Analyse des données “R” Diabetes dataset

INTRODUCTION :

Selon la page officielle de notre dataset “**Diabetes.csv**”, toutes les observations sont prises auprès de femmes d’au moins 21 ans d’origine indienne Pima.

Grossesses : Nombre de grossesses

Glucose: Concentration plasmatique de glucose à 2 heures dans un test de tolérance au glucose par voie orale

Pression artérielle: Pression artérielle diastolique (mm Hg)

Skin Thickness: Triceps skinfold épaisseur (mm)

Insuline : insuline sérique 2 heures (mu U/ml)

IMC: Indice de masse corporelle (poids en kg / (taille en m) ^ 2)

Diabète Pedigree Function: Fonction pedigree du diabète

Âge : Âge (ans)

Outcome : Variable de classe (0 ou 1)

L’objectif de l’ensemble de données est de prédire de manière diagnostique si un patient est diabétique,basé sur ces mesures diagnostiques.

1 .La description du tableau de données:

```
> library(MASS)
> library(ggplot2)
> data=read.csv("diabetes.csv")
> dim(data)
[1] 768   9
> str(data)
'data.frame':   768 obs. of  9 variables:
 $ Pregnancies              : int  6 1 8 1 0 5 3 10 2 8 ...
 $ Glucose                  : int  148 85 183 89 137 116 78 115 197 125 ...
 $ BloodPressure            : int  72 66 64 66 40 74 50 0 70 96 ...
 $ SkinThickness            : int  35 29 0 23 35 0 32 0 45 0 ...
 $ Insulin                  : int  0 0 0 94 168 0 88 0 543 0 ...
 $ BMI                      : num  33.6 26.6 23.3 28.1 43.1 25.6 31 35.3 30.5 0 $
 $ DiabetesPedigreeFunction: num  0.627 0.351 0.672 0.167 2.288 ...
 $ Age                      : int  50 31 32 21 33 30 26 29 53 54 ...
 $ Outcome                  : int  1 0 1 0 1 0 1 0 1 1 ...
```

```
> summary(data)
  Pregnancies        Glucose      BloodPressure    SkinThickness
 Min.   : 0.000   Min.   :  0.0   Min.   :  0.00   Min.   : 0.00
 1st Qu.: 1.000   1st Qu.: 99.0   1st Qu.: 62.00   1st Qu.: 0.00
 Median : 3.000   Median :117.0   Median : 72.00   Median :23.00
 Mean   : 3.845   Mean   :120.9   Mean   : 69.11   Mean   :20.54
 3rd Qu.: 6.000   3rd Qu.:140.2   3rd Qu.: 80.00   3rd Qu.:32.00
 Max.   :17.000   Max.   :199.0   Max.   :122.00   Max.   :99.00
    Insulin           BMI        DiabetesPedigreeFunction      Age
 Min.   :  0.0   Min.   : 0.00   Min.   :0.0780           Min.   :21.00
 1st Qu.:  0.0   1st Qu.:27.30   1st Qu.:0.2437           1st Qu.:24.00
 Median : 30.5   Median :32.00   Median :0.3725           Median :29.00
 Mean   : 79.8   Mean   :31.99   Mean   :0.4719           Mean   :33.24
 3rd Qu.:127.2   3rd Qu.:36.60   3rd Qu.:0.6262           3rd Qu.:41.00
 Max.   :846.0   Max.   :67.10   Max.   :2.4200           Max.   :81.00
    Outcome
 Min.   :0.000
 1st Qu.:0.000
 Median :0.000
 Mean   :0.349
 3rd Qu.:1.000
 Max.   :1.000
```

-Nombre de valeur unique et valeur manquante “ NA” dans chaque colonne :

