TP Note 1

Kuassi Pierre

2024-09-23

```
# importation des données
df <- read.table("/Users/peter/Desktop/Master 2 /UFC/Apprentisage/Goga/TP/graisse2.txt", header = TRUE)
```

R Markdown

```
str(df)
```

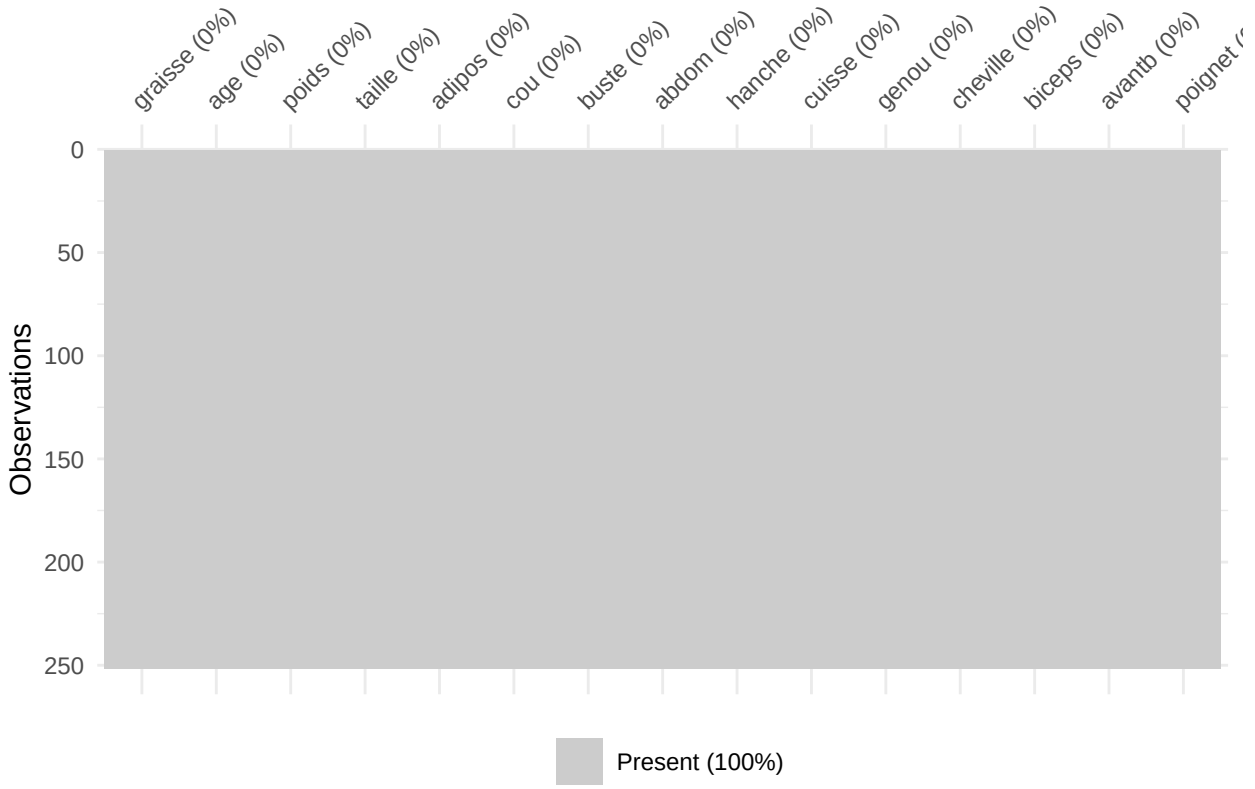
```
## 'data.frame':    251 obs. of  15 variables:
##  $ graisse : num  12.6 6.9 24.6 10.9 27.8 20.6 19 12.8 5.1 12 ...
##  $ age     : int  23 22 22 26 24 24 26 25 25 23 ...
##  $ poids   : num  154 173 154 185 184 ...
##  $ taille  : num  67.8 72.2 66.2 72.2 71.2 ...
##  $ adipos  : num  23.7 23.4 24.7 24.9 25.6 26.5 26.2 23.6 24.6 25.8 ...
##  $ cou     : num  36.2 38.5 34 37.4 34.4 39 36.4 37.8 38.1 42.1 ...
##  $ buste   : num  93.1 93.6 95.8 101.8 97.3 ...
##  $ abdom   : num  85.2 83 87.9 86.4 100 94.4 90.7 88.5 82.5 88.6 ...
##  $ hanche  : num  94.5 98.7 99.2 101.2 101.9 ...
##  $ cuisse  : num  59 58.7 59.6 60.1 63.2 66 58.4 60 62.9 63.1 ...
##  $ genou   : num  37.3 37.3 38.9 37.3 42.2 42 38.3 39.4 38.3 41.7 ...
##  $ cheville: num  21.9 23.4 24 22.8 24 25.6 22.9 23.2 23.8 25 ...
##  $ biceps  : num  32 30.5 28.8 32.4 32.2 35.7 31.9 30.5 35.9 35.6 ...
##  $ avantb  : num  27.4 28.9 25.2 29.4 27.7 30.6 27.8 29 31.1 30 ...
##  $ poignet : num  17.1 18.2 16.6 18.2 17.7 18.8 17.7 18.8 18.2 19.2 ...
```

```
summary(df)
```

```
##     graisse           age            poids           taille     
##  Min.   : 0.00   Min.   :22.00   Min.   :118.5   Min.   :64.00  
##  1st Qu.:12.80   1st Qu.:35.50   1st Qu.:158.8   1st Qu.:68.25  
##  Median :19.00   Median :43.00   Median :176.2   Median :70.00  
##  Mean   :18.89   Mean   :44.89   Mean   :178.8   Mean   :70.31  
##  3rd Qu.:24.55   3rd Qu.:54.00   3rd Qu.:196.9   3rd Qu.:72.25  
##  Max.   :45.10   Max.   :81.00   Max.   :363.1   Max.   :77.75  
##      adipos           cou            buste           abdom      
##  Min.   :18.10   Min.   :31.10   Min.   : 79.3   Min.   : 69.40  
##  1st Qu.:23.10   1st Qu.:36.40   1st Qu.: 94.3   1st Qu.: 84.55  
##  Median :25.00   Median :38.00   Median : 99.6   Median : 90.90  
##  Mean   :25.42   Mean   :38.00   Mean   :100.8   Mean   : 92.51  
##  3rd Qu.:27.30   3rd Qu.:39.45   3rd Qu.:105.3   3rd Qu.: 99.20  
##  Max.   :48.90   Max.   :51.20   Max.   :136.2   Max.   :148.10  
##      hanche           cuisse          genou          cheville    
##  Min.   : 85.00   Min.   :47.20   Min.   :33.00   Min.   :19.1  
##  1st Qu.: 95.50   1st Qu.:56.00   1st Qu.:36.95   1st Qu.:22.0
```

