**Sujet :**

les techniques du clustering hiérarchique

(CHA) et non-hiérarchique (Kmeans).

**Module**:

Data-mining

**Professeur**

Mr.Boufenar

**Module**:

Data-mining

**Professeur**

Mr.Boufenar

**Noms :**

* Benrabah Amira
* Smail Sabiha
* Zaiter Nouara

**Groupe 4**

**Noms :**

* Benrabah Amira
* Smail Sabiha
* Zaiter Nouara

**Groupe 4**

**Introduction**

Le clustering est une méthode d’apprentissage automatique qui consiste à regrouper des points de données par similarité ou par distance. C’est une méthode d’apprentissage non supervisée et une technique populaire d’analyse statistique des données. Il y’a deux façons de l’utiliser avec CAH (Classification Hiérarchique Ascendante) et la méthode K-means, c’est exactement le but de ce projet, on les appliquera dans deux jeu de données différents.

Voici un ensemble de points qui représentera notre jeu de données pour la première partie :

|  |  |  |
| --- | --- | --- |
| Point | X | Y |
| A | 1,95 | 0.97 |
| B | 1.62 | 0.74 |
| C | 3.12 | 1.85 |
| D | 0.91 | 1.09 |
| E | 2.37 | 4.11 |
| F | 5.20 | 2.52 |
| G | 5.74 | 5.04 |
| H | 3.00 | 3.47 |
| I | 4.70 | 3.65 |
| k | 4.97 | 3.32 |

**Partie 1/ TD**

**Single Linkage :** Commençons par calculer la distance euclidienne de ces données :

D(A,B)=D[ (x,y) , (a,b) ] = = = =0.40

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | A | B | C | D | E | F | G | H | I | K |
| A | 0 | 0.40 | 1.46 | 1.04 | 3.16 | 3.60 | 5.56 | 2.71 | 3.83 | 3.82 |
| B | 0.40 | 0 | 1.86 | 0.79 | 3.45 | 3.99 | 5.95 | 3.05 | 4.23 | 4.22 |
| C | 1.46 | 1.86 | 0 | 2.33 | 2.38 | 2.18 | 4.12 | 1.62 | 2.39 | 2.36 |
| D | 1.04 | 0.79 | 2.33 | 0 | 3.35 | 4.52 | 6.23 | 3.16 | 4.57 | 4.63 |
| E | 3.16 | 3.45 | 2.38 | 3.35 | 0 | 3.24 | 3.49 | 0.89 | 2.37 | 2.71 |
| F | 3.60 | 3.99 | 2.18 | 4.52 | 3.24 | 0 | 2.57 | 2.39 | 1.23 | 0.83 |
| G | 5.56 | 5.95 | 4.12 | 6.23 | 3.49 | 2.57 | 0 | 3.15 | 1.73 | 1.88 |
| H | 2.71 | 3.05 | 1.62 | 3.16 | 0.89 | 2.39 | 3.15 | 0 | 1.70 | 1.97 |
| I | 3.83 | 4.23 | 2.39 | 4.57 | 2.37 | 1.23 | 1.73 | 1.70 | 0 | 0.42 |
| K | 3.82 | 4.22 | 2.36 | 4.63 | 2.71 | 0.83 | 1.88 | 1.97 | 0.42 | 0 |

A

B

Ensuite, nous devrions mettre à jour la matrice ci-dessus :

Min [distance (A, B), Z], Z ϵ{C, D, E, F, G, H, I, K}

* Min [distance (A, B), C] = Min [distance (A, C), distance (B, C)] = (1.86, 1.46) = 1.46
* Min [distance (A, B), D] = Min (1.04, 0.79) = 0.79
* Min [distance (A, B), E] = Min (3.16, 3.45) = 3.16
* Min [distance (A, B), F] = Min (3.60, 3.99) = 3.60
* Min [distance (A, B), G] = Min (5.56, 5.95) = 5.56
* Min [distance (A, B), H] = Min (2.71, 3.05) = 2.71
* Min [distance (A, B), I] = Min (3.83, 4.23) = 3.83
* Min [distance (A, B), K] = Min (3.82, 4.22) = 3.82

La matrice de la distance de mise à jour pour le cluster A, B et encore une fois, nous sélectionnons la distance min :

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | A-B | C | D | E | F | G | H | I | K |
| A-B | 0 | 1.46 | 0.79 | 3.16 | 3.60 | 5.56 | 2.71 | 3.83 | 3.82 |
| C | 1.46 | 0 | 2.33 | 2.38 | 2.18 | 4.12 | 1.62 | 2.39 | 2.36 |
| D | 0.79 | 2.33 | 0 | 3.35 | 4.52 | 6.23 | 3.16 | 4.57 | 4.63 |
| E | 3.16 | 2.38 | 3.35 | 0 | 3.24 | 3.49 | 0.89 | 2.37 | 2.71 |
| F | 3.60 | 2.18 | 4.52 | 3.24 | 0 | 2.57 | 2.39 | 1.23 | 0.83 |
| G | 5.56 | 4.12 | 6.23 | 3.49 | 2.57 | 0 | 3.15 | 1.73 | 1.88 |
| H | 2.71 | 1.62 | 3.16 | 0.89 | 2.39 | 3.15 | 0 | 1.70 | 1.97 |
| I | 3.83 | 2.39 | 4.57 | 2.37 | 1.23 | 1.73 | 1.70 | 0 | 0.42 |
| K | 3.82 | 2.36 | 4.63 | 2.71 | 0.83 | 1.88 | 1.97 | 0.42 | 0 |

I

K

A

B

Nous répétons la même étape pour chaque cluster, nous calculons le minimum entre deux distances et choisissons la distance minimum jusqu'à la dernière étape où nous trouvons tous les clusters regroupes (Ps :Pour les détails du calcule , consultez le pdf CAH\_Détailler)

Pour la dernière étape :

Min [distance (AB-D-C-EH-IK-F), Z], Z ϵ {G}

* Min [distance (AB-D-C-EH-IK-F), (G)] = 1.73

|  |  |  |
| --- | --- | --- |
|  | AB-D-C-EH-IK-F | G |
| AB-D-C-EH-IK-F | 0 | 1.73 |
| G | 1.73 | 0 |

La matrice de la distance de mise à jour pour le cluster A, B, D, C, E, G, H, encore une fois, nous sélectionnons la distance min :

Dendrogramme du « Single linkage »

I

K

F

E

H

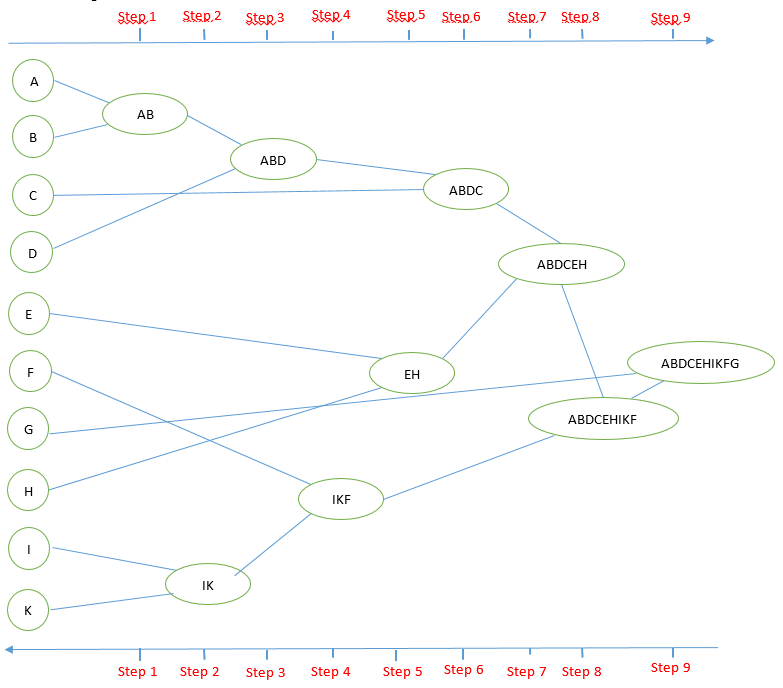
A

B

D

C

G



La hiérarchie du « Single linkage »

**Complete linkage :**

La Distance euclidienne :

D (A, B)=D [(x,y) , (a,b) ] = = = =0.40

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | A | B | C | D | E | F | G | H | I | K |
| A | 0 | 0.40 | 1.46 | 1.04 | 3.16 | 3.60 | 5.56 | 2.71 | 3.83 | 3.82 |
| B | 0.40 | 0 | 1.86 | 0.79 | 3.45 | 3.99 | 5.95 | 3.05 | 4.23 | 4.22 |
| C | 1.46 | 1.86 | 0 | 2.33 | 2.38 | 2.18 | 4.12 | 1.62 | 2.39 | 2.36 |
| D | 1.04 | 0.79 | 2.33 | 0 | 3.35 | 4.52 | 6.23 | 3.16 | 4.57 | 4.63 |
| E | 3.16 | 3.45 | 2.38 | 3.35 | 0 | 3.24 | 3.49 | 0.89 | 2.37 | 2.71 |
| F | 3.60 | 3.99 | 2.18 | 4.52 | 3.24 | 0 | 2.57 | 2.39 | 1.23 | 0.83 |
| G | 5.56 | 5.95 | 4.12 | 6.23 | 3.49 | 2.57 | 0 | 3.15 | 1.73 | 1.88 |
| H | 2.71 | 3.05 | 1.62 | 3.16 | 0.89 | 2.39 | 3.15 | 0 | 1.70 | 1.97 |
| I | 3.83 | 4.23 | 2.39 | 4.57 | 2.37 | 1.23 | 1.73 | 1.70 | 0 | 0.42 |
| K | 3.82 | 4.22 | 2.36 | 4.63 | 2.71 | 0.83 | 1.88 | 1.97 | 0.42 | 0 |

A

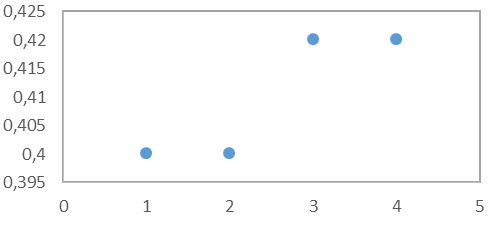
B

Ensuite, nous devrions mettre à jour la matrice ci-dessus :

Max [distance (A, B), Z], Z ϵ {C, D, E, F, G, H, I, K}

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | A-B | C | D | E | F | G | H | I | K |
| A-B | 0 | 1.86 | 1.04 | 3.45 | 3.99 | 5.95 | 3.05 | 4.23 | 4.22 |
| C | 1.86 | 0 | 2.33 | 2.38 | 2.18 | 4.12 | 1.62 | 2.39 | 2.36 |
| D | 1.04 | 2.33 | 0 | 3.35 | 4.52 | 6.23 | 3.16 | 4.57 | 4.63 |
| E | 3.45 | 2.38 | 3.35 | 0 | 3.24 | 3.49 | 0.89 | 2.37 | 2.71 |
| F | 3.99 | 2.18 | 4.52 | 3.24 | 0 | 2.57 | 2.39 | 1.23 | 0.83 |
| G | 5.95 | 4.12 | 6.23 | 3.49 | 2.57 | 0 | 3.15 | 1.73 | 1.88 |
| H | 3.05 | 1.62 | 3.16 | 0.89 | 2.39 | 3.15 | 0 | 1.70 | 1.97 |
| I | 4.23 | 2.39 | 4.57 | 2.37 | 1.23 | 1.73 | 1.70 | 0 | 0.42 |
| K | 4.22 | 2.36 | 4.63 | 2.71 | 0.83 | 1.88 | 1.97 | 0.42 | 0 |

* Max [distance (A, B), C] = Max [distance (A, C), distance (B, C)] = Max (1.86, 1.46) = 1.86
* Max [distance (A, B), D] = Max (1.04, 0.79) = 1.04
* Max [distance (A, B), E] = Max (3.16, 3.45) = 3.45
* Max [distance (A, B), F] = Max (3.60, 3.99) = 3.99
* Max [distance (A, B), G] = Max (5.56, 5.95) = 5.95
* Max [distance (A, B), H] = Max (2.71, 3.05) = 3.05
* Max [distance (A, B), I] =



I

K

A

B

Max (3.83, 4.23) = 4.23

* Max [distance (A, B), K] =

Max (3.82, 4.22) = 4.22

La matrice de la distance de mise à

jour pour le cluster A, B et

encore une fois, nous sélectionnons

la distance minimale :

Nous répétons la même étape pour chaque cluster, nous calculons le maximum entre deux distances et choisissons la distance minimum jusqu'à la dernière étape où nous trouvons tous les clusters regroupes (Ps : Pour les détails du calcule, consultez le pdf CAH\_Détailler)

|  |  |  |
| --- | --- | --- |
|  | AB-D-C-EH | IK-F-G |
| AB-D-C-EH | 0 | 6.23 |
| IK-F-G | 6.23 | 0 |

Pour la dernière étape : Nous répétons la même chose que nous avons faite avec AB-D-C à AB-D-C-EH : Max [distance (AB-D-C-EH), Z], Z ϵ {IK-F-G}

* Max [distance (AB-D-C-EH), (IK-F-G)] = 6.23

La matrice de la distance de mise à jour pour le cluster I, K, F, G et encore une fois, nous sélectionnons la distance min :

A

B

D

C

G

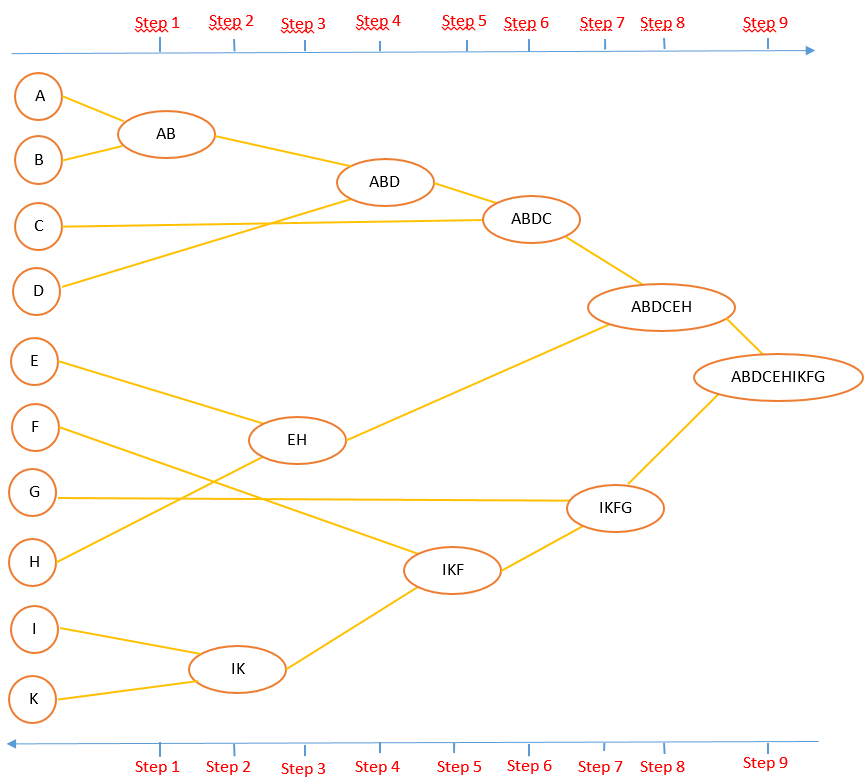
F

I

K

E

H



La hiérarchie du « Complete linkage »

