CDA541 Assignment2

Bhuvanendra Jonnalagadda, 50475864

2022-09-28

```
getwd()
```

```
## [1] "C:/Users/jonna/OneDrive/Desktop/CDA541 Statistical Data Mining 1/Assignments/Answers/Assignment1
```

```
setwd("C:/Users/jonna/OneDrive/Desktop/CDA541 Statistical Data Mining 1/Assignments/Answers/Assignment2

# Question 1:

library(ISLR)

data("College")

df = College

head(df, 10)
```

```
##                              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
## Albertus Magnus College          Yes  353    340    103        17        45
## Albion College                   Yes 1899   1720    489        37        68
## Albright College                 Yes 1038    839    227        30        63
## Alderson-Broaddus College        Yes  582    498    172        21        44
##                              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
## Albertus Magnus College              416         230    13290       5720   500
## Albion College                      1594          32    13868       4826   450
## Albright College                     973         306    15595       4400   300
## Alderson-Broaddus College            799          78    10468       3380   660
##                              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
## Albertus Magnus College          1500  90       93      11.5          26   8861
## Albion College                    850  89      100      13.7          37  11487
## Albright College                  500  79       84      11.3          23  11644
## Alderson-Broaddus College        1800  40       41      11.5          15   8991
##                                Grad.Rate
## Abilene Christian University          60
## Adelphi University                    56
## Adrian College                        54
## Agnes Scott College                   59
## Alaska Pacific University             15
## Albertson College                     55
## Albertus Magnus College               63
## Albion College                        73
## Albright College                      80
## Alderson-Broaddus College             52
```

```
#View(College)
dim(df)     #College data is very small dataset
```

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

```
str(df)
```

```
## 'data.frame':    777 obs. of  18 variables:
##  $ Private    : Factor w/ 2 levels "No","Yes": 2 2 2 2 2 2 2 2 2 2 ...
##  $ Apps       : num  1660 2186 1428 417 193 ...
##  $ Accept     : num  1232 1924 1097 349 146 ...
##  $ Enroll     : num  721 512 336 137 55 158 103 489 227 172 ...
##  $ Top10perc  : num  23 16 22 60 16 38 17 37 30 21 ...
##  $ Top25perc  : num  52 29 50 89 44 62 45 68 63 44 ...
##  $ F.Undergrad: num  2885 2683 1036 510 249 ...
##  $ P.Undergrad: num  537 1227 99 63 869 ...
##  $ Outstate   : num  7440 12280 11250 12960 7560 ...
##  $ Room.Board : num  3300 6450 3750 5450 4120 ...
##  $ Books      : num  450 750 400 450 800 500 500 450 300 660 ...
##  $ Personal   : num  2200 1500 1165 875 1500 ...
##  $ PhD        : num  70 29 53 92 76 67 90 89 79 40 ...
##  $ Terminal   : num  78 30 66 97 72 73 93 100 84 41 ...
##  $ S.F.Ratio  : num  18.1 12.2 12.9 7.7 11.9 9.4 11.5 13.7 11.3 11.5 ...
##  $ perc.alumni: num  12 16 30 37 2 11 26 37 23 15 ...
##  $ Expend     : num  7041 10527 8735 19016 10922 ...
##  $ Grad.Rate  : num  60 56 54 59 15 55 63 73 80 52 ...
```

```
summary(df)
```

```
##  Private        Apps           Accept          Enroll       Top10perc
##  No :212   Min.   :   81   Min.   :   72   Min.   :  35   Min.   : 1.00
##  Yes:565   1st Qu.:  776   1st Qu.:  604   1st Qu.: 242   1st Qu.:15.00
```

```
##            Median : 1558   Median : 1110   Median : 434   Median :23.00
##            Mean   : 3002   Mean   : 2019   Mean   : 780   Mean   :27.56
##            3rd Qu.: 3624   3rd Qu.: 2424   3rd Qu.: 902   3rd Qu.:35.00
##            Max.   :48094   Max.   :26330   Max.   :6392   Max.   :96.00
##    Top25perc      F.Undergrad     P.Undergrad         Outstate
##  Min.   :  9.0   Min.   :  139   Min.   :    1.0   Min.   : 2340
##  1st Qu.: 41.0   1st Qu.:  992   1st Qu.:   95.0   1st Qu.: 7320
##  Median : 54.0   Median : 1707   Median :  353.0   Median : 9990
##  Mean   : 55.8   Mean   : 3700   Mean   :  855.3   Mean   :10441
##  3rd Qu.: 69.0   3rd Qu.: 4005   3rd Qu.:  967.0   3rd Qu.:12925
##  Max.   :100.0   Max.   :31643   Max.   :21836.0   Max.   :21700
##    Room.Board       Books           Personal         PhD
##  Min.   :1780   Min.   :  96.0   Min.   : 250   Min.   :  8.00
##  1st Qu.:3597   1st Qu.: 470.0   1st Qu.: 850   1st Qu.: 62.00
##  Median :4200   Median : 500.0   Median :1200   Median : 75.00
##  Mean   :4358   Mean   : 549.4   Mean   :1341   Mean   : 72.66
##  3rd Qu.:5050   3rd Qu.: 600.0   3rd Qu.:1700   3rd Qu.: 85.00
##  Max.   :8124   Max.   :2340.0   Max.   :6800   Max.   :103.00
##     Terminal       S.F.Ratio      perc.alumni        Expend
##  Min.   : 24.0   Min.   : 2.50   Min.   : 0.00   Min.   : 3186
##  1st Qu.: 71.0   1st Qu.:11.50   1st Qu.:13.00   1st Qu.: 6751
##  Median : 82.0   Median :13.60   Median :21.00   Median : 8377
##  Mean   : 79.7   Mean   :14.09   Mean   :22.74   Mean   : 9660
##  3rd Qu.: 92.0   3rd Qu.:16.50   3rd Qu.:31.00   3rd Qu.:10830
##  Max.   :100.0   Max.   :39.80   Max.   :64.00   Max.   :56233
##    Grad.Rate
##  Min.   : 10.00
##  1st Qu.: 53.00
##  Median : 65.00
##  Mean   : 65.46
##  3rd Qu.: 78.00
##  Max.   :118.00
```

```
#checking missing values?
sum(is.na(df))  #no missing values!!
```

```
## [1] 0
```

```
set.seed(41)
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.2 --
## v ggplot2 3.3.6      v purrr   0.3.4
## v tibble  3.1.8      v dplyr   1.0.10
## v tidyr   1.2.0      v stringr 1.4.1
## v readr   2.1.2      v forcats 0.5.2
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(caret)
```

```
## Loading required package: lattice
##
## Attaching package: 'caret'
##
## The following object is masked from 'package:purrr':
##
##     lift
```

```
#(a) Split data into train-test:
training_samples = df$Apps %>%
  createDataPartition(p = .8, list = FALSE)

train_data = df[training_samples, ]
test_data = df[-training_samples, ]

linear_model = train(Apps ~ ., data = train_data, method = 'lm')
summary(linear_model)
```

```
##
## Call:
## lm(formula = .outcome ~ ., data = dat)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3553.5  -430.8     5.5   332.4  6633.8
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -140.50424  439.91427  -0.319 0.749541
## PrivateYes  -361.06483  145.79454  -2.477 0.013538 *
## Accept         1.69530    0.04225  40.125  < 2e-16 ***
## Enroll        -0.91987    0.19100  -4.816 1.85e-06 ***
## Top10perc     51.26505    5.81408   8.817  < 2e-16 ***
## Top25perc    -14.40648    4.54052  -3.173 0.001586 **
## F.Undergrad    0.03026    0.03336   0.907 0.364743
## P.Undergrad    0.06994    0.03177   2.202 0.028064 *
## Outstate      -0.08037    0.02051  -3.919 9.91e-05 ***
## Room.Board     0.12229    0.05018   2.437 0.015095 *
## Books          0.38317    0.29294   1.308 0.191353
## Personal      -0.05594    0.06604  -0.847 0.397312
## PhD           -8.18262    4.80209  -1.704 0.088899 .
## Terminal      -3.41339    5.27560  -0.647 0.517868
## S.F.Ratio      5.04763   14.04510   0.359 0.719430
## perc.alumni   -3.95426    4.35154  -0.909 0.363867
## Expend         0.05249    0.01389   3.779 0.000173 ***
## Grad.Rate      7.10749    3.17647   2.238 0.025613 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 981.2 on 606 degrees of freedom
## Multiple R-squared:  0.9378, Adjusted R-squared:  0.936
## F-statistic:   537 on 17 and 606 DF,  p-value: < 2.2e-16
```

```r
# The model is is good as R-squared and adjusted R-squared is higher and
# p-value is comparitively lower overall.

pred = predict(linear_model, newdata = test_data)

compare = data.frame(actual = test_data$Apps, predicted = pred)
head(compare,10)
```

```
##                                       actual predicted
## Albertson College                        587  828.1704
## Albion College                          1899 2507.7977
## Albright College                        1038  977.6855
## Allentown Coll. of St. Francis de Sales 1179 1788.4226
## Amherst College                         4302 3472.9640
## Appalachian State University            7313 6023.8322
## Aquinas College                          619  302.1167
## Augustana College                        761  440.7268
## Belmont University                      1220 1321.8190
## Bentley College                         3466 3137.7093
```

```r
# Error obtained:
linear_model$results
```

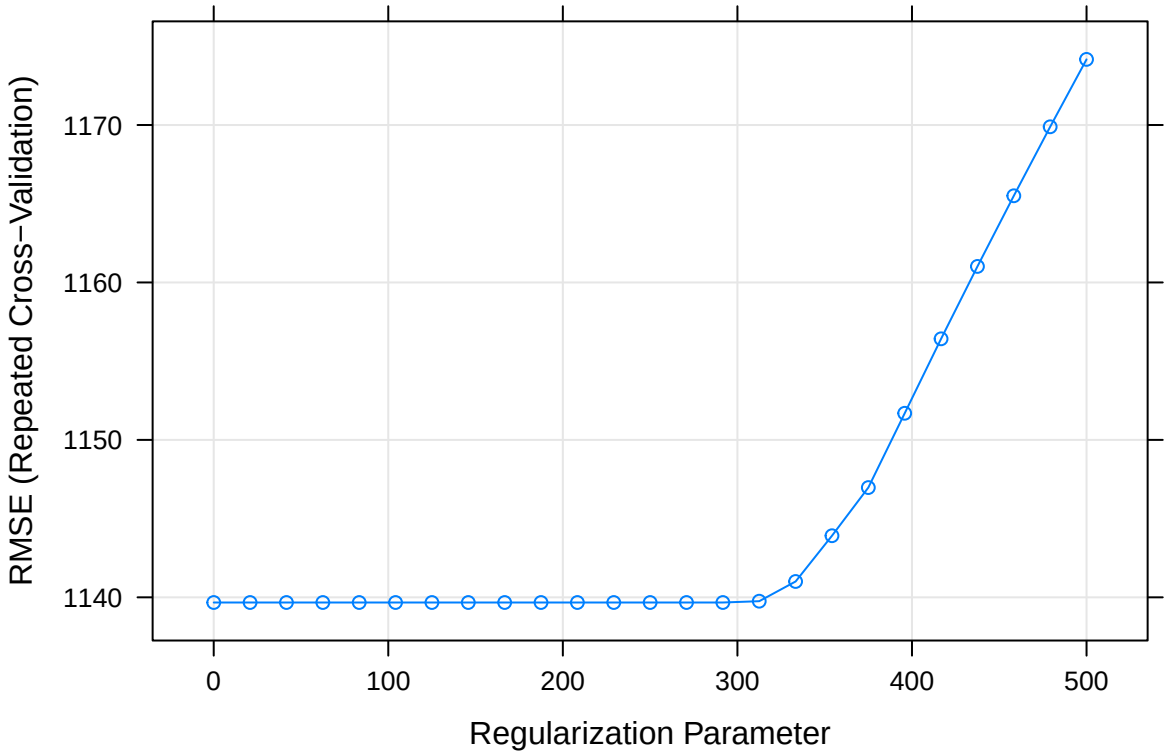
```
##   intercept     RMSE  Rsquared      MAE   RMSESD RsquaredSD    MAESD
## 1      TRUE 1084.414 0.9177633 624.7942 161.9587 0.02336535 44.61854
```

```r
#(b) Ridge model

# Parameter tuning:
param_control = trainControl(method = 'repeatedcv', number = 10, repeats = 5)

ridge_reg_model = train(Apps ~ ., data = train_data, method = 'glmnet',
                        trControl = param_control,
                        tuneGrid = expand.grid(alpha =0,
                                               lambda = seq(0.01, 500,
                                                            length = 25)))

ridge_reg_model
```

```
## glmnet
##
## 624 samples
##  17 predictor
##
## No pre-processing
## Resampling: Cross-Validated (10 fold, repeated 5 times)
## Summary of sample sizes: 563, 562, 560, 562, 562, 562, ...
## Resampling results across tuning parameters:
##
##   lambda     RMSE      Rsquared   MAE
##     0.01000  1139.672  0.9225831  629.3541
##    20.84292  1139.672  0.9225831  629.3541
```

```
##    41.67583  1139.672  0.9225831  629.3541
##    62.50875  1139.672  0.9225831  629.3541
##    83.34167  1139.672  0.9225831  629.3541
##   104.17458  1139.672  0.9225831  629.3541
##   125.00750  1139.672  0.9225831  629.3541
##   145.84042  1139.672  0.9225831  629.3541
##   166.67333  1139.672  0.9225831  629.3541
##   187.50625  1139.672  0.9225831  629.3541
##   208.33917  1139.672  0.9225831  629.3541
##   229.17208  1139.672  0.9225831  629.3541
##   250.00500  1139.672  0.9225831  629.3541
##   270.83792  1139.672  0.9225831  629.3541
##   291.67083  1139.672  0.9225831  629.3541
##   312.50375  1139.755  0.9225788  629.3674
##   333.33667  1141.006  0.9224979  629.6288
##   354.16958  1143.913  0.9222871  630.2176
##   375.00250  1146.973  0.9220584  630.9788
##   395.83542  1151.690  0.9215227  632.9437
##   416.66833  1156.419  0.9209637  634.9639
##   437.50125  1161.018  0.9204172  636.9721
##   458.33417  1165.506  0.9198757  639.0429
##   479.16708  1169.887  0.9193461  641.1631
##   500.00000  1174.170  0.9188207  643.3160
##
## Tuning parameter 'alpha' was held constant at a value of 0
## RMSE was used to select the optimal model using the smallest value.
## The final values used for the model were alpha = 0 and lambda = 291.6708.
```

```
plot(ridge_reg_model)
```




```
# Error obtained:
ridge_reg_model$results
```

```
##    alpha    lambda     RMSE  Rsquared      MAE   RMSESD RsquaredSD    MAESD
## 1      0   0.01000 1139.672 0.9225831 629.3541 574.8718 0.02763403 138.6862
## 2      0  20.84292 1139.672 0.9225831 629.3541 574.8718 0.02763403 138.6862
## 3      0  41.67583 1139.672 0.9225831 629.3541 574.8718 0.02763403 138.6862
## 4      0  62.50875 1139.672 0.9225831 629.3541 574.8718 0.02763403 138.6862
## 5      0  83.34167 1139.672 0.9225831 629.3541 574.8718 0.02763403 138.6862
## 6      0 104.17458 1139.672 0.9225831 629.3541 574.8718 0.02763403 138.6862
## 7      0 125.00750 1139.672 0.9225831 629.3541 574.8718 0.02763403 138.6862
## 8      0 145.84042 1139.672 0.9225831 629.3541 574.8718 0.02763403 138.6862
## 9      0 166.67333 1139.672 0.9225831 629.3541 574.8718 0.02763403 138.6862
## 10     0 187.50625 1139.672 0.9225831 629.3541 574.8718 0.02763403 138.6862
## 11     0 208.33917 1139.672 0.9225831 629.3541 574.8718 0.02763403 138.6862
## 12     0 229.17208 1139.672 0.9225831 629.3541 574.8718 0.02763403 138.6862
## 13     0 250.00500 1139.672 0.9225831 629.3541 574.8718 0.02763403 138.6862
## 14     0 270.83792 1139.672 0.9225831 629.3541 574.8718 0.02763403 138.6862
## 15     0 291.67083 1139.672 0.9225831 629.3541 574.8718 0.02763403 138.6862
## 16     0 312.50375 1139.755 0.9225788 629.3674 575.1176 0.02763389 138.7194
## 17     0 333.33667 1141.006 0.9224979 629.6288 578.7662 0.02768299 139.3744
## 18     0 354.16958 1143.913 0.9222871 630.2176 586.9609 0.02787854 140.7681
## 19     0 375.00250 1146.973 0.9220584 630.9788 594.7475 0.02806016 142.1225
## 20     0 395.83542 1151.690 0.9215227 632.9437 601.6554 0.02829395 143.2623
## 21     0 416.66833 1156.419 0.9209637 634.9639 608.0993 0.02852764 144.2892
## 22     0 437.50125 1161.018 0.9204172 636.9721 614.3084 0.02876196 145.2527
```

