IRAP

User's Manual

Version 2.2

July 2002

Introduction

IRAP is an interactive computer program which calculates both point estimates and confidence intervals for attributable risk, based on regression models. Modeling consists of three conceptual phases: creating a library of variables, defining the model, and running the model.

IRAP keeps a "library" of variables which the user has defined. A variable can take its value directly from the data file, or it can be defined in terms of other variables in the library. Typically, the user will create a separate library for each data file that he or she uses. A library can be saved to disk, then loaded whenever the corresponding data file is used.

Once a library of variables has been created (or loaded from disk), the user may define a model. IRAP supports six data sampling methods:

- Simple Random Sampling
- Stratified Random Sampling
- Frequency Matching
- Individual Matching
- Cohort Analysis
- Cross-Sectional Analysis

After choosing the appropriate sampling method, the user defines a model by specifying the appropriate variable names. A model can contain both discrete and continuous variables. IRAP correctly parameterizes the discrete variables, so the user need not explicitly create "dummy variables."

Once the model has been specified, the user tells IRAP to fit the model. The program asks for the names of the input and output files. It then reads the data, and fits the model.

While these three phases are conceptually distinct, they can be interleaved. For example, after fitting a model, the user may decide to combine two levels of exposure into one. It is easy to define a new variable, then rerun the model, substituting the new exposure variable for the old.

Installation

IRAP requires an IBM-compatible PC with a 386 or better processor. A mouse is helpful, but not required. Although the program will run with only 2MB of memory, large models will need more memory to run. Because numerical modeling involves large amounts of floating point arithmetic, a 486DX- or Pentium-based computer is recommended. 386- and 486SX-based computers can often be fit with a numeric coprocessor chip, which will greatly improve performance on these machines.

The distribution diskette contains the following files:

irap.exe	Main program
irap.txt	On-line help file
dos4gw.exe	Rational Systems DOS/4GW DOS extender

To install the program, create a new directory on your hard disk, and copy all three files into the new directory. To run the program, double-click on the “irap.exe” icon.

Concepts

This section discusses the ideas and structures underlying IRAP: how this program organizes the world, the objects it defines, and how they are used. It does not cover the mechanics of how to perform specific tasks. If you want to know how to represent a problem using IRAP, or why IRAP is not behaving as you would expect, then read this section. If you know what you want to do, but do not know how to do it, then skip ahead to the "How-to" section.

Data Files

The data files used by IRAP are very simple: they are standard text files (i.e., "ASCII" or "flat" files, not word processor files), with one line per record. The lines are "column-delimited." That is, a variable is defined by the value occupying a given range of columns in the file. Each line is considered a separate record. Leading and trailing blanks are allowed. Thus, a small data file such as

```
00
100
101
```

contains three records. Assuming that each variable in this particular file occupies exactly one column, it contains separate variables. The variable in column 1 would be missing for the first record.

Records do not need to be sorted, and they may contain missing data. If a record contains a missing value for a variable which is actually used in the model, then it will not be used for fitting the model. In the above example, a model which uses the variable in column 1 would only use two records; models which do not use that variable will use all three. A "missing value" is anything that cannot be interpreted as a number, for example, a blank field or a letter. A field may contain leading and trailing blanks, so long as it has a real value in it somewhere. There is no upper limit on the length of a record. Most types of model require individual data. The exception is the cohort model, which takes data in the form of a person-year table.

Variables

Although the data file is laid out in columns, IRAP's models are defined in terms of named variables. IRAP maintains a list of variables that may be used in the model, called the "library." Variables can be added to or deleted from the library, and the entire library may be saved to disk for future reference. Typically, an entire file specification will be saved as a library. Then, that library will be loaded at the beginning of a session, and used to define models.

There are two types of variable: the "file variable" and the "defined variable." A file variable takes its values directly from the data file; a defined variable is a function of any number of file variables. Each variable has three attributes: its name, its number (that is, its position in the library), and optionally, a "missing code." In addition, file variables must include a starting column and field length, and may also specify an implicit decimal point's position. Defined variables replace these attributes with a definition.

IRAP's variable names are much like those in many programming and statistical languages. They may consist of letters, numbers, and underscores ('_'), but may not begin with a number. Variable names are “case insensitive.” That is, uppercase and lowercase letters are

interchangeable. Although IRAP variable names can use mixed case for readability (for example, “Age” or “GroupedWeight”), case alone does not distinguish two different variables (for example, “AGE” and “age” both refer to the same variable). In certain contexts, a variable can be specified either by name or by number. However, the number is merely the variable's current position in the library: if a variable is added or removed, then the numbers of all the variables after it will be shifted.

Both file variables and defined variables can have a "missing code." This is a value which, when encountered, is converted to an internal "missing" value, as if the field in the data file were blank. Consider a variable called "AGE," with values 0, 1, 2, and 3. Many files may contain a "9" for subjects whose age is unknown. Declaring 9 to be the "missing code" for AGE will cause those subjects to be excluded from any analysis involving the AGE variable.

The function of a missing code in defined variables is a bit more subtle. As would be expected, a defined variable is automatically considered missing if it depends on any missing variables. Using the "AGE" variable as above, the defined variable

```
AGEp1 = AGE + 1
```

will assume the values 1, 2, 3, and 4. No missing code is needed for AGEp1, because AGE's missing status (whether due to a blank or a "9" in the data file) is propagated through the definition.

Although IRAP only allows one missing code per variable, a defined variable with a missing code can be used to account for several different missing codes in the file. Consider the file variable RACE:

RACE	Meaning
1	white
2	black
3	other
4	refused to answer
5	misc. unknown

For most analyses, both 4 and 5 should be considered "missing," but IRAP only allows one missing code. A workaround is to define RACE_1, with missing code 4, as follows:

```
RACE_1 = RACE * (RACE <= 3) + 4 * (RACE > 3)
```

When RACE is 1, 2, or 3, the expression will be evaluated as

```
RACE_1 = RACE * (1) + 4 * (0)
       = RACE
```

but when RACE is 4 or 5,

```
RACE_1 = RACE * (0) + 4 * (1)
```

= 4

which is its missing code. There is nothing magic about which number is used as the missing code – it must only be outside of the variable's normal range. In the above example, 5, 6, -1 or 99 could just as easily have been used as a missing code (with the corresponding change in the definition).

