RLab6 - Demonstrate the use of Ridge Regression

Mayuri Salecha -2048043

26 February 2021

# Question 1

#Load mtcars dataset  
data(mtcars)  
head(mtcars)

## mpg cyl disp hp drat wt qsec vs am gear carb  
## Mazda RX4 21.0 6 160 110 3.90 2.620 16.46 0 1 4 4  
## Mazda RX4 Wag 21.0 6 160 110 3.90 2.875 17.02 0 1 4 4  
## Datsun 710 22.8 4 108 93 3.85 2.320 18.61 1 1 4 1  
## Hornet 4 Drive 21.4 6 258 110 3.08 3.215 19.44 1 0 3 1  
## Hornet Sportabout 18.7 8 360 175 3.15 3.440 17.02 0 0 3 2  
## Valiant 18.1 6 225 105 2.76 3.460 20.22 1 0 3 1

# Question 2

#Install ridge and glmnet packages  
#install.packages("ridge")  
#install.packages("glmnet")  
library(ridge)  
library(glmnet)

## Loading required package: Matrix

## Loaded glmnet 4.1

# Question 3

#Perform the exploratory data analysis  
  
#Preprocessing  
df = mtcars  
str(df)

## 'data.frame': 32 obs. of 11 variables:  
## $ mpg : num 21 21 22.8 21.4 18.7 18.1 14.3 24.4 22.8 19.2 ...  
## $ cyl : num 6 6 4 6 8 6 8 4 4 6 ...  
## $ disp: num 160 160 108 258 360 ...  
## $ hp : num 110 110 93 110 175 105 245 62 95 123 ...  
## $ drat: num 3.9 3.9 3.85 3.08 3.15 2.76 3.21 3.69 3.92 3.92 ...  
## $ wt : num 2.62 2.88 2.32 3.21 3.44 ...  
## $ qsec: num 16.5 17 18.6 19.4 17 ...  
## $ vs : num 0 0 1 1 0 1 0 1 1 1 ...  
## $ am : num 1 1 1 0 0 0 0 0 0 0 ...  
## $ gear: num 4 4 4 3 3 3 3 4 4 4 ...  
## $ carb: num 4 4 1 1 2 1 4 2 2 4 ...

summary(df)

## mpg cyl disp hp   
## Min. :10.40 Min. :4.000 Min. : 71.1 Min. : 52.0   
## 1st Qu.:15.43 1st Qu.:4.000 1st Qu.:120.8 1st Qu.: 96.5   
## Median :19.20 Median :6.000 Median :196.3 Median :123.0   
## Mean :20.09 Mean :6.188 Mean :230.7 Mean :146.7   
## 3rd Qu.:22.80 3rd Qu.:8.000 3rd Qu.:326.0 3rd Qu.:180.0   
## Max. :33.90 Max. :8.000 Max. :472.0 Max. :335.0   
## drat wt qsec vs   
## Min. :2.760 Min. :1.513 Min. :14.50 Min. :0.0000   
## 1st Qu.:3.080 1st Qu.:2.581 1st Qu.:16.89 1st Qu.:0.0000   
## Median :3.695 Median :3.325 Median :17.71 Median :0.0000   
## Mean :3.597 Mean :3.217 Mean :17.85 Mean :0.4375   
## 3rd Qu.:3.920 3rd Qu.:3.610 3rd Qu.:18.90 3rd Qu.:1.0000   
## Max. :4.930 Max. :5.424 Max. :22.90 Max. :1.0000   
## am gear carb   
## Min. :0.0000 Min. :3.000 Min. :1.000   
## 1st Qu.:0.0000 1st Qu.:3.000 1st Qu.:2.000   
## Median :0.0000 Median :4.000 Median :2.000   
## Mean :0.4062 Mean :3.688 Mean :2.812   
## 3rd Qu.:1.0000 3rd Qu.:4.000 3rd Qu.:4.000   
## Max. :1.0000 Max. :5.000 Max. :8.000

