



Data Analysis In Economics by Using Stata

Dr. Alok Kumar Pandey

Assistant Professor Department of Economics DAV PG College, Varanasi

ABSTRACT

The present paper discuss about the basic use of STATA software. In economic discipline researches are supported by data. For the purpose sometimes we need calculation on data to show results and support our theory in the research. This paper help new research to understand about the STATA software and also provide some initial guidelines for using the software. shrimp farming in India expanded rapidly after trade liberalisation initially were introduced in early 1990s. This paper examines the social, economic and environment impact that has been generated in the community along Odisha-coast in East India. The result suggests that macro-level policies such as trade liberalization are useful at the national level, but at the local level they can generate imbalanced growth. Although the shrimp farming has generated substantial foreign exchange at the local level, it has widened the gap between rich and poor. To protect the livelihoods of the rural communities and the environment India needs to formulate effective coastal resource management policies and established adequate institutions at the local level this will help ensure a stable source of income from shrimp farming for small scale farmers and minimise adverse impact on the local environment.

KEYWORDS

STATA, DATA ANALYSIS, SOFTWARE

Introduction:
Stata is a statistical analysis package with programming capabilities. A variety of tasks can be accomplished by issuing commands interactively from the command line. There are commands built into Stata that allow the user to do statistical analysis such as regression on pre-formatted data sets.

Stata's commands can also be combined in sequences to solve complex data management and analysis problems. These sequences of commands can be saved in Stata "do files" and run over and over.

Starting STATA
Go to start icon on windows computer. We can also start SATA by double clicking on an STATA file.



Various Windows:
Results window:
The big window, result of all Stata commands appear here (except graphs which are shown in their own windows).

Command window:
It appears below the results window. Commands are entered here. Just type the command and hit the "Enter" key. Alternatively, groups of commands can be entered into do files which can then be executed.

Review window:
All Stata commands have been entered are appear here. A

previous command can be repeated by double-clicking the command in the Review window.

Variables window:
It shows the record of all variables in the dataset that is currently being used.

Various files in STATA:
.gph–Stata graph file
.smcl –Stata output files
.do –Stata command file
.ado -Stata program
.dta -Stata dataset

STATA'S Online Help
There is online help available inside Stata. To get help for a command, simply type "help" and the name of the command. If the command exists and we typed in the name correctly then the help screen for that command will display.

LOG
The log command: From then on everything we type and all the output Stata produces automatically is recorded in the .log file we specify.

log using xyz.log

The .log file saves automatically when we exit Stata.

STATA'S Built-in Calculator: display
There is a Stata command called display (abbreviated di) that when used from the command line acts as a built-in calculator: di (2+2)*5

Data Files
A data set is just a file in which rows represent observations and columns represent variables.

Stata can read data sets in various text formats as well as in Stata's proprietary format. A common text format is the delimited file. Delimited files are most commonly tab- or comma-delimited. This just means that the variables in each observation are entered one after the other on a line and separated by tabs or commas, while the observations are separated by hard returns. The example above is actually how a comma-de-

limited text file would look if opened in Word.

The command syntax to read in a tab- or comma-delimited file is: `insheet using [filename]` where filename is the name of the file that contains the tab- or comma-delimited data.

`insheet` is often used to read spreadsheets saved as “`csv`” (comma-delimited) files from a package such as Excel.

There are two other commands `infile` and `infix` - that read other, less common types of text files.

Save File in Stata:

For new file: We can save a file with the Stata `save` command in Stata format:

```
save [filename]
```

```
save xyz
```

For update file: The Stata `save` command with the `replace` option can be used

```
save [xyz], replace
```

where xyz is the name of the file we wish to replace, (i.e. `save XYZ, replace`)

To save an updated version of the active file, : `save, replace`

Missing Values

Sometimes a variable is missing (there is no value i.e. the person didn't answer the question) for some observations.

Missing values in numeric variables are represented by a `(.)`.

Observations with missing values are left out of tables produced by `tab`, and are also left out of regressions.

