STAT 6543 Exam

Santanu Mukherjee , zes254

07/16/2022

R Markdown

I True (T) or False (F). (20 Points)

Q1

For these problems, if false, briefly justify your answer. Each problem worth 2 points.

1 __**F**_ In general the more flexible a method is, the lower its RMSE of the test data will be.

Reason

As the model becomes flexible, initially the test RMSE will decrease , but in the long run as flexibility increases, the test RMSE will increase.

2 __**T**_ When we fit the linear regression model, the collinearity between predictors will improve the coefficient estimates.

Reason

A collinearity is a special case when two or more variables are correlated. Then these variables will have an influence on the coefficients of the estimates that is being generated by the model. As such it is wiser to remove collinearity while building the model so that model performance can be interpreted properly.

3 __**F**_ All types of statistical models discussed in this course are beneficial from data pre-processing.

Reason

Although majority of models benefit from data pre-processing, not all models benefit from data pre-processing. For example, if the predictors are mostly binary variables, data pre-processing will accomplish very little.

4 __**T**_ One advantage of Principal component analysis (PCA) is that it is a data reduction technique which creates uncorrelated components.

Reason

A commonly used data reduction technique is PCA which seeks to find linear combinations of the predictors, known as principal components (PCs), which capture the most possible variance. The first PC is defined as the linear combination of the predictors that captures the most variability of all possible linear combinations. Then, subsequent PCs are derived such that these linear combinations capture the most remaining variability while also being uncorrelated with all previous PCs.

5 __**T**_ The bias-variance trade-off means that as a method gets more flexible the bias will decrease and the variance will increase but expected RMSE of the testing data may go up or down.

Reason

As model becomes more flexible, bias will decrease and variance will increase. There will be areas of under fitting and there will be areas of overfitting and accordingly the testing error may go up and down.

6 __**T**_ The trade-off between prediction accuracy and interpretability means that a predictive model that is most powerful is usually the least interpretable.

Reason

The predictive models that are most powerful are usually the least interpretable. The perceived improvement in interpretability gained by manual categorization is usually offset by a significant loss in performance.

7 __**F**_ When the sample size n is extremely large, and the number of predictors p is small, we do not expect the performance of a flexible statistical learning method to be better than an inflexible method.

Reason

In this scenario, a flexible method will fit the data closer and with the large sample size, would perform better than an inflexible approach.

8 __**T**_ Elastic net, OLS, Ridge regression, Lasso regression can all be used and implemented in situations where the number of predictors is larger than the sample size.

Reason

These are examples of multiple linear regression and here number of predictors can be larger than the sample size. Here shrinkage methods are employed to shrink the coefficients of predictors towards 0. Although shrinkage introduce bias, it can significantly reduce variance. The shrinkage also potentially reduces test error.

9 __**F**_ The bootstrap is a widely applicable and extremely powerful statistical tool that can be used to quantify the uncertainty associated with a given estimator. Each “bootstrap set” is created by sampling without replacement, and the size is smaller than our original dataset.

Reason

Bootstrap set is created by resampling **WITH** replacement.

10 __**F**_ The last name of the instructor of this course is Min.

Reason

The name of the instructor is Min Wang, where Min is the first name and Wang is the last name.

II Free Response Questions (40 Points)

Problem 1 (a)

Problem 1 (Total: 10 Points) You think of some real-life applications for statistical learning and predictive modelling.

a. Describe a real-life application in which classification might be useful. Describe the response, as well as the predictors. Is the goal of this application inference or prediction? Clearly explain your answer. (5 Points)

Answer 1 (a)

One example of classification problem is to predict whether an email is a *Spam* or not. If it is a *Spam*, it needs to be moved to a Spam folder. The response variable is binary let's say (y) : *Spam* or not
The predictors are simple sequence of words , lets say (X).
The goal of the application is prediction.

Problem 1 (b)

b. Describe a real-life application in which regression might be useful. Describe the response, as well as the predictors. Is the goal of this application inference or prediction? Clearly explain your answer. (5 Points)

Answer 1 (b)

In today's environment where the inflation is high and gas prices are rising because of war and supply chain problems, there is no better real life example of a regression problem than to predict the total cost of driving from city A to city B. We can think of a linear regression model, where the regression equation is
Total Cost (y) = β_0 + $\beta_1 x_{miles}$ + $\beta_2 x_{gas-price}$ + ϵ,
y is the response variable
x_{miles}, $x_{gas-price}$ are the predictors
β_1 , β_2 are the co-efficients
β_0 is the Intercept

Problem 2 (a)

Problem 2 (Total: 10 Points) During the class time, we learned k-fold cross-validation.

a. (5 points) Explain how k-fold cross-validation is implemented.

Answer 2 (a)

$k - fold$ cross validation is one of the resampling techniques that are used for model performance and optimization. In this method, the samples are randomly partitioned into k sets ("folds") of roughly equal size. A model is fit using all the $k-1$ samples except the $first$ subset (called the first fold). The held-out samples are predicted by this model and used to estimate performance measures. The $first$ subset is returned to the training set and procedure repeats with the $second$ subset held out, and so on. The k resampled estimates of performance are summarized (usually with mean and standard error) and used to understand the relationship between the tuning parameter(s) and model utility.

A special case where k is the number of samples is $Leave - One - Out - Cross - Validation(LOOCV)$. In this case, since only one sample is held-out at a time, the final performance is calculated from the k individual held-out predictions. Additionally, repeated k-fold cross-validation replicates the procedure multiple times. For example, if 10-fold cross-validation was repeated five times, 50 different held-out sets would be used to estimate model efficacy.

The choice of k is usually 5 or 10, but there is no formal rule. As k gets larger, the difference in size between the training set and the resampling subsets gets smaller. As this difference decreases, the bias of the technique becomes smaller (i.e., the bias is smaller for k = 10 than k = 5). In this context, the bias is the difference between the estimated and true values of performance.

Problem 2 (b)

b. (5 points) What are the advantages and disadvantages of k-fold cross-validation relative to the bootstrap sample.

Answer 2 (b)

Advantages:

1. In $k - fold$ **Cross-validation**, bias is reduced as gives us an idea about how the model will perform on an unknown dataset.
2. $k - fold$ **Cross-validation** helps to determine a more accurate estimate of model prediction performance than **Bootstrap**
3. $k - fold$ **Cross-validation** as low bias compared to **Bootstrap**

Disadvantages:

1. In $k - fold$ **cross-validation**, there is higher uncertainty in error rates compared to **Bootstrap**
2. $k - fold$ **cross-validation** is computational heavy compared to **Bootstrap** as in **Bootstrap** not all samples are picked.

Problem 3

Problem 3 (Total: 10 Points)

What are the advantages and disadvantages of a very flexible (versus a less flexible) approach for regression? Under what circumstances might a more flexible approach be preferred to a less flexible approach? When might a less flexible approach be preferred?

