Stata Introduction

Maria R. Munch, Ulrich Halekoh

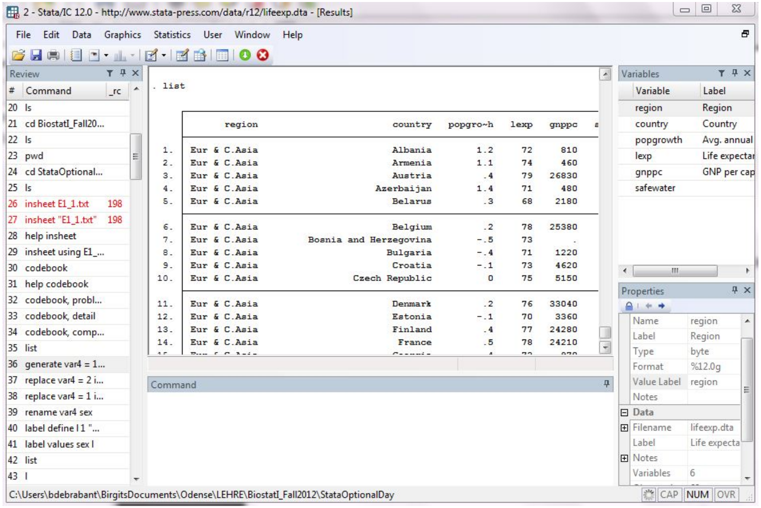
Epidemiology, Biostatistics and Biodemography; SDU

Revision 1

September 1, 2014

Outline

Stata-Environment



2 - Stata/IC 12.0 - http://www.stata-press.com/data/r12/lifeexp.dta - [Results]
File Edit Data Graphics Statistics User Window Help
Review
Variables
Properties
Command
CAP NUM OVR

#	Command	_rc
20	ls	
21	cd BiostatI_Fall20...	
22	ls	
23	pwd	
24	cd StataOptional...	
25	ls	
26	insheet E1_1.txt	198
27	insheet "E1_1.txt"	198
28	help insheet	
29	insheet using E1_...	
30	codebook	
31	help codebook	
32	codebook, probl...	
33	codebook, detail	
34	codebook, comp...	
35	list	
36	generate var4 = 1...	
37	replace var4 = 2 i...	
38	replace var4 = 1 i...	
39	rename var4 sex	
40	label define l 1 "...	
41	label values sex l	
42	list	
43	l	

```
. list

     +-----------------------------------------------------------------------
     |        region                      country   popgro~h   lexp   gnppc   s
     |-----------------------------------------------------------------------
  1. |  Eur & C.Asia                      Albania        1.2     72     810
  2. |  Eur & C.Asia                      Armenia        1.1     74     460
  3. |  Eur & C.Asia                      Austria         .4     79   26830
  4. |  Eur & C.Asia                   Azerbaijan        1.4     71     480
  5. |  Eur & C.Asia                      Belarus         .3     68    2180
     |-----------------------------------------------------------------------
  6. |  Eur & C.Asia                      Belgium         .2     78   25380
  7. |  Eur & C.Asia       Bosnia and Herzegovina        -.5     73       .
  8. |  Eur & C.Asia                     Bulgaria        -.4     71    1220
  9. |  Eur & C.Asia                      Croatia        -.1     73    4620
 10. |  Eur & C.Asia               Czech Republic          0     75    5150
     |-----------------------------------------------------------------------
 11. |  Eur & C.Asia                      Denmark         .2     76   33040
 12. |  Eur & C.Asia                      Estonia        -.1     70    3360
 13. |  Eur & C.Asia                      Finland         .4     77   24280
 14. |  Eur & C.Asia                       France         .5     78   24210
```

Variable	Label
region	Region
country	Country
popgrowth	Avg. annual
lexp	Life expectar
gnppc	GNP per cap
safewater	

