Projet : étude du cancer du sein ER+

```
library(FactoMineR)
library(glmnet)
```

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

```
## Loaded glmnet 4.1-3
```

```
load("/Users/bouacha_lazhar/OneDrive/Master MMA/M2 MMA/M2 S3/Apprentissage en Grande Dimension/Partie I
# le jeu de données est accessible via MA_TCGA
load("/Users/bouacha_lazhar/OneDrive/Master MMA/M2 MMA/M2 S3/Apprentissage en Grande Dimension/Partie I
load("/Users/bouacha_lazhar/OneDrive/Master MMA/M2 MMA/M2 S3/Apprentissage en Grande Dimension/Partie I
```

I. Chargement et nettoyage des données

1.

```
data <- scale(MA_TCGA, center = TRUE, scale = FALSE)
dim(data)
```

```
## [1]  1093 16021
```

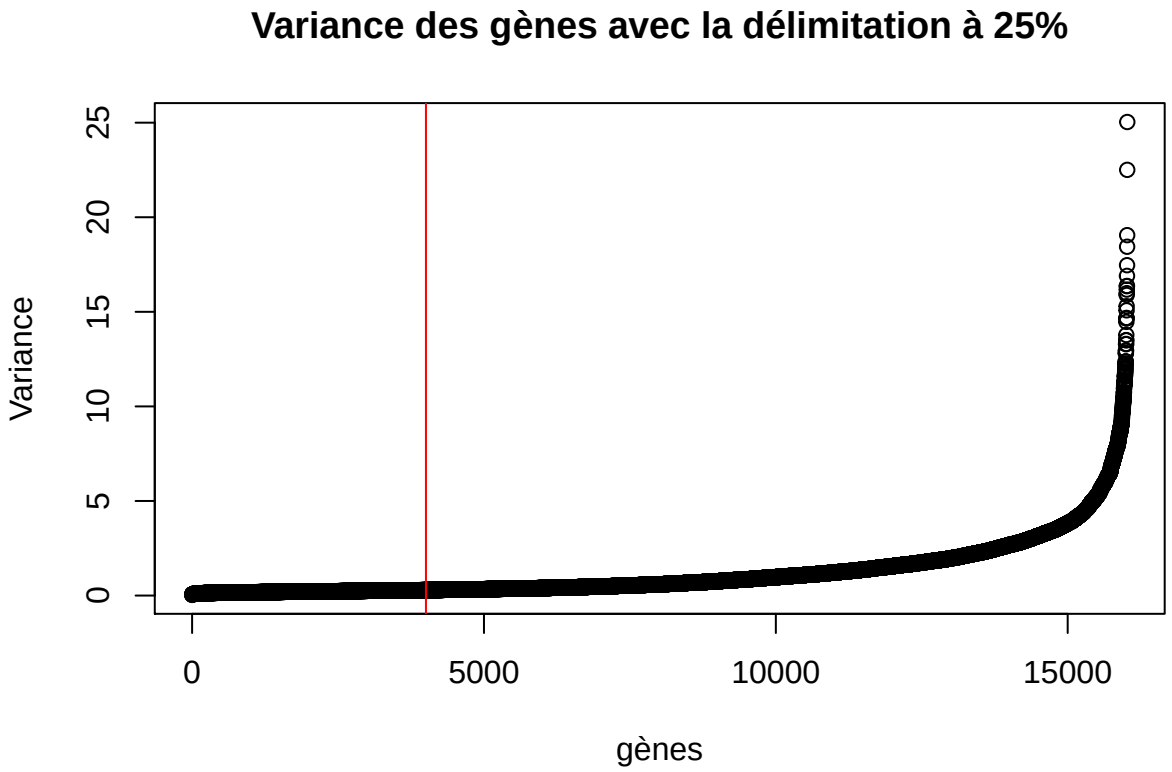
2.

```
Vdata <- apply(data, 2, var)
Vdata2 <- order(Vdata)
index <- floor(length(Vdata2)*0.25) + 1
data2 <- data[,-Vdata2[1:index]]
dim(data2)
```

