Boot camp in Computer Science: Introduction to STATA

Yao Thibaut Kpegli

Master 1 in Economics, ENS Paris-Saclay
2023–2024

Introduction

Purpose

- A quick and intensive reminder about STATA (4.5 hours)
  - to make you operational during your econometrics classes over the year

STATA: definition

STATA is a fee-based software to conduct statistical and econometrics analysis, optimization, simulation, numerical computation, etc...

Installation

Installation

- download the setup of STATA from the intranet of ENS Paris-Saclay
- Install the setup (use the informations of the license provided by ENS)

STATA as a simple calculator

Display and scalar

- sca a = 2
- sca b = 3
- sca c = - 4
- di a
- di b

Common operations

- addition : di 2+3 or di a+b
- subtraction: di 2-3 or di a-b
- product : di 2*3 or di a*b
- division: di 2/2 or di a/b
  - remark: STATA accepts shortened command name (true also for variable)

Common math functions

Common math functions

- square/exponent : di 2^3 or di a^b
- square root: di sqrt(2) or di di sqrt(a)
- exponential : exp(2) or di exp(a)
- logarithm : di log(2) or log(a)
- min: di min(2,3) or di min(a,b)
- max: di max(2,3) or di max(a,b)
- absolute val.: di abs(-4) or di abs(c)
- sign: di sign(-4) or di sign(c)
- round: di round(4.56789, 0.01)
- ceiling: di ceil(4.56789)
- integer: di int(4.56789)

Matrix operations

Matrix operations

$$A = \begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix}$$

- define matrix: matrix A = (1,2 \ 3,4)
- show the matrix: matrix list A
- extract element at line i=2 and column j=1 : di A[2,1]

Identy matrix

- mat I = I(2)
- mat li I

$$\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$$

Repetition matrix J(nrow,ncol,value)

- mat V1 = J(2,1,1)
- mat li V1

Matrix operations

Matrix operations

- transposition:
  - mat define B = A'
  - mat li B
- matrix product
  - mat define C = A*C
  - mat li C
- matrix addition
  - mat D = A+C
  - mat li D
- Substraction
  - mat E = A-C
  - mat li E
- division by a scalar
  - mat F= A/2
  - mat li F

Kronecker products

Kronecker product

$$A = \begin{pmatrix} 0.5 \\ 2 \end{pmatrix} \qquad \text{and} \qquad B = \begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix}$$

- Kronecker product
  - mat A = (0.5 \ 2)
  - mat B = (1,0 \ 1,1)
  - mat C = A # B
  - mat li C

$$\begin{pmatrix} 0.5 & 0 \\ 0.5 & 0.5 \\ 2 & 0 \\ 2 & 2 \end{pmatrix}$$

Matrix operations

$$A = \begin{pmatrix} 1 & 3 \\ 0 & 4 \end{pmatrix} \qquad \text{and} \qquad B = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$$

Row join

- mat A =(1,2 \ 3,4)
- mat B = I(2)
- mat C = (A,B)
- mat li C

Column join

- mat D = (A \ B)
- mat li D

Matrix operations

Useful functions

- inverse: inv()
  - mat inv = inv(A)
  - mat li inv
- diagonal vector: vecdiag()
  - mat diag = vecdiag(A)
  - mat li diag
- trace : trace()
  - di trace(A)
- determinant : det()
  - di det(A)

Exercise

- Create a matrix X that takes the values

$$\begin{pmatrix} 1 & 1 \\ 1 & 4 \\ 1 & 2 \\ 1 & 5 \end{pmatrix}$$

- Compute $B = (X'X)^{-1}$
  - determine the dimension
  - extract the elements on the diagonal
- Create a matrix Y that takes value

$$\begin{pmatrix} 3 \\ 6 \\ 4 \\ 7 \end{pmatrix}$$

- Compute $(X'X)^{-1}X'Y$

Result

- mat X = (1 4 2 5)
- mat X = J(4,1,1),X
- mat li X
- mat B = inv(X'*X)
- mat li B
- mat diag = vecdiag(B)
- mat li diag
- mat Y = (3 6 4 7)
- mat li Y
- mat beta = B*X'*Y
- mat li beta

Working Directory

Working Directory

- Useful to import/export data and results
  - set the work directory: cd "D:\ENS Paris Saclay\R2023_2024"