Name	region
Label	Region
Type	byte
Format	%12.0g
Value Label	region
Notes	
Data	
Filename	lifeexp.dta
Label	Life expecta
Notes	
Variables	6

C:\Users\bdebrabant\BirgitsDocuments\Odense\LEHRE\BiostatI_Fall2012\StataOptionalDay

Stata has several windows

- Command You type commands like the logarithm of 100 and display

  ```
  display log(100)
  ```

  To execute the command press the ENTER-key
  To recall the last command type
  Windows: Page Up or Page Down
  Mac: Fn Up or Fn Down
  The result is shown in the:
- Result window
- the Review window holds the history of all your typed commands

Where am I? Your working directory!

Stata points to a certain directory if opened

```
pwd
```

```
C:/Users/uhalekoh/Documents
```

This directory is not the directory you may want to use for your work.
Create a directory for this purpose

```
mkdir c:/Users/UHHStata
```

Put yourself into this directory

```
cd   c:/users/UHHStata
```

Where am I? Your working directory!

Check whether you are really there

```
pwd
```

Find out the contents of the directory

```
ls
```

Stata- Basic work flow

Stata has the following basic philosophy:

- open some data to Stata session
- work on these data

Open data Stata provides on the web, e.g.the data `apple`

```
webuse apple
```

We list (=print) just the first two rows (=observations)

```
list in  1/2
```

Create a new variable that is the square of weight

```
generate weightQ= weight^2
```

We remove (=drop) weight from our data set

```
drop weight
```

Save/Read data in Stata format

Having done all this hard work; save the data onto your computer

```
save appleNy , replace
```

It should be saved as an `appleNy.dta` file.

Convince yourself: it's in your working directory

```
ls
```

If you want to read such a Stata data from this directory use

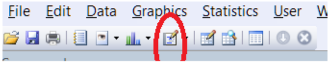
```
use appleNy , clear
```

Do-file

- Script file: keeps record of Stata commands
- allows to run sequence of commands several times
- default extension .do

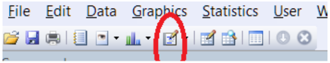
Example: Do-file

1. Open a new do-file via the task line



Example: Do-file

1. Open a new do-file via the task line

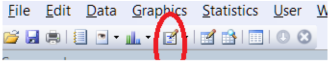


2. Type in the do-file the command

```
disp "Good Day"
```

Example: Do-file

1. Open a new do-file via the task line



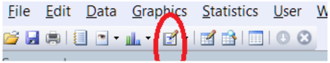
2. Type in the do-file the command

```
disp "Good Day"
```

3. Mark at least part of the line, and execute via CTRL D (on a Mac: SHIFT CMD D)

Example: Do-file

1. Open a new do-file via the task line



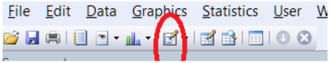
2. Type in the do-file the command

```
disp "Good Day"
```

3. Mark at least part of the line, and execute via CTRL D (on a Mac: SHIFT CMD D)
4. Run the whole file: CTRL A and CTRL D (on a Mac: CMD A and SHIFT CMD D)

Example: Do-file

1. Open a new do-file via the task line



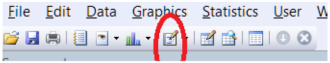
2. Type in the do-file the command

```
disp "Good Day"
```

3. Mark at least part of the line, and execute via CTRL D (on a Mac: SHIFT CMD D)
4. Run the whole file: CTRL A and CTRL D (on a Mac: CMD A and SHIFT CMD D)
5. Save via menu or CTRL S (give the file the name `sky`)

Example: Do-file

1. Open a new do-file via the task line



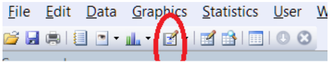
2. Type in the do-file the command

```
disp "Good Day"
```

3. Mark at least part of the line, and execute via CTRL D (on a Mac: SHIFT CMD D)
4. Run the whole file: CTRL A and CTRL D (on a Mac: CMD A and SHIFT CMD D)
5. Save via menu or CTRL S (give the file the name `sky`)
6. Close Stata!