Answer 3

Flexible method advantages:

1. As model becomes more flexible, bias decreases (less assumptions about the functional form of **f**)
2. Flexible model do help capture non-linear relationships or relationships between interaction variables
3. They capture complex variable interactions
4. Flexible models require less assumptions
5. Often outperform less flexible methods in prediction accuracy

Flexible method disadvantages:

1. It is evident from the biased-variance trade off graph that variance increases as model becomes more flexible (easier to overfit)
2. Flexible models require tuning
3. For these models, training times increase (cross-validation or equivalent is necessary to mitigate high variance & tune hyper parameters)
4. These models require more variables and observations to work optimally
5. It is often less interpretable

The circumstances WHEN a more flexible approach is preferred to a less flexible approach

1. Their exists non-linear relationships and / or interaction variables are related
2. The primary objective is to have more predictive power and accuracy
3. There is sufficient computational power and tools for variance-controlling measures (e.g. cross-validation)
4. The number of variables in large
5. The amount of observations (data) is large

The circumstances WHEN a less flexible approach is preferred

1. The objective is to interpret the model
2. There is less computational power requirement
3. There are assumptions like linearity, specific variables are more important than others etc.
4. There is requirement of rationale as to why an observation has a particular prediction

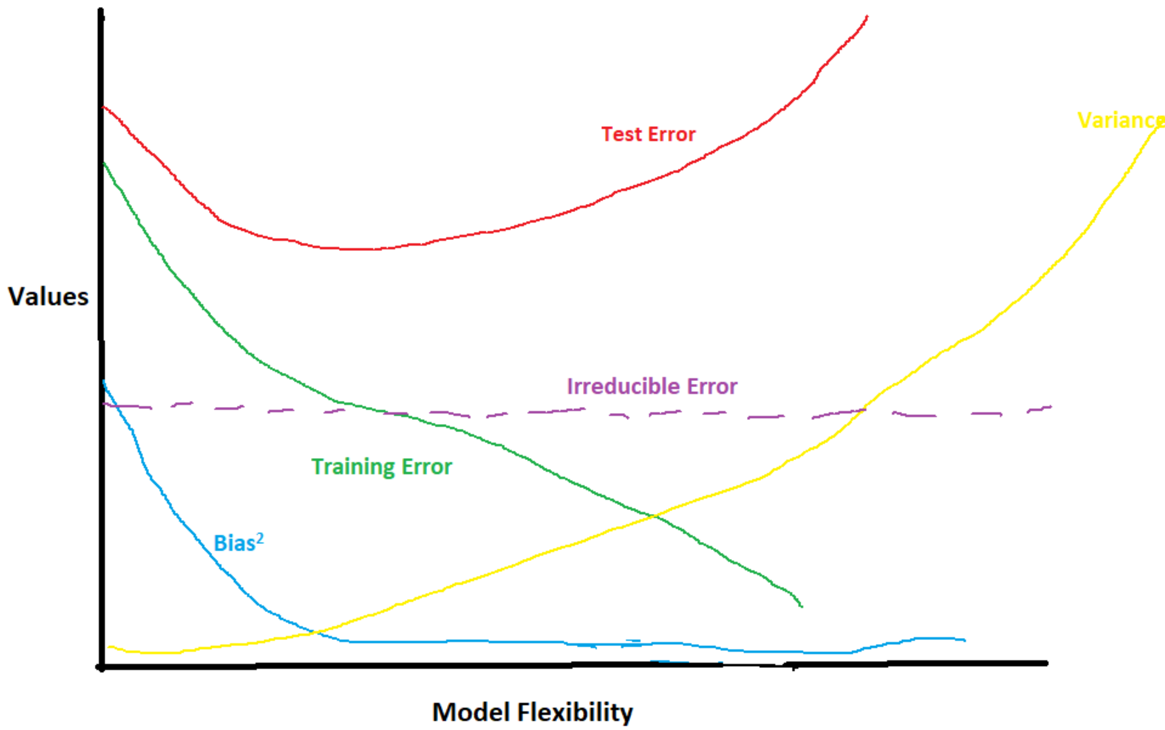
Problem 4 (a)

Problem 4 (Total: 10 Points)

In this class, we discussed the bias-variance trade-off. Answer the following questions.

a. Provide a sketch of typical (squared) bias, variance, training error, test error curves, on a single plot, as we go from less flexible statistical learning methods towards more flexible approaches. The x-axis should represent the amount of flexibility in the method, and the y-axis should represent the values for each curve. There should be four curves. Make sure to label each one.

Answer 4 (a)



Sketch with the five curves

Problem 4 (b)

b. Briefly explain why each of the four curves has the shape displayed in part (a)

Answer 4 (b)

Bias is the error introduced when the complexity of a problem is not sufficiently modeled by the simplicity of the chosen method (e.g. linear regression for non-linear relationships). As model flexibility increases (linear->trees->boosting, decreasing K in KNN, etc.), bias decreases monotonically, because less assumptions are being made about the data structure and its relationship with the response.

Variance refers to the amount by which our predictions would change if the training data were changed, and can be thought of as the error introduced when a model is overfit to the training data. As model flexibility increases, variance increases monotonically, because the method becomes more specified (and then overspecified) to the nuances of the training data, to the point where $\hat{f}$ doesn't generalize to new data.

Training Error decreases monotonically as flexibility increases. More flexible methods are generally higher variance, and can learn more complex relationships more completely, but also run the risk of overfitting, which is seen where the training error and test error diverge. Think of a decision tree, where the number of terminal nodes = the number of training observations (this model will have 0 training error and a high test error).

Test Error decreases, levels-out then increases. The minima is the point of optimal bias-variance trade off, where E[(Y− $\hat{Y}$)^2]=[Bias($\hat{f}$ (X))]^2+Var($\hat{f}$ (X))+Var(ϵ) is minimized. To the right of this minima, the method is overfitting ($\hat{f}$ is too high variance to make up for its lack of bias), and to the left the method is under fitting ($\hat{f}$ is too high bias to make up for its lack of variance).

Irreducible Error refers to the error introduced by inherent uncertainty/noise in the system being approximated. It is constant and > 0 regardless of the flexibility of the model, because ϵ may contain unmeasured variables not in X that could be used to predict y, and because ϵ may contain unmeasurable variation in y that could not be accounted for in X even if we wanted to. This means that it doesn't matter how closely $\hat{f}$ models the 'true' function f, there will still be an (unknown) minimum error of Var(ϵ)>0.

III Coding Questions (40 Points)

Problem 5 (Total: 16 Points) Suppose we are interested in examining the relationship between the response variable sales and the amount of money spent advertising on the TV, radio, and newspapers (i.e, there are three predictors: TV, radio, and newspapers). We fit a multiple linear regression with four predictors (TV, radio, newspaper, and the TV and radio interaction term, denoted by TV:radio (TV:radio)) and obtain the following results:

Problem 5 (a)

a. (4 points) Provide an appropriate interpretation for the coefficient 1.91e-02.

Answer 5 (a)

Here sales is the response variable and there are 4 predictors (TV, radio, newspaper, and the TV and radio interaction term, denoted by TV:radio (TV:radio)). So, for a given amount of TV, radio and newspaper advertising, spending an additional $ 1 on TV advertising will result in an increase in sales (response variable) by approximately 1.91e-02 units.

Problem 5 (b)

b. (4 points) True or false: Since the coefficient for the TV and radio interaction term “TV:radio (TV:radio)” is quite small, there is very little evidence that this interaction term is important in predicting the response variable “sales”. Justify your answer.

Answer 5 (b)

Answer = False

Here are few things to consider:

Although the coefficient of the interaction term “TV:radio (TV:radio)” is quite small, the t-test performed to identify whether interaction exists has shown us that the p-value is extremely low, which suggests that the interaction term is significant and so it is very important to keep this interaction term in the model.

So, this interaction term is important in predicting the response variable “sales”.

Problem 5 (c)

c. (4 points) Suppose that the company has two options to split $12,000 for the three types of advertising: (i) invest equally $4,000 for each type of advertising, (ii)invest $6,000 for TV, $3,000 for radio, and $3,000 for newspapers. Which option should be recommended for the company. Justify your answer.

