DSCI 5260 Final Project - Group 5

ChintanR

11/17/2021

```
#Loading Packages
pacman::p_load(tidyr, mice, glmnet, shiny, tidyverse, caret, MASS, rmarkdown, leaflet, lattice,  rpart,
               reshape2, factoextra, readr, DataExplorer,skimr, lubridate, fpp3, GGally, gridExtra, tsi
options(max.print = 10000000)
```

```
#Loading libraries

library(githubinstall)
library(devtools)
```

```
## Loading required package: usethis
```

```
library(mice)
library(readr)
library(VIM)
```

```
## Loading required package: colorspace

## Loading required package: grid

## VIM is ready to use.

## Suggestions and bug-reports can be submitted at: https://github.com/statistikat/VIM/issues

##
## Attaching package: 'VIM'

## The following object is masked from 'package:datasets':
##
##     sleep
```

```
library(ggplot2)
library(pacman)
library(cluster)
library(factoextra)
library(ISLR)
library(MASS)
library(readr)
library(devtools)
library(data.table)
```

```
##
## Attaching package: 'data.table'

## The following object is masked from 'package:tsibble':
##
##     key

## The following objects are masked from 'package:lubridate':
##
##     hour, isoweek, mday, minute, month, quarter, second, wday, week,
##     yday, year

## The following objects are masked from 'package:reshape2':
##
##     dcast, melt

## The following objects are masked from 'package:dplyr':
##
##     between, first, last

## The following object is masked from 'package:purrr':
##
##     transpose
```

```
library(dplyr)
library(plyr)
library(skimr)
library(DataExplorer)
```

```
#Loading in the data
cardata <- read.csv("car_data_df.csv")
head(cardata)
```

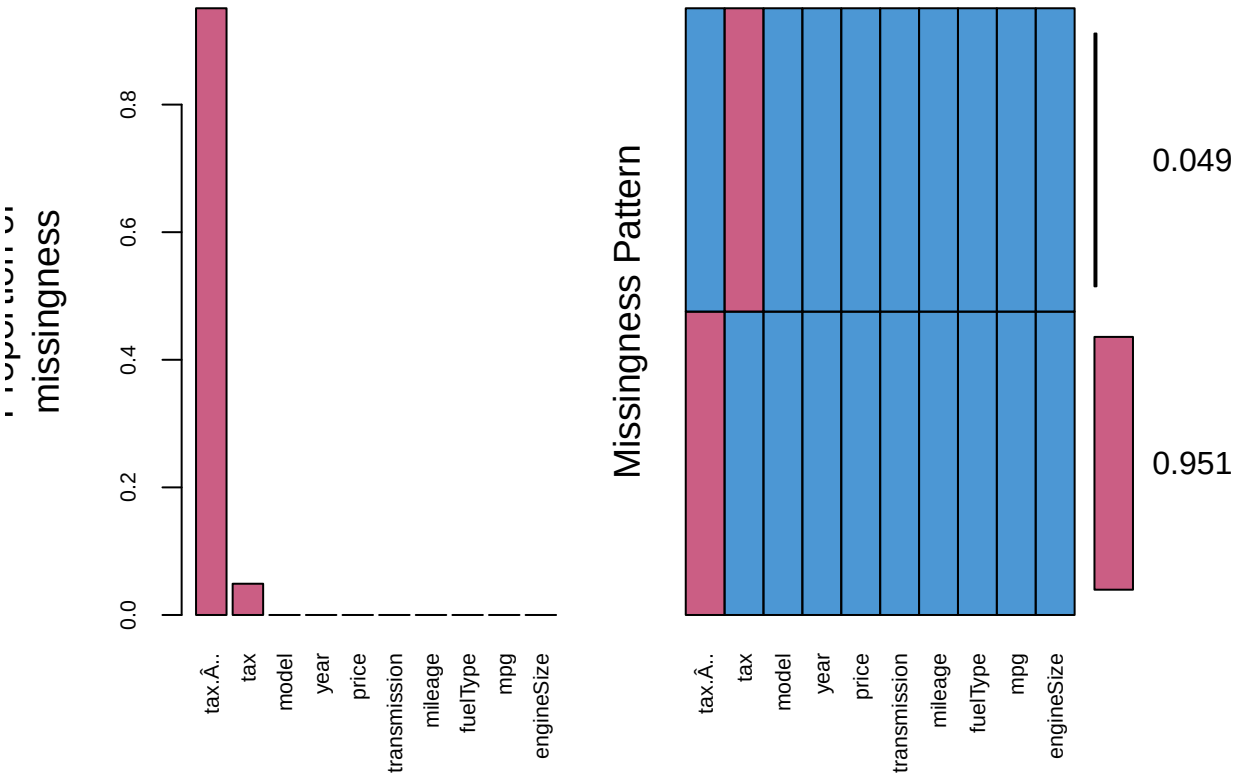
```
##   model year price transmission mileage fuelType tax  mpg engineSize tax.Â..
## 1    A1 2017 12500       Manual   15735   Petrol 150 55.4        1.4      NA
## 2    A6 2016 16500    Automatic   36203   Diesel  20 64.2        2.0      NA
## 3    A1 2016 11000       Manual   29946   Petrol  30 55.4        1.4      NA
## 4    A4 2017 16800    Automatic   25952   Diesel 145 67.3        2.0      NA
## 5    A3 2019 17300       Manual    1998   Petrol 145 49.6        1.0      NA
## 6    A1 2016 13900    Automatic   32260   Petrol  30 58.9        1.4      NA
```

```
#Searching for Missing Values
summary(cardata)
```

```
##     model                year          price        transmission
##  Length:396748      Min.   :1970   Min.   :   450   Length:396748
##  Class :character   1st Qu.:2016   1st Qu.:  9999   Class :character
##  Mode  :character   Median :2017   Median : 14495   Mode  :character
##                     Mean   :2017   Mean   : 16805
##                     3rd Qu.:2019   3rd Qu.: 20870
##                     Max.   :2060   Max.   :159999
```

```
##
##     mileage          fuelType              tax              mpg
##  Min.   :     1   Length:396748      Min.   :  0.0   Min.   :  0.30
##  1st Qu.:  7424   Class :character   1st Qu.:125.0   1st Qu.: 47.10
##  Median : 17460   Mode  :character   Median :145.0   Median : 54.30
##  Mean   : 23059                      Mean   :120.3   Mean   : 55.17
##  3rd Qu.: 32340                      3rd Qu.:145.0   3rd Qu.: 62.80
##  Max.   :323000                      Max.   :580.0   Max.   :470.80
##                                      NA's   :19440
##    engineSize       tax.Â..
##  Min.   :0.000   Min.   :  0.0
##  1st Qu.:1.200   1st Qu.:125.0
##  Median :1.600   Median :145.0
##  Mean   :1.663   Mean   :121.1
##  3rd Qu.:2.000   3rd Qu.:145.0
##  Max.   :6.600   Max.   :555.0
##                  NA's   :377308
```

```
cardata_Miss = aggr(cardata, col=mdc(1:2),
                            numbers=TRUE, sortVars=TRUE,
                            labels=names(cardata), cex.axis=.7,
                            gap=3, ylab=c("Proportion of
missingness","Missingness Pattern"))
```



