SAS Analyst for Windows Tutorial

Statistics
+ Data
Sciences
Statistical
CONSULTING

Updated: August 2012

Table of Contents

Section 1: Introduction

1.1 About this Document

This document introduces you to SAS Version 8. It is primarily aimed at first-time users, however, if you are already familiar with previous versions of SAS, you can use this document to learn about some of SAS's new features. The document is organized into six sections. The first section provides a brief introduction to SAS. The second section will guide you through each of the windows in SAS and discuss their functions. The third section demonstrates how to create SAS data sets by importing data from other applications, such as Excel. The fourth section introduces the *Analyst Application* in SAS. The *Analyst Application* provides a window-driven interface that you can use to both manipulate data and run statistical procedures. The fifth and sixth sections of this document demonstrate how you can use the *Analyst Application* to perform some common data manipulation and analytical tasks. Throughout these sections, a single dataset, *fitness,* is used for all examples. Thus, you will have access to the dataset and will be able to test your knowledge by replicating the examples contained in this document. You can download instructions for creating this dataset. This page contains detailed instructions on how to use the *Program Editor* in SAS to create the *fitness* dataset on your computer.

1.2 Introduction to Version 8 of SAS

Over the years, SAS has developed a reputation of being powerful and full-featured, yet also having a steep learning curve. First-time users are often daunted by the necessity of working with complex computer syntax in order to perform even the most elementary kinds of statistical analysis. However, the new release of SAS has a number of new features that promise to make SAS more user-friendly. In particular, the current version of SAS has a substantially enhanced windows-driven interface that allows you to point and click your way through many tasks that previously required knowledge of SAS programming syntax. This document places a heavy emphasis on the new features available in SAS Version 8, and deemphasizes working with SAS syntax.

Section 2: An Overview of SAS V.8 for Windows

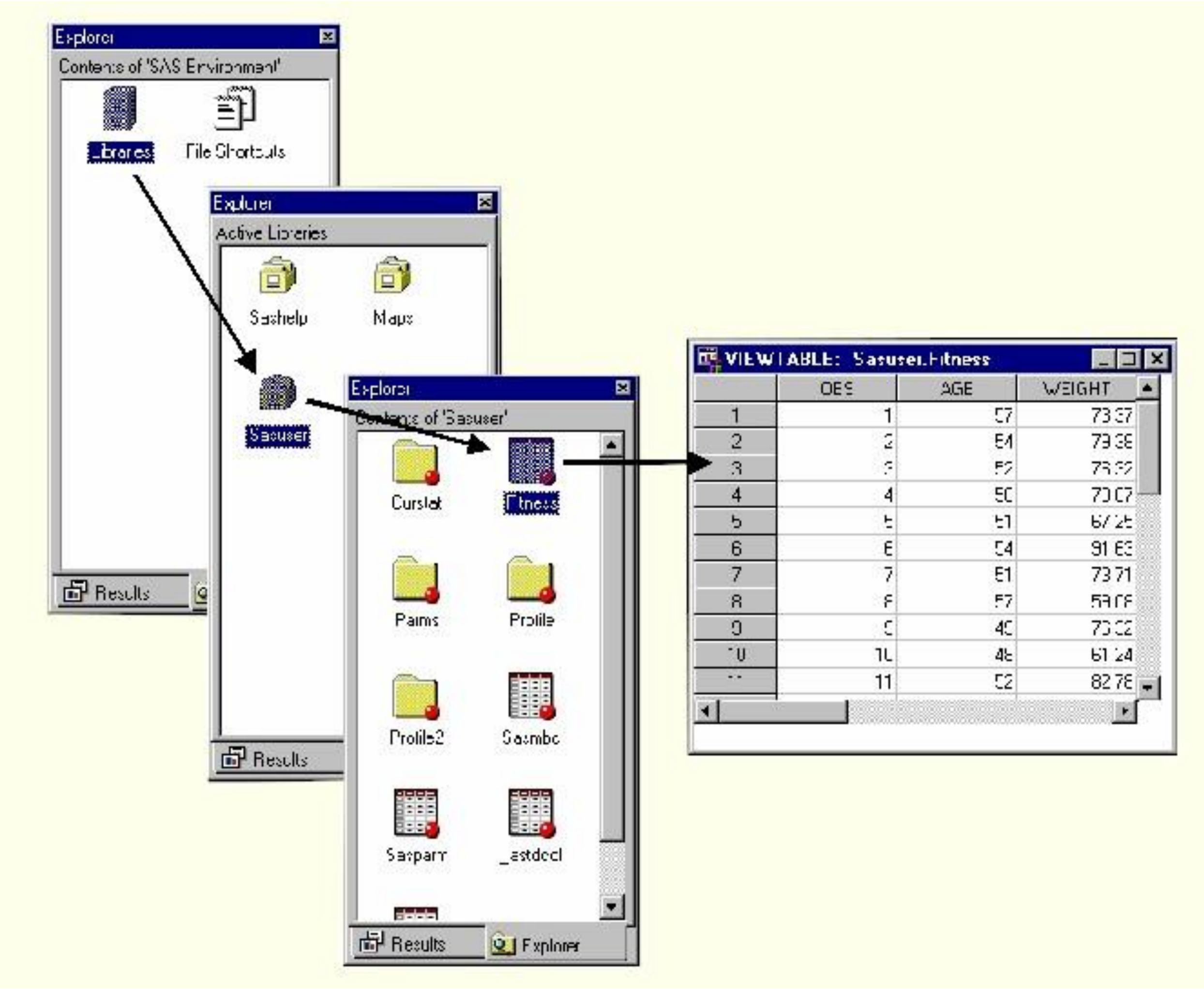
2.1 Navigating through SAS for Windows

In SAS Version 8, there are a variety of ways to navigate through the various SAS windows. If you are familiar with SAS Version 6.12, you will find many new features in Version 8.

2.2 The Explorer/Results Window

If you are familiar with SAS V.6.12, the Explorer/Results Window will probably be one of the first new features of SAS you will notice. After starting SAS Version 8, the Explorer/Results Window appears on the left side of your screen. The Explorer/Results Window is a tool for browsing SAS *libraries*, program files, and the results of statistical procedures. By default, the Explorer Window will appear in front of the Results Window upon startup. However, you can switch between these two windows by clicking on the tabs located at the bottom of the Explorer/Results Window. The Results Window is used in connection with the Output Window, and both of these features will be discussed later in this document. The Explorer Window is primarily used to locate SAS data sets and other SAS-related files. By clicking on the icon labeled *Libraries*, you can explore the various default libraries that contain a number of sample SAS datasets. SAS datasets themselves have their own icons that should be easy to recognize. You can view a SAS dataset in a spreadsheet by double-clicking its icon. The example below illustrates how you can access the SAS dataset *fitness* in the SAS library *sasuser*.

Note: *In order to move backwards in the Explorer Window, simply click the leftmost icon on the toolbar that looks like a folder.*



VIEWTABLE: Sasuser.Fitness

	OBS	AGE	WEIGHT
1	1	57	73.37
2	2	54	79.38
3	3	52	76.32
4	4	50	70.87
5	5	51	67.25
6	6	54	91.63
7	7	51	73.71
8	8	57	59.08
9	9	49	76.32
10	10	48	61.24
11	11	52	82.78

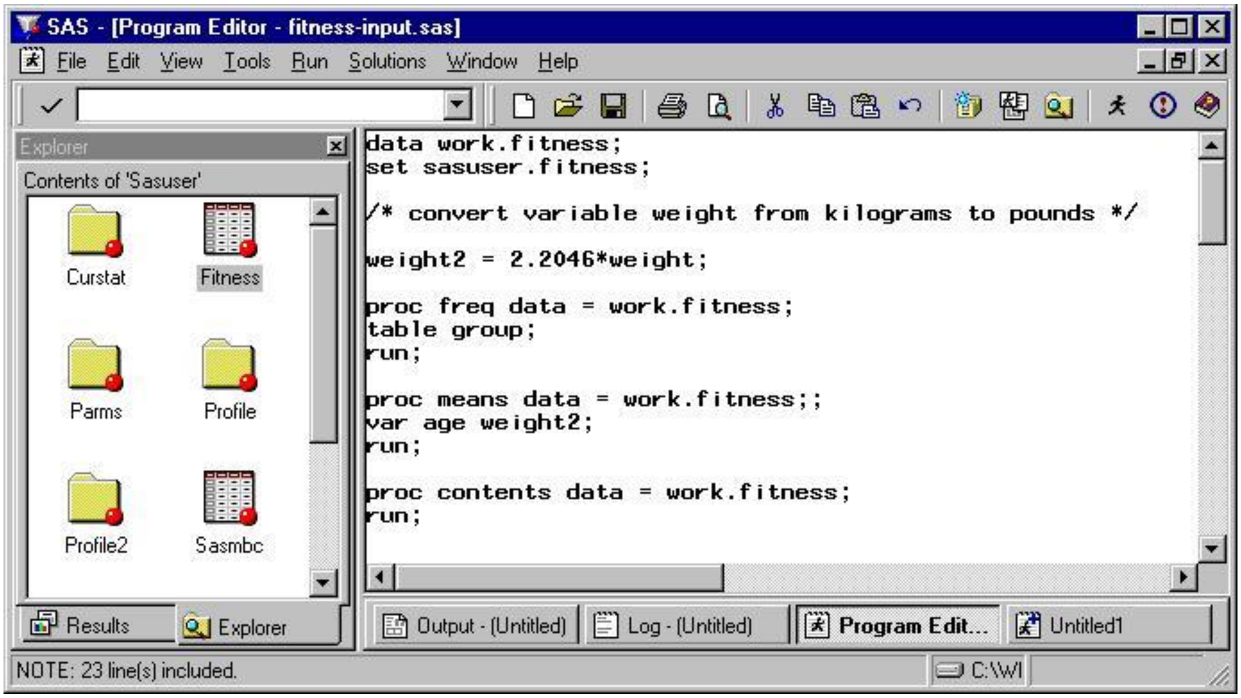
A SAS library is best thought of as a pointer to a location or directory on a computer disk that contains SAS datasets. This unique feature of SAS was originally intended to make the writing of SAS programs more efficient; however it is often a source of confusion to new users. SAS libraries will be discussed in more detail later in the course. The *File Shortcuts* icon serves slightly different purposes. Primarily, it is offers a convenient way of accessing often used SAS programs. However, because this course does not emphasize SAS programming, this feature is not covered in detail.

2.3 The Program Editor

The Program Editor is used to create, edit, and execute SAS programs. In previous versions of SAS, the program editor was the primary interface between you and SAS's data processing engine. All data manipulation and analytical tasks were performed through the writing of syntax that instructed SAS to process your data. While the Program Editor is still a critical feature of SAS, more recent versions of SAS contain pulldown menus with dialog boxes that allow you to submit commands to SAS without ever writing syntax. The new windows-driven interface has not completely supplanted all of SAS's functionality--many data manipulation and analytical tasks are only available through SAS syntax. You will also find it advantageous to preserve any work you have done in SAS by saving it in the form of syntax. In addition to providing a written record of the work you have done, it is also a convenient way of re-running a particular analysis at a later date. Thus, a later section of this document will demonstrate how to generate SAS syntax.

The example below is meant to guide you through the remaining windows in SAS. It first demonstrates how a relatively simple SAS program can be opened and then submitted within the Program Editor. It then provides a brief introduction to a new feature of Version 8, the Enhanced Program Editor. Finally, it demonstrates how you can use the Output and Log windows to view the results requested by the submitted program, as well as to check the program for errors. While the example below is not intended to introduce you to SAS programming, it is important that you learn how to use previously written SAS programs and programs generated in SAS applications. As you become more familiar with how SAS operates, you might consider taking another SAS course that focuses on how to write and edit SAS programs.

In order to open a SAS program, make the Program Editor the active window and pull down the *File* menu item and select *Open*. A standard windows dialog box will then appear, allowing you to select the type of file you want to open as well as browse local drives and directories. SAS programs files contain the suffix *.sas,* and when the file type *SAS Files* is selected in the open file of type box, any SAS programs that exist on the currently selected directory should appear in the dialog box. In the example below, the SAS program titled *fitness-input.sas* has been opened and is currently being displayed in the Program Editor.



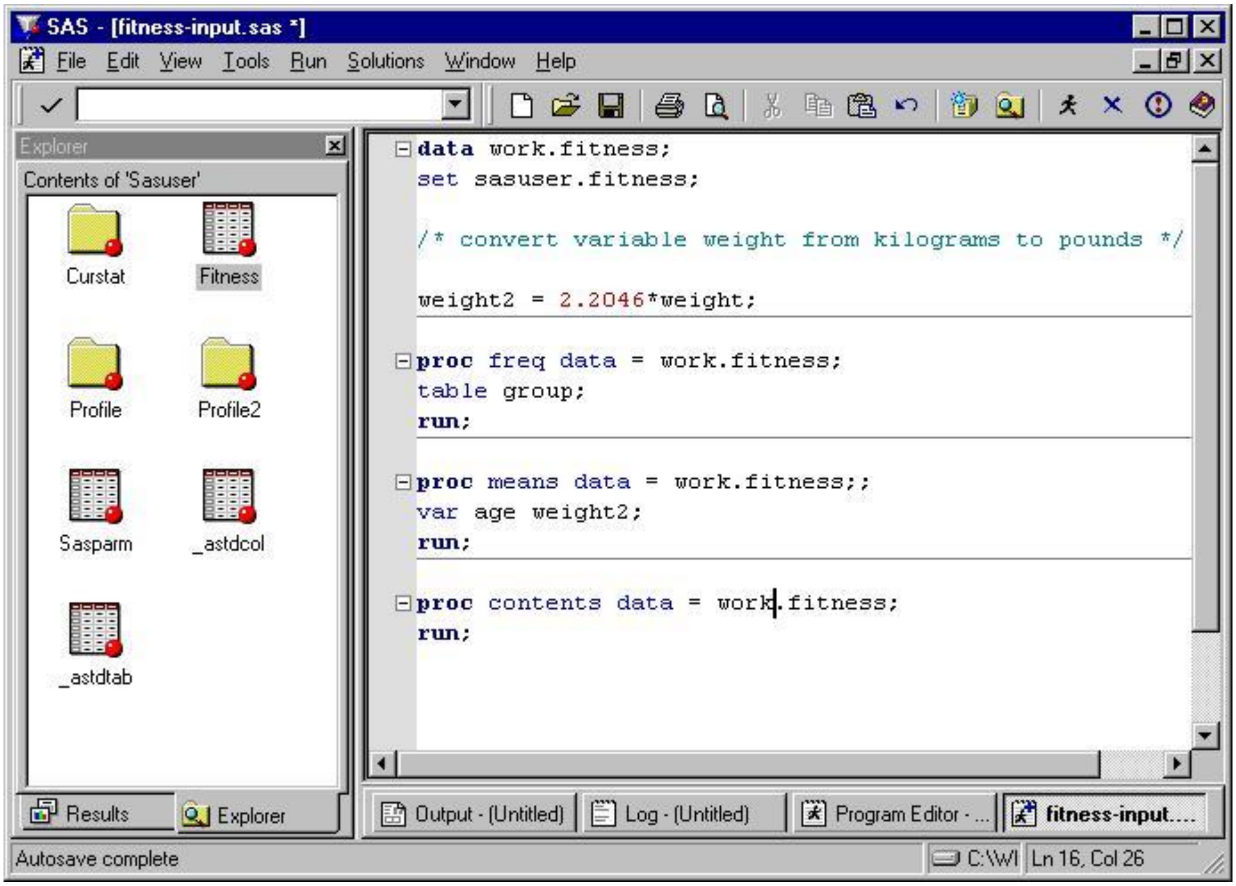
```
data work.fitness;
set sasuser.fitness;

/* convert variable weight from kilograms to pounds */

weight2 = 2.2046*weight;

proc freq data = work.fitness;
table group;
run;

proc means data = work.fitness;;
var age weight2;
run;

proc contents data = work.fitness;
run;
```

NOTE: 23 line(s) included.

