
Stata 10 Tutorial 1

TOPIC: Getting Started with *Stata*

DATA: **auto1.raw** and **auto1.txt** (two *text-format* data files)

TASKS: *Stata 10 Tutorial 1* is intended to introduce (or re-introduce) you to some of the features of your computing environment that you will need to use the *Stata* statistical software program for tutorials and assignments. Topics covered include:

- Accessing and exploring the ECON 351 web site
 - Downloading a text-format data file from the ECON 351 web site
 - Starting, recording and ending a *Stata* session
 - Inputting a text-format data file into *Stata*
 - Displaying and summarizing data in *Stata*
 - Creating a *Stata*-format dataset
 - Labelling variables and *Stata*-format datasets
- The ***Stata* commands** introduced in this tutorial are:

log using	Opens and names a log file to record your <i>Stata</i> session
log close	Closes the log file of your <i>Stata</i> session
cd	Changes the <i>Stata</i> working directory
cf	Compares two datasets
dir	Lists the contents of the current <i>Stata</i> working directory
infile	Loads an unformatted text-format data file into memory
insheet	Loads certain text-format data files into memory
describe	Displays a summary of the current dataset in memory
sort	Orders or re-orders the observations in the current dataset
summarize	Displays descriptive summary statistics for numeric variables
list	Displays values of some or all variables in the current dataset
save	Saves current dataset in memory as a <i>Stata</i> -format dataset
use	Loads a <i>Stata</i> -format data file into memory
label variable	Assigns a label to a variable in a <i>Stata</i> -format dataset
label data	Assigns a label to a <i>Stata</i> -format dataset
exit	Ends a <i>Stata</i> session

NOTE: *Stata* commands are *case sensitive*. All *Stata* commands must be typed in the Command window in **lower case letters**.

HELP: *Stata* has an extensive on-line **Help** facility that provides fairly detailed information (including examples) on all *Stata* commands. Students should become familiar with the *Stata* on-line **Help** system. In the course of doing this tutorial, take the time to browse the **Help** information on some of the above *Stata* commands. To access the on-line **Help** for any *Stata* command:

- choose (click on) **Help** from the *Stata* main menu bar
- click on **Stata Command...** in the **Help** drop down menu
- type the full name of the *Stata* command in the *Stata* command dialog box and click **OK**

Introduction to the ECON 351 Web Site

This first part of the tutorial introduces you to the ECON 351 web site. You will probably need to use this web site for several purposes: to obtain the data files required for tutorials and assignments; to learn when assignments are due and exams are scheduled; to get another copy of the course outline and reading list; and to find out about the *Stata* tutorial schedule.

□ Accessing and Browsing the ECON 351 Web Site

Three widely used web browsers are **Netscape**, **Microsoft Internet Explorer** and **Mozilla Firefox**. **Netscape** is the browser installed on many computers located on the Queen's campus.

- **On the computers in Dunning 350**, **Mozilla Firefox** is the installed web browser. (**Microsoft Internet Explorer** is also installed on the PCs in Dunning 350.)

This section outlines how to use the **Firefox** web browser to locate and browse the ECON 351 web site, specifically the web site for ECON 351 (Fall Term 2008). But the **Netscape** and **Internet Explorer** browsers work similarly.

1. **Start your web browser** by *double-clicking* the **Firefox icon** on the Windows XP desktop.
2. **Proceed to the homepage of the ECON 351 web site.**

How you find your way to the ECON 351 homepage depends on where you start from when beginning a browser session.

- **On the computers in Dunning 350**, the **Firefox** and **Internet Explorer** web browsers begin at the QED (Queen's Economics Department) homepage, the URL for which is **http://www.econ.queensu.ca**.
- The **most direct way to get to the ECON 351 homepage** from wherever your browser starts is to type or copy in your browser's web address window the address (URL) of the ECON 351 web site, which is:

<http://www.econ.queensu.ca/pub/faculty/abbott/econ351/>

Once you have typed or copied the above address into the browser's web address window, press the Enter key on the keypad.

3. Browse the links listed on the homepage of the ECON 351 Web Site.

Now that you are at the ECON 351 homepage, take a few minutes to explore the various pages of the web site. To do this, simply click on the links listed on the ECON 351 homepage.

□ Downloading Data Files from the ECON 351 Web Site

You will frequently need to obtain data files for the tutorials and assignments in ECON 351. These data files can all be downloaded to the computer you are working at from the ECON 351 web site. This section explains how to download a text-format data file using either the **Firefox** or **Internet Explorer** web browser.

