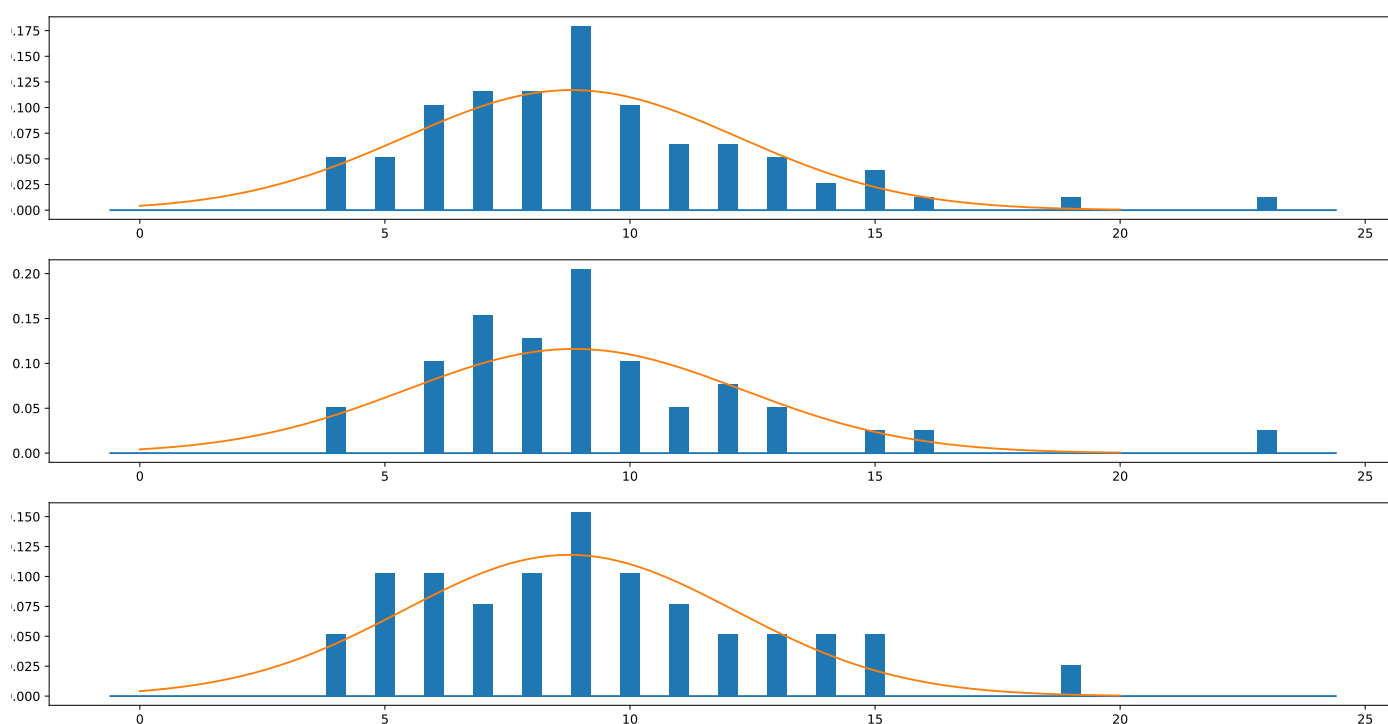


STATISTIQUES ET REPRÉSENTATIONS GRAPHIQUES

OBJECTIF : L'objectif de ce tp est de rendre l'élève capable :

- de charger des données à partir d'un fichier de type `.csv`
- regrouper en lots des données brutes
- déterminer la moyenne, la variance et l'écart type d'une série statistique
- déterminer la fréquence d'apparition d'éléments d'une série statistique
- représenter la série statistique sous forme d'histogramme
- modéliser la répartition statistique par une loi normale



1 Données brutes

Les notes des deux classes de PCSI au DS-Commun-1 de SI sont regroupées dans un fichier `DS-commun-1.csv`. Pour chaque ligne, la première colonne donne le nom de la classe d'un élève ; la deuxième colonne donne la note de cet élève.

REMARQUE : les notes sont dans un ordre plus ou moins aléatoire.

Q - 1 : Construire deux listes `classes` et `notes` contenant respectivement la liste des noms de classes et la liste des notes ordonnées identiquement au fichier `.csv`.

Q - 2 : Regrouper les notes des deux classes dans une variable `PCSI`, puis dans la variable `PCSI_1` les notes des PCSI-1 et dans la variable `PCSI_2` les notes des PCSI-2.

2 Répartition des notes

2.1 Éléments statistiques

On appelle moyenne \bar{x} , variance \mathcal{V} et écart-type σ d'une série statistique $(x_i)_0^{n-1}$ (liste de valeurs dans \mathbb{R}), les expressions données par les relations suivantes :

$$\bar{x} = \frac{1}{n} \sum_{i=0}^{n-1} x_i \quad ; \quad \mathcal{V} = \frac{1}{n} \sum_{i=0}^{n-1} (x_i - \bar{x})^2 = \left(\frac{1}{n} \sum_{i=0}^{n-1} x_i^2 \right) - 2 \cdot \bar{x} \cdot \left(\frac{1}{n} \sum_{i=0}^{n-1} x_i \right) + \bar{x}^2 = \left(\frac{1}{n} \sum_{i=0}^{n-1} x_i^2 \right) - \bar{x}^2 \quad \text{et} \quad \sigma = \sqrt{\mathcal{V}}$$

Q - 3 : Écrire une fonction *moyenne* qui prend en argument une liste *L* de flottants et renvoie la moyenne des valeurs de *L*.