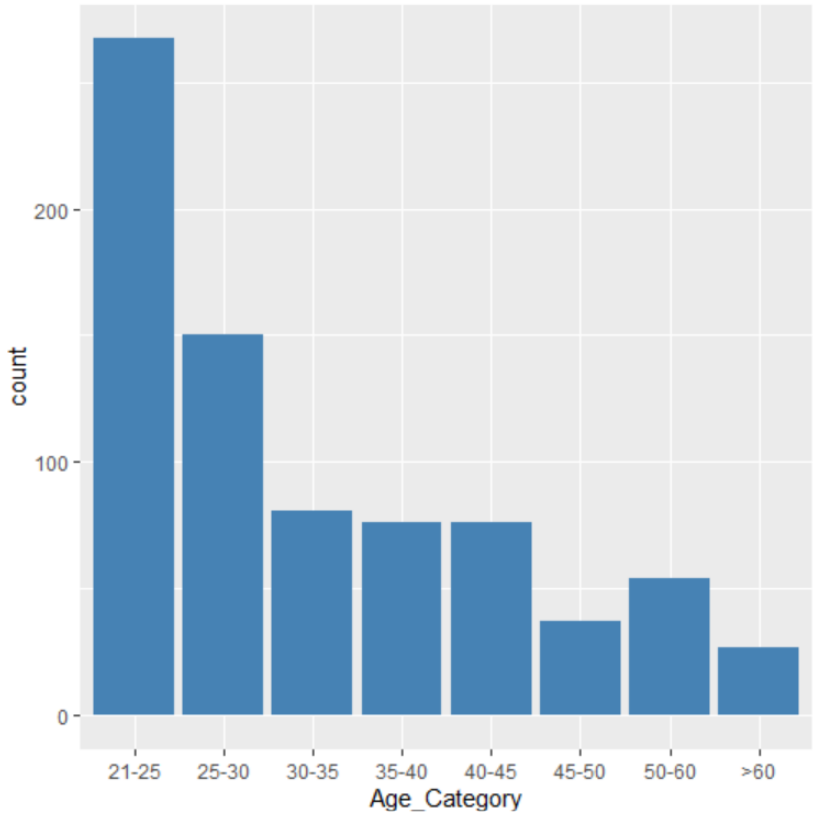
```
> UniqueValue = function (x) {length(unique(x)) }
> apply(data, 2 ,UniqueValue)
             Pregnancies                  Glucose            BloodPressure
                      17                      136                       47
           SkinThickness                  Insulin                      BMI
                      51                      186                      248
DiabetesPedigreeFunction                      Age                  Outcome
                     517                       52                        2
> NaValue = function (x) {sum(is.na(x)) }
> apply(data, 2, NaValue)
             Pregnancies                  Glucose            BloodPressure
                       0                        0                        0
           SkinThickness                  Insulin                      BMI
                       0                        0                        0
DiabetesPedigreeFunction                      Age                  Outcome
                       0                        0                        0
```

2.Exploratory data analysis & Graphical representations:

2.1 -Analyse du groupe d'âge

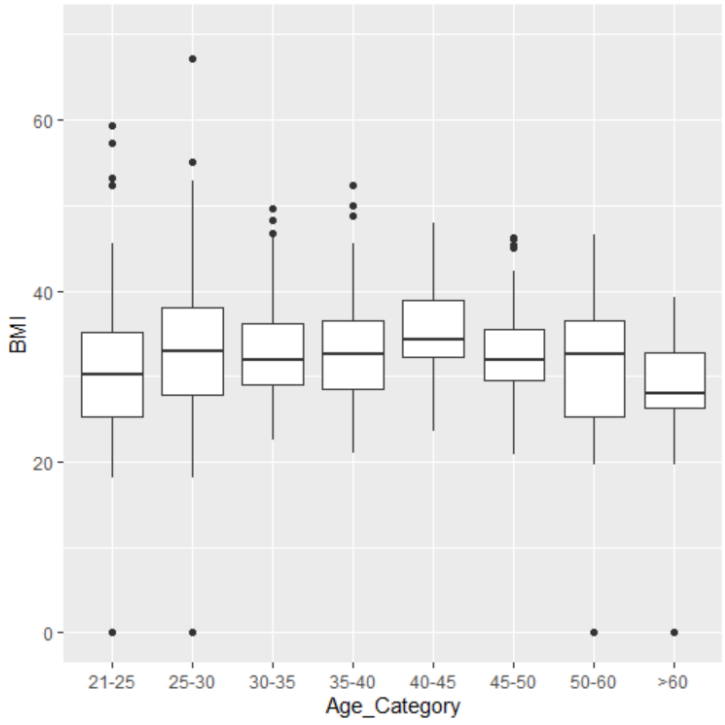
```
> diabetes=data
> diabetes$Age_Category <- ifelse(diabetes$Age < 21, "<21",
+                                 ifelse((diabetes$Age>=21) & (diabetes$Age<=25), "21-25",
+                                        ifelse((diabetes$Age>25) & (diabetes$Age<=30), "25-30",
+                                               ifelse((diabetes$Age>30) & (diabetes$Age<=35), "30-35",
+                                                      ifelse((diabetes$Age>35) & (diabetes$Age<=40), "35-40",
+                                                             ifelse((diabetes$Age>40) & (diabetes$Age<=45), "40-45",
+                                                                    ifelse((diabetes$Age>45) & (diabetes$Age<=50), "45-50",
+                                                                           ifelse((diabetes$Age>50) & (diabetes$Age<=60), "50-60",">60")))))))$
> diabetes$Age_Category <- factor(diabetes$Age_Category, levels = c('<21','21-25','25-30','30-35','35-40','40-45','45-50','50-60','>60'))
> table(diabetes$Age_Category)

  <21 21-25 25-30 30-35 35-40 40-45 45-50 50-60   >60
    0   267   150    81    76    76    37    54    27
> ggplot(aes(x = Age_Category), data = diabetes) + geom_bar(fill='steelblue')
```

