicated by continuous underline characters. Any
printable character can be entered in this field. Blank fields (spaces) are interpreted
as missing data in ANALYSIS. The length of the variable or field will be the
number of underline characters used. The maximum length of a text field is 80
characters.

#
##.#
etc.

Numeric fields. Only numbers or spaces will be accepted. If nothing is entered,
the result will be a blank, which is interpreted as a missing value and displayed as a
period (.) in ANALYSIS. The number of digits is indicated by the number of "#"'s.
If a decimal point is given, the field will be in "fixed decimal" format, allowing
exactly the indicated number of digits to the right of the decimal point. For
monetary amounts from 000.00 to 999.99, for example, ###.## would be the
appropriate field. A number field can be up to 14 characters long, counting the
decimal point as 1 character. Numbers containing decimals and any number field
containing more than four digits are "real" (floating point) numbers. Fields of less
than 5 digits, without decimals, are integers within Epi Info.

<A>
<A   >

Upper-case fields. These are similar to general-purpose or "underline" fields, but
entries will be converted to upper case. The length is indicated by the number of
characters between the "less-than" and "greater-than" symbols, including the upper
case "A".

<Y>

A **"Yes/No" field.** Only **Y**, **N**, and **SPACE** or **<Enter>** are accepted. Spaces or the <Enter> key are interpreted as missing data in ANALYSIS. Entries are converted to upper case after they are entered. Yes/No fields are always one character in length.

<mm/dd/yy>
<mm/dd>
<dd/mm/yy>
<dd/mm>

U.S. and European Date fields. Dates will be checked when entered to be sure they are valid. When entering data, you type only the numbers in the date; the program provides the separating slash marks. In creating a questionnaire, however, be sure to type the field format exactly as they appear above.

<phonenum>

A **local** phone number, entered as **xxx-xxxx** where the **x**'s are digits.

<long distance>

A **long-distance** phone number, entered as **(xxx)xxx-xxxx**, where the **x**'s are digits.

<today>
<today/yy>
<today/yyyy>

Today's-date or date-of-last-change field, which will automatically enter the date when the computer record was last saved. If the record is later edited and saved again, the contents of the field will be the latest date on which the record was saved. The three forms allow the field to contain month and day only, or month, day, and two-digit year, or month, day, and four-digit year.

The date which automatically appears in this field is the "system date" from the computer. Be sure to enter today's date each time when turning the computer on, unless your computer automatically restores the correct date on powering up. Within Epi Info, the date in this field is supplied automatically, and you will not be

able to enter anything but a carriage return (<Enter>).

<IDNUM>
<IDNUM   >

This is a special-purpose field that maintains sequential identification numbers. The first number in the file will be number 1, and each succeeding record will have an IDNUM one higher than the previous record. This field is used to maintain unique identification numbers automatically. The number of digits is the number of characters between the brackets, including the letters "IDNUM," and must be 5 or more.

Since IDNUM fields are filled automatically, the cursor skips them during data entry. If you wish to set the value of an <IDNUM> field to a number other than 1 in the first record of a file, use the up arrow key to enter the field and type the desired number before saving the record. The first IDNUM in a file can also be set by using commands in a .CHK file.

_____

You should not use the underline, number (#) sign, or less-than or greater-than characters in the questionnaire except in the formats shown above, as the program will think you are trying to create an incorrect field. Specifically, **< >** and unpaired ">" or "<" signs are not allowed. The fields with special words such as **<today/yy>** must be exactly the length shown to contain the date or telephone number information correctly.

Helping Epi Info To Create Useful Variable Names

Unlike most data entry programs, Epi Info creates variable or field names automatically from the original questionnaire. When it encounters an underline or other special character, it looks for text on the same line which can be the basis for a *variable name*. The first 10 non-punctuation characters become the name. This name is used in the ANALYSIS program to refer to the data entered in this field. If the line is

Name _____

the field will be called NAME. If it is

What is your name? _____

the field will be called ISYOURNAME, since selected words like "what" are discarded automatically, along with spaces and punctuation. It is wise to put significant words within the first 10 characters of the text preceding the blank, or to mark them with curly brackets { } as described below.

If there is more than one blank on a line, Epi Info begins the search for a useful field name after the preceding blank. Thus

Name _____ **Age** ___   **Sex** _

produces three fields entitled NAME, AGE, and SEX.
Now comes the tricky part; suppose that there is no preceding text on the same line, as in

Address   _____
                          _____

Epi Info will call these two fields ADDRESSand ADDRESS2. The program also detects duplications, such as the occurrence of another

Address _____

many lines removed from the first two. Rather than making a duplicate field name, the program would call this one ADDRESS3. ENTER can handle up to 99 such duplications of a single field name.

If you wish to number your fields, this can be done as

1. Name _____

To prevent trouble in programs requiring that a field name start with a letter, Epi Info places an "N" in front of numbers if they are the first item in a variable name. You may prefer to avoid this and use:

Name 1. _____

Field names surrounded by **curly brackets** { } will be used in preference to those normally generated, so that you can control the process. Thus, **What is your {name}** will result in the field name **NAME** rather than **WHATISYOUR**. More than one set of curly brackets can be used to form a field name if they are on the same line, as in **{What} is your {name}** , giving the field name **WHATNAME**.

It is also possible to change field names after the data file has been created, as described in the chapter on the CHECK program.

A Sample Questionnaire

The following questionnaire might be used to summarize the results collected by an interviewer. Questionnaires may be up to 500 lines long and have more extensive comments or questions, but this is a good example of a form used to abstract data from a longer, written questionnaire.

```
        Questionnaire for Church Supper  -- Oswego

Name _____        Case ID <IDNUM>

Telephone <phonenum>

Age: ###     Sex: <A>     Time of Supper(24 hour):####

Ill? <Y>  Diarrhea? <Y>  Vomiting? <Y>
Onset Date (MM/DD): <mm/dd> Onset Time (24 hour):####

Food History (Mark Y or N):

Baked Ham:        <Y>
Spinach:          <Y>
Mashed Potatoes: <Y>
Cabbage Salad:    <Y>
Jello:            <Y>
Rolls:            <Y>
Brown Bread:      <Y>
Milk:             <Y>
Coffee:           <Y>
Water:            <Y>
Cakes:            <Y>
Ice Cream:  {Vanilla}: <Y>  Chocolate: <Y>
Fruit Salad:      <Y>
```

Note the curly brackets around "Vanilla," so that the field name will be VANILLA rather than ICECREAMVA.

When you have finished typing this or another questionnaire, press the <F9> key to save the file, give it a name ending in .QES (e.g., MARY.QES), and then press <F10> to leave EPED. In a working, rather than a class environment, the file name should, of course reflect the contents of the file, such as the place or other identifier for the investigation.

Making a Questionnaire with the Help of EPIAID

In addition to word processing facilities, EPED contains a system called EPIAID that provides tutorials and assistance with text composition. One of the programs in EPIAID is designed to assist in making questionnaires. Later chapters tell much more about EPIAID, but the questionnaire program will serve as an introduction.

To run EPIAID from EPED, press the <F3> or EPIAID key. A menu of choices appears. Move the cursor bar to "Make a Questionnaire" and press <Enter>. Then follow the instructions to make a questionnaire. You may choose the "Foodborne Questionnaire" or less specific but flexible guidance for questionnaires on other topics. Advice on questionnaire composition is provided if you desire, but can be skipped if you are already expert or have run this program.

Related Topics in Other Chapters

Chapter 25 covers making several questionnaires to be linked together in a single (relational) data entry system. Methods for adding range checking, skip patterns, automatic coding, repeat fields, and other features to questionnaires are described in Chapter 10 on the CHECK program. Advanced methods for combining several fields logically or mathematically, linking to programs from other languages, etc., are described in Chapters 22 and 23.

```
              Questionnaire for Church Supper - Oswego

     Age:▮                   Sex:▮           Time of Supper(24 hour):▮▮▮▮

     Ill? ▮       Onset Date (MM/DD): ▮▮▮▮         Onset Time (24 hour):▮▮▮

     Food History (Mark Y or N):
             Baked Ham:▮
             Spinach:▮
             Mashed Potatoes:▮
             Cabbage Salad:▮
             Jello:▮
             Rolls:▮
             Brown Bread:▮
             Milk:▮
             Coffee:▮
             Water:▮
             Cakes:▮
             Ice Cream:    Vanilla: ▮         Chocolate:▮
             Fruit Salad:    ▮



AGE: Integers allowed
<Ctrl-N>-New  <Ctrl-F>-Find  F5-Print F6-Delete F9-Choices  F10-Done  Rec=  76
```

Chapter **8**

Entering Data Using the ENTER Program

Tutorial

Create a questionnaire as described in the previous chapter. Be sure that it has a name ending in ".QES".

Run ENTER from the main menu.

In the space for "Data file (.REC):", type the name of your questionnaire, but leave off the ".QES". If your questionnaire was MARY.QES, then type MARY, or, if you prefer, MARY.REC.

Enter "2" as the menu choice and press <Enter>. The program asks for the name of the questionnaire file. To see a list of .QES files, press <F9> and then move the bar cursor with the arrow keys and press <Enter> to make a choice. You may also choose to type the name of your questionnaire file, e.g., MARY (or MARY.QES).

Press <Enter>. The program will make a data file (.REC file) from your questionnaire and display the questionnaire on the screen, ready to receive data.

Fill in the blanks in the questionnaire, pressing <Enter> after each until you have completed the form. The message "Write data to disk? (Y/N)" will appear at the bottom of the screen. Answer "Y" or "y". Note that record number 2 is now displayed in the lower right corner of the screen.

> **Complete several questionnaire records. Try the arrow keys and the <PgUp>, <PgDn>, <Home>, and <End> keys, and watch where they place the active cursor in the questionnaire.**
>
> **After saving the last record, use the <F7> and <F8> keys to move back and forth through the records and review your work. Note that you can move to the next available empty record with <F8>, but not beyond this point.**
>
> **Edit one of the entries by placing the cursor in any field and changing the contents. Then press the <End> key and "Y" to save the edited copy.**
>
> **See the section on "Finding Records That Match Criteria" in this chapter and try a few searches. Note that <Ctrl-F> displays the function keys that are used in finding records.**
>
> **To return to entry mode and add a new record, press <Ctrl-N> for "New record." Note that you are now in the next available empty record.**
>
> **To quit the program, press <F10>.**

Overview

The ENTER program will create a data file from a questionnaire. Once the file is created, ENTER guides the data entry process so that any number of data records ("copies" of the questionnaire) may be added.

To use ENTER, first create a questionnaire as described in the previous chapter, using EPED in the QES mode. You can also create the questionnaire in another word processor such as WordPerfect if you save it as a plain text (DOS) file and give it a name ending in .QES. The ENTER program will create an Epi Info database (.REC) file using the questionnaire to provide information on the file structure. Once created, the file may be loaded into ENTER for adding more records or editing those already entered.

If the questionnaire is revised, ENTER can be instructed to revise the .REC file accordingly. Data verification through duplicate data entry is offered as one method of quality control for the data-entry process. ENTER can also be instructed to rebuild the index file for "key" fields in the data file, a process that is described in a later chapter.

Version 6.03 of Epi Info contains a new program called ENTERX, that uses Random Access Memory (RAM) above 640 K. It allows more complex questionnaires, larger check files, and more related files at one time than the

standard ENTER. If you are having difficulty with RAM capacity, try running
ENTERX rather than ENTER.

Running ENTER

On the main EPI menu, move the cursor bar to the ENTER program choice and
press <Enter> or simply type "N" and press <Enter>.   A screen will appear, asking
for the name of the data file, and displaying the following menu choices:

```
1. Enter or Edit data
2. Create new data file from .QES file
3. Revise structure of data file using revised .QES
4. Reenter and verify records in existing data file
5. Rebuild index files(s) specified in .CHK file
```

Editing an Existing Data File (Menu Choice 1)

Epi Info database files have names ending in ".REC," because they contain data
RECords. To enter new data or edit existing data in an existing data (.REC) file,
enter the name of the data file, or press <F9> to see those available. Choose the file
from the menu by moving the bar cursor with the arrow keys and then press
<Enter> to choose the file. Enter a "1" for the menu choice, press <Enter>, and the
the file will appear on the screen ready for editing or addition of new records. More
details on entering and editing data are given in subsequent sections of this chapter.

Making a New Data (.REC) File (Menu Choice 2)

ENTER constructs a new data file by reading the questionnaire (.QES) file and
using the information to set up a new data (.REC) file. The format of the
questionnaire is stored at the top of the file (the file "header"), followed by the data
records, one for each "copy" or data record. The internal structure of Epi Info files
is given in Chapter 37, although the details are not needed to use Epi Info.

To make a new data file, first make a .QES file as described in the previous chapter.
In the ENTER program, give the name of the data file to be created (the .REC file),
press <Enter>, and then enter "2" for the menu choice. ENTER will now ask for
the name of the .QES file. Usually it is convenient to use the same file name for the
data and questionnaire files, calling them BOTULISM.REC and BOTULISM.QES,
for example. There is no need to type the suffixes .REC and .QES because they are
supplied by the program.

When the cursor is in either the "Data file" or "Questionnaire file" blanks, you may
press <F9> to see the available files. To see those in another directory or on
another disk, type the disk letter and/or directory and then press <F9>. Moving the
cursor bar around with the arrow keys and then pressing <Enter> will choose a file

from the list.

Questionnaire file names must end in .QES to be recognized. Suppose that you have created a questionnaire called OUTBREAK.QES. When you enter OUTBREAK in the data file blank, you will be asked for the questionnaire name; enter OUTBREAK.

When you have specified both the .REC and the .QES files, ENTER takes a few seconds to read the questionnaire and make a data file and then displays the questionnaire on the screen. You are now ready to enter data.

Entering Data

Entering data means typing the appropriate responses in the blanks on the screen. The cursor will move automatically from blank to blank.

At the bottom of the screen the prompt line displays the available commands with the current record number at the right end of the line. Pressing <Ctrl-F> (for "Find") will display other commands used for searching and moving from record to record.

Just above the prompt line, the type of data that will be accepted in the field containing the cursor is displayed. These limits can be changed using the CHECK program described in a later chapter, but for now, they reflect only numeric, general purpose, date, and other field types.

The following commands have special meaning during data entry:

Backspace   Deletes the character to the left of the cursor.

Right Arrow   Moves the cursor one character to the right or left.
Left Arrow

Up Arrow   Moves to the previous or next entry field.
Down Arrow

<**PgUp>**   Moves the screen up or down a page.
<**PgDn>**

<**Home>**   Moves to the first entry field in the questionnaire.

<**End>**   Moves to the last entry field in the questionnaire.

<Ins>ert   Turns the insert mode on or off. When insert mode is
                  off, new characters will overwrite (replace) old
                  characters. When it is on, new characters will be
                  inserted between the old characters.

<Del>ete   Deletes a character at the cursor location.

<F10>   Closes files and exits from the program. (Other
                  function keys are described in the following sections
                  and summarized at the end of this chapter.)

When you enter data:

- Fields will accept only the type of data shown above the prompt line.
- Pressing <Enter> in a blank field without entering data enters a missing value.
- When a field is full, the cursor moves automatically to the next field unless this feature has been turned off as described at the beginning of Chapter 35.
- Errors on entry are signaled by a "beep" and can be corrected immediately.
- Each blank is called a "field." Each copy of the questionnaire you complete is called a "record." The records are stored in the file with the name ending in .REC, called either the data file or the .REC file.

After the information has been entered in the last field of a questionnaire, the question "Write data to disk (Y/N)?" appears at the bottom of the screen. Replying "Y" or "y" saves the record and brings up the next available empty record. Note that the record number on the lower right changes. If the reply is "N", the cursor jumps to the first field in the questionnaire and you have another opportunity to edit the record.

To exit from the program at any time, use the F10 function key. If you haven't saved the current record, ENTER asks if you would like to and then returns to the EPI6 menu or to DOS.

Moving from Record to Record

The current record number is always shown in the lower right corner of the screen. To move to the **previous** record, press the <F7 key>. <F8> will go to the **next** record in the file, if any.

If you have been "browsing" through the file with <F7> and <F8> and now wish to enter another record, hold down the <Ctrl> (Control) key and press "N" for "New."

(You may need to press <Esc> first if a message appears as you encounter either end of the file.) The <Ctrl-N> will clear the screen and move to the next available new record for data entry.

Finding Records That Match Criteria

ENTER allows you to search the data file and **find** records that match your criteria. To find a record, first be sure the current record has been saved, if necessary. Then press <Ctrl-F> for "Find," and type in the items you want to find, followed by <F3> for "Find 1st". You might type SMITH in a "Lastname" field, for example, and then press the <F3> key. All the records with LASTNAME = SMITH will be listed on the screen.

Use the up and down arrow keys to move to the record desired and select it by pressing <Enter>. The record will then appear on the screen in questionnaire format.

ENTER will search in this way using **any combination of fields that you enter after pressing <Ctrl-F> for the FIND command**--a multi-field search on any combination of fields. You could have entered "Smith", "John", and "47" for the age. The search would then be confined to records containing 47-year-old John Smiths.

Sometimes it is useful to do a "fuzzy" search, looking for all names beginning with "Peter", for example, if you are not sure how "Petersen" or "Peterson" was spelled. You can use the "wildcard" character to specify the search, entering "Peter*" in this case. Wildcards are only used in non-numeric fields.

SOUNDEX searches, described further in Chapter 35, are another way to do approximate matches. A SOUNDEX search can be done by pressing <Ctrl-S> rather than <Ctrl-F> to begin the search process.

If you have only a few hundred records, searches will be reasonably fast. Epi Info provides an indexing facility so that searches of thousands of records can be done almost instantaneously. Creating indexed fields is described in Chapter 22 on the CHECK program. From the user's point of view there is no difference in the commands for searching; indexing merely makes the search faster.

A command called AUTOSEARCH is also described in Chapter 22. AUTOSEARCH performs a search for duplicate records as soon as you have entered a selected data item in the questionnaire. It might be set up to perform the search after you have entered first and last names, for example. A matching record already in the file will be called to your attention so that additions can be made or exact duplication avoided.

Editing Records

To edit a previously entered record, first find and retrieve it as described above. Then change any of the items in the record, entering a new item with the aid of the arrow, delete, and insert keys. Be sure to press <Enter> when leaving a field that has been changed. When you have made all the desired changes, go to the last field in the record and press <Enter>, or merely press the <**END>** key. The question "Write data to disk (Y/N)?" will appear at the bottom of the screen (although this feature can be turned off--see Chapter 35). If you reply "Y", the record as it now appears will replace the old record in the file. If you type "N," the cursor will return to the questionnaire. If you move to another record with <F7> or <F8> without saving the edits you have made, they will be discarded and the edited record will revert to its previous form in the file.

Another method of editing and permanently changing records will be described in the chapter on the ANALYSIS program under the UPDATE command. This allows editing in a spreadsheet format that may be more convenient for changing the same field in many records, for example.

Deleting or Undeleting Records

Pressing the <F6> key will mark the current record as deleted. An asterisk (*) will appear next to the record number in the lower right corner of the screen. The data items in the record are still visible in the ENTER program, but tabulations done in the ANALYSIS program will skip this record. The WRITE RECFILE command, described later in the chapter on ANALYSIS, will create a copy of the data file without the deleted records and thus provides a permanent way to rid the file of such records, if desired.

A record, once deleted, may be made active again (undeleted) by pressing the <F6> key again. This key thus alternates (toggles) between deleting and undeleting.

Revising the Structure of a Data File (Menu Choice 3)

If ENTER detects that the date/time of a .REC file is older than that of the .QES file with the same name, you will be asked if you would like to update the .REC file from the .QES.

At any time, you can revise a questionnaire (.QES file), using EPED, and then create a new data file in the revised format. This often occurs when you have entered a number of records and suddenly have an inspiration about adding a few questions or wish to correct an error in the questionnaire.

Revising the **structure** of the data file should be carefully distinguished from simply revising the data, as described above for the editing process. To revise the questionnaire structure, return to EPED and make appropriate changes in your original .QES question naire file. Then return to the ENTER program. Enter the name of the .REC file under "Data file:" and press <Enter>. Enter "3" for the menu choice, and then give the name of the revised .QES file for the questionnaire.

When you press <Enter>, ENTER will automatically make a .REC file in the new format, merge the old data into the new file, and then rename the old .REC file to .OLD. The process is described on the screen as it is performed.

The process works correctly only if the field names in the .REC file have not been changed. If you have used the CHECK program to change the field names in the .REC file, you will need to edit the questionnaire to produce these same field names, or the previous field names will be constructed from the questionnaire again.

The MERGE program can also be used to revise a .REC file--in fact, ENTER is really calling MERGE to do the work after setting things up properly. The details of updating are described in the chapter on MERGE, but the ENTER program takes care of these details in most instances.

Reentering and Verifying Records in a Data File (Menu Choice 4)

After records have been entered in a data file, ENTER allows another operator to enter them again and verify that they have been entered correctly. To do this, run ENTER and give the name of the existing data file, choosing "Reenter and verify records in existing file" from the menu. The name of one or more fields uniquely identifying each record is requested. The data in the identifier field(s) will be visible on the screen; other data will not until it has been entered the second time.

When verification is active, the user enters (or accepts) data in the identifier (key) fields and the program searches for a record matching these entries. Records not matching a record in the file cannot be entered. When a match is found, the identifying fields are shown on the screen.

The operator enters data exactly as for new records. When a field entry matches the data in the file, the cursor moves as for new entries. When an entry does not match, however, a window appears, offering a choice between reentering (<Esc>) or seeing the old and new values (<F9>. Pressing the <F9> key displays a help window with the old and new entries. A choice can be made between them by pressing <F1> or <F2>, and the chosen value appears in the field.

When the last entry in the record has been made, the statement, "Write Data to

Disk? (Y/N)?" appears. A "Y" answer causes the record just entered to be compared with the record in the file. Any fields that do not agree are highlighted, and the cursor is placed in the first of these. The user can then reconcile these fields and press the <End> key or go past the last field to bring up the "Write Data to Disk" prompt and repeat the check. When the check has been satisfied, the next record is brought up.

Verified (double-entered) records will be identified by a change in the end-of-record marker in the data file (from "!" to "^"). In ANALYSIS, a new system variable called VERIFIED will be set to Y if the record is verified and N if it is not. This allows writing programs that handle verified or unverified records in special ways.

Rebuilding Index Files (Menu Choice 5)

Index files specified in .CHK files (see later chapters) are normally maintained automatically. Occasionally an index file may become damaged. This can happen if a power failure occurs while an index is being updated or a copying error destroys an index file. In such cases, the Choice 5 on the ENTER menu is available to correct the problem. After specifying a data file, Choice 5 will rebuild the index files for the database. Although this may take a few minutes, the index files will be rebuilt automatically so that they correspond to the data currently in the .REC files.

Printing the Questionnaire and the Data

To print a blank copy of the questionnaire or to list records from the file, press <F5>. ENTER asks if you want a copy of the questionnaire to be printed. If you do, enter 'Y'; otherwise, enter 'N'. The questionnaire will be printed with all entry blanks indicated by underline characters, suitable for pencil-and-paper entry.

You are then asked for the starting and ending record numbers. Be sure that your printer is ready, and enter the numbers of the first and last records to be printed, or zero for both if you want only the questionnaire. Answer "Y" to Single Sheets (Y/N) only if you want the printing to stop after every page for insertion of a new sheet of paper in the printer. After a short pause, ENTER sends the data to the printer. Records are printed in condensed form, as shown below:

```
Record number 60
AGE         :53    SEX          :F  TIMESUPPER :1930    ILL         :Y
ONSETDATE   :04/18                  ONSETTIME  :2330    BAKEDHAM    :Y
SPINACH     :Y     MASHEDPOTA :Y  CABBAGESAL :Y         JELLO       :Y
ROLLS       :N     BROWNBREAD :Y  MILK       :N         COFFEE      :Y
WATER       :Y     CAKES      :Y  VANILLA    :Y         CHOCOLATE :Y
FRUITSALAD :N
```

Another way to list the records is provided in the analysis program, where the LIST
* command will print the data in column-and-row format, and the FORM command
may be used to print the entire questionnaire with data inserted in the blanks.
Printed copy produced by any of these methods may be used for various purposes
such as hard-copy backup, checking data, or forwarding copies to other interested
parties.

Using ENTER When a Check File Is Present

".CHK" files are explained in the chapter on the program called CHECK. They
contain commands that guide and refine the data entry process. For further details
on setting up .CHK files, please see the chapter on CHECK.

.CHK files are optional and offer a convenient way to set up constraints for data
entry that will help to eliminate errors at this stage.   .CHK files can be set up or
changed at any time, even though some records have already been entered. They
perform checks only on data entered while the .CHK file is present, not on
previously entered records.

As the cursor moves to each field in ENTER, the name of a variable appears at the
bottom of the screen with a message giving the type of entry that will be accepted.
If no .CHK file is active, this may be "All entries allowed," or "Integers allowed."

If a check file is present with the same name as the data file (e.g., OSWEGO.CHK
for OSWEGO.REC) or one by another name has been specified, messages about
legal values, ranges, and codes may appear for some of the fields. These messages
should be self-explanatory; they indicate what type of entry will be accepted in each
field.   If the data item entered does not meet the criteria, the program will beep and
request another entry. Normally, no exceptions to legal values and codes are
allowed, but a special command when ENTER is first run, called the /OVER
parameter, allows exceptions to legal values or codes to be entered with the <down
arrow key>. The /OVER parameter is described at the beginning of Chapter 35.

If the field you are entering is specified as a "jump" field in the CHECK program, the
cursor may skip over parts of the questionnaire when you enter certain values, skipping
the symptom questions if you indicate that no illness occurred, for example. This
allows use of a single questionnaire for several purposes.

If a field contains codes set up through CHECK, entering a value may cause the
corresponding member of a code pair to appear in another field, or even in the same
field. Thus, entering 101 may cause "Hepatitis A" to appear, or "A" may result in
"Hepatitis A".

In some fields, pressing <F9> will display a window of legal values or codes on the

screen. One of these can be chosen by moving the cursor bar with the arrow keys and pressing <Enter>. Typing one or more letters will position the cursor bar on the closest match. Typing "J" will move the cursor to the first code beginning with "J," for example.

Summary of Special Function Keys

The special function keys are listed at the bottom of the screen. Those used for finding records become visible when you press <Ctrl-F> or <Ctrl-S>.

<F1> Help Displays a help file created for the current questionnaire if one is specified in the current .CHK file. In this case, the questionnaire should contain a reminder such as "Press <F1> for help," so that the user will be aware of the resource. Pressing <Shift-F1> will display help specific for the field containing the cursor.

<F2> Rec# Finds a particular record number. Enter a number, and the record with that number will be brought to the screen**.**

<F3> 1st Searches for a record with information matching the information found in all non-blank fields except <IDNUM> and <Today> fields. The search starts from the beginning of the file and goes forward. Partial matches ("SMI*" is a partial match of "SMITH") can be specified on non-numeric fields by using the asterisk or "wildcard" character.

<F4> Next Same as <F3>, except the search starts from the current file position and goes forward. Finds the next occurrence of a record matching the search criteria**.**

<Shift-F4> Prev Same as <F3>, except the search starts from the current file position and goes backward. Finds the previous occurrence of a record matching the search criteria**.**

<F5> Print Allows you to print the questionnaire or the data stored in specified records. Prompts will appear at the bottom of the screen to guide you through the printing process**.**

<F6> Delete Marks the current record as deleted. Such records will be ignored in the ANALYSIS program. If the record has already been deleted, pressing this key will undelete it (restore the record to active status). Deleted records are indicated by an asterisk in the lower right corner of the screen beside the record number. The computer will beep when this key is pressed; you can turn this warning signal off or on by pressing <Shift-F6> (or <Ctrl-F6>).

<F7> Rec< Moves back to the record just before the one on the screen**.**

<F8> Rec> Moves forward to the next record, if any.

<F9> Choices Displays codes, if any, that are set up for automatic coding in the current field. Choices may be made from the menu that appears by moving the bar cursor with the arrow keys and then pressing <Enter>. Typing a letter of the alphabet will move the cursor to the first choice beginning with that letter. A choice can also be typed after pressing <Esc> to remove the menu**.**

<F10> Done Leaves the ENTER program and returns to the EPI6 menu or to DOS.

<Ctrl-N> New Clears all the fields in the current record and sets the record position to point to the next new record to be added to the data file. Use this command if you have finished editing records and wish to enter new records, or to leave the search (<Ctrl-F>/<Ctrl-S>) mode.

<Ctrl-F> Find Prepares the screen to enter criteria to find specific records.

<Ctrl-S> SOUNDEX Prepares the screen to enter criteria for a SOUNDEX search.

Changing the Standard Disk Directory and Color Settings

On the main ENTER menu, note the function key command "F2 Setup" at the bottom. When you press the <F2> key, the Default Value Screen appears. The settings entered here apply to other parts of Epi Info and remain in effect until changed, even though the computer is turned off.

The "Default Data File Path" sets the disk and directory in which data files will be stored. If no entry is made, the files will be stored in the current directory. Entering a drive or directory here will cause the EPED, ENTER, and ANALYSIS programs to create and access data files in the specified directory. If you would like to keep data files in a directory called "\DATA" on disk "C:", for example, enter **C:\DATA** in this blank. If you are using a computer with floppy drives only or if you are working on a computer belonging to someone else, you may want to keep all data files on a floppy disk by entering the letter of the Drive and a colon in this blank, for example, **A:.**

The color choices refer to the colors that are used when a data file (.REC) file is made by the ENTER program. Usually this is a blue background (color 1) with yellow-on-blue (30) text , black-on-gray (112) data entry fields, and black-on-red (64) search fields. The "field character" used in entry fields is initially set to blank (space), although you may prefer the underline character. If you wish to change these settings, choose a number from the color display in the lower half of the

screen for each item. Note that some combinations (any color on itself as background) make the text invisible and would be poor choices unless you prefer not to display the text. Background colors must be between 0 and 7. Further customization of the colors and "field characters" in the questionnaire is available in the CHECK program, as described in Chapter 10.

Finding More Memory with ENTERX

ENTERX is a new program in Verson 6.03 of Epi Info that uses higher memory, beyond the usual DOS limit of 640 K. Operating ENTERX is identical to running ENTER. If you have more than 640 K of Random Access Memory (RAM) and a 286 or higher processor and wish to work with large questionnaires or .CHK files or with a number of related files (see Chapter 25), try running ENTERX from the DOS command line by typing **ENTERX**.

If all goes well, you may wish to save your ENTER.EXE program under another name and rename ENTERX.EXE to ENTER.EXE so that it is routinely accessed from the EPI6 Menu. ENTERX does not use the ENTER.OVR file that is necessary for ENTER.EXE, but it does require the presence of two other files, DPMI16BI.OVL and RTM.EXE, facts you need to know only if you copy ENTERX to another location for separate use.

Related Topics in Other Chapters

Example Chapters 14, 23, and 26 describe questionnaires on the Epi Info disks that can serve as models for more advanced data entry functions. The CHECK program described in Chapters 10 and 22 can be used to add many features to the data entry process such as range checking, legal values, skip patterns, automatic coding, help windows, cross-field checking, mathematical and logical functions, and even linking in programs written in other languages. The results of such programming can be seen in the example chapters.

```
Dataset:  C:\EPI6\OSWEGO.REC (75 records)           Free memory: 295K
Criteria: All records selected                      Time:    0.11 sec
─────────────────────────── Output - Screen ──────────────────────────

                        ILL
|VANILLA     |    +      -  | Total

         +      43     11      54
         -       3     18      21

    Total |     46     29      75


                    Single Table Analysis


|Odds ratio                                                    23.45
|Maximum likelihood estimate of OR (MLE) (2)                   22.15
|Exact 95% confidence limits for MLE (2)          5.22 < OR < 138.39
|Exact 95% Mid-P limits for MLE (2)               5.93 < OR < 109.15
|Probability of MLE >= 22.15 if population OR = 1.0 (2)   0.00000026
|<more>
─────────────────────────────────────────────────────────────────────

|EPI> read oswego
|EPI> tables vanilla ill

 F1-Help  F2-Commands  F3-Variables  F4-Browse  F5-Printer on  F9-DOS  F10-Quit
```

Chapter **9**

ANALYSIS: Producing Lists, Frequencies, Tables, Statistics, and Graphs from Epi Info Files

---

Tutorial

Run the ANALYSIS program from the main menu.

When the prompt "EPI6>" appears, type:

    RUN TUTOR1

Enjoy the tutorial.

To leave the program at any time, press <F10>

Overview

ANALYSIS produces lists, frequencies, tables, statistics, and graphs from Epi Info or dBASE files. Simple commands will cause ANALYSIS to select records using specified criteria, sort or list records, do frequencies or cross tabulations, do logical or mathematical operations on a variable, put the results in a new variable, and direct the results to the screen, a printer, or a disk file.

The commands constitute a programming language, and they may be entered one by one from the keyboard or placed in a program file that is then "run" from ANALYSIS. In this chapter, we will describe only the simpler operations--those that might be used during a field investigation for preliminary analysis of a questionnaire. They include those needed to analyze a questionnaire and produce line lists, frequencies, cross tabulations, means, and graphs. The writing of programs to "clean" data sets and for permanent disease surveillance systems is described in later chapters.

We suggest that you run the tutorial as instructed above. It covers most of the material in this chapter interactively. Chapters 11, 13, 25, and many other chapters contain further information about ANALYSIS. For particular commands, the reference section in Chapter 36 provides much more detail about additional uses and options than can be given in this introductory chapter.

Getting Acquainted

By running ANALYSIS from the EPI6 menu, you will see a lower window for entering commands and a larger one above where the results of the commands will appear. At the bottom of the screen some key commands are indicated, and information about data files and available memory is displayed at the top.

Whenever the cursor is at the EPI6> prompt, commands may be entered from the keyboard. If you make a mistake or want to change part of the line being entered, use the <Backspace> or <Left arrow> key to return to the correct position and then enter the new material. The <Del> and <Ins> keys work as in EPED, deleting characters and turning the Insert mode on and off. The <Home> and <End> keys skip to the beginning or end of the current line.

Pressing the function keys shown at the bottom of the screen allow selection of help topics, commands, and variable names from lists that appear on the screen. After pressing one of the function keys, choose an item by moving the highlight bar with the up- and down-arrow keys and pressing <Enter>.

As results appear in the upper window, those that have scrolled out of sight at the top of the screen may be reviewed by pressing the <PgUp> and <PgDn> keys to move up and down one screen at a time. For finer movements, <Ctrl-PgUp> and <Ctrl-PgDn> will move one line at a time.

The Help System

To see a list of the commands in ANALYSIS, press <F2>, the COMMANDS key. A list of commands will appear in a window on the screen. Moving the cursor to a command and pressing <Enter> will place that command next to the prompt. You may also type commands from the keyboard.

Whenever a command has been entered--READ, for example--pressing <F1> will bring up further information about that command, including the format in which field names and other information must be supplied. If no command has been

entered, <F1> brings up a list of topics; choosing a topic then produces information on that topic.

For many commands, such as READ, typing the command and then <Enter> will bring up a menu of appropriate choices--in this case, the files that can be read. Choosing a file then carries out the command with the appropriate file or variable.

To summarize, help is never more than one keystroke away:

- For **general information** on a topic, press <F1> without a command on the EPI6> line, and then choose a topic from the list.
- For **command** information, press <F2> for commands.
- For **more information about a command**, enter the command and then press <F1>.
- For **file** information, type READ <Enter>, and then choose the file from the menu that appears.
- For a list of **variables** in the current file, press <F3> for variables and choose from the list. Several variables can be "tagged" with the <+> key or "untagged" with the <-> key before pressing <Enter> to finalize the choice(s).

The help system is intelligent enough to change the order of words in a command; if you choose a variable and then a command from the menus, their order is automatically reversed.

The First Step--READing a File

Analysis must be performed on the records in a **file**. The file may be either an Epi Info file, produced by entering data with the ENTER program, or a dBASE file from another source. Epi Info files can be produced from many other formats by the IMPORT program (Chapter 17), and the resulting file can be used in ANALYSIS.

The command that tells ANALYSIS what file to use is READ <file name>, and this is usually the first command given in ANALYSIS.

To see a list of available files, type

EPI6> **READ**

and press <Enter>. To see the files in another directory, such as c:\Data\ , you, first type READ c:\Data\ before pressing <Enter>. To see all dBASE files, type READ

*.DBF. A directory of files will appear in a window. Move the cursor bar with the arrow keys and choose a file by pressing <Enter>.

Whether you use the file directory or simply type READ and the file name, ANALYSIS will use this file for all subsequent operations until another READ is performed. It has become the "active data set."
ANALYSIS will also read and process files made in the dBASE format. To use a dBASE file rather than an Epi Info file, simply type the file name followed by .DBF. The only operation in ANALYSIS that cannot be performed with a dBASE file is UPDATEing of records, described in a later section.

Allowable READ commands include the following:

> READ OSWEGO   (OSWEGO.REC is understood)
> READ OSWEGO.REC
> READ OSWEGO.DBF   (A dBASE file)
> READ OSWEGO1 OSWEGO2 OSWEGO3   (Three files in the same format to be analyzed as a single unit)

For the rest of this chapter, we will assume that you have chosen the file OSWEGO.REC, supplied with the Epi Info system, by giving the command:

EPI6>**READ OSWEGO**

OSWEGO.REC, often used in teaching exercises, contains interview information from a foodborne outbreak . The file contains the results of interviews with attendees at a church supper that was followed by a gastroenteritis outbreak.

Producing a Line Listing

The first step in data analysis is to scan the data visually to gain an overall impression and see what further analysis might be appropriate. A "line listing" is helpful for this purpose. To produce a listing of the records in the file, type:

EPI6>**LIST**

The result is:

REC	AGE	SEX	TIMESUPPER	ILL	ONSETDATE	ONSETTIME	BAKEDHAM	SPINACH	MASHEDPOTA
1	11	M	9999	N			N	N	N
2	52	F	1500	Y	04/19	30	Y	Y	Y
--- ETC. ---									

The command LIST will display only as many variables as will fit across the current screen width. If you would like to list all variables, use

EPI6>**LIST ***

The "*" is shorthand for "all fields." LIST followed by one or more variable names lists only those variables, a handy way to produce a list of names and addresses of persons to be interviewed, or to omit the names and addresses from a data listing.

LIST * NOT NAME ADDRESSwill list all fields *except* NAME and ADDRESS.

Frequencies

The frequencies command (FREQ) will count each category for a specified variable and give the absolute and relative frequencies for each category.

EPI6>**FREQ SEX**

produces this result:

```
 SEX   |  Freq  Percent   Cum.
------+----------------------
F     |    44   58.7%    58.7%
M     |    31   41.3%   100.0%
------+----------------------
Total |    75  100.0%
```

The number (frequency) in each category is given first, followed by the percentage of the total and the cumulative percentage. If statistics is SET to ON (see the SET command in Chapter 36 for more details) and the field types are numeric, the sum, mean, and standard deviation are also printed.

The command FREQ * will produce frequencies for all the variables in your questionnaire, a convenient way to begin the analysis of a new data set. FREQ with a series of variable names following, separated by spaces, will do separate frequencies for each of the variables listed.

As outlined in the Chapter on Statistics, placing **/C** after the FREQ command will add confidence limits to the output. Thus **FREQ SEX /C** will produce the frequency with confidence limits for each value. A sample of the results is shown in Chapter 32.

Cross Tabulations: the TABLES Command

The TABLES command will count the records in which the values fulfill criteria for two fields at the same time. Thus, the command

EPI6>**TABLES SEX ILL**

will result in:

```
                        ILL
SEX          |     +      -  | Total
-----------+-------------+------
        F  |    30     14  |    44
        M  |    16     15  |    31
-----------+-------------+------
    Total  |    46     29  |    75

                     Single Table Analysis

Odds ratio                                                    2.01
Cornfield 95% confidence limits for OR            0.70 < OR <   5.85
Maximum likelihood estimate of OR (MLE)                       1.99
Exact 95% confidence limits for MLE               0.70 < OR <   5.76
Exact 95% Mid-P limits for MLE                    0.77 < OR <   5.25
Probability of MLE >=  1.99 if population OR = 1.0      0.11324319


RISK RATIO(RR)(Outcome:ILL=+; Exposure:SEX=F)                 1.32
95% confidence limits for RR                      0.89 < RR <   1.96

              Ignore risk ratio if case control study

                      Chi-Squares    P-values
                      -----------    --------

        Uncorrected:       2.11     0.14679237
        Mantel-Haenszel:   2.08     0.14951418
        Yates corrected:   1.46     0.22620244
```

Note that the interpretation of relative risk (risk ratio) depends on the orientation of the table, and that not all relative risks will be meaningful. The relative risk given is the rate of the first factor on the first line compared with the rate of the first factor on the second line or (a/a+b)/(c/c+d) as defined in Chapter 32. An interpretation is printed with the relative risk value. If risk factors are on the left side of the table and disease across the top, with presence indicated first in each case, the relative risk represents the risk of disease for persons with the first factor relative to those with the second factor. **As indicated in the note, relative risk should be ignored in a case-control study, since it cannot be interpreted meaningfully.** Additional material on statistical output is provided in Chapter 32.

Stratified Analyses

Series of stratified tables can be produced by listing more than two variables in the TABLES command. The variables after the first two serve as the basis for dividing the tables into levels or **strata**, one for each combination of variables after the second. For a quick preview, try

EPI6>**TABLES VANILLA ILL SEX**
(Tables of VANILLA consumption by ILLness, stratified by SEX)

This will result in separate tables for male and female cases. A Mantel-Haenszel stratified analysis will follow the two tables. The summary of the statistical combination of the two strata appears as follows for VANILLA ILL SEX:

```
              ** Summary of 2 Tables With Non-Zero margins **
                            N =     75

Mantel-Haenszel Summary Chi Square                          25.28
P value                                                0.00000050
Mantel-Haenszel Summary Chi Square (uncorrected)            28.01
P value                                                0.00000012

                     SUMMARY ODDS RATIO
Crude OR                                                     23.45
Mantel-Haenszel weighted Odds ratio                         37.64
95% conf limits for M-H OR
   (Robins, Greenland, Breslow)            7.19 < OR < 197.01
Maximum likelihood estimate of OR (MLE)                     28.58
Exact 95% confidence limits for MLE        6.01 < OR < 207.50
Exact 95% Mid-P limits for MLE             6.92 < OR < 160.22
Probability of MLE >= 28.58 if population OR = 1.0     0.00000011

Unable to test for interaction due to zero cell values

                    SUMMARY RISK RATIO (RR)
               (Ignore if Case-Control Study)
Crude RR without stratification                              5.57
Summary RR of (ILL=+) for (VANILLA=+)                        5.73
95% confidence limits for RR               1.99 < RR <  16.51
               ** End of Stratified Analysis **
```

The very large odds ratio indicates a strong association between eating vanilla ice cream and the occurrence of illness. The Mantel-Haenszel weighted odds ratio, summarizing the results of stratification by sex, is even larger. Both the confidence limits and the extremely small p value indicate that the Mantel-Haenszel weighted odds ratio is significantly different from 1.0, the value indicating no association.

Both the crude and stratified relative risks are also significantly greater than 1.0. In this outbreak of staphylococcal food poisoning in Oswego, New York, the vanilla ice cream was indeed identified as the cause, through both statistical and

microbiologic evidence.

An asterisk or star(*****) may be used in the TABLES command to indicate all variables. Thus, **TABLES ILL *** would produce tables of illness status against each variable in the questionnaire. This is a quick way to obtain preliminary analysis of a questionnaire for which comparison of two groups is the key feature.

Rothman's test for interaction, when available, indicates differences in the odds ratios between strata, if the p value is small. Such differences suggest that the Mantel-Haenszel weighted odds ratio, while still valid as a summary of all strata, does not necessarily reflect the value for one of the strata.

The MEANS Command

The TABLES command is for data items that are arranged in categories and counted. There is another whole world of statistics for numbers that are *continuous,* such as height, weight, and age. The MEANS command produces a table that displays continuous or ordinal data and then performs appropriate statistical analysis.

The MEANS command requires two items of information--the variable containing data to be analyzed and the variable that indicates how groups will be distinguished. The command is:

MEANS [Numeric variable to be analyzed] [Variable for Grouping]

> If you prefer not to display the table of values, append "/N" to the command to indicate "No tables."

Using the OSWEGO.REC file, MEANS AGE ILL will compare ages for persons in whom ILL ="Y" with those for whom ILL = "N".

Using the MEANS AGE ILL example, you should obtain the following results:

```
                         ILL
AGE          |     +      -  | Total
-----------+-------------+------
         3 |     1      0 |     1
.......etc........
        77 |     1      0 |     1
-----------+-------------+------
    Total |    46     29 |    75

ILL              Obs      Total      Mean   Variance    Std Dev
+                 46       1806    39.261    477.264     21.846
-                 29        955    32.931    423.709     20.584
Difference                         6.330
```

```
ILL            Minimum     25%ile     Median     75%ile     Maximum       Mode
+                3.000     17.000     38.500     59.000      77.000      15.000
-                7.000     14.000     35.000     50.000      69.000      11.000

                                     ANOVA
                       (For normally distributed data only)

Variation          SS   df          MS  F statistic    p-value    t-value
Between        712.655    1     712.655        1.560   0.213137   1.249147
Within       33340.732   73     456.722
Total        34053.387   74

                   Bartlett's test for homogeneity of variance
      Bartlett's chi square =   0.119  deg freedom =  1   p-value = 0.729786

                 The variances are homogeneous with 95% confidence.
        If samples are also normally distributed, ANOVA results can be used.

   Mann-Whitney or Wilcoxon Two-Sample Test (Kruskal-Wallis test for two groups)

   Kruskal-Wallis H (equivalent to Chi square) =      1.161
                          Degrees of freedom =          1
                                   p value =      0.281226
```

At first this may be more numbers than you ever wanted, but the overall results can be quickly understood. The means of the two sets of ages are 39 and 33. Now look at the p values under the ANOVA and Kruskal-Wallis tests. Since both are more than 0.05, we can conclude that the difference in mean age between the ILL and NOT ILL groups is not "significant." Bartlett's test helps you decide which one of the two methods to choose, but in this case, both tests lead to the same conclusion. Further details are given in Chapter 32 on Statistics.

Missing Values

Missing values in Epi Info are entered as blanks in the actual records. During data entry, pressing <Enter> in a field rather than entering data will result in a missing value. In the MEANS, TABLES, and FREQ procedures, missing values will be ignored if SET IGNORE is ON (the normal condition; see the SET command in Reference Chapter 36).

If, however, you have used another code, such as 99, for missing values, be sure to select only the non-missing values before using the means procedure. This can be done by using **SELECT AGE <> 99**, for example (<> means "not equal to"). Be particularly aware of this point if the data have been imported from another system in which missing values may be coded differently.

Titles

The TITLE command allows you to specify up to five lines of text that will appear

at the top of a table, frequency, chart, or graph. TITLE can be used before the TABLES, FREQ, and graphics commands to produce an appropriate title for the results. To cancel the effect after the command has been executed, TITLE can be used again with blank entries or new text for the next command. The following commands will define a two-line title for subsequent commands:

TITLE 1 Number of Cases by Date of Onset
TITLE 2 North Chicago, January 1990

To remove this title for later commands use:

TITLE 1
(All titles with numbers 1 or greater are removed.)

Charts and Graphs

ANALYSIS produces histograms, scatter plots, pie charts, and bar and line graphs directly from data files. Making a graph requires a single command; the following examples will give the idea:

> **HISTOGRAM ONSETDATE**
> **PIE RACE**
> **BAR AGEGROUP**
> **LINE DAY**
> **SCATTER SYSBP BODYMASS**

If /R is placed after the SCATTER command, a least squares regression line will be drawn through the data points, as in:

> **SCATTER SYSBP BODYMASS /R**

Sometimes it is important to compare two graphs and to make the scales on the vertical axes the same for this purpose. Adding a slash (a forward slash, not a backslash) to the command and specifying the maximum value of the Y axis will accomplish this, as in:

EPI6>**Bar ROLLS /Y=0-75**

(The lower value must be 0.)
For scatter plots, the maximum and minimum numbers on the X (horizontal) and Y (vertical) axes can be made more extreme by specifying new numbers:

EPI6>**Scatter age BP /X=0-300 /Y=0-100**

The minimum cannot be increased nor the maximum decreased from the values chosen automatically.

Sending Results to the Printer or to a File

Before using the print commands, be sure that a printer is connected to the computer, turned on, and supplied with paper. Pressing the Printer On key <F5> will send all subsequent output to the printer. It does not print results already on the screen. To do this, turn the printer on with <F5> and then press the up arrow to recall the previous command. Pressing <Enter> will run the command again, and this time the results will be printed. To stop sending output to the printer, press <F5> again.

Note that when you press <F5>, the command ROUTE PRINTER or ROUTE SCREEN appears. This key provides a shortcut to these commands, but you can also type them on the command line to control printing.

The ROUTE command will also direct the results to a **file** by the name of your choice.

`EPI6>`**`ROUTE C:RESULTS.TXT`**

will send all results to the file RESULTS.TXT on drive C, until another ROUTE command sends results to the SCREEN, PRINTER, or another file. All files are closed by leaving ANALYSIS with <F10>.

When a graph or chart is requested as in the preceding section and printing is on, the graph will be printed; otherwise, it will appear only on the screen. Printing of graphs and charts can be done on Hewlett Packard-compatible printers and plotters and on IBM and Epson compatible dot-matrix printers, although other printers will usually work for tables and lists. Graphs **may** print with other printers but graphics are more subject to incompatibility problems than plain text. See the section of Chapter 36 on the SET command for methods of setting up ANALYSIS for laser printers and plotters.

Recalling ANALYSIS Commands

As commands are entered, ANALYSIS saves each line for possible reuse. When the cursor is at the EPI6> prompt, pressing the <up arrow> key will bring back

previous commands and allow you to correct errors or change the commands.

Once a line has been recalled, it may be edited. When the <Enter> key is pressed, the command will be executed just as though it had just been typed.

Previous commands are stored in a space called the Command Stack. ANALYSIS normally saves the last 20 commands, but this can be increased by resetting the size of the Command Stack. This is described in the reference chapter for ANALYSIS under the SET command.

Related Topics in Other Chapters

The commands we have described so far will produce results from a questionnaire, but we have not yet discussed how to omit selected records from the ANALYSIS, how to group or recode data items, take action only if certain conditions are met, reorganize and write new files, and many other operations that are essential to complete analysis of a data set. As commands become more complex, it is useful to save them for repeated use in a program file. These and other procedures are discussed in Chapter 11, which should be consulted as soon as you have mastered the elementary operations in ANALYSIS and need to begin serious data analysis.

ANALYSIS also contains features useful in setting up permanent systems such as those used in U.S. state health departments for maintaining communicable disease reports. Two examples of such systems are included on the Epi Info disks and described in Chapters 14 and 26. Commands of interest to those programming such systems are described in Chapters 13, 22, 24, 25, and 33-36.

```
                Questionnaire for Church Supper - Oswego

      Age:▮                        Sex:▮            Time of Supper(24 hour):▮▮▮▮

      Ill? ▮        Onset Date (MM/DD): ▮▮▮▮          Onset Time (24 hour):▮▮▮▮

      Food History (Mark Y or N):
              Baked Ham:▮
              Spinach:▮
              Mashed Potatoes:▮
              Cabbage Salad:▮
              Jello:▮
              Rolls:▮
              Brown Bread:▮
              Milk:▮
              Coffee:▮
              Water:▮
              Cakes:▮
              Ice Cream:     Vanilla: ▮        Chocolate:▮
              Fruit Salad:   ▮

AGE: Integers allowed
F1/F2-Min/Max    F5-Link fields   Legal: F6-Add   Shft-F6-Display   Ctrl-F6-Delete
F3-Repeat        F9-Edit field    Jump:  F7-Add   Shft-F7-Display   Ctrl-F7-Delete
F4-MustEnter     F10-Quit         Codes: F8-Add   Shft-F8-Display   Ctrl-F8-Delete
```

Chapter **10**

The CHECK Program: Optional Error Checking, Coding, and Skip Patterns During Data Entry

<div style="border:1px solid black">

Tutorial

Run the CHECK program from the EPI6 menu.

Enter the file name OSWEGO and respond that you are "Ready" by typing "Y" and pressing <Enter>.

You will see the OSWEGO questionnaire on a screen that looks like the ENTER screen but that has a different series of prompts at the bottom.

Enter "1" in the AGE field. Now press <F1> to make this the minimum age allowed in the field. Note the change in the line just above the function key prompts. Now enter "90" in this field and press <F2> to define the maximum of the range.Move the cursor to the field labelled SEX.

Enter "M" and press <F6> to make this a "legal value." Do the same for "F" and for "U" (Unknown). Press <F4> to make this a "Must Enter" field. The result is that only M, F, and U will be accepted in this field, and a missing value is not allowed, since we have decided to use U for this purpose.

Now let's install a conditional jump or skip pattern so that date and time of onset are skipped if there is no illness. Place the cursor in the ILL field, enter "N", and press <F7>. Following the instructions on the screen move the cursor to BAKED HAM and press <F7> again. The cursor will now jump automatically to BAKED HAM when N is entered for ILL. If Y or blank is entered, the cursor will proceed normally to Date of Onset.

</div>

> **Press <F10> and affirm that you wish to save the results.**
> **The EPI6 menu will appear.**
>
> **Run the ENTER program with the file OSWEGO. Enter a**
> **record to verify that the range, legal values, and jump work**
> **properly. Do not save the results, but press <F10> and "N"**
> **to return to the menu.**
>
> **Other features of CHECK that were not demonstrated**
> **include automatic coding, repeat fields, file indexing, and**
> **changing the names of variables. These are described in this**
> **chapter or in Chapters 22 and 23.**

Overview

Often it is useful to have the computer check for errors during the data entry process, to do automatic coding of entries, and to skip over parts of the questionnaire if certain conditions are met. The CHECK program makes it possible to instruct ENTER to perform such operations automatically. By using CHECK and investing a little extra time in setting up conditions, you can protect your data against many common types of error and also make data entry easier and more automatic. This is particularly helpful if several people or those with minimal computer experience will be entering the data.

CHECK makes a file with a name ending in .CHK. The .CHK file contains instructions for ENTER to restrict the data entered in specified fields. When ENTER is run, it automatically looks for a file with the same name as the .REC file but ending in .CHK.

Using CHECK is optional. The ENTER program will function just as well if you decide not to make a .CHK file.

Running CHECK

Before running CHECK, a .REC file must already exist for the questionnaire you wish to enhance. Use ENTER to make a .REC file from the .QES file if necessary before running CHECK. Do not enter any data items, but exit immediately from ENTER with <F10> to make an empty .REC file; then run CHECK from the main menu. Enter the name of the .REC file in the data file space in CHECK and answer "Y" in the "Ready?" blank.

CHECK presents the questionnaire on the screen with the following function keys indicated on the bottom line:

F1/F2-Min/Max	**F5-Link fields**	**Legal:**	**F6-Add**	**Shift-F6-Display**	**Ctrl-F6-Delete**
F3-Repeat	**F9-Edit field**	**Jump:**	**F7-Add**	**Shift-F7 Display**	**Ctrl-F7-Delete**
F4-MustEnter	**F10-Quit**	**Codes:**	**F8-Add**	**Shift-F8 Display**	**Ctrl-F8-Delete**

Information on the status of functions set up for the field in which the cursor rests will be displayed just above this series of function key prompts.

The function keys act on the field in which the cursor is currently located. Pressing a function key once sets the condition, and in some cases, pressing it a second time will remove the condition. Thus, if you wish to make CITY a Repeat field, as described below, place the cursor in CITY, and press the <F3> key. To remove the Repeat condition, press <F3> again. The condition currently in effect is shown on the line above the function key information. Several different conditions may be entered for one field.

Range Checking (Min/Max) <F1><F2>

When a minimum and maximum are specified, the value entered in a field must be greater than or equal to a specified lower bound and less than or equal to a given upper bound. Any field type may have a range, although the meaning is somewhat different if the field is non-numeric. In this case, the name must lie within the two bounds when sorted in alphabetic order.

To enter a lower or upper range, enter a value in the appropriate field and then press <F1> or <F2>. Additional values can be entered as "Legal", using the <F6> key so that "1-3, 9" can be the final result, for example:

Lower range **1**, Upper range **3**, will accept only values of 1, 2, or 3.
Lower range **Smith**, Upper range **Thompkins**, will accept "Smith",
     "Smithson", and "Thomas", but reject "Thompson".

Legal Values <F6>

Legal values are those that will be accepted as entries in a field. When a list of legal values is specified, an entry must match one of a specified list of values, or it will be rejected. Pressing <Enter> for a missing (blank) value is accepted unless "Must Enter" is also specified as discussed below.

To enter a legal value, type the value in the appropriate field and press <F6>. The value will be added to the list of legal values. To remove an existing

legal value, place the value in the field and then press <Ctrl-F6>, i.e., hold down the <Ctrl> key and press <F6>. Pressing <Shift-F6> will display current legal values for the field if there are more than will fit on the line at the bottom of the screen.

Must Enter <F4>

When Must Enter is specified, missing (blank) values are not allowed. The value entered must be another valid entry.

To specify Must Enter for a field, place the cursor in the proper field and press <F4>. Pressing <F4> again removes the Must-Enter condition.

Repeat <F3>

During entry, the value of the field will be set automatically to the value in the last record entered or displayed. This is useful when a series of questionnaires all contain the same name, for city or state, for example. The few that do not can be changed during entry.

To designate a repeat field, place the cursor in this field and press <F3>. Pressing <F3> again removes the Repeat condition.

Conditional Jumps (Skip Patterns) <F7>

Frequently questionnaires contain sections that are to be skipped, depending on the answer to a prior question. If no illness occurred, for example, the cursor should jump over the section on symptoms and proceed to the next useful item. It is also useful at times to have the cursor jump to an entirely different part of the questionnaire containing special-purpose questions. If "Rabies" is entered for DISEASE, the cursor could jump to a series of questions specific for animal rabies, and then after the last question, jump back to the question following DISEASE.

To set up conditional jumps, enter a value in the appropriate field, e.g., N for ILL, and press <F7>. You will be asked to indicate the destination field for the jump. Put the cursor in this field, and press <F7> again. In the example given if ILL is entered as N, the ENTER program will automatically jump to the designated field, skipping over questions pertaining to illness. Any other answer, such as Y, will cause the cursor to move on to the next field as usual.

You can specify several values with different conditional jumps for a field if desired. To specify a jump regardless of the entry in the field, press <F7> while the field is blank. Such unconditional jumps are useful for changing the order in which fields are entered on a questionnaire.

To jump to one field on "1" and to another on all other entries, you would create one general jump with a blank field and then a different jump for the "1" value. Entry of values for which a jump is not specified will, of course, cause the cursor to go to the next field as usual.

To display the jumps for a field, use <Shift-F7>. A jump may be removed with <Ctrl-F7>.

Linking of Fields and Automatic Coding <F5> and <F8>

If automatic coding is specified, an entry will be compared with a table consisting of pairs of codes, words, or phrases, and converted to the value of the other member of the pair. The result will be placed in the same field, or, if another field has been specified with the link command, in the linked field.

Suppose that you link two fields called DISEASE and DISEASECOD and set up codes for DISEASE. When you enter "Hepatitis B" in a field called DISEASE, the corresponding code (say, 10) would be found automatically and placed in a field called DISEASECOD. If you enter 10 in DISEASECOD, "Hepatitis B" will be placed in DISEASE.

Unless you wish the entries to be recoded to the same field, you must set up the Link between fields before setting up the coding. To do this, place the cursor in the first field and press <F5>. Then following the directions on the screen, move the cursor to the field to be linked and press <F5> again. The fields are now linked and you can proceed with setting up the codes.

Suppose again that there fields for DISEASE and DISEASECOD and these have been linked as described above. When you enter a value in DISEASE, like "Hep A", and then press <F8>, the cursor will jump to DISEASECOD, the linked field. You then enter the code, in this case 10, and press <F8> again. The cursor will jump back to DISEASE, and you are ready to enter as many other codes as you wish, using the same method.

Coding restricts the entries allowed, as though the values had been designated LEGAL. The ENTER program treats upper and lower case alike in coded fields unless the codes are enclosed in quotation marks. During data entry, if

you must enter a value which is not a valid code or legal value, this can be done using the down-arrow key instead of <Enter> to record the value.

You can use the coding facility to accept several synonyms and recode them to a single uniform entry. To do this, OMIT the linking step, thereby linking the field to itself. Then enter the synonyms, such as "Hepatitis B", "Hep B", and "B", giving "HEPATITIS B" as the code for each. This will have the effect of coding all three entries to the same final data item.

Codes can be removed by pressing <Ctrl-F8> with the code to be removed and the cursor in one of the two relevant fields. <Shift-F8> will display all codes for a field.

The Edit Field Command <F9>

The .CHK file contains a series of instructions to the ENTER program. The Edit Field function allows you to see and edit the commands that are inserted automatically in the file by CHECK. It confines the view to commands for the current field--the one containing the cursor.

Try entering one or more CHECK commands and then examine the CHECK file by pressing <F9> with the cursor in the same field. In the example given above for the coding function, if there are many disease codes, you may prefer to use the Edit function to type them directly into the .CHK file.

To do this, press <F9> with the cursor in DISEASE. An editing window appears with the portion of the .CHK file for DISEASE displayed. You will note code pairs in the section called "Codes", terminated by the word END. The first would be "HEP A"   10. You may type in other code pairs, taking care to put quotation marks around any codes that contain spaces. To leave the editing window, press <ESC>.

Editing the file directly is more likely to result in errors than working through the automatic coding process, but if you are careful not to disrupt the structure of the .CHK file, it should be more efficient for large code sets like all the counties in a state or a series of diseases. Several hundred codes can be handled this way, if desired, and thousands can be handled by placing the codes in a .REC file as described in Chapter 35.

Changing Variable Names

The variable names that Epi Info constructs automatically from text preceding the entry blank are not always the best representation of the meaning of the variable. These names can be changed in the data file from the Editor in the CHECK program. To change a variable name, place the cursor in the corresponding field and press <F9>. Then press <F1> for "Change name" as indicated at the bottom of the screen and enter the new name for the field. The variable name will be changed in both the .CHK file and the .REC file.

If you intend to update the questionnaire and use ENTER or MERGE to insert the new changes in the .REC file, however, the data in fields whose names have been changed in CHECK will not be recognized as belonging to the original fields.

Comment Legal Codes

Sometimes it is useful to be able to display codes with comments on their meaning, but without automatic recoding. Using the Edit Field function, a LEGAL statement may be changed to COMMENT LEGAL to display descriptions of legal values. Suppose that a LEGAL statement appears in the .CHK file as:

```
SEX
   Legal
      M
      F
      U
   END
END
```

Pressing <F9> from the main CHECK screen when the cursor is in the SEX field will display these statements.

Add the word COMMENT and the comments or labels, as follows:

```
SEX
   Comment Legal
      M Male
      F Female
      U Unknown
   END
END
```

Now the ENTER program will display the codes in a pop-up window if the <F9> or SEE CODES key is pressed during data entry. This is useful if there are a number of codes and you do not wish to have the longer versions actually entered into the data file, as would occur if you set up codes in CHECK.

The indentation to the left of the items in .CHK files is helpful to understanding the file, but is ignored by the ENTER program and is optional if you wish to enter more items.

Check Features Covered Elsewhere

ENTER can be programmed to do many interesting and complex operations not discussed in this chapter, including mathematical and logical operations with more than one field, changing the colors on the screen, popping up help windows, and calling programs (TSR's) in other languages that act on the contents of fields during data entry. These functions are described in Chapters 22 and 23.

```
F1-Hlp F2-File F3-Epiaid F4-Txt F5-Print F6-Set F7-Find F8-Blk F9-Save F10-Done
read combine
DEFINE FOOD _____
DEFINE ATE _
ERASE COMBINED.REC
Route combined.rec
ATE = Hamburger
FOOD = "HAMBURGER"
WRITE RECFILE FOOD ATE

read combine
DEFINE FOOD _____
DEFINE ATE _
Route combined.rec
ATE = FRENCHFRIE
FOOD = "FRENCHFRIE"
WRITE RECFILE FOOD ATE

read combine
DEFINE FOOD _____
DEFINE ATE _
Route combined.rec
ATE = SALAD
1  C:\EPI6\COMBINE.PGM              EPED  319184     L1   C1  TXTInsInd
```

Chapter **11**
Writing Programs and Preparing Data for ANALYSIS

> **Tutorial**
>
> **Run ANALYSIS from the EPI6 Menu.**
>
> **At the EPI6> prompt, type:**
>
> **EPI6>RUN Tutor2**
>
> **Follow the instructions given to demonstrate some of the major features covered in this chapter.**
>
> **The chapter after this one describes files and exercises for an epidemic of paralytic shellfish poisoning. After you have completed the present chapter, review what you have learned by working with the data from the epidemic in Chapter 12.**

Overview

Chapter 9 presented methods for producing tabular, statistical, and graphic output from data files in the ANALYSIS program. For perfectly "clean" data sets, this would be enough to do a complete analysis. Real-life situations, however, produce "dirty" data, and often the most time-consuming part of the analysis is the "cleaning" of the data set.

This chapter describes methods for viewing and editing data in ANALYSIS, sorting records into numeric or alphabetical order, selectively including or excluding records from analysis, defining and assigning values to new variables, grouping or otherwise recoding variables, performing conditional operations (IF's), managing dates and time intervals, and setting a number of options in ANALYSIS that affect results.

These operations are not individually complex, but a given data set may require many such operations before all fields are in satisfactory condition for analysis. It is useful to incorporate the commands for these operations into a program file, so that the program can be run many times as small additions and changes are made. We will therefore describe how to make program files before we discuss the commands useful for data transformation.

Writing Programs

Programs are files created with a word processor that contain the same ANALYSIS commands you have been using. They can be created in EPED or most other word processors in text-file format (Word Wrap OFF in EPED; NON-document mode if you use Wordstar).

If you create a program called JULY.PGM, the RUN command is used to perform all the commands that are contained in the program:

RUN JULY.PGM

The suffix .PGM is optional, but it is a good idea to use such a marker for program files.

The following program, created on a word processor, reads a file called "DISEASE.REC", which contains data entered in ENTER, using a questionnaire called "DISEASE.QES". It selects records with the month of report equal to 07 (July), prints a list of disease, record number, age, sex, and county in order by disease and record number, and prints a cross tabulation of Disease and County with a separate table for each Age/Sex combination.
(Comments are set off by "*".)

*Listing and Crosstabs of DISEASE.REC
*   Month of July only (These files are not on the Epi Info disks.)

READ DISEASE
SELECT MONTH = 07
SORT DISEASE RECNUM
ROUTE PRINTER
SET PAUSE=OFF
LIST DISEASE RECNUM AGE SEX COUNTY
TABLES DISEASE COUNTY AGE SEX
ROUTE SCREEN
ECHO The tabulation is complete. Returning to DOS.

QUIT
Programs can be as short as one line or quite long. They are most useful
when the operations are complex enough to be a typing burden or when they
will be performed repeatedly. Programs can be set up for people who rarely
use Epi Info or who perform mostly data entry, so that producing a weekly
report requires only the command RUN WEEKLY.PGM.

Programs are also useful for "debugging" statements with complicated logic
(lots of IF's, AND's, and OR's). These can be typed once to form a program
and then revised as the "bugs" are worked out of the statements by repeated
RUNning. It is helpful to set up the READ, SELECT, RECODE, IF, and
LET statements for processing a .REC file, and then to RUN the program
before using the keyboard to enter commands for tables, lists, frequencies, or
means.

Commands that have been entered from the keyboard may be saved to form
programs using the SAVE command as described in the next section.

Recalling, Saving, and Loading Analysis Commands

One way to make a program file is to enter commands interactively at the
prompt. If you wish to save the commands already entered (up to the limit of
the "Command Stack", usually 20), use the SAVE command and the name of
a file, for example:

SAVE OUTBREAK.PGM

All the commands in the current Command Stack will be placed in the file,
which can then be edited in EPED to remove unwanted commands or add
new ones. SAVE is useful if you have entered a number of commands and
decide you would like to make a permanent program file. Use the SAVE
command to make the file and then EPED to edit and perhaps refine the
program so that it can be RUN repeatedly in ANALYSIS.

If you have SAVEd a command stack and would like to recall it from the file,
this can be done using the LOAD command. LOAD <File name> will place
the commands on the Command Stack where they are available for editing,
but, in contrast to RUN, commands will not be executed until you recall them
and press <Enter>.

The number of commands that can be SAVEd can be changed. See the SET
CMDSTACK command in Reference Chapter 36.

Access to DOS Commands Directly from ANALYSIS

Any DOS command, and even other programs, can be run directly from ANALYSIS by pressing the <F9> DOS key. The DOS prompt appears, and you may give any DOS command or run other programs. Typing "EXIT" at the DOS prompt brings back ANALYSIS with all previous processing intact. We will illustrate the usefulness of this command for making a backup copy of a data file.

The UPDATE command, to be introduced next, is the only command in ANALYSIS that changes the data file; the others act only by reading the file. It is important, therefore, to make a backup copy of the data file before using UPDATE. Suppose that the data file is called OSWEGO.REC.

Press <F9> to bring up the DOS prompt. Then type:

C>**COPY OSWEGO.REC OSWEGO1.REC**

The DOS COPY command will make the backup file.

Now type:

C>**EXIT**

ANALYSIS reappears instantly and you are ready to continue processing.

Putting Records in Order -- The SORT Command

Sometimes it is important to have the records in a file arranged in a particular order -- alphabetically by NAME, for example, or numerically by AGE. This is particularly useful for the LIST and WRITE commands. The SORT command will do this for several fields at once, if desired. The records are temporarily rearranged into the new order and may be LISTed or written to another file with the WRITE RECFILE command.

To sort alphabetically by name, use:

SORT NAME

To sort on more than one field, put the field of primary interest
first; if you want to have all females first in the list, followed
by all males, with ages in order within each group, use:

SORT SEX AGE

This command will put the records in order by sex, with the
ages ordered within each sex category.

The order resulting from a SORT disappears with the next
READ or on exiting from ANALYSIS. A permanent file of
the sorted records can be made by using the ROUTE and
WRITE RECFILE commands as described in **Making a New
Epi Info File** in Chapter 13. It is not necessary to SORT
records to do tables or frequencies, since tabulations are kept in
order automatically by the program.

Selecting Records for Processing

ANALYSIS normally processes all records in a file. You can,
however, SELECT records for processing based on specific
criteria. The following commands would select only males
with ages 20 through 49:

SELECT SEX = "M"
SELECT AGE < 50
SELECT AGE > 19

Successive SELECT statements accumulate as though AND had been
inserted, so that the same commands could be combined as:

SELECT (SEX = "M") AND (AGE > 19) AND (AGE < 50)

Parentheses are needed to separate conditions in some cases,
and it is safest to use them routinely, as they do no harm.
SELECT will also accept the shortened form for multiple
conditions:

SELECT AGE <50 and >19

Be careful with the logic; the statements

SELECT GROUP = 1

SELECT GROUP = 2

will not select any records for processing, since no record has the variable called GROUP set to 1 and simultaneously to 2! Probably SELECT (GROUP = 1) OR (GROUP = 2) is what was intended. This will select all the records that are in either group 1 or group 2.

SELECT can be used to omit records also. If you want to eliminate records with GROUP equal to 9, they can be marked as deleted in the ENTER program, or SELECT can be used to omit them from processing with the command:

SELECT GROUP <> 9
(<> means "not equal to")

SELECT conditions remain in effect until you READ another file, leave ANALYSIS, or cancel them by issuing the command:

SELECT

without any criteria. A permanent file containing only the new records can be made by using the ROUTE and WRITE RECFILE commands as described in **Making a New Epi Info File** in Chapter 13. Further details on SELECT are given in the reference section of this chapter.

Examining and Editing Records--BROWSE and UPDATE

There are several methods of reviewing and editing data. The ENTER program may be used to examine and perhaps change each record. In ANALYSIS, frequencies can be performed on each field with FREQ *, and outlying data points can be examined for errors.

ANALYSIS offers another method in which records are displayed in tabular fashion like a spreadsheet. Using the BROWSE command, the records can be examined in this format.

The UPDATE command is provided separately so that you do not inadvertently change records if you mean only to examine or BROWSE. UPDATE allows the columns of data in the file to be changed permanently from ANALYSIS. **Before you use UPDATE, be sure to make a backup copy of the .REC file, since UPDATE causes permanent changes in the data. (See "Access to DOS Commands Directly from ANALYSIS" earlier in this chapter.)**

To use BROWSE, press the <F4> BROWSE key. The records in the current file appear on the screen in spreadsheet format. By moving the cursor to the right or down you can see all the fields and all records. If you prefer the original questionnaire format, press <F4> as indicated at the bottom of the screen for "Full Screen." The questionnaire will appear as if in ENTER, and additional records can be viewed with the <F8> and <F7> keys.

Try experimenting with the other commands listed at the bottom of the screen. Either <Esc> or <F10> will bring back the EPI6> prompt.

UPDATE works in exactly the same way as BROWSE, but you type UPDATE at the EPI6> prompt rather than pressing a function key. The same spreadsheet format appears, but the current field is presented at the top of the page, ready to receive an entry. Some, but not all, of the CHECK functions work in UPDATE.

If you make an entry and then press the <Down arrow> key rather than <Enter>, the cursor will subsequently move down the column rather than across the page when you press <Enter>, much simplifying the editing of a column of data. BROWSE and UPDATE can display selected subsets of the data file. SELECTing is described below. SORTed data appear in the sorted order, a useful feature for eliminating duplicate records in a data file. Note that a record can be deleted just as in ENTER by pressing the <F6> key, and that the entire questionnaire can be displayed as in ENTER if desired.

When you have completed the changes on a record, you will be asked if you want to save the changes. If you respond with "Y," the record will be permanently changed in the data file.

You will find UPDATE a convenient way to examine a column
of entries and correct typing or coding mistakes made on data
entry.

Defining New Variables

For many data "cleaning" operations such as recoding variables, new
variables must be created. The results of recoding or other operations are
placed in the new variables temporarily while screen or printed output is
obtained. These **defined variables** disappear on exit from ANALYSIS, but
can be preserved in a permanent file as described under **Making a New Epi
Info File** in Chapter 13. The commands that create the variables can be
preserved in program files as described at the beginning of this chapter, so
that they can be created again each time the program is run.

There are four data types for defined variables: Text (String), Number, and
European and American Dates. Text variables hold alpha-numeric characters
(letters and/or digits) and would include "Sam", "ID4956", "Y", "(404) 329-
1326", and "No lab. result". Numeric variables handle numbers only and
could be assigned the value 3 or 4567.99, but not 287-4952, since it contains
the non-numeric hyphen character. Dates are in the format mm/dd/yy or
dd/mm/yy.

To define a new variable in ANALYSIS, the same special characters used in
the questionnaire can be used to describe the type and length of the variable.
Thus,

Define NAME _____
DEFINE AGE ###
DEFINE ID #############
DEFINE PRICE ###.##
DEFINE ONSET <mm/dd/yy>
DEFINE ONSET <dd/mm/yy>
DEFINE NAME <A   >

are all allowed. Phone number, <IDNUM>, and <Today> formats are not,
but may be treated as text variables.

Standard defined variables are reset to missing as ANALYSIS reads each
record in the file, such as during a LIST, FREQ, or TABLES command. They
differ from variables in Version 3 of Epi Info, which retain values from
record to record unless reset by a program statement.

CUMULATIVE defined variables are provided for summing or otherwise accumulating values across all the records in a file. They are reset to zero by each READ statement and after every command such as LIST, FREQ, or TABLES that reads the entire file. They are defined by placing the word CUMULATIVE after the definition statement.

GLOBAL defined variables are created by placing the word GLOBAL after the definition. They have two properties that are different from standard defined variables:

1. They retain their definitions and values even when another ANALYSIS program file is run and thus are useful for carrying a message from one program to another. Ordinary defined variables disappear when a new program file is run.

2. Like CUMULATIVE variables, they retain their value when a new record is read.

GLOBAL variables are set initially to "missing," in contrast to the zero or blank values they assume in Version 3. Any desired initial value can be set, however, by indicating the value after the definition.

DEFINE TODATE <dd/mm/yy> GLOBAL "03/03/90"

will create a GLOBAL European date variable TODATE with the initial value of 03/03/90.

In the next two sections, we will see how values can be assigned to defined variables and how they can be assigned recoded values from existing fields.

When Commands Are Processed

In writing programs and using defined variables it is important to understand when commands are processed. Certain commands, such as LIST, FREQ, TABLES, MEANS, MATCH, and the graphing commands cause ANALYSIS to read every record in the current data file and send the results to the screen or other output device. Only records currently SELECTed (and having the delete state determined by the SET PROCESS command) are processed. Records having missing values for the fields being processed are ignored if SET IGNORE is ON.

Some commands, such as LET, IF, RECODE, and WRITE are executed **each time a record is processed.** Thus, in a file of 3,000 records, the command

IF AGE > 20 THEN AGELABEL = "Adult" will be processed 3,000 times, during a TABLES command, each time before the record is tabulated for the table. Hence the table may display 1765 "Adults" just as though the field AGELABEL had been present in the original questionnaire and a clerk had typed "Adult" 1,765 times during entry of the 3,000 records.

Some commands, such as IMMEDIATE, TYPE, ECHO, and DEFINE are executed only once, at the place in the program where they occur. They are usually used for commun icating with the user or the screen and are described more fully in Chapter 13.

A command called PROCESS causes ANALYSIS to read the file and to perform all the calculations that are associated with each record. Thus a variable may be set up to calculate a cumulative sum for a particular field, for example

DEFINE SUM ####.## CUMULATIVE
SUM = SUM + COST
PROCESS
TYPE "@SUM"

PROCESS causes the calculation to occur, and a TYPE command may then be used to display the results once after all the records have been processed.

Assigning a Single Value: the LET Command

The LET command, like the RECODE and IF commands, is performed each time a record is processed. LET assigns a value to a defined variable, possibly as a result of an arithmetic calculation.

The simplest operation is to assign the value of an existing variable to a newly defined variable. If your questionnaire contains the variable AGE, the following commands define a new variable called AGE2 and copy the value of AGE to AGE2 for each record:

DEFINE AGE2 ##
LET AGE2 = AGE

The use of the word "LET" is optional, and the second command may be written as:

AGE2 = AGE

LET may be used with any of the mathematical or logical operations listed under OPERATORS in reference Chapter 36. Thus, to obtain a variable representing weight squared from the existing WEIGHT:

DEFINE WTSQUARED #####
WTSQUARED = WT*WT

Be sure that the variable(s) with existing values are to the right of the equals sign, and that the variable that is to acquire a new value is to the left.

A global variable can be used to count or accumulate values from record to record. If you have not included a variable in a questionnaire for record number, one can be simulated using LET, as follows:

DEFINE RECNO ### CUMULATIVE
LET RECNO = RECNO + 1

At the beginning of processing, RECNO will be assigned the value zero. For each succeeding record, the value will be one higher than the last, so that the effect is that of counting 1,2,3,4, etc., giving sequential record numbers from the beginning of the file.

Missing Values

Missing values are represented in ANALYSIS programs by a period. Thus, to set AGE to the missing value, the command is:

AGE = .

If a new file is written from ANALYSIS, missing values appear in the records as blanks, just as though entered from a questionnaire. The "." convention is for ANALYSIS programming only. The "." symbol does not have special meaning in ENTER, where missing values are entered as blanks by pressing <Enter> in an empty field.

Recoding Data

The data items entered are not always in the form needed for processing. Frequently ages are entered to the nearest year, and tables are desired by 5-year age groups. County names or codes may be given, but tables are needed by district or region, or you may want to substitute descriptive words for

numeric codes. The RECODE command allows you to assign new values to existing data and to place the results in defined variables.

To use RECODE, it is usually best to DEFINE a new variable of the right type to receive the data.

DEFINE AGEGROUP _____
RECODE AGE TO AGEGROUP BY 10

will place ages into 10-year age groups so that you can use AGEGROUP for tables or frequencies.

You can also specify the exact codes desired for each value in a RECODE statement:

DEFINE CTYNAME _____
RECODE CTYCODE TO CTYNAME 1=ROME 2=AUGUSTA

A frequency of CTYNAME will then list the number of records for "ROME" and "AUGUSTA" rather than "1" and "2".   Further details on the use of RECODE are given in the reference section, Chapter 36, under RECODE.

IF Statements

Suppose that food histories have been collected from ill and not ill attendees at a church supper, as in the sample file, OSWEGO.REC. You would like to know how many of the ill and not ill consumed any dairy product at the dinner. The dairy products are MILK, CHOCOLATE, and VANILLA (ice cream). First, define a variable called DAIRY:

DEFINE DAIRY _

and then use an IF statement to give it values:

IF (VANILLA = "Y") or (MILK = "Y") OR (CHOCOLATE = "Y")
THEN DAIRY = "Y" ELSE DAIRY = "N"

Note that the statement is too long for one line, and that the "\ " (backslash) has been used to tell ANALYSIS that the next line is to be considered part of the same statement in a program. Statements can be up to 254 characters long, using this device.

In writing conditional statements in the IF and SELECT commands, it is never *in*correct to surround each complete condition with parentheses and to

separate parts of a statement with spaces, although neither procedure is necessary in all cases. Values of string variables (but not variable names) should be enclosed in quotation marks. "Nested" IF statements, in which one IF results in another IF statement, are not allowed.

The ELSE clause at the end of the IF statement is important to specify what should happen if the conditions are not met. Without the ELSE clause, DAIRY would remain "Y" for each record after the first true DAIRY case was encountered, since the value of defined variables is only set to zero or blank at the beginning of processing, not for each record processed.

Controlling When the IF Command Is Processed

IF statements are executed each time a record is processed. Thus, if you have a file of 3,000 records, the command IF AGE > 100 THEN TYPE "Error: Age > 100" will type the error message on the screen or current output device as many times as there are records with ages of 100 or greater. There are times when you want to perform an operation only once, however, and the IMMEDIATE command, described more fully in the Reference Chapter 36, may be used for this purpose.

Dates

Dates have been a nuisance both before and since Pope Gregory XIII produced the modern calendar in 1582. One must cope with leap years, European and American date order, varying numbers of days in a month, and the fact that leap years are omitted every hundred years unless the year is divisible by 400. Scholars say that the event which marked the beginning of the *Anno Domini* count actually occurred in 4 to 6 B.C. Against this background, Epi Info does its best to make date handling easy.

Date variables are created on the original questionnaire or as defined variables during analysis. The standard variable types for dates within questionnaires are:

<mm/dd>   U.S. Format
<mm/dd/yy>
<dd/mm>   European and Military Format
<dd/mm/yy>
<today>
<today/yy>
<today/yyyy>

To sort records using dates, use the SORT command. Epi Info will take care of sorting so that the year is placed first and the sort will be in correct order.

In ANALYSIS, a new date variable is defined as follows:

DEFINE NAMEOFVAR <Date field type>

Date field type may be any of those given above except those containing "today**."**

Arithmetic operations may be performed with dates; the results will be in days. If RECOVERY and ONSET are date variables in the active data set, then

DEFINE DURATION ####
LET DURATION = RECOVERY - ONSET

will place the duration of illness in days in the variable called DURATION. Dates are actually recorded in the file as text variables and must be surrounded by quotation marks if used as literal parts of commands:

LET DURATION = "12/31/87" - ONSET

If DURATION is in days, the number of weeks or years is obtained by dividing by 7 or 365.25:

DEFINE WEEKS NUMBER
DEFINE YEARS NUMBER
LET WEEKS = DURATION / 7
LET YEARS = DURATION / 365.25

To round off the result to whole weeks, throwing away any extra days, we would use the DIV operator instead of "/":

LET WEEKS = DURATION DIV 7

Programming methods for other operations with dates are given in reference Chapter 36 under LET.

Adjusting the Output of ANALYSIS: The SET Commands

A number of commands help to control optional features of ANALYSIS. They allow you to include titles, comments, and values of variables in the final report, to

control page size and placement of lines, and to produce new files in specified formats. Most of these commands are not needed to use Epi Info for epidemic investigation; they are included for use in permanent systems in which special features are desired, and extra time spent in "programming" is worth the effort.

The SET command gives control of a number of optional features of ANALYSIS. Normally, these options are set as follows:

- Numeric fields are filled with spaces rather than zeroes.
- ANALYSIS does not pause after each page displayed or printed.
- Statistics are produced for each table printed.
- Missing values (blanks) are ignored in tabulations.
- Criteria currently active in the SELECT command are printed.
- Records marked as deleted in ENTER are not processed.
- Vertical lines are not included between cells in tables.
- Line graphics are converted to DASHes in printed output.
- Percentages are not printed in tables.
- Record numbers are included in output of the LIST command.
- Yes/No variables will be converted from "Y" and "N" to "+" and "-" so that their order in tables follows conventional epidemiologic practice.
- European date formats are not used in calculations.
- User confirmation is required before UPDATE saves changes in a record.
- The maximum size of a "cell" in a table is 15 characters. Each cell will be at least as wide as its variable name.
- The work drive for temporary files is C:
- Page size is 17 lines by 78 characters wide for the screen and 66 by 79 for printing.
- Decimal places for tables = 1.
- Confidence limits for statistics = 95%.
- The buffer for reading disk files is 16,384 bytes.
- The command stack will hold up to 20 previous commands.
- The BREAKCMD is set to NEWPAGE.
- When output is ROUTEd to a file, the driver used will be $CGM, producing Computer Graphics Metafiles.
- The printer is set to EPSON connected to parallel port LPT1.

Each of these features may be altered by using the SET command and SET commands can be placed in a file called CONFIG.EPI to make them permanent. To see the current settings, simply type SET and press <Enter>. Further details are given in Reference Chapter 36 under SET.

```
Dataset:  C:\EPI6\EPIDEMIC.REC (126 records)        Free memory: 256K
Criteria: All records selected
                          ┌─────── Output - Screen ───────┐
                          │
=== ┌─────────────────── Variables ───────────────────┐
    │ ┌──────────┐                                      │
    │ │          │  FACE       NAUSEA      COLDDRINK  SHRIMPSOUP │
    │ │ INTERVDATE │ LIPS       FEVER       WATER      LOBSTRSOUP │
    │ NUMINHOUSE  FINGERS     DIZZINESS   BREAD      CLAMSOUP   │
    │ IDNUMBER    TOES        HEADACHE    ARROZ      VEGISOUP   │
    │ AGE         ARMS        WEAKNESS    BEANS      PLA        │
    │ SEX         LEGS        COORDINA    CHEESE     SPOON      │
    │ ILL         EARS        DOBLE       CARNE      ADQUIRIST  │
    │ WEIGHT      TALKING     BLINDNESS   FISH       DEAD       │
    │ EMPLOYED    BREATHING   SORETHROAT  SHRIMP     SYSTEMDATE │
    │ OCCUPATION  WALKING     OTHERSYMP   LOBSTER    SYSTEMTIME │
    │ DATEONSET   GETTINGUP   PRIMERO     CLAMS      │
    │ HOURONSET   ABDPAIN     LUNCHTIME   CHICKEN    │
    │ CLINIC      DIARRHEA    MILK        BROTH      │
    │ CLINICHOUR  VOMITING    BEER        FISHSOUP   │
    └──────────────────────────────────────────────────┘
┌─────────────────────────┌──────────┐────────────────────────┐
│EPI> read epidemic       └──────────┘                         │
│EPI>                                                          │
│                                                              │
   ↑←↓-Move bar  + Tags field  - Untags field  CR-Done  ESC-Abort
```

Chapter **12**

Example of Epidemic Investigation: an Epidemic of Paralytic Shellfish Poisoning

On July 30, 120 cases of sudden neurologic symptoms and 26 deaths occurred in the village of "San Felipe del Mar" in the country "Felicidad." Paralytic shellfish poisoning, caused by toxic dinoflagellates (the "Red Tide"), was suspected. On request of the Felicidad Ministry of Health through the Pan American Health Organization, two epidemiologists from the Centers for Disease Control (CDC) traveled to Felicidad.

The CDC epidemiologists met with Ministry and local officials, and they and Felicidadian epidemiologists carried out several studies. A team of five local health professionals reviewed hospital records and death certificates and interviewed ill persons (cases) and household controls or their relatives. The interviews were conducted in Spanish, using a two-page typewritten questionnaire.

The CDC investigators had brought a portable computer and printer. These were set up in a motel room, and an Epi Info questionnaire was constructed to enter the data. One investigator read the data aloud while the other typed at the keyboard, and 126 cases were entered in about 6 hours. They then produced a listing of all variables in ANALYSIS, which they checked against the questionnaires to find errors. By the third day of investigation they had produced tables for a descriptive study and a case-control study.

The file EPIDEMIC.REC contains the corrected records. The variable names have been changed to English to facilitate understanding by English-speaking readers. You can look at and edit the file using the ENTER program.

We will describe methods for using Epi Info to analyze the file. Although these methods are not exactly the same as those used in the original investigation, they address the same questions posed by the investigators and provide illustrations of Epi Info use. They will be most instructive if you examine the sample files in your computer as you read.

Getting Started

The files from the sample epidemic are EPIDEMIC.REC and CASECON.PGM. If they are not already accessible, install them on your hard or floppy disk from the Epi Info distribution disks, using the INSTALL program on Disk 1.