The first two lines of the program simply instruct SAS to open the SAS dataset *fitness* located in the SAS library *sasuser* and then write another dataset with the same name to the SAS library *work*. The fourth line of the program creates a new variable in the data set called *weight2*. *Weight2* is created by multiplying the variable *weight* by 2.2046. The comment indicated by asterisks on each end explains that this is done in order to convert the values of *weight* from kilograms into pounds. Finally, the last group of commands requests a variety of descriptive statistics for a few of the variables in the *fitness* dataset.

2.4 The Enhanced Program Editor

The Enhanced Program Editor is a new feature of Version 8 and is primarily intended to make the reading, writing, and editing of SAS programs easier. The most noticeable difference between the two program editors is the use of color in the Enhanced Program Editor, which allows you to easily distinguish between different elements of a SAS program. These colors are applied automatically by the editor so that programs can be read more easily as they are written.



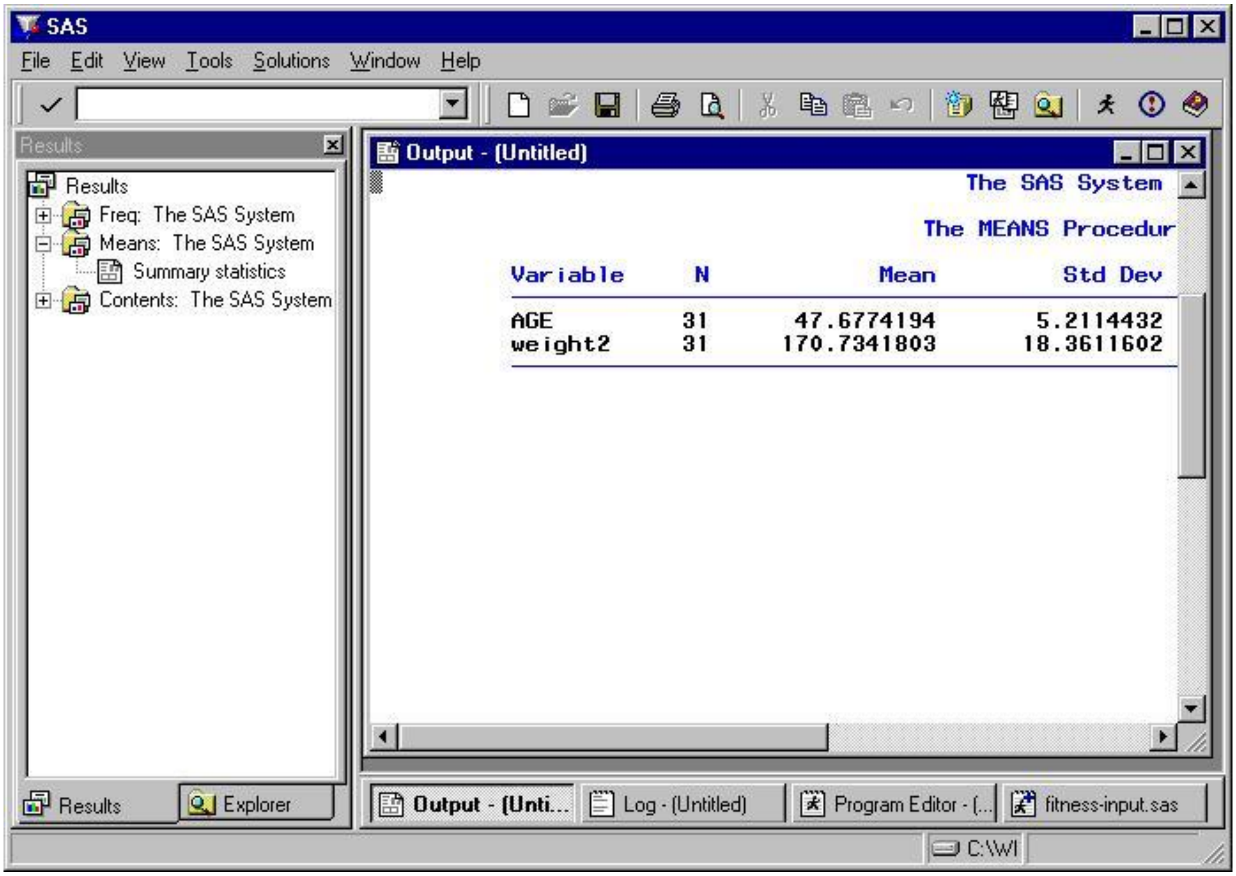
```
data work.fitness;
set sasuser.fitness;

/* convert variable weight from kilograms to pounds */

weight2 = 2.2046*weight;

proc freq data = work.fitness;
table group;
run;

proc means data = work.fitness;;
var age weight2;
run;

proc contents data = work.fitness;
run;
```

After a program has been opened, you can submit it from either Program Editor by clicking on the running man icon located on the right side of the toolbar. If the program you submitted contains requests for output (e.g., descriptive statistics, dataset descriptions), then you can view this output in the Output Window described below.

2.5 The Output Window

After a SAS program has been submitted from either Program Editor, any output requested by the program is printed in the Output Window. The Output Window only allows you to view, print, or save the information it displays. If you want to edit SAS output, you will have to save the contents of the Output Window as a text file and then use an application like Microsoft Word to make changes or include additional information. As you become a more experienced SAS user, knowledge of SAS syntax will allow you to alter the standard output provided by the numerous statistical procedures in SAS.

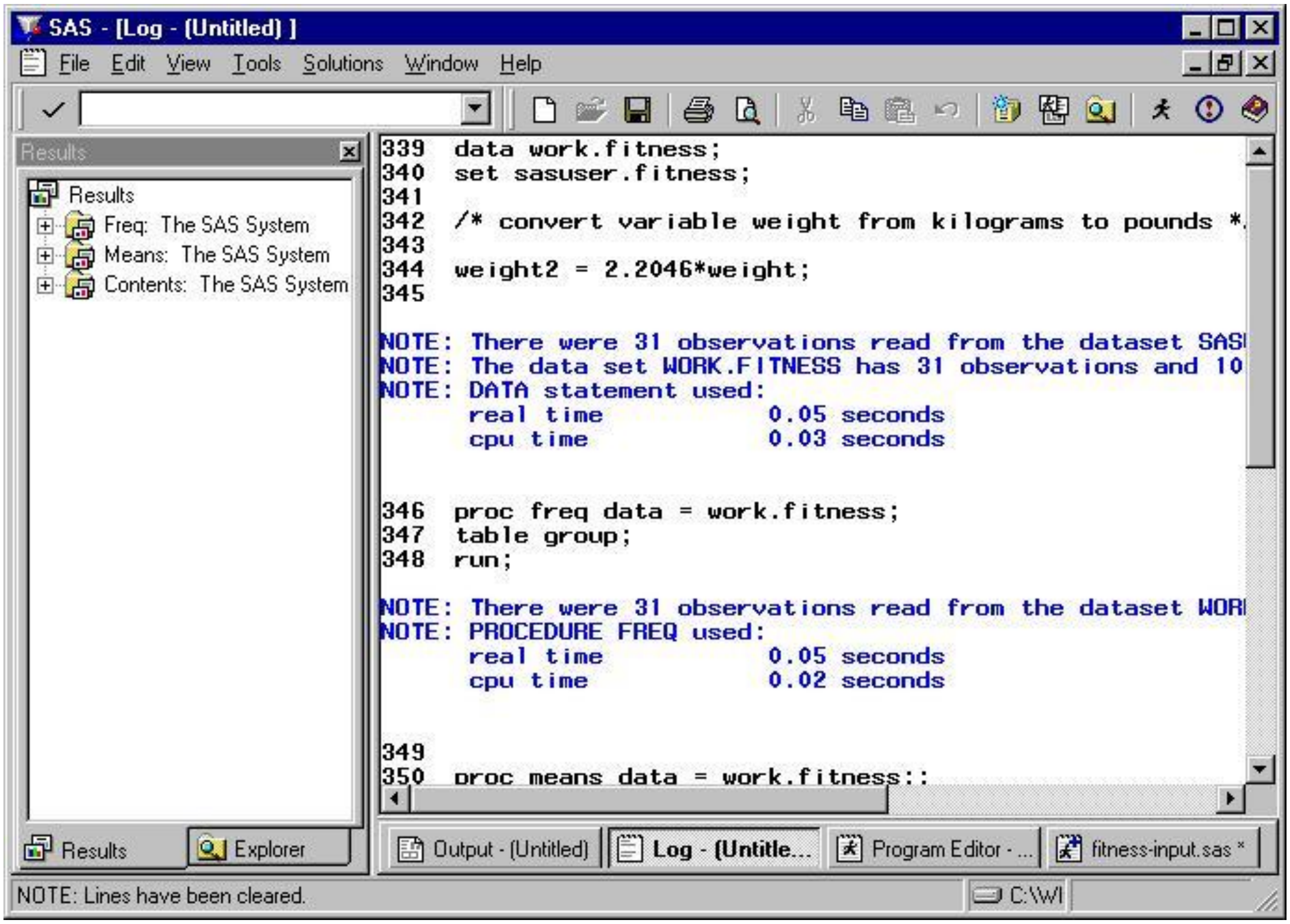
The Results Window works in conjunction with the Output Window, and it primarily serves as a way of organizing the information contained in the Output Window. In the default mode, output pointers appear in a procedural hierarchy. To work with your SAS output you can locate the folder that matches the output for a given procedure you want to view and use the expansion icons (+ or - icons) next to the folder to open or hide its contents. In the example below, the output from the *MEANS* procedure has been expanded so that the page of output titled *Summary statistics* can be viewed. In order to display a particular page, simply double-click the appropriate page icon.



```
                                 The SAS System

                              The MEANS Procedur

     Variable      N          Mean       Std Dev
     ------------------------------------------------
     AGE          31    47.6774194     5.2114432
     weight2      31   170.7341803    18.3611602
     ------------------------------------------------
```

2.6 The Log Window

The Log Window is a tool for diagnosing problems with SAS programs. However, even if you are not experienced in writing SAS programs, it is a good idea to check the Log Window after submitting a SAS program in order to ensure that the program did not encounter any errors. The Log Window also contains important summary information that might be useful to you. For example, in the Log Window below, the number of observations and variables in the *fitness* dataset is given. Comments of this kind always appear in blue. Error messages, on the other hand, appear in red and usually indicate that some portion of the program failed to work properly.



```
339  data work.fitness;
340  set sasuser.fitness;
341
342  /* convert variable weight from kilograms to pounds *
343
344  weight2 = 2.2046*weight;
345

NOTE: There were 31 observations read from the dataset SAS
NOTE: The data set WORK.FITNESS has 31 observations and 10
NOTE: DATA statement used:
      real time           0.05 seconds
      cpu time            0.03 seconds


346  proc freq data = work.fitness;
347  table group;
348  run;

NOTE: There were 31 observations read from the dataset WOR
NOTE: PROCEDURE FREQ used:
      real time           0.05 seconds
      cpu time            0.02 seconds


349
350  proc means data = work.fitness;;
```

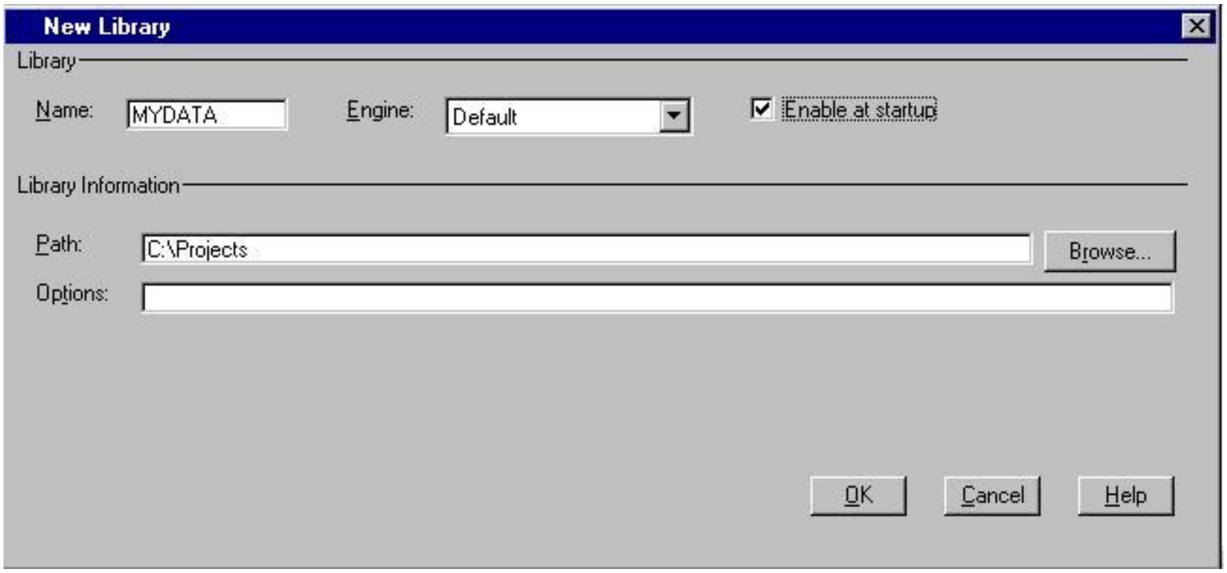
Section 3: Importing Data into SAS

3.1 Introduction

In order to get started with SAS, you will need to know how to read in your data into SAS. There are two steps involved in reading data into SAS. The first step is to tell SAS where it can both find and write SAS datasets. This is accomplished by creating *user-defined libraries.* The second step is to determine the file format in which your own data is stored. In the section below, the document takes you through each of these steps. First, it demonstrates how to create a SAS library using the Explorer Window. Second, it demonstrates how you can import data stored in a variety of file formats using the SAS Import Wizard.

3.2 Creating a SAS Library

As was mentioned above, SAS libraries are best thought of as pointers to locations or directories on a computer disk. In order to read in or write out a SAS dataset, you will either have to create one of your own or use one of the existing default libraries. In order to create one of your own, make the Explorer Window the active window and double-click on the *library* icon. Next, right-click on an empty space within the window and select *New*. A dialog box like the one below should appear.



Name: MYDATA

Engine: Default

☑ Enable at startup

Path: C:\Projects

Options:

To create a new SAS library, you will need to define at least two things within this dialog box: the name and the location of the new library. The name of the new library goes in the box in the upper right hand corner of the dialog box. To define a location, you can either type in a path name or simply click the **Browse** button in order to browse local drives and directories. In the example above, the library *MYDATA* is being created. As can be seen in the path specification below the *Name* box, *MYDATA* is located in the *Projects* directory on the C drive. The *Enable at startup* box has also been checked. This means that *MYDATA* will be a permanent SAS library and will not have to be defined every time SAS is started up again. If you plan to access data from a particular location repeatedly, enabling this feature prevents you from having to redefine a library every time you start up SAS.

SAS libraries are also defined by particular *engines* that instruct SAS to read data stored in different types of file formats. However, this document recommends using the Import Wizard when your data is not in SAS dataset format. You should therefore leave the engine setting in the default mode when creating a new library.

