

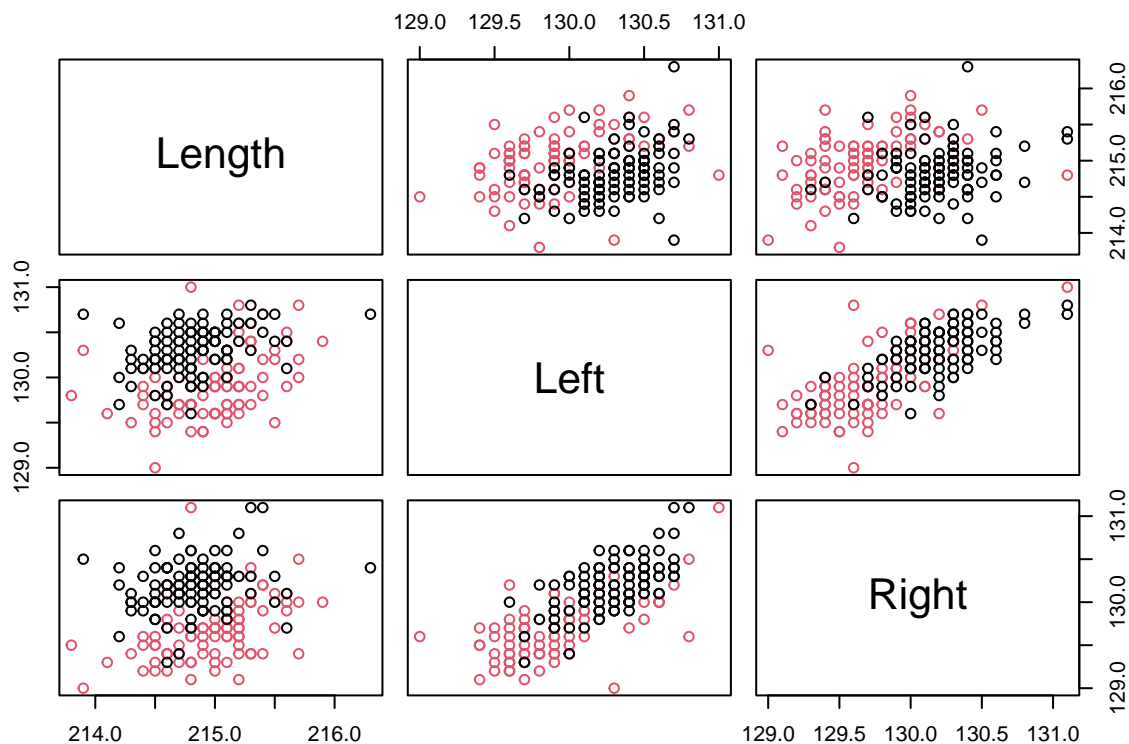
Projet_2

2023-10-16

```
rm(list=ls())
library(Rmixmod)

## Loading required package: Rcpp
## Rmixmod v. 2.1.9 / URI: www.mixmod.org

banknote <- read.csv("~/Documents/Master Iref/M2/Scoring /banknote.csv")
data = as.data.frame(banknote[,2:4])
note = as.factor(banknote[,1])
new = data.frame(Length=214.90, Left=130.12, Right=129.96)
plot(data, col = note)
```



Exercice 1

```
#hétérocédastique
learnht = mixmodLearn(data,knownLabels=note,
                      models=mixmodGaussianModel(listModels = c('Gaussian_pk_Lk_Ck')),
                      criterion="BIC") #calcul du BIC cas hétérocédastique

predictionht = mixmodPredict(data=new,classificationRule=learnht['bestResult'])
```

```
#prediction
print(predictionht[6])
```

```
##           [,1]      [,2]
## [1,] 0.6027744 0.3972256
```

```
print(predictionht[5])
```

```
## [1] 1
```

D'après le classifieur Gaussien hétéroscédastique nous affectons le nouveau billet à la classe 1 qui correspond à la classe counterfeit (60% > 40%).

```
#homoscédastique
learnhm = mixmodLearn(data,knownLabels=note,
                      models=mixmodGaussianModel(listModels = c('Gaussian_pk_L_C')),
                      criterion="BIC")
predictionhm = mixmodPredict(data=new,classificationRule=learnhm['bestResult'])
#prediction
print(predictionhm[6])
```

```
##           [,1]      [,2]
## [1,] 0.4998558 0.5001442
```

```
print(predictionhm[5])
```

```
## [1] 2
```

D'après le classifieur Gaussien homoscédastique nous affectons le nouveau billet à la classe 2 qui correspond à la classe genuine (50,01% > 49,9%).

