

Why Stata ?

- portability over different platforms Macintosh, Unix, Windows
- speed
- simplicity of commands compare to other statistical software and homogenous syntax of all commands
- big spectrum of statistical procedures fast updated by the Stata Company and private users
- very good help in form of manuals or web materials
- advanced programming language
- price

possible drawback

- if the size of data exceeds RAM Stata is very very slow, however for example:
with 400MB memory set to Stata = 1 million observations with 100 variables or
= 10 millions observations with 10 variables
- Stata/SE can use not more than 32 000 variables

STATA = Data Manipulation + Statistical Analysis + Graphics
Basically can be used and think of independently

www.stata.com

Check what directory you are in

pwd

Stata as a calculator

display "2+2=" 2+2

recalling previous commands: Pg-Up; Pg-Down

Inputting data by hands: input command

```
input  v1 v2 v3
      1  2  3
      22.3 4 34
end
```

Inputting data which is in free format from the external file

infile id year sex height str10 city using "data 1.txt"

- applying "..." for the path of the file ← space in the name
- str10 precedes a string variable(city), 10 ← length of the string variable

other commands for inputting data

- infix ← ASCII file in fixed-column format
- insheet ← ASCII file, separator is comma or tabulation sign
- Stat Transfer software transfers data between Stata and other 25 different packages like SAS, Splus, Excell,...

Check the commands outfile and outsheet corresponding to infile and insheet but for outputting data from Stata.

Saving data

ALL DATA AND CHANGES MADE TO IT IS LOST IF NOT SAVED.

save data1 ← if the file data1.dta does not exist

save data1, replace ← if the file data1.dta exists

- saving without space in the name ← one does not have to use "..." for the name
- *.dta ← dta is a default extension of Stata data file

Opening data

ONLY ONE FILE CAN BE OPEN=ACTIVE=ACCESSIBLE AND OPENING A NEW FILE CAUSES DROPPING THE PREVIOUS FILE FROM BEEING ACTIVE. THE CHANGES IN THE PREVIOUS FILE, IF NOT SAVED, WILL BE LOST FOREVER.

use data1

use data1, clear ← opening the data without caring about possible lost of changes made earlier in the session.

The clear command

THE CLEAR COMMAND AS SUCH OR USED AS AN OPTION ESSENTIALLY RESETS STATA.

Windowing in Stata

Stata Results Window: changing fonts
 Window: Results, Commands, Viewer, ...
 Prefs→Save Windowing Prefs;
 Begin Log → use *.log extension if you are going to transform output later after the session

Looking at data

describe ← quick look at an active data file, if any active

STATA COMMANDS ARE CASE SENSITIVE AND CAN BE ABBREVIATED.

describe or describ or ... or d still works the same

The rules of abbreviation are specific for each command.

list or list year sex year ← listing all or choosen variables

list year sex year ,divider ← modifying the basic charcter of the command by applying option divider

MOST STATA COMMANDS HAVE OPTIONS WHICH ARE IMPORTANT PART OF THE COMMAND AS THEY OFTEN MODIFY CONSIDERABLY THE BASIC CHARACTER OF THE COMMAND. ALWAYS CHECK OPTIONS FOR THE COMMAND.

list year sex year if sex==0 in 1/5 ← listing subset of data: males appearing in the first 5 observations; after if logical condition; after in range of observations

EACH DATA FILE CAN BE THOUGHT OF AS A MATRIX WITH VARIABLES AS COLUMNS AND OBSERVATIONS AS RAW.

Summarizing data

summarize; summarize height; ← summary basic statistics about all or choosen variables

summarize height year, detail ← more detailed statistics if option detail applied

Creating new and modifying existing variables, missing values in expressions

generate age=2003-year ← creating new variable

replace height=height/100 ← modifying existing variable

generate v1=log(sex)+2*abs(height) ←great variety of functions available in Stata
 generate v2= v1+abs(height) if city=="Oulu" to use in expressions like for example:

generate y=abs(x) ←absolute value |x|

generate y=exp(x) ←exponential

generate y=ln(x) ←natural logarithm

generate y=log10(x) ←base 10 logarithm
 generate y=sqrt(x) ←square root
 generate y=int(x) ←integer part of x. *int(5.8) = 5*
 generate y=round(x) ←nearest integer. *round(5.8) = 6*
 generate y=mod(x1,x2) ←modulus; the remainder after dividing x1 by x2
 generate y=max(x1,...xn) ←maximum value of arguments
 generate y=min(x1,...xn) ←minimum value of arguments
 generate y=sum(x) ←cumulative sum across observations, from first to current obs.