Dendrogramme du « Complete linkage »

**Average linkage :**

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | A | B | C | D | E | F | G | H | I | K |
| A | 0 | 0.40 | 1.46 | 1.04 | 3.16 | 3.60 | 5.56 | 2.71 | 3.83 | 3.82 |
| B | 0.40 | 0 | 1.86 | 0.79 | 3.45 | 3.99 | 5.95 | 3.05 | 4.23 | 4.22 |
| C | 1.46 | 1.86 | 0 | 2.33 | 2.38 | 2.18 | 4.12 | 1.62 | 2.39 | 2.36 |
| D | 1.04 | 0.79 | 2.33 | 0 | 3.35 | 4.52 | 6.23 | 3.16 | 4.57 | 4.63 |
| E | 3.16 | 3.45 | 2.38 | 3.35 | 0 | 3.24 | 3.49 | 0.89 | 2.37 | 2.71 |
| F | 3.60 | 3.99 | 2.18 | 4.52 | 3.24 | 0 | 2.57 | 2.39 | 1.23 | 0.83 |
| G | 5.56 | 5.95 | 4.12 | 6.23 | 3.49 | 2.57 | 0 | 3.15 | 1.73 | 1.88 |
| H | 2.71 | 3.05 | 1.62 | 3.16 | 0.89 | 2.39 | 3.15 | 0 | 1.70 | 1.97 |
| I | 3.83 | 4.23 | 2.39 | 4.57 | 2.37 | 1.23 | 1.73 | 1.70 | 0 | 0.42 |
| K | 3.82 | 4.22 | 2.36 | 4.63 | 2.71 | 0.83 | 1.88 | 1.97 | 0.42 | 0 |

D (A, B)=D [(x,y) , (a,b) ] = = = =0.40

A

B

Maintenant, nous avons la distance minimale, qui est le lien entre A et B de sorte que le dendrogramme est :

Ensuite, nous devrions mettre à jour la matrice ci-dessus :

AVG [distance (A, B), Z], Z ϵ {C, D, E, F, G, H, I, K}

* AVG [distance (A, B), C] =[distance (A, C), distance (B, C)] = (1.86, 1.46) = 1.66
* AVG [distance (A, B), D] = (1.04, 0.79) = 2
* AVG [distance (A, B), E] = (3.16, 3.45) = 3.21
* AVG [distance (A, B), F] = (3.60, 3.99) = 3.795
* AVG [distance (A, B), G] = (5.56, 5.95) = 5.75
* AVG [distance (A, B), H] = (2.71, 3.05) =2.88
* AVG [distance (A, B), I] = (3.83, 4.23) = 4.03
* AVG [distance (A, B), K] = (3.82, 4.22) = 4.02

Nous répétons la même étape pour chaque cluster, nous calculons le moyenne entre deux distances et choisissons la distance minimum jusqu'à la dernière étape où nous trouvons tous les clusters regroupes (Ps :Pour les détails de calcule , consultez le pdf CAH\_Détailler)

|  |  |  |
| --- | --- | --- |
|  | AB-C-D | IK-F-G -EH |
| AB-C-D | 0 | 3.712 |
| IK-F-G -EH | 3.712 | 0 |

Nous répétons la même chose que nous avons faite avec AB-C à IK-F-G -EH :

AVG [distance (IK-F-G -EH), Z], Z ϵ {AB-C-D}

* AVG [distance (IK-F-G -EH), (AB-C-D)] = 3.712

La matrice de la distance de mise à jour pour le cluster A, B, C, D, E, H et encore une fois, nous sélectionnons la distance min :

Dendrogramme du « Average linkage »

A

B

C

D

E

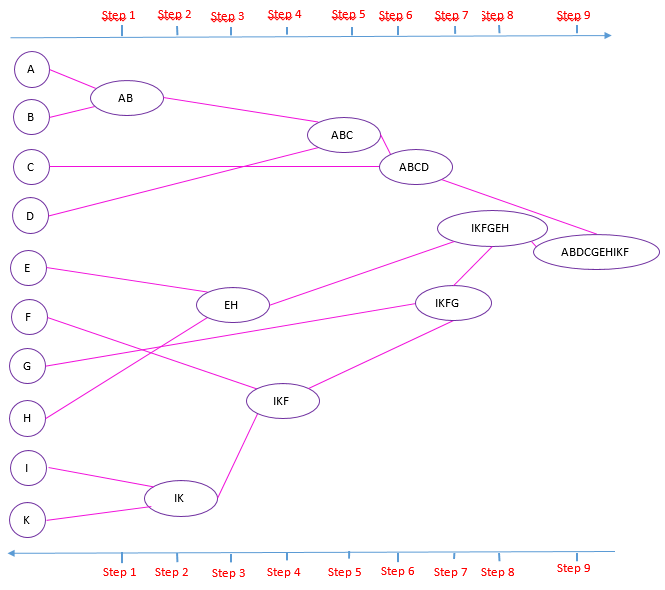
H

I

K

F

G



La hiérarchie du « Average linkage »

Apres avoir coupé les trois dendrogramme, nous nous retrouverons dans Linkage simple et complet 5 grappe mais pour Linkage average  nous avons 6 groupes comme suit :

Linkage simple : < coupe au niveau 1 >   A B D C E H I K F G

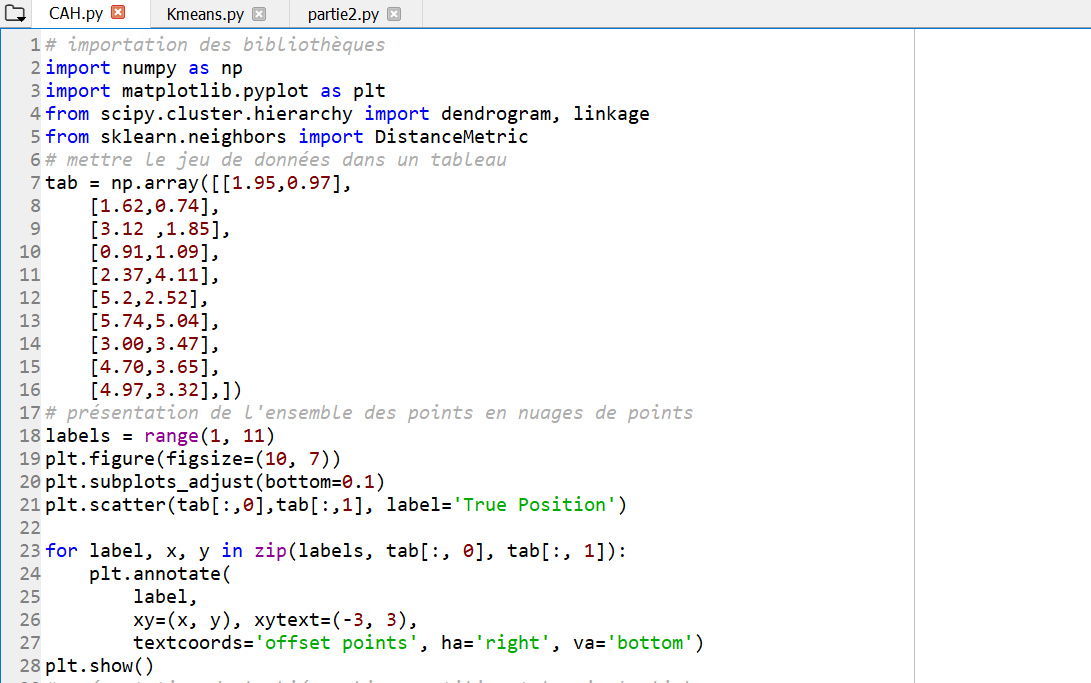
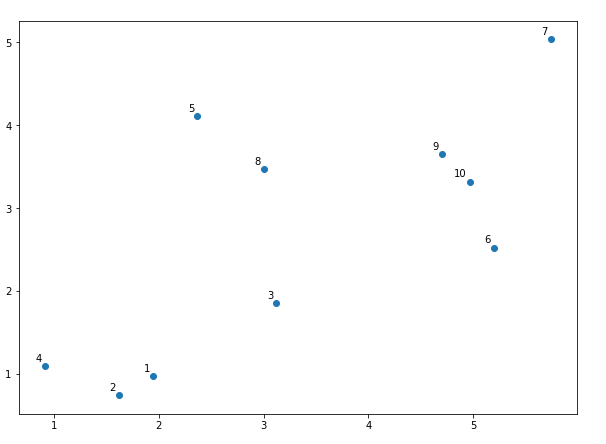
Linkage complet : < coupe au niveau 1.5  A B D C E H I K F G

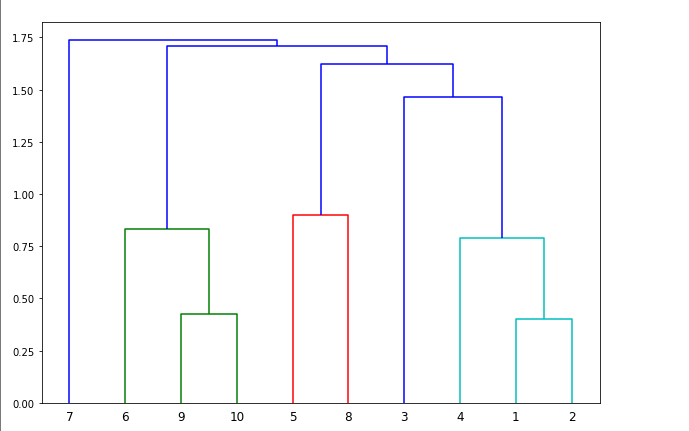
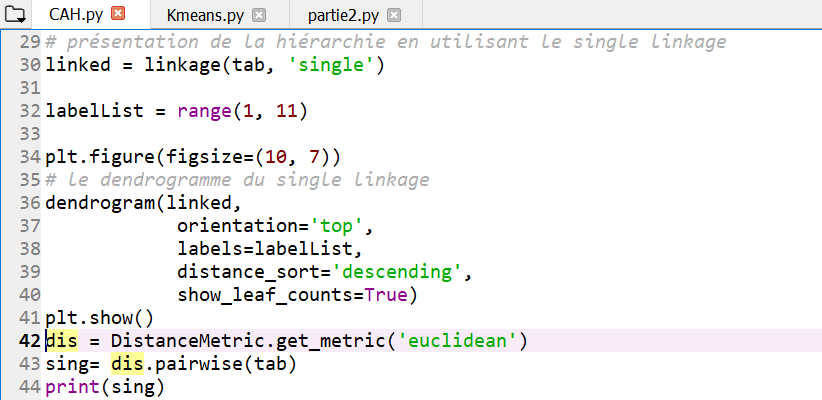
Linkage average : < coupe au niveau 1.40 > A B C D E H I K F G

**Partie TP**

Dans cette partie, nous allons aborder le côté programmation de cet algorithme (CAH) en utilisant le langage de programmation Python avec sa bibliothèque *Scikit-learn*:

Pour commencer on procède à l’importation des bibliothèques utiles pour cet algorithme, on présente notre ensemble de points en tableau, et enfin dessiner ces données sous forme de nuages de points avec la bibliothèque  *matplotlib*.

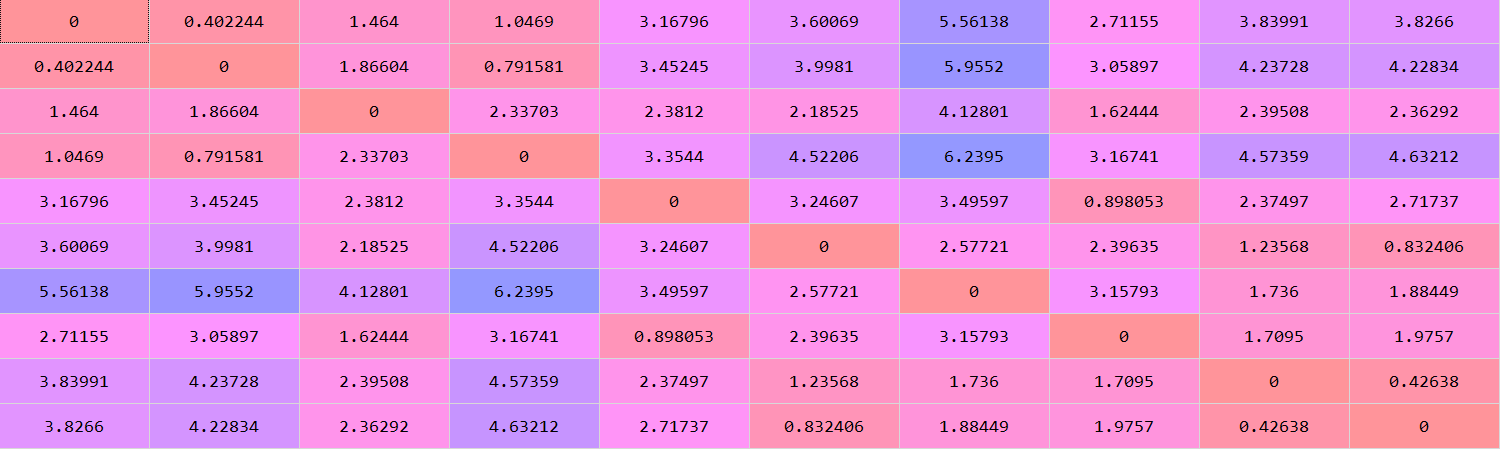
On constate que la distribution des données est répartie dans l’ensemble de l’espace des données, et certains points sont à côté des uns des autres et que instinctivement, vont former des clusters. Après ça, en utilisant la bibliothèque *scipy* et ses fonction prédéfinies nous allons présenter la hiérarchie et dendrogramme pour chaque linkage :

