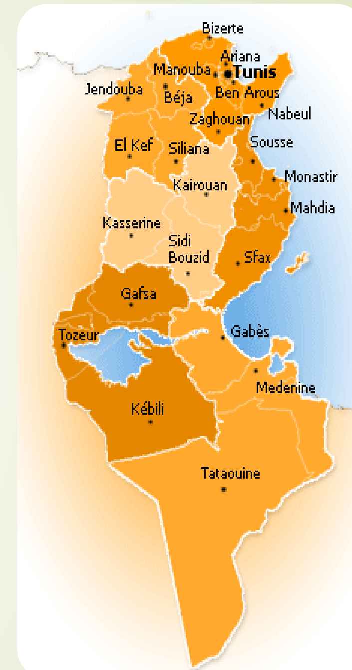
L'inégalité de développement régional en Tunisie



Numéro d'inscription: 21885

Enjeux sociétaux: environnement, sécurité, énergie.

plan

Introduction générale

Analyse en composantes principales

1/ Démonstration mathématique

2/ Algorithme

3/ Interprétation des résultats

Kmeans (‘k_moyennes mobiles’)

1/ Principe

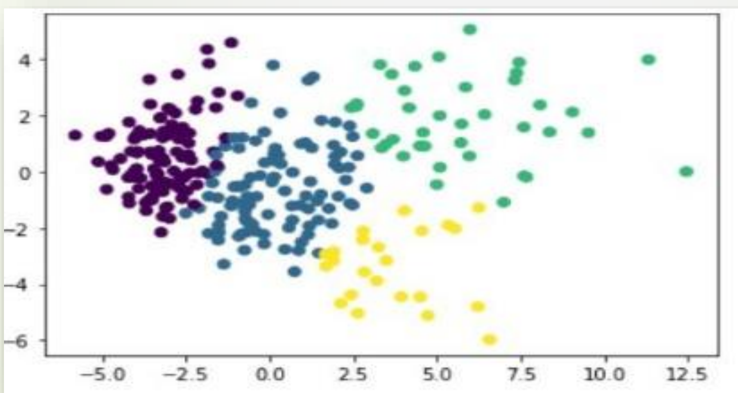
2/ Interprétation du résultat

Présentation

- Voulant comprendre l'origine de l'hétérogénéité entre les délégations Tunisiennes j'ai procédé comme suit:

Rechercher de données

Compréhension des données



Modélisation

Interprétation

4

Présentation

- J'ai pu collecter ces informations à partir du site de l'INS (institut national de la statistiques).
- Cette base représente les 264 délégations tunisiennes caractérisées par 20 variables économiques, démographiques et sociales obtenue lors du recensement de 2014.

