⋆ Statistiques descriptives univariées ⋆

Compétences attendues

- ✓ Regrouper une série statistique par modalités ou par classes.
- ✓ Connaître les indicateurs de position (moyenne, médiane, quartiles) et les commandes associées.
- ✓ Connaître les indicateurs de dispersion (écart-type, étendue, distance inter-quartile) et les commandes associées.
- ✓ Représenter graphiquement une série statistique.

Liste des commandes Python exigibles aux concours

- Dans la librairie `numpy` : `sum`, `min`, `max`, `cumsum`, `mean`, `median`, `var`, `std` ;
- Dans la librairie `matplotlib.pyplot` : `hist`, `show`

Objectifs. L'objet des statistiques descriptives univariées (ou unidimensionnelles) est de fournir des résumés synthétiques, graphiques et numériques, de séries de valeurs observées sur une population ou un échantillon. On présente ici les indicateurs les plus couramment employés pour décrire une série statistique.

1 Principales notions en statistiques descriptives

1.1 Présentation des données

On considère un ensemble Ω appelé *population* en statistique descriptive. On appellera ses éléments ω des individus.

Exemple 1.1. $\Omega =$ l'ensemble de la population française, $\Omega =$ l'ensemble des voitures immatriculées en France.

On étudie un *caractère* de cette population.

Définition 1.1

Un caractère (ou variable) sur la population Ω est une application $X : \Omega \to E$, où E désigne un ensemble quelconque.
Si E est un ensemble de nombres, on dit que X est un caractère quantitatif. Dans le cas contraire, on parle de caractère qualitatif.

Exemple 1.2. Un caractère possible sur la population française est la taille (caractère quantitatif) ou encore la couleur des yeux (caractère qualitatif).

Nous ne traiterons que du cas des caractères quantitatifs.

La connaissance complète d'un caractère X peut être rendue difficile, voir impossible, de part la taille de la population Ω. Afin de pouvoir l'étudier, on peut considérer ce caractère seulement pour une partie finie $\{\omega_1, \dots, \omega_n\}$ de Ω appelée échantillon. Son cardinal n est alors la taille ou l'effectif de l'échantillon.

Définition 1.2

- On appelle *série statistique* d'un échantillon $\{\omega_1, \dots, \omega_n\} \subset \Omega$ (ou *échantillon observé*) pour le caractère X la donnée de la liste $(x_1, \dots, x_n) = (X(\omega_1), \dots, X(\omega_n))$ des valeurs prises par X sur l'échantillon.
- Les valeurs prises par X sont appelées *modalités*.
- L'*effectif d'une modalité* m est le nombre n_m de fois où m apparait dans la série statistique $(x_1, \dots, x_n)$.
- La *fréquence d'une modalité* m est le réel $f_m = \frac{\text{effectif}}{\text{effectif total}} = \frac{n_m}{n}$.
- La *fréquence cumulée d'une modalité* m est le réel $p_m = \sum_{m' \leq m} f_{m'}$.

Remarque 1.1. Les statistiques sont nées en Angleterre, au début du 17ème siècle pour décompter les décès lors d'une épidémie de peste. Ce n'était à l'époque que des données numériques, sans outil théorique pour les analyser. Il faut attendre le 19ème siècle pour voir l'apparition de méthodes mathématiques pour l'étude de telles données. Ce n'est qu'à la fin du 19ème siècle que la statistique devient une discipline à part entière des mathématiques sous l'impulsion des savants anglais Karl Pearson et Udny Yule.

Exemple 1.3. On considère la série statistique suivante :

$$x = (2, 11, 7, 2, 15, 4, 5, 5, 5, 13, 5, 15, 7, 7, 8, 10, 10, 10, 11, 13, 7, 2, 15, 15).$$

L'ensemble des modalités est $\{2, 4, 5, 7, 8, 10, 11, 13, 15\}$. L'effectif de la modalité $m = 5$ est $n_5 = 4$, sa fréquence est $f_5 = \frac{n_5}{n} = \frac{4}{24}$ et sa fréquence cumulée est $p_5 = \frac{8}{24}$.

