Session 1

Introduction to STATA

SESSION 1: Introduction to STATA

STATA is a Statistical Analysis System designed for research professionals. The official website is http://www.stata.com/. It is an environment for manipulating and analyzing data using statistical and graphical methods. **STATA** is an integrated package — not a collection of separate modules. However, some of the statistical features are contributed and differences in language sometimes appear.

It can handle:

- Data Management
- Basic statistics
- Linear models
- Graphics
- ANOVA, etc.

Notation Conventions used throughout the course

1. These notes use either of the following conventions

Menu topic
Item
Item

or

Menu Item ➢ Item ➢ Item

to indicate that a sequence of 3 clicks are nested.

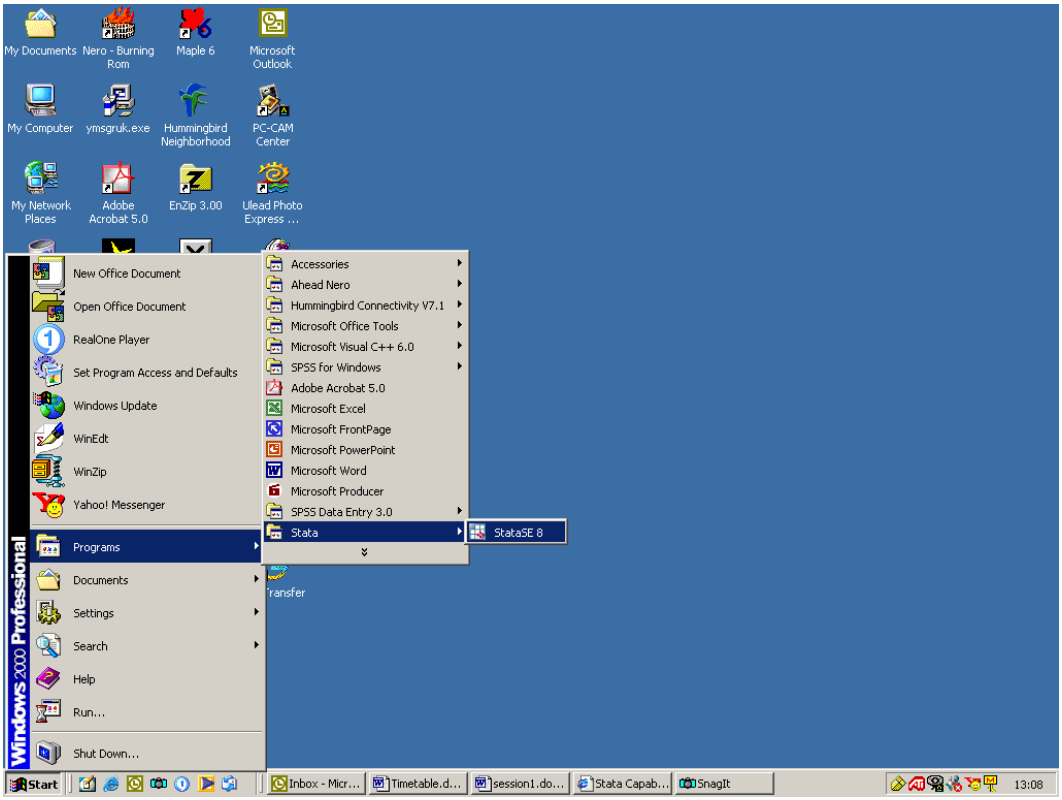
2. When following through the demonstrations, a bullet point indicates that action is required, e.g.

▪ Now do this

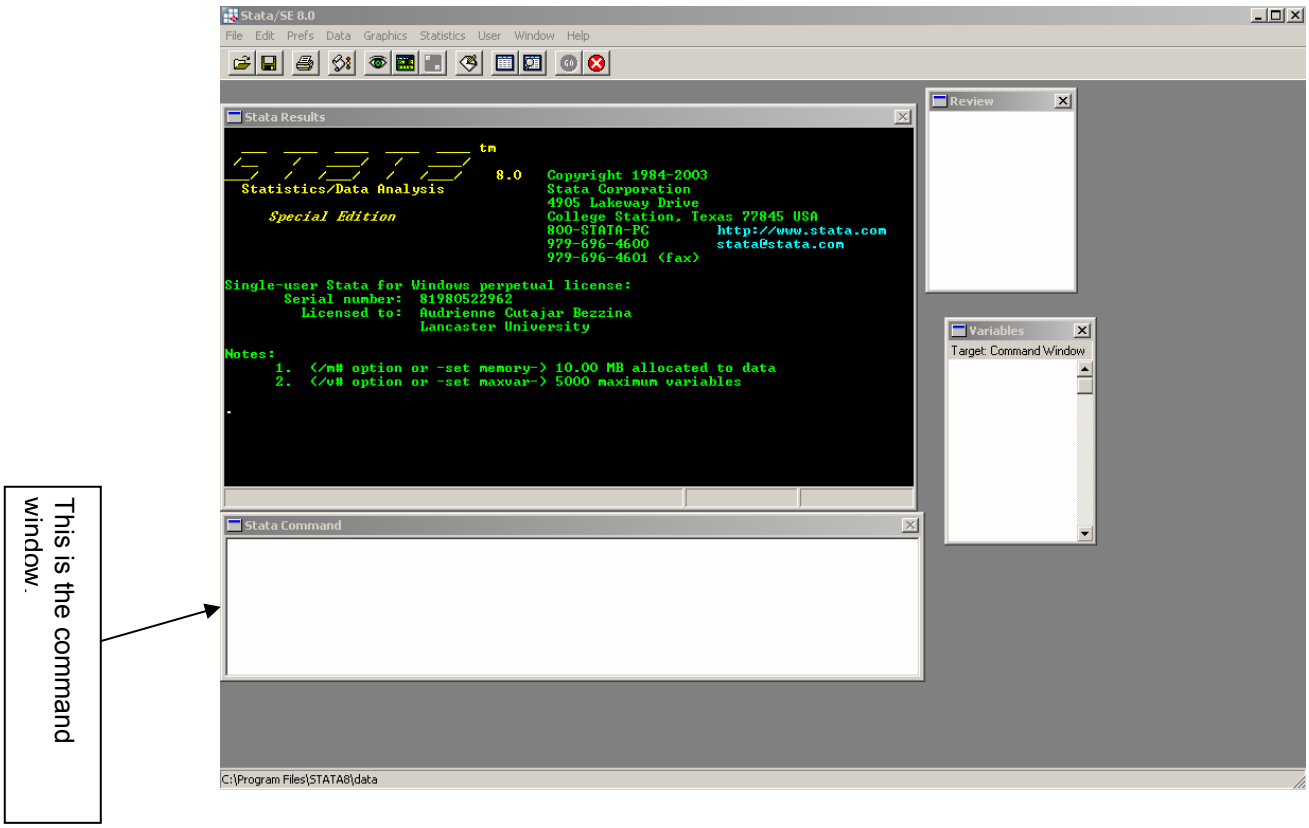
Starting STATA for Windows

If you have followed the typical installation, you have to start **STATA** by clicking on

Start ➢ Programs ➢ Stata ➢ StataSE 8 as seen below.



This will initiate the **STATA** program.



A few things to know before starting to program in **STATA** is that

1. **STATA** is case-sensitive.

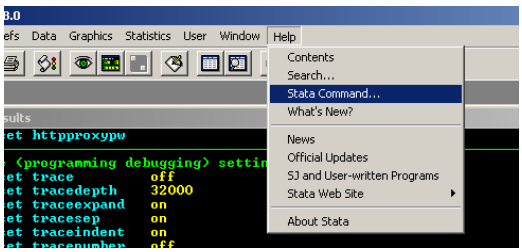
2. **.** is the **STATA** prompt.

3. **exit** is the command to exit STATA which could either be typed in the command window, or use the File menu.

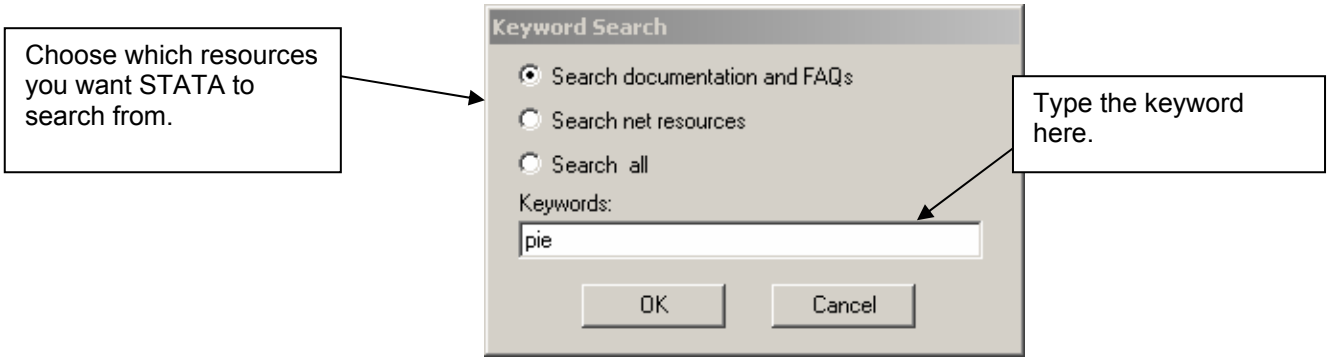
Getting Help

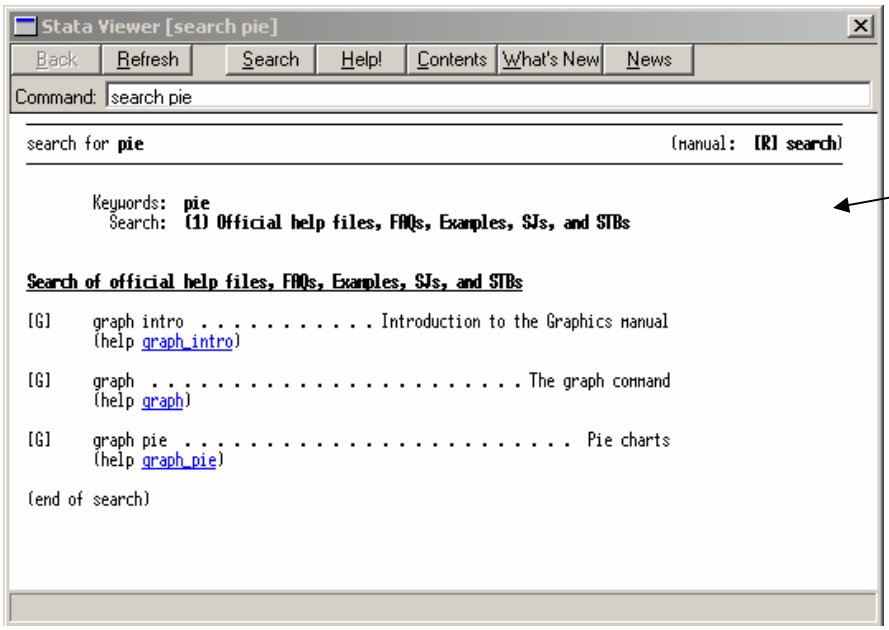
STATA has an in built help facility. If you want to search for a particular function (e.g. pie), just click

Help ➢ Search…



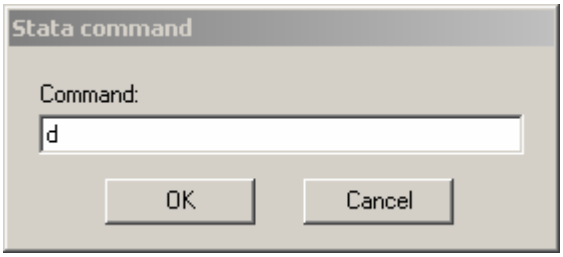
This will open the following form



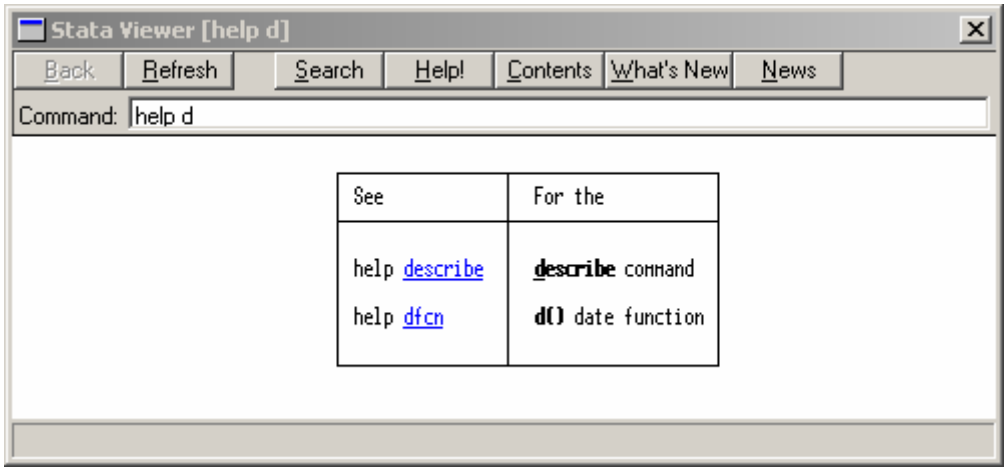


The result for 'pie' search.

