Feuille TP 1: Représentations graphiques

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
```

Dans les exercices suivants, on s'intéresse à une sélection des résultats recueillis dans une étude sur l'alimentation d'un échantillon de personnes âgées résidant à Bordeaux (Gironde, France), interrogées en 2000 dans le cadre d'une enquête nutritionnelle [2]. L'échantillon est constitué de 226 sujets. Charger le jeu de données suivant, en utilisant les commandes :

```
df=pan.read_csv("https://math.univ-lyon1.fr/~dabrowski/nutriage.csv",sep="\t")
```

Remarque 1 *Les deux types fondamentaux de* `pandas` *sont le* `DataFrame` *(par exemple* `df` *) et la* `Series` *(comme* `df["age"]`*). 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.*

Exercice TP1.1 Le tableau ci-contre explique le codage des données.

1. Pour chaque variable statistique, indiquez sa nature (qualitative ou quantitative, discrète ou continue).
2. On peut indiquer à *pandas* que le sexe est une variable nominale (en anglais *categorical variable*) puis changer le nom de ses caractères par les commandes suivantes

```
df['sexe']=df['sexe'].astype('category')
df['sexe'].cat.categories=["Femme","Homme
    "]#une liste avec les noms voulus pour
     les caractères de la variable
```

Faites ainsi pour toutes les variables qualitatives

Description	Unité ou Codage	Variable
Sexe	F=Femme; H=Homme	sexe
Consommation journalière de thé	Nombre de tasses	the
Consommation journalière de café	Nombre de tasses	cafe
Taille	cm	taille
Poids	kg	poids
Age le jour de l'entretien	Années	age
Consommation de viande	0=Jamais 1=Moins d'une fois par semaine 2=Une fois par semaine 3=2/3 fois par semaine 4=4/6 fois par semaine 5=Tous les jours	viande
Consommation de poisson	Idem	poisson
Matière grasse préférentiellement utilisée pour la cuisson	1=Beurre 2=Margarine 3=Huile d'arachide 4=Huile de tournesol 5=Huile d'olive 6=Mélange d'huile (type Isio4) 7=Huile de colza 8=Graisse de canard ou d'oie	matgras

3. Typez de même les variables quantitatives discrètes comme `int64` et continues comme `float64`.

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.
2. Les données sont extraits du livre *Le logiciel R : maîtriser le langage effectuer des analyses statistiques* de Pierre Lafaye de Micheaux, Rémy Drouilhet et Benoit Liquet aux éditions Springer et disponible à la page :

   `http://www.biostatisticien.eu/springeR/jeuxDonnees4.html`

Exercice TP1.2 On cherche à représenter graphiquement les variables qualitatives.

1. A l'aide de la fonction `pan.crosstab`[3], calculer un tableau de contingence des effectifs des variables `sexe` et `matgras`. Combien de Femmes de l'échantillon utilisaient de l'huile de Tournesol ?
2. Calculer un tableau de contingence des fréquences empiriques des mêmes variables, en ajoutant les fréquences marginales (c'est-à-dire de `sexe` et `matgras` séparément). Quelle est la proportion de personnes interrogées de l'échantillon qui utilisaient de l'huile de Tournesol ?
3. En utilisant `plt.pie`, tracer un diagramme circulaire pour `matgras`.
4. En utilisant `plt.bar`, tracer un diagramme en tuyau d'orgue pour `viande`.
5. En utilisant `plt.plot` et `np.cumsum`, tracer une courbe des fréquences cumulées pour `viande`.
6. Rassembler le diagramme en tuyau d'orgue et celui des fréquences cumulées pour `viande` sur le même graphique, on pourra utiliser ce schéma :

```
fig, ax = plt.subplots();
#ax. remplace plt. dans une commande graphique
ax2=ax.twinx()
#ax2. remplace plt. dans une commande graphique graduée sur l'axe de droite
```

Exercice TP1.3 On cherche à représenter graphiquement les variables quantitatives.

1. On représente les variables discrètes par des diagrammes en bâton (similaire à ceux en tuyaux d'orgue mais de largeur assez fine pour suggérer que la valeur de l'abscisse est exacte). Tracer les diagrammes en bâton pour `cafe` (on pourra utilise l'option `plt.plot(...,width=0.5)`).
2. On représente les fréquences cumulées d'une variable discrète par une fonction en escalier (c'est à dire constante par morceau, continue à droite qui correspondra à la *fonction de répartition* vue plus tard). En utilisant `plt.step`, tracer les fréquences cumulées de `the`.
3. Avec `plt.hist`, tracer un histogramme du `poids` avec un pas de subdivision uniforme de 5kg.
4. Tracer un histogramme du `age` avec un pas de subdivision uniforme de 2 ans entre 65 et 83 ans, puis deux intervalles plus grands $[83, 87]$, et $[87, 91]$.
5. Tracer un nuage de point représenter pour chaque individus le poids en fonction de la taille. Ajouter un titre, et des étiquettes aux axes x et y.
6. En utilisant `df.boxplot`, tracer un diagramme à moustache de `age,poids,taille` sur le même graphique. Puis de `the,cafe` sur un autre. Commenter.

Exercice TP1.4 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 $X = (1, 8, 5, 1, 1, 8, 5, 1, \cdots)$)

3. On pourra trouver de l'aide avec la commande `help(pan.crosstab)`.