

# STATA FOR RESEARCH

Noyel Sebastian

PhD Scholar  
CITD/SIS, JNU

February 21, 2025

## ① Introduction

## ② Basics

## ③ Data Cleaning

## ④ Stats and Graphs

## ⑤ Regression

# Some Basics

- STATA is a powerful tool for data analysis and statistical modelling.
- STATA Manual - basic handbook.

# Some Basics

- STATA is a powerful tool for data analysis and statistical modelling.
- STATA Manual - basic handbook.
- Have more doubts?



# Some Basics




- STATA is a powerful tool for data analysis and statistical modelling.
- STATA Manual - basic handbook.
- Have more doubts?



- Let's familiarise the interface.

# Some Basics





- Different types of STATA files:
  - DTA (.dta): Dataset - STATA's native data file format
  - DO (.do): Scripts a sequence of STATA commands
  - LOG (.log): Records command history and output

 log file	20-02-2025 10:57	Text Document
 STATA Tutorial	19-02-2025 23:21	Microsoft Excel W...
 Student_data	20-02-2025 10:59	DTA File
 Tutorial Session Do File	20-02-2025 12:15	DO File

- An ideal practice: write commands in .do file.

# Some Basics

- Different types of STATA files:
  - DTA (.dta): Dataset - STATA's native data file format
  - DO (.do): Scripts a sequence of STATA commands
  - LOG (.log): Records command history and output

 log file	20-02-2025 10:57	Text Document
 STATA Tutorial	19-02-2025 23:21	Microsoft Excel W...
 Student_data	20-02-2025 10:59	DTA File
 Tutorial Session Do File	20-02-2025 12:15	DO File

- An ideal practice: write commands in .do file.
- How about making some comments?

# Comments and Readability

Type	Stata Command	Purpose
Full-line comments	*	Create section dividers
Line breaks	///	To break commands
In-line comments	//	For short comments
Block comments	/*...*/	Use to explain in detail

- We will see the use of each of these later.



# Let's start working

- Using STATA's in-built data sets - `sysuse`
- See all available datasets:  
`sysuse dir`

# Let's start working

- Using STATA's in-built data sets - `sysuse`
- See all available datasets:  
`sysuse dir`
- To load STATA's built-in dataset:  
`sysuse auto, clear`
- Why do we use `clear`?

# Let's start working

- Using STATA's in-built data sets - `sysuse`
- See all available datasets:  
`sysuse dir`
- To load STATA's built-in dataset:  
`sysuse auto, clear`
- Why do we use `clear`?
- Let's open a `.dta` file from menu.

# More on data

- What is a Working Directory?  
Default folder where STATA looks for and saves file when you run commands

# More on data

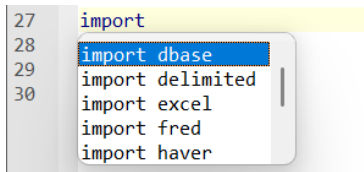
- What is a Working Directory?  
Default folder where STATA looks for and saves file when you run commands
- commands:  
`cd "file path"`
- To see the current working directory, `pwd`  
let's see..

# Importing Data

- Now, we will import data to STATA.

# Importing Data

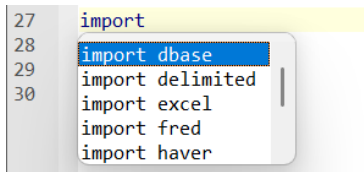
- Now, we will import data to STATA.



```
import excel "file path"
```

# Importing Data

- Now, we will import data to STATA.



```
import excel "file path"  
import excel "file path",firstrow
```

- saving the file:  
save "filename.dta", replace
- replace?



# Let's advance

- To view the data, we will **browse** it. A short command:  
`br`

# Let's advance

- To view the data, we will **browse** it. A short command:  
`br`
- Remember the Log file?
- Open a log file:  
`log using "name.log", replace`
- If you want to continue writing on the same log file:  
`log using "name.log", append`
- Close the log file: `log close`

# Variables

- STATA stores variables in different types. To see the variables use, `describe` or `de`
- The command provides an overview of the dataset, including variable names, types, labels, and storage formats.
- How to distinguish?

