TP 1 – Analyse de données

A. 2)

```
#2. Execussion des commandes "df.head" et "df.shape"
print("Résultat concernant la commande df.head")
print(df.head())
print("Résultat concernant la commande df.shape")
print(df.shape)
```

```
Résultat concernant la commande df.head
                Species  Diet    BOW      BRW    AUD     MOB     HIP
0  Rousettus aegyptiacus     1  136.3  2070.00   9.88  105.77  125.97
1      Epomops franqueti     1  120.0  2210.00  10.44  107.80  159.80
2     Eonycteris spelaea     1   58.7  1310.00   5.48   67.00   97.70
3      Cynopterus sphinx     1   48.3  1184.33   4.77   65.27   95.40
4    Dobsonia praedatrix     1  184.0  3028.00   7.09  213.43  233.30
Résultat concernant la commande df.shape
(29, 7)
```

La commande « head() » affiche uniquement les 5 premières lignes du tableau CSV
L'attribut « shape » affiche les dimensions du tableau de données. Ici, nous avons 29 lignes et 7 colonnes

A. 3)

```
#3. Calculs mathématiques
print("Poids moyen d'une chauve souris")
print(np.mean(df.BOW))
print("Variance")
print(np.var(df.BOW))
print("Ecart type")
print(sqrt(np.var(df.BOW)))
print("Poids médian du cerveau d'une chauve souris")
print(np.median(df.BRW))
```

Ce qui donne :

```
Poids moyen d'une chauve souris
87.41896551724139
Variance
34436.17685755053
Ecart type
185.56987055432927
Poids médian du cerveau d'une chauve souris
814.0
```

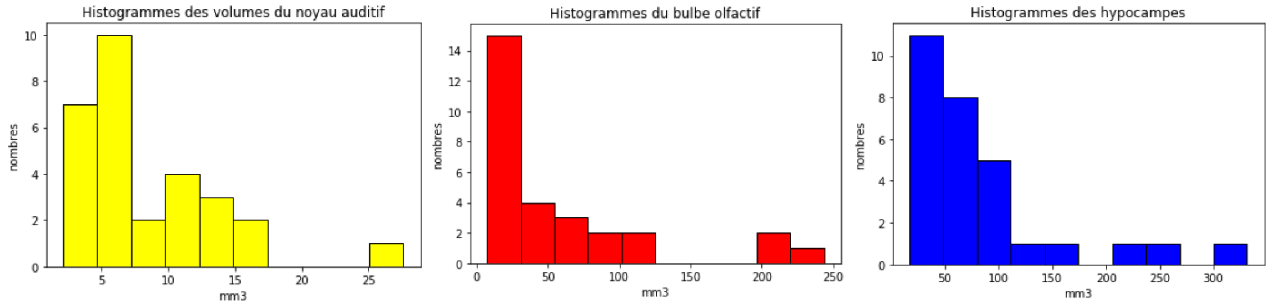
Le résultat de la variance paraît absurde mais je n'arrive pas à comprendre pourquoi

A. 4)

Pour réaliser les histogrammes, on utilise la fonction « plt.hist() » . A l'aide des attributs, on rend les histogrammes plus lisibles :

```
#4. Histogrammes
plt.hist(df.AUD, color = 'yellow', edgecolor = 'black')
plt.title('Histogrammes des volumes du noyau auditif')
plt.xlabel('mm3')
plt.ylabel('nombres')
plt.show()
plt.hist(df.MOB, color = 'red', edgecolor = 'black')
plt.title('Histogrammes du bulbe olfactif')
plt.xlabel('mm3')
plt.ylabel('nombres')
plt.show()
plt.hist(df.HIP, color = 'blue', edgecolor = 'black')
plt.title('Histogrammes des hypocampes')
plt.xlabel('mm3')
plt.ylabel('nombres')
plt.show()
```

Ce qui donne les histogrammes suivants :

