R-Bootcamp: Assignment / Group Work

Claude Renaux and Co.

26. August - 29. August 2024

Admin

In order to obtain the credits for the course "R-Bootcamp" students must provide evidence of their successful participation. To do that, students must hand in a document, hereafter *the assignment*, where the tasks listed below are carried out.

Find a use case that comes with some data

Roughly speaking the assignment represents a complete data analysis where you use a wide variety of R functionalities. The first task is to find an interesting use case that comes with data. The choice of the case is completely up to you. Nevertheless, makes sure the following criteria are fulfilled:

General comments

- The use case comes with some data and you should have two (or more) data sets which have to be joined / merged. You will learn how to join two data sets in the course. See examples below.
- The data can be publicly available or come from e.g. your employer. If the data is not publicly available make sure that you can use the data and discuss the results with the course instructors.

First data set

- The data set must contain at least a few hundred observations and a dozen variables.
- Among the variables there must be numeric and categorical ones.
- The data set should also contain variables that are dates (e.g. as YYYY-MM-DD and not jus the year) or geographic locations. Both type of data are even better.

Second data set (to be joined)

- Find a second data set which enriches the information of the first data set.
- If the first data set itself does not contain dates, geographic locations, categorical variables or any other information, you must find another additional data set with this information.
- Can be simple. E.g. join the average temperature or other weather information with the first data set or enrich your data with coordinates of locations from the first data.
- Join the two data sets and highlight in your report where you have done this. This means showing the line of code where the joining is performed.

Ideally, your data sets should come into different formats (e.g. xlsx, zip, . . .) and from different sources (directly from websites via url, locally downloaded files, . . .).

Example of a use case I:

We want to model ice cream sales in Switzerland. So we get the data about ice cream sales in each Swiss city during the past 5 years (say weekly sales provided by Friscolino AG), data about climate (i.e. day temperatures and precipitations e.g. from MeteoSuisse) and a data about city populations (e.g. from the Swiss Federal Office of Statistics). We will then merge all these data sets into a single data set.

Example of a use case II:

We want to model train arrival times in Switzerland. So we get the departure and arrival times of all SBB trains (e.g. from Puntlichkeit.ch), we then get the stations coordinates and we also get the local holidays (cantons homepage).

Prepare the data for the analysis

The data sets must be cleaned before they are merged and analysis begins. You can hide the code, e.g. in the html report, but please show the line of code where the data is merged. In both situations where you hide or do not hide the code, please explain in words (i.e. as text, not as comments in the code) what you are doing or have done.

Visualise the data appropriately

Once the data have been prepared, the first step is to analyse them using summary statistics and (most importantly) graphs. The latter is a requirement for your assignment.

Some recommendations for plots:

- Avoid using pie charts.
- Use the alpha level to add transparency to points / lines.
- Avoid using too many stacked bar plots (or do not use them at all and opt for, say, line plots).
- Show the entire range of values on the axes, without zooming (in or out) too much.
- When comparing graphs, use the same scale on the axes, otherwise it could be misleading.
- Show all the observations or additional information than just a mean and standard deviation.
- Do not put too many dimensions in a single graph: it should not be too complicated to read.

Fit model(s)

Modelling is not the focus of this course. Therefore, do not invest time in finding fancy models. A very simple model will do. Note that it is not enough to add a summary statistics to a plot as we want you to call a separate dedicated function like *lm()*, *t.test()*, *glm()*, etc. and look at the results.

You may want to produce graphs (e.g. predictions or residual diagnostics) for your model fits.

If you wish, you can also compare several models via CV. However, remember that the focus of this course in not the modelling part. But rather the coding part.

A chapter of your choice

In this chapter we want you to use a package that was not mentioned in the main part of the course (data manipulation, visualisation using ggplot2, regular expressions and reporting using Rmarkdown does not count) and perform a task that was not directly discussed in the course. Be creative!

Note: Here we don't want you to use a new statistical/machine learning method. We rather want you to use a method to prepare or display data. It could even be a package that enables you to create prettier documents. What we don't want to see is you fitting e.g. a regression model to your data. Please insert a separate section in your report and call it "Chapter of choice" such that we can easily recognise it!

