

TP : Importer des données à partir d'un fichier CSV

Un fichier csv (Comma- Separated Values) est un fichier texte.
Nous allons traiter deux fichiers csv dans ce TP.

Partie I : températures

Le code suivant est le même que celui vu en cours.

Tester le tpCSV.py qui contient les lignes suivantes :

```
nom_fichier="ballonsonde.csv"
donnees=open(nom_fichier) # Ouverture du fichier

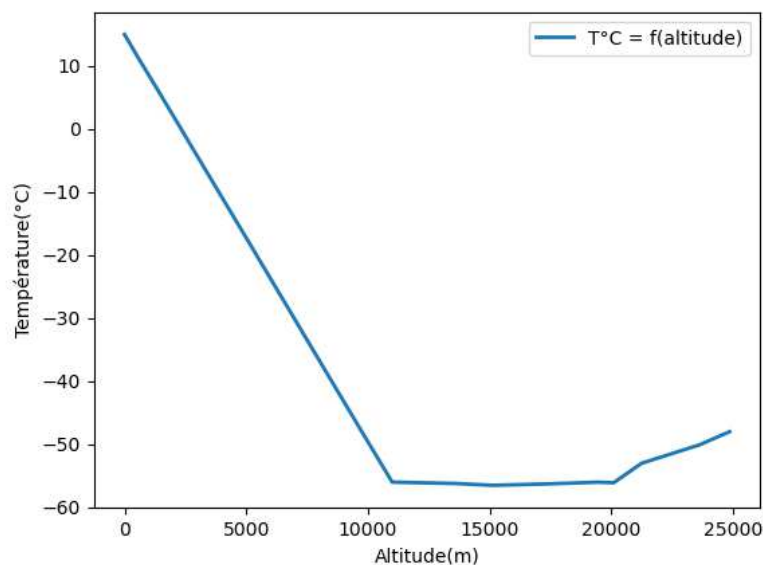
# Création d'une liste avec les entêtes des deux colonnes
# rstrip() supprime les retours à la ligne \n et espaces vides s'il y en a
# split(";") sépare les différentes valeurs en utilisant ";" comme séparateur, le résultat est une liste

# Création d'une liste (table) avec les lignes de valeurs du csv
tabl=[] # Création d'une liste vide
lignes=donnees.readlines() # On transforme les données en liste

for ligne in lignes:
    ligne=ligne.rstrip().split(";")
    tabl.append(ligne) # Ajout de la liste (ligne) représentant
                        # une ligne du csv à table

donnees.close()
print (tabl)
```

Nous voulons, dans ce premier exercice, afficher le graphique suivant :



Nous allons utiliser pour cela le module pylab.
Le code pour obtenir ce graphique est le suivant :

```
from pylab import *

""" A FAIRE """

plot(valeurs_alt, valeurs_temp, "-", label="T°C = f(altitude)", linewidth=2)
xlabel(labelx)
ylabel(labely)
legend()
show() # affiche le graphique à l'ecran
```

valeurs_alt est une liste qui contient les valeurs d'altitude (première colonne du fichier).
valeurs_temp est une liste qui contient les valeurs de température (deuxième colonne du fichier).

labelx est le nom des données de la première colonne.
labely est le nom des données de la deuxième colonne.

Il faut donc extraire les informations du fichier afin de créer ces deux listes ainsi que labelx et labely.

Attention : les informations de ce fichier sont de type str et pylab attend des valeurs numériques pour tracer un graphique.

Partie II : population

Le fichier csv à utiliser est "Population.csv"

Programme 1 :

Ouvrir le fichier csv, en extraire les données dans une liste (nommée contenu_fichier) de listes comprenant chacune l'année, la tranche d'âge, le nombre d'individus.

Attention, les nombres d'individus et années devront être convertis en nombres entiers (par défaut ce sont des chaînes de caractères).

Résultat attendu :

```
liste_contenu_fichier : [[1975, '0-19 ans', 16888065], [1975, '20-39 ans', 14486874], [1975, '40-59 ans', 11552754], [1975, '60-74 ans', 7015885], [1975, '75 ans et +', 2656422], [1976, '0-19 ans', 16809211], [1976, '20-39 ans', 14678387], [1976, '40-59 ans', 11726623], [1976, '60-74 ans', 6861583], [1976, '75 ans et +', 2722534], [1977, '0-19 ans', 16703587], [1977, '20-39 ans', 14901863], [1977, '40-59 ans', 11989832], [1977, '60-74 ans', 6613226], [1977, '75 ans et +', 2810497], [1978, '0-19 ans', 16612734], [1978, '20-39 ans', 15132004], [1978, '40-59 ans', 12226703], [1978, '60-74 ans', 6613226], [1978, '75 ans et +', 2810497]]
```

Programme 2 :

Créer une fonction pop_o_19 qui, à partir de la liste générée au programme précédent, renvoie une liste (nommée population_o_19) de tuples contenant chacun l'année et la population de 0 à 19 ans.