File Variables

IRAP's data files are column-delimited: file variables are specified in terms of starting column and field length. Often, the decimal point is omitted from column-delimited data; because it occurs in the same place in all records, its position can be inferred. This situation is handled by the "precision" attribute of file variables. A variable with two implied decimal places (e.g., 1.23 represented as "123") has a precision of 2. An explicit decimal in the data will override the implied decimal, so "12." will always be interpreted as 12, not as 0.12 or 1.2.

Please note that leading and trailing spaces can present some odd results if values are not lined up consistently in their fields. Consider the file:

81009
81019
8 209
820 9

The variable "a" starts at column 2, has length 3, and precision 2. Leading and trailing blanks are ignored, so the records are interpreted as follows:

Data	a
100	1.00
101	1.01
<space>20	0.20
20<space>	0.20

This situation occurs rarely, if ever, because data represented with implicit decimal places are usually right-justified in the field.

Defined Variables

Like its naming rules, IRAP's variable definition language borrows from more general languages. It combines arithmetic and boolean operators, assigning precedence levels which naturally group expressions as would normally be expected. Parentheses may be used to override the default operator precedence, or simply for clarity. The operators are as follows, in order of descending precedence:

Operators	Meaning	Precedence
`( )`	Grouping	Highest

`- (unary minus)`
`*  /`
`+  -`
	Arithmetic	

`< <= > >=  ==  !=`
`&&`
`||`
	Boolean	

`@saturate(...)`	Combination	Lowest

The first row of boolean operators contains the relational operators, "less than" (<), "less or equal" (<=), "greater than" (>), "greater or equal" (>=), "equals" (==), and "not equal" (!=). Below them are the "and" (&&) and "or" (||) operators.

As IRAP only handles real numbers, the boolean operators bear some explanation Any boolean expression will evaluate to 1 if true, or 0 if false. The operands of a boolean operator may have any value; zero is interpreted as false, nonzero as true. Thus,

Expression	Result
`1 < 2`	1
`2 < 2`	0
`2 <= 2`	1
`4.98 && -3`	1
`0 && 3.14`	0
`2 || 3.14 && 0`	1 (&& is evaluated first)
`(2 || 3.14) && 0`	0 (explicit grouping causes || to be evaluated first)

The operators may be mixed freely, with the exception of the "@saturate()" operator, which may not be part of a larger expression. This operator creates a "saturation variable," which is a variable with a unique value for each observed combination of its operands' values. The syntax is:

@saturate(var1, var2, ...)

One or more variables may be listed, separated by commas. Given variables var1 (1..2) and var2 (0..1), a saturation variable will have the following values:

@saturate(var1, var2)	var1	var2
0	1	0
1	1	1
2	2	0
3	2	1

Note that the baseline level of the saturation variable represents the lowest observed value of all the component variables. The saturation operator is offered primarily as a convenient way to create saturated models. The above example could be modeled with three effects: var1, var2, and an interaction term. However, in models with more variables, it becomes more convenient to define a saturation variable than to explicitly model many interaction terms. Another use for the saturation operator involves the trivial case, "@saturate(var1)". No matter what values var1 assumes, the operator will map them to integers (0, 1, ...). This matter will be discussed more in the next section.

Models

IRAP supports six data sampling methods, based on three types of regression model. Simple and stratified random sampling, frequency matched, and cross-sectional data fit a logistic model; individual-matched data fits a conditional logistic model; and cohort data fits a Poisson model. All models except for the conditional logistic model may be fit either with or without an intercept parameter, and stratified models may include or exclude parameters representing the stratification variables.

The model makes a syntactic distinction between discrete and continuous variables. A discrete variable with n levels is automatically modeled as –1 0/1 parameters, using the variable's minimum value as the baseline. A continuous variable, marked by a "#" in the model definition, is modeled as a single parameter, whose values reflect the data literally. A continuous variable used as an exposure must take on a baseline value of 0.

Discrete variables are assumed to take on consecutive integer values; fractions are rounded down. While this rule causes no problems in most cases, variables that have non-integer values, or whose values are not consecutive integers, cause problems. Consider the variable "fat," which has the following values:

fat
1
2
4.1
4.2
4.3

If fat really is a discrete variable, then it should be modeled with four parameters: "fat=2,"

"fat=4.1," "fat=4.2," and "fat=4.3." However, IRAP rounds the highest three values to 4, so it is interpreted as follows:

fat	IRAP interpretation
1	1
2	2
4.1	4
4.2	4
4.3	4

Now, seeing that "fat" has a minimum value of 1, and a maximum value of 4, IRAP creates three model parameters: "fat=2," "fat=3," and "fat=4." Thus, the model would look something like this:

fat	fat == 2	fat == 3	fat == 4
1	0	0	0
2	1	0	0
4.1	0	0	1
4.2	0	0	1
4.3	0	0	1

Three levels of the variable have been collapsed into the "fat=4" parameter, while the "fat=3" parameter has no use at all.

The solution to this problem is to define a new variable as "satfat = @saturate(fat)." The new variable will be defined as follows:

satfat	fat
0	1
1	2
2	4.1
3	4.2
4	4.3

This variable will be modeled correctly, contributing four meaningful parameters to the model:

Variables		Parameters			
fat	satfat	satfat == 1	satfat == 2	satfat == 3	satfat == 4
1	0	0	0	0	0
2	1	1	0	0	0
4.1	2	0	1	0	0
4.2	3	0	0	1	0
4.3	4	0	0	0	1

IRAP automates the modeling of interactions in much the same way as with main effects. An interaction is expressed as the product of two or more variables, such as:

sex * age

or

sex * age * race.

An interaction can contain any combination of discrete and continuous terms. Continuous variables are marked with a number sign:

sex * #ContAge.