What is a **text-format (or ASCII-format) data file**? A text-format file is one that contains no special characters or formatting. It can be viewed and edited with a text editor (such as Windows Notepad or Windows Wordpad) or with any document processor (such as MS-Word or Corel WordPerfect).

Each of the text-format data files **auto1.raw** and **auto1.txt** contains the raw data you will use in all your ECON 351 tutorials. They can be downloaded, or copied, from the ECON 351 web site by proceeding through the following steps.

1. Start your web browser, and **make your way to the ECON 351 homepage**, the address for which is

<http://qed.econ.queensu.ca/pub/faculty/abbott/econ351/>.

2. Under the heading **Information on STATA Tutorials** on the ECON 351 homepage, click on **Download Text-Format Dataset auto1.raw**.

In the **Firefox** web browser, clicking on **Download Text-Format Dataset auto1.raw** displays on the screen the contents of the text-format data file **auto1.raw**. Use the vertical scroll bar to browse the contents of data file **auto1.raw**.

In the **Internet Explorer** web browser, clicking on **Download Text-Format Dataset auto1.raw** opens the **File Download** dialog box. Click the **Open** button to display the file contents on the screen; you can then use the vertical scroll bar to browse the contents of data file **auto1.raw**. Alternatively, click the **Save** button to open the **Save As** dialog box and proceed directly to Step 4 below.

3. To save the text-format data file **auto1.raw** to disk, click on the **File** menu at the top of your browser screen. From the **File** menu select (click on) either the **Save Page As...** or **Save As...** item. The **Save As** dialog box appears.
4. In the **Save As** dialog box, select the drive and/or directory to which you want to save the data file.
 - *To download the data file to a flash memory stick in the E:-drive*, click on the down arrow beside the **Save in:** window; then click on the **E:** icon.
 - *To download the data file to a directory on the hard drive*, for example, **C:\data**, click on the **C:** icon in the **Save in:** window. Next, scroll through the folder list and double-click on the folder to which you want to copy the data file, for example the **data** folder.
5. Click on the **Save** button in the **Save As** dialog box. This copies the data file **auto1.raw** to the directory and drive selected in step 4 above. Close the **Save As** dialog box.
6. Click the **Back** button (←) in the **Firefox** button bar to return to the ECON 351 homepage.
7. Follow essentially the same procedure you used to download the data file **auto1.raw** to download the text-format data file **auto1.txt**, specifically steps 2-6 above.

Before you start your Stata session, make sure that you know where on the hard drive of your PC you have placed the data files **auto1.raw** and **auto1.txt**. Once you have started your *Stata* session, you will want to change the *Stata working directory* to the folder in which you have placed your data files.

What is the default *Stata* working directory?

- **On the computers in Dunning 350**, the default *Stata* working directory is usually **C:\data**.

When working in Dunning 350, you may download your data files to this directory/folder, or if you prefer to some other directory such as **C:\courses**. The important thing is that you keep track of the directory in which you placed the downloaded data files; this is the directory in which you will probably find it most convenient to work during your *Stata* session.

- **On the computers in MC B111**, the default *Stata* working directory is usually **D:\courses**.

Getting Started with Stata

Before beginning this part of the tutorial, make sure that you have downloaded the data files **auto1.raw** and **auto1.txt** and know where on the hard drive of your PC you have placed them.

□ Starting a *Stata* Session

There are two ways to start a *Stata* session.

- If you see a ***Stata 10* icon** on the Windows desktop, simply ***double-click*** it.
- If there is no *Stata 10* icon on the Windows desktop of the computer at which you are working, click on the **Start** button located at the left end of the Windows XP taskbar along the bottom of the desktop window. From the **All Programs** menu, select (click on) the ***Stata 10* icon**.

After you start your *Stata 10* session, the first screen you will see contains four *Stata* windows:

- the ***Stata Command window***, in which you type all *Stata* commands.
- the ***Stata Results window***, which displays the results of each *Stata* command as you enter it.
- the ***Review window***, which displays the past commands you have issued during the current *Stata* session.
- the ***Variables window***, which lists the variables in the currently-loaded data file.

□ Recording a *Stata* Session

Stata is an interactive software program that executes commands as you enter them during a *Stata* session. Being interactive is cool, but it also means that none of your commands or the results they produce will be saved for future reference unless you create a ***log file*** – that is, a file that has the extension ".log". Failure to record your *Stata* session often means wasting a lot of your time. So I strongly recommend that you always record your *Stata* sessions in *Stata* log files.