Answer 5 (c)

Based on 1. the p-values and their significance (importance) in prediction AND 2. the coefficients of the predictors AND 3. the importance and significance of the interaction term “TV:radio (TV:radio)” in model prediction

it is recommended that **option (ii)invest $6,000 for TV, $3,000 for radio, and $3,000 for newspapers is better**.

This is because TV and radio (separately and together) are more significant and can provide more sales and so their share of advertisement together ($ 6000 + $ 3000 = $ 9000) is more in **option (ii)** than $ 4000 + $ 4000 = $ 8000 in **option (i)**.

Problem 5 (d)

d. (4 points) Based on this model fit, which predictors are important in predicting the sales? In other words, explain what conclusions you can draw based on the p-values. Your explanation should be phrased in terms of sales, TV, radio, newspaper, and TV:radio (TV:radio), rather than in terms of the coefficients of the linear model.

Answer 5 (d)

Based on the p-values provided, it is clear that 1. newspaper is not significant and so not important for predicting the sales. 2. We also see that the p-values for TV and the interaction term “TV:radio (TV:radio)” are very small compared to the p-value for radio (which is also significant). 3. As the p-value of the interaction term is very small, this suggests that the interaction term is significant and so it is very important to keep this interaction term in the model. 4. This also proves that the model is not additive and the interaction term “TV:radio (TV:radio)” will play a significant role in prediction.

In conclusion, based on p-values we can say that TV, radio and interaction term “TV:radio (TV:radio)” are very important predictors in predicting the sales based on this model.

Problem 6 (a)

Problem 6 (Total: 24 Points) we will predict the number of applications received using the other variables in the College data set available in the R package ISLR, which can be accessed as follows.

```
library(ISLR)
data(College)
#data basic information
head(College)
```

```
##                             Private Apps Accept Enroll Top10perc Top25perc
## Abilene Christian University     Yes 1660   1232    721        23        52
## Adelphi University               Yes 2186   1924    512        16        29
## Adrian College                   Yes 1428   1097    336        22        50
## Agnes Scott College              Yes  417    349    137        60        89
## Alaska Pacific University        Yes  193    146     55        16        44
## Albertson College                Yes  587    479    158        38        62
##                             F.Undergrad P.Undergrad Outstate Room.Board Books
## Abilene Christian University        2885         537     7440       3300   450
## Adelphi University                  2683        1227    12280       6450   750
## Adrian College                      1036          99    11250       3750   400
## Agnes Scott College                  510          63    12960       5450   450
## Alaska Pacific University            249         869     7560       4120   800
## Albertson College                    678          41    13500       3335   500
##                             Personal PhD Terminal S.F.Ratio perc.alumni Expend
## Abilene Christian University     2200  70       78      18.1          12   7041
## Adelphi University               1500  29       30      12.2          16  10527
## Adrian College                   1165  53       66      12.9          30   8735
## Agnes Scott College               875  92       97       7.7          37  19016
## Alaska Pacific University        1500  76       72      11.9           2  10922
## Albertson College                 675  67       73       9.4          11   9727
##                             Grad.Rate
## Abilene Christian University        60
## Adelphi University                  56
## Adrian College                      54
## Agnes Scott College                 59
## Alaska Pacific University           15
## Albertson College                   55
```

```
dim(College)
```

```
## [1] 777  18
```

a. Appropriately split the data set into a training set (80%) and a test set (20%). [4 points]

Answer 6 (a)

Based on the question, randomly selected **80%** of the observations for the training set and **20%** for the test set.

```
college <- College
attach(college)

set.seed(1)
split <- sample(1:nrow(college), 0.8 * nrow(college))

train <- college[split,]
test <- college[-split,]

train_percent  = round(nrow(train) *100 / nrow(college),0)
test_percent = round(100 - train_percent,0)

print(paste("The training data set percent is ",train_percent,"%"))
```

```
## [1] "The training data set percent is  80 %"
```

```
print(paste("The test data set percent is ",test_percent,"%"))
```

```
## [1] "The test data set percent is  20 %"
```

Problem 6 (b)

b. Fit a linear model using least squares on the training set, and report the test error obtained. [5 points]

Answer 6 (b)

```
lm_model <- lm(Apps ~ ., data = train)
summary(lm_model)
```

```
##
## Call:
## lm(formula = Apps ~ ., data = train)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -5555.2  -404.6    19.9   310.3  7577.7
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -630.58238  435.56266  -1.448 0.148209
## PrivateYes  -388.97393  148.87623  -2.613 0.009206 **
## Accept         1.69123    0.04433  38.153  < 2e-16 ***
## Enroll        -1.21543    0.20873  -5.823 9.41e-09 ***
## Top10perc     50.45622    5.88174   8.578  < 2e-16 ***
## Top25perc    -13.62655    4.67321  -2.916 0.003679 **
## F.Undergrad    0.08271    0.03632   2.277 0.023111 *
## P.Undergrad    0.06555    0.03367   1.947 0.052008 .
## Outstate      -0.07562    0.01987  -3.805 0.000156 ***
## Room.Board     0.14161    0.05130   2.760 0.005947 **
## Books          0.21161    0.25184   0.840 0.401102
## Personal       0.01873    0.06604   0.284 0.776803
## PhD           -9.72551    4.91228  -1.980 0.048176 *
## Terminal      -0.48690    5.43302  -0.090 0.928620
## S.F.Ratio     18.26146   13.83984   1.319 0.187508
## perc.alumni    1.39008    4.39572   0.316 0.751934
## Expend         0.05764    0.01254   4.595 5.26e-06 ***
## Grad.Rate      5.89480    3.11185   1.894 0.058662 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 993.8 on 603 degrees of freedom
## Multiple R-squared:  0.9347, Adjusted R-squared:  0.9328
## F-statistic: 507.5 on 17 and 603 DF,  p-value: < 2.2e-16
```

```
pred_ols <- predict(lm_model, test)
ols_mse <- mean((pred_ols - test$Apps)^2)

print(paste("The MSE as the test error metric is ", round(ols_mse,0)))
```

```
## [1] "The MSE as the test error metric is  1567324"
```

Problem 6 (c)

c. Fit a ridge regression model on the training set, with λ chosen by cross-validation.Report the test error obtained. [5 points]

Answer 6 (c)

