Lab 1: Descriptive statistics

Erik Thorsén (earlier version by Benjamin Allévius and Fredrik Olsson)

2018-10-31

Important: Before you read on

First do the following:

1. In the top menu of RStudio, press Tools --> Global Options... --> Code --> Saving. In the box that appears, where it says "Default text encoding", press "Change" and select "UTF-8".
2. Go to the course website and download the template for this laboratory. Open it in RStudio. Write yourselves report in this file.

Requirements for laboratory 1

Consider the following **requirements** for your report:

- The report must be readable by someone who has not read the lab instructions. So you have to write what that is what you must do before you do it, and tell what the purpose is.
- The report must be written in **R Markdown** and compiled into a pdf. This can be done by paste the following code snippet at the beginning of your .Rmd file

  ```
  ---
  title: 'Example Title' author: "John Doe"
  date: '2018-10-31'
  output:
    pdf_document
  ---
  ```

  Now click on the knit button to get a .pdf instead of a .html file.
- All code used must appear in the lab report, but should not be described in detail in the report.
- All **tables and figures** must be **numbered and descriptive** and must be properly referenced in the main body of the report. Charts must have appropriate axis headings and tables appropriate column headings.

This lab

This first computer lab in the course **Statistical analysis** essentially consists of three different parts:

1. An introduction (or memory refresher) to R, which will be used in all labs.
2. An exercise in determining whether a random sample can be considered normally distributed.
3. A descriptive (descriptive) analysis of a small data material.

In preparation for the laboratory, you must read through the introduction to descriptive statistics and solve the theoretical Task (Task 1). These do not need to be reported in writing. Tasks 2 and 3 must be reported in writing no later than the date indicated on the schedule.

Introduction to Descriptive Statistics in R

In the course Probability Theory I you got an introduction to R. All the documents used in this introduction can also be found on the website for this course, so you can take a look at them to refresh your memory. Here we will instead look at what R has to offer when it comes to describing data sets. To get an idea of how you can describe a data material numerically and graphically, you must complete the following simple task.

In the city of Grötköping, the body height of 11 of the city's residents was measured and the results (in cm) were

[1] 174.6 173.2 189.6 167.7 179.2 179.6 170.5 168.5 185.3 164.1 178.4

Begin by entering data into the variable x. The usual measures of mode and dispersion can be easily obtained using the functions mean (for the mean), var (for the sample variance), sd (for the sample standard deviation), and summary (for the minimum, maximum, median and quartiles).

A tree-leaf diagram is obtained with stem, histogram with hist, boxplot with boxplot, and normal distribution plot with qqnorm. In all cases, it is enough to specify the vector x as the only argument, and R draws the diagram automatically. However, we want you to put appropriate headings on the axes when you create charts—to see how to do that, look at the help page for the plot function, e.g. by typing ?hist in Console in RStudio.

When it comes to histograms, you will sometimes want a different classification than the one that R provides automatically. Try, for example ,

```
hist(x, breaks = seq(from = 162, to = 192, by = 5))
```

and

```
hist(x, breaks = seq(from = 162, to = 190, by = 4))
```

and see what happens. Here we see that we specify the break points (argument breaks) for the histogram by specifying a vector which in the second case is a sequence of numbers with 162 as the first value, 190 as the last value, and with numbers in between that have the distance 4 to the numbers just before and after.

As for the normal distribution plot, it is constructed in such a way that the data will lie along a straight line if the data is truly normally distributed. Such a comparison is facilitated if you draw a straight line, which can easily be achieved with the qqline(x) command. That is, you write **qqnorm(x) qqline(x)**

Assignment 1: Two theory questions (does not need to be presented in writing)

1. If an exponentially distributed random variable has expected value a, what is its standard deviation?
2. The random variable *X* is uniformly distributed with expected value a. What should the upper and lower limits of the distribution be, expressed in a, for the standard deviation to be as large as the expected value? Corollary: If *X* ÿ *U[ÿ,* ÿ], then Var(X) = *(ÿ* ÿ ÿ) 2/12.