If you need input: feel free to ask us. **Please indicate or highlight in your report what is your chapter of choice.**

Dynamic documents and reproducibility

We want you to create the pdf/html document with Rmarkdown. Make sure that your analysis is fully reproducible and comprehensible for anyone reading it. This will be shown during the course.

Comments

The analysis and your R code needs to be commented. Keep in mind that you should make up a story to tell to a client. Putting 5 uncommented plots of the model fits is not something a client would like to see. Please comment on the results, what you are doing, and why you are doing things.

Note, however, that you should not add a very long text section just for the purpose of having some text. Always keep in mind that a potential client will read your work, they want to understand what you did and why and what the results of the analysis are. However, they do not want to get bored.

Finally, note also that very often Rmarkdown documents become very long. Two hints to shorten them is to use the chunk options *message* and *results* such that only really needed message and results are shown. For example, in the chunk where you load packages, you may want to set *message* to *FALSE*. Additionally, the R code can be collapsed in an html or you can hide R code using the chunk option *echo* in a pdf output.

It is better to conceal your R code and lengthy structure outputs in order to avoid generating an excessively long document. If you need or want to highlight any specific aspect in your R code, make the reader aware by referring to it and feel free to show those lines of code like joining two data sets.

Sell the story

Your analysis must be a story that you would like to sell to a client and in this sense, it should be complete, easy to read, and have a thread (what are you doing, why, conclusions, interpretations, overview . . .). It is completely up to you how the story goes and flows. Nevertheless, keep in mind that you need to make clear:

- Where the data comes from.
- What the aims / hypotheses of the analysis are.
- What the interpretation of every step is.
- What the conclusions of your analysis are.
- What potential limitations are.

Put yourself into the position of a client and check if your analysis is detailed, comprehensible, and easy to follow.

Generative AI

"ChatGPT" and co. are useful tools that can provide great support during your work, but it is important to avoid abusing it.

Write a quarter to maximum a half a page about your learning in using gerative AI for solving tasks in R. What have you used it for? For what kind of tasks turned they out to be helpful? How do you check correctness of the answer? What did not work? Those are example questions which you can ask yourself and reflect, but you are free to write about your experiences.

Data set

Where to find data:

- Anywhere!

- Note that kaggle is one of many sources of data sets (too often used by students)
- You may want to consider websites like opendata.swiss, https://opendata.swisscom.com, https://datasetsearch.research.google.com/, http://puenktlichkeit.ch/, www.mldata.io, https://www.pxweb.bfs.admin.ch/, https://www.ostluft.ch/, http://www.agrometeo.ch/de/meteorology/datas, https://data.stadt-zuerich.ch/, https://data.world/uci/, https://databank.worldbank.org/source/world-development-indicators, https://data.gov.sg/dataset, (just to mention a few examples…)
- R itself comes with a few hundreds data sets (type "data()" to see the list), the vast majority of add-on packages also comes with data sets
- Note also that if you found a cool data set that seems to be too small/simple… you may complement it with another data set (e.g. add coordinates, meteo or similar things).

Additional formal requirements

Here a few additional formal points about the assignment:

- Students must work in pairs.
- Deadline to hand in the assignment is on Friday 3 weeks after the end of the course at 5 pm (hard deadline! No later submission will be accepted). Plan enough time to finalise and compile the document.
- The assignment must be uploaded on Ilias on the delivery folder.
- Please upload only once your group work, i.e., one member of your group can upload your work.
- You must hand a zip file named after your group number and your family names (e.g. 1_Trisconi_Renaux.zip).
- The zip file **must** contain:
  - a readme.txt file;
  - structured folders (e.g. "Data", "Scripts", …);
  - the data set analysed;
  - the Rmarkdown file (.Rmd);
  - the pdf or html output file.
- The length of the pdf/html document:
  - should not exceed 25 pages;
  - anything longer than 30 pages will not be considered;

Grading

Your work will be graded and the evaluation criteria are:

- Story line and quality of the end product (e.g., commented plots and summaries, conclusions, interpretations, easy to read).
- Formal requirements (e.g., large enough data sets, date or geographic variable, join two data sets, code style, …).
- Structure of your report and required chapters (read all the detailed instructions above, e.g., it should contain a graphical analysis, a statistical/ML method, a chapter of choice, …).
- Quality and diversity of the plots.
- Sophistication and innovation.