Rendu TP Noté

Birane BA

1/07/2022

Voici le lien de mon répertoire GitHub pour avoir accès au code : https://github.com/birane906/R-gression-Scoring

```
library(ggplot2)
```

Introduction

Le sujet à traiter est Fish. Nous disposons d'un jeu de données contenant les mesures des caractéristiques physiques de différents poissons. Nous avons les variables suivantes : - Species : prend la valeur 1 si l'individu appartient à l'espèce à étudier 0 sinon - Weight : pour le poids de l'individu étudié - Height : pour la taille de l'individu étudié - Width : pour la largeur de l'individu étudié L'objectif est de prédire l'espèce d'un poisson donné en fonction de leur mensuration.

Lecture des données

Etant donné que notre dataset est sous la forme d'un fichier csv, j'ai utilisé la fonction read.csv() de R pour lire les données. J'ai précisé le séparateur adéquat pour mon jeu de données afin de l'ouvrir correctement.

```
Fish <- read.csv("/cloud/project/Fish.csv", sep=";")
```

Analyse des données

Résumé des données

J'ai tout d'abord affiché les premières valeurs de mon dataset grâce à la fonction head() pour vérifier si l'importation s'est bien passée.

```
head(Fish)
```

```
##   Species Weight  Height  Width
## 1       0    242 11.5200 4.0200
## 2       0    290 12.4800 4.3056
## 3       0    340 12.3778 4.6961
## 4       0    363 12.7300 4.4555
## 5       0    430 12.4440 5.1340
## 6       0    450 13.6024 4.9274
```

Par la suite, on a grâce aux fonctions str() et summary() le résumé statistique de l'ensemble de mes données.

```
str(Fish)
```

```
## 'data.frame':    111 obs. of  4 variables:
##  $ Species: int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Weight : num  242 290 340 363 430 450 500 390 450 500 ...
```

```
##  $ Height : num  11.5 12.5 12.4 12.7 12.4 ...
##  $ Width  : num  4.02 4.31 4.7 4.46 5.13 ...
```

```
summary(Fish)
```

```
##     Species           Weight           Height           Width
##  Min.   :0.0000   Min.   :   0.0   Min.   : 2.112   Min.   :1.408
##  1st Qu.:0.0000   1st Qu.: 137.5   1st Qu.: 6.192   1st Qu.:3.624
##  Median :1.0000   Median : 300.0   Median : 8.877   Median :4.566
##  Mean   :0.5045   Mean   : 415.0   Mean   : 9.960   Mean   :4.765
##  3rd Qu.:1.0000   3rd Qu.: 687.5   3rd Qu.:13.681   3rd Qu.:6.011
##  Max.   :1.0000   Max.   :1100.0   Max.   :18.957   Max.   :8.142
```

Nous remarquons qu'on a que des variables numériques. Comme dit dans l'énoncé de ce TP, la variable Species peut prendre que les valeurs 0 ou 1. Donc, nous allons par la suite représenter cette variable graphiquement pour mieux la comprendre.

Représentations graphiques des variables

Représentation graphiques de la variable à expliquer

Ce graphe ci-dessous représente la répartition des différentes espèces des poissons.

