STA 380: Intro to ML - Take Home Exam

Parthiv Borgohain

2022-07-29

Note: For this take home exam, I have used **ISLR Edition 1.**

Chapter 2 | Problem 10

```
##
## Attaching package: 'dplyr'

## The following object is masked from 'package:MASS':
##
##     select

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union

## starting httpd help server ... done
```

We first take a look at the Boston dataset and its rows and columns.

```
##      crim zn indus chas   nox    rm  age    dis rad tax ptratio black lstat
## 1 0.00632 18  2.31    0 0.538 6.575 65.2 4.0900   1 296    15.3 396.9  4.98
## 2 0.02731  0  7.07    0 0.469 6.421 78.9 4.9671   2 242    17.8 396.9  9.14
##   medv
## 1 24.0
## 2 21.6
```

Part (A)

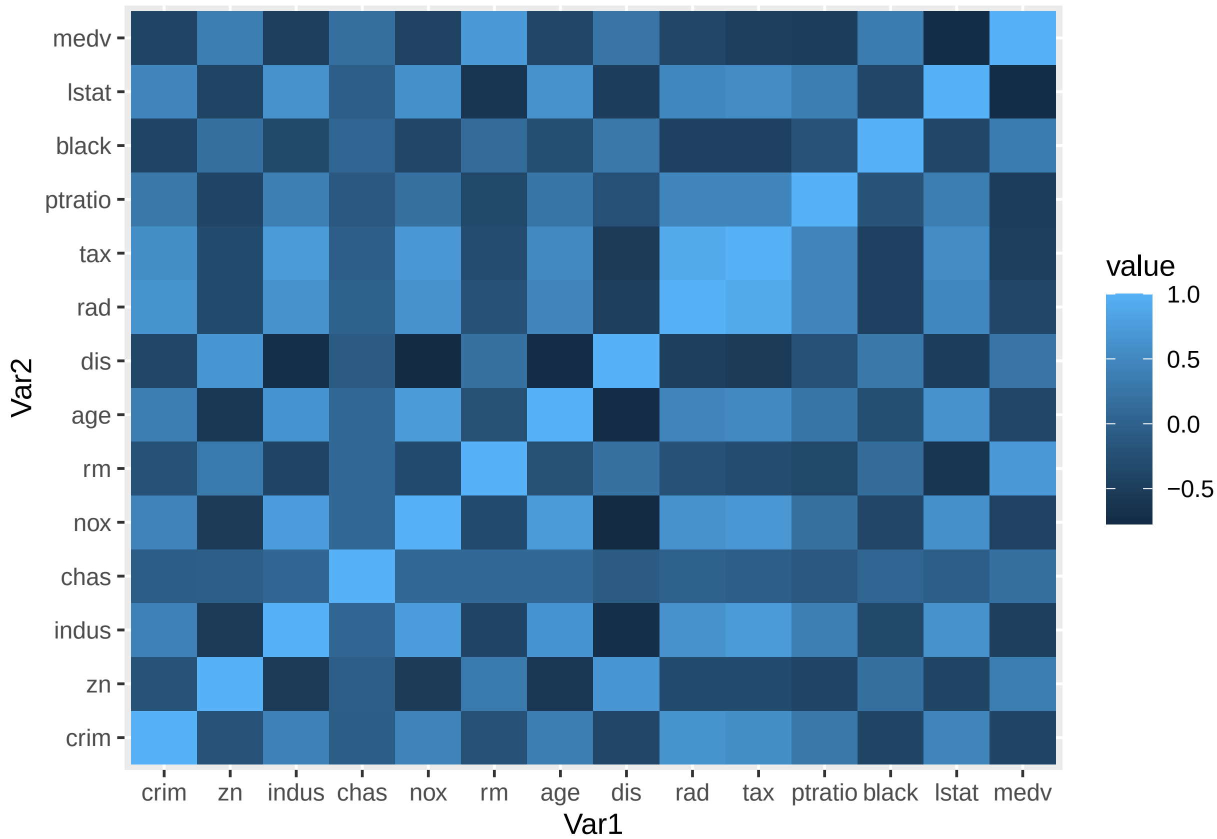
```
## [1] "No. of Rows in Boston DF is 506"

## [1] "No. of Columns in Boston DF is 14"

## [1] "Every row represents a suburb of Boston and column represents different variables associated wi
```

Part (B)

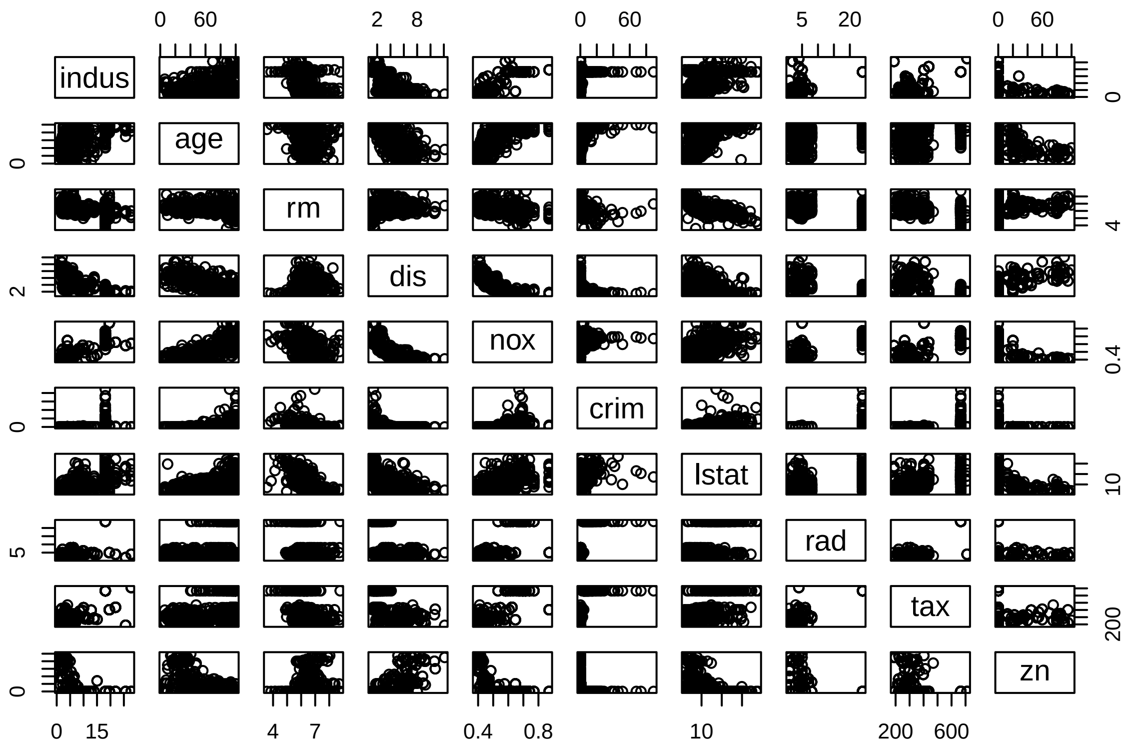
We try to create a heatmap for the correlation matrix to get a preliminary idea of the correlation between variables

```
##        crim    zn indus  chas   nox    rm   age   dis   rad   tax ptratio black
## crim   1.00 -0.20  0.41 -0.06  0.42 -0.22  0.35 -0.38  0.63  0.58    0.29 -0.39
## zn    -0.20  1.00 -0.53 -0.04 -0.52  0.31 -0.57  0.66 -0.31 -0.31   -0.39  0.18
## indus  0.41 -0.53  1.00  0.06  0.76 -0.39  0.64 -0.71  0.60  0.72    0.38 -0.36
## chas  -0.06 -0.04  0.06  1.00  0.09  0.09  0.09 -0.10 -0.01 -0.04   -0.12  0.05
## nox    0.42 -0.52  0.76  0.09  1.00 -0.30  0.73 -0.77  0.61  0.67    0.19 -0.38
## rm    -0.22  0.31 -0.39  0.09 -0.30  1.00 -0.24  0.21 -0.21 -0.29   -0.36  0.13
##       lstat  medv
## crim   0.46 -0.39
## zn    -0.41  0.36
## indus  0.60 -0.48
## chas  -0.05  0.18
## nox    0.59 -0.43
## rm    -0.61  0.70
```

```
##    Var1 Var2 value
## 1  crim crim  1.00
## 2    zn crim -0.20
## 3 indus crim  0.41
## 4  chas crim -0.06
## 5   nox crim  0.42
## 6    rm crim -0.22
```



Using the heatmap, we will now plot some of the correlated variables pairwise



So, we can note the following observations from the pairwise plots:

- There seems to be a negative relationship between zn and indus. This is probably explained by the fact that residential areas are usually built away from industries
- There seems to be a negative relationship between zn and nox. This is probably explained by the fact that residential areas are usually built away from areas having pollutants like NOX.
- There seems to be a positive relationship between tax and rad.
- There seems to be a strong positive relationship between indus and nox. This seems to make sense as industrial areas will have more pollutants like NOX.

Part (C)

Now we will take a look at the correlation of crime with other predictor variables

```
##                [,1]
## zn      -0.20046922
## indus    0.40658341
## chas    -0.05589158
## nox      0.42097171
## rm      -0.21924670
## age      0.35273425
## dis     -0.37967009
```

```
## rad      0.62550515
## tax      0.58276431
## ptratio  0.28994558
## black   -0.38506394
## lstat    0.45562148
## medv    -0.38830461
```

Below are the findings:

- The variable crim does not appear to have an extremely strong correlation with any other variable in the predictor set
- The variables tax and rad are the variables with highest positive correlation with crim (0.63 and 0.58 respectively)
- The variables medv and dis are the variables with most negative correlation with crim (-0.39 and -0.38 respectively)

Part (D)

We will check the range of each of these 3 attributes

```
##        crim tax ptratio
##     0.00632 187    12.6
##    88.97620 711    22.0
## x 88.96988 524     9.4
```

So, Tax has the highest range.

Now let us look for the top 10 suburbs with highest crime rate per capita, tax and pupil teacher ratio each:

```
##         crim medv
## 381 88.9762 10.4
## 419 73.5341  8.8
## 406 67.9208  5.0
## 411 51.1358 15.0
## 415 45.7461  7.0
## 405 41.5292  8.5
## 399 38.3518  5.0
## 428 37.6619 10.9
## 414 28.6558 16.3
## 418 25.9406 10.4
```

```
##     tax medv
## 489 711 15.2
## 490 711  7.0
## 491 711  8.1
## 492 711 13.6
## 493 711 20.1
## 357 666 17.8
## 358 666 21.7
## 359 666 22.7
## 360 666 22.6
## 361 666 25.0
```

```
##     ptratio medv
## 355    22.0 18.2
## 356    22.0 20.6
## 128    21.2 16.2
## 129    21.2 18.0
## 130    21.2 14.3
## 131    21.2 19.2
## 132    21.2 19.6
## 133    21.2 23.0
## 134    21.2 18.4
## 135    21.2 15.6
```

Part (E)

```
## [1] "No. of suburbs which bound Charles river :  35"
```

Part (F)

```
## [1] "Median PT Ratio amongst Towns is 19.05"
```

Part (G)

```
##        crim zn indus chas   nox    rm age    dis rad tax ptratio  black lstat
## 399 38.3518  0  18.1    0 0.693 5.453 100 1.4896  24 666    20.2 396.90 30.59
## 406 67.9208  0  18.1    0 0.693 5.683 100 1.4254  24 666    20.2 384.97 22.98
##     medv
## 399    5
## 406    5
```

So, clearly there are two suburbs (**Suburb #399** and **#406**) with the **lowest median value** of owner-occupied homes.

Now, To compare the other predictors with the rest of the dataset, we can use percentiles.

```
##    Variables Suburb_399 Suburb_406
## 1       crim 0.98814229 0.99604743
## 2         zn 0.73517787 0.73517787
## 3      indus 0.88735178 0.88735178
## 4       chas 0.93083004 0.93083004
## 5        nox 0.85770751 0.85770751
## 6         rm 0.07707510 0.13636364
## 7        age 1.00000000 1.00000000
## 8        dis 0.05731225 0.04150198
## 9        rad 1.00000000 1.00000000
## 10       tax 0.99011858 0.99011858
## 11   ptratio 0.88932806 0.88932806
## 12     black 1.00000000 0.34980237
## 13     lstat 0.97826087 0.89920949
```

So, Comparison of these variables with their overall ranges:

- Crim: ~99%ile for both the suburbs

- Zn : ~75%ile for both the suburbs
- indus: ~90%ile for both the suburbs
- Chas: ~93%ile for both the suburbs
- nox: ~86%ile for both the suburbs
- rm: ~7%ile for Suburb #399 and ~13%ile for Suburb #406
- dis: ~5%ile for Suburb #399 and ~4%ile for Suburb #406
- rad: ~100%ile for both the suburbs
- tax: ~99%ile for both the suburbs
- ptratio: ~89%ile for both the suburbs
- black: ~100%ile for Suburb #399 and ~35%ile for Suburb #406
- lstat: ~98%ile for Suburb #399 and ~90%ile for Suburb #406

Part (H)

We will now filter and get the counts of both these types of suburbs

```
## [1] "No. of Suburbs with > 7 rooms per dwelling = 64"
```

```
## [1] "No. of Suburbs with > 8 rooms per dwelling = 13"
```

Analyzing further for suburbs with >8 rooms per dwelling:

Calculating the percentiles for these mean values:

```
##    colnamesdf          A          B
## 1        crim 0.62648221 0.58498024
## 2          zn 0.75494071 0.73517787
## 3       indus 0.39920949 0.35770751
## 4        chas 0.93083004 0.93083004
## 5         nox 0.53754941 0.41501976
## 6          rm 0.99011858 0.98814229
## 7         age 0.44861660 0.50988142
## 8         dis 0.53952569 0.46442688
## 9         rad 0.69169960 0.69169960
## 10        tax 0.47430830 0.45454545
## 11    ptratio 0.17786561 0.28458498
## 12      black 0.35573123 0.38537549
## 13      lstat 0.07509881 0.07114625
## 14       medv 0.95454545 0.96442688
```

So, the key takeaways are:

- Their median values are extremely high (about 95% percentile),
- the proportion of residential land zoned for lots over 25,000 sq.ft is also pretty high at about 75% percentile
- their index of accessibility to radial highways is also pretty high at about 69% percentile
- their lstat values are very low with about 7% percentile for both categories.

Chapter 3 | Problem 15

Part (A)

First we will combine all columns into a single vector and remove crim variable from it.

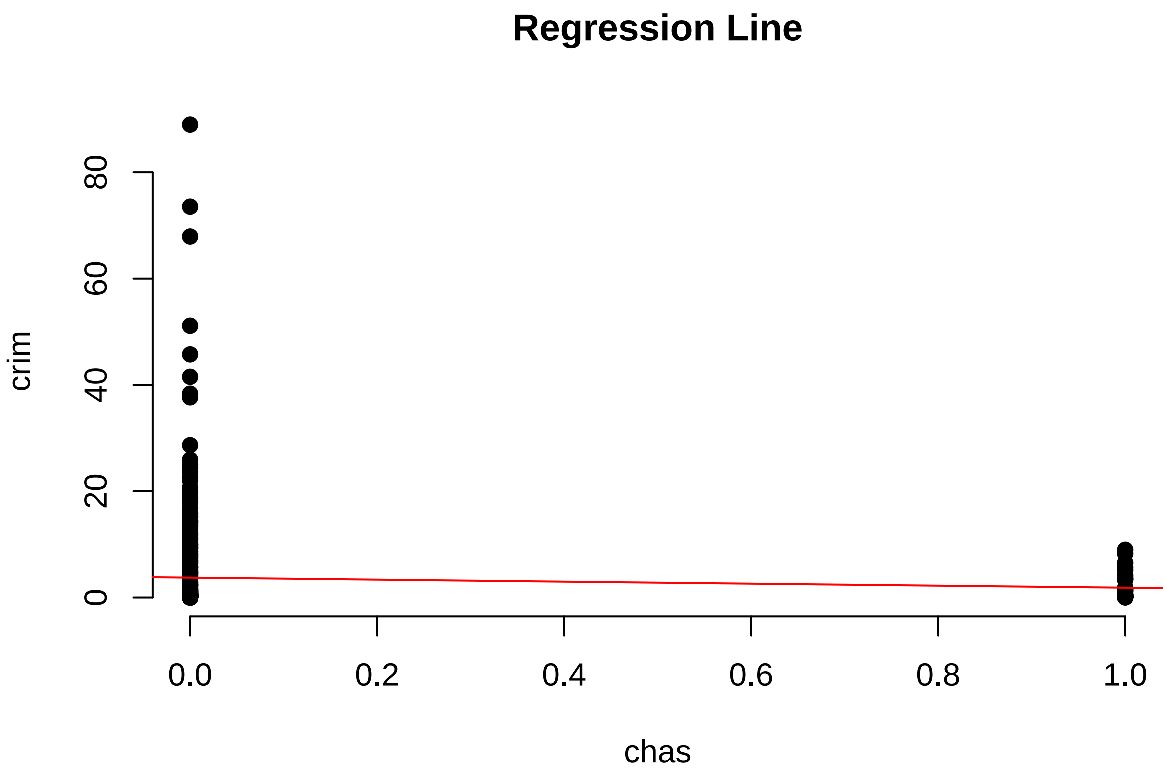
We fit each variable with a linear regression model.

