Homework 1 (Class 6350)

Student: **Thanh Hung Duong**

Hsien Hao Hsu

Contribution:

Thanh Hung Duong	Coded the draft of R program, refined the report.
Hsien Hao Hsu	Refined the R program, wrote the report.

Part 1.

1) Compute the mean and standard deviation of each feature

Table 1. Mean and Std of 5 features

	cyl	**dis**	**hor**	**wei**	**acc**
Means	5.471939	194.411990	104.469388	2977.584184	15.541327
Standard deviation	1.705783	104.644004	38.491160	849.402560	2.758864

Mean $\bar{X} = \frac{X1+\cdots+Xn}{n}$

Standard deviation $\sigma = \sqrt{\frac{\Sigma(X-\bar{X})^2}{n}}$

2) Display the histogram of each feature, and the histogram of mpg

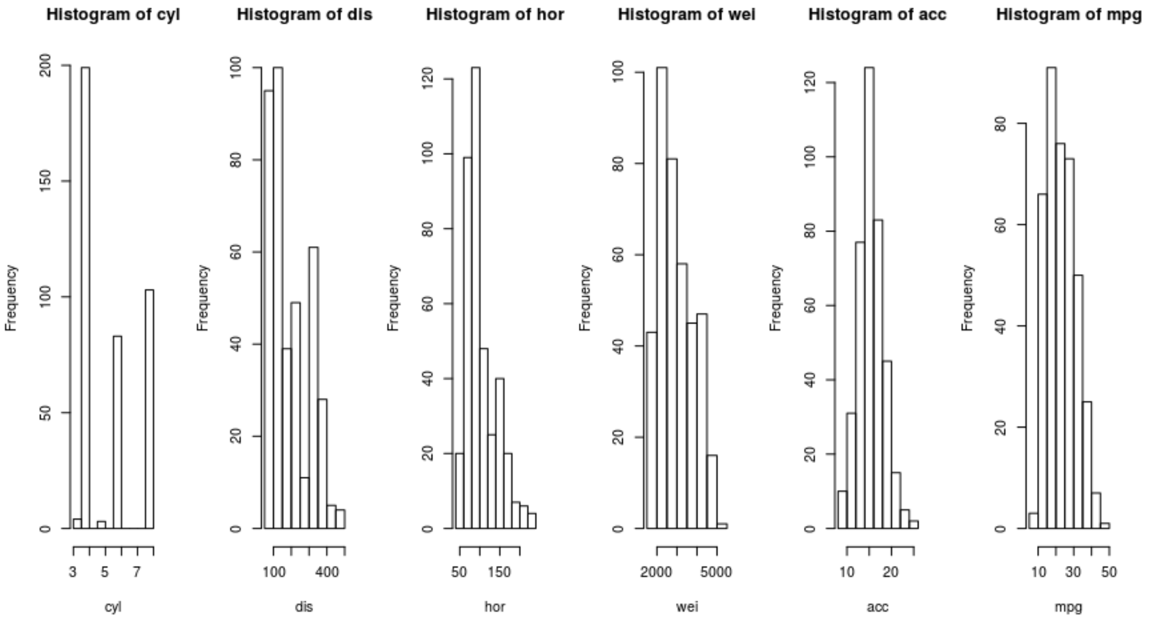

Figure 1. Histograms of 5 features and response variable

3) Display the 5 following scatterplots

(cyl , mpg) , (dis , mpg) , (hor , mpg) , (wei , mpg) , (acc , mpg)