- **To record your current Stata session**, including all the *Stata* commands you enter and the results (output) produced by these commands, make a text-format log file named **351tutor1.log**. To open (begin) the log file **351tutor1.log**, type in the Command window the following command:

```
log using 351tutor1.log
```

or

```
log using e:351tutor1.log
```

The first command opens a *text-format file* called **351tutor1.log** in the current *Stata* working directory on the hard drive of your PC. The second command opens a text-format file called **351tutor1.log** on a portable electronic storage device in the E:-drive. Once you have opened it, a copy of all the commands you enter during your *Stata* session and of all the results they produce is recorded in that **351tutor1.log** file. A log file with file extension **.log** is a *text-format file*, which means it can be viewed with any text editor such as the MS-DOS Editor or Windows Notepad. You can also view a text-format file using your favorite document processor, such as Microsoft Word or Corel WordPerfect.

Note: It is very important that you include the file extension **.log** in the name of your log file. Otherwise, *Stata* will open a *formatted log file* that has file extension **.smcl**; this is the default file type for *Stata* log files in *Stata 10*. The major limitation of **.smcl** log files is that you must have the *Stata 10* program installed on your computer in order to read **.smcl** log files once you have finished your *Stata* session.

An alternative way to open the log file **351tutor1.log** is to click on the **Log** button in the button bar near the top of the *Stata* window; this opens the **Begin Logging Stata Output** dialog box. In the **Begin Logging Stata Output** dialog box: click on **Save as type:** and select **Log (*.log)**; click on the **File name:** box and type the file name **351tutor1**; and then click on the **Save** button.

- If the log file **351tutor1.log** already exists in the current working directory of your C:-drive or E:-drive, you can overwrite it by simply adding the **replace** option to the **log using** command:

```
log using 351tutor1.log, replace
```

or

```
log using e:351tutor1.log, replace
```

- **To temporarily stop recording your current *Stata* session** while leaving the log file open, type in the Command window:

```
log off
```

- **To resume recording your *Stata* session** to the currently open log file, type in the Command window:

```
log on
```

- **To stop recording your current *Stata* session and close the log file**, type in the Command window:

```
log close
```

- **To add the results for your current *Stata* session to the end of an existing log file**, add the **append** option to the **log using** command. For example, to start writing the results of your *Stata* session to the end of the file **351tutor1.log**, which you have just closed, enter in the Command window one of the following two commands:

```
log using 351tutor1.log, append
```

or

```
log using e:351tutor1.log, append
```

□ Changing the Working Directory – **cd**

Stata expects to find the files it needs (such as *Stata* datasets) and to write the files it creates (such as *Stata* log files) in the current working directory. The name of the current working directory is displayed in the status bar of the *Stata* Command window. You may want to change the *Stata* working directory during a *Stata* session. For instance, if the datasets you wish to use are stored in a directory different from the current working directory, you will want to switch to that directory before loading the datasets into memory. The command you use for this purpose is the **cd** command.

Basic Syntax

- **cd**

Displays the name of the current working directory.

- **cd *drive*:**

Changes the current working directory to a new drive, such as the A:-drive or the C:-drive.

- **cd *directory***

Changes the current working directory to a new directory in the same drive as the original directory.

- **cd *drive:directory***

Changes the current working directory to the new drive and directory specified.

- **cd "*drive: directory name*"**

Changes the current working directory to the new drive and directory specified, where the new directory name contains embedded spaces.

Usage Note: Windows XP allows directory names with embedded spaces. To change the working directory to a directory with spaces in its name, enclose the entire drive and directory specification in double quotation marks.

- **cd ..**

Moves the current working directory one level up the directory tree from the original directory.

Simple Examples

- `cd` Displays current working directory
- `cd e:` Changes working directory to the flash memory stick in the E:-drive
- `cd c:\courses` Changes working directory to *C:\courses*
- `cd c:\data` Changes working directory to *C:\data*

An Extended Example

Enter the following sequence of **cd** commands and take the time to observe the current working directory that each produces. Notice that in some cases there is more than one way to enter the **cd** command.

- Enter the command: `cd`
C:\data
- Enter the command: `cd \` *or* `cd ..`
C:
- Enter the command: `cd courses` *or* `cd c:\courses`
C:\courses
- Enter the command: `cd e:`
E:
- Enter the command: `cd c:`
C:
- Enter the command: `cd data`
C:\data

□ **Displaying the Contents of Directories – `dir`**

From time to time you will probably want to see what files are contained in the current directory (or folder). The **`dir`** command is used for this purpose; it displays a list of the names of some or all of the files in the current *Stata* working directory.

