TP5a - Vocabulaire des statistiques

Dans tout le TP, on importe les modules suivants :

```
1 import numpy as np
  import numpy.random as rd
  import matplotlib.pyplot as plt
```

1 Généralités

Dans ce TP, nous allons basculer des probabilités aux statistiques. Les statistiques sont assez peu présentes au programme (on en fera un tout petit peu en fin d'année) car elles sont assez difficiles à formaliser. Mais c'est ironiquement une tâche qui vous attend *a priori* beaucoup plus que l'étude des probabilités dans vos futurs métiers après les écoles de commerces. Le but de ce TP est d'introduire quelques notions de statistiques et d'en profiter pour apprendre à manipuler les représentations graphiques.

Exercice 1 - Définitions ⋆

Définir `statistiques` et `probabilités`. Quelle est la différence entre les deux ?

2 Vocabulaire des statistiques

- **Population :** Ensemble mathématique représentant la population étudiée. Le formalisme est proche de celui des probabilités. La population est notée $\Omega = \{\omega_1, \dots, \omega_n\}$. Puisque l'on cherche à traiter des données réelles, Ω est nécessairement finie.

  Exemple : Ω peut être l'ensemble des élèves de la classe.

- **Effectif :** c'est le cardinal de la population. Si $\Omega = \{\omega_1, \dots, \omega_n\}$ alors l'effectif est $\text{card}(\Omega) = n$.

  Exemple : Avec l'exemple précédent, l'effectif est 24.

- **Caractère :** un caractère X est une application de Ω dans un ensemble quelconque. Cela représente les différentes quantités que l'on va étudier. On dit qu'un caractère est **quantitatif** si $X(\Omega) \subset \mathbb{R}$ (on dit sinon qu'il est **qualitatif**).

  Pour pouvoir faire des graphes et plus généralement des maths, on s'intéresse plutôt (voire exclusivement) à des caractères quantitatifs.

  Exemple : Toujours dans la classe, la couleur de cheveux des élèves est un caractère qualitatif tandis que leurs tailles en `cm` est un caractère quantitatif.

- **Série statistique :** c'est l'ensemble des valeurs $x_1, \dots, x_n$ prises par un caractère X, comptées avec leurs multiplicités. Les valeurs prises sont quant à elles appelées des **modalités**. Le nombre de fois qu'une modalité apparaît (c'est aussi le nombre d'antécédents par X de la modalité) est appelé l'**effectif de la modalité**.

  Exemple : Les modalités de la série statistiques $\{5, 1, 4, 2, 2, 3, -2, 5, 7, 3\}$ sont $-2, 1, 2, 3, 4, 5, 7$. L'effectif de 3 est 2 et celui de 4 est 1.

- **Moyenne :** la moyenne d'une série statistique est $\overline{x} = \frac{x_1 + \dots + x_n}{n} = \frac{1}{n}\sum_{i=1}^{n} X(\omega_i)$. On peut la calculer sur un tableau `numpy` avec `np.mean`.

  Exemple : la moyenne de la série précédente est : 3.

- **Variance et écart-type :** la variance d'une série statistique est la quantité :$\sigma(X)^2 = \frac{1}{n}\sum_{i=1}^{n}(x_i - \overline{x})^2$. On utilise parfois la formule : $\frac{1}{n-1}\sum_{i=1}^{n}(x_i - \overline{x})^2$ avec un dénominateur différent. On en reparlera dans le chapitre sur les estimateurs. Le point important est que par défaut, la commande `np.var` utilise la première formule mais on on peut lui faire utiliser la seconde formule avec `np.var(serie,ddof=1)`.

  L'écart-type est la racine carrée de la variance.

  Exemple : Avec la série précédente, la variance (première formule) est $5,6$ et l'écart-type environ $2,37$.

- **Étendue :** l'étendue d'une série statistique est la différence entre la plus grande et la plus petite valeur. On peut l'obtenir en `Python` avec `np.max(serie)` − `np.min(serie)`.

  Exemple : dans la série utilisée en exemple l'étendue est 8.

- **Médiane :** une médiane est un nombre qui sépare une série statistique en deux parties égales, une partie de nombres inférieurs à la médiane et une partie de nombres supérieurs. Il n'est absolument pas garantie qu'un tel nombre existe ou soit unique.
  On a cependant une convention utile (qui est celle de `numpy` avec `np.median`) qui est d'ordonner les valeurs de la série numérique $x_1 \leq x_2 \leq x_3 \leq \cdots \leq x_n$ puis de prendre le nombre :
  - $x_{(n+1)/2}$ si n est impair ;
  - $\frac{x_{n/2}+x_{n/2+1}}{2}$ si n est pair.

  Exemple : dans la série utilisée en exemple la médiane est 3.
- **Fréquence d'une modalité :** c'est son effectif ramené à l'effectif total.
  Exemple : dans la série utilisée en exemple la fréquence de 5 est $2/8 = 1/4$.
- **Fréquence cumulée :** c'est la somme des fréquences des modalités inférieures (ou égales). Si les modalités sont $x_1 \leq x_2 \leq \cdots \leq x_n$, et leurs fréquences respectives $f_1,\ f_2, \ldots\ , f_n$ alors la fréquence cumulée de x_i est $f_1 + f_2 + \cdots + f_i$.
  Exemple : la fréquence cumulée de la plus petite modalité est simplement sa fréquence et la fréquence cumulée de la plus grande est 1.
- **Quartiles :** le premier quartile q_1 est la plus petite modalité ayant une fréquence cumulée supérieure à $1/4$. Le troisième quartile q_3 est la plus petite modalité ayant une fréquence cumulée supérieure à $3/4$.
  On appelle **intervalle interquartile** l'intervalle $[q_1, q_3]$. Environ la moitié des valeurs de la série sont dans cette intervalle. On peut utiliser la méthode `np.quantile` pour les estimer. Attention, elle prend deux arguments : la série et une fréquence cumulée de référence.
  Exemple : `np.quantile(serie,0.25)` renvoie une estimation du premier quartile.

3 Applications

Exercice 2 ⋆

1. Générer dans la variable `A` un tableau `numpy` contenant 1000 valeurs tirées au hasard selon la loi $\mathcal{U}([0, 10])$.
2. Calculer la moyenne, l'écart-type, la médiane et les quartiles de la série. Les résultats étaient-ils prévisibles ?

Exercice 3 ⋆

Comme dans l'exercice précédent, générer une série statistique mais avec un petit effectif n ($n \leq 10$). Calculer les quartiles. Commenter. Que constate-t-on si $n < 4$?

Exercice 4 ⋆⋆

1. Compléter le code suivant qui simule le lancer de 2 dés et qui renvoie la série de n somme des dés.

```
def simul_somme_des(n):
  A = np.zeros(...)
  for i in range(...):
    de1 = rd.randint(...)
    de2 = ...
    A[i] = ...
  return A
```

2. Utiliser la fonction précédente pour créer une série statistique de 1000 valeurs stockées dans un tableau `A`. On utilise la ligne suivante `valeurs,effectifs = np.unique(A,return_counts=True)` pour générer deux nouveau tableaux. Le premiers contient les modalités (classées par ordre croissante) et le second leurs effectifs. Rappeler le rôle de la fonction `np.cumsum` puis écrire un code `Python` pour obtenir le tableau des fréquences cumulées qu'on appellera `freq_cumul`.
3. On va utiliser la bibliothèque `matplotlib` pour faire des tracés. Utiliser le code suivant :

```
plt.plot(valeurs,freq_cumul)
plt.show()
```

Que se passe-t-il ? Commenter le graphique.