Example: Do-file

1. Open a new do-file via the task line



2. Type in the do-file the command

```
disp "Good Day"
```

3. Mark at least part of the line, and execute via CTRL D (on a Mac: SHIFT CMD D)
4. Run the whole file: CTRL A and CTRL D (on a Mac: CMD A and SHIFT CMD D)
5. Save via menu or CTRL S (give the file the name `sky`)
6. Close Stata!
7. Double click on the file `sky.do`: Stata should open `sky.do`

Do-files: Comments and commands spanning several lines

- comments in do-files: * at the beginning of a line

```
* The preliminary analysis
```

or everything within /* and */

```
/* The preparation of the important
     part of my data   */
```

- commands over multiple lines: split by ///

```
display A B  ///
C D
```

Do-files: helpful key strokes

Using `do` file is much easier using the following key strokes

Task	Windows	Mac
mark all	CTRL A	CMD A
execute marked region	CTRL D	SHIFT CMD D
copy marked region to clipboard	CTRL C	CMD C
insert from clipboard	CTRL V	CMD V
delete marked region	CTRL X	CMD X
recover deletion	CTRL Z	CMD Z

Where to get help?

- `help drop`
- `search drop`
- Google
- Search for Stata -programs by others:

  `findit unitroot`

Exercise 1

The running example dataset for the course:

- Dataset is partly from Cropper C 1977 [1]. Honours thesis available at StatSci.org
- 48 stroke patients were randomly allocated to three treatment groups
- followed weekly for eight weeks
- recovery over time was measured using the Barthel Index and Goteburg Evaluation of Hemiplegia

[1] http://www.statsci.org/data/oz/stroke.html

Exercise 1

Table: The stroke data set `fakestrokeData.dta`

Variable	Description
subject	Subject ID, 1,. . . , 48
group	Experimental (E), pre-existing (F) or non-treatment (G)
sex	Male (M) or female (F)
side	Side of brain affected, left (L) or right (R)
age	Age of subject in years
lapse	Time lapse from stroke to start of program in weeks
ei	Upper extremities score (out of 36) at week i=1,. . . ,8
hwi	Hand-wrist score (out of 24) at week i
lei	Lower extremities score (out of 30) at week i
bali	Balance score (out of 14) at week i
barti	Barthel Index score (out of 100) at week i

Exercise 1

1. Make sure that the data file (`fakestrokeData.dta` delivered with the material for this day) is in your working directory
2. Open a new do-file
3. Read the data (= write the command in your do file and execute! Excute all the following Stata commands from the do file!)
4. Get a first description of all variables by executing

   describe

5. Keep only the variables `subject`, `age`, `bart1`, `group`, using **keep** (the list of variables must only be separated by blanks!)
6. Execute **codebook**

Exercise 1

7. Generate a new variable `kup` which is the square root of age. Use the $\boxed{\textbf{generate}}$ command and the $\boxed{\textbf{sqrt (X)}}$ $=\sqrt{X}$ function.
8. Make a simple summary of your data : $\boxed{\textbf{summarize}}$
9. Save the .do file and give it a reasonable name
10. Open it again and execute it.

Collecting output into a log-file

Save the output from all Stata commands into a log file:
First line of your .do file

```
log using "myOutput.log" , replace
```

Close the log file:
Last line of your .do file

```
log close
```

Exercise 1 (continuation)

11. Capture the output of your previously defined .do file in the file `results.log`.

Dataimport/-export: Excel Files

Import (reading) of Excel-file:

```
import excel mydata.xlsx , firstrow clear
```

Export (writing) of actual Stata file to an Excel-file:

```
export excel using  mydata.xlsx , firstrow(var) replace
```

For more information about import/export of Excel files see

```
help import excel
```

Exercise 2

Human beta-endorphin (HBE) is a hormone secreted under stress. For 10 patient measurements in two months are available in the Excel-file `hbe.xlsx`.