Discrete terms are parameterized automatically. An interaction of discrete variables results in one model parameter representing each combination of non-baseline levels of the component variables. Thus, an interaction of any number of binary variables will result in a single model parameter, which is 1 when all of the interaction terms are "exposed." Consider the following examples:

Variable	Values
age	1, 2, 3
sex	1, 2
exposed	1, 2

a) sex * exposed

Values		Model Parameters
sex	exposed	(sex == 2) * (exposed == 2)
1	1	0
1	2	0
2	1	0
2	2	1

b) age * sex

Values		Model Parameters	
age	sex	(age == 2) * (sex == 2)	(age == 3) * (sex == 2)
1	1	0	0
1	2	0	0
2	1	0	0
2	2	1	0
3	1	0	0
3	2	0	1

Continuous terms are treated literally, as demonstrated below:

Variable	Values
ContAge	10, 25, 50
ContWgt	120, 155
age	1, 2, 3

a) #ContAge * #ContWgt

Values		Model Parameters
ContAge	ContWgt	ContAge * ContWgt
10	120	1200
10	155	1550
25	120	3000
25	155	3875
50	120	6000
50	155	7750

b) #ContWgt * age

Values		Model Parameters	
ContWgt	age	ContWgt * (age == 2)	ContWgt * (age == 3)
120	1	0	0
120	2	120	0
120	3	0	120
155	1	0	0
155	2	155	0
155	3	0	155

The variables that contribute to a model are grouped by function into lists of "effects." Some lists are common to all models, while others, such as "case indicator," are specific to certain types of model. Similarly, some lists may contain many variables, while others are limited to one. The following types of model effect occur in all models:

Exposures	Each model must have one or more exposure variables. These are the variables used to determine exposure level in calculating attributable risk. For trend tests, a continuous variable used as an exposure must take on a baseline value of zero.
Confounders	Optional. These variables do not contribute to the attributable risk, but are modeled in the regression.
By-variables	Optional. Separate analyses are performed for each combination of values for these variables.
Interactions	Optional. This is a special type of effect that allows interactions to be modeled without resorting to huge lists of defined variables. An interaction can contain any combination of discrete and continuous variables, and as with main effects, discrete variables are automatically parameterized. To correctly calculate attributable risk, IRAP automatically treats any interactions involving an exposure variable as exposures as well.

In addition to these effects, each type of model also has other, required, effects, as listed below.

Simple Random Sampling, Cross-sectional:

Case indicator	A model must have exactly one case indicator. This variable must be a file variable. 0 represents a control; other (non-missing) values indicate a case.

Stratified Random Sampling, Frequency Matching:

Case indicator	A model must have exactly one case indicator. This variable must be a file variable. 0 represents a control; other (non-missing) values indicate a case.
Stratifiers	The model may be stratified by one or more discrete variables. The "Stratifiers Modeled" checkbox indicates whether these will also be included as co-variants in the regression, or merely used for stratifying the attributable risk.

1:K_i Individual Matching:

Case indicator	A model must have exactly one case indicator. This variable must be a file variable. 0 represents a control; other (non-missing) values indicate a case.
Set number	This variable tells which matched set each individual belongs to. The specific values are arbitrary, but must be the same for each member of a given matched set.

Cohort Analysis:

Event count	The number of events that occurred in the specific cell.
Person-years	The number of person-years in the specific cell.

All types of model except for "1:K_i individual matching" allow an "Include Intercept" option. By default, an intercept parameter will be included in the regression model. However, you may choose instead to use one of the model's discrete variables to span the intercept. In this case, the baseline level of one variable will also be represented by a parameter in the model. This variable is chosen automatically. The first discrete confounder, if it exists, is used. Otherwise, the first discrete exposure variable is used. However, the baseline parameter never contributes to the attributable risk estimate.

A similar option exists for modeling stratifiers. By default, stratifier variables are modeled as discrete co-variants. If you disable this option, they will not be modeled, but will still define the strata in the attributable risk analysis.

The final set of modeling options offered by IRAP adjusts how the model is fit, not the model's contents *per se*. IRAP allows you to tune three parameters: maximum iterations, convergence tolerance, and singularity threshold. The maximum iterations parameter (default, 30) adjusts how long the regression is allowed to run before the model is deemed divergent. The convergence tolerance (default, 1x10**-8) is used to test for convergence in the regression. The singularity threshold (default, 1x10**-12) tests for information matrix singularity. While these defaults are usually adequate, you may change them to suit individual needs and tastes. They are saved as part of the program configuration, so your settings will be restored every time you start the program.

Bootstrap Variance

Instead of relying solely on the analytical procedure as proposed by Benichou and Gail (1990), there has been some interest shown for estimating the variance and 95% confidence interval of an attributable risk by using the bootstrap procedure (Efron and Tibshirani, 1993).

Some justifications for the need for this capability are given as follows:

- the ability to get an estimate of the variance when observations within each matched case-control set are correlated e.g. matched case-control sets arising from familial studies.
- when the analytical procedure is unable to produce an estimate of the variance due to the instability of the parameter estimates, e.g. model 15, motivating example of Benichou (1991).

Towards this end (bootstrap variance estimation), an ad hoc procedure, in conjunction with IRAP, was devised to implement this capability. Being ad hoc, this procedure is not as automated as one would desire, and necessitates some non-trivial involvement on the investigator's part. In a nutshell, the user needs to generate the bootstrap replicated samples prior to the use of IRAP. The original data along with the user-generated bootstrap samples are then to be used as input to IRAP. The proposed approach is a three step procedure and should *only* be undertaken by investigators who are thoroughly familiar with the sampling mechanism which generated the data and are comfortable with programming or have access to programming

support. IRAP will not generate the bootstrap samples internally. This is the responsibility of the user and must be performed outside the framework of IRAP.

Step 1
Prior to the use of IRAP, determine the sampling method of your data and generate bootstrap replicated samples by adhering closely to this sampling scheme. The following sampling of controls in the case-control setting are amenable for the bootstrap procedure and are accommodated by IRAP for analysis:

- Simple random sampling of controls
- Stratified random sampling of controls
- Frequency matching of controls to cases
- Matched case-control data (conditional logistic)

In addition, data generated via a cross-sectional design is supported for analysis by IRAP and the bootstrap variance estimation procedure is also applicable.

Furthermore, IRAP also supports analysis of Cohort data. However, it is not clear how bootstrap replicated random samples are to be constructed.

Once the sampling method has been determined, bootstrap samples are to be "CONSTRUCTED" by the USER by mimicking the sampling mechanism. This "large"assembled file includes the original data as well as N bootstrap replicated samples (N to be determined by the user).

A key "variable" for this "large" file must be added to each record in the file. The variable should take on the SPECIAL VARIABLE NAME of "_boot_". The value of _boot_ should be set to 0 for each record of the original data. The value of _boot_ should equal 1 for each record of the first bootstrap replicated sample. Finally, the value of _boot_ should be N for each record of the last bootstrap replicated sample. Thus a total of N+1 samples should go into the construction of this "large" data file, the original data plus N bootstrap samples.

It is crucial that the "special" variable be named _boot_, since this is the only way for IRAP to realize that one wishes to perform a bootstrap analysis and "execute" accordingly. All records on this file must be sorted in ascending order of the _boot_ variable i.e. all records with _boot_=0 come first, followed by records with _boot_=1, so on so forth.

