



Algèbre Linéaire et Analyse de Données



Licence 2 MIASHS (2021-2022)

Guillaume Metzler

Institut de Communication (ICOM)
Université de Lyon, Université Lumière Lyon 2
Laboratoire ERIC UR 3083, Lyon, France


guillaume.metzler@univ-lyon2.fr

Résumé

Cette deuxième séance a pour objectif de vous travailler sur l'Analyse en Composantes Principales (ACP). Dans un premier temps, nous allons, sur un exemple simple, effectuer manuellement les différentes transformations à effectuer sur les données pour mettre en oeuvre cette méthode. Nous regarderons ensuite comment effectuer cela rapidement sur  en utilisant les fonctions appropriées. On pourra ainsi comparer les résultats obtenus manuellement avec ceux obtenus avec . Enfin, on se propose également d'interpréter les résultats de cette ACP sur le jeu de données étudiée, l'objectif étant de synthétiser l'information et d'en avoir une représentation visuelle "simple" et facile à interpréter.

1 Analyse en Composantes Principales

1.1 Introduction

Pour mettre en oeuvre l'ACP sur  ainsi que l'interprétation de ces résultats, nous allons travailler sur un jeu de données qui donne des informations sur *le passage des jeunes du système éducatif au travail*, i.e. notre étude portera sur des données Sociologiques.

Le cas qui nous intéresse aujourd'hui est extrait de la référence suivante : D. Busca et S. Toutain, *Analyse factorielle simple en sociologie. Méthodes d'interprétation et étude de cas*, 2009, De Coeck.

Les données consistent en l'étude de 5 critères sur un de 9 pays.

Les 9 pays étudiés sont l'Autriche, la Belgique, l'Espagne, la Finlande, la France, la Grèce, la Hongrie, l'Italie et la Suède.

Enfin les 5 critères/variables étudiées sont les suivants :

- **V1_1** : l'âge moyen lors de la sortie du système éducatif de la population active (15-64 ans)
- **V1_2** : l'âge moyen d'obtention d'un niveau de formation primaire ou secondaire (collège) lors de la sortie du système éducatif
- **V1_4** : l'âge moyen d'obtention d'un niveau de formation premier ou deuxième cycle de l'enseignement supérieur (licence ou master) lors de la sortie du système éducatif
- **V3_1** : pourcentage de parents ayant terminé un niveau de formation primaire ou secondaire¹
- **V3_3** : pourcentage de parents ayant terminé un niveau de formation premier ou deuxième cycle de l'enseignement supérieur.

Nous serons, par la suite, amenés à renommer ces variables comme suit : *age_moy*, *age_moy_ps*, *age_moy_sup*, *pc_par_ps*, *pc_par_sup* afin d'identifier clairement les variables étudiées mais aussi pour nous aider à interpréter les résultats dans la suite.

Les données se trouvent ci-dessous, vous avez simplement à copier-coller le code

```
#### Création du jeu de données ####  
  
# Variables  
  
V1_1=c(19.9,20.6,19.1,21.6,20.8,19.4,18.3,18.4,23.9)
```

1. Sous-entendu, sans avoir terminé un niveau de formation supérieur (universitaire). Il s'agit donc de parents ne disposant de "hauts" diplômes.

```

V1_2=c(18,18.3,15.2,16.9,17.9,14.5,14.9,14.6,22.6)
V1_4=c(24.7,22.8,22.5,25.3,23.2,23.3,23.1,25.0,26.4)
V3_1=c(27,45,80,21,51,66,26,68,26)
V3_3=c(19,26,10,36,15,9,13,6,36)

# Nom des individus

liste_obs=c("Autriche","Belgique","Espagne","Finlande",
            "France","Grèce","Hongrie","Italie","Suède")
liste_var=c("age_moy","age_moy_ps","age_moy_sup",
            "pc_par_ps","pc_par_sup")

# Enregistrement des données dans une base

# save(V1_1,V1_2,V1_4,V3_1,V3_3,liste_obs,liste_var,file="europe.RData")

```

1.2 Préparation des données

Cette première partie se concentre sur la préparation et la transformation des données. Cette étape là est essentielle lorsque l'on souhaite réaliser notre ACP manuellement.