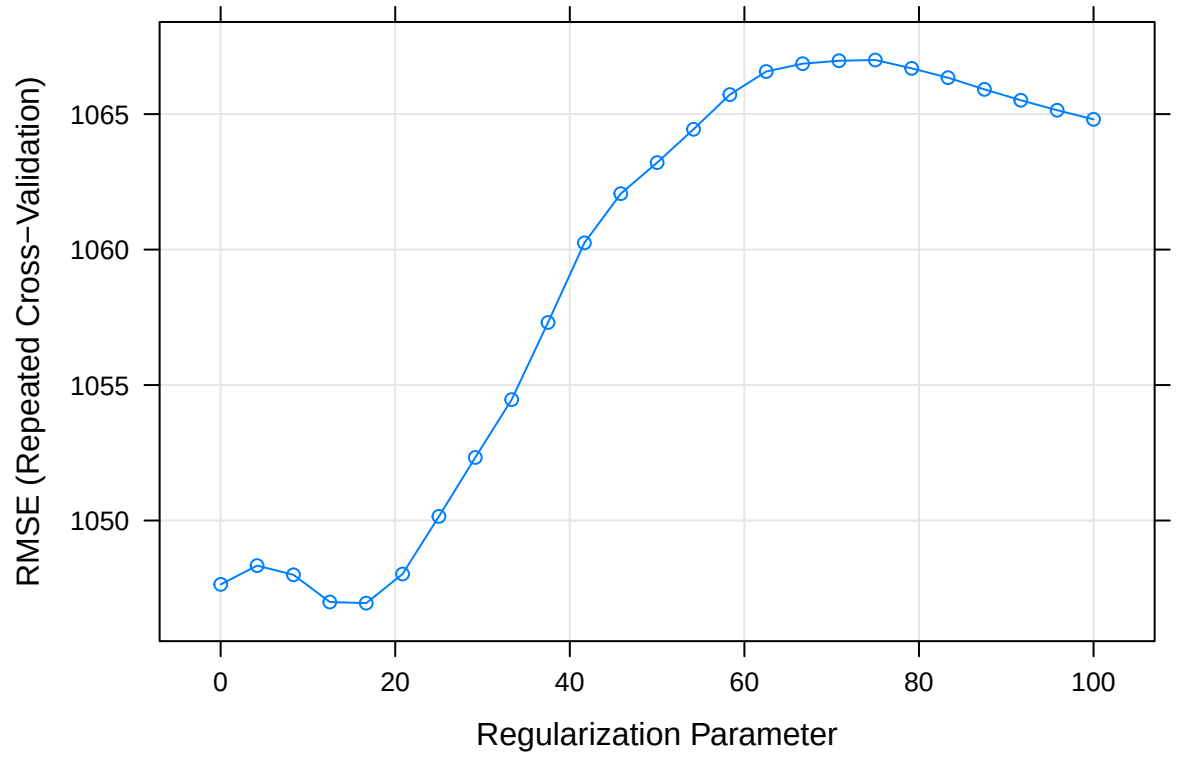
```
## 23     0 458.33417 1165.506 0.9198757 639.0429 620.1173 0.02899038 146.0648
## 24     0 479.16708 1169.887 0.9193461 641.1631 625.7501 0.02921701 146.8311
## 25     0 500.00000 1174.170 0.9188207 643.3160 631.0082 0.02943463 147.5012
```

```
#(d) Lasso Model

# Parameter tuning:
param_control = trainControl(method = 'repeatedcv', number = 10, repeats = 5)

lasso_model = train(Apps ~ ., data = train_data, method = 'glmnet',
                    trControl = param_control,
                    tuneGrid = expand.grid(alpha = 1,
                                           lambda = seq(0.01, 100, length = 25)))

lasso_model
```

```
## glmnet
##
## 624 samples
##  17 predictor
##
## No pre-processing
## Resampling: Cross-Validated (10 fold, repeated 5 times)
## Summary of sample sizes: 562, 562, 561, 560, 562, 560, ...
## Resampling results across tuning parameters:
##
##   lambda     RMSE      Rsquared   MAE
##     0.01000  1047.642  0.9261715  620.3737
##     4.17625  1048.336  0.9265154  617.9576
##     8.34250  1047.996  0.9271092  612.7034
##    12.50875  1046.992  0.9273916  608.6341
##    16.67500  1046.951  0.9275069  605.6637
##    20.84125  1048.024  0.9274511  604.2247
##    25.00750  1050.152  0.9272321  603.9242
##    29.17375  1052.327  0.9270309  603.7412
##    33.34000  1054.463  0.9268072  603.4531
##    37.50625  1057.308  0.9264477  603.5653
##    41.67250  1060.247  0.9260361  603.8921
##    45.83875  1062.064  0.9257121  604.0470
##    50.00500  1063.210  0.9254380  604.0408
##    54.17125  1064.440  0.9251179  604.2864
##    58.33750  1065.720  0.9247961  604.4518
##    62.50375  1066.572  0.9245461  604.1842
##    66.67000  1066.861  0.9243869  603.2313
##    70.83625  1066.969  0.9242531  601.9308
##    75.00250  1066.997  0.9241343  600.4707
##    79.16875  1066.689  0.9240769  598.7916
##    83.33500  1066.344  0.9240329  597.1175
##    87.50125  1065.912  0.9240112  595.4618
##    91.66750  1065.511  0.9239893  593.8584
##    95.83375  1065.143  0.9239696  592.3128
##   100.00000  1064.806  0.9239458  590.8431
##
## Tuning parameter 'alpha' was held constant at a value of 1
```

```
## RMSE was used to select the optimal model using the smallest value.
## The final values used for the model were alpha = 1 and lambda = 16.675.
```

```
plot(lasso_model)
```




```
# Error obtained:
lasso_model$results
```

```
##    alpha    lambda     RMSE  Rsquared      MAE   RMSESD RsquaredSD     MAESD
## 1      1   0.01000 1047.642 0.9261715 620.3737 315.6468 0.03638406  96.33569
## 2      1   4.17625 1048.336 0.9265154 617.9576 323.6362 0.03614373  97.67973
## 3      1   8.34250 1047.996 0.9271092 612.7034 334.9202 0.03591728  99.58732
## 4      1  12.50875 1046.992 0.9273916 608.6341 338.9001 0.03609402 100.24381
## 5      1  16.67500 1046.951 0.9275069 605.6637 342.1308 0.03632816 100.80668
## 6      1  20.84125 1048.024 0.9274511 604.2247 344.9012 0.03660692 100.92185
## 7      1  25.00750 1050.152 0.9272321 603.9242 347.1707 0.03692506 100.81986
## 8      1  29.17375 1052.327 0.9270309 603.7412 349.2721 0.03712216 100.37324
## 9      1  33.34000 1054.463 0.9268072 603.4531 350.9905 0.03733698 100.23369
## 10     1  37.50625 1057.308 0.9264477 603.5653 352.0353 0.03757289 100.29972
## 11     1  41.67250 1060.247 0.9260361 603.8921 352.6131 0.03780458 100.29463
## 12     1  45.83875 1062.064 0.9257121 604.0470 353.3281 0.03804436 100.44114
## 13     1  50.00500 1063.210 0.9254380 604.0408 354.2745 0.03833378 100.65541
## 14     1  54.17125 1064.440 0.9251179 604.2864 355.0614 0.03867130 100.75997
## 15     1  58.33750 1065.720 0.9247961 604.4518 355.7562 0.03901040 100.46723
## 16     1  62.50375 1066.572 0.9245461 604.1842 356.4293 0.03929175 100.14497
```

```
## 17     1  66.67000 1066.861 0.9243869 603.2313 357.2136 0.03951077 100.02475
## 18     1  70.83625 1066.969 0.9242531 601.9308 357.9744 0.03970588 100.06295
## 19     1  75.00250 1066.997 0.9241343 600.4707 358.7125 0.03988474 100.14423
## 20     1  79.16875 1066.689 0.9240769 598.7916 359.5973 0.04003056 100.33136
## 21     1  83.33500 1066.344 0.9240329 597.1175 360.5758 0.04016255 100.56339
## 22     1  87.50125 1065.912 0.9240112 595.4618 361.6555 0.04029357 100.83299
## 23     1  91.66750 1065.511 0.9239893 593.8584 362.7362 0.04042383 101.09296
## 24     1  95.83375 1065.143 0.9239696 592.3128 363.8030 0.04055760 101.34060
## 25     1 100.00000 1064.806 0.9239458 590.8431 364.8697 0.04069219 101.64276
```

```
# no.of non-zero estimates:
coef(lasso_model$finalModel, lasso_model$bestTune$lambda)
```

```
## 18 x 1 sparse Matrix of class "dgCMatrix"
##                          s1
## (Intercept) -387.93379120
## PrivateYes  -293.06863509
## Accept         1.59997966
## Enroll        -0.47956774
## Top10perc     39.09071464
## Top25perc     -5.88673816
## F.Undergrad    .
## P.Undergrad    0.04591347
## Outstate      -0.05512975
## Room.Board     0.10410870
## Books          0.24493022
## Personal      -0.01786759
## PhD           -5.78518170
## Terminal      -3.19376641
## S.F.Ratio      .
## perc.alumni   -3.91296652
## Expend         0.04534940
## Grad.Rate      4.41247925
```

```
#(g) Which is best?

cbind(linear_model$results$Rsquared,
      ridge_reg_model$results$Rsquared,
      lasso_model$results$Rsquared)
```

```
##            [,1]      [,2]      [,3]
##  [1,] 0.9177633 0.9225831 0.9261715
##  [2,] 0.9177633 0.9225831 0.9265154
##  [3,] 0.9177633 0.9225831 0.9271092
##  [4,] 0.9177633 0.9225831 0.9273916
##  [5,] 0.9177633 0.9225831 0.9275069
##  [6,] 0.9177633 0.9225831 0.9274511
##  [7,] 0.9177633 0.9225831 0.9272321
##  [8,] 0.9177633 0.9225831 0.9270309
##  [9,] 0.9177633 0.9225831 0.9268072
## [10,] 0.9177633 0.9225831 0.9264477
## [11,] 0.9177633 0.9225831 0.9260361
## [12,] 0.9177633 0.9225831 0.9257121
```

```
## [13,] 0.9177633 0.9225831 0.9254380
## [14,] 0.9177633 0.9225831 0.9251179
## [15,] 0.9177633 0.9225831 0.9247961
## [16,] 0.9177633 0.9225788 0.9245461
## [17,] 0.9177633 0.9224979 0.9243869
## [18,] 0.9177633 0.9222871 0.9242531
## [19,] 0.9177633 0.9220584 0.9241343
## [20,] 0.9177633 0.9215227 0.9240769
## [21,] 0.9177633 0.9209637 0.9240329
## [22,] 0.9177633 0.9204172 0.9240112
## [23,] 0.9177633 0.9198757 0.9239893
## [24,] 0.9177633 0.9193461 0.9239696
## [25,] 0.9177633 0.9188207 0.9239458
```

```
cbind(linear_model$results$RMSE,
      ridge_reg_model$results$RMSE,
      lasso_model$results$RMSE)
```

```
##           [,1]     [,2]     [,3]
##  [1,] 1084.414 1139.672 1047.642
##  [2,] 1084.414 1139.672 1048.336
##  [3,] 1084.414 1139.672 1047.996
##  [4,] 1084.414 1139.672 1046.992
##  [5,] 1084.414 1139.672 1046.951
##  [6,] 1084.414 1139.672 1048.024
##  [7,] 1084.414 1139.672 1050.152
##  [8,] 1084.414 1139.672 1052.327
##  [9,] 1084.414 1139.672 1054.463
## [10,] 1084.414 1139.672 1057.308
## [11,] 1084.414 1139.672 1060.247
## [12,] 1084.414 1139.672 1062.064
## [13,] 1084.414 1139.672 1063.210
## [14,] 1084.414 1139.672 1064.440
## [15,] 1084.414 1139.672 1065.720
## [16,] 1084.414 1139.755 1066.572
## [17,] 1084.414 1141.006 1066.861
## [18,] 1084.414 1143.913 1066.969
## [19,] 1084.414 1146.973 1066.997
## [20,] 1084.414 1151.690 1066.689
## [21,] 1084.414 1156.419 1066.344
## [22,] 1084.414 1161.018 1065.912
## [23,] 1084.414 1165.506 1065.511
## [24,] 1084.414 1169.887 1065.143
## [25,] 1084.414 1174.170 1064.806
```

```
# Based on the error values and accuracy of all 3 models,
# I would say, all models performed very well.
```

```
##############################################################################
# Question 2:
##############################################################################

df = read.csv("caravan-insurance-challenge.csv")
#head(df)
#str(df)   #origin column is 'str'.
#summary(df)

train_df = subset(df, ORIGIN == 'train')
test_df = subset(df, ORIGIN == 'test')
train_df = subset(train_df, select = -c(ORIGIN))
test_df = subset(test_df, select = -c(ORIGIN))
#str(train_df)
#str(test_df)

dim(train_df)
```

```
## [1] 5822   86
```

```
dim(test_df)
```

```
## [1] 4000   86
```

```
#missing values?
sum(is.na(train_df))  #no missing values
```

```
## [1] 0
```

```
sum(is.na(test_df))   #no missing values
```

```
## [1] 0
```

```
set.seed(41)

############
# OLS model:
############

ols_model = train(CARAVAN ~ ., data = train_df, method = 'lm')
```

```
## Warning in train.default(x, y, weights = w, ...): You are trying to do
## regression and your outcome only has two possible values Are you trying to do
## classification? If so, use a 2 level factor as your outcome column.

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
## may be misleading
```

```
## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
## may be misleading
```

```
summary(ols_model)
```

```
##
## Call:
## lm(formula = .outcome ~ ., data = dat)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.67293 -0.08720 -0.04593 -0.00639  1.04628
##
## Coefficients:
```

```
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.7685381  0.4298406   1.788 0.073835 .