Note: For usual "by" analysis, AVOID using the variable name _boot_ or _BOOT_. Consider them as reserved IRAP system variable names.

Step 2
Use the "large" data file described in Step 1 above for input to IRAP. Respond to all questions and prompts issued by IRAP in the usual fashion. Furthermore, in the model building dialog part of IRAP, indicate that an analysis "by" the variable _boot_ is to be performed. Upon recognition of an analytical model which requests "by _boot_", IRAP automatically goes into it's bootstrap operating mode.

Usually when the "by" option is invoked, IRAP will output attributable risk and associated results to the listing file, for all combination values of the "by" variables. Upon entering the bootstrap mode, IRAP only outputs the results for the original sample (_boot_=0) to the listing

file and suppresses the listing of results for all the replicated bootstrap samples (_boot_=1,2,....N) from the listing file.

However, IRAP will output attributable risk estimates and partial attributable risk estimates, when appropriate, for all values of _boot_, i.e. _boot_=0,1,2,3 ....N, to a file, say the bootstrap result file, which has the same name as the listing file but with an extension of ".bot". For example, if your input data set name is CASCNT.DAT and you request that the listing file be named KASKNT.LST, the estimates of the attributable risk as well as any partial attributable risk will be written to a file with the name of KASKNT.BOT. All estimates for the original sample as well as for all the bootstrap replicated samples are included. Each record of the ".bot" file would correspond to a value of _boot_ on your input file, with the first record associated with the
original sample (_boot_=0) and the last record associated with the Nth bootstrap replicated sample (_boot_=N). A description of the data layout of the ".bot" file is contained in your listing file and a file structure diagram is given below:

boot	AR	1st partial AR	...	kth partial AR
0	a.aaa	p.ppp	...	k.kkk
1	b.bbb	q.qqq	...	l.lll
...	...	...	...	...
N	x.xxx	y.yyy	...	z.zzz

Note: Since the bootstrap result file has an extension of ".bot", this should preclude the user from using the extension of ".bot" for the name of the input file or listing file when one is invoking the bootstrap option. All ".bot" files have a total of N+1 records.

Step 3
Open and examine the listing file and the bootstrap result file (.bot) using a text editor such as WordPad or NotePad. Summarize the results by inputting the ".bot" file to some standard statistical software to get the estimate of the variance of the attributable risk as well as the 95% confidence intervals either by making the usual normality assumption or by using the empirical distribution of the bootstrap estimates of the attributable risks to determine the confidence limits. Optionally, one might also be interested in viewing the histogram of the empirical distribution(s).

For more information on bootstrap estimation of attributable risk variance, consult these sources:

Benichou, J and Gail, M (1990) Variance Calculations and Confidence
Intervals for Estimates of the Attributable Risk Based on Logistic Models.
Biometrics 46:991-1003

Efron, B and Tibshirani, R (1993) An Introduction to the Bootstrap.
Chapman and Hall, New York.

Benichou, J (1991) Methods of Adjustment for Estimating the Attributable

Risk in Case-control Studies: A Review.
Statistics in Medicine 10:1753-1773.

Instructions

This section teaches you how to perform specific tasks using IRAP. Although much of the information from the previous section is repeated here, you should look there for more general information on the overall organization of the program.

The Screen

The IRAP screen is divided into three major components: the menu bar, the variable window, and the model window. The menu bar contains five "pull-down" menus. Usually, the contents of each menu are hidden--only the title shows. However, when a menu is selected, it displays a list of commands from which to choose. These commands are used to perform all actions in IRAP.

The variable window shows the contents of the current "library" of variables. If the entire library does not fit on the screen at once, the scroll bar or the arrow keys may be used to scroll through the contents. The model window shows the current model definition. Usually, the model window is automatically adjusted so that it shows the entire model. If this option is turned off (see the "Automatic tiling" command under the Options menu), then you can scroll through its contents in the same manner.

The File Menu

New library	
Open library	[F2]
Save library	[F3]
DOS	
Exit	[Alt+X]

The File menu contains the commands that are used to control the variable library as a whole, and also general system functions (starting a DOS shell, and exiting the program) which are traditionally found here.

New Library
This command clears the entire library of variables. Because the model cannot contain variables which are not defined in the library, the "New library" command has the side-effect of destroying the model definition.

Open Library
Opening a library replaces the current library with one that was previously saved. This command presents you with the dialog box shown below.

The Files window shows the names of files in the current directory. The Directories window lists subdirectories, and all of your disk drives. You will notice that only filenames ending in ".var" are shown.

> To change the criterion for displaying filenames
1. Move to the Filename field.
2. Type the pattern for files you wish to see.
"*" matches zero or more characters
"?" matches exactly one character
"*.*" shows all files.
"a*.v" shows all files beginning with "a", with a ".v" extension.
3. Click the "OK" button, or press RETURN.

> To change disk drive or directory
1. Click on the Directories window, or press Alt+D.
2. Double-click on the drive letter or directory name of your choice. Or, use the arrow keys to select your choice, then press RETURN. Choose ".." to move to the parent directory.

OR

1. Type the name of the directory in the Filename field, followed by a backslash ("\").
2. Press RETURN.

> To load a file

1. Move to the Filename field.
2. Type the name of the file.
3. Press RETURN.

OR

1. Select the file in the Files window.
2. Click the "OK" button.

Save Library

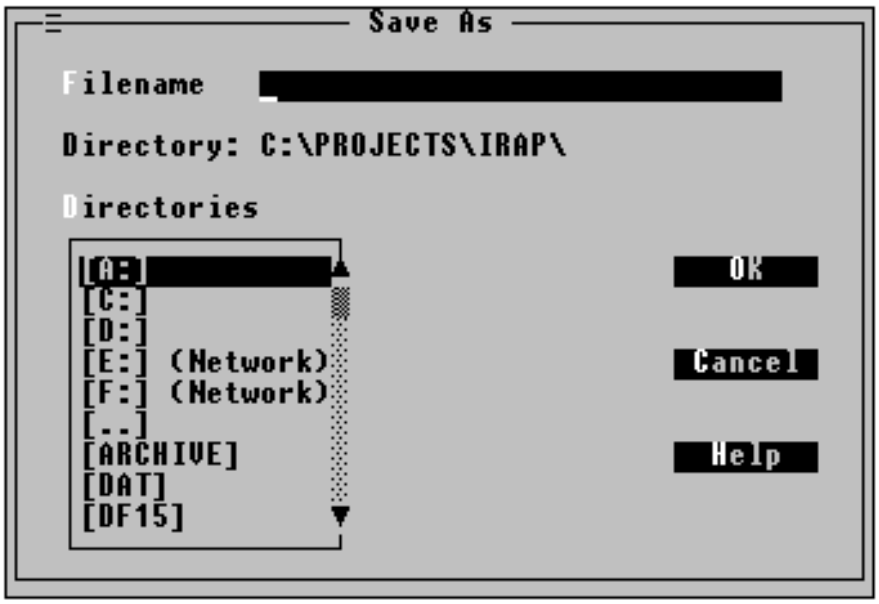
Having created a library of variables, you can save it to disk for future reference. By building a library that contains all of the variables used in a study, you ensure that the same variables are used throughout the study, and that they are named consistently.