If you know a particular keyword, such as'd', then you could use the **STATA command** search under **Help**,

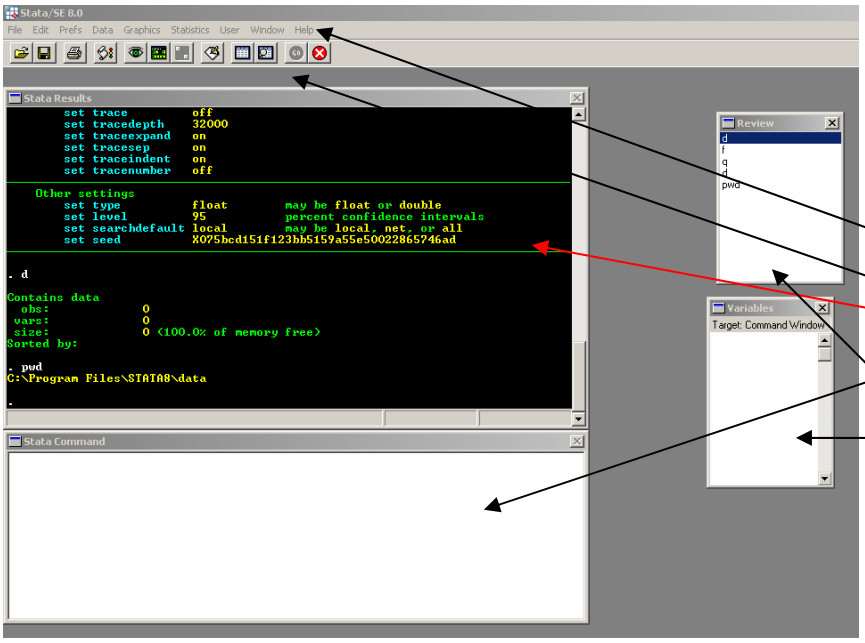


to obtain



The underlined words are hypertexts links – clicking on them will link you to the relevant page of the help information.

The STATA Console

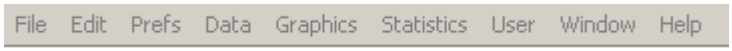


This is the **STATA** Console.

It contains

- a menu bar
- a tool bar
- a results window
- a command window
- a review window
- a variables window

The Menu bar



The menu bar lists 9 pull down menus. When you click on one of these menus, **STATA** displays a pull down menu listing the available commands. You will be using all of these at some time during the course. Some users might prefer typing the commands in the commands window, while most users will prefer using the menu interface.

However, if you are running large batch programs, then it is advisable to use the command window, together with a log file. We will see how to create a log file later on during this session.

The Toolbar

The toolbar, located just below the menu bar, provides quick and easy access to many frequently used facilities. When you put the mouse pointer on a given tool, a description of that tool appears.

Open (use). Displays the Open File dialog box for an already saved Source STATA code.

Save. Saves the current workspace.

Print results. Prints the results window.

Begin Log. Used to configure the log file.

Start Viewer. Opens the **STATA** viewer window.

Bring Results Window to Front.

Bring Graph Window to Front. Only available when graph window is open.

Do-file Editor. Opens the **STATA** Do-file Editor.

 **Data Editor**. Opens the Data Editor.

 **Data Browser**. Opens the Data Editor in browse mode.

 **Clear – more -- Condition**. Only available when data is loaded.

 **Break**. Will stop the computation that the **STATA** processor was doing.

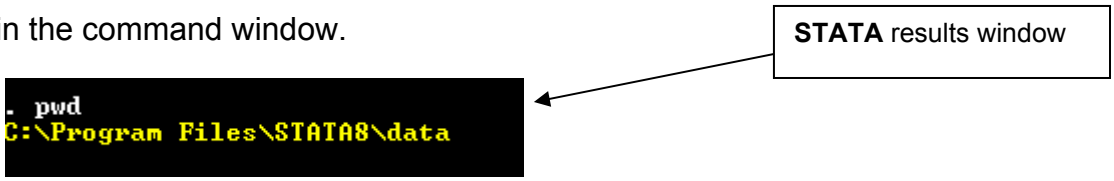
Creating a log file

For all serious work, a log file needs to be created to store the results. If this is not done, then all the results that are visible on the screen will disappear as soon as they scroll off the top of the window.

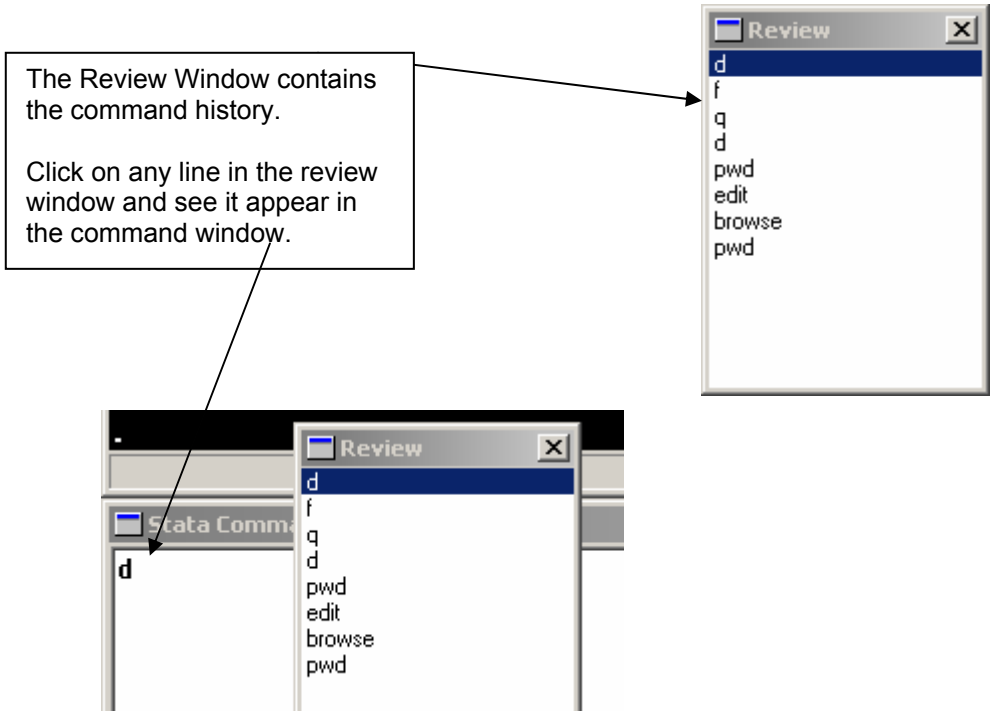
You need to first check that the working directory is correct. This is done by typing

pwd

in the command window.



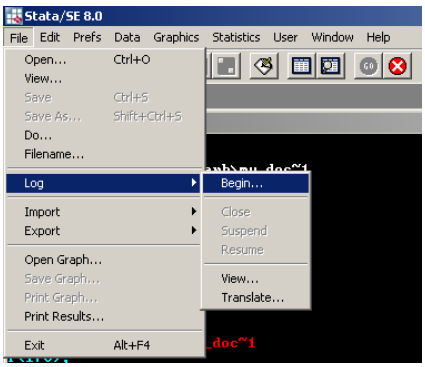
The current directory will appear in the results window.

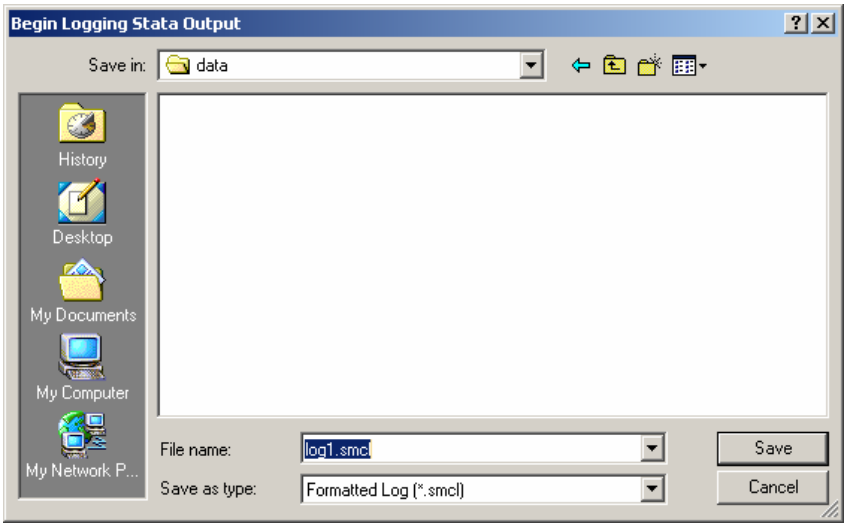


If the current directory is not correct, you can use the ***cd*** command to change it. However, this requires that you type the correct directory structure.

Another easy way would be to click on

File ➢ Log➢ Begin





Choose the directory where you want the log file to be saved.

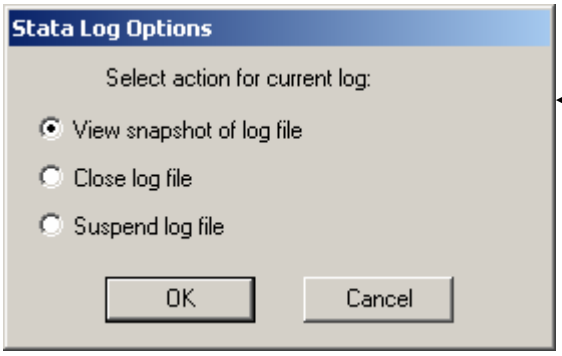
Clicking on the icon on the toolbar will generate the same result.