## MOSTYPE      0.0035209  0.0022512   1.564 0.117866
## MAANTHUI    -0.0072642  0.0076739  -0.947 0.343875
## MGEMOMV     -0.0012739  0.0071737  -0.178 0.859055
## MGEMLEEF     0.0107473  0.0049596   2.167 0.030279 *
## MOSHOOFD    -0.0154869  0.0101044  -1.533 0.125405
## MGODRK      -0.0056016  0.0056016  -1.000 0.317353
## MGODPR      -0.0002069  0.0060664  -0.034 0.972795
## MGODOV       0.0003569  0.0054592   0.065 0.947874
## MGODGE      -0.0030237  0.0058038  -0.521 0.602399
## MRELGE       0.0086829  0.0075479   1.150 0.250036
## MRELSA       0.0020367  0.0072008   0.283 0.777310
## MRELOV       0.0055682  0.0076295   0.730 0.465526
## MFALLEEN    -0.0038250  0.0065474  -0.584 0.559107
## MFGEKIND    -0.0050625  0.0066861  -0.757 0.448980
## MFWEKIND    -0.0026253  0.0069795  -0.376 0.706824
## MOPLHOOG     0.0021357  0.0068161   0.313 0.754038
## MOPLMIDD    -0.0048456  0.0071396  -0.679 0.497358
## MOPLLAAG    -0.0113977  0.0073004  -1.561 0.118525
## MBERHOOG     0.0021884  0.0045182   0.484 0.628153
## MBERZELF    -0.0004665  0.0052201  -0.089 0.928796
## MBERBOER    -0.0050974  0.0050426  -1.011 0.312122
## MBERMIDD     0.0041254  0.0044806   0.921 0.357228
## MBERARBG    -0.0006060  0.0044709  -0.136 0.892190
## MBERARBO     0.0019733  0.0044532   0.443 0.657690
## MSKA        -0.0013674  0.0051653  -0.265 0.791225
## MSKB1       -0.0031701  0.0050198  -0.632 0.527724
## MSKB2       -0.0012603  0.0044827  -0.281 0.778603
## MSKC         0.0024879  0.0049115   0.507 0.612502
## MSKD        -0.0008866  0.0047145  -0.188 0.850832
## MHHUUR      -0.0454201  0.0376622  -1.206 0.227872
## MHKOOP      -0.0432242  0.0376290  -1.149 0.250730
## MAUT1        0.0085964  0.0075592   1.137 0.255502
## MAUT2        0.0077871  0.0068554   1.136 0.256038
## MAUT0        0.0047215  0.0072646   0.650 0.515762
## MZFONDS     -0.0561024  0.0444643  -1.262 0.207094
## MZPART      -0.0593733  0.0443897  -1.338 0.181097
## MINKM30      0.0070879  0.0051150   1.386 0.165884
## MINK3045     0.0069414  0.0049276   1.409 0.158986
## MINK4575     0.0049679  0.0050144   0.991 0.321862
## MINK7512     0.0059267  0.0052728   1.124 0.261053
## MINK123M    -0.0098939  0.0069270  -1.428 0.153258
## MINKGEM      0.0063044  0.0045645   1.381 0.167277
## MKOOPKLA     0.0029097  0.0022664   1.284 0.199250
## PWAPART      0.0284931  0.0166017   1.716 0.086166 .
## PWABEDR     -0.0101533  0.0205121  -0.495 0.620625
## PWALAND     -0.0201220  0.0390424  -0.515 0.606301
## PPERSAUT     0.0102787  0.0026346   3.901 9.67e-05 ***
## PBESAUT      0.0014405  0.0148574   0.097 0.922765
## PMOTSCO     -0.0061279  0.0079415  -0.772 0.440364
## PVRAAUT     -0.0249190  0.0415892  -0.599 0.549083
## PAANHANG     0.0588044  0.0557610   1.055 0.291662
## PTRACTOR     0.0121481  0.0142358   0.853 0.393504
```

```
## PWERKT      -0.0062440  0.0370186  -0.169 0.866060
## PBROM        0.0078683  0.0152793   0.515 0.606598
## PLEVEN      -0.0155397  0.0064753  -2.400 0.016433 *
## PPERSONG     0.0098926  0.0335157   0.295 0.767880
## PGEZONG      0.1937254  0.0793370   2.442 0.014644 *
## PWAOREG      0.0647933  0.0256913   2.522 0.011696 *
## PBRAND       0.0132643  0.0035906   3.694 0.000223 ***
## PZEILPL     -0.1917507  0.1439848  -1.332 0.182998
## PPLEZIER    -0.0299076  0.0269224  -1.111 0.266666
## PFIETS      -0.0107777  0.0549693  -0.196 0.844564
## PINBOED     -0.0441620  0.0307404  -1.437 0.150883
## PBYSTAND    -0.0184858  0.0288890  -0.640 0.522269
## AWAPART     -0.0377952  0.0323794  -1.167 0.243154
## AWABEDR      0.0185448  0.0529740   0.350 0.726296
## AWALAND      0.0180904  0.1374585   0.132 0.895300
## APERSAUT     0.0002821  0.0127496   0.022 0.982347
## ABESAUT     -0.0214816  0.0652955  -0.329 0.742175
## AMOTSCO      0.0203252  0.0310683   0.654 0.513004
## AVRAAUT      0.0563675  0.1589388   0.355 0.722866
## AAANHANG    -0.0804238  0.0944352  -0.852 0.394455
## ATRACTOR    -0.0395651  0.0353795  -1.118 0.263484
## AWERKT      -0.0010526  0.0728240  -0.014 0.988468
## ABROM       -0.0236462  0.0467611  -0.506 0.613101
## ALEVEN       0.0372344  0.0154024   2.417 0.015661 *
## APERSONG    -0.0464279  0.0954471  -0.486 0.626684
## AGEZONG     -0.4050642  0.1898715  -2.133 0.032938 *
## AWAOREG     -0.2304561  0.1243310  -1.854 0.063852 .
## ABRAND      -0.0211374  0.0116048  -1.821 0.068593 .
## AZEILPL      0.4958051  0.2815591   1.761 0.078304 .
## APLEZIER     0.3633887  0.0885318   4.105 4.11e-05 ***
## AFIETS       0.0416061  0.0408644   1.018 0.308650
## AINBOED      0.0959436  0.0699079   1.372 0.169983
## ABYSTAND     0.1312250  0.0983836   1.334 0.182319
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.23 on 5736 degrees of freedom
## Multiple R-squared:  0.0729, Adjusted R-squared:  0.05916
## F-statistic: 5.306 on 85 and 5736 DF,  p-value: < 2.2e-16
```

```
# The model is good as R-squared and adjusted R-squared is higher and
# p-value is comparitively lower overall.

pred = predict(ols_model, newdata = test_df)

compare = data.frame(actual = test_df$CARAVAN, predicted = pred)
head(compare,10)
```

```
##      actual   predicted
## 5823      0 0.014441132
## 5824      1 0.215946829
## 5825      0 0.099937482
## 5826      0 0.095439888
## 5827      0 0.005945841
```

```
## 5828      0 0.027520016
## 5829      0 0.101836066
## 5830      0 0.059439617
## 5831      0 0.097974707
## 5832      0 0.165365361
```

```
# Error obtained:
ols_model$results
```

```
##   intercept      RMSE  Rsquared       MAE     RMSESD  RsquaredSD       MAESD
## 1      TRUE 0.2363827 0.0347082 0.1168571 0.00578745 0.007201092 0.002507655
```

```
#####################
# Backward Selection:
#####################

bwd_model = lm(CARAVAN ~ ., data = train_df)
summary(bwd_model)
```

```
##
## Call:
## lm(formula = CARAVAN ~ ., data = train_df)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.67293 -0.08720 -0.04593 -0.00639  1.04628
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.7685381  0.4298406   1.788 0.073835 .
## MOSTYPE      0.0035209  0.0022512   1.564 0.117866
## MAANTHUI    -0.0072642  0.0076739  -0.947 0.343875
## MGEMOMV     -0.0012739  0.0071737  -0.178 0.859055
## MGEMLEEF     0.0107473  0.0049596   2.167 0.030279 *
## MOSHOOFD    -0.0154869  0.0101044  -1.533 0.125405
## MGODRK      -0.0056016  0.0056016  -1.000 0.317353
## MGODPR      -0.0002069  0.0060664  -0.034 0.972795
## MGODOV       0.0003569  0.0054592   0.065 0.947874
## MGODGE      -0.0030237  0.0058038  -0.521 0.602399
## MRELGE       0.0086829  0.0075479   1.150 0.250036
## MRELSA       0.0020367  0.0072008   0.283 0.777310
## MRELOV       0.0055682  0.0076295   0.730 0.465526
## MFALLEEN    -0.0038250  0.0065474  -0.584 0.559107
## MFGEKIND    -0.0050625  0.0066861  -0.757 0.448980
## MFWEKIND    -0.0026253  0.0069795  -0.376 0.706824
## MOPLHOOG     0.0021357  0.0068161   0.313 0.754038
## MOPLMIDD    -0.0048456  0.0071396  -0.679 0.497358
## MOPLLAAG    -0.0113977  0.0073004  -1.561 0.118525
## MBERHOOG     0.0021884  0.0045182   0.484 0.628153
## MBERZELF    -0.0004665  0.0052201  -0.089 0.928796
## MBERBOER    -0.0050974  0.0050426  -1.011 0.312122
## MBERMIDD     0.0041254  0.0044806   0.921 0.357228
## MBERARBG    -0.0006060  0.0044709  -0.136 0.892190
```

```
## MBERARBO     0.0019733  0.0044532   0.443 0.657690
## MSKA        -0.0013674  0.0051653  -0.265 0.791225
## MSKB1       -0.0031701  0.0050198  -0.632 0.527724
## MSKB2       -0.0012603  0.0044827  -0.281 0.778603
## MSKC         0.0024879  0.0049115   0.507 0.612502
## MSKD        -0.0008866  0.0047145  -0.188 0.850832
## MHHUUR      -0.0454201  0.0376622  -1.206 0.227872
## MHKOOP      -0.0432242  0.0376290  -1.149 0.250730
## MAUT1        0.0085964  0.0075592   1.137 0.255502
## MAUT2        0.0077871  0.0068554   1.136 0.256038
## MAUT0        0.0047215  0.0072646   0.650 0.515762
## MZFONDS     -0.0561024  0.0444643  -1.262 0.207094
## MZPART      -0.0593733  0.0443897  -1.338 0.181097
## MINKM30      0.0070879  0.0051150   1.386 0.165884
## MINK3045     0.0069414  0.0049276   1.409 0.158986
## MINK4575     0.0049679  0.0050144   0.991 0.321862
## MINK7512     0.0059267  0.0052728   1.124 0.261053
## MINK123M    -0.0098939  0.0069270  -1.428 0.153258
## MINKGEM      0.0063044  0.0045645   1.381 0.167277
## MKOOPKLA     0.0029097  0.0022664   1.284 0.199250
## PWAPART      0.0284931  0.0166017   1.716 0.086166 .
## PWABEDR     -0.0101533  0.0205121  -0.495 0.620625
## PWALAND     -0.0201220  0.0390424  -0.515 0.606301
## PPERSAUT     0.0102787  0.0026346   3.901 9.67e-05 ***
## PBESAUT      0.0014405  0.0148574   0.097 0.922765
## PMOTSCO     -0.0061279  0.0079415  -0.772 0.440364
## PVRAAUT     -0.0249190  0.0415892  -0.599 0.549083
## PAANHANG     0.0588044  0.0557610   1.055 0.291662
## PTRACTOR     0.0121481  0.0142358   0.853 0.393504
## PWERKT      -0.0062440  0.0370186  -0.169 0.866060
## PBROM        0.0078683  0.0152793   0.515 0.606598
## PLEVEN      -0.0155397  0.0064753  -2.400 0.016433 *
## PPERSONG     0.0098926  0.0335157   0.295 0.767880
## PGEZONG      0.1937254  0.0793370   2.442 0.014644 *
## PWAOREG      0.0647933  0.0256913   2.522 0.011696 *
## PBRAND       0.0132643  0.0035906   3.694 0.000223 ***
## PZEILPL     -0.1917507  0.1439848  -1.332 0.182998
## PPLEZIER    -0.0299076  0.0269224  -1.111 0.266666
## PFIETS      -0.0107777  0.0549693  -0.196 0.844564
## PINBOED     -0.0441620  0.0307404  -1.437 0.150883
## PBYSTAND    -0.0184858  0.0288890  -0.640 0.522269
## AWAPART     -0.0377952  0.0323794  -1.167 0.243154
## AWABEDR      0.0185448  0.0529740   0.350 0.726296
## AWALAND      0.0180904  0.1374585   0.132 0.895300
## APERSAUT     0.0002821  0.0127496   0.022 0.982347
## ABESAUT     -0.0214816  0.0652955  -0.329 0.742175
## AMOTSCO      0.0203252  0.0310683   0.654 0.513004
## AVRAAUT      0.0563675  0.1589388   0.355 0.722866
## AAANHANG    -0.0804238  0.0944352  -0.852 0.394455
## ATRACTOR    -0.0395651  0.0353795  -1.118 0.263484
## AWERKT      -0.0010526  0.0728240  -0.014 0.988468
## ABROM       -0.0236462  0.0467611  -0.506 0.613101
## ALEVEN       0.0372344  0.0154024   2.417 0.015661 *
## APERSONG    -0.0464279  0.0954471  -0.486 0.626684
```

```
