Challenge B - Liang Yan & Hu Tianxing

Liang Yan & Hu Tianxing

2017/12/7

## Task1 B

Step1: A non-parametric statistical method makes no assumption on the population distribution or sample size. In the random forest approach, a large number of decision trees are created. Every observation is fed into every decision tree. The most common outcome for each observation is used as the final output. A new observation is fed into all the trees and taking a majority vote for each classification model.

Step2: We loaded all the packages needed in the the beginning

Then we input the training data and the test data.

From Challenge A, we found that while NA in some variables indeed means “missing value”, while some NA in other variables has practical meanings. we removed the variable Id and variables with NAs meaning missing values. We also removed all the NAs from numeric variables and two character varialbes “Electrical” and “MasVnrType”, where NA has no practical meanings.

train1 is not the training data. It is just used to create the formula in a convenient way by removing the dependent variable “SalePrice”.

step2 We used the random forest as the nonparametric method.

Step 3: We used the model trained to make predictions on the test data. lm\_tra is the linear model we estimated in challenge A after comparing the exclusion of missing values and the comparison of the importance of variables. In order to use the random forest model to predict testing data, we have to ensure that the all the variables in the testing data have same levels as those in the training data and we created a loop to do this. We made the predictions. By summarising parameters and comparing the residual standard error, which means the variations of the difference between predictions and observations, we concluded that the random forest would give better predictions.

## Task2 B

First we created 150 independent simulations of x, e and y respectively based on the true model y=x^3 + e. The simulations are created by random draws from the normal distribution. Then we spilted them into training data and testing data.

Step 1: We trained a low flexibility model of bandwidth = 0.5 based on the training data.

Step 2: We trained a high flexibility model of bandwidth = 0.01 based on the training data.

Step 3: We made predictions on the training data using the low and high flexibility model respectively, then we gathered the predictions and the simulations, and plotted them on the same graph.

Step 4: The predictions of **ll.fit.highflex** are more variable. The predictions of **ll.fit.lowflex** have the least bias.

Step 5: We performed similar procedures, predicting with the two models, gathering the data and ploting them on the same graphs. The predictions of **ll.fit.highflex** are more variable. The predictions of **ll.fit.lowflex** have the least bias.

Step 6: We created a vector of bandwidth going from 0.01 to 0.5 with a step of 0.001 using the seq function.

Step 7: We estimated a local linear model y ~ x on the training data with each bandwidth. We created an empty list first. Then we used the for loop to create models with different bandwidth values and store them in the list we created.

Step 8 & Step 9: We computed for each bandwidth the MSE on the training data and the test data by using for loop and for both generating the predictions and creating the MSE.

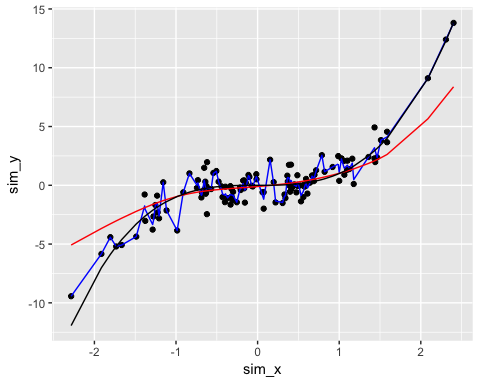
Step 10: We plotted the MSE with the increase of bandwidths from 0.01 to 0.5 of the training data and the testing data respectively.

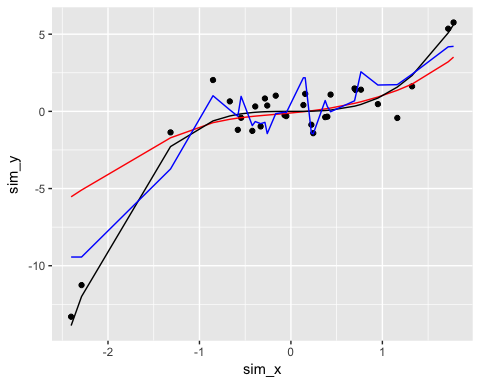
## Task3 B

Step 1: We imported the CNIL dataset from the Open Data Portal and separate the variables with the sep argument.

Step 2: We first converted the variable Code\_Postal to character in order to use the substr function.We used the substr function to represent each observation with the first two digits of their postcode. Then we found the number of “types” of the first two digits, which is equal to the number of organizations with a CIL per department, since all the firms in the list have CIL. Lastly we put the results in a table.

## Graphs

Graph of Task2B Step 3: The scatterplot of x-y, along with the predictions of ll.fit.lowflex and ll.fit.highflex, on the training data 

Graph of Task2B Step 5: The scatterplot of x-y, along with the predictions of ll.fit.lowflex and ll.fit.highflex, on the testing data 

Graph of Task3B Step 10: the MSE-bandwidth relations on training data, and test data 