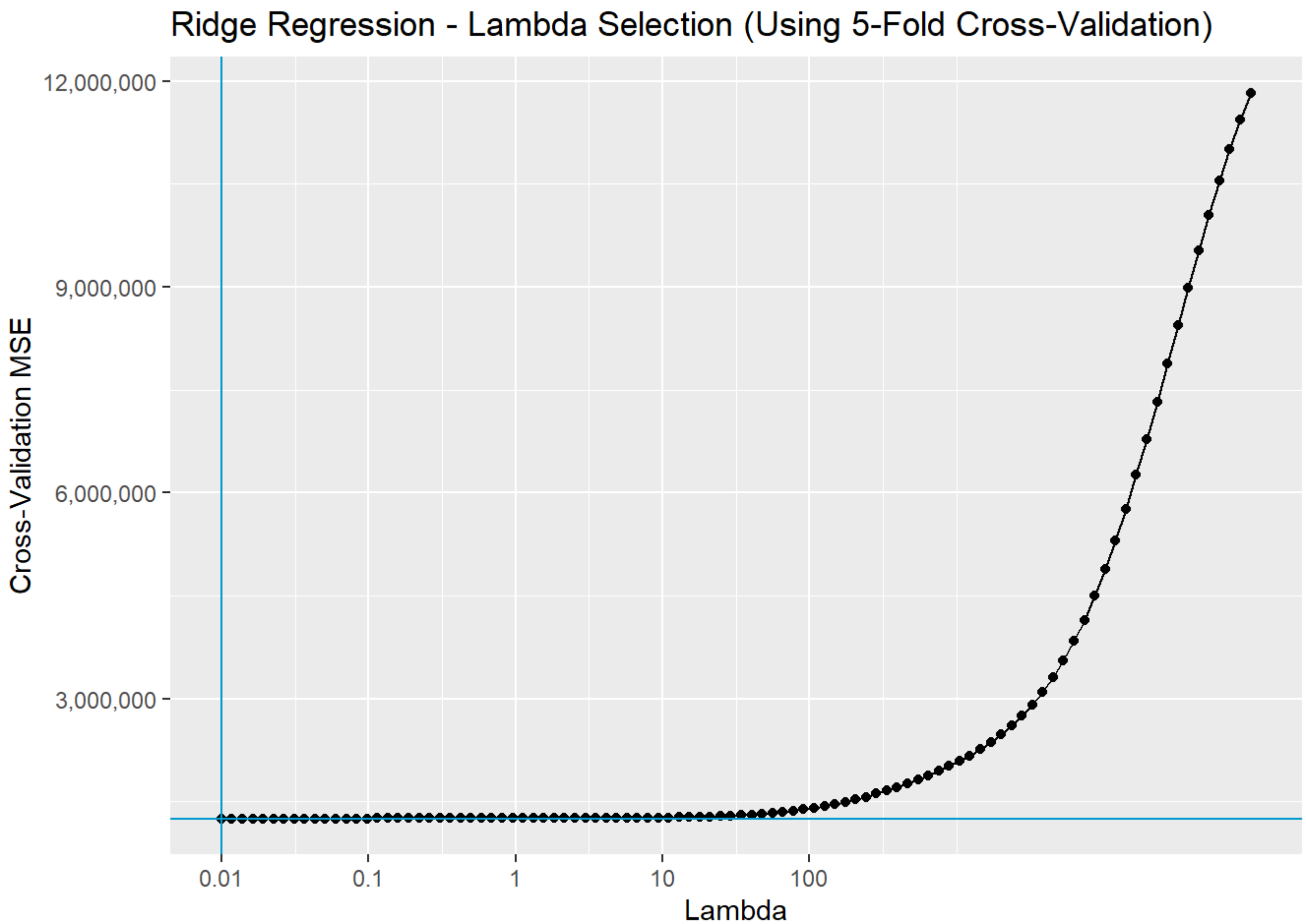
I first create *train.matrix* and *test.matrix*, which are train & test datasets.

```
train.matrix <- model.matrix(Apps~., data=train)
test.matrix <- model.matrix(Apps~., data=test)
grid = 10^seq(5,-2, length=100)
set.seed(3)
collegeridge <- cv.glmnet(train.matrix, train$Apps, alpha=0, lambda=grid)
bestLambda.ridge <- collegeridge$lambda.min


print(paste("The chosen value of lambda is ", bestLambda.ridge))
```

```
## [1] "The chosen value of lambda is  0.01"
```

I am here testing varying values of λ (from 0.01 to 100) using $5-fold$ cross-validation.

```
data.frame(lambda = collegeridge$lambda,
           cv_mse = collegeridge$cvm) %>%

  ggplot(aes(x = lambda, y = cv_mse)) +
  geom_point() +
  geom_line() +
  geom_vline(xintercept = collegeridge$lambda.min, col = "deepskyblue3") +
  geom_hline(yintercept = min(collegeridge$cvm), col = "deepskyblue3") +
  scale_x_continuous(trans = 'log10', breaks = c(0.01, 0.1, 1, 10, 100), labels = c(0.01, 0.1, 1, 10, 100)) +
  scale_y_continuous(labels = scales::comma_format()) +
  theme(legend.position = "bottom") +
  labs(x = "Lambda",
       y = "Cross-Validation MSE",
       col = "Non-Zero Coefficients:",
       title = "Ridge Regression - Lambda Selection (Using 5-Fold Cross-Validation)")
```




```
collegeridge_best <- glmnet(y = train$Apps,
                            x = train.matrix,
                            alpha = 0,
                            lambda = 10^seq(2,-2, length = 100))

ridge_pred <- predict(collegeridge_best, s = collegeridge$lambda.min, newx = test.matrix)
ridge_mse <- mean((ridge_pred - test$Apps)^2)

print(paste("The Ridge test MSE is ", round(ridge_mse,0)))
```

```
## [1] "The Ridge test MSE is  1567246"
```

Problem 6 (d)

d. Fit an ENET model on the training set with tuning parameters chosen by cross validation. Report the test error obtained, along with the number of non-zero coefficient estimates. [5 points]