## AGEZONG     -0.4050642  0.1898715  -2.133 0.032938 *
## AWAOREG     -0.2304561  0.1243310  -1.854 0.063852 .
## ABRAND      -0.0211374  0.0116048  -1.821 0.068593 .
## AZEILPL      0.4958051  0.2815591   1.761 0.078304 .
## APLEZIER     0.3633887  0.0885318   4.105 4.11e-05 ***
## AFIETS       0.0416061  0.0408644   1.018 0.308650
## AINBOED      0.0959436  0.0699079   1.372 0.169983
## ABYSTAND     0.1312250  0.0983836   1.334 0.182319
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.23 on 5736 degrees of freedom
## Multiple R-squared:  0.0729, Adjusted R-squared:  0.05916
## F-statistic: 5.306 on 85 and 5736 DF,  p-value: < 2.2e-16
```

```
step(bwd_model, direction = 'backward')
```

```
## Start:  AIC=-17029.26
## CARAVAN ~ MOSTYPE + MAANTHUI + MGEMOMV + MGEMLEEF + MOSHOOFD +
##     MGODRK + MGODPR + MGODOV + MGODGE + MRELGE + MRELSA + MRELOV +
##     MFALLEEN + MFGEKIND + MFWEKIND + MOPLHOOG + MOPLMIDD + MOPLLAAG +
##     MBERHOOG + MBERZELF + MBERBOER + MBERMIDD + MBERARBG + MBERARBO +
##     MSKA + MSKB1 + MSKB2 + MSKC + MSKD + MHHUUR + MHKOOP + MAUT1 +
##     MAUT2 + MAUT0 + MZFONDS + MZPART + MINKM30 + MINK3045 + MINK4575 +
##     MINK7512 + MINK123M + MINKGEM + MKOOPKLA + PWAPART + PWABEDR +
##     PWALAND + PPERSAUT + PBESAUT + PMOTSCO + PVRAAUT + PAANHANG +
##     PTRACTOR + PWERKT + PBROM + PLEVEN + PPERSONG + PGEZONG +
##     PWAOREG + PBRAND + PZEILPL + PPLEZIER + PFIETS + PINBOED +
##     PBYSTAND + AWAPART + AWABEDR + AWALAND + APERSAUT + ABESAUT +
##     AMOTSCO + AVRAAUT + AAANHANG + ATRACTOR + AWERKT + ABROM +
##     ALEVEN + APERSONG + AGEZONG + AWAOREG + ABRAND + AZEILPL +
##     APLEZIER + AFIETS + AINBOED + ABYSTAND
##
##            Df Sum of Sq    RSS    AIC
## - AWERKT    1   0.00001 303.35 -17031
## - APERSAUT  1   0.00003 303.35 -17031
## - MGODPR    1   0.00006 303.35 -17031
## - MGODOV    1   0.00023 303.35 -17031
## - MBERZELF  1   0.00042 303.35 -17031
## - PBESAUT   1   0.00050 303.35 -17031
## - AWALAND   1   0.00092 303.35 -17031
## - MBERARBG  1   0.00097 303.35 -17031
## - PWERKT    1   0.00150 303.35 -17031
## - MGEMOMV   1   0.00167 303.35 -17031
## - MSKD      1   0.00187 303.35 -17031
## - PFIETS    1   0.00203 303.35 -17031
## - MSKA      1   0.00371 303.35 -17031
## - MSKB2     1   0.00418 303.35 -17031
## - MRELSA    1   0.00423 303.35 -17031
## - PPERSONG  1   0.00461 303.35 -17031
## - MOPLHOOG  1   0.00519 303.35 -17031
## - ABESAUT   1   0.00572 303.35 -17031
## - AWABEDR   1   0.00648 303.35 -17031
## - AVRAAUT   1   0.00665 303.35 -17031
```

```
## - PBRAND    1    0.6842 305.36 -17097
## - MOPLLAAG  1    0.9207 305.60 -17092
## - APLEZIER  1    2.9734 307.65 -17053
## - PPERSAUT  1    4.9761 309.65 -17015
##
## Step:  AIC=-17108.4
## CARAVAN ~ MOSTYPE + MGEMLEEF + MOSHOOFD + MGODRK + MRELGE + MOPLMIDD +
##     MOPLLAAG + MBERBOER + MBERMIDD + MSKC + MHHUUR + MZFONDS +
##     MZPART + MINK123M + MINKGEM + MKOOPKLA + PWAPART + PWALAND +
##     PPERSAUT + PLEVEN + PGEZONG + PWAOREG + PBRAND + PINBOED +
##     ALEVEN + AGEZONG + AWAOREG + ABRAND + APLEZIER + AFIETS +
##     AINBOED + ABYSTAND
##
##            Df Sum of Sq    RSS    AIC
## <none>                  304.75 -17108
## - MOSHOOFD  1    0.1142 304.86 -17108
## - MGODRK    1    0.1151 304.86 -17108
## - MSKC      1    0.1167 304.87 -17108
## - MOSTYPE   1    0.1223 304.87 -17108
## - PINBOED   1    0.1265 304.87 -17108
## - MBERMIDD  1    0.1288 304.88 -17108
## - AINBOED   1    0.1293 304.88 -17108
## - MKOOPKLA  1    0.1333 304.88 -17108
## - MZFONDS   1    0.1549 304.90 -17107
## - MINKGEM   1    0.1598 304.91 -17107
## - MZPART    1    0.1658 304.91 -17107
## - AWAOREG   1    0.1686 304.92 -17107
## - MHHUUR    1    0.1693 304.92 -17107
## - ABRAND    1    0.1790 304.93 -17107
## - MBERBOER  1    0.1804 304.93 -17107
## - AGEZONG   1    0.2557 305.00 -17106
## - MOPLMIDD  1    0.2651 305.01 -17105
## - MINK123M  1    0.2784 305.03 -17105
## - PWALAND   1    0.3116 305.06 -17105
## - PWAOREG   1    0.3143 305.06 -17104
## - AFIETS    1    0.3204 305.07 -17104
## - PGEZONG   1    0.3275 305.07 -17104
## - ALEVEN    1    0.3456 305.09 -17104
## - MRELGE    1    0.3556 305.10 -17104
## - PLEVEN    1    0.3600 305.11 -17104
## - PWAPART   1    0.3635 305.11 -17104
## - MGEMLEEF  1    0.3950 305.14 -17103
## - ABYSTAND  1    0.4254 305.17 -17102
## - PBRAND    1    0.6873 305.44 -17097
## - MOPLLAAG  1    0.9206 305.67 -17093
## - APLEZIER  1    3.0776 307.82 -17052
## - PPERSAUT  1    4.9622 309.71 -17016

##
## Call:
## lm(formula = CARAVAN ~ MOSTYPE + MGEMLEEF + MOSHOOFD + MGODRK +
##     MRELGE + MOPLMIDD + MOPLLAAG + MBERBOER + MBERMIDD + MSKC +
##     MHHUUR + MZFONDS + MZPART + MINK123M + MINKGEM + MKOOPKLA +
##     PWAPART + PWALAND + PPERSAUT + PLEVEN + PGEZONG + PWAOREG +
```

```
##     PBRAND + PINBOED + ALEVEN + AGEZONG + AWAOREG + ABRAND +
##     APLEZIER + AFIETS + AINBOED + ABYSTAND, data = train_df)
##
## Coefficients:
## (Intercept)      MOSTYPE     MGEMLEEF     MOSHOOFD       MGODRK       MRELGE
##    0.609801     0.003347     0.011133    -0.014469    -0.004749     0.004911
##    MOPLMIDD     MOPLLAAG     MBERBOER     MBERMIDD         MSKC       MHHUUR
##   -0.006590    -0.012620    -0.006375     0.002960     0.003377    -0.002227
##     MZFONDS       MZPART     MINK123M      MINKGEM     MKOOPKLA      PWAPART
##   -0.068297    -0.070615    -0.013556     0.005143     0.003477     0.010240
##     PWALAND     PPERSAUT       PLEVEN      PGEZONG      PWAOREG       PBRAND
##   -0.015892     0.010367    -0.016719     0.196612     0.061930     0.012693
##     PINBOED       ALEVEN      AGEZONG      AWAOREG       ABRAND     APLEZIER
##   -0.047258     0.038948    -0.415267    -0.220124    -0.020997     0.282649
##      AFIETS      AINBOED     ABYSTAND
##    0.035385     0.108030     0.072438
```

```
# final AIC for backward elimination: AIC=-17108.4

# lm(formula = CARAVAN ~ MOSTYPE + MGEMLEEF + MOSHOOFD + MGODRK +
# MRELGE + MOPLMIDD + MOPLLAAG + MBERBOER + MBERMIDD + MSKC +
# MHHUUR + MZFONDS + MZPART + MINK123M + MINKGEM + MKOOPKLA +
# PWAPART + PWALAND + PPERSAUT + PLEVEN + PGEZONG + PWAOREG +
# PBRAND + PINBOED + ALEVEN + AGEZONG + AWAOREG + ABRAND +
# APLEZIER + AFIETS + AINBOED + ABYSTAND, data = train_df)

# Suggested 32 independent variable out of 86 variables by backward selection

bwd_reg_model = lm(formula = CARAVAN ~ MOSTYPE + MGEMLEEF + MOSHOOFD + MGODRK +
                   MRELGE + MOPLMIDD + MOPLLAAG + MBERBOER + MBERMIDD + MSKC +
                   MHHUUR + MZFONDS + MZPART + MINK123M + MINKGEM + MKOOPKLA +
                   PWAPART + PWALAND + PPERSAUT + PLEVEN + PGEZONG + PWAOREG +
                   PBRAND + PINBOED + ALEVEN + AGEZONG + AWAOREG + ABRAND +
                   APLEZIER + AFIETS + AINBOED + ABYSTAND, data = train_df)

pred_bwd = predict(bwd_reg_model, newdata = test_df)

compare = data.frame(actual = test_df$CARAVAN, predicted = pred_bwd)
head(compare,10)
```

```
##      actual  predicted
## 5823      0 0.02551718
## 5824      1 0.21287843
## 5825      0 0.08923228
## 5826      0 0.08120096
## 5827      0 0.01336200
## 5828      0 0.01426295
## 5829      0 0.10028739
## 5830      0 0.04882722
## 5831      0 0.10260585
## 5832      0 0.16459805
```

```
summary(bwd_reg_model)
```

```
##
## Call:
## lm(formula = CARAVAN ~ MOSTYPE + MGEMLEEF + MOSHOOFD + MGODRK +
##     MRELGE + MOPLMIDD + MOPLLAAG + MBERBOER + MBERMIDD + MSKC +
##     MHHUUR + MZFONDS + MZPART + MINK123M + MINKGEM + MKOOPKLA +
##     PWAPART + PWALAND + PPERSAUT + PLEVEN + PGEZONG + PWAOREG +
##     PBRAND + PINBOED + ALEVEN + AGEZONG + AWAOREG + ABRAND +
##     APLEZIER + AFIETS + AINBOED + ABYSTAND, data = train_df)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.62496 -0.08655 -0.04658 -0.00809  1.04179
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.609801   0.356706   1.710 0.087406 .
## MOSTYPE      0.003347   0.002196   1.524 0.127555
## MGEMLEEF     0.011133   0.004064   2.739 0.006175 **
## MOSHOOFD    -0.014469   0.009824  -1.473 0.140859
## MGODRK      -0.004749   0.003211  -1.479 0.139216
## MRELGE       0.004911   0.001890   2.599 0.009375 **
## MOPLMIDD    -0.006590   0.002937  -2.244 0.024861 *
## MOPLLAAG    -0.012620   0.003018  -4.182 2.93e-05 ***
## MBERBOER    -0.006375   0.003444  -1.851 0.064183 .
## MBERMIDD     0.002960   0.001892   1.564 0.117785
## MSKC         0.003377   0.002268   1.489 0.136619
## MHHUUR      -0.002227   0.001242  -1.793 0.072973 .
## MZFONDS     -0.068297   0.039815  -1.715 0.086334 .
## MZPART      -0.070615   0.039793  -1.775 0.076020 .
## MINK123M    -0.013556   0.005895  -2.300 0.021504 *
## MINKGEM      0.005143   0.002951   1.742 0.081476 .
## MKOOPKLA     0.003477   0.002185   1.591 0.111634
## PWAPART      0.010240   0.003897   2.628 0.008622 **
## PWALAND     -0.015892   0.006532  -2.433 0.015004 *
## PPERSAUT     0.010367   0.001068   9.709  < 2e-16 ***
## PLEVEN      -0.016719   0.006394  -2.615 0.008948 **
## PGEZONG      0.196612   0.078829   2.494 0.012653 *
## PWAOREG      0.061930   0.025347   2.443 0.014583 *
## PBRAND       0.012693   0.003513   3.613 0.000305 ***
## PINBOED     -0.047258   0.030483  -1.550 0.121119
## ALEVEN       0.038948   0.015201   2.562 0.010424 *
## AGEZONG     -0.415267   0.188419  -2.204 0.027567 *
## AWAOREG     -0.220124   0.122991  -1.790 0.073546 .
## ABRAND      -0.020997   0.011386  -1.844 0.065216 .
## APLEZIER     0.282649   0.036967   7.646 2.41e-14 ***
## AFIETS       0.035385   0.014342   2.467 0.013647 *
## AINBOED      0.108030   0.068941   1.567 0.117168
## ABYSTAND     0.072438   0.025481   2.843 0.004487 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

