Stata Lab 2.

Univariate Analysis

Dr. Adam William Chalmers

Table of Contents

[Getting Started 2](#_Toc63696203)

[Part 1. Rename, recode, and generate 2](#_Toc63696204)

[The ‘drop’ and ‘keep’ commands (again) 4](#_Toc63696205)

[Stata Exercises: 1 5](#_Toc63696206)

[Part 2. Generate and Egen 6](#_Toc63696207)

[The ‘generate’ command 6](#_Toc63696208)

[Extensions of the Generate Command: egen 7](#_Toc63696209)

[Stata Exercises: 2 9](#_Toc63696210)

[Part 3. Visualising Data 10](#_Toc63696211)

[Histograms 10](#_Toc63696212)

[Box and whisker plots 11](#_Toc63696213)

[Dot charts 14](#_Toc63696214)

[Editing Graphs 16](#_Toc63696215)

[Stata Exercises. 3 17](#_Toc63696216)

[Advanced Stata Exercises 18](#_Toc63696217)

# **Getting Started**

A picture containing drawing

Description automatically generated

Step 2. Double-click the icon to launch Stata.

Load the following data file: **Data1.dta**

* The Data file can be found on Keats

# **Part 1.** **Rename, recode, and generate**

**The ‘rename’ command**

The ‘rename’ command is used to rename variables. It is useful if you find that the variable labels provided in existing data are cumbersome or ambiguous. The general syntax for this command is as follows:

. rename *var newvar*

Example: I want to rename the variable ‘np11701’ so that it is easier to use and identify. We can see that the variable label is ‘General Life Satisfaction.’ Hence, let’s rename np11701 so that it is easier to identify and use. Let’s make the new name for np11701 ‘LifeSatisfaction’. To do this, execute the following command.

. rename np11701 LifeSatisfaction

**Graphical user interface, text, application

Description automatically generated**

You should now see your renamed variable in the List of Variables window (above).

**\*\*Pro Tip**: Stata will not recognize spaces in variable names. Be sure to use variable names that don’t have spaces. You can also use an underscore (\_) to create spaces. E.g., instead of ‘LifeSatisfaction’, you could use ‘Life\_Satisfaction’. However, you cannot use ‘Life Satisfaction’.

**The ‘recode’ command**

The ‘recode’ command is very powerful and can be used to change labels for different values of a variable but can also be used to create new categories within a variable. Here is the general syntax.

. recode *var* (rule)(rule)(rule) … , gen(*newvar*)

Example 1: I want to recode ‘gender’ so that women = 1 and men = 0. Currently, our dataset has women = 2 and men = 1. There are two ways to do this: first without labels and second with labels.

Without labels:

. recode gender (2=1)(1=0), gen(gender1)

Note: in the recoding above, we are just telling Stata that for the variable ‘gender’ we want current values 2 to now equal 1, and current values 1 to now equal 0, and we want to make a new variable for these changes called ‘gender1’. We can call the new variable anything we like, but this makes sense.

With labels:

. recode gender (2=1 "F")(1=0 "M"), gen(gender2)

Or

. recode gender (2=1 "Female")(1=0 "Male"), gen(gender3)

Check your work by creating a frequency table (the ‘tab’ command). Below we can see the results of all three “recodings”.

Table

Description automatically generated

Example 2: I want to recode ‘state’ in a way that categorizes the German Bundesländer into two groups: 1 = former East Germany and 0 = Former West Germany and call the new variable ‘East\_West’.

Step 1. We need to get the values for each label

. tab state

. tab state, nolabel

|  |  |
| --- | --- |
|  |  |

\*\*Note: I have placed these tables side by side. Stata will **not** do this automatically.

. recode state (11 13 14 15 16 = 1 “EastGermany”)(0 1 2 3 4 5 6 7 8 9 10 = 0 “WestGermany”), gen(East\_West)

Let’s check our work.