Représentation informatique. Sous `Python`, nous représenterons une série statistique $(x_1, \dots, x_n)$ par un vecteur

```
x=np.array([x1, ...,xn])
```

L'effectif de la série est obtenu à l'aide de la commande `np.shape(u)[0]`.

Remarque 1.2.

— Si $x = (x_1, \dots, x_n)$ est une série statistique, $(m_1, \dots, m_p)$ ses modalités d'effectifs $(n_1, \dots, n_p)$ et de fréquences $(f_1, \dots, f_p)$, alors on a :

$$\sum_{i=1}^{p} n_i = n \quad \text{et} \quad \sum_{i=1}^{p} f_i = \sum_{i=1}^{p} \frac{n_i}{n} = \frac{\sum_{i=1}^{p} n_i}{n} = 1$$

— Les notions suivantes se correspondent entre probabilités et statistiques :

$$\begin{aligned} \text{Variable aléatoire } X &\leftrightarrow \text{ Caractère } X \\ \text{Ensemble image } X(\Omega) &\leftrightarrow \text{ Ensemble des modalités de } X \\ \text{Probabilités ponctuelles } \mathbb{P}(X = x) &\leftrightarrow \text{ Fréquences } f_m \\ \text{Fonction de répartition } F_X : x \mapsto \mathbb{P}(X \leq x) &\leftrightarrow \text{ Fréquences cumulées } p_m = \sum_{m' \leq m} f_{m'} \end{aligned}$$

Une série statistique brute ne permettant pas une lecture efficace des données, on souhaite la présenter de manière synthétique. Pour cela, on procède de deux manières distinctes selon le nombre de ses modalités.

Regroupement par modalité

Dans le cas où le nombre de modalités de la série est raisonnable, on regroupe la série par *modalités-effectifs*, c'est-à-dire qu'on donne :

— la liste (m_i) des modalités du caractère X,

— les effectifs (n_i) correspondants.

On peut aussi choisir de présenter cette série regroupée par *modalité - fréquence*, en donnant les modalités (m_i) et les fréquences correspondantes (f_i).

Exemple 1.4. Le tri par modalités de la série statistique x donne :

2	3
4	1
5	4
7	4
8	1
10	3
11	2
13	2
15	4

Tri par modalités - effectifs de la série x.

2	0.125
4	0.0416667
5	0.1666667
7	0.1666667
8	0.0416667
10	0.125
11	0.0833333
13	0.0833333
15	0.1666667

Tri par modalités - fréquences de la série x.

Regroupement par classes

Dans le cas où le nombre de modalités est trop grand, plutôt que de conserver toutes les valeurs, il est plus intéressant de les regrouper par classes :

— on considère une suite de réels $c = (c_0 < \dots < c_k)$ définissant les classes $I_1 = [c_0, c_1], I_2 =]c_1, c_2], \dots, I_k =]c_{k-1}, c_k]$, l'amplitude de la classe I_i étant $c_i - c_{i-1}$;

— on note n_i le nombre d'éléments de X appartenant à l'intervalle I_i pour $1 \leq i \leq k$.

On se ramène ainsi à une série statistique de taille k, dont les modalités sont les milieux $y_i = \frac{c_{i-1}+c_i}{2}$ des classes et d'effectifs correspondants les n_i.

Commandes utiles

On rappelle les commandes suivantes qui pourront être utiles dans la suite.