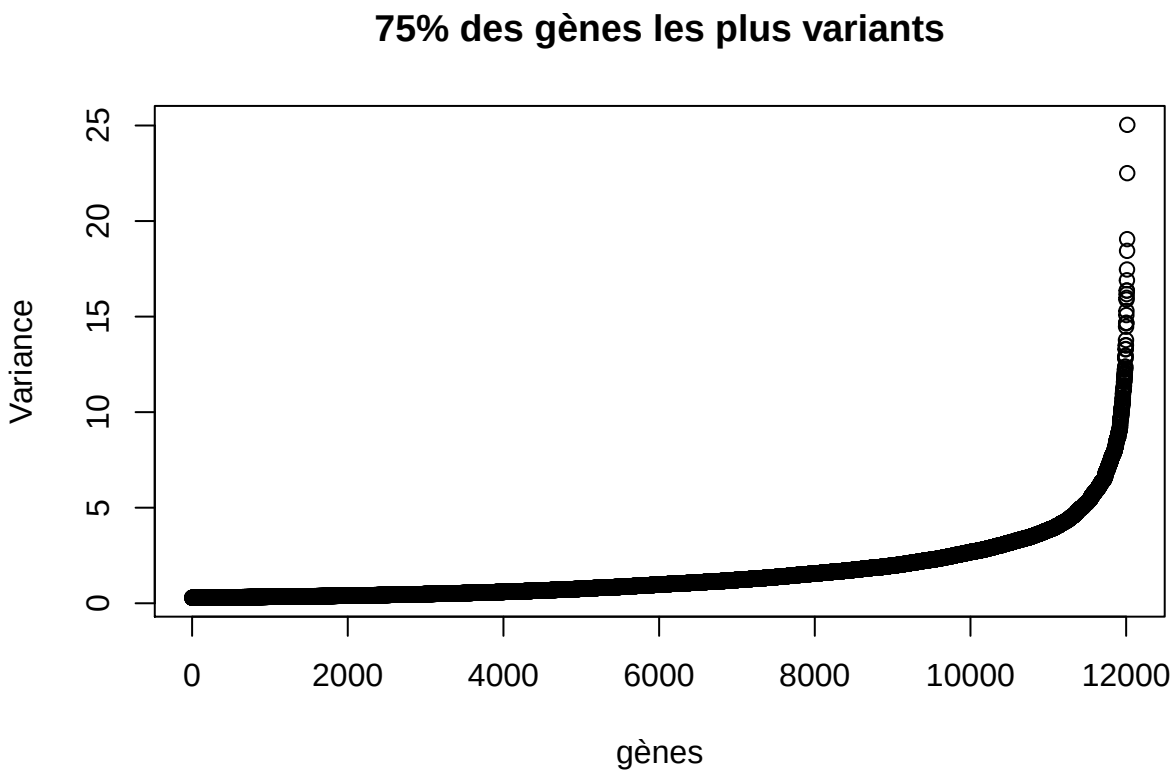
```
## [1]  1093 12015
```

```
Vdata3 <- apply(data2, 2, var)

plot(1:length(Vdata), sort(Vdata), col = "black", main = "Variance des gènes avec la délimitation à 25%
abline(v = index, col = 'red')
```




```
plot(1:length(Vdata3), sort(Vdata3), col = "black", main = "75% des gènes les plus variants", xlab = "g
```



II. Tests multiples pour réduire la dimension du jeu de données

1.

```
ER <- clinicalData$patient.breast_carcinoma_estrogen_receptor_status
Clin <- subset(clinicalData, !is.na(ER) & !ER == 'indeterminate')
ER <- as.data.frame(Clin$patient.breast_carcinoma_estrogen_receptor_status)
newER <- as.numeric(ER[,1]) - 2
length(newER)
```

```
## [1] 1043
```

```
Clin2 <- Clin
Clin2$patient.breast_carcinoma_estrogen_receptor_status <- newER
dim(Clin2)
```

```
## [1] 1043 3721
```

```
data3 <- data2[row.names(Clin2),]
dim(data3)
```

