Projet 4/5 ACP et AFD

Nicolas SALVAN - Alexandre CORRIOU

2024-05-24

Lecture des données nettoyées

Importation du dataset

```
data <- read.csv("data/data_cleaned.csv", header = TRUE, sep = ",")
data$diagnosis <- as.factor(data$diagnosis)
```

Aperçu rapide

```
head(data)
```

```
##   diagnosis radius_mean texture_mean perimeter_mean area_mean smoothness_mean
## 1         M       17.99        10.38         122.80    1001.0         0.11840
## 2         M       20.57        17.77         132.90    1326.0         0.08474
## 3         M       19.69        21.25         130.00    1203.0         0.10960
## 4         M       11.42        20.38          77.58     386.1         0.14250
## 5         M       20.29        14.34         135.10    1297.0         0.10030
## 6         M       12.45        15.70          82.57     477.1         0.12780
##   compactness_mean concavity_mean concave.points_mean symmetry_mean
## 1          0.27760         0.3001             0.14710        0.2419
## 2          0.07864         0.0869             0.07017        0.1812
## 3          0.15990         0.1974             0.12790        0.2069
## 4          0.28390         0.2414             0.10520        0.2597
## 5          0.13280         0.1980             0.10430        0.1809
## 6          0.17000         0.1578             0.08089        0.2087
##   fractal_dimension_mean radius_se texture_se perimeter_se area_se
## 1                0.07871    1.0950     0.9053        8.589  153.40
## 2                0.05667    0.5435     0.7339        3.398   74.08
## 3                0.05999    0.7456     0.7869        4.585   94.03
## 4                0.09744    0.4956     1.1560        3.445   27.23
## 5                0.05883    0.7572     0.7813        5.438   94.44
## 6                0.07613    0.3345     0.8902        2.217   27.19
##   smoothness_se compactness_se concavity_se concave.points_se symmetry_se
## 1      0.006399        0.04904      0.05373           0.01587     0.03003
## 2      0.005225        0.01308      0.01860           0.01340     0.01389
## 3      0.006150        0.04006      0.03832           0.02058     0.02250
## 4      0.009110        0.07458      0.05661           0.01867     0.05963
```

```
## 5      0.011490        0.02461      0.05688           0.01885     0.01756
## 6      0.007510        0.03345      0.03672           0.01137     0.02165
##   fractal_dimension_se radius_worst texture_worst perimeter_worst area_worst
## 1             0.006193        25.38         17.33          184.60     2019.0
## 2             0.003532        24.99         23.41          158.80     1956.0
## 3             0.004571        23.57         25.53          152.50     1709.0
## 4             0.009208        14.91         26.50           98.87      567.7
## 5             0.005115        22.54         16.67          152.20     1575.0
## 6             0.005082        15.47         23.75          103.40      741.6
##   smoothness_worst compactness_worst concavity_worst concave.points_worst
## 1           0.1622            0.6656          0.7119               0.2654
## 2           0.1238            0.1866          0.2416               0.1860
## 3           0.1444            0.4245          0.4504               0.2430
## 4           0.2098            0.8663          0.6869               0.2575
## 5           0.1374            0.2050          0.4000               0.1625
## 6           0.1791            0.5249          0.5355               0.1741
##   symmetry_worst fractal_dimension_worst
## 1         0.4601                 0.11890
## 2         0.2750                 0.08902
## 3         0.3613                 0.08758
## 4         0.6638                 0.17300
## 5         0.2364                 0.07678
## 6         0.3985                 0.12440
```

```
# dim(data)
# str(data)
```

ACP - Analyse en Composantes Principales

Lancement d'une ACP sur les données

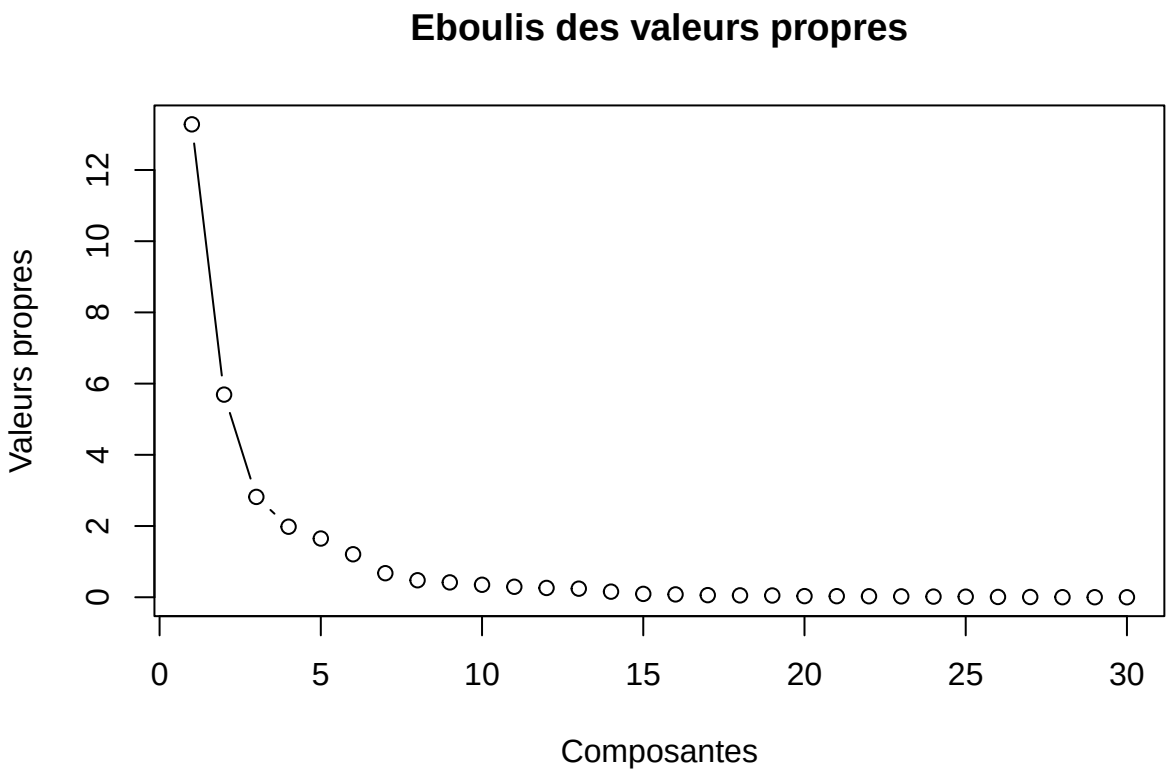
Nous allons réaliser une ACP sur nos données avec la bibliothèque `FactoMineR`.

```
# install.packages("FactoMineR")
library("FactoMineR")
```

```
res.pca <- PCA(data, scale.unit = TRUE, graph = FALSE, quali.sup = 1)
```

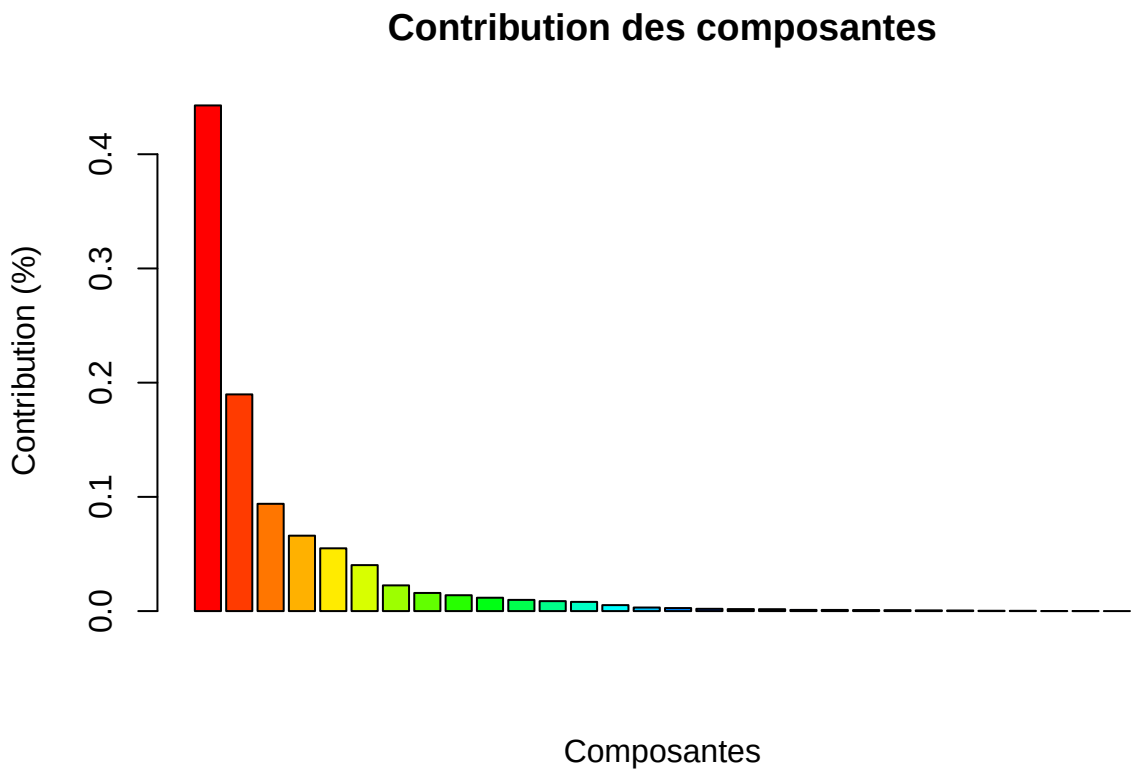
Choix du nombre de composantes

```
plot(res.pca$eig[,1], type = "b", xlab = "Composantes", ylab = "Valeurs propres", main = "Eboulis des va
```