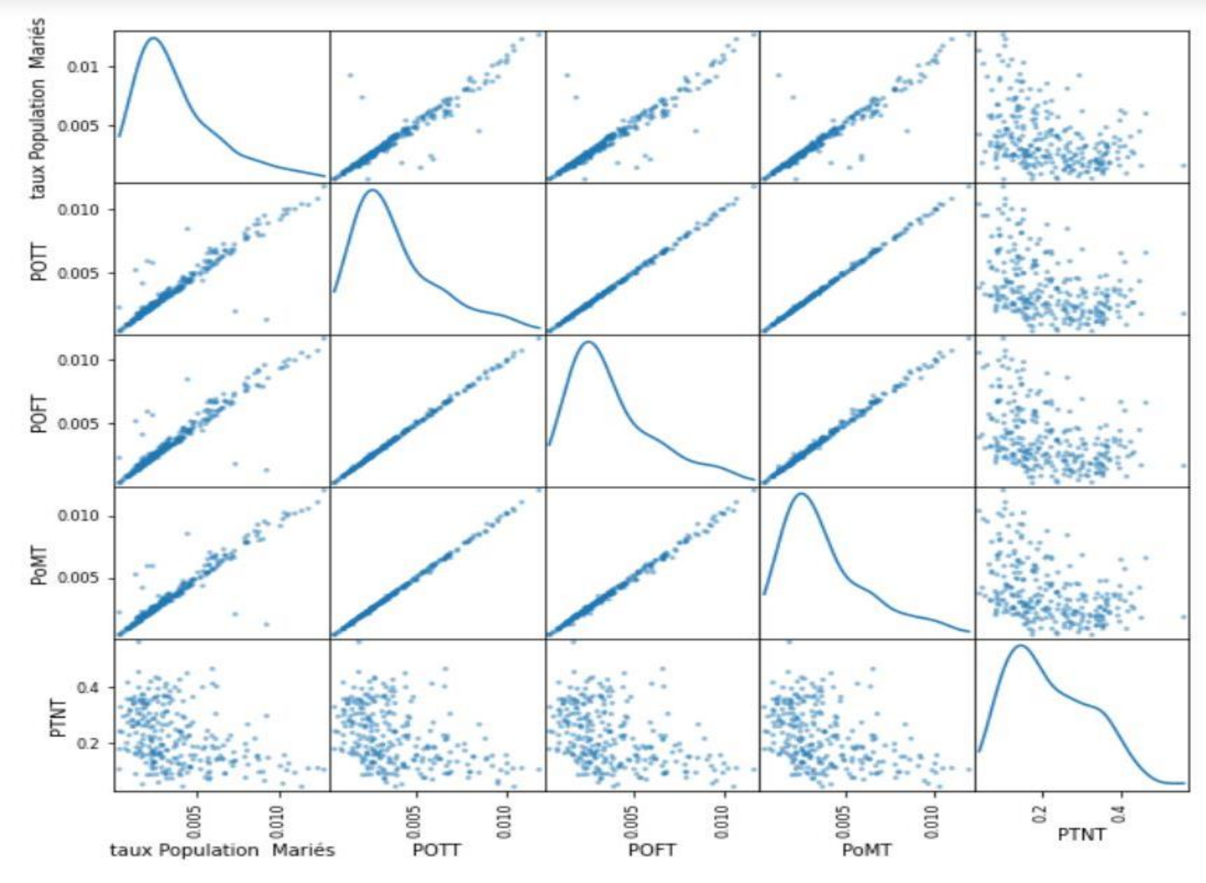
A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
NOM De la Délégations	taux Popul	POTT	POFT	PoMT	PTNT	PTST	PTUT	TAT10T	Cht	TCUTT	TCUMT	TCUFT	AOT	EIUTT	AOMT	EIUMT	AOFT	EETT	LRST	LROT
Agareb	0,004066	0,003728	0,003679	0,003778	0,239356	0,298511	0,066707	0,23822	0,004462	0,300337	0,165259	0,471037	0,003049	0,116031	0,00351	0,089316	0,001862	1161	0,824274	0,15684
Ain Drahem	0,003367	0,003223	0,003289	0,003157	0,362733	0,268779	0,061218	0,361543	0,004467	0,341499	0,207831	0,464088	0,002062	0,14641	0,002301	0,104907	0,001446	576	0,531671	0,244851
Akouda	0,003321	0,003141	0,003089	0,003193	0,106187	0,401152	0,18056	0,102679	0,001795	0,11976	0,067251	0,181135	0,003737	0,237902	0,003399	0,202703	0,004606	3125	0,944027	0,635827
Amdoun	0,002102	0,001929	0,001925	0,001934	0,393065	0,23135	0,04353	0,391746	0,001783	0,313149	0,193182	0,414013	0,001441	0,089455	0,00172	0,05561	0,000725	282	0,609703	0,285207
Ariana Ville	0,011304	0,010424	0,010476	0,010372	0,044395	0,377759	0,427816	0,044002	0,006391	0,06113	0,042103	0,084447	0,013803	0,648232	0,011352	0,611856	0,020106	15637	0,975517	0,985594
Bab Bhar	0,002701	0,003297	0,003082	0,003514	0,059563	0,487084	0,282015	0,059102	0,002892	0,088084	0,069946	0,113079	0,00469	0,440124	0,004359	0,385441	0,005541	5887	0,962597	0,988445
Bab Souika	0,002414	0,002657	0,002672	0,002643	0,129631	0,46507	0,136409	0,129559	0,003141	0,157998	0,109924	0,2085	0,00311	0,231467	0,002858	0,186081	0,003761	2589	0,957494	0,985068
Balta Bou Aouane	0,003935	0,00353	0,003573	0,003486	0,385274	0,266732	0,064147	0,385016	0,004941	0,39625	0,242386	0,545567	0,002189	0,150146	0,002505	0,110532	0,001375	1434	0,532207	0,193212
Bargou	0,001217	0,001137	0,001149	0,001124	0,339534	0,263623	0,064168	0,338438	0,001071	0,238384	0,154982	0,339286	0,000968	0,128253	0,001129	0,09403	0,000552	259	0,574342	0,397085
Béja Nord	0,007032	0,006483	0,006466	0,0065	0,226499	0,377998	0,095103	0,225341	0,008693	0,251549	0,159325	0,334032	0,006881	0,16225	0,006795	0,118931	0,007103	3383	0,820659	0,730919
Béja Sud	0,003743	0,003469	0,003462	0,003476	0,247225	0,36152	0,107684	0,24674	0,00423	0,243966	0,154255	0,329702	0,003593	0,188434	0,003638	0,139201	0,003479	2038	0,887135	0,607341
Bekalta	0,001664	0,001625	0,001591	0,00166	0,16441	0,376487	0,120435	0,163395	0,00076	0,20297	0,130898	0,288288	0,001869	0,169778	0,001917	0,137143	0,001746	673	0,956234	0,714734
Belkhir	0,001269	0,001346	0,001468	0,001224	0,363848	0,258362	0,05793	0,363516	0,001026	0,340344	0,1875	0,505976	0,001242	0,09653	0,001125	0,09704	0,001541	624	0,395912	0
Ben Arous	0,002934	0,002834	0,00288	0,002788	0,095754	0,46238	0,156309	0,095375	0,002841	0,159054	0,103756	0,218071	0,003158	0,263714	0,002874	0,219387	0,00389	2392	0,960002	0,983535
Ben Guerdane	0,006613	0,007276	0,007276	0,007277	0,191111	0,369683	0,09654	0,189916	0,008156	0,378677	0,1843	0,58567	0,006218	0,145938	0,007611	0,113504	0,002637	2130	0,85574	0
Benbla	0,002903	0,002964	0,002923	0,003006	0,145262	0,38463	0,093526	0,144294	0,0015	0,167598	0,10882	0,223555	0,003544	0,12714	0,003066	0,103079	0,004773	961	0,95955	0,947642
Beni Hassen	0,001299	0,001263	0,001244	0,001282	0,163449	0,386529	0,089088	0,163165	0,000897	0,276882	0,221289	0,328165	0,00143	0,120942	0,001224	0,102891	0,001958	460	0,916934	0,55539
Beni Khadech	0,002131	0,002357	0,002638	0,002074	0,278215	0,284307	0,077776	0,274847	0,002299	0,39303	0,203354	0,555755	0,001427	0,153977	0,001749	0,10648	0,000597	425	0,642902	0
Beni Khaled	0,003651	0,003457	0,003451	0,003463	0,146354	0,394066	0,095722	0,146078	0,002449	0,182209	0,109029	0,256722	0,004389	0,138393	0,003938	0,115117	0,005547	2183	0,901793	0,621289
Beni Khiar	0,004205	0,003927	0,003844	0,004011	0,110489	0,383219	0,119793	0,11009	0,002158	0,178808	0,092489	0,267949	0,004963	0,165607	0,004678	0,135381	0,005697	3041	0,891106	0,850425
Bir Lahmar	0,000715	0,00077	0,000838	0,000702	0,179548	0,366391	0,122317	0,179638	0,001296	0,406015	0,222973	0,552846	0,000591	0	0,000638	0	0,000473	525	0,909248	0
Bir Ali Ben khlifa	0,004769	0,004797	0,004849	0,004744	0,363695	0,245221	0,055699	0,362124	0,004873	0,371448	0,263529	0,517516	0,003419	0,103993	0,004241	0,079567	0,001305	840	0,273366	0,051475

