

Introduction to Stata version 12.0

Preliminary short course for
readers of the book
“Regression Models in Medical Research”
written by Werner Vach and published by Chapman and Hall

Primrose Beryl
Clinical Epidemiology group
Institute of Medical Biometry and Medical Informatics
University Medical Center Freiburg
Freiburg, Germany

Werner Vach
Clinical Epidemiology group
Institute of Medical Biometry and Medical Informatics
University Medical Center Freiburg
Freiburg, Germany

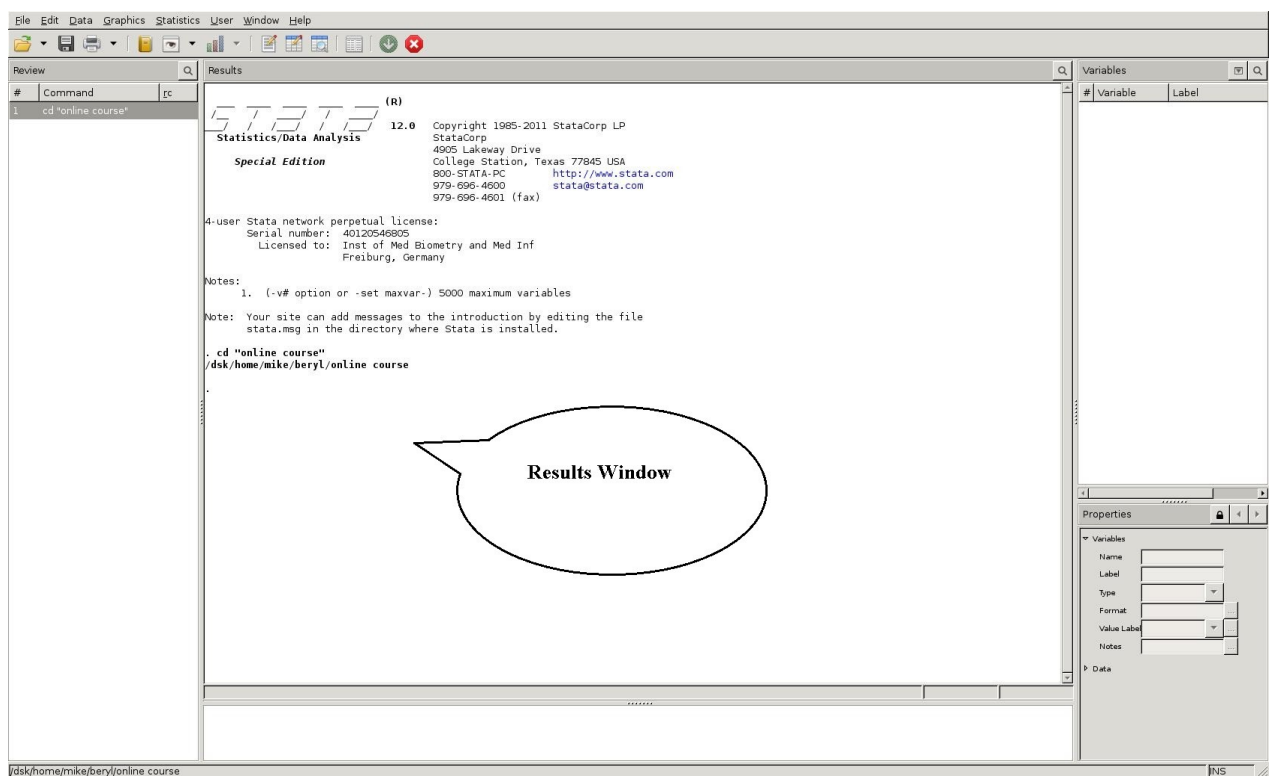
0 First steps in STATA

To start with, let us open STATA. When you open STATA, you will find that there are a number of windows within STATA. Let us first introduce you to two of the windows which you are going to use at the moment

- The “command” window is where you can enter your commands



- The “results” window is where you can see the output for your command



Now let us start using it. You can either type in the commands or click on the menu system. And during the whole period of our course, we will be concentrating on the syntax commands which you can type into the command window.

The first step would be to make sure that you are at the right place where your data is and where you want to store your output files – your do-files and log files.

Therefore we would like you to list the names of files in the current directory in which you are at the moment and for that, you type `dir` in the command window,

```
. dir
```

Please note that there is a dot in front of the command in the above line. This is to show how it appears on the output window. But when you type the command, you should not type the dot in front of the command. It applies to all the commands displayed throughout the course material.

You will see the output similar to the following depending on where you are in your computer and what files are there in the current directory

```
3.1k 7/01/06 13:51 auto.dta
1.6k 6/04/06 10:40 cancer.dta
3.8k 5/06/09 17:06 census.dta
0.5k 4/21/07 11:39 med1.dta
```

If you are in the right place, you will find our dataset among the list

```
9.5k 3/01/10 12:00 numberofwords.dta
```

and to know the name of the path of the current directory, you type `pwd`,

```
. pwd
```

and the output would be something like

```
C:/RMMR
```

In case, you do not find yourself at the right directory, please navigate yourself to the right place. You can use `cd ..` (`cd` space dot dot) to come out of the subdirectory you are in.

```
. cd ..
```

and then use

```
. cd subdirectory
```

until you are in the right directory. You can also specify the whole pathname, e.g.,

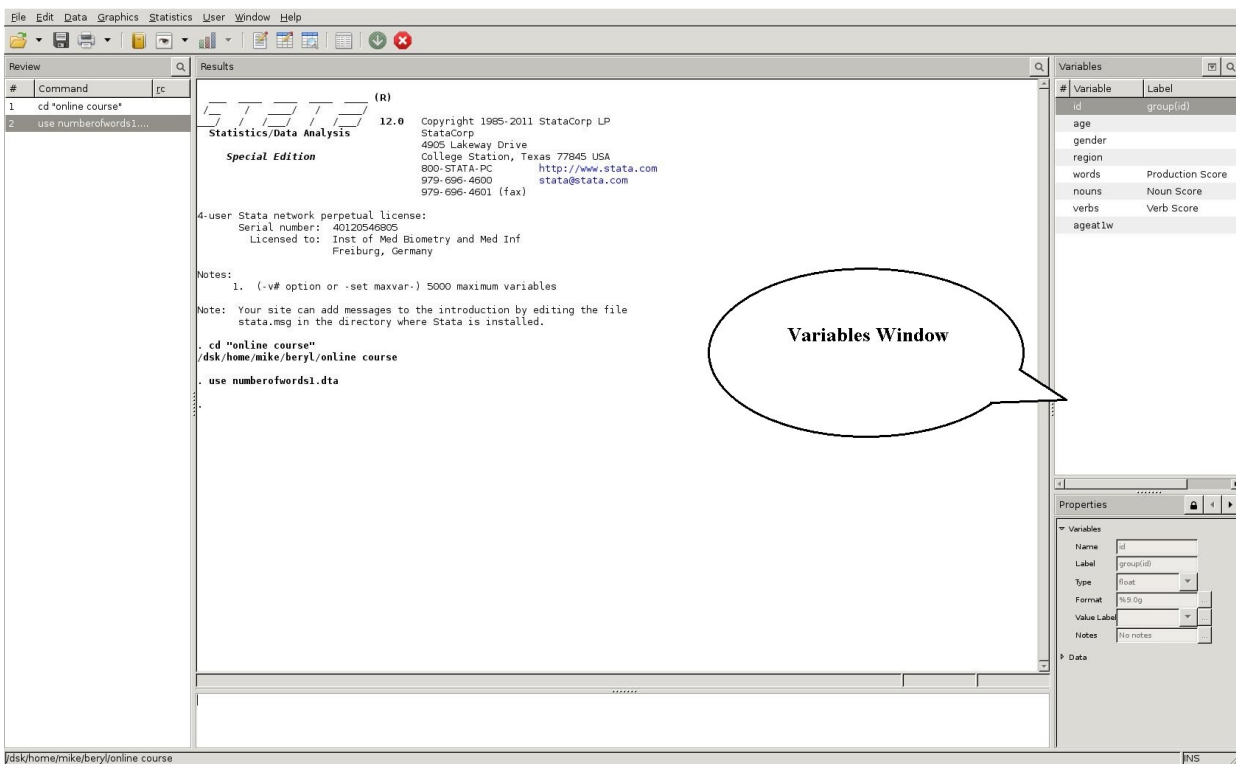
```
. cd "C:\My Documents\RMMR"
```

The pathname is in quotes as it includes a blank in it. Now that you have arrived and stationed yourself at the right place, we can start to work in STATA and we have to first specify a dataset.

