EpiInfo
2000

Epi Info 2000

A Database, and Statistics Program
for Public Health Professionals
for use on Windows 95, 98, and NT Computers

Program design by Andrew G. Dean, MD, MPH,
and Thomas G. Arner (Epi Map)

Programming by
Godha Sunki , Sireesha Sangam, Thomas G. Arner,
Roger Friedman, Matthew Lantinga,
Andrew G. Dean, and Saliil Diskalkar

Testing by
Juan Carlos Zubieta, MD, Cassandra Brown,
Natalie Huet, and Uliecia Jones

Manual by
Andrew G. Dean, Kevin M. Sullivan, and Juan Zubieta

The Division of Surveillance and Informatics
Epidemiology Program Office
Centers for Disease Control and Prevention (CDC)
Atlanta, Georgia 30341

Previous versions produced in collaboration with
World Health Organization (WHO)
Geneva, Switzerland, 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.

Suggested citation: Dean AG, Sullivan K, Arner TG, Sangam S, Sunki G, Friedman R, Lantinga M, Zubieta J, Smith DC. Epi Info 2000, a database and statistics program for public health professionals for use on Windows 95, 98, and NT computers. Centers for Disease Control and Prevention, Atlanta, Georgia, USA, 2000.

Epi Info Hotline for Technical Assistance

epiinfo@cdc.gov

(770) 488-8440

FAX (770) 488-8456

Acknowledgements

Environmental Sciences Research Institute, Inc., the publishers of ArcView and ArcInfo, provided a special waiver for royalty-free distribution of the MapObjects mapping engine contained in Epi Map 2000, on condition that Epi Map 2000 is distributed only with Epi Info 2000. Robert K. Jung, ARJ Software, Norwood, Massachusetts, gave permission for use of ARJ compression programs used in distributing translations. Dr. David Martin, Brookline, Massachusetts, and A. Ray Simons, Atlanta, Georgia, provided Turbo Pascal procedures for exact confidence limits. Mr. Jan Verhoeven (http://jans/hypermart.net) gave permission to distribute his freeware program, Andante4, with Epi Info on CD-ROM. The programs for logistic regression and Kaplan-Meier survival analysis are based on the MULTLR and KMSURV programs of Eduardo Franco, PhD, and Nelson Campos-Filho with their kind permission and collaboration. Consuelo M. Beck-Sagué, MD, developed the teaching exercise on Rhodococcal infection. We thank the many beta testers who provided feedback and suggestions.

Notes

These programs are provided for use by the public health community. Please give copies of the programs and the manual to friends and colleagues. The programs may be freely translated, copied, distributed, or even sold without restriction except as noted below. No warranty is made or implied for use of the software for any particular purpose.

Distribution

"Epi Info" is a trademark of the Centers for Disease Control and Prevention (CDC). Please observe the following requests:

1. When distributing more than 10 copies of Epi Info, please report to the Epi Info Help Line the number of copies distributed, the countries to or in which they were distributed, and whether the distribution was by Internet or CD-ROM.
2. The programs can be translated and the examples altered for regional use, but the programs must be distributed in essentially the form supplied by CDC. Epi Map 2000 cannot be distributed separately from Epi Info 2000.

Epi Info is written in Visual Basic, Version 6, from Microsoft, Inc. The manual was written and indexed in Microsoft Word. The cover art is by Pete Seidel, Public Health Practice Program Office, CDC, and Dr. John Snow, London. 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.

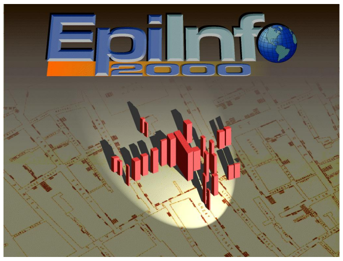
Technical Support

For new versions of the software and answers to commonly asked questions, please visit the Epi Info web site at www.cdc.gov/epo/epi/epiinfo.htm. Technical assistance by e-mail, telephone, or FAX is provided. Information for obtaining Epi Info technical assistance is given on the title page.

The Epi Info Worldwide Discussion Group (LISTSERV) provides a forum for user questions and answers. Join the group by sending an e-mail to *lists@list.cdc.gov*, with the words *Subscribe epi-info* in the body of the text. A message containing further instructions will be sent in response.

Please send **comments and suggestions** for future versions to:
Andrew G. Dean, M.D., M.P.H.
Epidemiology Program Office, Mailstop K74
Centers for Disease Control and Prevention
Atlanta, GA 30341-3717
agd1@cdc.gov
Telephone (770) 488-8432
Home Telephone (770) 458-2271
(Emergencies only please)

Overview



What is Epi Info 2000?

Epi Info is a series of Windows 95, 98, and NT programs for use by public health professionals in conducting outbreak investigations, managing databases for public health surveillance and other tasks, and general database and statistics applications. With Epi Info and a personal computer, physicians, epidemiologists, and other public health and medical workers can rapidly develop a questionnaire or form, customize the data entry process, and enter and analyze data. Epidemiologic statistics, graphs, and tables are produced with simple commands like READ, FREQ, LIST, TABLES, and BAR. A component called Epi Map displays geographic maps with data from Epi Info. Epi Info is in the public domain and can be downloaded from the Internet. CD-ROM copies and printed manuals may also be available from private vendors.

A study in 1997 documented 145,000 copies of the DOS versions of Epi Info and Epi Map in 117 countries. The DOS manual and/or programs have been translated into 13 non-English languages.

Epi Info 2000 is an entirely new series of programs for Microsoft Windows 95, 98, and NT, written in Visual Basic, Version 6. It uses the Microsoft Access file format as a gateway to industry database standards. Although Epi Info 2000 data is stored in Microsoft Access files for maximum compatibility with other systems, many other file types can be analyzed, imported, or exported.

Epi Info 2000 includes a Geographic Information System (GIS), called Epi Map 2000, built around the MapObjects program from Environmental Sciences Research Incorporated, the producers of ArcView. Epi Map is compatible with GIS data from numerous Internet sites in the popular ESRI formats.

Epi Info 2000 retains many features of the familiar Epi Info for DOS, while offering Windows strengths like point-and-click ease of use, graphics, fonts, and painless printing. The programs, documentation, and teaching materials are in the public domain (although “Epi Info” is a CDC trademark), and may be freely copied, distributed, or translated.

Key Features of Epi Info 2000

- Maximum compatibility with industry standards
- Extensibility, so that centers outside CDC can produce additional modules
- Epi Map, an ArcView-compatible GIS
- Logistic regression and Kaplan-Meier survival analysis
- Teaching exercises
- Ease of use

- Entirely new, not just a "port" of Epi Info for DOS.

System Requirements

- Windows 95, 98, or NT, with 32 megabytes of RAM
- A 200-megahertz processor is recommended but not required.

Planning an Upgrade to Epi Info 2000 from Epi Info for DOS

- Windows 3.1x users should continue to use Epi Info 6.04. The Version 6.04 "b-to-c update" uses 4-digit years for year-2000 compatibility
- Logistic regression and Kaplan-Meier survival analysis can be used directly with Epi Info 6.xx files
- Conversion programs are provided for moving data files from DOS versions of Epi Info to the Windows version
- For permanent systems, some reprogramming of program (PGM) and Check (CHK) files is necessary.

Technical Support and Further Information

CDC provides funding for the Epi Info Hotline, offering free technical support to Epi Info users during normal East Coast working hours (8 AM – 5 PM, U.S. Eastern Standard or Daylight Savings Time). See the title pages of this manual for phone and FAX numbers and e-mail address.

The Epi Info Worldwide Discussion Group

CDC maintains a LISTSERVer for users of Epi Info. Users who send an e-mail to subscribe to the List will receive an email with instructions in return, and then will automatically receive messages submitted to the List by other users. Typically, a request for advice or opinions on a question related to Epi Info will be answered by several messages from other users. The List is an excellent way to stay in touch with other users. The Epi Info Development Team participates in the List, and announcements of upgrades, bug fixes, or other events are sent via the List. It serves as an important forum for guiding future development of Epi Info.

Downloading Epi Info or Related Programs from the Internet

From the CDC home page at www.cdc.gov, choose "Publications, Products, and Software," then "Software," and "Download Epi Info." Click on the Download choice to see further instructions, or browse around the Epi Info pages to learn more about the programs. Epi Info and related materials are available on other Web sites. A search for "Epi Info" with one of the many Web search engines will provide access to these sites.

References

Burton AH, Dean JA, Dean AG. Software for data management and analysis in epidemiology. World Health Forum 1990; 11:75-77.

Dean AG, Dean JA, Coulombier D, Brendel KA, Smith DC, Burton AH, Dicker RC, Sullivan K, Fagan RF, Arner TB. Epi Info, Version 6.04a, a word processing, database, and statistics program for public health on IBM-compatible microcomputers. Atlanta: Centers for Disease Control and Prevention; July 1996.

Dean AG, Gerstman BB. Computing and epidemiology. In: Gerstman BB. Epidemiology kept simple. New York: John Wiley; 1998. p. 275-288.

Dean AG, Shah SP, Churchill J. DoEpi: Computer-assisted instruction in epidemiology and computing and a framework for creating new exercises. Am J Preventive Medicine 1998; 14(4):367-371.

Dean AG. A course in microcomputer use for epidemiologists and others who count things, using Epi Info. Atlanta: Centers for Disease Control and Prevention, Epidemiology Program Office; 1994.

Dean AG. Using a microcomputer for field investigation. In: Gregg, MB. Field epidemiology. New York and London: Oxford University Press; 1996. Chapter 12, p. 164-180.

Dean AG. Epi Info and Epi Map: Current status and plans for Epi Info 2000. J Pub Health Management and Practice 1999; 5(4): 54-57

Dean AG. Microcomputers and the future of epidemiology. Public Health Reports 1994;109(3):439-41.

Dean AG. EPIAID. Byte Magazine 1985 October; 10:225.

Dean AG, Dean JA, Burton AH, Dicker RC. Epi Info: A general purpose microcomputer program for health information systems. Am J Preventive Medicine 1991; 7:178-182.

Dean AG, Fagan RF, Panter-Connah B. Computerizing public health surveillance systems. In: Teutsch SM and Churchill RE. eds. Principles and Practice of Public Health Surveillance. New York: Oxford University Press; 1994. Chapter 11, p. 200-217.

Dean JA, Burton AH, Dean AG, Brendel KA. Epi Map 2.0: A mapping program for IBM-compatible microcomputers. Atlanta: Centers for Disease Control and Prevention; 1996.

Epidemiology Program Office. Epi Info users' manual, version 2. Atlanta: Centers for Disease Control; 1987.

Epidemiology Program Office. Epi Info users' manual, version 3. Atlanta: Centers for Disease Control; 1988.

Epidemiology Program Office. Epi Info users' manual, version 5. Atlanta: Centers for Disease Control; 1990.

Fegan G, Dean AG, Shah SP. Epi Info/Epi Map Internet Site. World Wide Web pages (approximately 60) on the Internet. Epidemiology Program Office, Centers for Disease Control and Prevention, 1996. (http://www.cdc.gov/epo/epi/epiinfo.htm)

Harbage B, Dean AG. Distribution of Epi Info software: An evaluation using the Internet. Am J Preventive Medicine 1999; 16(4): 314-317.

Stroup DF, Williamson GD, Dean AG, et al. Statistical software for public health surveillance (SSS1) (software manual).   Atlanta: Centers for Disease Control and Prevention, Epidemiology Program Office; 1994.

What's New?

For Users of Epi Info, Version 6, for DOS

Windows™ 95, 98, and NT

The Windows operating systems provide the graphical interface, point-and-click operation, and other features long familiar to Macintosh users. A variety of fonts, graphics programs, and word processors provide the opportunity to produce publishable output in many forms, and to focus on results rather than on programming or processing. When things go well, Windows programs are generally easier to learn and use than are DOS programs. When things are not going well, the additional complexity of Windows programs can make it hard to correct problems, and some users with well-developed typing skills may still prefer DOS.

Windows offers a number of mechanisms for interaction, such as left mouse clicks to activate a control and right mouse clicks to bring up a menu of choices. The familiar FILE, EDIT, and VIEW menus are found in most Windows programs, and serve to make a Windows user feel at home in a new program almost at once. Since users of Epi Info for DOS have already acquired reflexes that do not always match the Windows way of doing business, we have tried to compromise with Windows standards in some cases, to make Epi Info 2000 as familiar as possible to previous Epi Info users. In ***Analysis***, for example, commands all have dialog boxes designed to produce action by making choices and clicking on an OK button. At the same time, however, program statements in plain text are produced and placed in a program editor so that they can be viewed, edited, or run as PGM files. A user can also type the commands directly in the program editor, although the syntax is somewhat more complicated than in Epi Info for DOS because Windows offers more choices.

Visual Basic

Epi Info 2000 programs were developed in Microsoft Visual Basic, Version 6. Visual Basic is a computer language used by four million programmers worldwide. Since it uses the Microsoft C++ compiler behind the scenes, the resulting programs are as fast and responsive as those produced in any language. Epi Info users do not need to know or use Visual Basic, but advanced users can produce add-on modules for Epi Info using Visual Basic or other popular programming languages. Because Visual Basic is frequently upgraded by Microsoft Corporation to keep up with changes in Windows systems, Epi Info 2000 will be able to take advantage of new developments in computer technology as they emerge.

Microsoft Access File Format

Databases in Epi Info 2000 are Microsoft Access (MDB) files, and therefore compatible with the most popular Windows database format. Epi 2000 uses the Microsoft Jet Engine (Version 3.5) to manage databases and to read and write a variety of other file formats such as dBASE, Paradox, ODBC, text files, and even HTML tables. Files from Epi Info 6.xx can be imported or exported, but are not read or written directly by the Jet Engine.

Questionnaire Features

Questionnaires in Epi Info 2000 are called "Views." As in previous versions, a database is constructed by developing screen representations of forms or questionnaires from which Epi Info constructs the database. A program called ***MakeView*** is provided for developing Views/Questionnaires. Development of a question or data entry item starts with a right mouse click on the screen. This brings up a dialog in which the user can enter the prompt or question and the type of field desired, and choose from a number of properties that, in previous versions, required writing Check code. Legal values, automatic coding, and comment legal values can all be set up from this field dialog.

New features in Epi Info 2000 include multiline fields capable of receiving very large amounts of text, option buttons, check boxes, image fields, and the ability to embed a grid or table in a View. The grid offers an automatic method of creating a related file for which Epi Info 2000 creates and maintains appropriate keys automatically. Grids are particularly useful for repeating values in a questionnaire—age, sex, and immunization status of a number of children on a household form, for example.

Views can have many pages. Each page can have a background image and/or color, and can be assigned an optional name. Moving from page to page is done automatically, by clicking a page name or number in a visible list, or with the PgUp and PgDn keys.

Related views are represented as buttons on their parent views. Clicking on the button brings up the View for the related table. Conditions can be specified so that the button for a related view is only available when appropriate. A hepatitis form might only be accessible if the Disease field has the value "Hepatitis," for example.

Check code can be inserted in any field to implement IF statements, ASSIGN values to variables, GOTO another field, enforce particular rules, or perform calculations during entry or exit from the field.

Data Entry Features

During data entry, field properties such as REPEAT, READ ONLY, and REQUIRED entry are enforced. Legal and coded fields have dropdown lists of values. Multiline fields scroll automatically when filled with text, and plain text fields scroll to hold up to 128 characters.

Data values entered on a page are saved automatically when the page is completed. First, last, previous, or next records are accessed by pressing left and right arrow keys. For compatibility with Epi Info for DOS, the F7 key moves to the previous record and F8 to the next.

The values for Yes/No fields can be set to any desired screen representation to allow for use in non-English languages, although the underlying values are stored as 0 for “No” and 1 for “Yes” so that data files can be exchanged internationally. The Options entry on the menu can be used to set the values of Yes and No.

Internet Features

The Internet offers a standard page description language (HTML) and two major systems for displaying and accessing documents—the Netscape and Microsoft browsers. To provide for future advancements and prepare the way for even more Internet compatibility, Epi Info 2000 produces output in HTML format. Lists, frequencies, tables, and other results from ***Analysis*** are in HTML, compatible with either major browser. Because there are passionate adherents of one browser or the other, or of no browser at all, Epi Info 2000 provides a freeware, off-line browser called Andante4 to display results. Andante4 is small and uncomplicated, and does not access the Internet, but will run an Internet browser automatically when needed if one is installed in the computer. It does not, however, violate any institutional rules against having an Internet browser installed.

The Epi Info 2000 menu can be programmed with the EXECUTE command to access Internet sites automatically, if an Internet browser is present in the computer. EXECUTE www.cdc.gov\epo\epi\epiinfo.htm, for example, will bring up the Epi Info home page on the CDC web site from a user-configured menu.

Logistic Regression and Kaplan-Meier Survival Analysis

Two Windows programs written in C++ are provided to do logistic regression and Kaplan-Meier survival analysis. “MVAWin” and “KMWin” are patterned after MULTLR and KMSURV, DOS programs from Dr. Eduardo Franco and Nelson Campos-Filho. The Windows versions read Epi Info 6 files. When used with Epi Info 2000, the necessary files are produced automatically by ***Analysis***.

Both conditional and unconditional logistic regression are offered, and KMWin provides several graphs of survival curves. Settings can be altered within the programs and saved for future use.

Geographic Information System (GIS)

An entirely new version of ***Epi Map*** is included with Epi Info 2000. The core of the program is the same mapping engine that is used in ArcView, a popular Geographic Information System from Environmental Sciences Research Institute (ESRI), Inc. Although ***Epi Map*** does not have all the features of the high-end commercial programs, ArcView and ARC/INFO, it is capable of reading many of the same file formats, allowing users to tap the enormous mapping resources that are offered on the Internet as SHAPE (.SHP) files. A catalog of resources is included with Epi Info 2000 to illustrate how these can be found.

Epi Map 2000 can display SHAPE files in multiple layers. Each layer can be linked to an Epi Info 2000 data table containing geographic names or codes for the entities of the map. Data for each entity (a count or rate, for example) can be displayed in color/pattern (choropleth) maps as in ***Epi Map*** for DOS, or as dots randomly distributed within the map's polygons.

New features in ***Epi Map 2000*** allow displaying streets and placing symbols by coordinates. Thus, one can produce a map similar to John Snow's original map of cholera cases and water-pump locations in London.

Epi Map 2000 also allows images to be displayed, and will allow images with coordinates available from the Internet to be mixed with the traditional polygon structure. Hence one can overlay a polygon or street map on a satellite image or aerial photo.

Flexible Graphing

The GRAPH command in ***Analysis*** offers many types of charts for displaying the values of one or more fields in a data table. A toolbar within the graphing module can be activated to allow customization of the resulting graphs. Settings can be saved as templates and used again from ***Analysis***.

A Standard Interface for Statistical Modules

Statistical modules as well as the mapping and graphing programs communicate with other Epi Info programs via a standard interface called the "Broad Street Library of Statistics" or IEPI interface. Methods are specified for sending data to the statistical programs and for receiving the results of calculations in the form of HTML pages or as arrays giving the names of parameters and their values. It should be possible for anyone familiar with Visual Basic to write a new statistical procedure to be used with ***Analysis***.

Interactive Teaching Exercises

Teaching exercises are provided in HTML format and are available from the main Epi Info 2000 menu under Tutorials. The Oswego foodborne exercise has been expanded to include both data entry and analytic components. The Rhodococcal Infection exercise provides roughly the same components in a hospital outbreak setting. A surveillance exercise provides a basis for planning computerization of a surveillance system. It can be used for classroom discussion and does not have hands-on computer components.

A new exercise, Simulation of Public Health in the Year 2012, provides discussion material for classroom or individual use concerning future developments in computerization of public health systems. A pictorial history of Epi Info can be used as an aid to teaching or understanding Epi Info.

Translation Features

Translation of Epi Info programs, help files, and exercises into non-English languages is made by placing all English phrases used in the programs in a database called LANGUAGE.MDB, located in the ENGLISH subdirectory under the main directory of Epi Info 2000 (usually \EPI2000). Help files and exercises are also contained in the ENGLISH directory or its subdirectories. Translated phrases, help files, and exercises are placed in similar directories named after the relevant language, such as SPANISH. Hence, translation can be done without changing the names of files, and individual translations can be installed or uninstalled without affecting the main programs. Switching from one language to another can be done from the main menu, and is mediated through setting the LANGUAGE variable in the file EPIINFO.INI.

Converting Epi Info 6.xx Systems to Epi Info 2000

When is it Wise to Convert?

New investigations, surveys, and studies can be done in Epi Info 2000 immediately after its final release. As with any new program, it is a good idea to question any unusual findings and to check them against those from other programs. Any difficulties should be reported to the Epi Info Hotline (see title page of this manual for phone number and e-mail address).

Approaches to Conversion

For systems already developed in Epi Info for DOS and working satisfactorily, it may be practical to retain the previous programs and start by converting the menus in the program to Epi 2000 format. Because the Epi Info 2000 ***Menu*** will read MNU files from Epi Info 6 with few or no changes, this is a quick way to produce better looking programs with very little effort.

Data (.REC) Files

Converting Epi Info 6 data files and screen formats to Epi Info 2000 is done with the IMPORT entry on the FILE menu of the ***MakeView*** program.

Check (.CHK) Files

Check code (.CHK files) is not converted automatically. Many commands in Epi Info 6 Check code (e.g., RANGE, Read-only status) have become properties that can be selected while creating a field. Making a field READ ONLY, or REPEAT, for example, is done by checking a box in the dialog for that field rather than by writing a command. The DIALOG command allows construction of dialogs for user interaction that should remove the need for some complicated methods that clever users employed to work around this problem in Epi Info 6. Although the syntax of Epi Info 2000 Check code is somewhat different from that of Epi Info 6, most of the commands are quite similar, and conversion should start by cutting and pasting blocks of Check code into the same blocks in the program editor of Epi Info 2000. Errors that occur on running the code can then be corrected by changing the syntax as specified under individual commands in the COMMANDS section of the manual.

Analysis (.PGM) Programs

Analysis programs are generally similar in the DOS and Windows versions of Epi Info, but there are important differences that require changes. One difference is that Epi Info 2000 databases are no longer simple files, but consist of tables within MDB files, and changes in syntax, for example, allow specification of both the MDB and the table name in the READ command. Since each command has a dialog box that generates the command after information is supplied, it should relatively easy to get the syntax right.

PGM “files” are not stored as files, but in tables inside an MDB—usually the same MDB where the data reside. This makes a neat package of data and programs that can be copied to another system or sent by e-mail for use elsewhere. PGM files from Epi Info 6 can be copied into the program editor in ***Analysis*** of Epi Info 2000 by placing them on the clipboard from the Epi Info 2000 word processor or from another Windows program such as Notepad or Microsoft Word, and then pasting them into the program editor.

Systems with Related Files

(XXX to be completed when features are stable)

Helping Epi 6 and Epi 2000 Systems to Coexist

Epi Info 6 files can be imported using ***MakeView*** and exported from ***Analysis*** using the WRITE command, but it is not possible to enter data directly into Epi Info 6 files from Epi Info 2000. Hence, a one-time conversion from one system to the other is easiest to maintain. If a single file is needed for analysis by both systems, dBASE files could be used, since they can be read directly by ***Analysis*** in both Epi Info 6 and Epi Info 2000.

Gradual, piecemeal conversion of previous systems can be done, linking both Epi Info 6 and Epi Info 2000 programs to the same menu, as long as the data entry program for the correct program is used for entry into the corresponding DOS or Windows files.

The Epi Info 2000 Programs

An Overview of Program Functions

Using Epi Info 2000

Epi Info 2000 is a database and statistics program for public health professionals. Although it can be programmed to produce systems for repeated or permanent use, it can also be used interactively for rapid questionnaire design, data entry, and analysis during an investigation.

Epi Info 2000 can be downloaded from the Internet or obtained on CD-ROM from a variety of sources. Using the programs requires that they be installed on a computer or network under Windows 95, 98, or NT, using the ***Setup*** program provided. See the Guided Tour chapter for specific installation instructions.

This chapter provides a description of the capabilities of each of the Epi Info programs. The Guided Tour chapter gives examples of each of the major functions. It is recommended that you follow the instructions on the Guided Tour to become familiar with Epi Info 2000 whether you are a new user of Epi Info or have experience with the DOS versions of the programs, since many of the features of Epi Info 2000 are different from those of Epi Info, Version 6, for DOS.

Setup: The Installation Program

Windows programs require a special installation program that analyzes the type of Windows software already on a particular computer, copies files onto the hard disk, and then registers modules that require registration. According to the usual practice, this program is called SETUP.EXE in Epi Info 2000. It offers choices of various modules to install and locations for installation. After installation, a second program offers to install available translations so that the programs can operate in languages other than English.

During installation, a list of steps called INSTALL.LOG is created so that the UNINSTAL program can be used to remove Epi Info 2000 from the computer, if desired. Since UNINSTAL removes programs that may have been placed in COMMON or SYSTEM directories and cleans up the system registry, it should always be used for uninstalling rather than merely deleting files from the hard disk. UNINSTALL will not remove files that are created after installation, such as database files, but they should be created with names other than the SAMPLE.MDB and NUTRI.MDB that are supplied with Epi Info, as these files will be removed, or copied over, during installation.

After installation, a program called TSETUP is available from the Epi Info 2000 menu under LANGUAGE | INSTALL TRANSLATION. Running this program allows you to install files necessary for running Epi Info 2000 in a language other than English. An example of the Spanish translation is provided. Those wishing to provide a translation should consult the chapter on Translations for further instructions.

A guided tour of Installation using ***Setup*** is provided.

The Epi Info 2000 Menu: A General Purpose, Configurable Menu

The menu provides access to the other Epi Info programs, but is also a general purpose, configurable menu that can be used in Windows 95, 98, and NT, to offer menus and command buttons that lead to other programs, functions, images, and text to create a theme. By choosing menu items or buttons on a menu, the user can run other programs or any function for which a Windows or DOS program is available. A command called EXECUTE can be used to allow Windows to determine from the file extension (.HTM or .TXT, for example) which program should be used to display the file. Because the menu can run DOS batch file commands, a series of operations, such as copying files for backup or transmission to another destination, can be performed from a menu choice or button.

The menu items and commands to be run are determined by a text file with the extension .MNU that can be created or edited in ***Eped***, the Epi Info Word Processor, or in the Notepad or WordPad programs that come with Windows. Menu files can also be created in Microsoft Word or Corel WordPerfect, but be sure to save the file as a DOS text file and confirm that it has only the .MNU extension.

More than one menu can be created to form a series of connected menus. Each menu can have a different image, set of menu items and corresponding functions, and a language translation database for easy implementation in non-English languages. The global variable LANGUAGE can be set from a menu choice to indicate which translation tables should be used by both the menu and the other Epi Info programs. The SETTINGS menu item can be used to specify a Working Directory; the computer is logged into the specified directory before each program.

A guided tour of the ***Menu*** program is provided.

MakeView: The Questionnaire and Form Designer

Questionnaires and forms in Epi Info 2000 are called Views. The ***MakeView*** program is used to place prompts and data entry fields on one or many pages of a View. Right-clicking the mouse establishes the location of a field. The prompt or question, type of field, and other details are then entered in a dialog box. Special features such as legal values, codes, ranges, repeat fields, and read-only status are specified in this field dialog. The field can be moved by left-clicking and dragging the prompt, and its size can be changed by clicking on the field to bring up sizing bars.

Data or field types are available for text, numbers, dates, yes/no, and other types of data. Special field types include multiline text fields that can hold large amounts of text; grids or tables that automatically result in related files for repeating groups of values, such as data on children in a household questionnaire; and images.

Questionnaires can have as many pages as desired; additional data tables are created and linked automatically if the number of fields exceeds the limit of 255 imposed by the Microsoft Access file format. Images can be placed in the background of a View page, and a background color can be specified. The font and size for each field can be set separately.

Check code instructions can be inserted for any field to do calculations, perform quality control, or provide advice to the user. Temporary variables can be defined and assigned values from fields on the screen or other defined variables. IF statements allow conditional operations, such as skipping part of the questionnaire, under specified conditions. Check code can be executed either as the cursor enters a field or as it leaves the field after data entry.

Related Views can be constructed. These are linked to a parent view automatically by unique keys generated by the system. A related view is made accessible through an Internet-style button; the button can be designed so that it is available only under specified conditions—when additional information is needed about a particular disease condition, for example.

Functions are provided for importing files from Epi Info for DOS, for aligning fields, and for placing a layout grid on the screen. Logical groups of fields can be chosen for display on panels so that, for example, Symptoms are displayed as a group and can be referred to by the group name in ***Analysis***. In ***Analysis***, the command FREQ SYMPTOMS would then produce frequencies of all the fields displayed in the Symptoms group.

A guided tour of ***MakeView*** is provided.

Enter: The Data Entry Program for Windows

The ***Enter*** program displays the View constructed in ***MakeView***, constructs a data table if necessary, and controls the data entry process, using the settings and Check code specified in ***MakeView***. The cursor moves from field to field and from page to page automatically, saving data as necessary. Buttons provide access to new, previous, next, first, and last records, and to related tables.

A menu item allows the user to set the values for display of Yes and No. The values actually stored in data tables are 0 for No, 1 for Yes, and 2 for Unknown, but ***Enter*** and ***Analysis*** can display any equivalents, such as "True" and "False" or "Si" and "No" that the user wishes to provide.

When related files are accessed with their corresponding buttons in ***Enter***, the main form can always be brought up with the HOME button and the parent of the current form can be reached with the BACK button. Navigation thus resembles that for web pages on the Internet as much as possible.

A guided tour of ***Enter*** is provided.

Analysis: Cleaning, Transforming, and Analyzing Data to Produce Tables, Maps, and Graphs

The ***Analysis*** program provides access to existing data either directly or through Views. It is able to read data from files and tables created in Epi Info, Microsoft Access, FoxPro, dBASE, Paradox, and any other format for which an "ODBC" driver is available. Since SAS, Oracle, Informix, and many other databases have ODBC drivers, it is possible to read a wide variety of data sources. Most other programs can produce one of the supported formats, or at least a fixed-field or delimited text file that will be read by ***Analysis***. ***Analysis*** can even read HTML tables, allowing data to be extracted directly from Internet Web pages.

The Microsoft Access/Epi Info 2000 format is standard for ***Analysis***. Other types of files are accessed through "Links" created in an Access database. If you choose to read a file format other than ACCESS/EPI2000, you will be asked to specify a new or existing Access database file (.MDB). When you choose the data file, a link is placed in the specified MDB file that can be accessed thereafter as an Access data table. Although it actually contains only information on the type and location of the non-Access file, it serves as a proxy for that file even if it is on a network or in another computer.

Analysis works from commands like READ, LIST, FREQ, and TABLES, but a dialog is offered for each command so that you can generate the command by making choices from the dialog box. You can also type the command directly or save it and run the resulting program later (as in Epi Info for DOS). The dialog boxes make it easy to get the right syntax for the commands.

After having read a file or data table, you will want to see its contents using the LIST command, and count the instances of each value of a variable using the FREQuency command. If you READ the view called viewOswego in SAMPLE.MDB, for example, a frequency of the ILL variable will reveal 46 records with ILL equal to "Yes" and a total of 75 records. A cross tabulation of ILL by another variable, using the TABLES command, will produce a 2x2 table with the count for each combination of the two variable values (if the variable is also a yes/no variable). Frequency tables are accompanied by confidence limits on the proportions, and 2x2 tables by odds ratios, risk ratios, and several types of confidence limits on these ratios, as well as chi square and Fisher exact tests. Stratified analyses result in Mantel-Haenszel summary odds ratios and confidence limits. For continuous variables like blood pressure or height, the MEANS command provides one-way ANOVA and Student's t-tests, plus the Kruskal-Wallis non-parametric test in case the former are inappropriate. Output from statistical analysis consists of Internet-compatible files in Hypertext Markup Language (HTML) that can be displayed in any browser. A small offline freeware browser is supplied with Epi Info, and no browser or Internet connection is required to view the results.

Linear regression and logistic regression can be performed. Logistic regression is done in a separate program that also analyzes files in the Epi Info for DOS format. It can be run from ***Analysis***, however, after choosing the variables to be included in the mode. A companion program for Kaplan-Meier survival analysis is provided for analysis of longitudinal studies, such as treatment trials with varying degrees of follow-up.

Analysis offers GRAPH and MAP commands with commonly used options. Both maps and graphs can be customized and saved in .MAP or Chart (.CHT) files that can be recalled later to reproduce the custom graph or map with a different data set.

Epidemiologic data in the field is rarely "clean," as it may be in a textbook or teaching laboratory. The cleaning process in ***Analysis*** is facilitated by commands to DEFINE new variables and ASSIGN values using a variety of mathematical and logical functions. IF statements provide conditional operations and RECODE allows grouping of data for age and other variables, or transformation of coding from one system to another.

Analysis commands can be saved in program (PGM) tables that can be run repeatedly or at a later time. Programs can be linked to an Epi Info menu to provide processing options for complete systems, such as public health surveillance systems.

A guided tour of ***Analysis*** is provided.

The Windows Word Processor

The Epi Info 2000 menu provides access to one of several word processors that should already be on your computer. It looks for Microsoft Word, then for Corel WordPerfect, Microsoft's WordPad, and finally the Windows Notepad. By editing the Epi2000.MNU file, you can change the word processor associated with the menu choice or button by that name. To do this, first make a backup copy of your Epi2000.MNU file under another name. Then, edit the block of commands in the MNU file called WordProcess. Between the words BEGIN and END, you will find a number of EXECUTE or OrEXECUTE statements. Insert an EXECUTE statement of your own with the correct path and name of your word processor. The statement should be the same as one that will run the word processor from the RUN window of the Windows START menu. Save the Epi2000.MNU file as a text file and test the menu to see that it runs your word processor.

Troubleshooting Hint: A common cause of problems in editing an MNU file is that some word processors add an extra extension when saving to a text file, resulting in Epi2000.MNU.TXT. The latter extension must be removed by renaming the file in the Windows Explorer or My Computer function before the Epi2000 menu will work with the MNU file.