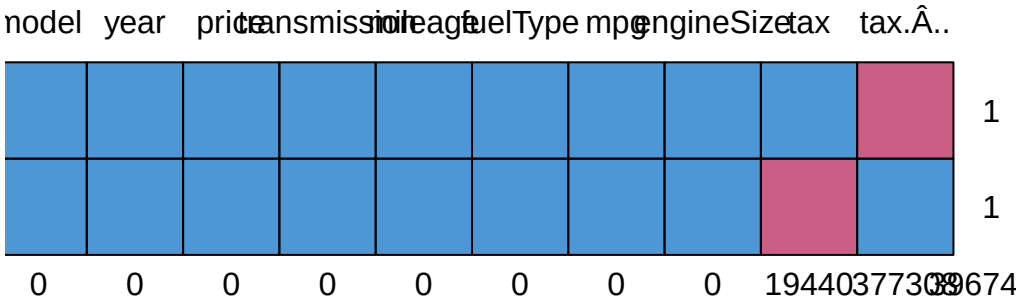

```
##
```

```
##  Variables sorted by number of missings:
##      Variable      Count
##       tax.Â.. 0.95100164
##           tax 0.04899836
##         model 0.00000000
##          year 0.00000000
##         price 0.00000000
##  transmission 0.00000000
##       mileage 0.00000000
##      fuelType 0.00000000
##           mpg 0.00000000
##    engineSize 0.00000000
```

```
p <- function(x) {sum(is.na(x))/length(x)*100}
apply(cardata, 2, p)
```

```
##        model         year        price transmission      mileage     fuelType
##     0.000000     0.000000     0.000000     0.000000     0.000000     0.000000
##          tax          mpg   engineSize      tax.Â..
##     4.899836     0.000000     0.000000    95.100164
```

```
md.pattern(cardata, plot = TRUE)
```




```
##        model year price transmission mileage fuelType mpg engineSize   tax
```

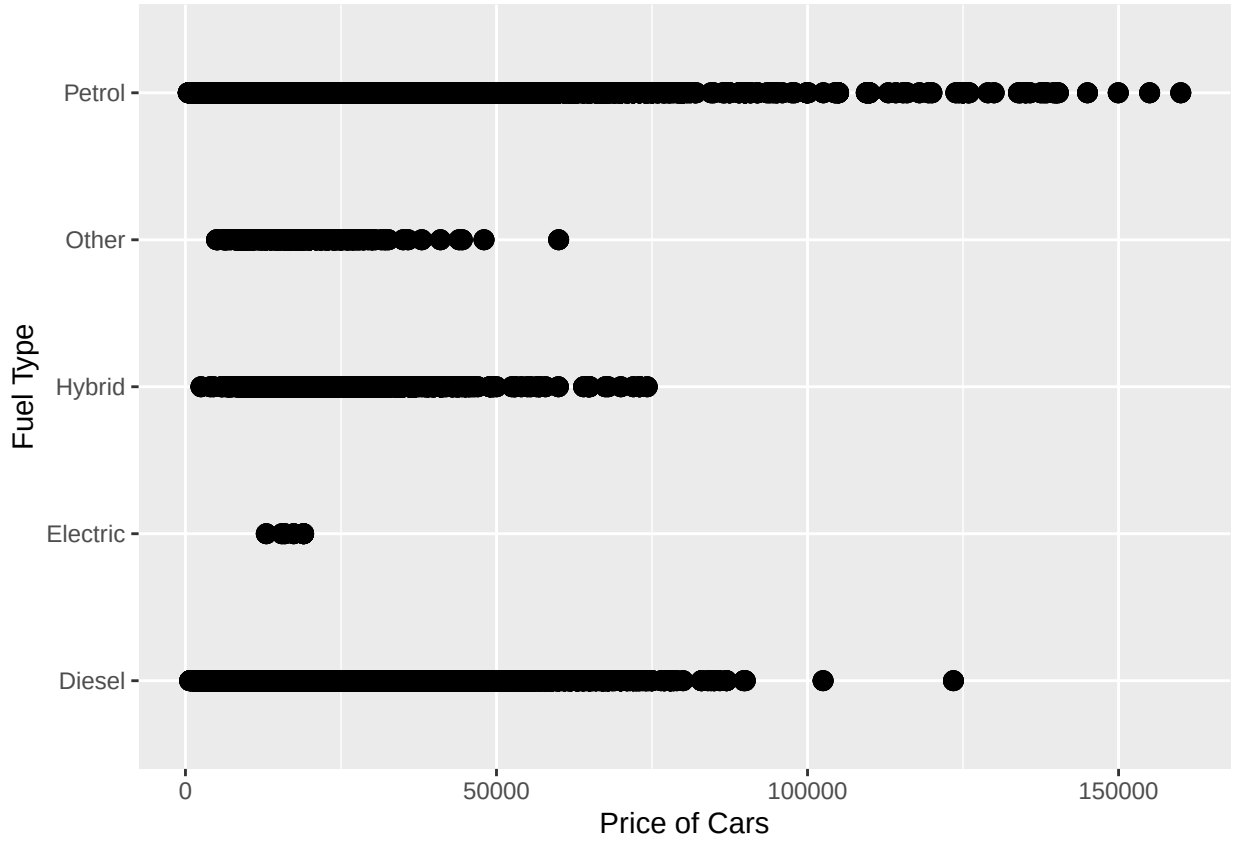
```
## 377308     1    1     1            1       1        1   1          1     1
## 19440      1    1     1            1       1        1   1          1     0
##            0    0     0            0       0        0   0          0 19440
##        tax.Ã..
## 377308       0      1
## 19440        1      1
##         377308 396748
```

##We can see 377308 missing values in TAx.A.. column. We drop it.

