

Basics of STATA Software

Introduction to the STATA interface and its functions. Entering and importing data into STATA from economic databases and various data files. Basic data management: opening, viewing, and editing variables. Generating and labeling variables, creating data subsets

Tomáš Oleš

Department of Economic Policy
Faculty of Economics and Finance

February 1, 2025

Agenda

- Get accustomed to the STATA interface.
- Enter and import data into STATA.
- Get comfortable with using the STATA command language.
- Learn common data management commands in STATA.

Why Research Needs to be Reproducible?



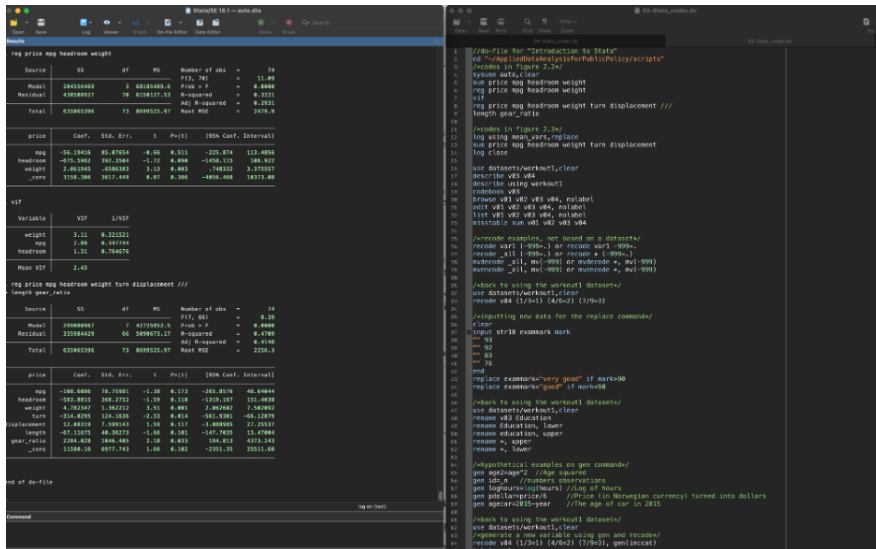
Why Research Needs to be Reproducible?

- "The natural (social) scientist is concerned with a particular kind of phenomena . . . he has to confine himself to that which is reproducible . . . I do not claim that the reproducible by itself is more important than the unique. But I do claim that the unique exceeds the treatment by scientific method. Indeed it is the aim of this method to find and test natural laws." — Wolfgang Pauli
- "Kohn's Second Law: An experiment is reproducible until another laboratory tries to repeat it." — Alexander Kohn

STATA Interface

```
Stata12 16.1 -- auto.dta
Log Viewer Output Data Editor
Results
Mean VIF | 2.43
.
end of do-file
. do "C:\Users\teemeslas\Library\CloudStorage\Dropbox\Teaching\AppliedDataAnalysisForPublicPolicy/scripts"
  "C:\Users\teemeslas\Library\CloudStorage\Dropbox\Teaching\AppliedDataAnalysisForPublicPolicy/scripts"
.
. sysuse auto, clear
(1978 Automobile Data)
. sum price mpg headroom weight
+-----+-----+
| Variable | Obs | Mean | Std. Dev. | Min. | Max. |
+-----+-----+
| price    | 74  | 6165.257 | 2949.498 | 3091 | 15849 |
| mpg      | 74  | 22.2973 | 5.78580 | 12  | 41  |
| headroom | 74  | 2.89393 | .805998 | 1.5  | 5  |
| weight   | 74  | 3019.459 | 777.1936 | 1769 | 4848 |
+-----+-----+
. reg price mpg headroom weight
+-----+-----+
| Source | SS | df | MS | Number of obs = 74 |
+-----+-----+
| Model | 28455.609 | 3 | 9485.203 | F(3, 70) = 15.89 |
| Residual | 4385.8927 | 70 | 62.79877 | Prob > F = 0.000 |
| Total | 32841.5016 | 73 | 449.885 | Adj R-squared = 0.291 |
+-----+-----+
| Total | 32841.5016 | 73 | 449.885 | Root MSE = 247.9 |
+-----+-----+
+-----+-----+
| Price | Coef. | Std. Err. | t | P>|t| | 95% Conf. Interval |
+-----+-----+
| mpg | -56.19410 | 85.97004 | -0.66 | 0.511 | -225.874 | 113.4856 |
| headroom | -835.5802 | 392.3584 | -2.12 | 0.038 | -1658.133 | 146.972 |
| weight | 2.861945 | .6580283 | 4.33 | 0.000 | 1.54332 | 4.17957 |
| _cons | 3158.388 | 3617.449 | 0.87 | 0.388 | -4056.488 | 10373.88 |
+-----+-----+
. vif
+-----+-----+
| Variable | VIF | 1/VIF |
+-----+-----+
| weight | 3.11 | 0.321521 |
| mpg | 2.88 | 0.347144 |
| headroom | 1.31 | 0.764676 |
+-----+-----+
| Mean VIF | 2.43 |
+-----+-----+
.
end of do-file
.
Command
```

