

L'objectif de cette fiche est l'étude statistique de données en **Python**. Nous aurons besoin de la bibliothèque **pandas** que l'on importera ainsi :

```
import pandas as pd
```

Pour les graphiques, nous aurons besoin de la bibliothèque **matplotlib.pyplot** et de **numpy** pour certains calculs.

### STATISTIQUES DESCRIPTIVES AVEC PANDAS

La première chose pour traiter des données est d'importer le fichier contenant ces données. Généralement, le fichier sera en format **.csv** (comma-separated values).

1. Sur Teams, télécharger le fichier **notes\_promo\_2024.csv** et l'importer sous le nom **notes** avec la commande :

```
notes=pd.read_csv("notes_promo_2024.csv",sep=";",low_memory=False)
```

**Petite remarque**  
 Les données du fichier csv initial sont séparées par des ';' (parfois, elles le sont pas des ';'), on le précise donc.

2. Placer ce fichier, ainsi que le fichier **Python** du TP dans un même dossier.
3. Quelques premières commandes :

Commande Python	Résultat
<b>notes</b>	aperçu du tableau de données
<b>notes.head()</b> , <b>notes.head(n)</b>	5 premières lignes, <i>n</i> premières lignes
<b>notes.tail()</b> , <b>notes.tail(n)</b>	5 dernières lignes, <i>n</i> dernières lignes
<b>notes.shape</b>	taille du tableau

- 3.a. Combien le tableau contient-il de lignes et de colonnes?

Le tableau contient 28 lignes et 17 colonnes. La ligne des titres des colonnes n'est donc pas considérée comme une ligne de données.

- 3.b. Écrire et exécuter le code suivant :

```
1 for x in notes:
2     print(x)
```

Que permet de faire ce programme? Quelles informations obtient-on?

- Boucler sur un tableau équivaut à boucler sur le nom de ses colonnes.
- On obtient les noms de toutes les colonnes du tableau. Bien pratique pour la suite...

- 3.c. Quelles sont les données présentes dans ce tableau?

Les notes lors des DS ainsi qu'aux épreuves écrites des concours; la moyenne annuelle, la moyenne des épreuves TOP5, celle des épreuves TOP3 et la moyenne des écrits pour la classe de ECG4 promo 2024.

4. Pour ordonner et trier :

Commande Python	Résultat
<b>notes[['Colonne']]</b>	extraction de la colonne intitulée <b>Colonne</b> (possible de renommer la colonne)
<b>notes[['Colonne1','Colonne2']]</b>	extraction des colonnes intitulées <b>Colonne1</b> et <b>Colonne2</b> (possible de renommer le tableau obtenu)
<b>notes.sort_values('Colonne')</b>	ordonne la colonne <b>Colonne</b> dans l'ordre croissant

**★Subtil...★**  
 Il y a une différence entre **notes['Colonne']** et **notes[['Colonne']]**...

**Petite remarque**  
 Si le tableau ne contient qu'une colonne, la commande **tableau.sort\_values()** permet de l'ordonner.

- 4.a. Écrire une commande permettant d'afficher les notes du DS0.

```
notes[['DS0']]
```

- 4.b. Écrire une commande permettant d'afficher les noms et prénoms des étudiantes et étudiants ainsi que leur moyenne aux concours.

```
notes[['NOM','Prenom','EcritsTotal']]
```

- 4.c. Que permet d'obtenir la commande **notes[notes['Prenom']=='Julie']**?

Elle permet d'afficher le sous-tableau (la ligne ici) tel que Julie soit écrit en colonne Prenom. Autrement dit, elle permet d'afficher les notes et moyennes de Julie.

- 4.d. Écrire une commande permettant d'afficher la note obtenue par Julie lors de l'épreuve 1 du CB1.

```
notes.at[2,'CB1 Epreuve1']
```

**Petite remarque**  
 La commande **tableau.at['ligne','colonne']** répond à la question...

4.e. Écrire une commande permettant d'obtenir les étudiantes et étudiants dont la moyenne aux écrits TOP5 est supérieure ou égale à 13.

```
tableau[tableau['EcritsTOP5']>=13]
```

5. Indicateurs statistiques sur un tableau nommé `tableau` :

Commande Python	Résultat
<code>tableau.describe()</code>	indicateurs statistiques usuels
<code>tableau.count()</code>	effectif
<code>tableau.min()</code> , <code>tableau.max()</code>	minimum, maximum
<code>tableau.mean()</code>	moyenne
<code>tableau.std()</code>	écart-type
<code>tableau.median()</code>	médiane
<code>tableau.sum()</code>	somme

5.a. Écrire une commande permettant d'obtenir les indicateurs statistiques sur la moyenne annuelle ainsi que sur la moyenne aux épreuves écrites.