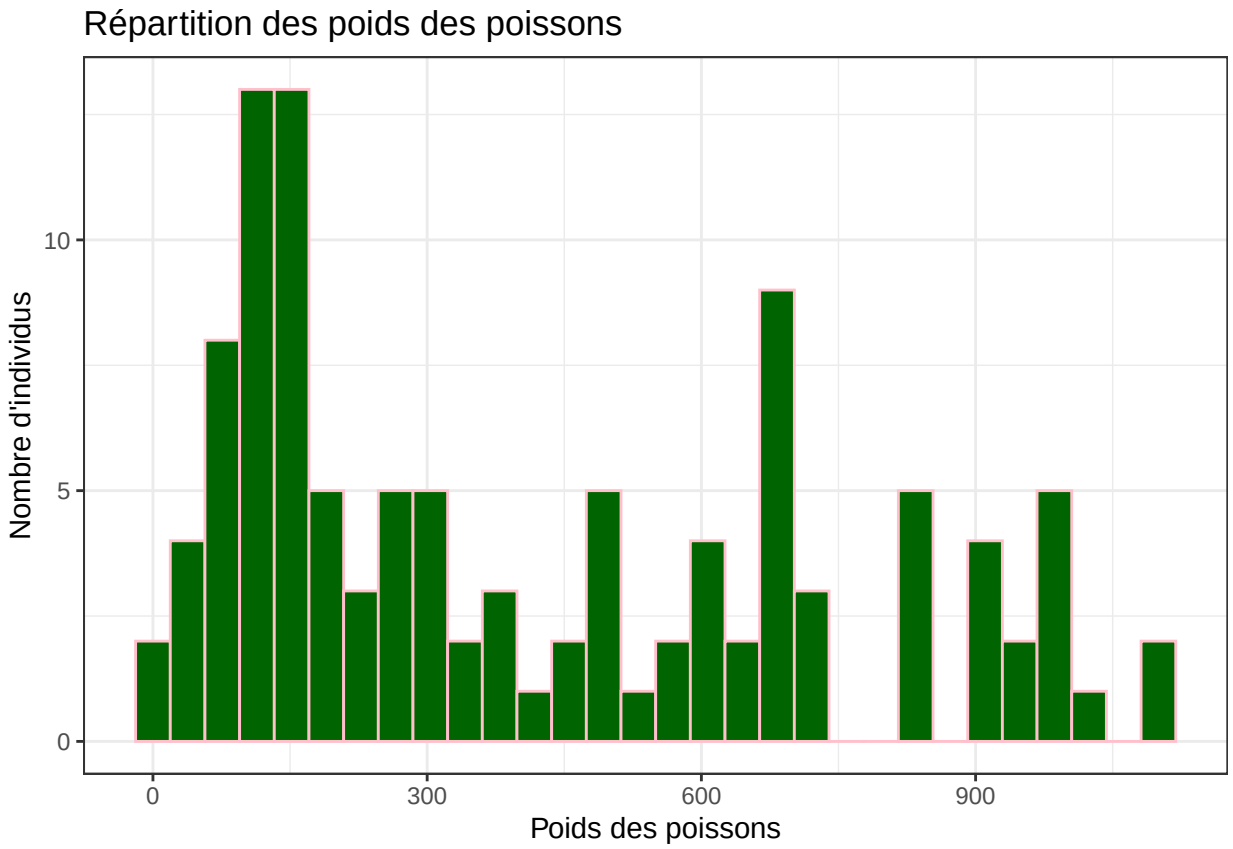
```
library(ggplot2)
ggplot(Fish) + geom_bar(aes(x = as.factor(Species))) + xlab("Espèce des poissons") + ylab("Nombre d'indi
```

Représentation graphiques des variables explicatives

```
ggplot(Fish, aes(x = Weight)) + geom_histogram(color="pink", fill="darkgreen") + xlab("Poids des poisso
```