The **Result Window** will output the directory structure where the log file will be saved.

```
.
. log using "C:\Documents and Settings\cutajarb\My Documents\Course notes\Introdu
> ction STATA\data\log1.smcl"
-------------------------------------------------------------------------------
      log:  C:\Documents and Settings\cutajarb\My Documents\Course notes\Introdu
> ction STATA\data\log1.smcl
  log type:  smcl
 opened on:   2 May 2003, 10:31:14
```

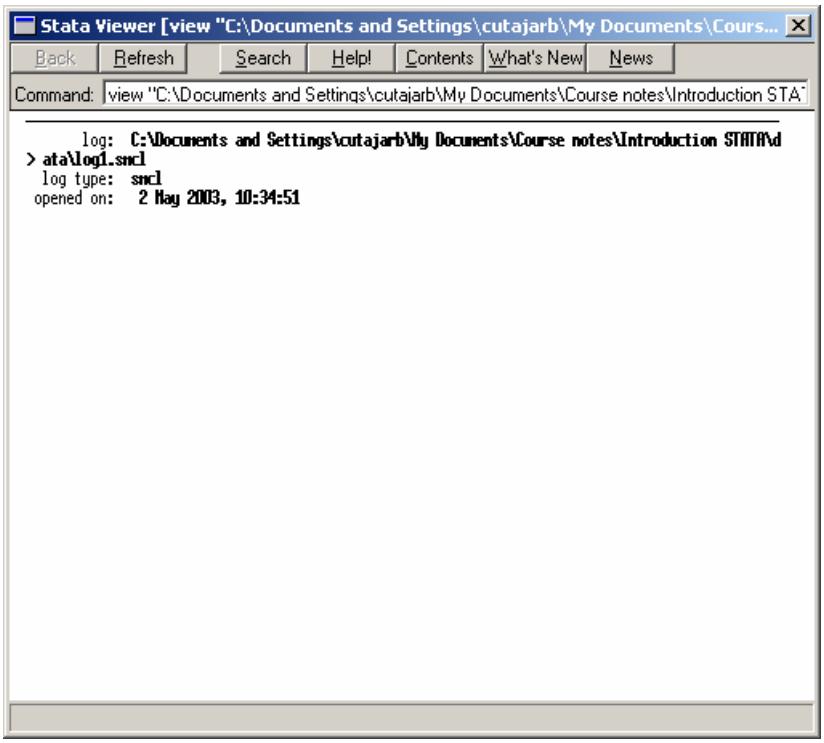
The extension used by default by the log file is ***smcl***. This is **STATA** mark up control language. This produced prettier output but it needs to be printed and opened in **STATA**. You can also choose a plain text file by changing the extension to ****.log***.

The logging can be stopped by clicking on the icon again. This will open the following form.



You can view a piece of the log file, close the log file completely or suspend until you re-start the logging process.

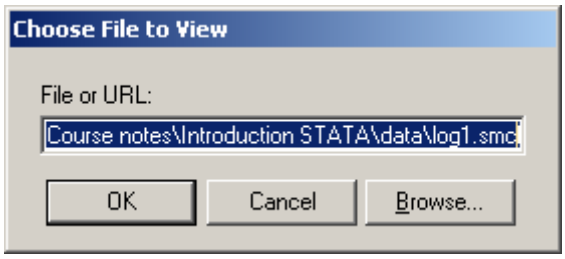
Suppose you decided to view a snapshot of the log file. The **STATA** viewer will be opened showing all the commands and results that you obtained since you start logging.



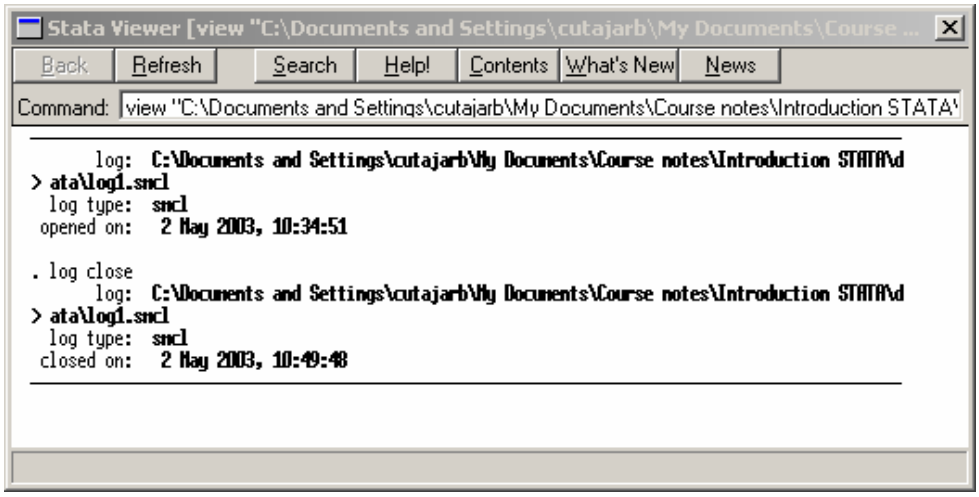
When you close the log file, you can always view it by clicking on

File ➢ Log➢ View

Type or browse for the directory and press **OK**.



The **STATA** viewer will open showing the log file.



You can then print results from the viewer by clicking on

File ➢ Print Viewer…

Practical Session 1a

1. Objective of this exercise

In this exercise you will retrieve an **STATA** data file and carry out a simple analysis. (In a future exercise, you will create your own data files from scratch.) This exercise also will allow you to become familiar with the main **STATA** windows.

Starting STATA

From the Start menu in Windows, select

Start ➢ Programs ➢ Stata ➢ StataSE 8

Suppose we wanted to load the cars data file. First of all, notice the extension with the file **cars.dta**. This shows that the file contains **STATA** data.

Click on

File ➢ Open

and choose **cars.dta**.

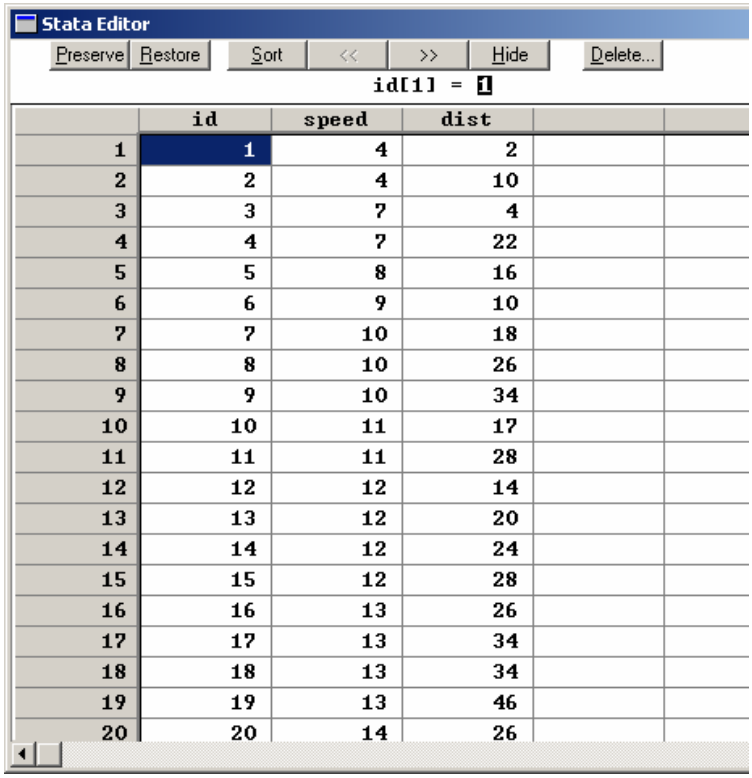
This will load a dataset containing speed and stopping distances for cars. However, no variables appear. The Results Window displays the following notification

```
.
. use "C:\Documents and Settings\cutajarb\My Documents\Course notes\Introduction STAT
> A\data\cars.dta", clear
```

indicating that the file has now been loaded into memory.

You can also notice that the Variables window contains the 3 variables that have been loaded: ***id***, ***speed*** and ***dist***.

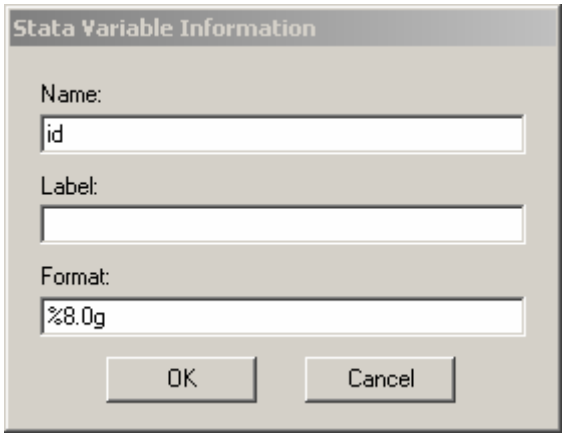
By clicking on the ***browse*** or ***edit*** button, we obtain the following:



	id	speed	dist
1	1	4	2
2	2	4	10
3	3	7	4
4	4	7	22
5	5	8	16
6	6	9	10
7	7	10	18
8	8	10	26
9	9	10	34
10	10	11	17
11	11	11	28
12	12	12	14
13	13	12	20
14	14	12	24
15	15	12	28
16	16	13	26
17	17	13	34
18	18	13	34
19	19	13	46
20	20	14	26

The **browse** function is exactly the same as the **edit** function, except browsing the data does not allow the dataset to be changed. A spreadsheet display of the data is produced.

By clicking on the variable name at the top of the **STATA Editor** window, we can see the details of the selected variable in the **STATA Variable Information**. You can choose the format of the variable, and modify its name and label as well.



To finish, click on **Preserve** and close the window.

After changing the labels, you can see the script that **STATA** has written.

```
. edit
- preserve
- label var id "id of the car"
- label var speed "speed of car"
- label var dist "distance"
- preserve
```

A simple analysis

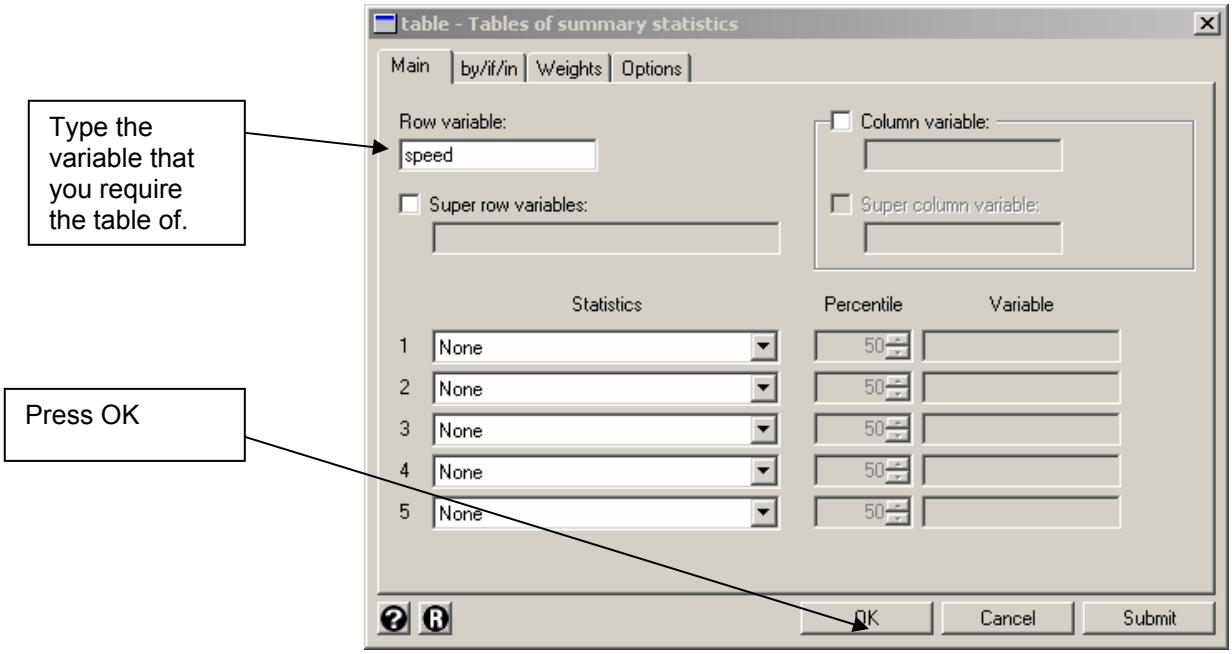
To get **STATA** to calculate a frequency count for the variable ***speed***, you need to use the command:

-table speed

Alternatively, it is easier to use the **Statistics** menu in the menu bar. Hence, click on

Statistics ➢ Summaries, Tables & Tests ➢ Table➢ Table of Summary Statistics (Table)

to obtain



```
. table speed

speed of car |      Freq.
-------------+-----------
           4 |          2
           7 |          2
           8 |          1
           9 |          1
          10 |          3
          11 |          2
          12 |          4
          13 |          4
          14 |          4
          15 |          3
          16 |          2
          17 |          3
          18 |          4
          19 |          3
          20 |          5
          22 |          1
          23 |          1
          24 |          4
          25 |          1
```

The first column of output gives the data points, while the second column gives the frequency of each data point respectively, i.e. there are 2 data points having a value of 4, 2 data points having a value of 7, etc.