Carte descriptive

*PoTT : Population Totale en 2014
*PoFT : Population Féminine en 2014
*PoMT :Population Masculine en 2014
*PTNT :Pourcentage Population Totale dans les deux milieux sans
niveau d'instruction dans population totale de plus de 10 ans dans les deux milieux en 2014
*PTST : Pourcentage Population Totale dans les deux milieux avec
niveau d'instruction Secondaire dans population totale de plus de 10 ans dans les deux milieux en 2014
*PTUT :Pourcentage Population Totale dans les deux milieux avec
niveau d'instruction Supérieur dans population totale de plus de 10 ans dans les deux milieux en 2014
*TAT10T: Taux analphabète total sexe dans les deux milieux pour la population de plus de 10 ans en 2014
*ChT :Total Chômeurs de plus de 15 ans dans l'ensemble des milieux en 2014
*TCUTT: Taux de Chômage des Diplômés du Superieur Total Sexe et Total Milieu en 2014
*TCUMT :Taux de Chômage des Diplômés du Superieur Masculin et Total Milieu en 2014
*TCUFT :Taux de Chômage des Diplômés du Superieur Féminin et Total Milieu en 2014
*AOT: Population Totale Active Occupée dans l'ensemble des milieux en 2014
*EIUTT :Pourcentage Emploi Total(Population Active Occupée de plus de 15 ans) avec
Niveau d'Instruction Superieurt dans l'ensemble des milieux
*EIUMT Pourcentage Emploi Masculin(Population Active Occupée de plus de 15 ans) avec
Niveau d'Instruction Superieurdans l'ensemble des milieux
*AOMT :Population Masculine Active dans l'ensemble des mileux en 2014
*AOFT: Population Féminine Active Occupée dans l'ensemble des milieux en 2014
*EETT: Entrants Délégation Total Sexe Total Milieu en 2014
*LRST: Part Logement Raccordé en Eau Potable SONEDE dans total Logements en % dans les deux milieux (T) en 2014
*LROT: Part Logement Raccordé aux réseaux Assainissement ONAS dans total Logements en % dans les deux milieux (T) en 2014

Présentation

Première lecture de la base



Les variables qui présentent une courbe presque linéaire avec une pente positive sont des variables corrélées positivement entre elles

(Code n°3 , annexe)

L’ACP: analyse en composantes principales

- À partir de cette base, on souhaite extraire des informations supplémentaires.
- On applique une ACP qui nous permet de réduire la dimension de représentation de la base avec une perte minimale en information.

L’ACP est une méthode de la famille de l’analyse de données et plus précisément de la statistique multi variée qui permet de réduire le nombre de variables et de rendre l'information moins redondante.

L'ACP: analyse en composantes principales

1/ Démonstration mathématique

Chaque délégation parmi les 264

- Un point(a_i) de l'espace vectoriel $\mathbb{R}^{20}$
- $(a_i)_r$ pour $r \in [1,20]$ sont les coordonnées du point (a_i)

Étant donnée qu'on ne peut pas visualiser 264 points dans un espace de dimension 20 → on cherche une représentation de ces points dans un espace de dimension plus faible.

L'ACP: analyse en composantes principales

En remplaçant les points (a_i) par des points (b_i) qui appartiennent à un même espace affine D on définit *l'erreur globale* :

$$\mathbf{E(b)=\sum_i \| b_i - a_i \|^2}$$

Étape 1

On prend la dimension de D égale à zéro:

- On cherche un unique point A qui minimise l'erreur $E(A)=\sum_{i=1}^{264} \| A - a_i \|^2$

$$E(A)=\sum_{i=1}^{264} \| a_i \|^2 + \sum_{i=1}^{264} \| a_i \|^2 - 2\langle a_i, A \rangle$$

$$\nabla E(A)=0$$

Le barycentre des points (a_i)

$$A=\frac{1}{264}\sum_i a_i$$

L'ACP: analyse en composantes principales

Étape 2

Étant donnée un espace affine D de dimension supérieure à 1

On commence par montrer que A$\in$ D:

- On pose $B=\frac{1}{264}\sum_i b_i$ → B$\in$ D
- On montre que B=A :

On pose $b'_i=b_i+A-B$ → $b'_i \in D$

$E'=\sum_i \| b'_i - a_i \|^2 = \sum_i \| b_i - a_i + A - B \|^2 = \sum_i (\| b_i - a_i \|^2 + \| A - B \|^2 + 2 \langle (b_i - a_i), (A-B) \rangle)$
$E'=E- 246\| A - B \|^2$

Pour que les points (b_i) réalisent le minimum et soient les points les plus proches de(a_i)appartenant à D il faut que → A=B

L'ACP: analyse en composantes principales

Étape 3 Centrer le nuage de points:

- $\forall\, i \in [1{,}246] \qquad a_i \longleftarrow a_i - A$

Code n°2, annexe

Étape 4

- Si dimension de D égale à 1 $\Longrightarrow$ D est une droite contenant A
- On note $\lambda = (\lambda_r)$ la direction de D avec $\|\lambda\| = 1$

$\forall\, b_i \in D$ il existe K_i $\begin{cases} \forall i \quad b_i = K_i\,\lambda \\ \forall i \quad K_i \text{ minimise } E = \sum_i \| b_i - a_i \|^2 = \sum_i \| K_i\,\lambda - a_i \|^2 \\ \qquad = \sum_i \| a_i \|^2 + \sum_i K_i^2 - 2 < a_i, \lambda > K_i \end{cases}$

$$\nabla E = 0 \Longrightarrow \forall i \quad \frac{\partial E}{\partial K_i} = 2K_i - 2 <\lambda, a_i> = 0 \Longrightarrow K_i = <\lambda, a_i>$$

L'ACP: analyse en composantes principales

- On cherche une direction λ qui minimise l'écart de représentation sachant que:

$b_i = K_i \lambda$ $K_i = \langle \lambda, a_i \rangle$ $\| \lambda \| = 1$

- $E(\lambda) = \sum_i \| a_i \|^2 - \sum_i \langle a_i, \lambda \rangle^2$
- Minimiser l'erreur E $\Longleftrightarrow$ maximiser $I(\lambda) = \sum_i (\langle a_i, \lambda \rangle)^2$ appelé inertie expliquée par λ

L'ACP: analyse en composantes principales

Théorème admis

Théorème : Soient $f, g_1 \dots, g_p$ des fonctions de classe $\mathcal{C}^1$ sur un ouvert U de $\mathbb{R}^n$, à valeurs dans $\mathbb{R}$ et X l'ensemble défini par :

$$X = \{x \in U;\ g_1(x) = \dots = g_p(x) = 0\}.$$

Si la restriction de f à X admet un extrémum local en a, et si les différentielles $dg_1(a), \dots, dg_p(a)$ sont des formes linéaires indépendantes, alors il existe des réels $c_1, \dots, c_p$ tels que :

$$df(a) = c_1 dg_1(a) + \dots + c_p dg_p(a).$$

www.bibmath.tn

Ces réels $c_1, \dots, c_p$ sont appelés **multiplicateurs de Lagrange**.

