```
/*******************************************************************************************************
Title: STATA Tutorial 2
*** Purpose: Stata setup, basic data manipulation, Mean comparision test, Basic OLS regressions ***
*******************************************************************************************************
/
clear all

******************
****General setup**
******************
capture log close /* close the log file if any already open. capture: This command tells Stata to
ignore any error messages and keep going*/

/*A macro is used as shorthand: you type a short macro name but are actually referring to some
numerical value or a string of characters. Macros are of two types: local and global
1 local: work within the program or do-file in which they are created
2 global: work in all programs and do files
*/

global computer "/Users/huihuaxie"

// if you wanna invoke, use $ as a pre-fix
global dropbox "$computer/Dropbox"
global datapath "$computer/Data"
global projectpath "$dropbox/Lectures/ECO3211_2021Fall/Tutorials/Tutorial 1"

global rawdata "$datapath/Lectures"
global cleandata "$projectpath/Data"

global dopath "$projectpath/Do"
global result "$projectpath/Results"

cd "$cleandata"

*********************************
***Put your notes in do-file *****
*********************************

use hlth hi age age2 angrist brooks educ empl famsize inc1 inc2 inc3 inc4 inc5 inc6 inc7 inc8
marstat nwhite sex perweight using "$rawdata/NHIS2009_tutorial.dta", clear
/*use varlist using filename */

/*
/*It is good practice to keep extensive notes within your do-file
Thus, when you look back over it you know what you were trying to achieve with each command or
set of commands
You can insert comments in several different ways */

// Stata will ignore a line if it starts with two consecutive forward slashes "//"

// cd "D:/Dropbox/lectures/ECO3211_2021Fall/STATA/Stata 1"

/*cd "D:/Dropbox/lectures/ECO3211_2021Fall/STATA/Stata 1" */

// blocking a whole set of commands

/*
rename sex gender  /* example of renaming the variable*/

drop marstat /*drop a variable*/

keep if _n<=50 /*keep observations1-50 for all variables*/
*/

* You can use three consecutive slashes "///" which will result in the rest of the line being
ignored and the next line added at the end of the current line. This comment is useful for
splitting a very long line of code over several lines.
It works in do-file window, but not in command window.
```

```
*/

reg hlth hi age age2 angrist brooks educ empl famsize inc1 inc2 inc3 ///
inc4 inc5 inc6 inc7 inc8 marstat nwhite sex, robust

sum hlth if hi==0
local controlmean=r(mean)

outreg2 using Table1, excel dec(4) addstat(controlmean, `controlmean') replace //arrange regression results into an excel table. "dec(4)" is used to fix deceimals, "addstat" is used to add additional statistics
* An additional row or column?

* Three ways to install a new package
net search outreg2 //1. search Internet for installable packages
ssc install outreg2, all replace //2. install or uninstall packages from the Boston College Statistical Software Components (SSC) archive
help outreg2 //3. displays useful information about the how to use specified command or specific topic

***************
**** Macros ****
***************
// firstly define Lcontrols and Gcontrols.
local Lcontrols age age2 educ empl famsize nwhite sex
global Gcontrols age age2 educ empl famsize nwhite sex

reg hlth hi `Lcontrols', robust //symbol` is located next to number 1

*Compare reg hlth hi `lcontrols', ro
// Stata is a case-sensitive, sensitive to Upper and lower cases.


outreg2 using Table1, excel dec(4) addstat(controlmean, `controlmean')
/*
. local controls age age2 educ empl famsize nwhite sex

. reg hlth hi `Lcontrols', robust

Linear regression                               Number of obs     =     18,790
                                                F(8, 18781)       =     228.29
                                                Prob > F          =     0.0000
                                                R-squared         =     0.0984
                                                Root MSE          =     .90476

------------------------------------------------------------------------------
             |               Robust
        hlth |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
          hi |   .1424578   .0214263     6.65   0.000     .1004604    .1844553
         age |  -.0151922    .007768    -1.96   0.051    -.0304182    .0000339
        age2 |  -.0000198   .0000918    -0.22   0.829    -.0001998    .0001601
        educ |   .0589874   .0021703    27.18   0.000     .0547333    .0632415
        empl |   .2437649   .0204933    11.89   0.000     .2035961    .2839336
     famsize |   .0093462   .0054408     1.72   0.086    -.0013182    .0200105
      nwhite |  -.1781165    .017185   -10.36   0.000    -.2118007   -.1444324
         sex |   .0010022   .0137301     0.07   0.942    -.0259101    .0279145
       _cons |    3.35254   .1602921    20.92   0.000     3.038352    3.666727
------------------------------------------------------------------------------

. outreg2 using Table1, excel dec(4) addstat(controlmean, `controlmean')
*/

sum hlth if hi==0
local controlmean=r(mean)

outreg2 using Table1, excel dec(4) addstat(controlmean, `controlmean')

```

```
/*Local macros are ”private”
 If you use several programs within a single do-file, you need not worry about whether some other program has been using local macros with the same names*/
reg hlth hi `Lcontrols', robust

/* See result below
. reg hlth hi `Lcontrols', robust

Linear regression                               Number of obs     =     18,790
                                                F(1, 18788)       =     268.90
                                                Prob > F          =     0.0000
                                                R-squared         =     0.0159
                                                Root MSE          =     .94507

------------------------------------------------------------------------------
             |               Robust
        hlth |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
          hi |   .3288268   .0200527    16.40   0.000     .2895217    .3681318
       _cons |   3.655683    .018636   196.16   0.000     3.619154    3.692211
------------------------------------------------------------------------------
*/

// Global macros are ”public”
/*Gcontrols refers to exactly the same list of variables irrespective of the program that uses it, global macros are prefixed by the dollar sign: $
You should refrain from using global macros when a local macro suffices
This is good programming practice as it forces you to define these macro variables explicitly instead of defining them in some hard-to-find place in your code
If you use global macros you should make sure that you define them at the beginning of your code.
