Data Management and Analyses Using STATA: An Introductory Course

10 – 14 November 2014, Malawi Institute of Management (MIM)

1 We thank The Bill and Melinda Gates Foundation for funding the Training through the Guiding Investments in Sustainable Agricultural Intensification in Africa (GISAIA) Project.
2 Associate Professor and Director of Research and Outreach, Lilongwe University of Agriculture and Natural Resources (LUANAR), Bunda Campus, Malawi
3 PhD Scholar, Purdue University, USA
4 Research Fellow, Centre for Agricultural Research and Development (CARD), LUANAR, Bunda Campus, Malawi
# Table of Contents

1. Course Objectives .................................................................................................................................................4
2. Introducing Stata ..................................................................................................................................................4
   2.1 What is Stata ................................................................................................................................................4
   2.2 Why Use Stata? ............................................................................................................................................4
   2.3 Types of Stata ...............................................................................................................................................4
   2.4 What can Stata do? ......................................................................................................................................5
   2.5 The Stata Interface .......................................................................................................................................6
      2.5.1 Stata Windows: ...................................................................................................................................7
      2.5.2 The Tool Bars .......................................................................................................................................8
      2.5.3 Menus and dialogs ............................................................................................................................10
3. How to load your dataset from disk and save it to disk ......................................................................................10
   3.1 Reading Data into Stata ..............................................................................................................................10
   3.2 Saving data in Stata ....................................................................................................................................12
   3.3 Getting Help in Stata ..................................................................................................................................13
      3.3.1 Manuals .............................................................................................................................................13
      3.3.2 Stata In-Built Help and Website ........................................................................................................13
      3.3.3 The Web ............................................................................................................................................13
      3.3.4 Colleagues .........................................................................................................................................14
   3.4 Stata Documentation: Keeping Track of Things .........................................................................................14
      3.4.1 Do-file ................................................................................................................................................14
      3.4.2 Using logs ..........................................................................................................................................16
   3.5 Defining Stata Working Folder ...................................................................................................................17
4. Data Management ..............................................................................................................................................18
   4.1 The Data Editor ..........................................................................................................................................18
   4.2 Variable Manager .......................................................................................................................................19
   4.3 Labelling Data .............................................................................................................................................20
      4.3.1 Naming variables ...............................................................................................................................20
      4.3.2 Labeling variables ..............................................................................................................................20
      4.3.3 Labeling the various levels of a categorical variable ........................................................................21
4.4 Generating new variables from existing variables(s) ................................................................. 21
4.5 Changing string to numeric and vice versa .............................................................. 23
4.6 Merging Datasets ........................................................................................................... 24
  4.6.1 Merging Datasets for Latest Stata Versions *(11 and above)* ................................ 24
  4.6.2 Merging Data Sets for Older Stata Versions *(10 and below but works for newer versions as well)* .. 24
4.7 Appending datasets ........................................................................................................ 26
4.8 Collapsing Variables .................................................................................................... 26
4.9 Keep and drop ................................................................................................................ 27
5 Examining the Data ........................................................................................................... 28
  5.1 List .............................................................................................................................. 28
PRACTICAL SESSION 2: Data Management ........................................................................... 28
  5.2 Browse/Edit ................................................................................................................ 29
  5.3 Assert ........................................................................................................................ 30
  5.4 Describe ...................................................................................................................... 30
  5.5 Codebook .................................................................................................................. 31
  5.6 Summarize ................................................................................................................ 32
  5.7 Tabulate ...................................................................................................................... 32
  5.8 Inspect ......................................................................................................................... 33
  5.9 Graph ........................................................................................................................ 34
  5.10 Correlations ............................................................................................................ 34
  5.11 Hypothesis Testing .................................................................................................. 35
6 Regression Analyses ........................................................................................................... 36
  6.1 Estimation Procedure ................................................................................................ 36
PRACTICAL SESSION 3: Basic Data Analyses ........................................................................ 36
  6.2 Post-estimation .......................................................................................................... 38
  6.3 Prediction ..................................................................................................................... 38
7 Final Remarks ................................................................................................................ 39
PRACTICAL SESSION 4: Regression Analysis – Linear Regression (OLS) ......................... 39
PRACTICAL SESSION 5: Regression Analysis – Binary Logistic Regression ......................... 39
Appendices ......................................................................................................................... 40
1 Course Objectives

This course is the first level of a series of analysis using Stata. The course is designed to suit the needs of those who wish to acquire basic skills to analyze statistical data sets and produce technical reports. After completion of this course, the participant should be able to perform data entry, manipulation and some basic analysis using Stata.

Normally we combine this course with policy analysis and/or impact evaluation. Although in this course we only deal with Introduction to Stata, we still try to give a brief overview of policy analysis in Appendix 1 for the participant to appreciate the applicability of Stata in policy analysis.

2 Introducing Stata

2.1 What is Stata

Stata (pronounced “stay-tuh”) [most Malawians pronounce it “stah-tah” – still fine!] is a powerful statistical package with smart data-management facilities, a wide array of up-to-date statistical techniques, and an excellent system for producing publication-quality graphs. The word Stata is not an abbreviation but rather a corruption of the word Statistics. Stata is fast and easy to use. We will explore these Stata functionalities in this training course.

2.2 Why Use Stata?

There are numerous comparable statistical packages such as SPSS, R, SAS, Matlab, Eviews, etc. So the first question you should ask yourself is why should I use Stata? Stata’s main strengths are handling and manipulating large data sets (e.g. millions of observations!), and it has ever growing capabilities for handling panel and time-series regression analysis. The most recent (2014) version is Stata 13 and with each version there are improvements in computing speed, capabilities and functionality. It now has pretty flexible graphics capabilities. Furthermore, Stata is constantly being updated or advanced by users with a specific need – this means that even if a particular regression approach is not a standard feature, you can easily find someone on the web who has written a programme to carry out the analysis and this is easily integrated with your own software. In short, one Stata user summed up why they prefer Stata to other packages as, “a very interactive package, which makes you feel like you are talking to it and does exactly what you are telling it to do.”

2.3 Types of Stata
There are four different types (sizes) available for each version of Stata: Stata MP (Multi-Processor), which is the most powerful, Stata SE (Special Edition), Stata Intercooled (IC) and Stata Small. The main difference between these versions is the maximum number of variables, regressors and observations that can be handled. It is important to know these types if one is to make a good choice of what to buy. Most of us will be asked to advise our organization which type of Stata to buy. You will find this information handy. The table below summarises the characteristics of the four Stata types.

<table>
<thead>
<tr>
<th>Stata Type</th>
<th>Maximum Number of Variables</th>
<th>Maximum Number of Regressors</th>
<th>Maximum Number of Observations</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stata/MP</td>
<td>32,767</td>
<td>10,998</td>
<td>2,147,583,647*</td>
<td>• Runs on multiple CPUs or cores, from 2 to 64 but can also run on single core. The number of cores depends on the licence. • Fastest version of Stata</td>
</tr>
<tr>
<td>Stata/SE</td>
<td>32,767</td>
<td>10,998</td>
<td>2,147,583,647*</td>
<td>• Run on single core. • Can run on multiple core computers but uses only single core.</td>
</tr>
<tr>
<td>Stata/IC</td>
<td>2,047</td>
<td>798</td>
<td>2,147,583,647*</td>
<td>• Run on single core. • Can run on multiple core computers but uses only single core</td>
</tr>
<tr>
<td>Small Stata</td>
<td>99</td>
<td>99</td>
<td>1,200</td>
<td>• Run on single core. • Can run on multiple core computers but uses only single core</td>
</tr>
</tbody>
</table>

*Assuming you have enough memory.

Source: [www.stata.com](http://www.stata.com)

In this Training Course, we are going to use Stata/SE (Version 13)

2.4 What can Stata do?

Stata is a command-driven package. Although the newest versions also have pull-down menus from which different commands can be chosen, the best way to learn Stata is still by typing in
the commands. This has the advantage of making the switch to programming much easier for those doing serious econometric/statistical work. Moreover, it is the typing of commands that makes Stata more interactive and flexible than using pull-down menus. Arguably, you will never realize Stata’s full potential by using pull-down menus. However, sometimes the exact syntax of a command is hard to get right – in these cases, it is often advised to use the menu-commands to do it once and then copy the syntax, which is automatically inserted in the Command Window (see below) after executing any command (including from pull-down menus). Alternatively, use HELP to get a syntax (details on how to use Help later).

This section will introduce you to the Stata interface and the tasks that can be done in Stata. As you would expect, we will only brush the surface of many of these topics. This approach should give you a sample of what Stata can do and how Stata works. We will run through the section by using both menus and dialogs and Stata’s commands so that you can become familiar with them both.

Appendices 2a and 2c summarize some of the basic introductory features one need to know (most, if not all, of them already presented above).

### 2.5 The Stata Interface

[Also refer to Appendix 2c]

This diagram below introduces the core of Stata’s interface: its main windows, its toolbar, its menus, and its dialogs.

The windows:
2.5.1 Stata Windows:

The five main windows are the Review, Results, Command, Variables, and Properties windows. Except for the Results window, each window has its name in its title bar. These five windows are typically in use the whole time Stata is open. There are other, more specialized windows such as the Viewer, Data Editor, Variables Manager, Do-file Editor, Graph, and Graph Editor Windows.

a. **The commands window**: Commands are submitted to Stata from the Command window. The Command window supports basic text editing, copying and pasting, and a command history. The command history allows you to recall a previously submitted command, edit it if you wish, and then resubmit it. Commands submitted by Stata’s dialogs are also included in the command history, so you can recall and submit a command without having to open the dialog again.

b. **The Results window**: The Results window contains all the commands and their textual results you have entered during the Stata session. While you can scroll through the Results window to look at work you have done, it is much simpler to search within the Results window by using the find bar. By default, the find bar is hidden. You can expose it by selecting Edit > Find. You can clear out the Results window at any time by right-
clicking in the Results window and selecting Clear Results from the contextual menu. This action is not undoable.

c. **The Review window:** The Review window shows the history of commands that have been entered. It displays successful commands in black and unsuccessful commands, along with their error codes, in red. Filter button in the Review window titlebar toggles the visibility of these tools. Text entered in the Filter commands here field will filter the commands appearing in the Review window. By default, the filter will ignore case and find any commands containing any of the words in the filter. Clicking on the wrench on the left will allow you to change this behavior. Clicking on the exclamation mark button toggles the hiding of commands that ran with an error. No commands are deleted by using these tools—all that is affected is their visibility.

To enter a command from the Review window, you can click once on a past command to copy it to the Command window, replacing the contents of the Command window; or double-click on a past command to resubmit it. Executing the command adds the command to the bottom of the Review window. Right-clicking on the Review window displays a menu from which you can select various actions.

d. **The variables window:** The Variables window shows the list of variables in the dataset, along with selected properties of the variables. By default, it shows all the variables and their variable labels. You can change what properties get displayed by right-clicking on the header of any column of the Variables window. Click once on a variable in the Variables window to select it. Multiple variables can be selected in the usual fashion, either by Ctrl-clicking on nonadjacent variables or by clicking on a variable and shift-clicking on a second variable to select all intervening variables. Double-clicking on a variable in the Variables window puts the selected variable at the insertion point in the Command window. The Variables window supports filtering and reordering of variables. You can reorder the variables in the Variables window by clicking on any column header. Right-clicking on a variable in the Variables window displays a useful menu.

e. **The Properties windows:** The Properties window displays variable and dataset properties. If a single variable is selected in the Variables window, its properties are displayed. If there are multiple variables selected in the Variables window, the Properties window will display properties that are common across all selected variables.

To open any window or to reveal a window hidden by other windows, select the window from the Window menu, or select the proper item from the toolbar.

### 2.5.2 The Tool Bars

The toolbar contains buttons that provide quick access to Stata’s more commonly used features. If you forget what a button does, hold the mouse pointer over the button for a moment, and a tooltip will appear with a description of that button. Buttons that include both an icon and an arrow display a menu if you click on the arrow. Here is an overview of the toolbar buttons and their functions:
**Open**: opens a Stata dataset. Click on the button to open a dataset with the Open dialog.

**Save**: saves the Stata dataset currently in memory to disk.

**Print**: displays a list of windows. Select a window name to print its contents.

**Log**: begins a new log or closes, suspends, or resumes the current log.

**Viewer**: opens the Viewer or brings a Viewer to the front of all other windows. Click on the button to open a new Viewer. Click on the arrow to select a Viewer to bring to the front.

**Graph**: brings a Graph window to the front of all other windows. Click on the button to bring the Graph window to the front. Click on the arrow to select a Graph window to bring to the front.

**Do file Editor**: opens the Do-file Editor or brings a Do-file Editor to the front of all other windows. Click on the button to open a new Do-file Editor. Click on the arrow to select a Do-file Editor to bring to the front.

**Data Editor (Edit)**: opens the Data Editor or brings the Data Editor to the front of the other Stata windows.

**Data Editor (Browse)**: opens the Data Editor in browse mode

**Variables Manager**: opens the Variables Manager.
Clear more Condition: tells Stata to continue when it has paused in the middle of long output.

Break: stops the current task in Stata.

2.5.3 Menus and dialogs

There are two ways by which you can tell Stata what you would like it to do: you can use menus and dialogs, or you can use the Command window. Stata’s Data, Graphics, and Statistics menus provide point-and-click access to almost every command in Stata. You could type Stata’s regress command, or you could select Statistics > Linear models and related > linear regression

This dialog provides access to all the functionality of Stata’s regress command. The first time you use the dialog for a command, it is a good idea to look at the contents of each tab so that you will know all the dialog’s capabilities.

The dialogs for many commands have the by/if/in and Weights tabs. These provide access to Stata’s commands and qualifiers for controlling the estimation sample and dealing with weighted data.

The command issued by a dialog is submitted just as if you had typed it by hand. You can see the command in the Results window and in the Review window after it executes. Looking carefully at the full command will help you learn Stata’s command syntax.

In addition to being able to access the dialogs for Stata commands through Stata’s menus, you can also invoke them by using two other methods. You may know the name of a Stata command for which you want to see a dialog, but you might not remember how to navigate to that command in the menu system. Simply type db commandname to launch the dialog for commandname. For example db regress lunches the regress dialog box.

