Thème : Traitement des données

19

Manipulation de fichiers CSV

Table des matières

- Ouverture d'un fichier CSV par des logiciels classiques
- Exploitation avec Python
 - Exercice n°1: Exploitation d'un fichier CSV en Python avec le module CSV
 - Exercice n°2 : les joueurs de rugby du TOP14
 - Extraction de données particulières
 - Exploitation graphique
 - Application
- Trier des données

Programme N.S.I

Contenus	Capacités attendues	Commentaires
Indexation de tables	Importer une table depuis un fichier texte tabulé ou un fichier CSV.	Est utilisé un tableau doublement indexé ou un tableau de p-uplets qui partagent les mêmes descripteurs.
Recherche dans une table	Rechercher les lignes d'une table vérifiant des critères exprimés en logique propositionnelle	La recherche de doublons, les tests de cohérence d'une table sont présentés

1. Exploitation avec Python

Les fichiers **CSV** (pour Comma Separated Values) sont des fichiers-texte (ils ne contiennent aucune mise en forme) utilisés pour stocker des données, séparées par des virgules (ou des points-virgules, ou des espaces...). Il n'y a pas de norme officielle du CSV.

2. Ouverture d'un fichier CSV par des logiciels classiques

- Ouvrir avec le Bloc-Notes le fichier Les colleges_Puy_De_dome_2022.csv.
- · Ouvrir le fichier avec LibreOffice.

2.1 Exercice n°1 : Exploitation d'un fichier CSV en Python avec le module CSV

L'utilisation d'un tableur peut être délicate lorsque le fichier CSV comporte un très grand nombre de lignes. Python permet de lire et d'extraire des informations d'un fichier CSV même très volumineux, grâce à des modules dédiés, comme le bien-nommé csv (utilisé ici) ou bien pandas (qui sera vu plus tard).

```
import csv
f = open('Les_colleges_Puy_De_Dome_2022.csv', "r", encoding = 'utf-8')
# 1
donnees = csv.reader(f) # 2
for ligne in donnees:
    print(ligne)
f.close() # 3
```

2.1.1 Problèmes:

- 1. Les données ne sont pas structurées : la première ligne est la ligne des «descripteurs» (ou des «champs»), alors que les lignes suivantes sont les valeurs de ces descripteurs.
- 2. La variable donnees n'est pas exploitable en l'état.

2.1.2 Améliorations

Au lieu d'utiliser la fonction csv.reader(), utilisons csv.DictReader(). Comme son nom l'indique, elle renverra une variable contenant des dictionnaires.

```
import csv
f = open('Les_colleges_Puy_De_Dome_2022.csv', "r", encoding = 'utf-8')
donnees = csv.DictReader(f)
```

```
for ligne in donnees:
    print(ligne)

f.close()
```

C'est mieux ! Les données sont maintenant des dictionnaires. Comment ré-accéder au premier d'entre eux ?

```
donnees[0]
```

Erreur normal, nous allons donc créer une liste de dictionnaires.

A titre d'information

```
import csv
global colleges
f = open('Les_colleges_Puy_De_Dome_2022.csv', "r", encoding = 'utf-8')
donnees = csv.DictReader(f)
colleges = []
for ligne in donnees:
    colleges.append(dict(ligne))

descripteurs=colleges[0] # 1
colleges.pop(0) # 2

f.close() # 3
```

- 1. on stocke les descripteurs dans une variable
- 2. on enlève les descripteurs
- 3. toujours fermer le fichier

descripteurs print(colleges[1]['Nom du Canton']) print(colleges[2]['Nom']) print(colleges[2]['Nom du (de la) Principal(e)'])

Question 1 :

Compléter la fonction plus_grand_college(colleges) qui renvoie la commune, le nom du collège et le nom du (de la) principal(e)

```
def plus_grand_college():
    maxi=0
    nom=''
    commune=''
    principal=''
    for col in ...:
        if int(...)>maxi:
            maxi=...
            nom=...
            commune=...
            principal=...
    return nom, commune, principal, maxi
```

? Question 2:

Ecrire une fonction <code>plus_petit_college(colleges)</code> qui renvoie la commune, le nom du collège et le nom du (de la) principal(e)

```
def plus_petit_college():
    mini=1E99
    nom=''
    commune=''
    principal=''
    pass
    return nom,commune,principal,mini

plus_petit_college()
```

? Question 3:

Ecrire une fonction moyenne () qui renvoie le nombre d'èléves moyen dans les collèges.

```
def moyenne():
    somme=0
    pass

round(moyenne(),1)
```

? Question 4:

Ecrire une fonction renseignement (college) qui renvoie les renseignements sur le collége choisi.

```
def renseignement(college):
    for ... in ...:
        if ...==...:
            return col
    return ...

renseignement('Collège I. et F. Joliot Curie')
```

? Question 5:

Ecrire une fonction renseignement (commune) qui renvoie les renseignements sur le(s) colléges de la commune ainsi que leur nombre.

```
def renseignement_commune(commune):
    rep=[]
    pass
    return nb, rep

renseignement_commune('ISSOIRE')
```

2.2 Exercice n°2: les joueurs de rugby du TOP14

Le fichier Top14.csv contient tous les joueurs du Top14 de rugby, avec :

- ID
- Nom
- · Âge, Taille, Poids
- Nationalité
- Poste
- Matchs
- E, T, P, D, J, R
- Temps de jeu -Club

Ce fichier a été généré sur le site https://www.allrugby.com/stats/moteur.html

? Question 1.

Stocker dans une variable joueurs les renseignements de tous les joueurs présents dans ce fichier csv.

```
import csv
f = open('data/Top14.csv', "r", encoding = 'utf-8')
donnees = csv.DictReader(f)
pass
f.close()
```

Question 2.

Combien de joueurs sont présents dans ce fichier ?

? Question 3.

Quel est le prénom du joueur n°486?

Question 4.

Ecrire une fonction permettant d'obtenir le numero d'un joueur connaissant son nom.

```
def numero(nom):
    pos=0
    pass
numero('SERIN Baptiste')
```

2.2.1 Extraction de données particulières

? Question 5.

Ecrire une fonction clubJoueur(joueur) permettant de savoir où joue un joueur? A appliquer avec Baptiste SERIN?

La méthode la plus naturelle est de parcourir toute la liste jusqu'à trouver le bon joueur, puis d'afficher son équipe :

Note

Une méthode plus efficace est d'utiliser une liste par compréhension incluant un test. Rappel :

```
exemple = [k for k in range(1,50) if k % 3 == 0]
print(exemple)
```

Procéder de même pour retrouver le club de Baptiste SERIN.

```
clubSerin = [k['Club'] for k in joueurs if k['Nom'] == 'SERIN
Baptiste']
print(clubSerin)
```

? Question 6.

Ecrire une fonction <code>joueurPoid(poid)</code> qui renvoie la liste des joueurs de plus de poid kg ?

A tester avec les joueurs de plus de 140 kg

Il est impératif de convertir la chaine de caractère renvoyée par k['Poids'] en entier, par la fonction int().

? Question 7.

Ecrire une fonction taille_moyenne(poste) renvoyant la taille moyenne des joueurs en fonction du poste.

Question 8.

Ecrire une fonction poid_moyenne(poste) renvoyant le poid moyen des joueurs en fonction du poste.

? Question 9.

Ecrire une fonction renvoyant la taille et le poid moyen des joueurs par équipe.

2.2.2 Exploitation graphique

Nous allons utiliser le module Matplotlib pour illustrer les données de notre fichier csv.

2.2.3 Exemple

```
import matplotlib.pyplot as plt
plt.close()
X = [0,1,3,6]
Y = [12,10,7,15]
plt.plot(X,Y,'ro') # (1)
plt.show()
```

1. r pour red, o pour un cercle. voir https://matplotlib.org/api/markers_api.html

2.2.4 Application

? Question 10.

Afficher sur un graphique tous les joueurs de rugby du top14, en mettant le poids en abscisse et la taille en ordonnée.

```
plt.close()
X=[]
Y=[]
for k in joueurs:
    pass
plt.plot(X,Y,'ro')
plt.show()
```

? Question 11.

Faire apparaître ensuite les joueurs évoluant au poste de Centre en bleu (b), les 2ème lignes en vert (g), les demis de Mêlée en magenta (m), les Piliers en jaune et les Arrières en cyan (c).

```
plt.close()
#tous les joueurs
X = [int(k['Poids']) for k in joueurs]
Y = [int(k['Taille']) for k in joueurs]
plt.plot(X,Y,'ro')
#on recolorie les Centres en bleu
X = [int(k['Poids']) for k in joueurs if k['Poste'] == 'Centre']
Y = [int(k['Taille']) for k in joueurs if k['Poste'] == 'Centre']
plt.plot(X,Y,'bo')
#on recolorie les 2ème ligne en vert
#on recolorie les Pilier en jaune
#on recolorie les demi de Mêlée en magenta
#on recolorie les Arrières en cyan
```

3. Trier des données

Programme N.S.I

Contenus	Capacités attendues	Commentaires
Tri d'une table	Trier une table suivant une colonne	Une fonction de tri intégrée au système ou à une bibliothèque peut être utilisée.

