MATH 6350: Statistical Learning and Data Mining
Homework 1

Contributions of Co-Authors

Stephanie Dinh (sdinh@central.uh.edu): Computed the results and provided interpretation for question #1-5.

Ying-Yu Huang (yingyu010365@gmail.com): Computed the results and provided interpretation for question #6-10.

Patricia Sieng (patricia.sieng@yahoo.com): Computed the results and provided interpretation for question #11-15.

Part 1:Results

*Preliminary treatment of the dataset

Number of N cases = 392 rows kept

1) Mean and standard deviation of each feature

Notation

$\underline{x}$= Sample population mean of x

σ_x = Standard deviation of x

$\underline{cyl}$ = 5.471939

σ_{cyl} = 1.705783

$\underline{dis}$ = 194.412

σ_{dis} = 104.644

$\underline{hor}$ = 104.4694

σ_{hor} = 38.49116

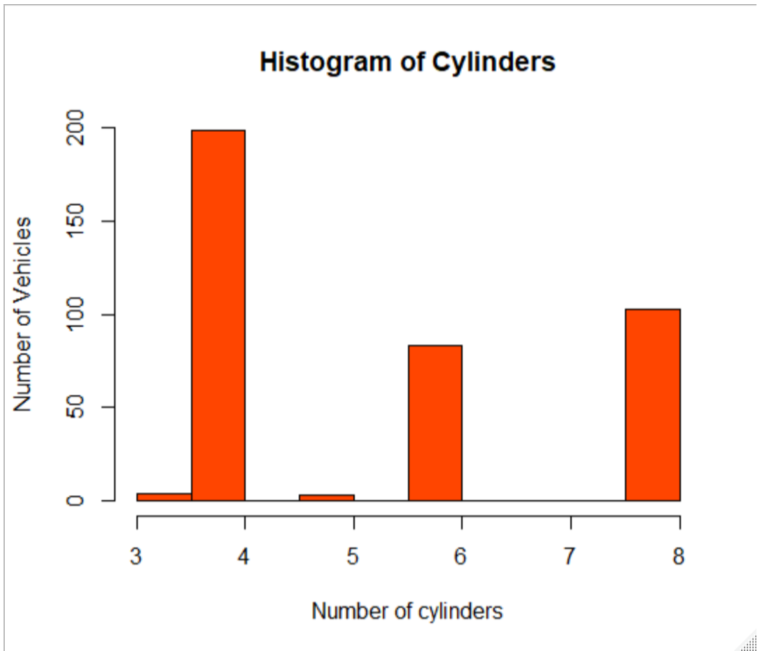
$\underline{wei}$ = 2977.584

σ_{wei} = 849.4026

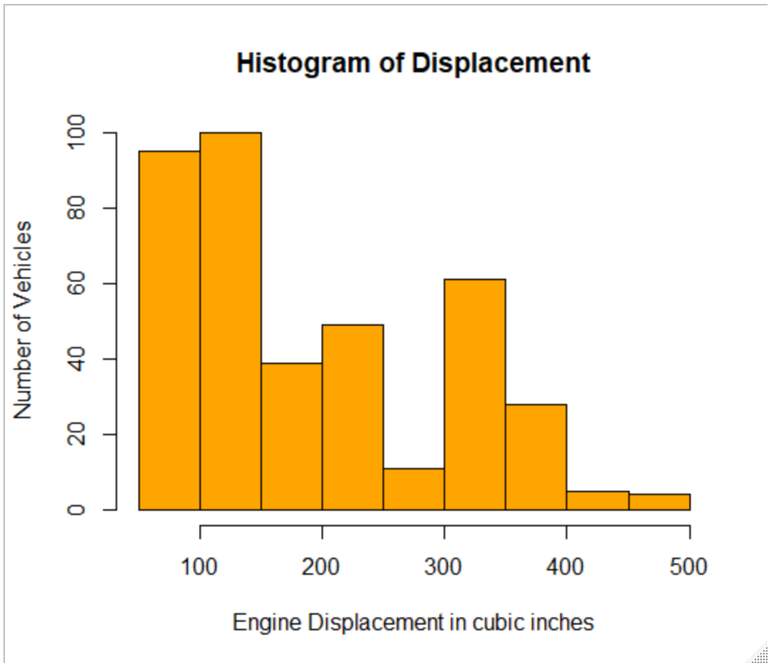
$\underline{acc}$ = 15.54133

σ_{acc} = 2.758864

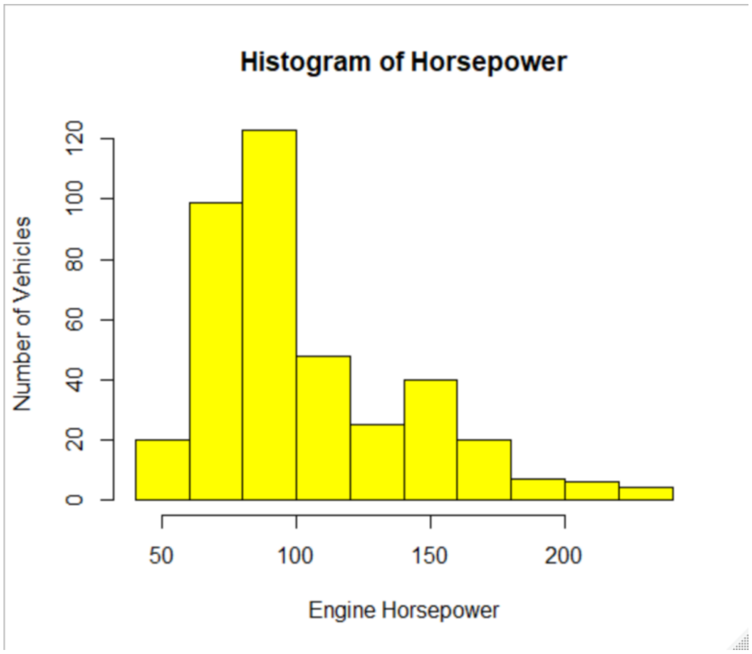
2) Histograms



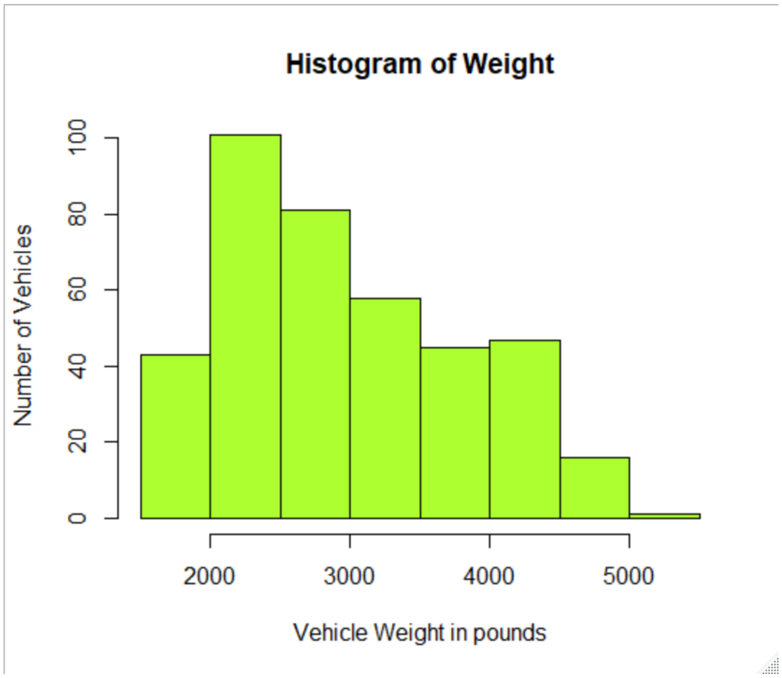