NutStat: The Anthropometric Calculator

NutStat compares a child's age, sex, height, weight, and arm and head circumference with reference growth curves from either of two sources, and calculates percentiles, z-scores (number of standard deviations), or percent of median, as well as body mass index. Graphs of the growth standards with or without values from a particular child or group of children can be displayed or printed.

For height, weight, and height for weight, ***NutStat*** offers a choice of reference growth curves from the 1977/1985 CDC/WHO International standards[i] or the 2000 NCHS (National Center for Health Statistics) reference growth curves from the US[ii]. Arm circumference is compared with XXX[iii]. Head circumference and body mass index are compared with the 1999 NCHS US reference data.

To use ***NutStat***, choose it from the menu, use the FILE menu to OPEN a project database and a suitable table, and enter the age, sex, height, weight, and other optional characteristics of a child. Extensive configuration options for including or excluding items in both the English and Metric systems are available from the CUSTOMIZE option on the FILE menu.

NutStat will import data already available in Microsoft Access files, and perform calculations during the importation, storing the results in the ***NutStat*** file format. ***NutStat*** will also perform calculations on an Access file and add the results to fields specified by the user within the original file.

In an Epi Info View, the RELATE button in a ***MakeView*** field dialog can be used to create a button and a relationship with ***NutStat***, so that the nutritional items can become part of a larger questionnaire--a clinical history form, for example.

A guided tour of ***NutStat*** is provided.

Epi Map: A Mapping Program Compatible with Popular GIS Programs

Epi Map, the mapping component of Epi Info, is built around MapObjects software from ESRI, the makers of ArcView and ARC/INFO, popular Geographic Information System (GIS) tools. ***Epi Map*** displays SHAPE files from these two systems, and thus can use the enormous reservoir of map boundaries and geographic data available on the Internet in ESRI-compatible formats.

Epi Map is designed to show data from Epi Info 2000 files by relating data fields to SHAPE files containing the geographic boundaries. SHAPE files also can contain data on population or other variables, and can therefore provide numeric data that become part of the display either as numerator or denominator. Numeric data can be displayed either as color/pattern (choropleth) maps or as dot density maps with the dots randomly distributed within geographic regions. (To avoid calling geographic regions “polygons,” we will refer to each named polygon as a “region.”)

Point locations can be plotted automatically from data files containing x and y coordinates in various symbols, colors, and sizes. SHAPE files can contain lines or points to represent streets or point locations, and points can be placed on top of the SHAPE file layer to represent homes in which cases occurred or other geographic points of interest.

Most of the work in ***Epi Map*** is done from the Map Manager, which allows Layers to be constructed with SHAPE files and related data variables. Layers can be removed or moved to the front or rear of a series of layers. Many properties affecting colors, shapes, grouping of data, type of display, and other characteristics can be set in the Map Manager. When a map has been completed, it can be saved as a template or .MAP file that can be recalled at a later session or from ***Analysis*** to reproduce the original map.

A guided tour ***Epi Map*** is provided.

Visualizing Data: The VisData Program

VisData is a utility program, supplied by Microsoft with Visual Basic, for reading data files and examining and changing their properties. ***VisData*** can be used to examine the fields and properties in a variety of data file types, to display the data in spreadsheet (grid) format, and to edit or delete data items, rows, or columns. Utility functions are offered to COMPACT or REPAIR data files. COMPACT does not perform compression, but will reorganize files in which a great deal of editing has been done so that they occupy less disk space. REPAIR examines damaged files and repairs the damage if possible.

A guided tour ***VisData*** is provided.

Rapid Tour

For Experts In a Hurry

Epi Info 2000 Menu

Explore the main menu of Epi Info 2000. Using the Word Processor, examine the file EPI2000.MNU, which configures and programs this particular menu. Compare it with the file SURVEIL.MNU, which configures the Surveillance menu that pops up when you choose SURVEILLANCE SYSTEM under EXAMPLES.

MakeView

Run ***MakeView***. Choose NEW VIEW from the FILE menu, and give your first or last name to the database, and "NEW1" as the View name. Then create the first field by clicking the right mouse button where you would like to place a question and responding to the dialog. Create several fields and then use the ADD PAGE button to move to page 2. Be sure to include a grid field and a multiline field. Use the RELATED VIEW button to create a button to access a related View, and then create the view by clicking on the button and creating one or more fields. A related form is created, and is represented and accessed by a button. When you have completed the view(s), choose ENTER DATA from the File menu to run the ***Enter*** program and enter data in your new view. If you wish to add Check commands, return to ***MakeView*** by choosing EDIT VIEW from the File menu and then clicking on the Program button. Choose a field and then add commands with the aid of the dialogs that appear when you choose a command.

Enter

Run ***Enter***. From the FILE menu, choose OPEN, the database SAMPLE.MDB, and the View SURVEILLANCE. Enter several records until you are familiar with the program features. Choose Hepatitis B for the DISEASE field in one record and Lyme Disease in another.

Analysis

Run ***Analysis***, click on the READ button, and choose the SAMPLE.MDB database and the OSWEGO view. Try LIST, FREQ, TABLES (of VANILLA by ILL) and GRAPH (of AGE). Experiment with the SELECT and WRITE commands to write a new table (or perhaps a dBASE file) containing only cases (ILL= "+"). READ the new file and verify that you have 46 records only for ILL= "+".

NutStat: The Nutritional Anthropometry Program

Run ***NutStat***. Open the NUTRI.MDB database and then the nutChildren database. Choose CUSTOMIZE from the menu and adjust the settings for your locality and measurement preferences. Enter several records, using the NEW buttons at the top to move to a NEW child's ID or a NEW date of measurement for a particular child. Return to Record 101 and click on GRAPH. Set the graph to ZSCORE on the CUSTOMIZE screen and choose GRAPH again.

Epi Map

Run ***Epi Map*** from the main menu. Choose MAP MANAGER from the FILE menu. Use ADD MAP LAYER to display MXState.SHP, selecting NAME as the Geographic Field. Use ADD DATA to select MXMapOct from the SAMPLE.MDB database, and choose PerAdolBIRTHoct98 as the Data Field. You should see a choropleth (color/pattern) map of "Percentage of Births to Adolescents for October 1998."

To display points from a database, using John Snow's example of the 1854 cholera outbreak in the Soho District of London, choose CLEAR ALL LAYERS and then use ADDMAPLAYER to load the SohoSt.SHP shape file. Choose ADD POINTS and then, in Sample.MDB, select the SohoPumps table. Choose X_COORD as the X Field and Y_COORD as the Y Field, click on NAME to produce labels, click on Color to choose a color, and select 15 as the size. You should see the water pumps locations displayed on the map. Now display houses with cases by going through the same process with ADD POINTS and the SohoDead table in Sample.mdb. Size 4 is a good choice for the symbols. Now you should see the houses with cases and their relationship to the pump locations. Bearing in mind that denominators are an important issue, you can appreciate a strong geographic association with the Broad Street pump.

The VisData Utility for Viewing Data

Run ***VisData*** and choose Open Database and then Microsoft Access. Open the LANGUAGE.MDB database and then double-click on the NUTSTAT table. You should see a grid filled with phrases from the ***NutStat*** program in six languages. This is a part of the database that allows for easy translation of programs.

Guided Tour

Getting Acquainted With the Programs

Taking the Guided Tour

This chapter provides step-by-step instructions for exercising the main features of each program. If you are reading from the screen, you may find it convenient to print the instructions using the PRINT command on the FILE menu of your browser or word processor. On a large screen, it is also possible to keep the instructions open in one window and work with Epi Info elsewhere on the same screen.

Guided Tour of the Epi Info 2000 Menu

Highlights:

- Configurable features of the main menu
- Example of an entirely different menu constructed by copying, renaming, and editing the Epi2000.MNU file.
- Brief look at an MNU file

The screen image for the main menu of Epi Info 2000 is John Snow's famous map of the location of cholera cases surrounding the Broad Street pump in London in 1846. Artistic license has been taken in representing cases near the pump as vertical polygons rather than with Dr. Snow's neat tick marks.

The main programs of Epi Info can be accessed either through the PROGRAMS menu or by clicking on the buttons. The buttons can be turned on or off with the BUTTONS item on the SETTINGS menu. The Guided Tour is part of the Epi Info 2000 manual and help-file system represented on the MANUAL menu. Several exercises for learning epidemiology and computing are found on the TUTORIALS menu.

On the EXAMPLES menu, choose SURVEILLANCE MENU. A second menu appears with different menus, background picture, and screen text. This is the beginning of a Surveillance system to be developed in Epi Info 2000, but most of the entries on this menu are not yet active. The menu is presented to show how easy it is to customize the menu and use it for your own programs. Close the Surveillance menu by choosing EXIT from the ENTER DATA menu.

To gain confidence and to see that your system is functioning well, you might click on each of the buttons on the main menu and briefly examine the program that appears. Exit from each one by clicking an Exit button, the small “x” box in the upper right corner, or EXIT in the FILE menu. We will visit each program again in more detail, as described below.

If you are interested in the details of the files behind the menu, use the word processor and OPEN the files EPI2000.MNU and SURVEIL.MNU. Note that differences in these text files are responsible for the differences in appearance and function of the two menus.

Guided Tour of the MakeView Program

Highlights

- Designing a new form or questionnaire (a View)
- Text and numeric fields
- Specifying a list of Legal values
- Inserting a grid, the automatic way to deal with repeating data within a questionnaire
- Large text (multiline) fields

To run ***MakeView***, click on the MakeView button on the main menu screen. You should see a blank page for constructing a “View.” Questionnaires are called Views in Epi Info 2000 because there can be more than one View of a database or data table. A database table with the prefix “view” stores the screen appearance of the questionnaire, the characteristics of the fields, and any Check code that gives special instructions for the data entry process. Data values entered in the ***Enter*** program are stored in another table, without a special prefix.

To make a view, from the FILE menu choose Make New View. The dialog CREATE OR OPEN PROJECT appears. Enter a name for your project database, such as your name or initials, and click OPEN.

A project or database (.MDB for "Microsoft Database") file can hold as many Views and data tables as you wish (well, up to 1000, anyway). Generally it is best to create a new MDB file for each project you develop, as an MDB containing hundreds of tables will be hard to copy to diskettes. ***Analysis*** programs for processing the data can be stored in the same MDB with the data, making a convenient project package.

In the "Name the View" dialog, enter MOTHER as the name of the View within the MDB, and then click OK. Place the cursor near the upper left corner of the blank page and click the *right* mouse button. The field dialog box that appears offers options for entering the prompt, the field type and length, and a number of the characteristics that were previously implemented in Check files.

For the first field, enter the prompt “First Name” (without the quotation marks) and press Enter twice. This makes a text field that can hold up to 255 characters.

For the next field, you could move the cursor and right-click with the mouse on a suitable location, but, to see a shortcut method, press Enter instead while the cursor is in the FIRST NAME field. The field dialog pops up again and you are ready to enter “Last Name” as the prompt. After doing so, press Enter twice, and note that the second field is now automatically positioned on the View.

Below First Name, right-click to add another field. Enter the prompt “Today’s Date,” and use the scroll bar to the right of the field types to see the rest of the list of types. Choose the DATE type and the appropriate date format as MM-DD-YYYY or DD-MM-YYYY in the dialog. Click OK. Add another field for “Date of Birth,” using the same field type and pattern. Click OK.

Right-click on the form to make a field for AGE. Type “Age” as the prompt. Choose NUMBER for the TYPE and then choose ### or ## from the PATTERN list. You can also type patterns into the pattern window. Click on OK at the bottom of the dialog.

The next field is “Sex.” We will use it to illustrate how variable names are constructed. Right-click where you would like to place the field. Type “Male, Female, or Unknown Sex” in the prompt window, press Enter, and note what appears in the Field Name window on the right. Now click again in the prompt window, and with the left mouse button held down, select just the word “Sex”. Double-click on the selected word, “Sex,” in the prompt window. Note that the variable name becomes SEX. (In Epi Info 6, we would have enclosed “Sex” in curly brackets.)

Now create legal values for SEX by clicking the LEGAL VALUES button. In the dialog box that appears, choose CREATE NEW, and then enter suitable values (Male, Female, Unknown) in the list that appears, pressing Enter after each to obtain a new blank line. Click OK, and then OK again in the field dialog box. Note the button on the right side of the SEX field. Left-click on the button to show the list of legal values from which to select during data entry.

To move a field on the screen, click on the prompt for the field and drag it to a new location while holding the left mouse button down. Use this method to space the fields on the page. Most types of fields can be resized by clicking in the field and then clicking and dragging the colored “handles” that appear. Text fields are limited to one line, but we will add a multiline field later.

It is time to save the page and add another one. Click on the ADD PAGE button under the page window on the left side of the screen. The first page is saved automatically and a blank page appears.

We are going to insert a Grid (table of columns and rows) on Page 2 to record the names, ages, and immunization status of the children in the household. Right-click in the upper left corner of the form and enter “Children in the Household” as the prompt. Click on the INSERT GRID button in the dialog, and then click the OK button.

Enter the name of the first grid column: “Name”. Click on SAVE COLUMN about midway down the form (DONE at the bottom is now for the entire grid.). Enter the second column as “Age” and make it a NUMBER. Click SAVE COLUMN and enter the last column as a text field called “Immunization.” Now click SAVE COLUMN once more and then DONE at the bottom of the dialog. The grid will appear on the View.

Click on the grid so that handles appear around it. Click and drag the lower right handle or others to adjust the size of the space for the grid. Click outside the grid to remove the handles. To adjust the size of the columns, hover the mouse over the line between two column headings until a right/left arrow appears. Then hold the left mouse button down and drag the line to the right or left to adjust the column width.

The grid will create a related file that will allow the user to enter as many children as needed for each household. The program automatically maintains an appropriate key for linking the related file.

Click ADD PAGE again and you are ready to make page 3 of the View. Now make a field of the MULTILINE type having the prompt, “Comments of Interviewer.” Click OK and then click on the field, adjusting its size as you did with the grid to make it big enough to enter a number of comments. There is no practical limit on the amount of text that can be entered in MULTILINE fields. Add a text field for “Interviewer’s Initials,” and save this page with the SAVE command on the FILE menu.

Checking and Controlling Data Entry

Highlights

- Inserting commands to customize the data entry process
- Calculating an age from two dates

We would like AGE to be calculated automatically after entering Today's Date and a Date of Birth. If both of these dates are given then the cursor should skip over AGE after the calculation is performed. These functions can be programmed in the Check code environment.

Return to page 1 by clicking on its entry in the page list on the upper left. Bring up the Check environment by clicking on the PROGRAM button to the left of the view. A list of fields and available Check commands appear at the top of the screen in a tabbed dialog, and the program editor is displayed below.

Click on the arrow under "Choose Field Where Action Will Occur" to pull down the choices. Since we want Age to be calculated after DateOfBirth is entered, choose the DateOfBirth field. Click on the VARIABLES tab and then the ASSIGN command. Display the list of available variables for the ASSIGN VARIABLE by clicking on the arrow. Choose AGE, and then type Years (*TodaysDate-DateOfBirth)* in the next space. Choose the field names from the pull-down list if you do not know them in advance. Click on the SAVE button and note that the command,
"ASSIGN Age=Years(DateOfBirth, TodaysDate)" appears in the program editor. YEARSOF is a function that calculates the interval between two date fields in years rather than days, weeks, or months. The starting date is listed first and the ending date second in the parentheses, separated by a comma. A list of functions is contained in the Functions and Operators chapter of the manual.

To make the cursor skip over AGE if it has been calculated, choose AGE as the FIELD WHERE ACTION WILL OCCUR. A prompt asks if you wish to save the previous commands. Click the YES button. Now from the RECORDS tab, choose the IF command. In the CONDITION blank, type (or select from the pull-down list and the buttons) the condition, AGE>1. Click on the THEN button. Now from the tabbed selection of commands, click GOTO, and within the next dialog, type or select the variable SEX. Click OK, and you should see in the program window, the command,

```
IF AGE >1 THEN
    Goto  Sex
END
```

You have written your first Epi Info 2000 Check program. Exit from the Check programming facility by clicking on the OK button and answering "Yes" to the prompt about saving Check code. This completes the view. Although you could exit from ***MakeView*** at this point and run the ***Enter*** program from the Epi Info 2000 menu, it is more convenient to run ***Enter*** from within ***MakeView***.

Making a Database

Remain in ***MakeView***, choose ENTER DATA from the FILE menu and respond “OK” to have the program construct a database from the view. The data table will have the name displayed unless you choose to edit the name. The ***Enter*** program displays the view for data entry.

Guided Tour of the Enter Program

Highlights

- Entering data and verifying that your age calculation works
- Moving from page to page
- Opening an existing View and database
- Navigating from record to record
- Searching for particular records
- A sample surveillance view with related views available according to disease condition

You should have the MOTHER questionnaire on the screen. If not, go back to the main menu and choose ENTER DATA, OPEN on the File menu, and then the database that you created and the MOTHER view. Enter data in the fields displayed. After you enter Date of Birth, the age should be calculated automatically, and the cursor should jump to the Sex field. At the end of each page, the entries will be saved automatically. On page 2, fill in the grid with reasonable answers. After the first line, a second is created automatically. Enter as many children as desired, moving the cursor with the ENTER key and/or arrow keys. Press the Enter or Esc key to move to the next page, saving page 2 on the way. You can also click on the NEW button to save the current record and move to a blank record.

After entering the data for page 3, press the Enter key. You should now see an empty record 2, ready for entry. Note that the record number appears on the lower left.

Opening Another View

To see another view containing more data, choose OPEN from the FILE menu and then click on CHANGE PROJECT. Choose the database SAMPLE and View OSWEGO. Note that the number of records + 1 appears at the lower left.

Moving From Record to Record

Examine the records in the file by moving from record to record with the arrow buttons on the lower left. The double arrows move to the first and last records; single arrows move one record at a time. To move to a new record, click on the double right arrows twice.

Finding Records

To find records matching specified criteria, click on the FIND button on the left. A dialog box appears. Choose the AGE field and then type "11" (without quotation marks) in the field that appears. Click on the OK button to find all the records in which AGE is 11. To choose one of these records for editing, double-click on the left side of its row until the entire row is highlighted and the selected record appears on the screen. If you prefer not to select one of the records shown, but to continue with the current record, click on the BACK button.

A More Complicated View with Groups of Variables and Related Views

Open the view called SURVEILLANCE in the SAMPLE.MDB database. Note that the variables are arranged on panels. Each panel is a GROUP. In ***Analysis***, group names can be referred to as a shortcut to perform operations on all the variables in the group. For example, LIST PersonalInfo would display data from all the variables in the lower group panel.

Related Views are displayed in the box in the lower part of the left panel. Because each of the Views shown is designed for a particular disease, they are inactive and do not respond to mouse clicks when you first open SURVEILLANCE and a new, blank record is displayed.

Use the single and/or double left arrow buttons in the lower left panel to move back to record 1. The double arrow buttons move immediately to the first or last records in the table. The single arrows move one record at a time. The New button moves to the next empty record.

Note that the Disease in record 2 is "Hepatitis," and that the HEPATITIS DETAILS button is therefore active. Click on this button to see the special form for Hepatitis. Use the Back button to return to the main Surveillance form.

After experimenting with SURVEILLANCE and perhaps entering one or more records, choose EXIT from the FILE menu to return to the main menu.

Guided Tour of the Analysis Program

Highlights

- READ a view or a data file or table
- LIST the contents of the database
- Obtain the FREQuency of values for a field
- Cross-tabulate with the TABLES command and resulting epidemiologic statistics

- The library of previous output, all in HTML for the Internet
- Choose how "Yes" and "No" are displayed
- Define a new variable and assigning a value
- Use an IF statement to determine and assign case status
- SELECT a subset of records to process
- RECODE values to group the AGE field
- WRITE data to another file or table
- READ a non Access file
- READ related tables in a view automatically

To run ***Analysis***, click on the ANALYZE DATA button on the main menu screen. Note that all commands are shown in the tree view on the left. Clicking on a command will bring up a dialog that places the command in appropriate form in the program editor at the bottom of the screen. Results appear in the third window above the program editor, which is a simplified version of the Microsoft Internet browser.

READing a View in *Analysis*

Click on the READ command. A dialog box appears so that you can choose a database and a view. Choose the project SAMPLE.MDB and the View OSWEGO. Click OK and note that the READ command appears in the program editor in the proper syntax. You are creating a program by responding to questions in the dialog boxes.

Lists

Click on the LIST command. In the dialog that appears, choose one or more variables or click on “ALL” to choose all. Choose GRID as the output format and click OK. The variables are displayed in columns in a scrolling window. Click on the EXIT button on the right side of the screen to leave the Grid. Try the LIST command again, but choose HTML as the output format. This time the results appear in the form of an Internet web page displayed in the small browser included with Epi Info 2000. This browser displays web pages on the local machine, but is not itself connect with the Internet. If you have another browser on your computer, the Epi Info browser will use it for Internet access if necessary.

Frequencies

Choose the FREQuencies command. In the dialog box, use the dropdown menu to select one or more variables, and then click OK. After a short wait, the results should appear in the browser window. Scroll up and down and note that each table is accompanied by yellow bars to the right that indicate the frequencies. Statistics will be displayed after the table if the value of the variable is numeric, as in AGE, but not for Yes/No fields like ILL.

Tables

Click the TABLES command. In the FREQUENCY OF field, choose VANILLA and in FOR EACH VALUE OF, choose ILL. This will perform a cross-tabulation of VANILLA by ILL. Note that stratified analyses can be done by inserting the name of the stratifying variable(s). Summary data can be processed by setting WEIGHT equal to the name of a COUNT field. Click OK.

Note that the output in the browser includes a table and a graphic representation of the table values in each cell. Statistics are displayed below the table. If you have a printer connected, try printing the table by clicking on the PRINT button. (Information on the statistics in Epi Info is available in the Statistics file under HELP on the main menu.)

Viewing Previous Results

Click on the entry called RESULTS LIBRARY at the top or bottom of the TABLES output in the browser. An index page appears, showing previous commands that have produced output files. Click on any of the entries to display it. An archiving system is provided so that important results can be selected and saved for future reference. You can learn more about storage of results by choosing the OUTPUT tab and examining the choices under STORING OUTPUT.

Setting the Displayed Values for Yes/No Fields

Click on SET under OPTIONS in the command tree on the left side of the screen. Note the options for customizing output. Change the values to be displayed for Yes/No fields, choosing from those available or typing your own, such as “Si” and “No” in Spanish. Choose LIST again from the STATISTICS tab and verify that the values displayed are those of your choice.

Defining a New Variable

Under Variables, choose the DEFINE command. Type STATUS as the name of a new variable. We want to set this variable to “Case” if the person was ILL and “Control” otherwise. A Standard variable, with the value reset for each record as the program passes through a table, is the best choice for this purpose. Click OK, and the necessary statement will appear in the program window.

An IF Statement

We can use an IF statement to set the value of the new STATUS variable. Choose the SELECT/IF tab and click on IF. The first item in the dialog is the condition under which the following statements should or should not be executed. The final format of the necessary IF statement is:

```
IF  ILL = "Yes"  THEN
        STATUS = "CASE"
ELSE
```

```
        STATUS = "CONTROL"
END
```

You will not have to remember the format, however, since filling in the blanks in the dialog will allow Epi Info 2000 to write the necessary command in the program editor. In the first blank, fill in the Condition as ILL="Yes". To do so, you can choose the variable ILL from the list of variables in the second window and choose the equal sign and condition “Yes” by clicking on buttons with these labels. The “Yes” button will display whatever label is specified for “Yes” in the SET dialog—“(+)”, “Yes”, “Si”, etc.

Regardless of the setting for Yes/No fields, “Yes” is always represented in the database as 1, No as 0, and Unknown as a blank or null value. These values can also be used, as in:

```
IF ILL = 1 THEN       (Note that numbers do not have quotation marks)
  Etc.
```

Now click on the THEN button and type STATUS="CASE". Click on ELSE and enter STATUS="Control". Click OK to finish the command.

Use the LIST command to verify that STATUS is indeed properly set, and that records where ILL = “Yes” have “CASE” as the value of STATUS.

The SELECT Command

The SELECT statement limits subsequent analysis to particular records based on criteria that you specify. To work with cases only, for example, choose the SELECT statement and enter the condition STATUS = "CASE". Use LIST or FREQ STATUS to show that cases only are included in the analysis. To return to working with all the records, choose CANCEL SELECT and note that it places the word SELECT, without conditions, in the program.

The RECODE Command

To Group the values of AGE, first DEFINE a new variable called AGEGROUP, using the DEFINE command. Then Choose RECODE and specify that you will recode from AGE to AGEGROUP, using the drop down list. Then enter 0, 18, and “Child” on the first line of the grid and 19, 120, and “Adult” on the second line. Click OK, and then do a LIST to see the results.

Creating a New File with the WRITE Command

At this point, we have made several improvements in the dataset, and might want to create a file containing the new variables. The new file can be either an Epi Info 2000 (Microsoft Access) file or one of many other file types. Choose the WRITE command, and then, from the list of Output Formats, choose “dBASE IV”. Specify “All” variables and “Replace” so that an existing file by the same name will be overwritten. Give your Initials for the file name. Click on OK to write the file.

READing a dBASE File

Now that you have produced a dBASE IV file, it is time to test the flexibility of ***Analysis*** in READing a variety of file types. Choose READ from the DATA tab, specify dBASE IV format, and read the file you have just produced. Use LIST and/or FREQ to verify that the variables you created and their values are contained in the new file.

READing a View with Related Views

Analysis will manage related views so that you can produce analytic results easily. *This feature is currently under construction.*

Guided Tour of the NutStat Program for Nutritional Anthropometry

Highlights

- Entering data from one child's measurements
- Interpreting results from calculations based on accepted reference standards
- Graphing more than one result to show a child's growth
- Customizing the data entry screen

Run the NutStat program by clicking the NUTRITION button on the main menu. The program should obtain the name of the last database accessed from the EPIINFO.INI file and load this database automatically. If you do not see record number 105 and the data table name NUTCHILDREN at the top of the program, use the OPEN command on the FILE menu to open first the NUTRI.MDB database and then the table called NUTCHILDREN.

Click the left arrow next to the IdNumber several times until you see a record belonging to Alouetta Delia. Alouetta has four records in the table. To see the others, click on the left-pointing arrow in the DATE OF MEASUREMENT box and look at each of the earlier records. Click on the GRAPH button. Choose “Z-Scores” as the graph type and click OK to see a graph of her results compared with the International growth reference curves. Age is shown across the bottom of the graph and the left axis shows by how many standard deviations the child's measurement differs from the International average reference standard for that age and sex. The advantage of z-scores (standard deviations) is that both height and weight can be plotted on the same graph. In interpreting the graph, it is important to know that two standard deviations in either direction from 0 is approximately the 95th percentile and three is near the 99th percentile. Hence, in Alouetta's second set of measurements, her height is nearly at the 99th percentile for girls her age, but her weight is slightly below the norm; she must have been tall and slender at age 4 (48 months) when this set of measurements was taken.

To move to another child's record, choose the next IDNumber by clicking on the right arrow in the IDNumber box. By clicking repeatedly, you will discover how many children there are in this database. Click on the second button to the right of the IDNumber to create a new record and IDNumber Note that clicking the left arrow from the new record will return to the last completed record in the table.

In the new record, enter a name, sex, and birth date. Note that the age is calculated automatically after you press Enter. Now enter values for height and weight (for a 10-year old, 140 centimeters and 40 kilos will do). Note that the statistical calculations appear as soon as you press Enter in the weight field. The body mass index is calculated automatically (but we are awaiting new standards before calculating the age-specific values for body mass index).

Enter 15 centimeters or 6 inches for arm circumference and note the z-scores. Now click on New in the IDNumber box to save the record and go on to the next.

From the File menu, choose Customize to see the options available. On the Units tab, choose either the English or Metric system of measurement, click on Arm Circumference to deselect it, and then choose OK. Note that the main screen now displays only the measurement units selected and that Arm Circumference no longer appears. Return to Customize and set up the main screen in the configuration you find most useful for your own environment.

NutStat is designed to accept the new NCHS growth reference curves as soon as they become available, but these features have been disabled pending release of the new standards.

Guided Tour of the Epi Map Program

Highlights

- Adding a shape file to create a map
- Adding data to be represented by color density
- Creating a map of cholera cases in John Snow's London using X and Y coordinates of case households

First, let's make a map of Mexico representing one numeric result for each State by coloring the polygons that represent the States. Run ***Epi Map*** from the main menu. In the FILE menu of ***Epi Map*** choose MAP MANAGER, and from the LAYERS card, click on ADD MAP LAYER. You should see a choice of SHAPE (.SHP) files, containing the boundaries of geographic regions. Open the one called MexState.shp and choose NAME as the field containing the names of Mexico's states. Next is an opportunity to choose a database containing public health data from Mexico. Open the SAMPLE.MDB database and choose the table called MexMapOct. The appropriate Geographic Field is STATE and the data field is PerADOLBirthOct98, the percentage of births in which the mother was an adolescent in October 1998. You should see a map of Mexico with the adolescent pregnancy percentage for each state represented as a shade of gray.

You can experiment with features such as FIND on the EDIT menu (also represented by the binocular button). Try finding "Vera," for example. The magnifying glasses with + and - signs are for zooming to larger or smaller sizes. Click on the one with the + sign and then click and drag a rectangle over an interesting part of the map. It will zoom to fill the frame. Try panning—moving the map with the white mitten. You can restore things to normal either with the negative magnifying glass or with the world button that represents the global or home view. The "I" symbol means "information." Click it and then click on a state to see the data values for that state—both those contained in the SHAPE file and those in the linked database.

Epi Map can represent data as discrete points on the screen, as well as quantitative values within polygons. The following set of instructions will produce a map resembling John Snow's famous map of cholera cases in the Soho district of London in relation to the various public pumps used for drinking water. A popular theory of the time related cases to the presence of an old cemetery where plague victims from a previous century had been buried; hence the cemetery will be included as a separate layer in the map.

First run ***Epi Map*** from the main menu. From the FILE menu, choose MAP MANAGER. Choose ADD MAP LAYER, and then open the SOHOST.SHP file. SHP stands for SHAPE, the most common type of file used in mapping with Epi Map. Now you should see a map of the streets of Soho. Return to the map manager and add the cemetery by using ADD MAP LAYER and choosing SOHOBURI.SHP. To add the locations of the water pumps, use ADD POINTS and select SAMPLE.MDB and then SOHOPUMP. Another dialog asks for the x and y coordinate fields. This is an easy choice in this exercise; X Coord is the X coordinate and Y Coord is the Y coordinate. Click on NAME in the choice in the lower left so that the names of the pumps will be displayed. To make the pumps larger, choose 15 for their size, and change the color by clicking in the black rectangle and choosing a brighter color. Click OK after choosing the color, and then again for the dialog. You should see the pumps and their names arrayed on the map.

To display the cases, repeat the ADD POINTS process, this time choosing SOHODEAD from SAMPLE.MDB and 5 for the size of the symbols. Use a contrasting color and do not choose a field for displaying text. You should see a large number of points representing the addresses where a case occurred. Since some households had multiple cases, the visible points actually represent households having at least one case rather than cases individually.

Although the map suggests that the Broad Street pump was central to the location of cases, and other evidence did incriminate this pump, one must remember that the distribution of dots is a function not only of the mortality rate but also of population distribution, and that rates rather than numbers of cases would be necessary to draw a scientific conclusion. Assuming, however, that the population was fairly evenly distributed through Soho, the impression that the map gives is useful. Dr. Snow's similar map helped him convince the neighborhood council to remove the handle of the Broad Street pump, and the epidemic subsided.

Guided Tour of the VisData Program

Highlights

- Seeing the inside of the language translation database
- Seeing the inside of an Epi Info View table

VisData is a utility program for viewing and performing useful operations on database files or tables. Choose Visualize Data from the main menu. Choose OPEN from the FILE menu and the MICROSOFT ACCESS file type. Open the SAMPLE.MDB database. The buttons on the toolbar control what happens when you click the icon for one of the tables. Tool tips explaining the function of each button appear if you allow the mouse to hover over a button. Click on the leftmost toolbar button and then the one with the grid just to the left of the blue cylinder. Double-click the table name AgeWithCount to see a spreadsheet view of the data in this table. Note that the variables are represented as column headings and records as rows. ***VisData*** is a useful utility for seeing the internal details of databases. Microsoft Access, if you have it, can be used for the same purpose.

Next Steps

This completes the Guided Tour. You are now familiar with the main features of Epi Info 2000. We suggest that you try working with your own data, either by designing a View and then entering data or by READing an existing dataset in the ***Analysis*** program. The rest of the manual provides more detail on the programs, and the How To chapter has instructions for solving specific problems. Reference material is in the Commands chapter and is also available from the programs via the Help buttons in particular dialogs.

We hope that you have enjoyed the Tour, and will send your impressions and suggestions to the Development Team at the contact addresses in the front of the manual. If you would like to participate in the Epi Info Worldwide Discussion Group (a LISTSERV) by e-mail, please use the instructions in the front of the manual to join. The Epi Info Web Site is available at all times to provide information, updates of programs, and access to vendors, trainers, and the Epi Info Technical Support staff.

How To…

Specific Tasks and Frequently-Asked Questions

Examples of Public Health Tasks--Epidemiologic Investigation

Some of the possible steps in conducting an epidemiologic investigation are listed, in the order in which they might occur. Page numbers indicate where to find relevant information.

.

Install Epi Info
Learn to use Epi Info
Study sample epidemic data files
Design a questionnaire
Manage hierarchical data using grids or related views
Make a data file
Set up checks on the data entry process
Set up calculated fields for data entry
Handle anthropometric data
Enter data
Check the accuracy of data entry by:
Viewing data in ***Enter***
Printing data using ***Enter***
Viewing data in ***Analysis***
Listing and printing data using ***Analysis***
Make backup files
Find records using selected criteria
Edit data in ***Enter***
Edit data in ***Analysis*** using UPDATE
Revise the questionnaire, adding or deleting variables
Read data for ***Analysis***
Produce lists
Select records
Recode/group data
Dates, use and format
Define new variables
Frequencies
Tables (cross tabulations)
Stratified analysis
Means
Matched analysis
Make graphs
Make maps
Print results
Produce files for other statistical systems
Read files from other systems

Programming
Debugging, coping with errors

Examples of Public Health Tasks--Disease or Injury Surveillance

Some of the 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.