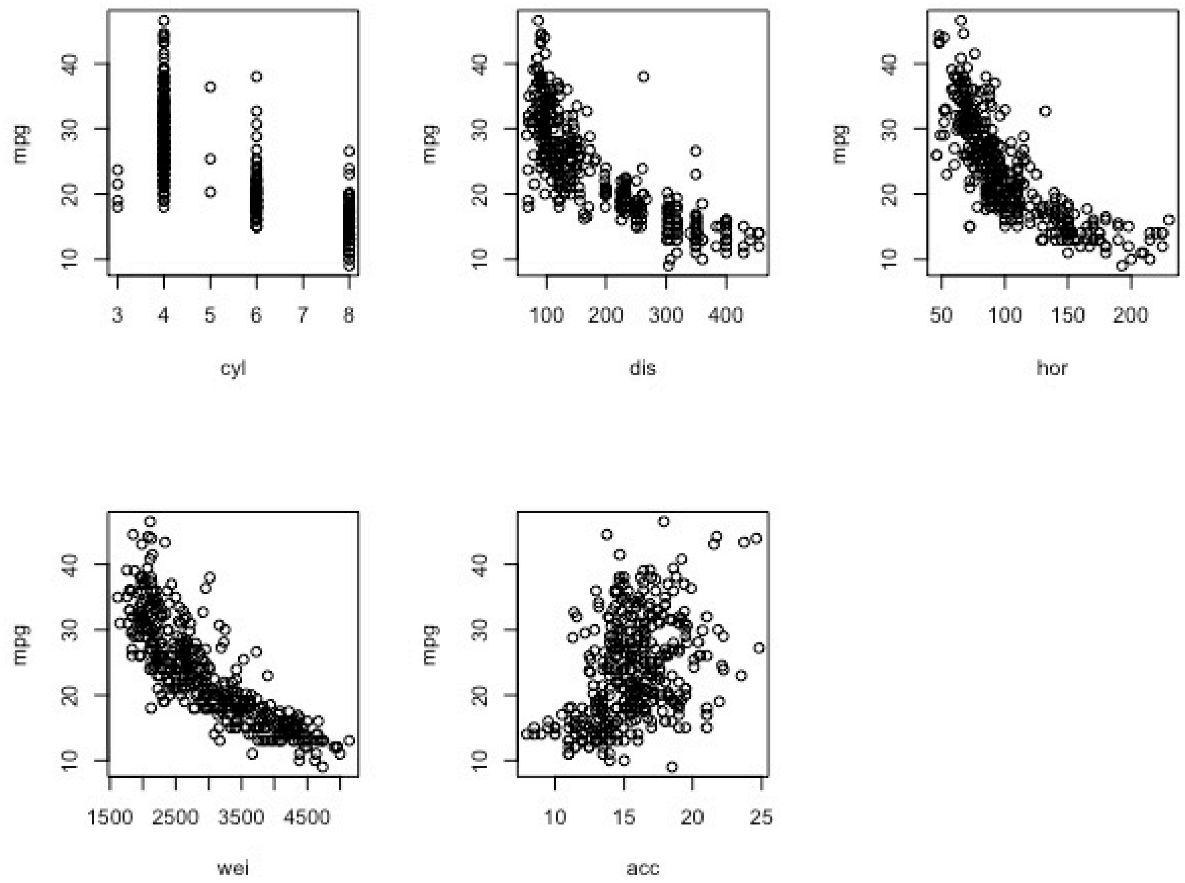

Figure 2. Scatterplot of five features

4) Interpret these 5 scatterplots to guess which features may have stronger capacity to predict mpg

Among 5 above plots, only scattering dots in (dis,mpg), (hor,mpg) and (wei,mpg) shaped a curve like a function while dots in (acc,mpg) appeared randomly so we think that dis, hor and wei feature may have stronger capacity to predict mpg. Within these feature, because there are several gaps in the middle of the (dis,mpg) graph and in the right bottom of the (hor,mpg), **we believe that the wei may be the best feature to predict mpg**.

5) Compute the 5 correlations cor(cyl , mpg) , cor(dis , mpg) , cor(hor , mpg), cor(wei , mpg), cor(acc , mpg) interpret these correlations to guess which features may have stronger capacity to predict msg

Table 2. Correlations of 5 features

cyl	dis	hor	wei	acc
-0.7776175	-0.8051269	-0.7784268	-0.8322442	0.4233285

Covariance $Cov(x, y) = \frac{\sum(x_i - \bar{x})(y_i - \bar{y})}{n}$

Correlation $r_{xy} = \frac{Cov(x,y)}{\sigma_x + \sigma_y}$

6) Compute the covariance matrix COV and the correlation matrix CORR of the 5 features

"The covariance matrix of 5 features is "

$$COV = \begin{bmatrix} 2.91 & 169.72 & 55.35 & 1300.42 & -2.38 \\ 169.72 & 10950.37 & 3614.03 & 82929.10 & -156.99 \\ 55.35 & 3614.03 & 1481.57 & 28265.62 & -73.19 \\ 1300.42 & 82929.10 & 28265.62 & 721484.71 & -976.82 \\ -2.38 & -156.99 & -73.19 & -976.82 & 7.61 \end{bmatrix}$$