The "Save library" command presents a dialog box similar to the "Open Library" dialog:

The "Directories" window contains a list of all the subdirectories of the current directory, and all of your disk drives.

> To change disk drive or directory

1. Click on the Directories window, or press Alt+D.



2. Double-click on the drive letter or directory name of your choice. Or, use the arrow keys to select your choice, then press RETURN. Choose ".." to move to the parent directory.

OR

1. Type the name of the directory in the Filename field, followed by a backslash ("\").
2. Press RETURN.

Once the "Directory:" field shows the proper directory, you can save your library by typing its name.

> To save the library
1. Move to the Filename field.
2. Type the name under which you want to save the library. If you did not change to the desired directory, you may explicitly specify the drive letter, directory, or both:
d:\directory\subdir\file.var
3. Click the "OK" button, or press RETURN.

DOS
When fitting several models at once, it is often convenient to check a report, or copy a data file without exiting IRAP. For this reason, IRAP offers a DOS "shell" facility. By choosing this command, you will start a command prompt while IRAP is still running. From this prompt, you can do routine chores without having to leave IRAP. Because IRAP is still running behind the shell, less memory than normal will be available, but most programs will not be hindered by this condition.

To return to IRAP, type "exit" at the DOS prompt. Do not restart IRAP with the "irap" command; doing so will start another copy of the program, not return you to your previous session.

Exit
This command terminates the program. The current model definition and library will be lost. Of course, you can save the library (see "Save library," above) for use in future sessions. At present, there is no way to save a model definition.

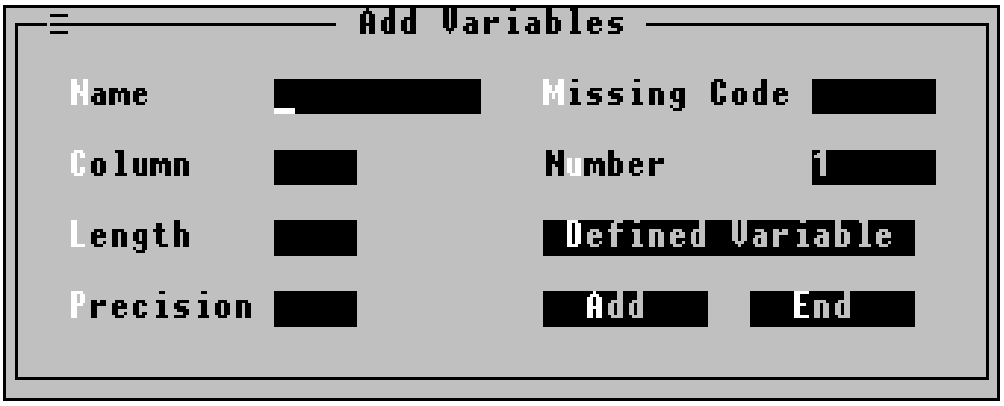
The Variables Menu

Add [Alt+A]
Modify
Delete

The Variables menu contains commands which manipulate a single variable.

Add
Choose this command to add a new variable to the library. There are two types of variable: the "file variable" and the "defined variable." In short, a file variable takes its value directly from the data. Because it is often desirable to re-code data without creating a special data file, IRAP also provides "defined variables," which are specified as functions of any number of file variables.

Initially, IRAP presents you with a dialog for adding file variables:



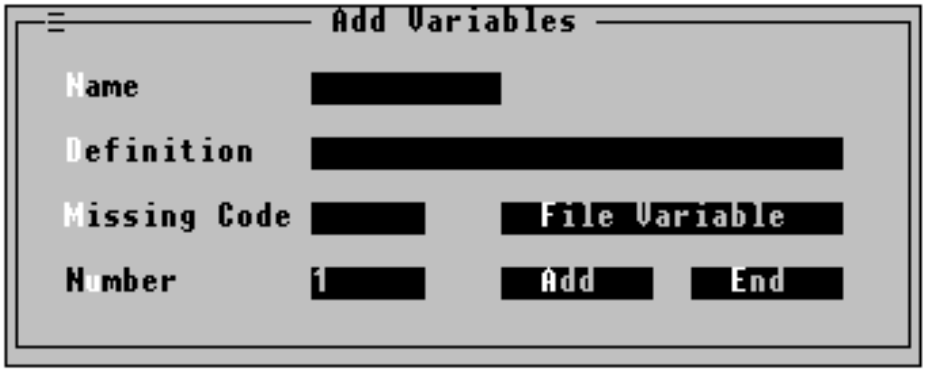
> To add a file variable

1. Fill in the Name, Column, and Length fields with the information for this variable. Use TAB or the mouse to move between fields.
2. If the field in your data contains an implicit decimal point (e.g., 1.5 represented as "15"), enter the number of decimal places in the "Precision" field (in the above example, you would enter "1").
3. If the data file contains a numeric value which should be interpreted as "missing," then enter that value in the "Missing Code" field. Some files have several codes that mean "missing." To handle this problem, see the "Concepts" section.
4. By default, the variable will be added after the last variable in the library. If you want to insert it elsewhere, enter the position in the "Number" field.
5. Click the "Add" button, or press RETURN.

At this point, the dialog box will clear itself, so that you may add another variable.

> To quit adding variables

* Click the "End" button, or press ESCAPE.



> To add a defined variable

1. From the dialog box described above, click the "Defined Variable" button, or press Alt+D.
2. Fill in the "Name" and "Definition fields. The variable definition language is described in the "Concepts" section.
3. If you wish to define a missing code for the variable, type it in the "Missing Code" field.
4. By default, the variable will be added after the last variable in the library. If you want to insert it elsewhere, enter the position in the "Number" field.
5. Click the "Add" button, or press RETURN.

