Cours Analyse de Données

L3 EURIA - UBO

Pierre Ailliot

Année secolaire 2020-2021

L'analyse de données est une branche de la statistique qui recouvre un certains nombre des méthodes qui ont pour objectif de faire ressortir de l'information contenue dans un jeu de données. Dans le cadre de cours, nous allons nous intéresser principalement à deux familles de méthodes : l'analyse en composante principale (ACP) et la classification non-supervisée.

ACP : l'ACP fait partie des méthodes d'**analyse factorielle**. Les méthodes d'analyse factorielles ont pour objectif de projeter des données multivariées sur un espace de petite dimension. L'espace sur lequel on projette est choisi de manière 'optimale' pour perdre le moins possible d'information.

Classification non-supervisée: les méthodes de classification non-supervisées ou "automatiques" cherchent à regrouper un ensemble d'individus en groupes 'homogènes'.

On pourra consulter les liens suivants

https://fr.wikipedia.org/wiki/Analyse_des_donn%C3%A9es

http://wikistat.fr/

pour un panorama des méthodes d'analyse de données qui existent.

L'utilisation de R occupe une part importante dans ce cours (et donc aussi dans l'évaluation finale).

1 Quelques éléments de statistiques descriptives univariées et bivariées

La première étape pour analyser un jeu de données consiste à faire des statistiques descriptives.

Pour illustrer ce cours, nous allons considérer un jeu de données qui décrit les températures mensuelles moyennes dans plusieurs villes Françaises. Le jeux de données, ainsi que les coordonnées géographique des villes, sont disponibles sur la page web du cours

https://pagesperso.univ-brest.fr/~ailliot/L3EURIA.html

Les températures sont données en dixième de degré (il faut donc diviser les valeurs par 10 pour obtenir des degrés Celsius).

Exercice 1 *Importer les données avec R puis faire des statistiques descriptives du jeu de données. L'objectif est de résumer l'information contenue dans les données en utilisant des résumés numériques et graphiques adaptés.*

Correction de l'exerice 1

```
X=data_temp <- read.csv("~/public_html/doc_cours/L3EURIA/AD/data_temp.txt",
                        sep= ' ',row.names = 1 )/10
#modifier le chemin d'accès
head(X)
```

```
##           Jan Feb  Mar  Apr  May  Jun  Jul  Aug  Sep  Oct  Nov Dec
## Angouleme 4.2 4.9  7.9 10.4 13.6 17.0 18.7 18.4 16.1 11.7  7.6 4.9
## Angers    4.6 5.4  8.9 11.3 14.5 17.2 19.5 19.4 16.9 12.5  8.1 5.3
## Besancon  1.1 2.2  6.4  9.7 13.6 16.9 18.7 18.3 15.5 10.4  5.7 2.0
```

```
## Biarritz  7.6 8.0 10.8 12.0 14.7 17.8 19.7 19.9 18.5 14.8 10.9 8.2
## Bordeaux  5.6 6.6 10.3 12.8 15.8 19.3 20.9 21.0 18.6 13.8  9.1 6.2
## Brest     6.1 5.8  7.8  9.2 11.6 14.4 15.6 16.0 14.7 12.0  9.0 7.0
```

Individus et variables Chaque individu (ici les villes) est caractérisé par les valeurs prises par les variables (ici les températures). Dans le cadre de ce cours et quand on analyse des un jeu de données avec R, **les individus sont les lignes et les variables sont les colonnes**. Dans la suite du cours, on note

$$X = \begin{pmatrix} x_{1,1} & \cdots & x_{1,p} \\ \vdots & \ddots & \vdots \\ x_{n,1} & \cdots & n_{n,p} \end{pmatrix}$$

le tableau des n observations des p variables. On notera $X_{i,.}$ la ième ligne du tableau (c'est à dire le ième individu) et $X_{.,j}$ la jème colonne du tableau (c'est à dire la jème variable). Dans le cadre de ce cours, nous nous intéressons uniquement aux variables **quantitatives** (par opposition aux variables qualitatives ou ordinales).

```
dim(X)  #dimension du tableau de données
```

```
## [1] 30 12
```

```
str(X)  #permet de vérifier la nature des variables
```

```
## 'data.frame':    30 obs. of  12 variables:
##  $ Jan: num  4.2 4.6 1.1 7.6 5.6 6.1 2.6 1.3 1.5 2.4 ...
##  $ Feb: num  4.9 5.4 2.2 8 6.6 5.8 3.7 2.6 3.2 2.9 ...
##  $ Mar: num  7.9 8.9 6.4 10.8 10.3 7.8 7.5 6.9 7.7 6 ...
##  $ Apr: num  10.4 11.3 9.7 12 12.8 9.2 10.3 10.4 10.6 8.9 ...
##  $ May: num  13.6 14.5 13.6 14.7 15.8 11.6 13.8 14.3 14.5 12.4 ...
##  $ Jun: num  17 17.2 16.9 17.8 19.3 14.4 17.3 17.7 17.8 15.3 ...
##  $ Jul: num  18.7 19.5 18.7 19.7 20.9 15.6 19.4 19.6 20.1 17.1 ...
##  $ Aug: num  18.4 19.4 18.3 19.9 21 16 19.1 19 19.5 17.1 ...
##  $ Sep: num  16.1 16.9 15.5 18.5 18.6 14.7 16.2 15.9 16.7 14.7 ...
##  $ Oct: num  11.7 12.5 10.4 14.8 13.8 12 11.2 10.5 11.4 10.4 ...
##  $ Nov: num  7.6 8.1 5.7 10.9 9.1 9 6.6 5.7 6.5 6.1 ...
##  $ Dec: num  4.9 5.3 2 8.2 6.2 7 3.6 2.1 2.3 3.5 ...
```

Le jeu de données considéré a $n = 30$ individus et $p = 12$ variables. Toutes les variables sont quantitatives (variable de type 'num').

Résumés numériques pour les variables quantitatives. On peut calculer des **résumés numériques** pour les différentes variables. Pour les variables quantitatives, on considère généralement

- $\bar{x}_j = \frac{1}{n}\sum_{i=1}^{n} x_{i,j}$ la **moyenne empirique** de la $j^{ème}$ variable.
- $v_j = var(X_{.,j}) = \frac{1}{n}\sum_{i=1}^{n}(x_{i,j} - \bar{x}_j)^2$ la **variance empirique** de la $j^{ème}$ variable.
- $s_j = \sqrt{v_j}$ l'**écart-type empirique** de la $j^{ème}$ variable.

```
moyenne=apply(X,2,mean)  #calcul de la moyenne
variance=apply(X^2,2,mean)-apply(X,2,mean)^2  #calcul de la variance
#on peut utiliser la fonction Kable pour faire des tableaux avec RMarkdown

rbind(moyenne,variance)
```

```
##                Jan      Feb      Mar       Apr       May       Jun       Jul
## moyenne  3.920000 4.766667 8.126667 10.880000 14.343333 17.706667 19.706667
## variance 4.778267 4.252889 2.883956  2.112267  1.999789  2.925289  4.118622
##                Aug       Sep       Oct      Nov      Dec
## moyenne  19.400000 16.843333 12.206667 7.850000 4.796667
## variance  3.821333  3.593122  3.941289 4.051833 4.669656
```

```
#Attention : la fonction var calcule l'estimateur sans biais avec n-1 au dénominateur
#apply(X,2,var)
#nrow(X)*v/(nrow(X)-1)
```

On voit ainsi que le mois le pus chaud est juillet, le plus froid est janvier. Le mois avec le plus de variabilité entre les villes est janvier. D'autres statistiques peuvent être calculées (médiane, distance interquartile,...).

```
summary(X)  #la fonction summary permet d'afficher les quantiles
```

```
##       Jan             Feb             Mar             Apr       
##  Min.   :0.400   Min.   :1.500   Min.   : 5.500   Min.   : 8.90  
##  1st Qu.:2.175   1st Qu.:3.225   1st Qu.: 6.900   1st Qu.: 9.80  
##  Median :3.450   Median :4.250   Median : 7.700   Median :10.50  
##  Mean   :3.920   Mean   :4.767   Mean   : 8.127   Mean   :10.88  
##  3rd Qu.:5.575   3rd Qu.:6.400   3rd Qu.: 9.725   3rd Qu.:11.90  
##  Max.   :8.600   Max.   :9.100   Max.   :11.300   Max.   :13.90  
##       May             Jun             Jul             Aug       
##  Min.   :11.60   Min.   :14.40   Min.   :15.60   Min.   :16.00  
##  1st Qu.:13.32   1st Qu.:16.65   1st Qu.:18.40   1st Qu.:17.98  
##  Median :13.95   Median :17.25   Median :19.20   Median :18.75  
##  Mean   :14.34   Mean   :17.71   Mean   :19.71   Mean   :19.40  
##  3rd Qu.:14.90   3rd Qu.:18.65   3rd Qu.:20.85   3rd Qu.:20.70  
##  Max.   :17.10   Max.   :21.10   Max.   :23.80   Max.   :23.30  
##       Sep             Oct             Nov             Dec       
##  Min.   :14.70   Min.   : 9.40   Min.   : 4.900   Min.   :1.300  
##  1st Qu.:15.35   1st Qu.:10.75   1st Qu.: 6.525   1st Qu.:3.175  
##  Median :16.15   Median :11.50   Median : 7.150   Median :4.300  
##  Mean   :16.84   Mean   :12.21   Mean   : 7.850   Mean   :4.797  
##  3rd Qu.:18.45   3rd Qu.:13.68   3rd Qu.: 9.075   3rd Qu.:6.425  
##  Max.   :20.50   Max.   :16.50   Max.   :12.600   Max.   :9.700
```

Résumés numériques de la dépendance entre deux variables quantitatives. Afin de résumer la relation qui existe entre deux variables quantitatives, on considère généralement la covariance et la corrélation linéaire définies ci-dessous

- $v_{j,k} = cov(X_{.,j}, X_{.,k}) = \frac{1}{n}\sum_{i=1}^{n}(x_{i,j} - \bar{x}_j)(x_{i,k} - \bar{x}_k)$ la **covariance empirique** entre les variables j et k,
- $r_{j,k} = cor(X_{.,j}, X_{.,k}) = \frac{v_{j,k}}{s_j s_k}$ la **corrélation empirique** entre les variables j et k.

La **matrice de covariance empirique** V est la matrice carrée de dimension p qui contient les covariances empiriques $V = (v_{j,k})$ et la **matrice de corrélation empirique** R est la matrice carrée de dimension p qui contient les corrélations empiriques $R = (r_{j,k})$.

Pour les données de température, on obtient la matrice de corrélation suivante, qui montre que les mois proches sont fortement corrélés entre eux.

```
cor(X)
```

```
##            Jan       Feb       Mar       Apr       May       Jun       Jul
## Jan 1.0000000 0.9873710 0.9174653 0.7768682 0.5902129 0.5555955 0.5360142
## Feb 0.9873710 1.0000000 0.9640393 0.8579105 0.6970370 0.6650910 0.6492428
## Mar 0.9174653 0.9640393 1.0000000 0.9439792 0.8234412 0.7982269 0.7814320
## Apr 0.7768682 0.8579105 0.9439792 1.0000000 0.9610470 0.9436962 0.9299161
## May 0.5902129 0.6970370 0.8234412 0.9610470 1.0000000 0.9903702 0.9857612
## Jun 0.5555955 0.6650910 0.7982269 0.9436962 0.9903702 1.0000000 0.9935398
## Jul 0.5360142 0.6492428 0.7814320 0.9299161 0.9857612 0.9935398 1.0000000
## Aug 0.6191451 0.7242421 0.8430424 0.9607885 0.9870768 0.9878088 0.9912968
## Sep 0.7779506 0.8601558 0.9379040 0.9858210 0.9531991 0.9400553 0.9370408
## Oct 0.9227019 0.9667239 0.9780700 0.9398621 0.8355324 0.8089992 0.8015980
## Nov 0.9734662 0.9899279 0.9583493 0.8714179 0.7306522 0.7002079 0.6885199
## Dec 0.9961367 0.9850745 0.9139825 0.7828365 0.6051128 0.5721629 0.5558529
##            Aug       Sep       Oct       Nov       Dec
## Jan 0.6191451 0.7779506 0.9227019 0.9734662 0.9961367
## Feb 0.7242421 0.8601558 0.9667239 0.9899279 0.9850745
## Mar 0.8430424 0.9379040 0.9780700 0.9583493 0.9139825
```

```
## Apr 0.9607885 0.9858210 0.9398621 0.8714179 0.7828365
## May 0.9870768 0.9531991 0.8355324 0.7306522 0.6051128
## Jun 0.9878088 0.9400553 0.8089992 0.7002079 0.5721629
## Jul 0.9912968 0.9370408 0.8015980 0.6885199 0.5558529
## Aug 1.0000000 0.9709066 0.8615815 0.7588511 0.6373514
## Sep 0.9709066 1.0000000 0.9550548 0.8859703 0.7908545
## Oct 0.8615815 0.9550548 1.0000000 0.9811039 0.9317754
## Nov 0.7588511 0.8859703 0.9811039 1.0000000 0.9801624
## Dec 0.6373514 0.7908545 0.9317754 0.9801624 1.0000000
```

Utilisation de graphiques.

Il est vite fastidieux d'analyser ces résumés numériques. On privilégie plutôt les **graphiques** (on parle de 'visualisation de données'). Par exemple, pour résumer une distribution on peut utiliser des boxplots (ou 'boîtes à moustache'). Sur la figure ci-dessous, on visualise la distribution des températures chaque mois.

```
boxplot(X,xlab='Mois',ylab='Température (deg Celsius)')
lines(1:12,X[6,],col='red')  #représentation de Brest en rouge
```

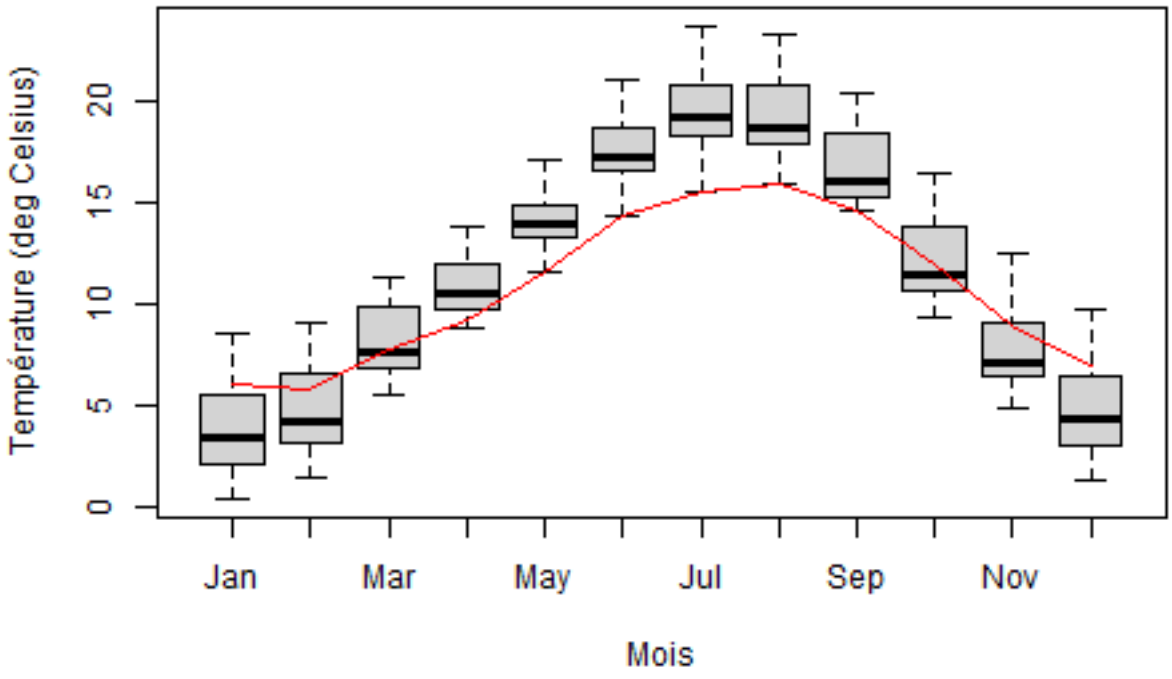


Figure 1: Boite à moustache (boxplot) des températures en fonction du mois et températures mensuelles à Brest (rouge)

Remarques 1 *Pour les rapports, il faut que toutes les figures aient une légende et que les axes aient des noms (avec éventuellement les unités).*

Pour le jeu de données considéré, le nombre d'individus ($n = 30$) est relativement faible. Il est donc possible de représenter individuellement chaque individu comme sur la figure Lorsque le nombre d'individus est grand (ce qui est souvent le cas en actuariat), ceci n'est plus possible.

```
xi=apply(X,1,mean)  #température moyennes dans les villes
boxplot(t(X[order(xi),]),ylab='Température (deg Celsius)',las=2)
```

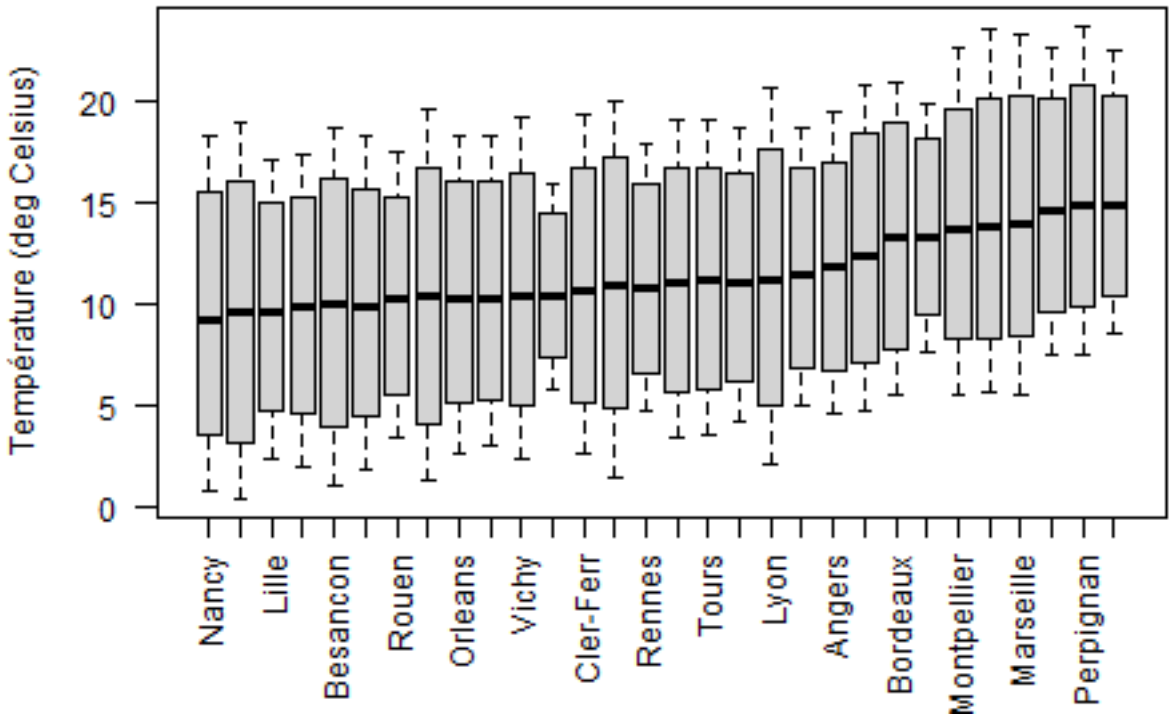


Figure 2: Boite à moustache (boxplot) des températures en fonction de la ville

```
#les villes sont triées par température moyenne croissante
#las=2 permet d'écrire le nom des villes verticalement
```

Pour les données géographiques, on peut aussi faire des cartes.

```
library(leaflet)
coord <- read.csv("~/public_html/doc_cours/L3EURIA/AD/latlon.txt",
                  sep= ' ',row.names = 1 )
 coord$Lon=-coord$Lon
 xi=apply(X,1,mean)  #température moyennes dans les villes
 domain <- range(xi)
 library(leaflet)
 #pal=colorNumeric("Blues", domain = domain)
 pal <- colorNumeric("Reds", domain = domain)
 leaflet() %>%
   addTiles()   %>%
   setView(lng = mean(coord$Lon), lat = mean(coord$Lat), zoom = 5)%>%
   addCircleMarkers(lng=coord$Lon, lat=coord$Lat,color = pal(xi),
                    label = as.character(sort(round(xi, 1))),
                    labelOptions = labelOptions(noHide = T, textsize = "12px",
                                                textOnly = T, direction = "center"))%>%
  addLegend(pal = pal, values = domain, position = "bottomleft")
```

Sur la carte précédente, on peut voir les températures moyennes pour les différentes villes de France. Pour faire ressortir les différentes régions climatiques de France, on peut aussi s'intéresser aux différences de température entre l'été et l'hiver (cf carte ci-dessous). Il y a moins de différence dans les villes situées proches de l'océan atlantique (dont Brest) que dans les villes situées à l'est (climat plus continental).