Barplot des catégories d'âge:



2.2 Analyse d'Âge Category vs BMI et la création de box plot:

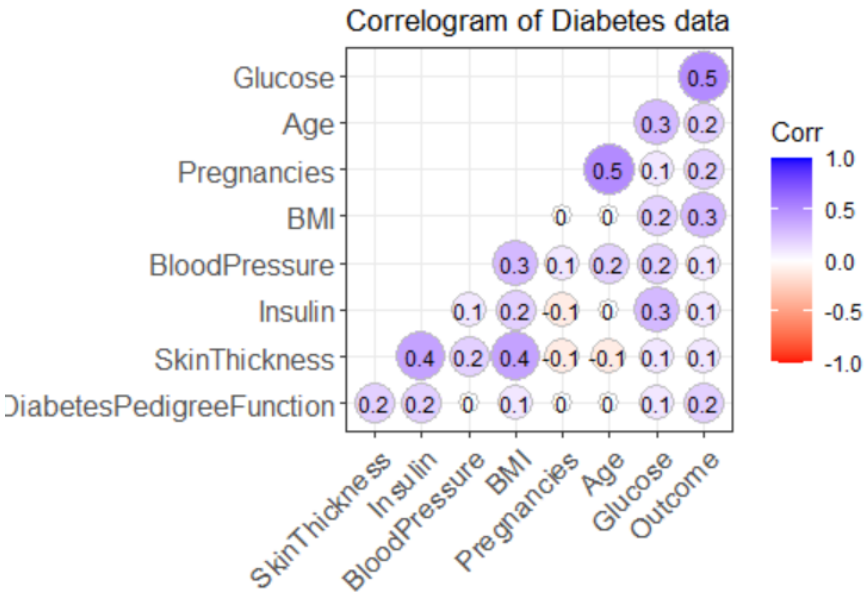
```
> by(diabetes$BMI, diabetes$Age_Category, summary)
diabetes$Age_Category: <21
NULL
------------------------------------------------------------
diabetes$Age_Category: 21-25
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
   0.00   25.25   30.20   30.36   35.20   59.40
------------------------------------------------------------
diabetes$Age_Category: 25-30
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
   0.00   27.80   33.00   33.04   38.10   67.10
------------------------------------------------------------
diabetes$Age_Category: 30-35
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  22.50   29.00   32.00   32.81   36.10   49.70
------------------------------------------------------------
diabetes$Age_Category: 35-40
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  21.00   28.48   32.60   32.97   36.58   52.30
------------------------------------------------------------
diabetes$Age_Category: 40-45
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  23.60   32.30   34.35   35.30   38.92   47.90
------------------------------------------------------------
diabetes$Age_Category: 45-50
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  20.80   29.50   32.00   32.86   35.50   46.20
------------------------------------------------------------
diabetes$Age_Category: 50-60
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
   0.00   25.27   32.65   31.11   36.52   46.50
--+---------------------------------------------------------
```