3 How to load your dataset from disk and save it to disk

[Also refer to Appendix 3]

3.1 Reading Data into Stata

There are numerous ways of reading data into Stata. Some of the ways are listed below. The first three are for data that are already in the Stata format; and the fourth one is for data in other formats.
1. Double click the Stata data file to load it into Stata
2. From the Stata interface, click on “file” and then on “open”. Use the “open” window to locate the file in your computer. Click on the file, and then click on “open” to load the file into Stata. After locating the file in your computer, via the “open” window, you can just double-click on the file to load it into Stata.
3. You can also use the “open” icon on the menu bar to do the same as 2) above.
4. To load files in other formats into Stata use the following procedure: click on “file” on the menu bar. In the file drop down menu, click on “Export” and then choose the format of your dataset. The import dialog box will appear. Choose the options in the dialog box as appropriate and click on “OK”. The import dialog box for importing Excel files, for example, is shown below.

5. The command window can also be used to load data into Stata using the `use, import` command for data in Stata format; and the `import and insheet` commands for the other data formats.

Examples:

a. `use "C:\Users\user\Desktop\data_phd\New data2\responserate.dta", clear`

The clear option will clear any dataset currently in memory before opening the new dataset.
b. `import excel "C:\Users\user\Desktop\data_phd\New data2\responserate.xls", sheet("Sheet1") firstrow`

The sheet("Sheet1") option indicates that the data is in the first sheet of the excel file. The firstrow option indicates that first row of the data has the variable names.

c. `insheet using "C:\Users\user\Desktop\data_phd\New data2\responserate.csv"`

Note: The `insheet` command appears to work only for csv files.

Because Stata has at most one dataset open at a time, opening a dataset will cause Stata to discard the dataset that is currently in memory. If there have been changes to the data in the currently open dataset, Stata will refuse to discard the dataset unless you force it to do so. If you open the file with any method other than the Command window, you will be prompted. If you use the Command window and the current data have changed, you will get the following error message:

```
. sysuse auto
  no; data in memory would be lost
  r(4);
```

These behaviors protect you from mistakenly losing data.

To save an unnamed dataset (or an old dataset under a new name):
1. select File > Save As...; or
2. type `save filename` in the Command window.

To save a dataset for use with Stata 11 or Stata 12 (Stata 11 can load Stata 12 datasets),
1. select File > Save As..., and select Stata 12 Data (*.dta) from the Save as type list; or
2. type `saveold filename` in the Command window.

### 3.2 Saving data in Stata

To save a dataset that has been changed (overwriting the original data file),
1. select File > Save;
2. click on the Save button; or
3. type `save, replace` in the Command window.

Once you overwrite a dataset, there is no way to recover your original dataset. With important datasets, you may want to either keep a backup copy of your original filename.dta or save your changes to a dataset under a new name. This is no different from working with a word-processing document, except that recovering from an inadvertent save of a dataset is nearly impossible.
Important note: Changes you have made to a dataset are not permanent until you save them. You work with a copy of the dataset in memory, not with the data file itself. This should not be surprising, because it is the way that you work with almost all applications on your computer. If you do not want to save your dataset, you can clear the dataset in memory and open a new dataset by typing use filename, clear.

To save a dataset for the first time

1. select **File > Save As**... and use the Save As dialog box to complete the saving.

### 3.3 Getting Help in Stata

As already pointed out, Stata is a command-driven language – there are over 500 different commands and each has a particular syntax to get any various options. Learning these commands can be time-consuming but not hard. There is no way this Introductory Course can cover all the commands. Luckily, Stata has a fantastic option for getting help. In fact, what we will do in this is simply to introduce you to various operational aspects of Stata. From here, most of your learning to use Stata will be self-teaching by using manuals, colleagues and Stata’s own ‘help’ function.

#### 3.3.1 Manuals

Stata comes with User Manuals, which provide an overall view on using Stata. There are several separate manuals for special topics such as Graphics Manual, Panel Data Manual or Survey Data. You can get these manuals either as hardcopies or you can access them as part of the help command elaborated below.

#### 3.3.2 Stata In-Built Help and Website

Stata also has an abbreviated version of its manuals built-in. Clicking **Help**, then **Contents**, will display an outline of the major topics of help. In addition, Stata’s website has a very useful Frequently Asked Questions (FAQ) section at [http://www.stata.com/support/faqs/](http://www.stata.com/support/faqs/). Alternatively, we can type command **help [topic]** to get help on a specific command syntax and description. We will be demonstrating how this is done as we progress in the training. Stata’s website also has a list of helpful links at [http://www.stata.com/links/resources-for-learning-stata/](http://www.stata.com/links/resources-for-learning-stata/).

#### 3.3.3 The Web

As with everything nowadays, the web is a great place to look to resolve problems. Google should help you here. Stata has an excellent website at [http://www.stata.com](http://www.stata.com). Among other things you will find that they make available online all datasets used in the official
documentation, that they publish a journal called Stata Journal, and that they have an excellent bookstore with texts on Stata and related statistical subjects. Stata also offers email and web-based training courses called NetCourses, see http://www.stata.com/netcourse/. In addition, there are numerous chat-rooms about Stata commands, and plenty of authors put new programmes on their websites.

There is an independent listserv at the Harvard School of Public Health, where you can post questions and receive prompt and knowledgeable answers from other users. For detailed instructions on how to join the list see http://www.stata.com/statalist/ and follow the link to subscribe. The postings are archived by Stata, Harvard University and Yahoo.

UCLA maintains an excellent Stata portal at http://www.ats.ucla.edu/stat/stata/, with many useful links, including a list of resources to help you learn and stay up-to-date with Stata. Don't miss their starter kit, which includes "class notes with movies", a set of instructional materials that combine class notes with movies you can view on the web, and their links by topic, which provides how-to guidance for common tasks. There are also more advanced learning modules, some with movies as well, and comparisons of Stata with other packages such as SAS and SPSS.

3.3.4 Colleagues

The other place where you can learn a lot is from speaking to colleagues more familiar with Stata. Sometimes it is funny how one can spend several hours or days trying to crack a problem when help is just a phone call away or just in the next office! Do not be shy or hesitate to ask for help from colleagues.

3.4 Stata Documentation: Keeping Track of Things

Stata has a number of tools to help you keep track of what work you did to datasets, what’s in the datasets, and so on.

3.4.1 Do-file

Stata comes with an integrated text editor called the Do-file Editor, which can be used for many tasks. It gets its name from the term do-file, which is a file containing a list of commands for Stata to run (called a batch file or a script in other settings).

Do-file Editor has advanced features that can help in writing such files, it can also be used to build up a series of commands that can then be submitted to Stata all at once. This feature can be handy when writing a loop to process multiple variables in a similar fashion or when doing complex, repetitive tasks interactively.
A do-file can be launched by either clicking on the Do-file editor button or by typing `doedit` in the command window.

Do-files have a toolbar as shown below:

![Do-file Editor Toolbar](image.png)

- **New**: Open a new do-file in a new tab in the Do-file Editor.
- **Open**: Open a do-file from disk in a new tab in the Do-file Editor.
- **Save**: Save the current do-file to disk.
- **Print**: Print the contents of the Do-file Editor.
- **Find**: Open the Find dialog for finding text.
- **Cut**: Cut the selected text and put it in the Clipboard.
- **Copy**: Copy the selected text to the Clipboard.
- **Paste**: Paste the text from the Clipboard into the current document.
- **Undo**: Undo the last change.
- **Redo**: Undo the last undo.
- **Toggle Bookmark**: Turn on or off the bookmark on the current line. Bookmarks are a way to move quickly within the do-file. They are quite useful in long do-files or when debugging.
- **Previous Bookmark**: Go to the previous bookmark (if any).
- **Next Bookmark**: Go to the next bookmark (if any).
- **Show File in Viewer**: Show the contents of the do-file in a Viewer window. This is worthwhile when editing files that contain SMCL tags, such as log files or help files.
- **Execute (do)**: Run the commands in the do-file, showing all commands and their output. If text is highlighted, the button becomes the Execute Selection (do) button and will run only the selected lines, showing all output. We will refer to this as the Do button.

Apart from working on a do file directly and saving it, as discussed earlier, while working interactively with Stata, you can send highlighted commands or even the entire contents to the Do-file Editor from the Review window.

The diagram below shows how a do-file looks like:
You will notice that the color of the text changes as you type (this is true for Stata 11 upwards). The different colors are examples of the Do-file Editor’s syntax highlighting. The colors and text properties of the syntax elements can be changed by selecting Edit > Preferences... from the Do-file Editor menu bar and then clicking on the Syntax Color tab in the resulting window. Here are some common [default] features of a do-file:

- The green color indicates a description of a procedure as written by the user. If you want describe anything you are doing (for future reference), precede the statement, word or phrase by an asterisk (*).
- Alternatively, if you want to describe something in the same line as the command, you may use double slash (//) but beware that using three slashes (///) has another use, i.e. creating a loop (probably beyond the scope of this course).
- The blue color is for a command while words between inverted commas (“ ”) are red.

3.4.2 Using logs

[Also refer to Appendix 4]
When you work on an analysis, it is worthwhile to behave like a bench scientist and keep a lab notebook of your actions so that your work can be easily replicated. Everyone has a feeling of complete omniscience while working intensely—this feeling is wonderful but fleeting. The next day, the exact small details needed for perfect duplication have become obscure. Stata has a lab notebook at hand: the log file.

A log file is simply a record of your Results window. It records all commands and all textual output as it happens. Thus it keeps your lab notebook for you as you work. Because it writes the file to disk while it writes the Results window, it also protects you from disastrous failures, be they power failures or computer crashes. It is recommend that you start a log file whenever you begin any serious work in Stata.

All the output that appears in the Results window can be captured in a log file. Stata can save the file in one of two different formats. By default, Stata will save the file in its Stata Markup and Control Language (SMCL) format, which preserves all the formatting and links from the Results window. You can open these results in the Viewer, and they will behave as though they were in the Results window. If you would rather have plain-text files without any formatting, you can save the file as a plain log file. We recommend using the SMCL format because SMCL files can be translated into a variety of formats readable by applications other than Stata with the File > Log > Translate... menu

To start a log file, click on the Log button. This will open a standard file dialog that allows you to specify a directory and filename for your log. If you do not specify a file extension, the extension .smcl will be added to the filename.

If you click on the log file icon while a log file is active, you will be asked whether you want to view a snap shot, suspend or close the log file append the new log to the file or overwrite the file with the new log.

If you choose an existing log file, you will be asked whether you want to view, append or overwrite the existing log file.

You can view the log file using the Viewer window in two ways:
1. Viewer > file > open...>Browse
2. File > log> view...

To print a standard SMCL log file, you need to have the log file open in a Viewer window. Once the log file is in the Viewer, you can click on the Print button, right-click on the Viewer window, and select Print..., or select File > Print. A Print dialog will appear. After you click on Print, an Output Settings dialog will appear.

### 3.5 Defining Stata Working Folder

Stata works with one folder at a time. This has to be pre-defined within each working session. Otherwise, Stata will work in the default installation folder, which in most cases is My Documents. As you will notice, with time, working with Stata leads to creation of so many files
and if these are not properly stored, it becomes difficult to access them later on (this is true for all other computer applications). Another advantage of pre-defining the working folder is that all the files saved and created within a working session go to that folder. It is therefore, highly recommended that we define the route of the working folder for each working session by using the `cd` (change directory) command. For example, if we have created a working folder called “Stata Training” in “My Documents” in Drive D, our working folder will be defined by giving the whole route to the “Stata Training” as follows:

```
    cd "d:\My Documents\Stata Training"
```

The above command can be typed directly in the Command Window or Do-File. However, you are likely to make some mistakes when typing as in some cases the route can be very long (and Stata is very sensitive to typing mistakes no matter how small). To avoid this, it recommended to just copy the folder route from the File Explorer and paste to the Command Window or Do-file (to be illustrated during practicals).

---

### PRACTICAL SESSION 1: Exercises on Syntaxes

You will be provided with Dataset(s) to have a hands-on experience with Stata Environment. Among other things, you will be guided to create your own do-files, log-files, define a working folder, opening, importing and saving data files, etc.

---

## 4 Data Management

### 4.1 The Data Editor

The Data Editor gives a spreadsheet-like view of data that are currently in memory. You can use it to enter new data, edit existing data, and edit attributes of the data in the dataset, such as variable names, labels, and display formats, as well as value labels. In addition to the view of the data, there are two windows for manipulating variables and their properties: the Variables window and the Properties window. These are similar to the same-named windows in the main Stata window.

Any action you take in the Data Editor results in a command being issued to Stata as though you had typed it into the Command window. This means that you can keep good records and learn commands by using the Data Editor.

The Data Editor can be kept open while you work in Stata, giving you a live view of your dataset as you work. To protect your data from inadvertent changes, the Data Editor has two modes: edit mode for active editing and browse mode for viewing. In browse mode, editing within the Data
Editor window is disabled. It is highly recommend that you use the Data Editor in browse mode and switch to edit mode only when you want to make changes.

The toolbar for the Data Editor has some standard buttons and some buttons we have not yet seen:

- **Open**: Opens a Stata dataset. Stata will warn you if your current dataset has unsaved changes.
- **Save**: Saves the dataset visible in the Data Editor.
- **Copy**: Copies the current selection to the Clipboard.
- **Paste**: Pastes the contents of the Clipboard. You may paste only if one cell is selected—this cell will become the upper-left corner of the pasted contents. Warning: This action will paste over existing data.
- **Edit Mode**: Changes the Data Editor to edit mode.
- **Browse Mode**: Changes the Data Editor to browse mode for safely looking at data.
- **Filter Observations**: Filters the observations visible in the Data Editor. This button is useful for looking at a subset of the current dataset.

Right-clicking within the Data Editor brings up a contextual menu that allows you to manipulate the data and what you are viewing. Right-clicking on the Data Editor window displays a menu from which you can do many common tasks:

The purpose of using the Data Editor in browse mode is to look at data without altering them by stray keystrokes. You can start the Data Editor in browse mode by clicking on the Data Editor (Browse) button or by typing browse in the Command window. When you work in browse mode, all contextual menu items that would let you alter the data, the labels, or any of the display formats for the variables are disabled. You may view a variable’s properties with the Variable Properties menu item, but you may not make any changes. You still can filter observations and hide variables to get a restricted view because these actions do not change the dataset.

Note: Because you can still use Stata menus not related to the Data Editor and because you can still type commands in the Commands window, it is possible to change the data even if the Data Editor is in browse mode. In fact, this means you can watch how your commands affect the dataset. You are merely restricted from using the Data Editor itself to change the data.

### 4.2 Variable Manager

The Variables Manager is a tool for managing properties of variables both individually and in groups. It can be used to create variable and value labels, rename variables, change display
formats, and manage notes. It has the ability to filter and group variables as well as to create variable lists. Users will find these features useful for managing large datasets. Any action you take in the Variables Manager results in a command being issued to Stata as though you had typed it in the Command window. This means that you can keep good records and learn commands by using the Variables Manager.