Summarizing all the models we have run into one table, we get -

```
##         coefficients_list  pvalue_list rsquare_list
## zn            -0.07393498 5.506472e-06  0.040187908
## indus          0.50977633 1.450349e-21  0.165310070
## chas          -1.89277655 2.094345e-01  0.003123869
## nox           31.24853120 3.751739e-23  0.177217182
## rm            -2.68405122 6.346703e-07  0.048069117
## age            0.10778623 2.854869e-16  0.124421452
## dis           -1.55090168 8.519949e-19  0.144149375
## rad            0.61791093 2.693844e-56  0.391256687
## tax            0.02974225 2.357127e-47  0.339614243
## ptratio        1.15198279 2.942922e-11  0.084068439
## black         -0.03627964 2.487274e-19  0.148274239
## lstat          0.54880478 2.654277e-27  0.207590933
## medv          -0.36315992 1.173987e-19  0.150780469
```

Clearly, only the variable **'chas'** has a **p value of greater than 0.05**. So, 'chas' does **not have a statistically significant relationship** with the variable **'crim'**

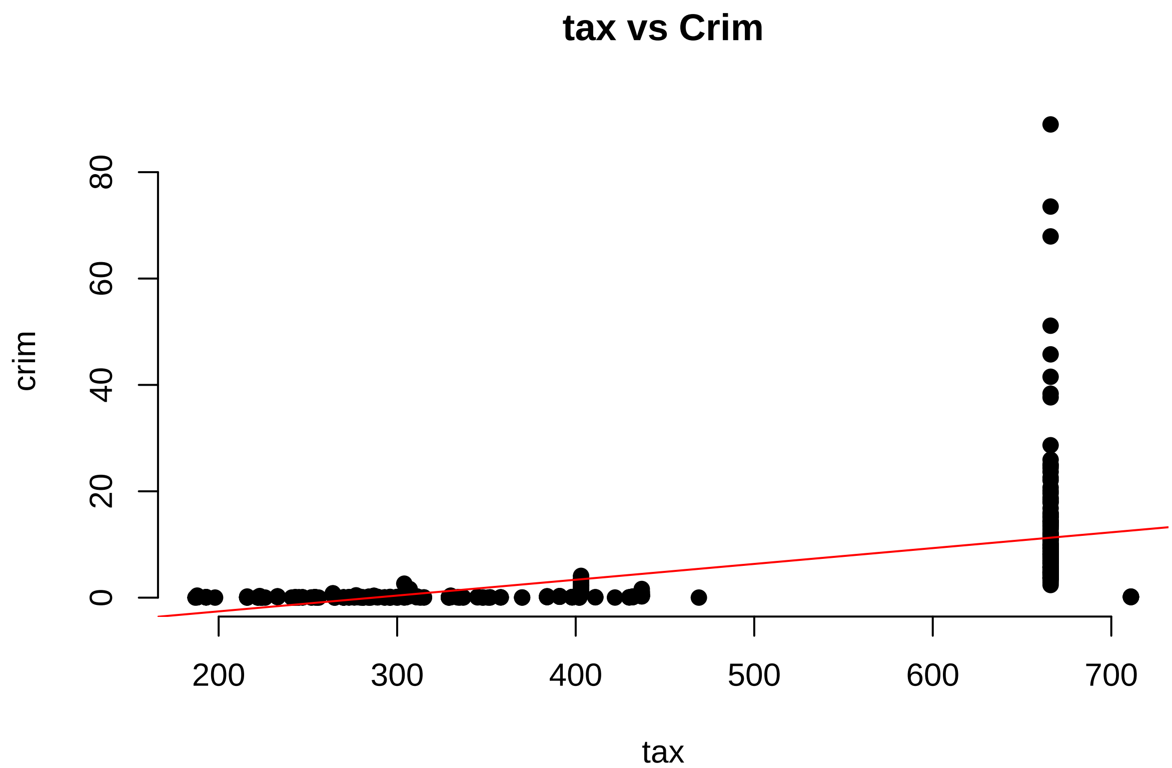
Exploring the relationship between *'chas'* and *'crim'* by a plot, we get-

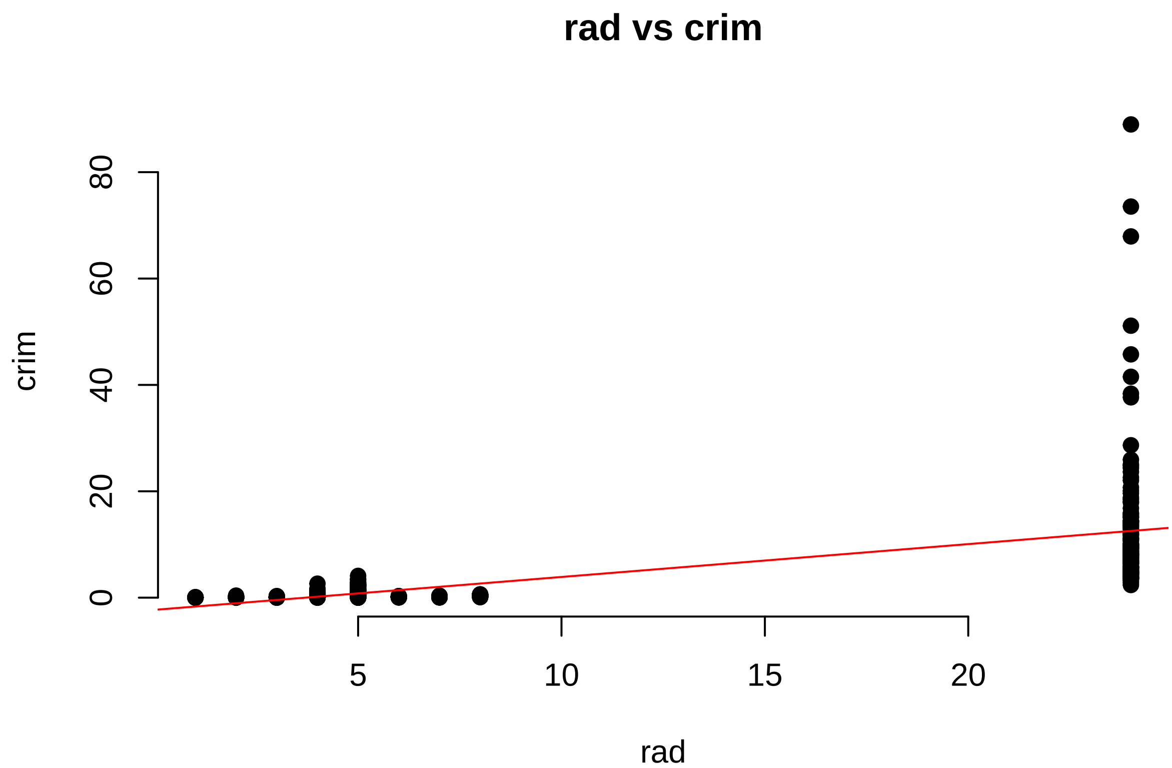


Clearly, the regression line is **almost horizontal**. So, the **coefficient is almost 0** and **not statistically significant**.

On the other hand, we saw **high R Square values** for the variables **tax and rad** (**0.34 and 0.39** respectively).

Exploring this further, we get for variables tax and crim-


tax vs Crim
crim
0
20
40
60
80
200
300
400
500
600
700
tax



So, by looking at the scatterplots, there seems to be some sort of a **statistically significant relationship** between **rad and crim**;**tax and crim**.

Part (B)

Fitting a multiple linear regression model using all of the predictors-

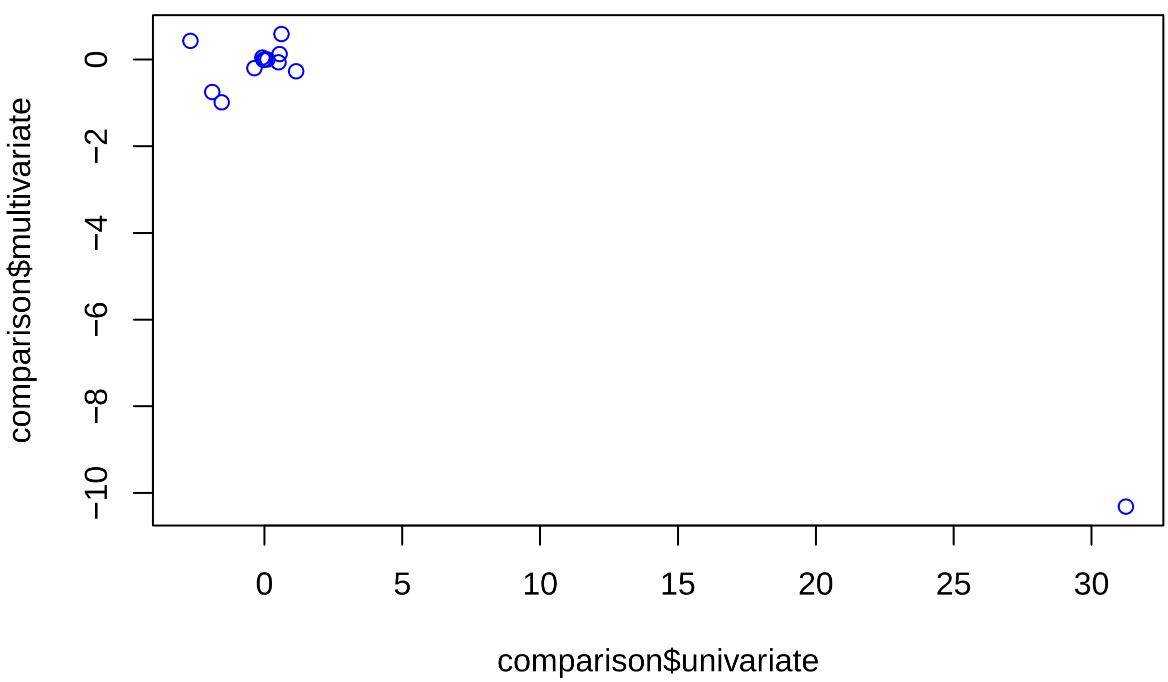
```
##                  Estimate  Std. Error      t value     Pr(>|t|)
## (Intercept)  17.033227523 7.234903031  2.35431317 1.894909e-02
## zn            0.044855215 0.018734071  2.39431224 1.702489e-02
## indus        -0.063854824 0.083407241 -0.76557890 4.442940e-01
## chas         -0.749133611 1.180146772 -0.63478004 5.258670e-01
## nox         -10.313534912 5.275536315 -1.95497373 5.115200e-02
## rm            0.430130506 0.612830309  0.70187538 4.830888e-01
## age           0.001451643 0.017925128  0.08098372 9.354878e-01
## dis          -0.987175726 0.281817266 -3.50289299 5.022039e-04
## rad           0.588208591 0.088049274  6.68044796 6.460451e-11
## tax          -0.003780016 0.005155587 -0.73318838 4.637927e-01
## ptratio      -0.271080558 0.186450494 -1.45390099 1.466113e-01
## black        -0.007537505 0.003673322 -2.05195893 4.070233e-02
## lstat         0.126211376 0.075724837  1.66671043 9.620842e-02
## medv         -0.198886821 0.060515990 -3.28651687 1.086810e-03
```

Keeping only those variables whose P value < 0.05, we get-

```
##                  Estimate  Std. Error   t value     Pr(>|t|)
## (Intercept) 17.033227523 7.234903031  2.354313 1.894909e-02
## zn           0.044855215 0.018734071  2.394312 1.702489e-02
## dis         -0.987175726 0.281817266 -3.502893 5.022039e-04
## rad          0.588208591 0.088049274  6.680448 6.460451e-11
## black       -0.007537505 0.003673322 -2.051959 4.070233e-02
## medv        -0.198886821 0.060515990 -3.286517 1.086810e-03
```

As these 5 variables have **p value $<$ 5%**, they are **statistically significant**.

Part (C)



Clearly, there is **one outlier** here. The coefficient for **Nox** in the **univariate model** is about **32** while in the **multivariate model** it is about **-10**.

In the univariate model, the effect of Nox on crim was being analyzed without considering other variables. But in the multivariate regression model, as other variables are also considered, **nox probably already was highly correlated with some other variable** resulting in a very negative coefficient in the multivariate model.

So, in the **multivariate model, nox** has a **highly negative impact on crim**.

Part (D)

So we will be **fitting a cubic model** for each of the predictor variables.

```
## [1] "zn"
##
## Call:
## lm(formula = crim ~ x + x2 + x3)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -4.821 -4.614 -1.294  0.473 84.130
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  4.846e+00  4.330e-01  11.192  < 2e-16 ***
## x           -3.322e-01  1.098e-01  -3.025  0.00261 **
## x2           6.483e-03  3.861e-03   1.679  0.09375 .
## x3          -3.776e-05  3.139e-05  -1.203  0.22954
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 8.372 on 502 degrees of freedom
## Multiple R-squared:  0.05824,    Adjusted R-squared:  0.05261
## F-statistic: 10.35 on 3 and 502 DF,  p-value: 1.281e-06
##
## [1] "indus"
##
## Call:
## lm(formula = crim ~ x + x2 + x3)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -8.278 -2.514  0.054  0.764 79.713
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  3.6625683  1.5739833   2.327   0.0204 *
## x           -1.9652129  0.4819901  -4.077 5.30e-05 ***
## x2           0.2519373  0.0393221   6.407 3.42e-10 ***
## x3          -0.0069760  0.0009567  -7.292 1.20e-12 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7.423 on 502 degrees of freedom
## Multiple R-squared:  0.2597, Adjusted R-squared:  0.2552
## F-statistic: 58.69 on 3 and 502 DF,  p-value: < 2.2e-16
##
## [1] "chas"
##
## Call:
## lm(formula = crim ~ x + x2 + x3)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -3.738 -3.661 -3.435  0.018 85.232
##
## Coefficients: (2 not defined because of singularities)
```

```
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   3.7444     0.3961   9.453   <2e-16 ***
## x            -1.8928     1.5061  -1.257    0.209
## x2                NA         NA      NA       NA
## x3                NA         NA      NA       NA
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 8.597 on 504 degrees of freedom
## Multiple R-squared:  0.003124,   Adjusted R-squared:  0.001146
## F-statistic: 1.579 on 1 and 504 DF,  p-value: 0.2094
##
## [1] "nox"
##
## Call:
## lm(formula = crim ~ x + x2 + x3)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -9.110 -2.068 -0.255  0.739 78.302
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   233.09      33.64   6.928 1.31e-11 ***
## x           -1279.37     170.40  -7.508 2.76e-13 ***
## x2           2248.54     279.90   8.033 6.81e-15 ***
## x3          -1245.70     149.28  -8.345 6.96e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7.234 on 502 degrees of freedom
## Multiple R-squared:  0.297,  Adjusted R-squared:  0.2928
## F-statistic: 70.69 on 3 and 502 DF,  p-value: < 2.2e-16
##
## [1] "rm"
##
## Call:
## lm(formula = crim ~ x + x2 + x3)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -18.485  -3.468  -2.221  -0.015  87.219
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 112.6246    64.5172   1.746   0.0815 .
## x           -39.1501    31.3115  -1.250   0.2118
## x2            4.5509     5.0099   0.908   0.3641
## x3           -0.1745     0.2637  -0.662   0.5086
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 8.33 on 502 degrees of freedom
## Multiple R-squared:  0.06779,    Adjusted R-squared:  0.06222
```

```
## F-statistic: 12.17 on 3 and 502 DF,  p-value: 1.067e-07
##
## [1] "age"
##
## Call:
## lm(formula = crim ~ x + x2 + x3)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -9.762 -2.673 -0.516  0.019 82.842
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -2.549e+00  2.769e+00  -0.920  0.35780
## x            2.737e-01  1.864e-01   1.468  0.14266
## x2          -7.230e-03  3.637e-03  -1.988  0.04738 *
## x3           5.745e-05  2.109e-05   2.724  0.00668 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7.84 on 502 degrees of freedom
## Multiple R-squared:  0.1742, Adjusted R-squared:  0.1693
## F-statistic: 35.31 on 3 and 502 DF,  p-value: < 2.2e-16
##
## [1] "dis"
##
## Call:
## lm(formula = crim ~ x + x2 + x3)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -10.757  -2.588   0.031   1.267  76.378
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  30.0476     2.4459  12.285  < 2e-16 ***
## x           -15.5543     1.7360  -8.960  < 2e-16 ***
## x2            2.4521     0.3464   7.078 4.94e-12 ***
## x3           -0.1186     0.0204  -5.814 1.09e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7.331 on 502 degrees of freedom
## Multiple R-squared:  0.2778, Adjusted R-squared:  0.2735
## F-statistic: 64.37 on 3 and 502 DF,  p-value: < 2.2e-16
##
## [1] "rad"
##
## Call:
## lm(formula = crim ~ x + x2 + x3)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -10.381  -0.412  -0.269   0.179  76.217
```

```
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.605545   2.050108  -0.295    0.768
## x            0.512736   1.043597   0.491    0.623
## x2          -0.075177   0.148543  -0.506    0.613
## x3           0.003209   0.004564   0.703    0.482
##
## Residual standard error: 6.682 on 502 degrees of freedom
## Multiple R-squared:    0.4,  Adjusted R-squared:  0.3965
## F-statistic: 111.6 on 3 and 502 DF,  p-value: < 2.2e-16
##
## [1] "tax"
##
## Call:
## lm(formula = crim ~ x + x2 + x3)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -13.273  -1.389   0.046   0.536  76.950
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.918e+01  1.180e+01   1.626    0.105
## x           -1.533e-01  9.568e-02  -1.602    0.110
## x2           3.608e-04  2.425e-04   1.488    0.137
## x3          -2.204e-07  1.889e-07  -1.167    0.244
##
## Residual standard error: 6.854 on 502 degrees of freedom
## Multiple R-squared:  0.3689, Adjusted R-squared:  0.3651
## F-statistic:  97.8 on 3 and 502 DF,  p-value: < 2.2e-16
##
## [1] "ptratio"
##
## Call:
## lm(formula = crim ~ x + x2 + x3)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -6.833 -4.146 -1.655  1.408 82.697
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 477.18405  156.79498   3.043  0.00246 **
## x           -82.36054   27.64394  -2.979  0.00303 **
## x2            4.63535    1.60832   2.882  0.00412 **
## x3           -0.08476    0.03090  -2.743  0.00630 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 8.122 on 502 degrees of freedom
## Multiple R-squared:  0.1138, Adjusted R-squared:  0.1085
## F-statistic: 21.48 on 3 and 502 DF,  p-value: 4.171e-13
##
```

```
## [1] "black"
##
## Call:
## lm(formula = crim ~ x + x2 + x3)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -13.096  -2.343  -2.128  -1.439  86.790
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.826e+01  2.305e+00   7.924  1.5e-14 ***
## x           -8.356e-02  5.633e-02  -1.483    0.139
## x2           2.137e-04  2.984e-04   0.716    0.474
## x3          -2.652e-07  4.364e-07  -0.608    0.544
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7.955 on 502 degrees of freedom
## Multiple R-squared:  0.1498, Adjusted R-squared:  0.1448
## F-statistic: 29.49 on 3 and 502 DF,  p-value: < 2.2e-16
##
## [1] "lstat"
##
## Call:
## lm(formula = crim ~ x + x2 + x3)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -15.234  -2.151  -0.486   0.066  83.353
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.2009656  2.0286452   0.592   0.5541
## x           -0.4490656  0.4648911  -0.966   0.3345
## x2           0.0557794  0.0301156   1.852   0.0646 .
## x3          -0.0008574  0.0005652  -1.517   0.1299
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7.629 on 502 degrees of freedom
## Multiple R-squared:  0.2179, Adjusted R-squared:  0.2133
## F-statistic: 46.63 on 3 and 502 DF,  p-value: < 2.2e-16
##
## [1] "medv"
##
## Call:
## lm(formula = crim ~ x + x2 + x3)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -24.427  -1.976  -0.437   0.439  73.655
##
## Coefficients:
```

```
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept) 53.1655381  3.3563105  15.840  < 2e-16 ***
## x           -5.0948305  0.4338321 -11.744  < 2e-16 ***
## x2           0.1554965  0.0171904   9.046  < 2e-16 ***
## x3          -0.0014901  0.0002038  -7.312 1.05e-12 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6.569 on 502 degrees of freedom
## Multiple R-squared:  0.4202, Adjusted R-squared:  0.4167
## F-statistic: 121.3 on 3 and 502 DF,  p-value: < 2.2e-16
```