Study examples of surveillance system programs
Make questionnaire Views
Set up relational (hierarchical) file systems
Program automatic checking on data entry
Set up calculated fields for data entry
Set up help messages or files for data entry
Restructure data and write new data files
Develop a menu system
Check the accuracy of data entry by:
Viewing data in ***Enter***
Listing data using ***Enter***
Viewing data in ***Analysis*** using BROWSE
Listing data using ***Analysis***
Make backup files
Find records using search criteria
Edit data in ***Enter***
Edit data in ***Analysis*** using UPDATE
Revise the questionnaire, adding or deleting variables
Write or modify programs for ***Analysis***
SELECT
IF
RECODE
Screen and printer communication
Relational file analysis
RUNning other programs
Loops and subroutines
Debugging, coping with errors
Make graphs
Make maps
Customize ***Analysis***
Develop custom report formats
Export (convert) files to other formats
Import data
Use character sets other than English

EPI 2000 Menu

Develop a new menu (.MNU) file (*Epi 2000 Menu*)

To develop a new menu file, examine EPI2000.MNU in the Word Processor. Make your own .MNU file, using SAVE AS on the FILE menu of the Word Processor. Save a copy of EPI2000.MNU by another name, and edit the menu items and commands to your own taste. (See MENU in *Commands* chapter.)

Change the picture on the menu (*Epi 2000 Menu*)

Select SETTINGS on the menu, then click on PICTURE. From the file dialog that appears, select a suitable image file of the .BMP type. Other types of images can be converted to the .BMP format by using commercial or shareware image editing programs.

Images can also be specified on the command line when running the menu program from the DOS command line, from the RUN command on the Windows START menu, or from another program.

```
C:\Epi2000\EPI2000 EPI2000.MNU REFCAMP.BMP
```

This will display the Epi2000 menu, but with the picture normally shown on the Surveillance menu example instead of the EPI2000.BMP image that is specified in the EPIINFO.INI file.

A third method of specifying an image is with the PICTURE command.

```
PICTURE REFCAMP.BMP
```

Inserting this command in a block of commands in a .MNU file allows a temporary change of the menu image. (See PICTURE in *Commands* chapter.)

Add or change a menu item (*Epi 2000 Menu*)

To add or change a menu item, edit the .MNU file. For example, to add an item to run the Windows Notepad program, first add the MENUITEM command into the .MNU file. Enter the words that will appear on the menu, followed by the command block name. An ampersand (&) can be used to choose a single character "hot key." Hot keys must be different for each item on a simple menu.

```
MENUITEM "&Notepad", Notepad
```

Next, begin to add your CommandBlock section:

```
Do Notepad
Begin
Run Notepad.EXE
End
```

A menu entry is created on the POPUP menu within which the MENUITEM occurs. When the item is selected, the commands in the CommandBlock named are executed. (See MENUITEM in *Commands* chapter.)

Add or change a button (*Epi 2000 Menu*)

To add or change a button, edit the .MNU file. The BUTTON command is used to create a button. The command

```
BUTTON "&NOTEPAD", Do Notepad, 50%,80%
```

creates a button on the screen with the caption "Notepad" and the locations given as percentages of screen width and height: half way across the screen (50%) and (80%) of the down from the top. Clicking on the button with the left mouse button activates the commands in a block called Do Notepad. (See BUTTON in *Commands* chapter.)

Program the response to a menu item or button (*Epi 2000 Menu*)

A named block of commands is needed in the .MNU file for the program to respond to a menu item or button.

For example, the MENUITEM "Word Processing," when selected by the user, runs the commands in the block called Eped, which looks like this:

```
Eped
Begin
Run EPED.exe
End
```

You can create these lines or edit the ones that are already there. A number of menu or DOS batch file commands can be inserted between the Begin and End markers. (See MENUITEM in the *Commands* chapter.)

Set the working directory (*Epi 2000 Menu*)

Select SETTINGS on the menu and click SET WORKING DIRECTORY.

Turn buttons on or off (*Epi 2000 Menu*)

Select SETTINGS on the menu and click BUTTONS ON OR OFF.

Let users set the working directory from a menu (*Epi 2000 Menu*

Insert the SETWORKDIR command into a block of commands in a .MNU file. For example

```
WorkingDir
Begin
SETWORKDIR "Please choose a working directory for which you have write privileges."
End
```

The menu changes the logged directory to the specified directory before executing commands. (See SETWORKDIR in *Commands* chapter.)

Let users turn buttons on or off within a menu (*Epi 2000 Menu*)

Insert the SETBUTTONS command into a block of commands in a .MNU file, as in:

```
ButtonSetting
Begin
SETBUTTONS
End
```

(See SETBUTTONS in *Commands* chapter.)

Send information to other programs using permanent variables (*Epi 2000 Menu*)

Use the ASSIGN command to create a variable and assign a value, as in:

```
ASSIGN LANGUAGE="SPANISH"
```

The value of LANGUAGE becomes available as a "permanent" variable within ***Analysis*** and in check code written in ***MakeView***. In menu (.MNU) programs it is not necessary to define variables; they are defined automatically by being included in an ASSIGN statement. (See ASSIGN in *Commands* chapter.)

Bring up the help system (*Epi 2000 Menu*)

Epi Info 2000 provides help on a variety of topics. To bring up the HELP files you can go to MANUAL on the menu and click on the topic with which you need help. The Main Menu also provides a button to allow easy access to the Guided Tour, which can help you get acquainted with the programs.

MAKEVIEW

Make a view (*MakeView*)

Open ***MakeView*** by clicking on the MakeView button on the Main Menu screen or by going to the PROGRAMS menu and selecting MakeView (Questionnaire). Right click under Page Names and select MAKE NEW VIEW. You can also select MAKE NEW VIEW from the FILE menu.

Insert a grid (table) in a view (*MakeView*)

Right click on the page to open the **Field** dialog. Enter your prompt, choose GRID as the TYPE, and click OK. Enter the name of the columns one at a time; click on SAVE COLUMN after each entry. When finished, click DONE at the bottom of the dialog.

Line up fields (*MakeView*)

Select fields to be aligned by holding down the left mouse button and dragging to draw a dotted line around the fields. Select FIELDS from the menu, click ALIGNMENT, and select VERTICAL or HORIZONTAL.

Insert a background picture or color on a page (*MakeView*)

Select VIEW from the menu and click BACKGROUND. When the **Background for Page** dialog comes up, choose whether you want a background color or an image. Click in box of desired action. Once you click in a check box, either a **Color Scheme** dialog or **Background Image** dialog will come up. Choose desired color or image and click OK.

Group fields together as a functional unit (*MakeView*)

Select fields and the area for a group by holding down the left mouse button and dragging to draw a dotted line around the fields. Select FIELDS from the menu, then select NEWGROUP. When the **Group** dialog appears, type in a group description or name, choose the desired color, and click OK.

Cut, Copy, and Paste fields (*MakeView*)

Select fields to be cut or copied, go to EDIT, and choose which action will be taken. To paste, go to EDIT and click on PASTE.

Delete an existing data table without deleting the view (*MakeView*)

Choose DELETE EXISTING DATA from the VIEW menu and respond to the warnings and questions that appear.

Change the Prompt, Name, or Type of a Field in a view (*MakeView*)

Right click on the field that needs to be changed. When the **Field** dialog appears, make the changes needed and click OK.

Create a Related Form/View (*MakeView*)

Right click on the screen to open the **Field** dialog. Enter your prompt, choose RELATE as the TYPE, and click OK. When the **Conditions for Related Form to Be Active** dialog appears, select ANYTIME from the options and click OK. The next dialog is the **Related View Choice** dialog, where you can choose to create a new related view or relate to existing view. By choosing to create a new related view, you are creating a new view; this allows you to get another screen on which to create more or different fields. By choosing to relate to existing view, you are referring to another view or another form that already exists.

Move Related Field buttons (*MakeView*)

Hold down the SHIFT key and the left mouse button and drag the field to the desired location.

Convert a file from Epi Info, Version 6, format to Epi Info 2000 (*MakeView*)

Select FILE from the menu and IMPORT. When the **Select Import File** dialog comes up, select the .REC file from Epi Info, Version 6, to be imported. Click OPEN and choose a destination file (.MDB) in Epi Info 2000. Accept or edit the name of the new table to be created.

Make a view of an existing database (*MakeView*)

Select FILE on menu, click on MAKE VIEW, and select MAKE NEW VIEW FROM EXISTING DATATABLE.

Compare two dates to obtain an age or duration (*MakeView*)

Bring up the Check environment by clicking on the PROGRAM button to the left of the view. Click on the arrow under **Choose Field Where Action Will Occur** to pull down a list of fields. Click on the VARIABLES tab, then select the ASSIGN command. Display the list of available fields to the left of the "=" sign by clicking on the arrow. Choose the field in which the result of the calculation should appear. Type in the formula that you would like to use to the right of the "=" sign, or choose the field names from the pull-down list if you do not know them in advance. Click on the SAVE button and click on the OK button and answer "YES" to the prompt about saving check code to exit. Now choose ENTER DATA from the FILE menu. After you have entered both dates, the results of the calculation should appear in the AGE field. (An example is found under *Checking and Controlling Data Entry* in the Guided Tour of the ***MakeView*** program.)

Specify Legal Values (*MakeView*)

When creating a new field, you can choose legal values for that field by clicking the LEGAL VALUES button in the **Field** dialog. Choose CREATE NEW, and enter suitable values in the list that appears, pressing Enter after each to move to a new blank line. Click OK, then OK again in the **Field** dialog.

Do Skip Patterns (*MakeView*)

Skip patterns can be done by changing the TABORDER or by using the GOTO command in the PROGRAMS section. To change the TABORDER, go to FIELDS on the menu and select TABORDER. When the **TabOrder Assistant** comes up, use the UP and DOWN buttons to change the order of your fields. To use the GOTO command, click on PROGRAMS on the Page Name Panel. Click on the arrow under **Choose Field Where Action Will Occur** to pull down a list of fields. You can use the GOTO command under the FIELDS tab, or you can type in a GOTO program using IF statements in the **Program Editor**.

Do automatic coding between fields or into the same field (*MakeView*)

Open the **Field** dialog, enter your prompt, and click CODES. Select the field(s) to receive codes. Existing tables can be used, or a new table can be created to set up code fields.

Set up conditional related forms (*MakeView*)

Right click on the screen to open the **Field** dialog. Enter your prompt, choose RELATE as the TYPE, and click OK. When the **Conditions for Related Form to Be Active** dialog appears, select ONLY WHEN CERTAIN CONDITIONS ARE TRUE from the options. When the **Access** section appears, choose which variables or values are to be true before accessing this page, then click OK. The next dialog is **Related View Choice**, where you can choose to create a new related view or relate to the existing view. By choosing to create a new related view, you are creating a new view, which allows you another screen to create more or different fields. By choosing to relate to existing view, you are referring to another view or another form that already exists.

Change the Prompt, Name, or Type of a Field in a View (*MakeView*)

If MakeView has not yet made a data table, or there are no records in the data table, names, field types, and patterns can be changed by right-clicking on the screen prompt for the field. When the field dialog appears, change the desired characteristic, and click OK. Then Save the view, either by choosing SAVE from the FILE menu or in response to a reminder. If a data table exists and contains data, you can change the *prompt* by popping up the field dialog in ***MakeView*** as above. If you try to change the field *name*, however, you will receive the following message:

Changing the name of a data variable could cause problems if there is more than one View linked to this file, as during multiuser entry on a Local Area Network. If you are sure this is not the case, please give the new name below and choose "Change Variable Name."

Change Variable (Variable name) to ________________________________

[ ] Play it safe and create a new field rather than renaming the old one

[Change Variable Name]   [Cancel--Do not change name]

If you change the field type or pattern, the following message will appear:

Changing the data Type, from (old type) to (new type) can cause problems if the existing data type is not compatible with the new type. Therefore, MakeView will create a new field. Please give a name for the new field that is different from the old name.

Name for New Field ________________________________

[ ] Copy data from old field to new field, discarding values that are incompatible

[OK]   [Cancel]

ENTER

Change the values displayed in Yes/No fields (*Enter*)

Click on OPTIONS on the menu.

Move from record to record (*Enter*)

Move to the first or last records by using the double arrows in the lower left panel of the **Page Name** section. By using the single arrows, you can move one record at a time. Clicking the NEW button brings up the next available empty record.

Save a page (*Enter*)

Pages are saved automatically. The current page is saved when you move to the next record or page.

Save a record (*Enter*)

Saving records can be done by clicking on SAVE DATA on the **Page Names** panel or by clicking on NEW. (Records are also saved automatically. Once you input your data into a field the information is saved when you move to the next record.)

Move among parent and related forms (*Enter*)

The HOME and BACK buttons can be used to move to and from parent and related forms. These buttons do not appear until you have clicked on your related button. Once you have clicked on the desired button, another form comes up. This form contains the HOME and BACK buttons. The HOME button will take you back to the parent page. The BACK button can be used to take you back to the last page you were at or to the parent page.

ANALYSIS

Read Dbase, Paradox, FoxPro, Excel, and Microsoft Access database files (*Analysis*)

Click on READ on the **Analysis Commands** panel. From the **Project Format** dialog, choose the file you want to read. If it is an Access/Epi2000 file, view the tables within the .MDB file. (See READ in *Commands* chapter.)

Read an ODBC database such as SQL Server or Oracle (*Analysis*)

Read from one file format and write to another (*Analysis*)

Click on READ from the **Analysis Commands** panel. Choose the **Project Format** that your file is in. After reading the file, click on the WRITE command, and from the list of output formats (file types), choose the one to write to. Give your file a name, using the LOOK IN dialog to choose the desired directory. Click on OK to write the file. (See WRITE in *Commands* chapter.)

Select records (*Analysis*)

To select particular records based on criteria that you specify, click on SELECT from the **Analysis Commands** panel. Enter the condition that you want to specify, click OK, and choose the LIST command to show only those records that meet that condition. (An example is found under *SELECT Command* in the Guided Tour of the ***Analysis*** program.)

Cancel a previous selection (*Analysis*)

Click on the CANCEL SELECT command from the **Analysis Commands** panel. (See SELECT in *Commands* chapter.)

Compose groups in a study (*Analysis*)

Process data from a questionnaire study (*Analysis*)

Click on FREQuencies from the **Analysis Commands** panel. In the **FREQ** dialog that appears, select one or more variables, and click OK. FREQ produces a table from the table(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 using 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. (See FREQ in *Commands* chapter.)

Combine data in more than one table (*Analysis*)

Click on RELATE from the **Analysis Commands** panel. To use RELATE, a dataset must have been made active with the READ command. The table to be linked must have a key field (such as HOUSID) that has identifiers relating the records in the two tables. After issuing the RELATE command, the variables in the related table may be used as though they were part of the main table. (See RELATE in *Commands* chapter.)

Process only certain columns (variables) (*Analysis*)

Click on LIST from the **Analysis Commands** panel. In the **LIST** dialog that appears, choose one or more variables to be listed. (See LIST in *Commands* chapter.)

Obtain an Odds Ratio, Relative Risk, Chi Square and Fisher Exact test (*Analysis*)

Obtain Exact statistics (*Analysis*)

Do an ANOVA test (*Analysis*)

Click on MEANS from the **Analysis Commands** panel. Input the name of a numeric variable containing data to be analyzed and the name of the variable that indicates how groups will be distinguished. Depending on the size of the table and whether the data are normally distributed, MEANS provides the equivalent of ANOVA for two or more samples. (See MEANS in *Commands* chapter.)

Do a Student's t-Test (*Analysis*)

Click on MEANS from the **Analysis Commands** panel. Input the name of a numeric variable containing data to be analyzed and the name of the variable that indicates how groups will be distinguished. Depending on the size of the table and whether the data are normally distributed, MEANS provides the equivalent of STUDENT'S T TEST for two samples. (See MEANS in *Commands* chapter.)

Add population data to a dataset (*Analysis*)

Click on DEFINE from the **Analysis Commands** panel. DEFINE allows creation of new variables for use in ANALYSIS. Type in Population as a variable name and choose PERMANENT. PERMANENT variables are stored in the system registry and retain any value assigned until the value is changed by another assignment or the variable is UNDEFINED. (See DEFINE in *Commands* chapter.)

Change the appearance of a graph (*Analysis*)

Click on GRAPH from the **Analysis Commands** panel and input your data into the **Graph** dialog. Select a different graph by clicking on the **Graph Type** pull-down arrow and selecting OK. If you want to change the appearance of your graph after it has come up, click on MENU, select TOOLS, then TOOL BAR. When the toolbar appears, click on the sixth icon, the **Change Gallery Type** icon. (See GRAPH in *Commands* chapter.)

Graph more than one variable (*Analysis*)

Click on GRAPH from the **Analysis Commands** panel and input your data into the **Graph** dialog. If the word COMBINE is present, the graph is combined with whatever graph command follows, and the second is overlaid on the first. This may be repeated until several graphs are displayed in overlaid fashion. (See GRAPH in *Commands* chapter.)

NutStat

Open or create a data file (*NutStat*)

To open a file, click FILE on the menu and select OPEN. To create a new file, click FILE on the menu and select NEW. When creating a new file, place a check mark in the **New File** dialog at bottom of the **Select a Table** dialog.

Set up units of measurement (*NutStat*)

Select FILE from the menu and choose CUSTOMIZE. From the **Customize Anthropometric Data** dialog, click the UNITS tab. Choose international or US reference standards

Select FILE from the menu and choose CUSTOMIZE. From the **Customize Anthropometric Data** dialog, click the DISPLAY tab.

Assess height and weight (*NutStat*)

Enter the height and weight on the main screen of NutStat. Click RECUMBENT if the child was measured lying down. Statistics will appear on the right side of the screen, comparing the child with others of the same sex and age in the growth reference curves selected in the Customize page.

Assess head circumference (*NutStat*)

Enter the head circumference on the main screen of NutStat. Click RECUMBENT if the child was measured lying down. Statistics will appear on the right side of the screen, comparing the child with others of the same sex and age in the growth reference curves selected in the Customize page.

Assess arm circumference (*NutStat*)

Enter the arm circumference on the main screen of NutStat. Click RECUMBENT if the child was measured lying down. Statistics will appear on the right side of the screen, comparing the child with others of the same sex and age in the growth reference curves selected in the Customize page.

Use Nutstat in combination with another data entry view (*NutStat*)

Use Nutstat in combination with an entirely different database program (*NutStat*)

EPI MAP

Display a map (*Epi Map*)

Choose a boundary layer (*Epi Map*)

Select FILE from the menu, then MAP MANAGER. In the **Map Manager** dialog, click ADD MAP LAYER from the LAYERS tab and select a shape file. (An example of this is found in the Guided Tour of the ***Epi Map*** program.)

Prepare a data file for use with a map (*Epi Map*)

Display values from a data file in a map using *Analysis* (*Epi Map*)

Display values from a data file using *Epi Map* (*Epi Map*)

Change colors or patterns (*Epi Map*)

Click the GENERAL tab in MAP MANAGER to set a background color.

Make a choropleth (color/pattern) map (*Epi Map*)

Make a dot density map (*Epi Map*)

Plot cases identified by x and y coordinates (*Epi Map*)

Load an image (*Epi Map*)

Combine an image and a boundary file (*Epi Map*)

Draw a map (*Epi Map*)

Add annotations (*Epi Map*)

Control the Legend (*Epi Map*)

Display Rates (*Epi Map*)

VIZDATA

See data in spreadsheet format (*VizData*)

See variables in a database (*VizData*)

Click FILE from the menu, choose OPEN DATABASE, and select the desired file type.

Delete a table or file (*VizData*)

Rename a table or file (*VizData*)

Compact a database (*VizData*)

Click FILE from the menu and choose COMPACT MDB. Select the type of file to be compacted and choose a file.

Repair a damaged database (*VizData*)

Click FILE from the menu, choose REPAIR MDB, and select the files to be repaired.

Samples

Examples Included With the System

Oswego: An Outbreak Investigation

Found in: SAMPLE.MDB:viewOSWEGO

Refugee: A Group of Related Data Views for Families, Individuals, Evaluations, and Specific Conditions

Found in: REFUGEE.MDB

Surveillance: A Menu and Series of Data Input Forms

Found in: SURVEIL.MNU and SAMPLE.MDB:viewSURVEILLANCE

Statistical Examples

SAMPLE.MDB

NutStat

A Nutritional Anthropometry Program

Nutstat is a program for recording and evaluating measurements stature, weight, head and arm circumference for children and adolescents. It can be run as a standalone program or as part of an Epi Info 2000 questionnaire view.

The program calculates percentiles, numbers of standard deviations from the mean (z-scores), and in some cases, percent of median, using data from the following sources:

- The CDC/WHO 1977/1985 reference curves for age, sex, height, and weight
- The WHO reference data for arm circumference
- (When available) The 2000 U.S. National Center for Health Statistics reference curves for age, sex, height, weight, head circumference, and body mass index. The release date for the new NCHS data has not yet been announced, and the program does not yet include the data.

The main screen of NutStat is configurable to use either English or Metric units and to perform calculations with the desired reference data set. Measurements entered on the screen are automatically stored in a Microsoft Access database and can be retrieved by patient name or identification number. Data from one person or from the entire population can be displayed graphically in several formats. NutStat databases can be read by the Analysis program to produce summary tables, lists, frequencies, and graphs, or to perform more complex data manipulations.

Existing data in Microsoft Access files can be imported to produce a standard NutStat database with appropriate statistics. An Add-Statistics feature adds the results of calculations to external Microsoft Access data files that are not necessarily in the NutStat format.

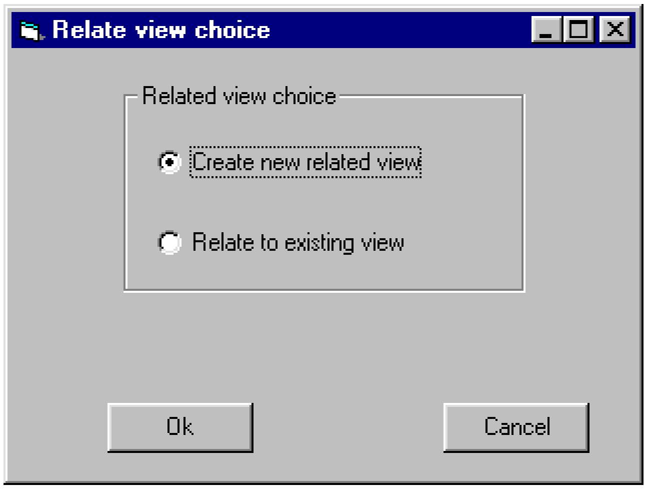
Operation of Nutstat

Configuring Nutstat

Nutstat as Part of an Epi Info 2000 Questionnaire View

NutStat can be run from within an Epi Info questionnaire view as though it is a related table. Nutstat is represented by a button on the questionnaire that can be clicked to bring up the nutritional entry screen.

To set up the relationship and the button, right click on the screen of a questionnaire in MakeView, enter a name for the button to be created ("Growth," for example), and then click on the button labeled CREATE RELATED VIEW. In the next dialog, selection criteria can be entered so that the button is active only under certain conditions, or it can be made active at all times. A reasonable condition might be that an AGE field contain a value between 0 and 18 years, since most of the calculations in NutStat are limited to this age group.



The following sections were written by Kevin Sullivan, PhD, ,,,,,,,, Department of Pediatrics, School of Medicine, and Department of Epidemiology, Rollins School of Public Health, Emory University, Atlanta, Georgia.

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]

NCHS is in the process of developing new growth reference curves for the US. When these become available, they will be incorporated into the NutStat program so that both sets of curves will be available.

To calculate 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 or otherwise confirmed as correct.

Using NutStat for Anthropometric Calculation

What the Anthropometric Calculation Program Does

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	Index Flagged			
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.

Other Considerations

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]

Other Anthropometric Software and Sources of Information

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:

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.
Nutrition Unit
World Health Organization
1211 Geneva 27
Switzerland

Acknowledgments

Special thanks to Phillip Nieburg, M.D., M.P.H., Norman Staehling, M.S., Ronald Fichtner, Ph.D., and Fredrick Trowbridge, M.D., Centers for Disease Control and Prevention, for their assistance.

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.

Commands

Detailed Descriptions of Commands in the Epi Info Programming Language

Overview

Epi Info 2000 is easy to operate in interactive mode, but more complex operations require saving the steps as programs. Programs (similar to "scripts" in some software) can be used to set up Menus, guide and limit the data entry process, restructure data, and do analysis. Template files save the settings for Graphs and Maps so that similar graphics can be produced more than once.

In ***MakeView*** and in ***Analysis***, the process of programming consists of interacting with a series of dialogs that produce the actual program statements. Experienced users may want to edit the statements or type them directly in the program editor. For this reason, the details of command syntax are provided in this chapter. A definition of each command and its operation is given. Because a single command such as EXECUTE may be found in ***MakeView***, ***Analysis***, and the Menu environments, each command is presented in a single section of the text in this chapter, with notation of differences that may exist in its implementation in the various programs. Each section has check marks at the top that clearly show which program allows use of which commands.

Functions and operators appear within commands and are used for such common tasks as extracting a year from a date, combining two numeric values, or testing logical conditions. Because there are many functions and operators, they are presented separately in the Functions and Operators Chapter.

Epi Info 2000 Commands

PROGRAMS
ANALYSIS.EXE
ENTER.EXE
EPI2000.EXE
EPICHART.EXE
EPIMAP.EXE
MAKEVIEW.EXE
NUTSTAT.EXE

VARIABLES
ASSIGN
DEFINE
UNDEFINE

DATA
CANCEL SELECT
CANCEL SORT
COPY
IF THEN ELSE
IMPORT FROM EPI6
LIST
MERGE
READ
RECODE
RELATE
SELECT
SORT
WRITE

USER INTERACTION
DIALOG
HELP
MENU

OUTPUT
FREQ
GRAPH
MAP
MATCH
MEANS
TABLES
TITLE
TYPE

STATISTICS
KMSURVIVAL
LOGISTIC
REGRESS

MOVEMENT
AUTOSEARCH
GOTO

SYSTEM
EDITVIEW
EXECUTE
PRINT
PRINT VIEW

SCREEN
CLEAR
DIALOG
HELP
HIDE
UNHIDE

PROPERTIES OF VIEW
CODES
COMMENT LEGAL
ENCRYPTION
LEGAL
MUSTENTER
PASSWORD
RANGE
READ ONLY
REPEAT
SOUNDEX

MENU
BUTTON
MENUITEM
POPUP
SCREENTEXT

COMMAND BLOCKS
@@
BLOCK
COMMENTS
EXECUTE
EXIT
HELP
MENU
PICTURE

SYSTEM CONFIGURATION
AUTOSEARCH
DOSWIN
QUIT (EXIT)
RUNPGM
SET
SET BUTTONS
SET PICTURE
SETWORKDIR
SYSINFO

CHECK Programs: General Features

EPI2000.EXE

Program

Description

The Epi Info Menu

Command Line Parameters

<Path and Name of a menu (.MNU) file>

Syntax Element	Description

Comments

Buttons, menu items, screen text, background pictures, and procedures to be executed from the menu are controlled by the MNU file and can be programmed by the user.

Examples

Epi2000.MNU and Surveil.MNU, supplied with the programs.

MAKEVIEW.EXE

Program

Description

The Design Environment for View Questionnaires

Command Line Parameters

Syntax Element	Description

Comments

This is an ActiveX executable module that can be run from within the ENTER program and vice versa. Views are saved as tables with a specific format within an MDB (Microsoft Access) database.

Examples

ENTER.EXE
Program

Description

The Data Entry Program

Command Line Parameters

`<Path and Name of MDB file>:<View Table Name>`

Syntax Element	Description

Comments

This is an ActiveX executable module that can be run from within the ***MakeView*** program and vice versa. When a View is opened that has no corresponding data table, the data table is created automatically by ***Enter*** (if the user agrees).

Examples

ANALYSIS.EXE
Program

Description

A general-purpose data management and analysis program that reads many types of files and will produce lists, frequencies, tables, epidemiologic statistics, graphs, and maps.

Command Line Parameters

<MDB path and Name>:<PGM Name>

or

<PGM Path and Name>

Syntax Element	Description

Comments

Analysis can be run from the menu or from other programs. If the name of a program (.PGM) is given as a parameter, the PGM is run immediately. Hence, PGMs can be stored in a database or as DOS text files with the extension PGM. For example:

```
ANALYSIS  SURVEILLANCE.MDB:WEEKLY.PGM
```

might run a weekly report program stored in the database called SURVEILLANCE.

PGM files can also be saved in text format, in which case they are run by supplying a suitable path and program name, as in:

```
ANALYSIS c:\Epi2000\Weekly.pgm
```

Examples

```
ANALYSIS  SURVEILLANCE.MDB:WEEKLY.PGM
ANALYSIS c:\Epi2000\Weekly.pgm
```

From the Menu

Values can be passed from the menu directly to ***Analysis***. The syntax for executing a program is:

EXECUTE Analysis ‘<Filename>”: “<Program>”

NUTSTAT.EXE

Program

Description

A program for Nutritional Anthropometry

Command Line Parameters

<UniqueID of parent record>

Syntax Element	Description

Comments

Can be run from a View in ***Enter*** or from other databases by passing the record ID of a parent record. Records in ***NutStat*** are for body measurements on a particular date.

Examples

EPIMAP.EXE

Program

Description

A mapping program built around MapObjects from Environmental Sciences Research Institute (ESRI)

Command Line Parameters

<Path and Name of .MAP file>

Syntax Element	Description

Comments

When run as an ActiveX executable object, many properties can be set. NOTE: CDC's license for MapObjects does not permit *EpiMap* to be used independently from Epi Info.

Examples

Special Features of Check Commands

Check commands are optional, and not all fields need to have blocks of commands in a particular view.

Check commands must be placed in a named block of commands beginning with the name of a variable in the database and ending with the word END. Special blocks are provided to execute commands before or after entering a View, Record, or Page. These are:

BEFORE VIEW, AFTER VIEW, BEFORE RECORD, AFTER RECORD, BEFORE PAGE, and AFTER PAGE.

Comments, preceded by an asterisk ("*"), may be placed within blocks of commands. For commands not having a "Command…end" structure, the line continuation character ("\" at the end of the line) is used to indicate that a command continues on the next line.

Commands in a block are activated either before or after an entry is made in the field. The default is that the commands are performed after an entry has been completed with <Enter>, <PgUp>, <PgDn>, or <Tab>, or another command causes the cursor to leave the field.

Check commands for each field are stored in the VIEW in a record associated with a particular field or with one of the special BEFORE/AFTER blocks mentioned above.

The commands are inserted automatically through interaction with the PROGRAM dialog boxes, chosen from a tabbed display of commands. Text versions of the commands appear in the Program Editor as they are generated by the dialogs, and can be edited there if desired.

	Menu	Check	Analysis
@@ Command	✓		

Description

Replaces a variable name with its current value

Syntax

@@<VarName>

Syntax Element	Description
<VarName>	Variable name

Comments

The name of a variable created with ASSIGN or DIALOG can be embedded in a menu command, a MenuItem, Screentext, or a Button description. When the command is executed, the Value of the variable will be substituted for @@VarName.

Users of Epi Info 6 will note that that @@ replaces @ to avoid confusion with Internet addresses that frequently contain a single @ sign.

Examples

```
Assign FavoriteURL =  "WWW.CDC.GOV"
LINKTO @@FavoriteURL
```

If all goes well (and you have an Internet connection and browser), these two commands will connect to the CDC web page. Of course, LINKTO WWW.CDC.GOV will do the same thing, but for repeated use, allowing users to set the variable value, or passing a value to another program, the embedded variable approach can be very helpful.

Program-Specific Features

	Menu	Check	Analysis
Assign			
Command	✓	✓	✓

Description

Syntax

Analysis

<*Variable*> **=** <Expression>

Check

ASSIGN <*Variable*> = <Expression>

Menu

ASSIGN <*Variable*> = <Expression>

Syntax Element	Description
<*Variable*>	A variable in a database or a defined variable created in a program.
<Expression>	Any valid expression in Epi Info 2000

Comments

Assigns the value of an expression to a variable. The variable may be a database variable in a View or Data Table or a user-defined variable, created by the DEFINE command in a program. Temporary variables must be DEFINEd; they are not created automatically as in previous versions of Epi Info, except in the menu program. They are flexible in accepting any type of data (text, numeric, or date). Conversion of data type (from numeric to text, for example) will be done automatically, if necessary.

Examples

```
County = "Yellow Medicine"
DURATION = 01/01/88 - DATEONSET
                               (Duration will be in days.)
DURATION = YEARS(01/01/88, DATEONSET)
                               (Now it will be in years)
ASSIGN ILL = (-)
GROUP = 1                     (If Group is a numeric variable)
GROUP = "1"                    (If group is a text variable)
```

For methods of dealing with a part of a text or date value, see FUNCTIONS.

Program-Specific Features

Menu

Available for use in menu command blocks

This is one way to assign a value to a variable. The Value is placed in the file called EPIINFO.INI; therefore all variables created in the menu file will be permanent in any of the Epi Info programs. In the menu environment, there is no need to first DEFINE a variable; they are defined automatically by the ASSIGN or DIALOG commands, and will accept either text or numeric data. Text should be enclosed in quotation marks, although this is not always necessary. Variable names cannot contain spaces or punctuation, but both are permitted in values. In the menu program.

Functions available for ***Analysis*** and Check commands are not available for menu.