You open the Variables Manager by selecting **Data > Variables Manager** or clicking on the Variables Manager button.

### 4.3 Labelling Data

This chapter discusses, in brief, naming of variables and labeling of variables and values. Such naming and labeling is critical to careful use of data. Labeling variables with descriptive names clarifies their meanings. Labeling values of numerical categorical variables ensures that the real-world meanings of the encodings are not forgotten. These points are crucial when sharing data with others, including your future self. Labels are also used in the output of most Stata commands, so proper labeling of the dataset will produce much more readable results.

#### 4.3.1 Naming variables

Variable names can have up to 32 characters, but many commands print only 12, and shorter names are easier to type. Stata names are *case sensitive*, Age and age are different variables! It pays to develop a convention for naming variables and sticking to it. It helps to use short lowercase names and single words or abbreviations rather than multi-word names, for example use effort or fpe to family_planning_effort or familyPlanningEffort, although all four names are legal. Note the use of underscores to separate words.

#### 4.3.2 Labeling variables

Variables can be labeled using the following Stata syntax

```stata
label variable var1 "description"
```

where var1 is the variable to be labeled; and description is the label of var1

Dialog Box: **Data > Variables Manager**

Example: Label a variable called nation_maize as total maize production in the nation

```stata
label variable nation_maize "total maize production in the nation"
```
4.3.3 Labeling the various levels of a categorical variable

The various levels of a categorical variable can be labeled using the following two Stata syntaxes together:

\textit{label} define \textit{var1} 1 “name of the first category” 2 “name of the second category”

\textit{label} values var1 var1

Where var1 is the name of the categorical variable; and 1 and 2 are the levels of the categorical variable.

Example: A variable called \textit{gender} has two categories – 1 for male and 2 for female. The categories if gender can be labeled as follows:

\textit{label} define \textit{gender} 1 “male” 2 “female”

\textit{label} values \textit{gender} \textit{gender}

4.4 Generating new variables from existing variables(s)

The two most common commands for creating new variables are \textit{generate} and \textit{egen} depending on the definition of the new variable

Syntax: \textit{generate} new\_variable = expression

Example: Generate a variable called \textit{income} which is the sum of farm income (\textit{fincome}) and non-farm income (\textit{nfincome}):

\textit{generate} income = fincome + nfincome

\textbf{Dialog}: Data > Create or change data > Create new variable
There are some details you should know about the `generate` command:

1. You will get an error message if you try to generate a variable that already exists.
2. An algebraic calculation using a missing value yields a missing value, as does division by zero, the square root of a negative number, or any other computation which is impossible.
3. If missing values are generated, the number of missing values in new variable is always reported. If Stata says nothing about missing values, then no missing values were generated.

The `egen` command typically creates new variables based on summary measures, such as sum, mean, min and max. It is useful for working across groups of variables or within groups of observations.

Syntax: `egen newvar = fcn(arguments)`

Where `fcn` refers to function. Depending on the `fcn`, `arguments` refers to an expression, a list of variables, or a list of numbers. Some notable functions include: `rowmin` (gives the row minimum of the variables in `varlist`); `rowmax` (gives the row maximum of the variables in `varlist`); `rowmean` (gives the row mean of the variables in `varlist`); `rowmedian` (gives the row median of the variables in `varlist`); `rowtotal` (gives the row total of the variables in `varlist`) etc. `egen` usually ignore missing values in the dataset.

Example: generate a variable called `nation_maize` which is the sum of the maize output from the three regions of Malawi, `north_maize`, `central_maize` and `south_maize`. The `egen` command is as follows:

```
egen nation_maize = rowtotal(north_maize, central_maize and south_maize)
```

Dialog: Data > Create or change data > Create new variable (extended)
Another command that is frequently used with the `generate` command is the `replace` command. `replace` modifies the content of a variable by replacing some or all of its content.

Syntax: `replace var1 = a if var1==b`

where `var1` is the variable to be modified; `a` is the replacement; and `b` is the content of `var1` that is to be replaced.

Example: modify a variable called `maize_output` by replace the missing values with 0.

`replace maize_output = 0 if maize_output==.`

The contents of a variable can also be replace with contents of another variable or the results of an expression.

**Dialog:** Data > Create or change data > Change contents of variable

### 4.5 Changing string to numeric and vice versa

The `destring` and `tostring` commands are used to change string variables to numeric, and numeric variables to string variables respectively.

Syntax: 1) `destring var1, generate (var2)`

2) `destring var1, replace`

The first syntax converts the string variable, `var1`, to a numeric variable, calling the newly created numeric variable `var2`. The second syntax converts the string variable `var1` to a numeric variable by replacing the string data with its numeric data.

**Example:** change a string variable called `string_var` to a numeric variable (remember you can either give the newly generated numeric variable a name, say `numeric_var`, or keep the same name but replace the string data to numeric data)

`destring string_var, generate (numeric_var)`

`destring string_var, replace`

**Note 1:** The `tostring` command has the same syntax as the `destring` command
Note 2: A string variable can only be “destringed” if the data is made up of numbers. Thus the destring command will not work on string variable in which the data has some letters or words.

4.6 Merging Datasets

[Also refer to Appendix 5]

4.6.1 Merging Datasets for Latest Stata Versions (11 and above)

The `merge` command joins corresponding observations from the dataset currently in memory (called the master dataset) with those from a second dataset (called the using dataset), matching on one or more key variables. The `merge` command can perform one-to-one, one-to-many, many-to-one, and many-to-many matches.

The Syntaxes below apply to Stata 11 and above (for Stata 10 and below see the next section):

One-to-one merge on specified key variables

```
merge 1:1 varlist using “location and name of second file”, keepusing(vars)
```

Many-to-one merge on specified key variables

```
merge m:1 varlist using “location and name of second file”, keepusing(vars)
```

One-to-many merge on specified key variables

```
merge 1:m varlist using “location and name of second file”, keepusing(vars)
```

Many-to-many merge on specified key variables

```
merge m:m varlist using “location and name of second file”, keepusing(vars)
```

Where `varlist` is the name(s) of the key variables on which the matching will be done. The `keepusing` option specifies the variables in the `using` dataset that you are interested in merging onto the dataset in memory. All the variables in the `using` dataset will be merged if the `keepusing` option is not specified.

4.6.2 Merging Data Sets for Older Stata Versions (10 and below but works for newer versions as well)

For Stata 10 and below, like above you also need to determine which one will be your master and using files. However, you do not need the 1:1, m:1, etc. All you need is to sort the identifying variable (`id`) in both master and using files. The syntax then simply become:
sort id

merge id using filename

The above syntax assumes you have already defined your working folder as described in section 2.5. Otherwise, if you haven’t defined your working folder, the syntax should be

sort id

merge id using “location and name of second file”

The older syntax also works for upper versions (11 and up) but you will always get a notification that you are using an old syntax, e.g:

    . merge houscode using env_forest_income_by_hh.dta
       (note: you are using old merge syntax; see [D] merge for new syntax)

Do not get scared if you get the above message. The merging works just fine. Most people would find the old syntax easier. Use whatever is easier for you as long as you follow the right procedure.

After merging, it is always important to check how the files have been merged. For this purpose, Stata automatically generates a variable _merge (note the underscore). The _merge variable has numbers 1 up to 3, where:

- 1 indicates the number of cases from the master file not successfully merged (sometimes for a good reason but always try to figure out why);
- 2 indicates the number of cases from the using file not successfully merged (also might be for a good reason but always try to figure out why); and
- 3 indicates the number of cases that have been successfully merged.

To check the merging outcome, tabulate (tab) the _merge variable as follows:

    tab _merge

Below is an example of an output for tab _merge:
The above result shows that out of the 259 total observations, only 137 could be matched; 66 and 56 observations from the master file and using file respectively were not matched. Otherwise, if all the observations were perfectly matched. In the practical sessions we will explore some of the reasons why some observations can fail to match perfectly.

Finally, we need to drop the _merge variable. Otherwise, no further merging to the same merged file can be done.

\textit{drop} \_merge

4.7 Appending datasets

[Also refer to Appendix 5]

The \texttt{append} command appends a Stata dataset to the end of the dataset in memory.

Syntax: \texttt{append using "location and name of second file"}

\textbf{Dialog:} Data > Combine datasets > Append datasets

Appending two datasets require that both have exactly the same number of variables with xactly the same names. If using categorical data, make sure the categories on both datasets refer to exactly the same thing (e.g. 1 “Agree”, 2 “Disagree”, 3 “Don’t know” on both files).

4.8 Collapsing Variables

This command converts the data into a dataset of summary statistics, such as sums, means, medians, and so on.

Syntax: \texttt{collapse} (sum_stat) varname, \textit{by}(categorical_var)

\begin{verbatim}
. tab _merge

<table>
<thead>
<tr>
<th></th>
<th>Freq.</th>
<th>Percent</th>
<th>Cum.</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>66</td>
<td>25.48</td>
<td>25.48</td>
</tr>
<tr>
<td>2</td>
<td>56</td>
<td>21.62</td>
<td>47.10</td>
</tr>
<tr>
<td>3</td>
<td>137</td>
<td>52.90</td>
<td>100.00</td>
</tr>
<tr>
<td>Total</td>
<td>259</td>
<td>100.00</td>
<td></td>
</tr>
</tbody>
</table>
\end{verbatim}

The above result shows that out of the 259 total observations, only 137 could be matched; 66 and 56 observations from the master file and using file respectively were not matched. Otherwise, if all the observations were perfectly matched. In the practical sessions we will explore some of the reasons why some observations can fail to match perfectly.

Finally, we need to drop the _merge variable. Otherwise, no further merging to the same merged file can be done.

\textit{drop} \_merge

4.7 Appending datasets

[Also refer to Appendix 5]

The \texttt{append} command appends a Stata dataset to the end of the dataset in memory.

Syntax: \texttt{append using "location and name of second file"}

\textbf{Dialog:} Data > Combine datasets > Append datasets

Appending two datasets require that both have exactly the same number of variables with xactly the same names. If using categorical data, make sure the categories on both datasets refer to exactly the same thing (e.g. 1 “Agree”, 2 “Disagree”, 3 “Don’t know” on both files).

4.8 Collapsing Variables

This command converts the data into a dataset of summary statistics, such as sums, means, medians, and so on.

Syntax: \texttt{collapse} (sum_stat) varname, \textit{by}(categorical_var)
Sum_stat can be any of the statistics in the table below; varname is the name of the variable to be collapsed; and categorical_var is the name of the categorical variable over which the variable will be collapsed.

<table>
<thead>
<tr>
<th>Statistic</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>mean, medians</td>
<td>Means (default)</td>
</tr>
<tr>
<td>p1, p2, ...</td>
<td>1st percentile, 2nd percentile, ...</td>
</tr>
<tr>
<td>p50</td>
<td>50th percentile (same as median)</td>
</tr>
<tr>
<td>p95</td>
<td>95th percentile</td>
</tr>
<tr>
<td>sd, semean</td>
<td>Standard deviation, standard error of the mean (sd/sqrt(n))</td>
</tr>
<tr>
<td>sembinomial, apoisson</td>
<td>Standard error of the mean, binomial (sqrt(p(1-p)/n))</td>
</tr>
<tr>
<td>sum</td>
<td>Sum</td>
</tr>
<tr>
<td>rawsum</td>
<td>Sum, ignoring optionally specified weight except observations with a weight of zero are excluded</td>
</tr>
<tr>
<td>count</td>
<td>Number of nonmissing observations</td>
</tr>
<tr>
<td>max, min, 1q, 3q</td>
<td>Maximum, minimum, interquartile range</td>
</tr>
<tr>
<td>first, last</td>
<td>First value, last value</td>
</tr>
<tr>
<td>firstnvm, lastnvm</td>
<td>First nonmissing value, last nonmissing value</td>
</tr>
</tbody>
</table>

Example 1: Say you have a monthly dataset that you want to aggregate to annual data:

```
collapse (sum) monthoutput, by(year)
```

Example 2: Say you have a firm-level data that you want to aggregate to industry level:

```
collapse (sum) firmoutput, by(industry)
```

Note that if your dataset contains other variables beside the indicator variables and the variables you are collapsing, they will be erased.

One possible problem that arises in the use of collapse is in its treatment of missing values. Stata will ignore missing values in the calculation of summary statistics. Sometimes it is important to know the number of observations that was used to create an average (i.e., how many nonmissing observations we have). This can be achieved by adding a variable and using the rawsum statistic. The rawsum statistic ignores weights, so the collapsed variable will tell us the number of observations.

### 4.9 Keep and drop

The original dataset may contain variables you are not interested in or observations you don’t want to analyze. It’s a good idea to get rid of these first – that way, they won’t use up valuable memory and they won’t inadvertently sneak into your analysis. You can tell Stata to either keep what you want or drop what you don’t want – the end results will be the same. For example, we can get rid of unwanted variables as follows:

```
keep yield fert seed labor hybrid
drop organic soil_good soil_fair
```
You can also drop or keep observations using the *drop or keep* command together with the *if* command.

```
keep if yield >= 0
drop if yield < 0
```

### PRACTICAL SESSION 2: Data Management

You will be provided with Dataset(s) to have a hands-on experience with data management. Among other things, you will be guided to recode, (re)name, label, create and collapse variables etc. You will also practice how you can enter and edit data in addition to importing and exporting different formats of data. Finally, we will practice how we can merge or append datasets from different files.

---

## 5 Examining the Data

It is a good idea to examine your data when you first read it into Stata – you should check that all the variables and observations are present and in the correct format.

### 5.1 List

As we have seen, the browse and edit commands start a pop-up window in which you can examine the raw data. You can also examine it within the results window using the *list* command – although listing the entire dataset is only feasible if it is small. If the dataset is large, you can use some options to make the output of list more tractable.

The *list* command displays the values of variables. If no varlist is specified, the values of all the variables are displayed.

**Syntax:** `list varlist, options`

Where varlist is the list of variables to be listed; and options is any or a combination of any of the options associated with the *list* command. The table below shows the option command associated with the *list* command.
5.2 Browse/Edit

The browse command is similar to edit, except that modifications to the data by editing in the grid are not permitted. browse is a convenient alternative to the list command.

The edit command brings up a spreadsheet-style data editor for entering new data and editing existing data. The difference with edit is that this allows you to manually change the dataset.

Syntaxes:

Edit using Data Editor

edit varlist, nolabel

Browse using Data Editor

browse varlist, nolabel

nolabel causes the underlying numeric values, rather than the label values (equivalent strings), to be displayed for variables with value labels
Most of the time we only want to view a few variables at a time, especially in large datasets with a large number of variables. The `varlist` in the syntaxes, help in such cases. Simply list the variables you want to examine after the `edit` and `browse` commands.