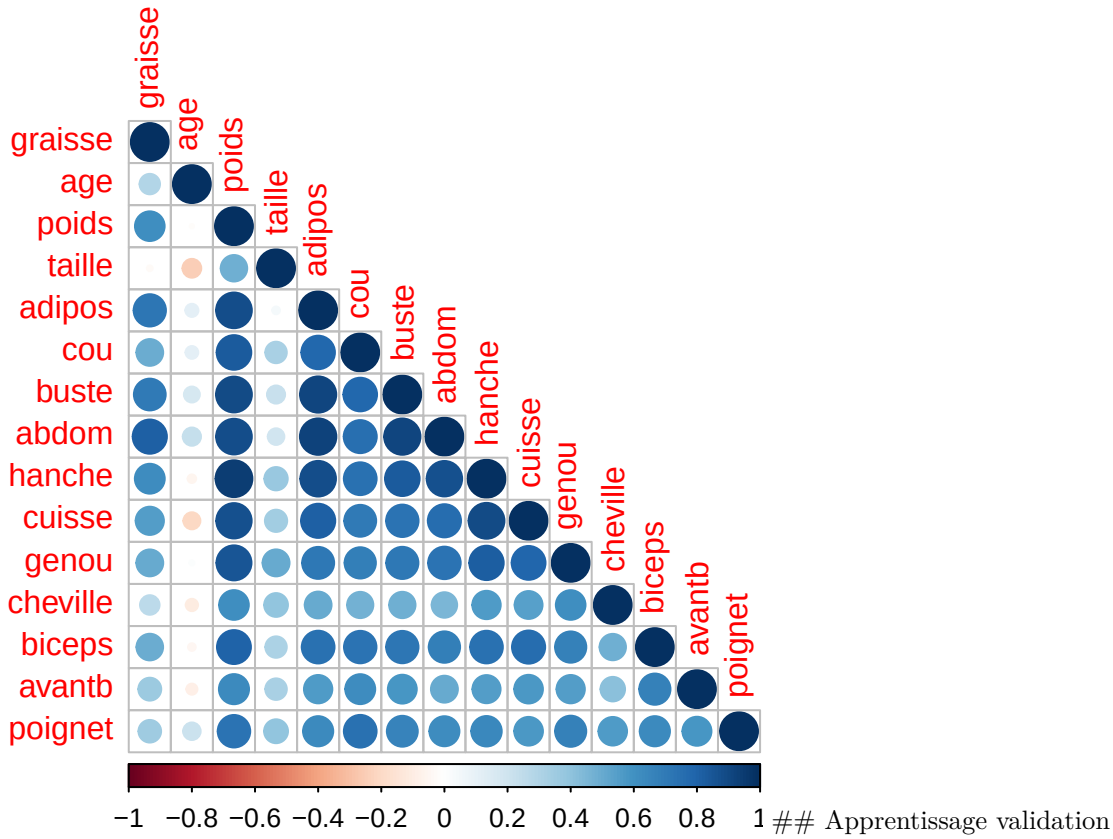
```
##  Median : 99.30   Median :59.00   Median :38.50   Median :22.8
##  Mean   : 99.84   Mean   :59.36   Mean   :38.57   Mean   :23.1
##  3rd Qu.:103.35   3rd Qu.:62.30   3rd Qu.:39.90   3rd Qu.:24.0
##  Max.   :147.70   Max.   :87.30   Max.   :49.10   Max.   :33.9
##      biceps          avantb          poignet
##  Min.   :24.80   Min.   :21.00   Min.   :15.80
##  1st Qu.:30.20   1st Qu.:27.30   1st Qu.:17.60
##  Median :32.00   Median :28.70   Median :18.30
##  Mean   :32.27   Mean   :28.66   Mean   :18.23
##  3rd Qu.:34.35   3rd Qu.:30.00   3rd Qu.:18.80
##  Max.   :45.00   Max.   :34.90   Max.   :21.40
```

```
library(naniar)
vis_miss(df)
```



