Université Claude Bernard - Lyon 1 **Semestre d'automne 2022–2023**

L2 - Informatique UE : Statistiques pour l'Informatique

Feuille TP 1

Dans ce TP, on utilisera les bibliothèques suivantes [1] :

```
import numpy as np
import matplotlib.pyplot as plt
import pandas as pan
import scipy.stats.mstats as ms
```

Exercice TP1.1 Représentation graphique d'une variable quantitative.

En `Python` *il y a un certain nombre de bases de données disponibles dans la librairie* `PyDataset`*, pour en voir la liste, il suffit de taper*

```
pip install pydataset#uniquement la 1ère fois, ou si la commande suivante renvoie "No
    module named 'pydataset'"
from pydataset import data
pydata=data()#En cliquant sur l'objet pydata dans l'explorateur de variable de Spyder
    , on voit la liste. Alternativement data() seule montre quelques éléments
```

À partir du jeu de données `cars` disponible dans `PyDataset` (utiliser `cars=data('cars')` pour le charger, et `data('cars', show_doc=True)` pour obtenir une description des données) :

1. Calculer la moyenne, variance empirique, variance empirique non-biaisée, médiane, écart-type des variables `speed` et `dist`.
2. Calculer les 3 quartiles selon la définition du cours.
3. Que fait la fonction `cars.describe()` ?
4. Selon le type de variables à notre disposition, quelle représentation graphique suggérez-vous ? Tracer l'histogramme de la variable `speed`. Ajouter un titre, et des étiquettes aux axes x et y.

Exercice TP1.2 Représentation graphique d'une variable qualitative.

1. Charger le jeu de données `iris`.
2. Tracer le diagramme circulaire pour la variable qualitative. Calculer la moyenne, variance empirique, variance non-biaisée, le minimum et le maximum pour les variables quantitatives. Calculer la moyenne et la variance non-biaisée par type de `Species`. Que remarquez vous ? Quelle est l'autre façon de représenter une variable qualitative ?
3. Tracer la boite à moustache de `Sepal.Length` en fonction de `Species`. Quel est le but de cette représentation ?

Exercice TP1.3 Rappels sur les vecteurs de `numpy`.

1. Créer le vecteur $x = (1, 8, 5, 1)$ grâce à la commande `np.array`.
2. Créer le vecteur $y = (0, 1, 3, 5, 7, 9)$ en utilisant `np.array`, `range` et `np.concatenate`.
3. Étudier les résultats des commandes `y[4]`, `y[2:4]`, `y[-2]` et `y[y<=5]`.
4. Extraire les éléments en position paire de y. Extraire les éléments plus grands que 1 de y.
5. Conserver tous les éléments de y, sauf le premier.
6. A l'aide de les commandes `np.repeat()` et `np.reshape()`, créer un vecteur X de taille 100 obtenu en mettant bout à bout 25 copies de x. (Donc, X commence ainsi $X = (1, 8, 5, 1, 1, 8, 5, 1, \cdots)$)

1. Une feuille de TP0 d'introduction à `Python` est disponible sur claroline. Si vous ne connaissez pas Python, lisez-la en détail et testez les commandes qui sont présentées, avant de faire les exercices suivants.

Exercice TP1.4 On considère les vecteurs de nouveau $x = (1, 8, 5, 1)$ et $y = (0, 1, 3, 5, 7, 9)$.

1. Pourquoi la commande `plt.plot(x,y)` retourne-t-elle une erreur ?
2. Ajouter $(3, 5)$ au vecteur x et représenter le nuage des points (x_i, y_i).
3. Ajouter le point $(\overline{x}, \overline{y})$ en rouge en utilisant la commande `plt.plot`.
4. Ajouter la droite de régression en utilisant `plt.axline` et `ms.linregress`.
5. Calculer la corrélation empirique `cor(x,y)` pour décider si une approximation par une droite est raisonnable.

Exercice TP1.5

1. Charger le jeu de données `women`. Représenter les deux variables "taille" et "poids" par une nuage de points, avec la droite de régression du poids en fonction de la taille.
2. Discuter si cette approximation est raisonnable. Comment interprète-t-on un point qui se trouve nettement au-dessus/au-dessous de la droite de régression ?

Exercice TP1.6 Charger le jeu de données suivant, en utilisant les commandes :

```
Don=pan.read_csv("http://math.univ-lyon1.fr/~dabrowski/Donnees.csv",sep="\t")
```

Les données correspondent à l'âge, au poids, à la taille, à la consommation hebdomadaire d'alcool (en nombre de verres bus), au sexe, au ronflement et au tabagisme, d'un échantillon de 100 personnes. On récupère (par exemple) les données concernant l'âge et le ronflement, de la façon suivante.

```
age = Don["age"]; ronfle = Don["ronfle"]
```

On peut aussi faire une petite boucle pour nommer toutes les variables.

```
for nom in Don.keys():
    globals()[nom] = df[nom]
```

1. Commencer par identifier les variables qualitatives nominales, ordinales et quantitative discrètes et continues. Typer les données qualitatives correctement en *Python* (on pourra utiliser les commandes `astype`) et renommer les niveaux avec la variable `cat.categories`.

   Remarque 1 *Les deux types fondamentaux de* `pandas` *sont le DataFrame (*`Don` *en est un) et la Series (*`Don["age"]`*, etc., en sont). Un DataFrame est un tableau de données avec des colonnes aux noms arbitraires représentant des variables de n'importe quel type. Une Series est une colonne d'un DataFrame, c'est-à-dire une variable statistique.*

2. Trouver la corrélation entre poids et taille. Que retournent les commandes suivantes ?

   ```
   df=pan.DataFrame({"age":Don['age'],"
       poids":Don['poids'],"taille":Don[
       'taille']})
   df.cov();df.corr()
   import seaborn as sns
   sns.pairplot(df)
   ```

   Quelles sont les variables continues les plus corrélées entre elles ?

3. Tracer un nuage de points du poids en fonction de la taille, calculer la droite de régression de ces deux variables et l'ajouter en rouge au nuage de points. Discuter si cette approximation est raisonnable. Avez-vous un commentaire sur cet échantillon de données ?
4. Tracer sur un même graphique les diagrammes à moustaches de age, poids et taille.
5. Calculer la table de contingence des fréquences de ronfle et tabac. Quel est le mode du couple (tabac,ronfle) ?
6. Tracer les fonctions de répartition empirique de alcool et poids. Le diagramme en escalier vous paraît-il pertinent pour les deux ? Indication : `from statsmodels.distributions.empirical_distribution import ECDF`