Analysis

If the right side of the assignment does not contain a data variable (one in a database table), or a variable that depends on a data variable, then the assignment will be made immediately. Thus:

```
DEFINE YEAR PERMANENT
YEAR = 2000
```

However:

```
DEFINE INCUBATION
INCUBATION = ONSETDATE-EXPOSUREDATE
```

contains two VIEW variables, ONSETDATE and EXPOSUREDATE. INCUBATION will only be calculated during processing of the current dataset. It will be calculated for each record during processing and may be used like a dataset variable in TABLES, FREQ, and GRAPH commands.

Prior to and after processing a dataset, INCUBATION will have a “missing” value, although it could be assigned a value with another statement such as INCUBATION = 999.

The value is calculated each time a record that meets the conditions of SELECT is read from the dataset. Any legal expression can be used that combines variables or specific values and operators like +, -, *, /, =, <, >, AND, OR, and NOT. See OPERATORS for the complete list.

Menu	Check	Analysis
	✓	

AUTOSEARCH

Command

Description

AUTOSEARCH causes ***Enter*** to search for one or more matching records. If a match is found, the user can choose to display and edit the matching record(s) or to ignore the match and continue to enter the current record.

Syntax

```
AUTOSEARCH <VarName(s)>
```

Syntax Element	Description
<VarName(s)>	A group of variables to look for.

Comments

If more than one record is found, the 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, a single row can be selected by clicking to the left of the first field or by using the up and down arrow keys, and a particular record selected for editing by pressing <Enter> or selecting OK. To avoid editing any of the matches, press <Esc> or the Cancel button. The fields displayed as the result of a LIST search are determined as follows:

1. If a single field is the key, that 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.

AUTOSEARCH will use the Soundex keys on all search fields that have them.

Examples

AUTOSEARCH can be given a list of fields to use for the search, rather than relying on the order of the fields in the view. If a view has the fields:

FIELD1 ...

FIELD2 ...

FIELD3 ...

FIELD4 ...

FIELD5

```
FIELD3
  AUTOSEARCH FIELD2 FIELD3
END
```

then AUTOSEARCH will ignore the value of FIELD1 and search only on the values of FIELD2 and FIELD3.

Program-Specific Features

BEEP

Command

	Menu	Check	Analysis
		✓	✓

Description Generates a sound.

Syntax BEEP

Comments

Examples

Program-Specific Features

Menu	Check	Analysis
✓		

BUTTON(S)

Command

Description

Creates a button on the screen with the caption ItemText and the locations given as percentages of screen width and height. When the button is clicked with the mouse (left mouse click), the commands in the CommandBlock named are executed. If a text tip is given, this appears when the mouse cursor hovers over the button for several seconds.

Syntax

BUTTON "<Caption>", <Block>,
<Xlocation>,<YLocation>,"<Dynamic text>"

Syntax Element	Description
<Caption>	Text to be displayed in the button
<Block>	The name for the block to be executed
<X-Location>	A number from 1 to 99 representing the distance from the left side of the window in percentage points.
<Y-Location>	A number from 1 to 99 representing the distance from the top of the window in percentage points.
<Dynamic Text>	Text to be displayed when the cursor is pointing to that button

Comments

Hot keys may be specified by preceding them with an ampersand (&), in which case the Alt key plus the hot key will activate the button. Hot keys should not be duplicated between POPUP items (menu headings) and BUTTONs.

Examples

```
*BUTTONS
*Begin
BUTTON "Ma&keView", MakeView, 5,50,"Make or Edit a Questionnaire"
BUTTON "E&nter Data", Enter, 5, 60, "Enter and store data, using\
a View"
```

Program-Specific Features

Menu	Check	Analysis
		✓

Cancel Sort/Select

Command

Description

Cancels a previous SORT or SELECT command.

Syntax

- SORT
- SELECT

Comments

Examples

```
Read 'C:\EPI2000\SAMPLE.MDB': viewOswego
Select ill = (+)
List *
Select
List *
```

Program-Specific Features

Menu	Check	Analysis
	✓	

CLEAR

Command

Description

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. More than one field may be specified. CLEAR will frequently be followed by a GOTO, to put the cursor in position for further entry after an error.

Syntax

CLEAR <VarName(s)>

Syntax Element	Description
<VarName(s)>	A valid name for a variable

Comments

CLEAR will delete the value only for the current record. Clear cannot be used in GRID Tables.

To CLEAR a variable in ***Analysis***, it is necessary to ASSIGN the variable to null (.)

Examples

```
If EndDate < StartDate then
        CLEAR EndDate
        GOTO EndDate
END IF
```

Program-Specific Features

Analysis

To clear a variable in ***Analysis***, it is necessary to ASSIGN null (.) to it

Menu	Check	Analysis
		✓

CLOSEOUT Command

Description

Closes the output specified in a previous ROUTEOUT command.

Syntax

CLOSEOUT

Comments

Examples

```
ROUTE Outbreak1.HTM
Tables * ILL
Tables AGE SEX ILL
MEANS AGE ILL
CLOSEOUT
```

Program-Specific Features

Menu	Check	Analysis
		✓

CMD Command

Description

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 <CommandName>
        [Commands]
        [Commands]
END
```

Syntax Element	Description
<CommandName>	A string that will execute the defined block of instructions

Comments Whenever CommandName appears elsewhere in the .PGM file, the block of commands is executed.

Examples

```
CMD HELLO
   DIALOG “Welcome to Epi2000” TITLETEXT = “Greetings”
END
HELLO
HELLO
HELLO
```

Program-Specific Features

Menu	Check	Analysis
✓		

Command Blocks

Command

Description Command blocks called from the MENUITEM and BUTTON commands

Syntax

```
<Block Name>
Begin
       Command
       Command
End
```

Syntax Element	Description
<Block Name>	A Name containing only text characters and/or digits. Special characters such as (, #, $, &, and blank spaces are not allowed as part of the block name.
Begin/ End	Delimiters for the command block.

Comments Commands in Command Blocks are executed when a menuitem or button that refers to the block is chosen by the user

Examples

```
MAKEVIEW
Begin
         EXECUTE MakeView.EXE
End
```

Menu

Program-Specific

Features

	Menu	Check	Analysis
COMMENTS (*) Command	✓	✓	✓

Description

Syntax

* <Text>

Syntax Element	Description
<Text>	Any comment

Comments

The (*) character must be the first non-blank character on a line to be recognized as comment mark. In other words, comments must be on separate lines from commands. Comments of more than one line must have the comment symbol at the beginning of each line.

Examples

```
*End of Menu
*ToolTips are included with the Buttons
```

Program-Specific Features

	Menu	Check	Analysis
DEFINE Command	✓	✓	✓

Description

DEFINE allows creation of new variables for use in ***Analysis***. Variables in Epi Info 2000 do not have predefined types; they will accept any type of data that is assigned with the assignment command ("=").

Syntax

DEFINE *<Variable>* [Type]

Syntax Element	Description
<Variable>	The name of the variable to be created
[Type]	Global/Permanent/Standard

Comments

PERMANENT variables are stored in the EPIINFO.INI file or system registry and retain any value assigned until the value is changed by another assignment or the variable is UNDEFINEd. They are shared among Epi Info programs and persist even if the computer is shut down.

Generally GLOBAL variables are used for operations not involving FREQ, LIST, TABLES, and MEANS. Standard defined variables are used as temporary variables behaving like variables in the database. They lose their values and definitions at the next READ statement.

Examples

```
DEFINE Var1
DEFINE Var2 GLOBAL
DEFINE Var2 PERMANENT
```

Program-Specific Features

Menu

The ASSIGN command in menu command blocks automatically defines a variable if it does not already exist. The variable is automatically of the PERMANENT type.

Check Code

The DEFINE command can create standard, GLOBAL, or PERMANENT variables. Standard variables retain their value only within (all pages of) the current view and are reset when a new record is loaded. GLOBAL variables retain values across related Views and even when a new View is opened by the program, but are removed when the ***Enter*** program is closed.

Analysis

Standard Variables—those not defined as GLOBAL or PERMANENT—exist only until the next READ command. GLOBAL variables remain in effect as long as ***Analysis*** is running and can be used to carry values from one PGM to another. Permanent variables can be defined only from the text editor. If a permanent variable is created and no value is assigned, it will be deleted before closing ***Analysis***.

	Menu	Check	Analysis
DIALOG Command	✓	✓	✓

Description

The Dialog command provides interaction with the user from within a program. Dialogs can display information, ask for and receive input, and offer lists for making choices.

Syntax

Menu

- DIALOG "<*Text Prompt*>" {TITLETEXT="<Title>"}

Check code

Analysis

- **DIALOG** "<*Text Prompt*>" {**TITLETEXT**="<Title>"}
- **DIALOG** "<*Text Prompt*>" **DATABASES**
- **DIALOG** "<*Text Prompt*>" <*Variable*> [<Entry type>] <*table*> <*Var1*> {**TITLETEXT=**"<Title>"}
  - **DIALOG** "<*Text Prompt*>" <*Variable*> "<Value1>", "<Value2>", "<Value3>", … ,"<Value$_n$>" {**TITLETEXT=**"<Title>"}

Syntax Element	Description
<Text Prompt>	Text to be displayed as message
<Variable>	Variable to store value entered.
[<Entry Type>]	A reserved word that defines the type of input to be taken and stored the variable. The default value is number/date. **TEXTINPUT** will define variable as text **YN** will define variable as Boolean. **DBVALUES** will display a combo box in which the user can select the desired value. **DBVIEWS** **DBVARIABLES**
<Title>	Text to be used as windows title.

Comments

The first form of DIALOG places a dialog box on the screen, using the text provided, with an "OK" button.

The second form of DIALOG displays the text prompt and provides buttons “Yes,” “No,” and “Cancel.” IF YN is specified, the buttons are presented as “Yes,” “No,” and “Cancel,” in the appropriate language, and the specified variable is set to “+”, “-”, or a missing value.

If YN is not specified and a variable name is given, an entry field for user response is provided, with OK and Cancel buttons. The variable is assigned the value of the user’s input, with a missing value if Cancel is chosen. If a Format Pattern is given, it controls the type and format of the entries that will be allowed, typically as numbers or text. For legal Format Patterns, see the FORMAT function.

Examples

```
DIALOG "Enter Value " var1 TITLETEXT="Dialog 1"
DIALOG " Enter Value " var1 DATABASES
DIALOG " Enter Value " var1 DATABASES TITLETEXT="
Dialog 1"
DIALOG " Enter Value " var1 "aaa", "bbb", "ccc", "ddd"
TITLETEXT=" Dialog 1"
DIALOG "please select database" var1 DATABASES
TITLETEXT="database"
```

Program-Specific Features

EndBefore

Command

Menu	Check	Analysis
	✓	

Description

ENDBEFORE divides Check commands to be executed before data entry from those executed after entry.

Syntax

ENDBEFORE

Comments

These commands will control actions so that they occur before or after accessing a View, record, page, or field.

The default time for execution of commands associated with a variable is “AFTER ENTRY.” If execution is to occur before entry, place the relevant commands in a block called BEFORE ENTRY … END within the block of commands for that variable. ENDBEFORE must be placed starting in the first position of the line. No other command can be placed in the same line.

If end before is not present, Epi Info 2000 will execute all commands after the condition.

ENDBEFORE can be used in pages, views, entries, and records.

Examples

The following commands represent the Check code for a variable called Demo1

```
Dialog ‘This is Before entry’ TITLETEXT = “Dialog1”
ENDBEFORE
Dialog ‘This is After entry’ TITLETEXT = “Dialog 2”
```

Program-Specific Features

ENDBEFORE is only used in Check code.

Menu	Check	Analysis
✓	✓	✓

EXECUTE(LINKTO OR ACTIVATE)

Command

Description

Executes a Windows or DOS program--either one explicitly named in the command or one designated within the Windows registry as appropriate for a document with the file extension that is named. This provides a mechanism for bringing up whatever word processor or browser is the default on a computer without first knowing its name.

Syntax

- EXECUTE ExecutableFile.EXE
- EXECUTE Filename.FileExtension

Syntax Element	Description
<ProgramName>	Path and program name for EXE and COM files Filename for programs registered in windows URL address.

Comments

This command is so flexible that we had to give it three names, all synonyms. Use whichever one appeals to you.

If the name of an executable program, such as ENTER.EXE or MYBATCH.BAT or even DOIT.COM is given, the program will be run in a separate window. When the program terminates, the window will be closed.

If the name given is not a program, but a file with an extension (the 3 characters after the ".") registered by Windows for displaying the document, the correct program to display the file will be activated. For example, WRITEUP.DOC might cause Microsoft Word © to run and load the file on one computer. On another machine, however, .DOC might correspond to Corel WordPerfect©. Usually .TXT will run NOTEPAD.EXE© or WORDPAD.EXE©, and image files will appear either in a browser or in a graphics program on your computer. An .HTM file will bring up the default browser on your computer.

The exciting thing about this form of EXECUTE is that you do not need to supply the location or even the name of the program that will be run. These details are stored in the Windows registry for common file extensions. It is as though you said, "I want to drive a nail," and the operating system hands you a hammer. You do not have to search for the hammer and then the nail.

The same concept applies to Internet addresses (URLs). Give a URL (Universal Resource Locator) that ends in .HTM and Windows brings up the default browser on your computer, connects to the Internet (if possible), and goes directly to the site indicated. It is the beginning of “the World as your disk drive.” With this device and an Internet connection, you could set up a menu with buttons for websites in neighboring countries or states, favorite search engines, or whatever you need for daily work, thus making contact with these information sources just a mouse click away.

Examples

Program-Specific Features

Check Command

In the Check command execute can be placed in any variable, usually a command button. In the command execute is associated to a command button, it should not be placed before entry.

Menu

Alternative syntax (Link, Activate) is available only for the menu.

EXIT (QUIT)

Command

	Menu	Check	Analysis
	✓	✓	✓

Description

Closes current data files and terminates the current program, closing ***Analysis***.

Syntax

EXIT

Syntax Element	Description

Comments

EXIT and QUIT have the same function

Examples

Program-Specific Features

Menu

Closes the current menu. EXIT does not necessarily close programs that have been started by the menu in other windows. They must be closed by the user, if this is desired.

Analysis

Exit will stop the execution of a program and close ***Analysis***.

FREQUENCIES Command

Menu	Check	Analysis
		✓

Description

FREQ produces a table from the table(s) specified in the last READ statement, showing how many records have each value of the variable. Confidence limits for each proportion are included, and other statistics are provided if the variable is numeric.

Syntax

```
FREQ <Variable>
FREQ * [EXCEPT <Variable>]
```

Syntax Element	Description
<Variable1>	Donor variable (where the values are)
<Variable2>	Receiver Variable (where recoded values will be)

Comments

Records may be included or excluded from the count by using 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 the variable is numeric, confidence limits for the binomial proportions are produced. Further details on the confidence limits are given in the chapter on Statistics.

If a WEIGHTVAR is specified, the value of the WEIGHTVAR variable is treated as a count of instances of the variable being FREQed. For example, in the following command a record containing the value 30 for AGE and 15 for COUNT would give a result equivalent to 15 individuals of age 30:

```
FREQ  AGE WEIGHTVAR = COUNT
```

If STRATUM is specified, a separate frequency will be produced for each value of the stratifying variable.

FREQ ILL STRATAVAR=SEX, for example, will produce a table showing ILL (perhaps Yes/No/Unknown) for Males and another for Females. Note that the same numbers can be obtained using TABLES ILL SEX, but the latter will give the results in one table rather than in separate tables, and produce statistics to test for an association between ILL and SEX.

FREQ * will make a table for each variable in the current view other than unique identifiers. It is often used to begin analysis of a new data set. To do frequencies of all variables except a few, use

FREQ * EXCEPT VarName(s) 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 * EXCEPT NAME (Does frequencies of all variables except NAME)

FREQ AGEGROUP STRATAVAR = COUNTY (Does a frequency of AGEGROUP for each COUNTY)

FREQ AGEGROUP WEIGHTVAR = COUNT (Processes summary data containing a COUNT in each record, so that each record is considered to represent the number of individuals contained in its COUNT field. This replaces the SUMFREQ command in Version 6 of Epi Info.)

Program-Specific Features

Menu	Check	Analysis
	✓	

GOTO

Command

Description

GOTO can be used alone or as a consequence of an IF statement to transfer the cursor to a particular named variable field. The special uses are:

Syntax

GOTO <Event>

Syntax Element	Description
<Event>	Can be a variable, PageEnd, RecordEnd, or NextRec

Comments

+1 – Automatically saves the current page, if changes have been made, and goes to the next page.

-1 – Automatically saves the current page, if changes have been made, and goes to the previous page.

<Page Number> -- Automatically saves the current page, if changes have been made, and goes to the page indicated by the number.

Examples

Program-Specific Features

Menu	Check	Analysis
		✓

GRAPH

Command

Description

Numerous settings are available in the GRAPH module, and these can be saved as a Graph Template. When a Graph Template is referred to by name, the settings are taken from this template. If explicit settings are given as above, they override the settings in the Graph Template.

Syntax

GRAPH <*VarName(s)*> * <*Varname*> **GRAPHTYPE=** "<GraphType>" {**TITLETEXT=**"<*Text*>"} {**WEIGHTVAR=<***Variable***>**} {**TEMPLATE**='<*Filename*>'} {**BITMAP**='<*Filename*>' {**YTITLE**="<*Text*>"} {**PERCENTS=(+)**} {**XTITLE**="<*Text*>"} {**XRANGE**= <low> **TO** <High>} {**YRANGE**= <Low> **TO** <High>} {**XTICK**= <Xnumber>} {**YTICK**=<Ynumber>} {**THREED**= (+)}

GRAPH [VarName(s)] <TemplateFile>

Syntax Element	**Description**	
<GraphType>	HISTOGRAM BAR SCATTER LINE PIE HORIZONTAL BAR SPLINE MARK	AREA PARETO HI-LOW SURFACE POLAR CUBE DOUGHNUT
<Low>	Represents the lower limit to be plotted in the graph for that specific axis	
<High>	Represents the upper limit to be plotted in the graph for that specific axis	

Comments

WEIGHTVAR is intended to be used for summarized data in which one variable contains the frequency for the event.

Since creating graphs might be tedious, all properties for a specific graph can be stored in a **TEMPLATE,** which will be identified with the .CHT extension. If the graph will be used by any other application, it can be saved as a **BITMAP** image.

Proportions can be plotted by selecting **PERCENTS=(+).** Axis titles can be defined using **YTITLE** and **XTITLE**

Axis scale as well as ticks interval can be defined using **XRANGE** and **YRANGE** and **XTICK** and **YTICK** respectively

THREED= (+)

Examples

```
GRAPH BAKEDHAM CABBAGESAL * AGE
GRAPHTYPE="Bar" TITLETEXT="titulo" WEIGHTVAR=SEX
TEMPLATE='temp' BITMAP='bit' YTITLE="Count"
PERCENTS=(+) XTITLE="label2"
```

Program-Specific Features

HEADING

Command

Menu	Check	Analysis
		✓

Description

Set up specific headings as part of the output in ***Analysis***

Syntax

HEADER <Level> "<Text>" (<FontProperty>) **TEXTFONT** <Color> <Size>

Syntax Element	Description
<Level>	0 Body Text 1 Window Title 2 File Title 3 Data Source Title 4 Procedure Title 5 Variable Title 6 Stratum Title
<Text>	Type text to be displayed
<FontProperty>	Properties (Underline, Bold, Italic) will be separated by commas (,)
<Color>	Aqua Black Blue Fuchsia Gray Green Lime Maroon Navy Olive Purple Red Silver Teal White Yellow
<Size>	Any number in the following ranges: From –7 to –1, from 1 to 7, and from +1 to +7

Comments

Values for font size represent standard values for HTM font sizes.

Examples

```
READ 'D:\EPI2000\Sample.Mdb':viewOswego
HEADER 0 "This is HEADER 0" (BOLD) TEXTFONT Green  7
HEADER 1 "This is HEADER 1" (ITALIC) TEXTFONT Yellow  6
HEADER 2 "This is HEADER 2" (UNDERLINE) TEXTFONT Blue  5
HEADER 3 "This is HEADER 3" (BOLD, ITALIC,) TEXTFONT Red  4
HEADER 4 "This is HEADER 4" (BOLD, ITALIC, UNDERLINE)
TEXTFONT Teal  3
HEADER 5 "This is HEADER 5" (BOLD, ITALIC, UNDERLINE)
TEXTFONT Olive  2
HEADER 6 "This is HEADER 6" (BOLD, ITALIC, UNDERLINE)
TEXTFONT White  1
FREQ  ILL
TABLES VANILLA ILL
TABLES VANILLA ILL STRATAVAR= SEX
```

Program-Specific Features

	Menu	Check	Analysis
	✓	✓	✓

HELP

Command

Description

Syntax **HELP** <HTMLFileName{#BlockName}>

Syntax Element	Description
<FileName>	Helpfile

Comments

HTMLFileName must end in .HTM or .HTML. If an optional BlockName is given, the browser searches for named anchors ("bookmarks") in the HTML file and positions the anchor at the top of the screen. An anchor in HTML can be created by using a tag like <a name="Bali Hi">.

If no directory path is given, the HELP command will look first in the HELP directory under the currently designated language directory (e.g., .\SPANISH\HELP. If the file is not found, it looks in the default HELP directory, .\ENGLISH\HELP. That failing, it looks in the currently logged directory. This allows translated help files to be accessed automatically as long as they retain the name (and block names) of the corresponding English versions.

Examples

```
HELP  SouthPacific.HTM#Bali Hi
```

HELP Function.htm#YEAR

Program-Specific Features

	Menu	Check	Analysis
HIDE, UNHIDE Command		✓	

Description

HIDE hides a field from view and makes it NOENTER.

UNHIDE makes a field visible and returns it to the status it had before it was hidden.

Syntax

```
HIDE <Field name(s)>}
UNHIDE  <Field name(s)>
```

Syntax Element	Description
<Field name(s)>	One or more valid variables

Comments

If no field name is specified, the current field (the one to which the Check code block pertains) is assumed.   Plain text (text only) fields can be hidden or unhidden, allowing for the positioning of messages on the screen, and the display of alternate messages.

Examples

Program-Specific Features

	Menu	Check	Analysis
IF THEN ELSE Command		✓	✓

Description

IF defines conditions and one or more consequences which result when 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.

Syntax

```
IF <Expression> THEN
       [Command(s)]
{ELSE
       Command(s)}
END
```

Syntax Element	Description

Comments

If the entire IF statement is on one line and there is no ELSE, then END is not required.

In contrast to Version 6 of Epi Info, Epi Info 2000's IF statement is executed immediately if possible, that is, if it does not refer to a database variable and if any DEFINEd variables have been assigned literal values. Thus, if the statement, YEAR = 97 has already occurred, then an IF statement dependent on it, such as IF YEAR = 97 then …., will be executed immediately

```
IF AGE > 15 THEN GROUP = "ADULT" ELSE GROUP
="CHILD"
DEFINE CASE
DEFINE MAJORSYMP
MAJORSYMP = 0
IF DIARRHEA = (+) THEN MAJORSYMP = MAJORSYMP +
1
IF FEVER = (+) THEN MAJORSYMP = MAJORSYMP + 1
IF VOMITING = (+) THEN MAJORSYMP = MAJORSYMP + 1
IF MAJORSYMP >= 2 THEN
  CASE = (+)
ELSE
  CASE = (-)
END
```

(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 IF statements, to avoid gaps in the logic and results.

Sometimes it is important to have an "ELSE" condition that Program-Specific Features 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 _
ASSIGN ILL = (-)
IF VOMITING = (+) THEN ILL = (+)
IF DIARRHEA = (+) THEN ILL = (+)
IF FEVER = (+) THEN ILL = (+)
```

Sets Ill = (+) only if one or more symptoms are present.

The same result would be achieved by:

```
IF (DIARRHEA = (+)) OR (VOMITING = (+)) OR \
   (FEVER = (+)) THEN ILL = (+) ELSE ILL = (-)
```

(Note that "\" is used to indicate continuation of a long line.)

An “expression” is a combination of variables, literal values, and operators that can be evaluated to produce a single result. Expressions can be quite complex. Both dataset and defined variables can be used in an expression would automatically set DURATION to the number of days between ONSETDATE and the beginning of 1999.   Note that the literal date is enclosed in quotation marks**.** This is essential, since dates are a special form of string variable.

Examples

```
IF (CHEESE = (+)) AND NOT (PEPPERONI = (+)) THEN VEGI = (+)
IF (AGE <30) AND (AGE > 17) THEN GROUP = "Young Adult"
```

Program-Specific Features

KMSURVIVAL Command

Menu	Check	Analysis
		✓

Description

Syntax

KMSURVIVAL <Exposure> **=** <Outcome> * <CensorVar> **(**<Value>**) TITLETEXT=**"<Text>" **TIMEUNIT**="<TimeUnit>" **TEMPLATE=**'<Filename>'

Syntax Element	Description

<Exposure>	<Variable in database>	
<Outcome>		
<CensorVar>		
<Value>	Value for uncensored variables	
<Text>		
<TimeUnit>	Days Hours Months	Weeks Years
<Filename>		

Comments

The KMWin program can also be run from the main menu of Epi Info. In this mode, however, it reads only Epi Info 6 (DOS) files. When run from ***Analysis***, ***Analysis*** generates the necessary Epi Info 6 file automatically and runs the program after transmitting the parameters to set up the model. Once in KMWin, the model can be modified using the menu commands in KMWin.

Examples

Program-Specific Features

LIST Command

Menu	Check	Analysis
		✓

Description

LIST does a line listing of the current dataset. 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

Syntax

- **LIST** [*** EXCEPT**] <VarName(s)> {**COLUMSIZE**= <Num1>, <Num2>, <Num3>} {**NOIMAGE**} {**NOWRAP**} {**LINENUMBERS**}

- **LIST** <variable> [**GRIDTABLE/UPDATE**]

Syntax Element	Description
*	Used to represent all variables
<VarName(s)>	One or more variable names
<Num1>	A number from 1 to 99 representing the number of columns to be displayed
NOIMAGE	Does not display images
NOWRAP	If nowrap is included, column width will depend on the number of characters in the variable
LINENUMBERS	Displays line numbers
GRIDTABLE	Data is displayed as grid instead of HTM format
UPDATE	Allows permanent changes in the database

Comments

Changes made to the database using UPDATE do not use Check code or the properties set in ***MakeView*** (for example, must enter or no enter). If Check code features are desired, use ***Enter*** to edit data if the Update is convenient for changing a series of records in which one variable may have errors, for example.

Adding an EXCEPT Variable list will exclude all the named variables from a LIST or LIST *. This 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 the command SET LISTREC = OFF

If the dataset has been sorted with the SORT command, the records are listed in sorted order. Otherwise, they are listed in their physical order in the dataset.

If /UPDATE is used, the data in the grid can be edited.

The program will ask if the user wants to save edited items.

Examples

```
LIST
LIST name age sex race
LIST *
LIST * EXCEPT NAME SSN ADDRESS CITY STA
```

Program-Specific Features

	Menu	Check	Analysis
LOGISTIC Command			✓

Description

Syntax

LOGISTIC <Outcome>= *<VarName(s)>* {**STRATAVAR** = *<Variable>*} {**WEIGHTVAR =** *<Variable>*} {**TITLETEXT =** <text>} {**TEMPLATE =** <filename>}

Syntax Element	Description
<Outcome>	One variable in the database to be considered outcome. This must be a binary (2-value) variable or a Yes/No variable.
<VarName(s)>	One or more variables to be included in the model
<Variable>	Select variable containing summary data
<text>	Title of the project
<VarName(s)>	

Comments

Individual variable names may also be connected by "+". "*" defines an interaction term. Interaction terms must contain only variables that have previously been included as single terms. If MATCH is specified, conditional logistic regression is done. Any reasonable number of variables and interactions can be specified

Examples

Program-Specific Features

MAP
Command

Menu	Check	Analysis
		✓

Description

Numerous settings are available in the MAP module. These can be saved into a MAP template. When a MAP template is referred to by name, the settings are taken from this record. If explicit settings are given (as above), they override the settings in the MAP Template

Syntax

MAP <Aggregate Function>**(**<TABLE>**)** <*VarName*> :: '<*Shape File*>':<*Variable*> {**TITLETEXT** = "<text>"} {**TEMPLATE** = '<filename>'} {**DENOMINATOR** = <*var2*>} {**WEIGHTVAR** = <*Var3*> }

Syntax Element	Description	
Aggregate Function	**SUM** **COUNT** **MAXIMUM**	**AVERAGE** **MINIMUM**
<*VarName*>	Represents the variable in the dataset that contains information about geographic variables.	
<*Shape File*>	Path and file name for the SHP file to be used.	
<*Variable*>	Variable name in the SHP file that contains information about geographic variable.	
<*var2*>	Variable that contains the denominator for the population	
<*Var3*>	Variable that contains the denominator to calculate rates	

Comments

All rates are given using the denominator 1 to create other rates if necessary to create a new variable that stores the desired rate (1000, 100,000, etc) and use it as WEIGHTVAR

Examples

```
MAP SUM(DENGUE) ADMIN_NAME ::
'D:\EPI2000\Venezuela\venezuel.shp':ADMIN_NAME
```

Program-Specific Features

Menu	Check	Analysis
		✓

MATCH
Command

Description

MATCH does a cross-tabulation of the specified variables. Values of the first variable will appear on the left margin of the table, and those of the second variable will appear across the top of the table. Cells normally contain counts of records matching the values in the corresponding margin labels

Syntax

MATCH <Exposure> <Outcome> **MATCHVAR=** <*Variable1*>
{**WEIGHTVAR=** <*Variable2*>} {**OUTTABLE =** <*Filename*>}
{**COLUMNSIZE =** <Number>} {**NOWRAP**}

Syntax Element	Description
<Exposure>	Variable in the database to be considered the risk factor
<Outcome>	Variable in the database considered Disease of consequence
<Variable1>	Variable used to identify matched pairs
<Variable2>	Variable containing summary data
<Filename>	Name for a table to store summarized data
<Number>	Number of columns per line of output

Comments

The **MATCHVAR** variable is used for stratification. It must be a unique identifier for each group of cases and controls. If every case has a corresponding control, then each pair is given an identifier, perhaps PAIRID, that identifies the pair. The MATCHVAR variable is PAIRID and the MATCH command internally performs a Mantel-Haenszel stratified analysis using PAIRID as the stratifier. Matched groups can have several controls per case, or even more than one case matched to several controls.

WEIGHTVAR is intended for summary data in which the number of cases will be stored in a variable such as count.

Examples

In a review of hospital records of patients with and without postoperative infection, a single case is matched by age, sex, and operation type to as many controls (1 to 4) as are available. Each case record and the matched control record(s) is given a GROUPID number.

The INFECTION variable indicates the presence or absence of postoperative infection. The analysis of participation of Dr. A. in the operation is done as follows:

```
MATCH INFECTION DRA GROUPID
```

Program-Specific Features

Menu	Check	Analysis
		✓

MEANS Command

Description

Syntax

MEANS <*Variable1*> <*Variable2*> {**STRATAVAR=** <variable3>} {**WEIGHTVAR=** <variable4>} {**OUTTABLE =** <Filename>}

Syntax Element	Description
<Variable1>	A numeric variable to be used to calculate means
<Variable2>	Any variable used for cross-tabulation
<Variable3>	Variable to be used for stratified analysis
<Variable4>	Variable containing summary data
<Filename>	Name for a table to be created

Comments

The MEANS command requires two pieces of information: the name of a numeric variable containing data to be analyzed and the name of the variable that indicates how groups will be distinguished. MEANS normally produces a table that displays the data and then performs appropriate statistics.

Depending on the size of the table and whether the data are normally distributed, MEANS provides:

<u>Parametric tests</u>

ANOVA (for two or more samples)

Student's t-test (for two samples)

<u>Non-parametric tests</u>

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 the chapter on Statistics.

Examples

MEANS AGE ILL

will compare ages for those in whom ILL =(+) with those for whom ILL = (-). 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 (+), (-), 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).

Program-Specific Features

MENU

Command

Menu	Check	Analysis
✓		

Description

Syntax

```
MENU <FileName>.MNU {<Image>.BMP }{<Left>, <Top>},
{<Width>, <Height>}
```

Syntax Element	Description
<FileName>	
<Image>	Image to be used as background for the menu. Epi2000.exe supports GIF, JPEG, and BITMAP images.
<Left>	A number from 1 to 99 representing the distance from the left side of the window in percentage points.
<Top>	A number from 1 to 99 representing the distance from the top border of the window in percentage points.
<Width>	A number from 1 to 99 representing the width of

	the image desired.
<Height>	A number from 1 to 99 representing the distance from the top border of the window in percentage points.

Comments

The MENU command loads another menu with the given file name from within a block of commands in an MNU file. If the name of an image file is given (either .BMP or .BID), the image is loaded on the main form. Left and Top locations can be given to position the upper left corner of the menu on the screen. They are given as percents of screen width and height: 0, 0 is the upper left corner of the screen. Width and Height are also percentages of the screen. If no locations and sizes are given, those of the main menu are used.

Examples

```
MENU SURVEIL.MNU  REFCAMP.BMP  5,5, 50, 50
```

It displays the surveillance menu, an image of a refugee camp from the air, and places the menu 5% right and 5% down from the upper left corner of the screen, extending half the screen width and half the screen height so that it occupies one quarter of the screen.

Program-Specific Features

MENUITEM Command

Menu	Check	Analysis
✓		

Description

MENUITEM is used to create a non- hierarchical option in the vertical menu

Syntax

MENUITEM “<Text>”, <Block>

Syntax Element	Description
<Text>	Text to be displayed
<Block>	Block to be executed (see Command Block)

Comments

Every Menu in Epi Info 2000 contains two main components. The first part defines the vertical menu tree and the second part defines the command blocks.