So, the **p values** for the **cubic term** are **statistically significant** for the following variables:

'medv','ptratio','dis','age','nox','indus'

This suggests that a **cubic relationship** is a **decent fit** for these variables.

Chapter 6 | Problem 9

Part (A)

First we will split the data into training set and test set (75%-25% ratio). We get-

```
## [1] "overall observations :777"
```

```
## [1] "Train observations : 582"
```

```
## [1] "Test observations : 195"
```

Part (B)

We will now fit a linear model using least squares-

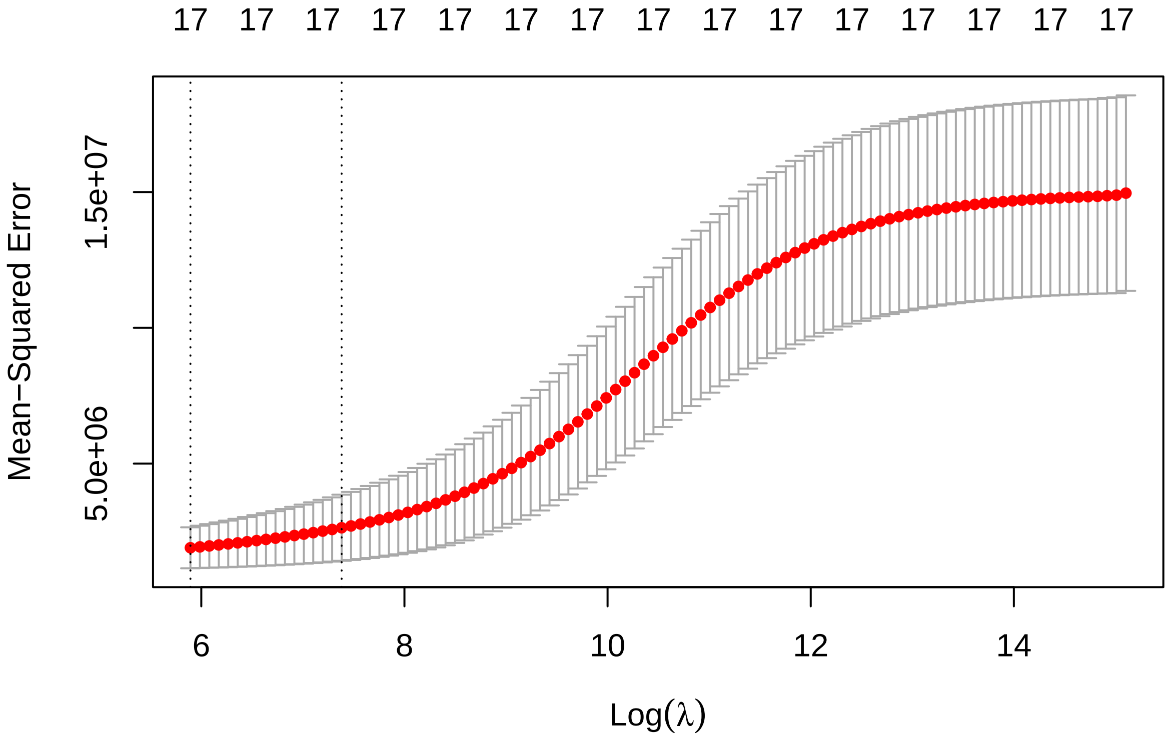
```
## [1] "Mean Square Error in test dataset for Least Sqaures is 946027.509426667"
```

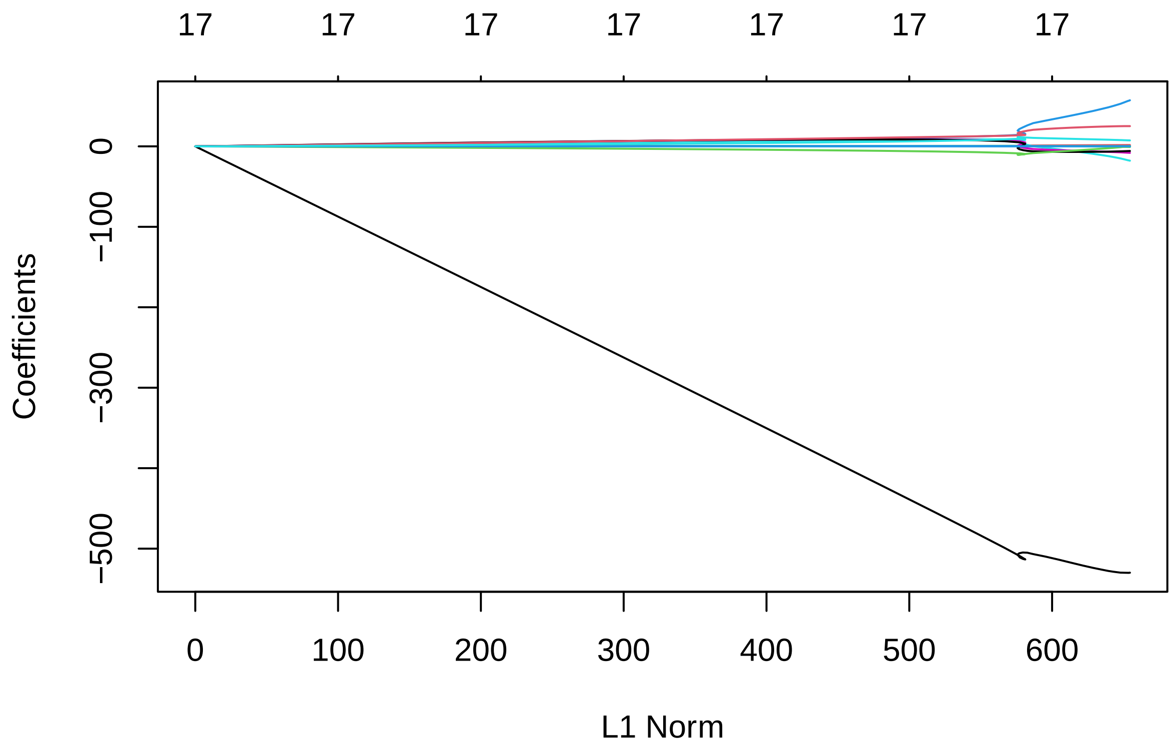
Part (C)

We will now fit a ridge regression model and choose lambda by 10-fold cross validation-

```
## Loading required package: Matrix
```

```
## Loaded glmnet 4.1-4
```


17 17 17 17 17 17 17 17 17 17 17 17 17 17 17
Mean−Squared Error
1.5e+07
5.0e+06
6
8
10
12
14
Log(λ)




```
## [1] "From the plot, best Lambda Value is: 362.499833110275"
```

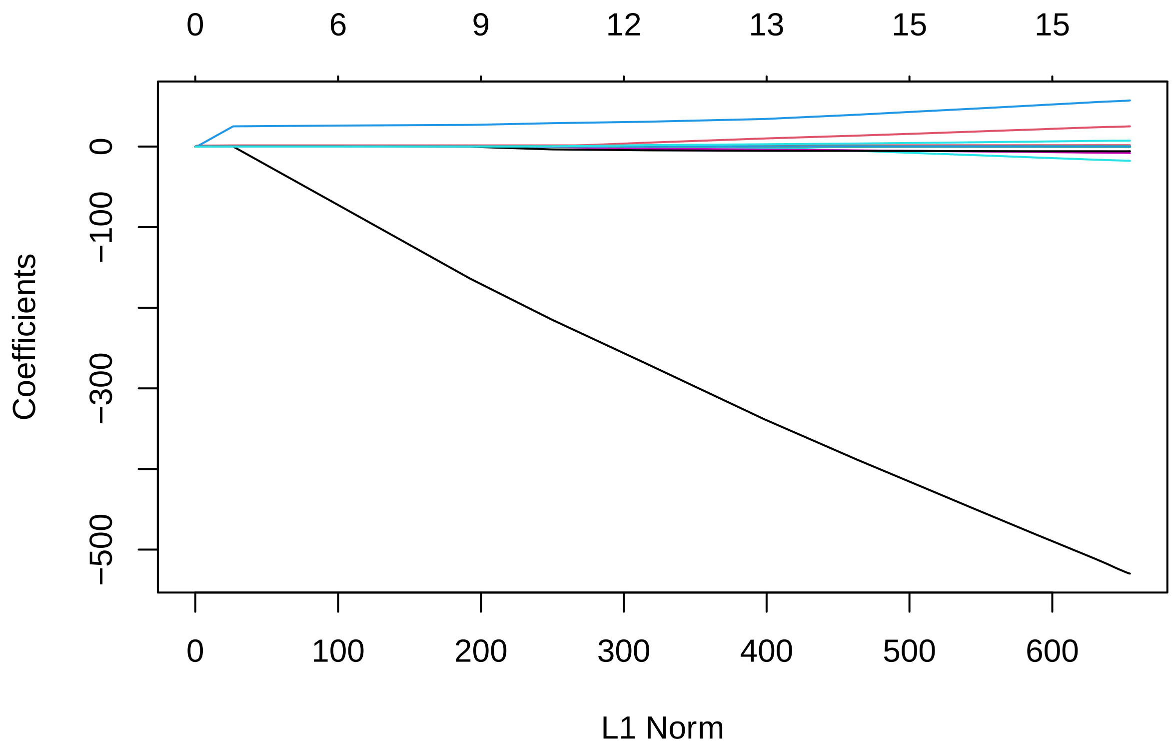
Now we will create a model with the best value of lambda-

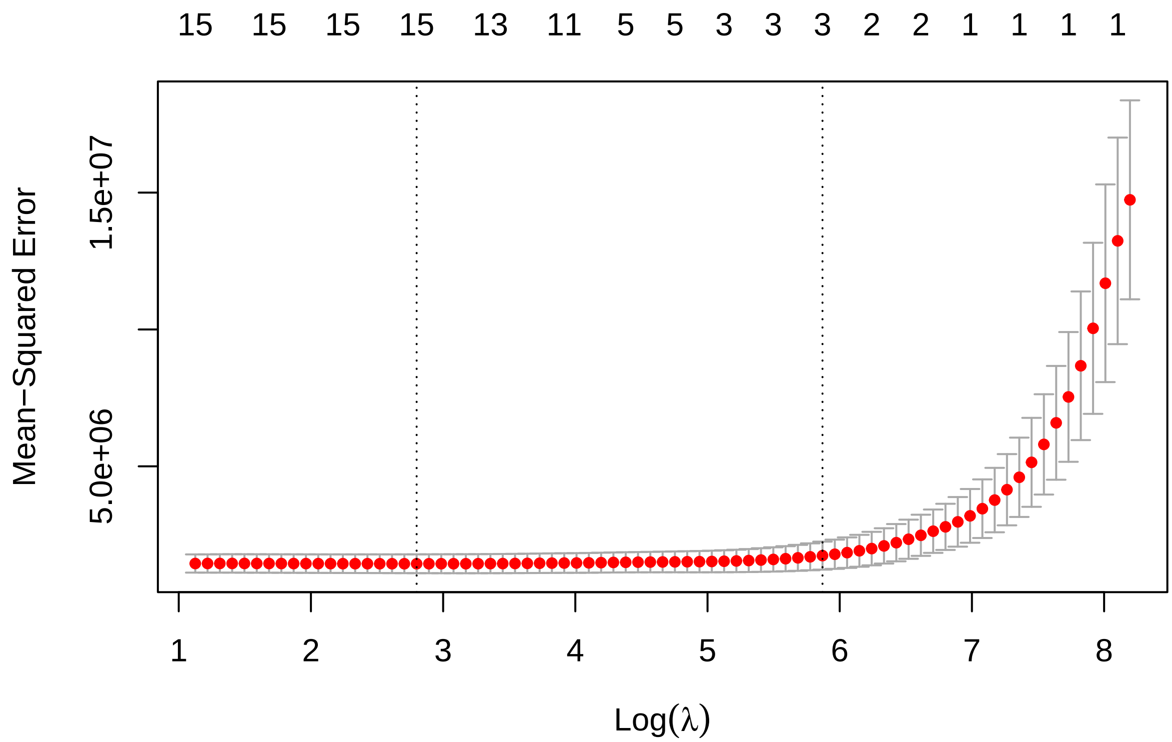
```
## [1] "Mean Square Error in test dataset for Ridge is 787423.370078229"
```

```
## [1] "Reduction in error is 16.7652777290333 %"
```

We will fit a lasso regression model now. Again we will use cross validation to find the best value of lambda

```
## Warning in regularize.values(x, y, ties, missing(ties), na.rm = na.rm):
## collapsing to unique 'x' values
```


0 6 9 12 13 15 15
Coefficients
0
−100
−300
−500
0 100 200 300 400 500 600
L1 Norm




```
## [1] "From the plot, best Lambda Value is: 16.4389270237198"
```

Now using this lambda value to fit a lasso regression model, we get-

```
## [1] "Mean Square Error in test dataset for Lasso is 884203.765389609"
```

```
## 19 x 1 sparse Matrix of class "dgCMatrix"
##                        s1
## (Intercept) -6.513413e+02
## (Intercept)  .
## PrivateYes  -4.358948e+02
## Accept       1.574802e+00
## Enroll      -4.967304e-01
## Top10perc    4.528939e+01
## Top25perc   -9.434910e+00
## F.Undergrad  .
## P.Undergrad  1.447851e-02
## Outstate    -6.503599e-02
## Room.Board   1.688291e-01
## Books        .
## Personal     6.321901e-03
## PhD         -5.746684e+00
## Terminal    -5.604042e+00
## S.F.Ratio    1.731442e+01
## perc.alumni -3.447091e-01
```

```
## Expend        6.974937e-02
## Grad.Rate     4.894400e+00
```