Figure 3: Carte des température moyennes en France

```
library(leaflet)
coord <- read.csv("~/public_html/doc_cours/L3EURIA/AD/latlon.txt",
                  sep= ' ',row.names = 1 )
 coord$Lon=-coord$Lon
 xi=X[,7]-X[,1]  #différence de température entre juillet et janvier
 domain <- range(xi)
 library(leaflet)
 #pal=colorNumeric("Blues", domain = domain)
 pal <- colorNumeric("Reds", domain = domain)
leaflet() %>%
   addTiles()   %>%
   setView(lng = mean(coord$Lon), lat = mean(coord$Lat), zoom = 5)%>%
   addCircleMarkers(lng=coord$Lon, lat=coord$Lat,color = pal(xi),label = as.character(sort(round(xi, 1))),
                    labelOptions = labelOptions(noHide = T, textsize = "12px",
                                                textOnly = T, direction = "center"))%>%
  addLegend(pal = pal, values = domain, position = "bottomleft")
```

Figure 4: Différence de température entre le mois de juillet et le mois de janvier moyennes en France

Afin de visualiser la dépendance entre deux variables, on peut faire des nuages de points mais l'analyse est vite fastidieuse si le nombre de variables est important.

```
plot(X)
```

Figure 5: Nuages de points des températures aux différents mois

On peut alors choisir de représenter seulement les matrices de corrélation.

```
library(corrplot)
```

```
## corrplot 0.84 loaded
```

```
corrplot(cor(X))
```

On est cependant limité à analyser les relations entre les couples de variables (éventuellement on peut représenter la relation entre 3 variables en faisant des graphiques 3D).

Quelques notations d'algèbre linéaire.

- Si X est une matrice, on notera X' sa transposée.
- $\mathbf{1}_n$ est le vecteur colonne de $\mathbb{R}^n$ dont toutes les coordonnées sont égales à 1 :

$$\mathbf{1}_n = \begin{pmatrix} 1 \\ \vdots \\ 1 \end{pmatrix}.$$

- On note 0 une matrice dont tous les coefficients sont nuls (abusivement, sans préciser la dimension).
- Si x, y sont deux vecteurs de $\mathbb{R}^n$, on note $(x, y) = x'y = \sum_{i=1}^n x_i y_i$ le produit scalaire usuel.

Exercice 2 *On reprend les notations du cours.*

1. *Montrer que* $v_j = \frac{1}{n}\left(\sum_{i=1}^n x_{i,j}^2\right) - (\bar{x}_j)^2$.
2. *Montrer que* $\bar{x}_j = argmin_{\mu \in \mathbb{R}}(F(\mu))$ *avec* $F(\mu) = \sum_{i=1}^n (x_{i,j} - \mu)^2$. *Interprétation?*

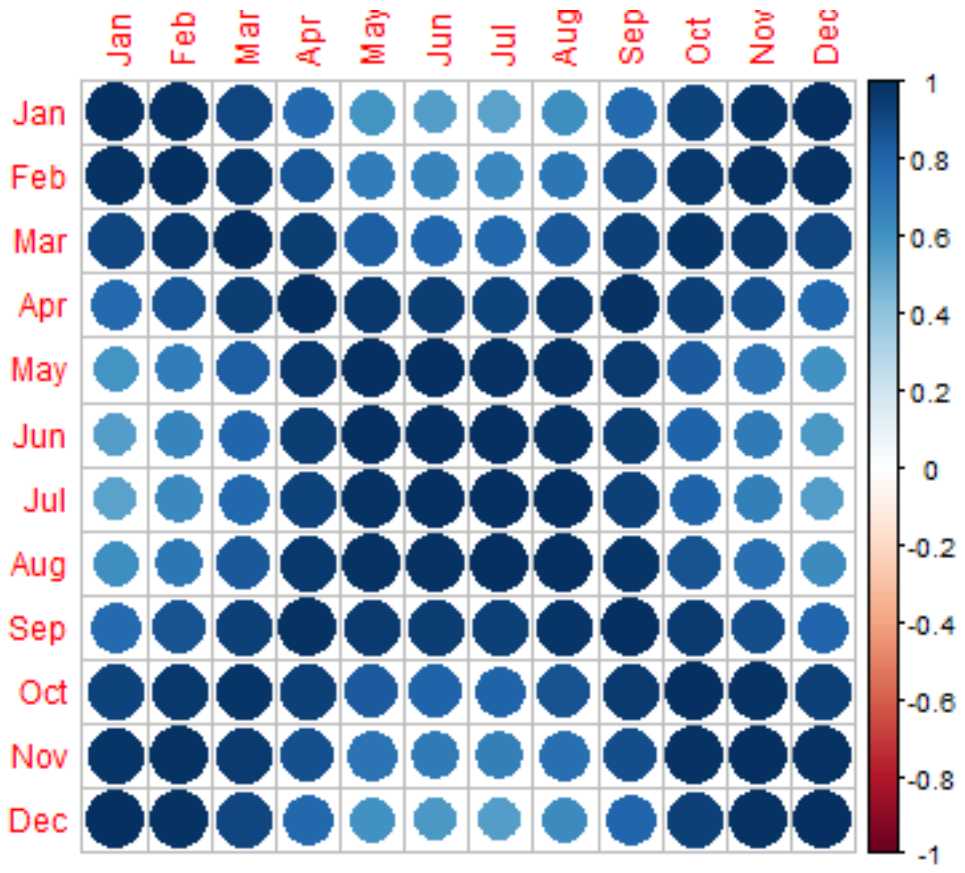


Figure 6: Matrice de corrélation entre les températures aux différents mois

3. *Calculer* $\mathbf{1}_n' X$.
4. *On considère le tableau de données* $Y = (y_{i,j})$ *avec* $y_{i,j} = x_{i,j} - \bar{x}_j$. *Calculer* $\bar{y}_j$ *et* $var(Y_{\cdot,j})$. *On appelle* Y *le tableau de données centré. Pourquoi?*
5. *Calculer* $\mathbf{1}_n' Y$.
6. *On considère le tableau de données* $Z = (z_{i,j})$ *avec* $z_{i,j} = (x_{i,j} - \bar{x}_j)/s_j$. *Calculer* $\bar{z}_j$ *et* $var(Z_{\cdot,j})$. *On appelle* Z *le tableau de données centré-réduit. Pourquoi?*
7. *Vérifier que* $V = \frac{1}{n} Y'Y$ *et* $R = \frac{1}{n} Z'Z$. *Quelle est la diagonale de la matrice* R*? Quelle est la diagonale de la matrice* V*?*
8. *En déduire que les matrices* V *et* R *sont des matrices symétriques positives.*
9. *Vérifier que* $r_{j,k} = \frac{(Y_{\cdot,j}, Y_{\cdot,k})}{\|Y_{\cdot,j}\| \|Y_{\cdot,k}\|} = (Z_{\cdot,j}, Z_{\cdot,k})$.
10. *En déduire que* $r_{j,k} \in [-1, 1]$ *avec les cas extrêmes*
    - *si* $r_{j,k} = 1$ *alors il existe* $\mu \in \mathbb{R}$ *et* $\lambda > 0$ *tel que* $x_{i,j} = \mu + \lambda x_{i,k}$
    - *si* $r_{j,k} = -1$ *alors il existe* $\mu \in \mathbb{R}$ *et* $\lambda < 0$ *tel que* $x_{i,j} = \mu + \lambda x_{i,k}$
11. *Proposer un exemple de couple de variables* $(X_{\cdot,1}, X_{\cdot,2})$ *tel que* $x_{i,2} = f(x_{i,1})$ *avec* $f : \mathbb{R} \to \mathbb{R}$ *mais* $cor(X_{\cdot,1}, X_{\cdot,2}) = 0$

Correction de l'exerice 2 *L'exercice a été corrigé au tableau. Les codes R ci-dessous correspondent à la dernière question.*

```
x=seq(-5,5,by=.1) #vecteur symétrique par rapport à 0
y=x^2  #dépendance quadratique
cor(cbind(x,y))   #corrélation nulle
```

```
##              x           y
## x 1.000000e+00 1.763581e-16
## y 1.763581e-16 1.000000e+00
```

A retenir de cet exercice :

- on peut calculer le vecteur des moyennes et les matrices de covariance et de corrélation en faisant des opérations matricielles sur le tableau de données,
- il ne faut pas confondre absence de corrélation et absence de relation : le coefficient de corrélation mesure seulement la dépendance linéaire entre deux variables.

2 ACP et réduction des matrices de covariance

Dans la suite du cours, on suppose que le tableau de données X est centré, c'est à dire que la moyenne de toutes les variables est nulle. Si le tableau de données de départ n'est pas centré, il suffit de centrer chacune des variables au préalable en enlevant la moyenne, c'est à dire en faisant la transformation

$$x_{i,j} \leftarrow x_{i,j} - \bar{x}_j.$$

Cela permet de simplifier de nombreuses formules. Par exemple, la covariance entre les variables j et k s'écrit

$$c_{j,k} = \frac{1}{n}\sum_{i=1}^{n}(x_{i,j} - \bar{x}_j)(x_{i,k} - \bar{x}_k) = \frac{1}{n}\sum_{i=1}^{n} x_{i,j}x_{i,k}$$

si les variables sont centrées. On en déduit que la matrice de covariance empirique V est donnée par $V = \frac{1}{n}X'X$ (cf exercice 2).

Réduction de la matrice de covariance.

Proposition 1 *La matrice de covariance V est semi-définie positive. Elle est définie positive si X est de rang p.*

Preuve 1

Pour tout $u \in \mathbb{R}^p$ non nul,

$$u'Vu = u'X'Xu = (Xu)'(Xu) = \| Xu \|^2 \geq 0$$

donc V est semi-définie positive L'inégalité précédente est stricte ssi *$Xu \neq 0$. Enfin dire que $Xu \neq 0$ pour tout $u \neq 0$ est équivalent à ce que X soit de rang p.*

On rappelle qu'une **matrice symétrique est diagonalisable dans une base orthonormée (b.o.n)**. Notons $(u_1, ...u_p)$ une b.o.n. de vecteurs propres associés aux valeurs propres $\lambda_1 \geq \lambda_2 \geq ... \geq \lambda_p \geq 0$ triées par ordre décroissant pour la matrice de covariance V (les valeurs propres sont positives car la matrice est positive). Notons $U = (u_1, ..., u_p)$ la matrice des vecteurs propres. On a

- $(u_i, u_j) = \delta_{i,j}$ (car c'est une b.o.n), c'est à dire $U'U = I_p$,
- $Vu_i = \lambda_i u_i$, c'est à dire $VU = UD$ avec $D = diag(\lambda_1, ..., \lambda_p)$,
- la projection orthogonale de x sur $E_k = vect(u_1, ..., u_k)$ est $\pi_{E_k}(x) = \sum_{i=1}^{k}(x, u_i)u_i$.

Application en analyse de données.

- Remarquons tout d'abord que multiplier le tableau de données X par une matrice à droite revient à définir de nouvelles variables en faisant des transformations linéaires des variables d'origine ("changement de variable").

  Exemple 1 . *On reprend les données de température pour les villes en France.*

  1. *On pose $u_1 = (1/12, 1/12, 1/12, ..., 1/12)' \in \mathbb{R}^{12}$. Xu_1 est une nouvelle variable qui décrit les températures annuelles moyennes dans chaque ville.*
  2. *On pose $u_2 = (1/2, -1/2, ..., 0)' \in \mathbb{R}^{12}$. Xu_2 est une nouvelle variable qui décrit la différence de température entre les mois de janvier et février dans chaque ville.*

- Posons alors $C = XU$. D'après la remarque précédente, C s'interprète comme des nouvelles variables obtenues par transformation linéaire des variables d'origine. On appelle $C_{.,j}$ la jème **composante principale**. On vérifie que $C_{i,j} = (X_{i,}, u_j)$ et donc $\pi_{vect(u_j)}(X_{i,.}) = C_{i,j}u_j$. Le vecteur $C_{.,j}$ donne donc les coefficients de la projection orthogonale des individus dans la direction u_j.
- Comme le tableau X est centré, le tableau de données C est également centré (il suffit de vérifier $\mathbf{1}_n' X = 0$). Sa matrice de covariance est donc donnée par $V_C = C'C = U'X'XU = D$.
- En particulier on a
  - $var(C_{.,j}) = \lambda_j$ (les nouvelles variables sont triées par ordre décroissant de variance : $C_{.,1}$ est la variable avec la plus forte variance, $C_{.,p}$ celle avec la plus faible variance).
  - $cov(C_{.,j}, C_{.,k}) = 0$ si $j \neq k$ (les covariances/corrélations entre les nouvelles variables sont nulles).

Exercice 3 *On considère les données de température en France.*

1. *Calculer les vecteurs propres et les valeurs propres de la matrice de covariance.*
2. *Quelle est la variance de la première composante principale? De la dernière composante principale? Qu'est-ce que ça implique en pratique?*
3. *Proposer une interprétation des variables $C_{.,1}$ et $C_{.,2}$ à partir des coordonnées des vecteurs u_1 et u_2.*
4. *Tracer le nuage de point $C_{.,1}, C_{.,2}$ en rajoutant le nom des villes sur les points avec la commande R 'text'. Proposer une interprétation.*
5. *Vérifier que $cov(C_{.,1}, C_{.,2}) = 0$*

Correction de l'exerice 3

```
#Question 1
X=scale(X,center=TRUE,scale=FALSE)  #centrage des données
V=1/(nrow(X))*t(X)%*%X  #calcul de la matrice de covariance (estimateur biaisé)
res=eigen(V) #calcul des valeurs propres
res$values   #les valeurs propres sont triées par ordre décroissant
```

```
##  [1] 3.712683e+01 5.678507e+00 1.524163e-01 8.915504e-02 4.110496e-02
##  [6] 1.955078e-02 1.571462e-02 1.099793e-02 6.052900e-03 3.887854e-03
## [11] 3.170728e-03 9.244589e-04
```

```
#on peut aussi vérifier que les vecteurs propres sont orthonormés
#question 2
C=as.matrix(X)%*%res$vectors  #composantes principales
VC=1/(nrow(C))*t(C)%*%C  #matrice de covariance de C
VC[1,1] #variance de la première composante principale
```

```
## [1] 37.12683
```

```
res$values[1] #même résultat en regardant la valeur propre
```

```
## [1] 37.12683
```

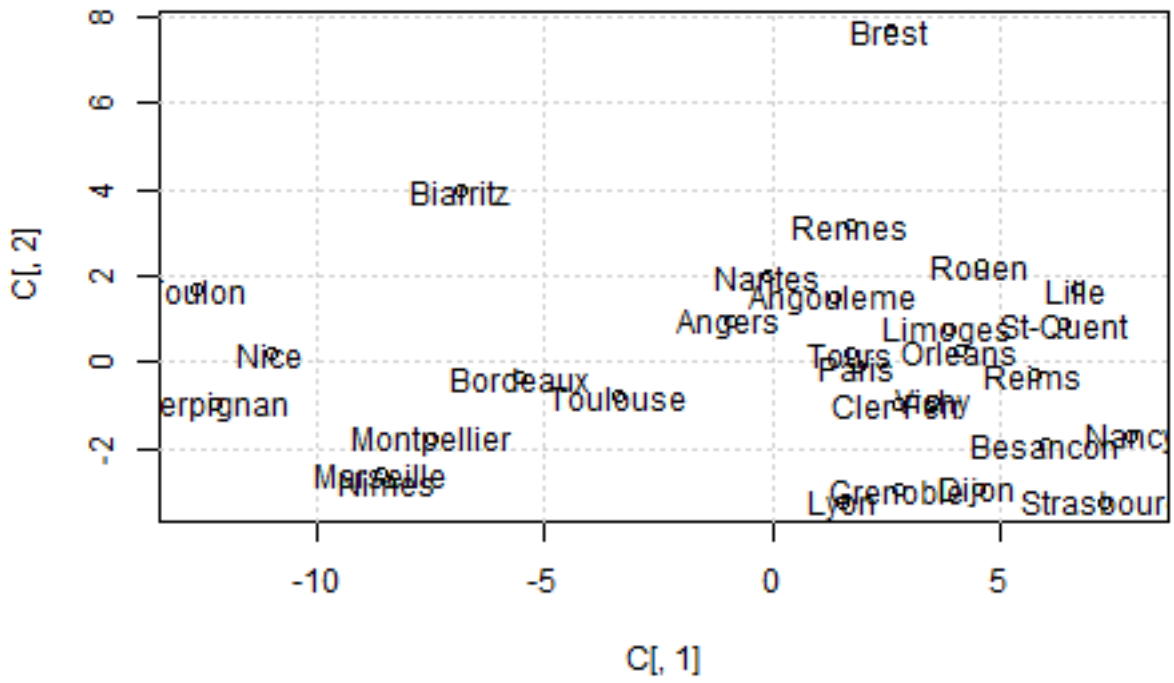
```
res$values[ncol(X)] #variance de la dernière composante principale
```

```
## [1] 0.0009244589
```

```
#la variance de la dernière composante principale est beaucoup plus petite
#que la variance de la première composante principale
#implication pratique : la dernière composante principale apporte moins d'information
#que la première composante principale
#Question 3
# Les nouvelles variables sont des sommes pondérées des températures des différents mois
#les poids sont donnés par les vecteur propres
res$vectors[,1:2]
```

```
##              [,1]       [,2]
##  [1,] -0.3212437  0.4052201
##  [2,] -0.3212764  0.2672625
##  [3,] -0.2732748  0.0650229
##  [4,] -0.2312379 -0.1290046
##  [5,] -0.2048896 -0.2710376
##  [6,] -0.2421028 -0.3572129
##  [7,] -0.2839775 -0.4404708
##  [8,] -0.2895629 -0.3496510
##  [9,] -0.3033023 -0.1659780
## [10,] -0.3234656  0.0707444
## [11,] -0.3178782  0.2183796
## [12,] -0.3204398  0.3819966
```

```r
#Interprétation c_1
# A peu près les mêmes coefficients pour tous les mois
# Proportionnelle à la moyenne annuelle (au signe et à la normalisation près)
#Interprétation c_2
# Poids opposés pour les mois d'hiver et les mois d'été
# Mesure la différence de température hiver /été
#Question 4
plot(C[,1],C[,2])
grid()
text(C[,1],C[,2],row.names(X))
```




```r
#D'après l'interprétation ci-dessus
# - la direction horizontale est liée à la température moyenne sur l'année
# on retrouve donc les villes avec un climat chaud (Toulon, Nice,...) d'un côté
# et celles avec un climat froid (Nancy, Strasbourg) de l'autre côté
# - la direction verticale est liée au gradient de température entre l'hiver et l'été
# on retrouve donc les villes avec un climat océanique (Brest en particulier) d'un côté
```

```r
# et celles avec un climat continental (Lyon, Grenoble,...) de l'autre côté

#question 5
cov(C[,1],C[,2])  #proche de 0
```

```
## [1] 3.561873e-15
```

A retenir

- Réduire la matrice de covariance dans une base orthonormée permet de définir des nouvelles variables qui sont décorrélées.
- On peut ordonner ces variables en fonction de leur variance. Les variables avec une faible variance sont telles que tous les individus prennent des valeurs "similaires". Elle apportent moins d'information sur les individus que les variables avec une forte variance (on détaillera cette idée dans le paragraphe et l'exercice 5).
- u_1 donne la direction sur laquelle il faut projeter les individus pour obtenir une variable de variance maximale λ_1. u_p donne la direction sur laquelle il faut projeter les individus pour obtenir une variable de variance minimale λ_p (cf Exercice 4).
- Si on veut représenter les individus dans un espace de dimension 2, ceci suggère de représenter les individus en les projetant sur le plan engendré par u_1 et u_2. Les coordonnées de ces projections sont données par les deux premières composantes principales $C_{,1}$ et $C_{,2}$.