Note: *SAS V.8 can read datasets created by SAS V.6.12 under the default engine setting.*

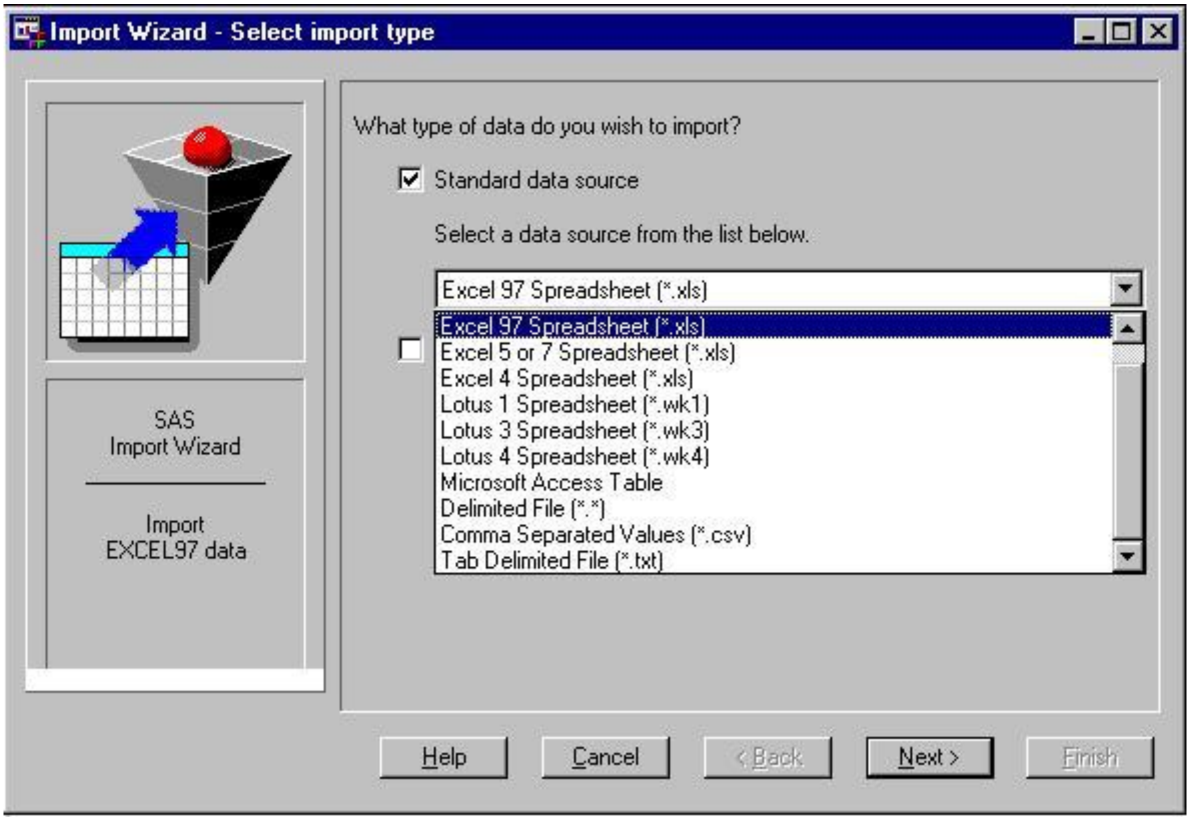
3.3 Using the Import Wizard

The Import Wizard is a convenient tool for reading in data stored in file formats other than the SAS dataset format. The example below demonstrates how the Import Wizard can be used to convert an Excel spreadsheet into a SAS dataset and then save it to the SAS library *MYDATA* that was created in the previous step. For the purposes of this example, the SAS dataset *fitness* was converted into an Excel document and written to a floppy disk.

The Import Wizard can be accessed from any of the SAS windows covered in this document. Simply go to the *File* menu and select *Import data*:

File
Import Data...

In the first dialog box of the Import Wizard, you are asked to specify whether your data exists in a standard file format (e.g., Excel, Access, tab-delimited files, etc.) or a user-defined format. Because this document assumes your data is more likely to be stored in one of the standard file formats, the user-defined format option will not be covered here. In the example below, the standard data source box has been checked and the *Excel 97 Spreadsheet* format has been selected. In order to proceed to the next step, simply click the box labeled **Next** at the bottom of the dialog box.



Import Wizard - Select import type

What type of data do you wish to import?

☑ Standard data source

Select a data source from the list below.

Excel 97 Spreadsheet (*.xls)

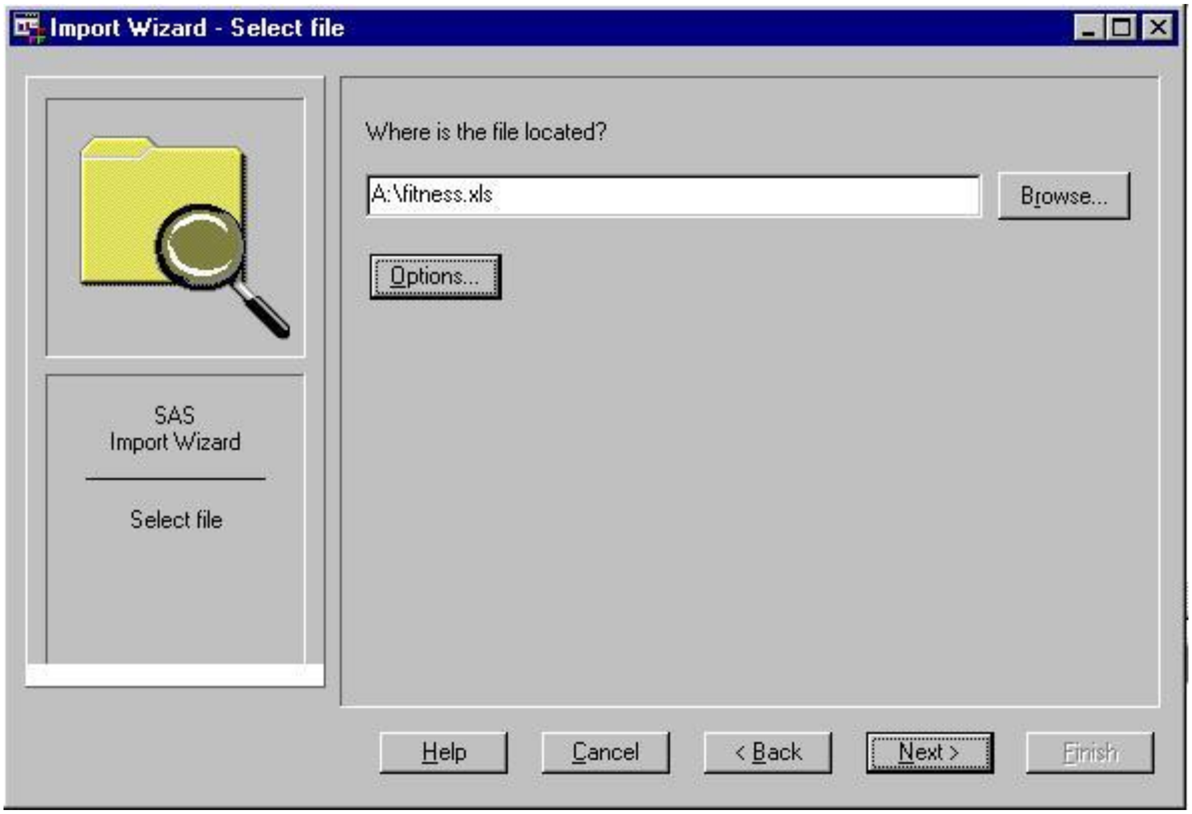
☐

SAS Import Wizard

Import EXCEL97 data

Excel 97 Spreadsheet (*.xls)
Excel 5 or 7 Spreadsheet (*.xls)
Excel 4 Spreadsheet (*.xls)
Lotus 1 Spreadsheet (*.wk1)
Lotus 3 Spreadsheet (*.wk3)
Lotus 4 Spreadsheet (*.wk4)
Microsoft Access Table
Delimited File (*.*)
Comma Separated Values (*.csv)
Tab Delimited File (*.txt)
Help
Cancel
< Back
Next >
Finish

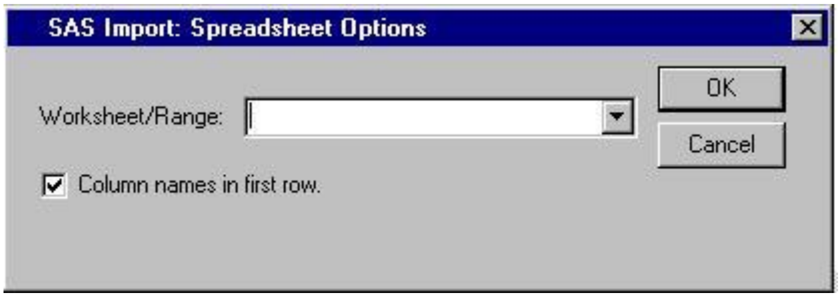
In the next dialog box that appears, you will be asked to specify the name of the file and where it is located. If you know the path and the exact name of the file, then you can directly enter this information in the appropriate box. Otherwise, you may click the **Browse**... button to browse local drives and directories. Because the Excel document *fitness.xls* is on a floppy disk in the A drive, the path and file name "A:\fitness.xls" has been entered in the example below.



Where is the file located?

A:\fitness.xls

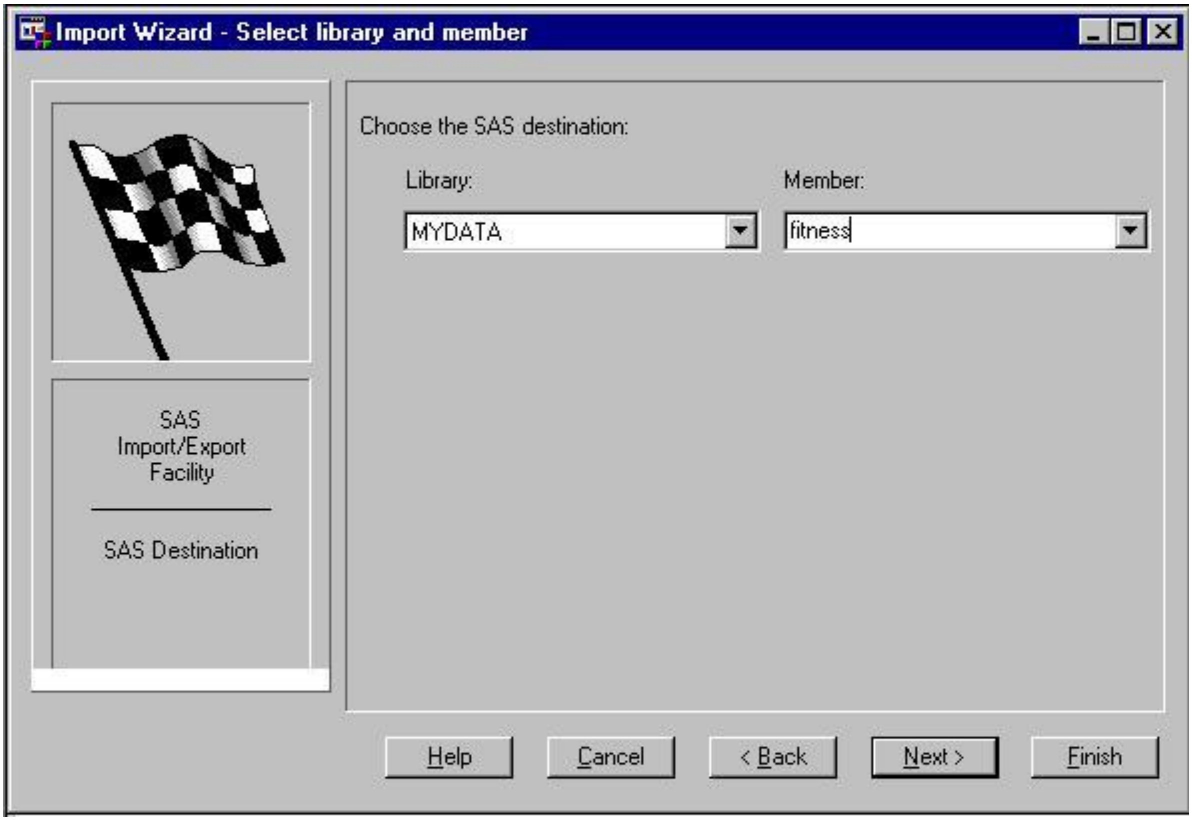
If your data are contained on multiple worksheets in an Excel document, the **Options** button allows you to import data from a specified range of worksheets rather than importing the entire document. In addition, the **Options** button allows you to specify whether the first row of the spreadsheet contains the variable names you would like to use in the SAS dataset you are about to create. The default setting for this feature assumes that the variable names you want to use *are* located in the first row of the Excel spreadsheet.



Worksheet/Range:

☑ Column names in first row.

In the final step, you are asked to specify a SAS library and member name for your new SAS dataset. The member name corresponds to the filename containing the new dataset. In the example below, the SAS library *MYDATA* has been selected and the member name *fitness* has been given to the new dataset. After this information has been entered, you may click the **Finish** button located at the bottom of the dialog box to save your new dataset. Afterwards, it is a good idea to examine the Log Window for any errors that may have occurred during the conversion process.



Choose the SAS destination:

Library: MYDATA

Member: fitness

You can also request that the syntax for the *Import* procedure be written to a SAS program file. To do so, simply click **Next** button in the dialog box above. You will then see a final dialog box that asks you to specify a location for saving the requested program file.

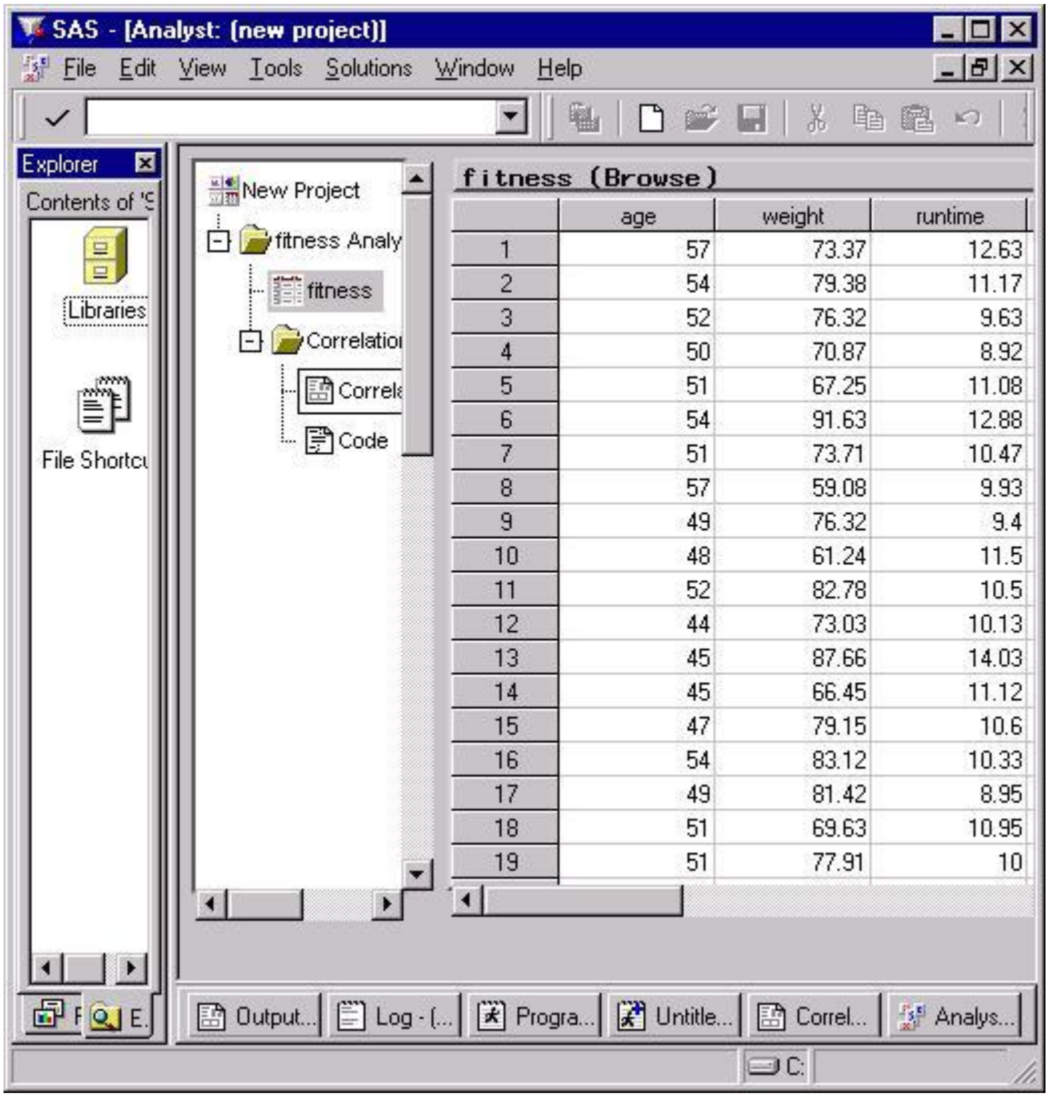
Section 4: The Analyst Application

The SAS System has several applications and systems that operate within it. These components interact with the standard components of the SAS System, and they have several unique features as well. The *Analyst Application* provides you with a user-friendly interface in which you can perform data manipulation, conduct statistical analyses, visualize your data, and create graphical displays of your data. The Analyst Application interacts with the SAS system and can be used to generate SAS syntax that can be run in the Program Editor.