"The correlation matrix of 5 features is"

$$CORR = \begin{bmatrix} 1.000 & 0.951 & 0.843 & 0.898 & 0.505 \\ 0.951 & 1.000 & 0.897 & 0.933 & -0.544 \\ 0.843 & 0.897 & 1.000 & 0.865 & -0.689 \\ 0.898 & 0.933 & 0.865 & 1.000 & -0.417 \\ 0.505 & -0.544 & -0.689 & -0.417 & 1.000 \end{bmatrix}$$

7) Compute the 5 eigenvalues L1 >L2>L3>L4>L5 of CORR

Table 3. Five Eigen values of CORR matrix

L1	L2	L3	L4	L5
4.07185982	0.69386125	0.13349305	0.06426839	0.03651750

Eigen value L_i => $A \times v_i = L_i \times v_i$

8) Verify that L1+L2 +...+L5 =5

$\sum_{i=1}^{5} Li$ =4.07185982+0.69386125+0.13349305+0.06426839+0.03651750 = 5

9) For i =1 2 3 4 5 compute the ratios Ri = (L1+L2+ ... +Li)/5

Table 4. Ratio Ri

R1	R2	R3	R4	R5
0.8143720	0.9531442	0.9798428	0.9926965	1.0000000

*10) interpret these 5 ratios

The ratios mean that how much accurate is the analysis is going to be if we project all the cases to the eigen values. For example, if we project all the cases onto R1, the first eigenvalue, then we can analyze the 1-dim projection on a plane and the accuracy will be around 81.44%. Moreover, if we project all the cases onto R1 and R2, then we can analyze the 2-dim projection in a space for a 95.31% accuracy. By projecting all the cases onto all five eigen values, we can get the exact result as we are analyzing. However, 5-dim projection is too difficult to observe. Therefore, we will take the result by projecting all the cases onto R1 and R2 because it has 95% accuracy which is a generally acceptable result.

11) Reorder the rows of the .csv data set so that the first column msg becomes increasing, separate then the data set into two tables,

- **The LOWmsg table will include all the cases for which msg is inferior to median(msg)**
- **The HIGHmsg table will include all the cases for which msg is larger than median(msg)**

12&13) Let F be any one of the five features cyl, dis, hor, wei, acc

Display side by side

- **The histogram histlow(F) computed on the F values corresponding to the cases belonging to the table LOWmsg**
- **The histogram histhigh(F) computed on the F values corresponding to the cases belonging to the table HIGHmsg**

This will give you 5 pairs of histograms, one pair for each feature F

Interpret each one of these 5 pairs of histograms to guess which feature has a good capacity to discriminate between high mpg and low mpg

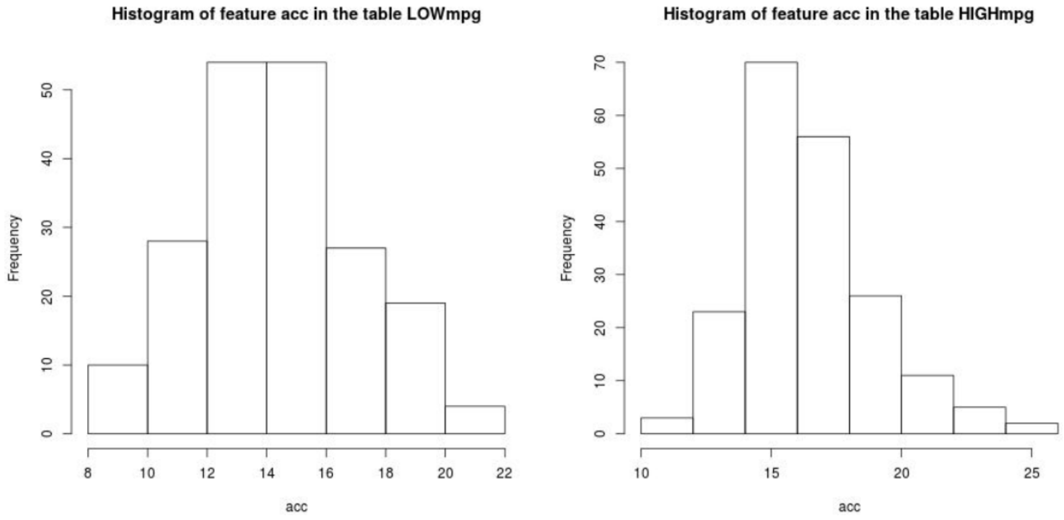

Figure 3. Histograms of feature acc in 2 table LOWmpg and HIGHmpg

As seen from the histograms of feature acc, the highest frequencies of acc values in table LOWmpg and HIGHmpg are similar, which is around from 14 to 16. Therefore, this feature does not have a good capacity to discriminate between high mpg and low mpg

