Introduction to STATA

Dr. Md. Atiqul Islam
M.Sc. (SUST, BD), M.Sc. (UHasselt, BE), PhD (RuG, NL)

Professor
Department of Statistics
Jagannath University, Dhaka-1100
E-mail: atique@stat.jnu.ac.bd

What is STATA?

- STATA ("Software for Statistics and Data Science") is an integrated software package that provides all your data science needs data manipulation, visualization, statistics, and reproducible reporting.
- It is a multi-purpose statistical package to help you explore, summarize and analyse **datasets**.
- **A dataset** is a collection of several pieces of information called variables (usually arranged by columns). A variable can have one or several values (information for one or several cases).
- Like other statistical packages e.g., SPSS, SAS and R.

What is STATA?

- STATA is a general-purpose statistical software package created in 1985 by StataCorp.
- Most of its users work in research, especially in the fields of economics, sociology, political science, biomedicine and epidemiology.
- STATA's capabilities include data management, statistical analysis, graphics, simulations, regression analysis, and custom programming.

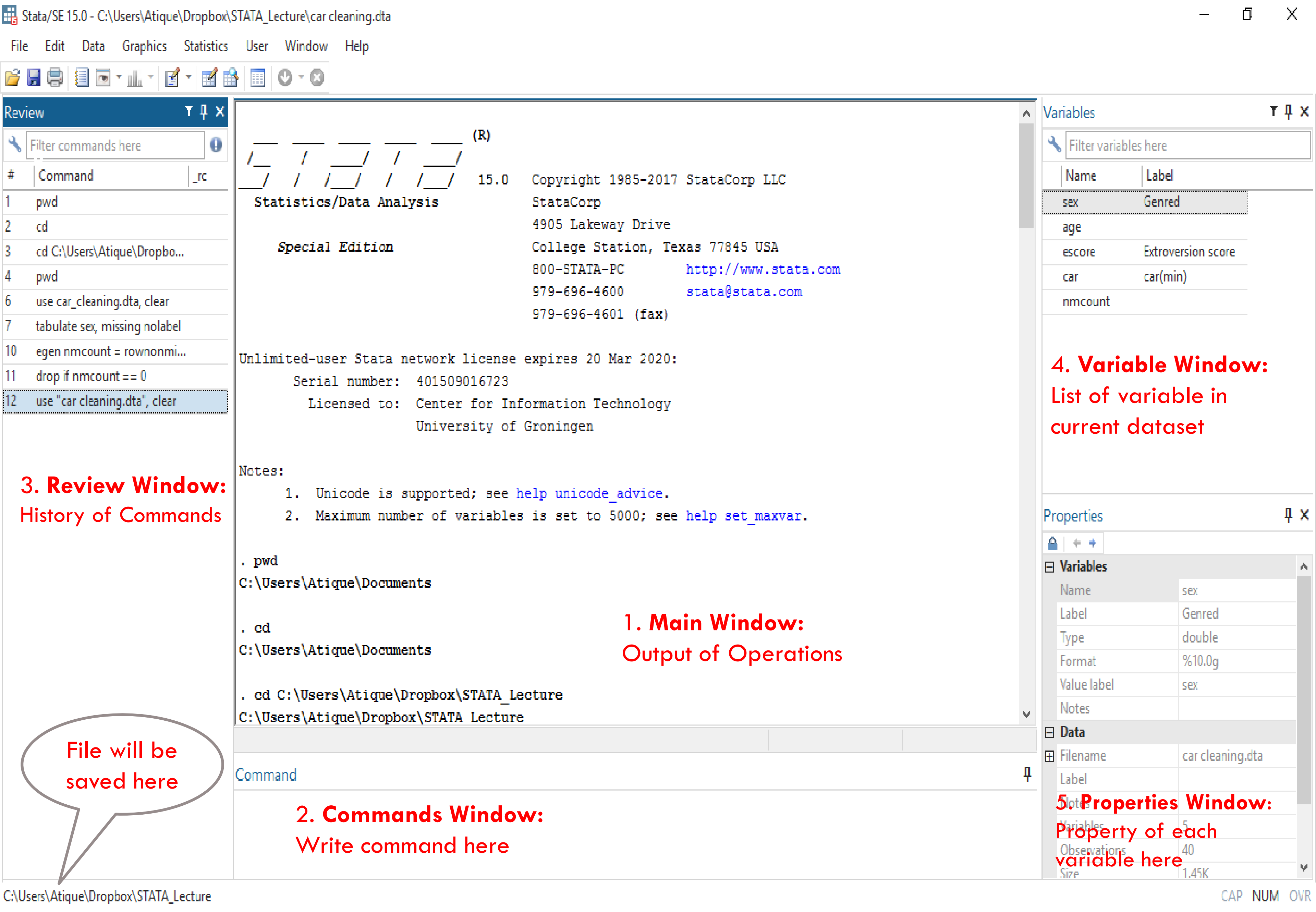
STATA Version

- There are four major builds of each version of Stata:
- **Stata/IC** (the standard version) - limited to 2,047 variables.
- **Stata/SE** (an extended version)- for large databases (5,000, variables)
- **Stata/MP** (for multiprocessing)- for large databases (>5,000 variables), and
- **Small Stata** - a smaller, student version for educational purchase only.
- The major difference between the versions is the number of variables allowed in memory.

STATA vs. SPSS vs. SAS vs. R vs. Python

Features	SPSS	SAS	Stata	JMP (SAS)	R	Python (Pandas)
Learning curve	Gradual	Pretty steep	Gradual	Gradual	Pretty steep	Steep
User interface	Point-and-click	Programming	Programming/ point-and-click	Point-and-click	Programming	Programming
Data manipulation	Strong	Very strong	Strong	Strong	Very strong	Strong
Data analysis	Very strong	Very strong	Very strong	Strong	Very strong	Strong
Graphics	Good	Good	Very good	Very good	Excellent	Good
Cost	Expensive (perpetual, cost only with new version). Student disc.	Expensive (yearly renewal) Free student version, 2014	Affordable (perpetual, cost only with new version). Student disc.	Expensive (yearly renewal) Student disc.	Open source (free)	Open source (free)
Released	1968	1972	1985	1989	1995	2008

STATA Display Manager


Stata/SE 15.0 - C:\Users\Atique\Dropbox\STATA_Lecture\car cleaning.dta
File Edit Data Graphics Statistics User Window Help
Review
Filter commands here
Variables
Filter variables here
Properties
Command
C:\Users\Atique\Dropbox\STATA_Lecture
CAP NUM OVR

#	Command	_rc
1	pwd	
2	cd	
3	cd C:\Users\Atique\Dropbo...	
4	pwd	
6	use car_cleaning.dta, clear	
7	tabulate sex, missing nolabel	
10	egen nmcount = rownonmi...	
11	drop if nmcount == 0	
12	use "car cleaning.dta", clear	

3. Review Window:
History of Commands

```
  ___  ____  ____  ____  ____ (R)
 /__    /   ____/   /   ____/
___/   /   /___/   /   /___/   15.0   Copyright 1985-2017 StataCorp LLC
  Statistics/Data Analysis            StataCorp
                                      4905 Lakeway Drive
     Special Edition                  College Station, Texas 77845 USA
                                      800-STATA-PC        http://www.stata.com
                                      979-696-4600        stata@stata.com
                                      979-696-4601 (fax)

Unlimited-user Stata network license expires 20 Mar 2020:
       Serial number:  401509016723
         Licensed to:  Center for Information Technology
                       University of Groningen

Notes:
      1.  Unicode is supported; see help unicode_advice.
      2.  Maximum number of variables is set to 5000; see help set_maxvar.

. pwd
C:\Users\Atique\Documents

. cd
C:\Users\Atique\Documents

. cd C:\Users\Atique\Dropbox\STATA_Lecture
C:\Users\Atique\Dropbox\STATA_Lecture
```