A Key to Reproducibility is to Learn to Code < 3



The image displays two side-by-side screenshots of the RStudio interface, illustrating the process of data analysis and code execution.

Left Screenshot (RStudio Console and Environment):

The console shows the results of a linear regression model:

```
reg price mpg headroom weight
```

Source	SS	df	MS	Number of obs	F(3, 76)	Prob > F	R-squared	Adj R-squared	Root MSE
Model	28055660	3	9351886.67	79	11.49	0.0000	0.8221	0.8231	2479.9
Residual	43048027	76	566290.00						
Total	63063396	79	800925.97						

The Environment pane shows the following variables:

price	Coef.	Std. Err.	t	Pr(> t)	[95% Conf. Interval]
mpg	-56.10406	85.47654	-0.66	0.511	-225.874 113.4856
headroom	-475.3962	392.3164	-1.22	0.404	-1456.115 148.322
weight	2.461843	6386383	3.13	0.003	-748332 3.370557
_cons	3158.386	3617.449	0.87	0.386	-4856.468 14373.80

The VIF (Variance Inflation Factor) table shows:

Variable	VIF	1/VIF
weight	3.21	0.311521
mpg	1.86	0.537744
headroom	1.21	0.826476

The Mean VIF is 2.43.

The console also shows the results of a second regression model:

```
reg price mpg headroom weight turn displacement ///
length gear_ratio
```

Source	SS	df	MS	Number of obs	F(7, 61)	Prob > F	R-squared	Adj R-squared	Root MSE
Model	29888867	7	4271266.71	68	0.0000	0.0000	0.4769	0.4146	2256.2
Residual	33598429	60	559973.82						
Total	63063396	79	800925.97						

The Environment pane shows the following variables:

price	Coef.	Std. Err.	t	Pr(> t)	[95% Conf. Interval]
mpg	-185.6888	78.75981	-2.36	0.173	-263.4576 48.68844
headroom	-583.8815	368.2752	-1.59	0.118	-1319.167 151.4839
weight	4.782347	1.362212	3.51	0.001	2.062682 7.502002
turn	-316.4255	124.3436	-2.53	0.014	-561.9361 -68.12879
displacement	32.48109	7.949143	4.09	0.000	16.58895 48.37323
length	-67.11675	48.36273	-1.39	0.161	-167.7635 33.47404
gear_ratio	2284.628	1846.485	1.24	0.223	-184.813 4373.243
_cons	11584.16	6977.743	1.66	0.102	-2351.25 25511.68

Right Screenshot (RStudio Editor):

