Clinical research I. - observational studies

Stata

26 Feb 2019

Table of Contents

Introduction

Today I'm going to introduce to the statistical program called Stata as we are going to use it during the practices. Therefore I will use only this program today. However, I won't use all the functions of Stata, only those I find useful and interesting for your studies and of course, we have time for it. Now I start typing after the word *display* in the bottom of the screen - that is the command line, wher we could give commands. So I typed into the command line:

```
. display "Why should we use a statistical software at all?"
Why should we use a statistical software at all?
```

As we can see it, the command and the result of it has appeared in the results window in the middle of the screen.

```
The answer in my opinion:

1. So we don't need to use our fingers for calculations - we can do calculations easily
```

This easy counting is assisted by the “clicking” tabs part of the Stata. So let’s click…

```
2. Being able to use what others have already discovered and proved

3. The counting and it's result has to be reproduceable by us and others as well
```

This latter one is important and needed some further explanation.

For being able to repeat our calculations, we have to know what datebase we used, how we calculated and why and how we drew *inferences* etc. - so for all these we need to “documentate our results” in a way which is understandable for others too, so they can repeat and reproduce it.

This documentation is assisted by the *Do - file Editor*.

Documentation

Why is documentation beneficial?

- reproduceability
- noticing mistakes, miscalculations is easier
- we can prove that “we did not cheat”

Furthermore, if we pay a little more attention how we make it and devote a little time to use *dynamic documentation*:

- If there is a change, we can evaluate again only using some “clicks”
- we can avoid mistakes from copying
- the program can provide again the refreshed graphs and charts
- we can refer to the results of the analyses, so **the analysis and the documentation appears together** - so we can show it to others in a well understandable way (and we can understand later ehat we have done)

For example a word document could be created on what I am writing now and what we calculating today - comments and calculation together in a nice way.

In order to make you capable of writitng such a dynamic document, I am writing the documentation of my lecture in this form too.(Certainly, there are more amazing possibilities within dynamic documentation that I will not reveal - but you can do it yourself.)

Guidance

Put the comments into

new line /***

new line blalba

new line ***/

structure - it is visible that these parts have a different colour in the do file

If we would like to see something in the next line in our dynamic document, then we can do it with **leaving an empty line**

It means that if I write something in the following line, it won't be in the new line in the document. (I need it for the practical reson to be able to see it "in front of me" on the screen in do file.)

The commands - eg.: 2*3= ? - we write in a new line: (this lines called code chunks)

```
    . display 2*3
    6
```

Let's see what we get if we use the do file and also create a dynamic document from it with a *db markdoc* command (Before that don't forget to save the do file) We can even run only one line of the do file by marking it or copying it as a command

The followings could be observed in the previous code:

- the text will be **bold** if it is put between **(asterics)
- the text will be in *italics* if it is put between _ _ (underscore)
- titles can be made by # (hashtag) - subtitles by ##
- lists can be made by +, -, * etc. (line spacing should not be forgotten! before starting the list)
- putting /**/ before the command will show only the result but not the command
- putting /***/ before the command will show only the command and not the result
- if you write your comment after * into a command code chunk, it will only appear in the *.do file* but not in the dynamic document

Importing, reading data

Before we can import data let's look the source. In the medical practice - unfortunately - excel tables are used and wrongly used … let's see wah.xlsx.

In this form there is no statistical software that could handle it!!!

How should the dataset be that we can use:

- the file should contain only the data (if it is necessary short comments in a variable could be included), NO additional calculations, longer notes
- use a given language (if it will be used for a paper, poster in a conference… use english)
- do NOT include special characters(@!"*+-$#/őéá…)
- all data have to be in a given sheet (in excel), NO more sheet for the same dataset
- do NOT create empty rows or columns
- do NOT merge cells
- put 1 data in 1 cell
- rows have to contain "cases" and variables have to be in columns (1 variable in 1 column and NOT 1 outcome in 1 column - see wah file 3rd sheet…)
- 1 data type for 1 variable - eg. do NOT mix tex and numbers - drop this case/data or create new variable
- use "days/hours… passes" from a certain point instead of dates (using dates could be very problematic
- clearly indicate that the data is 0 or missing (or forgotten to enter …) - the missing data can be program-dependent, the stata prefers the point (.)
- for numeric variables, pay attention to the decimator - stata likes the point (.)
- the name of the ketegorical variables should be short, lowercase is recommended, be careful when typing (mistyping and space after the word could be problematic)
- always have a variable for patient identification (ID) - do NOT use the name of patients (ethics…)
- the names of our variable should be short, clear and informative
- do NOT categorize the variables - use the highest possible scale!
- can be coloring, but it is completely unnecessary

Import, read data

To avoid having to click so much, set up your workbook from where you are open, save files or graphs, drawings:*File->Change working directory…*.

```
. cd "C:\pendrivok\oktatos\klinkut\kurzus1\sajat_ea\stata_ea\en\"
C:\pendrivok\oktatos\klinkut\kurzus1\sajat_ea\stata_ea\en
```

There are several ways to read data:

- copy-paste - it is simple but we could not put it into the documentation - not reproduceable
- import - we can use several file format
- open - if dataset in stata file format (*.dta) exist

We see that the data set names and properties appear on the right. Delete the current data set and read (*import*) the new one. Here we leave only the part after the work directory path in the code.

```
. clear

. import excel "frmgham_ea.xls", sheet("frmgham2") firstrow
```

Preparing data

We can view the data in "edit" and "browsing" mode using the appropriate icons (_Data Editor_s in the middle), or using *Data->Data Editor* tab.

Next to these icons we can find the *Variables manager*(*Data->Variables manager*), with that we can more easily review the variables, set the type, or give description of the data eg. adding a unit to the variable(*Label*). It is always a good idea to start our examinations to check our variables this way. Eg. we can realize if we have text in our numeric variables - the type of variable will be text. Furthermore, let's see what variables we have at all to appear. If we make a change, we can save the corresponding command into the *.do file*.

```
. label variable RANDID "RANDID blalb"

. format %10,1f SYSBP
```

And after this preparations we can save our data(eg. keeping comma as decimal separator). *File->Save as*

```
. save "frmgham_ea1.dta", replace
file frmgham_ea1.dta saved
```

The replace in the end of the command means to overwrite the file if exist

Delete our dataset (*clear* command) then load the saved file. (*File->Open*)

```
. clear

. use "frmgham_ea1.dta"
```

In many cases, we have to convert a categorical text variable into a "numerical" one called *factor* variable so that we can use it (factor variable: assigns number to the text categories). For example, in regressions (see later lectures, practices) this is an important situation.

```
. capture logsitic AGE i.SEX2
```

We can do the necessary change at _Data->Create or change data->Other variable-transformation commands ->encode value labels form string variable option. Let's make a factor variable called GENDER from the categorical variable SEX2. Then let's see what is we got using a command or in the *Data editor*.

```
. encode SEX2, generate(GENDER)

. label list GENDER
GENDER:
           1 female
           2 male
```

We could observe in the Data Editor that factor variable is blue. We can see the corresponding numerical value if we click on it.

If we want to convert a categorical variable encoded with numbers into a factor (this is because we would like to see the category names in the analyzes), this can be done in 2 steps: first we create the number-text matching, then we assign it to the variable. We can do it in the *Variable Manager* that creates the next codes:

```
      . label define SEX 1 "female" 2 "male"

      . label values SEX SEX
```

If we would like we can change the value of a variable using recode and replace. (*Data -> Create or change data -> Other variable-transformation commands -> Recode categorical variable*; or *Data -> Create or change data -> Change contents of variable*), eg.:

```
      . recode HYPERTEN (1=2) (0=1)
      (HYPERTEN: 199 changes made)

      . replace SEX2 ="fe" if SEX2 == "female"
      (118 real changes made)
```

It is more common that we would like to create a new variable based on previous variables, eg. let's create the arterial mean pressure (MAP).

```
      . generate MAP=(SYSBP +2*DIABP)/3
```

If we don't need a variable we can delete it using *drop*.

```
      . drop MAP
```

Some other things I like to use:

- the command *cls* clears the result window
- we don't need to type the full name of a variable: pushing *TAB* helps us, or double click on the name of the variable in the right side (*Variables* side)
- for most commands it is enough to enter the first few letters (eg. gen instead of generate)
- we can recall the previous commands with pushing *ctrl+r* or pgup/pgdn
- if I know a command (I found it in the Net) but I don't know the menu option for it we can use *help_stata command* to find it (the description of the data gives the tab path too)

- of course the help is very usefull anyway

```
       . cls

       . gen MAP=(SYSBP +2*DIABP)/3
```

Descriptive statistics

Finally, we have reached the point of describing our dataset to be analysed. We can find several options for that in *Statistics->Summaries tables and tests -> Summary and descriptive statistics->Summary statistics*.

