

SAÉ 15 - Traitement des données

TP 1 - Recensement



Exercice 1 : Problème ouvert

On entend souvent dire que la population d'Auxerre et/ou de l'Yonne est en diminution.

À l'aide du script [TP1.py](#), nous avons traité les données des trois fichiers CSV pour considérer la population d'Auxerre seule, puis de l'agglomération immédiate (Appoigny, Auxerre, Monéteau, Perrigny, Saint-Georges-sur-Baulche), et enfin de l'agglomération totale (Appoigny, Augy, Auxerre, Bleigny-le-Carreau, Branches, Champs-sur-Yonne, Charbuy, Chevannes, Chitry, Coulanges-la-Vineuse, Escamps, Escolives Sainte-Camille, Gurgy, Gy-l'Evêque, Irancy, Jussy, Lindry, Monéteau, Montigny-la-Resle, Perrigny, Quenne, Saint-Bris-le-Vineux, Saint-Georges-sur-Baulche, Vallan, Venoy, Villefargeau, Villeneuve - Saint-Salves, Vincelles, Vincelottes) au entre 2008 et 2021.

```
15 - TD2.py x SAE15 - TP1 - Recensement.py x
import csv
from pylab import *

def importation(fichier,separateur):
    liste=[]
    with open(fichier,newline='',encoding="UTF-8") as csvfile:
        reader=csv.reader(csvfile,delimiter=separateur)
        for row in reader:
            liste.append(row)
    return liste

#permet d'importer un csv dans une liste en précisant le fichier et le séparateur

donnees_2008 = importation("donnees_2008.csv",",")
donnees_2016 = importation("donnees_2016.csv",",")
donnees_2021 = importation("donnees_2021.csv",",")
metadonnees_communes = importation("metadonnees_communes.csv",";")
#On a maintenant 4 listes qui contiennent toutes les infos des fichiers
```

Figure 1 : Première partie du programme

On commence par créer un sous programme qui nous permet d'importer tous les fichiers dont on se servira plus tard.

Il prend en entrée le nom du fichier et le séparateur, en effet, tous les fichiers n'ont pas le même. Il donne en sortie une liste de liste qui contient chaque ligne des csv dans une liste.

```

18 def selec_region(donnees, annee, code_dep, meta, agglo):
19     liste=[]
20     if annee < 2020 :
21         for ligne in donnees :
22             if ligne[2].isdigit() and int(ligne[2]) == code_dep and ligne[6] in agglo:
23                 liste.append([ligne[6],int(ligne[7]),int(ligne[8]),int(ligne[9])])
24             else :
25                 for ligne in donnees :
26                     if ligne[1].isdigit() and int(ligne[1]) == code_dep:
27                         for communes in meta :
28                             if ligne[2] == communes[2] and communes[3] in agglo:
29                                 liste.append([communes[3],int(ligne[3]),int(ligne[4]),int(ligne[5])])
30     return liste
31
32 agglo_total=["Appoigny", "Augy", "Auxerre", "Bleigny-le-Carreau", "Branches", "Champs-sur-Yonne", "Charbuy", "Chevannes", "Chitry", "Coulanges-la-Vineuse"]
33 agglo_imm=["Appoigny","Auxerre", "Monéteau", "Perrigny", "Saint-Georges-sur-Baulche"]
34
35 donnees_2008=selec_region(donnees_2008, 2008,89, metadonnees_communes,agglo_total)
36 donnees_2016=selec_region(donnees_2016, 2016,89, metadonnees_communes,agglo_total)
37 donnees_2021=selec_region(donnees_2021, 2021,89, metadonnees_communes,agglo_total)
38 #On a maintenant 3 listes qui contiennent dans l'ordre : nom de la commune, population municipale, population comptée à part et population totale
39

```

Figure 2 : Fonction filtrant les données des fichiers CSV

On crée une seconde fonction, qui permettrait de ne garder que les informations utiles pour la suite.

Elle prend en entrée la liste créée précédemment, l'année, le département d'étude, la liste des métadonnées et la liste des communes de l'agglomération totale. Elle renvoie en sortie une liste des 29 communes de l'agglomérations totale avec dans l'ordre leur nom, leur population municipale, leur population comptée à part et leur population totale.