Définition 1.3

- Si `u` et `v` sont deux vecteurs de même format, l'instruction `u==v` renvoie un vecteur de même format que `u` dont les éléments sont `True` ou `False` selon que les coefficients correspondants de `u` et `v` à cette même place sont égaux ou non.
- Si tous les éléments de `v` sont égaux à un même réel `x`, on peut écrire simplement `u==x`.
- On définit de même les vecteurs booléens `u>v`, `u>=v`, `u<v`, `u<=v` et `u !=v`.
- On rappelle également les connecteurs logiques pour les booléens : `and` (et), `or` (ou), `not` (négation).

Définition 1.4

Si `u` est un vecteur dont les composantes sont des booléens, alors :

- la commande `np.sum(u)` renvoie le nombre de booléens qui ont pris la valeur `True`;
- la commande `np.mean(u)` renvoie la proportion de booléens qui ont pris la valeur `True`.

Exemple 1.5. Supposons avoir représenté la série statistique x à l'aide d'un vecteur `x`. L'instruction `np.sum(x==7)` renvoie l'effectif de la modalité 7, `np.mean(x==7)` renvoie sa fréquence, et `np.mean(x<=7)` sa fréquence cumulée.

1.2 Indicateurs de position

Définition 1.5

On appelle moyenne empirique de la série statistique $x = (x_1, \dots, x_n)$ le réel :

$$\bar{x} = \frac{1}{n} \sum_{i=1}^{n} x_i.$$

Remarque 1.3. Si la série statistique x est groupée par modalités - effectifs, avec les modalités $(m_1, \dots, m_p)$ d'effectifs $(n_1, \dots, n_p)$ et de fréquences $(f_1, \dots, f_p)$, alors on a :

$$\bar{x} = \frac{1}{n} \sum_{i=1}^{p} m_i \cdot n_i = \sum_{i=1}^{p} m_i \cdot \frac{n_i}{n} = \sum_{i=1}^{p} m_i \cdot f_i.$$

On notera sur cette dernière formule la correspondance entre les notions d'espérance en probabilité et de moyenne en statistique :

$$E(X) = \sum_{x_i \in X(\Omega)} x_i P(X = x_i) \quad \leftrightarrow \quad \bar{x} = \sum_{i=1}^{p} m_i \cdot f_i.$$

Définition 1.6 [Médiane]

La *médiane* d'une série statistique ordonnée est un réel m partageant la série en deux séries d'effectifs égaux. Si $(x_1 \leq x_2 \leq \cdots \leq x_n)$ est la série statistique ordonnée, m est défini par :

- si $n = 2p - 1$ est impaire, $m = x_p$ (la valeur du milieu);
- si $n = 2p$ est paire, $m = \frac{x_p + x_{p+1}}{2}$ (la moyenne des deux termes du milieu).

Proposition 1.1 [Moyenne et Médiane en Python]

- `np.mean(x)` donne la moyenne du vecteur `x`.
- `np.median(x)` donne une médiane du vecteur `x` (non nécessairement ordonné).

Définition 1.7 [Quartiles]

Soit $x = (x_1, \dots, x_n)$ une série statistique.

- Le premier quartile q_1 de x est la plus petite valeur de x telle que 25% des valeurs lui soient inférieures ou égales.
- Le troisième quartile q_3 de x est la plus petite valeur de x telle que 75% des valeurs lui soient inférieures ou égales.

Remarque 1.4. De même, on définit les déciles et les centiles d'une série statistique :

- Pour $k \in [\![1, 99]\!]$, le k-ième centile est la valeur c_k de la série pour laquelle moins de $k\%$ de la population prend des valeurs

strictement inférieures à c_k et moins de $(100-k)\%$ de la population prend des valeurs strictement supérieures à c_k.

— Pour $k \in [\![1, 9]\!]$, le k-ième décile est la valeur d_k de la série pour laquelle moins de $10k\%$ (ou k dixième) de la population prend des valeurs strictement inférieures à d_k et moins des $(10-k)$ dixièmes de la population prend des valeurs strictement supérieures à d_k.

Définition 1.8 [Mode]

On appelle *mode* d'une série statistique toute modalité pour laquelle l'effectif est maximal (il peut y en avoir plusieurs).