We could make a summary description on all variables.

```
. summarize

    Variable |        Obs        Mean    Std. Dev.       Min        Max
-------------+---------------------------------------------------------
      RANDID |        199    230622.9    142231.7       2448     477082
         SEX |        199    1.592965    .4925206          1          2
        SEX2 |          0
     TOTCHOL |        197    236.7919    38.92342        150        332
         AGE |        199    49.72864    9.054087         35         67
-------------+---------------------------------------------------------
       SYSBP |        199    133.5829    22.40636         96        212
       DIABP |        199    82.92462    11.86687         60      124.5
    CURSMOKE |        199    .4924623    .5012041          0          1
     CIGPDAY |        197    9.213198    12.19214          0         43
         BMI |        198    26.20869     4.66816      16.75       45.8
-------------+---------------------------------------------------------
    DIABETES |        199    .0301508    .1714333          0          1
      BPMEDS |        197    .0304569    .1722785          0          1
     GLUCOSE |        189    82.52381    23.14813         45        225
      ANGINA |        199    .1708543    .3773308          0          1
    HYPERTEN |        199    1.763819    .4258058          1          2
-------------+---------------------------------------------------------
      GENDER |        199    1.407035    .4925206          1          2
         MAP |        199    99.81072    14.51487   73.33334   146.6667
```

Always review the results in details and critically. Notice for example:

- although the program calculates the mean, standard deviation… for our factor variables but usually it has no meaning!
- we can detect that the maximum of BMI is impossibly high - suspect in this situation that the decimal separator is in wrong position. Let's look at our ORIGINAL data in this case, and if this is really the case, fix it (but only if we sure that it was really a mistyping). Now I put this mistake on purpose at the 5th case of BMI :).

```
 . replace BMI = 29.43 in 5
 (1 real change made)
```

We could make description only on a few variables and we can add conditions with *if/in* options. - eg. making calculations in given groups. We could display only a few or more parameters, even in the format of the variable too.

```
. summarize AGE SYSBP, detail format

                             AGE
-------------------------------------------------------------
      Percentiles      Smallest
 1%           36             35
 5%           37             36
10%           38             36       Obs                 199
25%           42             36       Sum of Wgt.         199
```

```
      50%           49                          Mean           49.72864
                              Largest       Std. Dev.      9.054087
      75%           57              67
      90%           63              67       Variance        81.9765
      95%           65              67       Skewness       .2462743
      99%           67              67       Kurtosis       1.832025

                                  SYSBP
      -------------------------------------------------------------
            Percentiles      Smallest
       1%        100.0           96.0
       5%        102.0          100.0
      10%        109.5          100.0       Obs                 199
      25%        116.5          100.0       Sum of Wgt.         199

      50%        130.0                      Mean              133.6
                              Largest       Std. Dev.          22.4
      75%        145.0          191.0
      90%        160.0          200.0       Variance          502.0
      95%        180.5          206.0       Skewness            1.0
      99%        206.0          212.0       Kurtosis            3.9

      . by SEX, sort : summarize SYSBP, format

      ----------------------------------------------------------------------------------------------------
----------------------------
      -> SEX = female

          Variable |        Obs        Mean    Std. Dev.       Min        Max
      -------------+---------------------------------------------------------
             SYSBP |         81       129.5        17.1      100.0      180.0

      ----------------------------------------------------------------------------------------------------
----------------------------
      -> SEX = male

          Variable |        Obs        Mean    Std. Dev.       Min        Max
      -------------+---------------------------------------------------------
             SYSBP |        118       136.4        25.1       96.0      212.0


      . cls

      . by CURSMOKE, sort : summarize BMI SYSBP if AGE < 50, format

      ----------------------------------------------------------------------------------------------------
----------------------------
      -> CURSMOKE = 0

          Variable |        Obs        Mean    Std. Dev.       Min        Max
      -------------+---------------------------------------------------------
               BMI |         39    27.56564    5.667047      20.68       45.8
             SYSBP |         40       127.8        20.7       96.0      191.0

      ----------------------------------------------------------------------------------------------------
----------------------------
      -> CURSMOKE = 1

          Variable |        Obs        Mean    Std. Dev.       Min        Max
      -------------+---------------------------------------------------------
               BMI |         65    24.35108    3.443228      16.75       33.8
             SYSBP |         65       123.9        15.5      100.0      162.0
```

To calculate specific statistical parameters, use the *Statistics->Summaries tables and tests ->Other tables ->Compact table of summary statistics* tab. With this we have a lot of options.

```
      . tabstat SYSBP GLUCOSE, statistics( mean kurtosis count ) by(SEX)

      Summary statistics: mean, kurtosis, N
```

```
        by categories of: SEX (SEX)

         SEX |     SYSBP   GLUCOSE
      -------+--------------------
      female |   129.463  81.37975
             |  3.355297   21.1724
             |        81        79
      -------+--------------------
        male |   136.411  83.34545
             |  3.268421  20.04047
             |       118       110
      -------+--------------------
       Total |  133.5829  82.52381
             |    3.8645  20.67695
             |       199       189
      ----------------------------

      . tabstat SYSBP GLUCOSE, statistics( mean kurtosis count ) by(SEX) columns(statistics)

      Summary for variables: SYSBP GLUCOSE
           by categories of: SEX (SEX)

         SEX |      mean  kurtosis         N
      -------+------------------------------
      female |   129.463  3.355297        81
             |  81.37975   21.1724        79
      -------+------------------------------
        male |   136.411  3.268421       118
             |  83.34545  20.04047       110
      -------+------------------------------
       Total |  133.5829    3.8645       199
             |  82.52381  20.67695       189
      --------------------------------------

      . tabstat SYSBP GLUCOSE, statistics( mean kurtosis count ) by(SEX) columns(statistics) longstub

      SEX        variable |      mean  kurtosis         N
      --------------------+------------------------------
      female        SYSBP |   129.463  3.355297        81
                  GLUCOSE |  81.37975   21.1724        79
      --------------------+------------------------------
      male          SYSBP |   136.411  3.268421       118
                  GLUCOSE |  83.34545  20.04047       110
      --------------------+------------------------------
      Total         SYSBP |  133.5829    3.8645       199
                  GLUCOSE |  82.52381  20.67695       189
      ---------------------------------------------------

      . by CURSMOKE, sort : tabstat SYSBP GLUCOSE, statistics( mean kurtosis count ) by(SEX)
columns(statistics) longstub

      -----------------------------------------------------------------------------------------------------
----------------------------
      -> CURSMOKE = 0

      SEX        variable |      mean  kurtosis         N
      --------------------+------------------------------
      female        SYSBP |  128.6029  2.305632        34
                  GLUCOSE |  82.84848   22.0279        33
      --------------------+------------------------------
      male          SYSBP |  144.9403  2.731798        67
                  GLUCOSE |  86.33871  14.74394        62
      --------------------+------------------------------
      Total         SYSBP |  139.4406  3.167852       101
                  GLUCOSE |  85.12632  17.18262        95
      ---------------------------------------------------

      -----------------------------------------------------------------------------------------------------
----------------------------
      -> CURSMOKE = 1

      SEX        variable |      mean  kurtosis         N
      --------------------+-----------------------------
      female        SYSBP |  130.0851  3.544742        47
                  GLUCOSE |  80.32609  16.56266        46
      --------------------+-----------------------------
      male          SYSBP |  125.2059  6.784597        51
                  GLUCOSE |  79.47917  5.259766        48
      --------------------+-----------------------------
      Total         SYSBP |  127.5459  4.895964        98
                  GLUCOSE |  79.89362  19.74424        94
      --------------------------------------------------
```

You might also find it useful when testing missing values.

```
      . cls

      . tabstat BMI, statistics( mean count ) by(BPMEDS)

      Summary for variables: BMI
           by categories of: BPMEDS (BPMEDS)

        BPMEDS |      mean         N
      ---------+--------------------
             0 |  26.26495       190
             1 |  26.63667         6
      ---------+--------------------
         Total |  26.27633       196
      ------------------------------

      . tabstat BMI, statistics( mean count ) by(BPMEDS) missing

      Summary for variables: BMI
           by categories of: BPMEDS (BPMEDS)

        BPMEDS |      mean         N
      ---------+--------------------
             0 |  26.26495       190
             1 |  26.63667         6
             . |    22.745         2
      ---------+--------------------
         Total |  26.24066       198
      ------------------------------

      . by HYPERTEN, sort : tabstat AGE, statistics( mean count ) by(BPMEDS) missing longstub
format(%9.2f)

      ---------------------------------------------------------------------------------------------------
----------------------------
      -> HYPERTEN = 1

      BPMEDS       variable |      mean         N
      ----------------------+--------------------
      0                 AGE |     45.98     46.00
      .                 AGE |     44.00      1.00
      ----------------------+--------------------
      Total             AGE |     45.94     47.00
      -------------------------------------------

      ---------------------------------------------------------------------------------------------------
----------------------------
      -> HYPERTEN = 2

      BPMEDS       variable |      mean         N
      ----------------------+--------------------
      0                 AGE |     51.22    145.00
      1                 AGE |     45.67      6.00
      .                 AGE |     36.00      1.00
      ----------------------+--------------------
      Total             AGE |     50.90    152.00
      -------------------------------------------
```

And even more complex tables can be made …

Even with these simple descriptive statistical calculations, we can get a lot of interesting information and suspicion.