Descriptive Study

To obtain a description of the cases, run ANALYSIS, beginning with

EPI6>**READ EPIDEMIC**

If you have a printer and would like to have a printed copy of the results, use

EPI6>**ROUTE PRINTER**

To see how many cases there are:

EPI6>**FREQ ILL**

A listing of the cases and controls can be obtained by typing

EPI6>**LIST ***

To turn the printer off, use

EPI6>**ROUTE SCREEN**

The investigators' first step, after "cleaning" the data, was to perform a descriptive study of the cases. The easy way to perform this analysis for all variables is:

EPI6>**SELECT ILL = "Y"** (This selects ill persons only)
EPI6>**FREQ *** (Displays a frequency distribution for each variable)

A useful scheme for analyzing epidemic information is to divide items into **Time, Place, and Person**, or Time, Geography, and Everything Else. **Time** is conveniently shown on an epidemic curve, or graph of the course of the

epidemic, and you may wish to graph several types of frequencies from the analysis of cases, including time of onset for cases who survived and for those who died. Calculating incubation period requires some mathematical gymnastics because the hours are expressed as 1200, 1830, etc., with the first two digits in hours and the third and fourth in minutes, with a base of 60 per hour. The necessary calculations are shown in the program called TIME.PGM, so

EPI6>**RUN TIME.PGM**

and then use INCUBATION to explore hypotheses, first doing

EPI6>**FREQ INCUBATION**

and then

EPI6>**MEANS INCUBATION DEAD**

Would you conclude that there is an association between death and incubation time among the cases?

Use the HISTOGRAM command to make histograms of HOURONSET, CLINICHOUR, and INCUBATION. Cases with negative incubation periods had onset before lunch, which was considered to be an artifact of the two weeks between the episode and the date of interview. To eliminate these cases from the analysis and graph the rest, use the following commands:

EPI6>**SELECT INCUBATION > 0**
EPI6>**HISTOGRAM HOURONSET**
EPI6>**HISTOGRAM CLINICHOUR**
EPI6>**HISTOGRAM INCUBATION**

Place was explored initially, and it was found that all the cases in San Felipe del Mar clustered in a few households in the poorest part of town. Subsequent studies in San Felipe del Mar were limited to these households; therefore, **Place** does not figure prominently in the analysis.

Cases were identified and studied in several other surrounding towns, but for the sake of simplicity, our considerations will be limited to the cases in San Felipe del Mar. The investigators also chose to focus their analysis on these cases while they were still in the field.

Person usually includes all the aspects of demography such as age, sex, and race, as well as exposure variables such as food history, medications, other

illnesses, and behavioral risk factors. Many of these were explored with the help of a case-control study.

Case-Control Study

For the case-control study, a more precise case definition was used, namely "A person in San Felipe del Mar who developed an acute illness on July 30 characterized by headache and at least two of the following sensory symptoms: (numbness of the lips, face, ears, fingers, toes, legs, arms) and at least two of the following motor symptoms (difficulty talking, difficulty walking, difficulty getting up from a lying or sitting position, difficulty breathing, dizziness)."

Programming ANALYSIS to identify the cases and controls by this definition is somewhat complex, and the investigators did not do so in the field. It is sometimes easier to make such decisions manually and enter the result in an extra variable like GROUP that has been included in the questionnaire, or that can be merged with it. As an illustration of techniques, however, we have included a program to identify the cases and controls in the file CASECON.PGM. To see the commands, print or display the file. To use it with the file EPIDEM1.REC, run ANALYSIS, and give the command RUN CASECON.PGM.

CASECON.PGM does the following:

- READs the file EPIDEM1.REC
- DEFINEs variables for GROUP, NEURO symptoms, and MOTOR symptoms

For each record read, ANALYSIS performs the following operations:

- SETS NEURO and MOTOR counts to zero
- Using IF statements, puts the count of NEURO and MOTOR symptoms in these variables
- Sets GROUP to blank
- IF illness, onset on July 30, headache, two or more NEURO symptoms, and two or more MOTOR symptoms are all present, GROUP is set to "CASE"
- For noncases who are not ILL, GROUP is set to "CONTROL"

- SELECT is used to limit processing to CASES and CONTROLS
- SET is used to make ANALYSIS ignore records with missing values
- A page HEADER is set up to provide title information on the printout
- Variables ANYCLAMS, ANYFISH, ANYSHRIMP, and ANYLOBSTER are set up to be "+" if either the food or soup made from the food (or both) were eaten
- A message is displayed on the screen to instruct the user

After RUNning this program, you can give other commands as desired. If you would like a printed copy of the results, type ROUTE PRINTER before running CASECON.PGM. To display the contents of CASECON.PGM, use TYPE CASECON.PGM. You will want to check the results of CASECON.PGM by LISTing the cases and controls:

EPI6>**SORT GROUP IDNUMBER**
EPI6>**LIST (any of the variables of interest)**

How comparable are the two groups? You might compare them with respect to AGE and WEIGHT, two continuous variables, as follows:

EPI6>**MEANS AGE GROUP**
EPI6>**MEANS WEIGHT GROUP**

Now comes the exciting part--comparing the cases and controls with respect to food consumption:

EPI6>**TABLES (food variable) GROUP**

Do as many foods as you like, but don't skip the CLAMS or CLAMSOUP.

It may be that the food histories are less accurate for cases who died. You can repeat the analysis for living cases and controls by using SELECT:

EPI6>**SELECT DEAD = "N"**
EPI6>**HEADER 2 "Living Cases and Controls Only"**

Now repeat any of the commands used above. When you are through, the best way to return to the former selection criteria is to:

EPI6>**RUN CASECON.PGM**

again, and then change the header

EPI6>**HEADER 2 "Cases, Including Those Who Died, and Controls"**

So far you have obtained some very large odds ratios for CLAMS and CLAMSOUP, but if something toxic is contained in the clams, the appropriate tabulation would be for consumption of *any* kind of clams. We have set up a variable in CASECON.PGM called ANYCLAMS, which is "+" if either CLAMS or CLAMSOUP is "Y". The "+" is used instead of "Y" because "+" comes before "-" in the computer's alphabet, and "Y" does not come before "N." This sets up the order of things in the tables so that they follow conventional epidemiologic format with both risk factor and disease presence in the upper left corner and their absence in the lower right corner of the table.

Now make tables of the defined variables:

EPI6>**TABLES ANYCLAMS GROUP**
EPI6>**TABLES ANYLOBSTER GROUP**
etc.

The odds ratio of over 400 obtained for ANYCLAMS is truly astronomical when compared with the odds ratios of 2 or 3 that can be important in chronic disease studies. The relative risks given for each table should be ignored because this is a case-control study, and the size of the exposed population is not known. Can you think of any reason why some of the odds ratios show negative ("protective") associations, having odds ratios less than one?

Despite the striking results obtained so far, we should look for explanations other than the simple hypothesis of a clam-associated toxin or microorganism. Perhaps water consumption is linked either positively or negatively to clam soup consumption, and also to disease incidence. Water consumption would then be a *confounding factor*. We can test for confounding in the analysis by *stratifying* on WATER, that is, by making separate tables for water consumers and nonconsumers and by combining the results in a special way called the Mantel-Haenszel stratified analysis. To do this for CLAMSOUP, type:

EPI6>**TABLES CLAMSOUP GROUP WATER**

Try the same approach with other possible confounders until you are satisfied that the analysis is sound. By stratifying on a variable, we eliminate the effect of confounding by that variable. The approach is only valid, however, if the odds ratios in individual strata are similar in value. If the p-value for Woolf's test is less than 0.05, this provides an indication that a significant difference exists between strata and that the Mantel-Haenszel summary odds ratio should be interpreted cautiously.

Alternatives to Programming

The investigators avoided many of the programming steps we have included in the .PGM files by using the FREQ commands combined with use of the delete feature in ENTER. To perform an analysis, they deleted unwanted records temporarily and then did the analysis. This can be done by using several copies of the data file with different names. They then combined the results from cases and controls, using STATCALC to do the statistics. Although the method was laborious, it was far less so than hand tabulation and also faster and more accurate. Elegance in programming should always take second place to getting accurate results in a field investigation.

General Results

The investigators concluded that consumption of clams and clam soup was responsible for the epidemic and for the 26 deaths. Bioassay and chemical analysis of clams showed high levels of saxitoxin, the chemical basis of the toxicity of shellfish during periods of "red tide."

The investigating team made recommendations 1) for the monitoring of clams for saxitoxin and 2) for health education, since (despite the odds ratios) many residents did not realize the danger of eating clams during periods of "red tide." They also recommended a system for warning citizens during danger periods and provided ideas for further research, including exploration of the relationship of clam beds to agricultural runoff and to a nearby shrimp farm.

A preliminary report--containing key tables, hand-drawn maps, the questionnaire, a summary of methods, results of the descriptive and case-control studies, a summary of cases outside San Felipe del Mar, conclusions, and recommendations--was printed from the portable computer, discussed verbally with CDC headquarters staff, and submitted to the Ministry of Health and to the Pan American Health Organization before the investigators left Felicidad, approximately two weeks after they had arrived. Further analysis was conducted after their return to Atlanta. For this purpose, the EXPORT program was used to produce a SAS file for analysis on the CDC mainframe, where logistic regression and other tests could be performed.

Chapter **13**

More ANALYSIS: Writing New Files; Restructuring Records; Communicating with the Screen and Printer; Summary Records

Making a New Epi Info File

Sometimes it is useful to make a new Epi Info file containing records that have been selected or altered as a result of processing and containing defined variables and other temporary changes. This can be done with the ROUTE and WRITE RECFILE commands. To make a file called CLEANED.REC from an active data set by another name, use the following commands:

ROUTE CLEANED.REC
WRITE RECFILE

The ROUTE command sends output to the new file. It is important to include the .REC part of the file name, since Epi Info data files must have this suffix. WRITE followed by RECFILE creates an Epi Info .REC file format. This command can be followed by a list of field names separated by spaces so that only these fields are included in the new file. Only records currently active (SELECTed) will be included.

Deleted records, those marked as deleted in the ENTER program, will be omitted from the new file if SET PROCESS = UNDELETED or SET PROCESS = BOTH is active, but if the command SET PROCESS = DELETED has been given, only deleted records will be written. The normal condition when ANALYSIS is run is to process only undeleted records.

The ROUTE SCREEN command closes the file and sends output solely to the screen again. Files are also closed when <F10> is pressed to leave ANALYSIS.

Restructuring Records

The WRITE command can be used to change the number or
order of variables in a record, to write defined variables to a
file, or to shorten or lengthen a variable. As described above,
the first step is to ROUTE to an appropriate file. If the file is
to be an Epi Info file, the WRITE command is followed by
RECFILE; otherwise, a plain text or ASCII file will be made.
If the file is a plain text file, it will not be closed
automatically. The command ROUTE SCREEN tells
ANALYSIS that writing to the file is complete and closes the
file.

Suppose that the active file in ANALYSIS contains the fields
DISEASE, STATE, AGE, and SEX. You would like to make
a new file containing these fields in a different order and add
a new field of width 10 that will be filled with 9's to indicate
blank entries. The following commands would be used:

DEFINE MISC _____
MISC = "9999999999"
***In case a file already exists by this name**
ERASE NEW.REC
ROUTE NEW.REC
WRITE RECFILE State:2 Sex:1 Disease:3 Age:2 Misc
***Close the file--optional, since this is done automatically**
ROUTE SCREEN

The file will be an Epi Info file because of the RECFILE command. Note
that other elements are separated by spaces, the widths of fields are
designated by colons and numbers, and the literal "9999999999" must be
written to the .REC file as a variable.

Suppose that you need a customized ASCII file with fields delimited by
spaces for use in a logistic regression program that requires this format. The
following commands would write the file:

ERASE REGRESS.DAT
ROUTE REGRESS.DAT
WRITE DISEASE " " AGE " " SEX " " RACE " "
***Close the file--not optional, since this is not a .REC file**
ROUTE SCREEN

The section on the WRITE command in Chapter 36 also gives methods for specifying the number of decimal places and for including special characters like <Enter> so that a mailing label format can be produced.

Communicating with the Screen or the Printer: ? ?, PICKLIST, TYPE, ECHO, BREAK, IMMEDIATE

In writing programs to be used by others, frequently a question must be asked of the user, and the result is interpreted by the program. If you write a program called WEEKLY that carries out the tasks for processing of weekly surveillance data, you will need to ask the user what week he or she wishes to process or what the beginning and ending dates are for the period to be covered.

To do this, you place a prompt to the user between two question marks:

WEEK = ?What week would you like to process (1-52):?

The prompt will be displayed on the screen, and the user's answer will be placed in the variable WEEK. This device may be used in other commands for which variable values are needed:

LIST ?What fields would you like to list (names, separated by spaces): ?
TABLES ILL ?Name of Exposure Variable: ?

The ? ? command is also useful for inserting pauses in the output, as in:

?Press any key to continue... ?

This can be extremely useful during testing and debugging of a new program, or for writing tutorial programs.

A command called PICKLIST, described in Chapter 36, provides a popup menu on the screen with a number of choices for the user. The word or phrase selected is available for use in the program.

The TYPE command, with a phrase enclosed in quotes, can be used to display phrases on the screen, printer, or output file currently active. The command:

TYPE "\ CWeekly Report for Week #@WEEK"

will, if WEEK is equal to 14, display

```
              Weekly Report for Week 14
```

The "\ C" means "Center the text on the screen."

There is another meaning for the TYPE command; TYPE TEXTFILE.TXT will display the contents of the file TEXTFILE.TXT on the screen or send it to the printer, depending on the last ROUTE command issued. It is useful for displaying the contents of a program file or for including text from previously prepared files in a report.

ECHO is similar to the ECHO command in a DOS batch file. It sends the text that follows the command *only to the screen*. ECHO HELLO will send HELLO to the screen, but TYPE "HELLO" will send the HELLO to the screen, the printer, or a file designated by ROUTE. ECHO is useful when you are writing programs and want to issue instructions or messages to the user on the screen but do not want to include them in the printed or file output.

Several special expressions can be included in TITLE or TYPE commands. If you would like to begin a new line in the middle of a statement, use "\n" to indicate the "newline" character. The newline character has the same effect as typing <Enter>. For example:

TYPE "First line\nSecond line"

will display:

```
 First line
 Second line
```

You can also include the *value* of a variable in the expression by using the @ sign, as in:

TYPE "This page shows the data for @COUNTY county."

The current value of the variable COUNTY will be substituted in the sentence when it appears.

TYPE can be used to send special commands to a printer. When ROUTE
PRINTER is in effect, the command <Esc><n> would be sent to the printer
by:

 TYPE "\27n"

The ASCII code for <Esc>ape, obtained from a DOS manual or other source,
is 27. Most printer manuals contain lists of codes that can be used to change
type styles and a number of other features of printer operation.

BREAK performs a preset command every time the value of a particular field
or group of fields changes during a LIST. Normally the preset command is to
skip to a new page, so that listings can be done with each county, for
example, on a separate page. To do this, you must first SORT the file by
COUNTY, then give the command BREAK COUNTY, and finally LIST the
records. Further details for setting the break command to any Epi Info
command are given under BREAK and SET in reference Chapter 36.

The IMMEDIATE command is followed by another command of the type
that is executed once for each record in the file. It executes the command that
follows a single time rather than repeatedly. IF and assignments (LET) are
the most useful commands to use with IMMEDIATE.
Examples are:

 IMMEDIATE IF DISTRICT = 5 THEN TYPE FIVE.TXT
 IMMEDIATE YR = ?Year to be processed?

Further details on screen interaction are contained in Chapter 36.

Processing Summary Records: OUTPUT, SUMFREQ, and SUMTABLES

Many situations require the use of summary records to represent more than
one person or case. Sometimes it is handy to be able to use results from an
Epi Info table and conduct further processing on the table itself. A local
health department may process individual disease reports monthly and place
the results in a summary file. The summary file could then be sent to the
state or provincial health department, where results from many local
departments would be placed in a single file and processed to obtain the total
for the state or province.

The OUTPUT command, when used with the TABLES or FREQ commands,
makes an Epi Info data record from each cell in the table. When combined

with the ROUTE command, this can create an Epi Info file of records summarizing the cells in the table. Thus, the commands:

 ROUTE SUMMARY.REC
 OUTPUT TABLES DISEASE COUNTY

will create a file called SUMMARY.REC that will contain records summarizing the number of cases of DISEASE by COUNTY. The file will be a standard Epi Info .REC file and will have the following variables:

 DISEASE COUNTY COUNT

The variable COUNT was added by the program and for each record will contain the contents of one cell in the table produced by TABLES DISEASE COUNTY. A record might contain the values "Hepatitis", "Monroe", and 12. In other words, the table of DISEASE by COUNTY has 12 cases in the cell for hepatitis in Monroe county.

In using the example above, the results would be more useful if they contained a variable giving the number of the week being summarized. This could be done by using

 OUTPUT TABLES DISEASE COUNTY WEEKNUM

that would include WEEKNUM in each record.

SUMFREQ and SUMTABLES are used for processing records produced by the OUTPUT command. These two commands produce tables similar to those from the FREQ and TABLES commands but with cell values representing sums rather than counts. FREQ and TABLES count the number of records satisfying specified conditions and put the count in appropriate cells of the tables. SUMFREQ and SUMTABLES *add* the contents of the first variable specified for all combinations of the other variables and place the result in one or more tables. Examples are given in reference Chapter 36.

The OUTPUT and SUMTABLES sequence is a convenient way to maintain intermediate files with summary records, so that large files do not have to be processed "from scratch" each time a further summary is desired. These commands are also essential for systems in which summary data records are submitted, perhaps by county or regional health de part ments, to a central system (at the state level) to be combined into a single file for further processing. Many examples of the use of these commands are found in Chapter 26 in a disease surveillance system that handles summary or aggregate data.

Graphing Summary Data

Sometimes it is important to plot summary data in which the SUM of the values in each record, rather than the count of records having each value, is to be plotted. Summary data records can be created in two ways:

1. By direct entry of summary figures into a questionnaire. One variable might be DISEASE, and summary counts from several countries or time periods might be entered under CASES. Individual records might contain values like:

DISEASE	CASES
HEP A	600
MALARIA	412
HEP A	314

2. By using the OUTPUT TABLES command to produce summary records.A database containing individual case records could be summarized with OUTPUT TABLES DISEASE.

To produce a bar graph of the sum of the values in Variable B and label the graph according to the contents of Variable A, use commands in the form:

BAR <Variable A> /SUM = <Variable B>

In the examples given above, the desired graph would be produced by:

BAR DISEASE / SUM = CASES

The bars would show 914 cases of HEP A and 412 cases of MALARIA.

Controlling Program Flow: RUN, GOTO, and RETURN Commands

Sometimes, when one program has been completed, it is desirable to run another program automatically. This can be done with the RUN command.

It may be useful to skip part of a program if certain conditions are true. This can be done with the GOTO command, which skips to a specified label within the same program.

A sequence of lines in a program may need to be executed more than once. If the sequence is executed several times at one point in the program, it is called a *loop* because the sequence of execution loops back from the last line to the beginning several times. Loops are discussed under the GOTO command in Chapter 36. They employ the IMMEDIATE IF command to specify conditions for looping.

If the same sequence of lines is to be called for execution from several points within a program, these lines can be placed in a *subroutine* to be called with the GOTO command. The RETURN command marks the end of the subroutine and returns execution to the line after the GOTO that called it. Subroutines are described under the RETURN command in Chapter 36.

Combining Data in Multiple Fields in Same File: The COMBINE Command

COMBINE causes the contents of several fields to be treated as though they were in the same field. Suppose, for example, that you have several fields in a case questionnaire as follows:

> Diagnosis1 _____
> Diagnosis2 _____
> Diagnosis3 _____
> Diagnosis4 _____

You would like to know how many times "pneumonia" appears as a diagnosis among all the cases. Since it may appear in any of the four fields, you would have to write an IF statement to count pneumonia if it appears in any of the four fields, or even a program to create new records for each diagnosis rather than for each "case." In database parlance, this is called "normalizing" the database. It may be thought of in simple terms as "putting all the data you want to count or sum into the same field."

COMBINE solves the problem of data spread across multiple fields automatically. When you give the commands

DEFINE DIAGNOSIS _____

COMBINE VALUE = DIAGNOSIS DIAGNOSIS1 TO DIAGNOSIS4

subsequent FREQ and TABLES commands that include DIAGNOSIS will simulate a file with four times as many records as the original file. There will be one case record for each of the four diagnosis fields. In the first, DIAGNOSIS will contain the data for DIAGNOSIS1, in the second it will contain the data for DIAGNOSIS2, etc. We now have a simulated file based on the number of DIAGNOSES (including blanks).

The command FREQ DIAGNOSIS might produce a frequency as follows:

```
Amebiasis           5
Pneumonia          12
Xerophthalmia       1
```

A command such as TABLES AGE DIAGNOSIS will produce a table in the following format, where XX represents an individual cell:

```
              DIAGNOSIS
            XX   XX   XX   XX
    AGE     XX   XX   XX   XX
            XX   XX   XX   XX
            XX   XX   XX   XX
                              TOTAL
```

FIELDS DIAGNOSIS1..DIAGNOSIS4 have been COMBINEd. The actual number of records may differs from that shown in the table or frequency, since each entry in DIAGNOSIS1 to DIAGNOSIS4 contributes a "record" to the tabulation. If SET IGNOREMISSING is OFF, then the number of "records" will be exactly four times the number of actual records; otherwise, missing values will not contribute.

TOTAL will represent the number of diagnoses if SET IGNOREMISSING is ON and four times the number of records (because there are four diagnostic fields) if IGNOREMISSING is OFF.

Only one COMBINE command can be active at one time. Giving a second COMBINE command will turn off any previous COMBINE. The word COMBINE alone without any field names will turn off the COMBINE function.

After giving the COMBINE command WRITE RECFILE will produce a file with a record for each diagnosis (IGNOREMISSING ON) or each field in the COMBINE statement (IGNOREMISSING OFF). In our example, with IGNOREMISSING OFF, four times as many records would be produced as were in the original file. Each would have a copy of all other data fields (such as NAME, SEX, ADDRESS) containing the same data as the original record that contained the fields DIAGNOSIS1 through DIAGNOSIS4.

Sometimes it is important to include the name of the COMBINEd fields in the new "records" so that a table may be constructed.

In a foodborne outbreak, the following fields might appear:

```
            Chicken    <Y>
            Beef       <Y>
            Salad      <Y>
```

You would like to make a graph or do a frequency of exposure to each type of food. Instead of the three separate fields, you would like one field called FOOD containing the names of the foods eaten. Instead of the records:

```
NAME            CHICKEN   BEEF    SALAD
Jones              Y        N       Y
Smith              N        Y       Y
```

You would like to have:

```
 NAME          FOOD       EATEN
Jones         Chicken          Y
Jones         Beef             N
Jones         Salad            Y
Smith         Chicken          N
Smith         Beef             Y
Smith         Salad            Y
```

so that you can do:

TABLES FOOD EATEN

and get the following table:

```
                        EATEN
            FOOD         +    -
              Chicken    1    1   2
              Beef       1    1   2
              Salad      2    0   2
```

```
                         4       2      6
```

If PERCENTS is SET to ON, this table will give attack rates for the various foods. If you wanted to graph the number of exposures to various foods, you could use:

SELECT EATEN = "Y"
BAR FOOD

to place the food names along the horizontal axis and the number of eaters on the vertical axis of a bar graph.

Suppose that a questionnaire representing a district of a country has the following fields:

Number of cases MALARIA #### DIARRHEA #### NEPHRITIS ####

Summary data like this can best be COMBINEd as follows:

DEFINE COUNT #####
DEFINE DISEASE _____
COMBINE NAME=DISEASE VALUE=COUNT MALARIA
DIARRHEA NEPHRITIS

SUMTABLES COUNT DISEASE

This would produce a table of disease counts for the country as a whole.

Related Topics in Other Chapters

A program to print mailing labels is included with Epi Info and may be found under EXAMPLES in the EPI6 menu. Chapter 26 shows how many programs are used in a reportable disease surveillance system for a state health department. A method of using the WRITE command to print mailing labels is given under WRITE in Chapter 36. More details on each command are given in Reference Chapter 36.

EPI Info Cluster Sampling Analysis vers 6.00

trata

↓   ↓

SU   eight

↓   ↓

Value

rosstab
                               Value

↓ —————————

Output options
 (•) Scr en
 ( ) Pr nter   File   ame
 ( ) ile   ——————————

ables

          Cancel   So t

eans
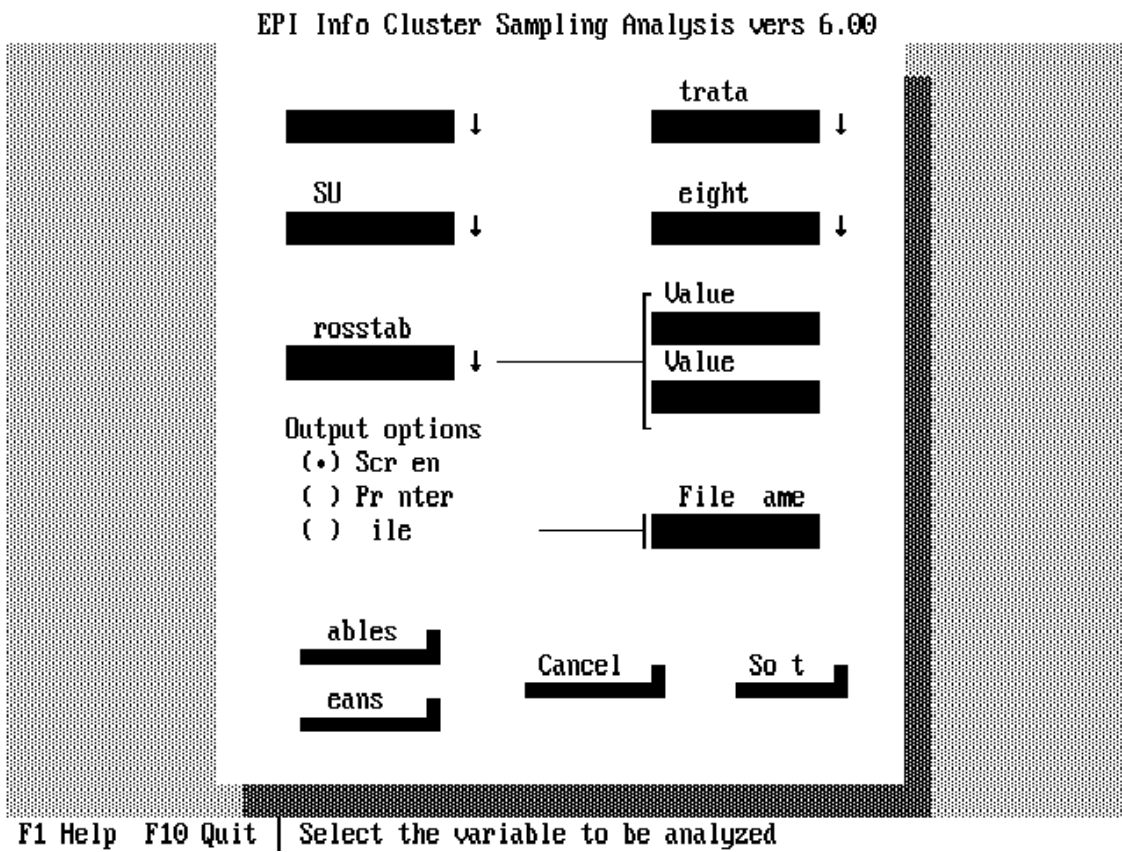
Chapter **14**

CSAMPLE: Analyzing Data from Complex Survey Samples

Concept by Kevin Sullivan, Ph.D.[1]
Statistics by William Kalsbeek, Ph.D.[2]
Statistical Programming by Mario Chen,[2]
Manual by William Kalsbeek and Ralph Frerichs, DVM, DrPH[3]

Tutorial

From the EPI6 menu or from the DOS command line, run the CSAMPLE program. A dialog box will ask for the file name. Type EXAMPLE.REC or press <Enter> to place the cursor in the file list, and then move to EXAMPLE.REC with the <Down arrow> key. Press <Enter>.

The next dialog box asks for the names of variables that describe the study design and the items for analysis.

For the Main variable, enter SMOKE. Leaving all the other entries blank, press <Enter> until the program runs and results appear. Since no details of study design were specified, the analysis is for a simple random sample, and the results are the same as would be obtained by FREQ SMOKE in the ANALYSIS program.

---

[1]Emory University School of Public Health

[2]University of North Carolina at Chapel Hill

[3]Department of Epidemiology, University of California, Los Angeles (UCLA)

Now press <Esc> to return to the Variables dialog. Leave SMOKE as the Outcome variable, but this time enter STRATA for the Strata variable, PSUID for the PSU (Primary Sampling Unit) variable, and SAMPW for the Weight variable. These variables describe a study with several different strata (independently sampled areas), cluster sampling, and weighting of individual records to adjust for survey design and/or response rates, etc.

Now run the program, and note that the results are slightly different from those for simple random sampling. Return to the Variables dialog by pressing <Esc>. Leave all the variable names as before, but this time add a second dimension to the table by entering RACE in the Crosstab field. If you wish to calculate the difference between two values of RACE, enter "1" as VALUE 1 and "2" as VALUE 2; otherwise, leave the VALUE blanks empty to use all the values of RACE. Note that the tabulation now produces a 2 by 2 or 2 by n table. Since the sample was not designed to support two-way analysis, the confidence limits may be somewhat wide, but CSAMPLE will easily perform this kind of post-study, cross-tabular analysis.

SMOKE is a categorical (discreet or coded) variable. Try using NUMCIGAR, a continuous variable, as the Outcome variable. If NUMCIGAR is a real number (decimal) variable, CSAMPLE will automatically switch from TABLES (on the lower left) to MEANS; if not, you must indicate that a MEANS is appropriate by choosing MEANS. Some variables such as AGE may be analyzed either way, although AGE would usually be grouped before using the TABLES approach.

Try sending the output from CSAMPLE to the printer, or to a file, by indicating FILE and giving a file name before choosing TABLES or MEANS.

To leave the program, press <F10>.

You have now seen all the features of CSAMPLE. To appreciate how powerful this deceptively simple program can be, read this chapter to see how many and complex study designs can be represented by variables for PSU, Stratum, and Weights. They are the key to analysis of data from complex study designs.

Overview

The ANALYSIS program in Epi Info performs statistical calculations that assume the data come from simple random (or unbiased systematic) samples. In many survey applications more complicated sampling strategies are used. These may involve sampling features like stratification, cluster sampling, and the use of unequal sampling fractions. Surveys that include some form of complex sampling include the coverage surveys of the WHO Expanded Program on Immunization (EPI) (Lemeshow and Robinson, 1985) and CDC's Behavioral Risk Factor Surveillance System (Marks et al., 1985).

CSAMPLE computes proportions or means with standard errors and confidence limits for studies in which the data did not come from a simple random sample. If tables with two dimensions are requested, the odds ratio and risk ratio, and risk difference are also calculated.

Data from complex sample designs should be analyzed with methods that account for the sampling design. In the past, easy-to-use programs were not available for analysis of such data. CSAMPLE provides these facilities, and, with an understanding of sampling design and analysis, can form the basis of a complete survey system.

Sampling Concepts and Terminology

Calculations in CSAMPLE assume that the individual records are members of a sample in which random (or complete or unbiased systematic) sampling has been used in some part of the design. There are three other basic features of sample design that might be used and that must be accounted for in analysis: *cluster sampling, stratification,* and *unequal sampling rates*. When none of these features is used, methods for simple random samples (FREQ, TABLES, and MEANS in ANALYSIS) are appropriate.

Before discussing the three complicating features of sampling designs more fully, it is necessary to define a few basic concepts related to samples and sampling. A sample is chosen to represent a larger universe called the sample's *target population*. This population consists of individuals called *members* who are the object of study. The data items gathered for the study describe these population members, although data gathering may not involve them directly (e.g., when data is abstracted from medical records).

A sample of population members is chosen from one or more lists called the *sampling frame* whose individual entries are called *sampling units*. In list

samples (e.g., a sample chosen for a physician survey from a list of physicians), the population members and sampling units are the same, but in other samples this is not necessarily the case (a list of households for a survey of individuals). Simple random sampling is a form of list sampling in which selections are made at random and with equal probability from a complete population frame. Simple random selection can be done *with replacement,* meaning that a member can be chosen again once it is selected the first time, or it can be done *without replacement,* in which case a selected member is not allowed to be chosen again. The calculations in CSAMPLE assume that random sampling has been done *with replacement.* If the sample is a small fraction of the population, the assumption of sampling with replacement may be made for practical purposes, as described below, even if this was not the original sampling method.

Stratification

Stratification is a common feature in sampling designs. In sampling terminology, stratification means that the frame is subdivided into mutually exclusive and complete (exhaustive) groups called strata and that samples are chosen separately from each stratum. This use of the word "stratum" should be clearly distinguished from "stratification" during data analysis--the process that occurs when you add a third dimension to a TABLES command in ANALYSIS and a separate table are made for each stratum. In sampling the strata are determined prior to data collection; in epidemiology, data are generally stratified after data collection.

Sample stratification is commonly used in list sampling, with simple random sampling being the selection method within each stratum. The result is what is called a stratified simple random sample. Stratification is also often used in conjunction with cluster sampling. In general stratification tends to reduce the variance (narrow the confidence limits), at least partially offsetting the opposite effect of cluster sampling. While, in principle, stratification can be used in each stage of a multi-stage cluster sample, it is most commonly used for the first stage of selection. Stratification for choosing Primary Sampling Units (PSUs) is called primary stratification, and is described below.

Cluster Sampling

A sampling design is said to involve *cluster sampling* if at some point in the selection process the sampling units consist of one or more mutually exclusive groups, called clusters. The clusters used for survey sampling are typically

spatial (e.g., a sample of residential households that is obtained by selecting counties or villages), organizational (e.g., a sample of students that is identified by sampling schools), or temporal (e.g., a sample of patients visiting a health clinic chosen by sampling days the clinic is open).

The Expanded Program in Immunization (EPI) coverage surveys are cluster surveys. All villages and cities (i.e., clusters) are listed, and then a sample of villages and cities is selected for the survey.

The clusters in real populations rarely have the same number of members and may vary greatly in size. Cluster sampling is often done in more than one step or stage of selection. This type of design produces what is called a multi-stage cluster sample. To do so requires the existence of a hierarchical configuration of clusters, so that the clusters at any given stage consist of members or clusters in the subsequent stage(s). For example, a three-stage sample of households in the U.S. might be chosen by designating counties to be the sampling units in the first stage (called primary sampling units, or PSU's), by assigning block groups to be the second stage sampling units for sampling within each PSU, and by designating households to be chosen separately within selected block groups as the third stage.

Cluster sampling usually increases the variance (widens confidence intervals) of survey estimates. This happens because members of the same cluster tend to be more alike than the population as a whole. Members of a sample from the same cluster therefore tend to provide less information about the population than do members from different clusters. This reduction in information from a clustered sample translates into estimates that are likely to be less precise than estimates obtained using simple random sampling.

Primary sampling units (PSU's) are the units chosen from a list in the first (upper) stage of sampling that involves choosing clusters. For example:

Stage	List Used	Sampling Method
One	Regions	Stratification
Two	**States**	**Random within region**
Three	Counties	Random within state
Four	Census Tract	Random within county
Five	Block	Random within census tract
Six	Household	Random within block
Seven	Person	Random within household

Stage two contains the Primary Sampling Units--the "clusters" at the first stage where clusters are randomly chosen. Each record must contain an identifier for the state from which it came. The file may contain identifiers for each of

the other levels as well, but those below stage two will not be used by CSAMPLE. They may be necessary for the calculation of weights prior to running CSAMPLE, however.

Unequal Selection Probabilities

The third feature that complicates a sampling design is having unequal selection probabilities for population members. This happens when the ratio of sample to population size differs for different parts of the sample. In stratified sampling this can happen if the sample sizes are not proportional to the population of the strata. Unequal selection probabilities may be found in cluster samples in a variety of ways (e.g., sampling clusters with unequal probabilities in a one-stage sample, using simple random samples of the same size in both stages of a two-stage design where the clusters vary in size, etc). They also occur during the course of a survey through differing response rates in different areas, and other factors that may be ascertained through the survey itself, such as response rates or household size.

Unequal selection probabilities are accounted for in the analysis of data by computing sample weights for each member of the sample (i.e., record of the data set). Producing these weights is usually done just prior to analysis. Determining selection probabilities requires that good records be kept for each selection step of the sampling process so that the selection sequence can be explicitly recreated. For multi-stage cluster samples this means knowing the first stage selection probability for the PSU of which the member is a part, the selection probability for the second stage cluster of which the member is a part, and so on up to the selection probability of the member in the final stage cluster. For designs with stratified sampling this implies having separate information for sampling in each stratum (e.g., sampling rates for stratified simple random sampling).

A sample weight for a sample member in its most basic form is simply 1 divided by the member's selection probability. More intuitively, it is the number of population members whom this member represents. The sample weights found in survey data sets are rarely seen in this form, however. They are often adjusted (i.e., multiplied by some appropriate adjustment factor) to compensate for such things as imbalance in the sample due to failure of the frame to fully cover the population (i.e., undercoverage), failure to secure participation from all sample members (i.e., nonresponse), and departures from the demographic composition of the population due to randomized

sampling. Users who are interested in more details about weights and adjustments are referred to Lessler and Kalsbeek (1992, Section 8.1). The weights may also be normalized so that they sum to a designated value (e.g., the total sample size), although this is not required by CSAMPLE.

CSAMPLE's Calculations

The variance calculations in CSAMPLE use the Taylor Linearized Deviation approach. This approach approximates the variance by simplifying the complicated form of the difference between the estimate and the population value being estimated. The version that we use enables you to account for the design, regardless of how complex it is, by telling the program three things about each sample member (i.e., data record):

       1) The *Primary Sampling Unit (PSU)* from which it came,

       2) The primary *stratum* from which its PSU was chosen, and

       3) Its sample *weight*.

To do so requires that either of the following hold:

       1) PSUs were selected with replacement, or

       2) First stage sampling rates for without-replacement PSU selection were small (less than 5 percent).

If sampling rates are greater than 5 percent for without-replacement PSU sampling, the variance estimates will tend to overstate the actual variance. The amount of the overstatement will be directly proportional to the size of the sampling rate. Note that overstated variances will generally produce conservative statistical results (e.g., confidence intervals that indicate less than actual precision for survey estimates; tests of hypothesis that return larger than actual significance levels). This is not necessarily disadvantageous but it should be kept in mind. The bottom line is that sampling *without* replacement, while it will not produce misleading results, may not have quite as much statistical power as sampling *with* replacement, if the sample is a sizable fraction of the population.

163

An Example of Stratified Simple Random Sampling

To illustrate these design features and some of the concepts that were previously defined, consider a telephone survey of adults living in an area defined by four unequal sized counties. If the sample of member adults from this target population is chosen by separately selecting a simple random sample of a fixed number (say, 200) from voter registration lists in each of the four counties, the resulting sample would be a stratified simple random sample, with *county* as the stratification variable and unequal sampling rates (i.e., selection probabilities) among strata.

This design would not involve cluster sampling, although technically one could consider each adult to be a one-member PSU. The names or codes of counties would be included in the analysis as the STRATUM variable.

Stage	List Used	Sampling Method
One	County	Stratification
Two	Voters	Random

An Example of Stratified Multi-stage Cluster Sampling

A random digit telephone sample might be selected as follows. Clusters of telephone numbers are defined, with each cluster consisting of numbers with the same first eight digits of a 10-digit telephone number. Separately for each county, a with-replacement sample of clusters is randomly chosen with probabilities that are proportional to the size (PPS) of (i.e., number of residential telephone numbers in) the cluster. Next, a random sample of three participating households is selected within each selected cluster. Finally, an interview is completed with one adult who is chosen at random within each participating households. This design produces a stratified three-stage sample, with clusters of telephone numbers as PSUs, primary stratification by county, residential telephone numbers as second stage clusters, and member adults as sampling units in the third stage. While this design would produce an equal-probability sample of residential telephone numbers, the final sample of adults would have unequal selection probabilities (and therefore require sample weights) since the number of adults and the number of telephone numbers that can be dialed to reach the households varies among households.

Stage	List Used	Sampling Method
One	8-digit clusters by county	Random PPS within 8-digit clusters Stratified by county
Two	Clusters from stage one	3-random households per cluster
Three	Households from stage two	One random adult per participating household

A record would be created for each adult interviewed in stage three. Primary sampling units in this case are identified as 8-digit clusters. Stratification is by COUNTY.

WEIGHTS would be determined by the number of adults in each household interviewed, the simplest method being to use the number of adults in the household as the weight. If one household contained 100 adults, the interview in that household would be weighted as 100 in comparison with an interview with a single-adult household (weight 1). If participation rates differed among counties, additional weighting would be done to compensate for this--a county with 90% participation would have a weight of 100/90 in comparison with a county with 70% participation having a weight of 100/70. A household with two telephones has twice the chance of being selected and would have a weight of 0.5 in comparison with a one-telephone household.

Design Effect

A useful design-related measure for surveys is the so-called "design effect" (Kish, 1965), the ratio of the variance of the estimate under the actual design used to produce the estimate to the variance of the estimate assuming the same data to have come from a simple random sample (this definition may differ from that used by other computer programs for complex survey analysis). The design effect reflects the estimated variance of the survey data relative to that of a simple random sample. Design effect for multi-stage cluster samples will usually exceed 1, sometimes substantially, while for stratified simple random samples and other list samples design effect will be near or slightly less than 1.

Generally stratification tends to reduce the design effect and cluster sampling to increase it. Widely variable sample weights tend to increase design effect.

Preparing the Data

CSAMPLE has several important data requirements. The data must be organized with records in the Epi Info file representing population members. When the sample design features primary stratification, the *stratum identifier* must be included in each record. If cluster sampling is used, a *PSU identifier* must be included. Finally, if unequal selection probability (samples not proportional to population) has occurred, *sample weights* should be included in each record. (The program presumes that the weight for each member has been properly computed. Consult with a survey statistician if you are unsure about this.)

None of these variables is strictly required. If, for example, you have used multi-stage cluster sampling, where no primary stratification was used, only the PSU identifier and the weight variable are needed. Moreover, if the weight variable was not computed since selection probabilities are the same, the program will still properly account for the design but will assume equal probabilities of selection.

Sample Design and Data Fields Needed for CSAMPLE
(In addition to the Variable Being Analyzed)

Information needed if the design is:	Stratum Identifier	PSU Identifier	Sample Weight
Simple random sampling	No	No	No
Stratified simple random sample	Yes	No	Optional
Unstratified cluster sample (one or more stages)	No	Yes	Optional
Stratified cluster sample (one or more stage)	Yes	Yes	Optional

CSAMPLE requires the data to be sorted by PSU within stratum, and provides facilities for sorting.

A portion of a properly sorted data set would look like the following:

Primary Stratum	PSU	Weight	Outcome Variable
1	1	35.67	45
1	1	20.53	42

```
        1        2              16.02        71
        1        2              21.76        64
        2        1              47.87        34
        2        2              37.69        41
        2        2              21.12        38
        2        3              41.33        21
        2        3              28.12        19
```

Running CSAMPLE

CSAMPLE may be run from the EPI6 menu or directly from the DOS prompt. The controls are operated either with the mouse or with the keyboard. To move from one entry blank or control to another, use the mouse or press the <Tab> key. <Shift-Tab> moves to the previous blank or control button. To see the available variables on the main screen, press the <Down arrow> with the cursor in the entry blank or click on the arrow to the right of the blank. After making an entry, press <Enter> as elsewhere in Epi Info.

In the following examples, the entries to be made in the blanks will be indicated. It is recommended that you do the tutorial at the beginning of this chapter to become familiar with running CSAMPLE.

Example: A World Health Organization Cluster Survey of Vaccination Status

In the EPI1.REC data set, there are 210 children, with 7 in each of 30 different clusters (see Table 1). Information was gathered in accord with the Expanded Program on Immunization (EPI) methodology as described by Lemeshow and Robinson (1985). Each subject has two measured variables: PRENATAL and VAC. The former assesses whether the mother received prenatal care (Y=1, N=2), while the later indicates if the child has received a complete set of vaccinations (Y=1, N=2). Accompanying the two variables is a third variable CLUSTER, that identifies the cluster where the child resided. The clusters are numbered from 1 to 30.

Table 1. Data in EPI1.REC

CLUSTER	PRENATAL	VAC	CLUSTER	PRENATAL	VAC	CLUSTER	PRENATAL	VAC
	Column One			Column Two			Column Three	
1	2	2	11	1	1	21	2	1
1	2	1	11	2	2	21	2	1
1	1	1	11	2	1	21	2	1
1	2	2	11	2	2	21	1	1
1	2	2	11	2	1	21	1	2
1	1	1	11	1	1	21	2	2
1	2	2	11	2	1	21	2	2
2	2	1	12	1	1	22	2	2

167

```
        2       1       1       12      2       1       22      2       2
        2       1       1       12      1       1       22      2       2
        2       2       1       12      2       1       22      2       2
        2       1       1       12      2       1       22      2       2
        2       1       1       12      2       1       22      2       2
        2       2       1       12      1       2       22      1       1
        3       2       2       13      2       1       23      2       1
        3       1       1       13      2       1       23      2       2
        3       2       2       13      2       2       23      2       2
        3       2       2       13      2       2       23      1       1
        3       2       1       13      2       1       23      2       1
        3       1       1       13      2       1       23      2       1
        3       1       1       13      2       1       23      2       2
        4       2       1       14      1       1       24      2       1
        4       1       1       14      2       1       24      2       2
        4       2       2       14      1       1       24      2       1
        4       2       2       14      1       1       24      2       2
        4       1       1       14      1       2       24      2       1
        4       1       1       14      2       2       24      1       1
        4       2       1       14      1       2       24      2       1
        5       1       1       15      1       1       25      2       1
        5       1       1       15      1       1       25      2       2
        5       1       1       15      1       1       25      2       2
        5       1       1       15      1       1       25      1       1
        5       1       1       15      1       1       25      2       1
        5       2       2       15      1       1       25      2       1
        5       2       2       15      1       1       25      1       1
        6       1       1       16      2       1       26      2       1
        6       1       1       16      2       1       26      2       1
        6       2       1       16      2       2       26      2       1
        6       2       1       16      2       1       26      2       1
        6       2       1       16      2       1       26      2       1
        6       2       1       16      1       1       26      1       1
        6       2       1       16      2       1       26      2       1
        7       1       1       17      1       1       27      2       2
        7       2       2       17      1       2       27      2       2
        7       1       1       17      1       1       27      2       1
        7       1       1       17      1       1       27      2       1
        7       2       2       17      1       1       27      2       1
        7       1       2       17      1       1       27      2       1
        7       1       1       17      1       1       27      2       1
        8       1       1       18      1       2       28      1       1
        8       1       2       18      1       1       28      1       1
        8       1       1       18      2       1       28      2       1
        8       2       1       18      2       2       28      2       1
        8       1       1       18      2       2       28      2       1
        8       1       1       18      2       2       28      2       1
        8       1       1       18      2       2       28      2       1
        9       1       2       19      1       1       29      2       1
        9       1       1       19      1       1       29      1       1
        9       2       1       19      1       1       29      2       1
        9       1       1       19      1       1       29      2       1
        9       1       1       19      1       1       29      1       1
        9       1       1       19      1       1       29      1       1
        9       1       1       19      1       1       29      2       1
       10       2       2       20      2       1       30      2       1
etc.
```

The data will be analyzed in three ways: 1) In ANALYSIS, (which assumes random sampling) 2) in CSAMPLE, specifying only the Main variable, which indicates a random sample analysis, and 3) In CSAMPLE specifying the CLUSTER identifier as the PSUID, taking into account the cluster design.

Using the ANALYSIS program, the following output is obtained after giving the commands:

Epi> **READ EPI1**
Epi> **FREQ VAC**

168

```
VAC   | Freq  Percent   Cum.
------+----------------------
1     |  155   73.8%    73.8%
2     |   55   26.2%   100.0%
------+----------------------
Total |  210  100.0%
```
Conclusion: 73.8% of the children are vaccinated

Using CSAMPLE to analyze the file as though it were a random sample, the items in the input dialog box are:

Data File	**EPI1.REC**
Main Variable:	**VAC**
Strata:	<None>
PSU:	<None>
Weight	<None>
Crosstab:	<None>
Output:	<As desired>
Analysis button:	**TABLES**

```
            CSAMPLE COMPLEX SAMPLE DESIGN ANALYSIS
Analysis of VAC
|VAC          |
|             |Total        |

| 1           |             |
|  Obs        |         155 |
|  Percent  V |      73.810 |
|  Percent  H |     100.000 |
|  SE %       |       3.041 |
|  Lower CL % |      67.849 |
|  Upper CL % |      79.770 |

| 2           |             |
|  Obs        |          55 |
|  Percent  V |      26.190 |
|  Percent  H |     100.000 |
|  SE %       |       3.041 |
|  Lower CL % |      20.230 |
|  Upper CL % |      32.151 |

|Total Obs    |         210 |

|Design eff.  |       1.005 |

   Sample Design Included:
   -----------------------
   Sampling Weights--None
   PSU--None
   Stratification--None

0 records with missing values
```

Note the V and H indicators on the line after the word "Percent." V means "vertical" percent and H the "horizontal" percent. The bottom of the V points to its denominator.

Conclusion: 73.8% of the children are vaccintated, with 95% confidence limits from 67.8 to 79.8

The design effect differs from 1.0 only by rounding effects.

To account for the cluster sampling in CSAMPLE, the name of the field containing identifiers for the primary sampling unit, in this case, the cluster number, must be given:

Data File	**EPI1.REC**
Main Variable:	**VAC**
Strata:	\<None\>
PSU:	**CLUSTER**
Weight	\<None\>
Crosstab:	\<None\>
Output to:	\<As desired\>
Analysis button:	**TABLES**

```
   COMPLEX SAMPLE DESIGN ANALYSIS
 Analysis of VAC
VAC         |
|           |Total      |

|1          |           |
| Obs       |        155|
| Percent  v|     73.810|
| SE %      |      4.599|
| Lower CL %|     64.795|
| Upper CL %|     82.824|

|2          |           |
| Obs       |         55|
| Percent  v|     26.190|
| SE %      |      4.599|
| Lower CL %|     17.176|
| Upper CL %|     35.205|

|Total Obs  |        210|

|Design eff.|      2.298|

  Sample Design Included:
  -----------------------
  Sampling Weights--None
  Primary Sampling Units from CLUSTER
  Stratification--None

  0 records with missing values
```

Conclusion: 73.8% of the children are vaccinated, with confidence limits from 64.8 to 82.8, several percentage points wider than the result obtained by assuming random sampling above.

The variance is estimated to be 2.298 times that for a simple random sample

Example: A World Health Organization Cluster Survey of Vaccination Status Crosstabulated With Prenatal Care

Whether or not the mother received prenatal care was also recorded in this survey. The authors were interested in the hypothesis that receiving prenatal care is associated with a higher rate of vaccination among children. Using the ANALYSIS program and (incorrectly) treating the sample as a simple random sample, the commands would be:

Epi> **READ EPI1**
Epi> **TABLES PRENATAL VAC**

```
                            VAC
PRENATAL    |     1      2 | Total
-----------+-------------+------
         1 |    78      9 |    87
         2 |    77     46 |   123
-----------+-------------+------
     Total |   155     55 |   210


              Single Table Analysis

Odds ratio                                         5.18
Maximum likelihood estimate of OR (MLE)            5.14
Exact 95% confidence limits for MLE      2.29 < OR <  12.79
Exact 95% Mid-P limits for MLE (2)       2.41 < OR <  11.82
Probability of MLE >= 5.14 if population OR=1.0 = 0.00000547

RISK RATIO(RR)(Outcome:VAC=1; Exposure:PRENATAL=1)  1.43
95% confidence limits for RR               1.23 < RR < 1.67
```

Conclusion: Children whose mothers had prenatal care are 1.43 times more likely to be vaccinated than children whose mothers did not have prenatal care, with (incorrect) confidence limits of 1.23 and 1.67 for the ratio.

To perform the necessary cross tabulation in CSAMPLE, using the cluster-sample design in the calculations, the following choices are made in CSAMPLE:

Data File	**EPI1.REC**
Main Variable:	**VAC**
Strata:	<None>
PSU:	**CLUSTER**
Weight	<None>
Crosstab:	**PRENATAL** limited to values 1 and 2
Output to:	<As desired>

Analysis button: **TABLES**

```
                          CSAMPLE COMPLEX SAMPLE DESIGN ANALYSIS

 Analysis of VAC by PRENATAL
 Comparison between PRENATAL 1 and 2

|PRENATAL     |VAC
|             |1              |2             |Total      |

|1            |               |              |           |
| Obs         |           78  |           9  |        87 |
| Percent   V |       50.323  |      16.364  |    41.429 |
| Percent   H |       89.655  |      10.345  |   100.000 |
| SE %        |        3.017  |       3.017  |           |
| Lower CL %  |       82.824  |       4.432  |           |
| Upper CL %  |       95.568  |      16.258  |           |
|Design eff.  |       0.8538  |      0.8538  |           |

|2            |               |              |           |
| Obs         |           77  |          46  |       123 |
| Percent   V |       49.677  |      83.636  |    58.571 |
| Percent   H |       62.602  |      37.398  |   100.000 |
| SE %        |        6.837  |       6.837  |           |
| Lower CL %  |       49.201  |      23.998  |           |
| Upper CL %  |       76.002  |      50.799  |           |
|Design eff.  |        2.456  |       2.456  |           |

|Total        |               |              |           |
| Obs         |          155  |          55  |       210 |
| Percent   V |       100.00  |      100.00  |           |
| Percent   H |        73.81  |       26.19  |   100.000 |
| SE %        |        4.599  |       4.599  |           |
| Lower CL %  |       64.795  |      17.176  |           |
| Upper CL %  |       83.804  |      35.204  |           |
|Design eff.  |        2.298  |       2.298  |           |


 CSAMPLE COMPLEX SAMPLE DESIGN ANALYSIS OF 2 X 2 TABLE


 Odds Ratio        (OR)          5.177
 95% Conf. Limits          (  2.29,  11.69 )

 Risk Ratio        (RR)          1.432
 RR=(Risk of VAC=1 if PRENATAL=1)/(Risk of VAC=1 if PRENATAL=2)

 Risk Difference   (RD)          27.1%
 95% Conf. Limits          ..(  13, 41 )
 RD=(Risk of VAC=1 if PRENATAL=1)-(Risk of VAC=1 if PRENATAL=2)

 Sample Design Included:
 -----------------------
 Sampling Weights--None
 Primary Sampling Units from CLUSTER
 Stratification--None

 0 records with missing values
```

Conclusion: Children whose mothers had prenatal care are 1.43 times more likely to be vaccinated than children whose mothers did not have prenatal care. The 95%

confidence limits do not include 1.0, indicating that the association is statistically significant.

Example: Combining Several WHO Cluster Surveys

The EPI10.REC file is similar but more involved (see Table 2). Here there are 10 different EPI surveys. The subjects all have the same three measured variables included in EPI1.REC: PRENATAL, VAC and CLUSTER. The record for each person contains a numbers that identify the survey (LOCATION) and the population in the area where the survey took place (POPW). There are a total of 2,125 records in EPI10.REC, far too many to list. Therefore Table 2 contains an abbreviated list. The immunization levels varied in the different surveys, from 31.1% to 84.8%.

```
Table 2. Abbreviated data in EPI10.REC
_____
LOCATION   POPW    CLUSTER   PRENATAL    VAC
_____

   1      43.87       1         2         1
   1      43.87       1         2         2
   .        .         .         .         .
   1      43.87      30         2         1
   1      43.87      30         2         1
   2     153.42       1         1         2        _____
   2     153.42       1         1         2        _                  _
   .        .         .         .         .        _ Note: the        _
   2     153.42      30         1         1        _ entire file has  _
   2     153.42      30         1         2        _ 2,125 records.    _
   3      66.65       1         1         1        _ This is an        _
   3      66.65       1         2         2        _ abbreviated       _
   .        .         .         .         .        _ file to show      _
   3      66.65      30         1         2        _ the contents      _
   3      66.65      30         2         1        _____
   4     127.4        1         1         2
   4     127.4        1         2         1
   .        .         .         .         .
   4     127.4       30         1         2
   4     127.4       30         1         1
   5      60.14       1         1         1
   5      60.14       1         1         2
   .        .         .         .         .
   5      60.14      30         2         1
   5      60.14      30         2         1
   6      73.88       1         2         1
   6      73.88       1         2         2
   .        .         .         .         .
   6      73.88      30         1         1
   6      73.88      30         1         1
   7      76.43       1         1         2
   7      76.43       1         1         2
   .        .         .         .         .
   7      76.43      30         1         1
   7      76.43      30         1         1
   8     853.02       1         1         2
   8     853.02       1         1         1
   .        .         .         .         .
   8     853.02      30         1         2
   8     853.02      30         1         2
   9      41.61       1         1         1
   9      41.61       1         1         1
   .        .         .         .         .
   9      41.61      30         1         1
   9      41.61      30         1         1
```

```
10    120.99      1        1        1
10    120.99      1        2        1
 .       .        .        .        .
10    120.99     30        1        1
10    120.99     30        1        2
```

As mentioned previously, the two-stage cluster surveys used by WHO/EPI are self-weighted samples. As such, each person or household (i.e., the sampling unit) in the population is selected with equal probability. We can say, therefore, that each sampling unit represents some given number of units in the population. For example, if a survey was done of 225 children in a population of 9,870 children, each child in the sample would represent 9,870/225 or 43.87 children in the sampled population. In this case, 43.87 would be the population weight for each child in the survey. The 225 children in the sample would represent (225 x 43.87) or 9,870 children in the population. Equivalent population weights can be derived for each child in multiple surveys.

In the file EPI10.REC, containing data from 10 EPI-style surveys, there are two variables necessary for combining surveys: LOCATION (the survey identification number) and POPW (the population weight).

To combine the results from the ten locations, LOCATION is used as the Strata variable:

Data File	**EPI10.REC**
Main Variable:	**VAC**
Strata:	**LOCATION**
PSU:	**CLUSTER**
Weight:	**POPW**
Crosstab:	<None>
Output to:	<As desired>
Analysis button:	**TABLES**

The results from the analysis are as follows:

```
              COMPLEX SAMPLE DESIGN ANALYSIS
Analysis of VAC
|VAC        |
|           |Total       |
|           |            |

|1          |            |
| Obs       |      1242  |
| Percent   v    55.263  |
| SE %      |     2.620  |
| Lower CL %|    50.128  |
| Upper CL %|    60.398  |

|2          |            |
| Obs       |       910  |
| Percent   v    44.737  |
| SE %      |     2.620  |
| Lower CL %|    39.602  |
| Upper CL %|    49.872  |
```

Conclusion: The vaccination level for the 10 areas is 55.3% (50.1, 60.4). If the data had been (inappropriately) analyzed in ANALYSIS or in CSAMPLE, specifying only the main variable, the results would have been 57% (55.6, 59.8)

```
|Total Obs  |       2152|

|Design eff.|      5.975|
```

```
    Sample Design Included:
    -----------------------
    Sampling Weights from POPW field
    Primary Sampling Units from CLUSTER
    Stratification from LOCATION
```

With PRENATAL as a Crosstab variable, the results are:

```
 Analysis of VAC by PRENATAL
 Comparison between PRENATAL 1 and 2
```

```
|PRENATAL      |VAC
|              |1              |2          |Total      |

|1             |               |           |           |
| Obs          |          675  |       413 |      1088 |
| Percent    V |       76.817  |    61.349 |    69.897 |
| Percent    H |       60.734  |    39.266 |   100.000 |
| SE %         |        3.375  |     3.375 |           |
| Lower CL %   |       54.118  |    32.650 |           |
| Upper CL %   |       67.350  |    45.882 |           |
|Design eff.   |        5.264  |     5.264 |           |

|2             |               |           |           |
| Obs          |          567  |       497 |      1064 |
| Percent    V |       23.183  |    38.651 |    30.103 |
| Percent    H |       42.560  |    57.440 |   100.000 |
| SE %         |        2.414  |     2.414 |           |
| Lower CL %   |       37.828  |    52.708 |           |
| Upper CL %   |       47.292  |    62.172 |           |
|Design eff.   |        2.491  |     2.491 |           |

|Total         |               |           |           |
| Obs          |         1242  |       910 |      2152 |
| Percent    V |       100.00  |    100.00 |           |
| Percent    H |       55.263  |    44.737 |   100.000 |
| SE %         |        2.620  |     2.620 |           |
| Lower CL %   |       50.128  |    39.602 |           |
| Upper CL %   |       60.398  |    49.872 |           |
|Design eff.   |        5.975  |     5.075 |           |
```

```
 CSAMPLE COMPLEX SAMPLE DESIGN ANALYSIS OF 2 X 2 TABLE

Odds Ratio        (OR)            2.088
95% Conf. Limits          (  1.50,   2.90 )

Risk Ratio        (RR)            1.427
95% Conf. Limits          (  1.23,   1.66 )
RR=(Risk of VAC=1 if PRENATAL=1)/(Risk of VAC=1 if PRENATAL=2)

Risk Difference   (RD)           18.174%
95% Conf. Limits          ( 10.29, 26.06 )
RD=(Risk of VAC=1 if PRENATAL=1)-(Risk of VAC=1 if PRENATAL=2)

Sample Design Included:
-----------------------
```

175

```
Sampling Weights from POPW field
Primary Sampling Units from CLUSTER
Stratification from LOCATION

   0 records with missing values
```

Calculating Population Weights

The population weights had to be calculated separately for each survey and then entered as POPW in the data set (see Table 3). They could have been placed in a separate file called WEIGHTS.REC, containing only the variables LOCATION and POPW. In ANALYSIS, this would be RELATEd to the main file, EPI10.REC with the statement

EPI6>**RELATE LOCATION WEIGHT.REC**

A new file would be written for analysis in CSAMPLE with:

EPI6>**ROUTE EPI10.REC**
EPI6>**WRITE RECFILE**

Table 3. Population weights for children in each
location

LOCATION[†]	Population	Sample	POPW[†]
1	9,870	225	43.87
2	33,600	219	153.42
3	14,130	212	66.65
4	27,900	219	127.40
5	12,750	212	60.14
6	15,810	214	73.88
7	16,050	210	76.43
8	180,840	212	853.02
9	9,030	217	41.61
10	25,650	212	120.99
Total	345,630	2,152	

[†] Value entered for each child = Population/Sample

For example, all children in the first survey have LOCATION coded as 1 and
POPW coded as 43.87. In the second survey, the codes for LOCATION and
POPW are all 2 and 153.42, respectively.   The coding follows the same
pattern for the remaining eight surveys using the data in Table 3. If weights
are calculated correctly with this method, each weight is the number of
children represented by the record in which it appears, and the sum of the
weights for the entire sample should equal the size of population from which
the sample was drawn. It should be noted that this is not a requirement in
CSAMPLE, and weights chosen by other valid methods will still give correct
results.

Example: A Telephone Survey With Multistage Stratified Cluster Design

To analyze the telephone survey of adults following the stratified three-stage
design presented previously, use the file called EXAMPLE.REC:

Data File	**EXAMPLE.REC**
Main Variable:	**SMOKE**
Strata:	**STRATUM**
PSU:	**PSUID**
Weight:	**SAMPW**
Crosstab:	**RACE** with value 1 set to 1 and value 2 set to 2

 (This assures that the table is confined to 2 values for this variable)

Output to:	<As desired>
Analysis button:	**TABLES**

This command will separate weighted estimates of the proportion of people who smoke (SMOKE=1) and do not smoke (SMOKE=2) for those with RACE=1 and those with RACE=2, along with estimates of their differences, standard errors, and confidence limits. Accompanying these estimates are estimates (with standard errors) of relative risk, odds ratio, and risk difference for the two-by-two table of smoking status versus RACE.

```
CSAMPLE COMPLEX SAMPLE DESIGN ANALYSIS
 Analysis of SMOKE by RACE
 Comparison between RACE 1 and 2
```

RACE		SMOKE 1	2	Total
1				
Obs		73	60	133
Percent	V	89.025	89.553	89.262
Percent	H	54.887	45.113	100.000
SE %		4.121	4.121	
Lower CL %		46.810	37.035	
Upper CL %		62.965	53.190	
Design eff.		0.912	0.912	
2				
Obs		9	7	16
Percent	V	10.975	10.447	10.738
Percent	H	56.250	43.750	100.000
SE %		11.627	11.627	
Lower CL %		33.462	20.961	
Upper CL %		79.039	66.538	
Design eff.		0.879	0.879	
Total				
Obs		82	67	149
Percent	V	100.00	100.00	
Percent	H	55.034	44.966	100.000
SE %		3.938	3.938	
Lower CL %		47,315	37.248	
Upper CL %		62.752	52,685	
Design eff.		0.934	0.934	

```
        CSAMPLE COMPLEX SAMPLE DESIGN ANALYSIS OF 2 X 2 TABLE

 Odds Ratio         (OR)          0.946
 95% Conf. Limits             (  0.36,   2.49 )

 Risk Ratio         (RR)          0.976
 95% Conf. Limits             (  0.64,   1.49 )
 RR=(Risk of SMOKE=1 if RACE=1)/(Risk of SMOKE=1 if RACE=2)

 Risk Difference    (RD)         -1.363%
 95% Conf. Limits             ( -25.19,  22.46 )
 RD=(Risk of SMOKE=1 if RACE=1)-(Risk of SMOKE=1 if RACE=2)

 Sample Design Included:
 ----------------------
 Sampling Weights from SAMPW field
 Primary Sampling Units from PSUID
 Stratification from STRATUM
```

```
188 records with missing values
```

Conclusion: There is very little difference in smoking rate between the two races.

Computing Means for Continuous Data--An Example

The variable in EXAMPLE.REC called NUMCIGAR represents the number of cigarettes smoked per day in the telephone survey of adults following the stratified three-stage design presented previously. To analyze the data in CSAMPLE, use:

Data File	**EXAMPLE.REC**
Main Variable:	**NUMCIGAR**
Strata:	**STRATUM**
PSU:	**PSUID**
Weight:	**SAMPW**
Crosstab:	**RACE** with values 1 = 1 and value 2 = 2
Output to:	<As desired>
Analysis button:	**MEANS**

Since we wish to treat NUMCIGAR as a continuous variable and compute the mean number of cigarettes per day, choose the MEANS button rather than TABLES button to perform the tabulation.

CSAMPLE will show weighted estimates for the mean of the number of cigarettes the subjects smoke a day among population members in each racial subgroup.

```
                        COMPLEX SAMPLE DESIGN ANALYSIS

      Analysis of NUMCIGAR by RACE
                                                 Confidence Limits
      RACE         Obs        Mean      Std Error    Lower      Upper
      1             73      17.959        1.033     15.934     19.983
      2              9      11.555        1.747      8.131     14.980
      -----
      Total         82      17.256        0.964     15.367     19.145
      -----
      Difference            6.403         2.076      2.334     10.473
      -----

      RACE              Minimum       Maximum
      1                   2.000        40.000
      2                   2.000        20.000
      -----
      Total               0.000        40.000
      -----

      Sample Design Included:
      -----------------------
      Sampling Weights from SAMPW field
      Primary Sampling Units from PSUID
      Stratification from STRATA

      255 records with missing values
```

The mean number of cigarettes per day is higher in RACE 1 smokers than in RACE 2 smokers. The difference is 6.4 cigarettes per day, with confidence limits of (2.3, 10.5).

Note that Race=5 was ignored because the Values 1 and 2 were specified. The estimates for the Total only apply to those with RACE=1 (White) or RACE=2 (Black).

REFERENCES

Kish L Survey Sampling. Wiley and Sons, New York. 1965.

Lemeshow S and Robinson D Surveys to measure programme coverage and impact: a review of the methodology used by the Expanded Programme on Immunization, World Health Statistics Quarterly 1985; 38, 65-75.

Lessler JT and Kalsbeek WD Nonsampling Errors in Surveys. John Wiley and Sons, New York, 1992.

Marks JS, Hogelin GC, and Gentry EM, et al. The behavioral risk factor surveys: I. State-specific prevalence estimates of behavioral risk factors. Am J Prev Med 1985;1:1-14.

```
┌──────────────────────────────────────────────────────────────────────┐
│ EpiInfo Version 6            Statcalc                    November 1993 │
├──────────────────────────────────────────────────────────────────────┤
│     + Disease -                          Analysis of Single Table      │
│  ┌───────────┐                             Odds ratio = Undefined      │
│ +│ 14 │ 112  │    126            Cornfield 95% confidence limits for OR │
│  ├────┼──────┤               *Cornfield not accurate. Exact limits preferred. │
│ -│ 0  │ 309  │    309            Ignore relative risk if case control study. │
│  └───────────┘                                                         │
│ E   14   421      435                            Chi-Squares  P-values  │
│ x                                                ───────────  ────────  │
│ p                                                                       │
│ o                               Uncorrected    :   35.48   0.0000000 ◄── │
│ s                               Mantel-Haenszel:   35.39   0.0000000 ◄── │
│ u                               Yates corrected:   32.00   0.0000000 ◄── │
│ r                        Fisher exact: 1-tailed P-value: 0.0000000 ◄── │
│ e                                      2-tailed P-value: 0.0000000 ◄── │
│                                                                         │
│                          An expected cell value is less than 5.        │
│                            Fisher exact results recommended.            │
│                                                                         │
│                  F2 More Strata; <Enter> No More Strata; F10 Quit       │
│                                                                         │
│                                                                         │
├──────────────────────────────────────────────────────────────────────┤
│   F1-Help    F2-Stratum            F5-Print    F6-Open File   F10-Done │
└──────────────────────────────────────────────────────────────────────┘
```

Chapter **15**

STATCALC and EPITABLE: Statistical Calculators

Tutorial

Run STATCALC from the EPI6 menu. Choose "Tables" by pressing <Enter>.

Enter four numbers, pressing <Enter> after each number. Then press <Enter> again or <F4> to perform calculations. Follow the screen instructions to see exact confidence limits.

Repeat the process, but this time press <F2> for another stratum after entering four numbers and seeing the results. Enter four numbers for stratum 2. Press <Enter> again or <F4> to see the calculations for both strata combined.

Enter 6, 8, or 10 numbers and then press <F4>. Press <F10> to return to the STATCALC menu.

Choose "Sample size." Experiment with each of the three types of sample size by entering numbers and performing calculations.

Choose "Chi Square for Trend." Enter data from the example in this chapter, pressing <F2> after each stratum and <F4> when you are ready to calculate the final results.

Run EPITABLE from the EPI6 menu. Press <F1> to see the documentation, and then try out the various functions as you did with STATCALC.

Overview

STATCALC does statistical analysis of data entered from the keyboard into tables on the screen. Facilities are provided to perform analyses of single and stratified 2-by-2 tables, to calculate sample sizes for studies involving proportions, and to analyze trends with or without stratification.

Running STATCALC

Run STATCALC from the main EPI6 menu. A menu will appear, containing the following choices:

```
        Tables (2 x 2, 2 x n)
        Sample size & power
        Chi square for trend
```

To make a selection, move the cursor bar to the selection with the arrow keys and press <Enter> or press the letter key that is highlighted, such as **T** for the first choice. Help screens containing information about the program and the statistical methods are displayed by pressing <F1>. We recommend that you read these help screens carefully.

Analysis of Single and Stratified Tables (2-by-2 to 2-by-10)

This choice presents a 2-by-2 table on the screen. To do statistics for a single 2-by-2 table, you simply enter four numbers in the table and then press <F4> to perform the calculations. STATCALC calculates the odds ratio and relative risk with confidence limits, three types of chi-square tests, and, if appropriate, Fisher's exact probability calculation. After these have appeared, you may choose to do exact confidence limits. This is most useful if the numbers in the table are small.

For 2-by-2 tables, additional strata may be entered by pressing <F2> after the statistics have appeared. After all strata have been entered, pressing <F4> will present the results of the stratified analysis and the opportunity to do exact confidence limits, if desired.

For single tables with up to 10 cells in the vertical dimension, simply continue entering numbers, followed by an extra carriage return. STATCALC will do a chi square for the 2-by-n table.

The statistics produced in STATCALC are similar to those done from data files by

the ANALYSIS program, except that STATCALC does both Cornfield and *exact* confidence limits for single and for stratified tables. The confidence limits for relative risk in stratified and single tables are those of Greenland and Robins (Biometrics 1985; 41:55-68). Exact confidence limits based on the algorithm of Mehta, Patel, and Gray (J Am Statistical Assoc 1985;80:969-973) are provided for use with tables in which the Cornfield confidence limits are not sufficiently accurate (sparse contingency tables). Further information on statistical methods is given in Chapter 32.

Sample Size and Power

If you choose "Sample size & power," a secondary menu is presented, with the following choices:

```
Population survey
Cohort or cross-sectional
Unmatched case-control
```

All calculations are designed for studies in which the results are proportions expressed in percentages. In a population survey or descriptive study, finding the proportion of persons answering yes to a particular question is a typical goal. The program asks you to specify probabilities that your sample will predict the true situation in the population(s) being sampled and the amount of inaccuracy you are willing to tolerate, and then it calculates the sample size based on these assumptions.

These programs assume random sampling of the population, 100% participation, 100% accuracy in obtaining and recording answers, and other conditions that are seldom present in real life. If you expect 90% participation rather than 100%, you will need to multiply the sample size by 100/90 to obtain a more realistic estimate.

Calculated sample sizes are only a guide, and in many studies the number of available cases or the number of interviews that can be done within the study budget is the final deciding factor. These programs provide a useful way to obtain estimates against which other factors can be weighed. In the long run the assumptions about what result you will obtain in the study will be replaced by actual results, and the assumed confidence levels will be replaced by p values or confidence limits derived from the results. Until such hindsight is available, however, sample size calculations provide a degree of foresight.

Operation of the three calculations proceeds by filling in a confidence level, power, and assumptions about how close an estimate of the actual proportion, relative risk, or odds ratio you would like. For the latter, choose the value furthest from the real

population value that you are willing to tolerate. For example, in the Prospective or Cross-Sectional Study, suppose that you choose 10% as the hypothetical proportion of "Yes" answers to a question in the unexposed persons. You may wish to enter 20% as the closest value you would be able to distinguish in the exposed persons, meaning that any value of 20% or over would give a p value in the final study of .05 or less. This corresponds to a relative risk of 2.0 or an odds ratio of 0.5. Any of these three values may be entered initially to obtain the same results.

After assumptions have been entered, pressing <F4> will result in calculations. The result for the exact assumptions entered will be highlighted, and a series of other values based on varying the assumptions will also be presented. The results can be printed or sent to a file as described below.

Chi Square for Trend

The test for trend is often used for dose-response studies and can also be used to test for trends with age, passage of time, or any ordered variable. The data can be stratified on other variables like age and sex to eliminate confounding from these variables.

The Extended Mantel-Haenszel chi square that is calculated reflects the departure of a linear trend from horizontal (i.e., no trend). If the associated p value is less than .05, there is 95% probability that a trend exists in the underlying population.

Data for analysis of trend by this method must be grouped according to an ordered numerical sequence. The simplest groups are arbitrary ones of 0, 1, 2, 3, etc. These are the "observations" or "scores." Other scores might be the mid-points of groups, such as 0 for 0 glasses of water, 2 for 1-3 glasses, 4 for 3-5, etc. These are the groups over which the presence of trend will be measured. Strata are determined by the confounding variables; common strata would be age or age-sex groups. The following table may be used as an example.

Frequency of Cigarette Smoking Among Women with Myocardial Infarction (MI) and Controls (C), Stratified by Age

| Stratum | | 1 | | 2 | | 3 | | 4 | | 5 | |
| Group | | 25-29 | | 30-34 | | 35-39 | | 40-44 | | 45-49 | |
Cig./day	Score	MI	C	MI	C	MI	C	MI	C	MI	C
None	0	1	131	0	188	3	161	11	169	23	157
1-24	1	1	101	6	152	12	130	21	134	42	97
>24	2	4	51	15	83	22	65	39	68	34	52

From Schlesselman, JJ. Case-Control Studies. Oxford Univ.
Press, NY, 1982, p. 205.

In this example you would enter Stratum 1 on the first screen, then press <F2> and
enter Stratum 2 on the second screen. After entering all five strata in this way, you
would press <F4> to see the results.

Printing or Sending Results to a File

To send calculations for the current screen to the printer, press <F5>. If you prefer
to have output go to a file rather than to the printer, press <F6> instead. The first
time you press <F6> a prompt will request a file name. Once the file is open,
subsequent <F5> commands will add material to the file rather than sending it to
the printer. Pressing <F6> again closes the file. If you give the name of an existing
file for the <F6> command, new material will be added to the existing file without
harming its original contents.

EPITABLE

EPITABLE, designed and programmed by Denis Coulombier, is a late addition to
Epi Info. The documentation is included in a help file accompanying the program.
To see this file, press <F1>. Experiment with the many statistical functions and
graphs available to become acquainted with the program.

Related Topics in Other Chapters

Formulas and references for the statistical procedures are given in Chapter 32.

```
                    EPI Info EXPORT Vers 6.00

              nput file name
              oswego.rec


              ( )   YSTAT     ( )  pistat     ( ) SPSS- C
                              ( ) dBASE       ( )   IXED
              ( )   ELIM      ( ) dBASE       ( ) S atpac
              ( )   OTUS      ( ) dBASE       ( )   ULTLR
              ( ) SPSS-       ( )  ASIC       ( ) E ret

              utput file name
              oswego.SAS


                        O              Cancel



  F1 Help  F10 Quit │ SAS micro/mainframe statistical package format
```

Chapter **16**

EXPORT: Producing Files for Use in Other Database and Statistical Systems

---

> **Tutorial**
>
> **Run EXPORT from the EPI6 menu.**
>
> **The screen that appears has a dialog box with controls that can be manipulated either with a mouse or from the keyboard.   With the cursor in the first field, press enter to see the available .REC files.   Press the <Tab> key to place the cursor in the list of files, then move to OSWEGO.REC with the arrow keys, and press <Enter> to choose this file.**
>
> **If your computer has a mouse, you can use the mouse to select items from the dialog box or to move the cursor.   To select an object or position the cursor, move the cursor with the mouse and then press the left mouse button when the cursor is over its intended destination.**
>
> **The cursor should now be in the section of the dialog called Output Format.   To EXPORT a file for use in SAS, use the down arrow or the mouse to move to SAS, the second choice on the list.   Click the left mouse button or press <Enter> to make the choice.**
>
> **The Output File Name appears as OSWEGO.SAS. Press <Enter> to accept it. The cursor is now on the OK button. Press <Enter> once more to begin processing.**
>
> **To see the file that is produced, return to the EPI6 menu and use EPED to examine the original file, OSWEGO.REC. Now examine OSWEGO.SAS in EPED to see the results of the conversion.**

Overview

Data files created in Epi Info have file names ending in ".REC". They consist entirely of "ASCII" (printable) characters and can be transmitted over electronic mail systems. If you prefer to do analyses in a program other than Epi Info, however, the program called EXPORT will transform ".REC" files into files that can be used in a variety of commercial software systems, as described below. To change the format of data files made in other programs to the Epi Info format, use the IMPORT program as described in Chapter 17.

Types of Files Produced by EXPORT

Menu Choice	File name Suffix	Program
1 - Systat	SYS	Systat (SYSTAT, Inc., Evanston, IL)
2 - SAS	SAS	SAS (Statistical Analysis System, SAS Institute, Inc., Cary, NC)
3 - Delim	SDF	Comma and quote delimited file format. All fields are delimited by quotation marks and separated by commas. The first three records contain the field lengths, data types, and field names.
4 - Lotus	WKS	Lotus 1-2-3 (Lotus Development Corp., Cambridge, MA)
5 - SPSS-X	SPS	SPSS (Statistical Package for the Social Sciences, SPSS Inc., Chicago, IL) This format is for the mainframe versions of SPSS.
6 - EpiStat	EPI	EPISTAT. A statistical analysis program for epidemiologists (Epistat Services, Richardson, TX). This format is for the original shareware version of the program. True Epistat, the commercial version, can use the Fixed-Field format described below.
7 - dBASE 2	DBF	dBASE II (Borland International, Scotts Valley, CA)

8 - dBASE 3	DBF		dBASE III and III Plus (Borland International, Scotts Valley, CA)
9 - dBASE 4 Valley, CA)	DBF		dBASE IV (Borland International, Scotts
10 - BASIC	DAT		Fields are delimited by double quotation marks and separated by commas. This one is identical to DELIM, but without the field name and length information.
11 - SPSS/PC above).	SPS		The microcomputer version of SPSS (see
12 - Fixed	CAR		Fixed field or card format. A popular format for information exchange, especially on mainframe computers.
13 - Statpac	none		StatPac (Walonick Associates, Minneapolis, MN), a statistical package for MS-DOS computers. The Codebook file has a "C" appended to the file name and the Data file name ends with a "D".
14 - MULTLR	DAT DIC		MULTLR, COXPH, and KMSURV. Public-domain .DIC programs for logistic regression, Cox proportional hazards, and Kaplan-Meier Survival Analysis by Nelson Campos and Eduardo Franco. The programs are available from the Epidemiology Monitor, Marietta, Georgia, U.S.A.
15 - Egret	BDF		Egret. A commercial system for advanced epidemiologic statistics by Statistics and Epidemiology Research Corporation, Seattle, Washington, U.S.A.
16 - xBASE	DBF		dBASE format without limits on the number of fields. Can be used in ProDAS.

Using EXPORT

EXPORT will produce files for any of these packages. To run EXPORT, choose
this option from the main menu. The Export menu shown at the beginning of the
chapter will appear.

If you have a mouse, you may move the cursor and then select items by pressing the
left mouse button. Without a mouse, move the cursor from field to field with the
<TAB> key (<Shift-Tab> to go backwards), and select with <Enter>. Holding
down the <Alt> key and pressing one of the highlighted "hotkeys" will select an
item directly. Help is available at any time by pressing the <F1> key.

Give the name of the file to be EXPORTed and the format desired. EXPORT will
produce the new file and return to EPI6 menu or to DOS. If you would like to see
the new file, return to the EPI6 menu and use EPED to examine the result.

Notes on Particular File Types

The ".SAS" file begins with a SAS data statement describing the field names and
assigning columns to each field as though the data had been entered on standard
keypunch cards. Each "card" begins with a four-column field containing the record
number and a two-column field containing the "card" number within the record.
Multiple "cards" per record are produced automatically if the total field length of all
fields in the original questionnaire exceeds 80 characters.

The StatPac statistical package requires both a codebook and a data file, and these are
produced automatically. Neither file name is allowed to have an extension (the part
after the "."). To name these files logically, EXPORT substitutes a "C" or "D" for the
last letter of the file name you entered to get the codebook and data-file names.

The two types of SPSS files created are similar in many ways to the SAS file. They
begin with a description of the data set that tells where the variables begin and end
on the "cards"; the data description is followed by the data, in the format specified
in the DATA LIST at the beginning of the file. The files produced for mainframe
systems are SPSSX files and may not work with older SPSS mainframe systems,
which limit the length of string fields to four characters.

The Lotus 1-2-3 file created by EXPORT acts just like a Lotus ".WKS" file. You
can use the file in Lotus just as if it had been created in Lotus 1-2-3. The widths of
the columns are adjusted so that you can view the entire field instead of just a few
characters. Note: Since ".WKS" files are not in ASCII format, they cannot be
transferred over an ASCII-only electronic mail system.

The EPISTAT file format is somewhat complex. The first set of numbers is the number of fields, followed by the number of records. The next lines contain the number of records having data for each field. The data lines follow, starting with the first field of every record, followed by the second field of every record, etc. Each group of fields is separated by a pair of empty quotation marks (""). Next come the numbers of the records for each field as if they had been sorted using that field as a key. Finally, at the end of the file is a line for each field, containing the name of the field, the sum of all the entries for that field, the sum of the squares of the entries for that field, the median of the entries for that field, and the standard deviation for the entries of that field.

The dBASE II and dBASE III files are "native" dBASE files. They are similar to each other, but not identical. DBASE III has a program to convert dBASE II files to dBASE III format but not vice versa, and dBASE II can read only dBASE II files. Both types of files use a non-ASCII format, and, like Lotus 1-2-3 files, they cannot be transferred over ASCII-only electronic mail systems.

Running EXPORT from the DOS Command Line

EXPORT may be run from DOS with the following parameters:

EXPORT <Input file name>.REC <Output file name> <Conversion choice number>

The conversion choice numbers are:

1.	SYSTAT	6.	EPISTAT	11.	SPSS-PC
2.	SAS	7.	dBASE II	12.	FIXED
3.	DELIM	8.	dBASE III	13.	STATPAC
4.	LOTUS	9.	dBASE IV	14.	MULTLR
5.	SPSS-X	10.	BASIC	15.	EGRET
		16.	xBASE		

C:\>**EXPORT DISEASE DISEASE 11**

produces a file for SPSS-PC called DISEASE.SPS

Related Topics in Other Chapters

Importing files from other systems is described in Chapter 17. Importing and exporting text files from EPED is discussed in EPED Reference Chapter 33.

195

```
                    EPI Info IMPORT Vers 6.00



              Input format
              ( )  ixed     ( )  otus
              ( )  elim     (•) d ase


           nput file name
           data1.DBF



           data1.REC




                0                      Cancel



 F1 Help  F10 Quit │ Select the EPI Info (.REC) file that will be created.
```

196

Chapter **17**

IMPORTing Data Files from Other Programs

Tutorial

Make a file to import, using the EXPORT program. Use the Epi Info file OSWEGO and choose number 12 for the fixed or card-image format. The resulting file will be called OSWEGO.CAR. A mouse may be used as described in the tutorial for Chapter 18.

To tell IMPORT what kinds of fields are in the file OSWEGO.CAR, you must make a questionnaire of exactly the right description and then make a blank .REC file from the questionnaire. Luckily there is already an OSWEGO.QES file on your disk.

Run ENTER from the main menu. Name the data file IMPORTED and use the questionnaire file OSWEGO.QES to make the file. Do not enter any records, but press <F10> to return to the menu.

Now run IMPORT. Enter IMPORTED for the Epi Info file and OSWEGO.CAR for the file to be imported. Choose fixed-length records, and then press <F4>.

To test the new file, run ENTER and examine the data file IMPORTED. There should now be 75 records in the file. Press <F7> to browse through the records and confirm that they are correct.

When importing dBASE files, it is not necessary to make an Epi Info questionnaire or .REC file first. IMPORT will construct the file automatically. This is also true of Lotus .WKS and .WK1 files if the first line of the Lotus spreadsheet contains the names of fields. The data items must begin on the second line of the spreadsheet.

Overview

IMPORT allows files created in other systems to be brought into Epi Info for processing or for conversion to still other file formats. It accepts files either in fixed-length card format or in comma-delimited format with string or text fields enclosed in quotation marks. It will also import Lotus .WKS and .WK1 files and dBASE II, III, or IV files directly. Most microcomputer and mainframe statistics or database programs will produce files in one of these formats.

You must have an exact description of the field names, lengths of fields, and types of data so that you can create an Epi Info questionnaire and .REC file to accept the data. Headers and comments in the original file which do not fit the data format specified must be removed by editing before importation into Epi Info.

Before you import a fixed-length or comma-delimited file, you must first create an Epi Info questionnaire in the format of the file to be imported. If you are importing a dBASE or Lotus .WK1 or .WKS file, IMPORT creates the Epi Info file automatically.

In a fixed-length or comma-delimited file, if the first field in the file is RECORD NUMBER, then the first item on your questionnaire should be RECORDNO followed by the number of underline characters needed to match the RECORD NUMBER field in the original file. It is easier to use underline characters for all fields unless you are absolutely sure that they contain only numbers. The questionnaire does not have to be elegant. Simple phrases to create appropriate field names, followed by underlines of the right length, are sufficient. The layout may be either:

Record No ____   Name _____ Age ____   Sex _

or:

Record No ____
Name _____
Age ___
Sex _

After you have created the questionnaire in the word processor, you must create a record file by running the ENTER program. Do this exactly as you would for a new questionnaire and data file, but do not enter any data. When the questionnaire comes up on the screen ready for data entry, press <F10> to exit the program. This will create an empty data file which can then receive the imported data.

For dBASE or Lotus 1-2-3 files, the task is easier. Variable (field) names in the file being imported will be transferred to the Epi Info .REC file. Lotus worksheets for importation should contain variable names on the first line, and only on the first line. Data items should not be placed on the first line.

When the empty .REC file (if necessary) and the foreign data file are ready, choose IMPORT from the main Epi Info menu. The screen shown at the beginning of this chapter will appear.

IMPORT can also be operated with a mouse. If your computer has a mouse, you can move the cursor and then select items by pressing the left mouse button. Without a mouse, move the cursor from field to field with the <TAB> key (<Shift-Tab> to go backward>), and select with <Enter>. Holding down the <Alt> key and pressing one of the highlighted "hotkeys" will select an item directly. Help is available at any time by pressing the <F1> key.

Enter the file names and the choice of data format and then select the OK button (Alt-K). IMPORT will return to the main menu when through. You may then examine the records in the new file using the ENTER program. It may take several tries and some experimentation to produce a questionnaire that fits the external data set exactly.

Running IMPORT from the DOS Command Line

IMPORT may be run from DOS with the following parameters:

IMPORT <Output (.REC) file name> <Input file name> <Format number>

The format numbers are:

1. Fixed field
2. Comma-delimited
3   Lotus 1-2-3
4. dBASE

C:\ >**IMPORT STUDY.REC STUDY.DBF 4**

produces an Epi Info .REC file from a dBASE file. If the imported file is of type 1 or 2 or if a Lotus 1-2-3 file does not have field names as the first item in each column, an empty .REC file must be created by making a questionnaire and running ENTER, as described earlier in the chapter.