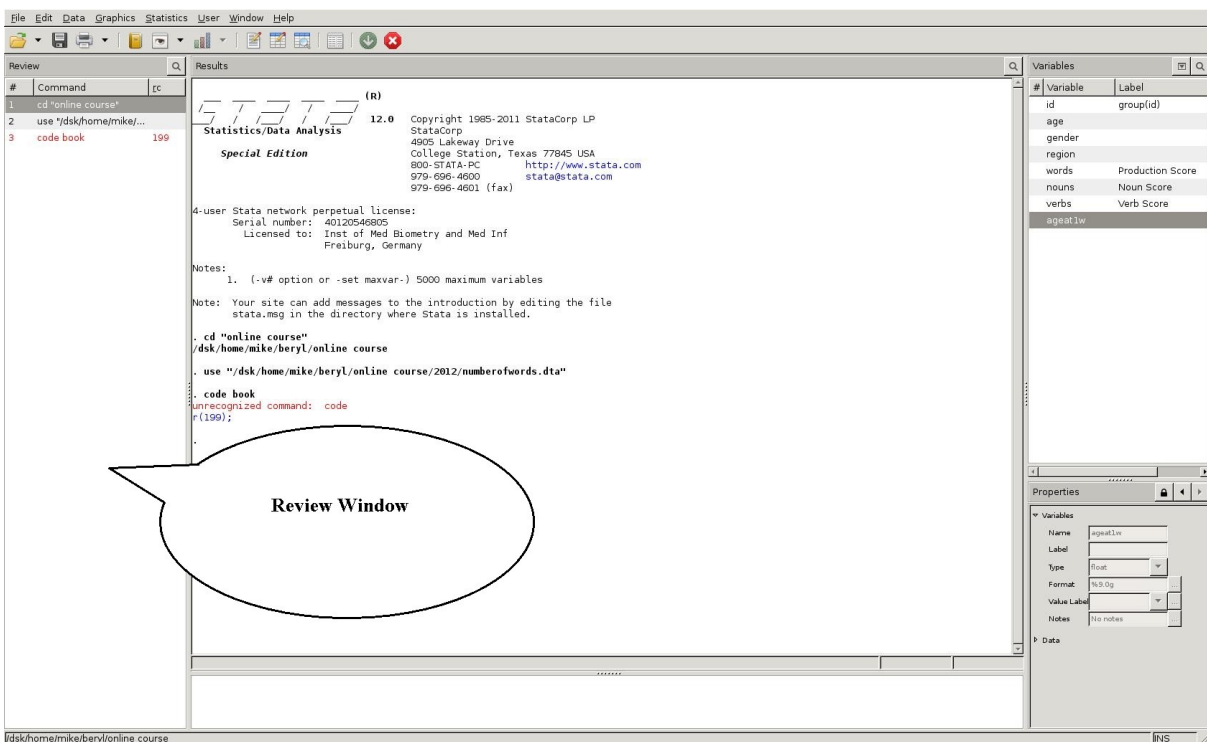
The dataset we are going to use is already in STATA format and is `numberofwords.dta`. To open the dataset, type in the command window

```
. use numberofwords
```

Now you will be able to see the variables appearing in one of the windows. This is the “variables” window which lists all the variables that are in your dataset.



And again, you will find that all the commands you have entered and run are being listed in a window which is the “review” window. By clicking on the command in the review window, you can execute the same command again. And in case, there was an error in the command you typed, then it will be highlighted in red.



1 The basic commands

To have an idea of how the data looks, you can take a look at the first 10 observations using the `list` command:

```
. list in 1/10
```

```

+-----+
| id   age   gender  region  words  nouns  verbs  ageat1w |
+-----+
1. | 1    20    male   urban   34    18     0     18 |
2. | 2    20    male   rural   19     2     0     18 |
3. | 3    20    female urban   40    11     2     18 |
4. | 4    20    female urban  540   292    81    12 |
5. | 5    20    male   urban   34    14     4     18 |
+-----+
6. | 6    20    male   urban   36    16     1     18 |
7. | 7    20    male   rural   58    26     3     17 |
8. | 8    20    female rural   23     3     0     18 |
9. | 9    18    female urban   35    11     3     16 |
10. | 10   18    male   urban  133    91     4     14 |
+-----+

```

The `list` command just lists the observations.

This data is based on a study, in which parents were asked to fill out a check list with several hundreds of words. For each word the parents had to mark whether their child has already started to say this word. The marks for each child were counted and this can be found in the variables `words`, `nouns` and `verbs` as the corresponding counts for these word types. The parents were also asked at which age their child has produced the first word and this information is found in the variable `ageat1w`. If `words` is equal to 0, then `ageat1w` is missing as the parents cannot know this age. Background information on the children's age, gender and the type of region they come from, is given in the corresponding variables. Only children of age 16, 18 and 20 months are included in this dataset.

We can use the `codebook` command to obtain further information on the variables

```
. codebook id age gender region words
```

```

-----
id                                                                 group(id)
-----
                                type:  numeric (float)
                                range:  [1,268]
                                unique values: 268
                                mean:    134.5
                                std. dev: 77.5091
                                percentiles:
                                10%      25%      50%      75%      90%
                                27      67.5    134.5    201.5    242
-----
age                                                                 (unlabeled)
-----
                                type:  numeric (float)
                                range:  [16,20]
                                units:  1

```

```
unique values: 3                               missing .: 0/268
```

```
tabulation:  Freq.  Value
              91   16
              79   18
              98   20
```

```
-----
gender                                                    (unlabeled)
-----
```

```
type: numeric (byte)
label: sex1

range: [1,2]                                units: 1
unique values: 2                            missing .: 0/268
```

```
tabulation:  Freq.  Numeric  Label
              125     1     male
              143     2     female
```

```
-----
region                                                    (unlabeled)
-----
```

```
type: string (str5)

unique values: 2                            missing "": 0/268
```

```
tabulation:  Freq.  Value
              157   "rural"
              111   "urban"
```

```
-----
words                                                    Production Score
-----
```

```
type: numeric (float)

range: [0,644]                                units: 1
unique values: 133                            missing .: 0/268

mean: 76.2836
std. dev: 105.359

percentiles:  10%    25%    50%    75%    90%
              7     17     35     88.5  201
```

You can see that the above output is quite informative, as you can know what type of variable it is, the summary - range, mean, standard deviation, percentiles (for numeric variables) and frequency (string variables), total count, count of unique values and missing values. Note that the variables also can have a label. For example, the variable *words* has the label "Production score" and this label will be used in all the outputs by STATA (for example, while using the variable in a graph). Numeric values and their label values are also displayed for variables which are labelled, e.g., for *gender*, 1 is for male and 2 is for female.

Now, if you would like to know, how many boys and girls are in the dataset, you can use the command `tabulate` which can be abbreviated as `tab`

```
. tab gender
```

gender	Freq.	Percent	Cum.
male	125	46.64	46.64
female	143	53.36	100.00
Total	268	100.00	

You will be able to view the percentages as well as the cumulative percentages alongside.

The next step would be to see how many boys and girls of age 16, 18 and 20 are there in our dataset.

```
. tab age gender
```

age	gender		Total
	male	female	
16	41	50	91
18	36	43	79
20	48	50	98
Total	125	143	268

This would provide you with the raw numbers but if you need to know whether the ratio of boys to girls is similar in all the age groups, you have to look at the relative frequencies within each row of the table. We can tell STATA this by using the `row` option:

```
. tab age gender, row
```

```
+-----+
| Key          |
|-----|
| frequency    |
| row percentage |
+-----+
```

age	gender		Total
	male	female	
16	41	50	91
	45.05	54.95	100.00
18	36	43	79
	45.57	54.43	100.00
20	48	50	98
	48.98	51.02	100.00
Total	125	143	268
	46.64	53.36	100.00

The row percentage is a proportion of the column variable in the row variable. e.g., among the children who are 16 months old, 54.95% were girls. But if you would like to compare the age distribution between boys and girls, you have to look at the relative frequencies within each column. We can tell STATA this by using the `column` option:

```
. tab age gender, col
```

```

+-----+
| Key |
+-----+
| frequency |
| column percentage |
+-----+

```

age	gender		Total
	male	female	
16	41	50	91
	32.80	34.97	33.96
18	36	43	79
	28.80	30.07	29.48
20	48	50	98
	38.40	34.97	36.57
Total	125	143	268
	100.00	100.00	100.00

And here the explanation would be as follows: Of all the girls, 34.97% are 16 months old. Of course, you can interchange the variables which you want to form rows and columns:

```
. tab gender age, row
```

```

+-----+
| Key |
+-----+
| frequency |
| row percentage |
+-----+

```

gender	age			Total
	16	18	20	
male	41	36	48	125
	32.80	28.80	38.40	100.00
female	50	43	50	143
	34.97	30.07	34.97	100.00
Total	91	79	98	268
	33.96	29.48	36.57	100.00

The most important thing about percentages in the table is to be able to interpret them in the right way.