Let’s create the dynamic document to see what we have.

We can put more nice dynamic tables into the dynamic document but it is a little bit complicated - I recommend this only for the enthusiastic, here I present a simpler “intermediate” example. I hide the commands in the document so that the “soulless” will not see:) - it will only appear in the do file.

Variable	Count	Mean	SD
AGE	199	49.7	9.05

To describe the categorical variables in tables we can use the *Statistics-> Summaries tables and tests->Frequency tables* tab. Eg.: *One-way tables*:

```
. tabulate SEX if AGE < 50

        SEX |      Freq.     Percent        Cum.
------------+-----------------------------------
     female |         49       46.67       46.67
       male |         56       53.33      100.00
------------+-----------------------------------
      Total |        105      100.00

. tabulate BPMEDS, missing

     BPMEDS |      Freq.     Percent        Cum.
------------+-----------------------------------
          0 |        191       95.98       95.98
          1 |          6        3.02       98.99
          . |          2        1.01      100.00
------------+-----------------------------------
      Total |        199      100.00
```

We can characterize the combined distribution of more categorical variables using *Two-way tables*:

```
. tabulate SEX CURSMOKE

           |       CURSMOKE
       SEX |         0          1 |     Total
-----------+----------------------+----------
    female |        34         47 |        81
      male |        67         51 |       118
-----------+----------------------+----------
     Total |       101         98 |       199
```

Making figures

A very important part of statistical analysis is the drawing of figures. In *Stata* we can clicking in *Graphics* tab.

One of the most important figures for both continuous and categorical variables is the *Histogram*.

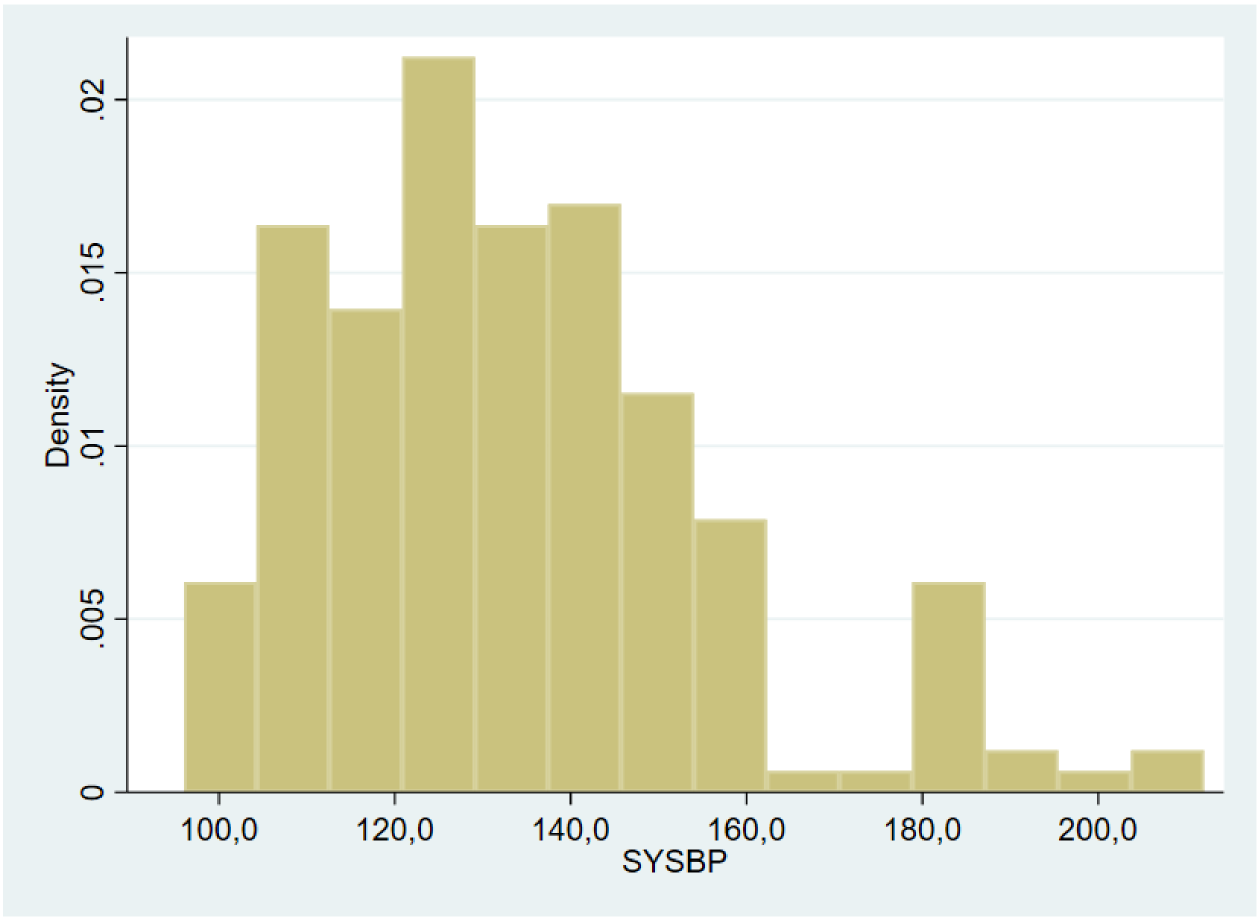
In order to display the image in our dynamic document, you must save the graph and paste it into the document with a given command. If we would like to use the graph in a presentation or in a paper it is worth to save the file into *.wmf* or *.tif*. The previous could be easily edited in powerpoint (changing labels, colours…). But if we would like to use it only for saving or reporting in dynamic documents the best is to save as *.png*. Let's do that.

```
. histogram SYSBP
(bin=14, start=96, width=8.2857143)


. graph export "./histo1.png", as(png) replace
(file ./histo1.png written in PNG format)
```

We should use "$![képcím](elérésiútvonal/filenév)$" in *.do* file to show the graph in the dynamic document. (I'm using $$ here because I would like to show the command without execution) If we use the previously setted working directory we could use "$./filename$" instead of the full path.

Therefore, it is advisable to always define the path at the beginning, because if you copy the files and use on the pendrive and run it elsewhere, then you only need to set the working directory path. Insert the image as described above.



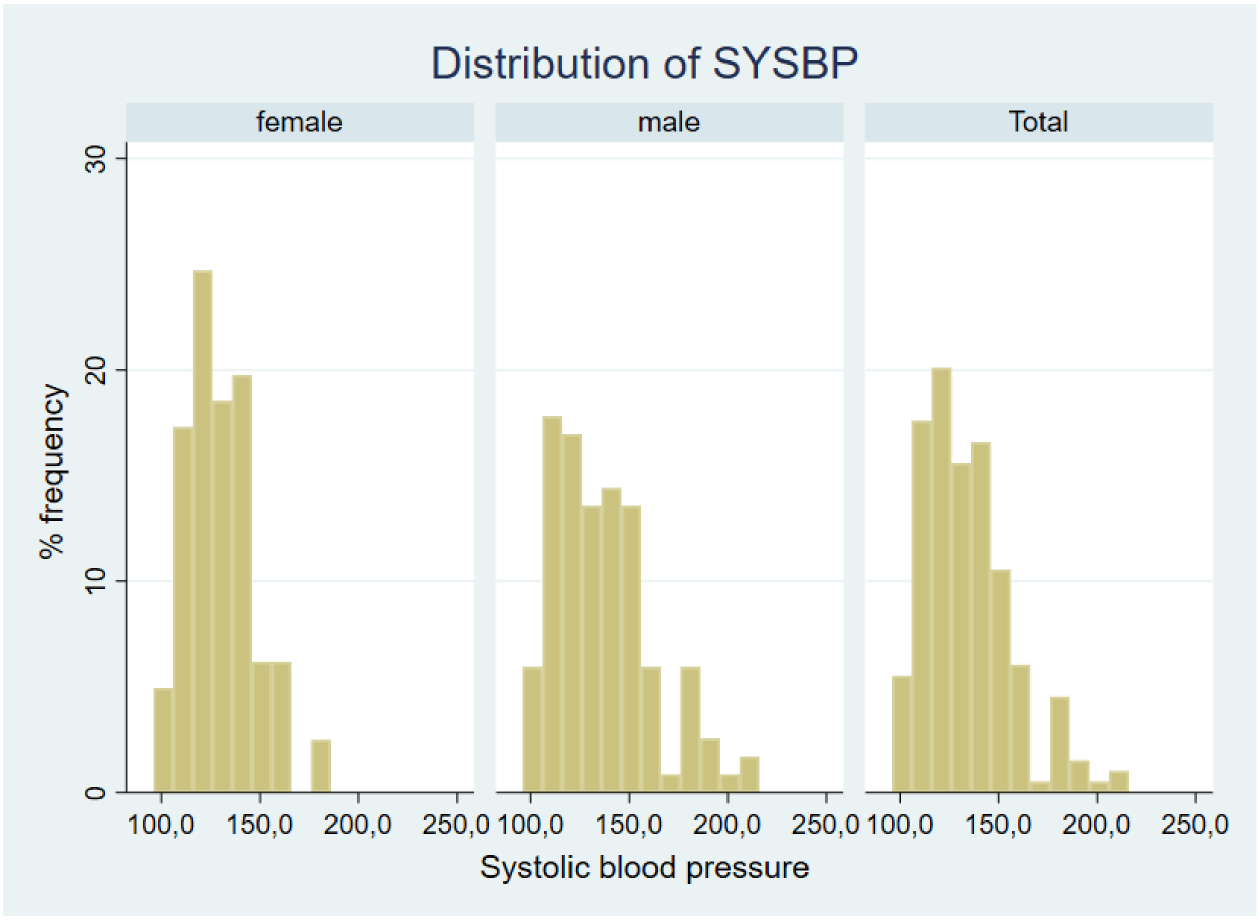
Distribution of Systolic blood pressure

There are a several options available in the graphs - let's see a few of them. Now I paste only a few of them into the document.