2.3 Trouver une corrélation entre différents champs:

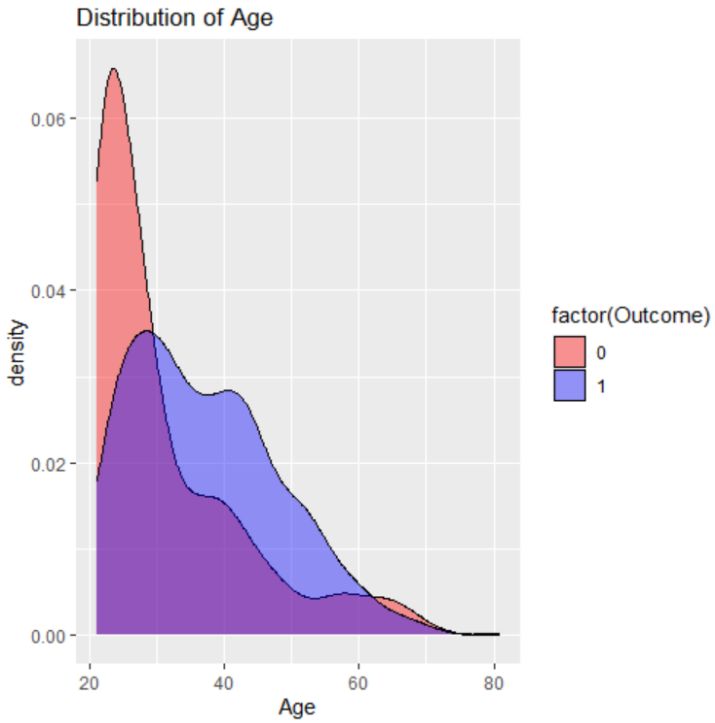
```
library(ggcorrplot)
library(corrplot)
corr<-round(cor(data),1)
corr

ggcorrplot(corr, hc.order = TRUE,
           type = "lower",
           lab = TRUE,
           lab_size = 3,
           method="circle",
           colors = c("red", "white", "blue"),
           title="Correlogram of Diabetes data",
           ggtheme=theme_bw)
```

```
> corr
                          Pregnancies Glucose BloodPressure SkinThickness
Pregnancies                       1.0     0.1           0.1          -0.1
Glucose                           0.1     1.0           0.2           0.1
BloodPressure                     0.1     0.2           1.0           0.2
SkinThickness                    -0.1     0.1           0.2           1.0
Insulin                          -0.1     0.3           0.1           0.4
BMI                               0.0     0.2           0.3           0.4
DiabetesPedigreeFunction          0.0     0.1           0.0           0.2
Age                               0.5     0.3           0.2          -0.1
Outcome                           0.2     0.5           0.1           0.1
                          Insulin BMI DiabetesPedigreeFunction  Age
Pregnancies                  -0.1 0.0                      0.0  0.5
Glucose                       0.3 0.2                      0.1  0.3
BloodPressure                 0.1 0.3                      0.0  0.2
SkinThickness                 0.4 0.4                      0.2 -0.1
Insulin                       1.0 0.2                      0.2  0.0
BMI                           0.2 1.0                      0.1  0.0
DiabetesPedigreeFunction      0.2 0.1                      1.0  0.0
Age                           0.0 0.0                      0.0  1.0
Outcome                       0.1 0.3                      0.2  0.2
                          Outcome
Pregnancies                   0.2
Glucose                       0.5
BloodPressure                 0.1
SkinThickness                 0.1
Insulin                       0.1
BMI                           0.3
DiabetesPedigreeFunction      0.2
Age                           0.2
Outcome                       1.0
>
```