- On cherche donc à maximiser la fonction $M(\lambda)=\sum_i <a_i,\lambda>^2$ sous la contrainte $\|\lambda\| = 1$
- La contrainte : $g(\lambda)= \|\lambda\| -1 = 0$
- En considérant A la matrice associée à la base on a : $\sum_i <a_i,\lambda>^2 = \|A\lambda\|^2$
- $\|A\lambda\|^2 = <A\lambda, A\lambda> = <{}^tAA\lambda, \lambda>$ $\Rightarrow$ $\nabla M(\lambda)=2\,{}^tAA\lambda$ et $\nabla g(\lambda)= 2\lambda$
- Il existe f tel que $\nabla M(\lambda)= \nabla g(\lambda)$ $\Rightarrow$ ${}^tAA\lambda=f\lambda$ et $\|A\lambda\|^2= f$

λ est un vecteur propre de la matrice symétrique réelle M

f est la valeur propre associée

f=I(λ) l'inertie expliquée par cet axe

L'ACP: analyse en composantes principales

2/Algorithmes

- **1/ Recherche du premier axe** → Méthode des puissances

C'est un algorithme permettant de calculer (plutôt, d'estimer) la valeur propre de plus grand module et un vecteur propre associé d'une matrice A.

Théorème : Soit $A \in \mathcal{M}_n(\mathbb{R})$ une matrice, $\lambda_1, \ldots, \lambda_n$ ses valeurs propres, et on suppose que $|\lambda_1| > \max(|\lambda_2|, \ldots, |\lambda_n|)$. Soit aussi $\vec{v}_1$ un vecteur propre associé à λ_1. Alors, si x_0 est un vecteur non orthogonal à $\vec{v}_1$, et si (x_n) est définie par la relation de récurrence $x_{n+1} = Ax_n$, alors :

$$\begin{cases} \frac{x_n}{\|x_n\|} \text{ converge vers un multiple de } v_1 \\ \frac{\langle x_{n+1}, x_n \rangle}{\|x_n\|^2} \to \lambda_1. \end{cases}$$

Démonstration

- A matrice réelle symétrique

Théorème spectral

Il existe une base (v_1,….., Vn) de vecteurs propres associés aux valeurs propres (λ_1,……,λn).

La suite (Xn) de vecteurs de ℝn définit par:

- $x_{i+1}=Ax_i$
- x_0 le vecteur initial tel que $x_0 \notin$ vect (V_2….. V_i) pour tout i

Par récurrence on montre que : $X_i=A^iX_0$

On pose $X_0=\sum_{i=1}^{n} a_i V_i$

A $X_0=\sum_{i=1}^{n} a_i y_i \lambda^i = a_1\lambda^i(V_1+\sum_{i=2}^{n} a_i/a_1 * (\lambda_i/\lambda_1)^i V_i$

Ceci montre que :

1/ la suite (Xn) converge vers V_1. $\Longrightarrow$

2/ $< x_{i+1}, x_i >= < Ax_i, x_i >= x_i.T\, Ax_i$ $\quad$ $T.V_1AV$ $=\lambda_1 \|V_1\|$

(voir annexe code n°1)

L'ACP: analyse en composantes principales

- **2/Recherche des autres axes** → **Méthode des puissances itérées**
- On pose la matrice B=A- $\lambda_1 V_1 T V_1$
- Cette matrice admet (λ_2,……,λn,0)
- La méthode de la puissance appliquée sur B nous donne λ_2 et V_2

(voir annexe code n°2)

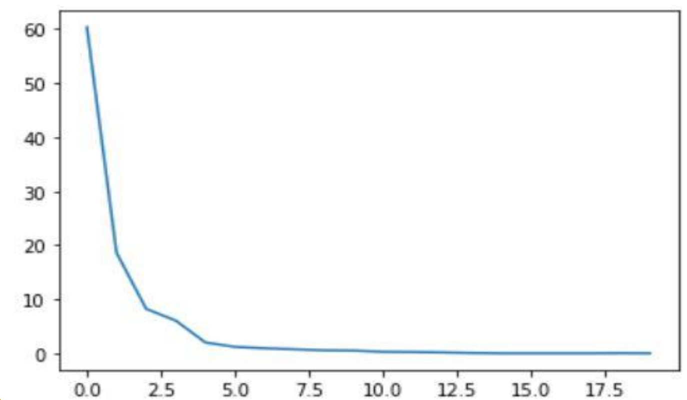
L'ACP: analyse en composantes principales

3/ Interprétation des résultats

```
60.37281604130095
18.608111457121364
8.232931586384147
6.0540261979412975
2.0021040327813675
1.194726199200963
0.9328078521624164
0.7443417793291216
0.5540301622503266
0.5127385407461053
0.2755339099126727
0.2480786328552258
0.16582102330115575
0.06847501762984184
0.00045272789061516776
-0.0002110815871110303
-2.808851705703031e-05
-2.1011969850041917e-05
0.0332307203139569
3.430095248338196e-05
```