Exemple 1.6. Reprenons l'exemple de la série statistique x, qu'on trie par modalités - fréquences et par modalités - fréquences cumulées :

2	0.125
4	0.0416667
5	0.1666667
7	0.1666667
8	0.0416667
10	0.125
11	0.0833333
13	0.0833333
15	0.1666667

Tri par modalités - fréquences de x.

2	0.125
4	0.1666667
5	0.3333333
7	0.5
8	0.5416667
10	0.6666667
11	0.75
13	0.8333333
15	1

Tri par modalités - fréquences cumulées de x.

Déterminer le premier quartile, le troisième quartile et le huitième décile, ainsi que le(s) mode(s) de la série x.

1.3 Indicateurs de dispersion

Définition 1.9 [Variance et Écart-type]

Soit $x = (x_1, \dots, x_n)$ une série statistique.

— On appelle variance de x le nombre réel positif : $v = \frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{x})^2$.

— On appelle écart-type de x le réel $\sigma = \sqrt{v}$.

Remarque 1.5.

— La variance mesure la dispersion de la série statistique autour de sa moyenne.

— Comme en probabilités, la formule de Koenig-Huygens est valable :

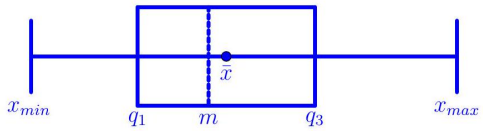
$$v = \frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{x})^2 = \frac{1}{n}\sum_{i=1}^{n}(x_i^2 - 2x_i\bar{x} + \bar{x}^2) = \frac{1}{n}\sum_{i=1}^{n}x_i^2 - 2 \times \bar{x} \times \underbrace{\frac{1}{n}\sum_{i=1}^{n}x_i}_{=\bar{x}} + \bar{x}^2 = \frac{1}{n}\left(\sum_{i=1}^{n}x_i^2\right) - \bar{x}^2.$$

Définition 1.10 [Étendue et distance inter-quartile]

— On appelle étendue d'une série statistique la différence entre la plus grande et la plus petite modalité.

— On appelle distance inter-quartile le réel $q_3 - q_1$.

Remarque 1.6. La distance inter-quartile est un indicateur de dispersion : c'est la longueur de l'intervalle inter-quartile $[q_1, q_3]$, lequel contient la moitié des valeurs de la série, réparties autour de la médiane m. On représente parfois la boite à moustache de la série statistique :



Proposition 1.2 [Variance, écart-type et étendue sur Python]

— `np.var(x)` donne la variance du vecteur `x`.

— `np.std(x)` (pour standard deviation) donne l'écart-type du vecteur `x`.

— `np.max(x) - np.min(x)` donne l'étendue du vecteur `x`.

Exemple 1.7. Déterminer l'écart-type de la série statistique x, et représenter son diagramme à moustache.

2 Représentations graphiques

On suppose avoir importé la bibliothèque `matplotlib.pyplot` à l'aide de l'instruction :

```
import matplotlib.pyplot as plt
```

2.1 Diagrammes en bâtons

Définition 2.1

On représente une série statistique groupée par modalités à l'aide d'un diagramme en bâtons, en plaçant sur l'axe des abscisses les modalités et en dressant à la verticale de chacune d'elles un bâton de hauteur égale à son effectif ou sa fréquence.

On dispose de la commande Python suivante (non exigible).