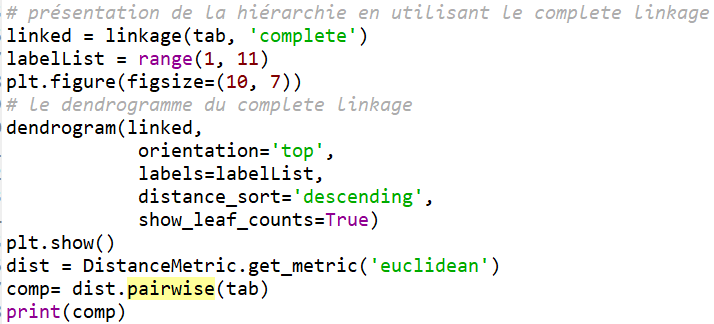
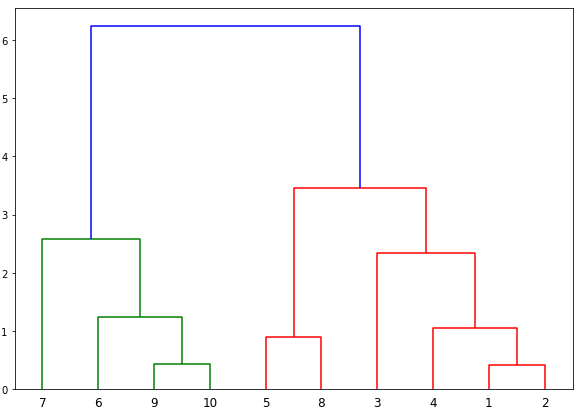
**Single linkage :**

Les chiffres de l’axe des abscisses représentent les points :

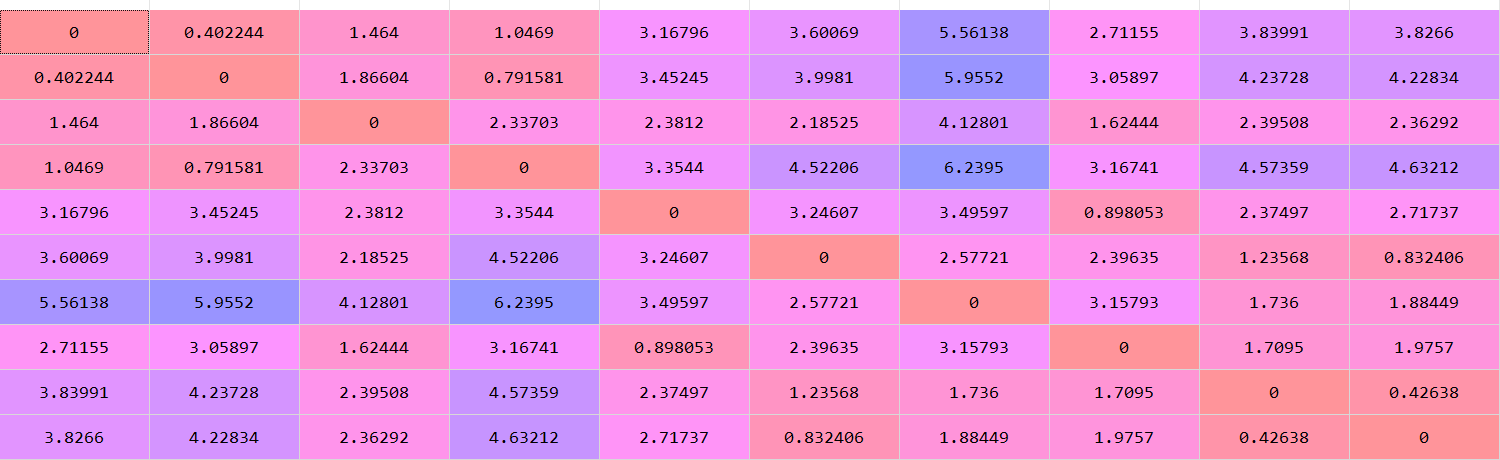
|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| A | B | C | D | E | F | G | H | I | J |
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |

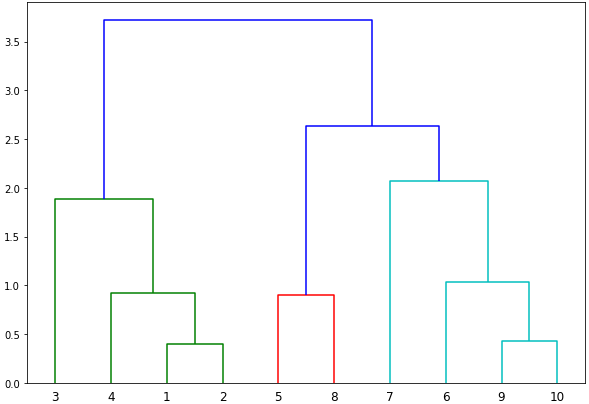
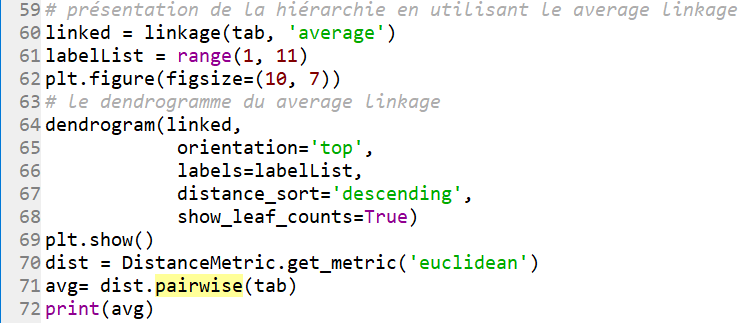
Le tableau de la métrique en utilisant euclédienne correspondant :



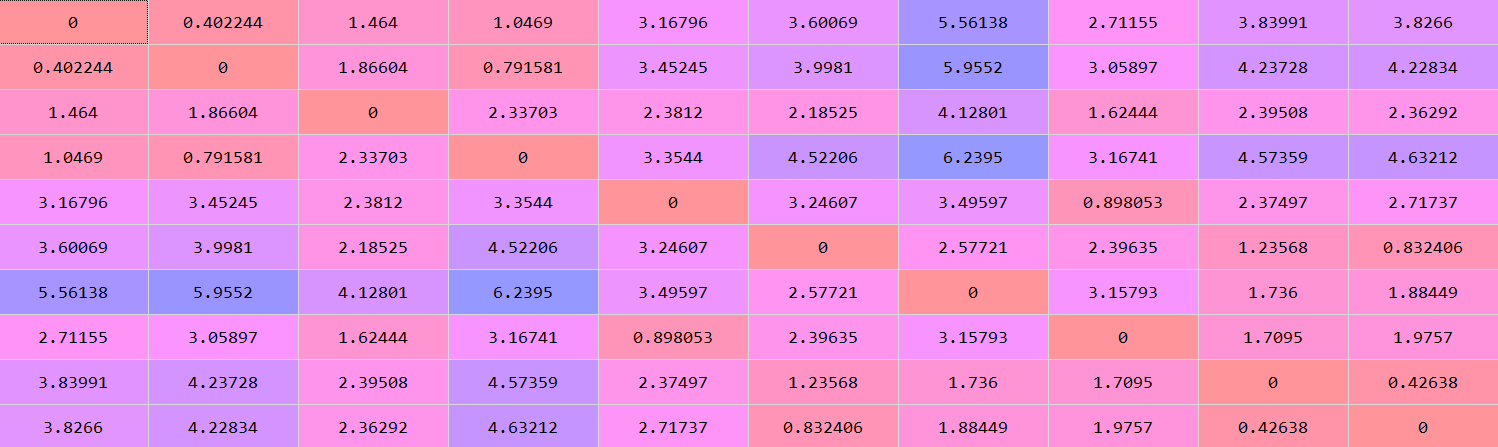
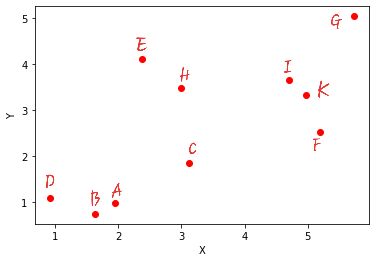
**Complete linkage :**

Le tableau de la métrique en utilisant euclédienne correspondant :



**Average linkage :**

Le tableau de la métrique en utilisant euclédienne correspondant :

******Kmeans :**

La matrice de distance de Manhattan entre différents objets est donnée dans le tableau.

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | A | B | C | D | E | F | G | H | I | K |
| A | 0 | 0.56 | 2.05 | 1.16 | 3.56 | 4.8 | 7.86 | 3.55 | 5.43 | 5.37 |
| B |  | 0 | 2.61 | 1.06 | 4.16 | 5.36 | 8.42 | 4.11 | 5.99 | 5.93 |
| C |  |  | 0 | 2.97 | 3.01 | 2.7 | 5.81 | 1.74 | 3.2 | 3.32 |
| D |  |  |  | 0 | 4.48 | 5.72 | 8.78 | 4.47 | 6.35 | 6.29 |
| E |  |  |  |  | 0 | 1.24 | 4.26 | 1.27 | 2.97 | 3.39 |
| F |  |  |  |  |  | 0 | 3.06 | 3.15 | 1.63 | 1.03 |
| G |  |  |  |  |  |  | 0 | 4.31 | 2.43 | 2.49 |
| H |  |  |  |  |  |  |  | 0 | 1.88 | 2.12 |
| I |  |  |  |  |  |  |  |  | 0 | 0.6 |
| K |  |  |  |  |  |  |  |  |  | 0 |

**Initialisation Step :**

Nous choisissons en utilisant la mesure de distance Manhattan (dans notre cas) les deux objets les plus éloignés (D, G)

|  |  |  |
| --- | --- | --- |
|  | Objects | Centroïdes |
| Cluster 1 | D | (0.91 , 1.09) |
| Cluster 2 | G | (5.74 , 5.04) |

**1er itération :**

**Assignement step :**

|  |  |  |  |
| --- | --- | --- | --- |
| Object | Dist(i,D) | Dist(i,G) | Assignement cluster |
| A | 1.16 | 7.86 | D |
| B | 1.06 | 8.42 | D |
| C | 2.96 | 5.81 | D |
| D | 0.0 | 8.78 | D |
| E | 4.48 | 4.3 | G |
| F | 5.72 | 3.06 | G |
| G | 8.78 | 0.0 | G |
| H | 4.47 | 4.31 | G |
| I | 6.35 | 2.43 | G |
| K | 6.29 | 2.49 | G |

**Update step :**

|  |  |  |
| --- | --- | --- |
|  | Objects | Centroïdes |
| Cluster 1 | {A,B,C,D} | U1=(1.9 , 1.17) |
| Cluster 2 | {E,F,G,H,I,K} | U2=(4.33 , 3.7) |

**2éme itération :**

|  |  |  |  |
| --- | --- | --- | --- |
| Object | Dist(i,U1) | Dist(i,U2) | Assignement cluster |
| A | 0.25 | 5.11 | U1 |
| B | 0.71 | 5.67 | U1 |
| C | 1.09 | 3.06 | U1 |
| D | 1.07 | 6.03 | U1 |
| E | 3.41 | 2.37 | U2 |
| F | 4.65 | 2.05 | U2 |
| G | 7.71 | 2.75 | U2 |
| H | 3.4 | 1.56 | U2 |
| I | 5.28 | 0.42 | U2 |
| K | 5.22 | 1.02 | U2 |

**Update step :**

|  |  |  |
| --- | --- | --- |
|  | Objects | Centroïdes |
| Cluster 1 | {A,B,C,D} | U1=(1.9 , 1.17) |
| Cluster 2 | {E,F,G,H,I,K} | U2=(4.33 , 3.7) |

* **Stop.** No new relocation

Maintenait on essaie de calculer l'inertie et la distorsion en faisant varier le K valeur dans la plage {1 .. 10}

K=3

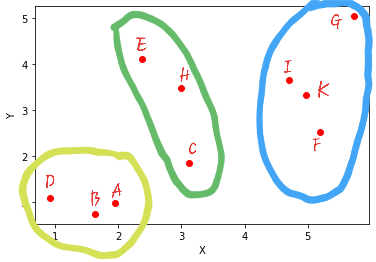
**Initialisation Step :**

|  |  |  |
| --- | --- | --- |
|  | Objects | Centroïdes |
| Cluster 1 | D | U1=(0.91 , 1.09) |
| Cluster 2 | G | U2=(5.74 , 5.04) |
| Cluster 3 | H | U3= (3 , 3.47) |

**Last Assignement step :**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Object | Dist(i,U1) | Dist(i,U2) | Dist(i,U3) | Relocation |
| A | 0.5 | 5.86 | 3.05 | U1 |
| B | 0.32 | 6.42 | 3.61 | U1 |
| C | 2.55 | 3.81 | 1.58 | U3 |
| D | 0.74 | 6.78 | 3.97 | U1 |
| E | 4.06 | 3.26 | 1.43 | U3 |
| F | 5.3 | 1.16 | 2.99 | U2 |
| G | 8.36 | 2 | 4.81 | U2 |
| H | 4.05 | 2.31 | 0.5 | U3 |
| I | 5.93 | 0.47 | 2.38 | U2 |
| K | 5.87 | 0.49 | 2.31 | U2 |

**Last Update step :**



|  |  |  |
| --- | --- | --- |
|  | Objects | Centroïdes |
| Cluster 1 | {A,B,D} | U1=(1.49 , 0.93) |
| Cluster 2 | {F,G,I,K} | U2=(5.15 , 3.63) |
| Cluster 3 | {C,E,H} | U3= (2.83 , 3.14) |

K=4

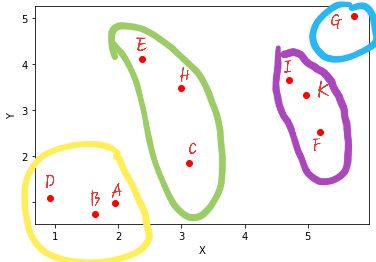
**Initialisation Step :**

|  |  |  |
| --- | --- | --- |
|  | Objects | Centroïdes |
| Cluster 1 | D | U1=(0.91 , 1.09) |
| Cluster 2 | G | U2=(5.74 , 5.04) |
| Cluster 3 | H | U3= (3 , 3.47) |
| Cluster 4 | F | U4=(5.2 , 2.52) |