### 5.3 Assert

With large datasets, it often is impossible to check every single observation using `list` or `browse`. Stata has a number of additional commands to examine data which are described in the following. A first useful command is `assert` which verifies whether a certain statement is true or false.

Syntax: `assert` expression

For example, you might want to check whether all values in the `yield` variable are nonnegative as they should be:

```
assert yield !< 0  or  assert yield >= 0
```

If the statement is true, `assert` does not yield any output on the screen. If it is false, `assert` gives an error message and the number of contradictions.

### 5.4 Describe

The `describe` command produces a summary of the dataset in memory or of the data stored in a Stata-format dataset.

Syntax:

Describe data in memory

```
describe varlist, memory_option
```

Describe data in file

```
describe varlist using “location and name of the file”, file_options
```
5.5 Codebook
The codebook command examines the variable names, labels, and data to produce a codebook describing the dataset.

Syntax: codebook varlist, options

Options
- all: print complete report without missing values
- header: print dataset name and last saved date
- notes: print any notes attached to variables
- mv: report pattern of missing values
- tabulate(#) set tables/summary statistics threshold; default is tabulate(9)
- problems: report potential problems in dataset
- detail: display detailed report on the variables; only with problems
- compact: display compact report on the variables
- dots: display a dot for each variable processed; only with compact

Codebook without a list of variables will give information on all variables in the dataset.

Dialog: Data > Describe data > Describe data contents (codebook)
5.6 Summarize

This provides summary statistics, such as means, standard deviations, and so on.

Syntax: `summarize, options`

<table>
<thead>
<tr>
<th>options</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>detail</code></td>
<td>display additional statistics</td>
</tr>
<tr>
<td><code>meanonly</code></td>
<td>suppress the display; calculate only the mean; programmer’s option</td>
</tr>
<tr>
<td><code>format</code></td>
<td>use variable’s display format</td>
</tr>
<tr>
<td><code>separator(#)</code></td>
<td>draw separator line after every # variables; default is separator(5)</td>
</tr>
<tr>
<td><code>display_options</code></td>
<td>control spacing, line width, and base and empty cells.</td>
</tr>
</tbody>
</table>

5.7 Tabulate

The `tabulate` command is a versatile command that can be used, for example, to produce a frequency table of one variable or a cross-tab of two variables.

Syntax: 1) `tabulate varname, options`

Syntax: 2) `tabulate varname1 varname2, options`
Dialogs for syntax 1):

Statistics > Summaries, tables, and tests > Frequency tables > One-way table

Statistics > Summaries, tables, and tests > Frequency tables > Multiple one-way tables

Dialogs for syntax 2):

Statistics > Summaries, tables, and tests > Frequency tables > Two-way table with measures of association

Statistics > Summaries, tables, and tests > Frequency tables > All possible two-way tables

5.8 Inspect

The *inspect* command is a way to eyeball the distribution of a variable, including as it does a mini-histogram. It is also useful for identifying outliers or unusual values, or for spotting non-integers in a variable that should only contain integers. The command provides a quick summary of a numeric variable that differs from the summary provided by summarize or tabulate. It reports the number of negative, zero, and positive values; the number of integers and nonintegers; the number of unique values; and the number of missing; and it produces a small
histogram. Its purpose is not analytical but is to allow you to quickly gain familiarity with unknown data.

Syntax: `inspect varlist`

**Dialog**: Data > Describe data > Inspect variables

## 5.9 Graph

The `graph` commands draws graph

The table below provides the syntaxes for various graphs in Stata

The commands that draw graphs are

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>graph twoway</code></td>
<td>scatterplots, line plots, etc.</td>
</tr>
<tr>
<td><code>graph matrix</code></td>
<td>scatterplot matrices</td>
</tr>
<tr>
<td><code>graph bar</code></td>
<td>bar charts</td>
</tr>
<tr>
<td><code>graph dot</code></td>
<td>dot charts</td>
</tr>
<tr>
<td><code>graph box</code></td>
<td>box-and-whisker plots</td>
</tr>
<tr>
<td><code>graph pie</code></td>
<td>pie charts</td>
</tr>
<tr>
<td><code>other</code></td>
<td>more commands to draw statistical graphs</td>
</tr>
</tbody>
</table>

The commands that save a previously drawn graph, redisplay previously saved graphs, and combine graphs are

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>graph save</code></td>
<td>save graph to disk</td>
</tr>
<tr>
<td><code>graph use</code></td>
<td>redisplay graph stored on disk</td>
</tr>
<tr>
<td><code>graph display</code></td>
<td>redisplay graph stored in memory</td>
</tr>
<tr>
<td><code>graph combine</code></td>
<td>combine multiple graphs</td>
</tr>
</tbody>
</table>

The “Graph” tab on the manu bar can also be used.

## 5.10 Correlations

Correlation measures association/relationship between variables. The `correlate` command displays the correlation matrix or covariance matrix for a group of variables. The `pwcorr` displays all pairwise correlation coefficients between variables. The syntaxes for these two commands are:
**correlate variable_list, correlate_options**

**pwcorr variable_list, pwcorr_options**

correlate_options Description

<table>
<thead>
<tr>
<th>Options</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>means</td>
<td>display means, standard deviations, minimums, and maximums</td>
</tr>
<tr>
<td>noformat</td>
<td>ignore display format associated with variables</td>
</tr>
<tr>
<td>covariance</td>
<td>display covariances</td>
</tr>
<tr>
<td>wrap</td>
<td>allow wide matrices to wrap</td>
</tr>
</tbody>
</table>

**pwcorr_options Description**

<table>
<thead>
<tr>
<th>Main</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>obs</td>
<td>print number of observations for each entry</td>
</tr>
<tr>
<td>sig</td>
<td>print significance level for each entry</td>
</tr>
<tr>
<td>listwise</td>
<td>use listwise deletion to handle missing values</td>
</tr>
<tr>
<td>casewise</td>
<td>synonym for listwise</td>
</tr>
<tr>
<td>print(#)</td>
<td>significance level for displaying coefficients</td>
</tr>
<tr>
<td>star(#)</td>
<td>significance level for displaying with a star</td>
</tr>
<tr>
<td>bonferroni</td>
<td>use Bonferroni-adjusted significance level</td>
</tr>
<tr>
<td>sidak</td>
<td>use Sidak-adjusted significance level</td>
</tr>
</tbody>
</table>

[Also refer to Appendix 6 for a summary of data manipulation techniques]

### 5.11 Hypothesis Testing

The results of each estimation automatically include for each independent variable a two-sided t-test (for linear regressions) and a z-test (for regressions such as logit or probit) on the null hypothesis that the “true” coefficient is equal to zero. You can also perform an F-test or chi-squared test on this hypothesis using the `test` command:

```plaintext
regress yield fert seed hybrid organic
```

To test whether seed hybrid and organic are jointly equal to zero, use the following syntax

```plaintext
test seed hybrid = 0;
```

The test results will appear in the Results window

You equally can perform any test on linear hypotheses about the coefficients, such as:

```plaintext
test fert = 0.5;  test coefficient on fert equals 0.5
```

```plaintext
test seed = 2.5;  test coefficients on seed equals 2.5
```
Regression Analyses

[You will be presented with a brief theoretical background to econometric/statistical modeling (See Appendices 7a, 7b and last part of Appendix 8b). You are however, encouraged to revise your Statistics/Econometrics. For those without statistical/econometric background, we will try to simplify it as we can. You are however, encouraged where possible, to take introductory statistical/econometric courses later on]

6.1 Estimation Procedure

Stata can do a lot of general and very specialized regression. The syntax for most of them is very similar and so we will focus on few commands in detail rather than discuss the whole list. Just so that you know the main ones, here is an abbreviated list of other regression commands that may be of interest:

anova  analysis of variance and covariance
cnreg  censored-normal regression
gmm   Generalized methods of moments estimator
heckman Heckman selection model
intreg interval regression
ivregress instrumental variables (2SLS) regression
newey  regression with Newey-West standard errors
prais  Prais-Winsten, Cochrane-Orcutt, or Hildreth-Lu regression
qreg   quantile (including median) regression
reg    ordinary least squares regression
reg3   three-stage least squares regression
rreg   robust regression (NOT robust standard errors)
sureg  seemingly unrelated regression
tobit  tobit regression
treatreg treatment effects model
truncreg truncated regression
**xtabond**  Arellano-Bond linear, dynamic panel-data estimator  
**xtintreg**  panel data interval regression models  
**xtreg**  fixed- and random-effects linear models  
**xtregar**  fixed- and random-effects linear models with an AR(1) disturbance  
**xttobit**  panel data tobit models

The dialog box all regressions can be found by clicking the statistics button on the manu bar.

The regress dialog box for instance can be accessed as follows:

Statistics > Linear models and related > linear regression.

The figure below shows the regress dialog box

![Regress Dialog Box](image)

As the dialog box indicates, the dependent and the independent variables, and the if/in qualifiers and weights can be specified. By default Stata will include a constant, but we can suppress that behavior with by ticking the appropriate box at the bottom of the dialog box. The standard errors will be calculated under the assumption of homoscedasticity of the error terms. The command allows for several alternatives, robust will calculate White-Huber robust standard errors, an ex-post correction that accounts for heteroscedasticity. We can add more flexibility with cluster, which accounts for correlation within clusters. Stata reports the 95% confidence interval, but we can change this by using the appropriate options.
As described above, most estimation commands will follow this type of syntax but the available options will differ and so you should check the relevant help files if you wish to use these approaches. Of course, Stata has a number of defaults and so you don’t need to include any options if you don’t wish to change the default (though it is always good to figure out what the default is!)

### 6.2 Post-estimation

Once you have run your regression, you usually want to carry out some extra analysis such as forecasting or hypothesis testing. Here is a list of the most useful post-estimation commands:

The following postestimation commands are of special interest after `regress`:

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>dfbeta</td>
<td>DFBETA influence statistics</td>
</tr>
<tr>
<td>estat hettest</td>
<td>tests for heteroskedasticity</td>
</tr>
<tr>
<td>estat intest</td>
<td>information matrix test</td>
</tr>
<tr>
<td>estat ovtest</td>
<td>Ramsey regression specification-error test for omitted variables</td>
</tr>
<tr>
<td>estat szroeter</td>
<td>Szroeter's rank test for heteroskedasticity</td>
</tr>
<tr>
<td>estat vif</td>
<td>variance inflation factors for the independent variables</td>
</tr>
<tr>
<td>estat esize</td>
<td>eta-squared and omega-squared effect sizes</td>
</tr>
</tbody>
</table>

The following standard postestimation commands are also available:

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>contrast</td>
<td>contrasts and ANOVA-style joint tests of estimates</td>
</tr>
<tr>
<td>estat ic</td>
<td>Akaike's and Schwarz's Bayesian information criteria (AIC and BIC)</td>
</tr>
<tr>
<td>estat sumvar</td>
<td>summary statistics for the estimation sample</td>
</tr>
<tr>
<td>estat von</td>
<td>variance-covariance matrix of the estimators (VCE)</td>
</tr>
<tr>
<td>estimatess</td>
<td>postestimation statistics for survey data</td>
</tr>
<tr>
<td>forecast</td>
<td>dynamic forecasts and simulations</td>
</tr>
<tr>
<td>hausman</td>
<td>Hausman's specification test</td>
</tr>
<tr>
<td>lincom</td>
<td>point estimates, standard errors, testing, and inference for linear combinations of coefficients</td>
</tr>
<tr>
<td>lincomtest</td>
<td>link test for model specification</td>
</tr>
<tr>
<td>(2) ltest</td>
<td>likelihood-ratio test</td>
</tr>
<tr>
<td>margins</td>
<td>marginal means, predictive margins, marginal effects, and average marginal effects</td>
</tr>
<tr>
<td>marginplot</td>
<td>graph the results from margins (profile plots, interaction plots, etc.)</td>
</tr>
<tr>
<td>niom</td>
<td>point estimates, standard errors, testing, and inference for nonlinear combinations of coefficients</td>
</tr>
<tr>
<td>predict</td>
<td>predictions, residuals, influence statistics, and other diagnostic measures</td>
</tr>
<tr>
<td>predictnl</td>
<td>point estimates, standard errors, testing, and inference for generalized predictions</td>
</tr>
<tr>
<td>pwcompare</td>
<td>pairwise comparisons of estimates</td>
</tr>
<tr>
<td>ssqstat</td>
<td>seemingly unrelated estimation</td>
</tr>
<tr>
<td>test</td>
<td>Wald tests of simple and composite linear hypotheses</td>
</tr>
<tr>
<td>testnl</td>
<td>Wald tests of nonlinear hypotheses</td>
</tr>
</tbody>
</table>

(1) forecast is not appropriate with nl or svy estimation results.
(2) ltest is not appropriate with svy estimation results.

### 6.3 Prediction

A number of predicted values can be obtained after all estimation commands, such as `reg`, `cnsreg`, `logit` or `probit`. The most important are the predicted values for the dependent variable and the predicted residuals. For example, suppose we run the basic regression again:
regress yield fert seed hybrid organic

predict y_yield : predicts the values for dependent variable
predict r : predicts the residuals of the model

Stata creates new variables containing the predicted values, and these variables can then be used in any other Stata command, e.g. you can graph a histogram of the residuals to check for normality.

[Also refer to Appendices 8a and 8b for some details on Probit and Logit models]

PRACTICAL SESSION 4: Regression Analysis – Linear Regression (OLS)
You will be provided with Dataset(s) to have a hands-on experience with Linear Regression Analysis. In this session, among other things, we will explore the pre-estimation checks, estimation, post-estimation and some predictions

PRACTICAL SESSION 5: Regression Analysis – Binary Logistic Regression
You will be provided with Dataset(s) to have a hands-on experience with Binary Logistic Regression Analysis. In this session, among other things, we will explore the pre-estimation checks, estimation, post-estimation and some predictions

7 Final Remarks

We hope you found this Manual useful in your introduction to Stata. This is the first in a series of trainings that the Centre for Agricultural Research and Development (CARD) at LUANAR would be conducting. The next in this series is Intermediate Stata Analysis to be followed by Advanced Stata Analysis. In subsequent courses, depending on need, we will be combining the analyses with their application in policy analysis.

Once again thank you for attending the course and/or using the Manual. You are welcome to make any suggestions and comments.
Appendices
Appendix 1: Introduction to Policy Analysis
Introduction to Policy Analysis

By Charles Jumbe, PhD
What is a policy?