Answer 6 (d)

```
# Elastic net model
library(elasticnet)

enetGrid = expand.grid(.lambda=seq(0,0.1,length=20), .fraction=seq(0.05, 0.5, length=20))
set.seed(0)
enet.model = train( model.matrix(train$Apps~., data=train)[,-1], train$Apps, method="enet",
                    # fit the model over many penalty values
                    tuneGrid = enetGrid,
                    preProcess=c("center","scale"), trControl=trainControl(method="repeatedcv",repeats=5) )

enet.model
```

```
## Elasticnet
##
## 621 samples
##  17 predictor
##
## Pre-processing: centered (17), scaled (17)
## Resampling: Cross-Validated (10 fold, repeated 5 times)
## Summary of sample sizes: 560, 558, 558, 561, 559, 559, ...
## Resampling results across tuning parameters:
##
##   lambda       fraction    RMSE      Rsquared   MAE
##   0.000000000  0.05000000  3295.055  0.8898515  2199.9802
##   0.000000000  0.07368421  3126.611  0.8898515  2084.1658
##   0.000000000  0.09736842  2959.652  0.8898515  1968.7338
##   0.000000000  0.12105263  2794.449  0.8898515  1853.5201
##   0.000000000  0.14473684  2631.340  0.8898515  1738.7323
##   0.000000000  0.16842105  2470.752  0.8898515  1624.2334
##   0.000000000  0.19210526  2313.234  0.8898515  1510.2070
##   0.000000000  0.21578947  2159.494  0.8898515  1396.9843
##   0.000000000  0.23947368  2010.463  0.8898515  1284.6008
##   0.000000000  0.26315789  1867.375  0.8898515  1175.3663
##   0.000000000  0.28684211  1731.881  0.8898515  1070.9617
##   0.000000000  0.31052632  1606.187  0.8898515   972.2609
##   0.000000000  0.33421053  1493.330  0.8899543   879.1495
##   0.000000000  0.35789474  1397.024  0.8909323   796.3299
##   0.000000000  0.38157895  1308.740  0.8975258   726.2647
##   0.000000000  0.40526316  1231.253  0.9053769   665.5190
##   0.000000000  0.42894737  1172.034  0.9112180   621.4351
##   0.000000000  0.45263158  1130.395  0.9154296   597.3770
##   0.000000000  0.47631579  1103.255  0.9184347   593.2234
##   0.000000000  0.50000000  1089.366  0.9205100   601.2019
##   0.005263158  0.05000000  3320.021  0.8898515  2217.2113
##   0.005263158  0.07368421  3163.118  0.8898515  2109.5212
##   0.005263158  0.09736842  3007.450  0.8898515  2002.1197
##   0.005263158  0.12105263  2853.224  0.8898515  1894.9286
##   0.005263158  0.14473684  2700.692  0.8898515  1788.0774
##   0.005263158  0.16842105  2550.164  0.8898515  1681.5170
##   0.005263158  0.19210526  2402.033  0.8898515  1575.2119
##   0.005263158  0.21578947  2256.791  0.8898515  1469.5518
```

```
##   0.005263158  0.23947368  2115.069  0.8898515  1364.4223
##   0.005263158  0.26315789  1977.683  0.8898515  1260.4307
##   0.005263158  0.28684211  1845.696  0.8898515  1159.4279
##   0.005263158  0.31052632  1720.504  0.8898515  1062.8311
##   0.005263158  0.33421053  1603.937  0.8898515   971.2192
##   0.005263158  0.35789474  1498.459  0.8898912   884.2910
##   0.005263158  0.38157895  1407.764  0.8905059   806.5372
##   0.005263158  0.40526316  1324.966  0.8956883   739.5436
##   0.005263158  0.42894737  1249.601  0.9033953   680.3581
##   0.005263158  0.45263158  1189.666  0.9094385   633.6580
##   0.005263158  0.47631579  1145.296  0.9138259   604.2262
##   0.005263158  0.50000000  1114.971  0.9171527   592.3275
##   0.010526316  0.05000000  3339.009  0.8898515  2230.3552
##   0.010526316  0.07368421  3190.899  0.8898515  2128.8643
##   0.010526316  0.09736842  3043.856  0.8898515  2027.5850
##   0.010526316  0.12105263  2898.044  0.8898515  1926.5732
##   0.010526316  0.14473684  2753.660  0.8898515  1825.7586
##   0.010526316  0.16842105  2610.949  0.8898515  1725.2670
##   0.010526316  0.19210526  2470.208  0.8898515  1624.9702
##   0.010526316  0.21578947  2331.807  0.8898515  1525.0053
##   0.010526316  0.23947368  2196.213  0.8898515  1425.7656
##   0.010526316  0.26315789  2064.012  0.8898515  1326.9133
##   0.010526316  0.28684211  1935.958  0.8898515  1229.6572
##   0.010526316  0.31052632  1813.021  0.8898515  1135.2732
##   0.010526316  0.33421053  1696.454  0.8898515  1045.2102
##   0.010526316  0.35789474  1587.877  0.8898515   959.4126
##   0.010526316  0.38157895  1489.418  0.8898723   878.0460
##   0.010526316  0.40526316  1404.778  0.8903996   805.4427
##   0.010526316  0.42894737  1326.811  0.8951967   742.2203
##   0.010526316  0.45263158  1255.971  0.9026120   685.9438
##   0.010526316  0.47631579  1197.380  0.9085301   639.6723
##   0.010526316  0.50000000  1153.356  0.9128850   608.5860
##   0.015789474  0.05000000  3353.744  0.8898515  2240.6302
##   0.015789474  0.07368421  3212.467  0.8898515  2143.9862
##   0.015789474  0.09736842  3072.135  0.8898515  2047.4948
##   0.015789474  0.12105263  2932.882  0.8898515  1951.3129
##   0.015789474  0.14473684  2794.874  0.8898515  1855.2417
##   0.015789474  0.16842105  2658.306  0.8898515  1759.4808
##   0.015789474  0.19210526  2523.419  0.8898515  1663.9187
```

```
##   0.015789474  0.21578947  2390.505  0.8898515  1568.5635
##   0.015789474  0.23947368  2259.928  0.8898515  1473.7708
##   0.015789474  0.26315789  2132.139  0.8898515  1379.4075
##   0.015789474  0.28684211  2007.708  0.8898515  1285.7109
##   0.015789474  0.31052632  1887.355  0.8898515  1194.0304
##   0.015789474  0.33421053  1772.001  0.8898515  1105.2421
##   0.015789474  0.35789474  1662.822  0.8898515  1020.7741
##   0.015789474  0.38157895  1561.315  0.8898515   939.8285
##   0.015789474  0.40526316  1469.424  0.8898810   863.5905
##   0.015789474  0.42894737  1390.836  0.8905178   796.0698
##   0.015789474  0.45263158  1317.935  0.8956044   736.9430
##   0.015789474  0.47631579  1252.376  0.9027433   683.8981
##   0.015789474  0.50000000  1197.557  0.9083188   639.5964
##   0.021052632  0.05000000  3365.150  0.8898515  2248.7244
##   0.021052632  0.07368421  3229.166  0.8898515  2155.8947
##   0.021052632  0.09736842  3094.035  0.8898515  2063.1876
##   0.021052632  0.12105263  2959.873  0.8898515  1970.7936
##   0.021052632  0.14473684  2826.820  0.8898515  1878.4714
##   0.021052632  0.16842105  2695.042  0.8898515  1786.4300
##   0.021052632  0.19210526  2564.737  0.8898515  1694.6102
##   0.021052632  0.21578947  2436.148  0.8898515  1602.9413
##   0.021052632  0.23947368  2309.571  0.8898515  1511.6138
##   0.021052632  0.26315789  2185.368  0.8898515  1420.8955
##   0.021052632  0.28684211  2063.991  0.8898515  1330.5061
##   0.021052632  0.31052632  1946.006  0.8898515  1241.2944
##   0.021052632  0.33421053  1832.123  0.8898515  1154.2092
##   0.021052632  0.35789474  1723.243  0.8898515  1070.2816
##   0.021052632  0.38157895  1620.506  0.8898515   990.5796
##   0.021052632  0.40526316  1525.339  0.8898515   913.9580
##   0.021052632  0.42894737  1439.849  0.8899217   842.5662
##   0.021052632  0.45263158  1368.422  0.8909875   780.3396
##   0.021052632  0.47631579  1302.833  0.8967428   725.6854
##   0.021052632  0.50000000  1243.323  0.9033984   675.8191
##   0.026315789  0.05000000  3374.525  0.8898515  2255.4147
##   0.026315789  0.07368421  3242.895  0.8898515  2165.7356
##   0.026315789  0.09736842  3112.047  0.8898515  2076.1721
##   0.026315789  0.12105263  2982.083  0.8898515  1986.8887
##   0.026315789  0.14473684  2853.124  0.8898515  1897.6867
##   0.026315789  0.16842105  2725.313  0.8898515  1808.7079
```

