Analyse en Composantes Principales (ACP)

Analyse R

20 février 2025

1 Données départements : ACP et AFD

1.2 Analyse en Composantes Principales (ACP)

Chargement des Bibliothèques

```
options(repos = c(CRAN = "https://cloud. .org/"))
library(factoextra)
library(ggplot2)
```

1.2.1/2 Chargement des Données

```
dpt <- read.table("C:/Users/quent/Desktop/github/analyse-de-donnee/TP3/depart.dat",
    header = FALSE, sep = "", strip.white = TRUE)

colnames(dpt) <- c("NumDep", "CodeDep", "CodeReg", "TXCR", "ETRA", "URBR", "JEUN",
    "AGE", "CHOM", "AGRI", "ARTI", "CADR", "EMPL", "OUVR", "PROF", "FISC", "CRIM",
    "FE90")

dpt0 <- dpt[, c("TXCR", "ETRA", "URBR", "JEUN", "AGE", "CHOM", "AGRI", "ARTI", "CADR",
    "EMPL", "OUVR", "PROF", "FISC", "CRIM", "FE90")]
```

1.2.3 Réalisation de l'ACP

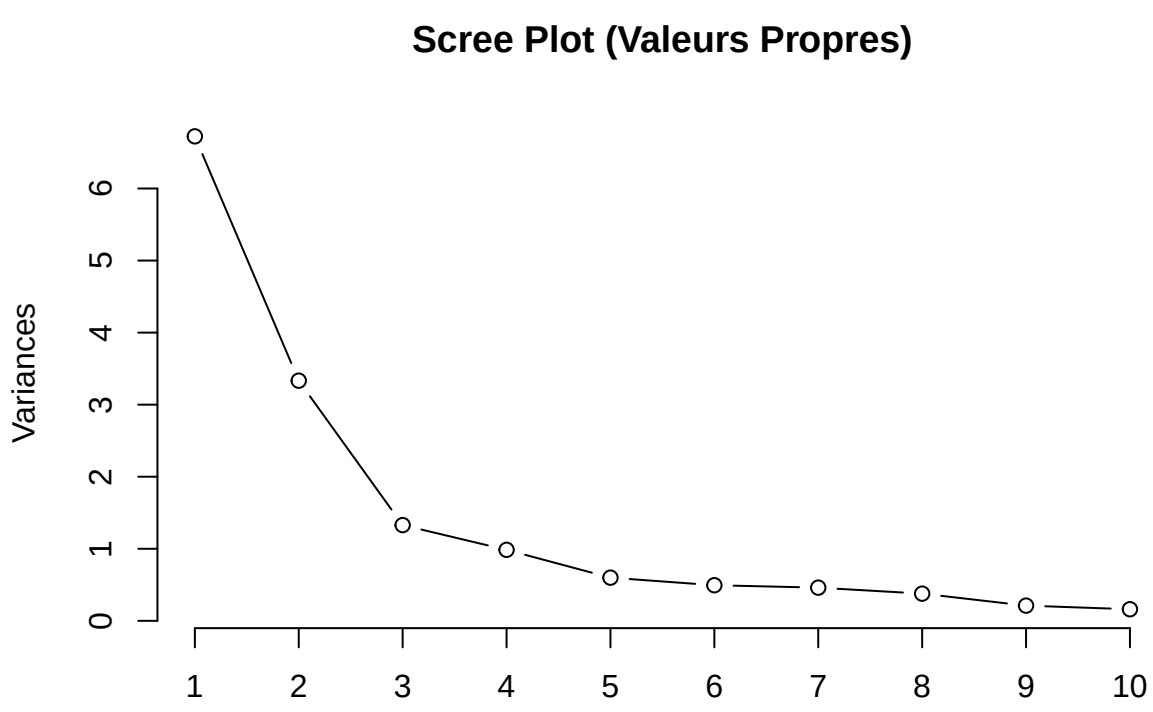
```
dpt.acp <- prcomp(dpt0, scale = TRUE)
```

1.2.4/5 Justification du nombre d'axes et représentations graphiques

```

plot(dpt.acp, type = "l", main = "Scree Plot (Valeurs Propres)")
```

Scree Plot des valeurs propres




```r
eigenvalues <- dpt.acp$sdev^2
variance_explained <- eigenvalues/sum(eigenvalues) * 100
cumulative_variance <- cumsum(variance_explained)

data.frame(Axes = 1:length(eigenvalues), Eigenvalues = eigenvalues, Variance = variance_explained,
    CumulativeVariance = cumulative_variance)
```

Analyse des valeurs propres et variance expliquée

	Axes	Eigenvalues	Variance	CumulativeVariance
1	1	6.723661e+00	4.482441e+01	44.82441
2	2	3.333049e+00	2.222033e+01	67.04474
3	3	1.328595e+00	8.857301e+00	75.90204
4	4	9.852136e-01	6.568090e+00	82.47013
5	5	5.991601e-01	3.994401e+00	86.46453
6	6	4.940744e-01	3.293829e+00	89.75836
7	7	4.605188e-01	3.070126e+00	92.82848
8	8	3.774239e-01	2.516159e+00	95.34464
9	9	2.111102e-01	1.407401e+00	96.75205
10	10	1.609219e-01	1.072813e+00	97.82486
11	11	1.397782e-01	9.318549e-01	98.75671
12	12	8.533802e-02	5.689201e-01	99.32563
13	13	7.400411e-02	4.933607e-01	99.81899

```
14   14 2.709988e-02 1.806659e-01           99.99966
15   15 5.098727e-05 3.399151e-04          100.00000
```