Definition

Definition

- data = matrix of data
  - set of vectors with the same length
  - placed next to each other vertically
- Column = Variable
  - possible of different types: quantitative (in black), characters (in red), dates (in black), numerical but also qualitative (in black).
  - Color "blue" means qualitative variables that are codified and labeled
- Line = Observation

Example

- upload the data "auto" integrated in STATA
  - clear all
  - sysuse auto
  - browse
- remark: index of row "_n" and total rows " _N "

CREATION

index of row	one	two	three
1	1	11	21
2	2	12	22
.	.	.	.
.	.	.	.
.	.	.	.
10	10	20	30

Creation with the hands

- clear all
- edit
- start entering directly your data

Creation

Creation with codes

- clear all
- set obs 10
- gen one = _n
- gen two = _n+10
- gen three = _n +20
- browse

Quick infos on the data

- describe content of data: describe
- more detailed description: codebook
- stat des: summarize
  - summarize one
  - stat des (option detail): summarize one, detail

BASIC DATA MANIPULATION

Useful symbols

- Strict inequality : `>`, `<`
- equal or inequal : `=<`, `>=`
- equal: `==`
- different equal: `!=`
- and : &
- or : |

RENAME VAR. AND REPLACE OBS.

- rename var.: rename ratio new
- replace obs: replace one = 3 if one == 5

BASIC DATA MANIPULATION

NEW VARIABLE

- creation of ratio=one/two: gen ratio = one/two
- dummy (binary) var. (string):
  - gen dumy_ch = ""
  - replace dumy_ch="Low" if one <= 5
  - replace dumy_ch="High" if one > 5
- dummy (binary) var. (numerical):
  - gen dumy_num = (one <= 5)
- labialization of values
  - lab define dum 1 "Low" 0 "High"
  - label value dumy_num dum

Subset of dataframe

Keep and drop: apply on variables and observations

- keep lines with two<=16: keep if two <=16
- drop lines with two>=14: drop if two >=14
- delete column ratio: drop ratio
- keep columns one and three: keep one three
- you can think of a combination of conditions

Exportation

Exportation in text file

save example , replace

Exportation in Excel file

export excel using example.xlsx , firstrow(variables) replace

Exportation in CSV file

export delimited using example.csv, replace

Exportation in text file

export delimited using example.txt, replace

Importation

Exportation in text file

import example , clear

Exportation in Excel file

import excel example.xlsx , firstrow clear

Exportation in CSV file

import delimited example.csv, clear

Exportation in text file

import delimited example.txt, clear

“VERTICAL MERGE” (APPEND)

TWO DATA

- First data
  - sysuse auto, clear
  - keep if _n <=37
  - save first37 , replace
- Second data
  - sysuse auto, clear
  - keep if _n>37
  - save after37 , replace

"VERTICAL MERGE" (APPEND)

"VERTICAL MERGE" (APPEND)

- you are merging two data by using the names of variables:
  - use first37 , clear
  - append using after37

"HORIZONTAL MERGE" (MERGE)

Two data

- First data
  - sysuse auto, clear
  - keep if _n <=37
  - keep make price
  - rename (make price)(make_1 price_1)
  - gen identifier = _n
  - save first37_merge , replace
- Second data
  - sysuse auto, clear
  - keep if _n>37
  - keep make price
  - gen identifier = _n
  - save after37 , replace

"HORIZONTAL MERGE" (MERGE)

"HORIZONTAL MERGE" (MERGE)

- you are merging two data by using "identifier(s)" of row:
  - use first37 , clear
  - merge 1:1 identifier using after37

Exercise

- Use data "auto" integrated in STATA
- describe the data
- detailed sta des for variables : price, weight, rep78, trunk, length, and foreign
- frequency table for rep78 and look for missing values
- cross tabulation of rep78 and foreign
- generate the dummy/qualitative variables:
  - domestic (counterpart of foreign)
  - small_trunk taking value 1 if trunk < 10
  - standard_trunk taking value 1 if trunk ≥ 10 and trunk $<$ 20
  - huge_trunk taking value 1 if trunk ≥ 20
  - trunk_categ taking values 1, 2, and 3 respectively for small, standard, and huge trunks

Exercise

- create a label and applies it on the variable trunk_categ
- create lprice = log(price)
- Keep variables foreign price trunk (all) weight length
- export your data in STATA and excel formats