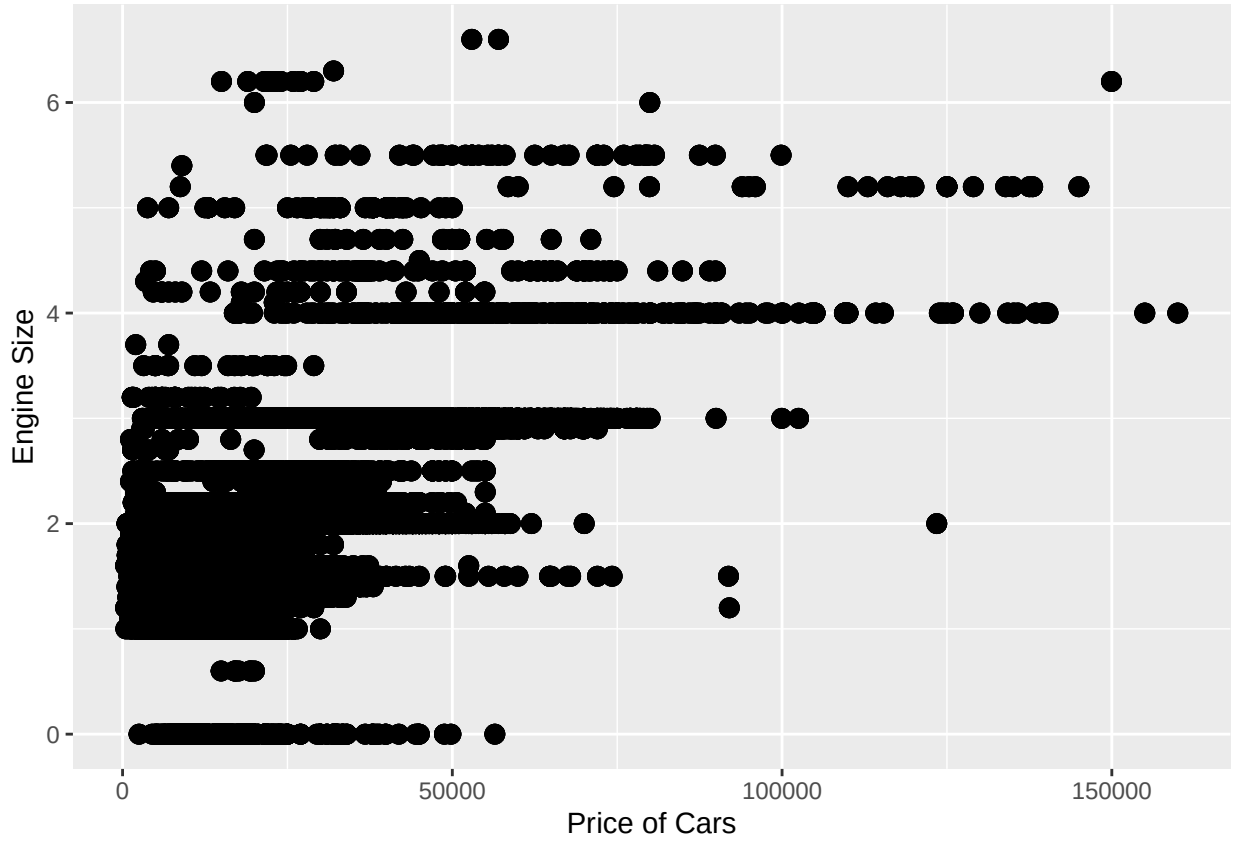
```
#Change the fuelType and Transmission from chr to factors
cardata$fuelType <- factor(cardata$fuelType)
cardata$transmission <- as.factor(cardata$transmission)
cardata <- cardata[-c(10)]
summary(cardata)
```

```
##     model                year          price           transmission
##  Length:396748      Min.   :1970   Min.   :   450   Automatic: 80224
##  Class :character   1st Qu.:2016   1st Qu.:  9999   Manual   :225780
##  Mode  :character   Median :2017   Median : 14495   Other    :    36
##                     Mean   :2017   Mean   : 16805   Semi-Auto: 90708
##                     3rd Qu.:2019   3rd Qu.: 20870
##                     Max.   :2060   Max.   :159999
##
##     mileage           fuelType           tax             mpg
##  Min.   :     1   Diesel  :163712   Min.   :  0.0   Min.   :  0.30
##  1st Qu.:  7424   Electric:    24   1st Qu.:125.0   1st Qu.: 47.10
##  Median : 17460   Hybrid  : 12312   Median :145.0   Median : 54.30
##  Mean   : 23059   Other   :   988   Mean   :120.3   Mean   : 55.17
##  3rd Qu.: 32340   Petrol  :219712   3rd Qu.:145.0   3rd Qu.: 62.80
##  Max.   :323000                     Max.   :580.0   Max.   :470.80
##                                     NA's   :19440
##    engineSize
##  Min.   :0.000
##  1st Qu.:1.200
##  Median :1.600
##  Mean   :1.663
##  3rd Qu.:2.000
##  Max.   :6.600
##
```

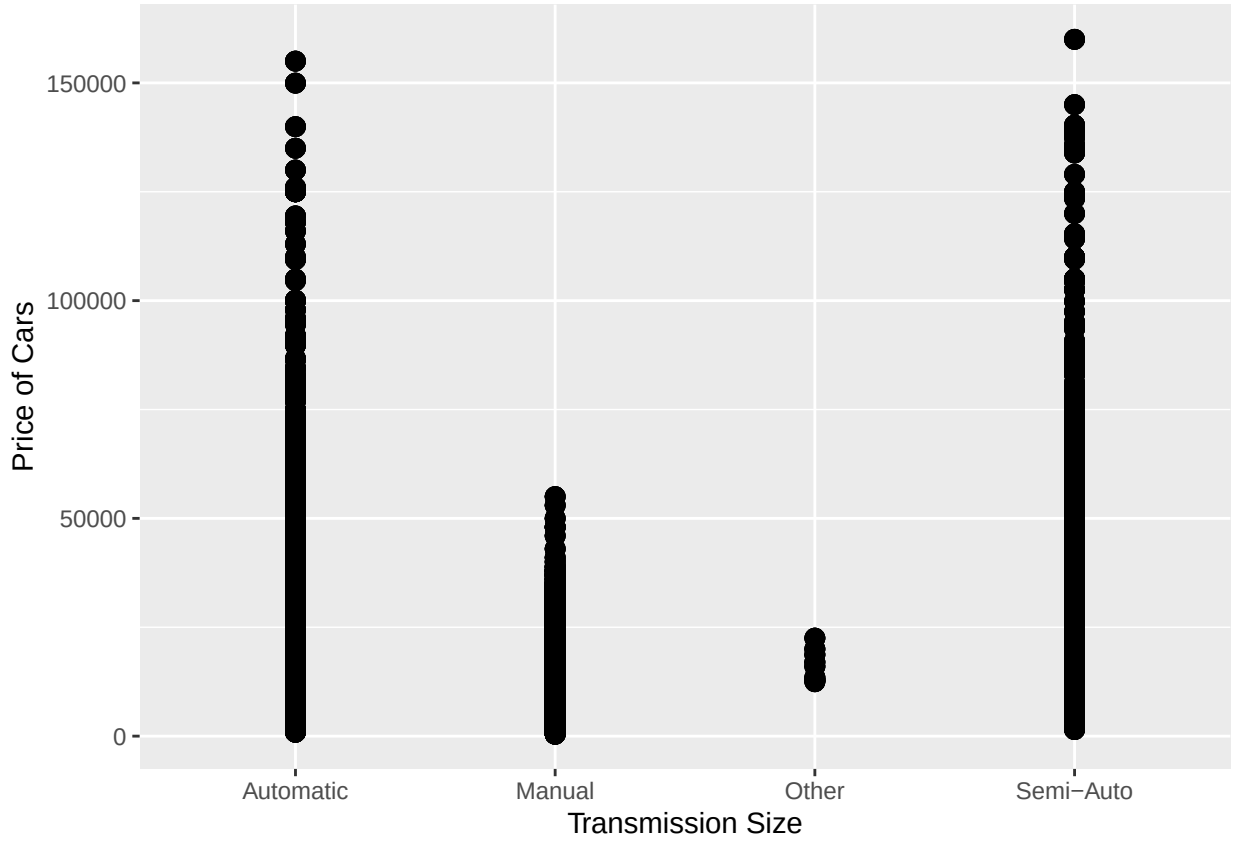
```
cardata.df.plot1 <- ggplot(cardata) +
  geom_point(aes(price, fuelType), size = 3, shape = 19) +
  scale_color_manual(values=c('red', 'blue')) +
  xlab("Price of Cars") +
  ylab("Fuel Type")
cardata.df.plot1
```