On observe que les composantes principales sont les trois premières. On peut le vérifier en affichant leur contribution.

```
eig_percentage = res.pca$eig[,2]/sum(res.pca$eig[,2])
barplot(eig_percentage, names.arg = FALSE, col = rainbow(26), main = "Contribution des composantes", xla
```




```
eig_percentage[1:3]
```

```
##     comp 1     comp 2     comp 3
## 0.44272026 0.18971182 0.09393163
```

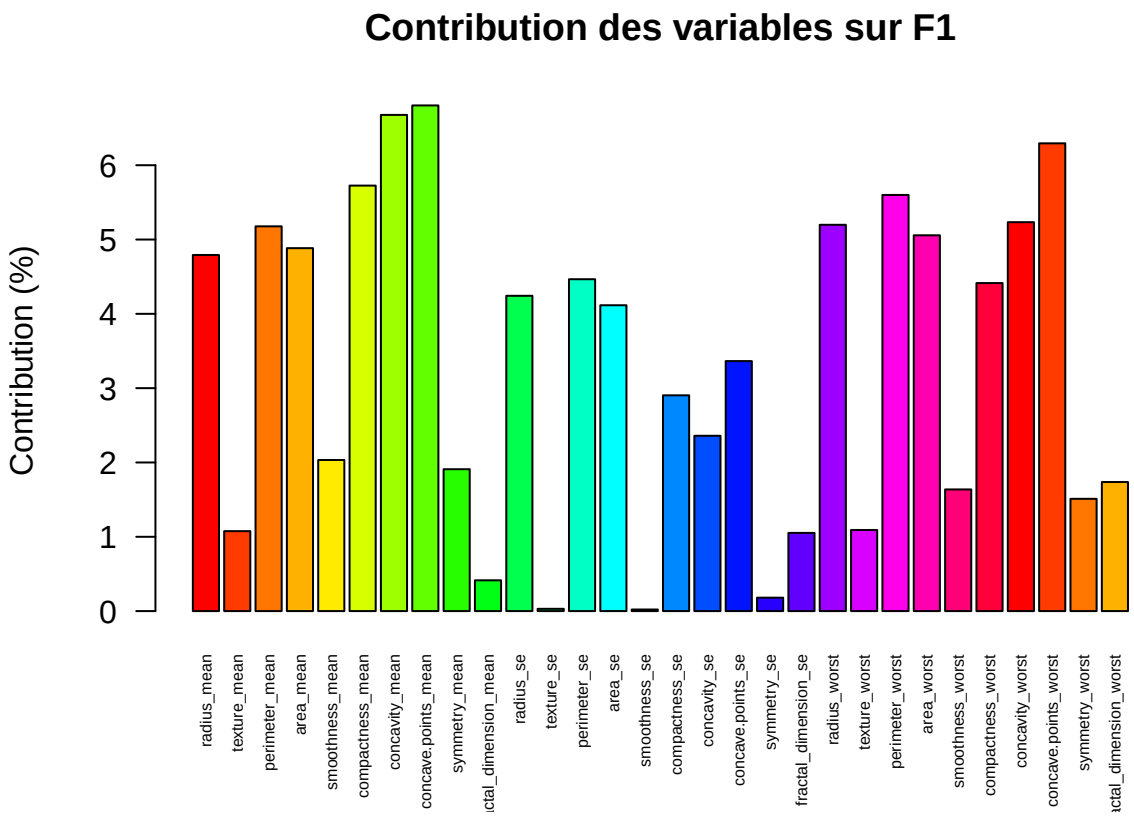
```
sum(eig_percentage[1:3])
```

```
## [1] 0.7263637
```

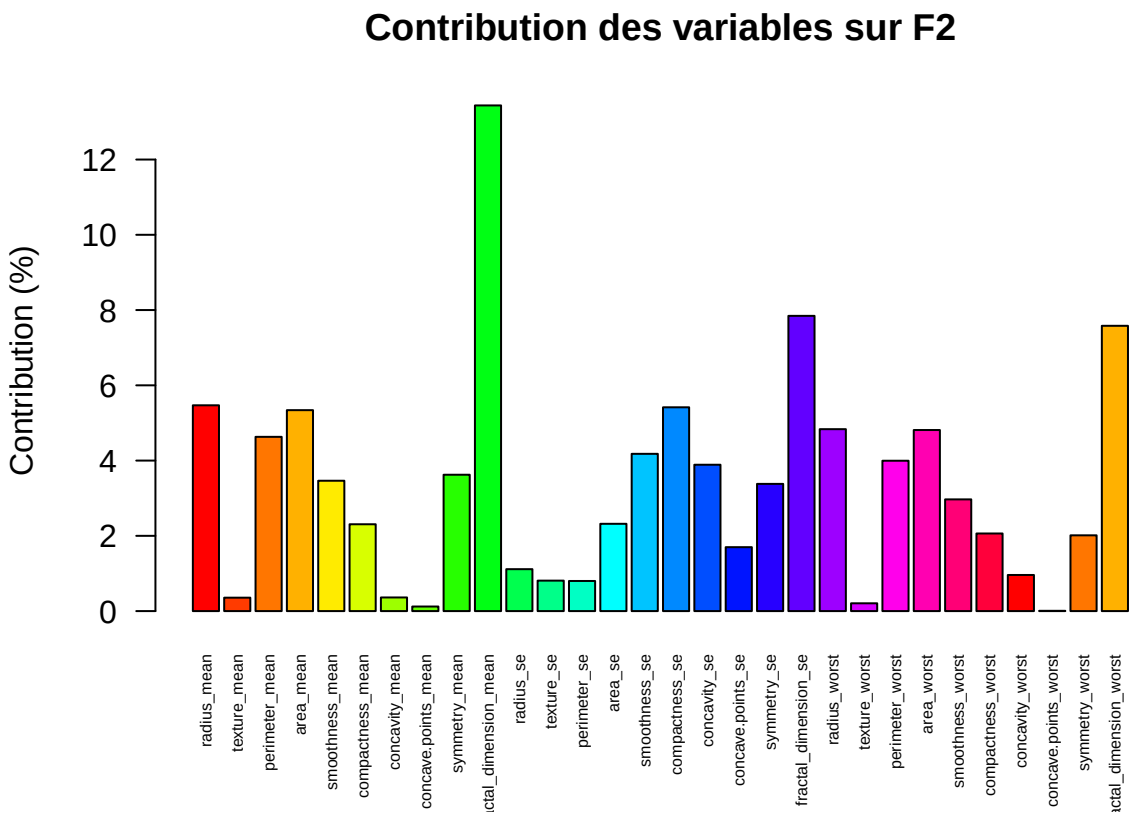
Les trois premières composantes représentent 72.6% de l'information.

Affichage des variables ayant le plus d'influence sur les axes

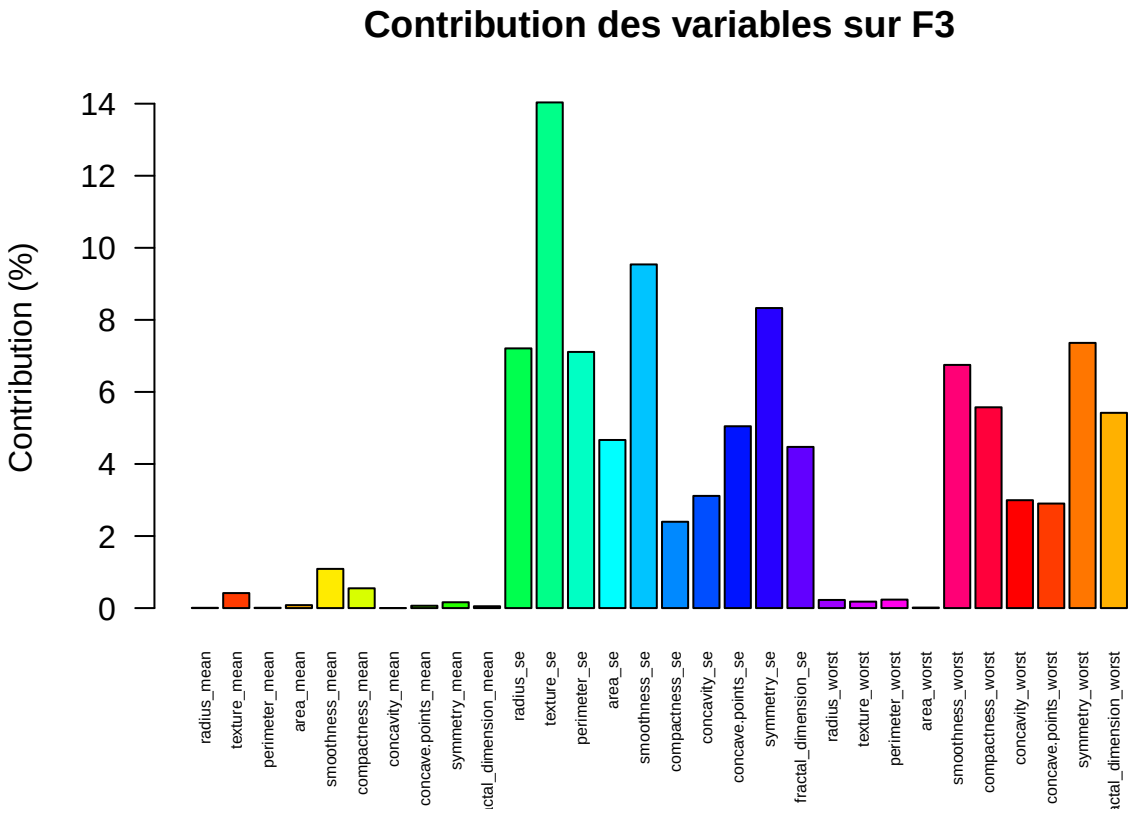
```
res.pca.contrib <- res.pca$var$contrib[, 1:3]
barplot(res.pca.contrib[,1], names.arg = rownames(res.pca$var$contrib), col = rainbow(26), main = "Cont
```




```
barplot(res.pca.contrib[,2], names.arg = rownames(res.pca$var$contrib), col = rainbow(26), main = "Conti
```