On a donc 3 listes, une par année, contenant les informations des communes de l'agglomération totale.

```

donnees=[]
for i in range(29):
    donnees.append([donnees_2008[i][0],donnees_2008[i][3],donnees_2016[i][3],donnees_2021[i][3]])

donnees_agglo=[0,0,0]
for commune in donnees:
    for annee in range(3):
        donnees_agglo[annee]=donnees_agglo[annee]+commune[annee+1]

donnees_agglo_imm=[0,0,0]
for commune in donnees:
    for annee in range(3):
        if commune[0] in aggro_imm:
            donnees_agglo_imm[annee]=donnees_agglo_imm[annee]+commune[annee+1]

donnees_aux=[0,0,0]
for commune in donnees:
    for annee in range(3):
        if commune[0] == "Auxerre":
            donnees_aux[annee]=commune[annee+1]

```

Figure 3 : création des tableaux de données

On peut maintenant commencer l'étude des données.

La liste donnees contient dans l'ordre le nom de la commune, sa population en 2008, sa population en 2016 et sa population en 2021.

Les listes donnees suivantes contiennent pour chaque localisation précisé dans leur nom 3 chiffres : la population en 2008, celle en 2016 et celle en 2021.

```

61 annees=[2008,2016,2021]
62 figure("Evolution de la population")
63 plot(annees,donnees_agglo,label="Agglomération totale")
64 plot(annees,donnees_agglo_imm,label="Agglomération immédiate")
65 plot(annees,donnees_aux,label="Auxerre")
66 plt.legend()
67 show()
68

```

Figure 4 : Affichage de l'évolution de la population

On crée la liste des années qui représente les abscisses des courbes.

On peut maintenant afficher 3 courbes dans un tableau pour représenter l'évolution des populations globales.

