Majeure *Science des données* 2020\_21

Compte-rendu du TP n°1 - Analyse en composantes principales [[1]](#footnote-0)

**Noms du groupe**

***Wenxu ZHAO***

***Wenjing YE***

***Liwei XU***

***Yuteng WANG***

1. **Programmer l’ACP sur l’espace de variables :**

*Fournir les pseudo code et les scripts développés et répondre aux questions :*

1. - Visualiser la matrice de dimension (*n,p*) ( *n* individus et *p* variables)
2. - Construire les indicateurs statistiques classiques : variance, covariance, écart-type

*Code et commentaire mis au point sous Python ou R …*

data1<-read.table('data\_PDE20.txt',header=TRUE)

plot(data1)

par(R)

-------------------------------------------------------------------------------------------------------------------------------------------------------------

path = 'F:/EMSE/3A/BIG Data/tp3 analyse donne/tp1/data\_PDE20.txt'

data = pandas.read\_csv(path,sep = '\s+',header=0)

Data = np.array(data)

plt.plot(Data)

plt.show()

mean = []

var = []

std = []

colone = Data.shape[1]

for i in range(colone):

mean.append(np.mean(Data[:,i]))

var.append(np.var(Data[:,i]))

std.append(np.std(Data[:,i]))

mean.pop(0)

var.pop(0)

std.pop(0)

print("mean ：",mean)

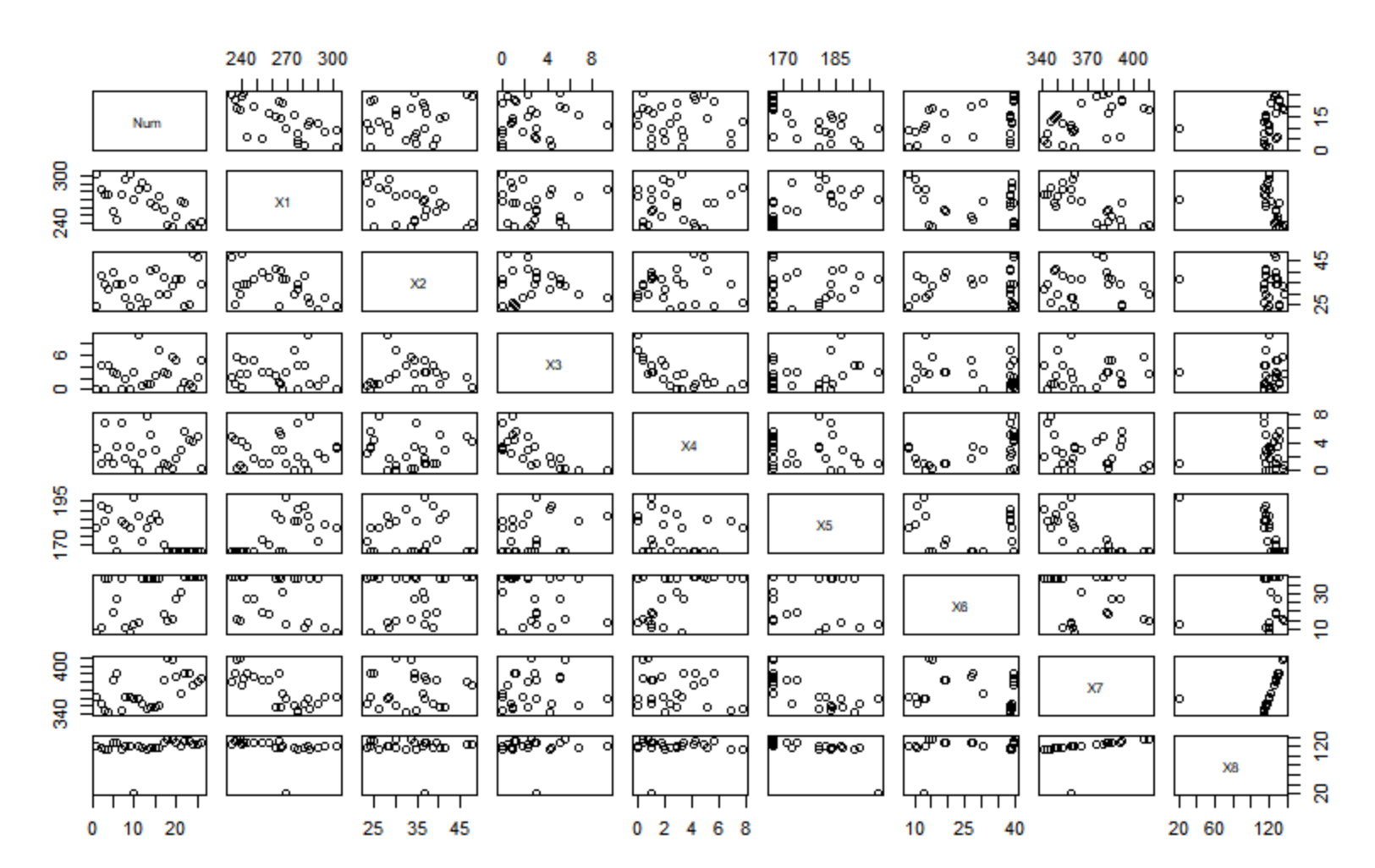
print("var ：",var)