GENERALLY: MISSING VALUES (“.”) ARE PRODUCED BY EXPRESSION IF THE FUNCTION IN THE EXPRESSION DOES NOT MAKE SENSE, THE EXPRESSION DOES NOT APPLY IN A GIVEN CONTEXT (IF...) OR INCLUDES MISSING VALUES AS ARGUMENTS

IN STATA8 THERE ARE MISSING VALUES: ., .a, .b, .c, .d, ..., .z

ANY NUMBER < MISSING VALUE AND BETWEEN MISSING VALUES: .< .a< .b<< .z

CHECK HOW COMMANDS TREAT MISSING VALUES, USUALLY BY CHECKING OPTIONS OF THE COMMAND OR HELP OF THE COMMAND.

Data editor and data browser icons

The general syntax of a Stata command:

command [varlist] [= expression] [weight] [if exp] [in range] [, options]
 ←[...] means that ... is optional

The Stata operators:

arithmetic operators: + (addition), - (subtraction), * (multiplication), / (division), ^ (power)

string operators: + (string concatenation), for example the expression "abc" + "def" produces "abcdef"

relational operators: < (less than), > (greater than), <= (less or equal), >= (greater or equal), == (equal), != (not equal);

logical operators: & (and), | (or), . = (not)

= is a part of expression: replace year =1995 in 53 whereas

== is a logical operator: replace year =1900 if year == 1800

generate v3=year^2 + height if sex==0 & year>=1960 in 10/100

Changing and renaming variables

rename old_name new_name ← renaming variable
 rename height height_cm

Changing order of variables

order id year city ← changing order of variables in data to id year city, the rest as it was

One can use replace with if to change variables but it is much easier and clearer with

recode x 1=2 3=4 ← changes 1 in x to 2 and 3 to 4

recode x 2 7/9=12 ← changes 2,7,8,9 in x to 12

OTHER USEFUL STATA COMMANDS

Counting how many observations satisfies conditions, also in different subsets (prefix: by)

[by varlist:] count [if exp] [in range]
by sex: count if city=="Helsinki"

Sorting variables

sort varlist ← sorts variables in ascending order
gsort +height ← the same as gsort height or sort height
gsort +city ← sorts the character variable city in alphabetical order
gsort -city ← sorts the character variable city in reverse alphabetical order
gsort +city -height ← sorts the character variable city in alphabetical order and the variable height in ascending order

Calculating certain functions of variables giving constant as a result and storing the result as variable

egen newvar = function(stuff) [if exp] [in range] [, options]

The egen command provides an extension to generate. egen creates newvar equal to function(stuff). Depending on function() stuff refers to an expression, a list of variables, or a list of numbers. The options are similarly function dependent. Read help for full details.

egen kurt_height_male = kurt(height) if sex==0

append appends a Stata-format dataset stored on disk to the end of the dataset in memory

append using newdata ← appends the data set: newdata.dta to the active data set

Merging observations one-to-one from two data sets using variables as criteria for joining

merge [varlist] using filename ← the basic form, check help for more options

This command joins corresponding observations from the dataset currently in memory (called the master dataset) with those from the Stata-format dataset stored as filename (called the using dataset) into single observations (if filename is specified without an extension, .dta is assumed). The _merge variable will mark the source of the resulting observation:

_merge==1 (obs. from master data)
_merge==2 (obs. from using data)
_merge==3 (obs. from both master and using data)

The variables in [varlist] have to be sorted in both files before applying the command.

```
use data2
sort id
save, replace
use data1
sort id
merge id using data2
keep if _merge==3
```

Reshaping data

collapse ← generally speaking this command converts data from individual to grouped(aggreated) data

xpose ← transposes the matrix of data: observations and variables, i.e. observations become variables and variables become observations

reshape ← converts data from wide to long form and vice versa

<u>wide form</u>		<u>long form</u>
id popu1 popu2 popu3 popu4		id age popu
1 140 120 130 135	← reshape →	1 1 140
2 155		1 2 120
		1 3 130
		1 4 135
		2 1 155
	