.tab East\_West



Above we can see that we have created a new variable called ‘East\_West’ and that it has three categories, ‘WestGermany’, ‘EastGermany’ and ‘12’. The category for 12 is a mistake that I made in my recoding. What can we do?

**\*\*Pro Tip**: You don’t have to type out each number. You can also tell Stata to include all numbers between certain values by using ‘/’. The order by which you recode is also important. Try this.

. recode state (11 13 14 15 16 = 1 “EastGermany”)(0/10 = 0 “WestGermany”), gen(East\_West\_alt)

## The ‘drop’ and ‘keep’ commands (again)

We can use drop and keep for many different functions.

Example: I am unhappy with my coding of ‘East\_West’ because I forgot to include ‘12’ (Meck-Vor). I want to delete this new variable and try again.

. drop East\_West

. recode state (11 12 13 14 15 16 = 1 “EastGermany”)(0/10 = 0 “WestGermany”), gen(East\_West)

Check your work

. tab East\_West

A screenshot of a cell phone

Description automatically generated

. drop if *exp*

Example: I want to exclude East Berlin from my dataset.

. drop if state == 11

**\*\*Pro Tip**: I have just dropped individuals who come from East Berlin from my dataset. The only way to get them back is to reload Data1.dta. You could save your data at this point, but it is not necessary as we want to use the full dataset.

Example 2: I want to exclude East Berlin and West Berlin from my dataset. We can either drop each one individually, or we can use the “|” symbol to list things we want to drop.

. drop if state == 11 | state == 0

🡪 again, reload your original data after this exercise: Data1.dta

. keep if *exp*

Example: I want to only include Berlin and East Berlin in my dataset

. keep if state == 11 | state == 0

🡪 again, reload your original data after this exercise: Data1.dta

# **Stata Exercises: 1**

Please complete the following tasks. Execute commands from a Do-file.

**Task 1:** Rename the following using ‘appropriate’ descriptive names. Appropriate descriptive names express what the variable is about and will make them easier for us to recognize.

* np9501
* np9502
* np9503
* np9504
* np9506
* np9507

**Task 2.** Recode house size using ‘sqfeet’ into four approximately equal categories with the following values and labels. It’s up to you do decide which values fall into which categories. Call your new variable ‘HouseSize’ Check your work by using the ‘tabulate’ command for your new variable.

1. Tiny
2. Small
3. Medium
4. Large

**Task 3.** Recode neighbourhood type using the variable ‘area’ with the following changes: create a category where 1 = ‘new houses’ (which include new houses and mixed age), a category where 2 = ‘old houses’ (which includes ancienty houses), and a third category for all other values (shops and houses, industrial area, and other) that is coded as missing (.).

How many 'new houses' are there in your new variable?

How many missing are in your new variable?

# **Part 2. Generate and Egen**

## 

## The ‘generate’ command

The ‘generate’ command allows you to create new variables, often by transforming existing variables. We have already used the ‘generate’ when we learned to recode variables. Here we are learning how to generate variables from scratch.

**Example 1: Creating a new Variable**

Using **Data1.dta**, I want to create a new variable called ‘age’ for the age of each individual in the dataset. Currently we only know their date of birth (ybirth). To calculate a person’s age, we subtract the current year by their year of birth. Let’s stick with 2020 for this one. After creating this new variable, you can check your work by using the sum or tab commands.

. gen age = 2020-ybirth

Let’s check our work with the ‘summarize’ command.

. sum age

A screenshot of a cell phone

Description automatically generated

**Note**: we can see that the youngest person in the data is 39 years old and the oldest is 118. This is a bit misleading because the data were collected in 1997.

Looks what happens if I calculated age at the time the data were actually collected.

