This document provides an overview of the statistical analysis package Stata (version 11). It will cover some of the fundamentals of data management, including opening, saving and cleaning data sets. The auto.dta dataset, which comes pre-installed with Stata 11, will be used throughout this guide.

**Command Line and Graphical User Interfaces**

Stata offers both a graphical user interface (GUI) as well as a command line interface (CLI). This guide will focus on entering commands directly into the CLI prompt, however users are welcome to use the GUI if they prefer to do so.

**Setting up the working directory**

A working directory is the reference location to all datasets and output during a Stata session. The following commands are useful in dealing with directories:

1. `pwd` – prints the current working directory
2. `dir` – prints the kind of files that are in the current directory
3. `cd` – changes directories (note: the directory to change to needs to be specified and for longer directories it may be easier to use the GUI by selecting the “Change Working Directory” option under the “File” menu)

Examples of changing directories can be found below:

```plaintext
in Windows : cd “C:uers”
in Mac : cd “/Users”
```

**Opening the data**

It is important to know the format of a dataset before attempting to open it. Different formats of data require different commands. The two most commonly used commands for opening data are “use” and “insheet”. “use” opens Stata formatted datasets (.dta files) and “insheet” opens comma separated (.csv files) files. Data files formatted as excel (.xls or .xlsx) can be transformed into comma separated (.csv) files by using the “save as” option within excel and Stata offers additional functions, such as “infix” and “infile” for reading data. If you ever have a question about how to accurately use a Stata command then use the “help” function, for example “help infile”.

Examples of using the “use” and “insheet” commands can be found below:

```plaintext
1. auto.dta
   use auto
```
2. auto.csv

insheet using “auto.csv”

The exercise below opens the auto.dta file stored within Stata. For this type of the data, you can use “sysuse” command.

**sysuse auto**

Immediately after the command is executed, we can notice the variables in the file displayed in the lower left panel of the main Stata window.

<table>
<thead>
<tr>
<th>Name</th>
<th>Label</th>
<th>Type</th>
<th>Format</th>
</tr>
</thead>
<tbody>
<tr>
<td>make</td>
<td>Make and Model</td>
<td>str8</td>
<td>%18s</td>
</tr>
<tr>
<td>price</td>
<td>Price</td>
<td>int</td>
<td>%8.0gc</td>
</tr>
<tr>
<td>mpg</td>
<td>Mileage (mpg)</td>
<td>int</td>
<td>%8.0g</td>
</tr>
<tr>
<td>rep78</td>
<td>Repair Record 1978</td>
<td>int</td>
<td>%8.0g</td>
</tr>
<tr>
<td>headroom</td>
<td>Headroom (in.)</td>
<td>float</td>
<td>%6.1f</td>
</tr>
<tr>
<td>trunk</td>
<td>Trunk space (cu. ft.)</td>
<td>int</td>
<td>%8.0g</td>
</tr>
<tr>
<td>weight</td>
<td>Weight (lbs.)</td>
<td>int</td>
<td>%8.0gc</td>
</tr>
<tr>
<td>length</td>
<td>Length (in.)</td>
<td>int</td>
<td>%8.0g</td>
</tr>
<tr>
<td>turn</td>
<td>Turn Circle (ft.)</td>
<td>int</td>
<td>%8.0g</td>
</tr>
<tr>
<td>displacement</td>
<td>Displacement (cu. in.)</td>
<td>int</td>
<td>%8.0g</td>
</tr>
<tr>
<td>gear_ratio</td>
<td>Gear Ratio</td>
<td>float</td>
<td>%6.2f</td>
</tr>
<tr>
<td>foreign</td>
<td>Car type</td>
<td>byte</td>
<td>%8.0g</td>
</tr>
</tbody>
</table>

**Saving the data**

To save the data file, you can use the “save” command with the name of a new dataset.

**save autonew**

It is important to note that the procedure just described saves only the dataset, not the results. To save any results we must use a log-file, described in the next section.

**Saving the results**

To begin saving the contents of the Results window we have to start a log-file using the following command.

**log using autolog**

Once we do this, all the output that is displayed in the Results window will be recorded in the newly created log-file. It is important to start logging the results early in the Stata session to document a session’s results and data manipulations. When you complete your work, you can close the log with **log close**
Previously saved log-files can be viewed by double clicking on them from Windows Explorer. Unless you set up a different directory for a log file, it will be saved in the working directory.

**Viewing data**

To display a spreadsheet of the variables and observations we can type “browse” in the command window. In this mode, you cannot make any changes to the dataset.

```
browse
```

**Editing data**

To edit data, Stata uses the same spreadsheet used to view the data with difference that the cells are editable. To bring the Data Editor up we can use the command `edit`.

```
edit
```

To modify a particular entry of data, simply select the cell you would like to change and type in the desired value. Notice that every action that we do, using the command interface or browsing our way through the graphical interface, is documented in the Results window.