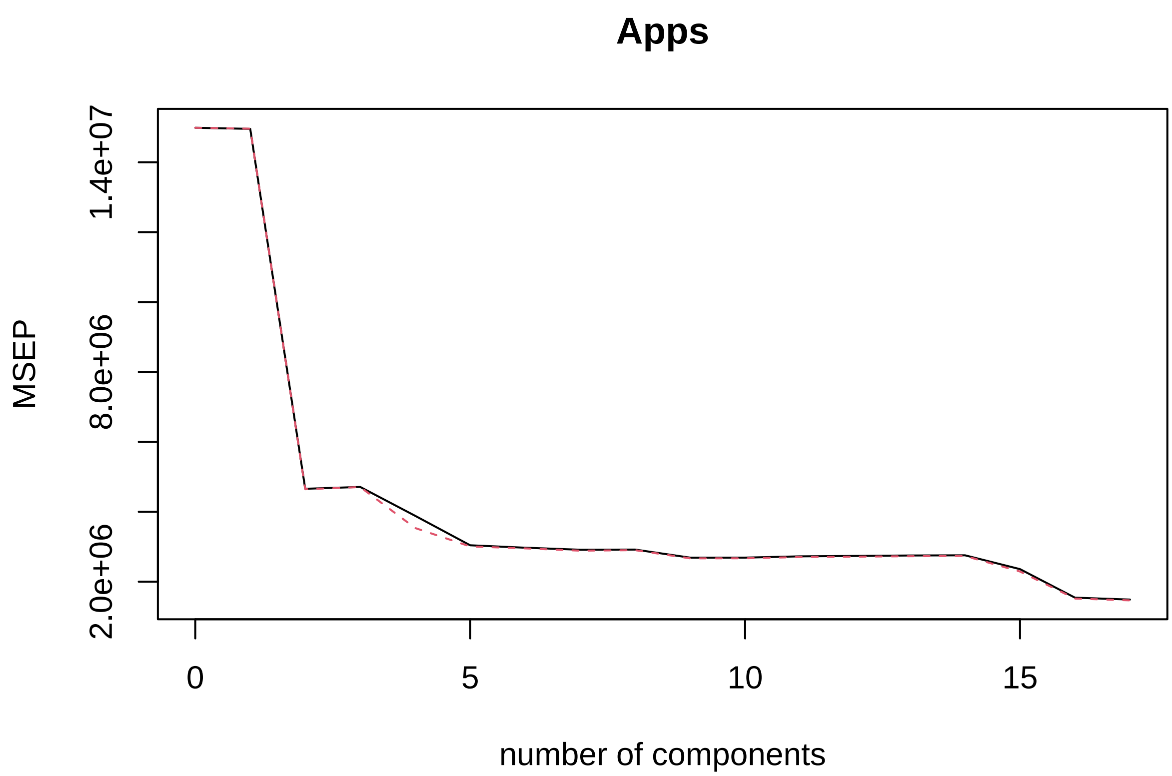
Out of Sample MSE for **Lasso (884203)** is **more** than that of **Ridge (787423)**

The **coefficients** for **F.Undergrad and Books are 0**

Part (E)

We will now fit a ridge regression model and choose lambda by 10-fold cross validation-

```
##
## Attaching package: 'pls'

## The following object is masked from 'package:stats':
##
##     loadings
```



So, clearly from the plot,the error is reduced when no. of components is 17.

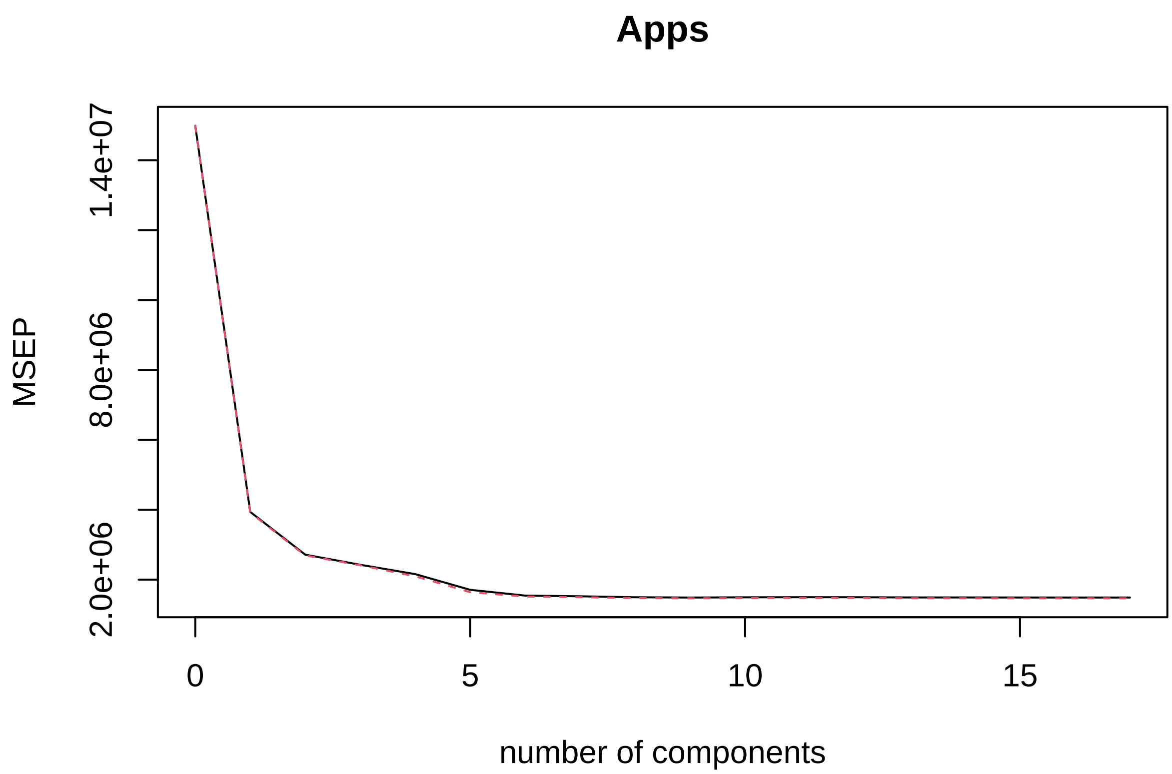
So, no variables were removed and **there is no difference between PCR and OLS models**.

If we are to check the MSE for PCR, we get-

```
## [1] "Mean Square Error in test dataset for PCR is 946027.509426667"
```

So the MSE for PCR is same as OLS which was expected.

Part (F)



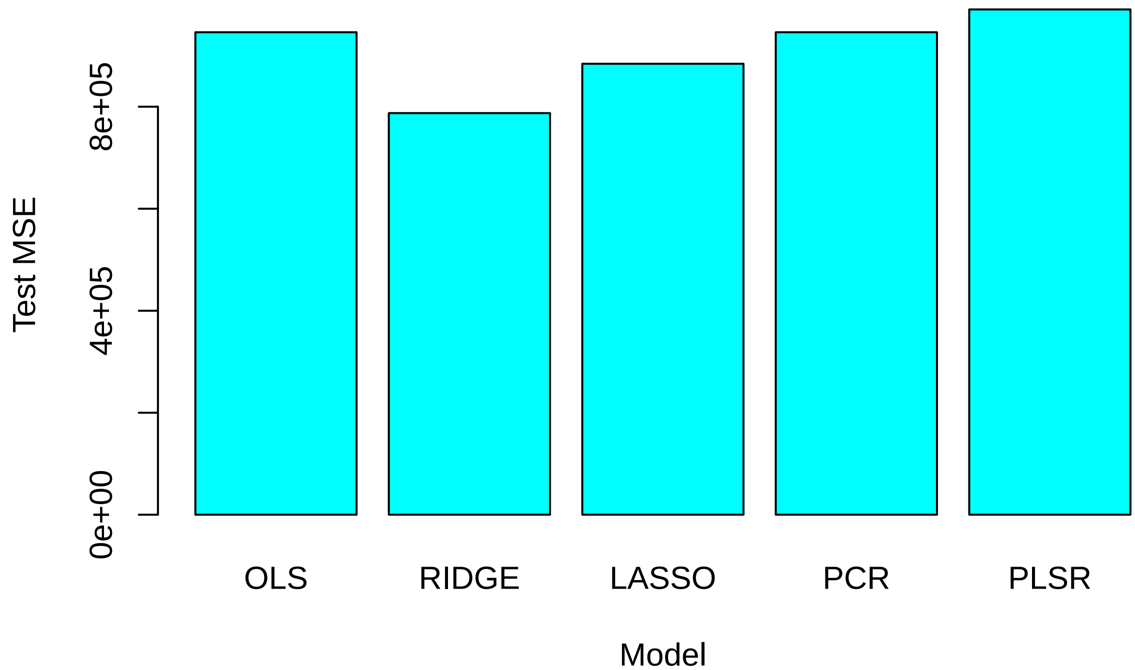
From this plot, we can see that the **total error is reduced when the no of components is 6**. Even if we add more components after 6, the total error stays the same.

We will now choose 6 components as we prefer a simpler model.

```
## [1] "Mean Square Error in test dataset for PLS with 6 components is 990764.154410396"
```

Part (G)

Now, having run all the models, we will compare the errors of all the models we have run so far by using a Bar Plot. We get-

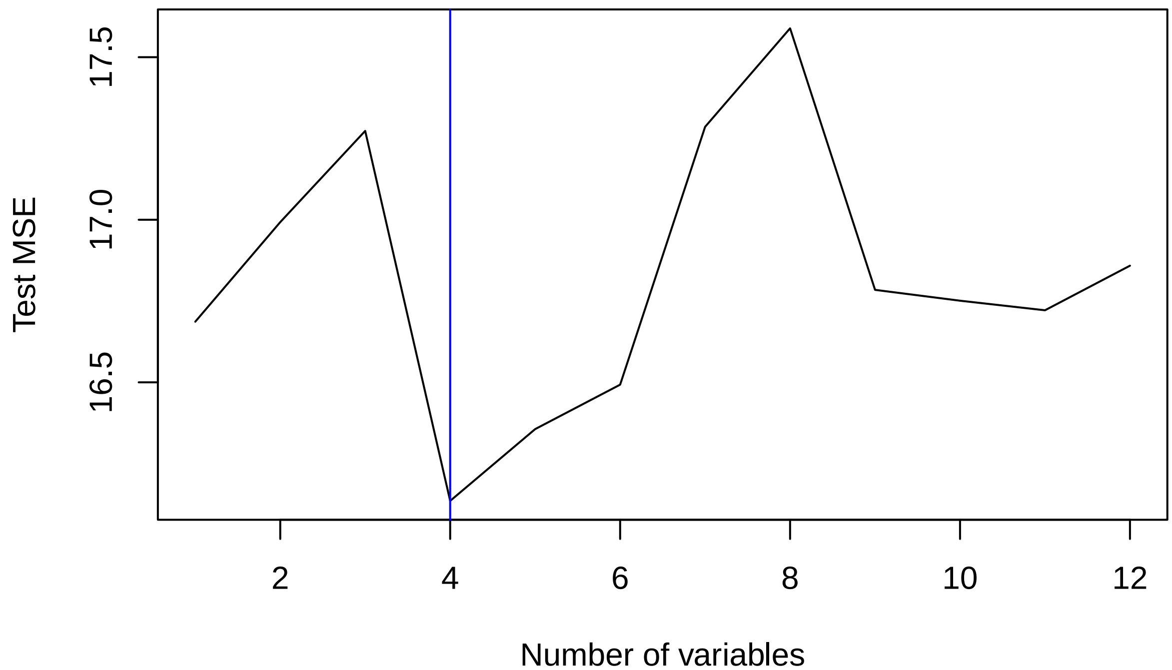


So, clearly from the plot, Ridge has the best performance followed by Lasso and then PCR=OLS and then finally PLSR

Chapter 6 | Problem 11

Part (A) and Part (B)

We will first try subset selection




```
## [1] "min Test MSE corresponding to best model is : 16.1354146346498"
```

So, a Model with **4 predictors** is giving us the **best MSE**. We will choose this model.

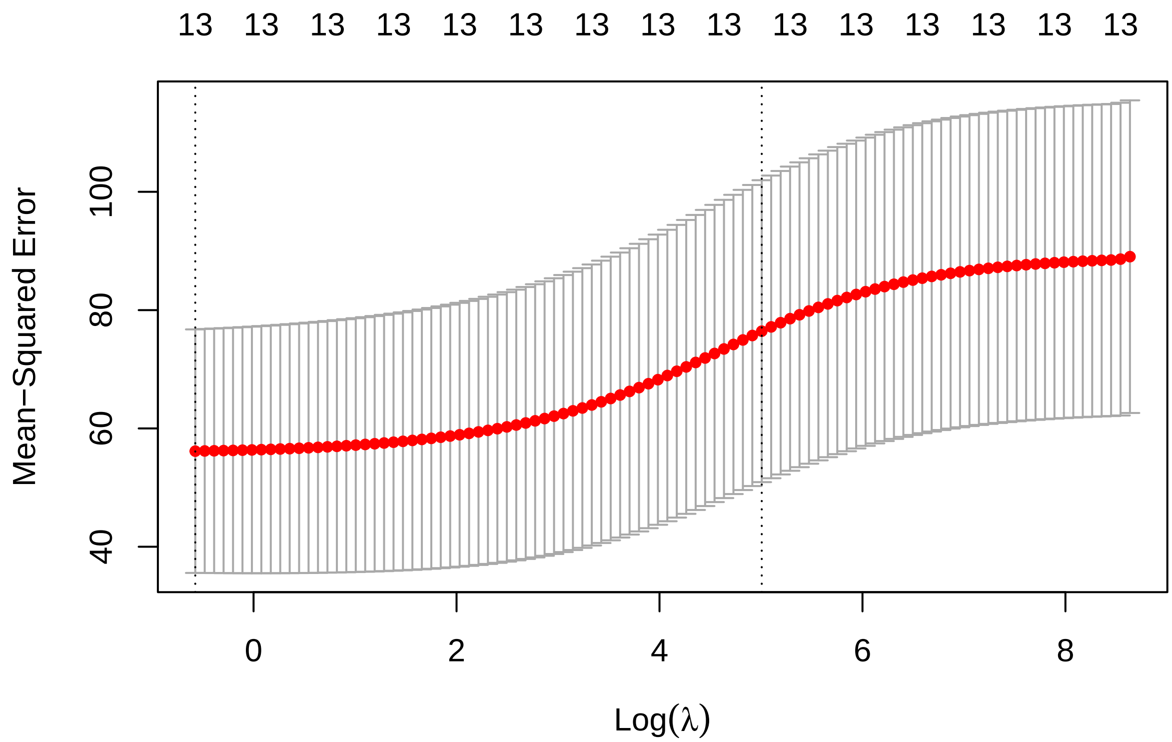
Among the variables, we want to figure out which variables are the best.

```
## [1] "Variables selected in the training dataset model"

## (Intercept)          zn         dis         rad        medv
##  6.05258361  0.06161735 -0.73492531  0.50939778 -0.23238041

## [1] "Variables selected in the overall dataset model"

## (Intercept)          zn         dis         rad        medv
##  5.26547997  0.05486634 -0.72291374  0.50020971 -0.19121698
```

The 4 variables associated with both the models are exactly the same. So, we can use this model.

So, out of 13 variables, our model used 4.

Now implementing Ridge Regression, we get-




```
## [1] "Best Lambda value is 0.562987756690407"
```

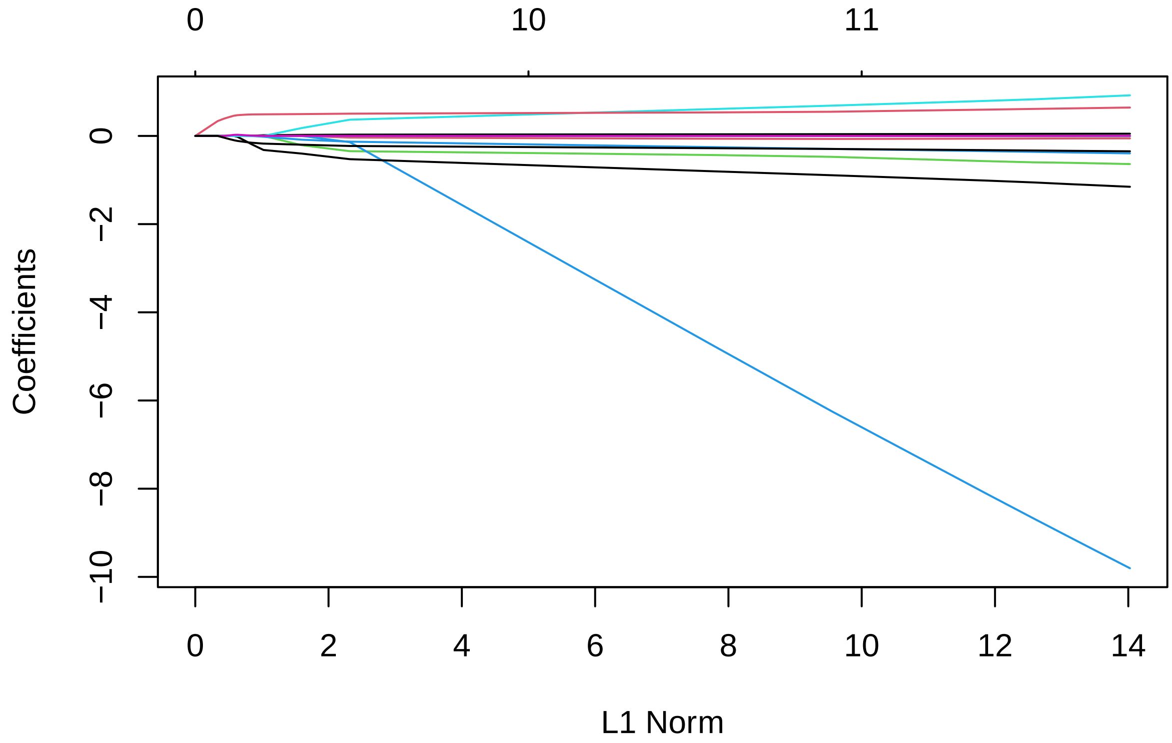
Using the above obtained value of lambda, we will now create a ridge regression model.

