R-Bootcamp: Series 3

Dr. Matteo Tanadini

February 2022

1 Exercise: Fitting models

1.1 Question:

The `cars` *built-in dataset has two columns: "speed" (of the car) and "distance" (needed to stop the travelling car). Perform the following tasks.*

1. use the `str()` function to check that both columns are indeed numeric variables and to see how many observations are present in the data
2. create a new dataframe named `cars.short` that contains only the first 36 observations present in the original dataset.
3. add a new column that contains the values 1 to 6 repeated 6 times (for a total length of 36). Name this column "test_day"
4. display the data (you can use "base" graphs or the add-on packages {*lattice*} / {*ggplot*2}) (i.e. create a scatter plot for "dist" against "speed" and a boxplot for "dist" against "test_day"
5. fit a linear model where "dist" is the response variable, "speed" and "test_day" the predictors
6. look at the summary of the model, how good is the fit? (hint: look at the adjusted-r-squared value)
7. fit a model that only contains "speed" as predictor. Then compare the two models. What is the p-value of the ANOVA test?
8. plot the model diagnostics in one single device (hint: you may want to set the parameter "mfrow" via the `par()` function).
9. the very first plot produced by the command you just ran displays the residuals against the fitted values. Reproduce a equivalent graph with the `ggplot()` function

Going further (*)

10. get the regression coefficient from the fitted model. Use the corresponding extractor function first. Then use `str()` or `names()` to find out in which slot of the lm object the coefficients are stored and extract them as you would do when access a slot in a list.
11. get the adjusted-r-squared and the regression coefficients and store them in a list. Pay attention to the choice of the list name (remember must comprehensible and consistent)
12. use the `gam()` function in {*mgcv*} package to fit a Generalise Additive Model on the same data. Look at the summary of the fitted model and compare the adjusted-r-squared with the one of the linear model.

1.2 Question:

The `airquality` *built-in dataset has several columns. Fit a linear model where you try to predict "Ozone" concentration from "Solar.R" and "Wind". Comment on the data and results.*

2 Exercise: Missing values

2.1 Question:

*Create a dataset in a spreadsheet (e.g. excel) that contains empty cells. Make sure your dataset has both numeric and categorical variables that contain empty cells. Read the data in **R** and comment on how where the empty cells coded.*

3 Exercise: Packages

3.1 Question:

1. how many packages are available on CRAN?
2. search on CRAN for the "MASS" package
   - is this a recommended package?
   - how many authors are there?
   - who is the maintainer? Is that a good reference?
   - is it true that we can find this package in the "econometric" task view on CRAN?
   - is there any companion book to this package?
3. go to the "Machine Learning" task view on CRAN and find what is the reference implementation of the "Random Forest" algorithm
4. go to CRAN and click on the "Contributed" section. There search for a tutorial on how to create **R** packages. Name a document that seems to be good

3.2 Question:

In which package can you find the following objects:

1. the `ls()` function
2. the `plot()` function
3. a function that contains the word "sunflower"
4. a dataset named "swiss"
5. the `apropos()` function

3.3 Question:

*Assume you realised that loading a package, say package "nlme", creates a conflict. Assumed that you don't really need this package, how can you unload it from your **R**-session?*

3.4 Question:

Assume you wrote a few functions yourself and want to write a package to store them. It that feasible? Could you then pass this package to someone else or are you forced to put it on CRAN?