Again, the dialog box will ready itself for another variable. To stop adding variables, click the "End" button, or press ESCAPE. IRAP will remember that you were adding defined variables, and return you to that dialog box the next time you add variables.

> To return to adding file variables
* Click the "File Variable" button, or press Alt+F.

Modify
Modifying a variable is essentially like adding one. You will be presented with a dialog box that is completed with the variable's current specifications. Simply make any necessary changes.

> To modify a variable
1. You will be asked which variable you want to modify. Enter the variable's name, or its number. Click the "OK" button, or press RETURN.
2. A dialog box similar to the "Add Variables" dialog will appear. Make all necessary changes to the information that is displayed.
3. Click the "Add" button, or press RETURN.
4. To cancel the operation, click the "End" button, or press ESCAPE.

Delete
Deleting a variable removes it from both the library and the model. To delete a variable, you need only give its name or number. If it was a part of the model, a message will appear indicating that the model was changed.

The Model Menu

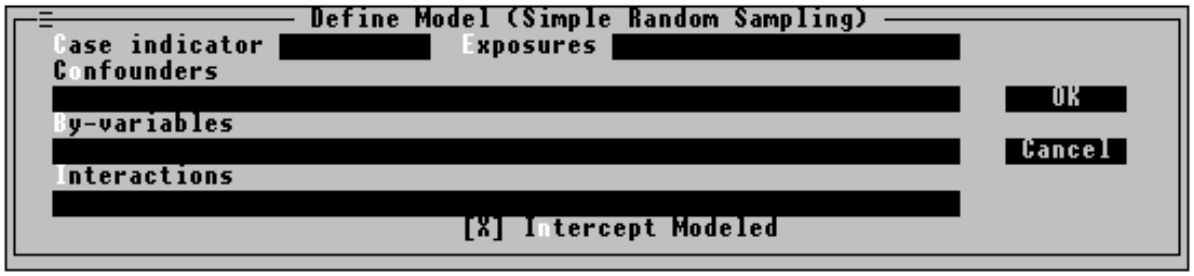
New
Define model [Alt+D]
Sampling scheme
Convergence criteria
Run [Alt+R]

The model menu offers commands that pertain to defining and running a model.

New
The "New" command clears the entire model definition. The sampling scheme does not change, but the "Stratifiers Modeled" and "Intercept Modeled" options (if applicable to the sampling

scheme) are returned to their default values.

Define Model
This command allows you to specify the contents of the model. It presents a dialog box with a field for entering each part of the model. The exact form of the dialog depends on the current sampling scheme (see the "Sampling scheme" command, below), but all versions are similar. When the program starts, the sampling scheme is "Simple Random," which produces the dialog box shown here.



> To enter a model definition
1. When the dialog box appears, the cursor will be in the "Case indicator" field. Type the name of the case indicator variable.
2. Using the mouse, or by pressing Alt+E or TAB, move to the "Exposures" field. Type the names of one or more exposure variables, separated by spaces.
3. If the model includes any co-variants which do not contribute to attributable risk, move to the "Confounders" field. Type the names of the confounder variables, separated by spaces.
4. If the model should be fit separately for each value of one or more variables, move to the "By-variables" field. Type the variable names, separated by spaces.
5. IRAP can automatically create interactions involving any number of variables. To specify an interaction, move to the "Interactions" field. An interaction has the form
var1*var2
Type as many interactions as you wish, separating the terms of each with an asterisk ("*"), and separating different interaction with spaces.
6. IRAP can include a separate parameter to model the intercept, or use a covariant in the model to span the intercept. To span the intercept, clear the "Intercept Modeled" checkbox by clicking it with the mouse, or by pressing Alt+N.
7. When done, click the "OK" button, or press RETURN to accept the model. Click the "Cancel" button, or press ESCAPE to cancel the model definition.

Although the other sampling schemes require slightly different information, the form is the same.

The stratified models (Stratified Random Sampling and Frequency Matching) offer a checkbox labeled "Stratifiers Modeled." When this option is checked, the stratification variables will appear in the regression model as discrete co-variants. Clearing the checkbox will remove the stratification variables from the regression model, while retaining the stratification of the attributable risk estimate.

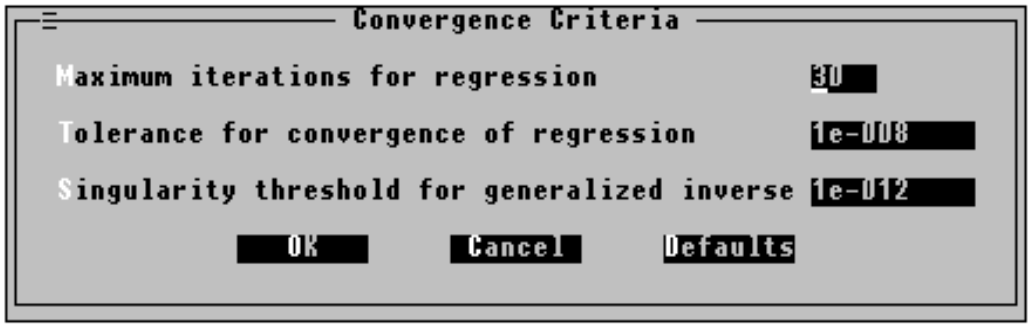
Sampling Scheme

The "Sampling scheme" command offers a submenu listing the six modeling options:

Simple Random Sampling
Stratified Random Sampling
Frequency Matching
1:K_ii Individual Matching
Cohort Analysis
Cross-sectional Analysis

> To select a sampling scheme
* Click your choice with the mouse, or use the arrow keys to highlight your choice, then press RETURN.

Convergence Criteria
This command is used to tune several parameters used in fitting the model. These parameters are saved by the "Save configuration" command in the "Options" menu, and restored automatically when the program starts. The command presents this dialog box.