1. Main Window:
Output of Operations

2. Commands Window:
Write command here

Name	Label
sex	Genred
age	
escore	Extroversion score
car	car(min)
nmcount	

4. Variable Window:
List of variable in current dataset

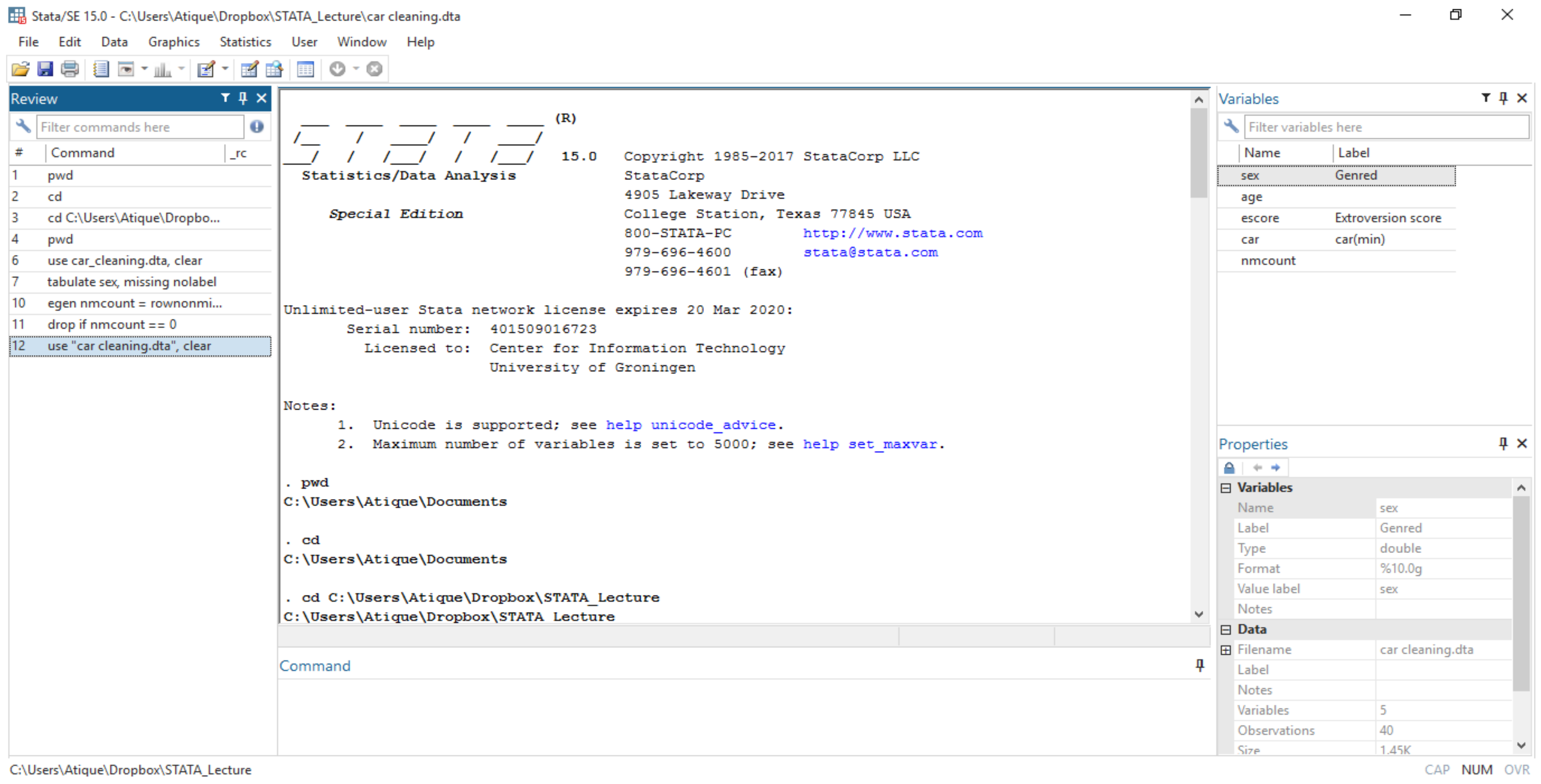
Variables	
Name	sex
Label	Genred
Type	double
Format	%10.0g
Value label	sex
Notes	
Data	
Filename	car cleaning.dta
Label	
Notes	
Variables	5
Observations	40
Size	1.45K

5. Properties Window:
Property of each variable here

File will be saved here

Working Directory

- To see your Working Directory, Type: pwd
- To change Working Directory, Type: cd "C:\folder"


Stata/SE 15.0 - C:\Users\Atique\Dropbox\STATA_Lecture\car cleaning.dta
File Edit Data Graphics Statistics User Window Help
Review
Filter commands here
Variables
Filter variables here
Properties
Command
CAP NUM OVR

#	Command	_rc
1	pwd	
2	cd	
3	cd C:\Users\Atique\Dropbo...	
4	pwd	
6	use car_cleaning.dta, clear	
7	tabulate sex, missing nolabel	
10	egen nmcount = rownonmi...	
11	drop if nmcount == 0	
12	use "car cleaning.dta", clear	

```
  ___  ____  ____  ____  ____ (R)
 /__    /   ____/   /   ____/
___/   /   /___/   /   /___/   15.0   Copyright 1985-2017 StataCorp LLC
  Statistics/Data Analysis            StataCorp
                                      4905 Lakeway Drive
     Special Edition                  College Station, Texas 77845 USA
                                      800-STATA-PC        http://www.stata.com
                                      979-696-4600        stata@stata.com
                                      979-696-4601 (fax)

Unlimited-user Stata network license expires 20 Mar 2020:
       Serial number:  401509016723
         Licensed to:  Center for Information Technology
                       University of Groningen

Notes:
      1.  Unicode is supported; see help unicode_advice.
      2.  Maximum number of variables is set to 5000; see help set_maxvar.

. pwd
C:\Users\Atique\Documents

. cd
C:\Users\Atique\Documents

. cd C:\Users\Atique\Dropbox\STATA_Lecture
C:\Users\Atique\Dropbox\STATA_Lecture
```

Name	Label
sex	Genred
age	
escore	Extroversion score
car	car(min)
nmcount	

Variables	
Name	sex
Label	Genred
Type	double
Format	%10.0g
Value label	sex
Notes	
Data	
Filename	car cleaning.dta
Label	
Notes	
Variables	5
Observations	40
Size	1.45K

C:\Users\Atique\Dropbox\STATA_Lecture

Components of Stata Program

A. .dta Files

i) This is the file containing your Stata format dataset.

ii) Stata also reads non- .dta formats with infile or insheet commands.

B. .do Files

i) These files are your programs. SPSS er syntax file er moto

ii) Write programs in Wordpad/Notepad/**Notepad++**, or the Stata program editor. Just be sure to save your files with the .do extension.

iii) If you have a .do file written, you can run it by typing the following in the Stata command window: do filename.do, clear

C. .log Files

i) This is your output file.