*/

reg hlth hi $Gcontrols, robust

sum hlth if hi==0
local controlmean=r(mean)
outreg2 using Table1, excel dec(4) addstat(controlmean, `controlmean')

/*. reg hlth hi $Gcontrols, robust

Linear regression                               Number of obs     =     18,790
                                                F(8, 18781)       =     228.29
                                                Prob > F          =     0.0000
                                                R-squared         =     0.0984
                                                Root MSE          =     .90476

------------------------------------------------------------------------------
             |               Robust
        hlth |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
          hi |   .1424578   .0214263     6.65   0.000     .1004604    .1844553
         age |  -.0151922    .007768    -1.96   0.051    -.0304182    .0000339
        age2 |  -.0000198   .0000918    -0.22   0.829    -.0001998    .0001601
        educ |   .0589874   .0021703    27.18   0.000     .0547333    .0632415
        empl |   .2437649   .0204933    11.89   0.000     .2035961    .2839336
     famsize |   .0093462   .0054408     1.72   0.086    -.0013182    .0200105
      nwhite |  -.1781165    .017185   -10.36   0.000    -.2118007   -.1444324
         sex |   .0010022   .0137301     0.07   0.942    -.0259101    .0279145
       _cons |    3.35254   .1602921    20.92   0.000     3.038352    3.666727
------------------------------------------------------------------------------
*/


**********************
***Organize data******
**********************
//compare egen & gen
egen inc_sexmean = mean(inc), by(sex)
```

```
*egen age_mean = mean(age), by(sex)  remember by followed with parentheses
//egen typically creates new variables based on summary measures, such as sum, mean, min and max. Use function mean to get mean income for each gender

egen educ_sexmax = max(educ), by(sex)

egen inc_count = count(inc)

egen inc_diff = diff(inc inc1) //An indicator. generate a variable indicating whether variables inc and inc1 are different or not

//sort the data by age first, generate the mean income for each age group
bysort age: egen inc_serial = mean(inc)
*by age, sort: egen inc_serial2 = mean(inc)

//numeric or string variables
/*
Stata stores or formats data in either of two ways-numeric or string. Numeric will store numbers while string will store text. Numeric variables are in black/blue color and string variables are in red color. String variale can also be used to store numbers, but you will not be able to perform numerical analysis on those numbers.
*/
tostring year, replace //change variable to the form of string. either replace or generate

destring year, gen(year1) //change variable from string to numeric
//at the same time, gen a new variable. parentheses

gen yr=substr(year, 3, 2)
* 2021 to 21
/*substr: Divide up a variable or to extract part of a variable to create a new one
The first term in parentheses is the string variable that you are extracting from
The second term (3) is the position of the first character you want to extract
The third term (2) is the number of characters to be extracted*/
// substr only works for string type not float type. If not tostring first, substr cannot be used.

gen yr2=substr(year,-2,2)
* 2021 to 21
/*Alternatively, you can select your starting character by counting from the end (2 positions from the end instead of 3 positions from the start)*/


collapse (mean) mean_inc=inc (max) max_inc=inc (count) count_inc=inc, by (age)
                                                                *(median) median_inc = inc
//Dangerous!! The change could be eternal.
* age is the first column
//collapse: This command converts the data into a dataset of summary statistics, such as sums, means, medians, and so on. And eternally leaves out all original information. This command is useful only if you want to aggregate dataset from individual level.
// definitely see the results. (browse)
// ctrl+shift+s ‘save to other’ to carefully save the result and protect the original data.

compress // Different var types take up different sizes of memories. compress attempts to reduce the amount of memory used by your data

/* Example:
compress
variable mpg was float now byte
variable price was long now int
variable yenprice was double now long
variable weight was double now int
variable make was str26 now str17
See help compress for detailed explanation
*/

save "$cleandata/temp.dta", replace
* here we invoke a global defined string again.

```