Bielefeld University · Econometrics · Summer 2024

Statistical and Econometric Models

Lectures: Lennart Oelschläger · Tutorials: Sebastian Büscher ✉

Problem Set – Week 1 (for tutorial on April 22, 2024)

R Introduction

The statistical software R is used in this course. This decision is based on two essential properties of R.

- R can be downloaded free and legally from `https://cran.r-project.org/`,
- the range of functions is very large and is constantly being expanded,
- the RStudio integrated development environment (IDE) greatly helps with the workflow when coding in R `https://posit.co/download/rstudio-desktop/`.

There is a multitude of literature and online tutorials that help getting started with R, for example, *simpleR* by John Verzani[1], the official R introduction[2] or *The Art of R Programming* by Norman Matloff[3].

Problem 1.1: Obtaining and importing data
The functionality of commands can be displayed using the `help(command)` or `?command` functions (e.g. `?read.table`). Lists and explanations of the `arguments` of the respective functions can also be found there. Experimenting with commands and the associated arguments is expressly encouraged.

(a) Start R and create a new R script. Save the script in a directory that you want to use for the exercises in R. It may be useful to create a new folder for this, for example

`C:/SEM_2024/R`.

(b) Download the data set `Data_Income_Savings.csv` from the moodle and save it in the same folder on your computer.

(c) The data is saved in the form of a text file. What does the file extension `.csv` stand for? First look at the data with a simple text editor and describe the existing structure.

(d) Make the newly created folder the working directory in R (`?getwd` and `?setwd`).

```
setwd('C:/SEM_2024/R')
```

(e) Read the data into a variable `data` (`?'<-'` and `?read.table` with the arguments `file`, `sep` and `header`). What do the individual arguments stand for?

```
data <- read.table(file='Data_Income_Savings.csv', header=TRUE, sep=';')
```

(f) Which variables does the data set contain and how many observations does it consist of (`?str`)?

[1] `http://cran.r-project.org/doc/contrib/Verzani-SimpleR.pdf`
[2] `http://cran.r-project.org/doc/manuals/R-intro.html`
[3] `https://www.researchgate.net/publication/254296013_The_Art_of_R_Programming_A_Tour_of_Statistical_Software_Design_by_Norman_Matloff`

Problem 1.2: Analysis of the gross savings rate

(a) The data was read in as a *data.frame* (try `is.data.frame(data)`). The first N lines can be displayed with `head(x = data, n = N)`, a complete overview can be generated with `View(data)`.

(b) In R, many procedures can be simplified considerably if vectors (and matrices) are used. There are several options for creating vectors. First create two vectors:

- using the command `?c` to create a vector `a` containing the entries 6, 14 and 24,
- with the help of `?':'` another vector `b`, which contains the number sequence from 1 to 3.

As for numbers, the usual arithmetic operations are also implemented for vectors in R. Create a vector `d` that contains the difference between the entries of `a` and `b`.

The functions $*$ and / for multiplication and division can also be used for vectors. However, as in the case of + and −, these are element-wise arithmetic operations. Check this by dividing `d` by `b`. Save the quotient vector in a variable `p`.

(c) Using the square brackets (`?'['`) one can access subsets of the data. [4] Have R output the fifth to seventh row of the third column of the data frame `data`. To do this, use your vector `p` from task part (b). What are the values?

(d) If row or column labels are available (the latter as in our case), these can be used to select rows or columns using `?'['`. This can increase the readability of the code. In our case, the column name is entered in quotation marks (also possible as a vector) at the intended position. [5]. In this way, you can display the net savings rate of the fifth row.

(e) A typing error has crept into the name of one of the variables. Correct the error. (`?names`, `?'['` and `?'<-'`).

(f) Logical queries are possible in R. A simple example is `3 > 6`. Familiarise yourself with the variants listed under `?'<'`. Also compare the vectors `a` and `b` with each other. For `a < 15`, a vector is compared with a number. This functionality is called *recycling*. So what happens here?

(g) The command `which(x)` can be used to query which entries of a *logical* vector `x`, which only contains `TRUE` and `FALSE` as entries, are actually `TRUE`. Use the comparison `a > b` to create such a vector `x` and use `which` to specify the entries for which the condition is fulfilled. Is the output of `which` also a vector (`?is.vector`)?