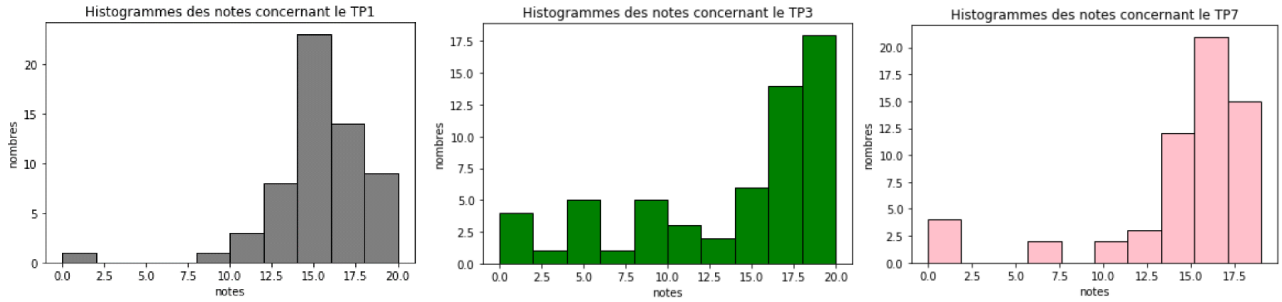


B. 1) Il y a un problème avec les virgules dans le fichier csv. On remplace donc toute les virgules par des points (à l'aide de la fonction remplacement sur Excel). Par ailleurs, il y a des valeurs parasites en dehors du tableau que l'on supprime.

B. 2)

```
#2. Histogrammes
plt.hist(nt.Lab1, color = 'grey', edgecolor = 'black')
plt.title('Histogrammes des notes concernant le TP1')
plt.xlabel('notes')
plt.ylabel('nombres')
plt.show()
plt.hist(nt.Lab3, color = 'green', edgecolor = 'black')
plt.title('Histogrammes des notes concernant le TP3')
plt.xlabel('notes')
plt.ylabel('nombres')
plt.show()
plt.hist(nt.Lab7, color = 'pink', edgecolor = 'black')
plt.title('Histogrammes des notes concernant le TP7')
plt.xlabel('notes')
plt.ylabel('nombres')
plt.show()
```

Ce qui donne les histogrammes suivants :



A l'aide de Python, on affiche la note moyenne, minimale, maximale et l'écart-type pour les projets :

```
#2.2 Calculs
print("Valeur moyenne du projet")
print(np.mean(nt.Project))
print("Valeur minimale du projet")
print(min(nt.Project))
print("Valeur maximale du projet")
print(max(nt.Project))
print("Ecart type du projet")
print(sqrt(np.var(nt.Project)))
```

```
Valeur moyenne du projet
12.533898305084746
Valeur minimale du projet
0.0
Valeur maximale du projet
16.0
Ecart type du projet
3.2192014047855717
```

Enfin, on affiche un diagramme en camembert pour le GPA :

```
#2.3 Diagramme pour le GPA
na = 0
nb = 0
nc = 0
nd = 0
ne = 0
nf = 0
for i in nt.GPA:
  if i == "A":
    na += 1
  elif i == "B":
    nb += 1
  elif i == "C":
    nc += 1
  elif i == "D":
    nd += 1
  elif i == "E":
    ne += 1
  elif i == "F":
    nf += 1

X=[na, nb, nc, nd, ne, nf]
plt.pie(X, labels = ['A', 'B', 'C', 'D', 'E', 'F'])
plt.legend()
```

J'utilise les valeurs na, nb, nc, nd, nd comme des compteurs pour connaître le nombre de «A», «B», «C», «D», «E», «F» dans le tableau de données mais j'aurais pu aussi utiliser le commande "value_counts ()".