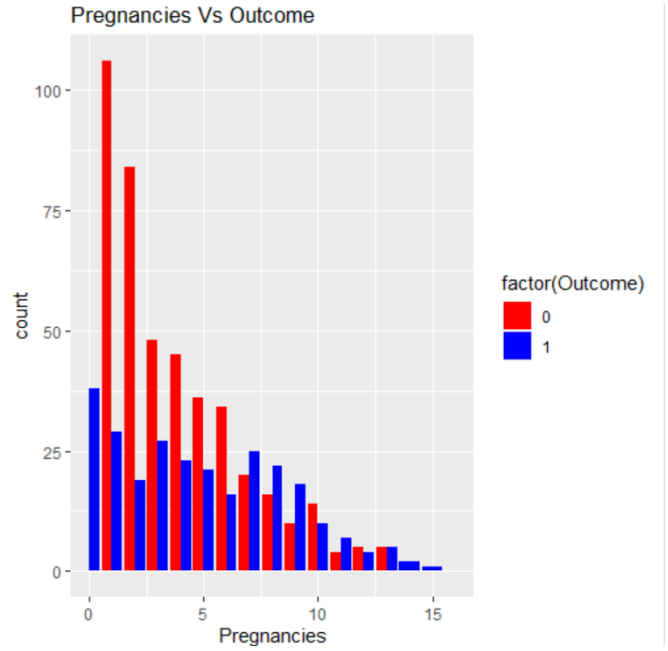
2.4 Check quel est l'impact de l'âge sur le Outcome:

```
##Check what is the impact of age over the Outcome
ggplot(data,aes(x=Age,fill=factor(Outcome)))+geom_density(alpha=0.4)+scale_fill_manual(values=c("red", "blue"))+labs(title="Distribution of Ag
```



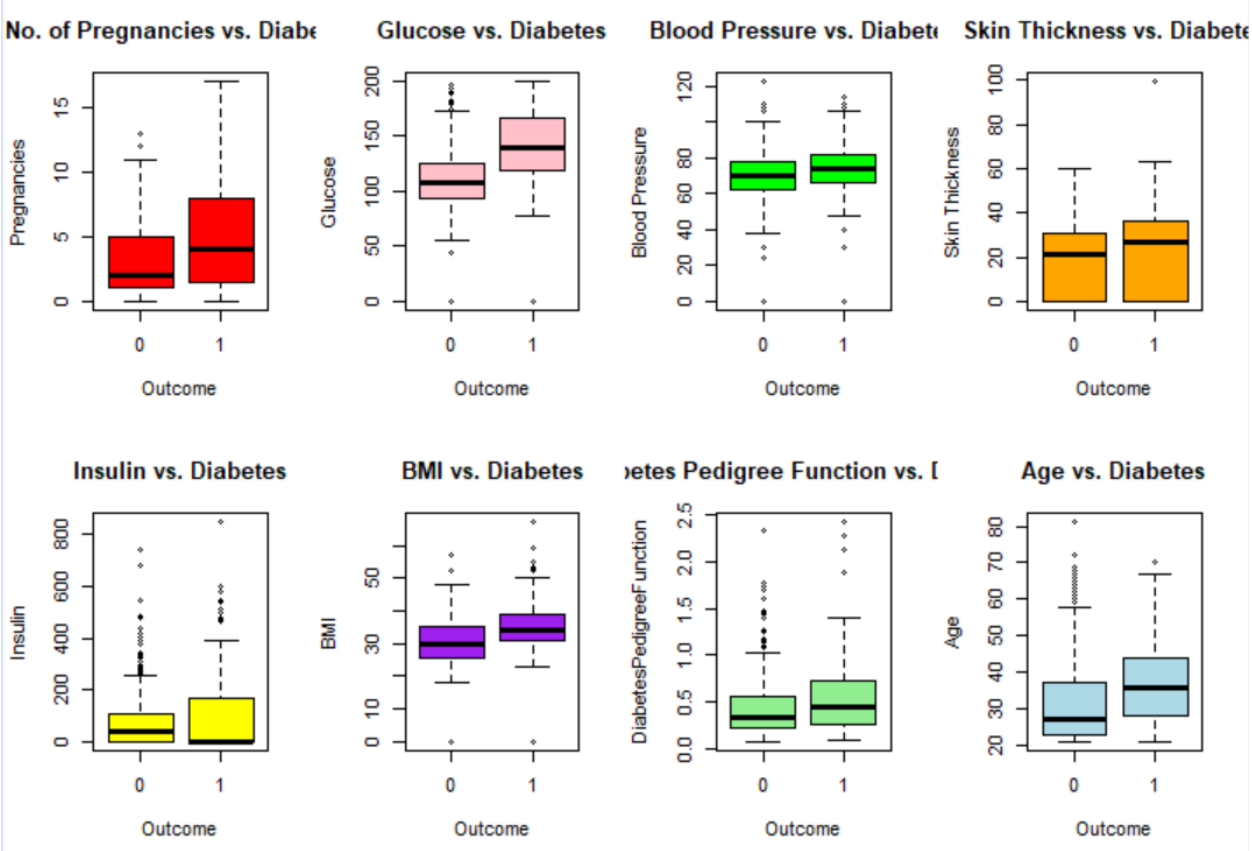
2.5 Number des grossesses a t-il un impact sur l'Outcome:

```
#Number of Pregnancies has an impact over diabetes outcome
ggplot(data,aes(x=Pregnancies,fill=factor(Outcome)))+geom_bar(position="Dodge")+scale_fill_manual(values=c("red","blue"))+scale_x_continuous(limits=c(0,16)
```