## Types

Numeric (Integers, long, float)

String

Labels

## Colour

Black

Red

Blue


# Create and Modify Variables

- To create new variables, we use **gen** or **generate**
- Read more on **egen** command

# Create and Modify Variables

- To create new variables, we use **gen** or **generate**
- Read more on **egen** command
- To improve readability, we use Labels:

Variables ▼ 🔍 ✕

 Filter variables here

Name	Label
make	Make and model
price	Price
mpg	Mileage (mpg)
rep78	Repair record 1978
headroom	Headroom (in.)
trunk	Trunk space (cu. ft.)
weight	Weight (lbs.)
length	Length (in.)
turn	Turn circle (ft.)
displacement	Displacement (cu. in.)
gear_ratio	Gear ratio
foreign	Car origin

- 1 Variable Labels – Provide a descriptive name for a variable.
- 2 Value Labels – Assign text to numeric values of a variable.

Car origin	Freq.	Percent	Cum.
0. Domestic	52	70.27	70.27
1. Foreign	22	29.73	100.00
Total	74	100.00	

# Create and Modify Variables

- How to label a variable?

```
label variable foreign "Car origin"
```

```
label variable rep78 "Repair record 1978"
```

# Create and Modify Variables

- How to label a variable?

```
label variable foreign "Car origin"
```

```
label variable rep78 "Repair record 1978"
```

- Let's put some value labels. See the variable **foreign**:

- ① Define the value label

```
label define car 0 "Domestic" 1 "Foreign"
```

# Create and Modify Variables

- How to label a variable?

```
label variable foreign "Car origin"
```

```
label variable rep78 "Repair record 1978"
```

- Let's put some value labels. See the variable **foreign**:

- 1 Define the value label

```
label define car 0 "Domestic" 1 "Foreign"
```

- 2 Assign the value label to the variable

```
label values foreign car
```



# Practice Time

- ① Create a new working directory to a folder in your desktop.
- ② Import the given excel file to STATA and save it in default STATA file format. [use commands]
- ③ Start a Log file session. Name the log file appropriately.
- ④ Create a new variable **score** which shows a student's total score. Label the variable appropriately.
- ⑤ Provide appropriate value labels for the variable **Tuition**.

# Organise and Manipulate

- We can **sort** variables, arranging the dataset in ascending order based on the given variable(s).

# Organise and Manipulate

- We can **sort** variables, arranging the dataset in ascending order based on the given variable(s).

Example: Sort the Student data based on their ages

```
sort Age
```

# Organise and Manipulate

- We can **sort** variables, arranging the dataset in ascending order based on the given variable(s).  
Example: Sort the Student data based on their ages  
`sort Age`
- **bysort** will sort the dataset based on the grouping variable, then apply the command.  
`bysort residence:su English`  
Hold on, we will apply this soon!
- What if you have some conditions?

# Organise and Manipulate

- We can **sort** variables, arranging the dataset in ascending order based on the given variable(s).  
Example: Sort the Student data based on their ages  
`sort Age`
- **bysort** will sort the dataset based on the grouping variable, then apply the command.  
`bysort residence:su English`  
Hold on, we will apply this soon!
- What if you have some conditions?  
Rescue: **if** - will execute the command for those observations that meet the specific condition.

# If and more

- Let's generate a new variable “A\_grade” for Maths score  $> 40$  and if the student attends Tuition.

# If and more

- Let's generate a new variable “A\_grade” for Maths score  $> 40$  and if the student attends Tuition.

```
gen A_grade = Maths > 40 if Tuition == 1
```

- == ??

STATA has logical operators. Some will be handy now on:

== (Equal to), != (Not equal to), & (And), | (Or). Read more.

## If and more

- Let's generate a new variable “A\_grade” for Maths score  $> 40$  and if the student attends Tuition.

```
gen A_grade = Maths > 40 if Tuition == 1
```

- `== ??`

STATA has logical operators. Some will be handy now on:

`==` (Equal to), `!=` (Not equal to), `&` (And), `|` (Or). Read more.