**Last Assignement step :**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Object | Dist(i,U1) | Dist(i,U2) | Dist(i,U3) | Dist(i,U4) | Relocation |
| A | 0.5 | 7.86 | 3.05 | 5.19 | U1 |
| B | 0.32 | 8.42 | 3.61 | 5.75 | U1 |
| C | 2.55 | 5.81 | 1.58 | 3.14 | U3 |
| D | 0.74 | 8.78 | 3.97 | 6.11 | U1 |
| E | 4.06 | 4.3 | 1.43 | 3.53 | U3 |
| F | 5.3 | 3.06 | 2.99 | 0.89 | U4 |
| G | 8.36 | 0.0 | 4.81 | 2.67 | U2 |
| H | 4.05 | 4.31 | 0.5 | 2.26 | U3 |
| I | 5.93 | 2.43 | 2.38 | 0.74 | U4 |
| K | 5.87 | 2.49 | 2.32 | 0.18 | U4 |

**Last Update step :**



|  |  |  |
| --- | --- | --- |
|  | Objects | Centroïdes |
| Cluster 1 | {A,B,D} | U1=(1.49 , 0.93) |
| Cluster 2 | {G} | U2=(5.74 , 5.04) |
| Cluster 3 | {E,H,C} | U3= (2.83 , 3.14) |
| Cluster 4 | {F,I,K} | U4=(4.95 , 3.16) |

K=5

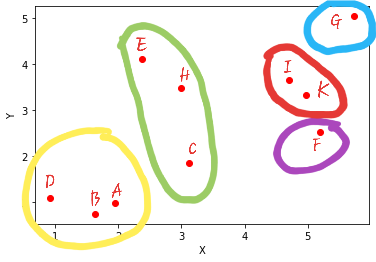
**Initialisation Step :**

|  |  |  |
| --- | --- | --- |
|  | Objects | Centroïdes |
| Cluster 1 | D | U1=(0.91 , 1.09) |
| Cluster 2 | G | U2=(5.74 , 5.04) |
| Cluster 3 | H | U3= (3 , 3.47) |
| Cluster 4 | F | U4=(5.2 , 2.52) |
| Cluster 5 | I | U5=(4.7 , 3.65) |

**Last Assignement step :**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Object | Dist(i,U1) | Dist(i,U2) | Dist(i,U3) | Dist(i,U4) | Dist(i,U5) | Relocation |
| A | 0.5 | 7.86 | 3.05 | 4.8 | 5.39 | U1 |
| B | 0.32 | 8.42 | 3.61 | 5.36 | 5.95 | U1 |
| C | 2.55 | 5.81 | 1.58 | 2.75 | 3.34 | U3 |
| D | 0.74 | 8.78 | 3.97 | 5.72 | 6.31 | U1 |
| E | 4.06 | 4.3 | 1.43 | 4.42 | 3.03 | U3 |
| F | 5.3 | 3.06 | 2.99 | 0.0 | 1.33 | U4 |
| G | 8.36 | 0.0 | 4.81 | 3.06 | 2.47 | U2 |
| H | 4.05 | 4.31 | 0.5 | 3.15 | 1.84 | U3 |
| I | 5.93 | 2.43 | 2.38 | 1.63 | 0.3 | U5 |
| K | 5.87 | 2.49 | 2.32 | 1.03 | 0.3 | U5 |

**Last Update step :**



|  |  |  |
| --- | --- | --- |
|  | Objects | Centroïdes |
| Cluster 1 | {A,B,D} | U1=(1.49 , 0.93) |
| Cluster 2 | {G} | U2=(5.74 , 5.04) |
| Cluster 3 | {E,H,C} | U3= (2.83 , 3.14) |
| Cluster 4 | {F} | U4=(5.2 , 2.52) |
| Cluster 5 | {I,K} | U5=(4.83 , 3.48) |

K=6

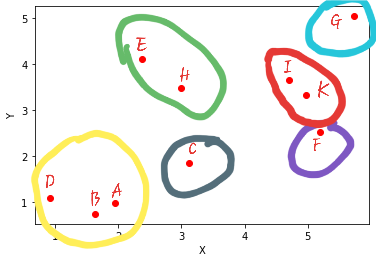
**Initialisation Step :**

|  |  |  |
| --- | --- | --- |
|  | Objects | Centroïde |
| Cluster 1 | D | U1=(0.91 , 1.09) |
| Cluster 2 | G | U2=(5.74 , 5.04) |
| Cluster 3 | H | U3= (3 , 3.47) |
| Cluster 4 | F | U4=(5.2 , 2.52) |
| Cluster 5 | I | U5=(4.7 , 3.65) |
| Cluster 6 | C | U6=(3.12 , 1.85) |

**Last Assignement step :**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Object | Dist(i,U1) | Dist(i,U2) | Dist(i,U3) | Dist(i,U4) | Dist(i,U5) | Dist(i,U6) | Relocation |
| A | 0.5 | 7.86 | 3.55 | 4.8 | 5.39 | 2.05 | U1 |
| B | 0.32 | 8.42 | 4.11 | 5.36 | 5.95 | 2.61 | U1 |
| C | 2.55 | 5.81 | 2.38 | 2.75 | 3.34 | 0.0 | U6 |
| D | 0.74 | 8.78 | 4.47 | 5.72 | 6.31 | 2.97 | U1 |
| E | 4.06 | 4.3 | 0.63 | 4.42 | 3.03 | 3.01 | U3 |
| F | 5.3 | 3.06 | 3.79 | 0.0 | 1.33 | 2.75 | U4 |
| G | 8.36 | 0.0 | 4.31 | 3.06 | 2.47 | 5.81 | U2 |
| H | 4.05 | 4.31 | 0.63 | 3.15 | 1.84 | 1.74 | U3 |
| I | 5.93 | 2.43 | 2.16 | 1.63 | 0.3 | 3.38 | U5 |
| K | 5.87 | 2.49 | 2.76 | 1.03 | 0.3 | 3.32 | U5 |

**Last Update step :**



|  |  |  |
| --- | --- | --- |
|  | Objects | Centroïde |
| Cluster 1 | {A,B,D} | U1=(1.49 , 0.93) |
| Cluster 2 | {G} | U2=(5.74 , 5.04) |
| Cluster 3 | {E,H} | U3= (2.68 , 3.79) |
| Cluster 4 | {F} | U4=(5.2 , 2.52) |
| Cluster 5 | {I,K} | U5=(4.83 , 3.48) |
| Cluster 6 | {C} | U6=(3.12 , 1.85) |

K=7

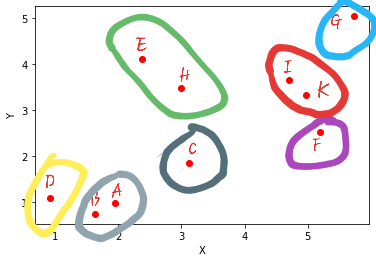
**Initialisation Step :**

|  |  |  |
| --- | --- | --- |
|  | Objects | Centroïde |
| Cluster 1 | D | U1=(0.91 , 1.09) |
| Cluster 2 | G | U2=(5.74 , 5.04) |
| Cluster 3 | H | U3= (3 , 3.47) |
| Cluster 4 | F | U4=(5.2 , 2.52) |
| Cluster 5 | I | U5=(4.7 , 3.65) |
| Cluster 6 | C | U6=(3.12 , 1.85) |
| Cluster 7 | A | U7=(1.95 , 0.97) |

**Last Assignement step :**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Object | Dist(i,U1) | Dist(i,U2) | Dist(i,U3) | Dist(i,U4) | Dist(i,U5) | Dist(i,U6) | Dist(i,U7) | Relocation |
| A | 1.16 | 7.86 | 3.55 | 4.8 | 5.39 | 2.05 | 0.29 | U7 |
| B | 1.06 | 8.42 | 4.11 | 5.36 | 5.95 | 2.61 | 0.27 | U7 |
| C | 2.97 | 5.81 | 2.38 | 2.75 | 3.34 | 0.0 | 2.34 | U6 |
| D | 0.0 | 8.78 | 4.47 | 5.72 | 6.31 | 2.97 | 1.11 | U1 |
| E | 4.48 | 4.3 | 0.63 | 4.42 | 3.03 | 3.01 | 3.85 | U3 |
| F | 5.72 | 3.06 | 3.79 | 0.0 | 1.33 | 2.75 | 5.09 | U4 |
| G | 8.78 | 0.0 | 4.31 | 3.06 | 2.47 | 5.81 | 8.15 | U2 |
| H | 4.47 | 4.31 | 0.63 | 3.15 | 1.84 | 1.74 | 3.84 | U3 |
| I | 6.35 | 2.43 | 2.16 | 1.63 | 0.3 | 3.38 | 5.72 | U5 |
| K | 6.29 | 2.49 | 2.76 | 1.03 | 0.3 | 3.32 | 5.67 | U5 |

**Last Update step :**



|  |  |  |
| --- | --- | --- |
|  | Objects | Centroides |
| Cluster 1 | {D} | U1=(0.91 , 1.09) |
| Cluster 2 | {G} | U2=(5.74 , 5.04) |
| Cluster 3 | {E,H} | U3= (2.68 , 3.79) |
| Cluster 4 | {F} | U4=(5.2 , 2.52) |
| Cluster 5 | {I,K} | U5=(4.83 , 3.48) |
| Cluster 6 | {C} | U6=(3.12 , 1.85) |
| Cluster 7 | {A,B} | U7=(1.78 , 0.85) |

K=8

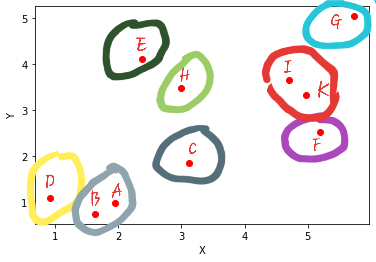
**Initialisation Step :**

|  |  |  |
| --- | --- | --- |
|  | Objects | Centroides |
| Cluster 1 | D | U1=(0.91 , 1.09) |
| Cluster 2 | G | U2=(5.74 , 5.04) |
| Cluster 3 | H | U3= (3 , 3.47) |
| Cluster 4 | F | U4=(5.2 , 2.52) |
| Cluster 5 | I | U5=(4.7 , 3.65) |
| Cluster 6 | C | U6=(3.12 , 1.85) |
| Cluster 7 | A | U7=(1.95 , 0.97) |
| Cluster 8 | E | U8=(2.37 , 4.11) |

**Last Assignement step :**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Object | Dist(i,U1) | Dist(i,U2) | Dist(i,U3) | Dist(i,U4) | Dist(i,U5) | Dist(i,U6) | Dist(i,U7) | Dist(i,U8) | Relocation |
| A | 1.16 | 7.86 | 3.55 | 4.8 | 5.39 | 2.05 | 0.29 | 3.56 | U7 |
| B | 1.06 | 8.42 | 4.11 | 5.36 | 5.95 | 2.61 | 0.27 | 4.12 | U7 |
| C | 2.97 | 5.81 | 1.74 | 2.75 | 3.34 | 0.0 | 2.34 | 3.01 | U6 |
| D | 0.0 | 8.78 | 4.47 | 5.72 | 6.31 | 2.97 | 1.11 | 4.48 | U1 |
| E | 4.48 | 4.3 | 1.27 | 4.42 | 3.03 | 3.01 | 3.85 | 0.0 | U8 |
| F | 5.72 | 3.06 | 3.15 | 0.0 | 1.33 | 2.75 | 5.09 | 4.42 | U4 |
| G | 8.78 | 0.0 | 4.31 | 3.06 | 2.47 | 5.81 | 8.15 | 4.3 | U2 |
| H | 4.47 | 4.31 | 0.0 | 3.15 | 1.84 | 1.74 | 3.84 | 1.27 | U3 |
| I | 6.35 | 2.43 | 1.88 | 1.63 | 0.3 | 3.38 | 5.72 | 2.79 | U5 |
| K | 6.29 | 2.49 | 2.12 | 1.03 | 0.3 | 3.32 | 5.67 | 3.39 | U5 |

**Last Update step :**



|  |  |  |
| --- | --- | --- |
|  | Objects | Centroides |
| Cluster 1 | {D} | U1=(0.91 , 1.09) |
| Cluster 2 | {G} | U2=(5.74 , 5.04) |
| Cluster 3 | {H} | U3= (3 , 3.47) |
| Cluster 4 | {F} | U4=(5.2 , 2.52) |
| Cluster 5 | {I,K} | U5=(4.83 , 3.48) |
| Cluster 6 | {C} | U6=(3.12 , 1.85) |
| Cluster 7 | {A,B} | U7=(1.78 , 0.85) |
| Cluster 8 | {E} | U8=(2.37 , 4.11) |

k=9

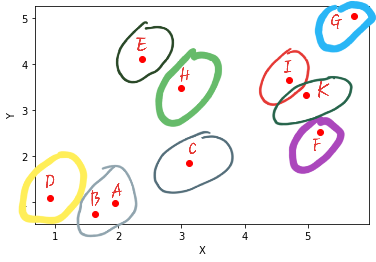
**Initialisation Step :**

|  |  |  |
| --- | --- | --- |
|  | Objects | Centroides |
| Cluster 1 | D | U1=(0.91 , 1.09) |
| Cluster 2 | G | U2=(5.74 , 5.04) |
| Cluster 3 | H | U3= (3 , 3.47) |
| Cluster 4 | F | U4=(5.2 , 2.52) |
| Cluster 5 | I | U5=(4.7 , 3.65) |
| Cluster 6 | C | U6=(3.12 , 1.85) |
| Cluster 7 | A | U7=(1.95 , 0.97) |
| Cluster 8 | E | U8=(2.37 , 4.11) |
| Cluster 9 | K | U9=(4.97 , 3.32) |