1. Make sure that the file is located in your working directory.
2. Read the file
3. Print the first 4 observations
4. Make a variable called `diff` that contains the difference between the May to the January measurements.
5. Save the actual Stata file as the Excel file `hbeNew.xlsx` and open it with Excel.
6. Save the actual Stata file as a Stata-data file.

Dataimport: ASCII (Text Files)*

- Tab separated (Tab symbolised here by ->)

```
name -> age
Ute    Hansen -> 77.3
Ib Ibsen ->  22.0
```

import delim using `mydataSpace.txt` **, clear**

- Space separated (make sure that variable entries are separated by exactly ONE space (blank))

```
name age
"Ute   Hansen" 77.3
"Ib Ibsen" 22.0
```

import delim using `mydataSpace.txt` **, clear delim(" ")**

Dataimport: ASCII (Text Files): csv-files*

- Comma separated (often these files have the extension `.csv`)

```
name,age
Ute   Hansen,77.3
Ib Ibsen,22.0
```

import delim using `mydata.csv`**, clear delim(",")**

Dataexport: write Stata files to csv files*

To write the actual Stata file say

```
input str4 animal weight
   dog      6.3
   cat      8.9
   whale        8
end
```

to the comma-separated file `animals.csv` do

```
export delim using animals.csv , replace
```

For more information about import/export of ASCII files see

```
help import delimited
```

Data manipulation

For the following we use the data `fakestrokeData.dta`.

```
use fakestrokeData , clear
```

Data manipulation- generate new variable

- Generate a new variable
```
gen ageD = int(age/10)
```
ageD contains the age decade values
- Change an existing variable (here: set to missing =.)
```
replace ageD = .  if ageD >= 6
```
- Rename an existing variable
```
rename  ageD age_decade
```
- Print first four observations for the two variables
```
list age  age_decade  in  1/4
```

Data manipulation-conditional change of values

Create a dichotomous variable from a continuous variable

- **generate** `bart1_ct = 0` **if** `bart1 <= 59`
- **replace** `bart1_ct = 1` **if** `bart1 >=60`

Warning: Stata saves missing values as a large value
need to add an additional condition to the if statement

- **replace** `bart1_ct = 1` **if** `bart1 >= 60` **& !missing(**`bart1`**)**

Data manipulation-add labels to categorical variables

For variables which

- take only few different values
- the values have no interpretation as numbers

it is helpful to add descriptive labels
Adding value labels to the binary bart1_ct
Define label variable `bartLab`

```
label define bartLab 0 "dependent"
```

```
label define bartLab 1 "independent" , add
```

Attach labels to the variable

```
label values  bart1_ct bartLab
```

Data manipulation- drop or keep

Drop or keep variables

- `drop age_decade` Drops the variable age_decade
- `keep subject sex age group bart bart1_ct` e

  Keeps only the mentioned variables in the data file

Drop or keep observations

- `drop if subject <= 3 | subject == 18`

  Drops first three subjects and subject 18 (| means in Stata OR)

- `keep in 5 /12`

  Keeps observations from observation number 5 to 12

Data manip.- Add summarizing variables to data file

```
egen avg_bart1 = mean(bart1)
```

Adds the overall mean of `bart1` to the data file
Important is the **by** option

```
egen avg_bart1= mean(bart1) , by(group)
```

Adds the group specific mean of `bart1` to the data file

Notice: the option is separated from the main command by a comma!

Exercise 3

1. Use the data `fakestrokeData.dta`. Keep only the variables `subject, age, sex, group`
2. Generate a variable `ageMed` that contains the overall median of age for all observations.
3. Generate a variable `ageDiff` that is the difference between age and the overall median.
4. Generate a variable `ageMed2` that is the median for each group.
5. Generate a variable `ageDiff2` that is the difference between age and the group-specific median for each group level.

