R Notebook

Code

Hide

```
#Loading and cleaning the data
attach(breast.cancer.wisconsin)
```

```
The following objects are masked from breast.cancer.wisconsin (pos = 10):

    V1, V10, V11, V2, V3, V4, V5, V6, V7, V8, V9

The following objects are masked from breast.cancer.wisconsin (pos = 11):

    V1, V10, V11, V2, V3, V4, V5, V6, V7, V8, V9

The following objects are masked from breast.cancer.wisconsin (pos = 15):

    V1, V10, V11, V2, V3, V4, V5, V6, V7, V8, V9

The following objects are masked from breast.cancer.wisconsin (pos = 16):

    V1, V10, V11, V2, V3, V4, V5, V6, V7, V8, V9

The following objects are masked from breast.cancer.wisconsin (pos = 17):

    V1, V10, V11, V2, V3, V4, V5, V6, V7, V8, V9

The following objects are masked from breast.cancer.wisconsin (pos = 18):

    V1, V10, V11, V2, V3, V4, V5, V6, V7, V8, V9

The following objects are masked from breast.cancer.wisconsin (pos = 19):

    V1, V10, V11, V2, V3, V4, V5, V6, V7, V8, V9

The following objects are masked from breast.cancer.wisconsin (pos = 20):

    V1, V10, V11, V2, V3, V4, V5, V6, V7, V8, V9

The following objects are masked from breast.cancer.wisconsin (pos = 21):

    V1, V10, V11, V2, V3, V4, V5, V6, V7, V8, V9

The following objects are masked from breast.cancer.wisconsin (pos = 22):

    V1, V10, V11, V2, V3, V4, V5, V6, V7, V8, V9

The following objects are masked from breast.cancer.wisconsin (pos = 23):

    V1, V10, V11, V2, V3, V4, V5, V6, V7, V8, V9
```

Hide

```
#Missing Value check and Solution
breast.cancer.wisconsin$V7 <- replace(breast.cancer.wisconsin$V7, breast.cancer.wisconsin$V7
== "?", NA)
breast.cancer.wisconsin$V7 <- as.integer(breast.cancer.wisconsin$V7)

breast_cancer_wisconsin<-na.omit(breast.cancer.wisconsin)
breast_cancer_wisconsin
```

	V1 <int>	V2 <int>	V3 <int>	V4 <int>	V5 <int>	V6 <int>	V7 <int>	V8 <int>	V9 <int>
1	1000025	5	1	1	1	2	1	3	1
2	1002945	5	4	4	5	7	10	3	2
3	1015425	3	1	1	1	2	2	3	1
4	1016277	6	8	8	1	3	4	3	7
5	1017023	4	1	1	3	2	1	3	1
6	1017122	8	10	10	8	7	10	9	7
7	1018099	1	1	1	1	2	10	3	1
8	1018561	2	1	2	1	2	1	3	1
9	1033078	2	1	1	1	2	1	1	1
10	1033078	4	2	1	1	2	1	2	1