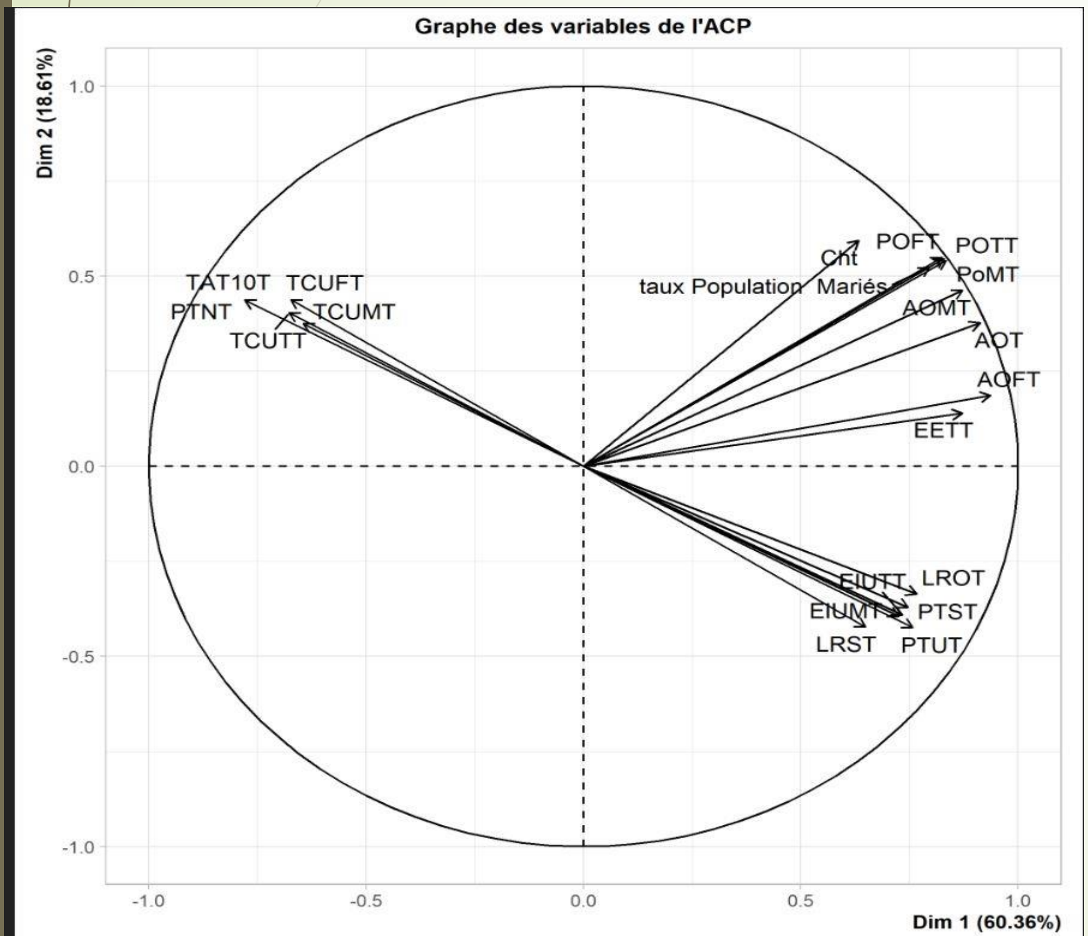
```
x=[k for k in range(20)]
plt.plot(x,j.T[1])
```

```
[<matplotlib.lines.Line2D at 0x1f78a2b7400>]
```



(voir annexe code n°7 , annexe)

