UPPSALA UNIVERSITY
Dept of Matematics
Rolf Larsson - rolf.larsson@math.uu.se

Matematical statistics
Inference 1
fall 2022

LABORATION 1:

Descriptive statistics and one sample inference methods

1 Introduction

The purpose with this lab is to introduce the program R. We illustrate how it can be used for descriptive statistics, how to handle probability distributions, we give some guidance as to how to save plots, and how to read a data set from a file. Finally, we describe how R can be used to apply the most common one sample inference methods.

This lab can be seen as a help for solving part of hand-in assignment 1.

2 Getting started

The simplest way to start R in the computer rooms is to choose `RStudio` in the menu. You then get an environment with different windows similar to when running Matlab.

You can also download `RStudio` to you laptop from the web page `rstudio.com/products/rstudio/download/`.

Commands are written by the prompter `>`.

2.1 General commands, vectors and simple plottning

At first, we use R as a calculator. For example, you can test the following commands:

```
> 2+7
> 2/(3+5)
> sqrt(9)+5^2
> sin(pi/2)-log(exp(1))
```

Help for specific commands may be obtained by writing a question mark before the command. For example, try

```
> ?log
```

Vectors

R is very much built around vectors (as opposed to Matlab, which is based on matrix operations). For example, you can create the vector `x` with entries $7, -2, 5$ by writing

```
> x = c(7,-2,5)
```

You can then refer to, for example, the second element by writing

```
> x[2]
```

or to elements 2 and 3 using

```
> x[2:3]
```

Some simple vector algebra may be performed. For example, you can add 100 to all elements in your vector by writing

```
> x + 100
```

A vector `y` consisting of the numbers $1, 2, \ldots, 10$ may be created by writing

```
> y = c(1:10); y
```

Note that `y` is not displayed directly when it is defined, for this purpose you must refer to it again. A semi colon separates different commands given in the same row.

Another alternative is

```
> y = seq(1,10)
```

The latter command may also create vectors with specified distances between the elements, take for example

```
> z = seq(0,10,2)
```

Another useful feature is that you can put two or more vectors together to form a longer vector. For example, try

```
> c(seq(1,10),seq(11,21,2),25,30)
```

Simple graphics

Functions may be plotted by plotting suitable vectors. For example, to plot the function $y = \sqrt{x}$ for $0 \leq x \leq 10$ you can write

```
> x = seq(0,10,0.2); y = sqrt(x); plot(x,y,type='l');
```

Observe that adding `type='l'` in the plot command results in tying the points in the plot together with a line. Try what happens if you plot without adding this specification!

Distributions

As an alternative to using tables, in R it is possible to calculate density functions, probability functions and distribution functions for specified standard distributions. Let us take the normal distribution as an example.

Say that you want to check the value of the standard normal distribution function at the 'famous' value 1.96, $\Phi(1.96)$. Then, you can write

```
> pnorm(1.96)
```

Does this give the result that you expected?

You can also calculate the distribution function of non standard normals by specifying μ and σ in two additional arguments. For example, write

```
> pnorm(4.92,1,2)
```

Are you surprised by this result?

By using `qnorm`, you can also invert the distribution function in a simple way, to get quantiles. Try what happens if you write

```
> qnorm(0.975)
```

In this way, you actually get the $1 - 0.975 = 0.025$ quantile. If you want to get the 0.025 quantile directly, you can write

```
> qnorm(0.025,lower.tail=FALSE)
```

The value of the density function may be obtained by using the command `dnorm`, but this is most often of less interest.

It is also possible to simulate data from a specified distribution, like normal. To simulate 100 data points from the standard normal distribution and store them in a vector `z`, write

```
> z=rnorm(100)
```

Now, the proportion of data points in z that are smaller than 1.96 should be close to 0.975 (why?). You can check this by writing

```
> sum(z<1.96)/100
```

This command works because `z<1.96` gives a new data vector of length 100 containing ones for those `z` that are smaller than 1.96, and zeros for the others. Hence, summing over this vector gives the number of `z` that are smaller than 1.96, and dividing by 100 gives the corresponding proportion.

Now, try to see if you get closer to the correct proportion 0.975 by increasing the sample size 100 to greater numbers like 1000, 10 000,...

