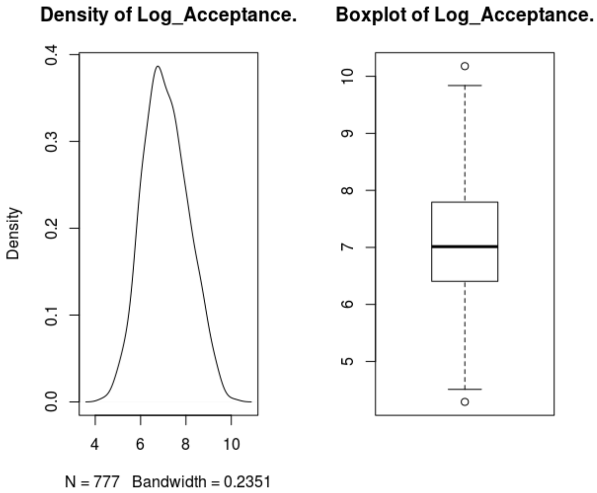
P. Sai Kiran
STA - Homework2

Q1)

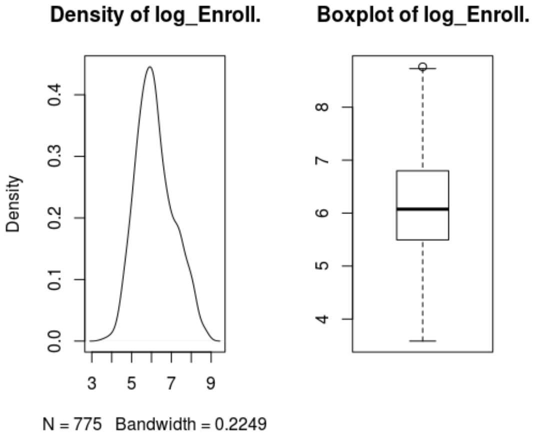
- Checking the sanity of the data by doing, sum(is.na()) to the data give that there are no 'na' vaules in the data.
- For further analysis down the line, converting the rowname, college name into a feature.
- There are some features like which are numeric but should have been categorical.
  So converting the features, private and college_name into factors.
- We can create some new features such as, acceptance_rate, not_enrolled and total_fee by apps/accept, apps-enroll, outstate+room.board+books+personal.
- Doingsummary of the data, it can be seen that some of the features are in thousand scale. Hence applying log transformation on them.
  The features are, Accept, Enroll, F.Undergrad, P.Undergrad, Outstate, Room.Board, Books, Personal, Expend, not_enrolled, total_fee.
- Outlier removal : Looking at the distributions and the boxplots of the features.

Feature: Log_Acceptance



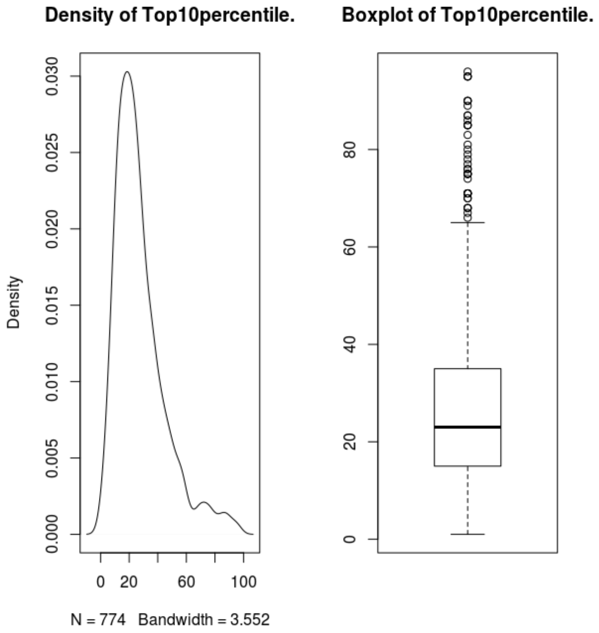
Inference: Normal Distribution with a couple of outliers.

Feature: log_enroll



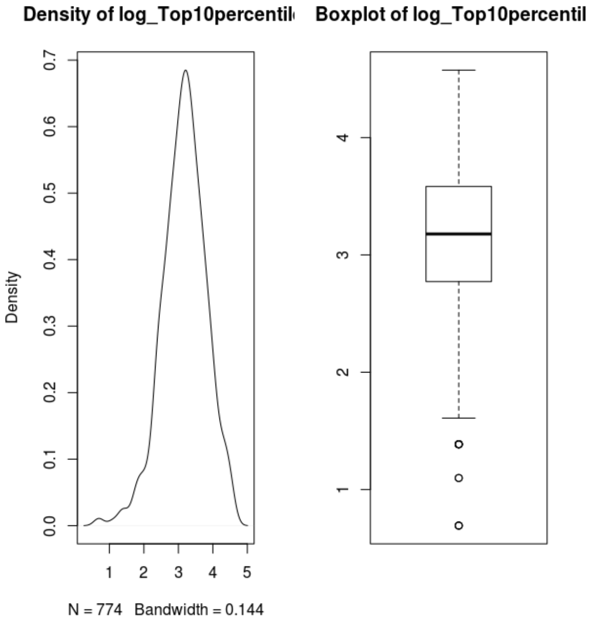
Inference: Normal distribution with a single outlier.