Simple Examples

- To illustrate the use of the **`dir`** command, change the current working directory to **`C:\courses`** if this is the directory in which you placed your data files. Otherwise, change to the directory where you placed your data files. Type in the Command window:

```
cd c:\courses      or      cd \ courses
```

- To display in the Results window the names of all the files in the current working directory, type in the Command window:

```
dir
```

- To display in the Results window the names of all the files in the current working directory that have the extension **`.dta`**, type in the Command window:

```
dir *.dta
```

Note that the asterisk "*" can be used to indicate any string of characters.

- To display in the Results window the names of all the files in the current working directory that have the extension **`.log`**, type in the Command window:

```
dir *.log
```

- To display in the Results window the names of all the files in the current working directory that begin with the character string 'auto', type in the Command window:

```
dir auto*.*
```

□ Loading a Text-Format Data File into Stata: Method 1 – *infile*

Before starting your *Stata* session, you downloaded the text-format (or ASCII-format) data file **auto1.raw** from the ECON 351 web site and placed it in a directory or folder on the C:-drive of your computer. This section demonstrates how to input that data file into *Stata* using the **infile** command.

When can the *infile* command be used?

A text-format (or ASCII-format) data file must exhibit certain properties if it is to be correctly read by an **infile** command.

1. The data can be **space-separated**, **tab-separated**, or **comma-separated**. Another term for **space-separated** is **space-delimited**; another term for **tab-separated** is **tab-delimited**; and another term for **comma-separated** is **comma-delimited**.
2. Strings with embedded spaces or commas *must be enclosed in single or double quotes*.
3. The text-format data file *must contain only variable values*. This means that the first line of the text-format data file *cannot contain variable names*.
4. A single observation can be on more than one line, or there can be more than one observation on each line.

To use the **infile** command correctly, you need to know four things about the text-format data file you are inputting: (1) the *number of variables* in the data file; (2) the *order* in which the values of the variables are written in the data file for each observation or record; (3) which of the variables are *numeric variables* and which are *string variables*; and (4) for each string variable, the *maximum length* of the values it takes, i.e., the maximum number of characters (including embedded spaces) its values contain.

Basic Syntax

infile varlist using filename

Loads the data in the text file *filename* on the variables named in *varlist*, where the data values for the variables are separated by one or more spaces, by tabs, or by commas. Such a file is called an unformatted, or free format, text file.

1. Note that the *varlist* is **required**, not optional. This means that you must know both the number and order of the variables for each observation.
2. The text file *filename* **must not contain variable names in the first line**.

What does the text-format data file auto1.raw look like?

- To see what the data file **auto1.raw** looks like, enter in the Command window:

```
type auto1.raw, showtabs
```

Here is the listing of the first 10 lines and last 3 lines of data file **auto1.raw** produced by the above **type** command:

"AMC Concord"	4099	22	2930	0
"AMC Pacer"	4749	17	3350	0
"AMC Spirit"	3799	22	2640	0
"Buick Century"	4816	20	3250	0
"Buick Electra"	7827	15	4080	0
"Buick LeSabre"	5788	18	3670	0
"Buick Opel"	4453	26	2230	0
"Buick Regal"	5189	20	3280	0
"Buick Riviera"	10372	16	3880	0
"Buick Skylark"	4082	19	3400	0
<i>(output omitted)</i>				
"VW Rabbit"	4697	25	1930	1
"VW Scirocco"	6850	25	1990	1
"Volvo 260"	11995	17	3170	1

Note the following properties of the data file **auto1.raw**:

- (1) The number of variables in the data file is 5.
- (2) The order of the variables by name is: **make**, **price**, **mpg**, **weight**, **foreign**.
- (3) The first variable in each record, **make**, is a *string variable*; its values are contained in double quotation marks.
- (4) The other four variables are *numeric variables*.
- (5) The maximum length of the string variable **make** is 18 characters (though this is not apparent from the partial listing of the data file given above).

- Suppose you placed the text-format data file **auto1.raw** in the directory **C:\data**. To load the text-format data file **auto1.raw** into memory, type in the Command window one of the following two commands:

```
infile str18 make price mpg weight foreign using auto1.raw
```

or

```
infile str18 make price mpg weight foreign using auto1
```

Note two things about these **infile** commands:

- ◆ The filename can be given as either **auto1.raw** or **auto1**. If the filename has the extension **.raw**, *Stata* knows that the data file is an unformatted text file and automatically assumes that filename **auto1** actually refers to the file **auto1.raw**. If the text-format data file *does not* have the extension **.raw**, then the full filename must be given after the keyword **using** in the **infile** command.
- ◆ The keyword **str18** appears in the *varlist* immediately before the variable name **make**. This keyword must appear in the *varlist* **immediately in front of each string variable name**; the number **18** specifies the maximum length taken by the values of the string variable **make**.