Poids des poissons

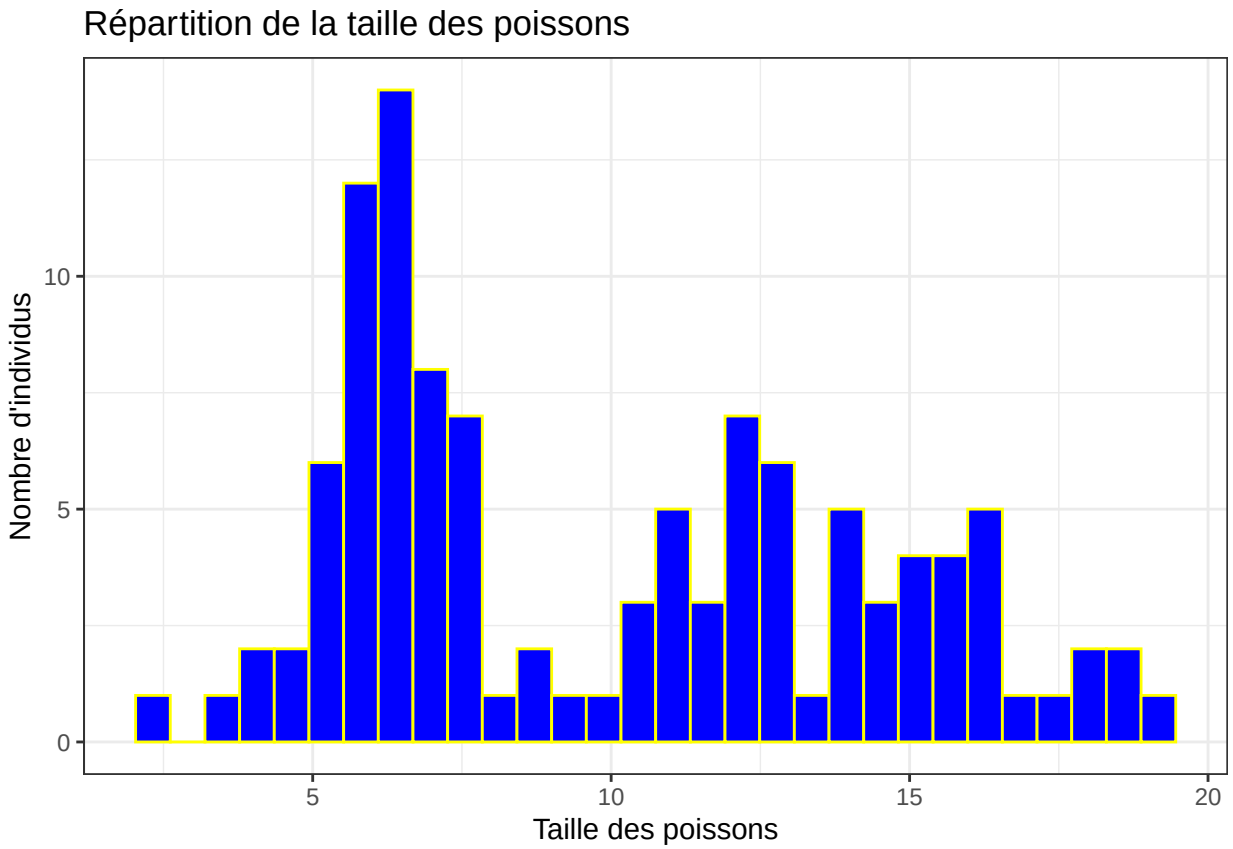
```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



Taille des poissons