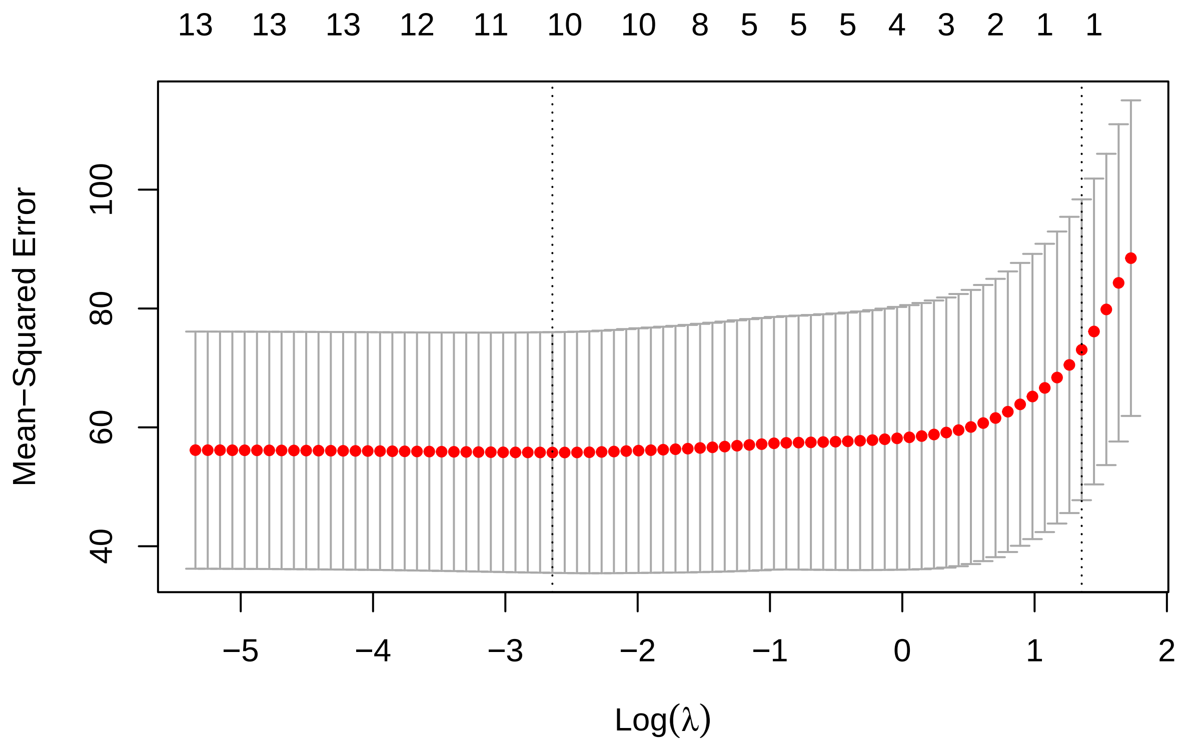
```
## [1] "MSE in test dataset for Ridge Regression Model using best Lambda value is 14.6155105523637"
```

So, the MSE for **Ridge Regression Model** is **slightly better** than the **Subset Selection Model (14.61 vs 16.13)**

Now we will try to fit a Lasso Regression Model. Firstly, we will use cross validation to find the best Lambda value just like before.

```
## Warning in regularize.values(x, y, ties, missing(ties), na.rm = na.rm):
## collapsing to unique 'x' values
```


0
10
11
Coefficients
0
−2
−4
−6
−8
−10
0
2
4
6
8
10
12
14
L1 Norm




```
## [1] "Best Lambda value is 0.0710409975861688"
```

Using this lamda value, we now fit a lasso regression model.

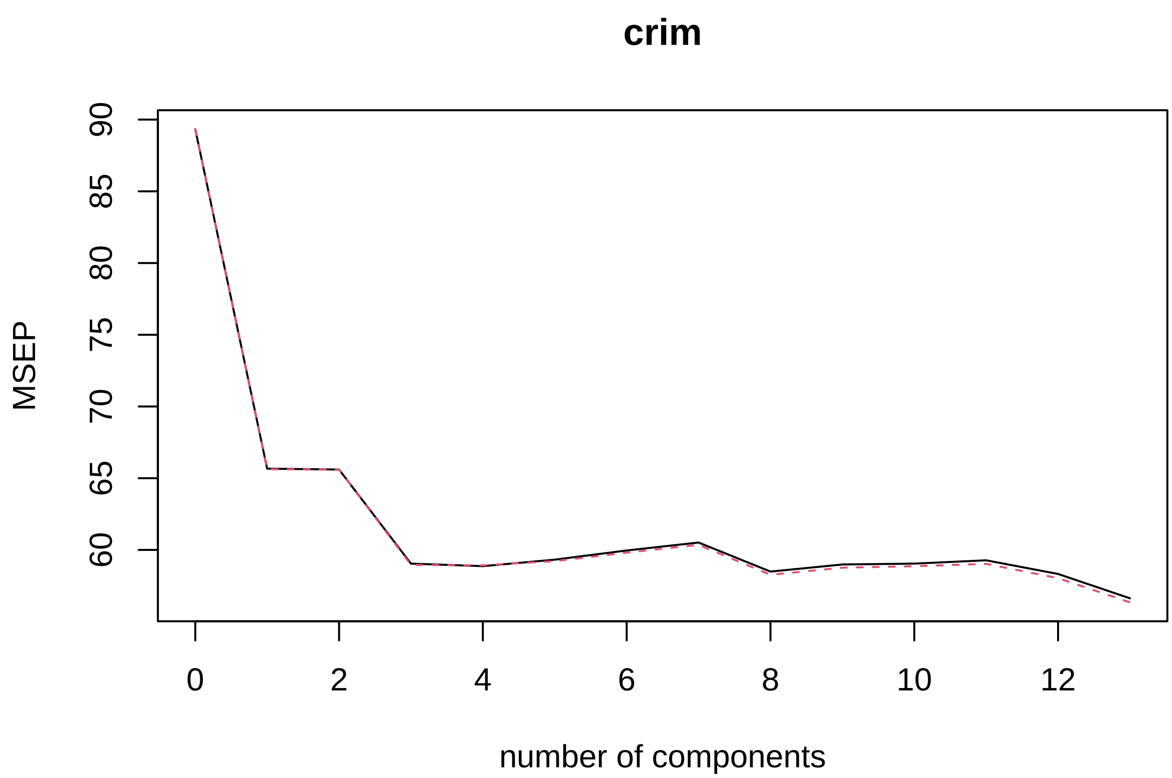
```
## [1] "Mean Square Error in test dataset for Lasso is 15.7270703406509"
```

```
## 15 x 1 sparse Matrix of class "dgCMatrix"
##                         s1
## (Intercept) 15.2844681137
## (Intercept)   .
## zn           0.0395529782
## indus       -0.0693783865
## chas        -0.4729988885
## nox         -6.2123705681
## rm           0.6842565042
## age          .
## dis         -0.8895878202
## rad          0.5478777891
## tax         -0.0004067214
## ptratio     -0.2933110405
## black       -0.0050553382
## lstat        .
## medv        -0.2951632641
```

MSE for **Lasso Regression** here is **slightly higher than Ridge Regression**.

Coefficients for **age** and **lstat** are **0**.

Now we will try to fit a PCR Model.



Here least MSE is obtained by using all the variables i.e all 13 variables.

```
## [1] "MSE from best PCR Model is 16.7952060949636"
```

The MSE of PCR is slightly worse than both Ridge and Lasso So, we will use Ridge Regression as it is giving us the least MSE among all the models

Part (C)

We selected **Ridge Regression** out of all the given models as it had the least MSE. The Ridge Regression Model uses **all the 13 predictor variables**.

Amongst the other models used, Subset Selection used only 4 predictor variables whereas Lasso Regression used 11 predictor variables and PCR used all 13 predictor variables in the model.

Chapter 8 | Problem 8

```
## Loading required package: lattice
```

```
##
## Attaching package: 'caret'

## The following object is masked from 'package:pls':
##
##     R2
```

Part (A)

Firstly we will encode the categorical variables with Yes or No Categories

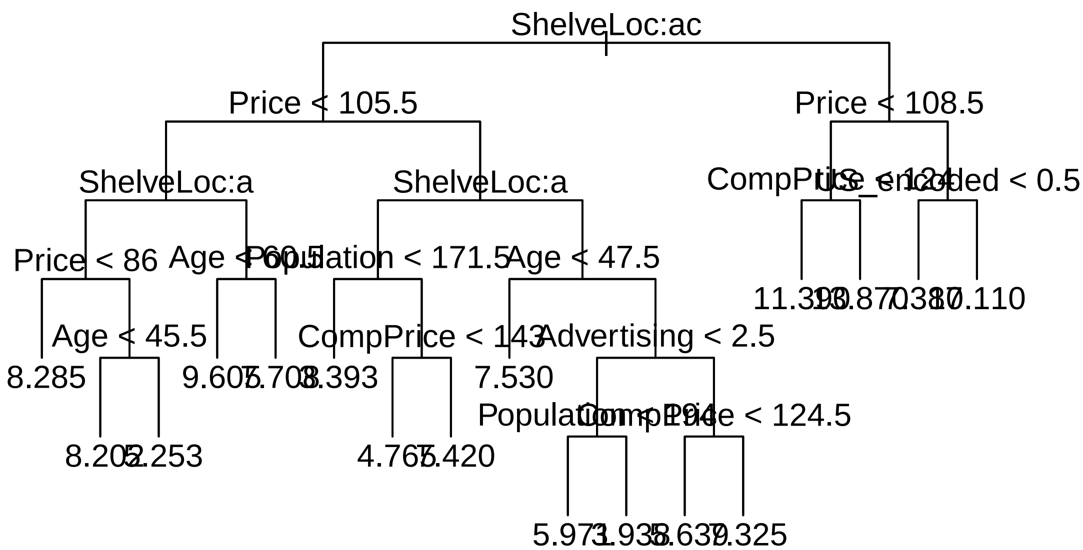
Now we will split the dataset into Training and Test Datasets as instructed. We will split it into a 75%-25% ratio.

```
## [1] "overall observations : 400"

## [1] "Train observations :  300"

## [1] "Test observations :  100"
```

Part (B)

Now we will fit a regression tree to the training set.

```
##
## Regression tree:
## tree(formula = Sales ~ ., data = train)
## Variables actually used in tree construction:
## [1] "ShelveLoc"   "Price"       "Age"         "Population"  "CompPrice"
## [6] "Advertising" "US_encoded"
## Number of terminal nodes:  17
## Residual mean deviance:  2.619 = 741.3 / 283
## Distribution of residuals:
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
## -4.86000 -0.95630 -0.00338  0.00000  1.03200  4.78200
```




```
## [1] "MSE for Test Dataset using Big tree is :  4.8164224870796"
```

```
## [1] "MSE for Train Dataset using Big tree is :  2.47094140429428"
```

Root variable used by tree is **“ShelveLoc”**.

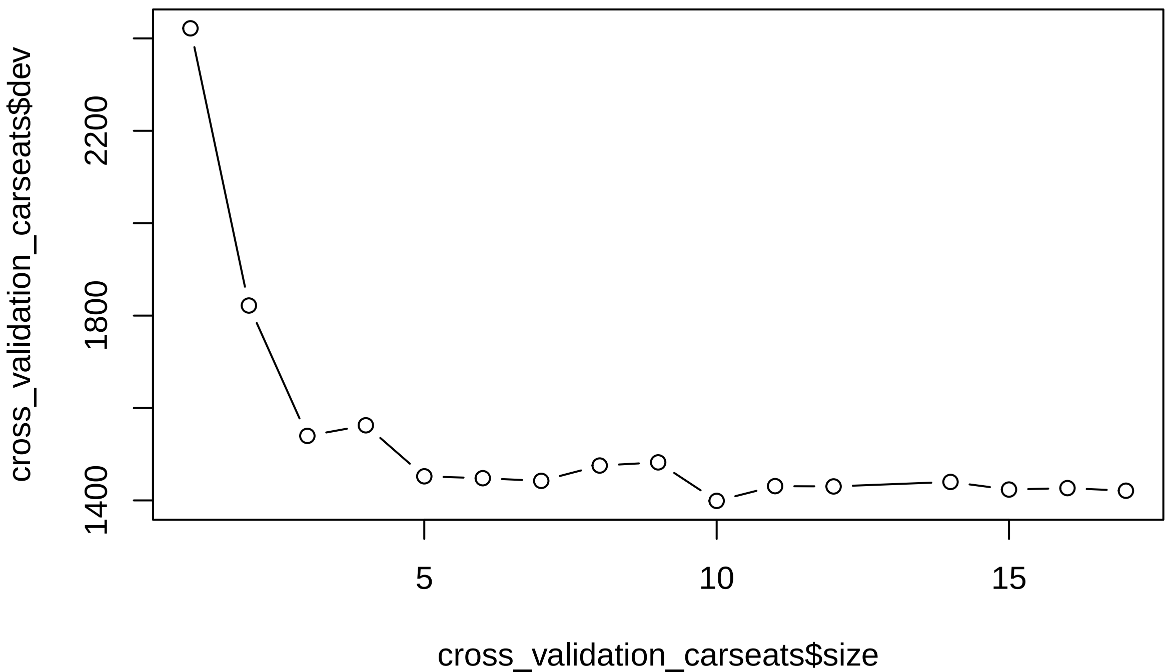
Only the variables “ShelveLoc”,“Price”,“Age”,“Population”,“CompPrice”,“Advertising”,“US_encoded” have been used to build the tree. This suggests that the remaining variables are not as important.

17 Terminal Nodes are present in this Tree.

The **MSE** for Test Set is **4.81**

Part (C)

Now we will perform cross validation to find out the optimal tree complexity.



We can see that at **tree size=10** we get least deviation, so we will prune tree to 10 nodes.

```
## [1] "MSE for Train Dataset using Pruned tree is :  3.16023012173643"
```

```
## [1] "MSE for Test Dataset using Pruned tree is : 4.687793221518"
```

So by using Pruned Tree, MSE for Training Set increased significantly (**from 2.47 to 3.16**) but for Test Dataset there is an improvement in MSE from **4.82 for Big tree to 4.69 for pruned tree**. So, **bias of the model increased but variance decreased.**

Part (D)

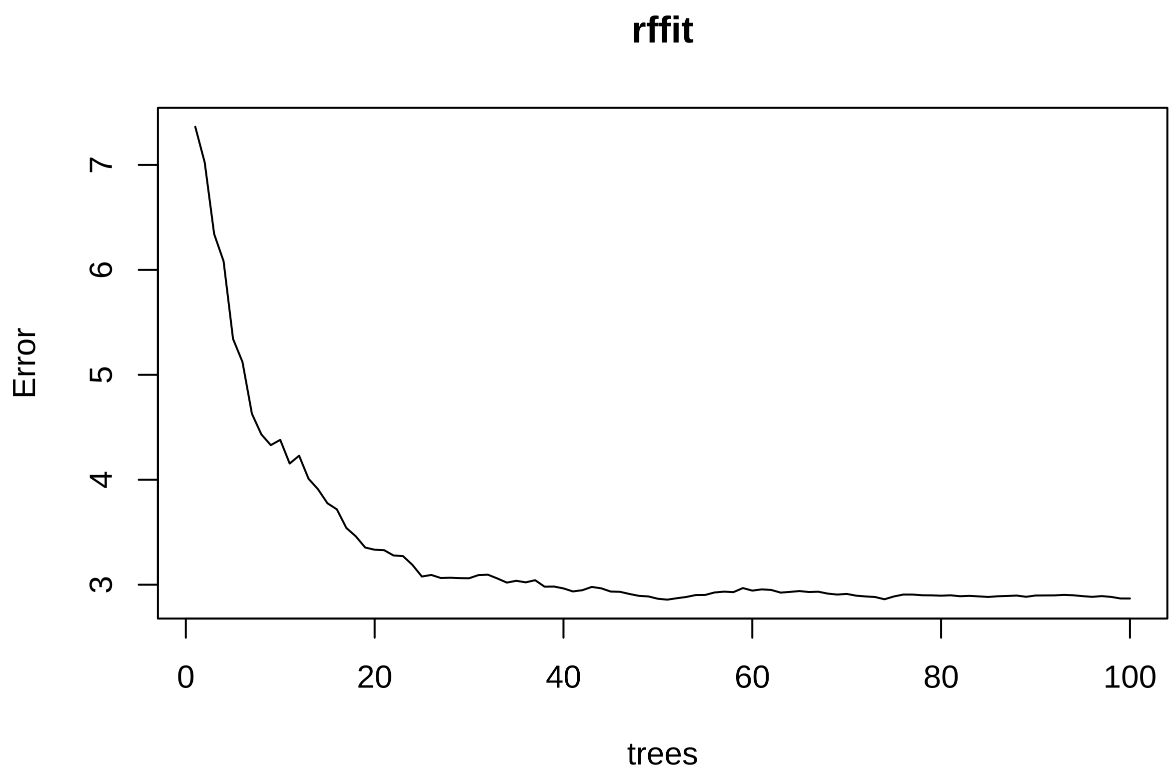
```
## randomForest 4.7-1.1
```

```
## Type rfNews() to see new features/changes/bug fixes.
```

```
##
## Attaching package: 'randomForest'
```

```
## The following object is masked from 'package:dplyr':
##
##     combine
```

```
## The following object is masked from 'package:ggplot2':
##
##     margin
```