The frequency of vehicles with cylinders between 4 to 8. The histogram appears to indicate that most vehicles have 4 cylinders in their engine.



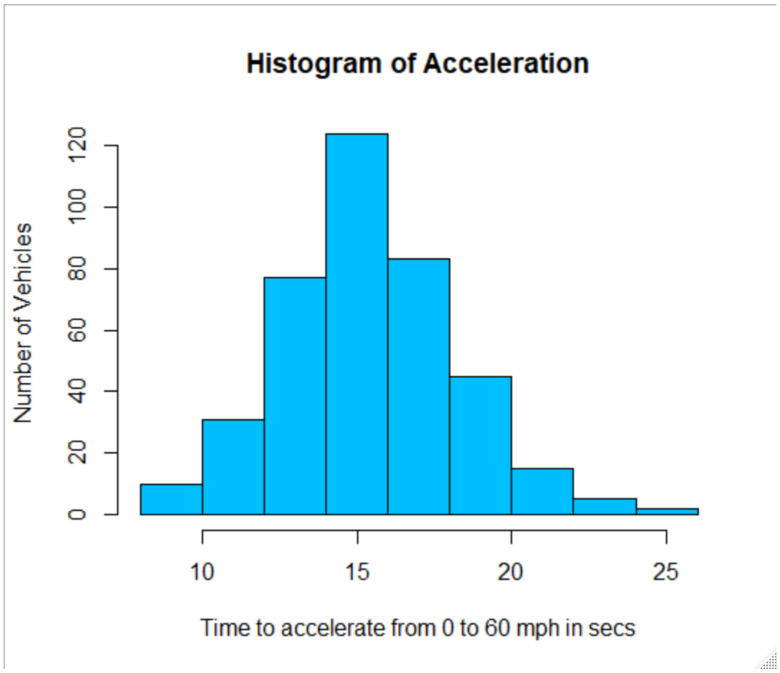
The frequency of vehicle engine displacement in in^3. The histogram appears to be roughly right-skewed with a few outliers, indicating that most vehicles have an engine displacement less than 250 in^3.



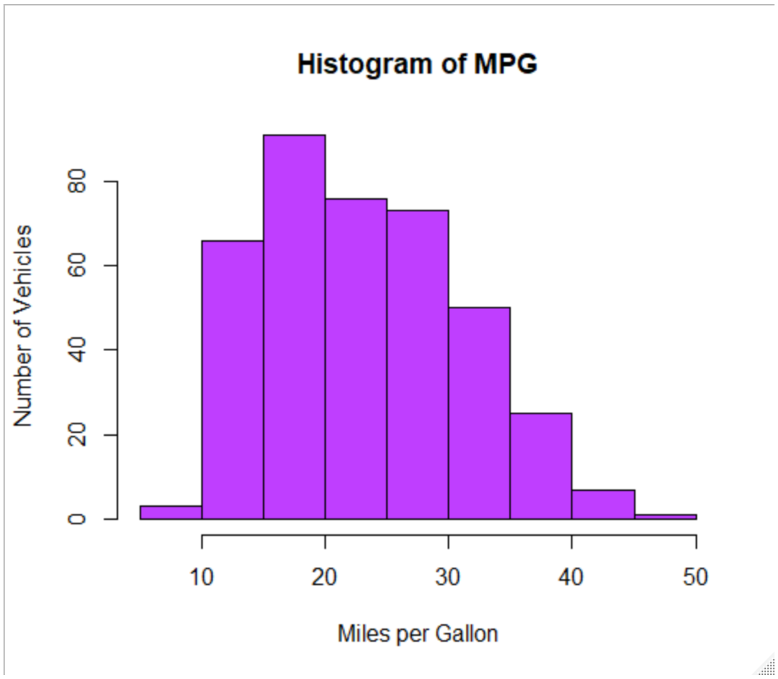
The frequency of vehicle engine horsepower. The histogram appears to be right-skewed, indicating that most vehicles have an engine horsepower less than 100.



The frequency of vehicle weight in lbs. The histogram is fairly uniform in distribution, yet unsymmetric in shape with most values concentrated to the right. This indicates that most vehicles weigh less than 3,500 pounds.

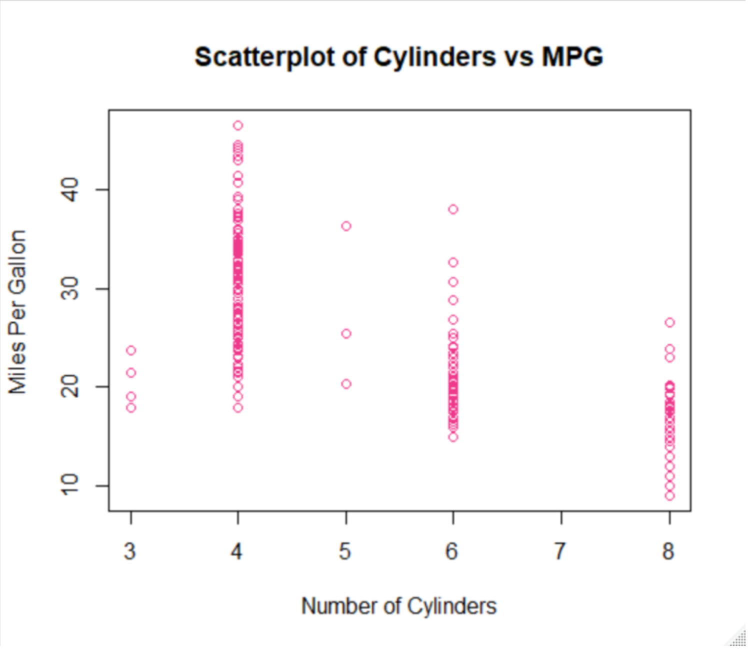


The frequency of vehicle acceleration time from 0 to 60 mph in seconds. The histogram appears to be fairly symmetric and unimodal in distribution, indicating most vehicles take 15 seconds to accelerate from 0 to 60 mph.