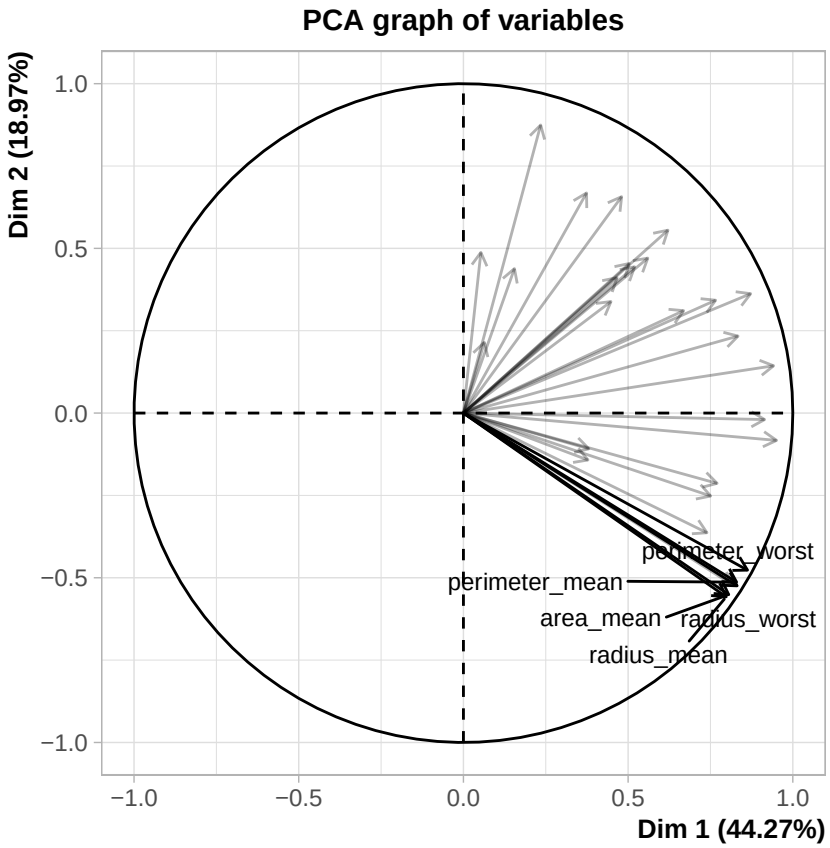

```
barplot(res.pca.contrib[,3], names.arg = rownames(res.pca$var$contrib), col = rainbow(26), main = "Cont
```



On observe ici les variables qui contribuent le plus dans les plans principaux de l'ACP.

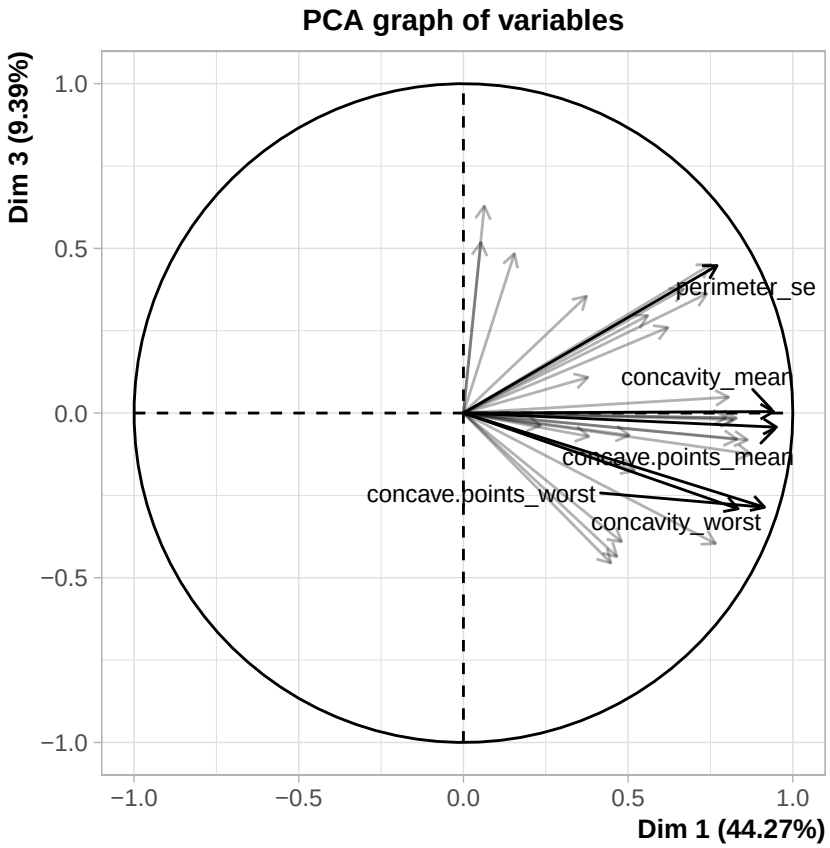
Plan (F1, F2)

```
plot(res.pca, choix = "var", cex = 0.8, col.var = "black", select = "contrib 5")
```



Plan (F1, F3)

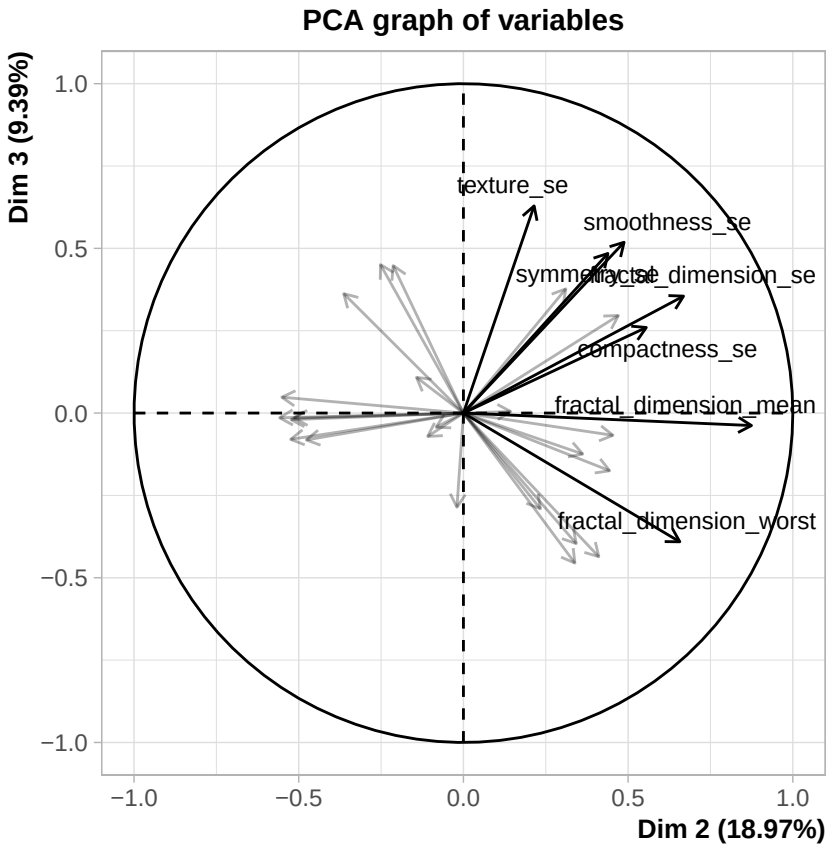
```
plot(res.pca, choix = "var", cex = 0.8, col.var = "black", select = "contrib 5", axes = c(1,3))
```



On obtient grâce à ce graphe les variables qui contribuent le plus dans ce plan.