1-10 of 683 rows | 1-10 of 11 columns Previous **1** 2 3 4 5 6 ... 69 Next

Hide

```
#1-Descriptive Dataset's Info
#Class Label- 2 - Benign and 4- Malignant
names(breast_cancer_wisconsin)<-c('ID', 'C_Thickness','UCSize', 'UCShape', 'M_Adhesion', 'Sin
gle_ECSize', 'Bare_Nuclei', 'Bland_Chromatin', 'Normal_Nucleoli', 'Mitoses', 'Class_Label')

#Encondings of Class Label
breast_cancer_wisconsin$Class_Label<-ifelse(breast_cancer_wisconsin$Class_Label== 2, 0, 1)
breast_cancer_wisconsin$Class_Label
```

```
  [1] 0 0 0 0 0 1 0 0 0 0 0 0 1 0 1 1 0 0 1 0 1 1 0 0 1 0 0 0 0 0 0 1 0 0 0 1 0 1 1
 [40] 1 1 1 1 0 1 0 0 1 1 1 1 1 1 1 1 1 1 1 1 0 1 1 0 1 0 1 1 0 0 1 0 1 1 0 0 0 0 0
 [79] 0 0 0 0 1 1 1 1 0 0 0 0 0 0 0 0 0 0 1 1 1 1 0 1 1 1 1 1 0 1 0 1 1 1 0 0 0 1 0
[118] 0 0 0 1 1 1 0 1 0 1 0 0 0 1 0 0 0 0 0 0 0 0 1 0 0 1 0 0 1 0 1 1 0 0 1 0 0 1 1
[157] 0 0 0 0 1 1 0 0 0 0 0 1 1 1 0 1 0 1 0 0 0 1 1 0 1 1 1 0 1 1 0 0 0 0 0 0 0 0 1
[196] 1 0 0 0 1 1 0 0 0 1 1 0 1 1 1 0 0 1 0 0 1 1 1 1 0 1 1 0 1 1 1 0 1 0 1 1 1 1 0
[235] 0 0 0 0 0 1 1 0 0 1 0 1 1 1 0 0 0 0 1 1 1 1 1 0 1 1 1 0 1 0 1 1 0 0 0 0 1 0 0
[274] 1 1 1 1 1 0 1 1 0 0 1 1 0 0 1 1 0 1 0 1 1 0 0 1 0 0 0 1 0 0 1 1 0 0 1 0 1 0 0
[313] 1 0 1 1 1 0 0 1 1 0 1 0 0 1 1 0 0 0 1 0 0 0 1 1 0 0 0 1 0 0 1 1 1 1 1 1 0 0 0
[352] 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 1 0 0 0 0 1 0 0 0 0 0 0 0 0 1 0 0 0
[391] 0 0 0 0 0 0 0 1 0 1 0 1 0 0 0 0 1 0 0 0 1 0 1 0 0 0 0 0 0 0 1 1 0 0 0 1 0 0 0
[430] 0 0 0 0 0 1 0 0 0 1 0 1 1 1 0 0 0 0 0 0 0 1 1 1 0 0 0 0 0 0 0 0 0 0 0 1 0 0 1
[469] 1 0 0 0 1 1 1 0 1 0 1 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 1 1 0 0 0 1 0 0
[508] 1 1 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 1 0 0 0 0 0 0 0 0 0 0 0
[547] 0 0 0 0 1 0 0 1 1 1 1 0 0 1 0 0 0 0 0 0 1 1 0 0 0 1 0 1 0 1 1 1 0 1 0 0 0 0 0
[586] 0 0 0 1 1 1 0 0 1 0 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 1 0 0 1 0 0 0
[625] 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 1 1 1 0 0 0 0 0 0 0 0
[664] 0 1 1 0 0 0 0 0 0 0 0 0 1 0 0 0 0 1 1 1
```

Hide

```
summary(breast_cancer_wisconsin)
```

```
       ID             C_Thickness         UCSize          UCShape      
 Min.   :   63375   Min.   : 1.000   Min.   : 1.000   Min.   : 1.000  
 1st Qu.:  877617   1st Qu.: 2.000   1st Qu.: 1.000   1st Qu.: 1.000  
 Median : 1171795   Median : 4.000   Median : 1.000   Median : 1.000  
 Mean   : 1076720   Mean   : 4.442   Mean   : 3.151   Mean   : 3.215  
 3rd Qu.: 1238705   3rd Qu.: 6.000   3rd Qu.: 5.000   3rd Qu.: 5.000  
 Max.   :13454352   Max.   :10.000   Max.   :10.000   Max.   :10.000  
   M_Adhesion    Single_ECSize     Bare_Nuclei      Bland_Chromatin  Normal_Nucleoli
 Min.   : 1.00   Min.   : 1.000   Min.   : 1.000   Min.   : 1.000   Min.   : 1.00  
 1st Qu.: 1.00   1st Qu.: 2.000   1st Qu.: 1.000   1st Qu.: 2.000   1st Qu.: 1.00  
 Median : 1.00   Median : 2.000   Median : 1.000   Median : 3.000   Median : 1.00  
 Mean   : 2.83   Mean   : 3.234   Mean   : 3.545   Mean   : 3.445   Mean   : 2.87  
 3rd Qu.: 4.00   3rd Qu.: 4.000   3rd Qu.: 6.000   3rd Qu.: 5.000   3rd Qu.: 4.00  
 Max.   :10.00   Max.   :10.000   Max.   :10.000   Max.   :10.000   Max.   :10.00  
    Mitoses         Class_Label    
 Min.   : 1.000   Min.   :0.0000  
 1st Qu.: 1.000   1st Qu.:0.0000  
 Median : 1.000   Median :0.0000  
 Mean   : 1.603   Mean   :0.3499  
 3rd Qu.: 1.000   3rd Qu.:1.0000  
 Max.   :10.000   Max.   :1.0000  
```