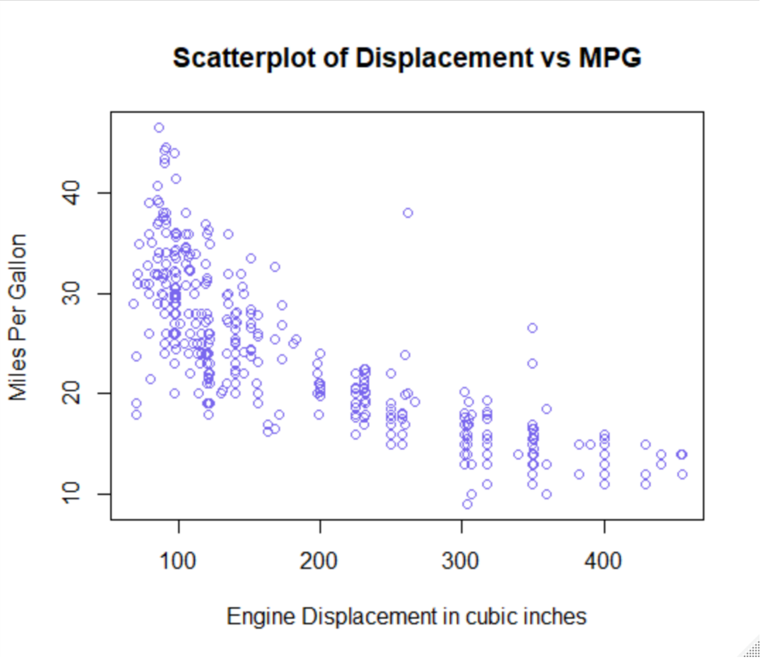


The frequency of vehicle mileage per gallon. The histogram appears to be symmetric and uniform in distribution with few outliers. The mpg of vehicles range from 10 to 40 miles per hour.

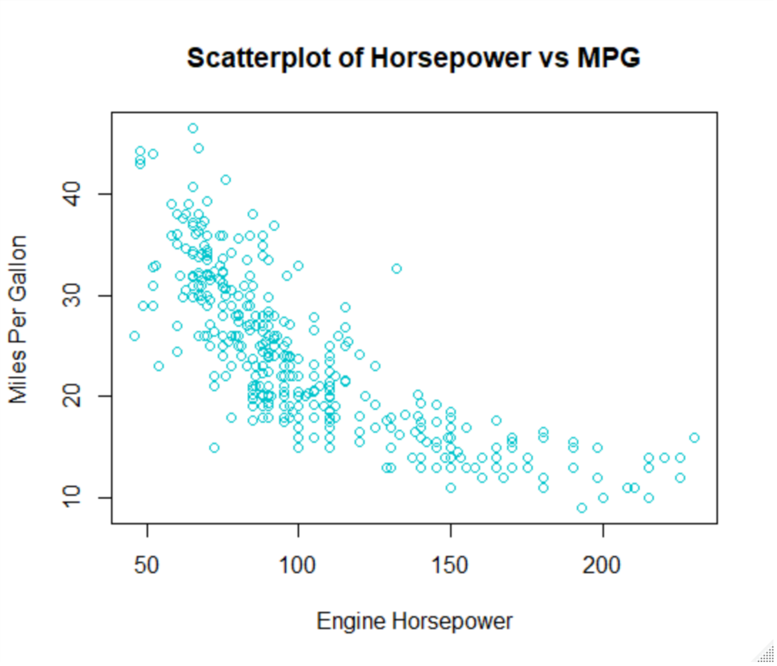
3,4) Scatterplots



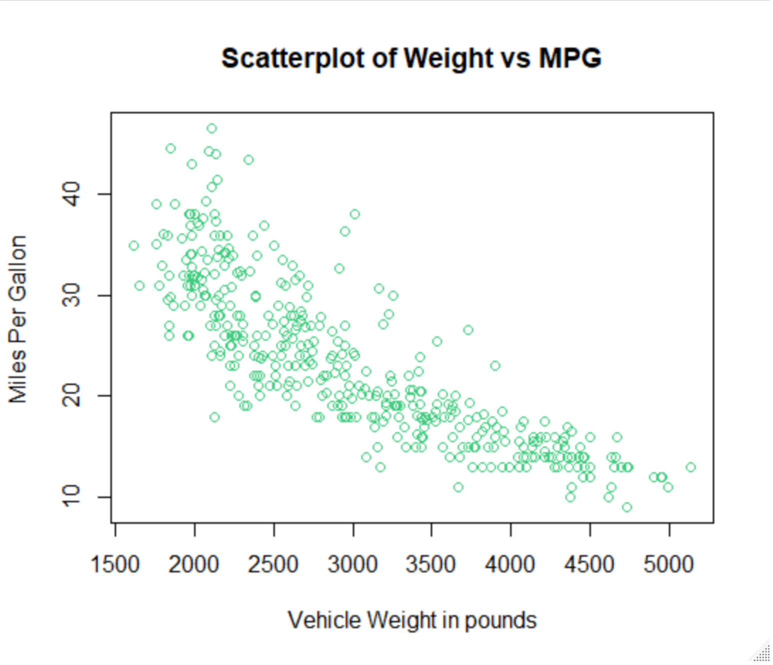
There is a strong, negative relationship between number of cylinders and mpg. Cylinders could be used to predict mpg.



