Modèle de rendu

Compte-rendu du TP n°2 - Analyse en composantes principales partie 2

**Noms du groupe : AMINI Nada, ESSAYEGH Nour**

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

**Partie 1** - **l’ACP et étude de nuage de point dans** Rp

**Etude de la forme du nuage initiale et de sa répercussion sur la réduction de dimension**

* 1. Nuage isotrope

#On génère un nuage de données isotrope

import numpy as np

import pandas as pd

import matplotlib.pyplot as plt

import TP\_ACP as tp1

n=1000

#loi normale N(0,1)

X=np.random.randn(n)

Y=np.random.randn(n)

Z=np.random.randn(n)

X1=pd.DataFrame(data=X,columns=['X1'])

X2=pd.DataFrame(data=Y,columns=['X2'])

X3=pd.DataFrame(data=Z,columns=['X3'])

# le vecteur aléatoire (il est bien centrée réduit par construction on peut le vérifier en appliquant le moyenne et la variance des variables)

V=pd.concat([X1,X2,X3],axis=1)

#on peut le visualiser à l’aide de la commande

fig = plt.figure()

ax = plt.axes(projection='3d')

ax.scatter3D(X, Y,Z, 'Greens')

#la matrice de covariance

Mat\_cov=tp1.matrice\_covariance(V)

print(Mat\_cov)

#la cascade de valeurs propres

valeurs\_p,vect\_p=tp1.classeur(Mat\_cov)

print("\n les valeurs propres en ordre décroissant :\n"+ str(valeurs\_p))

Les valeurs propres en ordre décroissant :

[1.012357836979781, 0.9539452212144901, 0.8909531164387171]

#les valeurs propres sont toutes importantes mais on fait une projection sur les deux premiers hyperplans.

#la qualité

nv\_coord=tp1.coord\_Rk(V,2).loc[:50,:]

Q = [tp1.quali\_representation(V,nv\_coord,i) for i in range(nv\_coord.shape[0])]

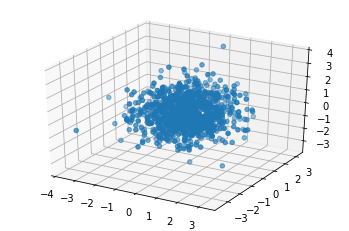
plt.hist(Q)

print("\n La qualité de la projection: \n"+str(Q))

Voici les résultats de l’étude d’un nuage isotrope :

#l’apeçu du nuage de points pour n=1000

On remarque que la distribution ne tend pas vers un des axes en particulier.



#la matrice de covariance :

Par construction du tableau V les variables X1, X2 et X3 sont indépendantes et on peut le visualiser sur la matrice de covariance ci-dessous :

Pour n=100

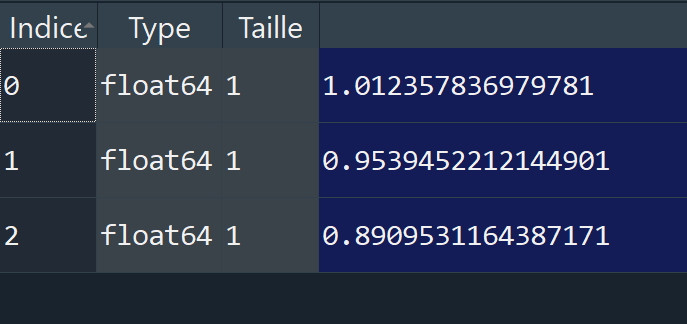


Pour n=10000



On remarque que pour un n plus grand la matrice de covariance tend vers la matrice identité. Ce résultat est assez intuitif.

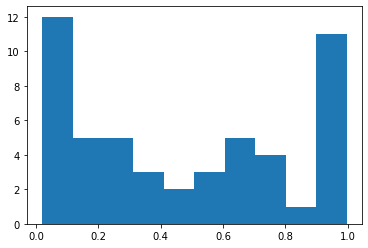
#les valeurs propres



On remarque ici que toutes les valeurs propres ont un poids important. Si on envisage de faire la projection sur les deux premières et qu’on visualise l’histogramme des qualités de projection on remarque qu’il y a beaucoup de projection entre 0 et 0.5 donc nous ne pouvons pas retenir une diminution de variables dans el cas de ce nuage.