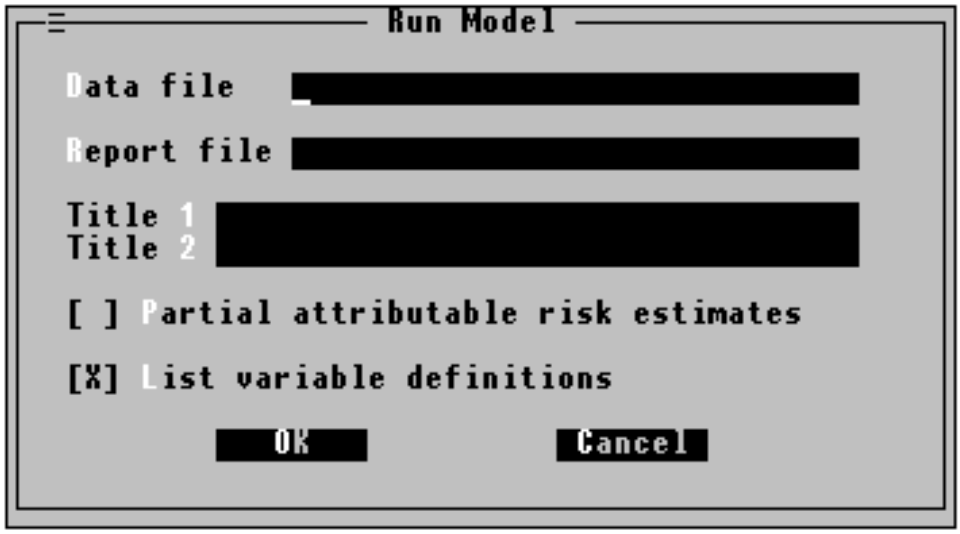


The first line adjusts the number of iterations allowed before the regression model is deemed divergent.

> To change a convergence criterion
1. Move to the field you want to change.
2. Enter the new value.
3. Click the "OK" button, or press RETURN.

> To restore the original values
1. Click the "Defaults" button, or press Alt+D.
2. Click the "OK" button, or press Alt+O.

Run
Use this command to fit the completed model. When you "run" a model, IRAP first reads the data into memory. Little is gained by creating a file that only contains the variables for a certain model; only the values which are used by the current model are actually read from the file. With the data in core, the program builds its parameter tables, then fits the model.



> To run the model
1. In the "Data file" field, type the name of the input file.
2. In the "Report file" field, type the name under which the model's output should be stored.
3. The report file can have up to two title lines at the top of each page. To specify a title, type its text in the "Title 1" and "Title 2" fields.
4. IRAP always computes an estimate of attributable risk for all levels of exposure combined. To produce an estimate for each individual exposure pattern as well, check the "Partial attributable risk estimates" box by clicking it with the mouse, or by pressing Alt+P.
5. By default, IRAP prints the entire file description at the beginning of its report. To suppress this list, clear the "List variable definitions" checkbox by clicking it with the mouse, or by pressing Alt+L.
6. When all settings are correct, click the "OK" button, or press RETURN.

If you specify a report file which already exists, IRAP offers three options:

Overwrite (Alt+O) - Replace the old file with the new one.
Append (Alt+A) - Add the report to the end of the old file.
New name (Alt+N) - Go back, and enter a new name for the report.

The Options Menu

Variable columns
Automatic tiling
Display
Save configuration

The Options menu contains commands for tailoring IRAP's appearance to your taste.

Variable Columns
When the library becomes large, not all of the variables will fit into the "Variables" window at once. While you can always scroll through its contents to find a particular variable, life is easier if you do not need to do so. For this reason, the library can be displayed in one, two, three, or five columns, trading the number of variables displayed against the amount of information shown for each.

> To select the number of variable columns
1. The "Variable columns" command will offer a submenu listing the display options.
2. Click the option of your choice with the mouse, or type the number of columns you wish to see (1, 2, 3, or 5).

In one column, all information is displayed about each variable, but only about 13 or 14 can be seen at once. A horizontal scroll bar will appear for viewing definitions that do not fit in one line. This mode is good for checking long variable names or definitions, but can be annoying for general use.

The default setting is two columns. This mode also shows all of the attributes for file variables, but it truncates variable names longer than nine characters. For defined variables, only the beginning of the definition appears, and the missing code is not shown at all.

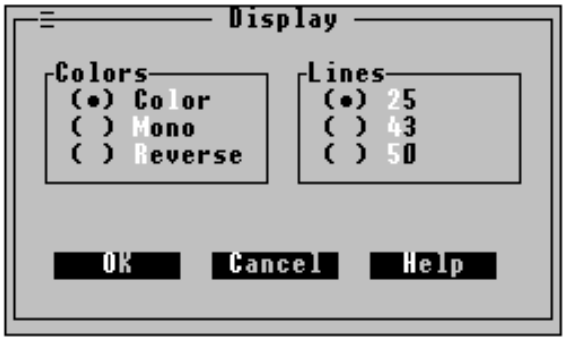
In three column mode, the missing codes disappear for file variables, as well. In five columns, only the variable names themselves are displayed, but about 65 of them will fit into the window at once.

Automatic Tiling
Regardless of the variable format chosen, you may want to manually move or resize the windows, for example to hide part of the model definition in order to see more variables at once. Unfortunately, IRAP has its own idea of how the screen should look. In an effort to be helpful, the program automatically positions the variable window above the model window, so that the model window is just large enough to show the whole model, and the variable window gets the rest of the screen. This feature is called "automatic tiling," because it places the windows next to each other like tiles on a floor.

> To turn Automatic tiling on or off
* choose the "Automatic tiling" command from the Options menu.

Display
The display command is used to change the color scheme and the display mode used by your computer. Because it depends on the video hardware in your computer, its action changes depending on your computer.

Most computers that support IRAP have some form of VGA display. For these computers, the following dialog box will appear.



The default choices of "color" and "25 lines" are appropriate for most cases. Two alternative color schemes, "mono" and "reverse mono," are offered; these may look better on monochrome screens. The "Lines" option offers a choice of video modes. More lines means more information, but it also means smaller type.

> To change the color scheme
1. click the button of your choice, or press:
    Alt+L for color
    Alt+M for monochrome
    Alt+R for reverse monochrome.
2. Click the "OK" button, or press RETURN.

> To change the number of screen lines
1. click the button of your choice, or press:
    Alt+2 for 25 lines
    Alt+4 for 43 lines
    Alt+5 for 50 lines.
2. Click the "OK" button, or press RETURN.

On a system with a CGA or Herculese display, you will find a checkbox marked "Snowy." On some systems, mostly those with CGA monitors, the screen will become snowy when the computer is updating its contents. While this is not dangerous to you or your computer, it can be extremely annoying. If this is a problem on your screen, marking the "Snowy" box should help to alleviate it. Because the EGA and VGA displays handle the problem themselves, the option is not offered for them.

Save Configuration

IRAP can save its configuration settings for future use. To save the current configuration, simply choose the "Save configuration" command in the "Options" menu. No dialog boxes or submenus will appear, but a file called "irap.cfg" will be created in the directory that contains irap.exe. The next time IRAP is started, it will read this file, setting the configuration options automatically.