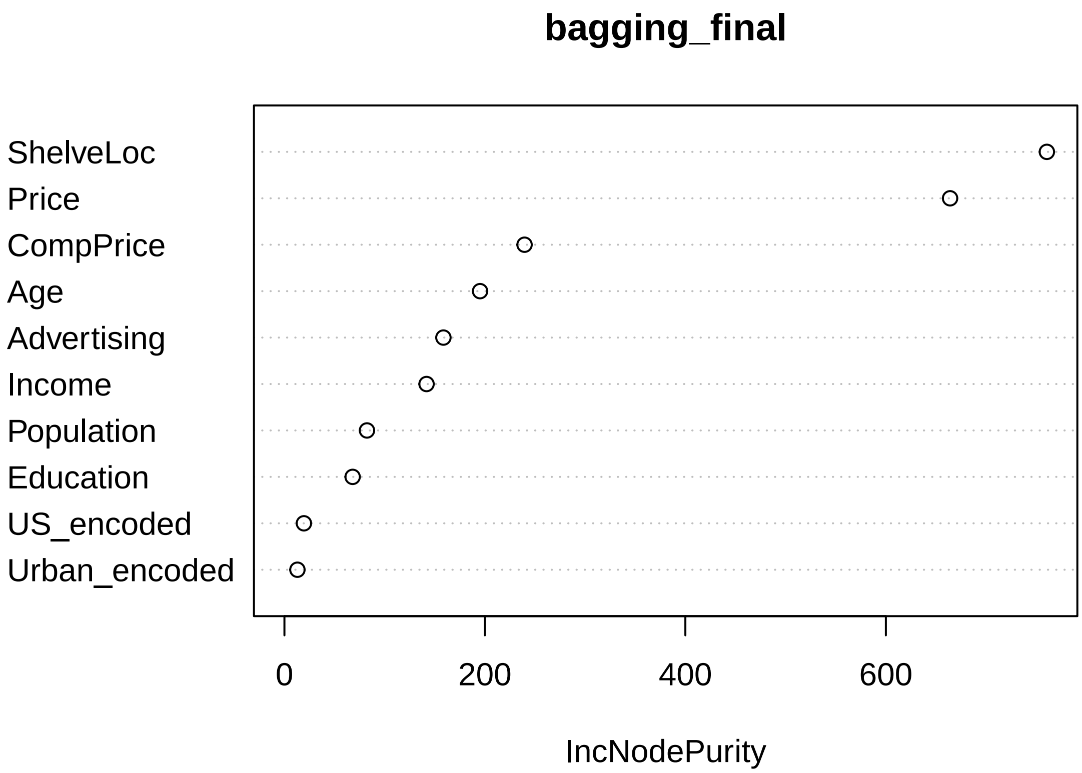
So lowest error is when no. of trees is around **75.**

So, we now build a bagging model with **trees=75** and **no. of predictors=10** (the entire predictor set)

```
## [1] "MSE for Test Dataset using Best tree is :  2.78074755190094"
```

Test Test MSE obtained here is about **2.78**. This is **much lesser** than the test MSE we obtained via a pruned decision tree (**4.69**).

Now, using this bagging model, we will plot the importance of variables-

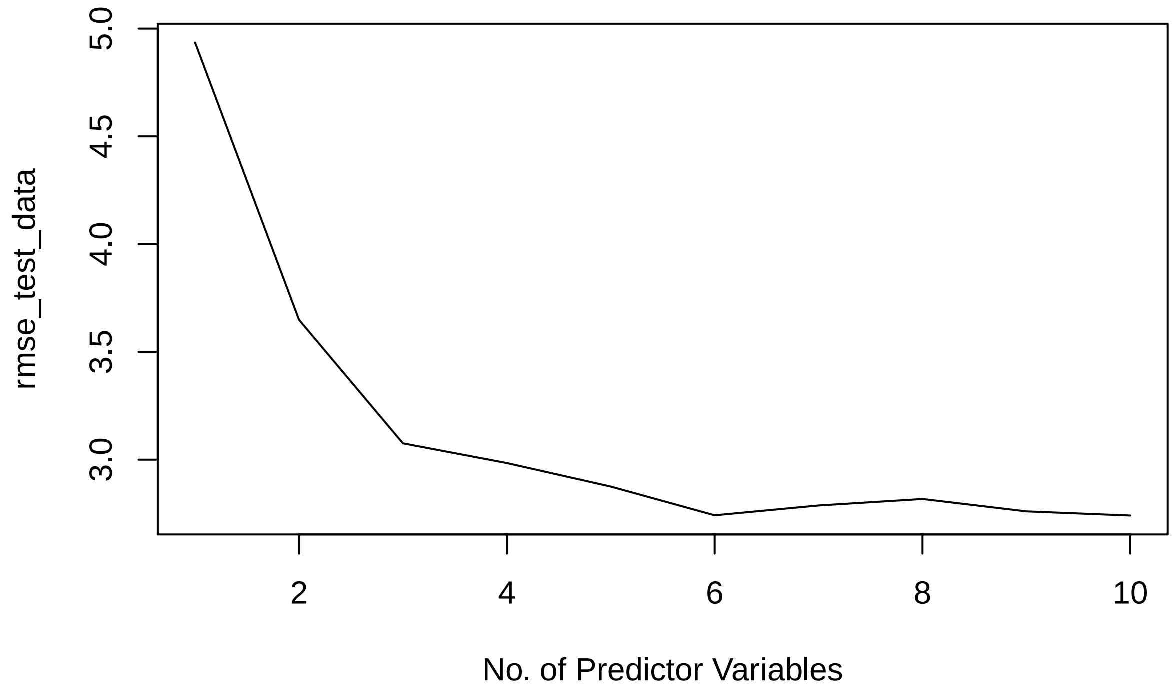



```
##                IncNodePurity
## CompPrice          239.54146
## Income             141.77882
## Advertising        158.59993
## Population          82.34495
## Price              664.08593
## ShelveLoc          760.63741
## Age                195.17349
## Education           68.03700
## US_encoded          19.43747
## Urban_encoded       12.95025
```

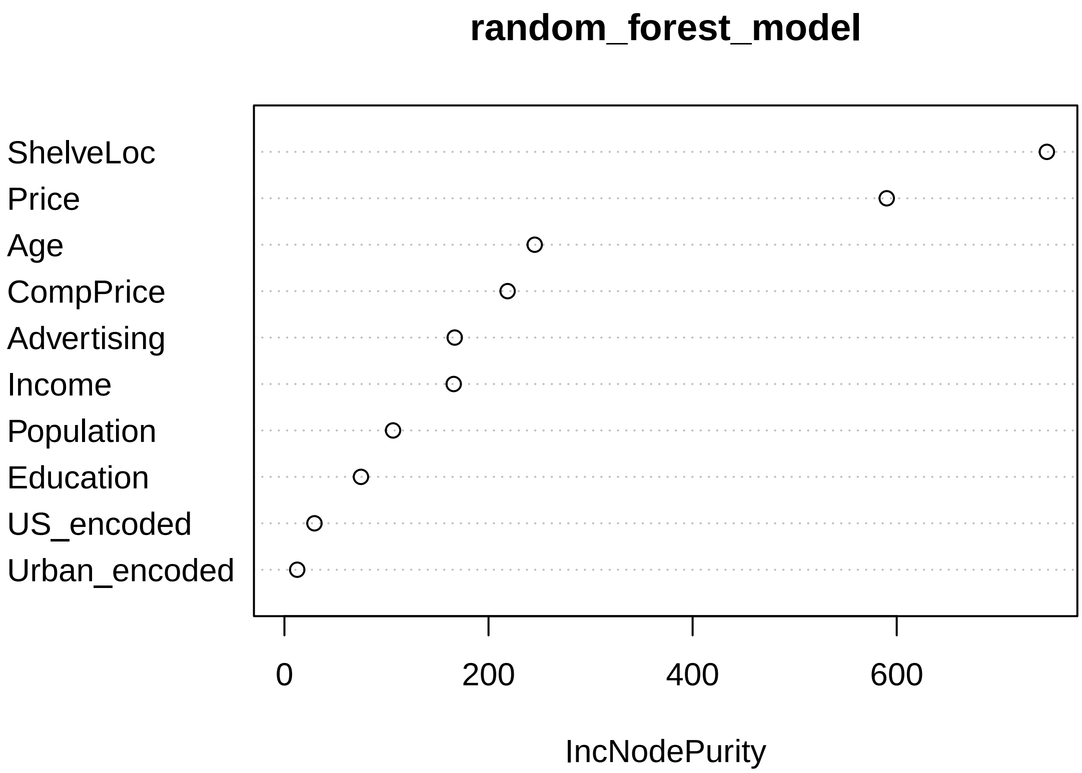
So, clearly the most important variables are **ShelveLoc** and **Price**. **US_encoded** and **Urban_encoded** are the least important variables as per the plot.

Part (E)

Now we will try building Random Forests with different values of m from 1-10. Let us take the **no. of trees = 75** (same as in Bagging).



So, the least error is obtained when **m=6**. So, building a Random Forest model with **m=6**, we get-




```
##                IncNodePurity
## CompPrice          218.67649
## Income             165.87163
## Advertising        166.86583
## Population         106.42177
## Price              590.22050
## ShelveLoc          747.12563
## Age                245.25493
## Education           74.98408
## US_encoded          29.41719
## Urban_encoded       12.52151
```

Once again, the most important variables are **ShelveLoc** and **Price** and the least important are **Urban_encoded** and **US_encoded**. The results seem similar to ones obtained via Bagging.

Chapter 8 | Problem 11

```
## Loaded gbm 2.1.8
```

We will perform data cleaning. We need to encode the Purchase variable.

Part (A)

As instructed, we will keep 1000 rows as training set and the remaining as test set

```
## [1] "Rows in Training Dataset are :  1000"
```

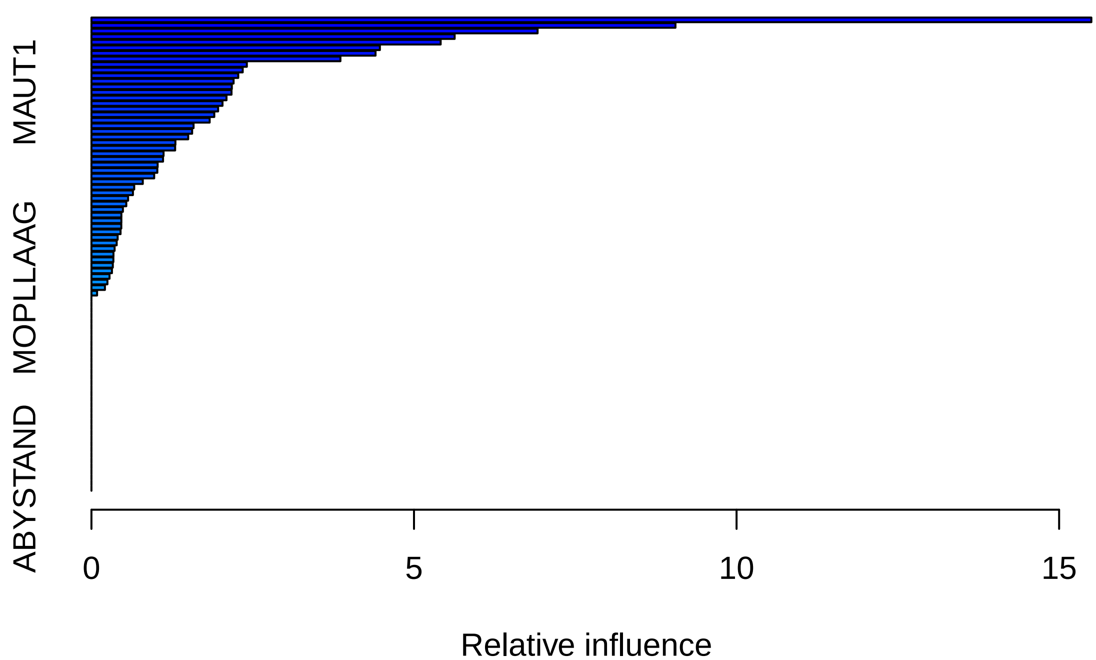
```
## [1] "Rows in Testing Dataset are :  4822"
```

Part (B)

We will now fit a Boosting Model with **no. of trees=1000** and **shrinkage parameter=0.01** to the training set.

```
## Warning in gbm.fit(x = x, y = y, offset = offset, distribution = distribution, :
## variable 50: PVRAAUT has no variation.
```

```
## Warning in gbm.fit(x = x, y = y, offset = offset, distribution = distribution, :
## variable 71: AVRAAUT has no variation.
```




```
##                   var     rel.inf
## PPERSAUT PPERSAUT 15.49989017
## MKOOPKLA MKOOPKLA  9.05289495
## MOPLHOOG MOPLHOOG  6.91606553
## PBRAND     PBRAND  5.63022222
## MBERMIDD MBERMIDD  5.41325272
## MINK3045 MINK3045  4.47115058
## MGODGE     MGODGE  4.40406789
## ABRAND     ABRAND  3.85938452
```

```
## MSKA         MSKA  2.40898085
## MSKC         MSKC  2.34487132
## MAUT2       MAUT2  2.27582500
## PWAPART   PWAPART  2.20354003
## MBERARBG MBERARBG  2.17655343
## MAUT1       MAUT1  2.17159225
## MOSTYPE   MOSTYPE  2.09353380
## MGODPR     MGODPR  2.03188459
## MFWEKIND MFWEKIND  1.96373482
## MINKGEM   MINKGEM  1.90443015
## MRELGE     MRELGE  1.83283054
## MAUT0       MAUT0  1.58329901
## MGODOV     MGODOV  1.56072741
## PBYSTAND PBYSTAND  1.50017494
## MBERHOOG MBERHOOG  1.30158671
## MSKB1       MSKB1  1.29749550
## MRELOV     MRELOV  1.11833559
## MFGEKIND MFGEKIND  1.10990180
## MINK7512 MINK7512  1.02647138
## MHKOOP     MHKOOP  1.02151649
## MGODRK     MGODRK  0.97254661
## MINKM30   MINKM30  0.79601630
## MHHUUR     MHHUUR  0.66284922
## MOPLMIDD MOPLMIDD  0.64335880
## MBERBOER MBERBOER  0.56820183
## PLEVEN     PLEVEN  0.54022775
## MINK4575 MINK4575  0.48768588
## MGEMOMV   MGEMOMV  0.46433447
## MSKD         MSKD  0.46370647
## MGEMLEEF MGEMLEEF  0.46278608
## MFALLEEN MFALLEEN  0.45131665
## PMOTSCO   PMOTSCO  0.40617001
## MZPART     MZPART  0.39133647
## MZFONDS   MZFONDS  0.35913399
## MBERARBO MBERARBO  0.34151150
## APERSAUT APERSAUT  0.34037181
## MOSHOOFD MOSHOOFD  0.33134723
## MINK123M MINK123M  0.31807358
## MSKB2       MSKB2  0.28020970
## MRELSA     MRELSA  0.24811103
## MOPLLAAG MOPLLAAG  0.20903129
## MBERZELF MBERZELF  0.08745917
## MAANTHUI MAANTHUI  0.00000000
## PWABEDR   PWABEDR  0.00000000
## PWALAND   PWALAND  0.00000000
## PBESAUT   PBESAUT  0.00000000
## PVRAAUT   PVRAAUT  0.00000000
## PAANHANG PAANHANG  0.00000000
## PTRACTOR PTRACTOR  0.00000000
## PWERKT     PWERKT  0.00000000
## PBROM       PBROM  0.00000000
## PPERSONG PPERSONG  0.00000000
## PGEZONG   PGEZONG  0.00000000
## PWAOREG   PWAOREG  0.00000000
```

```
## PZEILPL   PZEILPL  0.00000000
## PPLEZIER PPLEZIER  0.00000000
## PFIETS     PFIETS  0.00000000
## PINBOED   PINBOED  0.00000000
## AWAPART   AWAPART  0.00000000
## AWABEDR   AWABEDR  0.00000000
## AWALAND   AWALAND  0.00000000
## ABESAUT   ABESAUT  0.00000000
## AMOTSCO   AMOTSCO  0.00000000
## AVRAAUT   AVRAAUT  0.00000000
## AAANHANG AAANHANG  0.00000000
## ATRACTOR ATRACTOR  0.00000000
## AWERKT     AWERKT  0.00000000
## ABROM       ABROM  0.00000000
## ALEVEN     ALEVEN  0.00000000
## APERSONG APERSONG  0.00000000
## AGEZONG   AGEZONG  0.00000000
## AWAOREG   AWAOREG  0.00000000
## AZEILPL   AZEILPL  0.00000000
## APLEZIER APLEZIER  0.00000000
## AFIETS     AFIETS  0.00000000
## AINBOED   AINBOED  0.00000000
## ABYSTAND ABYSTAND  0.00000000
```

So from the variable importance plot, the 5 most important predictor variables are: **PPERSAUT,MKOOPKLA,MOPLHOOG,PBRAND,MBERMIDD**

Part (C)

```
## Using 1000 trees...
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    0    1
##          0 4413  255
##          1  120   34
##
##                Accuracy : 0.9222
##                  95% CI : (0.9143, 0.9296)
##     No Information Rate : 0.9401
##     P-Value [Acc > NIR] : 1
##
##                   Kappa : 0.1167
##
##  Mcnemar's Test P-Value : 4.525e-12
##
##             Sensitivity : 0.9735
##             Specificity : 0.1176
##          Pos Pred Value : 0.9454
##          Neg Pred Value : 0.2208
##             Prevalence : 0.9401
##          Detection Rate : 0.9152
```

```
##    Detection Prevalence : 0.9681
##       Balanced Accuracy : 0.5456
##
##        'Positive' Class : 0
##
```

So, **prediction accuracy** of the Boosting model is 34/(120+34)~**22.08%**.