Often, we would like to have such tabulations within subgroups. STATA allows restricting the application of a command by using an `if` construct prior to the options. So if you are interested in the age/gender distribution among children from the urban region, you can type:

```
. tab age gender if region=="urban", row
```

```

+-----+
| Key |
+-----+

```

```

| frequency |
| row percentage |
+-----+

```

age	gender		Total
	male	female	
16	20	18	38
	52.63	47.37	100.00
18	20	10	30
	66.67	33.33	100.00
20	21	22	43
	48.84	51.16	100.00
Total	61	50	111
	54.95	45.05	100.00

You can see that the table has been done for the 111 children living in urban region alone.

But if you would like to know this distribution within children living in the urban and children living in rural regions, you can use a “`bysort:`” construct to tell STATA to create the table for both the subpopulations:

```
. bysort region: tab age gender, row
```

```
-----
-> region = rural
```

```

+-----+
| Key |
|-----|
| frequency |
| row percentage |
+-----+

```

age	gender		Total
	male	female	
16	21	32	53
	39.62	60.38	100.00
18	16	33	49
	32.65	67.35	100.00
20	27	28	55
	49.09	50.91	100.00
Total	64	93	157
	40.76	59.24	100.00

```
-----
-> region = urban
```

```

+-----+
| Key |
|-----|
| frequency |
| row percentage |
+-----+

```

age	gender		Total
	male	female	
16	20	18	38
	52.63	47.37	100.00
18	20	10	30
	66.67	33.33	100.00
20	21	22	43
	48.84	51.16	100.00
Total	61	50	111
	54.95	45.05	100.00

This construct is quite helpful as you can work on subgroups of data with a single command.

Now you should know enough to perform the first exercise, i.e., to try to find answers to the following questions using STATA:

Exercise 1

1. How many children of ages 16, 18 and 20 months, are in the data set?
2. How many children belonging to the rural region are of ages 16, 18 and 20 months?
3. Is there a difference in the age distribution between children belonging to the urban and rural regions?
4. What are the variables you find in the dataset `raw.dta`?

2 The graphical user interface

To become familiar with the syntax of STATA, you can also use STATA's graphical user interface (GUI). At the top of the STATA Window you can find a menu with the items "Data", "Graphics" and "Statistics". Here you can pick up certain commands and specify the variables and the options in an interactive dialogue. At the end STATA creates the command for you, and you can see it in the Results window and the Review window.

3 Simple tables and graphs with STATA

We continue using the dataset `numberofwords.dta`

```
. use numberofwords.dta, clear
```

Here, you will notice that the option `clear` is used and `clear` specifies that it is okay to replace the data in memory even though there are unsaved changes made in the already open data set. If there has been no changes in the current data set, then `clear` is not necessary.

Now that the study is about the language skills and development of children, you would like to know about the number of words spoken by the children. One can try to describe the distribution of a continuous variable like `words` by some statistics like the mean and the standard deviation. The `summarize` command provides you with some basic numbers:

```
. summarize words
```

Variable	Obs	Mean	Std. Dev.	Min	Max
words	268	76.28358	105.3587	0	644

However, it is possible to tell STATA exactly, which statistics you want using the `tabstat` command :

```
. tabstat words, s(n mean median p10 p90)
```

variable	N	mean	p50	p10	p90
words	268	76.28358	35	7	201

You can, of course, use `by` option to look into subgroups:

```
. tabstat words, s(n mean median p10 p90) by(gender)
```

```
Summary for variables: words  
by categories of: gender
```

gender	N	mean	p50	p10	p90
male	125	80.752	34	7	227
female	143	72.37762	39	8	195
Total	268	76.28358	35	7	201

or else, the `bysort`: construct as shown below.

```
. bysort gender: tabstat words, s(n mean median p10 p90)
```

```
-> gender = male
```

variable	N	mean	p50	p10	p90
words	125	80.752	34	7	227

```
-----  
-> gender = female
```

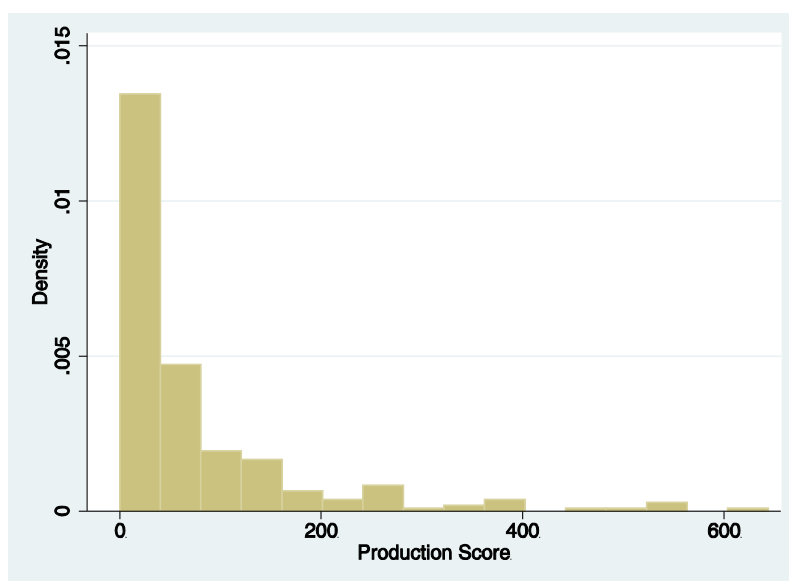
variable	N	mean	p50	p10	p90
words	143	72.37762	39	8	195

```
-----
```

However, pictures speak more than numbers. Several graphical techniques are available to visualise the distribution of a continuous variable.

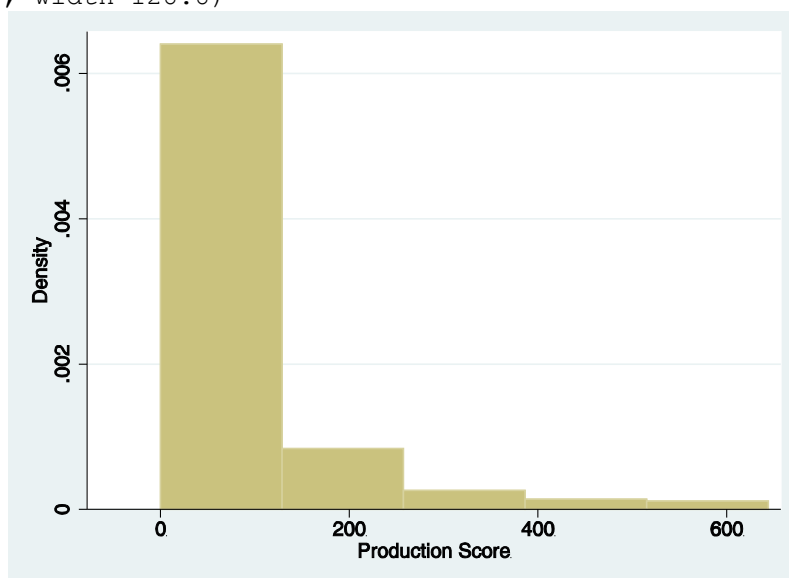
We can start with a histogram

```
. histogram words  
(bin=16, start=0, width=40.25)
```