**Exploratory data analysis**

Exploratory analysis gives us a brief glimpse at the basic statistics of the dataset in use. First we can look at a description of the dataset and its variables by using the command `describe`. This will display a table with basic information about the variables in the dataset.
We can get a short summary of descriptive statistics by using the command `summarize`. This last command will display a table with the nonmissing numeric observations, mean, standard deviation, minimum and maximum for all the variables.

```
. summarize

Variable | Obs  | Mean  | Std. Dev. | Min  | Max
---------|------|-------|-----------|------|-----
      make | 74   | 13.75676 | 4.277404  | 5    | 23  
     price | 74   | 6165.257 | 2949.496  | 3291 | 15906
     mpg   | 74   | 3019.459 | 777.1936  | 1760 | 4840
    rep78 | 69   | 3.405797 | 0.989323  | 1    | 5   
  headroom | 74   | 2.993243 | 0.8459948 | 1.5  | 5   
     trunk | 74   | 13.75676 | 4.277404  | 5    | 23  
    weight| 74   | 3019.459 | 777.1936  | 1760 | 4840
     length| 74   | 187.9324 | 22.26634  | 142  | 233 
     turn  | 74   | 39.64865 | 4.399354  | 31   | 51  
displacement | 74 | 197.2973 | 91.83722  | 79   | 425 
  gear_ratio | 74   | 3.014865 | 0.4562871 | 2.19 | 3.89
     foreign| 74   | .2972973 | .4601885  | 0    | 1   
```

We can get some more statistics by using the same command with the `detail` option, that is:
You can specify the variable you want to summarize with adding a variable after “summarize”. At this point it is useful to note that most Stata commands have an abbreviation, that is, we can type on a fraction of the characters of a given command and Stata will interpret it in the full form. Abbreviations are denoted by underlined letters in the Stata help system, so that an item presented like this indicates that this may be abbreviated to th or more characters in the full command. For example, the same results from above are obtained by any of the following commands:

```
summarize, detail
summar, detail
sum, det
su, d
```

We can also do simple tabulations for discrete variables. “tab” command will generate a simple one-way frequency table like the following.
. tab foreign

<table>
<thead>
<tr>
<th>Car type</th>
<th>Freq.</th>
<th>Percent</th>
<th>Cum.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Domestic</td>
<td>52</td>
<td>70.27%</td>
<td>70.27</td>
</tr>
<tr>
<td>Foreign</td>
<td>22</td>
<td>29.73%</td>
<td>100.00</td>
</tr>
<tr>
<td>Total</td>
<td>74</td>
<td>100.00%</td>
<td></td>
</tr>
</tbody>
</table>

Also, you can generate two-way tables like the following. Remember that tabulations are only useful for discrete variables with a few unique values.

. tab rep78 foreign

<table>
<thead>
<tr>
<th>Repair Record 78</th>
<th>Car type</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Domestic</td>
<td>Foreign</td>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>8</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>27</td>
<td>3</td>
</tr>
<tr>
<td>4</td>
<td>9</td>
<td>9</td>
</tr>
<tr>
<td>5</td>
<td>2</td>
<td>9</td>
</tr>
<tr>
<td>Total</td>
<td>48</td>
<td>21</td>
</tr>
</tbody>
</table>

Generating and Dropping Variables

To create a new variable you can use the command “generate”. To delete a variable you can use the command “drop”.

```
generate [filename]
drop [filename]
```

Recoding Variables

Suppose you want to recode the continuous variable price into a new variable with two categories such as “low” and “high”. Based on the summary of “price” variable, you want to assign those with below 5006.5 in the low group and those with higher than that into the high group.
The first way is to use “recode” command like the following

```
recode price (min/5006.5 = 1) (5006.5/max =2), generate(price2)
```

Inside the bracket, you specify how you want to recode the variable. (min/5006.5 =1) implies that you want observations between the minimum value to one with 5006.5 into the group of 1.

Then, you can use “tab” command how the new variable is created.

```
. tab price2
```

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>37</td>
<td>50.00</td>
<td>50.00</td>
</tr>
<tr>
<td>2</td>
<td>37</td>
<td>50.00</td>
<td>100.00</td>
</tr>
<tr>
<td>Total</td>
<td>74</td>
<td>100.00</td>
<td></td>
</tr>
</tbody>
</table>

The second way is to generate a new variable with missing values and replace them with specifying the conditions. In the example below, you create a variable “price2” with missing values and then, assign the value of 1 to those with the price lower than 5006.5 and the value of 2 to those with the price higher than 5006.5.
Labeling Variables

To label the variable proceeds in two steps. First, you need to create a label. Second, you assign the label to variables. This process is efficient especially when you assign the same label to multiple variables. To create a label you use “label define” command and to assign the label, you use “label values” command. In the first line, “category” refers to the name of the label you want to create. Thus, the command implies that you create a label category which names “low” to the group of observations with value 1 and “high” to the group of observations with the value 2. The second line implies that you attach the label “category” to the variable “price2”. You can see the outcome with “tab” command.

```
. label define category 1 "low" 2 "high"
. label values price2 category

. tab price2
```

<table>
<thead>
<tr>
<th>price2</th>
<th>Freq.</th>
<th>Percent</th>
<th>Cum.</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>37</td>
<td>50.00</td>
<td>50.00</td>
</tr>
<tr>
<td>2</td>
<td>37</td>
<td>50.00</td>
<td>100.00</td>
</tr>
<tr>
<td>Total</td>
<td>74</td>
<td>100.00</td>
<td></td>
</tr>
</tbody>
</table>

More resources

The Stata community is very user friendly, you can learn a lot from the various tutorials on the web, of which a widely used is the one at UCLA. StataCorp also maintains a list of online tutorials on the use of Stata, this can be accessed at [http://stata.com/links/resources1.html](http://stata.com/links/resources1.html).

The NYU Data Service Studio also provides introduction to Stata and various other statistical/mapping software, the scheduled can be viewed at [http://library.nyu.edu/forms/research/classes.html#dss](http://library.nyu.edu/forms/research/classes.html#dss).