Hide

```
str(breast_cancer_wisconsin)
```

```
'data.frame':	683 obs. of  11 variables:
 $ ID             : int  1000025 1002945 1015425 1016277 1017023 1017122 1018099 1018561 1033
078 1033078 ...
 $ C_Thickness    : int  5 5 3 6 4 8 1 2 2 4 ...
 $ UCSize         : int  1 4 1 8 1 10 1 1 1 2 ...
 $ UCShape        : int  1 4 1 8 1 10 1 2 1 1 ...
 $ M_Adhesion     : int  1 5 1 1 3 8 1 1 1 1 ...
 $ Single_ECSize  : int  2 7 2 3 2 7 2 2 2 2 ...
 $ Bare_Nuclei    : int  1 10 2 4 1 10 10 1 1 1 ...
 $ Bland_Chromatin: int  3 3 3 3 3 9 3 3 1 2 ...
 $ Normal_Nucleoli: int  1 2 1 7 1 7 1 1 1 1 ...
 $ Mitoses        : int  1 1 1 1 1 1 1 1 5 1 ...
 $ Class_Label    : num  0 0 0 0 0 1 0 0 0 0 ...
 - attr(*, "na.action")= 'omit' Named int [1:16] 24 41 140 146 159 165 236 250 276 293 ...
  ..- attr(*, "names")= chr [1:16] "24" "41" "140" "146" ...
```

Hide

```
#2- Correlation between attributes
cor(breast_cancer_wisconsin)
```

```
                         ID C_Thickness      UCSize     UCShape  M_Adhesion
ID               1.00000000 -0.05634966 -0.04139605 -0.04222123 -0.06963009
C_Thickness     -0.05634966  1.00000000  0.64248149  0.65346999  0.48782872
UCSize          -0.04139605  0.64248149  1.00000000  0.90722823  0.70697695
UCShape         -0.04222123  0.65346999  0.90722823  1.00000000  0.68594806
M_Adhesion      -0.06963009  0.48782872  0.70697695  0.68594806  1.00000000
Single_ECSize   -0.04864387  0.52359604  0.75354402  0.72246241  0.59454777
Bare_Nuclei     -0.09924781  0.59309144  0.69170875  0.71387755  0.67064829
Bland_Chromatin -0.06196640  0.55374245  0.75555916  0.73534350  0.66856706
Normal_Nucleoli -0.05069861  0.53406591  0.71934604  0.71796341  0.60312106
Mitoses         -0.03797243  0.35095717  0.46075470  0.44125758  0.41889833
Class_Label     -0.08470103  0.71478993  0.82080144  0.82189095  0.70629414
                Single_ECSize Bare_Nuclei Bland_Chromatin Normal_Nucleoli
ID                -0.04864387 -0.09924781      -0.0619664     -0.05069861
C_Thickness        0.52359604  0.59309144       0.5537424      0.53406591
UCSize             0.75354402  0.69170875       0.7555592      0.71934604
UCShape            0.72246241  0.71387755       0.7353435      0.71796341
M_Adhesion         0.59454777  0.67064829       0.6685671      0.60312106
Single_ECSize      1.00000000  0.58571613       0.6181279      0.62892640
Bare_Nuclei        0.58571613  1.00000000       0.6806149      0.58428020
Bland_Chromatin    0.61812790  0.68061486       1.0000000      0.66560153
Normal_Nucleoli    0.62892640  0.58428020       0.6656015      1.00000000
Mitoses            0.48058330  0.33921044       0.3460109      0.43375727
Class_Label        0.69095816  0.82269587       0.7582276      0.71867719
                    Mitoses Class_Label
ID              -0.03797243 -0.08470103
C_Thickness      0.35095717  0.71478993
UCSize           0.46075470  0.82080144
UCShape          0.44125758  0.82189095
M_Adhesion       0.41889833  0.70629414
Single_ECSize    0.48058330  0.69095816
Bare_Nuclei      0.33921044  0.82269587
Bland_Chromatin  0.34601089  0.75822755
Normal_Nucleoli  0.43375727  0.71867719
Mitoses          1.00000000  0.42344792
Class_Label      0.42344792  1.00000000
```