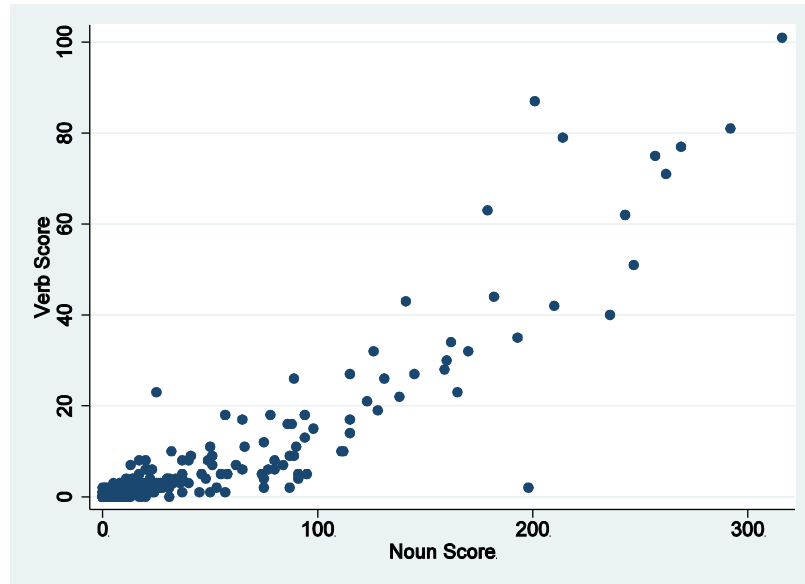
You can see that most of the children still have a small vocabulary. In a histogram, you can also specify the number of bars you would like to see by specifying the `bin()` option.

```
. histogram words, bin(5)  
(bin=5, start=0, width=128.8)
```



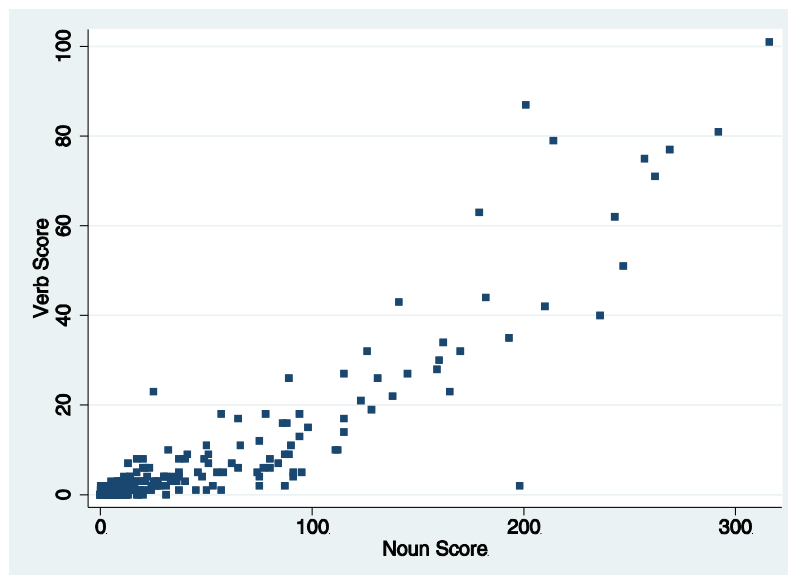
Scatterplots are very useful in studying correlation between two continuous variables. In our study, we would like to visualize the relationship between the number of nouns the children were able to speak and the number of verbs. Many of the graph commands of STATA require starting with the `graph` command but for a scatterplot, you just use the `scatter` command.

```
. scatter verbs nouns
```



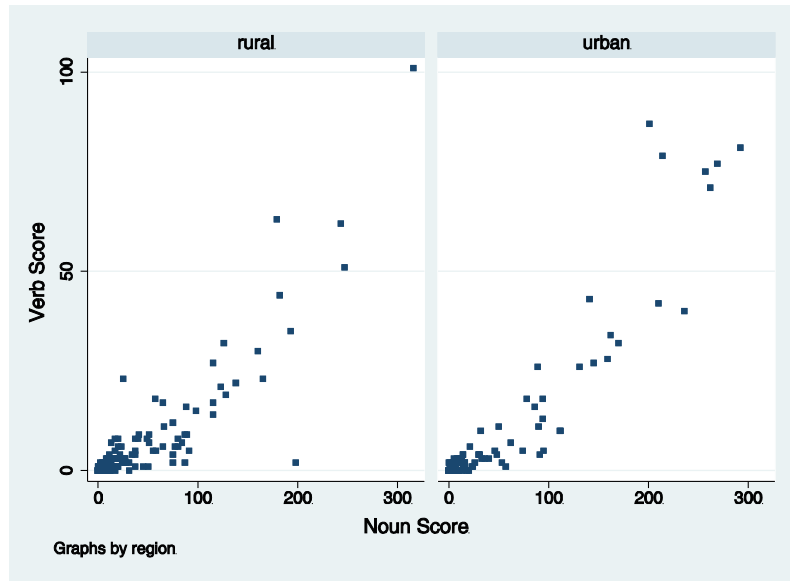
From the above scatterplot, we can see that there is a very high correlation between the number of verbs and the number of nouns. Here, you can assign any specific symbol for the markers as well as the size of it, by using the `msym()` and `msize()` option. If you want the marker to be a small square, then we type

```
. scatter verbs nouns, msize(small) msym(S)
```



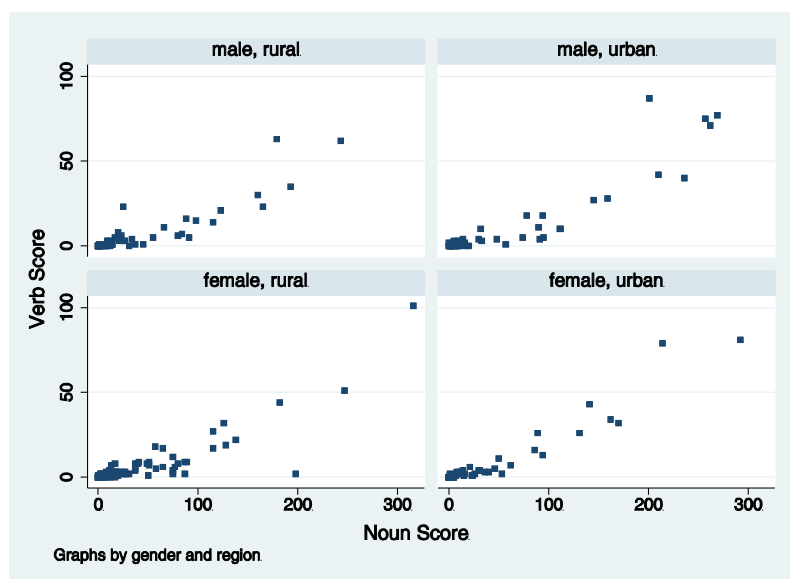
You can find the different symbol styles and marker sizes available by typing `help scatter` or by clicking (in the menu system) on Graphics>Twoway graph>create>marker properties. If you want to see whether the relationship between the number of nouns and verbs are different in children from urban and rural regions, you can use the `by` option for this task:

```
. scatter verbs nouns, msize(small) ms(S) by(region)
```



You find that the pattern is quite similar among the rural and the urban regions. You can also use more than one variable in the `by` option to stratify the scatter plot by many variables:

```
. scatter verbs nouns, msize(small) ms(S) by(gender region)
```



There exist a lot of other options to manipulate or improve a graph and to create other types of graphs which we will be dealing with later.

However, you should be able to use STATA to make some simple graphs by now.

Exercise 3:

1. Describe the distribution of the number of verbs acquired with the help of a graph.
2. Graphically represent how the number of nouns varies with the number of words?
3. How would you visually clarify that the relationship of nouns and verbs are not different between boys and girls?

4 The general syntax of STATA commands

Syntax

The general syntax of a command is

```
commandname varlist selector, options
```

`varlist` can be one or several variable names, or a command can be without any variable too.

`selector` can be something like

```
if sex=="m"  
if age>18  
in 1/3
```

Note: the “equal to” symbol is two times “=”

`options` vary from command to command. They are either single names (e.g. `row`, `col`) or include additional information in parentheses (e.g. `by(age)`, `bin(5)`, `xlabel(0 10 20 30)`)

Note: There is at most one comma in a STATA command!

There exist some commands, which allow an extension of this syntax. Nearly all commands allow to be preceded by a `bysort var:` construction, which tells STATA to apply the command in each stratum defined by the values of `var`.

Abbreviations

Usually you can abbreviate command-names and options. For example, the following two commands are equivalent:

```
. bysort region: tabulate age gender, row column exact  
. bys region: tab age gender, r co e
```

Each option and command has a minimal number of letters to be used, you can look up this using the `help` command. The minimal number of letters is underlined by STATA. You can also abbreviate variable names by their first letter(s), as long as the identification remains unique.

Varlists

In the case of several variables, it is possible to give the `varlist` as, say, `var1-var5`, which means all the variables from `var1` to `var5` in the current order, or you may use `var*`, which means all the variables in the dataset that start with the letters "`var`". There exist also further possibilities to abbreviate a list of variables, just try `help varlist`.