Cela est prédictible vu la construction du nuage isotrope.

#la qualité



1.2 Nuage non isotrope

#On génère un nuage de données non isotrope avec une forte corrélation (relation de linéarité)

n=1000

X\_ni=X

Y\_ni=2\*X+1

Z\_ni=Z+X\_ni

X1\_ni=pd.DataFrame(data=X\_ni,columns=['X1'])

X2\_ni=pd.DataFrame(data=Y\_ni,columns=['X2'])

X3\_ni=pd.DataFrame(data=Z\_ni,columns=['X3'])

V\_ni=pd.concat([X1\_ni,X2\_ni,X3\_ni],axis=1)

#il faut normer ce tableau

V\_ni=tp1.acp\_normee(V\_ni)

#on visualise le tableau normalisé

ax = plt.axes(projection='3d')

ax.scatter3D(V\_ni['X1'], V\_ni['X2'],V\_ni['X3'], 'Greens')

plt.show()

#la matrice de covariance

Mat\_cov2=tp1.matrice\_covariance(V\_ni)

print(Mat\_cov2)

#la cascade de valeurs propres

valeurs\_p2,vect\_p2=tp1.classeur(Mat\_cov2)

print("\n les valeurs propres en ordre décroissant :\n"+ str(valeurs\_p2))

#la qualité de la projection

nv\_coord2=tp1.coord\_Rk(V\_ni,2).loc[:20,:]

Q2 = [tp1.quali\_representation(V\_ni,nv\_coord2,i) for i in range(nv\_coord2.shape[0])]

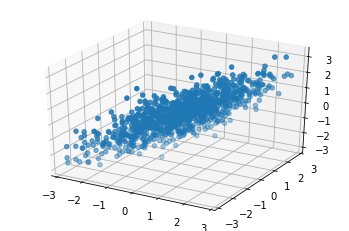
plt.hist(Q2)

plt.show()

Voici les résultats de l’étude d’un nuage non isotrope :

#l’apeçu du nuage de points après normalisation pour n=1000

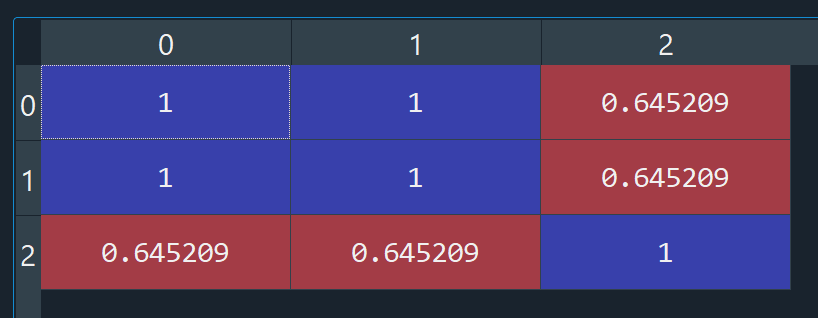
Ici on retrouve une forme elliptique allongée suivant un certain axe ce qui exprime une forte dépendance des variables Xi (notamment une relation de linéarité)



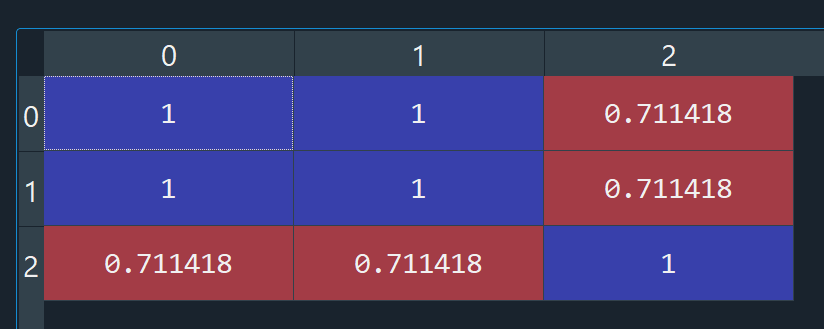
#la matrice de covariance :