Tabulating data

tabulate weight if sex==1 ← tabulating one variable

Producing two by two tables with different options:

tab2 weight_big height_big if sex==1, row ← relative frequency of cells in rows

tab2 weight_big height_big if sex==1, column ← relative frequency of cells in columns

tab2 weight_big height_big if sex==1, all ← tests for independence rows and columns

tab2 weight_big height_big if sex==1, exact ← exact Fisher's test for independence

tab2 weight_big height_big city if sex==1, all ← more 2 by 2 tables produced by one command

To tabulate data over more than two dimensions use by prefix:

sort sex city

by sex city: tab2 weight_big height_big

Some common statistics for data

means height weight if sex==0, level(90) ← estimation of different means with 90% ci

centile height weight if sex==0, centile(5 50 95) ← 5th, 50th, and 95th centiles of variables, ci calculated assuming binomial distribution(default)

centile height weight if sex==0, centile(5 50 95) normal ← 5th, 50th, and 95th centiles of variables, ci calculated assuming normal distribution

correlate height weight if sex==0 ← correlation matrix of variables

correlate height weight if sex==0, covariance ← covariance matrix of variables

Sometimes one have to stop calculations: Break-icon or Ctrl-Break

tab2 height weight if sex==0, exact

Check log icon to see if Stata is running

Linear regression

regress depvar [varlist] [weight] [if exp] [in range] [,options]
 depvar ← dependent variable

regres height weight if sex==1
 regres height weight if sex==1, nocnstant ← fitting without a constant term

Defining categorical variables and including them in linear predictor:

The xi: prefix has to be used if categorical variables are in linear predictor. From a n-level categorical variable xi: generates (n-1) indicator variables and they are referred to by the i. prefix to the original variable name:

xi: regres height weight i.sex i.city , noconstant

By default the first (lowest numerically or alphabetically) category will be omitted, i.e. be the reference category.

One may, before the analysis, select a different category to be the reference one by using the command char:

char sex[omit] 1 ← choosing a reference category for sex when the code is 1

Interactions terms are defined by using a star operator:

xi: regres height i.sex*weight

Use a help command in Stata, help xi, to check the full syntax for defining interactions between categorical and numerical variables.

Logistic regression

logistic depvar [varlist] [weight] [if exp] [in range] [,options]

The dependent variable depvar must be coded 0/1 (no/yes).

The xi: prefix for categorical variable applies as described above.

xi: logistic cancer_yes i.sex i.agegrp i.smoke ← results reported as odds ratios
 xi: logistic cancer_yes i.sex i.agegrp i.smoke, coef ← results reported as estimated coefficients

After running logistic, many commands available to obtain different statistics, for example:

- obtaining Hosmer-Lemeshow's goodness-of-fit test with 10 groups: lfit , group(10)
- obtaining a classification table, including sensitivity and specificity with a cut-off point of 0.3: lstat , cutoff(0.3).

Poisson regression

poisson depvar [varlist] [weight] [if exp] [in range] ,exposure(varname) [other options]

exposure(var) ← specifies the variable var that reflects the amount of exposure over which depvar events

xi: poisson deaths smokes i.age i.sex, exposure(pyears)