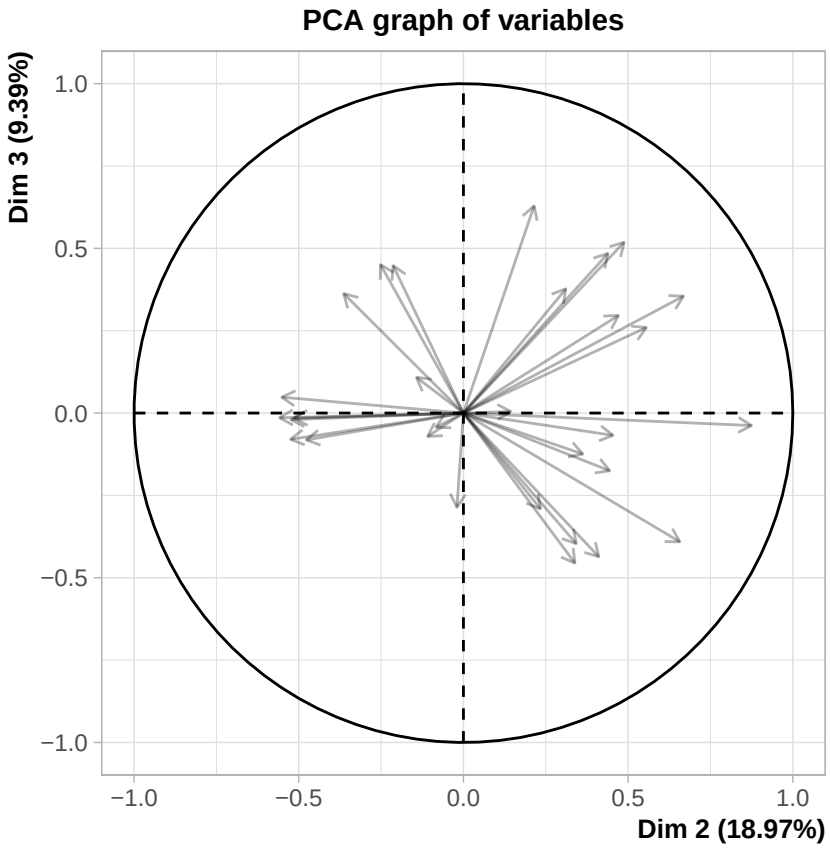
Plan (F2, F3)

```
plot(res.pca, choix = "var", cex = 0.8, col.var = "black", select = "contrib 7", axes = c(2,3))
```



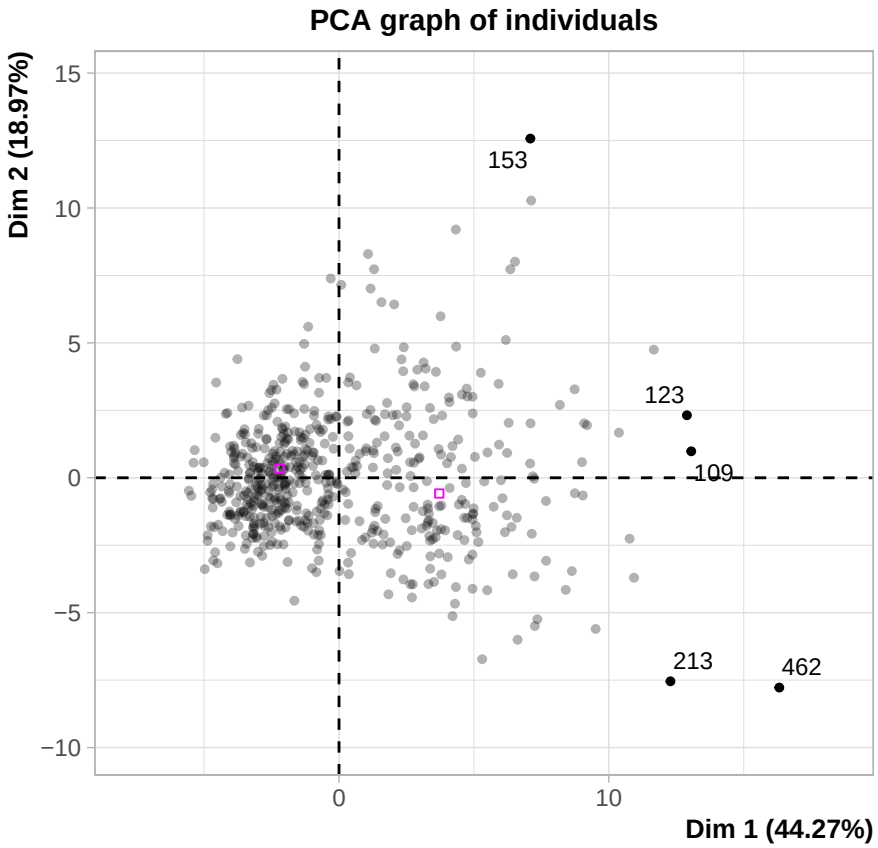
On observe que dans ce plan, les variables ne sont pas très bien représentées. On peut le voir en affichant celles avec un cos2 supérieur à 0.8 (aucune).

```
plot(res.pca, choix = "var", cex = 0.8, col.var = "black", select = "cos2 .8", axes = c(2,3))
```



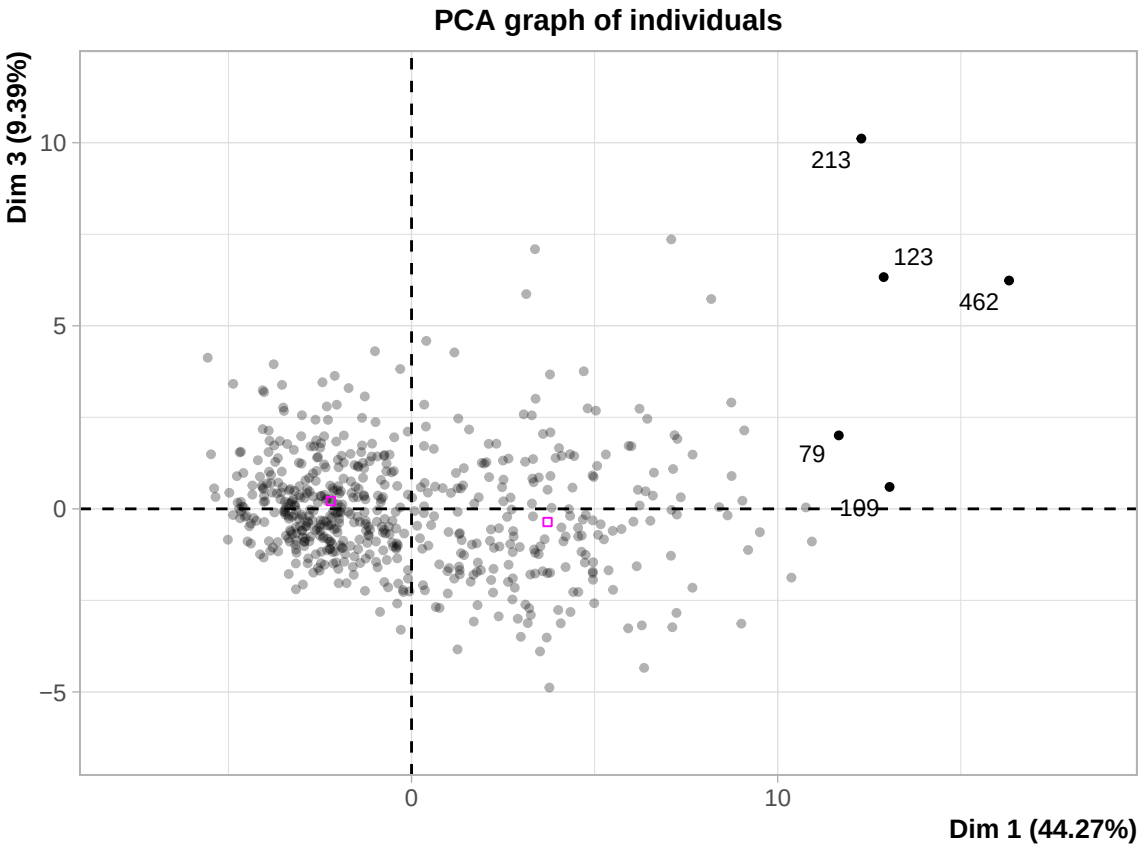
Représentation des individus

```
plot(res.pca, choix = "ind", cex = 0.8, col.ind = "black", select = "contrib 5", label = "ind")
```

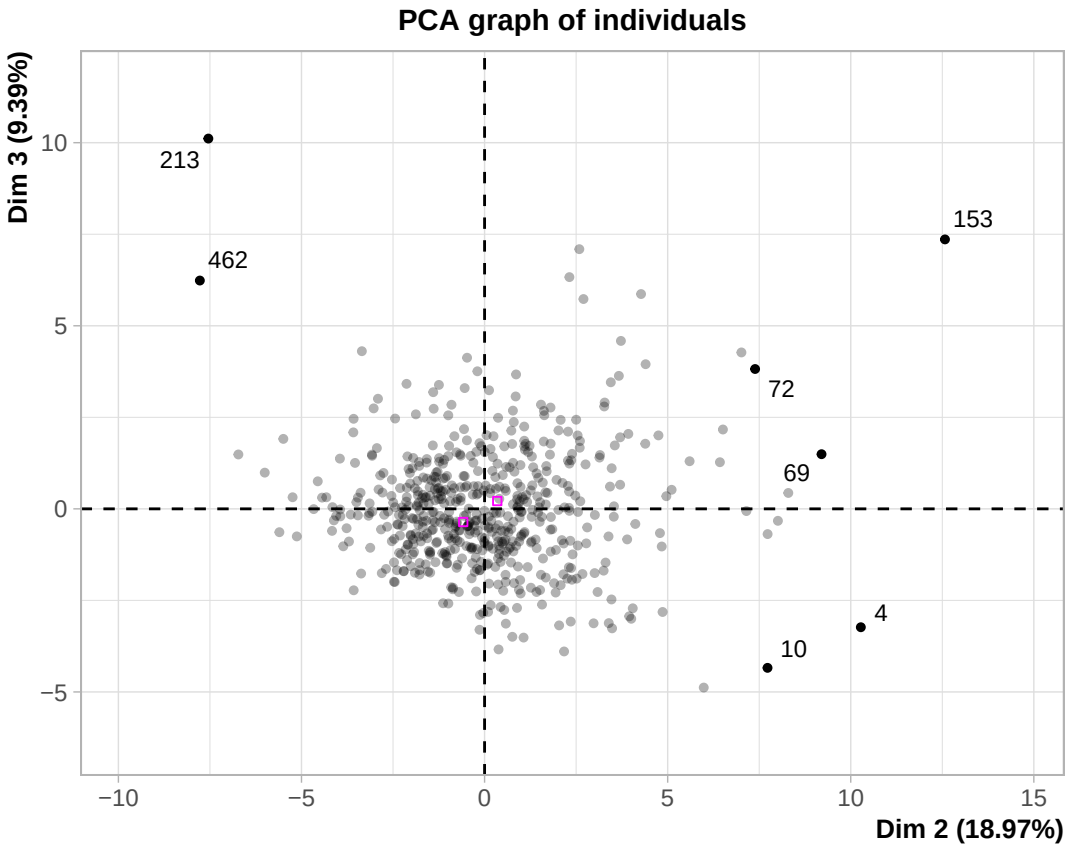


On observe que certains individus contribuent beaucouo dans ce plan. Regardons si ces individus ont autant d'influence sur les autres plans (F1, F3) et (F2, F3).