Examples

```
BEGIN
  POPUP "&Programs "
  BEGIN
       MENUITEM "Ma&ke View (Questionnaire)" , MakeView
       MENUITEM "&Enter Data", Enter
       MENUITEM "&Analyze Data", Analysis
       MENUITEM "&StatCalc", StatCalc
       MENUITEM "Epi Ma&p", EpiMap
       MENUITEM "&Nutrition", Nutstat
       MENUITEM "&Visualize Data", VisualizeData
       MENUITEM "&Word Processing", Eped
       MENUITEM "E&xit ", Exit
    END
END
```

Program-Specific Features

	Menu	Check	Analysis
			✓

MERGE
Command

Description

Merges records in one dataset with those in another, using one or more defined identifiers to establish the match between records. Records in the second dataset can be appended to the end of a dataset, used to update records in the main dataset, or the two datasets can be joined “side-by-side.”

Syntax

MERGE “<Database format>” ‘<Filename>:’<View>’ <**Type**> **{LINKNAME**= <Text>} <Key(s)>

FILESPEC

Syntax Element	Description
<Database>	The type of database or file to be read as the Merge file.
‘<Filename>:’<View>’ <Type>	The name of the Merge file or table, that is, the source of records that will be merged into the currently active dataset.
<Type>	If no type is specified, the effect is the same as both APPEND and UPDATE below: matching records are updated with new material that may appear in the merging file, and those that do not match are added to the end of the main file. APPEND adds records in the Merge file or table to the end of the currently active dataset. Only fields found in both datasets will be added. UPDATE will replace records in the active table with those in the Merge file or table if the UniqueKeys match. RELATE moves the unique key of the current table to the foreign key of the related table to make a permanent relationship. The related (Merge) table must be an Access/Epi2000 table. If it started life as an Epi Info 6 file, it must be imported using the FILE|IMPORT feature, which

will insert the necessary fields (FKEY and others) in the resulting Access/Epi2000 table.

Key(s)

One or more expressions that designate keys on which the match or relate will be performed. These are in the form:

<ExpressionCurrentTable>::<ExpressionMergeTable>

In most cases the expressions are the names of variables in the tables that will serve as keys for the match. They can also be mathematical expressions including variable names that would produce a unique match on particular records. More than one key pair ("multiple keys") can be designated, separated by commas.

FILESPEC

A special set of instructions for accessing files other than Access files, often those with password security set. See READ for more details.

Comments

See READ for more information about database formats and FileSpecs.

Examples

Program-Specific Features

MUSTENTER

Property

Menu	Check	Analysis
	✓	

Description

MUSTENTER specifies that missing values are not allowed. 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.

Comments

Examples

Program-Specific Features

PICTURE Command

Menu	Check	Analysis
✓		

Description

Loads the specified image, but does not store the image name between sessions.

Syntax

```
PICTURE <Image>
```

Syntax Element	Description
<Image>	Image to de displayed. Epi Info 2000 supports BMP, GIF, and JPEG images.

Comments

Examples

Program-Specific Features

POPUP Command

Menu	Check	Analysis
✓		

Description

POPUP defines a main menu that will appear on the top line of the window. Individual MENUITEMS define the pull-down menu within the POPUP item.

Syntax

```
POPUP  "<ItemText>"
Begin
            MENUITEM(s)
End
```

Syntax Element	Description
BEGIN	Delimiter used to identify the beginning of the popup menu
END	Represents the end of the popup menu

Comments

Omitting the Begin or End statement will cause unpredictable results in the menu structure. (Yes, it might have been better to call this a POPDOWN command, since the menus do come down rather than up.)

Examples

```
POPUP "&Programs "
          BEGIN
             MENUITEM "Ma&ke View (Questionnaire)" , MakeView
             MENUITEM "&Enter Data", Enter
             MENUITEM "&Analyze Data", Analysis
             MENUITEM "&StatCalc", StatCalc
             MENUITEM "Epi Ma&p", EpiMap
             MENUITEM "&Nutrition", Nutstat
             MENUITEM "&Visualize Data", VisualizeData
             MENUITEM "&Word Processing", Eped
             MENUITEM "E&xit ", Exit
           END
```

Program-Specific Features

RANGE

Property

Menu	Check	Analysis
	✓	

Description

Values falling outside a specified 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).

Comments

Examples

Program-Specific Features

Menu	Check	Analysis
		✓

READ

Command

Description

READ makes one or more views the active dataset. It also removes any previously active datasets and associated DEFINEd variables, and dataset-specific commands.

Syntax

READ "<Database format>" '<*Filename*>': <*View or data table*>
LINKNAME= <LinkTable Name>
FILESPEC HDR="NO" **FMT**="<File format>"

Syntax Element	Description
"<Database format>"	One of the database Specifiers listed in the Comments section below
'<*Filename*>':	The name of the an Access/Epi2000 database (MDB) or of an external file if this is indicated in the Database Format
<*View*>	Name of the View or data table if the table is in Microsoft Access/Epi2000 format. If a View table is designated, ***Analysis*** reads the names of the data table and of related Views from the View specified. If the file is a Microsoft Excel file, a range, preceded by an exclamation point (!), can be can be specified in place of the View.
LinkTable Name	The name of a link table in the current (home) MDB that constitutes a link to an external file or data table
FILESPEC	A string describing conditions for access to an external file or table. Must be placed on a separate line following the line of the READ command. The argument is expressed in two parts: the database type, followed by a semicolon (;) and the optional arguments. You must first provide the database type, such as "ODBC;" or "Excel 5.0;". The optional arguments follow in no particular order, separated by semicolons. One of the parameters may be the password (if one is assigned). For example: "Excel 5.0; pwd=mypassword"

HDR	No Yes
<File format>	

Comments

If the file to be READ is not in Microsoft Access format, a Microsoft Access/Epi2000 database (MDB) must be designated in which LINKs to the external files will be placed. A link is a small Access table giving information about the external file. After a link is created, it can be treated like an internal Access/Epi2000 table, although the data remain in the original file.

READ erases or “forgets” all previous IF, SELECT, ASSIGN, SORT, and RECODE statements. All DEFINEd variables except for those designated GLOBAL or PERMANENT 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 and PERMANENT variables, which will retain their definitions and values).

The path as shown in the table below is the full path for the directory containing the database tables and must be preceded by the identifier "DATABASE=". In some cases (as with Jet and Microsoft Excel databases) a specific filename is included in the database path argument. The others do not have more than one dataset (table) within a file.

The following table shows possible database types and their corresponding database specifiers and paths.

Database type	Specifier	Path
Jet database	";"	"drive:\path\filename.MDB"
dBASE III	"dBASE III;"	"drive:\path"
dBASE IV	"dBASE IV;"	"drive:\path
Paradox 3.x	"Paradox 3.x;"	"drive:\path"
Paradox 4.x	"Paradox 4.x;"	"drive:\path"
FoxPro 2.0	"FoxPro 2.0;"	"drive:\path"
FoxPro 2.5	"FoxPro 2.5;"	"drive:\path"
FoxPro 2.6	"FoxPro 2.6;"	"drive:\path"
Excel 3.0	"Excel 3.0;"	"drive:\path\filename. XLS"
Excel 4.0	"Excel 4.0;"	"drive:\path\filename.XLS"
Excel 5.0.	"Excel 5.0;"	"drive:\path\filename.XLS"

Text (Fixed) Text (Delimited)	"Text;"	"drive:\path"

Other file formats can be accessed through ODBC drivers provided by third parties.

Examples

Program-Specific Features

Menu	Check	Analysis
	✓	

READ ONLY

Property

Description

Allows defining variables that contain calculated values.

Comments

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.

Examples

Program-Specific Features

Menu	Check	Analysis
	✓	

REPEAT

Property

Description

Causes the value of the same field in the last record accessed to become the default value for the designated field in the current record.

Syntax

Syntax Element	Description

Comments

Saves keystrokes during data entry for fields where the value seldom changes.

Examples

A county or state name is often the same for record after record in a given locality. Making it a REPEAT field brings up the last value entered with each new record. In the few records in which the correct answer may be another state or county, the data entry person can easily change the value before the record is saved.

Program-Specific Features

Menu	Check	Analysis
		✓

RECODE

Command

Description

Syntax

```
RECODE <Variable1> TO <Variable2>
        Value1 – Value2 = "<Recoded Value>"
        Value2 – Value3 =  "<Recoded Value>"
END
```

Syntax Element	Description
<Variable1>	Donor variable (where the values are)
<Variable2>	Receiver Variable (where recoded values will be)

Comments

Text must be enclosed in quotation marks. Numeric ranges are separated by a space, hyphen, and space, as in 1 - 5. Multiple entries are permitted, as in 1,3,5, 8 -10 = "Allowed values". Negative values are permitted, as in -10, -9 – -8.

Examples

```
RECODE AGE TO var1
        LOVALUE - 0 = "<=0"
        0 - 10 = ">0 - 10"
        10 - 20 = ">10 - 20"
        20 - 30 = ">20 - 30"
        30 - 40 = ">30 - 40"
        40 - 50 = ">40 - 50"
        50 - 60 = ">50 - 60"
        60 - 70 = ">60 - 70"
        70 - 80 = ">70 - 80"
        80 - 90 = ">80 - 90"
        90 - 99 = ">90 - 99"
        99 - HIVALUE = ">99"
END
```

Program-Specific Features

Analysis

Analysis cannot RECODE more than 12 levels of values. To bypass this limitation, there is an example in the how to section about large recoding.

REGRESS Command

Menu	Check	Analysis
		✓

Description

This command performs multiple linear regression with up to XXX variables.

Syntax

```
REGRESS  <Outcome> = <Predictor(s)>
```

Syntax Element	Description
<Outcome>	Variable containing result or effect
<Predictor(s)>	One or more numeric variables that can be used as predictor for a linear regression model.
<Weight>	Variable containing summary data

Comments

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).

Examples

Program-Specific Features

	Menu	Check	Analysis
RELATE Command		✓	✓

Description

The RELATE command links one or more tables to the current dataset during analysis, using a common identifier to find matching records. The identifier may span several fields, in which case values in each of the fields must match. The changes are not permanent, and the linked tables remain separate on the disk, having been linked only temporarily by a system of pointers in random access memory.

Syntax Element	Description

Comments

To use RELATE, at least one table must have been made active with the READ command. The table to be linked must have a key field such as HOUSEID that has identifiers relating records in the two tables. In Epi Info 2000, the keys in the main and related tables or files do not have to have the same name. Epi Info 2000 can also create relationships with more than one variable at the same time; the number of variables on both sides of the :: must be the same, however.

If the keys were created automatically by Epi Info 2000, they will both be called "UniqueID" and you do not have to specify keys. If you try to relate two tables with no automatic relationship established, ***Analysis*** will notify you that the relationship could not be completed.

The related table may have fewer records than the main table, but at least some of the records should have the same identifier in the identifier field. Identifiers in the related table, but not necessarily in the main table, must be unique--that is, without duplication.

After issuing the RELATE command, the variables in the related table may be used as though they were part of the main table.

Frequencies, cross-tabulations, and other operations involving data in both the main and related tables can be performed. If a permanently linked table is desired, the WRITE command may be used to create a new, permanently linked table.

More than one table may be RELATEd to the main table by using successive RELATE commands.

Sometimes a variable name may be duplicated in more than one table. ***Analysis*** keeps the variables in related tables separate, and they can be displayed with the VARIABLES command. In referring to a variable in a related table, you may (optionally) use the form: HOUSE.AGE to represent the variable AGE in the View HOUSE. This will distinguish it from another variable AGE that might be in the main table.

By using the relational features of ***Enter*** and Check code 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***.

Examples

Program-Specific Features

Check Code

Relationships are created in Epi Info 2000 as a new type of variable. A relationship created in Epi6 will not be converted automatically into one in Epi Info 2000. Use MERGE to convert an Epi6 relationship into one in Epi Info 2000.

Analysis

If the relationship was created in ***MakeView***, ***Analysis*** will identify variables without definition.

Grid tables and relate buttons are examples of automatic relationships. ***Analysis*** does not load related tables automatically; therefore, the user must establish the relationship from a program or from the Relate form.

REPEAT

Property

Menu	Check	Analysis
	✓	

Description

When REPEAT is enabled, the variable will be initialized with the value taken from the previous record.

Syntax

Syntax Element	Description

Comments

This feature can save effort when entering items like STATE that remain constant with only a few exceptions over many records. Only the exceptions need be entered in detail.

Examples

Program-Specific Features

ROUTEOUT

Command

Menu	Check	Analysis
✓	✓	✓

Description

Directs output to the named file until the process is terminated by CLOSEOUT. Output from commands such as FREQ and LIST is appended to the same output file as it is produced.

Syntax

```
ROUTEOUT  <filename>
```

Syntax Element	Description
<Filename>	An HTM document where the output is to be stored.

Comments

Epi Info 2000 will always send the output to an HTM document that can be read by any browser.

If no output is selected, Epi Info 2000 will create a new file called OUTXXX.HTM where XXX is a sequential number. These files will be placed in the working directory (usually Epi2000). To define new working directories see SETWORKDIR.

Examples

See Also

The storage facilities for creating a library of output files are described at the end of this chapter.

	Menu	Check	Analysis
RUNPGM Command			✓

Description

Control is transferred to the second program, returning to the first automatically, beginning with the line following the RUN statement. Thus a program being RUN from another program is somewhat like an INCLUDE file or subroutine in other systems.

Syntax

RUN '<FileName>' **:** "<Program>"

Syntax Element	Description
<FileName>	Path and Name for the MDB file or DOS .PGM file where the program is stored.
<Program>	Program Name within the MDB

Comments

Since the filename can include any path and database name, the program to be executed can be stored in a different database than the one in use.

Examples

```
RUNPGM 'C:\EPI2000\SAMPLE.Mdb':"Test1"
```

Program-Specific Features

	Menu	Check	Analysis
SCREENTEXT Command	✓	✓	✓

SCREENTEXT
Command

Description

This places the specified text on the screen at the location specified, with the size, color, font, and style designated. Horizontal and vertical positions are given in percentage of screen width and height. Thus 50, 50 would be centered on the screen. If the item is in the middle 50% of screen width, it is centered. If Horizontal is less than 25, the text is left justified. When Horizontal is more than 75, the text is right justified. Sizes vary with different fonts, but can be determined through examining the font possibilities in ***Eped***. Text Fonts must be enclosed in quotation marks and must be spelled exactly as they appear in a Windows font selection window, as in ***Eped***.

Syntax

```
SCREENTEXT "<ItemText>", <Horizontal>, <Vertical>, <Size>,
<Color>, <Font> <Style>
```

Syntax Element	Description
<Horizontal>	A number from 1 to 99 representing the distance from the left border of the window in percentage points.
<Vertical>	A number from 1 to 99 representing the distance from the top border of the window in percentage points.
<Size>	A number between 1 and 999 representing the font size
<Color>	Black Red Green Yellow
<Font>	Any of the fonts registered in the operating system of the computer. If the font is not available, text will be displayed using the default Windows font.
<Style>	Bold Underlying Italic Blue Magenta Cyan White

Comments

Examples

ScreenText "Configurable Menu. See EPI2000.MNU.", 3, 0, 8, yellow, Arial, Bold

Program-Specific Features

Menu

Screen text is a separated part of the MNU program. It should be closed with the command END. If END is missing, results can be unpredictable.

Menu	Check	Analysis
		✓

SELECT Command

Description

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.

Syntax

```
SELECT <expression>
```

Syntax Element	Description
<expression>	Any valid Epi Info 2000 expression.

Comments

SELECT expressions are additive so that the two expressions:

```
SELECT AGE > 35
SELECT SEX = "F"
```

are equivalent to

```
SELECT (AGE>35) AND (SEX = "F")
```

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].

Program-Specific Features

Analysis

This takes the place of SET PROCESS = DELETED/UNDELETED/BOTH in Epi Info for DOS.

Normally, however, deleted records are not processed unless a special SELECT statement has included them.

Menu	Check	Analysis
		✓

SET
Command

Description

Includes in ***Analysis*** programs the settings that are made from the SET dialog in ***Analysis***, so that these settings will be used whenever the program runs. SET provides various options that affect the performance of ***Analysis*** or its output.

Syntax

Syntax Element	Description

Comments

Parameter	Values	Response
(-)	"<Text>"	In Boolean variables, NO will be represented as <Text>
(.)	"<Text>"	In Boolean variables, MISSING will be represented as <Text>
(+)	"<Text>"	In Boolean variables, NO will be represented as <Text>
DELETED	YES NO	Deleted records are included

	ONLY	Deleted records are not included Only deleted records are included
FREQGRAPH	(+) (-)	Show frequency graph next to the table Do not include freqgraph.
IGNORE	(+) (-)	Do not include missing values for analysis Include missing values for analysis
HYPERLINKS	(+) (-)	Hyperlinks are included in the output Hyperlink excluded
MISSING	(+) (-)	Include missing values for analysis Do not include missing values for analysis
PROCESS	DELETED UNDELETED BOTH	Include only deleted records Include only undeleted records Include all records
SELECT	(+) (-)	Display selection criteria Do not display selection criteria
SHOWPROMPTS	YES NO	Displays the full prompt for variable name Displays Variable name only
STATISTICS	NONE MINIMAL INTERMEDIATE COMPLETE	No statistics are displayed
YN	"<Text1>", "<Text2>", "<Text3>"	<Text1> represents Boolean variables containing value = (+) <Text2> represents Boolean variables containing value = (-) <Text3> represents Boolean variables containing value = missing (.)

Examples

Program-Specific Features

SETBUTTONS Command

Menu	Check	Analysis
✓		

Description

Turns the buttons on the menu on or off.

Syntax

SETBUTTONS

Syntax Element	Description

Comments

Use to set up a user option for making buttons visible or invisible. If the buttons are currently visible, they are made invisible and inactive, and vice versa.

Examples

Program-Specific Features

SETPICTURE Command

Menu	Check	Analysis
✓	✓	✓

Description

Allows the user to choose an image and set it as the image for the current menu.

Syntax

```
SETPICTURE <Image>.BMP
```

Syntax Element	Description

Comments

The name and location of the chosen image is stored in the EPIINFO.INI file so that the menu can display the same image next time it runs. SETPICTURE does not set up a picture; it merely allows the user to do so.

Examples

Program-Specific Features

Menu	Check	Analysis
		✓

SORT
Command

Description

SORT allows you to specify the sequence in which records will be written with the LIST and WRITE commands. If no variable names are specified after the SORT command, the current sort is cleared, and subsequent output of records will be in their order in the original data table. 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.

If ORDER=DESCENDING is specified, the sort is in reverse order, with Z coming before A.

Syntax

```
SORT  <VarName(s)>  {DESCENDING}
```

Syntax Element	Description
<Varname(s)>	One or more variables in memory
DESCENDING	Indicates that the sort order is descending.

Comments

The parameter DESCENDING must be placed next to the variable to be sorted in descending order. Should several variables be sorted in descending order, one “DESCENDING” should be included for each of them.

Examples

```
SORT AGE
SORT AGE DESCENDING ILL DESCENDING SEX DESCENDING
```

Program-Specific Features

Menu	Check	Analysis
	✓	

SOUNDEX

Property

Description

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, so that subsequent searches will find both names.

Comments

Examples

Program-Specific Features

SYSINFO Command

Menu	Check	Analysis
✓		

Description

This command displays a remarkable amount of information about the Windows 95, 98, or NT operating system and the hardware and software of the computer. It includes type of processor, memory usage, disk space, programs currently active, date and version of all system modules, and many other items useful for solving problems.

Syntax

SYSINFO

Comments

Examples

Program-Specific Features

TABLES Command

Menu	Check	Analysis
		✓

Description

TABLES does a cross-tabulation of the specified variables and sends the table to the screen, printer, or other current output. 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 /WEIGHT parameter is given, the cells represent sums exactly as in SUMTABLES. TABLES DISEASE COUNTY /WEIGHT=COUNT gives the same results as SUMTABLES COUNT DISEASE COUNTY in Epi Info for DOS.

Syntax

TABLES <Exposure> <Outcome> **STRATAVAR=** <*Variable*> {**WEIGHTVAR=** <*Variable*>} {**OUTTABLE =** <*Filename*>} {**COLUMNSIZE =** <Number>} {**NOWRAP**}

Syntax Element	Description
<Exposure>	Variable in the database to be considered the risk factor
<Outcome>	Variable in the database considered Disease of consequence.
<Variable>	Variable active in the database
<Filename>	A valid name to be used to store output
<Number>	A number from1 to 99 representing max number of columns.

Comments

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 dataset 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 (+) and (-) 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 3 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
```

Program-Specific Features

Analysis

Columsize will be used only when the output is too large to fit in regular browser output. The browser allows up to XXX characters wide.

Menu	Check	Analysis
		✓

TYPE

Command

Description

The TYPE command has two meanings, determined by whether the information that follows is in quotation marks. If text is in quotation marks, TYPE will send the text to the screen, printer, or other output file designated in the last ROUTEOUT command. Within the quoted text, the following special commands may be used:

Syntax

```
TYPE  "Text" ALIGN=  SIZE=  COLOR = FONT=  STYLE=
or
TYPE  FileName     (in single quotes if it contains spaces)
```

Syntax Element	Description

Comments

\N	Begins a new line, useful for creating mailing labels, for example.
\number	The character with the ASCII decimal code given by the number.
@@VarName	Inserts the current value of the variable into the text.

The only real difference between TITLE and TYPE so far is that TYPE places the text once when the command is encountered, and TITLE sets ***Analysis*** to place the title at the top of each segment of output until it is cleared by the TITLE CLEAR or TITLE CLEAR ALL command.

If no text in quotation marks follows the TYPE command, TYPE sends the contents of the file specified to the current output. Thus it is possible to create “boilerplate” output that can be included in other output. This output can be in HTML, including complicated logos, and even references to ActiveX controls, although you must be sure that graphics files are available and ActiveX controls have been properly registered.

Examples

```
TYPE  'My Fancy Logo'
```

Copies the contents of the file to output.

```
TYPE "Weekly Report for Week @@WEEK", Align=Center
```

Centers the text on the page with the current value of WEEK substituted for @@WEEK.

Program-Specific Features

	Menu	Check	Analysis
UNDEFINE Command	✓	✓	✓

Description

UNDEFINE removes a defined variable and any assigned values from the system.

Syntax

UNDEFINE <Variable>

Syntax Element	Description
<Variable>	A Defined Variable

Examples

Program-Specific Features

Check code

Undefine can be used in any variable at any time.

Analysis

Permanent variables cannot be undefined from the GEN screen. To Undefine a permanent variable, type the instruction and run it from the text editor.

Menu

Undefine can be achieved by assigning and empty string “” to the variable.

	Menu	Check	Analysis
WRITE Command			✓

Description

WRITE will send records to the current output table or file in the format that you specify. You can specify what variables will be written, the order in which they will appear, and the type of file to be written. Allowable output types are:

Syntax

WRITE <METHOD> {<Output Type>} {<Project>:}Table {<VarName(s)>}

WRITE <METHOD> {<Output Type>} {<Project>:}Table * **EXCEPT** {<VarName(s)> }

Syntax Element	**Description**
<METHOD>	REPLACE APPEND
<Output Type>	See table below
<Project>	Path and filename

Output types available in Epi2000

Database type	Specifier	Path
Jet database	";"	"drive:\path\filename.MDB"
dBASE III	"dBASE III;"	"drive:\path"
dBASE IV	"dBASE IV;"	"drive:\path
Paradox 3.x	"Paradox 3.x;"	"drive:\path"
Paradox 4.x	"Paradox 4.x;"	"drive:\path"
FoxPro 2.0	"FoxPro 2.0;"	"drive:\path"
FoxPro 2.5	"FoxPro 2.5;"	"drive:\path"
FoxPro 2.6	"FoxPro 2.6;"	"drive:\path"
Excel 3.0	"Excel 3.0;"	"drive:\path\filename. XLS"
HTML Table	"HTML"	"drive:\path\filename.HTM"
Excel 4.0	"Excel 4.0;"	"drive:\path\filename.XLS"

Excel 5.0.	"Excel 5.0;"	"drive:\path\filename.XLS"
Text (Fixed)	"Text;"	"drive:\path"
Text (Delimited)		

Comments

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. Global and permanent variables will not be written unless explicitly specified.

If you wish to write only selected variables, they may be listed. The word EXCEPT may be inserted to indicate all variables except those following EXCEPT.

Either APPEND or ERASE must be specified to indicate that an existing file/table by the same name will be erased or records will be appended to the existing file/table if present. If some but not all of the fields being written match those in an existing file during an APPEND, only those for which the file has fields will be written. The WRITE command will not add new variables to an existing dataset during an APPEND, but will, of course, create all necessary database variables if ERASE is specified.

Examples

Program-Specific Features

Notes on the Output Library in Analysis

By default, output files are stored in the directory of the current project with a name composed of a prefix and a sequence number. The prefix and sequence number can be changed in the storage screen. The output of a single statistical procedure is called a result. Results accumulate in a given output file until the data source changes by READ, RELATE, SORT, SELECT or SET PROCESS, or until a CLOSEOUT command is executed.

Users can specify the location of the output by using the ROUTEOUT command. If no directory is given, the file is placed in the directory of the current project. Results accumulate until a CLOSEOUT command is executed.

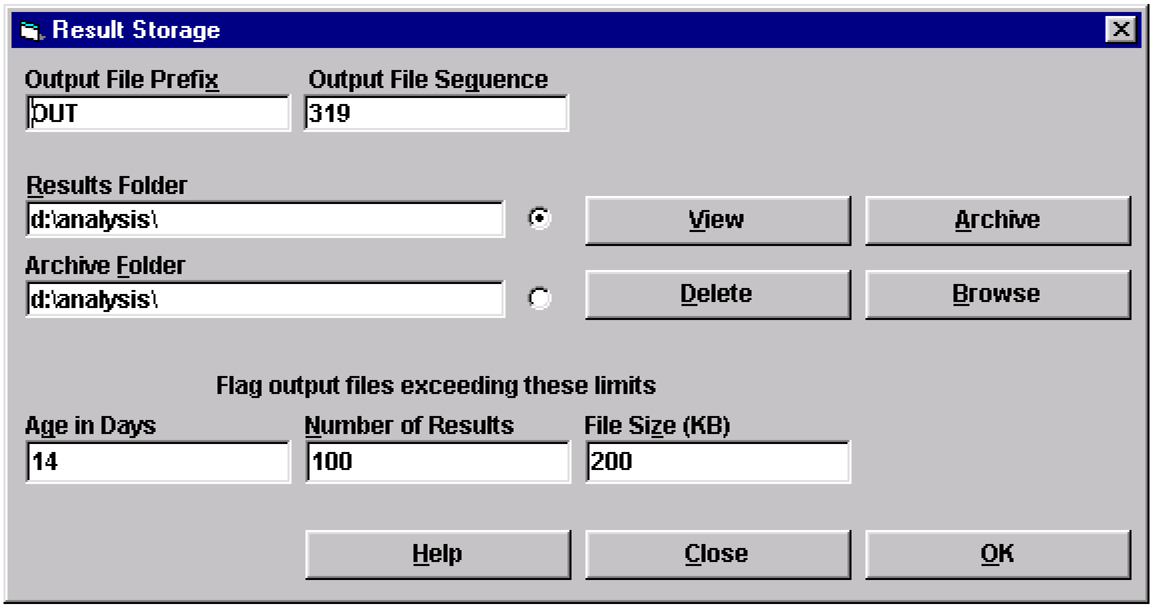
Each result is logged to a results index file named IResults.htm, located in the project directory. Thus, if several projects share the same directory, they share the same results index. Output files generated with HYPERLINKS set on will contain a link to the results index.

Results can be selected for archiving using the storage screen. Results so selected are logged to an archive index file named IArchive.htm. The archive index may be located in any directory, and may include references to results taken from more than one results index. Archived results are marked as archived in the results index. A result may be archived in more than one archive index, but this is not recommended.

Results can be selected for deletion using the storage screen. Because an output file can contain more than one result, it is not actually deleted until all its results have been deleted from the index. Archived results cannot be deleted when viewing the results index, and therefore are protected from deletion within the ***Analysis*** storage system. When viewing an archive index, archived results can be deleted from within the ***Analysis*** storage system, subject to the following limitations: 1) If a result is archived to more than one archive index, it can only be deleted from the first archive to which it was assigned, and deletion from the first archive will be effective even if the result has been archived elsewhere. 2) If a result is stored in an output file located in a different directory than its results index, it cannot be deleted from its results index.

Neither output files nor index files are protected from deletion using the operating system.

Storage is an interactive command only; no code is generated and its functions cannot be programmed. The storage screen looks like this:



Result Storage

Output File Prefix: OUT

Output File Sequence: 319

Results Folder: d:\analysis\ ●

Archive Folder: d:\analysis\ ○

Flag output files exceeding these limits

Age in Days: 14

Number of Results: 100

File Size (KB): 200

The output file prefix and sequence are set by the top two text boxes. The next sequence assigned will be one greater than that shown. Existing files of the same name are not overwritten; the sequence is incremented until an unused name is found.

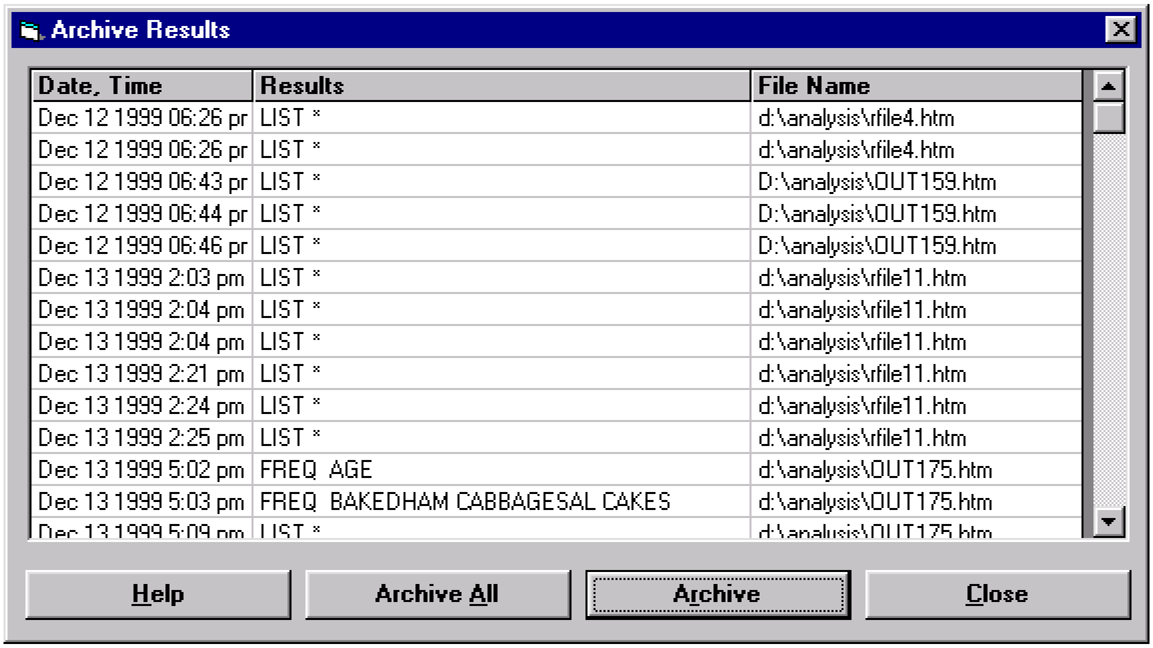
The limits on age, number of results, and total file size are set by the bottom three text boxes. These are used in the Flag Files operation.

Selecting OK will close the window and save the current values of the five text boxes just discussed in the INI file. Selecting Close will close the window without saving the values.

The View, Delete, and Browse operations can be done on either a results index or an archive index. Since the names of these files are fixed, the user need only choose the folder in which they reside. The user chooses the file to be used by selecting the corresponding option button. The Archive operation can be done only on a results file; its button is disabled when the archive folder is selected.

Selecting Browse will open a common dialog file open window. The user must choose a file in the appropriate directory. Although new folders can be created, they cannot be selected because they will have no files in them; however, the user can then type in the folder name.

Selecting one of the other operations will open a new window, which looks the same for each command except that the operations available on it are different. The window for Archive looks like this:

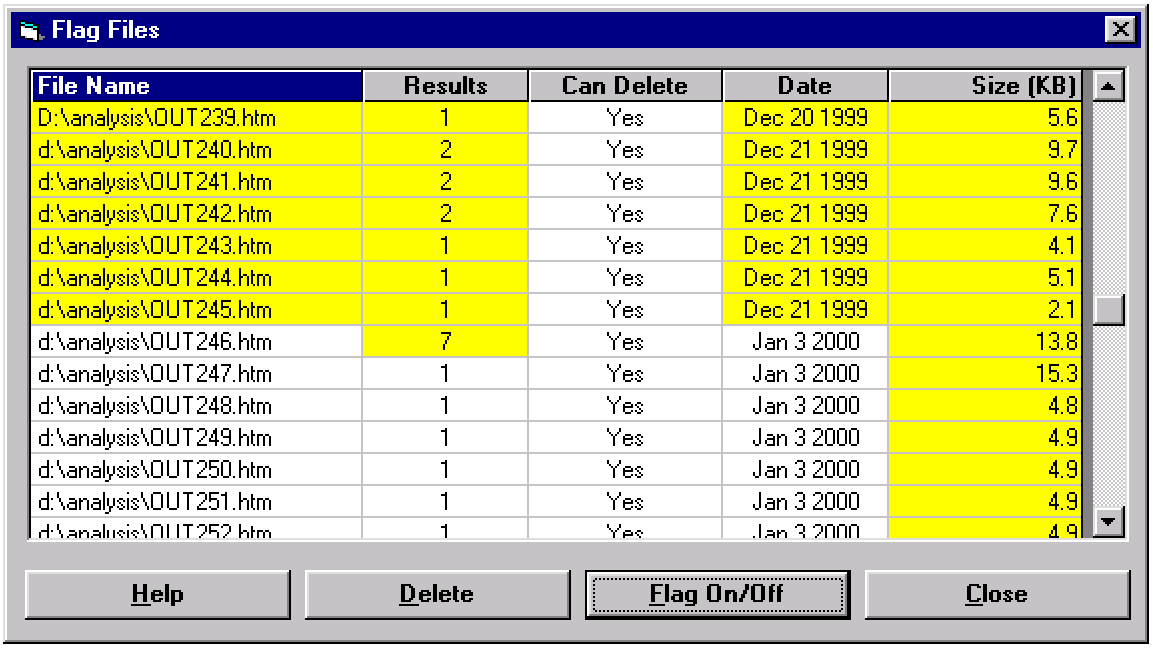


Date, Time	Results	File Name
Dec 12 1999 06:26 pr	LIST *	d:\analysis\rfile4.htm
Dec 12 1999 06:26 pr	LIST *	d:\analysis\rfile4.htm
Dec 12 1999 06:43 pr	LIST *	D:\analysis\OUT159.htm
Dec 12 1999 06:44 pr	LIST *	D:\analysis\OUT159.htm
Dec 12 1999 06:46 pr	LIST *	D:\analysis\OUT159.htm
Dec 13 1999 2:03 pm	LIST *	d:\analysis\rfile11.htm
Dec 13 1999 2:04 pm	LIST *	d:\analysis\rfile11.htm
Dec 13 1999 2:04 pm	LIST *	d:\analysis\rfile11.htm
Dec 13 1999 2:21 pm	LIST *	d:\analysis\rfile11.htm
Dec 13 1999 2:24 pm	LIST *	d:\analysis\rfile11.htm
Dec 13 1999 2:25 pm	LIST *	d:\analysis\rfile11.htm
Dec 13 1999 5:02 pm	FREQ AGE	d:\analysis\OUT175.htm
Dec 13 1999 5:03 pm	FREQ BAKEDHAM CABBAGESAL CAKES	d:\analysis\OUT175.htm
Dec 13 1999 5:09 pm	LIST *	d:\analysis\OUT175.htm

Selecting Archive will cause the selected results to be archived. Selecting Archive All will cause all the results to be archived.

If the Delete operation was selected, the available operations are Delete and Delete All. Selecting Delete will cause the selected results to be deleted. Selecting Delete All will cause all the results to be deleted. Prior to deletion, a confirmation dialog will appear. Deletion will only affect the index file, unless all references to a particular output file are deleted. In that case, the confirmation dialog will indicate which files would be deleted.

If the View operation was selected, the available operations are Display and Flag Files. Selecting Display will cause the selected result to be displayed in the browser. Selecting Flag Files will cause a new window to be opened, which looks like this:



File Name	Results	Can Delete	Date	Size (KB)
D:\analysis\OUT239.htm	1	Yes	Dec 20 1999	5.6
d:\analysis\OUT240.htm	2	Yes	Dec 21 1999	9.7
d:\analysis\OUT241.htm	2	Yes	Dec 21 1999	9.6
d:\analysis\OUT242.htm	2	Yes	Dec 21 1999	7.6
d:\analysis\OUT243.htm	1	Yes	Dec 21 1999	4.1
d:\analysis\OUT244.htm	1	Yes	Dec 21 1999	5.1
d:\analysis\OUT245.htm	1	Yes	Dec 21 1999	2.1
d:\analysis\OUT246.htm	7	Yes	Jan 3 2000	13.8
d:\analysis\OUT247.htm	1	Yes	Jan 3 2000	15.3
d:\analysis\OUT248.htm	1	Yes	Jan 3 2000	4.8
d:\analysis\OUT249.htm	1	Yes	Jan 3 2000	4.9
d:\analysis\OUT250.htm	1	Yes	Jan 3 2000	4.9
d:\analysis\OUT251.htm	1	Yes	Jan 3 2000	4.9
d:\analysis\OUT252.htm	1	Yes	Jan 3 2000	4.9

On this screen, for each file, the following information is displayed: its name; the number of results it contains; whether it can be deleted (true if results in it are not archived elsewhere); file date; and file size in KB. The file date, number of results, and file size columns are highlighted if they exceed the limits set on the main storage screen. The number of results and file size limits are cumulative; the accumulation is done first for non-deletable files, then for deletable files; within these classes, more recent files are done first. If a file exceeds all three limits, its name is highlighted initially. Selecting Flag On/Off inverts the highlighting on deletable files. Selecting Delete deletes all highlighted files; the actual deletion is preceded by a confirmation dialog.

Functions and Operators in Epi Info 2000

Operators ARITHMETIC + - * / ^ MOD COMPARISON > = <	**Boolean Operators** AND OR XOR NOT	**Numeric Functions** EXP SIN, COS, TAN LOG LN ABS RND TRUNC ROUND NUMTODATE NUMTOTIME
Date and Time Functions YEARS MONTHS DAYS YEAR MONTH DAY	**Time Functions** HOURS MINUTES SECONDS HOUR MINUTE SECOND	**Text Functions** TXTTONUM TXTTODATE SUBSTRING UPPERCASE FINDTEXT FORMAT
	System Functions SYSTEMDATE SYSTEMTIME RECNUMBER RECVERIFIED RECDELETED ENVIRON EXISTS FILEDATE (.)	

Functions and Operators

Modifying Values Within Expressions

Overview

Expressions are combinations of literal values, variables, functions, and operators that can be evaluated to a single result. Expressions can be used in ***MakeView*** (check files) and ***Analysis***. Within an expression, the values of variables can be modified by a number of functions and operators. .

An Expression consists of one or more Operands (variables or literal values) and one or more Operators (like +, -, *, and /). Expressions, no matter how complex, can eventually be evaluated to produce a single value, like 1.323, or “True” or “False.”

Functions modify the value of one or more variables to produce a result. For example, ROUND (2.33333) produces the value 2.. Operators are used to combine two items. For example, the “+” operator combines Var1 and Var2 to produce a sum, as in Var3=Var1+Var2.

The many functions and operators available in Epi Info 2000 are described in this chapter.

Operators

Arithmetic (+ - * / ^ MOD)

Operators

Description (See individual descriptions below, under Syntax.)

Syntax [Expression] *<Operator>* [Expression]

Syntax Element		Description
<Operator>	^	Exponentiation
	+ , -	Unary minus and plus
	* , /	Numeric Multiplication or Division
	MOD	Modulus or Remainder
	&	String Concatenation
[Expression]	A numeric value or a variable containing data in numeric format	

Comments The result will be expressed in numeric format. The basic mathematical operators that can be used in Epi Info 2000 are Addition (+), Subtraction (-), Multiplication (*), Division (/), Exponentiation (^), and Modulo or Remainder (MOD). Mathematical operators can be used in combination with other commands. Arithmetic operators are shown in order of precedence. Parentheses can be used to control the order in which operators are evaluated, although frequently the default order will achieve the right result.

Example The check program can calculate current age:

AGE = (TODAY – DOB) / 365.25

where TODAY and DOB are date variables. In ***Analysis***, commands can be combined with mathematical operators. For example:

SELECT ((date1 – date2) = 50)

In this statement, ***Analysis*** will select only those records for which the difference between date1 and date2 is 50 days.

ASSIGN <*Var1*> = 1250 MOD 100

Variable <Var1> will have the value of 50 (the remainder after 1250 is divided by 100).

$$\text{¿} Var1 >= \frac{1,250}{100} = 1.250$$

See Also Comparison Operators, Boolean Operators

COMPARISON (> = <)

Operators

Description (See individual descriptions below, under Syntax.)

Syntax [Expression] *<Operator>* [Expression]

Syntax Element	Description	
<Operator>	=	Equal to
	>	Greater than
	<	Less than
	>=	Greater than or equal to
	<=	Less than or equal to
	<>	Not equal to
[Expression]	Any valid expression	

Comments Comparison operators can be used in If Then, Select, Relate, Merge, Recode, and assign statements in both Check code and the ***Analysis*** program.

Operator	Description	Alternatives
>	Greater Than	
>=	Greater Equal	=>
<	Less than	
<=	Less equal	=>
<>	Different	NOT =
=	Equal	

Comparison operators will be executed from left to right. There is no hierarchy among them.

Example