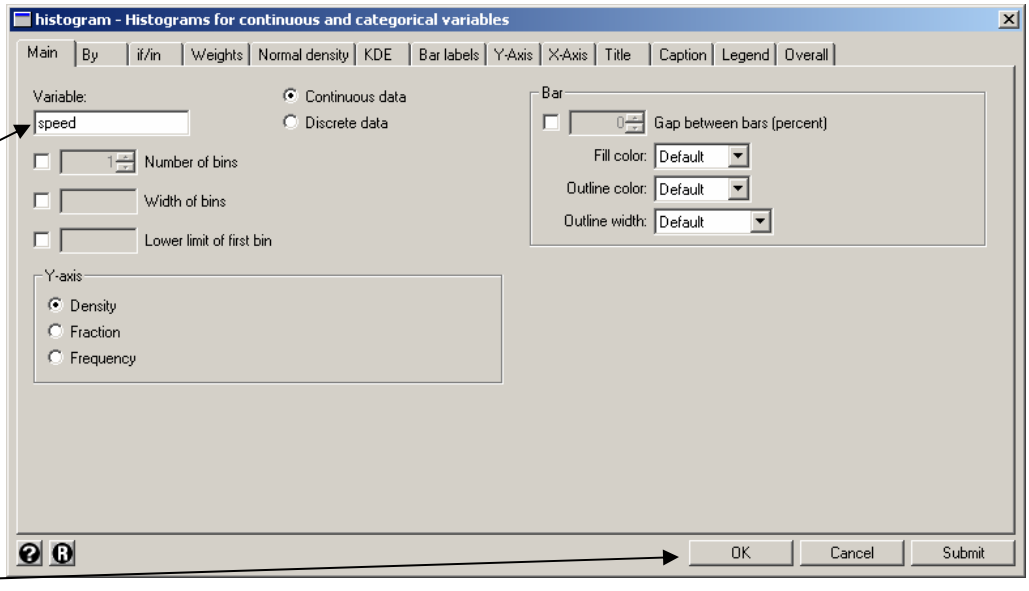
To obtain a histogram showing the above table, type

- histogram speed

or click on

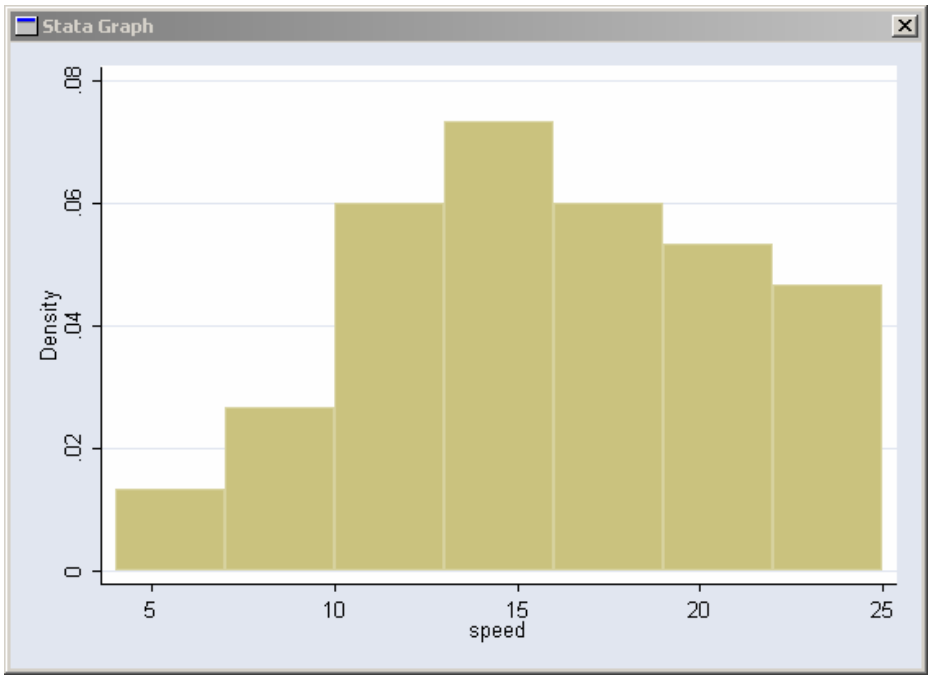
Graphics ➢ Histogram

For the time being, type ***speed*** under the variable name and ignore all the other options. These will be covered in more detail in a later session.



Click OK.

This will open a new console window showing the following graph:



There are options available how to modify this graph, but we will look at these later on.

Leaving STATA

To exit from **STATA**, click on

File ➢ Exit

2. Creating your own STATA Data File Using the Data Editor

The data

In this exercise you will create a new data set, defining your own variables and entering some data collected about 10 visiting students. The pieces of information collected were:

1. Surname of student
2. Sex of student
3. Distance travelled to the University

Once you enter the data, you can get some summary statistics about the distance travelled by the students.

The data to be used is given in the following table:

Surname	Sex	Distance
Brown	1	12
Smith	2	15
Robinson	1	93
Fligelstone	2	1
Green	1	12
Harris	2	6
Jenkins	1	25
Johnson	2	42
Frank	1	3
Stone	2	11

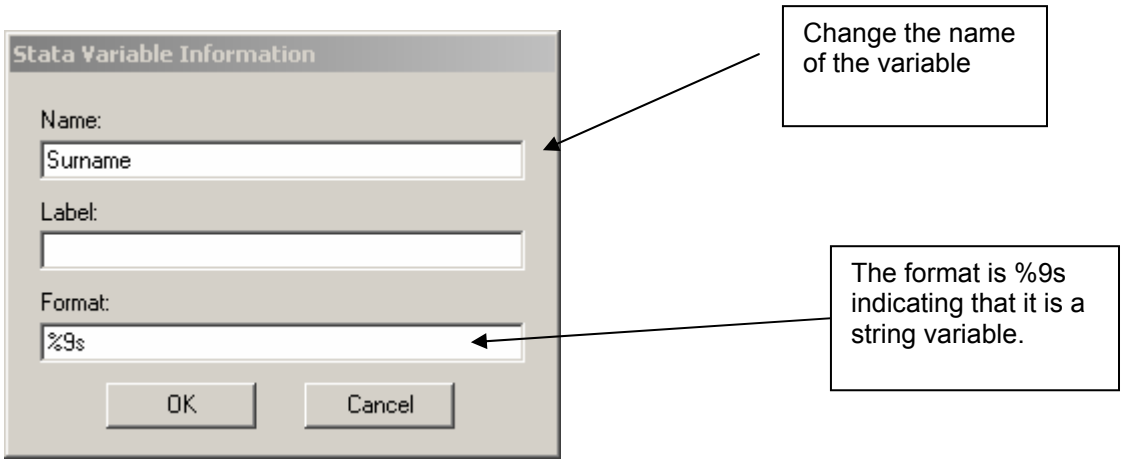
Defining the variables

Start **STATA** in the usual way by clicking on

Start ➢ Programs ➢ Stata ➢ StataSE 8

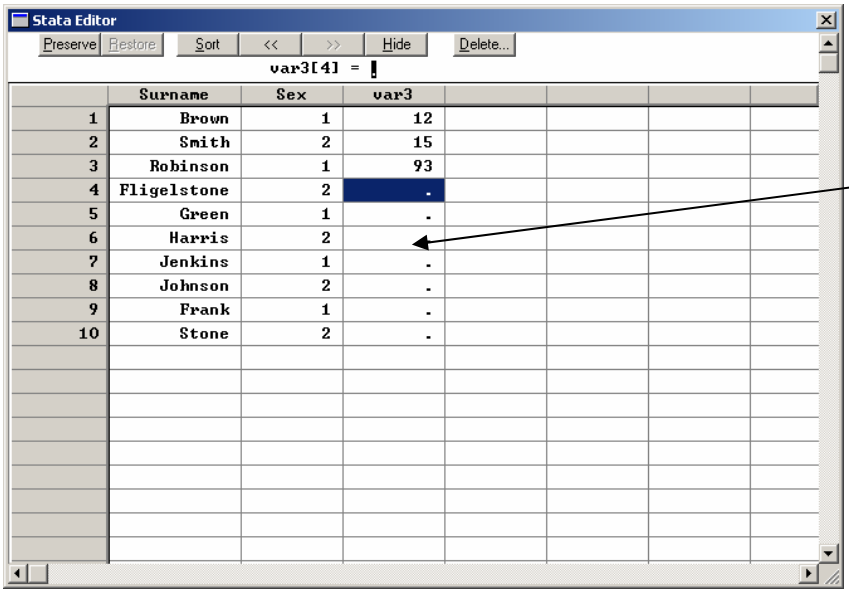
Activate the **Data Editor** Window by clicking on the  icon. Start typing **Brown** in the 1st row, 1st column.

STATA will automatically name the 1st variable as **var1**. Double click on this variable name to change this.



This allows you to enter the surnames as letters, rather than numbers. Press OK to close this box. Fill in all the data.

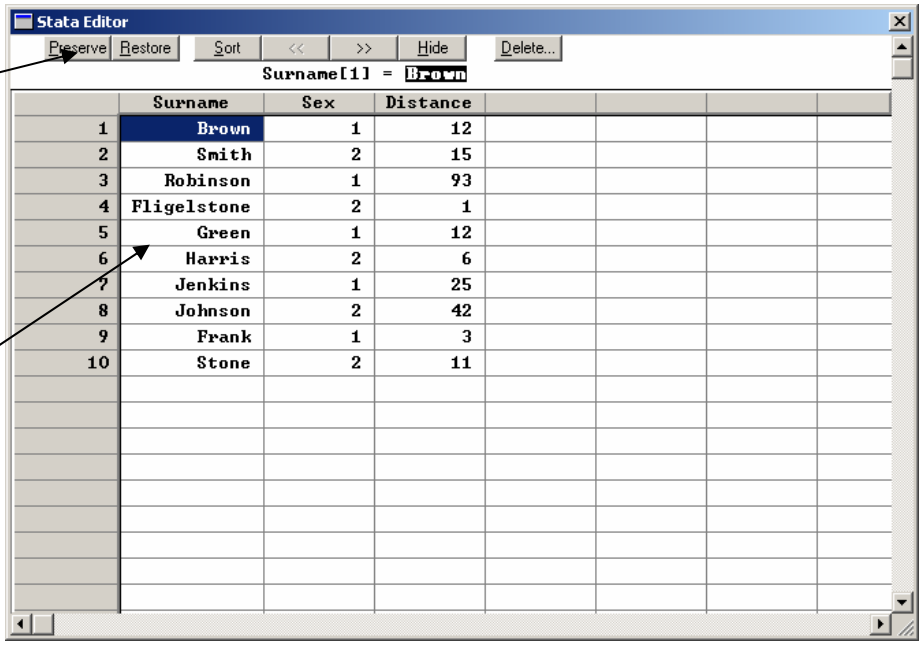
Rename the second variable as **Sex** of type numeric, and the third variable as **Distance**, also numeric.



STATA defines missing data as **-**.

Click on Preserve before closing the Editor.

The finished data set.