. gen age1 = 1997-ybirth

**Table

Description automatically generated**

**Now** the data are more realistic. The youngest person is 16 and the oldest is 95. The average age is 45.

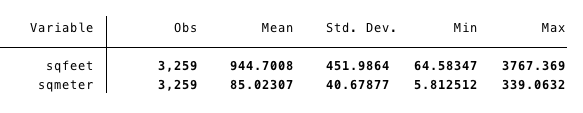
**Example 2. Creating a New Variable**

The Data1.dta dataset records house size in square feet. However, I want to include a variable for house size in square meters. Note: 1 square foot = 0.09 square meters.

. gen sqmeter = sqfeet\*0.09

Take a look at the difference between square feet and square meters.

. sum sqfeet sqmeter



**Example 3. Creating an Index**

In statistics, an index is a compound measure that aggregates multiple indicators. It can be very useful if you have a number of indicators measuring different dimensions of the same concept.

Consider this example. I want to create a ‘Satisfaction Index’ that combines ‘General life satisfaction’ (np11701) and ‘Satisfaction with living conditions’ (np0105). My logic is that these two things are measuring something broader, namely satisfaction with life. To create a basic index, we just add these two variables together.

. gen Satisfaction\_Index = np11701+np0105

Check out the distribution of values in your new index.

. tab Satisfaction\_Index

****

## Extensions of the Generate Command: egen

Stata allows for multiple extensions of the Generate command. To see a full list of extensions, check out the help file for ‘egen’.

. help egen

We will only look at a few of the most important egen commands.

**Egen within groups**

The egen command is useful for generating new variables about groups within a given variable.

**Example 1**. I want to know the average (mean) house size of the six different groups of neighbourhoods in the dataset. In this case, the variable with groups is ‘area’ and the parameter for each group will be ‘sqfeet’. The general syntax for this egen command is:

. by *var1*: egen *newvar* = mean (*var2*)

Note that you can use other measures of central tendency as well (mode and median) as well as things like min (minimum value) and max (maximum value) as well as count and sum. To see a full list of options, pull up Stata’s help file for egen (help egen).

Egen requires that we first ‘sort’ by the group we are interested.

. sort area

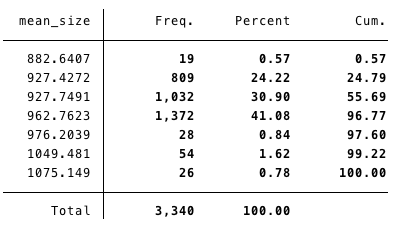
. by area: egen mean\_size = mean (sqfeet)

Or we can it it all in one command.

. by area, sort: egen mean\_size = mean (sqfeet)

We can look at the results of our new variable ‘mean\_size’

. tab mean\_size



And we can look at mean size in each group using the ‘tabdisp’ command, which allows us to display tables with more than one variable. Here’s the generic syntax for tabdisp.

. tabdisp rowvariab, cellvariable(var)

. tabdisp area, cell(mean\_size)

Table

Description automatically generated

Note: above we can see the row variable is displayed as the categories of ‘area’ and the cellvariable are the values for mean\_size. In more practical terms, the values in mean\_size are the average house sizes (in sqfeet) for each the these different ‘areas’.

Example 2. I am interested in knowing how many houses have balconies in each neighbourhood grouping. I will now use the ‘sum’ command in egen.

. sort area

. by area: egen hasbalcony = sum(balcony)

. tab hasbalcony

A picture containing text, receipt, screenshot

Description automatically generated

. table area, cell(hasbalcony)

Table

Description automatically generated

# Stata Exercises: 2

Please complete the following tasks. Execute commands from a Do-file.

**Task 1:**

Create a new variable called ‘SqFeetperRoom’ that tells us the average square feet per room in a given home.

* What is the minimum and maximum values for this new variable?
* What is the mean for this new variable?
* What is the median for this new variable?

**Task 2:**

Using your renamed variables (np9501 np9502 np9503 np9504 np9506 np9507) from Stata Exercise 1, create a ‘Worry Index’. Once you generate the index, use ‘recode’ to create four (4) meaningful descriptive categories with corresponding labels for each of the categories of the Worry Index.