Related Topics in Other Chapters

Exporting files from Epi Info is described in Chapter 16. Importing and exporting text files from EPED is discussed in EPED Reference Chapter 33.

```
                         EPI Info MERGE Vers 6.00

              File
              oswego.rec

              File
              oswego1.REC

               utput file
              results.rec



                  ( )  oncatenate      ( )  pdate
                                       ( )  evise


                      O                      Cancel


 F1 Help  F10 Quit | Merge all fields of dissimilar files (side to side).
```
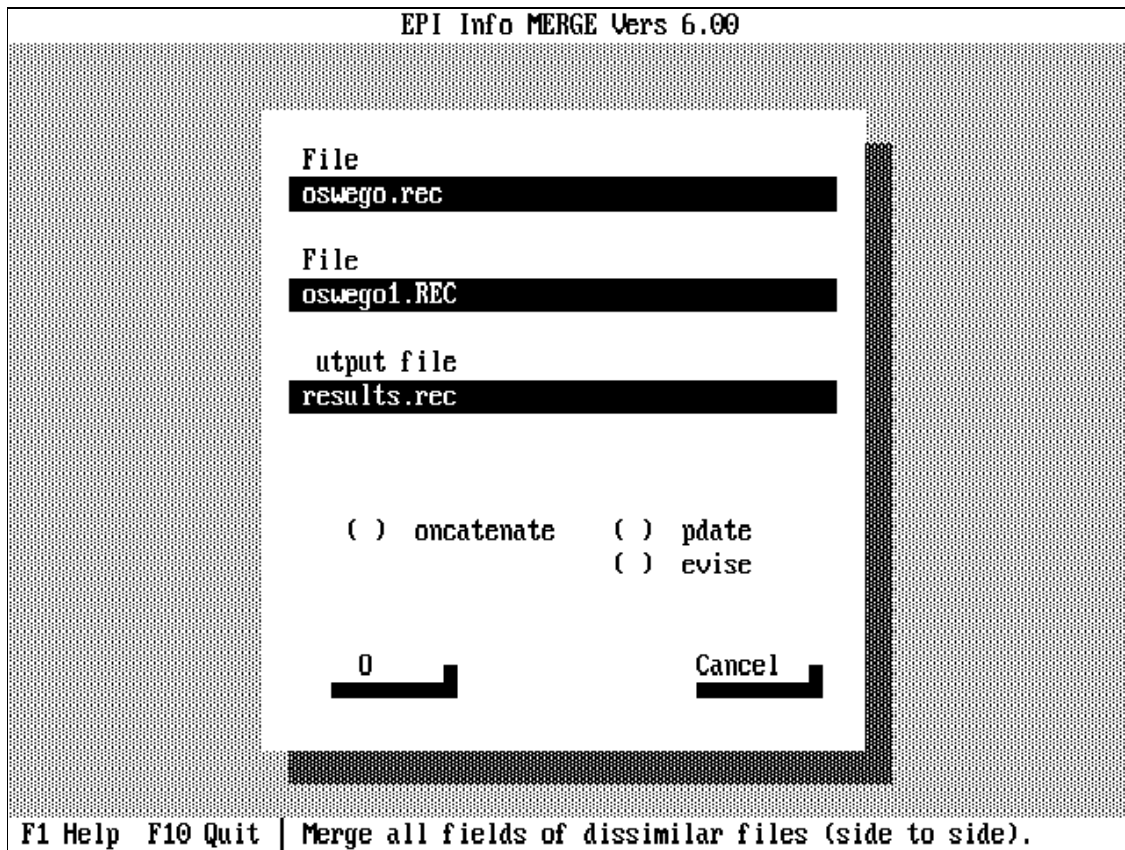
Chapter **18**

MERGE: Merging and Updating Files and Records

Tutorial
Combining Data from Files with the Same Structure

Suppose that several people enter data from the same questionnaire on different computers, each with its own .REC file. All the .REC files were made from the same .QES file and have similar structures. The following technique will put all the records in the same file so that they can be analyzed together. To simulate the situation, we will use two copies of OSWEGO and pretend that they contain different records.

Run ANALYSIS from the EPI6 Menu.

Make a copy of OSWEGO.REC by typing

EPI6>COPY OSWEGO.REC OSWEGO1.REC

Press <F10> to leave ANALYSIS.

Run MERGE from the EPI6 menu. Give the file names OSWEGO, OSWEGO1, and RESULTS. Choose the CONCATENATE option and then OK. After the files have been concatenated, press <F10>, return to the EPI6 menu, and use ENTER to examine the records in RESULTS and verify that there are 150 records rather than the original 75 in OSWEGO.

Other functions of MERGE link related records in different files. See this chapter and Chapter 25 on the relational features of Epi Info for more information.

Overview

MERGE can be used for combining Epi Info files in several different ways or for updating records in one file, using data in another file. It operates in "batch" mode, making a permanent file containing the results of merging two existing Epi Info files.

Many of the functions of MERGE can be accomplished in a dynamic way with the relational features of ANALYSIS. Sometimes, however, a batch program like MERGE is useful to incorporate into a permanent surveillance system or other database application. MERGE may be used to combine records from many different sources submitted to a central processing facility in different files, or to perform batch updates using "update" records sent in from other sites.

Running MERGE

Choosing MERGE from the EPI6 menu will display the screen shown at the beginning of this chapter. If your computer has a mouse, you can move the cursor and then select items by pressing the left mouse button. Without a mouse, move the cursor from field to field with the <TAB> key (<Shift-Tab> to go backward>), and select with <Enter>. Holding down the <Alt> key and pressing one of the highlighted "hotkeys" will select an item directly. Help is available at any time by pressing the <F1> key.

File 1 is the main .REC file for the merge. File 2 is the .REC file containing records that are to be merged into or on top of those in File 1. Output File is a .REC file to be created by MERGE that will contain the results of the merge.

Concatenating Similar Files Top to Bottom -- CONCATENATE

Two files that have the same format (came from the same questionnaire) can be concatenated, with one appended to the end of the other, as follows:

File 1	Name of one file
File 2	Name of the other file
Output file	Name of a new file to be created
Merge Option	Concatenate

After you have answered the questions on the screen, choose OK or press <Enter> twice. The program will convert the file and then return to the merge screen. Press <F10> to return to the EPI6 menu or to DOS.

This option is used for combining files made from the same questionnaire but entered at different times or on different computers. This often happens during an outbreak, when several workers enter data to save time. It also occurs when a number of persons on a network submit periodic reports as separate files, and the results must be combined for further analysis.

Merging All Fields of Dissimilar Files (Side to Side) -- JOIN

This command does the same thing as RELATING the two files in ANALYSIS and writing a new file with WRITE RECFILE. This is discussed in Chapter 25.

Suppose that information has been collected on individuals and on their households in a study. Different questionnaires were used for each purpose, and the result is a file called PERSON.REC, containing one record for each individual, and a file called HOUSE.REC, containing one record for each household. The investigator was careful to see that a field called "HOUSEID" appears on both questionnaires so that they can be linked together.

Now run MERGE as follows:

File 1	HOUSE
File 2	PERSON (The one with multiple records for each one in File 1)
Output file	Name of a new file--HHPERS
Merge Option	JOIN

When you choose OK or press <Enter> twice, a dialog box appears with the names of fields present in both files. Using the mouse button or arrow keys and <Spacebar>, select a field or fields that uniquely identify each record in File 1. Press the <Tab> key to move to the check box for, "Include unmatched records." If you wished to include records that do not have matching identifiers, you would place an "X" in this box with the <Spacebar>. Pressing <Enter> twice will begin the merging process.

For each person record that matches a household record number, the new file will contain a record combining the contents of both the PERSON record and the HOUSE record. The person records in HHPERS now contain copies of matching HOUSE information, and can be processed in ANALYSIS. Processing information for households alone, however, can still be done with the HOUSE.REC file, which has not been disturbed by the merging process. Some questions that can be answered are:

How many households are there? The answer is given by any
frequency or table using HOUSE.REC. Note that using the
merged file would give the wrong answer, since household
information has been replicated for each person in the study.

How many persons under the age of 10 are there? Selecting for
AGE < 10 in the file PERSON.REC or the new file,
HHPERS.REC, followed by a frequency or table, will give the
answer.

How many persons under age 10 live in houses having fewer than
five rooms? This can be answered only by using the new file,
since neither of the original files contains the complete story.
Using HHPERS.REC, first select for AGE < 10 and then for
ROOMS <= 5, followed by FREQ AGE. Or the answer may be
easily extracted from the table that is produced by TABLES AGE
ROOMS.

Once you understand the method of combining files and replicating information in
this way, you will be able to handle two or more levels of information such as
HOUSEHOLDS, INDIVIDUALS, and multiple VISITS to individuals, using Epi
Info files and MERGE. If such work is an objective, you should also read Chapter
25 on the relational features of Epi Info.

Updating File 1 from Matching Records in File 2 -- UPDATE

In disease surveillance work, the first report received may be incomplete or not
entirely correct. In conducting field studies, not all blanks in a questionnaire may
be completed at the first interview, and the information may need to be
supplemented later. This requires that the initial records be updated.

A convenient way to do this with MERGE is to create a second file of records with
the same unique identifier, such as person identification number. Data items are
entered in the second set of records only if the corresponding item on the first
record is to be updated (replaced or overwritten). If a field in File 2 has a missing
value, the corresponding field in File 1 will NOT be replaced, and whatever was in
the field before will remain.

Suppose you wish to update the file CASES.REC from a file called
CASESUPP.REC and create a new file called REVCASES.REC.

CASES.REC contains

CASENO	AGE	SEX
1001	33	
1002	15	M
0999	30	F

CASESUPP.REC contains:

CASENO	AGE	SEX
1001	0	F
1002	16	
1005	56	M

Now run MERGE with the appropriate file names entered, Merge UPDATE, and CASENO as the MatchField. If you choose to include non-matching records, the results will be:

REVCASES.REC

CASENO	AGE	SEX
0999	30	F
1001	33	F
1002	16	M
1005	56	M

If you choose not to include non-matching records, then CASENO 1005 will not appear in REVCASES.REC.

This procedure provides a handy method for updating old records that might already have been sent to another location in a surveillance system, for example. Only the update information needs to be sent; the previous records can then be updated by the recipient using MERGE.

Note that if you wish to erase an item in File 1, it is not sufficient to send blank information in File 2. You will need to adopt a convention for missing data other than 0 or blank for this purpose, such as "XXX" or "---" for an underline field and 99 for a numeric field. This information in File 2 will then replace the item in File 1.

Transferring a File into Another Format (Revising the File Structure)--REVISE

In conducting an investigation, suppose that after entering a number of records you decide to revise the questionnaire. You therefore edit the original .QES file, adding several questions and perhaps subtracting several others.

The following steps are performed automatically by the ENTER program when you press <F6> to revise a questionnaire. In case you wish to do this within MERGE or understand better what happens, the steps are as follows.

First, create a .REC file from your new questionnaire by using ENTER and giving a new name for the data file. Instead of entering data, however, use <F10> to leave the program and save the (empty) new .REC file.

Now choose MERGE from the menu, and enter the following information:

File 1	Name of the new, empty file
File 2	Name of the original file, containing the data
Output file	Name of a new file to be created
Merge Option	REVISE

When you choose OK, MERGE will combine the two files so that the revisions are included. Some kinds of revisions may work better if you reverse the order of the two files, and this might be worth trying if difficulties occur with the order suggested.

Running MERGE from the DOS Command Line

MERGE may be run from DOS with the following parameters:

MERGE <File 1> <File 2> <Output file> <Option number> { Merge
    variable} { Y/N}

The option numbers are:

1. Concatenate
2. Join
3. Update
4. Revise

as explained in this chapter.

<div align="center">

C:\ >**MERGE PERSON HOUSE NEWFILE 2 HOUSEID Y**

</div>

produces NEWFILE.REC by joining HOUSE.REC to PERSON.REC side-by-side, matching records with the same HOUSEID. The "Y" means that records that fail to match should be included in the new file.

Related Topics in Other Chapters

Revising file data structure by revising a questionnaire, creating a new data file, and merging existing data into the new data file is done automatically by the ENTER program, as described in Chapter 8. ENTER actually performs this function by calling the MERGE program, but the details are taken care of internally.

MERGE produces files that contain permanent results of merging. It merges two files "relationally" in a batch mode. Almost all of the operations that MERGE performs can be duplicated in a dynamic way by using the relational features of ENTER and ANALYSIS. Chapter 25 describes these in detail. Once files have been RELATEd in ANALYSIS, a new file can be created to preserve the relationship by using the ROUTE <File name> and WRITE RECFILE commands, thus accomplishing the same thing as a merge with the JOIN option.

```
                  EPI Info VALIDATE vers 6.00

         File
         oswego.rec
         File
         oswego1.REC

                        Output to
                        (•)  creen
                        ( )  rinter
                        ( )  ile

         utput file name


              [ ] Include  eleted records


                 O                  Cancel


  F1 Help  F10 Quit │ Select this if the records are in a different order.
```

Chapter **19**

Duplicate Data Entry and Validation: Two Approaches

Overview

The VALIDATE program is provided for users of Epi Info who like to assure the accuracy of data entry by having two different operators enter the same data into separate files. The files are then compared by the VALIDATE program and any differences are displayed so that questionable entries can be discussed and reconciled in the two files.

In the ENTER program, Option 4, "Reenter and verify records in existing data file," allows a second data entry operator to verify records in an existing data file in a microcomputer version of the "keypunch and verify" machines of the 1960's and 70's.

Running VALIDATE

Before running the VALIDATE program, two separate Epi Info .REC files containing data from the same source must be available. Usually these are produced by having two people enter the data from the same original source, such as a stack of interview forms.

If a unique identifying number or name is entered in each record, the records do not necessarily have to be entered in the same order in both files. There must be an equal number of the records in both files, however, and these records must match in most respects. Only the data-entry errors that you hope to detect will distinguish the pairs of records having the same identifying number or name.

VALIDATE is run from the PROGRAM menu of EPI6 or from DOS by typing VALIDATE at the DOS prompt. The screen shown at the beginning of the chapter will appear.

If your computer has a mouse, you can move the cursor and then select items by pressing the left mouse button. Without a mouse, move the cursor from field to field with the <TAB> key (<Shift-Tab> to go backward>), and select with <Enter>. Holding down the <Alt> key and pressing one of the highlighted "hotkeys" will select an item directly. Help is available at any time by pressing the <F1> key.

Enter the names of the two Epi Info files to be compared. Choose Output to the Screen, Printer, or File. Only when results are to go to a file is it necessary to enter the name of an Output File.

If you wish to include deleted records in the comparison, choose "Include deleted records" with the <Spacebar>. If the records are or may be in different order, choose "Use Unique IDs." You will be prompted to choose a field containing unique identifiers so that the program can select matching records. If the records in both files were entered in exactly the same order, it is not necessary to use the unique identifier field.

When you choose OK or press <Enter> twice, the program will examine both files and produce a report showing differences between the records in each file and the fields in which they occur. Changes can then be made either using ENTER as described in Chapter 8 or the UPDATE facility in ANALYSIS as described in Chapter 11.

Using ENTER to Validate Data

The simplest method of using ENTER to validate data is to use the <F7> and <F8> commands to page through the records and compare them visually with the original forms from which the data items were entered. The BROWSE and UPDATE commands in ANALYSIS offer another way of examining the data while comparing with written records. Be sure to make a backup copy of the file before using UPDATE.

ENTER also offers a more rigorous method for data validation. After records have been entered and saved in a file, option 4 on the ENTER menu, "Renter and verify records in existing data file," allows a data-entry operator to reenter each record and verify the data entered the first time.

When "Renter and verify records in existing data file" is chosen from the ENTER menu, the name of the field that uniquely identifies each record is requested. Enter the name of this identifier field.

When verification is active, the user enters (or accepts) data in the identifier (key) field and the program does an automatic AUTOSEARCH on this field. Records not matching a record in the file cannot be entered.

The operator enters data exactly as for new records. When a field entry matches the data in the file, the cursor moves exactly as for new entries. When an entry does not match, however, a message appears, and the verify operator is given a chance to reenter the field or to compare the original entry with the new one and make a

choice.

Verified (double-entered) records will be identified by a change in the record-end marker (to ^). The record number in the lower right corner of the screen appears with a check mark if the record has been verified.

When the last entry in the record has been made, the statement, "Write Data to Disk? (Y/N)?" appears. A "Y" answer causes the record just entered to be compared with the record in the file. If all fields match, entry begins for the next record. If one or more fields does not match, the cursor returns to the unverified fields one by one. The operator can then reconcile these fields and press the end key or go past the last field to bring up the "Write Data to Disk" prompt and repeat the check. This final verification assures that no fields were missed and that all fields have been verified.

If one of the search functions is activated by pressing a function key on the Ctrl-F menu, and the record has not been completely verified, a message appears:

> "You will be able to examine, but not edit records. To resume verification and editing, press <Ctrl-N>."

This offers a way to search for and examine records without being able to edit them. Pressing <Ctrl-N> for "New record" will allow the verification process to continue.

In ANALYSIS, a new system variable called RECVERIFIED is set to Y if the record is verified and N if it is not. This allows writing programs that handle verified or unverified records in special ways, for example, producing a new file containing only verified records for forwarding to a central agency.

Related Topics in Other Chapters

The other features of ENTER are described in Chapter 8.

```
 File  Edit  System  Input  Programs  Menus  Output  Combo  Hands on!



                Epi Glue

         EXECUTIVE  HEALTH  INFORMATION  SHELL




 F1-Help  F2-New  F3-Open                                      F10-Quit
```
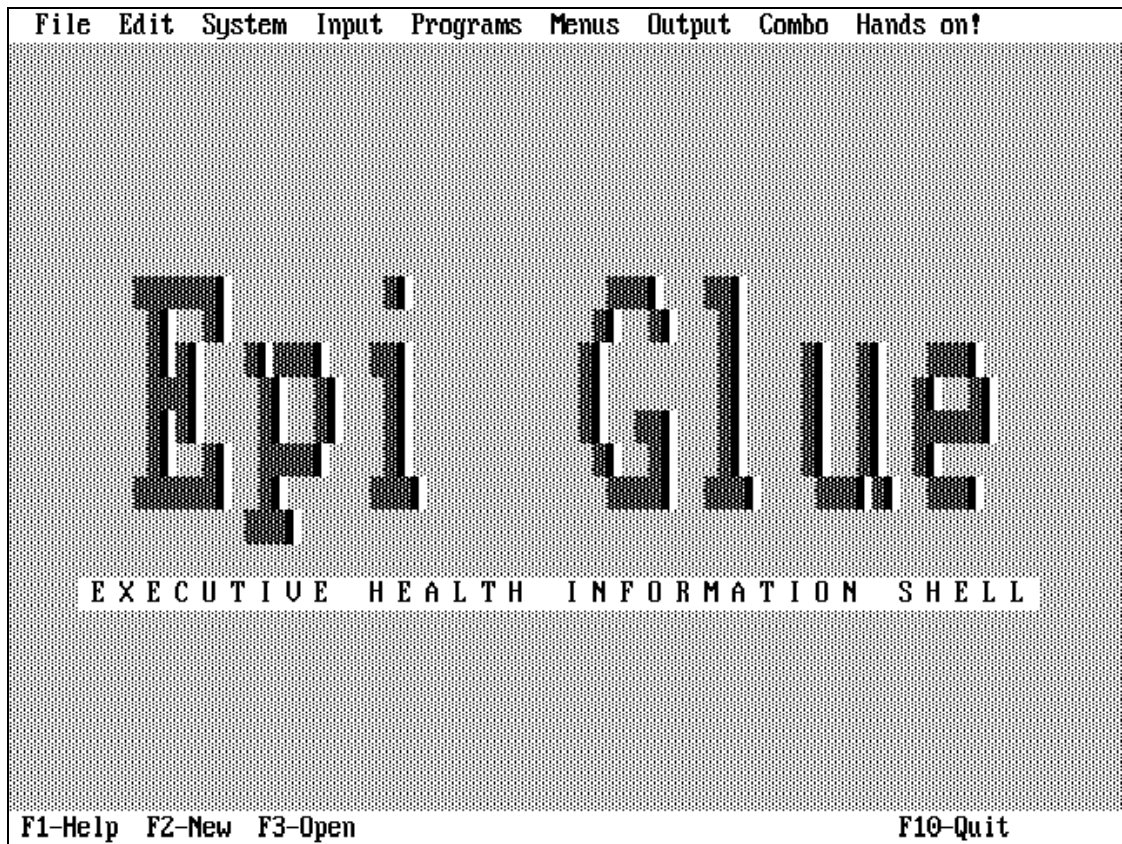
Chapter **20**

EPIGLUE/EPI6: Menuing and Executive Health Information Shell with Hypertext

Denis Coulombier, M.D.

> **Tutorial**
> **Run the EPI6 main menu by typing EPI6 from the DOS prompt. With the mouse or the arrow keys, move the cursor to the EXAMPLES menu and pull down the menu by pressing <Enter> or the left mouse button. Choose EPIGLUE, and give the password "EPIGLUE". Read the startup screen, if any, and then press <Esc>. You will see a new menu containing many choices that illustrate the possibilities for programming your own menus. Try each of the choices, referring to this chapter when necessary, until you understand what each one does.**
>
> **To see the file that controls this menu, choose OPEN from the FILE menu and give the file name EPIGLUE.MNU. You will see a text file containing instructions that drive the EPIGLUE menu and tutorial system. When you have finished examining EPIGLUE, return to the EPI6 menu by pressing <F10> or choosing QUIT from the FILE menu in EPIGLUE.**
>
> **From the FILE menu of EPI6, choose the OPEN command and give the path and file name where Epi Info resides, for example, \EPI6\EPI6.MNU. The text that you see is the file that controls the appearance and activities of the EPI6 menu. Examine the parts of the file and the comments until you understand how it works. You can also open the file \EPI6\EPI6.SCR to see the background that appears when EPI6 is run.**
>
> **You may choose to alter one or both of these files. Before doing so, make your own set of menu files by using the COPY command in DOS:**
>   **COPY EPI6.* <New Name>.***
> **<New name> is, of course, a name of your choice of up to 8 characters.**

> **Then edit the new .MNU and .SCR files to make your own menu. To run the new menu, you would type**
>
> **EPI6 <New name>**
>
> **from the DOS prompt, or, if you wish, just**
>
> **<New name>**
>
> **since you have a copy of EPI6.EXE called <New name>.EXE. Many menus can be run from the same EPI6.EXE file if you invoke them with the EPI6 <New name> method. In this case, you may delete <New name>.EXE.**

Overview

The menu for Epi Info, Version 6, is actually a renamed copy of a program called EPIGLUE.EXE. EPIGLUE provides features for creating menus of your own design, complete with pulldown menus and popup dialog boxes, simply by editing a text file of menu names and commands. It also contains editing and display features that provide quick access to text files. The menu functions with or without a mouse, and a decorative background can be inserted from a character file of your choice.

EPIGLUE provides facilities to display text reports or tables, either as text or as "hypertext," and for navigating and invoking actions by clicking on highlighted key words. Programs such as Harvard Graphics or Epi Map can be called from EPIGLUE to display graphs or maps. As illustrated in the EPIGLUE tutorial, you can also add editing and file management functions or a calendar, calculator, or ASCII table to the menu by adding one-line commands to the .MNU file controlling the menu.

Using EPIGLUE, it is possible to set up an Executive Health Information Shell that displays the latest reports from a number of database systems by making menu choices. The reports may be text, graphics, maps, or other formats. Examples of such systems are included in the epidemiologic surveillance system called NETSS, described in Chapter 26, and in the MULTIFUNCTION menu of EPIGLUE. On a local area network, many departments in an organization can contribute reports that are made accessible from a single, customized version of the EPI/EPIGLUE menu, putting the latest information at the fingertips of executives who need it.

EPIGLUE is the glue that can hold together diverse outputs from many systems on a local area network, or run programs from a variety of sources as a single unified system. Since the copy of EPIGLUE that has been included with Epi Info, Version 6, has been renamed to EPI6, it will be referred to as EPI6 in the rest of this chapter, as it was in the tutorial.

Using the Editor/Viewer in EPI6

You can display and edit files in multiple windows in the EPI6 editor/viewer, accessible from the menu. The size of files is limited to 64 K; use the menu editor for convenience and for displaying items from the menu, and EPED for larger files or more features.

Command	**Key**	**Function**
New	<F2>	Opens a new, empty edit window
Open	<F3>	Opens an existing file in a new window
Save	<F9>	Saves window contents to a file
Save as	<Ctrl-F9>	Allows saving window as a file with a different name

Working with Text Blocks

The editor recognizes <Shift-Arrow key> combinations to highlight a block of text. Holding down <Shift> while moving the cursor highlights a block of text. To copy, move, or delete a block of text, highlight the text by pressing the <Shift> key and the arrow keys. You can copy it to the "clipboard" with <Ctrl-Ins> or cut (delete) it with <Shift-Del>. Text copied or deleted can be pasted to a new location with <Shift-Ins>.

Commands used in EPED for highlighting text are recognized as well, but behave slightly differently. A block can be defined by using <Ctrl-K B> and <Ctrl-K K> keys to mark its beginning and end. As soon as the cursor is moved to a new location, the block disappears, but it has been copied to the clipboard, and can therefore be pasted by using <Ctrl-K C>.

The main difference is in moving text. In the menu editor, the block of text being copied needs to be erased before the cursor is moved. The block is first highlighted using <Ctrl-K B> and <Ctrl-K K> keys, then deleted using <Ctrl-K Y>, and only then the cursor is moved to the new location, and the text pasted using <Ctrl-K C>. <Ctrl-K V> does not do anything.

The remaining functions of the editor are all available from the EDIT and FILE menus. Keyboard shortcuts are also shown for some of the functions.

UNDO Undoes the last deletion at the cursor location. A deletion on a block of text, using ERASE (<Ctrl-Del>) can be undone as long as the cursor was not moved. Characters deleted using the <Delete> key can be restored as long as the cursor was not moved.

CUT <Shift-Del>: Cuts the highlighted block and copies it to the clipboard. Erased text can be pasted back in the document somewhere else using the PASTE function.

COPY <Ctrl-Ins>: Copies the highlighted text block to the clipboard without deleting

it. The text can be pasted in the document at the cursor location.

PASTE <Shift-Ins>: Pastes the contents of the clipboard at the cursor location. The clipboard contains the (single) last block of text that was cut or copied.

ERASE <Ctrl-Del> or <Ctrl-K Y> or <Del>: Erases the highlighted block, without sending a copy to the clipboard as CUT and COPY do. The erased block cannot be retrieved by PASTE, but it can be retrieved by UNDO if the cursor was not moved.

FIND <Ctrl-Q F>: Displays the search dialog box where text to be searched for and search modalities can be specified.

REPLACE <Ctrl-Q A>: Displays the REPLACE dialog box, allowing a Find/Replace. You may enter a word or phrase to be found and replaced and choose several options about how this should occur.

REPEAT FIND/REPLACE <Ctrl-Q L>: Repeats last find or replace, using the same criteria. This is useful when searching for several occurrences of a word in a file. When using REPLACE, if REPLACE ALL is activated, all occurrences of the specified text are replaced from the location of the cursor.

WINDOW Provides a submenu with the following choices:

   SIZE/MOVE <Ctrl-F6>: Moves or resizes the active window. Choose this command to change the size or position of the active window. After you choose SIZE/MOVE, the active window moves in response to the arrow keys. Once you've moved the window to where you want it, press <Enter>. You can also move a window by dragging its title bar with the mouse. If you press <Shift> while using the arrow keys, you can change the size of the active window. Once you've adjusted its size or position, press <Enter>. Windows also have a resize corner at the lower right corner of their frame. You can click and drag on this corner to resize a window if your computer is equipped with a mouse.

   ZOOM <F7>: Zooms the active window to the maximum size. If the window is already zoomed, you this command restores it to its previous size. You can also double-click anywhere on the window's title bar (except where an icon appears) to zoom or unzoom the window.

   TILE <Ctrl-F7>: Windows are arranged on the desktop so that they do not overlap. Vertical tiling or horizontal tiling can be specified in the MENU SETUP option of the FILE menu.

   CASCADE <Alt-F7>: Windows on the desktop are stacked on top of one another but offset so that they can be accessed by clicking with the mouse button or pressing <F6>.

NEXT <F6>: Activates the next window on the desktop. Choose NEXT to cycle forward through the windows on the desktop. Windows are activated in the order they were created.

CLOSE <F4>: Closes the active window. You can also use a mouse to click the Close box in the upper left corner to close a window, or press the <Esc> key.

QUESTIONNAIRE FIELDS <Alt-F8>: This option is available while an editor window is active. It pops up a list of Epi Info fields that may be inserted into a questionnaire, the equivalent of <Ctrl-Q-Q> in EPED.

LARGE LETTERS <Alt-F9>: When the cursor is in an editor window this option pops up a dialog box to let you enter up to 10 characters that produce large letters in the document. A choice of large and medium size is available. These oversize characters can be used for background screen (.SCR) files.

Miscellaneous
<Ctrl-[> and <Ctrl-]> will insert << (ASCII 174) and >> (ASCII 175) respectively.

Double-clicking on the right mouse button pops up a "local menu" with options appropriate to the context. If no windows are active on the desktop, choices for opening a text file, a hypertext file, or creating a new text file are available. If a window is already active, menu choices for printing, tiling, cascading, and switching windows are added. To make a selection, double-click and hold the second click, then move the mouse to highlight a menu option and release. To close the local menu without carrying out an action, release the right mouse button outside the menu frame.

Using the Hypertext Viewer

The hypertext viewer displays files containing hypertext for non-linear viewing of information. A hypertext file allows the user to navigate from one block of information to another by clicking on highlighted options. The hypertext approach is appropriate when there is a large amount of data to access, when text items form a hierarchy, or when users want quick access to only a small portion of the information.

A hypertext topic can be larger than a screen. <PgDn> and <PgUp> will scroll the window one line at a time, while <Ctrl-PgDn> and <Ctrl-PgUp> will scroll one screen at a time. The left and right arrow keys will scroll sideways if the topic is wider than the hypertext window.

In a hypertext file, the cursor may be moved from one highlighted item to another with the <Down Arrow>, <Up Arrow>, <Tab> and <Shift-Tab> keys or the mouse. To choose a highlighted item, press <Enter> or double click on the item with the left mouse button. The cursor will move to the item designated by the selected text.

Hypertext screens can be printed (PRINT menu option or <F5>) or copied to the clipboard (COPY menu option or <Shift-Ins>). From the clipboard, the hypertext screen can be PASTEd to a NEW text file for further editing and saving. Multiple copies of the same hypertext file can be opened on the desktop at the same time, allowing multiple views of the same file. Pressing the <Insert> key while a hypertext window is active toggles the frame of the window on and off.

The hypertext window keeps track of the topics accessed. Previous topics can be recalled by using the following keys:

Previous topic	<Ctrl-Left>
Next topic	<Ctrl-Right>
First topic	<Ctrl-Home>
Last topic	<Ctrl-End>
List of topics	<Shift-F1>

Similar actions can be carried out by clicking on the tape-recorder like icons on the lower left corner of the window frame.

Constructing Your Own Menu

When you type EPI6 from the DOS prompt, the menu program, EPI6.EXE appears. It uses a set of instructions in a file called EPI6.MNU, an optional background screen in EPI6.SCR, a help file called EPI6.HLP, an optional startup file called EPI6.CMD, and printer configuration and printer port information in EPI6.CFG. Each of these files is described in greater detail later in this chapter.

Another way to run the menu is to type EPI6 followed by the name of a menu descriptor file: EPI6 EPIGLUE.MNU. EPI6, in this case, assumes that the menu descriptor file is called EPIGLUE.MNU, that the optional background file, if any, is called EPIGLUE.SCR, the help file EPIGLUE.HLP, the startup file EPIGLUE.CMD, and the configuration file EPIGLUE.CFG. Typing: EPI6 EPIGLUE is the equivalent of EPI6 EPIGLUE.MNU.

It is possible to copy all the EPI6.* files, whatever their extension, to construct a new menu, by typing the command: **COPY EPI6.* MYSYSTEM.***
in which MYSYSTEM is a name of your choice with eight or fewer characters. This will create 5 new files: MYSYSTEM.EXE, MYSYSTEM.MNU, MYSYSTEM.HLP, MYSYSTEM.SCR, and MYSYSTEM.CFG (and, perhaps, MYSYSTEM.BMK). We use COPY rather than RENAME in order to keep a copy of the original EPI6.* files. You can now customize the file MYSYSTEM.SCR and MYSYSTEM.MNU to meet your needs.

Menu (.MNU) Files

A menu file is a text (ASCII) file containing the description of one or more menus,

including display of menu options, corresponding "strings" (phrases) to appear on the status line, and the set of commands to be executed when the user selects an option. An asterisk (*) as the first character on a line can be used to indicate comments anywhere in a menu descriptor file. Material on the same line following an asterisk will be ignored by the program.

The file is organized in three sections that can appear in any order:
   - The menu layout
   - The string table
   - The command definitions

If the active .MNU file is edited, the changes take effect as soon as the file is saved. When the menu file is in an external editor, pressing <Ctrl-F8> on return to the menu updates the menu.

Describing the Contents of a Menu

A menu starts with the name of the menu, followed by a space and the reserved word MENU, as in EPI6 MENU. This line should be followed by a set of instructions starting with BEGIN and ending with END. In this block, five instructions are allowed: MENUITEM, MENUITEM SEPARATOR, MENUEDIT, EPISETUP, and POPUP. More than one menu may be defined in a single file, as in the EPIGLUE.MNU file included with Epi Info.

MENUITEM
Syntax: MENUITEM "&Text to display { ^<Symbol>}",<Command Block Name>

MENUITEM represents one option of the menu. The string of characters between quotes that follows is what appears in the menu. The letter immediately following the "&" sign is highlighted and becomes the "hotkey" that can be used as a short cut to select the menu item.

The optional ^<Symbol> is used to add a symbol to the right side of the item name in the pull down menu. By default, EPI6 adds a triangle (ASCII character 16), if the menu option calls a submenu. The ^<Symbol> allows you to define your own symbol. <Symbol> can be a character, or, if preceded by a number sign (#), a numeric ASCII code. Symbols can be used to indicate on the menu that a symbol or FILEPARAMETER built-in function is used, and thus, that <Ctrl-Enter> will pop up a dialog box for parameter input. The musical notes in the EPI6 PROGRAM menu, signifying a program with command line parameters that may be entered after pressing <Ctrl-Enter>, are an example. Using ^G1 to ^G9 displays the content of the corresponding global variables (see "global variables" below).

<Command Block Name>, after the comma, names a block of commands to be performed when the item is selected (see below). The STRINGTABLE may contain a

help phrase that will appear at the bottom of the screen when the cursor rests on this menu item. Command Blocks and the STRINGTABLE will be explained later in this section. The block of commands corresponding to the name entered is defined later in the file. Alternatively the name of another menu defined in the file can be used to display a new menu. There are several built-in commands that refer to predefined Command Blocks with special meaning:

Command	Action
IDQUIT	Quits the program and exits to DOS
IDCLND	Displays a popup calendar that calculates the epidemiologic week (as used in the United States) for any date selected
IDCALC	Displays a popup calculator
IDASCII	Displays a table of ASCII characters and their codes
IDSETUP	Displays a dialog box for setting-up default directories
IDPASS	Display a dialog box for setting up password
IDFIELD	Display a dialog box for setting field colors
IDCHAR	Display a dialog box for setting default field characters
IDPATH	Display a dialog box for setting the default data path

MENUITEM "&Analysis", DoAnalysis
Displays "Analysis" as a menu choice, with the hot key "A." When this item is chosen, performs the commands in the Command Block DoAnalysis.

MENUITEM "&Analysis^#14", DoAnalysis
Similar to the first example, but displays "Analysis", with a double musical note to the right of the menu option, indicating that it runs a program, and that command line parameters can be entered by pressing <Ctrl-Enter> (see PARAMETER and FILEPARAMETER below).

POPUP
Syntax: POPUP "&Text to appear"
                  BEGIN
                       MENUITEM   (as above, 1 or more)
                  END

POPUP defines a popup menu that will contain one or more menu items. A POPUP menu may be an item on the main menu or on a lower-level menu. In either case, a new menu pops up when the defined item is chosen. POPUP requires that the contents of the submenu it represents be defined by instructions between the words BEGIN and END.

MENUITEM SEPARATOR Syntax: MENUITEM SEPARATOR {"<Text>"}

MENUITEM SEPARATOR displays a horizontal line in the submenu. It is useful for separating groups of functions in the menu. If text is specified, it appears embedded in

the line as a title for a group of menu items.

MENUEDIT Syntax: MENUEDIT

MENUEDIT adds a standard set of two submenus that work with the editor. The two
menus FILE and EDIT can appear anywhere on the top menu bar. The first option is
FILE, with options NEW, OPEN, SAVE, SAVE AS, VIEW HYPERTEXT, CLEAR
DESKTOP, and PRINT for files. MENU SETUP and an ABOUT information box are
also in the FILE menu. The second menu, EDIT, contains options to UNDO the last
deletion, CUT, COPY, PASTE, or ERASE a block of text, FIND, REPLACE, or
REPEAT FIND/REPLACE for text items; and the SIZE/MOVE, ZOOM, TILE,
CASCADE, NEXT, and CLOSE suboptions under WINDOW. It includes items for
entering Epi Info QUESTIONNAIRE FIELDS and LARGE LETTERS.

If only some of the submenus are desired, the following options are available:

MENUE	Adds only the EDIT menu
MENUF	Adds only the FILE menu
MENUW	Adds only the WINDOW menu

The SETUP options configure the printer port and initialization string and offers
AUTOINDENT and VERTICAL TILING default options. The printer options are
automatically saved in the .CFG file when leaving the program.

EPISETUP Syntax: EPISETUP
EPISETUP adds a standard set of three menu options that can be used for setting
questionnaire colors, default file directories, and other values for ENTER and
ANALYSIS in Epi Info.

*A Sample .MNU File (See also EPI6.MNU, EPIGLUE.MNU, NETSS.MNU)
MAINMENU MENU
BEGIN
  POPUP "&Utility"
  BEGIN
    MENUITEM "Ca&lendar", IDCLND
    MENUITEM "&Calculator", IDCALC
    MENUITEM "&ASCII table", IDASCII
  END
  MENUITEM "Item &1", ACTION1
  MENUEDIT
  MENUITEM "Item &2", ACTION2
  MENUITEM "New menu bar", MENU1
  POPUP "&Sub menu 1"
  BEGIN
    MENUITEM "Item &3", ACTION3

        MENUITEM "Item &4", ACTION4
        MENUITEM SEPARATOR
        MENUITEM "Item &5", ACTION5
    END
    MENUITEM "Item &6", ACTION6
END
*An entirely different menu distinguished by the word MENU and
* the BEGIN and END reserved words.
MENU1 MENU
BEGIN
  MENUITEM "Do something", DOSOMTHG
  MENUITEM "Back to main", MAINMENU
END
    Blocks of commands for ACTION1 to ACTION6 and perhaps a String Table would
    follow.

The String Table

The String Table provides short help phrases that appear at the bottom of the screen to
instruct the user about the nature of a menu choice. They appear automatically when
the bar cursor is on the corresponding menu item. String table help is implemented
with a single command:

STRINGTABLE
Syntax: STRINGTABLE
       BEGIN
         <Command Block Name>   "<Help phrase>"
         {/CLOCK} {_NOSEC} {_DOWN}
       END

The STRINGTABLE is a list of phrases or "strings" corresponding to the various
menu options surrounded by a BEGIN END block, and immediately following a
STRINGTABLE statement. The strings appear at the bottom of the menu screen when
the cursor rests on that item. It may be used to provide short help phrases that further
instruct the user about the nature of a menu choice. If a menu item has no help phrase,
the status line at the bottom of the screen is blank. **/CLOCK** displays a digital clock
at the top right corner of the screen. **/CLOCK_NOSEC** does not displays the
seconds. **/CLOCK_DOWN** displays the clock at the right bottom corner of the
screen. Note that a underscore character and no space should appear between the
options. STRINGTABLEs are optional.
STRINGTABLE
BEGIN
  /CLOCK_NOSEC
  ACTION1 "This will do action 1"

    ACTION2   "This will do action 2"
END

Command Blocks

The remaining part of the .MNU file is used for defining commands to be executed for each of the menu options. The block of instructions between BEGIN and END is similar to a DOS batch file. Any DOS instruction appearing there will be executed after shelling to DOS. An external program such as ANALYSIS can be called, and almost any DOS batch file command can be used. Several additional commands are available for capturing user input, and displaying outputs. A Command Block should follow a predefined sequence: input, process, output.

INPUT commands:
PASSWORD, FILEDIALOG, DIALOG, PARAMETER, FILEPARAM, PICKLIST, and PICKDIR can be used in Command Blocks to create interactive dialog boxes to obtain input from the user. As many input commands as needed can be used in a block. The value entered or selected by the user can be passed to the DOS commands using the standard DOS parameters %1 to %9. Refer to your DOS manual for detailed explanations concerning the use of these parameters. Values returned by the input commands will be stored sequentially in the DOS parameters.

DIALOG "Enter your name","<A   >"
DIALOG "Enter your age","<##>"
ECHO Your name is %1 and your age is %2 years.
PAUSE

In the above example, the first dialog returns the name entered in the %1 parameter, and the second dialog returns the age in the %2 parameter. ECHO and PAUSE are standard DOS commands.

DOS Commands:
Next, DOS commands doing a certain job are indicated. Any number of DOS commands can be entered. EPI6 produces a batch file which is executed after the input commands have been processed. The values selected through the input can be referred to by the use of %1 to %9 parameters. Example:

FILEDIALOG "*.REC", "Select a .REC file to copy to a disk"
PICKDIR " ","Select a drive or a directory"
COPY %1 %2

The above example copies a file selected through a FILEDIALOG input command (referred to as %1 since FILEDIALOG is the first dialog) to the drive or directory selected by PICKDIR (the value of which is stored in %2 since it is the second dialog).

OUTPUT commands:
Output is specified by either VIEW, HELP, ROUTEFILE, or ROUTEMESSAGE.
These commands will display output in text windows, hypertext windows, or dialog
boxes. Their use is covered in the next section.

Regardless of the order in which they have been entered, commands will be executed
in the order: input, process, output . *Input* commands will almost never be used alone.
Other types of commands can be used alone, however. In the example later in this
section, ACTION1 uses a DOS command only, ACTION2 an output command only,
ACTION3 input and DOS commands, and ACTION4 includes all three types of
commands.

User input can be passed to DOS or output functions using %1 in the command line as
in DOS batch files (see example of VIEW and HELP functions). There are also 18
system variables, %P1 to %P9, and %G1 to %G9, that can be used in a similar way.
Refer to the discussion of global variables later in this chapter for further explanations.

Special Commands for Command Blocks

DOS batch commands and the following special commands may be used in .MNU
Blocks.

PASSWORD
A PASSWORD statement in a Command Block will prompt the user for entering a
password if any has been defined using the IDPASS command. The password is saved
in the configuration file after encryption. PASSWORD in a .CMD file will prompt the
user for a password when launching the program. Although this mechanism offers
protection from casual intruders, it is quite easy to overcome by erasing or altering
files and should not be thought of as a complete security system. See IDPASS under
MENUITEM above. (New in Version 6.03)

DIALOG
Syntax: DIALOG "{ [Prompt file]} {/c/n} Prompt text"{,"<Fieldtype>"}
        {/D=<Default value>} {/Gx}

The DIALOG instruction in a Command Block displays a popup window with a
prompt string, a caption field, and two buttons for confirming or canceling the entry.
<Fieldtype> is used to specify the type of data entry expected. It can be an Epi Info
field type or a series of characters similar to those used by the Paradox database
program to give a picture of a data entry field. The available field types are described
below. If no <Fieldtype> is specified, the dialog acts as a message box, but does not
accept user input. Pressing "Cancel" stops the process and does not execute
subsequent commands.

The value entered can be passed to subsequent commands in the block as %1. If more

than one parameter is expected from the user, items separated by spaces may be retrieved as "%1", "%2", etc., in the way parameters are passed to batch files in the DOS environment. The optional **/D=** switch is used to specify a default value for the dialog. No consistency check with the picture format is carried out, however; be careful! Quotes "" are needed if spaces are used.

The optional **/Gx** switch, in which x is 1 to 9, assigns the value returned by the dialog to the corresponding global variable. These global variables can be used in Command Blocks by referring to them as %G1 to %G9. See the section on global variables later in this chapter.

Prompt text is displayed over the input line. Prompts can be up to 255 characters long. Two formatting codes can be inserted: /C to center the prompt, and /N to enter a carriage return. If the prompt includes a valid file name between square brackets ([]), then the content of the correspond- ing file is displayed. If the file is more than 12 lines long, it is displayed in a scrolling window.

COMMAND2
BEGIN
  DIALOG "Enter year: ","##" /D=94
  C:\EPI6\ENTER GEPI%1.REC
END

This program will prompt the user for the year, and then run the Epi Info ENTER program with the corresponding file. If the user enters "93", the command C:\EPI6\ENTER   GEPI93.REC will be executed.

COMMAND2
BEGIN
  DIALOG "[ONEFILE]/cEnter option after reading the file: ","##" /D=94
  ECHO You chose %1
END

(Displays ONEFILE in a scrolling window and centers the prompt above the input line.)

Epi Info field-type definitions are recognized for data entry input:

_	General text variable
<A >	Uppercase text variable
#	Numeric variable
<DD/MM/YY>	European date
<MM/DD/YY>	American date

Paradox picture-format symbols are also available:

#	Accept digit only
?	Accept upper or lower case letters
&	Accept only a letter and convert to uppercase
@	Accept any character
!	Accept any character and convert to uppercase

;   Use the next character in the format literally
*   Repetition counter
[]   Contents optional
{ }   Grouping
,   Alternative value separator

Examples of Paradox formats:
    Social Security Number   ###-##-####
    Long distance phone number   (###)-###-####

PARAMETER
Syntax: PARAMETER "{[Prompt file]} {/c/n} Prompt text"{,"<Fieldtype>"}
        {/D=<Default value>} {/Gx}

PARAMETER is a DIALOG function which is invoked only if the <Ctrl Enter> key or
right mouse button is used to activate the menu item. It is used to set up optional entry
of parameters for a program chosen from the menu by pressing <Ctrl Enter> and then
typing the parameters in the dialog box that appears. If the <Enter> key or left mouse
button is used, the PARAMETER function is ignored, and the %1 parameter is reset to
an empty string.
COMMAND2
BEGIN
  PARAMETER "Enter command line parameters for IMPORT","<A >"
   C:\EPI6\IMPORT %1 %2 %3
END
This example will prompt the user to enter the command line parameters for the
IMPORT.EXE program, if, and only if, the <Ctrl Enter> key combination or the right
mouse button is used to activate this menu option. If the user enters three items
separated by spaces, as required by IMPORT, IMPORT will be run with these three
items on the command line. If the menu item is chosen in the normal way by pressing
<Enter> or clicking twice with the mouse button, IMPORT.EXE is run immediately
without command line parameters.

FILEDIALOG
Syntax: FILEDIALOG "<File path>","Prompt text" {/Gx} {/BREAK}

The FILEDIALOG instruction displays a popup window with "Prompt text" as title.
Files specified in the <File path> parameter can be selected by highlighting them with
the cursor and pressing <Enter> or clicking with the mouse.

<File path> specifies a drive and directory and may use the standard DOS wildcard
characters such as *.* to specify all files, or *.REC to display only files with a .REC
extension. It must be enclosed in quotation marks.
The file selected can be passed as a parameter to subsequent commands in the block

using %1 in the way parameters are passed to batch files in the DOS environment.

The optional /BREAK breaks the file name into its components. The path is %1, the file name %2 and the extension %3. Without /BREAK, %1 holds the full file name and path.

The optional /Gx character, in which x is 1 to 9, assigns the value returned by the dialog to the corresponding global variable. These global variables can be used in Command Blocks by referring to them as %G1 to %G9.

COMMAND2
BEGIN
  FILEDIALOG "*.REC","Select a file for data entry"
  C:\EPI6\ENTER %1
END

This program will prompt the user for the desired file and then run the Epi Info ENTER program with the .REC file chosen by the user.

FILEPARAM   Syntax: FILEPARAM "<File path>","Prompt text"{/Gx} {/BREAK}

FILEPARAM is a FILEDIALOG function which is invoked only if <Ctrl Enter> key or right mouse button is used to activate the menu item. If the <Enter> key only is used, the FILEPARAM function is ignored, and the %1 parameter is reset to an empty string. It allows file choice only if the user presses <Ctrl Enter> or the right mouse button. A note to this effect should be included in a String Table message to make the user aware of the option.

COMMAND2
BEGIN
  FILEPARAM "*.REC","Select a file for ENTER"
  C:\EPI6\ENTER %1
END

This program will prompt the user to select a file for the ENTER.EXE program, if, and only if, the <Ctrl Enter> key combination or the right mouse button is used to activate this menu option. Otherwise, it launches ENTER.EXE without command line parameters.

PICKLIST

Syntax:	PICKLIST	<Filename.REC> <Variable name>	{/MULT}
		<Filename.REC>	/VAR {/MULT}
		<Filename>	/ASCII {/MULT}

A picklist instruction in a block of commands pops up a window with a list of items that can be selected. Entering a character will move the cursor to the first item starting with this character. The items can be loaded from an Epi Info data file or from an

ASCII file.

If an Epi Info data file is used, then an additional parameter should be specified. It is either the name of an existing variable in the data set, or /VAR meaning that the names of the variables are to appear in the list. If the name of a variable is specified, the various values taken by this variable will be displayed. No duplicates are allowed, and the list is sorted alphabetically. The value selected can be later accessed by using the %1 to %9 parameters.

If a /MULT switch is added, then multiple selections are allowed and can be selected by using the space bar. A mark on the line distinguishes selected items.
If an item in a list includes an ASCII character 255 (the null character, looking just like a space, but entered by pressing down the <Alt> key while typing 255 on the numeric keypad), only the characters before the null character will be returned. This can be useful to pass codes to a command while displaying the meaning of the code in the list.

The file CHOICE.TXT includes the following:
> **01 Choice_1**
> **02 Choice_2**

The spaces between the 0x number and the Choice_x statement in the above file are not spaces, but ASCII character 255.
> **PICKLIST CHOICE.TXT /ASCII**

Would return **01** or **02** in %1 without **CHOICE_1** or **CHOICE_2**.

> **DoSomething**
> **BEGIN**
>   **FILEDIALOG "*.REC","Select a data file"**
>   **PICKLIST %1 /VAR**
>   **PICKLIST %1 %2**
>   **ECHO READ %1 >TST.PGM**
>   **ECHO SELECT %2 = "%3">>TST.PGM**
>   **ECHO BROWSE>>TST.PGM**
>   **ANALYSIS TST**
> **END**

This program displays a file dialog to select a data file. The selected file name is stored in %1. The picklist reads the selected data file (%1) and lists all the variables in this data set for selection. The selected variable is stored in %2. Then, the second PICKLIST reads the same data file (%1), but displays this time the value taken by the selected variable (%2), putting the result in %3. An ANALYSIS program, TST.PGM, is produced using DOS redirection instructions (> and >>). In this program file, %1 will be replaced by the data file name, %2 by the variable name, and %3 by the value selected. Finally, ANALYSIS runs TST.PGM, reading the file, selecting the specified values of the specified variable, and displaying selected records in the BROWSE spreadsheet format.

PICKDIR   Syntax: PICKDIR "{Default DIR} ","Prompt text" {/Gx} {/ADDSLASH}

PICKDIR displays a dialog for selecting a directory or a drive. The selected directory is returned in the DOS parameter %1. {Default DIR}   specifies the directory selected when creating the dialog. %Gx and %Px variables can be used for default values. The value selected can be passed to a global variable using the /Gx (where x is the number of the variable from 1 to 9). By default the returned directory does not include a trailing backslash (\ ) unless it is a drive (such as A:\). Specifying /ADDSLASH puts a trailing backslash in all instances.
DoSelectDir
Begin
   PICKDIR "%G1","Select a drive for backup" /G1
End
This example lets you change the value of global variable G1.

HELP   Syntax: HELP{ {<PATH>} <Filename>.HLP} {xxxx}{/BMK}

HELP displays the contents of <Filename>.HLP in a hypertext file viewer. The file to be displayed must have been compiled using the HYPER.EXE program supplied with the system. .HLP files may be many megabytes in size.

<Filename>.HLP can be a DOS file path with file name or a directory. If only the directory is given, a file selection box is displayed, allowing for selection of the desired file.

{ xxxx} is an optional parameter that let you specify the topic to appear on screen when the file is loaded. If none is specified, the first topic appears. xxxx is the topic number as listed in the <FILENAME>.LST file produced by HYPER.EXE after compilation. Refer to the HYPER.EXE discussion in chapter 37 for further information.

/BKM is an optional switch that lets you open the file on the topic accessed last time the file was called. This allows you to continue work where you stopped.

VIEW   Syntax: VIEW { {<PATH>} <FILENAME>}   {/NOEDIT}
VIEW followed by a FILENAME displays the content of the given file in the editor. The file must be a standard (ASCII) text file. It may exist in the system, or can be generated by the Epi Info ANALYSIS program through the use of a ROUTE command, to display the output of an ANALYSIS program, for example. If FILENAME is a DOS wildcard file descriptor or a directory, a file selection box is displayed, allowing for selection of the desired file. The file may be viewed or edited as in any editor. The size of files is limited to 64 K, but much larger files may be viewed (only) in the HELP system or edited in EPED.
/NOEDIT is an optional switch that will prevent users from modifying the file. With

the switch active, the file cannot be edited.

FUNCTION1
BEGIN
      VIEW MYFILE.TXT
END
*Displays MYFILE.TXT in the menu editor, where it may be viewed or edited

ROUTEFILE   Syntax: ROUTEFILE   {<FILENAME> {"Message"}}

ROUTEFILE redirects the output of commands in the same block to a file which is then displayed in a read-only editor window. An optional file name can be indicated, as well as a message while the command is executed. If no file name is indicated, $$OUT. is used. If no message is mentioned, "Processing the command ..." is displayed.

DoDir
Begin
      ROUTEFILE DIRLIST.TXT "Generating directory listing..."
      DIR *.*
End

This example will display the content of the directory listing in a read-only window, after having displayed a message "Generating directory listing...**"**

ROUTEMESSAGE Syntax: ROUTEMESSAGE {/NOCLS}

ROUTEMESSAGE in a Command Block will redirect DOS messages to a dialog box provided by the menu. /NOCLS is an optional switch that prevents the screen from being erased while the command is executed.

DoBackup
Begin
      ROUTEMESSAGE
      IF '%G1'=='' GOTO NOT_DEFINED
      DEL ~*.REC>NUL
      COPY *.REC %G1
      GOTO DONE
      :NOT_DEFINED
            ECHO The default drive for backup was not defined.
      :DONE
End

This example will display a message box with "One file copied" if it is successful, or with "The default drive for backup was not defined" if it failed. Note that the >NUL statement (standard DOS redirection to the NUL device) prevents the message concerning this option from appearing in the message box.

RETURN   Syntax: RETURN

RETURN in a Command Block forces the menu to return to the calling menu option upon completion. This is useful for commands embedded deep in the menu structure.

DoOneTask
Begin
       FILEDIALOG "*.REC","Select a file to copy"
       COPY %1 %G1
       * Assuming G1 holds the value of a default drive
       RETURN
End

Global variables

There are 18 global variables that can be used in the menu system--%G1 to %G9, and %P1 to %P9. The only difference between %P and %G global variables is the way their value are set. %P variables are set by a call to IDSETUP in a MENUITEM definition. This displays a dialog box that allows you to specify the value of the 9 %P variables. %G variables are set through DIALOG, FILEDIALOG, PICKLIST, or PICKDIR special commands. They take the value returned by the command. These values are automatically saved when exiting the menu.

A Command Block can include a %Px or a %Gx instruction, which will be replaced at run time by the current value of the global variable. Global variables can be used to design generic applications.

Example to allow users to a define default drive for backing up files:
In the menu:
       Menuitem ...
       Menuitem "Default drive for backup: ^G1",DoSetBackup
       Menuitem "Backup data files",DoBackup...
In the command definitions:
DoSetBackup
Begin
       PICKDIR "%G1","Select default drive for backup" /G1
End
DoBackup
Begin
       IF '%G1'=='' GOTO NOT_DEFINED
       COPY *.REC %G1
       GOTO DONE
       :NOT_DEFINED
              ECHO The default drive for backup was not defined.
              PAUSE
       :DONE
End
All the commands used in the above Command Block are standard DOS commands

except the use of %G1. Refer to a DOS manual for explanations Both EPI6.EXE global variables and DOS environment variables can be used in a Command Block in the same way as in DOS batch files.

> **ShowEnvVar**
> **Begin**
> > **SET ONEVAR=Value of var one**
> > **Echo %ONEVAR%**
> > **PAUSE**
>
> *** All the above commands are standard DOS instructions**
> **END**

The EpiGlue Menu and .MNU File Examples

The EpiGlue menu contains many tutorial examples of use of .MNU files and hypertext to perform useful functions Examples of .MNU files distributed with Epi Info include EPIGLUE.MNU, EPI6.MNU, and NETSS.MNU (the latter found in the \NETSS directory). They all have the following general structure:

> **<Menuname> MENU**
> > **BEGIN**
> > > **<POPUP and MENUITEM section>**
> >
> > **END**
>
> **STRINGTABLE**
> **<Function block name>**
> > **BEGIN**
> > > **<Commands>**
> >
> > **END**

Help (.HLP or Hypertext) Files

There are two types of help files. Blocks of text in context-sensitive help files are displayed if the <F1> key is pressed when a menu item is highlighted. Hypertext help files are displayed by the HELP command. Both types are similar in the way they are constructed, but differ in the way they are accessed. The special commands and markings that make up help files are described in Chapter 37, where instructions are given for creating your own help files.

Context-Sensitive Help

The EPI6.EXE program loads the relevant help screen by keeping track of what is happening on the desktop, in a system variable called CONTEXT. The default CONTEXT value of 0 means that no item on a menu has been highlighted.

Command Blocks, either built-in or user defined, are assigned a CONTEXT value. CONTEXT values in the range 1 to 1999 are reserved for the EPI6.EXE program to display help screens for built-in functions. Values in the range 2000 to 4999 are available for user defined menu items. Values of 5000 or more are reserved for the

234

hypertext viewer launched by the HELP function.

Context sensitive help is provided for built-in functions such as those on the FILE and EDIT menus. The EIS_HELP.HLP file contains the necessary help screens. User-defined context sensitive help screens can be created for any system. The .HLP file should have the same main file name as the corresponding .MNU file. A .TXT extension can be used for the hypertext file before it is compiled. Rules and syntax for writing such files are treated in Chapter 37.

Hypertext Help Files

Hypertext help files can have any name. They follow the same rules for construction as the context-sensitive help files, except for the numbering of the TOPICS. Although help files often consist of paragraphs of text, they can also contain tables or statistics from other programs. Hypertext is a general system for indexing any kind of text file for easy navigation. A complex use of this method to index over 100 tables output by ANALYSIS in the NETSS system is available on the NETSS menu as EPI BULLETIN TABLE 1 on the NETSS surveillance system described in Chapter 26.

Background Screen (.SCR) Files

The optional background file is a text file that appears as background on the desktop. Any ASCII character on this file is displayed "as is" on the background. The file should cover the whole screen, 80 columns by 23 lines. If the number of lines is less than 23, the last line is repeated until the entire desktop is covered, allowing for screen definition in a single 80-column line.

%Px and %Gx global variables can be used in the background. They will be replaced by their value when the background is drawn. In the background, the value of the global variable will overwrite existing background text.

Ten colors can be defined for the background, using a #x statement at the beginning of a given line. The tilde (~) character will alternate the default and the alternate color for each line. The color codes are as follows: #1--Default blue on gray background; #2--Black on gray; #3--Red on gray; #4--White on gray; #5--Green on gray; #6--Yellow on gray; #7--Dark gray on gray; #8--Dark green on gray, #9--Reverse gray on blue; #0--Blinking white on a red background.

Switching Menus

A new menu can be loaded from within a .MNU file. The menu being loaded may be defined within a menu block of the same .MNU file or (if called with <MENU>.EXE <MENU>.MNU) contained in another .MNU file. If a .SCR file of the same name exists, it is also loaded.

    EPIGLUE MENU
    Begin

      MENUITEM "Goes to new menu bar MENU1", MENU1
    End

    MENU1 MENU
    Begin
      MENUITEM "Comes back to the main menubar", EPIGLUE
    End

In this example, if there are two background files, EPIGLUE.SCR and MENU1.SCR, then the background will change as the menu bar changes. The main menu needs to have the same name as the executable file (EPIGLUE.EXE in this application) in order to switch back to the default background. An example is found in the EPIGLUE.MNU file.

Setting up the Printer Port, Initialization Strings, and Other Options

In the FILE menu of EPI6 is an item for MENU SETUP. This allows you to set up the port, a printer initialization string for the printer used to print files from the editor function in EPI6, the default autoindent mode for the editor and the direction for window tiling. The options are saved when quitting the program.

The PORT setting allows three choices for parallel ports. If your printer is functioning from a serial port, a DOS batch file can be used with the MODE command to assign the serial port, COM1, for example, to a parallel port such as LPT1. Instructions for doing this can be found in your printer or DOS manual.

Setup strings for initialization of printers can be found in printer manuals and in Chapter 37. These instructions can be used to reset the printer to default settings or to set up a compressed font so that wide reports will fit on a page. EPI6 accepts either the decimal code of the character or the actual character. If an ASCII decimal code is used it should follow a "\ " character, and should include three digits. Such ASCII codes can be found in most operating system manuals. Many printer codes include the Escape character, entered as \027 (meaning the 27th character of the standard ASCII table).

System Files

SWAP Files Two files are used internally by EPI6.EXE: EPIu.BAT and EPIu.SWP. "u" stands for the NULL character, ASCII #255. EPIu.BAT is a DOS batch file which is created whenever EPI6 shells out and runs DOS functions. EPIu.SWP is a temporary hidden file which stores the content of random access memory to make room for another program to run while EPI6 is shelled out. The .BAT and .SWP files take the name of the current .MNU file, so that running EPIGLUE with the NETSS.MNU file will create NETSu.SWP and NETSu.BAT files, for example.

By default, these files are created in the C:\EPI6 directory. If there is no such directory, or if the user does not have write privileges on this directory, the program

attempts to write these files in the root directory of drives C: to Z:. If this fails, then an error message is displayed. When several users attempt to create a swap file in the same location such as in a network environment, files get numbered sequentially to keep track of who is doing what. Starting the program with a /SWP= parameter will force the swap file to be written to the indicated directory. A trailing backslash is not required, but the validity of the directory is not checked.

EPI6   /SWP=N:\SWAPDIR

Both files are normally erased after completion of the operation. However, if the computer is shut down while the EPI6 program is still active, these 2 files may remain in the root directory of drive C. They will be updated the next time the menu is used.

COMMAND Files If a file with the same name as the menu file, but with a .CMD extension exists, commands listed in this file will be executed upon starting the program.

   PASSWORD prompts for the password for starting the application.
   HELPFILE=<File name> will load the corresponding help file at startup
   TEXTFILE=<File name> will load the corresponding text file at startup
   COMMAND=<Value> execute the corresponding command. Value can be "TILE", "CASCADE" or a numeric value corresponding to the rank of the command in the menu definition file. A value of 1 would execute the first Command Block specified in the .MNU file.
   /E:<xxxx> increases the size of the memory reserved for environmental variables to xxxx. This is useful if you get an "Out of environment space" when running the program.
These commands will be executed sequentially until the end of the file is reached.

Configuration Files: The current configuration is saved when quitting the program. This includes the value of all 10 global variables, the autoindent and vertical tiling modes, default printer port and initialization string. This file is named after the .MNU file, with a .CFG extension. When starting, the program checks for the presence of the configuration file in the calling directory, then in the C:\EPI6 directory, then in the root directory of all drives from C: to Z:. If none is found, default values are assumed, and a file is created in the calling directory. Starting EPI6 with a /CFG=<Configuration file> forces the program to use the configuration file mentioned regardless of the presence of a valid .CFG file.

Related Topics in Other Chapters

Instructions for running the EPI6 menu are given in Chapter 5. The NETSS system in Chapter 26 is an example of the use of the menuing system to set up a comprehensive system for data entry, processing, and display of output from many programs. The structure of EPI6/EPIGLUE help files and the use of the help compiler, HYPER.EXE, are described in Chapter 37. Special instructions for setting up EPI6 on a local area network are given in Chapter 38.

```
                    ┌─────────────────────┐
                    │ EPIAID MAIN MENU │
                    └─────────────────────┘

            ┌───────────────────────────────────┐
            │                                   │
            │      EPIDEMIC INVESTIGATION        │
            │                                   │
            │  ┌─────────────────────────────┐  │
            │  │         Epi 1 Memo          │  │
            │  └─────────────────────────────┘  │
            │     Make Epi Info Questionnaire   │
            │      Develop Study Design         │
            │                                   │
            │         EPED TUTORIAL             │
            │                                   │
            │     Word Processing Tutorial      │
            │                                   │
            └───────────────────────────────────┘

         SELECT A TOPIC USING THE UP AND DOWN ARROW KEYS
                       THEN PRESS <ENTER>
                  PRESS <F10> TO RETURN TO EPED

1  UNTITLED                  EPIAID 360368      L1   C1  TXTInsInd
```
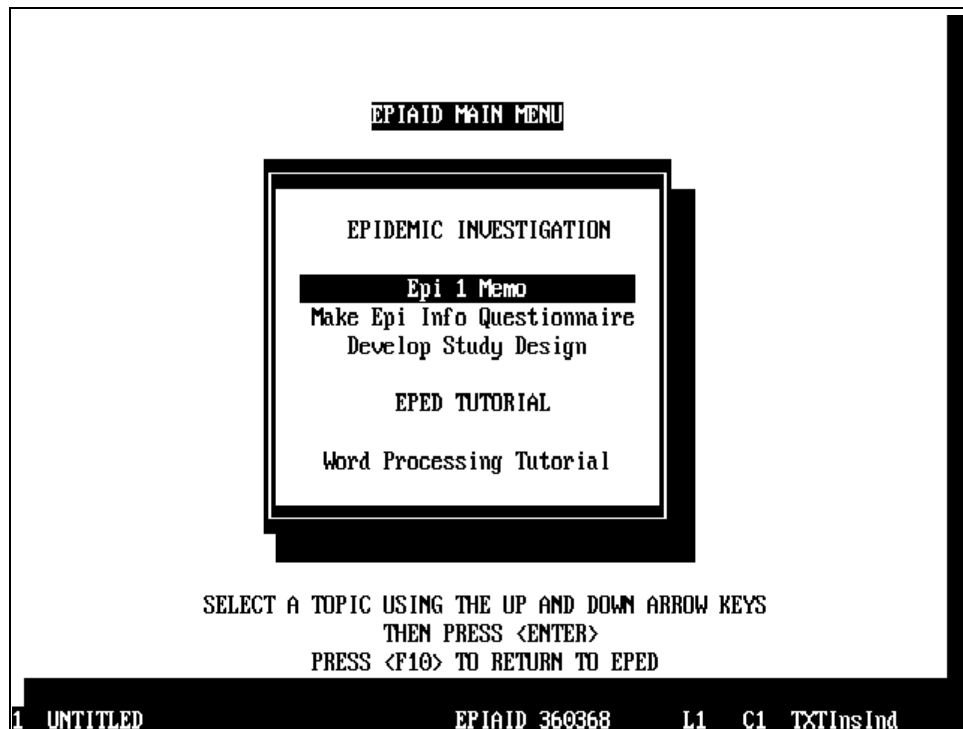
Chapter **21**

EPIAID, the Tutorial and Interactive Text Function in EPED

> **Tutorial**
>
> **Run EPED from the EPI6 menu. Press <F3> to see the EPIAID Menu. Now choose either tutorials or one of the Epidemic Investigation programs. Follow the instructions to do a tutorial or to generate text that can become part of a report of an investigation.**
>
> **Whenever a window appears on the screen, press the <Esc> key after you have read the contents. When a one-line window of a different color requests input, enter as many lines as desired on the screen, and then press <Esc> to signal that you are ready for the next step. When the input window is active, you may use any of the facilities of EPED without disturbing the EPIAID program.**

Overview

EPIAID is a programmable word processing program within the EPED word processor. EPIAID programs are provided to guide you through tutorials on the use of EPED and other programs and to assist in constructing memoranda, questionnaires, and epidemiologic study designs. Your interaction with EPIAID can produce text for a report on an epidemic investigation or other document. If you use EPIAID early in an investigation, it will help you to think through the design of the investigation and of the questionnaire and see that no major topic is overlooked.

EPIAID programs can be written in EPED, as described in Chapter 28, and used to assist in any text-producing or tutorial process. Producing programs that help in composing correspondence, standard memo formats, and many other semi-

structured tasks is possible for anyone who wishes to undertake a small amount of
simple programming and experimentation.

Although it is not nearly as complex as the program called Hypercard on the
Macintosh computer, EPIAID has many of the same qualities of allowing users to
program special applications related to text processing and to display information
with a minimum of programming skill.

Programs in EPIAID present windows on the screen with text describing the subject
of the tutorial. At appropriate times, another type of window, called the "input
window," gives instructions and allows you to operate EPED's commands or type
text on the screen. Whenever the input window is visible, you may type a single
character or word or as many lines as desired, even many pages, before pressing
<Esc>. These lines become part of EPED's working file.

At times the EPIAID program will contribute text that also becomes part of the
EPED working file. The text that you and EPIAID construct together can be saved
as a permanent file by pressing <F9>, the EPED SAVE command.

Using EPIAID

To run an EPIAID program, choose EPED from the EPI6 menu. From the EPED
screen, press <F3> to access the EPIAID main menu. Move the highlight bar to the
EPIAID program you wish to run, and press <Enter>.

EPIAID will guide you through the program, indicating what key to press to
continue and when to type text on the screen. Informational text sometimes appears
in a multi-line display window that overlies the EPED screen. Instructions for
interacting with EPIAID, such as typing sections of a questionnaire or practicing the
EPED word processing commands, will appear in single-line prompt window of a
different color. In either kind of window, pressing the <Esc> key tells EPIAID that
you are through reading or entering text and ready for the next step.

Many of the EPIAID programs run in conjunction with EPED, the word processor,
so that by interacting with EPIAID, you are actually creating a document in EPED.
Whenever a prompt window is present, you can enter and edit text on the screen as
you would when using EPED. The window will indicate what to type or what
commands to invoke and will remain on the screen until you press <Esc> to
continue.

You may end an EPIAID program at any time by pressing <F10>. This will take
you back to the main EPIAID menu, where you can choose another program or
return to EPED.

Using the EPED Tutorials in EPIAID

This tutorial is designed for users with little or no experience with word processing. It introduces the basic concepts of word processing and demonstrates how to invoke commands using both pop-up menus and <Ctrl> key sequences.

The tutorial is divided into five lessons:

-The basics of word processing
-Creating a document
-Retrieving and editing a document
-Using find/replace and cursor movement commands
-Using block commands

Throughout the tutorial, the user is given an opportunity to interact with EPED, practicing the keystrokes and commands as they are explained, and creating, saving, and retrieving a document.

To use the tutorial, choose EPED from the main EPI6 menu. Press <F3> for the EPIAID menu, and choose the EPED Word Processing Tutorial.

Using the Epidemic Investigation Modules in EPIAID

Three EPIAID programs are provided for epidemic investigation. They assist in preparing a preliminary memorandum describing the investigation, in making a questionnaire, and in preparing a study design. The products of your interaction with EPIAID are a memorandum (Epi 1 memo), a questionnaire, and a description of your study that may be used in the methods section of the study report.

Epi 1 Memo

The Centers for Disease Control and Prevention uses a form, called the Epi 1 Memorandum, to record the beginning of an investigation. Essential information about the nature of the problem, why investigation is needed, and the names and titles of concerned persons is included. This program assists in producing the Epi 1 memo in the new format adopted in 1989. If your institution uses a different format, or you wish to devise your own, the program EPI1.EPA can be revised to reflect your preferences, as described in Chapter 28. In any setting it is advisable to

have a systematic approach to deciding that an "epidemic" exists and making sure that appropriate people are consulted and informed. The completed memo can be given a name that includes an arbitrary investigation number, such as EPI1.001.

Make Foodborne Questionnaire

Although it is easy to sit down and write random questions as they come to mind, it is a difficult task to make a questionnaire that proves truly useful all the way through an investigation. The method encouraged in this program is to proceed backward from the items needed for analysis and then construct questions to obtain values for these variables.

For example, if you want to compare the ILL with the NOT ILL, then a number of questions will be needed to ascertain whether a given individual meets your case definition for illness.

"Make Foodborne Questionnaire" provides advice on questionnaire construction, headings for the various parts of your questionnaire, and the pieces of a generic questionnaire useful for foodborne-illness investigations. You may use the foodborne questions or reject them in favor of producing your own questionnaire almost completely "from scratch." The facilities of EPED, particularly the "questions" command, are available any time the input prompt is on the screen. The questionnaire must be given a name ending in .QES, such as Q001.QES. Chapter 8 tells how to enter data in the completed questionnaire.

Develop Study Design

Although there are many subtleties in epidemiologic study design, several basic decisions need to be made. These include whether the study is to be CASE based or EXPOSURE based, and whether or not there will be a comparison group. With a comparison group, a CASE based study becomes a CASE-CONTROL study and the EXPOSURE-based study a COHORT or follow-up study. Without the comparison group, the result is a descriptive study of CASES or of EXPOSED PERSONS.

The "Develop Study Design" program assists in thinking through these decisions and in recording the results in text form. The resulting methods statement can be incorporated into the final report of the epidemic. The program will encourage you to record your decisions before undertaking the study, so that the study can benefit from the thinking incorporated in the text. The file can be given a name such as

DESIGN.001 and then further edited in EPED to produce a "Methods" section for a final report of the investigation.

Related Topics in Other Chapters

The preceding chapter gives a general overview of EPIAID. Chapter 28 tells how to write or modify EPIAID programs yourself. The examples in this chapter, all of which have the file name extension .EPA, can be modified or used for ideas as you develop programs suitable for use in your own epidemiologic environment.

```
                   CONFIDENTIAL MORBIDITY REPORT

         ID          58             UPDATED   03/15/94

  LAST NAME      ┌─────Pick a code─────┐    FIRST     ┌──────────────┐
                 │ AEROMONAS SP  80000 │             └──────────────┘
  ADDRESS        │ AMEB          11040 │    CITY      ┌──────────────┐
                 │ ANTHRAX       10350 │             └──────────────┘
  COUNTY         │ ASEP MEN      10010 │    COUNTY CODE ▢   REGION ▢
                 │ BACT MEN OTH  10650 │
  ZIP CODE       │ BOT FDB       10530 │DATE  / /    AGE 48  AGE TYPE �
                 │ BOT INF       10540 │                        DAYS
  RACE           │ BOT OTH       10550 │HNIC ▢ HISPANIC  SEX F FEMALE
                 └─── ↓ for more ───┘

  EVENT NAME   ┌─────────────┐    EVENT  ┌───────┐             ┌─────1┐

  EVENT TYPE ▢         EVENT DATE  ┌────────┐   OUTBREAK ASSOC. ▢

  MMWR WK    ▢            YR ▢      REPORTED  ┌──────────┐

  OTHER DATA ┌──────────┐  IMPORTED ▢    REPORT STATUS  ▢

EVENTNAME: Valid values: AEROMONAS SP, AMEB, ANTHRAX, ASEP MEN, BACT MEN OTH, BO
<Ctrl-N>-New  <Ctrl-F>-Find  F5-Print F6-Delete F9-Choices  F10-Done  Rec=  58
```

```
F1-Hlp F2-File F3-Epiaid F4-Txt F5-Print F6-Set F7-Find F8-Blk F9-Save F10-Done
EVENTNAME
   help GEPI.HLP 16 5 +1 CONTEXT
   CodeField EVENT
   Codes discodes.REC
   enter
      IF Lastname = "S" and (EVENT=10570 OR EVENT=10220 OR EVENT=10280 OR EVENT=
      THEN
        SUMMDIS = "Y"
        GOTO CASECOUNT
      ENDIF
      IF SUMMDIS = "N" AND EVENT<>10340
      THEN
           help " THIS IS NOT A SUMMARY DISEASE, PLEASE ENTER A NAME" 10 5
           GOTO LASTNAME
      ENDIF
END

EVENT
   help GEPI.HLP 17 5 +1 CONTEXT
   codefield eventname
   noenter
END
1  C:\NETSS\GEPIXX.CHK              EPED   347936       L213 C1  TXTInsInd
```

Chapter **22**

Programming the Data Entry Process with .CHK Files

Chapter 10 describes the function of the CHECK program in making a file with the suffix .CHK. The .CHK file, if present, guides the data entry process to produce range checking, automatic skip patterns, coding of variables, legal values, and repeat fields. CHECK inserts these instructions into the .CHK file automatically, and most users do not
need to know what is in the file.

For many kinds of studies it is useful to be able to perform more complex functions during data entry, such as mathematical or logical operations, checking of one field against one or more other fields, placing text on the screen or removing text, changing the color of parts of the screen, displaying help windows or error messages, and even calling procedures written in other languages such as Pascal and Assembly. These operations can all be performed by adding instructions directly to the .CHK file. The instructions may be added by using the editor that is contained within the CHECK program or another editor such as EPED to edit the .CHK file. In effect, .CHK files are programs that direct data entry.

To edit a .CHK file from within the CHECK program, first run CHECK from the menu or the DOS command prompt. Enter the file name and continue by entering "Y" and pressing <Enter>. To use the CHECK editor, place the cursor in a field and press <F9>. The editor will appear on the screen showing any instructions for that field that are already in the .CHK file. The editor works in similar fashion to EPED for basic text manipulation, responding to Wordstar-like commands. To leave the editor and save changes press <Esc>. The command to abandon changes is <Ctrl-K Q>.

To add commands to the .CHK file, type them into the text in the CHECK editor. The commands will be inserted automatically into the check file. If you prefer to use another text editor instead of the CHECK program to edit the .CHK file, be sure that the editor produces plain text (ASCII) files; if this is true, any editor can be used.

If an error occurs, ENTER will display the name of the .CHK file and line number where the problem was encountered. If this happens, note the line number and edit the .CHK file in EPED. Using the line number displayed on the bottom line of the screen, find the offending line, correct the error, save the .CHK file, and run ENTER again to test the correction.

Examples of operations that can be programmed in check files include:

- Changing the screen colors or displaying messages that appear to be part of the questionnaire
- Calculated fields resulting from mathematical operations on one or more other fields
- Checking one or more fields for relationships, such as one date's being later than another, or inconsistencies, such as male pregnancies
- Help or error messages resulting from improper entries in a field
- Complex statistics or other operations that are written in other languages
- Automatic indexing of fields for fast searching
- Automatic searches during data entry

An operation can be performed each time a field is entered or conditionally when a certain value is entered. IF statements similar to those in ANALYSIS (but with a different format) can be used to produce complex operations. In many ways, an Epi Info .REC file with its associated .CHK file can be made to perform many of the functions of a spreadsheet, in which formulas or "macros" are also associated with each entry location and can direct results to any other entry location. In Epi Info, the format of the screen can be much more flexibly determined, since "cells" (fields) do not have to be contiguous as in a spreadsheet.

General Features of CHECK Programs

.CHK files are arranged in blocks. Each block begins with a field name (or a special term like BEFORE FILE) and ends with the word END. All commands must be within a field-name block. Commands in a block are activated either before or after an entry is made in the field. The usual case is that the commands are performed after an entry has been made and after <Enter> has been pressed or the cursor has left the field automatically. This can be altered by placing commands in blocks called BEFORE ENTRY or AFTER ENTRY within the Field-name block.

The general format for a .CHK file is:

```
<Field name>
        <Command>
        <Command>
        etc.
END
```

```
          <Field name>
             <Command>
             <Command>
             etc.
          END
```

The simpler commands (AUTOJUMP, CODEFIELD, CODES, JUMPS, LEGAL, MUSTENTER, RANGE, REPEAT) are inserted in .CHK files automatically by the CHECK program when function keys are pressed. Others must be inserted with a text editor such as EPED or the editor that appears in CHECK when <F9> is pressed. These commands include CALL, CLEAR, COMMENT LEGAL, COLOR, ENTER, GOTO, HELP, IF, LET, LOAD, TYPE, UNLOAD, and UPDATE. It is often helpful to use CHECK to insert a few commands or a few codes or jumps and then to edit the resulting file by adding additional codes, jumps, or legal values. This is the easiest way to handle large numbers of codes for diseases or counties, for example.

.CHK files are optional, and a block of commands within a .CHK file is not required for every field. The order in which fields appear in the .CHK file is not important, but the order of those containing the word KEY should not be changed unless the KEYs are numbered. Indentation is useful for clarity but is ignored by the program. Every field name must have a corresponding END marker (as must every command that requires an END).

Organizing .CHK Files with INCLUDE

Portions of a .CHK file can be kept in separate files and referenced using the INCLUDE command. The statement INCLUDE <File name> in a block of a .CHK file will cause the file to be included in the check file as though its contents were part of the original .CHK.

```
  COUNTY
   CODEFIELD COUNTYCODE
   CODES
     INCLUDE COUNTIES.TXT
   END
  END
```

The INCLUDEd file name (e.g., COUNTIES.TXT) may include a drive and path. If none is specified, the current drive and directory are used. The extension of the INCLUDEd file name (e.g., .TXT in COUNTIES.TXT) must be specified.

The contents of the INCLUDEd file are processed in place of the INCLUDE

command. When its processing is completed, processing of the previous file will continue exactly as if it had been the only file, and the INCLUDEd file had been embedded within it.

INCLUDE files are a good method for organizing your programming, as it is easier to understand a structure composed of several logical pieces rather than one that extends into the sunset in a single continuous stream. An INCLUDE file may INCLUDE another file.

Mathematical Calculations in CHECK Programs

To perform mathematical calculations on a data item in a field and place the results in a second field, a .CHK program like the following might be used:

```
VISITDATE
        IF BIRTHDATE = .
        THEN
                GOTO AGE
        ELSE
                LET AGE = (VISITDATE -BIRTHDATE)/30.44
        ENDIF
END
```

This says, "If BIRTHDATE is missing then move the cursor to AGE so that the user can enter an age; otherwise, calculate age from the VISITDATE and BIRTHDATE in average-length months and put the result in the AGE field."

The mathematical operators used in ANALYSIS are available in .CHK programs for use with variables contained in the questionnaire or DEFINEd in the .CHK file.

Cross Field Verification

To compare two dates and issue an error message if they are in the wrong sequence, the following program might be used:

```
INVESTREPO
        IF INVESTREPO > "  /  /  " AND INVESTREPO < INVESTSENT
        THEN
          HELP "SOMETHING WRONG\N  PLEASE CHANGE THE DATES" 40 19
          CLEAR INVESTREPO
          GOTO INVESTREPO
        ENDIF
END
```

Here INVESTREPO is the date a report of an investigation is submitted. If INVESTSENT, the date an investigator was sent, is after the report date, then the report was inappropriately submitted before the investigation was begun, and a help window is displayed calling attention to the error. Two other instructions, the CLEAR and GOTO, erase the incorrect data item and place the cursor in the same field again so that the error can be corrected. The IF statement in the first line says, "If INVESTREPO is not blank and INVESTREPO is before INVESTSENT, then put a HELP window 40 characters from the left and 19 lines from the top of the screen, clear the field, and repeat the entry." Note that the multi-line IF statement allows several consequences to be executed for a single IF clause, in this case, three separate actions.

Help Windows

As illustrated above, HELP windows containing text can be put on the screen with the HELP statement. Messages can be more than one line if the Newline symbol \N is included to show where a new line should begin. Choosing the proper location for the HELP window may take some experimentation to place it out of the way of important data items and other messages. HELP windows are automatically erased from the screen when the user presses the <Esc> key.

Automatic Indexing of Fields for Fast Searches

In some data-entry situations, it is important to be able to find existing records quickly. Normally, ENTER does what is called a "sequential" search, examining all records, one by one, to find a match. For files with a few hundred records, this is all that is required. Larger systems requiring frequent searches need indexed files. The index makes searching almost instantaneous, no matter how large the file. Such indexing can be set up in the CHECK program with a single program statement.

To establish and maintain an index for a particular field, simply place the word "KEY" in the .CHK file in the block for that field. If the field has a unique value for each record, then KEY UNIQUE should be used. When the record file is requested in ENTER, an index (.IX) file will be created for each indexed field if none exists, or the index will be updated if out of date. Adding a new record to the file automatically updates the index.

KEY fields are used to make searching faster, but are also necessary for relational file handling. This is described in Chapter 25, and additional details can be found in Chapter 35 under KEY.

SOUNDEX Searches

Soundex is a method of coding names so that differences in spelling are minimized. "Smith" and "Smyth" are both converted to the same Soundex code, for example, so that subsequent searches will find both names.

Soundex searches can be done only on key fields, which must be declared as KEY [UNIQUE] SOUNDEX [key number]. A Soundex key field will actually have two keys/indexes: the normal key/index and the Soundex key/index. If present, UNIQUE and the key number apply to the normal key. The Soundex key will have the next higher number.

Automatic Searching: the AUTOSEARCH Command

If the command AUTOSEARCH is included in the .CHK file for a field, ENTER will automatically search for an existing record that matches whatever values have been entered and notify the user if one is present. A choice is then offered between editing the matching record or continuing with data entry in the new record. Duplicate records are thus detected during data entry and, if the user desires, prevented.

The AUTOSEARCH command in a .CHK file can be followed by a list of fields to match on. If no fields are specified, AUTOSEARCH looks for a match on all fields that have been entered except <IDNUM> and <Today> fields.

AUTOSEARCH can specify whether or not to use available Soundex keys. AUTO SEARCH SOUNDEX will use the Soundex keys on all search fields that have them; AUTOSEARCH alone will use only the normal keys.

Similarly in doing manually guided searches <Ctrl-S> will use the Soundex keys on all search fields that have them.

By default, AUTOSEARCH displays only the first matching record, unless given the argument LIST:

   AUTOSEARCH LIST {SOUNDEX}

(The curly brackets mean that word SOUNDEX is an option; they are not part of the command.)

NOTE that the arguments LIST and SOUNDEX, if present, must precede any field names: The words LIST and SOUNDEX should not be used as variable names.

Codes or Legal Values in Files Rather Than in Memory

Some questionnaires require large lists of codes, legal values, or comment legal values, such as all the cities and towns in a state or the International Classification of Diseases (ICD) codes. If these were inserted into .CHK files or INCLUDEd files all the codes would have to be loaded into memory at one time, and available Random Access Memory (RAM) might not be adequate.

Code, legal values, or comment-legal lists can be put into external (.REC) files rather than in the .CHK file. The files are indexed so that a code can be found very quickly. To do this, make a questionnaire containing either one (for LEGAL values) or two fields (for CODE and COMMENT LEGAL values). Make a .CHK file for this code file that designates both fields as KEY, by inserting the word KEY in the block for each field. Enter the codes or legal values in the questionnaire. An index for the file will be made automatically by ENTER.

If the file is for CODES, the first variable is for the first value (the one the user enters, e.g., "Hepatitis") and the second variable is for the code (perhaps 101). (Usually, the first variable is for the field that declares CODES, the second is for its CODEFIELD.) The names of the fields in the code file do not need to match those in the main file, although this would be a good idea just to keep things straight.

Now return to the .CHK file for the original questionnaire and insert the command, for example, CODES <FILENAME.REC>, where FILENAME.REC is the name of the .REC file containing the codes. Now the codes should be available just as though they had been inserted into the .CHK file. This method allows quick access to a very long list of codes, however, and the list does not occupy space in RAM. It should be possible to have a list of all the cities in a state or the thousands of international disease codes accessible from a questionnaire.

An example is given in the .CHK file for the GEPI system in the NETSS directory. The field is called EVENTNAME and the file containing the codes is DISCODES.REC. The file DISCODES.CHK makes both of the entry fields KEY and therefore indexed.

DEFINEd Variables

DEFINE allows new variables to be defined as they are in ANALYSIS. These variables may be used to hold intermediate values in calculations, to carry over

items from one record to the next, or to transfer the contents of a field from one file to another related file.

Variables declared as GLOBAL can be accessed by related files; otherwise, variables can be accessed only by the file which DEFINEs them. Variables declared as GLOBAL or CUMULATIVE keep their values when new records are read, and should be initialized either in the DEFINE statement or by a LET statement before their first use. Other variables are set to missing each time a record is read.

DEFINE VARNAME <Field indicator, such as ####> {CUMULATIVE or GLOBAL}

(Note: Field types must be specified as _____, #####, etc. The older designations STRING, NUMBER, etc., are not supported)

Standard, GLOBAL, and CUMULATIVE variables differ. Standard variables will be set to MISSING between records and do not retain their values in related files. Global variables may be used to pass variable values to a related file. Cumulative variables retain their value from record to record, but not in related files.

Here is a modified HOUSE/PERSON system that uses GLOBAL DEFINEd variables:

The file, HOUSE.REC, contains the following variables:

HOUSEHOLDR
HOUSEID
ADDRESS
CITY
STATE
BEDROOMS
WATER

The .CHK file, HOUSE.CHK, contains the following blocks:

```
BEFORE FILE
  HELP HOUSE.HLP 1 1 1 CONTEXT
  DEFINE VARADDRESS _____     GLOBAL
  DEFINE VARCITY _____ GLOBAL
  DEFINE VARSTATE __ GLOBAL
END

HOUSEID
  KEY UNIQUE 1
END
```

```
          ADDRESS
            HELP HOUSE.HLP 2 1 1 CONTEXT
          END



          CITY
            HELP HOUSE.HLP 3 1 1 CONTEXT
          END

          STATE
            HELP HOUSE.HLP 4 1 1 CONTEXT
            LEGAL
              GA
              FL
              AL
              SC
            END
            AFTER ENTRY
              VARADDRESS = ADDRESS
              VARCITY = CITY
              VARSTATE = STATE
            END
          END

          BEDROOMS
            HELP HOUSE.HLP 5 1 1 CONTEXT
          END

          WATER
            HELP HOUSE.HLP 6 1 1 CONTEXT
            AFTER ENTRY
              RELATE HOUSEID PERSON
            END
          END
```

PERSON.CHK, for the related file, contains the following blocks:

BEFORE FILE
 ADDRESS = varADDRESS

(Pass the value to ADDRESS in the related file. It will appear in the
ADDRESSfield on the screen)

 CITY   = varCITY
 STATE   = varSTATE
END

 HOUSEID
 KEY 1

        END

USEing the Same Codes and Legal Values in More Than One Field (to Save Random Access Memory)

CODES, LEGAL, COMMENT LEGAL, JUMPS and RANGEs can be listed in a single field but referenced in others with the USE command. Since the list is then loaded into memory only once rather than multiple times, a larger and more complex questionnaire can be accommodated than would otherwise be possible.