2.6 Corrélation entre les variables numériques et l'Outcome:

```
> par(mfrow=c(2,4))
> boxplot(Pregnancies~Outcome, main="No. of Pregnancies vs. Diabetes",
+         xlab="Outcome", ylab="Pregnancies",col="red")
> boxplot(Glucose~Outcome, main="Glucose vs. Diabetes",
+         xlab="Outcome", ylab="Glucose",col="pink")
> boxplot(BloodPressure~Outcome, main="Blood Pressure vs. Diabetes",
+         xlab="Outcome", ylab="Blood Pressure",col="green")
> boxplot(SkinThickness~Outcome, main="Skin Thickness vs. Diabetes",
+         xlab="Outcome", ylab="Skin Thickness",col="orange")
> boxplot(Insulin~Outcome, main="Insulin vs. Diabetes",
+         xlab="Outcome", ylab="Insulin",col="yellow")
> boxplot(BMI~Outcome, main="BMI vs. Diabetes",
+         xlab="Outcome", ylab="BMI",col="purple")
> boxplot(DiabetesPedigreeFunction~Outcome, main="Diabetes Pedigree Function vs. Diabetes", xlab="Outcome",
> boxplot(Age~Outcome, main="Age vs. Diabetes",
+         xlab="Outcome", ylab="Age",col="lightblue")
> box(which = "outer", lty = "solid"
```



3. Diviser le tableau de données en apprentissage/test (70/30%):

```
> dt = sort(sample(nrow(data),nrow(data)*0.7))
> train = data[dt,]
> test = data[-dt,]
> dim(train)
[1] 537   9
> dim(test)
[1] 231   9
```

Scale te data: L'une des hypothèses clés de l'analyse discriminante linéaire est que chacune des variables prédictives a la même variance.

Un moyen facile de s'assurer que cette hypothèse est respectée est de mettre à l'échelle Chaque variable telle qu'elle aura une moyenne de 0 et un écart-type de 1.

```
> data[1:8] <- scale(data[1:8])
> #find mean of each predictor variable
> apply(data, 2, mean)
              Pregnancies                   Glucose
            -6.901102e-17             -3.640265e-18
             BloodPressure             SkinThickness
             1.177826e-17              4.668542e-17
                  Insulin                       BMI
            -4.414552e-17             -1.971323e-16
DiabetesPedigreeFunction                       Age
             6.894834e-17              1.987660e-16
                  Outcome
             3.489583e-01
> apply(data, 2, sd)
              Pregnancies                   Glucose
                1.0000000                 1.0000000
             BloodPressure             SkinThickness
                1.0000000                 1.0000000
                  Insulin                       BMI
                1.0000000                 1.0000000
DiabetesPedigreeFunction                       Age
                1.0000000                 1.0000000
                  Outcome
                0.4769514
```