Exercise 3 (continue)

6. Execute

   ```
   encode sex  , generate (sexNew )
   ```

   This will generate a new variables `sexNew` that is no longer a string, but a numeric variable with the levels 1 and 2. Convince yourself by executing

   ```
   codebook sex sexNew
   ```

7. Change the labels of `sexNew`, choosing 'female' for the numeric value 1 and 'male' for the value 2.
8. Create a new variable `groupN` which has the definition
   - groupN equals LOWER if group=E or group=F
   - groupN equals UPPER if group=G

   The OR is represented in Stata by |.

Exercise 3-continued

Stata has functions to make quick statistical summaries

9. Execute (only selecting people 70 years or older:)

```
use fakestrokeData, clear
keep subject age bart1 group
keep if age >= 70
```

10. Execute `describe`

11. Execute `summarize` . Why do you think you get reasonable information about age but not `group`?

    Execute `codebook age group` to find the answer.

Exercise 3-continued

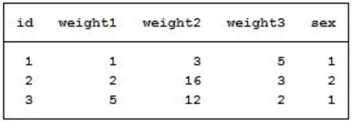
12. Execute `table age` . How many people are 72 years old?
13. Execute `tab age` . How many people are 70 years or older?
14. Execute `tab age group` . How many people are 72 years old and belong to group E?
15. Execute `tab2 age group , cell` . Which proportion of people that are 72 years old are in group E?

Reshaping Data

Generally, data exists in two formats: wide and long.

Assume we have measurements on j occasions for i subjects.

- wide: one line per subject id
  each occasion j is represented by a variable weight$j$

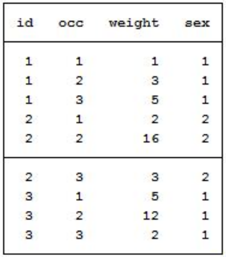


id	weight1	weight2	weight3	sex
1	1	3	5	1
2	2	16	3	2
3	5	12	2	1

```
reshape long weight, i(id) j(occ)
```

  Note: the variable occ will be newly generated for the long table.

- long: one line per occasion



id	occ	weight	sex
1	1	1	1
1	2	3	1
1	3	5	1
2	1	2	2
2	2	16	2
2	3	3	2
3	1	5	1
3	2	12	1
3	3	2	1

```
reshape wide weight, i(id) j(occ)
```

Exercise 4

Reduce the stroke data to three subjects with the Barthel indices for the first 5 weeks by executing

```
use fakestrokeData.dta, clear
keep subject bart1-bart5
keep if subject <=3
```

```
+---------------------------------------------------+
     | subject   bart1   bart2   bart3   bart4   bart5 |
     |--------------------------------------------------|
  1. |       1      45      45      45      45      80 |
  2. |       2      20      25      25      25      30 |
  3. |       3      50      50      55      70      70 |
     +--------------------------------------------------+
```

Exercise 4

Reshape the data into

```
+-----------------------+
     | subject   week   bart |
     |-----------------------|
  1. |       1      1     45 |
  2. |       1      2     45 |
  3. |       1      3     45 |
  4. |       1      4     45 |
  5. |       1      5     80 |
     |-----------------------|
  6. |       2      1     20 |
  7. |       2      2     25 |
  8. |       2      3     25 |
  9. |       2      4     25 |
 10. |       2      5     30 |
     |-----------------------|
 11. |       3      1     50 |
 12. |       3      2     50 |
 13. |       3      3     55 |
 14. |       3      4     70 |
 15. |       3      5     70 |
     +-----------------------+
```

Graphic

Stata is also a powerful graphic tool.
A graph is created by one command of sometimes considerable length.
Often one can easily produce a basic plot and then improve by using the inbuilt Graph Editor[2] on the graph.