L'ACP: analyse en composante principales

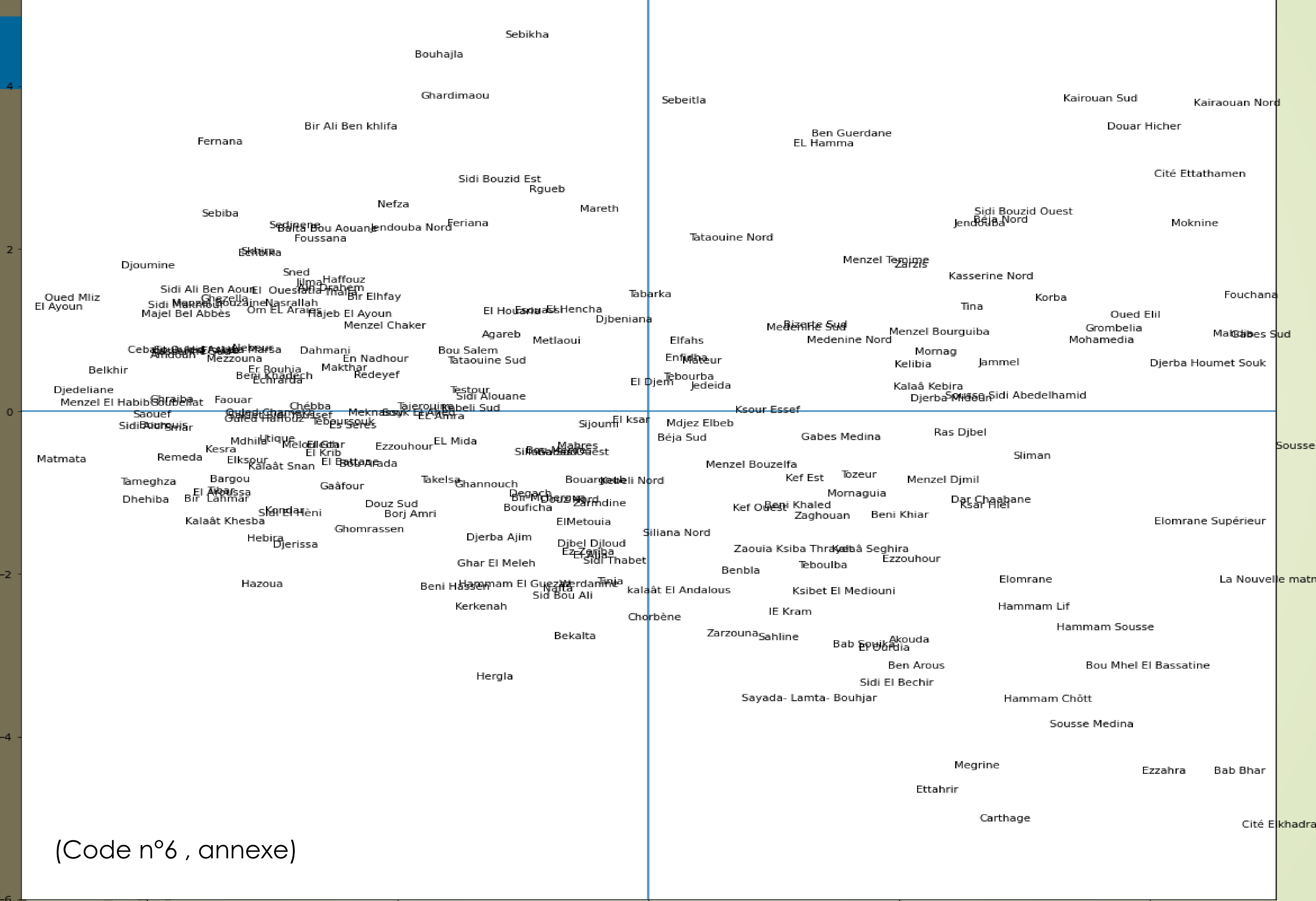


(Voir code n°8 , annexe)

		Axe 1	Axe 2
0	taux Population Mariés	0.79583628	0.52279973]
1	POTT	0.82974187	0.54516829]
2	POFT	0.82455878	0.54918204]
3	PoMT	0.83389067	0.54059246]
4	PTNT	-0.77952344	0.43650861]
5	PTST	0.74616949	-0.3705567]
6	PTUT	0.75744956	-0.4242675]
7	TAT10T	-0.7794443	0.43659419]
8	Cht	0.63263356	0.59378101]
9	TCUTT	-0.6770426	0.4038188]
10	TCUMT	-0.64428496	0.37536399]
11	TCUFT	-0.67320235	0.43822459]
12	AOT	0.91261937	0.37626126]
13	EIUTT	0.73332053	-0.39106914]
14	AOMT	0.87216619	0.46229505]
15	EIUMT	0.72552703	-0.39473207]
16	AOFT	0.93676187	0.18601617]
17	EETT	0.87168836	0.13935654]
18	LRST	0.64877815	-0.4228181]
19	LROT	0.76629271	-0.33592607]

(Voir code n°5 , annexe)

Représentation du nuage de points



(Code n°6 , annexe)

K means

1/ Le principe

Algorithme de partitionnement

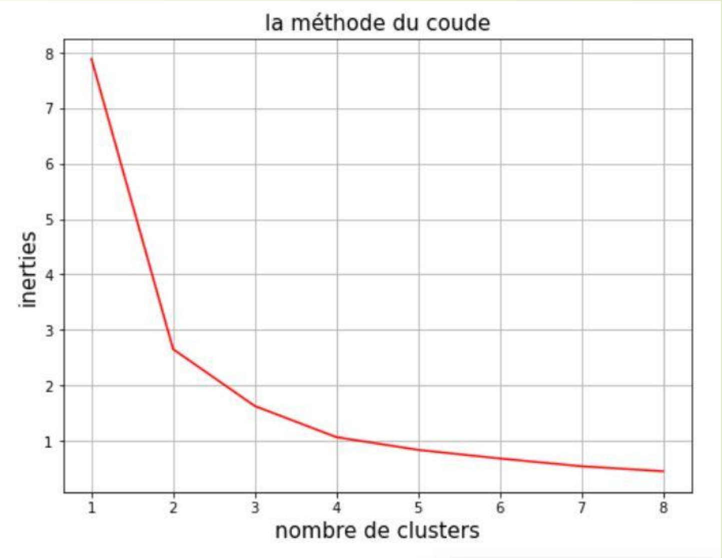
- 1/ choisir le nombre de clusters
- 2/ choisir au hasard parmi les individus les centres de clusters(centroïdes).
- 3/ affecter chaque individu au centre de cluster le plus proche.
- 4/ calculer le centre de gravité de chaque cluster.
- 5/ continuer jusqu’à ce que les centres de gravité et les centroïdes soient très proches.

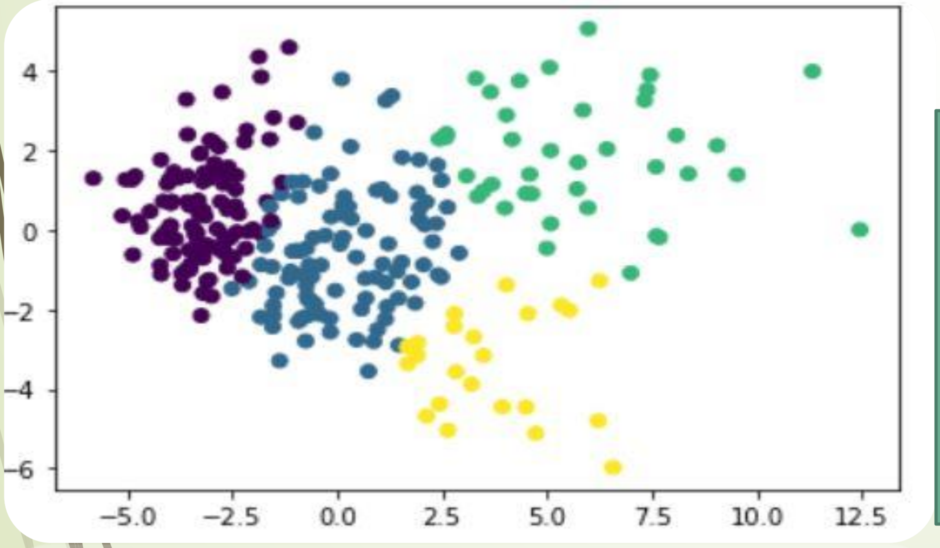
Kmeans

2/Interprétation du résultat

La méthode du coude est une méthode qui permet de choisir le nombre de groupe,

(voir code n°9 et n°10, annexe)





On a pu repartir les délégations en quatre groupe plus au moins homogènes. Le groupes en jaune présente des délégations

Akouda
Bab Bhar
Ben Arous
Bou Mhel El Bassatine
Carthage
Cité Elkhadra
El Menzah
El Ourdia
Elomrane
Elomrane Supérieur
Ettahrir
Ezzahra
Hammam Chôtt
Hammam Lif
Hammam Sousse
La Medina
La Nouvelle matmata
La Nouvelle Medina
Manouba
Megrine
Radès
Sidi El Bechir
Sousse Medina

conclusion

Une analyse des données nécessite des démonstrations mathématiques.