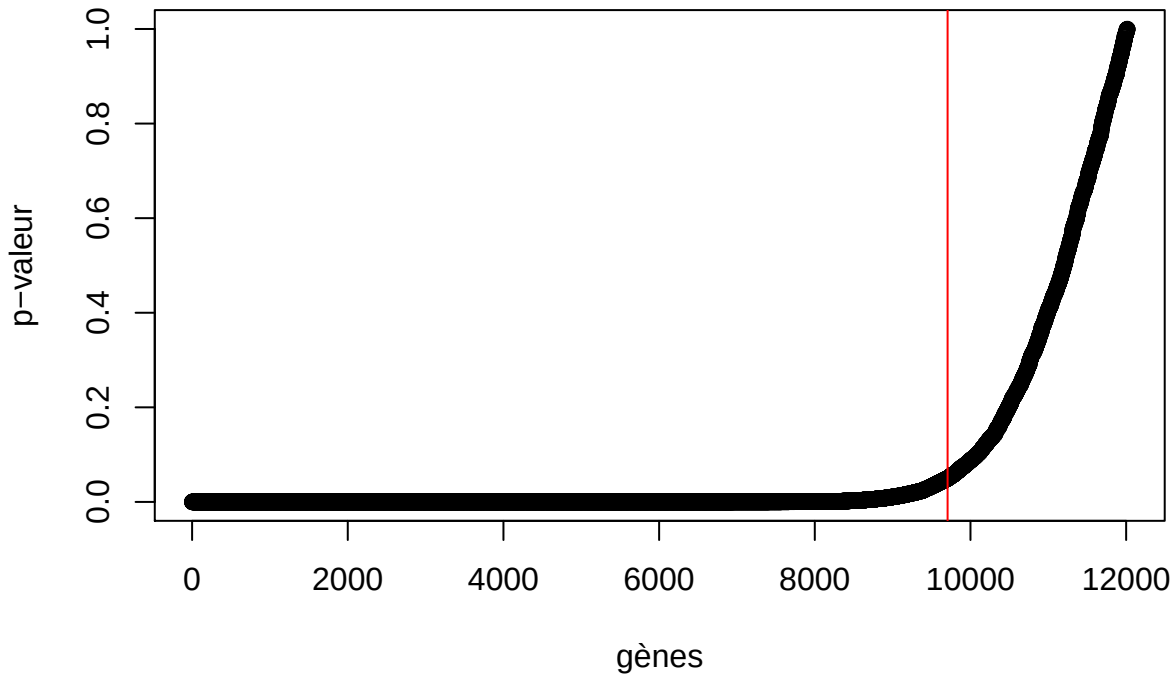
```
## [1]  1043 12015
```

2.

```
ERn <- which(newER == 0)
ERp <- which(newER == 1)
pval <- apply(data3, 2, function(x){
  t.test(x[ERn], x[ERp])$p.value
  })
pvalBH <- p.adjust(pval,method="BH")
length(which(pvalBH<0.05))
```

```
## [1] 9706
```

```
plot(1:length(pvalBH), sort(pvalBH), main = "Analyse différentielle entre les cancers ER+ et non ER+", 
abline(v = length(which(pvalBH<0.05)), col = 'red')
```



Sur les 12015 gènes, 9706 sont identifiés comme différentiellement exprimés avec un contrôle de la FDR de 5%.

```
data4 <- data3[,which(pvalBH<0.05)]
dim(data4)
```

```
## [1] 1043 9706
```