```
READ ‘D:\EPI2000\SAMPLE.MDB’:VIEW OSWEGO
SELECT AGE < 15
```

Records selected will include ages 0 to 14 years.

See Also NOT, SELECT

Boolean Operators (AND, OR, XOR)

Epi Info 2000 can use AND, OR, and XOR logical operators to perform specific commands.

AND

Operator

Description Logical AND. If all conditions are true, it returns true; otherwise, it returns false.

Syntax [Expression] **AND** [Expression]

Syntax Element	Description
[Expression]	Any valid expression in Epi Info 2000

Comments AND can be used with SELECT, IF THEN, SORT, and RELATE commands

Example

```
READ FILE1
DEFINE RESULT
IF AGE < 18 AND SEX = “F” THEN
   RESULT = “GIRL”
END
```

Age	Sex	Result
5	F	GIRL
7	M	<Missing>
18	F	<Missing>
21	F	<Missing>
27	M	<Missing>

In this case the value of “GIRL” will be assigned to all records that meet both criteria (Age < 18) and (sex = “F”).

See Also OR, XOR, NOT

OR

Operator

Description Logical OR. Returns true if at least one of the expressions is true.

Syntax [Expression] **OR** [Expression]

Synt	Description

ax Element	
[Expression]	Any valid expression in Epi Info 2000

Comments OR can be used with SELECT, IF THEN, SORT, and RELATE commands

Example

```
READ C:\EPI2000\SAMPLES.MDB:VIEW OSWEGO
DEFINE ICE_CREAM
IF VANILLA = (+) OR (CHOCOLATE=(+) THEN
   ICE_CREAM = (+)
ELSE
  ICE_CREAM = (-)
END
```

VANILLA	CHOCOLATE	ICE_CREAM
Yes	Yes	Yes
No	Yes	Yes
Yes	No	Yes
No	No	No
Yes	Yes	Yes

See Also AND, XOR, NOT, IF

XOR (eXclusive OR)

Operator

Description Logical XOR. Returns true if one and only one value is true, and false if both are true or both are false

Syntax [Expression] **XOR** [Expression]

Syntax Element	Description
[Expression]	Any Valid Expression in Epi Info 2000

Comments XOR can be used with SELECT, IF THEN, SORT, and RELATE commands

Example

```
READ C:\EPI2000\SAMPLE.MDB:VIEW OSWEGO
DEFINE ONE_ICE_CREAM
IF VANILLA = (+) XOR CHOCOLATE = (+) THEN
    ONE_ICE_CREAM = (+)
ELSE
    ONE_ICE_CREAM = (-)
END
```

VANILLA	CHOCOLATE	ICE_CREAM
Yes	Yes	No
No	Yes	Yes
Yes	No	Yes
No	No	No
Yes	Yes	No

See Also AND, OR, NOT, IF

NOT

Operator

Description

Reverses the True or False value of the expression that follows

Syntax

NOT [Expression] *<Operator>* [Expression]

Syntax Element	Description
[Expression]	Any valid expression in Epi Info 2000
<Operator>	Any valid operator

Comments

If the value of an expression is TRUE, NOT returns the value FALSE. If the expression is FALSE, NOT (<expression>) is TRUE.

Example

```
READ C:\EPI2000\SAMPLE.MDB:VIEW OSWEGO
DEFINE NO_VANILLA
IF NOT VANILLA  = (+) THEN
     NO_VANILLA = (+)
ELSE
     NO_VANILLA = (-)
END
```

VANILLA	NO_VANILLA
Yes	No
No	Yes

See Also

AND, OR, NOT, IF

Numeric Functions

EXP

Function

Description Raises the base of the natural logarithm to the power of (X)

Syntax **EXP**(*<Variable>*)

Syntax Element	Description
Comments *<Variable>*	Can be an existing numeric variable, a defined variable containing numbers, or a numeric constant

Example

```
READ 'D:\EPI2000\SAMPLE.MDB': VIEW OSWEGO
DEFINE EXP_A
EXP_A = EXP(AGE)
LIST EXP_A
```

See Also LOG, LN

SIN, COS, TAN

Function

Description Returns the respective trigonometric value for the specified variable

Syntax **SIN**(*<Variable>*)

Syntax Element	Description
<Variable>	Can be an existing numeric variable, a defined variable containing numbers, or a numeric

	constant

Comments The result will always be expressed in radians.

Example

```
READ 'D:\EPI2000\SAMPLE.MDB': VIEW OSWEGO
DEFINE SIN_A
SIN_A = SIN(AGE)
LIST SIN_A
```

See Also How to transform radians to degrees

LOG

Function

Description Returns the base 10 logarithm (decimal logarithm) of a numeric value or variable. If the value is not numeric, it will return a null value.

Syntax **LOG**(*<Variable>*)

Syntax Element	Description
<Variable>	Can be an existing numeric variable, a defined variable containing numbers, or a numeric constant

Comments The result will be expressed in numeric format.

Example

```
READ 'D:\EPI2000\SAMPLE.MDB': VIEW OSWEGO
DEFINE DEC_LOG
DEC_LOG = LOG (AGE)
LIST AGE DEC_LOG
```

See Also LN

LN

Function

Description Function LN will return the natural logarithm (logarithm in base e) of a numeric value or variable. If the value is not numeric, it will return a null value.

Syntax **LN**(*<Variable>*)

Syntax Element	Description

<Variable>	Can be an existing numeric variable, a defined variable containing numbers, or a numeric constant

Comments The result will be expressed in numeric format.

Example

```
READ 'D:\EPI2000\SAMPLE.MDB': VIEW OSWEGO
DEFINE NATLOGOFAGE
NATLOGOFAGE = LN(AGE)
LIST AGE NATLOGOFAGE
```

See Also LOG

ABS

Function

Description Function ABS will return the absolute value of a variable by removing the negative sign, if any.

Syntax **ABS**(*<Variable>*)

Synt ax Ele men t	Description
<Variable>	Can be an existing numeric variable, a defined variable containing numbers, or a numeric constant.

Comments The result will be expressed in numeric format.

Value	ABS Function
-2	2
1	1
0	0
-0.0025	0.0025

Example

```
READ 'D:\EPI2000\SAMPLE.MDB': VIEW OSWEGO
DEFINE AGE2
DEFINE AGE3
AGE2 = AGE * -1
AGE3 = ABS (AGE2)
LIST AGE AGE2 AGE3
```

See Also TRUNC

RND

Function

Description RND will generate a random number between *<Var1>* and *<Var2>*. The optional seed controls the starting point. The same seed always generates the same series of random numbers.

Syntax **RND**(*<Min>* , *<Max>* , *<Seed>*)

Syntax Element	Description
<Min>	A number or numeric variable that corresponds to the lowest value for the random number to be generated
<Max>	A number or numeric variable that corresponds to the highest possible value for the random number to be generated
<Seed>	Seed for random sequence. Giving the same seed causes a series of random numbers to be the same, that is, to start at the same point. Each successive number is then random, but repeating the listing will generate the same series.

Comments The result will be returned in numeric decimal format. It can be converted to integers with the INT function.

Example

```
READ 'D:\EPI2000\SAMPLE.MDB': VIEW OSWEGO
DEFINE Random1
DEFINE Random2
DEFINE Random3
Random1=RND(1,100,1)
Random2=RND( 1,100,2)
Random3=RND(1,100,3)
LIST  Random1 Random2 Random3
```

See Also RECNUMBER, TRUNC

TRUNC

Function

Description Removes decimals from a numeric variable, returning the integer part of the number. Can be thought of as "rounding down."

Syntax **TRUNC**(*<Variable>*)

Synt	Description

ax Ele men t	
<Variable>	Can be an existing numeric variable, a defined variable containing numbers, or a numeric constant.

Comments The result will be returned in numeric format.

Example

```
READ ‘D:\EPI2000\SAMPLE.MDB’: VIEW OSWEGO
DEFINE Random1
DEFINE Random2
DEFINE Random3
Random1=RND(1,100,1)
Random2=RND( 1,100,2)
Random3=RND(1,100,3)
DEFINE TRC1
DEFINE TRC2
DEFINE TRC3
TRC1 = TRUNC(Random1)
TRC2 = TRUNC(random2)
TRC3 = TRUNC(RANDOM3)
LIST TRC1 RANDOM1 TRC2 RANDOM2 TRC3 RANDOM3
```

See Also ROUND

ROUND

Function

Description Rounds the number stored in the variable to the closest integer, rounding up to the next integer if the number is greater than or equal to the previous integer + 0.5.

Syntax **ROUND** (<*Variable*>)

Synt ax Ele men t	Description
<Variable>	Can be an existing numeric variable, a defined variable containing numbers, or a numeric constant.

Comments The result will be returned in numeric format.

Differences between ROUND and TRUNC

Value	Trunc	Round
0.123456	0	0
7.999999999	7	8

45.455	45	46

Example

```
READ 'D:\EPI2000\Sample.Mdb':viewOswego
FREQ  AGE
DEFINE Decade
Decade = ROUND(AGE/10) + 1
LIST  AGE Decade
```

See Also TRUNC

NUMTODATE

Function

Description Transforms three numbers into a date format.

Syntax **NUMTODATE**(*<Year>*,*<Month>*,*<Date>*)

Syntax Element	Description
<Year>	A numeric variable or a number representing the year
<Month>	A numeric variable or a number representing the month
<Day>	A numeric variable or a number representing the day

Comments If the date resulting from the conversion is not valid, the date will be recalculated to the corresponding valid value. When *<Year>* ranges between 0 and 29, it will be represented as the respective year between 2000 and 2029, while values from 30 to 99 will be represented as the respective year between 1930 and 1999. The oldest date that can be recorded is Jan 01, 100.

Day	Month	Year	Date created
02	02	1999	02/02/1999
60	01	1999	03/01/1999
15	18	2000	03/18/2001
99	99	99	06/07/0107
20	08	74	08/20/1974

Example

```
READ 'D:\epi2000\My File.mdb':viewMain
DEFINE Day1
DEFINE month1
DEFINE year1
day1= day(dob)
month1 = month(dob)
```

```
year1 = year(dob)
define date2
date2= NUMTODATE(year1,month1,day1)
```

See Also FORMAT

NUMTOTIME

Function

Description Transforms three numbers into a time or date/time format.

Syntax **NUMTOTIME** (*<Hour>*, *<Minute>*, *<Second>*)

Syntax Element	Description
<Hour>	Numeric constant or variable representing hours
<Minute>	Numeric constant or variable representing minutes
<Second>	Numeric constant or variable representing seconds

Comments Time must be entered in 24-hour format. Invalid dates will be recalculated to the respective valid time. If the number of the hour exceeds 24, the resulting variable will have a date/time format and the default day 1 will be December 31, 1899.

Hour	Minute	Second	Time created
00	00	00	12:00:00 AM
00	00	90	12:01:30 AM
15	84	126	04:26:06 PM
25	00	00	12/31/1899 1:00:00 AM
150	250	305	01/05/1900 10:15:05 AM
15000	7500	8954	09/21/1901 07:29:14 AM

Example

```
READ 'D:\EPI2000\SAMPLE.MDB':VIEWOSWEGO
DEFINE VAR3
VAR3 = SYSTEMTIME
DEFINE HOUR1
DEFINE MINUTE1
DEFINE SECOND1
HOUR1 = HOUR(VAR3)
MINUTE1 = MINUTE(VAR3)
SECOND1 = SECOND(VAR3)
DEFINE TIME2
TIME2 = NUMTOTIME(HOUR1,MINUTE1,SECOND1)
LIST VAR3 TIME2
```

See Also NUMTODATE, FORMAT

Date and Time Functions

YEARS

Function

Description

Returns the number of years from *<VAR1>* to *<VAR2>* in numeric format. If any of the variables or values included in the formula is not a date, the result will be Null.

Syntax

YEARS(*<VAR1>, <VAR2>*)

Syntax Element	Description
<Var1>	Variable in date format
<Var2>	Variable in date format

Comments

NOTE: In the current implementation, the function merely subtracts the year portion of the dates. Hence 12/31/1999 and 01/01/2000 will have a difference of 1 year, although we would consider a child born on the former date to be one day old on the latter date.

If the date stored in Var1 is more recent than that in Var2, the result will be the difference in years expressed as a negative number.

Example

```
READ 'C:\EPI2000\SAMPLE.mdb':viewSURVEILLANCE
DEFINE TODAY
TODAY = ‘01/01/2000’
DEFINE AGE_YEARS
AGE_YEARS = YEARS (DOB, TODAY)
MEANS AGE_YEARS
```

See Also

YEAR, MONTH, MONTHS, DAY, DAYS

MONTHS

Function

Description

Returns the number of months between *<Var2>* and *<Var1>*. If any of the variables or values included in the formula is not a date, the result will be Null.

Syntax

MONTHS(*<VAR1>, <VAR2>*)

Syntax Elemen	Description

t	
<Var1>	Variable in date format
<Var2>	Variable in date format

Comments If the date stored in Var1 is older than that in Var2, the result will be the difference in months expressed as a negative number.

Example

```
READ ‘C:\EPI2000\MYFILE.MDB’:VIEW DATES
DEFINE AGE_MONTHS
AGE_MONTHS = MONTHS (DOB, 01/01/2000)
LIST AGE_MONTHS
```

See Also YEAR, YEARS, MONTH, DAY, DAYS

DAYS

Function

Description Returns the number of days between *<Var2>* and *<Var1>*. If any of the variables or values included in the formula is not a date, the result will be Null.

Syntax **DAYS**(*<VAR1>, <VAR2>*)

Syntax Element	Description
<Var1>	Variable in date format
<Var2>	Variable in date format

Comments If the date stored in Var1 is more recent than that in Var2, the result will be the difference in days expressed as a negative number.

Example

```
READ ‘C:\EPI2000\MYFILE.MDB’:VIEW DATES
DEFINE AGE_DAYS
AGE_DAYS = DAYS (DOB, 01/01/2000)
LIST AGE_DAYS
```

See Also YEAR, YEARS, MONTH, MONTHS, DAY

YEAR

Function

Description Extracts the year from a date.

Syntax **YEAR**(*<Variable>*)

Syntax Element	Description
<Variable>	Variable in date format

Comments If the date is stored in a Text variable, the function will not be processed and the result will be Null.

Example

```
READ ‘C:\EPI2000\SAMPLE’:VIEW OSWEGO
DEFINE CURR_YEAR
CURR_YEAR = YEAR(01/01/1990)
LIST CURR_YEAR
```

See Also YEARS, MONTH, MONTHS, DAY, DAYS

MONTH

Function

Description Extracts the month from a date.

Syntax **MONTH**(*<Variable>*)

Syntax Element	Description
<Variable>	Variable in date format

Comments If the date is stored in a Text variable, the function will not be processed and the result will be Null.

Example

```
READ ‘C:\EPI2000\SAMPLE’:VIEW OSWEGO
DEFINE CURR_MONTH
CURR_MONTH = MONTH(01/01/1990)
LIST CURR_MONTH
```

See Also YEAR, YEARS, MONTHS, DAY, DAYS

DAY

Function

Description Extracts the day from a date.

Syntax **DAY**(*<Variable>*)

Syntax Element	Description
<Variable>	Variable in date format

Comments If the date is stored in a Text variable, the function will not be processed and the result will be Null.

Example

```
READ ‘C:\EPI2000\SAMPLE’:VIEW OSWEGO
DEFINE Curr Day
Curr_Day = DAY(01/01/1990)
LIST CURR_Day
```

See Also YEAR, YEARS, MONTH, MONTHS, DAYS

Time Functions

HOURS

Function

Description Returns the number of hours between *<VAR2>* and *<VAR1>* in numeric format.

Syntax **HOURS**(*<VAR1>, <VAR2>*)

Syntax Element	Description
<Var1>	Variable in time or date/time format
<Var2>	Variable in time or date/time format

Comments If the date stored in *<Var1>* is older than that in *<Var2>*, the result will be the difference in hours expressed as a negative number. Both variables must contain data in date, time, or date/time format. If any of the variables or values included in the formula is not a date, the result will be Null.

Example

```
READ ‘C:\EPI2000\SAMPLE’:VIEW OSWEGO
DEFINE HOUR1
```

```
HOUR1=HOURS(DATE,DTOFARR)
LIST  HOUR1
```

See Also HOUR, MINUTES, MINUTE, SECONDS, SECOND

MINUTES

Function

Description Returns the number of minutes between *<VAR2>* and *<VAR1>* in numeric format.

Syntax **MINUTES**(*<VAR1>, <VAR2>*)

Syntax Element	Description
<Var1>	Variable in time or date/time format
<Var2>	Variable in time or date/time format

Comments If the date stored in *<Var1>* is older than that in *<Var2>*, the result will be the difference in minutes expressed as a negative number. Both variables must contain data in date, time, or date/time format. If any of the variables or values included in the formula is not a date, the result will be Null.

Example

```
READ ‘C:\EPI2000\SAMPLE’:VIEW OSWEGO
DEFINE MIN1
MIN1=MINUTES(DATE,DTOFARR)
LIST  MIN1
```

See Also HOURS, HOUR, MINUTE, SECONDS, SECOND

SECONDS

Function

Description

Returns the number of seconds between *<VAR2>* and *<VAR1>* in numeric format.

Syntax

SECONDS(*<VAR1>, <VAR2>*)

Syntax Element	Description
<Var1>	Variable in time or date/time format
<Var2>	Variable in time or date/time format

Comments

If the date stored in *<Var1>* is older than that in *<Var2>*, the result will be the difference in seconds expressed as a negative number. Both variables must contain data in date, time or date/time format. If any of the variables or values included in the formula is not a date, the result will be Null.

Example

```
READ ‘C:\EPI2000\SAMPLE’:VIEW OSWEGO
DEFINE SEC1
SEC1=SECONDS(DATE,DTOFARR)
LIST  SEC1
```

See Also

HOURS, HOUR, MINUTES, MINUTE, SECOND

HOUR

Function

Description

Returns a numeric value that corresponds to the hour recorded in a date/time or time variable.

Syntax

HOUR(*<Variable>*)

Syntax Element	Description
<Variable>	Variable in date format

Comments

If the time is stored in a Text variable, the function will not be processed and the result will be Null.

Example

```
READ 'C:\EPI2000\SAMPLE':VIEW OSWEGO
DEFINE Local
Local = SYSTEMTIME
LIST  Local
DEFINE hour1
hour1=hour(local)
LIST  Local hour1
LIST  Local
LIST  hour1
```

See Also HOURS, MINUTES, MINUTE, SECONDS, SECOND

MINUTE

Function

Description Returns a numeric value that corresponds to the minute recorded in a date/time or time variable.

Syntax **MINUTE**(*<Variable>*)

Synt ax Ele men t	Description
<Variable>	Variable in date/time or time format

Comments If the time is stored in a Text variable, the function will not be processed and the result will be Null.

Example

```
READ 'C:\EPI2000\SAMPLE':VIEW OSWEGO
DEFINE Local
Local = SYSTEMTIME
LIST  Local
DEFINE min1
min1= minute(local)
LIST  Local min1
LIST  Local
```

See Also HOURS, HOUR, MINUTES, SECONDS, SECOND

SECOND

Function

Description Returns a numeric value that corresponds to seconds recorded in a date/time or time variable.

Syntax **SECOND**(*<Variable>*)

Syntax Element	Description
<Variable>	Variable in date/time or time format

Comments If the time is stored in a Text variable, the function will not be processed and the result will be Null.

Example

```
READ ‘C:\EPI2000\SAMPLE’:VIEW OSWEGO
DEFINE Local
Local = systemtime
LIST  Local
DEFINE sec1
sec1= second(local)
LIST  Local sec1
```

See Also HOURS, HOUR, MINUTES, MINUTE, SECONDS

Text Functions

TXTTONUM

Function

Description Returns a numeric value that corresponds to the string.

Syntax **TXTTONUM**(*<Variable>*)

Syntax Element	Description
<Variable>	Variable in text format

Comments Old comment was deleted. Is there a replacement?

Example Incorrect:

```
VAR1 = TXTTONUM(VAR1)
```

Correct:

```
    DEFINE VAR2
VAR2 = TXTTONUM (VAR1)
```

See Also TXTTODATE, FORMAT, SUBSTRING

TXTTODATE

Function

Description Returns a numeric value that corresponds to the string.

Syntax **TXTTODATE**(*<Variable>*)

Syntax Element	Description
<Variable>	Variable in text format

Comments .Old comment was deleted, but not replaced.

Example Incorrect:

```
VAR1 = TXTTODATE (VAR1)
```

Correct:

```
DEFINE VAR2
VAR2 = TXTTODATE (VAR1)
```

See Also TXTTONUM, FORMAT, SUBSTRING

SUBSTRING

Function

Description SUBSTRING(*<Variable>* , [First], [Length])

Syntax

Syntax Element	Description
<Variable>	Variable in text format
[First]	The first character to extract from the file
[Length]	The number of characters to extract

Comments SUBSTRING cannot be used with non-string variables.

Example

```
READ ‘C:\EPI2000\SAMPLE’:VIEW OSWEGO
DEFINE TEXT1
TEXT1 = “01/01/1999”
DEFINE YEAR1
YEAR1 = TXTTONUM(SUBSTRING(TEXT1,7,4))
MEANS YEAR1
```

See Also TXTTONUM, FORMAT, TXTTODATE

UPPERCASE

Function

Description Transforms the content of a string (text) variable to uppercase letters.

Syntax **UPPERCASE**(*<Variable>*)

Syntax Element	Description
<Variable>	Variable in text format

Comments

Example

```
READ 'D:\EPI2000\MyFile.Mdb':viewDemographics
DEFINE LASTNAME2
LASTNAME2 = UPPERCASE(LASTNAME)
LIST LASTNAME2 LASTNAME
```

See Also SUBSTRING, FINDTEXT

FINDTEXT

Function

Description Returns the position in a variable in which the string is located.

Syntax **FINDTEXT**([String],*<Variable>*)

Syntax Element	Description
<Variable>	Variable in text format
[String]	String of characters to be found

Comments If the string is not found, the result is 0; otherwise it is a number corresponding to the position of the string starting from the left.

Example

```
READ ‘C:\EPI2000\SAMPLE’:VIEW OSWEGO
DEFINE VAR11
VAR11=FINDTEXT("H",LASTNAME)
LIST  LASTNAME VAR11
```

See Also SUBSTRING, TRUNC, FORMAT

FORMAT

Function

Description Transforms the content of a string (text) variable in uppercase letters.

Syntax **FORMAT**(*<Variable>* [, “Format specification”])

Syntax Element	Description	
<Variable>	Variable in any format	
[Format Specification]	Any of the following words	
	a) Date formats	
	General Date	11/11/1999 05:34
	Long Date	System’s Long date format
	Medium Date	System’s Medium date format
	Short Date	System’s Short date format
	Long Time	System’s Long time
	Medium Time	System’s Medium time
	Short time	System’s Short time
	b) Number formats	
	General Number	No thousand separator
	Currency	Thousand separator plus two decimal places (based on system settings)
	Fixed	At least #.##
	Standard	#,###.##
	Percent	Number divided by 100 plus a percent sign
	Scientific	Standard scientific notation
	Yes/No	Displays NO if number = 0, else displays Yes
	True/False	False if number = 0 True if number <> 0
	On/Off	Displays 0 if number = 0, else Displays 1
	Custom format	Allows user to create customized formats

Comments Output may vary based on specific configuration settings of the computer.

```
Format(Time, "Long Time")

MyStr = Format(Date, "Long Date")
MyStr = Format(MyTime, "h:m:s")   ' Returns "17:4:23".
MyStr = Format(MyTime, "hh:mm:ss AMPM")   ' Returns
"05:04:23 PM".
MyStr = Format(MyDate, "dddd, mmm d yyyy")   ' Returns
"Wednesday, ' Jan 27 1993".

' If format is not supplied, a string is returned.
MyStr = Format(23)   ' Returns "23".