Résultat attendu :

```
liste_population_0_19 : [(1975, 16888065), (1976, 16809211), (1977, 16703587), (1978, 16612734), (1979, 16510923), (1980, 16418623), (1981, 16379718), (1982, 16327004), (1983, 16302740), (1984, 16199216), (1985, 16091820), (1986, 15999391), (1987, 15919634), (1988, 15852690), (1989, 15793067), (1990, 15719647), (1991, 15605455), (1992, 15473515), (1993, 15330473), (1994, 15179775), (1995, 15084361), (1996, 15058227), (1997, 15055742), (1998, 15026922), (1999, 15017908), (2000, 15044092), (2001, 15062693), (2002, 15086076), (2003, 15112516), (2004, 15182339), (2005, 152112516)]
```

Programme 3 :

Créer une fonction `scinde_liste` qui permet de séparer cette liste en deux listes dont une contient les années et l'autre les nombres d'individus.

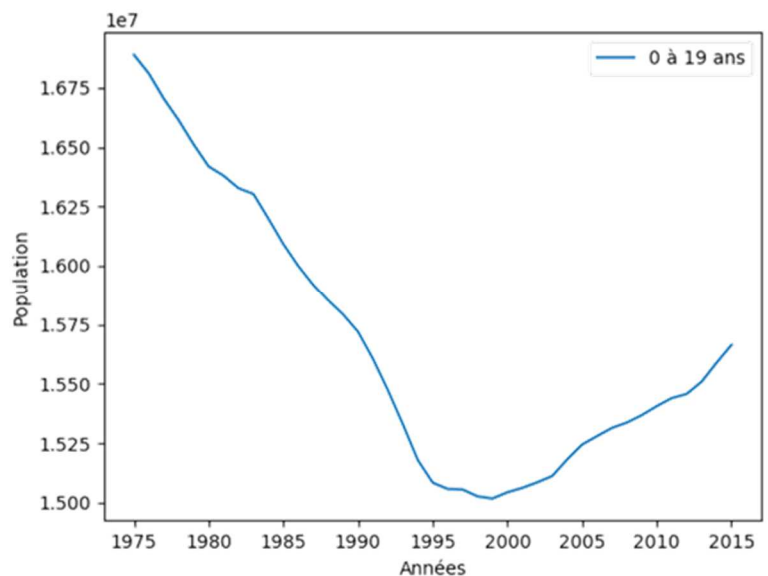
Résultat attendu :

liste des années : [1975, 1976, 1977, 1978, 1979, 1980, 1981, 1982, 1983, 1984, 1985, 1986, 1987, 1988, 1989, 1990, 1991, 1992, 1993, 1994, 1995, 1996, 1997, 1998, 1999, 2000, 2001, 2002, 2003, 2004, 2005, 2006, 2007, 2008, 2009, 2010, 2011, 2012, 2013, 2014, 2015]
 liste des nb d'individus : [16888065, 16809211, 16703587, 16612734, 16510923, 16418623, 16379718, 16327004, 16302740, 16199216, 16091820, 15999391, 15919634, 15852690, 15793067, 15719647, 15605455, 15473515, 15330473, 15179775, 15084361, 15058227, 15055742, 15026922, 15017908, 15044092, 15062693, 15086076, 15112516, 15182339, 15244989, 15280401, 15315094, 15337575, 15368840, 15406592, 15440408, 15457656, 15509305, 15589046, 15664305]

Programme 4 :

Réaliser une fonction (nommée `trace_graph`) qui génère un graphique à partir d'une des listes renvoyées par la fonction du programme 3.