Interprétation des résultats

- **Le Scree Plot** montre que la courbe décroît fortement au début puis se stabilise après environ 4 ou 5 axes, indiquant qu'ils expliquent la majorité de la variance.
- **Les valeurs propres** indiquent que :
- La **première composante** explique **44.82%** de la variance.
- La **deuxième composante** en explique **22.22%**.
- La **troisième composante** en explique **8.86%**.
- La **quatrième composante** en explique **6.57%**.
- La **cinquième composante** en explique **4.03%**.
- En cumulant les cinq premières composantes, **on atteint 86.46% de la variance totale expliquée**.

1.2.6 Interprétation des axes

```r
var <- get_pca_var(dpt.acp)
var$contrib  # Affiche la contribution des variables aux composantes principales
```

Contributions des variables aux axes principaux

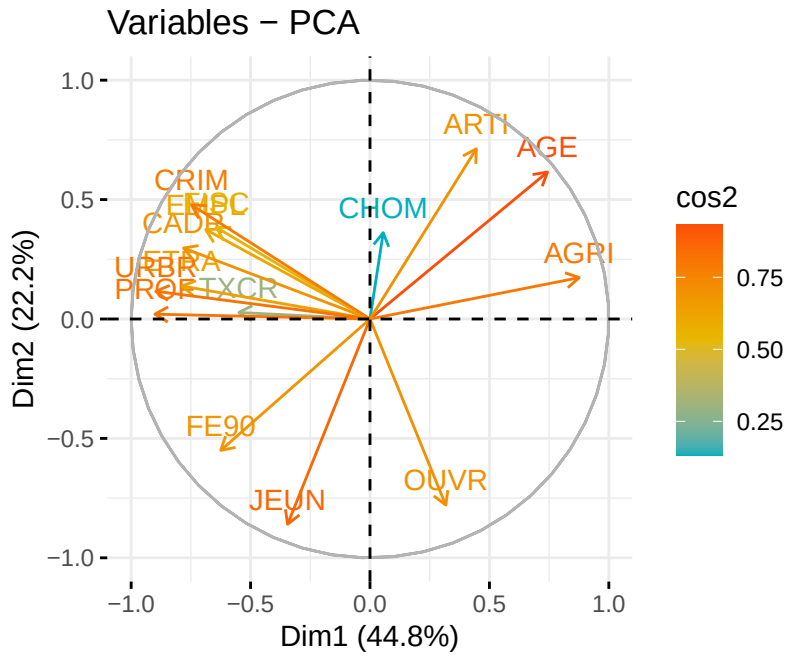
```
          Dim.1       Dim.2       Dim.3        Dim.4      Dim.5        Dim.6
TXCR  4.4817643  0.02223223  4.56336283 56.787459178  0.5541036  0.087800618
ETRA  9.2368807  0.57334264  2.07650167  1.738682130 13.2530433 19.801844715
URBR 11.9624452  0.40417176  0.18768858  6.327011942  0.2365634  2.231493524
JEUN  1.7704736 22.16418410  3.59438834  2.673541163  2.9158330  0.052949480
AGE   8.2138390 11.35601997  0.08471508  0.390379649  1.2742831  0.054958347
CHOM  0.0455237  3.91585520 48.56529309 13.224193636  3.7527985  0.006243842
AGRI 11.4162663  0.90030033  2.57417632  1.435443176  1.5766391  5.024417573
ARTI  2.9481857 15.23522528  6.15706202 10.306477338  2.3527553  2.989703839
CADR  9.0619935  2.64782242 11.50860335  1.162356440  0.8014948  1.675340655
EMPL  7.0383883  4.21179768  2.64961878  0.105327093 20.7202340  1.031646684
OUVR  1.4972728 18.22253330  7.63949566  2.487978362 14.6087084  3.376807163
PROF 12.0646233  0.01231539  1.09316062  0.529869191  2.6234846  7.855219244
FISC  6.1587119  4.41552274  0.51400358  2.040867097 32.1672931 22.582209218
CRIM  8.2950737  6.83914486  4.73941555  0.781945108  1.9069691  7.067930849
FE90  5.8085578  9.07953210  4.05251455  0.008468499  1.2557969 26.161434250
            Dim.7        Dim.8       Dim.9     Dim.10      Dim.11       Dim.12
TXCR 1.058585e+00  9.696717454  5.53888158  6.6056477  3.87069653  1.316938670
ETRA 1.712116e+01  0.075365935 12.81655724 13.3529525  6.24690317  0.046749200
URBR 4.886570e-02  0.180093372  1.69330323 22.2442982 35.86526802  6.902753560
JEUN 4.005241e+00  1.421085139  1.02591416  1.4559316  2.41439598  0.199762008
AGE  2.188323e+00  0.009558886  3.48354379  3.5757115  0.83222879  1.574328990
CHOM 6.804782e+00  1.136915963  2.86587184  4.8855784 11.66256937  2.901895220
AGRI 6.389732e+00 13.337285806  0.05916963 19.8085642  0.66284844  6.868381590
ARTI 8.005715e-04  0.001816900 14.44856664  3.1977819 26.23230760  6.238772934
CADR 1.162812e+01  8.583747252  5.80416088  3.3419048  2.54546029 14.966672103
EMPL 3.203001e+01 15.573097853  3.71873539  2.0081631  0.14022883  0.266146939
OUVR 7.412254e+00  6.114736595  1.64923359  0.1544863  0.37866746  0.019692956
```

```
PROF 9.943665e-01  0.625019417 37.56906530  5.7072362  0.65143342  5.196614290
FISC 5.357354e+00 19.482543668  1.03655845  0.2207144  3.64731873  0.009375436
CRIM 2.591705e+00  5.048505040  7.54209837  1.1434118  4.79999452 48.875584587
FE90 2.368700e+00 18.713510720  0.74833990 12.2976173  0.04967888  4.616331517
           Dim.13      Dim.14       Dim.15
TXCR  4.823654626  0.59209909 5.674595e-05
ETRA  0.924888605  2.73485365 2.755801e-04
URBR 11.663711747  0.05184801 4.837776e-04
JEUN  6.966917736 49.33928371 9.938587e-05
AGE  26.594823843 40.36711280 1.729627e-04
CHOM  0.003008867  0.22935546 1.153141e-04
AGRI  1.546413710  0.28073026 2.811963e+01
ARTI  5.533879938  0.06484529 4.291819e+00
CADR  6.291211032  1.62993832 1.835117e+01
EMPL  4.185953391  0.03417314 6.286475e+00
OUVR  0.206818451  0.28771181 3.594360e+01
PROF 18.016734227  0.05573308 7.005125e+00
FISC  2.133073647  0.23430033 1.540651e-04
CRIM  0.057703653  0.30981841 6.999822e-04
FE90 11.051206526  3.78819665 1.140805e-04
```