```
## Residual standard error: 0.2294 on 5789 degrees of freedom
## Multiple R-squared:  0.06862,    Adjusted R-squared:  0.06347
## F-statistic: 13.33 on 32 and 5789 DF,  p-value: < 2.2e-16
```

```
rss_error_bwd = mean((test_df$CARAVAN - pred_bwd)^2)
rss_error_bwd
```

```
## [1] 0.05398562
```

```
rmse_error_bwd = sqrt(mean(test_df$CARAVAN - pred_bwd)^2)
rmse_error_bwd
```

```
## [1] 0.0001784071
```

```
####################
# Forward Selection:
####################

fwd_model = lm(CARAVAN ~ 1, data = train_df)
summary(fwd_model)
```

```
##
## Call:
## lm(formula = CARAVAN ~ 1, data = train_df)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.05977 -0.05977 -0.05977 -0.05977  0.94023
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 0.059773   0.003107   19.24   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2371 on 5821 degrees of freedom
```

```
step(fwd_model, direction = 'forward', scope = formula(bwd_model))
```

```
## Start:  AIC=-16758.55
## CARAVAN ~ 1
##
##            Df Sum of Sq    RSS    AIC
## + PPERSAUT  1    7.4515 319.75 -16891
## + APERSAUT  1    6.8046 320.39 -16879
## + APLEZIER  1    3.6556 323.54 -16822
## + PWAPART   1    3.0446 324.15 -16811
## + MKOOPKLA  1    3.0116 324.19 -16810
## + PBRAND    1    2.9186 324.28 -16809
## + PPLEZIER  1    2.7311 324.47 -16805
## + MOPLLAAG  1    2.6835 324.52 -16805
```

```
## + MINKGEM   1    2.6612 324.54 -16804
## + AWAPART   1    2.6144 324.58 -16803
## + MOPLHOOG  1    2.3438 324.86 -16798
## + MINKM30   1    2.0818 325.12 -16794
## + MHHUUR    1    2.0495 325.15 -16793
## + MHKOOP    1    2.0126 325.19 -16793
## + MAUT0     1    1.9149 325.28 -16791
## + MRELGE    1    1.6154 325.58 -16785
## + MAUT1     1    1.6021 325.60 -16785
## + MOSTYPE   1    1.5752 325.62 -16785
## + MOSHOOFD  1    1.5714 325.63 -16785
## + ABYSTAND  1    1.4538 325.75 -16783
## + MBERHOOG  1    1.3859 325.81 -16781
## + MSKA      1    1.3192 325.88 -16780
## + MSKD      1    1.3016 325.90 -16780
## + PBYSTAND  1    1.2900 325.91 -16780
## + MRELOV    1    1.2694 325.93 -16779
## + ABRAND    1    1.2328 325.97 -16779
## + MZFONDS   1    1.1151 326.08 -16776
## + MINK7512  1    1.0977 326.10 -16776
## + MINK4575  1    1.0890 326.11 -16776
## + MZPART    1    1.0825 326.12 -16776
## + MBERARBO  1    0.9766 326.22 -16774
## + MBERBOER  1    0.9542 326.24 -16774
## + MFALLEEN  1    0.9226 326.28 -16773
## + MBERMIDD  1    0.7450 326.45 -16770
## + ABROM     1    0.6660 326.53 -16768
## + PBROM     1    0.6453 326.55 -16768
## + MOPLMIDD  1    0.6196 326.58 -16768
## + MBERARBG  1    0.5909 326.61 -16767
## + MSKC      1    0.5857 326.61 -16767
## + MGODGE    1    0.5692 326.63 -16767
## + PGEZONG   1    0.5290 326.67 -16766
## + ALEVEN    1    0.4508 326.75 -16765
## + MGEMOMV   1    0.4141 326.78 -16764
## + AFIETS    1    0.3815 326.82 -16763
## + AGEZONG   1    0.3683 326.83 -16763
## + MRELSA    1    0.3642 326.83 -16763
## + MGODPR    1    0.3534 326.85 -16763
## + MFWEKIND  1    0.3423 326.86 -16763
## + PWAOREG   1    0.3049 326.89 -16762
## + MSKB1     1    0.2707 326.93 -16761
## + PFIETS    1    0.2694 326.93 -16761
## + AZEILPL   1    0.2246 326.97 -16761
## + AWAOREG   1    0.1633 327.04 -16760
## + MBERZELF  1    0.1580 327.04 -16759
## + PWALAND   1    0.1531 327.05 -16759
## + AWALAND   1    0.1482 327.05 -16759
## + PLEVEN    1    0.1464 327.05 -16759
## <none>                  327.20 -16759
## + AINBOED   1    0.1063 327.09 -16758
## + ATRACTOR  1    0.0968 327.10 -16758
## + PWERKT    1    0.0677 327.13 -16758
## + PAANHANG  1    0.0522 327.15 -16758
```

```
## + MINK4575  1  0.037851 305.65 -17107
## + PLEVEN    1  0.036794 305.65 -17107
## + MAANTHUI  1  0.028336 305.65 -17107
## + PTRACTOR  1  0.028214 305.65 -17107
## + MRELOV    1  0.025260 305.66 -17107
## + PPLEZIER  1  0.024867 305.66 -17107
## + ALEVEN    1  0.024525 305.66 -17107
## + MINK7512  1  0.023448 305.66 -17107
## + PAANHANG  1  0.023213 305.66 -17107
## + APERSONG  1  0.020920 305.66 -17107
## + MOSTYPE   1  0.018088 305.67 -17107
## + PBYSTAND  1  0.016111 305.67 -17107
## + MBERZELF  1  0.015949 305.67 -17107
## + AWABEDR   1  0.013890 305.67 -17107
## + MBERARBG  1  0.013881 305.67 -17107
## + MSKA      1  0.012988 305.67 -17107
## + AINBOED   1  0.012843 305.67 -17107
## + PPERSONG  1  0.011853 305.67 -17107
## + PZEILPL   1  0.010744 305.67 -17107
## + MSKD      1  0.009109 305.67 -17107
## + AAANHANG  1  0.008145 305.68 -17107
## + MBERHOOG  1  0.006919 305.68 -17107
## + MOSHOOFD  1  0.006872 305.68 -17107
## + PMOTSCO   1  0.006243 305.68 -17107
## + MRELSA    1  0.006203 305.68 -17107
## + MGEMOMV   1  0.005944 305.68 -17107
## + PINBOED   1  0.005845 305.68 -17107
## + MINKM30   1  0.003865 305.68 -17107
## + APERSAUT  1  0.002360 305.68 -17107
## + MFALLEEN  1  0.001981 305.68 -17107
## + MGODGE    1  0.001827 305.68 -17107
## + PBROM     1  0.001129 305.68 -17107
## + MBERARBO  1  0.001100 305.68 -17107
## + MSKB2     1  0.000778 305.68 -17107
## + PFIETS    1  0.000620 305.68 -17107
## + AWALAND   1  0.000424 305.68 -17107
## + MSKB1     1  0.000364 305.68 -17107
## + MAUT2     1  0.000180 305.68 -17107
## + ABROM     1  0.000047 305.68 -17107
## + MOPLMIDD  1  0.000038 305.68 -17107
## + AMOTSCO   1  0.000005 305.68 -17107

##
## Call:
## lm(formula = CARAVAN ~ PPERSAUT + APLEZIER + MKOOPKLA + PWAPART +
##     MOPLLAAG + MRELGE + PBRAND + ABYSTAND + MBERBOER + AFIETS +
##     PWALAND + PWAOREG + MGEMLEEF + MINK123M + MINKGEM + ABRAND +
##     AWAOREG + MOPLHOOG + MGODPR + MZPART + MZFONDS + PGEZONG +
##     AGEZONG, data = train_df)
##
## Coefficients:
## (Intercept)     PPERSAUT     APLEZIER     MKOOPKLA      PWAPART     MOPLLAAG
##    0.601139     0.010303     0.283549     0.003178     0.010328    -0.005564
##      MRELGE       PBRAND     ABYSTAND     MBERBOER       AFIETS      PWALAND
```

```
##    0.005090     0.013250     0.070120    -0.008560     0.037378    -0.016014
##     PWAOREG     MGEMLEEF     MINK123M      MINKGEM       ABRAND      AWAOREG
##    0.062101     0.009609    -0.014090     0.006175    -0.021893    -0.220446
##    MOPLHOOG       MGODPR       MZPART      MZFONDS      PGEZONG      AGEZONG
##    0.004894     0.003566    -0.078183    -0.074830     0.191535    -0.405557
```

```
# final AIC for forward selection: AIC=-17108.56

# lm(formula = CARAVAN ~ PPERSAUT + APLEZIER + MKOOPKLA + PWAPART +
# MOPLLAAG + MRELGE + PBRAND + ABYSTAND + MBERBOER + AFIETS +
# PWALAND + PWAOREG + MGEMLEEF + MINK123M + MINKGEM + ABRAND +
# AWAOREG + MOPLHOOG + MGODPR + MZPART + MZFONDS + PGEZONG +
# AGEZONG, data = train_df)

# Suggested 23 independent variables out of 86 variables by forward selection

fwd_reg_model = lm(formula = CARAVAN ~ PPERSAUT + APLEZIER + MKOOPKLA + PWAPART +
                   MOPLLAAG + MRELGE + PBRAND + ABYSTAND + MBERBOER + AFIETS +
                   PWALAND + PWAOREG + MGEMLEEF + MINK123M + MINKGEM + ABRAND +
                   AWAOREG + MOPLHOOG + MGODPR + MZPART + MZFONDS + PGEZONG +
                   AGEZONG, data = train_df)

pred_fwd = predict(fwd_reg_model, newdata = test_df)

compare = data.frame(actual = test_df$CARAVAN, predicted = pred_fwd)
head(compare,10)
```

```
##      actual   predicted
## 5823      0 0.027720983
## 5824      1 0.183988141
## 5825      0 0.113121968
## 5826      0 0.092352964
## 5827      0 0.002689990
## 5828      0 0.009042519
## 5829      0 0.112027249
## 5830      0 0.059854104
## 5831      0 0.102759045
## 5832      0 0.161578486
```

```
summary(fwd_reg_model)
```

```
##
## Call:
## lm(formula = CARAVAN ~ PPERSAUT + APLEZIER + MKOOPKLA + PWAPART +
##     MOPLLAAG + MRELGE + PBRAND + ABYSTAND + MBERBOER + AFIETS +
##     PWALAND + PWAOREG + MGEMLEEF + MINK123M + MINKGEM + ABRAND +
##     AWAOREG + MOPLHOOG + MGODPR + MZPART + MZFONDS + PGEZONG +
##     AGEZONG, data = train_df)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.62186 -0.08667 -0.04661 -0.00780  1.03249
##
```