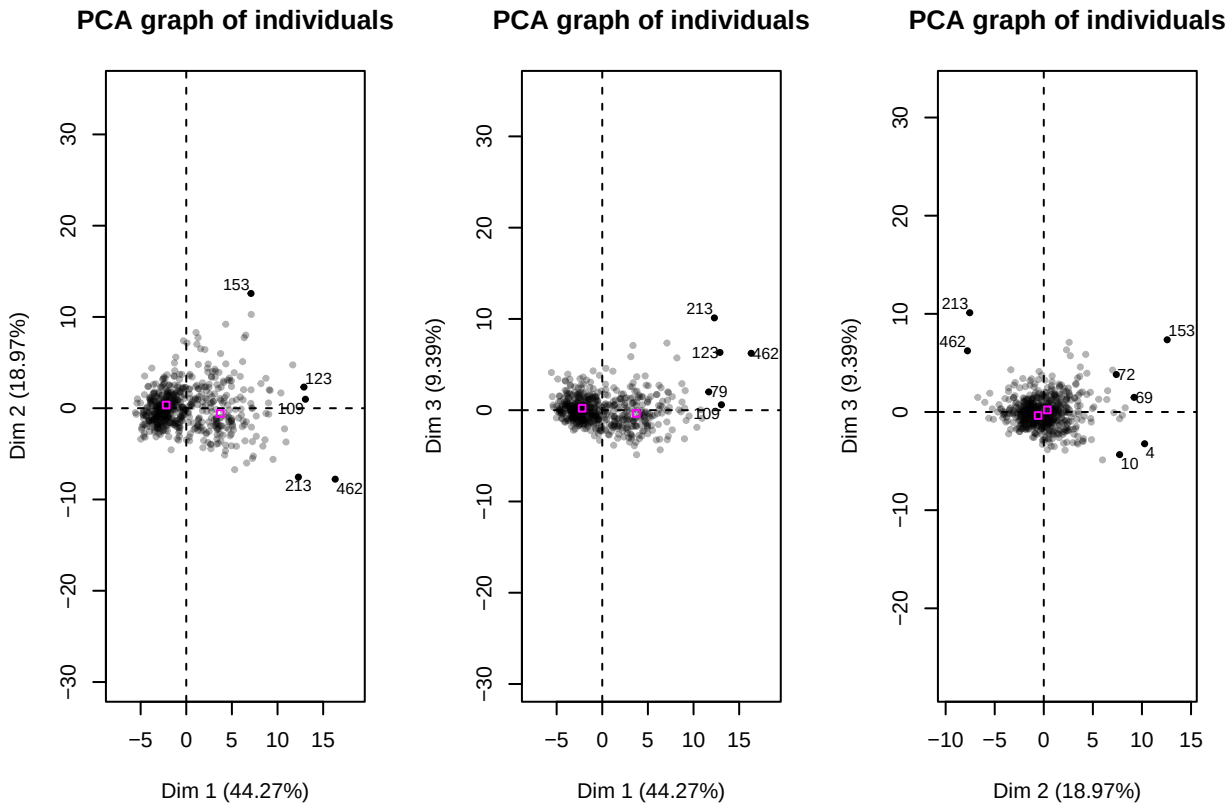
```
plot(res.pca, choix = "ind", cex = 0.8, col.ind = "black", select = "contrib 5", axes = c(1,3), label =
```




```
plot(res.pca, choix = "ind", cex = 0.8, col.ind = "black", select = "contrib 7", axes = c(2,3), label =
```




```
par(mfrow=c(1,3))

plot(res.pca, graph.type = "classic", choix = "ind", cex = 0.8, col.ind = "black", select = "contrib 5"

plot(res.pca, choix = "ind", graph.type = "classic", cex = 0.8, col.ind = "black", select = "contrib 5"

plot(res.pca, choix = "ind", graph.type = "classic", cex = 0.8, col.ind = "black", select = "contrib 7"
```



Les individus 462, 213, 123 sont très influents sur les plans (F1, F2) et (F1, F3). Il serait intéressant de les étudier plus en détail.

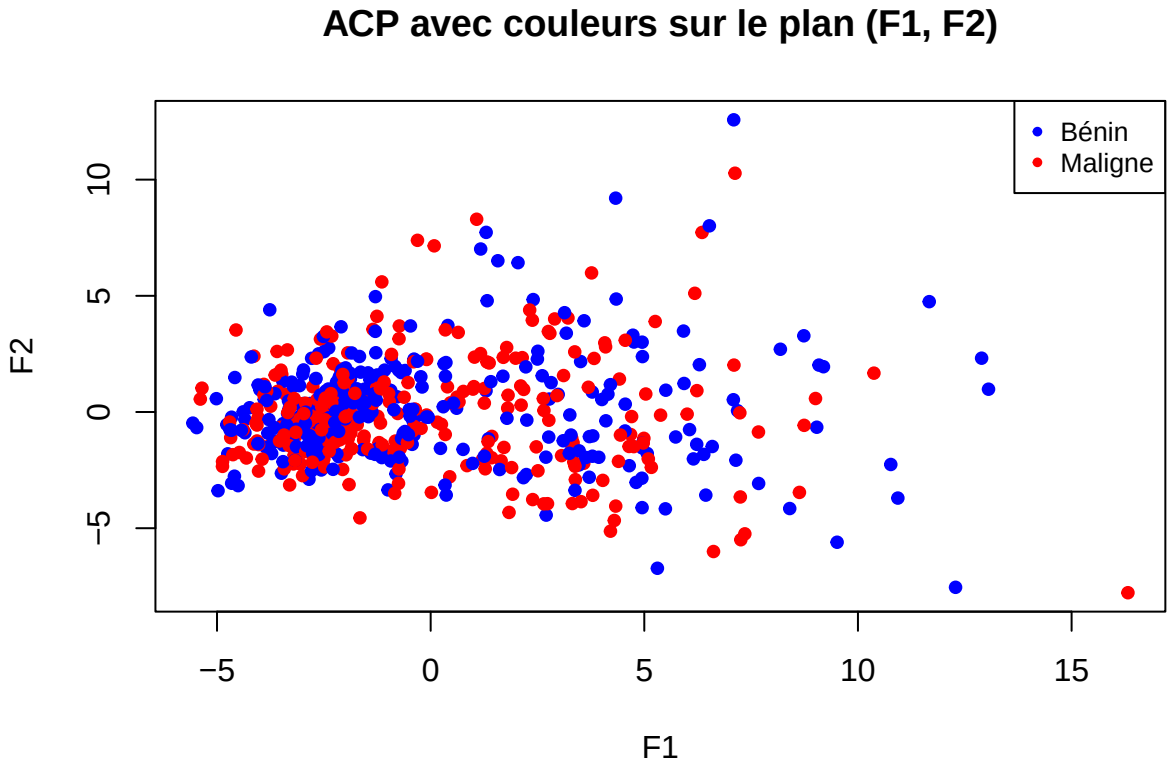
```
data[c(462, 213, 123),]
```

```
##     diagnosis radius_mean texture_mean perimeter_mean area_mean smoothness_mean
## 462         M       27.42        26.27          186.9      2501          0.1084
## 213         M       28.11        18.47          188.5      2499          0.1142
## 123         M       24.25        20.20          166.2      1761          0.1447
##     compactness_mean concavity_mean concave.points_mean symmetry_mean
## 462           0.1988         0.3635              0.1689        0.2061
## 213           0.1516         0.3201              0.1595        0.1648
## 123           0.2867         0.4268              0.2012        0.2655
##     fractal_dimension_mean radius_se texture_se perimeter_se area_se
## 462                0.05623     2.547      1.306       18.650   542.2
## 213                0.05525     2.873      1.476       21.980   525.6
## 123                0.06877     1.509      3.120        9.807   233.0
##     smoothness_se compactness_se concavity_se concave.points_se symmetry_se
## 462       0.00765        0.05374      0.08055           0.02598     0.01697
## 213       0.01345        0.02772      0.06389           0.01407     0.04783
## 123       0.02333        0.09806      0.12780           0.01822     0.04547
##     fractal_dimension_se radius_worst texture_worst perimeter_worst area_worst
## 462             0.004558        36.04         31.37           251.2       4254
## 213             0.004476        28.11         18.47           188.5       2499
## 123             0.009875        26.02         23.99           180.9       2073
##     smoothness_worst compactness_worst concavity_worst concave.points_worst
```