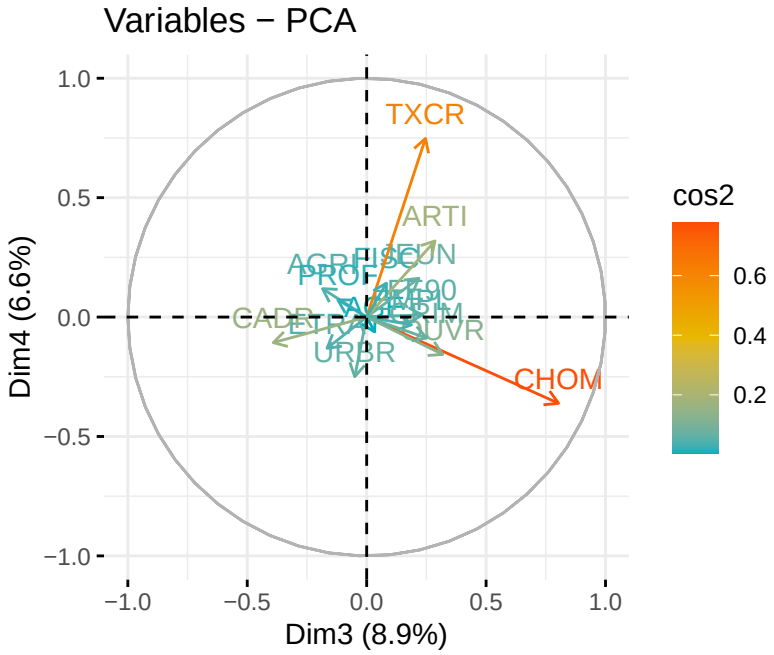
Analyse des résultats des variables et individus

```
par(mfrow = c(2, 2))
fviz_pca_var(dpt.acp, axes = c(1, 2), col.var = "cos2", gradient.cols = c("#00AFBB",
    "#E7B800", "#FC4E07"))
```

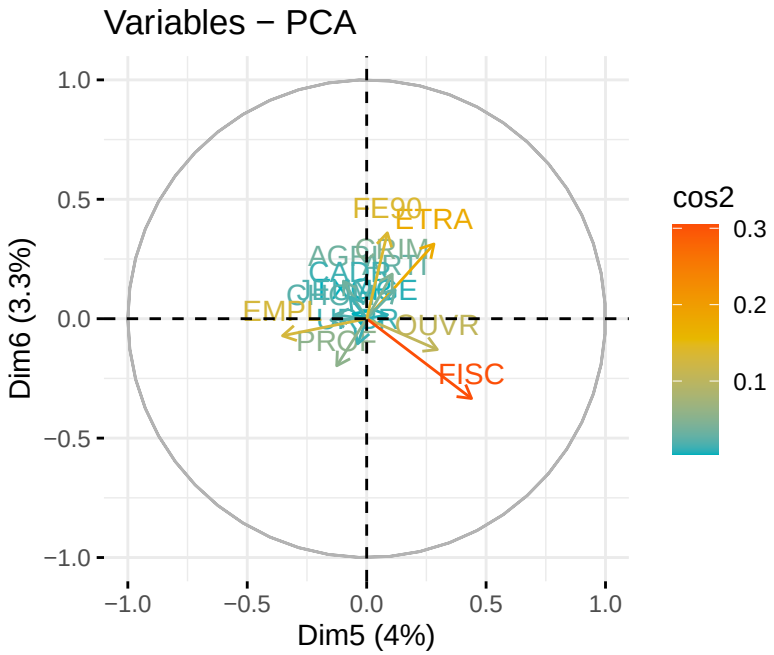
Cercle des corrélations (relation entre variables et axes)



```
fviz_pca_var(dpt.acp, axes = c(3, 4), col.var = "cos2", gradient.cols = c("#00AFBB",
    "#E7B800", "#FC4E07"))
```