print("std ；",std)

print(type(mean))

Par python

Vos résultats sur le cas d’étude :



mean ： [265.4473076923077, 33.39923076923077, 2.5503846153846155, 2.8888461538461536, 176.77657692307696, 27.892461538461546, 368.4529269230769, 118.95334615384617]

var ： [444.90679659763305, 45.46763017751479, 5.438603698224853, 4.855033284023668, 93.00877685946742, 154.63357609467454, 398.50493136427525, 437.483045995562]

std ； [21.0928138615414, 6.742968943834369, 2.3320814090045943, 2.203414006496207, 9.644105809221891, 12.435174952314686, 19.962588293211763, 20.916095381202535]

3. Le code qui permet de centrer le nuage dans l’espace initial

4. Le code qui permet d’obtenir le nouvel espace projeté : les hyperplans sont associés aux *p* directions de l’espace, chaque axe factoriel est de vecteur propre (respectivement matrice) et une valeur propre λ (respectivement matrice) :

**Cas ACP centrée** et cas de **l’ACP centrée- normée** (standardisée)

*Code et commentaire mis au point sous Python ou R …*

A = Data[:,1:8] #centrer le nuage : X -moyenne

for i in range(colone):

for j in range(row):

A[j, i] = A[j, i] - mean[i]

Par python

----------------------------------------------------------------------------------------------------------------------------------------------

#question 4 valeur propre et vecteur propre

d<-scale(data1, center = TRUE, scale = FALSE) #donnee centree

#d<-scale(data1) #donnee normal

MatCov<-cov(d, y = d, use = "everything", method = c("pearson", "kendall", "spearman"))

r<-eigen(MatCov)

VecteursPropres<-r$vectors

ValeursPropres<-as.vector(r$values)

Par R

5. Quelle est la valeur de l’inertie projetée totale ?

*Formule et résultat sur le cas exemple*

d<-scale(data1, center = TRUE, scale = FALSE) #donnee centree

#d<-scale(data1) #donnee normal

#Omi = x u

donnees <- d%\*%VecteursPropres

Q<-(rowSums(donnees[,1:8]^2))

plot(Q,main = "", xlab = "Individus")

IPT <- sum(Q)/26

print(IPT)

Par R

----------------------------------------------------------------------------------------------------------------------------------------------