```
ggplot(Fish, aes(x = Height)) + geom_histogram(color="yellow", fill="blue") + xlab("Taille des poissons
```

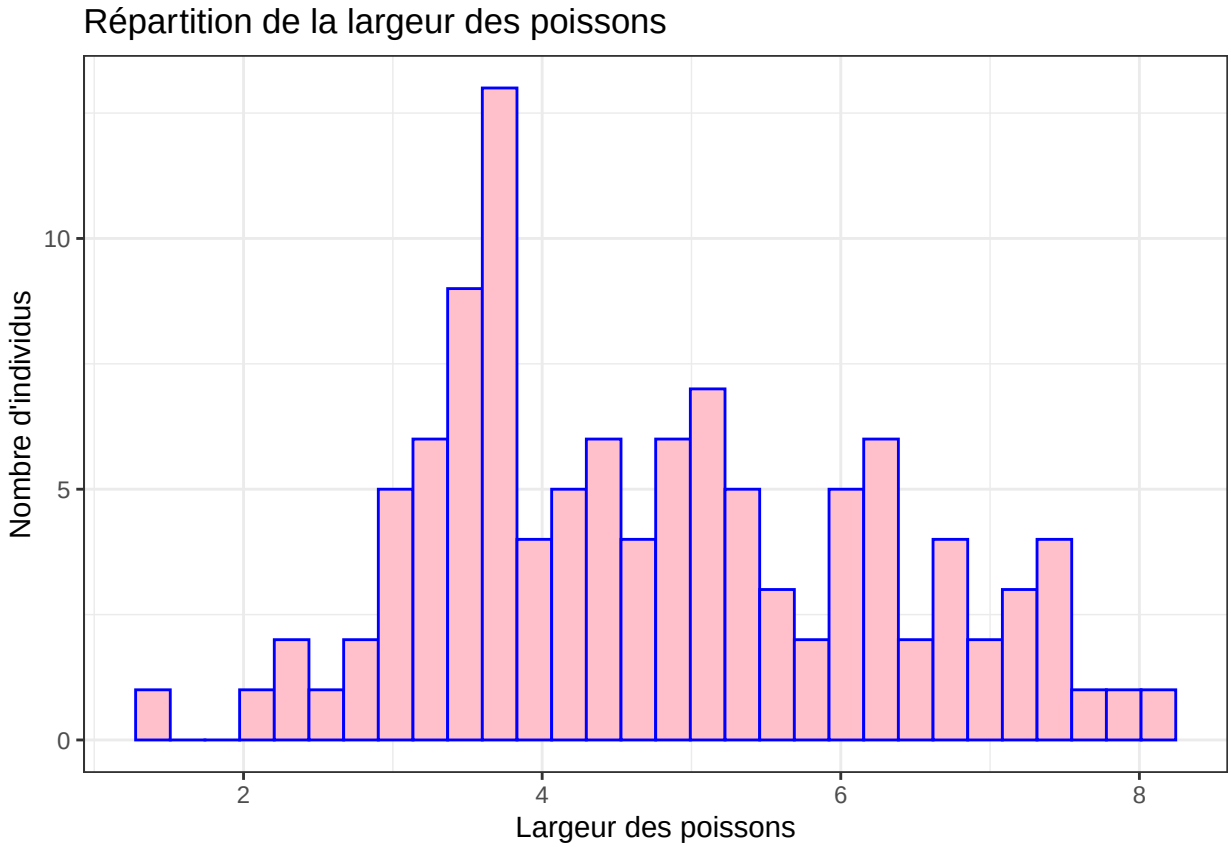
```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```




```
ggplot(Fish, aes(x = Width)) + geom_histogram(color="blue", fill="pink") + xlab("Largeur des poissons")
```

Largeur des poissons

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



Prédiction de l'espèce des poissons

Afin de mettre en place notre modèle de prédiction, nous allons tout d'abord séparer notre dataset en 2 parties : une pour l'apprentissage et l'autre pour le test.

Séparation des données

L'échantillon d'aprentissage contiendra 70% de nos données. Elle permettra d'apprendre les données. Celui de test contiendra les 30% restants et nous servira à tester les performances de prédictions de notre modèle.

```
# taille échantillon
n <- nrow(Fish)
# indices des individus dans l'échantillon d'apprentissage
train_index <- sample(x = 1:n, size = round(0.7 * n), replace = FALSE)
# train et test sets
train_data <- Fish[train_index,]
test_data <- Fish[-train_index,]
```

Apprentissage du modèle

On va prédire l'espèce d'un poisson à l'aide des mesures de ses caractéristiques physiques. Nous allons utiliser les sélections Forward et Backward pour voir la combinaison de quelles caractéristiques physiques nous donne plus d'informations.