**Last Assignement step :**

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Object | Dist(i,U1) | Dist(i,U2) | Dist(i,U3) | Dist(i,U4) | Dist(i,U5) | Dist(i,U6) | Dist(i,U7) | Dist(i,U8) | Dist(i,U9) | Relocation |
| A | 1.16 | 7.86 | 3.55 | 4.8 | 5.43 | 2.05 | 0.29 | 3.56 | 5.37 | U7 |
| B | 1.06 | 8.42 | 4.11 | 5.36 | 5.99 | 2.61 | 0.27 | 4.12 | 5.93 | U7 |
| C | 2.97 | 5.81 | 1.74 | 2.75 | 3.38 | 0.0 | 2.34 | 3.01 | 3.32 | U6 |
| D | 0.0 | 8.78 | 4.47 | 5.72 | 6.35 | 2.97 | 1.11 | 4.48 | 6.29 | U1 |
| E | 4.48 | 4.3 | 1.27 | 4.42 | 2.79 | 3.01 | 3.85 | 0.0 | 3.39 | U8 |
| F | 5.72 | 3.06 | 3.15 | 0.0 | 1.63 | 2.75 | 5.09 | 4.42 | 1.03 | U4 |
| G | 8.78 | 0.0 | 4.31 | 3.06 | 2.43 | 5.81 | 8.15 | 4.3 | 2.49 | U2 |
| H | 4.47 | 4.31 | 0.0 | 3.15 | 1.88 | 1.74 | 3.84 | 1.27 | 2.12 | U3 |
| I | 6.35 | 2.43 | 1.88 | 1.63 | 0.0 | 3.38 | 5.72 | 2.79 | 0.6 | U5 |
| K | 6.29 | 2.49 | 2.12 | 1.03 | 0.6 | 3.32 | 5.67 | 3.39 | 0.0 | U9 |

**Last Update step :**



|  |  |  |
| --- | --- | --- |
|  | Objects | Centroides |
| Cluster 1 | {D} | U1=(0.91 , 1.09) |
| Cluster 2 | {G} | U2=(5.74 , 5.04) |
| Cluster 3 | {H} | U3= (3 , 3.47) |
| Cluster 4 | {F} | U4=(5.2 , 2.52) |
| Cluster 5 | {I} | U5=(4.7 , 3.65) |
| Cluster 6 | {C} | U6=(3.12 , 1.85) |
| Cluster 7 | {A,B} | U7=(1.78 , 0.85) |
| Cluster 8 | {E} | U8=(2.37 , 4.11) |
| Cluster 9 | {K} | U9=(4.97 , 3.32) |

K=10

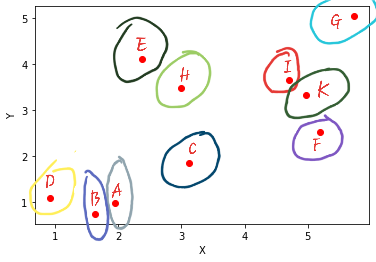
**Initialisation Step :**

|  |  |  |
| --- | --- | --- |
|  | Objects | Centroides |
| Cluster 1 | D | U1=(0.91 , 1.09) |
| Cluster 2 | G | U2=(5.74 , 5.04) |
| Cluster 3 | H | U3= (3 , 3.47) |
| Cluster 4 | F | U4=(5.2 , 2.52) |
| Cluster 5 | I | U5=(4.7 , 3.65) |
| Cluster 6 | C | U6=(3.12 , 1.85) |
| Cluster 7 | A | U7=(1.95 , 0.97) |
| Cluster 8 | E | U8=(2.37 , 4.11) |
| Cluster 9 | K | U9=(4.97 , 3.32) |
| Cluster 10 | B | U10=(1.62 , 0.74) |

**Last Assignement step :**

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Object | Dist(i,U1) | Dist(i,U2) | Dist(i,U3) | Dist(i,U4) | Dist(i,U5) | Dist(i,U6) | Dist(i,U7) | Dist(i,U8) | Dist(i,U9) | Dist(i,U10) | Relocation |
| A | 1.16 | 7.86 | 3.55 | 4.8 | 5.43 | 2.05 | 0.0 | 3.56 | 5.37 | 0.56 | U7 |
| B | 1.06 | 8.42 | 4.11 | 5.36 | 5.99 | 2.61 | 0.56 | 4.12 | 5.93 | 0.0 | U10 |
| C | 2.97 | 5.81 | 1.74 | 2.75 | 3.38 | 0.0 | 2.05 | 3.01 | 3.32 | 2.61 | U6 |
| D | 0.0 | 8.78 | 4.47 | 5.72 | 6.35 | 2.97 | 1.16 | 4.48 | 6.29 | 1.06 | U1 |
| E | 4.48 | 4.3 | 1.27 | 4.42 | 2.79 | 3.01 | 3.56 | 0.0 | 3.39 | 4.12 | U8 |
| F | 5.72 | 3.06 | 3.15 | 0.0 | 1.63 | 2.75 | 4.8 | 4.42 | 1.03 | 5.36 | U4 |
| G | 8.78 | 0.0 | 4.31 | 3.06 | 2.43 | 5.81 | 7.86 | 4.3 | 2.49 | 8.42 | U2 |
| H | 4.47 | 4.31 | 0.0 | 3.15 | 1.88 | 1.74 | 3.55 | 1.27 | 2.12 | 4.11 | U3 |
| I | 6.35 | 2.43 | 1.88 | 1.63 | 0.0 | 3.38 | 5.43 | 2.79 | 0.6 | 5.99 | U5 |
| K | 6.29 | 2.49 | 2.12 | 1.03 | 0.6 | 3.32 | 5.37 | 3.39 | 0.0 | 5.93 | U9 |

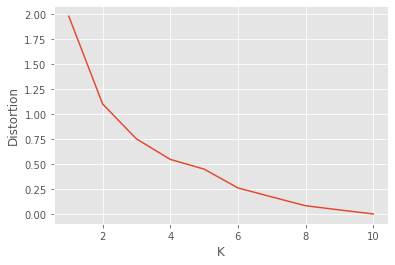
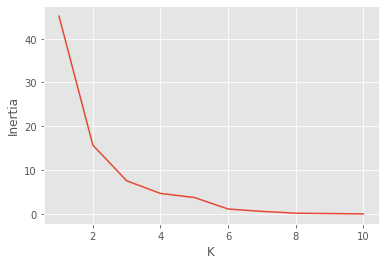
**Last Update step :**

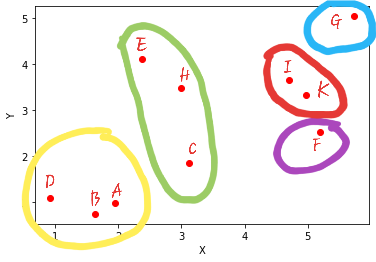


|  |  |  |
| --- | --- | --- |
|  | Objects | Centroïdes |
| Cluster 1 | {D} | U1=(0.91 , 1.09) |
| Cluster 2 | {G} | U2=(5.74 , 5.04) |
| Cluster 3 | {H} | U3= (3 , 3.47) |
| Cluster 4 | {F} | U4=(5.2 , 2.52) |
| Cluster 5 | {I} | U5=(4.7 , 3.65) |
| Cluster 6 | {C} | U6=(3.12 , 1.85) |
| Cluster 7 | {A} | U7=(1.95 , 0.97) |
| Cluster 8 | {E} | U8=(2.37 , 4.11) |
| Cluster 9 | {K} | U9=(4.97 , 3.32) |
| Cluster 10 | {B} | U10=(1.62 , 0.74) |

Donc pour chaque valeur de K on calcule la **Distortion** et l’**Inertia** et on obtient le tableau ci-dessous

|  |  |  |
| --- | --- | --- |
| K= | Distortion | Inertia |
| 1 | 1.976 | 45.12 |
| 2 | 1.099 | 15.7 |
| 3 | 0.75 | 7.56 |
| 4 | 0.545 | 4.675 |
| 5 | 0.448 | 3.75 |
| 6 | 0.26 | 1.122 |
| 7 | 0.17 | 0.575 |
| 8 | 0.082 | 0.171 |
| 9 | 0.04 | 0.08 |
| 10 | 0 | 0 |



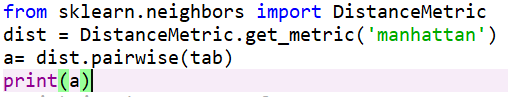
  
La meilleure valeur de k est 5. Ainsi,

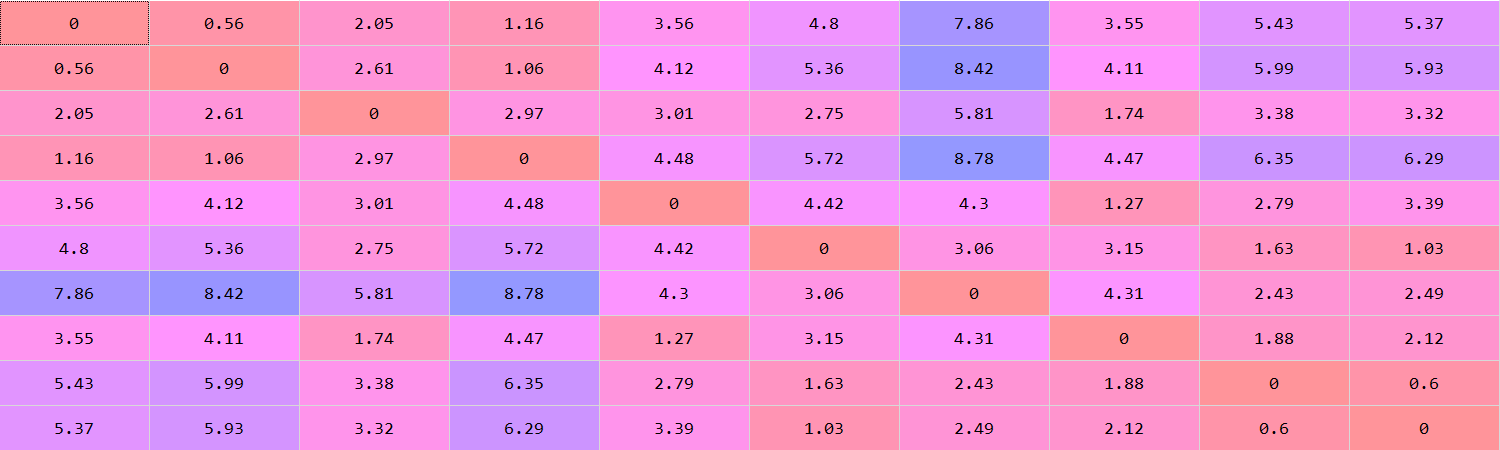
nous aurons cinq clusters regroupant

les objets comme suit :

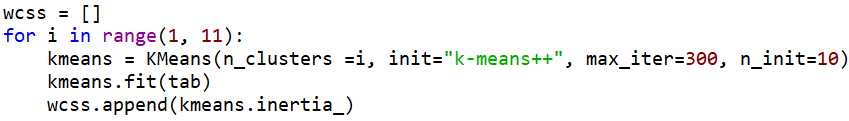
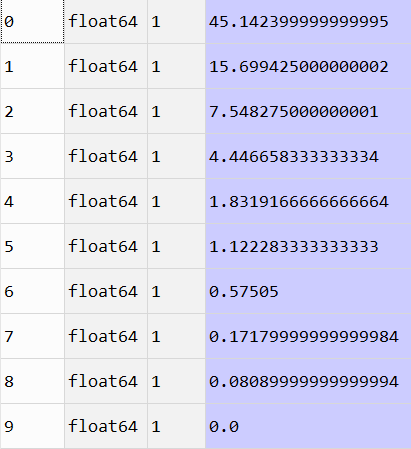
**Partie TP**

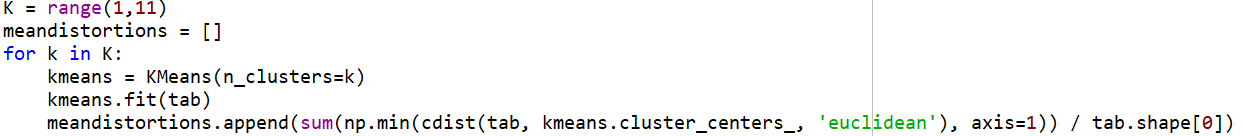
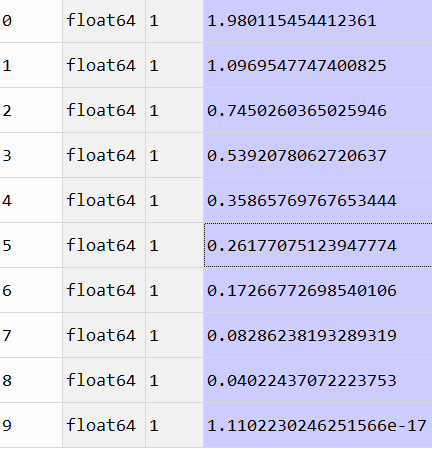
La matrice des distances en utilisant la métrique distance de Manhattan voici le code correspondant et les résultats obtenues, nous avons utilisé la bibliothèque *sklearn*:

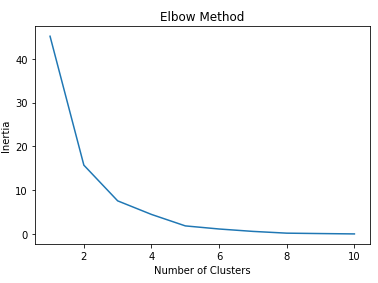


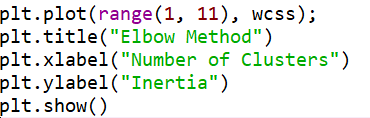


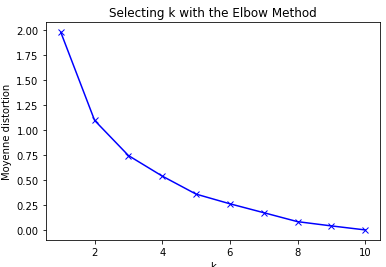
Calcule de la distorsion et l’inertie pour chaque K [1..10] :

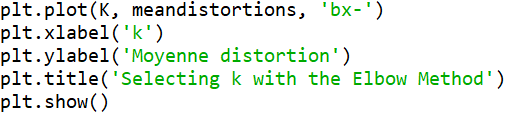
**L’inertie :**

**La distorstion :**

**La méthode d'Elbow :**

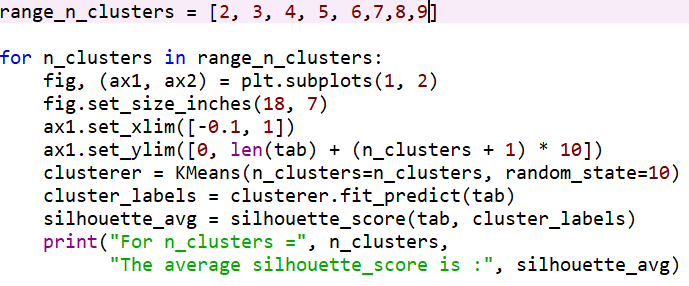
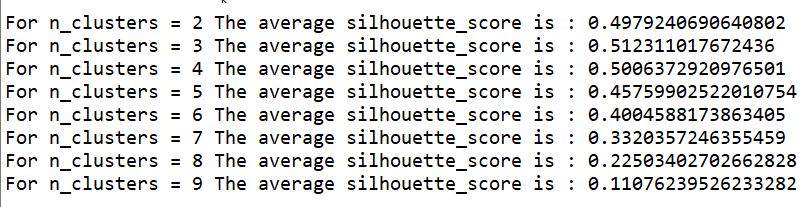
****

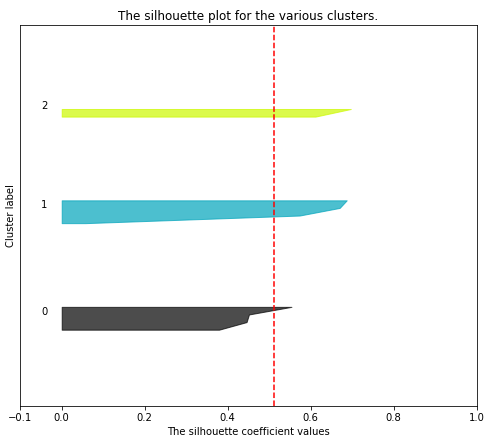
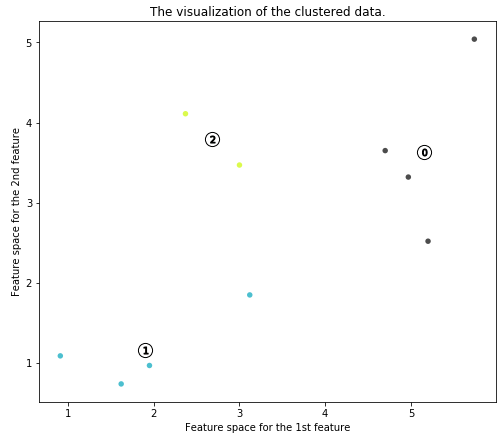
****

****

Comme on peut le voir dans le graphe et le tableau, lorsque k=5 les valeurs de l’inertie commencent à se stagner. On en conclue donc, le nombre de clusters correspondant est 5 avec la méthode K-means.