Now applying Logistic Regression to the training set, we get-

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```
## Warning in predict.lm(object, newdata, se.fit, scale = 1, type = if (type == :
## prediction from a rank-deficient fit may be misleading
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    0    1
##          0 4183  231
##          1  350   58
##
##                Accuracy : 0.8795
##                  95% CI : (0.87, 0.8886)
##     No Information Rate : 0.9401
##     P-Value [Acc > NIR] : 1
##
##                   Kappa : 0.1035
##
##  Mcnemar's Test P-Value : 9.807e-07
##
##             Sensitivity : 0.9228
##             Specificity : 0.2007
##          Pos Pred Value : 0.9477
##          Neg Pred Value : 0.1422
##              Prevalence : 0.9401
##          Detection Rate : 0.8675
##    Detection Prevalence : 0.9154
##       Balanced Accuracy : 0.5617
##
##        'Positive' Class : 0
##
```

From the confusion matrix, prediction accuracy of the Logistic Regression Model is 58/(350+58)~**14.21%**

So, in this case, **Boosting** has **higher precision** than **Logistic Regression.**

Chapter 10 | Problem 7

We want to show that the proportionality holds true for the dataset USArrests.

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.2 --
## v tibble  3.1.7     v purrr   0.3.4
## v tidyr   1.2.0     v stringr 1.4.0
## v readr   2.1.2     v forcats 0.5.1
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x randomForest::combine() masks dplyr::combine()
## x tidyr::expand()        masks Matrix::expand()
## x dplyr::filter()        masks stats::filter()
## x dplyr::lag()           masks stats::lag()
## x purrr::lift()          masks caret::lift()
## x randomForest::margin()  masks ggplot2::margin()
## x tidyr::pack()          masks Matrix::pack()
## x dplyr::select()        masks MASS::select()
## x tidyr::unpack()        masks Matrix::unpack()
```

We will first load the dataset.

```
## # A tibble: 50 x 4
##    Murder Assault UrbanPop  Rape
##     <dbl>   <int>    <int> <dbl>
##  1   13.2     236       58  21.2
##  2   10       263       48  44.5
##  3    8.1     294       80  31
##  4    8.8     190       50  19.5
##  5    9       276       91  40.6
##  6    7.9     204       78  38.7
##  7    3.3     110       77  11.1
##  8    5.9     238       72  15.8
##  9   15.4     335       80  31.9
## 10   17.4     211       60  25.8
## # ... with 40 more rows
## # i Use `print(n = ...)` to see more rows
```

We will now have to center and scale the variables in the dataset to between 0 and 1.

```
## # A tibble: 50 x 4
##    Murder Assault UrbanPop  Rape
##     <dbl>   <int>    <int> <dbl>
##  1   13.2     236       58  21.2
##  2   10       263       48  44.5
##  3    8.1     294       80  31
##  4    8.8     190       50  19.5
##  5    9       276       91  40.6
##  6    7.9     204       78  38.7
##  7    3.3     110       77  11.1
##  8    5.9     238       72  15.8
##  9   15.4     335       80  31.9
## 10   17.4     211       60  25.8
## # ... with 40 more rows
## # i Use `print(n = ...)` to see more rows
```

```
##      Murder           Assault           UrbanPop             Rape
##  Min.   :-1.6044   Min.   :-1.5090   Min.   :-2.31714   Min.   :-1.4874
```

```
##  1st Qu.:-0.8525   1st Qu.:-0.7411   1st Qu.:-0.76271   1st Qu.:-0.6574
##  Median :-0.1235   Median :-0.1411   Median : 0.03178   Median :-0.1209
##  Mean   : 0.0000   Mean   : 0.0000   Mean   : 0.00000   Mean   : 0.0000
##  3rd Qu.: 0.7949   3rd Qu.: 0.9388   3rd Qu.: 0.84354   3rd Qu.: 0.5277
##  Max.   : 2.2069   Max.   : 1.9948   Max.   : 1.75892   Max.   : 2.6444
```

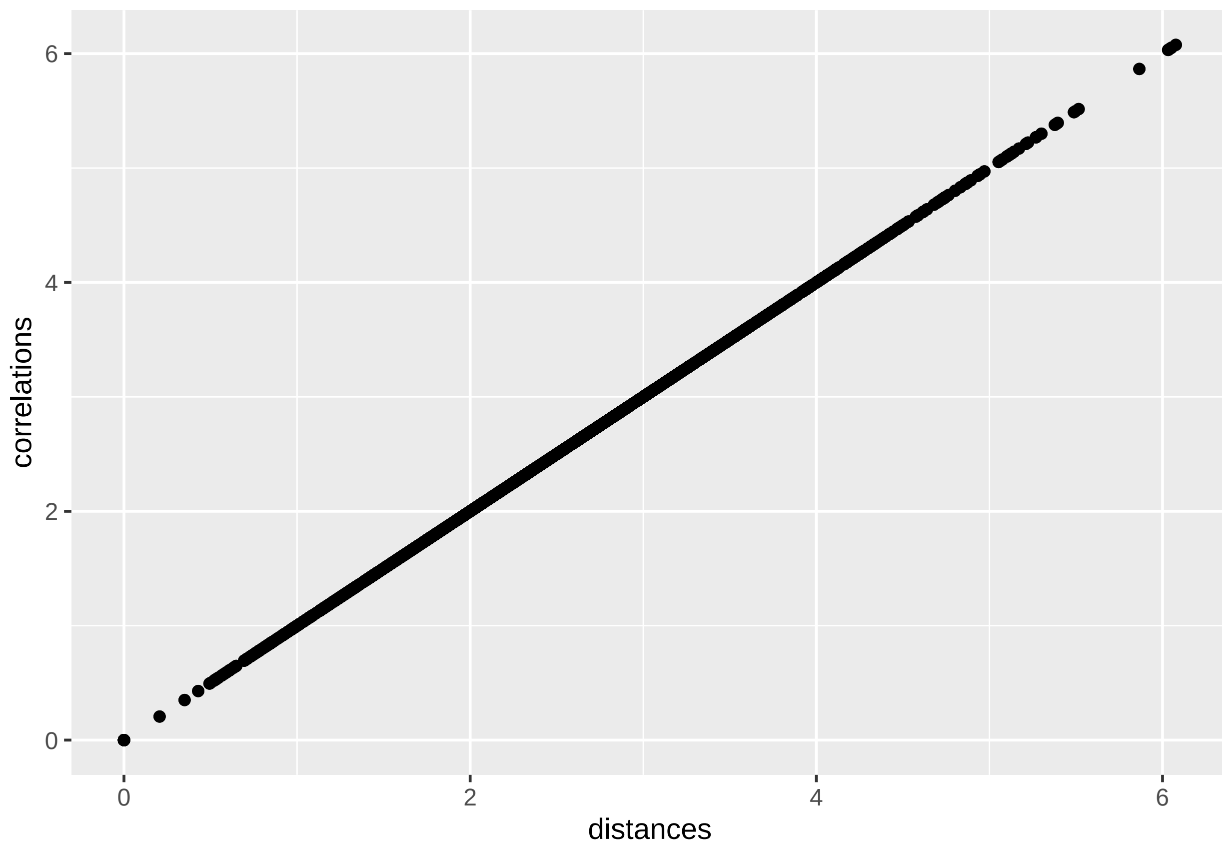
Now we will compute the pair-wise Euclidean distances for all the points in the scaled dataset.

```
##   1        2       3       4       5        6        7        8        9
## 1 0 2.703754 2.29352 1.28981 3.26311 2.651067 3.215297 2.019293 2.298135
##         10      11       12       13       14       15      16       17
## 1 1.131435 3.38853 2.914662 1.873499 2.076141 3.487895 2.29411 1.847588
##          18       19       20       21       22       23       24       25
## 1 0.7722224 3.485111 1.289646 2.987481 1.881477 3.231434 1.283191 1.630969
##         26       27      28       29       30       31       32       33
## 1 2.331727 2.662517 3.10243 3.561983 2.698023 1.599397 2.072368 1.604366
##         34       35       36       37       38      39        40       41
## 1 4.061499 2.269852 1.957087 2.370568 2.516134 3.39513 0.9157968 3.083559
##          42       43       44       45       46       47       48       49
## 1 0.8407489 1.646323 3.090601 3.979153 1.485973 2.648182 3.124347 3.504733
##         50
## 1 1.829103
```

Now, we will compute the pairwise correlation for all the points in the scaled dataset and then compute (1-correlation) for all the points.

```
##      [,1]      [,2]     [,3]       [,4]     [,5]     [,6]     [,7]     [,8]
## [1,]    0 0.7138308 1.446595 0.08774168 1.865922 1.687231 1.713587 1.142818
##           [,9]     [,10]    [,11]    [,12]    [,13]    [,14]    [,15]    [,16]
## [1,] 0.1049203 0.1162517 1.805901 1.479235 1.309096 1.474989 1.895387 1.769245
##          [,17]      [,18]    [,19]     [,20]    [,21]     [,22]    [,23]
## [1,] 0.2262068 0.04168043 1.244741 0.3068187 1.789492 0.7852281 1.938039
##             [,24]    [,25]     [,26]    [,27]    [,28]    [,29]    [,30]
## [1,] 0.0001410904 1.428062 0.3613202 1.955204 1.354691 1.805246 1.703448
##          [,31]    [,32]    [,33]    [,34]    [,35]    [,36]    [,37]    [,38]
## [1,] 0.4866251 1.440538 0.100542 1.832552 1.762949 1.916317 1.699757 1.577225
##         [,39]      [,40]     [,41]    [,42]     [,43]    [,44]     [,45]
## [1,] 1.548534 0.02912883 0.2970145 0.204103 0.9358083 1.993661 0.4044726
##          [,46]    [,47]      [,48]    [,49]     [,50]
## [1,] 0.3445471 1.956846 0.03840366 1.793068 0.3432853
```

Finally, we plot distances vs (1-correlation) for all the points and observe the relationship.



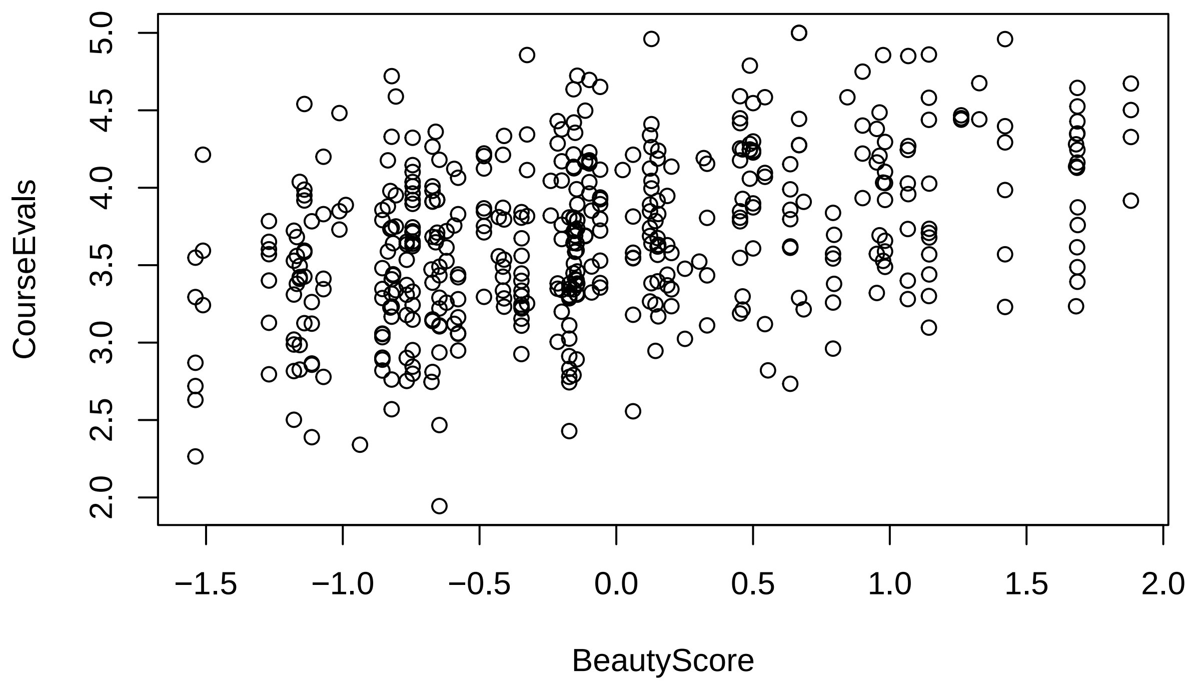
Clearly, the relationship is **perfectly linear**. Hence, the **proportionality holds true** for the two quantities.

Problem 1 | Beauty Pays!

Part 1

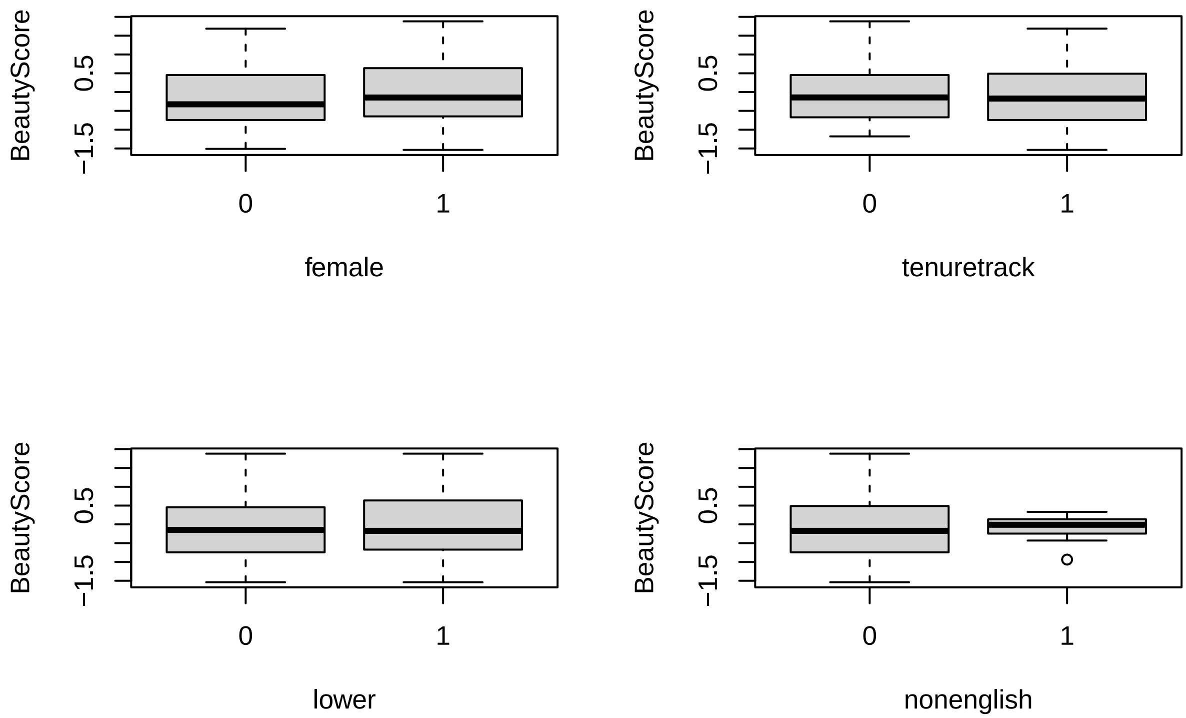
Let us first perform some exploratory data analysis on this dataset.

We will try to plot a scatterplot for CourseEvals and BeautyScore.