3 Analysis of a data material: trees

Some data materials are directly accessible in R. We are going to exemplify here with the material `trees`. If you simply write

```
> trees
```

you will find displayed a material with 31 observations and three columns. These columns display measurements of the diameter, height and volume of timber in 31 felled black cherry trees. Note that the diameter (in inches) is erroneously labeled Girth in the data. It is measured at 4 ft 6 above the ground.

This information, and more, will be given in your lower right window if you write

```
> ?trees
```

In order to simplify in the following, let us now download the three columns by writing

```
> d=trees$Girth
> h=trees$Height
> v=trees$Volume
```

3.1 Descriptive statistics

The information in data may be summarized by using simple measures like mean, median, standard deviation, quantiles etcetera. A print of some of these measures, e.g. over our variable v, is obtained by writing

```
> summary(v)
```

We may also study the measures one by one, e.g.

```
> mean(v); median(v); quantile(v); max(v)
> var(v); sd(v)
```

which give, in turn, the mean, median, the quantiles, the maximum, the variance and the standard deviation.

3.2 Data visualization

Histogram

A histogram can be used to study the distribution of real valued (continuous) data. To get a histogram over the variable `v` with an automatic choice of the number of classes (or interval widths), we may simply write

```
> hist(v)
```

which in this case gives seven classes, each of which has width 10. To get another partition, we can e.g. write

```
> hist(v,10)
```

which in this case turns out to give 14 classes (not 10!), each of width 5. In case we want the relative frequency rather than the frequency (count) on the y axis, we may write

```
> hist(v,freq=FALSE);
```

For more information about the command `hist`, write

```
> ?hist
```

It is often of interest to check if the data are normally distributed. If so, the histogram should resemble the shape of the normal density. Do you think it does so for your variable `v`? Also, plot histograms over the other variables (`h` and `d`). Does any one of these appear to be more 'normal-like'?

QQ plot

A simpler alternative to check normality of data (that avoids specifying the number of classes or something similar) is the QQ plot. This is a plot of the sample quantiles (on the y axis) vs the so called 'theoretical quantiles' (on the x axis), which are the estimated quantiles under the assumption that the data are normally distributed. If the normal assumption is true, the dots in the QQ plot should be close to a straight line.

To get the QQ plot for the variable `v`, write

```
> qqnorm(v)
```

Are the data close to a straight line here? Try the same also for the variables `h` and `d`. Which of the three variables seems to be closest to being normally distributed?

Saving the histogram/qqplot to a file

To save your histogram to a file, you can click on `Export` located above the window showing the histogram. Here, you can choose between saving the histogram as a pdf file by choosing `Save as PDF`. If instead, you choose `Save as Image`, you can save it as e.g. jpg or eps.

Box plots

The box plot is another way of visualizing a data material. One advantage is that it highlights the minimum, median, maximum and quartiles. If you write

```
> boxplot(v)
```

you can see what happens. You get a "box" that ranges between the lower and upper quartiles. with a thick line in between that marks the median. The "whiskers" of the plot either go out to the maximum/minimum or, if e.g. (as in this example) the maximum is "far out", the whisker goes to the observation which is most extreme within the range of the median plus/minus 1.5 times the interquartile range. Observations outside of this limit are separately indicated. In this example, we have one such observation, the maximum 77.

Note that 1.5 here is the default choice. We can change this value to something else. For example, check what you get if you write

```
> boxplot(v,range=1)
```

The box plot can be saved to a file in the same way as a histogram.

Stem and leaf plots

For discrete data, another visualization option is the stem and leaf plot. This is simply a list of the data which is organized in such a way that it gives the idea of the shape of the distribution (i.e. a form of turned over histogram that simultaneously shows the whole data set). In our data set `trees`, we have no discrete variables, but the heights, `h`, are rounded to integer values, so for now, let us "pretend" that these are indeed discrete.

A stem and leaf plot of this variable is obtained by writing

```
> stem(h,scale=0.5)
```

from which we get

```
  6 | 34
  6 | 569
  7 | 012244
  7 | 55566789
  8 | 000001123
  8 | 567
```

Here, `scale` controls the vertical length of the plot. (The default value is 1, and in our case, this value does not seem reasonable as it turns out to produce a very confusing result.)