#Checking for missing values.  
colSums(is.na(df))

## mpg cyl disp hp drat wt qsec vs am gear carb   
## 0 0 0 0 0 0 0 0 0 0 0

#Checking for Empty Values  
colSums(df=='')

## mpg cyl disp hp drat wt qsec vs am gear carb   
## 0 0 0 0 0 0 0 0 0 0 0

#Checking for Duplicate values  
library(tidyverse)

## -- Attaching packages ---------------------------------------------------------------------------- tidyverse 1.3.0 --

## v ggplot2 3.3.2 v purrr 0.3.4  
## v tibble 3.0.3 v dplyr 1.0.0  
## v tidyr 1.1.0 v stringr 1.4.0  
## v readr 1.4.0 v forcats 0.5.0

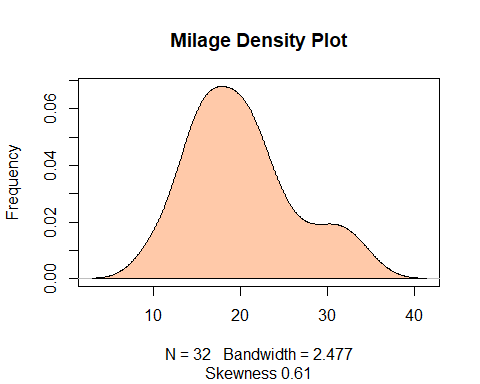
## -- Conflicts ------------------------------------------------------------------------------- tidyverse\_conflicts() --  
## x tidyr::expand() masks Matrix::expand()  
## x dplyr::filter() masks stats::filter()  
## x dplyr::lag() masks stats::lag()  
## x tidyr::pack() masks Matrix::pack()  
## x tidyr::unpack() masks Matrix::unpack()

duplicated(df)

## [1] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [13] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [25] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE

## The dataset is clean, there are no missing, null or duplicate values.

#Exploratory Data Analysis  
  
#Checking Normality of Response Variable  
  
library(e1071)  
plot(density(df$mpg), main = "Milage Density Plot", ylab="Frequency", sub=paste("Skewness",round(e1071::skewness(df$mpg),2)))  
polygon(density(df$mpg), col='#FFC9A9')

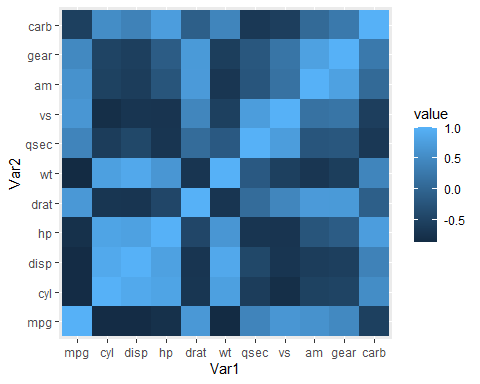


#Slightly Right Skweked, which implies most of the values are posititve in nature.   
  
#Correlation Heat Map  
library(ggplot2)  
library(reshape2)

##   
## Attaching package: 'reshape2'

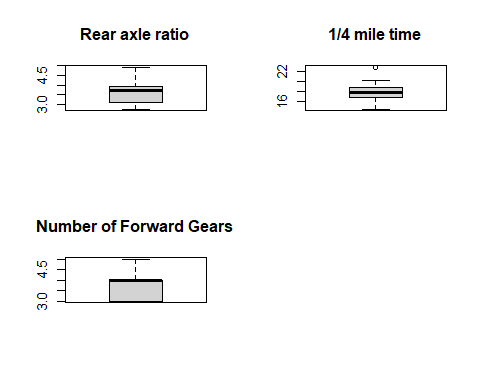
## The following object is masked from 'package:tidyr':  
##   
## smiths

cormat <- round(cor(df),2)  
melted\_cormat <- melt(cormat)  
ggplot(data = melted\_cormat, aes(x=Var1, y=Var2, fill=value)) +   
 geom\_tile()



#It is evident that most of the variables possess a high correlation with each other, thus we can assume multicollinearity is present.

