



Examen d'informatique

1^{er} semestre AU : 2021 – 2022

Date : 03 Janvier 2022

Section : BG2

Nombre de page : 3

L'utilisation des calculatrices n'est pas autorisée pour cette épreuve.

Le langage de programmation sera obligatoirement Python.

*

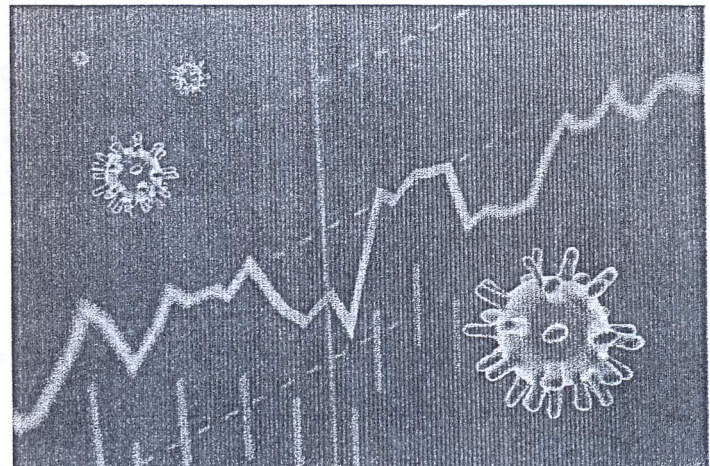
* *

Implémentation. Dans ce sujet, nous adopterons la syntaxe du langage Python. On rappelle qu'en Python, il importe de bien respecter les indentations car elles permettent de définir des blocs.

Covid 19 et Data Science

La pandémie de Covid-19 a provoqué une mobilisation générale des gens et institutions ayant vocation à sauver des vies.

Virologie, immunologie, pharmacologie, épidémiologie... les sciences de la vie ont été les premières à se mobiliser, avec pour toutes un besoin impérieux de quantifier, modéliser, simuler... Au XXI^e siècle, les sciences du vivant brassent des données. C'est pourquoi les *data sciences* sont montées au front. Mobilisée, la communauté des *data sciences* multiplie les initiatives dans la lutte contre le Covid-19. Elle participe à la compréhension du virus et de sa propagation en vue de fournir des outils à l'action publique et de concevoir un arsenal diagnostique et pharmacologique.



Dans notre sujet, nous allons mettre l'accent à ces types de données pour donner des outils d'analyse pour aider les experts à comprendre ces données afin de prendre les bonnes décisions.

Pour cela, on va se baser sur des données des patients infectés par le virus covid 19 enregistrées dans un fichier texte "patients_covid.csv".

Voici un extrait de ce fichier :

age	genre	douleur	pression	cholester	sucre	angine	coeur	date	city
53	feminin	D	140	203	B	oui	presence	2020-03-21	Sfax
67	masculin	A	110	211	A	oui	absence	2020-03-21	Monastir
53	feminin	C	140	199	A	oui	absence	2020-03-21	Tunis
67	masculin	D	120	229	A	oui	presence	2020-03-21	Sfax
67	masculin	B	130	245	A	non	absence	2020-03-21	Monastir

Partie I. Patient

Chaque ligne du fichier csv représente les propriétés d'un patient.

Question 1 :

Écrire une fonction `get_Feature(nomF, sep)` qui retourne la liste des propriétés des patients à partir du fichier `nomF`, sachant que les features existent dans la première ligne du fichier séparée par le paramètre `sep`.

```
>>> features = get_Feature("patients_covid2.csv" , ';')
>>> print(features )
['age', 'genre', 'douleur', 'pression', 'cholester', 'sucre', 'angine', 'coeur', 'date', 'ville']
```

Question 2 :

Un patient sera représenté sous la forme d'un dictionnaire tels que :

- les clés sont les propriétés (feature) du patient, par exemple (age , genre , pression, ...)
- les valeurs du dictionnaire sont les valeurs des features.

Écrire une fonction qui retourne le dictionnaire représentant le patient dans la i^{ème} ligne du fichier nommé `nomF`, sachant que la première ligne d'indice 0 est celle des features.

```
>>> patient2 = get_patient(2 , "patients_covid2.csv" , ";")
>>> print(patient2)
{'age': '67', 'genre': 'masculin', 'douleur': 'A', 'pression': '110', 'cholester': '211', 'sucre': 'A', 'angine': 'oui', 'coeur': 'absence', 'date': '2020-03-21', 'ville': 'Monastir'}
```

Partie II. Serie

Chaque colonne dans le fichier CSV sera représentée par une liste `serie` tels que :

- `serie[0]` : le nom de la propriété (feature)
- `serie[1]` : la liste des valeurs de cette feature.

Question 3 :

Ecrire la fonction `average(serie)` qui calcule et renvoie la moyenne de la `serie`.

Question 4 :

Ecrire la fonction `variance(serie)` qui calcule et renvoie la variance de la serie.

Sachant que : $\text{variance}(s) = \frac{1}{n} \sum_{k=0}^{n-1} (v_k - \bar{v})^2$;

où v_k est la $k^{\text{ème}}$ valeur de la Serie s et \bar{v} sa moyenne.

Question 5 :

Ecrire la fonction `value_counts(serie)` qui renvoie un dictionnaire dont :

- les clés sont les différents valeurs de la liste `serie[1]`.
- les valeurs sont les occurrences de chaque valeur de la liste `serie[1]`.

```
>>> s_age = [ 'age' , [76, 40, 76, 55, 14, 76, 55, 68] ]
>>> print("value counts = " , value_counts(s_age))
value counts = {76: 3, 40: 1, 55: 2, 14: 1, 68: 1}
```

Question 6:

Ecrire la fonction `plot_counts(serie)` qui trace la courbe tels que les abscisses sont les différents valeurs la liste `serie[1]` et les ordonnées sont les occurrences de ces valeurs.

Partie III. DataSet

L'ensemble des données dans le fichier CSV sera représenté sous la forme d'une liste de liste `serie` nommé `dataSet`

Question 8 :

Compléter la fonction `initialise_dataSet(nomF, sep)` qui initialise et renvoie la liste de liste `dataSet` à partir du fichier `nomF` dont les colonnes sont séparées par `sep`.

```
def initialise_dataSet(nomF , sep) :
    dataSet = list()
    f = open(nomF , 'r' )
    ligne = f.readline()
    features = ... # faire appel à la fonction de la question 1
    for i in range(len(features)) :
        serie = [features[i] , []]
        ... # Ajouter serie à la liste dataSet
    f.close()
    return dataSet
```

Question 9 :

Compléter la fonction `load_dataSet(nomF, sep)` qui remplit et renvoie la liste de liste `dataSet` à partir du fichier `nomF` dont les colonnes sont séparées par `sep`.

```
def load_dataSet(nomF , sep) :
    f = open(nomF , 'r' )
    ligne = f.readline()
    dataSet = ... # initiser la dataSet
    for ligne in f :
        ligne = ligne.strip() # supprimer le \n
        L = ligne.split(sep) # [40 , masculin , sfax , ...]
        for i in range(len(L)) :
            ... # ajouter l'élément de L dans la dataSet
    f.close()
    return dataSet
```