Feature: top10perc



Inference: The outliers that are showed in the boxplot are too many to remove. Hence changing the scale of the feature to log.
Plotting the boxplot then, gives us the following.

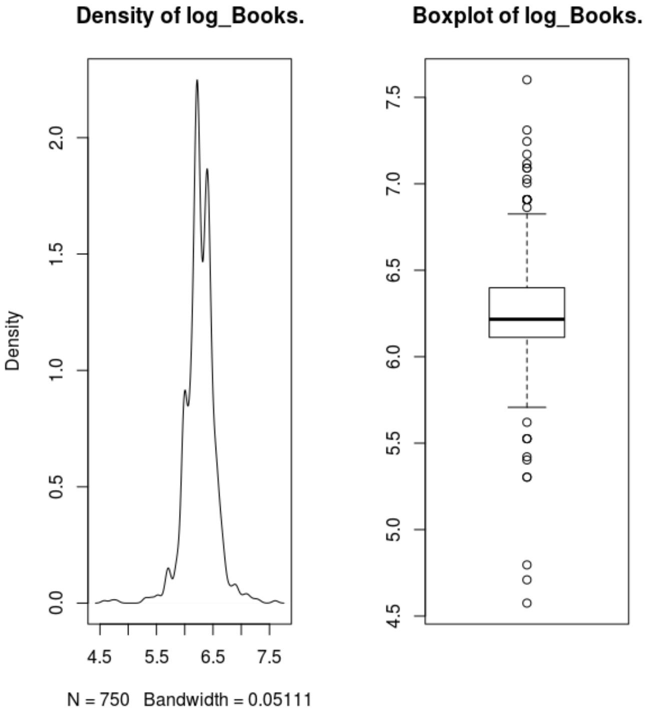
Feature: Log_top10perc



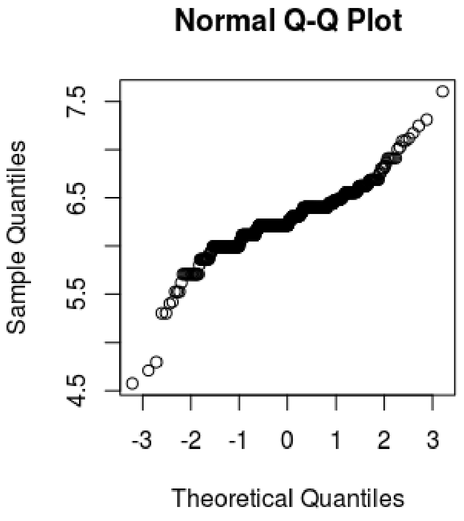
Inference: Normal distribution with left tails. There are three outliers that are to be removed.

All the outliers in the features are removed with a similar approach.
There were some features where the outlier number is big despite doing a log transformation, like in log_books. The following approach is used to remove the outliers from such features.

Feature: Log_Books



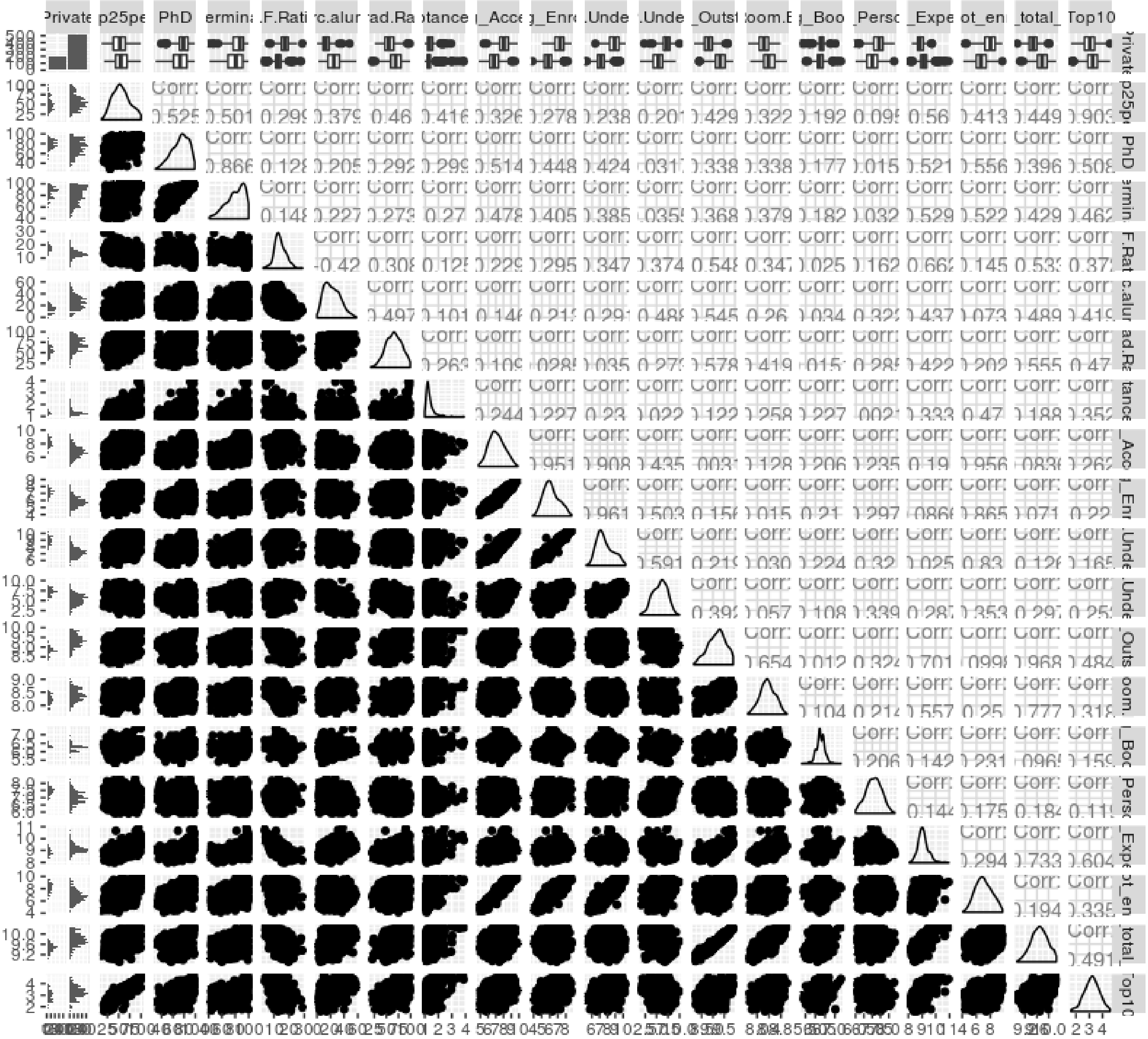
Inference: The outlier number is big and we cannot lose to many data points. Hence looking at the qqnormplots for further analysis, we get the following.