□ Summarizing the contents of the current dataset – describe

The **describe** command displays a summary of the contents of the current dataset in memory. In the present case, the current dataset in memory is the text-format data file **auto1.raw**. To summarize the contents of the current dataset, type in the Command window:

```
describe
```

□ Displaying the values of variables -- list

The **list** command displays the values of some or all of the variables in the current dataset. There are several available forms for the **list** command.

-
- To display in the Results window the values of all variables for all observations in dataset **auto1.raw**, type in the Command window:

```
list
```

- To display the values of all variables only for the first 20 observations in **auto1.raw**, type in the Command window:

```
list in 1/20
```

- To display the values of all variables for observations 50 to last, type in the Command window:

```
list in 50/1
```

- To display the values of the variables **make**, **price**, **mpg**, and **weight** for all observations in dataset **auto1.raw**, type in the Command window:

```
list make price mpg weight
```

- To temporarily stop recording your *Stata* session to the currently open log file **351tutor1.log**, enter in the Command window:

```
log off
```

- To display the values of the variables **mpg**, **weight**, and **foreign** only for those observations for which the value of **mpg** is greater than 20, type in the Command window:

```
list mpg weight foreign if mpg > 20
```

- To resume recording your *Stata* session to the log file **351tutor1.log**, enter in the Command window:

```
log on
```

- To display the values of the variables **mpg**, **weight** and **foreign** only for those observations for which the value of **mpg** is greater than 20 and **foreign** equals 1, type in the Command window:


```
list mpg weight foreign if mpg > 20 & foreign==1
```

Describe the results of this command; that is, explain what it does. Note the use of the double equality "==" in the **if** statement; *Stata* uses "==" for the equal sign in all *logical* statements, but not in any other kind of statements.

□ Calculating descriptive summary statistics – summarize

The **summarize** command calculates and displays descriptive summary statistics for some or all of the *numeric variables* in the current dataset. The basic descriptive summary statistics produced for each numeric variable by the **summarize** command are: the number of observations (Obs); the mean (Mean); the standard deviation (Std. Dev.); and the range, specifically the minimum value (Min) and the maximum value (Max). These summary statistics obviously cannot be computed for *string variables*, since string variables do not take numerical values.

- To calculate and display basic descriptive summary statistics for all variables and all observations in the **auto1.raw** dataset, type in the Command window:

```
summarize
```

Note that this command does not calculate summary statistics for the variable **make**. The reason is that **make** is a *string variable*; the summarize command can only compute descriptive statistics for *numeric variables*.

- To calculate and display basic descriptive summary statistics for the variables **price**, **mpg** and **weight** for those observations in the current dataset **auto1.raw** for which the value of the variable **mpg** is greater than 20, enter the command:

```
summarize price mpg weight if mpg > 20
```

- To calculate and display a table of basic descriptive summary statistics for the variables **price**, **mpg** and **weight** for as many subsets of observations as there are distinct values of the categorical variable **foreign**, enter the commands:

```
sort foreign
by foreign: summarize price mpg weight
```

The **sort** command sorts the observations in the current dataset in ascending order according to the values of the variable **foreign**. Since the variable **foreign** takes only two distinct values (i.e., 1 for “Foreign” cars and 0 for “Domestic” cars), the second command produces two tables of summary statistics for the variables **price**, **mpg** and **weight**.

- An alternative way to calculate and display separate tables for domestic and foreign cars of basic descriptive summary statistics for the variables **price**, **mpg** and **weight** is to use the **if option** on the summarize command. Enter the commands:

```
summarize price mpg weight if foreign==0
summarize price mpg weight if foreign==1
```

- Enter the following **summarize** command, and interpret the results it produces:

```
by foreign: summarize price mpg weight if mpg>20
```

- The **summarize** command has a **detail** option. To see how it works, enter the following command:

```
summarize price mpg weight, detail
```

□ Creating or Saving a Stata-format Dataset – save

Stata-format Data Files – .dta files

Stata-format data files are *binary format* files, which means that you cannot read them with a text editor or word processor. Only the *Stata* program can read a *Stata*-format data file.

