R Programming 2017 - Challenge A

Rossi Abi-Rafeh

10/24/2017

You will solve, in your group, two challenges for the R programming course. The tasks you have to solve in Challenge A are described in this document. Your submissions for Challenge A will count for 25% of your final grade. You have to submit your answers before Friday 3rd of November at 11 AM in the morning. In this document, you have all the information about Challenge A.

There is a 2nd take-home challenge for this class: Challenge B. It will count for the rest 25% of your final grade. You have to submit answers for Challenge B before Friday 8th of December at 11 AM in the morning. All the information about Challenge B are in a separate document on Moodle.

Rules for Challenge A:

The Challenge is in groups :

- You work with your partners and submit the same answers.
- Submit your answers on the Moodle challenge 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 load external data, please make the command visible so that we can change the path on our computers when we grade.

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 "tbl_df" on the first line of your script, it will have an error message since line 1. You will then get a 0 on your assigmment. There will be no exceptions to this rule, nor negociations.

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 predict 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 features describing very precisely every property. The training data is on Moodle (train.csv).

Information about the data : http://www.amstat.org/publications/jse/v19n3/decock/DataDocumentation.txt

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 target/outcome/dependant variable? How many features can you include?

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

Step 4 : What are two numeric features in your data?

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? How many observations have missing data? 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 : Fit 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 - Overfitting in Machine Learning

ML algorithms are flexible methods that you can use to understand the relationship between an outcome y and features/explanatory variables x, and to predict y for some new observation of x. Generally, we conceptualize the problem as $y = f(x) + \epsilon$. Only y and x are observed in practice. We do not observe f, or ϵ (it's the noise). The goal of the econometrics/ML algorithms is to find (estimate/train) a $\hat{f}$ that fits the data well, and allows us to predict y_i for a new individual x_i.

The flexibility of a ML algorithm is chosen by the analyst. When you do a linear regression in econometrics, you're doing a low-flexibility algorithm : you assume the shape of f is linear AND that it is the same on all the domain of x.

If in reality, your f is actually equal to the square function $f(x) = x^2$ than a linear regression won't give you good results, because the assumption you make on f is too restrictive and false.

If in reality, your f is equal to the absolute value function $f(x) = |x|$ then again using a linear regression to estimate f won't give good results : you will have a horizontal line as a result, but the absolute value

function is not a horizontal line! Although the absolute value function is linear on $(-\inf, 0]$ and then again $[0, +\inf)$, it does not have the same slope parameter on these two segments (slope equal to -1, then equal to 1). A linear regression is not flexible enough to allow estimating two different slopes on two different regions.

If simple linear regression is not flexible enough, other ML algorithms are too flexible. This is known as overfitting, and is a common and important problem in applied econometrics and machine learning.

In this task, you will replicate a simulation and the graphs below, and by doing so, you will have more insights about why linear regression may not be the best method to use for prediction. In this simulation, you will create data for which you know the true model (because you created the data), and you will be able to check how linear regression performs compared to the true model. In real-life situations, you do not know f, so you cannot do this comparison.

The true model you will use is :

(T) $y = x^3 + \epsilon$;

x is normally distributed mean 0, and standard deviation 1;

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

Set your seed to 1 for this challenge.

Step 1 : If the true model is (T), what is f in this case?

Step 2 : You have a new individual of interest, Paul. You know that $x_{Paul} = 2$. If you know f in practice, and $E[\epsilon|x] = 0$ but not the exact value of ϵ_{Paul}, what is the best prediction for y_{Paul} that you can make? Note : I call this best prediction $\hat{y}^T$. Its value for Paul is $\hat{y}^T_{Paul}$.

Step 3 : Simulate 150 independant draws of (x, y) following (T). Put them in a table with columns x and y. *Hint : you need also to simulate 150 points of ϵ.*

Step 4 : Make a scatterplot of the simulated data. See Figures.

Step 5 : On the same plot, draw the line of $\hat{y}^T$.

Step 6 : Split your sample into two. A training set and a test set. Plot the same scatterplot as in Step 1, differenciating in colour between the points you will use for training and the points you're keeping aside for the test.

Step 7 : Train (fit/estimate) a linear regression model on your training set. Call it lm.fit in R.

Step 8 : Draw, in red, the line of the predictions from lm.fit on the scatterplot of the training data. (This is the same as the regression line) Compare it to $\hat{y}^T$. Is a linear regression a good method here? Why or why not?