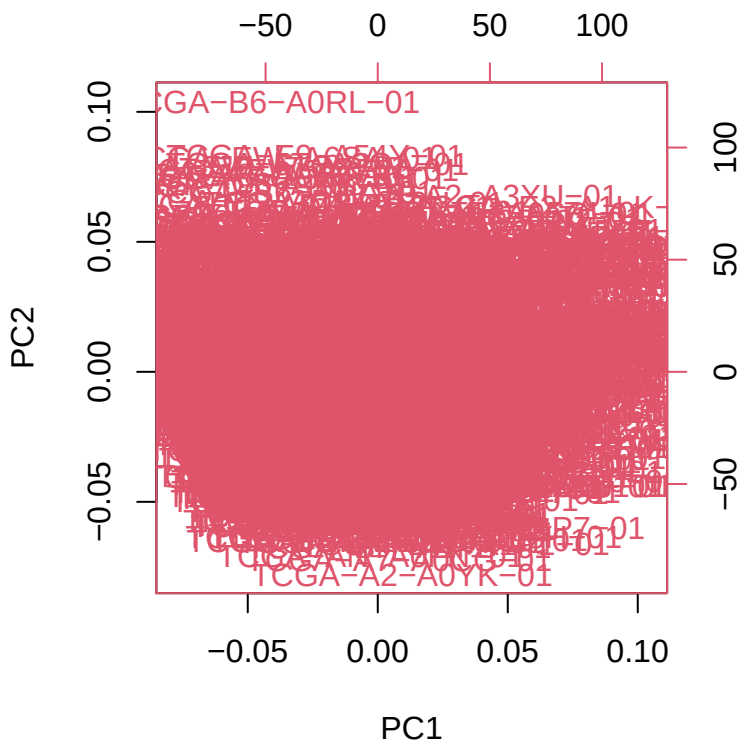
III. ACP pour différencier les deux groupes

1.

```
data.acp <- prcomp(data4)
```

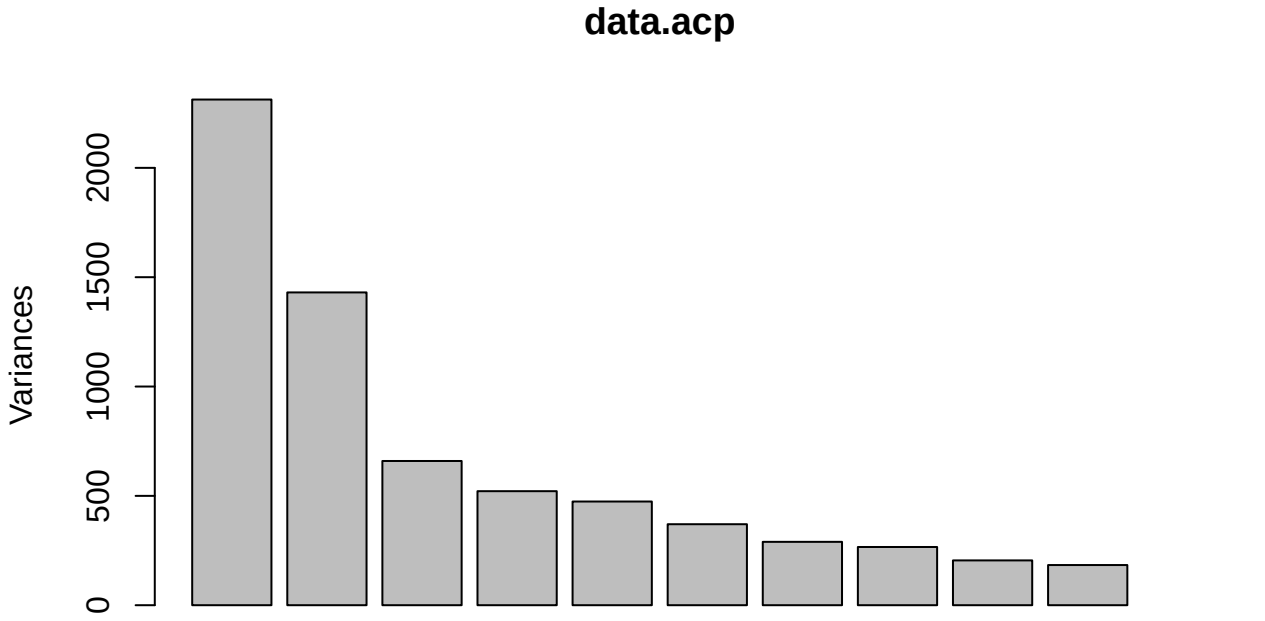
2.

```
biplot(data.acp, col = newER + 1)
```