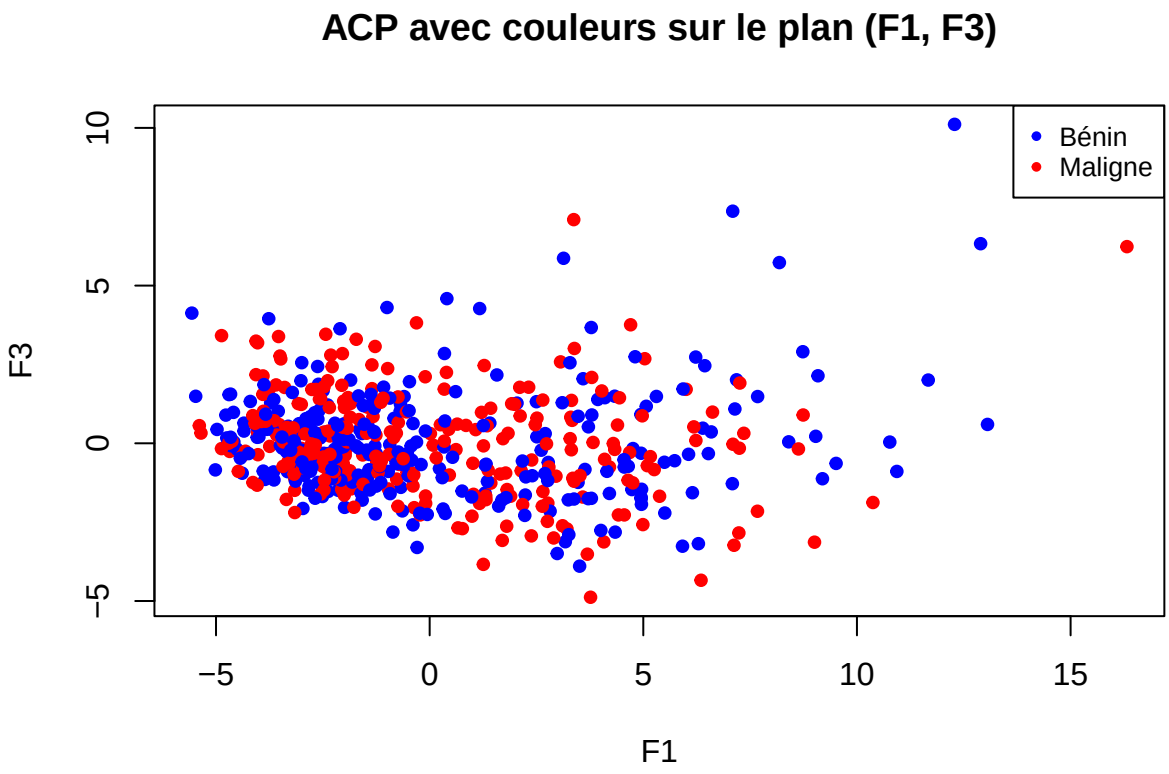
```
## 462           0.1357            0.4256          0.6833               0.2625
## 213           0.1142            0.1516          0.3201               0.1595
## 123           0.1696            0.4244          0.5803               0.2248
##     symmetry_worst fractal_dimension_worst
## 462         0.2641                 0.07427
## 213         0.1648                 0.05525
## 123         0.3222                 0.08009
```

Représentation des individus labelisés sur les plans principaux

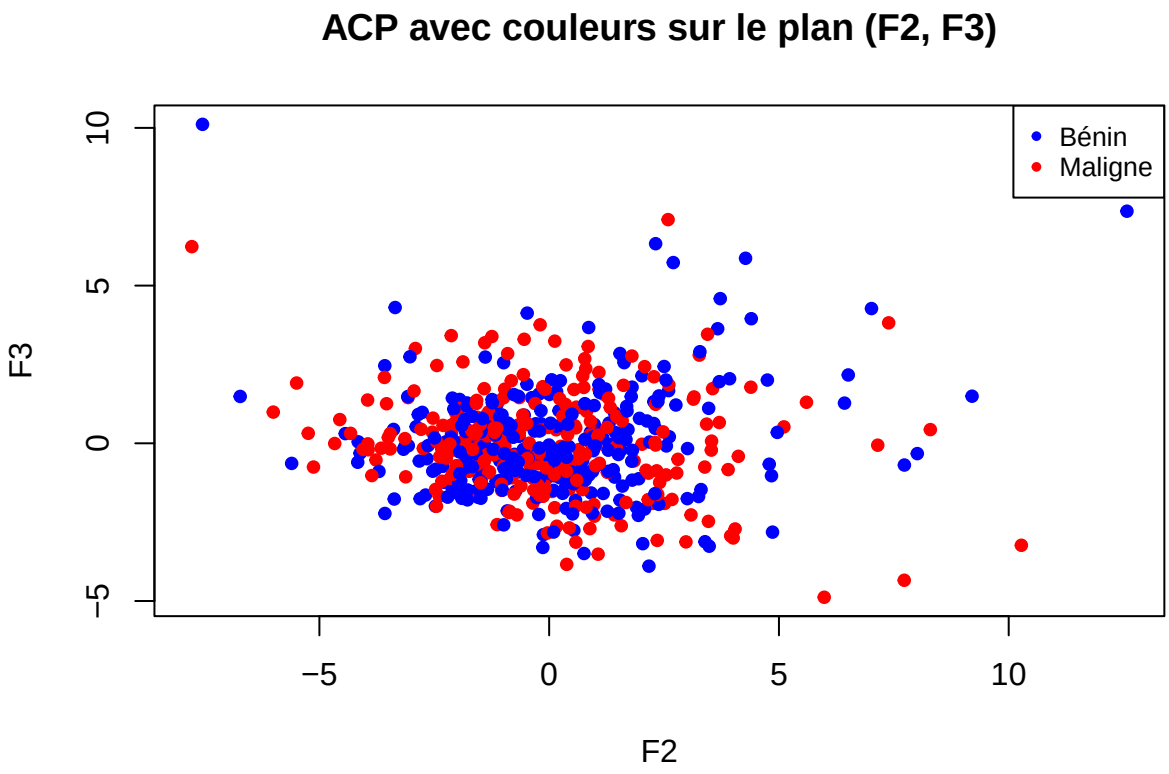
```
plot(res.pca$ind$coord[,1], res.pca$ind$coord[,2], col = c("blue", "red"), pch = 20, main = "ACP avec c
legend("topright", legend = c("Bénin", "Maligne"), col = c("blue", "red"), pch = 20, cex = 0.8)
```




```
plot(res.pca$ind$coord[,1], res.pca$ind$coord[,3], col = c("blue", "red"), pch = 20, main = "ACP avec c
legend("topright", legend = c("Bénin", "Maligne"), col = c("blue", "red"), pch = 20, cex = 0.8)
```



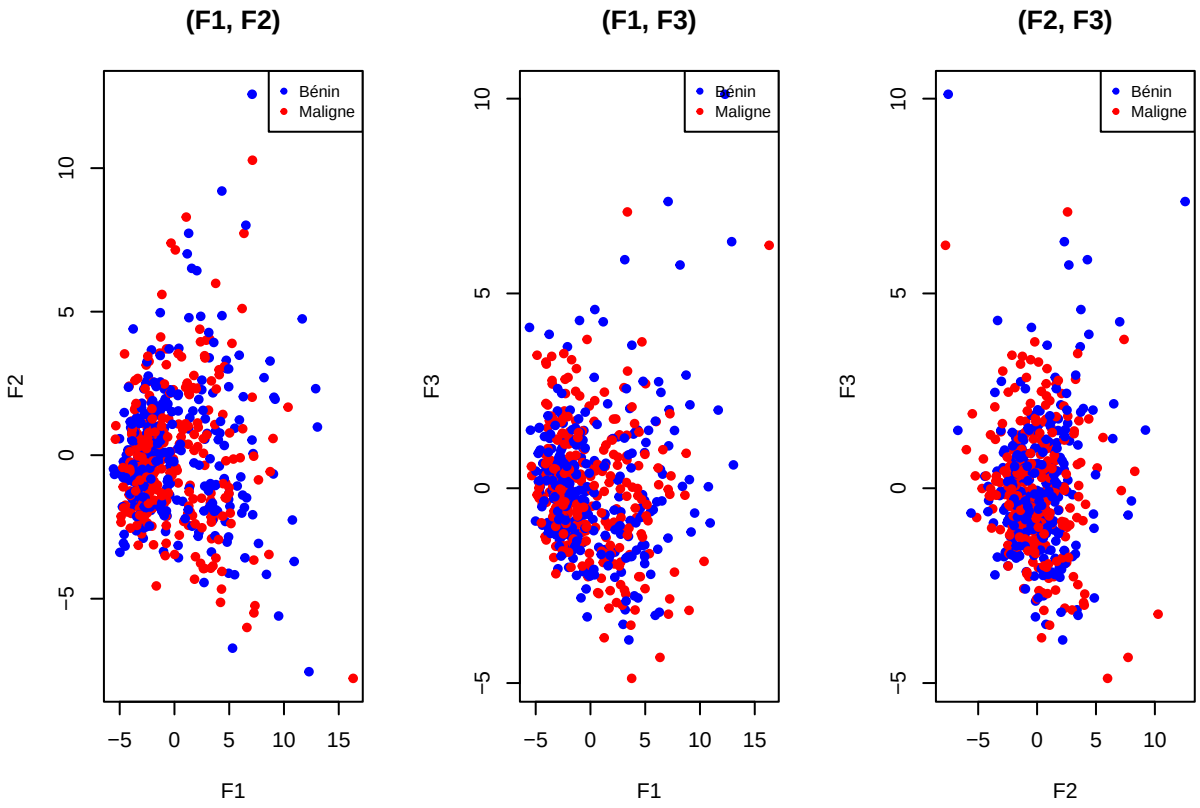

```
plot(res.pca$ind$coord[,2], res.pca$ind$coord[,3], col = c("blue", "red"), pch = 20, main = "ACP avec c
legend("topright", legend = c("Bénin", "Maligne"), col = c("blue", "red"), pch = 20, cex = 0.8)
```



On s'aperçoit qu'il est très difficile de distinguer les individus sur les plans principaux de l'ACP, d'où la nécessité de réaliser une AFD.