Dans les paragraphes suivants, nous allons en particulier discuter les points suivants :

- Est-ce que le changement de variables associé aux vecteurs propres de la matrice de covariance est optimal? Si oui, dans quel sens?
- Comment peut-on interpréter plus précisément les résultats obtenus?

Exercice 4 *On reprend les notations du cours. Soit $v \in \mathbb{R}^p$ tel que $\| v \| = 1$ et $y = Xv$ la nouvelle variable associée.*

1. *On pose $w = U'v$. Montrer que $\| w \| = 1$*
2. *Montrer que $var(y) = \sum_{i,j=1}^{p} v_i V_{i,j} v_j = \sum_{i=1}^{p} \lambda_i w_i^2$*
3. *En déduire que $\lambda_p \leq var(y) \leq \lambda_1$*

Correction de l'exerice 4 *Exercice corrigé au tableau.*

3 ACP et inertie

3.1 Inertie et projections orthogonales

La dispersion d'une variable autour de sa moyenne se mesure par sa variance. L'inertie généralise cette notion en multivarié (c'est à dire lorsqu'on a plusieurs variables).

Définition 1 *On note $\bar{x} = (\bar{x}_1, ..., \bar{x}_p)$. L'**inertie totale** du nuage de points associée au tableau X est la moyenne des distances au carré des points à leur centre de gravité :*

$$I = \frac{1}{n} \sum_{i=1}^{n} ||X_{i,\cdot} - \bar{x}||^2$$

I caractérise la dispersion du nuage par rapport à son centre : plus I est grand, plus le nuage est dispersé autour de son centre de gravité $\bar{x}$. Une inertie nulle signifie que tous les individus sont identiques.

Proposition 2 *On a*

$$I = \sum_{j=1}^{p} Var(X_{\cdot,j}) = Tr(V).$$

avec $Tr(V)$ la trace de la matrice V.

Dans la suite on suppose à nouveau que la tableau est centré. On a alors $\bar{x} = 0$ et

$$I = \frac{1}{n}\sum_{i=1}^{n} ||X_{i,.}||^2 = \frac{1}{n}\sum_{i=1}^{n}\sum_{j=1}^{p} x_{i,j}^2 = \frac{1}{n}\sum_{j=1}^{p} ||X_{.,j}||^2 = \frac{1}{n}Tr(X'X)$$

Rappel : Soit F un sous-espace vectoriel de $\mathbb{R}^p$. Notons $F^\perp$ l'orthogonal de F :

$$F^\perp = \{x \in \mathbb{R}^p, \forall y \in F, (x, y) = 0\}.$$

La projection orthogonale $\pi_F(x)$ de x sur F est l'unique élément vérifiant $\pi_F(x) \in F$ et $x - \pi_F(x) \in F^\perp$

Rappelons que tout vecteur x de $\mathbb{R}^p$ peut s'écrire

$$x = \pi_F(x) + \pi_{F^\perp} x,$$

Cette relation dit aussi que

$$\pi_{F^\perp} x = x - \pi_F x.$$

Rappelons également que si $(f_1, ..., f_p)$ est une b.o.n. de F alors $\pi_F(x) = \sum_{i=1}^{p}(x, f_i)f_i$.

Proposition 3 *Notons I_F l'inertie totale du nuage de point projeté sur l'espace F, c'est à dire*

$$I_F = \frac{1}{n}\sum_{i=1}^{n} \parallel \pi_F(X_{i,.}) \parallel^2 .$$

Alors

$$I = I_F + I_{F^\perp}$$

.

Preuve 2 *Remarquons tout d'abord que $\frac{1}{n}\sum_{i=1}^{n} \pi_F(X_{i,.}) = \pi_F(\frac{1}{n}\sum_{i=1}^{n} X_{i,.}) = 0$ et donc l'inertie du nuage de points $\pi_F(X_{i,.})$ est bien donnée par $I_F = \frac{1}{n}\sum_{i=1}^{n} \parallel \pi_F(X_{i,.}) \parallel^2$.*

On a

$$\begin{aligned} I &= \frac{1}{n}\sum_{i=1}^{n} \parallel X_{i,.} \parallel^2 = \frac{1}{n}\sum_{i=1}^{n} \parallel X_{i,.} - \pi_F(X_{i,.}) + \pi_F(X_{i,.}) \parallel^2 \\ &= \frac{1}{n}\sum_{i=1}^{n} \parallel X_{i,.} - \pi_F(X_{i,.}) \parallel^2 + \frac{1}{n}\sum_{i=1}^{n} \parallel \pi_F(X_{i,.}) \parallel^2, \\ &= \frac{1}{n}\sum_{i=1}^{n} \parallel \pi_{F^\perp}(X_{i,.}) \parallel^2 + \frac{1}{n}\sum_{i=1}^{n} \parallel \pi_F(X_{i,.}) \parallel^2, \end{aligned}$$

par le théorème de Pythagore.

Une conséquence de la proposition précédente est que $I_F \leq I$: la dispersion globale d'un ensemble de points ne peut que diminuer par projection sur un sous-espace.

L'objectif de l'ACP est obtenir une représentation approchée du nuage des n individus, situés dans un espace de dimension p, dans un sous-espace de faible dimension k (avec $k = 2$ ou 3 en général, pour permettre une représentation graphique).

Lorsqu'on fait une ACP, l'espace de dimension réduit sur lequel on projette les individus est choisi de telle manière à réduire le moins possible l'inertie. Cette idée est formalisée dans la définition ci-dessous.

Définition 2 *On appelle **sous-espace principal** de dimension k, tout sous-espace vectoriel E_k de dimension k vérifiant*

$$I_{E_k} = \sup\{I_E;\ E \text{ s.e.v. de } \mathbb{R}^p \text{ de dimension } k\}$$

Remarque 1 *D'après la proposition précédente, on a*

$$I = I_F + \frac{1}{n}\sum_{i=1}^{n} \|X_{i,.} - \pi_F(X_{i,.})\|^2,$$

où le deuxième terme décrit la distance entre les points originaux et leurs projetés orthogonaux, c'est à dire la 'déformation' du nuage de point. Maximiser le premier terme revient à rechercher le sous-espace E_k tel que la moyenne des carrés des distances des points $X_{i,.}$ aux points projetés $pi_F(X_{i,.})$ soit minimale, ce qui s'interprète aussi en terme de moindre déformation du nuage.

Exercice 5 *Cet exercice est à réaliser avec R.*

1. *Simuler un jeu de données X avec $n = 100$ individus et $p = 2$ variables de telle manière que*
   - *la première variable est simulée selon une loi $\mathcal{N}(0, 1)$*
   - *la deuxième variable est simulée selon le modèle $x_{i,2} = x_{i,1} + \epsilon_i$ avec ϵ_i une réalisation d'un échantillon iid de la loi $\mathcal{N}(0, 0.1)$.*

   Les variables seront centrées avant de continuer l'exercice.
2. *Tracer le nuage de points. Calculer une base orthonormée (u_1, u_2) constituée de vecteurs propres de la matrice de covariance et représenter les espaces vectoriels $vect(u_1)$ et $vect(u_2)$ sur la même figure.*
3. *Quelle est l'inertie totale I du nuage de points? Que vaut $I_{vect(u_1)}$? Que vaut $I_{vect(u_1)}$? Vérifier que $I = I_{vect(u_1)} + I_{vect(u_2)}$.*
4. *Ecrire une fonction R **Itheta=function(theta,Z)** qui calcule $I_{vect(u_\theta)}$ avec $u_\theta = (cos(\theta), sin(\theta))'$.*
5. *Tracer la fonction $\theta \mapsto I_{vect(u_\theta)}$ et vérifier que cette fonction atteint son minimum et son maximum lorsque u_θ est un vecteur propre u_i.*

Correction de l'exerice 5

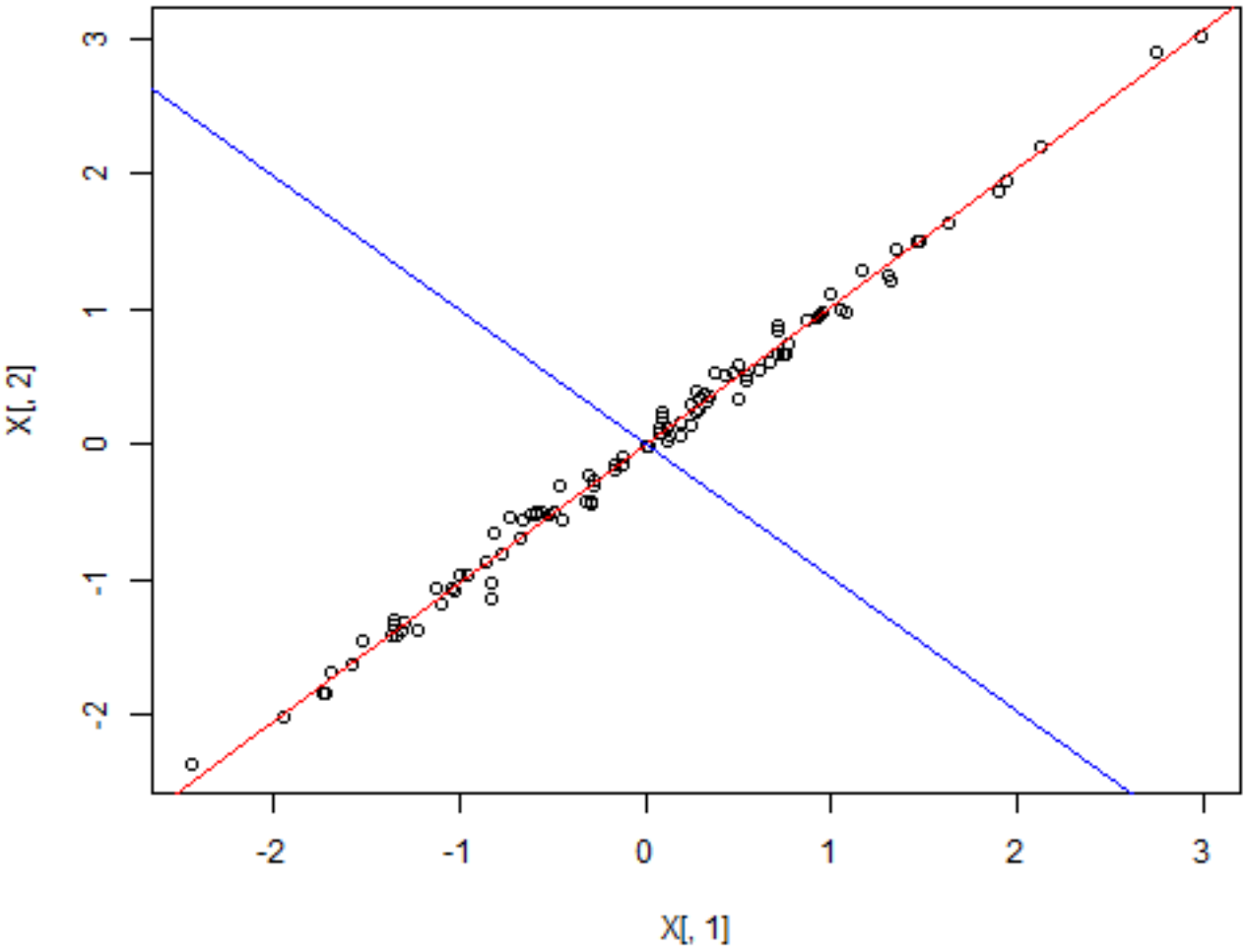
```
#Question 1 Simulation des données
n=100
sig=.1
x1=rnorm(n)  #simulation première variable
x2=x1+sig*rnorm(n)  #simulation deuxième variable
X=cbind(x1,x2)
X=scale(X,scale=FALSE)  #centrage des données


#Question 2
plot(X[,1],X[,2])  #nuage de points
C=t(X)%*%X/n  #matrice de covariance
vp=eigen(C)$vectors  #calcul éléments propres
vp[1,1]^2+vp[2,1]^2  #les vecteurs renvoyés par eigen sont normés
```

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

```
sum(vp[,1]*vp[,2])  #les vecteurs renvoyés par eigen sont orthogonaux
```

```
## [1] 0
```

```
abline(0,vp[2,1]/vp[1,1],col='red')  #représentation vect(u1)
abline(0,vp[2,2]/vp[1,2],col='blue') #représentation vect(u2)
```




```
#Question 3
I=sum(X^2)/nrow(X)  #inertie totale
coordproj1=X%*%vp[,1]   #coordonnées projection sur vect(u1)
I1=sum(coordproj1^2)/nrow(X)    #I_vect(u1)
coordproj2=X%*%vp[,2]   #coordonnées projection sur vect(u2)
I2=sum(coordproj2^2)/nrow(X)    #I_vect(u2)
I
```

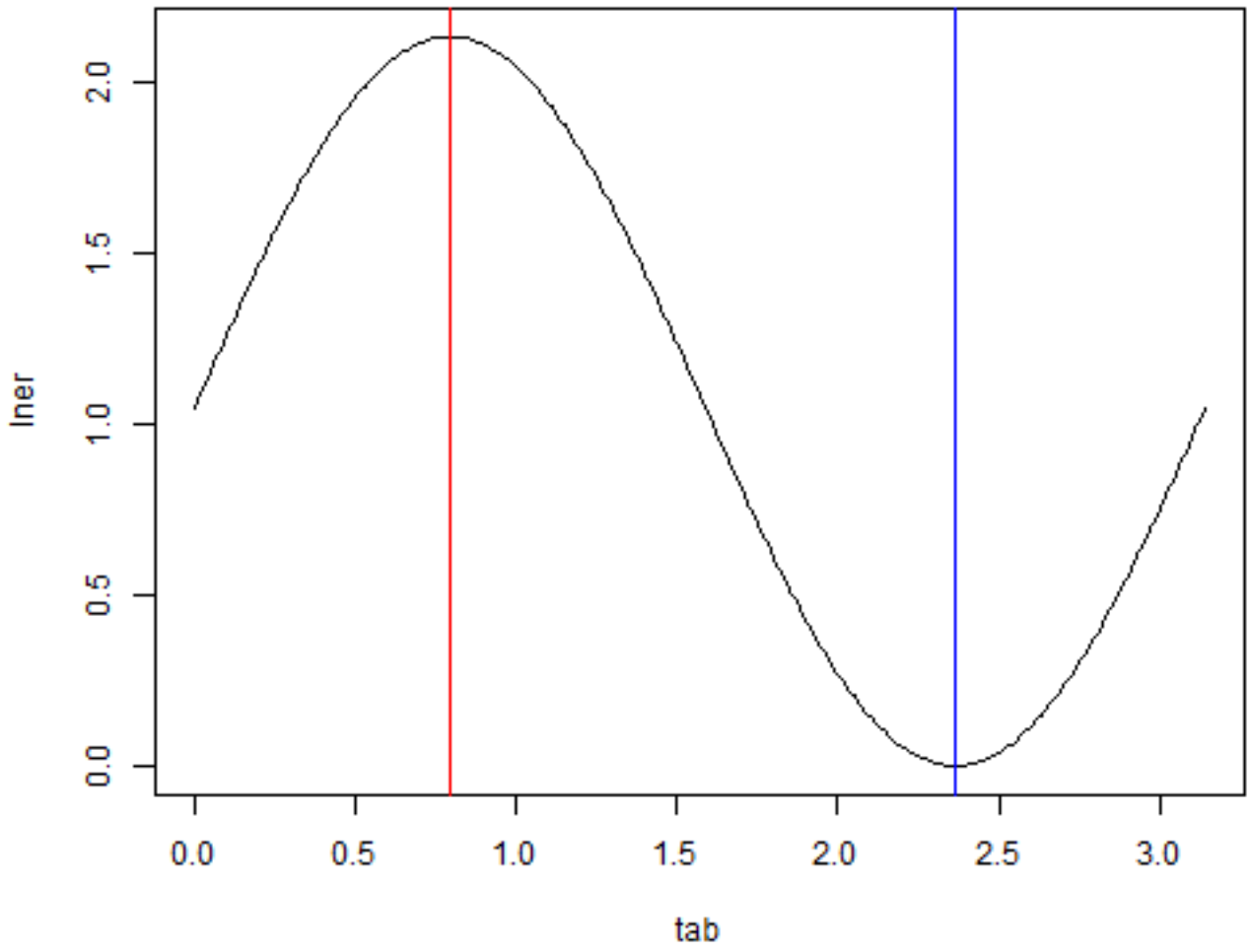
```
## [1] 2.136147
```

```
I1+I2  #on retrouve les mêmes valeurs
```

```
## [1] 2.136147
```

```
#Quesion 4
Itheta=function(theta,X){
  u=c(cos(theta),sin(theta))
  coordproj=X%*%u
  return(sum(coordproj^2)/nrow(X))
}

#Quesion 4
tab=seq(0,pi,by=.01)
```

```
Iner=NULL
for (i in 1:length(tab)){
  Iner[i]=Itheta(tab[i],X)
}
plot(tab,Iner,type='l')
abline(v=atan2(vp[2,1],vp[1,1])%%pi,col='red')  #direction associée à u1 (inertie max)
abline(v=atan2(vp[2,2],vp[1,2])%%pi,col='blue')  #direction associée à u2 (inertie min)
```




```
#Conclusion : il faut projeter le nuage de points sur vect(u1) pour obtenir inertie maximale
#vect(u1) est l'espace de dimension 1 sur lequel il faut projeter
#pour déformer le moins possible le nuage de points
```

3.2 Espaces principaux

Calcul du sous-espace principal de dimension 1.

On cherche E_1 de dimension 1 tel que

$$I_{E_1} = \sup\{I_E;\ E \text{ s.e.v. de } \mathbb{R}^p \text{ de dimension } 1\}$$

Soit E un espace de 1. Soit u un vecteur unitaire qui appartient à E de telle manière que $E = vect(u)$.

On a $\pi_E(X_{i,.}) = (X_{i,.}.u).u$ et donc

$$I_E = \frac{1}{n}\sum_{i=1}^{n} \parallel \pi_E(X_{i,.}) \parallel^2 . = \frac{1}{n}\sum_{i=1}^{n}(X_{i,.}.u)^2$$

car $\parallel u \parallel = 1$. On en déduit que

$$I_E = \frac{1}{n} \parallel Xu \parallel^2 = \frac{1}{n}u'X'Xu = \frac{1}{n}u'Vu = var(y)$$

avec $y = Xu$ On a vu dans l'exercice 5 que $\lambda_p \leq I_E \leq \lambda_1$ avec $\lambda_1 \geq ... \geq \lambda_p$ les valeurs propres ordonées de V. De plus si $u = u_1$ est un vecteur propre normé associé à λ_1, on obtient

$$I_{vect(u_1)} = \frac{1}{n}u_1'Vu_1 = \lambda_1$$

car $Vu_1 = \lambda_1 u_1$.