Exercise: result

- sysuse auto, clear
- d
- codebook
- sum rep78 price weight foreign trunk length, det
- tab rep78
- tab rep78, mis
- tab foreign
- tab foreign, nolabel
- gen domestic = 1- foreign
- sum trunk, det
- gen trunk_small = (trunk<10)

Exercise: result

- gen trunk_standard = (trunk>=10)*(trunk<20)
- gen trunk_huge=(trunk>=20)
- gen trunk_categ= trunk_small + 2*trunk_standard+ 3*trunk_huge
- lab def catego 1 "Small" 2 "Standard" 3 "Huge"
- lab value trunk_categ catego
- keep foreign price trunk* weight length
- gen lprice = log(price)
- save autoME, replace
- export excel using autoME.xlxs , replace firstrow(variables)

EGEN

EGEN

- allows to create a variable based on *STATA's specific functions* that involve several variables or rows
  - sysuse auto, clear
  - egen rowmean= rowmean(price - turn)
    - edit
    - sum rowmean
  - egen mean_price = mean(price) , by(rep78)
    - edit
    - ta mean_price rep78, m
- type "help egen" to see all available functions for egen

Collapse

Collapse

- allow to make dataset of summary statistics
  - sysuse auto , clear
  - collapse (mean) price weight (min) trunk , by(rep78)
  - edit

PRESERVE/RESTORE

PRESERVE/RESTORE

- preserve/restore: allow to (i) have two copies of a data (= preserve stage), (ii) manipulate a copy, (iii) restore the un-manipulate copy (== restore stage)
    - sysuse auto , clear
    - preserve
        - collapse (mean) price weight (min) trunk , by(rep78)
        - save data_manipulate , replace
    - restore
    - edit

Scatter plot

Scatter plot

- sysuse auto, clear
- scatter plot: scatter price weight
- remove the color at the background: scatter price weight , graphregion(color(white))
- add linear fit:
  graph twoway (scatter price weight, msize(small))(lfit price weight), graphregion(color(white)) title("Scatterplot and OLS fitted line")

correlation: corr price trunk

Useful plots

Histogram

- gen lprice = log(price)
- hist lprice , graphregion(color(white)) name(gh)

Empirical cumulative density function

- cumul lprice, gen(cum_lprice)
- sort cum_lprice
- line cum_lprice lprice, graphregion(color(white)) title("Cumulative of median family income") name(gc)

Empirical density

kdensity lprice , graphregion(color(white)) name(gk)

Box plot

graph box lprice , graphregion(color(white)) name(gb)

Useful plots

Put all plots togheter

graph combine gh gc gk gb , graphregion(color(white))

PLOT FUNCTIONS

PLOT FUNCTIONS

- $f(x) = x^2$: tw function y = x^2 , range(-2 2) graphregion(color(white)) ytitle("f(x)")
- plot density of lprice + normal distribution:
  - sum lprice
  - sca m=r(mean)
  - sca sd= r(sd)
  - tw (kdensity lprice) (function y = (1/(scalar(sd)*sqrt(2*_pi)))*exp(-0.5*((x-scalar(m))/scalar(sd))^2), range(7 10)), graphregion(color(white)) ytitle(density) legend(position(11) col(1) ring(0) label(1 "Kernel empirical pdf") label(2 "Normal pdf"))

Otimization over data (e.g. ML routine)

$$\min_{a,b} \sum_{i=1}^{n} (y_i - a x_i - b)^2$$

with y and x the lprice and weight in data auto

Result

- sysuse auto , clear
- gen lprice=log(price)
- program drop _all
- the program:
  program define ols_ml
  args lnf a b
  tempvar y x
  qui {
  generate double ‘y’ = $ML_y1 generate double ‘x’ = $ML_y2
  replace ‘lnf’ = -(‘y’- ‘a’*‘x’-‘b’)^2
  }
  end
- ml model lf ols_ml (a: lprice weight =) (b:) , maximize
- ml di
- checking with reg: reg lprice weight

Otimization over data (e.g. ML routine)

$$\max_p \sum_{i=1}^{n} \left[y_i log(p) + (1 - y_i) log(1 - p) \right]$$

with y the dummy variable taking value 1 if rep78 ≤ 2 and 0 otherwise.