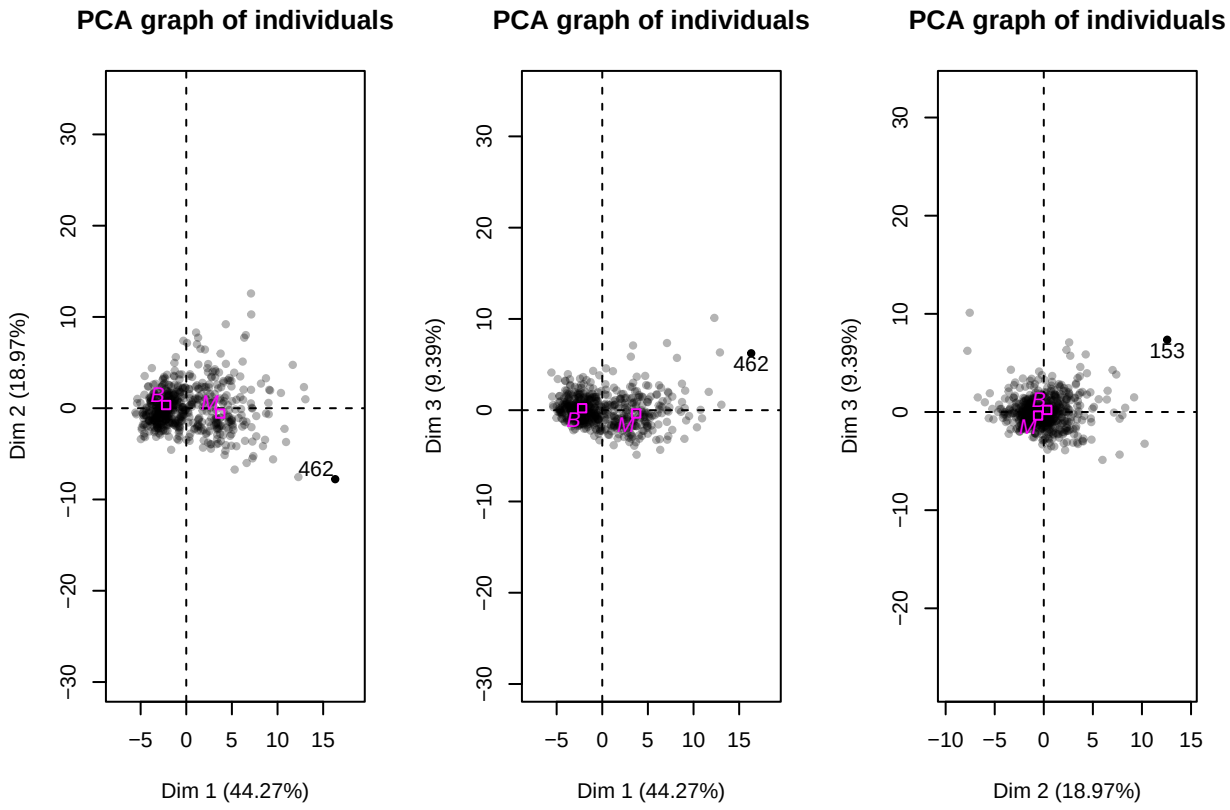
On peut représenter ces trois représentations ci-dessus sur un seul graphique.

```
par(mfrow=c(1,3))

plot(res.pca$ind$coord[,1], res.pca$ind$coord[,2], col = c("blue", "red"), pch = 20, main = "(F1, F2)",
legend("topright", legend = c("Bénin", "Maligne"), col = c("blue", "red"), pch = 20, cex = 0.8)

plot(res.pca$ind$coord[,1], res.pca$ind$coord[,3], col = c("blue", "red"), pch = 20, main = "(F1, F3)",
legend("topright", legend = c("Bénin", "Maligne"), col = c("blue", "red"), pch = 20, cex = 0.8)

plot(res.pca$ind$coord[,2], res.pca$ind$coord[,3], col = c("blue", "red"), pch = 20, main = "(F2, F3)",
legend("topright", legend = c("Bénin", "Maligne"), col = c("blue", "red"), pch = 20, cex = 0.8)
```



Représentation des catégories sur les plans principaux

```
par(mfrow=c(1,3))
plot(res.pca, graph= "classic", choix = "ind", cex = 1, select = "contrib 0")
plot(res.pca, graph= "classic", choix = "ind", cex = 1, select = "contrib 0", axes = c(1,3))
plot(res.pca, graph= "classic", choix = "ind", cex = 1, select = "contrib 0", axes = c(2,3))
```



On observe que l'axe 1 est le plus discriminant par rapport aux centres de gravité des groupes. En effet, les centres des catégories sont bien séparés le long de cet axe. Néanmoins, le manque de séparation visible sur les plans (F1, F2), (F1, F3) et (F2, F3) nous pousse à réaliser une AFD.

AFD - Analyse Factorielle discriminante

Lancement d'une AFD sur les données

Nous allons réaliser une AFD sur nos données avec la bibliothèque `MASS`.

```
library(MASS)
```

```
res.afd <- lda(data$diagnosis ~ ., data = data)
res.afd
```

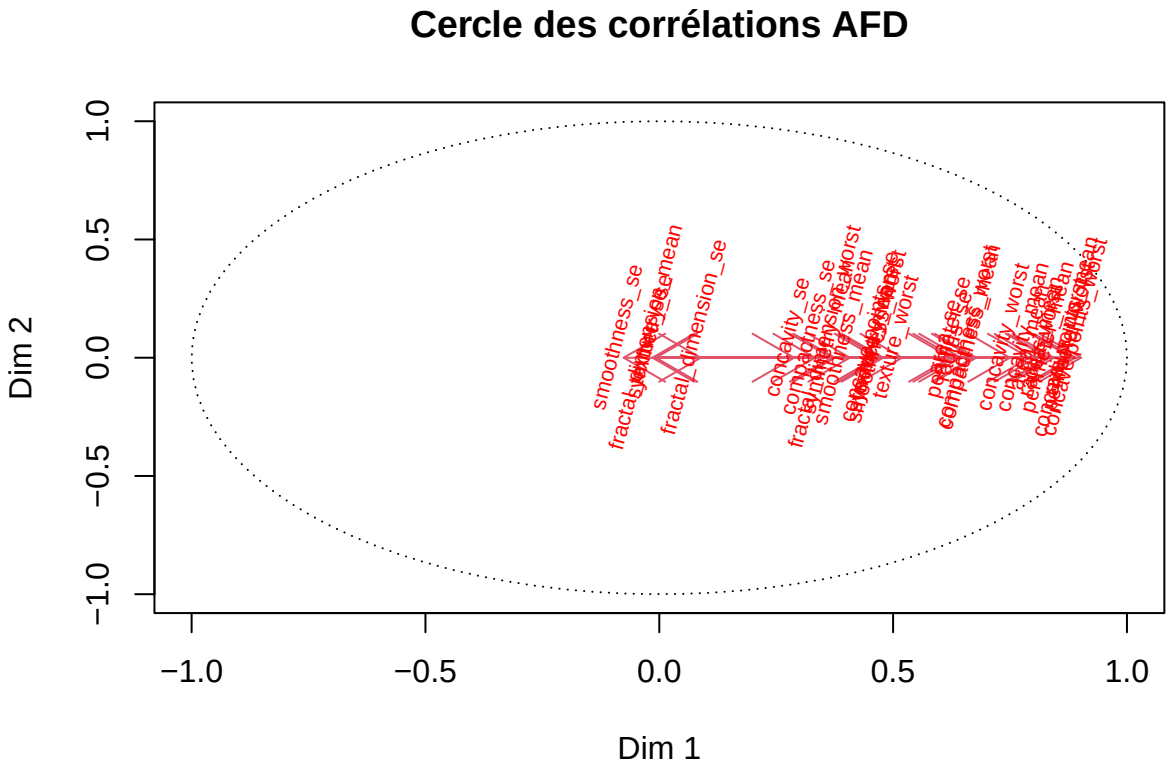
```
## Call:
## lda(data$diagnosis ~ ., data = data)
##
## Prior probabilities of groups:
##         B         M
## 0.6274165 0.3725835
##
## Group means:
##   radius_mean texture_mean perimeter_mean area_mean smoothness_mean
```

```
## B    12.14652     17.91476       78.07541  462.7902      0.09247765
## M    17.46283     21.60491      115.36538  978.3764      0.10289849
##   compactness_mean concavity_mean concave.points_mean symmetry_mean
## B       0.08008462     0.04605762          0.02571741      0.174186
## M       0.14518778     0.16077472          0.08799000      0.192909
##   fractal_dimension_mean radius_se texture_se perimeter_se  area_se
## B             0.06286739 0.2840824   1.220380     2.000321 21.13515
## M             0.06268009 0.6090825   1.210915     4.323929 72.67241
##   smoothness_se compactness_se concavity_se concave.points_se symmetry_se
## B   0.007195902     0.02143825   0.02599674       0.009857653  0.02058381
## M   0.006780094     0.03228117   0.04182401       0.015060472  0.02047240
##   fractal_dimension_se radius_worst texture_worst perimeter_worst area_worst
## B          0.003636051     13.37980      23.51507        87.00594   558.8994
## M          0.004062406     21.13481      29.31821       141.37033  1422.2863
##   smoothness_worst compactness_worst concavity_worst concave.points_worst
## B        0.1249595         0.1826725       0.1662377           0.07444434
## M        0.1448452         0.3748241       0.4506056           0.18223731
##   symmetry_worst fractal_dimension_worst
## B      0.2702459              0.07944207
## M      0.3234679              0.09152995
##
## Coefficients of linear discriminants:
##                                  LD1
## radius_mean              -1.075583600
## texture_mean              0.022450225
## perimeter_mean            0.117251982
## area_mean                 0.001569797
## smoothness_mean           0.418282533
## compactness_mean        -20.852775912
## concavity_mean            6.904756198
## concave.points_mean      10.578586272
## symmetry_mean             0.507284238
## fractal_dimension_mean    0.164280222
## radius_se                 2.148262164
## texture_se               -0.033380325
## perimeter_se             -0.111228320
## area_se                  -0.004559805
## smoothness_se            78.305030179
## compactness_se            0.320560148
## concavity_se            -17.609967822
## concave.points_se        52.195471457
## symmetry_se               8.383223501
## fractal_dimension_se    -35.296511336
## radius_worst              0.964016085
## texture_worst             0.035360398
## perimeter_worst          -0.012026798
## area_worst               -0.004994466
## smoothness_worst          2.681188528
## compactness_worst         0.331697102
## concavity_worst           1.882716394
## concave.points_worst      2.293242388
## symmetry_worst            2.749992654
## fractal_dimension_worst  21.255049570
```