```
##   0.026315789  0.19210526  2598.821  0.8898515  1719.9752
##   0.026315789  0.21578947  2473.853  0.8898515  1631.3932
##   0.026315789  0.23947368  2350.657  0.8898515  1543.0097
##   0.026315789  0.26315789  2229.537  0.8898515  1455.2013
##   0.026315789  0.28684211  2110.862  0.8898515  1367.7267
##   0.026315789  0.31052632  1995.092  0.8898515  1280.8531
##   0.026315789  0.33421053  1882.801  0.8898515  1195.6280
##   0.026315789  0.35789474  1774.705  0.8898515  1112.7073
##   0.026315789  0.38157895  1671.705  0.8898515  1033.4398
##   0.026315789  0.40526316  1574.927  0.8898515   957.8318
##   0.026315789  0.42894737  1486.540  0.8898273   885.3985
##   0.026315789  0.45263158  1410.931  0.8899694   819.4690
##   0.026315789  0.47631579  1348.540  0.8920273   763.0910
##   0.026315789  0.50000000  1288.062  0.8982504   711.8656
##   0.031578947  0.05000000  3382.544  0.8898515  2261.1361
##   0.031578947  0.07368421  3254.642  0.8898515  2174.1518
##   0.031578947  0.09736842  3127.464  0.8898515  2087.2828
##   0.031578947  0.12105263  3001.101  0.8898515  2000.6499
##   0.031578947  0.14473684  2875.661  0.8898515  1914.1313
##   0.031578947  0.16842105  2751.269  0.8898515  1827.7764
##   0.031578947  0.19210526  2628.075  0.8898515  1741.6742
##   0.031578947  0.21578947  2506.257  0.8898515  1655.7298
##   0.031578947  0.23947368  2386.027  0.8898515  1569.9418
##   0.031578947  0.26315789  2267.644  0.8898515  1484.5495
##   0.031578947  0.28684211  2151.421  0.8898515  1399.6562
##   0.031578947  0.31052632  2037.743  0.8898515  1315.0903
##   0.031578947  0.33421053  1927.086  0.8898515  1231.6144
##   0.031578947  0.35789474  1820.036  0.8898515  1149.9979
##   0.031578947  0.38157895  1717.330  0.8898515  1071.1036
##   0.031578947  0.40526316  1620.366  0.8898357   996.3020
##   0.031578947  0.42894737  1531.587  0.8897892   924.4171
##   0.031578947  0.45263158  1452.795  0.8897198   856.7212
##   0.031578947  0.47631579  1387.751  0.8900456   796.9711
##   0.031578947  0.50000000  1329.506  0.8935569   744.9194
##   0.036842105  0.05000000  3389.486  0.8898515  2266.0878
##   0.036842105  0.07368421  3264.813  0.8898515  2181.4356
##   0.036842105  0.09736842  3140.816  0.8898515  2096.8980
##   0.036842105  0.12105263  3017.579  0.8898515  2012.5584
##   0.036842105  0.14473684  2895.197  0.8898515  1928.3697
```

```
##   0.036842105  0.16842105  2773.784  0.8898515  1844.2887
##   0.036842105  0.19210526  2653.470  0.8898515  1760.4592
##   0.036842105  0.21578947  2534.414  0.8898515  1676.7996
##   0.036842105  0.23947368  2416.802  0.8898515  1593.2747
##   0.036842105  0.26315789  2300.858  0.8898515  1510.0003
##   0.036842105  0.28684211  2186.854  0.8898515  1427.3045
##   0.036842105  0.31052632  2075.119  0.8898515  1344.8616
##   0.036842105  0.33421053  1966.055  0.8898515  1263.0587
##   0.036842105  0.35789474  1860.161  0.8898515  1182.8132
##   0.036842105  0.38157895  1758.167  0.8898475  1104.7221
##   0.036842105  0.40526316  1662.447  0.8898038  1030.4106
##   0.036842105  0.42894737  1574.058  0.8897151   959.6958
##   0.036842105  0.45263158  1493.483  0.8896209   891.6250
##   0.036842105  0.47631579  1423.327  0.8896517   829.0404
##   0.036842105  0.50000000  1365.501  0.8906481   774.8292
##   0.042105263  0.05000000  3395.302  0.8898515  2270.2531
##   0.042105263  0.07368421  3273.336  0.8898515  2187.5611
##   0.042105263  0.09736842  3152.008  0.8898515  2104.9837
##   0.042105263  0.12105263  3031.395  0.8898515  2022.5759
##   0.042105263  0.14473684  2911.582  0.8898515  1940.3471
##   0.042105263  0.16842105  2792.673  0.8898515  1858.1860
##   0.042105263  0.19210526  2674.788  0.8898515  1776.2633
##   0.042105263  0.21578947  2558.067  0.8898515  1694.5285
##   0.042105263  0.23947368  2442.678  0.8898515  1612.9227
##   0.042105263  0.26315789  2328.820  0.8898515  1531.4683
##   0.042105263  0.28684211  2216.733  0.8898515  1450.5773
##   0.042105263  0.31052632  2106.707  0.8898515  1369.9999
##   0.042105263  0.33421053  1999.095  0.8898515  1289.7836
##   0.042105263  0.35789474  1894.327  0.8898515  1210.8355
##   0.042105263  0.38157895  1793.922  0.8898229  1133.8521
##   0.042105263  0.40526316  1699.927  0.8897423  1059.9748
##   0.042105263  0.42894737  1611.838  0.8896301   989.9434
##   0.042105263  0.45263158  1529.874  0.8895238   922.1291
##   0.042105263  0.47631579  1456.665  0.8894707   858.5587
##   0.042105263  0.50000000  1395.807  0.8897208   801.8025
##   0.047368421  0.05000000  3399.751  0.8898515  2273.4624
##   0.047368421  0.07368421  3279.855  0.8898515  2192.2811
##   0.047368421  0.09736842  3160.570  0.8898515  2111.2126
##   0.047368421  0.12105263  3041.964  0.8898515  2030.2972
```

```
##   0.047368421  0.14473684  2924.121  0.8898515  1949.5792
##   0.047368421  0.16842105  2807.134  0.8898515  1868.9074
##   0.047368421  0.19210526  2691.115  0.8898515  1788.4527
##   0.047368421  0.21578947  2576.194  0.8898515  1708.1959
##   0.047368421  0.23947368  2462.524  0.8898515  1628.0705
##   0.047368421  0.26315789  2350.290  0.8898515  1548.0623
##   0.047368421  0.28684211  2239.710  0.8898515  1468.5258
##   0.047368421  0.31052632  2131.048  0.8898515  1389.3984
##   0.047368421  0.33421053  2024.623  0.8898515  1310.5136
##   0.047368421  0.35789474  1921.145  0.8898398  1232.6790
##   0.047368421  0.38157895  1823.140  0.8897740  1156.8752
##   0.047368421  0.40526316  1730.322  0.8896629  1083.5028
##   0.047368421  0.42894737  1642.250  0.8895415  1013.6064
##   0.047368421  0.45263158  1559.584  0.8894242   946.4384
##   0.047368421  0.47631579  1485.105  0.8893495   882.9539
##   0.047368421  0.50000000  1421.068  0.8893970   824.3211
##   0.052631579  0.05000000  3401.806  0.8898515  2274.8974
##   0.052631579  0.07368421  3282.869  0.8898515  2194.3918
##   0.052631579  