À partir des analyse et interprétation des résultats des deux méthodes adaptées : l'analyse en composante principale et kmeans on a pu avoir une première description du profil des inégalités en Tunisie:

Le nuage de points représenté à partir des axes principaux de l'ACP et à l'aide des corrélations entre les 20 variables initiales et les deux nouveaux axes, on peut décrire le comportement de chaque délégations face à toute les variables.

Le partitionnement des individus de notre base de données en 4 groupes nous permet d'avoir des partitions homogènes présentant les mêmes caractéristiques.

Merci pour votre attention.

Annexe

Code n°1

importation de données

```
Entrée [3]: import pandas as pd
df=pd.read_excel(r'C:\Users\user\Desktop\tipe\base\newnew.xlsx')
df.head()
```

Out[3]:

	NOM De la Délégations	taux Population Mariés	POTT	POFT	PoMT	PTNT	PTST	PTUT	TAT10T	Cht	...	TCUMT	TCUFT	AOT	EIUTT	AOI
0	Agareb	0.004066	0.003728	0.003679	0.003778	0.239356	0.298511	0.066707	0.238220	0.004462	...	0.165259	0.471037	0.003049	0.116031	0.0035
1	Ain Drahem	0.003367	0.003223	0.003289	0.003157	0.362733	0.268779	0.061218	0.361543	0.004467	...	0.207831	0.464088	0.002062	0.146410	0.0023
2	Akouda	0.003321	0.003141	0.003089	0.003193	0.106187	0.401152	0.180560	0.102679	0.001795	...	0.067251	0.181135	0.003737	0.237902	0.0033
3	Amdoun	0.002102	0.001929	0.001925	0.001934	0.393065	0.231350	0.043530	0.391746	0.001783	...	0.193182	0.414013	0.001441	0.089455	0.0017
4	Ariana Ville	0.011304	0.010424	0.010476	0.010372	0.044395	0.377759	0.427816	0.044002	0.006391	...	0.042103	0.084447	0.013803	0.648232	0.0113

5 rows × 21 columns

Code n°2

Centrage et réduction des données

```
z=df.drop(columns=['NOM De la Délégations'])
x=z.mean()
y=z.columns
r=std(z)
```

```
for i in range(20):
    for j in range (264):
      z.loc[j,[y[i]]]=(z.loc[j,[y[i]]]-x[i])/r[i]
```

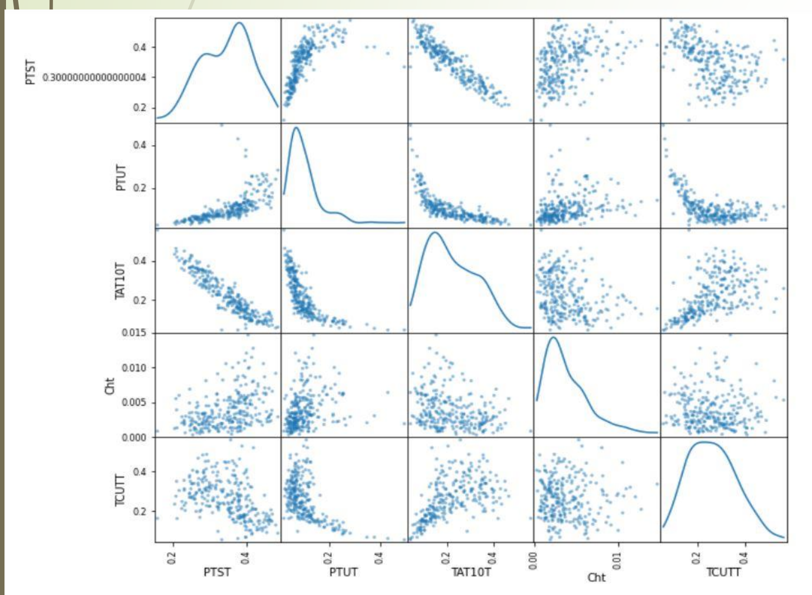
Annexe

Première lecture de la base

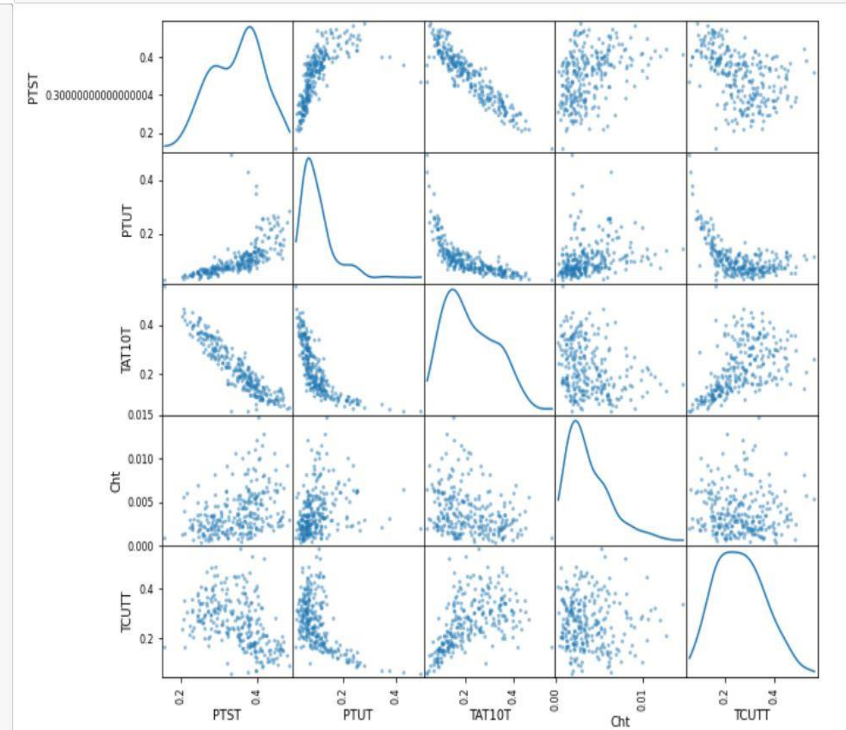
Code n°3

```
pd.plotting.scatter_matrix(df.iloc[:,0:6], diagonal='kde', figsize=(10,9));
```

```
Entrée [16]: pd.plotting.scatter_matrix(df.iloc[:,6:11], diagonal='kde', figsize=(10,9));
```




```
: pd.plotting.scatter_matrix(df.iloc[:,6:11], diagonal='kde', figsize=(10,9));
```