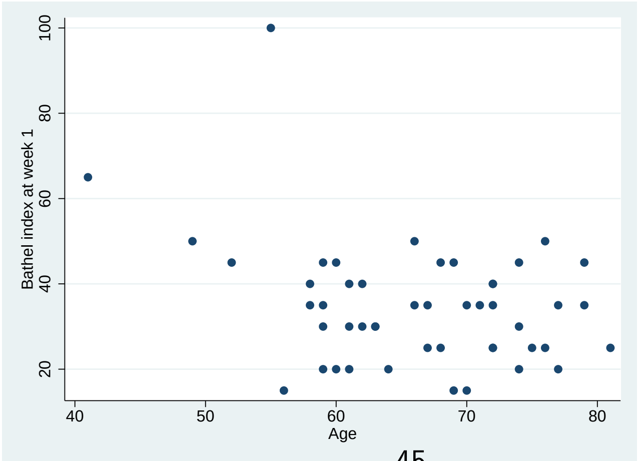
A comprehensive overview of graphics is
Mitchel, Michael N. (2012) *A Visual Guide to Stata Graphics*, 3rd Edition, Stata Press

[2] `http://www.stata.com/manuals13/gsu15.pdf`

Scatter plot

```
scatter bart1 age , ytitle( Bathel index at week 1)
```

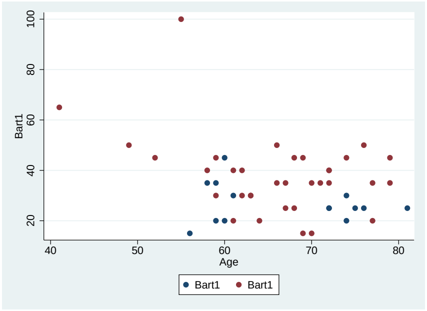
y axis variable (`bart1`) by x axis variable (`age`)
The `ytitle` is an option, separated from the main plotting command by a comma!



Scatter plot, differentiating between groups

```
twoway scatter bart1 age if sex == "F" ///
```

```
|| scatter bart1 age if sex == "M"
```



Scatter plot, differentiating between groups

In principle we overlay two scatterplots (one for males one for females)
Each plotting command separated by the || lines.
The legend with `Bart1` and `Bart1` is rather uninformative.
We can improve on this.

Scatter plot, differentiating between groups*

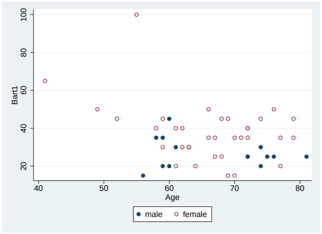
Different symbols to the two groups, improved labelling

```
twoway scatter bart1 age if sex == "F" ///
```

```
|| scatter bart1 age if sex == "M" ///
```

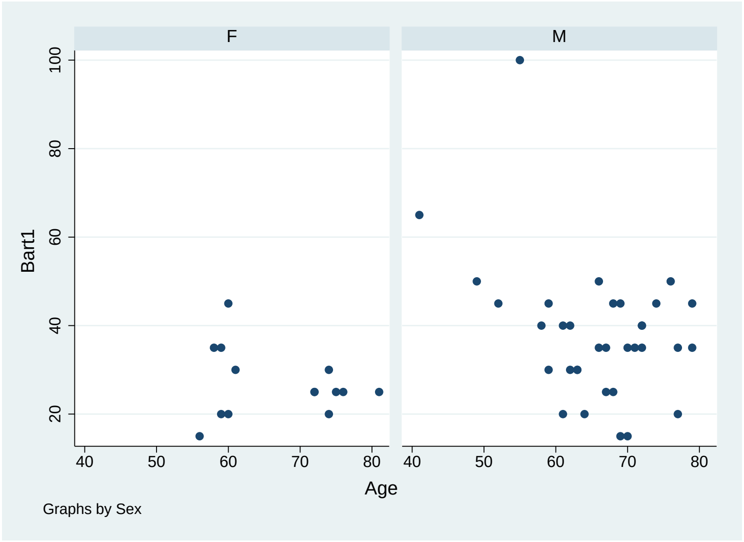
```
,msymbol( Oh t ) ///
```

```
legend(label ( 1 "male") label(2 "female") )
```