Hide

```
#3-Divide the data into training(80%) and testing(20%)

#a) Divide the data into training and testing sets
library(caret)
set.seed(1) # for reproducibility

# Create a vector of row indices
rows <- 1:nrow(breast_cancer_wisconsin)

# Randomly sample 80% of the row indices for the training set
training_rows <- sample(rows, floor(0.8 * length(rows)))

# The remaining rows are for the testing set
testing_rows <- setdiff(rows, training_rows)

training_data <- breast_cancer_wisconsin[training_rows, ]
testing_data <- breast_cancer_wisconsin[-training_rows, ]

# Create X.train and X.test data frames that exclude the class label
X.train <- training_data[, -which(names(training_data) == "Class_Label")]
X.test <- testing_data[, -which(names(testing_data) == "Class_Label")]
Y.train <- training_data$Class_Label
Y.test <- testing_data$Class_Label

#b) Division Verification in number of Examples
cat("Number of examples in training data:", nrow(training_data), "\n")
```

```
Number of examples in training data: 546
```

Hide

```
cat("Number of examples in testing data:", nrow(testing_data), "\n")
```

```
Number of examples in testing data: 137
```

Hide

```
#4- Logistic Regression Model Generation

#Logistic Regression fit on the training data
LRModel = glm(Class_Label~.,training_data, family = binomial)

# Use the model to make predictions on the testing data
LR_Prediction<-predict(LRModel, testing_data)
LR_Prediction <- ifelse(LR_Prediction > 0.5, "0", "1")
summary(LRModel)
```

```

Call:
glm(formula = Class_Label ~ ., family = binomial, data = training_data)

Deviance Residuals: 
     Min        1Q    Median        3Q       Max  
-2.42565  -0.07391  -0.03251   0.00706   2.51136  

Coefficients:
                  Estimate Std. Error z value Pr(>|z|)    
(Intercept)     -1.243e+01  2.244e+00  -5.539 3.04e-08 ***
ID               4.766e-08  9.949e-07   0.048 0.961793    
C_Thickness      6.698e-01  1.834e-01   3.652 0.000260 ***
UCSize          -2.048e-01  2.542e-01  -0.806 0.420397    
UCShape          3.652e-01  2.763e-01   1.322 0.186121    
M_Adhesion       4.904e-01  1.564e-01   3.136 0.001710 ** 
Single_ECSize    2.440e-02  1.932e-01   0.126 0.899491    
Bare_Nuclei      4.782e-01  1.242e-01   3.850 0.000118 ***
Bland_Chromatin  5.896e-01  2.249e-01   2.622 0.008744 ** 
Normal_Nucleoli  4.095e-01  1.560e-01   2.624 0.008678 ** 
Mitoses          9.444e-01  3.267e-01   2.891 0.003839 ** 
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 716.293  on 545  degrees of freedom
Residual deviance:  63.735  on 535  degrees of freedom
AIC: 85.735

Number of Fisher Scoring iterations: 9
```

Hide

```
#a) Calculate the test error
test_error <- sum(LR_Prediction != Y.test) / length(Y.test)
cat("Test error:", test_error, "\n")
```

```
Test error: 0.9562044 
```

Hide

```
#Test Accuracy
mean(LR_Prediction==Y.test)
```

```
[1] 0.04379562
```

Hide

```
# Confusion matrix for Logistic Regression
table(LR_Prediction, Y.test)
```

```
             Y.test
LR_Prediction  0  1
            0  3 37
            1 94  3
```

Hide

```
#b)Significant predictors
# Clump Thickness, Marginal Adhesion, Bare Nuclei,Bland Chromatin,Normal Nucleoli and Mitoses
were found as the most significant predictors due to having a p-value<0.05 when assuming an o
verall significance test of 95%.
```

Hide

```
#5- KNN Model. 1st Test K=sqrt(683), 2nd Test K= 5
library(class)

# Build the KNN Classifier model
# Train and test KNN with K=sqrt(683)

knn.pred = knn(X.train, X.test, Y.train, k=sqrt(683))
mean(knn.pred==Y.test)
```

```
[1] 0.6423358
```