**Task 3**: Generate a new variable that captures the average party intensity (np9403) for each party group (np9402).

* Present a Frequency Table of your variable

# **Part 3.** **Visualising Data**

A more intuitive way to present results of univariate analysis is to visualize them using Stata’s graphing functions. Stata has a large number of graphing options. In fact, there are entire books dedicated to explaining all of these graphing options. At this stage we will only look at a few that are relevant for univariate analysis.

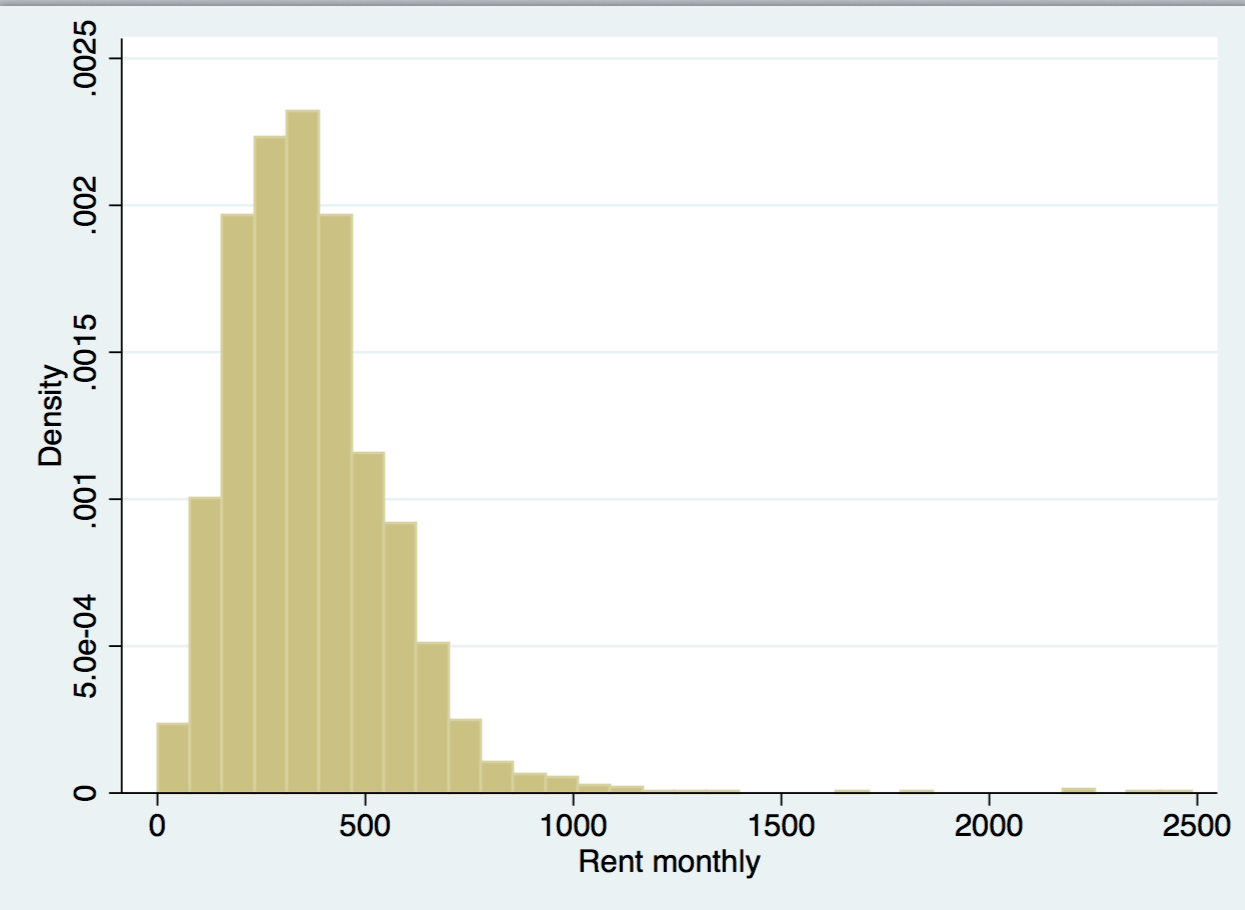
## Histograms

Histograms ‘are graphical displays of continuous variables with many outcomes’ (Kohler and Kreuter, 164) and plot information otherwise contained in frequency tables. They are not normally used to present results. Instead, histograms are great for giving us a very quick look at how data is distributed. Here is the general Stata syntax.

. histogram *var*, options

Example. I want to visualise the distribution of values for the variable ‘rent’ in a histogram.