To open the Analyst Application, go to the *Analysis* submenu of the *Solutions* menu and select *Analyst*:

Solutions
  Analysis
    Analyst

Doing so will open the Analyst Application. An example of the Analyst applications in which some analyses have been conducted is shown below:



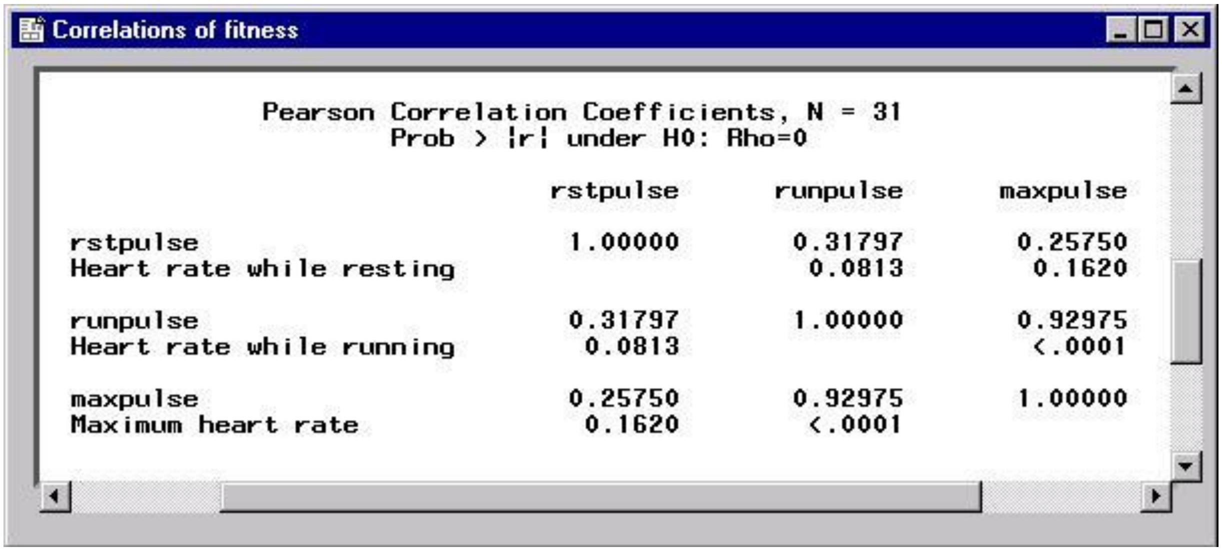
fitness (Browse)

	age	weight	runtime
1	57	73.37	12.63
2	54	79.38	11.17
3	52	76.32	9.63
4	50	70.87	8.92
5	51	67.25	11.08
6	54	91.63	12.88
7	51	73.71	10.47
8	57	59.08	9.93
9	49	76.32	9.4
10	48	61.24	11.5
11	52	82.78	10.5
12	44	73.03	10.13
13	45	87.66	14.03
14	45	66.45	11.12
15	47	79.15	10.6
16	54	83.12	10.33
17	49	81.42	8.95
18	51	69.63	10.95
19	51	77.91	10

The screenshot above contains an open SAS dataset, *fitness*, on which some correlations have been run. There are several features in the above figure that are common to all Analyst sessions. The diagram immediately to the left of the dataset is a map of the components of the session in the order that they were produced. These icons can be used to move to the items they represent. They also can be used to edit these items. By right-clicking on the icons, you will obtain a menu containing several options that include the ability to print, delete, or save that component of your Analyst session. The first icon, labeled *New Project*, is present in all Analyst sessions and merely represents the fact that the application is open. Below that is a folder labeled *fitness Analysis,* which indicates that a SAS dataset was opened. Opening a dataset is the first step for using the Analyst Application. You can do this by going to the *File* menu and selecting the *Open* option for a list of SAS datasets:

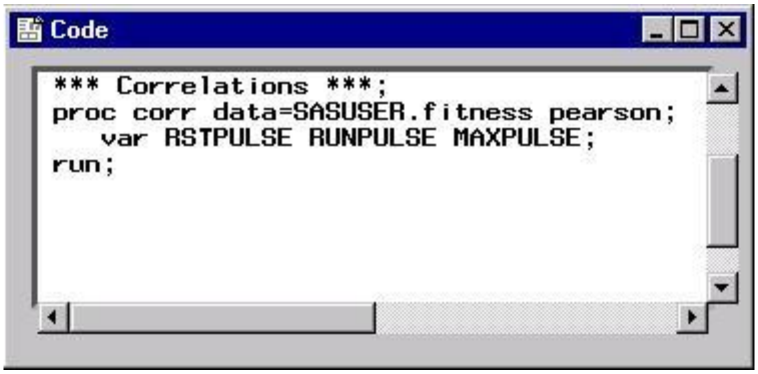
File
Open...

After the *fitness* dataset was opened in the above Analyst session, the next level in the diagram represents analyses that were performed on that data set. This includes a folder labeled *Correlations* that has two icons below it that are labeled *Correlations of fitness* and *Code*, respectively. These two icons roughly correspond to the Output and the Program Editor windows described earlier. When a procedure is run, the output will open in the foreground automatically in a window as shown below:



```
                Pearson Correlation Coefficients, N = 31
                        Prob > |r| under H0: Rho=0

                              rstpulse      runpulse      maxpulse

rstpulse                       1.00000       0.31797       0.25750
Heart rate while resting                      0.0813        0.1620

runpulse                       0.31797       1.00000       0.92975
Heart rate while running        0.0813                      <.0001

maxpulse                       0.25750       0.92975       1.00000
Maximum heart rate              0.1620        <.0001
```

If you have minimized the above window and done other analyses, you can reopen it by double-clicking on the *Correlations of fitness* icon that represents this output. You can also access the SAS syntax by double-clicking on the icon labeled *Code*. This will produce the following window:

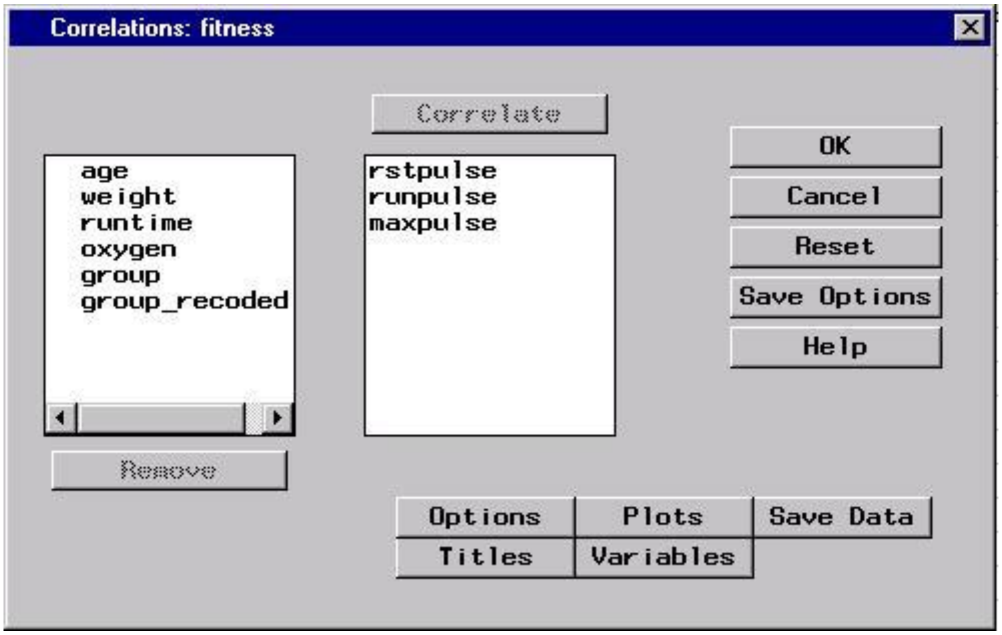


```
*** Correlations ***;
proc corr data=SASUSER.fitness pearson;
   var RSTPULSE RUNPULSE MAXPULSE;
run;
```

The syntax contained in this window is the SAS syntax that can be used to generate the output in the *Correlations of fitness* window. In fact, there is a useful menu item that can be used for pasting this syntax into the Program Editor, where it can be executed or saved for future use. This menu item is the *Copy to Program Editor* option available from the *Edit* menu:

Edit
Copy to Program Editor

The techniques used to produce analyses such as the correlations in the above figure are done through the use of dialog boxes available through the menu system. While these dialog boxes contain many features unique to the analysis that you are conducting, there are some features that are common across most dialog boxes and are thus worth discussing before proceeding with the some of the specific procedures discussed below. The correlations dialog box shown below, which was used to generate the example output and syntax above, contains features that are common to many other dialog boxes:



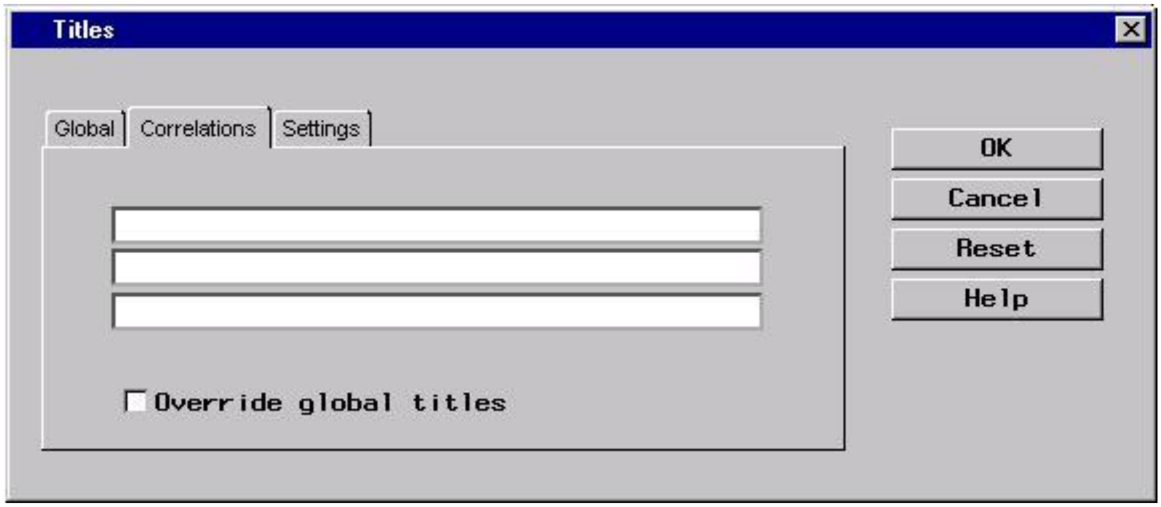
The features shown above that will be present in most if not all dialog boxes in the Analyst Application are these:

- Variables in the dataset are listed on the left side of the dialog box
- The **OK** button executes a procedure
- The **Reset** button clears all specified options
- The **Titles** button allows you to add titles to your output
- The **Variables** button provides options for specifying variable characteristics relevant to your analysis

The first point on the above list refers to the fact that when you open a dialog box, variables in your dataset are listed in the leftmost box. However, as can be seen in the previous example, these variables can be moved from one box to another. This can be done by first clicking on the name or names of variables in the leftmost box, and then clicking on the button above a box to the right to move variables into the box below. For example, in the above dialog box, the variables *rstpulse, runpulse,* and *maxpulse* were clicked in the leftmost box, and then moved to the box below the **Correlations** button by clicking on that button. By selecting these three variables, you would obtain correlations among only those variables and not any of the other variables in the leftmost box.

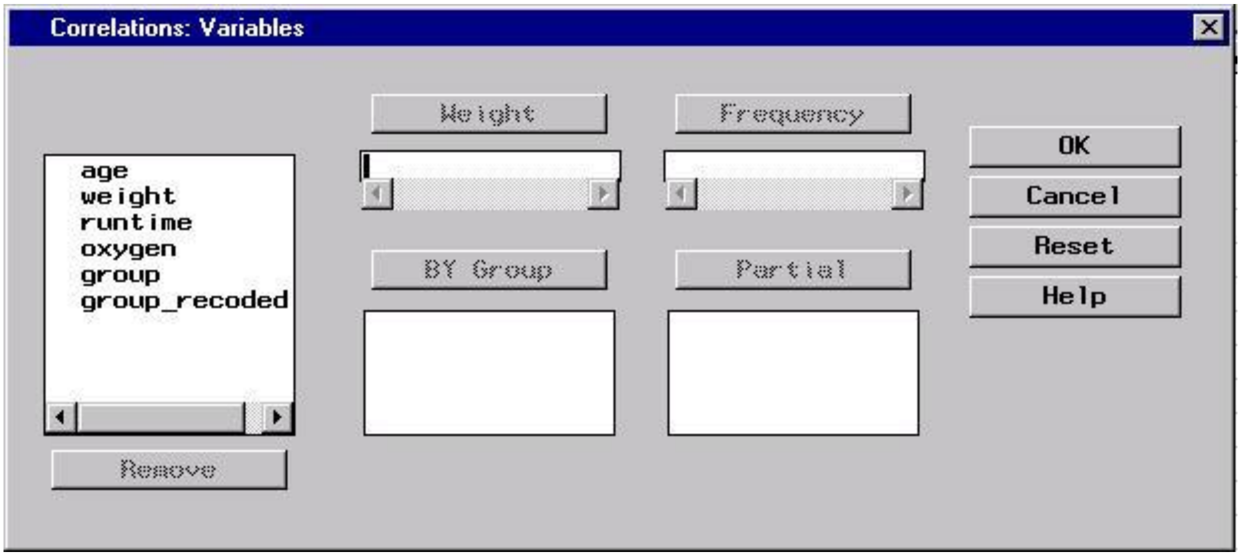
The **OK** button is used to execute a specified procedure. For example, clicking the **OK** button in the above dialog box would produce a correlation matrix between the three variables listed under the **Correlations** button. The **Reset** button is another button that is present in all dialog boxes. It is used to clear all of the options, including the selection of variables, and will reset the dialog box back to the default options.

Clicking the **Titles** button will produce the following dialog box:



The tab that controls the title for the specific procedure you are using will be selected when you open this dialog box. For example, in the above dialog box, you would specify a title that would appear at the head of your correlations output. The *Global* tab, identical to the *Correlations* tab above, controls titles that appear at the head of all output. When you specify a title in the *Global* tab, it will continue to appear in the output that is generated for subsequent procedures. The *Settings* tab includes options for including page numbers, dates, and information about filters. These options are all selected by default, but they can be easily deselected by clicking the check boxes for each option.

Clicking the **Variables** button in the main dialog box will produce this dialog box:



The *Variables* dialog box for the correlation procedure in SAS contains four ways to define special variables. You move variables to selected boxes in the same manner that was described for moving variables in the main dialog box. A *weight* variable is one that indicates how many cases are represented by each line. For example, if you used the group variable to weight cases, individuals in group 2 would be counted as two cases, individuals in group 1 would only be counted once, and individuals in group 0 would not be counted. *Frequency* variables are defined the same way as *weight* variables. Despite the fact that the *weight* and *frequency* variables are defined the same way, they are treated differently by the correlation procedure, and thus you should research their specific properties before using them in correlations. The *By Group* box is used to define a variable by which the output is split. That is, if you placed the variable *group*, which has three levels, in that box, you would have a separate correlation matrix in the Output window for each of the three levels of *group*.