All *Stata*-format data files have the filename extension **.dta**. Thus, a generic *Stata*-format data file has the name **filename.dta**, where **filename** is a user-supplied name of 32 characters or less.

Creating or saving a Stata-format data file

1. To save a *new dataset* to disk (or to save an old dataset under a new name):

- either type in the Command window: **save *filename***
- *or* pull down the **File** menu and choose **Save As**

This command saves to the current working directory on disk the *Stata*-format data file ***filename.dta***, where *filename* is a user-supplied file name.

2. To resave a dataset that has been changed (and overwrite the original **.dta** file):

- type in the Command window: **save *filename*, replace**
- *or* simply type in the Command window: **save, replace**
- *or* pull down the **File** menu and choose **Save**

Note that this command overwrites the original *Stata*-format data file of the same name. If you do not wish to lose the original **.dta** file, give the changed dataset a new name using the save command given in **1.** above.

- To save the data file you have been working on as a *Stata*-format dataset, type in the Command window:

```
save auto1
```

This command saves on disk in the *Stata* working directory the *Stata*-format dataset **auto1.dta**. Note that the *Stata* **save** command automatically attaches the extension **.dta** to the *filename* you specify. Note too that the text-format data file **auto1.raw** you originally read into *Stata* is left unchanged on disk by the *Stata* **save** command.

- The current dataset in memory is now the *Stata*-format dataset **auto1.dta**. To summarize the contents of this current dataset, type in the Command window:

```
describe
```

Inspect the results of this command.

- To confirm that the *Stata*-format dataset **auto1.dta** has been saved to the current *Stata* working directory, type in the Command window:

```
dir *.dta
```

Among the listed filenames you should see **auto1.dta**.

□ Removing a Dataset From *Stata* Memory – clear

Stata can have only one dataset in memory at a time. To clear the current dataset **auto1.dta** from memory, type in the Command window:

```
clear
```

Once *Stata* executes this **clear** command, there is no current dataset in memory. But of course the *Stata*-format dataset **auto1.dta** is still sitting on your computer's C:-drive since you previously put it there with the **save** command.

Stata now has no dataset in memory on which to work. In the following section, you will learn how to use the **insheet** command to read certain text-format datasets into memory.

□ Loading a Text-Format Data File into *Stata*: Method 2 – insheet

Before starting your *Stata* session, you should have downloaded the text-format (or ASCII-format) data file **auto1.txt** from the ECON 351 web site and placed it in a directory or folder on the C:-drive of your computer. This section demonstrates how to input that data file into *Stata* using the **insheet** command.

When can the *insheet* command be used?

The **insheet** command is intended to read text files created by a spreadsheet or database program. A text-format (or ASCII-format) data file must exhibit certain properties if it is to be correctly read by an **insheet** command.

1. The data must be **tab-separated** or **comma-separated**; the **insheet** command cannot load data files that are space-separated.

2. A single observation must be on only one line. In other words, **there must be one, and only one, observation on each line** of the data file.
3. The **first line** of the text-format data file **can optionally contain variable names**.

To use the **insheet** command correctly, you need to know relatively little about the text-format data file; *Stata* will figure out many of the file's characteristics and correctly read the data into memory. However, if the first line of the data file does not contain variable names, you should know at least the order of the variables in each observation. You should also know that the data file is **tab-delimited** or **comma-delimited**, and that each line of the file contains one, and only one, observation.

Basic Syntax

insheet [*varlist*] using *filename* [, names tab comma clear]

Loads the data in the text file *filename* on the variables named in *varlist*, where the data values for the variables are separated (or delimited) by tabs or by commas. Such a file is called an unformatted, or free format, text file.

1. Note that the *varlist* is **optional**. If the first line of the text file *filename* contains variable names, the *varlist* can be omitted and the file will still be correctly read by the **insheet** command. In other words, when the first line of the text file *filename* contains variable names, you need not know either the number or order of the variables for each observation.
2. The text file *filename* **may contain variable names in the first line**. If it does not contain variable names in the first line, the variable names can optionally be given in a *varlist*.
3. The **names option** tells *Stata* that the first line of the data file contains **variable names**. If the data file does in fact include variable names in the first line, the **names** option simply reads the file more quickly.
4. The **tabs option** tells *Stata* that the data values are **tab-separated**, or **tab-delimited**. If the data file is in fact tab-separated, the **tabs** option simply reads the file more quickly.