Normally, a .CHK file is loaded into Random Access Memory (RAM) when the file is accessed. In Version 5, codes used by more than one field, such as disease codes, or county codes in a large questionnaire, had to be repeated several times within the same .CHK file, once for each field in which they were used.

Note that the field USEd (COUNTY1 in the following example) must have been declared before the field USEing it (e.g., COUNTY2), and must have had the appropriate list (CODES, LEGAL, etc.) declared.

```
COUNTY1
  CODEFIELD COUNTYCODE
  CODES
    ALLEN          1
      ...
    WYANDOTTE    105
  END
END

COUNTY2
  CODEFIELD COUNTYCOD2
  CODES USE COUNTY1
*      use the same list of codes as in COUNTY1
  END
```

Use of CODEFIELD in the second field is optional. If it is omitted, the coding is one-way only:

```
COUNTY
  CODEFIELD COUNTYCODE
  CODES
    ALLEN          1
      ...
    WYANDOTTE    105
  END
END
```

```
          COUNTYCODE
            CodeField REGION
                CODES
                  1 3
                    ...
                105 4
              END
        END

        REGION
          NOENTER

    *When the cursor is in REGION and its value is changed,
    *   COUNTYCODE will not change!
    *If you WANT it to change, uncomment the following
    *line:
    *       CODEFIELD COUNTYCODE
            END
```

Commands for Actions at a Particular Time

Sometimes is is desirable to perform an action such as a calculation or the presentation of a help message at a specific time during data entry that is not the standard time for activation of a block of .CHK commands--i.e., after an entry has been made in a field and the <Enter> key has been pressed.

The following commands will control actions so that they occur before or after a file, record, or field is accessed:

> **BEFORE FILE**
> **AFTER FILE**
> **BEFORE RECORD**
> **AFTER RECORD**
> **BEFORE ENTRY**
> **AFTER ENTRY**

These commands are used in blocks, starting with the command itself and ending with END. BEFORE ENTRY and AFTER ENTRY are always contained within a block for a field within the questionnaire. (See examples below.)

BEFORE FILE is executed each time the file is loaded (i.e., the"core" file's BEFORE FILE is executed when ENTER starts it up; a related file's BEFORE FILE is executed whenever ENTER jumps to the related file). An AFTER FILE block is activated when the file is closed (when ENTER terminates successfully or when it leaves a related file). You might use these to present instructions to the data entry

person, reminding him/her BEFORE FILE about how to get help or see codes, and AFTER FILE to make a backup copy, for example.

BEFORE ENTRY is executed when the cursor moves to a field, before an entry is made. AFTER ENTRY is executed at the time the cursor leaves the field in whose block the command occurs. AFTER ENTRY is the normal time when commands in a field block are executed. It is provided for clarity and in case there are commands both BEFORE ENTRY and AFTER ENTRY within a single field block.

NOTE: There is no functional difference between using a BEFORE ENTRY block and/or an AFTER ENTRY block, and using the command ENTER by itself in the middle of a block without BEFORE ENTRY and/or AFTER ENTRY. These new commands add clarity and allow better self-documentation of the code. (The ENTER command alone also functions and may work more reliably than in Version 5).

BEFORE RECORD is executed when the cursor moves into a new or existing record. AFTER RECORD is executed before the record can be saved to disk--that is, when the message "Write data to disk? Y/N" would normally appear. An AFTER RECORD block is a good place to check the contents of key fields to be sure that the entries are correct before they can be saved. To prevent the record from being saved to disk, be sure to include a GOTO command in the AFTER RECORD in conjunction with an IF statement.

EXIT is used to terminate processing of a sequence of commands, as when one error condition has been determined in a sequence of several checks. EXIT can be used to move the user's focus from an error message to the appropriate field for error correction. EXIT can also be used to avoid an endless line of ELSE IF ... ELSE IF ... ENDIF ENDIF ENDIF, since it will bypass subsequent statements in the block if executed. The necessary ENDIF statements are still necessary to complete the logic in a block, however, even if the flow of control may skip some of the elements.

Here are samples of a .REC file and .CHK file to experiment with some of these commands. The .REC file contains fields named FIELD1, FIELD2, FIELD3.

```
*The .CHK file
BEFORE FILE
  HELP "This is 'BEFORE FILE'" 20 2
  FIELDCHAR X ALL
END

BEFORE RECORD
```

```
  HELP "This is 'BEFORE RECORD'" 20 6
END

FIELD1
 BEFORE ENTRY
  HELP "This is 'BEFORE ENTRY' for FIELD1" 20 10
 END
 ENTER
 AFTER ENTRY
  HELP "This is 'AFTER ENTRY' for FIELD1" 20 14
 END
END

FIELD2
 HELP "This is implicit 'BEFORE ENTRY' for FIELD2" 20 10
 ENTER
 HELP "This is implicit 'AFTER ENTRY' for FIELD2" 20 14
END

AFTER RECORD
 HELP "This is 'AFTER RECORD'" 20 20
 GOTO FIELD2
END

AFTER FILE
 HELP "This is 'AFTER FILE'" 20 23
END
```

Here is a partial example of AFTER RECORD and EXIT used in a GEPI file:

```
AFTER RECORD
 IF ID < 101000001 THEN
  HELP "ID problem.  Quit without saving and restart." 10 15
  GOTO LASTNAME
   * GOTO in order to avoid saving the record
  EXIT
 ENDIF
 IF EVENTNAME = .  THEN
  HELP " EVENTNAME cannot be missing.  " 10 15
  goto EVENTNAME
  EXIT
 ENDIF
 IF EVENTDATE = .  THEN
  HELP " EVENTDATE cannot be missing.  " 10 15
  goto EVENTDATE
```

```
      EXIT
     ENDIF
    IF MMWRWK = .  THEN
      HELP " MMWRWK cannot be missing.  " 10 15
      goto MMWRWK
      EXIT
     ENDIF
   END
```

Changing the Appearance of the Screen with a CHECK Program

Commands in a .CHK file can be used to change the color of the screen. To make this effective for the entire data-entry process, the COLOR commands must be used in the BEFORE FILE block of the .CHK file, and the UPDATE command must follow within the same block. Here is a series of commands to set the screen colors:

```
BEFORE FILE                    (The first field in the questionnaire)
     UPDATE OFF
     COLOR QUESTION 62
     COLOR DATA  63
     COLOR BACKGROUND 3
     SET ALL FIELDCHAR "  "
     UPDATE
END
```

To display a confirmation message on the screen after a coded item is entered, you would use the TYPE COMMENT command, giving the location where the message should appear relative to the current field.

The TYPE COMMENT Command

TYPE COMMENT <Horizontal location> <Vertical location> <Color>

TYPE COMMENT instructs the TYPE command to use the COMMENT (in a COMMENT LEGAL field) as the string to be typed. If the X offset is +0, the first character in the COMMENT will not be typed. The location of the confirming

258

comment or code will be that given by the horizontal and vertical coordinates that accompany the command. If the coordinates are preceded by "+" or "-" signs, they are assumed to be relative to the first character of the entry field.

 (You do not need "" around comments unless they have internal spaces. Shorter comments will overwrite longer ones appropriately without padding with trailing spaces.)

 (If one of your legal values is given as '.' (dot for missing), then comment is taken as the one for a missing value and displayed appropriately. Blanks are displayed for the value in F9 code lists.)

Here is an example of fields with TYPE COMMENT commands. SEX1 will display a value of M as "M MALE" while SEX2 will display "MALE". In each case, the first M (the one in the field entry) will be in the field color, and the remainder will be in the color used in the TYPE command:

```
SEX1
  COMMENT LEGAL
    M  MALE
    F  FEMALE
    U  UNKNOWN
    .  MISSING
  END
  IF SEX1 = M THEN
    HIDE PREGNANT
  ELSE
    IF SEX1 = F THEN
      UNHIDE PREGNANT
    ENDIF
  ENDIF
  TYPE COMMENT +2 +0 16
END

SEX2
  COMMENT LEGAL USE SEX1
  TYPE COMMENT +0 +0 16
END
```

The HIDE, UNHIDE, and FIELDCHAR Commands

HIDE hides a field from view and makes it NOENTER. UNHIDE makes a field visible and returns it to the status it had before HIDE was used.

FIELDCHAR sets a field's "field character" to the specified character. The syntax is

FIELDCHAR x

where "x" is any single character, such as an underline, that is desired as the "background" for a field.

Commands HIDE, UNHIDE and FIELDCHAR take an optional argument of a field name. If no name is specified, they operate on the current field (the one to which the check code block pertains). If a name is given, they operate on the specified field. If the field name is given as ALL (and no field named ALL actually exists) then they operate on all fields in the current .REC file.

IF Statements

IF statements are the key to conditional action during data entry. They control activities in which you want a particular action to occur IF a certain set of conditions is true, and perhaps another set of actions is to occur if the conditions are not true. IF..ELSE statements have the following format in ENTER:

IF <Condition expression> THEN
        <Command>
        {<Command>}
ELSE
        <Command>
        {<Command>}
ENDIF

IF statements in .CHK files differ from those in ANALYSIS by being on several lines and requiring the word ENDIF at the end. Each ENDIF in this example is under the corresponding IF statement, but indenting is not required, and all statements could have been on the left margin of the program. The commands can be in either upper or lower case. The proper syntax and further details about each .CHK command are given in reference Chapter 35. The ELSE clause is optional.

IF statements can be as simple as:

IF AGE > 65 THEN
        MEDICARE = "Eligible"
ENDIF

or quite complex with may layers of nesting and a number of ENDIF's at the end of a series of IF statements.

The HELP System in ENTER

The HELP command allows you to pop up a help window containing a message, or even a window on a large file that allows the user to move from one block of text to another by choosing highlighted portions of the text. This is a simple form of what is known as "hypertext."

Setting up a help window can be as simple as:

HELP "This value extreme. Enter a number between 1 and 55." 15 10
In most cases, it would appear within an IF statement:

IF AGE > 55 OR AGE < 1
THEN
        HELP "Please try again. Must be 1-55." 15 10
ENDIF
CLEAR AGE
GOTO AGE

If an entry is outside the range 1-55, this series of commands will display the message with its upper left corner in column 15, line 10, clear the field, and keep the cursor in the AGE field waiting for another entry.

Longer help messages can be placed in blocks in a help file, as described in Chapter 37. Be sure to distinguish the help file structures for the EPIGLUE menu from those for ENTER, as they are different.

Help files are divided into blocks called TOPICs, each of which has a number. Any block may be displayed by referring to its number in the HELP command in a .CHK file. In the example given above, a suitable message included in TOPIC 1 of a help file called PROBLEMS.HLP could be displayed with the command:

HELP PROBLEMS.HLP 1 15 10

in which 1 is the TOPIC number.

Context-Sensitive Help

If a HELP statement is contained in the BEFORE FILE block of a .CHK file, and the word CONTEXT is added, the message or TOPIC specified will pop up whenever the user presses <F1> during data entry, regardless of the location of the

cursor in the questionnaire.

Further, more specific help can be set up for each field by placing a HELP statements referring to appropriate blocks in the .HLP file and containing the word CONTEXT in suitable blocks of the .CHK file. These help windows will only appear if the user presses <Shift-F1> or enters a question mark in the corresponding field. The HOUSE.CHK file supplied with Epi Info illustrates context sensitive help and displays messages providing more information about how to set up the HELP system. Run ENTER with HOUSE.REC, pressing <F1>, <Shift-F1>, and entering "?" in several fields to see how context-sensitive help works. The HOUSE.CHK file is as follows:

```
BEFORE FILE
  HELP HOUSE.HLP 1 1 1 CONTEXT
  DEFINE VARADDRESS _____GLOBAL
  DEFINE VARCITY _____ GLOBAL
  DEFINE VARSTATE __ GLOBAL
END

HOUSEID
  KEY UNIQUE 1
END

ADDRESS
  HELP HOUSE.HLP 2 1 1 CONTEXT
END

CITY
  HELP HOUSE.HLP 3 1 1 CONTEXT
END

STATE
  HELP HOUSE.HLP 4 1 1 CONTEXT
  LEGAL
    GA
    FL
    AL
    SC
  END
  AFTER ENTRY
    VARADDRESS = ADDRESS
    VARCITY = CITY
    VARSTATE = STATE
  END
END

BEDROOMS
  HELP HOUSE.HLP 5 1 1 CONTEXT
```

```
      END

      WATER
        HELP HOUSE.HLP 6 1 1 CONTEXT
        AFTER ENTRY
          RELATE HOUSEID PERSON
        END
      END
```

Calling Programs Written in Other Languages from .CHK Files (for Advanced Programmers in Other Languages)

Many operations, including moderately complex statistical formulas, can be programmed in .CHK files using the CHECK programming language alone. Nevertheless, there are times when it is useful to call programs written in other languages from a .CHK file.

The simplest procedure is to write an assembly language routine, convert this to object code using an assembler, produce an .EXE file using the DOS LINK program, and then convert this to binary code using the DOS program EXE2BIN. The program must have an origin of 0.

To run the program, use CHECK's LOAD command to load the file into memory. At the right time, the program is run using the CALL command. If desired, the program can be removed from memory with the UNLOAD command, although this is not necessary except to conserve memory, since it will be unloaded as soon as ENTER stops running.

Routines written in other languages like Pascal must be accessed from a binary file that is LOADed and CALLed from the .CHK program. A standard file for this purpose, called ENTFACE.BIN, is provided. The Pascal routine must be written to load itself into memory as a Terminate and Stay Resident (TSR) program. It must set up an interrupt vector with the same number as that used by ENTFACE.BIN, currently 75.

A library of Pascal functions and procedures is provided with Epi Info that installs interrupts and allows access to fields in the current Epi Info questionnaire so that values can be passed to and from these fields by the Pascal program. A concrete example is provided in Chapter 28 on anthropometric software, and instructions for programmers are in Chapter 39.

Related Topics in Other Chapters

Examples of .CHK files containing program statements, including many of those shown as examples in this chapter, are found in Chapters 23 and 26. Details of each of the commands, such as CLEAR, COMMENT LEGAL, COLOR, CODEFIELD, ENTER, GOTO, HELP, IF, JUMPS, AUTOJUMP, LEGAL, LET, RANGE, TYPE, UPDATE, KEY, and AUTOSEARCH are given in Chapter 35. The use of the KEY and ENTER statements for setting up relational databases and of KEY for indexing files are described in Chapter 25.

The three commands necessary for calling programs in other languages are LOAD, UNLOAD, and CALL. Chapters 23 and 35 provide more information on these commands, and Chapter 39 discusses their features for Turbo Pascal or C++ programmers. See Chapter 8 for information on the "protected-mode" version of ENTER, called ENTERX, that uses higher memory in computers with 286 or later processors and more than 640 K of RAM.

```
                         ANTHROPOMETRIC DATA
═══════════════════════════════════════════════════════════════════
   NAME/ID    ┌──────────────────┐      SEX  █
              └──────────────────┘

      AGE    mos  ┌──────┐    BIRTHDATE   ┌──────┐   mm/dd/yy
                  │192.00│                │      │
             yrs  │      │    VISITDATE   │      │   mm/dd/yy
                  └──────┘                └──────┘

   WEIGHT   kilos ┌──────┐              PERCENTILE    Z-SCORE      %MEDIAN
                  │56.00 │
             lbs  │      │
                  │      │    HT for AGE   HAP ┌─────┐  HAZ ┌─────┐  HAM ┌─────┐
             ozs  │      │                     │31.09│      │-0.49│      │97.96│
                  └──────┘                     └─────┘      └─────┘      └─────┘

   HEIGHT     cm  ┌──────┐    WT for AGE   WAP ┌─────┐  WAZ ┌─────┐  WAM ┌─────┐
                  │170.00│                     │24.13│      │-0.70│      │90.18│
              ft  │      │                     └─────┘      └─────┘      └─────┘
                  │      │    WT for HT.   WHP ┌─────┐  WHZ ┌─────┐  WHM ┌──────┐
              in  │      │                     │99.90│      │ 9.99│      │999.90│
                  └──────┘                     └─────┘      └─────┘      └──────┘

                                             Record Flag █




 Write data to disk (Y/N/<Esc>)?
```

266

Chapter **23**

Example: Programs for Nutritional Anthropometry

**Kevin Sullivan, Division of Epidemiology, Emory University,
Jonathan Gorstein, Department of International Health, University
of Michigan,
and Denis Coulombier, EPO, CDC**

Tutorial
**To use the skeletal nutritional anthropometry questionnaire from the
EPI6 menu, choose EXAMPLES, NUTRITIONAL
ANTHROPOMETRY, and then NUTRI, pressing <Enter> after
each choice.**

**A questionnaire will appear. Fill in values for sex, age, weight, and
height (for example, a male, 3 months old, 3.1 kilograms in weight,
and 52 centimeters in length). When you press <Enter> after
entering height, the calculated anthropometric information will
appear in the 10 fields in the middle of the screen. When the record
is saved, you will be able to analyze the computed anthropometric
values in ANALYSIS just as though they had been entered from the
keyboard.**

**This questionnaire can be modified to include other questions for a
nutritional survey. Be careful to do this only on a *copy* of
NUTRI.QES, and do not modify the names of the fields involved in
calculations**

Overview

This chapter presents information on anthropometric calculations. Two different methods for performing anthropometric calculations are presented. The first method works from within the ENTER program during data entry, and illustrates how to call other programs from within ENTER (see Chapter 39). The other method is to perform calculations on data that have already been entered into a data file. This method is called "batch processing," and uses the program, EPINUT, included with Epi Info.

In order to use the anthropometry programs within Epi Info, they must be installed. When installing Epi Info onto a computer, the installation of the anthropometry programs is optional. If you are not sure whether the anthropometry programs were installed, please consult Chapter 4 on installing Epi Info. If the anthropometry programs were installed, the following files will have been copied into the Epi Info directory:

> Files needed for calculating nutritional indices during data entry:
> DOMEAS.EXE   A Terminate-and-Stay-Resident (TSR) program
>   that calculates the anthropometric indices
> ENTFACE.BIN   A module that links ENTER with DOMEAS or
>   other TSR's
> MARK.COM   Marks where DOMEAS is loaded
> REMOVE.COM   Removes DOMEAS from memory
> DOMEAS.PAS, ENTFACE.ASM, ENTFACE.PAS, MEASURE.PAS
> Source code for corresponding modules. Not needed for calculation.
>
> Questionnaires and associated files:
> NUTRI.BAT, .QES, .REC, .CHK
> EUROPE.BAT, .QES, .REC, .CHK
> MEASURE.BAT, .QES, .REC, .CHK
>
> ANALYSIS programs for analyzing nutritional data:
> EPIANTH.PGM, EPIANTH.TXT, ECENTILE.RPT, EFLAG1.RPT
> EFLAG2.RPT, ELOWHAWH.RPT, EPREV.RPT, EZSCORE.RPT,
> NORMAL.REC, AMISS1.PGM, AMISS2.PGM
>
> The EPINUT program for analysis of nutritional data:
> EPINUT.EXE
> EPINUT.OVR
> EPINUT.HLP
> EPINUT.RES
> NUTTEST.REC
> EPINUT.CFG

This chapter is arranged in three sections: The first section provides an overview of the growth reference curves and their interpretation; the second section describes how anthropometric calculations can be done from within the ENTER program; and the third section provides an introduction to the batch processing program, EPINUT.

Overview of Growth Reference Curves

The anthropometric calculations described in this chapter are based on the growth reference curves developed by the National Center for Health Statistics (NCHS) and CDC using data from the Fels Research Institute and US Health Examination Surveys.[1] These growth curves are recommended by the World Health Organization (WHO) for international use.[2]

To calculate the anthropometric indices information is needed on each individual's sex, age, weight, and height. From these data it is possible to form different indices, including those that relate to height-for-age (HA), weight-for-age (WA), and weight-for-height (WH). These indices can be expressed in terms of Z-scores, percentiles, and percent of median relative to the international growth reference population mentioned above. The following abbreviations will be used throughout this chapter:

HAP   Height-for-Age Percentile
HAZ   Height-for-Age Z-score
HAM   Height-for-Age percent of Median

WAP   Weight-for-Age Percentile
WAZ   Weight-for-Age Z-score
WAM   Weight-for-Age percent of Median

WHP   Weight-for-Height Percentile
WHZ   Weight-for-Height Z-score
WHM   Weight-for-Height percent of Median

Interpretation and Uses of Anthropometry

Anthropometry can be used to assess nutritional status at both the individual and the population level. Ideally, individuals should have several weight and height measurements over time so that growth velocity can be assessed. A decline in an individual's anthropometric index from one point in time to another could be an indication of illness and/or nutritional deficiency that may result in serious health outcomes. In some situations, a single set of measurements may be used for screening populations or individuals to identify abnormal nutritional status and priority for

treatment.

At the population level, data are most commonly available from cross-sectional surveys in which the prevalence of low anthropometric indices can be assessed by determining the proportion of the population that falls below a cutoff value. In addition, the mean or median anthropometric value of a population can be compared with the reference value to assess the status of the study population relative to the reference population.

The two preferred anthropometric indices for determining nutritional status are WH and HA, as these discriminate between different physiological and biological processes.[2,3] Low WH is considered an indicator of wasting (i.e., "thinness") and is generally associated with failure to gain weight or a loss of weight. Low HA is considered an indicator of stunting (i.e., "shortness"), which is frequently associated with poor overall economic conditions and/or repeated exposure to adverse conditions. The third index, WA, is primarily a composite of WH and HA, and fails to distinguish tall, thin children from short, well-proportioned children.

The distribution of the indices can be expressed in terms of Z-scores, percentiles, and percent of median. Z-scores, also referred to as standard deviation (SD) units, are frequently used. The Z-score in the reference population has a normal distribution with a mean of zero and standard deviation of 1. For example, if a study population has a mean WHZ of 0, this would mean that it has the same median WH as the reference population. The Z-score cutoff point recommended by WHO, CDC, and others to classify low anthropometric levels is 2 SD units below the reference median for the three indices. The proportion of the population that falls below a Z-score of -2 is generally compared with the reference population in which 2.3% fall below this cutoff. The cutoff for *very* low anthropometric levels is usually more than 3 SD units below the median.

Percentiles, or "centiles," range from zero to 100, with the 50th percentile representing the median of the reference population. Cutoff points for low anthropometric results are generally < 5th percentile or < 3rd percentile. In the reference population, 5% of the population falls below the 5th percentile; this can be compared with the proportion that falls below this cutoff point in the study population.

The calculation of the percent of median does not take into account the distribution of the reference population around the median. Therefore, interpretation of the percent of median is not consistent across age and height levels nor across the different anthropometric indices.[2]

Traditionally, in the United States and some other countries, percentiles are used as cutoff points. In other parts of the world, either Z-scores or percent of median are used, although WHO favors the use of Z-scores.[2]

Z-scores and percentiles are directly related. Both rely on the fitted distributions of the indices across age and height values and are consistent in their interpretation across anthropometric indices. Z-scores are useful because they have the statistical property of being normally distributed, thus allowing a meaningful average and standard deviation for a population to be calculated. In addition, Z-scores have a greater capacity to determine the proportion of a population that falls below extreme anthropometric values than do percentiles.

Percentiles are useful because they are easy to interpret (e.g., in the reference population 3% of the population falls below the 3rd percentile). Percentiles, however, are generally not normally distributed in either the reference or the study populations.

The more common cutoff value used is < -2 SD. The prevalence of < -2SD can be compared with other countries as shown in Table 1.[3] For example, in a survey of children, if the prevalence of weight-for-height <-2 SD if found to be 16.7%, this would be considered to be a *very high* prevalence of low WH. If the prevalence of low height-for-age is found to be 18.9%, this would be a low prevalence of stunting. The prevalence of low anthropometric indices should be presented by one-year intervals for children less than six years of age, or, if age is unknown, for children <85 centimeters compared with those ≥85 centimeters, which approximates comparing the children <2 years of age to those ≥2 years.

Table 1: Prevalence of low anthropometric values (<-2SD) compared to other surveys for children five years of age or less

Relative Prevalence of Low Anthropometric Values				
Index	**Low**	**Medium**	**High**	**Very High**
Low WH	<5.0%	5.0-9.9%	10.0-14.9%	≥15.0%
Low HA	<20.0%	20.0-29.9%	30.0-39.9%	**≥40.0%**
Low WA	<10.0%	10.0-19.9%	20.0-29.9%	≥30.0%

Limitations of Growth Reference Curves

HA and WA indices can be calculated for individuals from birth up to 18 years of age. WH indices are calculated for males to 138 months (11.5 years) of age and less than 145 cm (57 inches) and for females to 120 months (10 years) of age and less than 137 cm (53 inches). WH cannot be calculated for children less than 49 cm (19.3 inches). For children less than 2 years of age, recumbent (i.e., lying down) length measurements are assumed; for children 2 years of age and older, height refers to standing height.

No anthropometric indices are calculated if sex is unknown or miscoded because there are separate growth reference curves for males and females. If weight is unknown, only HA will be calculated; if height is unknown, only WA will be calculated; and if age is unknown, only WH will be calculated. When age is unknown, children shorter than 85 centimeters are assumed to be less than 2 years of age; otherwise, WH is calculated with the assumption that the child is 2 years of age or older.

How to Reduce Anthropometric Errors

Below are some basic steps to follow to ensure that age, weight, and height data are collected accurately.

- Make sure the equipment is correctly calibrated on a regular basis.
- Thoroughly train those who collect the data.
- To reduce errors on the child's age, collect information on both the child's age and the dates of birth and measurement. The year of birth is frequently given incorrectly. Compare the calculated age with the age provided by the child's caretaker. If there is a large discrepancy between the two age values, the age provided by the caretaker is probably closer to the true value. Check the year of birth and see how the anthropometric indices change if you correct the year of birth to correspond with the stated age.
- After the age, sex, weight, and height information are collected,check the data against a growth chart or by calculating the anthropometric indices on a computer. Children with extreme values should be remeasured.
- For research projects, data should be entered twice and compared using the VALIDATE program ("double data entry"--see Chapter 19).

Using ENTER for Anthropometric Calculation

The .BAT, .QES, .REC, and .CHK files called NUTRI, EUROPE, and MEASURE that are supplied with the Epi Info system are for use with a program called DOMEAS which does anthropometric calculations to compare a child's age, sex, height, and weight with the international growth reference curves. To see this program in action, you must first load DOMEAS into memory so that it can be called by ENTER after age, sex, height, and weight have been entered. You can start NUTRI from the EPI6 menu as described in the tutorial at the beginning of this chapter.
It is also possible to run NUTRI.BAT from the DOS prompt, as follows:

C:/EPI6>**NUTRI**

Both methods run a DOS batch file called NUTRI.BAT that in turn loads DOMEAS (a terminate-and-stay resident or TSR program containing the reference standard calculations) into memory and runs ENTER with the NUTRI.REC questionnaire. Provide answers for the items in the questionnaire, for example, age 46 months, sex F, weight 14.6 kgs, height 102.9 cms. After height has been entered, there will be a brief pause while the calculations are being performed, and then the results will appear in the 10 fields in the middle portion of the screen:

```
Sex  (1/M/m or 2/F/f)        F
AGE in months               46
WEIGHT in kilograms       14.6
HEIGHT in centimeters     102.9

                PERCENTILE      Z-SCORE        %MEDIAN

HT for AGE   HAP    73.25     HAZ    0.62    HAM 102.46

WT for AGE   WAP    26.04     WAZ   -0.64    WAM  93.18

WT for HT    WHP    13.90     WHZ   -1.08    WHM  90.40

                   Record Flag 0
```

You can save the results, as with any Epi Info record, and then enter another child's values. Since all the results will appear in the .REC file, you will be able to analyze the data the same way you would measure any other data that had been entered. Other examples can be seen by running EUROPE.BAT and MEASURE.BAT files to see EUROPE.REC and MEASURE.REC, respectively.

The differences among NUTRI, EUROPE, and MEASURE are as follows:

	NUTRI	**EUROPE**	**MEASURE**
Calculate age from 2 dates	no	yes	yes
Date format	-	dd/mm/yy	mm/dd/yy
Weight units	kilograms	kilograms	kilograms, pounds, or ounces
Height units	centimeters	centimeters	centimeters, inches, or feet

EUROPE uses the dd/mm/yy date format and allows the entry of only metric weight and height units. MEASURE uses the US date format; allows for weight information to be input as kilograms, pounds, ounces, or a combination of pounds and ounces; and allows height information to be input as centimeters, inches, feet, or a combination of feet and inches.

What the Anthropometric Calculation Program Does

DOMEAS, the program that performs the anthropometric calculations requires the following input from the Epi Info questionnaire:

SEX - coded as 1/M/m for males, 2/F/f for females
AGE - in months
WEIGHT - in kilograms
HEIGHT - in centimeters

The program performs the calculations and returns the following values:

	Percentiles	**Z-scores**	**Percent of Median**
Height-for-Age:	HAP	HAZ	HAM
Weight-for-Height:	WHP	WHZ	WHM
Weight-for-Age:	WAP	WAZ	WAM

A record FLAG, coded 0 to 7, described below.

The first nine fields contain the results of the anthropometric calculations. For the Z-scores, a code of 9.99 means that the index could not be calculated because of missing data or data values that were out of the appropriate range. An example of the latter would be an age of 18 years or older. A code of 9.98 for Z-scores denotes that the Z-score was greater than or equal to 9.98 and most likely indicates an error in measurement. For percentiles and percent of median, a similar coding scheme is used (99.9 and 99.8 for percentiles and 999.9 and 999.8 for percent of median, respectively).

A tenth field, the record FLAG field, is used to identify records where there are missing data points or a strong likelihood that some of the data items are incorrect (based on extreme Z-scores). The criteria for "flagging" an anthropometric index are as follows:

Index	Minimum	Maximum
HAZ	-6.00	+6.00
WHZ	-4.00	+6.00
WAZ	-6.00	+6.00

Two additional criteria for "flagging" a record are combinations of data items:

(HAZ > 3.09 and WHZ < -3.09) *or* (HAZ < -3.09 and WHZ > 3.09)

It is recommended that all "flagged" records be verified for accuracy. Common errors include incorrect data entry, incorrect age/dates, weight or height measurements

entered incorrectly or in the wrong units, and missing/blank data. When anthropometric data are being analyzed in the Epi Info ANALYSIS program or elsewhere, it is recommended that certain indices be set to missing (and therefore excluded from analyses) based on the coding in the FLAG field (Table 2). Note that when a Z-score is flagged, the corresponding percentiles and percent-of-median values are also flagged.

Table 2: Record flag coding scheme

Flag Code	HAZ	WHZ	WAZ	Notes
0				No indices flagged
1	Y			HAZ flagged only
2		Y		WHZ flagged only
3	Y	Y		Both HAZ and WHZ flagged
4			Y	WAZ flagged only
5	Y		Y	Both HAZ and WAZ flagged
6		Y	Y	Both WHZ and WAZ flagged
7	Y	Y	Y	All three indices flagged

Y=Index flagged, blank means index not flagged.

Interpretation of the flags is as follows:

Flag 0: This means that none of the indices were flagged. However, this does not necessarily mean the information is correct. Either sex, age, weight, or height could be incorrect but not extreme enough to be flagged.

Flag 1: HA is flagged but not WH or WA. This could be an extremely short or tall individual. Assure that the height information entered onto the computer file is correct. If height is incorrect, then WHZ would generally be close to -3.09 or 3.09 (a WHZ value beyond these would produce a flag error number 5). The other alternative is that the age information is incorrect, which would make the WAZ extreme (near -6 or 6).

Flag 2: WH is flagged but HA and WA are not. First, check the age and height of the child and make sure they are within the limits described in the section *Limitations of Growth Reference Curves*. If the child is within the age and height limitations, then either height or weight may be incorrect. If height is incorrect, then HAZ would be expected to be near an extreme value (but not extreme enough to be flagged), and if weight is incorrect, then WAZ would be close to an extreme value (but not extreme enough to be flagged). Finally, this could truly be an exremely thin or obese child.

Flag 3: HA and WH are both flagged but WA is not. This is an indicator that height

may be incorrect or missing.

Flag 4: WA is flagged but not HA or WH. If the weight is incorrect, then WHZ would be near an extreme value (but not extreme enough to be flagged), and if age is incorrect, then HAZ is likely to be near an extreme value (but not extreme enough to be flagged).

Flag 5: HA and WA are flagged but not WH. This is an indication that the age information is incorrect, missing, or out of range.

Flag 6: WH and WA are flagged but not HA. This is an indication that weight is likely to be incorrect or missing.

Flag 7: All three indices are flagged. This can occur if sex is unknown or incorrectly coded; or at least two of the following are missing, incorrectly coded, or beyond the limitation of the growth curve: age, weight, or height.

An example of an Epi Info coding scheme in the ANALYSIS program file AMISS1.PGM follows:

```
If (flag=1) or (flag=3) or (flag=5) or (flag=7) then HAZ = .
If (flag=1) or (flag=3) or (flag=5) or (flag=7) then HAP = .
If (flag=1) or (flag=3) or (flag=5) or (flag=7) then HAM = .
If (flag=2) or (flag=3) or (flag=6) or (flag=7) then WHZ = .
If (flag=2) or (flag=3) or (flag=6) or (flag=7) then WHP = .
If (flag=2) or (flag=3) or (flag=6) or (flag=7) then WHM = .
If (flag>3) and (flag<8) then WAZ = .
If (flag>3) and (flag<8) then WAP = .
If (flag>3) and (flag<8) then WAM = .
```

An equivalent way to create the missing variable codes without using the FLAG field that can be found in AMISS2.PGM is:

```
Define NOHA #
Define NOWH #
Define NOWA #
If (haz < -6) or (haz > 6) then NOHA = 1 else NOHA = 0
If waz < -6 or waz > 6 then NOWA = 1 else NOWA = 0
If whz < -4 or whz > 6 then NOWH = 1 else NOWH = 0
If ((haz>3.09) and (whz<-3.09)) or ((haz<-3.09) and (whz>3.09))\
 then NOHA=1
If ((haz>3.09) and (whz<-3.09)) or ((haz<-3.09) and (whz>3.09))\
 then NOWH=1
If NOHA then HAZ = .
If NOHA then HAP = .
If NOHA then HAM = .
If NOWA then WAZ = .
If NOWA then WAP = .
```

```
If NOWA then WAM = .
If NOWH then WHZ = .
If NOWH then WHP = .
If NOWH then WHM = .
```

This second method for creating missing variables can be modified to create your own missing-variable coding scheme. In severe famine situations, some users may want to allow for a lower minimum WHZ value. Other possible criteria for identifying records that are most likely in error are:

(HAZ > 3.09 and WAZ > 3.09) **OR** (HAZ < -3.09 and WAZ < -3.09)

(WHZ > 3.09 and WAZ > 3.09) **OR** (WHZ < -3.09 and WAZ < -3.09)

Other Considerations in Using ENTER for Anthropometry

As mentioned above, the subroutine that calculates anthropometry assumes that age is in months, sex is coded as 1/M/m for males and 2/F/f for females, weight is in kilograms, and height is in centimeters. However, with Epi Info, you can calculate age from two dates (i.e., date of birth and date of measurements) and convert US. measurements (e.g., inches and pounds) entered on the screen to metric within a .CHK file (described in more detail later in this chapter).

When possible, it is always preferable to have age calculated from the birth date and the date of the measurement(s). The reference curves are based on "biologic" age rather than calendar age. Biologic age in months divides the year into 12 equal segments as opposed to calendar age in which months have from 28 to 31 days. Although this makes little difference in older children, it can have an effect on the anthropometric calculations for infants. To calculate biologic age, the number of days between the two dates is calculated by the program. The age in days is divided by 365.25 and then multiplied by 12. Entering age by rounding to the nearest month and/or the most recently attained month can have a substantial effect on the anthropometric calculations, especially for infants.[4]

Examples of Standard Anthropometric Analysis and Graphs

Examples of standard analyses for anthropometric data are included as programs for ANALYSIS. AMISS1.PGM and AMISS2.PGM convert anthropometric values that are missing (coded as 99.99 or 999.99) or extreme values to missing values.

EPIANTH.PGM is an ANALYSIS program to calculate Z-score distributions, percentile distributions, the prevalence of low anthropometric values by age and sex,

and the distribution of Flag codes. To use this program, choose EXAMPLES from the EPI6 main menu and then NUTRITIONAL ANTHROPOMETRY and EPIANTH. You can also run EPIANTH from ANALYSIS by typing, from the EPI6> prompt:

EPI6>**RUN EPIANTH**

After two introductory screens, a menu will show the various types of analyses that can be performed. For more information on the EpiAnth program, you can read the documentation (option 92 of the EpiAnth menu) or the text file EPIANTH.TXT.

Another method of analyzing anthropometric data is to use the EPINUT program which is described later in this chapter.

How to Add Anthropometric Calculations to .QES/.REC/.CHK Files

Anthropometry is usually only one part of a survey. This section will describe how to incorporate anthropometric calculations into a questionnaire. In order to do this successfully, you must know how to create and restructure .REC files and how to use .CHK files. The way anthropometric values are calculated within a questionnaire is somewhat complex. Because the program that calculates the anthropometry is a fairly large program itself, a special system has been devised in Epi Info to allow the ENTER program to call other programs to perform complex calculations. If this method for calculating anthropometry appears to be too complex, an alternative method would be to use the program EPINUT, described later in this chapter.

In addition to the .QES, .REC, and .CHK files, several other files are necessary to be able to calculate anthropometric indices, including a batch (.BAT) file. The other necessary files are:

DOMEAS.EXE	The file that calculates the anthropometry
ENTER.EXE	The Epi Info ENTER program for data entry
ENTFACE.BIN	A program that interfaces ENTER and DOMEAS
MARK.COM	A program to load DOMEAS into memory
REMOVE.COM	A program to remove DOMEAS from memory

(DOMEAS.PAS, ENTFACE.ASM, ENTFACE.PAS, MEASURE.PAS contain source code for the corresponding modules.)

The batch file NUTRI.BAT contains the following commands:

```
MARK
DOMEAS
ENTER NUTRI
REMOVE
```

The first calls the command MARK which *marks* an area of memory in which to load the next file, DOMEAS. Next, the ENTER program is called along with the .REC file NUTRI. After exiting the ENTER program, the command REMOVE is called to *remove* DOMEAS from memory.

Anthropometric calculations in Epi Info rely on field names. Fields having the following names *must* be in the .QES/.REC file:

	FIELD	**PURPOSE**
Input Fields		
	SEX	Coded as 1/M/m or 2/F/f
	AGE	Age *in months*
	WEIGHT	Weight *in kilograms*
	HEIGHT	Height *in centimeters*
Calculated Fields		
	HAP	Height-for-age percentile
	HAZ	Height-for-age Z-score
	HAM	Height-for-age percent of median
	WHP	Weight-for-height percentile
	WHZ	Weight-for-height Z-score
	WHM	Weight-for-height percent of median
	WAP	Weight-for-age percentile
	WAZ	Weight-for-age Z-score
	WAM	Weight-for-age percent of median
	FLAG	Record flag for extreme or missing information

See the NUTRI.QES, EUROPE.QES, or MEASURE.QES file for examples of questionnaires with anthropometry information. The final important detail is the check (.CHK) file. Within the .CHK file there must be commands to "call" the necessary programs to perform the anthropometric calculations. For example, in the NUTRI.CHK file, under the codes referring to HEIGHT is the following information:

```
HEIGHT
  if age = .
  then
    age = 999.99
  endif
  Range 0 +inf
  LOAD ENTFACE.BIN
  CALL ENTFACE 75
  if age = 999.99
  then
    age = .
  endif
```

```
     UPDATE
     GOTO WRITEREC
   END
```

The first "if" statement changes ages that are unknown (.) to 999.99. The range for age in months is from 0 to +inf. The next two commands, LOAD ENTFACE.BIN and CALL ENTFACE 75, are the commands that set up the DOMEAS program for the anthropometric calculations. The calculations are returned to the ENTER program and presented to the screen (through use of the UPDATE command). The GOTO WRITEREC command presents the "Write data to disk (Y/N)?" command at the bottom of the screen.

Also in the NUTRI.CHK file is the calculated field information with the NOENTER command:

```
   HAP
     NoEnter
   End
```

The NoEnter command prevents users from editing the calculated fields. The EUROPE.CHK file is more complicated because it includes the ability to calculate age from two dates (the date of measurements and the date of birth) and allows for age to be input in years (which is converted to age in months). Even more complex is the MEASURE.CHK file which can also calculate age from the two dates, and also converts from US measurement to metric units.

In MEASURE, weight can be entered in pounds or ounces. To use the anthropometric subroutine, the total number of pounds is calculated and converted to kilograms within a .CHK file as:

   Kilograms = (pounds + ounces/16)* 0.45359237

Height can be entered in feet or inches. The total number of inches is calculated and then converted to centimeters within a .CHK file as:

   Centimeters = (feet * 12 + inches) * 2.54

In summary, to add the ability to calculate anthropometry in ENTER, the following steps need to be followed:

1. Make sure the field names and their description in the .QES/.REC files are the same as described previously
2. Make sure the .CHK file is set up correctly (e.g., LOAD ENTFACE, etc.)
3. Create a batch (.BAT) file with the necessary information

4. Call the .BAT file from the DOS prompt

Anthropometric Calculations Using EPINUT

EPINUT is a program for performing calculations with anthropometric data in Epi Info files and for displaying summary statistics from the data. To use EPINUT, you must have a .REC file with the relevant information already entered (e.g., sex, age, weight, and height).

EPINUT can be used to add nutrition indices to an existing Epi Info data file containing these variables, or to display frequency distribution tables and graphs for each nutrition index.

Adding Nutrition Indices To A File Using EPINUT

EPINUT can add anthropometric indices to an Epi Info file that already contains data in fields called AGE, SEX, WEIGHT, and HEIGHT.

AGE should represent biologic age, in months.
SEX can be coded as "1", "m", or "M" for boy and "2", "f", or "F" for girls.
WEIGHT and HEIGHT should be numeric, in metric units.

When all indices are calculated, 10 new variables are added to the file: HAZ, HAM, HAP, WAZ, WAM, WAP, WHZ, WHM, WHP, and FLAG. Any existing variable with one of these names will be overwritten and replaced by the newly calculated value. Refer to the previous section "What the Anthropometric Calculation Program Does" for further descriptions of the indices.

EPINUT will also summarize the new variables and compare them to international reference standards, producing tables and graphs.

Using EPINUT Interactively

EPINUT has pulldown menus and can be operated with a mouse or from the keyboard. To use EPINUT interactively, choose it from the PROGRAM portion of the EPI6 menu, or, from the DOS command line, type:

C:\>**EPINUT**

In the NUTRITIONAL INDICES menu, ADD INDICES TO A FILE allows suitable

variables (HAZ, HAM, HAP, WAZ, WAM, WAP, WHZ, WHM, WHP, and FLAG) to be added to a file. Variables coding for age, sex, weight and height can be individually selected from a list of the available variables in the file, and then the calculations are performed and the results placed in the file. The results can be restricted to certain indices only. Cutoff values for flagging records can be altered, and flagged records can be ignored or included.

Sometimes it is tempting to put several sets of nutritional measurements in a single file, such as indices at admission and at discharge from a renutrition unit, or longitudinal follow-up of children in a clinic. From a data management perspective, however, it is almost always preferable to use a relational database structure, keeping in a core record all information on individual children, and in a related data file the variables specific to each visit and the information necessary for index calculation, such as the age, sex, weight and height. Thus, index calculation takes place on the related data file where only one set of indices is stored in each record. If several sets of observations are really needed in the same file, Eped or Analysis can be used to rename existing index variables prior to running EPINUT for a new set.

The menu item, ANALYZE INDICES FROM A FILE, provides tools for building frequency distributions and graphs and calculating malnutrition indicators. Information on AGE and SEX is needed. EPINUT assumes that nutritional indices are stored in variables called HAZ, HAM, HAP, WAZ, WAM, WAP, WHZ, WHM, WHP, and FLAG. Analysis can be restricted to a specific age range. The process skips flagged records by default.

Additional analysis options include taking into account a cluster sampling design effect if a variable identifying clusters is indicated, and the presence of oedema in the calculation of nutrition indicators if a variable coding for the presence of oedema is indicated. In this case, children with oedema will not be incorporated in the distribution for weight/height and weight/age indices, since oedema may mask a loss of weight. Cutoff values for the various indicators of malnutrition can be specified.

The output includes descriptive information on each index frequency distribution, such as the mean, median, standard deviation, skewness and kurtosis of the distribution, the proportion of subjects below predefined cutoff values, and the standardized prevalence. Malnutrition indicators are displayed with their confidence interval (Fleiss quadratic method), taking into account the design effect if indicated.

Frequency distributions can be saved to a file or printed. Graphs can be exported in ASCII delimited format for further use in graphic packages.

CALCULATE INDICES provides a nutritional calculator. Enter age, sex, weight and height information, then press calculate to get the nutrition indices. The age button allow entry of a birthdate and a visit date for calculation of biologic age.

The SURVEY UTILITIES menu provides several utilities that are useful in nutritional surveys. These include a random-number generator for random selection of subjects in field surveys, a sample size calculator that can incorporate design effect, and a calculator of standardized prevalence.

Running EPINUT From the DOS Command Line

Like other Epi Info programs, EPINUT can be run from the DOS command line so that it may be run from DOS .BAT files or other programs as an integral part of other systems. The simplest form of command-line processing is:

EPINUT <FILENAME.REC>

EPINUT will add nutrition indexes to the file <FILENAME.REC> if the appropriate variables were found. If the variable names are non standard, they may be designated interactively, but can also be changed using additional command line parameters:

Typing

EPINUT MYFILE.REC /A=MYAGE /S=MYSEX /H=MYHEIGHT /W=MYWEIGHT (on 1 line)

will add nutrition indices to the file, looking for relevant information in the variables MYAGE, MYSEX, MYHEIGHT, and MYWEIGHT, rather than AGE, SEX, HEIGHT, and WEIGHT.

Comparison of Questionnaire vs. EPINUT Calculation of Anthropometric Indices

Which method should you use to calculate anthropometric indices? The ENTER program calculates the indices record by record from the questionnaire; EPINUT performs the calculations in a batch processing mode. Which method to use depends on a number of factors.

If you already have sex, age, weight, and height data entered into an Epi Info .REC file (or can import the data into a .REC file), then use EPINUT for the calculations. If the data have not yet been entered, you can use either method depending on your needs. Using ENTER for calculations is preferable if:

• There is a need to calculate and use the anthropometry once data are entered, as in a

283

clinic situation.

• In a survey, having calculations for each record is useful for rapid detection of data entry and data collection errors.

EPINUT is preferable if:

• The sex, age, weight, and height data have already been entered onto computer; only the anthropometry needs to be calculated

• It has been decided to check the data for entry and collection errors related to the anthropometry after all the data have been entered
• The method for performing the calculations in ENTER is too complex.

Other Anthropometric Software and Sources of Information

The anthropometry module in Epi Info and EPINUT is a replacement for CASP (CDC Anthropometric Software Package), a program previously distributed and supported by CDC and WHO. The anthropometric results in Epi Info and EPINUT are identical to those in CASP.

Another program jointly developed by CDC and WHO is ANTHRO. ANTHRO is a program that performs batch processing of records from dBase data files, although data can be imported to a dBase file structure within ANTHRO from either ASCII or comma-delimited formats.

For persons currently using the CASP "Entry" program, a utility program called ETBCONV converts.ETB files to Epi Info data files (.ETB is the file extension used for CASP data files). ETBCONV can also convert ETB files to dBase, ASCII, or comma-delimited files. To obtain a copy of the ETBCONV program, please contact the Division of Nutrition, CDC at the address shown below.

For additional information on the use and interpretation of anthropometry, please refer to the articles by: WHO Working Group on the Purpose, Use, and Interpretation of Anthropometric Indicators of Nutritional Status;[2] Gorstein et al.,[3] Dibley et al.;[5] Waterlow et al.,[6] and Beaton et al.[7] To obtain additional information on anthropometric software, please contact one of the following:

<div align="center">

Division of Nutrition
Center for Chronic Disease Prevention and Health Promotion
Centers for Disease Control and Prevention
1600 Clifton Road NE, MS A08
Atlanta, GA 30333 U.S.A.

</div>

Nutrition Unit
World Health Organization
1211 Geneva 27
Switzerland

Frequently Asked Questions Concerning Anthropometry: Troubleshooting

Q. When I use the NUTRI.REC file to calculate anthropometry, after entering sex, age, weight, and height, no calculations are performed.

A. To use the NUTRI files, you must run the NUTRI batch file from the DOS prompt. If you go directly into ENTER and provide the .REC file name NUTRI, EUROPE, or MEASURE, you are bypassing the loading of two important programs needed for the calculation of anthropometry.

Q. When I enter NUTRI at the DOS prompt I get the statement "Bad command or file name."

A. Make sure the anthropometry module was loaded when Epi Info version 6 was installed on the computer. To see if they were installed, see if the files NUTRI.QES, NUTRI.BAT, NUTRI.REC, and NUTRI.CHK are in the same subdirectory as the Epi Info files. If there are not there, they will need to be installed from the Epi Info disks. Also, when you enter NUTRI at the DOS prompt, make sure you are in the same directory as the Epi Info files (usually the EPI6 subdirectory).

Q. When I enter NUTRI at the DOS prompt, I am left in ENTER with the prompt waiting for the name of a .REC file.

A. Make sure you are in the same subdirectory as the Epi Info 6 files when you type NUTRI at the DOS prompt.

Q. For some of the participants in my survey, the anthropometry was not calculated (i.e., many of the anthropometric fields are "99.99").

A. Check the limitations of the growth curves. Weight-for-height has the most restrictions because of the upper age limitation of 10 years for females and 11.5 years for males (in addition to height limitations). Weight-for-height also has a length minimum of 49 cms. Height-for-age and weight-for-age can be calculated for children from birth to their 18th birthday. Make sure that sex is coded correctly; if sex is unknown, the anthropometry will not be calculated.

Frequently Asked Questions Concerning Anthropometry: More Advanced Techniques

Q. I want to calculate other anthropometric indices, such as head circumference, arm circumference, tricep skinfold thickness, and body mass index.

A. There is a head circumference reference used in the US. However, it is not available in Epi Info. There is a FORTRAN program called FORANTH that performs batch processing on IBM compatible computers available from the Division of Nutrition, CDC.   Arm circumference does not require any special program because simple cutoffs are used: "adequately nourished:" $\geq$ 13.5 cm; "severely malnourished:" 12.5-13.4 cm; and "critically" malnourished: <12.5 cm. For tricep skinfold thickness and body mass index, there are no internationally recognized references for children.

Q. Instead of calling NUTRI from the DOS prompt, why not create a batch file that will load the DOMEAS program every time you start Epi Info?

A. You could create a batch file that would load the MARK and DOMEAS commands every time you start Epi Info. Then you would not need to call NUTRI, EUROPE, or MEASURE from the DOS prompt; they would work directly from within the ENTER module of Epi Info. The reason not to load these files into memory every time you start Epi Info is that they take up around 55K of RAM.

Q. I have created the .QES/.REC/.CHK files for my nutrition survey that utilize the anthropometry calculations. This information will be entered in many different sites on different computers, many of which may not have Epi Info on them. Which files do I have to distribute to make the system work correctly?

A. The files needed are your .REC and .CHK files along with the following files: ENTER.EXE, ENTER.OVR, DOMEAS.EXE, ENTFACE.BIN, MARK.COM, and REMOVE.COM. Be sure to install all of the files in the same directory (or, if installed in separate directories, in the .BAT file be sure to put the paths as to where the files are located). A DOS .BAT file similar to MEASURE is also desirable.

Q. I have my data in a SAS file and just need to have the anthropometry calculated. What is the best method for doing this?

A. If you have data in a format other than an Epi Info file, there are a number of possibilities. First, you could program the anthropometry within the program you are using (such as SAS or SPSS). However, the programming is complex and takes around 600 lines of code; therefore, this approach is not recommended. Another possibility is to get the data into an Epi Info format. Epi Info can import dBase, Lotus, fixed column, and comma-delimited files. A commercial program called DBMS

COPY converts files from many different statistics, database, and spreadsheet programs, including Epi Info files. Once the data are in an Epi Info .REC file, you could use EPINUT to perform a batch processing of the file. Alternatively, if the data are in the dBase format or can be easily converted to a dBase file, you could use the program called ANTHRO available from CDC and WHO. ANTHRO performs a batch processing with dBase files.

Q. I want to follow children longitudinally and calculate their anthropometric indices. How can this be done?

A. Most statistical programs for longitudinal data analysis expect each visit to be in a separate record with a field in each record that identifies each individual. The simplest method is to create a separate record for each visit, making sure there is an identification field where each individual's unique ID number is entered. A more complex method is to use the RELATE functions to RELATE two .REC files. One .Rec file would contain information on each patient (such as ID number, name, sex, date of birth, , etc.) and the other file would have visit information (ID number, date of measurement, weight, and height). Chapter 25 tells how to RELATE files.

Acknowledgments

References

1. Dibley MJ, Goldsby JB, Staehling NW, Trowbridge FL. Development of normalized curves for the international growth reference: historical and technical considerations. Am J Clin Nutr 1987;46:736-48.

2. WHO working group. Use and interpretation of anthropometric indicators of nutritional status. Bull WHO 1986;64:929-41.

3. Gorstein J, Sullivan K, Yip R, de Onis M, Trowbridge F, Fajans P, Clugston G. Issues in the assessment of nutrition status using anthropometry. Bull WHO 1994 (in press).

4. Gorstein J. Assessment of nutritional status: effects of different methods to determine age on the classification of undernutrition. Bull WHO 1989;67:143-50.

5. Dibley MJ, Staehling N, Nieburg P, Trowbridge FL. Interpretation of Z-score anthropometric indicators derived from the international growth reference. Am J Clin Nutr 1987;46:749-62.

6. Waterlow JC, Buzina R, Keller W, Land JM, Nichaman MZ, Tanner JM. The presentation and use of height and weight data for comparing the nutritional status of groups of children under the age of 10 years. Bull WHO 1977;55:489-98.

7. Beaton G, Kelly A, Kevany J, Martorell R, Mason J. Appropriate uses of anthropometric indices in children: a report basd on an ACC/SCN workshop. ACC/SCN Nutrition Policy Discussion Paper no. 7, 1990.

```
Dataset:  C:\EPI6\OSWEGO.REC (75 records)          Free memory: 294K
Criteria: All records selected                     Time:    0.11 sec
─────────────────────────── Output - Screen ───────────────────────

=========> read oswego

=========> report oswego.rpt
                        ATTACK RATES BY FOOD EATEN

FOOD EATEN        TOTAL       ATTACK RATES
                  EXPOSED     EXPOSED      UNEXPOSED

──────────        ─────       ───────      ─────────
BAKED HAM          46          63%           59%
ROLLS              37          57%           66%
VANILLA
  ICE CREAM        54          80%           14%




─────────────────────────────────────────────────────
EPI> read oswego
EPI> report oswego.rpt
EPI>
F1-Help  F2-Commands  F3-Variables  F4-Browse  F5-Printer on  F9-DOS  F10-Quit
```

```
F1-Hlp F2-File F3-Epiaid F4-Txt F5-Print F6-Set F7-Find F8-Blk F9-Save F10-Done
#USES   BAKEDHAM ILL
                        ATTACK RATES BY FOOD EATEN

FOOD EATEN        TOTAL       ATTACK RATES
                  EXPOSED     EXPOSED      UNEXPOSED
──────────        ─────       ───────      ─────────
BAKED HAM         [Y,]        {100*[Y,Y]/[Y,]}:2%          {100*[N,Y]/[N,]}:2%
#USES ROLLS ILL
ROLLS             [Y,]        {100*[Y,Y]/[Y,]}:2%          {100*[N,Y]/[N,]}:2%
#USES VANILLA ILL
VANILLA
  ICE CREAM       [Y,]        {100*[Y,Y]/[Y,]}:2%          {100*[N,Y]/[N,]}:2%












1  C:\EPI6\OSWEGO.RPT                 EPED  319656      L1   C1  TXTInsInd
```

290

Chapter **24**

Precise Control of Tables: The REPORT Command

<div style="border:1px solid">

Tutorial

Run ANALYSIS from the EPI6 menu.

Give the commands:

READ DISEASE
REPORT DISEASE.RPT

Observe the report that is produced. To see the report file that produced it, use TYPE DISEASE.RPT. You may have to press the <PgUp> key to see the whole file.

Try:

READ OSWEGO
REPORT OSWEGO

Then examine OSWEGO.RPT with the TYPE command.

Press <F10> and return to the main menu. Then choose EPED and open the file \ NETSS\ TABLE1.RPT with the <F2> menu and the Open File command. This is a good example of a report file in tabular form.

Experiment with the commands in report files and begin writing your own reports.

</div>

Overview

For some purposes, it is useful to control the printing of a table so that only some of the values are printed or they are printed in a different order from those that would be produced by the TABLES command. You may wish to intersperse textual explanations of the values with the tabular results. This is the purpose of the REPORT command in ANALYSIS. It allows customization of surveillance reports so that they conform to a predefined newsletter format, for example, or summarization of tables in narrative, rather than tabular, form.

Before using REPORT, it is a good idea to explore other means of altering the format of the tables produced by ANALYSIS. You can influence the content of tables by using the commands **SET STATISTICS = OFF (**or **ON)**, **SET PERCENTS=ON (**or **OFF)**, **SET MAXCELLWIDTH=**<Integer>, **SET CRITERIA = OFF (**or **ON), SET DECIMALS=**<Integer>, and **SET LINES= OFF** (or **ON**). Further details are given in Chapter 36.

Multiple tables can be produced through the use of stratifying variables. If you wish to produce a table of disease by race for each county, for example, the command TABLES DISEASE RACE COUNTY, combined with imaginative use of the TITLE command, may be all that is required.

To use the REPORT command, you first use EPED or another text editor to make a file describing the report. It serves as a template to tell ANALYSIS how to print the report, much as a .QES file tells ENTER how to construct a data file. Report files contain a list of variables to be tabulated, text that is to appear in the report, and commands that steer ANALYSIS to place the contents of any cell in the specified set of tables in the report. They are made in EPED and usually, but not necessarily, have the file name extension .RPT.

A Simple Report

The file DISEASE.REC will be used to demonstrate features of the REPORT generator. The file has the following structure:

```
Name        Type    Len   Name        Type    Len   Name        Type    Len
----------  ------- ---   ----------  ------- ---   ----------  ------- ---
ID          Real     5    NAME        Alpha    13   AGE         Integer  3
SEX         ALPHA    1    RACE        ALPHA     1   DOB         Date     8
DISEASECOD  Integer  3    DISEASENAM  ALPHA    18   DATEREPORT  Date     8
```

LIST shows the following data items:

```
REC  ID    NAME            AGE SEX RACE DOB       DISEASECOD DISEASENAM           DA
---  ----- -------------   --- --- ---- -------   ---------- ------------------   --
  1    1 Heidi Earlich     21 F    W   08/26/69        10 HEPATITIS           01
  2    2 Jeffrey Burr      21 M    W   07/23/68       100 SALMONELLA          01
  3    3 Tony Leighton     35 M    B   01/02/54        99 AIDS                01
  4    4 Andrew Burr       50 M    W   04/04/38        10 HEPATITIS           01
  5    5 Erika Earlich     19 F    W   .               99 AIDS                01
  6    6 Mona Smith        20 F    B   .               99 AIDS             01
  7    7 Robert Peters     33 M    B    07/14/52      100 SALMONELLA           01
  8    8 Jack Smith        33 F    B   01/01/57        10 HEPATITIS           01
```

The commands:

SET STATISTICS = OFF
TABLES DISEASENAM RACE SEX

produce the following tables:

```
        SEX = F
                      RACE
          DISEASENAM |   B     W | Total
------------------+-----------+------
AIDS              |   1     1 |     2
HEPATITIS         |   1     1 |     2
SALMONELLA        |   0     0 |     0
------------------+-----------+------
          Total |   2     2 |     4
        SEX = M
                      RACE
          DISEASENAM |   B     W | Total
------------------+-----------+------
AIDS              |   1     0 |     1
HEPATITIS         |   0     1 |     1
SALMONELLA        |   1     1 |     2
------------------+-----------+------
          Total |   2     2 |     4
```

Note that there is one table for males and another for females. Now suppose that you had access to any of the cells, FUNCTIONs, or marginal totals in these tables, and could arrange them on the screen any way you want. This is the idea behind the REPORT command. It makes a table or series of tables in the computer's

293

memory and provides a shorthand language that lets you transfer the information from the memory table to the screen or printer.

The specifications that you develop are in a report or .RPT file that you create in EPED. To produce a report, the command is simply REPORT <file name>.

The first item in any report file **must** be a description of the table that is to be created in memory. This is in the following form:

#USES <variable name> <variable name> etc.

The variable names are the same ones you would use for the TABLES command. A sample report file, DISEASE.RPT, is provided with Epi Info. The first half looks like this:

#USES DISEASENAM RACE SEX

This report demonstrates use of the REPORT command in ANALYSIS

\cOn @SystemDate
\cAmong a Total of [,,M]:2 Males

**   [HEPATITIS,,M] had hepatitis.**
** [SALMONELLA,,M] had Salmonellosis.**
**   [AIDS,,M] had AIDS.**

\cAmong a Total of [,,F]:2 Females,

**   [HEPATITIS,,F] had hepatitis.**
** [SALMONELLA,,F] had Salmonellosis.**
**   [AIDS,,F] had AIDS.**

The total number of cases was [,,]:2.
{[,,M]*100/[,,]} :3% of the cases were male.
{[,B,]*100/[,,]} :3% of the cases were black.

The line of text after the #USES statement is printed as is; the next line inserts the current date in place of @SystemDate and centers the expression on the screen. The important elements of the report are the items in square brackets that indicate values of cells in the table specified by #USES. Examine the #USES statement in conjunction with one of the expressions in square brackets.

#USES DISEASENAM RACE SEX

\cAmong a Total of [,,M]:2 Males

The brackets tell ANALYSIS to put a value from the table on the screen at the point

where the brackets appear. \C means center the line on the screen, and the other items are just text to be put on the screen as it appears, except for the :2 that indicates there are to be two digits in the table value indicated by the brackets.

Since there are three variable names in the #USES statement, there must be three items within each set of parentheses that describe a value in the table. They must be in the same order as they are in the #USES statement. Here the expression [,,M] indicates the cell in the table containing the number of males, without regard to DISEASENAM or RACE, since these are not specified. If the expression had been [HEPATITIS, W, M], the number of white males with hepatitis would be supplied. Note that for text variables, upper and lower case make a difference, and **Hepatitis** is not the same as **HEPATITIS**; the case must match that of the items in the data file.

The rest of the report is constructed of the same elements except for the calculation of the percentage of males in the curly brackets. The curly brackets indicate a calculation, and this one says, "Multiply the number of males by 100 and divide by the total number of all cases; make the result a three-digit number." The % sign that follows the 3 is treated as text and merely printed after the result.

Totals are produced by specifying only the values that describe the total. Thus the total number of people with hepatitis who are male is [HEPATITIS,,M] and the grand total of all cases regardless of disease, race, or sex, is [,,] or [ ].

Now use the REPORT command in analysis to produce the report as follows:

    READ DISEASE
    REPORT DISEASE.RPT

The results should be as follows:

This report demonstrates use of the REPORT command in ANALYSIS

```
                          On 01/10/90
                    Among a Total of  4 Males


          1 had hepatitis.
          2 had Salmonellosis.
          1 had AIDS.

                    Among a Total of  4 Females,

          2 had hepatitis.
          0 had Salmonellosis.
          2 had AIDS.
```

```
The total number of cases was  8.
 50% of the cases were male.
 50% of the cases were black.
```

The #FOREACH/#ENDFOR Command

For large tables, it would not be efficient to specify all the values in the format given, especially if several variables are included in each set of brackets. There is a shorthand method of specifying reports that gives much more power and generality to report specification, although for non programmers, it may take a bit of trial and error to understand.

Several more definitions are needed to understand the next report:

^SEX   means the *Value Label* of SEX, e.g., "M".

[,,^SEX]   means the *Count* for the current SEX.

#foreach SEX   means "print all the value labels for SEX (i.e., **F** and **M**,)
  ^SEX \   for each value of SEX." The " \ " after "^SEX" means
#endfor   "Print three spaces and do not do a carriage return before
            the next value, putting the values on the same line."
            Without the "\ ", the values would be printed on separate
            lines.

#newpage   means skip to the next page.

The report written in this format is the second part of the file DISEASE.RPT.

#USES DISEASENAM RACE SEX

Note: The following is a similar report produced by using the #FOREACH command. If SEX had been replaced by COUNTY, for example, the report might be much longer than the .RPT file that produced it, although we have used a variable with only two values for illustrative purposes.

```
\cOn @SystemDate
#FOREACH SEX
\cAmong a Total of [,,^SEX]:2 of the ^SEX sex

    [HEPATITIS,,^SEX] had hepatitis.
  [SALMONELLA,,^SEX] had Salmonellosis.
        [AIDS,,^SEX] had AIDS.
```

```
#ENDFOR

The total number of cases was [,,]:2.
{[,,M]*100/[,,]}:3% of the cases were male.
{[,B,]*100/[,,]}:3% of the cases were black.
```

Macros, #IF/#ENDIF, and a More Complex Report

A *macro* is a variable name that represents a longer expression in the report file. For example, if you get tired of typing [^race,^sex,^diseasenam]:6 repeatedly or if it takes up too much room for the spacing available on your report page, you can define a macro called CELLNM1 with the following command:

#CELLNM1 [^race,^sex,^diseasenam]:6

Wherever you would use the expression [^race,^sex,^diseasenam]:6, you refer to it by its macro name, #CELLNM1.

#IF/#ENDIF commands are used to control execution of other statements based on special criteria. They could, for example, be used to suppress printing or print a special character when a cell value is less than 5 to preserve confidentiality. In the report that follows, they are used to suppress printing of items with zero values.

The use of macros and of #FOREACH commands "nested" in several layers is also illustrated by the following report, supplied with Epi Info as DISEASE2.RPT. The comments on the right are not part of the file.

```
#uses diseasenam                                     First USES statement
                    Disease Report – Totals          Title for report
                                                     Skip a line
  In the current sample, there were:                 A second title

#foreach diseasenam                                  Execute for each disease.
          [^diseasenam] cases of ^diseasenam  ^ says do for each value of.
#endfor                                              End instructions for disease.


#uses  race sex diseasenam                           Second USES statement
#cellnm1 [^race,^sex,^diseasenam]:6                  MACRO - can be used anywhere

             Disease  Report – Demographics          First title
                   as of @systemdate                 Second title
```

```
#foreach diseasenam                              Do for all values of disease.
              Disease =  ^diseasenam             Print disease name here.
#foreach sex                                     Do for all values of sex.
 sex = ^sex                                      Print sex name here.
 total [,^sex]                                   Print total freq. for sex.

#foreach race                                    Do for all values of race.
#if #cellnm1 > 0                                 MACRO & IF suppresses zeroes.
 race = ^race:4                                  Print race label here.
 total [^race]                                   Print total freq. for race.
 For ^diseasenam                                 Do for all values of disease.
 total for sex= ^sex and race= ^race is #cellnm1 MACRO see above for value.
#endif                                           Closes instruction for IF.
#endfor                                          Closes instruction for race.
#endfor                                          Closes instruction for sex.
#endfor                                          Closes instruction  disease.
#newpage                                         Clears report in printer.
```

The major additions in this report are 1) the use of macros and 2) the use of IF statements. The salmonellosis portion of the report shows a 0 for sex = F. If you wanted to suppress this from the report you would add a #IF statement: **#IF #cellnm1 > 0** under the **#foreach sex** line. The > 0 refers to the quantity displayed in the cell from the table race sex diseasenam.

The report produced by REPORT DISEASE2.RPT is:

```
                    Disease Report - Totals

  In the current sample, there were:
            3 cases of AIDS
            3 cases of HEPATITIS
            2 cases of SALMONELLA

                 Disease Report - Demographics
                      as of 11/20/89


                 Disease =  AIDS
 sex = F
 total       2

 race = B
 total       1
 For AIDS
 total for sex= F and race= B is       1

 race = W
 total       1
 For AIDS
```

298

```
total for sex= F and race= W is       1


sex = M
total       1

race = B
total       1
For AIDS
total for sex= M and race= B is       1

                        Disease =   HEPATITIS
sex = F
total       2

race = B
total       1
For HEPATITIS
total for sex= F and race= B is       1

race = W
total       1
For HEPATITIS
total for sex= F and race= W is       1


sex = M
total       1


race = W
total       1
For HEPATITIS
total for sex= M and race= W is       1
                        Disease =   SALMONELLA
sex = F
total       0


sex = M
total       2

race = B
total       1
For SALMONELLA
total for sex= M and race= B is       1

race = W
total       1
For SALMONELLA
total for sex= M and race= W is       1
```

Reports Using Aggregate Data

The #USESSUM statement specifies that sums rather than counts are to be used in each cell of the table. This is useful when you are dealing with aggregate data in records, and want the table to represent the sum of the aggregates. The table produced is analogous to that produced by SUMTABLES, rather than TABLES. Large report files called TABLE1.RPT and TABLE2.RPT that handle aggregate data are included on the Epi Info disks in the directory called \ NETSS. They are part of the surveillance system described in Chapter 26. They also illustrate the construction of a large table with the REPORT command rather than the text-oriented reports we have used for illustration in this chapter.

Multiple #USES Commands To Produce Foodborne Attack Rates

The #USES statement may appear more than once in the same .RPT file. Thus, if you wish to produce a report that deals with the frequencies of foods eaten at a particular meal, the efficient way to produce the tables is to do a frequency for a food, present the information, and then do a frequency for the next food. The alternative, specifying all the foods in one #USES command, would produce a huge multidimensional table, most of which would not be needed. An efficient way to do this report, supplied as OSWEGO.RPT, is:

```
#USES  BAKEDHAM ILL
                          ATTACK RATES BY FOOD EATEN


FOOD EATEN        TOTAL        ATTACK RATES
                  EXPOSED    EXPOSED      UNEXPOSED
----------        -----      -------      ---------
BAKED HAM         [Y,]     { 100*[Y,Y]/[Y,]} :2%     { 100*[N,Y]/[N,]} :2%
#USES ROLLS ILL
ROLLS             [Y,]     { 100*[Y,Y]/[Y,]} :2%     { 100*[N,Y]/[N,]} :2%
#USES VANILLA ILL
VANILLA
  ICE CREAM       [Y,]     { 100*[Y,Y]/[Y,]} :2%     { 100*[N,Y]/[N,]} :2%
```

To produce the report use the commands:

READ OSWEGO
REPORT OSWEGO.RPT

The results are:

```
                     ATTACK RATES BY FOOD EATEN

        FOOD EATEN         TOTAL          ATTACK RATES
                          EXPOSED     EXPOSED     UNEXPOSED

        ----------        -----       -------     ---------
        BAKED HAM            46         63%          59%
        ROLLS                37         57%          66%
        VANILLA
          ICE CREAM          54         80%          14%
```

Conclusion and Related Topics in Other Chapters

Although at first the report generator may appear complicated, its concepts are quite simple, and a little experimentation can lead to better looking and more appropriate reports for permanent systems in which appearance is worth the programming time.

Information on the SET command and other means of tailoring tabular output appears in Chapter 36, the reference chapter for ANALYSIS. A list of commands that can be used in report files is given under REPORT in the same chapter. Examples of .RPT files are contained in the surveillance system described in Chapter 26.

(Blank for pagination compatible with Version 6.02 Manual)

(Blank for pagination compatible with Version 6.02 Manual)

(Blank for pagination compatible with Version 6.02 Manual)

(Blank for pagination compatible with Version 6.02 Manual)

Chapter **25**

Linking Files Together: Relational Features of ANALYSIS, MERGE, and ENTER

Tutorial
Run ENTER from the EPI6 menu.

Enter the file name HOUSE.

A household questionnaire will appear. The HouseID number is supplied automatically. Answer the other questions. Note that when you have completed "Running Water," the screen changes to a new form, called Person Record, and that HouseID has been copied automatically to this form. PersonID is supplied.

Answer the questions for this person. After ILL, the form changes again to a Visit record containing a copy of the PersonID number. Complete the first Visit record and then answer "Y" to "Write data to disk?"

Enter several other Visit records for this person. When you have completed them, press <F10> to return to the Person record. Save the Person record and enter a record for the next person in the same household. Note that you are given a chance to enter several Visit records for each person, returning to the Person record after pressing <F10>. When you have entered all the persons in the household, press <F10> at the end of the Person record, and you will return to the House record. Save the House record. You have completed the entries for this household. Press <F10> to return to the EPI6 menu.

Now run ANALYSIS from the EPI6 menu. Give the commands:

READ VISIT
RELATE PERSONID PERSON
RELATE HOUSEID HOUSE

Type VARIABLES to see the variables that are available in the related files. Now examine the data:

SORT VISIT
LIST VISIT NAME FINDINGS
LIST VISIT NAME ADDRESS
(One field from each file)

Try using TABLES to obtain more information. READ one of the related files and see that it behaves just as a single file normally behaves in ANALYSIS•• so that you have a choice of single or multi-file analytic approaches.

READ HOUSE
FREQ WATER

is a good example. How will the frequency of WATER differ from those obtained when the files are related as above?

Overview

Up to this point, we have described the use of Epi Info with a single file at a time or with several files of identical structure being analyzed as one. This is adequate for most statistical purposes, and statistical programs generally do not offer facilities for working with files of different structures at the same time.

Database programs, such as dBASE, are designed to join several files together temporarily so that items in one file can be cross tabulated with those in another. Thus, multiple follow-up visits for one person can be related to information on the person in another file, and a third file might be maintained with information on the person's household. Epi Info is able to RELATE several files in this way, matching records by common identification numbers. We will first show how multi-file data structures can be used and then give details for setting up and manipulating such structures in the ENTER, ANALYSIS, and MERGE programs.

Why Use Relational or Hierarchical File Structures?

Relational or hierarchical file structures in computer terminology allow linkage between records in different data sets (files in Epi Info). They allow variable numbers of records to be linked to a single "core" record.

Biology, medicine, and epidemiology abound with examples of hierarchies. Atoms and molecules; cells, organs, and individuals; and persons, families, and communities are just a few. In this chapter, we will work with a few examples from epidemiologic work, although books on database management usually use examples from the business world, such as:

> **Personnel Management**
> > Person pay periods, persons, departments, companies

> **Inventory**
> > Parts, products, product lines

Epidemiologic examples are:

> **Follow-up studies or clinical records**
> > Visits, persons, households

> **Immunizations**
> > Immunizations, persons, families, or schools

> **Disease Surveillance Systems**
> > Persons, Hepatitis Forms, Enteric Disease Forms, etc.
> > Cases, County Populations   (for rate calculations)

> **Hospital Nursery Infection Studies** (in which one or several babies
> > are assigned to a particular nurse for each shift)
> > Nurse-Baby-Shifts, Nurse information, Baby information

Theoretically, information on each of these hierarchies can be handled in a single questionnaire. If you are surveying families for health status, for example, the questionnaire could contain:

> > Family and Household Information
> > Individual Information (repeated 12 times to accommodate large
> > > families)

In a paper questionnaire, this structure may be a convenient way to collect data. The questionnaire may have hundreds of entry blanks. For most families, many

will be left empty, but paper is inexpensive, and the eye easily skips over the blank areas.

Processing such forms on a computer has two drawbacks. One is that the hundreds of blanks take considerable disk storage space even though they are left empty. For a few hundred records, this may not be a problem, but with larger studies or surveillance systems, disk space may be important. The second problem is more serious. Suppose that a questionnaire contains information for up to 12 children in a household. We will deal with only one item, AGE, here as an example.

Household **21**   Mother's age   **34**

Child1	age	**1**	Child7	age _
Child2	age	**6**	Child8	age _
Child3	age	**3**	Child9	age _
Child4	age	_	Child10 age	_
Child5	age	_	Child11 age	_
Child6	age	_	Child12 age	_

Suppose that there are many records of this general type and that we wish to obtain a frequency by age of all the children in the survey. In ANALYSIS, the command FREQ CHILD1AGE will give the frequency of children by age for all those who happened to be listed in the Child1 blank, but to obtain the complete frequency, we must do 12 different frequencies and add the results together, clearly an awkward way to answer a simple question! Finding the youngest child is also cumbersome, since there is no assurance that it will be Child1; we would have to write a program to inspect each of the 12 blanks and find the smallest number.

Now let's break the problem into manageable pieces and place each child in a single record, as follows:

File CHILD.REC   Household No. ___ Child No. __ Age __

We also create a household questionnaire:

   File HH.REC   Household No. ___ Mother's Age __

For the sample household we complete three child questionnaires and one for the
household. There is no need to provide information for nine non-existent children,
and storage space is thus conserved.

Now finding the number and ages of children in the survey requires only a simple
frequency in the file CHILD.REC. The number of households can easily be
obtained from the HH.REC file. The only problem comes when we want to
information in CHILD.REC to that in HH.REC, to count, for example, the number
of children whose mothers are less than 20 years of age. This is the purpose of the
RELATE command in ANALYSIS, described in the next section.

The RELATE Command in ANALYSIS

The RELATE command links one or more files to the main file during analysis,
using a common identifier to find matching records. In the example given above,
the common identifier is Household Number, and the instructions to ANALYSIS
would be:

READ CHILD
RELATE HOUSEHOLDN HH

(RELATE <Common Identifier Field> <File name>)

The LIST command then produces:

```
REC   HOUSEHOLDN CHILDNO AGE MOTHERSAGE
---   ---------- ------- --- ----------
 1             1       1   1           34
 2             1       2   3           34
 3             1       3   6           34
 4             2       1   9           44
 5             3       1   5           19
 6             3       2   8           19
```

FREQ AGE can still be used to count the children:

```
AGE   |  Freq  Percent   Cum.
------+-----------------------
  1   |     1   16.7%    16.7%
  3   |     1   16.7%    33.3%
  5   |     1   16.7%    50.0%
  6   |     1   16.7%    66.7%
  8   |     1   16.7%    83.3%
  9   |     1   16.7%   100.0%
------+-----------------------
Total |     6  100.0%
```

but now, SELECT MOTHERSAGE < 20 will allow us to count the children of young mothers by repeating FREQ AGE:

```
Current selection: mothersage < 20

AGE   |  Freq  Percent   Cum.
------+-----------------------
  5   |     1   50.0%    50.0%
  8   |     1   50.0%   100.0%
------+-----------------------
Total |     2  100.0%
```

The LIST now looks like this:

```
REC   HOUSEHOLDN  CHILDNO  AGE  MOTHERSAGE
---   ----------  -------  ---  ----------
  5            3        1    5          19
  6            3        2    8          19
```

More than one file may be RELATEd to the main file. In a hospital-infection investigation, the primary unit of analysis was the "Nurse-Baby-Shift," in which a given baby was assigned to one nurse for a particular nursing shift. The number of such "exposures" was being investigated to explore transmission from ill babies to nurses and vice versa. The files might be NBSHIFT.REC, BABY.REC, and NURSE.REC. NBSHIFT records might contain only the fields DATE, SHIFTNUM, BABYID, and NURSEID. NURSE and BABY files would contain information about illness, dates of onset, etc., for nurses and babies.

To set up the analysis, the following commands would be used:

> **READ NBSHIFT**
> **RELATE BABYID BABY**
> **RELATE NURSEID NURSE**

This would produce the logical equivalent of a single record structure in which each NBSHIFT had full information on the baby and nurse that were paired for that shift. The analysis could then explore associations between illness and number of shifts exposed to an ill person of either type.

Note that the main file, the one invoked with the READ command, may have many records for each matching record in the RELATEd files. The RELATEd files, however, must have *only one instance of each identifier*, that is, the identifiers must be unique and identify only one record in the RELATEd files. This makes sense, since ANALYSIS can relate only one record at a time, and there would be no way of choosing among two matching records in the related files.

For this reason, it is important to identify the main file as the one with the most basic information. A RELATEd file can never have more than one record that matches a single record in the main file, but the reverse is not necessarily true. In the example used above the main file is NBSHIFT because each NBSHIFT record has only one related NURSE or BABY record. A nurse, however, could be related to more than one baby per shift. With PERSONs and HOUSEholds, PERSON is the main file because each person belongs to only one house, but the reverse is not necessarily true. These characteristics make Epi Info technically a "hierarchical" database system, although in other respects it is "relational."

Using RELATE to Link Denominator Information

The calculation of rates is essential in many epidemiologic situations. In a disease surveillance system, population information forms the denominator for rate calculations, but it would be wasteful to include the county population in each data-entry record for an individual case. The RELATE command can be used to access a file of populations as follows:

The records might be:

Main file:		Related population file:
Case information	COUNTYID	COUNTYID POPULATION

After RELATing these two files by COUNTYID, the report generator can be used to produce a report with rates given for each county.

Using MERGE To Link Related Files

As described in the chapter on the MERGE program, MERGE will produce new files by performing linking similar to that done by the RELATE command. The files produced by MERGE are permanent copies of the linked records and may be useful if repeated analyses are likely.

Permanent copies of the files RELATEd in ANALYSIS can be made without using MERGE simply by using the WRITE RECFILE command after the READ, RELATE, and ROUTE commands have been given.

Relational Data Entry Using CHECK and ENTER

There are times when data items for related files are most conveniently entered at the same time. In disease surveillance systems, for example, major diseases such as AIDS, hepatitis, malaria, and tuberculosis each have their own form that calls for specific information. Usually a common core of questions about name, address, age, sex, and diagnosis describes any case and can be kept in a single file of "disease reports." The disease-specific information can most logically be stored in separate files, one for each major disease, perhaps with diseases requiring less information combined into a single form. To describe a relational file entry system in Epi Info, we will assume that a single disease-report file is to be supplemented by separate files for hepatitis, AIDS, and malaria in files called COMMON, HEP, AIDS, and MALARIA.

We would like to enter records in the COMMON file. When the question on DISEASE is answered as HEP A, HEP B, HEP NANB, or HEP UNS, the screen will change to the questionnaire for hepatitis. Similarly, AIDS or MALARIA will bring up the AIDS or MALARIA questionnaires. After the appropriate information has been entered, the screen will switch automatically back to the COMMON form.

The first task in setting up such a system is to make questionnaires for all of the forms involved, and, using ENTER, to make data files for all. All of the forms must have a common identifier field that we shall call CASEID. The few instructions necessary to set up the multi-file entry facility are inserted in the .CHK file for the main data file.

Setting Up the .CHK File for Relational Data Entry

To set up files for relational linking during data entry, the main file and *each of the related files must have an identifier field by the same name, and each of these fields must be indexed*, by adding the word KEY to the proper field in the corresponding.CHK file. The identifier field in the main file, but not necessarily those in the related files, must have the words KEY UNIQUE for that field in the .CHK file. Thus, in the example given above, the identifier field might be CASEID. This would be present in each of the files: COMMON, HEP, AIDS, and MALARIA. HEP, AIDS, and MALARIA would each have a .CHK file with KEY specified for the CASEID field. In the .CHK file for COMMON, CASEID would be designated KEY UNIQUE.

Note that, although the identifier field in the main file may be (and often is) an automatically incrementing field of type <IDNUM>, the corresponding field in the related files *may not be an auto-incrementing field*, as its value is automatically set by the relational process. Such fields should be given the same name, but be numeric in type and of the same size as the <IDNUM> field in the main file.

The .CHK file for the main program must contain a RELATE <Field Name> <File Name> command in the field where the change to the secondary questionnaire will occur. This RELATE may be contained in an IF statement, as in:

IF (DISEASE = "HEP A") or (DISEASE = "HEP B")
THEN
** RELATE   CASEID HEPATITIS 1**
ENDIF

This will cause ENTER to switch to the HEPATITIS.REC questionnaire. The format of the RELATE statement for this purpose is:

  RELATE   <identifier field> <file name> {1}

Note: In Version 5 of Epi Info, the command in the .CHK file to do the same thing was

  ENTER <File Name> <Identifier Field> {1}

This format is still allowed as an alternative to provide compatibility with Version 5, but the RELATE statement is more like the RELATE statement in ANALYSIS, and is therefore preferred.

The <file name> is the name of the file with the related questionnaire, containing an <identifier field> with the same name, type, and length as the one used for linking records in the main file. The number of repetitions is specified as either "1" or many (not specified).

The main file must have a .CHK file containing the statement KEY UNIQUE for the <identifier field>, and the related file must have a .CHK file containing the KEY statement for this field. If the index (.IX) file's date and time and the secondary file's date and time are different, the index will be automatically updated to take care of any new records that may have been added since the two files were last used together.

When this form of the RELATE statement is encountered in the main .CHK file, ENTER switches to the related file, in this case, HEPATITIS.REC. If there is a record in which the <identifier field> matches that of the active record in the main

316

file, that record is displayed on the screen; otherwise, a new record is created and the appropriate value is placed in the <identifier field> as it is displayed.

When the related file is displayed, and the number of repetitions is 1, then the display returns to the next record in the and main file when a single related record has been entered; otherwise, it offers the opportunity to enter other related records until the <F10> key is pressed to return to the previous file.

If you are not sure how many secondary questionnaires will be needed for each primary questionnaire do not specify the number of repetitions so that the user will remain in the secondary file until he presses the <F10> key or answers "N" to the "Write data to disk (Y/N)?" question at the end of the questionnaire.

Secondary files behave exactly like other .REC files in ANALYSIS. They can be updated either in ANALYSIS or in ENTER, since ENTER will always check if an update has occurred before using the index, and will update the index if necessary before allowing multi-file entry. The index files created for relational data entry are the same as those created to speed up searching during data entry.

If the secondary file also has a .CHK file containing a RELATE statement, then another level of data entry will occur and ENTER will switch to yet another questionnaire, returning to the secondary questionnaire when entry is complete. Up to six levels of relations of this type are permitted, although more than three are seldom useful. When editing records, pressing <ALT-F> will allow jumping from level to level. A file can have several GOTO's separately to other files, using the same or different identifier fields.

Complete Relational File Systems

Suppose that files for recording household information, personal data, and multiple visits to individuals are organized in the following way:

Main File	HOUSEHOLDS	Contains HOUSEID variable
Secondary File	PERSONS	Contains HOUSEID and PERSONID
Tertiary File	VISITS	Contains PERSONID

One record in the Main file may lead to several records in the Secondary file. A record in the Secondary file may lead to several records in the Tertiary file. The Main file must have only one record for each unique identifier, e.g., HOUSEID. The Secondary file must have only one record for each identifier leading to Tertiary records, e.g., PERSONID.

The groups of related records are arranged like pyramids or hierarchies with the apex at the left.

		Visit 1
	Person 1	Visit 2
Household 1	Person 2	Visit 1
	Person 3	Visit 1
		Visit 2
		Visit 3
Household 2	etc.	
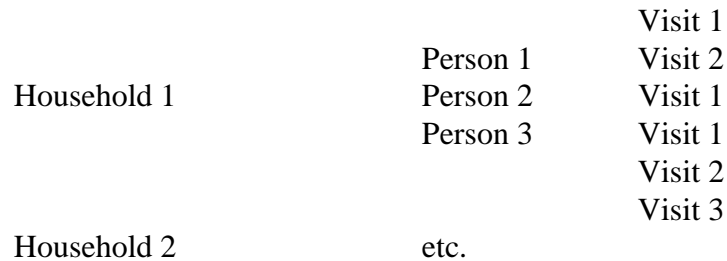
One-to-one relationships are also permitted, in which case the main file is the one that has one record for every record in the other file. The secondary file does not necessarily have a record for every record in the main file.

In ANALYSIS, *the order of the hierarchy is **reversed***. The main file may have several records for each one in a RELATEd file, but the converse is not true. In ANALYSIS, the files mentioned above would be set up as follows:

Main File	VISITS	Contains PERSONID
Secondary File	PERSONS	Contains HOUSEID and PERSONID
Tertiary File	HOUSEHOLDS	Contains HOUSEID variable

After the statements,

**READ VISIT
RELATE PERSONID PERSON
RELATE HOUSEID HOUSE**

the related records look like the following, even though the files actually contain only one copy of what appears as duplicate information:

Visit 1	Person 1	Household 1
Visit 2	Person 1	Household 1
Visit 1	Person 2	Household 1
Visit 1	Person 3	Household 1
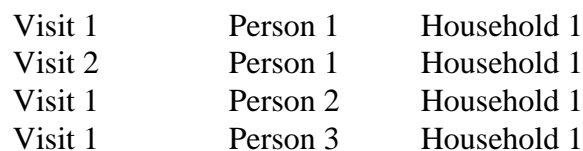
Visit 2	Person 3	Household 1
Visit 3	Person 3	Household 1
etc.		Household 2

The files illustrating this example are included with Epi Info and instructions for using them are contained in the tutorial at the beginning of this chapter.