Régression logistique:

```
attach(banknote)
rule=glm(as.factor(Status)~Length+Left+Right,family=binomial(link='logit'))
rule$coefficients
```

```
## (Intercept)      Length      Left      Right
## 157.918650    2.966799   -2.000533   -4.117872
```

```
score <- predict(rule,new=new)
print(score)
```

```
##           1
## 0.01562366
```

```
prob <- exp(score)/(1+exp(score))
print(prob)
```

```
##           1
## 0.5039058
```

```
#predict(rule,new=data.frame(Length=Length,Left=Left,Right=Right))
```

Le score est positif avec la classe 1, de plus la probabilité d'appartenir à la classe 1 est supérieur à 50% qui correspond à la classe genuine.

Exercice 2

Erreur du modèle homoscédastique:

```

predict = mixmodPredict(data=data,classificationRule=learnhm['bestResult'])
erght = 1-sum(predict[5]==as.numeric(note))/length(note) # taux d'erreur sur les données de t
cat('Erreur apparente par resubstitution:',erght,'\n')

```

Erreur apparente par resubstitution: 0.18

Erreur du modèle hétéroscédastique:

```

predict = mixmodPredict(data=data,classificationRule=learnht['bestResult'])
erght = 1-sum(predict[5]==as.numeric(note))/length(note) # taux d'erreur sur les données de t
cat('Erreur apparente par resubstitution:',erght,'\n')

```

Erreur apparente par resubstitution: 0.155

Erreur du modèle de régression logistique:

```

predicted <- predict(rule, type = "response")

# Convertir les prédictions en classes (genuine ou counterfeit) en fonction d'un seuil (par exemple, 0.5)
predicted_class <- ifelse(predicted >= 0.5, "genuine", "counterfeit")

# Calculer l'erreur apparente par resubstitution
misclassification_error <- mean(banknote$Status != predicted_class)
cat("Erreur apparente par resubstitution:", misclassification_error, "\n")

```

Erreur apparente par resubstitution: 0.18

Exercice 3

Séparation des données en données de test et données d'entraînement:

```

library(caTools)
set.seed(123)

sample = sample.split(banknote$Status, SplitRatio = 0.8)

train = banknote[sample, ]
test = banknote[!sample, ]

X_train = train[,2:4]
X_test = test[,2:4]

y_train = as.factor(train$Status)
y_test = as.factor(test$Status)

```

Erreur par validation croisée du modèle homoscédastique:

```

learnhm = mixmodLearn(X_train,knownLabels=y_train,
                      models=mixmodGaussianModel(listModels = c('Gaussian_pk_L_C')),
                      criterion="BIC")

predict = mixmodPredict(data=X_test,classificationRule=learnhm['bestResult'])
erght = 1-sum(predict[5]==as.numeric(y_test))/length(y_test) # taux d'erreur sur les données de t
cat('Erreur par validation croisée:',erght,'\n')

```

Erreur par validation croisée: 0.15

Erreur par validation croisée du modèle hétéroscédastique:

```

learnht = mixmodLearn(X_train,knownLabels=y_train,
                      models=mixmodGaussianModel(listModels = c('Gaussian_pk_Lk_Ck')),
                      criterion="BIC")

predict = mixmodPredict(data=X_test,classificationRule=learnht['bestResult'])
erght = 1-sum(predict[5]==as.numeric(y_test))/length(y_test) # taux d'erreur sur les données de t
cat('Erreur par validation croisée:',erght,'\n')

## Erreur par validation croisée: 0.125

library(boot)
attach(banknote)
formula <- as.formula("as.factor(Status) ~ Length + Right + Left")

# Création du modèle de régression logistique
model <- glm(formula, data = train, family = binomial)

predicted <- predict(model, newdata = X_test ,type = "response")

# Convertir les prédictions en classes (genuine ou counterfeit) en fonction d'un seuil (par exemple, 0,
predicted_class <- ifelse(predicted >= 0.5, "genuine", "counterfeit")

# Calculer l'erreur apparente par resubstitution
misclassification_error <- mean(y_test != predicted_class)
cat("Erreur par validation croisée:", misclassification_error, "\n")

## Erreur par validation croisée: 0.15