5. The **comma option** tells *Stata* that the data values are *comma-separated*, or *comma-delimited*. If the data file is in fact comma-separated, the **comma** option simply enables **insheet** to read the file more quickly.
6. The **clear option** tells *Stata* to remove the current dataset from memory before attempting to load the new dataset *filename*. Remember that *Stata* can have only one dataset in memory at a time.

Usage: To read many data files, you need only enter the following simple form of the **insheet** command:

insheet using filename

What does the text-format data file auto1.txt look like?

- Assuming the text-format data file **auto1.txt** is in the current working directory, you can view its contents by entering the following **type** command:

```
type auto1.txt, showtabs
```

The lines returned by the above **type** command are:

```
make,price,mpg,weight,foreign
AMC Concord,4099,22,2930,0
AMC Pacer,4749,17,3350,0
AMC Spirit,3799,22,2640,0
Buick Century,4816,20,3250,0
Buick Electra,7827,15,4080,0
Buick LeSabre,5788,18,3670,0
Buick Opel,4453,26,2230,0
Buick Regal,5189,20,3280,0
Buick Riviera,10372,16,3880,0
Buick Skylark,4082,19,3400,0
```

(output omitted)

```
VW Rabbit,4697,25,1930,1
VW Scirocco,6850,25,1990,1
Volvo 260,11995,17,3170,1
```

Note the following properties of the text-format data file **auto1.txt**:

- (1) The *first line* of the data file **auto1.txt** contains the variable names, which are separated by commas.
- (2) The variable values are *comma-separated*, or *comma-delimited*.
- (3) Each observation is entirely contained in one line; there is *only one observation per line*.
- (4) The number of variables in the data file is 5.
- (5) The order of the variables by name is: **make, price, mpg, weight, foreign**.
- (6) The values of the *string variable* **make** are not enclosed in single or double quotation marks.
- (7) The other four variables are *numeric variables*.

This is exactly the sort of data file that the **insheet** command can read very easily.

- To load the text-format data file **auto1.txt** into memory, type in the Command window the following command:

```
insheet using auto1.txt
```

- To display a summary of the contents of the dataset you have just loaded into memory, enter in the Command window the **describe** command:

```
describe
```

- To display in the Results window the values of all variables for all observations in the dataset **auto1.txt**, enter in the Command window:

```
list
```

Here is part of the display returned by the above **list** command:

	make	price	mpg	weight	foreign
1.	AMC Concord	4099	22	2930	0
2.	AMC Pacer	4749	17	3350	0
3.	AMC Spirit	3799	22	2640	0
4.	Buick Century	4816	20	3250	0
5.	Buick Electra	7827	15	4080	0
6.	Buick LeSabre	5788	18	3670	0
7.	Buick Opel	4453	26	2230	0
8.	Buick Regal	5189	20	3280	0
9.	Buick Riviera	10372	16	3880	0
10.	Buick Skylark	4082	19	3400	0

(output omitted)

72.	VW Rabbit	4697	25	1930	1
73.	VW Scirocco	6850	25	1990	1
74.	Volvo 260	11995	17	3170	1

- To compute descriptive summary statistics for the four *numeric variables* in the dataset **auto1.txt**, enter in the Command window:

```
summarize price mpg weight foreign
```

□ Comparing Two Datasets – cf

You currently have in memory the dataset **auto1.txt**. You previously saved the dataset **auto1.raw** in the *Stata*-format dataset **auto1.dta**. You can compare the contents of **auto1.txt** and **auto1.dta** using the *Stata* **cf** command.

Basic Syntax

cf *varlist* using *filename* [, **verbose**]

Compares the values of the variables in *varlist* in the dataset in memory with the values of the corresponding variables in *filename*.

The **cf** command returns nothing if the specified variables are identical in the two datasets, and a return code of 9 if there are differences in the variables.

The **verbose** option provides a variable-by-variable comparison of the two datasets.

Simple Examples

- To compare all the variables in the datasets **auto1.txt** and **auto1.dta**, enter in the Command window:

```
cf _all using auto1
```

or

```
cf _all using auto1.dta
```


- To do a more detailed comparison of all the variables in the datasets **auto1.txt** and **auto1.dta**, add the **verbose option** to the above **cf** command. Enter in the Command window:

```
cf _all using auto1, verbose
```

or

```
cf _all using auto1.dta, verbose
```

- To compare only the variables **price** and **weight** in the datasets **auto1.txt** and **auto1.dta**, enter in the Command window:

```
cf price weight using auto1
```

or

```
cf price weight using auto1.dta, verbose
```

- The results of the previous **cf** commands should be sufficient to persuade you that all the variables in the datasets **auto1.txt** and **auto1.dta** are identical. You can therefore discard the current dataset in memory, which is **auto1.txt**, by entering the following clear command in the Command window:

```
clear
```

There is now no dataset in memory.