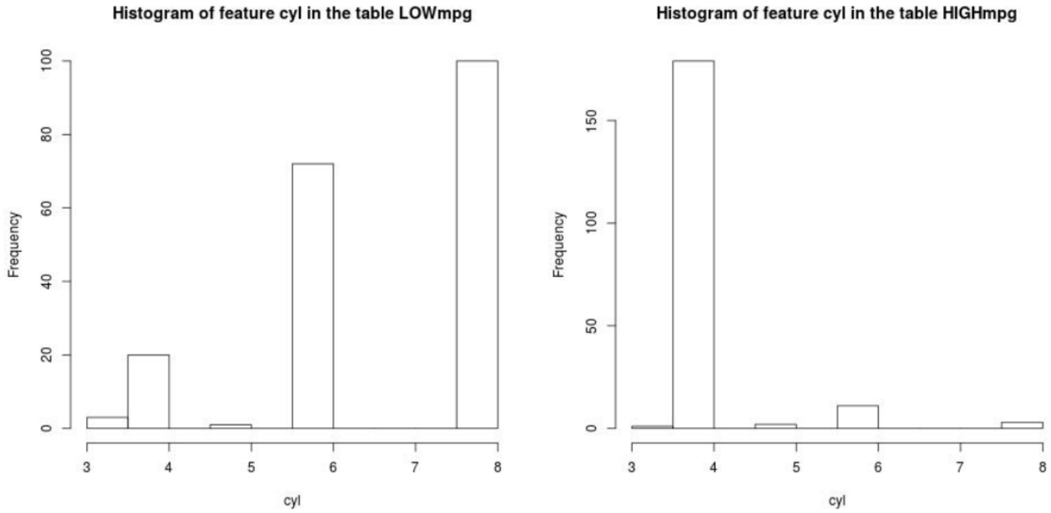

Figure 4.Histograms of feature cyl in 2 table LOWmpg and HIGHmpg

From the histogram in the graph we can see that feature cyl is much more distinguished between highmpg and low mpg than the other features. The reason is that we can see when cyl = 8 then it directly belong to lowmpg and when cyl = 6 then it mostly belong to lowmpg too and when cyl = 4 then it likely belong to highmpg. **Therefore, we guess that feature cyl has a good capacity to discriminate between high msg and low msg.**

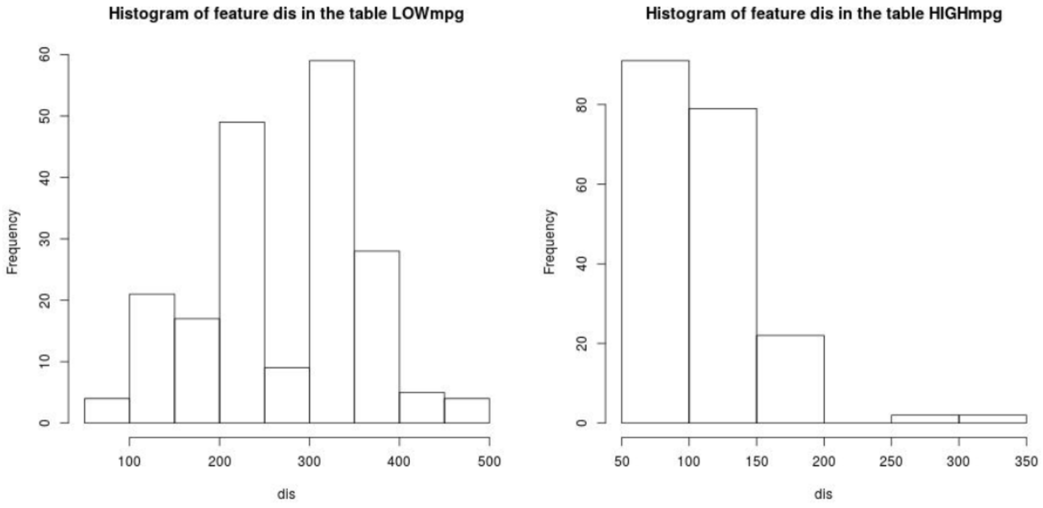

Figure 5.Histograms of feature dis in 2 table LOWmpg and HIGHmpg

In case of displacement feature, most of dis values in table LOWmpg distributes in the range from 200 to 400, while those in HIGHmpg are mainly from 50 to 200. So we believe that this feature can be used to discriminate between high and low mpgs.