3.

```
plot(data.acp)
```

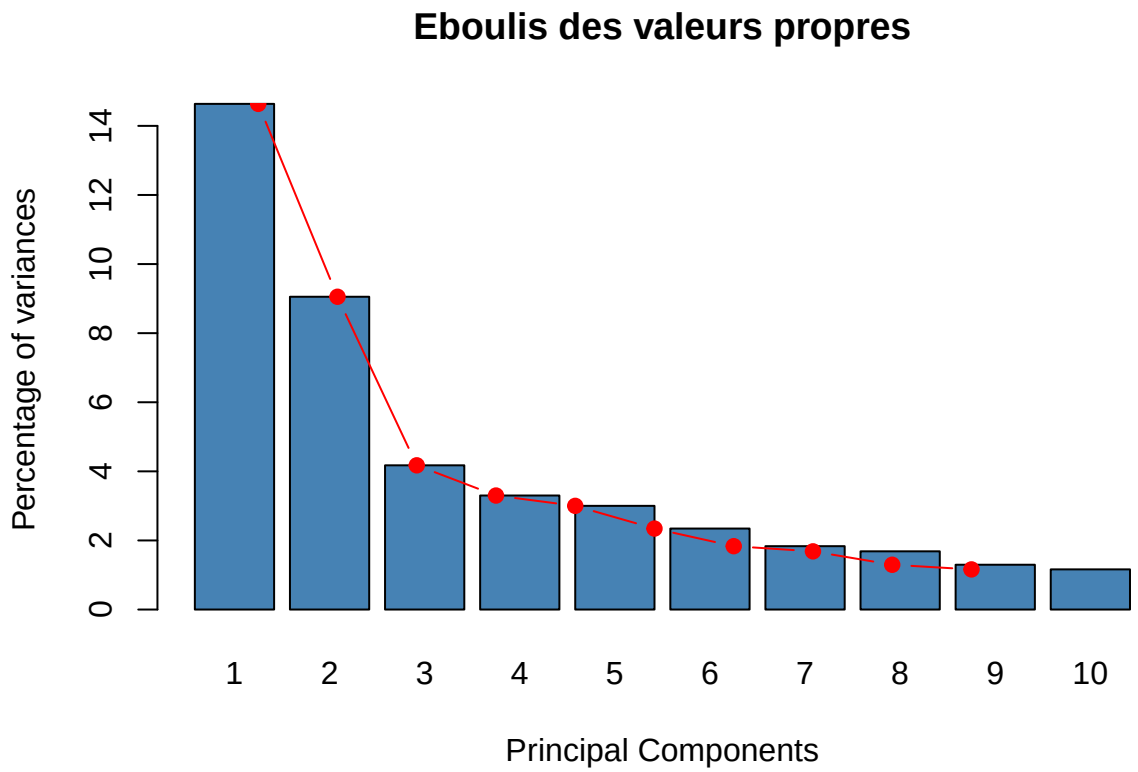



```
eigenvalues <- 100*(data.acp$sdev^2/sum(data.acp$sdev^2))
id <- length(which(eigenvalues > mean(eigenvalues)))
id
```

```
## [1] 140
```

```
genes <- sort(data.acp$rotation[,"PC1"], decreasing = TRUE)[1:id]
genes <- row.names(as.data.frame(genes))

barplot(eigenvalues[1:10], names.arg=1:10,
        main = "Eboulis des valeurs propres",
        xlab = "Principal Components",
        ylab = "Percentage of variances",
        col ="steelblue")
lines(x = 1:10, eigenvalues[1:10],
      type="b", pch=19, col = "red")
```