#Checking for Outliers in highly positive correlated values with Milage  
par(mfrow=c(2,2))  
boxplot(df$drat, main = "Rear axle ratio")  
boxplot(df$qsec, main = "1/4 mile time")  
boxplot(df$gear, main = "Number of Forward Gears")

 #Computing a regular Model

#Building initial model  
X = model.matrix(mpg~. , mtcars)[,-1]  
Y = mtcars$mpg  
  
#Splitting the data  
set.seed(57)  
  
trainingRow <- sample(1:nrow(df), 0.7\*nrow(df))  
trainset <- df[trainingRow,]  
testset <- df[-trainingRow,]  
  
lrm <- lm(trainset$mpg~.,data=trainset)  
  
summary(lrm)

##   
## Call:  
## lm(formula = trainset$mpg ~ ., data = trainset)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -3.1759 -1.4218 -0.7548 1.0168 4.3028   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)  
## (Intercept) 8.55175 25.48468 0.336 0.744  
## cyl -0.07409 1.35897 -0.055 0.957  
## disp 0.01402 0.02943 0.476 0.643  
## hp -0.04564 0.03718 -1.227 0.245  
## drat 1.01463 2.83182 0.358 0.727  
## wt -3.66520 2.79708 -1.310 0.217  
## qsec 1.22513 0.93256 1.314 0.216  
## vs -1.03224 2.89376 -0.357 0.728  
## am 4.89791 2.80389 1.747 0.108  
## gear -0.76347 2.07162 -0.369 0.719  
## carb 1.00937 1.36147 0.741 0.474  
##   
## Residual standard error: 3.006 on 11 degrees of freedom  
## Multiple R-squared: 0.8924, Adjusted R-squared: 0.7945   
## F-statistic: 9.12 on 10 and 11 DF, p-value: 0.0005314

library(car)

## Loading required package: carData

##   
## Attaching package: 'car'

## The following object is masked from 'package:dplyr':  
##   
## recode

## The following object is masked from 'package:purrr':  
##   
## some

vif(lrm)

## cyl disp hp drat wt qsec vs am   
## 14.567223 35.146018 19.377286 5.983737 20.942264 7.286081 4.928560 4.627183   
## gear carb   
## 4.922279 12.942380

#All the values are above 5, there is strong multicollinearity present.  
  
MLR\_pred <- predict(lrm,testset)  
compare <- cbind(actual=testset$mpg,MLR\_pred)  
compare

## actual MLR\_pred  
## Mazda RX4 21.0 25.73138  
## Mazda RX4 Wag 21.0 25.48283  
## Hornet Sportabout 18.7 16.18709  
## Merc 450SE 16.4 13.86304  
## Merc 450SL 17.3 15.35424  
## Dodge Challenger 15.5 15.86656  
## Camaro Z28 13.3 12.02072  
## Pontiac Firebird 19.2 15.22923  
## Lotus Europa 30.4 25.48280  
## Ferrari Dino 19.7 21.79969

mean (apply(compare, 1, min)/apply(compare, 1, max))

## [1] 0.8654496

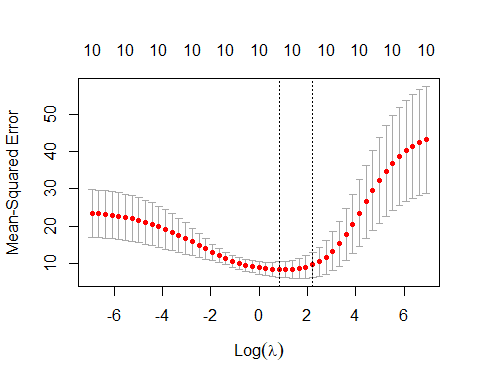
RMSE = sqrt(mean((testset$mpg-MLR\_pred)^2))  
RMSE# calculate accuracy

## [1] 3.242592

#Accuracy is only 81%, which is not verry efficient.