```

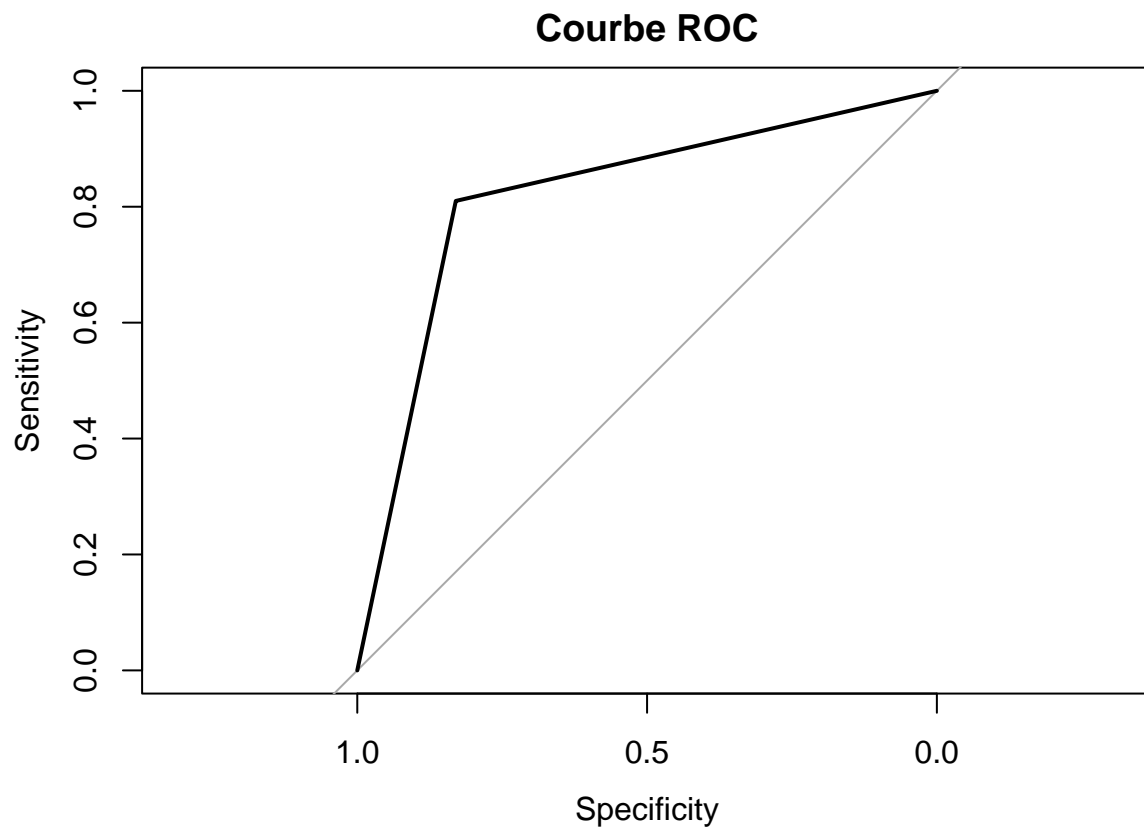
Exercice 4

```

library(pROC)
predict = mixmodPredict(data=data,classificationRule=learnhm['bestResult'])
scores = predict[5]
labels = ifelse(banknote$Status=='counterfeit', 2, 1)

roc_obj = roc(labels, scores)
plot(roc_obj, main = "Courbe ROC")

```



```
auc_value = auc(roc_obj)
cat("Aire sous la courbe:", auc_value)
```

```
## Aire sous la courbe: 0.82
```