1. Stockez votre jeu de données, *i.e.* les 5 vecteurs qui contiennent les informations relatives aux 5 variables dans une variable que l'on notera D .

```

D = cbind(V1_1,V1_2,V1_4,V3_1,V3_3)
colnames(D) = liste_var

```

2. Entrer et exécuter les commandes suivantes :

```

mean(D[1,])
## [1] 21.72

apply(D,1,mean)
## [1] 21.72 26.54 29.36 24.16 25.58 26.44 19.06 26.40 26.98

mean(D[,1])
## [1] 20.22222

m=apply(D,2,mean)

```

Que représentent ces différentes quantités et notamment m ?

m est le vecteur du barycentre du nuage de points.

3. Déterminer à partir de D le nombre d'individus et le nombre de variables à l'aide des commandes `nrow` et `ncol`. Ces informations là seront stockées dans les variables n et p .

```
# nombre de lignes
n = nrow(D)
# nombre de colonnes
p = ncol(D)
```

4. Entrer et exécuter les commandes suivantes :

```
# vecteur des écart-types des variables
apply(D,2,sd)

##      age_moy  age_moy_ps age_moy_sup  pc_par_ps  pc_par_sup
##      1.766195   2.610768   1.349074   21.938044   11.340684

# estimation non biaisée
s = apply(D,2,sd)
# estimation biaisée
s = s*sqrt((n-1)/n)
```

Que représente s ?

s est un vecteur contenant l'écart-type **biaisé** des différentes variables.

Remarque : la fonction *sd* (standard deviation) est l'estimateur sans biais de l'écart-type d'une variable. Dans le cas l'ACP, on utilise plutôt l'estimateur biaisé. C'est la raison pour laquelle nous effectuons l'opération $\sigma\sqrt{\frac{n-1}{n}}$.

5. Nous allons à présent centrer, réduire et diviser par \sqrt{n} le terme général de la matrice D et nous allons stocker la nouvelle matrice X dans une variable X . Ce que l'on peut faire avec la commande suivante

```
X = scale(D, center = m, scale = s)/sqrt(n)
```

Vérifier que le barycentre des individus de X est le vecteur nul. Vérifier également la norme des vecteurs colonnes de X vaut 1. Pour cela, utiliser la commande *apply*.

```
# Barycentre des individus dans X
apply(X,2,mean)

##      age_moy      age_moy_ps      age_moy_sup      pc_par_ps      pc_par_sup
## 1.572756e-16 2.197316e-16 -4.009139e-16 -2.023242e-17 -2.164188e-17

# Norme des colonnes
apply(X^2,2,sum)

##      age_moy      age_moy_ps      age_moy_sup      pc_par_ps      pc_par_sup
##              1              1              1              1              1
```

6. Pour que la table de données soit riche en information, nous pouvons donner des noms aux lignes et colonnes d'une matrice. Entrer et exécuter les commandes suivantes :

```
rownames(X) = liste_obs
colnames(X) = liste_var
```

1.3 Analyse du nuage des individus

1. A partir de X , stocker dans la variable C , la matrice des corrélations des variables.

```
C=t(X)%*%X
```

2. Procéder à la décomposition en éléments propres de C et stocker le résultat de cette décomposition dans la variable $C.eigen$.

```
C.eigen=eigen(C)
```

3. Entrer et exécuter la commande suivante :

```
C.eigen$values
## [1] 3.66779589 0.60428535 0.51885166 0.17559116 0.03347594
sort(C.eigen$values)
## [1] 0.03347594 0.17559116 0.51885166 0.60428535 3.66779589
```