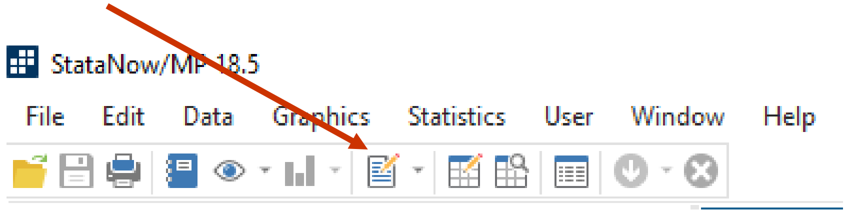
ii) At the beginning of your .do file, type log using filename.log, replace
"replace" tells Stata to write over the previous log while you run the program

iii) At the end of the .do file, type log close

iv) You can open your .log file with Wordpad/Notepad/**Notepad++**.

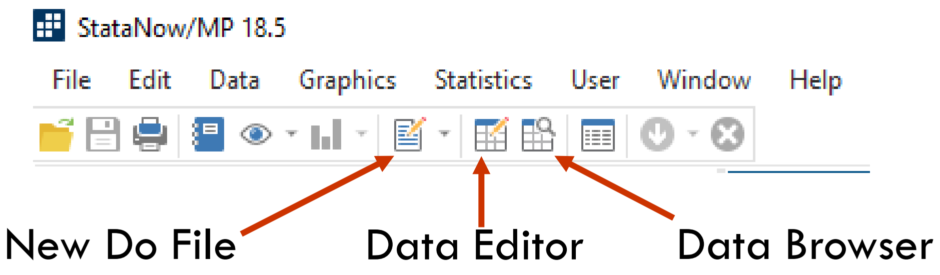
Command window and do-file

- **Command Window:** Here we type all the instructions (commands) that we ask Stata to execute.
- Alternatively, a set of lines of commands containing instructions can be executed from a text file, called a do-file. Stata has a built-in do file creator, the Do-Editor Window. Files created from the Do-editor (these are called Stata programs) have the extension .do at the end of the filename.
- You can use any word processor and save the file in ASCII format, or you can use Stata's 'do-file editor' with the advantage that you can run the commands from there. Either in the command window type: doedit or click here



.do-file

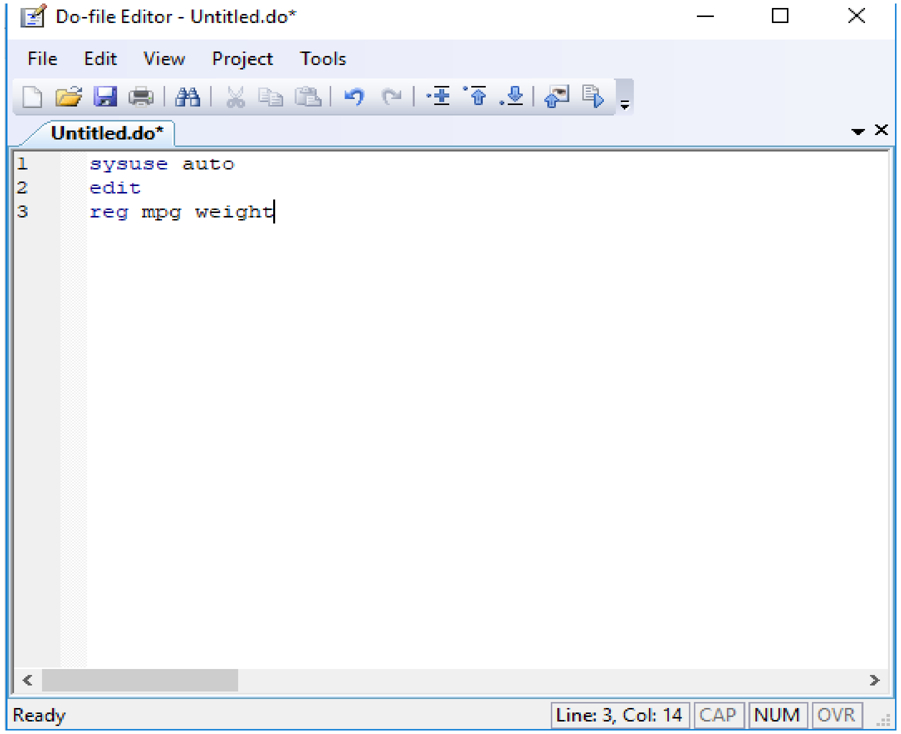
- A do file is a set of Stata Commands typed in a plain text file.
- How to open a new Do file
  - Press **Ctrl+9** (not numpad) to open a Do file.
  - Type **doedit** in the command window to open a Do file.
  - Alternatively, you can open it from the Stata Window.



- To run a Stata command in the do file, select the command lines and press **Ctrl+D**. Or, press the execute button.
- Always use a do file.

Command window and do-file

- Do-files are almost similar to SPSS's SYNTAX window.
- Do-files are useful if we want to keep our command/code for further use or recreation.



```
sysuse auto
edit
reg mpg weight
```

Variables in Stata

- **There are six variable types supported by STATA.**
- **Int:** Can take integer type data only. Takes four byte space memory.
- **Long:** It also takes an integer but larger than int, takes 8 bytes in memory.
- **Byte:** Byte is a smaller version of int, which can take numerical values from 1-100 and takes very low memory.
- **Float:** Float type variables can take floating-point numbers (decimal up to 8 digits).
- **Double:** Similar to float, but can take larger data than float.
- **str:** Can take character(s) type data and a sequence of characters called a String.

Opening / saving Stata files (*.dta)

- To open files already in Stata with extension *.dta, run Stata, and you can do either:
  - Go to file->open in the menu, or
  - Type use “c:\mydata\mydatafile.dta”
  - If your working directory is already set to c:\folder, just type: use mydatafile
- To save a data file from Stata, go to file –> save as or just type: save, replace in the command window
- If the dataset is new or just imported from another format, go to file –> save as or just type:

  save mydatafile /*Pick a name for your file*/

Import: Excel to Stata

- If you have a file in Excel format you can use

  import excel “c:\mydata.xlsx”, sheet(“Sheet1”) firstrow clear

  ```
  import excel "C:\Users\Atique\Dropbox\STATA_Lecture\data1.xlsx",sheet("Sheet1") firstrow clear
  ```

- If you have a file in csv format you can use

  import delimited “c:\mydata.csv", clear

  ```
  import delimited "C:\Users\Atique\Dropbox\STATA_Lecture\bodyfat.csv", clear
  ```

  OR

  insheet using “c:\mydata.csv", clear

  ```
  insheet using "C:\Users\Atique\Dropbox\STATA_Lecture\bodyfat.csv", clear
  ```

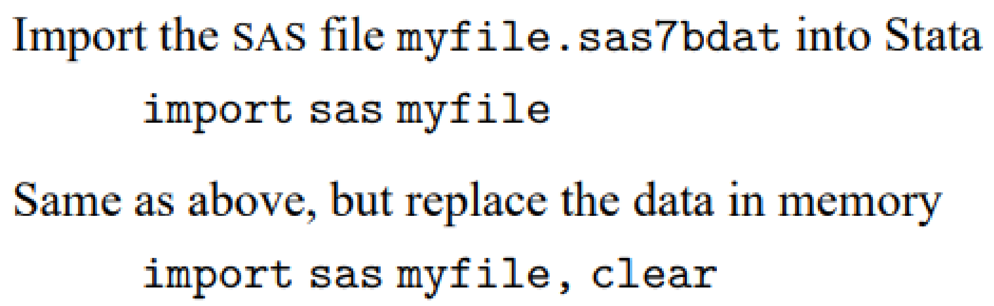
From SPSS/SAS to Stata (16+ Version)