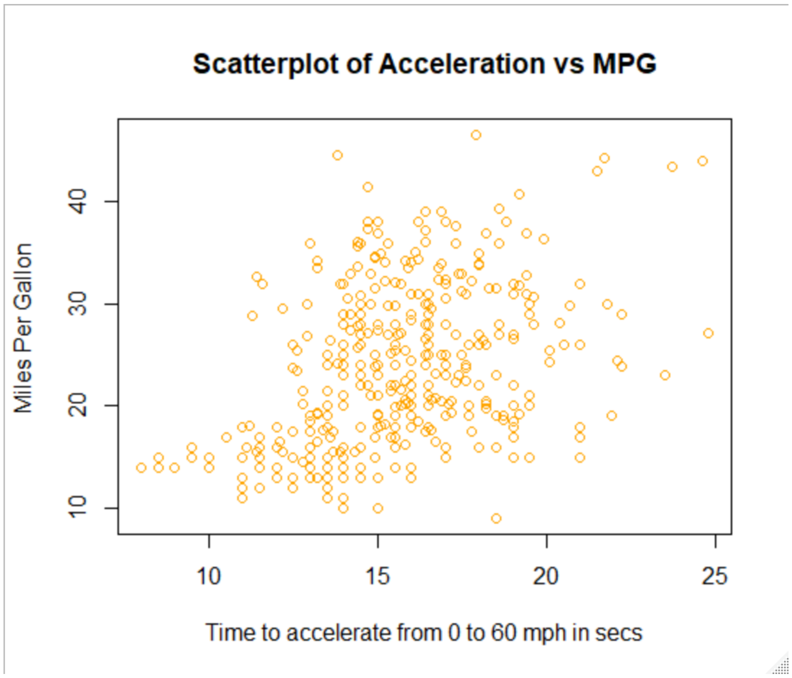
There is a strong, negative relationship between engine displacement and mpg. This indicates that displacement could be used to predict mpg.



There is a strong, negative relationship between engine horsepower and mpg. This indicates that horsepower could be used to predict mpg.



There is a strong, negative relationship between vehicle weight and mpg. This indicates that weight could be used to predict mpg.



There is a weak, positive relationship between acceleration and mpg. This indicates that acceleration is a weak predictor to predict mpg.

5) Correlations

Notation

corr(x, y) = The correlation of x and y

corr(cyl, mpg) = -0.7776175

There is a strong, negative relationship between number of cylinders and mpg. When the number of cylinders increases, mpg will decrease.Cylinders could be used to predict mpg.

corr(dis, mpg) = -0.8051269

There is a strong, negative relationship between engine displacement and mpg. When engine displacement increases, mpg will decrease. Displacement could be used to predict mpg.

corr(hor, mpg) = -0.7784268

There is a strong, negative relationship between engine horsepower and mpg. When engine horsepower increases, mpg will decrease. Horsepower could be used to predict mpg.

corr(wei, mpg) = -0.8322442

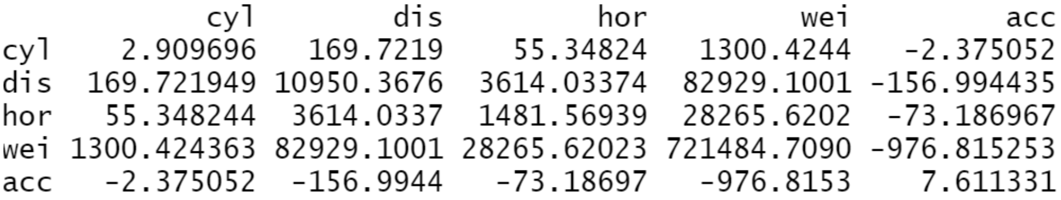
There is a strong, negative relationship between vehicle weight and mpg. When vehicle weight increases, mpg will decrease. Weight could be used to predict mpg.

corr(acc, mpg) = 0.4233285

There is a weak, positive relationship between acceleration and mpg. Acceleration is a weak predictor for mpg.

6) Covariance and correlation matrices

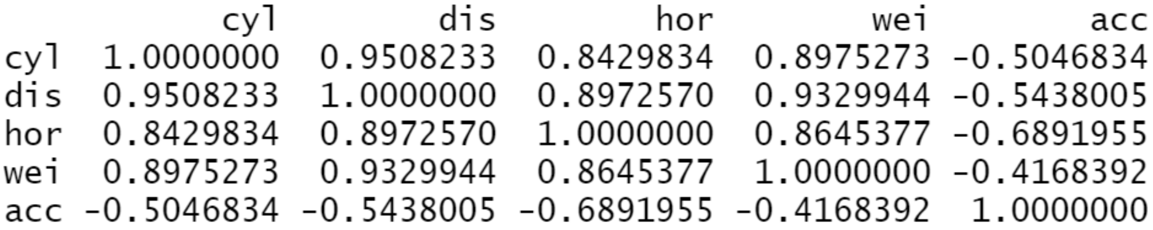
Covariance matrix



```
            cyl        dis         hor         wei        acc
cyl    2.909696   169.7219    55.34824   1300.4244  -2.375052
dis  169.721949 10950.3676  3614.03374  82929.1001 -156.994435
hor   55.348244  3614.0337  1481.56939  28265.6202  -73.186967
wei 1300.424363 82929.1001 28265.62023 721484.7090 -976.815253
acc   -2.375052  -156.9944   -73.18697   -976.8153   7.611331
```

In the above matrix, we see that the dimension of the covariance matrix is 5 × 5. This is basically a symmetric matrix i.e. a square matrix that is equal to its transpose matrix. The covariance of the j-th variable with the k-th variable is equivalent to the covariance of the k-th variable with the j-th variable.

Correlation matrix



```
           cyl        dis        hor        wei        acc
cyl  1.0000000  0.9508233  0.8429834  0.8975273 -0.5046834
dis  0.9508233  1.0000000  0.8972570  0.9329944 -0.5438005
hor  0.8429834  0.8972570  1.0000000  0.8645377 -0.6891955
wei  0.8975273  0.9329944  0.8645377  1.0000000 -0.4168392
acc -0.5046834 -0.5438005 -0.6891955 -0.4168392  1.0000000
```

The line of 1.00 going from the top left to the bottom right is the main *diagonal*, which shows that each variable always perfectly correlates with itself. This matrix is symmetrical, with the same correlation shown above the main diagonal being a mirror image of those below the main diagonal.

The value of covariance is affected by the change in scale of the variables. However, the value of correlation is not influenced by the change in scale of the values. The correlation coefficients lie between -1 and +1, covariance can take any value between -∞ and +∞.

7) Prove eigenvalues L1>L2>L3>L4>L5 of correlation matrix

4.07185982 > 0.69386125 > 0.13349305 > 0.06426839 > 0.03651750
The above notation holds true.

8) Prove L1 + L2 + L3 + L4 + L5 = 5

4.07185982 + 0.69386125 + 0.13349305 + 0.06426839 + 0.03651750 = 5
The above notion holds true.

9,10) For i = 1, 2, 3, 4, 5; compute Ratios Ri = (L1 + L2 + ... + Li)/5

R1 = 0.814372
The first principal component accounts for 81% of the total variance. From the eigenvector, we can see that the number of cylinders, displacement, horsepower and weight are all similarly negatively correlated with component 1.

R2 = 0.9531442
The first and second principal components accounts for 95% of the total variance. With this, we can say that a majority of the data can be explained in 2 dimensions.