❑ Loading a Stata-format Dataset – use

The **use** command is *Stata's* way of loading into memory a previously-created *Stata-format* dataset.

To load a Stata-format data file:

1. type in the Command window: **use filename**

or

2. pull down the **File** menu and choose **Open**

This command loads into memory the *Stata-format* dataset **filename.dta**, where **filename** is a user-supplied filename of 32 characters or less.

Simple Examples

- `use auto1` *or* `use auto1.dta`
- `use C:\courses\auto1`
- To load or read into memory the *Stata*-format dataset **auto1.dta**, do either of the following:
 1. type in the Command window: `use auto1`
or
 2. pull down the **File** menu, choose **Open**, then choose **auto1.dta**.

This command loads into memory the *Stata*-format dataset **auto1.dta**.

- To see that the *Stata*-format dataset **auto1.dta** is now the current dataset in memory, type in the Command window the following two commands:

```
describe
summarize
```

□ Labeling Variables in a *Stata*-format Dataset – label variable

It is often helpful to give the variables in a *Stata*-format dataset variable labels that provide brief definitions of the variables. The command you use for this purpose is the **label variable** command.

Basic Syntax

label variable *varname* "*user-supplied variable label*"

where *varname* is the name of the variable you want to label and *user-supplied variable label* is the label you wish to give the variable *varname*. Note that *user-supplied variable label* must be enclosed in double quotation marks.

-
- To assign labels to the five variables in the *Stata*-format dataset **auto1.dta**, type in the Command window the following commands:

```
label variable make "Make of car"  
label variable price "Price (US dollars)"  
label variable mpg "Miles per gallon"  
label variable weight "Weight (pounds)"  
label variable foreign "Foreign car indicator"
```

- You can also give your dataset a short descriptive label using the **label data** command. The **label data** command works just like the **label variable** command. Type in the Command window:

```
label data "North American Car Prices, 1978"
```

This command assigns the label **North American Car Prices, 1978** to the *Stata*-format dataset **auto1.dta**.

- You have to this point made some changes to the original dataset **auto1.dta**. In particular, you have assigned labels to the dataset and its variables. To save these changes, enter the following **save** command with the **replace** option:

```
save, replace
```

This command saves on disk the revised dataset in the *Stata*-format data file **auto1.dta**. It actually overwrites the original disk copy of the dataset **auto1.dta**, and replaces it with the revised dataset that currently resides in memory.

- Finally, you can summarize the contents of the revised dataset **auto1.dta** by entering in the Command window:

```
describe
```

Inspect the results of this command to see how the new labels you have created are displayed.

□ Ending a Stata Session

- **Before you end your Stata session**, you should always close the log file you have been recording. Enter the command:

```
log close
```

An alternative way to close a log file is to click on the **Log** button in the button bar near the top of the *Stata* window. Then in the **Stata Log Options** dialog box select **close log file** and click on the **OK** button.

- Suppose you now remember one more thing you would like to do before ending your *Stata* session. To add the additional results to the log file you have just closed, enter in the Command window the following command:

```
log using 351tutor1.log, append
```

- You want to display the values of the variables **make**, **price**, **weight** and **foreign** separately for foreign cars (for which `foreign = 1`) and domestic cars (for which `foreign = 0`). Enter in the Command window the following two commands:

```
list make price weight foreign if foreign==1  
list make price weight foreign if foreign==0
```

You are now ready to end your *Stata* session.

- **To end your Stata session**, use the **exit** command. Enter the command:

```
exit    or    exit, clear
```

Note that the log file **351tutor1.log** is closed automatically when you exit, or end, your *Stata* session.

□ Cleaning Up and Clearing Out

After returning to Windows, you should copy all the files you have used and created during your *Stata* session to your own electronic storage device. These files will be found in the *Stata working directory*, which is usually **C:\data** on the computers in Dunning 350. There are four files you will want to be sure you have: the text-format data files **auto1.raw** and **auto1.txt**, the *Stata*-format dataset **auto1.dta**, and the *Stata* log file **351tutor1.log**. Use the Windows **copy** command to copy any files you want to keep to your own portable electronic storage device (e.g., a flash memory stick) in the E:-drive (or a diskette in the A:-drive).

Finally, **as a courtesy to other users of the computing classroom**, please delete all the files you have used or created from the *Stata* working directory.