Section 5: Data Transformations

5.1 Computing new variables

There are several tasks that are routinely done prior to data analysis. One such task is to create new variables based on existing variables. For example, if you have a survey with twenty items that can be averaged together into four factors, each factor consisting of five questions, and you wish to perform your data analysis on those factors, then you will first need to create variables that represent those four factors. To do this, you would average the values of the five items that comprise a given factor and create a new variable. While this is a common example of a new variable that might be created, you are not limited to averaging variables, as you will see below. In fact, you can create any type of new variable that can be defined as a numeric expression.

Using the *fitness* example dataset, a new variable will be created that represents the ratio between a person's pulse rate while running and their resting pulse rate. The ratio is obtained by dividing the running pulse rate, *runpulse*, by the resting pulse rate, *rstpulse*, using the *Compute* option in the Analyst Application.

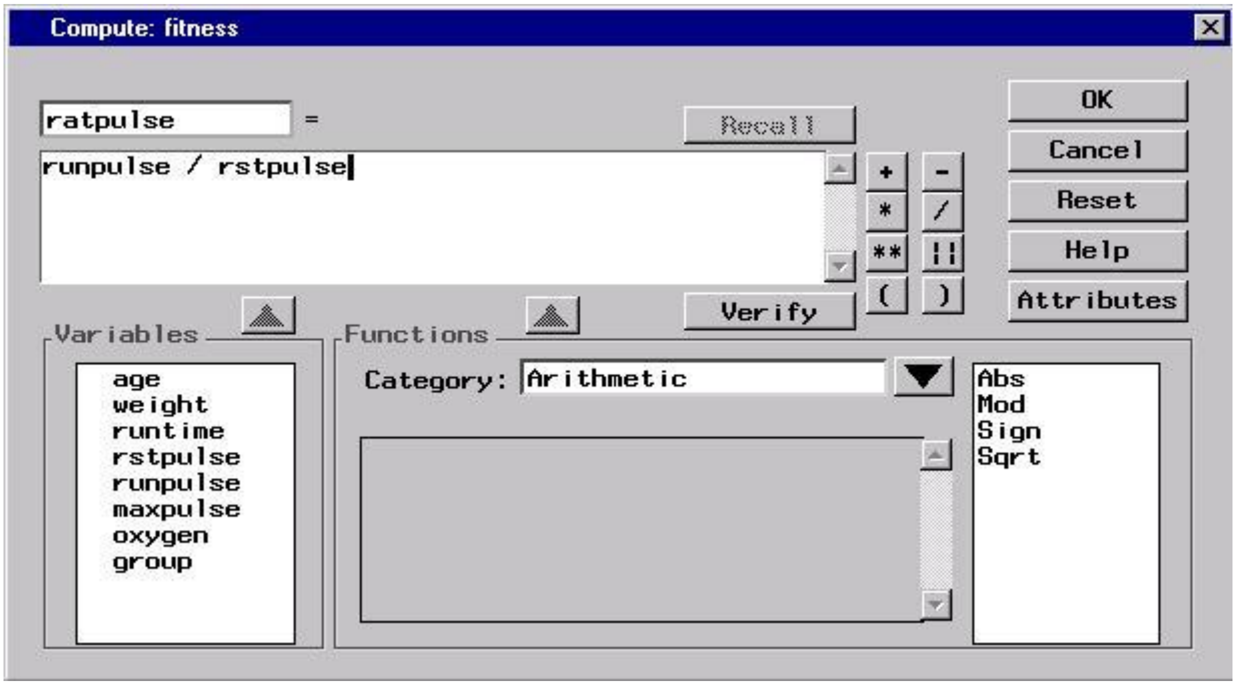
However, before beginning, it is necessary to distinguish between the two modes in which the Analyst application operates. The *Browse* mode is the default mode in which you can view your data but cannot make changes in your data. While in *Browse* mode, you cannot use data manipulation functions such as the *Compute* function to make changes to your existing datasets. *Edit* is the mode in which you can make changes to your dataset manually or by using the data manipulation functions available through the pulldown menus. To change from the *Browse* mode to the *Edit* mode, select *Mode* from the *Edit* menu:

Edit
 Mode
  Edit

Once you have switched to *Edit* mode, you can calculate a new variable. To calculate a person's resting pulse divided by their running pulse, use the *Compute* option in the *Transform* submenu of the *Data* menu:

Data
 Transform
  Compute...

After selecting these menu items, you will obtain the following dialog box:



Compute: fitness

ratpulse =

runpulse / rstpulse

Category: Arithmetic

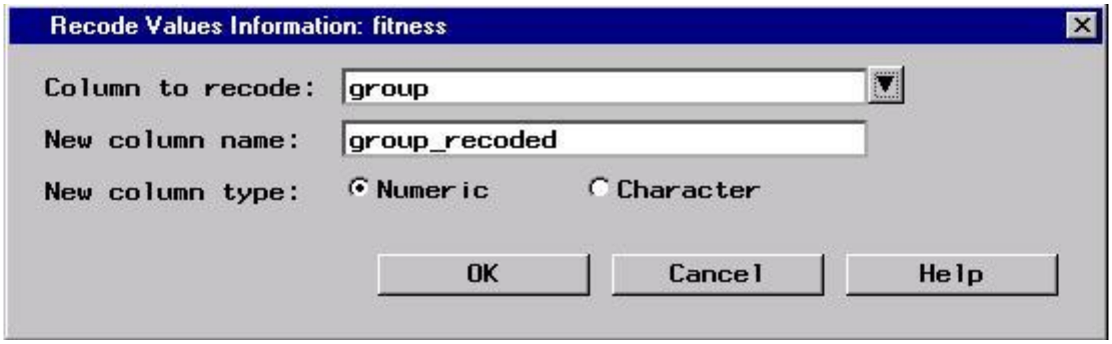
A new variable, *ratpulse,* will be created by dividing a person's running pulse by their resting pulse. The first step is to fill in the box in the upper left of the *Compute* dialog box with the name of the new variable. In this case, it is *ratpulse*. Next, define the numeric expression in the box below the next variable name. This can be done by either typing in the expression or by moving variable names and operators into the box by double-clicking on variable names and single-clicking on operators. For example, you could create the expression shown above by double-clicking on *runpulse* in the list of variables, then clicking on the / operator, and then double-clicking on *rstpulse*. Once you have defined your numeric expression, you can create the new variable by clicking the OK button. After doing so, the new variable *ratpulse* will appear in the dataset.

5.2 Recoding Variables

In addition to creating new variables by defining a numeric expression, it is common for data analysts to create a new variable by redefining an existing variable. For example, you may wish to collapse two levels of a variable into a single variable. You could do this with the variable *group* in the dataset that has three levels: 0, 1, and 2. If the goal was to collapse the 0 and 1 levels into a single group, you could recode all of the 0's as 1's to produce a variable that contains two levels: 1 and 2. To do this, use the *Recode Values* option in the *Transform* submenu of the *Data* menu in the Analyst Application:

Data
Transform
Recode Values...

This will produce the following dialog box:



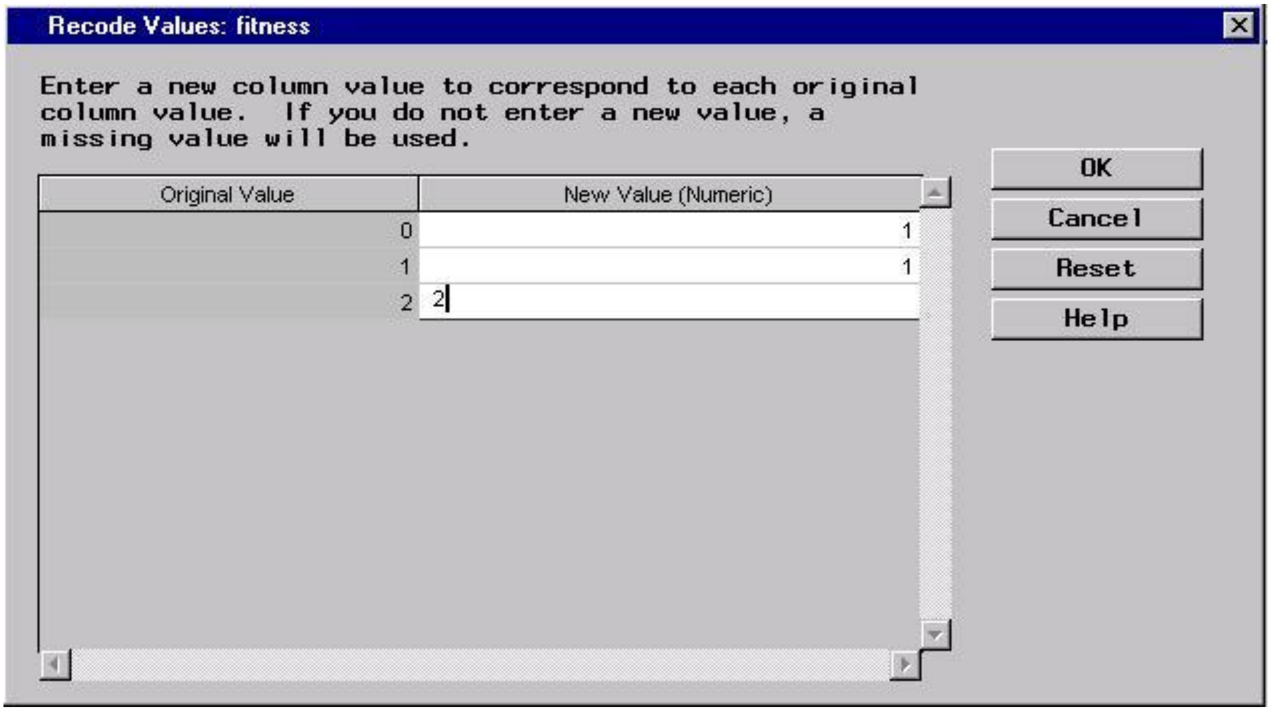
Recode Values Information: fitness

Column to recode: group

New column name: group_recoded

New column type: ● Numeric ○ Character

Use the *Column to Recode* box to select the variable you wish to recode. To do this, click on arrow on the right side of the box to obtain a list of the variables in your dataset. In the above example, *group* was selected from this list. Next, you can assign a name to your new variable by typing that name in the *New column name* box or use the default name assigned by SAS as shown above, where the new variable will be given the name *group_recoded*. Once this has been accomplished, click **OK** to define the values that you are recoding. When you do so, you will obtain the following dialog box:



Enter a new column value to correspond to each original column value. If you do not enter a new value, a missing value will be used.

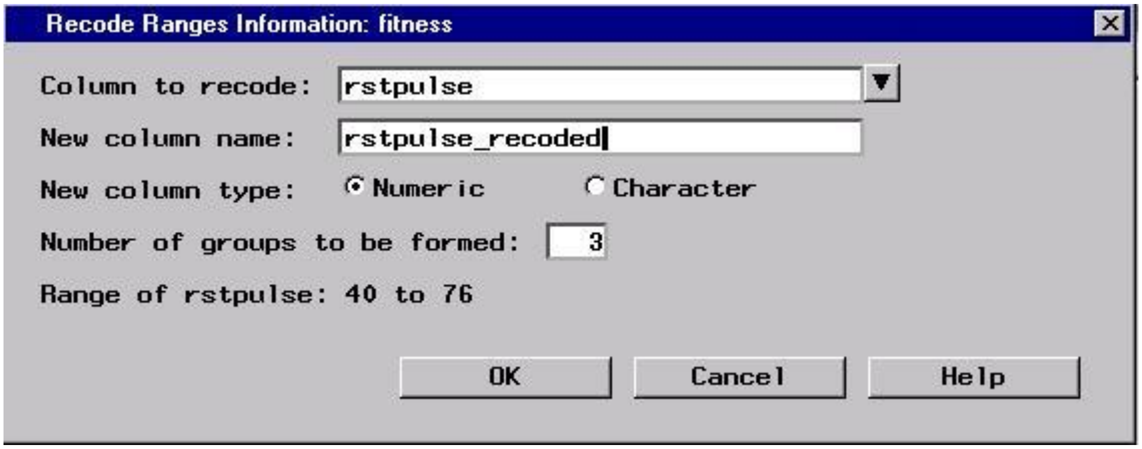
Original Value	New Value (Numeric)
0	1
1	1
2	2

Each of the existing values is listed in the *Original Value* column. You supply the recoded values in the column labeled *New Value (Numeric)*. In the above example, the 0 and 1 levels of *group* were collapsed by assigning them both the value 1, and the level 2 was assigned the value 2 in the *New Value (Numeric)* column, remaining the same. After the recoded values have been assigned, click **OK** to perform the recode transformation.

You may also have situations where you wish to convert a range of values into discrete values. For example, people could be categorized in our example dataset as having a low, medium, or high resting pulse rate. To do this, select the *Recode Ranges* option from the *Transform* submenu of the *Data* menu:

Data
Transform
Recode Ranges...

By doing so, you will obtain the following dialog box:



Recode Ranges Information: fitness

Column to recode: rstpulse

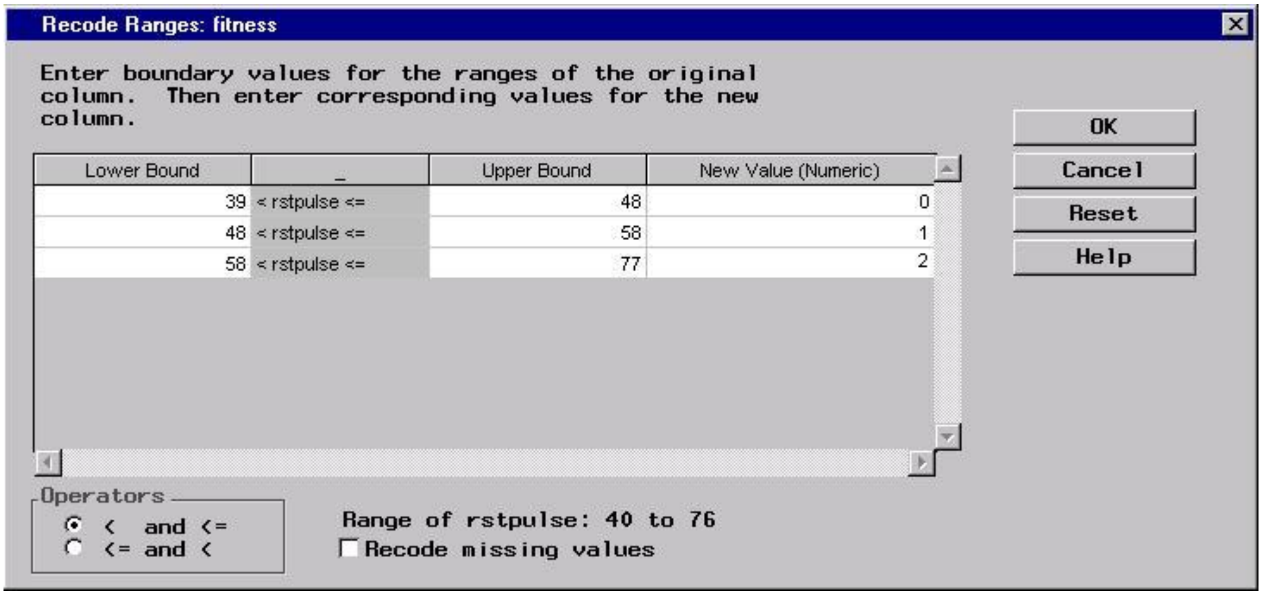
New column name: rstpulse_recoded

New column type: ● Numeric ○ Character

Number of groups to be formed: 3

Range of rstpulse: 40 to 76

As in the *Recode Values* dialog box, you begin by selecting the variable that you wish to recode by using the drop-down menu in the *Column to recode* box. In the example above, the variable selected is *rstpulse.* Again, you will use the new variable name supplied by SAS, *rstpulse_recoded.* In the *Number of groups to be formed* box, you specify the number of levels that your new variable will have. In this example, the number is 3, as the goal is to create groups that we define as being low, medium, and high in their pulse rate. After filling in these input boxes, click **OK** to specify the ranges of values to be recoded and the new values that these ranges will take. By doing so, you will obtain the following dialog box:



Recode Ranges: fitness

Enter boundary values for the ranges of the original column. Then enter corresponding values for the new column.