Note: the notation *U[ÿ,* ÿ] is equivalent to the notation Re[ÿ, ÿ] or Re(ÿ, ÿ) you have encountered before. The uniform distribution is called "the uniform distribution" in English, hence the U. Sometimes you also see Uniform(ÿ, ÿ) or Uniform[ÿ, ÿ].

Task 2: Does the data come from a normal distribution?

In the future, when you will deal with practical applications of mathematical statistics, you will probably be faced with the question of whether a set of data can be considered to come from a normal distribution (possibly with some approximation). In that case, you thus (or hopefully) have a random sample of *n* independent observations from an unknown probability distribution. The question is whether the unknown distribution can be a normal distribution.

Here we will compare different methods (mainly graphical) that can be used to answer the above question. We will also try to answer the question of how large *n* needs to be in order for us to be able to determine with a reasonable approximation whether the data is normally distributed or not. For this purpose, we will simulate data from distributions known to us, both normally distributed and non-normally distributed data. All distributions that you will compare must have an expected value and standard deviation equal to a, where *a* is the last two digits of your social security number (if you work in pairs, choose the social security number of one to work with).

Data requirements

The questions you must answer in this Task are the following:

1. What is the minimum sample size needed for the **distribution** you are simulating from reveal themselves as normal or non-normal?
2. Which graphical method do you think is most effective for determining whether a sample is normally distributed? Justify with the various graphic methods.

In the sub-questions that follow below, we simulate from different **distributions.** In subtask 1, we also introduce you to the graphic methods you need to be able to complete the task. Your answers to the above questions must be reported in writing, subtasks 2–3 below. Answer the questions in running text, not in list form. Justify your answers properly.

We set the following **requirements** for task 2:

1. **Only show plots for one value of your sample size** (any of the ones you have to choose from). However, you have to look at plots for different values of the sample size on your own to answer the questions.

2. In the diagrams you include in the report, you must enter appropriate **headings on the axes** and **number these diagrams and give them descriptive texts!**

Exercise 2.1: Normally distributed data

Normally distributed data can be easily simulated in R with the function rnorm(n, m, s), where n indicates number of values to be simulated, m indicates expected value, and s indicates standard deviation of the normal distribution to be simulated from. Start by simulating a random sample of size 10 according to

```
set.seed(19880210) # fill in your own date of birth. If you work in pairs, choose the one. x1 <- rnorm(10, a, a)
```

where you have assigned a the value as above (and filled in your date of birth as an argument to the set.seed function).

We can now make a comparison between the random data we just simulated and the true normal distribution, by plotting a histogram of density on the y-axis, and superimposing the graph of the density function of the normal distribution:

```
hist(x1, prob = TRUE) x <-
seq(from = low, to = high, length.out = 100) lines(x, dnorm(x, a,
a))
```

Here you need to define the values of low and high, which are the lower and upper limits, respectively, between which the normal distribution will be plotted in the histogram. Find suitable values for these. When we write lines(x, dnorm(x, a, a)) we draw a curve with x-coordinates defined by the variable x and y-coordinates dnorm(x, a, a), which is the value of the density function at the points x for a normal distribution with expectation value and standard deviation both equal to a. Also draw a boxplot and a normal distribution plot for the data: # *Boxplot: https:// en.wikipedia.org/ wiki/ Box_plot*

```
boxplot(x1)


# Normal distribution plot (QQ plot): https:// en.wikipedia.org/ wiki/ Q%E2%80%93Q_plot qqnorm(x1)
qqline(x1)
```

