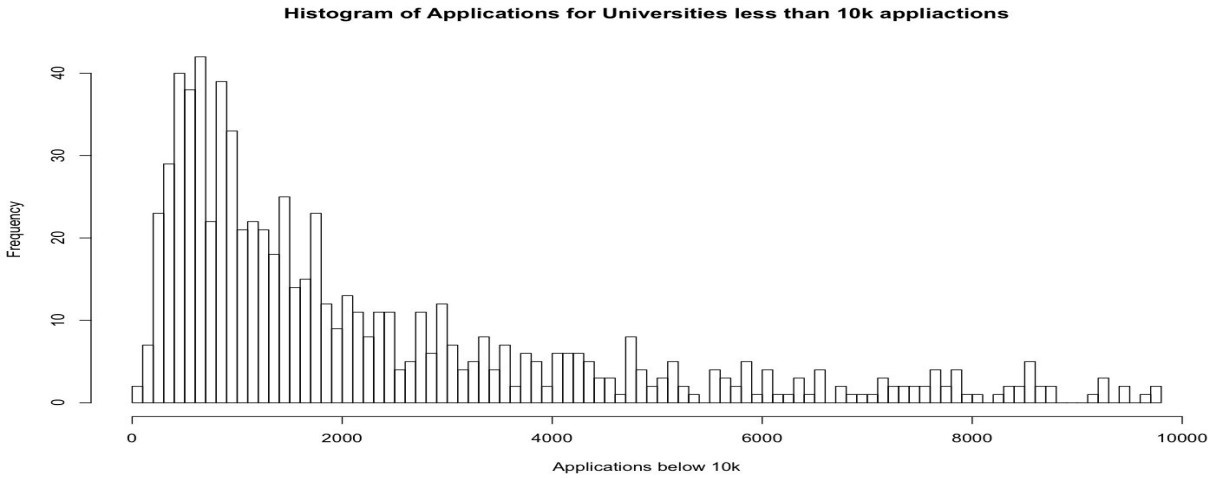
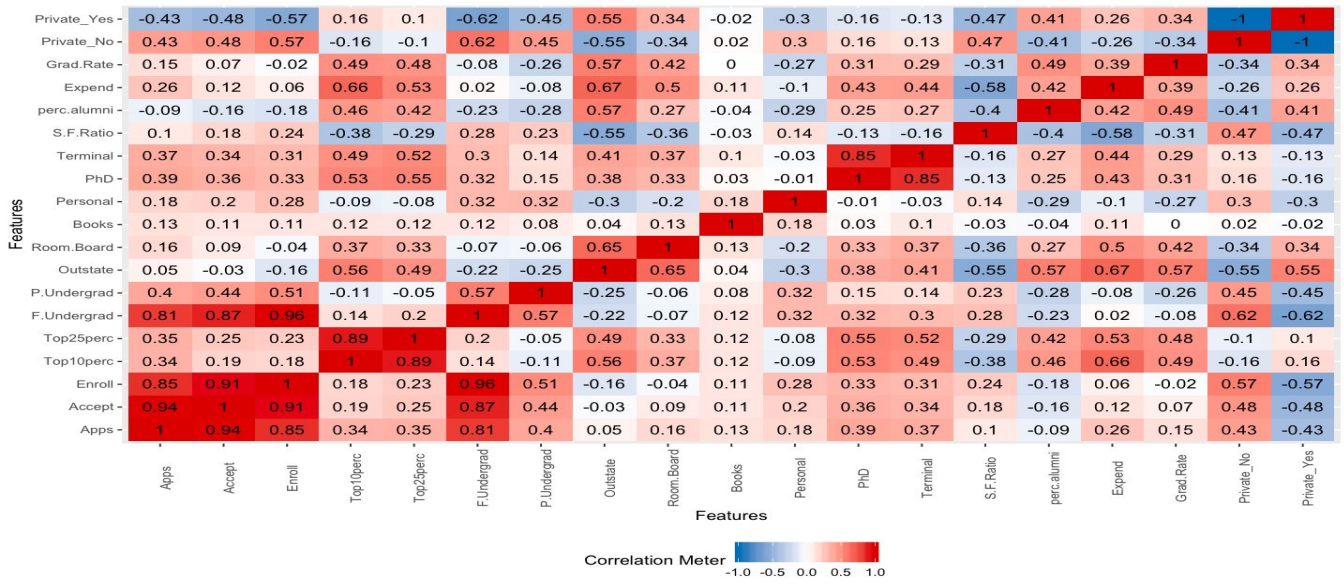
The Regression Fit:

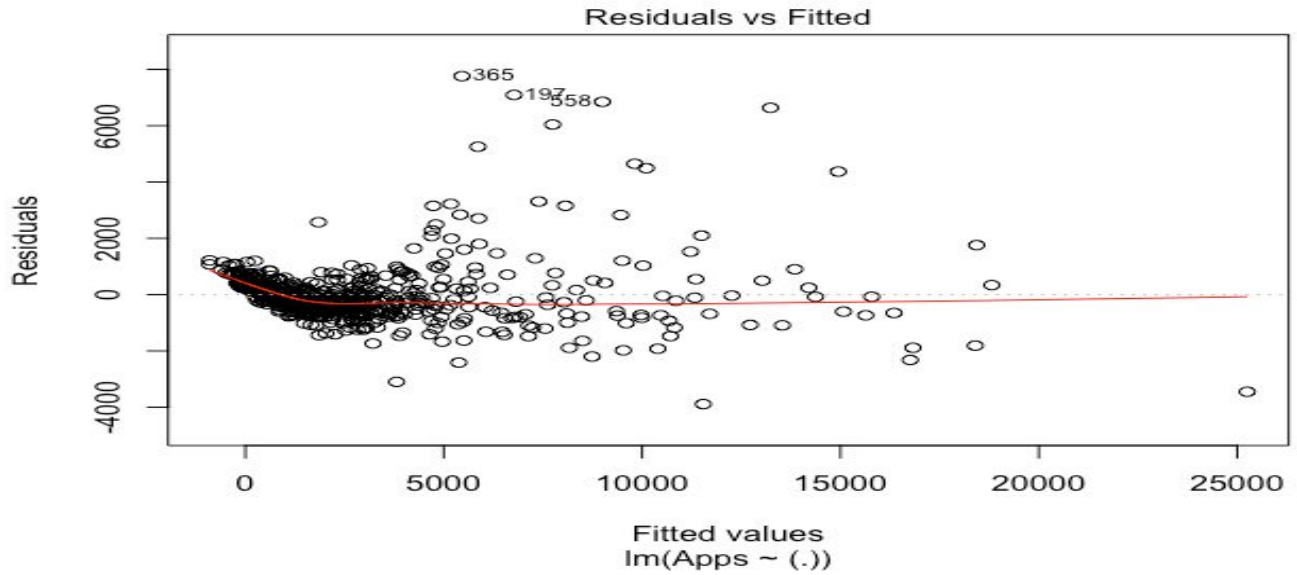
We want to predict the number of applications received using the college dataset. The linear model assumes that the dependent and independent variables are in a normal distribution. Here we can see that the dependent variable is in not a normal distribution and is left-skewed but as long as our kurtosis is not huge we should be fine. We can see that the kurtosis for applications is approximately 6(removing one outlier which has 48k applications) which while not perfect shouldn't affect our residuals too much ideally. We will check for homoscedasticity and independence.