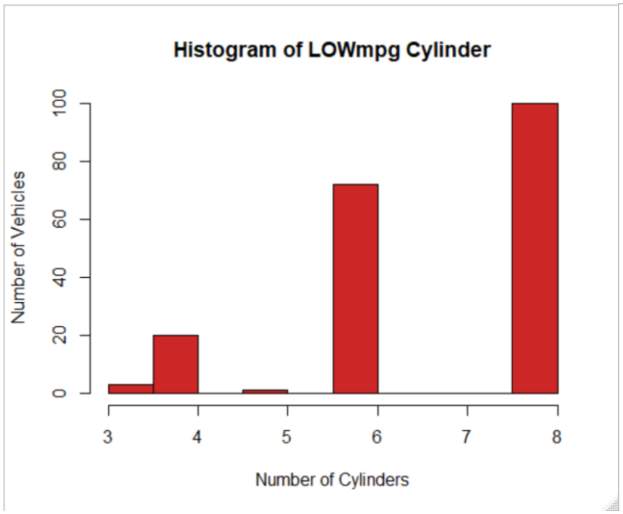
R3 = 0.9798428
The first, second, and third principal components accounts for 97% of the total variance.

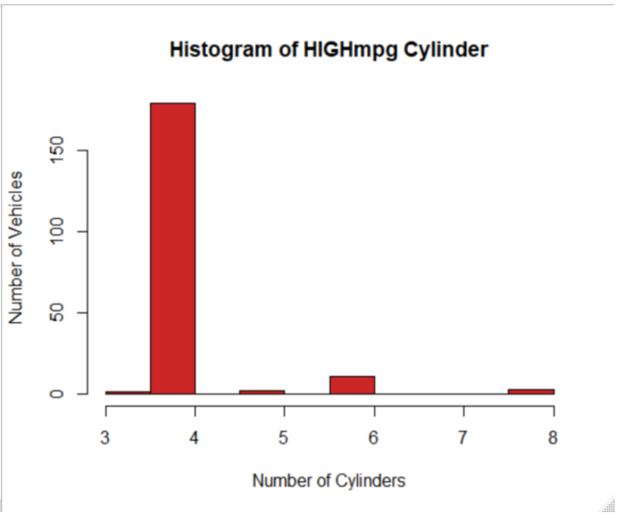
R4 = 0.9926965
99% of the total variance can be explained through the first four principal components. With the exception of 1% estimation, the information can be represented in four dimensions.

R5 = 1
R5 accounts for the total variance of the cloud of data in R^5.

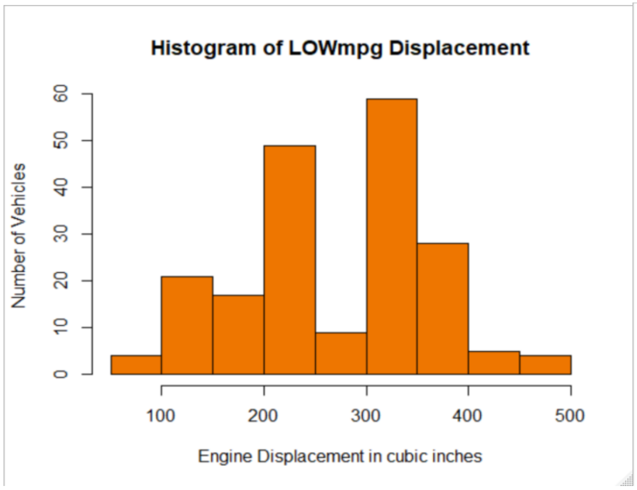
11) Reordering the rows from ascending order and creating LOWmpg and HIGHmpg table
Refer to its respective code in Part 2.

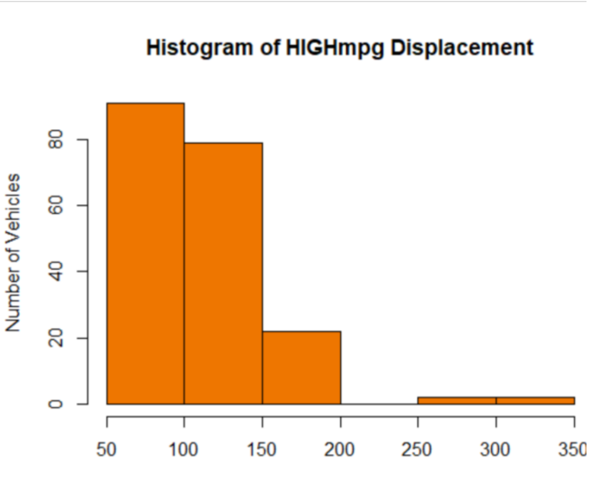
12,13) Side by side histograms for features of LOWmpg and HIGHmpg



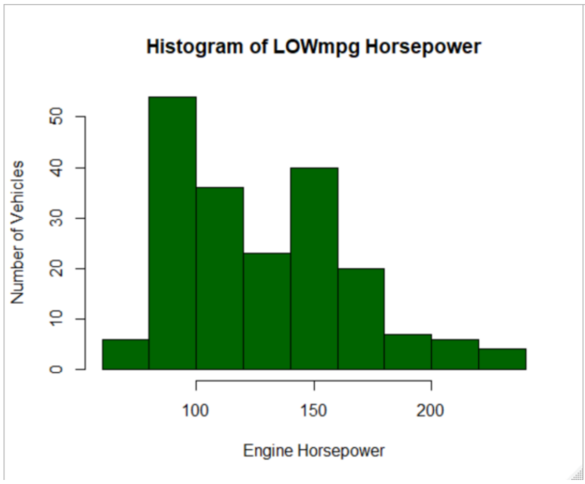


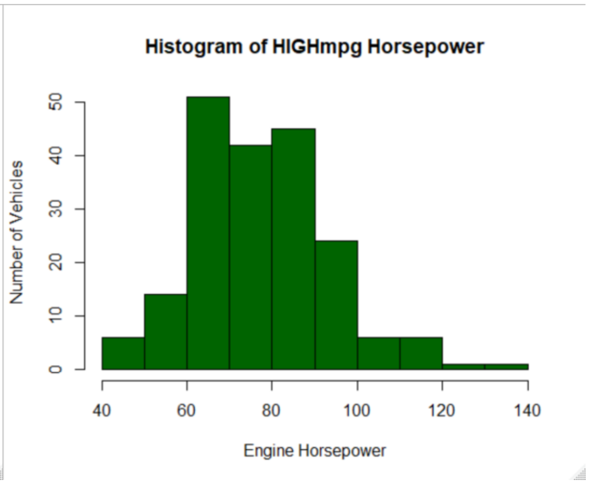
The histograms are both skewed in the opposite direction. The histogram for LOWmpg cylinder is left-skewed while the histogram for HIGHmpg cylinder is right-skewed. Cylinder could be used to discriminate between low mpg and high mpg.