4 Reading data from a file

To read a file in RStudio, you must at first get into the same directory as where the file is. Say that you have saved the file in the directory `Downloads`. At first, click on `Session`, and then go to `Set Working Directory` and `Choose Directory`. Here, you can choose the directory `Downloads`. Then, click on `Open`. If you have done this right, the names of the files that you have saved in `Downloads` will appear it the upper right window of `RStudio`.

Say that your file is called `test.dat`. If the first column of the file contains the variable x and the second contains the variable y, you can write

```
>x=read.table("test.dat")$V1
>y=read.table("test.dat")$V2
```

If you want to read data files that have headers, you can also consider the command `read.csv`, see further the help information that you can access by writing

```
> ?read.csv
```

5 One sample methods

5.1 Two-sided confidence intervals and hypothesis tests

There are simple commands in R that simultaneously deal with confidence intervals and hypothesis tests. For example, let us take data from "övning 7.6.1 c)" of the Alm and Britton book. The following gives a 95% confidence interval for the mean μ using one sample taken from the normal distribution with unknown variance. We also get the test statistic and p value for the test of $H_0 : \mu = 3.2$ vs $H_1 : \mu \neq 3.2$ (a two-sided test).

```
> x=c(2.0,3.2,3.8,2.5,3.3,2.8,3.0,3.4)
> t.test(x,mu=3.2)

One Sample t-test

data:  x
t = -1.0045, df = 7, p-value = 0.3486
alternative hypothesis: true mean is not equal to 3.2
95 percent confidence interval:
 2.529191 3.470809
sample estimates:
mean of x
        3
```

As you can see, the p value is 0.3486. It is up to the user to compare the p value with his or her own significance level, but in particular, we can see here that we may not reject H_0 at e.g. the 5% level ($0.3486 > 0.05$). The 95% confidence interval is about $(2.53, 3.47)$.

The 95% confidence level is default in R. If you want (for example) a confidence level of 99%, you can write

```
> t.test(x,mu=3.2,conf.level=0.99)

One Sample t-test

data:  x
t = -1.0045, df = 7, p-value = 0.3486
alternative hypothesis: true mean is not equal to 3.2
99 percent confidence interval:
 2.303235 3.696765
sample estimates:
mean of x
        3
```

Here, you can see that you get the same test result as before, but the confidence interval is now a little wider than the previous one. Why is that natural?

Now, check what happens if you change the confidence level to 90%. Do you get a wider interval than above, or is it more narrow? Why?

5.2 One-sided confidence intervals and hypothesis tests

To perform the test of $H_0 : \mu = 3.2$ vs $H_1 : \mu < 3.2$ (a one-sided test), and at the same time get the corresponding 95% one-sided confidence interval, you can write

```
> t.test(x,mu=3.2,alternative='less')

One Sample t-test

data:  x
t = -1.0045, df = 7, p-value = 0.1743
alternative hypothesis: true mean is less than 3.2
95 percent confidence interval:
    -Inf 3.37722
sample estimates:
mean of x
        3
```

Observe that the confidence interval here is about $(-\infty, 3.38)$. An alternative is to write `greater` instead of `less`. Check what this gives!

5.3 Discrete distributions

If you have a discrete distribution, e.g. Binomial or Poisson, it is easiest to obtain e.g. p values of a test by calculating the distribution function. This works in the same way as we saw in section 2 for the normal distribution. For example, for $X \sim \text{Bin}(10, 0.5)$, you can get $P(X \leq 4)$ by writing

```
> pbinom(4,10,0.5)
```

Compare this value the the corresponding figure in table 2! Similarly, you can obtain $P(X > 4)$ directly by writing

```
> pbinom(4,10,0.5,lower.tail=FALSE)
```

Observe: this is not equal to $P(X \geq 4)$.

Now, let us look at "övning 7.6.14" in Alm-Britton. We may obtain the P value there by considering $X \sim \text{Bin}(144, 1/6)$ and calculating $P(X \geq 32) = P(X > 31)$, from

```
> pbinom(31,144,1/6,lower.tail=FALSE)
```

At the 5% level, does this give empirical evidence that the dice is manipulated to give a six too often? (Observe that the intended method in 7.6.14 is to use normal approximation, but here we use exact calculation of the P value instead.)

Other discrete distributions are available in the same way. For example, for $X \sim \text{Po}(2)$, we get $P(X \leq 3)$ by writing

```
> ppois(3,2)
```

Compare this result to the corresponding figure in table 3!