```
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.601139   0.358860   1.675 0.093962 .
## PPERSAUT     0.010303   0.001067   9.655  < 2e-16 ***
## APLEZIER     0.283549   0.036931   7.678 1.89e-14 ***
## MKOOPKLA     0.003178   0.001882   1.689 0.091287 .
## PWAPART      0.010328   0.003877   2.664 0.007749 **
## MOPLLAAG    -0.005564   0.001912  -2.909 0.003637 **
## MRELGE       0.005090   0.001790   2.844 0.004477 **
## PBRAND       0.013250   0.003467   3.822 0.000134 ***
## ABYSTAND     0.070120   0.025468   2.753 0.005919 **
## MBERBOER    -0.008560   0.003011  -2.842 0.004492 **
## AFIETS       0.037378   0.014335   2.608 0.009144 **
## PWALAND     -0.016014   0.006531  -2.452 0.014241 *
## PWAOREG      0.062101   0.025342   2.450 0.014295 *
## MGEMLEEF     0.009609   0.003969   2.421 0.015514 *
## MINK123M    -0.014090   0.005875  -2.398 0.016500 *
## MINKGEM      0.006175   0.002905   2.125 0.033598 *
## ABRAND      -0.021893   0.011332  -1.932 0.053410 .
## AWAOREG     -0.220446   0.123001  -1.792 0.073148 .
## MOPLHOOG     0.004894   0.002644   1.851 0.064269 .
## MGODPR       0.003566   0.001820   1.959 0.050139 .
## MZPART      -0.078183   0.039816  -1.964 0.049623 *
## MZFONDS     -0.074830   0.039825  -1.879 0.060301 .
## PGEZONG      0.191535   0.078831   2.430 0.015142 *
## AGEZONG     -0.405557   0.188269  -2.154 0.031271 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2296 on 5798 degrees of freedom
## Multiple R-squared:  0.06576,    Adjusted R-squared:  0.06205
## F-statistic: 17.74 on 23 and 5798 DF,  p-value: < 2.2e-16
```

```
rss_error_fwd = mean((test_df$CARAVAN - pred_fwd)^2)
rss_error_fwd
```

```
## [1] 0.05393412
```

```
rmse_error_fwd = sqrt(mean(test_df$CARAVAN - pred_fwd)^2)
rmse_error_fwd
```

```
## [1] 6.881192e-05
```

```
######################
# Stepwise Selection: (Combining both forward and backward)
######################

step(fwd_model, direction = 'both', scope = formula(bwd_model))
```

```
## Start:  AIC=-16758.55
## CARAVAN ~ 1
##
```

```
##             Df Sum of Sq    RSS    AIC
## + PPERSAUT   1    7.4515 319.75 -16891
## + APERSAUT   1    6.8046 320.39 -16879
## + APLEZIER   1    3.6556 323.54 -16822
## + PWAPART    1    3.0446 324.15 -16811
## + MKOOPKLA   1    3.0116 324.19 -16810
## + PBRAND     1    2.9186 324.28 -16809
## + PPLEZIER   1    2.7311 324.47 -16805
## + MOPLLAAG   1    2.6835 324.52 -16805
## + MINKGEM    1    2.6612 324.54 -16804
## + AWAPART    1    2.6144 324.58 -16803
## + MOPLHOOG   1    2.3438 324.86 -16798
## + MINKM30    1    2.0818 325.12 -16794
## + MHHUUR     1    2.0495 325.15 -16793
## + MHKOOP     1    2.0126 325.19 -16793
## + MAUT0      1    1.9149 325.28 -16791
## + MRELGE     1    1.6154 325.58 -16785
## + MAUT1      1    1.6021 325.60 -16785
## + MOSTYPE    1    1.5752 325.62 -16785
## + MOSHOOFD   1    1.5714 325.63 -16785
## + ABYSTAND   1    1.4538 325.75 -16783
## + MBERHOOG   1    1.3859 325.81 -16781
## + MSKA       1    1.3192 325.88 -16780
## + MSKD       1    1.3016 325.90 -16780
## + PBYSTAND   1    1.2900 325.91 -16780
## + MRELOV     1    1.2694 325.93 -16779
## + ABRAND     1    1.2328 325.97 -16779
## + MZFONDS    1    1.1151 326.08 -16776
## + MINK7512   1    1.0977 326.10 -16776
## + MINK4575   1    1.0890 326.11 -16776
## + MZPART     1    1.0825 326.12 -16776
## + MBERARBO   1    0.9766 326.22 -16774
## + MBERBOER   1    0.9542 326.24 -16774
## + MFALLEEN   1    0.9226 326.28 -16773
## + MBERMIDD   1    0.7450 326.45 -16770
## + ABROM      1    0.6660 326.53 -16768
## + PBROM      1    0.6453 326.55 -16768
## + MOPLMIDD   1    0.6196 326.58 -16768
## + MBERARBG   1    0.5909 326.61 -16767
## + MSKC       1    0.5857 326.61 -16767
## + MGODGE     1    0.5692 326.63 -16767
## + PGEZONG    1    0.5290 326.67 -16766
## + ALEVEN     1    0.4508 326.75 -16765
## + MGEMOMV    1    0.4141 326.78 -16764
## + AFIETS     1    0.3815 326.82 -16763
## + AGEZONG    1    0.3683 326.83 -16763
## + MRELSA     1    0.3642 326.83 -16763
## + MGODPR     1    0.3534 326.85 -16763
## + MFWEKIND   1    0.3423 326.86 -16763
## + PWAOREG    1    0.3049 326.89 -16762
## + MSKB1      1    0.2707 326.93 -16761
## + PFIETS     1    0.2694 326.93 -16761
## + AZEILPL    1    0.2246 326.97 -16761
## + AWAOREG    1    0.1633 327.04 -16760
```

```
## + PFIETS    1    0.0000 305.93 -17104
## - MZPART    1    0.2124 306.14 -17104
## - MINKGEM   1    0.2482 306.18 -17103
## - MGEMLEEF  1    0.2976 306.23 -17102
## - MINK123M  1    0.3065 306.24 -17102
## - PWAOREG   1    0.3168 306.25 -17102
## - PWALAND   1    0.3309 306.26 -17102
## - AFIETS    1    0.3340 306.26 -17102
## - PWAPART   1    0.3645 306.29 -17101
## - MRELGE    1    0.4192 306.35 -17100
## - ABYSTAND  1    0.4230 306.35 -17100
## - MBERBOER  1    0.4241 306.35 -17100
## - MOPLLAAG  1    0.4496 306.38 -17099
## - PBRAND    1    0.7609 306.69 -17093
## - APLEZIER  1    3.1073 309.04 -17049
## - PPERSAUT  1    4.9334 310.86 -17015
##
## Step:  AIC=-17108.56
## CARAVAN ~ PPERSAUT + APLEZIER + MKOOPKLA + PWAPART + MOPLLAAG +
##     MRELGE + PBRAND + ABYSTAND + MBERBOER + AFIETS + PWALAND +
##     PWAOREG + MGEMLEEF + MINK123M + MINKGEM + ABRAND + AWAOREG +
##     MOPLHOOG + MGODPR + MZPART + MZFONDS + PGEZONG + AGEZONG
##
##            Df Sum of Sq    RSS    AIC
## <none>                  305.68 -17109
## + MBERMIDD  1    0.0989 305.58 -17108
## + AZEILPL   1    0.0928 305.59 -17108
## + PWERKT    1    0.0860 305.60 -17108
## + MHHUUR    1    0.0836 305.60 -17108
## + MAUT1     1    0.0833 305.60 -17108
## + MHKOOP    1    0.0775 305.61 -17108
## + MSKC      1    0.0769 305.61 -17108
## + ATRACTOR  1    0.0752 305.61 -17108
## + AWERKT    1    0.0698 305.61 -17108
## + MINK3045  1    0.0666 305.62 -17108
## + AWAPART   1    0.0651 305.62 -17108
## + MGODOV    1    0.0624 305.62 -17108
## + ABESAUT   1    0.0617 305.62 -17108
## + PVRAAUT   1    0.0609 305.62 -17108
## - MKOOPKLA  1    0.1504 305.83 -17108
## + MAUT0     1    0.0478 305.64 -17108
## + MFGEKIND  1    0.0467 305.64 -17107
## + AVRAAUT   1    0.0456 305.64 -17107
## + PBESAUT   1    0.0451 305.64 -17107
## + PWABEDR   1    0.0446 305.64 -17107
## + MGODRK    1    0.0430 305.64 -17107
## + MFWEKIND  1    0.0430 305.64 -17107
## - AWAOREG   1    0.1693 305.85 -17107
## + MINK4575  1    0.0379 305.65 -17107
## + PLEVEN    1    0.0368 305.65 -17107
## - MOPLHOOG  1    0.1806 305.86 -17107
## + MAANTHUI  1    0.0283 305.65 -17107
## + PTRACTOR  1    0.0282 305.65 -17107
## + MRELOV    1    0.0253 305.66 -17107
```

```
## - APLEZIER  1    3.1079 308.79 -17052
## - PPERSAUT  1    4.9149 310.60 -17018

##
## Call:
## lm(formula = CARAVAN ~ PPERSAUT + APLEZIER + MKOOPKLA + PWAPART +
##     MOPLLAAG + MRELGE + PBRAND + ABYSTAND + MBERBOER + AFIETS +
##     PWALAND + PWAOREG + MGEMLEEF + MINK123M + MINKGEM + ABRAND +
##     AWAOREG + MOPLHOOG + MGODPR + MZPART + MZFONDS + PGEZONG +
##     AGEZONG, data = train_df)
##
## Coefficients:
## (Intercept)     PPERSAUT     APLEZIER     MKOOPKLA      PWAPART     MOPLLAAG
##    0.601139     0.010303     0.283549     0.003178     0.010328    -0.005564
##      MRELGE       PBRAND     ABYSTAND     MBERBOER       AFIETS      PWALAND
##    0.005090     0.013250     0.070120    -0.008560     0.037378    -0.016014
##     PWAOREG     MGEMLEEF     MINK123M      MINKGEM       ABRAND      AWAOREG
##    0.062101     0.009609    -0.014090     0.006175    -0.021893    -0.220446
##    MOPLHOOG       MGODPR       MZPART      MZFONDS      PGEZONG      AGEZONG
##    0.004894     0.003566    -0.078183    -0.074830     0.191535    -0.405557
```

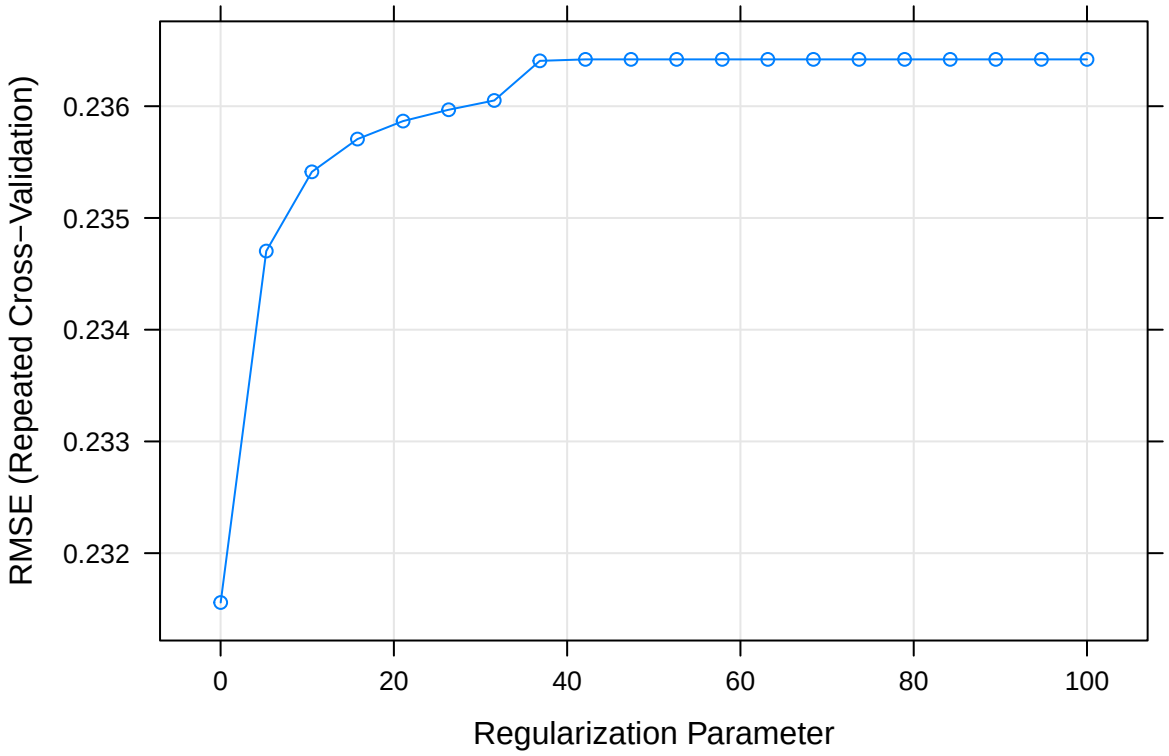
```
# final AIC for stepwise selection: AIC=-17108.56

# lm(formula = CARAVAN ~ PPERSAUT + APLEZIER + MKOOPKLA + PWAPART +
# MOPLLAAG + MRELGE + PBRAND + ABYSTAND + MBERBOER + AFIETS +
# PWALAND + PWAOREG + MGEMLEEF + MINK123M + MINKGEM + ABRAND +
# AWAOREG + MOPLHOOG + MGODPR + MZPART + MZFONDS + PGEZONG +
# AGEZONG, data = train_df)

# Suggested 23 independent variables out of 86 variables by stepwise selction

# Here, we can see that, although both backward, forward selection and
# stepwise selection got similar 'AIC' value, forward selection done better
# in reducing the model complexity by reducing no.of variables.