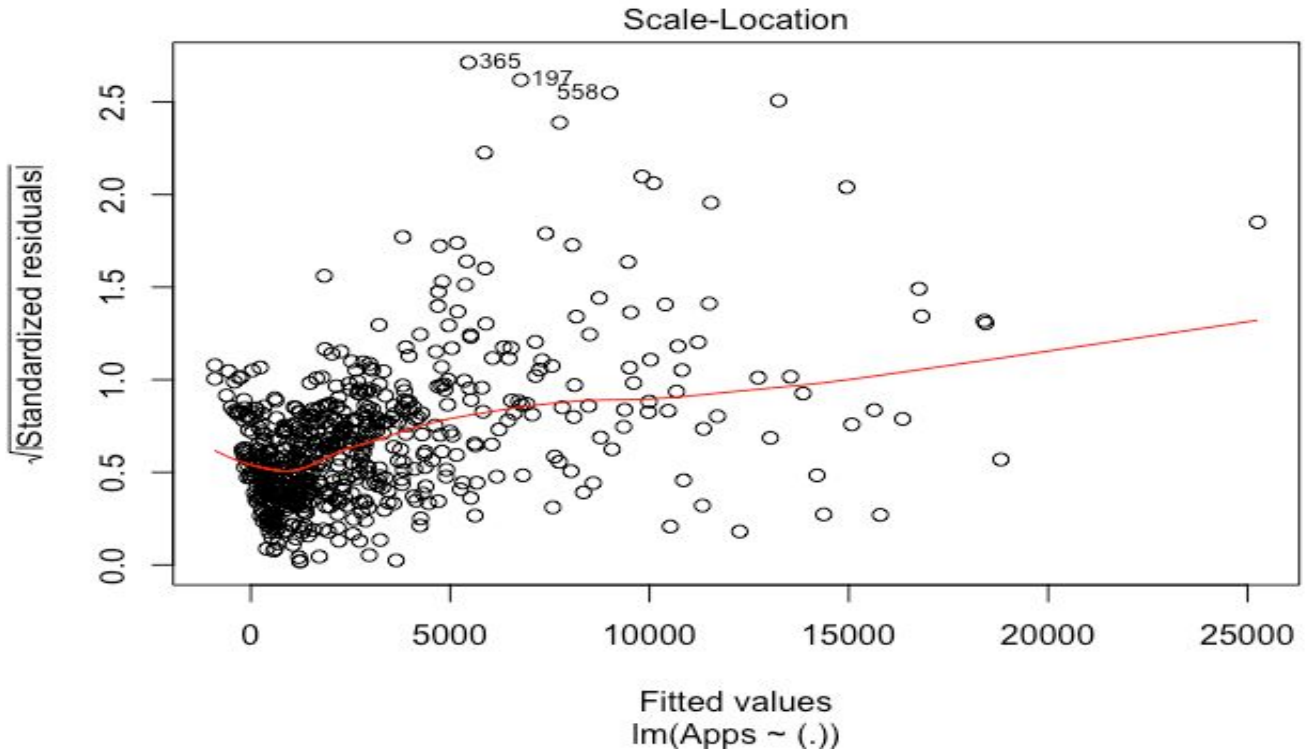


We can see from the correlation plots that 3 variables Accept, Enroll and F Undergraduate are highly correlated so we can remove two of these. Since Accept is the one which is most correlated with Apps we can remove the other two. Also for the same reason, we remove Terminal and Top10perc

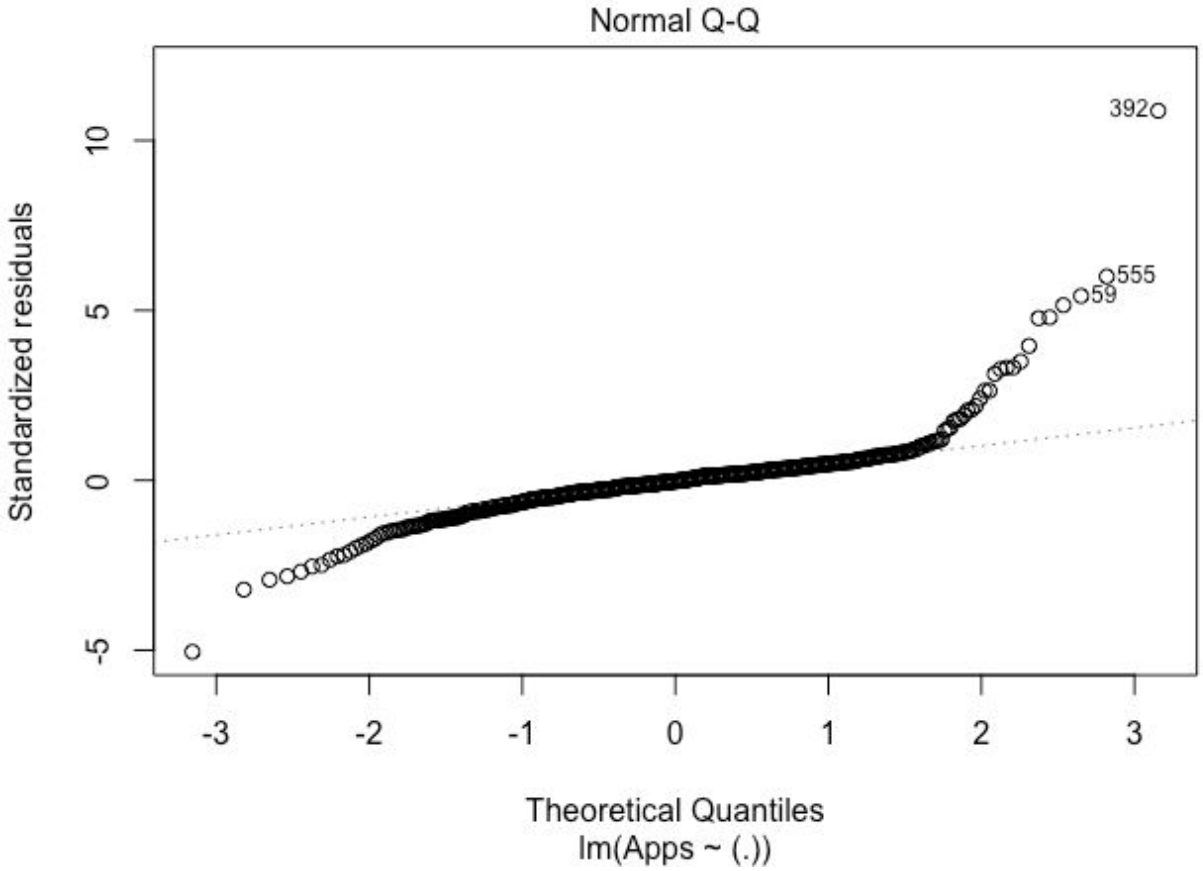


As we can see from the plot of residuals and fitted values, they are not totally homoscedastic as the distribution of points is not equal on the x-axis and the red lines are not entirely flat. We can accept such deviance





From the Q-Q plot, we can see that the standard residuals are a good fit for data which is in the middle but not so much for data that are towards either end.