*Comparaison entre ACP centrée et ACP normée*

*ACP centrée :*



*ACP normée :*



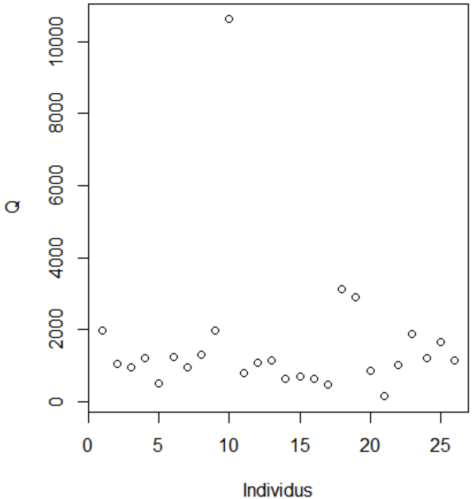


Figure *ACP centrée*

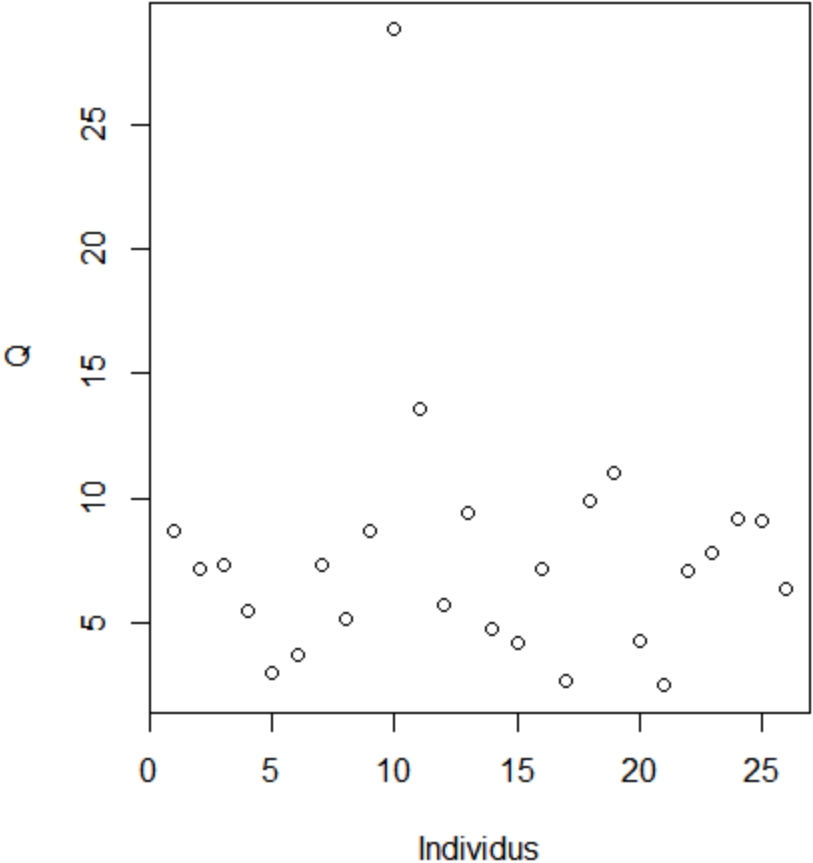


Figure *ACP normée*

La centralité est la moyenne des données individuelles dans l'ensemble de données moins la moyenne de l'ensemble de données.

La normalisation des données est l'écart-type des données après centralisation lorsqu'il est divisé par l'écart-type de l'ensemble de données, c'est-à-dire les données individuelles dans l'ensemble de données moins la valeur moyenne de l'ensemble de données divisée par l'écart-type de l'ensemble de données

Nous pouvons voir que la distribution des projections est plus concentrée au bas de l'image après le centralisation des données, et les projections sont normalisées après la normalisation des données.

6. Donner la cascade des valeurs propres ? combien y en a-t-il ?

*Formule et résultat sur le cas exemple*

***Plot 1 Poids de chaque valeur propre***

boxplot(donnees,main="Visualisation des donnee projete", xlab="Axes de projections")

TauxInertie=NULL

TauxInertie[1]<-ValeursPropres[1]/sum(ValeursPropres)

barplot(ValeursPropres/sum(ValeursPropres),main="Poids de chaque valeur propre dans le spectre",

xlab="Valeur Propre i",ylab="Valeur Propre i/Somme des valeurs propres")

**Plot 2 Taux d'inertie explique**

TauxInertie=NULL

TauxInertie[1]<-ValeursPropres[1]/sum(ValeursPropres)

for (i in 2:8){TauxInertie[i]<-TauxInertie[i-1]+ValeursPropres[i]/sum(ValeursPropres)}