The editor shows a script file named "intro2do-file.R" with the following code:

```
1 //do-file for "Introduction to Stata"
2 cd "~/AppliedDataAnalysisForPublicPolicy/scripts"
3 //codes in figure 2.24/
4 sysuse auto,clear
5 sum price mpg headroom weight
6 reg price mpg headroom weight
7 vif
8 reg price mpg headroom weight turn displacement ///
9 length gear_ratio
10
11 //codes in figure 2.34/
12 log using mean_vars,replace
13 sum price mpg headroom weight turn displacement
14 log close
15
16 use datasets/workout1,clear
17 describe v84
18 describe using workout1
19 codebook v84
20 browse v81 v82 v83 v84, nolabel
21 edit v81 v82 v83 v84, nolabel
22 list v81 v82 v83 v84, nolabel
23 xtestable sum v81 v82 v83 v84
24
25 //encode examples, not based on a dataset/
26 recode v81 (-999=-) or recode v81 (-999=-)
27 recode _all (-999=-) or recode _all (-999=-)
28 recode _all, mv(-999) or mvrecode *, mv(-999)
29 mvrecode _all, mv(-999) or mvrecode *, mv(-999)
30
31 //back to using the workout1 dataset/
32 use datasets/workout1,clear
33 recode v84 (1/3=1) (4/6=2) (7/9=3)
34
35 //inputting new data for the replace command/
36 clear
37 input str16 exammark mark
38 == 91
39 == 92
40 == 93
41 == 76
42 end
43 replace exammark="very good" if mark=90
44 replace exammark="good" if mark=98
45
46 //back to using the workout1 dataset/
47 use datasets/workout1,clear
48 rename v83 Education
49 rename Education, lower
50 rename education, upper
51 rename *, upper
52 rename *, lower
53
54 //hypothetical examples on gen command/
55 gen age2=age^2 //age squared
56 gen id_n //numbers observations
57 gen loghours=log(hours) //log of hours
58 gen pdollar=price/6 //Price (in Norwegian currency) turned into dollars
59 gen agncar=2015-year //The age of car in 2015
60
61 //back to using the workout1 dataset/
62 use datasets/workout1,clear
63 //generate a new variable using gen and recode/
64 recode v84 (1/3=1) (4/6=2) (7/9=3), gen(secat)
65 lab secat
```

Create Your First Code and Run It

```
display("Hello Word!")
```

Using log-file

```
*Load data and create a log-file
log using "~/AppliedDataAnalysisForPublicPolicy/log/my_first_log.log",
    ///
replace
sum price mpg headroom weight turn displacement
log close
```

Entering data

```
/*inputting new data for the replace command*/  
clear  
input str10 exammark mark  
" " 93  
" " 92  
" " 83  
" " 76  
end
```

Importing Data

Default Data Format: STATA primarily uses `.dta` files but supports multiple formats, including:

- `.xls`, `.xlsx` (Excel)
- `.csv` (Comma-separated values)
- `.shp`, `.dbf` (Shapefiles & dBase)
- ...

What if my data is in an unsupported format? Convert it to a STATA-compatible format (e.g., `.csv`) using Python or another tool.

Limitation: STATA can handle only one dataset (table) at a time.

Tasks:

- Import `datasets/highest-points-by-state.csv` into STATA.
- Import `datasets/workout1.dta`.

Solution: Importing Data into STATA

Importing a CSV File:

```
import delimited "datasets/highest-points-by-state.csv", clear
```

Importing a STATA (.dta) File:

```
use "datasets/workout1.dta", clear
```

Converting Unsupported Formats: Use Python (e.g., Pandas) to convert files:

```
import pandas as pd
df = pd.read_json("data.json")
df.to_csv("data.csv", index=False)
```

Examining the Data

```
use datasets/workout1,clear
describe v03 v04
describe using workout1
codebook v03
browse v01 v02 v03 v04, nolabel
edit v01 v02 v03 v04, nolabel
list v01 v02 v03 v04, nolabel
misstable sum v01 v02 v03 v04
```