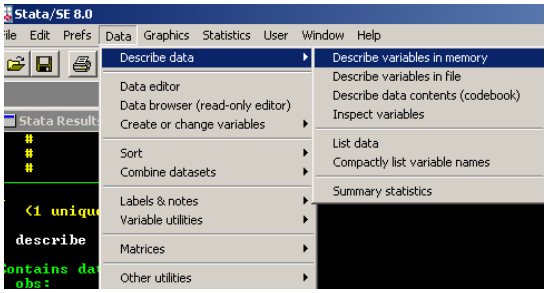


To make sure that all the variables are there and that they are in the format you need them, we can use the 'describe' command. This can be abbreviated to simply ***d***, and it will provide basic information about the file and the variables.

```
. d

Contains data
  obs:            10
 vars:             3
 size:           170 (99.9% of memory free)

              storage  display     value
variable name   type   format      label      variable label

Surname         str11  %11s
Sex             byte   %8.0g
Distance        byte   %8.0g

Sorted by:
     Note:  dataset has changed since last saved
```

The command can be also accomplished by the pull down menu system

Data ➢ Describe Data ➢ Describe Variables in Memory

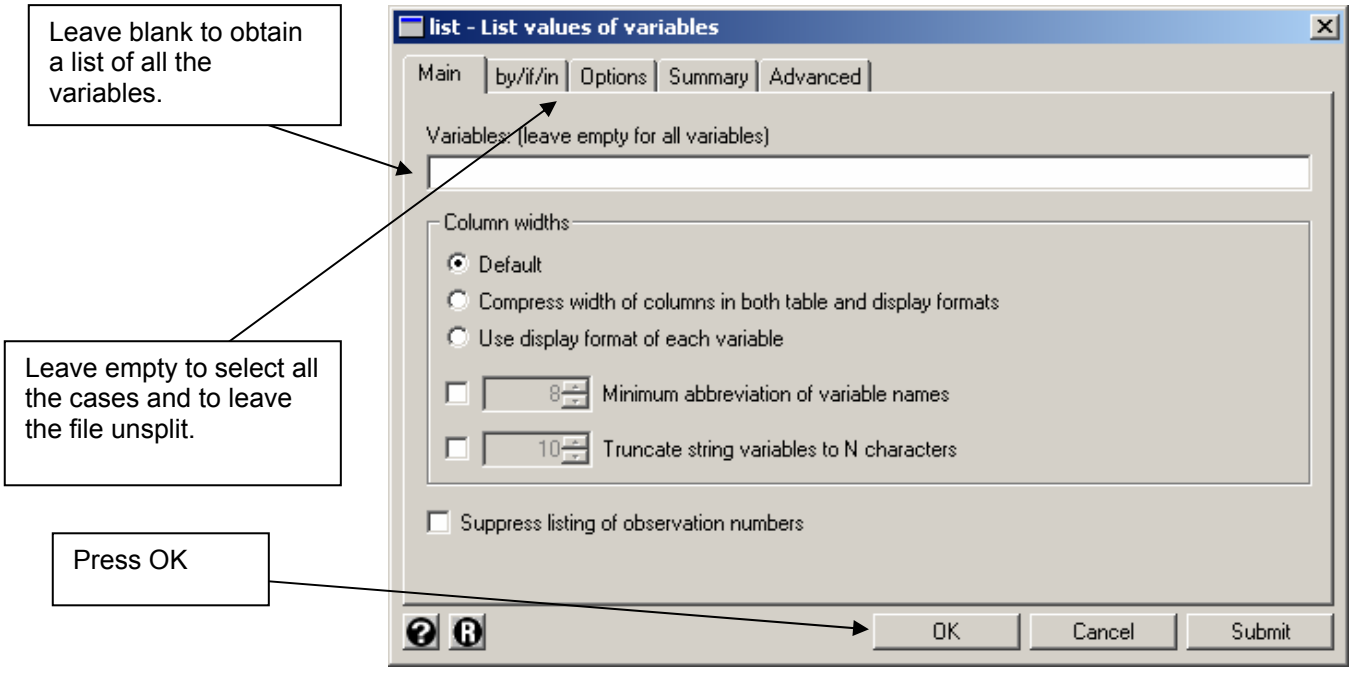


Alternatively, you can obtain other type of descriptions, from the same pull down menu system.

If you click on

Data ➢ Describe Data ➢ List Data

you would obtain the following Window.



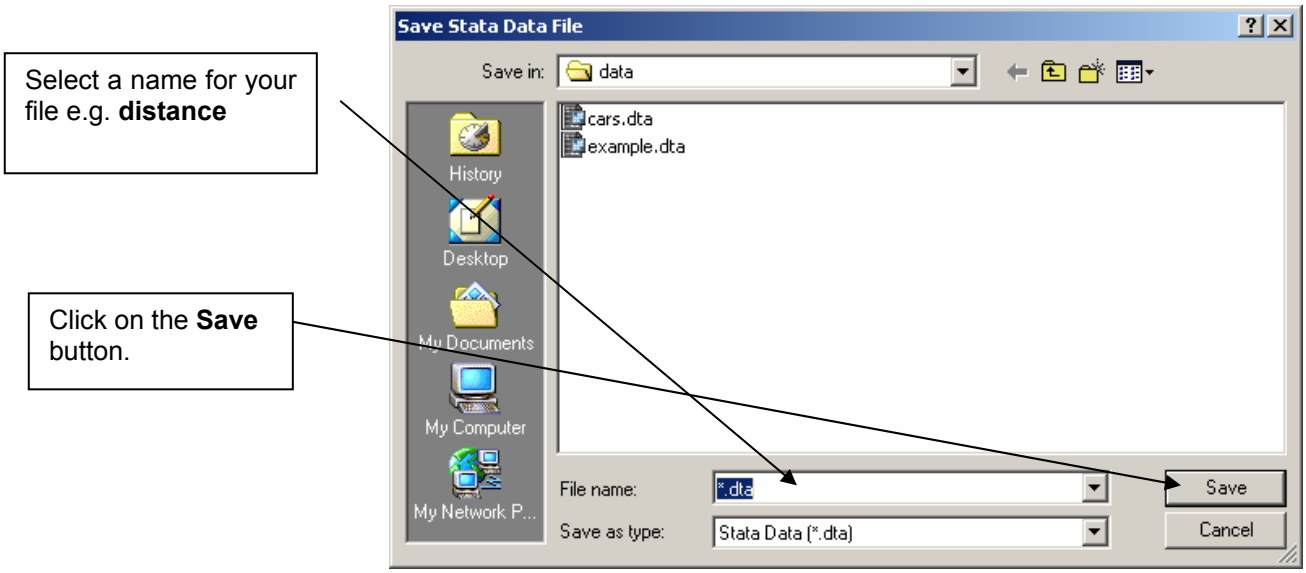
STATA produces the following list. It has increased a new column, ***_delete***. At the moment, all values under this column are 1; indicating that all rows will be considered and none are deleted.

```
. list, separator(5)

     | Surname   Sex   Distance   _delete |
  1. |       ?     1         12         1 |
  2. |       ?     2         15         1 |
  3. |       ?     1         93         1 |
  4. |       ?     2          1         1 |
  5. |       ?     1         12         1 |
  6. |       ?     2          6         1 |
  7. |       ?     1         25         1 |
  8. |       ?     2         42         1 |
  9. |       ?     1          3         1 |
 10. |       ?     2         11         1 |
```

Saving the Data

To save the new data file created, click on

File➢Save As… from the Main Menu bar.



If you have a floppy disk with you, put it into the 'a:' drive now. It is a good idea to save your work to a floppy disk rather than the hard disk of the computer for two reasons:

- You are not tied to using the same machine each time.
- Your file may be erased from the hard disk.

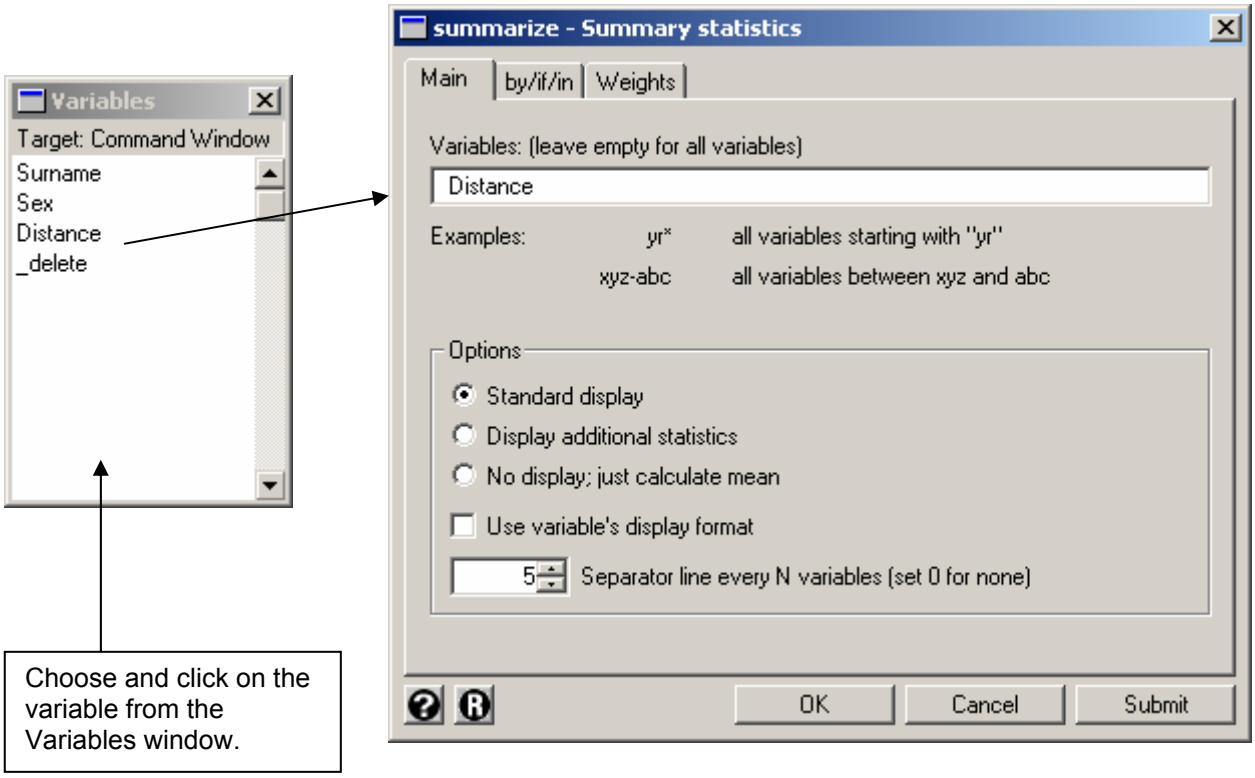
STATA will display the result of the Save file operation in the Results Window.

```
. save "C:\Documents and Settings\cutajarb\My Documents\Course notes\Introducti
> on STATA\data\distance.dta"
file C:\Documents and Settings\cutajarb\My Documents\Course notes\Introduction
> STATA\data\distance.dta saved
```

Some descriptive statistics

To find out the average distance the students travelled, click on

Statistics ➢ Summaries, Tables & Tests ➢ Summary Statistics➢ Summary Statistics



This should give the following table.

```
. summarize Distance
    Variable |        Obs        Mean    Std. Dev.       Min        Max
-------------+---------------------------------------------------------
    Distance |         10          22    27.64457          1         93
```

Note that **STATA** displayed some other summary statistics, such as the standard deviation, the number of observations, and the minimum and maximum values.