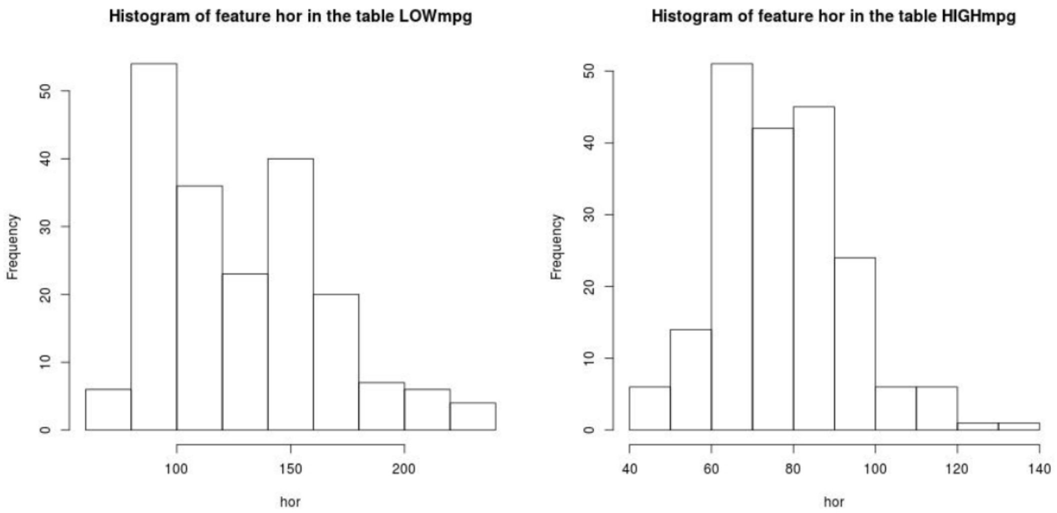

Figure 6. Histograms of feature hor in 2 table LOWmpg and HIGHmpg

In the histogram of feature hor in the table LOWmpg, despite a large number of hor feature appears in the range from 10 to 200, the hor values that are smaller than 100 have highest frequency. Because most of the hor data in the table HIGHmpg are between 60 and 100, it can be concluded that the feature hor is can be used to apply in the discrimination between high and low mpgs but its discrimination capacity may be not as good as other features.