Related Topics in Other Chapters

Methods for merging related files permanently are described in Chapter 18. Examples of relational file structures are contained in the \ NETSS system in Chapter 26. Further information on the RELATE command is in Chapter 36 and on KEY and ENTER in Chapter 35.

```
 File   Edit    Entry   ███████████   Comm    Utilities    Epi Info/Map
                   ┌──────────────────────────────────┐
                   │ ████████████████████████████     │
                   │ Demographic frequency tables     │
                   │ Graph historical trends          │
                   │ Map of State disease totals      │
                   │ ──────────────────────────────── │
                   │ Line list + check Duplicates     │
                   │ Show status of reports           │
                   │ Archive data - Summarize         │
                   └──────────────────────────────────┘




 F1-Help  F10-Quit │ « Create table of disease by time for each county
```
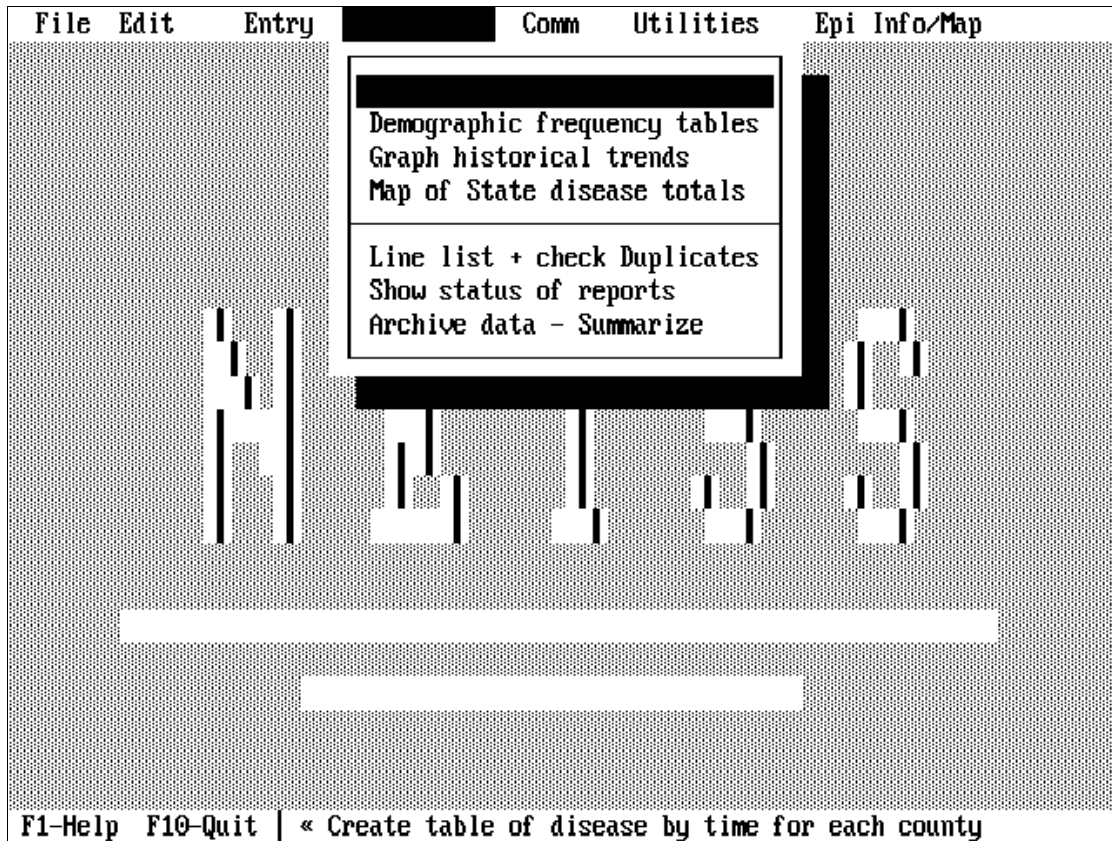
Chapter 26

Example: NETSS, a Disease Surveillance System using Relational File Structure and Hypertext Output

Designed and programmed by Robert Fagan and Denis Coulombier
with assistance from Barbara Panter-Connah, Lee Maddox, Gerald
Jones, and Harry Holden

> **Tutorial**
>
> **If you did not install the NETSS software modules when installing Epi Info, do so now, as described in Chapter 4.**
>
> **From the \NETSS directory, type NETSS to run the menu for the surveillance system. Choose ENTRY from this menu, enter 93 for the year, and follow the directions under "Data Entry" below to enter several surveillance records.**
>
> **From the NETSS menu, run each of the other functions. Use EPED or the menu editor to examine the many programs and .RPT files provided. The names of the programs can be found in this chapter or in the command blocks of the NETSS.MNU file.**
>
> **To construct your own surveillance system, make a copy of the files in a new directory and alter the programs to suit your own needs. Think carefully about other functions you would like to add and then construct new programs, adding each to the .MNU file as needed.**

Overview

The National Electronic Telecommunications System for Surveillance (NETSS), is a system by which each of the states and territories and two large cities in the United States transmit data to the Centers for Disease Control and Prevention (CDC) for weekly examination and publication. The system uses a standard national record format that allows each participating unit to maintain records in any desired format as long as standard records are produced weekly for transmission to CDC.

Although some participants use mainframe or minicomputer programs or commercial microcomputer software approximately 40 of the 56 use Epi Info-based systems installed and maintained by CDC staff. The programs included with Epi Info as examples are an abridged version of the software that is supplied to NETSS participants on request, usually with customization to adapt to local needs.

The NETSS programs supplied with Epi Info, Version 6, are designed to store, analyze, and forward records of reportable diseases in a State Health Department in the United States. They allow the reporting agency to analyze data locally and produce reports, graphs, and maps through a series of menu choices.

The NETSS programs supplied provide examples of Epi Info programming that can be adapted to many uses. Some of the functions include:

- A pulldown menu like the EPI6 menu, constructed by creating NETSS.MNU and other files for EPIGLUE.
- User Help in both the menu and the data entry form
- A core data entry form with extensive error checking, calculation of ages, skip patterns, hiding of fields, confirmation of codes, and other functions
- Related questionnaires for Hepatitis, Lyme Disease, and Meningitis/Hemophilus influenzae
- Programs for converting data to the standard NETSS record format and for transmitting files to CDC, with the aid of a commercial communication program (PROCOMM Plus)
- Management of both individual records and summary records in the same files
- The use of the report generator to customize tables for reports
- Automatic generation of tables embedded in hypertext so that unusual results are highlighted and the user can easily navigate through a large number of tables
- Linking Epi Info programs with Epi Map, a companion program (available from many of the same sources), to produce maps

showing surveillance results

- Linking Epi Info programs with Harvard Graphics, a commercial program, to produce "slide shows" of graphs with moving averages and linear trends

The NETSS programs are designed for use by those who may be unfamiliar with the use of computers. Almost anyone can learn to enter data in a short time and produce the programmed reports that are provided with the system. More experienced users can do ad hoc analyses using the Epi Info ANALYSIS program. For those learning to program Epi Info, NETSS offers abundant examples of both .CHK files and ANALYSIS .PGM programs that can be studied, modified, or used as examples in producing other systems.

Standard and Customizable Features

The System contains both standardized and customizable elements. Cases or summary records are entered on a core data screen and a series of related, disease-specific questionnaires. A customizable screen for state-specific data is also related to the core data form. Extensive checks of internal consistency and correct coding are performed during data entry and corrections are suggested or required immediately. Data entry can be performed by clerical personnel with very little computer knowledge.

After records have been entered, a number of different results can be produced. The system will automatically restructure the data for any selected week and prepare a file in the U.S. standard disease report format for sending to the CDC. A menu choice allows the file to be sent automatically via a commercial communication program called Procomm (not supplied). These functions may be altered to work with other communications programs such as Crosstalk.

Over the past six years, the staff of the Epidemiology Program Office have installed and maintained Epi Info computer systems for reportable disease surveillance in 40 U.S. State health departments. Most of these systems have been highly customized to meet local desires and requirements and for the first five years, no two departments had the same data-entry screen, although all the systems produced files in the national standard reporting format for forwarding to the CDC.

Although extensive customization was extremely popular and made it possible to enroll all 50 states and several territories and large cities in the national computerized reporting system, it had a high cost in terms of difficulty in maintenance and limitation on the number of features that could be provided. It is much more difficult to maintain 40 different computer systems than to maintain 40

copies of the same system.

The benefits of mass production and standardization of parts apply to computer systems as well as automobiles, and the latest NETSS programs have been designed around standard modules for easy installation and maintenance, with customizable features contained in other modules so that innovation is given free reign in areas where it can be most useful. This combination of standard and changeable features has allowed many more functions to be included than in previous Epi Info systems for surveillance. The additions include graphs, maps, special questionnaires for hepatitis, meningitis, and other conditions, access to data from past years, the ability to handle both summary and individual case records on the same form, and much more error-checking and automatic coding during data entry.

States are free to add as many questions as they find useful in the customizable part of the system, and to add new types of reports to the menus as they are developed. Others using the system as a model for writing entirely different surveillance systems outside the United States will necessarily make more extensive changes, but the standard modules are available as a guide in performing the programming.

Running the NETSS Programs

The main NETSS menu resembles the EPI6 menu and is constructed by programming EPIGLUE as described in Chapter 20. Within the NETSS menu and data entry screen, HELP may be obtained by pressing F1. The menu may be operated with a mouse or from the keyboard.

Files that illustrate how this function is programmed:
>   NETSS.EXE   (A copy of EPI6.EXE or EPIGLUE.EXE)
>   NETSS.MNU
>   NETSS.HLP, compiled from NETSSMAN.TXT
>   NETSS.SCR

Note especially:
>   NETSS.MNU, the file that contains menu items, help hints (in
>   STRINGTABLE), and Command Blocks to run the NETSS
>   programs.

Entering data

To enter or edit data: select ENTRY and then enter the year ## for the file you wish to use. All records are stored in files by year of report in 2 parts. Disease specific

files are 'RELATED' or attached to the main file and can be read by selecting the 'CORE' or main part of the record and then pressing <Alt-F> and selecting the disease specific file.

For new records the program will automatically bring up the appropriate linked or 'RELATED' file. To begin entering data, highlight ENTRY on the menu bar and press <Enter>. A request to enter the year will appear on the screen. Enter the current year, for example, "93." This is necessary because records for each year are kept in a separate file. When you press <Enter> you should see the main data entry screen, entitled "Confidential Morbidity Report."

This is the standard core screen of the Modular System. Although it looks quite simple, there are numerous aids to coding and entering correct data that are not visible. Each entry item is followed by a blank called a "field" in which you will enter information. The record number is shown on the lower right corner of the screen, and instructions are given on the last two lines. The bottom line gives information on the most common operations in the ENTER program, the part of Epi Info that is operating behind the scenes to display the report form. On the next line up is the name of the current field and information about what kind of entry is expected. Further information can be obtained for most fields by pressing the <F9> key to see the codes or other possible entries that are provided.

Let's enter one report (one "record"). The ID number and today's date are already supplied. ID will increment automatically as each new record is entered. The UPDATED field will always indicate the date the report was entered into the computer or the last date on which it was revised. The cursor will skip these fields.

List of fields

LAST NAME	FIRST		
ADDRESS	CITY	ZIP CODE	
COUNTY	CODE	REGION	
BIRTHDATE	AGE TYPE	AGE	
RACE	ETHNIC	SEX	EVENT
NAME	EVENT DATE		
EVENT TYPE	OUTBREAK ASSOCIATED		
MMWRWK	YEAR REPORTED		
OTHER	IMPORTED	STATUS	

LAST NAME If this report is for an individual case, enter the last name. If the record is to be a summary record (e.g. for 10 cases), enter the letter "S" for "summary" and press <Enter>. The cursor will jump past the name and address fields so that you can enter the type of event (disease) and number of cases. Names may be entered in upper or lower case; they will be automatically converted to

325

upper case letters so that a search for particular records does not have to deal with Jones, jones, and JONES as three separate names. All will be JONES in the computer record. When you have completed an entry, press the <Enter> key to go on to the next.

FIRST Enter the first name in this field, followed by an initial if desired. It is important to have a uniform method for dealing with initials and punctuation that is followed by everyone entering data in the system.   generally omitting the period after the initial is best unless the records will be used to generate mailing labels.

ADDRESSand CITY are filled in as for any form.

COUNTY codes are provided in CNTYCODE.REC. Press <F9> to show the counties in a window on the screen. To find a particular county, type the first letter or few letters of the county name. The bar cursor will move to the first county matching those letters. You can type additional letters until the county is identified, or move the cursor with the up and down arrow keys to the correct choice. Press <Enter> as usual to confirm the entry, and the name will appear in the COUNTY field. The county code appears automatically in COUNTY CODE. The cursor jumps to ZIP CODE unless REGION has been specifically set up for your state.

ZIP CODE Enter a five-digit zip code.

BIRTHDATE If the birthdate is known, enter it in the format mm/dd/yy-- 06/01/92, for example. You enter only the numbers as the slash marks are inserted automatically. If the birthdate is unknown, press <Enter> to proceed to AGE. Sometimes it is necessary to enter a birthdate with an unknown day. In the present system, this must be done by approximating the day as mid-month, generally "15." or by using the AGE instead. If you enter a birthdate, the cursor will skip to RACE and the age will be calculated automatically when EVENT DATE is entered further down on the page.

AGE If the cursor jumps to this field, fill in an age. Usually this is in years, but you can enter a number representing days, weeks, months, years, or a census code and then indicate in the AGE TYPE field what the AGE represents. A window of codes will pop up if you enter an AGE; enter the correct code for AGE TYPE and it will be placed in the AGE TYPE field. It is important to have a system for deciding which units to use so that a child of 4 months, for example is not entered as 4 months one time and 120 days another.

RACE is coded as Native American, Asian/Pacific Islander, Black, White, Other, and Unknown. Press <F9> to see the codes and use the arrow keys to move the cursor to the correct choice or merely type the letter. Note that the full text for your choice, such as BLACK appears beside the field so that you can see that you

entered the correct code.

ETHNIC means Hispanic or Non Hispanic. As for RACE, the codes are displayed with <F9> and confirmed after you make an entry.

SEX offers the choices Male, Female, and Unknown.
EVENT NAME is a "must-enter" field, meaning that you cannot move the cursor from the field until an entry has been made. EVENT is a general term that covers DISEASE, INJURY, or whatever else the surveillance system is designed to cover. For most disease-reporting systems it means DISEASE. Press <F9> to see the disease names available from the file DISCODES.REC. When you have become experienced with the system, it may be easier to type the names rather than selecting from the menu. If you misspell one, the error will be caught automatically. When the entry is made, a code is placed in the EVENT field and the cursor will skip to EVENT DATE.

EVENT DATE is also a must-enter field. If possible enter the date of onset here. If the date of onset is not available, enter the date of diagnosis, the date of laboratory diagnosis, the date of report to the county, the date of report to the State, or, least desirable of all, any date associated with the case. This field is designed so that you can always enter some date, designating the kind of date it is in the EVENT TYPE field.

EVENT TYPE The choices here describe the EVENT DATE. Is it 1) Date of onset, 2) Date of diagnosis, 3)Date of laboratory diagnosis 4)Date of report to the county 5) Date of report to the State, or 9) An unknown or other type of date. Enter the one closest to the date of onset--the one that is most desired for surveillance purposes.

OUTBREAK ASSOCiated Press <Enter> if the case is not part of an outbreak. If it is outbreak associated, enter either 1 for "Yes," 2 for No, or 9 for Unknown.   If your state has a system of assigning numbers to outbreak investigations, type the outbreak number.

MMWR WK MMWR Week is the week of the epidemiologic year for which the report is to be published at the Centers for Disease Control. A calendar of week numbers for the decade of the 90's should be kept beside the computer. Except in very unusual cases, the current week should be entered. An exception might be made if a number of cases have not been entered for a period of several months in the past; they should then be given week numbers corresponding to the date they normally would have been reported. When such backdated reports are received at CDC, however, they are excluded from reports for the current week, and this should be avoided in all but extreme cases. The MMWR WK should definitely not be backdated to agree with the onset date in routine cases. It is expected that the two

dates will be different in many cases. MMWRWK must be entered for every case record.

YR The year must agree with the MMWRWK. Enter the last two digits of the year, as in "94."

REPORTED is a date in the form mm/dd/yy that represents the date the EVENT was reported to the State. Consult your State Epidemiologist to see if there is a local preference for the type of date that should be entered here or whether the date should be skipped entirely.

OTHER DATA is a place where laboratory values and other comments can be entered. For certain diseases, a window will pop up to ask specific questions for entry in this field. If the EVENT NAME is animal rabies, for example, the window will ask what species of animal; for rubella, information on immunization status and pregnancy is requested.

IMPORTED For this field, the codes may be seen by pressing <F9>. Choose 1) Acquired in State, 2) Acquired Outside USA, or 3) Acquired in USA but outside state, or press <Enter> if unknown. If the EVENT NAME is MEASLES, an entry must be made for this field; otherwise, it is optional.

At this point the screen changes entirely to a new form, called the Supplemental Report, containing information tailored to State circumstances and preferences. This form is completely customizable and the fields presented in the Modular System Supplemental Report are only examples. The ID Number is carried over from the main screen, allowing later linkage of the information on the two screens, which show information in two different files.

Fill in the blanks in the Supplemental Report. After the last field, the question "Write data to disk (Y/N)?" appears at the bottom of the screen. Answer "Y." Depending on the type of EVENT NAME you entered, another disease-specific form may pop up at this point. These forms are discussed below in the next section. After you have filled out all the necessary supplementary forms, the cursor will return to REPORT STATUS, the last field on the CONFIDENTIAL MORBIDITY REPORT (the CORE screen).

REPORT STATUS The choices here are 1) Confirmed 2) Probable 3) Suspect, and 9) Unknown. These definitions are found in the document, "Case Definitions for Public Health Surveillance," (MMWR Recommendations and Reports #13, October 19, 1990, USDHHS, Centers for Disease Control, Atlanta, GA, U.S.A.), and have been agreed upon jointly by the U.S. Council of State Epidemiologists and CDC. Enter the correct status for the case and the disease.

Data entry is now complete for this case or summary record. The question "Write data to disk (Y/N)?" appears at the bottom of the screen. Check over your entries, and if all are correct, answer "Y." If you need to correct an entry, type "N" and the cursor will return to the top of the form. Move to the field to be corrected using the up and down arrows and make the correction. Press the <End> key to complete the form and then answer "Y" to the "Write data..." question that appears.

> **Files that illustrate how this function is programmed:**
> NETSS.MNU
> GETP92.REC - GEPI94.REC
> GEPIXX.CHK
>
> GEPIREL.CHK--This file is created by the menu each time you choose a
>   year for data entry and erased after you are finished.
> CNTYCODE.REC--Codes for counties
> DISCODES.REC--Codes for diseases (Events)
>
>
> **Note especially:**
>
> In NETSS.MNU: MENUITEM "E&ntry" and function IDM_ENTRY.
> These produce GEPIREL.CHK containing the year chosen and then
> run ENTER with the .REC file for the correct year. GEPIREL.CHK is
> INCLUDEd in the generic GEPIXX.CHK automatically.

Entering Summary or Aggregate Data

The same form used for individual cases can also be used to enter summary or aggregate data, such as "10 cases of Gonorrhea from Washitaw County for week 42." The system is set up so that a particular disease is entered either as individual cases or as aggregate numbers. In NETSS, Tuberculosis, Syphilis and Gonorrhea are set up to be entered as aggregate numbers. You can alter this by changing the .CHK file for the field called EVENTNAME.

Summary reports for the diseases programmed to be entered in the aggregate are entered by typing "S" for "Summary" in the LASTNAME field and pressing <Enter>. The cursor will jump to EVENTNAME and the intervening fields will disappear from the screen. Fill in the rest of the blanks as for an individual case. After EVENT NAME, the cursor will jump to the field called CASE COUNT. Fill in the number of cases being reported for which the other information on the form is true. Leave blank any fields that are not true for the entire group of cases, although at least EVENT, MMWR WEEK, and YR should be filled in. If EVENT DATE is filled in, it should be 5 for date reported to the state health department.

Files that illustrate how this function is programmed:
GEPIXX.CHK

Note especially:
The use of the HIDE and UNHIDE commands

Disease-Specific Forms

Disease-specific forms are provided for Hepatitis, Meningitis, and Lyme Disease. They appear after the STATUS field if you have entered one of these conditions under EVENT NAME. They take the place of the printed forms for these diseases that are normally sent to the Hepatitis, Meningitis, and Lyme Disease programs at CDC.

The HEPATITIS form contains many questions that are coded as 1) Yes, 2) No, and 9) Unknown. Pressing <F9> for any of these fields will bring up the codes and allow a selection by placing the bar cursor on the choice and pressing <Enter>. The other fields require dates, either in the mm/dd/yy format or as a single year, such as "92."

If the diagnosis is LISTeriosis, Hemophilus Influenzae (H FLU), Meningococcal Disease (MEN), or BACTerial MENingitis OTHer, the form for Bacterial Meningitis and Bacteremia appears. It asks a number of questions about day care contact, pregnancy, laboratory isolates and antimicrobial sensitivity that are important in the national surveillance of meningitis and bacteremia.

If LYME Disease is chosen, a two-screen form asking for details for clinical laboratory details of the case will appear. The results entered in this form and transmitted to CDC will be made available to the CDC Lyme Disease Program at Fort Collins, Colorado, each week.

Files that illustrate how this function is programmed:
GEPIXX.CHK
MEN*.*(meaning "all files with names beginning with MEN")
HEP*.*
LYME*.*

Note especially:
The RELATE commands used to link the supplementary form screens to
GEPI*.REC, found in GEPIREL.CHK

Creating New Files in a Standard Record Format

The program will process your data file and prepare data for sending to CDC in the standard format agreed upon by the U.S. Council of State and Territorial Epidemiologists. The file includes information on your database for the entire year along with the new or revised records entered during the Transmit Week. After selecting you will be asked to select a time period for data format. This will usually be the current week unless you wish to transmit records from a previous time period.

You will now see a number of comments on the screen indicating activity of the program, and the main menu will again appear. A file called NETSS.DAT has been prepared for sending to CDC. It contains all records entered or updated on the week-ending date given above or any of the seven days preceding that date. It will also contain summary records and special verification records that give information about the numbers of records for each EVENTTYPE in your entire year-to-date file. A list of special forms transmitted and of record deletion requests is sent to your printer. The printer must be turned on for this program to function.

Files that illustrate how this function is programmed:
NETSS.MNU
NETSS.PGM

Note especially:
The use of PICKLIST and of background characters in
NETSS.PGM to construct menus and the WRITE statement
that sends restructured data to a new ASCII file for forwarding
to CDC.

Sending Data Using a Commercial Communications Program

This menu selection offers a convenient method for transmitting data to CDC. This can only be done if someone has set up your system with the necessary communications program and the telephone number of the appropriate network connection for your department. If these programs have been set up and configured for your modem and telecommunication method, you will see a description of what is occurring on the screen and the file will automatically be transmitted to the proper destination.

Files that illustrate how this function is programmed:
NETSS.MNU

Note especially:
The IDM_SEND Command Block that would contain commands to run
a commercial communications program like Procomm Plus or CrossTalk with
a suitable script to send NETSS.DAT to its destination.

Providing Remote Access to the Computer

This program uses Carbon Copy (not included) to provide access from CDC to your
system. Using Carbon Copy systems support personnel at CDC can update files
and provide training without having to travel to your location.

Files that illustrate how this function is programmed:
NETSS.MNU

Note especially:
The IDM_REMOTE Command Block that would run a commercial
 program (such as Carbon Copy) allowing access to the computer
from another computer via modem.

Creating Summary Data from Individual Case Records

Aggregating a large case based data set is often the first step when analyzing the
data. The amount of information used for data entry and maintenance such as
address, last name and first name, is no longer necessary. In fact, quite often, it has
to be removed for ensuring confidentiality. This steps involve deciding which
"UNITS" are the most appropriate to describe the condition in terms of PLACE,
TIME and PERSON. For instance, the data file may be aggregated by week of
notification, condition reported, and county of residence, as in our example.

Files that illustrate how this function is programmed:
SUMMARY.PGM

Note especially:
The use of the OUTPUT SUMTABLES command to produce the summary
data.

Producing Hypertext Containing Tables

Menu Choice: EPI BULLETIN - TABLE 1

Epi Bulletin Table 1 is a hypertext report. A menu will come to the screen allowing
the user to 'mouse click' on the Region, County or disease of interest. Each table

shows current week totals for disease followed by a sliding last 4 week total, 4 week historical average, a sliding last 52 week total, and a sliding last 52 week historical average.

Portions of the report may be printed or cut and pasted into other files using the file and edit options found on the NETSS main menu.

The report is generated by running the TABLE1.PGM. This is an example of using Epi Info analysis to produce attractive hypertext reports. Almost 100 tables are generated by the program. This may vary according to the number of diseases reported by each state and the number of counties reporting. This is an example of using hypertext technology to access large highly hierarchical information (State, Region, County) easily. Furthermore, hypertext designators are highlighted for counties with more than a certain cutoff number of cases for the current week, and for the past 4 weeks, allowing a quick access to area where there may be a problem.

Feel free to explore and customize the report to suit your need.

Files that illustrate how this function is programmed:
NETSS.MNU
TABLE1.*
HYPER.EXE The hypertext compiler, automatically run from NETSS.MNU

Note especially:
The IDM_TABLE1 Command Block in NETSS.MNU. The Command Block IDM_TABLE1 in NETSS.MNU, and the use of the use of the REPORT function with TABLE1.RPT to specify output of ANALYSIS in hypertext. The resulting file is automatically compiled by HYPER.EXE so that it will function in the hypertext viewer of the main menu.

Demographic Frequency Tables

Menu Choice: DEMOGRAPHIC FREQUENCY TABLES

This report produces a set of tables of diseases by race, sex and age. It displays information about the past 52 weeks, up to the current MMWR week. Frequency tables of disease by age, by sex, and by race are built using a generic hypertext table report file PERSON.RPT, called with various combinations of the previous variables. This report file could be customized with very little modification to generate other types of tables.

This is an example of creating hypertext reports from within ANALYSIS, using embedded code for the hypertext compiler HYPER.EXE.

Files that illustrate how this function is programmed:
NETSS.MNU
HYPER.EXE
PERSON.*

Note especially:
The Command Block IDM_DEMOGRAPHICS in NETSS.MNU, and the use
of the REPORT function with PERSON.RPT to specify output of
ANALYSIS in hypertext. The resulting file is automatically compiled by
HYPER.EXE so that it will function in the hypertext viewer of the main
menu.

Linking Epi Info programs with Harvard Graphics

Menu Choice: GRAPH HISTORICAL TRENDS

This report runs an Epi Info program that generates a set of ASCII files used to
export current data to an Harvard Graphics slide show (version 3), NETSS.SH3,
supplied with NETSS as an example. A similar approach could be developed with
Harvard Graphics version 2.x, Freelance Graphics, or any graphic package that
accepts data links to ASCII files, and run in slide show.

The slide show is stored in file NETSS.SH3, while individual graphs for the five
diseases included in this demo are stored in 5 template files called with the name of
the event code and a .TP3 extension: 11000.TP3, 10280.DAT, 10500.DAT,
10590.DAT, and 10100.DAT . A link was established between these template files
and ASCII files produced by the program.

Display includes disease specific weekly notifications for the past 4 years, 10 week
moving average to show seasonality, and trend over the entire period.

The Epi Info program TIME.PGM gives an example of structured programming in
Epi Info through the use of sub routines called by the GOTO ... RETURN
statement. Very little is needed in order to add a new disease to the slide show, and
only one set of code for the actual selection and calculation involved needs to be
maintained. It exemplifies the use of the WRITE command to generate ASCII files
that can be used to export output to virtually any software.

Customizing this program to suit your needs may require some modifications, such
as adapting to the ASCII format used by your graphic software, designing the slide
show that displays linked data and so forth.

Files that illustrate how this function is programmed:
NETSS.MNU
TIME.PGM

Note especially:
The IDM_GRAPH Command Block in NETSS.MNU.

Linking Epi Info programs with Epi Map

Menu Choice: MAP OF STATE DISEASE TOTALS

This option runs the program that update disease specific data used to make State maps that can be displayed using Epi Map, a CDC public domain software for mapping data from Epi Info or dBase files.

Five demonstration maps are supplied with the NETSS application. Each map is a double map of the State of Michigan. One map shows disease specific counts of notifications received in the past 4 weeks at the State Health Department, the other one is a map showing disease specific rates over the past 52 weeks in the State of Michigan.

This Epi Info program PLACE.PGM exemplifies the steps necessary when analyzing geographical surveillance data: selection of time and person characteristics, aggregation by geographical units, link (relation) with a file holding population counts by geographical units, calculation of rates, and final data output for Epimap.

Files that illustrate how this function is programmed:
NETSS.MNU
*.MAP
USMI.BND
PLACE.PGM

Note especially:
The IDM_MAP Command Block in NETSS.MNU.

Checking for Duplicates

Menu Choice: LINE LIST + CHECK DUPLICATES

This hypertext report produces a line listing of notification matching for the first four letters of the last name and first name. It produces a general line listing, by alphabetic order, highlighting the matching notifications. When clicking on the ID number of a matching record, a screen containing detailed information about the matching record is displayed. Print the list of matching records (using <F5>), and the edit in enter to correct or update information.

The "case definition" for a match is located in the header of the .PGM file called CHECKDUP.PGM. Change this to elaborate any more sophisticated matching definition.

This program uses intensively the TYPE and WRITE commands to create a hypertext file.

> **Files that illustrate how this function is programmed:**
> NETSS.MNU
> NETSS.MNU
>
> **Note especially:**
> The IDM_DUPES Command Block in NETSS.MNU.

Displaying Report Status

Menu Choice: SHOW STATUS OF REPORTS

This option shows the current status of all reports generated by the "REPORT" menu, including their names, and date and time of last update.

This allows you to check whether a particular report is still accurate or needs to be run again, in order to reflect the latest changes in the data file.

> **Files that illustrate how this function is programmed:**
> NETSS.MNU
> STATUS.PGM
> UPDATE.TXT
> STATUS.REC

Note especially:
The IDM_STATUS block in NETSS.MNU.

Maintaining Lists

Menu Choice: MAINTAIN CODES, then COUNTIES or DISEASES

This choice allows adding a new county or disease to the code lists. It uses Epi Info ENTER to access the list allowing you to add or edit in a safe controlled manner.

Menu Choice: MAINTAIN CODES then DISEASES

Periodically a new disease will be added to the list of reportable diseases. This program will allow the user to add new codes to your official list of diseases. It uses Epi Info ENTER to access the list allowing you to add or edit in a safe controlled manner.

Files that illustrate how this function is programmed:
NETSS.MNU
CNTYCODE.REC
DISCODES.REC

Note especially:

The Command Blocks, IDM_RECCOUN and IDM_RECDIS in NETSS.MNU.

Editing the Menu

Menu Choice: EDIT THE MENU

This program uses Epi Info EPED text editor to access the set of instructions which produce the NETSS menu. It is important to note that changes made will not show until the user exits the menu and reloads it.

Files that illustrate how this function is programmed:
NETSS.MNU

Note especially:
The IDM_EDIT block in NETSS.MNU.

Backing Up and Restoring Data

Menu Choices: BACKUP and RESTORE

BACKUP will copy data files from the hard disk to floppy disks.

RESTORE will copy data files from floppy disks to the hard disk.

> **Files that illustrate how these functions are programmed:**
> NETSS.MNU
>
> **Note especially:**
> The IDM_BACKUP and IDM_RESTORE blocks in NETSS.MNU.

Setting Up Directories

Menu Choice: DIRECTORIES in UTILITIES

This option allows to define the five built-in variables that are used to store directories of the various external programs used. The default value in NETSS are C:\ EPIMAP for Epimap, and C:\ HG3 for Harvard Graphics. This may need to be changed to reflect your disk structure or if you are operating from a network.

> **Files that illustrate how this function is programmed:**
> NETSS.MNU
>
> **Note especially:**
> ID_SETUP

Related Topics in Other Chapters

Chapters 20 describes how menus such as the NETSS menu can be created and used to call programs and display hypertext. NETSS provides examples of many of the features described in Chapters 13, 22, 24, 25, 35, and 36, and a thorough reading of these chapters is recommended as background for dissection of the NETSS programs.

Chapter 27

Generating Artificial Data Files and Random Numbers

Tutorial

Suppose that for a study we need a list of 1,000 random integers ranging from 1 to 100.

Run EPED from the EPI6 menu. Make a questionnaire file containing only one field:

Number ######

Run ENTER and make a .REC file called RANDOM, but do not enter any records. Press <F10> to return to the EPI6 menu.

Run ANALYSIS from the EPI6 menu and READ RANDOM. Then set up the random integers:

NUMBER = RAN(100)
GENERATE 1000

The integers will be placed in the file. You can now use the commands, LIST NUMBER, FREQ NUMBER, or LINE NUMBER, to check the distribution. To print them for use, use ROUTE PRINTER followed by LIST NUMBER. If you want real numbers instead of integers, RND (100) command will produce these.

RUN the ANALYSIS programs SAMPLE and SSAMPLE to take samples of a data file such as OSWEGO.REC.

Overview

In investigations, a list of random numbers is often useful for choosing telephone

numbers, addresses, or other samples. The RAN and RND operators can be used to specify integer or real ("decimal") random numbers within a desired range. The GENERATE command can be used to put these numbers in a permanent data file from which they can be viewed, printed, or exported.

The GENERATE command combined with the RND, RAN, and RECODE functions will also produce a data set of any size up to 32,000 records tailored to your specifications, but derived from random numbers. Such files are useful for teaching, for software demonstrations, or for testing. The GENERATE function could even be used to produce a different set of "epidemic" investigation results for each member of a class.

We assume that Epi Info users will not risk their scientific reputations or the health of others by portraying such data as real results. These files should be carefully labelled and used only in situations in which their origin is clear. Please report any misuse to one of the authors.

Another ANALYSIS command, CMD, is used to communicate with external programs. It can be used to set up new commands in ANALYSIS that call external programs and make use of their functions as though they were part of ANALYSIS itself. This opens the door for creation of special-purpose additions to Epi Info by other programmers.

Producing Random Numbers

The process of generating a list of random numbers is described in the tutorial above. It is a good idea to assure yourself that the numbers generated have a mean fairly close to that expected, that they "look" random when plotted, etc., since any given finite list might have unusual properties. The RAN and RND functions produce different lists when GENERATE is run more than once. Each time GENERATE is run with the same active data set, the new values are appended to those already present.

A second, and more convenient, method of producing random numbers is offered by the EpiTable program . Using the random number options on the menu, you tell EpiTable how many random numbers are desired, and a custom-made set is produced that may be printed or sent to a file. See the help file that comes with EpiTable by pressing <F1> in EpiTable for more information.

Setting Up RECODEs for Generating Data

ANALYSIS can generate an entire data set for teaching or testing purposes. The RECODE command is used in a special way in combination with the GENERATE command to produce values for variables that are based on random numbers. To produce a generated data set, first create a questionnaire to describe the variables, and use ENTER to produce an *empty* .REC file. Then make an ANALYSIS program file to generate the data using RECODE and the random functions, followed by the command GENERATE and the number of records desired. The use of RECODE for this purpose is illustrated in the example below and in the GENERATE section of ANALYSIS Reference Chapter 36.

Sample Program

A program called ARTIFACT.BAT, a questionnaire, ARTIFACT.QES, and an ANALYSIS program, ARTIFACT.PGM, are included in Epi Info to demonstrate how GENERATE works. To run the program, choose GENERATE RANDOM DATA from the EXAMPLES menu in EPI6. The .BAT file erases any previous copy of ARTIFACT.REC and then runs ENTER to make a new, empty .REC file, ARTIFACT.REC. It then runs ARTIFACT.PGM in ANALYSIS to generate the necessary data. Before running the program you should erase any file called ARTIFACT.REC and create a new one in the ENTER program from the questionnaire ARTIFACT.QES.

```
*ARTIFACT.PGM, A program to generate data for the questionnaire ARTIFACT.REC
READ ARTIFACT
ECHO
ECHO Generating data.  Please be patient and watch the upper right screen corner.
*Generate ID numbers to put in the new file using a cumulative variable
DEFINE ID number width = 4 CUMULATIVE
ID = ID + 1
IDNUMBER = ID
*A special feature of RECODE assigns M if the random number from 0 to 1 is
* between 0 and .70, F if from .71 to .95, and unknown if .96 to 1.00.
RECODE RND TO SEX 70% = M 25%=F ELSE = .
*Assign a random integer from 0 to 65 to AGE
AGE = RAN (65)
*Modify this so that there is a 65% chance of having an age from 20 to 30.
IF RND(1) > 0.45 THEN AGE = RAN(10) + 20
*Make the reporting period start on a particular date.
REPORTDATE = "01/01/89"
*Allow it to continue for 400 days.
REPORTDATE = REPORTDATE + RAN 400
*Make 60% of answers Y; 37% N, and the rest unknown.
RECODE RND TO OPPORTUNIS 60% = Y 37% = N ELSE = .
*Set up the two remaining variables.
```

RECODE RND TO KAPOSISARC 55%=y 42%=N ELSE = .
RECODE RND TO HIV1 72%=y 27%=N ELSE= .
*Now create 100 such records.
GENERATE 100
*Check the frequencies to see the results.
TYPE ARTIFACT.REC
TITLE 1 "Frequencies from a Generated Data Set"
FREQ *
ECHO End of Results; Use PgUp to examine more closely.
?Press <F10> to leave ANALYSIS, or any other key to remain in ANALYSIS?

Note that the results are different if the program is run more than once.

Taking a Sample of Records from a File

It is often useful to take a sample of records from a large data file and thus reduce the number of records that need to be processed for a more detailed analysis. There seems little reason to process 100,000 records to obtain simple frequencies, for example, when a random or systematic sample of 1000 or 10000 records would probably provide nearly the same results with much faster processing times.

Two programs are provided for sampling. RSAMPLE.PGM writes a random sample of records from the original data file to a new file. SSAMPLE.PGM chooses a random starting point and then takes a systematic sample of records thereafter.

Both programs ask for the name of the data file and then for the denominator of the sampling fraction. If you wish to take a 5% sample, for example, the sampling fraction is 1/20 and you would enter 20 for the denominator. SSAMPLE would start at a random number in the first 20 records and then take every 20th record thereafter. SAMPLE will choose a random number from 1 to 20 for each record and select the record only if the number happens to be 20. Note that SSAMPLE will produce a sample containing 5% of the records in the file. SAMPLE, however, will produce a samples of varying size, although for a large file, the most likely size will be very close to 5% of the total records.

For SSAMPLE, as with any systematic sampling procedure, it is important to avoid sampling fractions that may produce a systematic bias. In a list of brides and grooms, for example, a 1/2 sample would be apt to produce a sample heavily biased toward either males or females. Sorting the list by SEX before taking the sample, however, would remove the bias.

Related Topics in Other Chapters

Chapters 9, 13, and 36 contain material about ANALYSIS programming necessary for understanding of this chapter.

```
F1-Hlp F2-File F3-Epiaid F4-Txt F5-Print F6-Set F7-Find F8-Blk F9-Save F10-Done
_CHOOSE 1 1 80 25
Color 1 14
Title 31 4 "EPIAID MAIN MENU"
Color 3 1
Title 29 8 "EPIDEMIC INVESTIGATION"

Choice 25 53 10 "       Epi 1 Memo" RUN epi1.epa
Choice 25 53 11 "Make Epi Info Questionnaire" RUN ques.epa
Choice 25 53 12 "   Develop Study Design" RUN design.epa
Color 3 1
Title 33 14 "EPED TUTORIAL"
%Color 4 12
Choice 25 53 16 " Word Processing Tutorial" RUN wptutor.epa
Color 1 14
Box 22 6 56 19
Color 7 0
Title 21 "SELECT A TOPIC USING THE UP AND DOWN ARROW KEYS"
Title 22 "THEN PRESS <ENTER>"
Title 23 "PRESS <F10> TO RETURN TO EPED"
_ENDCHOOSE
```

```
1  C:\EPI6\EPIAID.EPA              EPED  319360    L1   C1  TXTInsInd
```

Chapter **28**

Writing EPIAID Programs

Tutorial

Examine the program EPIAID.EPA, the menu of EPIAID, in EPED. Read the descriptions of the commands in Chapter 34 until you understand how EPIAID works.

Now make a backup copy of EPI1.EPA (like EPI1ORG.EPA). Edit EPI1.EPA in EPED to make the memo format suitable for your own organization. You may decide to change only the wordings or to get fancy and insert IF statements, jumps, etc. You may do the same with DESIGN.EPA or QUES.EPA.

Try writing a program to assist in writing letters or memos with your own return address, salutation, and other details.

Overview

EPIAID programs consist of text files that drive the EPED word processor. They are given names ending in .EPA. Two major types of programs may be written-- menus and interactive text programs.

Each line of an EPIAID program can be one of the following types, identified by the first character--the one on the left margin:

> •A line of **text**, beginning with any character except "_",
> "[", "%", or ":". The text will be put on the screen as part
> of the current document, using whatever form of
> justification (word wrap, etc.) is currently in effect.

•A request for **user input,** beginning with "[" and ending
with "]". The text in between will be put on the screen in
a user input window, and then the user will be allowed to
type text and use EPED commands until he or she signals
the end of the input by pressing <Esc>.

•An EPIAID **command**, beginning with an underline
character, "_". See Chapter 34 for the list of EPIAID
commands. They allow formatting text; conditional
jumps; defining, assigning, and retrieving variables;
access to DOS; construction of menus; display of blocks
of text; mathematical and logical operations; opening and
closing files; importation of text; and many other actions.

•A **comment**, beginning with "%", that explains the
program to other programmers.

•A **label**, beginning with ":", marking a location where a
JUMP command will direct the program to go next.

By combining these statements, almost any word processing task can be partially
automated while still allowing the user enough latitude to contribute the creative
parts.

Since each **user input** can be any length from one character to many pages, the
documents produced by different users can be quite different if this is desirable.
EPIAID programs can be written to provide a very tight harness for the user, or only
a light trace of suggestions, depending on the application.

Menu Programs in EPIAID

The program EPIAID.EPA is the main menu for EPIAID and a good example for
constructing other menus. EPIAID itself recognizes the name EPIAID.EPA as
having special meaning and always runs this program first if it is present. Hence,
you may choose to construct a new main menu for EPIAID with this name so that
your own choices will always be available.

If no EPIAID.EPA is present, EPIAID offers a choice of all the .EPA programs that
are present in the same directory with EPED. This may be more convenient when
you are working on a new program, and can be accomplished temporarily by
renaming EPIAID.EPA to another file name like MENU.EPA.

The core of a menu program is the _CHOOSE/_ENDCHOOSE command. This sets up the choices on the screen and ensures that the bar cursor will move from choice to choice. Between the two parts of the command, several other commands can be used to set up boxes on the screen, draw titles, and designate the actual choices and the actions you want to occur when they are made.

Many of the commands require screen locations. The upper left corner of the screen has location X=1, Y=1. X (column) locations move across the screen to 80, and Y (row or line) locations move down to the 25th line. Hence the X,Y location 40, 12 is near the center of the screen.

When other .EPA programs cease running, EPIAID tries to run EPIAID.EPA. If EPIAID.EPA is absent, then EPIAID presents the generic menu. If you would like your programs to return to a different menu, place a _RUN <NameOfMenu>.EPA command at the end of each program.

Interactive Text Programs in EPIAID

Programs other than menus are illustrated by most of the other .EPA programs in the Epi Info system. DESIGN.EPA is a good example of interaction that generates text, and EPI1.EPA illustrates how to program a more structured document. The EPED tutorial programs make good use of the _DISPLAY command to present multi line text windows on the screen.

Text interaction proceeds through a combination of text lines and input requests. An entire document can be generated by having the computer contribute a phrase or sentence and then ask the user to add input that fits the context. Thus, you could write an EPIAID program containing only text lines and input requests; the other commands only add flexibility.

The interaction becomes more flexible and interesting if you use the EPIAID commands, the ones that begin with the underline character. Now you can ask the user to supply information, and, depending on the answer, jump to another section of the program, present a helpful display or discussion, calculate a mathematical result, or import a file of relevant information.

The concept of **variables** is important in understanding EPIAID programs. They may be created by the _DEFINE command or automatically when you assign a value to a new variable as in:

_Alpha = 25   or
_Study = "case-control study"

Variables do not have types in EPIAID; they may contain either numbers or text strings. You cannot, of course, do mathematics with the strings.

There are several predefined variables. The most important is ENTRY. After every user input line, ENTRY is automatically assigned the value of the first 80 characters of the user's input. Hence if you ask the user to enter:

[Basis of study--Cases or Controls]

and he/she types:

Cases

then ENTRY will have the value "Cases" until the next user input occurs. If you would like to use the value later, then transfer the value from ENTRY to another variable defined for the purpose, as in:

StudyBasis = Entry

StudyBasis can then be used for the rest of the program whenever its value is needed.

The other commands will become clear by reading their descriptions in reference Chapter 34 and studying and experimenting with the .EPA files provided as examples. It is usually best to begin programming by first altering an existing program. Later, after having learned how basic operations are performed, one can write a program entirely "from scratch."

Related Topics in Other Chapters

Chapters 20 and 21 describe the EPIAID system and tell how to run EPIAID programs. Reference Chapter 34 gives more information on each of the commands used in writing EPIAID programs.

```
┌══════════════ Epi Info Версия  5.01b - Июль 1991 ═══════════════┐
│                                                                   │
│     Общественные Программы для Эпидемиологии и Наблюдений за Болезнями │
│                                                                   │
│   Центры Контроля Заболеваний      Всемирная Организация Здравоохранения │
│   Служба Программ Эпидемиологии     Глобальная Программа СПИД      │
│         Атланта, Джорджия              Женева, Швейцария           │
│                                                                   │
│ ┌═══════ Программы ═══════┬═══════════════ Описание ═══════════════┐ │
│ │ EPED     Редактор текстов│       Используйте EPED чтобы:          │ │
│ │ ENTER    Ввод данных     │ ▶ Создавать вопросники                │ │
│ │ ANALYSIS Анализ данных   │ ▶ Писать заметки и письма             │ │
│ │                          │ ▶ Получить руководство при расследовании │ │
│ │ CHECK    Проверка ввода  │ ▶ Писать или изменять программы Epi Info │ │
│ │ STATCALC Epi калькулятор │                                       │ │
│ │ CONVERT  Экспорт данных  │ EPED - обработчик слов для создания Epi Info │ │
│ │ IMPORT   Импорт данных   │ вопросников или для работы с текстами. │ │
│ │ MERGE    Слияние файлов  │ Он содержит систему ЭПИПОМ для обдумывания │ │
│ │ VALIDATE Сравнение файлов│ расследований эпидемий и получения черновиков │ │
│ │                          │ отчетов о расследованиях.             │ │
│ │ Об  Epi Info             │                                       │ │
│ │ Выход из Epi Info        │ Парам: EPED <Текст_файл или имя программы ЭПИПОМ> │ │
│ ├──────────────────────────┴───────────────────────────────────────┤ │
│ │ EPED                                                              │ │
│ └═ ↑↓, буква или пробел для выбора ════════ F9 Установка, F10  Выход ═┘ │
```

Chapter 29

Using Epi Info with Languages Other Than English

Overview

Characters in the English alphabet are given codes below 128 in the standard ASCII codes used by IBM-compatible microcomputers. Extra characters for languages other than English are coded from 128 to 255, introducing special problems for software that is used internationally. Epi Info questionnaires can be produced in a variety of non-English languages, and Epi Info can function quite well in such a setting.

Making a Data File from a Non-English Questionnaire

Once a questionnaire has been prepared in a non-English language, the ENTER program will produce a data file automatically. No extra steps or techniques are needed; the questionnaire should appear on the screen with all the non-English characters intact.

Field Names in Non-English Languages

Field names made by the ENTER program will not contain non-English characters, since the "high bit" is set to zero, essentially subtracting 128 from the ASCII code of the character. This produces field names that are sometimes difficult to remember or relate to. There are several solutions to this problem: 1) Avoid non-English characters in the phrases preceding entry blanks in the questionnaire, 2) Change the field names with CHECK to make others that are more meaningful, or 3) Use numbers or short English-letter words prior to fields in the questionnaire, and surround these with curly brackets {   } so that they become the field names.

Text Data in Non-English Languages

ASCII characters higher than 128 are accepted as data in text fields. This means that Arabic or Swedish names, for example, can be entered and handled as easily as English characters.

Dates

Much of the world uses either the <mm/dd/yy> or the <dd/mm/yy> format for dates. Scandinavian and Asian countries, however, use <yy/mm/dd>, a format for which no special field type is supplied in Version 6. One way to produce the yy/mm/dd format is to define separate fields for year, month, and day in the questionnaire. If these are called YR, MO, and DA, the following ANALYSIS program will convert the three separate fields to a new variable DEFINEd in ANALYSIS that has the U.S. date format.

*Assume that the questionnaire contains text or string fields for DA, MO, and YR,
* created with two underline characters each in the questionnaire.
DEFINE USDATE DATE WIDTH = 8
DEFINE SDATE STRING WIDTH = 8
*Now construct a string representation of the U.S. date format by pasting together
* the necessary pieces.
SDATE = MO + "/" + DA + "/" + YR
*Transfer the string variable data to the USDATE variable.
LET USDATE = SDATE
*List the results just to make sure everything is working.
LIST SDATE USDATE

U.S. dates can be used in mathematical operations such as subtracting one date from another to obtain an answer in days. Dates in the yy/mm/dd format can be sorted directly with the SORT command, since no conversion is required to perform a valid sort. If no mathematics is required with dates, the problem is even simpler and one can omit the steps involving USDATE, merely converting MO, DA, and YR to the string SDATE in order to SORT or LIST this single text string rather than the separate pieces.

Greek and Mathematical Symbols in EPED

The IBM character set contained in the DOS code page for the United States contains a number of characters that are not part of the standard ASCII standard, but are quite useful for drawing boxes and typing mathematical expressions and non-English languages. These characters may be produced in EPED after pressing the Scroll-Lock key.

Alternate character sets are supplied with DOS in some countries. Not all of the following instructions will apply to these non-English character sets, but most keys in the Scroll-lock mode will function as described.

Not all printers will print the full IBM character set. If your printer does not, you can still use the special characters for producing menu screens and other files, but printing requires a fully IBM-compatible printer. To type selected Greek letters and mathematical symbols, turn <Scroll Lock> on. The following Greek and mathematical symbols are available by typing on the main keyboard:

Scroll lock on--No Shift Key

```
Type:              ` 1 2 3 4 5 6 7 8 9 0 - =
To obtain:            1/2   1/4                 ≡

Type:              q w e r t y u i o p [ ] \
To obtain:             Ã         _ ½ ¹    _

Type:              a s d f g h j k l ; '
To obtain:         α o _

Type:              z x c v b n m , . /
To obtain:                         µ     /
```

Scroll lock on--Shift Key Pressed

```
Type:              ~ ! @ # $ % ^ & * ( ) _ +
To obtain:         Å   ²       _   • ■       ±

Type:              Q W E R T Y U I O P {  }
To obtain:

Type:              A S D F G H J K L : "
To obtain:             _       ¬         ÷

Type:              Z X C V B N M < > ?
To obtain:                     ß     ² ³
```

Non-English-Language Characters in EPED

When using EPED with the U.S. code page for DOS, it is possible to type a non-English character, by holding down the <Alt> key and using the following keys to produce vowels and other characters with special marks:

Use <Alt> with	a	e	i	o	u	c	n	y
to produce	áàâäåæ	éèêë	íìîï	óòôöØ	úùûü	ç	ñ	ÿ
or with <Alt> and <Shift>								
to produce	ÅÄÆ	É				ç	Ñ	

The first time the key is pressed, the first of the special characters appears. To produce the others, continue to hold down the <Alt> key and press the same letter key repeatedly until the correct character is obtained. Thus, holding the <Alt> key down and pressing "a" three times will produce "â ". After this, pressing the "e" key will first produce "ê ," with repeated presses of <Alt e> cycling through the various types of "e's." To produce the capital letters, hold the <Alt> key and the <Shift> key. Thus <Alt Shift A> produces Å and then Ä.

Translations of Epi Info

The programs and/or manual for Version 5 of Epi Info were translated into French, Spanish, Arabic, Chinese, Czech, German, Portugese, Italian, and Russian, by various organizations and individuals who volunteered to do so. Translation of the programs requires a knowledge of the programming language, Turbo Pascal, and, of course, the translator must be familiar with computer terminology in two languages. Usually the organization responsible for translation also develops a plan for distribution of the translated version.

The Epidemiology Program Office is happy to discuss the possibility of translation of Version 6 of Epi Info into other languages, and to provide the necessary source code to appropriate organizations for translation.
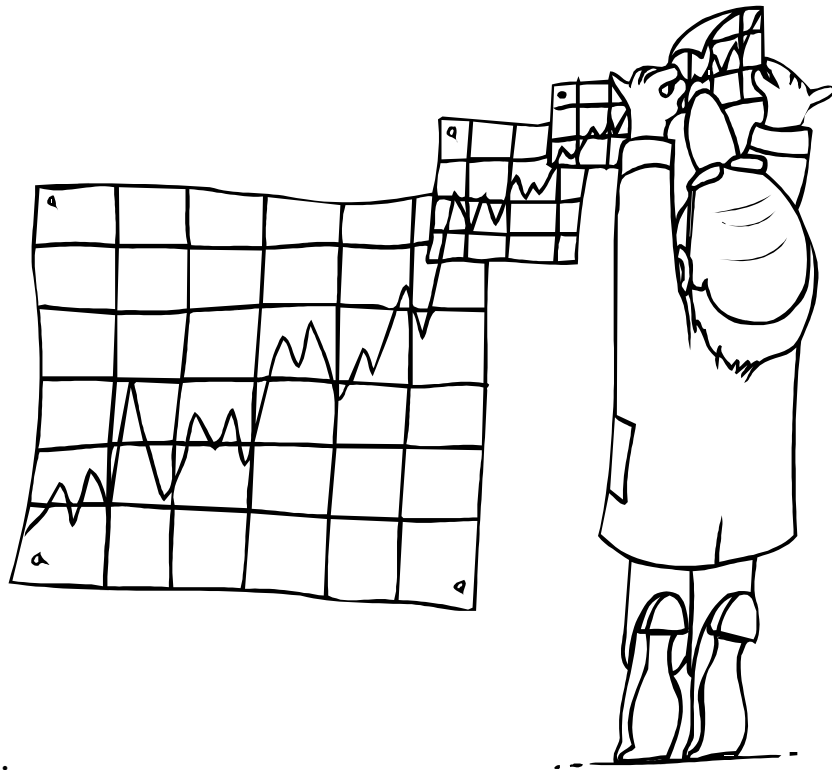
Chapter **30**

Portable Computers and Epidemiologic Field Investigation

A number of factors may contribute to the successful use of portable or desktop computers in field situations. We offer them here as a summary of experience with epidemiologists who have used portable computers in the field.

1. **Preparing through prior training and experience.** A field investigation is usually too pressing to offer the chance to learn new computer skills or try previously untested software. Computer operations that will be used in the field should be practiced under other circumstances so that there is reasonable assurance that they will succeed. Small problems in data formats or in understanding a particular program can consume large amounts of time unexpectedly, something a solo investigator cannot afford in the field.

2. **Having adequate hardware, software, and supplies.** The computer used should be in good working order and have a history of operating without problems, particularly disk errors or memory glitches. Unless you are an expert, avoid recently written or "home-brew" software, and test all software before using it in an investigation. Be sure that the disks you bring are the correct ones for the computer and are a quality brand, preferably one you have used personally and found satisfactory.

A number of accessories are important, although they add to the bulk of the computer. If your computer does not have an internal power-spike filter, one of the commercial ones available with a cord, several outlets, and a switch will be extra insurance and also solve the power-outlet problem.

To transmit messages or data over telephone lines, you will need a modem suitably installed in or connected to the computer and a means of connecting the modem to the phone line. Most modems use an "RJ11C" plug, the small, clear, plastic connector with four metal connector wires opposite a springy plastic "catch" lever that is found on modern touchtone phones. In an office building, you may find only multi-line phone plugs, and in an older phone system there may be still different types of connections to the phone line. Adapters for multi-line phone plugs which will let you insert an RJ11C plug and other types of adapters can be purchased at electronic or phone equipment stores.

3. **Protecting the computer and disks from theft, heat, shock, power spikes, and curious eyes.** Problems with theft of computer equipment are very real in many hotel rooms and office buildings. Marking the equipment indelibly with the name of your organization in a prominent place may discourage thieves, and you may want to bring a bicycle chain and lock, or the computer equivalents, which are available from a number of suppliers.

In an office setting, there is a temptation for passersby or evening staff to fiddle with an unlocked computer. Take backup copies of your disks home in the evening, and arrange to lock the door if at all possible. Taking the computer's power cord to lunch or home for the night is one additional minor security measure. Disks should be kept in boxes that will resist crushing and jostling. A plastic box with crushed paper or other padding inside is useful.

A printer is essential if you want to have printed copy of tables and documents during the investigation. Printed output is also a useful form of backup in case something should happen to your disks or computer. If the printer's weight is an absolute limitation, you should probably invest in one of the expensive but convenient IBM-compatible lap-top computers and an accompanying miniaturized printer.

4. **Traveling with a computer.** If your computer is on the heavy side, a two-wheeled luggage cart is a good investment. Generally, one can carry the computer in the cabin of an airplane and put the printer in a solid suitcase for checking. Most airlines will not let you insure the computer as checked luggage unless it is packed in the original shipping container, which may be quite bulky.

Airlines have varying rules about the use of laptop computers during flight. Be sure to check with airline staff before using the computer in the air.

If you take a computer out of your own country, it is important to be able to prove to customs officials that you purchased it before traveling and therefore need not pay importation duty when you return. Customs officials will provide a receipt as you leave your own country certifying that you left with the computer, or you can carry a sales receipt or other evidence of place of purchase to show when you return.

Some older desktop computers require that you run a program to "park" the heads on the hard disk drive before they can be moved safely. On laptops and most newer computers, this happens automatically when you turn off the power. Check the documentation if you are not sure.

Passing magnetic disks, including the hard disk inside a computer, through the x-ray machines in airports should not normally be harmful if done only a few times

and if the machines are relatively new. When possible, however, it may be better to have the box of disks hand inspected rather than taking any chances that the security attendant will irradiate your particular piece of luggage longer than the factory-prescribed time and thus exceed the dose that is safe for magnetic media. The magnetic fields generated by the motors that drive the moving belt on x-ray machines are said to be more hazardous and unpredictable than the x-rays.

Metal detectors may generate magnetic fields, however, and disks should not be walked or passed through the metal detector or scanned with the hand variety of metal detector.

Normally, when travelling in a car, put computer equipment in the trunk where it is out of sight; in hot weather, however, you should be as careful of heat damage to floppy or hard disks as you are with film. Never leave a box of disks on the dashboard or seat in the sun, and if the computer has a hard disk, be sure to protect it from getting cooked by the sun. Black or dark-colored containers are particularly apt to heat up in the sun. To avoid problems, treat computer disks as you would valuable exposed film or perishable laboratory specimens.

Many countries have 220- to-240 volt, 50-cycle electricity. Newer portable computers may adapt automatically or require that you change a switch. Some older computers may require an external transformer to supply 120 volts. You should consult the instructions for the computer before the trip and obtain advice if necessary. If you visit a country with 220- to-240 volt electricity, it is likely that the plug on your power cord will not be the same as the style used. Adapters can be purchased before the trip, or a suitable plug can be purchased in the new country and installed on the end of the cord. In emergencies bare wires pushed into the socket holes and secured with toothpicks or other insulating material may work, but this method is dangerous and is definitely not recommended.

5. **Protecting disks and making adequate backups.** Disks should be protected from fingerprints, scratches, coffee, magnets, sharp bending, and denting by firm objects like ball-point pens. Electric motors are a source of magnetic fields, as illustrated by the well-known story of computer tapes on a shelf near the floor that were erased when a janitor with a large floor polisher passed by.

A good rule of thumb for making backup disks is to back up often enough so that you would not be seriously handicapped by losing the data not yet backed up. Many people would interpret this to mean transferring data kept in the computer's memory to a disk about every 20 minutes and transferring data on a hard disk or floppy disk to another disk or tape after a half-day session.

A reasonable system for backup disks is to have one original and two backups, used in rotation. The oldest backup is overwritten each time a backup is made, leaving

the second backup disk intact in case something goes wrong during the copying process.

The backup copies should be carefully labelled and preferably kept in a different place from the originals. If your computer is stolen, it would be a shame to have the results of the investigation go with it because the diskettes were kept nearby. On a large or prolonged investigation, you can transmit data or mail disks to your home office as extra backups.

6. **Having access to a backup computer if yours fails.** It is useful to identify another computer with similar hardware and software that can be borrowed or used to read data in case the primary computer fails. This is a good reason to use commonly available brands of computers rather than "odd-ball" or "one-of-a-kind" machines.

7. **Knowing where to find hardware and software expertise when needed.** Often a particular problem can be solved immediately on the phone by the right technician. Most larger software companies offer telephone support for users of their packages; usually the telephone numbers are in the manual. The Epi Info support number is in the front of this manual.

Computer-knowledgeable people in your own organization may be helpful. Most larger communities now have amateur computer clubs, and the president may be able to steer you to a person who can solve a particular problem. A local university or technical college may have computer-knowledgeable staff members who might help out in an emergency or allow you to use local equipment. Computer-store personnel vary greatly in their knowledge, but a few well-placed questions may help separate the wheat from the chaff. Generally, if repairs on a computer are contemplated, you will need to call your administrative support office to find out how authorization and payment are arranged.

8. **Arranging an appropriate computer workplace.** This may be a hotel room or an office, but it should be secure from theft, noise, and interruptions. If it has a window, be sure to protect the computer and disks from direct sun. Glare on the computer screen is sometimes a problem and can be reduced with a polarized or other glare screen in front of the display. Light sources both behind and in front of the computer should be diffuse and modest in intensity. Very dark or very bright conditions in either location seem to increase eyestrain when a person is working at the computer.

There should be a reliable source of standard electricity, with a grounded (three-prong) electrical outlet. If the electricity is in a residential setting, be sure that the circuit you are on does not contain high current-consuming devices such as toasters or electric motors, which can create voltage drops or spikes. If you will be using a

telephone line for a modem, there must be a way to connect the modem to the line, i.e., the type of plug must be compatible or you will need to locate an adapter that makes it so.

9. **Taking periodic breaks during data entry and text processing.** When frustrated, get advice or do something else for a while. Prolonged computer use can lead to eye fatigue, mental fuzziness, or headaches. Taking a break every hour or two seems to help, and may prevent costly mistakes. Some people find that making a conscious effort to blink every few minutes helps to prevent eyestrain.

One source of neck and back fatigue is improper orientation of the screen and keyboard for the operator. Your line of vision should be perpendicular to the screen, with the screen slightly below eye level. If you wear *bifocals*, you may find that your head is tilted far back to focus properly and that a few millimeters off will result in distortion. Be sure you are directly in front of the screen, in this case. If you use a computer often, investigate having special "computer" glasses prescribed, which are similar to reading glasses but suited to reading a computer screen about 18 inches away and slightly below the horizontal. When Benjamin Franklin invented bifocals, he was obviously thinking of reading a book on one's lap, not a computer on the table ahead.

10. **Doing the simple things first and the complex later in processing data**.   If the computer is not the right instrument to do the job, do it manually. Many epidemiologists, when first using a computer, go through unnecessary complexities in trying to make the computer select cases according to a logical definition, such as "any three major symptoms and two minor ones, with onset between June 1 and July 15." If you are dealing with only a few hundred cases, it may be easier to leave a blank on the questionnaire in which you record the "group" as "case" or "non-case" rather than trying to learn how to have the computer do the Boolean logic to classify cases automatically. With thousands of cases, it might be more efficient to have the computer do the work, but in many outbreak investigations, the questionnaires can be sorted into piles and marked in a few minutes before data entry, possibly avoiding hours of complex programming and debugging. Computers can be made to do almost anything with appropriate programming, but some things are not worth attempting for small stakes.

Since computer processing is fairly rapid, it is preferable to do simple frequencies and two-dimensional tables on all the items in your questionnaire before proceeding to stratification and other more complex procedures.

11. **Pretesting methods of transferring data to other programs or other sites before you rely on them.** Some experiments with microcomputers in outbreak settings have resulted in frustration because an investigator unknowingly used a proprietary program to enter data, and the statistical support staff at the home

363

office--not having a copy of the program--had to spend several weeks to convert the data into a form that could be used in their available statistics program. It is important to understand file formats and compatibilities when using commercial software.

File formats are a veritable tower of Babel, and sometimes a small difference can be as fatal as a large one. Data entered in Epi Info can be exported in any of 15 different formats for use in other programs. Some commercial electronic mail systems require that files be entirely in the ASCII format. ASCII (pronounced "askee" and standing for American Standard Code for Information Interchange) is a set of 128 codes for letters, numbers, punctuation, and other characters, which is internationally accepted. Other networks and direct computer-to-computer transfer allow sending "binary" files that can contain any type of character codes.

If you intend to send files to someone else for processing or to process them yourself in another system, be sure to check on the details, and preferably do a practice run with sample data to make sure that the transfer is smooth. Ask your favorite computer guru to help, if necessary, and then test the results before you do the real thing under pressure.

12. **Trying again if communicating over telephone lines fails.** Line noise and the complexity of ports, stop bits, data bits, baud rates, and other variables can make telecommunication a test of skill and patience. Once the right combination has been found, connections usually become easy and reliable. Sometimes, however, noise on a telephone line interrupts frequently, and it may help to try another line, perhaps in a different building, or wait until evening when the line may be quieter. Be sure that all connections are tight and free of corrosion and that AC power cords do not lie near or across the telephone cord to minimize noise on the line. If the telephone line has "call waiting," it must be turned off by dialing "*70" on a touch-tone phone or 1170 on a rotary phone.

13. **Notes for CDC researchers performing official investigations** (information provided by appropriate offices at CDC). The Department of Health and Human Services and the Office of Management and Budget (OMB) discourage the collection or retention of full names or Social Security numbers except in very limited instances. The usual procedure is for individually identified data to remain with the state or local health department, with the permanent CDC file using a "Case ID" or similar field as the identifier.

Any collection of information sponsored by CDC (whether through contract, cooperative agreement, or in-house study) involving more than nine respondents must be reviewed and approved by OMB; however, CDC has OMB approval to collect information in emergency situations. (Normally, in non-emergency situations, investigators should reserve 4 months for receipt of OMB approval

before any data are collected.) The OMB clearance number, 0920-008, should be displayed on the top right-hand corner of all data-collection instruments (e.g., questionnaires) used in an emergency investigation. Upon completion of the data collection, a copy of the instrument and instructions, the number of respondents, and the estimated time for completion of the data instrument should be sent to the CDC Reports Clearance Officer, Mailstop D24, CDC, Atlanta, GA 30333.

Because some epidemic investigations involving humans are considered research and some are not, investigators should consult with their Center/Institute/Office Human Subjects Review contact person regarding the possible requirement for review and approval by an Institutional Review Board for Human Subjects Research before collecting data.

```
Dataset:  C:\EPI6\COMBINED.REC (12 records)          Free memory: 268K
Criteria: ATE = "+"
─────────────────────────── Output - Screen ───────────────────────────
HAMBURGER Y!
File closed. (Automatic ROUTE SCREEN)
String variable created.
String variable created.
FRENCHFRIEN!
FRENCHFRIEY!
FRENCHFRIEY!
FRENCHFRIE !┌───────────────── Error ─────────────────┐
File closed. │ (Line 29) FOODS is an invalid field!  Press ESC. │
String varia └──────────────────────────────────────────┘
String variable created.
SALAD     Y!
SALAD     N!
SALAD     Y!
SALAD     Y!
File closed. (Automatic ROUTE SCREEN)



EPI> run test.pgm



F1-Help  F2-Commands  F3-Variables  F4-Browse  F5-Printer on  F9-DOS  F10-Quit
```

Chapter **31**

Limitations of Speed and Memory; What To Do When Problems Occur

Hardware and Software Requirements and Appropriate Use of Epi Info

Epi Info programs will run on IBM-compatible microcomputers with 640 kilobytes (K) of random access memory (RAM) and either two floppy disk drives or a floppy and a hard disk. A hard disk is recommended for speed and convenience. The maximum size of questionnaires and check files is memory dependent. The maximum size of files that can be SORTed or RELATEd in ANALYSIS is no longer RAM dependent, since the hard disk is used for the necessary index files if necessary. We have sorted 1.5 million records using Version 6 of ANALYSIS (although it took some hours).

In practice, questionnaires of over 300 variables have been processed, and larger ones may be possible, although such large questionnaires are seldom desirable from an epidemiologic standpoint. Questionnaire size is limited to 500 lines by the programs. If you run into memory limitations, be sure that memory resident programs such as network drivers are not running at the same time, since they effectively reduce the size of the computer's memory. A commercial memory manager such as QEMM may allow such programs to be loaded into memory above 640K and free more memory for Epi Info. Running Epi Info programs directly from DOS without using the menu will also save a small amount of memory space (about 12 K).

Epi Info uses memory above the 640 K limit of conventional memory for overlay modules. This means that if you have extra memory, some programs will run faster when you call for a function that is not often used, and may therefore have been placed in an overlay (.OVR) module to be loaded into conventional memory when needed. This occurs much faster from higher memory than from the hard disk. The ENTERX program can also use higher memory for other purposes.

Files produced in ENTER and EXPORT can theoretically contain up to *2 billion*

records, given adequate disk capacity. When entering large data sets, it is advisable to divide the data into several files; the ANALYSIS program can treat them as one file (see READ in the ANALYSIS Reference Chapter 36). Be sure to exit from ENTER and periodically copy the data file to a backup disk, in case someone spills coffee on a disk or loses a file. ENTER files are written to the disk after each record is entered, and theoretically a power failure can result in the loss of only the record being entered. Nevertheless, computers occasionally follow Murphy's law and make a disk unreadable for complex reasons. Put large data sets into more than one file, and make backups as you proceed.

The size of data sets that should be manipulated is dependent on the types of hardware and software that are available in addition to the capacity of the programs themselves. The kind and frequency of processing and the patience of persons doing the processing are also considerations. Very large data sets are better processed on a mainframe computer after the file has been exported in an appropriate format.

We find that doing frequencies with 200,000 records of 32 bytes each in Epi Info on a 386-type microcomputer takes a little over 3 minutes and would therefore be reasonably practical. Speed is more or less inversely proportional to record size, however, and 600-byte records might take 10 to 20 times as long.

Coping with Errors

Troubleshooting often begins by identifying whether the problem is a hardware or software problem. The easiest way to rule out hardware problems is to run the program on another computer. If the error occurs only on one computer, it may be hardware related; if it occurs on both computers, software errors are more likely.

Occasionally you may see an error message ending with "Abort, Retry, Ignore?" This is a message from DOS usually indicating that a printer is not functioning or that a disk cannot be read or written to. Check that the printer cable is securely connected, that the printer is turned on and supplied with paper, and that the paper moves freely through the mechanism. Be sure that the door to the active disk drive is closed and that the disk in the drive is the one needed. Then press "R" for retry.

A program may stop running and display a message such as:

```
Runtime error 203, PC=BD47
Program aborted
```

or simply return to the EPI6 menu. If the latter happens, try running the same program or giving the same instruction again, but this time run the Epi Info

program, such as ANALYSIS, from the DOS prompt without using the EPI6 menu, by typing the name of the program. This time, if the error occurs, the program will not return to the EPI6 menu, and you will be able to see the error message on the screen. Write down the entire message, or press the <PrtSc> key to print it, and call or write to one of the Epi Info information sources listed in the front of the manual.

Turbo Pascal error messages like the one above are coded. The first number given tells the type of error, and the second tells the location in the program where the error occurred. As an aid to diagnosing difficulties, a few of the common Turbo Pascal error codes are given below, even though you may never see such messages in your use of Epi Info.

Selected Turbo Pascal Error Messages

Error 02A	file cannot be found. Be sure that the entire Epi Info system is installed on the disk(s) being used and that none of the files have been renamed or erased.
Error 04	There are too many files open at one time. DOS allows only a certain number of files to be open at one time. This is specified by including the statement FILES = 20 or some other suitable number in the CONFIG.SYS file. This file is probably in the root (\ ) directory or the \ DOS directory in your computer. The Epi Info INSTALL program automatically inserts the FILES = 20 statement, but if you did not use INSTALL, you may need to insert the statement in the CONFIG.SYS file using EPED.
Error 101	The disk is full. Return to DOS and use DIR to check the amount of disk space remaining. Use a new floppy disk or erase old files to make space.
Error 106	An item being read from a file should be a number but contains non numeric characters. This may mean that the data file has been damaged or that the questionnaire does not properly describe the data in the file--when using the IMPORT program, for example. Check numeric fields to be sure that they are properly placed and that the data items are all numeric.

Error 201	Range check error; please notify an Epi Info staff member.
Error 202	Stack overflow error; please notify an Epi Info staff member.
Error 203	"Heap/stack collision." This means that the program has run out of random access memory. If the computer has only 256 K of memory, try running ANALYSIS or EPED (the big memory users) directly from DOS by typing ANALYSIS or EPED at the DOS prompt. Memory can also be exhausted by an error in an ANALYSIS program statement. Check the wording of your commands against those in the manual, particularly for SELECT, IF, and LET statements. Simplify your ANALYSIS program by putting asterisks (*) in front of the more complex statements to remove them temporarily from the program and identify which statement might be the offender.
Errors 200, 205, 206, 207	These indicate "floating point overflow," division by zero, taking the square root of a negative number, taking the logarithm of zero or a negative number, or difficulties with the 80x87 coprocessor. If these occur, press the PrtSc key to make a printed record of the error, and make careful notes of the conditions under which it occurred. Save a copy of the data file which caused the error. Then call or write the Epi Info staff for further assistance.

Obtaining Assistance

For further assistance, first consult other Epi Info users in your own organization or with whom you are acquainted at the regional, state, or national level. The Epi Info Helpline can provide limited assistance though voice telephone, FAX, or E-mail through the numbers listed on the first page of this manual. Unfortunately, resources do not allow the assistance to include writing programs or extensive programming consultation. When possible, the Helpline will provide the names of consultants who can provide such services for a fee.

```
Dataset:  C:\EPI6\OSWEGO.REC (75 records)              Free memory: 295K
Criteria: All records selected                        Time:    0.11 sec
┌──────────────────────── Output - Screen ────────────────────────────┐
│                                                                      │
│                    ILL                                               │
│ VANILLA     │    +     -  │ Total                                    │
│             │             │                                          │
│        + │   43    11  │   54                                        │
│        - │    3    18  │   21                                        │
│             │             │                                          │
│    Total │   46    29  │   75                                        │
│                                                                      │
│                      Single Table Analysis                           │
│                                                                      │
│ Odds ratio                                                   23.45   │
│ Maximum likelihood estimate of OR (MLE) (2)                  22.15   │
│ Exact 95% confidence limits for MLE (2)         5.22 < OR < 138.39   │
│ Exact 95% Mid-P limits for MLE (2)              5.93 < OR < 109.15   │
│ Probability of MLE >= 22.15 if population OR = 1.0 (2)  0.00000026   │
│ <more>                                                               │
├──────────────────────────┌────────────┐────────────────────────────┤
│ EPI> read oswego         └────────────┘                             │
│ EPI> tables vanilla ill                                             │
│                                                                      │
│ F1-Help  F2-Commands  F3-Variables  F4-Browse  F5-Printer on  F9-DOS  F10-Quit
```

<div align="right">

Chapter 32

</div>

Statistics: Understanding the Results

Overview

The FREQ and TABLES commands provide automatic calculation of statistics for categorical data such as sex, race, risk factor, and disease presence or absence. STATCALC and EpiTable are epidemiologic calculator programs that produce statistics from summary data entered on the computer screen. Although many of the commands in STATCALC are also included in EpiTable, STATCALC is included with Version 6 of Epi Info for those who are familiar with its features and wish to use it for quick calculation or to use the examples in the help files for epidemiologic teaching.

Another command called MEANS produces statistics suitable for continuous or ordered data such as age, height, and weight. The MATCH command does analysis of matched case-control studies, and REGRESS provides multiple linear (not logistic) regression.

This chapter contains a series of notes on the output of ANALYSIS and later a more detailed explanation of the calculations that lie behind the results. Although the programs have been checked against manual calculations and other computer programs to a certain extent, this is no guarantee of their accuracy under all conditions.

Any result which does not seem reasonable should be checked against other independent methods before being accepted. Since it is easy to produce meaningless results with a computer by applying tests in an inappropriate way, you should seek expert statistical advice whenever there is doubt about the appropriateness of a test or a result needs interpretation. Statistical significance is, of course, not a substitute for biological plausibility.

Annotated Examples of Output From ANALYSIS

The output from ANALYSIS shown in this section `in this typeface` was obtained by READing files supplied with Epi Info and giving the commands shown in **bold type**. The accompanying comments provide explanation of the output for

the situations illustrated. Reference to statistical textbooks and consultation with experienced statisticians will also help to interpret the results.

Example: Obtaining confidence limits for simple proportions.

A "binomial proportion" is a simple ratio, such as 82/149. Sometimes, when doing a FREQuency, it is convenient to know the confidence limits associated with such proportions. By using the "/C" parameter after the FREQ, the confidence limits will be produced for each value of the frequency.

We will use the survey file, EXAMPLE, to illustrate the point, although, as described elsewhere, the survey is actually from a stratified cluster sample, and should be analyzed using the CSAMPLE program. Pretending that EXAMPLE is from a simple random-sample survey, however, we give the commands:

EPI6> **READ EXAMPLE**
EPI6> **FREQ SMOKE /C**   Command for a frequency with confidence limits

The results (in part) are as follows:

```
SMOKE | Freq  Percent   Cum.  95% Conf Limit
------+-------------------------------------
1     |   83   24.6%    24.6%    20.2%-29.7%
2     |  254   75.4%   100.0%    70.3%-79.8%
------+-------------------------------------
Total |  337  100.0%
```

SMOKE = 1 for 83 or 24.6% of the 337 respondents. If large numbers of samples were taken from the same population, 95% of the results would be expected to fall within the limits 20.2% to 29.7%. These confidence limits might be reported to give an idea of the precision of a simple random survey.

The confidence limits are obtained by the Fisher exact method if the denominator is less than 1000 and several other conditions are met, and by the Fleiss Quadratic method otherwise. These confidence limits may be used to give an idea of the precision of a survey based on a simple random sample.

Confidence limits are calculated only if the values of the variable are numeric and if the FREQ command is followed by "/C" as in:

FREQ SMOKE /C

Example: Two-by-Two Table

EPI6> **READ OSWEGO**   Tells ANALYSIS to work with OSWEGO.REC
EPI6> **TABLES VANILLA ILL**

Calls for a cross tabulation of the fields VANILLA (Vanilla ice cream consumption) and ILL (with gastroenteritis). Both are Yes/No variables. To make the tables match those in many epidemiologic textbooks, "Y" and "N" are automatically replaced by "+" and "-" during construction of the table.

```
                  ILL
VANILLA    |     +      - | Total
-----------+--------------+------
        + |    43     11 |    54
        - |     3     18 |    21
-----------+--------------+------
    Total |    46     29 |    75
```

There were 75 persons interviewed. Of these 46 were ill and 29 were not ill. Among the ill, all but 3 had eaten Vanilla ice cream. Among the well, 18 had not eaten vanilla ice cream.

```
                    Single Table Analysis

Odds ratio  (OR)                                              23.45
Cornfield 95% confidence limits for OR          5.07 < OR < 125.19*
                    *May be inaccurate
Maximum likelihood estimate of OR (MLE)                       22.15
Exact 95% confidence limits for MLE             5.22 < OR < 138.39
Exact 95% Mid-P limits for MLE                  5.93 < OR < 109.15
Probability of MLE >= 22.15 if population OR =        0.00000026

RISK RATIO(RR)(Outcome:ILL=+; Exposure:VANILLA=+)             5.57
95% confidence limits for RR                    1.94 < RR <  16.03

              Ignore risk ratio if case control study

                        Chi-Squares    P-values
                        -----------    --------
          Uncorrected:      27.22      0.00000018 <---
      Mantel-Haenszel:      26.86      0.00000022 <---
       Yates corrected:     24.54      0.00000073 <---
```

Table layout for further discussion

```
A     B     A+B         These two figures 43 and 18 (cells A and D, or the
C     D     C+D         Northwest-Southeast corners of the
A+C   B+D   A+B+C+D     table) support the hypothesis of an association.
```

The opposing evidence is in cells B and C, or the Northeast-Southwest diagonal-- the 11 non-ill eaters and the 3 ill non-eaters.

The odds of exposure among the diseased (A/C) is divided by the odds of the exposure among the non-diseased (B/D) to form the Odds Ratio or AD/BC. An odds ratio of 1.0, represents no association.

<div align="center">Single Table Analysis</div>

```
Odds ratio                                                    23.45
```
This is the product of the supporting cell counts divided by the product of the detracting cell counts, AD/BC.   The odds ratio of 23.45 indicates a strong association.

```
Cornfield 95% confidence limits for OR      5.07 < OR < 125.19*
```
This is an estimate of the confidence limits. For tables with fairly large numbers in each cell, the estimate is quite accurate. If the numbers are small, the exact confidence limits given below are more accurate. An asterisk accompanies estimates that may not be accurate.

```
Maximum likelihood estimate of OR (MLE)                       22.15
```
This is an estimate of the odds ratio for the true population represented by this sample, done by a mathematical method that finds a value for a population OR such that the probability of observing the given cell counts is higher than it would be for any other population OR.

```
Exact 95% confidence limits for MLE              5.22 < OR < 138.39
```
If a large number of such estimates were made (a large number of samples taken), 95% of them could be expected to fall within these confidence limits. In this particular case, 1.0--the Odds ratio when there is no association--is clearly outside the confidence limits, and one might conclude that there is much less than a 5% chance that this is due to random chance. The confidence interval includes only a wide range of strongly positive values.

```
Exact 95% Mid-P limits for MLE                   5.93 < OR < 109.15
```
A correction has been made for the fact that there are "gaps" between integers, i.e., that a series of counts is not continuous like a series of blood pressures or exact heights. The correction produces slightly "tighter" confidence limits, giving a greater chance of finding a statistically significant result. In this case, both types of confidence limits lead to the same conclusion. When the table has small numbers in one or more cells, however, which one should I believe? Statisticians differ on this point, and you should probably consult one or more.

```
Probability of MLE >= 22.15 if population OR = 1.0 (2)      0.00000026
```
This is similar to the Fisher exact test. The result given is called the "Lo p" if the MLE is < 1 and "Hi p" otherwise. If the OR equaled 1.0 in a large population and samples of the same size were taken 100,000,000 times, it is likely that 26 of the samples would predict an MLE this large or larger by chance alone.

<div align="center">376</div>

```
RISK RATIO(RR)(Outcome:ILL=+; Exposure:VANILLA=+)              5.57
```
Note the warning below that these results apply only to cohort and cross-sectional studies, NOT TO CASE CONTROL STUDIES. In the OSWEGO investigation, everyone who attended the meal was interviewed, and the study can be regarded as a cross-sectional study (or even a retrospective cohort study if you prefer). In any case, this risk ratio represents the risk of illness in the exposed (A/A+B, 43/54, or 79.6%) divided by the risk of illness in the unexposed (C/C+D, 3/21, or 14.8%), the result being 5.57. Note that small changes in cell C (moving two persons from cell C to cell B, for example) could change the Risk Ratio dramatically (from 5.57 to 14.59, or 2.6-fold) and the Odds Ratio as well (from 23.45 to 59.54, or 2.5-fold). Only when the disease is very rare in the population and a cohort or cross-sectional study is conducted, can the Odds Ratio and the Risk Ratio be expected to be similar in value.

```
95% confidence limits for RR (1)                    1.94 < RR <  16.03
```
This study is most consistent with a risk ratio of 5.57. At the 95% level of confidence, it is also consistent with values ranging from 1.9-16.0. Since the Risk Ratio is 1.0 when no association exists, and 1.0 lies outside of these confidence limits, we may conclude that an association is likely.

```
                    Ignore risk ratio if case control study
```
                        This is the warning noted above.

Suppose that we had not had access to a list of the guests and had taken the first 46 ill people who called in, asking them who else attended and was not ill so that we acquired 29 "controls" who were not ill, but we had no way of knowing how many attended the meal (perhaps it was hundreds). That would be a case control study. Note that we would now not have real denominators to calculate rates of illness in the exposed and unexposed, since we took samples of ill and well from a pool of unknown size. Hence the Risk Ratio would have no meaning.

```
(1) Greenland and Robins  Biometrics 1985;41:55-68
(2) Martin, D and Austin, H  Epidemiology 1991;1(2):359-362
(3) Rothman, K 'Modern Epidemiology' 1986;p. 222
```
(References)

Now suppose that we suspect "confounding by SEX" (in the epidemiologic sense). Specifically, perhaps the rate of illness is higher in females (it is, 68%) than in males (47%), and coincidentally females happen to prefer vanilla ice cream. This could lead to the association discovered in the VANILLA ILL table through "confounding."

Example: Stratified Mantel-Haenszel Analysis

The simplest analytic tool for dealing with confounding is the Mantel-Haenszel Stratified Analysis. It divides the analysis into separate tables for each value of the "stratifying" variable (SEX), and then combines the results in a way that removes the effect of confounding. If males and females have quite different odds ratios for VANILLA ILL tables, however, another phenomenon, called "interaction" or "effect modification" is present, and the Mantel-Haenzel results must be taken with a grain of salt, or not at all. A test to see if this is happening is included in the results. To stratify by SEX, we give a command in the format:

> **TABLES** <Exposure> <Illness> <Confounder>, in this case,

> EPI6>**TABLES VANILLA ILL SEX**

It is possible to use more than one field as the stratifier or potential confounder. If you suspect that boys and girls, men and women, might have different responses to the exposure, you might define another variable called AGEGROUP, recode AGE to AGEGROUP so that the values were ADULT and CHILD, and then do TABLES VANILLA ILL AGEGROUP SEX. This would produce a table for each combination of AGEGROUP and SEX, or ADULT M, ADULT F, CHILD M and CHILD F. With the modest number of interviews in this study, stratification to produce more than two tables quickly runs into "small cell" problems, and is not advisable, although there is no harm in experimenting to see what produces useful results.

The results are:

```
** Beginning Stratified Analysis **


            SEX =F                 A table for females
              ILL
VANILLA    |    +     -  | Total
-----------+-------------+------
        + |   27     4  |   31
        - |    3    10  |   13
-----------+-------------+------
    Total |   30    14  |   44

                        Single Table Analysis

                            Stratum 1
Odds ratio                                             22.50
Cornfield 95% confidence limits for OR        3.39 < OR < 184.37*
              *May be inaccurate
Maximum likelihood estimate of OR (MLE) (2)            20.14
Exact 95% confidence limits for OR (2)        3.44 < OR < 168.77
Exact 95% Mid-P limits for OR (2)             4.09 < OR < 128.37
Probability of MLE >= 20.14 if population OR = 1.0 (2) 0.00008138

RISK RATIO(RR)(Outcome:ILL=+; Exposure:VANILLA=+)        3.77
```

```
95% confidence limits for RR (1)              1.39 < RR <  10.28
```

```
           SEX =M                      A table for males
                          ILL
VANILLA    |     +     -  | Total
-----------+-------------+------
       +   |    16     7  |    23
       -   |     0     8  |     8
-----------+-------------+------
   Total   |    16    15  |    31
```

                              Single Table Analysis

                                  Stratum 2
```
Odds ratio                                                ??????
Maximum likelihood estimate of OR (MLE) (2)               ??????
Exact 95% confidence limits for OR (2)        2.77 < OR < ??????
Exact 95% Mid-P limits for OR (2)             3.69 < OR < ??????
Probability of MLE >= ?????? if population OR = 1.0 (2) 0.00081572
```

Note that the odds ratio cannot be calculated because there is a zero in cell C, and the results therefore are shown as "??????". Although it is tempting to say that this means it is "very large" or "infinite," consider the case where cells A and D (supporting the association) each contain 1, and cell B (against the association) contains 29, with cell C again being 0. The odds ratio is still not calculable, but no one would claim a positive association from such a table. Hence, it is better to say that the Odds Ratio cannot be calculated and to draw no conclusion from this fact alone. It is important to note however, that perfect association, (0 in both B and C) will also lead to an incalculable (undefined) odds ratio, and that this is definitely not a reason to discard the study. In such cases the confidence limits excluding 1.0 and the small p value as above are the items of interest.

```
RISK RATIO(RR)(Outcome:ILL=+; Exposure:VANILLA=+)         ??????
95% confidence limits for RR (1)              ?????? < RR < ??????
```

That pesky 0 also prevents calculating the RR

                    Ignore risk ratio if case control study
                        (Chi squares and Fisher exact for males shown here.)