On a donc montré que $E_1 = vect(u_1)$ est un sous espace principal de dimension 1 avec u_1 un vecteur propre normé associé à la plus grande valeur propre λ_1 de la matrice de covariance du tableau de données.

Construction itérative des espaces principaux.

Théorème 1 *Soit E_k un sous-espace vectoriel de dimension $k < p$ portant l'inertie maximale du nuage. Alors le sous-espace de dimension $k+1$ portant l'inertie maximale est*

$$E_k \oplus vect(u_{k+1})$$

où $vect(u_{k+1})$ est une droite orthogonale à E_k portant l'inertie maximale parmi toutes les droites orthogonales à E_k.

Preuve 3 *Soit F un sous-espace de dimension $k+1$. Comme*

$$dim(E_k^{\perp}) + dim(F) = (p-k) + (k+1) = p+1$$

$E_k^{\perp}$ et F ont au moins une intersection commune. Soit $u \in E_k^{\perp} \cap F$. On peut alors écrire $F = \overline{F} \oplus vect(u)$, où $\overline{F}$ est le supplémentaire de $vect(u)$ dans F. $\overline{F}$ est de dimension k, et par définition de E_k on a donc $I_{\overline{F}} \leq I_{E_k}$. Par ailleurs, par définition de u_{k+1}, on a aussi $I_u \leq I_{u_{k+1}}$. Ainsi

$$I_F = I_{\overline{F}} + I_u \leq I_{E_k} + I_{u_{k+1}} = I_G$$

où $G = E_k \oplus vect(u_{k+1})$ est de dimension $k+1$. On a bien $I_G = E_{k+1}$

Définition 3 *Les espaces vectoriels $vect(u_1), \cdots, vect(u_p)$ sont appelés **axes principaux** de l'ACP et les vecteurs $u_1, \cdots, u_p$ **vecteurs principaux** de l'ACP.*

Remarque 2 *Le théorème précédent dit que les sous-espaces principaux E_k peuvent se calculer de la façon itérative suivante :*

1. *Rechercher un axe u_1 maximisant l'inertie $I_{vect(u_1)}$. On note $E_1 = vect(u_1)$. On a vu que u_1 est un vecteur propre associé à la plus grande valeur de V.*
2. *Rechercher un axe $vect(u_2)$ orthogonal à E_1, maximisant l'inertie $I_{vect(u_2)}$. On peut montrer que u_2 est un vecteur propre associé à la deuxième plus grande valeur de V. On note $E_2 = E_1 \oplus vect(u_2)$.*
3. *$\cdots$,*
4. *Rechercher un axe $vect(u_k)$ orthogonal à E_{k-1}, maximisant l'inertie $I_{vect(u_k)}$. On peut montrer que u_k est un vecteur propre associé à la kième plus grande valeur de V. On note $E_k = E_{k-1} \oplus vect(u_k)$.*

3.3 Le théorème principal

Théorème 2 *La matrice V admet p valeurs propres $\lambda_1 \geq \lambda_2 \geq \ldots \lambda_p \geq 0$ de vecteurs propres unitaires associés $u_1, \ldots, u_p$ deux à deux orthogonaux.*

Pour $k = 1, \ldots, p$, le sous-espace vectoriel E_k de dimension k portant l'inertie maximale est engendré par les vecteurs $u_1, \ldots, u_k$. De plus, $I_{E_k} = \sum_{i=1}^{k} \lambda_i$. Les vecteurs u_j sont appelés vecteurs principaux de l'ACP.

Ceci montre que les vecteurs propres associés à la matrice de covariance donnent les espaces optimaux pour projeter le nuage de point initial en le "déformant le moins possible".

4 Analyse rapide des résultats d'une ACP

On rappelle que le but de l'ACP est de fournir une représentation graphique du nuage des individus sur un espace de dimension $q < p$ (généralement $q = 2$ ou 3). On sait maintenant que la "meilleure" représentation graphique (au sens de l'inertie) est donnée par la projection du nuage sur l'espace principal E_q engendré par les q premiers axes principaux $u_1, \cdots, u_q$, et que la coordonnée de l'individu i sur l'axe u_k est $C_{i,k}$ avec $C_{.,k}$ la kième composante principale.

4.1 Qualité globale d'une ACP

Le critère le plus simple et le plus utilisé pour mesurer la qualité globale de la représentation du nuage de point sur les sous-espace principal E_k est donné dans la définition suivante.

Définition 4 *La qualité du sous-espace principal E_k se mesure par le* ***pourcentage (ou part) d'inertie totale expliquée*** *:*

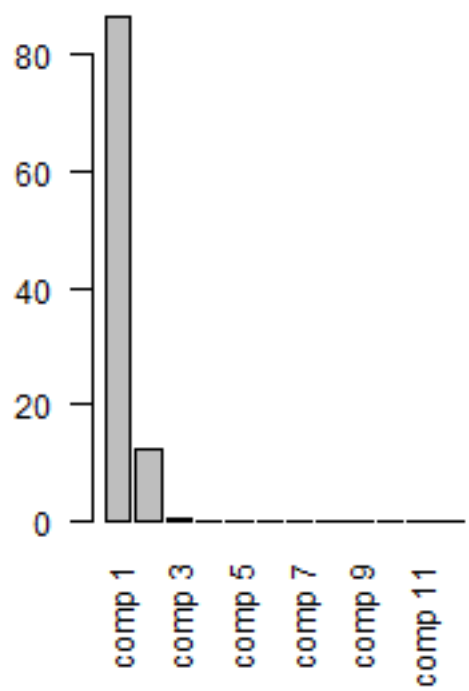
$$\frac{\lambda_1 + \lambda_2 + \cdots + \lambda_k}{I} = \frac{\lambda_1 + \lambda_2 + \cdots + \lambda_k}{\lambda_1 + \lambda_2 + \cdots + \lambda_k + \cdots + \lambda_p}$$

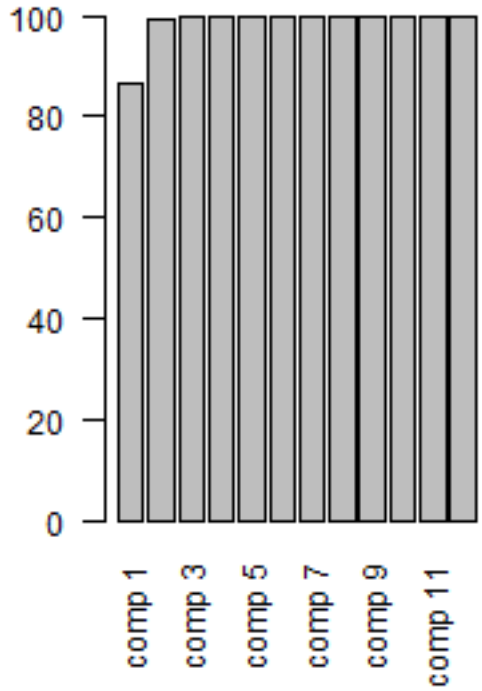
On appelle ***part d'inertie expliquée par la αième composante principale*** *la quantité suivante*

$$\frac{\lambda_\alpha}{\lambda_1 + \lambda_2 + \cdots + \lambda_p}$$

Il n'y a pas de règle absolue pour déterminer à partir de quel pourcentage la projection sur le sous-espace retenu représente bien le nuage. Cela dépendra en particulier du nombre de variables. Cependant, plus cette qualité est proche de 1, plus le nuage de points initial est "concentré" autour de E_k, et plus son image projeté sur E_k est fidèle.

```
X=data_temp <- read.csv("~/public_html/doc_cours/L3EURIA/AD/data_temp.txt",
                        sep= ' ',row.names = 1 )/10
library(FactoMineR)  #package utilisé dans la suite du cours
fit=PCA(X,graph=FALSE)  #réalisation de l'ACP
#l'option graph=FALSE empêche l'affichage des fenêtres graphiques
par(mfrow=c(1,2))
#pourcentage d'inertie totale expliquée par les composantes principales
barplot(100*fit$eig[,1]/sum(fit$eig[,1]),las=2)
#pourcentage d'inertie totale expliquée par les sous-espaces principaux
barplot(100*cumsum(fit$eig[,1])/sum(fit$eig[,1]),las=2)
```





La première direction principale explique 86.5% de l'inertie totale et le premier plan principal 99.1% de l'inertie totale. On perd donc très peu d'information en projetant sur le premier plan principal!

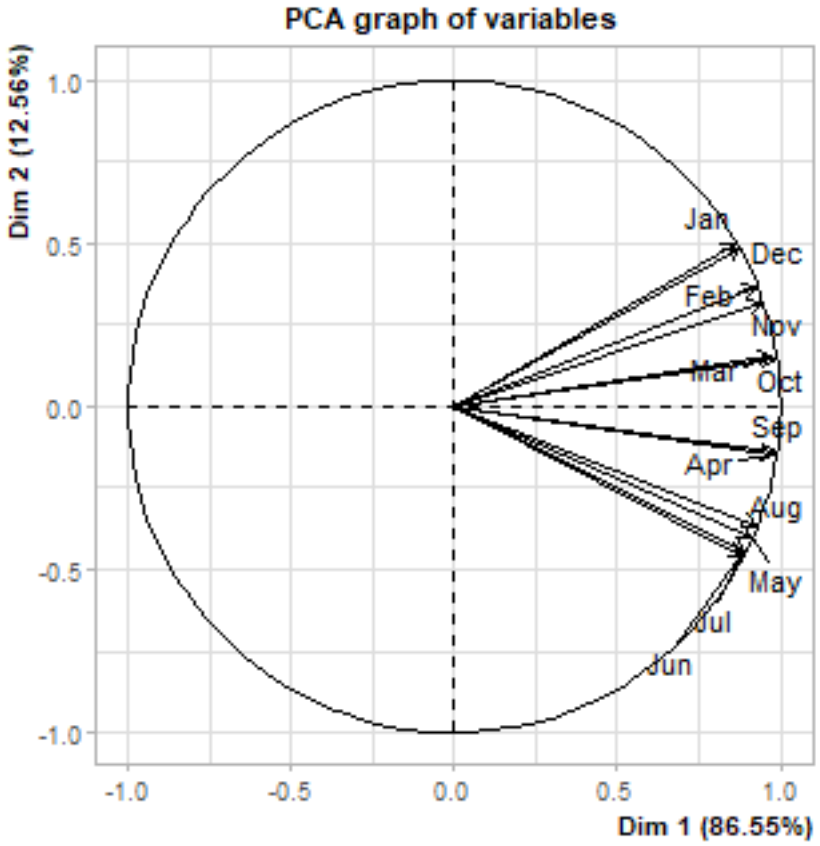
4.2 Corrélation entre composantes principales et variables initiales

La méthode la plus naturelle pour donner une signification à une composante principale $C_{.,k}$ est de la relier aux variables initiales $X_{.,j}$ en calculant les coefficients de corrélation linéaire $cor(C_{.,k}, X_{.,j})$. Remarquons qu'une forte corrélation entre $C_{.,k}$ et $X_{.,j}$ signifie que les individus ayant une coordonnée fortement positive sur le k-ième axe sont caractérisés par une valeur de la j-ième variable nettement supérieure à la moyenne (nb : les données sont supposées centrées).

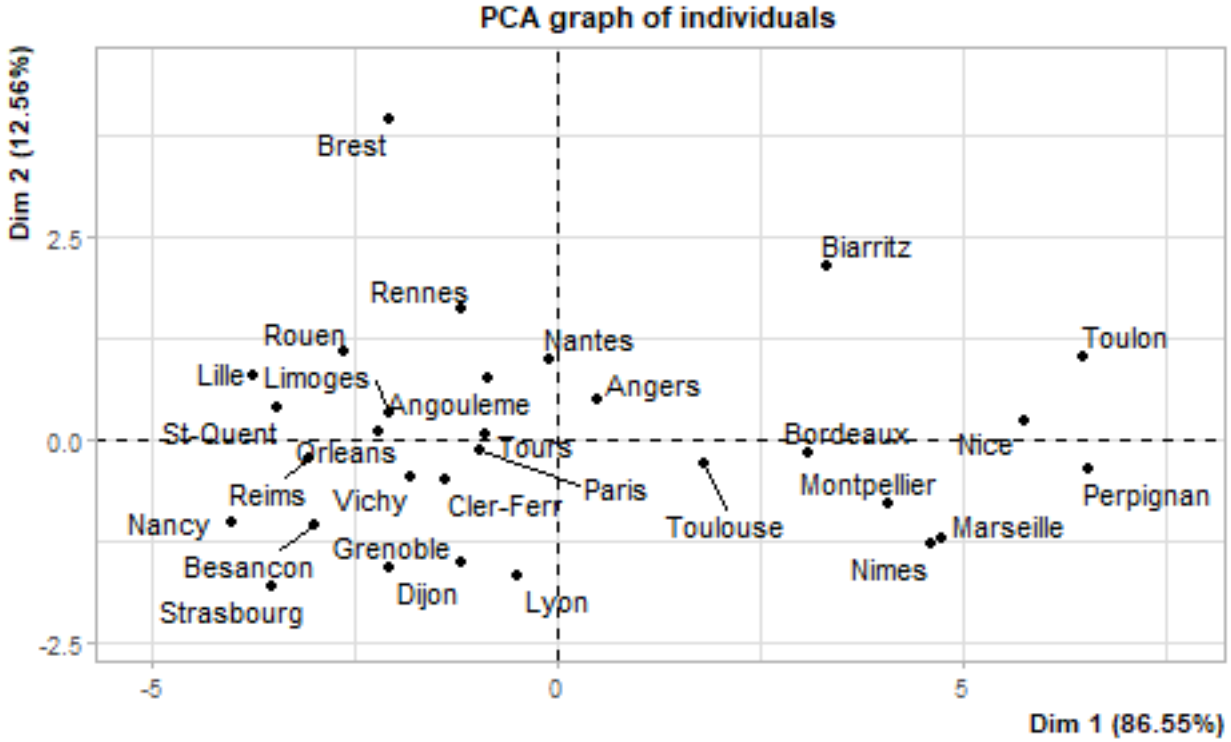
Pour interpréter les deux premières composantes principales, une représentation graphique courante est le **cercle des corrélations** : elle consiste à associer à chaque variable $X_{.,j}$ le point de coordonnées $(r(C_{.,1}, X_{.,j}), r(C_{.,2}, X_{.,j}))$, qui se trouve à l'intérieur du disque unité (cf exercice 6).

On essaie alors d'interpréter simultanément le cercle des corrélations et la position des individus dans le premier principal.

```
par(mfrow=c(1,2))
fit=PCA(X,graph=FALSE)  #réalisation de l'ACP sans graphique
par(mfrow=c(1,2))
plot(fit,choix="var") #cercle des corrélations
```




```
plot(fit,choix="ind") #position des individus dans le premier plan principal
```



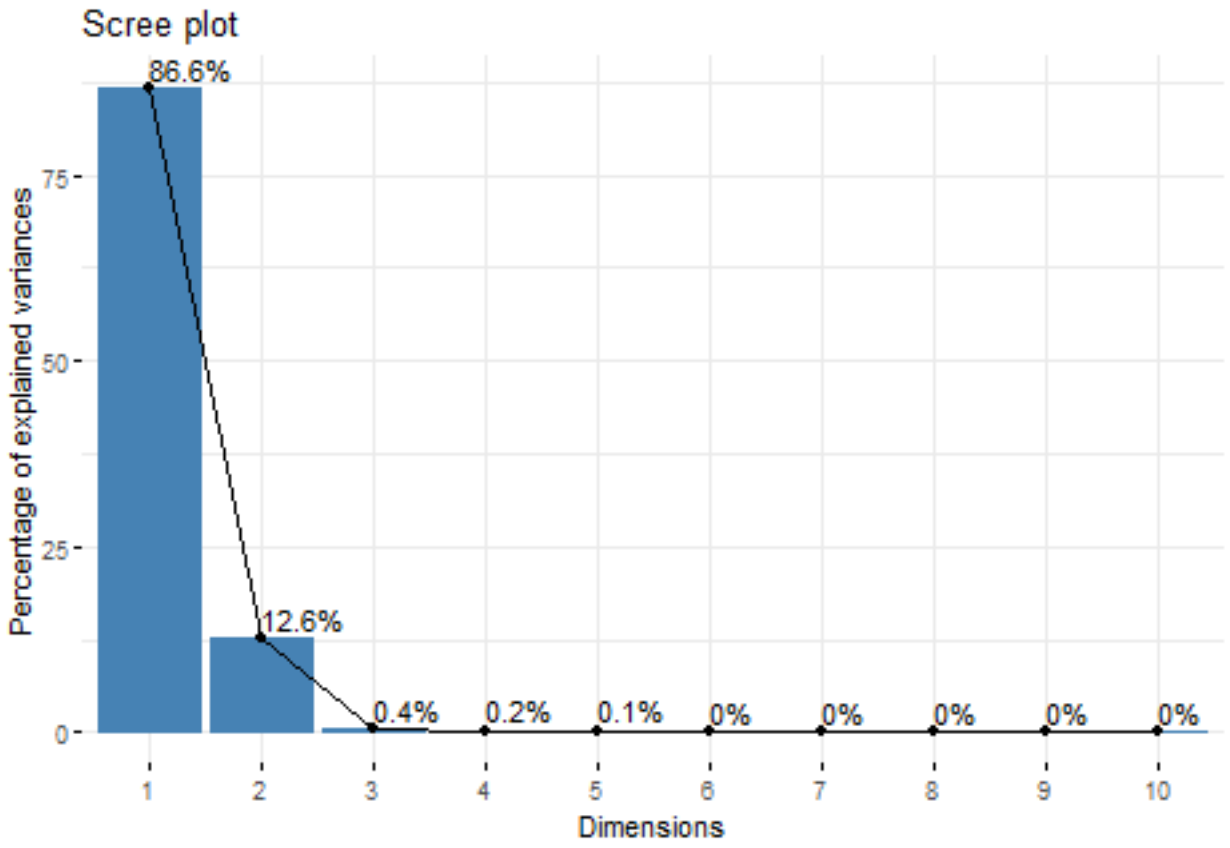
Il est parfois pratique de représenter les variables et les individus sur le même graphique pour faciliter l'interprétation. Ceci peut se faire avec le package factoextra

```
library("factoextra")  #package supplémentaire pour visualiser les résultats
```

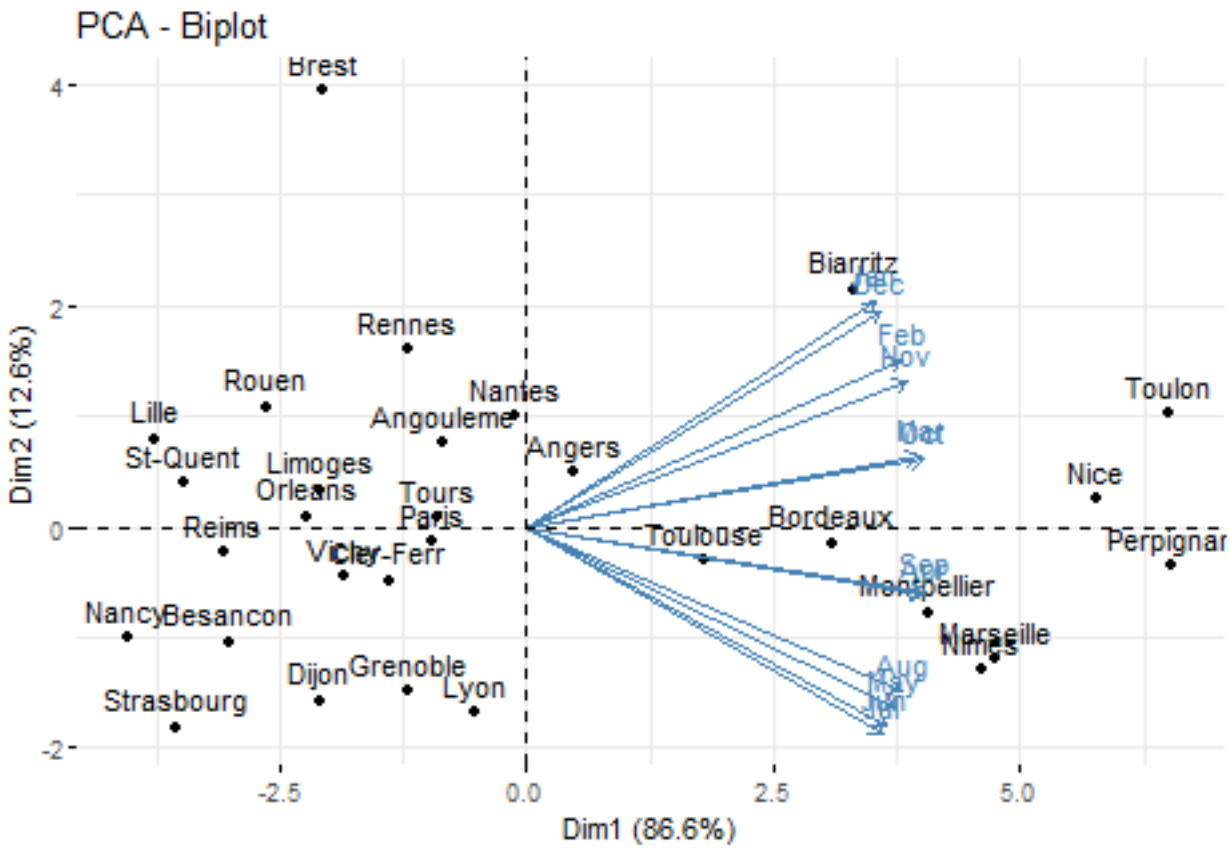
```
## Loading required package: ggplot2
```

```
## Welcome! Want to learn more? See two factoextra-related books at https://goo.gl/ve3WBa
```

```
fviz_eig(fit, addlabels = TRUE)  #valeurs propres
```




```
fviz_pca_biplot(fit) #biplot
```



Il est important que vous sachiez interpréter ce type de graphiques.