- If your data is in SPSS format (*.sav) or SAS(*.sas7bcat).
- For SPSS, enter the command in the Command window

  import spss using "C:\path\to\your\filename.sav“

  import spss using "C:\Users\Atique\Dropbox\STATA_Lecture\Body Temperature.sav", clear
- To convert variable names to lowercase, use the case(lower) option:

  import spss using "C:\Users\Atique\Dropbox\STATA_Lecture\Body Temperature.sav", case(lower)
- For SAS, enter the command in the Command window



Import the SAS file `myfile.sas7bdat` into Stata

```
        import sas myfile
```

Same as above, but replace the data in memory

```
        import sas myfile, clear
```

From SPSS/SAS to Stata (Older Version)

- If your data is in SPSS format (*.sav) or SAS(*.sas7bcat).
- For SPSS and SAS, you may need to install them by typing

```
ssc install usespss

ssc install usesas
```

- Once installed, just type

```
usespss using “c:\mydata.sav”

usesas using “c:\mydata.sas7bcat”
```

Input Dataset

- A Stata program that begins with an input statement and typically creates a Stata data set or a report

```
input IDNUMBER str20(NAME  TEAM) STARTWEIGHT ENDWEIGHT
1023 DAVID "SHAW RED" 189 165
1023 DAVID "SHAW RED" 189 165
1049 AMELIA "SERRANO YELLOW" 145 124
1219 ALAN "NANCE RED" 210 192
1246 RAVI "SINHA YELLOW" 194 177
1078 ASHLEY "MCKNIGHT RED" 127 118
end
list
```

Creating New Variable

- **gen (generate):** Primarily used for creating new variables based on simple transformations or calculations of existing variables, often on an observation-by-observation basis.

  ```
  gen loss = STARTWEIGHT - ENDWEIGHT
  ```

- **egen (extensions to generate):** Designed for more advanced operations, particularly those involving group-level calculations, summary statistics, or the creation of variables based on patterns and relationships across observations.

  ```
  egen mean_weight_loss = mean(loss), by(TEAM)
  ```

  //Calculates mean weight loss for each Team//

RECODE

- **Recode Command:** With recode you specify a list of rules in the form (*old values* = *new value*). The old values can be a single number, a list of numbers, or a range of numbers which you describe with *start*/*end*:

  ```
  recode loss (0/17=0) (18/25=1) , gen(loss_cat)
  ```

- The gen option tells recode to create a new variable (loss_cat) to store the results. If you don't include a gen option, recode will change the original variable.

Labels

- **Variable labels** can tell you more about the variable itself

```
label variable loss_cat "loss of weight"
```

- **Value labels** tell you what the individual values of the variable mean. To set them, you first define the labels and then apply them to a variable:

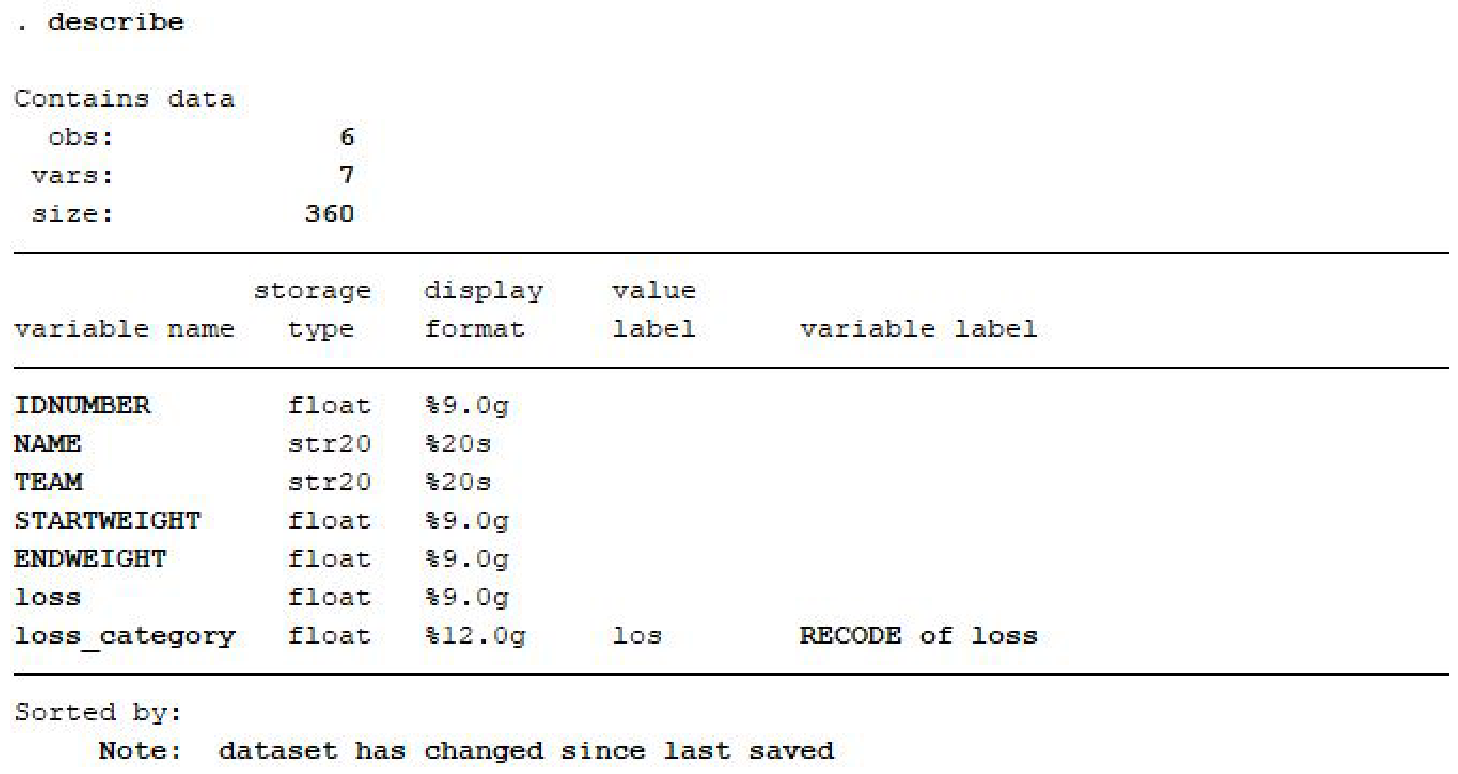
```
label define los 0 "less than 17" 1 "more than 17"
label values loss_cat los
```

- Rename command changing meaningless variable names

```
rename loss_cat loss_category
```