Stata Variables Types

There are two types of variables in Stata: numeric and string. A third type, `date`, is really a special type of numeric, as we will see. Numeric variables are simple - they contain numbers. String variables contain text which can contain any characters on the keyboard: letters, numbers, and special characters. On `auto3`, `make` is a string variable - all the others are numeric. We can do numeric calculations and statistical analysis on numeric variables - we can't on string variables. String variables are usually used as identifiers for the observation.

One of the numeric variables, `date`, is intended to represent a date - let's say it was the date the data about each car was collected.

Variable Naming

The rules for naming variables in Stata are simple:

1. Stata is case-sensitive, so using all lower case letters in variable names is a good idea.
2. They can contain no more than 32 characters.
3. They can contain letters, numbers, or underscores (`_`).
4. Spaces or other special characters (like `&`, `*`, `%`, etc.) are not allowed.
5. The first character must be a letter or underscore, not a number. Starting variable names with underscores is a really bad idea, since Stata's built-in variables begin with an underscore.

Summary Statistics

Two commands that are useful for getting basic descriptive statistics for variables are `summarize` and `tabulate` (abbreviated `sum` and `tab` respectively). `sum` gives the number of valid observations, mean, standard deviation, minimum and maximum values for any variables we specify. We can do the entire dataset at once:

```
sum
```

or just a subset of variables:

```
sum income age
```

The `tab` command gives us a frequency distribution (for one variable) or a crosstabulation (for two).

```
tab gender
```

This gives frequency distribution for gender.

```
tab category gender
```

This provides crosstabulation showing values of category for each level of gender

```
tab category gender, row col
```

same as above, but with row and column percentages

```
tab category gender, row col chi2
```

same as above, but calculates Pearson's chi-squared for the

The `summarize` command can be combined with the `tabulate` command to produce summaries of one variable for each value of another. The following table shows separate summaries of income for males and females.

Correlation

For correlation “`correl`” command used in the stata.

```
correl income age
```

Regression

`regress` (abbreviated `reg`) is the command that runs simple linear regression. A binary regression of income on age would look like this:

```
reg income age
```

Predicted Values

The `predict` command is used after regression commands to calculate predicted values, residuals, and other quantities based on the regression results, and store them in new variables, which can then be analyzed or used in further calculations. We will use ordinary regression as an example.

First, run a regression:

```
reg income age gender
```

Then, to store the predicted values (the default) in a variable called `pred_income`, simply type:

```
. predict pred_income
```

To find the residuals and store them in a variable called `r`, type:

```
. predict r, resid
```

STATA Commands used for Analysis:

Sl. No	Command	Purpose
1	<code>cd</code>	Change directory
2	<code>use</code>	Load dataset into memory
3	<code>list</code>	List the contents of a dataset
4	<code>log</code>	Create a log file
5	<code>summarize</code>	Descriptive statistics
6	<code>tabstat</code>	Table of descriptive statistics
7	<code>table</code>	Create a table of statistics
8	<code>sort</code>	Sort observations in a dataset
9	<code>Codebook</code>	Show codebook information for file
10	<code>label data</code>	Apply a label to a data set
11	<code>Order</code>	Order the variables in a data set
12	<code>Label variable</code>	Apply a label to a variable
13	<code>label define</code>	Define a set of labels for the levels of a categorical variable
14	<code>label values</code>	Apply value labels to a variable

15	List	Lists the observations
16	Rename	Rename a variable
17	Recode	Recode the values of a variable
18	Generate	Creates a new variable
19	Replace	Replaces one value with another value
20	Egen	Extended generate - has special functions that can be used when creating a new variable

The "if" qualifier is used to isolate a set of observations with variables meeting some particular criteria. This is useful for creation of Dummy Variables.

Sl no	Operator	Meaning
1.	==	equal to
2.	>	greater than
3.	>=	greater than or equal to
4.	<	less than
5.	<=	less than or equal to
6.	!= or ~=	not equal to