Then simulate another seven random samples x2, x3, . . . , x8 of size 10 with the same expectation value and standard deviation as above:

```
set.seed(19880210) # fill in your own date of birth. If you work in pairs, choose the one. x1 <- rnorm(10, a, a) #
will be the same sample as above as we have the same value in set.seed x2 <- rnorm(10, a, a) # this will be a different sample
from x1, likewise the below x3 <- rnorm(10, a, a) # etc (fill in the rest yourself)
```

Start by comparing the samples with a common boxplot according to

```
boxplot(x1, x2, x3, x4, x5, x6, x7, x8)
```

Notice that the eight samples appear to differ quite significantly even though they are all simulated from the same distribution. To get a corresponding comparison between histograms, we must first tell R to divide the graphics window into smaller subwindows, then give the commands to plot histograms, and finally tell R to stop expecting more plots for the same window: old_par <- par(mfrow = c(2, 4)) # *2 rows, 4 columns* hist(x1) hist(x2) hist(x3) # *etc (fill in the rest*

```
yourself)* par(old_par) # tell R to stop expect more plots to the same window
```

Problem 2.2: Uniformly distributed data

For simulating uniformly distributed samples, the function runif(n, min, max) is used in R, where n indicates the sample size, min and max indicate the lower and upper limits of the interval for which the distribution is defined. The solution to Problem 0 gives you the range limits for the value of *a* that you use. Use the runif function to simulate five independent samples u1, u2, etc of size 10 and compare them

graphically; partly with each other and partly with the normally distributed random samples above. For example, plot five normally distributed and five uniformly distributed samples in the same plot.

When simulating the five independent samples, first use the set.seed function in the same way as for the normally distributed samples above, ie use set.seed with your date of birth **once** above the definitions of u1, u2 etc, in the same piece of code. Repeat your analysis with sample sizes $n = 20$, $n = 100$ and any additional value of n that you choose yourself. Then answer the questions previously asked.

Exercise 2.3: Exponentially distributed data

Perform the same comparison also for exponentially distributed samples using the function rexp(n, r), where n is the sample size and r is the intensity (1 through the expectation) of the exponential distribution. Come up with suitable names for these samples and be sure to use set.seed in the same way as above. Gradually increase your sample size and answer the questions previously asked.

Task 3: Exploratory data analysis

You will now examine a real data material with the graphic methods above as well as with so-called scatterplots that illustrate dependencies between variables in a good way. A scatter plot is simply what you get by writing **plot(x,** y)

and as you know from the course Probability Theory I, you can give more arguments to this function to make the plot look nicer. **Remember to put appropriate headings on the axes, and definitely don't forget to number the diagram and give it a descriptive text!**

On the course website is the file olvinsprit.csv which contains data on average consumption of beer, wine and spirits in some OECD countries. Start by saving the file in **the same folder** that you placed your .Rmd file for the lab report in, for example Documents/Kurser/statan1/Labb1 or similar. Make sure the file is saved as a .csv file and nothing else (such as .php). To be clear: save the file as olvinsprit.csv.
We can then read it into R through data <-

```
read.csv("olvinsprit.csv", header = TRUE)
```

Then create the four variables

```
country <- data$Land
beer <- data$beer wine
<- data$wine liquor <-
data$liquor
```

Now we can draw a scatterplot with the command

```
plot(beer, wine) # When doing this yourself, specify appropriate axis headings etc
```

which in this case gives the beer consumption and wine consumption for all countries. If you want to see more clearly which countries the points correspond to, write

```
plot(beer, wine) # When doing this yourself, specify appropriate axis headings etc text(beer, wine,
country)
```

then the names of the countries appear instead of the dots.

Now illustrate the distribution for the three variables separately. Answer the following questions, not in point form but in running text.

• Can the data be considered to come from the normal distribution? • How does Sweden compare to other countries? • Is Sweden extreme in any direction? • Which countries are extreme? • Are there common features of the extreme countries? • Does high consumption of one type of alcohol lead to high or low consumption of others?

Task 3 must be reported in writing. Summarize your conclusions about alcohol consumption in OECD countries.