            ** Summary of 2 Tables With Non-Zero margins **
                           N =   75

                              SUMMARY ODDS RATIO
```
Crude OR                                                   23.45
```

This will be the same OR obtained from TABLES VANILLA ILL unless: a) SEX has some missing values and IGNOREMISSING is ON, or b) One or more of the tables has a zero marginal (only ILL, only non-ILL, only eaters, or only non-eaters). In that case the N above could be less than 75.

```
Mantel-Haenszel weighted Odds ratio                              37.64
```

The odds ratio that combines the two tables is even higher than the crude odds ratio. Because the two are different, we say that confounding was present, and because the MH Odds Ratio is higher, the confounding tended to conceal the effect. In this study, however, both results are so dramatic that the outbreak is now in its 5th decade as a teaching example. Odds ratios of 10 or 20 are not uncommon in foodborne outbreaks, however.

```
Maximum likelihood estimate of OR (MLE)                          28.58
Exact 95% confidence limits for MLE               6.01 < OR < 207.50
Exact 95% Mid-P limits for MLE                    6.92 < OR < 160.22
Probability of MLE >= 28.58 if population OR = 1.0      0.00000011
```

These are stratified versions of the statistics produced for single tables, thanks to Dr. David Martin, who produced computerized programs in Turbo Pascal fast enough so that they could be included in ANALYSIS. If the numbers in the tables are very large, you may notice that the program falls back to doing the Cornfield approximation for the confidence limits. Luckily, when the numbers in each cell are large, the Cornfield approximation is very accurate, and this should not be a problem. The exact calculations shown here are most useful when the numbers in one or more cells are small, and the Cornfield tends to be inaccurate.

```
Unable to test for interaction due to zero cell values.
```

Alas--but not surprising since we would be testing to see if 20.14 is different from "???????".

```
                   SUMMARY RISK RATIO (RR)
               (Ignore if Case-Control Study)
Crude RR without stratification                              5.57
Summary RR of (ILL=+) for (VANILLA=+)                        5.73
95% confidence limits for RR (1)               1.99 < RR <  16.51
```

If Crude and Summary values are similar, confounding is not large, but 23.45 and 37.64 are not what you would call similar, so we conclude that confounding was present, and that we should use the Mantel-Haenszel stratified results--the odds ratio of 37.64.

Interaction is present if the odds ratios in different strata differ significantly, as in this case, where the odds ratios for females and males are 22.5 and ???????. However, the odds ratio for females is close to the 23.45 obtained for the (crudely) combined groups, and one could maintain that whatever the odds ratio might be for males, it did not change the overall total.

What should you do if the odds ratios differ dramatically (or in a statistically significant way)? There are two possibilities:

   a. Set up the stratification in a different way by combining groups, etc., to
      reduce the number of differing strata to a minimum.
   b. If strata still differ, report the result for each of the differing strata (e.g.,
      males and females) separately, rather than in a combined result.

When should you use logistic regression (even though you are not doing graduate work in epidemiology)? Answer: if the outcome variable is dichotomous (e.g., ill or not ill), and there is more than one major predictor variable (risk factor, exposure) of interest, or if interaction is present. With only one predictor variable and no interaction, Mantel-Haenszel Stratified analysis can compensate for the confounding if the number of confounders is small. If there are many confounders, the number of strata becomes large and each one contains small numbers, so that Mantel-Haenszel analysis becomes impractical.

Experts say that you should always do simple and stratified univariate analysis such as that provided by ANALYSIS before you embark on logistic regression. It gives a feeling for the dataset in a simple setting. If the results of logistic regression differ radically from the univariate analysis, it is an indication to check the logistic regression methods to be sure that there is not some underlying misconception or mistake.

```
               ** End of Stratified Analysis **
                      That's all, folks...
```

Example: Continuous Data--The MEANS Command

The TABLES command deals with categorical (discontinous, discreet, coded) data--items with Yes/No answers, codes, or even integer categories, but no decimal points.

Ages, being integers, can be thought of as either continous or discontinuous (discreet, coded) data. Generally, ages are grouped in categories, such as 5- or 10-year age groups if they are to be analyzed with the TABLES command. When there are many categories, as in 1-year age groups, or when the data points are truly continuous, as with real numbers (height to the nearest 0.1 centimeter), the MEANS command is used. We will use it to see whether the ages of ill and well persons in the OSWEGO outbreak were similar, since a difference might affect the comparability of the exposure data.

We give the command:

      EPI6> **MEANS AGE ILL**

Calculate the means of AGE for all values of ILL, and do appropriate statistics.

```
                          ILL
    AGE         |     +      -  | Total
    ----------+------------+------
            3 |     1      0 |     1
            7 |     1      1 |     2
            8 |     2      0 |     2
            9 |     0      1 |     1
           10 |     1      0 |     1
           11 |     0      4 |     4
           12 |     1      0 |     1
           13 |     1      1 |     2
           14 |     0      1 |     1
           15 |     3      0 |     3
           16 |     1      0 |     1
           17 |     1      3 |     4
           18 |     1      0 |     1
           20 |     1      1 |     2
           21 |     1      0 |     1
           24 |     0      1 |     1
           25 |     1      1 |     2
           32 |     1      0 |     1
           33 |     2      0 |     2
           35 |     2      2 |     4
           36 |     1      2 |     3
           37 |     1      1 |     2
           38 |     0      1 |     1
           40 |     1      1 |     2
           44 |     1      0 |     1
           45 |     1      0 |     1
           48 |     1      0 |     1
           50 |     1      1 |     2
           52 |     3      0 |     3
           53 |     1      0 |     1
           54 |     0      1 |     1
           57 |     1      0 |     1
           58 |     1      0 |     1
           59 |     2      0 |     2
           60 |     1      0 |     1
           62 |     2      2 |     4
           63 |     1      0 |     1
           64 |     0      1 |     1
           65 |     1      2 |     3
           68 |     1      0 |     1
           69 |     0      1 |     1
           70 |     1      0 |     1
           72 |     1      0 |     1
           74 |     1      0 |     1
           77 |     1      0 |     1
    ----------+------------+------
       Total |    46     29 |    75
```

There are 46 ill and 29 non ill persons, for a total of 75

```
    ILL             Obs     Total     Mean   Variance    Std Dev
    +                46      1806   39.261    477.264     21.846
```

For the ill, there were 46 Observations. Multiplying the number of observations in each cell by the age for that cell (e.g., 1 x 77, 2 x 62, etc.), and adding the results

together gave a total of 1806. Dividing this by the total of 75 gave a mean of 39.261. The variance is a measure of the dispersion shown by a set of observations. For the ill persons, it is the sum of, for all ages:

(The age value - Mean age for the group)^2 (Cell count/46)

The standard deviation is the square root of the variance.

```
-                   29       955    32.931    423.709     20.584
```
Same thing for the non ill.

```
Difference                           6.330
```

The difference between the two means.

```
ILL         Minimum   25%ile     Median    75%ile    Maximum       Mode
+             3.000    17.000     38.500    59.000     77.000     15.000
-             7.000    14.000     35.000    50.000     69.000     11.000
```

These results may be useful in deciding whether the two sets of data are normally distributed, or are "skewed" to one side or another of the mean. The median (or 50th percentile) is reasonably close to the mean, and the ranges of the two (Minimum to Maximum) are more or less comparable. The modes (the ages with the most observations) are not near the median or mean, but the number of observations in a cell never exceeds 4, and this is therefore not a very reliable indicator. By the "eyeball" test, the data might be considered to be consistent with samples from a normal distribution. A plot using LINE AGE ILL might give further information.

```
                                  ANOVA
                    (For normally distributed data only)

Variation      SS   df         MS  F statistic   p-value    t-value
Between    712.655   1    712.655        1.560  0.213137   1.249147
Within   33340.732  73    456.722
Total    34053.387  74
```

The ANOVA results give the Sum of the Squares for variation between and within the two groups, ill and well. MS is the sum of the squares of differences from the mean (SS) divided by the degrees of freedom (df). The final result is the ratio of two variances and is called an F statistic, from which a p-value is calcu-lated. The p-value in this case is considerably greater than 0.05, the level we often choose for "significance," (but feel free to choose your own). We conclude that the ages in the two groups do not differ statistically (given this modest sample).

You may have encountered "Student's t test" and be wondering why the F statistic is given instead of the "t". The difference is that F can be used to evaluate the amount of difference among more than two sets of observations. If ILL had 3 or 4 categories rather than 2, the F statistic and the Analysis of Variance (ANOVA) would test whether there were significant differences between any 2 of the several groups (without telling which groups or what kind of difference).

In the present case, where there are two groups only, the "t" test is appropriate, and the "t" appears. Since the p-value derived from either the "t" or the F is the same when there are only two groups, you would report the "t" result and the p-value.

```
            Bartlett's test for homogeneity of variance
Bartlett's chi square =   0.119  deg freedom = 1   p-value = 0.729786
```

This is a test to find out if one of the assumptions required for ANOVA is satisfied. The assumption is that the variances for the groups being compared are similar. In this case the variances are 424 for the well and 477 for the ill group. The small chi square and large p-value in Bartlett's test suggest that there is no significant difference, and we therefore assume that:

```
        The variances are homogeneous with 95% confidence.
  If samples are also normally distributed, ANOVA results can be used.
```

The other assumption required by the ANOVA test is that the samples are normally distributed. We have examined this through inspection of the medians, percentiles, and modes, and concluded that this seems reasonable.

Because it requires the assumption of normal (parametric) distributions, the ANOVA is a PARAMETRIC test. In case these assumptions are not satisfied, the following test is also offered. It is a NON PARAMETRIC test. Non parametric tests are always appropriate, but may not be as powerful in detecting differences as parametric tests. Of course, if both tests show significant differences, there is less reason to worry about which one to use--the non parametric test may be quite enough and cannot be challenged on the basis of underlying normality assumptions.

```
Mann-Whitney or Wilcoxon Two-Sample Test (Kruskal-Wallis test for two groups)
```

This one comes with the names of five statisticians attached, since the tests for two samples were described first, and then Kruskal and Wallis developed a test for three or more samples that collapses into the same calculation when there are only two samples.

```
Kruskal-Wallis H (equivalent to Chi square) =      1.161
                       Degrees of freedom =          1
                               p value =    0.281226
```

384

Conclusion: The Mann-Whitney or Wilcoxon Two-Sample or Kruskal-Wallis test does not show a significant difference between the ill and well groups with respect to AGE. If it had, the p value would have been 0.05 or smaller.

Example: Matched Analysis

Suppose that a study of Toxic Shock Syndrome begins by selecting 41 cases of TSS. For each case, two controls are selected, matched by age, geographic area, and acquaintanceship. An IDNUMber is assigned to each group of cases and controls. Each subject is asked about the use of RELY brand tampons.

Since the data set is MATCHED, it is not adequate to compare the ill and well groups by using TABLES RELY ILL, since TABLES assumes that the ill and the well groups are random samples. In effect, each group of case and matched controls forms a separate stratum, and the analysis could be performed using:

```
READ MATCH.REC
TABLES RELY ILL IDNUM
```

This would produce 41 separate tables, one for each value of IDNUM. The Mantel-Haenszel weighted odds ratio summarizing the 41 tables would reflect the degree of association between use of RELY tampons and illness. The MATCH command in ANALYSIS actually performs this analysis, but for convenience sake, does not display the 41 tables. The MATCH command syntax is also slightly different and easier to understand. Here is how you would perform the analysis:

EPI6> **READ MATCH**   Read MATCH.REC, the sample file
EPI6> **MATCH IDNUM RELY ILL**

The results are:

```
               MATCHED ANALYSIS
                41 Matched Sets
                   N =   100
```

The analysis ignores non discordant groups, i.e., groups in which the result (exposure) for cases and controls is the same. If the case and both controls with the same IDNUM used RELY, or all did not use RELY, then the table has a zero marginal and is omitted both from the analysis and from the N above. To see which ones were omitted, you can use the command TABLES RELY ILL IDNUM to explore individual tables.

Note that the following results are the same with either the TABLES or the MATCH command.

```
                    SUMMARY ODDS RATIO
Crude OR                                                 15.59
Mantel-Haenszel matched Odds ratio                        8.01
Maximum likelihood estimate of MLE                        7.70
Exact 95% confidence limits for MLE         2.82 < OR <  26.26
Exact 95% Mid-P limits for MLE              3.04 < OR <  23.01
Probability of MLE >=  7.70 if population OR = 1.0 (2)   0.00000230
```

We conclude that a matched analysis produces a Mantel-Haenszel matched odds ratio of 8.01 or an MLE estimate of 7.7, with confidence limits excluding 1.0, and a p-value much smaller than 0.05, strongly suggesting an association between the use of RELY tampons and the occurrence of Toxic Shock Syndrome.

```
        Unable to test for interaction due to zero cell values
```

Example: Performing a Paired t-Test

There have been many requests for a t-test for paired data. The following method will accomplish this test without adding another command to those already in ANALYSIS.

Suppose that students are tested before and after an educational experience to measure the extent of their knowledge or skills. Since the same student is the subject of both tests, the data are matched, or, in fact, paired. We would like to know if there has been an improvement in scores or not between the two tests. The file SCORES.REC contains fields STUDENTNUM, SCORE1, and SCORE2.

The following commands (contained in SCORES.PGM) will perform the analysis of SCORES.REC.

```
    READ SCORES
    DEFINE DIFFERENCE ###
    DIFFERENCE = SCORE2-SCORE1
    FREQ DIFFERENCE
```

The output looks like this:

```
DIFFERENCE |  Freq  Percent   Cum.
-----------+----------------------
-14        |    1    2.6%     2.6%          These 7 students failed to
-13        |    1    2.6%     5.1%          benefit from the educational
-12        |    2    5.1%    10.3%          experience.
 -9        |    1    2.6%    12.8%
 -2        |    3    7.7%    20.5%
 -1        |    6   15.4%    35.9%
```

```
    0           |     2    5.1%    41.0%
    1           |     4   10.3%    51.3%        ---------------------------
    5           |     3    7.7%    59.0%
    7           |     4   10.3%    69.2%
   11           |     1    2.6%    71.8%
   12           |     1    2.6%    74.4%
   13           |     1    2.6%    76.9%        This group showed improvement.
   14           |     1    2.6%    79.5%
   20           |     4   10.3%    89.7%
   33           |     1    2.6%    92.3%
   46           |     1    2.6%    94.9%
   47           |     1    2.6%    97.4%
   54           |     1    2.6%   100.0%
  ----------+----------------------
     Total |    39  100.0%


       Total         Sum       Mean    Variance    Std Dev     Std Err
          39         285      7.308     250.113     15.815       2.532
```

The mean (average) improvement was 7.3 points.

```
    Minimum      25%ile      Median      75%ile     Maximum        Mode
    -14.000      -1.000       1.000      13.000      54.000      -1.000
```

The range of improvement was from -14 to 54 points. Since the median does not lie at the mid point between the 25th and 75th percentiles and the two extremes, the plot of differences is not symmetrical. The median of only 1 point suggests that the educational effort was not a practical success. Probably the median is a better measure for this asymmetric series than is the mean.

```
        Student's "t", testing whether mean differs from zero
        T statistic = 2.886,  df =    38   p-value = 0.00641
```

If this series could be taken as a random sample from a population, the p-value of less than 0.05 could be taken to mean that the mean (difference) is significantly different from zero. Because the results appear to be skewed, the validity of using this result is in question. The non parametric sibling of the "t" test, called the Wilcoxon signed-rank test (not included in Epi Info), might be done if further results are desired. Even without this, we conclude that, from the educational point of view, the educational experience, or the method of testing, leave something to be desired and should be revised.

The Calculations in Epi Info

The examples above give an idea how the statistics in Epi Info may be interpreted. Further details on appropriate use and interpretation are given in textbooks of biostatistics, several of which are listed at the end of this chapter.

Enough information is given on the underlying formulas in the following section and the references cited so that those familiar with statistics can understand the calculations, and appreciate the compromises and assumptions that have been made.

Tables--Statistics for Categorical Data

Statistical calculations will be done automatically by the TABLES command. They may be turned off by the command

<div align="center">

SET STATISTICS = OFF

</div>

in which case, the tables will be produced without statistics. The other commands that affect tabulation are listed under SET in Chapter 36.

Notation

```
                      Disease

                  ┌──────┬──────┐
                  │      │      │
                  │  a   │  b   │  H1
                  │      │      │
      Exposure    ├──────┼──────┤
                  │      │      │
                  │  c   │  d   │  H2
                  │      │      │
                  └──────┴──────┘

                    V1     V2     N
```

H1 and V1 represent horizontal and vertical totals in which disease or exposure is present; H2 and V2 are totals for exposure and disease absent; and N is the grand total of the counts in cells a, b, c, and d.

"Sum" means the summation of values from individual tables for all strata.
|x| indicates the absolute value of x.
^ means "raised to the power of." X^2 is X squared.

The type of calculation produced by the TABLES command depends on the size of the table and on its contents. Single 2-by-2 tables that are not stratified (e.g., TABLES CABBAGE ILL) result in a basic set of statistics, and stratified tables (TABLES CABBAGE ILL SEX) result in additional statistics summarizing all

tables in the group.

Calculations for 2-by-2 Tables

The odds ratio (a*d / b*c), relative risk (a/H1)/(c/H2), and, if possible, 95% confidence limits for these two values are displayed. A comment may appear that one or more of the confidence limits may be unreliable.

Expected values for a cell with the vertical coordinate i and horizontal coordinate j are calculated as:

$E = (H_i * V_j) / N$

If E is less than 5 for any of the four cells, a Fisher exact test is calculated for the table, and one- and two-tailed results are given. It is more conservative and, in most cases, more correct to use the two-tailed probability value unless the only hypothesis being tested is for either a positive or a negative association and the opposite result is not a possibility.

Three chi square values are given, derived from the uncorrected, the Mantel-Haenszel, and the Yates-corrected formulas. Values for p are then given for one degree of freedom. The formulas used are as follows.

$$\text{Chi square(uncorrected)} = N((a*d)-(b*c))^2 / (H1*H2*V1*V2)$$

$$\text{Chi square(Yates)} = N(|(a*d)-(b*c)|-0.5*N)^2 / (H1*H2*V1*V2)$$

$$\text{Chi square(Mantel-Haenszel)} = (N-1)((a*d)-(b*c))^2 / (H1*H2*V1*V2)$$

Confidence limits for the odds ratio in 2-by-2 tables are obtained by using Cornfield's approximation as described by Fleiss (*1*, pp. 71-75). Although this method produces an approximation to the exact calculation of the limits as described by Thomas (*2*), the agreement is quite good (*3*). The program uses a method described by Kleinbaum, Kupper, and Morgenstern (*4*, p. 305), and attributed to Mantel and Fleiss, to determine whether Cornfield confidence limits give values reasonably close to the exact values. If not, the result is labeled as possibly "inexact."

The relative risk (RR) is the ratio of the rate of disease in exposed persons to the rate of disease in unexposed persons, or:

$$RR = (a/H1)/(c/H2)$$

Note that the interpretation RR depends on how the codes for each variable cause information to be placed in the table. Analysis provides help with this interpretation by presenting the RR as, for example:

```
Risk Ratio(RR) (Outcome:ILL=+);(Exposure: VANILLA=+)
```

In some cases it may be necessary to recode the variables so that the table is properly oriented to provide a true RR. The TABLES command itself should be followed by variables in the order TABLES <Exposure Variable> <Outcome Variable>.

Confidence limits for the RR are calculated from the formula of Greenland and Robins given below for stratified tables (*5*).

The Fisher exact probability calculation for 2-by-2 tables in which one of the expected values is less than 5 is taken from Rosner, page 308 ff (*6*).

Calculation of probability value, p, from chi square for any degree of freedom, uses a formula given by Poole and Borchers (*7*).

Stratified 2-by-2 Tables

Tables are stratified when more than two variables are used in a table. In

TABLES SALAD ILL SEX RACE

SALAD will be the vertical variable, and ILL will be horizontal across the page. A separate table will be produced for each possible SEX-RACE combination or stratum. The set of tables will be separated from others by a line of asterisks. Mantel-Haenszel summary statistics are calculated for the series of tables as follows:

The first item given is the crude odds ratio. This is the odds ratio for the first two variables in the TABLES command using data in all strata which have nonzero marginal values. If a table has one or more marginal values of zero, it is left out of the calculation, since it will not contribute to the Mantel-Haenszel weighted odds ratio either.

The next item is:

Mantel-Haenszel (*8*) weighted estimate of the odds ratio =
        Sum(a*d/N) / Sum (b*c/N)

Comparing these two estimates of the odds ratio will indicate whether or not the stratifying variables were confounding variables. If the two odds ratios are the same, confounding was not present.

Confidence limits for the Mantel-Haenszel weighted odds ratio are calculated by using the formulas of Robins, Greenland, and Breslow (*9*):

Lower RGB confidence limit = exp (ln OR_{MH} -1.96 (Square root of V))
Upper RGB confidence limit = exp (ln OR_{MH} +1.96 (Square root of V))

$$V = Sum (P_iR_i) / 2(Sum(R_i))^{\wedge}2 + Sum(P_iS_i + Q_iR_i) / 2Sum(R_i)Sum(S_i) +$$
$$Sum(Q_iS_i) / 2(Sum(S_i))^{\wedge}2$$

$P_i = (a_i + d_i)/N_i$
$Q_i = (b_i + c_i)/N_i$
$R_i = a_id_i/N_i$
$S_i = b_ic_i/N_i$

The crude RR is calculated as for a single table showing the first two variables in the TABLES command, but omitting any stratum which has a zero value for any marginal total. The summary RR over all strata is calculated as:

Summary RR for stratified analysis = RR_{MH} = (Sum (aH2)/ N) /
        (Sum(cH1)/N)

Confidence limits for the summary RR are calculated from the formulas of Greenland and Robins (*5*):

Lower GR confidence limit = exp (ln RR_{MH} -1.96 (Square root of V))
Upper GR confidence limit = exp (ln RR_{MH} +1.96 (Square root of V))

$V_i = Sum (D_i) / (Sum(R_i) Sum(S_i))$

$D_i = (H1_iH2_iV1_i - a_ic_iN_i) / N_i^{\wedge}2$

$R = a_iH2_i / N_i$
$S = c_iH1_i / N_i$

For a single table, this reduces to the Taylor series confidence limits, which are used for the RR in single tables in ANALYSIS.

Finally, the stratified analysis shows the Mantel-Haenszel chi-square value, which applies to both the summary OR and the summary RR. The formula is as follows (*8*):

MH chi square = $(| Sum(a_i) - Sum(E(a_i)) | -0.5)^{\wedge}2 / Sum(V(a_i))$

in which $E(a_i) = V1_i*H1_i / N_i$

and $V(a_i) = V1_i*H1_i*V2_i*H2_i / N_i{}^{\wedge}2 (N_i-1)$.

The p-value is calculated from the MH chi square, using one degree of freedom.

Tables Larger Than 2 by 2

For tables of up to 50 cells, an overall chi-square value is calculated as:

Sum((observed-expected)^2/expected) for all cells.

Observed is the value in the cell. The expected value is calculated as described above for 2-by-2 tables. The number of degrees of freedom ((rows-1)*(columns-1)) is given, and the p-value, calculated as described above, is displayed.

The TABLES command will produce very large tables. It is difficult to interpret the practical meaning of chi square for a large table. Further statistics can be obtained by using the SELECT command to reduce the number of values of each variable analyzed and to produce more readily understandable tables.

Additional Exact Tests Available in STATCALC

For tables containing small numbers, the Cornfield approximations to the exact confidence limits produced by ANALYSIS may not be entirely accurate. The exact confidence limits contained in STATCALC are based on an algorithm and

FORTRAN program by Mehta, Patel, and Gray (*10*), programmed in Turbo Pascal by Eduardo Franco and Nelson Campos-Filho, of the Ludwig Cancer Institute, Sao Paulo, Brazil, and used with their kind permission.

STATCALC works for reasonably small numbers and up to about 20 strata. A program called StatXact for computing additional exact tests is available from Dr. Mehta, Harvard School of Public Health and Cytel Software Corporation (137 Erie Street, Cambridge, MA 02139; telephone 617-661-2011).

STATCALC does stratified analyses and will calculate summary odds ratios and relative risks and confidence limits for stratified tables.

MEANS--Parametric and Non Parametric Statistics for Continuous or Ordinal Data

The results of the MEANS command are shown in Chapter 9 on ANALYSIS. The calculations of the one-way analysis of variance (ANOVA) follow Rosner (*6*, p. 417), as do those for Bartlett's test (p. 435). The Kruskal-Wallis one-way analysis of variance by ranks is taken from the text by Siegel (*11*, p. 184 ff.).

The REGRESS Command in ANALYSIS

This command performs multivariate linear regression according to the methods in Draper and Smith (14, p. 84 ff).

The MATCH Command in ANALYSIS

The MATCH command uses the same calculations as the stratified Mantel-Haenszel analysis, stratifying by the identification number for each case-control group. This number represents a number of factors that have been matched so that the effect is similar to stratification on all the matching variables at one time.

The same results can be obtained by using the TABLES command, although TABLES will display a separate table for each case-control group and the tables are omitted from the MATCH command results. The two commands that follow, however, will give exactly the same results:

TABLES VAR1 VAR2 IDNUM
MATCH IDNUM VAR1 VAR2

if IDNUM is a number or text variable identifying each matched case-control group.

Sample Size Calculations in STATCALC

The sample size calculations for proportions in a descriptive study or survey use the following method:

Sample size=size of sample randomly selected from the population
Population=size of population which the sample is to represent
P=true proportion of factor in the population (guess)
D=(Maximum) difference between sample mean and population mean
Z=area under normal curve corresponding to the desired confidence level.

Confidence=	**Z**
.90	1.645
.95	1.960
.99	2.575
.999	3.29

The formula is:

Sample size=n/(1+(n/population))

in which: n=Z*Z(P(1-P)/D*D)

Reference: Kish (*12*).

The sample size formula for comparison of two groups of equal or unequal size is taken from Fleiss (*1*), pp. 44-45, formulas 3.19 and 3.20, and table A.2 on p. 259. These formulas do not assume matching of the two populations, and they apply to univariate comparisons of the differences in proportions (rates) in the two populations.

m=size of sample from population 1
r*m=size of sample from population 2
P1=true proportion of factor in population 1
P2=true proportion of factor in population 2
alpha=chance of falsely declaring the two proportions to differ ("Significance").
beta=chance of not detecting a difference which is present (could be called "Lack of sensitivity.")
1-beta=chance of detecting a real difference ("Power")

In the equations the values of c are as follows:

If 1-alpha is:	Then c[alpha/2] is:
.90	1.645
.95	1.960
.99	2.575
.999	3.29

If 1-beta is:	Then c[1-beta] is:
.70	-0.524
.80	-0.842
.90	-1.282
.95	-1.645
.99	-2.326

These correspond to Table A.2 in Fleiss, but it takes considerable attention to detail and to the example on p. 45 to learn how to look them up correctly.

The formulas are:

m=mprime + (r+1)/(r|P2-P1|)

in which "|" denotes absolute value and:

MPRIME=((((C[ALPHA/2]*SQRT1)-(C[1-BETA]*SQRT2))^2)/(R*((PR2-PR1)^2))

in which "[ ]" denotes a subscript and:

SQRT1=((R+1)*PB*QB)^0.5
SQRT2=((R*PR1*(1-PR1))+(PR2*(1-PR2)))^0.5
PB=(PR1+(R*PR2))/(R+1)
QB=1-PB

Trend Analysis in STATCALC

These methods are taken from Schlesselman (*13*), p. 203 ff.

References

1. Fleiss JL. Statistical methods for rates and proportions. New York: John Wiley and Sons, 1981.

2. Thomas DG. Exact and asymptotic methods for the combination of 2 x 2 tables. Computers in Biomedical Research 8:423-46, 1975.

3. Gart JJ, Thomas DG. The performance of three approximate confidence limit methods for the odds ratio. Am J Epidemiol 115:453-70, 1982.

4. Kleinbaum DG, Kupper LL, Morgenstern H. Epidemiologic research: principles and quantitative methods. Belmont, California: Lifetime Learning Publications, 1982.

5. Greenland S, Robins JM. Estimation of a common effect parameter from sparse follow-up data. Biometrics 41:55-68, 1985.

6. Rosner, BA. Fundamentals of biostatistics. Boston: Duxbury Press, 1982.

7. Poole L, Borchers M. Some common BASIC programs. Berkeley, California: Adam Osborne & Associates, Inc, 1977.

8. Mantel N, Haenszel W. Statistical aspects of the analysis of data from retrospective studies of disease. J Natl Cancer Inst 22:719-48, 1959.

9. Robins J, Greenland S, Breslow NE. A general estimator for the variance of the Mantel-Haenszel odds ratio. Am J Epidemiol 124:719-23, 1986.

10. Mehta CR, Patel NR, Gray R. Computing an exact confidence interval for the common odds ratio in several 2 x 2 contingency tables. J Am Statistical Assoc 80:969-73, 1985.

11. Siegel S. Nonparametric statistics for the behavioral sciences. New York: McGraw-Hill, 1956.

12. Kish L. Survey sampling. John Wiley & Sons, N.Y., 1965.

13. Schlesselman JJ. Case-control studies. New York: Oxford University Press, 1982.

14. Draper N, Smith H. Applied regression analysis, second edition. New York: John Wiley & Sons, 1981.

Analysis of Data from Complex Survey Designs

Introduction

Statistical analysis of survey data typically involves the need to produce a point estimate (ϑ) of some population value (ϑ) and an accompanying estimate of the variance of the point estimate (var(ϑ)). The analyst views ϑ as the best available estimate of the population parameter based on the survey data, and var(ϑ) is a measure of error of ϑ among all possible samples that the sampling design chosen for the survey might yield. In some sense var(ϑ) is a measure of the statistical quality of ϑ which, as much as ϑ itself, is of importance and relevance to the analyst.

Two complicating features of the sample design from a survey are relevant for analysis of data from that survey: (1) the selection probabilities for sample respondents, as determined by the process followed in deciding whether or not a member of the sampled population is to be chosen into the sample, and (2) the design's structure, as determined by the units to be sampled and how they are grouped in drawing the sample. The selection probability for a respondent quantifies the relative likelihood that on repeated applications of the sampling design that individual would be chosen in the sample. Design structure, on the other hand, is determined by the number of sampling stages and the definition of the sampling units is each stage of a cluster sample, as well as any stratification used to select the sampling units in each stage.

Selection probabilities are incorporated into analysis by using sample weights that are computed and added to each respondent record. Sample weights are inversely proportional measures of selection probabilities. Survey estimates which incorporate these weights are generally called "weighted estimates." For example, a weighted point estimate of a population mean for some measure of interest is

$$\vartheta = \sum_{i=1}^{n} w_i y_i \Big/ \sum_{i=1}^{n} w_i$$

where w_i is the sample weight and y_i is the measure of interest for the i-th sample respondent. The symbol, _ is used to denote summation, which for this estimate is over all n sample respondents. For wi that has been computed as the reciprocal of the selection probability for the i-th respondent, • can be viewed as a ratio of estimated totals. The numerator estimate the total of the y-variable among all

397

population members, while the denominator estimates the number of members in the population.

The focus of the remainder of this appendix is on variance estimation; i.e., the problem of estimating the variance of • among all possible samples that could be generated from the sample design. Variances for eight types of point estimates are needed for the MEANS and TABLES procedures in CSAMPLE. Each type involves one or two nonoverlapping population subgroups, or domains, and each is a function of some number of population totals (number of totals given in brackets):

    (1) Single domain mean [2],
    (2) Single domain proportion [2],
    (3) Difference of domain means [4],
    (4) Difference of domain proportions [4],
    (5) Ratio of domain means [4],
    (6) Ratio of domain proportions [4],
    (7) Odds ratio for exposure domains [4], and
    (8) Relative risk for exposure domains [4].

These eight types of estimates, in turn, fall into the following three categories:

    (1) Simple ratio,
    (2) Difference of ratios, and
    (3) Ratio of ratios.

The domains for which estimate types (1)-(6) are produced could be defined by one or more domain ("Crosstab") variables like gender, region, age group, or other socio-demographic measures. The exposure domains for estimate types (7)-(8) are usually defined by the health-related conditions or instances to which a person is exposed.

Several approaches have been suggested for producing variance estimates for analysis of survey data from complex samples (see Wolter, 1985). All are built in one way or another from the statistical notion of statistical replication. Under these various approaches, the overall survey sample either actually consists of a set of separate replicate subsamples, or the overall sample can be subdivided into a set of representative but overlapping replicate subsamples, each of which is statistically representative of the population sampled. Since under the latter classification the subsamples are not true replicates in the sense that each is independently chosen, the subsamples under this rubric are viewed as "pseudo replicates." The Designed Replication and Taylor Linearized Deviation approaches as presented here utilize true replication principles, whereas the Balance Repeated Replication and Jackknife approaches employ the notion of pseudo replication to obtain variances.

Taylor Linearized Deviation Approach to Variance Estimation

The Taylor Linearized Deviation (TLD) approach to variance estimation was adapted for use in Version 6 because of its statistical and computational advantages. This approach is based on the use of a Taylor series approximation for the difference between the point estimate and the population value it estimates. The particular version of the TLD approach, sometimes called the "propagation of variances" method, is usually attributed to Woodruff (1971), who first adapted the idea for complex point estimates from stratified samples with unequal selection probabilities and who also suggested an important computational short-cut.

The population value to be estimated is viewed as a function $\phi(\ )$ of a P-dimensional vector of population totals, U=(U1,U2,...,UP), whose point estimate is,

$$\vartheta = \phi(u), \text{ (1)} \tag{1}$$

where $u = (u_1, u_2, ..., u_P)$ is a vector of simple weighted estimates for the corresponding entries to U, such that E(u)=U.

Using the first-order (i.e., linear) terms of a Taylor series expansion for $\phi(u)$, we have

$$\phi(u) \text{ Å } \phi(U) + \underset{j=1}{\overset{P}{\_\_}} \phi'(U)(u_j - U_j) = \phi(U) + \phi_U d',$$
(2)

where $\phi_j'(U)$ is the first partial derivative of $\phi(U)$ with respect to U_j, d=u-U, and $\phi_U = [\phi_1'(U), \phi_2'(U),...,\phi_j'(U),...,\phi_P'(U)]$ is a vector whose j-th entry is the first partial derivative of $\phi(U)$ with respect to U_j. Using this approximation, the mean square error for ϑ can written as

$$MSE[\vartheta] \equiv E[\phi(u)-\phi(U)]2 \text{Å} E[\phi_U d']2 = \phi_U E[d'd]\phi_U' = \phi_U \ \Phi \ \phi_U',$$

where Φ is a PxP variance-covariance matrix whose jk-th entry is $Cov(u_j, u_k)$. The TLD estimator of MSE[ϑ], usually but incorrectly referred to as a "variance" estimator of ϑ, is thus simply

$$\wedge$$

$$\mathrm{var}[\vartheta] \;=\; \varphi u \;\; \Phi \;\; \varphi u' , \qquad\qquad\qquad (3)$$

^

where $\hat{\Phi}$ is the estimated variance-covariance matrix and φu is the same as φ_U but with entries to u replacing corresponding entries to U.

For stratified cluster samples where the PSUs are chosen with replacement, or in some similar manner thereof, the computational short-cut for producing mse[ϑ] is used. The first step is to compute the point estimate $\vartheta = \varphi(u)$, where the j-th entry entry of u is

$$u_j \;=\; \sum_{h=1}^{H} \; \sum_{a=1}^{a_h} \; \sum_{\ss=1}^{b_{ha}} \; w_{ha\ss} \; u_{jha\ss} \; ,$$

$u_{jha\ss}$ is the j-th measure of relevance for the ß -th respondent of the a-th PSU in the h-th primary stratum, and $w_{ha\ss}$ is the sample weight for the same respondent.

There is a modest difference between theory and practice in the computation of sample weights. In principle, to have E(u)=U requires that there be neither imperfect sample coverage nor survey nonresponse, and that $w_{ha\ss} = 1/^1{}_{ha\ss}$, where $^1{}_{ha\ss}$ is the respondent's selection probability. Reality in fact requires that the $^1{}_{ha\ss}$ be adjusted for sample imbalance due to coverage problems and differential nonresponse. Moreover, for many survey analyses, such as those done by Epi Info, an allowable sample weight for a survey data set is $w_{ha\ss} = G/^1{}_{ha\ss}$, where G is some constant (other than 1) among all respondents. For example, sample weights that have been normalized to the overall sample size (n) have

$$G \;=\; n \;/\; \sum_{ha\ss}^{n} \; 1/^1{}_{ha\ss}, \quad \text{where } \sum_{ha\ss}^{n} \text{ indicates summation over all n respondents}.$$

A constant multiple of $1/^1{}_{ha\ss}$ is acceptable for $w_{ha\ss}$ in these analyses, since G cancels out in computing both ϑ and var(ϑ).

To obtain the computational form of the TLD variance estimator, first note from Equation (2) that

$$\text{MSE}[\varphi(u)] \mathring{A} \ E[\sum_{j=1}^{P} \varphi_j{}'(U)(u_j-U_j)]2 = \text{Var}[\sum_{j=1}^{P} \varphi j{}'(U)u_j]$$

$$= \text{Var}[\sum_{j=1}^{P} \varphi_j{}'(U) \sum_{ha\text{ß}}^{n} w_{ha\text{ß}} \ u_{jha\text{ß}}]$$

$$= \text{Var}[\sum_{ha\text{ß}}^{n} w_{ha\text{ß}} \ Q_{ha\text{ß}}] \ ,$$

where $Q_{ha\text{ß}} = \sum_{j=1}^{P} \varphi_j{}'(U) \ u_{jha\text{ß}}$. Since U is unknown, u is used

in its place to produce the TLD computational variable,

$$q_{ha\text{ß}} = \sum_{j=1}^{P} \varphi j{}'(u) \ u_{jha\text{ß}} \ , \tag{4}$$

which, in turn, is used to produce the computational form of the TLD variance estimator,

$$\text{var}_T(\vartheta) = \sum_{h=1}^{H} [a_h \sum_{a=1}^{a_h} q_{ha} - \{\sum_{a=1}^{a_h} q_{ha}\} 2]/(a_h-1) \ , \tag{5}$$

where $q_{ha} = \sum_{\text{ß}=1}^{b_{ha}} w_{ha\text{ß}} \ q_{ha\text{ß}}$, a_h is the number of sample PSUs in the h-th stratum, and

b_{ha} is the number of respondents in the a-th PSU of the h-th stratum.

The variance estimator in Equation (5) is the so-called ultimate cluster variance estimator as first suggested for simple linear estimators by Hansen, Hurwitz, and Madow (1953). In using this variance estimator one assumes either that PSU sampling was done with replacement or that the PSU sample was chosen without replacement and the PSU sampling rate is sufficiently small so that, in effect, sampling was with replacement.

The principle of replication is the central vehicle by which a variance is obtained in the other three approaches to variance estimation. Not so with the TLD approach, where replication only comes up in conjunction with the ultimate cluster variance estimator of Equation (5). The primary estimation vehicle for the TLD approach is the Taylor approximation for the sampling deviation of ϑ. Therein lies the approach's main utility. The approximation enables the analyst to assess the variance for a complex point estimate as if it were a simpler one. Moreover, the TLD approach requires no significant manipulation of the sample data other than to create a separate measure, the TLD computation variable, for each respondent record. The approach's main drawback is that the specific computational formula for $q_{ha\beta}$ will vary according to the type of point estimates, $\vartheta = \varphi(u)$, since Equation (4) will vary for each different function $\varphi(\ )$. From a programming standpoint, this means that the source code must make provision for computing a separate TLD computational variable for each function of totals for ϑ that the program is designed to accommodate (e.g., a simple ratio vs. a ratio of ratios).

Formulae for Computations in CSAMPLE

TLD variance estimation formulae are needed for point estimates that are simple ratios, differences of ratios, and ratios of ratios. Specifically required to do this is a TLD computational measure ($q_{haß}$) for each functional form. Each $q_{haß}$ is then related back to the eight types of point estimates produced by CSAMPLE.

A few additional definitions are needed. First, the subscripts E and F will be used to denote two nonoverlapping population domains. Then, for Domain E we define the two weighted sample totals,

$$y_E = \sum_{haß}^{n} w_{haß}\, g_{Ehaß}\, y_{haß} = \sum_{haß}^{n} w_{haß}\, y_{Ehaß}$$

and

$$x_E = \sum_{haß}^{n} w_{haß}\, g_{Ehaß}\, x_{haß} = \sum_{haß}^{n} w_{haß}\, x_{Ehaß}$$

where $y_{Ehaß} = g_{Ehaß}\, y_{haß}$, $x_{Ehaß} = g_{Ehaß}\, x_{haß}$, $w_{haß}$ is the sample weight for the $haß$-th among n respondents (as found on the data set), $y_{haß}$ and $x_{haß}$ are, respectively the numerator and denominator measures for the domain-specific ratio, and

$$g_{Ehaß} = 1 \text{ if the } haß\text{ -th respondent is a member of Domain E, and}$$
$$= 0 \text{ if otherwise.}$$

The sample weighted totals, y_F and x_F, the domain indicator, $g_{Fhaß}$, and the unit domain measures, $y_{Fhaß}$ and $x_{Fhaß}$, are similarly defined for Domain F. Using the weighted sample totals, we can define the ratios, $r_E = y_E/x_E$ and $r_F = y_F/x_F$ for Domains E and F, respectively.

TLD Computational Variable for Simple Ratio (r_E)

The simple ratio, r_E, for Domain E is a function of P=2 weighted estimates of totals with $u_{1ha\beta} = y_{Eha\beta}$ and $u_{2ha\beta} = x_{Eha\beta}$ from the general notation of Section 2. Using Equation (4) and taking the necessary first partial derivatives, we have,

$$q_{ha\beta} = r_E[(y_{Eha\beta} - r_E x_{Eha\beta})/y_E] \tag{6}$$

which must be computed for each respondent prior to using Equation (5).

TLD Computational Variable for Difference of Ratios (dEF)

The between-domain difference of simple ratios, $d_{EF} = r_E - r_F$, is a function of P=4 weighted estimates of totals, with $u_{1ha\beta} = y_{Eha\beta}$, $u_{2ha\beta} = x_{Eha\beta}$, $u_{3ha\beta} = y_{Fha\beta}$, and $u_{4ha\beta} = x_{Fha\beta}$. From this setup we have,

$$q_{ha\beta} = r_E[(y_{Eha\beta} - r_E x_{Eha\beta})/y_E] - r_F[(y_{Fa\beta} - r_F x_{Fa\beta})/y_F] \tag{7}$$

TLD Computational Variable for Ratio of Ratios (t_{EF})

The ratio of ratios for exposure domains, tEF=rE/rF, arises in conjunction with the odds ratio and relative risk measures in Epi Info. These measures can be expressed as functions of weighted totals within the following 2x2 table, which resembles the one seen earlier in this chapter in the section on TABLES in ANALYSIS:

```
                Disease
              Yes    No
   Exposure +-----+-----+
       E    |  a  |  b  |   H₁
            +-----+-----+
       F    |  c  |  d  |   H₂
            +-----+-----+
              V1    V2
```

Within this framework the odds ratio is defined as (a x d)/(b x c) and relative risk is (a x H_2)/(c x H_1). With t_{EF} being a function of P=4 weighted totals we once again

have $u_{1ha\beta} = y_{Eha\beta}$, $u_{1ha\beta} = x_{Eha\beta}$, $u_{3ha\beta} = y_{Fha\beta}$, and $u_{4ha\beta} = x_{Fha\beta}$, from which we obtain,

$$q_{ha\beta} = t_{EF}\{[(y_{Eha\beta} - r_E x_{Eha\beta})/y_E] - [(y_{Fa\beta} - r_F x_{Fha\beta})/y_F]\} \quad (8)$$

Final computational details for ϑ and $\text{var}_T(\vartheta)$ are presented in Table 1. The first two columns describe which functional form (and thus which of Equations 6-8 are needed) to compute $q_{ha\beta}$. The other columns describe the domains and unit measures to be used in the calculations.

TABLE 1 Computational Specifications for Eight Types of Point Estimates Computed by CSAMPLE

| | | | Nonoverlapping Domains | | | | | |
| | | | Domain E* | | | Domain F** | | |
Type	Point Estimate (J)	Form	Type	yhaß =	xhaß =	Type	yhaß =	xhaß =
(1)	Domain mean	rE	PS	CV	1	-	-	-
(2)	Domain Proportion	rE	PS	AV	1	-	-	-
(3)	Difference of domain means	dEF	PS	CV	1	PS	CV	1
(4)	Difference of domain proportions	dEF	PS	AV	1	PS	AV	1
(5)	Ratio of domain means	tEF	PS	CV	1	PS	CV	1
(6)	Ratio of domain proportions	tEF	PS	AV	1	PS	AV	1
(7)	Odds ratio for exposure domains	tEF	EG	AV	1-yhaß EG	AV	1-yhaß	
(8)	Relative risk for exposure domains	tEF	EG	AV	1	EG	AV	1

Legend:

rE = Simple Ratio   PS = Population Subgroup
EG = Exposure Group   dEF = Difference of Ratios
CV = Continuous Variable   AV = Attribute Variable***
tEF = Ratio of Ratios

*   For Domain E: yEhaß=gEhaßyhaß and xEhaß=gEhaßxhaß

**   For Domain F: yFhaß=gFhaßyhaß and xFhaß=gFhaßxhaß

*** For an Attribute Variable (AV): yhaß = 1 if the haß-th respondent possesses some attribute
(e.g., response to some category of a categorical response variable, suffer from some disease, etc.),
and = 0 if otherwise.

References

Hansen MH, Hurwitz WN, and Madow WG (1953). <u>Sample Survey Methods and Theory</u>, Vol. I, Wiley and Sons, New York.

Wolter KM (1985). Introduction to Variance Estimation, Springer-Verlag, New York.

Woodruff RS (1971). "A simple method for approximating the variance of a complex estimate," <u>Journal of the American Statistical Association,</u> 66:411-414.

Blank to maintain chapter pagination of Version 6.00 Manual.

Blank to maintain chapter pagination of Version 6.00 Manual.

Blank to maintain chapter pagination of Version 6.00 Manual.

Blank to maintain chapter pagination of Version 6.00 Manual.

(Blank to maintain chapter pagination of Version 6.00 Manual.)

```
F1-Hlp F2-File F3-Epiaid F4-Txt F5-Print F6-Set F7-Find F8-Blk F9-Save F10-Done

                            Setup
                     WW/TXT/QES mode      Txt
                     Insert mode          ON
                     Indent as above      ON
                     Pagination           OFF
                     Justify R&L          OFF
                     Auto word wrap       ON
                     Left margin          1
                     Right margin         65
                     Top margin           3
                     Bottom margin        5
                     Page length          60
                     Reload last file     OFF
                     Tabs..
                     Load file options..
                     Display options..
                     Colors..
                     Restore defaults this mode
                        Enter/Space/Esc/HKey


 1  UNTITLED                    EPED  311088     L1   C1  TXTInsInd
```

Chapter **33**

EPED• Commands

Running EPED from the DOS Command Line

When running EPED from DOS or from the EPI menu using the space bar to choose the program, the name of a file to be edited may be given as follows:

EPED <File name>

If the file name is preceded by a slash, the file is understood to be a program file for the EPIAID feature of EPED and must end in the extension, .EPA, for example:

EPED /CREATE

The CREATE.EPA program will be run as soon as EPED appears.

In either case, the initial title screen of EPED will be skipped.

Setting EPED for Your Printer

Printer Definition Files (.PDF) are provided for Epson, IBM Proprinter, Okidata, Hewlett Packard Deskjet and Laserjet, and Postscript printers. To choose the correct file, press <F5> to bring down the print menu, and then select press WHICH PRINTER with the <Down arrow> and <Enter> to see the selection of .PDF files. Choose the one that most closely matches your printer. Postscript printing requires that the file POSTSCPT.INI supplied with Epi Info, be present in a directory in the current path.

If necessary, the special character sequences in the .PDF file can be edited using the EDIT PRINTER CODES command in the <F5> Print menu. To set up a particular sequence of characters that EPED will send to the printer to turn bold text on, for example, choose BOLD TEXT ON from this menu. A window will appear. To enter a special sequence of keys in the file, press the <Scroll Lock> key once. This turns on the literal interpretation of keys, so that <Esc> is entered into the file rather than causing the menu to disappear. Enter the special sequence of keys and then press <Scroll Lock> again. To remove characters, first turn the literal key entry

mode off with <Scroll Lock> and then use <Backspace> or <Delete> to remove characters.

When the sequence of characters is correct according to the manual for your printer, press <Esc> to remove the menu; choose "Save Printer Setup" from the Print menu, and then save the results in a .PDF file. Once the printer setup has been saved, you will not have to repeat the process unless you wish to alter the setup.

Menu Commands in EPED

Most of the commands in EPED are accessible through the 10 function keys or the 10 <Ctrl> function keys. Each key either executes a command, like HELP, or pops up a menu on the screen from which other commands can be chosen. The commands are described below as they appear in the EPED menus.

<Ctrl> Function Keys

Several frequently used functions are accessible using the <Ctrl> key with a function key. To see these functions, hold down the <Ctrl> key and examine the prompt line at the top of the screen. The functions are:

  <Ctrl F1> Begin Block (Repeat twice to begin a Column Block)
  <Ctrl F2> End Block
  <Ctrl F3> Copy Block
  <Ctrl F4> Move Block
  <Ctrl F5> Delete Block
  <Ctrl F6> Find Text
  <Ctrl F7> Replace Text
  <Ctrl F8> Format (justify) a paragraph
  <Ctrl F9> Mode--Change Word wrap mode to Text (TXT) mode or vice versa

Shortcut Keyboard Commands

Many EPED commands in the menu have keyboard equivalents, often using the control key **<Ctrl>**, held down while pressing one or two other keys in succession. These will be particularly useful if you have already learned to use a WordStar-like editor and want to use the same commands. They are faster than using the menu commands, once you have memorized the control sequences. Several commands

not accessible through the menus are also available through control sequences.

The control-key commands for each menu item are shown on the top line of the screen as you move the cursor bar to each selection on a menu. Commands listed below are followed by their keyboard equivalents, if any.

<F1> The HLP or HELP Command

When <F1> is pressed and other menus are absent, a menu of help topics appears. Move the cursor bar to the topic desired and press <Enter> to see the information. When a menu is already present, pressing <F1> will bring up information about the topic in the menu where the cursor bar rests.

<F2> The FILE Commands

OPEN FILE THIS WINDOW <Ctrl K O> (Control K "Oh", not "zero")
Opens a new or existing file and brings the text into the work space. If there is already a file in the work space, you will be given a chance to save or discard it. You will then be asked for the new file name. If the file specified exists, it will be brought into the work space. If not, the name of the file will appear at the bottom of the screen, and a blank screen will be presented.

If you do not know the name of the file, use a "wildcard" character, either "*" or "?", to bring up a list of files. "*.*" will list all files in the directory. "**\Data*.TXT**" will list all files in the **Data** directory with the suffix **.TXT**. "?" stands for "any one character" and "*" means "any sequence of characters." When the list appears on the screen, a file may be opened by moving the cursor bar to the choice and pressing <Enter>.

Once a file is open, new information typed on the screen may be saved with the <F9>-Save command. You will also be reminded to save it if you press <F10> Done to quit or attempt to open another window.

ADD FILE OTHER WINDOW   <Ctrl O A>

Opens another file in a different "window." Windows in this case consist of full screens, with the window number indicated at the lower left corner of the screen. EPED can manage up to four open windows at a time. If you attempt to open more than four windows, an error message is displayed.

CHANGE WINDOW   <Ctrl J W>

If more than one window is open, a list of the open windows is displayed in a pop-up menu. Selection of another window can be made by using the cursor bar. This provides a method for switching rapidly from file to file or from one part of the same file to another.

CLOSE FILE THIS WINDOW   <Ctrl O Q>

If changes have been made in the current file, the option to save the file will be given. If you do not wish to save it, type "N", and the changes will be discarded. This command is also useful for closing one of several windows without closing all, as the <F10> Done command does.

SAVE FILE TO..   <Ctrl K N>

This saves the file currently in the window to a specified file name. If you want to open an existing file, modify it, and make another file with a different name, this will accomplish the task without disturbing the original file. Note that the file name at the bottom of the screen changes to the new file name and that next time you Save, the material will go to the **new** file.

FILE DIRECTORY

Displays a listing of all files available in the current directory. You can search for a file by using a wildcard (*). For example, "*.*" will list all file names; "*.TXT" will list all files with an extension of .TXT; or "EPED.*" will list all "EPED" files with any file extension. This command displays file names; you cannot retrieve a file with this command.

USE DOS   <Ctrl J O>

This allows you to invoke DOS commands or to access other programs from EPED. At the prompt box, enter the DOS command and press <Enter>. After the DOS command has been executed, you will be prompted to press a key to return to EPED.

Pressing <Enter> at the prompt box without entering a command will invoke a DOS shell, from which any sequence of DOS commands can be executed. Return to EPED by typing EXIT.

SET DEFAULT DIRECTORY <Ctrl J D>
At the prompt box, enter a new drive or path name to change the default directory.

INFO ON FILE AND SYSTEM <Ctrl J V>
Displays information about the current file (name, # of lines, size of file, # of words, # of pages) and about the disk (RAM, current path, disk space, version of DOS).

HELP <F1>
Help is available either through the <F2>-File menu or by pressing <F1>. A list of topics is displayed. Choose a topic with the cursor bar and press <Enter> to bring the information to the screen. You can also press <F1> from within any of the pop-up menus. EPED will display information regarding the command that is currently highlighted on the menu.

<F3> The EPIAID Command

This command brings up the EPIAID menu. The use of EPIAID is described in Chapters 20, 21, and 28 and in the EPIAID reference section later in this chapter.

<F4> The TXT or TEXT Commands

QUESTIONS <Ctrl Q Q>
This displays a pop-up menu of data field types for creating an Epi Info questionnaire. Fields can be entered into the questionnaire by placing the cursor in the data input location in the questionnaire and then selecting the data field type from the menu using the selection bar.

FORMAT PARAGRAPH <Ctrl F9> or <Ctrl B>

Functions only when word wrap is ON. This changes the length of lines in a paragraph if items have been deleted or added, or if right and left margins have been changed.

TEMPORARY MARGIN   <Ctrl O G>
Moves the left margin one tab stop to the right. Remains in effect until you leave the current paragraph. Works only if word wrap and fixed tabs are ON.

RELEASE MARGIN   <Ctrl O X>
Allows you to type beyond the left or right margin when word wrap is ON. Remains in effect until cursor is moved to a new line.

STYLE OF TYPE
Font styles are selected by holding down the <Alt> key and pressing a number (1-7).

<Alt-1> **Bold**
<Alt-2> **Double**
<Alt-3> <u>Underscore</u>
<Alt-4> Superscript
<Alt-5> Subscript
<Alt-6> Compressed
<Alt-7> *Italic*

To use a particular style, press the corresponding Alt key combination. The color of the text on the screen will change to indicate the new style. Type the text to be printed in the new style. Press the <Right arrow> key to return to the normal text style. (This jumps over the control character that is inserted to the right of the special-style text.)

To print text in the styles selected, a printer configuration file (.PDF) file must be loaded to handle the conversion of style indicators into directions for particular printers. Further details on .PDF files are given in the previous section on "Printing."

MACRO USE
A macro is a "recording" of a number of keystrokes that can be played back under your direction to repeat a complex operation . A macro can be played back once by choosing the <F4> TEXT menu and then MACRO USE and PLAYBACK. The real power of macros comes, however, from using them in combination with the

FIND/REPLACE function. To do this, choose the <F7> FIND menu, then USE
MACRO WITH FIND/REPL. Depending on what you enter for FIND, you can
have the macro operate each time a particular character, word, or phrase is
encountered in the file, or, by FINDing "^L", at the end of each line in the file.

To record a macro, choose the <F4> TXT menu and then MACRO USE. A menu
appears, offering several choices, as follows:

LOAD MACROS--allows you to load a file containing up to 10 macros already
recorded. A file called EPED.MAC is loaded automatically, if one is present, but
this option allows you to load other files of macros that you may create. This file is
supplied with Epi Info, but may be altered or replaced with macros of your own
choosing.

STORE MACROS--saves the 10 or fewer current macros into a .MAC file. You
will be asked for the name of the file in which to store the macros.

PLAY BACK MACRO--Plays back a macro from the current cursor position. You
will be asked to choose the macro from a list of 10 in the current macro file.

RECORD--ON OR OFF--<Ctrl J T> The first time this choice is made, EPED
starts recording macro keystrokes. Up to 255 keystrokes can be recorded. When
you have completed the operation(s), make this choice again to turn off recording,
and you will be asked to choose one of the ten "slots" in the current macro file and
give a name to the macro. To make the recording permanent, you should also
choose STORE MACROS before leaving EPED to save the recording in a file. The
keystrokes <Ctrl-J T> are a shortcut method of choosing RECORD-ON or OFF.
Whichever the method of recording, only the material between the two choices will
be recorded--not the menu choices that lead to the recording.

EDIT--Allows editing or creation of a macro. You are asked to choose a macro.
An editing window appears. If there are already keystrokes in the macro, these are
represented in symbolic form--"<Enter>" for the <Enter> key, for example. The
window has two modes, controlled by the <Scroll lock> key. In Literal Mode, the
keystrokes are entered directly in the window as in an editor. In Macro Command
Mode, pressing <Enter> makes the symbol "<Enter>" appear on the screen, and the
<Backspace> and <Delete> keys are represented by "<Ctrl-H>" and "<Del>." In
literal mode, <Backspace> and <Delete> may be used to remove symbols so that
the correct commands can be included in the macro in the desired sequence. The
Wordstar commands listed in Chapter 33, the EPED Reference Chapter, are
particularly useful for macros, as they are easier to visualize that the keystrokes
necessary to go through the menuing system.

A sample macro file called EPED.MAC comes with EPED. It is loaded

automatically when EPED is run. To see how macros work, press <F4> and then choose MACRO USE from the TXT menu, and then PLAYBACK. Choose a macro from the 10 choices presented and you should see the results on the screen.

Two shortcut methods of playing back macros are offered. The first allows you to play back one of macros 1 through 9 in the current macro file, by pressing <Ctrl O> (that's "oh," not "zero") and then the number of the macro (since <Ctrl 1> is not recognized by the keyboard, you must release the <Ctrl> key before pressing the number.)

The second series of shortcuts allows you to play back macro number zero from 1 to 9 times by pressing <Ctrl J> and then a digit 1 through 9 (after releasing the <Ctrl> key).

Macros can contain up to 255 keystrokes. A macro file can hold up to 10 macros. You can have as many macro files as your disk will hold. Only one file of 10 macros can be kept in memory at a time.

CENTER CURRENT LINE <Ctrl O C>
>   Centers current line between the left and right margins. Works
>   whether word wrap is ON or OFF.

PLACE MARKER
>   Sets one of the 10 (0-9) text markers at the current position of
>   the cursor. You can jump to the marker directly by using the
>   "Marker #, Move to" command in the <F7> menu.

UNDO THIS LINE <Ctrl Q L>
>   Undoes any changes made to a line of text as long as you have
>   not left the line. The line is restored to the state it was in before
>   changes were made.

UNDELETE <Ctrl Q U>
>   Restores whole lines deleted with the DELETE LINE (<Ctrl-
>   Y>) command or the DELETE BLOCK (<Ctrl-K Y>)
>   command. The number of lines of a block that will be restored
>   is determined by the "Set undo limit"; this specifies the
>   number of deleted lines that will be held in the undo buffer.
>   This can be set in the <F6>-Load file options menu. (The
>   default setting is 20 lines.) This command will not restore
>   single characters, words, or phrases removed from the current
>   line with the delete or backspace keys; use UNDO THIS LINE
>   for that purpose.

<F5> The PRINT Command

PRINT FILE NOW
> Starts a print job after print options have been selected.

LEFT MARGIN
> Sets the left margin of the page for printing. Normally a margin of 5 works well unless your lines are unusually long.

NAME OF FILE
> Specifies the file to be printed. The current file is the default file name. To select another file to print, enter the file name in the prompt box. You may use wildcards to request a list of files, and then choose the file using the selection bar.

DOUBLESPACE
> If this is ON, the printed output will be double spaced; otherwise, it will be single spaced.

FIRST PAGE
> Sets the number of the first page to be printed, allowing you to skip pages at the beginning of a file. To set the number that appears on the printed page, use the command @PN, described in the section "Print Formatting Commands" below.

LAST PAGE
> Sets the number of the last page to be printed, allowing you to skip pages at the end of a file.

WHICH PRINTER
> Specifies the name of a printer definition file.

DEVICE
> Indicates where printer output should go: FILE, LPT1, or LPT2. If your printer is attached to a serial port, use the DOS MODE command to redirect output (e.g., MODE LPT2 = COM1). You will also need a MODE command to set the speed, parity, and other characteristics of the serial port as described in the manual for your printer. If your printer functions with other programs, it is likely that no further arrangements are necessary to set it up for Epi Info, however.

MANUAL PAPER FEED
> ON/OFF toggle. When ON, you are prompted to insert a new
> sheet of paper at the end of each page. When OFF, the printer
> prints continuously.

USE FORM FEEDS
> ON/OFF toggle. When ON, sends a form-feed command to the
> printer when it reaches the end of a page. When OFF, blank
> lines are used to fill out the page.

EDIT PRINTER CODES
> Allows you to edit the string of characters sent to the printer
> when starting a new job, changing fonts, and so on. To use the
> code editor, choose a command sequence such as Bold Text
> ON. When the editing window appears, press <Scroll Lock>
> once and enter the keys corresponding to this command from
> your printer manual. Then press <Scroll Lock> again and
> <Esc>. Use Save Printer Setup to save the results in a .PDF
> file.

SAVE PRINTER SETUP
> Saves to a specified file the printer codes you entered using the
> previous commands.

<F6> The SET Commands

WW/TXT/QES Mode <Ctrl F9> or <Ctrl O W>
> Pressing the <Spacebar> cycles through the Word Wrap, Text,
> and Questionnaire modes of EPED. An indicator of the mode
> appears at the bottom of the screen as WW, TXT, or QES. It is
> also possible to toggle between WW and TXT mode by pressing
> <Ctrl-F9>. In WW mode, word wrap is ON, and usually the
> margins are set to less than the full 80 character width of the
> screen. In TXT and QES modes the right margin is set to 80.
> TXT mode comes with Pagination turned off so that the text is a
> continuous stream, suitable for programs. Questionnaire mode
> initially has the page length set to 23 lines so that each page
> corresponds to one screen in the ENTER program.

` Any of the settings on the <F6> menu can be changed. The
> changes apply to the current WW/TXT/QES mode. If you wish
> to set TXT mode so that it has different margins, page sizes,
> etc., merely choose the settings while in TXT mode. As soon as
> you leave the <F6> SET menu, the setting will be saved for the

current mode. To save settings for another mode, repeat the process and be sure to exit from the SET menu after setting the mode.

INSERT MODE   <Ctrl V>

When Insert mode is ON, text to the right of the cursor is moved to the right as new text is entered. When OFF, Overwrite mode is ON; any text above the cursor is overwritten when text is entered. The INS indicator appears at the bottom of the screen, when Insert mode is ON.

INDENT AS ABOVE: <Ctrl Q I>

When ON, the New Line command (Enter or Ctrl-M) will move the cursor to the next row and line it up with the first non-blank character on the previous line. The IND indicator appears.

PAGINATION   <Ctrl O B>

When ON, displays markers on both sides of the screen in the first line of each page. The current page number is displayed on the Status Line. Printed documents will include page breaks at the locations marked.

JUSTIFY R & L   <Ctrl O J>

When ON, each line is filled so that the last character touches the right margin and word wrap is turned ON. The JST indicator appears in the lower right corner of the screen.

AUTO WORD WRAP

When ON, paragraphs are reformatted automatically when new text is added or old text deleted. The reformatting occurs when a <Space> is inserted or <Delete> or <Backspace> is pressed. When OFF, paragraphs may still be reformatted by pressing <Ctrl F9> or <Ctrl B> with the cursor anywhere in the paragraph.

LEFT MARGIN: <Ctrl O L>

Sets left margin for text displayed on the screen. (Default setting is 1.)

RIGHT MARGIN: <Ctrl O R>

Sets right margin for text displayed on the screen. (Default setting is 65 for WW mode and 80 for TXT and QES mode.)

TOP MARGIN:

>   Sets the default top margin for printed documents. (Default
>   setting is 3 lines.)

BOTTOM MARGIN:

>   Sets the default bottom margin for printed documents. (Default
>   setting is 5 lines.)

PAGE LENGTH

>   Sets the default number of lines per page for printed documents.
>   (Usually 66 for EPSON/IBM dot matrix printers and 60 for
>   laser printers.)

RELOAD LAST FILE

>   When ON, EPED will "remember" the name of the file that is
>   current when you quit EPED (with <F10>). The next time
>   EPED is run, that file is loaded automatically. This feature is
>   especially useful for projects in which the same file is loaded
>   many times for repeated editing, as in drafting an article or
>   debugging a program.

TABS

>   EDIT OR SEE TABS: Allows you to set tabs interactively. Use
>   the cursor keys to move along the displayed tab line.
>   Pressing the space bar toggles the setting ON/OFF. Pressing the
>   <Ins> key will add a tab and <Del> will delete a tab at the
>   cursor position. Press <Enter> when editing is complete, or
>   <Esc> to undo any changes.
>
>   SET TABS FROM CURRENT LINE: The tab settings will be
>   set according to the contents of the current line of text. The
>   start of each word will set a tab at that column. All other tabs
>   will be cleared. You can return to the default tab settings by
>   using the RESTORE EVEN TABS command.
>
>   RESTORE EVEN TABS: Tabs will be set with an even
>   spacing, determined by the current default tab spacing. Any
>   other tab setting will be cleared.
>
>   PUT TABS ON NONPRINTING LINE: The tab settings are
>   stored into the current window as a formatted text line. Using
>   this command together with the "Set tabs from current line"
>   allows you to store a custom tab line as part of a document.

DISPLAY 22 LINES + PROMPTS: Display ON reserves one line of the current window for display of the menu prompts. If this setting is OFF, the screen accommodates a full 23 lines, the same as the screens in ENTER. In preparing questionnaires, DISPLAY 22 LINES + PROMPTS is useful to visualize the way paging will work.

TAB SPACING - # CHARACTERS: Changes the tab size from the default setting of eight characters. After changing the tab size, use the "Restore even tabs" command to initialize a different set of regularly spaced tabs.

LOAD FILE OPTIONS

HOME DIRECTORY: Tells EPED where its optional support files are located. These include printer definition, default macro, and help files.

FILE EXTENSION: Specifies the default extension for file names.

EXPAND TABS: Expands tab characters to spaces when reading in files created with other editors. The assumed size of the tabs is specified with the Set Tab Size command. Spaces may be converted back to tabs when output is written.

WRITE TABS: When ON, converts strings of blanks to tabs when a file is written to disk. (This saves disk space.) EPED uses the current default tab spacing when it converts the blanks into tabs.

STRIP HI-BIT: When ON, strips "high-order bits" when reading in files created with certain editors which use the 8th (highest) bit in a byte to store information. To see whether this option is needed, use the DOS TYPE command; if the last character in each word or line is garbled, then high-order bits are present.

UNDO LIMIT: Sets the size of the "undo buffer" that stores deleted lines. The default setting is 20 lines.

DISPLAY OPTIONS

SNOW CONTROL: When ON, the screen interference (snow) on color graphics adapters is avoided. Ignore the command if using monochrome adapter. If you don't experience the

problem, turn it OFF; this will improve the speed of EPED's performance.

BLOCK CURSOR: When ON, hides blinking cursor you normally see on the screen and replaces it with the block cursor. (Blinking cursor is the default.)

43/50 LINE: Activates special 43-line display mode available on systems equipped with EGA or VGA graphics adapters. Ignore this command if EGA or VGA is not installed. The normal mode is 25 lines.

FONT DISPLAY: Turns ON and OFF the display of the font styles as shaded or colored text. If the FONT DISPLAY is ON, text in styles other than normal will be highlighted and surrounded by shaded block characters marking the position of the control characters that turn on and turn off the special style. If FONT DISPLAY is OFF, the special style is not highlighted, but the control characters surrounding it appear in shaded form. <Ctrl B>, for example, appears as a shaded "B".

KEY HELP: By default, symbols for the keyboard commands associated with particular menu choices are displayed above the prompt line as you scroll through the lists of choices.

ZOOM STATE: If set, new windows are automatically zoomed to fill the screen when they are opened. This command is normally ON. (This setting does not affect the "Zoom current window" command. Other window commands are described later in this chapter.)

COLORS

Changes the colors used for text windows, status lines, pull-down menus, etc. Select the setting you wish to change, using the selection bar, and press <Enter>. The choices available to you are displayed, and the current setting is marked with a snowflake character. Change the setting by moving the snowflake with the arrow keys; as it moves, the item will reflect the change in colors. When you find the color you want, press <Enter> (<Esc> to restore the original setting.)

RESTORE DEFAULTS THIS MODE

Restores the default settings for the current mode (WW, TXT, or QES).

SAVE SETTINGS THIS MODE
> Saves the current configuration settings specified by commands
> in the Setup menu by altering the EPED.EXE file. It is
> important that you specify the drive and directory where EPED
> support files are located. If EPED.EXE is not found in either
> the current or the home directory, an error message is
> displayed.

<F7> The FIND Commands

FIND <Ctrl-F6> or <Ctrl Q F>
> Searches for a string of up to 67 characters. A prompt box
> displays the last search string entered. You may select it with
> <Enter>, edit it, or enter a new search string. After the search
> string is entered, specify the search options

C=Case Specific	Match upper/lower case of your entry and ignore those that fail to match. Normal default is to match regardless of case. Note: This is a change from Version 5 of EPED.
U=Upward	Look upwards in the file from the current cursor position.
W=Word	Find the string only if surrounded by spaces.
B=Block	Find "color" only within a block of text that has been selected with the block commands. This allows for confining choices to a vertical or horizontal column.

FIND AND REPLACE <Ctrl-F7> or <Ctrl Q A>
> Searches for a string, and replaces it with another string. A
> prompt box displays the last search and replace strings entered.
> You may select them by pressing <Enter>, edit them, or enter
> new search and replace strings. The search and replace options
> are the same as those above, with three additional options:

G=Global	Searches the entire file (globally). The search begins at the top of the file if doing a forward search; it begins at the end of the file if doing an upward search.
N=No ask	Replaces without asking (no ask); it does not prompt for confirmation at each occurrence of the search/replace string.
1..n=How many	Change the specified number of instances before stopping.

REPEAT LAST FIND/REPLACE <Ctrl L>
The previous find or replace operation is repeated.

USE MACRO
Searches for a string of up to 67 characters and then executes the specified macro. Prompts for the search string and then for the macro to invoke.

PAGE NUMBER, MOVE TO <Ctrl J P>
The cursor will move to the first line of the page specified in the prompt box. If the number specified is preceded by a plus (+) or minus (-), the target page number will be calculated relative to the current page.

LINE NUMBER, MOVE TO <Ctrl J L>
The cursor will move to the specified line number of the current window. If the number specified is preceded by a plus (+) or a minus (-), the target line number will be calculated relative to the current line.

COLUMN NUMBER, MOVE TO <Ctrl J C>
The cursor will move to the specified column number of the current window. If the number specified is preceded by a plus (+) or a minus (-), the target column number is calculated relative to the current column number.

TOP OF FILE, MOVE TO <Ctrl PgUp>
Moves the cursor to the beginning of the file.

BOTTOM OF FILE, MOVE TO <Ctrl PgDn>
Moves the cursor to the end of the file.

MARKER #, MOVE TO
Moves the cursor to the position of a previously stored text marker, set with the <F4>-"Place marker" command. A list of previously set markers will be displayed from which to select the desired marker.

START OF BLOCK, MOVE TO <Ctrl Q B>
Moves the cursor to the beginning of a block. This command functions even if the block is hidden. (See HIDE BLOCK below.)

END OF BLOCK, MOVE TO <Ctrl Q K>
>   Moves the cursor to the end of a block. This command functions
>   even if the block is hidden.

<F8> The BLK or BLOCK Commands

BEGIN BLOCK <Ctrl F1> or <Ctrl K B>
>   Marks the beginning of a block. For Column Block, give the command twice
>   without moving the cursor.

END BLOCK <Ctrl F2> or <Ctrl K K>
>   Marks the end of a block. The block is highlighted only if both
>   the beginning and end are marked.

HIDE BLOCK DISPLAY <Ctrl K H>
>   Toggles ON/OFF the visual marking of a block. (Most of the
>   block-manipulation commands work only when the block is
>   displayed.) Another way to make a deselect a block is by giving
>   the BEGIN BLOCK and END BLOCK commands at the same
>   cursor location.

COPY BLOCK TO HERE <Ctrl F3> or <Ctrl K C>
>   Creates a copy of a marked and displayed block at the current
>   cursor position. The original block is left unchanged.

MOVE BLOCK TO HERE <Ctrl F4> or <Ctrl K V>
>   Moves a marked and displayed block from its current position
>   to the cursor's position. Markers remain around the block at its
>   new position.

DELETE BLOCK <Ctrl F5> or <Ctrl K Y>
>   Deletes a marked and displayed block. There is no command to
>   restore a deleted block in its entirety. The "Undelete" command
>   can restore only the number of lines specified in the undo buffer
>   limit. (The default limit is 20 lines.)

WRITE BLOCK TO FILE <Ctrl K W>
>   Writes currently marked block to the specified file. Prompts
>   first for a file name. If the file already exists, you are asked if
>   you want to overwrite it or append to it. If the file does not
>   exist, a new file is created.

READ FILE IN AS BLOCK <Ctrl K R>
>   Reads an entire file into the current file at the cursor position.
>   The file read in is marked as a block.

FORMAT CURRENT BLOCK <Ctrl-F8> or <Ctrl K F>
>   Reformats all text within the currently marked block. Word
>   wrap must be ON.

<F9> The SAVE Command

This command saves the file. If no file has been designated, it first prompts for a
file name.

<F10> The DONE Command

This command leaves EPED and returns to the EPI menu or to DOS. If new
material has been added to the screen since the last SAVE, you are reminded and
given a chance to save it before leaving EPED.

Print Formatting Commands

Commands to control the format of printed copy can be embedded in text. They are
two-letter commands preceded by the "@" sign. Although they appear on the
screen, they will not be printed. The commands available are:

@PO n Indents remainder of document by n columns.

@PA   Starts a new page.

@CP n Starts a new page if fewer than n lines remain on the page.

@PN n Sets page number to n.

@OP   Omits page numbers. They are omitted automatically if the document
       contains only one page, and EPED is in Word Wrap mode.

@PG   Turns page numbers on.

@HE <Text> Prints the material in <Text> at the top of each page.

@FO <Text> Prints the material in <Text> at the bottom of each page.

Miscellaneous Keyboard Commands

Additional keyboard commands that do not have equivalent entries on the menu are:

Insert control character	Ctrl-P
Toggle case	Ctrl-O O
Lower case	Ctrl-O V
Upper case	Ctrl-O U
Mark single word	Ctrl-K T
Previous cursor position	Ctrl-Q P
Up to equal indent	Ctrl-J B
Down to equal indent	Ctrl-J E
Next sentence	Ctrl-Q Z
Previous sentence	Ctrl-Q W
Character left	Left arrow or Ctrl-S
Character right	Right arrow or Ctrl-D
Word left	Ctrl-Left arrow or Ctrl-A
Word right	Ctrl-Right arrow or Ctrl-F
Line up	Up arrow or Ctrl-E
Line down	Down arrow or Ctrl-X
Scroll up	Ctrl-W
Scroll down	Ctrl-Z
Page up	PgUp or Ctrl-R
Page down	PgDn or Ctrl-C
Beginning of line	Home or Ctrl-Q S
End of line	End or Ctrl-Q D
Tab	Tab or Ctrl-I
Backward Tab	Shift-Tab
Top of screen	Ctrl-Home or Ctrl-Q E
Bottom of screen	Ctrl-End or Ctrl-Q X
New line	Enter or Ctrl-M
Insert line	Ctrl-N
Delete current character	Del or Ctrl-G
Delete character left	Backspace or Ctrl-H
Delete word	Ctrl-T
Delete to end of line	Ctrl-Q Y
Delete line	Ctrl-Y
Deactivate menu	Esc

Most of the commands involving movement of the cursor on the screen are more easily accomplished with the **arrow keys, <Ins>, <Del>, <Home>, <End>, <Backspace>**, **<Enter>, <PgUp>**, and **<PgDn>**.

Other Editing Functions

The **UNDO** command saves the number of lines specified in the **UNDO LIMIT** after a delete. Using the Undelete command from the <F4> menu will bring these lines back to the work space one by one, performing a rescue operation in case you did not really mean to delete them.

Setting margins, word wrap, and formatting are similar to these functions in WordStar. After a margin has been changed or text has been inserted into a line in insert mode, place the cursor at the beginning of the insertion or of the paragraph, and use the <Ctrl-B> command to reformat the text when word wrap is ON.

Do not place carriage returns at the end of lines that are to be formatted as part of a paragraph, as EPED considers a carriage return to be the end of a paragraph.

Related Topics in Other Chapters

Many of the features of EPED are described in Chapters 6 and 7. The EPIAID system in EPED is described in Chapters 21, 28, and 34. Chapter 38 contains information on running EPED in a local area network (LAN).

EPIAID Commands

_CENTER

_CHOOSE
    BOX
     CHOICE
     TITLE
     COLOR
_ENDCHOOSE

_CLEARALL
_CLOSEFILE
_DEFINE
_DISPLAY
_ENDDISPLAY
EXISTFILE
_IF
_JUMP
_LEFT
_LET
_MARK
_OPENFILE
_QUESTIONS
_QUIT
_READFILE
_REMOVE
_REPEAT
_UNTIL
_RLJUSTIFY
_RUN
_SINGLELINES
_WRITE

Chapter 34

EPIAID Commands

Running EPIAID Programs

When the EPIAID function in EPED is chosen by pressing <F3>, EPED looks for a program called EPIAID.EPA. If it finds this program--a menu program supplied with the system and possibly modified for your own use--it will run this program. If not, a list of available .EPA programs will appear and the user is asked to choose one.

EPIAID programs, with names ending in .EPA, may be run from the DOS command line by giving the command:

EPED /<File name>

in which <File name> is the name of the program, with the extension .EPA. What happens after the program has run to completion is determined from within the .EPA program. If there is a QUIT command, the program will exit to DOS. If not, it will exit to EPED.

EPIAID Program Lines or Statements

✔A line of **text**, beginning with any character except **"_", "[", "%", or ":".** The text will be put on the screen as part of the current document, using whatever form of justification (word wrap, etc.) is currently in effect.

✔A request for **user input,** beginning with **"["** and ending with **"]"**. The text in between will be put on the screen in a user input window, and then the user will be allowed to type text and use EPED commands until he or she signals the end of input by pressing <Esc>. The brackets and enclosed text must fit in one 80-column line.

✔An EPIAID **command**, beginning with an underline character, **"_"**. The commands allow formatting text; conditional jumps; defining, assigning, and

retrieving variables; access to DOS; construction of menus; display of blocks of text, mathematical and logical operations; opening and closing files, importation of text, and many other actions. EPIAID commands are listed later in this chapter.

✔A **comment** that explains the program, beginning with **"%"**.

✔A **label**, beginning with **":"**, marking a location where a JUMP command will direct the program to go next.

EPIAID Files and File Names

EPIAID program files provide instructions to EPIAID to drive interactive text processing. Program files end in the file extension .EPA (for EPIAID). They are ordinary text (ASCII) files that can be produced on most word processors.

Text or output files are those produced in the editor by your interaction with EPIAID programs. They are in the same form as any file produced by the word-processing part of the EPIAID editor and can be further edited in EPED if desired.

What Happens If You Stop in the Middle of a Program?

In order to resume processing at the point where an interruption occurs, several items must be preserved. These include the partially completed text file, the contents of temporary variables (created by DEFINE commands or simple assignments), and the place in the program file where the processing stopped. When a program is interrupted with the <F10> key, the user may choose to save the file, in which case another file called <Program name>.INT, such as DESIGN.INT, is made automatically. This file contains the location of the last line processed in the program, the contents of defined variables, and the name of the file that was saved at the time of the interruption.

The next time the program is run, EPIAID checks for the presence of the .INT file. If one is present, you will be asked to decide whether to resume processing where the interruption occurred or to begin again from the beginning. The latter choice will bring back the partially processed text file and allow resumption of the processing where it stopped.

EPIAID Commands

The following commands are available for use in EPIAID programs. Each must be preceded by the underline character, "_".

_CENTER

Syntax   CENTER ON (If ON and OFF are omitted, ON is assumed)
                       OFF

When CENTER is ON, each line of text is centered between the current margins. RLJUSTIFY and WORDWRAP are turned OFF and SINGLELINES is turned on.

Examples: **_CENTER**
           _CENTER ON
           _CENTER OFF

_CHOOSE
_ENDCHOOSE

Syntax : _CHOOSE X1 Y1 X2 Y2
        in which X1,Y1 are the horizontal and vertical coordinates of the upper left
        corner of a screen area to be reserved, and X2 and Y2 are the coordinates
         of the lower right corner.

The effect of CHOOSE is that the designated area of the screen will be saved to memory. ENDCHOOSE will restore the original screen and end the effect of CHOOSE. Every CHOOSE must be followed by a single ENDCHOOSE before the end of the file is reached.

CHOOSE is used to set up the screen for the following commands that can occur only when CHOOSE is active:

        BOX
        CHOICE
        TITLE
        COLOR

        BOX

Syntax : BOX X1 Y1 X2 Y2

This draws a box within the CHOOSE area and clears the center of the box. The upper left and lower right corners of the box are given by the coordinates as for CHOOSE. The color of the box will be as set up in the current COLOR command or with the editor's default colors if no color command has been given.

CHOICE

Syntax : CHOICE X1 X2 Y1, "Phrase To Display", Command To Perform

X1 is the column in which the phrase begins; X2 is the last column of the phrase; and Y1 is the line number. X2 is not necessarily X1 + the length of the phrase, since phrases with a different background color from the menu should all be the same length when in the same menu.

This command displays the phrase in quotation marks on the screen at the location given. Successive CHOICE commands constitute a menu, with a cursor that moves up and down with the arrow keys and allows a choice by the user. When a choice is made, the original screen replaces the menu, and the command associated with the phrase chosen is executed. If the command is blank, the menu is simply removed, as might occur with a help screen, for example.

TITLE

Syntax : TITLE X1 Y1 "Text to be displayed"

The text in quotation marks is displayed at the location given.

COLOR

Syntax : COLOR Backgroundcolor Textcolor

Backgroundcolor must be an integer 0-7. For the available Textcolors, see the setup screen by running the main EPI menu and pressing <F9>. This command sets the color of the boxes subsequently created before the next ENDCHOOSE.

BOX, TITLE, CHOICE, and COLOR can only be used after a
CHOOSE command and before ENDCHOOSE has occurred.
Note that they are not preceded by the underline character.

Example: The file EPIAID.EPA, the main menu for the EPIAID system,
illustrating the use of CHOOSE, ENDCHOOSE, COLOR, TITLE,
CHOICE, and BOX.

```
_CHOOSE 1 1 80 25
Color 1 14
Title 32 4 "EPIAID MAIN MENU"
Color 3 1
Title 30 8 "EPIDEMIC INVESTIGATION"
Choice 28 53 10 "        Epi I Memo" RUN epi1.epa
Choice 28 53 11 "   Make Questionnaire" RUN ques.epa
Choice 28 53 12 "  Develop Study Design" RUN design.epa
Color 3 1
Title 36 14 "TUTORIAL"
%Color 4 12
Choice 28 53 16 "Word Processing Tutorial" RUN
wptutor.epa
Color 1 14
Box 25 6 56 19
Color 7 0
Title 21 "SELECT A TOPIC USING THE UP AND DOWN ARROW
KEYS"
Title 22 "THEN PRESS <ENTER>"
Title 23 "PRESS <F10> TO RETURN TO EPED"
_ENDCHOOSE
```

_CLEARALL

Syntax: _CLEARALL

Removes all text from the screen. This is useful if the text developed has
been for illustrative or educational purposes only.
Example:
   _READFILE Helpful.txt
   [Press <Esc> when through]
   _CLEARALL

_CLOSEFILE

Syntax: _CLOSEFILE <File name>

This closes a previously open file, asking the user if new text should be saved, if any has been added.

Example: **_CLOSEFILE MEMO.TXT**

_DEFINE

Syntax: _DEFINE <Variable name>

A variable is created. It may then be assigned values with the LET command, and the values may be retrieved with the WRITE command or used in expressions such as those in the IF command. Variables are kept in random access memory (RAM) and disappear when the end of a program file is reached, a QUIT command is given, or EPIAID stops running. If they are to be preserved for future use, they should be placed in a file using the STORE command and later brought back to memory with the RETRIEVE command.

A variable name must begin with an alphabetic character, not a number, and may be up to 20 characters long. Upper and lower case characters are treated as though they were all upper case. Variable names must be all one word, i.e., they cannot contain spaces. Underline characters, periods, and colons may be included, but not hyphens (which would be confused with a minus sign), parentheses, or brackets.

Examples:
>   **_Define A**
>   **_Define WhatsItsName**
>   **_Define Random.Number**
>   **_DEFINE RANGE3**
>   **_Define AlmostTooLongVarName**

_DISPLAY
_ENDDISPLAY

Syntax :

```
  _DISPLAY
 Text of any kind to be displayed here.  May be more than one line.
 _ENDDISPLAY
```

The text between the two commands will be displayed in a window centered on the screen. To move to the next item, the user presses <Esc>.

_DOS

Syntax: _DOS { (DOS command or file to run)}

This command allows use of DOS commands from within EPIAID. If a
DOS command is given, it will be automatically performed and then the
program will return to EPIAID. If no DOS command is given, the user is
presented with the DOS prompt and allowed to give DOS commands.
When the command EXIT is given, however, the program instantly
returns to EPIAID at the place where the DOS command occurred.

Example: **_DOS ENTER FORM2**
(Runs the ENTER program and allows entry in the questionnaire called
FORM2.)

EXISTFILE

IF EXISTFILE <File name> THEN...

This feature is useful for writing Epiaid programs where an action
depends on the existence of a file. If a file does not exist, for example,
you may want to display a message, or have the program search in another
directory for the file, or skip a command that requires the file.

Example:
 _IF EXISTFILE PRACTICE.TXT THEN DOS DEL PRACTICE.TXT

_IF

Syntax:

_IF (Boolean Expression) THEN (Command) ELSE (Command)

This is the conditional command. If the boolean expression is true (= 1),
the command after THEN will be executed; otherwise, the ELSE
command, if any, will be executed. In EPIAID, nested IF's (IF statements
containing other IF statements) are not allowed.

Example:
_IF ENTRY = "N" then JUMP CASECONTROL

_JUMP

Syntax: _JUMP (Label)

The program file is advanced to the label name given. Labels consist of a
single line preceded by a colon on the left margin. They must not contain
spaces or hyphens (see variable names under LET). The label given must
be below (after) the JUMP statement in the file. If no such label exists, an
error message is given.

Example:

  _IF ILL = YES THEN JUMP IllnessSurvey
  % Processing for Exposure Survey (skipped if the answer was YES)
  :IllnessSurvey
  % Processing for Illness Survey

_LEFT

Syntax: _LEFT ON   { If ON and OFF are omitted, ON is assumed}
              OFF

When LEFT is ON, lines begin at the left margin but are not padded with
spaces to reach the right margin. This command has the same effect as
turning RLJUSTIFY OFF.

Examples:
          _LEFT
          _LEFT ON
          _LEFT OFF

_LET

Syntax: _{ LET} <Variable name> = <Expression>
     (The assignment command LET is optional.)

The value of the expression is assigned to the named variable. An
expression can be any combination of numbers, variable names, and
operators that has a numeric or logical result, or it can be a string (text
phrase) enclosed in quotation marks. A simple assignment would be
_A = 10

which is identical to

_LET A = 10

If the variable "A" does not exist, it will be automatically defined.

A standard variable called ENTRY is automatically defined by EPIAID. After user input has occurred, prompted by a request in square brackets such as [Your name please:], the user's response will be contained in ENTRY. It may then be used for a decision:

_IF ENTRY = "" then JUMP END1   ("" means a blank entry)

or transferred to another variable for later use

_NAME = ENTRY

Examples:
    _Study = "Case-Control"
    _Date = "02/27/89"
    _Let Date = "02/27/89"
    _Name = Entry
    _Numcases = 22

_MARK

Syntax: _MARK

This command marks a place in the text, like a bookmark. The REMOVE command can then be used to remove text back to the MARK. See REMOVE for example.

_OPENFILE

Syntax: _OPENFILE <File name>

This command opens the named file if it already exists and brings it into the screen workspace. If the file does not exist, the name is placed at the bottom of the screen, and any subsequent file saves will place text in that file.

Example: **_OPENFILE MEMO.TXT**

_QUESTIONS

Syntax: _QUESTIONS

Displays Epi Info question formats and allows a choice of one. Further information is requested, if needed, about the length of the field. The chosen field is inserted into the text stream.

Example: **_Questions**

_QUIT

Syntax: _QUIT

Closes the current program file and deletes all defined variables from memory. It also closes the variable and backup files created by EPIAID. QUIT is optional at the end of a program file, since EPIAID QUITs automatically when the end of the file is reached.

Example: **_QUIT**

_READFILE

Syntax: _READFILE <File name>

Inserts the contents of the named file into the workspace at the current cursor position. Unlike _OPENFILE, it does not change the name of the currently open file, if any.

Example: **_READFILE CASEDEF.TXT**

_REMOVE

Syntax: _REMOVE

Removes text from the screen back to the last _MARK command. This is often used to remove a phrase from the text if the user presses <Enter> rather than making an entry.

Example: **_MARK**

> **Further information was obtained from**
> **[Source of information (<Enter> if none)]**
> **_If Entry = "" then REMOVE**
> **% The sentence is removed from the text if there**
> **% is no "source of information."**

_REPEAT
_UNTIL

Syntax:

    _REPEAT
          (Other commands)
    _UNTIL (Boolean expression)

The commands between the REPEAT and the UNTIL are performed at
least once, and then again and again until the Boolean expression is true.
This command is useful for repetitive operations that will be terminated
by having the user type a word such as "done." Care should be taken that
the Boolean expression does indeed become true, or the program will
remain in the loop indefinitely.