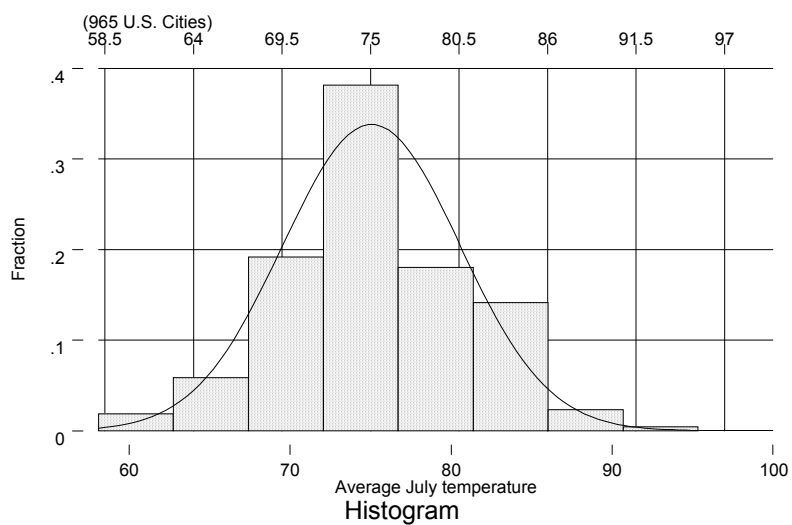
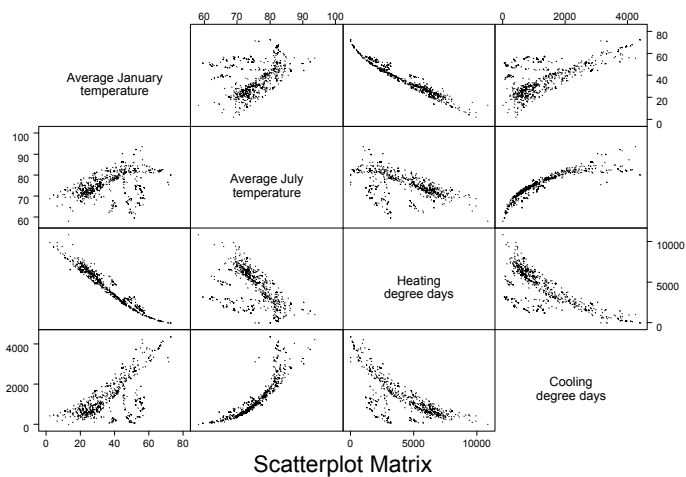
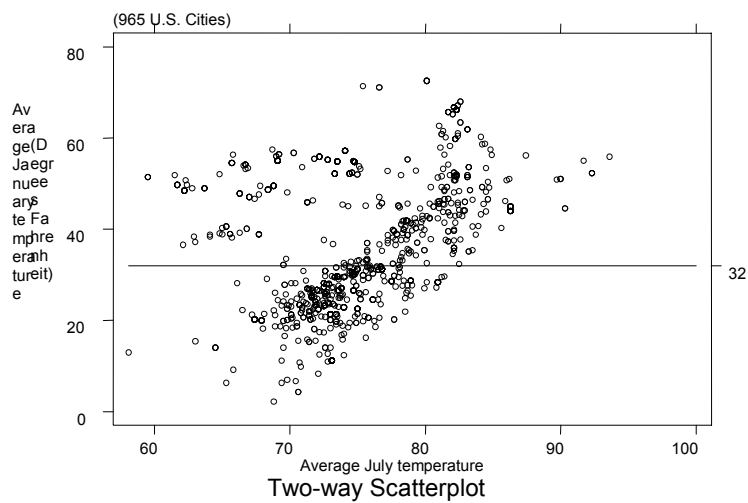
The xi: prefix for categorical variables applies as described above.

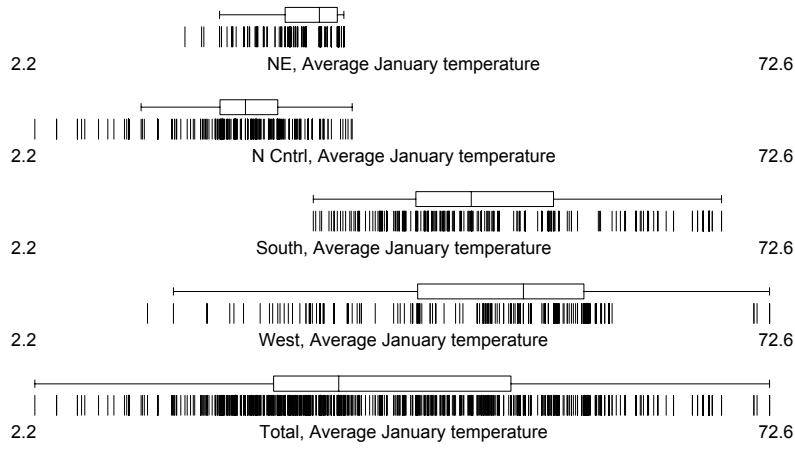
Use a help command in Stata, help estcom, to check the features of all estimation commands in Stata.