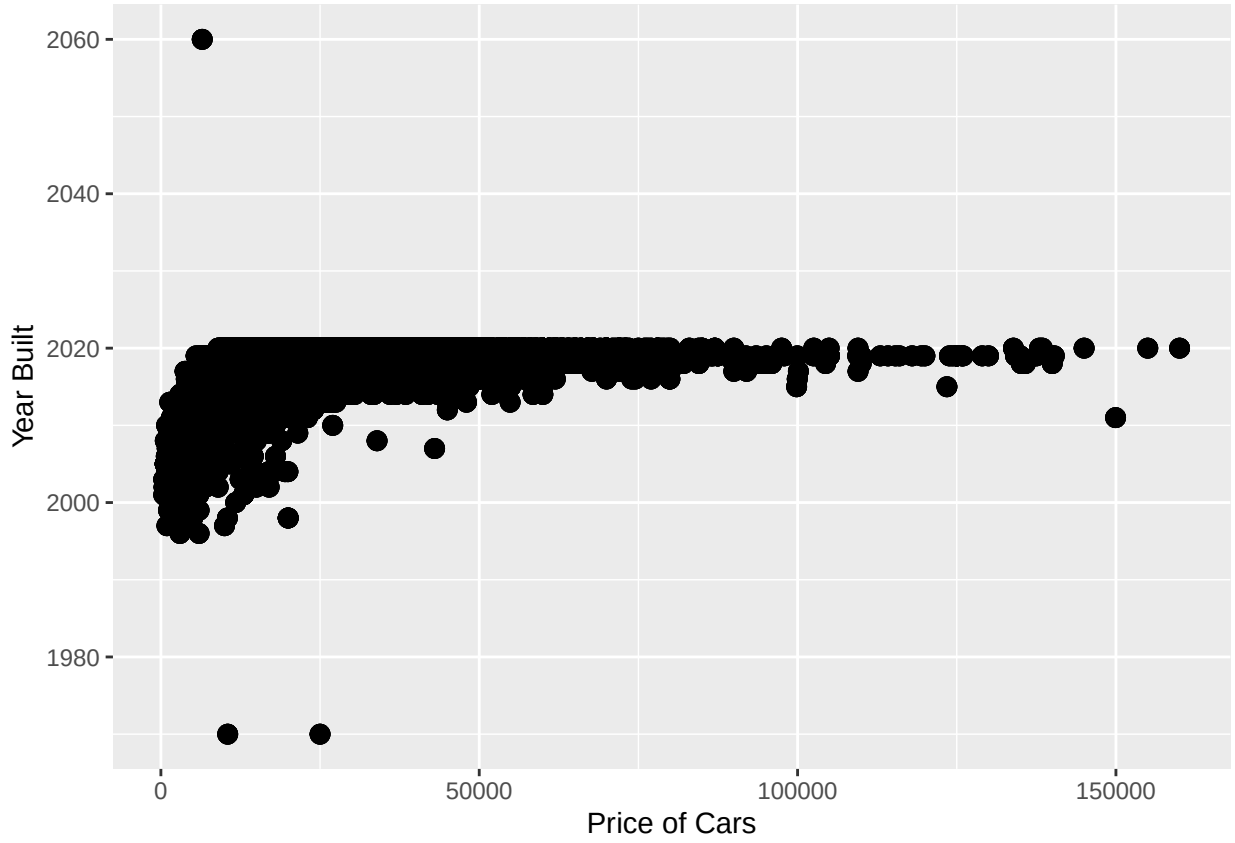

```
cardata.df.plot2 <- ggplot(cardata) +
  geom_point(aes(price, engineSize), size = 3, shape = 19) +
  scale_color_manual(values=c('red', 'blue')) +
  xlab("Price of Cars") +
  ylab("Engine Size")
cardata.df.plot2
```



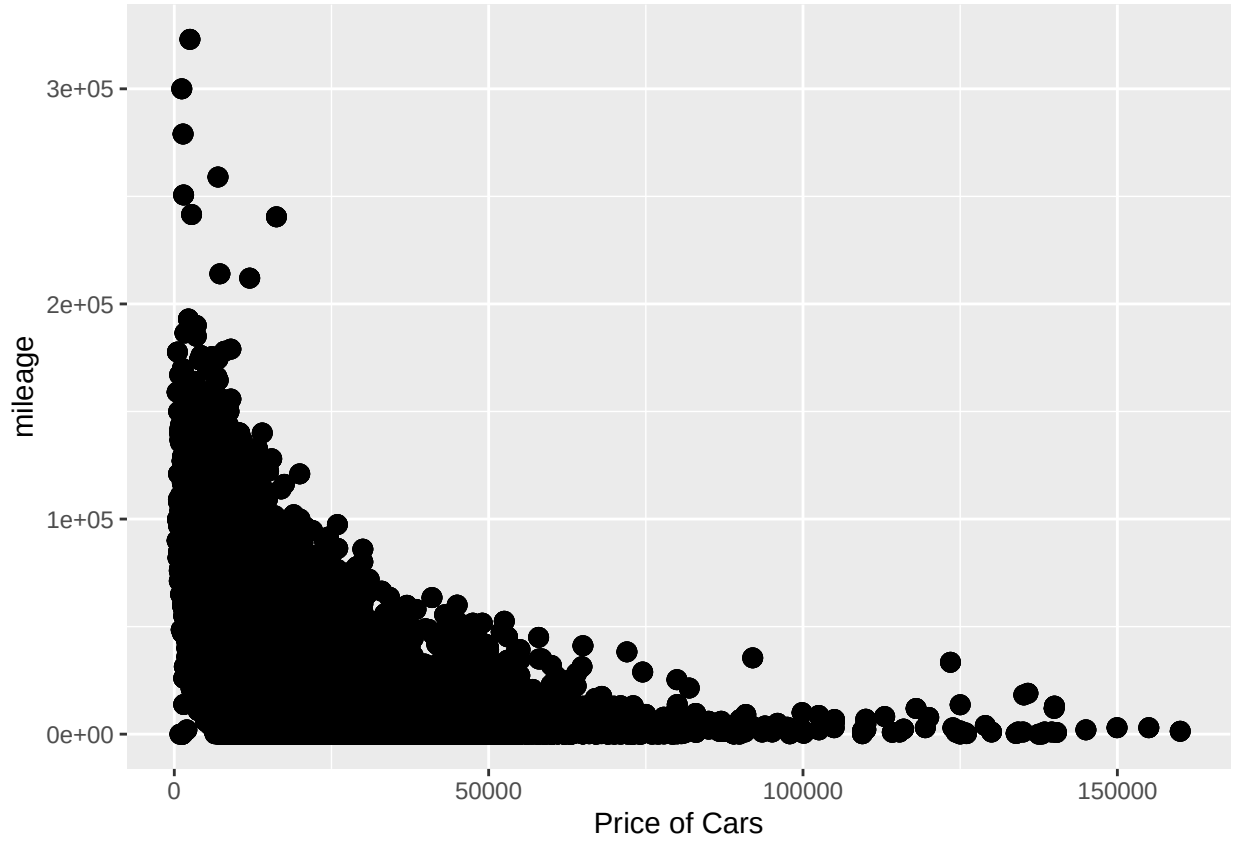

```
cardata.df.plot3 <- ggplot(cardata) +
  geom_point(aes(transmission, price), size = 3, shape = 19) +
  scale_color_manual(values=c('red', 'blue')) +
  xlab("Transmission Size") +
  ylab("Price of Cars")
cardata.df.plot3
```