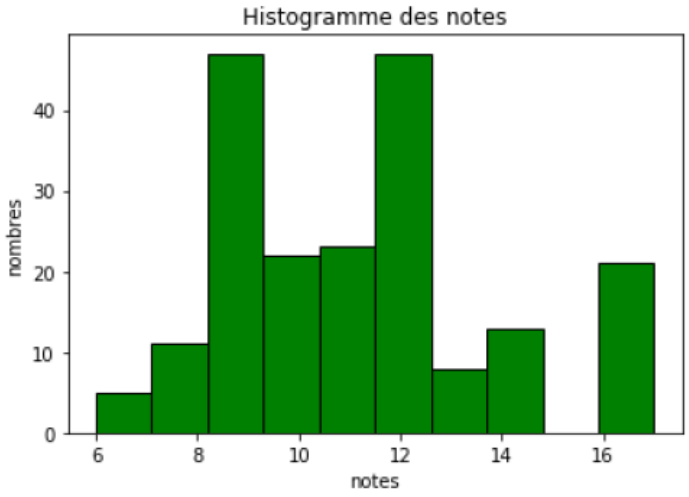
B. 3)

```
moyenne du TP1 avec l'outil mean
15.021186440677965
moyenne du TP2 avec l'outil mean
16.66949152542373
moyenne du TP3 avec l'outil mean
13.635593220338983
moyenne du TP4 avec l'outil mean
12.443247126551723
moyenne du TP5 avec l'outil mean
14.84322033898305
moyenne du TP6 avec l'outil mean
14.455508474576272
moyenne du TP6 avec l'outil mean
14.440677966101696
Ecart type de tout les TPs avec l'outil sqrt(var())
1.1999433122928502
moyenne de tout les TPs avec l'outil mean
14.501275013236201
moyenne de l'Exam final avec l'outil mean
10.679824561403509
```

On voit que les TP ont des moyennes assez élevées, autour de 14, avec un léger écart type.
C'est bien au-dessus des résultats des examens finaux avec une moyenne de 10,68.

C) 1. On crée le jeu de données correspondant et on l'affiche sous forme d'un histogramme :

```
#Histogramme
Note = [6, 8, 9, 10, 11, 12, 13, 14, 17]
Nb = [10, 12, 48, 23, 24, 48, 9, 14, 22]
X=[]
for i in range(5):
  X.append(6)
for i in range(11):
  X.append(8)
for i in range(47):
  X.append(9)
for i in range(22):
  X.append(10)
for i in range(23):
  X.append(11)
for i in range(47):
  X.append(12)
for i in range(8):
  X.append(13)
for i in range(13):
  X.append(14)
for i in range(21):
  X.append(17)
plt.hist(X, color = 'green', edgecolor = 'black')
plt.xlabel('notes')
plt.ylabel('nombres')
plt.title('Histogramme des notes')
plt.show()
```



On aurait pu créer un fichier csv à l'aide d'Excel pour créer l'ensemble de données, mais la taille des données est assez faible donc pas forcément utile.

C) 2. La moyenne est de 11,27, la médiane est de 11, le mode est égal à 9 et 12, l'écart type est égal à 2,64, la variance est égale à 6,96.
La différence entre le maximum et le minimum est de 11.

C) 3. Cette série est une distribution bimodale car elle a deux modes : 9 et 12.

D) 1. On importe le fichier csv :

```
#Exercice D

#1. Importation du ficher csv
mt = pd.read_csv("malnutrition.csv", usecols=[0])
```

D) 2. On calcule le nombre de personne dans l'échantillon :

```
#2. Calcul du nombre de personne dans l'échantillon
print("Nombre de personne dans l'échantillon")
print(len(mt)+1)
```

Ce qui nous donne 100 personnes dans l'échantillon :

```
Nombre de personne dans l'échantillon
100
```

D) 3. On calcule avec les fonctions « mean » et « sqrt(var) » la moyenne et l'écart-type de cet échantillon. On trouve un écart-type de 9.63 et un QI moyen de 87.98.

D) 4. Les personnes qui ne font pas attention à leur alimentation ont un QI moyen de 10% inférieur à celui des autres. La malnutrition a donc un effet sur le QI des gens.
La dispersion est plus faible car l'écart type passe de 15 à 10.