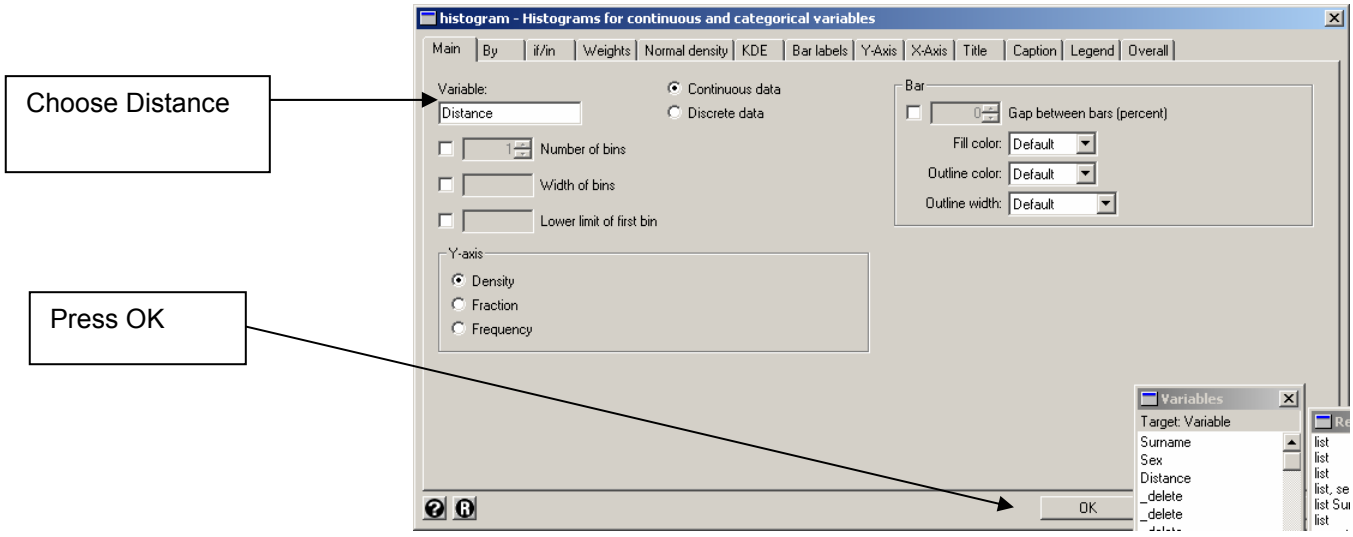
You can obtain additional statistics by choosing the 'Display additional statistics' option.

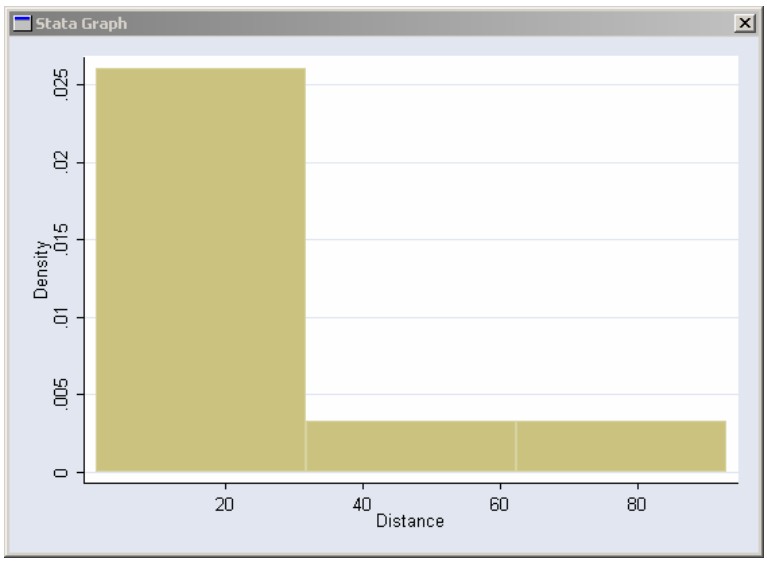
```
. summarize Distance, detail

                        Distance
-------------------------------------------------------------
      Percentiles      Smallest
 1%            1              1
 5%            1              3
10%            2              6       Obs                  10
25%            6             11       Sum of Wgt.          10

50%           12                      Mean                 22
                      Largest         Std. Dev.       27.64457
75%           25             15
90%         67.5             25       Variance        764.2222
95%           93             42       Skewness        1.896242
99%           93             93       Kurtosis        5.495847

.
```

The variance is now given to be 764.222.

Finally obtain a histogram of **Distance** by clicking on

Graphics ➢ Histogram





Clearing the memory

To remove the variables from memory, you could use the command ***clear***.

```
. clear

. d

Contains data
  obs:             0
 vars:             0
 size:             0 (100.0% of memory free)
Sorted by:
```

Reading a Raw Data File

In the previous exercise we retrieved and created **STATA** data files. These are special files that only STATA can read or create. In many instances you may want **STATA** to read a *raw* data file that has been created by using a word processor, spreadsheet or database – or files that are in ASCII (Text) format.

'Text' files can be arranged in several ways. For instance, if you have only collected information for a few variables for each person, the data could be written to the data file so that a new line is started for each person. You could also decide that **each** variable will occupy the **same** column in the data file. This is often known as **fixed format**.

1	2	3	4	5	6	7	8	9	10	11	12	Column numbers
id		age			sex		v1		v2		v3	Variable names
1		2	2		M		4		2		1	
2		4	0		F		2		3		1	Filename:
3		2	7		M		3		3		2	**example.dat**
4		3	5		M		2		2		4	
5		2	4		F		1		2		2	

This data file above is in **fixed format**. Each variable is in its own column(s) and together they take up a total of 12 columns. It is normal to go on to the next line after column 80, which is the width of most screens. Once again, each variable must be in the same location for each case. So if the variable V101 is in column 5 on the second record of data for person 1, then it must also be in that location for the next and subsequent cases.

With 300 variables we could have 6 records to an individual, for e.g.

CASE1.1 . V001, V002 - V80, V81
CASE1.2 . V82, - - - - - - - - - V101 -
CASE1.3 .- -
CASE1.4 .- -
CASE1.5 .- -

CASE1.6 .- V300
CASE2.1 . V001, V002 - V80, V81
CASE2.2 . V82,- - - - - - - - - - V101 -
...etc.

A note about variable names in STATA:

Each of the variables in your data must be given a name. When deciding upon a name it is a good idea to follow some conventions...

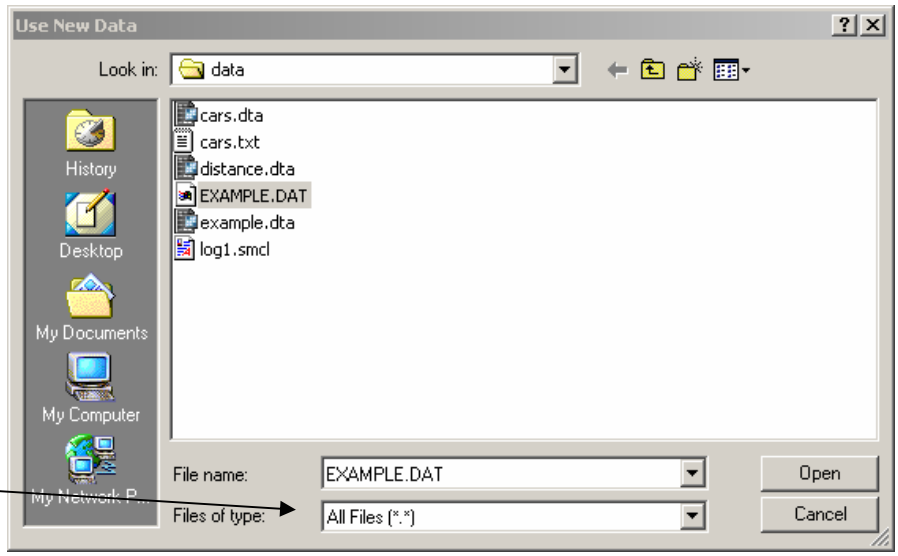
1. Variable names are unique in any one file.
2. Variable names can be up to 32 characters long.
3. Variable names start with a letter or an underscore.
4. Variable names must contain **no** spaces.

Suppose we want to open the file **example.dat**. If you click on

File ➢ Open

The **Open File** dialogue box appears for you to select the ASCII data file you wish to read into STATA.

Raw data is usually in a file with either **.txt** or **.dat** as the suffix. Change the type of file STATA is looking for from **All Files (*.*)**

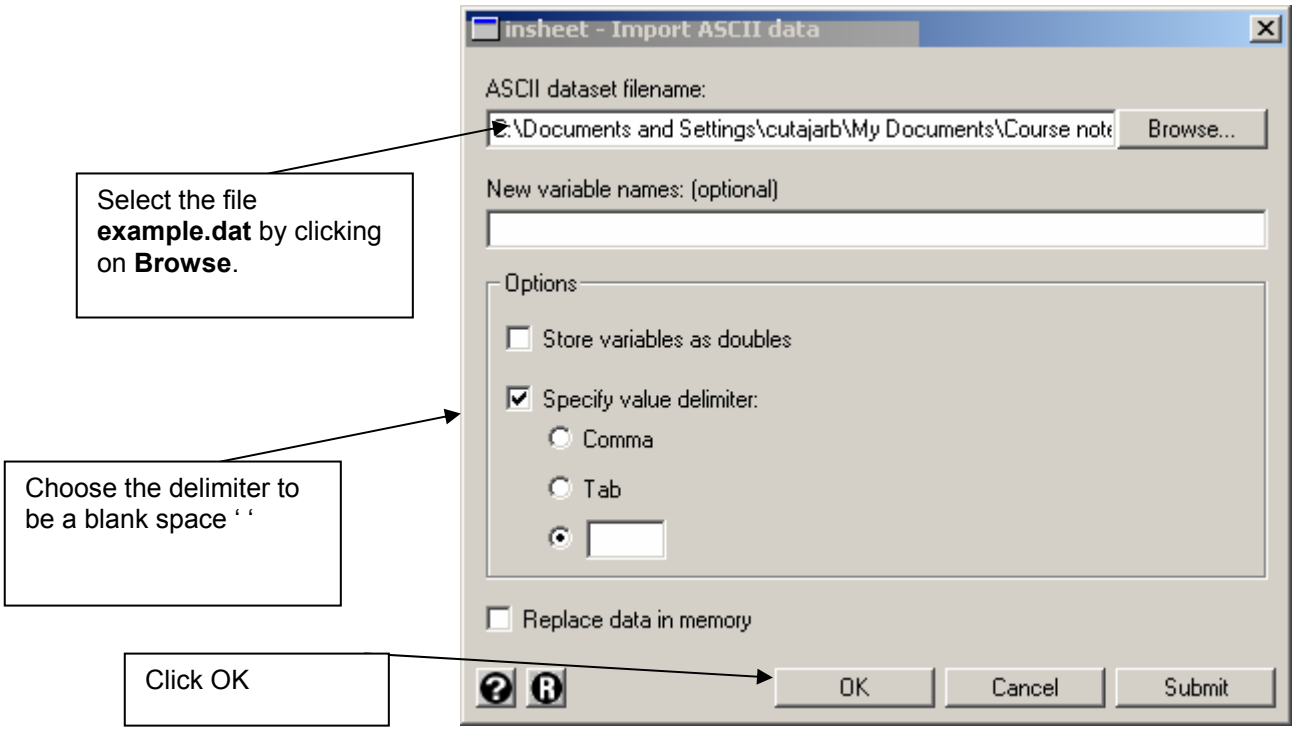



```
. use "C:\Documents and Settings\cutajarb\My Documents\Course notes\Introduction
> STATA\data\EXAMPLE.DAT", clear
file C:\Documents and Settings\cutajarb\My Documents\Course notes\Introduction ST
> ATA\data\EXAMPLE.DAT not Stata format
r(610);
```

This generated an error as the file is not in a format that STATA recognizes.