• A policy is a course of action or inaction chosen by public authorities to address a given problem or interrelated set of problems.
Dimensions of Policy

- Legal
- Economic
- Social
- Political
Dimensions of Policy

• **Economic**
  – Increased competition to lower transport cost and improve quality of service (foreign vs. local)
  – Devaluation of the currency to stimulate exports

• **Political**
  – Lifting the ban on maize exports
  – Setting the minimum price of tobacco

• **Legal**
  – Bill proposing the minimum age for a girl to get married
  – Amendment of the Constitution to restrict the incumbent VP from taking the high office of the President.

• **Social**
  – Implementation of the social cash transfer programme
What Is Policy Analysis?

• Several aspects
  – Gathering info on problems (causes & effects)
  – Identifying several ways of dealing with the problem (alternatives)
  – Assessing the likely results of those alternatives

• Involves both empirical and normative issues
• Involves both science and politics
• Utilizes many disciplines (e.g. economics, statistics, management science)
What is Policy Analysis

• Use of reason and evidence to choose course of action to attain a given set of goals/objectives;

• A body of concepts and principles aimed at helping the decision maker make choices intelligently, ethically and effectively.

• Analyzes existing practices/policies for effectiveness.
Why Do Policy Analysis?

• Need for thoughtful, impartial assessment of problems and seek solutions
• Avoid “shooting from the hip” without knowing the underlying causes
• Anticipate potential outcomes and helps to plan for mitigation actions
• Assessing impact (ex-post evaluation)
Goals of Policy Analysis

• Evaluating alternative policy prescriptions/redesigning policies/programs
• Reducing uncertainty & providing information for decision makers in the public arena
• As systematic evaluation of the technical and economic feasibility and political viability of alternative policies, strategies for implementation, and consequences for policy adoption
1. What should be our goals?
2. Which option or option mix promises fewest negatives and greatest benefits?
3. What is the problem?

1. Is the policy politically viable?
2. What variables are available to help ensure the successful implementation of the policy?

1. By what criteria can policy be judged fair? Judged good?
Approaches in Policy Analysis

• **Basic Analysis**
  – a subset of policy analysis methods comprising quickly applied but theoretically sound ways to aid in making policy decisions

• **Policy Research**
  – a subset of policy analysis methods requiring substantial budget, time and data to achieve results
Types of Analysis

• **Scientific**
  - Search for truth and build theory about policy actions and effects
  - May be too theoretical for most decision makers
  - May take too much time for most decision makers
Types of Analysis (con’t)

• Professional
  – Analyze alternatives to solve problems
  – Goal is for practical value
  – Research can be too narrow due to time or resource constraints
  – Examples: Most academic research (e.g., Impact of decentralization on public health service delivery.)
• Political
  – Advocate and support preferred policies
  – Often ideological or partisan; may lack analytical depth
  – Examples: *Whether or not to continue hosting AU Summit*
Policy Analysis Framework

Policy Analysis

Theoretical Basis
- Economic
- Political

Process
- Formulation
- Implementation
- Evaluation

Tools
- Quantitative
- Qualitative
Steps in Policy Analysis

• Define and analyze the problem
  – What is it
  – Who is affected and how seriously
  – What are the causes & how did it develop

• Construct policy alternatives
  – What are the options for dealing with it
  – Might be the most important step
  – Want to encourage creativity
Steps in Policy Analysis (con’t)

• Develop evaluative criteria
  – What is most suitable: effectiveness, efficiency, equity, political & social feasibility
  – Will vary depending on the problem

• Assess policy alternatives
  – Which is likely to produce desired outcomes
  – Evidence

• Draw conclusions
  – Which alternative is most desirable
  – Single policy action, or combination
What Analysis Is Needed?

• **Root causes vs. pragmatic adjustments**
  – Should focus be on the underlying issues or on addressing the issue at hand

• **Comprehensive vs. short-term relevance**
  – Comprehensive is more thorough; better methodologically, but also takes more time
  – Short-term may be less rigorous and raise quality concerns, but may also be more timely
What Analysis Is Needed? (con’t)

• Consensual vs. contentious
  – Should analysts do studies which closely adhere to mainstream values (norms), or
  – Should they do studies which challenge those values

• Rational, technical analysis vs. democratic politics
  – Rational, technical analysis tends to focus efficiency, & to rely on highly trained analysts
  – What about citizen involvement also be considered when making decisions in a democratic state

• Ethical issues
  – How studies are conducted
  – Bias or funding source of the analyst
Policy Analysis Process

• Approach to problems that is logical, structured, valid, and replicable
• Generation of feasible courses of action
• A search for information and evidence of benefits and other consequences of courses of action
• In order to help policy makers choose the most advantageous policy action
Policy Analysis Process

- **Ex-Ante**
  - (Rational model)

- **Ex-Post**
  - (Program evaluation)
Ex-ante evaluation

– Projecting future states with and without policy or program
Evaluating Alternative Policies or Programs

**Ex-post evaluation**

- Did policy or program achieve its objective?
- Was policy or program efficient, equitable, and politically acceptable?
- How best should it have been designed/implemented to achieve maximum impact?
Appendix 2a: Introducing Stata
Introducing Stata

Thabbie Chilongo

10 November 2014
What is Stata

• Stata is a powerful statistical package with:
  – Smart data-management facilities
  – A wide range of up-to-date statistical techniques
  – A graph-producing capability (arguably not the best at this)
  – Fast and relatively easy to use

• The word Stata:
  – Not an abbreviation but corruption of the word Statistics.
Why Use Stata?

• Numerous comparable packages on the market, e.g. SPSS, R, SAS, Matlab, Eviews, etc.

• So why Stata and not them?
  – Capability to handle and manipulate large datasets (up to slightly over 2 billions of observations!)
  – Constantly being updated or advanced by users
  – In short as one user summed it all:
    • “When working with Stata it’s like you are talking to a very obedient person who does exactly what you want them to do.”
Types of Stata

• Four different types (sizes) available for each version of Stata:
  – Stata/MP (Multi-processor)
  – Stata/SE (Special Edition)
  – Stata/IC (Intercooled)
  – Small Stata
### Characteristics of Stata Types

<table>
<thead>
<tr>
<th>Stata Type</th>
<th>Maximum Number of Variables</th>
<th>Maximum Number of Regressors</th>
<th>Maximum Number of Observations</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stata/MP</td>
<td>32,767</td>
<td>10,998</td>
<td>2,147,583,647*</td>
<td>• Runs on multiple CPUs or cores (but also on single core). • Fastest.</td>
</tr>
<tr>
<td>Stata/SE</td>
<td>32,767</td>
<td>10,998</td>
<td>2,147,583,647*</td>
<td>• Runs on single core • Can run on multiple core computers but uses only single core.</td>
</tr>
<tr>
<td>Stata/IC</td>
<td>2,047</td>
<td>798</td>
<td>2,147,583,647*</td>
<td></td>
</tr>
<tr>
<td>Small Stata</td>
<td>99</td>
<td>99</td>
<td>1,200</td>
<td></td>
</tr>
</tbody>
</table>

*Assuming you have enough memory

• Stata/SE Version 13 to be used in this course
What Can Stata Do?

• Stata is command-driven package
• Also has pull-down menus as will be shown, but:
  – Best way to learn Stata is through typing the commands
  – For those interested in programming, typing makes it easy to switch into programming.
  – Arguably, you get the best out of Stata through typing the commands.
Appendix 2b: Introduction to Stata
Introduction to STATA
- Basics of STATA

By
Charles Jumbe, PhD
What is this Course About?
The Course

• This course is designed to suit the needs for those who wish to acquire basic skills to analyze statistical data sets (for policy analysis) and produce technical reports.

• Learning Outcomes

• After completion of this course the delegate should be able to perform data entry, manipulation and analysis in STATA as outlined in the program.

• Stata is one of the big three general purpose statistical programs.

• Version 1 Born in 1985, current running as ver 13.
What is STATA

• Stata is one of the big three general purpose statistical programs. Version 1 Born in 1985, current running as version 10.

• **Basic Principles**
  – Command language
  – Reproducible
  – Platform independent
  – Backward compatible

• It is an intuitive data management and manipulation, wide range of Statistics

• Fully programmable publication quality graphics

• Inexpensive (c.f. SPSS, SAS) and widely used
What is Stata?

• Stata is one of the big three general purpose statistical programs. Version 1 Born in 1985, current running as version 10.

• **Basic Principles**
  – Command language
  – Reproducible
  – Platform independent
  – Backward compatible

• It is an intuitive data management and manipulation, wide range of Statistics

• Fully programmable publication quality graphics

• Inexpensive (c.f. SPSS, SAS) and widely used
Introduction

Why use Stata?

• Stata provides easy access to important analyses that are not available in many standard statistical packages.
  1. Panel / cross-sectional time series data
  2. Complex survey data
  3. Limited dependent variables.

• Stata includes commands for tobit regression (tobit), heckman selection models (heckman, heckprob)

• Specialized statistical capabilities at no additional cost (e.g. xt: Time series, svy: Survey, st: Survival, Robust S.E’s e.t.c.)

• Handles complicated data collected from complicated designs
Strength of STATA

• Stata is an excellent tool for data manipulation:
  – moving data from external sources into the program, cleaning it up, generating new variables, generating summary data sets, merging data sets and checking for merge errors, reshaping data sets from “long” to “wide”...

• Stata provides all of the standard univariate, bivariate and multivariate statistical tools, from descriptive statistics and t-tests through one-, two- and N-way ANOVA, regression, principal components, and the like.

• Stata provides all of the standard univariate, bivariate and multivariate statistical tools, from descriptive statistics and t-tests through one-, two- and N-way ANOVA, regression, principal components, and the like.
Strength of STATA

• Stata has a very powerful set of techniques for the analysis of limited dependent variables:
  – logit, probit, ordered logit and probit, multinomial logit, etc
  – Stata’s regression capabilities are full-featured,
    • regression diagnostics, prediction, robust estimation of standard errors
    • instrumental variables / two-stage least squares,
    • seemingly unrelated regressions / three-stage least squares, etc.
• Stata graphics have been extensively improved and enhanced.
• Graphics are excellent tools for exploratory data analysis,
• Produce high–quality 2-D publication-quality graphics, in a variety of “schemes”.
Strength of STATA

• Stata is very well supported by telephone and email technical support, as well as the more informal support provided by other users on StataList, the listserv.
• The manuals are useful–particularly the User’s Guide
• Full details of the command syntax are available online in the windowed “Help Viewer” in hypertext form.
• Even in the command–line environment, the full help files are available.
• STATA runs interactively with short and simple commands, making it relatively easy to learn.
• STATA allocates a default amount of memory into which it loads a copy of the input data set- very fast to produce results
Basics of STATA

- Stata is invoked by double-clicking on the STATA icon.
- You enter your commands in the Command Window.
Basics of STATA

• There are four **windows**.

• **Command Window**- You enter your commands in this window.

• **Review Window** records your commands.

• **Results window** displays your output.
  – The **Results window** is the Log Window. This can be named, save, and reused later

• **Variables window** lists the variables in the data set you are using.

• The **Display Colours** may be adjusted by going into the **Edit window**, clicking on **general preferences**, and selecting the **background colours** for the different windows.
Basics of STATA

• In the **header bar** at the top of the screen is a list of topics:
  
  • Version 10: **File, Edit, Data, Graphics, Statistics, User, Window,** and **Help**.
  
  • The **Help option** in the Header bar provides a **Contents** option and a **Search** Option.
  
  • The ‘Contents’ option can be used by beginners unfamiliar with STATA commands.
  
  • The ‘Search’ option can be used by users who know the name of the command or topic they wish to search.
  
  • The STATA web site, at http://www.stata.com,
    
    – You can find many useful links, resources,
    
    – Available publications on how to use STATA.
    
    – Archive of solutions problems- stata serverlist
General Syntax

[prefix]: command [varlist] = [exp] [if] [in] [weight] [using filename], [options]

- [Prefix]: Some commands precede a Stata command and modify its behaviour
  - (e.g., regression by EPA—by EPA, reg dep var indep varlist)
- command: Tell STATA what it should do for you
  - Describe in short ‘des’
  - Tabulate in short ‘tab’
  - Regress in short ‘reg’
  - Graph in short ‘gr’
  - Summarize in short ‘sum’
- [Varlist]: List of one or many variables
  - sum age landsize income sex
[prefix]: command [varlist] = [exp] [if] [in] [weight] [using filename] , [options]

• [= exp]: Used in commands where algebraic expressions produce a new or update variable - **DATA MANIPULATION**

• [if] & [in]: Conditions and ranges - *choose data to do analysis on*

• [using]: Analysis to be done on data saved with a file name

• [, options]: These are specified depending on the type of analysis
  – e.g., estimation of robust standard errors instead on the normal errors

• **GOOD NEWS:** All Stata commands have help files.

• The *help command* shows you how to use the command as well as the *options* associated with a particular *command*
  – *help mean; help heckman; help graph*

• Use the *search command* for any type of analysis or procedure
  – *Search regression; Search tobit; search switching regression*
Appendix 2c: Stata Interface
Introduction to Stata Interface and Syntaxes

Francis Darko
The Stata Interface

1. Windows
2. Toolbar
3. Pull down manus
The Stata Interface

Parts of the STATA Interface

1. Menu bar (pull down menu)
2. Toolbar
3. Five windows

- Results window
- Variables window
- Properties window
- Commands window
- Review Window
Stata Interface: Windows

<table>
<thead>
<tr>
<th>Window</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>Variables</td>
<td>The Variables window shows the list of variables in the dataset, along with selected properties of the variables.</td>
</tr>
<tr>
<td>Properties</td>
<td>The Properties window displays variable and dataset properties</td>
</tr>
<tr>
<td>Command</td>
<td>Commands are submitted to Stata from the Command window</td>
</tr>
<tr>
<td>Results</td>
<td>The Results window contains all the commands and their textual results you have entered during the Stata session</td>
</tr>
<tr>
<td>Review</td>
<td>The Review window shows the history of commands that have been entered</td>
</tr>
</tbody>
</table>
Stata Interface: Toolbar

**Open:** opens a Stata dataset. Click on the button to open a dataset with the Open dialog.

**Save:** saves the Stata dataset currently in memory to disk.

**Print:** displays a list of windows. Select a window name to print its contents.

**Log:** begins a new log or closes, suspends, or resumes the current log.

**Viewer:** opens the Viewer or brings a Viewer to the front of all other windows. Click on the button to open a new Viewer. Click on the arrow to select a Viewer to bring to the front.

**Graph:** brings a Graph window to the front of all other windows. Click on the button to bring the Graph window to the front. Click on the arrow to select a Graph window to bring to the front.