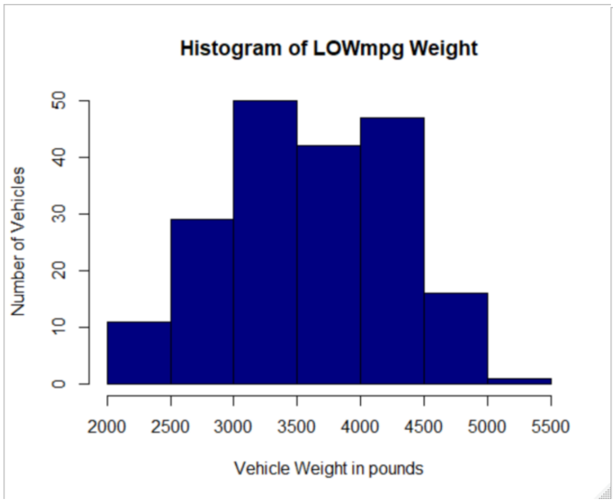


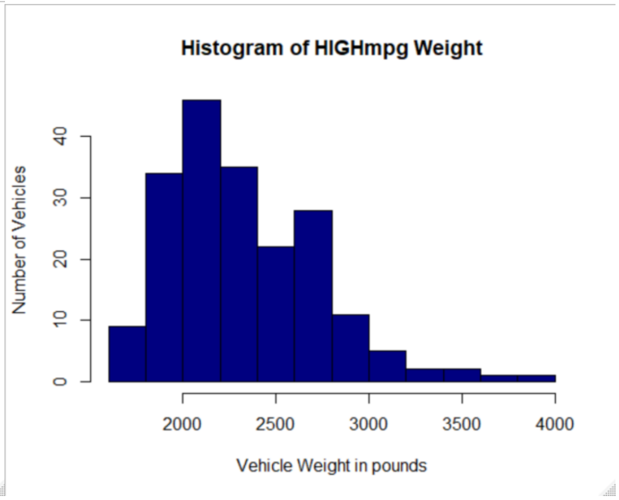
The histogram for LOWmpg displacement appears to be symmetric and bimodal in distribution while the histogram for HIGHmpg displacement appears to be right-skewed. Displacement could be used to discriminate between low and high mpg.



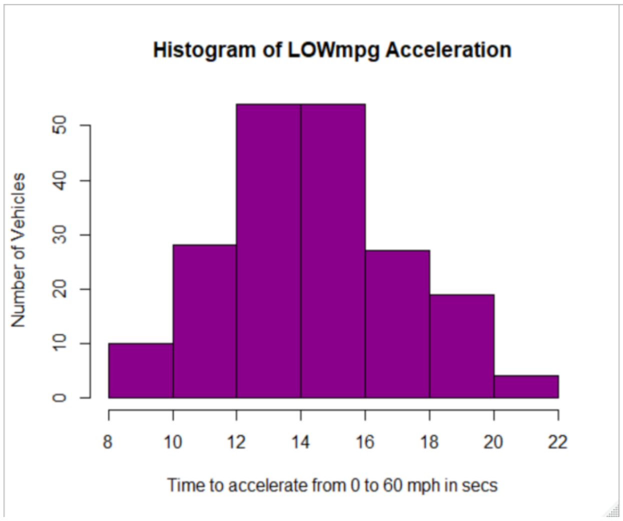


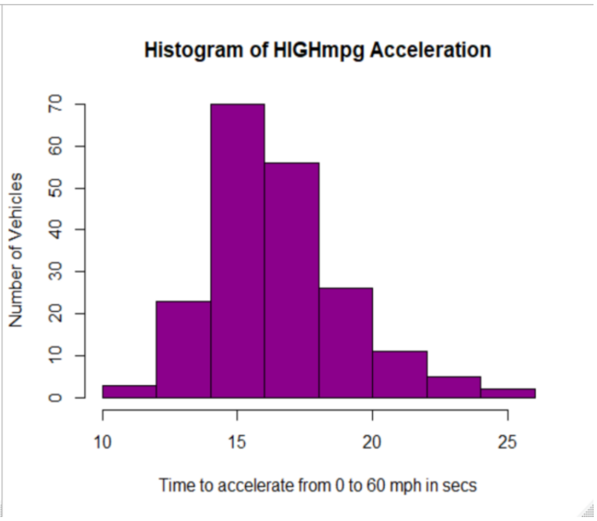
The histogram for LOWmpg horsepower appears to be right-skewed while the histogram for HIGHmpg horsepower appears to be symmetric and uniform with a few outliers. Horsepower could be used to discriminate between low and high mpg.





The histogram for LOWmpg weight appears to be symmetric and uniform while the histogram for HIGHmpg weight is right-skewed. Weight could be used to discriminate between high and low mpg.





Both histograms appear to be roughly symmetric and unimodal in distribution. Acceleration would not be a good feature to discriminate between high and low mpg.

14) Mean and standard deviations of the F values corresponding to LOWmpg and HIGHmpg

Notation

$\underline{x}$= Sample population mean of x

σ_x = Standard deviation of x

Mean and standard deviations of LOWmpg features

$\underline{LOWcyl}$ = 6.765306

σ_{LOWcyl} = 1.420011

$\underline{LOWdis}$ = 273.1582

σ_{LOWdis} = 89.52399

$\underline{LOWhor}$ = 130.1122

σ_{LOWhor} = 37.35564

$\underline{LOWwei}$ = 3620.403

σ_{LOWwei} = 676.9322

$\underline{LOWacc}$ = 14.58571

σ_{LOWacc} = 2.685154

Mean and standard deviations of HIGHmpg features

$\underline{HIGHcyl}$= 4.178571

σ_{HIGHcyl} = 0.6746319

$\underline{HIGHdis}$= 115.6658

σ_{HIGHdis} = 38.42951

$\underline{HIGHhor}$ = 78.82653

σ_{HIGHhor} = 15.91969

$\underline{HIGHwei}$ = 2334.765

σ_{HIGHwei} = 397.1924

$\underline{HIGHacc}$ = 16.49694

σ_{HIGHacc} = 2.493168

15) discr(F) = |mhigh(F) - mlow(F)| / s(F) where s(F) = (stdlow(f) + stdhigh(F)) / sqrt(N)

Notation
discr(F) = Discriminatory value of feature F

Discriminatory values of the features
discr(cyl) = 24.45034
discr(dis) = 24.36971
discr(hor) = 19.05958
discr(wei) = 23.69774
discr(acc) = 7.307447

Cylinder, displacement, horsepower, and weight all have a high capacity to discriminate between low and high mpg. As in our previous results, acceleration is poor predictor to predict mpg. Thus, it is not a surprise that the above results show that acceleration does not have a high capacity to discriminate between low and high mpg.