Exercise 4:

1. What is the minimal typing in order to obtain a scatterplot of the number of nouns by words, stratified by age ?

5 Generating new variables

Often one would like to create new variables based on existing variables. The `generate` command allows to generate such new variables. The general syntax of the `generate` command is

```
gen newvar=expr
```

where `expr` is a combination of variables and numbers with operators like `+` or `<` and functions like `sin(varname)` and `newvar` is a name you can choose for the new variable formed.

Let us look at some examples:

We would like to express the number of verbs as a relative frequency with respect to the total number of words spoken:

```
. use numberofwords.dta, clear  
. gen fracverbs=verbs/words
```

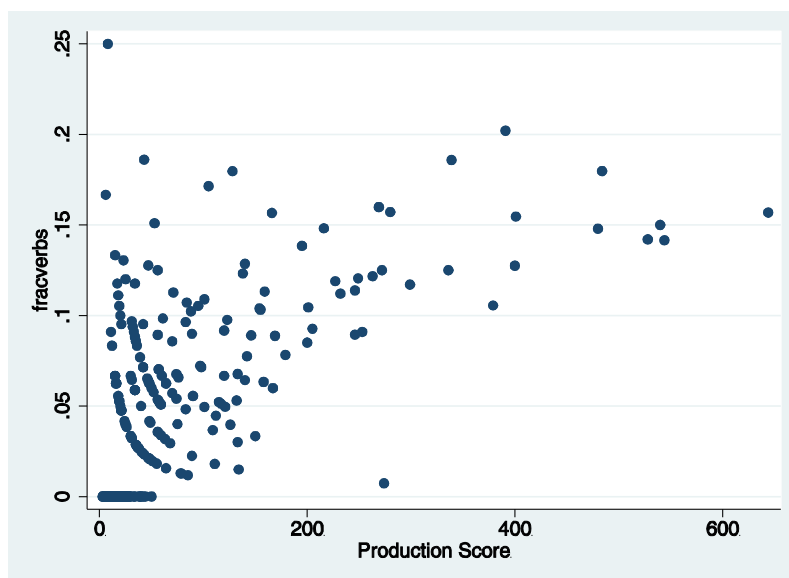
Let us have a look at the new variable created,

```
. list id words verbs fracverbs in 1/10
```

```
+-----+  
| id   words   verbs   fracverbs |  
+-----+  
1. | 1     34      0      0 |  
2. | 2     19      0      0 |  
3. | 3     40      2     .05 |  
4. | 4    540     81    .15 |  
5. | 5     34      4    .1176471 |  
+-----+  
6. | 6     36      1    .0277778 |  
7. | 7     58      3    .0517241 |  
8. | 8     23      0      0 |  
9. | 9     35      3    .0857143 |  
10. | 10    133     4    .0300752 |  
+-----+
```

Now we can analyse the new variable `fracverbs` like any other variable in the data set:

```
. scatter fracverbs words
```



In the above plot, you can find a very interesting pattern where the fractions of verbs tend to increase with the vocabulary size and reaches a plateau at about 15%. We can also try to form an index of the nouns i.e. we can look at the proportion of nouns of all the words spoken:

```
. gen fracnouns=(nouns/words)*100
. list id words nouns fracnouns in 1/10
```

	id	words	nouns	fracno~s
1.	1	34	18	52.94118
2.	2	19	2	10.52632
3.	3	40	11	27.5
4.	4	540	292	54.07407
5.	5	34	14	41.17647
6.	6	36	16	44.44444
7.	7	58	26	44.82759
8.	8	23	3	13.04348
9.	9	35	11	31.42857
10.	10	133	91	68.42105

Now we can take a look on how this quantity changes with the number of words in the urban and rural children:

```
. scatter fracnouns words, by(region)
```



You can see that there is not much difference between the children living in the urban and rural regions.

Moving on with the analysis, we would like to classify the children based on the number of words they could speak. For example, we may regard a child as poor in language skills development if the child could speak only fewer than 25 words.

```
. gen poor=words<25
. list id word poor in 1/10
```

```

+-----+
| id  words  poor |
+-----+
1. | 1    34    0 |
2. | 2    19    1 |
3. | 3    40    0 |
4. | 4   540    0 |
5. | 5    34    0 |
+-----+
6. | 6    36    0 |
7. | 7    58    0 |
8. | 8    23    1 |
9. | 9    35    0 |
10. | 10  133    0 |
+-----+

```

Having done that, you would be curious to know whether the proportion of “poor” children is different among boys and girls and in each age group.

```
. bys age: tab gender poor, row
```

```
-> age = 16
```

```

+-----+
| Key          |
+-----+
| frequency    |
| row percentage |
+-----+

```

gender	poor		Total
	0	1	
male	15 36.59	26 63.41	41 100.00
female	13 26.00	37 74.00	50 100.00
Total	28 30.77	63 69.23	91 100.00

```
-> age = 18
```

```

+-----+
| Key          |
+-----+
| frequency    |
| row percentage |
+-----+

```

gender	poor		Total
	0	1	
male	23 63.89	13 36.11	36 100.00
female		31	12
	43		

	72.09	27.91	100.00
Total	54	25	79
	68.35	31.65	100.00

-> age = 20

```
+-----+
| Key      |
|-----|
| frequency|
| row percentage|
+-----+
```

gender	poor		Total
	0	1	
male	40	8	48
	83.33	16.67	100.00
female	43	7	50
	86.00	14.00	100.00
Total	83	15	98
	84.69	15.31	100.00

Let us say that you later found out or decided that the cut-off for “poor” is not 25 but 20 words. And if you want to overwrite the existing variable, you cannot use `generate`, but you have to use the `replace` command:

```
. gen poor=words<20
poor already defined
r(110);

. replace poor=words<20
```

An overview about all allowed operators and functions can be obtained using `help`:

```
. help operators
. help functions
```

Exercise 5

1. Children with a vocabulary size above 400 may be classified as “fast”. Can you find a difference between boys and girls with respect to being “fast”?
2. Graphically represent the distribution of the noun/verb ratio separately for children of each age group.

6 Getting help

Often you forget the syntax of a command or the name of an option. Then you can use the `help` command to obtain information about a command. In the following example, we obtain information for the `list` command:

```
. help list
```

You can also use the `help` command to refresh your knowledge about certain topics of STATA:

```
. help language
```

```
. help varlist
```

If you want to know more about the commands, details of how it is implemented in particular, you can click on the menu Help>Pdf documentation, then you can access the whole help information as a pdf document. If you have forgotten a command or if you are in doubt, whether a specific technique is available in STATA, you can try to `search` for it. For example you might want to know, whether you can use a Kruskal-Wallis test with STATA. So you type

```
. search kruskal
```

Keyword search

```
Keywords: kruskal
```

```
Search: (1) Official help files, FAQs, Examples, SJs, and STBs
```

Search of official help files, FAQs, Examples, SJs, and STBs

```
[R] kwallis . . . . . Kruskal-Wallis equality-of-populations rank test  
(help kwallis)
```

```
[MV] mds postestimation Postestimation tools for mds, mdsmat, and mdslong  
(help mds postestimation)
```

```
FAQ . . . . . What statistical analysis should I use?  
. . . . . UCLA Academic Technology Services  
5/08 http://www.ats.ucla.edu/stat/STATA/whatstat/whatstat.htm
```

```
Example . . . . . Applied Linear Statistical Models  
. . . . . UCLA Academic Technology Services  
4/08 examples from the book Applied Linear Statistical Models  
by Neter, Kutner, et.al.  
http://www.ats.ucla.edu/stat/STATA/examples/alsm/
```

(end of search)

So the `kwallis` command satisfies your needs.

The search command covers not only the STATA commands, but also the entries in the STATA Technical Bulletin, a collection of user-written commands.