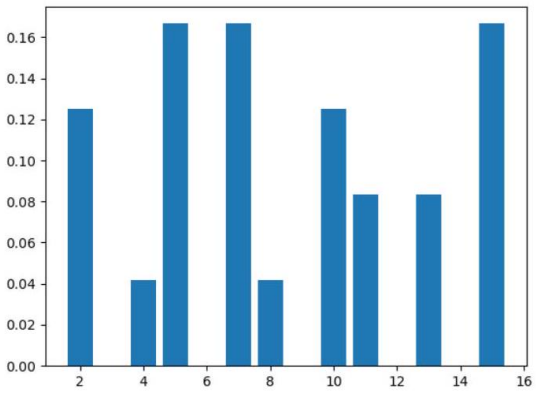
Définition 2.2

Si `x` et `y` sont des vecteurs, `plt.bar(x,y)` trace le diagramme en bâtons d'abscisse `x` et d'ordonnées `y`.

Exemple 2.1. Reprenons la série statistique x. En entrant les instructions suivantes dans la console :

```
m = np.array([2.,4.,5.,7.,8.,10.,11.,13.,15.])
f = np.array([0.125,0.04167,0.16667,0.16667,0.04167,0.125,0.08333,0.08333,0.16667])
plt.bar(m,f)
plt.show()
```

on obtient le diagramme en bâtons des effectifs de la série x :



Remarque 2.1. La commande `plt.bar` nécessite d'avoir trié au préalable la série statistique par modalités - effectifs, ce que nous ne pourrons pas toujours faire. En effet, nous n'avons malheureusement pas de commande Python pour effectuer ce tri. Nous expliquerons ci-dessous comment tracer le diagramme en bâtons d'une série statistique brute à l'aide de la commande `plt.hist`.

2.2 Histogrammes

Définition 2.3

On représente une série statistique **groupée par classes** à l'aide d'un *histogramme,* en plaçant les c_i sur un axe horizontal et en traçant à la verticale un rectangle de base $[c_i, c_{i+1}]$ et d'aire égale à la fréquence de la classe correspondante.

Définition 2.4

Soit `x` un vecteur.

- L'instruction `plt.hist(x, bins=n)` trace l'histogramme associé à la série `x` en `n` classes équiréparties entre la plus petite valeur de `x` et la plus grande (par défaut, `n` vaut 10).
- L'instruction `plt.hist(x,c)` trace l'histogramme associé à la série `x` dont les classes sont définies par le vecteur aux composantes strictement croissantes `c`.

Méthode 2.1 [Comment tracer un diagramme en bâtons d'une série statistique brute ?] *P*

our tracer le diagramme en bâtons d'une série statistique brute (non triée) `x` à valeurs entières, on procède ainsi :

1. on détermine les modalités $m_1 < m_2 < \cdots < m_k$ de la série statistique x ;
2. on définit les classes $c = (m_1 - 0,5 < m_1 + 0,5 < m_2 - 0,5 < m_2 + 0,5 < \cdots < m_k - 0,5 < m_k + 0,5)$;
3. on dessine l'histogramme (le "diagramme en bâtons") des effectifs à l'aide de la commande :

```
plt.hist (x, c, edgecolor='k', color='...', label="...")
```

et l'histogramme (le "diagramme en bâtons") des fréquences à l'aide de la commande :

```
plt.hist(x, c, density='True', edgecolor='k', color='... ', label="...")
```

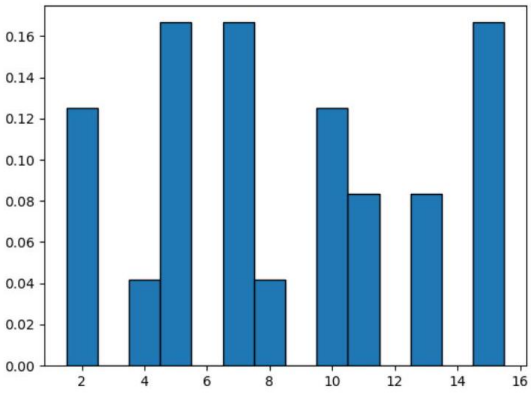
où l'on a ajouté les options de tracé suivantes (non exigibles) :