**Do file Editor:** opens the Do-file Editor or brings a Do-file Editor to the front of all other windows. Click on the button to open a new Do-file Editor. Click on the arrow to select a Do-file Editor to bring to the front.

**Data Editor (Edit):** opens the Data Editor or brings the Data Editor to the front of the other Stata windows.

**Data Editor (Browse):** opens the Data Editor in browse mode.

**Variables Manager:** opens the Variables Manager.

**Clear more Condition:** tells Stata to continue when it has paused in the middle of long output.

**Break:** stops the current task in Stata.
Stata Interface: Manus

Stata Pull down manus

- Provides another way of telling STATA to do what you want
- Click on each on of them for the pull down manus
The Stata Syntax – The Language of STATA

- The general syntax:

```
[prefix :] command [varlist] [=exp] [if] [in] [weight] [using filename] [, options]
```

<table>
<thead>
<tr>
<th>language element</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>prefix :</td>
<td>prefix command</td>
</tr>
<tr>
<td>command</td>
<td>Stata command</td>
</tr>
<tr>
<td>varlist</td>
<td>variable list</td>
</tr>
<tr>
<td>=exp</td>
<td>expression</td>
</tr>
<tr>
<td>if</td>
<td>if exp qualifier</td>
</tr>
<tr>
<td>in</td>
<td>in range qualifier</td>
</tr>
<tr>
<td>weight</td>
<td>weight</td>
</tr>
<tr>
<td>using filename</td>
<td>using filename modifier</td>
</tr>
<tr>
<td>options</td>
<td>options</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>prefix</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>by</td>
<td>run command on subsets of data</td>
</tr>
<tr>
<td>capture</td>
<td>run command and capture its return code</td>
</tr>
<tr>
<td>noisily</td>
<td>run command and show the output</td>
</tr>
<tr>
<td>quietly</td>
<td>run command and suppress the output</td>
</tr>
<tr>
<td>version</td>
<td>run command under specified version</td>
</tr>
</tbody>
</table>
Stata Syntaxes: Example

- Example 1: descriptive statistics (mean, min, max etc) of yield fert seed for the three region districts of Malawi

Syntax:  
```
by region:
  summarize yield fert seed if year==2010 ,detail
```
Appendix 3: Data Entry, Import and Export
Getting Data Into Stata:
Data Entry, Importing and Exporting

Francis Darko
Getting Data into STATA

- **STATA Data:**
  1. Double click the Stata data file to load it into Stata
  2. From the Stata interface, click on “file” and then on “open”.
  3. You can also use the “open” icon on the manu bar to do the same

- **Other formats (Excel, SAS etc)**
  Click on “file” on the manu bar. In the file drop down manu, click on “Export” and then choose the format of your dataset. The import dialog box will appear. Choose the options in the dialog box as appropriate and click on “OK”.

Getting Data into STATA

- Using the command window:
  
  a. Stata file: `use` command
     
     Syntax: `use "location of file\filename", clear`
     
     Example: `use "C:\Users\user\Desktop\responserate.dta", clear`

  b. Excel file: `import` command
     
     Syntax: `import excel "file location\filename.xls", sheet("Sheet1") firstrow`
     
     Example: `import excel "C:\Users\user\Desktop\responserate.xls", sheet("Sheet1") firstrow`

  c. CSV file: `insheet` command
     
     Syntax: `insheet using "file location\filename.csv"`
     
     Example: `insheet using "C:\Users\user\Desktop\responserate.csv"`

  d. SPSS file: `usespss` command
     
     `ssc install usespss` (installs usespss)
     
     `usespss using location of file\filename.sav`
     
     Example: `usespss using C:\Users\user\Desktop\responserate.sav`
Getting Data into STATA

- Manual data entry

Steps:

1. Open STATA using the start button or the icon on the desktop
2. Click on the “Data Editor” on the toolbar
3. Enter data
   - variable names appear as var1, var2, var3, …,
   - String data appear in red
4. Rename variables

Demonstrate
Saving Data

• Saving in STATA format for the first time:
  1. select **File > Save As**... and use the Save As dialog box to complete the saving. OR
  2. type *save filename* in the Command window.

• Saving a modified data in STATA format
  1. select **File > Save**; OR
  2. click on the *Save* button; OR
  3. type *save, replace* in the Command window.

Saving in other formats:
Select **File > Export**, then select the format and use the dialog box to complete saving.
Appendix 4: Data Documentation (Log File)
Stata Documentation: Creating a Log File

Francis Darko
What is a log file?

• A log file is simply a record of your Results window. It records all commands and all textual output as it happens.

• All the output that appears in the Results window can be captured in a log file.

• It is recommend that you start a log file whenever you begin any serious work in Stata.

• Stata can save the file in one of two different formats.
  1. Stata Markup and Control Language (SMCL) format (Default)
  2. Log format

• SMCL format is recommended because SMCL files can be translated into a variety of formats readable by applications other than Stata.
Creating a log file

1. File > Log > Begin
2. Issue commands to get output
3. Suspend log ??? : File > Log > Suspend
4. Resume log ??? : File > Log > Resume
5. Close: File > Log > Close

• If you choose an existing log file, you will be asked whether you want to view, append or over-write the existing log file.
Viewing and translating Log files

• You can view the log file using the Viewer window in two ways:
  Viewer > file > open...>Browse
  File > log> view...

You can convert from SMCL format to other formats:
  File > Log > Translate... menu
Appendix 5: Merging and Appending Datasets
Merging and Appending Datasets in Stata

Francis Darko
Merging and Appending Datasets

Merging

The *merge* command joins corresponding observations from the dataset currently in memory (called the *master dataset*) with those from a second dataset (called the *using dataset*).
Merging and Appending Datasets

Appending
Adding a Stata dataset to the end of the dataset in memory

<table>
<thead>
<tr>
<th>Year = 2010</th>
<th>Household</th>
<th>HH_Size</th>
<th>Income</th>
<th>Expenditure</th>
<th>Year</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>5</td>
<td>50000</td>
<td>40000</td>
<td></td>
<td>2010</td>
</tr>
<tr>
<td>B</td>
<td>4</td>
<td>80000</td>
<td>65000</td>
<td></td>
<td>2010</td>
</tr>
<tr>
<td>C</td>
<td>7</td>
<td>40000</td>
<td>35000</td>
<td></td>
<td>2010</td>
</tr>
<tr>
<td>D</td>
<td>3</td>
<td>20000</td>
<td>25000</td>
<td></td>
<td>2010</td>
</tr>
<tr>
<td>E</td>
<td>6</td>
<td>37000</td>
<td>30000</td>
<td></td>
<td>2010</td>
</tr>
<tr>
<td>F</td>
<td>8</td>
<td>45000</td>
<td>40000</td>
<td></td>
<td>2010</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Year = 2013</th>
<th>Household</th>
<th>HH_Size</th>
<th>Income</th>
<th>Expenditure</th>
<th>Year</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>7</td>
<td>55000</td>
<td>41000</td>
<td></td>
<td>2013</td>
</tr>
<tr>
<td>B</td>
<td>5</td>
<td>85000</td>
<td>67000</td>
<td></td>
<td>2013</td>
</tr>
<tr>
<td>C</td>
<td>7</td>
<td>45000</td>
<td>36000</td>
<td></td>
<td>2013</td>
</tr>
<tr>
<td>D</td>
<td>6</td>
<td>25000</td>
<td>26000</td>
<td></td>
<td>2013</td>
</tr>
<tr>
<td>E</td>
<td>8</td>
<td>39000</td>
<td>31000</td>
<td></td>
<td>2013</td>
</tr>
<tr>
<td>F</td>
<td>7</td>
<td>49000</td>
<td>41000</td>
<td></td>
<td>2013</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Household</th>
<th>HH_Size</th>
<th>Income</th>
<th>Expenditure</th>
<th>Year</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>5</td>
<td>50000</td>
<td>40000</td>
<td>2010</td>
</tr>
<tr>
<td>B</td>
<td>4</td>
<td>80000</td>
<td>65000</td>
<td>2010</td>
</tr>
<tr>
<td>C</td>
<td>7</td>
<td>40000</td>
<td>35000</td>
<td>2010</td>
</tr>
<tr>
<td>D</td>
<td>3</td>
<td>20000</td>
<td>25000</td>
<td>2010</td>
</tr>
<tr>
<td>E</td>
<td>6</td>
<td>37000</td>
<td>30000</td>
<td>2010</td>
</tr>
<tr>
<td>F</td>
<td>8</td>
<td>45000</td>
<td>40000</td>
<td>2010</td>
</tr>
<tr>
<td>A</td>
<td>7</td>
<td>55000</td>
<td>41000</td>
<td>2013</td>
</tr>
<tr>
<td>B</td>
<td>5</td>
<td>85000</td>
<td>67000</td>
<td>2013</td>
</tr>
<tr>
<td>C</td>
<td>7</td>
<td>45000</td>
<td>36000</td>
<td>2013</td>
</tr>
<tr>
<td>D</td>
<td>6</td>
<td>25000</td>
<td>26000</td>
<td>2013</td>
</tr>
<tr>
<td>E</td>
<td>8</td>
<td>39000</td>
<td>31000</td>
<td>2013</td>
</tr>
<tr>
<td>F</td>
<td>7</td>
<td>49000</td>
<td>41000</td>
<td>2013</td>
</tr>
</tbody>
</table>
Merging and Appending Datasets

Types of merging:

1. One-to-one: Unique observations in both master and using data

<table>
<thead>
<tr>
<th>Plot</th>
<th>Yield</th>
<th>Plot</th>
<th>Fertilizer</th>
<th>Seed</th>
<th>Hybrid</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1000</td>
<td>1</td>
<td>182</td>
<td>32</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>1200</td>
<td>2</td>
<td>150</td>
<td>30</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>1100</td>
<td>3</td>
<td>120</td>
<td>17</td>
<td>1</td>
</tr>
<tr>
<td>4</td>
<td>900</td>
<td>4</td>
<td>183</td>
<td>20</td>
<td>0</td>
</tr>
<tr>
<td>5</td>
<td>1300</td>
<td>5</td>
<td>110</td>
<td>27</td>
<td>0</td>
</tr>
<tr>
<td>6</td>
<td>1450</td>
<td>6</td>
<td>150</td>
<td>33</td>
<td>1</td>
</tr>
<tr>
<td>7</td>
<td>970</td>
<td>7</td>
<td>150</td>
<td>29</td>
<td>0</td>
</tr>
</tbody>
</table>
Merging and Appending Datasets

Types of merging:

1. One-to-many: Unique observations in the master dataset, but duplicates in the using dataset

<table>
<thead>
<tr>
<th>Household</th>
<th>HH_Size</th>
<th>Income</th>
<th>Expenditure</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>5</td>
<td>50000</td>
<td>30000</td>
</tr>
<tr>
<td>B</td>
<td>4</td>
<td>65000</td>
<td>50000</td>
</tr>
<tr>
<td>C</td>
<td>7</td>
<td>25000</td>
<td>24000</td>
</tr>
<tr>
<td>D</td>
<td>3</td>
<td>40000</td>
<td>35000</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Plot</th>
<th>Yield</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1000</td>
</tr>
<tr>
<td>2</td>
<td>1200</td>
</tr>
<tr>
<td>3</td>
<td>1100</td>
</tr>
<tr>
<td>4</td>
<td>900</td>
</tr>
<tr>
<td>5</td>
<td>1300</td>
</tr>
<tr>
<td>6</td>
<td>1450</td>
</tr>
<tr>
<td>7</td>
<td>970</td>
</tr>
</tbody>
</table>
Merging and Appending Datasets

- Types of merging:
  1. Many-to-one: Duplicate observations in the master dataset, but unique in the using data.

<table>
<thead>
<tr>
<th>Plot</th>
<th>Yield</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1000</td>
</tr>
<tr>
<td>2</td>
<td>1200</td>
</tr>
<tr>
<td>3</td>
<td>1100</td>
</tr>
<tr>
<td>4</td>
<td>900</td>
</tr>
<tr>
<td>5</td>
<td>1300</td>
</tr>
<tr>
<td>6</td>
<td>1450</td>
</tr>
<tr>
<td>7</td>
<td>970</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Household</th>
<th>HH_Size</th>
<th>Income</th>
<th>Expenditure</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>5</td>
<td>50000</td>
<td>30000</td>
</tr>
<tr>
<td>B</td>
<td>4</td>
<td>65000</td>
<td>50000</td>
</tr>
<tr>
<td>C</td>
<td>7</td>
<td>25000</td>
<td>24000</td>
</tr>
<tr>
<td>D</td>
<td>3</td>
<td>40000</td>
<td>35000</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Plot</th>
<th>Yield</th>
<th>HH_Size</th>
<th>Income</th>
<th>Expenditure</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1000</td>
<td>5</td>
<td>50000</td>
<td>30000</td>
</tr>
<tr>
<td>2</td>
<td>1200</td>
<td>5</td>
<td>50000</td>
<td>30000</td>
</tr>
<tr>
<td>3</td>
<td>1100</td>
<td>4</td>
<td>65000</td>
<td>50000</td>
</tr>
<tr>
<td>4</td>
<td>900</td>
<td>7</td>
<td>25000</td>
<td>24000</td>
</tr>
<tr>
<td>5</td>
<td>1300</td>
<td>7</td>
<td>25000</td>
<td>24000</td>
</tr>
<tr>
<td>6</td>
<td>1450</td>
<td>3</td>
<td>40000</td>
<td>35000</td>
</tr>
<tr>
<td>7</td>
<td>970</td>
<td>3</td>
<td>40000</td>
<td>35000</td>
</tr>
</tbody>
</table>
Merging and Appending Datasets

Merging syntaxes

- One to one
  \( \text{merge 1:1 varlist using "location and name of second file"}, \text{keepusing}(\text{vars}) \)

- One to many
  \( \text{merge 1:m varlist using "location and name of second file"}, \text{keepusing}(\text{vars}) \)

- Many to one
  \( \text{merge m:1 varlist using "location and name of second file"}, \text{keepusing}(\text{vars}) \)

*Using drop down manu:* Data > Combine datasets > merge two datasets
Merging and Appending Datasets

- **Append syntax**

  - Syntax: *append using* “location and name of second file”

- **Using the pull down menu:** Data > Combine datasets > Append datasets
Exercise 1: Exporting and Importing data

1. Download the auto.dta and auto2.dta datasets
   File > Example Datasets... > Example datasets installed with Stata > click once on the use by auto.dta > save the file as auto.dta