(h) Now you should only output the countries whose net savings rate is above 10%. To do this, first create a logical vector `y` with the help of the above queries. We have already seen that vectors can be used within the square brackets to access only a subset of the data. Use the vector `y` to output the countries whose net savings rate is above 10%. Then output them again, but this time using the function `?which`. Describe the different effects of the two methods.

(i) The data set has net and gross variables. Calculate the gross savings rate using the elementwise division of two vectors from task (b). Save this in a new variable `Savingsrate_gross`

[4] In order to be able to select such subsets, it is necessary to know the dimension of the data. In the case of matrices and the data frame here, there are two dimensions (rows and columns), the lengths of which can be displayed using `?dim`.

[5] If you only want to look at one column, you can also use the symbol $: `data$Country` to display the country column as a vector. The quotation marks are then omitted

and bind it to the existing data set (`?cbind`). By a savings rate, we mean the proportion of income saved as a percentage.

(j) Generate a plot of the gross savings rate (`?plot`). How could the plot be improved (sort by size using the command `order` and use the argument `type` of the function `plot`, for example)?

(k) Various statistical location and dispersion parameters are now to be used to analyse the gross savings rate. Calculate the following sample statistics: Mean, median, minimum, and maximum.

(l) Now you are to calculate the sum of the squared deviations from the mean value of the gross savings rate in two different ways. To do this, use the element wise operations for vectors from task part (b) and for the first variant the function `?sum`, for the second variant the scalar product of two vectors, which can be calculated using `?'%*%'`.[6] Calculate the empirical standard deviation of the gross savings rate using the sum of the squared deviations. The length of a vector can be determined using `?length`.

Distributions and conditional expectations

Problem 1.3:
The data set `module_exam.csv` contains the scores of a module examination.

(a) Save the data set `module_exam.csv` in your working directory and load it as a `data.frame` into an object `scores`. (`?read.table`)

```
scores <- read.table(file = 'module_exam.csv', header = TRUE, sep = ',')
```

(b) The data set contains the scores achieved in a module examination divided into the two examination parts. First get a general idea of the data set. Then use the command `?cut` with the group boundaries `c(0,5,10,15,20,25,30)` and divide the data of each examination part (individually) into these six groups. Save these in the variables `dist_part1` and `dist_part2`.

(c) Create one table for each part of the test in which the frequencies of the individual groups are shown (`?table`) and one common table. The result is shown in the table below for comparison::

Table 1: Module examination results

Score	0-5	6-10	11-15	16-20	21-25	26-30
Part 1	0	11	58	39	53	38
Part 2	0	9	47	76	58	9
Grade	6	5	4	3	2	1

(d) In the following, let $\boldsymbol{x} \in S = \{1, 2\}$ be a random variable indicating the exam part. Furthermore, let $\boldsymbol{y} \in T = \{1, 2, 3, 4, 5, 6\}$ be a random variable indicating the grade, where the elements $t_j \in T$ stand for the individual grades. Calculate the joint distribution $\mathbb{P}(\boldsymbol{x} = s_i, \boldsymbol{y} = t_j),\ i = 1, 2,\ j = 1, \ldots, 6$ of module examination part and grade.

(e) Calculate the marginal distribution $\mathbb{P}(\boldsymbol{y} = t_j),\ j = 1, \ldots, 6$ and the expected grade.

[6] For inner and outer products, see `?'%*%'`, `?'%o%'` and `?t`. These commands can also be applied to matrices.

(f) Now calculate the conditional distributions $\mathbb{P}(\boldsymbol{y} = t_j \mid \boldsymbol{x} = s_1)$ and $\mathbb{P}(\boldsymbol{y} = t_j \mid \boldsymbol{x} = s_2)$, $j = 1, \ldots, 6$.

(g) Now also calculate the conditional expected values, $\mathbb{E}(\boldsymbol{y} \mid \boldsymbol{x} = s_1)$ and $\mathbb{E}(\boldsymbol{y} \mid \boldsymbol{x} = s_2)$, and compare them with the unconditional expected value from task (e).