Inference: Looking at the qqnormplot it is clear that we have one outlier at the top, and three at the bottom, that seem out of place. Hence those points are removed.

Similar approach is used where ever the features exhibited similar issues.

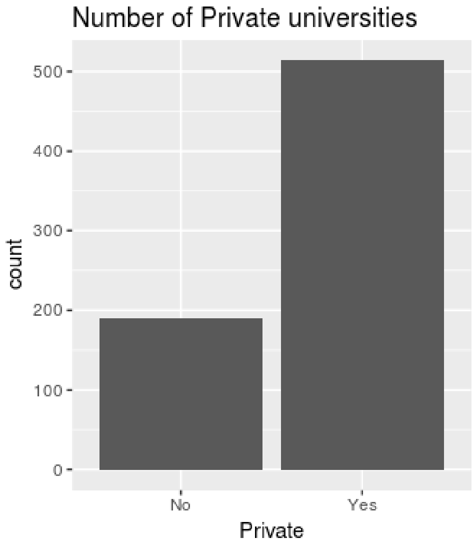
To get an overview of the patterns in the data, doing a pairsplot of the dataset gives the following plot.



Inference:
Looking at the general trend of the college_data, it looks like

The following seems to have interesting relationship.
(log_Enroll, log_accept) indicating that students are applying for the universities that only interest them.
(log_total_fee, log_outstate) indicating that most of the expenditure constitues tuition fee

- Looking at the distribution of the number of private universities in the dataset after the outlier removal is as follows.

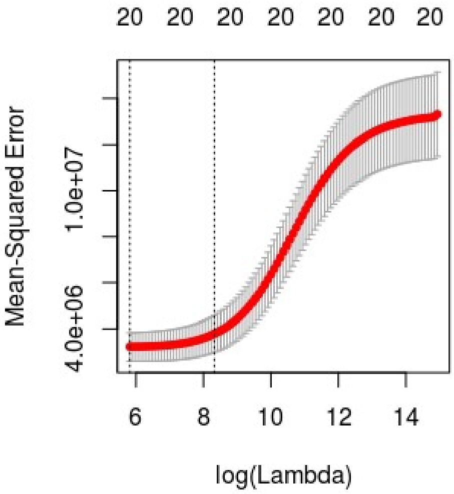


Inference: The number of private universities are more.

- Creating a train and a test set by a 70%, 30% split for train and test set.
- Fitting a linear model and taking a summary of the linear model. The model is a 3-star model with the following features as the important features in the summary. Acceptance_rate, log_Accept, log_F.Undergrad, log_P.Undergrad, log_Outstate, log_Room.Board, log_Personal, log_Expend, log_total_fee.
- The multiple-Rsquared value around 0.8 indicating that the target is explained 80% by the features in the dataset.

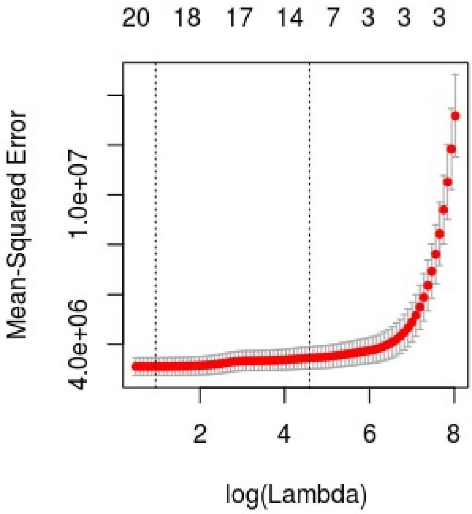
- Following are graphs of the models that were run.