Describe Command

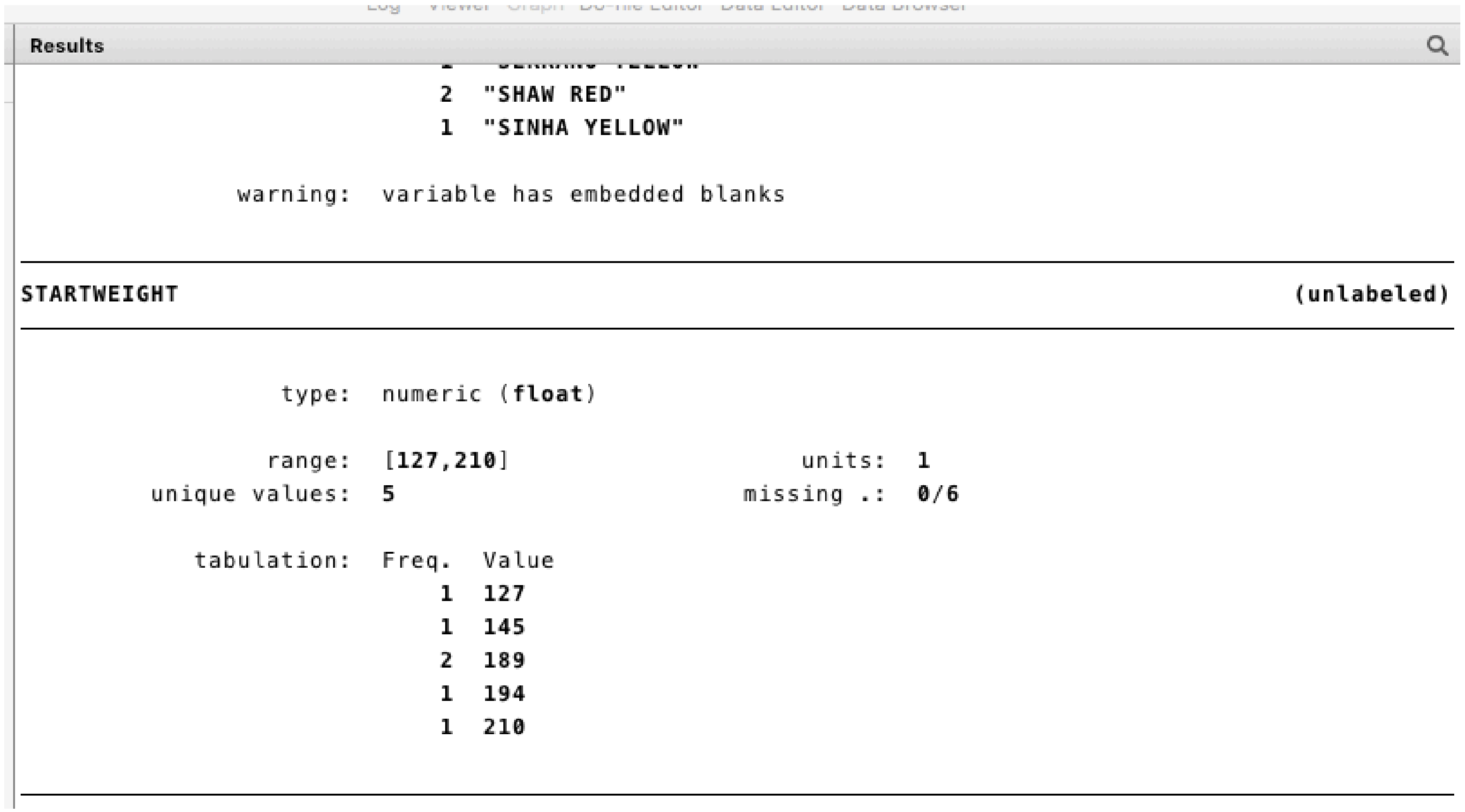
- To get a general description of the dataset and the format for each variable type describe



```
. describe

Contains data
  obs:             6
 vars:             7
 size:           360
--------------------------------------------------------------------------------
              storage  display    value
variable name   type   format     label      variable label
--------------------------------------------------------------------------------
IDNUMBER        float  %9.0g
NAME            str20  %20s
TEAM            str20  %20s
STARTWEIGHT     float  %9.0g
ENDWEIGHT       float  %9.0g
loss            float  %9.0g
loss_category   float  %12.0g     los        RECODE of loss
--------------------------------------------------------------------------------
Sorted by:
     Note:  dataset has changed since last saved
```

codebook Command

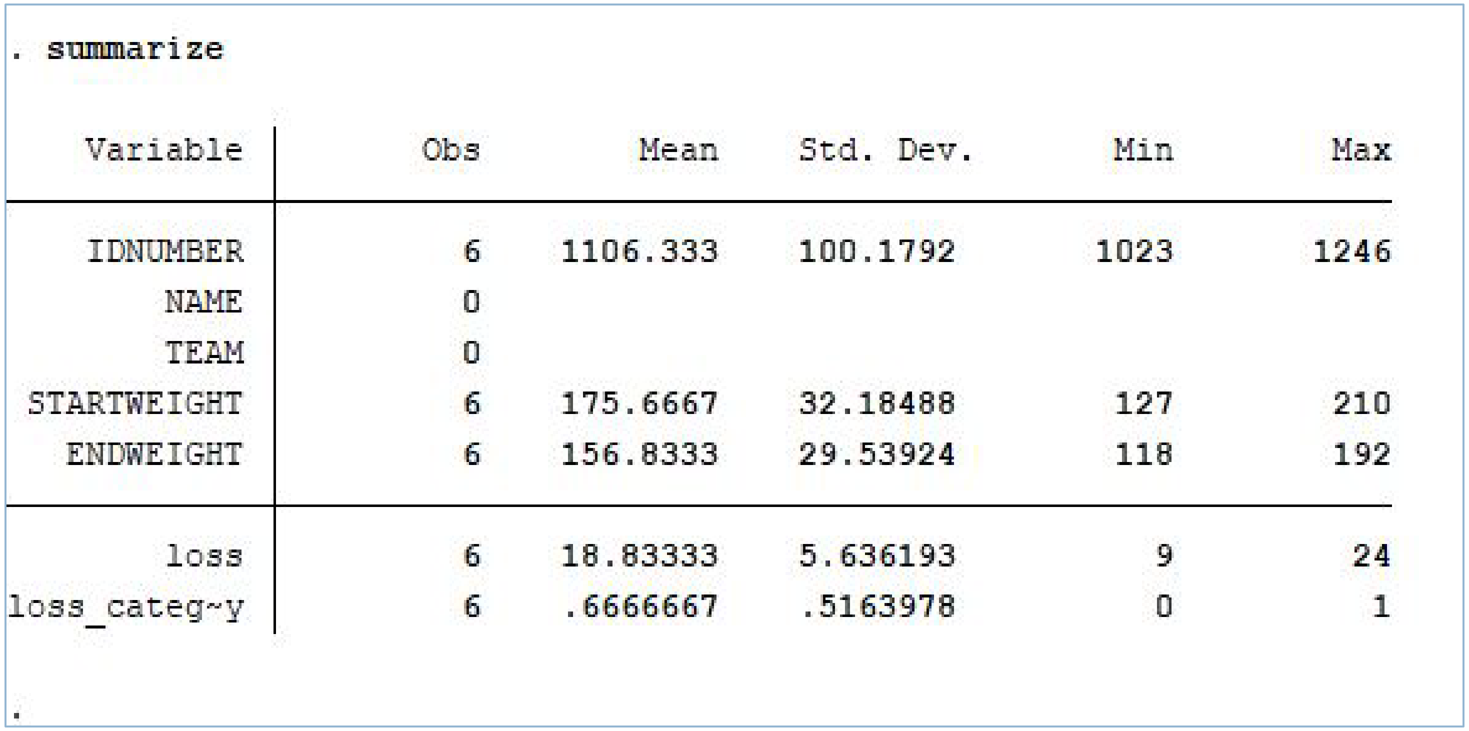
- Codebook examines the variable names, labels and data to produce a codebook describing the dataset.



```
                                2  "SHAW RED"
                                1  "SINHA YELLOW"

              warning:  variable has embedded blanks

------------------------------------------------------------------------------
STARTWEIGHT                                                       (unlabeled)
------------------------------------------------------------------------------

                 type:  numeric (float)

                range:  [127,210]                    units:  1
        unique values:  5                        missing .:  0/6

           tabulation:  Freq.  Value
                            1  127
                            1  145
                            2  189
                            1  194
                            1  210

------------------------------------------------------------------------------
```