Lower Bound	_	Upper Bound	New Value (Numeric)
39	< rstpulse <=	48	0
48	< rstpulse <=	58	1
58	< rstpulse <=	77	2

Operators
● < and <=
○ <= and <

Range of rstpulse: 40 to 76

□ Recode missing values

For each level of the new variable, three pieces of information are necessary: the *Lower Bound*, which is the low value in the range; the *Upper Bound*, which is the high value in the range; and *NewValue*, which is the value assigned to each level in the new variable. In the above example, it can be seen that for the low pulse rate group, the low value is 39 and the high value is 48 and this group is assigned a new value of 0 in the new variable, *rstpulse_recoded*. After filling in these three pieces of information for each level of the new variable, click **OK** and the new variable will be calculated and will appear in the dataset.

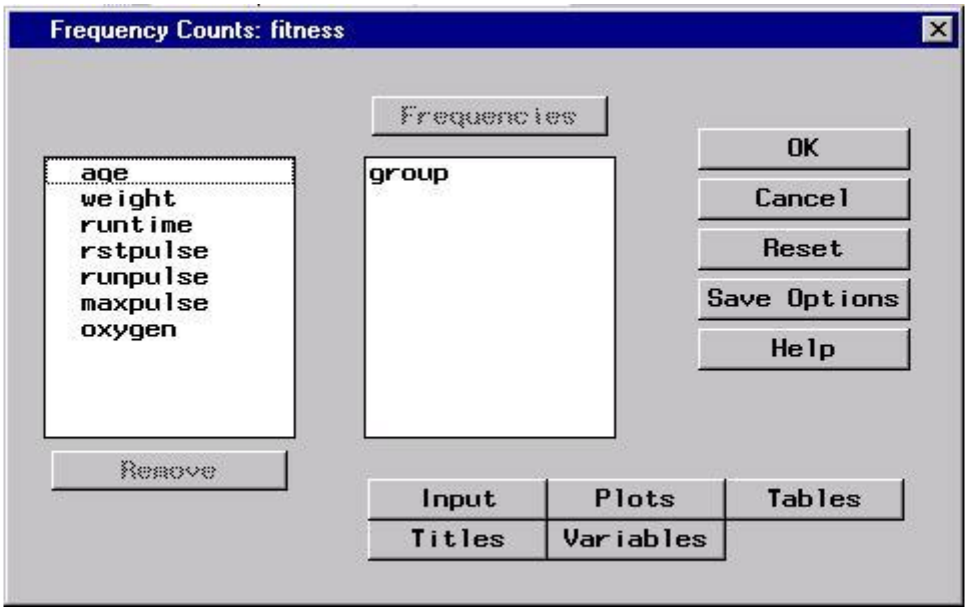
Section 6: Descriptive Statistics

6.1 Frequencies

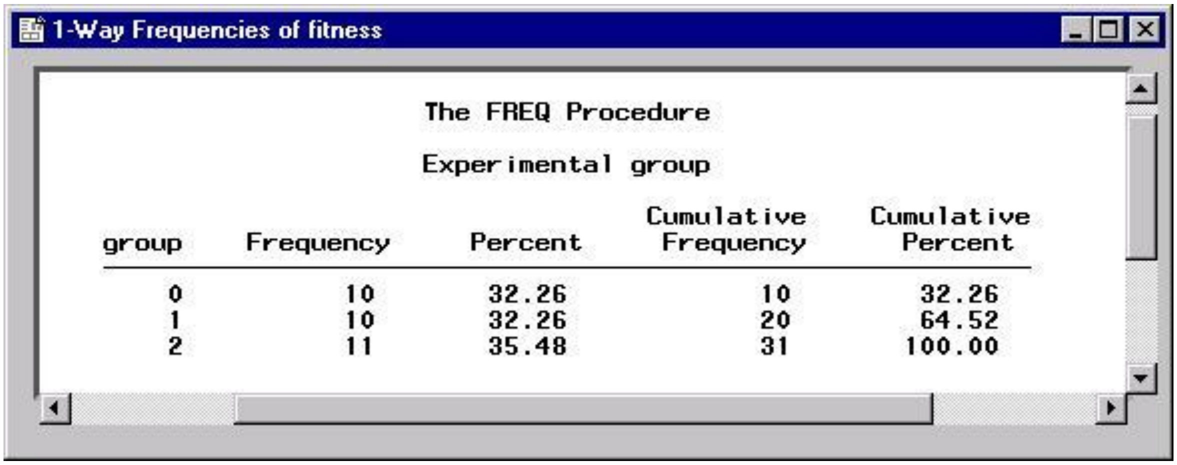
One common descriptive analysis of data is to obtain the counts of cases within levels of a variable. For example, a researcher may wish to obtain the numbers of males and females in a dataset or the number of cases in each experimental group in a dataset. To obtain frequency counts, use the *Frequency Counts* menu item in the *Descriptive* submenu of the *Statistics* menu:

Statistics
Descriptive
Frequency Counts...

Selecting this menu item will produce the following dialog box:

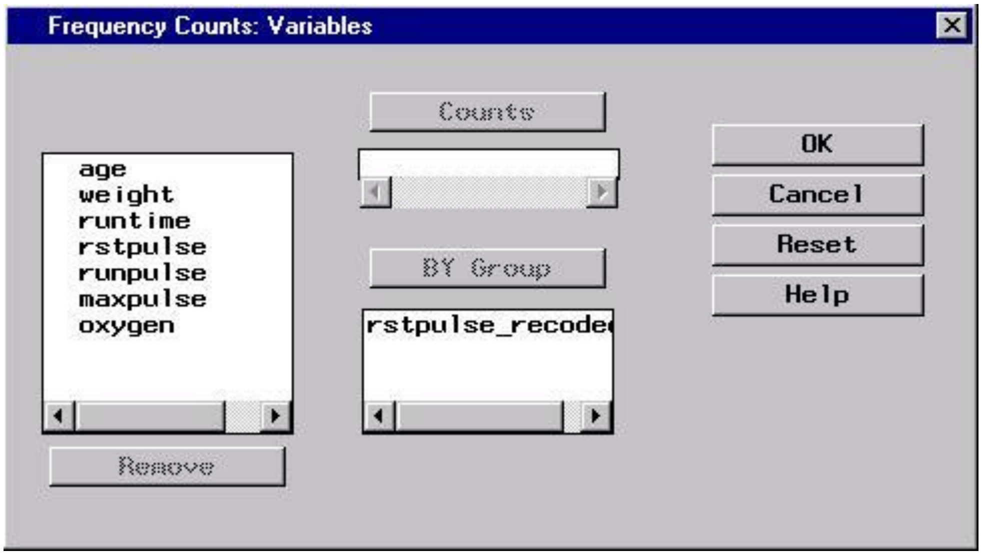


To obtain frequencies on a single variable, simply double-click on the variable name for which you wish to obtain frequencies and that variable name will move into the box to the right of the variable list. Clicking **OK** now will produce counts of cases in each group as shown below:

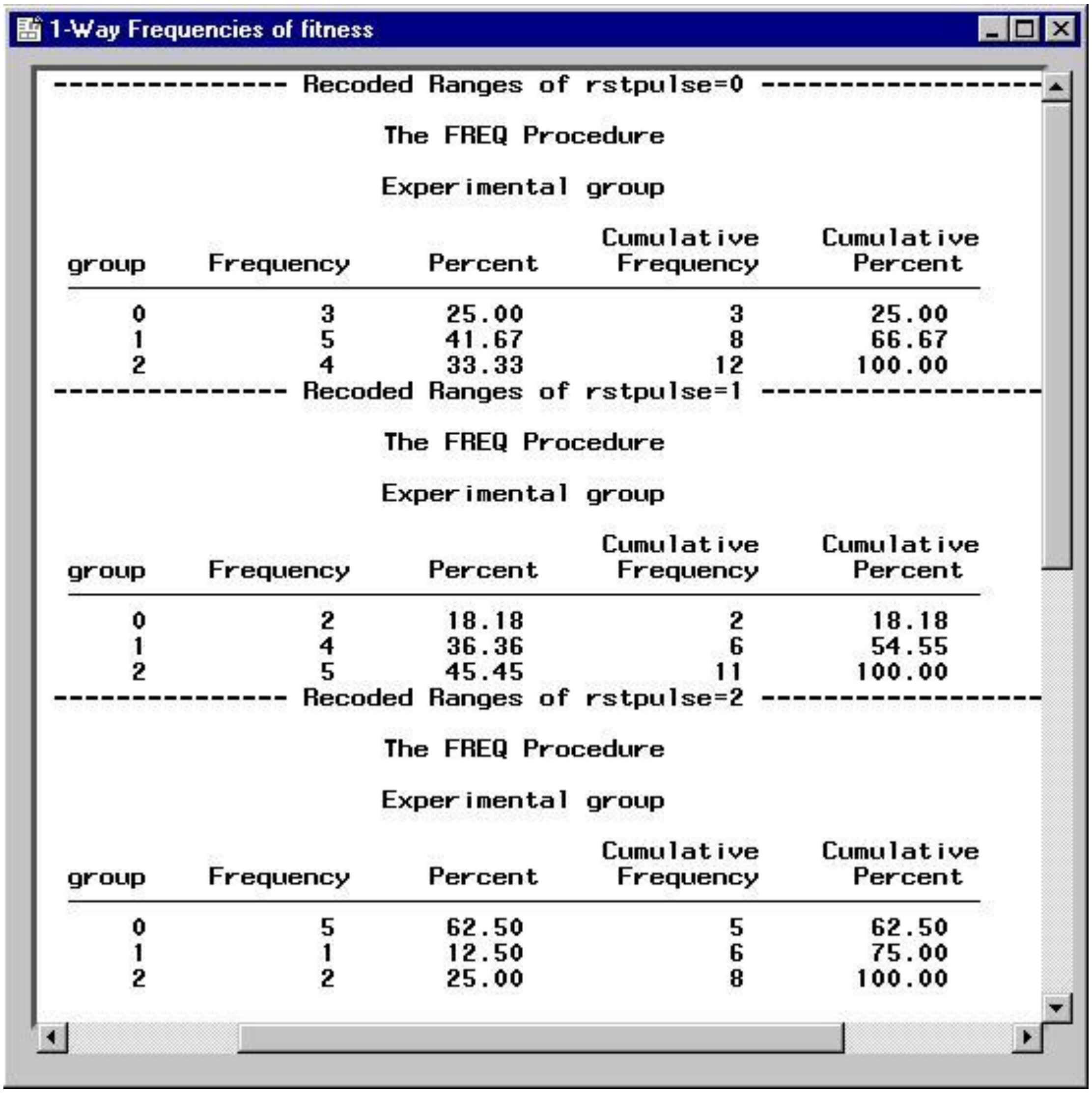


```
                    The FREQ Procedure

                    Experimental group

                                           Cumulative    Cumulative
group    Frequency     Percent    Frequency      Percent
-------------------------------------------------------------------
    0           10       32.26           10        32.26
    1           10       32.26           20        64.52
    2           11       35.48           31       100.00
```

You may also want to see how two groups are related to each other. For example, the variable *rstpulse_recode* that was created in the Recode section of this document contains three groups of pulse rates: low, medium, and high. You could see how many people within each of the experimental groups fall into each of these three levels. To obtain this analysis, select click the **Variables** button in the main *Frequencies* dialog box to obtain the following dialog box:



Here, you can define a *By* variable, which is a variable by which *group* will be crossed to obtain the numbers of people in the low, medium, and high pulse rate categories within each level of *group*. To place a variable in the *By Group* box, first click on that variable's name, then click **By Group** to place that variable in the box under that button.



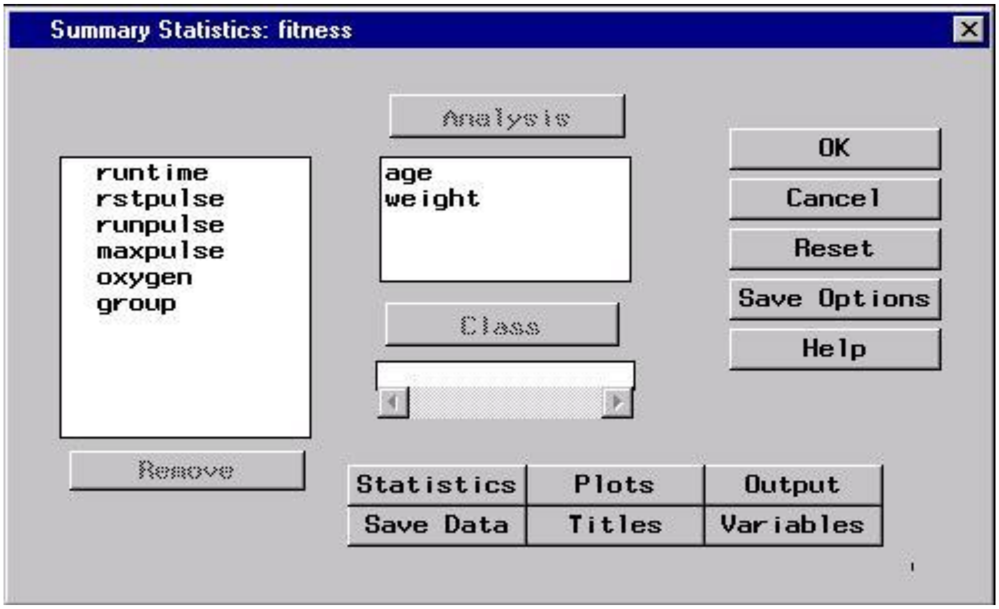
```
--------------- Recoded Ranges of rstpulse=0 ------------------

                    The FREQ Procedure

                    Experimental group

                                       Cumulative    Cumulative
group    Frequency     Percent     Frequency       Percent
--------------------------------------------------------------
    0           3       25.00             3         25.00
    1           5       41.67             8         66.67
    2           4       33.33            12        100.00
--------------- Recoded Ranges of rstpulse=1 ------------------

                    The FREQ Procedure

                    Experimental group

                                       Cumulative    Cumulative
group    Frequency     Percent     Frequency       Percent
--------------------------------------------------------------
    0           2       18.18             2         18.18
    1           4       36.36             6         54.55
    2           5       45.45            11        100.00
--------------- Recoded Ranges of rstpulse=2 ------------------

                    The FREQ Procedure

                    Experimental group

                                       Cumulative    Cumulative
group    Frequency     Percent     Frequency       Percent
--------------------------------------------------------------
    0           5       62.50             5         62.50
    1           1       12.50             6         75.00
    2           2       25.00             8        100.00
```

6.2 Summary Statistics

A common first step in data analysis is to obtain descriptive statistics on your sample. For example, you may wish to know how many cases are in your dataset and what their average age and weight are. Many of the most common descriptive statistics can be obtained through the *Summary Statistics* menu item in the *Descriptives* submenu of the *Statistics* menu:

Statistics
  Descriptives
    Summary Statistics

Selecting this menu item will provide you with the following dialog box:

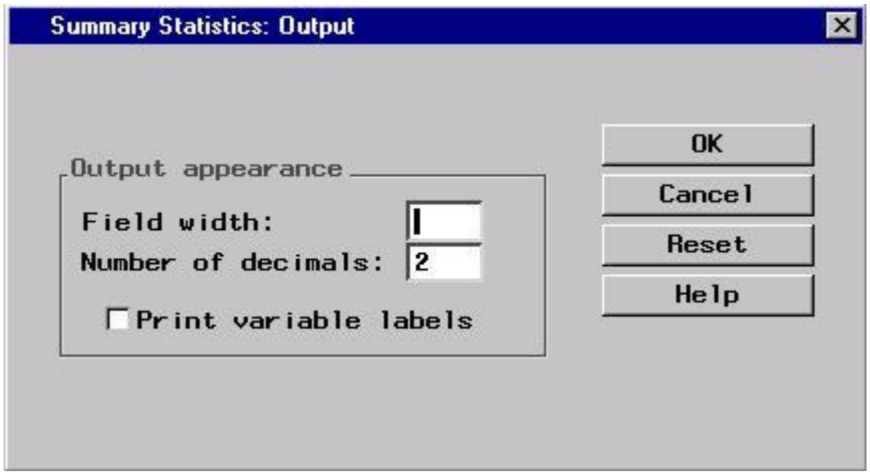


Summary Statistics: fitness

runtime
rstpulse
runpulse
maxpulse
oxygen
group

age
weight

To obtain statistics on the variables in your dataset, you first need to select the variables in which you are interested by either double-clicking on their names in the box on the left or by single-clicking on the variable name, then clicking the **Analysis** button. After selecting the variables in which you are interested, you next have several options as to which statistics and other analyses you wish to obtain. These options are available through the buttons on the bottom right side of the dialog box. Several of the buttons contain options that control the appearance of your output. The **Titles** button, for instance, controls the titles that appear at the top of pages and individual analyses as well as whether you would like the date, page numbers, and filter descriptions. Another useful button used for controlling the appearance of output is the **Output** button. Clicking this button will produce the following dialog box:



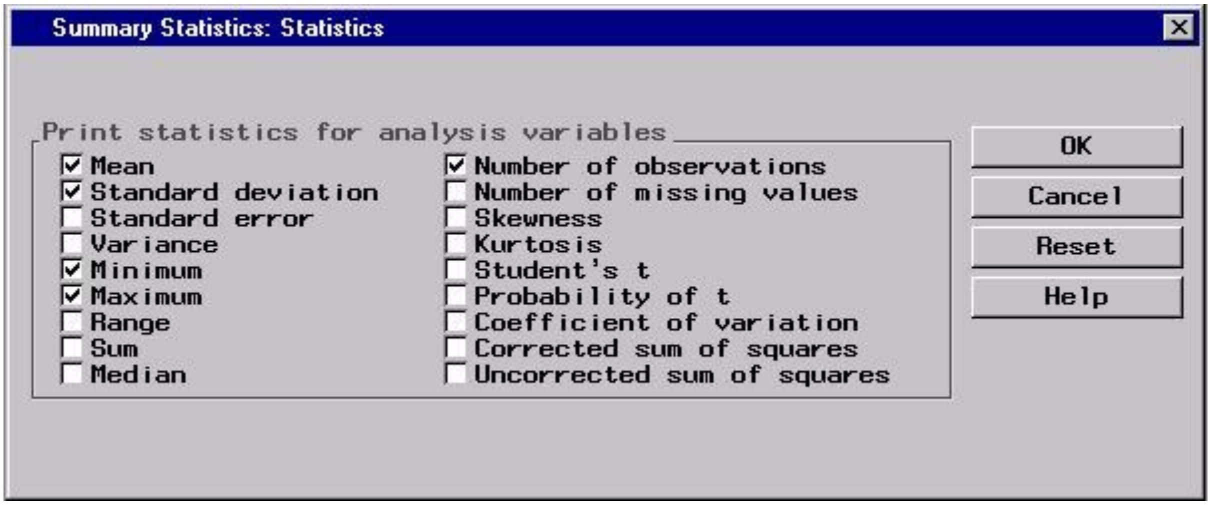
Summary Statistics: Output

Output appearance

Field width:
Number of decimals: 2
□ Print variable labels

In the above dialog box, the number of decimals has been set to 2, rather than the default setting of 7 decimal places. The *Field width* box controls the number of characters allotted to each column in an output table. If you have defined variable labels, you can have these labels displayed in your output instead of the variable's name by selecting this option.

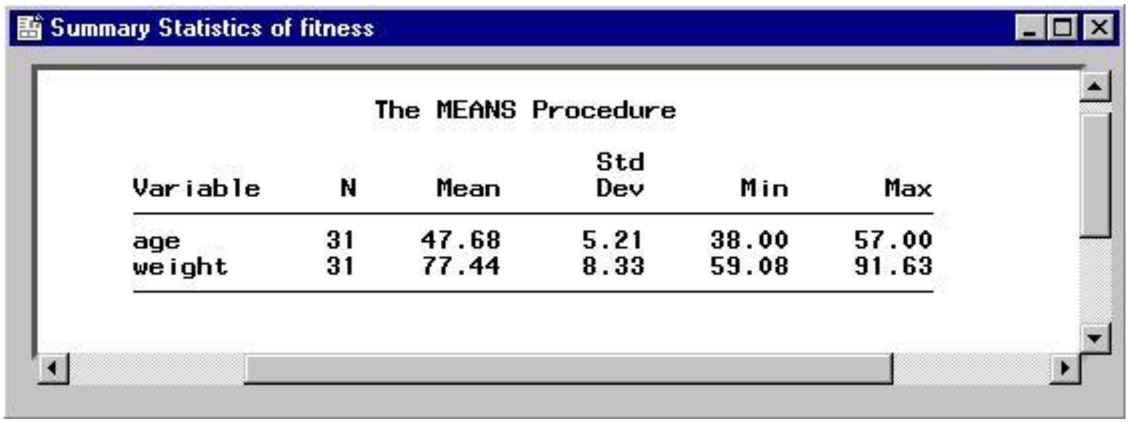
After you have specified the options controlling the appearance of your output, there are several options you might consider regarding the output you generate. For example, clicking the Statistics button will produce the following dialog box, which contains several options:



Print statistics for analysis variables

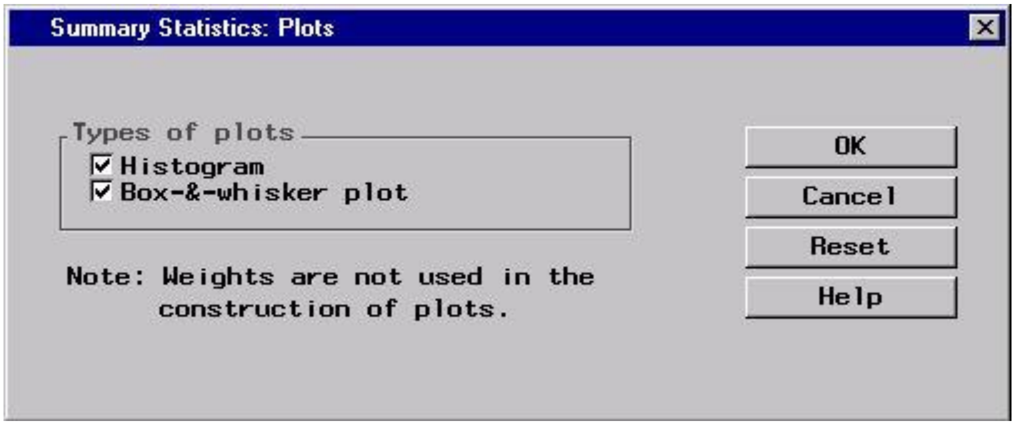
☑ Mean
☑ Standard deviation
□ Standard error
□ Variance
☑ Minimum
☑ Maximum
□ Range
□ Sum
□ Median
☑ Number of observations
□ Number of missing values
□ Skewness
□ Kurtosis
□ Student's t
□ Probability of t
□ Coefficient of variation
□ Corrected sum of squares
□ Uncorrected sum of squares

The dialog box above contains a list of all of the possible descriptive statistics that you could obtain for any given variable. Only the statistics that are checked will be computed in the output. Thus, the options shown above would produce output containing the mean, standard deviation, the minimum value, the maximum value, and the number of observations for the two selected variables, *age* and *weight*, as can be seen below:



```
                    The MEANS Procedure

                                     Std
Variable       N      Mean           Dev         Min         Max
----------------------------------------------------------------
age           31     47.68          5.21       38.00       57.00
weight        31     77.44          8.33       59.08       91.63
----------------------------------------------------------------
```

You may also desire a graphical display of your data. To obtain a list of the possible graphical displays, click on the Plots button in the main *Summary Statistics* dialog box. This will produce the following dialog box:



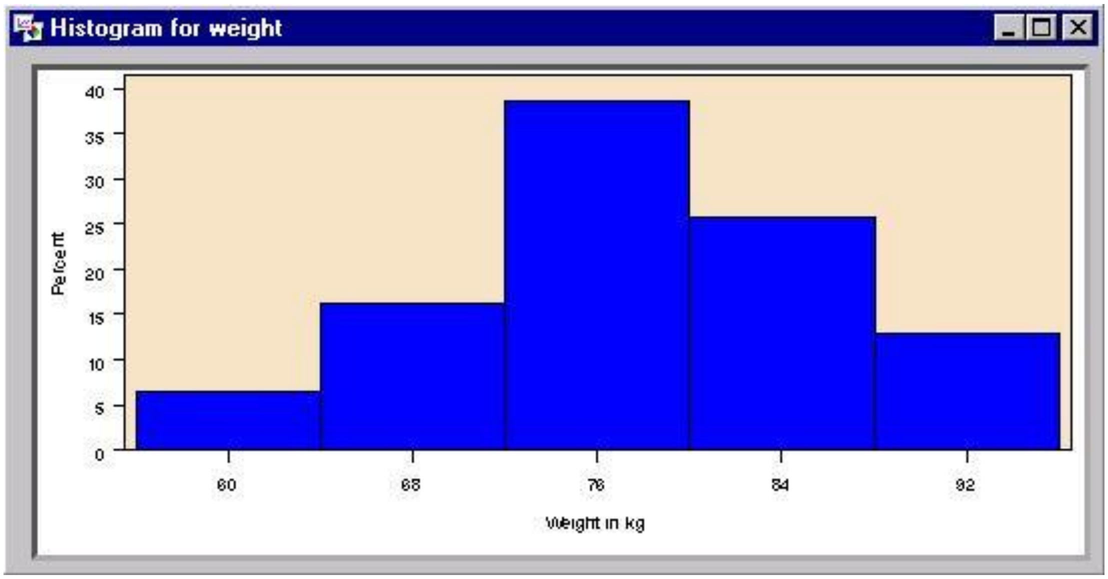
Types of plots

☑ Histogram

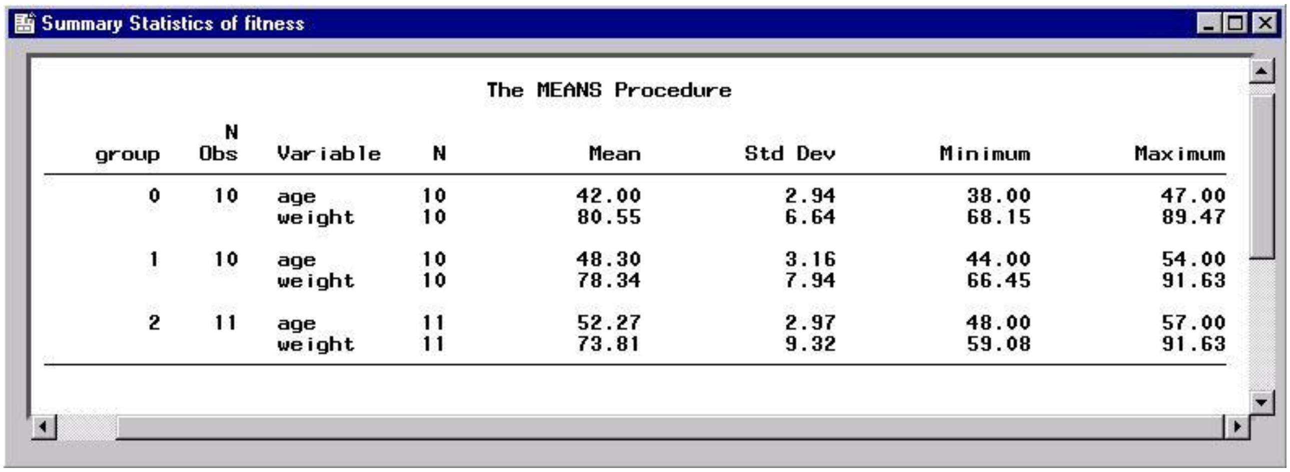
☑ Box-&-whisker plot

Note: Weights are not used in the construction of plots.

By checking both the *Histogram* and *Box-&-whisker* plot options, you will obtain both of these graphical displays for all of the variables selected in the main dialog box. Each plot for each variable will appear in a separate window in the output. For example, the histogram for *weight* will appear in a window as follows:



While the discussion of the Summary Statistics options has been performed on the entire dataset to this point, it is also possible to obtain statistics for subgroups within a dataset. For example, our sample dataset, *fitness*, contains a variable named *group* that defines three experimental groups. Because it is often interesting to see differences in groups, such as the three levels represented in the *group* variable, you may wish to obtain separate sets of analyses for each group, which would allow you to compare differences in means, standard deviations and other descriptive statistics. To do this, return to the main *Summary Statistics* dialog box. The main *Summary Statistics* box contains an input box labeled *Class*. By placing a variable name in this box, subsequent output will be displayed for each level of the variable whose name is in this box. After placing group in this input box, you could rerun any of the analyses described above and would obtain separate analyses for each level of the variable group. An example of this can be seen below where the same descriptive statistics as described above are obtained for each of the three groups defined by the *group* variable:



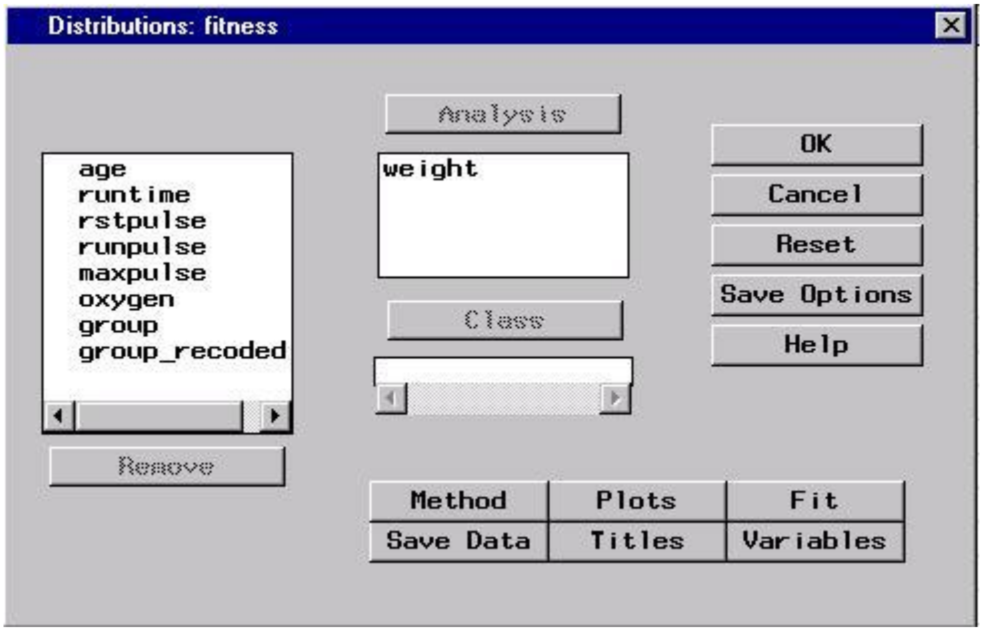
```
                              The MEANS Procedure

              N
 group      Obs    Variable      N          Mean        Std Dev        Minimum        Maximum
 --------------------------------------------------------------------------------------------
     0       10    age          10         42.00           2.94          38.00          47.00
                   weight       10         80.55           6.64          68.15          89.47

     1       10    age          10         48.30           3.16          44.00          54.00
                   weight       10         78.34           7.94          66.45          91.63

     2       11    age          11         52.27           2.97          48.00          57.00
                   weight       11         73.81           9.32          59.08          91.63
 --------------------------------------------------------------------------------------------
```

6.3 Distributions

In addition to obtaining values that describe your sample, another task that is frequently performed prior to conducting inferential statistics, such as ANOVA models and Regression models, is to examine the distribution of your variables. For example, if you were planning on conducting a regression analysis using the example dataset where you were using *age* to predict *weight*, you would want to test the assumption that your dependent variable, *weight*, is normally distributed. The Analyst application in SAS has a procedure that will test the assumption that your data are normally distributed as well as testing whether your data adhere to several other distributions. To access this procedure, select the *Distributions* menu item from the *Descriptives* submenu of the *Statistics* menu:

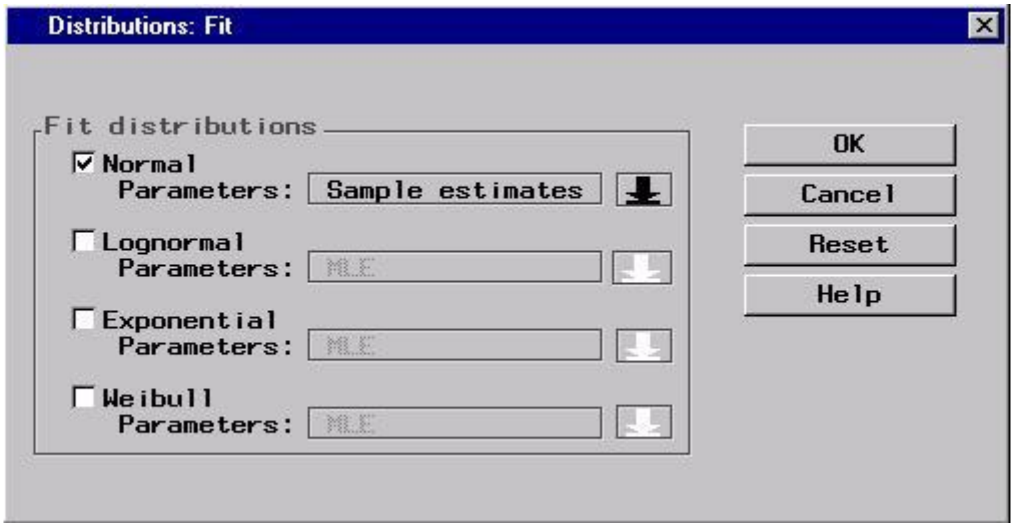
Statistics
  Descriptive
    Distributions...