Since the samples are quite low(only 777), we will go ahead with this model, but ideally, we would have liked to remove some data and re-build our data to remove heteroscedasticity and make sure our residuals are in a straight line. Also, as we should have ~385 data points to test so that we have enough range of data covered so that Confidence Interval >= 95%. For this dataset, we can use a 50:50 split so that we have about 390 data points in the test and train.

The training statistics are as follows:

- RSS: 1036
- R squared: 0.9273
- Adjusted R squared: 0.9247.

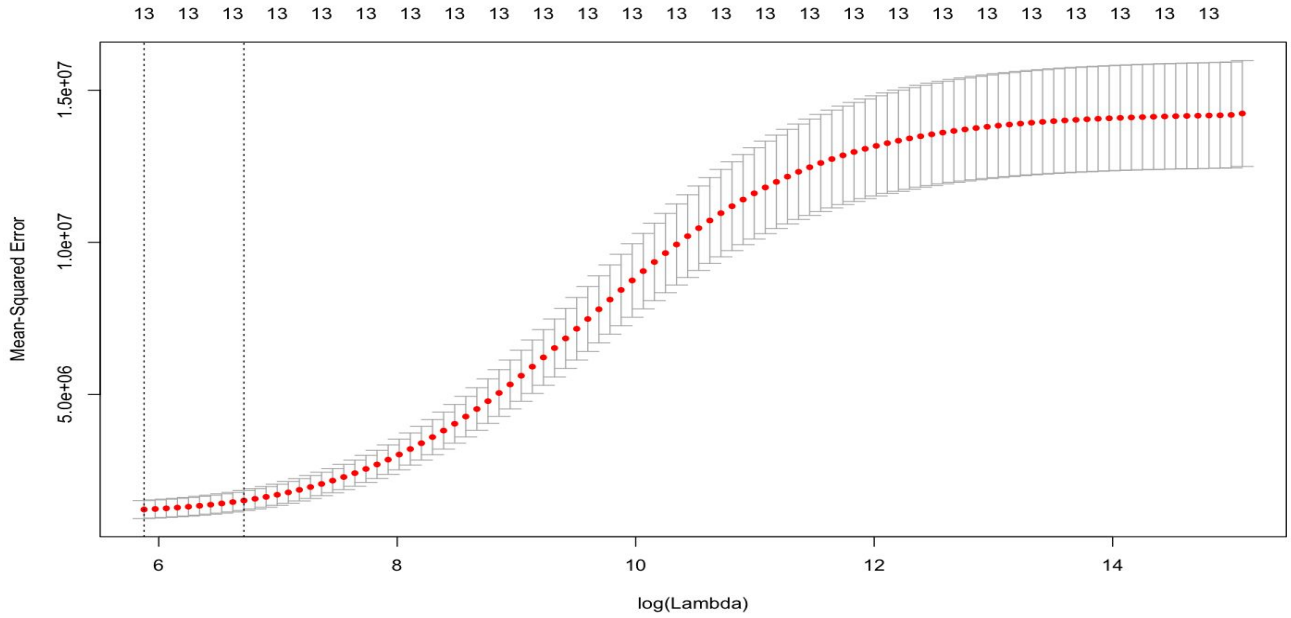
The variables which are most significant are Accept, Top25perc and Expend.