2. Download auto2.dta using the same procedure and in 1

3. Load auto.dta into Stata (using a do file)
   Command: `use "C:\Users\user\Desktop\auto.dta", clear`

4. Export auto.dta to excel and the save the excel file as auto.xls

5. Import auto.xls into stata
   command: `use ": import excel "C:\Users\user\Desktop\auto.xls", sheet("Sheet1") firstrow"`
Exercise: Appending and Merging

• 1. Append the auto and auto2 datasets

Command: `append using "location\auto2.dta"`

• 2. Merge HouseholdGeovariables_IHPS.dta and Household\HH_MOD_A_FILT using 1:1 merging

Command: `merge 1:1 y2_hhid using "location\HouseholdGeovariables_IHPS.dta"`

• 3. merge HouseholdGeovariables_IHPS.dta and PlotGeovariables_IHPS.dta using 1:m merging

Command: `merge 1:m y2_hhid using "location\PlotGeovariables_IHPS.dta"`

• 4. PlotGeovariables_IHPS.dta and HouseholdGeovariables_IHPS.dta using m:1 dataset

Command: `merge m:1 y2_hhid using "location\HouseholdGeovariables_IHPS.dta"`
Appendix 6: Introduction to Data Manipulation
Introduction to STATA
Documentation & Data Manipulation

By
Charles Jumbe, PhD
Prefix:
`bysort foreign: summarize price;` //summarizes price for domestic and foreign vehicles

Command:
`sum price displacement gear_ratio length;` //Gives summary statistics for the variables

*Use of `if`, `equality` and `&`:
`sum price if foreign==0 & price>=1000;` //summarise price for domestic cars that are US$1000
version 13

clear

# delimit; // From now on, each line will end with an semi-colon except for notes

set more off, permanently; //set more off

use "D:\Director of Research and Outreach\Seminars znd Workshops\Stata\auto.dta"; /* open auto data saved on the computer*/

log using "D:\Director of Research and Outreach\Seminars znd Workshops\Stata\day5.log", replace; //File that saves the results

log close;
describe; ///describe data in memory

codebook; //gives more details about the data

codebook price foreign; //details of price and foreign

list; /*list all variables*/

list mpg length rep78; //list variables

inspect; /* Gives different statistics */

summarize; //gives summary statistics for all variables

sum mpg length trunk rep78, detail; //provides summary statistics with details

browse; //gives a spreadsheet of the database

list make if rep78 == 5 | mpg > 25; // list the make of the vehicle with repair record of 5 or fuel consumption >25
**GENERATING TABLES (TABULATION)**

```plaintext
tab foreign rep78; /* This gives a table of statistics*/
tab foreign, summarize (price); //Table of price against source of vehicle/
tab foreign rep78, chi2 row col; //Cross tabulation showing X-square test, row and column percentages

**tab1** foreign price mpg weight; //Frequency tables for variables listed

**tab2** mpg foreign; // Frequency table of fuel consumption against source

**tabstat** price mpg weight, statistics(mean min range max skewness); // stats for variables

**tabstat** price weight mpg rep78, by(foreign); // separate stats for variables for imported and locally manufactured vehicles.

**tabstat** price in 20/70; //get the mean for price for obs 20 to 70)

**tabstat** price weight mpg rep78, by(foreign) stat(mean sd min max) nototal long col(stat); //Show statistics horizontally and variables vertically
```
scatter mpg price; // Gives a scatter plot of variables

graph matrix mpg price weight length; /* identifying outliers in the data set */

hist price if foreign==0; // draw histogram of price for domestic vehicles

hist price, by(foreign); // draw histogram of price for source of vehicles

graph matrix price weight mpg rep78; // do scatter plots for the chosen variables

graph matrix price weight mpg rep78, half; // do scatter plots for the chosen variables showing half of the matrix

graph box price, over(foreign); /* do a box graph of price over vehicle source */

twoway scatter price weight, title(Plot of Price vs Weight); // scatter plot of price & weight with a title//
### Data Manipulation in STATA

<table>
<thead>
<tr>
<th>Arithmetic</th>
<th>Logical</th>
<th>Relational (numeric and string)</th>
</tr>
</thead>
<tbody>
<tr>
<td>+   addition</td>
<td>&amp; and</td>
<td>&gt;   greater than</td>
</tr>
<tr>
<td>-   subtraction</td>
<td></td>
<td>or</td>
</tr>
<tr>
<td>*   multiplication</td>
<td>! not</td>
<td>&gt;=  greater than or equal</td>
</tr>
<tr>
<td>/   division</td>
<td>~ not</td>
<td>&lt;=  less than or equal</td>
</tr>
<tr>
<td>^   power</td>
<td></td>
<td>==  equal</td>
</tr>
<tr>
<td>-   negation</td>
<td></td>
<td>!=  not equal</td>
</tr>
<tr>
<td>+   string concatenation</td>
<td>~= not equal</td>
<td></td>
</tr>
</tbody>
</table>

A double equal sign (==) is used for equality testing.

The order of evaluation (from first to last) of all operators is ! (or ~), ^, - (negation), /, *, - (subtraction), +, != (or \=), >, <, <=, >=, ==, & and |.
Data Manipulation in STATA

* RENAMING VARIABLES;
  • ren mpg kpl ; /*kilometer per litre */

*GENERATE NEW VARIABLES;

  gen pricltre=price/kpl; /*generate the variable prive over kpl*/
  sum pricltre; //Gives summary statistics of generated variable
  generate USprice = price/520; //Creates the US equivalent of price
  generate ln_kpl=log(kpl); //generates log of mpg
  list kpl ln_kpl; //list kpl and generated log of kpl

*RECODING VARIABLES;

  Recoding variable: A variable X with continuous values may be collapsed into a dummy variable with 2 categories. One category can be 2 or below. The other category can be above 5 with the following command:

  recode rep78 0/2=1 3/5=2; //recoding repair record into 2 categories
  list rep78; /* display the recoded variables*/
Data Manipulation in STATA

* GENERATE NEW VARIABLES;

```stata
gen pricesq=price^2; // generate a square of price
list price pricesq; ///Displays price and the square of price
sum pricltre; //Gives summary statistics of generated variable
```

*CORRELATION ANALYSIS;

```stata
pwcorr foreign price pricesq kpl length trunk rep78; /*Gives correlation matrix */
pwcorr foreign price pricesq kpl length trunk rep78, sig ; //produce p-values
pwcorr foreign price pricesq kpl length trunk rep78, star(.01); //add star at 1% level
pwcorr foreign price pricesq kpl length trunk rep78, star(.05) bon; //Pairwise correlation with sign at 5% with star//
```
Appendix 7a: Introduction to Quantitative Economic Modelling
INTRODUCTION TO QUANTITATIVE ECONOMETRIC/STATISTICAL MODELLING

Thabbie Chilongo

Data Management and Analyses Using Stata

10-14 November 2014, MIM
Linear Regression

• A statistical tool for evaluating the relationship of one or more independent variables (x) to a single continuous dependent variable (y).

• Regression analysis characterizes the relationship between the dependent and independent variables by looking determining the extent, direction and strength of the association.
Causal relationships and ceteris paribus analysis

Goal is to find if change in one variable x causes change in another variable y.

Ceteris paribus- holding all other relevant factors fixed.

Correlation is rarely enough to conclude.

Therefore need to hold other factors constant.
The Population Model

- The population model is linear in parameters

\[ y = \beta_0 + \beta_1 x_1 \ldots + \beta_k x_k + u \]

- \( u \) is an unobservable random disturbance (error)
Key assumptions of OLS

• The dependent variable (y) is linear and continuous
• The error term has mean of zero, i.e. The differences between the estimated dependent variable (yhat) and the observed (actual) dependent variable (y) sums to zero ($\sum(yhat-y) = 0$)
• The independent variables are not correlated with each other (no multicollinearity).
• The independent variable are exogenous, i.e. they are not themselves dependent variables.

• NB: OLS is always the starting point of modelling. All other models are as a result of violating some of the OLS assumptions, e.g. a probit or logit model is applicable where the dependent variable is not continuous but binary categorical.
## Deviations from OLS: Model Diagnosis and Prescription

### Y dependent variable

<table>
<thead>
<tr>
<th>Linear</th>
<th>• OLS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Categorical</td>
<td>• Probit, Logit</td>
</tr>
<tr>
<td>Ordered more than 2</td>
<td>• Ordered probit, Ordered logit</td>
</tr>
<tr>
<td>Discrete choices more than 2</td>
<td>• Multinomial logit, Multinomial probit</td>
</tr>
<tr>
<td>Censored data Conner solutions</td>
<td>• Tobit</td>
</tr>
</tbody>
</table>

### X independent

<table>
<thead>
<tr>
<th>Correlated with each other</th>
<th>• proxy, • instrumental variable, • 2SLS, 3SLS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Selection</td>
<td>• Double hurdle, • Cragg, • Heckman</td>
</tr>
<tr>
<td>Y explains X</td>
<td>• Systems equations</td>
</tr>
</tbody>
</table>

---

**Note:**
- Linear models are used when the dependent variable follows a normal distribution.
- Categorical models are used when the dependent variable is categorical.
- Ordered models are used when the dependent variable is ordered but not normally distributed.
- Multinomial models are used when the dependent variable has more than two categories.
- Censored data models are used when the dependent variable has values that are not fully observed.
In addition...

• When Y and X are repeated for same observation, use time series, panel methods, DID, treatment effects

• When the dependent variable is count, use either Poisson or Negative Binomial models
Appendix 7b: Linear Regression Lectures
LINEAR REGRESSION ANALYSIS

- A statistical tool for evaluating the relationship of one or more independent variables (x) to a single continuous dependent variables (y).
- Regression analysis characterizes the relationship between the dependent and independent variables by looking determining the extent, direction and strength of the association.

Types of Linear Regression Models

- There are two broad types of linear regression models:
  1. Simple regression model
  2. Multiple regression model

Simple Linear Regression Model

- This is a model that associates one continuous dependent variable to one independent variable.
- For example, we may wish to consider how maize yield depends on several amount of fertilizer. Assume we have the following observations:

<table>
<thead>
<tr>
<th>Fertilizer (kg/ha)</th>
<th>Yield (tones/ha)</th>
</tr>
</thead>
<tbody>
<tr>
<td>100</td>
<td>4.0</td>
</tr>
<tr>
<td>200</td>
<td>5.0</td>
</tr>
<tr>
<td>300</td>
<td>5.0</td>
</tr>
<tr>
<td>400</td>
<td>7.0</td>
</tr>
<tr>
<td>500</td>
<td>6.5</td>
</tr>
<tr>
<td>600</td>
<td>6.5</td>
</tr>
<tr>
<td>700</td>
<td>8.0</td>
</tr>
</tbody>
</table>

- In this case, y = the dependent continuous variable and fertilizer level = independent (sometimes called independent variable or regressor) variable (x). Thus, the above regression is specified as:

\[ y = \alpha + \beta x + \varepsilon \quad (1) \]

Where \( \alpha \) and \( \beta \) are coefficients to be estimated. Specifically, \( \alpha \) is the intercept or constant of the equation and \( \beta \) is the slope of the equation line.

The intercept gives us the variation in y that is not being explained by the variable(s) under consideration. This is in
recognition of the fact that it is almost impossible to include all the independent variables that affects the dependent variable. For instance, in our example in Table 1, we have only included fertilizer but we know that yield is not affected by fertilizer alone. Other factors such as land size, soil type, rainfall, etc. are also bound to affect the yield.

The slope \( \beta \), tells us the magnitude and direction of the relationship between the dependent variable \( y \) and the independent variable. The magnitude is determined by looking at the size of the coefficient while direction is judged by the sign (i.e. positive or negative) of the coefficient. If \( \beta \) is positive it implies that there is a positive relationship between \( y \) and \( x \), i.e. increase in \( x \) results in an increase in \( y \). On the other hand, a negative \( \beta \) implies a negative relationship between \( y \) and \( x \), i.e. an increase in \( x \) results in a decrease in \( y \). Specifically, \( \beta \) shows the magnitude of increase in \( y \) as a result of a unit change in \( x \).

The fitted line of a simple regression model is written as:

\[
\hat{y} = \hat{\alpha} + \hat{\beta} x
\]  

(2)

**Multiple Linear Regression Model**

- This is a model that associates one continuous dependent variable to two or more independent variables.
- Multiple regression analysis examines the simultaneous effect of several independent variables.
- Mathematically, the model is specified as:

\[
y = \alpha + \beta_1 x_1 + \beta_2 x_2 + \ldots + \beta_k x_k + \varepsilon_i
\]  

(3)

- When fitted, the model is specified as:

\[
\hat{y} = \hat{\alpha} + \hat{\beta}_1 x_1 + \hat{\beta}_2 x_2 + \ldots + \hat{\beta}_k x_k
\]  

(4)

- The coefficients are interpreted in the same way as in simple regression model. The only difference is that in multiple regression model we have several coefficients to be estimated i.e. \( \alpha, \beta_1 \) to \( \beta_k \). The \( \beta_i \) show the magnitude and direction of association of their respective variables \( (x_i) \) and the dependent variable \( (y) \). For instance, \( \beta_1 \) measures the change in \( y \) associated with one unit change in \( x_1 \) assuming all other values for the remaining variables are held constant.
- Since other variables are held constant, the slope parameters \( (\beta_s) \) are called partial partial regression coefficients (i.e. they correspond to partial derivatives of \( y \) with respect to \( x_{is} \)).
• One of the frequently asked questions in multiple regression analysis is: how many independent variables can be used? In other words, is there a limit to the number of variables to be included in a multiple regression model?
• Yes, there is a limit to the number of independent variables we can use. This limit is set by:

\[ k < n - 1 \]  

(5)  

where \( k \) = number of independent variables and \( n \) = sample size. Therefore the number of independent variables should not exceed the sample size minus one, e.g. if the sample size is 30, the maximum number of independent variables is 29.

**Estimating a Linear Regression Model**

• Regression models can be estimated in two ways:
  1. Plotting \( x \) and \( y \) (this is easier with simple linear regression)
  2. Ordinary Least Squares (OLS) method.
• Here we are going to emphasize on OLS. You are however, advised to read any statistical book on the other methods of estimating regression models.

**OLS Estimation and Interpretation**

• Both ‘manual’ calculations and computers can be used to estimate regression models.
• It is easier to calculate [manually] OLS estimates for simple regression model than it is for multiple regression models. You are thus strongly encouraged to practice model estimation using any statistical package of your choice such as SPSS.