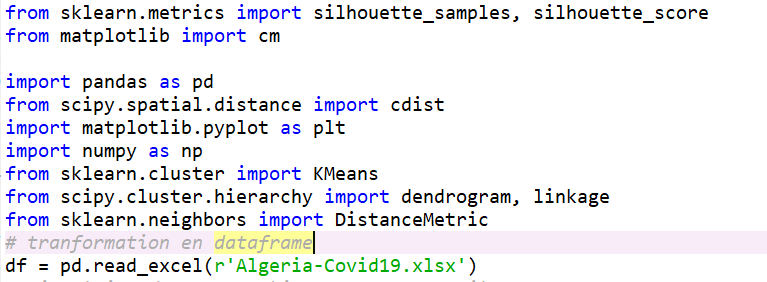
**La méthode silhouette :**

Les valeurs des coefficients de silhouette sont pour différents K = [2,9] :

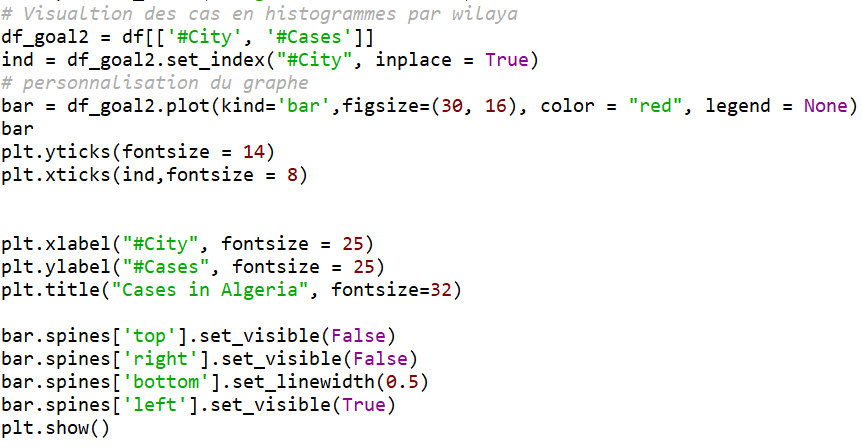
On constate que les valeurs de silhouette score se rapproche à 1 qui veut dire que chaque points est dans le bon cluster. Le nombre de cluster correspondant en utilisant cette méthode est k=3.

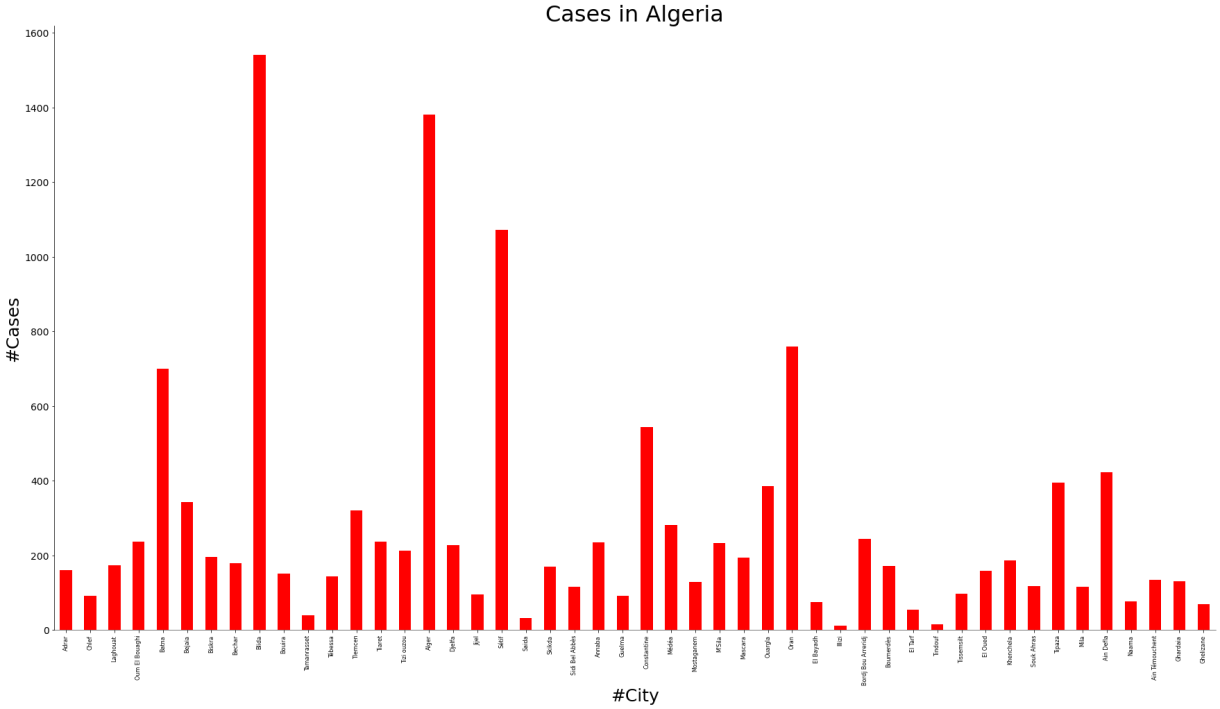
**Partie 2**

Dans cette partie nous allons utiliser une autre dataset qui est celle « Algeria-Covid19.xlsx » mis à jour le 27/06/20 qui est constitué de 4 colonnes qui représentent respectivement les wilayas, les personnes atteintes du virus, les rétablis et les morts et 48 lignes qui représentent les wilayas. Tout d’abord nous allons l’utiliser en dataframe en utilisant la bibliothèque *pandas* :



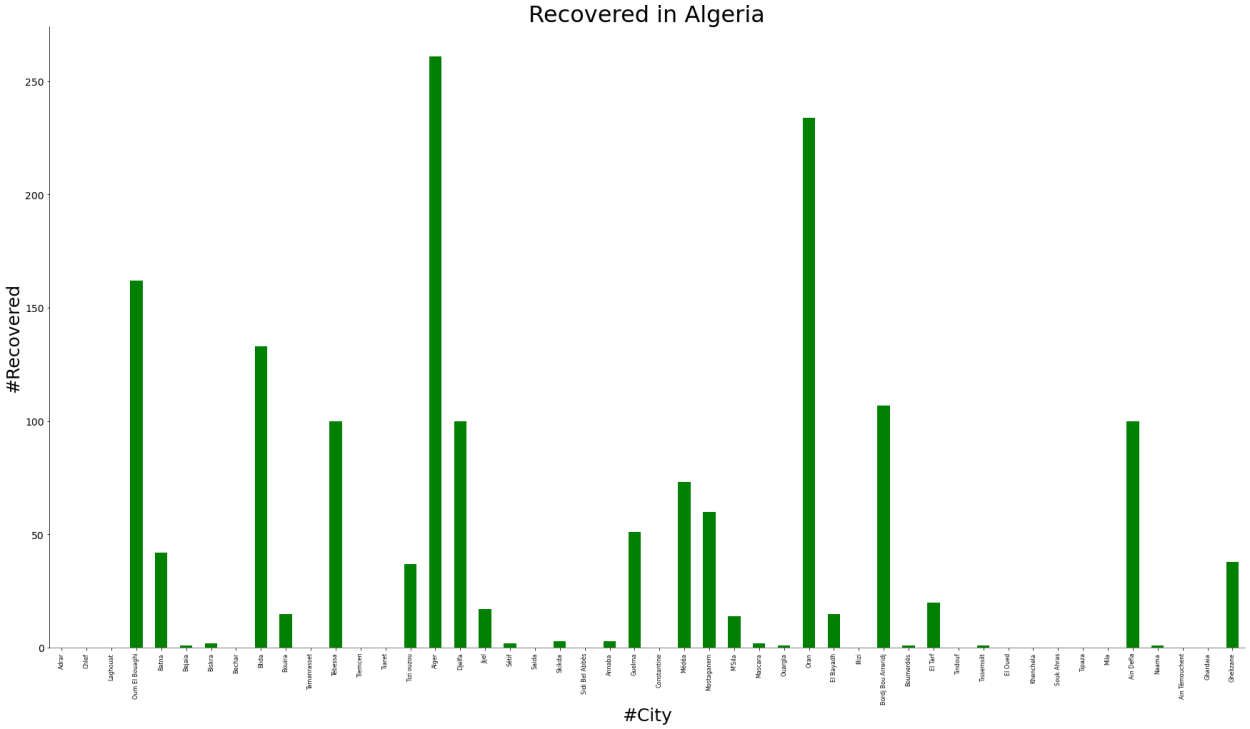
Après ça nous allons visualiser les données en histogrammes en utilisant la bibliothèque *matplotlib :*

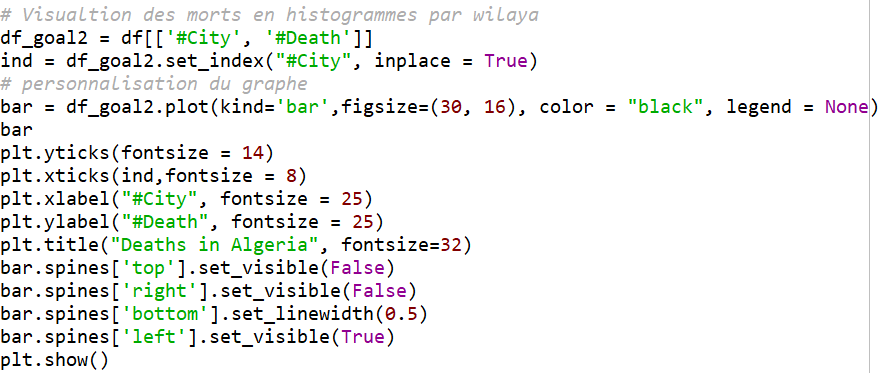
**Visualisation des cas par wilayas :**

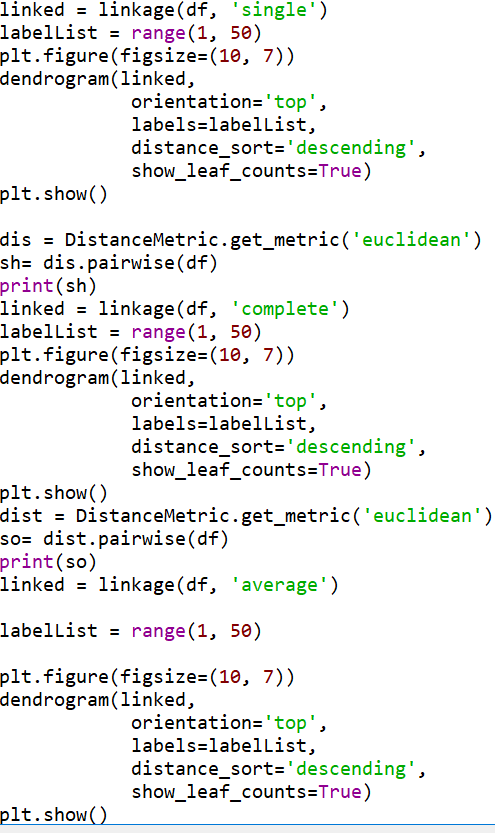
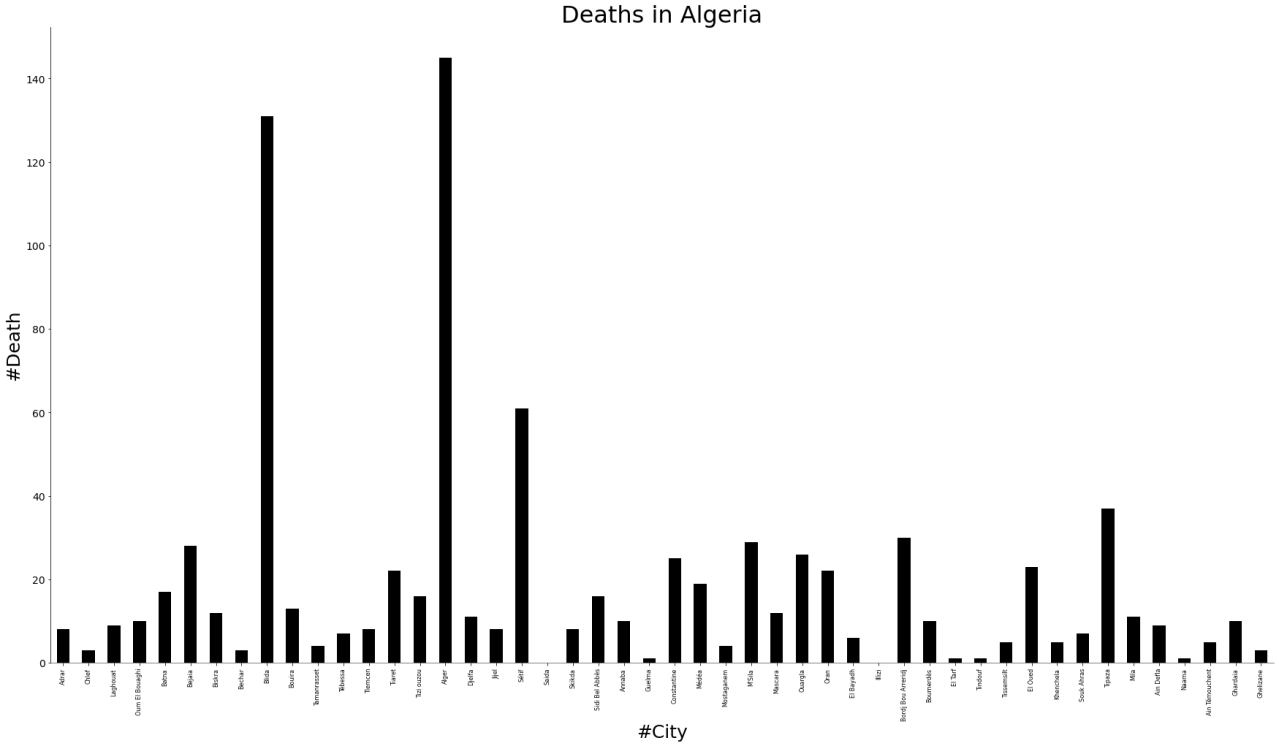