Plot: Ridge model cross validation plot.



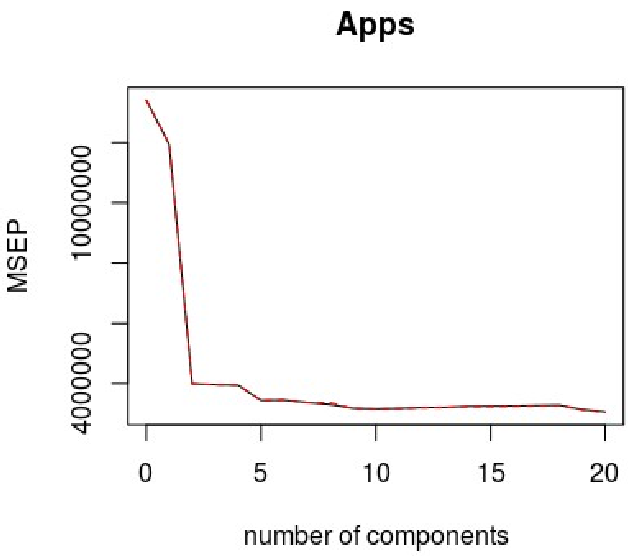
Inference: With the increase in lambda the shrinkage is more, indicating that those models were under-fitted.

Plot: Lasso model cross validation.



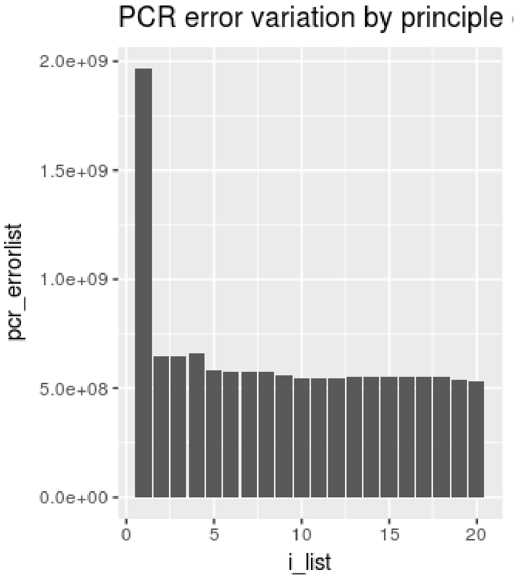
Inference: With increase in the lambda, the error of the model is increased indicating that the shrinkage is more leading the models to have gotten under-fit.

Plot: PCR validation plot.



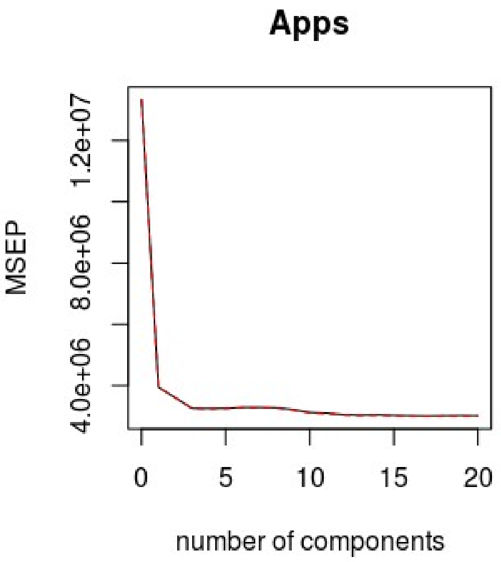
Inference: The total number of principle components to be considered can be chosen as 5. Since after first 5 principle components the error variation is not more.

Plot: PCR error variation on the test data by number of principle components.



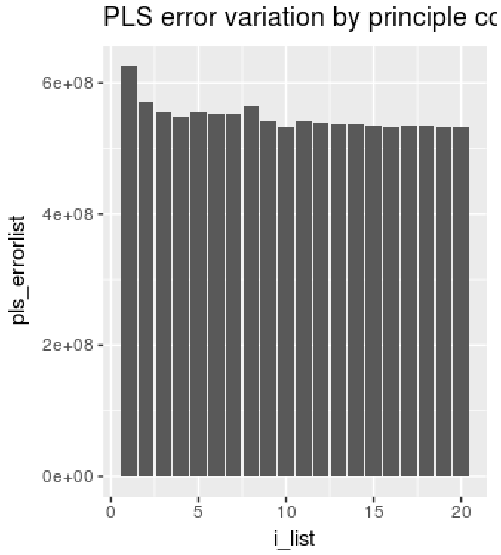
Inference: The number of principle components that can be considered to be taken is 2 according the error of the testing set. Since the error varition after first 2 principle components is very less.

Plot: PLS regression variation for the validation dataset.