**Manual Estimation of Linear Regression Model**

• For simple regression model, the estimates \( \alpha \) and \( \beta \) are estimated by simultaneously solving the following normal equations:

\[
\sum y = n\alpha + \beta \sum x \\
\sum xy = \alpha \sum x + \beta \sum y^2
\]  

(6)

**Example 1**

• From the following data, estimate the regression model:

| \( Y \) | 9 | 11 | 5 | 8 | 7 |
| \( X \) | 6 | 2  | 10| 4 | 8 |
From the normal equations (6), the \( \alpha \) and \( \beta \) are to be estimated. To do this we need \( \sum x \), \( \sum y \), \( \sum xy \) and \( x^2 \). These are calculated in Table 2:

**Table 2: Estimating Regression Model**

<table>
<thead>
<tr>
<th>( X )</th>
<th>( y )</th>
<th>( xy )</th>
<th>( x^2 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>6</td>
<td>9</td>
<td>54</td>
<td>36</td>
</tr>
<tr>
<td>2</td>
<td>11</td>
<td>22</td>
<td>4</td>
</tr>
<tr>
<td>10</td>
<td>5</td>
<td>50</td>
<td>100</td>
</tr>
<tr>
<td>4</td>
<td>8</td>
<td>32</td>
<td>16</td>
</tr>
<tr>
<td>8</td>
<td>7</td>
<td>56</td>
<td>64</td>
</tr>
</tbody>
</table>

\[ \sum x = 30 \quad \sum y = 40 \quad \sum xy = 214 \quad \sum x^2 = 220 \]

n (sample size) is 5.

- Substituting, the above summations into the normal equations, we have:
  \[
  40 = 5\alpha + 30\beta \\
  214 = 30\alpha + 220\beta
  \]

- This gives us \( \alpha = 11.9 \) and \( \beta = -0.65 \)
- Thus the regression equation is
  \[ y = 11.9 - 0.65x \]
- The interpretation of the above equation is that there is a negative (inverse) relationship between \( y \) and \( x \). Specifically, a unit increase in \( x \) results in a decrease of \( y \) by 0.65. The constant (11.9) tells us that in absence of \( x \), \( y \) is 11.9. As already stated, this accounts for those excluded variables that could also influence \( y \).

- It has again to be pointed out that in practice, these manual calculations are very cumbersome. Hence we need to look at how we can solve the same model using Stata (you could use any appropriate statistical package).

**Using Stata to Estimate Regression Equation**

- With computers, it is faster to estimate regression models than the manual calculations. It does not matter how many variables we have, the computer will give us the results within seconds upon executing appropriate commands.
- **Example 1 (simple regression model):** Entering the data is Stata and running the regression yielded the following output:
Output 1: Simple Linear Regression

Just like the manual calculations, the estimates are 11.9 (constant) and -0.65 (x).

Example 2 (multiple regression model): A family planning NGO wished to examine the relationship between expenditure, income and family size in Malawi. The organization drew a random sample of ten families and obtained the results given in Table 3. Determine the regression equation.

Table 3: Expenditure, Income and Family Size

<table>
<thead>
<tr>
<th>Family</th>
<th>Expenditure (Hundreds of Kwacha)</th>
<th>Annual Net Income (Thousands of Kwacha)</th>
<th>Family Size X2</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>19</td>
<td>6</td>
<td>3</td>
</tr>
<tr>
<td>B</td>
<td>20</td>
<td>7</td>
<td>4</td>
</tr>
<tr>
<td>C</td>
<td>14</td>
<td>6</td>
<td>2</td>
</tr>
<tr>
<td>D</td>
<td>10</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>E</td>
<td>22</td>
<td>7</td>
<td>6</td>
</tr>
<tr>
<td>F</td>
<td>23</td>
<td>8</td>
<td>5</td>
</tr>
<tr>
<td>G</td>
<td>17</td>
<td>6</td>
<td>3</td>
</tr>
<tr>
<td>H</td>
<td>15</td>
<td>4</td>
<td>3</td>
</tr>
<tr>
<td>I</td>
<td>7</td>
<td>2</td>
<td>4</td>
</tr>
<tr>
<td>J</td>
<td>23</td>
<td>10</td>
<td>3</td>
</tr>
</tbody>
</table>

The equation to be estimated is:
\[ \hat{y} = \hat{\alpha} + \hat{\beta}_1 x_1 + \hat{\beta}_2 x_2 \]

Stata gives us the following outputs:
Output 2: Multiple Linear Regression

- Thus the estimated regression equation is:
  \[ \hat{y} = 0.385 + 2.176x_1 + 0.963x_2 \]

- **Interpretation**: In general, both annual net income and family size positively affect expenditure. Holding other factors constant, an increase of family annual net income of MK1 increases expenditure by MK2.176. Similarly, an increase of family size by one member increases expenditure by MK0.963.
Testing Validity of a Regression Model

- So far, the interpretations that we have been making on the models do not have statistical backing because the validity of the models was not tested.
- There are three tests that are used to test validity of a model:
  1. F-statistic
  2. Coefficient of Determination ($R^2$)
  3. t-statistic

1. F-Statistic

- This measures the overall significance of the model.
- The F-statistic tests:
  - The null hypothesis (NH): $\beta_1 = \beta_2 = \ldots = \beta_k = 0$, i.e. that the coefficients are equal to zero implying that there is no relationship between the dependent variable and the independent variables. *(As an example try to substitute zero for the coefficients estimated in the previous examples and see how the equation looks like).*
  - The alternate hypothesis (AH): $\beta_i \neq 0$, i.e. none of the coefficients is equal to zero.
  - Note that if the NH is accepted it implies that there is no relation between the dependent and independent variables even if the coefficients are not zero.
  - If the NH is rejected, i.e. the F-statistic is valid, then the overall model is valid and we can go ahead and check the other two tests ($R^2$ and t-statistic).

- How do we do the F-test?
  - You can calculate the F-test manually or using the computer. This lecture will concentrate on computer use. You are strongly advised to read any statistical book for manual f-statistic calculation and how it is applied.
  - When you are running a regression model in Stata, by default it gives you all the validity tests including the F-test. What is important is how to know how to use and interpret them.
• The F-test is given at the top-right of the Stata Outputs.
• For Output 1, the F-statistic is 16.35 and is significant at 5%. Therefore the model is overall significant. We reject the null hypothesis that there is no relationship between Y and X. The bullets below explain how to determine significance.

• Had we calculated the F-value manually (through the ANOVA table on the top-left), we would have compared it with the tabulated (found at the back of most statistical books) F-values at different levels of significance (1%, 5% and 10%). If the calculated F-value is greater than the tabulated F-value then the calculated F-value is significant, hence we reject the null hypothesis and accept the alternate hypothesis that at least some of the coefficients are not equal to zero thereby concluding that the overall model is valid. The opposite is true if the calculated F-value is less than the tabulated F-value. However, this is not necessary if we have used a computer as the last column (Sig. Column) will tell us whether the estimated F-value is significant or not.

• We normally check the significance at three levels 1%, 5% and 10%. Note that 1% has the highest level of confidence (99%) followed by 5% (95%) and 10% (90%). That is why in some books they refer to 1% as the highest level; 5% as the moderate (medium) level; and 10% the lowest level.

• As a rule of thumb, start with the highest level then if necessary (i.e. if not significant at a higher level) then go down to the lower level, and so on, since when something is say, significant at 1% it is automatically also significant at the other lower levels and not vice versa.

• When using a computer (could be Stata or any package), for the F-test to be significant, the p-value for F (Prob > F) should be less than the level (1%, 5% and 10%) at which you are testing the F-value. A word of caution though, make sure that you are comparing the figures in the same format. The p-value (0.027) is given as a decimal fraction while the levels are in percentages. Either you convert the p-value (0.027) to percentage by multiplying it by 100 (thus giving us 2.7%) and then make comparisons or convert the levels to decimal fractions by dividing them by 100 (thus 1% becomes 0.01; 5% = 0.05 and 10% = 0.10). Both ways (comparing as decimal fractions or as percentages) yield the same result.

• The above figure (0.027 or 2.7%) is more than 0.01 or 1%. This means the F-value is not significant at 1% and we need to check at
the lower level of 5% (0.05). Certainly, 0.027 (2.7%) is less than 0.05 (5%). Therefore, overall, the model is significant at 5% (p<0.05). We therefore reject the null hypothesis that the coefficients are equal to zero and conclude that the model is valid.

- Note: in social science research, if a model (or any variable as we are going to see later on) is not significant at 10%, then it considered not significant as this is the lowest acceptable level of significance.

- From Output 2, similarly, this model is significant at 1% (p<0.01) (i.e. 0.0007<0.01). We therefore reject the null hypothesis and accept the alternate hypothesis. Thus, the model is valid.
- The F-test is a necessary but not a sufficient test for checking validity of a model. To sufficiently check regression model validity, we need to check the other two tests of (R²) and t-statistic.

### 2. Coefficient of Determination (R²) Test

- R² measures the proportion of the variation in the dependent variable (y) that is being explained by the independent variable(s).
- In our examples, the second tables in the output.

- The R² for Output 1 is 0.845, i.e. 84.5%. This means the predictor (independent) variable (x) is explaining 84.5% of the variation in y. The other, 15.5% is attributed to variables not included in the model. This is normal as it has already been said that in any regression model, it is impossible to include all the independent variables.

- From Output 2, the R² = 0.8725 meaning that the independent variables (X1 and X2), ie. annual income and family size are explaining 87.25% of the variation in Y (expenditure).

**Problem of Using R²**

- The major problem is that R² is sensitive to the number of independent variables included in a regression model. The greater the number of independent variables the higher the R² is likely to be, i.e. the more the independent variables we add (even if they are not valid), the bigger the R² becomes.
- This problem arises because R² does not take into account the number of degrees of freedom, i.e. R² is given by the following formula:
\[ R^2 = \frac{\text{regression sum of squares (SSR)}}{\text{total sum of squares (SST)}} = \frac{\sum (\hat{Y}_i - \bar{Y})^2}{\sum (Y_i - \bar{Y})} \]  

(7)

- To solve this problem, when testing the validity of a regression model we use the Corrected or Adjusted R\(^2\) (denoted as \(\bar{R}^2\)) which takes degrees of freedom into account as given in the following formula:

\[ \bar{R}^2 = 1 - (1 - R^2) \frac{(n-1)}{(n-k-1)} \]  

(8)

Where \(R^2\) = the coefficient of determination; \(n\) = sample size; and \(k\) = number of independent variables.

- For Output 1, the adjusted \(R^2\) is 0.793 as shown in the table below:
- Similarly, for Output 2, the \(R^2\) is 0.836.
- The \(R^2\) adjusted is interpreted in the same way as \(R^2\).

- The models have so far satisfied two validity tests (F-statistic and Coefficient of Determination), but we still need to do a final t-statistic test, which tests the significance of the individual independent variables.

4. **t-statistic Test**

- t-statistic is a test of significance for individual explanatory variables and the constant term within a model.
- The t-statistic is found by coefficient (unstandardized) divided by the standard error. The t-statistic is calculated for the constant and all the other coefficients.
- In our two examples, as already discussed, by default the t-statistics are also calculated. For example 1, the t-statistic for the coefficient is 11.16 and is significant at 1% (p<0.01). \(x\) has a t-statistic of -4.044 and is significant at 5% (p<0.05). Thus we can conclude with certainty that \(x\) negatively affects \(y\).
- In the same way, check whether the independent variables in Output 2 are significant or not.
Appendix 8a: Probit and Logit Models
Probit and Logit Models

Francis Darko
When do we use Probit and Logit models?

• When dependent variable is binary (i.e. takes only two values, e.g. 0 or 1)
  \[ Y = X\beta + u \]

• Examples:
  1. Participation in FISP (Y=1 if yes; Y=0 if no)
  2. Fertilizer adoption (Y=1 if yes; Y=0 if no)
  3. Admission into LUANAR? (Y=1 if yes; Y=0 if no)

• In such cases, linear regression model is not appropriate because the dependent variable is not continuous (also Homoscedasticity and Normality assumptions are violated)
Difference between Logit and Probit

• The difference between the probit and logit lies in the assumption that the researcher makes about the error term \( Y = X\beta + u \)
  
  Probit: Normal distribution of errors
  
  Logit: Logistic distribution of errors

• Both however yield identical results (marginal effects)
Interpretation of Results

• Interpretation of results is not as straightforward as in the linear regression model

• You can only interpret the sign of the coefficients; not the magnitude

• Marginal effects
Appendix 8b: Probit and Linear Regression Models
Probit and Linear Regression Analyses

By
Charles Jumbe, PhD
probit foreign price pricesq kpl length trunk rep78; //do a probit analysis of factors that influence whether to purchase from outside or procure internally with rep78 as a continuous variable
margins, dydx(*); /* Generate marginal effects for all variables*/
mfx; /* Generate marginal effects */
probit foreign price pricesq kpl length trunk i.rep78; //do a probit analysis of factors that influence whether to purchase from outside or procure internally with rep78 as an integer: Note, 1.rep78, 2.rep78 & 5.rep78 will be dropped due to few observations
*RECODING VARIABLES;

/* Recoding variable: A variable X with continuous values may be collapsed into a dummy variable with 2 categories. One category can be 2 or below. The other category can be above 5 with the following command: */

recode rep78 0/2=0 3/5=1; // recoding repair record into 2 categories with 0 if rep78 is less than 2 and 1 if rep78 is more than 2

list rep78; /* display the recoded variables */

probit rep78 price pricesq foreign kpl length trunk; // do a probit analysis of factors that influence whether or not to repair the vehicle. Note we recoded rep78 to 0 if rep78<2 and 1 if rep >2
**PROBIT REGRESSION ANALYSIS**

`probit` rep78 price pricesq foreign kpl length trunk ; //do a probit analysis of factors that influence whether or not to repair the vehicle. Note we recoded rep78 to 0 if rep78<2 and 1 if rep >2: Note foreign will be dropped due to collinearity

margins, dydx(*) ; /* Generate marginal effects for all variables*/
mfx; /* Generate marginal effects */
*regress price foreign kpl headroom weight length rep78; ///regression- of the determinants of vehicle price including integer rep78

bysort foreign: regress price foreign kpl headroom weight length rep78; //Do separate regressions for domestics and foreign vehicles: note if you erroneously include foreign as a regressor, it will be dropped. Again rep78 is dropped due to collinearity

regress price foreign kpl headroom weight length kpl#foreign; ///regression-of the determinants of vehicle price including an interaction between kpl and foreign i.e., for each value of kpl against foreign & domestic

gens repkpl=rep78*kpl; /*Generate interaction between kpl & rep78 */
regress price foreign kpl headroom weight length repkpl; /*Do a regression that includes an interaction variable kpl*rep78 */