Hide

```
# KNN with K=15 (by trial 15 was found to be an optimal parameter)
knn.pred1 = knn(X.train, X.test, Y.train, k=15)
mean(knn.pred1==Y.test)
```

```
[1] 0.6861314
```

Hide

```
##Confusion matrices
#k=sqrt(683)
table(knn.pred, Y.test)
```

```
        Y.test
knn.pred  0  1
       0 80 32
       1 17  8
```

Hide

```
#k=5
table(knn.pred1, Y.test)
```

```
         Y.test
knn.pred1  0  1
        0 82 28
        1 15 12
```

Hide

```
# Calculate the test errors
test_error1 <- sum(knn.pred != Y.test) / length(Y.test)
test_error2 <- sum(knn.pred1 != Y.test) / length(Y.test)

cat("Test error1:", test_error1, "\n")
```

```
Test error1: 0.3576642
```

Hide

```
cat("Test error2:", test_error2, "\n")
```

```
Test error2: 0.3138686
```

Hide

```
#6- LDA (Linear Discriminant Analysis) Model Generation.
library(MASS)
lda.fit=lda(Class_Label~., training_data)
lda.fit
```

```
Call:
lda(Class_Label ~ ., data = training_data)

Prior probabilities of groups:
        0         1 
0.6355311 0.3644689 

Group means:
       ID C_Thickness   UCSize  UCShape M_Adhesion Single_ECSize Bare_Nuclei
0 1123930    2.979827 1.305476 1.449568   1.340058      2.118156    1.314121
1 1010872    7.261307 6.592965 6.653266   5.472362      5.241206    7.608040
  Bland_Chromatin Normal_Nucleoli  Mitoses
0        2.048991        1.242075 1.063401
1        5.964824        5.874372 2.381910

Coefficients of linear discriminants:
                          LD1
ID              -4.875836e-08
C_Thickness      1.882809e-01
UCSize           9.524711e-02
UCShape          1.296649e-01
M_Adhesion       4.621327e-02
Single_ECSize    4.288567e-02
Bare_Nuclei      2.734438e-01
Bland_Chromatin  9.124872e-02
Normal_Nucleoli  1.176545e-01
Mitoses          2.920406e-02
```

Hide

```
# Predict output for test set
lda.pred=predict(lda.fit, testing_data)

#a) Calculate performance metrics
lda.class=lda.pred$class

#Test error
test_error3 <- sum(lda.class != Y.test) / length(Y.test)
cat("Test error3:", test_error3, "\n")
```

```
Test error3: 0.05839416
```

Hide

```
#Confusion Matrix
table(lda.class, Y.test)
```

```
         Y.test
lda.class  0  1
        0 94  5
        1  3 35
```

Hide

```
#Test Accuracy
mean(lda.class==Y.test)
```

```
[1] 0.9416058
```

Hide

```
#b)Predictors with more weight on class
lda.coef <- coef(lda.fit)
lda.coef
```

```
                          LD1
ID              -4.875836e-08
C_Thickness      1.882809e-01
UCSize           9.524711e-02
UCShape          1.296649e-01
M_Adhesion       4.621327e-02
Single_ECSize    4.288567e-02
Bare_Nuclei      2.734438e-01
Bland_Chromatin  9.124872e-02
Normal_Nucleoli  1.176545e-01
Mitoses          2.920406e-02
```