La première composante principale est corrélée positivement avec toutes les variables initiales. Les variables initiales sont les températures sur les différents mois après centrage, c'est à dire la différence (ou 'anomalie') entre la température de la ville considérée et la température moyenne en France. La première composante principale prendra donc une valeur positive pour une ville donnée si les températures sont au dessus de la moyenne nationale toute l'année (climat plus chaud que la moyenne), et une valeur négative si les températures sont au dessous de la moyenne nationale toute l'année (climat plus froid que la moyenne). Les villes qui se trouvent à droite sur le premier graphique (Toulon, Nice, Bordeaux,...) correspondent à des individus pour lesquels la première composante prend une grande valeur et donc à un climat chaud. Les villes qui se trouvent à gauche (Lille, Reims,...) sont des villes avec un climat plus froid.

En résumé, le premier axe principal 'oppose' les villes avec climat chaud aux villes avec un climat froid. Il explique 85.5% de l'inertie totale.

La deuxième composante principale est corrélée positivement avec les températures en hiver et négativement avec les températures en été. La deuxième composante principale prendra donc une valeur positive pour une ville donnée si les températures sont au dessus de la moyenne nationale en hiver (hiver doux) et en dessous de la température nationale en été. Les villes qui se trouvent en haut sur le premier graphique (Brest, Biarritz,...) correspondent à des individus pour lesquels le climat est tempéré (ou océanique) : peu de différence entre l'été et l'hiver. Les villes qui se trouvent en bas (Strasbourg, Lyon) sont des villes avec des différences plus importantes entre l'été et l'hiver (climat continental).

En résumé, le deuxième axe principal 'oppose' les villes avec un climat tempéré et les villes avec un climat continental. Il explique 12.5% de l'inertie totale.

Exercice 6 *L'objectif de cet exercice est de montrer que le point de coordonnées $(cor(C_{.,1}, X_{.,j}), cor(C_{.,2}, X_{.,j}))$ est dans le disque unité. On reprend les notations du cours.*

1. *Vérifier que $X_{.,j} = Xe_j$ avec $e_j = (0, ..0, 1, 0, ..., 0)'$ le vecteur colonne de longueur p qui a tous ses coefficients nuls sauf celui en position j qui vaut 1. En déduire que $\frac{1}{n}(C_{.,i}, X_{.,j}) = u_i' V e_j$*
2. *On note $u_i = (u_i(1), u_i(2), ..., u_i(p))'$ les coefficients du vecteur u_i. Déduire de la question précédente que $cov(C_{.,i}, X_{.,j}) = \lambda_i u_i(j)$ et $cor(C_{.,i}, X_{.,j}) = \sqrt{\lambda_i} \frac{u_i(j)}{sqrt(var(X_{.,j}))}$.*
3. *On note $U_{j,.} = (u_1(j), ..., u_p(j))$ la jème de la matrice U. Vérifier que $U_{j,.} = e_j' U$ et que que $\sum_{i=1}^p \lambda_i u_i(j)^2 = U_{j,.} D U_{j,.}'$. En déduire que $\sum_{i=1}^p \lambda_i u_i(j)^2 = var(X_{.,j})$.*
4. *En déduire que $\sum_{i=1}^p cor(C_{.,i}, X_{.,j})^2 = 1$ puis que le point de coordonnées $(cor(C_{.,1}, X_{.,j}), cor(C_{.,2}, X_{.,j}))$ est à l'intérieur du cercle unité.*
5. *Montrer que $X = CU'$ et en déduire que $X_{.,j} = \sum_{i=1}^p u_i(j) C., i$.*
6. *On suppose dans cette question que le point de coordonnées $(cor(C_{.,1}, X_{.,j}), cor(C_{.,2}, X_{.,j}))$ est sur le cercle unité. En déduire que $u_i(j) = 0$ pour $i \in \{3, ..., p\}$ puis que $X_{.,j} = u_1(j) C., 1 + u_2(j) C., 2$. Interprétez.*

Correction de l'exerice 6

1. *Un calcul matriciel direct montre que $X_{.,j} = Xe_j$. De la même manière, on a $C_{.,i} = Xe_i$. On en déduit que $(C_{.,i}, X_{.,j}) = e_i' C' X e_j$. On déduit alors facilement le résultat en utilisant successivement les relations $C = XU$, $u_i = Ue_i$, $V = \frac{1}{n} X'X$.*
2. *Comme le tableau de données est centré, on a*

$$cov(C_{.,i}, X_{.,j}) = \frac{1}{n} \sum_{k=1}^n C_{k,i} X_{k,j} = \frac{1}{n} (C_{.,i}, X_{.,j})$$

. D'après la question précédente, on en déduit que

$$cov(C_{.,i}, X_{.,j}) = \frac{1}{n} u_i' V e_j = \frac{1}{n} (V' u_i)' e_j.$$

V est une matrice symétrique et u_i est un vecteur propre de V. On en déduit que

$$V' u_i = V u_i = \lambda_i u_i.$$

Finalement, on en déduit que

$$cov(C_{.,i}, X_{.,j}) = \lambda_i u_i' e_j$$

puis le résultat demandé.

3. *Les relations $U_{j,.} = e_j' U$ et $\sum_{i=1}^p \lambda_i u_i(j)^2 = U_{j,.} D U_{j,.}'$ se vérifient par un simple calcul matriciel. On en déduit que*

$$\sum_{i=1}^p \lambda_i u_i(j)^2 = e_j' U D U' e_j.$$

On en déduit le résultat demandé en utilisant les relations $V = UDU'$ et $e_j' V e_j = V_{j,j} = var(X_{.,j})$

4. *En utilisant le résultat de la question 2., on obtient*

$$\sum_{i=1}^p cor(C_{.,i}, X_{.,j})^2 = \frac{\sum_{i=1}^p \lambda_i u_i(j)^2}{var(X_{.,j})}.$$

Le résultat de la question 3. nous donne alors que $\sum_{i=1}^p cor(C_{.,i}, X_{.,j})^2 = 1$.

On en déduit que

$$\sum_{i=1}^2 cor(C_{.,i}, X_{.,j})^2 \leq \sum_{i=1}^p cor(C_{.,i}, X_{.,j})^2 = 1$$

et donc que le point de coordonnées $(cor(C_{.,1}, X_{.,j}), cor(C_{.,2}, X_{.,j}))$ est à l'intérieur du cercle unité.

5. *D'après le cours, on a $C = XU$ avec $UU' = I_p$. On en déduit que $CU' = XUU' = X$. En faisant le produit matriciel CU', on vérifie que $X_{.,j} = \sum_{i=1}^p u_i(j) C., i$.*

6. *On suppose dans cette question que le point de coordonnées* $(cor(C_{.,1}, X_{.,j}), cor(C_{.,2}, X_{.,j}))$ *est sur le cercle unité, c'est à dire que* $\sum_{i=1}^{2} cor(C_{.,i}, X_{.,j})^2 = 1$. *On en déduit de la relation* $\sum_{i=1}^{p} cor(C_{.,i}, X_{.,j})^2 = 1$ *que* $\sum_{i=3}^{p} cor(C_{.,i}, X_{.,j})^2 = 0$ *et donc que* $cor(C_{.,i}, X_{.,j}) = 0$ *pour* $i \geq 3$. *En utilisant le résultat de la question 2., on en déduit que* $u_i(j) = 0$ *pour* $i \geq 3$ *puis, en utilisant le résultat de la question 5., que* $X_{.,j} = \sum_{i=1}^{2} u_i(j) C_{.,i}$. *Cette ralation nous indique que si le point de corrdonnées* $(cor(C_{.,1}, X_{.,j}), cor(C_{.,2}, X_{.,j}))$ *est sur le cercle unité, alors la jème variable* $X_{.,j}$ *peut s'écrire comme une combinaison linéaire des deux premières composantes principales* $C_{.,1}$ *et* $C_{.,2}$. *En pratique, cela signifie qu'on ne perd pas d'information sur la jème variable en projetant sur les deux premiers axes principaux de l'ACP.*

5 Analyse détaillée des résultats d'une ACP

5.1 Qualité de la représentation des individus

Rappel. Si u et v sont deux vecteurs non nuls de $\mathbb{R}^p$, on définit $\cos(u,v) = \dfrac{(u,v)}{\| u \| \times \| v \|}$). En particulier, si les vecteurs u et v sont colinéaires alors $\cos(u,v) = +-1$ (c'est à dire $\cos(u,v)^2 = 1$) et le projeté orthogonal de v sur $vect(u)$ est égal à v (interprétation en analyse de données : on ne perd pas d'information sur l'individu v en projetant dans la direction u). A l'opposé si $\cos(u,v)^2 = 0$ alors le projeté orthogonal de v sur $vect(u)$ est égal à 0 (interprétation en analyse de données : la direction u n'apporte aucune information sur l'individu v).

Définition 5 *La qualité de la représentation de l'individu* k *dans la direction principale* $I_{vect(u_i)}$ *se mesure par le cosinus carré de l'angle entre les vecteurs :*

$$COS2(k,i) = \cos(X_{k,.}, u_{i,})^2 = \frac{(X_{k,.}, u_i)^2}{\| X_{k,.} \|^2}$$

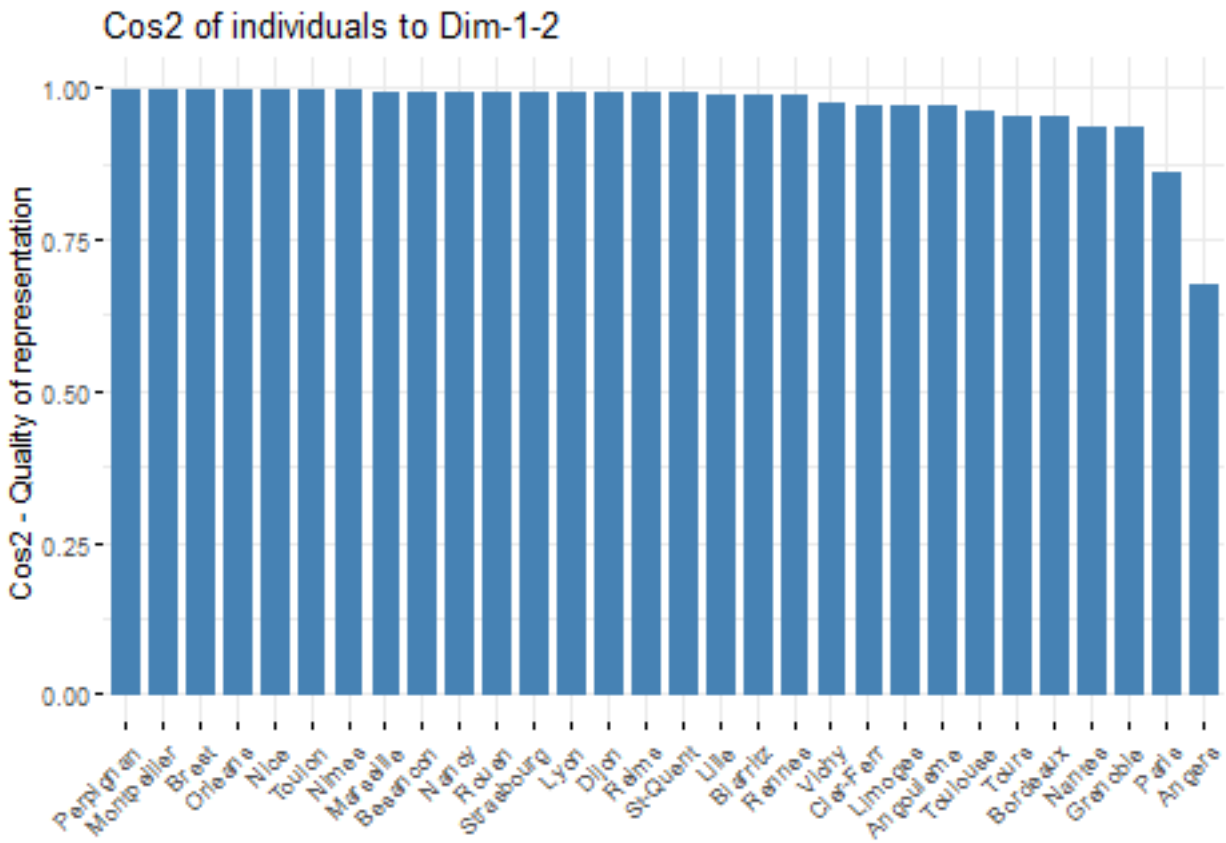
Plus globalement, la qualité de la représentation de l'individu $X_{k,.}$ *par le sous-espace principal de dimension* l E_l *engendré par les* l *premiers axes factoriels peut se mesurer par le carré du cosinus de l'angle entre* $X_{k,.}$ *et son projeté orthogonal* $\pi_{E_l}(X_{k,.})$ *sur* E_l. *Comme* $(u_1, ..., u_l)$ *est une base orthonormée de* E_l, *on a* $\pi_{E_l}(X_{k,.}) = \sum_{i=1}^{l} c_k^i u_i$. *On en déduit que*

$$cos(X_{k,.}, \pi_{E_l}(X_{k,.})) = QLT(k,l) = \frac{\| f_k^l \|^2}{\| X_{k,.} \|^2} = \sum_{i=1}^{l} \frac{(X_{k,.}, u_i)^2}{\| X_{k,.} \|^2} = \sum_{i=1}^{l} COS2(k,i)$$

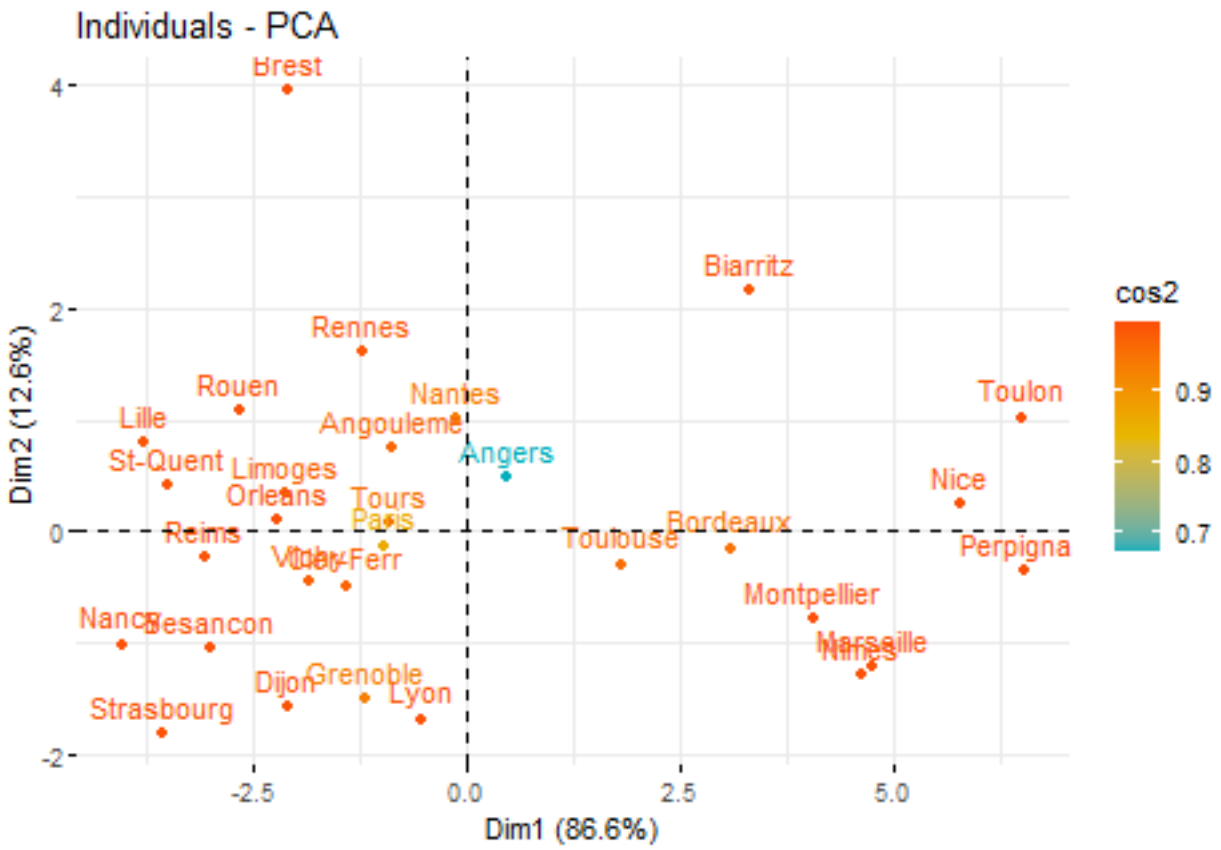
Un individu $X_{k,.}$ est bien représenté sur E_l si $QLT(k,l) \in [0,1]$ a une valeur proche de 1. Il faut se méfier cependant des individus proches du centre de gravité (ie proche de 0 dans le cas centré). Les vecteurs de norme faible sont nécessairement proches de leur projection sur E_l, quelle que soit la valeur de l'angle. Par définition, on a $QLT(p,l) = 1$.