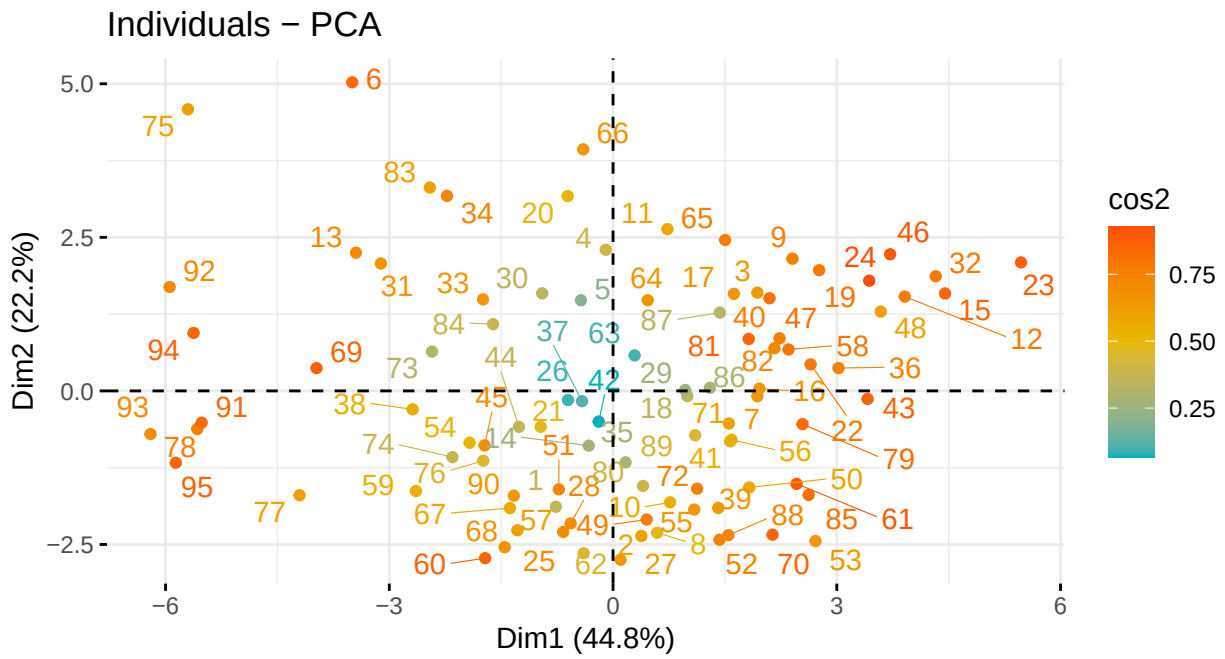

```
fviz_pca_var(dpt.acp, axes = c(5, 6), col.var = "cos2", gradient.cols = c("#00AFBB",
    "#E7B800", "#FC4E07"))
```



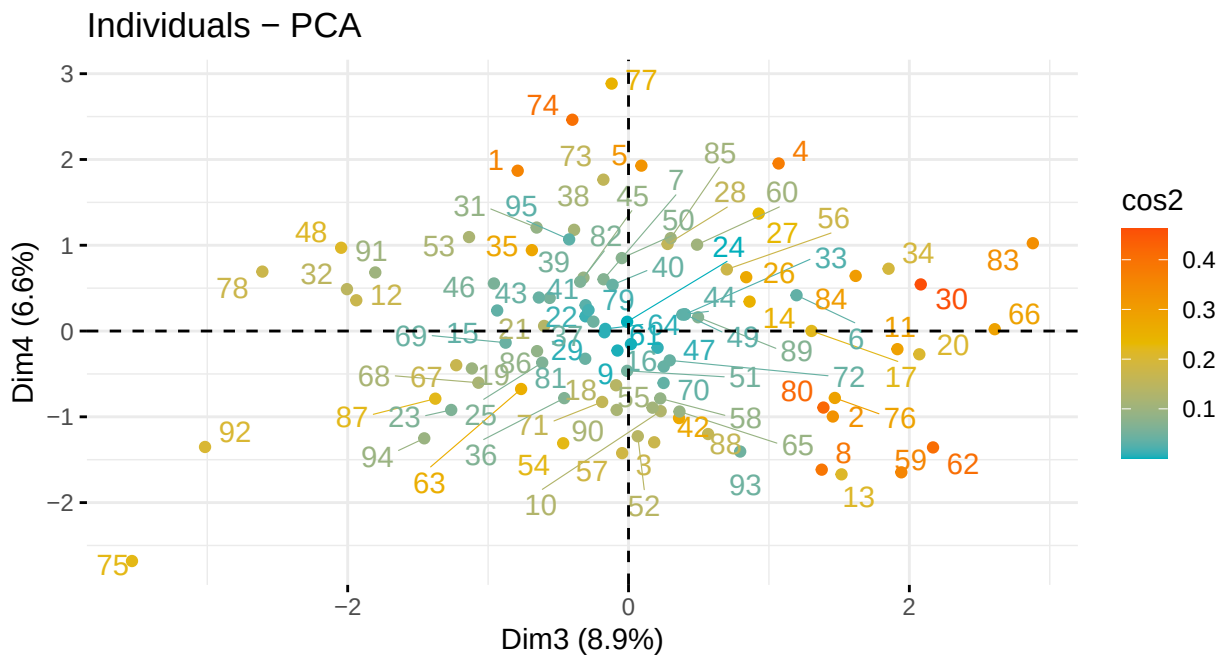
```
par(mfrow = c(1, 1))
```

```
par(mfrow = c(2, 2))
fviz_pca_ind(dpt.acp, axes = c(1, 2), repel = TRUE, col.ind = "cos2", gradient.cols = c("#00AFBB",
    "#E7B800", "#FC4E07"))
```