For example, you may be interested in Friedman test, a non parametric procedure for repeated measurements and would like to know how to carry it out in STATA. Hence you search for it using

```
. search friedman
```

```
Keyword search
```

```
Keywords:  friedman
Search:    (1) Official help files, FAQs, Examples, SJs, and STBs
```

```
Search of official help files, FAQs, Examples, SJs, and STBs
```

```
FAQ      . . . . . What statistical analysis should I use?
          . . . . . UCLA Academic Technology Services
          5/08    http://www.ats.ucla.edu/stat/STATA/whatstat/whatstat.htm
```

```
SJ-9-2   st0167  . . Skillings-Mack test (Friedman test when data are missing)
          (help skilmack if installed) . . . . . M. Chatfield and A. Mander
          Q2/09   SJ 9(2):299--305
          provides the Skillings-Mack test, a general Friedman-type
          test that can be used in almost any block design with an
          arbitrary missing-data structure
```

```
SJ-6-4   st0113  . Testing for cross-sectional dependence in panel-data models
          (help xtcsd if installed) . . . . . R. E. De Hoyos and V. Sarafidis
          Q4/06   SJ 6(4):482--496
          tests for the presence of cross-sectional dependence in
          panels with many cross-sectional units and few time-series
          observations
```

```
SJ-5-2   snp2_1  . . . . . Software update for friedman
          (help friedman, genvsum, kendall if installed) . . . . . R. Goldstein
          Q2/05   SJ 5(2):285
          bug fix for friedman
```

```
STB-3    snp2    . . . . Friedman's ANOVA & Kendall's coefficient of concordance
          (help friedman if installed) . . . . . R. Goldstein
          9/91    pp.26--27; STB Reprints Vol 1, pp.157--158
```

```
(end of search)
```

So you can see that one of the references to the Friedman test is available from Volume 3 of the STB which abbreviated “STATA Technical bulletin”. If the computer you are working at is connected to the internet, then you can download the material using STATA’s `findit` command:

```
. findit friedman
```

You have just to click on ”snp2” and follow the instructions to download the program. Afterwards, if you now want to know how to use the command, you can use

```
. help friedman
```

Note: `findit` covers all entries of the World Wide Web about STATA, and hence you get more results than with `search`. `search` covers only the official sources like the STB and the STATA

journal.

Exercise 6

1. Find out what the **keep** command is for.
2. Find out whether STATA offers a Pearson correlation coefficient with confidence intervals and try to install a command for this task and compute a confidence interval for the correlation between **words** and **nouns**

7 Working with do-files and log-files

In the long run, it can be very boring to type all commands again and again. Often, you would have to do the same steps again and again. Then it is wise to work with do-files.

Do-files are ordinary textfiles with the suffix ".do", which include STATA commands. You can create the do-files by any text editor producing textfiles, or you can use the do-file editor of STATA which can be opened from the Menu by clicking on Window > Do-file editor> New do-file editor.

Once you have created a do-file, you can execute it by typing

```
do dofilename
```

We have provided a do-file "scatter.do" as an example on the course directory and you can download the do-file so that you can work with it. You can view the content of the do-file using type command

```
. type scatter.do  
  
use numberofwords, clear  
tab age  
tab gender  
scatter verbs nouns, by(age)
```

And execution of the do-file looks like this:

```
. do scatter  
  
. tab age
```

age	Freq.	Percent	Cum.
16	91	33.96	33.96
18	79	29.48	63.43
20	98	36.57	100.00
Total	268	100.00	

```
. tab gender
```

gender	Freq.	Percent	Cum.
male	125	46.64	46.64
female	143	53.36	100.00
Total	268	100.00	

```
. scatter verbs nouns, by(age)  
  
. end of do-file
```

You can also load this file into the do-file editor, manipulate it, and execute it again.

You may also be interested to take results home or to exchange results with other people. This is possible by creating log files, where all output to the result window is stored. This may look like:

```
log using filename, text replace
commands
log close
```

You can also use the log-file button under the “File” menu for this.

By default, log-files are created in a special format called `smcl`. You can look at such files using the `view` command, and then it looks like the original Results window. However, they look quite distorted, if you try to look at them outside STATA. The `text` option creates log-files as ordinary text-files which can be read by anyone.

Many people use STATA in a way, such that each do-file creates also a log file. So the do file `scatter2.do` looks like this one:

```
log using scatter2, text replace
*** Creating tables and scatterplots ***
use numberofwords, clear
tab age
tab gender
scatter verbs nouns, by(age)
log close
```

and after its execution

```
do scatter2
```

we have a log file `scatter2.log` with the following contents:

```
-----
      name: <unnamed>
      log:  /dsk/home/mike/beryl/online course/2012/scatter2.log
      log type: text
      opened on: 30 Jan 2012, 11:53:19

. *** Creating tables and scatterplots ***
. use numberofwords, clear

. tab age

      age |          Freq.      Percent      Cum.
-----+-----
      16 |             91       33.96      33.96
      18 |             79       29.48      63.43
      20 |             98       36.57     100.00
-----+-----
      Total |             268     100.00

. tab gender

      gender |          Freq.      Percent      Cum.
-----+-----
```

male		125	46.64	46.64
female		143	53.36	100.00

Total		268	100.00	

```
. scatter verbs nouns, by(age)

. log close
  name: <unnamed>
  log: /dsk/home/mike/beryl/online course/2012/scatter2.log
  log type: text
  closed on: 30 Jan 2012, 11:53:21
-----
```

Note that you can include comments in the do-file just by starting a line with an * mark.

Exercise 7

1. Create a do file, which creates a log file including a cross tabulation of age and gender for the *numberofwords* dataset.

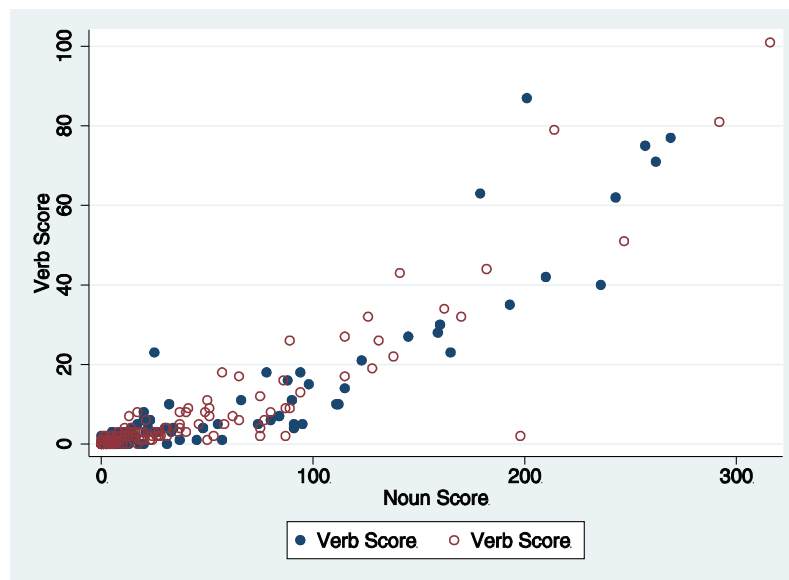
8 More about STATA graphs

There are instances when presenting your research results, that you would like to differentiate two or more categories of a variable e.g. region, gender, etc, on the same plot. This is possible by combining two graph commands which assign different marker symbols to the different categories.

For example, in the scatter plot between number of verbs versus number of nouns, if you would like to differentiate between boys and girls, the command would be

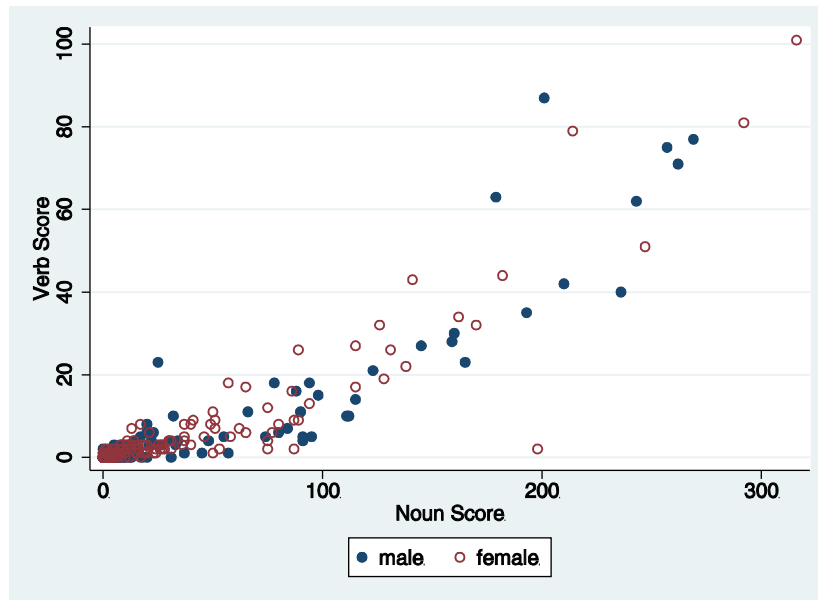
```
. scatter verbs nouns if gender==1, msym(O) ||scatter verbs nouns if gender==2,  
msym(Oh)
```

Here, you can see that these are two STATA commands where we have asked for a scatter plot for male children with the marker symbol as circle and another scatterplot for the female children with the marker symbol as hollow circle, which has been linked by “||”.