' User-defined formats.
MyStr = Format(5459.4, "##,##0.00")   ' Returns
"5,459.40".
MyStr = Format(334.9, "###0.00")   ' Returns "334.90".
MyStr = Format(5, "0.00%")   ' Returns "500.00%".
MyStr = Format("HELLO", "<")   ' Returns "hello".
MyStr = Format("This is it", ">")   ' Returns "THIS IS
IT".
MyStr = Format("This is it", ">;*")   ' Returns "THIS IS
IT".
```

The letters M,T, H, S are restricted in this format. To include those letters as a string, type the letter preceded by "\". For more information about format, please refer to Microsoft Access or Microsoft Visual Basic documentation.

Example

```
READ 'D:\EPI2000\MyFile.Mdb':viewDemographics
DEFINE date2
DEFINE date3
DEFINE date4
DEFINE date5
DEFINE date6
DEFINE date7
DEFINE date8
DEFINE date9
DEFINE date10
DEFINE date11
date2=format(BOH, "Currency")
date3=format(BOH, "fixed")
date4=format(BOH, "Standard")
date5=format(BOH, "Percent")
date6=format(BOH, "Scientific")
date7=format(BOH, "Yes/No")
date8=format(BOH, "True/false")
date9=format(BOH, "On/Off")
date10=format(BOH, "VB\s #,###.##")
list dob date2 date3 date4 date5 date6 date7 date8 date9
date10
```

See Also TXTTONUM, TXTTODATE, NUMTODATE,

System Functions

SYSTEMDATE

Function

Description Returns the date stored in the computer's clock.

Syntax **SYSTEMDATE**

Comments SYSTEMDATE cannot be changed (assigned) from ***Analysis***. To use the systemdate for computations, a new variable must be defined. For example, to calculate next week's date:

Example

```
READ 'D:\EPI2000\MyFile.Mdb':viewDemographics
DEFINE TODAY_DATE
TODAY_DATE = SYSTEMDATE + 7
```

See Also SYSTEMTIME

SYSTEMTIME

Function

Description Returns the time stored in the CPU clock at the time the command is executed.

Syntax SYSTEMTIME

Comments SYSTEMTIME cannot be changed from ***Analysis*** (assign). To use the system time for computations, a new variable must be defined. For example, to calculate a time 2 hours after current time:

Example

```
READ 'D:\EPI2000\MyFile.Mdb':viewDemographics
DEFINE LATER
LATER = SYSTEMTIME + (720 / 24 / 60 / 60)
```

Variable LATER stores current time plus 720 seconds (12 minutes).

See Also SYSTEMDATE

RECNUMBER

Function

Description Returns an integer corresponding to the record number.

Syntax RECNUMBER

Comments RECNUMBER cannot be changed from ***Analysis***, but can be assigned to a variable.

Example

```
READ 'D:\EPI2000\MyFile.Mdb':viewDemographics
SELECT RECNUMBER < 200
```

Selects the first 200 records in the database.

See Also How to obtain a random sample from a database, at the end of this chapter.

ENVIRON

Function

Description Returns the value of a DOS Environment variable, such as PATH or COMSPEC.

Syntax ENVIRON (<Name of DOS Environment Variable>)

Syntax Element	Description
<Variable>	Variable in text format

Comments

Example

```
READ 'D:\EPI2000\MyFile.Mdb':viewDemographics
```

See Also

EXISTS

Function

Description Returns TRUE if a file exists; otherwise returns FALSE.

Syntax EXIST (<Complete FilePath and Name>)

Syntax Element	Description
<Variable>	Variable in text format

Comments May return FALSE if user does not have permission to access the file.

Example

```
IF EXIST (“C:\Epi2000\Test.BAT”  then
Etc.
```

See Also

FILEDATE

Function

Description Returns the date a file was last modified (or created).

Syntax FILEDATE (“C:\Epi2000\EPI2000.MNU”)

Syntax Element	Description
<Variable>	Variable in text format

Comments

Example

See Also

(.)

Function

Description Represents “Missing Value” in an expression.

Syntax (.)

Syntax Element	Description
<Variable>	Variable in text format

Comments The “(.)” represents missing values

Example

```
READ 'D:\EPI2000\MyFile.Mdb':viewDemographics
DEFINE LATER
LATER = SYSTEMTIME
LATER = (.)
```

See Also UNDEFINE

Summary of Functions

Fuction	Syntax	Description
+	<Var1> = <Vari2> + <Var3>	<Var1> Stores the sum between <Var2> and <Var3>
-	<Var1> = <Vari2> - <Var3>	<Var1> Stores the difference between <Var2> and <Var3>
*	<Var1> = <Vari2> * <Var3>	<Var1> Stores the product of <Var2> times <Var3>
/	<Var1> = <Vari2> / <Var3>	<Var1> Stores the difference of <Var2> times <Var3>
^	<Var1> = <Vari2> ^ <Var3>	<Var1> Stores the result of <Var2> to the power of <Var3>
MOD	<Var1> = <Var2> **MOD** <Var3>	Returns The remainder for <var2> divideb by <var3>
AND	<Var1> = <Var2> **AND** <Var3> = <Var4>	True if both conditions are met
OR	<Var1> = <Var2> **OR** <Var3> = <Var4>	True if any of those conditions are met
XOR	<Var1> = <Var2> **XOR** <Var3> = <Var4>	
NOT	<Var1> **NOT** = <Var2>	True if <Var2> is not equal than Var1
=	<Var1> = <Var2>	True when <Var1> has the same value or structure than <Var2>
<>	<Var1> <> <Var2>	True when <Var1> has different value or structure than <Var2>
<	<Var1> < <Var2>	True when <Var1> is smaller than <Var2>
<=	<Var1> <= <Var2>	True when <Var1> is smaller or equal <Var2>
>	<Var1> > <Var2>	True when <Var1> is greater than <Var2>
>=	<Var1> >= <Var2>	True when <Var1> is greater equal than <Var2>
LIKE		
TXTTONUM	**TXTTONUM**(<Variable>)	
TXTTODATE	**TXTTODATE**(<Variable>)	
DATEFORMAT		
FORMAT	**FORMAT**(<Variable>,"**[Type]**")	"general" Produces string variables
SUBSTRING,	**SUBSTRING**(<Variable>,First, Length)	First is the first characto to include. Length is the number of characters
TRIM		
UPPERCASE	**UPPERCASE**(<Variable>)	
FINDTEXT	**FINDTEXT**([String],<Variable>) [Operator] possition	
YEARS	**YEARS**(<Date1>, <Date2>)	Return the difference between <Date2> and <Date1> in years
MONTHS	**MONTHS**(<Date1>, <Date2>)	Return the difference between <Date2> and <Date1> in Months
DAYS	**DAYS**(<Date1>, <Date2>)	Return the difference between <Date2> and <Date1> in Days
HOURS	**HOURS**(<Date1>, <Date2>)	Return the difference between <Date2> and <Date1> in Hours
YEAR	**YEAR**(<Variable>)	Returns the Year (in Numeric format) from a date or date/time variable.
MONTH	**MONTH**(<Variable>)	Returns the Month (in Numeric format) from a date or date/time variable.
DAY	**DAY** (<Variable>)	Returns the Day (in Numeric format) from a date or date/time variable.
HOUR	**HOUR** (<Variable>)	Returns the Hour (in Numeric format) from a date or date/time variable.
MINUTE	**MINUTE** (<Variable>)	Returns the Minute (in Numeric format) from a date or date/time variable.
SECOND	**SECOND** (<Variable>)	Returns the Second (in Numeric format) from a date or date/time variable.
MDY	**MDY(**<Variable>,<Variable>, <Variable>**)**	Transforms three numbers into a date string
SYSTEMDATE	<Variable> = **SYSTEMDATE**	Assigns the Date to <Variable>
SYSTEMTIME	<Variable> = **SYSTEMTIME**	Assigns current time to <Variable>
HMS	**HMS(**<Variable>,<Variable>, <Variable>**)**	
RND	**RND**(<Var1>, <Var2>, <Var3>).	<Var1> represents min output, <Var2> max output, and <Var3> optional seed).
LOG	**Log (**<Variable>**)**	Calculates the logaritm in base 10
LN	**Ln (**<Variable>**)**	Calculates the logarithm of Variable in base n
EXP	**EXP**(<Variable>)	
TRUNC	**TRUNC**(<Variable>)	Removes decimals from <Variable>
ROUND	**ROUND**(<Variable>)	Rounds the value of <Variable>
ABS	**ABS**(<Variable>)	Removes sign from negative values
SIN, COS, TAN	**<Function> (**<Variable>**)**	Returns the trigonometric function in radiants
RECNUMBER	**RECNUMBER()**	
RECVERIFIED		
RECDELETED	**RECDELETED()**	
ENVIRON	**ENVIRON**(EnvVar)	Returns the value of the environment variable EnvVar. Both the argument and the result are strings.
EXISTS	**EXISTS**(File)	Returns true if the file File exists. File is a string and Exists is a YesNo

Limitations and Problem Resolution

How Far Toward the Rainbow Can We Go?

Limits

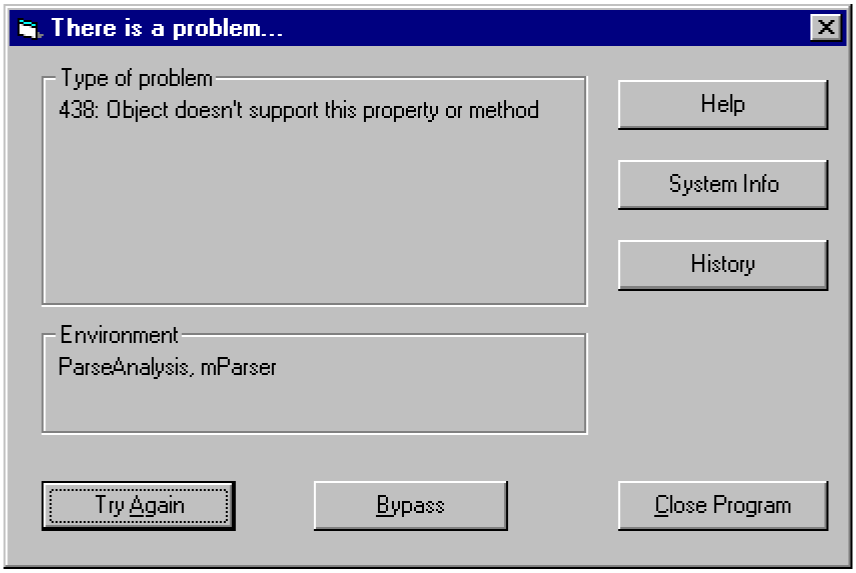
Epi Info 2000 inherits some limitations from the Windows environment and from microcomputers in general. A fairly fast processor (100 MHz or faster) is required for pleasant operation. For slower processors in particular, increasing Random Access Memory (RAM) beyond 32 or 64 megabytes will speed up program operation, although, at current prices, a new and faster computer may be a better choice.

Specific limitations in Epi Info 2000 include:

Number of fields per Questionnaire or View	A Microsoft Access table is limited to 256 fields, but Epi Info automatically creates additional related Views to remove this limitation. We have tested a View with XXX fields and XXX pages, for example.
Number of records in a data table	
Number of tables in a .MDB database file	1000
Number of tables open at one time in ***Analysis***	
Number of Views open at one time in ***Enter***	
Number of characters in a text field	128
Number of digits in a numeric field	
Number of characters in a multiline field	Theoretically 2 gigabytes, but you may get tired of typing or run out of disk space first
Number of layers in a map	
Number of points in a polygon in a .SHP file	

Problem Resolution

Since many "errors" are not the fault of the user, and can result from circumstances outside the current program, we prefer to call them "problems," and seek to resolve them as quickly as possible. Most of the programs in Epi Info contain a problem resolution module for handling "errors." When a problem arises that cannot be resolved by the program automatically, a dialog such as the following may appear.



There is a problem...

Type of problem

438: Object doesn't support this property or method

Environment

ParseAnalysis, mParser

The number and description of a Microsoft Visual Basic Error is displayed. At this point, several options are available. You can choose to BYPASS the error and see if the program functions normally. If the problem is a corrupt data file, for example, this may allow you to continue work. Sometimes a number of problem messages will occur one after the other, but clicking on BYPASS for each will eventually allow you to return to safe ground. If not, then choosing CLOSE PROGRAM will terminate the current program. You may lose work that has not been saved, but this may be necessary if BYPASS does not work. Since ***Enter*** and ***MakeView*** save each page automatically, the loss should not be great.

If a problem occurs repeatedly, further information can be obtained from the Problem module by choosing HISTORY or SYSTEM INFORMATION. HISTORY gives a list of the internal program functions that were called prior to the occurrence of the problem. The list may be helpful to the Epi Info programming team in resolving a problem. SYSTEM INFORMATION brings up a Windows function that provides a huge amount of information about the Windows operating system as it exists on the current computer at the time of the problem. SYSTEM INFORMATION can be used to determine the name, version, and date of all the Dynamic Linked Libraries (DLLs) on the computer, for example. SYSTEM INFORMATION is also available under SETTINGS on the main Epi Info 2000 menu.

If all else fails and a Windows program is hopelessly locked up, the time-honored "Ctrl-Alt-Delete" key combination should bring up a Windows dialog that shows what programs are running. It may indicate that one is "not responding," and allow the program to be terminated. It may also indicate that there are several instances of a single program in memory, and allow you to terminate the extra ones. This can be useful in resolving mysterious problems in writing to a file, caused by the fact that it is already being accessed by another copy of the same program.

Windows and DOS files have an attribute that can be set to “Read-Only.” This protects the file from changes, but also prevents its use as a database and prevents replacing it by copying another file of the same name into the same folder. To see the attributes of a file, use My Computer on the Windows desktop to find the file, right-click on the icon or name, and choose “Properties.” A check box in the property window indicates whether the file is “Read-Only.” Clicking in the check box will change the status to “unchecked” or writable. A grayed-out check mark means that some, but not all, of the files selected before the right click had read-only attributes set. In My Computer, Ctrl-A can be used to select all files within the current folder; you can then right-click and choose Properties to examine the status of all files in the folder.

Before contacting the Epi Info Helpline, it is reasonable to reboot Windows to see if a problem is resolved. It is also a good idea to ascertain that other programs are operating properly on the same computer, and that the Local Area Network to which you may be connected is not having temporary problems.

If these steps do not resolve the problem, contact information for the Epi Info Helpline is provided on the title page of this manual. Careful notes on the problem and on the computer environment in which Epi Info is installed will help in problem resolution. When reporting problems, please include the date of the Epi Info distribution version found on the bottom of the main menu.

Translation

Notes for Translators

Translating the Epi Info 2000 Programs

Epi Info 2000 is designed for easy translation into non-English languages. All the English phrases that can safely be translated are contained in tables in the database LANGUAGE.MDB. Reserved words in the ***Analysis*** and ***Check*** programming languages (see Appendix XXX, "Reserved Words") should not be translated, although the screen prompts that lead to their generation can be translated. Each Epi Info 2000 program has its own table in a LANGUAGE.MDB database (e.g., the phrases from ***Analysis*** are found in the ***Analysis*** table). Error messages common to several programs are found in the ERROR table. Each ***Menu*** (.MNU) file used in the menu produces a separate table in the database, which can be translated. English versions of these menu tables are generated automatically when a user creates a new MNU file.

Each language has its own directory of the installed Epi Info 2000 directory. Every installation has at least the directories ..\ENGLISH\ and ..\ENGLISH\HELP\, containing the English versions of the program phrases (in c:\Epi2000\ENGLISH\LANGUAGE.MDB, for example) and the English versions of the help files and manual. Exercises such as OSWEGO may also be located in subdirectories of the ..\ENGLISH\ directory. An installation program called TSETUP.EXE is provided to create proper directories for a new language and install the translated files.

To add a new language, first examine the LANGUAGE.MDB database, using ***VisData*** or Microsoft Access, to appreciate how the database is constructed. To make a translation in SPANISH, for example, first create a subdirectory called ..\SPANISH\ under the Epi Info installed directory (usually ..\EPI2000\). Make another subdirectory under ..\SPANISH\ called HELP. Then copy the English version of LANGUAGE.MDB to ..\SPANISH\. Use ***MakeView*** to create a new field called SPANISH in each of the program tables in LANGUAGE.MDB. Your translation should include the tables for each program and the ERRORS table. When you or a user create a new MNU file for the Epi Info 2000 ***Menu*** program, a new table is automatically created with a column containing the phrases from the MNU so that a translation can be provided in LANGUAGE.MDB. Now that you have a View and a Data table for each program, translating the program into Spanish is simply a matter of placing the Spanish translations beside the English phrases in the same record in the database. If no translation is needed, leave the item blank, and the program will automatically default to English. This would be appropriate for single characters or an "=" sign, for example.

After each translation is complete, set the LANGUAGE variable in EPIINFO.INI to the name of the desired column (e.g., Spanish) and then run the program. This can be done either by editing EPIINFO.INI or by editing the EPI2000.MNU file to allow the user to set the LANGUAGE variable equal to SPANISH. Several examples of these settings are provided in EPI2000.MNU.

After the LANGUAGE variable has been set to SPANISH, the programs should show Spanish phrases automatically rather than the English equivalents. Check the spacing of phrases on the programs screen and make appropriate changes if crowding occurs. A beta test should be conducted with representative users, to test the function of the programs and the clarity and correctness of the translation.

After the translation is complete, the Epi Info Team will insert it into the database for the next version of Epi Info if you send it to the Epi Info Help Line at epiinfo@cdc.gov.

Translating the Manual and Help Files

The manual and help files were written in Microsoft Word 2000 and saved as HTML files. They can be reconverted to Microsoft Word format by Word 2000, translated, and then saved once again as HTML files. If they are placed in the proper directory (e.g., ..\SPANISH\HELP\), they will be automatically displayed when the LANGUAGE variable is set to SPANISH through the choice on the main menu, which in turn sets LANGUAGE to SPANISH in the EPIINFO.INI file.

LANGUAGE.MDB in the Main Epi Info 2000 Directory

To demonstrate language capability, a single LANGUAGE.MDB is supplied with Epi Info 2000 that contains pseudotranslations in Spanish, French, Italian, Portuguese, and German. These translations were done by an inexpensive computer program (Easy Translator 2) and are not intended for serious use. The Epi Info language module looks first for LANGUAGE.MDB in the directory for the currently chosen language (e.g., SPANISH), and, if one is not found, looks in the main Epi2000 directory. Hence, if you provide a real translation for even a few of the programs, the machine translation will not be used.

Advantages of Having Separate Directories for Each Language

If you are preparing a translation for further distribution, DO NOT PLACE THE TRANSLATION IN THE MULTICOLUMN LANGUAGE.MDB in the main directory. Although doing so provides a quick demonstration mechanism, the system described above, with separate directories for each language, has many advantages and allows translators in various parts of the world to distribute their language modules independently, perhaps from a web site, without having to send them to Atlanta for incorporation into the core Epi Info system. Revisions can be made for any language without disturbing other translations.

Preparing Files for Installation by TSETUP.EXE

The Language database and help files must be placed in the proper directories and compressed (“zipped”) into a single, self-expanding module (a self-expanding EXE file) that can be run by the TSETUP.EXE program supplied with Epi Info 2000. The exact nature of the self-expanding EXE files is not crucial, but they must create directories of the correct name under the directory from which they are run. A commercially licensed copy of WinZip or a public-domain program can be used to create the files. The compressed executable file should be given the name of the language it contains—SPANISH.EXE in our example—and the file should be placed in the same directory with TSETUP.EXE. A translation of TSETUP.EXE itself can be provided by translators for end users, if desired.

Fonts, Unicode, Right-to-Left, and other Features Required for Non-European Languages

Epi Info is programmed to set its default font to that of the computer on which the program is running. The Visual Basic in which it is programmed is capable of displaying Unicode, the two-character code used for Chinese and other languages for which ASCII is not adequate. Arabic, Hebrew, and other right-to-left languages should display properly on computers that have this feature included in the Windows operating system. Since we have not been able to test this extensively, we look forward to working with translators to test and provide easy translation facilities.

Logistic Regression – The MVAWin Program

Questions and Answers

Authors: Cecile Delhumeau, Andrew Dean

How do I know when I need Logistic Regression?

In Epi Info 2000, either the TABLES command or logistic regression (LOGISTIC command) can be used when the outcome variable is dichotomous (for example, disease/no disease). Analysis with the TABLES command in Epi Info is adequate if there is only one "risk factor." Logistic regression is needed when the number of explanatory variables ("risk factors") is more than one. The method is often called "multivariate logistic regression."

Logistic regression shows the relationship between an outcome variable with two values and explanatory variables that can be categorical or continuous.

What kinds of epidemiologic data sets are particularly suitable for Logistic Regression? Not suitable?

Logistic regression can be applied to case-control, follow-up, and cross-sectional data. All types of variables (categorical and continuous) can be included in a logistic regression model, although categorical (coded) variables make results easier to interpret. The outcome variable must be dichotomous. Categorical data must be coded in "0/1", "0" being the class with the least exposure (Epi Info takes care of this for YES/NO variables).

Types of variables

Types of variable	Other terms ...	Examples ...
Continuous variables	Quantitative variables	Weight, Blood pressure
Categorical variables	Qualitative variables,	Presence or absence of a disease or

	discrete variables, nominal variables, coded variables.	risk factor, Race

What is a Logistic Regression "model"?

The purpose of logistic regression is to produce a mathematical equation that relates the probability of an outcome to the particular value of risk factor variables. A model might predict the probability of occurrence of a myocardial infarction (MI) over a 5-year period, given a patient's age, sex, race, blood pressure, cholesterol level, and smoking status.

In Epi Info, a model can be expressed as:

MI = age + sex + race + blood pressure + cholesterol + smoking status

The actual model is

1

MI =

$1 + e^{- (\alpha + \beta1*\text{age} + \beta2*\text{sex} + \beta3*\text{race} + \beta4*\text{blood pressure} + \beta5*\text{cholesterol} + \beta6*\text{smoking status})}$

or

$\text{logit MI} = \alpha + \beta1*\text{age} + \beta2*\text{sex} + \beta3*\text{race} + \beta4*\text{blood pressure} + \beta5*\text{cholesterol} + \beta6*\text{smoking status}$

Epi Info 2000 uses a process called "maximum likelihood estimation" to arrive at the best estimate of the relationships based (usually) on a follow-up study, such as the well known Framingham study. The results include values for the beta coefficients ("β"), but more important for epidemiologists, can produce an odds ratio (OR) for each value of a risk factor compared with its baseline ("absent" or "normal") state.

How do I pick the outcome variable?

The outcome variable is a dichotomous variable (two values) for which other variables may provide an explanation. Often, the outcome variable will indicate the presence or absence of a disease. To explore the risk factors for premature birth, however, the outcome variable might be low birthweight.

The coding of this variable must be in "0/1":

"1" for persons who experienced the event studied (disease or low weight),

"0" for persons who did not (no disease or normal weight).

NOTE: Epi Info does the coding if the outcome variable is a “yes / no” variable.

How can I use explanatory categorical variables with more than 2 values?

Explanatory categorical (coded) variables with more than 2 values must be recoded with dummy variables. If not, these variables will be considered to be quantitative variables, which may make it difficult to interpret the results of the logistic regression model. A variable with "k" values must be coded in "k-1" dichotomous dummy variables that have two values, with "0" value indicating no or least exposure.

For example:

		coding
Race:	White	1
	Black	2
	Asian/Pacific Islander	3

Dichotomous dummy variables coding used for Race:

	Design	Variables
Race	R1	R2
White	0	0
Black	1	0
Asian/ Pacific Islander	0	1

Two dichotomous dummy variables are enough to locate the three initial values of the "Race" variable. In the logistic regression model, only the variables R1 (Black Race = 1) and R2 (Asian/Pacific Islander = 1) will appear, and the risk computed will be in reference to White Race (risk = 1).

NOTE: Epi Info creates dummy variables for categorical variables.

How can I use explanatory continuous variables?

If there are continuous variables in the dataset (age, or duration of a treatment, for example), it is possible to include them as continuous variables in the logistic regression model. The results are expressed as the risk for each unit of the continuous variable (each year of age or each day of a treatment, for example) and the risk for the number of units desired will have to be computed (see examples below, under "What does the model look like in Epi Info?").

Continuous variables can also be recoded into categories to make interpretation easier, using the median as a cut point to recode them in two values, or quartiles as a cut point to recode them in four values, or other meaningful categories (e.g., age = adult / child or 10 year age groups, treatment = none, partial, or full). After recoding, test each version in a model containing only outcome and the variable being recoded. Choose the coded or continuous version that gives the most significant result in the likelihood ratio test (the smallest).

NOTE: The interpretation of the results for a continuous variable is more difficult than for a categorical variable, but in recoding a continuous variable in a categorical variable, some information is lost from the data. It is never a simple choice.

What is interaction and why should it be addressed early in the model?

Interaction means that the odds ratio (OR) for a variable varies with the value of another variable. For example, if the outcome is mesothelioma, a form of lung cancer caused by asbestos exposure, the effect (OR) of asbestos exposure differs greatly between smokers and nonsmokers.

Mesothelioma = asbestos + smoking

does not tell the whole story, and another term, called an “interaction term,” is needed:

Mesothelioma = asbestos + smoking +asbestos*smoking

Interaction must be addressed early in forming a model, because the model must contain all single variables that are found in significant interaction terms to be “hierarchically well structured.” One school of thought in modeling strategy suggests that all pertinent interaction terms be evaluated before eliminating any individual variables.

After choosing the outcome variable, how do I construct a good Logistic Regression model?

“Different people have different modeling strategies. It’s important that you develop a point of view for yourself (after considering other schools of thought).” **D.G. Kleinbaum**

We give you one strategy below. The references contain other modeling strategies to consider (Hosmer, Breslow).

One modeling strategy to find risk factors for an outcome involves two stages:

1) Variable specification and interaction assessment;

2) Confounding assessment followed by consideration of precision.

1) Variable and interaction specification: Choose the variables to include in the multivariate logistic regression model.

If you have only a few variables, start with all of them. Otherwise, include variables that may be risk factors or control variables, based on literature review, and then add all variables for which the p-value of the chi square, Fisher exact, or t-test is less than 0.25 (not the usual threshold of 0.05) in the Epi Info TABLES or MEANS commands. (Note: If you have several exposure and control variables, you have to build a model with all of them and their pertinent interactions.) Now that you have an outcome variable and a list of confounder/risk variables, choose one variable as your primary effect variable for analysis. In the L.A. Study example, the outcome variable is Ill (yes/no). To examine the role of *estrogens*, choose it as the effect variable. Construct the model as follows:

Ill = estrogen + age + hypertension + age*hypertension + estrogen*age + hypertension*estrogen

NOTE: If there are a lot of exposure variables (for example, in the Oswego study there are 20 variables), build a first model with all of the variables, choose those variables with the smallest p-value (around 7) and build another model with these variables and their pertinent interactions (biologically meaningful subsets).

Use a *backward elimination approach* to find the best model. This means eliminating variables or groups of variables one at a time, keeping only those that are “meaningful” in the model. “Meaningful” in this case stands for a p-value <0.05 and the likelihood ratio test <0.05 for the model containing this “chunk” versus the one from which it has been removed.

Frequently there is “interaction” among variables so that the OR for one depends on the value of another (e.g., the effect of age or smoking might vary in different races).

If you have two or more categorical explanatory variables in the best model, you must systematically test the interaction term in the logistic regression model (see previous question). The interaction term in the logistic regression model corresponds to the mathematical product of the two variables (variable 1*variable 2). In this case, interaction variables are included in the model, as smoking*race and ageclass*race. If you have more than two categorical variables included in the model, you must test the interaction term for biologically meaningful subsets (which may be all of them, depending on the literature). If there is an interaction term, it is important information for the study: the relationship between the exposure variable and the outcome variable is not the same for all levels of the explanatory variable (the risk factor varies).

All variables in interaction terms must be included elsewhere in the model as single terms, or Epi Info will not allow the interaction term. This results in what is known as a "hierarchically well formulated" model. Once an interaction has been found to be significant, all of its smaller subcomponents must be left in subsequent models.

The simplest interaction terms are pairs of categorical variables. In setting up the model, list the single variables first. The following example is from the Oswego study:

vanilla, chocolate, age

Then include the interaction term:

age + vanilla + chocolate + vanilla*chocolate + vanilla*age + chocolate*age (interaction term)

Third-order terms should also be included, as in vanilla*chocolate*age, if pertinent.

Use the likelihood ratio ("chunk") test to assess the effect of removing the third-order interaction term at the same time. Run the full model, then remove the three-variable term and run it again. The results of the log likelihood ratio test for the "chunk" removed will appear at the bottom of the output. If the likelihood ratio p-value is <0.05, at least part of the "chunk" is significant. Now do the same for the second-order terms, removing both at one time. If the likelihood ratio test p-value is significant, then at least one of the two terms is significant, and they should be evaluated individually. Identify any two variables and single variable terms that are components of significant three-variable terms. These must stay in the model, regardless of subsequent results. They are protected members of the "hierarchically well formulated model."

More details on interaction can be found below, under " How do I interpret the results of Logistic Regression containing interaction terms?"

2) Confounding assessment

At this point you have what is called the "Gold Standard" model, because the estimated OR for the exposure variable is likely to be the most "correct." Confounding is assessed without the use of statistical testing. The procedure involves determining whether the estimated OR changes meaningfully in comparing the OR of the best model versus the model without one or more possible confounders. If there is no interaction, the assessment of confounding is carried out by monitoring changes in the OR of the explanatory variable. However, if there is interaction, the assessment of confounding is more subjective, because it requires comparison of the OR of the exposure variable with the significant interaction terms. The goal is to find a model that gives OR estimates for the exposure variable and interaction terms similar to those given by the "Gold Standard" model. If confounders can be removed without changing these ORs, and the precision (width of the confidence interval) of the OR for the exposure improves, this should be done. Care should be taken not to remove any confounders used in interaction terms.

In the Oswego study, we find:

Ill = bakedham + mashedpota + brownbread + milk + coffee + cakes + vanilla + relevant interaction

and the model becomes:

Ill = vanilla

I have seen a lot of arguments in the literature about the improper use of Regression Analysis, even among experts. What can cause wrong conclusions? How do I know that I have used it correctly?

If you didn't take into account a confounder or an interaction factor, your conclusion may be incorrect. To avoid this, use:

* Good modeling strategy (see above, " What is interaction and why should it be addressed early in the model?"):

- good choice of the variables included in the multivariate logistic regression model
- good choice of the coding of variables (very important!)
- appropriate assessment of interaction terms
- appropriate assessment of the confounder variables.

* Quality of the results:

- precision of confidence interval: CI 0,95 (Bi +/- 1.96 s Bi)
- is the value "1" inside CI 0,95 or not?
- is Wald test of the differences among variables significant?
- is the likelihood ratio of the chi square test of the model significant?
- comparison of the results with other studies
- analysis of residuals of the model.

What does the model look like in Epi Info?

There are two examples provided with Epi Info 2000:

1) Oswego study: A study about an outbreak of acute gastrointestinal illness.

We use this example to illustrate the fact that we don't need logistic regression for this study.

On April 19, 1940, the local health officer in the village of Lycoming, Oswego County, New York, reported the occurrence of an outbreak of acute gastrointestinal illness to the district Health Officer in Syracuse. Seventy-five persons ate at a particular supper, and 46 persons with gastrointestinal illness were identified. The goal for the study was to find which food or foods caused the outbreak.

The outcome variable is *Ill* (yes/no).

Possible risk factors (predictor variables) are *foods and drinks consumed*.

This file is available in SAMPLE.MDB as viewOSWEGO.

2) L.A. Study: Los Angeles retirement community study of endometrial cancer.

We use this example to illustrate the methods for matched data analysis in the study of the effect of exogenous estrogens on the risk of endometrial cancer reported by Mack et al. (1976). These investigators identified 63 cases of endometrial cancer in a retirement community near Los Angeles. Each case was matched to 4 control women who were alive and had lived in the community one year at the time the case was diagnosed, and who had entered the community at approximately the same time. In addition, controls were chosen from among women who had not had a hysterectomy prior to the time the case was diagnosed, and who were therefore still at risk for the disease. Information on the history of use of several specific types of medicines, including estrogens, antihypertensives, sedatives, and tranquilizers, was abstracted from the medical record of each case and control. Other abstracted data relate to pregnancy history, mention of certain diseases, and obesity.

The analysis of these data is aimed at studying the risk associated with the use of estrogens, as well as with a history of gallbladder disease, and how these risks may be modified by other explanatory variables.

Variables in the file include:

The outcome variable, *Ill* (yes/no)

Possible risk factors (predictor variables): *age, hypertension, gallbladder disease, other drugs* (non-estrogen), *estrogens* (any) *conjugated estrogen: amount* (mg/day), or *conjugated estrogen:*

duration (months).

After I have a model, what can I do with it?

The use of a model depends on the design of the epidemiologic study. When using follow-up data, you can estimate a predicted risk (P), then calculate a relative risk (RR). In using case-control or cross-sectional data, however, a limitation is that you cannot estimate individual risks even though you can still obtain an OR. This limitation is not severe if the goal of the study is to obtain a valid estimate of an exposure-disease association in terms of an OR. As in univariate analysis, the OR becomes a reasonable estimate of the RR if the disease is “rare.”

Take, for example, a follow-up study:

The fitted model that we obtained by using the method of the maximum likelihood to estimate the parameters of the variables is:

$$\text{Logit}(D) = -3.911 + 0.652 * \text{cat} + 0.029 * \text{age} + 0.342\ \text{ecg}$$

where :

- D, the disease, is coded "0" if no disease
"1" if disease
- cat stands for catecholamine level, coded "0" if low level
"1" if high level
- age is a continuous variable
- ecg denotes electrocardiogram status, coded "0" if normal
"1" if abnormal.

The predicted risk of this model is:

$$P(D=1/\text{cat, age, ecg}) = \frac{1}{1+e^{-(\text{logit}(D))}} = P(X)$$

Suppose we want to use our fitted model, to obtain the predicted risk for a certain individual, for example:

cat = 1; age=40 and ecg=0.

$$P1(X) = \frac{1}{1+e^{-(-3.911+0.652(1)+0.029(40)+0.342(0))}} = \frac{1}{1+e^{-(-2.101)}} = \frac{1}{1+8.173} = 0.109$$

For a person with cat=1, age=40, and ecg=0, the predicted risk obtained from the fitted model is 0.109. That is, this person's estimated risk is about 11% over the time period studied.

cat = 0; age=40 and ecg=0.

$$P0(X) = \frac{1}{1+e{-}^{(-3.911+0.652(0)+0.029(40)+0.342(0))}} = 0.060$$

For a person with cat=0, age=40, and ecg=0, the predicted risk obtained from the fitted model is 0.109. That is, this person's estimated risk is about 6%.

$$\frac{P1(X)}{P0(X)} = \frac{0.109}{0.060} = 1.82 = RR$$

Here, for the same fitted model, we compare the predicted risk of a person with cat=1, age=40, and ecg=0 with that of a person with cat=0, age=40, and ecg=0. If we divide the predicted risk of the person with high catecholamine with the predicted risk of the person with low catecholamine, we get an RR estimate of 1.82. Thus, using the fitted model, we find that the person with high cat has almost twice the risk of the person with low cat, assuming both persons are of age 40 and have no previous ecg abnormality.

Two conditions must be specified to estimate RR directly. First, we must have a follow-up study; second, for two individuals being compared, we must specify values for all the independent variables in our fitted model to compute risk estimates for each individual.

If either of the above conditions is not satisfied, then we cannot estimate RR directly. We can only compute an OR that is a measure of the association directly estimated from a logistic regression model (without requiring special assumptions), regardless of whether the study design is follow-up, case-control, or cross-sectional. As in univariate analysis of a case-control study, the OR is a reasonable estimate of RR only if the disease is rare.

$$ORx1/x0 = e^{\Sigma\ Bi(x1i-x0i)}$$

So, for the same example:

X1= (cat=1; age=40; ecg=0)
X0= (cat=0; age=40; ecg=0)

$OR = e^{B1(1-0)+B2(40-40)+B3(0-0)}$

$OR = e^{B1+0+0}$

$OR = e^{B1}$ = 1.91 = adjusted OR

Using the fitted model, we find that the person with high cat has almost twice the risk of the person with low cat, OR adjusted on the age and the ecg.

For each dichotomous variable, it is possible to compute an adjusted OR.

If X_i is coding in (0,1): $OR = e^{Bi}$
If X_i is coding in (a,b): $OR = e^{\Sigma(Xai-Xbi)Bi}$

For the continuous variable, it is the same formula but you must multiply your Bi with a coefficient (e.g., for the age you can compute an adjusted OR).

If we compare the effect for a person of 40 years of age with that for a person of 35 years of age, then adjusted $OR = e^{(40-35)*Bage} = 1.15$.

So, a person of 40 years of age will have 1.15 times more risk of having the disease than a person of 35 years of age, OR adjusted on cat and ecg.

How do I interpret the results of Logistic Regression containing interaction terms?

If there is an interaction term (say E*X in a logistic regression model), then the risk factor E for outcome M differs with the values of the explanatory dichotomous variable X.

In mathematical expressions:

$$\text{Logit } P = \alpha + B1*E + B2*X + \underbrace{\gamma E*X}_{\text{interaction term}}$$

if X=0, then

$\ln OR = \alpha + B1 - \alpha = B1$

and if X=1, then
$\ln OR = \alpha + B1 + B2 + \gamma - (\alpha + B2) = B1 + \gamma$

There is an interaction between E and X if γ is different than "0" (Wald test (Z-Statistic) or the likelihood ratio chi square test is significant on the threshold of 0.05).

For example:

$$\text{Logit } P = -0.39 + 0.06 \text{ tobacco} + 0.19 \text{ inhale} + \underbrace{0.82 \text{ tobacco*inhale}}_{\text{interaction term}}$$

Coding of tobacco: no tobacco "0"
tobacco "1"

Coding of inhale: don't inhale "0"
inhale "1"

There are two ORs: one for people who inhale the tobacco (inhale = 1) and one for people who don't inhale the tobacco (inhale = 0). Two ORs are computed.

$OR/_{\text{people who don't inhale tobacco}} = e^{0.066} = 2.43$

$OR/_{\text{people who inhale tobacco}} = e^{(0.066+0.82)} = 1.07$

The people who inhale tobacco have 2.43 more risk to be ill than the people who don't smoke.

The people who don't inhale tobacco have 1.07 more risk of being ill than the people who don't smoke.

How should I present the results of Logistic Regression in a slide or an article? (Example, please.)

Table 1: Risk factors for endometrial cancer.

Variables	βi	Se βi	$OR = e^{\beta i}$	Confidence Interval ($CI_{0.95}$)	p-value

Gallbladder disease (dichotomous variable)	1.150	0.446	3.158	[1.316 ; 7.578]	0.010
Estrogen (dichotomous variable)	1.718	0.517	5.576	[2.021 ; 15.383]	< 0.001
Duration of estrogen treatment (Continuous variable)	0.011	0.005	1.012	[1.002 ; 1.022]	0.017

Likelihood ratio chi square test: 46.58 with 3 degrees of freedom ($p < 0.0001$).

NOTE: For a less technical audience, the beta coefficient (βi) and the standard error of the beta coefficient (Se βi) can be omitted.

References

Breslow NE, Days, NE. Statistical methods in cancer research, Vol. 1: The analysis of case-control studies. Lyon, France: IARC Scientific Publication No. 32; 1981.

Hosmer DW, Lemeshow S. Applied logistic regression (Wiley Series in Probability and Mathematical Statistics. Applied Probability and Statistics Section). New York: John Wiley; 1989.

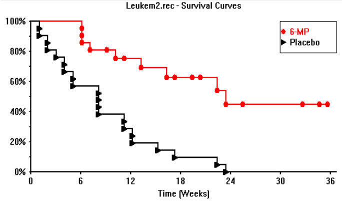
Kleinbaum DG, Kupper LL, Morgenstern H. Epidemiologic research: Principles and quantitative methods. New York: Van Nostrand Reinhold; 1982.

Kleinbaum DG, Kupper LL, Muller KA. Applied regression analysis and other multivariate methods. Second edition. Boston: Duxbury Press; 1987.

Kleinbaum DG. Statistics in the health sciences: Logistic regression. New York: Springer-Verlag; 1994.

Mack TM, Pike MC, Henderson BE, Pfeffer RI, Gerkins VR, Arthur M, Brown SE. Estrogens and endometrial cancer in a retirement community. N Engl J Med 1976 Jun 3;294(23):1262-7.

Kaplan-Meier Survival Analysis



The KMWin Program

Authors: Cecile Delhumeau, Andrew Dean

What is the objective of the Kaplan-Meier method?

In clinical epidemiology, particularly in the study of usually fatal chronic diseases, the measurement of patient survival has become an important criterion in evaluating the effectiveness of therapeutic modalities.

The objective of the Kaplan-Meier (KM) methodology is to estimate the probability of survival of a defined group at a designated time interval (conditional probability). KM uses a non-parametric survival function for a group of patients (in other words their survival probability after the time *t*) and therefore does not make assumptions about the survival distribution.

What is KM good for?

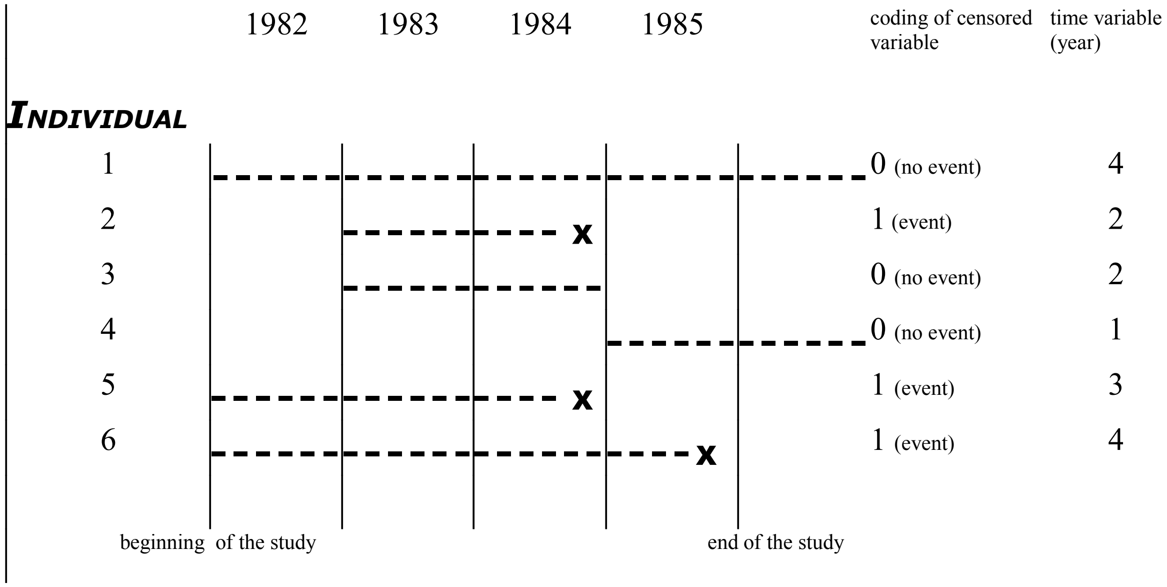
Each time a survival study is done (e.g., a follow-up study, a clinical trial, or a study of the occurrence of an event over time), the KM methodology could be used to estimate the probability of survival over a given time period. “Survival” means that the event of interest has not occurred. The event can be death, a complication of a treatment, or other defined adverse event. KM therefore provides an estimate of being free of the event at time *t*. Conversely, “1 *minus* the probability of being free of the event at time *t*” is the probability of having the event at time *t*.

What is the variable to be studied?

The variable to be studied is the time delay until the occurrence of an event (death, disease, treatment outcome, etc.). This time delay corresponds to survival duration (the difference between the beginning study date and the event date).

What distinguishes survival analysis from most other statistical methods is the presence of “censoring” for the incomplete observation. In a study of survival following two different treatment regimens, for example, analysis of the trial typically occurs well before all the patients have died. For those still alive at the time of analysis, the true survival time is known only to be greater than the time observed to date. Such an observation is said to be “censored.” There are two other sorts of incomplete observation: the “lost to follow-up” (patient missing during the study duration) or the appearance of an event other than the event being studied. These observations are also considered censored. For survival analysis, the censored variable, the time variable, the units of time (day, month, year), and the group of patients (if studying the effect of a treatment) must be specified. The time variable is numeric. The censored variable is coded: “1” if the patient experiences the event (uncensored data), “0” if the event is not known to occur (censored data). Survival data is often presented using a “+” for the censored observation, so that a set of times might be 8, 11+, 14, 2, 36+, etc.

Table 1: Coding of censored variable for 6 patients with bladder cancer



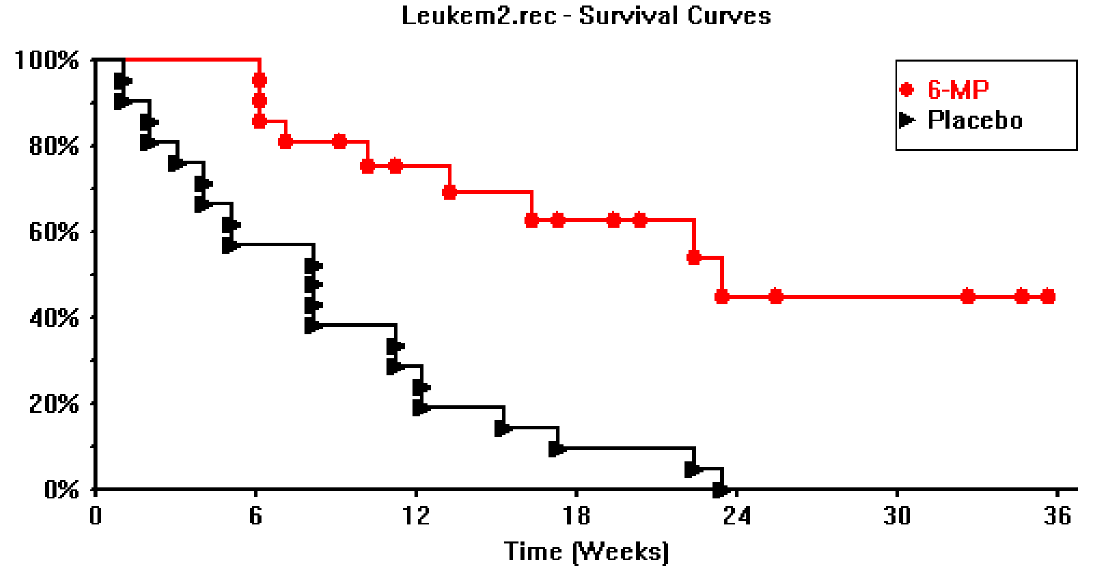
INDIVIDUAL	coding of censored variable	time variable (year)
1	0 (no event)	4
2	1 (event)	2
3	0 (no event)	2
4	0 (no event)	1
5	1 (event)	3
6	1 (event)	4

The censored variable will code “0” for individuals 1, 3, and 4, and “1” for individuals 2, 5, and 6.

What is the shape of the KM survival function?

The KM survival function is a decreasing series of straight line steps, constant between two consecutive death times, with a step for each time of observed death. This function is not defined after the last observation if this observation is censored.

Here is an example of the shape of the KM survival function:



How can I interpret the result of the KM curve?

If the survival rate for a group of patients 5 years after the beginning date of the study is 0.8, this means these patients have, on the average, 80 chances out of 100 of remaining alive after 5 years. In this example, the occurrence of an event is "death," but we could also describe other events, for example, the occurrence of complications during medical treatment.

How does Epi Info 2000 compute the KM estimator?

In the first step Epi Info sorts the records by time, ti, then for each time interval t_i up to t_{i+1} (but not including) it counts:

* the number of deceased patients d_i at time t_i,

* the number of censored patients c_i at time t_i,

* the number of risk patient n_i (number of patients living just before t_i)

$n_i = n_{i-1} - c_{i-1} - d_{i-1}$

After all the steps, it is possible to compute the KM estimator:

$S = \Pi \ (1-d_i/ n_i)$ ("Π" means "product of")

The example provided in this documentation uses the Leukemia study (Freireich, 1963**),** a study about the remission time delay for patients with leukemia who are given different treatments. This example is based on a randomized clinical trial to evaluate if patients assigned to treatment with 6-mercaptopurine (6-MP) would fare better than untreated (Placebo) patients.

The project file is provided with Epi Info 2000 as leukem2.kmp

Key variables:

* time delay (in weeks)

* censored variable (“0” for censored individual, “1” for uncensored individual)

* group of patients (6-MP or Placebo)

KM estimator for the group 6-MP

Time in Weeks	S(t)	Standard Error	Cumulative Failures	At Risk
6.00	0.95238	0.04647	1	20
6.00	0.90476	0.06406	2	19
6.00	0.85714	0.07636	3	18
6.00+			3	17
7.00	0.80672	0.08694	4	16
9.00+			4	15
10.00	0.75294	0.09635	5	14
10.00+			5	13
11.00+			5	12
13.00	0.69020	0.10681	6	11
16.00	0.62745	0.11405	7	10
17.00+			7	9
19.00+			7	8
20.00+			7	7
22.00	0.53782	0.12823	8	6
23.00	0.44818	0.13459	9	5

25.00+			9	4
32.00+			9	3
32.00+			9	2
34.00+			9	1
35.00+			9	0

+ denotes censored patient (c_i)

Where:

Time is the day of observation

S(t) is the survivor function => Pr (T>t)

Standard Error is the standard error of S(t)

Cumulative Failure is the number of deceased patients (d_i)

At Risk is the number of at-risk patients (n_i)

The patients in the 6-MP group have, on the average, 44.8 chances out of 100 of remaining alive after 23 weeks.

Quartile	Estimate
75th	13.00
50th	23.00

75 % from the sample live on the average 13 weeks

50 % from the sample live on the average 23 weeks

Mean Survival Time (MST) in Days