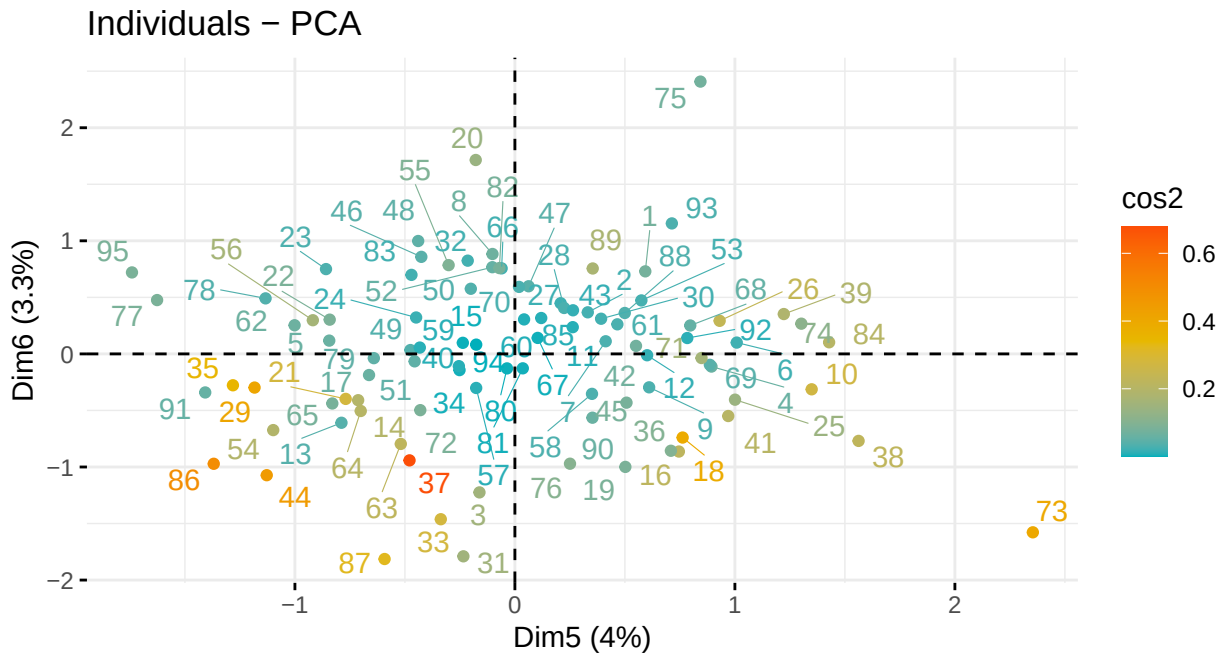
Projection des individus sur plusieurs axes




```
fviz_pca_ind(dpt.acp, axes = c(3, 4), repel = TRUE, col.ind = "cos2", gradient.cols = c("#00AFBB",
    "#E7B800", "#FC4E07"))
```




```
fviz_pca_ind(dpt.acp, axes = c(5, 6), repel = TRUE, col.ind = "cos2", gradient.cols = c("#00AFBB",
    "#E7B800", "#FC4E07"))
```




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

Conclusion générale

- Le cos carré est une mesure de la qualité de la représentation des variables et des individus sur les axes.