summarize Command

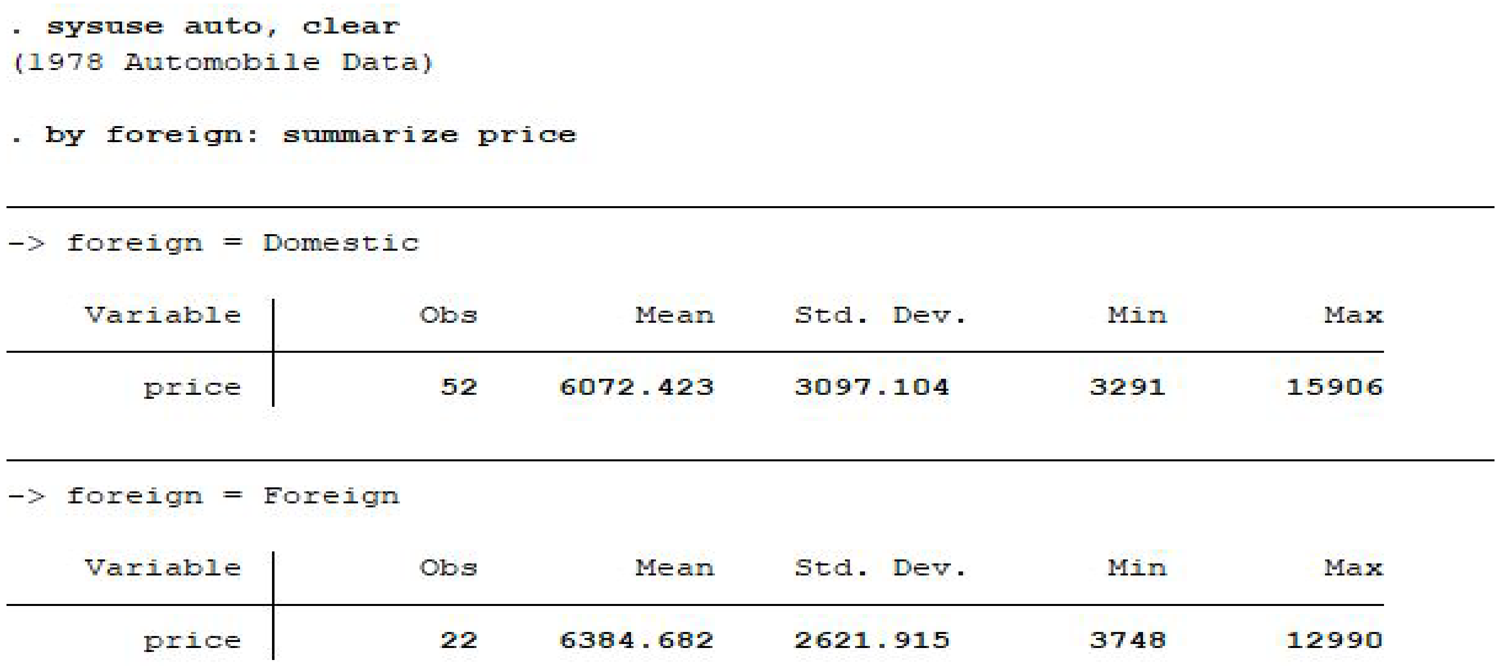
- Type summarize to get some basic descriptive statistics.



```
. summarize

    Variable |       Obs        Mean    Std. Dev.       Min        Max
-------------+--------------------------------------------------------
    IDNUMBER |         6    1106.333    100.1792       1023       1246
        NAME |         0
        TEAM |         0
 STARTWEIGHT |         6    175.6667    32.18488        127        210
   ENDWEIGHT |         6    156.8333    29.53924        118        192
-------------+--------------------------------------------------------
        loss |         6    18.83333    5.636193          9         24
loss_categ~y |         6    .6666667    .5163978          0          1

.
```

Using by processing

- Suppose we want to get the descriptive statistics for price by car type (foreign vs domestic). We can use what is called by processing.



```
. sysuse auto, clear
(1978 Automobile Data)

. by foreign: summarize price

---------------------------------------------------------------------------
-> foreign = Domestic

    Variable |       Obs        Mean    Std. Dev.       Min        Max
-------------+--------------------------------------------------------
       price |        52    6072.423    3097.104       3291      15906

---------------------------------------------------------------------------
-> foreign = Foreign

    Variable |       Obs        Mean    Std. Dev.       Min        Max
-------------+--------------------------------------------------------
       price |        22    6384.682    2621.915       3748      12990
```

sort variable

- One way to sort data is using a simple sort command followed by the variable name. Stata will sort the data in ascending order by default.

  sysuse auto, clear

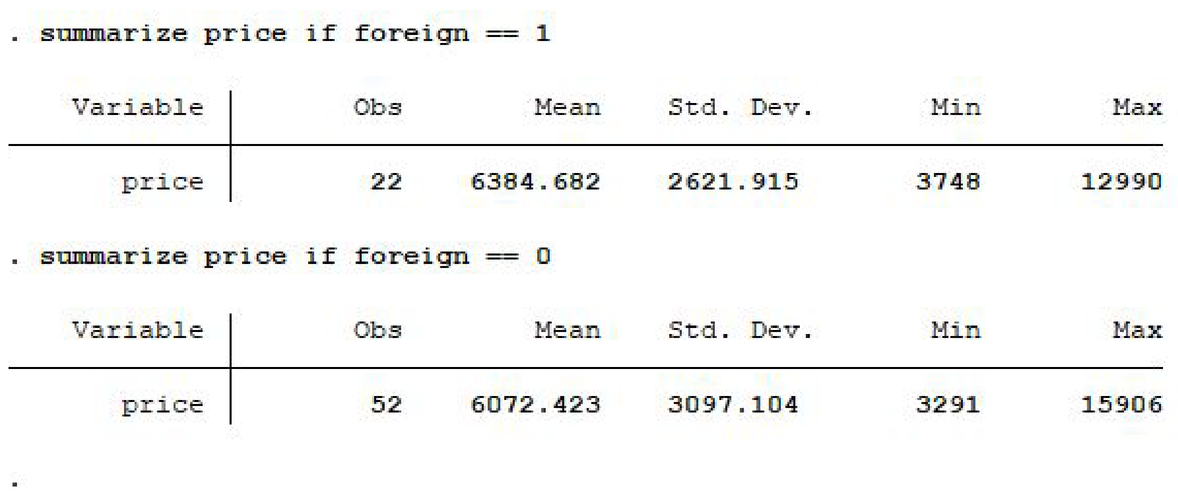
  sort mpg

- After we sort the data, we can then use the standard by mpg: command.
- In by processing, we can also sort the data and execute the by command at the same time using the bysort Command

  bysort mpg: summarize price

Using if

- The by statement will give us descriptives for all levels of the by variable (i.e., both foreign and domestic).
- Suppose we just want the describes for one level of the by variable. We can use the if statement for that. For foreign cars (i.e., foreign ==1):



```
. summarize price if foreign == 1

    Variable |       Obs        Mean    Std. Dev.       Min        Max
-------------+--------------------------------------------------------
       price |        22    6384.682    2621.915       3748      12990

. summarize price if foreign == 0

    Variable |       Obs        Mean    Std. Dev.       Min        Max
-------------+--------------------------------------------------------
       price |        52    6072.423    3097.104       3291      15906

.
```