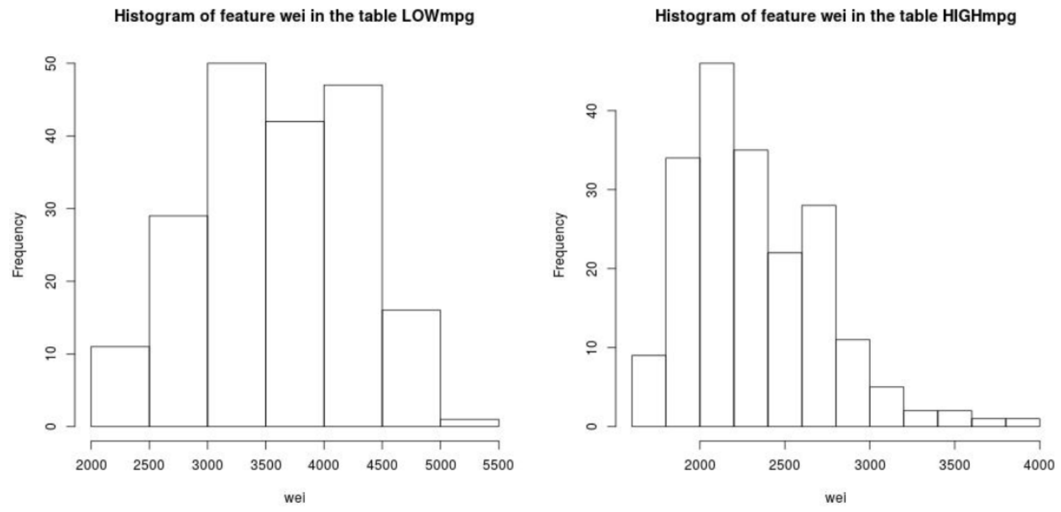

Figure 7.Histograms of feature wei in 2 table LOWmpg and HIGHmpg

Lastly, it can be seen clearly from the Fig.7 that most of weight values that smaller than 3000 are in the table HIGHmpg, otherwise, those that larger than 3000 belong the table LOWmpg. Therefore, it can be said that the feature weight has a good capacity to discriminate between high and low mpgs

14) Successively, for each one of the five features (denoted F)

- **Compute the mean mlow(F) and standard deviation stdlow(F) of the F values corresponding to the cases belonging to the table LOWmsg**
- **Compute the mean mhigh(F) and standard deviation stdhigh(F) of the F values corresponding to the cases belonging to the table HIGHmsg**

Table 5.Mean and Std of 5 features in table LOWmpg and HIGHmpg

	cyl	**dis**	**hor**	**wei**	**acc**
mlow(F)	6.77	273.16	130.11	3620.40	14.56
stdlow(F)	1.42	89.52	37.36	676.93	2.69
mhigh(F)	4.18	115.67	78.83	2334.77	16.50
Stdhigh(F)	0.67	38.43	15.92	397.19	2.49

***15) For each feature F compute the ratio**
discr(F) = | mhigh(F) - mlow(F) | / s(F) where s(F) = (stdlow(F)+ stdhigh(F))/sqrt(N)
this ratio is a simple but very rough characterization of the "discriminating power" of feature F when one attempts to evaluate the capacity of feature F to discriminate between low mpg and high mpg ;
after doing this for the five features F1 F2 F3 F4 F5 , use the five ratios discr(Fj) with j=1 2 3 4 5 to compare how helpful each feature Fj may be for the task of discriminating between low mpg and high mpg ;
- compute DISCR(F) = 1 -pval(F) which verifies 0 < DISCR(F) < 1

higher values of DISCR(F) tend to indicate that F may have stronger capcity to help disriminate between low mpg and high mpg

in standard statistical evaluations, a high value such as DISCR(F) > 95% is considered as a very strong indication that mhigh(F) and mlow(F) are significantly distinct

Table 6. Discrimination ratio for the capacity of 5 features

cyl	**dis**	**hor**	**wei**	**acc**
24.45	24.37	19.06	23.70	7.31

From the numbers we compute from the formula. We can evaluate that features cyl, dis and wei which have the largest amount, are more likely to have a very good capacity to discriminate between low mpg and high mpg. On the other hand, because feature acc has the smallest ratio, it has the lowest capacity of the discrimination of the response mpg.

Next, we used t.test() function to obtain p-values of all five features and the results is displayed in the table 7. First of all, all p-values are difference and much smaller than 1, which means all features can be used to discriminate between low mpg and high mpg but their capabilities are different. Because, the features cyl, dis and wei have the smallest p-value, we can conclude strongly that those features possess the strongest capacities to discriminate between low mpg and high mpg. On the other hand, p value of acceleration feature is the largest one, so this feature has the weakest capacity of the discrimination. Those conclusions matched the interpretation of discrimination ratios and histograms of 5 features.

To summarize, the capability C of five features to discriminate between low mpg and high mpg can be ranked as the following order: $C_{cyl} \approx C_{dis} \approx C_{wei} > C_{hor} > C_{acc}$

Table 7. P value of five features

cyl	dis	hor	wei	acc
1.77e-66	8.10e-64	1.14e-46	3.86e-69	1.62e-12

Part 2.