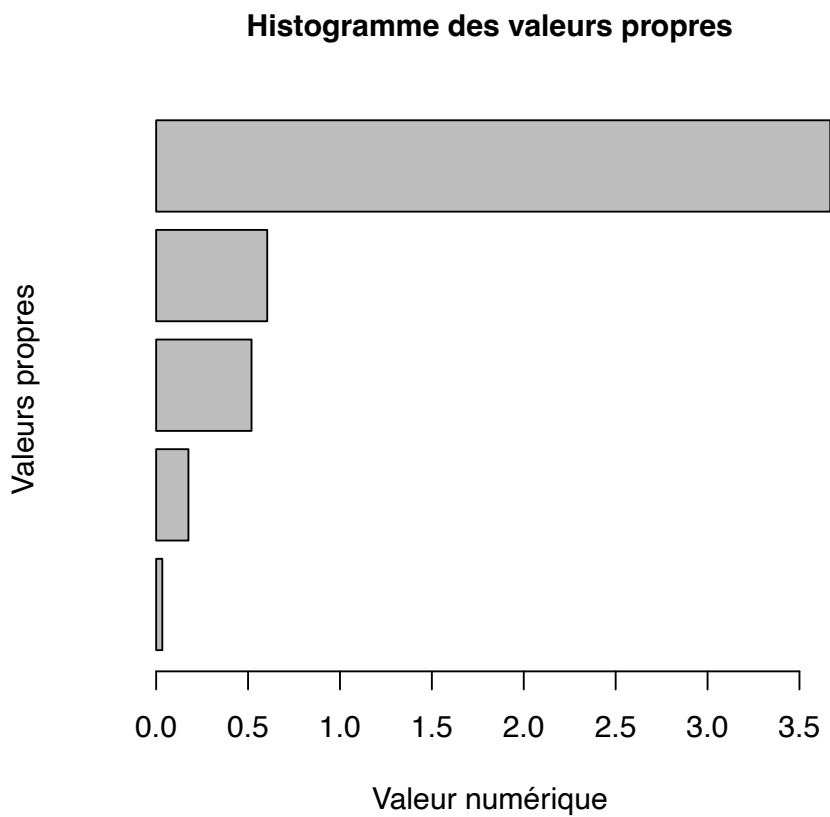
Que fait la commande *sort* ?

Cette commande va trier les valeurs propres par ordre croissant. Noter que vous pouvez également préciser que vous souhaitez trier les valeurs propres par ordre décroissant de la façon suivante :

```
sort(C.eigen$values, decreasing = TRUE)
## [1] 3.66779589 0.60428535 0.51885166 0.17559116 0.03347594
```

4. Entrer et exécuter la commande suivante :

```
# Histogramme des valeurs propres
barplot(sort(C.eigen$values),horiz=TRUE,
        main="Histogramme des valeurs propres",
        xlab="Valeur numérique", ylab="Valeurs propres ",
        cex.lab=1, cex.axis=1, cex.main=1)
```



Ceci est une commande graphique qui comporte plusieurs paramètres. À l'aide des noms de ces paramètres et en modifiant les valeurs de ces derniers, essayer de comprendre leur rôle. Vous pourrez également consulter l'aide pour vous aider.

5. Combien d'axes principaux proposez-vous de garder, au regard du graphique précédent.

Ici on se propose de garder les deux premiers axes principaux qui permettent de conserver plus de 90% de la variance initialement présente dans notre jeu de données.

On aurait presque pu être tenté de ne conserver que le premier axe principal mais cela aurait grandement limité le reste de l'étude.

6. Stocker dans deux variables $u1$ et $u2$, les deux premiers axes principaux.

```
#Axes principaux
u1=C.eigen$variables[,1]
u2=C.eigen$variables[,2]
```

7. Calculer les composantes principales associées aux deux premiers axes principaux. Il s'agit des coordonnées des individus sur ces deux axes. Vous stockerez ces deux vecteurs dans les variables $f1$ et $f2$.

```
#Composantes principales
f1=X%*%u1
f2=X%*%u2
```

8. Calculer la somme des valeurs propres de C que vous stockerez dans une variable $sum.eig$.

```
# Inertie totale
sum.eig=sum(C.eigen$values)
sum.eig

## [1] 5
```

A quoi correspond cette valeur ?