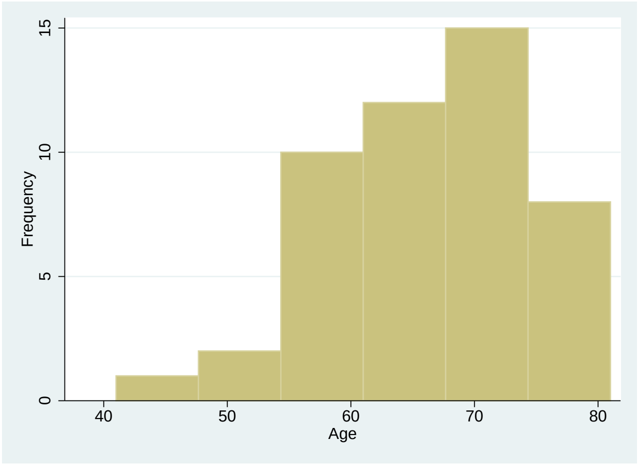
Scatter plot, side by side

```
twoway scatter bart1 age , by( sex )
```



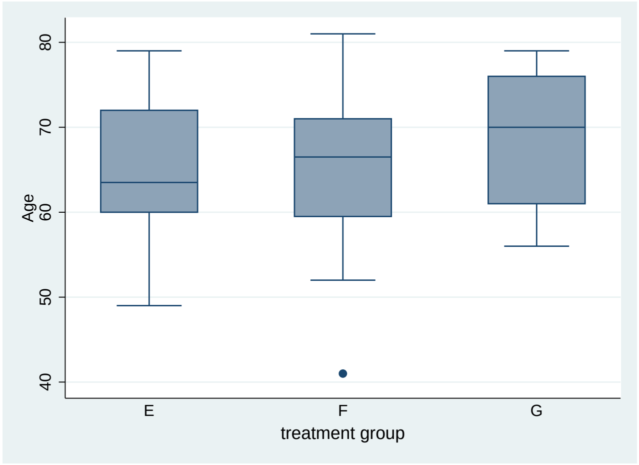
Histogram

```
histogram age , frequency
```



Boxplots

```
graph box age, over(group ) b1title(treatment group )
```



Stata graphs -saving

Graphs can simply be copied (Windows: CTRL+C) and then copied to a Word document (CTRL+V).
Graphs can be saved to files

- **graph save** `"boxplot.gph"`

  This format can be edited using Stata's graph editor (see `http://www.stata.com/manuals13/gsu15.pdf`)

Other possible formats: pdf, png, ...

- **graph export** `"boxplots.pdf"` **, replace**
- **graph export** `"boxplots.png"` **, replace**

Exercise 5

Use `fakestrokeData.dta`

1. Plot Barthel index in week 8 (`bart8`) against the Barthel index in week 1(`bart1`).
2. Using th Stata's graphic editor:
   2.1 Go to the graphic window and follow `File -> Stat Graph Editor`.
   2.2 Click on a plotted point and change its symbol, color or size.
   2.3 Save the plot as `bart.gph` with `File -> Save`.
   2.4 Close the graphic window.
   2.5 Open file `bart.gph` by clicking on its file name.
3. Same plot as before but differentiate between the three treatment groups (`group`).
4. Produce a separate plot for the relations of the two Barthel indices for each level of `group`.

Exercise 5

5. Make a boxplot of `bart1` and differentiate between males and females.
6. Make a boxplot of `bart1` and differentiate between males and females and the treatment group levels. (Use two **over()** statements.
7. Copy the graph into a Word document.
8. Save and export the last graph into a `.png` file and import it into a Word document.