R Programming 2018 - Take home exam A

Rossi Abi-Rafeh

You will solve two take-home exams for the R programming course. The tasks you have to solve in homework A are described in this document. Your submissions for homework A will count for 25% of your final grade. You have to submit your answers before Wednesday 24 october at 8 AM in the morning. We will correct it in class on the 24th of October. In this document, you have all the information about homework A.

There is a 2nd take-home exam for this class. It will count for the rest 25% of your final grade. You will receive information about it on wednesday 7 november.

Rules for homework A:

The homework is in groups :

- You work with your partners and submit the same answers.
- Submit your answers on the Moodle homework link.
- Each member of the group has to submit the answers separately through his/her own Moodle account to receive the grade. Example : Paul and Laura are a team. They make the same pdf file with their answers. Both Paul and Laura submit the document each from their own Moodle account.

Submissions are made of 3 files :

- one pdf file : 2 pages of text (not including the figures/plots) answering the questions, and explaining briefly what you're doing. You can produce the pdf using Latex/Word/knitr/Rmarkdown/etc, as long as it's 2 pages, and readable.
- one .R script : It has to run without an error message.
- one .csv file with your predictions for Task 1A - Step 11. It should look like sample_submission.csv, but with your own predicted values.

Grading policy :

Each step in this document gets you 1 point if done correctly. For questions requiring you to code, out of this 1 point, a well-documented and commented script gets you 0.25 points.

In the beginning of your R script, do not forget to install and load all the packages you will use later in your script.

If you import data in R, please (1) set your working directory in the beginning of the script and make that command clear so that we can change it on our computers when we grade, (2) put the data files directly in the working directory, (3) use the relative path when you import the data file (no manual Choose File.)

If your .R script does not run on our computers, or shows error messages, your grade is automatically 0 on all steps after the FIRST error message. For example, if you forget to load the tidyverse in the beginning, and you use the function "summarize" on the first line of your script, it will have an error message since line 1. You will then get a 0 on your assignment. There will be no exceptions to this rule.

If your code runs, and your output does not match what you have in the pdf for a given step, your grade for that specific step will be 0.

Task 1A - Predicting house prices in Ames, Iowa

In this task, you will forecast the sale price of residential property in Ames, a city in Iowa, using a simple linear model. To do so, you have the prices of the last sale of residential property in Ames, Iowa from 2006 to 2010, and a large number of variables describing very precisely every property. The data you will use for estimation is on Moodle (train.csv).

Information about the data : https://www.kaggle.com/c/house-prices-advanced-regression-techniques/data

Step 1 : Import the data in R in a data.frame (or similar) format

Step 2 : What is the number of observations? What is the dependant variable? How many explanatory variables can you include?

Step 3 : Is your dependent variable continuous or categorical - conceptually? What is its class in R? Is this a regression or a classification problem?

Step 4 : Name two continuous explanatory variables in your data? Explain what they are. Are they stored as numeric in R?

Step 5: Summarize the numeric variables in your data set in a nice table.

Step 6 : Plot the histogram of the numeric variables that you deem to be interesting. (At least 3.) Show me the plots in your pdf.

Step 7 : Are there any missing data? Compute the number of missing observations per variable. How do you solve this problem?

Step 8 : How many duplicate observations are there in the dataset? Remove any duplicates.

Step 9 : Convert all character variables to factors

Step 10 : Estimate a linear model including all the variables. Eliminate iteratively the least important variables to get to the most parsimonious yet predictive model. Explain your procedure and interpret the results. **NOTE 1 :** You should have an R2 of at least 70%. **NOTE 2 :** Do not use interaction terms. You can use powers and transformations (square, logs, etc. . .) of a feature/explanatory variable, but no interactions.

Step 11 : Use the model that you chose in step 10 to make predictions for the test set (found in test.csv). Export your predictions in a .csv file (like the example in sample_submissions.csv) and submit it.

Step 12 : How can you judge the quality of your model and predictions? What simple things can you do to improve it?

Task 2 A - Consistency of the OLS estimator

In this task, you will check the consistency of an OLS estimator. You will create data from a simulation. Since you create the data, you will know what is the true model (T) (i.e. you know the true coefficients), so you will be able to check how linear regression performs when the number of observations increase. In real-life situations, you do not know (T) the true model, so you cannot do this comparison directly, but you can argue for the assumptions under which consistency holds and check some of them with tests.

The true model you will use is :

(T) $y = \beta_0 + \beta_1 x + \epsilon$;

x is uniformly distributed between 0 and 1.

ϵ the noise, is normally distributed mean 0, and standard deviation 1.

$\beta_0 = 0$ and $\beta_1 = 2$. β_1 is the parameter of interest.

Set your seed to 1234 for this homework.

Step 1 : Simulate 100 independant draws of (x, y) following (T). Put them in a table with columns x and y. *Hint : you need also to simulate 100 points of* ϵ.

Step 2 : Make a scatterplot of the simulated data : y on the vertical axis and x on the horizontal axis.

Step 3 : Estimate a linear regression model of y on x using the simulated data. Call it model1 in R.

Step 4 : Draw, in red, the line of the predictions from model1 on the scatterplot of data. (This is the same as the regression line

Step 5 : Let $(\hat{\beta}_0, \hat{\beta}_1)$ be the estimated coefficients in model1. Are they exactly equal to the true coefficients (β_0, β_1)?

Step 6 : Compute the difference $d = |\hat{\beta}_1 - \beta_1|$. Is this difference the bias? Why/Why not?

Step 7 : Simulate again independant draws of (x, y) (like in step 1) for 1000, 5000, 10000 and 1e6 observations. Estimate a linear regression model of y on x using the simulated data each time (like in step 3). Compute $d = |\hat{\beta}_1 - \beta_1|$ (like in step 6) each time and store the values in R.

Step 8 : Draw a barplot of d for different numbers of observations (you computed these values of d in step 7). How is d changing with the number of observations? Which property of the OLS estimator did you just check with this graph? How can you change (T) so that this result does not hold anymore?