Par construction du tableau V les variables X1 et X2 sont linéairement dépendante ce qui s’illustre dans la matrice de covariance par M\_cov[0,1]=1 il y a aussi une dépendance de X1 et X3 et cela revient à l’ajout de X1 à un échantillon de loi normale.

Pour n=100

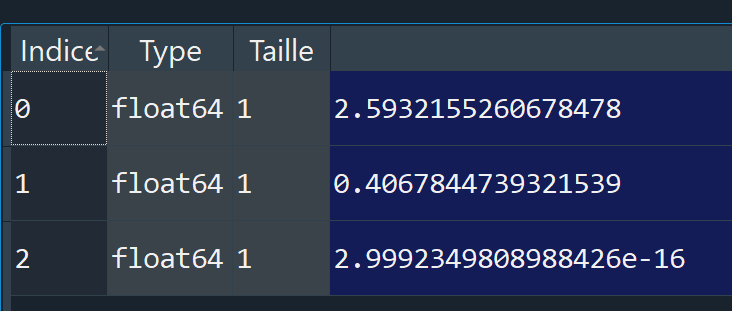


Pour n=10000



Les valeurs de la matrice de covariance s’affine de plus en plus n est important.

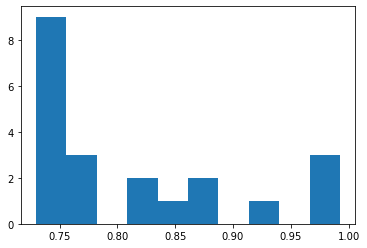
#les valeurs propres



On voit ici que la valeur propre lamda3 est très négligeable devant les deux autres donc on peut envisager une projection sur les deux hyperplans correspondant à lamda1 er lamda2.

#la qualité de projection

La qualité de la projection est très élevée puisque pour tous les individus est supérieur à 0.7. on le voit sur l’histogramme.



2) Cas du nuage proposé :*x <- sort(rnorm(1000)) y <- rnorm(1000) z <- rnorm(1000) + atan2(x, y)*

#On génère le nuage de données non isotrope proposé par l’ennoncé

n=1000

X\_ni\_ex=sorted(X)

Y\_ni\_ex=Y

Z\_ni\_ex=Z+np.arctan2(X\_ni\_ex,Y\_ni\_ex)

X1\_ni\_ex=pd.DataFrame(data=X\_ni\_ex,columns=['X1'])

X2\_ni\_ex=pd.DataFrame(data=Y\_ni\_ex,columns=['X2'])

X3\_ni\_ex=pd.DataFrame(data=Z\_ni\_ex,columns=['X3'])

V\_ex=pd.concat([X1\_ni\_ex,X2\_ni\_ex,X3\_ni\_ex],axis=1)

#la moyenne

print("la moyenne de ce tableau est :"+ str(np.round(np.mean(V\_ex,axis=0),4)))

La moyenne de ce tableau est :

X1 0.0

X2 -0.0

X3 -0.0

dtype: float64

#écart-type

print("l'écart type de ce tableau est:" +str(np.std(V\_ex,axis=0,ddof=0))

L’écart type de ce tableau est :

X1 1.006895

X2 1.063460

X3 2.111670

dtype: float64

#ACP centrée n’a pas d’intérêt puisque le tableau est déjà centré d’après le calcul de la moyenne des colonnes cependant il faut le normer puisqu’on remarque que la variable X3 n’est pas réduite.