Including Plots

```
correlation <- cor(df)
corrplot::corrplot(correlation, method = "circle", type = "lower")
```



Apprentissage validation

```
X <- model.matrix(graisse~., df)[,-1]
y <- df$graisse

set.seed(22)
train <- sample(1:nrow(X), nrow(X)/2)
ytest <- y[-train]
Xtest <- X[-train,]

vect.lambda <-10^seq(-3, 4, length= 80)

library(glmnet)
```

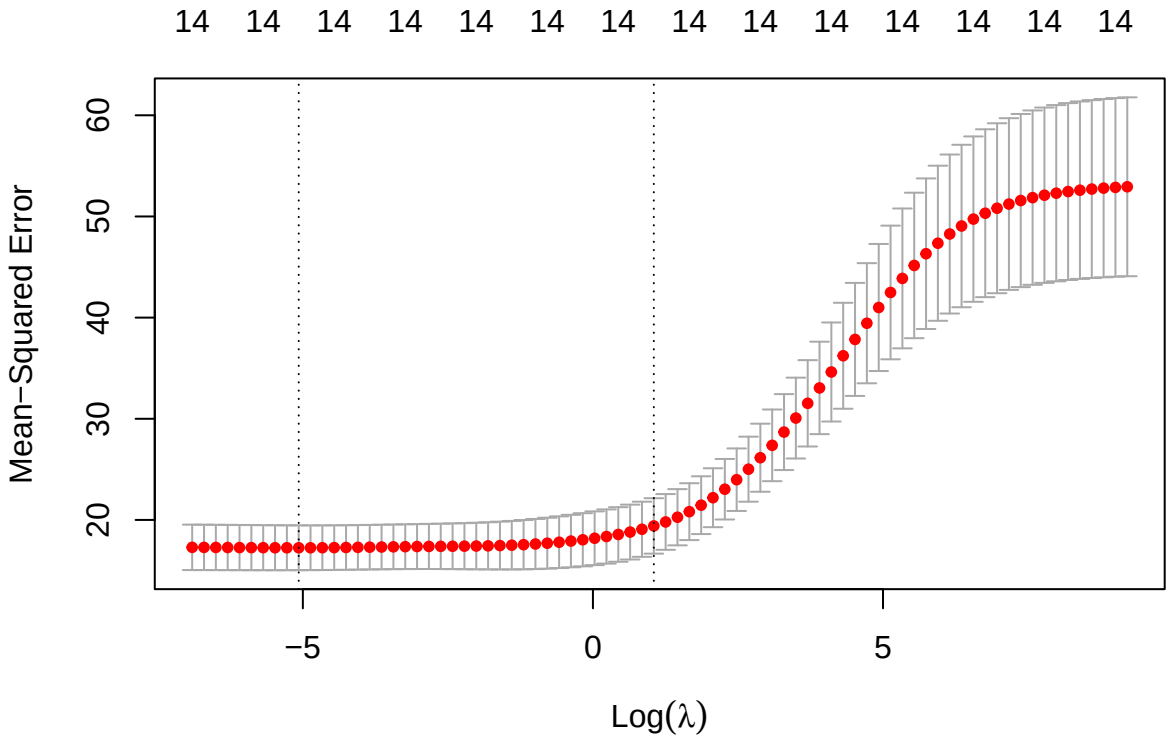
```
## Loading required package: Matrix
```

```
## Loaded glmnet 4.1-8
```

```
model.ridge.train <- glmnet(X[train,], y[train], alpha = 0, lambda = vect.lambda)
```

determinons le meilleur avec la validation croisé par bloc

```
model.ridge.cv <- cv.glmnet(X[train,],y[train], alpha=0, lambda = vect.lambda)
plot(model.ridge.cv)
```




```
lambda.optimal <- model.ridge.cv$lambda.min
```

```
ypred <- predict(model.ridge.cv, newx = Xtest, s ="lambda.min" )
mean((ytest - ypred)^2)
```

```
## [1] 18.9714
```

Methode 2

```
pred <- predict(model.ridge.train, newx =Xtest)
Y <- rep(ytest, length(vect.lambda))
r <- (Y-pred)^2
MSE <- apply(r, 2, mean )
lambda.opt <- model.ridge.train$lambda[which.min(MSE)]
```

Ici pour determiner la meilleur λ_{opt} on calcule le $MSE(MeanSquaredError)$ sur le groupe de validation pour évaluer la performance du modèle. Le MSE minimal MSE_{min} correspond la valeur optimale de λ.