Eg.: at the first look at the histogram it is worth setting the column width manually, because the automatic solution is usually not very lifelike, “aesthetic”.

```
      . histogram SYSBP, width(10) percent ytitle(% frequency) xtitle(Systolic blood pressure) by(,
title(Distribution of SYSBP) note("")) by(SEX, total rows(1))


      . graph export "./histo2.png", as(png) replace
      (file ./histo2.png written in PNG format)
```

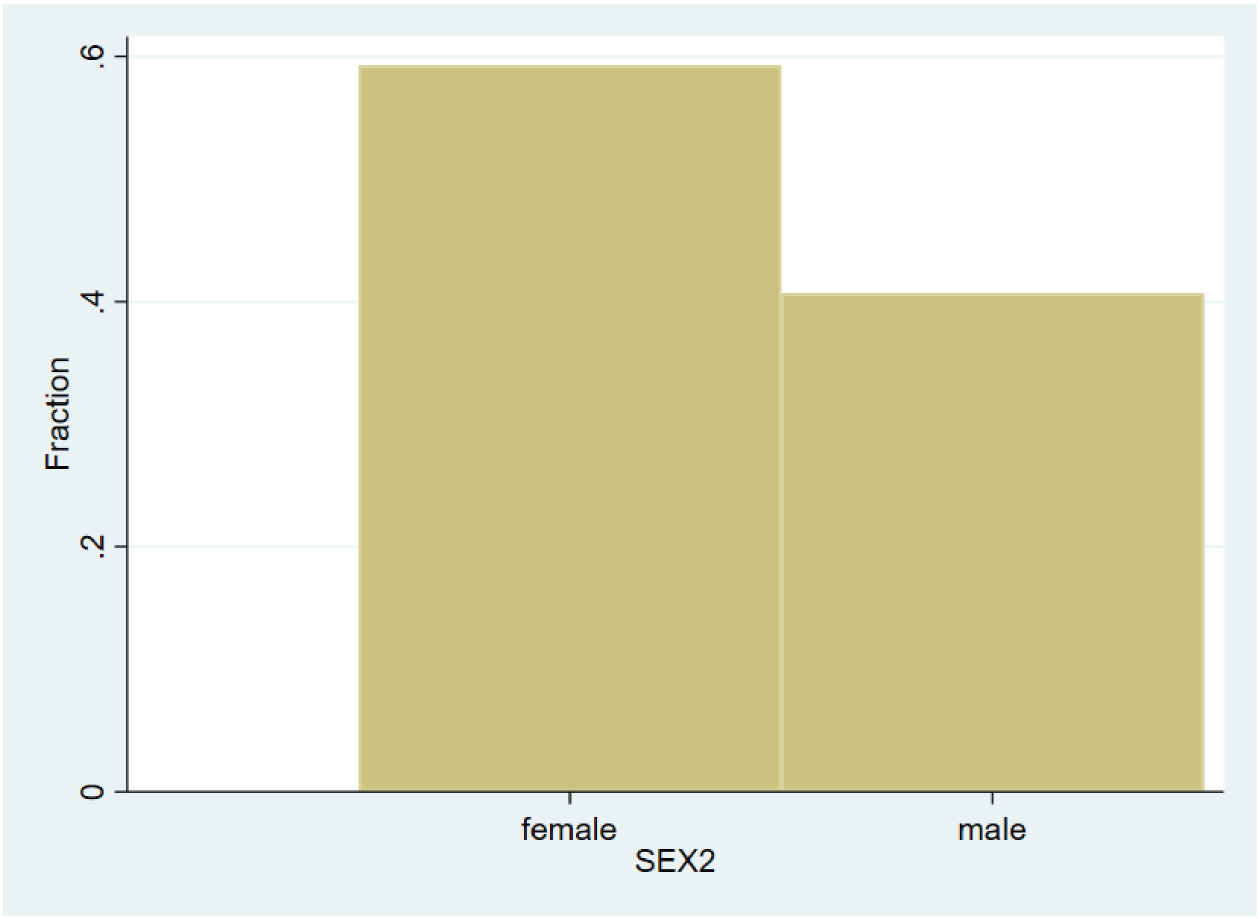


Distribution of Systolic blood pressure2

Let’s create a histogram for a categorical variable (though it is usually less fortunate):

```
      . histogram GENDER, discrete fraction xlabel(1(1)2, valuelabel)
      (start=1, width=1)


      . graph export "./histo3.png", as(png) replace
      (file ./histo3.png written in PNG format)
```

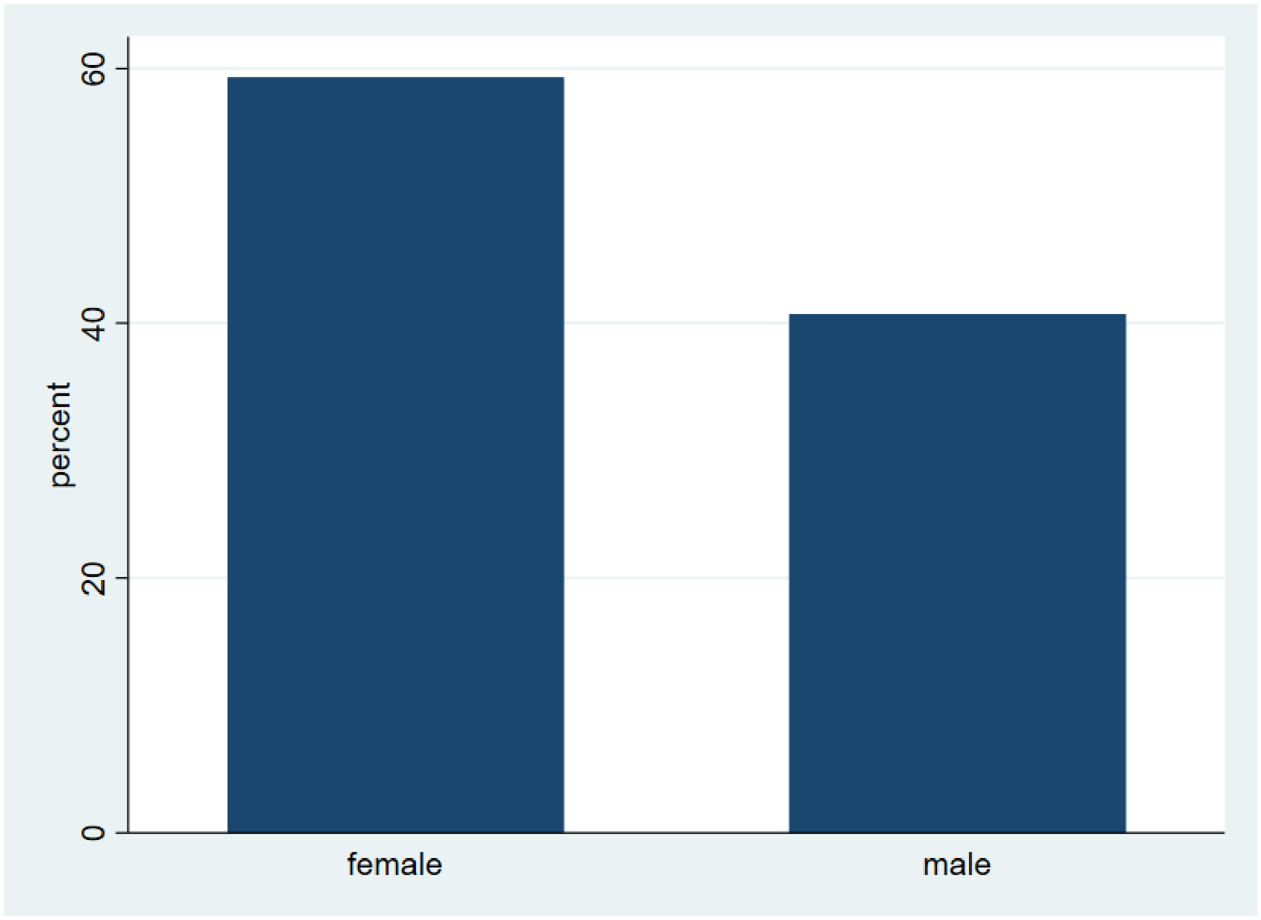


Distribution of gender

In the previous figure I really tried, but it wasn't really "nice", informative. In the case of categorical variables, the use of the *bar chart* is often more appropriate. Let's look at this as an example.

```
. graph bar, over(GENDER)


. graph export "./bar1.png", as(png) replace
(file ./bar1.png written in PNG format)
```