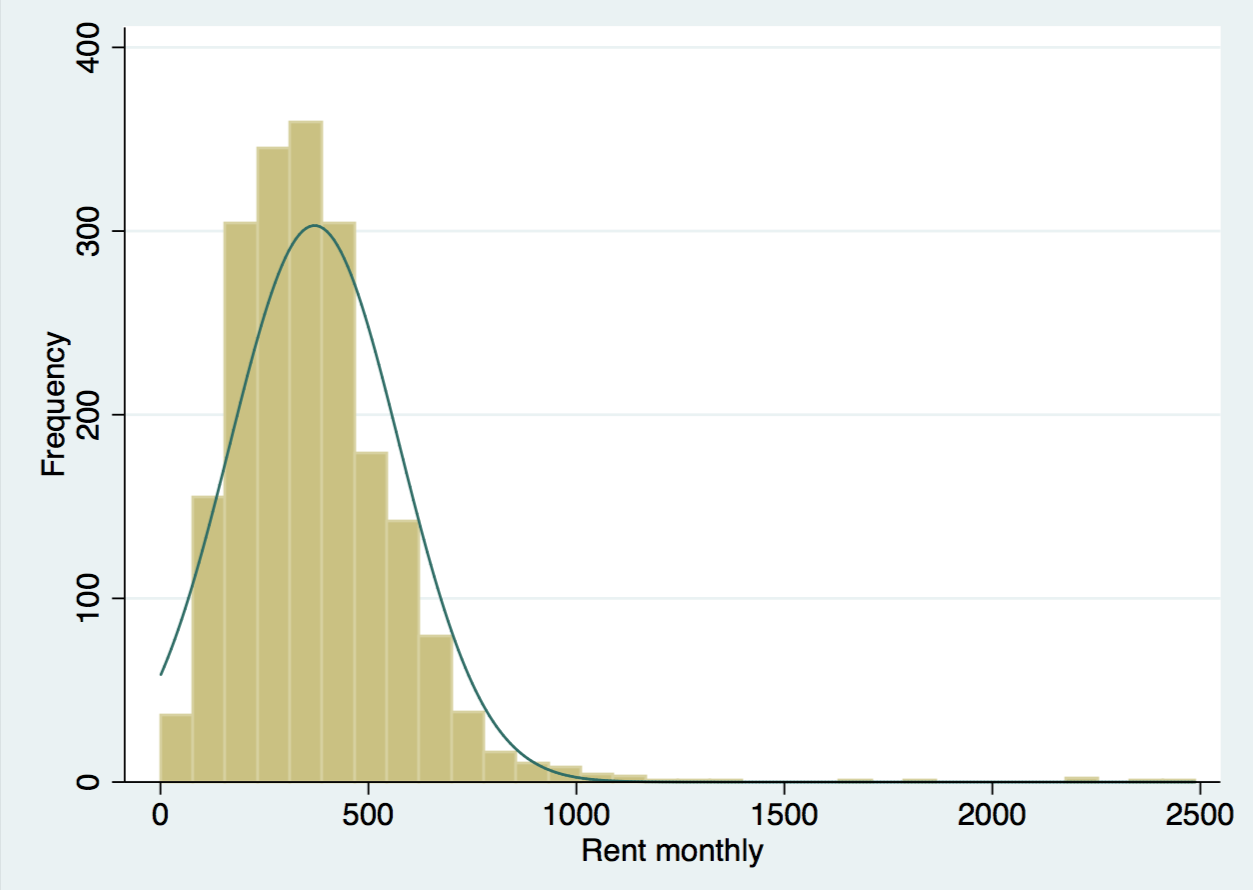
. hist rent



We can see that while the bulk of the values are normally distributed (a tight bell curve), there are some outliers (especially those values above 1000). We’d want to make special note of these outliers if we are conducting any type of analysis in the future as they may skew our results. In fact, we will learn how to handle this ‘outliers’ next week.

To get an even better picture of the distribution of our data, we can add a normal distribution curve to our plot.

. histogram rent, frequency normal



## Box and whisker plots

Box and whisker plots are very useful for visualising results for summary statistics, especial data related to percentiles, median, and outliers values.