Illustration sur l'exemple des données de température.

```
#calcul de la qualité de la représentation sur le premier plan principal
QTL2=apply(fit$ind$cos2[,1:2],1,sum)
#barplot(QTL2,las=2)  #las=2 permet de mettre le nom des villes verticalement
fviz_cos2(fit, choice = "ind", axes = 1:2)
```




```
fviz_pca_ind(fit, col.ind = "cos2", #représentation alternative
             gradient.cols = c("#00AFBB", "#E7B800", "#FC4E07"),
             )
```



Tous les individus sont assez bien représentés sur le premier plan principal. Angers est la ville qui est la moins bien représentée sur le premier plan principal.

5.2 Contributions des individus

Il s'agit de détecter les individus 'influents' ou 'aberrants' qui peuvent déterminer à eux seuls l'orientation des axes, et plus globalement l'ensemble des résultats de l'ACP.

Définition 6 *L'inertie du nuage projeté sur le jème axe principal est donné par*

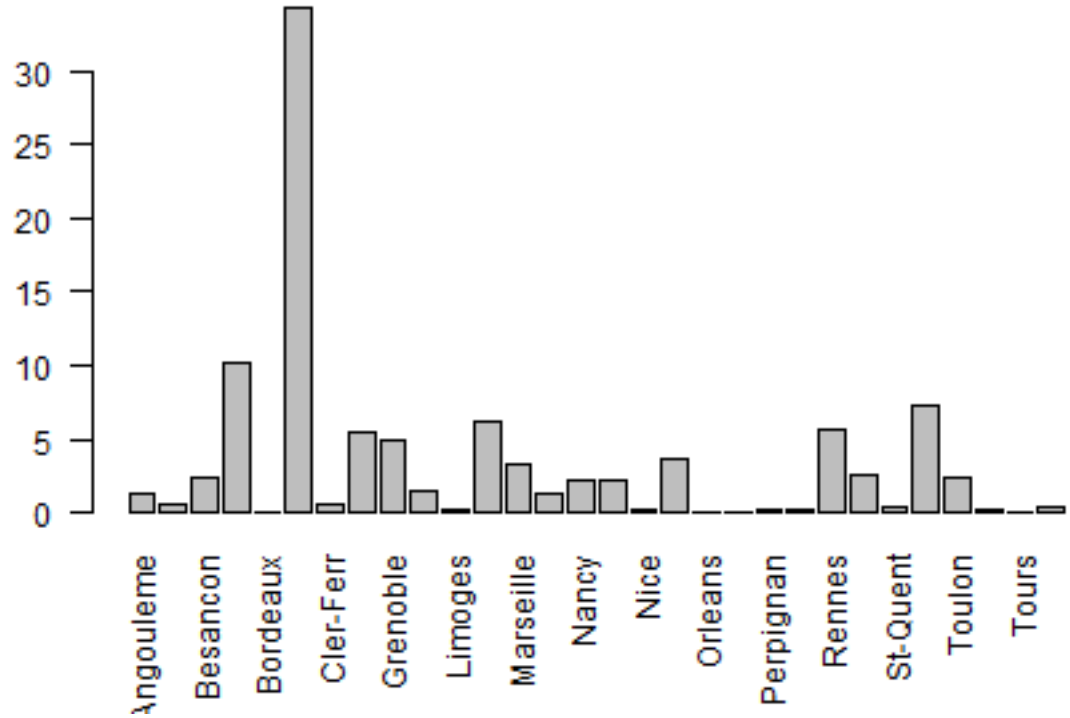
$$\lambda_j = \frac{1}{n}\sum_{i=1}^{n} c_{i,j}^2.$$

La contribution du kième individu au jème axe principal est alors définie par

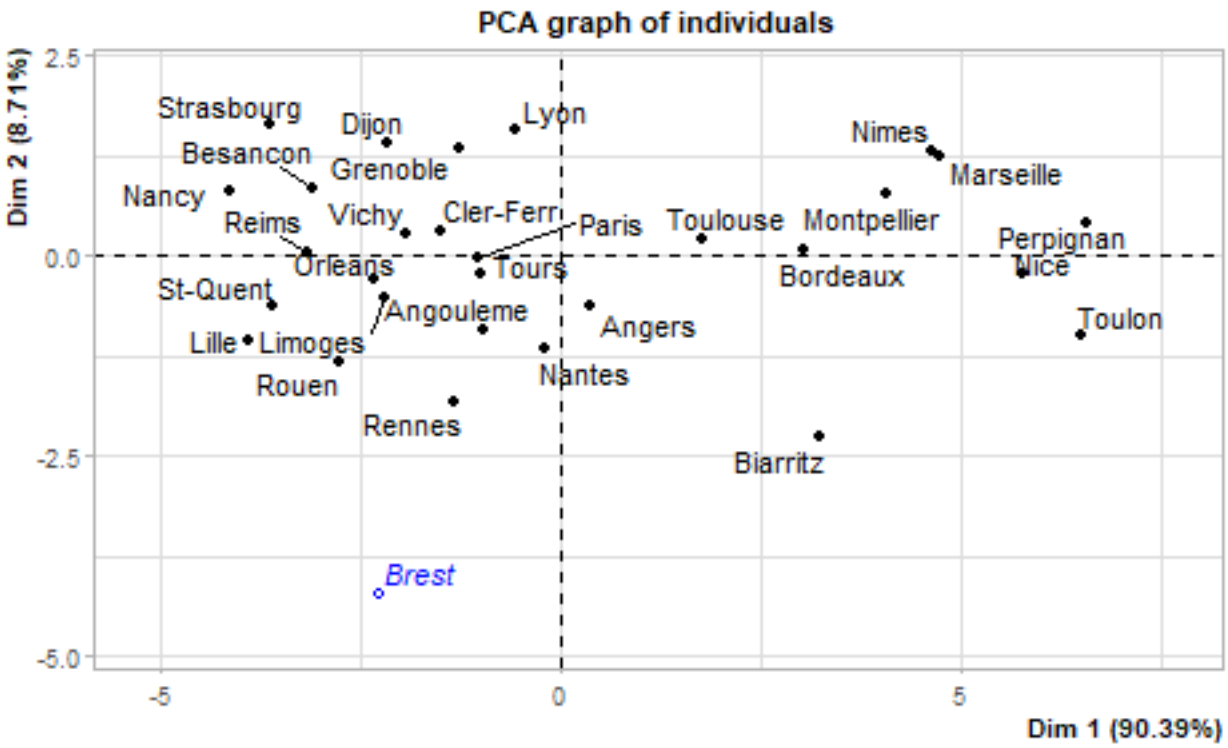
$$CTR(k,j) = \frac{100}{n}\frac{c_{k,j}^2}{\lambda_j}.$$

Par définition, on a $\sum_{k=1}^{n} CTR(k,j) = 100$ et un individu k qui a une coordonnée $c_{k,j}$ grande sur l'axe principale j aura une contribution importante sur cet axe. Il faut se méfier si un individu a une contribution excessive, car cela est un facteur d'instabilité : retirer cet individu modifie profondément le résultat de l'analyse. On peut réaliser une ACP en le retirant et considérer alors sa projection dans les sous-espaces principaux.

```
barplot(fit$ind$contrib[,2],las=2)
```




```
#brest a une forte contribution sur le deuxième axe
par(mfrow=c(1,2))
fit2=PCA(X,ind.sup = which(row.names(X)=='Brest'))
```



Dim 2 (8.71%)

```
#ACP sans Brest
#Brest est quand même représenté dans le premier plan principal
#avec une couleur différente.
```

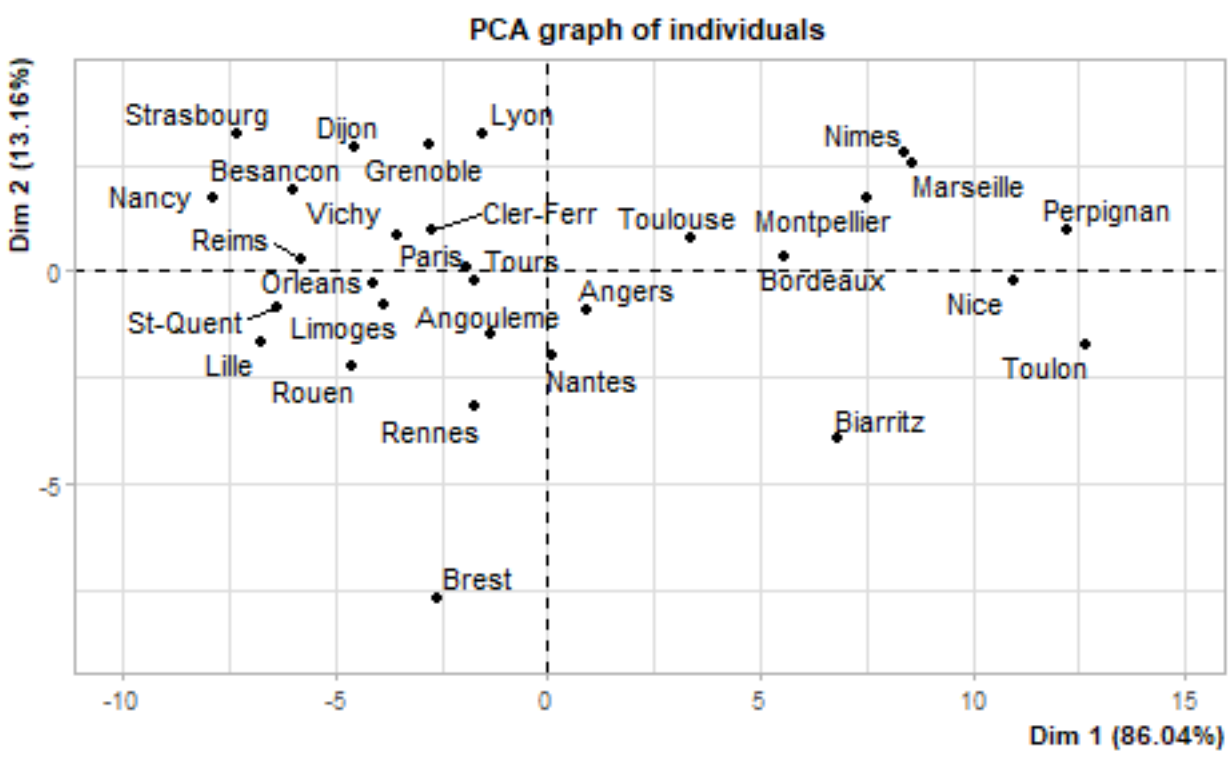
Remarque 3 *On peut définir de la même manière la qualité de la représentation et les contributions des variables.*

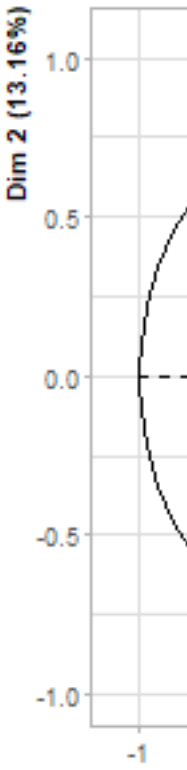
6 L'ACP en pratique

Ce chapitre discute différentes options possibles lorsqu'on réalise une ACP. Il reprend les différentes options de la fonction PCA du package FactoMineR de R. Après la lecture de ce chapitre, vous devez avoir compris les différents arguments d'entrée de la fonction PCA (disponibles avec la commande R ?PCA) et savoir les utiliser.

ACP normée. Le plus souvent les variables sont hétérogènes (elles n'ont pas les mêmes unités de mesure par ex.). On travaille alors généralement sur le tableau de données centré-réduit, c'est à dire en faisant la transformation $x_{i,j} < -\frac{x_{i,j}-\bar{x}_j}{s_j}$ ce qui revient à chercher les éléments propres de la matrice de corrélation au lieu de ceux de la matrice de covariance. On parle alors d'**ACP réduite** ou d'**ACP normée**. Le résultat est alors indépendant des unités utilisées (option scale.unit sous R avec la fonction PCA).

```
par(mfrow=c(1,2))
PCA(X,scale.unit=FALSE)  #ACP non normee
```



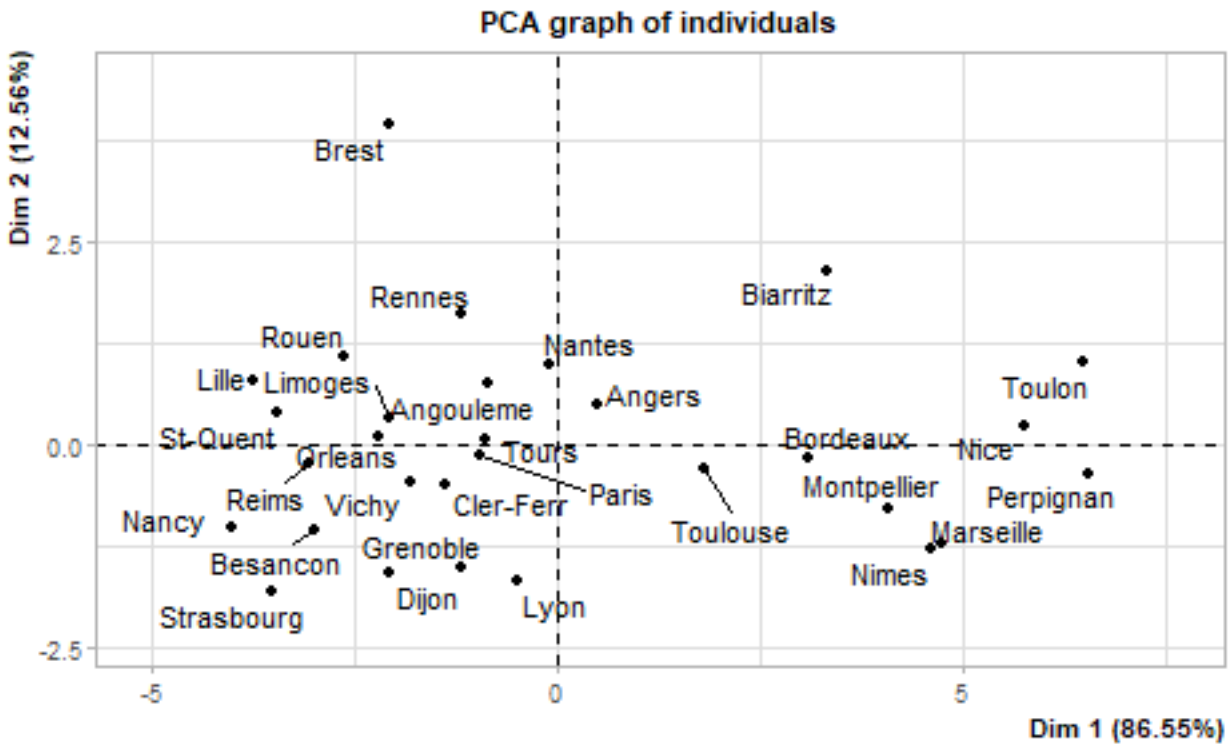



```
#NB : par défaut R fait une ACP normée
```

Les données de températures étant homogènes (écart-types similaires pour tous les mois), l'ACP normée et l'ACP non-normée donnent des résultats similaires.

Variables supplémentaires. On peut mettre certaines variables (quantitatives ou qualitatives) en variables supplémentaires (options quali.sup et quanti.sup sous R avec la fonction PCA). Ces variables ne sont alors pas utilisées pour construire les axes principaux, mais elles sont quand même représentées dans le plan principal.

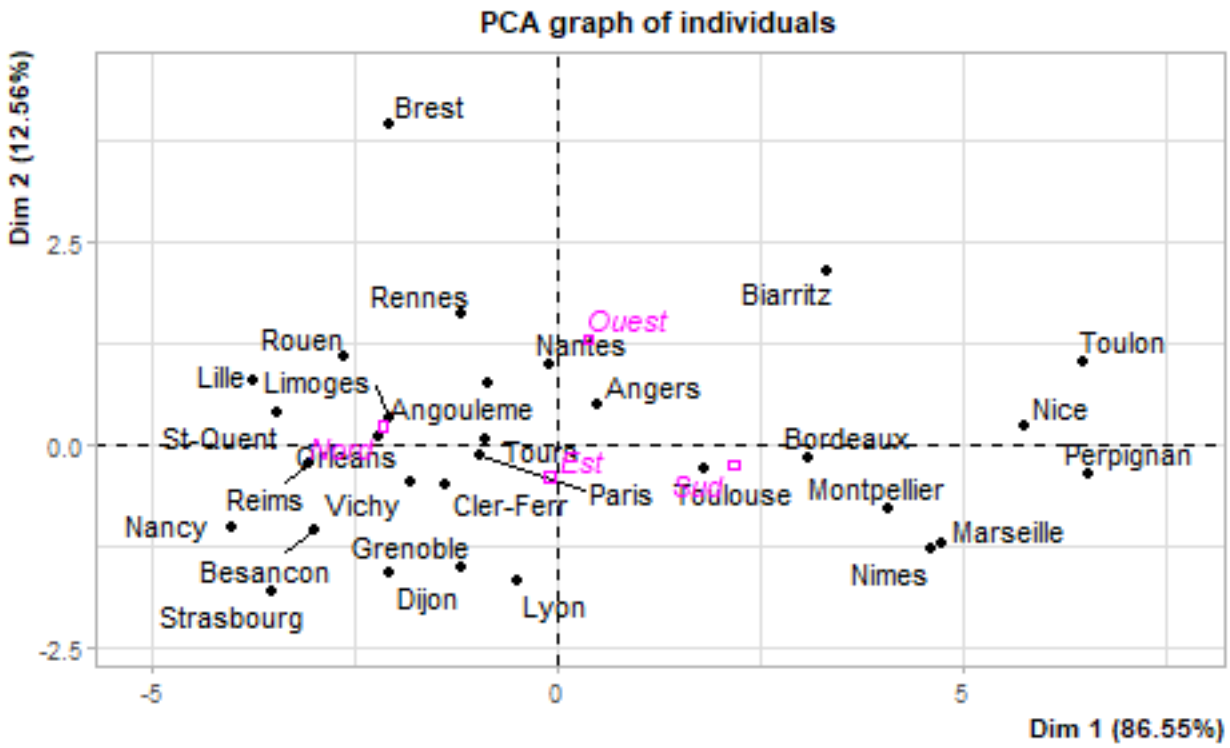
```
coord <- read.csv("~/public_html/doc_cours/L3EURIA/AD/latlon.txt",sep= ' ',row.names = 1 )
X2=cbind(X,coord)  #ajout des variables lon et lat
par(mfrow=c(1,2))
PCA(X2,quanti.sup=c(13,14))
```




```
#acp avec lon et lat en variables quantitatives supplémentaires
```

Les variables quantitatives supplémentaires sont représentées dans le cercle des corrélations en calculant les corrélations avec les variables créées par l'ACP. On voit que le premier axe est corrélé négativement avec la latitude (interprétation : il fait plus froid au nord) et que le deuxième axe est corrélé positivement avec la longitude (interprétation : hiver plus doux dans les villes de l'ouest de la France).

```
OUEST=rep('Est',nrow(X))
OUEST[coord$Lon>0]='Ouest'
#variable binaire Est/Ouest
#'Ouest' pour les villes qui sont à l'ouest
#du méridien de Greenwich
NORD=rep('Sud',nrow(X))
NORD[coord$Lat>47]='Nord'
#variable binaire Nord/Sud
#'Nord' pour les villes qui sont au nord
#de la lattitude 47
X3=data.frame(X,OUEST,NORD)  #ajout des variables créées
par(mfrow=c(1,2))
PCA(X3,quali.sup=c(13,14))
```

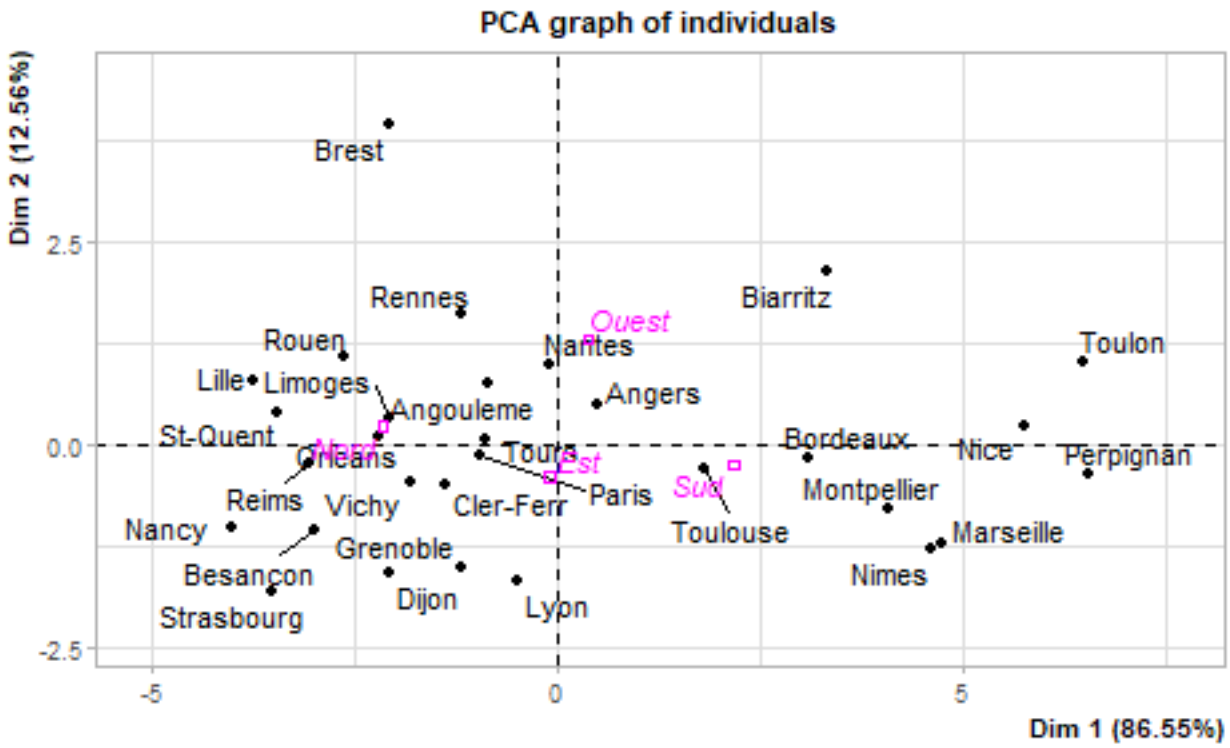



```
#acp avec OUEST et NORD en variables quantitatives supplémentaires
```

Les variables qualitatives supplémentaires sont représentées en projetant le centre de gravité des différentes modalités dans le premier plan principal. Ici on voit donc que la position des villes 'Nord' et 'Sud' dans le premier plan principal : la moyenne des villes 'Sud' est à droite sur le premier axe principal (interprétation : il fait plus chaud au sud de la France) alors que la moyenne des villes 'Nord' se retrouve à gauche (interprétation : il fait plus froid au nord de la France).