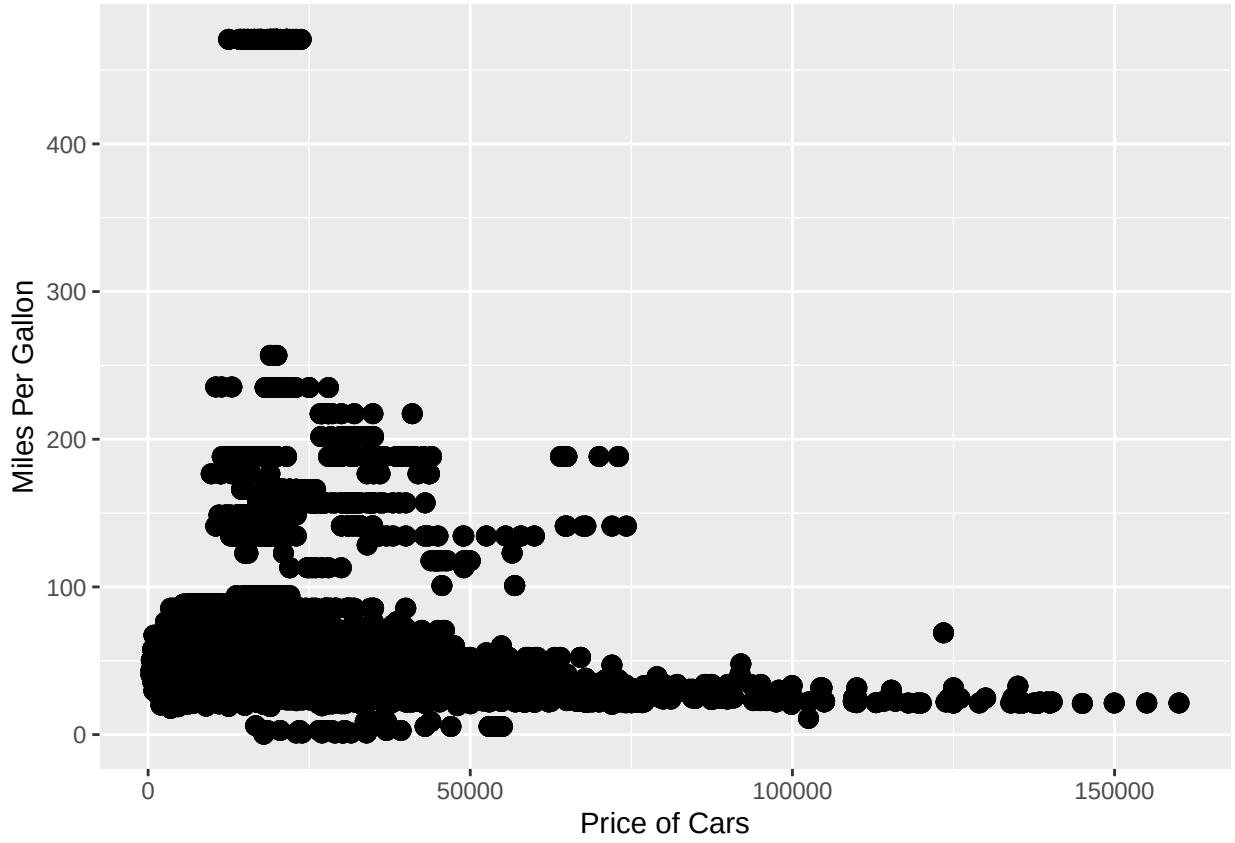

```
cardata.df.plot4 <- ggplot(cardata) +
  geom_point(aes(price, year), size = 3, shape = 19) +
  scale_color_manual(values=c('red', 'blue')) +
  xlab("Price of Cars") +
  ylab("Year Built")
cardata.df.plot4
```




```
cardata.df.plot5 <- ggplot(cardata) +
  geom_point(aes(price, mileage), size = 3, shape = 19) +
  scale_color_manual(values=c('red', 'blue')) +
  xlab("Price of Cars") +
  ylab("mileage")
cardata.df.plot5
```




```
cardata.df.plot6 <- ggplot(cardata) +
  geom_point(aes(price, mpg), size = 3, shape = 19) +
  scale_color_manual(values=c('red', 'blue')) +
  xlab("Price of Cars") +
  ylab("Miles Per Gallon")
cardata.df.plot6
```




```
#Looking at the range of year and changing the uncorrect year value
range(cardata$year)
```

```
## [1] 1970 2060
```

```
cardata$year[cardata$year == 2060] <- 2017
```

```
#Looking at the summary of year and MPG
summary(cardata$year)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    1970    2016    2017    2017    2019    2020
```

```
summary(cardata$mpg)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    0.30   47.10   54.30   55.17   62.80  470.80
```

```
#Imputing the missing values in Tax with mean
cardata$tax[is.na(cardata$tax)] <- 120.3
summary(cardata)
```

```
##     model                year          price          transmission
##  Length:396748      Min.   :1970   Min.   :   450   Automatic: 80224
##  Class :character   1st Qu.:2016   1st Qu.:  9999   Manual   :225780
##  Mode  :character   Median :2017   Median : 14495   Other    :    36
##                     Mean   :2017   Mean   : 16805   Semi-Auto: 90708
##                     3rd Qu.:2019   3rd Qu.: 20870
##                     Max.   :2020   Max.   :159999
##     mileage          fuelType          tax             mpg
##  Min.   :     1   Diesel  :163712   Min.   :  0.0   Min.   :  0.30
##  1st Qu.:  7424   Electric:    24   1st Qu.:120.3   1st Qu.: 47.10
##  Median : 17460   Hybrid  : 12312   Median :145.0   Median : 54.30
##  Mean   : 23059   Other   :   988   Mean   :120.3   Mean   : 55.17
##  3rd Qu.: 32340   Petrol  :219712   3rd Qu.:145.0   3rd Qu.: 62.80
##  Max.   :323000                     Max.   :580.0   Max.   :470.80
##    engineSize
##  Min.   :0.000
##  1st Qu.:1.200
##  Median :1.600
##  Mean   :1.663
##  3rd Qu.:2.000
##  Max.   :6.600
```

```
#Running Principal Component Analysis
str(cardata)
```

```
## 'data.frame':    396748 obs. of  9 variables:
##  $ model       : chr  "A1" "A6" "A1" "A4" ...
##  $ year        : num  2017 2016 2016 2017 2019 ...
##  $ price       : int  12500 16500 11000 16800 17300 13900 13250 11750 10200 12000 ...
##  $ transmission: Factor w/ 4 levels "Automatic","Manual",..: 2 1 2 1 2 1 1 2 2 2 ...
##  $ mileage     : int  15735 36203 29946 25952 1998 32260 76788 75185 46112 22451 ...
##  $ fuelType    : Factor w/ 5 levels "Diesel","Electric",..: 5 1 5 1 5 5 1 1 5 5 ...
##  $ tax         : num  150 20 30 145 145 30 30 20 20 30 ...
##  $ mpg         : num  55.4 64.2 55.4 67.3 49.6 58.9 61.4 70.6 60.1 55.4 ...
##  $ engineSize  : num  1.4 2 1.4 2 1 1.4 2 2 1.4 1.4 ...
```

```
pcs <- prcomp(cardata[,-c(1,4,6)],scale. = T)
summary(pcs)
```