The basic Stata syntax looks like this.

. graph box *var*

. graph box *var*, over(*var*)

Example 1: Create a box and whisker plot for ‘income’

. graph box income



Annotated Box and whisker plot



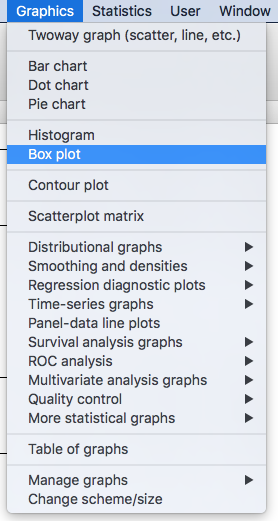
**Example 2**: I would to create a boxplot for income but compare differences between East and West Germany. For this this we will need to use our new ‘East\_West’ variable that we created earlier.

. graph box income, over(East\_West)

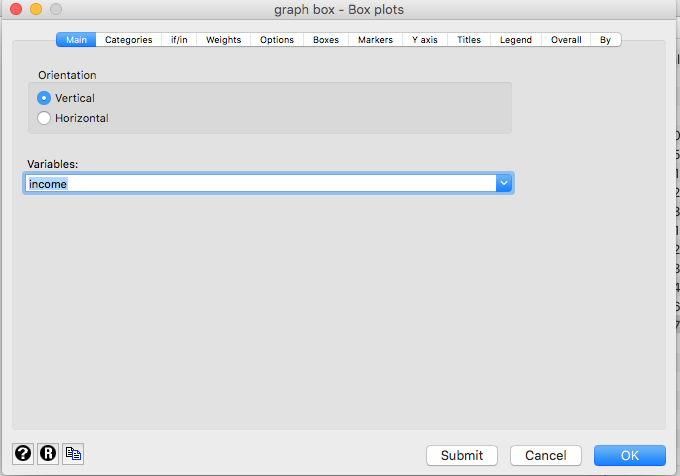


**\*\*Pro Tip**: Graphing in Stata is one instance where the dropdown mean can be very helpful.

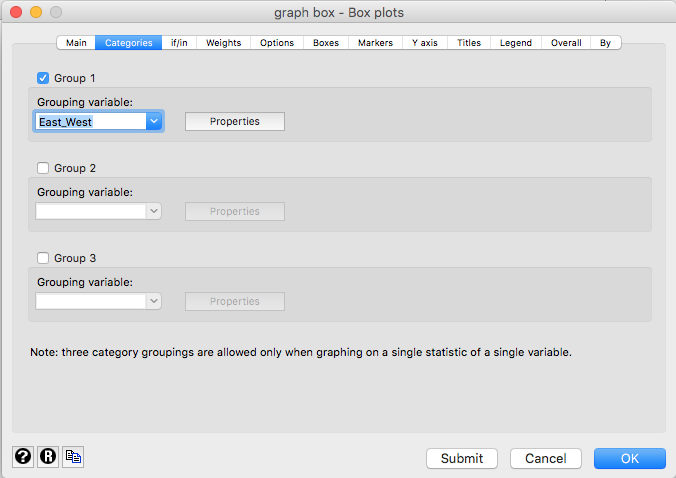
Step 1. From the dropdown menu, select Graphics, and Box plot.