On peut mélanger l'ajout de variables quantitatives et qualitatives supplémentaires comme dans l'exemple ci-dessous.

```
X3=data.frame(X,coord$Lat,coord$Lon,OUEST,NORD)  #ajout des variables créées
par(mfrow=c(1,2))
PCA(X3,quanti.sup=c(13,14),quali.sup=c(15,16))
```

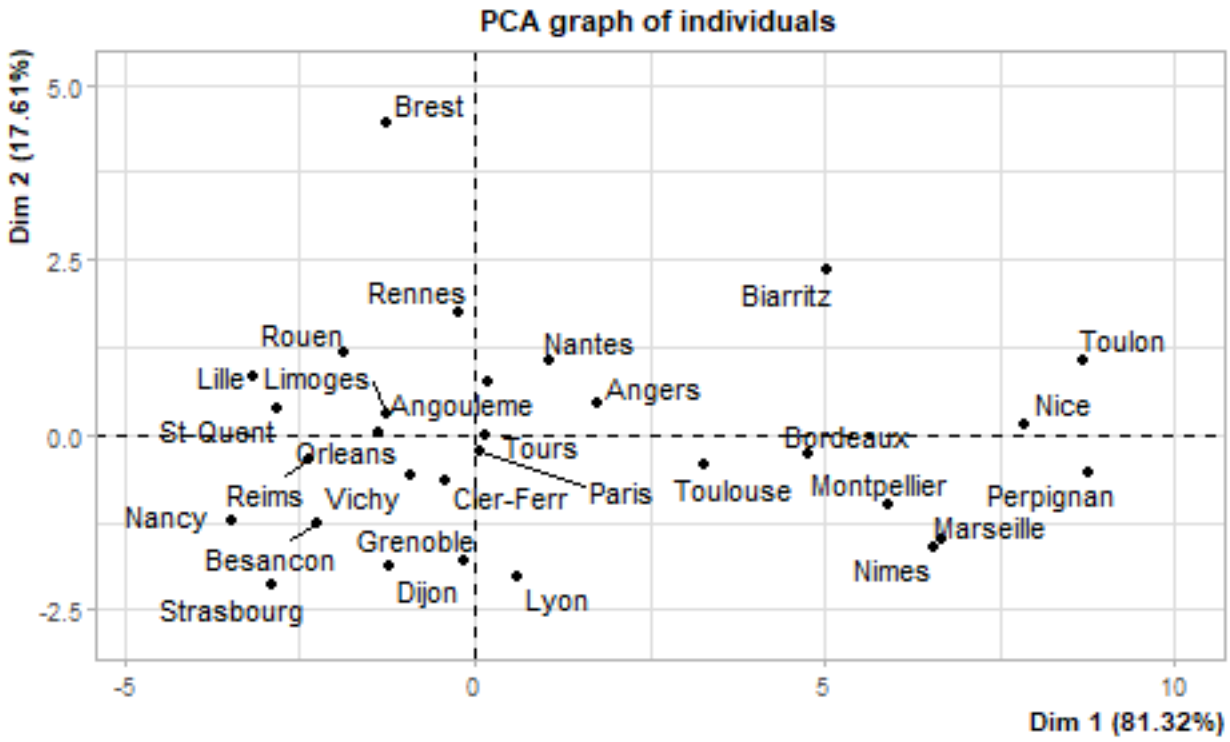



```
#acp avec variables quanti et quali summplémentaires
```

Individus supplémentaires. on peut mettre certains individus en individu supplémentaire, par exemple ceux qui ont une contribution excessive (option ind.sup). Un exemple est donné dans le paragraphe sur les contributions.

Pondération des individus. on peut rajouter des poids sur les individus (option row.w sous R avec la fonction PCA). Par exemple, si les individus sont des pays avec des populations différentes, on peut pondérer les pays par le nombre d'habitants (cf TP2).

```
poids=coord$Lat-40
#poids qui dépend de la latitude
par(mfrow=c(1,2))
PCA(X,row.w=poids)
```



Ici on a réalisé l'ACP en donnant plus de poids aux villes du nord. Cela va modifier les axes principaux de telle manière à mieux expliquer les villes qui ont un poids plus important.

7 Méthodes de classification non supervisée

7.1 Introduction

L'ACP permet de décrire les ressemblances, les liaisons entre des variables ou des individus. Dans cette seconde partie du cours, nous allons voir comment regrouper/classifier en un certain nombre de classes (ou groupes) dans lesquelles les individus sont les plus semblables. Le but de la classification est d'obtenir une partition des individus telle que deux individus d'un même groupe se ressemblent le plus et deux individus de groupes distincts diffèrent le plus possible.

On parle de classification supervisée lorsque les classes sont connues a priori (e.g. présence/absence dans un objet avec des images 'labellisées') et de classification non supervisée lorsque les classes sont inconnues. Dans ce cours, nous allons nous intéresser seulement aux méthodes non-supervisées. Certaines méthodes supervisées (régression logistique et autres méthodes de machine learning) seront étudiées en M1.

7.2 Inerties interclasse et intraclasse

Dans ce chapitre $E = \{e_1, \cdots, e_n\}$ désigne un ensemble fini de n individus caractérisés par p variables (en pratique $e_i \in \boldsymbol{R}^p$ est une ligne du tableau de données X). d est la distance euclidienne sur $\mathbb{R}^p$.

On rappelle la définition de l'inertie totale du nuage des n points :

$$I(E) = \frac{1}{n} \sum_{e \in E} d(e, G)^2$$

où $G = \frac{1}{n} \sum_{e \in E} e$ est le centre de gravité de E. De manière plus générale, si $a \in \mathbb{R}^d$, on peut définir l'iertie au point a par

$$I_a(E) = \frac{1}{n} \sum_{e \in E} d(e,a)^2.$$

On vérifie la relation de Huyghens

$$\begin{aligned} I_a &= \frac{1}{n} \sum_{e \in E} d(e,a)^2 \\ &= \frac{1}{n} \sum_{e \in E} \| (e-G) + (G-a) \|^2 \\ &= \frac{1}{n} \sum_{e \in E} \| (e-G) \|^2 + (G-a) \|^2 + 2(e-G, G-a) \\ &= I_G + d(a,G)^2 \end{aligned}$$

car $\frac{1}{n} \sum_{e \in E} (e-G, G-a) = (G-G, G-a) = 0$.

Pour une partition de E en k classes $S = \{C_1, \cdots, C_k\}$, définissons l'inertie totale de chacune des classes par

$$I_{G_j}(C_j) = \frac{1}{n_j} \sum_{e \in C_j} d(G_j, e)^2$$

où $n_j = card(C_j)$, et $G_j = \frac{1}{n_j} \sum e \in C_j e$ est le centre de gravité de C_j.

Définition 7 *On appelle **inertie intraclasse** la moyenne pondérée des inerties de chacune des classes*

$$I_{intra}(S) = \frac{1}{n} \sum_{j=1}^{k} n_j I(C_j).$$

*La dispersion de l'ensemble des centres de gravités $G_1, \cdots, G_k$ autour du centre de gravité G se mesure par l'**inertie interclasse***

$$I_{inter}(S) = \frac{1}{n} \sum_{j=1}^{k} n_j d(G_j, G)^2$$

Remarques 2

- *une faible valeur de $I_{intra}(S)$ correspond à des classes homogènes, peu étendues autour de leurs centres de gravité respectifs.*
- *une grande valeur de $I_{inter}(S)$ indique une bonne séparation des classes.*

Proposition 4 *Pour toute partition S de E, on a*

$$I(E) = I_{inter}(S) + I_{intra}(S).$$

Preuve 4 ***Faire la preuve en exercice.***

$$\begin{aligned} I_{inter}(S) + I_{intra}(S) &= \frac{1}{n} \sum_{j=1}^{k} n_j (d(G,G_j)^2 + I_{G_j}(C_j)) \\ &= \frac{1}{n} \sum_{j=1}^{k} n_j I_G(C_j) \text{ (relation de Huyghens)} \\ &= \frac{1}{n} \sum_{j=1}^{k} \sum_{e \in C_j} d(G,e)^2 \\ &= \frac{1}{n} \sum_{e \in E} d(G,e)^2 \text{ (partition de E)} \\ &= I \end{aligned}$$

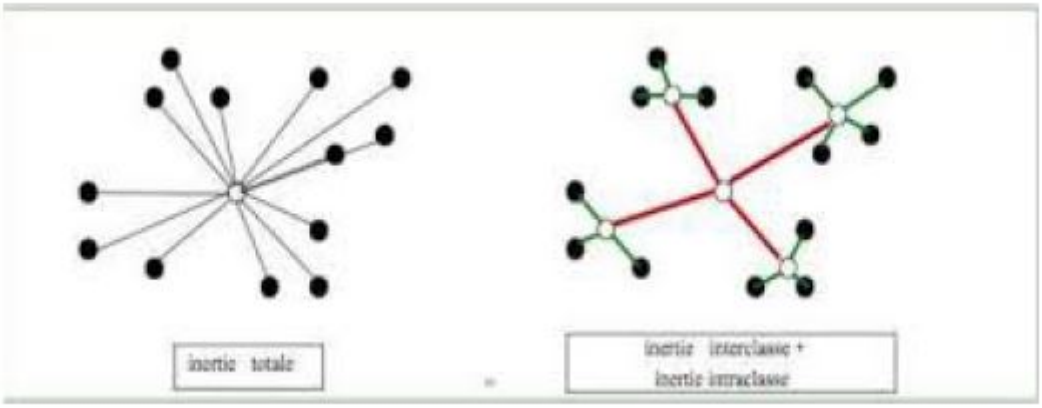


Figure 7: Inertie inter/intraclasse

Cas particulier de deux ensembles. On suppose que $E = C_1 \cup C_2$ avec $C_1 \cap C2 = \emptyset$, $n_i = card(C_i)$ pour $i \in \{1, 2\}$. On a $G = \frac{n_1 G_1 + n_2 G_2}{n1+n2}$ et donc

$$nI_{inter}(S) = n_1 \parallel G_1 - G \parallel^2 + n_2 \parallel G_2 - G \parallel^2 = 2\frac{n_1 n_2}{n_1 + n_2} d(G_1, G_2)^2.$$

Finalement, on obtient

$$I(C_1 \cup C_2) = \frac{1}{n_1 + n_2} \left(n_1 I(C_1) + n_2 I(C_2) + 2\frac{n_1 n_2}{n_1 + n_2} d(G_1, G_2)^2 \right)$$

7.3 La méthode de classification ascendante hiérarchique (CAH)

La méthode de **classification ascendante hiérarchique** consiste à construire une suite de partitions imbriquées $S_1, S_2, ..., S_n$, où S_i est une partition à $n - i + 1$ classes, en utilisant l'algorithme suivant :

Algorithme de CAH :

Soit $E = \{e_1, \cdots, e_n\}$ l'ensemble de n éléments à classer.

1. *Initialisation :* définir la partition initiale triviale $S_1 = \{\{e_1\}, \{e_2\}, \cdots, \{e_n\}\}$ (*ie* les classes initiales sont les singletons)
2. *Itération, jusqu'à ce que tous les points soient regroupés dans une seule classe:* regrouper les deux classes les plus proches

Pour pouvoir utiliser cet algorithme, on a besoin de pouvoir définir une 'distance' entre deux ensembles de points A et B (on parle de 'mesures de dissimiralité'). On peut par exemple considérer

- $d^*(A, B) = \min_{i \in A, j \in B} d(i, j)$ saut minimum (single linkage),
- $d^*(A, B) = \max_{i \in A, j \in B} d(i, j)$ saut maximum (complete linkage)
- $d^*(A, B) = \frac{1}{Card(A)Card(B)} \sum_{i \in A, j \in B} d(i, j)$ saut moyen (average linkage),
- $d^*(A, B) = \frac{n_A n_B}{n_A + n_B} d(G_A, G_B)^2$ saut de Ward .

Remarques

- Il n'y a pas toujours unicité, le minimum (distance la plus proche entre deux classes) n'étant pas nécessairement atteint par un seul couple de classes. On doit convenir d'une règle dans ce cas.

- Le saut de Ward joue un rôle particulier et est la stratégie la plus courante. En effet, ce critère induit, à chaque étape de regroupement, une minimisation de la variance intraclasse parmi les partitions possibles. En effet, soit $S = \{C_1, \cdots, C_k\}$ une partition de E et $\tilde{S}$ la partition obtenue en regroupant les classes C_i et C_j. On a

$$I_{intra}(S) = \frac{1}{n}\sum_{l=1}^{k} n_l I(C_l).$$

et

$$I_{intra}(\tilde{S}) = \frac{1}{n}\left(\sum_{l=1;l\neq i,j}^{k} n_l I(C_l) + (n_i + n_j) I_{C_i \cup C_j}\right)$$

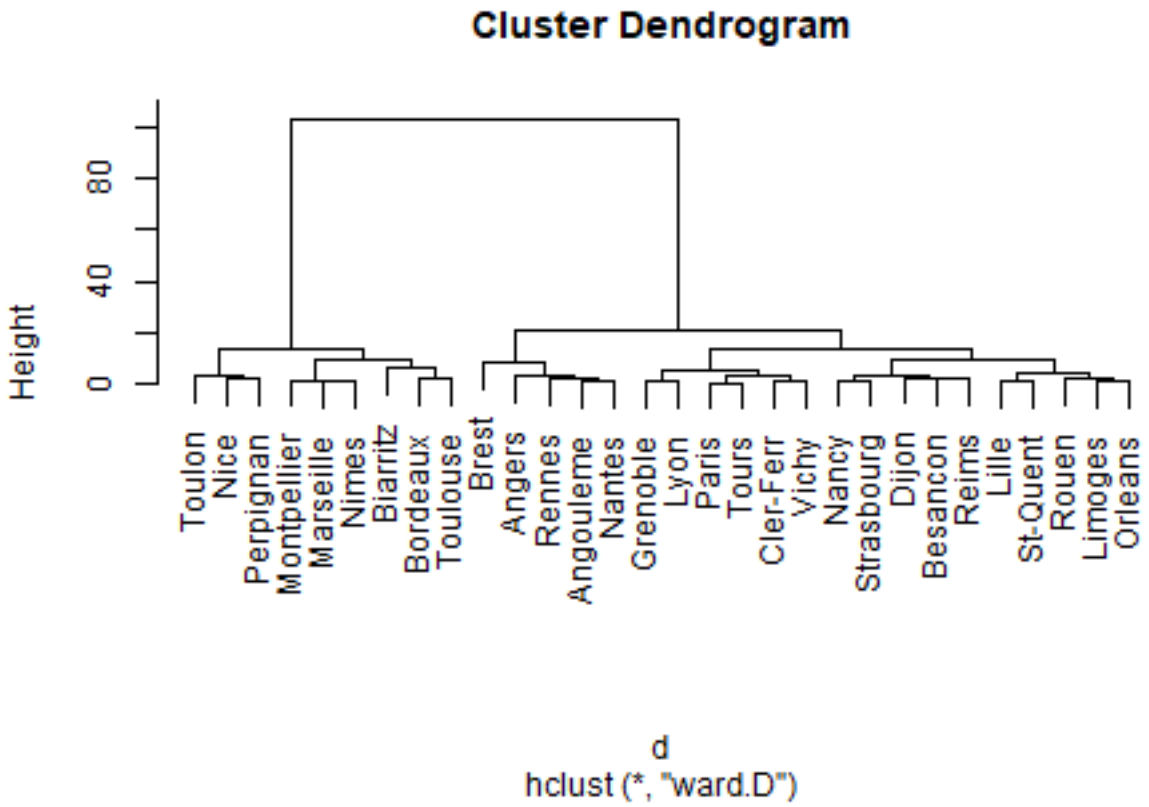
avec $(n_i + n_j) I_{C_i \cup C_j} = n_i I(C_i) + n_j I(C_j) + 2\frac{n_i n_j}{n_i + n_j} d(G_i, G_j)^2$ et donc

$$I_{intra}(\tilde{S}) = I_{intra}(S) + 2\frac{n_i n_j}{n_i + n_j} d(G_i, G_j)^2$$

- La suite de partitions obtenue peut se représenter graphiquement par un arbre hiérarchique indicé appelé **dendrogramme**. La hauteur des branches correspond à la valeur du saut d^* quand on regroupe les classes.

  L'observation de cet arbre peut aider notamment pour choisir un nombre de classes judicieux. En pratique, il faut couper à un endroit où il y a un saut important, ce qui correspond au regroupement de deux classes 'éloignées'. Sur l'exemple ci-dessous, la hauteur des deux dernières branches est grande, ce qui suggère qu'on a deux groupes de villes bien distincts (les villes du sud (Toulon, Nice,..., Toulouse) et les villes du nord (Brest,...,Orléans)). On pourrait aussi faire 3 classes (autre saut important). **Exercice : vérifier que vous êtes vous capables d'identifier les trois classes correspondantes sur le dendrogramme et que vous retrouvez les résultats donnés par la fonction cut**.