Q - 4 : Écrire une fonction *variance* qui prend en argument une liste *L* de flottants et renvoie la variance des valeurs de *L*.

Q - 5 : Écrire une fonction *ecart_type* qui prend en argument une liste *L* de flottants et renvoie l'écart-type des valeurs de *L*.

Q - 6 : Écrire une fonction *stat* qui prend en argument une liste *L* de flottants et renvoie un tuple formé, à partir de *L*, de son nombre d'élément, de sa moyenne, de sa variance et de son écart-type des valeurs de *L*.

Pour chaque entier *i* compris entre *note_min* et *note_max*, on cherche à connaître le nombre d'éléments d'une liste *L* dont l'arrondi à l'entier supérieur vaut *i*.

Q - 7 : Écrire une fonction *frequence* dont la première ligne est donnée ci-dessous :

```
def frequence(L, note_min=0, note_max=23):
```

On donne ci-dessous deux cas tests :

```
>>> frequence([1.4, 4.5, 3, 4.2])
[0, 0, 1, 1, 0, 2, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
>>> frequence([1.4, 4.5, 3, 4.2], note_min = 2, note_max = 6)
[1, 1, 0, 2, 0]
```

On note *hist*, *hist_1* et *hist_2* les listes renvoyées par la fonction *frequence* appliquée aux listes *PCSI*, *PCSI_1* et *PCSI_2*.

Q - 8 : Affecter à *hist*, *hist_1* et *hist_2* les données attendues.

2.2 Représentations graphiques

Q - 9 : Tracer, sur une même figure, les évolutions de *hist*, *hist_1* et *hist_2*.

On souhaite présenter les évolutions des listes *hist*, *hist_1* et *hist_2* sous forme d'histogrammes.

Dans un premier temps, on construit une fonction *aff_hist*(*L*, *lx* = 0.2, *offset* = 0) prenant en argument une liste *L* et deux arguments optionnels *lx* et *offset*, permettant de tracer pour chaque entier à partir

de `offset`, un profil rectangulaire de largeur $2 \cdot lx$ et de hauteur $L[i]$ où i est le i ème élément de la liste L .

Q - 10 : Établir le script d'une solution pour `aff_hist`.

Q - 11 : Tracer, dans trois graphiques différents d'une même figure, les évolutions sous formes d'histogrammes des répartitions de notes des listes `PCSI`, `PCSI_1` et `PCSI_2`.

Le module `matplotlib.patches` permet de dessiner des formes géométriques.

Q - 12 : Importer la méthode `Rectangle` du module `matplotlib.patches`.

Un rectangle de largeur lx et de hauteur ly est tracé à partir de son coin bas gauche en (x, y) avec la commande :

```
ax.add_patch(Rectangle((x, y), lx, ly))
```

où `ax` est un pointeur vers le graphique :

```
fig, ax = plt.figure() # à partir d'une figure
ax = plt.subplot(3, 1, 1) # à partir d'un graphique d'une figure
```

Q - 13 : Écrire une fonction `aff_hist_rect` permettant d'améliorer la fonction `aff_hist_rect`.

REMARQUE : je n'ai pas trouvé comme faire apparaître les rectangles sans tracer le moindre segment au préalable. Penser à tracer une droite allant de $(note_min, 0)$ à $(note_max, 0)$.

Q - 14 : Tracer avec fonction la `aff_hist_rect`, dans une figure `Rectangles`, les évolutions des répartitions de notes des listes `PCSI`, `PCSI_1` et `PCSI_2` dans trois graphiques différents.

3 Loi normale

Une loi normale est une loi de probabilité qui dépend de deux paramètres : son espérance, un nombre réel noté μ , et son écart type, un nombre réel positif noté σ . La densité de probabilité f de la loi normale d'espérance μ , et

d'écart type σ est donnée par :
$$f(x) = \frac{1}{\sigma \cdot \sqrt{2 \cdot \pi}} \cdot e^{-\frac{1}{2} \cdot \left(\frac{x - \mu}{\sigma}\right)^2}$$

Q - 15 : Écrire une fonction `f` prenant en argument trois flottants, une variable x , l'espérance et l'écart-type.

Q - 16 : Tracer, dans une figure `lois normales` et dans trois graphiques différents, les évolutions de la densité de probabilité des lois normales obtenues avec l'écart type et pour espérance la moyenne, des notes contenues dans les listes `PCSI`, `PCSI_1` et `PCSI_2`.

Q - 17 : Superposer aux courbes précédentes les histogrammes de répartitions de notes, normés par le nombre de notes.

4 Autres représentations graphiques

Le module `matplotlib.pyplot` possède une méthode `hist`.

Q - 18 : Prendre en main `hist` et déterminer comment fonctionne la méthode.

Le module `matplotlib.patches` contient différentes méthodes pour dessiner des formes.

Q - 19 : Laisser libre cours à votre imagination de statisticien pour faire de jolis graphes.