Annexe

```
r=z.values
a=(1/265)*matmul(r.T,r)
e= 10 ** (-3)
```

Code n°4

```
def puissance (a,e):
     x = rand(shape(a)[0], 1)
     x = x / norm(x)
     ax = matmul(a, x)
     err = ax - vdot(ax, x) * x
     while norm(err) > e :
            x = ax / norm(ax)
            ax = matmul(a, x)
            err = ax - vdot(ax, x) * x
     return x, vdot(ax, x)
```

Code n°5

```
def puissanceiteree(a, e):
    n = len(a)
    u, l =  puissance(a, e)
    vp = zeros(n)
    vp[0] = l
    vectp = zeros((n,n))
    vectp[:, 0] = u.reshape(n)
    for i in range(1, len(a)):
        a = a- l * matmul(u, u.T)
        u, l =  puissance(a, e)
        vp[i] = l
        vectp[:, i] = u.reshape(n)
    return vp, vectp
```

Annexe

Représentation des courbes

Nuage de points

Code n°6

```
l=[]
for i in range (264):
    l.append(df.loc[i,'NOM De la Délégations'])


P = zeros((2, shape(r)[0]))
plt.figure(figsize=(2,2))
for i in range(shape(r)[0]):
    P[0, i] = vdot(n[:, 0],  r[i, :])
    P[1, i] = vdot(n[:, 1],  r[i, :])
#scatter(P[0, :], P[1, :],edgecolor='black')


fig,ax=plt.subplots(figsize=(20,20))
ax.set_xlim(-5,5)
ax.set_ylim(-6,6)
for i,txt in enumerate(l):
   ax.annotate(txt,(P[0, i], P[1, i]))

plt.xlabel('first principal component')
plt.ylabel('second principal component')
plt.axhline(y=0,)
plt.axvline(x=0)
```

Code n°7

```
f,n=puissanceiteree(a, e)
```

```
j=zeros((20,2))
m=0
for i in range (20):
    m+=f[i]
for i in range(20):
    j[i][0]=f[i]
    j[i][1]=(f[i]/m) *100
    print(j[i][1])
```

Cercle de corrélation

Code n°8

```
x=[]
y=[]
l=z.columns
for i in range(20):
    x.append (corrcoef(P[0, :].T,r[:,i])[0,1])
    y.append(corrcoef(P[1, :].T,r[:,i])[0,1])

(fig, ax) = plt.subplots(figsize=(8, 8))
for i in range(0, 20):
    ax.arrow(0,0,x[i],y[i],head_width=0.1,head_length=0.1)
    plt.text(x[i]  + 0.05,y[i] + 0.05,z.columns.values[i])
an = np.linspace(0, 2 * np.pi, 100)
plt.plot(np.cos(an), np.sin(an))
plt.axis('equal')
ax.set_title('cercle de corrélation')
plt.xlabel("dim1 60.37%")
plt.ylabel("dim2 18.61%")
plt.show()
```

Annexe

Code n°9

```python
def méthodeducoude(df, column_indices, n_clusters=8, max_iter=300, tol=1e-04, init='k-means++', n_init=10, algorithm='auto'):
    import matplotlib.pyplot as plt
    valeursinerties = []
    for i in range(1, n_clusters+1):
        km = KMeans(n_clusters=i, max_iter=max_iter, tol=tol, init=init, n_init=n_init, random_state=1, algorithm=algorithm)
        km.fit_predict(df.iloc[:, column_indices])
        valeursinerties.append(km.inertia_)
    fig, ax = plt.subplots(figsize=(8, 6))
    plt.plot(range(1, n_clusters+1),valeursinerties , color='red')
    plt.xlabel('nombre de clusters', fontsize=15)
    plt.ylabel('inerties', fontsize=15)
    plt.title('la méthode du coude', fontsize=15)
    plt.grid()
    plt.show()
```

```python
méthodeducoude(df,[1,2,3,4,5,6,7,8,9,10])
```

```python
where(clusters == 3)#jaune
```

```
(array([  2,   5,  13,  29,  35,  39,  73,  77,  82,  83,  89,  91, 115,
       117, 118, 151, 154, 155, 162, 170, 198, 223, 239], dtype=int64),)
```

```python
where(clusters == 3)#jaune
```

```
(array([  2,   5,  13,  29,  35,  39,  73,  77,  82,  83,  89,  91, 115,
       117, 118, 151, 154, 155, 162, 170, 198, 223, 239], dtype=int64),)
```

```python
t=df.values
for i in[2,   5,  13,  29,  35,  39,  73,  77,  82,  83,  89,  91, 115,
       117, 118, 151, 154, 155, 162, 170, 198, 223, 239] :
    print( t[i,0] )
```

Code n°10

```python
def kmeans(X, k):
    n, d = shape(X)
    centroides = zeros((k, d))
    clusters = zeros(n)
    centroides_anciens = centroides.copy()
    for i in range(k):
        l = randint(n, size = (n // 4))
        centroides[i, :] = mean(X[l, :], axis = 0)
    #dist = zeros((k, n))
    for i in range (100):
        dist = sum((X[newaxis, :, :] - centroides[:, newaxis, :]) ** 2, axis = 2).T
        clusters = argmin (dist, axis = 1)
        for j in range(k):
            centroides[j, :] = mean(X[where(clusters == j)], axis = 0)
        if norm(centroides_anciens - centroides) < 0.001:
            break
        else:
            centroides_anciens = centroides.copy()
    return centroides, clusters
```

```python
centroides, clusters = kmeans(r,4)
```

```python
scatter(P[0, :], P[1, :], c = clusters)
where(clusters == 0)
```