The correct way of doing it is to click on

File ➢ Import ➢ ASCII data created by a spreadsheet



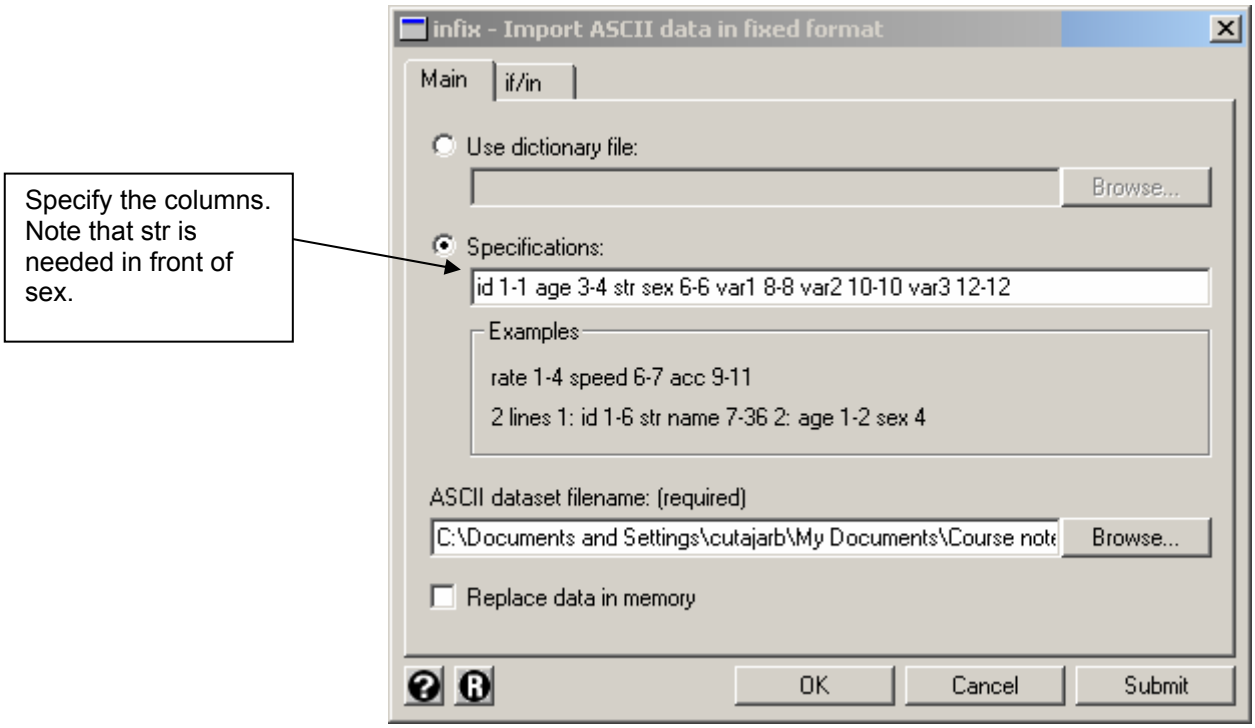

```
. insheet using "C:\Documents and Settings\cutajarb\My Documents\Course notes\Int
> roduction STATA\data\EXAMPLE.DAT", delimiter(" ")
(6 vars, 5 obs)

. l

     +-------------------------------------------+
     | v1   v2   v3   v4   v5   v6               |
     |-------------------------------------------|
  1. |  1   22   M    4    2    1                |
  2. |  2   40   F    2    3    1                |
  3. |  3   27   M    3    3    2                |
  4. |  4   35   M    2    2    4                |
  5. |  5   24   F    1    2    2                |
     +-------------------------------------------+
```

You see that **STATA** made use of the ***insheet*** command. This command is very useful in reading in data from a spreadsheet or database program where the data values are delimited.

Another way of doing this is to tell STATA where the columns for each variable are. By doing this, we can define the variable names while loading the data, rather than afterwards.

Click on

File ➢ Import ➢ ASCII data in fixed format



You can then view the variables loaded in memory by using the ***list*** command or the pull down menu.

```
. infix id 1-1 age 3-4 str sex 6-6 var1 8-8 var2 10-10 var3 12-12 using "C:\Docum
> ents and Settings\cutajarb\My Documents\Course notes\Introduction STATA\data\EX
> AMPLE.DAT"
(5 observations read)

. l

     +-------------------------------------------+
     | id   age   sex   var1   var2   var3       |
     |-------------------------------------------|
  1. |  1    22     M      4      2      1       |
  2. |  2    40     F      2      3      1       |
  3. |  3    27     M      3      3      2       |
  4. |  4    35     M      2      2      4       |
  5. |  5    24     F      1      2      2       |
     +-------------------------------------------+
```

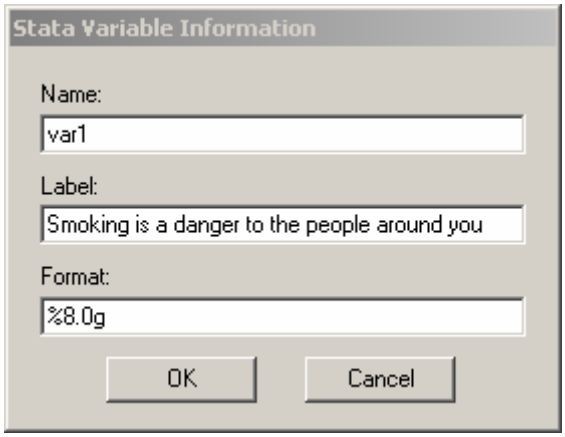
Adding Labels

After **STATA** has read the data for each of the variables into the **Data Editor** Window, labels can be defined to give meaningful descriptions for them. Obviously, a variable named **sex** does not really need a label, but one named **var1** clearly does so that anyone reading the **STATA** output can understand what information is stored as **var1**.

There are two kinds of label that can be applied to each variable, variable labels and value (or character) labels. Variable labels expand on the variable name - they tell you what question was asked, and value labels tell you what the

numerical code given to each response means. So “Sex of respondent” would be the variable label for **sex** and “Male” and “Female” would be the value labels given to the codes M and F respectively.

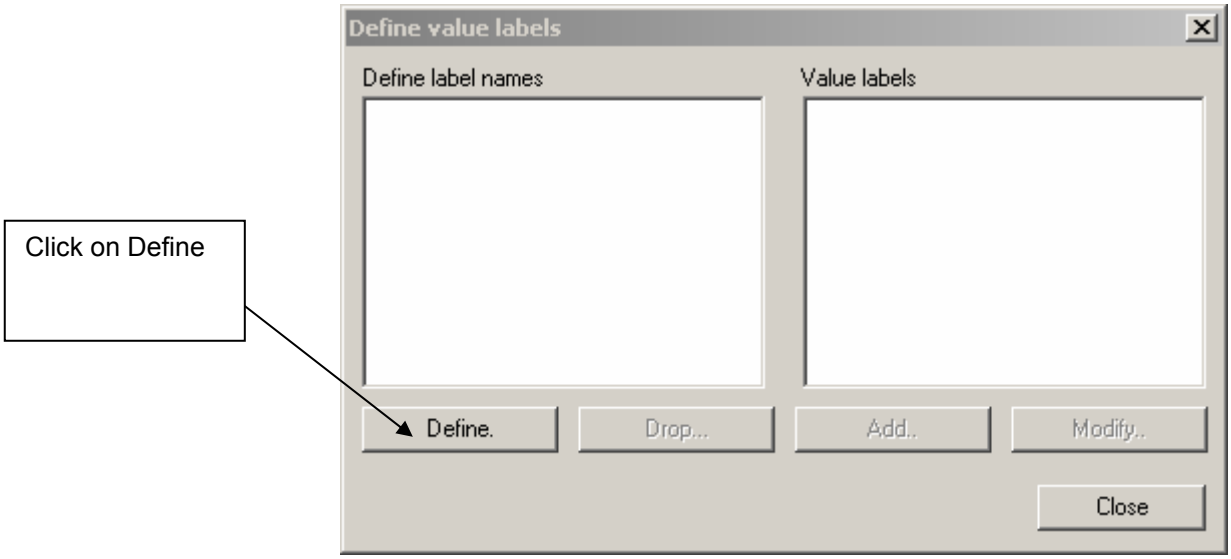
To label the variables, double click on the variable name **var1** in the column heading of the **Data Editor**. A **Variable Label** for **var1** can be typed into the cell in the **Label** column.

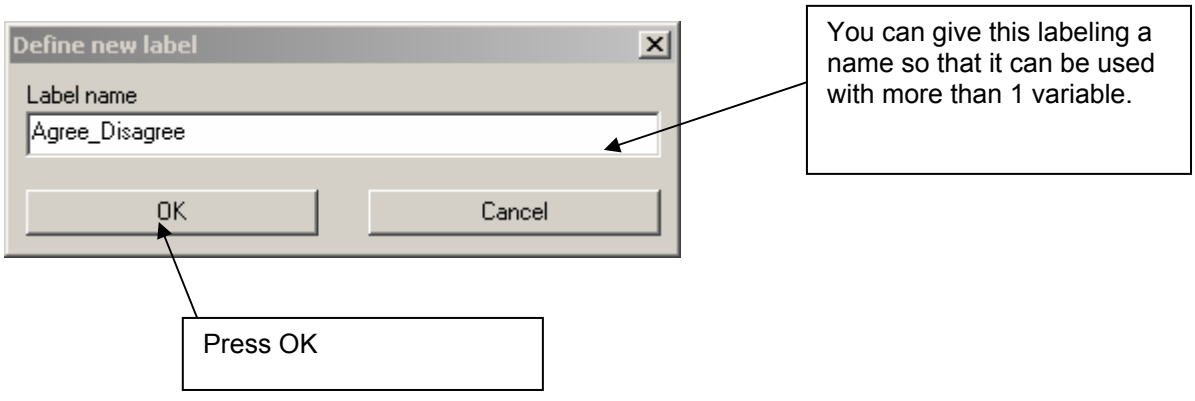


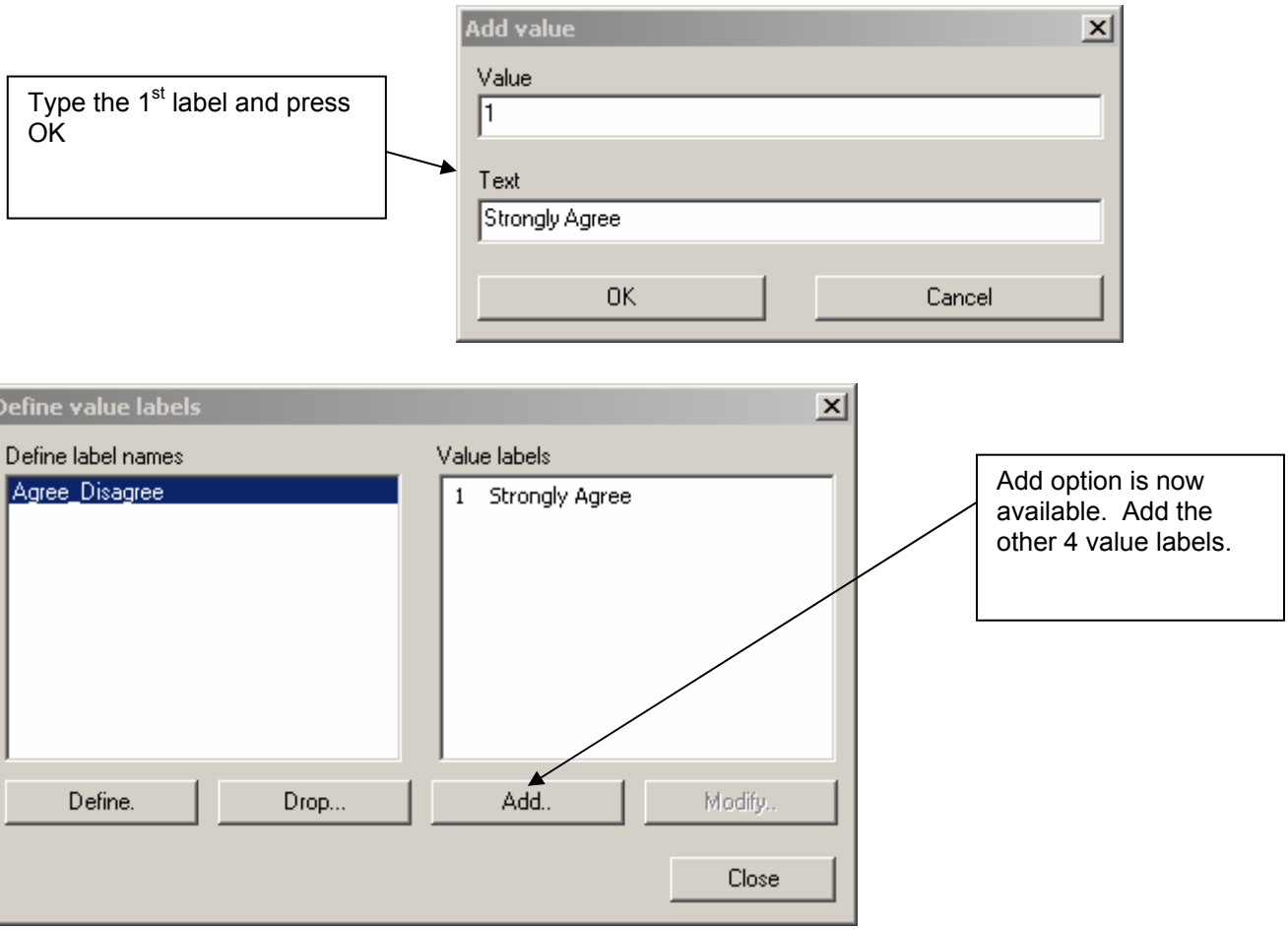
Initially, a variable will have no value labels attached to it. To add **Value Labels**, click

