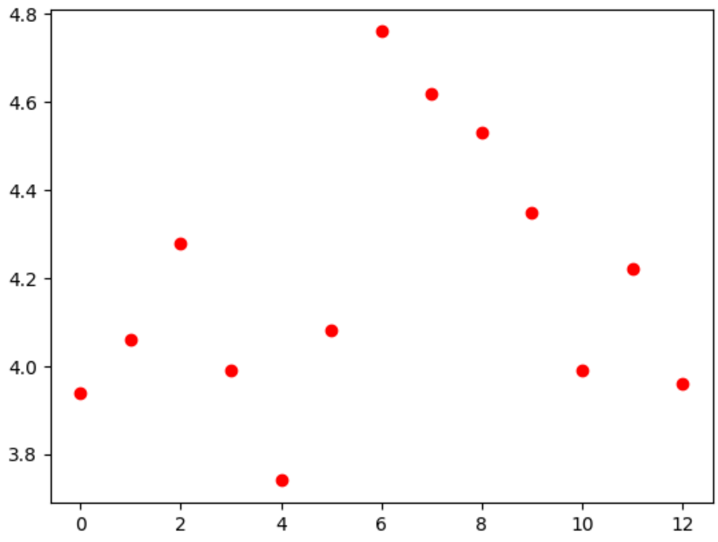
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

Commencez à coder ou à générer avec l'IA.

Exercice 1 : Serie Statistique

```
#data
concentrations = pd.Series([3.94,4.06,4.28,3.99,3.74,4.08,4.76,4.62,4.53,4.35,3.99,4.22,3.96])
n=len(concentrations)
plt.figure()
plt.scatter(np.arange(0,n),concentrations,color='r')
plt.show()
```



1. Calculer la moyenne empirique et la variance

On calcule la moyenne empirique en utilisant la formule suivante:

$$\begin{aligned}\bar{X} &= \frac{1}{n}\sum_{i=1}^{n} X_i \\ &= \frac{1}{13}\sum_{i=1}^{13} X_i \\ &= \frac{1}{13}(3.94+4.06+4.28+3.99+3.74+4.08+4.76+4.62+4.53+4.35+3.99+4.22+3.96) \\ &= 4.193846153846154\end{aligned}$$

où $n = 13$ est le nombre d'individus et $(X_1, \ldots, X_n)$ sont les observations. On calcule la variance en utilisant la formule suivante:

$$\begin{aligned}Var(X) &= \frac{1}{n}\sum_{i=1}^{n}(X_i^2) - \left(\frac{1}{n}\sum_{i=1}^{n} X_i\right)^2 \\ &= \frac{1}{n}\sum_{i=1}^{n}(X_i^2) - (\bar{X})^2 \\ &= 0.083592899408284\end{aligned}$$

```
#Moyenne et Variance empiriques
n=len(concentrations)
x_bar = 1/len(concentrations) * concentrations.sum()
Var_X = 1/len(concentrations) * concentrations.pow(2).sum() - (1/len(concentrations) * concentrations.sum())**2
print('Moyenne =', x_bar)
print('Variance=', Var_X)
```

```
Moyenne = 4.193846153846154
Variance= 0.08359289940828418
```

2.Calcul de la mediane et des quantiles

Calcul de la médiane

Il faut tout d'abord classer les valeurs par ordre croissant. Dans notre cas içi

$$\begin{array}{ll} 3.74 \leq 3.94 \leq 3.96 \leq 3.99 \leq 3.99 \leq 4.06 \leq 4.08 \leq 4.22 \leq 4.28 \leq 4.35 \leq 4.53 \leq 4.62 & \leq 4.76 \\ X_{(1)} \leq X_{(2)} \leq X_{(3)} \leq X_{(4)} \leq \cdots & \leq X_{(13)} \end{array}$$

Pour calculer la médiane (M) il y a deux cas de figures en fonction de la parité de $n = 13$

- Le nombre d'individus est pair (il s'écrit sous la forme $n = 2 \times \ell$). Dans ce cas la mediane est la moyenne en la ℓ-eme valeur et la $\ell + 1$ème valeur. (Ce n'est pas le cas dans notre exemple)
$$M = \frac{X_\ell + X_{\ell+1}}{2}.$$
- Le nombre d'individus est impair (il s'écrit sous la forme $n = 2 \times \ell + 1$). Dans ce cas la mediane est donnée par le $\ell + 1$-eme valeure (Ce n'est pas le cas dans notre exemple)

Pour nous $n = 2 * 6 + 1$ donc $\ell = 6$, la mediane est la 7ème valeur $X_{(7)}$.
$$M = X_{(7)} = 4.08$$

Calcul des autres quantiles et deciles

On notera

- D_1 le premier decile à $10\% = 0.1$
- Q_1 le premier quartile à $25\% = 0.25$
- Q_3 le troisieme quartile $75\% = 0.75$
- D_9 l'avant dernier decile $90\% = 0.9$

Ce sont tous des quantiles d'ordre $\alpha = 0.1, 0.25, 0.75, 0.9$. La mediane est le quantile d'ordre $\alpha = 0.5$

Methode: Pour calculer un quantile quand on a une serie statistique courte comme içi on peut se contenter de calculer la position p dans la liste triée grace à la formule suivante
$$p = \alpha \times (n + 1)$$