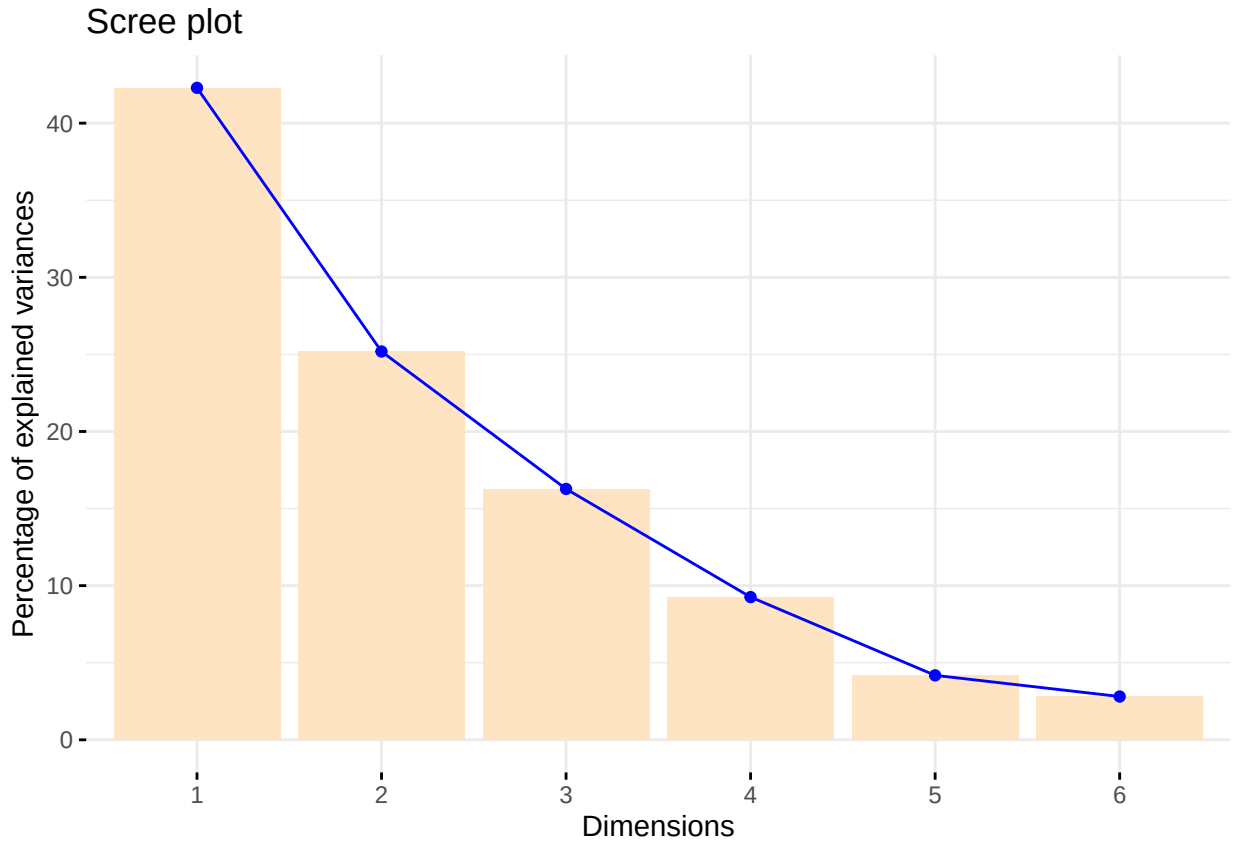
```
## Importance of components:
##                           PC1    PC2    PC3     PC4     PC5     PC6
## Standard deviation     1.5929 1.2294 0.9881 0.74530 0.50087 0.41044
## Proportion of Variance 0.4229 0.2519 0.1627 0.09258 0.04181 0.02808
## Cumulative Proportion  0.4229 0.6748 0.8375 0.93011 0.97192 1.00000
```

```
pcs$rot
```

```
##                  PC1        PC2        PC3         PC4          PC5
## year        0.4417368 -0.4879535  0.1168747 -0.01570584 -0.682697466
## price       0.5226547  0.1461787  0.4365598  0.06677394 -0.001391164
## mileage    -0.4282324  0.5129753  0.0344070 -0.02500506 -0.718493010
## tax         0.3727545  0.2413916 -0.5236387 -0.72335580 -0.032582468
```

```
## mpg        -0.3466827 -0.2677013  0.5758340 -0.68621182  0.072407347
## engineSize  0.2986350  0.5894537  0.4344993 -0.02321716  0.106703271
##                    PC6
## year       -0.29459773
## price       0.71443592
## mileage     0.18823389
## tax         0.06543257
## mpg         0.02080436
## engineSize -0.60220200
```

```
#Plotting PCA
fviz_eig(pcs, barfill = "bisque", barcolor = "NA", linecolor = "blue")
```




```
#Splitting Data
ind <- sample(2, nrow(cardata),replace = TRUE, prob =
               c(0.8,0.2))
traindata <- cardata[ind==1,]
testdata <- cardata[ind==2,]
head(traindata)
```

```
##   model year price transmission mileage fuelType tax  mpg engineSize
## 1    A1 2017 12500       Manual   15735   Petrol 150 55.4        1.4
## 2    A6 2016 16500    Automatic   36203   Diesel  20 64.2        2.0
## 4    A4 2017 16800    Automatic   25952   Diesel 145 67.3        2.0
## 5    A3 2019 17300       Manual    1998   Petrol 145 49.6        1.0
```

```
## 6    A1 2016 13900    Automatic   32260   Petrol  30 58.9        1.4
## 7    A6 2016 13250    Automatic   76788   Diesel  30 61.4        2.0
```

```
head(testdata)
```

```
##    model year price transmission mileage fuelType tax  mpg engineSize
## 3     A1 2016 11000       Manual   29946   Petrol  30 55.4        1.4
## 15    A6 2015 15400       Manual   47348   Diesel  30 61.4        2.0
## 26    A4 2017 18500    Automatic   17418   Diesel 145 62.8        2.0
## 27    A5 2017 19500    Automatic   33300   Diesel 145 61.4        2.0
## 38    A6 2016 19400    Automatic   34030   Diesel 125 58.9        2.0
## 40    A3 2017 17100       Manual   29545   Diesel 145 65.7        2.0
```

```
#Running Linear Regression Model
Model1<-lm(price ~ year + mpg + engineSize + mileage, data = traindata)

summary(Model1)
```

```
##
## Call:
## lm(formula = price ~ year + mpg + engineSize + mileage, data = traindata)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -42262  -3094   -444   2310 107881
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -3.208e+06  1.360e+04 -235.87   <2e-16 ***
## year         1.591e+03  6.738e+00  236.10   <2e-16 ***
## mpg         -2.459e+01  6.211e-01  -39.59   <2e-16 ***
## engineSize   1.178e+04  1.800e+01  654.29   <2e-16 ***
## mileage     -1.075e-01  6.884e-04 -156.10   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 5390 on 317294 degrees of freedom
## Multiple R-squared:  0.7013, Adjusted R-squared:  0.7013
## F-statistic: 1.863e+05 on 4 and 317294 DF,  p-value: < 2.2e-16
```

We got RSE of 70.08%

```
#Using Prediction Model to our test Data
LMPred <- predict(Model1, testdata)

actuals_preds <- data.frame(cbind(actuals=testdata$price, predicteds=LMPred))
# make actuals_predicteds dataframe.
correlation_accuracy <- cor(actuals_preds)
correlation_accuracy
```

```
##             actuals predicteds
## actuals    1.000000   0.838591
## predicteds 0.838591   1.000000
```