```
Method        MST     Limited    Standard
                        To         Error

Last Obs.     23.287   35.00      2.9990
Cutpoint      16.117   20.00      1.3518
Comparison    17.909   23.00      1.6474
```

KM estimator for group Placebo :

Time Weeks	S(t)	Standard Error	Cumulative Failures	At Risk
1.00	0.95238	0.04647	1	20
1.00	0.90476	0.06406	2	19
2.00	0.85714	0.07636	3	18
2.00	0.80952	0.08569	4	17
3.00	0.76190	0.09294	5	16
4.00	0.71429	0.09858	6	15
4.00	0.66667	0.10287	7	14
5.00	0.61905	0.10597	8	13
5.00	0.57143	0.10799	9	12
8.00	0.52381	0.10899	10	11
8.00	0.47619	0.10899	11	10
8.00	0.42857	0.10799	12	9
8.00	0.38095	0.10597	13	8
11.00	0.33333	0.10287	14	7
11.00	0.28571	0.09858	15	6
12.00	0.23810	0.09294	16	5
12.00	0.19048	0.08569	17	4
15.00	0.14286	0.07636	18	3
17.00	0.09524	0.06406	19	2
22.00	0.04762	0.04647	20	1
23.00	0	0	21	0

+ denotes censored patient (c_i)

Where :

Time is the week of observation

S(t) is the survivor function => Pr (T>t)

Standard Error is the standard error of S(t)

Cumulative Failure is the number of deceased patients (d_i)

At Risk is the number of risk patient (n_i)

The patients in the Placebo group have, on the average, 0 chances out of 100 of remaining alive after 23 weeks.

Quartile	Estimate
75th	4.00
50th	8.00
25th	12.00

75 % from the sample live on the average 4 weeks

50 % from the sample live on the average 8 weeks

25 % from the sample live on the average 12 weeks

Mean Survival Time (MST) in Days

Method	MST	Limited To	Standard Error
Last Obs.	8.667	23.00	1.4114
Cutpoint	8.429	20.00	1.3896
Comparison	8.667	23.00	1.4114

Summary Table for the Data:

Group	Total	Fail	Cens.	%Fail	%Cens.	FDR(*)	Lower	Upper
6-MP	21	9	12	42.86	57.14	0.0251	0.0153	3.7213
Placebo	21	21	0	100.00	0.00	0.1154	0.0250	3.9093
	-----	-----	-----	-----	-----	-----	-----	-----
TOTAL	42	30	12	71.43	28.57	0.0555	0.0183	3.7851

(*) Failure Density Rate (#Failures / Person * Time).

Where :

Total is the sample size

Fail is the number of deceased patients

Cens is the number of censored patients

%Fail is the percentage of deceased patients

%Cens is the percentage of censored patients

FDR is the Failure Density Rate (#Failures / Person * Time).

Lower and Upper are the limits of the 95 % Confidence Interval of FDR

How should I present the results of KM in a slide or an article?

The presentation of results for a KM analysis might be a graph like the one shown previously for each group of patients with 95% confidence interval at each time *t* (precision of the survival estimation) and a table with the survival estimates at 1, 2, and 5 years (all depend on the disease being studied).

How should I evaluate whether or not KM curves for two or more groups are statistically equivalent?

When we state that two KM curves are "statistically equivalent," we mean that, based on a testing procedure that compares the two curves in "some overall sense," we do not have evidence to indicate that the true (population) survival curves are different.

The software uses the log-rank test, a large-sample chi square test that uses as its test criterion an overall comparison of the KM curves being compared. This (log-rank) statistic, like many other statistics used in other kinds of chi square tests, makes use of observed versus expected cell counts over categories of outcomes. The software also uses the generalized Wilcoxon test, a non-parametric test.

This might be used, for example, to test for a significant difference in treated and untreated patients. This test assumes that the death rate is constant over time (the software does a logarithm transformation of the KM survival curve).

A p-value of the log-rank test <0.05 suggests a difference in survival between the two groups.

Example :

Summary Table for the Data:

```
Group     Total  Fail  Cens.  %Fail  %Cens.  FDR(*)  Lower   Upper

6-MP        21     9    12    42.86   57.14   0.0251  0.0153  3.7213
Placebo     21    21     0   100.00    0.00   0.1154  0.0250  3.9093
          -----  -----  -----  -----   -----   -----   -----   -----
TOTAL       42    30    12    71.43   28.57   0.0555  0.0183  3.7851
```

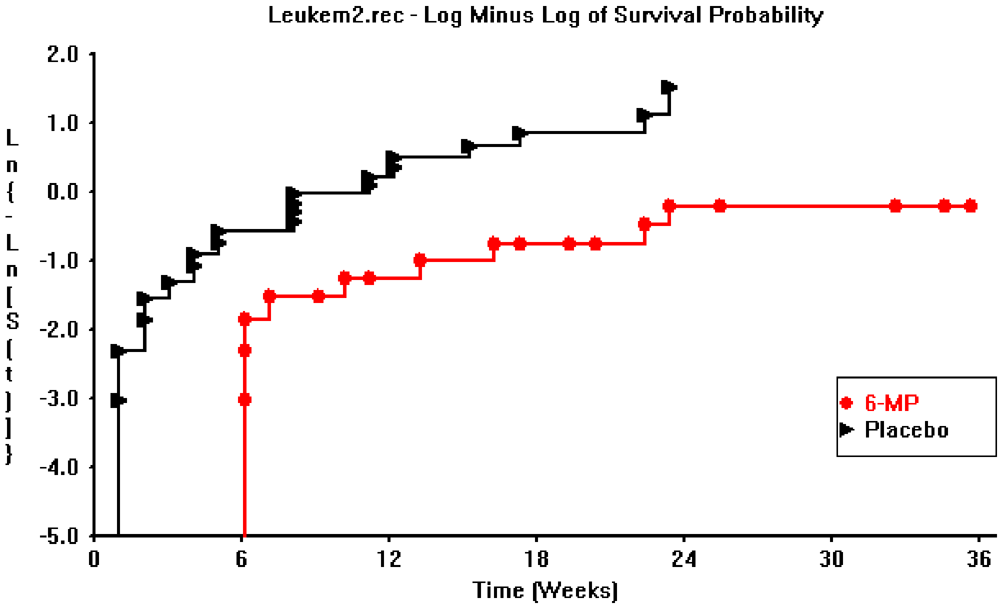
(*) Failure Density Rate (#Failures / Person * Time).

Test Statistics

Test	Statistic	DF	p-Value
Cox-Mantel (Log-Rank)	16.7929	1	0.00004
Generalized Wilcoxon (Breslow)	13.4579	1	0.00024

The two p-values are <0.05. We conclude that patients with the treatment 6-MP live longer than patients with the placebo treatment.

Testing the assumption that death rate is constant over time in this example:



References

Freireich EJ, Gehan E, Frei E. The effect of 6-mercaptopurine on the duration of steroid-included remissions in acute leukemia: a model for evaluation of the other potentially useful therapy. Blood 1963; 21: 699-716.

Kleinbaum DG. Statistics in the health sciences: Survival analysis. New York: Springer-Verlag; 1996.

Epi Info 2000
- How to Install
- Download
- Online Manual
- Tutorials
- FAQs
- Link 4
- Link 4
- Link 4

Epi Info 6

History

Translations

Resources for Creating Public Health Maps

Global Links

Contact Us

Resources for Creating Public Health Maps

Overviews	Global Public Health GIS Projects
Geographic Boundaries	Conferences and Symposia
Data	Software
Geostatistics and Spatial Analysis	Consultants

OVERVIEWS

- Information Architecture White Paper

  IA-5A10: Geographical Information Systems (GIS), Los Alamos National Laboratory, July 1997.

- Improving Public Health Through Geographical Information Systems An Instructional Guide to Major Concepts and Their Implementation, Web Version 1.0, December 1997. By Gerard Rushton at the University of Iowa.
- Public health geographic information systems (GIS) news and information: 1994-1997, by Charles Croner (January 1998). NCHS Cognitive Methods Working Paper Series Number 23. Can be ordered online.

GEOGRAPHIC BOUNDARIES

Use these resources to locate *.shp boundary files for use with any ESRI-based GIS software, including Epi Map 2000.

Geospatial Link Repositories
Principal Sites:

- ESRI ArcData Online: GIS Data on the Web
- ESRI Packaged Geographic Data Sets -- Global
- ESRI U.S. State Plane Zones Data Set
- Federal Geographic Data Committee
- Geospatial Data Clearinghouse Entry Points

- GIS Jump Station – Domestic Federal Sites
- GIS Jump Station – International Gov’t Sites
- GIS Resources for the Federal Government
- National Geodetic Survey Products and Services
- National Imagery and Mapping Agency (NIMA)
- US Census Bureau TIGER Mapping Service
- US Geological Survey National Mapping Information

See Also:

- Alexandria Digital Library, University of California, Santa Barbara
- Center for Advanced Spatial Technologies
- Epidemiology & Medical Geography Page, of Eutopia, University of Northumbria at Newcastle
- GDT's premier street network database, Dynamap/2000®
- Geographic Information Systems
- GIS-Expertisecentrum Rijksuniversiteit Groningen
- GIS Resource Centre
- Harvard Map Collection
- Health Geographic Resources
- Institut für Geographie Department of Geography Salzburg, Austria
- Just Another Medical Geography Page
- MAGIC: Map and Geographic Information Center
- NASA Langley Geographic Information Systems
- National Cartography and Geospatial Center (NCGC)
- Northwestern University Library Government Publications and Maps
- PennState University Libraries Maps and Geography
- Sandia National Laboratories GIS WWW Sites, by Topic
- Spatial Odyssey GIS Literature Database
- Terra Data Links to Mapping Related Sites
- Universiteit Utrecht Faculty of Geographical Sciences GLOBIS: Nice Geography Sites
- Universiteit Utrecht Oddens's Bookmarks: The Fascinating World of Maps and Mapping
- University of Edinburgh Department of Geography
- University of Virginia geostat: Geospatial and Statistical Data Center
- US Fish & Wildlife Service Region 5 GIS Lab Home Page
- US Geological Survey GTOPO30 Global 30 Arc Second Data Set
- US Geological Survey Map-It: Form-based Simple Map Generator

Geospatial – USA State GIS Data Centers

- Arkansas Interactive Mapper, Center for Advanced Spatial Technologies
- Colorado ELBERT Geographic Information Coordinating Committee
- Connecticut Dept of Environmental Protection DEP Store: The Maps and Publications Office of the Natural Resources Center
- Georgia GIS Data Clearinghouse
- Massachusetts Geographic Information System: MassGIS
- Maine Office of GIS
- New Hampshire GRANITNet
- New Jersey DEP's GIS web page
- New Mexico GIS Directory and Jump Station
- Oregon State Service Center for Geographic Information Systems
- Rhode Island Geographic Information System
- Texas Natural Resources Information System
- Vermont Geographic Information System
- Virginia TIGER/Line Data Browser

GPS

- Global Positioning System Resources – University of Iowa
- Trimble - All About GPS
- Trimble Community Base Stations Index
- US Coast Guard Maritime Differential GPS Service
- USDA Forest Service GPS Page
- Washington State GPS Reference Base Station Information

Images

- Aerial Photos, USDA Farm Service Agency
- Microsoft TerraServer World Imagery Database
- SPIN-2 Image Library: Worldwide Coverage

DATA

These sites provide numerators and denominators for data to be mapped in your public health projects.

Demographic

- Columbia University Center for International Earth Science Information Network (CIESIN) Socioeconomic and Applications Data Center (SEDAC)
- EUROSTAT: Statistical Office of the European Communities
- U.S. Census Bureau

- U.S. Census State Data Centers
- U.S. Census Bureau International Programs Center

Disease Surveillance

- EuroSurveillance: European Communicable Disease Bulletin
- Public Health Laboratory Service – United Kingdom
- US Centers for Disease Control and Prevention
- US Department of Health and Human Services

Environmental Data

- BioNet Environmental Information System
- Catalina Island Conservancy Intranet Ecological Management Programs GIS Submenu
- Center for Marine Conservation
- Charles Sturt University (Australia) GIS
- Compass Informatics, Irish Environmental Protection Agency's National Freshwater Quality Database
- Coral Reef Ecology and Geographic Information Science
- Environment Australia Online
- Environmental Measurements Laboratory Databases
- Florida Marine Research Institute
- Florida Statewide Ocean Resource Inventory
- Geospatial Technology Activities at the Patuxent Wildlife Research Center
- Groundwater.Com
- Holt's GAP analysis and GIS links
- ICE MAPS Interactive California Environmental Management, Assessment, and Planning System
- InfoRAIN: Bioregional Information System for the North American Rainforest Coast
- National Center for Ecological Analysis and Synthesis
- Pacific Northwest National Laboratory Geographic Information Systems
- RAY MINE PILOT STUDY: Advanced Monitoring of Hazardous Waste Sites: Discrimination and Screening of Problem Mine and Extractive Industry Wastes
- UCSD Scripps Institution of Oceanography Geoscience Information Center
- University of Miami Center for Marine and Environmental Analysis
- University of Miami Geocore GIS Research Facility
- US Environmental Protection Agency
- US Fish & Wildlife Service Environmental Contaminants Program
- USGS National Gap Analysis Program: A Geographic Approach to Planning for Biological Diversity

- US National Oceanic and Atmospheric Administration Coastal Services Center

GEOSTATISTICS AND SPATIAL ANALYSIS

- AI – GEOSTATS
- An Introduction to Geostatistics, Macquarie University
- Center for Spatial Analysis Technologies (Georgia Tech and USGS)
- Geographical Information Science Tutorials in the World Wide Web, University of Hong Kong
- Geovariances: The Universe of Geostatistics

GLOBAL PUBLIC HEALTH GIS PROJECTS

Use these resources to locate examples of use of GIS in public health projects. You may be inspired to create similar such maps with your own data.

Online Atlases

- An Atlas of Injury Death in Australia 1990-1992
- Anatomy of an Epidemic. An elementary example of student work.
- ArcData Online: GIS Data on the Web
- The Atlas of Mortality in Europe. A WHO/UN/Statistics Netherlands/CIVM collaboration with interactive maps by year, gender, and ICD9 code.
- Atlas of United States Mortality. A CDC NCHS Website with downloadable Acrobat (*.pdf) files with various causes of death. Also, some maps available for online viewing through your browser.
- Demographic Data Viewer, CIESIN/SEDAC
- Mortality Atlas of Cancer and Other Causes, Spain 1975-1986
- NASA Ocean Color Data and Resources: Health Applications of CZCS Data
- The Sentinel System, INSERM, French Ministry of Health.
- US Environmental Protection Agency Envirofacts Warehouse Maps on Demand
- WHO Health for All Database Queries for Europe and NIS
- WHO Division of Control of Tropical Diseases

CONFERENCES AND SYMPOSIA

- International Health Geographics Conference 2000, Washington, D.C., March 17 - 19, 2000.
- First International Health Geographics Conference, Baltimore, October 15-18, 1998.

- 1998 GIS in Public Health Conference, San Diego, August 18-20, 1998.
- GIS for Health and the Environment, Proceedings of an International Workshop, Colombo, Sri Lanka, September 5-10, 1994.

SOFTWARE

GIS software packages that you can either purchase or download for free.

- CaveTools for ArcView GIS
- Claritas
- Community 2020 HUD Community Planning Software
- DismapWin freeware developed by Dr. Peter Schlattmann at the Institut für Soziale Medizin und Medizinische Psychologie
- EpiMap, freeware developed by CDC and WHO.
- ESRI
- GAEA: Geoscience and Engineering Applications
- Healthmapper is the software of HealthMap, a joint WHO/UNICEF Programme.
- IDRISI, produced and supported by the Department of Geography at Clark University.
- Lakes Environmental Software: Air Modeling & Risk Assessment
- MapInfo
- Maplinx Professional
- Map Maker (Scotland, UK)
- PopMap, freeware developed by the University of Hanoi for the United Nations Statistics Division.

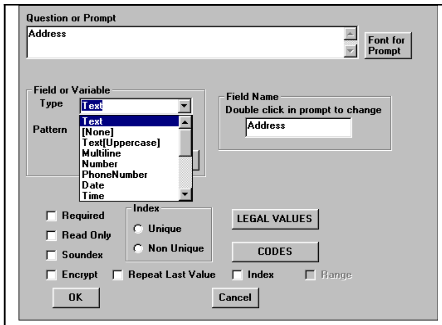
GIS CONSULTANTS
WITH EXPERIENCE IN PUBLIC HEALTH

CDC does not endorse or recommend specific vendors. Commercial firms experienced with public health GIS projects for clients such as U.S. state health departments are invited to submit links for placement in this section.

- Applied Geographics, Inc.
- Scientific Technologies Corporation
- Silent Spring Institute

This page last reviewed October 28, 1999.

Centers for Disease Control and Prevention
Epidemiology Program Office
Division of Public Health Surveillance and Informatics



Question or Prompt

Address

Field or Variable

Type: Text

Pattern

Field Name

Double click in prompt to change

Address

Index

○ Unique

○ Non Unique

□ Required

□ Read Only

□ Soundex

□ Encrypt

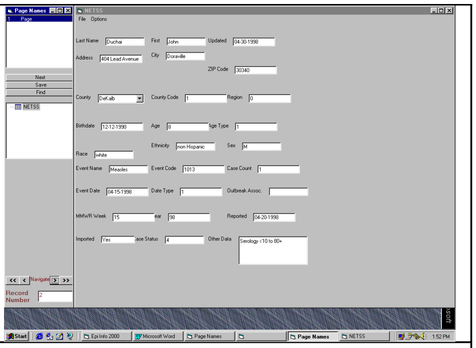
□ Repeat Last Value

□ Index

□ Range

Make a Questionnaire View

Enter Data



Page Names

1 Page

NETSS

Last Name: Duchai

First: John

Updated: 04-30-1998

Address: 404 Lead Avenue

City: Doraville

ZIP Code: 30340

County: DeKalb

County Code: 1

Region: 0

Birthdate: 12-12-1990

Age: 8

Age Type: 1

Ethnicity: non Hispanic

Sex: M

Race: white

Event Name: Measles

Event Code: 1013

Case Count: 1

Event Date: 04-15-1998

Date Type: 1

Outbreak Assoc.:

MMWR Week: 15

ear: 98

Reported: 04-20-1998

Imported: Yes

ase Status: 4

Other Data: Serology <10 to 80+

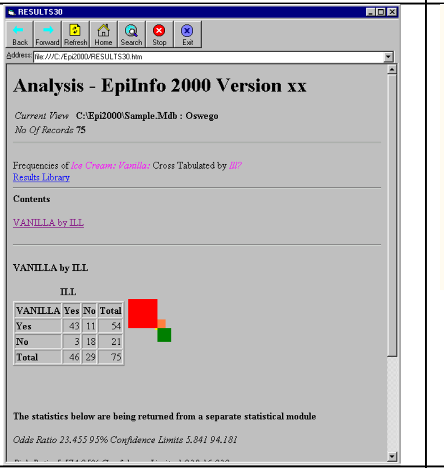
Record Number: 2

Do Epidemiologic Analysis

Person

Place

Time



RESULTS30

Address: file:///C:/Epi2000/RESULTS30.htm

Analysis - EpiInfo 2000 Version xx

Current View **C:\Epi2000\Sample.Mdb : Oswego**

No Of Records **75**

Frequencies of *Ice Cream: Vanilla:* Cross Tabulated by *Ill?*

Results Library

Contents

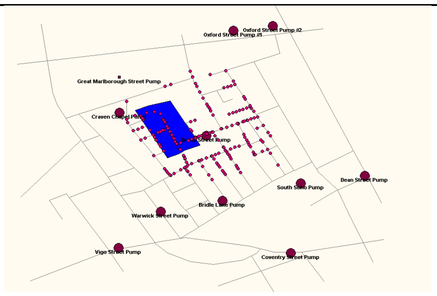
VANILLA by ILL

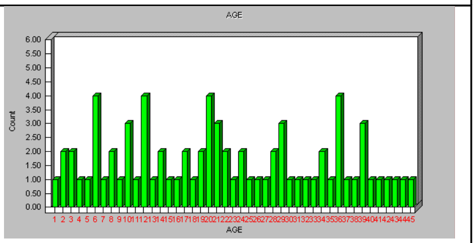
VANILLA by ILL

	ILL		
VANILLA	**Yes**	**No**	**Total**
Yes	43	11	54
No	3	18	21
Total	46	29	75

The statistics below are being returned from a separate statistical module

Odds Ratio 23.455 95% Confidence Limits 5.841 94.181





Develop a Permanent Data System

Customized Menu

Programming Data Entry

An Analysis Program

[i] CDC/WHO
[ii] NCHS
[iii] Arm