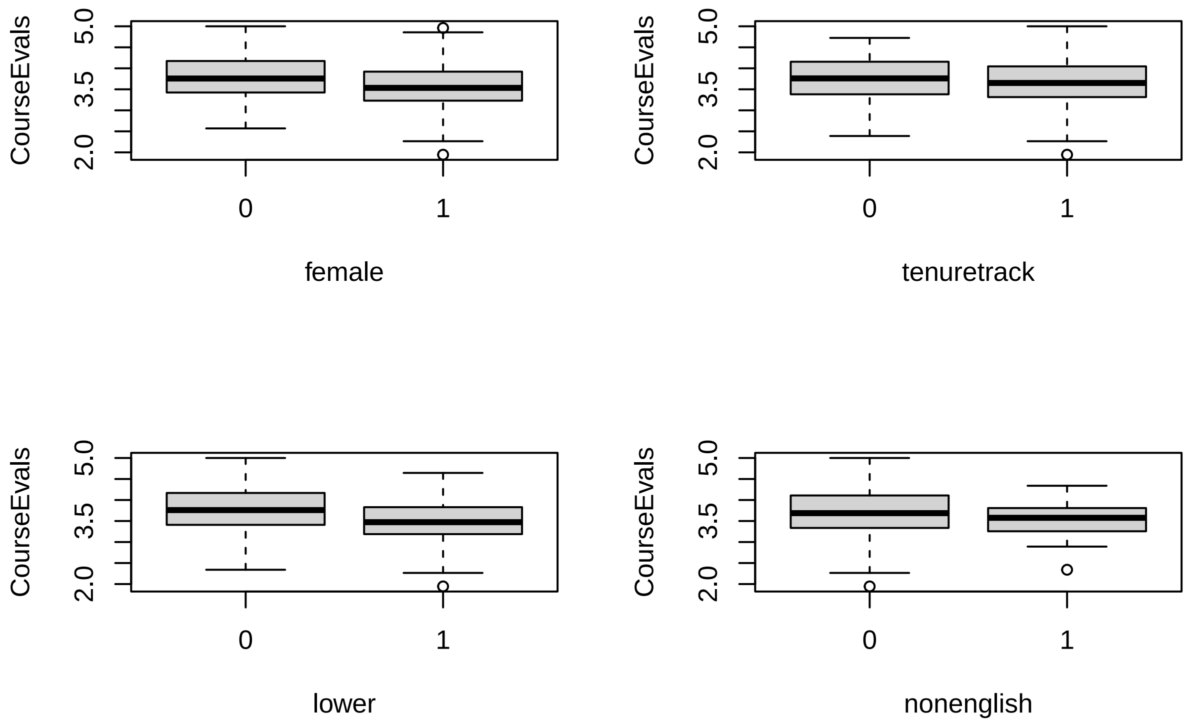
There seems to be some sort of a positive correlation between the two quantities.

Let us now explore if BeautyScore varies significantly across the different variables by using boxplots.



So, BeautyScore varies significantly across English and Non-English Speakers. There is some amount of difference in BeautyScores among Males and Females too (gender).

Now let us look if similar variances across variables occur for the variable CourseEvals too.



Here there is significant variance in almost all of the variables.

Let us now create a simple linear regression model for CourseEvals with only BeautyScore as a predictor.

```
##
## Call:
## lm(formula = CourseEvals ~ BeautyScore)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.5936 -0.3346  0.0097  0.3702  1.2321
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  3.71340    0.02249 165.119   <2e-16 ***
## BeautyScore  0.27148    0.02837   9.569   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4809 on 461 degrees of freedom
## Multiple R-squared:  0.1657, Adjusted R-squared:  0.1639
## F-statistic: 91.57 on 1 and 461 DF,  p-value: < 2.2e-16
```

The predictor BeautyScore seems significant as it has a very small p value. However, the adjusted R-squared is pretty low at only **0.1639.**

Now let us add the female variable and check if it improves the model.

```
##
## Call:
## lm(formula = CourseEvals ~ BeautyScore + female)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.40303 -0.29780  0.00792  0.31807  1.14350
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  3.84439    0.02833 135.706  < 2e-16 ***
## BeautyScore  0.29559    0.02720  10.869  < 2e-16 ***
## female      -0.30597    0.04339  -7.051 6.53e-12 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4573 on 460 degrees of freedom
## Multiple R-squared:  0.2471, Adjusted R-squared:  0.2438
## F-statistic: 75.49 on 2 and 460 DF,  p-value: < 2.2e-16
```

Here too both the predictor variables are significant with very low p values. The adjusted R-squared is **0.2438** which is more than the previous model. So, this model is a better fit than the last one.

We can check if BeautyScore and Female have any interactions by creating a new linear model.

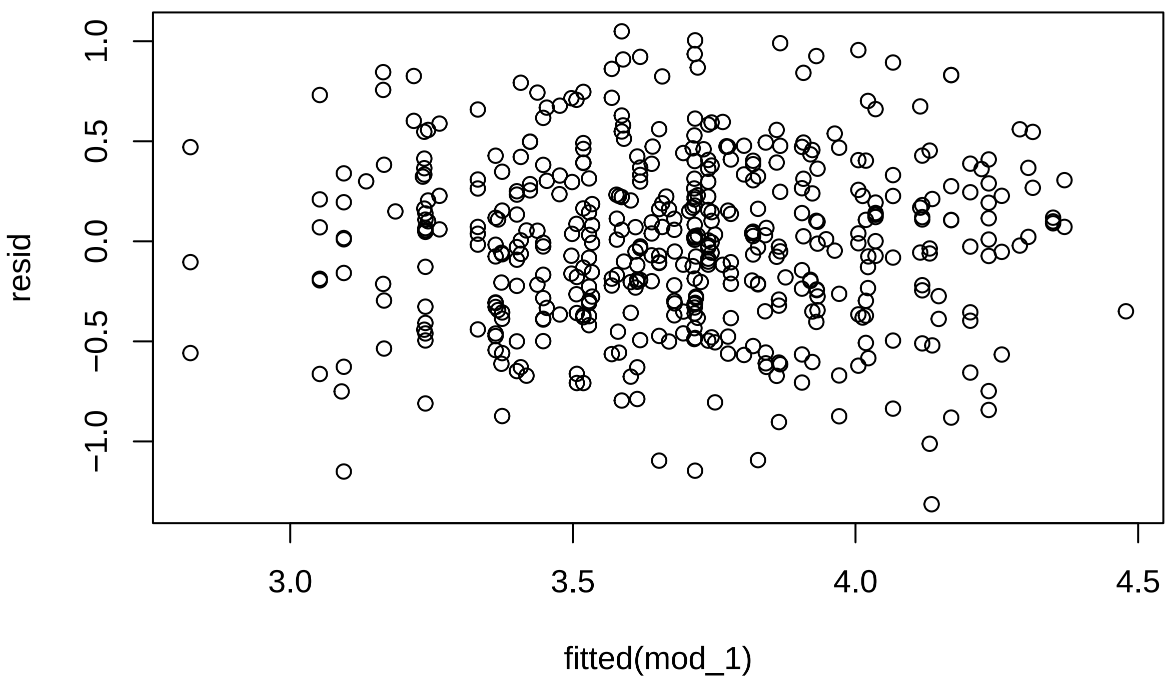
```
##
## Call:
## lm(formula = CourseEvals ~ BeautyScore + female + BeautyScore *
##     female)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.37133 -0.30235 -0.00191  0.31627  1.15215
##
## Coefficients:
##                    Estimate Std. Error t value Pr(>|t|)
## (Intercept)         3.83751    0.02861 134.133  < 2e-16 ***
## BeautyScore         0.25574    0.03690   6.930 1.43e-11 ***
## female             -0.30038    0.04346  -6.912 1.61e-11 ***
## BeautyScore:female  0.08685    0.05448   1.594    0.112
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4566 on 459 degrees of freedom
## Multiple R-squared:  0.2512, Adjusted R-squared:  0.2464
## F-statistic: 51.34 on 3 and 459 DF,  p-value: < 2.2e-16
```

The interaction term seems to have a **pretty high p-value**. So this term is **statistically insignificant** and hence can be discarded.

Let us try creating a linear model with all the predictor variables of the dataset and compare it with the previous models.

```
##
```

```
## Call:
## lm(formula = CourseEvals ~ BeautyScore + female + lower + nonenglish +
##     tenuretrack)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.31385 -0.30202  0.01011  0.29815  1.04929
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  4.06542    0.05145  79.020  < 2e-16 ***
## BeautyScore  0.30415    0.02543  11.959  < 2e-16 ***
## female      -0.33199    0.04075  -8.146 3.62e-15 ***
## lower       -0.34255    0.04282  -7.999 1.04e-14 ***
## nonenglish  -0.25808    0.08478  -3.044  0.00247 **
## tenuretrack -0.09945    0.04888  -2.035  0.04245 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4273 on 457 degrees of freedom
## Multiple R-squared:  0.3471, Adjusted R-squared:  0.3399
## F-statistic: 48.58 on 5 and 457 DF,  p-value: < 2.2e-16
```



- Here, clearly all the predictor variables seem to be **statistically significant** as they have **very small p values**. The **adjusted R-Squared of 0.3399** also seems to be an improvement over the previous model.

- The variables **female,lower,tenuretrack** seem to be **negatively correlated** with **CourseEvals**.
- The variable **BeautyScore seems** to be **positively correlated** with **CourseEvals**.
- As the variable **female** has a **negative coefficient**, so we can conclude that **females are usually scored lower than their male counterparts**.

Part 2

By the above statement, Professor Hamermesh means that it is difficult to ascertain whether correlation implies causation in this case.

While the possibility exists that people with higher BeautyScore are preferred during course evaluations, it could also mean that in the sample dataset we have taken, the people with higher beauty scores are more productive.

There is also the possibility of some other unknown variables influencing this correlation unbeknownst to us.

Problem 2 | Housing Price Structure

We first need to perform data preprocessing-

- We can drop the variable Home as it is only an index variable.
- We need to encode the variable Nbhd as it is categorical in nature.
- We also need to encode the binary categorical variable Brick.

Let us now fit a linear model with all the variables.

```
##
## Call:
## lm(formula = Price ~ . - nbhd_3, data = df1)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -27337.3  -6549.5    -41.7   5803.4  27359.3
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)    22840.536  10236.302   2.231  0.02752 *
## Offers         -8267.488   1084.777  -7.621 6.47e-12 ***
## SqFt              52.994      5.734   9.242 1.10e-15 ***
## Bedrooms        4246.794   1597.911   2.658  0.00894 **
## Bathrooms       7883.278   2117.035   3.724  0.00030 ***
## nbhd_1        -20681.037   3148.954  -6.568 1.38e-09 ***
## nbhd_2        -22241.616   2531.758  -8.785 1.32e-14 ***
## brick_encoded  17297.350   1981.616   8.729 1.78e-14 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 10020 on 120 degrees of freedom
## Multiple R-squared:  0.8686, Adjusted R-squared:  0.861
## F-statistic: 113.3 on 7 and 120 DF,  p-value: < 2.2e-16
```

Here are some of the preliminary observations-

- Every variable seems to be **significant** as their **p values are > 0.05**.
- We also have a **very high adjusted R-squared** value of **0.861**. So, we can use this model for our analysis.
- Also, we can see that the variable **nbhd_3** has been **removed** as it is a **singularity**.

Now we shall answer the given questions one by one.

Part 1

Given everything being equal, on average, a Brick house costs **$17297.3 more** than a non-brick house as per this model (as **17297.35** is the **coefficient of brick_encoded** in the linear regression model)

Part 2

As both **nbhd_1 and nbhd_2** have **negative coefficients (-20681.037 and -22241.616)**, this means that on average, on every other variable remaining the same, a house in Neighborhood 3 costs **$20681.037 more** than a house in Neighborhood 1 and **$22241.616 more** than a house in Neighborhood 2.

Part 3

As we are interested in analyzing the interaction of **nbhd_3** with **brick_encoded**, we need the variable nbhd_3 in our model which has been **dropped due to singularity**.

To resolve this, we can **drop nbhd_1** and instead **use nbhd_3**.

```
##
## Call:
## lm(formula = Price ~ SqFt + Bedrooms + Bathrooms + nbhd_2 + nbhd_3 +
##     brick_encoded + nbhd_3 * brick_encoded + Offers, data = df1)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -26939.1  -5428.7   -213.9   4519.3  26211.4
##
## Coefficients:
##                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)           3009.993   8706.264   0.346  0.73016
## SqFt                    54.065      5.636   9.593  < 2e-16 ***
## Bedrooms              4718.163   1577.613   2.991  0.00338 **
## Bathrooms             6463.365   2154.264   3.000  0.00329 **
## nbhd_2                -673.028   2376.477  -0.283  0.77751
## nbhd_3               17241.413   3391.347   5.084 1.39e-06 ***
## brick_encoded        13826.465   2405.556   5.748 7.11e-08 ***
## Offers               -8401.088   1064.370  -7.893 1.62e-12 ***
## nbhd_3:brick_encoded 10181.577   4165.274   2.444  0.01598 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

```
## Residual standard error: 9817 on 119 degrees of freedom
## Multiple R-squared:  0.8749, Adjusted R-squared:  0.8665
## F-statistic:   104 on 8 and 119 DF,  p-value: < 2.2e-16
```

Here, we can see that the **interaction term (nbhd_3 x brick_encoded)** is **significant** as it has a **very small p-value**. The **adjusted R-Square value of 0.8665** is also a **slight improvement** over the previous multiple linear regression model involving all predictor variables.

The coefficient of the interaction term tells us that on average, there is a **premium of $10181.577** on Brick Houses in **Neighbourhood 3**.

Part 4

To combine **nbhd_1 and nbhd_2** into one variable, we can **drop them both** and **keep only nbhd_3**. Whenever nbhd_3=1, it would mean that the Neighbourhood is Neighborhood 3. When nbhd_3=0, it would mean that it is either Neighborhood 1 or 2. This is exactly what we want for our analysis.

```
##
## Call:
## lm(formula = Price ~ SqFt + Bedrooms + Bathrooms + nbhd_3 + brick_encoded +
##     Offers, data = df1)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -26810.5  -5953.6   -266.5   5662.9  26793.0
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)    3067.471   8746.712   0.351 0.726423
## SqFt             52.149      5.572   9.359 5.44e-16 ***
## Bedrooms       4070.005   1570.921   2.591 0.010751 *
## Bathrooms      7810.698   2109.060   3.703 0.000322 ***
## nbhd_3        21937.572   2482.393   8.837 9.39e-15 ***
## brick_encoded 17058.771   1942.805   8.780 1.28e-14 ***
## Offers        -8019.003   1013.011  -7.916 1.32e-12 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 9995 on 121 degrees of freedom
## Multiple R-squared:  0.8682, Adjusted R-squared:  0.8616
## F-statistic: 132.8 on 6 and 121 DF,  p-value: < 2.2e-16
```

The **Adjusted R-Square** is **almost the same** as the original model. The **RSE** has, in fact, **reduced** for this model. So, combining Neighborhoods 1 and 2 into a single "older" Neighborhood might make this Linear Regression Model a **better model**.

Problem 3 | What causes what??

Part 1

Well, although it could be done, it may not be the most prudent approach to take as correlation may not imply causation. Also, the direction of the relationship between these relationships could be the other way around (i.e. more cops could have been deployed in the city by authorities due to high crime rate).

Also, there might be some other unknown variables and factors which might be impacting both the crime rate and the number of cops in the country. It would not be wise to make a definitive inference taking only crime and no. of cops into consideration.

Part 2

The Researchers at UPenn analyzed 'Orange Alert Days'. On 'Orange Alert Days', more cops are deployed in the DC Area due to the fear of terrorist attacks. By taking metro ridership levels as a substitute for the no. of victims, they ensured that the no. of victims is steady with respect to other days. This enabled them to reject the hypothesis that the fall in crime rate is because of decrease in the no. of victims.

Looking at the first table: on high alert days, the **coefficient is negative** which suggests a fall in crime rate. However, in this model, they did not control for the no. of victims.

Looking at the second table: in this model, they did control for the no. of victims. Even after controlling for this variable, the **coefficient remained negative** and the **relationship did not change much**. So, this indicates that **at the same level of metro ridership, crime rate fell on high alert days**.

Part 3

The researchers wanted to check if high alert days impacted crime rate given that the no. of victims have remained the same. They hence introduced the metro ridership attribute into the model so that it could act as a proxy for #Victims

Part 4

The researchers are attempting to estimate how the effect of high alert is different on district 1 and other districts.

The model can be interpreted as follow:- For District 1, **on high alert days, the crime rate is lower**. But for other districts, there seems to be **no significant relationship** between the crime rate and high alert days as the **p-values are high**.

Problem 5 | Final Project

I was primarily involved with running the various Tree models in our Group Project where we explored trips data from a Bike Share Company. Our project ran various regression models like Mulitple Linear Regression, Polynomial Regression, K Nearest Neighbors, Ridge Regression, Lasso Regression, Trees (Creating a Big Tree and then Pruning it), Bagging of Trees, Random Forests and Boosting of Trees.

Both me and Eric (Pengwei) were reponsible for running the Tree and other Ensemble Tree models on our data. We both ran our own separate models for Tree Pruning, Random Forests, Bagging and Boosting. We then combined our findings and results. It has to be noted that the results obtained from running both of our models were roughly similar.

For the presentation, I was also responsible for summarizing our findings and writing up a conclusion for the audience. I had to take a look at the various models run by our team and infer the relationship(s) between variables from the models and what it meant for the Bike Sharing Company and their business.