

— DS 4 —

Devoir surveillé du Mardi 12 Novembre

La calculatrice est interdite. Durée : 1h30

Dans tout ce devoir, on suppose avoir importé sur `Python` les bibliothèques :

- `numpy` avec le raccourci `np` ;
- `numpy.random` avec le raccourci `rd` ;
- `matplotlib.pyplot` avec le raccourci `plt` ;
- `pandas` avec le raccourci `pd`.

Exercice 1

1. Définir la fonction f suivante sur `Python` :

$$f(x) = \begin{cases} x^2 & \text{si } x < 0, \\ \ln(x + 1) & \text{si } x \geq 0. \end{cases}$$

2. Définir une subdivision de l'intervalle $I = [-3, 3]$ en 100 sous-intervalles de même longueur, puis écrire des instructions permettant de tracer la courbe de f définie sur I .

Exercice 2

1. Écrire une fonction `indicemin` en langage Python qui, pour une liste `L`, renvoie l'indice du minimum de la liste.

2. Si `L` est une liste, que fait l'instruction `del L[i]` ?

3. Dédurre des deux premières questions une fonction `min2` qui, pour une liste `L`, renvoie la valeur du deuxième minimum de la liste.

4. Compléter la fonction `tri` suivante pour que, étant donné une liste `L`, elle renvoie la liste triée dans l'ordre croissant des éléments de `L`. On détaillera ligne par ligne ce que fait cette fonction.

```
1 | def tri(L):  
2 |     M = []  
3 |     while len(L) > 0 :  
4 |         i = .....  
5 |         M.append( ..... )  
6 |         del .....  
7 |     return M
```

Exercice 3

1. Écrire en `Python` une fonction `facto` qui, étant donné un entier naturel n , retourne $n!$.
2. Écrire en `Python` une fonction `coefbin` qui, étant donnés deux entiers $0 \leq k \leq n$, retourne la valeur du coefficient binomial $\binom{n}{k}$. Cette fonction fera appel à la fonction `facto`.
3. Écrire en `Python` une fonction `somme` qui, étant donné un entier naturel n , retourne la valeur de la somme $\sum_{k=0}^n \binom{n}{k}$. Cette fonction fera appel à la fonction `coefbin`.

4. A la suite des fonctions précédentes, on ajoute les instructions Python suivantes :

```
1 | M = np.zeros(10)
2 | for k in range(10):
3 |     M[k] = somme(k)
4 | print(M)
```

Lorsqu'on exécute le programme, on obtient alors le résultat suivant dans la console :

```
[ 1.  2.  4.  8. 16. 32. 64. 128. 256. 512.]
```

Que contient la variable M ? Les résultats obtenus étaient-ils prévisibles ? Pourquoi ?

Exercice 4

L'indice de masse corporelle (IMC) d'une personne est définie par $I = \frac{P}{T^2}$ où P est le poids en kilogramme et T la taille en mètre de cette personne.

1. La commande `rd.normal(μ, σ, r)` permet de simuler un vecteur de taille r dont les coefficients suivent la loi normale $\mathcal{N}(\mu, \sigma^2)$.

On considère que la taille d'une personne (en mètres) et son poids (en kg) sont deux variables aléatoires indépendantes (ce qui est quand même très discutable, mais admettons...), suivant respectivement des lois normales $\mathcal{N}(1.78, 0.04)$ et $\mathcal{N}(72, 6)$.

Donner les instructions Python pour créer des vecteurs P et T contenant respectivement les simulations des poids et de la taille de 1000 personnes.

2. En déduire, en une ligne de commande Python, un vecteur I contenant l'IMC de ces 1000 personnes.
3. Donner les instructions Python pour calculer la médiane, la moyenne, la variance et l'écart type de l'IMC de ces 1000 personnes.

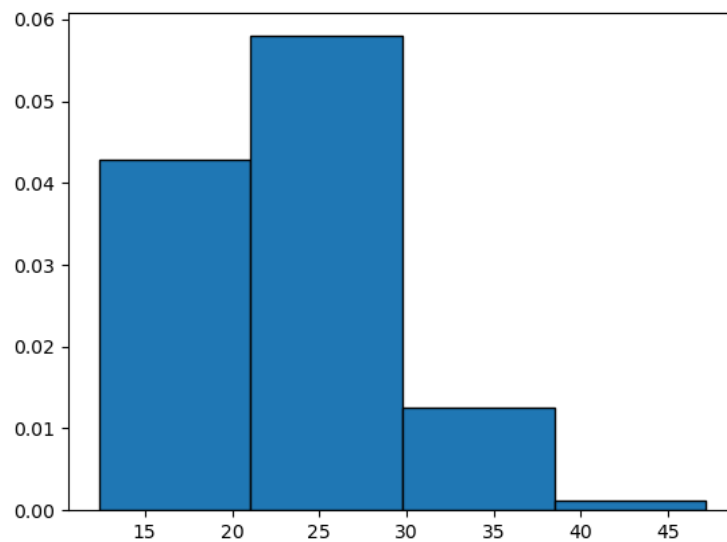
4. Une personne est considérée en surpoids lorsque son IMC dépasse 25. Donner une instruction Python permettant de déterminer la proportion de personnes en sur-poids.

5. A votre avis, quel est le plus pertinent : trier notre série statistique I par modalités ou par classes ? Justifier.

6. On considère le script Python suivant :

```
1 | m = np.min(I)
2 | M = np.max(I)
3 | c = np.linspace(m,M,5)
4 | plt.hist(I, c, density = 'True', edgecolor = 'k')
5 | plt.show()
```

ainsi que le graphique qu'il génère :



(a) Expliquer ce programme. En particulier, à quoi correspond la variable c ?

- (b) Déterminer graphiquement la classe modale de la série statistique.

Exercice 5

On importe dans la console un fichier `csv` nommé `test` :

```
>>> df = pd.read_csv("test.csv"); print(df)
```

	Prenom	Age	Ville	Dispo	Permis	Enfants
0	Ines	22	Bondy	8	non	2
1	Leo	19	Paris	10	oui	0
2	Tom	20	Paris	12	non	0
3	Lea	18	Orly	8	oui	0
4	Mick	20	Paris	8	oui	0
5	Eva	22	Paris	8	oui	1
6	Mael	17	Gagny	18	non	0

1. Que font les instructions Python suivantes :

(a) `df.shape`

(b) `df['Age'].std()`

(c) `df[df['Ville']=='Paris']`

2. Donner les instructions à entrer dans la console Python pour :

(a) avoir la médiane de la colonne 'Dispo' ;

(b) sélectionner les personnes qui ont au moins un enfant ;

(c) sélectionner les personnes qui ont le permis et pas d'enfant.