1) Compute the mean and standard deviation of each feature

```
data_auto= read.csv("Auto.csv") # import data
data_auto=data_auto[,-(7:9)] #remove 3 last column

#Remove rows that contain missing data
#Because missing data appears as '?', na.omit does not work
missing_rows=which(data_auto == '?',TRUE)[,1] # Find cell that contain '?' and its indexes
data_auto=data_auto[-missing_rows,] # Remove the above rows

#Indicate the number N of cases which are kept
print(paste("The real number of cases is",length(data_auto[,1])))

#Denote 5 explanatory variable
var_list=list('mpg','cyl', 'dis', 'hor', 'wei', 'acc')
names(data_auto)=var_list

# 1. Compute the mean and standard deviation of each feature
start_time = Sys.time() # obtain the start time
```

```
sapply(data_auto, is.numeric) # Check if there is a non-numeric data. In our case, data_auto$hor is nonnumeric data
class(data_auto$hor) # Find out what type of the data_auto$hor is
# Because the type of data_auto$hor is factor, it must be converted to character type first and then numeric type
data_auto$hor=as.numeric(as.character(data_auto$hor)) # Convert this column into numeric data

feature_means=colMeans(data_auto[,2:6]) # Mean of all features
feature_sd=apply(data_auto[,2:6], 2, sd) # Standard deviation of each feature

print('Means of each feature are')
print(feature_means)
print('Standard deviation of each feature are')
print(feature_sd)

end_time = Sys.time()
print(paste(' Computing time for question 1 is:', end_time - start_time, ' second'))
```

" Computing time for question 1 is: 0.00160574913024902 second"

2) Display the histogram of each feature, and the histogram of mpg

```
start_time = Sys.time() # obtain the start time

par(mfrow=c(1,6)) # setup multiplots
for (i in 2:6) {
  hist(data_auto[,i],main = paste('Histogram of',var_list[i]),xlab=var_list[i])
}
hist(data_auto[,1],main = paste('Histogram of',var_list[1]),xlab=var_list[1])

end_time = Sys.time()
print(paste(' Computing time for question 2 is:', end_time - start_time, ' second'))
```

```
" Computing time for question 2 is: 0.0287132263183594 second"
```

3) Display the 5 following scatterplots

(cyl , mpg) , (dis , mpg) , (hor , mpg) , (wei , mpg) , (acc , mpg)

```
start_time = Sys.time()

par(mfrow=c(2,3)) # setup multiplots
for (i in 2:6) {
  plot(data_auto[,i],data_auto$mpg,ylab = 'mpg',xlab = var_list[i])
}

end_time = Sys.time()
print(paste(' Computing time for question 3 is:', end_time - start_time, ' second'))
```

```
" Computing time for question 3 is: 0.0382888317108154  second"
```

4) Interpret these 5 scatterplots to guess which features may have stronger capacity to predict msg

5) Compute the 5 correlations

cor(cyl , mpg) , cor(dis , mpg) , cor(hor , mpg), cor(wei , mpg), cor(acc , mpg)

interpret these correlations to guess which features may have stronger capacity to predict   msg

```
start_time = Sys.time()


feature_corrs=NULL

for (i in 2:6) {

  feature_corrs[i-1]=cor(data_auto[,i],data_auto$mpg)

}


end_time = Sys.time()

print(paste(' Computing time for question 5 is:', end_time - start_time, ' second'))
```

```
" Computing time for question 5 is: 0.00344514846801758 second"
```

6) Compute the covariance matrix COV and the correlation matrix CORR of the 5 features

```
start_time = Sys.time()

data_feature=data_auto[,-1]
feature_matrix_cov=cov(data_feature)
feature_matrix_cor=cor(data_feature)

print('The covariance matrix of 5 features is ')
print(round(feature_matrix_cov,digits=2))
print('The correlation matrix of 5 features is')
print(round(feature_matrix_cor,digits=3))

end_time = Sys.time()
print(paste(' Computing time for question 6 is:', end_time - start_time, ' second'))
```

" Computing time for question 6 is: 0.00136089324951172 second"

7) Compute the 5 eigenvalues L1 >L2>L3>L4>L5 of CORR

```
start_time = Sys.time()

feature_eigen_values= eigen(feature_matrix_cor)$value

end_time = Sys.time()
print(paste(' Computing time for question 7 is:', end_time - start_time, ' second'))
```

" Computing time for question 7 is: 0.00173664093017578 second"

8) Verify that L1+L2 +...+L5 =5

```
start_time = Sys.time()

print(paste('Sum of all eigen values is ',ceiling(sum(feature_eigen_values))))
```

```
end_time = Sys.time()

print(paste(' Computing time for question 7 is:', end_time - start_time, ' second'))
```

" Computing time for question 8 is: 9.98973846435547e-05 second"