plot(0:8,c(0,TauxInertie[1:8]),xlim=c(0,8),ylim=c(0,1),

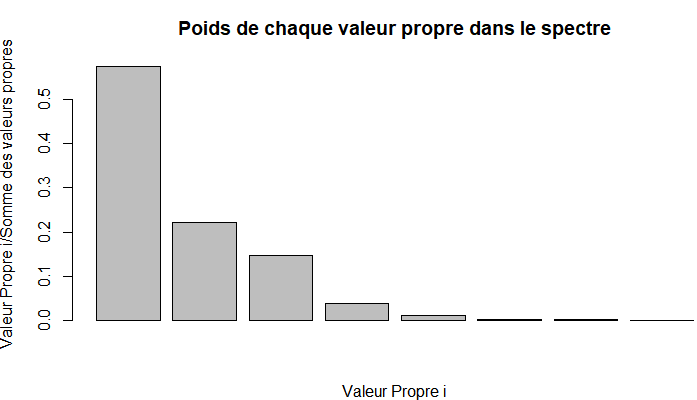
type="b",main="Taux d'inertie explique selon le nombre de valeurs propres retenues",

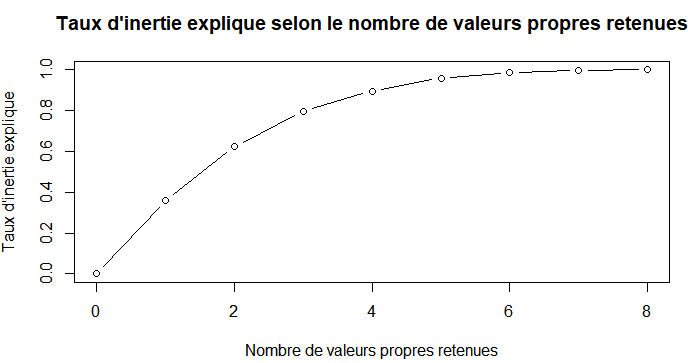
xlab="Nombre de valeurs propres retenues",

ylab="Taux d'inertie explique")

*Par R*

*----------------------------------------------------------------------------------------------------------------------------------------------*





*Combien retenez-vous de composantes dans l’exemple ?*

*Étant donné que seules les trois premières valeurs propres ont le plus grand poids, nous choisissons les trois premières*

*on dermine que le k=3(k<p)*

7. Comment obtenez-vous les nouvelles coordonnées des points dans l’espace réduit avec k < P

*Code de calcul*

*#on dermine que le k=3(k<p)*

*donnees2 <- d%\*%VecteursPropres[,1:3]*

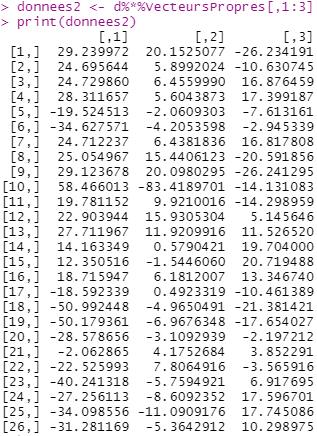


Figure les nouvelles coordonnées des points dans l’espace réduit

8. Donner les nouvelles coordonnées pour *k*=5 des 8 premiers individus puis leur qualité de projection

*Coordonnées de 8 premiers individus et de leur qualité de projection*

*Fournir le code*

*#on dermine que k=5*

*donnees3 <- d%\*%VecteursPropres[,1:5]*

*#donnees3 sont* les nouvelles coordonnées pour *k*=5

#leur qualité de projection

*Q<-(rowSums(donnees3[,1:5]^2)/rowSums(donnees[,1:8]^2))*

*print(Q)*

9. Fournir la contribution de ces 8 individus aux 5 premiers axes factoriels

*Calcul et résultats : sur ces 8 individus lesquels ont la plus forte contribution ?*

*# Fournir la contribution de ces 8 individus aux 5 premiers axes factoriels*

*Y=matrix(data = NA,nrow = 26,ncol = 8)*

*for (j in 1:5){#5 premiers axes factoriels*

*Y[,j]<-(1/26)\*(donnees3[,j]^2)/(ValeursPropres[j])*

*}*

*# 1 contribution*

*plot(1:26,Y[1:26,1],xlab="Individus",ylab="Contribution de l'individu sur l'axe principal",*

*main="Contribution des individus sur l'axe principal")*

*# 2 contribution*

*plot(1:26,Y[1:26,2],xlab="Individus",ylab="Contribution de l'individu sur l'axe principal",*

*main="Contribution des individus sur l'axe principal")*

*# 3 contribution*

*plot(1:26,Y[1:26,3],xlab="Individus",ylab="Contribution de l'individu sur l'axe principal",*