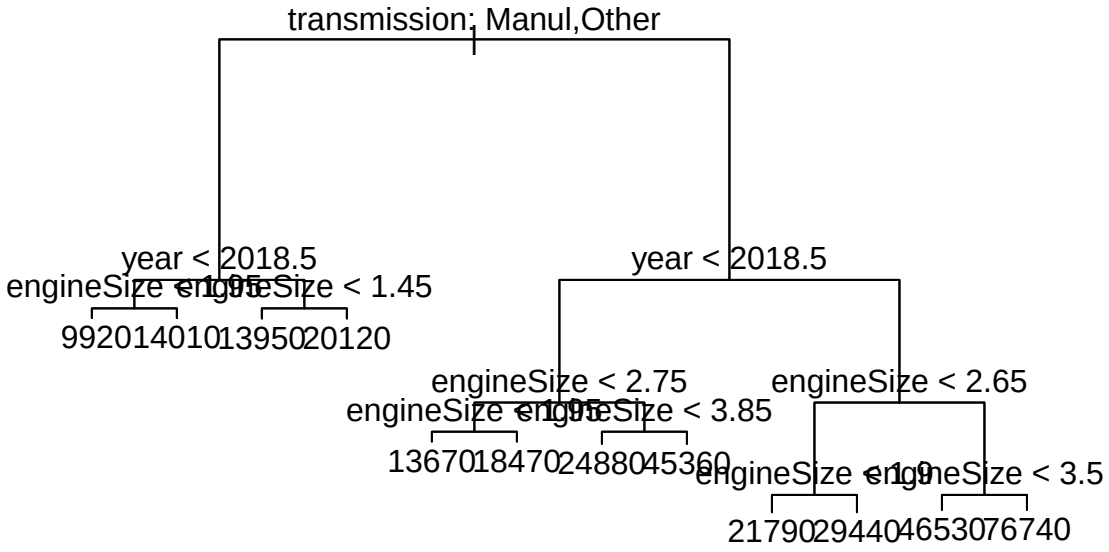
```
#Accuracy of 54% with Linear Regression Model
```

```
#Decision Tree Model
set.seed(42)

tree.train <- tree(price~ year + engineSize+transmission+fuelType + mileage + mpg,traindata)
summary(tree.train)
```

```
##
## Regression tree:
## tree(formula = price ~ year + engineSize + transmission + fuelType +
##     mileage + mpg, data = traindata)
## Variables actually used in tree construction:
## [1] "transmission" "year"         "engineSize"
## Number of terminal nodes:  12
## Residual mean deviance:  26120000 = 8.287e+12 / 317300
## Distribution of residuals:
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
## -41870.0  -2921.0   -313.6      0.0   2515.0 105000.0
```

```
#Plotting Decision Train
plot(tree.train)
text(tree.train, pretty = 5)
```




```
#Using Drecision Tree Model to predict
p <- predict(tree.train, testdata)
summary(p)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    9920    9920   13951   16780   20120   76744
```

```
Tree_preds <- data.frame(cbind(actual=testdata$price, predicted=p))
correlation_acc <- cor(Tree_preds)
correlation_acc
```

```
##              actual predicted
## actual    1.0000000 0.8529109
## predicted 0.8529109 1.0000000
```

We get an accuracy of 54%

```
#Ridge Regression Model
y = traindata$price
x = data.matrix(traindata[,c('year', 'engineSize', 'tax', 'mileage', 'mpg')])
model <- glmnet(x, y , alpha = 0)
summary(model)
```

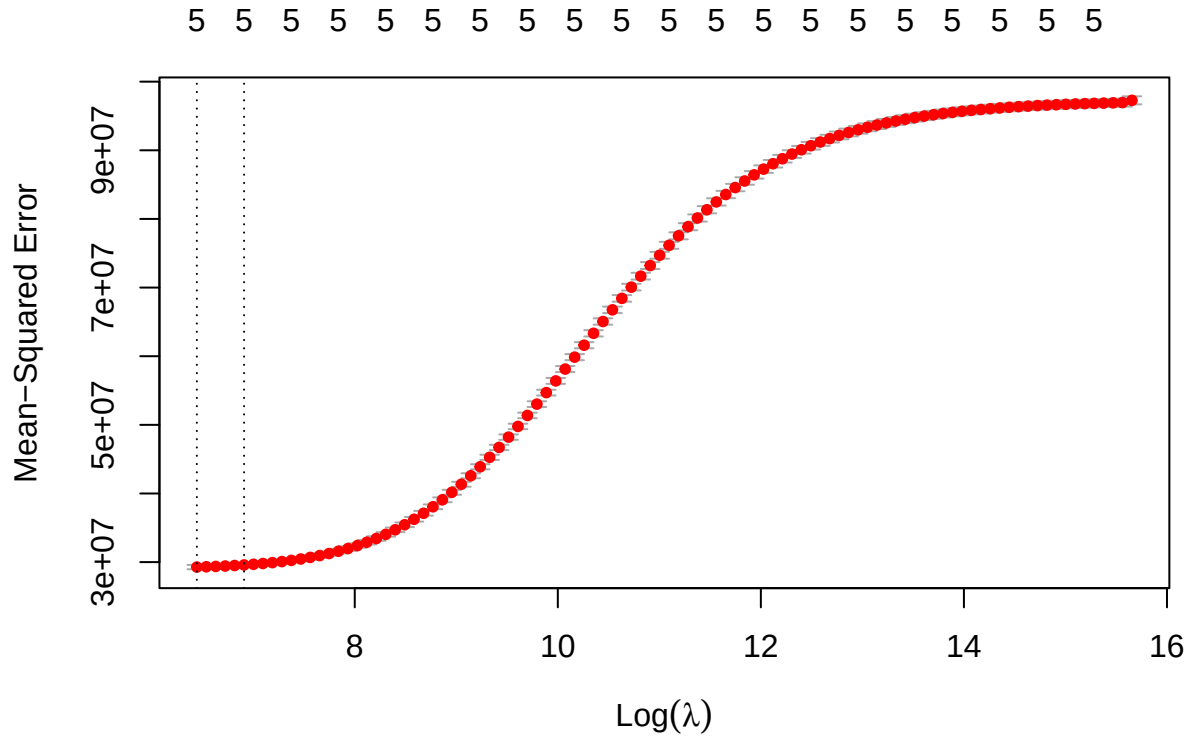
```
##           Length Class     Mode
## a0        100    -none-    numeric
## beta      500    dgCMatrix S4
## df        100    -none-    numeric
## dim         2    -none-    numeric
## lambda    100    -none-    numeric
## dev.ratio 100    -none-    numeric
## nulldev     1    -none-    numeric
## npasses     1    -none-    numeric
## jerr        1    -none-    numeric
## offset      1    -none-    logical
## call        4    -none-    call
## nobs        1    -none-    numeric
```

```
#k-fold cross-validation to find optimal lambda value
cv_model <- cv.glmnet(x, y, alpha = 0)

#find optimal lambda value that minimizes test MSE
best_lambda <- cv_model$lambda.min
best_lambda
```

```
## [1] 629.3524
```

```
#produce plot of test MSE by lambda value
plot(cv_model)
```



```
#find coefficients of best model
best_model <- glmnet(x, y, alpha = 0, lambda = best_lambda)
coef(best_model)
```

```
## 6 x 1 sparse Matrix of class "dgCMatrix"
##                         s0
## (Intercept) -3.036571e+06
## year         1.506718e+03
## engineSize   1.099227e+04
## tax          2.151072e-01
## mileage     -1.037568e-01
## mpg         -3.120202e+01
```

```
#use fitted best model to make predictions
d = data.matrix(testdata[,c('year', 'engineSize', 'tax', 'mileage', 'mpg')])
y_predicted <- predict(model, s = best_lambda, newx= d)
Ridge_preds <- data.frame(cbind(actual=testdata$price, prediction=y_predicted))

Ridge_accuracy <- cor(Ridge_preds)
Ridge_accuracy
```

```
##           actual        s1
## actual 1.0000000 0.8385805
## s1     0.8385805 1.0000000
```