0.09736842  3164.531  0.8898515  2113.9985
##   0.052631579  0.12105263  3046.859  0.8898515  2033.7548
##   0.052631579  0.14473684  2929.934  0.8898515  1953.7118
##   0.052631579  0.16842105  2813.846  0.8898515  1873.7079
##   0.052631579  0.19210526  2698.704  0.8898515  1793.9143
##   0.052631579  0.21578947  2584.632  0.8898515  1714.3237
##   0.052631579  0.23947368  2471.780  0.8898515  1634.8629
##   0.052631579  0.26315789  2360.324  0.8898515  1555.5084
##   0.052631579  0.28684211  2250.475  0.8898515  1476.5846
##   0.052631579  0.31052632  2142.484  0.8898515  1398.0966
##   0.052631579  0.33421053  2036.657  0.8898515  1319.8315
##   0.052631579  0.35789474  1934.836  0.8898126  1242.7771
##   0.052631579  0.38157895  1838.399  0.8897047  1167.6200
##   0.052631579  0.40526316  1746.197  0.8895794  1094.5418
##   0.052631579  0.42894737  1658.189  0.8894510  1024.6741
##   0.052631579  0.45263158  1575.263  0.8893234   957.8045
##   0.052631579  0.47631579  1500.330  0.8892266   894.3013
##   0.052631579  0.50000000  1434.879  0.8892291   834.9677
##   0.057894737  0.05000000  3402.492  0.8898515  2275.3626
##   0.057894737  0.07368421  3283.877  0.8898515  2195.0755
##   0.057894737  0.09736842  3165.856  0.8898515  2114.9008
```

```
##   0.057894737  0.12105263  3048.497  0.8898515  2034.8733
##   0.057894737  0.14473684  2931.879  0.8898515  1955.0480
##   0.057894737  0.16842105  2816.094  0.8898515  1875.2603
##   0.057894737  0.19210526  2701.246  0.8898515  1795.6791
##   0.057894737  0.21578947  2587.461  0.8898515  1716.3032
##   0.057894737  0.23947368  2474.885  0.8898515  1637.0573
##   0.057894737  0.26315789  2363.691  0.8898515  1557.9148
##   0.057894737  0.28684211  2254.088  0.8898515  1479.1874
##   0.057894737  0.31052632  2146.325  0.8898515  1400.9067
##   0.057894737  0.33421053  2041.024  0.8898388  1322.9226
##   0.057894737  0.35789474  1941.040  0.8897521  1246.3448
##   0.057894737  0.38157895  1845.228  0.8896274  1171.4567
##   0.057894737  0.40526316  1753.104  0.8894948  1098.4566
##   0.057894737  0.42894737  1665.111  0.8893605  1028.5873
##   0.057894737  0.45263158  1582.159  0.8892181   961.7905
##   0.057894737  0.47631579  1506.956  0.8891040   898.2001
##   0.057894737  0.50000000  1441.113  0.8890914   838.6644
##   0.063157895  0.05000000  3402.971  0.8898515  2275.6841
##   0.063157895  0.07368421  3284.579  0.8898515  2195.5483
##   0.063157895  0.09736842  3166.779  0.8898515  2115.5249
##   0.063157895  0.12105263  3049.639  0.8898515  2035.6464
##   0.063157895  0.14473684  2933.237  0.8898515  1955.9718
##   0.063157895  0.16842105  2817.663  0.8898515  1876.3331
##   0.063157895  0.19210526  2703.022  0.8898515  1796.8989
##   0.063157895  0.21578947  2589.437  0.8898515  1717.6720
##   0.063157895  0.23947368  2477.055  0.8898515  1638.5742
##   0.063157895  0.26315789  2366.047  0.8898515  1559.5773
##   0.063157895  0.28684211  2256.618  0.8898515  1480.9827
##   0.063157895  0.31052632  2149.015  0.8898515  1402.8433
##   0.063157895  0.33421053  2045.122  0.8898044  1325.2768
##   0.063157895  0.35789474  1946.077  0.8896851  1248.9417
##   0.063157895  0.38157895  1850.571  0.8895489  1174.2054
##   0.063157895  0.40526316  1758.503  0.8894099  1101.2686
##   0.063157895  0.42894737  1670.520  0.8892706  1031.3955
##   0.063157895  0.45263158  1587.719  0.8891130   964.6828
##   0.063157895  0.47631579  1512.182  0.8889830   900.9719
##   0.063157895  0.50000000  1446.104  0.8889188   841.3198
##   0.068421053  0.05000000  3403.332  0.8898515  2275.9264
##   0.068421053  0.07368421  3285.108  0.8898515  2195.9045
```

```
##   0.068421053  0.09736842  3167.476  0.8898515  2115.9950
##   0.068421053  0.12105263  3050.501  0.8898515  2036.2288
##   0.068421053  0.14473684  2934.262  0.8898515  1956.6676
##   0.068421053  0.16842105  2818.848  0.8898515  1877.1410
##   0.068421053  0.19210526  2704.363  0.8898515  1797.8175
##   0.068421053  0.21578947  2590.932  0.8898515  1718.7029
##   0.068421053  0.23947368  2478.697  0.8898515  1639.7166
##   0.068421053  0.26315789  2367.830  0.8898515  1560.8288
##   0.068421053  0.28684211  2258.534  0.8898515  1482.3330
##   0.068421053  0.31052632  2151.225  0.8898443  1404.3427
##   0.068421053  0.33421053  2048.980  0.8897478  1327.1816
##   0.068421053  0.35789474  1950.453  0.8896129  1251.0120
##   0.068421053  0.38157895  1855.020  0.8894698  1176.3594
##   0.068421053  0.40526316  1762.999  0.8893253  1103.4596
##   0.068421053  0.42894737  1675.036  0.8891811  1033.5911
##   0.068421053  0.45263158  1592.405  0.8890079   966.9435
##   0.068421053  0.47631579  1516.594  0.8888559   903.1494
##   0.068421053  0.50000000  1450.585  0.8886571   843.6642
##   0.073684211  0.05000000  3403.551  0.8898515  2276.0713
##   0.073684211  0.07368421  3285.432  0.8898515  2196.1174
##   0.073684211  0.09736842  3167.901  0.8898515  2116.2759
##   0.073684211  0.12105263  3051.028  0.8898515  2036.5769
##   0.073684211  0.14473684  2934.889  0.8898515  1957.0835
##   0.073684211  0.16842105  2819.574  0.8898515  1877.6241
##   0.073684211  0.19210526  2705.187  0.8898515  1798.3667
##   0.073684211  0.21578947  2591.850  0.8898515  1719.3191
##   0.073684211  0.23947368  2479.707  0.8898515  1640.3996
##   0.073684211  0.26315789  2368.928  0.8898515  1561.5770
##   0.073684211  0.28684211  2259.717  0.8898515  1483.1406
##   0.073684211  0.31052632  2153.506  0.8898124  1405.4413
##   0.073684211  0.33421053  2052.116  0.8896853  1328.4764
##   0.073684211  0.35789474  1953.857  0.8895389  1252.4018
##   0.073684211  0.38157895  1858.473  0.8893896  1177.8035
##   0.073684211  0.40526316  1766.482  0.8892404  1104.9112
##   0.073684211  0.42894737  1678.615  0.8890852  1035.0785
##   0.073684211  0.45263158  1596.103  0.8889019   968.4705
##   0.073684211  0.47631579  1520.123  0.8886872   904.6952
##   0.073684211  0.50000000  1454.340  0.8883585   845.5632
##   0.078947368  0.05000000  3403.646  0.8898515  2276.1295
```

```