# Question 4 & 5

#Choose optimum lamba value and Extract the model using k-cross validation.  
  
#Creating a sequence with an interval of -0.12  
lambda\_seq = 10^seq(3, -3, by = -.12)  
  
ridge\_model1 = cv.glmnet(X[trainingRow,], Y[trainingRow],alpha = 0, type.measure = "mse", lambda = lambda\_seq, nfolds = 5)  
  
plot(ridge\_model1)



#The minima of the graph os the optimum lambda value.  
  
best\_lam = ridge\_model1$lambda.min  
best\_lam

## [1] 2.290868

# Question 6

#Build the final model and interpret  
  
#Fitting a Regression model with optimum lambda value.  
linRidgeMod = linearRidge(trainset$mpg ~ ., data = trainset)  
predicted = predict(linRidgeMod, testset) # predict on test data  
compare1 = cbind (actual=testset$mpg, predicted)  
mean (apply(compare1, 1, min)/apply(compare1, 1, max))

## [1] 0.9029484

summary(linRidgeMod)

##   
## Call:  
## linearRidge(formula = trainset$mpg ~ ., data = trainset)  
##   
##   
## Coefficients:  
## Estimate Scaled estimate Std. Error (scaled) t value (scaled)  
## (Intercept) 14.742140 NA NA NA  
## cyl -0.326153 -2.753488 3.321952 0.829  
## disp -0.005767 -3.492460 2.841631 1.229  
## hp -0.015200 -5.409057 3.161192 1.711  
## drat 1.757967 4.564728 3.400698 1.342  
## wt -1.322165 -6.502449 3.091123 2.104  
## qsec 0.468642 4.077506 3.255874 1.252  
## vs 0.103222 0.238042 3.398804 0.070  
## am 2.909703 6.710119 3.236733 2.073  
## gear -0.086758 -0.279297 3.248383 0.086  
## carb -0.277200 -2.201791 3.031816 0.726  
## Pr(>|t|)   
## (Intercept) NA   
## cyl 0.4072   
## disp 0.2191   
## hp 0.0871 .  
## drat 0.1795   
## wt 0.0354 \*  
## qsec 0.2104   
## vs 0.9442   
## am 0.0382 \*  
## gear 0.9315   
## carb 0.4677   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Ridge parameter: 0.1286921, chosen automatically, computed using 3 PCs  
##   
## Degrees of freedom: model 5.755 , variance 4.066 , residual 7.443

#Creating another model with only signifiant values.  
  
linRidgeMod = linearRidge(trainset$mpg ~ ., data = trainset[, c(6,10,11)])  
predicted1 = predict(linRidgeMod, testset) # predict on test data  
compare2 = cbind (actual=testset$mpg, predicted1)  
mean (apply(compare2, 1, min)/apply(compare2, 1, max))

## [1] 0.9464945

summary(linRidgeMod)

##   
## Call:  
## linearRidge(formula = trainset$mpg ~ ., data = trainset[, c(6,   
## 10, 11)])  
##   
##   
## Coefficients:  
## Estimate Scaled estimate Std. Error (scaled) t value (scaled)  
## (Intercept) 27.642 NA NA NA  
## wt -3.316 -16.308 4.659 3.500  
## gear 1.924 6.193 4.089 1.514  
## carb -1.395 -11.082 4.262 2.600  
## Pr(>|t|)   
## (Intercept) NA   
## wt 0.000465 \*\*\*  
## gear 0.129929   
## carb 0.009319 \*\*   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Ridge parameter: 0.04208334, chosen automatically, computed using 2 PCs  
##   
## Degrees of freedom: model 2.698 , variance 2.457 , residual 2.939

#The accuracy has increased from 75% to 88%.  
  
RMSE = sqrt(mean((testset$mpg-predicted1)^2))  
RMSE

## [1] 1.389815

#RMSE has decreased too.