In the above graph, you find that the legends just display the variable name and thus are not self explanatory. You can add a better explanation for the two symbols by manipulating the legend using the `legend_option`. Type `help legend_option` for more information.

```
. scatter verbs nouns if gender==1, msym(O) ||scatter verbs nouns if gender==2,  
msym(Oh) leg(lab(1 "male") lab(2 "female"))
```



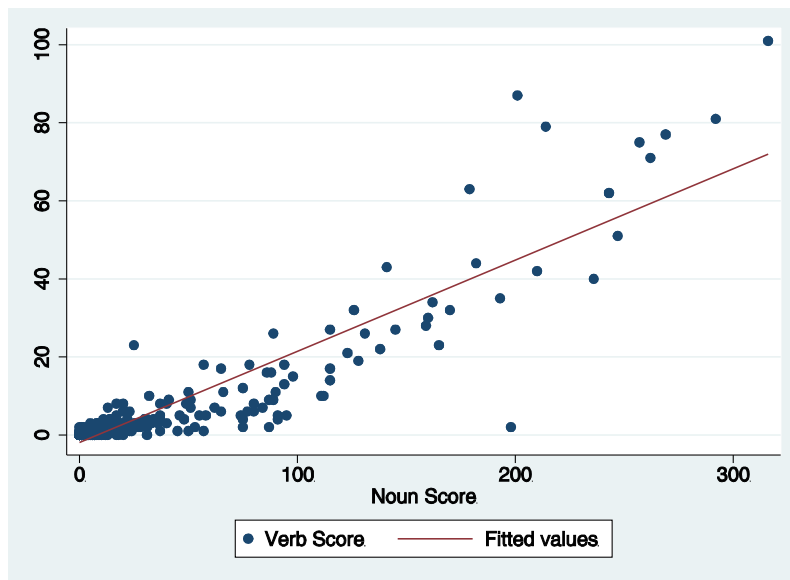
In case, you want to create the above graph for all the three age groups, then you can use the `by` option at the end.

```
. sc verbs nouns if gender==1, msym(0) ||sc verbs nouns if gender==2, msym(0h)
by(age) leg(lab(1 "male") lab(2 "female"))
```



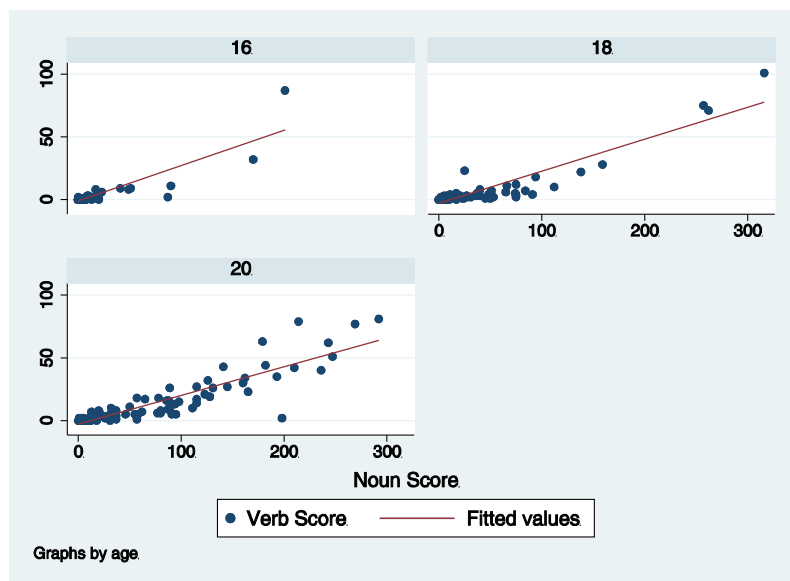
Generally, when we are using regression models, we would like to represent the scatter plot along with a fitted regression line. In that case, we use a similar construction except that we combine a command for scatter plot and a command for the fitted line.

```
. . scatter verbs nouns ||fit verbs nouns
```



Here too, you can use the `by` option.

```
. scatter verbs nouns ||lfit verbs nouns, by(age)
```



In the do-file, there may be instances when a single command is too long and you would like to split it into two lines. You can do it using `"/"`. Note that there must be a space before `///`. For example,

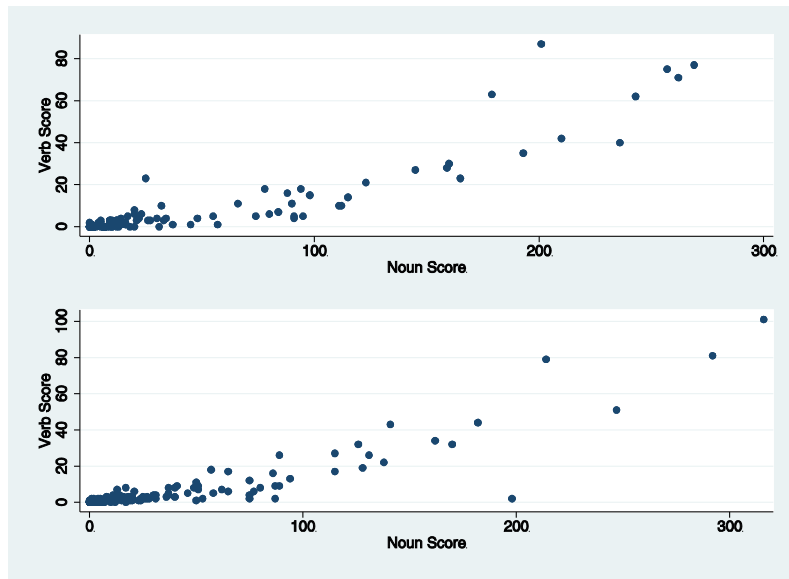
```
scatter verbs nouns if gender==1, msym(0) msize(small) ||scatter verbs nouns
if /// gender==2, by(age)msym(0h) msize(medium)
```

In STATA, you can also combine graphs using the `graph combine` command. `graph combine` arrays separately drawn graphs into one. Let us say that you made two scatter plots of verbs versus nouns – one for boys and another for girls by using the two commands below

```
. scatter verbs nouns if gender==1, saving(male, replace)
. scatter verbs nouns if gender==2, saving(female, replace)
```

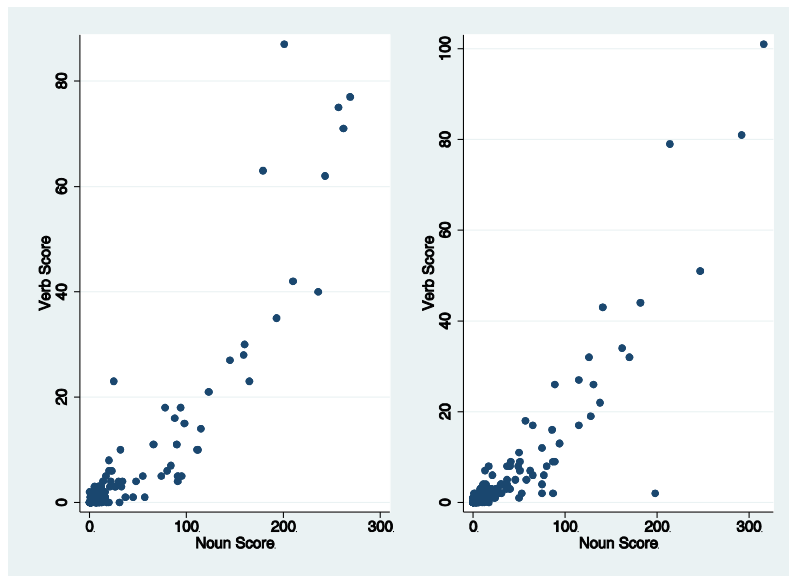
Now if you feel that it would be better if the graphs are presented together as one graph, then you can combine them by just typing

```
. graph combine male.gph female.gph, col(1)
```



You can use the `row` or `col` option to mention your preference of alignment.

```
. graph combine male.gph female.gph, row(1)
```



Exercise 8

1. Fit a line for the scatter plot between words and nouns.
2. Differentiate between the different groups of children based on age in the scatter plot between nouns and verbs (don't forget the legends).

9 Analysing survival data

In survival time analysis or more generally, while analysing time to event data, we are interested in time to a particular event, for example, time from diagnosis of cancer till death, time from treatment till recurrence of a disease, etc. A survival dataset consists of a time variable which measures the survival time and a failure variable which records whether the event of interest has taken place by the end of the observation period. Subjects for whom the event has not yet taken place at the end of the observation period are censored at that time-point.