On rappelle que la somme des valeurs propres est égale à la somme des variances des différentes variables. Ici toutes les variables sont centrées et réduites, donc de variance 1, ce qui explique que cette somme soit égale à 5 soit le nombre de variables.

9. Calculer le pourcentage de l'inertie associée à l'axe $u1$. Il s'agit de la valeur propre associée à cet axe divisé par l'inertie totale. Faites de même pour l'axe principal $u2$. Le premier plan principal est l'espace engendré par $u1$ et $u2$. Le pourcentage de l'inertie expliquée par ce plan factoriel (ou plan principal) est la somme des inerties de chaque axe le constituant. Quel est le pourcentage d'information que contient ce premier plan factoriel ?

```

#Pourcentage de l'inertie sur le premier axe
u1.int=C.eigen$values[1]/sum.eig
u1.int

## [1] 0.7335592

# Pourcentage d'inertie sur le deuxième axe
u2.int=C.eigen$values[2]/sum.eig
u2.int

## [1] 0.1208571

# Pourcentage d'inertie sur le plan factoriel
u1.int+u2.int

## [1] 0.8544162

```

10. Stocker dans la variable F , la matrice F qui comporte dans ses deux colonnes, les coordonnées des individus sur les axes $u1$ et $u2$. Donner des noms aux lignes de F . Les noms des colonnes de F seront $u1$ et $u2$.

```

# Plan factoriel
F=cbind(f1,f2)
rownames(F)=liste_obs
colnames(F)=c("u1", "u2")

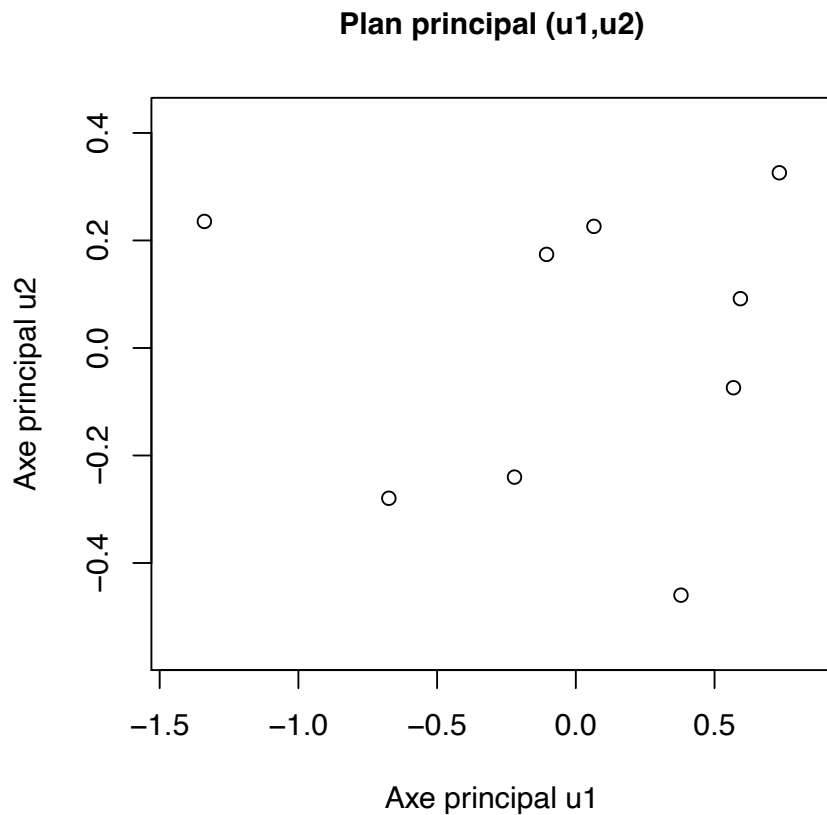
```