Graphs in Stata (general remarks only)

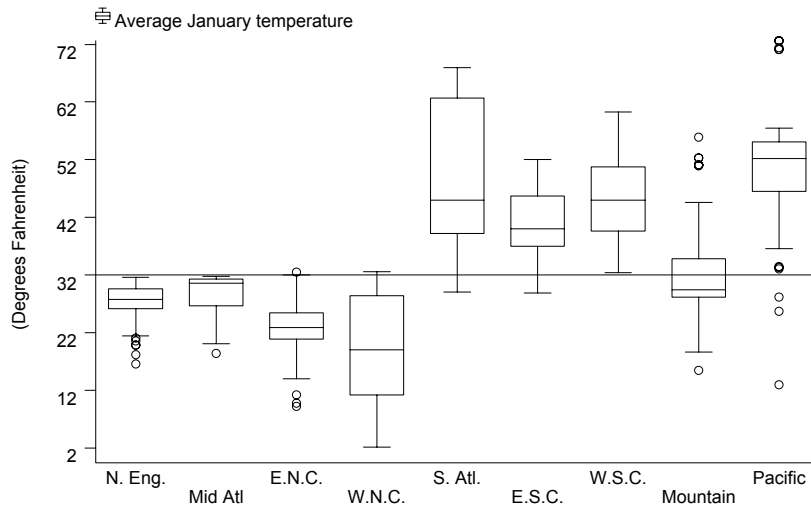
Graphic styles in Stata

The title below a graph gives the name of the style of the graph.

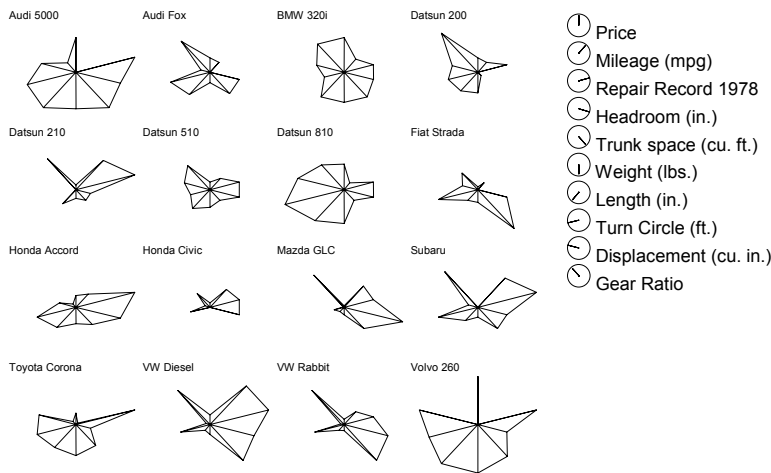




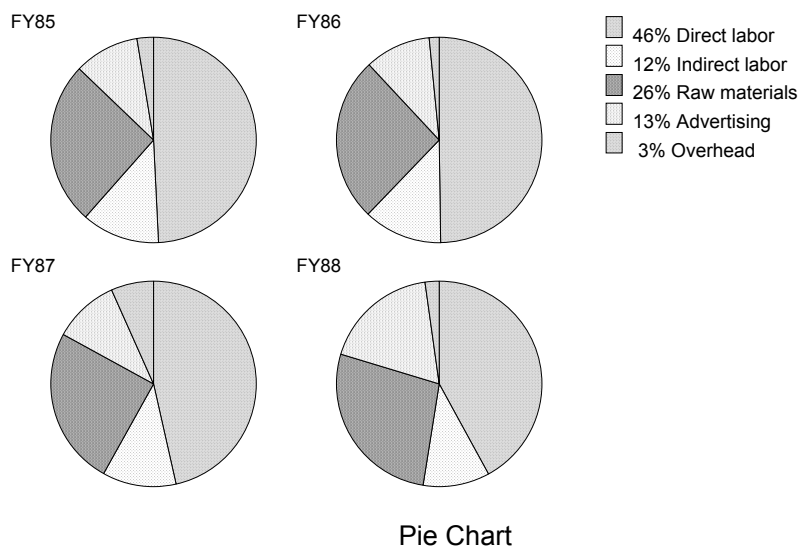
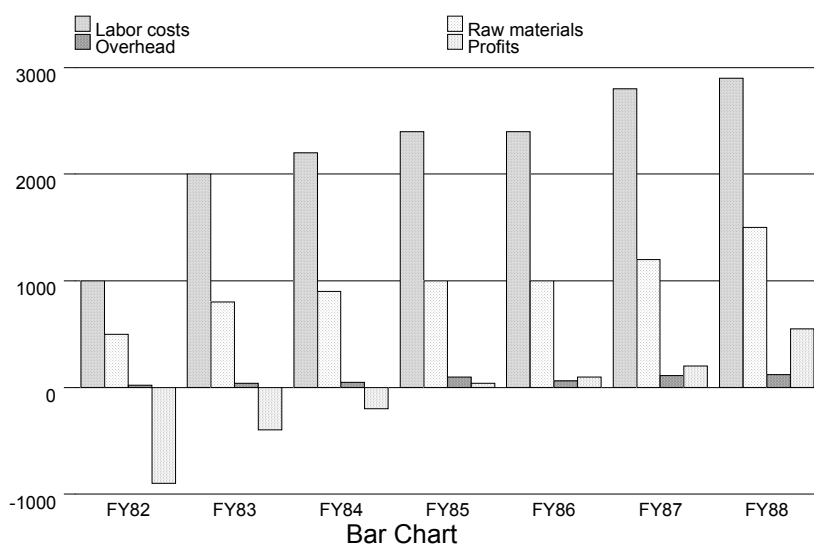
One-way Scatterplot