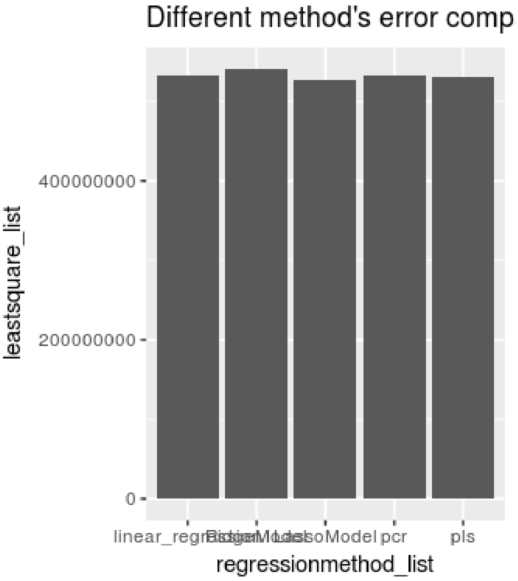
Inference: From the validation dataset, it seems that the optimal number of principle components are 4 since after first 4 principle components, the error variation seems to be very less.

Plot: PLS error variation on the test set by number of principle components.



Inference: From the test set errors, the number of principle components that should be considered for this dataset seems to be 3 since after that there is no large variation in the error.

Plot: Error comparison for different regression methods.



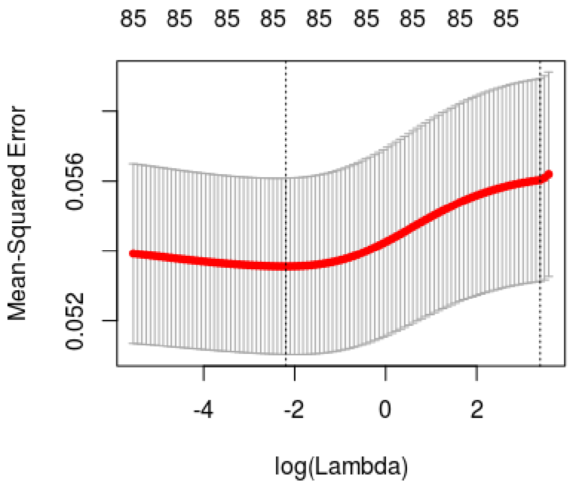
Inference: The error obtained from all the methods seem to be the same. Out of all, the error obtained from Lasso seems to be the best.

g) The predictive power of the model and it’s accuracy is subjective, since for example an error of 0.5 could be very good for one dataset but bad for another. Hence using LeastSquares as the loss function we cannot say if these models are good at predicting the applications. But out of all Lasso seems to be doing the best job like it can be infered from above graph.

Q2)

- Checking the sanity of the data by doing, sum(is.na()) we get that there are no ‘na’ values in the dataset.
- The features in the data seem to be some scoring given to each customer. But, there seem to be some features which are probably categorical features. Since we do not know what each feature means, we cannot be sure of that. Hence leaving the feature type as it is.
- It is a tedious process to remove the ouliers by a univariate approach. Hence a method has been written which removes the outliers. Running the method is taking off around 3000 data points and we are left with 1500 or so data points which is not optimal for running algorithms.
  Hence outlier removal for this problem is omitted.
- Running a summary for the regression object, we get that it is a 3-star model!
  Following are the features that are important.
  V4, V47, V55, V57, V58, V59, V76, V78, V82.
- A custom method has been written to decide which class a data point belongs to. Since when we run predict method on the data point, the range of the values is from ‘negative’ to >1 which doesn’t makes sense for the problem.
  The method calculates the distance of the predicted data point from either of the targets and assigns it to the class whose distance is minimum.
- Running the Ridge model we get the following graph as the crossvalidation variation.

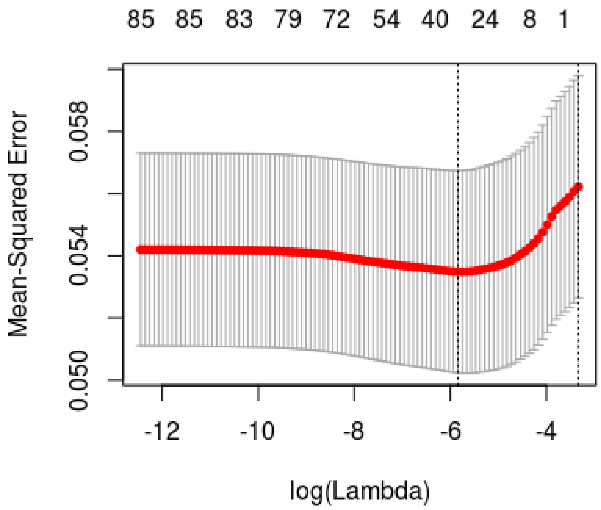
Graph: Ridge model cross validation.



Inference: It can be infered from the graph that, with increase in lambda, the mean squared error has increased indicating that the shrinkage of the coefficients has been too much as is leading to under-fitting.

- Running the Lasso model we get the following graph as the cross validation variation.