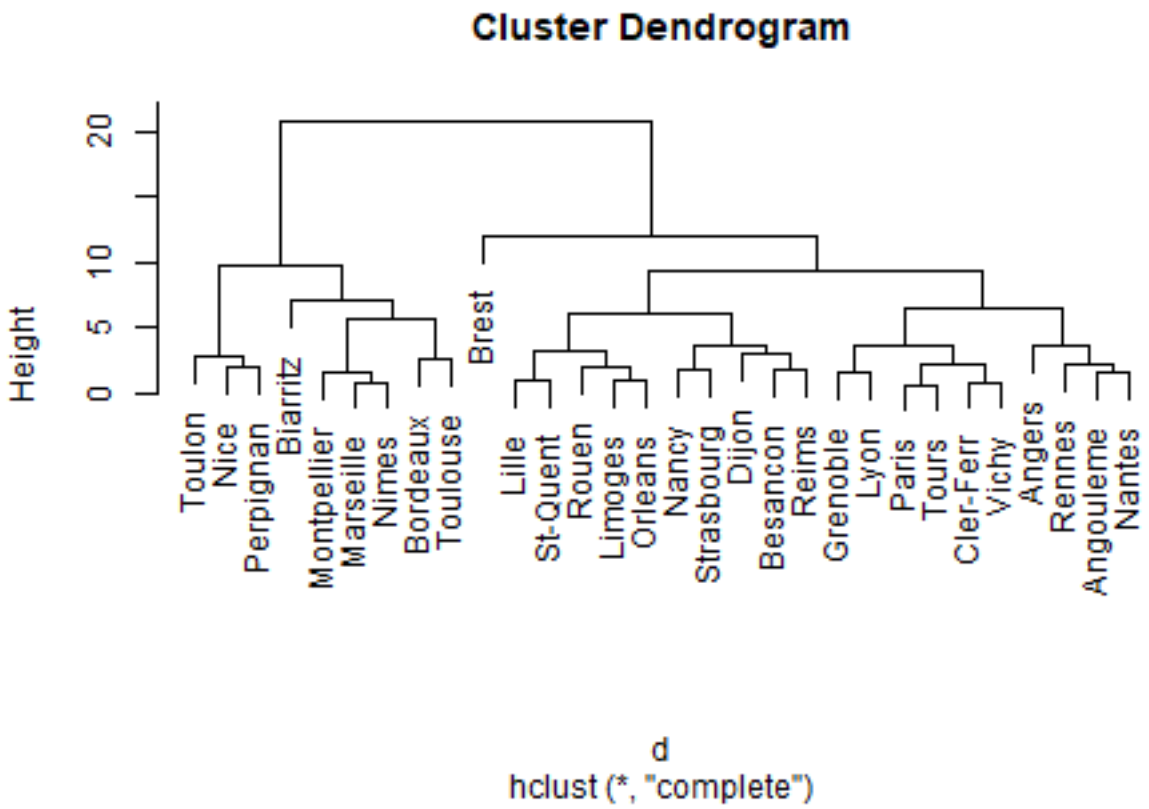
```
d=dist(X)
cah=hclust(d,method='ward.D') #utilisation du saut de ward
plot(cah)
```



```
memb <- cutree(cah, k = 3)  #coupe de dendograme à 3 classes
```

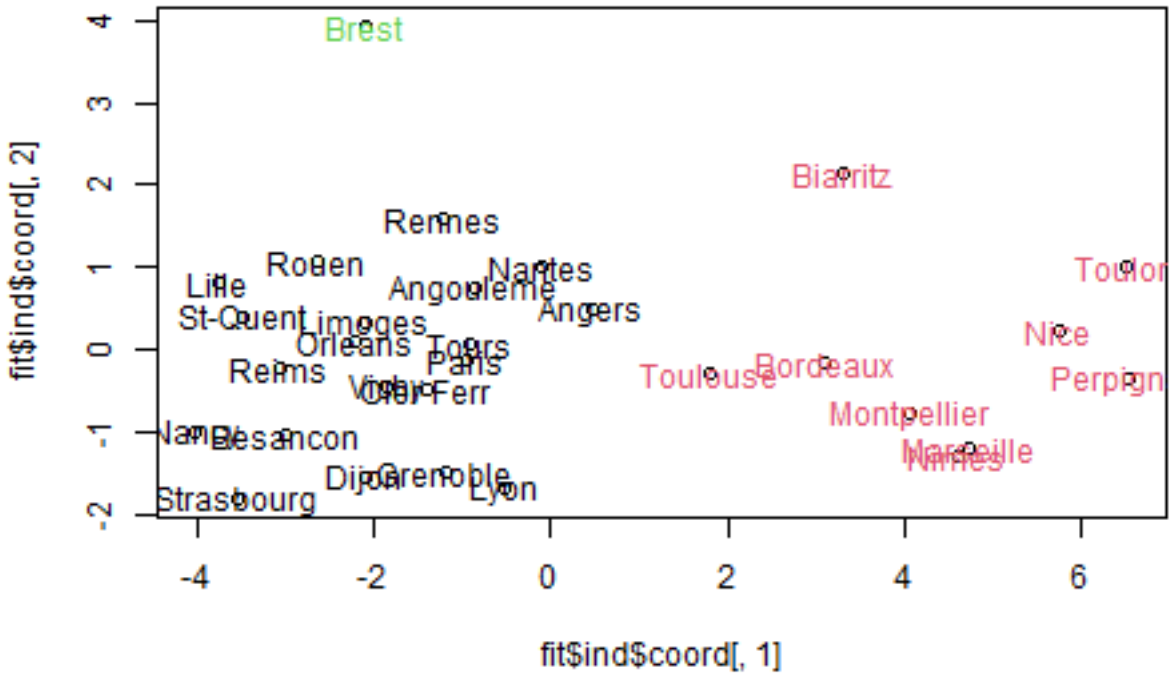
- Le choix de la distance d et du saut d^* sur les ensembles est important. Les différents sauts peuvent donner des résulats sensiblement différents. Cependant, si des classes sont bien distinctes dans E, on doit les retrouver approximativement pour plusieurs sauts. Généralement, la méthode "min" permet de bien reconstituer les classes filiformes, mais elles peuvent contenir des éléments très éloignés les uns des autres, ce qui peut être considéré comme un défaut majeur. La méthode "max" au contraire, en évitant de constituer des classes contenant des éléments éloignés, donne des classes plus "compactes".

```
cah=hclust(d,method='complete') #avec une autre mesure de dissimilarité
plot(cah)  #on obtient un dendrogramme différent (cf Brest par exemple)
```



- Il est parfois préférable de travailler sur les données centrées réduites, par exemple si E représente un ensemble d'individus sur lesquels on mesure des variables hétérogènes.
- Une fois la classification effectuée, on utilise généralement une ACP pour représenter les classes obtenues dans un sous-espace de dimension 2 et de se faire une idée de la pertinence de la classification obtenue.

```
memb <- cutree(cah, k = 3)  #coupe l'arbre à 3 classes
fit=PCA(X,graph=FALSE) #ACP
plot(fit$ind$coord[,1],fit$ind$coord[,2])
#individu dans le premier plan principal
text(fit$ind$coord[,1],fit$ind$coord[,2],row.names(X),col=memb)
```



La fonction HCPC du package FatoMineR réalise la CAH et représente les classes dans le premier plan priniapal. Elle permet de choisir le nombre de classes en cliquant avec la souris. En exercice, exécuter les commandes ci-dessous dans la console de R et liser l'aide de la fonction HCPC pour en comprendre précisemment son fonctionnement (vous pouvez aussi regarder les informations disponibles sur le web).

```
#res=PCA(X)  #à exécuter directement dans la console R
#HCPC(res)
```

7.4 Méthode des centres mobiles (k-means)

La CAH conduit rapidement à des temps de calculs importants lorsque le nombre d'individus augmente. La méthode des moyennes mobiles (k-means) peut être utilisée sur des jeux de données plus volumineux. Alors que la CAH ne fixe pas le nombre de classes a priori, ici on suppose que le nombre de classes k est connu.

Algorithme des centres mobiles :

1. *Initialisation :* on choisit aléatoirement k points dans $\mathbb{R}^p$, appelés centres des classes.
2. *Itération :* on itère les deux étapes suivantes jusqu'à ce que le critère à minimiser (inertie intraclasse) ne décroisse plus (*ie* on a un minimum local), ou bien jusqu'à atteindre un nombre d'itérations fixées :
   - tous les individus sont affectés à la classe dont le centre est le plus proche au sens de la distance choisie. On construit ainsi k classes d'individus,
   - on calcule les centres de gravité des classes créées qui deviennent les k nouveaux centres.

Remarques 3 • *L'algorithme fait diminuer l'inertie intraclasse à chaque itération. En effet, notons* $\{C_1^j, ..., C_k^j\}$ *la partition et* μ_i^j *les centres des classes à l'itération* j *et*

$$J(C_1^j, ..., C_k^j, \mu_1^j, ..., \mu_k^j) = \frac{1}{n} \sum_{l=1}^{k} \sum_{e \in C_l^j} d(e, \mu_l^j)^2 = \frac{1}{n} \sum_{l=1}^{k} n_l I_{\mu_l^j}(C_l^j).$$

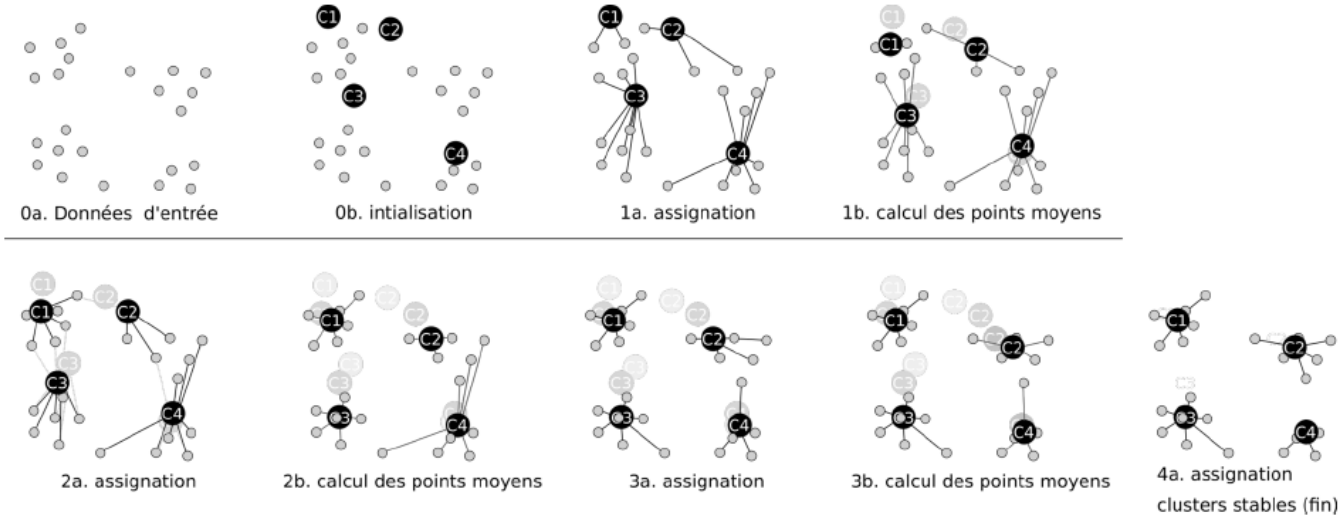


Figure 8: Algorithme des centres mobiles

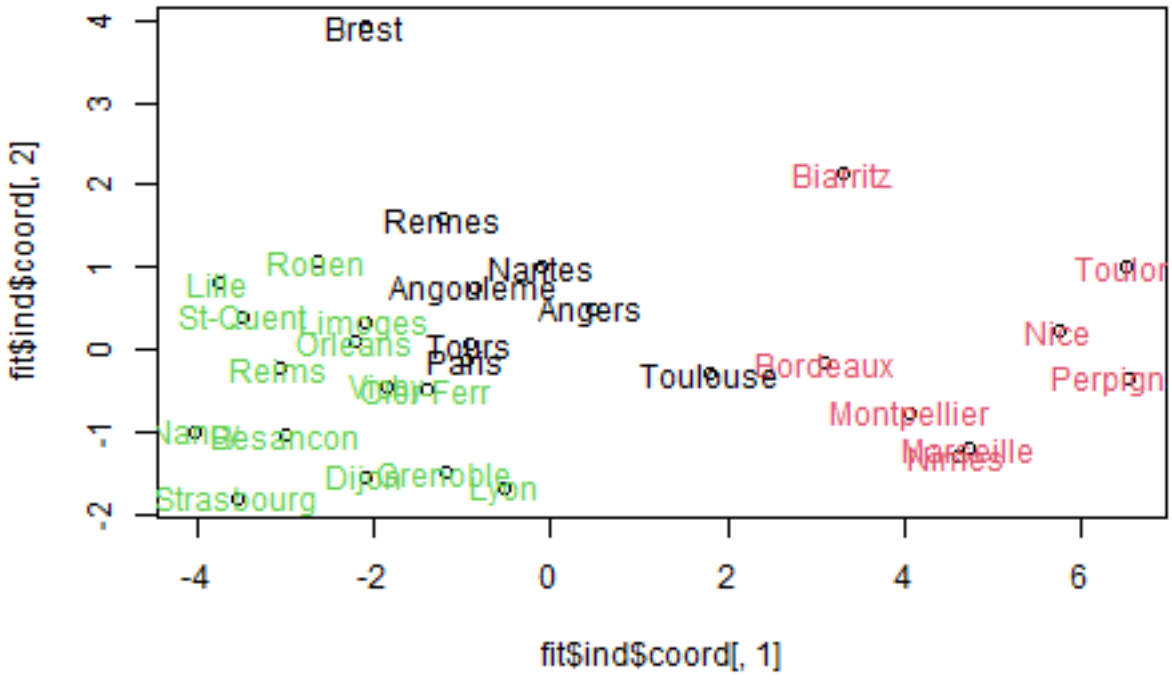
On vérifie (exercice) que les deux étapes de l'algorithme font décroître cette fonction et que, après mise à jour de la position des centres, $\mu_i^j = G_i^j$ est le centre de gravité de la classe C_i^j et donc

$$J(C_1^j, ..., C_k^j, G_i^j, ..., G_i^j) = I_{intra}(\{C_1^j, ..., C_k^j\}).$$

On en déduit que $I_{intra}(\{C_1^j, ..., C_k^j\}) \leq I_{intra}(\{C_1^{j-1}, ..., C_k^{j-1}\})$.

- *Le nombre de partitions possibles est fini, donc l'algorithme converge vers un minimum (local).*
- *L'initialisation est importante pour éviter les minima locaux pas intéressants, cf paragraphe suivant.*

```
k=3  #nombre de classes
cl=kmeans(X,k)  #3 classes, initialisation aléatoire
plot(fit$ind$coord[,1],fit$ind$coord[,2])
text(fit$ind$coord[,1],fit$ind$coord[,2],row.names(X),col=cl$cluster)
```




```
#répéter plusieurs fois les codes : les classes changent selon l'initialisation
```

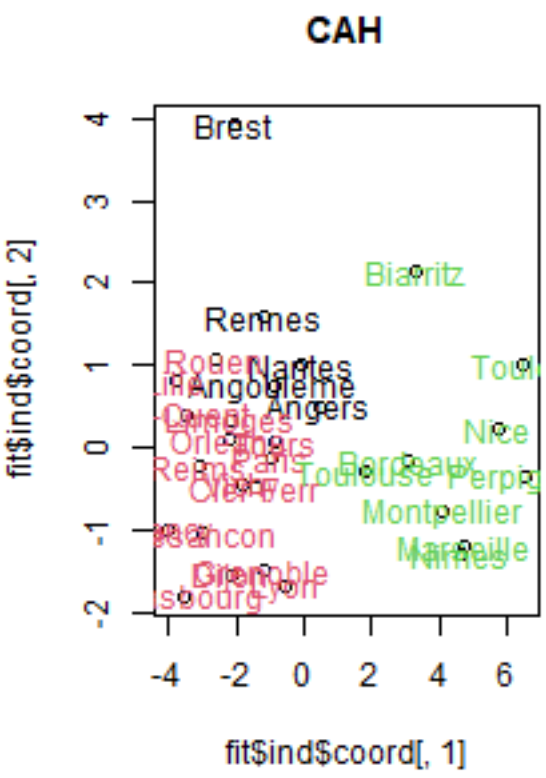
7.5 Combinaison CAH et k-means

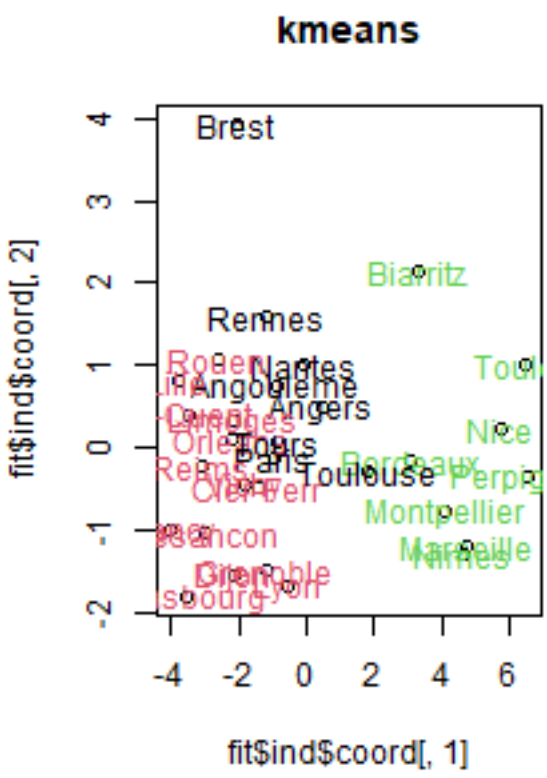
Il est courant de combiner les deux méthodes de classifications vues précédement. En effet, la partition obtenue par CAH n'est pas optimale et ne peut être utilisée que si le nombre d'observations est relativement faible. Par contre, la CAH aide à déterminer le nombre k de classes à utiliser dans la méthode des k-means et permettre d'initialiser le choix des centres de classe.

Algorithme de consolidation :

1. La partition obtenue par CAH est utilisée comme initialisation de l'algorithme k-means, éventuellement après sous-échantillonage si le jeu de données a beaucoup d'individus.
2. On itère plusieurs étapes de k-means.

```
cah=hclust(dist(X),method="ward.D")  #CAH
memb <- cutree(cah, k = k)
centers=matrix(0,nrow=k,ncol=ncol(X))
for (l in 1:k){ #centre de graivté des classes
  centers[l,]=apply(X[memb==l,],2,mean)
}

cl=kmeans(X,centers) #initialisation kmeans avec CAH
#comparaison CAH et CAH+k-means
par(mfrow=c(1,2))
plot(fit$ind$coord[,1],fit$ind$coord[,2],main='CAH')
text(fit$ind$coord[,1],fit$ind$coord[,2],row.names(X),col=memb)
plot(fit$ind$coord[,1],fit$ind$coord[,2],main='kmeans')
text(fit$ind$coord[,1],fit$ind$coord[,2],row.names(X),col=cl$cluster)
```


CAH
Brest
Biarritz
Rennes
Nantes
Angoulême
Angers
Nice
Montpellier
Marseille
Nîmes
fitindcoord[, 2]
fitindcoord[, 1]
-4
-2
0
2
4
6
-2
-1
0
1
2
3
4


kmeans
Brest
Biarritz
Rennes
Nantes
Angoulême
Angers
Tours
Toulouse
Nice
Montpellier
Marseille
Nîmes
fitindcoord[, 2]
fitindcoord[, 1]
-4
-2
0
2
4
6
-2
-1
0
1
2
3
4