Box-and-Whisker Plot



Star Plot



Graphical commands in Stata8 are different from those in Stata7.

Stata can produce high-quality graphs, suited for publication in 7 different graphical formats including Window Metafile, TIFF and Encapsulated PostScript. Stata does not produce three dimensional graphs.

Graphics manual is complicated to use and the on-line help works better: `help graph_intro`.

Stata8 Graphics by Example:

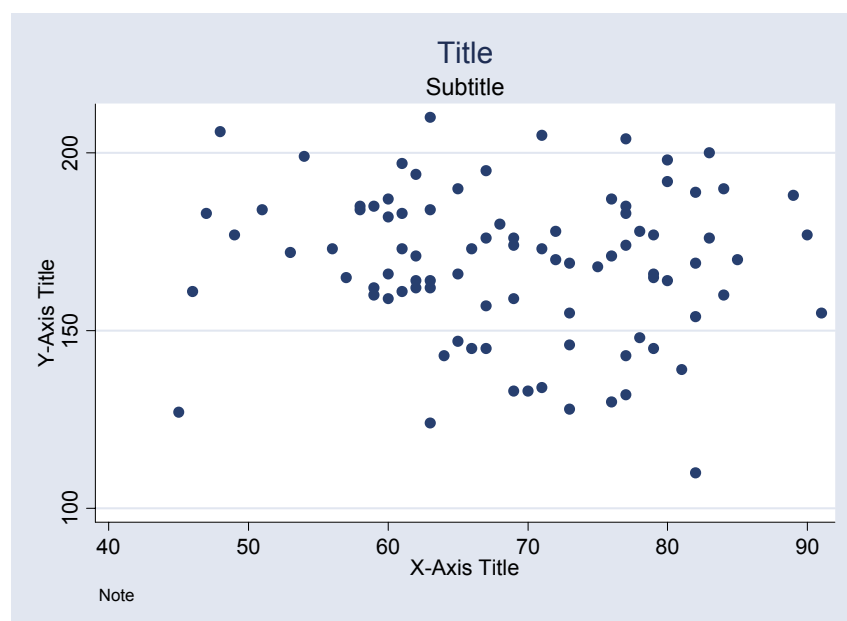
<http://www.ats.ucla.edu/stat/stata/webbooks/GraphByExample/default.htm>

With Stata's dialogs you can easily define a graph. Once you made your choices, press[Submit] rather than [OK]; this gives the opportunity to modify your choices after having looked at the result.

As a beginner, when creating a graph, you can also try to find the graph similar to your expectation in the manual and then modify the commands from the manual, part by part, in order to create your own graph.

Example of a basic graph command:

```
use data3, clear
tway (scatter height weight), ytitle(Y-Axis Title) xtitle(X-Axis Title) title(Title)
subtitle(Subtitle) note(Note)
produces:
```



Getting Help

help poisson or whelp poisson ← if you know the command
 search robust regression ← searching for words in help files
 findit robust regression ← as above plus searching net also

Stata User's Guide Release 8 is the best introduction to Stata8 if you have time.

Check also the command: tutorial contents for a tutorial.

**USUALLY ONE TASK CAN BE MADE IN MANY DIFFERENT WAYS IN STATA
 CHOOSING THE WAY IS A MATTER OF PERSONAL PREFERNCES OR HAVING
 ENOUGH TIME TO CHECK STATA MANULAS.**

Survey analysis in Stata

(based on “Stata Survey Data, Reference Manual, Release 8”)
which is a perfect booklet for self-studying

The prefix `svy` refers to survey data in Stata syntax.