V\_exn=tp1.acp\_normee(V\_ex)

#matrice de covariance non normé

Mat\_cov\_ex=tp1.matrice\_covariance(V\_ex)

print("La matrice de cov non normé"+str(Mat\_cov\_ex))

#matrice de covariance normé

Mat\_cov\_exn=tp1.matrice\_covariance(V\_exn)

print("La matrice de cov normé"+str(Mat\_cov\_exn))

#la cascade des valeurs propres

valeurs\_p3,vect\_p3=tp1.classeur(Mat\_cov\_ex)

print("\n les valeurs propres en ordre décroissant (non normée) :\n"+ str(valeurs\_p3))

valeurs\_p3n,vect\_p3n=tp1.classeur(Mat\_cov\_exn)

print("\n les valeurs propres en ordre décroissant (normée) :\n"+ str(valeurs\_p3n))

Les valeurs propres en ordre décroissant (non normée) :

[4.479571882439563, 0.978128326849662, 0.5331316544105277]

Les valeurs propres en ordre décroissant (normée) :

[1.6112889193526199, 0.9986053221538642, 0.3901057584935127]

#Ici on peut remarquer qu’entre les valeurs propres du tableau normé et non normé il peut y avoir une erreur d’interprétation dans la considération dans hyperplan pour la projection. En effet, avec la première liste de valeur propre on a envie de prendre une projection sur un seul plan et c’est celui qui correspond à lamda=4.48

Cependant après la normalisation on voit que les deux premières valeurs propres sont importantes et on aurait eu une perte d’information si l’on avait considéré que la première.

Conclusion :

Il est essentiel d’utilisé l’ACP normé pour une bonne interprétation des résultats.

1.3 Points extémaux

#on enlève quelques enregistrements (10)

#pour le nuage isotrope

V=V[:n-10]

valeurs\_p=cascade\_VP(V)

print("\n Les valeurs propres en ordre décroissant :\n"+ str(valeurs\_p))

# le valeurs propres ne changent pas

Les valeurs propres en ordre décroissant :

[1.045637135232715, 0.9922399785660027, 0.8883924763032149]

#pour le nuage non isotrope

V\_ex=V\_ex[:n-10]

valeurs\_p=cascade\_VP(V\_ex)

print("\n Les valeurs propres en ordre décroissant pour le nuage non normé:\n"+ str(valeurs\_p))

Les valeurs propres en ordre décroissant pour le nuage non normé :

[4.675993094517946, 0.993213662880317, 0.5250505672207688]

V\_exn=V\_exn[:n-10]

valeurs\_p=cascade\_VP(V\_exn)

print("\n Les valeurs propres en ordre décroissant pour le nuage non isotrope normé:\n"+ str(valeurs\_p))

Les valeurs propres en ordre décroissant pour le nuage non isotrope normé :

[1.559416537853434, 1.0009854796788364, 0.373194959402221]

**Partie 2. Etude de la forme du nuage initiale sur la réduction de dimension dans les deux espaces Rp ou Rn**

Le compte rendu devra restituer les différentes étapes, pas de modèle de rendu mais il doit contenir :

o L’ACP sur espace Rn sur les deux types de nuage isotrope et non isotrope

o La comparaison de l’ACP dans Rp sur les deux cas

o Les formules de passage à partir du cours et de vos résultats successifs

 Tester sur les deux cas : l’effet de la dimension sur les deux décompositions

o Proposer la projection de points supplémentaires (espace sur Rp ) et de points colonnes (variables) dans ce nouvel espace Rn . Pour cela vous devez proposer de générer soit des colonnes supplémentaires soit des lignes supplémentaires comme individus lignes :

o Cas 1 : un ou deux vecteurs gaussiens V indépendants de loi N(0,1) tel que V/||V|| (densité f(v) d’un vecteur gaussien est isotrope et dépend que de v) permettront ici de créer de nouvelles variables et/ou étendre l’échantillon pour placer ensuite ces nouveaux points dans l’espace réduit.

o Cas 2 : générer des points supplémentaires pour les 3 variables X,Y,Z et/ou générer une ou deux variables liées aux 3 précédentes . Attention ne pas refaire l’ACP sur ces données car elles sont dites non actives pour la réduction.