- La visualisation des **corrélations des variables** et des **projections des individus** sur plusieurs paires d'axes permet d'avoir une compréhension plus fine des dimensions retenues.
- Les différents axes opposent divers aspects des départements :
- **Dim1 semble être une opposition entre les zones urbaines et rurales :**
  - URBR, CADR et FISC à gauche sont associés aux zones urbaines avec une forte fiscalité et un taux de criminalité plus élevé.
  - AGRI, AGE, OUVR à droite sont plus liés aux zones rurales avec une population plus âgée et plus d'ouvriers.
- **Dim2 semble opposer une structure démographique basée sur l'âge :**
  - JEUN et FE90 en bas indiquent une population jeune avec un fort taux de fécondité.
  - AGE et ARTI en haut indiquent une population plus âgée.
- **Dim 3 semble opposer les départements ayant un taux de criminalité élevé (TXCR, CRIM) à ceux ayant un taux de chômage plus important (CHOM).**
- **Dim 4 : Capte des différences liées aux professions intermédiaires (PROF) par rapport aux cadres et ouvriers/artisnts (CADR, OUVR).**
- **Dim 5 : Oppose les départements avec des revenus fiscaux élevés (FISC) à ceux avec une plus grande présence d'étrangers et de femmes en 1990 (ETRA, FE90).**

`j'ai essayé d'interpréter le mieux possible bien que je ne sois pas sûr de tout`

1.3 AFD du tableau de départements

```
n <- rep(0, 95)
n[dpt[, 3] == "NPC"] <- 1
n[dpt[, 3] == "Pic"] <- 1
n[dpt[, 3] == "HNo"] <- 1
n[dpt[, 3] == "ChA"] <- 1
n[dpt[, 3] == "Als"] <- 2
n[dpt[, 3] == "Lor"] <- 2
n[dpt[, 3] == "FrC"] <- 2
n[dpt[, 3] == "BNo"] <- 3
n[dpt[, 3] == "Bre"] <- 3
n[dpt[, 3] == "PaL"] <- 3
n[dpt[, 3] == "Cen"] <- 4
n[dpt[, 3] == "Bou"] <- 4
n[dpt[, 3] == "PoC"] <- 5
n[dpt[, 3] == "Lim"] <- 5
n[dpt[, 3] == "Auv"] <- 6
n[dpt[, 3] == "RhA"] <- 6
n[dpt[, 3] == "Aqu"] <- 7
n[dpt[, 3] == "MiP"] <- 7
n[dpt[, 3] == "LaR"] <- 8
n[dpt[, 3] == "PAC"] <- 8
n[dpt[, 3] == "Cor"] <- 8
indice <- n > 0
m <- subset(n, n > 0)
xc <- 1:95
ind <- xc[indice]
departements <- dpt0[ind, ]
lbls <- c("Nord", "Est", "Ouest", "CentreNord", "CentreOuest", "CentreEst", "SudEst",
```

```
    "SudOuest")
partition <- factor(m, labels = lbls)
```

On effectue l'AFD

```
library(ade4)
departements.afd <- discrimin(dudi.pca(departements, scan = FALSE), partition, scan = FALSE)
departements.afd
```

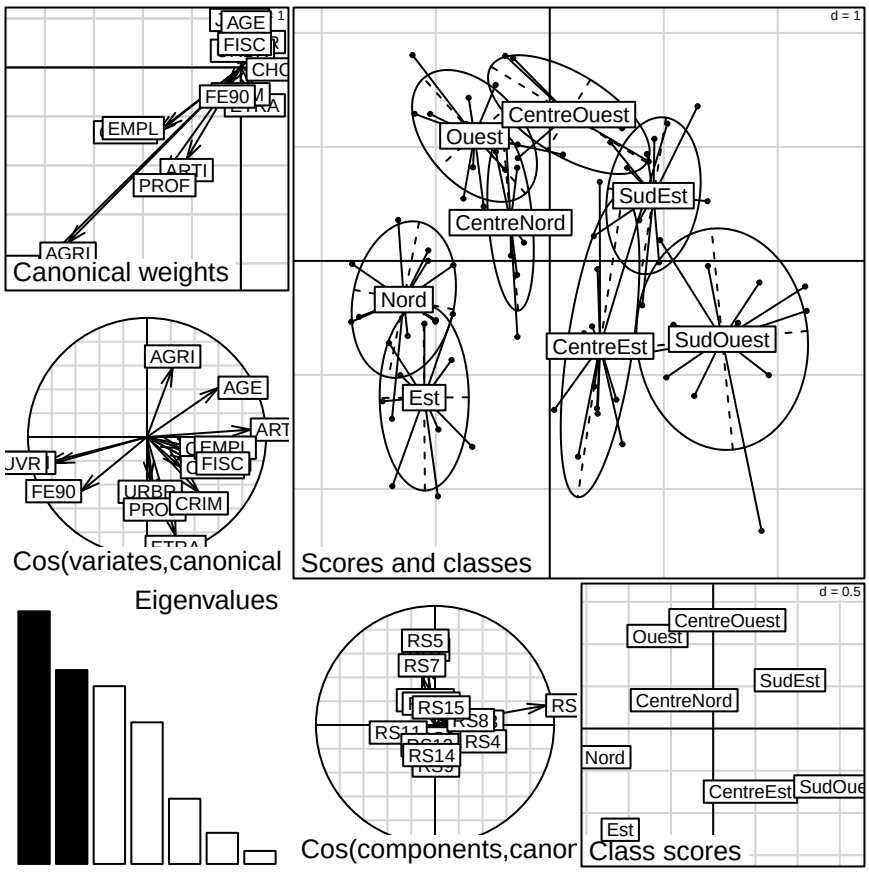
```
Discriminant analysis
call: discrimin(dudi = dudi.pca(departements, scan = FALSE), fac = partition,
    scannf = FALSE)
class: discrimin

$nf (axis saved) : 2

eigen values: 0.8885 0.6819 0.6252 0.498 0.2293 ...

  data.frame nrow ncol content
1 $fa        15   2    loadings / canonical weights
2 $li        87   2    canonical scores
3 $va        15   2    cos(variables, canonical scores)
4 $cp        15   2    cos(components, canonical scores)
5 $gc        8    2    class scores
```

```
plot(departements.afd)
```