Setting variables for a survey design and modifying the settings

`svyset` has to be used first before applying any other survey command in Stata

```
svyset [pweight = sampling_weight_variable],
      strata(strata_variable)
      psu(primary_sampling_unit_variable)
      [clear(clear_options) | clear]    ← use help svyset for the whole syntax
```

data from <http://www.stata-press.com/data/r8/nmihs.dta>

USA data about mothers who gave birth to their young.

```
use nmihs          ← opening the data
svyset             ← checking survey settings if any
svyset [pweight=finwgt], strata(stratan) psu(agegrp) ← setting the variable finwgt as
                  sampling weights, the variable stratan as defining different strata and the variable agegrp
                  to define primary sampling units
save, replace     ← saving survey settings with the data

clear             ← checking that the setting has been saved
use nmihs        ←
svyset           ←

g adjwgt = 2*finwgt          ← setting a new variable, adjwgt, as sampling weights
svyset [pweight=adjwgt]     ←

svyset, clear(strata psu)  ← erasing strata and psu settings from the design
svyset, clear              ← erasing all survey design settings
save, replace              ← saving without the survey settings
```

Calculating means

data from <http://www.stata-press.com/data/r8/nmihs.dta>

the same as above

```
use nmihs          ← opening the data
ci birthwgt       ← calculating the mean of the variable birthwgt WITHOUT survey design
svyset [pweight=finwgt], strata(stratan)          ← setting survey variables
svyset           ← checking survey settings
svymean birthwgt ← calculating mean of the variable birthwgt
Notice the difference in the estimates if the proper survey design is taken into account

svymean birthwgt, subpop(race) ← calculating means for one subpopulation, race is
                              is coded as 1/0 and when race==0 observations are omitted
svymean birthwgt, by(race marital) ← calculating means for more subpopulations
```

Check all options of the command `svymean` by using `help svymean`.

As the general rule do not use `if` in the survey analysis. Instead use `subpop()` options.

Tables

data <http://www.stata-press.com/data/r8/nhanes2b.dta>

USA data concerning diabetes

`svyset [pweight=finalwgt], strata(stratid) psu(psuid)` ← setting stratum variable as `stratanand` sampling weight as `finwgt`

`svytab race diabetes` ← proportions to total

`svytab race diabetes, ci` ← proportions to total with confidence intervals

`svytab race diabetes, row ci` ← proportions in rows with confidence intervals

`svytab race diabetes, col ci` ← proportions in columns with confidence intervals

Notice the difference in the test result after correction for survey design

Notice that row proportions are much easier to read and `ci` also show significant difference

Linear regression

data from <http://www.stata-press.com/data/r8/nhanes2e.dta>

USA data concerning high blood pressure

`svyset [pweight=leadwt], strata(stratid) psu(psuid)` ← setting sampling weight as `leadwt`, stratum variable as `stratid` and primary sampling unit as `psuid`

After setting survey variables the command `svyregress` is used in the exactly the same way as the command `regress` is used for nonsurvey data.

The above rule applies to all survey estimation commands.

`svyregress loglead age female black orace region2-region4` ← linear regression with
← `region2-region4 == region2 region3 region4`

`svyregress loglead age female black orace, subpop(region2)` ← fitting for a subset of data
- option `by()` not allowed with `svyregress`

`test region2 region3 region4` ← testing the hypothesis: `region2=region3=region4=0`

Similar test for linear hypothesis can be done for any survey estimation command. Testing has to be done immediately after fitting.

Logistic regression

data from <http://www.stata-press.com/data/r8/nhanes2d.dta>

USA data concerning blood pressure

`svyset [pweight=finalwgt], strata(strata) psu(psu)` ← setting survey variables

`svylogit highbp height weight age female black` ← fitting logistic regression
response: `highbp` coded as 1/0

`svylogit, or` ← getting output in form of odds ratios: option `OR`
executing `svylogit` only fits the last model

svylogit highbp height weight age female, or subpop(black) ← fit for subset of data
- option by() not allowed with svylogit

Other estimation commands currently available in Stata for survey data

svylogit ← ordered logistic regression
svyoprobit ← ordered probit regression
svymlogit ← multinomial logistic regression
svyologit ← ordered logistic regression
svynbreg ← negative binomial regression
.....

For the whole list check the “Stata Survey Data, Reference Manual, Release 8”

tadek.dyba@cancer.fi