- Pour la moyenne annuelle : `notes[['AnneeTotal']].describe()`
- Pour la moyenne aux écrits : `notes[['EcritsTotal']].describe()`

5.b. Écrire une commande permettant d'obtenir le nombre d'étudiantes et étudiants ayant eu une moyenne aux écrits TOP5 supérieur ou égale à 13.

Deux propositions :

- `notes[notes['EcritsTOP5']>=13].count()` convient..
- On peut toutefois exécuter successivement `top5=notes[['EcritsTOP5']]` puis `top5[top5['EcritsTOP5']>=13].count()`.

6. Représentations graphiques :

Commande Python	Résultat
<code>plt.boxplot(tableau['Colonne'])</code>	boite à moustaches des données de la colonne <b>Colonne</b>
<code>plt.hist(tableau['Colonne'],...)</code>	histogramme des données de la colonne <b>Colonne</b> ; "..." est soit vide, soit le nombre de rectangles, soit une liste de bornes des rectangles
<code>plt.bar(abscisses, ordonnees)</code>	diagramme en barres

**Petite remarque**  
 On peut aussi obtenir un histogramme avec des classes de tailles différentes, explicitement choisies. On précise alors `bins=liste_des_bornes` en paramètre à la place de `n`.

Représenter quelques graphiques sur différentes plages de données au choix.

RÉGRESSION LINÉAIRE SUR UN CAS PRATIQUE

On s'intéresse ici à la corrélation entre les notes obtenues par les étudiantes et étudiants durant l'année avec celles obtenues lors des écrits.

7. A la suite du programme précédent, recopier et exécuter le programme suivant :

```
1 x=notes["AnneeTotal"]
2 y=notes["EcritsTotal"]
3 plt.plot(x,y,"b+")
4 plt.xlabel("moyennes en CPGE")
5 plt.ylabel("moyennes aux écrits")
6 plt.show()
```

8. Exécuter les instructions `np.std(x)` et `x.std()`. Commenter.

Les deux instructions devraient renvoyer l'écart-type de la série statistique `x...` Mais elles renvoient deux valeurs différentes !!

9. Écrire une ligne de commande permettant de calculer l'écart-type de `x` "à la main".

`np.sqrt(1/len(x)*sum([(z-xb)**2 for z in x]))` permet bien de calculer la racine carrée de la variance de `x`, en revenant à la définition de la variance.

10. Écrire les instructions permettant de déterminer les coordonnées du point moyen du nuage de points ainsi représenté puis de le placer sur le nuage de points.

Le point moyen du nuage a pour coordonnées  $(\bar{x}, \bar{y})$ ...

```
xb=np.mean(x)
yb=np.mean(y)
plt.plot(xb,yb,'r+') (à placer avant plt.show())
```

**Pourquoi ?**  
 Explication lors du dernier chapitre de l'année...

11. Écrire les instructions permettant de représenter la droite de régression linéaire obtenue par la méthode des moindres carrés; puis calculer le coefficient de corrélation linéaire. Commenter.

Voici le fichier complet :

```
1 import matplotlib.pyplot as plt
2 import numpy as np
3 import pandas as pd
4
5 notes=pd.read_csv("notes_promo_2024.csv",sep=";",low_memory=False)
6
7 choix=input('TOP5,TOP3 ou total ? ')
8 if choix=='TOP5':
9     x=notes["AnneeTOP5"]
10    y=notes["EcritsTOP5"]
11 elif choix=='TOP3':
12    x=notes["AnneeTOP3"]
13    y=notes["EcritsTOP3"]
14 elif choix=='total':
15    x=notes["AnneeTotal"]
16    y=notes["EcritsTotal"]
17
18 plt.plot(x,y,"b+")
19 plt.xlabel("moyennes en CPGE")
20 plt.ylabel("moyennes aux écrits")
21
22 sx=np.std(x)
23 sy=np.std(y)
24 xb=np.mean(x)
25 yb=np.mean(y)
26 varx=1/len(x)*sum([(z-xb)**2 for z in x])
27 covxy=1/len(x)*sum([(x[i]-xb)*(y[i]-yb) for i in range(0,len(x))])
28 a=covxy/varx
29 b=yb-a*xb
30 plt.plot([min(x),max(x)],[min(x)*a+b,max(x)*a+b],"g")
31 plt.plot(xb,yb,"r+")
32 plt.show()
33 print("coefficient de corrélation linéaire=",covxy/(sx*sy))
34 print('coefficient directeur=',a)
35 print("ordonnée à l'origine=",b)
```

En prenant les moyennes annuelles et aux écrits (TOP3 et TOP5 confondus), on trouve :

- la droite de régression linéaire obtenue par la méthode des moindres carrés a pour équation  $y = ax + b$ , avec  $a \simeq 0,74$  et  $b \simeq 7,33$ .
- Le coefficient de corrélation linéaire vaut environ 0,93 : l'ajustement affine est très bon !