15) Alternative approach to computing discriminatory values of feature F using t-tests

Notation
pval(F) = The p-value of feature F
discr(F) = Discriminatory value of feature F

The p-value of the features
pval(cyl) < 2.2e-16
pval(dis) < 2.2e-16
pval(hor) < 2.2e-16
pval(wei) < 2.2e-16
pval(acc) = 1.617e-12
All p-values are near 0.

Discriminatory values of the features
discr(cyl) = 1-pval(cyl)

discr(dis) = 1-pval(dis)
discr(hor) = 1-pval(hor)
discr(wei) = 1-pval(wei)
discr(acc) = 1-pval(acc)
All discriminatory values are a little less than 1.

Prove 0 < discr(F) < 1
0 < discr(cyl) < 1
0 < discr(dis) < 1
0 < discr(hor) < 1
0 < discr(wei) < 1
0 < discr(acc) < 1
The above notion holds true for all.

All the p-values computed from the t-tests were near 0, thus resulting in high discriminatory values for all features. This indicates that all features have a strong capacity to help discriminate between low and high mpg and that there is a very strong indication that low and high mpg are significantly distinct.

Part 2: Code

```
# *Preliminary treatment of the data set
start_time <- Sys.time()
# Omit last three columns; Initially 397 rows
auto <- Auto[,c(1:6)]
# Horsepower designated as a factor variable
# Some rows contain non-numeric value (i.e. "?")
# Omit non-numeric rows; Number of N cases kept = 392
auto <- auto[!auto$horsepower == "?"]
# Convert horsepower to numeric variable
auto$horsepower <- as.numeric(as.character(auto$horsepower))
end_time <- Sys.time()
end_time - start_time # Computation time: 0.03091598 secs

# Rename the variable names of the dataset for convenience
install.packages(data.table)
library(data.table)
setnames(auto, old=c("cylinders", "displacement", "horsepower", "weight", "acceleration"),
new=c("cyl", "dis", "hor", "wei", "acc"))
```

```
# 1) Compute the mean and standard deviation for features
attach(auto)
start_time <- Sys.time()
mean(cyl)
sd(cyl)
mean(dis)
sd(dis)
mean(hor)
sd(hor)
mean(wei)
sd(wei)
mean(acc)
sd(acc)
end_time <- Sys.time()
end_time - start_time # Computation time: 0.02801418 secs

# 2) Generate histograms for features and mpg
start_time <- Sys.time()
hist(cyl,
    main="Histogram of Cylinders",
    xlab="Number of cylinders",
    ylab="Number of Vehicles",
    col="orangered")
hist(dis,
    main="Histogram of Displacement",
    xlab="Engine Displacement in cubic inches",
    ylab="Number of Vehicles",
    col="orange")
hist(hor,
    main="Histogram of Horsepower",
    xlab="Engine Horsepower",
    ylab="Number of Vehicles",
    col="yellow1")
hist(wei,
    main="Histogram of Weight",
    xlab="Vehicle Weight in pounds",
    ylab="Number of Vehicles",
    col="greenyellow")
hist(acc,
    main="Histogram of Acceleration",
    xlab="Time to accelerate from 0 to 60 mph in secs",
    ylab="Number of Vehicles",
    col="deepskyblue")
hist(mpg,
    main="Histogram of MPG",
    xlab="Miles per Gallon",
    ylab="Number of Vehicles",
    col="darkorchid1")
end_time <- Sys.time()
end_time - start_time # Computation time: 0.2850239 secs

# 3,4) Generate scatterplots for features vs mpg
start_time <- Sys.time()
plot(cyl, mpg, main="Scatterplot of Cylinders vs MPG",
    xlab="Number of Cylinders", ylab="Miles Per Gallon ", col="violetred2")
plot(dis, mpg, main="Scatterplot of Displacement vs MPG",
    xlab="Engine Displacement in cubic inches", ylab="Miles Per Gallon ", col="slateblue2")
plot(hor, mpg, main="Scatterplot of Horsepower vs MPG",
    xlab="Engine Horsepower", ylab="Miles Per Gallon ", col="turquoise3")
plot(wei, mpg, main="Scatterplot of Weight vs MPG",
    xlab="Vehicle Weight in pounds", ylab="Miles Per Gallon ", col="seagreen3")
plot(acc, mpg, main="Scatterplot of Acceleration vs MPG",
    xlab="Time to accelerate from 0 to 60 mph in secs", ylab="Miles Per Gallon ",
col="orange1")
end_time <- Sys.time()
end_time - start_time # Computation time: 0.242872 secs

# 5) Compute the correlations between features and mpg
start_time <- Sys.time()
cor(cyl, mpg)
cor(dis, mpg)
cor(hor, mpg)
cor(wei, mpg)
cor(acc, mpg)
end_time <- Sys.time()
end_time - start_time # Computation time: 0.02312088 secs

# 6) Compute the covariance and correlation matrices for the features
auto2 <- auto[,c(2:6)] # Created new data set that excludes mpg
start_time <- Sys.time()
```

```
cov(auto2)
cor(auto2)
end_time <- Sys.time()
end_time - start_time Computation time: 0.02203393 secs

# 7) Prove eigenvalues L1>L2>L3>L4>L5 of correlation matrix
eigen(cor(auto2))
L1 <- 4.07185982
L2 <- 0.69386125
L3 <- 0.13349305
L4 <- 0.06426839
L5 <- 0.03651750
start_time <- Sys.time()
L1>(L2>(L3>(L4>L5))) # TRUE
end_time <- Sys.time()
end_time - start_time # Computation time: 0.01670694 secs

# 8) Prove  L1 + L2 + L3 + L4 + L5 = 5
start_time <- Sys.time()
eigen_sum <- L1 + L2 + L3 + L4 + L5
eigen_sum # Eigenvalue sum is equal to 5
end_time <- Sys.time()
end_time - start_time # Computation time: 0.02139306 secs

# 9,10) For i = 1, 2, 3, 4, 5; compute Ratios Ri = (L1 + L2 + ... + Li)/5
start_time <- Sys.time()