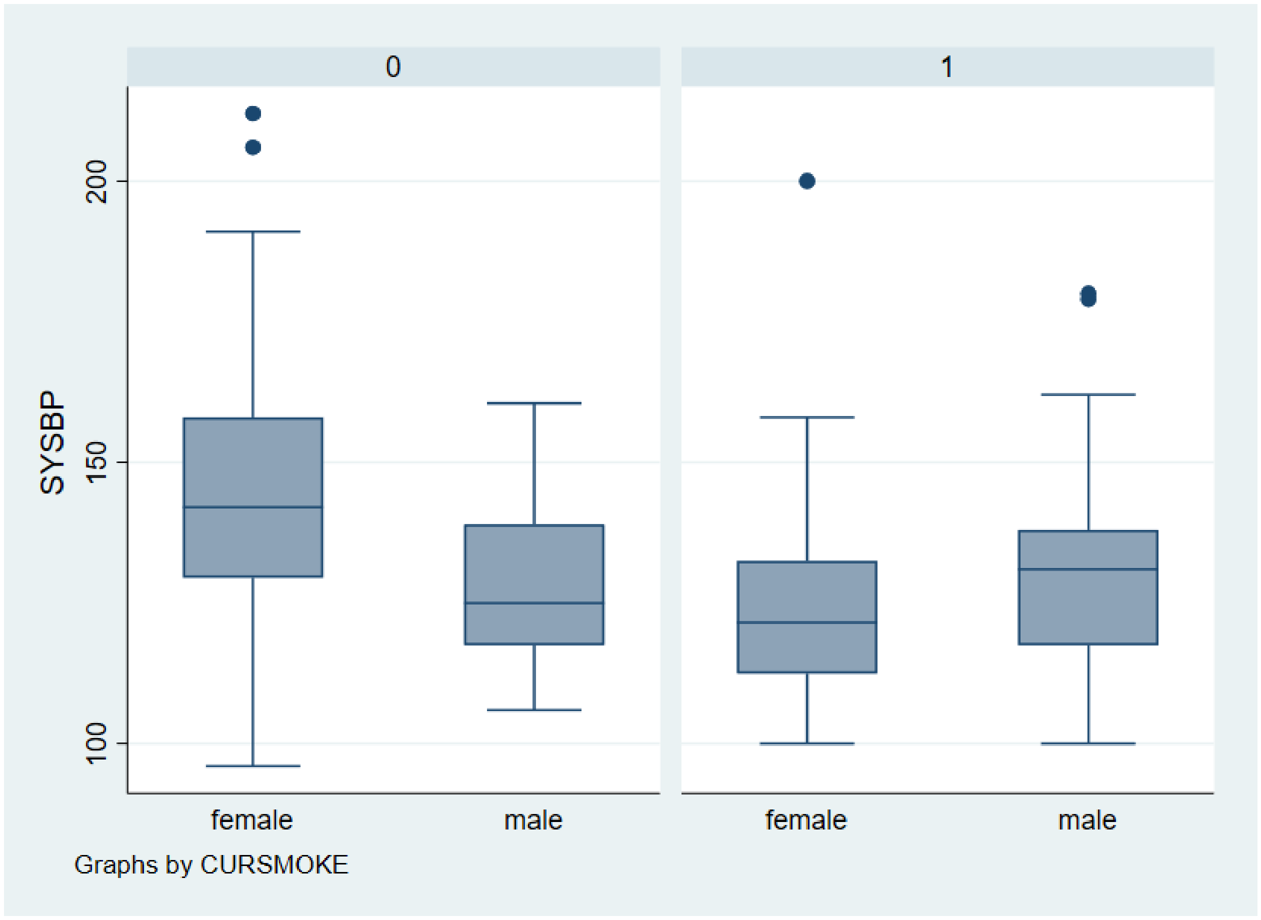


Distribution of gender2

The next very usefull graph is the *box plot*.

```
. graph box SYSBP, over(GENDER) by(CURSMOKE)


. graph export "./box1.png", as(png) replace
(file ./box1.png written in PNG format)
```

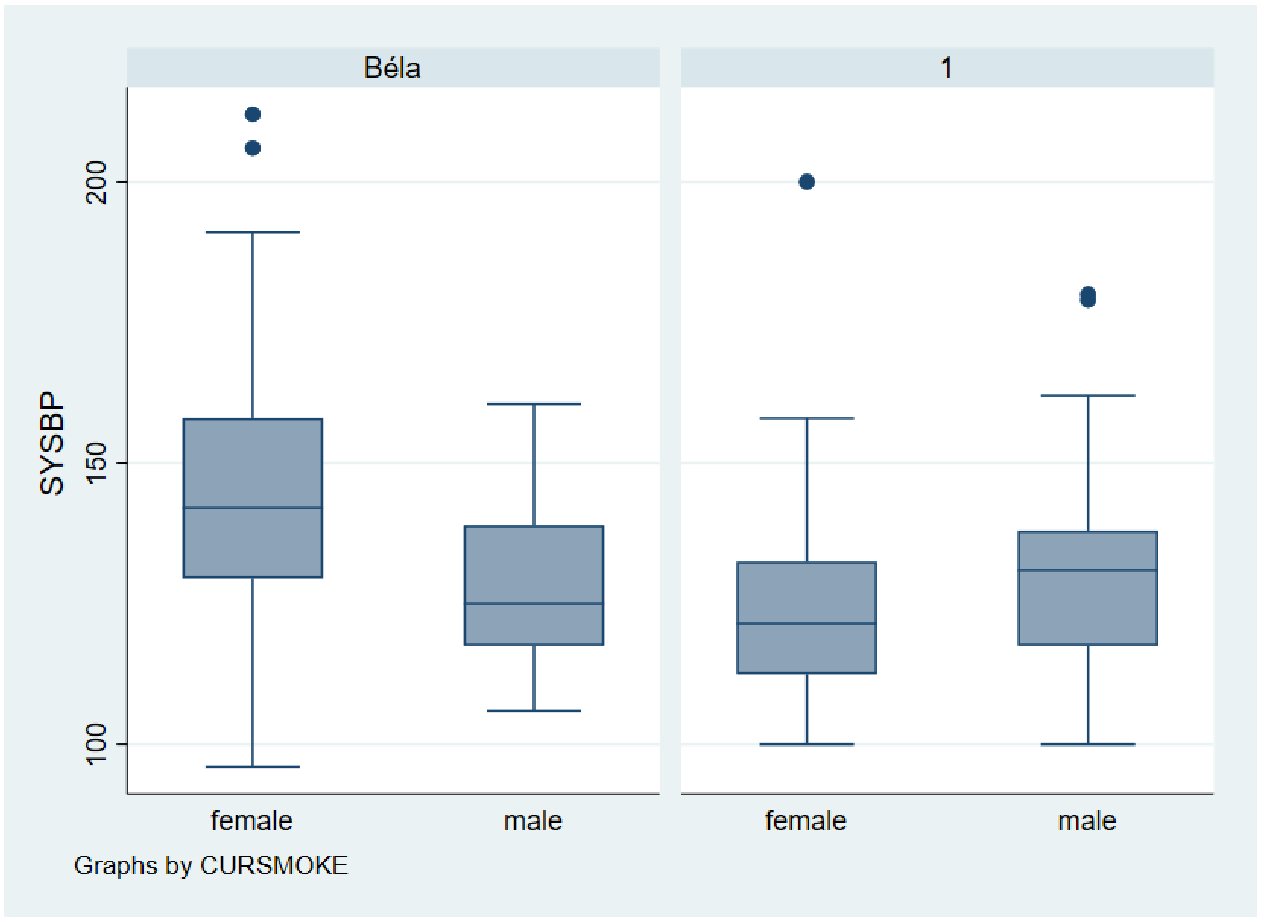


Box plot

After drawing the figure, we can make additional changes to the image with the help of the *start graph editor*. If you want to make these changes in an other image or insert it into our *.do file* code, you need to turn on *start recording* and make the changes and at the end *stop recording* and save.

```
. graph box SYSBP, over(GENDER) by(CURSMOKE) play(a)


. graph export "./box1m.png", as(png) replace
(file ./box1m.png written in PNG format)
```