####################
# Ridge regression:
####################

param_control = trainControl(method = 'repeatedcv', number = 10, repeats = 5)

ridge_reg_model_2 = train(CARAVAN ~ ., data = train_df, method = 'glmnet',
                          trControl = param_control,
                          tuneGrid = expand.grid(alpha =0,
                                                 lambda = seq(0.01, 100,
                                                              length = 20)))
```

```
## Warning in train.default(x, y, weights = w, ...): You are trying to do
## regression and your outcome only has two possible values Are you trying to do
## classification? If so, use a 2 level factor as your outcome column.

## Warning in nominalTrainWorkflow(x = x, y = y, wts = weights, info = trainInfo, :
## There were missing values in resampled performance measures.
```

```
ridge_reg_model_2
```

```
## glmnet
##
## 5822 samples
##   85 predictor
##
## No pre-processing
## Resampling: Cross-Validated (10 fold, repeated 5 times)
## Summary of sample sizes: 5240, 5240, 5240, 5240, 5240, 5240, ...
## Resampling results across tuning parameters:
##
##   lambda      RMSE       Rsquared    MAE
##     0.010000  0.2315585  0.04671303  0.1124647
##     5.272632  0.2347038  0.03960778  0.1115149
##    10.535263  0.2354128  0.03795081  0.1119137
##    15.797895  0.2357061  0.03730414  0.1120680
##    21.060526  0.2358669  0.03696178  0.1121502
##    26.323158  0.2359684  0.03674993  0.1122012
##    31.585789  0.2360514  0.03660781  0.1122429
##    36.848421  0.2364060  0.02916711  0.1124145
##    42.111053  0.2364193         NaN  0.1124203
##    47.373684  0.2364193         NaN  0.1124203
##    52.636316  0.2364193         NaN  0.1124203
##    57.898947  0.2364193         NaN  0.1124203
##    63.161579  0.2364193         NaN  0.1124203
##    68.424211  0.2364193         NaN  0.1124203
##    73.686842  0.2364193         NaN  0.1124203
##    78.949474  0.2364193         NaN  0.1124203
##    84.212105  0.2364193         NaN  0.1124203
##    89.474737  0.2364193         NaN  0.1124203
##    94.737368  0.2364193         NaN  0.1124203
##   100.000000  0.2364193         NaN  0.1124203
##
## Tuning parameter 'alpha' was held constant at a value of 0
## RMSE was used to select the optimal model using the smallest value.
## The final values used for the model were alpha = 0 and lambda = 0.01.
```

```
plot(ridge_reg_model_2)
```




```
###################
# Lasso Regression:
###################

param_control = trainControl(method = 'repeatedcv', number = 10, repeats = 5)

lasso_model_2 = train(CARAVAN ~ ., data = train_df, method = 'glmnet',
                      trControl = param_control,
                      tuneGrid = expand.grid(alpha = 1,
                                             lambda = seq(0.01, 100, length = 20)))
```

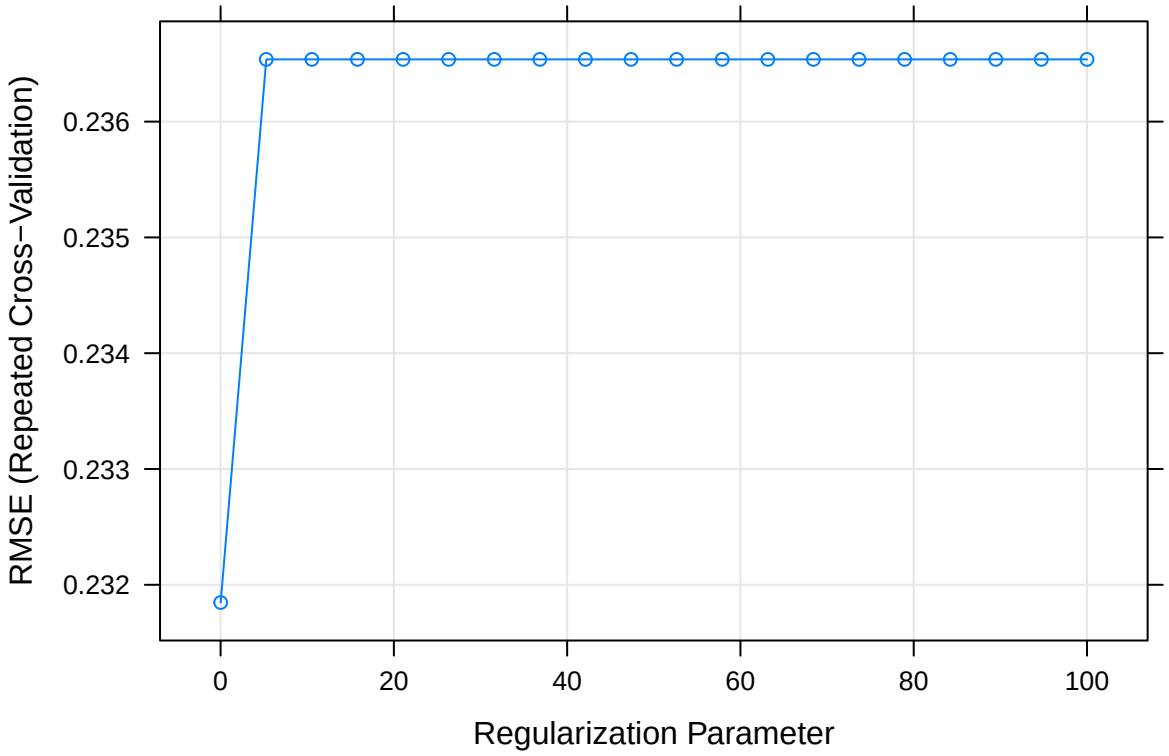
```
## Warning in train.default(x, y, weights = w, ...): You are trying to do
## regression and your outcome only has two possible values Are you trying to do
## classification? If so, use a 2 level factor as your outcome column.

## Warning in train.default(x, y, weights = w, ...): There were missing values in
## resampled performance measures.
```

```
lasso_model_2
```

```
## glmnet
##
## 5822 samples
##   85 predictor
##
```

```
## No pre-processing
## Resampling: Cross-Validated (10 fold, repeated 5 times)
## Summary of sample sizes: 5240, 5239, 5240, 5240, 5240, 5240, ...
## Resampling results across tuning parameters:
##
##   lambda      RMSE       Rsquared    MAE
##     0.010000  0.2318471  0.04706669  0.1090262
##     5.272632  0.2365375         NaN  0.1124183
##    10.535263  0.2365375         NaN  0.1124183
##    15.797895  0.2365375         NaN  0.1124183
##    21.060526  0.2365375         NaN  0.1124183
##    26.323158  0.2365375         NaN  0.1124183
##    31.585789  0.2365375         NaN  0.1124183
##    36.848421  0.2365375         NaN  0.1124183
##    42.111053  0.2365375         NaN  0.1124183
##    47.373684  0.2365375         NaN  0.1124183
##    52.636316  0.2365375         NaN  0.1124183
##    57.898947  0.2365375         NaN  0.1124183
##    63.161579  0.2365375         NaN  0.1124183
##    68.424211  0.2365375         NaN  0.1124183
##    73.686842  0.2365375         NaN  0.1124183
##    78.949474  0.2365375         NaN  0.1124183
##    84.212105  0.2365375         NaN  0.1124183
##    89.474737  0.2365375         NaN  0.1124183
##    94.737368  0.2365375         NaN  0.1124183
##   100.000000  0.2365375         NaN  0.1124183
##
## Tuning parameter 'alpha' was held constant at a value of 1
## RMSE was used to select the optimal model using the smallest value.
## The final values used for the model were alpha = 1 and lambda = 0.01.
```

```
plot(lasso_model_2)
```




```
##############
# Conclusion:
##############

# Forward selection/stepwise selection performed well overall. That is because,
# apart from the model not being complex with just 23 independent variables,
# unlike in backward selection with 32 variables or ridge or lasso with around
# 86 independent variable, the error value is better for forward selection.
```

```
################################################################################
# Question 3:
################################################################################


# Load train data

X <- as.matrix(read.table(gzfile("zip.train")))
head(X)
```

```
##      V1 V2 V3 V4     V5     V6     V7     V8     V9    V10    V11    V12    V13
## [1,]  6 -1 -1 -1 -1.000 -1.000 -1.000 -1.000 -0.631  0.862 -0.167 -1.000 -1.000
## [2,]  5 -1 -1 -1 -0.813 -0.671 -0.809 -0.887 -0.671 -0.853 -1.000 -1.000 -0.774
## [3,]  4 -1 -1 -1 -1.000 -1.000 -1.000 -1.000 -1.000 -1.000 -0.996  0.147  1.000
## [4,]  7 -1 -1 -1 -1.000 -1.000 -0.273  0.684  0.960  0.450 -0.067 -0.679 -1.000
## [5,]  3 -1 -1 -1 -1.000 -1.000 -0.928 -0.204  0.751  0.466  0.234 -0.809 -1.000
## [6,]  6 -1 -1 -1 -1.000 -1.000 -0.397  0.983 -0.535 -1.000 -1.000 -1.000 -1.000
##         V14    V15    V16 V17 V18 V19 V20    V21    V22    V23    V24    V25
## [1,] -1.000 -1.000 -1.000  -1  -1  -1  -1 -1.000 -1.000 -1.000 -0.992  0.297
## [2,] -0.180  0.052 -0.241  -1  -1  -1  -1  0.392  1.000  0.857  0.727  1.000
## [3,] -0.189 -1.000 -1.000  -1  -1  -1  -1 -1.000 -1.000 -1.000 -1.000 -1.000
## [4,] -1.000 -1.000 -1.000  -1  -1  -1  -1 -1.000 -0.114  0.974  0.917  0.734
## [5,] -1.000 -1.000 -1.000  -1  -1  -1  -1 -1.000 -0.370  0.739  1.000  1.000
## [6,] -1.000 -1.000 -1.000  -1  -1  -1  -1 -1.000 -1.000  0.692  0.536 -0.767
##         V26    V27    V28    V29    V30    V31    V32 V33 V34 V35 V36    V37
## [1,]  1.000  0.307 -1.000 -1.000 -1.000 -1.000 -1.000  -1  -1  -1  -1 -1.000
## [2,]  0.805  0.613  0.613  0.860  1.000  1.000  0.396  -1  -1  -1  -1 -0.548
## [3,] -1.000 -0.882  1.000  0.390 -0.811 -1.000 -1.000  -1  -1  -1  -1 -1.000
## [4,]  0.994  1.000  0.973  0.391 -0.421 -0.976 -1.000  -1  -1  -1  -1 -0.323
## [5,]  1.000  1.000  0.644 -0.890 -1.000 -1.000 -1.000  -1  -1  -1  -1 -1.000
## [6,] -1.000 -1.000 -1.000 -1.000 -1.000 -1.000 -1.000  -1  -1  -1  -1 -1.000
##         V38    V39    V40    V41    V42    V43    V44    V45 V46    V47    V48
## [1,] -1.000 -1.000 -0.410  1.000  0.986 -0.565 -1.000 -1.000  -1 -1.000 -1.000
## [2,]  1.000  1.000  1.000  1.000  1.000  1.000  1.000  1.000   1  1.000  0.875
## [3,] -1.000 -1.000 -1.000 -1.000 -1.000 -0.715  1.000  0.029  -1 -1.000 -1.000
## [4,]  0.991  0.622 -0.738 -1.000 -0.639  0.023  0.871  1.000   1 -0.432 -1.000
## [5,]  0.616  1.000  0.688 -0.455 -0.731  0.659  1.000 -0.287  -1 -1.000 -1.000
## [6,] -0.921  0.928 -0.118 -1.000 -1.000 -1.000 -1.000 -1.000  -1 -1.000 -1.000
##         V49 V50 V51 V52    V53    V54    V55    V56 V57    V58    V59    V60
## [1,] -1.000  -1  -1  -1 -1.000 -1.000 -0.683  0.825   1  0.562 -1.000 -1.000
## [2,] -0.957  -1  -1  -1 -0.786  0.961  1.000  1.000   1  0.727  0.403  0.403
## [3,] -1.000  -1  -1  -1 -1.000 -0.888 -0.912 -1.000  -1 -1.000 -0.549  1.000
## [4,] -1.000  -1  -1  -1  0.409  1.000  0.000 -1.000  -1 -1.000 -1.000 -0.842
## [5,] -1.000  -1  -1  -1 -1.000 -0.376 -0.186 -0.874  -1 -1.000 -0.014  1.000
## [6,] -1.000  -1  -1  -1 -1.000 -0.394  1.000 -0.596  -1 -1.000 -1.000 -1.000
##         V61    V62    V63   V64 V65 V66 V67    V68    V69    V70    V71    V72
## [1,] -1.000 -1.000 -1.000 -1.00  -1  -1  -1 -1.000 -1.000 -0.938  0.540  1.000
## [2,]  0.171 -0.314 -0.314 -0.94  -1  -1  -1 -1.000 -0.298  1.000  1.000  1.000
## [3,]  0.361 -1.000 -1.000 -1.00  -1  -1  -1 -1.000 -0.938  0.694  0.057 -1.000
## [4,]  0.714  1.000 -0.534 -1.00  -1  -1  -1 -0.879  0.965  1.000 -0.713 -1.000
## [5,] -0.253 -1.000 -1.000 -1.00  -1  -1  -1 -1.000 -1.000 -1.000 -1.000 -1.000
## [6,] -1.000 -1.000 -1.000 -1.00  -1  -1  -1 -1.000 -1.000  0.060  0.900 -0.951
##         V73    V74    V75    V76    V77    V78    V79 V80 V81 V82 V83    V84
```

