ADS2 Practical 1: R Refresher

Dmytro Shytikov and Duncan MacGregor

2023-09-21

Work through this guide alone or in groups. Facilitators are here to help. The time it takes to complete this practical can vary between individuals – this is OK. Do not worry if you do not finish within the session.

Learning Objectives

- Refresh your skills in R
- Learn how to manipulate files using R, how to import and export files
- Learn about different types of data structures in R and their conversion from one to another

Working in R

In IBMS1, you learned how to use R and RStudio. We will use R in this course, so here is a quick refresher. Are you able (or still able) to do the following?

Let's start with something very easy. Open RStudio and run a command, for instance

```
print("Hello world!")
```

You can assign a command to a variable so R can execute this action upon request:

```
var.a <- print("Hello world!")
var.a
```

As you can see, the result is very similar to Python that you studied last year.
Now, let's try simple calculations:

```
1+1
```

What did you get as a result?

You are free to try other mathematical functions as you wish. You can also wrap several commands in a single chunk of code that will be executed consequently. Do you remember how to do that? Try to create several consecutive commands:

```
{command.1;
command.2;
command.3}
```

If you copy this exact code on its own it will produce an error. How can you make it work?

Operations with files in R

R has numerous simple mathematical functions, but the main use of R is to work with external files that contain information to analyze. You can manipulate files using R from the console directly.

Files that are manipulated by R directly must be located in the working directory. Otherwise, the path to the file must be mentioned explicitly.

The command to retrieve the current work directory is `getwd()`. Changing the working directory is done by the command `setwd()`. Try to identify your current working directory and change it using the respective commands from the prompt and using the respective buttons in R Studio or R GUI (if you use GUI). As you may imagine, it is wise to keep your working folder tidy so the data are easily accessible.

When you work in R, you can get a list of files in the requested directory (the same as a folder) with `dir()`. Apart from using this command to retrieve the list of files, it can help to find particular types of documents using the `pattern` argument or search in a particular location `path`. Try to search your files using the `dir()` command with or without the above-mentioned arguments. Think and discuss how it can help in your work:

```
dir(pattern = ".csv", path = "Path to the file")
```

Compare the `dir()` command with the `list.files()` and `list.dirs()` commands.

You can also create and modify files in the console. Create a file using the console, check its existence, and content, and edit it using the following commands:

```
file.create()
file.exists()
file.info()
file.show()
file.edit()
file.rename()
file.remove()
scan()
```

Also, discuss a possible usage of these commands. Can you find the command to create a new directory?

If you have any questions, you can always ask for help from your instructors or from the program itself by typing `?Function.name` (obviously, you should write the exact command or package name; try to search the possible ways to get help if you want to get help, but do not know the exact name of the command). The latter variant will give you a more detailed description of the requested function.

Creating new vectors

Try to name the data types and structures that you know and their possible uses. Don‘t worry if you can name just a few of them. There will be a lesson specifically aimed at discussing this question in the coming weeks.

Test your knowledge by doing the following:

- Concatenate a vector consisting of 5 numbers: 5, 6, 4, 5, 10 and assign it to a certain value.
- Add “1” to all elements of the created vector.
- Extract the first element of the vector, add 3, and add the resultant number as the sixth element of the vector.
- Take the square root from the resultant vector and assign it as a new vector.
- Convert it to a matrix with 3 columns.
- Subtract the first four elements of originally created vector from the newly converted matrix and round the result to the second decimal. What is the result?

Working with tables in R

There are many types of data structures in R. You have just worked with vectors, very basic data structures. One of the most important types of data in R is data frames and matrices. Discuss the differences between these types of data and explain your opinion. What might they be useful for?

There will be a more detailed class about working with different types of data later, so we will cover some of the questions today only briefly. In short, matrices and data frames are tables where you can store multiple types of data and use them for data analysis.

Suppose you want to analyze tumor incidence in two groups of mice, one treated with a chemical (32 mice) and the other treated with a solvent (32 mice). 17 out of 32 mice in the treatment group and 7 out of 32 mice in the control group developed tumors by the end of the experiment.

Combine these data into the annotated matrix using `matrix(), rownames(), and colnames()` commands. You will learn how to analyze these data later, so no need to go further on the analysis. Just create a matrix and investigate how these data look, and what are their properties, using commands `mode(), class(), and str()`. Try to add some statistics into this matrix and a separate column with the group labels. Discuss possible applications of matrices.

More often than the matrix-like types of tables, you will use data frames, which are more convenient to store a wide variety of information. You have data about the percentage of double positive (DP, CD4^{+}CD8^{+} T-cell precursors) and double negative (DN, CD4^{-}CD8^{-} T-cell precursors) in mice with a gene knock-out (KO) and normal wild type (WT) mice. Here are the data:

DP:
KO – 57.27, 72.53, 64.67, 65.96, 55.99, 64.02, 65.60, 55.15, 72.90, 59.95
WT – 78.99, 77.63, 80.69, 83.49, 80.49, 80.79, 82.11, 80.10, 81.63, 78.82

DN:
KO – 19.72, 19.57, 26.02, 18.37, 12.35, 21.40, 17.75, 21.86, 15.07, 14.70
WT – 5.76, 3.80, 3.78, 7.21, 8.92, 5.75, 9.32, 5.82, 7.40, 10.69

Concatenate these data into a data frame with the name `thymocytes` (for example) and check its structure. There are several ways to do it:

1. By the direct command `data.frame()` and specifying the columns;
2. By concatenating vectors and combining them into a data frame with several commands `c() -> cbind() -> as.data.frame()`;
3. By creating a table in an external program and loading it in R. We will discuss this method later. Try the first two.
4. You are welcome to find out other possible methods!

You will learn more about how to deal with these data later, so no need to go further on the analysis. Just create a data frame and investigate how the form of these data and their properties using commands `mode(), class()`, and `str()`. Add a column that depicts the amount of single positive (SP) thymocytes (either CD4^{+}CD8^{-} or CD4^{-}CD8^{+} cells) into the data frame.

Also, add one more column which is non-numeric (the results of dissection):

- WT – all 10 mice did not have anything special;
- KO – Splenomegaly, Splenomegaly, Normal, Normal, Normal, Splenomegaly, Normal, Splenomegaly, Splenomegaly, Normal.

Again, what are the properties of the data frame? Use commands `head(), tail(), mode(), class(), and str()`. Discuss it with your instructors. Save this workspace so you may be able to load it later and use it for the problem set. You may use `save(file = "filename.RData")`, where `filename` is the name of your file. This command will store all the objects you have in your scope in the `.RData` file that can be loaded later. Of course, you can export your data into a `.csv` table by the `write.csv()` command. Or you may save your data in any other way you prefer.

Practice!

Hopefully, you are getting comfortable with the basics of R and R Studio again. The important thing is to practice, and the next step will be the problem set. However, the best motivation is often your own projects. Is there some data you want to analyse on something that interests you? There are lots of publicly available data sets on the internet. Maybe you can find an interesting one to practice with.

Originally, created by D Shytikov and DJ MacGregor in 2022.

Last update by D Shytikov in 2023.