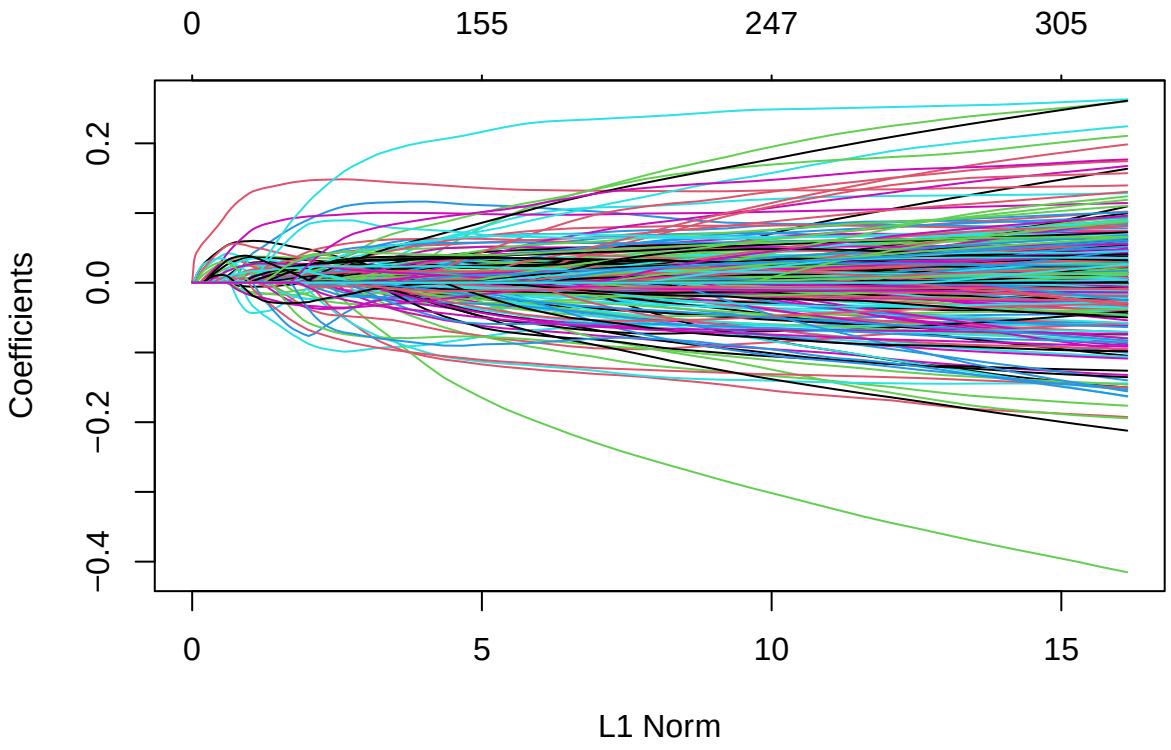


Suivant la règle de Kaiser : on ne conserve que les valeurs propres supérieures à leur moyenne. On ne garderait que les 140 1ers axes pour un total de 73.88% de l'inertie.

IV. Apprentissage par méthodes régularisées pour valider les résultats

1.

```
model <- glmnet(data4, newER, family="binomial", alpha=0.5)
plot(model)
```



2.

```
model.cv <- cv.glmnet(data4, newER, nfolds=5, type.measure = "mse", family="binomial")
glmnet.model <- glmnet(data4, newER, family="binomial",  alpha=0.5, nlambda=1, lambda = model.cv$lambda
genes1 <- colnames(data4[,which(abs(glmnet.model$beta)>0)])
length(which(abs(glmnet.model$beta)>0))
```

```
## [1] 241
```

```
length(append(genes, genes1)) - length(unique(append(genes, genes1)))
```

```
## [1] 3
```

Il y a 241 gènes dont 3 communs avec l'ACP de la partie III.

3.

```
data.test <- scale(MA_TCGA_test[,colnames(data4)], center = TRUE, scale = FALSE)
dim(data.test)
```

```
## [1]   48 9706
```

```
test.prediction <- predict.glmnet(glmnet.model, data.test, type="response")
test.prediction.bin <- test.prediction
test.prediction.bin[test.prediction.bin > 0.5] <- 1
test.prediction.bin[test.prediction.bin <= 0.5] <- 0
test.prediction.bin <- as.numeric(test.prediction.bin)
```

On retrouve une prédiction sur des patients avec une ER+ non donnée.