Step 2. In the ‘Main’ tab, enter ‘income’ into the Variables list

****

Step 3. In the ‘Categories’ tab, click on Group 1 and enter ‘East\_West’. Then click OK.

****

Click ‘Submit’ to produce your graph. It should look like this.



## Dot charts

Dot charts are also a powerful way to visualize summary statistics (especially means). Let’s create a dot chat visualising income for each of Germany’s 16 Bundesländer (state)

. graph dot (mean) income, over(state)



Using a few of tricks we have already learned, we can create a Dot chart of income over state by gender

. gen WomenIncome = income if gender == 2

. gen MenIncome = income if gender == 1

. graph dot (mean) WomenIncome MenIncome, over(state)



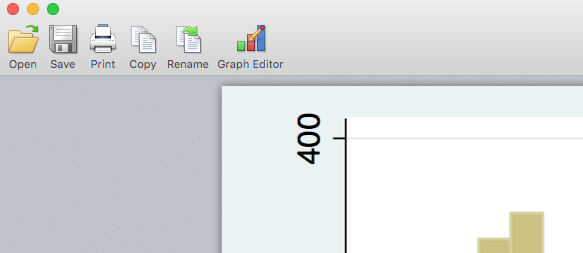
You can make the Dot Chart a bit easier to read if you sort the values by WomenIncome.

. graph dot (mean) WomenIncome MenIncome, over(state, sort((mean) WomenIncome))



## Editing Graphs

As we can see above, our graphs are not very aesthetically appealing, and the text is not right for presentation in a report or article (e.g., ‘mean of WomenIncome’). The easiest way to edit graphs is by using the Graph Editor. Note that the Graph Editor can only be used after you have generated a graph in Stata. It can be found at the top, left-hand side of the graph window.



After you click on ‘Graph Editor’ you will enter a new window where you can make any number of changes to the appearance of your graph. The best way to learn how to do this is by giving it a try. Just click on the element you wish to change, and Stata will present you with any number of options. It can be a bit tricky, but a bit of ‘trial and error’ will give you the basic tools for making aesthetically appealing graphs.

# Stata Exercises. 3

Please complete the following tasks. Execute commands from a Do-file.

**Task 1:**

As we saw above, there are some outliers in the variable ‘rent’ that skew the distribution. Let’s see if we can normalise our distribution by dropping outliers. After dropping outlier, create a histogram with a normal curve to assess if you’ve improved the distribution of the curve.

**Task 2:**

Use a box and whisker plot to visualise the distribution of ‘SqFeetperRoom’ by ‘HouseType’. Which has the highrer outlier, 'new houses' or 'old houses'?

**Task 3:**

Using a Dot chart and your new Worry Index, visualise levels of worry over:

* marital status
* gender

# Advanced Stata Exercises

**Task 1**:

Visualise the distribution of the age of individuals living in different types of neighbourhoods (area) using a box and whisker plot. However, be sure to:

* not include the following categories for area: ‘other’ and ‘shops and houses’.
* Rename ‘ancienty houses’ to ‘ancient houses’.

**Task 2**:

Create an index for all the amenities in a given home including: kitchen, shower, wc, heating, cellar, balcony, garden, and phone. Pay attention to how each indicator is measured.

* What is the median value?
* What is the minimum and maximum number of amenities?
* What is the average number of amenities for ‘ancient houses’?
* How evenly distributed is your new variable? Use an appropriate graph to visualise the distribution of this variable.