Data ➢ Labels & Notes ➢ Define Value Label

The following window will open

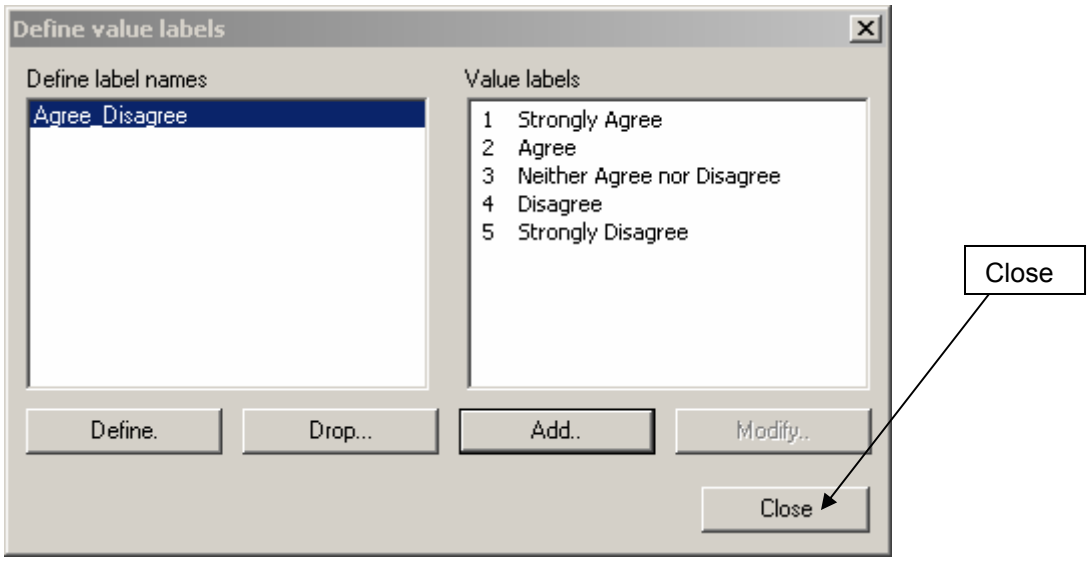






Each value is typed into the **Value** box, followed by its label in the **Value Label** box. Finally, the **Add** button is clicked. Any errors can be corrected by highlighting the label in the **Value Labels** box, and clicking on **Modify**.

When you have finished adding the **Value Labels**, the windows would look like this.

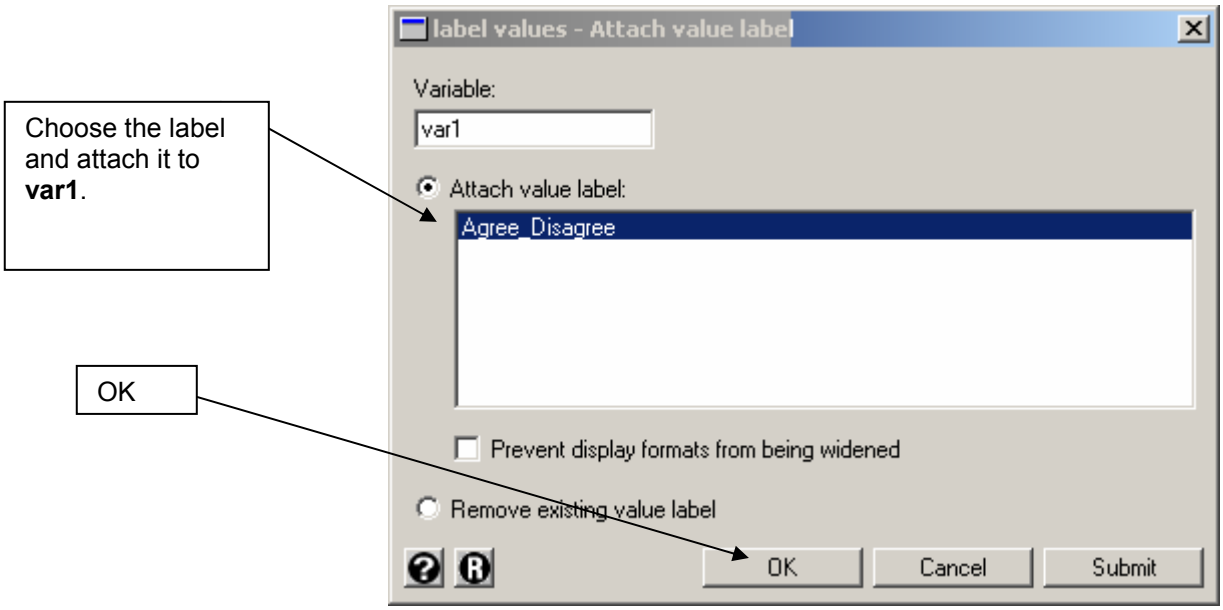


If you know list the data for this variable, you will still obtain the same numbers, rather than the new values.

```
. list var1, separator(5)

     +------+
     | var1 |
     |------|
  1. |    4 |
  2. |    2 |
  3. |    3 |
  4. |    2 |
  5. |    1 |
     +------+
```

This is because we have not attached the labels to the variable. To do so, click on

Data ➢ Labels & Notes ➢ Assign Value Label to Variable



Listing the variable will now output the value labels instead of the corresponding numeric values.

```
. list var1, separator(5)

     +-------------------------+
     |                    var1 |
     |-------------------------|
  1. |                Disagree |
  2. |                   Agree |
  3. | Neither Agree nor Disagree |
  4. |                   Agree |
  5. |           Strongly Agree |
     +-------------------------+
```

Practical Session 1b

Create a log file for this practical session.

In this exercise you will be using **STATA** for Windows to read a raw data file and define **variable labels** and **value labels** for each variable.

You will be using a very small set of data taken form the 1987 Social Attitude Survey. This survey is carried out annually by The Social and Community Planning Research Unit. We have extracted the responses of 25 people to five questions, from the survey. The data from these questions will be put into five **STATA** variables. The following is a 'coding sheet' which shows details about the questions.

	Variable label	**Variable name**	**Columns in data file**	**Value labels**	**Codes**
Q1	Respondent's sex	RSEX	2	Male Female	1 2
Q2	Respondent's age	RAGE	4-5	(Code is age in years) No response	 99
Q3	Which income group would you place yourself?	SRINC	6	High income Middle income Low income No response	1 2 3 9
Q4	How well are you managing on your income?	HINCDIFF	7	Very well Quite well Not very well Not at all well Don't know No response	1 2 3 4 8 9
Q5	Respondent's social class	RRGCLASS	11	Professional Intermediate Skilled Semi-skilled Unskilled Unable to classify Not applicable	1 2 3 4 5 8 0

Using the coding sheet, get **STATA** to read the five variables in the raw data file, '**sample.dat'**.

Add variable labels, value labels and missing values for all the variables. Obtain a frequency distribution for all the variables.

Check that all your variables have been labeled correctly and have missing values by looking through the output. Print the output.

Save your data file to a ***.dta** (or **STATA**) format.

2. Reading more than 1 record of data per case

The small ASCII data set we read into STATA in this session has been rearranged so that the data for each case is over 2 lines. The new ASCII data file is in **D:\Spsswin\Data\Example2.dat**, and is shown below:

Line 1:

1	2	3	4	5	6
id		age			sex
1		2	2		M
2		4	0		F
3		2	7		M
4		3	5		M
5		2	4		F

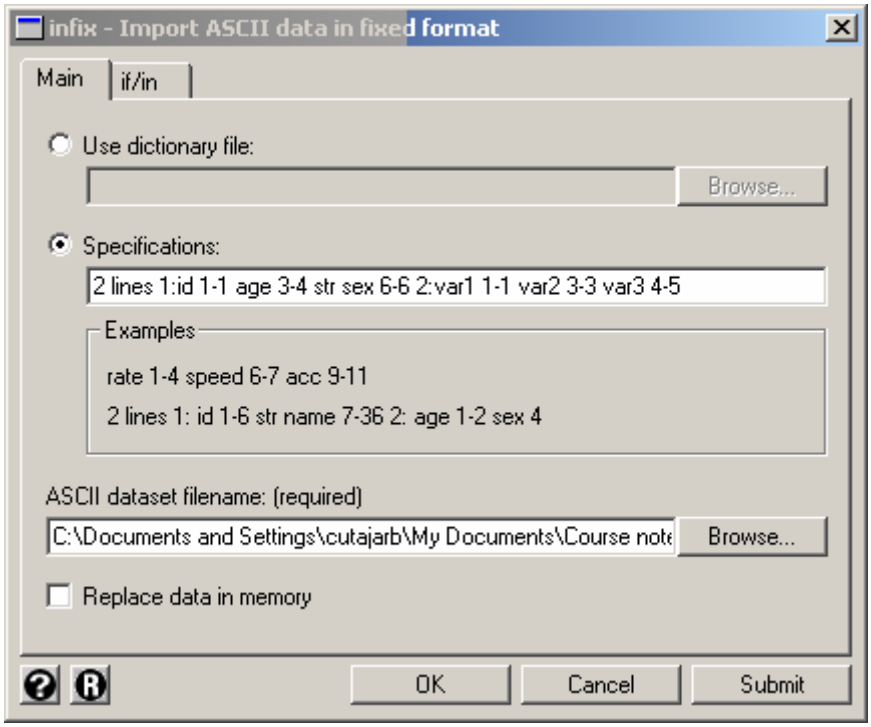
Column numbers
Variable names

Filename:
Example2.dat

Line 2:

1	2	3	4	5
var1		var2	var3	
1		4	2	1
2		2	3	1
3		3	3	2
4		2	2	4
5		1	2	2

Remember that you have to tell STATA which variables are in line 1 and which are in line 2.



Obtain a frequency distribution for the variables, as well as some summary statistics.

Create a new variable, **gender**, which is numeric. Create value labels ('M' and 'F') for this variable.

Obtain a histogram for each variable.