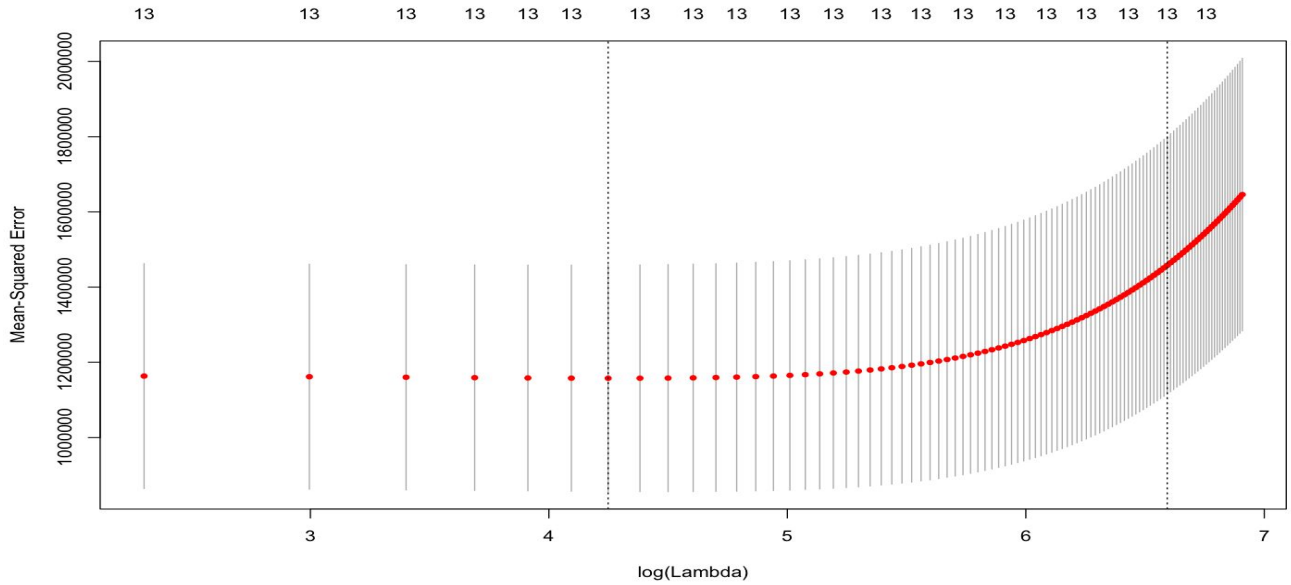
Q1.

a) The test error obtained from the linear model is as follows:

- RMSE: 1034.348
- R squared: 0.8981584

b) We can see that the minimum lambda that is computed by the method cv.glmnet on its own is the minimum of possible values which were scored by the model on its own. This lambda value is 356.7226. So there is a scope for reduction of lambda further which the model has not considered. Simulating lambda values between 1000 and 0 in steps of 10 we can see that the optimal lambda is 60.

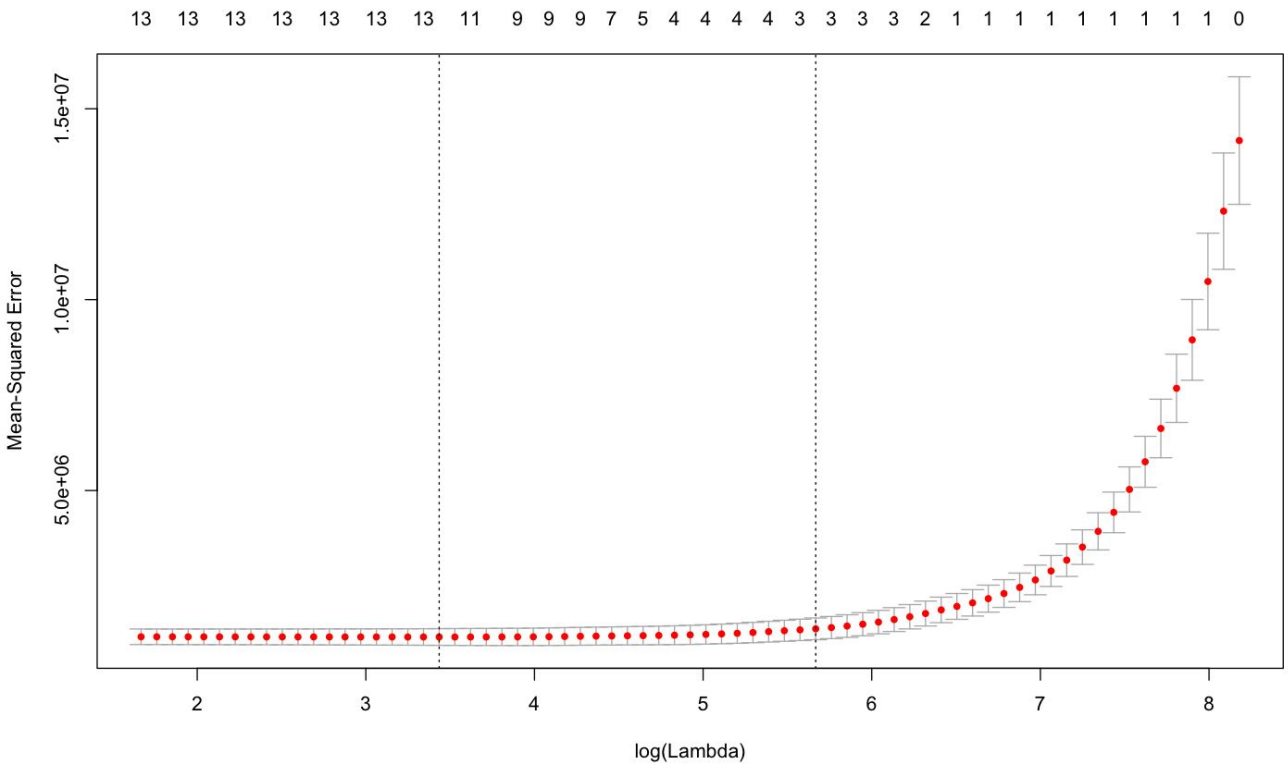




The test error obtained from the ridge regression with lambda = 60 is as follows:

- RMSE: 1041.323
- R squared: 0.8974535

c) The minimum lambda being computed by cross-validation for 21.38496 and the minimum of the lamdas taken is 5.297. So we can be confident that this is an appropriate lambda for which the error has been minimized. However, we can see from the graph that for lambda min none of the coefficients are reducing to zero hence, simulating for lambda at one standard deviation which is 348.5216 we can see only 3 independent variables are left and all else have been reduced to zero from the plot. We can see that R squared isn't much less compared to 13 variables for 3 variables so it is a good approximation.



The test error obtained from lasso regression with best lambda = 21.385 is as follows:

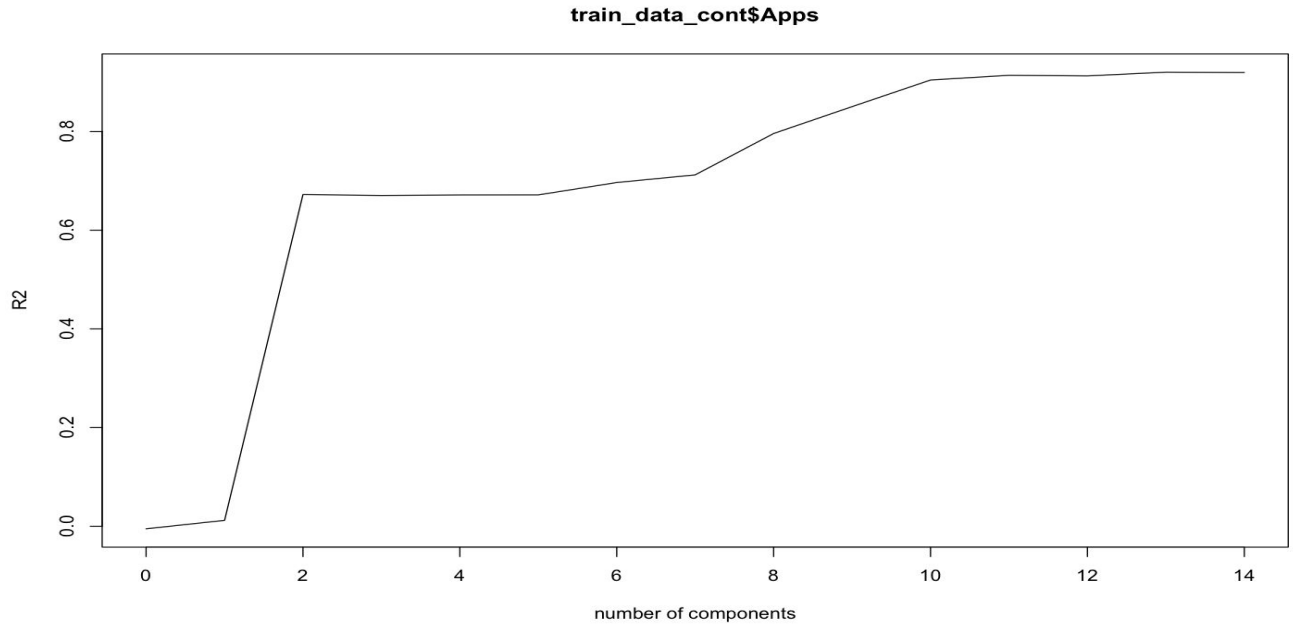
- RMSE: 1040.558
- R squared: 0.8974127

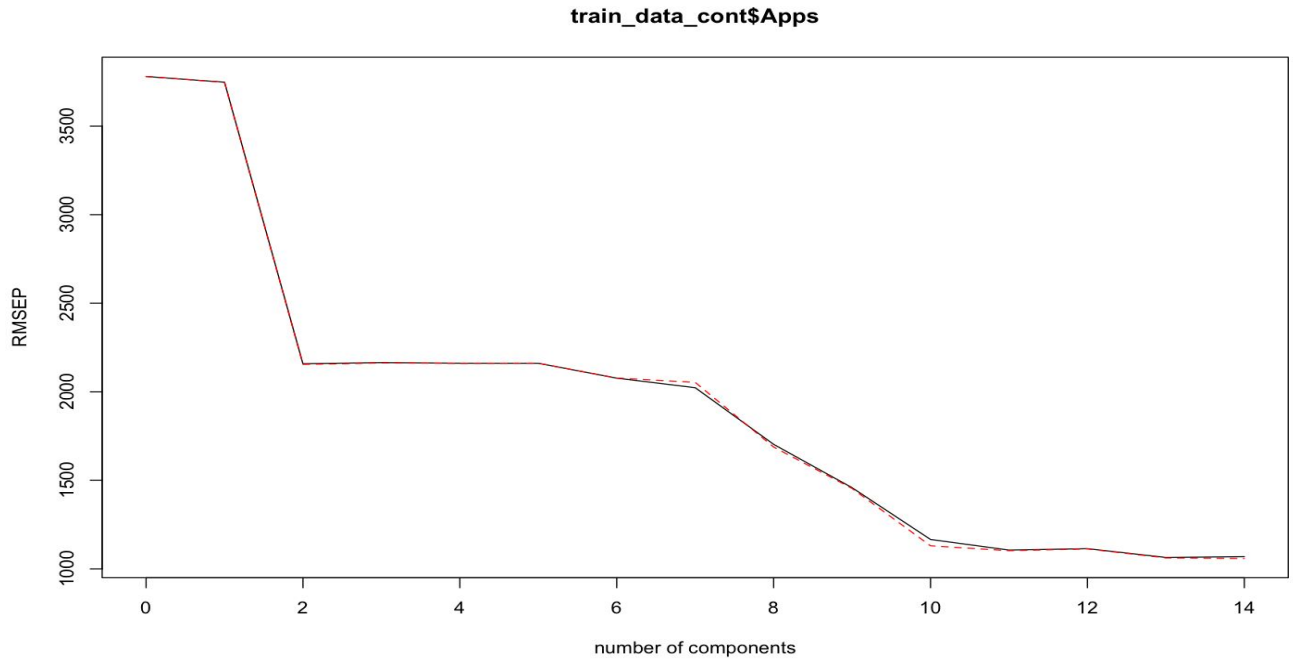
The test error obtained from lasso regression with std dev 1 lambda = 348.5216 is as follows:

- RMSE: 1207.665
- R squared: 0.8780836

The three non-zero coefficients are Accept, Top25perc and Expend.

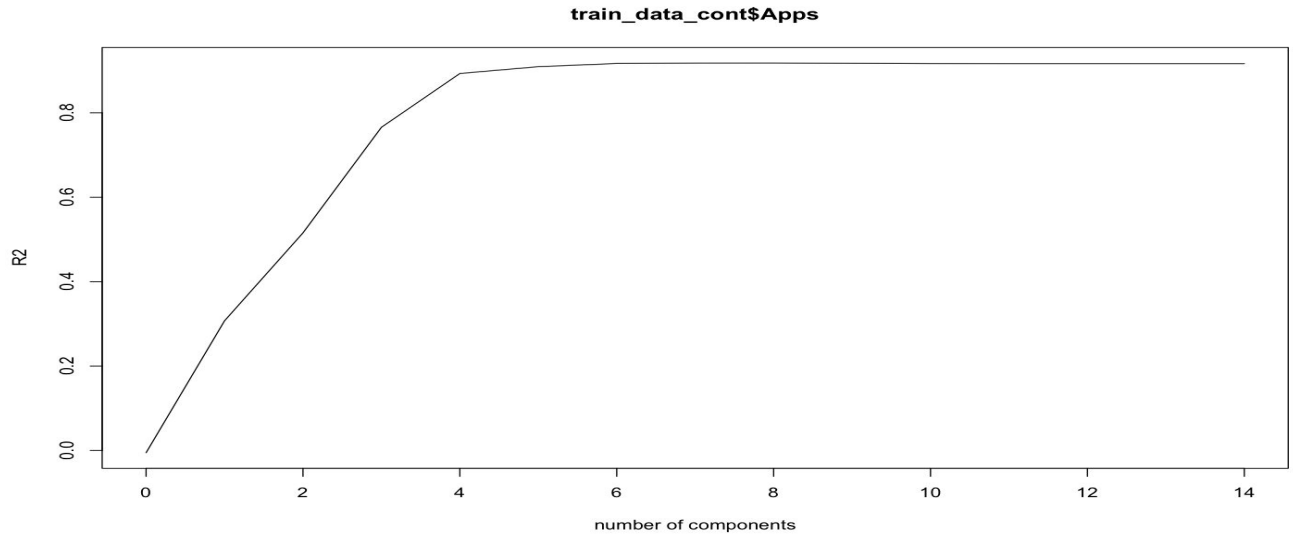
d) We can see from the plot that approximately 2 components give us 75% R squared on the training dataset and 10 components give us ~90% R squared. So we can choose 10 components which will give us a good R squared and reduce our variables.

