What I want to see

I) EDA:

optional:

- how you dealt with missing values

- how else you cleaned your data and how you engineered features

why optional? If you chose a clean dataset maybe you didnt have to do much

This was up to you

obligatory:

- renaming of columns if they are not intuitive

- an example of how to use t-test (it says in the brief – I am sorry, I also don’t think   
this is necessary and is problem specific – just do it so I see that you can use the t.test() function)

- correlationplots (optional: chi-squared matrixes)

- variable selection (preferably different way to select variables and then compare   
same models with different variables) e.g. backwards, forwards, based on p-value, based   
your brain/interactions (whether it makes sense or not → ptratio\*room made sense although  
corrplot showed high correlation, but garageCars and garageSize didnt!)

II)   
- Linear Regression OR Logistic Regression OR Multivariate Logistic Regression

(depending on your data)   
- selecting best model for linear regression/logistic regression (highest accuracy amongst those  
you tested with the different subsets of variables chosen in I) ­)  
- validating using cross validation (if you want also bootstrapping - caret::train() without  
train control uses bootstrap by default\*\*\*)

\*\*\*the reason I am saying this is because in the last session the teacher advertised doing this  
to “boost your grade”. He also taught how to do boostrap and crossvalidation manually…  
We don’t waste our time with that and just change the parameter in the train() function ;)

III)

- at least 2 tree based methods

pruned tree (train() selects the tree with the best length. You can play around with different

lengths by changing tuneLength parameter and test on test set – is the tree size that train()  
returns really the best one according to the accuracy of the test set? )

bagged tree – method=treebag

boosted tree – method=gbm

random forest – method=rf

Obviously for II and III I want some stats and plots of your model

If you are doing classification: try out KNN and NB too if you want