Sélection Forward

Avec cette méthode de sélection, on part d'un modèle qui est vide. On ajoute des attributs au fur et à mesure un par un afin de terminer avec un modèle qui est complet.

```
# le modèle de base est le modèle nul (celui avec uniquement un intercept)
log_reg0 <- glm(Species ~ 1, data = train_data, family="binomial")
# la regression forward part du modèle nul et l'enrichit
forward_sel <- step(log_reg0, direction="forward",
                    scope=list(lower=log_reg0, upper=~Weight+Height+Width))
```

```
## Start:  AIC=109.67
## Species ~ 1
##
##          Df Deviance     AIC
## + Height  1   83.747  87.747
## + Weight  1  104.897 108.897
## <none>       107.669 109.669
## + Width   1  107.051 111.051
##
## Step:  AIC=87.75
## Species ~ Height
##
##          Df Deviance    AIC
## + Weight  1   34.199 40.199
## + Width   1   45.182 51.182
## <none>        83.747 87.747
##
## Step:  AIC=40.2
## Species ~ Height + Weight
##
##         Df Deviance    AIC
## <none>       34.199 40.199
## + Width  1   33.896 41.896
```

```
summary(forward_sel)
```

```
##
## Call:
## glm(formula = Species ~ Height + Weight, family = "binomial",
##     data = train_data)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -2.49553  -0.01727   0.00002   0.38017   1.76544
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) 20.22773    6.16650   3.280  0.00104 **
## Height      -4.44603    1.42876  -3.112  0.00186 **
## Weight       0.05765    0.02022   2.851  0.00436 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
```

```
##     Null deviance: 107.669  on 77  degrees of freedom
## Residual deviance:  34.199  on 75  degrees of freedom
## AIC: 40.199
##
## Number of Fisher Scoring iterations: 8
```

```
hat_pi <- predict(forward_sel, newdata = test_data, type = "response")
hat_y <- as.integer(hat_pi > 0.5)
```

Sélection Backward

Même principe que le Forward sauf qu'ici, on part du modèle complet et on diminue les variables une par une

```
log_reg1 <- glm(Species ~ ., data = train_data, family="binomial")
back_sel <- step(log_reg1, direction="backward")
```

```
## Start:  AIC=41.9
## Species ~ Weight + Height + Width
##
##          Df Deviance     AIC
## - Width   1   34.199  40.199
## <none>        33.896  41.896
## - Weight  1   45.182  51.182
## - Height  1   98.112 104.112
##
## Step:  AIC=40.2
## Species ~ Weight + Height
##
##          Df Deviance     AIC
## <none>        34.199  40.199
## - Weight  1   83.747  87.747
## - Height  1  104.897 108.897
```

```
summary(back_sel)
```

```
##
## Call:
## glm(formula = Species ~ Weight + Height, family = "binomial",
##     data = train_data)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -2.49553  -0.01727   0.00002   0.38017   1.76544
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) 20.22773    6.16650   3.280  0.00104 **
## Weight       0.05765    0.02022   2.851  0.00436 **
## Height      -4.44603    1.42876  -3.112  0.00186 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 107.669  on 77  degrees of freedom
## Residual deviance:  34.199  on 75  degrees of freedom
```

```
## AIC: 40.199
##
## Number of Fisher Scoring iterations: 8
```

```
hat_pi1 <- predict(back_sel, newdata = test_data, type = "response")
hat_y1 <- as.integer(hat_pi1 > 0.5)
```

Bilan sur les différentes méthodes de sélection

Nous remarquons que toutes les 2 méthodes de sélection donnent le même résultat : la combinaison des variables Height et Weight nous donne de meilleurs résultats.

Elaboration de la matrice de confusion

```
library(caret)
```

```
## Loading required package: lattice
```

```
confusionMatrix(data = as.factor(hat_y), reference = as.factor(test_data$Species), positive = "1")
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  0  1
##          0 18  2
##          1  1 12
##
##                Accuracy : 0.9091
##                  95% CI : (0.7567, 0.9808)
##     No Information Rate : 0.5758
##     P-Value [Acc > NIR] : 3.054e-05
##
##                   Kappa : 0.8121
##
##  Mcnemar's Test P-Value : 1
##
##             Sensitivity : 0.8571
##             Specificity : 0.9474
##          Pos Pred Value : 0.9231
##          Neg Pred Value : 0.9000
##              Prevalence : 0.4242
##          Detection Rate : 0.3636
##    Detection Prevalence : 0.3939
##       Balanced Accuracy : 0.9023
##
##        'Positive' Class : 1
##
```

Nous pouvons dire qu'on a un très bon modèle car nous avons une accuracy bien élevée. Il y a peu d'erreurs c'est-à-dire de poissons mal classés. Par contre, en ayant exécuté le script Rmarkdown plusieurs fois, j'ai remarqué un changement qui peut parfois être important de l'accuracy qui reste tout de même élevé. Cela est peut-être dû à la faible quantité de données dont nous disposons.