**Visualisation des rétablis par wilayas :**

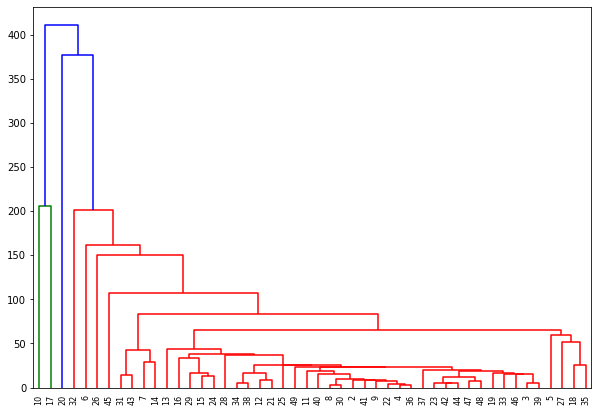
****

****

**Visualisation des morts par wilayas :**

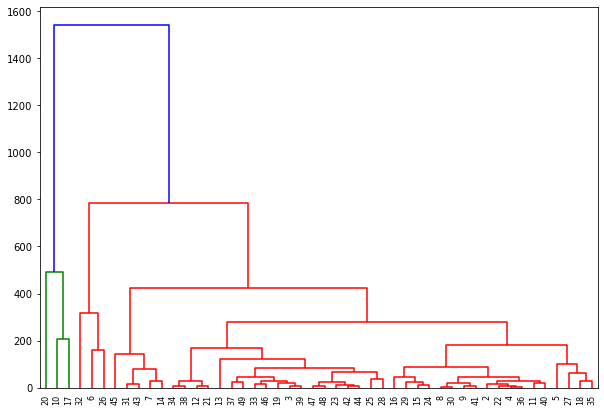
**Les dendrogrammes de chaque type de linkage :**

**CAH :**

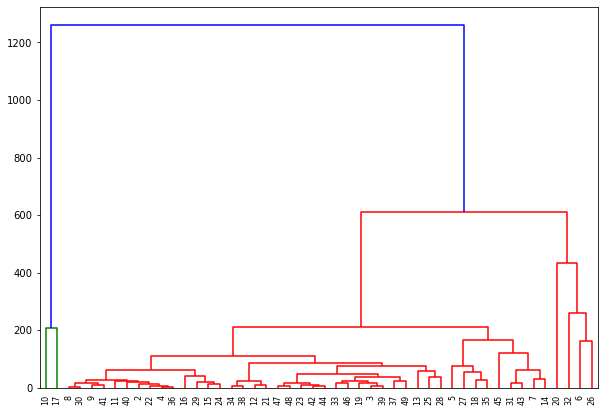
**Single linkage :**

Les chiffres de l’axe des abscisses représentent la numération de chaque wilayas.

**Complete linkage :**

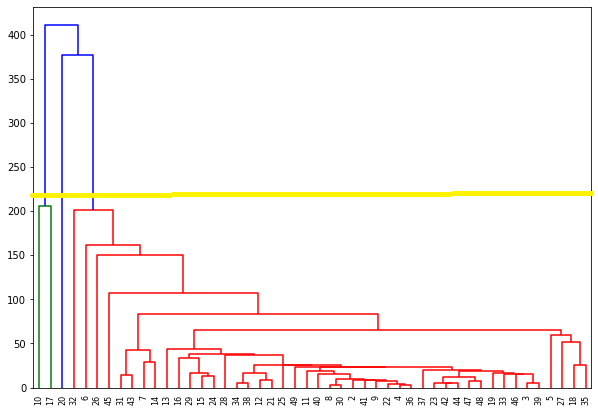
****

**Average linkage :**

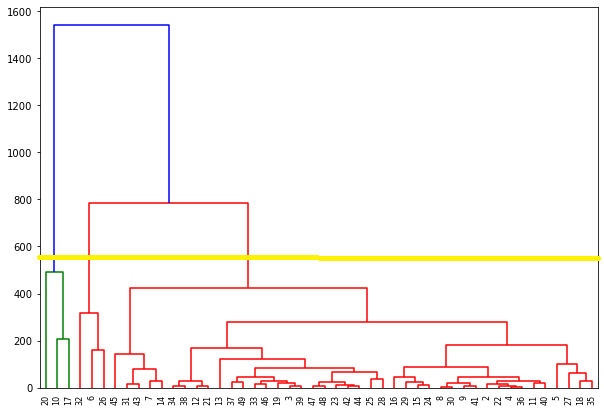
****

La meilleure coupe est

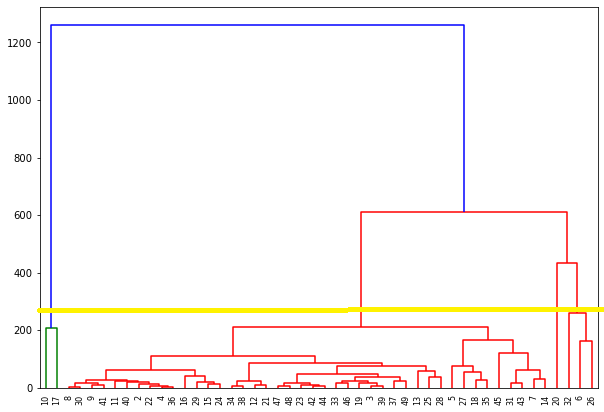
**Single linkage :**



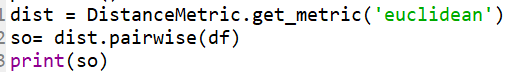
**Complete linkage :**

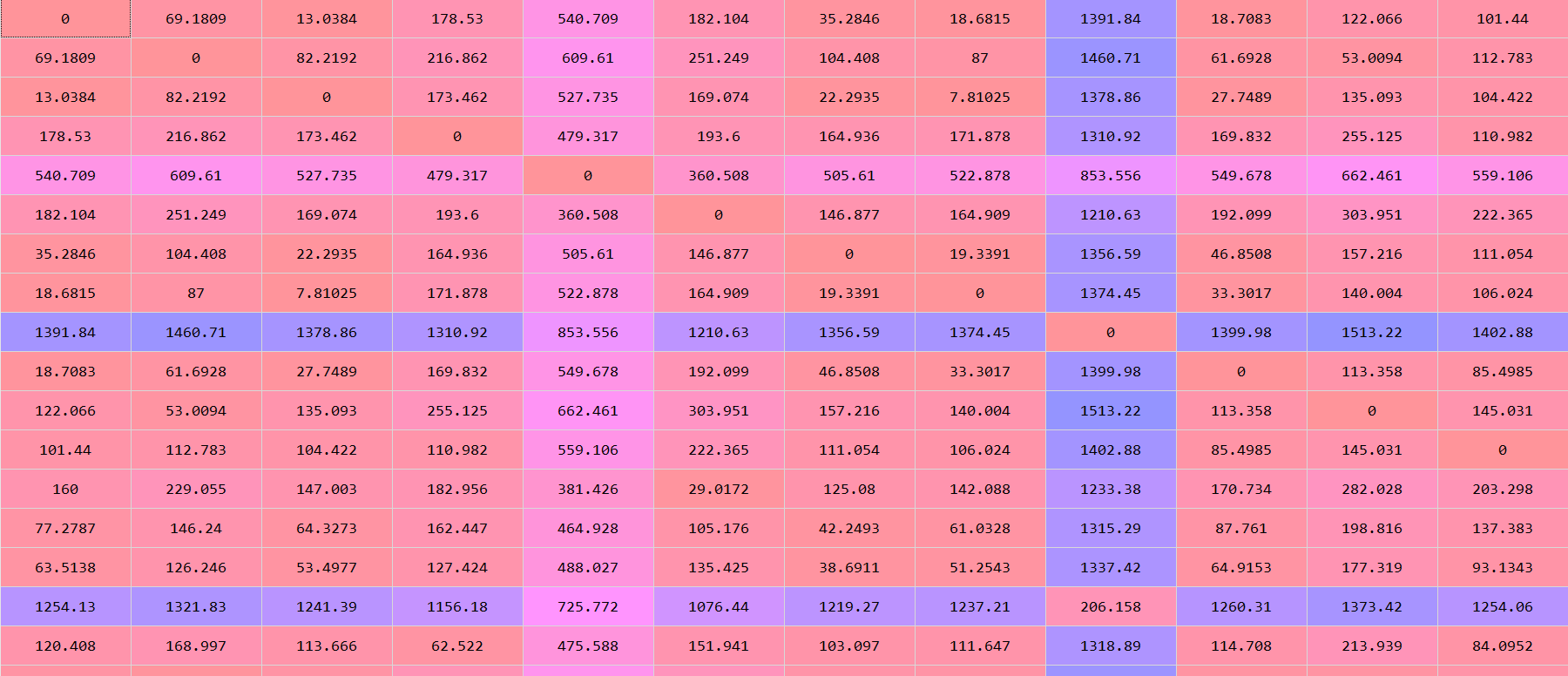


**Average linkage :**

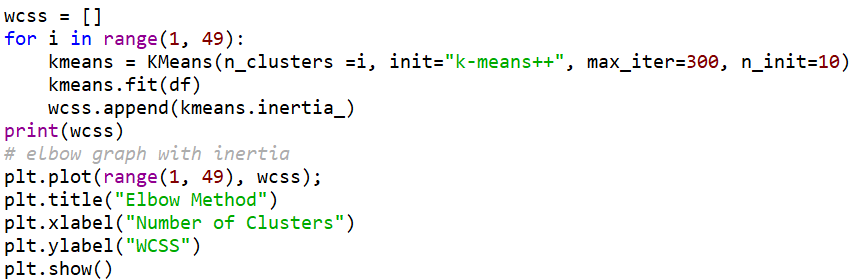


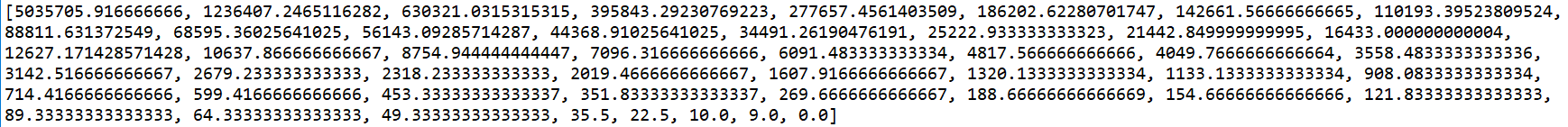
**Kmeans :**

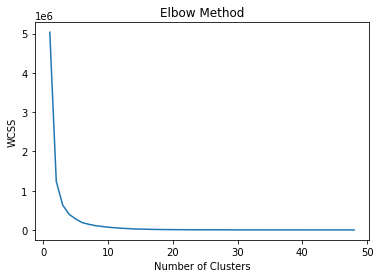
Calcul la matrice des distance avec distance Eucledienne :



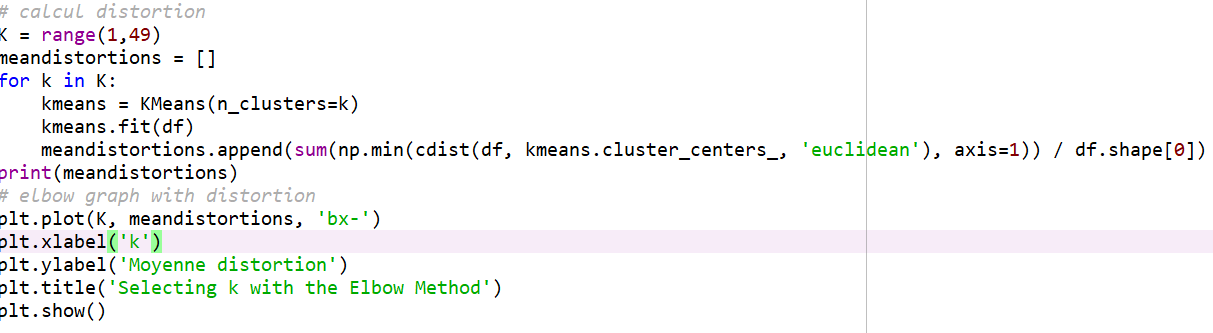
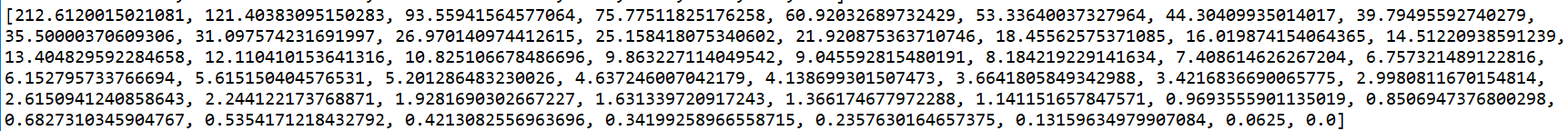
En exécutant le code vous trouverez l’intégralité de la matrice.