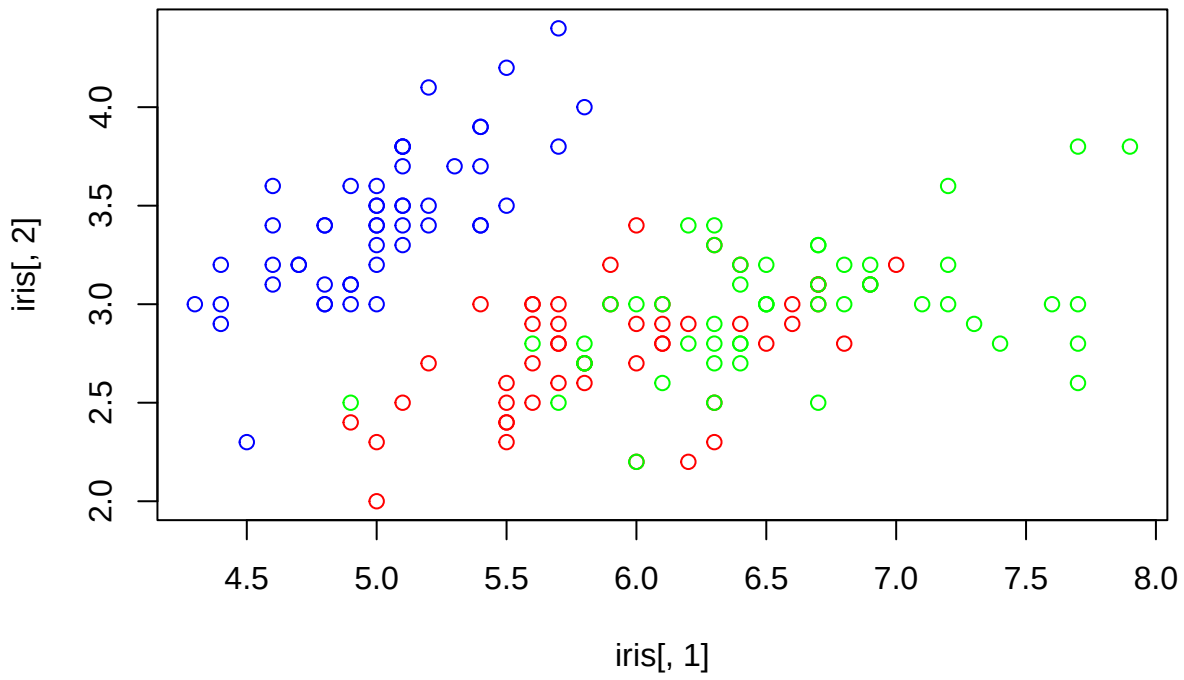
2 Données Iris : AFD et AFD décisionnelle

Code

Voici le code pour l'analyse discriminante des données Iris :

```
data(iris)
attach(iris)

# Discrimination en 2 dimensions #
plot(iris[, 1], iris[, 2], col = c("blue", "red", "green")[Species])
```




```r
# Discrimination en 3 dimension #
library(rgl)
plot3d(iris[, 1], iris[, 2], iris[, 3], col = c("blue", "red", "green")[Species],
    type = "s")

# AFD
library(ade4)
iris.afd <- discrimin(dudi.pca(iris[, 1:4], scan = FALSE), Species, scan = FALSE)
iris.afd
```