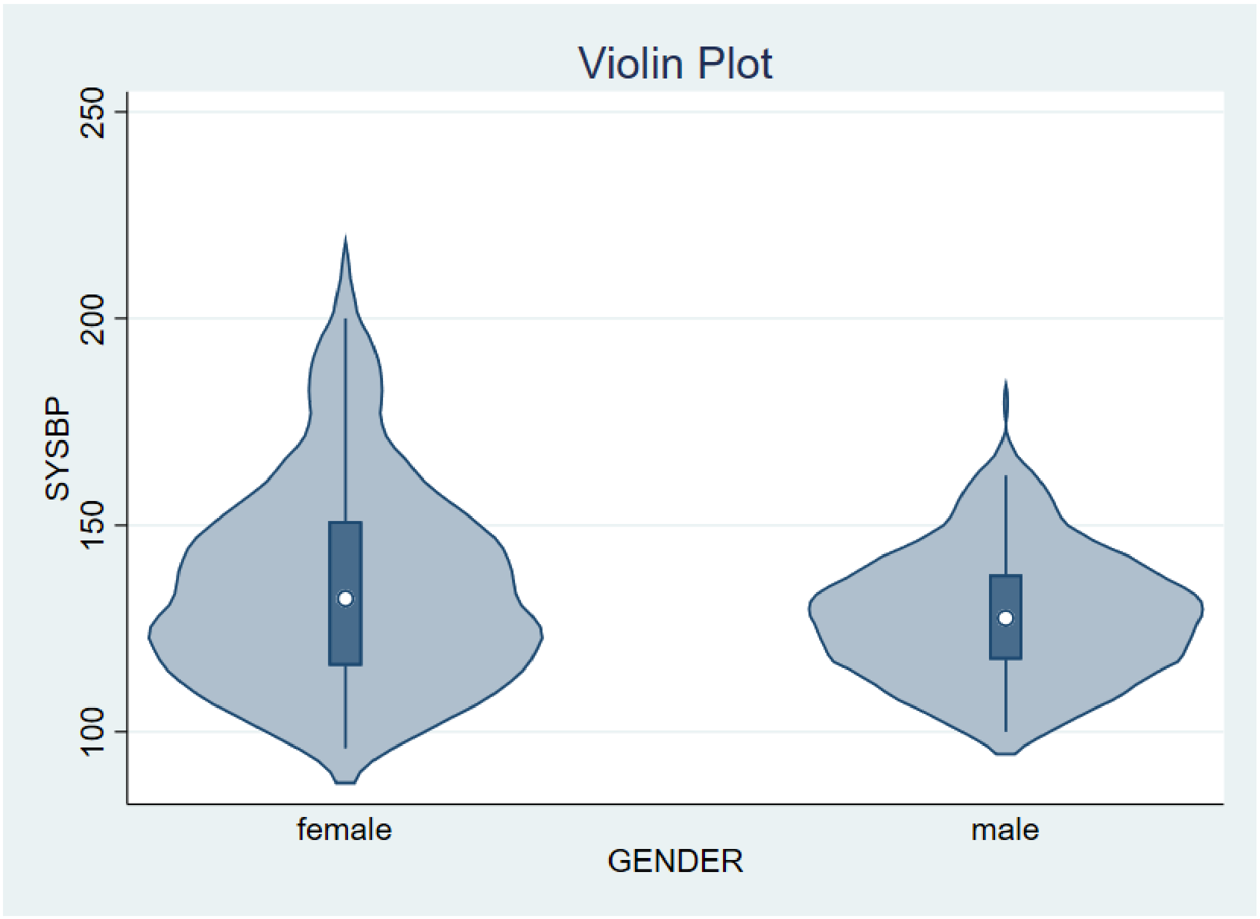
Modified Box plot

There are some drawings that we like to use, but not in the basic stata. In the Stata we could use additional user written *packages*. For example we could create violin plots that is included in the vioplot package. We could install packages using *ssc install packagename*. The user written packages usually don't have clicking screen, but they have help. (type help vioplot)

```
. ssc install vioplot
checking vioplot consistency and verifying not already installed...
all files already exist and are up to date.

. vioplot SYSBP, over(GENDER) title("Violin Plot") ytitle("SYSBP") xtitle("GENDER")

. graph export "./vioplot1.png", as(png) replace
(file ./vioplot1.png written in PNG format)
```

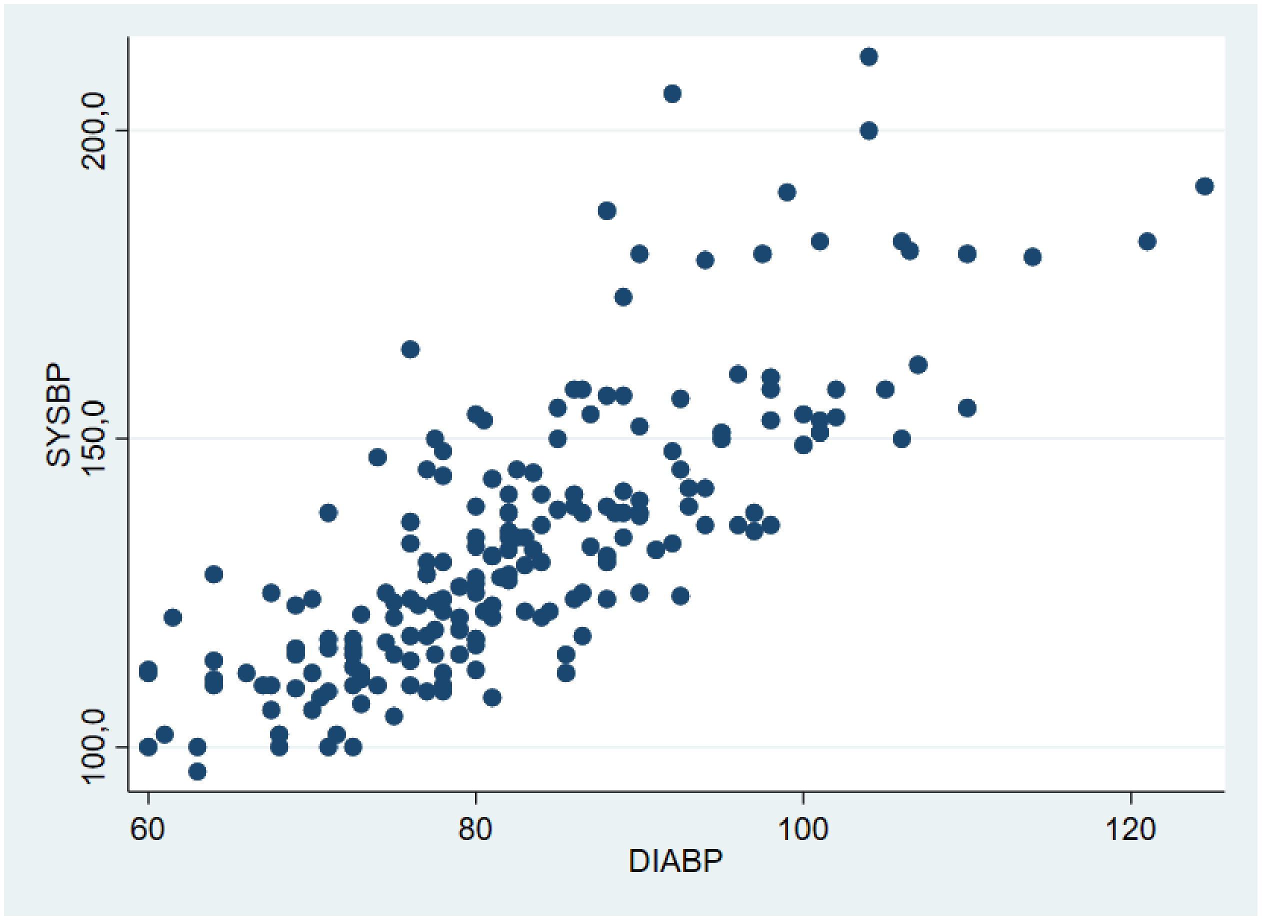


Violin plot

We can make a number of different graphs, but what I want to show now is a scatterplot. We usually use this to plot the relation of two numerical variable. We could make it via *Twoway graphs*.

```
. twoway (scatter SYSBP DIABP)


. graph export "./scatter1.png", as(png) replace
(file ./scatter1.png written in PNG format)
```