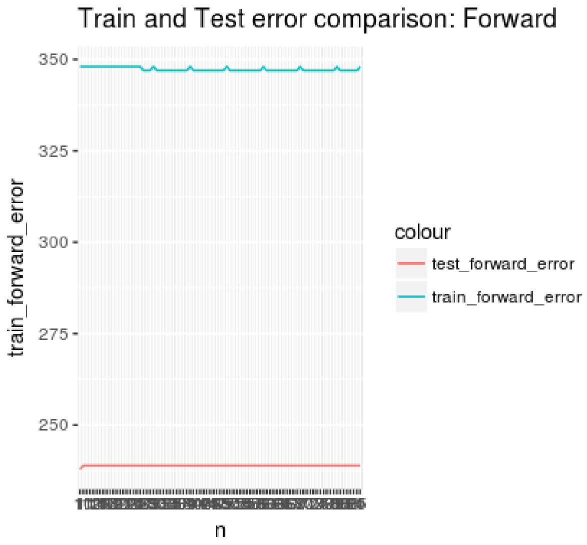
Graph: Lasso model cross validation



Inference: It can be infered from the graph that, with increase in lambda, the mean squared error has increased indicating that the shrinkage of the coefficients has been too much as is leading to under-fitting.

- Running a forward subset selection and running a regression model gives the graph below for every i. (‘i’ being the number of features to be included in the model)

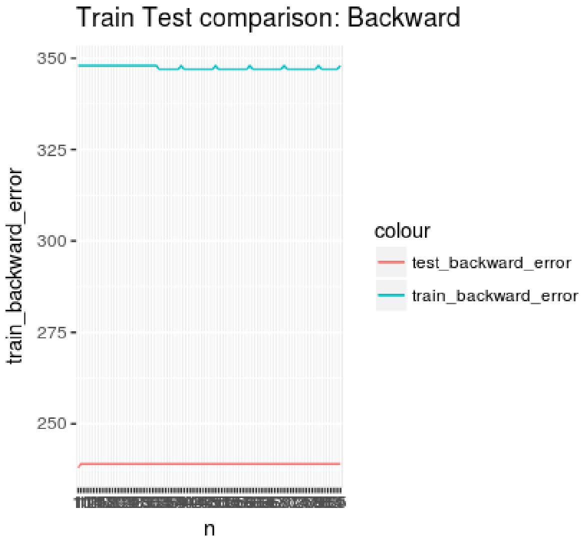
Graph: Forward subset selection.



Inference: It can be infered from the graph that for whatever i, there is no improvement in the predictions. Infact the number of 1’s in train and test are 348, 238 and since we are using a 1-0-lossfunction, we are predicting most of them wrong!

- Running a backward subset selection method, gives the following graph for every i (‘i’ being the number of features to be included in the model)

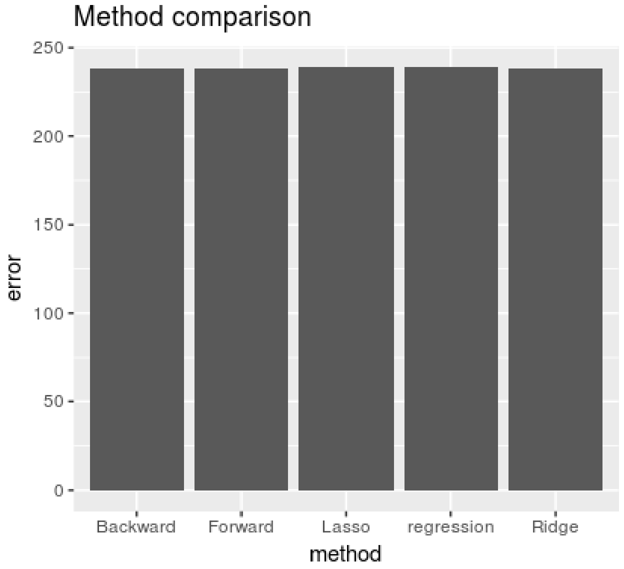
Graph: Backward subset selection



Inference: It can be infered from the graph that for whatever i, there is no improvement in the predictions. Infact the number of 1’s in train and test are 348, 238 and since we are using a 1-0-lossfunction, we are predicting most of them wrong!

- Comparing all the models that we have run, we get the following graph.

Graph: Different method comparison



Inference: It looks like all the methods are equally bad at predicting. Looking at the predictions we can conclude that with the current Data Mining techniques it is difficult to predict who is interested in buying a caravan insurance policy.

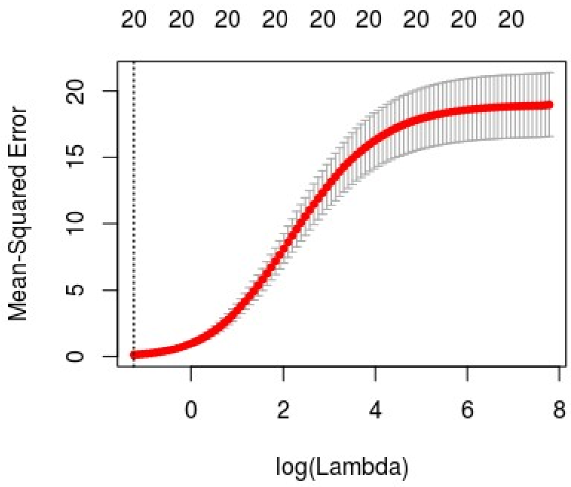
Q3)

- Generating random data – Most of the Machine Learning algorithms assume that the data is in gaussian distribution hence generating the numbers in gaussian distribution.
- Running a linear model and getting a summary gives us that, the model that we got is a 3-star model with most the features being significant.