Example:

_REPEAT
    [Enter an objective]
    [Enter as many methods as needed for this objective]
    [Type "Done" if no more objectives]
_UNTIL ENTRY = "Done"

_RLJUSTIFY

Syntax: _RLJUSTIFY ON   (If ON and OFF are omitted, ON is assumed)
                        OFF

When RLJUSTIFY is ON, lines are padded with spaces to fill the area
between the right and left margins. Turning RLJUSTIFY ON also turns
WORDWRAP ON, CENTER OFF, and SINGLELINES OFF.

Examples:

   _RLJUSTIFY
   _RLJUSTIFY ON
   _RLJUSTIFY OFF

_RUN

Syntax: _RUN <File name>.EPA

This command opens an EPIAID program file and executes the commands
in the file, line by line, until the end of the file is reached or a QUIT or
RUN command causes another file to run.

Example:
   _RUN QUES.EPA

_SINGLELINES

Syntax: _
SINGLELINES ON   (If ON and OFF are omitted, ON is assumed)
                OFF

When SINGLELINES is ON, each line of text from the program file
begins on a new line. WORDWRAP is turned OFF.

Examples:
   _SINGLELINES
   _SINGLELINES ON
   _SINGLELINES OFF
_WRITE

Syntax: _WRITE "(Text)"

Appends the text given in quotation marks to the end of the text on the
screen. A space is inserted before the text. If the text on the screen ends
with a period, exclamation point, or question mark, an extra space is
inserted.

Example:
   _WRITE "The date of first contact was "

(Blank to maintain chapter pagination of Version 6.00 manual.)

(Blank to maintain chapter pagination of Version 6.00 manual.)

(Blank to maintain chapter pagination of Version 6.00 manual.)

.CHK File Commands

After
Autojump
Autosearch
Before and **After File Record Entry**
Call
Clear
Comment Legal
Color
Codefield
Define
Enter
Exit
Fieldchar
Goto
Help
Hide, Unhide
Include
Jumps
Key
Legal
Let
Load
Mustenter
Noenter
Range
Relate
Repeat
Showmemory
Soundex
Type
Type Comment
Unhide
Unload
Update
Use

Chapter **35**

ENTER and CHECK Commands

Running ENTER from the DOS Command Line

The commands in ENTER are described in Chapter 8. Running ENTER from DOS or from the EPI menu with parameters allows quick entry without having to specify file names within ENTER. It also allows running ENTER from other programs or batch files that are not normally part of the Epi Info system. The available parameters are as follows:

ENTER <.REC file name> { <.QES file name>} {/WS} {/CONF}
        {/AUTO} {/OVER} {/CHK:<Name of .CHK file>} {/Single}
        {/VERIFY <Fieldname1> <Fieldname2> etc.}

/WS turns on a filter that removes non-ASCII characters in questionnaire files as a .REC file is made. Questionnaires made in WordStar should be processed with the /WS parameter to filter out the non-ASCII characters contained in WordStar files.

/CONF stands for CONFIRM. If this parameter is present, the cursor does not leave an entry field until <Enter> is pressed. This is useful when a questionnaire contains a mixture of long and short fields.

/AUTO causes ENTER to write each record to disk after the last field is entered,without asking the question, "Write data to disk (Y/N)?".

/OVER allows the cursor to escape from MUSTENTER fields with the <down arrow> key and will permit entering exceptions to legal values or codes with this key.

/SINGLE forces ENTER to operate in single-user mode, even if a local area network is present.

/CHK <Name of .CHK file> specifies a .CHK file that does not have the same name as the .REC file.

/VERIFY <Fieldname1> <Fieldname2> etc. puts ENTER into VERIFY mode-- choice number 4 on the ENTER menu. One or more fieldnames may be given as record identifiers. The data file name must be given before /VERIFY.

If a .QES file name is given and the .REC file exists, the .REC file is *revised* according to the .QES file. The .REC file is renamed to <File name>.OLD.

Parameters may also be used from the EPI6 menu. Place the bar cursor on the ENTER item in the menu and press <Ctrl-Enter>. A dialog box will pop up and allow entry of parameters.

General Features of CHECK Programs

.CHK files are arranged in blocks. A block begins with a field name and ends with the word END. All commands must be within a field name block. *Comments*, preceded by an asterisk ("*") may be placed between or within blocks. The "\" line continuation character in ANALYSIS is not allowed in .CHK files, but lines may be long--up to 255 characters.

Commands in a block are activated either before or after an entry is made in the field. The usual case is that the commands are performed after an entry has been made and after <Enter> has been pressed or the cursor has left the field automatically. Command within a block can be placed within an inner block called BEFORE ENTRY or AFTER ENTRY to cause the command to execute at the desired time.

A general format for a .CHK file is:

```
<Field name>
    BEFORE ENTRY
       <Command>
       <Command>
    END
    <Command>
    <Command>
    etc.
END

<Field name>
    <Command>
    <Command>
    etc.
END
```

The simpler commands are inserted in .CHK files automatically by the CHECK program, when function keys are pressed. Others must be inserted with a text editor

such as EPED or the editor that appears in CHECK when <F9> is pressed. It is often helpful to use CHECK to insert a few commands or a few codes or jumps and then to edit the resulting file by adding additional codes, jumps or legal values. This is the easiest way to handle large numbers of codes for diseases or counties, for example.

.CHK files are optional, and only a few fields may have blocks of commands within a .CHK file. The order in which fields appear in the .CHK file is not important. Indentation is useful for clarity but is ignored by the program. Every field name must have a corres ponding END marker, as must every command requiring an END. Special blocks called BEFORE RECORD and AFTER RECORD may be used for commands that are to be executed before or after all entries have been made in a questionnaire. BEFORE FILE and AFTER FILE blocks provide a way to execute commands before the questionnaire is presented to the user, or on exit from the ENTER program.

In the following instructions, the words "Relative Column #" refers to a number from 1 to 80 giving the left/right position on the screen. If the number is preceded by "+" or "-", it refers to the position relative to the first character of the current field; if no sign is given, the position is relative to the left side of the screen, taken as column 1.

"Relative Row #" is a number from 1 to 25 that refers to top-to-bottom screen position. If no sign is present, the position is relative to the top row as row 1; signs indicate position relative to the first character of the current field.

Program Commands in .CHK Files

AFTER See BEFORE

AUTOJUMP See JUMP

AUTOSEARCH

Syntax:
 AUTOSEARCH { SOUNDEX} { LIST} { <Field name>...<Field name>}

AUTOSEARCH causes ENTER to search for one or more matching records. If a match is found, a choice is offered of displaying and editing the matching record(s) or of ignoring the match and continuing to enter the current record.

While a search is occurring, the numbers of the records being searched are displayed. A search may be terminated at any time by pressing <Esc>. To resume the search, use the F3 and F4 keys to go to the previous or next match.

If LIST is specified, results are displayed in a spreadsheet format that allows several records to be displayed. To see more than one screen of such matches--all the SEX = "M" records, for example--press <PgDn> to see the screens after the first. When matches have been displayed, the bar cursor may be moved up and down with the up and down arrow keys, and a particular record selected for editing by pressing <Enter>. To avoid editing any of the matches, press <Esc>. The fields displayed as the result of a LIST search are determined as follows:

1. If a single field is the key, the field and as many other fields as possible are displayed.

2. If there are multiple keys, these are displayed first and then as many of the remaining fields as possible are shown, starting with the first.

The AUTOSEARCH command in a .CHK file can be followed by a list of fields to match on. If no fields are specified, AUTOSEARCH works as in Version 5, searching on all non-blank fields except <IDNUM> and <TODAY> fields.

AUTOSEARCH can specify whether or not to use available Soundex keys (see SOUNDEX below). AUTOSEARCH SOUNDEX will use the Soundex keys on all search fields which have them; AUTOSEARCH alone will use only the normal keys.

By default, AUTOSEARCH displays only the first matching record, unless given the argument LIST:

AUTOSEARCH LIST [SOUNDEX]

AUTOSEARCH can be given a list of fields to use for the search, rather than relying on the order of the fields in the questionnaire. If a .REC file has the fields:

FIELD1 ...
FIELD2 ...
FIELD3 ...
FIELD4 ...
FIELD5

and the .CHK file contains:

FIELD3

AUTOSEARCH FIELD2 FIELD3
END

then AUTOSEARCH will ignore the value of FIELD1 and search only on the values of FIELD2 and FIELD3.

NOTE that the arguments LIST and SOUNDEX, if present, must precede any field names:

AUTOSEARCH [LIST] [SOUNDEX] [FIELDNAME1 ... FIELDNAMEn]

If by some chance there are search fields actually named LIST or SOUNDEX, they will be misinterpreted if they appear as the first argument(s) to AUTOSEARCH.

BEFORE and **AFTER**

Syntax:
   BEFORE FILE
   AFTER FILE
   BEFORE RECORD
   AFTER RECORD
   BEFORE ENTRY
   AFTER ENTRY

These commands will control actions so that they occur before or after a file, record, or field is accessed.

Sometimes is is desirable to perform an action such as a calculation or to present a help message at a specific time during data entry that is not the standard time for activation of a block of .CHK commands--i.e., after an entry has been made in a field and the <Enter> key has been pressed.

The commands are used in blocks, starting with the command itself and ending with END. BEFORE ENTRY and AFTER ENTRY are always contained within a block for a field within the questionnaire. (See examples below.)

BEFORE FILE is executed each time the file is loaded (i.e., the "core" file's BEFORE FILE is executed when ENTER starts it up; a related file's BEFORE FILE is executed whenever ENTER jumps to the related file). An AFTER FILE block is activated when the file is closed (when ENTER terminates successfully or when it leaves a related file). You might use these to present instructions to the data entry person, reminding him/her BEFORE FILE about how to get help or see codes, and AFTER FILE to make a backup copy, for example.

BEFORE ENTRY is executed when the cursor moves into the field, before an entry is made. AFTER ENTRY is executed at the time the cursor leaves the field in whose block the command occurs. AFTER ENTRY is the normal time when commands in a field block are executed. It is provided for clarity and in case there are commands both BEFORE ENTRY and AFTER ENTRY within a single field block.

NOTE: There is no functional difference between using a BEFORE ENTRY block and/or an AFTER ENTRY block, and using the command ENTER by itself in the middle of a block without BEFORE ENTRY and/or AFTER ENTRY. These new commands are clearer and provide better self-documentation of the code.

BEFORE RECORD is executed when the cursor moves into a new or existing record. AFTER RECORD is executed before the record can be saved to disk--that is, when the message "Write data to disk? Y/N" would normally appear. An AFTER RECORD block is a good place to check the contents of key fields to be sure that the entries are correct before they can be saved. To prevent the record from being saved to disk, be sure to include a GOTO command in the AFTER RECORD in conjunction with an IF statement.

Here are samples of a .REC file and .CHK file to experiment with some of these commands:

The .REC file header contains fields named FIELD1, FIELD2, FIELD3.

```
*The .CHK file
BEFORE FILE
  HELP "This is 'BEFORE FILE'" 20 2
  FIELDCHAR X ALL
END

BEFORE RECORD
  HELP "This is 'BEFORE RECORD'" 20 6
END

FIELD1
  BEFORE ENTRY
    HELP "This is 'BEFORE ENTRY' for FIELD1" 20 10
  END
  ENTER
  AFTER ENTRY
    HELP "This is 'AFTER ENTRY' for FIELD1" 20 14
  END
END

FIELD2
  HELP "This is implicit 'BEFORE ENTRY' for FIELD2" 20 10
  ENTER
  HELP "This is implicit 'AFTER ENTRY' for FIELD2" 20 14
END
```

```
AFTER RECORD
  HELP "This is 'AFTER RECORD'" 20 20
  GOTO FIELD2
END

AFTER FILE
  HELP "This is 'AFTER FILE'" 20 23
END
```

Here is a partial example of AFTER RECORD and EXIT used in a GEPI file:

```
AFTER RECORD
 IF ID < 101000001 THEN
    HELP "Bad value of ID. Quit without saving and restart." 10 15
    GOTO LASTNAME
     * GOTO in order to avoid saving the record
    EXIT
   ENDIF
   IF EVENTNAME = . THEN
     HELP " EVENTNAME cannot be missing. " 10 15
     goto EVENTNAME
     EXIT
   ENDIF
   IF EVENTDATE = . THEN
     HELP " EVENTDATE cannot be missing. " 10 15
     goto EVENTDATE
     EXIT
   ENDIF
   IF MMWRWK = . THEN
     HELP " MMWRWK cannot be missing. " 10 15
     goto MMWRWK
     EXIT
   ENDIF
END
```

CALL

Syntax: CALL <Program name without suffix> <interrupt number>

This command accesses a Terminate and Stay Resident (TSR) program from a field during data entry. The process is described in Chapter 39. CALL runs a program with the suffix .BIN at the location directed by the interrupt. If no program has been LOADed, nothing happens. The program must terminate with the FAR RETURN instruction RETF followed by the parameter 18 for stack size so that control is returned to ENTER. Information can be passed to and from the binary routine by means of the stack. A sample program called ENTFACE.BIN is included on the disk to provide further information for assembly language programmers.

An example using a binary program to call a Pascal procedure is given in Chapter 23 on anthropometric software, in which

complex calculations are done by a Pascal program called from a .CHK file. Values are passed to the Pascal program from fields in the Epi Info questionnaire, and the Pascal program places values in other fields in the questionnaire before returning control to the .CHK file.

CLEAR

Syntax: CLEAR <Field name>

CLEAR sets the field named to the missing value, as though the field had been left blank. It is useful to clear a previous entry after an error has been detected, for example.

Example: **CLEAR DISEASE**

CODEFIELD

Syntax: CODEFIELD <Field name>

CODEFIELD statements are inserted by the <F5> command in CHECK or may be (carefully) inserted with an editor. CODEFIELD sets up a link for values entered in one field to result in codes being inserted in another field automatically, but no coding occurs without an accompanying CODES statement. When two fields are linked reciprocally, values may be entered in either field, and the corresponding value will appear in the other field. It is not necessary to have a two-way link, however, and sometimes it is desirable to have the CODEFIELD command in just one of the two blocks, so that coding proceeds in one direction only.

Example:
COUNTY
   CODEFIELD COUNTYCODE
   CODES
        "Yellow Medicine" 10
         Hennepin 11
          etc.
     END
 END

      COUNTYCODE
        CODEFIELD COUNTY
END

Note that "Yellow Medicine" is in quotation marks because it contains a space; this is not necessary for "Hennepin."

CODES

Syntax:
      CODES
           <Value> <Code>
           <Value> <Code>
           etc.
      END
or
      CODES
           USE <Field name>
      END
or
      CODES <File name>

CODES are inserted automatically in the .CHK file by the CHECK program when <F8> is pressed. In almost all cases, the <F5> or LINK command is used first to set up another field in which the codes will appear. The <F5> command inserts CODEFIELD statements in each of the two fields being linked. Without CODEFIELD, however, CODES inserts the coded results in the same field in which they are entered, performing a sort of RECODE on the fly. It might be used to allow synonyms or abbreviations to be entered with the result being a single uniform code.

Large code or legal-value lists can be put into external (.REC) files rather than in the .CHK file. To do this, use the syntax CODES <File name>.REC,where <File name>.REC is the name of the .REC file. Each .REC file containing codes must have at least two variables declared as KEY in the corresponding .CHK file. The first KEY variable (KEY 1) is for the first value (the one the user enters). The second KEY variable (KEY 2) is for the coded value. (Usually, the first variable is for the field that declares CODES, the second is

for its CODEFIELD.)

Example:
COUNTY
  CODES
      "Yellow Medicine" YM
      Hennepin HENN
      etc.
  END
END
COLOR

Syntax:
  COLOR QUESTION <colorno>
  COLOR DATA <colorno>
  COLOR BACKGROUND <colorno 0-7 only>
  COLOR <field name> <color of data> <color of question>

The COLOR command specifies the color for either the entire
questionnaire (with QUESTION, DATA, or
BACKGROUND), or for a single field. <Colorno> is an
integer from 0 to 255 that specifies the color of text and
background that will be used on the screen. To see the
possibilities, press <F9> from the main menu to view the
Setup screen where all 256 combinations are displayed.

The word QUESTION refers to text on the screen outside of
data entry blanks. DATA is the text inside entry blanks.
BACKGROUND is the background area.

The colors specified do not take effect until the screen is next
updated, usually when the bottom of a screen is reached. To
produce an immediate effect, the UPDATE command should
follow the COLOR command. COLOR commands applying
to an entire questionnaire should be placed in a BEFORE
FILE or BEFORE RECORD block.

Example:

BEFORE FILE
  COLOR QUESTION 14
  COLOR DATA 15
  COLOR BACKGROUND 3
  COLOR NAME 12 11

      UPDATE
END

This sequence displays the questionnaire in yellow (14) and white (15) on a green (3) background, with the NAME field having a cyan (11) prompt and red (12) data blank. If this sequence occurs in the BEFORE FILE or BEFORE REC block of the questionnaire, the change will take effect immediately; otherwise, it will occur according to the location of the color command within individual field blocks.

COMMENT LEGAL

Syntax:
COMMENT LEGAL
   <Code> <Comment>
   <Code> <Comment>
   etc.
END

or   COMMENT LEGAL <File name>.REC

COMMENT LEGAL inserted in a .CHK file displays a list of code choices when <F9> is pressed, and allows a choice of codes by moving the cursor bar and then pressing <Enter>. It differs from LEGAL by allowing "comments" that describe the meaning of the codes. LEGAL is inserted automatically by the <F6> command in the CHECK program. COMMENT must be added by using EPED or the CHECK editor.

If a file name is given rather than a list of codes, the file must be an Epi Info .REC file with at least two fields, and these two fields must be indexed by placing the word KEY in a .CHK file and using ENTER to perform the indexing. COMMENT LEGAL will use the first two fields as the source of legal values and comments, just as though they had been listed in the .CHK file itself. COMMENT LEGAL uses the fields that are KEY 1 and KEY 2 in the file, and other fields will be ignored.

Examples:

COMMENT LEGAL

    1   **"Ages 1-21"**   (Note quotation marks if spaces occur.)
    2   **Adult**
  END

   COMMENT LEGAL COMMENTS.REC

DEFINE

Syntax:
DEFINE VARNAME <Field indicator, such as ####>
   { CUMULATIVE or GLOBAL}

(Note: Field types must be specified as _____, #####, etc. The older designations STRING, NUMBER, etc., are not supported.)

DEFINE allows new variables to be defined as they are in ANALYSIS. These variables may be used to hold intermediate values in calculations, to carry over items from one record to the next, or to transfer the contents of a field from one file to another related file.

The value of GLOBAL and CUMULATIVE variables is retained from record to record. The value of GLOBAL, but not CUMULATIVE, variables is retained when a related file is accessed. GLOBAL variables, therefore may be used to pass variable values to a related record. If neither word is included, the variable will be set to MISSING between records.

Example:

This is a modified .CHK file for the files HOUSE.REC and PERSON.REC relational example that uses GLOBAL DEFINEd variables:

The file, HOUSE.REC, contains the following variables:

```
HOUSEHOLDR
HOUSEID
ADDRESS
CITY
STATE
BEDROOMS
WATER
```

The .CHK file, HOUSE.CHK, contains the following blocks:

```
BEFORE FILE
  DEFINE varADDRESS  _____  GLOBAL
  DEFINE varCITY     _____      GLOBAL
  DEFINE varSTATE    __                                GLOBAL
END

HOUSEID
  KEY UNIQUE
END


STATE
  Legal
    GA
    FL
    AL
    SC
  END
END

WATER
  AFTER FILE
    varADDRESS = ADDRESS
    varCITY    = CITY
    varSTATE   = STATE
    ENTER PERSON.REC HOUSEID
  END
END
```

PERSON.REC is a related file. PERSON.CHK contains the following blocks.

```
BEFORE
  ADDRESS = varADDRESS
  CITY    = varCITY
  STATE   = varSTATE
END

HOUSEID
  KEY 2
END
```

ENTER

Syntax:
ENTER
ENTER <.REC File name> <Linking field name> { 1}

The second form of this command is provided for
compatibility with Version 5 of Epi Info only. It is
recommended that new programs use the RELATE
command. See RELATE for description of this second form.

The command ENTER as a single word indicates the place in
a sequence of commands in a field of the .CHK file where
data entry will occur. Commands prior to ENTER will be
executed as soon as the cursor enters the field; those
following ENTER will occur after a data item has been
entered in the field. If ENTER is not included in the
commands for a field, all commands are executed after the
data item is entered, that is, ENTER is assumed to be at the
beginning of the command sequence for that field. ENTER
may occur only once in the commands for a given field.

Examples:
See COLOR above for an example using ENTER as a command to
determine the order of data entry and .CHK file commands.

EXIT

Syntax:
EXIT

EXIT is used to terminate processing of a sequence of
commands, as when one error condition has been determined
in a sequence of several checks. EXIT can be used to move
the user's focus from an error message to the appropriate field
for error correction. EXIT skips all remaining commands in
the same block.

Example:
If DISEASE = "N" then EXIT

FIELDCHAR

Syntax:
FIELDCHAR x { <Field name> or ALL}

where "x" is any single character.

FIELDCHAR sets a field's "field character" to the specified
character. The field character is the character that represents
a data entry field in the screen display, often a space or
underline character.

If no field name is specified, the current field (the one to
which the check code block pertains) is assumed. If a name is
given, the field character is set for the specified field. If the
field name is given as ALL (and no field named ALL actually
exists) then the command applies to all fields in the current
.REC file.

Example:

FIELDCHAR _ ALL
fills the empty portion of all fields with underline characters.

GOTO

Syntax:
 GOTO <Field name>
 GOTO WRITEREC

The GOTO command causes the cursor to jump to the field
name given. If the destination is the reserved word,
WRITEREC, the end of the questionnaire is understood, as
though the <End> key had been pressed.

Examples:
IF ILL = "N"
THEN
   GOTO BAKEDHAM
ENDIF

IF SYMPTOMS = "N"
THEN

    GOTO WRITEREC
    ENDIF

HELP

  Syntax:
   HELP "<Text>" <Col> <Row> { KEYS = <Input Chr(s)>}
   HELP <File name> <Topicno> <Col> <Row> { CONTEXT}

  The HELP command displays text specified in the command
  in a window at the location specified on the screen. Pressing
  the <Esc> key restores the screen to its previous state. If
  <Input chr(s)> are specified, any of the specified characters
  may be typed from the keyboard. There is a standard,
  predefined variable called RESULT that is set equal to the
  number of the character in the sequence of <Input chr(s)>.
  Thus, if the KEYS are ABCD and "C" is pressed, RESULT
  will be equal to 3. The value of RESULT may be tested for
  use in IF statements that follow.

  If HELP is used with a file name, the file referred to must be
  in the Epi Info Help File format, consisting of text, optional
  highlight commands, and markers that divide the file into
  topic sections. <Topicno> is an integer that identifies the
  Topic block within the file. The help file must have been
  indexed by using the HELPPREP program supplied with Epi
  Info. See "Epi Info Help Files" in Chapter 37.

  The "<Text>" may contain variable names preceded by the
  "@" sign. The following will show the value of the variable,
  MONTH, in the HELP message: HELP "This is @MONTH"
  5 5

If the special word CONTEXT is placed after the HELP statement, the help message
only appears if the user presses <F1> or <Shift-F1> with the cursor in the field for
this block of commands. In the special case where HELP CONTEXT appears in the
BEFORE FILE block, the TOPIC specified will pop up when <F1> is pressed. Help
specific for other fields requires pressing <Shift F1> or entering a question mark in
the field.

Examples:
 **IF MED = "TETRACYCLINE" AND (PREGNANT="Y") THEN
 HELP "Caution: This drug is contraindicated in pregnancy\nAre
you sure(Y/N)?" KEYS="YN"**
 (Note: The two lines above must be all on one line.)
   **IF RESULT = 1 THEN
      GOTO DOSE
   ELSE
      CLEAR MED
      GOTO MED
   ENDIF
ENDIF**

**NAME
   HELP USERHELP 12 5 5 CONTEXT
END**

In the second example, TOPIC 12 of USERHELP.HLP will be displayed if
the user types "?" or presses <Shift-F1>.

Other examples of CONTEXT sensitive help are contained in
HOUSE.CHK, and may be tested by running HOUSE.REC in ENTER.

Help files may be may be constructed to work with the HELP command in
.CHK files using an ASCII editor, such as EPED. They are then run through
the HELPPREP program to compress and index the file for use in the help
system.

Several commands are used within help files, and these are described in
more detail in Chapter 37. The help file commands are preceded by an
exclamation point on the left margin, as follows:

!WIDTH <Width of window on the screen, up to 78 characters>

This sets the width of the window that will be displayed. The actual width
available for text display is this number minus four, since the edges of the
window each take two characters.

!INDEX <Number indicating the order in which topic should appear

Normally when the automatic menu appears, it is sorted in alphabetical order by topic name. The numbers in an INDEX statement following the TOPIC statement will override this order. Small numbers cause topics to appear at the top of the list and large ones cause the topic to be near the end, allowing you to control the order in which the items appear, if alphabetical order is not desired.

!TOPIC <Topic number> <Name of topic>

Each block of subject matter must begin with this command. The numbers are from 1 to 32,000, and the topic name may be any phrase, with or without spaces. Quotation marks are not necessary unless you want them printed at the top of the help window with the topic name. See Chapter 37 for a description of the !SUBTOPIC command.

!NOWRAP

Normally, the text is automatically word wrapped to fit the size window selected with the !WIDTH command. If you do not want this to happen, the !NOWRAP statement will turn off this feature. Lines longer than the window would simply be truncated in this case. The !WRAP command turns wrapping back on.

!PAGE

This command marks the end of a screen page, so that text above will be displayed and blank lines will appear below the line containing !PAGE. This allows you to decide where to place page breaks in blocks that are longer than one screen and therefore require scrolling with the <PgUp> and <PgDn> keys.

When the help file is complete, it must be run through the HELPPREP program before use. To run HELPPREP, type from the DOS prompt:

HELPPREP <Text file name> <Completed help file name>

For example:

HELPPREP EHELP.TXT EHELP.HLP

HELPPREP will construct a binary file that can be called from the HELP command in a .CHK file as described in Chapter 22.

HIDE, UNHIDE

Syntax:
HIDE { <Field name> or ALL}
UNHIDE { <Field name> or ALL}

HIDE hides a field from view and makes it NOENTER.

UNHIDE makes a field visible and returns it to its normal
NOENTER status. (i.e., if it wasn't NOENTER before the
HIDE command, then it is no longer NOENTER.)

If no field name is specified, the current field (the one to
which the check code block pertains) is assumed. If the field
name is given as ALL (and no field named ALL actually
exists) then the commands apply to all fields in the current
.REC file.

IF

Syntax:
IF <expression> THEN
   <command(s)>
 { ELSE
   <command(s)>}
ENDIF

The IF command executes commands only if the
<expression> is true. If the optional ELSE clause is included,
the alternative commands are executed when the
<expression> is false. The commands may be any of those
that follow, or another IF command. Expressions may be any
logical or mathematical expressions valid in ANALYSIS.
See OPERATORS and the LET command in the ANALYSIS
reference section.

Examples:
IF DATEBIRTH > "01/01/89"
THEN

    CLEAR DATEBIRTH
    HELP "Please check date; it seems too recent.\nPress ESC" 5 5
ENDIF

IF (FIELD1 + FIELD2)^2 < 100 THEN
 FIELD3 = 10
ELSE
   FIELD3 = (FIELD1 + FIELD2)^2
ENDIF

IF HEPATITIS = "Y" THEN
   ENTER HEP CASEID
ENDIF

INCLUDE

Syntax: INCLUDE <File name>

Portions of a .CHK file can be kept in separate files and referenced using the INCLUDE command. The statement INCLUDE <File name> in a block of a .CHK file will cause the named file to be included in the check file as though its contents were part of the original check file.

```
  COUNTY
      CODEFIELD COUNTYCODE
      CODES
          INCLUDE COUNTIES.TXT
      END
  END
```

The INCLUDEd file name (e.g., COUNTIES.TXT) may include a drive and path. If none is specified, the current drive and directory are used. The extension of the INCLUDEd file name (e.g., .TXT in COUNTIES.TXT) must be specified.

The INCLUDEd file is processed EXACTLY as if it appeared in place of the INCLUDE command. When its processing is completed, processing of the previous file will continue exactly as if it had been the only file, and the INCLUDEd file had appeared within it.

An INCLUDE file may INCLUDE another file.

JUMPS

Syntax:
JUMPS
   <Value> <Target field name>
   <Value> <Target field name>
   etc.
END

AUTOJUMP <Target field name>

JUMPS are inserted automatically in the .CHK file by the
<F7> command in the CHECK program or may be typed in
using an editor. For each value given, the cursor will jump to
the target field when that value is entered in the field.
AUTOJUMP is a special form of this command that jumps to
the target field regardless of the value entered. It is also
inserted by the CHECK program.

Example:
ILL
 JUMPS
    Y   DIARRHEA
    N   HOURLUNCH
 END
END

LASTNAME
 AUTOJUMP FIRSTNAME
END

KEY

Syntax: KEY { UNIQUE} { <Key number>}

This statement creates an *index* for the field in which it
appears. The index is another file that speeds the search for
records containing a particular value in the field. If the word
UNIQUE appears after KEY, the index is a special one for
unique identifiers. JONES cannot appear twice, and NAME
would therefore not be a proper choice for a KEY UNIQUE

field. Such fields are those containing unique identification numbers that are assigned only to one record each. KEY (without UNIQUE) can be assigned to any field and allows data values to occur in more than one record, so that it can be used with fields like NAME, COUNTY, or AGE.

KEYs have three purposes:
1. Searching for records in ENTER using KEY fields is many times faster than normal searches if the file is longer than a few hundred records.

2.When files are related during data entry so that ENTER switches automatically from file to file (see the RELATE command), they must be linked by identifier fields. The identifier field in the main file must be KEY UNIQUE and that in the related file must be KEY, although it can also be KEY UNIQUE.

3. When a .REC file is used with the CODES, LEGAL, or COMMENT LEGAL commands, the fields must be indexed, and KEY will perform this task. The file must be accessed at least once in ENTER so that the necessary index file is constructed. CODES and COMMENT LEGAL use fields that are KEY 1 and KEY 2 in the file and LEGAL uses the field that is KEY 1. Other fields will be ignored.

The Key Number is an optional number that identifies the sequence of the keys in the index file. This is only necessary if you insert additional KEY commands in your .CHK file or alter the order of the blocks in which they occur after constructing the index. The CHECK program inserts the key numbers as a precaution, but they are otherwise optional, and do not affect the operation of the key. Particularly if you are writing programs that others may alter, it is a good idea to add the key numbers (1, 2, 3, etc.) in case more keys are added later.

If a KEY field is longer than 30 characters, only the first 30 characters are used in the key.

Examples:

(In a file of households)
HOUSENO

      KEY UNIQUE
END
(In a related file of persons living in these households)
HOUSENO
    KEY
END

(To make searching by name fast)
NAME
    KEY
END

LEGAL

Syntax:
  LEGAL
    \<Code\>
    \<Code\>
    etc.
  END

or

LEGAL \<File name\>.REC

LEGAL inserted in a .CHK file displays a list of code choices
when \<F9\> is pressed, and allows the user to choose a code
by moving the cursor bar and then pressing \<Enter\>. Only
values defined as legal, or within RANGE limits, or CODES
or missing values (\<Enter\>) may be entered. Other values are
rejected. Missing values can also be rejected by making the
field a MUSTENTER field.

LEGAL is inserted automatically by the \<F6\> command in
CHECK. LEGAL by itself does not allow "comments" that
describe the meaning of the codes but these may be added if
the command is changed to COMMENT LEGAL in an editor.
See COMMENT LEGAL above.

If a file name is given rather than a list of codes, the file must
be an Epi Info .REC file with at least one field, and this field
must be indexed by placing the word KEY in a .CHK file and
using ENTER to perform the indexing. COMMENT LEGAL

will use the first KEY field (KEY 1) as the source of legal values, just as though they had been listed in the .CHK file itself.

Examples:
LEGAL
M
F
U
END

LEGAL VALUES.REC

LET

Syntax:
{ LET} <Field name> = <mathematical or logical operation as in ANALYSIS, using any field names in the questionnaire>

The word LET is optional

This command assigns a value to a variable. The variable named is set equal to the value of the expression, and this value is displayed on the screen when that field is visible. See OPERATORS in the ANALYSIS reference section for more information on allowable expressions.

The word LET is optional and most users prefer to omit it as in the second and third examples below.

Examples:

LET AGE = "01/01/89" - DateBorn
AGE = "01/01/89" - DateBorn
Samplesize =(N/4) (1+(1+(4/(N(P2-P1)))^0.5)^2)

LOAD

Syntax: LOAD <File name>.BIN

Loads a binary file into memory at a DOS segment boundary

so that it may be called from the .CHK file. More than one program may be loaded, since programs are distinguished by name.

Example: **LOAD ENTFACE.BIN**

MUSTENTER

Syntax: MUSTENTER

MUSTENTER specifies that missing values are not allowed. It is inserted in the .CHK field by the <F4> command in the CHECK program. If <Enter> is pressed in a MUSTENTER field before a specific entry has been made, the cursor remains in the field until an entry is made.

Example:

PERSONID
    MUSTENTER
END

NOENTER

Syntax: NOENTER

NOENTER in a field keeps the user from placing the cursor in the field or entering data. It is particularly useful for calculated fields that are not to be changed directly.

Example: **NOENTER**

RANGE

Syntax: RANGE <Minimum> <Maximum>

RANGE is inserted automatically by the CHECK program when the <F1> or <F2> (MIN and MAX) functions are specified for a field. Values falling outside this range are

rejected unless they are LEGAL values. The missing value is accepted unless the field is also designated MUSTENTER. Ranges may be numeric, dates, or strings (text).
INFINITY is valid as either the minimum or maximum in a RANGE statement, having the effect of removing restrictions on the minimum or maximum value. For example, RANGE INFINITY 100 means that any value equal to or less than 100 is valid.

Examples:
RANGE 1 99
RANGE A L   (Place values in quotes if they contain a space; otherwise quotes are optional.)
RANGE "01/01/87" "12/31/87"

RELATE

Syntax: RELATE <Identifier Field> <File name> {1}

RELATE is the command for linking files relationally at data entry. It instructs the ENTER program to load a related .REC file and to continue data entry in the related questionnaire. The related .REC file can be any .REC file except the main file whose .CHK file contains this command. Both files must contain identification fields with the same name that are used for linking records. This is usually an identification number, possibly created as an <IDNUM> field, but may also be a text field. Items in the linking field must be unique in the main file, i.e., there can be only one record having a given link-field value, but several records in the related file can have the same number.

If only one related record for every main record is to be allowed, the number "1" should follow the command; otherwise, a variable number of related records will be allowed for each record in the main file. The user presses <F10> to return to the main record after entering one or more related records. See Chapters 25 and 26 for examples.

In commands for a single field, a RELATE statement referencing another file can appear only once. Thus an IF statement causing a link when a disease code of 10, 12, or 13

is entered should be written as a single statement rather than
three separate IF statements.

If an identifier field is longer than 30 characters, only the first
30 characters are used in the relation and carried over to the
field in the related file.

REPEAT

 Syntax: REPEAT

REPEAT is inserted in the .CHK file when the <F3> key is
pressed in the CHECK program. The data item inserted in
this field in the previous record will appear in the field
automatically. It can be accepted by pressing <Enter> or
changed if desired. This feature is useful for saving work on
data entry for items like STATE that remain constant with
only a few exceptions over many records. Only the
exceptions need to be entered in detail.

 Example:
 STATE
   REPEAT
 END

SHOWMEMORY

 Syntax: SHOWMEMORY <Optional phrase>

SHOWMEMORY is a debugging command. When
SHOWMEMORY is inserted into a command block, it
causes the program to pause after loading the main .REC file
header and the .CHK file up to the point where
SHOWMEMORY occurs and display the amount of free
RAM space remaining. The information is useful in
understanding problems that may be related to limited RAM
space. The <Optional phrase> can be used to indicate where
the SHOWMEMORY command was placed.

 Example:
 BEFORE FILE
   SHOWMEMORY FIRST BLOCK IN FILE
 END

SOUNDEX

Syntax: SOUNDEX is used as a modifying word in KEY and AUTOSEARCH commands

SOUNDEX is a method of coding names so that differences in spelling are minimized. "Smith" and "Smyth" are both converted to the same SOUNDEX code, for example, so that subsequent searches will find both names.
When SOUNDEX is specified in a KEY command, an extra index for the field will be created in ENTER using the SOUNDEX keys. If the word, SOUNDEX, is inserted in an AUTOSEARCH command, the specified search uses any SOUNDEX KEYs that have been specified. Searches initiated from the screen with the <Ctrl-S> command will also use available SOUNDEX keys.

Soundex searches can be done only on key fields, which must be declared as KEY [UNIQUE] SOUNDEX [key number]. A Soundex key field will actually have two keys/indexes: the normal key/index and the Soundex key/index. The word UNIQUE and the key number apply to the normal key.

A program called SOUNDEX.EXE can be linked to ENTER to put actual SOUNDEX codes in a field. See SOUNDEX.DOC for details.

NOTE to advanced programmers: The key number of a Soundex key is 1 (one) higher than the key number of the field's normal key. Be sure to allow for this if you are assigning key numbers directly.

TYPE

Syntax: TYPE "<Text>" <Column #> <Row #> { <Colorno>}

The text between quotation marks is placed on the screen at the column and row (x and y) location specified. Column and row locations are explained at the beginning of this section on commands and may be absolute or, if a sign is included, relative to the first character of the current field. If a color is given, the text will be in that color (0-255).

Example:
IF SEX = "F" THEN

```
        TYPE "emale" +1 +0
ELSE
   IF SEX = "M" THEN
     TYPE "ale" +1 +0
   ELSE
     IF SEX = "U" THEN
       TYPE "nknown" +1 +0
     ELSE
       HELP\
       "Must be M,F, or U.  Press ESC and try again."\
        5 5 12
     ENDIF
   ENDIF
 ENDIF
```

TYPE COMMENT

Syntax: TYPE COMMENT <Column #> <Row #> <Color #>

TYPE COMMENT instructs the TYPE command to use the COMMENT (in a COMMENT LEGAL field) as the string to be typed. If the X offset is +0, the first character in the COMMENT will not be typed. The location of the confirming comment or code will be that given by the column and row (horizontal and vertical) coordinates that accompany the command. If the coordinates are preceded by "+" or "-" signs, they are assumed to be relative to the first character of the entry field.

You do not need "" around comments unless they have internal spaces. Shorter comments will overwrite longer ones appropriately without padding with trailing spaces.

If one of your legal values is given as '.' (dot for missing), then the COMMENT is taken as the one for a missing value and displayed appropriately. Blanks are displayed for the value in F9 code lists.

Here is an example of fields with TYPE COMMENT commands. SEX1 will display a value of M as "MALE" while SEX2 will display "M MALE". In each case, the first M (the one in the field entry) will be in the field color, and the remainder will be in the color used in the TYPE

command:

SEX1
 COMMENT LEGAL
  M MALE
  F FEMALE
  U UNKNOWN
  . MISSING
 END
 IF SEX1 = M THEN
  HIDE PREGNANT
 ELSE
  IF SEX1 = F THEN
   UNHIDE PREGNANT
  ENDIF
 ENDIF
 TYPE COMMENT +0 +0 16
END

SEX2
 COMMENT LEGAL USE SEX1
 TYPE COMMENT +2 +0 16
END

UNHIDE

See HIDE

UNLOAD

Syntax: UNLOAD <File name>.BIN

Removes a binary file from memory, thus conserving memory space for other purposes. If multiple files have been loaded, they can still be UNLOADed individually, if desired. See LOAD.

Example: **UNLOAD ENTFACE.BIN**

UPDATE

Syntax: UPDATE { OFF}

UPDATE causes the screen to be updated or rewritten, putting into effect any color changes or other effects of commands since the last update. Updating occurs automatically when the cursor reaches the bottom of the screen but it is more reliable to place UPDATE in the appropriate location to force updating at the right time. UPDATE OFF prevents updating until the UPDATE command is encountered again.

For an example of the use of UPDATE, see COLOR.

USE

Syntax: Used only within other commands, as follows:

CODES USE <Field name>
LEGAL USE <Field name>
COMMENT LEGAL USE <Field name>
JUMPS USE <Field name>
RANGE USE <Field name>

The USE command allows one copy of identical commands or codes to be used for many fields and thereby conserves memory and reduces the size of the .CHK file.

CODES, LEGAL, COMMENT LEGAL, JUMPS and RANGEs can be listed in a single field but referenced in others with the USE command. Since the list is then loaded into memory only once rather than multiple times, a larger and more complex questionnaire can be accommodated than would otherwise be possible.

In Version 5, codes used by more than one field, such as disease or county codes in a large questionnaire, had to be repeated several times within the same .CHK file, once for each field in which they were used. Since a .CHK file is loaded into Random Access Memory (RAM) when the file is accessed, repeated copies of the same codes will use more RAM, leaving less available for other purposes.

Note that a block of commands for the field specified in the USE command (COUNTY1 in the following example) must appear before the block containing the USE command(e.g.

COUNTY2), and must contain one or more of the commands
mentioned above.

COUNTY1
 CODEFIELD COUNTYCODE
 CODES
  ALLEN   1
   ...
  WYANDOTTE   105
 END
END

COUNTY2
 CODEFIELD COUNTYCOD2
 CODES USE COUNTY1
*   use the same list of codes as in COUNTY1
 END

Use of CODEFIELD in the second field is optional. If it is omitted,the
coding is one-way only:

COUNTY
 CODEFIELD COUNTYCODE
 CODES
  ALLEN   1
   ...
  WYANDOTTE   105343
 END
END

COUNTYCODE
 CodeField REGION
  CODES
    1 3
    ...
    105 4
  END
END

REGION
 NOENTER

*When the cursor is in REGION and its value is changed,
*   COUNTYCODE will not change!
*   If you WANT it to change, uncomment the

```
*      following line:
*    CODEFIELD COUNTYCODE
   END
```

ANALYSIS Commands

Program Operation

<F1> Help	<F2> Commands	<F3> Variables
<F10> QUIT	CLS	CMD
GOTO LOAD	REPORT	RETURNROUTE
RUN	SAVE	
	SET	

Interaction with the User

? ?	ECHO	PICKLIST

File Manipulation

BROWSE	CLOSE	ERASE
GENERATE	READ	RELATE
RENAME	TYPE	UPDATE

Operations Performed or Checked Once for Each Record

BREAK	LIST	SELECT
SORT		WRITE

Operations Performed on Variables Within Each Record

LET	COMBINE	DEFINE
IF		RECODE

   +, -, *, /, DIV , ^, (,), <, >, >=, <=, =, <>, [,], AND, OR, NOT,
      RND(x), RAN(x), LOG(x), LN(x), EXP(x), TRIM(a)

Statistics

FREQ	MATCH	MEANS
OUTPUT FREQ	OUTPUT TABLES	REGRESS
SUMFREQ SUMTABLES	TABLES	

Graphics

BAR	HISTOGRAM	LINE
PIE		SCATTER

Output Format

FOOTER	FORM	HEADER
NEWPAGE		TITLE

Miscellaneous

*	\	COPY
DIR	DOS	IMMEDIATE

System Variables

RECDELETED	RECNUMBER	SYSTEMDATE
SYSTEMTIME		RECVERIFIED

Chapter **36**

ANALYSIS Commands

Running ANALYSIS from the DOS Command Line

ANALYSIS can be run from DOS, from DOS batch files or other programs, or from the EPI6 menu.   Parameters pass information to the program, speed up the process of running ANALYSIS, or invoke special features. The parameters that can be used are:

ANALYSIS <Program file name>
This runs ANALYSIS and automatically begins running the program. The file extension .PGM is assumed unless another is given. If the extension is .REC the file is READ.

ANALYSIS /<.REC file name>
This causes ANALYSIS to READ the named data file automatically, saving time on entry into the program. If the .REC extension is given, the "/" may be omitted.

ANALYSIS /BW
The screen colors will be black and white for monochrome displays. This may be useful for certain displays in which the normal colors do not present a clear image.

ANALYSIS /8
The screen will be set to display 43 lines rather than the normal 25. This works only on EGA, VGA, and other displays capable of displaying 43 lines.

Parameters may also be used from the EPI6 menu. Place the bar cursor on the ANALYSIS item in the menu and press <Ctrl-Enter>. A dialog box will pop up and allow entry of parameters.

Setting Up ANALYSIS To Work with Different Printers

Graphs in ANALYSIS can be sent to Epson-/IBM-compatible dot-matrix printers and to Hewlett Packard-compatible laser printers and plotters. Normal output is for a dot-matrix printer, but others can be selected with the SET PRINTER command. If your printer is not attached to the parallel port on your computer, you should use

the SET PORT command to tell ANALYSIS where to send graphic output. Finally, each type of printer has several MODES that can be used to control image size and resolution. These are set with the SET MODE command. All of these commands are described below in the section on SET.

SET and other commands may be placed in a program file with the name CONFIG.EPI that is run automatically when ANALYSIS is started, giving the effect of making the commands in the file the permanent default settings for ANALYSIS. This is especially recommended for commands used to set up for a particular printer or for preferences in table formatting guided by the SET command. See the section on the SET command below for specific printer settings.

Reference Information on Commands in ANALYSIS

The commands in ANALYSIS are listed below, with detailed descriptions of their formats and the results they produce. Any of the commands can be typed in response to the EPI6> prompt or included in a program file created in EPED or most other word processors. The section on "Recalling, Saving, and Loading ANALYSIS Commands" in Chapter 11 gives other details on operating ANALYSIS from the keyboard.

A command must have fewer than 225 characters. When typing a command on the screen, *do not press <ENTER>* until the entire command has been entered, even though the screen automatically puts characters in excess of 80 on another line.

Commands in program files created in EPED also may be longer than 80 characters. In EPED, you may divide a long command into several lines, placing a backslash ("\") at the end of each line that is to be continued. Do not use backslashes for this purpose when typing commands directly in ANALYSIS, however. Within a program file, the following would be executed as a single command:

**LIST AGE ILL BAKEDHAM CABBAGESAL BROWNBREAD **
WATER CHOCOLATE VANILLA

A comment is preceded by an asterisk. Comments are ignored by ANALYSIS, but are for the benefit of the programmer or other readers of the printed program. Thus

*This is a comment. It is a message to the programmer and has no other
* effects.

Labels mark a line in a program so that the GOTO command can be used to cause

program execution to skip to that line. Label lines begin with a colon followed by up to 10 letters for the label name. The line may not contain other commands. The following line is a label:

:FINISHUP

If a command is followed by **/NOECHO**, the results will not be echoed to the screen, although they will be sent to the printer or to a file if this has been set up by a previous ROUTE command.

In many commands, you can use the value of a variable within an expression preceded by an "@" sign. If FILENAME is a variable with the current value of "HEPATITIS.REC", then ROUTE @FILENAME is the equivalent of ROUTE HEPATITIS.REC, for example. You might also set a variable COREFIELDS equal to "AGE SEX RACE COUNTY DISEASE" and then save typing by giving the command LIST @COREFIELDS rather than LIST AGE SEX RACE COUNTY DISEASE. Different files may be read with READ GEPI@YEAR. The @ sign is not used in assignment statements like SUM = SUM + 1, or in tests of truth such as IF AGE > 30 THEN GROUP = "ADULT".

In any command requiring a list of variables, such as LIST, FREQ, WRITE, TABLES, or USES, you may use the word TO to indicate a list of variables in the order in which they appear in the questionnaire and in the Variables display (<F3>). LIST AGE TO COUNTY will list the variables AGE, COUNTY, and all those in between, perhaps AGE SEX RACE OCCUPATION COUNTY.

Program Commands in ANALYSIS

The available commands are listed below in alphabetical order.

Assignment or "=" -- See LET

BAR

Syntax:
BAR <Variable name> {/Y=<min>-<max>} {/SUM = <Field name>}
     {/LABEL=<Text>}

The BAR command will make a bar chart of the variable named. Bar charts are useful for displaying the counts for values of both ordered and non ordered variables. Normally, vertical bars represent counts of records

with different values of the variable. If the SUM is specified, the bars represent sums of the variable given in the SUM parameter.
Bar charts differ from histograms in having the bars separated by spaces and in omitting counts for values in which the count is zero.

The maximum vertical axis value can be increased by adding the parameter /Y=min-max, in which min is zero and max can be any number larger than the number automatically assigned by the program. This adjustment is useful if you are making a series of graphs and want the vertical axis values to be comparable. There should be a space before the "/", but no spaces in "Y=min-max".

Normally the label on the vertical axis is "Count." If the /LABEL=<Text> parameter is given, the label used is the one given as Text.

Titles may be added to the graph using the TITLE command. If ROUTE PRINTER is in effect, the graph will be printed.

Examples:
BAR ROLLS
BAR ROLLS /Y=0-75
BAR ROLLS /LABEL=Cases

See also: HISTOGRAM, LINE, PIE, SCATTER, TITLE

BREAK

Syntax: BREAK <variable name> {<variable name> ...etc.}

BREAK performs the command contained in BREAKCMD whenever the value of the <variable name(s)> changes during a LIST command. The records must first be sorted on the variable(s) specified. BREAKCMD is normally set to skip to a new page if output is ROUTEd to the printer or to a file. BREAKCMD may be changed to any appropriate Epi Info command by using SET BREAKCMD = (Command). See SET for further details.

Example:
READ OSWEGO
SORT AGE
ROUTE PRINTER
SET BREAKCMD = TYPE "\nAge @AGE \n"
BREAK AGE
LIST

This produces a list of records in order by age, with a heading over each age group. If you omit SET BREAKCMD, each group will be printed on a separate page.

BROWSE (<F4>)

Syntax: BROWSE {<Field name list>}

The BROWSE command displays the current data file(s) and any RELATEd files, in spreadsheet format on the screen. You can use the arrow keys to move around and see the entire data set. In BROWSE, the following functions are available:

<Home><End> These keys move to the first or last record.

<F4> Full Screen. Pressing this key alternately displays the records in the data set in questionnaire format just as they would appear in the ENTER program, or in spreadsheet format as when BROWSE is first activated. The more familiar question format provides orientation about the meaning of the questions. The Prev Rec and Next Rec familiar from the ENTER program are available in the questionnaire mode. To return from the questionnaire to the spreadsheet, press <F4> again or <Esc>.

<F10> Done. This terminates the BROWSE session and returns to the EPI6> prompt, ready for the next command.

Examples:
BROWSE
BROWSE AGE SEX RACE

CLOSE

Syntax: CLOSE {<File name>}

CLOSE closes the named file. If the file has been related, it is equivalent to removing the relation. It can be used to close a file before the ERASE command is used to erase it, or to close a related file so that another RELATE can be performed. If no file name is given, all open files will be closed.

Example:
CLOSE TEMP.REC
ERASE TEMP.REC

CLS

Syntax: CLS

CLS clears the output window.

Example: **CLS**

CMD

The CMD command can be used to define a block of commands that will act as a unit, similar to a subroutine or procedure in other languages.

Syntax:

```
CMD <Command name>
    <Block of Commands>
END
```

Whenever <Command name> appears elsewhere in the .PGM file, the block of commands is executed. To have the block executed once for each record in the file, you might place the <Command name> in a statement such as IF 1=1 THEN <Command name>. CMDs are intended to be called from an IF statement. If you want a block of commands to be executed only once, it is best to place a label at the beginning of the commands and use the GOTO and RETURN commands to create the effect of a subroutine.

Example:
ROUTE PRINTER
CMD DOLABEL
**   TYPE "@NAME"**
**   TYPE "@ADDR1"**
**   TYPE "@CITY , @STATE @ZIP"**
END
IF 1=1 THEN DOLABEL

The last line will print a label for each record in the file. For a more capable label program, see LABELS.PGM

The CMD command can also be used to run an external program from within ANALYSIS and to return the results to ANALYSIS, using the following syntax:

CMD <command name> { FILE= .EXE file name} { TABLE} { NOPARAM}
{ MAXFIELDS=[max. number of fields]} { NUMERIC=[numeric field set]}

This use of CMD is described in Chapter 39.

Example:
**CMD MATCHED FILE=MATCHED.EXE TABLE MAXFIELDS=3 **
     NUMERIC=1
*Sets up ANALYSIS and the external file MATCHED.EXE so that the *following
commands will work:
READ RELY
MATCHED ID CASE RELY

COMBINE

Syntax:

COMBINE { NAME=<NameVar>} VALUE=<ValueVar> <Variable>
<Variable> <Variable> etc.

COMBINE { NAME=<NameVar>} VALUE=<ValueVar> <Variable>
TO <Variable>

COMBINE causes the contents of several fields to be treated as though
they were in the same field. Suppose, for example, that you have several
fields in a case questionnaire as follows:

> Diagnosis1 _____
> Diagnosis2 _____
> Diagnosis3 _____
> Diagnosis4 _____

You would like to know how many times "pneumonia" appears as a
diagnosis among all the cases. Since it may appear in any of the four
fields, you would have to write an IF statement to count pneumonia if it
appears in any of the four fields, or even a program to create new records
for each diagnosis rather than for each "case." In database parlance, this is
called "normalizing" the database. It may be thought of in simple terms as
"putting all the data you want to count or sum into the same field."

COMBINE automatically solves the problem of data spread across
multiple fields. When you give the commands

DEFINE DIAGNOSIS _____

COMBINE VALUE=DIAGNOSIS DIAGNOSIS1 TO DIAGNOSIS4

subsequent FREQ and TABLES commands that include DIAGNOSIS will simulate a file with four times as many records as the original file. There will be one case record for each of the four diagnosis fields. In the first DIAGNOSIS will contain the data for DIAGNOSIS1, in the second it will contain the data for DIAGNOSIS2, etc. We now have a simulated file based on the number of DIAGNOSES (including blanks). The DEFINEd field, DIAGNOSIS, must be of the correct type and size to contain values from any of the other listed fields.

A FREQ DIAGNOSIS command might produce a frequency as follows:

```
        Amebiasis          5
        Pneumonia         12
        Xerophthalmia      1
```

A command such as TABLES AGE DIAGNOSIS will produce a table in the following format, where XX represents an individual cell:

```
              DIAGNOSIS
          XX   XX   XX   XX
   AGE    XX   XX   XX   XX
          XX   XX   XX   XX
          XX   XX   XX   XX
                            TOTAL
```

FIELDS DIAGNOSIS1..DIAGNOSIS4 have been COMBINEd. The actual number of records differs from that shown above.

TOTAL will represent the number of diagnoses if SET IGNOREMISSING is ON and four times the number of records (because there are four diagnostic fields) if SET IGNOREMISSING is OFF.

Only one COMBINE command can be active at one time. Giving a second COMBINE command will turn off any previous COMBINE. The word COMBINE alone without any field names will cancel a previous COMBINE function.

After giving the COMBINE command WRITE RECFILE will produce a file with a record for each diagnosis (IGNOREMISSING ON) or each field in the COMBINE statement (IGNOREMISSING OFF). In our example, with IGNOREMISSING OFF, four times as many records would be produced as were in the original file. Each would have a copy of all

other data fields (such as NAME, SEX, ADDRESS) containing the same data as the original record that contained the fields DIAGNOSIS1 through DIAGNOSIS4.

Sometimes it is important to include the name of the COMBINEd fields in the new "records" so that a table may be constructed.

In a foodborne outbreak, the following fields might appear:

```
Chicken  <Y>
Beef     <Y>
Salad    <Y>
```

You would like to make a graph or do a frequency of exposure to each type of food. Instead of the three separate fields, you would like one field called FOOD containing the names of the foods eaten.
Instead of the records:

```
NAME        CHICKEN  BEEF    SALAD
Jones          Y        N       Y
Smith          N        Y       Y
```

You would like to have:

```
 NAME          FOOD     EATEN
Jones         Chicken       Y
Jones         Beef          N
Jones         Salad         Y
Smith         Chicken       N
Smith         Beef          Y
Smith         Salad         Y
```

so that you can do:

TABLES FOOD EATEN

and get the following table:

```
                    EATEN
        FOOD        +    -
         Chicken    1    1    2
         Beef       1    1    2
         Salad      2    0    2
```

4   2   6

If PERCENTS is SET to ON, this table will give attack rates for the various foods. If you wanted to graph the number of exposures to various foods, you could use:

SELECT EATEN = "Y"
BAR FOOD

to place the food names along the horizontal axis and the number of eaters on the vertical axis of a bar graph.

Suppose that a questionnaire representing a district of a country has the following fields:

Number of cases   MALARIA ####   DIARRHEA ####   NEPHRITIS ####

Summary data like this can best be COMBINEd as follows:

DEFINE COUNT #####
DEFINE DISEASE _____
**COMBINE NAME=DISEASE VALUE=COUNT MALARIA **
  DIARRHEA NEPHRITIS

SUMTABLES COUNT DISEASE

This would produce a table of disease counts for the country as a whole.

COMMANDS (<F2>)

Syntax: None. Interactive only.

Pressing the <F2> key presents a menu of commands. A command can be chosen by moving the cursor bar and then pressing <Enter>. The command will automatically be inserted before any variable names that have already been chosen.

COPY

Syntax: COPY <Source file> <Destination file>

Copies a file like the DOS command COPY, using the same syntax.

Examples:

COPY A:ALPHA.BAT   B:ALPHA.BAT
COPY \EPI6\GAMMA.BAT \DOS\GAMMA.BAT

DEFINE

Syntax:

DEFINE <Variable name> _____ { GLOBAL} { CUMULATIVE} <Value>
                        ###
                        ####.##
                        <mm/dd/yy>
                        <dd/mm/yy>

DEFINE allows creation of new variables for use in ANALYSIS. The
variable type and length are given, as they are in questionnaires, by
underline characters, number symbols, or the American or European date
formats. The length of the underline or number fields can vary as it does
for questionnaires. If no decimal point is specified for a numeric field, *the
results will be rounded off to the nearest whole integer*. The WIDTH of
fields containing decimals should allow for the decimal point as one
character.

Variable *names* in the DEFINE statement must be 10 or fewer characters,
but the *width* of a field can be as large as 80 characters. Variable names
may not begin with a number or contain spaced, hyphens or other
punctuation, although the underline character "_" is allowed.

Numeric variables are integers if no decimal is specified and there are
fewer than five digits; otherwise they are real or "floating point" numbers.

Defined variables begin with values set to **missing**, unlike variables in
Version 3. A variable may be initialized to a desired value by placing a
space and the desired value after the definition.

There are three types of defined variables--*standard*, CUMULATIVE, and GLOBAL. The difference is in the time when the variable is reset to the missing or "initial" value.

Standard variables are reset after each record is processed in a procedure such as FREQ, LIST, or TABLES. They are useful for calculations that are to be performed within each record.

CUMULATIVE variables are reset to the missing or "initial" value whenever a new command that causes the file to be READ is given. They are useful for maintaining sums or other values that are cumulative from record to record but that should begin again when a FREQ, LIST, or TABLE is done.

Both standard and CUMULATIVE defined variables are reset by the READ statement and are therefore not useful for maintaining values that will be used during a series of programs with different data files. GLOBAL variables are provided for this purpose. They are not reset by the READ statement, and only a QUIT or an explicit assignment will reset their value. If the user enters the week of processing desired, this might be kept in a GLOBAL variable throughout a complex series of programs so that the user doesn't have to be asked again to enter the week number.

Values for dates or text (STRING) fields must be surrounded by quotation marks.

Examples:

DEFINE RACENAME _____
(Could be used in a RECODE statement to provide labels for
a numerically coded RACE variable)

DEFINE COUNTYCODE ###

DEFINE TOTALCOST #####.## CUMULATIVE 0
(Could be used to maintain a running total by adding cost in
each record to TOTALCOST. The 0 is redundant, since all cumulative
variables are initialized to 0, but could be replaced by another number,
such as 100.)

DEFINE FOLLOWUP <mm/dd/yy> (American date)

DEFINE FOLLOWUP <dd/mm/yy> GLOBAL "31/03/95"
(European date, global variable, preset to 31/3/95)

DEFINE HEIGHT NUMBER ###.#

DIR

Syntax: DIR <file name or wildcard entries as for DIR command in DOS>

DIR displays the names of files under the category requested.

Examples:
DIR A:
DIR B:*.REC

DOS (<F9>)

Syntax: DOS { <DOS command or program to run>}

The DOS command or the <F9> key provides access to DOS from within
ANALYSIS. Pressing <F9> or typing DOS will bring up the DOS
prompt just as though you had left ANALYSIS. DOS commands can then
be issued to COPY files, do DIRectories, or run other programs. To return
to ANALYSIS, type EXIT, and you will instantly be back in ANALYSIS
without having lost any of the commands issued up to that point. If a
valid DOS command follows the word DOS, the command will be
executed and the program will return automatically to ANALYSIS.

Since ANALYSIS is still present in memory when DOS operations occur,
you may not be able to run huge programs this way. The amount of
memory available for other programs is approximately that displayed at
the top of the ANALYSIS screen.

Examples:
DOS
Exits to the DOS prompt and allows you to give DOS commands. To
return to ANALYSIS, you type the word "EXIT".

DOS FORMAT A:
Formats a disk and returns to ANALYSIS.

DOS ENTER OSWEGO
Runs the program called ENTER with OSWEGO.REC

While in the DOS command, do not attempt to load TSR (Terminate and Stay Resident) programs like Sidekick. You may access such programs at any time however, if they have been loaded before ANALYSIS is run.

ECHO

Syntax: ECHO <text>

ECHO writes <text> on the screen, but not to the printer or to a file. This provides a means for telling the user what a batch file is doing, for issuing instructions, or for getting input. See "?   ?" under **Miscellaneous** at the end of the reference section for a method of getting input.

Example: **ECHO Sorting the database. Please wait a moment.**

ERASE

Syntax: ERASE <File name>

ERASE will erase the file <file name> from the current directory, or, if a directory is included as part of the file name, from any directory. It will not erase a currently active .REC file; the file must first be CLOSEd.

Example: **ERASE RESULTS.TXT**

FOOTER

Syntax: FOOTER < Footer number 1-3> { "<footer text>"}

Footers are lines of text that will appear at the bottom of every page of ANALYSIS results. Up to three lines of footers may be specified. The FOOTER command sets the footer designated to the <text> specified. The footers will be printed on each page until another FOOTER command changes the content or a FOOTER <number> without a succeeding phrase erases the text. The quotation marks are optional.

To avoid printing page numbers on output sent to the printer, specify a blank footer 2, such as **FOOTER 2 "   ".**

Examples:
FOOTER 1 "\C-----------------------------"   (a centered line)
FOOTER 2   (removes footer 2)

See also TYPE for additional commands that may be inserted with "\".

FORM

Syntax: FORM {/C}

The FORM command provides a way of printing forms containing data from ANALYSIS. This command itself causes a questionnaire to be printed that contains data from the current record. It may be combined with the IF, SELECT, and PROCESS commands to print any or all of the records in the file in questionnaire format.

Normally each form will begin on a new page. If you do not wish to issue the equivalent of NEWPAGE at the end of each questionnaire, the /C parameter (for Continue) will cause only one line to be skipped before printing the next questionnaire.

Examples:
READ OSWEGO
ROUTE OSWEGO.TXT
FORM

This sends a blank copy of the questionnaire to the file called OSWEGO.TXT, where it may be edited for use as a paper form.

READ OSWEGO
ROUTE PRINTER
IF AGE < 10 THEN FORM
PROCESS

This prints a copy of the questionnaire containing data for each person under age 10 in the file. Note that the PROCESS command is necessary to cause the program to pass through the file.

Various combinations of the IF command can be used, such as IF 1=1 THEN FORM. Since 1 is always equal to 1, a copy of each record in the file will result. Similar results can be achieved with IF AGE > 0 ..., or IF AGE < 900....

FREQ

Syntax:

FREQ <Variable name{s} or "*"> {/SUM=<Variable name>} {/C}

FREQ produces a table from the file(s) specified in the last READ statement, showing how many records have each value of the variable. Records may be included or excluded from the count by SELECT statements. Those marked as deleted in ENTER will be handled according to the current setting for SET PROCESS. If more than one variable name is given, FREQ will make a separate table for each variable.

If a SUM is specified, the results represent sums of the values of the variable given in the SUM parameter.

If "/C" is specified and the variable is numeric, confidence limits for the binomial proportions are produced. Further details on the confidence limits are given in Chapter 32.

FREQ * will make a table for each variable in the current questionnaire. It is often used to begin analysis of a new data set. To do frequencies of all variables except a few, use

FREQ * NOT

followed by the names of the variables to be excluded.

Examples:
FREQ age
FREQ AGE SEX RACE   (Does separate frequencies for each of the variables)
FREQ *   (Does frequencies for all variables)
FREQ * NOT IDNUM   (Does frequencies of all variables except IDNUM)

<F10>

See QUIT. <F10> and QUIT are synonyms for "Leave ANALYSIS and return to the menu, or to DOS."

GENERATE

Syntax: GENERATE <Number of records>

The GENERATE command combined with the RND, RAN, and
RECODE functions will produce a data set of any size up to 32,000
records tailored to your specifications, but derived from random numbers.
Such files are useful for testing, demonstrations, or training. We assume
that Epi Info users will not risk their scientific reputations or the health of
others by portraying such data as real results. To avoid such impressions,
you should carefully label and use these files only in situations where their
origin is clear. Please report any misuse to one of the authors.

To produce a generated data set, first create a questionnaire to describe the
variables, and then use ENTER to produce an empty .REC file. Then
make an ANALYSIS program file to generate the data using RECODE
and the random functions, followed by the command GENERATE and the
number of records desired.

The RECODE command has special functions that are used with
GENERATE as illustrated below. A more extensive example is given in
Chapter 27 and in the program ARTIFACT.PGM on the Epi Info disks.

Examples:

RECODE RND TO SEX 70%=M 25%=F ELSE= .

means assign "M" and "F" to the field SEX using random numbers so that
approximately 70% of the values are "M" and 25% are "F". The
remainder will be missing values.

To assign a random integer from 0 to 65 to AGE, use:

AGE = RAN (65)

And to create 100 such records:

GENERATE 100

GOTO

Syntax: GOTO <Label name>

GOTO causes the program to resume execution at the named label.
Labels are indicated by a colon in the first column, as in:

:DOQUESTION

The label may be either before or after the GOTO statement, thus allowing backward or forward jumps or loops using IF statements to control the number of times the loop is executed. Subroutines similar to those in BASIC can be called with GOTO in combination with RETURN at the end of the subroutine.

Examples:
GOTO DOQUESTION

DEFINE COUNTER CUMULATIVE
:ONCEMORE
TYPE "Hello"
IMMEDIATE LET COUNTER = COUNTER + 1
IMMEDIATE IF COUNTER < 3 THEN GOTO ONCEMORE

These statements will type "Hello" three times using a single TYPE command. See also RETURN. To call a block of commands from within an IF statement (not IMMEDIATE IF), see CMD.

HEADER

Syntax: HEADER <header number 1-3> { "<header text>"}

Headers are lines of text which will appear at the top of every page of ANALYSIS results. Up to three lines of headers may be specified. The HEADER command sets the header line <header number> to <header text>. The quotation marks are optional. The headers will be printed on each page until another HEADER command changes the content or a HEADER <number> without a succeeding phrase erases the text.

Examples:
HEADER 1 \cGeorgia Department of Health --Weekly Report
  (Centers the heading on each page)
HEADER 1   (Removes header 1)

HISTOGRAM

Syntax:

HISTOGRAM <Variable name> {/Y=<min>-<max>} {/SUM = <Field name>}
  {/LABEL=<Text>} (All on one line or with continuation symbol, \)

The HISTOGRAM command will make a histogram of the variable named. Normally, vertical bars represent counts of records with different values of the variable. If the SUM is specified, bars represent sums of the variable given in the SUM parameter.

Histograms are useful for displaying the numbers of cases occurring over a period of time, and this type of graph is often called an epidemic curve. The dates may be single or grouped, but if grouped, should be in groups containing the same numbers of years, days, etc.

Histograms differ from bar graphs in having the bars contiguous--not separated by spaces-- and in displaying counts even for values in which the count is zero.

The maximum vertical axis value can be increased by adding the parameter /Y=min-max, in which min is zero and max can be any number larger than the number automatically assigned by the program. This adjustment is useful if you are making a series of graphs and want the vertical axis values to be comparable. There should be a space before the "/", but no spaces in "Y=min-max".

Normally the label on the vertical axis is "Count." If the /LABEL=<Text> parameter is given, the label used is the one given as <Text>.

Titles may be added to the graph using the TITLE command. If ROUTE PRINTER is in effect, the graph will be printed.

Examples:
HISTOGRAM ONSETDATE
HISTOGRAM ONSETDATE /Y=0-75

See also: BAR, LINE, PIE, SCATTER, TITLE

IF

Syntax: IF <Condition(s)> THEN <Consequence1> { ELSE <Consequence2>}

IF defines conditions and one or more consequences which result **for every record** in which the conditions are met. An alternative consequence can be given after the ELSE statement, to be realized if the first set of conditions is not true. The consequence may be an assignment

of value, similar to a LET statement (but without the LET). It may not be another IF statement.

Examples:

IF AGE > 15 THEN GROUP = "ADULT" ELSE GROUP ="CHILD"
  (Note that "LET" is not included in the statement)

DEFINE CASE _
DEFINE MAJORSYMP #
MAJORSYMP = 0
IF DIARRHEA = "Y" THEN MAJORSYMP = MAJORSYMP + 1
IF FEVER = "Y" THEN MAJORSYMP = MAJORSYMP + 1
IF VOMITING = "Y" THEN MAJORSYMP = MAJORSYMP + 1
IF MAJORSYMP >= 2 THEN CASE = "Y" ELSE CASE = "N"
*(If two or more symptoms are present, the record is classified as a case.)

When the IF conditions do not describe all possibilities, standard defined variables will be reset after each record to the "missing" value. It is important to cover all the conditions in your IF statements that you want to result in nonmissing values.

Sometimes it is important to have an "ELSE" condition that applies to all those not covered by other IF clauses. This effect can be achieved by setting the variable initially to something other than missing, as in:

DEFINE ILL _
LET ILL = "N"
IF VOMITING = "Y" THEN ILL = "Y"
IF DIARRHEA = "Y" THEN ILL = "Y"
IF FEVER = "Y" THEN ILL = "Y"
*Sets Ill = "Y" only if one or more symptoms are present.
The same result would be achieved by:
**IF (DIARRHEA = "Y") OR (VOMITING = "Y") OR **
    (FEVER = "Y") THEN ILL = "Y" ELSE ILL = "N"
(Note that "\" is used to indicate continuation of a long line in a program file but not on the screen.)

Parentheses should be placed around each expression in a series of conditions so that the meaning is unambiguous, since sometimes the order in which conditions are evaluated and which elements are connected by OR or AND can be interpreted in more than one way.

See OPERATORS for help in constructing expressions.

IMMEDIATE

Syntax: IMMEDIATE <Command>

where <Command> may be a LET (assignment), IF, or WRITE command.

The IMMEDIATE command is used with commands that are normally performed for each record in the file. It causes the command to be executed only once at the time it appears in a program file or is typed on the screen, rather than once per record.

The comparable, but less general command, IMMEDIF, from Version 3 of Epi Info, is still valid to allow previous programs to run, but it should be replaced by IMMEDIATE IF in new programs.

Examples:
IMMEDIATE IF DISTRICT = 5 THEN TYPE FIVE.TXT
IMMEDIATE IF AVERAGE >= 75 THEN TYPE "PASS" ELSE
TYPE "FAIL"
IMMEDIATE SUM = 100

Sets SUM, presumably a cumulative variable, to 100. Without the IMMEDIATE command, SUM=100 would set the variable to 100 once for each record read from the file rather than once at the time the command is issued.

See OPERATORS for help in constructing expressions.

LET (Assignment of Variable Value)

Syntax: {LET} <variable> = <expression>
  The word LET is optional.

LET assigns the value of an expression to a variable. If the variable has not been defined, ANALYSIS will define a variable automatically, but will make the variable numeric. It is better programming practice to define your variables explicitly using the DEFINE command.

The value is calculated each time a record that meets the conditions of SELECT is read from the data file. Any legal expression can be used which combines variables or specific values and operators like +, -, *, /, =, <,>,AND, OR, and NOT. See OPERATORS for the complete list.

A portion of a text variable or date, such as two characters beginning with the seventh character (the year in a date field), can be used in a LET statement. See SUBSTRINGS for a complete explanation.

Examples:
LET Code = 10
CountyCode = 10

Specific values ("literals") can be included. Those that represent strings (text) or dates must be enclosed in quotation marks, for example:

County = "Yellow Medicine"
LET DURATION = "01/01/88" - DATEONSET
*(Duration will be in days.)
LET ILL = "N"
GROUP = 1
*(If Group is a numeric variable)
GROUP = "1"
*(If Group is a text variable)

To accumulate the sum of a variable, COST, first define a CUMULATIVE variable:

DEFINE SUM ####.## CUMULATIVE
SUM = SUM + COST

For methods of dealing with a part of a text or date value, see SUBSTRINGS.

LINE

Syntax:
LINE <Variable name> {<Variable name>} {SUM = <Field name>}
      {/LABEL=<Text>}

The LINE command will make a line chart of the variable named first. Normally, points on the line represent counts of records with different values of the variable. If SUM is specified, points represent sums of the variable given in the SUM parameter.

Line charts are useful for displaying the counts of ordered data such as age, blood pressure, or a series of dates. Grouping the data often makes the line smoother and easier to understand.

More than one line may be graphed by giving the name of an additional stratifying variable. The command LINE AGE ILL will produce one line for the ages of the ILL and one line for the ages of those not ILL, if ILL is a Yes/No variable.

Titles may be added to the graph using the TITLE command. If ROUTE PRINTER is in effect, the graph will be printed. Normally the label on the vertical axis is "Count." If the /LABEL=<Text> parameter is given, the label used is the one given as <Text>.

Examples:
LINE AGEGROUP
LINE AGE ILL

See also: BAR, HISTOGRAM, PIE, SCATTER, TITLE

LIST

Syntax: LIST { <Variable 1> <Variable 2> ... <Variable n>}
          LIST { NOT <Variable 1> <Variable 2>...etc.}
          LIST * { NOT <Variable 1> <Variable 2>...etc.}

LIST does a line listing of the current data set. If no variables are specified, it lists as many of the variables as possible across one page.   If variable names are given, LIST will list only these variables. LIST * will list all variables of all active records, using several pages across to accommodate all the variables if necessary.

Adding NOT <Variable list> will exclude all the named variables from a LIST or LIST *. It is convenient if you wish to leave out names or addresses, for example. Record numbers are listed automatically as the first item on each line, but this feature can be turned off by giving the command SET LISTREC = OFF.

If the data set has been sorted with the SORT command, the records are listed in sorted order. Otherwise, they are listed in their physical order in the file.

If your printer can accommodate more than 80-character columns, use the SET command to obtain a wider page. SET PAGE = 66, 150, for example, will print 66-line pages 150 characters wide and will allow many more variables to be listed across the page.

Examples:

LIST
LIST name age sex race
LIST *
LIST * NOT NAME SSN ADDRESSCITY STATE

LOAD

Syntax: LOAD <File name>

Loads a program file into memory, ready for use, but does not RUN the
program. The commands in the program can be examined with the up-
and down-arrow keys and then run individually if desired. See SAVE and
RUN.

Example: **LOAD TEST.PGM**

MATCH

Syntax:
MATCH <ID Variable> <OutcomeVariable> <Exposure Variable>

MATCH performs a matched analysis to produce the crude odds ratio, the
matched odds ratio, the Robins, Greenland, Breslow 95% confidence
limits for the matched odds ratio, and the Mantel-Haenszel summary chi
square and p value. The summary chi square is equivalent to the
McNemar chi square if the data are arranged as matched pairs.

This test may be performed only if records in a file are organized so that
matched sets have the same ID variable value. Thus, in a case-control
study, a case must have the same ID number as the control(s) with which
it is matched. The ID variable value then automatically causes the match
to occur during the analysis, even if the number of controls per case
varies.

Variables after the ID variable are listed in the same order as in a
TABLES command. Tables will be produced internally in the computer
but will not be printed. If you prefer to see the tables, the command
TABLES <Variable1> <Variable2> <ID Variable> will produce tables
and give the same result as the MATCH command.

Example:
READ MATCH
MATCH IDNUM ILL RELY

(The data contained in the file MATCH.REC on the Epi Info disks is a case-control study of women with and without toxic shock syndrome (ILL=Y/N) matched for various factors indicated by IDNUM, and exploring associations with a history of Rely brand tampons.)

MEANS

Syntax:
MEANS <Numeric variable to be analyzed> <Variable for Grouping> {/N}

The MEANS command requires two items of information--the name of a numeric variable containing data to be analyzed and the name of the variable which indicates how groups will be distinguished. Normally MEANS produces a table which displays the data and then performs appropriate statistics. If the /N parameter is included in the command, no table is produced, but the other statistics are displayed.

The maximum number of separate values (e.g., different dates) in a MEANS table is 250. If more are desired, try grouping the values using recode before doing MEANS.

Depending on the size of the table and whether the data are normally distributed, MEANS provides the equivalent of:

For Normally Distributed Data (parametric tests)
ANOVA (for two or more samples)
Student's t test (for two samples)

For Data Not Normally Distributed (non parametric tests)
Kruskal-Wallis one-way analysis of variance (for two or more samples)
Mann-Whitney U Test = Wilcoxon Rank Sum Test (for two samples)

Further details are given in Chapter 9 and in Chapter 32 on Statistics.

Example : **MEANS AGE ILL**

will compare ages for those in whom ILL ="Y" with those for whom ILL = "N". If there are more than two groups in the variable ILL, such as 1, 2, and 3, or if SET IGNORE = OFF has been invoked, so that "Y", "N", and blank are all considered groups, MEANS will do a comparison of all the groups and the answer will indicate whether or not the groups differ from each other (but not in what pattern).

NEWPAGE

Syntax: NEWPAGE

Inserts enough blank lines in the output to the printer to skip to the top of the next full page. NEWPAGE is the default setting for break, as though the command SET BREAKCMD = NEWPAGE had been given.

Example: **NEWPAGE**

OPERATORS

ANALYSIS provides the following operators for use in arithmetic and logical expressions.

Operator   Function

+	Addition
-	Subtraction
*	Multiplication

/	Division
DIV	Division, with integer results. A remainder of less than 1.0 is discarded.
^	Exponentiation. Raises a number to the power indicated. Roots can be taken by giving fractional values, such as 0.5 for the square root.
(	Expression grouping
)	Expression grouping
>	Greater than
<	Less than
>=	Greater than or equal to
<=	Less than or equal to
=	Equal to
<>	Not equal to
[	Beginning of substring information. See the LET command.
]	End of substring information
RND (x)	Produces a random real number between 0 and the value of x.
RAN (x)	Produces a random INTEGER between 0 and the value of x.
LOG (x)	Produces the logarithm to the base 10 of x.
LN (x)	Produces the natural logarithm (base 2.7182818 = e) of x.
EXP (x)	Raises the base of the natural logarithms, e, to the power x.
TRIM (a)	Strips trailing blanks from a text field.
TRUNC(x)	Removes the fractional part of a real number, producing an integer (rounds toward zero)
ROUND (x)	Rounds a real number to the nearest integer, rounding away

	from zero if the fraction is 0.5 or more
AND	Logical and. If *all* things being ANDed are true, then the expression is true.
OR	Logical or. If one or more of the conditions being ORed is true, then the expression is true.
NOT	Logical not. If the condition is false, then the expression is true. If the condition is true then the expression is false.

An "expression" is a combination of variables, literal values, and operators which can be evaluated to produce a single result. Expressions can be quite complex. Both permanent and defined variables can be used in the expression. When the expression is evaluated, the proper value for the variable will be substituted in the expression by ANALYSIS. For LET, IF, and SELECT commands, this will be done as each record is processed; the IMMEDIATE command causes the expression to be evaluated only once.

Examples: (IF, THEN, and LET are not part of the *expressions.*)

IF (CHEESE = "Y") AND NOT (PEPPERONI = "Y") THEN VEGI = "Y"

IF (AGE <=30) AND (AGE > 17) THEN GROUP = "Young Adult"

For convenience, the following form of the same expression is also allowed:
IF AGE < 30 AND >17 THEN GROUP = "Young Adult"

LET DAYS = WEEKS * 7
LET YEARS = DAYS / 365.25
LET VSQUARED = V^2
LET SQUAREROOT = V^0.5
LET EXPECTED = RATE * POP * HOURS

If mathematical operations on dates are performed, the result will be in days. Thus

LET DURATION = "01/01/88" - ONSETDATE

would automatically set DURATION to the number of days between ONSETDATE and the beginning of 1988. Note that the literal date is enclosed in quotation marks. This is essential, since dates are a special form of string variable.

If a date variable is in the short form <mm/dd>, the year is taken from the system date in DOS. It is therefore assumed to be the current year if the DOS date is correct. If both dates being subtracted are lacking a year, the same year will be assumed, and an incorrect system date will not cause any problem. The DOS date can be reset from the DOS prompt by typing DATE, or by giving the command DOS DATE from ANALYSIS.

Complicated expressions after the exponent sign (^) are not accepted. If an expression does not seem to work properly, break it into simpler components. For example **Var2=2.67^(1.96/Chi)** does not give the desired result, but the two simpler expressions that follow work correctly:

Var1 = 1.96 / Chi
Var2 = 2.67 ^ Var1

OUTPUT

Syntax: OUTPUT TABLES|FREQ|SUMTABLES|SUMFREQ <Variable name(s)>
in which "|" means "or" (choose one).

The OUTPUT command, when used with the FREQ, TABLES, SUMFREQ, or SUMTABLES commands, sends the table or frequency distribution to the destination specified in a previous ROUTE command. Thus, the commands:

ROUTE SUMMARY.REC
OUTPUT TABLES DISEASE COUNTY

will create a file called SUMMARY.REC, which will contain records summarizing the number of cases of DISEASE by COUNTY. The file will be a standard Epi Info .REC file and will have the following variables:

DISEASE COUNTY COUNT

The variable COUNT was added by the program and will contain the contents of the cell in the table which would have been produced by

TABLES DISEASE COUNTY

The OUTPUT command can be used to store the periodic results as summary records in a file. The summary file can then be processed further in ANALYSIS to produce grand totals without reprocessing the

entire original data file. The SUMFREQ and SUMTABLES commands
are designed to process files of data created by the OUTPUT command.

PICKLIST

Syntax:
PICKLIST { <Column #> <Row #> }   { KEEP}
  "<Text>" <Command>
  "<Text>" <Command>
  "<Text>" <Command>
END

The PICKLIST command causes a list of items to be displayed on the
screen so that the user may move the cursor to the desired item and make
a choice by pressing <Enter>.

If desired, the row and column of the upper left corner of the list may be
given; otherwise the picklist will be displayed in the center of the screen.

Individual choices for the list are given on successive lines, with the
choice word or phrase in quotation marks, followed by a space, and the
command to be executed when that choice is selected.

The PICKLIST command may be included in .PGM files to construct
menus or other choice boxes, but may not be executed from the keyboard,
since ANALYSIS only accepts one line at a time from the keyboard.

If the word KEEP is added to the PICKLIST line, multiple commands can
be executed until the user chooses END, otherwise, only one will be
executed before passing to the next command in the .PGM file.

Example:
PICKLIST 20 5
  "Do Table" TABLES VANILLA ILL
  "Do Frequency" FREQ AGE
  "Turn Page" NEWPAGE
END

PIE

Syntax: PIE <Variable name> {SUM = <Field name>}

The PIE command will make a PIE chart of the variable named.
Normally, segments of the pie will represent counts of records with

different values of the variable. If the SUM is specified, the segments represent sums of the variable given in the SUM parameter.

Pie charts should be used for non ordered data, such as race or sex, but are usually not appropriate for ordered data, such as age or blood pressure, where a bar or line chart would be better.

Titles may be added to the graph using the TITLE command. If ROUTE PRINTER is in effect, the graph will be printed.

Example: **PIE RACE**

See also: BAR, HISTOGRAM, LINE, SCATTER, TITLE

PROCESS

Syntax: PROCESS

This command causes ANALYSIS to pass through the current file and to perform all calculations, but no table is produced.   Output, if any, must be programmed with TYPE or similar commands. It is useful for setting the value of cumulative or global variables that require a pass through the file to set their desired values.

Example:
DEFINE SUM ###.## CUMULATIVE
SUM = SUM + COST
PROCESS
TYPE "The total cost is $@SUM"
This will display the sum of the items in COST for all records, without listing any of the individual COSTs.

QUIT (<F10>)

Syntax: QUIT

Typing QUIT exits the ANALYSIS program and returns to the EPI menu or to DOS if ANALYSIS was run directly from DOS. Pressing the <F10> function key has the same meaning.

Examples:
QUIT
<F10>   (pressing the <F10> key)

READ

Syntax:

READ <data file name> { <data file name>,..up to 10 names}

READ makes a .REC file the active data set. If no suffix is present in a
file name, it appends ".REC" to the file name before attempting to open
the file. If no file name is given, a list of available .REC files appears.
One of these can be chosen by moving the cursor bar with the arrow keys
and pressing <Enter>. READ *.DBF produces a similar menu for
dBASE files. If more than one file is specified, the files must be of
exactly the same format, i.e., from the same questionnaire. All the files
specified will be treated as though they were a single file. If you wish to
maintain several files of reports from different regional offices, for
example, READ using several file names will tabulate all the data in the
files as though they had been merged, but the files will remain separate for
other purposes.

READ erases or "forgets" all previous IF, SELECT, LET, and RECODE
statements. All DEFINEd variables except for those designated GLOBAL
are removed. READ can therefore be used to reset SELECT, IF, DEFINE,
and SORT statements previously entered and begin processing with a
clean slate, except for GLOBAL variables, which will retain their
definitions and values.

Examples:
READ ESP03
READ ESP04.REC
READ ESP01 ESP02 ESP03 ESP04 ESP05
READ DATA.DBF   (Reads a dBASE III or III Plus file)

RECODE

Syntax: RECODE <Var1> { TO <Var2> { BY <Num>} } Codes

Codes are represented in the form:
<Old value>=<New value> <Old value>=<New value> etc. ELSE=<New value>

The RECODE command can be used to group variables, translate codes
from one system to another, or assign labels to numeric codes. The first
variable listed is the source of the data to be recoded. The second variable
is the recipient of the recoded information. Directions for recoding are
given after the two variables.

RECODE is performed once for each active record in the file when the file is read as during a LIST, FREQ, TABLES, or MEANS command.

Examples:

RECODE RACE TO RACELABEL 1=White 2=Black 3=Asian
 4=Indian 5=Other 9=Unknown Else=Missing
RECODE NUM TO ODDEVEN 1,3,5,7,9=odd 0,2,4,6,8=even

If **ELSE** is included, all values not specifically mentioned will be recoded to the value given for "else". Two other special words, **LO** and **HI**, are used like ELSE, but indicate that all higher or lower values than the ones mentioned should be set to the LO or HI value.

RECODE AGE TO AGEGROUP 1-18=child else=adult
RECODE LVALUE TO LVALUE LO-79=HYPO
 80-200=INTERMED 201-HI=HYPER

If a second variable is not given, it is assumed to be the same as the first:

RECODE RACE 1=2 2=1 ELSE=2

will temporarily recode the data in RACE, until the next READ statement occurs.

Numeric fields may be recoded using BY:

DEFINE AGEGROUP _____
RECODE AGE TO AGEGROUP BY 10

This will produce 10-year age groups with appropriate labels. The second variable must be given in statements using BY, and a text variable of sufficient size (10 characters if AGE is three digits) must first be defined.

If more than one RECODE statement is given, the conditions accumulate so that all are implemented. You cannot recode a variable which is itself the result of a recode, however. Labels in RECODE may be specified in upper or lower case; they will all be treated as upper case. Quotation marks are only necessary if the label contains spaces or punctuation characters such as "-".

RECODE ANSWER TO ANSWER2 1="No Thank You" 2=yes

REGRESS

Syntax:
REGRESS <Dependent Var> = <Independent Var1> <Independent Var2> etc.

This command performs multiple linear regression with up to 15 variables.

The formula and an explanation of the results are given in Applied Regression Analysis, 2nd edition, by Draper and Smith (John Wiley & Sons, New York, 1981, p. 84 ff).

RELATE

Syntax: RELATE <Common Identifier Variable name> <File name>

The RELATE command links one or more files to the main file during analysis, using a common identifier to find matching records. In a sense, it performs the functions of the MERGE program dynamically from within ANALYSIS. The changes are not permanent, and the linked files remain separate on the disk, having been linked only temporarily by a system of pointers in random access memory.

To use RELATE, a main file must have been made active with the READ command. The file to be linked must have an identifier field such as HOUSEID that has identifiers relating records in the two files. The related file may have fewer records than the main file, but at least some of the records should have the same identifier in the identifier field. Identifiers in the related file, but not necessarily in the main file, must be unique--that is, without duplication.

After issuing the RELATE command, the variables in the related file may be used in all commands except UPDATE as though they were part of the main file. UPDATE may be used to change data in the main file, but not in related files, although the related fields will be visible in UPDATE.

Frequencies, cross tabulations, and other operations involving data in both the main and related files can be performed. If a permanently linked file is desired, the commands ROUTE <File name>.REC and WRITE RECFILE may be used to create a new, permanently linked file.

More than one file may be RELATEd to the main file by using successive RELATE commands.

Sometimes a variable name may be duplicated in more than one file. ANALYSIS keeps the variables in related files separate, and they can be displayed with the VARIABLES command. In referring to a variable in a related file, you may (optionally) use the form: HOUSE.AGE to represent the variable AGE in the file HOUSE.REC. This will distinguish it from another variable AGE that might be in the main file.

By using the relational features of ENTER and CHECK and the RELATE command in ANALYSIS, you can set up relational (actually "hierarchical") databases just as in dBASE or other database management systems. This provides powerful database handling in combination with the epidemiologic statistics available in ANALYSIS.