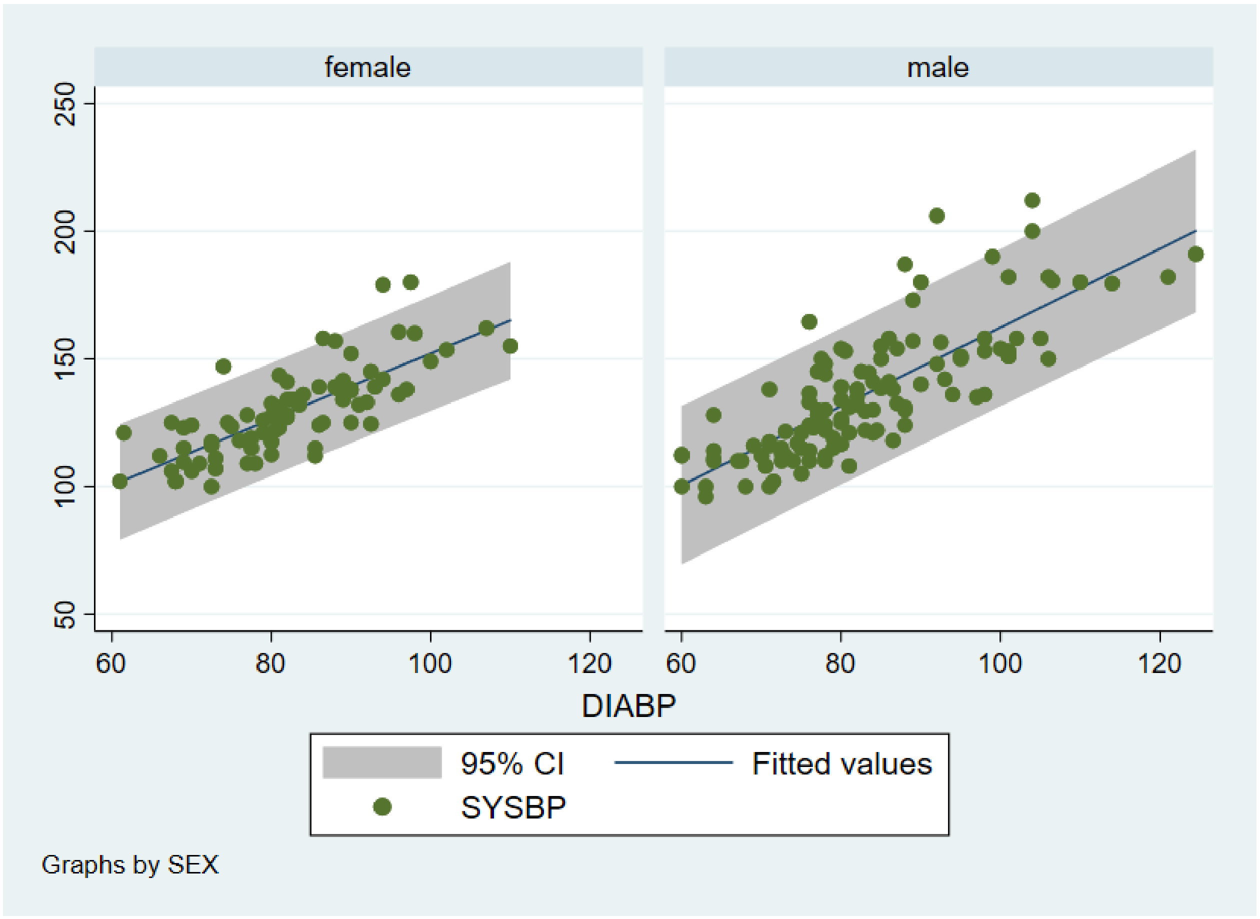


Scatterplot

We also have the ability to draw different figures, fittings on top of each other. Here, without further explanation, I show an example.

```
. twoway (lfitci SYSBP DIABP, stdf) (scatter SYSBP DIABP), by(SEX)


. graph export "./scatter2.png", as(png) replace
(file ./scatter2.png written in PNG format)
```



Scatter2

Hypothesis tests

Let's look at some examples of hypotheses tests that we discussed in the previous lecture. For example, we may be wondering whether systolic blood pressure (SYSBP) is different in the sexes. Looking at the previous figures, and knowing that we are working with large sample size let's make Welch test. *Statistics->Summaries, tables and tests->Classical tests of hypotheses->t-tests (mean comparison tests)*.

```
. ttest SYSBP, by(SEX) unequal welch

Two-sample t test with unequal variances
------------------------------------------------------------------------------
   Group |     Obs        Mean    Std. Err.   Std. Dev.   [95% Conf. Interval]
---------+--------------------------------------------------------------------
  female |      81     129.463    1.904121    17.13709    125.6736    133.2523
    male |     118     136.411    2.308771    25.07967    131.8386    140.9834
---------+--------------------------------------------------------------------
combined |     199    133.5829    1.588345    22.40636    130.4507    136.7152
---------+--------------------------------------------------------------------
    diff |           -6.948054    2.992674               -12.84948   -1.046629
------------------------------------------------------------------------------
    diff = mean(female) - mean(male)                              t =  -2.3217
Ho: diff = 0                             Welch's degrees of freedom =  198.992

    Ha: diff < 0                 Ha: diff != 0                 Ha: diff > 0
 Pr(T < t) = 0.0106         Pr(|T| > |t|) = 0.0213          Pr(T > t) = 0.9894
```

Interpret the results based on that we learned in the last class.

Examine the correlation between sex and smoking habit (CURSMOKE) - do a Fiher exact test. *Statistics->Summaries, tables and tests->Frequency tables->Two-way table with measures of association*.

```
. tabulate SEX CURSMOKE, cell column exact row

+-------------------+
| Key               |
|-------------------|
|     frequency     |
|  row percentage   |
| column percentage |
|  cell percentage  |
+-------------------+

           |       CURSMOKE
       SEX |         0          1 |     Total
-----------+----------------------+----------
    female |        34         47 |        81 
           |     41.98      58.02 |    100.00 
           |     33.66      47.96 |     40.70 
           |     17.09      23.62 |     40.70 
-----------+----------------------+----------
      male |        67         51 |       118 
           |     56.78      43.22 |    100.00 
           |     66.34      52.04 |     59.30 
           |     33.67      25.63 |     59.30 
-----------+----------------------+----------
     Total |       101         98 |       199 
           |     50.75      49.25 |    100.00 
           |    100.00     100.00 |    100.00 
           |     50.75      49.25 |    100.00 

           Fisher's exact =                 0.044
   1-sided Fisher's exact =                 0.028
```

Interpret the results based on that we learned in the last class.

Calculating confidence intervals

In the *Introduction to statistics I.* lecture we heard the next example:

Out of 52 patient we have 6 that lived more than 5 years. What is the probability of survival and what is the confidence of this estimation?

The probability of survival:

```
. display 6/52
.11538462
```

The confidence interval: *Statistics->Summaries, tables, and tests->Summary and descriptive statistics -> Proportion CI calculator*

(I have to use a different code in the documentation because the dynamic document creating package has not been updated yet. (it is a very rare situation when we have to use different command).

```
. version 14: cii 52 6, binomial

                                                         -- Binomial Exact --
    Variable |        Obs        Mean    Std. Err.       [95% Conf. Interval]
-------------+---------------------------------------------------------------
             |         52    .1153846    .0443047        .0435412    .2344083
```

Let's examine the proportion of smokers (CURSMOKE variable) in the sample, and what is the error of estimation made by the sample on the population based on the relative frequencies of smokers. Let's use *Statistics->Summaries, tables, and tests->Summary and descriptive statistics-> Confidence Intervals*.

```
. ci CURSMOKE, binomial

                                                         -- Binomial Exact --
    Variable |        Obs        Mean    Std. Err.       [95% Conf. Interval]
-------------+---------------------------------------------------------------
    CURSMOKE |        199    .4924623      .03544        .4210595     .564093
```

We could estimate the probability of been female in the population based on the relative frequency of female in the sample. We could estimate the confidence of this estimation too. We could to this using *Statistics->Summaries, tables, and tests->Summary and descriptive statistics-> Confidence Intervals*. Unfortunately this is not working only if we coding our variable to 0 and 1. But now we have 1 and 2 for female and male. Additional problem that in this way we could get the estimation only for the category coded by 1 (see later).