Before you analyse survival data in STATA, you have to set the data so that STATA can recognise the key variables by using `stset` command. `stset` declares the data in memory to be `st` data, informing STATA of key variables and their roles in a survival-time analysis. It is important that you `stset` the data before using any of the `st` commands. The command is as follows:

```
. stset timevar, fail(failvar[==numlist])
```

In our dataset here, we would like to analyse time to first word, ie, the age at which the child spoke the first word. Here, the time variable is `ageat1w` and we can see that ten children have not yet started to talk and we need to censor them at the time of the interview. First we have to create a variable to show whether the child started to talk or not.

```
. gen starttalk=words>0
```

For the ease of censoring, we create a new time variable where the time to the first word is available for all the children along with the ten censored children for whom the time from birth till the interview is used.

```
. gen agefw=ageat1w  
. replace agefw=age if agefw==.
```

Now we can `stset` the data using

```
. stset agefw, fail(starttalk==1)
```

The `fail` option tells STATA that the children with `starttalk==1` have experienced an event (often called “failure” in survival analysis) and that the remaining children are censored.

The output will be

```
failure event:  starttalk == 1  
obs. time interval:  (0, agefw]  
exit on or before:  failure
```

```
-----  
268 total obs.  
0 exclusions  
-----
```

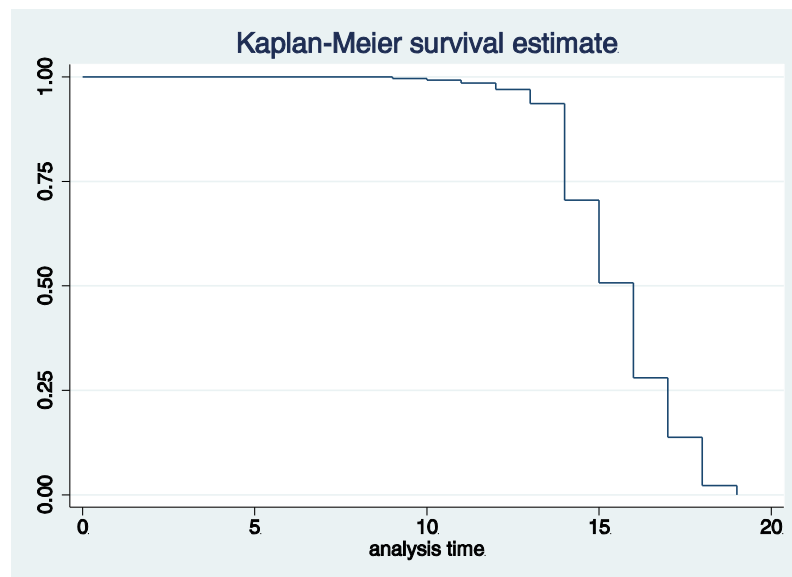
```
268 obs. remaining, representing  
258 failures in single record/single failure data  
4147 total analysis time at risk, at risk from t = 0  
earliest observed entry t = 0  
last observed exit t = 19
```

From the above output, you can check yourselves whether you have set the data correctly. It shows that all the 268 children have been included and 258 failures had occurred.

Now that we have informed STATA that we are using survival data, we can go ahead to analyse it. In this session, we will explain how you can draw a survival graph (Kaplan Meier graph) and how to do a log rank test to compare two survival curves.

A survival graph has the survival time at its X axis and the survival probability at its Y axis. In our example here, X axis represents the time until the first word and Y axis represents the probability not to have started to speak. And any point in the curve reflects the probability that a child has not yet spoken the first word at a specific time-point. To obtain a survival graph, you use the `sts` command and you can see how the children in this study has started to talk over time.

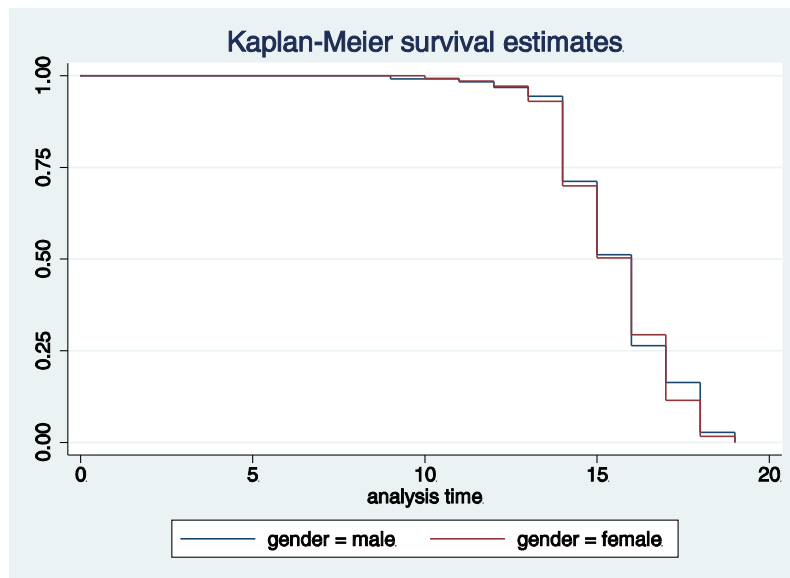
```
. sts graph
```



From the graph, we can read that until the age of 9 months, no child had started to talk and by the age of 16 months, half of the children have started to talk.

You can also obtain the survival graphs for subgroups. If you want to study the time to first word separately for boys and girls, then you can use

```
. sts graph, by(gender)
```



After obtaining the two survival graphs, you may want to know whether the time to first word among boys and girls are statistically different. `sts test` command analyses the equality of survivor functions across two or more groups using logrank test. It can be obtained by

```
. sts test gender
```

And the output will look like

```
failure _d: starttalk
analysis time _t: agefw
```

Log-rank test for equality of survivor functions

gender	Events observed	Events expected
male	121	123.64
female	137	134.36
Total	258	258.00

chi2(1) = 0.18
Pr>chi2 = 0.6727

where the chi square value and the p value will help you to ascertain the statistical significance of the difference.

Exercise 9

1. Compare the distribution of time to first word among children belonging to the urban and rural region.

10 Data management

Now that you are familiar with all the basic commands required for the course, we would like to introduce few other basic commands which will be useful for data management using STATA. For all the commands we introduce below, there are much more usefulness to it as they have more options and we would suggest you to get to know it using the help menu.

Sorting

The sorting of a dataset in STATA is not fixed. You can change the sorting of the datasets by commands like `sort` in addition to the command which we have introduced earlier - the `bysort:` construct. If you would like that the records have to be sorted in an ascending order of the `id` variable, then you type

```
. sort id
```

Eliminating

If you would like to delete certain variables or records, you can do so using `drop` command. If you want to drop the variable `nouns` or the records of the children for whom `words` are missing, you use:

```
. drop nouns                                or  
. drop if words==.
```

You can use the `keep` command as well for the sake of convenience. For example, you would like your data to have only three variables including the `id`, `age` and `words` or drop the records of the children for whom number of words are missing, then you type:

```
. keep id age words                          or  
. keep if words!=.
```

One has to be careful with these commands as you may not be able to retrieve the records which you have dropped unless you had saved the data.

Recoding

One of the usual ways for creating a new variable by grouping is to use `recode` command. Suppose you want to group the number of words spoken into 5 groups (≥ 20 , 21-40, 41-80, 81-120, >120) and you want to create a new variable `wordsgp` with 5 values, then you say,

```
. recode words (0/20=1) (21/40=2) (41/80=3) (81/120=4) (121/max=5), gen(wordsgp)
```

Encoding and Decoding

If you have a string variable and want to change it to a numerical variable, you can use the `encode` command:

```
. encode region, gen(region1)
```

And if you want to convert a numeric labelled variable to a string variable, you can use:

```
. decode region1, gen(region2)
```

Saving a file

Now you want to save the file after having created some changes or new variables as a new file, then you can use `save` command. For example,

```
. save numberofwords1.dta
```

You would use the option `replace` if you were going to overwrite an existing file with the same name. You have to be sure that you want to overwrite the existing file as you may lose important data if you had deleted some records or variables.

```
. save numberofwords1.dta, replace
```

Pocket calculator

If you are in need for a pocket calculator while using STATA, you can just use the command `display` or `di` in short. It just evaluates the expression you type after the command name. If you use a variable name in the expression, STATA uses the value of this variable in the first observation. Try this out:

```
. di (556*34)/5      or
```

```
. di nouns/words
```