Example:
READ PERSON
RELATE HOUSEID HOUSE
TABLES AGE HOUSE.AGE

RENAME

Syntax: RENAME <old file name> <new file name>

RENAME allows you to rename a disk file. If no file with <old file name> exists, or if a file with the name <new file name> already exists, the command gives a warning and does nothing.

Example: **RENAME esp03.rec esp03.bak**

REPORT

Syntax: REPORT <Report file name>

The REPORT command takes instructions from a text file that you construct in EPED and constructs a report according to the format specified. The file should be given the extension .RPT, although this is not required.

Lines in a report file must not exceed 255 characters. Macros are counted as they appear in expanded, not shorthand, form. Output to a file or printer cannot have a page width greater than 200 characters. To obtain printed output from REPORT, the command ROUTE PRINTER must be

given before the REPORT command. It may also be necessary to set the page width for the printer with the SET PAGE = command, if the width of the printed material will exceed 80 characters.

Commands begin with the # sign. Within a report file, the following commands are allowed:

#USES <Variable1 name> { <Variable2 name> <Variable3 name>, etc.} The #USES command must be on the first line of every .RPT file. It tells ANALYSIS to construct a table in memory using the variables specified. All cells and marginal totals in the table then become available for printing as specified in statements in square brackets. The instructions that you give in square brackets are a way of putting any cell or marginal total in the table anywhere on the page, or choosing to omit it from the output.

Example:
#USES COUNTY DISEASE SEX

#USESSUM <Variable1 name> { <Variable2 name> <Variable3 name>, etc.}

This command is similar to the #USES command except that the sum of the values of the variable replaces the count. The two commands are analogous to the TABLES and SUMTABLES commands in ANALYSIS. #USESSUM is used to handle aggregate data, as in the files TABLE1.RPT and TABLE2.RPT in the NETSS surveillance system on the Epi Info disks.

Example:
#USESSUM COUNTY DISEASE SEX

Plain text is simply printed as written in the page location where it appears. The \c and @<Variable name> notations can be used to center a line or print the value of a previously defined or questionnaire variable.

Examples:
\cExtended Report
prints "Extended Report" in the middle of the current page width.
Report for @SYSTEMDATE
prints the value of SYSTEMDATE after the words: "Report for".

[<Variable1 value>,{<Variable2 value>, <Variable3 value>, etc.}]{:<No. of digits>}

Square brackets enclosing a list of values of variables specify that the contents of the table cell determined by the values will be printed at the location where the bracketed expression appears. The list of values must correspond in number and position to the list of variable names in the #USES statement. Text values must be the same case as in the data file, i.e., upper and lower case are significant. Text should not be in quotation marks. Values are separated by commas. Totals may be specified by omitting one or more of the values, in which case the total for all values of that variable will be understood. The number of digits to be displayed is indicated by the optional colon and integer value after the second bracket.

Examples as they would appear in a report (.RPT) file:

#USES COUNTY DISEASE SEX

The number of cases of hepatitis for Monroe County among females is [MONROE, HEPATITIS, F]
The number of cases of all conditions in Monroe County is [MONROE,,].
The total number of hepatitis cases is [,HEPATITIS,].
The total number of males is [,,M].
The grand total is [,,]:3

Note the specification of three digits for the grand total.

{ <Mathematical expression> } :<Number of total places>.<Decimal places>

Curly brackets indicate a calculation that uses counts in square brackets or variable or literal values included in the curly brackets. The total number of digits, including the decimal point, and the placement of the decimal point are indicated to the right of the curly brackets. The operators allowed are those listed under Operators in the ANALYSIS reference section.

Examples:

The percentage of males is { [,,M] DIV [,,]} : 3%.
The circumference is { @RADIUS * 3.14159} :5.2

^<Variable name>
This expression means the *Value Label* of the variable, e.g., M for SEX.

When used inside the square brackets, ^<Variable name> refers to the current value label determined by the #FOREACH command described below. [,,^SEX] means the *Count* for the current SEX as determined by the progress of the #FOREACH operation.

#FOREACH <Variable name>
        <Bracket or text expressions or other report commands>
#ENDFOR

The #FOREACH/#ENDFOR command performs the operations between the two parts of the command repeatedly, once for each value of the specified variable. If the variable is SEX, for example, the operations will be performed for the missing value, for **F**, and for **M**.
#FOREACH/#ENDFOR statements can be nested so that one occurs inside another.

Examples:

#FOREACH SEX	Prints:	M
^SEX		F
#ENDFOR		

#FOREACH SEX	Prints: M F
**^SEX **	
#ENDFOR	

#FOREACH SEX
   Cases with Sex ^SEX = [,,^SEX]:2
#ENDFOR

These expressions print, for example:
```
Cases with Sex F = 24
Cases with Sex M = 25
```

See Chapter 24 for further examples.

#<Macro name>   <Bracket expression>

A Macro is a name that represents a longer expression in the report file. The Macro name as defined may be used elsewhere in the report in place of the bracket expression. This reduces the length of commands in the report file for better placement on the page.

Examples:
#CELL [^race,^sex,^diseasenam]:6

The word #CELL can then be used to represent the bracket expression: **#CELL** will print the count for the current values of the variables with six digits.

#IF <Condition>
   <Bracket or text expressions or other report commands>
#ENDIF

#IF/#ENDIF commands are used to control execution of other statements based on special criteria. They can, for example, be used to suppress printing or print a special character when a cell value is less than 5 to preserve confidentiality.

Example:
#IF #CELL > 5
     #CELL
ELSE
     --
ENDIF

This prints the count for the cell specified in the Macro example above if the value is greater than five; otherwise, it prints two hyphens. Several lines of expressions or commands can be placed between the parts of the #IF/#ENDIF.

#NEWPAGE
#NEWPAGE causes the display to skip to the next page or screen.

Example: **#NEWPAGE**

Examples of report files included with Epi Info are:

DISEASE.RPT for DISEASE.REC;
TABLE1.RPT for GEPI.REC; This produces a hypertext file with many embedded tables, marking extreme values based on algorithms contained in #IF statements.
OSWEGO.RPT for OSWEGO.REC.

RETURN

Syntax: RETURN

RETURN causes program execution to resume with the command after
the last GOTO statement that was executed.

Example:
<Program statements>
GOTO DOQUESTION
<More program statements>
QUIT

:DOQUESTION
<Statements to ask a question>
RETURN

In this "program," the action begins with <Program statements>. The
GOTO causes a skip to DOQUESTION and execution of <Statements to
ask a question>. At the RETURN, control goes back to just after the
GOTO DOQUESTION and <More program statements> are executed
until the program terminates with QUIT. The DOQUESTION subroutine
could be called from several points in the program in this way, if desired.

See GOTO.

ROUTE

Syntax:
ROUTE <File name>
ROUTE SCREEN
ROUTE PRINTER

Route opens the specified file as the new output file. All output from the
program, with the exception of selected error messages, will go to the file
specified. Two special "files", "SCREEN" and "PRINTER", allow output to the
screen and to the printer, respectively. To cancel a previous ROUTE to a file or
to the printer, use ROUTE SCREEN.

If a file name is given and the file already exists, ROUTE will not erase the
previous file but will append the new material to whatever was in the file. If you
wish to prevent this, erase the old file first, using the ERASE command.

The <F5> key may be used to alternate between "ROUTE PRINTER" and
"ROUTE SCREEN" as indicated at the bottom of the ANALYSIS screen.

Examples:
ROUTE results.txt

ROUTE A:\RESULTS.TXT
ROUTE PRINTER (Turns off previous ROUTE to a file)
ROUTE SCREEN (Turns off previous ROUTE and closes the output file if necessary)

RUN

Syntax: RUN <program file name>
RUN allows you to specify a file name containing commands. The
program will read commands from the specified file until the end of the
file is reached or another RUN command is encountered. One program
can be RUN from another program.

Control is transferred to the second program and does not return to the
first automatically, but a RUN command in the second program can run
the first file again.

Examples:
RUN esp.pgm
RUN (Presents a list of available .PGM files for choice, using the cursor
        bar and the <Enter> key.)

SAVE

Syntax: SAVE <File name>

SAVE stores previously entered commands in a program file for later use
and/or editing in EPED. The number of commands saved is determined
by the value of CMDSTACK, and is usually 20. It may be changed by the
command SET CMDSTACK = <a number>. SAVE saves program
commands, not data, and .REC files are not affected.

Example:
SAVE TEST.PGM

SCATTER

Syntax:
SCATTER <Variable1> <Variable2> {/R}{/X=<min>-<max> /Y=<min>-<max>}

SCATTER will make a scatter plot with Variable1 on the horizontal axis and
Variable2 on the vertical axis. If "/R" is included after the field names, a least-
squares regression line will be drawn through the data points. Scatter plots are
most useful when both fields contain continuous numeric data like age or blood
pressure rather than coded or yes/no information.

The maximum and minimum numbers on the X (horizontal) and Y (vertical) axes can be made more extreme by specifying new numbers with the X and Y parameters:

Scatter <Variable1> <Variable2> /X=<min>-<max> /Y=<min>-<max>

No spaces are allowed in the expressions Axis=<min>-<max>, but a space must be present before each slash ("/"). The minimum cannot be increased nor the maximum decreased from the values chosen by the program.

If ROUTE PRINTER is in effect, graphs will be printed as they are produced.

Titles may be added to the graph using the TITLE command. If ROUTE PRINTER is in effect, the graph will be printed.

Examples:
SCATTER SYSBP BODYMASS /R
SCATTER AGE SYSBP /X=0-100 /Y=0-250

See also: BAR, HISTOGRAM, LINE, PIE, TITLE

SELECT

Syntax: SELECT <expression>

SELECT allows you to specify an expression that must be true for a record to be processed. If the current selection is "age > 35", then only those records with age > 35 will be processed. SELECT used alone without an expression will cancel all previous SELECT statements.

SELECT expressions are additive so that the two expressions:

SELECT AGE > 35
SELECT SEX = "F"

are equivalent to **SELECT (AGE>35) AND (SEX = "F")**.

The length of all active SELECT expressions must not exceed 255 characters. This limitation can be overcome by using IF statements to set a single variable to a value that is used in a simpler SELECT statement such as SELECT CHOSEN = "Y".

Examples:

SELECT ((age > 35) and (sex ="F")) or ((age > 40) and (sex ="M"))

Note the use of parentheses to show relationships.

SELECT NAME = "MAC" will select only "MAC", "Mac", etc., an
exact match except for case.

SELECT NAME [1,3] ="MAC" will select "MAC", "MACDONALD",
etc., an inexact match, in which the first three letters of NAME are
"MAC". See SUBSTRINGS for the meaning of NAME[1,3].

SELECT   (Clears previous SELECT statements)

See OPERATORS for help in constructing expressions and
SUBSTRINGS for partial matches.

SET

Syntax: SET {<Condition> = <Value>}

SET allows you to set the various formatting parameters of ANALYSIS. SET by
itself displays the settings that are currently active.

SET commands can be issued from the keyboard or placed in program files. They
are especially useful when placed in a file called CONFIG.EPI, that is run
automatically when ANALYSIS is first run. Placing SET commands in this file
has the effect of making the setting permanent.

The Conditions that affect **printing** and their values are:

SET PRINTER Graphs in ANALYSIS can be sent to Epson-/IBM-compatible
dot-matrix printers and to Hewlett Packard-compatible laser printers and plotters.
There are 30 printer drivers supplied with Epi Info. The name of any of these files
can be used with SET PRINTER = <Driver name>. The file names of drivers
begin with "$" and end with .BGI. To set ANALYSIS to produce a CGM file, for
example, you would give the command, SET PRINTER = $CGM. Commands for
normal output are best placed in the file called CONFIG.EPI so that the printer is
set each time ANALYSIS is run.   The drivers supplied are:

```
File name       Size  Format
$BMP      BGI   17768 BMP (Windows 3.x) Format
$CGM      BGI   11818 Computer Graphics Metafile
$FX       BGI   20894 Epson FX80 Printer
```

```
$PS      BGI    15518 Postscript
$TIF     BGI    20862 Compressed TIF Format
$UTIF    BGI    20862 Uncompressed TIF Format
$WPG     BGI     9642 WordPerfect Graphics
$CFX     BGI    21128 Epson 9-pin Color Printer
$CLQ     BGI    23726 Epson 24-pin Color Printer
$DJ      BGI    21792 Hewlett-Packard Deskjet
$DJC     BGI    22252 Hewlett-Packard Color Deskjet
$DXF     BGI    16084 AutoCad DXF Format
$HP7090  BGI    16264 Hewlett-Packard Plotter 7090
$HP7470  BGI    16232 HP Plotter 7470
$HP7475  BGI    16336 HP Plotter 7475
$HP7550  BGI    16336 HP Plotter 7550
$HP7585  BGI    16368 HP Plotter 7585
$HP7595  BGI    16368 HP Plotter 7595
$IBMQ    BGI    20220 IBM Quietwriter Printer
$IMG     BGI    20402 GEM IMG Format
$LJ      BGI    23146 Hewlett-Packard Laserjet
$LJ3R    BGI    23350 Hewlett-Packard Laserjet III
$LQ      BGI    23492 Epson 24-Pin Printer
$OKI92   BGI    18064 Okidata 92 Printer
$PCX     BGI    17616 PCX (640 x 480) Format
$PJ      BGI    24940 Hewlett Packard Paintjet
$PP24    BGI    23562 IBM Proprinter, 24 Pin
$TSH     BGI    22306 Toshiba 24-Pin Printer
$VSHO    BGI    13112 ANSI NAPLPS/Video Show Format
```

SET GRAPHtofile This command selects a driver for graphic output to a file. If you choose SET GRAPH = $CGM, for example, the CGM driver will be used if (and only if) output has been ROUTEd to a file. If output is to the printer, however, the driver specified in SET PRINTER will be used.

SET PORT Directs output to the proper port if your printer is connected to a port other than the standard parallel port. The possible values are LPT1, LPT2, COM1, and COM2. Try this command if no output seems to be going to your printer when you print a graph. This command does not affect printing of tables or text.

SET BAUD Sets the rate of transmission for the port. This is necessary for plotters and sometimes for laser printers using a serial port. The choices are 300, 600, 1200, 2400, 4800, 9600, and 19200.

SET PMODE The printers served by ANALYSIS for printing graphs have a choice of modes that control image size and resolution. Generally the larger the

image and the higher the resolution, the longer it will take to print. You may wish to experiment with the following modes on your own printer, using SET PMODE = to set the mode. The allowable values for different printers, all single digits, are:

Epson dot-matrix printers: 0=Half page; 1=Full Page; 2=Full Page Bold. Epson LQ1500 printers have the preceding modes and also: 3=Letter quality, half page; 4=Letter quality, full page.

Hewlett Packard LaserJet printers have: 0=Half page, Low resolution; 1=Half page, Medium resolution; 2=Half page, High resolution; 3=Full page, Low resolution; 4=Full page, Medium resolution; 5=Full page, High resolution.

Hewlett Packard 7470A and 7475A pen plotters have two modes, both producing a full page: 0=Draft quality; 1=Letter quality.

SET PAGE Controls the size of the page, in characters of width and lines of height. Normally these are 23,80 for the screen and 66,80 for the printer.

SET PAUSE Controls whether or not the program stops output after every page (good for printing single sheets of paper). Pause is normally ON for the screen and OFF for the printer.

SET PLINES This represents the number of lines per page on the printer. Frequently it is necessary to set the page length for the printer independently of that for the screen. Normally PLINES is 66, but it may be set to 60 for many laser printers that skip the top and bottom three lines on every page, for example.

The conditions that affect the **appearance of Tables, Frequencies, or Lists** are:

SET CRITERIA When CRITERIA = ON, the current selection criteria will be printed out before each list, frequency, or table. The normal setting is ON.

SET FILLCHAR Controls the character used to fill blank space in numeric fields written with the WRITE or LIST command.

SET LINES When LINES = ON, tables will be printed with horizontal and vertical lines separating the cells. The normal setting is OFF.

SET LISTREC When LISTREC = OFF, record numbers will not be included in the output produced by the LIST command. The normal setting is ON.

SET MAXCELLWIDTH This is a number which specifies the maximum width of a table cell in number of characters, normally 15. It may be used to limit the width of cells and thus allow more entries across the page in a large table. If

MAXCELLWIDTH is not specified, the width of each cell depends on the size of the corresponding variable.

SET PERCENTS When PERCENTS = ON, tables will be printed with percentages below each count. The row percents are printed first, with an arrow pointing to the denominator on the right. The column percents are printed below the row percents. The normal setting is OFF.

SET STATISTICS When STATISTICS=ON, statistics will be produced for each table specified in the TABLES command. When OFF, only the table will result. ON is the normal condition.

SET SWITCH When SWITCH = OFF, the values "Y" and "N" will not be changed to "+" and "-". Normally this is done so that the position of these values in tables is convenient, with the "+" values coming first.

The commands that affect the **processing of records or other ANALYSIS functions** are:

SET BREAKCMD The command performed by BREAK when a change occurs in the field(s) specified in the BREAK statement can be set as in the following examples:

SET BREAKCMD = TYPE "New Age Group"
SET BREAKCMD = TYPE "\nAGE @AGE \n"

If BREAKCMD is not specifically SET, the action performed by BREAK is to skip to the next page during a LIST. This occurs only when output is ROUTEd to the printer or to a file. See BREAK for more information.

SET BUFSIZE A portion of memory called a "buffer" is used to increase the speed with which files are read. Large buffers provide faster processing, particularly after a file has been read the first time. Allowable values are from 0 to 32,767.

SET CMDSTACK Determines the number of commands entered from the keyboard which can be recalled with the up-arrow key or saved with the SAVE command.   For example, **SET CMDSTACK = 30** will allow the last 30 commands entered to be retained. Normally CMDSTACK is set to 20, but it can be set to a smaller number to conserve memory or a larger one if you make extensive use of the SAVE feature. The maximum number of commands that can be saved is 100.

SET CONFIRM Controls the amount of interaction required to change a record in UPDATE. When CONFIRM = ON (the normal state), changing an individual record requires a "Y"es response to, "Write record to disk (Y/N)?". SET CONFIRM = OFF allows an entire column representing a single variable in many records to be editied without confirmation other than the <down arrow> or <Enter> keys.

SET IGNORE When IGNORE = ON, missing values will be ignored in constructing tables. OFF is the normal condition. Set IGNORE = ON to produce 2-by-2 tables with Yes/No variables if there are missing values in some of your records.

SET NOECHO When NOECHO = ON, the output of commands like TABLES and FREQ will not be echoed to the screen. The normal setting is OFF. NOECHO may also be used as a parameter after any command, as in FREQ SEX /NOECHO.

SET PROCESS Process has three settings: DELETED, UNDELETED, and BOTH. It controls which records will be processed by ANALYSIS. If its value is UNDELETED, only undeleted records will be processed. This is the normal setting. If it is DELETED, only deleted records will be processed; this could be used to print a list of all deleted records, for example. SET PROCESS = BOTH will cause all selected records to be processed regardless of deletion status.

SET SORT = ASCENDING /DESCENDING determines whether SORTing proceeds in alphabetical/numeric order or the reverse (highest first or descending) order.

SET SPLIT SPLIT controls the appearance of the screen. When SPLIT is ON, the screen has an upper window for output and commands are typed in the lower window. When SPLIT is OFF, the entire screen is used for commands and output as in DOS. You will want to set SPLIT to OFF when constructing your own menus for users in .PGM files, for example, as is done in the examples given in the NETSS software (e.g., DUPS.PGM).

Examples:
SET PAGE=23,80 PAUSE=OFF FILLCHAR="0"
SET STATISTICS = ON
SET IGNORE = ON
SET PROCESS=UNDELETED
SET CRITERIA = OFF
SET LINES = OFF
SET PERCENTS = ON
SET MAXCELLWIDTH = 6
SET BREAKCMD = TYPE "\C*******************"**

SET PRINTER = HP
SET MODE = 2
SET PORT = COM1
SET CONFIRM = OFF
SET

SORT

Syntax: SORT {<variable name1> <variable name2> ... <variable name N>}

SORT allows you to specify the sequence in which records will be written with the LIST and WRITE commands. If there are no variable names specified after the SORT command, the current sort is cleared, and subsequent output of records will be by their order in the original file. If one variable name is given, the records will be sorted using that variable as the "key." If more than one variable is specified, the records will be put in order by the first variable, then within a group with the same value of variable 1, the ordering will be by variable 2, etc. The process is similar to putting a group of names in alphabetical order, first by the first letter, then by the second, and so on.

Descending sorts can be performed by first giving the command SET SORT = DESCENDING. The order can be returned to ascending with SET SORT = ASCENDING.

Examples:
SORT AGE

SET SORT = DESCENDING
SORT name sex age

SUBSTRINGS

Parts of string variables, or substrings, can be indicated in brackets as follows:

Variable name [Position of first character, Number of characters]

Example:

LET yeardate = date [7,2] + "/" + date [1,2] + "/" + date [4,2]

will transform DATE, in the format mm/dd/yy, to YEARDATE, as yy/mm/dd.

In a substring such as **DATE[7,2]**, the first number in brackets is the position of the first character in the substring. Thus, if **DATE** is **01/03/87**, the seventh character is "**8**". The second number is the number of characters in the substring, in this case, 2. **DATE[7,2]** would have the value "**87**" in this example.

See OPERATORS for help in constructing expressions.

SUMFREQ and SUMTABLES

Syntax: <Variable to be summed> <Variable for which it is to be summed>
    <Additional variables for which it is to be summed>

SUMFREQ and SUMTABLES are used for processing summary records produced by the OUTPUT command. These two commands produce tables similar to those from the FREQ and TABLES commands. FREQ and TABLES *count* the number of records satisfying specified conditions and put the count in appropriate cells of the tables. SUMFREQ and SUMTABLES *add* the contents of the first variable specified for all combinations of the other variables and place the result in one or more tables. Note that one more variable must be specified than for the corresponding FREQ and TABLES commands. FREQ and TABLES add one to the count for each record in a group. SUMTABLES and SUMFREQ add the contents of <Variable to be summed> to the total.

Examples:

Suppose that two records contain the following information:

	DISEASE	COUNTY	WEEK	COUNT
Record 1	Hepatitis A	Craig	24	30
Record 2	Hepatitis A	Craig	24	11

FREQ COUNT will show the following results:

COUNT	Number
11	1
30	1

SUMFREQ COUNT WEEK will produce:

WEEK COUNT

       24   41

SUMTABLES COUNT DISEASE COUNTY WEEK will produce:

                          WEEK = 24
                          COUNTY
DISEASE   Craig   Total
Hepatitis A   41   41

The original files, from which the OUTPUT command produced the two summary records, would have produced an identical table, using the TABLES command, if the files had been combined. The OUTPUT and SUMTABLES sequence is merely a convenient way to maintain intermediate files with summary records so that large files do not have to be processed "from scratch" each time a further summary is desired.

SYSTEM VARIABLES

These variables or functions return values as follows:

SYSTEMDATE   The current date as found in DOS in the form mm/dd/yy
SYSTEMTIME   The current time as found in DOS in the form hh/mm/ss AM(or PM)
RECNUMBER   The number of the current record in file sequence. If the file   has
been SORTed in ANALYSIS, RECNUMBER represents   the
                 original order of the records, not the SORTed order.
RECVERIFIED   "Y" if the current record has been verified in ENTER (See
                 Chapter 19), otherwise "N"
RECDELETED   "Y" if the current record is deleted, otherwise "N"

Examples:
YEAR = SYSTEMDATE [7,2]
HOUR = SYSTEMTIME [1,2]
SELECT RECNUMBER > 400
SELECT RECVERFIED = "Y"
IF RECDELETED = "Y" THEN TYPE "*DELETED*"

Note that RECDELETED will be affected by SET PROCESS, since records with "Y" and "N" values will only be processed if SET PROCESS = BOTH.

TABLES

Syntax:
TABLES <Variable name1> <Variable name2> {<Variable name3>...<Variable
       name 5>}   {/SUM=<Variable name>}

TABLES does a cross tabulation of the specified variables and sends the table to the screen, printer, or other current output file. Values of the first variable will appear on the left margin of the table, and those of the second variable will be across the top of the table. Normally cells contain counts of records matching the values in the corresponding marginal labels. If the /SUM parameter is given, the cells represent sums exactly as in SUMTABLES. TABLES DISEASE COUNTY /SUM=COUNT gives the same results as SUMTABLES COUNT DISEASE COUNTY.

All variables after the first two listed in the TABLES command are used for stratification. For every possible combination of values of these variables, a separate table (stratum) for variable 1 by variable 2 will be produced. TABLES BAKEDHAM ILL SEX will produce a table of BAKEDHAM by ILL for each value of sex--one for M and one for F. TABLES BAKEDHAM ILL SEX RACE will produce a separate table of BAKEDHAM by ILL for each combination of SEX and RACE--female/black, female/white, male/black, male/white, etc. No more than five variables are allowed in the TABLES command, two of which would specify the table and three of which would be stratifying variables.

If "*" is given instead of a variable name, each variable in the data set is substituted for "*" in turn. Thus, if you would like to analyze each variable by illness status, the command:

TABLES * ILL

will produce tables of SEX by ILL, AGE by ILL, etc.

It is important to consider carefully before using "*" or requesting multidimensional tables, since there may be more tables than you want in terms of time, paper, and other costs.

Variables used for stratifying data (those after the first two) should be recoded so that items like "age" are grouped in the coding process; otherwise, a separate table would be produced, for example, for each year of age.   If PAUSE is set to ON (SET PAUSE = ON), the <ESC> key can be used to interrupt processing during a pause on the screen.

The SET command can be used to determine the format of the tables by omitting statistics (SET STATISTICS = OFF), including percentage values in the tables (SET PERCENTS = ON), printing vertical lines (SET LINES = ON), determining the maximum column width (SET MAXCELLWIDTH = <number>), omitting the printing of selection criteria at the top of each table (SET CRITERIA = OFF), and preventing the conversion of "Y" and "N" in Yes/No variables to "+" and "-" (SET SWITCH = OFF). See the SET command for further details.

Text specified in the TITLE command as TITLE 1 through TITLE 5 will be printed at the top of each table. See the TITLE command for more details.

Examples:
 TABLES disease county
 TABLES * COUNTY
 TABLES VANILLA ILL AGE SEX
 TABLES DISEASE COUNTY /SUM = COUNT

TITLE

Syntax: TITLE <Line number 1-5> { "<title text>"}

TITLE sets the title with the designated <Line number> to the specified text. Quotation marks around the text are optional. Using line numbers 1 through 5 allows up to five lines of titles to be printed above a table. The titles will be printed on all tables until another TITLE command changes the content. Within the title text, the following special commands may be used:

\C   The title will be centered.
\L   The title will be left justified.
\R   The title will be right justified.
@<Variable Name> The current value of the variable will be inserted in the text.

TITLE with a line number but no text clears the text for that line and any subsequent lines so that TITLE 1 clears all five lines with one command.

Examples:
 TITLE 1 Cross Tabulation of Disease by County
 TITLE 3   (Clears line 3 - 5)
 TITLE 2 "\CAGE GROUP @AGEGROUP"
 (Centers the phrase on the page, supplying the current value of AGEGROUP.)

TRIM

Syntax: TRIM (<Variable Name>)

TRIM strips trailing blanks from a text field. This is useful in combining string fields to produce mailing labels or for new variables to be displayed compactly in a list.

Examples:
LINE3 = CITY +", "+ STATE+" " + ZIP

might produce:
  Atlanta   , GA 30333

but

LINE3 = TRIM(CITY) +", "+ STATE+" " + ZIP
would give:
  Atlanta, GA 30333

TYPE

Syntax: TYPE "<Text>"
          or
          TYPE <file name>

The TYPE command has two meanings, determined by whether the information that follows is in quotation marks. If text in quotation marks is given, TYPE will send the text to the screen, printer, or other output file designated in the last ROUTE command. Special commands may be embedded in the text as described above under TITLE. This use of TYPE is somewhat similar to the ECHO command, but TYPE sends text to the printer or to a file if indicated by ROUTE, and ECHO displays text on the screen only.

Within the quoted text, the following special commands may be used:

  \C   The text will be centered.
  \L   The text will be left justified.
  \R   The text will be right justified.
  \N   Begins a new line, useful for creating mailing labels, for example.
  \<number> The character with the ASCII decimal code given by the number.
  @<Variable Name> Inserts the current values of the variable into the text.

As with all commands, the length of a TYPE command is limited to 255 characters. The length of the text that is TYPEd as a result is also limited to 255 characters, even though the use of variables keeps the command relatively short. An extra 255 characters of output is allowed, however, for each \N command used so that the actual output on a mailing label, for example, can be longer than 255 characters.

If no quotation follows the TYPE command, TYPE displays the contents of the file specified and sends it to the current output file. If PAUSE is SET to ON (see SET), it pauses at the end of each page and waits for any key to be pressed to continue. (Pressing <ESC> at this point stops the display.) This use of TYPE is similar to the DOS TYPE command.

Examples:
TYPE Esp03.qes   Displays the contents of the file.
TYPE "\CWeekly Report for Week #@WEEK" Centers the text on the page
with the current value of WEEK substituted for @WEEK.
TYPE "\27n"   Sends the <Esc>ape character and "n" to the current output
device. This is often used for sending control codes to a printer.

UPDATE

Syntax: UPDATE

The UPDATE command is identical to the BROWSE command, except that
changes can be made in the data file. **Before using this command, be sure to
make a copy of your data file as a backup, since changes made are
permanent!!!!** This is the only command in ANALYSIS that actually changes the
original data file.

UPDATE displays the current data file(s), including any RELATEd file, in
spreadsheet format on the screen. You can use the arrow keys to move around and
see the entire data set. An entry field is displayed at the top of the screen, and any
new value you enter will become the value of the field under the cursor bar. To
change columns of entries without "Y/N" confirmation, give the command SET
CONFIRM = OFF before entering UPDATE.

While in UPDATE, the following functions are available:

<Home><End> These keys move to the first or last record.

<F4> Full Screen. Pressing this key alternately displays the records in the data
set in questionnaire format just as they would appear in the ENTER program, or in
spreadsheet format as when UPDATE is first activated. The more familiar
question format provides orientation about the meaning of the questions. The
<F7> Prev and <F8> Next keys from the ENTER program are available in the
questionnaire mode. Changes can be made in the data just as in ENTER, and the
normal CHECK functions operate if a .CHK file is present. After making changes
in a record, you must save the record by pressing the <End> key to bring up the
question "Write data to disk (Y/N)?" and then reply "Y", just as in editing records
with ENTER. To return from the questionnaire to the spreadsheet, press <F4>
again or <Esc>.

<F6> Delete. This marks a record as deleted, or undeletes a deleted record, just as it does in the ENTER program.

<F10> Done. This terminates the UPDATE session and returns to the EPI6> prompt, ready for the next command.

NOENTER, IDNUM, and TODAY fields and fields that are defined as KEY in a .CHK file cannot be modified in UPDATE. The TODAY field is set to today's date when an UPDATEd record is written to disk. Data may be entered in any field when entering a new record.

When leaving a record that has been changed, either by positioning to a new record, or exiting UPDATE, the user is asked if the updated record should be written to disk. In the past, the record was written to disk for each field that was changed, without giving the user an opportunity to review or reject the changes.

When using BROWSE or UPDATE with records that have been sorted, the records are displayed in sorted order. If a change is made that would affect the sorted order of the records, the change in order is not made until the records are again sorted.

Fields in related files may also be modified in UPDATE. If a record is added to the main file, a record with the appropriate relational field is added to the related file, in addition to the related fields that have been entered.   The files are not re-RELATED automatically, however; the related records that are added would not appear until the user RELATEs the files.

   Example: **UPDATE**

VARIABLES (<F3>)

   Syntax: VARIABLES

VARIABLES displays a list of the variable names in the current data set, with the type and length of each, as well as the names and contents of any user-defined variables.

The function key <F3> displays a list of variables as a menu. The cursor bar can be moved with the arrow keys to the proper choice. You can tag several variables with the <+> key on the right side of the keyboard or remove the tags with the <-> key. When <Enter> is pressed, the variable name(s) will automatically be inserted in the current command.

Examples:

<F3>   Allows a choice of one or several variables.

VARIABLES   Displays current variable names, types, and sizes
                   with current values for defined variables.

VARIABLES /S   Displays the names only, in a single vertical
                   column, useful for sending the names to a
                   file for another program to read, for example.

WRITE

Syntax:
WRITE {RECFILE} {<variable name> {": " <variable width>}} ... etc.

WRITE will send records to the current output file in the format that you specify.
You can specify what variables will be written, the width of the variables, the
order in which they will appear, and whether or not the file will be a .REC file.
Separators or other material to be included in each record can be specified.

If RECFILE is specified, an Epi Info file is created, containing all the variables
named. This file can be analyzed, edited, or subjected to further data entry.
Records deleted in ENTER or SELECTed in ANALYSIS are handled as in other
ANALYSIS commands, and defined variables may be written, allowing you to
create a new Epi Info file that makes the changes permanent. To do this, first use
the ROUTE command, and then WRITE RECFILE as follows:

ROUTE <File name ending in .REC>
WRITE RECFILE

If you wish to write only selected variables, they may be listed after **WRITE
RECFILE**. The word NOT may be inserted to indicate all variables except those
following NOT. If you ROUTE to a file that already exists, WRITE RECFILE
will write only the data records, not the header information, for the file. To
prevent this, if you want to make a new file complete with the header containing
variable names, first erase any previous file by the same name with the ERASE
command.

For numeric fields, the number of digits after the decimal point can be specified.
AVERAGE: 5:2 will write AVERAGE in a variable five characters wide, with
two digits after the decimal point. Any variable that won't fit in the specified
width is truncated or padded with spaces.

A quotation can contain specialized characters. These are preceded by a backslash
('\') and are useful for sending escape codes to printers, formatting output on the
screen, and doing other things that require special control characters.

The format for a specialized character is a quotation mark and a backslash,
followed by the decimal form of any of the 255 ASCII character codes and ending
in a quotation mark. For example, the string **"\12\13\10"** would send a form-feed
character (12), a carriage-return character (13), and a line-feed character (10).

In addition to ASCII code numbers, other specialized symbols are recognized;
"\n" sends a carriage-return/line-feed combination, and **"\b"** sends a backspace
character. The record number of the current record may be included as **#REC**.

The maximum width of lines output to the printer by the WRITE command is
limited only by the printer. The limitation of 255 or fewer in Version 5 no longer
exists.

Examples:

WRITE state:2 #rec:6 disease:3 age:2 sex:1 race:1 doo:6
  dor:6 "9999999999"

WRITE RECFILE name age sex race

WRITE RECFILE NOT TIMESUPP
will create a new .REC file containing all the fields from OSWEGO except
TIMESUPP. This avoids having to name fields explicitly if nearly all are to be
included.

WRITE name, "\n", address, "\n", city, ", ", state, "   ", zip
will write the items on different lines and could be used to print a mailing label.
The total length of the items in a WRITE statement is controlled by the page
width. If this is a problem, try issuing the command SET PAGE = 66,255 before
the WRITE command and then SET PAGE = 23,80 after the WRITE command.

MISCELLANEOUS

?   ?   (The Get-Input command)

Syntax: ? <Message to User> ?

This is an interactive input command. <Message to User> will be
displayed on the screen, and the system waits for input from the keyboard.
Whatever is typed by the user is substituted in the command line in which

the two question marks appear, as though it had been part of the original command.

A literal question mark can be included in an input prompt by preceding it with a backslash (\?). The escaped characters are:

\N   insert a carriage return and line feed in the prompt.
\R   insert a carriage return in the prompt.
\?   insert a question mark in the prompt.
****   insert a backslash in the prompt.

Examples:

LIST ? What variables would you like to list ?

If the user types "RECNUM AGE SEX", the list will contain columns for these variables.

LET FIRSTDATE = "? First date to be processed (mm/dd/yy):?"

The defined variable FIRSTDATE will be set equal to the date the user enters. Note that the quotation marks must be omitted if the variable is numeric, but present for text or date variables.

IMMEDIATE LET YR = ?What year\?\N
will print

What year?

and the cursor will be placed at the beginning of the following line.

.REC File

```
4 0
_REPORTNAME    1   1  14  12   1   1  30  15 REPORTNAME
_REPORTCODE    1   2  14  12   2   0   2  15 REPORTCODE
_UPDATEDATE    1   3  14  12   3   2   8  15 UPDATEDATE
_UPDATETIME    1   4  14  12   4   1  11  15 UPDATETIME
Epi Bulletin Table 1            101/28/94 5:27:40 PM!
Demographic frequency tables    201/28/94 2:16:38 PM!
Monthly disease totals          301/13/9412:45 76 AM!
Graph historical trends         401/13/9412:47 12 AM!
Map of State disease totals     501/13/94 4:09:22 PM!
Line list + check duplicates    601/28/94 3:08:34 PM!
```

Help File for EPIGLUE

```
.topic INDEX=0
<<Epidemiologic Information System (EPIGLUE) |INDEX>>
   <<Overview>> of EPIGLUE
   <<Starting>> and configuration EPIGLUE
   <<Using the Hypertext Viewer|VIEWHYPER>>
   <<Hypertext compiler|HYPER>>
   <<Menu>> description files
       <<Menu layout|LAYOUT>>
             <<MENUITEM>>
             <<MENUEDIT>>
             <<POPUP>>
       <<Hint table|HINT>>
       <<Command definitions|COMMAND>>
             <<DIALOG>>
             <<FILEDIALOG>>
             <<PARAMETER>>
             <<FILEPARAM>>
             <<FILTERS>>
```

Chapter 37

Epi Info File Structures; Printer Codes

The Structure of Epi Info .REC Files

For programmers wishing to write Epi Info-compatible programs, the structure of Epi Info .REC files is illustrated by the files FSTRUC.QES and FSTRUC.REC on the Epi Info disks. FSTRUC.QES was designed to illustrate all of Epi Info's field types. After two records have been entered, FSTRUC.REC has the following structure:

```
For the Prompt                        For the Data Entry Field
Chr     Name        X    Y Color X    Y  TypeLenColor   Prompt or question
21 1
#EPIINFO         36    1  14  44    1    0    0   15   Epi Info
#SAMPLEQUES      15    2  14  65    2    0    0   15   Sample Questionnaire To Show File Structure
_TEXTFIELD        1    4  14  12    4    1   21   15   Text Field
_UPPERCASEF      36    4  14  52    4    3   15   15   Uppercase Field
#NUMERICFIE       1    6  14  15    6    0    1   15   Numeric Field
#FOURPLACEN      21    6  14  40    6    0    4   15   Four-Place Numeric
#SIXPLACENU      47    6  14  65    6    6    6   15   Six-Place Numeric
#FIXEDDECIM       1    8  14  16    8  102    6   15   Fixed Decimal
_USDATE          27    8  14  37    8    2    8   15   U.S. Date
_USSHORTDAT      49    8  14  65    8    2    5   15   U.S. Short Date
_EUROPEANDA       1   10  14  15   10   11    8   15   European Date
_EUROPEANSH      28   10  14  48   10   11    5   15   European Short Date
_YESNOFIELD       1   12  14  14   12    5    1   15   Yes/No Field
_LOCALPHONE      19   12  14  38   12    9    8   15   Local Phone Number
_LONGDISTAN       1   14  14  33   14    7   13   15   Long Distance U.S. Phone Number
_DATESLASTC       1   16  14  22   16   10    5   15   Dates of last change
_D1TESLASTC      31   16  14  31   16   10    8   15   _
_D2TESLASTC      43   16  14  43   16   10   10   15   _
_AUTOMATICI       1   18  14  21   18   12    5   15   Automatic ID number
_TENPLACEID      33   18  14  49   18   12   10   15   Ten place Idnum
_COMMENT          1   20  14   9   20    1   58   15   Comment
Contents of Txt FieldUPPERCASE      12222333333444.4401/02/8901/0202/01/8902/0!
1Y456-2271(404)639-3311                        1    1          This is a sample !
questionnaire to demonstrate file struct.!
                                                                               !
                   09/0909/09/8909/09/19892    2          This record demon!
strates missing data and a deleted record?
```

The Header

The first number on the top line indicates the number of lines of header material that precede the data records. The second number is the color of the background on the screen for this questionnaire, 1 for dark blue, for example.

Each line of text or data-entry field in the questionnaire has a line of information in the file "header". The header describes both the data records and the appearance of the screen when the questionnaire is displayed. In order across each line of the header are the following data items:

The character to be displayed in the entry field. Number signs are used for text lines even though they are not displayed.

The field name, up to 10 uppercase characters, without punctuation or spaces. The name must begin with an alpha character, but may contain digits.

The column in which the text preceding the entry field begins. If the line is only text, it is displayed in this column. The first column on the left of the screen is column 1.

The line from the top of the screen where the text and/or field is displayed. The top line of the questionnaire is line 1.

The color of the text. The color numbers are illustrated on the Epi Menu Setup screen.

The column where the data-entry field begins.

The line where the data-entry field begins.

A code number for the field type. Note that code numbers for numeric fields also give the number of digits.

The number of characters in the field, or field width. This is 0 if the field contains only text without one of the designated Epi Info field types.

The color of the entry field itself.

The text that will be displayed on the screen at the location

specified in the third and fourth items above. This may be up to 80 characters wide.

Data Records

After the header, any number of data records can be arranged in lines of 78 columns with a termination character in column 79 or after the last variable on the last line of the record. Normally the terminator is an exclamation point(!). For deleted records the final terminator is a question mark(?) and for verified records a caret mark (^). The terminators were added to Epi Info records because some word processors like WordStar delete trailing blanks when a sequential file is edited.

Note that missing data items are represented by spaces, and that the format is that of fixed-length fields. The location of any data item can be calculated from the field lengths in the header, since the data items are in the same order as those in the header.

Other Details

Epi Info .REC files are handled in Pascal as a File of Byte, although they may be treated in other systems as sequential files. Each line is terminated by standard DOS end-of-line markers (Hex 0D 0A). An End of File marker is optional.

Epi Info Help Files

The help file for ANALYSIS and help files that may be constructed to work with the HELP command in .CHK files are created in an editor, such as EPED, in ASCII format and run through a program called HELPPREP. HELPPREP compresses the file and prepares it for use.

Several commands are used within help files. They are preceded by an exclamation point on the left margin, as follows:

!WIDTH <Width of window on the screen, up to 78 characters>
This sets the width of the window that will be displayed. The actual width available for text display is this number minus four, since the edges of the window each take two characters.

!INDEX <Number indicating order in which topic should appear in the menu of
        topics>

Normally when the automatic menu appears, it is sorted in alphabetical order by topic name. The numbers in an INDEX statement following the TOPIC statement will override this order. Small numbers cause topics to appear at the top of the list and large ones cause the topic to be near the end.

!TOPIC <Topic number> <Name of topic>
Each block of subject matter must begin with this command. The numbers are from 1 to 32,000, and the topic name may be any phrase, with or without spaces. Quotation marks are not necessary unless you want them printed at the top of the help window with the topic name.

!SUBTOPIC <Topic number> <Name of topic>
A SUBTOPIC is similar to a TOPIC with respect to screen display, but will be ignored during printing. All of a TOPIC will be printed at one time, including any SUBTOPICs.

!NOWRAP
Normally, the text is automatically word wrapped to fit the size window selected with the !WIDTH command. If you do not want this to happen, the !NOWRAP statement will turn off this feature. Lines longer than the window would simply be truncated in this case. The !WRAP command turns wrapping back on.

!PAGE
This command marks the end of a screen page, so that text above will be displayed and blank lines will appear below the line containing !PAGE. This allows users to decide where to place page breaks in blocks that are longer than one screen and therefore require scrolling with the <PgUp> and <PgDn> keys.

!LINE
Inserts a line break (end of line), even if word wrap is on.

When construction of the help file is complete, it must be run through the HELPPREP program before use. To run HELPPREP, type from the DOS prompt:

HELPPREP <Text file name> <Completed help file name> {/Q}

The optional "/Q" prevents HELPPREP from issuing warnings about line wrapping.

For example:

HELPPREP EHELP.TXT EHELP.HLP

HELPPREP will construct a binary file that can be called from the HELP command in a .CHK file as described in Chapter 22.

Creating Help Files for EPIGLUE/EPI6, the Menuing System

Help files for EPIGLUE/EPI6 (subsequently referred to as EPI6), have (unfortunately) a different structure from those for the rest of Epi Info. They are converted from ASCII text format to help files by a different compiler called HYPER.EXE, rather than by HELPPREP. EXE. A help file for EPI6 starts life as a standard text file that you can create in EPI6 or in the EPED editor. To be used in EPI6 or a related menu, it must have at least one TOPIC labeled with a period on the left margin, the word TOPIC, and a name for the topic. Many TOPICs may be indicated if desired, each one with its own block of text to display.

Within the text may be embedded the symbols "<<" and ">>" (ASCII 174 and 175, respectively), indicating special words or phrases that are active or "hypertext." Each hypertext marker also must indicate where the cursor should jump when the user selects the highlighted hypertext. EPIGLUE.HPT is an example of a help file before compiling. The special commands and markings for help files are:

.TOPIC
Syntax: .TOPIC <Topicname> {=x} {/NOWRAP}
                                   {=DEFAULT}
                                   {=MENU}
                                   {=GENERAL}
                                   {=MESSAGE}
                                   {=BATCH}
                                   {=BRANCH}

<Topicname> is a name of your choice, used to refer to the block of the text that will be displayed when a HELP command refers to <Topicname>. It may (Version 6.01 and later) contain spaces. The "<" and ">" here mean "Substitute your own topic name," and are not included in the .TOPIC line.

=x is the optional Context number for context sensitive help. This should be a unique number from 2000 to 4999 if the compiled help file is EIS_HELP.HLP. Numbers from 5000 to 9000 will be assigned automatically to TOPICs in other help files.

Special kinds of TOPICs can be specified, each of which has a special function in the hypertext system. The functions can include popping up messages, carrying out an activity that can be expressed with DOS batch commands, or branching to

display a TOPIC from another hypertext file. The special TOPICs help to make hypertext a complete aid to navigation though information sources or an "intelligent" aid to teaching and the provision of context-sensitive advice.

The special TOPIC types include:

=DEFAULT
Value=0. This corresponds to the default TOPIC that appears when HELP is first activated. There should be only one DEFAULT topic in a file. Range 1-1999 is reserved for built-in options of EPI6, such as CUT and COPY.

=MENU
Value=2000, defining the lower bound of the range 2000 - 4999. Indicates that subsequent topics will refer to user defined menu options. MENU should appear for the first menu option. There should be only one MENU topic in a file. All subsequent topics will be numbered incrementally, until a GENERAL instruction is encountered.

.TOPIC MENU1=MENU is equivalent to **.TOPIC MENU1=2000**

=GENERAL
Value=5000, defining the lower bound of the range 5000 - 9999. Indicates the beginning of context free help screens in a context sensitive help file. MENU topics, defined above, will provide the context-sensitive help. GENERAL should usually follow the group of topics designated for the user defined MENU. Only one GENERAL instruction is allowed in a help file.

.TOPIC All other topics=GENERAL is equivalent to
      .TOPIC All other topics=5000

=BRANCH
Defines a value in the range 10000 - 11999. **.TOPIC Look at another file=BRANCH**, followed on another line by the name of a help file and a block number, will display the specified help file, at the specified block. Up to 2000 BRANCH topics are allowed in a file.

Example:
.TOPIC Display the list of case definitions=BRANCH
CASEDEF.HLP,1100

The file CASEDEF.HLP is loaded with the topic 1100 active.

For BRANCH topics, you must specify the target TOPIC by number, not topic

name. If your TOPICs have been given names rather than numbers, the necessary numbers may be obtained from the .LST file generated by HYPER.EXE during compilation.

=BATCH

Defines a value in the range 12000 - 13999. TOPIC Run an external program=BATCH will treat the commands in the topic definition as DOS batch commands. It is possible to execute an external application from a hypertext file, such as displaying a graphic. Each BATCH topic should have at least one instruction. Up to 2000 BATCH topics are allowed in a file.

Example:
.**TOPIC Display a graph=BATCH**
CD \HG
HG SLIDESHOW.SHW

Reference can be made to global variables such as %P1 and %G3 in a batch block. Refer to chapter 20 for explanations concerning the %P and %G parameters.

=MESSAGE

Defines a value in the range 14000 - 15999. This is a reserved topic definition that will display a line of text in a popup box, at the cursor location, without erasing the screen. It can appear anywhere in a help file. Each MESSAGE topic should have at least one line of text. Up to 2000 MESSAGE topics are allowed in a file.

Example:
.TOPIC Display warning topic=MESSAGE
You have selected the warning topic . . .

/NOWRAP

When /NOWRAP is specified, text wrapping is disabled for the entire topic.

BOOKMARK

A bookmark is a mark in the hypertext file that allows the user to return to a given location. By default, a bookmark is inserted on the topic that was active when the window was closed. This allows the user to start later at the same place.

To return to the bookmark, you can either:

* Click on the exclamation mark ! on the window frame
* Insert a **<< Go to bookmark | _BOOKMARK>>** anywhere in the file.

The topic _BOOKMARK is automatically defined by the program.
* Call the hypertext file through a HELP command with the /BMK switch

Context Sensitive Help Files

When creating user-defined menu help screens, the first topic should be

.TOPIC <Topicname>=DEFAULT
   Text for default context sensitive help

The Topicname here could be "General Help." DEFAULT is the Context value for the first of the built-in functions. In the example, the text following .TOPIC General Help=DEFAULT will be displayed when the user presses the <F1> key if no menu options corresponding to functions are highlighted. A POPUP menu item has a CONTEXT equal to 0 since no function is defined for POPUP.

The next TOPIC in this context sensitive file should be assigned the value MENU, corresponding to the first user-defined menu option. Subsequent TOPICs should be declared in the same order as the commands were declared in the .MNU file so that the automatically assigned context numbers will be in the correct order.

Example: If the .MNU file contains the following:

```
MYMENU MENU
BEGIN
    MENUEDIT
    POPUP "Menu &1"
    BEGIN
      MENUITEM "Do 1.&1.",Do11
      MENUITEM "Do 1.&2.",Do12
      MENUITEM "Do 1.&3.",Do13
      MENUITEM "&Directory setup", IDSETUP
    END
    POPUP "Menu &2"
    BEGIN
      MENUITEM "Do 2.&1.",Do21
      MENUITEM "Do 2.&2.",Do22
    END
END
```

the corresponding .TXT file used to compile the .HLP file should look like this:

```
.TOPIC General Help=DEFAULT
     Default help screen when <F1> is pressed and the cursor is
     not on an active menu item.
```

```
.TOPIC Do11=MENU
     Context sensitive help screen when the cursor rests on menu
     item Do 1.1.
.TOPIC Do12
     Context sensitive help screen when the cursor rests on Do 1.2.
     This topic is automatically given the context number 2001.
.TOPIC Do13
     Context sensitive help screen when the cursor rests on Do 1.3.
     This topic is automatically given the context number 2002.
.TOPIC Do21
     Context sensitive help screen when the cursor rests on Do 2.1.
     This topic is automatically given the context number 2003.
.TOPIC Do22
     Context sensitive help screen when the cursor rests on Do 2.2.
     This topic is automatically given the context number 2004.
```

Since IDSETUP is a built-in function, no topic is declared. However, the help screen provided for this command will be displayed if the file EIS_HELP.HLP is present in the same directory as the .EXE file.

Hypertext Files for Use With the HELP Command

The first topic in a hypertext "context-free" file is automatically given a Context value of GENERAL (=5000), but it is not necessary to do this explicitly, for example:

.TOPIC Contents

Subsequent TOPICs in the same file do not require Context values to be added.

BATCH, MESSAGE, and BRANCH topics can appear anywhere in the file. Each needs to be specified, for example:

.TOPIC List all topics
    <<Show one message>>
    <<Show another message>>
    <<Run an external program>>
    <<Branch to a new file>>
    <<A last message>>
.TOPIC Show one message=MESSAGE
This is one message
.TOPIC Show another message=MESSAGE
This is another message
.TOPIC Run an external program=BATCH
EXTPROG.EXE
.TOPIC Branch to a new file=BRANCH
NEWFILE.HLP
.TOPIC A last message=MESSAGE
This is the last message
Examine EPIGLUE.HPT for more examples.

Hypertext Designators (Nodes, or active points)

Syntax   <<{<Highlighted text>|} <Topicname>>>

The symbols "<<" and ">>" correspond to ASCII codes 174 and 175, respectively, on IBM-compatible personal computers. The vertical bar (ASCII 124) sometimes appears on the IBM keyboard as a broken vertical line. The curly brackets and single <> signs are notation, meaning "optional" and "select your own Topicname," respectively.

Cross references activate a specific block of the file when selected by clicking or pressing <Enter>. They should always refer to <Topicname>s defined by a .TOPIC statement, even though the .TOPIC statement may appear later in the file. The "Highlighted text" is highlighted and displayed on the screen, often surrounded by the other text in the same TOPIC. <Topicname> must match <Topicname> in a TOPIC in the same file. It may include spaces, but not other non-alphanumeric characters. If no highlighted text is included, the <Topicname> is displayed instead. An example follows:

```
.TOPIC INDEX
This is the first screen that pops up when the file is called. This
first screen is usually a list of Topics that allows navigating
through the various blocks of the file.
<<Option 1|OP1>>      <<Option 2|OP2>>      <<Option 3|OP3>>
.TOPIC OP1
This is the screen for option 1
Press <Enter> or click on "Main menu" to go back to the first
screen.
                         <<Main menu|INDEX>>
.TOPIC OP2
This is the screen for option 2
                         <<Main menu|INDEX>>
.TOPIC OP3
This is the screen for option 3
                         <<Main menu|INDEX>>
```

Highlighted passive text

~Highlighted text~

Text between Tildes (~) is highlighted. The alternative characters, ASCII #169 and ASCII #170 may also be used for compatibility with versions of HYPER.EXE prior to 6.01. The text is highlighted on the screen, but no action is carried out if an attempt is made to select it. This type of highlighting is for emphasis only.

Setting the Colors for Hypertext Displays

The colors in which hypertext is displayed can be changed by pressing the <Space

bar> or clicking on the "*" in the display. The display cycles through several different color "palettes." This is especially handy for finding a suitable display on a monochrome screen.

Display palettes can also be set permanently by inserting a parameter in the file that will be compiled to produce the .HLP file. A parameter -- /PALETTE CYAN, /PALETTE BLUE, or /PALETTE GRAY -- after ".TOPIC" sets the color of the default palette for the Help file in which it occurs. Only one such TOPIC is allowed per file. If a palette is specified, pressing the space bar or clicking on the "*" character does not change the screen color when the file is displayed.

Since this option inserts an extra byte in a file header for the compiled hypertext, files containing the PALETTE parameter are not compatible with versions of the menu (EPI6.EXE) prior to 6.03, unless compiled with the Version 6.03 HYPER.EXE compiler and the /OLD switch (See HYPER). In the following example,

.TOPIC /PALETTE BLUE

the entire hypertext file will be displayed with a blue background and the color will not be affected by the space bar.

Enabling/Disabling Wrapping of Text

When a hypertext viewer window is resized, the hypertext viewer wraps the text to fit the new size of the window. This works well for text in paragraphs, but can be very annoying for tabular data.

To prevent a line from being wrapped, place one or more spaces (a blank " ") at the beginning of the line. If you want the line to start with a space, and still wrap, enter an ASCII code 255 as the first character of the line. This will "look like" a space, but not deactivate the wrapping of text.

 Examples:

 .TOPIC Wrapped_Text
 This text will
 automatically
 be displayed
 on a single line
 if it fits in the window

 .TOPIC Single_Lines
   This text will

> **automatically**
> **be displayed**
> **as typed,**
> **regardless of the size of the window**

HYPER.EXE, the Hypertext Compiler for EPI6

The plain (ASCII) text file described in the previous section needs to be compiled before it can be called by either the <F1> key or the HELP function. This is achieved by typing:

 HYPER <File name>.TXT <File name>.HLP {/N} {/OLD}

in which <File name> is the name of the help file. The .TXT is a text file, in which key words and cross references have been embedded, as described above. HYPER will create an indexed file, with the necessary cross reference for hypertext, ready to use with EPI6 and other menus created with this system. Naming FILENAME.HLP is optional. If no destination help file is mentioned, the name of the source text file is used with a .HLP extension.

/N   Turns off screen output from HYPER.EXE so that compiling appears to take place "behind the scenes." It is used mainly in conjunction with a ROUTEFILE statement in a function block (see above).

/OLD Necessary only when a .HLP file is to be displayed in a version of the EPI6.EXE menu older than Version 6.03 *and* the file to be compiled contains a /PALETTE topic. It is recommended that the newer version of the menu be used so that the PALETTE can take effect, but this method of suppressing the PALETTE is provided so that compatibility can be maintained. The /OLD switch is not needed unless the file being compiled contains .TOPIC /PALETTE....

The HYPER program returns a code in the DOS variable ERRORLEVEL corresponding to its status upon termination. If the file was properly compiled, an ERRORLEVEL of 0 is returned. Otherwise, an ERRORLEVEL of 1 is returned. This can be tested in the command block of the menu in order to display an error message in an automated system:

 DoCompileReport
 BEGIN
    ANALYSIS MAKHYPER.PGM
    HYPER MAKHYPER.TXT
    IF ERRORLEVEL 1 PAUSE
    HELP MAKHYPER.HLP

END

This command block creates the hypertext layout file, MAKHYPER.TXT, by running a suitable .PGM program in ANALYSIS, then compiles MAKHYPER.TXT in HYPER.EXE. If a problem occurs during compilation, DOS will PAUSE to allow the user to read the error message returned by HYPER. Otherwise EPI6 displays the resulting hypertext file MAKHYPER.HLP directly.

If there is an error in compilation, HYPER displays an error message with the line number of the text file where the problem occurred. This happens when:

- A reference points to an undefined topic
- There is an incorrect TOPIC definition

Lists of TOPICs in .LST Files

When compiling a text file using HYPER.EXE to produce a .HLP file, a file with the same name as the .HLP file and a .LST extension is automatically created. It contains in plain text (ASCII) format, an index of the TOPICS. In the HELP display system in EPI6.EXE and similar menus, this index can be displayed in a picklist by pressing <Shift-F1> or clicking on the question mark "?" below the hypertext window. The user can choose a topic and display it immediately. If there is only one topic in the list, this feature is disabled.

BATCH and MESSAGE topics are not included in the .LST file nor are TOPICs beginning with the underscore "_" character. This provides a way of controlling what appears in the list.

The contents of the .LST file can also be used to construct a customized index in the actual hypertext file by judicious cutting and pasting.

Default size for hypertext windows

By default a hypertext window occupies the full screen. However, it can be set to a different value by editing the .BMK file with the same name as the hypertext file. This .BMK file has a SIZE statement, followed by the X and Y coordinate of the upper left corner and the lower right corner of the window. For example,

SIZE=0 0 80 23

This is the default value, for full screen.

SIZE=10 5 70 18

This size would leave 10 columns on the left and right of the window, and 5 lines above and below the window.

Hypertext limitations and troubleshooting

Starting with Version 6.03, a single line in a TOPIC may be up to 255 characters. The maximum size of any one TOPIC is 30 K. No error message appears if an attempt is made to load a block larger than this, but the computer will hang. Not more than 16000 topics can appear in a single help file. The size of the file is virtually limited only by disk storage, and .HLP files of several megabytes are routinely handled.

Printer Codes

Unfortunately commercial printers do not have a common set of control codes. Codes for printers are listed below for use in setup in EPED or in the EPIGLUE/EPI6 menu.

```
Epson FX, MX, or RX printers Master reset                           \027@
        Select compressed mode                                     \015
        Cancel compressed mode                                     \018
        Select 12 cpi                                              \027M
        Select 10 cpi                                              \027P
        Select expanded mode                                       \014
        Cancel expanded mode                                       \020
        Select letter quality mode                                 \027x1
        Select draft mode                                          \027x0
        Select proportional mode                                   \027p1
        Cancel proportional mode                                   \027p0
        Single space text                                          \0272
        Double space text                                          \027J24

Epson LQ1500 printer    Master reset\027@
        Select compressed mode                              \027x0\015
        Select expanded mode                                \027w\001
        Select letter quality mode                          \027x1
        Select 12 cpi                                       \027M
        Select 10 cpi                                       \027P
        Single space text                                   \0272
        Double space text                              \027\065\020
        8 lines per inch                                    \0270

HP LaserJet printers
Portrait Letter-size paper 60 lines per page  10 cpi               \027E
            16.66 cpi                                   \027(s16.66H
        66 lines per page  10 cpi                           \027&l7.27C
            16.66 cpi                     \026&l7.27C\027(s16.66H
Legal-size paper (legal tray) 78 lines per page  10 cpi           \027E
            16.66 cpi                                   \027(s16.66H
        104 lines per page 10 cpi                           \027&l8D
            16.66 cpi                        \026&l8D\027(s16.66H

Legal-size paper (manual tray)78 lines per page 10 cpi       \027&l84p2H
            16.66 cpi                     \027&l84p2H\027(s16.66H
         104 lines per page 10 cpi                        \027&l84p2h0o8D
            16.66 cpi                 \026&l84p2h0o8D\027(s16.66H
```

```
Landscape Letter-size paper 60 lines per page 10 cpi                    \027&l1O
                16.66 cpi                                  \027&l1O\027(s16.66H
            66 lines per page 10 cpi                              \027&l1o5.45C
                16.66 cpi                              \026&l1o5.45C\027(s16.66H
Legal-size paper (legal tray) 78 lines per page 10 cpi                 \027&l1O
                16.66 cpi                                  \027&l1O(s16.66H
           104 lines per page 10 cpi                             \027&l1o5.45C
                16.66 cpi                              \026&l1o5.45C\27(s16.66H
Legal-size paper (manual tray)78 lines per page  10 cpi        \027&l84p2h1O
                16.66 cpi                              \027&l84p2h1027(s16.66H
           104 lines per page 10 cpi                           \027&l84p2h1o5.45C
                16.66 cpi                      \026&l84p2h1o5.45C\027(s16.66H

HP ThinkJet printers    Select compressed mode                           \016
          Select expanded mode                                           \014
          Select emphasized mode                                        \027E
          8 lines per page                                              \0270

IBM 5182 Color printer, Master reset                                     \024
Color Jetprinter, Select letter quality                                \027I2
Graphics, and Proprinter  Select compressed mode                         \015
          Cancel compressed mode                                         \018
          Select expanded mode                                         \027W1
          Cancel expanded mode                                         \027W0
          Select 12 cpi                                             \027\058
          Select skip-over perforation                             \027N(n)
          Cancel skip-over perforation                                  \027O
          Single-space text                                            \0272
          Double-space text                                          \027A24
          8 lines per inch                                         \027\048

IBM QuietWriter printers,Select compressed mode         \015\027\073\000
 Model 1 and 2 Select expanded mode                           \027\087\001
          Double-space text                         \027\065\024\027\050
          8 lines per inch                                       \027\048

NEC 8023A printer Select compressed mode                              \027Q
          Select 12 cpi                                               \027E
          Select 10 cpi                                               \027N
          Double-space text                                    \027\092,T,48
          Single-space text                                           \027A
          8 liners per inch                                           \027B

Okidata Microline printer Master reset                           \027\024
          Select letter quality mode                            \027\049
          Select compressed mode                                     \029
          Select 12 cpi                                               \028
          Select 10 cpi                                               \030
          Double-space text                                \027\037\057\048
          Single-space text                                     \027\054
          8 lines per inch                                      \027\056

Okidata Pacemark printer Select letter quality mode             \027\055
          Select compressed mode                                \027\066
          Select 12 cpi                                         \027\065
          Select 10 cpi                                         \027\054
          Double-space text                                \027\037\057\048
          Single-space text                                     \027\052
          8 lines per inch                                      \027\053

Okidata Pacemark and Microline Select letter quality mode       \027\073\003
 IBM compatble     Select compressed mode                            \015
          Select 12 cpi                                         \027\058
          Select 10 cpi                                               \018
          Double-space text                                    \027A0\0272
          Single-space text                                \027A0\024\0272
          8 lines per inch                                      \027\048
          Select expanded mode                                  \027\087\049

Star Micronix Gemini printer Master reset                       \027\064
```

```
        Select compressed mode                                \015
        Select 12 cpi                                 \027\066\002
        Select 10 cpi                                         \018
        Double-space text                              \027065\024
        Single-space text                                    \0272
        8 lines per inch                                     \0270
        Select expanded mode                          \027\087\001

Texas Instrument 850, 855, Master reset                      \027@
 and 865 printers Select letter quality                      \027q
        Select compressed mode                               \027P
        Select 12 cpi                                        \027z
        Select 10 cpi                                        \027y
        Double-space text                             \027\028\049
        Single-space text                                    \0272
        8 lines per inch                                     \0270
        Select expanded mode                                 \027F

Toshiba P351, P1350, Master reset                        \027\0261
 and P1351 printers Select compressed mode               \027\091
        Select 12 cpi                              \027*1\027E10
        Double-space text                                 \027L16
        Select expanded mode                               \027!
```
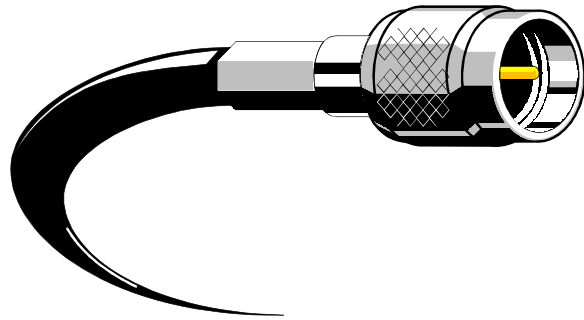
Chapter **38**

Epi Info and Local Area Networks (LANs)

What's a LAN?

A Local Area Network is a collection of computers connected to one another by wire or fiberoptic cables. Each computer forms a "node" on the LAN with it's own address and is connected through special hardware that provides a means of sending and receiving messages. Various ways of connecting the cables and transmitting messages are available. Two of the most popular are "Ethernet" and "Token Ring." The transmission protocols do not affect how the system looks to the computer user. Software to make the system work with more than one user and with various types of computers and other devices, called network software, is provided by many companies. Popular systems include "Novell" and "3Com."

A typical LAN has nodes that are microcomputers (also called "work stations"), printers, plotters, and file servers. A file server is a computer with one or more large hard disks, optical disks, or other storage devices accessible to network users.

On a DOS-based system, storage on the file server usually appears as additional "drives" on the user's computer (after logging in and giving the correct password). If you have floppy drives A and B and a hard disk C on your computer, you may log into the LAN and be able to access "drives" F through Z. One or more of these drives will probably be designated for shared use so that all users can both read and write files to it. Another drive or directory on a drive may be for your exclusive use for both reading and writing. The remaining drives will probably be designated "read only," meaning that you can examine their files or run programs from them, but cannot delete or alter the files unless you are the "network administrator."

Using Epi Info on a LAN as a Single User

Epi Info programs and other files can be copied to or installed in a LAN directory just as they are with a hard disk. On a LAN, usually only a designated Administrator has "read/write" privileges on the directories containing programs. Although it is possible for an individual user to install Epi Info in a "shared" or personal read/write directory, it is usually best to have the LAN Administrator install the programs and possibly provide menu access to make them easy to find. Individual users run the Epi Info that resides in a "read only" directory of the LAN, but keep data files on their own desktop hard disk or in a private "home" LAN directory in which they have "read/write" access.

What actually happens is that running a program causes it to be loaded from the file server disk to the Random Access Memory (RAM) on your own computer. To use Version 6 of ENTER in single-user mode, the "/SINGLE" parameter is used on the command-line or in the menu to run ENTER. If you and another user both try to enter data into the same .REC file, the first user will

obtain the file and have no problem. In most LANs, the second user will get a message from the LAN software saying that the file is not available, until the first user has closed the file. This protects against data corruption, in case both users have a portion of the same file in RAM, edit the contents, and then save the file to the LAN server, a situation in which it is impossible to predict which of the file pieces will overwrite the other. For the same reason, two users who plan to edit the same file should not copy it to their hard disks and then copy their edited copies back to the LAN server. As outlined below, there is a better way for several people to work with the same Epi Info data file, using Epi Info's LAN features.

Using Epi Info on a LAN in Multi-User Mode

The ENTER program in Epi Info allows several people to enter or edit data in the same file through "record locking." When one user accesses a record in the ENTER program, the record (but not the file) is locked so that other users cannot access the same record until it is relinquished by the first user. Because this could cause a problem if the first user went to lunch with a record still on the screen, the system actually does more complicated things to protect the integrity of the record, comparing the original record (on the LAN server) with a pre-edit copy of the record (in user RAM) to make sure that another user hasn't altered the same record. If this has happened, the second user will be notified and given a choice of reediting the record based on the updates by the other user, or of accepting the other user's changes. Since there is no problem in having several users enter new records, the likelihood of having two users edit the same record at the same time is small, and this protective mechanism causes little inconvenience.

ENTER will perform record locking on a LAN automatically unless this feature is turned off by running ENTER from the DOS command line with a /SINGLE parameter for "Single-user" mode. Most of the operations performed by ANALYSIS with existing data files involve reading, but not writing or correcting records. The UPDATE feature has been programmed so that it is safe for use on a LAN, and records will not be corrupted if more than one user tries to update the same file at the same time. EPED in this version does not allow two users to edit the same file at the same time. It does allow each user to have a configuration file (EPED.CFG) on his or her own hard disk, and thus to have EPED's configuration set differently from that of other users.

Information for LAN Administrators

The main programs in Epi Info are LAN compatible. ENTER allows multiple users to enter data in the same file, using record-locking techniques that work through DOS and are not LAN specific. ANALYSIS allows several users to read the same file and gives appropriate messages when trying to write files if this is not permitted.

A LAN or Wide Area Network (WAN) can be used as a distribution medium for Epi Info. To attach Epi Info to an Electronic-mail (E-mail) message, it is useful to package the files together as a single file. There are several commercial and shareware programs to do this, including PKZIP, LHARC, and ARJ. The "zipped" file can then be sent over the network to LAN Administrators for installation on each LAN.

We recommend that two installations of Epi Info be made on each LAN. Both are installed using the INSTALL program. The working version is installed with the "I" option for "Install." A second installation may be made in another directory with the "C" or "Copy" option. The second version contains the compressed files from the distribution diskettes, and is useful if network users wish to make floppy diskette copies of the system to take on field investigations or to give to others not on the network. The "Copy" option will prompt the user copying the compressed system to diskettes so that the appropriate files are put on each diskette regardless of diskette density or size (from 360 K upward). Users will need information about how to use the copy option to make their own installable diskette version of Epi Info, including the manual, and also where to obtain the printed manual.

Epi Info may be installed on Local Area Networks (LAN's) in several different configurations, depending upon the intended use:

1. The compressed installable system, containing INSTALL.EXE and FILES01.EXE, FILES02.EXE, etc., may be placed in a LAN directory to allow users to install the system on their hard disks (using the (I)nstall option in INSTALL) or copy it to floppy disks (using the (C)opy option) for distribution to others or installation on a laptop computer.

2. The executable version of Epi Info may be installed in a LAN directory for which most users do not have write privileges. It is important to use the INSTALL.EXE program to install the system. The main programs and examples will be installed in a directory called \EPI6, which may be made read-only to most users. Users should then run the system from another directory in which they have write privileges, either another LAN directory or their local hard disk drive. If the sample surveillance system is chosen during installation, a directory called \NETSS is also created. This directory MUST BE WRITABLE for users, or provision must be made to copy its contents to another directory for individual use.

3. The INSTALL program allows for installing Epi Info in subdirectories that are not directly subordinate to the root directory (e.g., in N:\APPS\EPI6 and N:\APPS\NETSS). For Epi Info to run properly with directories not directly off the root, a DOS environment variable called EIDIR must be set to tell Epi Info where to find its directories. In this case SET EIDIR=N:\APPS will do the job. See the SAMPLE below.

4. It is possible for a user to install and run Epi Info in a writable directory on the LAN without help from the LAN administrator, although others having access to the same directory can then alter the files.

Tutorials and examples require write privileges. Epi Info programs need to write config- uration files or save data files, and this cannot be done in a directory for which the user does not have read/write privileges. Although the system may reside in a directory for which the user does not have write privileges, the user must run it from a writable dir-ectory, either on the LAN (if users are to share a single data file), or on a local hard disk.

If tutorials or examples are run from the main menu while DOS is logged to an inappropriate directory a message now suggests using the SETUP function in the TUTORIAL or EXAMPLE menu. Depending upon circumstances, this either changes the logged directory to \EPI6 or offers to copy needed files from a read-only directory on the LAN so that Tutorials and Examples will run easily. The PATH statement in AUTOEXEC.BAT must contain the location of the Epi6 directory on the LAN. This allows Epi Info to be run from the LAN without being on the user's hard disk, although the latter is also a reasonable option if there is disk space available.

The programs should run faster if the system is copied to the user's own hard disk and run from there (perhaps accessing a common data file on the LAN). Programs residing only on the LAN must be loaded into RAM on the user's computer via the LAN, and this can slow things down in some cases.

Sample (Novell) LAN Installation

In a Novell LAN, suppose that a batch file is placed in a directory called M:\BATCH that is accessed when a user types EI6. The batch file contains the following commands:

```
@echo off
cls
set eidir=n:\apps              (Set an environment variable to tell Epi Info the
                                parent directory for \EPI6 and \NETSS)
map ins s6:=sys:/apps/epi6     (Set up mappings)
map ins s7:=sys:/apps/netss
call m:home.bat                (Be sure that the user is running system from a
                                directory  with write privileges)
call n:\apps\epi6\epi6.exe     (Run Epi Info)
map del s7:                    (Remove the mappings)
map del s6:
call m:home.bat                (Go back to home)
set eidir=                     (Erase the environment variable)
echo on
```

Although other LAN's may have different configurations or commands, perhaps this example will be helpful in setting up a smooth system for accessing Epi Info.

To run the sample surveillance system contained in the NETSS directory the user must be logged into the NETSS directory, and must have write privileges. There are several choices in setting up NETSS on a LAN, but one solution is as follows. In the menu configuration file EPI6.MNU, find the DONETSS block and delete "REM" from the beginning of the three lines where it occurs (5, 6, 7), and then add "REM" to the beginning of lines 3, 4, and 9 .

Chapter **39**

The Programmer's Toolkit and the REC2QES and MAKELIST Utilities

Overview

Participants in a workshop on "The Future of Microcomputers in Epidemiology," held in March, 1993, at the Centers for Disease Control and Prevention, stressed the need for an extendable software system that would manage data entry and provide basic analytic features, and also provide a framework on which others could build their own features. The Programmer's Toolkit is a beginning in that direction.

The Toolkit provides methods and source code for those working in the software development languages Turbo Pascal or C++ to write programs that work as an integral part of Epi Info. With the Toolkit a programmer can:

> •Write a Terminate-and-Stay-Resident (TSR) program that is called automatically from a field in a questionnaire. The TSR can calculate statistics, manage files, interact with a laboratory instrument, or do anything else that a TSR can be programmed to do. The results can be placed in other fields of the questionnaire.

> •Write a program that is called automatically from ANALYSIS and looks to the user like a new ANALYSIS command. The output from the program can be sent automatically to the ANALYSIS output window.

> •Write a standalone program that reads or writes Epi Info files. Object-oriented source code is provided in Turbo Pascal and in C++ for doing this. The programmer provides a user interface and other objects, but the code provided takes care of reading and writing the Epi Info files.

These procedures are for experienced Pascal or C++ programmers, and most users of Epi Info will have no need for the material in this chapter. It is provided for those who are themselves programmers or who may wish to work with programmers to extend Epi Info for special purposes.

Requirements for Calling a Turbo Pascal Program from ENTER

As described in Chapter 22, both Pascal and assembly language programs can be called from the ENTER program. The program that does the anthropometric calculations is

written in Turbo Pascal and will serve as an illustration of the methods involved. For complete details, please examine the files provided on the disk.

The files used in the anthropometry program are:

MEASURE.BAT	A batch file that loads the called program and runs ENTER
MEASURE.QES	The questionnaire
MEASURE.REC	The data file
MEASURE.CHK	The Check file, containing the instructions
	LOAD ENTFACE.BIN
	CALL ENTFACE 75
ENTFACE.ASM	Source code for the file, ENTFACE.BIN
ENTFACE.BIN	Allows ENTER to call Interrupt 75
ENTFACE.PAS	Provides routines for moving data between routines in MEASURE.PAS and the fields in the MEASURE questionnaire in ENTER
MEASURE.PAS	Performs anthropometric calculations and returns results to ENTER using routines in ENTFACE.PAS
DOMEAS.PAS	A TSR program that places ENTFACE.TPU and MEASURE.TPU routines in memory, ready to be called via Interrupt 75, and also sets up the interrupt vector to point to the proper routine in MEASURE.TPU.

How the Anthropometric Calculation Program Works

The Batch File, MEASURE.BAT

The batch file used to run ENTER with the MEASURE.REC file contains four commands:

> **MARK**
> **DOMEAS**
> **ENTER MEASURE**
> **REMOVE**

MARK.COM is a small program supplied with Epi Info that places a mark after the last TSR program in your system. Sidekick is an example of this kind of program. The mark allows the companion program, REMOVE.COM, to remove the TSR we will install, called DOMEAS, after use without damaging any TSR's you may have loaded previously. DOMEAS, as described below, is a TSR program that will be called from the MEASURE questionnaire as it is running in ENTER to perform anthropometric calculations.

If you would like to run anthropometric calculations frequently, you might construct a different batch file that would load the TSR program before running the EPI6 menu and remove it after EPI6 has ceased to run. A sample EPI.BAT might look like this:

MARK
DOMEAS
EPI6
REMOVE

The anthropometric program would be run from the main menu, choosing ENTER and the appropriate .REC file, such as MEASURE.

This much is simple, but from here on things get more complicated and the discussion is meant for programmers wishing to write their own TSR programs to use with ENTER. The example given is a Turbo Pascal program, but assembly language programmers should look at the file ENTFACE.ASM for more information. C programmers may want to develop a system similar to that used for Turbo Pascal.

The Binary Interface File, ENTFACE.BIN

The LOAD and CALL commands in the .CHK file cause the file ENTFACE.BIN to be loaded and executed. This small binary program in turn calls interrupt 75, assuming that the address of the program to be called has been stored there, and passes several pieces of information to the called program in appropriate memory locations. Assembly language programmers could write programs to do almost anything else, using the same conventions.

The binary file is produced by first assembling ENTFACE.ASM to produce ENTFACE.EXE and then using the DOS utility EXE2BIN to produce a binary file, with its entry point at location 0 in the file. The LOAD instruction in the .CHK file will load this program at the proper location and the CALL instruction will run it, passing the parameter(s) specified. ENTER sends the following information to the .BIN file:

   A pointer to the questionnaire's circular list header
   A pointer to the current node in the questionnaire
   An integer specified in the .CHK file to be passed

 The .BIN file sends back a value in register AX as follows:
   0 - No error
   1 - Make user enter field again

The Pascal Unit, ENTFACE.PAS

This is a Turbo Pascal (Version 5) unit that contains data definitions, procedures, and functions needed by any Turbo Pascal program that is to be called by ENTER.

Important data structures include:

The FieldList record, containing pointers to the previous and next members of the linked list that contains information on all fields in the questionnaire.

Within the FieldList record, the FieldRecord, giving access to the nearly 30 items of information maintained about each field in the questionnaire.

The procedures and functions include:

Function FindField (Header:FieldPtr; Field:String):FieldPtr;
> Returns a pointer to the FieldList for the field by the name given. If no such field exists, then a NIL pointer is returned.

Function InstallInterrupt (IntNo:Integer;ProcPtr:Pointer):Integer;
> Installs an interrupt of the IntNo given. The interrupt vector will be ProcPtr, the address of the entry point of your program.

Function GetString (Header:FieldPtr; QField:String): String;
> Returns a string from a field in the questionnaire.

Function GetNumber (Header:FieldPtr; QField:String): Float;
> Returns a number from a field in the questionnaire.

Procedure PutString (Header:FieldPtr; QField:String;S:String);
> Places a string, S, in the named questionnaire field.

Procedure PutNumber (Header:FieldPtr; QField:String; R:Float);
> Places a number, R, in the named questionnaire field.

Since these routines are declared in the Interface section of the Turbo Pascal Unit, they may be called from any program that USES the unit.

The Turbo Pascal TSR Program, DOMEAS.PAS

This program USES the MEASURE and ENTFACE units. It calls the procedure InstallInterrupt in ENTFACE.TPU to install the address of the routine MyDoScores in MEASURE.TPU as the vector for Interrupt 75, and then exits to DOS, leaving its own code still resident in memory in TSR fashion. When Enter calls ENTFACE.BIN, therefore, program control goes to the address at Interrupt 75, essentially calling the routine MyDoScores from ENTER.

The Anthropometric Calculation Unit, MEASURE.PAS

This is the unit you would use as a model in creating your own program. MEASURE.PAS contains two routines, JCS2ZS and MyDoScores. JCS2ZS performs the anthropometric calculations. It was developed by the Nutrition Division at the Centers for Disease Control and Prevention and is a close copy of Fortran routines that perform the same calculation. Essentially it compares the age, sex, height, and weight passed to it with the WHO growth reference curves and returns the results of the comparison as the Percentile, Z-score, and Percent of Median for Height-for-Age, Weight-for-Age, and Weight-for-Height.

MyDoScores takes AGE, SEX, HEIGHT, and WEIGHT from the screen entries, calls JCS2ZS to perform the calculations, and returns the results to the proper fields in the questionnaire on the screen. It illustrates how to access the data structures for each field of the questionnaire and how to use the "PUT" routines in ENTFACE.PAS. The GET routines are not called in MyDoScores, but they could have been used to get values for AGE, SEX, HEIGHT, and WEIGHT from the fields in the questionnaire.

Developing Your Own TSR Program for Use with ENTER

The easiest way to produce your own program is to make a copy of MEASURE.PAS by another name and then install your own procedures in place of JCS2ZS. MyDoScores can then be altered to GET or PUT strings or numbers from fields in your questionnaire or to otherwise alter the questionnaire screen. The only change needed in DOMEAS, other than renaming, is to change the name of your program at the place in the code where the program name is designated. More than one TSR can be present in memory at the same time, or you can design a single program and entry point for several functions.

By writing your own TSR routines to supplement the many operations that can be programmed directly in the .CHK file, you can produce programs that perform almost any desired function during the data entry process. Some possibilities include searching a database for information, performing complex error checks, putting messages on the screen (saving the underlying screen first, of course), and creating files in other formats. ENTER is happy to have your program dial Bangkok, receive or send information, and then put the results into the questionnaire, as long as you return a 0 for success and a 1 for "something went wrong." If something did go wrong, you will want your program to communicate the details, perhaps by putting a message on the screen, so that the user can take appropriate action, but this part is up to you.

CMD: The Programmer's Interface to ANALYSIS

The CMD command can be used to run an external program from within ANALYSIS and to return the results to ANALYSIS.

Syntax:

CMD <command name> { FILE= .EXE file name}{ TABLE} {NOPARAM}
       {MAXFIELDS=[max. number of fields]}
       {NUMERIC=[numeric field set]}

The TABLE parameters specifies that a TABLE is to be built with the variables specified on the command line. Two parameters modify the behavior of this command:

> MAXFIELDS specifies the maximum number of fields that will be built into a table.
> NUMERIC is a 16 bit number specifying the fields in the fields list which must be numeric. If Bit i of the NUMERIC parameter is 1, then variable i must be numeric. The default is NUMERIC=0, which specifies that any sort of variables are accepted in any position. Two common settings are NUMERIC=1, which specifies that the first variable must be numeric, and NUMERIC=65535, which specifies that all variables must be numeric.

If the external program is not "ANALYSIS aware," that is, it is just a regular .EXE file that you wish to make available to users from within ANALYSIS, use the NOPARAM option. An example would be making the STATCALC program accessible from within ANALYSIS.

FILE specifies which .EXE file the command is to execute. If no FILE is specified, ANALYSIS assumes the command name with the extension .EXE. By accessing the ANALYSIS command string that is passed in the parameter block, code for more than one command can be put into an .EXE file and the commands can be distinguished.

The CMD statement for a given command is usually placed in the CONFIG.EPI file so that the command appears transparent to the user. Some examples:

CMD REGRESS FILE=STATS.EXE TABLE MAXFIELDS=5 NUMERIC=65535
CMD MEANS FILE=STATS.EXE TABLE MAXFIELDS=2 NUMERIC=1
CMD CROSSTAB TABLE

CMD STATCALC NOPARAM
CMD DISPLAY FILE=C:\UTILITY\LIST.COM NOPARAM

Examples of using these commands:

DISPLAY OUTPUT.TXT
STATCALC
MEANS AGE SEX
REGRESS AGE HEIGHT WEIGHT
CROSSTAB VANILLA ILL SEX

If the NOPARAM option is specified for an external command, ANALYSIS simply does the equivalent of inserting the DOS command in front of the command line, passing the paramters to the command as the command line to the program. This is the trivial case, however.

The programming becomes more complex when a parameter block is built and passed to the program. ANALYSIS builds the table required (if the TABLE option was specified for the external command) and sets up the parameter block. The structure of the parameter block is as follows:

```
AnalysisParamBlock    = Record
                          AnalysisID     : String [8];
                          CStrPtr        : StringPtr;
                          OutFileName    : String80;
                          TableTree      : CrossPtr;
                          TableHeaders   : CrossArray;
                          TableNumFields : Integer;
                          TableFields    : FieldArray;
                          TableDiff      : IntArray
                      End (*AnalysisParamBlock*);
```

AnalysisID is a string of length 8 which can be checked to ensure that the address of the parameter block was correct. It is always set to 'ANALYSIS'.

CStrPtr is a pointer to the ANALYSIS command string. This is simply the full text (including the command name) of the parameter that the user typed in.

OutFileName is the name of the file from which ANALYSIS will retrieve any output from the command. The file will have already been erased, so a simple rewrite of the file will suffice.

The rest of the parameters only make sense if the TABLE option was specified.

TableTree is a pointer to the AVL-tree containing the table that was constructed. Each node has the key for that cell (formed by concatenating the table variables into a string), the count for that node, and Left and Right pointers to its children.

TableHeaders is an array of linked list headers specifying the unique keys that occurred in each dimension of the table.

The REC2QES Utility

The utility program, REC2QES.EXE, will construct a .QES file from a .REC file, reversing the process that occurs in ENTER when a .REC file is created. If the original .QES file is saved, this should never be necessary, but sometimes a .REC file is obtained from elsewhere without a .QES file or the original .QES file is lost. In this case, REC2QES can save the day by providing a .QES file for revision or for making a new .REC file.

To run REC2QES from the DOS prompt to recreate EPIDEMIC.QES from EPIDEMIC.REC, for example, type:

C:\EPI6>**REC2QES EPIDEMIC.REC**

REC2QES will create a file called EPIDEMIC.QES. If the original file contained complicated structures, such as fields without preceding text, REC2QES may not be able to do a complete reconstruction of the original .QES file, but it will provide a basis for further editing. Once the .QES file has been reconstructed, you can make further changes and create other .REC files or restructure the .REC file, using ENTER.

The MAKELIST Utility

The program MAKELIST.EXE makes lists of variables or values of variables in Epi Info files so that they can be placed in Picklists, DOS batch files or other programs. Please see the README file that accompanies Epi Info for more information on using MAKELIST.

Related Topics in Other Chapters

Chapter 23 describes the nutritional anthropometry programs from which many examples in this chapter stem. Additional material on the CMD command is found in Chapter 36. Chapter 7 describes how to restructure a .REC file from a revised .QES file.

Functional Index: Epidemic Investigation

Some of the possible steps in conducting an epidemiologic investigation are listed, in the order in which they might occur and references to pages containing relevant information are given. Advanced steps that may be optional are marked with "*".

Functional Index: Disease Surveillance

Some of the possible steps in setting up and operating a disease surveillance system are listed in the order in which they might occur, with references to pages containing relevant information.

Alphabetic Index

—D—

—T—

—Ÿ—

—Y—

—Z—

Field Types in Epi Info

—————

\<A\>
\<A   \>
Upper Case Text

#
##.#
Numeric

\<Y\>
Yes/No

\<mm/dd/yy\>
\<mm/dd\>
\<dd/mm/yy\>
\<dd/mm\>
Dates

\<phonenum\>
Local phone number

\<long distance\>
Long-distance phone number

\<today\>
**\<today/yy\>**
\<today/yyyy\>
Date of Last Change

\<Idnum\>
\<Id   \>
Automatically Incremented Identification Number
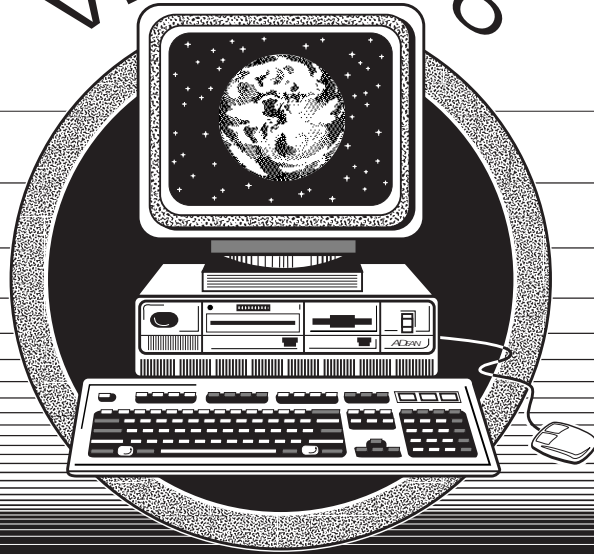
Note: The preceeding page is the back cover. The front cover was produced in Aldus Freehand on the Macintosh, and its quality will depend on the availability of the necessary fonts in a particular computer system.