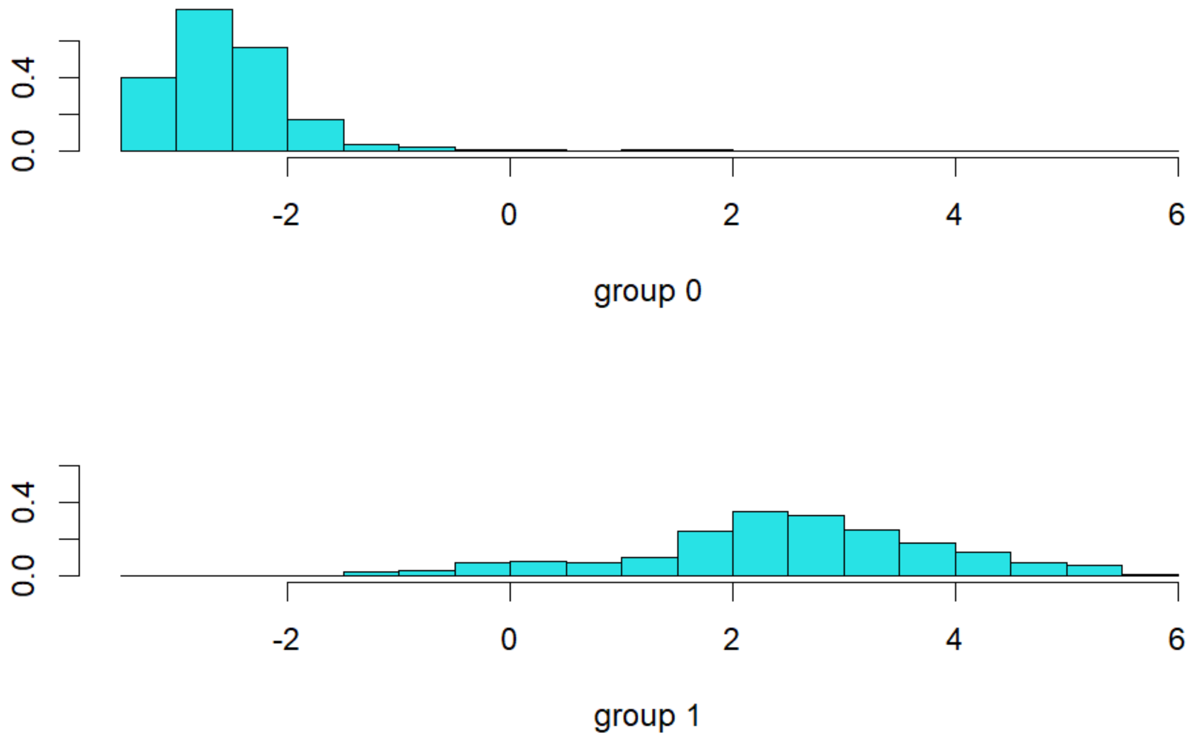
Hide

```
# Get the absolute values of the coefficients for each predictor variable
coef.abs <- abs(lda.coef[,1])

# Sort the coefficients in descending order
coef.sorted <- sort(coef.abs, decreasing = TRUE)

#Ranked list of predictor variables by strength of association
names(coef.sorted)
```

```
 [1] "Bare_Nuclei"     "C_Thickness"     "UCShape"         "Normal_Nucleoli"
 [5] "UCSize"          "Bland_Chromatin" "M_Adhesion"      "Single_ECSize"  
 [9] "Mitoses"         "ID"
```

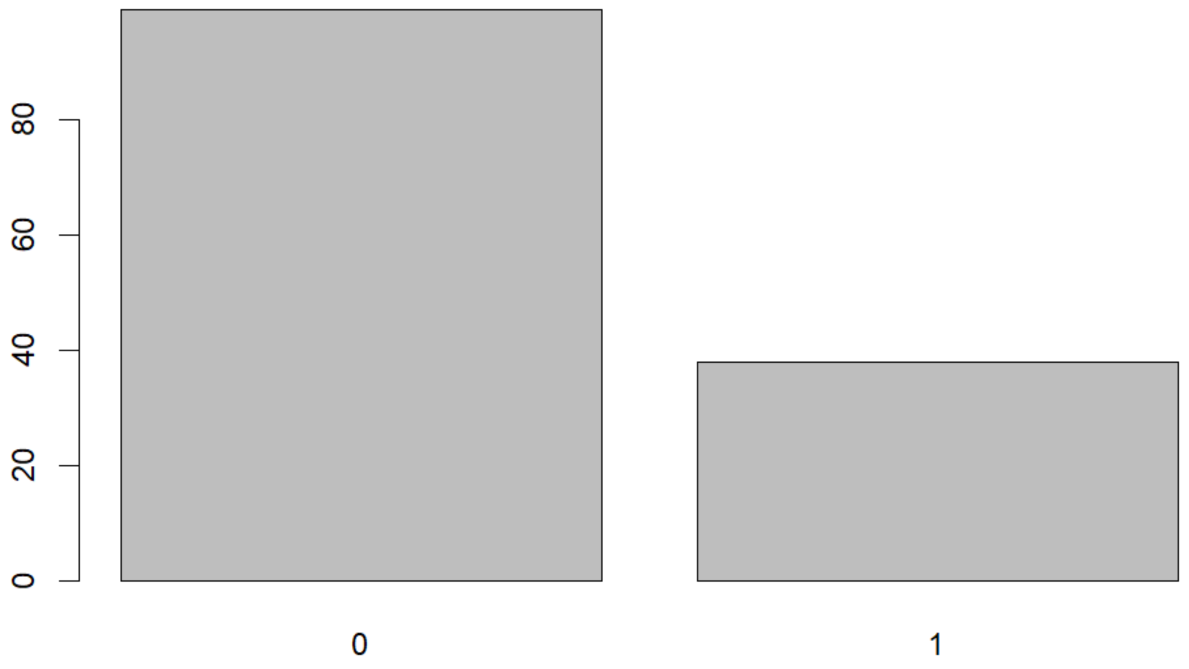
Hide

```
#c)
# Plot the linear discriminants for the LDA Model
plot(lda.fit)
```