In addition to the subset selection methods let’s **also run models ridge, lasso** to get an understanding of which model is doing better.

- Running a ridge model on the dataset and generating a plot, gives us the following.

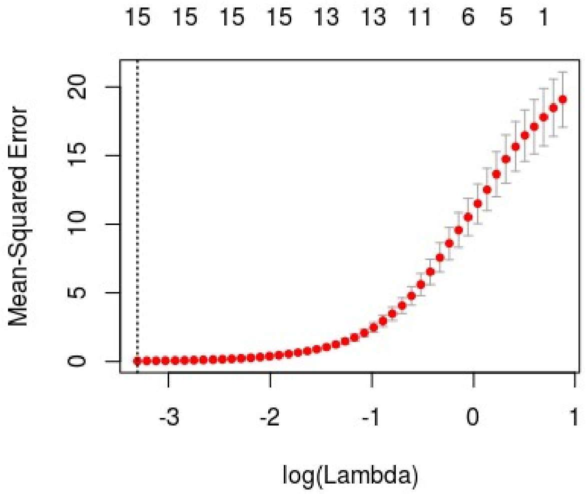
Plot: Ridge model



Inference: It can be infered from the graph that, with increase in lambda, the mean squared error has increased indicating that the shrinkage of the coefficients has been too much as is leading to under-fitting.

- Running a Lasso model on the dataset and generating a plot, gives us the following.

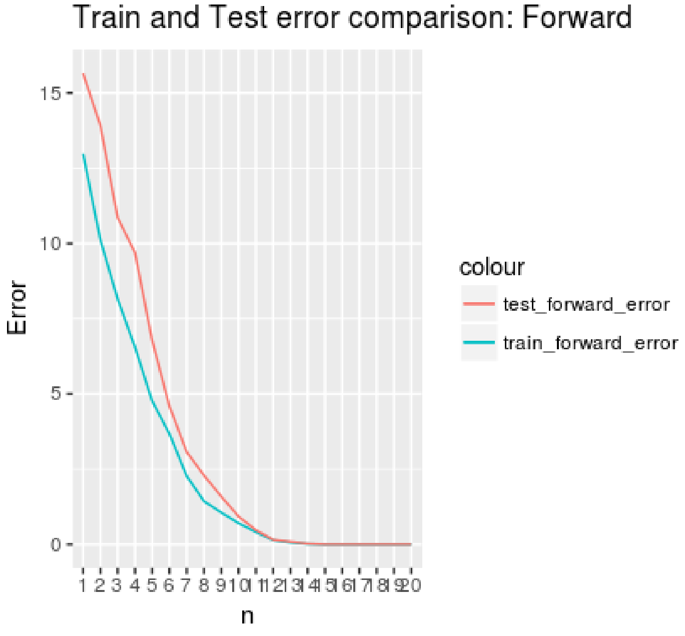
Plot: Lasso model.



Inference: It can be infered from the graph that, with increase in lambda, the mean squared error has increased indicating that the shrinkage of the coefficients has been too much as is leading to under-fitting.

- Running a forward subset selection and running a regression model gives the graph below for every i. ('i' being the number of features to be included in the model)

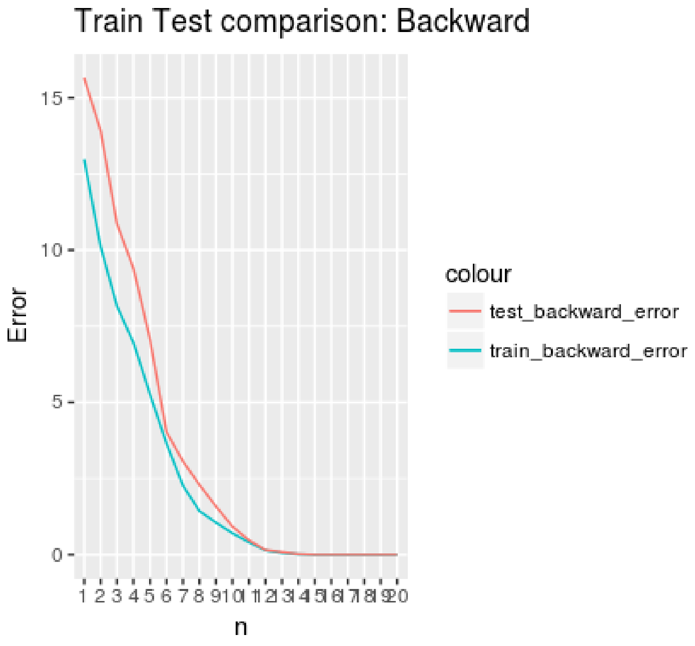
Plot: Train and Test error comparison in Forward subset selection



Inference: The optimal number of features look like 12 for this dataset. Since after that, there is no major variation in error and we look for minimalism in number of features.