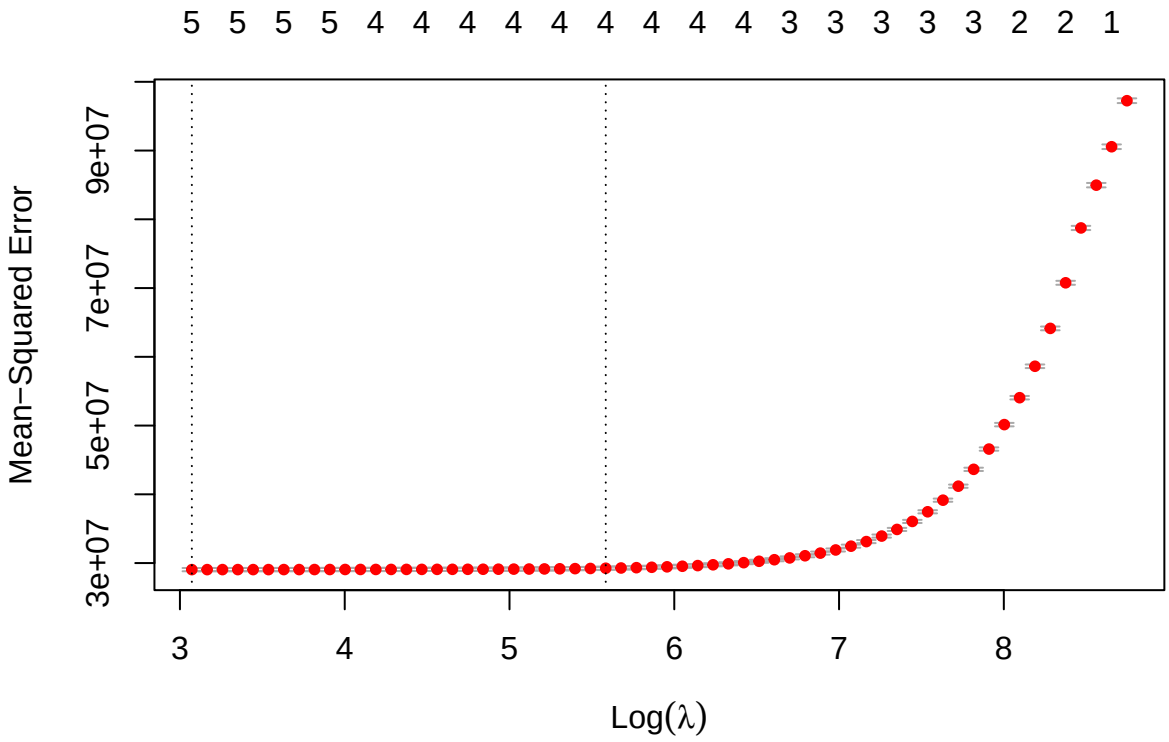
We get an accuracy of 54%

```
#LASSO Regression

LR_cv_model <- cv.glmnet(x, y, alpha = 1)
best_lambda_LR <- LR_cv_model$lambda.min
best_lambda_LR
```

```
## [1] 21.58974
```

```
#produce plot of test MSE by lambda value
plot(LR_cv_model)
```




```
LR_best_model <- glmnet(x, y, alpha = 1, lambda = best_lambda_LR)
coef(LR_best_model)
```

```
## 6 x 1 sparse Matrix of class "dgCMatrix"
##                         s0
## (Intercept) -3.203484e+06
## year         1.588826e+03
## engineSize   1.177348e+04
## tax         -1.557637e+00
## mileage     -1.073821e-01
## mpg         -2.593734e+01
```

```
#use fitted best model to make predictions
y_predicted_LR <- predict(LR_best_model, s = best_lambda_LR, newx = d)

Lasso_preds <- data.frame(cbind(actual=testdata$price, prediction=y_predicted_LR))

Lasso_accuracy <- cor(Lasso_preds)
Lasso_accuracy
```

```
##           actual        s1
## actual 1.0000000 0.8387469
## s1     0.8387469 1.0000000
```

We get an accuracy of 54%

```
create_report(cardata)
```

```
##
##
## processing file: report.rmd

##   |                                                                      |
##    inline R code fragments
##
##   |                                                                      |...
## label: global_options (with options)
## List of 1
##  $ include: logi FALSE
##
##   |                                                                      |.....
##   ordinary text without R code
##
##   |                                                                      |.......
## label: introduce
##   |                                                                      |........
##   ordinary text without R code
##
##   |                                                                      |.........
## label: plot_intro

##   |                                                                      |............
##   ordinary text without R code
##
##   |                                                                      |............
## label: data_structure
##   |                                                                      |..............
##   ordinary text without R code
##
##   |                                                                      |...............
## label: missing_profile

##   |                                                                      |.................
##   ordinary text without R code
##
##   |                                                                      |.................
## label: univariate_distribution_header
##   |                                                                      |.................
##   ordinary text without R code
##
##   |                                                                      |.................
## label: plot_histogram

##   |                                                                      |.................
##   ordinary text without R code
```

```
##
##   |                                                                      |.....................
## label: plot_density
##   |                                                                      |.....................
##   ordinary text without R code
##
##   |                                                                      |.....................
## label: plot_frequency_bar

##   |                                                                      |.....................
##   ordinary text without R code
##
##   |                                                                      |.....................
## label: plot_response_bar
##   |                                                                      |.....................
##   ordinary text without R code
##
##   |                                                                      |.....................
## label: plot_with_bar
##   |                                                                      |.....................
##   ordinary text without R code
##
##   |                                                                      |.....................
## label: plot_normal_qq

##   |                                                                      |.....................
##   ordinary text without R code
##
##   |                                                                      |.....................
## label: plot_response_qq
##   |                                                                      |.....................
##   ordinary text without R code
##
##   |                                                                      |.....................
## label: plot_by_qq
##   |                                                                      |.....................
##   ordinary text without R code
##
##   |                                                                      |.....................
## label: correlation_analysis

##   |                                                                      |.....................
##   ordinary text without R code
##
##   |                                                                      |.....................
## label: principal_component_analysis

##   |                                                                      |.....................
##   ordinary text without R code
##
##   |                                                                      |.....................
## label: bivariate_distribution_header
##   |                                                                      |.....................
```

```
##   ordinary text without R code
##
##   |                                                                                                  |.................
## label: plot_response_boxplot
##   |                                                                                                  |.................
##   ordinary text without R code
##
##   |                                                                                                  |.................
## label: plot_by_boxplot
##   |                                                                                                  |.................
##   ordinary text without R code
##
##   |                                                                                                  |.................
## label: plot_response_scatterplot
##   |                                                                                                  |.................
##   ordinary text without R code
##
##   |                                                                                                  |.................
## label: plot_by_scatterplot

## output file: C:/Users/loday/Downloads/report.knit.md

## "C:/Program Files/RStudio/bin/pandoc/pandoc" +RTS -K512m -RTS "C:/Users/loday/Downloads/report.knit.m

##
## Output created: report.html
```