- normalisation des rectangles (la surface totale vaut 1) : `density= 'True'`
- contours des rectangles en noir : `edgecolor=' k '`
- couleur des rectangles : `color='...'` (mettre le nom de la couleur en anglais)
- légende associée à chaque histogramme : `label="..."` (mettre la légende choisie)

Exemple 2.2. En entrant les instructions suivantes dans la console :

```
c = np.arange(np.min(x),np.max(x)+2)-0.5
plt.hist(x,c,density='True',edgecolor='k')
plt.show()
```

on obtient l'histogramme (le "diagramme en bâtons") des fréquences de la série x :



Notons qu'ici, nous n'avons pas eu au préalable à trier la série statistique x par modalités - fréquences, ceci est directement fait par Python en exécutant `plt.hist`.

3 Exercices

Dans les deux premiers exercices qui suivent, nous aurons besoin de fonctions de la bibliothèque `numpy.random`.

Exercice 1

1. À l'aide de la commande `rd.binomial`, simuler 10000 nombres suivant la loi $\mathscr{B}(10, 0.5)$. On notera `x` le vecteur contenant cette série statistique.
2. Déterminer l'effectif, la fréquence et la fréquence cumulée de la modalité 5.
3. Déterminer la moyenne, la médiane et l'écart-type de `x`. Était-ce prévisible ?
4. Créer un vecteur `m` de taille 11 tel que `m[k]` contient l'effectif de la modalité k. Déterminer le(s) mode(s) de la série `x`.
5. Représenter à l'aide de la commande `plt.bar` les diagrammes en bâtons des effectifs, des fréquences et des fréquences cumulées de la série `x`.
6. Représenter de nouveau le diagramme en bâtons des effectifs et des fréquences de la série `x`, cette fois à l'aide de la commande `plt.hist`.

Exercice 2

1. Créer un vecteur `x` contenant 10000 nombres réels choisis aléatoirement entre 1 et 5 .
2. Calculer la moyenne, la médiane, l'écart-type et l'étendue de la série statistique `x`.
3. Vaut-il mieux regrouper cette série statistique par modalités ou par classes ? Pourquoi ?
4. Tracer l'histogramme associé à cette série statistique en la regroupant par classes (choisir 100 classes de même amplitude). Que remarque-t-on ?

Exercice 3 [Loi de Benford - ++]

La loi de Benford prédit que statistiquement dans une liste de nombres données, la probabilité qu'un de ces nombres commence par le chiffre 1 est plus importante que celle qu'il commence par un 9. Plus précisément, la loi de Benford prédit que la probabilité qu'un nombre commence par le chiffre d est :

$$p_d = \log_{10}\left(1 + \frac{1}{d}\right),$$

où $\log_{10}$ désigne le logarithme en base 10. Il est possible de vérifier que la loi de Benford est la seule qui reste invariante par changement d'unités, i.e. en multipliant les nombres de la liste par une constante les probabilités restent inchangées.

1. Écrire une fonction `firstdigit(n)` qui pour un nombre `n` donné, retourne son premier chiffre et une fonction `occurences(liste)` qui retourne le nombre d'occurrence des premiers chiffres de `liste`.
2. Vérifier si la loi de Benford semble satisfaite pour la suite des nombres $(2^n)_{n\in\mathbb{N}}$ en comparant l'histogramme empirique avec la loi de Benford.
3. Vérifier si la loi de Benford semble satisfaire pour la suite des nombres $(3n+1)_{n\in\mathbb{N}}$.
4. En allant sur le site de l'INSEE à l'adresse `https://insee.fr/fr/statistiques/4171341?sommaire=4171351`, télécharger le fichier au format TXT contenant les données de la population par sexe et âge regroupé (POP1A). Importer ces données pour avoir la popumation par code postal, sexe et tranche d'âge.
5. Déterminer si la liste de toutes les populations par commune, sexe et âge suit la loi de Benford.
6. Sommer les données précédentes pour obtenir la liste des populations par commune et déterminer si elle suit la loi de Benford.