Result

- gen rep78small =(rep78<=2)
- program :
```
program define bin_ml
args lnf p
tempvar y x
qui {
generate double `y' = $ML_y1
replace `lnf' = `y'*log(`p')+(1-`y')*log(1-`p')
}
end
```
- ml model lf bin_ml (p: rep78small =) , maximize
- ml di
- checking with sum: sum rep78small
- rmk: ML routine maximizes functions
  - so need to adapt if the objective is to minimize functions

FORVALUE

FORVALUE

```
forval i=1/74 {
qui sum weight if _n <= `i'
di "mean of lprice of the `i' first observations is: " r(mean)
}
forval i=1/100 {
if mod(`i',3) == 0 {
di "`i' is a multiple of 3"
}
}
```

FOREACH

FOREACH

```
global VAR weight length
foreach x in $VAR {
scatter lprice `x' , graphregion(color(white)) name(`x')
}
graph combine $VAR , graphregion(color(white))
```

WHILE

WHILE

```
sum lprice
sca min = r(min)
sca j = 1
local i = 1
while (lprice[`i']!= r(min)) { local i = `i' + 1
sca j = `i'
}
di "the min of lprice is at the row: " j
```

Random values and re-allocation

Useful random values

- normal: rnormal(mu, sigma)
  - $N(0,1)$: gen sdnorm= rnormal(0,1)
- uniform: runiform(a,b)
  - $U[0,1]$: gen sdunif = runif(0,1)

Random draw with replacement

- bsample
  - very useful to conduct "manually" a bootstrapping analysis in order to compute standard deviation/confidence interval

Exercise: an illustration of CLT

- First replication
- simulates a variable X of $n = 1000$ random values of $U(0,1)$
- generate

$$Y = \begin{cases} 1 & \text{if} \quad X \leq 0.4 \\ 0 & \text{if} \quad X > 0.4 \end{cases}$$

  - remark that Y follows a $\mathcal{B}(p)$ with $p = 0.4$
- Compute the sequence $\{\overline{Y}_1, \overline{Y}_2, ..., \overline{Y}_{1000}\}$ with

$$\overline{Y}_i = \frac{1}{i} \sum_{j=1}^{i} Y_i$$

- plot all $(\overline{Y}_i, i)$

Result

- clear all
- set obs 1000
- set seed 123456789
- gen X = runiform()
- gen Y= (X<=0.4)
- gen Ybar=.
- qui forvalue i=1/1000 {
  sum Y if _n<=‘i’
  replace Ybar = r(mean) in ‘i’
  }
- gen i = _n
- line Ybar i, graphregion(color(white)) yline(0.4)

Exercise

- provides an approximated value for $F(x)$

$$F(x) = \int_{-10}^{10} f(x)dx$$

  with $f(.)$ the density function of the standard normal distribution, and $\pi \simeq 3.14$
  - hint: $F(x) = (10 - (-10)) \times E[f(x)]$, with $E[.]$ the expectation computed with $U[-10, 10]$
  - this value is almost 1

1. simulates a vector x of $n = 100000000$ random values of $U(-10, 10)$
2. compute f(.) over the simulated values of x
3. Compute the mean of the $\overline{f}$ over the n simulated values
   - rmk (central limit theorem): $\overline{f}$ converges in proba towards $E[f(x)]$
4. compute the approximated values as $20 \times \overline{f}$

Result

- clear all
- set obs 100000000
- gen x=runiform(-10,10)
- gen f= (1/sqrt(2*_pi))*exp(-0.5*x^2)
- sum f
- di "the approximated value is:" 20*r(mean)

Exercise

Provide an approximation for the quantity π

- hint: leverage on (i) the area of a circle $x^2 + y^2 \leq 1$, (ii) the area of a square centered at (0,0) and whose side is 2, and (ii) random draws of x and y from $U[-1,1]$.

Result

- clear all
- set obs 100000 simulate random values on the square whose side is 2:
  - gen x=runiform(-1,1)
  - gen y=runiform(-1,1)
- identify random values that belong to the area of a circle $x^2 + y^2 \leq 1$
  - gen circle = (x^2 + y^2 <=1)
- di "the approximated value is:" 4*r(mean)
  - note that "4" corresponds to the area of the square whose side is 2
- visualization: tw (scatter y x)(scatter y x if circle==1), graphregion(color(white)) legend(label(1 "Square") label(2 "Disque"))

Next: Latex and its connection with R and STATA !