Making Changes to Variables

**Recode examples, not based on a dataset*

```
recode var1 (-999=.) or recode var1 -999=.
```

```
recode _all (-999=.) or recode * (-999=.)
```

```
mvdecode _all, mv(-999) or mvdecode *, mv(-999)
```

```
mvencode _all, mv(-999) or mvencode *, mv(-999)
```

**Back to using the workout1 dataset*

```
use datasets/workout1,clear
```

```
recode v04 (1/3=1) (4/6=2) (7/9=3)
```

Replacing the Data

**Inputting new data for the replace command*

`clear`

`input str10 exammark mark`

`"" 93`

`"" 92`

`"" 83`

`"" 76`

`end`

`replace exammark="very good" if mark>90`

`replace exammark="good" if mark<90`

**Back to using the workout1 dataset*

`use datasets/workout1,clear`

`rename v03 Education`

`rename Education, lower`

`rename *, upper`

Mathematical Operators in STATA

Arithmetic		Logical		Relational	
+	addition	&	and	>	greater than
-	subtraction		or	<	less than
*	multiplication	¬	not	≥	greater or equal
/	division	!	not	≤	less or equal
^	power			==	equal
-	negation			!=	not equal
+	string concatenation			~=	not equal

Table: The three types of mathematical operators used in STATA

Today 09:09 AM

I love you.. will you be my Gf?



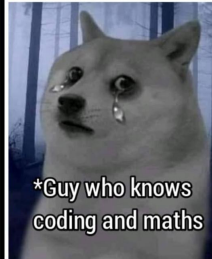
! yes



Message...



***Normal Guy**



***Guy who knows
coding and maths**

Generating variables: workout1.dta

```
*Hypothetical examples on gen command
gen age2=age^2    *Age squared
gen id=_n *Numbers observations
gen loghours=log(hours) *Log of hours
gen pdollar=price/6    *Price (in Norwegian currency) to dollars
gen agecar=2015-year    *The age of car in 2015
```

*Back to using the workout1 dataset

use datasets/workout1,clear

*Generate a new variable using gen and recode

recode v04 (1/3=1) (4/6=2) (7/9=3), gen(inccat)

tab inccat

*Generate a new variable using gen and replace

gen inccat2=.

replace inccat2=1 if (v04<=3)

replace inccat2=2 if (v04>=4) & (v04<=6)

replace inccat2=3 if (v04>=7) & (v04<.)

tab inccat2

*Hypothetical example showing labelling values of several variables

label define lablikert 1"disagree" 6"agree"

label values var1-var5 lablikert

Appending Data

**Inputting new data manually*

`clear`

`input id data var1 var2`

`1 1 3 2`

`2 1 4 3`

`3 1 5 1`

`end`

`save dataset1,replace`

`clear`

`input id data var1 var2`

`4 2 3 1`

`5 2 5 3`

`6 2 5 4`

`end`

`save dataset2,replace`

`clear`

*Appending data

append using dataset1 dataset2,gen(dataset3)

save dataset3

list,sep(0)

Merging Data

```
* Inputting new data manually
clear
input id v1_14 v2_14
1 3 5
2 4 5
3 2 3
4 1 2
5 1 2
end
save data14,replace
```

```
clear
input id v1_15 v2_15
1 4 5
2 5 5
3 3 4
4 2 3
5 2 3
end
save data15,replace
clear
```

```
*Merging data
use data14,clear
merge 1:1 id using data15
save data1415,replace
drop _merge *drops this variable
list, sep(0)
```

Reshaping Data

```
* Reshape from wide to long
use data1415,clear * We use the data from above since it is in a wide
    format
drop _merge
list
reshape long v1_ v2_ , i(id) j(year)
list
```

- Laffers, L. (2021). *Draft poznámok k predmetu Moderná Aplikovaná regresia 1*. UMB Banská Bystrica.
- Mehmetoglu, M. and Jakobsen, T. G. (2022). *Applied Statistics using Stata: a Guide for the Social Sciences*. Sage.