> To have IRAP look for (and save) the configuration file in a different place
* start the program with the command:
    irap @filepath

The "filepath" can be any valid DOS file specifier. For example,

irap c:\home\williams\williams.cfg

irap mine.cfg

irap \i.c

The following settings are saved in the configuration file:

* "Color scheme" display setting
* "Lines" display setting
* "Snowy" display setting
* Variable columns
* Automatic tiling
* Maximum iterations for regression
* Tolerance for convergence of regression
* Singularity threshold for generalized inverse

The Help Menu

Help for help
Keys help
Contents
About

The Help menu is one way to access on-line information about IRAP.

Help for Help
This command gives instructions for navigating the on-line Help system. The instructions it presents on the first screen should teach you enough to use the system to learn more.

Keys Help
This command gives instructions on using IRAP's interface.

Contents
This command gives a table of contents for the main body of on-line help. Each entry in the table of contents is a hypertext link to a specific topic. Or, to start reading the on-line manual at the beginning, simply press the "Next" button.

About
This command displays a brief message about the IRAP program, including the version number.

Appendix A: User Interface Basics

The IRAP interface should already be familiar to those who have used Windows, or a variety of programs based on the same interface standard. A mouse is useful, especially for the inexperienced user, but is by no means necessary.

The IRAP screen is divided into three major components: the menu bar, the variable window, and the model window. By default, the model window is automatically sized to show the entire model, and the variable window occupies the rest of the screen. However, this feature can be disabled, and the user (with a mouse) can move and resize these windows manually.

The menu bar contains five menus: File, Variables, Model, Options, and Help. Although normally hidden, each menu contains several commands. To view the contents of a menu, either click on the menu's title with the mouse, or press the Alt key and the highlighted letter in the menu title (Alt+F for the File menu). Having opened the menu, one may either select a command, or close the menu without further action. As with the titles on the menu bar, each selection has a highlighted letter. To select an item using the keyboard, either type the appropriate letter, or highlight your choice (using the arrow keys), and press Return. To close the menu without making a selection, press Escape. To make a selection using the mouse, just click on your choice.

To the right of some menu selections, you will find key combinations inside square brackets, for example:

Exit [Alt+X]

These are called "shortcut keys," and provide yet another way to execute commands. To execute the Exit command (i.e., to exit IRAP), you do not have to open the File menu, and then choose Exit; pressing Alt+X (hold the Alt key down while pressing the X key) will have the same effect. Shortcut keys allow an experienced user to perform routine tasks conveniently, while still offering the menu interface for quick reference, and for less common commands.

The simplest type of menu item is a "toggle," such as the "Automatic tiling" command in the "Options" menu. This command merely switches the automatic window option on and off, and reflects the current state of affairs by placing a check mark beside the option if it is currently selected.

The second type of menu item is a submenu, such as "Variable columns" command in the "Options" menu. This type of selection is denoted by an arrow at the far right of the menu. It acts like the toggle switch, but selects from more than two options: a menu appears showing several mutually exclusive options from which to choose--in this case, the number of columns for the variable library display.

Most menu commands do not have check marks or submenus. These commands usually open a "dialog box" to do their work (for example, the "Add variable" command in the "Variables" menu), but not all need to do so. "Save configuration" (in the "Options" menu) quietly updates the configuration file, then returns you to the main screen.

A "dialog box" is a window that can interact with the user. The interaction can be as simple as presenting a message, or as complex as presenting a list of filenames, and allowing you to browse through different drives and directories to make a selection. The "Define Model" dialog

box is a typical example. To open this dialog box, select the "Define model" command from the "Model" menu (or simply press Alt+D from the main screen). You will notice several gray boxes, each with a label. You will also notice that each label contains one highlighted letter. At the right of the dialog box are two "buttons," and at the bottom, a "checkbox" labeled "Intercept Modeled."

The gray boxes are for typed input, in this case for lists of variables. To move the cursor from one box to the next, press the TAB key. The SHIFT+TAB combination moves to the previous box. Alternately, you can use the highlighted letters to move to a particular box (Alt+B will put the cursor in the "By-variables" box), or merely click on the box with the mouse. To enter information, simply move the cursor to the desired box, and start typing.

The checkbox behaves much like the toggle selections in a menu. If the box is marked, the selection is enabled. To change the state of a checkbox, hold down the Alt key while pressing the highlighted letter, or click on the box with the mouse.

The buttons perform simple actions. "Pressing" the "OK" button accepts any changes made to the dialog box, and closes it. Pressing "Cancel" closes the dialog box, but discards any changes you have made. To "press" a button, TAB to it (eventually, the button's text will turn bright white), then press Return. Using the mouse, you can simply click on the button to "press" it. As a shortcut, if the cursor is not on a button (for example, if it is in a gray input box), the Return key is the same as "OK." Similarly, Escape is always "Cancel." Some buttons have highlighted letters in their names. For these buttons, the Alt+letter combination will "press" the button as well.

Another, less common, dialog box control is the "radio button," found in the "Display" dialog box under the "Options" menu. This acts like a station preset button on a car radio, or like the multiple-choice submenus discussed above: it allows the user to choose from a list of mutually exclusive options. Again, to make a selection, use the Alt+letter method, or click on the button of your choice. In each list of radio buttons, one will be selected. Picking another option will move the little dot to that option, indicating that it is the current selection.

The final type of window control is the "scroll bar." Scroll bars appear in several dialog boxes, but the most obvious scroll bars are on the Variable and Model display windows on the main screen. The contents of these windows can become too long to fit on the screen, so the scroll bars are provided to move through the data. To operate the scroll bars using a mouse, simply click on an arrow at the top or bottom of the bar. The text in the window will "scroll" up or down, until you release the button. Using the keyboard, press Alt+F6 until the right window is selected (i.e., it has double lines around the border). When the program starts, the Variable window will be selected. Now, you can use arrow keys or Page-up/Page-down to scroll through the text. Of course, if all the text already fits in the window, then there will be nothing to scroll. You only need to use scroll bars when not everything will fit into the window at once.

Changes from Version 2.0 to 2.2

- Variable names are now case-insensitive (although an “official” mixed-case spelling is retained for readability).
- The bootstrapping facility (via the special “_boot_” by-variable name) has been added.
- Versions prior to 2.2 had a problem with the “1:K_i individual matching” sampling scheme, if by-variables were used. Incorrect results were produced if the same set number was used for matched sets in more than one by-group (for example, each by-group contained a matched set “3"). This problem has been fixed.