*main="Contribution des individus sur l'axe principal")*

*# 4 contribution*

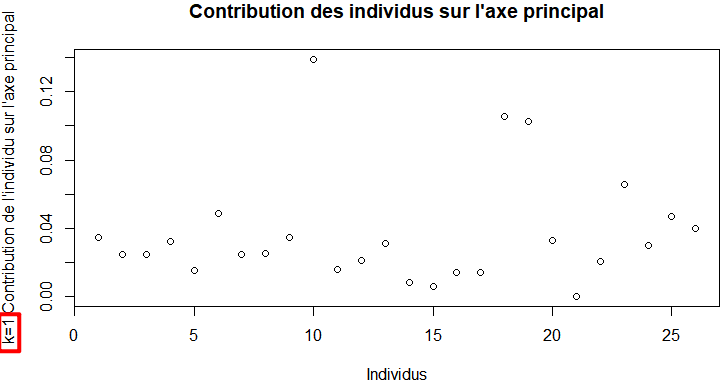
*plot(1:26,Y[1:26,4],xlab="Individus",ylab="Contribution de l'individu sur l'axe principal",*

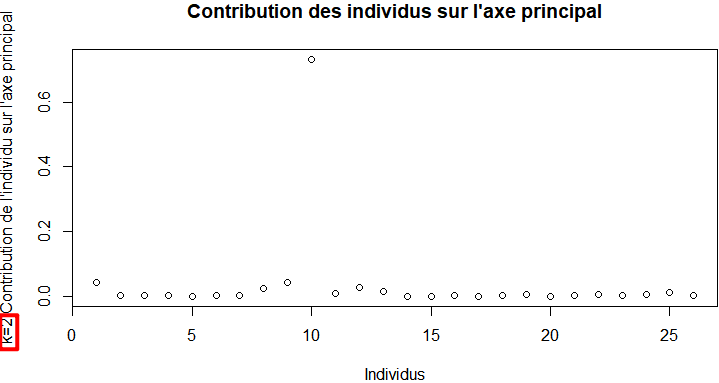
*main="Contribution des individus sur l'axe principal")*

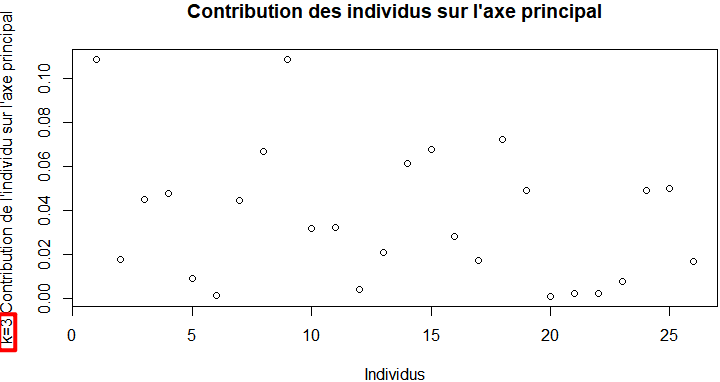
*# 5 contribution*

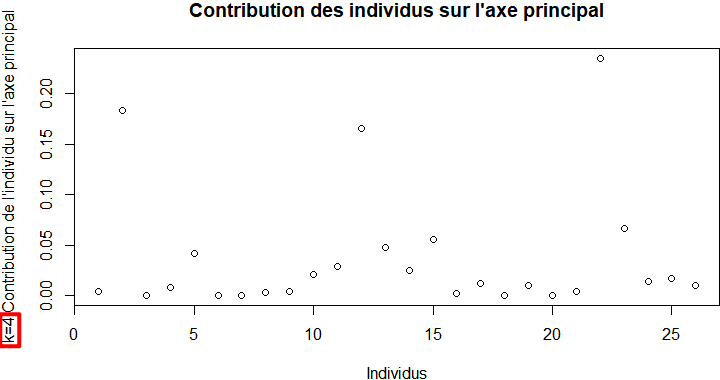
*plot(1:26,Y[1:26,5],xlab="Individus",ylab="Contribution de l'individu sur l'axe principal",*