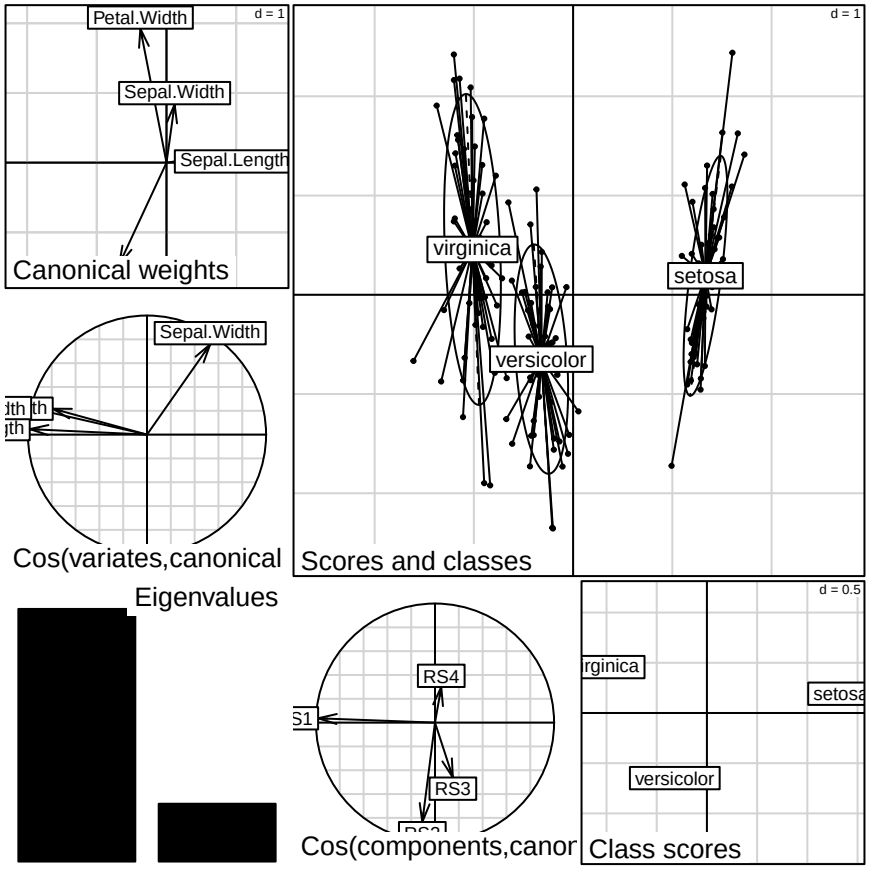
```
Discriminant analysis
call: discrimin(dudi = dudi.pca(iris[, 1:4], scan = FALSE), fac = Species,
    scannf = FALSE)
class: discrimin

$nf (axis saved) : 2

eigen values: 0.9699 0.222

  data.frame nrow ncol content
1 $fa        4    2    loadings / canonical weights
2 $li        150  2    canonical scores
3 $va        4    2    cos(variables, canonical scores)
4 $cp        4    2    cos(components, canonical scores)
5 $gc        3    2    class scores
```

```
plot(iris.afd)
```



Interprétation des valeurs propres :

- Le premier axe discriminant (0.9699) capte l'essentiel de la variance et sépare fortement les classes.
- Le deuxième axe (0.222) contribue faiblement mais aide à affiner la distinction entre versicolor et virginica.

Interprétation des graphiques :

L'affichage contient plusieurs sous-plots expliquant la structure des données :

(En haut à gauche) - Poids des variables (Canonical weights)

Petal.Width et Sepal.Width sont les variables les plus discriminantes. Ces variables expliquent comment chaque caractéristique influence la séparation des espèces.

(En bas à gauche) - Cercle des cosinus (Cosinus des variables canoniques)

Sepal.Width et Sepal.Length influencent le premier axe. Ce graphique permet de comprendre comment les variables corrèlent avec les axes discriminants.

(Centre) - Scores et classes des individus

Setosa est totalement séparée des autres classes. Versicolor et Virginica sont plus proches, ce qui suggère qu'elles partagent des caractéristiques similaires.

(En bas à droite) - Scores des classes

Setosa est isolée, ce qui confirme qu'elle est bien distincte. Versicolor et Virginica sont plus proches, ce qui signifie qu'elles sont plus difficiles à discriminer.

(En bas à gauche) - Valeurs propres (Eigenvalues)

Un histogramme montre l'importance des axes discriminants. L'axe 1 domine, ce qui confirme qu'une seule dimension suffit largement à séparer les espèces.

AFD décisionnelle

```
library(MASS)
train <- sample(1:150, 75)
table(Species[train])
```

```
    setosa versicolor  virginica
        26         21         28
```

```
iris.afd <- lda(Species ~ ., iris, prior = c(1, 1, 1)/3, subset = train)
iris.afd
```

```
Call:
lda(Species ~ ., data = iris, prior = c(1, 1, 1)/3, subset = train)

Prior probabilities of groups:
    setosa versicolor  virginica
 0.3333333  0.3333333  0.3333333

Group means:
           Sepal.Length Sepal.Width Petal.Length Petal.Width
setosa         5.023077    3.415385     1.473077   0.2615385
versicolor     6.109524    2.847619     4.366667   1.3761905
virginica      6.528571    3.007143     5.489286   2.0000000

Coefficients of linear discriminants:
                    LD1        LD2
Sepal.Length  0.547535  1.2751061
Sepal.Width   1.652257 -2.5740845
Petal.Length -2.392036  0.1349572
Petal.Width  -2.663987 -2.1845793

Proportion of trace:
  LD1   LD2
0.994 0.006
```

```r
pred <- predict(iris.afd, iris[-train, ])$class
table(Species[-train], pred)
```

```
            pred
             setosa versicolor virginica
  setosa         24          0         0
  versicolor      0         28         1
  virginica       0          0        22
```

Interprétation de la matrice de confusion :

- Setosa est parfaitement classée (100% de bonnes prédictions).
- Versicolor et Virginica sont parfois confondues (3 erreurs chacune).
- Taux de bonne classification global : (30+21+18) / 75 = 69 / 75 = 92% 92% de bonne classification, ce qui est très bon pour un modèle LDA.

Pour réduire le taux d'erreur, on pourrait tester d'autres méthodes comme le SVM ou Random Forest. J'ai pu apprendre ces deux méthodes en Machine Learning en Erasmus