This problem can be solved by slightly changing the command, which is useful for later commands too. Let's begin the command with *xi:* for factor variables and put *i.* before the name of the factor. The disadvantage of this solution is that we will see the numerical code and not the value label in the results. Therefore it worths to list the factor codes and labels. To set the value we would like see we have to gove a reference value ("base level") using *char variablename [omi] value*.

```
. label list SEX
SEX:
           1 female
           2 male

. char SEX [omit] 1

. xi: ci i.SEX, binomial
i.SEX             _ISEX_1-2           (naturally coded; _ISEX_1 omitted)

                                                         -- Binomial Exact --
    Variable |        Obs        Mean    Std. Err.       [95% Conf. Interval]
-------------+---------------------------------------------------------------
     _ISEX_2 |        199    .5929648     .034826        .5212301    .6618869

. char SEX [omit] 2

. xi: ci i.SEX, binomial
i.SEX             _ISEX_1-2           (naturally coded; _ISEX_2 omitted)

                                                         -- Binomial Exact --
    Variable |        Obs        Mean    Std. Err.       [95% Conf. Interval]
-------------+---------------------------------------------------------------
     _ISEX_1 |        199    .4070352     .034826        .3381131    .4787699
```

In this case, we can simplify the problem by asking the program not to omit a value:

```
. xi, noomit: ci i.SEX, binomial

                                                        -- Binomial Exact --
    Variable |        Obs        Mean    Std. Err.       [95% Conf. Interval]
-------------+---------------------------------------------------------------
     _ISEX_1 |        199    .4070352     .034826        .3381131    .4787699
     _ISEX_2 |        199    .5929648     .034826        .5212301    .6618869
```

Let’s create an estimation based on the SYSBP variable - that is a numerical one - for the mean systolic blood pressure of the population:

```
. ci SYSBP

    Variable |        Obs        Mean    Std. Err.       [95% Conf. Interval]
-------------+---------------------------------------------------------------
       SYSBP |        199       133.6         1.6           130.5       136.7
```

Commands that occurs in the next lectures, practices

In the next lectures and practices it will be common to calculate risks and odds. For example if we are interested in the risk, risk ratio and odds ratio of angina (ANGINA) regarding to smoking(CURSMOKE). We can use *Statistics -> Epidemiology and related -> Tables for epidemiologists -> Cohort study risk-ratio etc.* for calculations:

```
. cs ANGINA CURSMOKE

                 | CURSMOKE               |
                 |   Exposed   Unexposed  |      Total
-----------------+------------------------+------------
           Cases |        13          21  |         34
        Noncases |        85          80  |        165
-----------------+------------------------+------------
           Total |        98         101  |        199
                 |                        |
            Risk |  .1326531    .2079208  |   .1708543
                 |                        |
                 |      Point estimate    |    [95% Conf. Interval]
                 |------------------------+------------------------
 Risk difference |        -.0752677       |   -.1790651    .0285296
      Risk ratio |         .6379981       |    .3386405    1.201987
 Prev. frac. ex. |         .3620019       |   -.2019871    .6613595
 Prev. frac. pop |         .1782723       |
                 +-------------------------------------------------
                               chi2(1) =     1.99  Pr>chi2 = 0.1584

. cs ANGINA CURSMOKE, or

                 | CURSMOKE               |
                 |   Exposed   Unexposed  |      Total
-----------------+------------------------+------------
           Cases |        13          21  |         34
        Noncases |        85          80  |        165
-----------------+------------------------+------------
           Total |        98         101  |        199
                 |                        |
            Risk |  .1326531    .2079208  |   .1708543
                 |                        |
                 |      Point estimate    |    [95% Conf. Interval]
                 |------------------------+------------------------
 Risk difference |        -.0752677       |   -.1790651    .0285296
      Risk ratio |         .6379981       |    .3386405    1.201987
 Prev. frac. ex. |         .3620019       |   -.2019871    .6613595
```

```
       Prev. frac. pop |         .1782723       |
            Odds ratio |         .5826331       |     .276669    1.228469 (Cornfield)
                       +-------------------------------------------------
                                     chi2(1) =     1.99  Pr>chi2 = 0.1584
```

In addition, there will be plenty of different regressions, both on exercises and lectures, where ou need the next commands:

```
. regress MAP AGE SEX

      Source |       SS           df       MS      Number of obs   =       199
-------------+----------------------------------   F(2, 196)       =     12.18
       Model |  4612.45958         2  2306.22979   Prob > F        =    0.0000
    Residual |  37102.4649       196   189.29829   R-squared       =    0.1106
-------------+----------------------------------   Adj R-squared   =    0.1015
       Total |  41714.9245       198  210.681437   Root MSE        =    13.759

------------------------------------------------------------------------------
         MAP |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
         AGE |   .5186401   .1098116     4.72   0.000     .3020762    .7352041
         SEX |   1.121007   2.018685     0.56   0.579    -2.860125    5.102139
       _cons |   72.23372   5.894663    12.25   0.000     60.60862    83.85883
------------------------------------------------------------------------------


. xi: logistic i.DIABETES GLUCOSE AGE i.CURSMOKE i.SEX
i.DIABETES        _IDIABETES_0-1      (naturally coded; _IDIABETES_0 omitted)
i.CURSMOKE        _ICURSMOKE_0-1      (naturally coded; _ICURSMOKE_0 omitted)
i.SEX             _ISEX_1-2           (naturally coded; _ISEX_2 omitted)

Logistic regression                             Number of obs     =        189
                                                LR chi2(4)        =      37.65
                                                Prob > chi2       =     0.0000
Log likelihood = -7.7779797                     Pseudo R2         =     0.7076

------------------------------------------------------------------------------
_IDIABETES_1 | Odds Ratio   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
     GLUCOSE |   1.080306   .0274155     3.04   0.002     1.027887    1.135398
         AGE |   1.127684   .1290481     1.05   0.294     .9011121    1.411223
_ICURSMOKE_1 |   24.79489   53.45607     1.49   0.136      .362435    1696.267
     _ISEX_1 |   .0896862   .2225047    -0.97   0.331     .0006934    11.60094
       _cons |   3.23e-09   2.39e-08    -2.64   0.008     1.61e-15    .0064897
------------------------------------------------------------------------------
Note: _cons estimates baseline odds.
```

Furthermore is a common design when we make repeated measures on the same subjects. In this case we could arrange the dataset in two way:

- *long* (longitudinal) format, when we repeat the cases in the ROWS regarding to the repeated measurements.
- *wide* format, when we put the repeated measurements in different COLUMNS.

Both forms have their advantages and disadvantages, so we often have to transform them. Let's first look at some of our previously analyzed data in *long* format.

```
. clear

. import excel "frmgham_ea_long.xls", sheet("frmgham2") firstrow
```

We recognize that age (AGE), systolic blood pressure (SYSBP), and diastolic blood pressure (DIABP) were recorded at different time points (corresponding to PERIOD) for each individual.

(Obviously, SEX does not change). RANDID identifies individuals (cases). So we can convert our dataset into *wide* format: *Data -> Create or change data -> Other variable-transformation commands -> Convert data between wide and long*

```
. reshape wide AGE SYSBP DIABP, i(RANDID) j(PERIOD)
(note: j = 1 2 3)

Data                               long   ->   wide
-----------------------------------------------------------------------------
Number of obs.                      199   ->      77
Number of variables                   6   ->      11
j variable (3 values)            PERIOD   ->   (dropped)
xij variables:
                                    AGE   ->   AGE1 AGE2 AGE3
                                  SYSBP   ->   SYSBP1 SYSBP2 SYSBP3
                                  DIABP   ->   DIABP1 DIABP2 DIABP3
-----------------------------------------------------------------------------
```

Transforming from wide to long:

```
. reshape long AGE SYSBP DIABP, i(RANDID) j(TIME)
(note: j = 1 2 3)

Data                               wide   ->   long
-----------------------------------------------------------------------------
Number of obs.                       77   ->     231
Number of variables                  11   ->       6
j variable (3 values)                     ->   TIME
xij variables:
                         AGE1 AGE2 AGE3   ->   AGE
                   SYSBP1 SYSBP2 SYSBP3   ->   SYSBP
                   DIABP1 DIABP2 DIABP3   ->   DIABP
-----------------------------------------------------------------------------
```

Here we have to pay attention to the following: the repeated variables should be marked appropriately. (eg. numbered continously for all repeated variables!).

Also note that there were patients who were not measured in a given period - but the corresponding data in our data set - because of its “permanent” properties - is shown as missing values. We should eliminate this “cases” with the *drop* command. Now I do it in a strange way to show some other usefull operations: I delete the cases where AGE and (& symbol) SYSBP or (| symbol) DIABP is missing. There is an additional symbol for not equal that we have to learn: *!=*.

```
. drop if AGE ==. & (SYSBP ==. | DIABP ==.)
(32 observations deleted)
```

Estimating the sample size was also discussed in the previous lecture. If we are going to plan a study, it is good to know the sample size that needed to find a significant result for a relevant difference. In *Stata* this can be calculated by using the *Statistics -> Power and sample size* tab.

Let’s calculate how much sample size I need if we assume (based on previous knowledge):

- the means are 120 and 130
- the standard deviations are 10
- the first type error is 1%, the power is 90%
- the second group size is 3-times higher

- I willing to use 2 sample t-test

```
. power twomeans 120 135, sd(10) alpha(0.01) power(0.9) nratio(3)

Performing iteration ...

Estimated sample sizes for a two-sample means test
t test assuming sd1 = sd2 = sd
Ho: m2 = m1  versus  Ha: m2 != m1

Study parameters:

        alpha =    0.0100
        power =    0.9000
        delta =   15.0000
           m1 =  120.0000
           m2 =  135.0000
           sd =   10.0000
        N2/N1 =    3.0000

Estimated sample sizes:

            N =        40
           N1 =        10
           N2 =        30
```

At last look our final dynamic document! (*db markdoc*…)

——END——

If you would like to make dynamic documentation with markdoc

type the followings into the command line

1. net install github, from("https://haghish.github.io/github/")
2. github install haghish/markdoc
3. markdocpandoc 4.markdocwkhtmltopdf

check: http://www.haghish.com/packages/markdoc/example.do