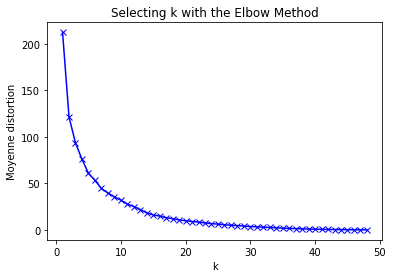
**Calcul de l’inertie et elbow et graphe :**



Comme on peut le constater avec le calcul de l’inertie et le elbow graphe le K le plus adéquat est 7.

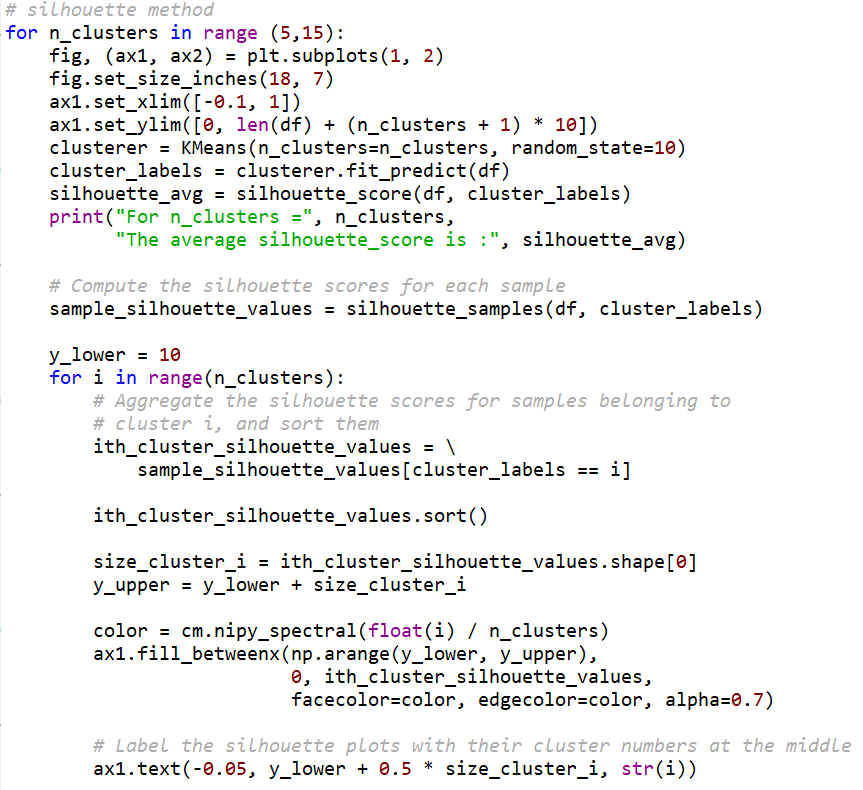
**Calcul de la distortion et elbow et graphe :**

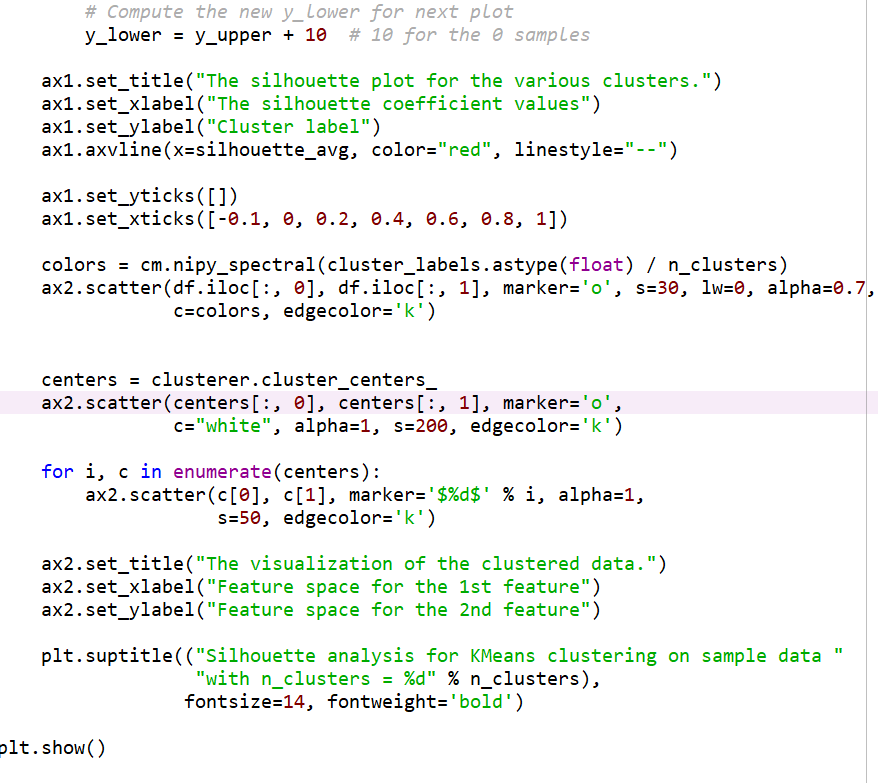


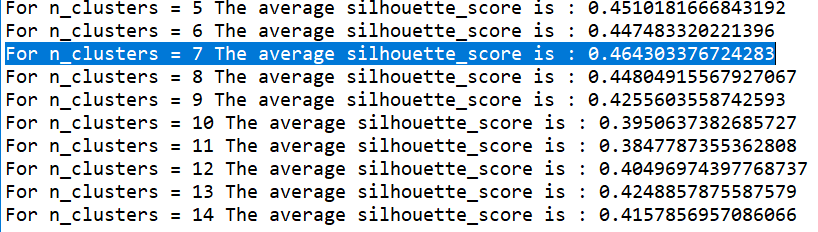


Comme on peut le constater avec le calcul de la distortion et le elbow graphe le K le plus adéquat est 7.

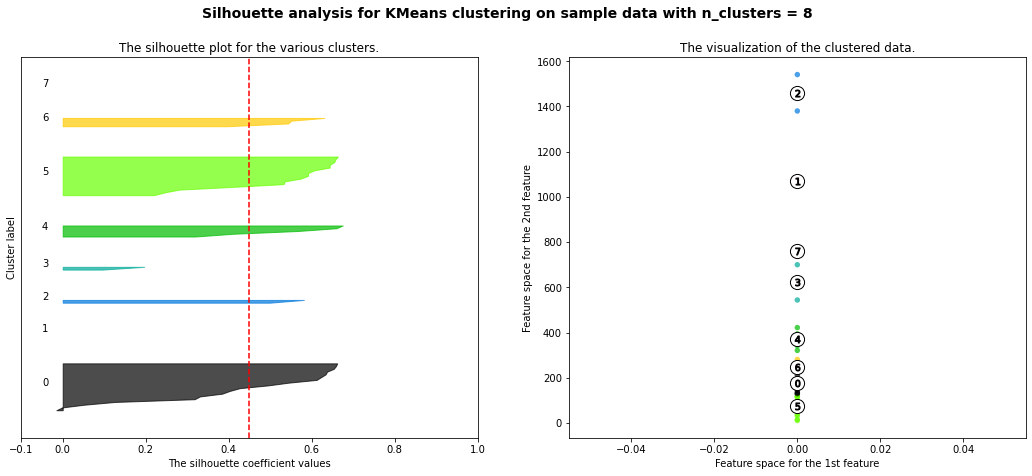
**La méthode silhouette :**

****

****

****

La valeur du coefficient silhouette la plus de 1 est nombre de clusters est 7s. C’est-à-dire k=7.

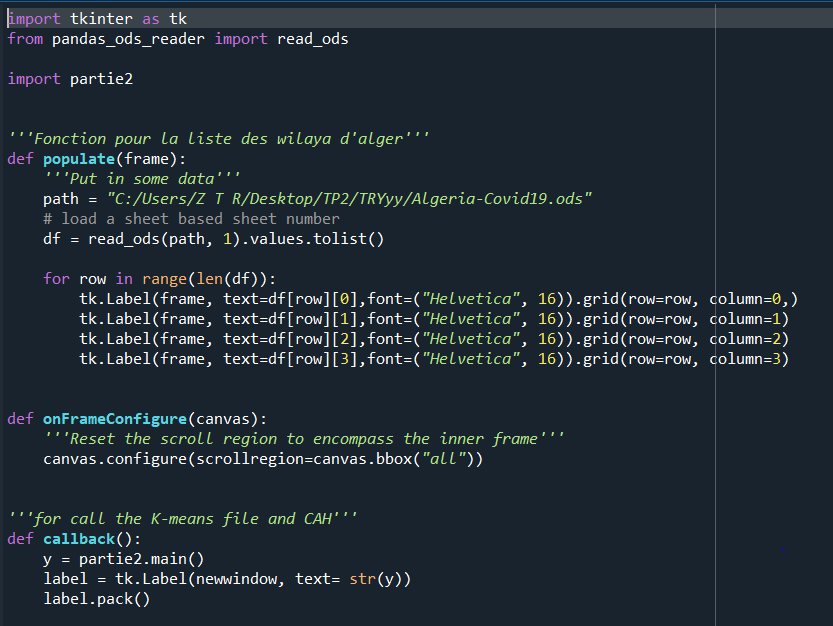
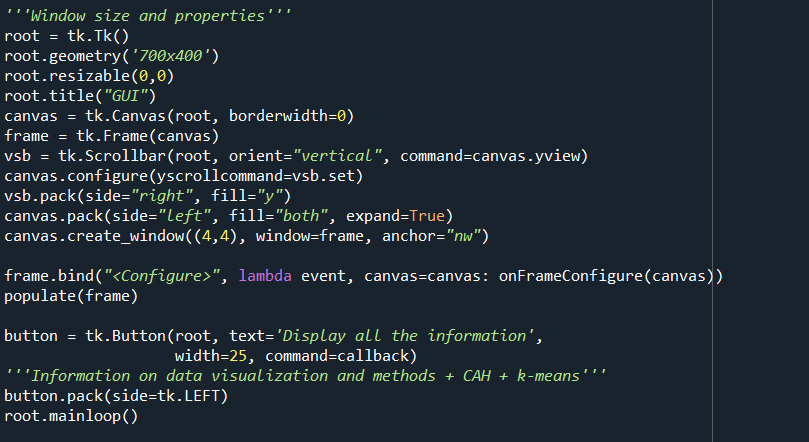


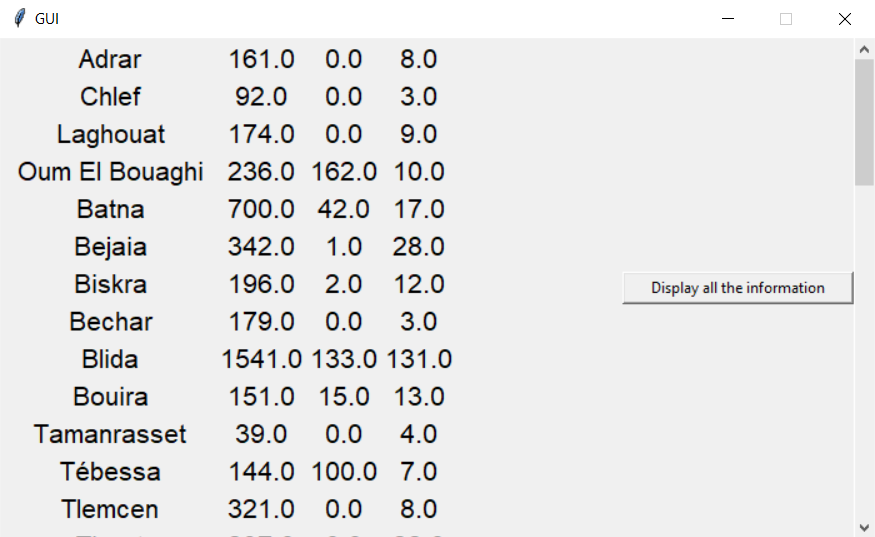
**L'interface graphique (GUI) :**

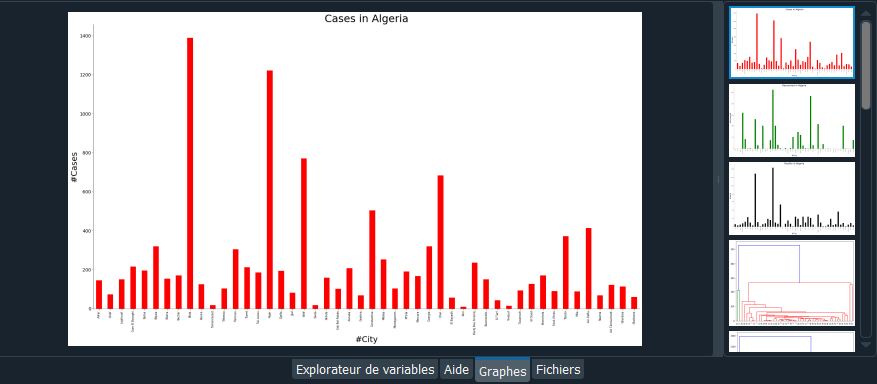
On veut récupérer les données à partir de l'Excel, puis on les affiche dans une fenêtre et on fait un scroll pour défiler toutes ces informations.

Un bouton qui nous permet d’afficher l’ensemble des résultats des deux fonctionnalités K-Means et CAH.

**Le code :**



Dès l’exécution de ce code, la fenêtre affichant les données apparait représentée par l’image ci-dessous :

En cliquant sur le bouton “Display all the information", les résultats s’affichent comme suit :

**Labellisation :**

|  |  |
| --- | --- |
| Features | Labels |
| #City et #Cases | #Deaths et #Recovered |

**Comparaison :**

|  |  |  |
| --- | --- | --- |
| Comparaison | CAH | K-means |
| Précision | Cette méthode n’est pas précise car elle généralise les cas par wilayas il n’a pas assez de clusters pour le nombre de wilaya. | Cette méthode est précise car elle prend plus en considération chaque cas c’est pour cela qu’on trouve plus de clusters. Et grâce aux graphes (elbow,silhouette), l’information du nombre de clusters est plus claire. |
| Résultats | Nombre de clusters K=3 | Nombre de cluster K=7. |

**Enquête :**

La suppression de l’attribut #Recovered n’est pas un choix intelligent car si on le supprime on ne pourra pas connaître la gravité du virus ni son ampleur, ni quelles mesures de sécurité et sanitaire prendre alors on ne pourra pas prédire l’évolution de cette pandémie ni quelles précautions prendre contre elle. Certes, le nombre de #Recovered n’est pas la soustraction entre #Cases et #Deaths mais cet attribut ne peut être supprimé et est important pour la prédiction. Ces calculs de soustraction on ne l’est aura pas réellement jusqu’à la fin totale de cette pandémie car tous les jours les chiffres changent constamment.