Symbol

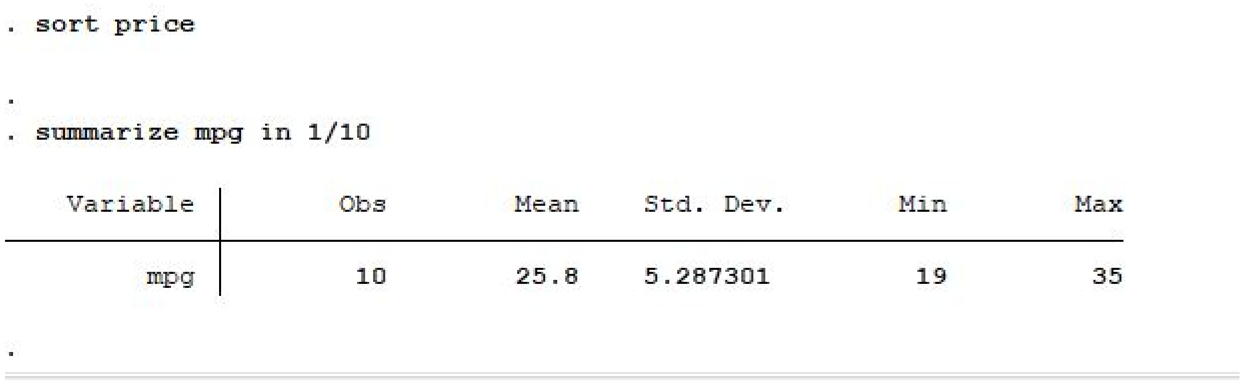
Symbol	Meaning
==	is or is equal to
!= or ~=	is not or is not equal to
>	is greater than
>=	is greater than or equal to
<	is less than
<=	is less than or equal to

Using in

- The in qualifier specifies a particular subset of cases based on their order in the dataset. For example, if we want to examine the mpg in the 10 least expensive cars, we would use the in command.

  sort price

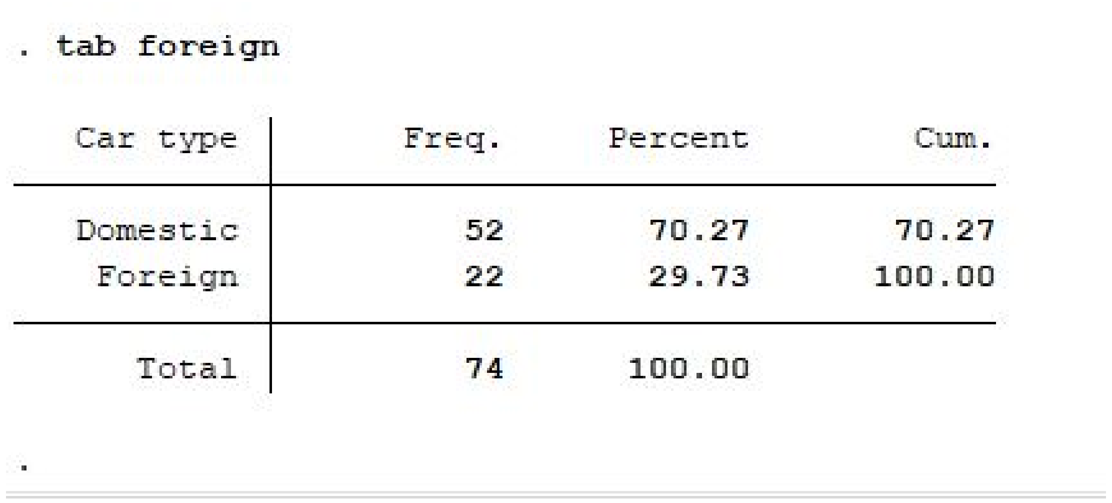
  summarize mpg in 1/10



```
. sort price

.
. summarize mpg in 1/10

    Variable |       Obs        Mean    Std. Dev.       Min        Max
-------------+--------------------------------------------------------
         mpg |        10        25.8    5.287301         19         35

.
```

Exploring data: frequency

- Frequencies are used to analyze categorical data. The tables below are frequency tables, values are in ascending order. In Stata use the command tab varname.



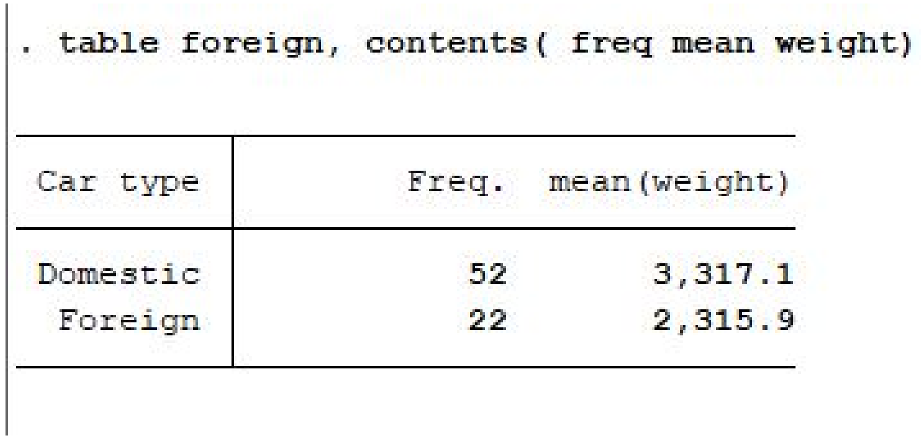
```
. tab foreign

   Car type |      Freq.     Percent        Cum.
------------+-----------------------------------
   Domestic |         52       70.27       70.27
    Foreign |         22       29.73      100.00
------------+-----------------------------------
      Total |         74      100.00

.
```

- Type help tab for more details.

Descriptive Statistics (using table)

- Command table produces frequencies and descriptive statistics per category. For more info and a list of all statistics type help table. Here are some examples, type

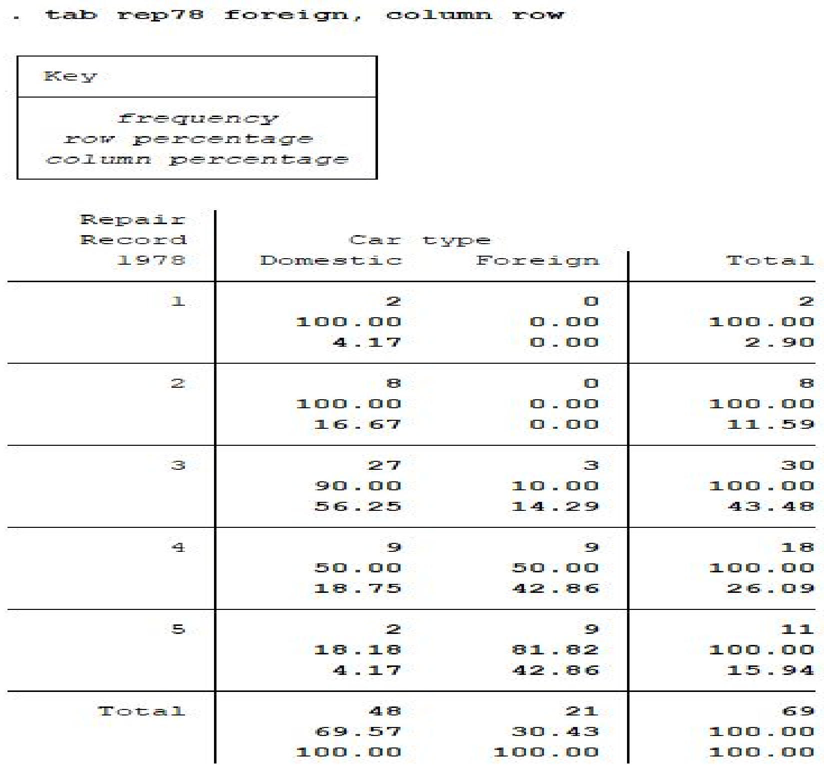
table loss_category, contents(freq mean loss)



```
. table foreign, contents( freq mean weight)

----------------------------------------
  Car type |      Freq.   mean(weight)
-----------+----------------------------
  Domestic |         52        3,317.1
   Foreign |         22        2,315.9
----------------------------------------
```