Selecting this menu item will produce the following dialog box:



Analysis: weight

In the box above, the variable weight has been selected for analysis. To specify the type of distribution you wish to compare with the distribution of your data, click the **Fit** button. This will produce the following dialog box:



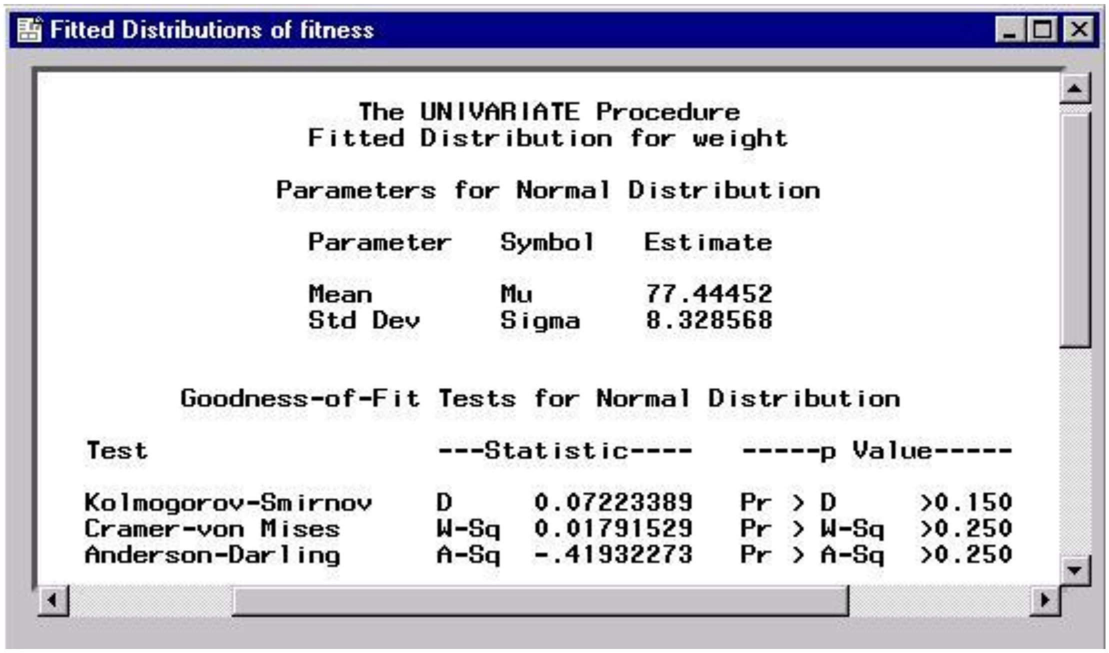
☑ Normal
Parameters: Sample estimates

□ Lognormal
Parameters: MLE

□ Exponential
Parameters: MLE

□ Weibull
Parameters: MLE

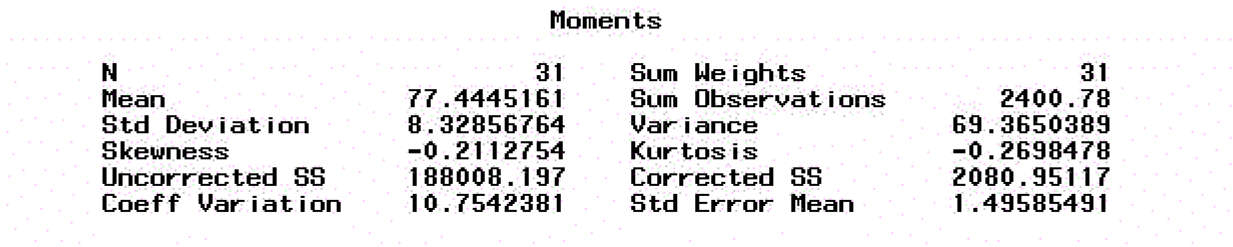
By selecting the box labeled *Normal*, you will obtain tests of the hypothesis that the variable(s) you selected in the main dialog box are normally distributed. In addition to selecting the type of distribution you wish to compare with your sample data, you can also specify parameters by clicking on the arrow to the left of the *Parameters* box. The default is the *Sample estimates* option which uses the mean and standard deviation of your sample in estimating how well your sample adheres to a normal distribution. The other option is to specify these parameters yourself, which is an option that you can obtain by clicking onto downward facing arrow. After you have selected the type of distribution and parameters, click the **OK** button to return to the main *Distributions* dialog box. Clicking **OK** in the Distributions dialog box will run the tests specified in the above dialog box. Running the example tests as specified above will produce the following output:



```
                The UNIVARIATE Procedure
               Fitted Distribution for weight

             Parameters for Normal Distribution

               Parameter    Symbol    Estimate

               Mean         Mu        77.44452
               Std Dev      Sigma     8.328568


          Goodness-of-Fit Tests for Normal Distribution

   Test                  ---Statistic----   -----p Value-----

   Kolmogorov-Smirnov    D      0.07223389   Pr > D      >0.150
   Cramer-von Mises      W-Sq   0.01791529   Pr > W-Sq   >0.250
   Anderson-Darling      A-Sq   -.41932273   Pr > A-Sq   >0.250
```

The output above indicates that the distribution of weight is not significantly different from a normal distribution with a mean of 77.44 and a standard deviation of 8.3. This conclusion is derived from the fact that none of the test statistics have a p value smaller than.05. A p value smaller than smaller than .05 would indicate that the observed distribution of the sample would occur fewer than five times in a hundred if the sample were drawn from a population that really was normally distributed. Thus, a failure to reject the null hypothesis indicates that the variable being tested is close enough to a normal distribution to meet the normality assumption that is a common assumption for many inferential statistics.

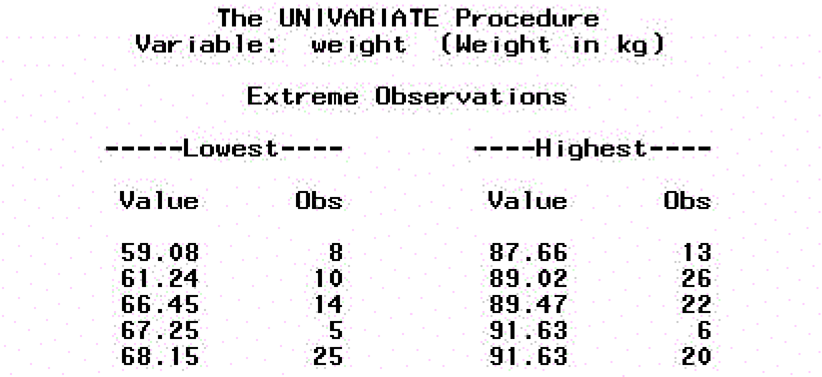
In addition to the *Fitted Distribution* output you obtain by running the above procedure, you will also have output labeled *Moments and Quartiles*. This output also contains several statistics that are useful for understanding your data and can help determine whether you need to transform your data or remove outliers from the data. Some of the statistics shown in the *Moments* section below are useful for understanding the shape of your distribution:



```
                                  Moments

N                              31    Sum Weights                 31
Mean                   77.4445161    Sum Observations       2400.78
Std Deviation          8.32856764    Variance            69.3650389
Skewness               -0.2112754    Kurtosis            -0.2698478
Uncorrected SS         188008.197    Corrected SS        2080.95117
Coeff Variation        10.7542381    Std Error Mean      1.49585491
```

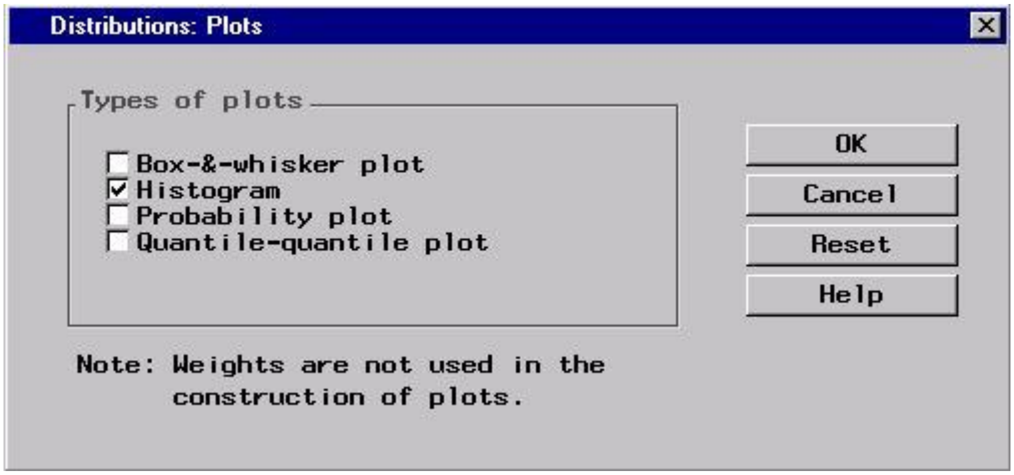
Skewness refers to whether the distribution of the variable you are analyzing is symmetrically distributed. A normal distribution has a skewness value of zero because there are equal numbers of cases on either side of the midpoint of the distribution and therefore there is no skewness to a normal distribution. In contrast, a negative distribution possesses skewness values that are disproportionately on the small side of the midpoint; the opposite is true of positively skewed distributions. *Kurtosis* refers to the peakedness of your distribution. Like skewness, a normal distribution possesses a value of zero for the kurtosis statistic. Positive values represent a high degree of peakedness, whereas negative values indicate that the distribution is flat and has too many cases in the tails of the distribution. The values shown in the above output indicate that both the skewness and kurtosis are close to zero, indicating that weight is nearly normally distributed, as was previously determined by the tests for normality discussed above.

Another part of the *Moments and Quartiles* output that is useful for screening your data is the section labeled *Extreme Observations*. This section identifies outliers in your data by listing the cases with the highest and lowest values. While having such a listing is useful for identifying cases that may have a disproportionate influence on your data, discarding outliers should be done with caution and should adhere to a predefined criteria or use a formula from a published statistical reference.



```
        The UNIVARIATE Procedure
     Variable:  weight  (Weight in kg)

            Extreme Observations

-----Lowest----        ----Highest----

 Value      Obs          Value      Obs

 59.08        8          87.66       13
 61.24       10          89.02       26
 66.45       14          89.47       22
 67.25        5          91.63        6
 68.15       25          91.63       20
```

After you have tested the assumption of normality or tested for other distributional properties, you may wish to visualize your data to get a better sense of how similar your sample's distribution is to a normal distribution. To do this, click on the **Plots** button in the main *Distributions* dialog box. This will produce the following dialog box:

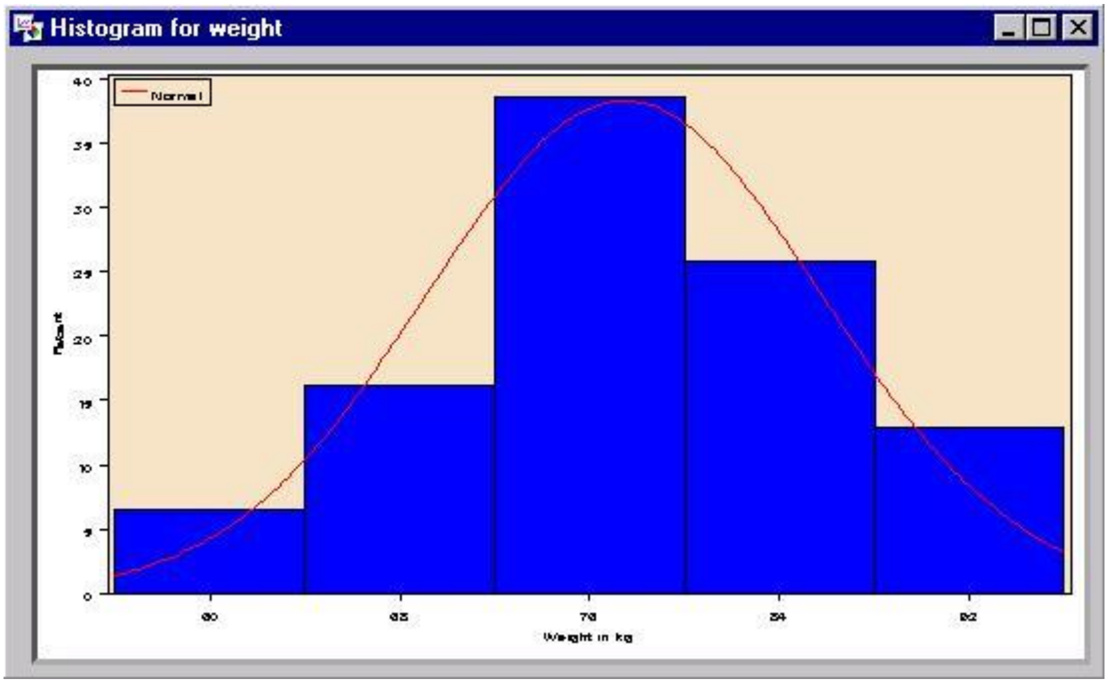


Types of plots

☐ Box-&-whisker plot
☑ Histogram
☐ Probability plot
☐ Quantile-quantile plot

Note: Weights are not used in the construction of plots.

Each of the above plots will incorporate information about distributions. The distributional information that is included in the plots is determined by your selections in *Fit* dialog box: the distributions you selected in that box will be incorporated into the selected plots. For example, by selecting the normal distribution earlier in the *Fit* dialog box, only information about the normal distribution will be included in the histogram that will be produced by selecting it in the above dialog box. Thus, the output from the example above will produce a histogram for the variable *weight* with a normal distribution with parameters identical to the example sample plotted over the histogram. To obtain this, click **OK** in the above dialog box, and then click **OK** in the main *Distributions* dialog box. Using the specifications from the above example, the following plot would be produced:



Histogram for weight
Normal
Percent
Weight in kg