- Some extras: `inlist` & `inrange`

- Quick example:

```
gen B_grade = inrange(Maths,30,40) & Tuition == 1  
gen outstand = inlist(Maths,50)|inlist(English,50)
```



# Replace and Recode

- To alter individual values of a variable, **replace** command can be used.

# Replace and Recode

- To alter individual values of a variable, **replace** command can be used.

Example: Replace the missing values in the variable `A_grade` with 0.

```
replace A_grade = 0 if A_grade == .
```

- To modify categorical or grouped numeric variables, we use **recode** command

Example: Create a new variable `Age_Group` where we recode the variable `Age` so that children aged 5-12 takes a value 1 and 0 if aged 13-20.

# Replace and Recode

- To alter individual values of a variable, **replace** command can be used.

Example: Replace the missing values in the variable `A_grade` with 0.

```
replace A_grade = 0 if A_grade == .
```

- To modify categorical or grouped numeric variables, we use **recode** command

Example: Create a new variable `Age_Group` where we recode the variable `Age` so that children aged 5-12 takes a value 1 and 0 if aged 13-20.

```
recode Age (5/12 = 1) (13/20 = 2), gen(Age_Group)
```

# Organise and Manipulate

- To simplify the analysis, keep the required variables only. (NFHS has n-variables!!)  
**keep** helps to retain the necessary variables  
**drop** deletes the variables.
- Some hacks:  
Make use of **preserve** and **restore** commands while you drop variables in a session. Explore more.

# Organise and Manipulate

- To simplify the analysis, keep the required variables only. (NFHS has n-variables!!)  
`keep` helps to retain the necessary variables  
`drop` deletes the variables.
- Some hacks:  
Make use of `preserve` and `restore` commands while you drop variables in a session. Explore more.
- Quick Example: Drop the variable `outstand`  
`drop outstand`

# Descriptive Statistics

- To get the summary statistics, we use `summarize` or `su` command.
  - returns observations, mean, SD, min, max.
- `tabulate` (`tab`) command is used to generate Frequency tables.  
Example: `tab Tuition`

# Descriptive Statistics

- To get the summary statistics, we use `summarize` or `su` command.
  - returns observations, mean, SD, min, max.
- `tabulate` (`tab`) command is used to generate Frequency tables.  
Example: `tab Tuition`
- Some extras:  
`numlabel, add`  
`tab Tuition`  
See the difference!
- There are more to work on with `tab`. Explore more.
- Let's jump to data visualisation in STATA.

# Regression

- Regression helps to understand the relation between one or more variables.
- See this random equation:

$$Y_i = \alpha + \beta_1 \textit{Weight} + \beta_2 \textit{Mileage} + \epsilon_i$$

where  $Y_i$  is the price of the automobile.

- How to run this in STATA?



# Regression

- Regression helps to understand the relation between one or more variables.
- See this random equation:

$$Y_i = \alpha + \beta_1 Weight + \beta_2 Mileage + \epsilon_i$$

where  $Y_i$  is the price of the automobile.

- How to run this in STATA?  
`regress price weight mpg`
- What if you have categorical variables?

# Exporting Results

- How to export results to word?
- Various packages in town: `outreg`, `estout`.
- Some extras: `ssc install`
  - `ssc install ivreg2` for IV regression (Also use `ivregress 2sls`)  
`reghdfe` for TWFE [many more for DiD]  
`coefplot` for coefficient plots, [say, for Event Study]
- Let's see how to export to word.

# Exporting Results

- How to export results to word?
- Various packages in town: `outreg`, `estout`.
- Some extras: `ssc install`
  - `ssc install ivreg2` for IV regression (Also use `ivregress 2sls`)  
`reghdfe` for TWFE [many more for DiD]  
`coefplot` for coefficient plots, [say, for Event Study]
- Let's see how to export to word.
- Need to be a pro? export to LaTeX. More beautiful tables.

Questions?

Thank You!