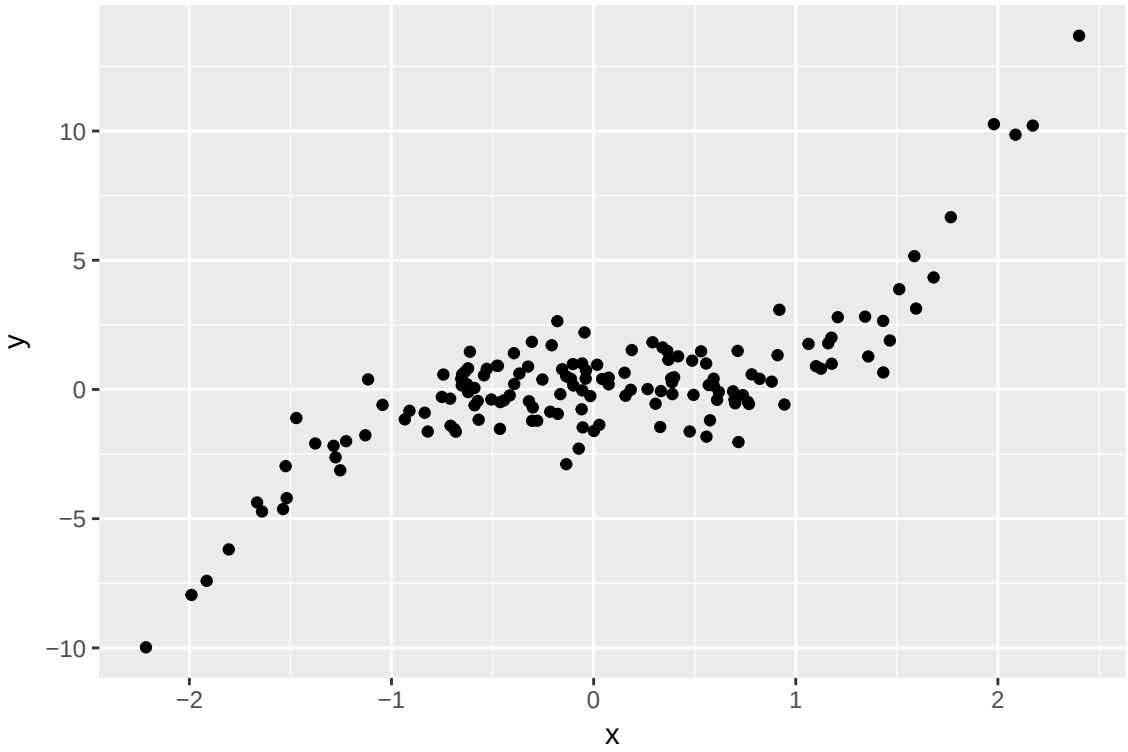

Figure 1: Step 4 - Scatterplot x - y

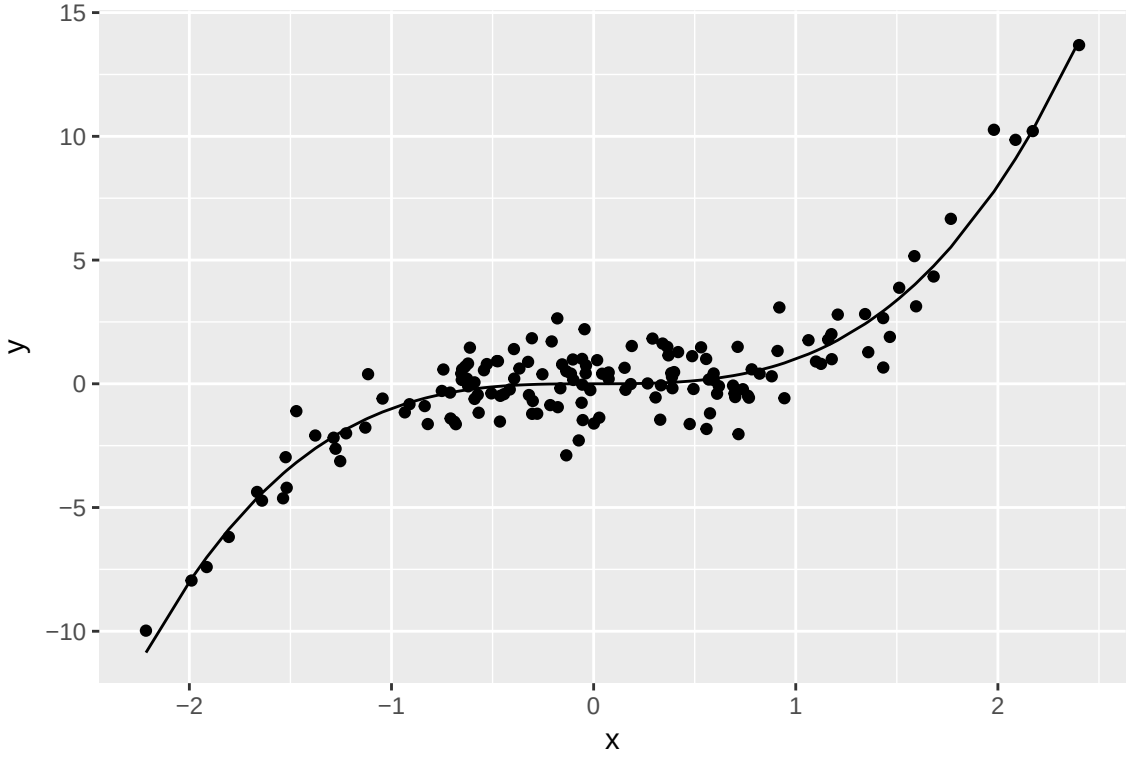

Figure 2: Step 5 - True regression line