```
## [4,] -1.000 -1.000 -1.000 -1.000 -1.00   -1   -1   -1 -1.000 -1.000 -1.000
## [5,]  1.000  1.000  0.583 -0.843 -1.00   -1   -1   -1 -0.877 -0.326  0.174
## [6,] -0.882 -1.000 -1.000 -1.000 -1.00   -1   -1   -1 -0.898  0.323  1.000
##        V248   V249   V250   V251   V252   V253   V254   V255   V256 V257
## [1,]  0.304  0.823  1.000  0.482 -0.474 -0.991 -1.000 -1.000 -1.000   -1
## [2,] -0.671 -0.671 -0.033  0.761  0.762  0.126 -0.095 -0.671 -0.828   -1
## [3,] -1.000 -1.000 -1.000 -0.109  1.000 -0.179 -1.000 -1.000 -1.000   -1
## [4,] -0.318  1.000  0.536 -0.987 -1.000 -1.000 -1.000 -1.000 -1.000   -1
## [5,]  0.466  0.639  1.000  1.000  0.791  0.439 -0.199 -0.883 -1.000   -1
## [6,]  0.803  0.015 -0.862 -0.871 -0.437 -1.000 -1.000 -1.000 -1.000   -1
```

```
dim(X)
```

```
## [1] 7291  257
```

```
X_7_9 <- which(X[, 1] == 7 | X[, 1] == 9)

X.train <- X[X_7_9, -1]
y.train <- X[X_7_9, 1] == 7

table(y.train)
```

```
## y.train
## FALSE  TRUE
##   644   645
```

```
# Load test data
X <- as.matrix(read.table(gzfile("zip.test")))
head(X)
```

```
##      V1 V2 V3 V4     V5   V6     V7     V8     V9    V10    V11    V12    V13
## [1,]  9 -1 -1 -1 -1.000 -1.0 -0.948 -0.561  0.148  0.384  0.904  0.290 -0.782
## [2,]  6 -1 -1 -1 -1.000 -1.0 -1.000 -1.000 -1.000 -1.000 -1.000 -1.000 -1.000
## [3,]  3 -1 -1 -1 -0.593  0.7  1.000  1.000  1.000  1.000  0.853  0.075 -0.925
## [4,]  6 -1 -1 -1 -1.000 -1.0 -1.000 -1.000 -1.000 -1.000 -1.000 -1.000 -1.000
## [5,]  6 -1 -1 -1 -1.000 -1.0 -1.000 -1.000 -0.858 -0.106  0.802 -0.210 -1.000
## [6,]  0 -1 -1 -1 -1.000 -1.0 -1.000  0.195  1.000  0.054 -1.000 -1.000 -1.000
##      V14 V15 V16 V17 V18 V19    V20    V21    V22    V23    V24    V25    V26
## [1,]  -1  -1  -1  -1  -1  -1 -1.000 -1.000 -0.748  0.588  1.000  1.000  0.991
## [2,]  -1  -1  -1  -1  -1  -1 -1.000 -1.000 -1.000 -0.783 -0.973 -1.000 -1.000
## [3,]  -1  -1  -1  -1  -1  -1 -0.553  0.998  1.000  1.000  1.000  1.000  1.000
## [4,]  -1  -1  -1  -1  -1  -1 -1.000 -1.000 -1.000 -1.000 -1.000 -1.000 -1.000
## [5,]  -1  -1  -1  -1  -1  -1 -1.000 -1.000 -1.000 -1.000 -0.854  0.597  1.000
## [6,]  -1  -1  -1  -1  -1  -1 -1.000 -1.000 -1.000 -0.801  0.790  1.000  0.856
##        V27    V28    V29    V30    V31 V32 V33 V34 V35    V36    V37    V38
## [1,]  0.915  1.000  0.931 -0.476 -1.000  -1  -1  -1  -1 -1.000 -0.787  0.794
## [2,] -1.000 -1.000 -1.000 -1.000 -1.000  -1  -1  -1  -1 -1.000 -1.000 -0.364
## [3,]  1.000  1.000  0.961 -0.076 -0.999  -1  -1  -1  -1  0.228  1.000  0.849
## [4,] -1.000 -1.000 -1.000 -1.000 -1.000  -1  -1  -1  -1 -1.000 -1.000 -1.000
## [5,]  0.798 -0.388 -1.000 -1.000 -1.000  -1  -1  -1  -1 -1.000 -1.000 -1.000
## [6,] -0.282 -0.831 -1.000 -1.000 -1.000  -1  -1  -1  -1 -1.000 -1.000 -0.937
##        V39    V40    V41    V42    V43    V44    V45    V46    V47 V48 V49 V50
```

```
## [4,]  1.000  1.000  1.000  1.000  1.000  0.896  0.177 -0.911 -1.000 -1.00
## [5,]  0.646  1.000  0.317 -0.926 -1.000 -0.849  0.598  1.000  0.169 -0.97
## [6,] -1.000 -1.000 -1.000 -0.740 -0.436  0.657  1.000  1.000  0.008 -1.00
##        V210   V211   V212   V213   V214   V215   V216   V217   V218   V219
## [1,] -1.000 -1.000 -1.000 -1.000 -1.000 -1.000 -1.000 -0.600  0.998  0.841
## [2,] -1.000 -0.640  0.661  0.971  1.000  1.000  1.000  0.950  0.774  0.774
## [3,] -1.000 -0.943  0.779  0.555 -0.333 -0.333 -0.333 -0.166  0.389  1.000
## [4,] -1.000 -1.000 -0.723 -0.451 -0.081 -0.611 -0.021 -0.414 -0.021 -0.182
## [5,]  0.631  1.000  0.754  0.046 -0.244 -0.661  0.984  1.000  0.142 -0.584
## [6,] -1.000 -1.000 -0.006  0.976  1.000  0.868  0.744  0.744  0.744  0.850
##        V220   V221 V222   V223   V224 V225   V226   V227   V228   V229   V230
## [1,] -0.932 -1.000   -1 -1.000 -1.000   -1 -1.000 -1.000 -1.000 -1.000 -1.000
## [2,]  0.302 -0.522   -1 -1.000 -1.000   -1 -1.000 -1.000 -1.000 -0.663 -0.606
## [3,]  1.000  1.000    1  0.497 -1.000   -1 -1.000 -1.000  0.507  1.000  1.000
## [4,] -0.648 -0.780   -1 -1.000 -1.000   -1 -1.000 -1.000 -1.000 -1.000 -1.000
## [5,]  0.075  0.833    1  0.123 -0.963   -1 -0.537  0.896  1.000  1.000  1.000
## [6,]  1.000  1.000    1  0.782 -0.736   -1 -1.000 -1.000 -1.000 -0.310  0.686
##        V231   V232   V233 V234   V235 V236  V237   V238   V239 V240 V241 V242
## [1,] -1.000 -1.000 -0.424    1  0.732   -1 -1.00 -1.000 -1.000   -1   -1   -1
## [2,] -0.606 -0.606 -0.688   -1 -1.000   -1 -1.00 -1.000 -1.000   -1   -1   -1
## [3,]  1.000  1.000  1.000    1  1.000    1  0.83  0.053 -0.946   -1   -1   -1
## [4,] -1.000 -1.000 -1.000   -1 -1.000   -1 -1.00 -1.000 -1.000   -1   -1   -1
## [5,]  1.000  1.000  1.000    1  1.000    1  0.83 -0.387 -0.976   -1   -1   -1
## [6,]  1.000  1.000  1.000    1  1.000    1  1.00  0.602 -0.906   -1   -1   -1
##        V243   V244   V245   V246   V247   V248   V249   V250   V251   V252
## [1,] -1.000 -1.000 -1.000 -1.000 -1.000 -1.000 -0.908  0.430  0.622 -0.973
## [2,] -1.000 -1.000 -1.000 -1.000 -1.000 -1.000 -1.000 -1.000 -1.000 -1.000
## [3,] -1.000 -0.941  0.059  0.615  1.000  1.000  0.717  0.333  0.162 -0.393
## [4,] -1.000 -1.000 -1.000 -1.000 -1.000 -1.000 -1.000 -1.000 -1.000 -1.000
## [5,] -0.697 -0.108  0.312  0.901  0.901  0.901  0.901  0.901  0.290 -0.369
## [6,] -1.000 -1.000 -1.000 -0.903  0.009  0.224  1.000  0.988  0.187  0.139
##        V253   V254 V255 V256 V257
## [1,] -1.000 -1.000   -1   -1   -1
## [2,] -1.000 -1.000   -1   -1   -1
## [3,] -1.000 -1.000   -1   -1   -1
## [4,] -1.000 -1.000   -1   -1   -1
## [5,] -0.867 -1.000   -1   -1   -1
## [6,] -0.641 -0.812   -1   -1   -1
```

```
dim(X)
```

```
## [1] 2007  257
```

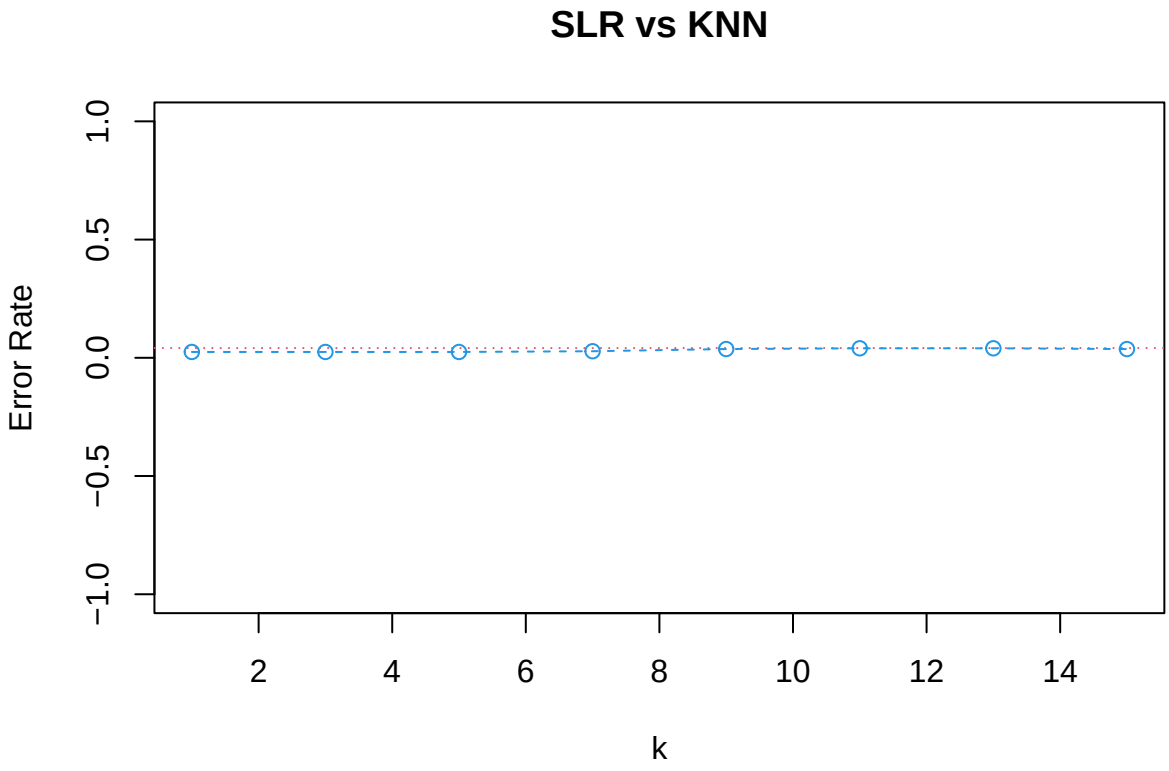
```
X_7_9 <- which(X[, 1] == 7 | X[, 1] == 9)

X.test <- X[X_7_9, -1]
y.test <- X[X_7_9, 1] == 7

table(y.test)
```

```
## y.test
## FALSE  TRUE
##   177   147
```

```r
# Linear Regression:

L <- lm(y.train ~ X.train)
yhat <- (cbind(1, X.test) %*% L$coef) >= 0.5
L.error <- mean(yhat != y.test)

# KNN:

library(class)
k <- c(1, 3, 5, 7, 9, 11, 13, 15)
k.error <- rep(NA, length(k))
for (i in 1:length(k)) {
    yhat <- knn(X.train, X.test, y.train, k[i])
    k.error[i] <- mean(yhat != y.test)
}

# Lets compare:

error <- matrix(c(L.error, k.error), ncol = 1)
colnames(error) <- c("Error Rate")
rownames(error) <- c("Linear Regression", paste("k-NN with k =", k))
error
```

```
##                   Error Rate
## Linear Regression         NA
## k-NN with k = 1   0.02469136
## k-NN with k = 3   0.02469136
## k-NN with k = 5   0.02469136
## k-NN with k = 7   0.02777778
## k-NN with k = 9   0.03703704
## k-NN with k = 11  0.04012346
## k-NN with k = 13  0.04012346
## k-NN with k = 15  0.03703704
```

```r
plot(c(1, 15), c(0, 1.1 * max(error)), type = "n", main = "SLR vs KNN",
     ylab = "Error Rate", xlab = "k")
abline(h = 0.04121, col = 2, lty = 3)
points(k, k.error, col = 4)
lines(k, k.error, col = 4, lty = 2)
```




```
# Conclusion:

# Here, both linear regression and KNN are performing nearly same with
# __`red line`__ indicating __`SLR`__ and __`blue line`__ indicating
# __`KNN`__ error values respectively. Both models error rate is close to zero.
```