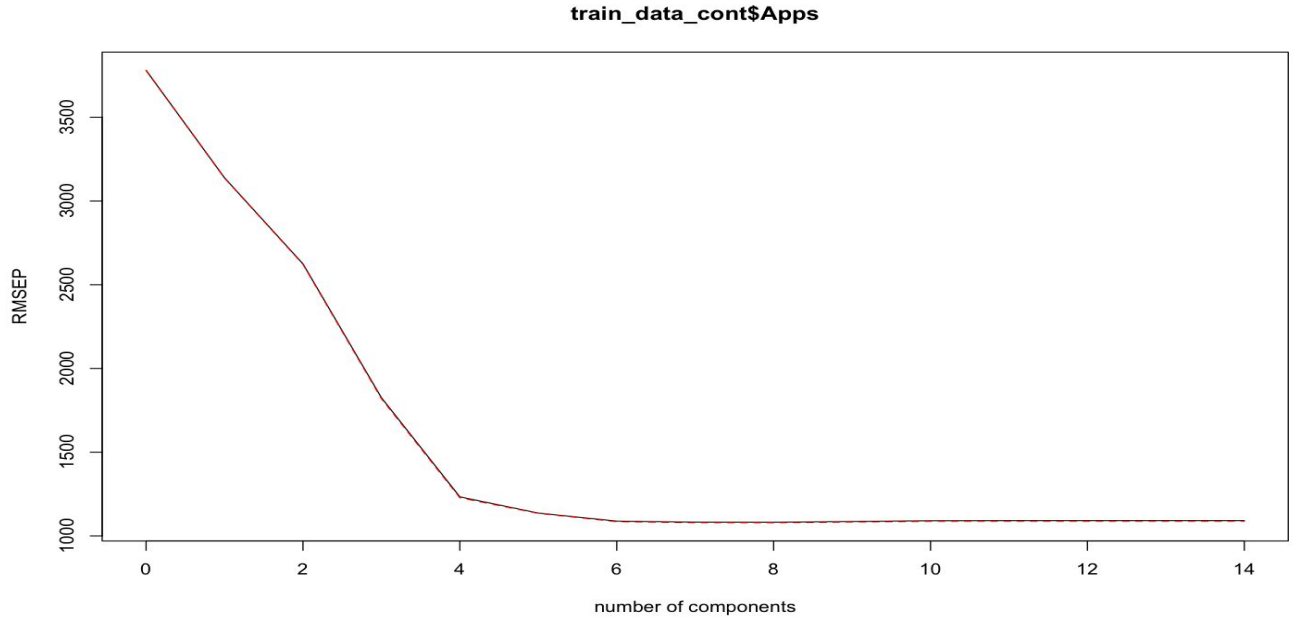




Looking at results we can see that test RMSE and R squared for 2 components are 1962.219 and 0.6353083 while the test RMSE and R squared for 10 components are 1103.3146 and 0.8839495. So we will choose 10 components as it gives a significantly better result than 2 components

e) We can see from the plot that approximately 4 components give us ~90% R squared on the training dataset. The RMSE reduces till 6 so we can choose 4 and 6 component and check our results.





Looking at results we can see that test RMSE and R squared for 4 components are 1237.698 and 0.8540396 while the test RMSE and R squared for 6 components are 1047.789 and 0.8959965. So we will choose 4 components as we have good R squared with lesser components.

The best solution of all is Lasso with a lambda value of one standard error as it has pretty high R squared of ~87 and the number of applications can be predicted to a high accuracy up to an error range of +-1040 applications using just Accept, Top25perc and Expend as independent variables

2) The following are the results are of various models which were run:

Model	#Variables	Accuracy	Test Precision	Test Recall
Linear Regression	85	0.94025	0.3333333	0.004201681
Stepwise Forward	47	0.94025	0.3333333	0.004201681
Stepwise Backward	32	0.94025	0.3333333	0.004201681
Ridge Regression	85	0.94025	0	0
Lasso Regression	29	0.94025	0.3333333	0.004201681

As we can see the precision and recall are pretty poor for all the models and even though the accuracy is high hence, the models are not correct for the dataset. This is due to the high-class imbalance in the dataset as we can see that the percentage of 1's in the dataset is close to only 6% rather than close to 50% which we have seen before. We will have to solve for this class imbalance using under-sampling or over-sampling the smaller class in the dataset but it is beyond the scope of the problem right now. Ridge regression has all predicted values lower than 0.5. This value has to be dynamically altered for maximum AUC.

3) The figure shown below is the plot for training and test MSE for the dataset. The training MSE has been depicted using a blue line while the test MSE has been depicted using the black line for various models of varying variables for best subset selection. The red dot shows the model with 12 variables is the one with least MSE for test dataset while the green dot shows that the model with 20 variables is the one with least MSE for the training dataset. The model which is minimized for test MSE will have almost the same number of non-zero coefficients as the training dataset and values of coefficients will be close to those of the original equation. The coefficients close to zero in the original equation may vanish to zero. The below model had 13 non-zero coefficients in the training dataset while the best test model is one with 12 coefficients.