11. Entrer et exécuter la commande suivante :

```

#Représentation graphique
plot(F,xlab = "Axe principal u1",ylab = "Axe principal u2",
     main = "Plan principal (u1,u2)",
     xlim = c(min(f1)-0.1,max(f1)+0.1),
     ylim=c(min(f2)-0.1,max(f2)+0.1),
     cex.lab=1,cex.axis=1,cex.main =1)

```

Remarque : *plot* est une commande graphique utilisée pour représenter un nuage de points dans un repère orthonormé. Dans la commande précédente, les points sont des lignes de F qui sont des vecteurs dont les composantes sont données par les colonnes de F (qui sont donc les éléments de la base qui est ici de dimension 2).

Que font les paramètres *xlim* et *ylim* ?

Ils permettent de modifier les valeurs min et max des axes des abscisses et ordonnées.

12. Entrer et exécuter l'une après l'autre les commandes suivantes :

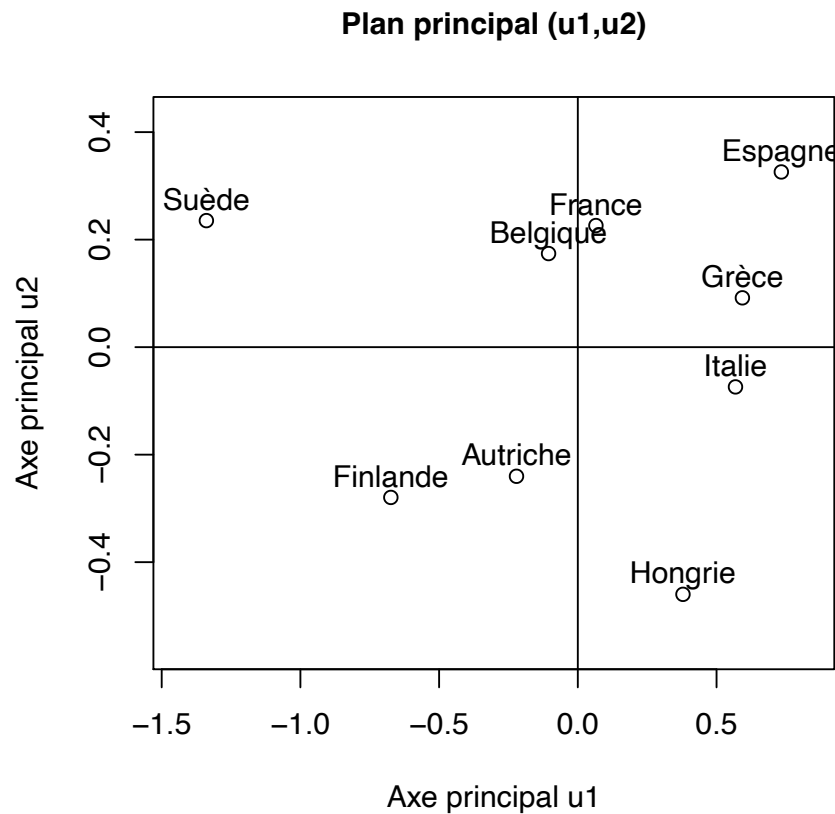
```
#Représentation graphique
plot(F,xlab = "Axe principal u1",ylab = "Axe principal u2",
     main = "Plan principal (u1,u2)",
     xlim = c(min(f1)-0.1,max(f1)+0.1),
```

```

ylim=c(min(f2)-0.1,max(f2)+0.1),
cex.lab=1,cex.axis=1,cex.main =1)

text(F,labels=rownames(F),pos=3,cex=1,offset=0.3)
abline(h=0)
abline(v=0)

```



Expliquer ce que font ces différentes commandes.

Les fonctions *abline* permettent de tracer une droite horizontale qui représentera l'axe des abscisses ($h=0$) ou une droite verticale qui représentera l'axe des ordonnées ($v=0$). Enfin, la fonction *text* permet d'attribuer les noms données aux lignes de notre F aux différents points qui forment notre nuage.