R1 <- L1/5
R1
R2 <- (L1 + L2)/5
R2
R3 <- (L1 + L2 + L3)/5
R3
R4 <- (L1 + L2 + L3 + L4)/5
R4
R5 <- (L1 + L2 + L3 + L4 + L5)/5
R5
end_time <- Sys.time()
end_time - start_time # Computation time: 0.0459311 secs

# 11) Reorder the rows of the dataset in ascending order
```

```
start_time <- Sys.time()
newAuto <- auto[order(mpg),]
median_mpg <- median(newAuto$mpg) # Calculate the median of mpg
# Create LOWmpg and HIGHmpg table
attach(newAuto)
LOWmpg <- newAuto[mpg < median_mpg,] # Table where mpg is less than mpg_median
HIGHmpg <- newAuto[mpg > median_mpg,] # Table where mpg is greater than mpg_median
end_time <- Sys.time()
end_time - start_time # Computation time: 0.03594804 secs

# 12,13) Generate side by side histograms for LOWmpg and HIGHmpg features
start_time <- Sys.time()
hist(LOWmpg$cyl,
    main="Histogram of LOWmpg Cylinder",
    xlab="Number of Cylinders",
    ylab="Number of Vehicles",
    col="firebrick3")
hist(HIGHmpg$cyl,
    main="Histogram of HIGHmpg Cylinder",
    xlab="Number of Cylinders",
    ylab="Number of Vehicles",
    col="firebrick3")
hist(LOWmpg$dis,
    main="Histogram of LOWmpg Displacement",
    xlab="Engine Displacement in cubic inches",
    ylab="Number of Vehicles",
    col="darkorange2")
hist(HIGHmpg$dis,
    main="Histogram of HIGHmpg Displacement",
    xlab="Engine Displacement in cubic inches",
    ylab="Number of Vehicles",
    col="darkorange2")
hist(LOWmpg$hor,
    main="Histogram of LOWmpg Horsepower",
    xlab="Engine Horsepower",
    ylab="Number of Vehicles",
    col="darkgreen")
hist(HIGHmpg$hor,
    main="Histogram of HIGHmpg Horsepower",
    xlab="Engine Horsepower",
    ylab="Number of Vehicles",
    col="darkgreen")
hist(LOWmpg$wei,
    main="Histogram of LOWmpg Weight",
    xlab="Vehicle Weight in pounds",
    ylab="Number of Vehicles",
    col="navy")
hist(HIGHmpg$wei,
    main="Histogram of HIGHmpg Weight",
    xlab="Vehicle Weight in pounds",
    ylab="Number of Vehicles",
    col="navy")
hist(LOWmpg$acc,
    main="Histogram of LOWmpg Acceleration",
    xlab="Time to accelerate from 0 to 60 mph in secs",
    ylab="Number of Vehicles",
    col="darkmagenta")
hist(HIGHmpg$acc,
    main="Histogram of HIGHmpg Acceleration",
    xlab="Time to accelerate from 0 to 60 mph in secs",
    ylab="Number of Vehicles",
    col="darkmagenta")
end_time <- Sys.time()
end_time - start_time # Computation time: 0.377131 secs

# 14) Compute the mean and standard deviation for LOWmpg and HIGHmpg features
start_time <- Sys.time()
mlow_cyl <- mean(LOWmpg$cyl)
mlow_cyl
stdlow_cyl <- sd(LOWmpg$cyl)
stdlow_cyl
mlow_dis <- mean(LOWmpg$dis)
mlow_dis
stdlow_dis <- sd(LOWmpg$dis)
stdlow_dis
mlow_hor <- mean(LOWmpg$hor)
mlow_hor
stdlow_hor <- sd(LOWmpg$hor)
stdlow_hor
mlow_wei <- mean(LOWmpg$wei)
```

```
mlow_wei
stdlow_wei <- sd(LOWmpg$wei)
stdlow_wei
mlow_acc <- mean(LOWmpg$acc)
mlow_acc
stdlow_acc <- sd(LOWmpg$acc)
stdlow_acc
mhigh_cyl <- mean(HIGHmpg$cyl)
mhigh_cyl
stdhigh_cyl <- sd(HIGHmpg$cyl)
stdhigh_cyl
mhigh_dis <- mean(HIGHmpg$dis)
mhigh_dis
stdhigh_dis <- sd(HIGHmpg$dis)
stdhigh_dis
mhigh_hor <- mean(HIGHmpg$hor)
mhigh_hor
stdhigh_hor <- sd(HIGHmpg$hor)
stdhigh_hor
mhigh_wei <- mean(HIGHmpg$wei)
mhigh_wei
stdhigh_wei <- sd(HIGHmpg$wei)
stdhigh_wei
mhigh_acc <- mean(HIGHmpg$acc)
mhigh_acc
stdhigh_acc <- sd(HIGHmpg$acc)
stdhigh_acc
end_time <- Sys.time()
end_time - start_time # Computation time: 0.07506585 secs

# 15) Compute discr(F) =  |mhigh(F) - mlow(F)| / s(F)
# where s(F) = (stdlow(f) + stdhigh(F)) / #sqrt(N)
# Function that computes the discriminatory values of the features
disValue <- function(mhigh, mlow, stdhigh, stdlow){
value <- abs(mhigh-mlow)/((stdlow+stdhigh)/sqrt(392))
return (value)}
start_time <- Sys.time()
discr_cyl <- disValue(mhigh_cyl, mlow_cyl, stdhigh_cyl, stdlow_cyl)
discr_cyl
discr_dis <- disValue(mhigh_dis, mlow_dis, stdhigh_dis, stdlow_dis)
```

```
discr_dis
discr_hor <- disValue(mhigh_hor, mlow_hor, stdhigh_hor, stdlow_hor)
discr_hor
discr_wei <- disValue(mhigh_wei, mlow_wei, stdhigh_wei, stdlow_wei)
discr_wei
discr_acc <- disValue(mhigh_acc, mlow_acc, stdhigh_acc, stdlow_acc)
discr_acc
end_time <- Sys.time()
end_time - start_time # Computation time: 0.03590107 secs

# 15) Alternative approach to computing discriminatory values of feature F using t-tests
# Vectors that list F values corresponding to the cases belonging to LOWmpg
start_time <- Sys.time()
x_lowcyl = LOWmpg$cyl
x_lowdis = LOWmpg$dis
x_lowhor = LOWmpg$hor
x_lowwei = LOWmpg$wei
x_lowacc = LOWmpg$acc
# Vectors that list F values corresponding to the cases belonging to HIGHmpg
x_highcyl = HIGHmpg$cyl
x_highdis = HIGHmpg$dis
x_highhor = HIGHmpg$hor
x_highwei = HIGHmpg$wei
x_highacc = HIGHmpg$acc
# Perform t-tests on F values
t.test(x_lowcyl, y_highcyl)
t.test(x_lowdis, y_highdis)
t.test(x_lowhor, y_highhor)
t.test(x_lowwei, y_highwei)
t.test(x_lowacc, y_highacc)
end_time <- Sys.time()
end_time - start_time # Computation time: 0.0977459 secs
```