Représentation des variables

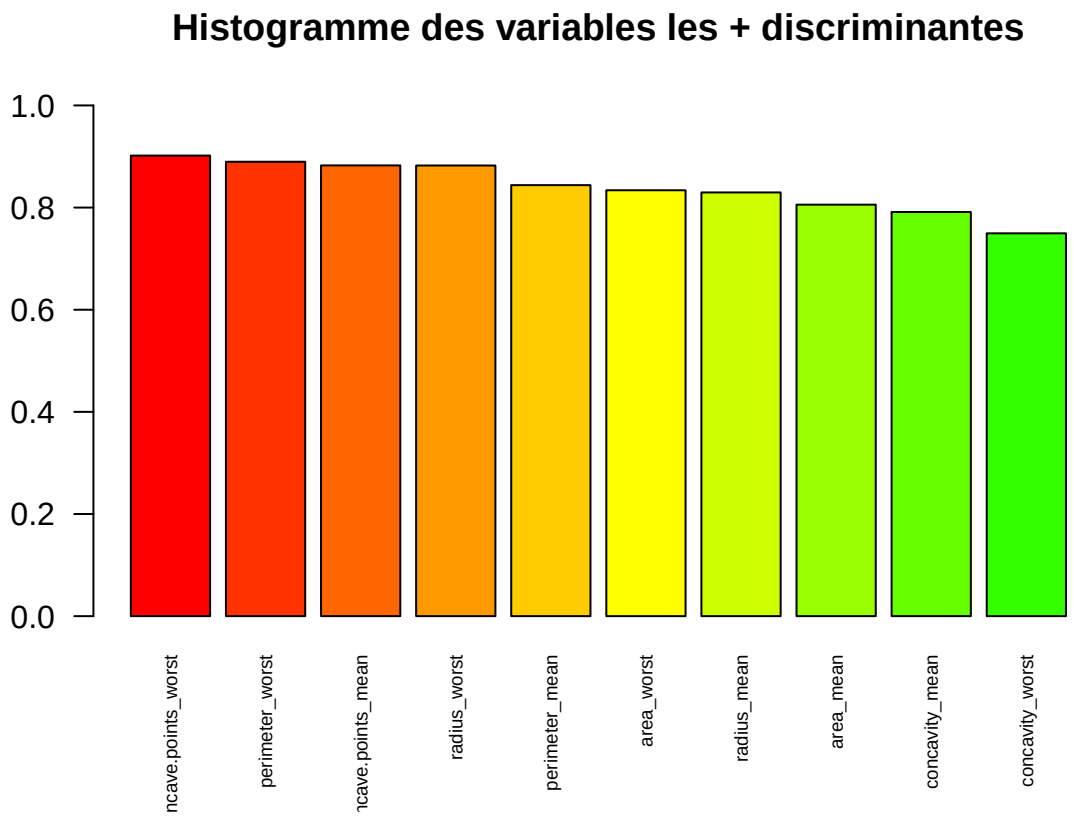
Pour tracer le cercle de corrélation, il nous faut les coordonnées des variables sur les axes discriminants. Or, comme il y a un seul axe, les coordonnées des variables sont les coefficients de l'axe discriminant.

```
F12 = predict(res.afd, prior=rep(1/2,2))$x
cercle_correlation=cor(data[,-1],F12)
cercle_correlation <- cbind(cercle_correlation, rep(0, nrow(cercle_correlation)))
a=seq(0,2* pi,length=100)
plot(cos(a), sin(a), type='l',lty=3,xlab='Dim 1', ylab='Dim 2',main="Cercle des corrélations AFD" )

arrows(0,0,cercle_correlation[,1],cercle_correlation[,2],col=2)
text(cercle_correlation,labels=colnames(data[,-1]), col = "red", cex = 0.7, pos = 3, offset = 0.5, srt =
```



On ne peut pas distinguer une variable en particulier qui influe sur la classification. En effet, le cercle manque un peu de visibilité. On peut faire un barplot des variables pour voir lesquelles sont les plus discriminantes.

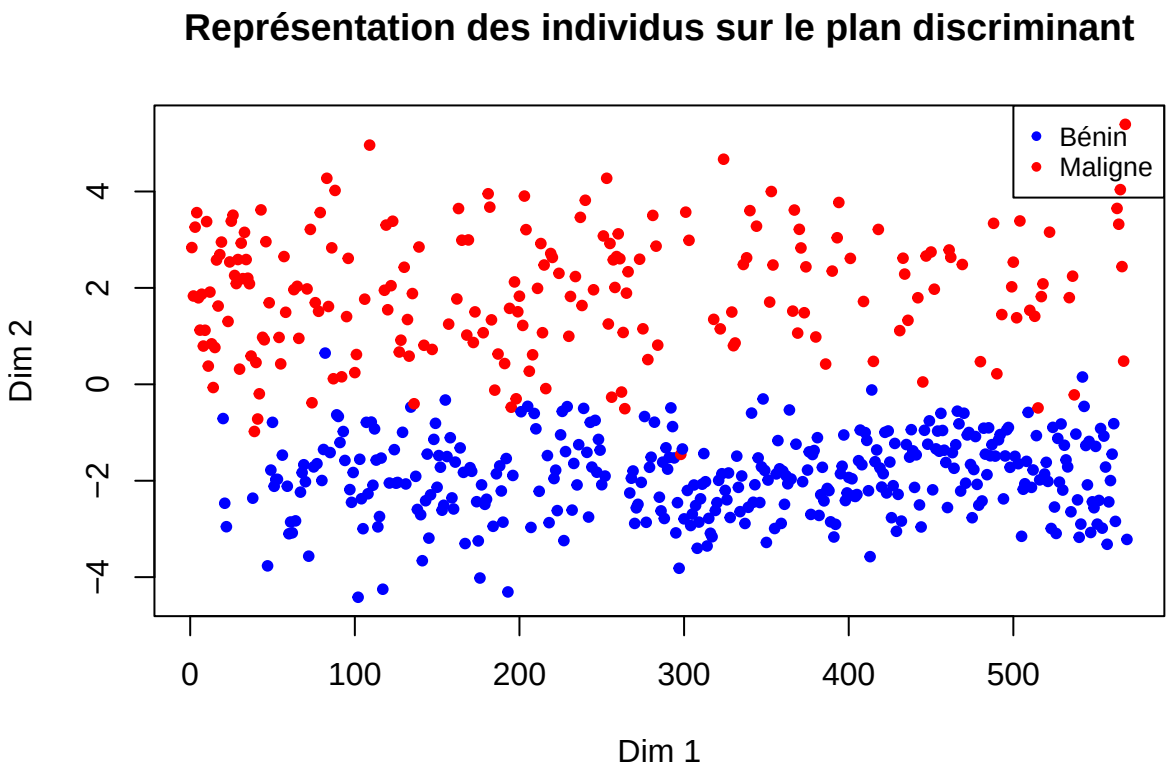
```
most_discriminant <- sort(abs(cercle_correlation[,1]), decreasing = TRUE)[1:10]
barplot(most_discriminant, names.arg=colnames(most_discriminant), las=2, col=rainbow(30), main="Histogra
```



Les variables les plus discriminantes sont les variables qui ont une influence sur l'axe discriminant. Parmi elles, on observe notamment les variables `concave.points_worst`, `perimeter.worst`, `concave.points_mean` et `radius_worst`.

Représentation des individus

```
plot(F12, col = c("blue", "red")[data$diagnosis], pch = 20, main = "Représentation des individus sur le
legend("topright", legend = c("Bénin", "Maligne"), col = c("blue", "red"), pch = 20, cex = 0.8)
```



On obtient une représentation des individus sur le plan discriminant. On observe que les individus sont bien séparés.

Conclusion

L'ACP sur les données a permis de mettre en évidence les variables qui contribuent le plus à la variance des données. L'AFD a permis de séparer les individus en fonction de leur diagnostic. Nous n'avons pas pu distinguer les variables les plus discriminantes sur le cercle de corrélation, mais nous avons pu les identifier grâce à un barplot. De plus amples recherches pourraient être faites sur les individus 462, 213 et 123 qui semblent être très influents sur les plans principaux de l'ACP.