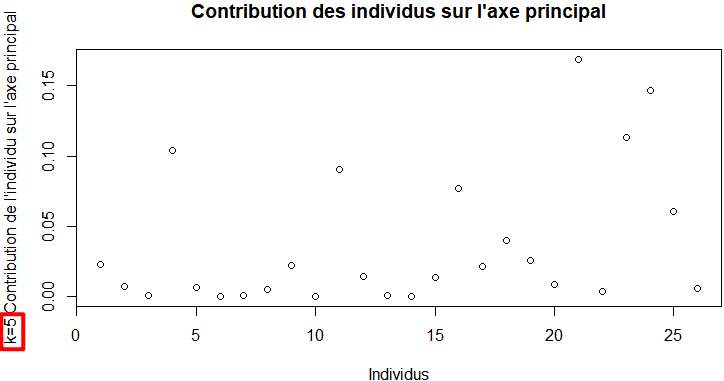
*main="Contribution des individus sur l'axe principal")*











10.Calculer la corrélation entre chacune des *p* variables avec les *p* composante :

*Code de calcul*

#Calculer la corrélation entre chacune des p variables avec les p composante

R=matrix(data = NA,nrow = 8,ncol = 8)

# dans le cas de normal

#for (i in 1:8){

# R[i,]<-sqrt(ValeursPropres[i])\*VecteursPropres[,i]}

# dans le cas de centree

for (i in 1:8){

R[i,]<-sqrt(ValeursPropres[i]/MatCov[i,i])\*VecteursPropres[,i]}

plot(R[1,],R[2,],xlab="Composantes c",ylab="Variables initiales j",

main="Correlation lineaire entre une composante c et une variable j")

*Pour les deux premières composantes : quelles sont les variables les plus corrélées à ces deux composantes : fournir vos résultats pour le cas ACP centrée et normée*

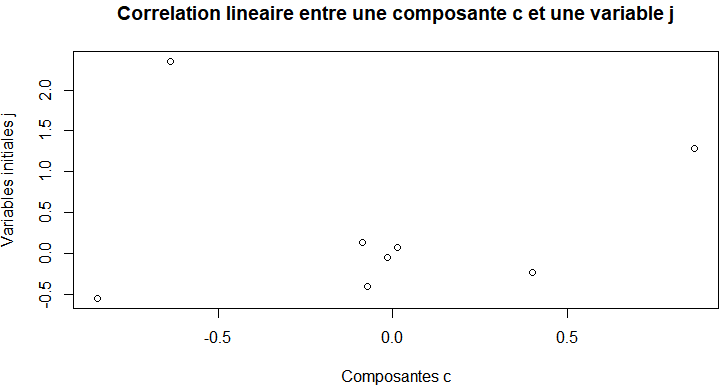


Figure *ACP centrée*

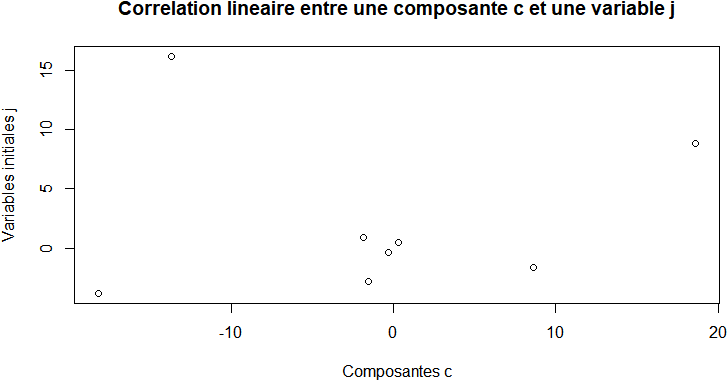


Figure *ACP normée*

1. Comparaison des ACP centrée et normée (standardisée)

La centralité est la moyenne des données individuelles dans l'ensemble de données moins la moyenne de l'ensemble de données.

La normalisation des données est l'écart-type des données après centralisation lorsqu'il est divisé par l'écart-type de l'ensemble de données, c'est-à-dire les données individuelles dans l'ensemble de données moins la valeur moyenne de l'ensemble de données divisée par l'écart-type de l'ensemble de données

Nous pouvons voir que la distribution des projections est plus concentrée au bas de l'image après le centralisation des données, et les projections sont normalisées après la normalisation des données.

1. *Dans les cadre : les réponses aux questions et éléments de compte-rendu* [↑](#footnote-ref-0)