- Running a backward subset selection and running a regression model gives the graph below for every i. ('i' being the number of features to be included in the model)

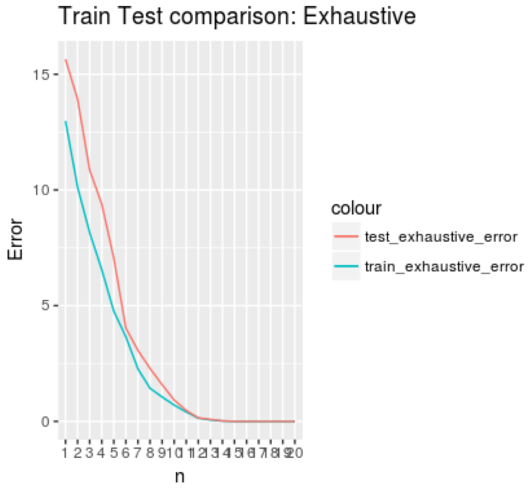
Plot: Train and Test comparison in Backward subset selection.



Inference: The optimal number of features look like 12 for this dataset. Since after that, there is no major variation in error and we look for minimalism in number of features.

- Running a exhaustive subset selection and running a regression model gives the graph below for every i. ('i' being the number of features to be included in the model)

Plot: Train and Test comparison in exhaustive subset selection.

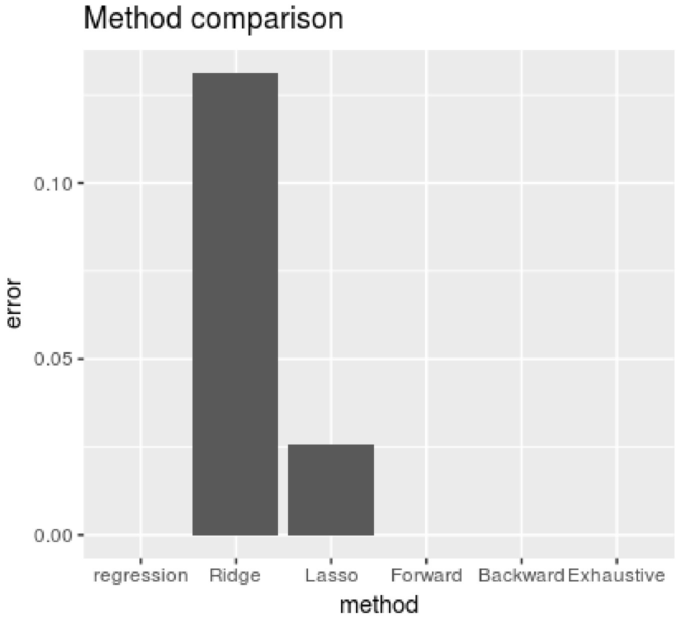


Inference: The optimal number of features look like 12 for this dataset. Since after that, there is no major variation in error and we look for minimalism in number of features.

The test error curve is expected to rise after reaching minimum. But despite using different distributions to generate data and do simulation multiple times the test curve is not rising up.

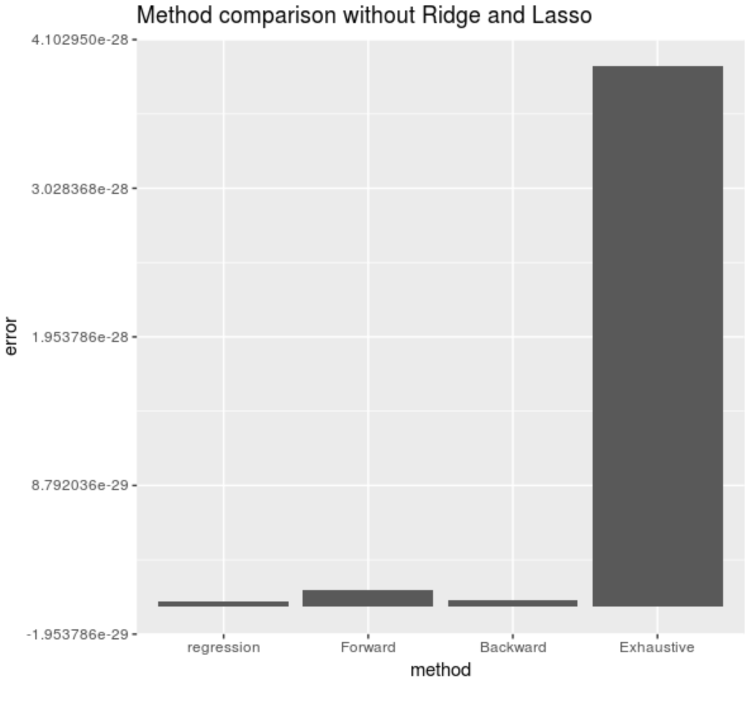
- Looking at the comparison of the methods we have attempted so far, let’s look at their test errors.

Plot: Method comparison



Inference: The error from Ridge and Lasso seem to be very high. Let’s remove Ridge and Lasso and plot the errors.

Plot: Method comparison with out Ridge and Lasso



Inference: Out of all the methods that were tried regression seem to have performed better since it's test error is the lowest.

Comparing the coefficients that are used to generate the data and the coefficients of regression we see that they are almost same. Doing a correlation between the two, we get correlation as 1!

And also like we expect, the coefficients that we made 0, in the code have become 0 in linear regression model indicating that those features are un-important (has nothing to do with data).