3.0.5 Créer une fonction filtre

Question 12.

L'objectif est de créer une fonction joueursClub(equipe) qui renvoie une liste contentant tous les joueurs de l'équipe equipe.

Le paramètre equipe sera donnée sous forme de chaîne de caractères. La valeur renvoyée sera de type liste.

```
def joueursClub(equipe):
    ret = []
    pass
    return ret

joueursClub("Clermont")
```

? Question 13.

Ecrire une fonction <code>nomJoueursClub(equipe)</code> qui cette fois ne renvoie que les noms des joueurs pour le club.

```
def nomJoueursClub(equipe):
    rep=joueursClub(equipe)
    jrs=[]
    pass

    return jrs

print(len(nomJoueursClub("Clermont")))
nomJoueursClub("Clermont")
```

? Question 14.

Définir de la même manière une fonction joueursPoste(poste) . pour l'ensemble des joueurs puis écrire une fonction pour le nom des joueurs club par poste.

3.1 Utilisation d'une fonction de tri

Comment classer les joueurs suivant leur taille ?

La fonction sorted(liste) est efficace sur les listes : elle renvoie une nouvelle liste triée dans l'ordre croissant.

```
mylist = [4,2,8,6]
mynewlist = sorted(mylist)
print(mynewlist)
```

[2, 4, 6, 8]

Mais comment trier un dictionnaire?

```
test = sorted(joueurs)

Traceback (most recent call last):
   File "<input>", line 1, in <module>
TypeError: '<' not supported between instances of 'dict' and 'dict'</pre>
```

Il est normal que cette tentative échoue : un dictionnaire possède plusieurs clés différentes. Ici, plusieurs clés peuvent être des critères de tri : la taille, le poids.

3.1.1 un exemple de tri de dictionnaire

10

La création de cette fonction age() va nous permettre de spécifier une clé de tri, par le paramètre key :

```
triSimpsons = sorted(Simpsons, key = age)

triSimpsons

triSimpsons = sorted(Simpsons, key = age, reverse = True)

triSimpsons
```

On va créer une nouvelle fonction ageP(nom) permettant de renvoyer l'age du personnage directement.

```
def ageP(nom):
    for n in Simpsons:
        if n['Prenom']==nom:
            return age(n)

ageP('Bart')
```

10

? Question 15.

Trier les joueurs du top14 par taille.

Adapter ensuite pour afficher que certains renseignements.

```
def taillePlayer(player) :
    return pass

joueurs_taille_croissant = sorted(joueurs, key = taillePlayer)
print(joueurs_taille_croissant)

for nom in joueurs_taille_croissant:
    print(nom['Nom'], nom['Taille'])
```

? Question 16.

Trier les joueurs de Clermont par taille.

? Question 17.

Trier les joueurs de Clermont suivant leur Indice de Masse Corporelle (IMC)

Question 18.

En s'inspirant de l'exemple, écrire une fonction permettant d'avoir l'IMC d'un joueur à l'aide de son nom et prénom

3.2 Recherche des joueurs de profil physique similaire

3.2.1 Distance entre deux joueurs

Construire une fonction distance(joueur1, joueur2) qui renvoie la somme des carrés des différences de tailles et de poids entre les joueurs joueur1 et joueur2 : $$$ d = (p_1-p_2)^2 + (t_1-t_2)^2$$

```
def distance(joueur1, joueur2):
    p1 = int(joueur1['Poids'])
    p2 = int(joueur2['Poids'])
    t1 = int(joueur1['Taille'])
    t2 = int(joueur2['Taille'])
    return (p1-p2)**2+(t1-t2)**2

def distanceJ(joueur1, joueur2):
    pass
    return d
```

3.2.2 Distance des joueurs avec Baptiste Serin

Retrouvons d'abord le numéro de Baptiste Serin et de joueur2 dans notre classement de joueurs. (Fonction déja faite)

python

```
distanceJ('PARRA Morgan', 'SERIN Baptiste')
16
```

Pour pouvoir utiliser le tri dans un dictionnaire, il faut intégrer la recherche du numero dans la fonction

```
def distanceSerin(joueur2):
    pass

joueurs_VS_Serin = sorted(joueurs, key=distanceSerin)

joueurs_VS_Serin

for j in joueurs_VS_Serin:
    print(j['Nom'], distanceSerin(j))
```

? Question 19.

Ecrire une fonction qui renvoie un dictionnaire de la forme dico={'France':nb de joueurs, ...}