9) For i =1 2 3 4 5 compute the ratios Ri = (L1+L2+ ... +Li)/5

```
start_time = Sys.time()

R=NULL
for (i in 1:5) {
  R[i]=sum(feature_eigen_values[1:i])/5
}
print(R)

end_time = Sys.time()
print(paste(' Computing time for question 9 is:', end_time - start_time, ' second'))
```

" Computing time for question 9 is: 0.00304937362670898 second"

10) interpret these 5 ratios

11) reorder the rows of the .csv data set so that the first column msg becomes increasing separate then the data set into two tables,
the LOWmsg table will include all the cases for which msg is inferior to median(msg)
the HIGHmsg table will include all the cases for which msg is larger than median(msg)

```
start_time = Sys.time()

data_reorder=data_auto[order(mpg),] # Sort data by mpg
median_mpg=median(data_reorder$mpg)
LOWmpg=NULL
```

```
HIGHmpg=NULL

splitted_data=split(data_reorder,data_reorder[,1]>median_mpg)

LOWmpg=data.frame(splitted_data[1])

HIGHmpg=data.frame(splitted_data[2])


end_time = Sys.time()

print(paste(' Computing time for question 11 is:', end_time - start_time, ' second'))
```

" Computing time for question 11 is: 0.0020756721496582 second"

12)

Let F be any one of the five features cyl, dis, hor, wei, acc

display side by side

the histogram histlow(F) computed on the F values corresponding to the cases belonging to the table LOWmsg

the histogram histhigh(F) computed on the F values corresponding to the cases belonging to the table HIGHmsg

This will give you 5 pairs of histograms, one pair for each feature F

```
start_time = Sys.time()


for (i in 2:6) {

  # Open a jpeg file

  jpeg(paste("Histogram of feature",var_list[i],".jpeg"), width = 960, height = 480, units = "px", pointsize = 12,

       quality = 75)

  # 2. Create a plot

  mtitle=paste('Histogram of feature',var_list[i])

  par(mfrow=c(1,2))

  hist(LOWmpg[,i],main=paste(mtitle,'in the table LOWmpg'),xlab=var_list[i])

  hist(HIGHmpg[,i],main=paste(mtitle,'in the table HIGHmpg'),xlab=var_list[i])
  # Close the jpeg file
  dev.off()

}
end_time = Sys.time()
print(paste(' Computing time for question 12 is:', end_time - start_time, ' second'))
```

```
" Computing time for question 12 is: 0.0918033123016357  second"
```

13) interpret each one of these 5 pairs of histograms to guess which feature has a good capacity to discriminate between high msg and low msg

14) successively, for each one of the five features (denoted F)

compute the mean mlow(F) and standard deviation stdlow(F) of the F values corresponding to the cases belonging to the table LOWmsg

compute the mean mhigh(F) and standard deviation stdhigh(F) of the F values corresponding to the cases belonging to the table HIGHmsg

```
start_time = Sys.time()


feature_mlow=colMeans(LOWmpg[,2:6]) # Mean of all features
feature_stdlow=apply(LOWmpg[,2:6], 2, sd) # Standard deviation of each feature
print(round(feature_mlow,2))
print(round(feature_stdlow,2))
feature_mhigh=colMeans(HIGHmpg[,2:6]) # Mean of all features
feature_stdhigh=apply(HIGHmpg[,2:6], 2, sd) # Standard deviation of each feature
print(round(feature_mhigh,2))
print(round(feature_stdhigh,2))


end_time = Sys.time()
print(paste(' Computing time for question 14 is:', end_time - start_time, ' second'))
```

" Computing time for question 14 is: 0.00150656700134277 second"

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

use these numbers to rughly evaluate the capacity of feature F to discriminate between low msg and high msg

```
start_time = Sys.time()


N=length(data_auto$mpg)
sF=(feature_stdhigh+feature_stdlow)/sqrt(N)
difference_F=abs(feature_mhigh-feature_mlow)
discr_F = round(difference_F/sF,2)


print(discr_F)
pval_F=NULL
for (i in 2:6) {
  pval_F[i-1]=t.test(LOWmpg[,i],HIGHmpg[,i])$p.value
}
print(pval_F)
print(1-pval_F)


end_time = Sys.time()
print(paste(' Computing time for question 15 is:', end_time - start_time, ' second'))
```

" Computing time for question 15 is: 0.00501322746276855 second"