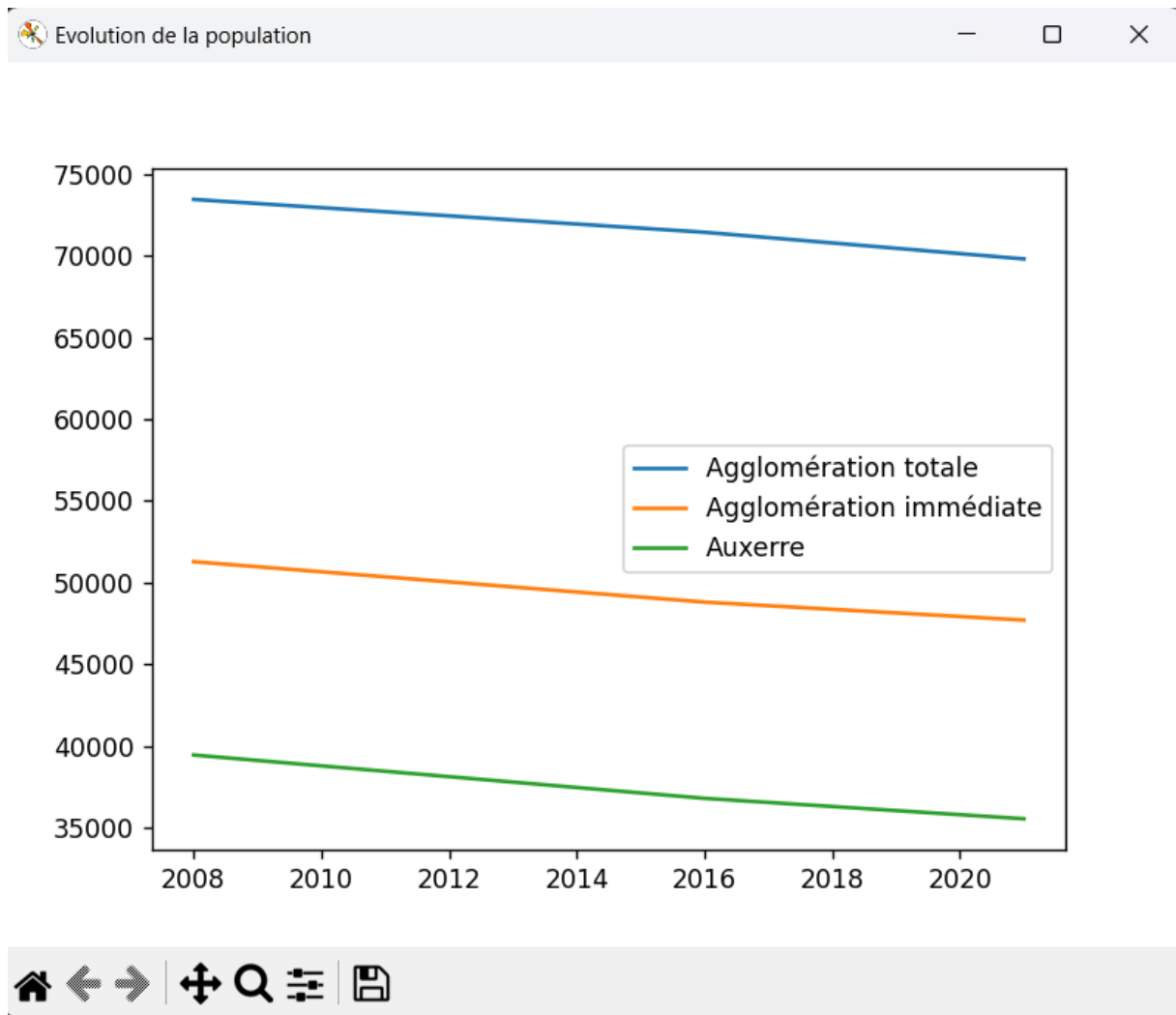


Figure 5 : Courbes de l'évolution de la population

On obtient ces trois courbes, on observe une population en baisse à tous les niveaux

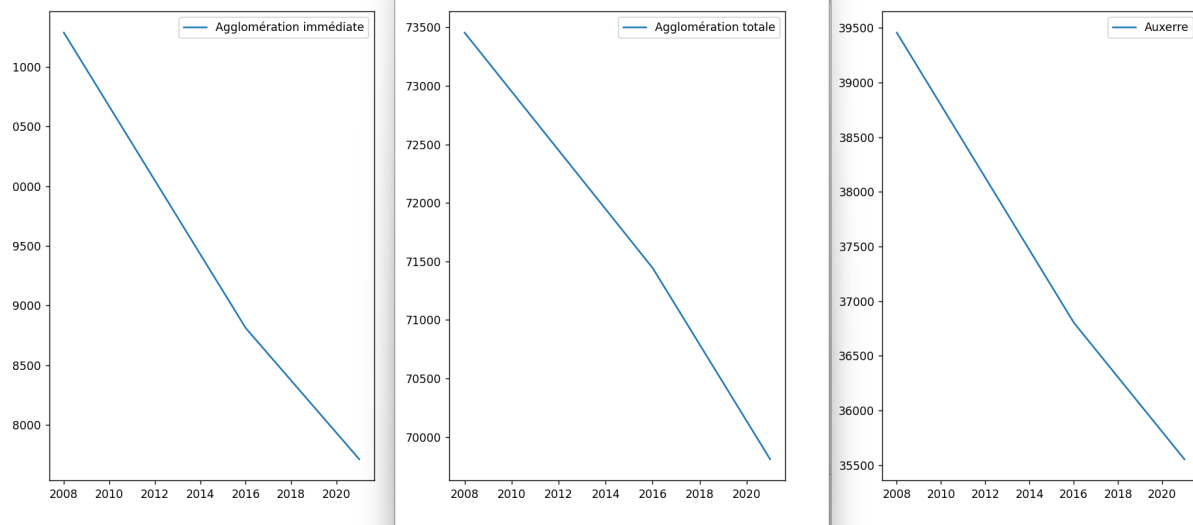


Figure 6 : Affichage des 3 courbes séparément

On peut également les afficher séparément pour voir si la baisse de population s'est accélérée ou a ralenti. On voit qu'elle a ralenti sur Auxerre et sur l'agglomération immédiate mais qu'elle accélère sur l'agglomération totale.