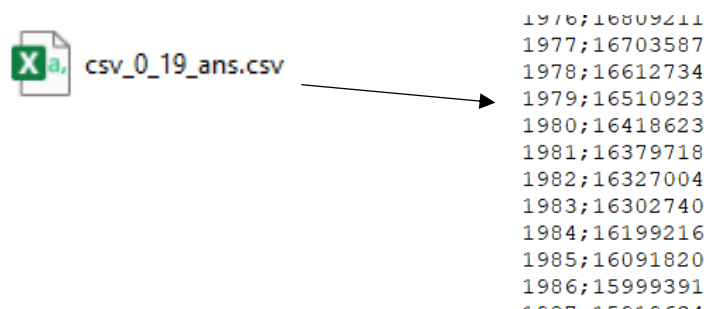
Résultat attendu :



Programme 5 :

Créer une fonction `genere_csv` qui crée un fichier csv à partir de la liste des 0 à 19 ans du programme 2.

Résultat attendu :



Le code pour créer le fichier csv est le suivant :

```
def genere_csv(liste,nom):
    """fonction qui génère un fichier csv à partir d'une liste de tuples à 2 éléments
    Il faut fournir un nom et une liste """
    fichier_csv=open(nom+'.csv','w')
    for element in liste:
        fichier_csv.write(str(element[0]))
        fichier_csv.write(";")
        fichier_csv.write(str(element[1]))
        fichier_csv.write("\n")
    fichier_csv.close()
```

Programme 6 :

Modifier la fonction `pop_o_19` du programme 2 et la nommer `pop_tranche_age` afin qu'elle renvoie 5 listes en fonction des tranches d'âges.

Résultat attendu :

liste population 0 à 19 : [(1975, 16888065), (1976, 16809211), (1977, 16703587), (1978, 16612734), (1979, 16510923), (1980, 16418623), (1981, 16379718), (1982, 16327004), (1983, 16302740), (1984, 16199216), (1985, 16091820), (1986, 15999391), (1987, 15919634), (1988, 15852690), (1989, 15793067), (1990, 15719647), (1991, 15605455), (1992, 15473515), (1993, 15330473), (1994, 15179775), (1995, 15084361), (1996, 15058227), (1997, 15055742), (1998, 15026922), (

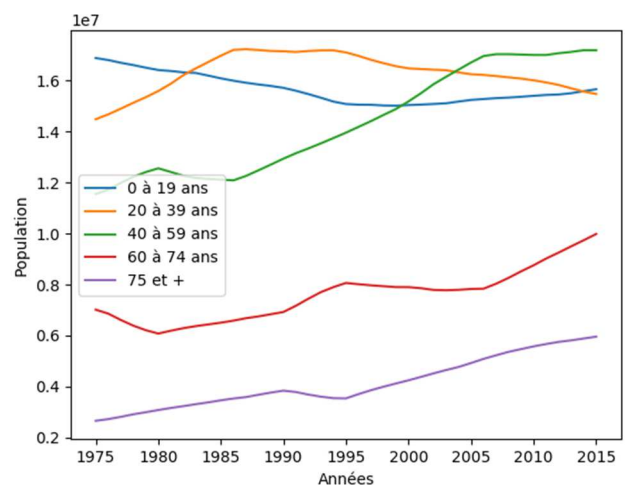
1999, 14984361), (2000, 14901863), (2001, 14809211), (2002, 14703587), (2003, 14612734), (2004, 14510923), (2005, 14418623), (2006, 14379718), (2007, 14327004), (2008, 14302740), (2009, 14199216), (2010, 14091820), (2011, 13999391), (2012, 13919634), (2013, 13852690), (2014, 13793067), (2015, 13719647), (1991, 17126616), (1992, 17163472), (1993, 17188587), (1994, 17190600), (1995, 17108296), (1996, 16973323), (1997, 16818685), (1998, 16687479),

Etc

Programme 7 :

Tracer sur le même graphique l'évolution des différentes tranches d'âges de la population.

Résultat attendu :



Programme 8 :

Créer une fonction `pop_totale` qui, à partir de la liste contenu_fichier générée au programme 1, renvoie une liste de tuples (nommée `population_totale`) contenant chacun l'année et la population totale de la France.

Résultat attendu :

liste population totale : [(1975, 52600000), (1976, 52798338), (1977, 53019005), (1978, 53271566), (1979, 53481073), (1980, 53731387), (1981, 54028630), (1982, 54335000), (1983, 54649984), (1984, 54894854), (1985, 55157303), (1986, 55411238), (1987, 55681780), (1988, 55966142), (1989, 56269810), (1990, 56577000), (1991, 56840661), (1992, 57110533), (1993, 57369161), (1994, 57565008), (1995, 57752535), (1996, 57935959), (1997, 58116018), (1998, 58298962), (

Programme 9 :

Ajouter au graphique du programme 7 l'évolution de la population totale française (de façon apparente avec un type de trait différent)

Résultat attendu :