##   0.078947368  0.07368421  3285.570  0.8898515  2196.2029
##   0.078947368  0.09736842  3168.085  0.8898515  2116.3888
##   0.078947368  0.12105263  3051.256  0.8898515  2036.7169
##   0.078947368  0.14473684  2935.161  0.8898515  1957.2505
##   0.078947368  0.16842105  2819.890  0.8898515  1877.8179
##   0.078947368  0.19210526  2705.546  0.8898515  1798.5875
##   0.078947368  0.21578947  2592.252  0.8898515  1719.5676
##   0.078947368  0.23947368  2480.151  0.8898515  1640.6756
##   0.078947368  0.26315789  2369.413  0.8898515  1561.8792
##   0.078947368  0.28684211  2260.241  0.8898515  1483.4657
##   0.078947368  0.31052632  2155.572  0.8897598  1406.1001
##   0.078947368  0.33421053  2054.623  0.8896170  1329.2343
##   0.078947368  0.35789474  1956.398  0.8894636  1253.1952
##   0.078947368  0.38157895  1861.039  0.8893089  1178.6158
##   0.078947368  0.40526316  1769.063  0.8891555  1105.7265
##   0.078947368  0.42894737  1681.407  0.8889833  1035.9345
##   0.078947368  0.45263158  1598.845  0.8887832   969.3375
##   0.078947368  0.47631579  1522.999  0.8884270   905.8746
##   0.078947368  0.50000000  1457.330  0.8880913   847.0299
##   0.084210526  0.05000000  3403.557  0.8898515  2276.0626
##   0.084210526  0.07368421  3285.441  0.8898515  2196.1046
##   0.084210526  0.09736842  3167.915  0.8898515  2116.2589
##   0.084210526  0.12105263  3051.048  0.8898515  2036.5562
##   0.084210526  0.14473684  2934.915  0.8898515  1957.0582
##   0.084210526  0.16842105  2819.607  0.8898515  1877.5941
##   0.084210526  0.19210526  2705.229  0.8898515  1798.3337
##   0.084210526  0.21578947  2591.901  0.8898515  1719.2835
##   0.084210526  0.23947368  2479.769  0.8898515  1640.3611
##   0.084210526  0.26315789  2369.003  0.8898515  1561.5334
##   0.084210526  0.28684211  2260.257  0.8898352  1483.1962
##   0.084210526  0.31052632  2156.640  0.8897036  1406.0090
##   0.084210526  0.33421053  2055.957  0.8895475  1329.1703
##   0.084210526  0.35789474  1957.720  0.8893879  1253.1207
##   0.084210526  0.38157895  1862.352  0.8892284  1178.5161
##   0.084210526  0.40526316  1770.370  0.8890715  1105.6209
##   0.084210526  0.42894737  1682.943  0.8888811  1035.8570
##   0.084210526  0.45263158  1600.344  0.8886231   969.3005
##   0.084210526  0.47631579  1524.785  0.8881432   906.2959
##   0.084210526  0.50000000  1459.171  0.8878326   847.6247
```

```
##   0.089473684  0.05000000  3403.333  0.8898515  2275.9014
##   0.089473684  0.07368421  3285.113  0.8898515  2195.8674
##   0.089473684  0.09736842  3167.485  0.8898515  2115.9462
##   0.089473684  0.12105263  3050.516  0.8898515  2036.1692
##   0.089473684  0.14473684  2934.285  0.8898515  1956.5952
##   0.089473684  0.16842105  2818.881  0.8898515  1877.0557
##   0.089473684  0.19210526  2704.409  0.8898515  1797.7217
##   0.089473684  0.21578947  2590.992  0.8898515  1718.5982
##   0.089473684  0.23947368  2478.774  0.8898515  1639.6017
##   0.089473684  0.26315789  2367.926  0.8898515  1560.7001
##   0.089473684  0.28684211  2260.393  0.8897889  1482.5590
##   0.089473684  0.31052632  2157.118  0.8896408  1405.3776
##   0.089473684  0.33421053  2056.380  0.8894775  1328.4835
##   0.089473684  0.35789474  1958.097  0.8893124  1252.3793
##   0.089473684  0.38157895  1862.695  0.8891489  1177.7190
##   0.089473684  0.40526316  1770.780  0.8889807  1104.8284
##   0.089473684  0.42894737  1683.416  0.8887686  1035.0752
##   0.089473684  0.45263158  1600.944  0.8883775   968.7670
##   0.089473684  0.47631579  1525.645  0.8878934   906.0939
##   0.089473684  0.50000000  1460.139  0.8875717   847.4810
##   0.094736842  0.05000000  3403.012  0.8898515  2275.6750
##   0.094736842  0.07368421  3284.643  0.8898515  2195.5343
##   0.094736842  0.09736842  3166.868  0.8898515  2115.5067
##   0.094736842  0.12105263  3049.754  0.8898515  2035.6248
##   0.094736842  0.14473684  2933.381  0.8898515  1955.9445
##   0.094736842  0.16842105  2817.838  0.8898515  1876.2993
##   0.094736842  0.19210526  2703.231  0.8898515  1796.8617
##   0.094736842  0.21578947  2589.683  0.8898515  1717.6346
##   0.094736842  0.23947368  2477.340  0.8898515  1638.5336
##   0.094736842  0.26315789  2366.373  0.8898515  1559.5277
##   0.094736842  0.28684211  2259.995  0.8897391  1481.5586
##   0.094736842  0.31052632  2156.958  0.8895762  1404.3314
##   0.094736842  0.33421053  2056.136  0.8894065  1327.3558
##   0.094736842  0.35789474  1957.784  0.8892366  1251.1714
##   0.094736842  0.38157895  1862.327  0.8890695  1176.4404
##   0.094736842  0.40526316  1770.579  0.8888837  1103.5630
##   0.094736842  0.42894737  1683.160  0.8886271  1033.8142
##   0.094736842  0.45263158  1601.003  0.8881059   967.9491
##   0.094736842  0.47631579  1525.817  0.8876415   905.3897
```

```
##   0.094736842  0.50000000  1460.623  0.8872825   846.8614
##   0.100000000  0.05000000  3402.626  0.8898515  2275.4080
##   0.100000000  0.07368421  3284.077  0.8898515  2195.1414
##   0.100000000  0.09736842  3166.124  0.8898515  2114.9882
##   0.100000000  0.12105263  3048.836  0.8898515  2034.9823
##   0.100000000  0.14473684  2932.291  0.8898515  1955.1769
##   0.100000000  0.16842105  2816.580  0.8898515  1875.4075
##   0.100000000  0.19210526  2701.809  0.8898515  1795.8477
##   0.100000000  0.21578947  2588.102  0.8898515  1716.4979
##   0.100000000  0.23947368  2475.606  0.8898515  1637.2735
##   0.100000000  0.26315789  2364.989  0.8898318  1558.2464
##   0.100000000  0.28684211  2259.406  0.8896825  1480.3550
##   0.100000000  0.31052632  2156.352  0.8895099  1403.0372
##   0.100000000  0.33421053  2055.428  0.8893344  1325.9529
##   0.100000000  0.35789474  1956.989  0.8891603  1249.6785
##   0.100000000  0.38157895  1861.465  0.8889901  1174.8678
##   0.100000000  0.40526316  1769.837  0.8887767  1101.9933
##   0.100000000  0.42894737  1682.388  0.8883970  1032.3771
##   0.100000000  0.45263158  1600.586  0.8878648   966.9248
##   0.100000000  0.47631579  1525.530  0.8873860   904.3262
##   0.100000000  0.50000000  1460.880  0.8869610   845.9827
##
## RMSE was used to select the optimal model using the smallest value.
## The final values used for the model were fraction = 0.5 and lambda = 0.
```

```
# Test Error for Elastic net model

enet.pred = predict(enet.model, data=test)
MSE.enet.test <- mean((enet.pred - test$Apps)^2)
#RMSE.enet.test = sqrt(MSE.enet.test)

print(paste("The Elastic Net Model test MSE is ", round(MSE.enet.test,0)))
```

```
## [1] "The Elastic Net Model test MSE is  28056722"
```

Problem 6 (e)

e. Comment on the results obtained. How accurately can we predict the number of college applications received? Is there much difference among the test errors resulting from these three approaches? [5 points]

Answer 6 (e)

Comparison of Model Statistics

	OLS	Ridge	Elastic Net
MSE	1567324	1567246	28056722

We see that the test errors for OLS and Ridge are pretty close, however for ENET, the test error is a bit high. If we look at the R^2, they are almost the same (> 0.9). So, based on the **MSE** data , we can say that the Ridge regression model has greater prediction accuracy as it has the lowest MSE.