Figure 3: Step 6 - Training and test data

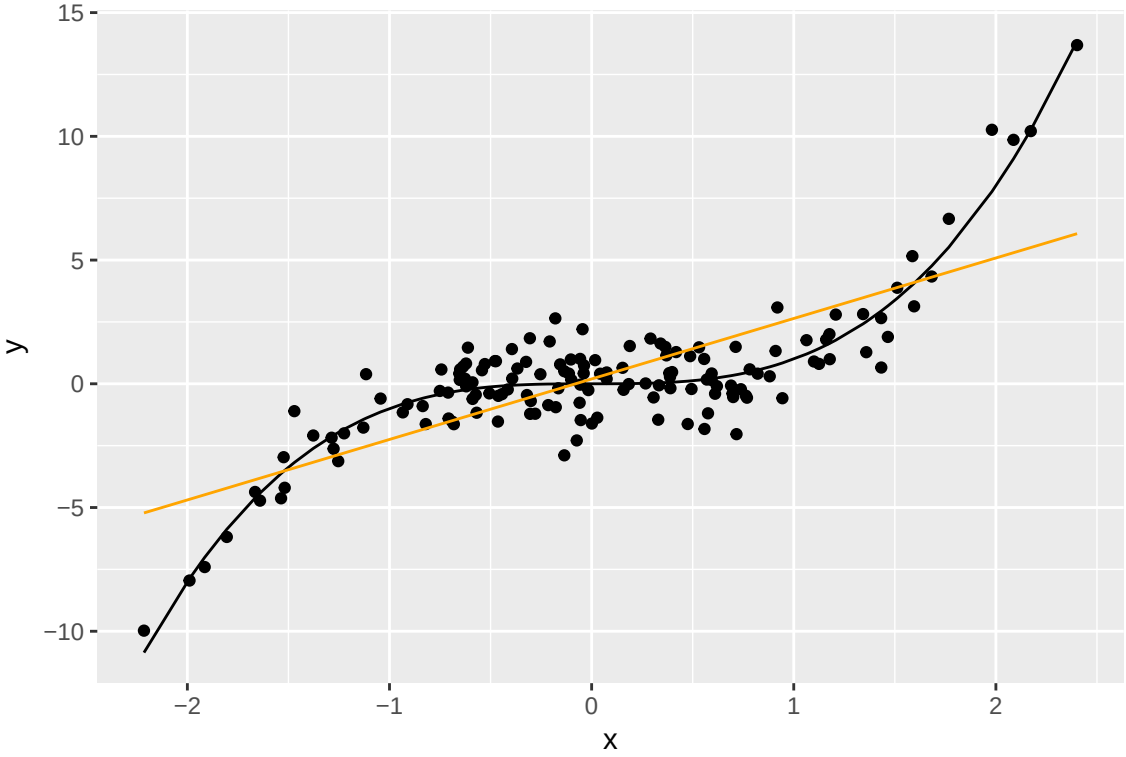

Figure 4: Step 8 - Linear regression line