R-Bootcamp: Assignment

Dr. Matteo Tanadini

10-13 February 2020 @ HSLU Luzern

Admin

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

Below a few more information about the assignment:

- students must work in pairs
- deadline to hand in the assignment is Friday 6th of March at 5 pm (hard deadline!)
- you must hand a zip named after your family names (e.g. Tanadini_Mueller.zip)
- the zip file must contain:
  - the data
  - the Rmd file
  - the pdf or html file
- the zip file should contain:
  - structured folders (e.g.data, code,...)
  - a readMe.txt file
- the assignment must be uploaded on Ilias on a delivery folder (detailed information follows)
- the length of the pdf file should not exceed 20 pages (anything longer than 25 pages will not be considered)
- the end product will be graded as passed or not passed
- evaluation criteria are:
  - completeness (all topics addressed in class there?)
  - sofistication (adapt to your level)
  - quality of the end product (e.g. commented plots and summaries)

List of functions

Make a list that contains all the functions we have seen during the course. Add also those functions that were not discussed in the course, but that you discovered while e.g. discussing with the course instructors. NB: that the list of function does not count towards the page limit. Note also that we would like you to export your list as a pdf and append it to the other pdf.

Find a use case that comes with some data

Roughly speaking the assignment represents a data analysis. Therefore, the first task is to find an interesting use case that comes with data. The choice of the case is open to you. Nevertheless, makes sure the following criteria are fulfilled.

- the use case comes with some data
- the dataset must contain at least a few hundred observations and a dozen variables
- among the variables there must be numeric and categorical ones
- the dataset should also contain dates or geographic locations (both is even better)

- if the dataset itself does not contain dates or geographic locations, find another dataset with these information (see example below)
- the data can be publicly available or come from e.g. your employer. If not publicly available make sure that you can use the data and discuss the results with the course instructors.

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 year (say weekly sales provided by Friscolino AG), data about climate (i.e. day temperatures and rain e.g. from MeteoSuisse) and a dataset about city population (e.g. from the Swiss Federal Office of Statistics homepage). We will then merge all these three dataset into a single dataset.

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 get the stations coordinates (?) and we get the local vacations days (cantons homepage).

Ideally, your datasets should come into different formats and from different sources (e.g. xlsx, zip, ... and from websites or locally downloaded files).

Prepare the data for the analysis

As mentioned above, datasets need to be merged and cleaned before starting the analysis.

Visualise the data appropriately

The first step to carry out once the data is ready to be used is to inspect the data (with summary statistics, for missing values, but also visually).

For simplicity we may want to assume we are aiming at predicting ice cream sales or whether the train are late or not (nothing too fancy in terms of modelling). In other words, we want you to invest little time in modelling, which is the focus of other courses.

To fully grasp the power of add-on graphical packages we want you to produce at least one graph with...

Fit a starting model (lm)

Find a good starting model and apply all functions and action you find useful. For example if you were to fit a linear model, you may want to inspect the results, check the residuals diagnostics and make some graphs with predictions.

Fit further (more complex) models

In this chapter we want you to fit further models that you will than compare to the starting model. Use 10-fold cross validation to compare models. As a "good of fitness measure"" you may want to use the *Mean Squared Error* (for regression) or the *AUC* (for binary classification).

A chapter on your choice

In this chapter we want you to use a package that was not mentioned in the course and perform a task that was not directly discussed in the course. Be creative!

If you need input, you may ask us.

Dynamic documents and reproducibility

We want you to create the pdf with Rmarkdown (or Knitr). Make sure that your analysis is fully reproducible and comprehensible for anyone.