Exploring data: crosstabs

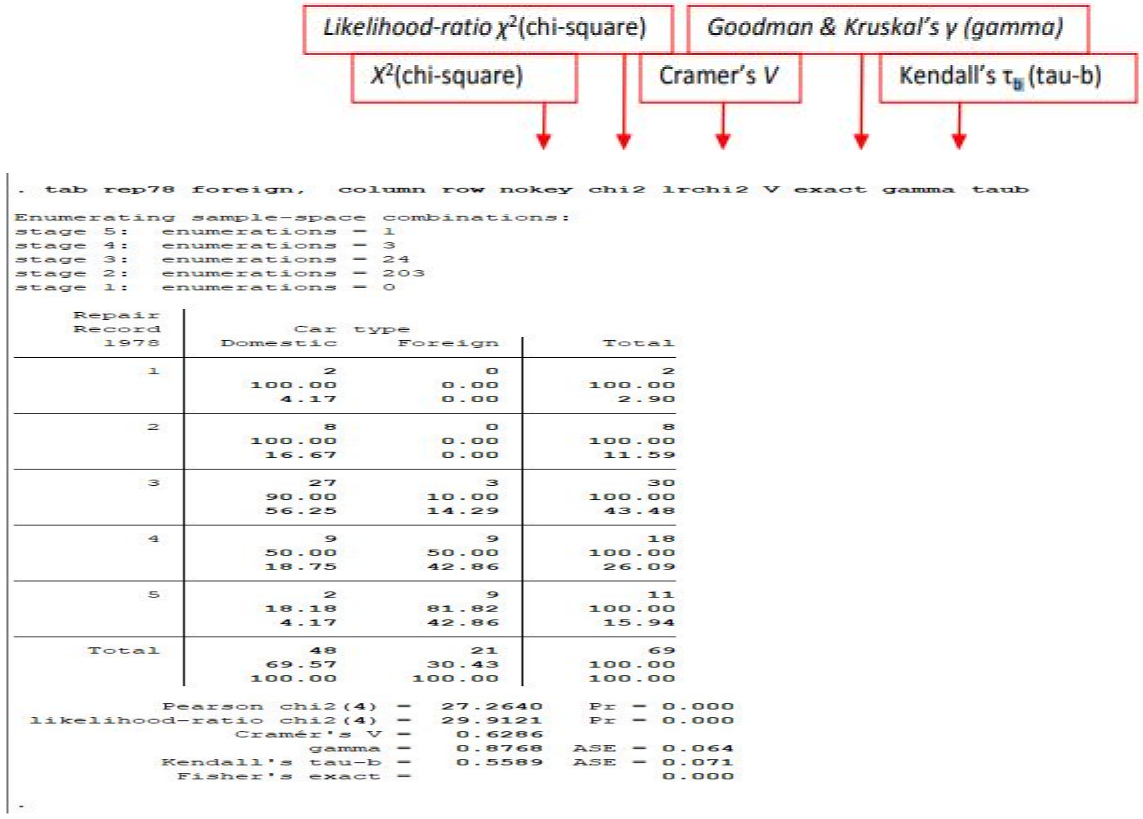
- Also known as contingency tables, crosstabs help you to analyze the relationship between two or more categorical variables. Below is a crosstab between the variable ‘ecostatu’ and ‘gender’. We use the command tab var1 var2



```
. tab rep78 foreign, column row

+-------------------+
| Key               |
|-------------------|
|     frequency     |
|   row percentage  |
| column percentage |
+-------------------+

    Repair |
    Record |       Car type
      1978 |  Domestic    Foreign |     Total
-----------+----------------------+----------
         1 |         2          0 |         2
           |    100.00       0.00 |    100.00
           |      4.17       0.00 |      2.90
-----------+----------------------+----------
         2 |         8          0 |         8
           |    100.00       0.00 |    100.00
           |     16.67       0.00 |     11.59
-----------+----------------------+----------
         3 |        27          3 |        30
           |     90.00      10.00 |    100.00
           |     56.25      14.29 |     43.48
-----------+----------------------+----------
         4 |         9          9 |        18
           |     50.00      50.00 |    100.00
           |     18.75      42.86 |     26.09
-----------+----------------------+----------
         5 |         2          9 |        11
           |     18.18      81.82 |    100.00
           |      4.17      42.86 |     15.94
-----------+----------------------+----------
     Total |        48         21 |        69
           |     69.57      30.43 |    100.00
           |    100.00     100.00 |    100.00
```

Crosstabs (Test for Associations)

- To see whether there is a relationship between two variables you can choose a number of tests. Some apply to nominal variables some others to ordinal. I am running all of them here for presentation purposes.


Likelihood-ratio χ^2(chi-square)
Goodman & Kruskal's γ (gamma)
X^2(chi-square)
Cramer's V
Kendall's τ_b (tau-b)

```
. tab rep78 foreign,  column row nokey chi2 lrchi2 V exact gamma taub

Enumerating sample-space combinations:
stage 5:  enumerations = 1
stage 4:  enumerations = 3
stage 3:  enumerations = 24
stage 2:  enumerations = 203
stage 1:  enumerations = 0

    Repair |
    Record |       Car type
      1978 |  Domestic    Foreign |     Total
-----------+----------------------+----------
         1 |         2          0 |         2
           |    100.00       0.00 |    100.00
           |      4.17       0.00 |      2.90
-----------+----------------------+----------
         2 |         8          0 |         8
           |    100.00       0.00 |    100.00
           |     16.67       0.00 |     11.59
-----------+----------------------+----------
         3 |        27          3 |        30
           |     90.00      10.00 |    100.00
           |     56.25      14.29 |     43.48
-----------+----------------------+----------
         4 |         9          9 |        18
           |     50.00      50.00 |    100.00
           |     18.75      42.86 |     26.09
-----------+----------------------+----------
         5 |         2          9 |        11
           |     18.18      81.82 |    100.00
           |      4.17      42.86 |     15.94
-----------+----------------------+----------
     Total |        48         21 |        69
           |     69.57      30.43 |    100.00
           |    100.00     100.00 |    100.00

          Pearson chi2(4) =  27.2640   Pr = 0.000
 likelihood-ratio chi2(4) =  29.9121   Pr = 0.000
               Cramér's V =   0.6286
                    gamma =   0.8768  ASE = 0.064
          Kendall's tau-b =   0.5589  ASE = 0.071
           Fisher's exact =                 0.000

.
```

Deleting cases (selectively)

- You can drop cases selectively using the conditional "if", for example
- drop if var1==1 /*This will drop observations (rows) where gender =1*/
- drop if turn>40 /*This will drop observation where turn>40*/
- Alternatively, you can keep the options you want

```
keep if var1==1
keep if turn <40
keep if rep78==1 | country==4
```

| = "or", & = "and"