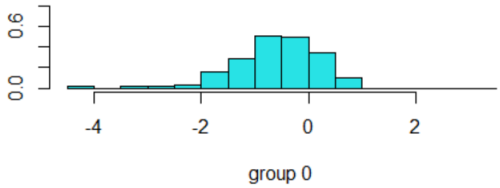
4. Application de l'analyse discriminante sur l'échantillon d'apprentissage:

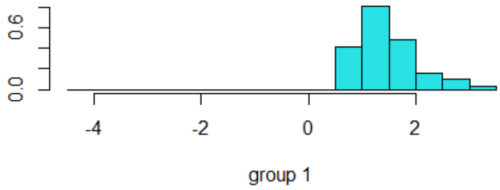
4.1 LDA:

```
> model <- lda(Outcome ~Pregnancies+Glucose+BloodPressure+SkinThickness+Insulin +BMI +DiabetesPedigreeFunction +Age, data=train)
> model
Call:
lda(Outcome ~ Pregnancies + Glucose + BloodPressure + SkinThickness +
    Insulin + BMI + DiabetesPedigreeFunction + Age, data = train)

Prior probabilities of groups:
        0         1
0.6629423 0.3370577

Group means:
  Pregnancies  Glucose BloodPressure SkinThickness  Insulin      BMI
0    3.328652 110.5197      69.17135      19.71348 69.57865 30.43427
1    4.911602 142.1547      70.40331      21.76796 93.65193 35.44144
  DiabetesPedigreeFunction      Age
0                0.4391067 31.68539
1                0.5337624 37.58011

Coefficients of linear discriminants:
                                LD1
Pregnancies               0.0898958366
Glucose                   0.0278490300
BloodPressure            -0.0103578883
SkinThickness            -0.0018686231
Insulin                  -0.0009667256
BMI                       0.0660060672
DiabetesPedigreeFunction  0.6405674502
Age                       0.0084916438
```

5. Prédiction des classes pour l'échantillon test et Évaluation:

```
> pred = predict(model,newdata = test)# prediction des données test
> conf =table(test$Outcome,pred$class)
> conf

     0   1
  0 131  13
  1  38  49
> diag(conf)
  0   1
131  49
> ldahist(pred$x[,1], g= pred$class)
>
> acc = (sum(diag(conf)))/sum(conf)*100
> print(paste(acc,"%"))
[1] "77.9220779220779 %"
```





4.2- QDA:

```
> model2 <- qda(Outcome ~., data=train)
> model <- lda(Outcome ~Pregnancies+Glucose+BloodPressure+SkinThickness+Insulin +BMI +DiabetesPedigreeFunction +Age, data=train)
> model2
Call:
qda(Outcome ~ ., data = train)

Prior probabilities of groups:
        0         1
0.6629423 0.3370577

Group means:
  Pregnancies  Glucose BloodPressure SkinThickness  Insulin      BMI
0    3.328652 110.5197      69.17135      19.71348 69.57865 30.43427
1    4.911602 142.1547      70.40331      21.76796 93.65193 35.44144
  DiabetesPedigreeFunction      Age
0                0.4391067 31.68539
1                0.5337624 37.58011
```

```
> conf2

     0   1
  0 128  16
  1  44  43
> diag(conf2)
  0   1
128  43
```

```
> print(paste(acc2,"%"))
[1] "74.025974025974 %"
```

Distribution:

```
> table(pred$class)

  0   1
169  62
> table(train[9])
Outcome
  0   1
356 181
> table(test[9])
Outcome
  0   1
144  87
>
```

5-The prior probabilities used: the prior probabilities of class membership. If unspecified, the class proportions for the training set are used. If present, the probabilities should be specified in the order of the factor levels:

```
> pred3 = predict(model2,newdata = test ,prior =c(1,1)/2)# prediction des données test
> conf3 =table(test$Outcome,pred$class)# matrice de confusion
> conf3

     0   1
  0 131  13
  1  38  49
> diag(conf3)
  0   1
131  49
> acc = (sum(diag(conf3)))/sum(conf3)*100 # accurate
> print(paste(acc,"%"))
[1] "77.9220779220779 %"
```