Hide

```
# Plot of the linear discriminants for the LDA predictions
plot(lda.class)
```



Hide

```
#7- QDA (Quadratic Discriminant Analysis) Model Generation
qda.fit=qda(Class_Label~., training_data)
# The ouput does not contain the coefficients
qda.fit
```

```
Call:
qda(Class_Label ~ ., data = training_data)

Prior probabilities of groups:
        0         1 
0.6355311 0.3644689 

Group means:
       ID C_Thickness   UCSize  UCShape M_Adhesion Single_ECSize Bare_Nuclei
0 1123930    2.979827 1.305476 1.449568   1.340058      2.118156    1.314121
1 1010872    7.261307 6.592965 6.653266   5.472362      5.241206    7.608040
  Bland_Chromatin Normal_Nucleoli  Mitoses
0        2.048991        1.242075 1.063401
1        5.964824        5.874372 2.381910
```

Hide

```
# Prediction
qda.class=predict(qda.fit, testing_data)$class

#a)Test error
test_error4 <- sum(lda.class != Y.test) / length(Y.test)
cat("Test error4:", test_error4, "\n")
```

```
Test error4: 0.05839416
```

Hide

```
#Confusion Matrix
table(qda.class, Y.test)
```

```
         Y.test
qda.class  0  1
        0 93  2
        1  4 38
```

Hide

```
#Test Accuracy
mean(qda.class==Y.test)
```

```
[1] 0.9562044
```

Hide

```
#8-Comments on generated models
#Starting on the Logistic Regression Model, for this application the logistic regression misc
lassified most of the data which generated an error of 95% percent and a test accuracy of abo
ut 5%. The confusion matrix shows evidence of the misclassification

# Confusion matrix for Logistic Regression
table(LR_Prediction, Y.test)
```

```
             Y.test
LR_Prediction  0  1
            0  3 37
            1 94  3
```

Hide

```
#Moving on to the KNN Classifier, the first parameter(K) used was sqrt of the sample size whi
ch gave an error of about 36% which looked way better when comparing with to the logistic reg
ression, and that was also improved when I found that the optimal parameter for k was 15 by t
rial which reduced the error to about 31%. The confusion matrix shows evidence of how the cla
ssification was made.

#Confusion matrix for KNN Classifier for k=sqrt(n) and k=5
table(knn.pred, Y.test)
```

```
        Y.test
knn.pred  0  1
       0 80 32
       1 17  8
```

Hide

```
table(knn.pred1, Y.test)
```

```
         Y.test
knn.pred1  0  1
        0 82 28
        1 15 12
```

Hide

```
#Finally, the lda and the qda on the other hand gave more accurate results and much reduced t
est error. The test error for the LDA was about 6% whereas the test error for the QDA is also
about the same meaning for this particular application LDA and QDA classified the data better
than the KNN classifier and the Logistic Regression. The confusion matrices show evidence of
how the classifications were made.

#LDA Confusion Matrix
table(lda.class, Y.test)
```

```
         Y.test
lda.class  0  1
        0 94  5
        1  3 35
```

Hide

```
#QDA Confusion Matrix
table(qda.class, Y.test)
```

```
         Y.test
qda.class  0  1
        0 93  2
        1  4 38
```