- *Pour* D_1: On calcule $p = 0.1 \times (13 + 1) = 1.4$ le decile D_1 se trouve alors entre la 1ère et la 2ème valeur. On prendra la valeur supérieure
$$D_1 = 3.94$$
- *Pour* Q_1: On calcule $p = 0.25 \times 14 = 3.5$ donc Q_1 se trouve alors entre la 3ère et la 4ème valeur. On prendra la valeur supérieure
$$Q_1 = 3.99$$
- *Pour* Q_3: On calcule $p = 0.75 \times 14 = 10.5$ le decile Q_3 se trouve alors entre la 10ère et la 11ème valeur. On prendra la valeur supérieure
$$Q_3 = 4.53$$
- *Pour* D_9: On calcule $p = 0.9 \times 14 = 12.6$ le decile D_9 se trouve alors entre la 12ère et la 13ème valeur. On prendra la valeur supérieure
$$D_9 = 4.76$$

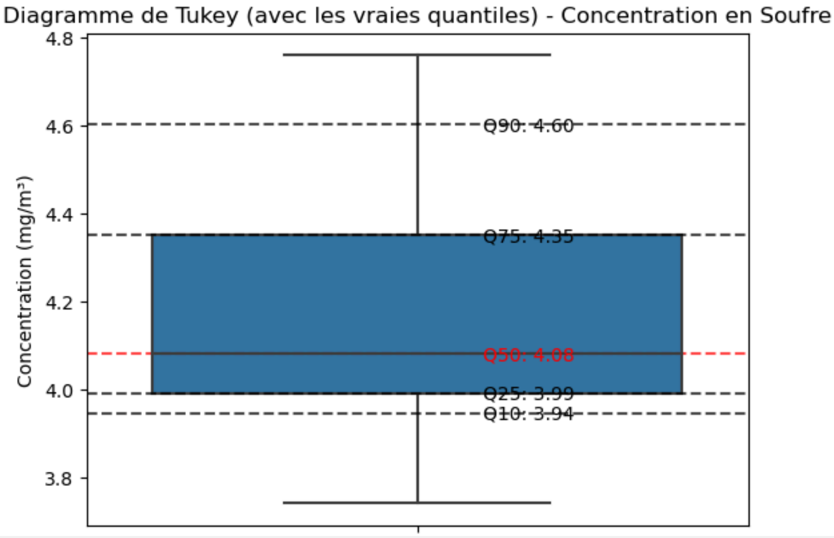
Une autre methode est possible en utilisant les fréquences cumulées qu'on verra à l'exercice suivant

** Cela ne correspond pas au vrai quantiles mais seulement à une estimation. Une version la plus précise du quantile est obtenue par interpolation linéaire qu'on verra à la partie 2 de l'exercice suivant**

```
# Mediane/Quartiles
sorted_concentrations= concentrations.sort_values().set_axis(np.arange(1,14,1),axis=0) # trier les valeurs
print('Tableau trié')
print(sorted_concentrations)
```

```
Tableau trié
1     3.74
2     3.94
3     3.96
4     3.99
5     3.99
6     4.06
7     4.08
8     4.22
9     4.28
10    4.35
11    4.53
12    4.62
13    4.76
dtype: float64
```

```
# Afficher le diagram de Tuckey avec les quantiles
plt.figure('Tuckey Diagram')
sns.boxplot(y=concentrations)
plt.title("Diagramme de Tukey (avec les vraies quantiles) - Concentration en Soufre")
plt.ylabel("Concentration (mg/m³)")
quantiles = concentrations.quantile([0.10, 0.25, 0.50, 0.75, 0.90])

for q, value in quantiles.items():
    plt.axhline(y=value, color="red" if q == 0.50 else "black", linestyle="--", alpha=0.7)
    plt.text(0.1, value, f"Q{int(q*100)}: {value:.2f}", va="center", ha="left", fontsize=10, color="red" if q == 0.50 else "black")
```



Exercice 2

Les taille des données est trop grande pour être présenter de façon extensive, on préfère exprimer les modalités

- Ici le nombre d'individus total $n = 89$ $X_1, X_2, \ldots, X_{89}$
- Le nombre de modalité est $k = 17$ qu'on notera avec des miniscules $x_1, x_2, \cdots, x_{17}$

```
# data
tailles=np.arange(52,69,1)
effectif=[1,0,3,2,3,8,7,11,11,10,6,14,6,3,2,1,1]
data_t= pd.DataFrame({'Taille': tailles, 'Effectifs': effectif},index=np.arange(1,18,1))
print(data_t)
```

```
    Taille  Effectifs
1       52          1
2       53          0
3       54          3
4       55          2
5       56          3
6       57          8
7       58          7
8       59         11
9       60         11
10      61         10
11      62          6
12      63         14
13      64          6
14      65          3
15      66          2
16      67          1
17      68          1
```

Partie I

Question 1

On calcule la moyenne, la variance, l'écart-type et le coefficient de variation.

- **Moyenne**. On utilise la formule suivante qui tire partie des modalités x_i et des effectifs n_i

$$\bar{x} = \sum_{i=1}^{k} x_i \frac{n_i}{n} = \left(52 \times \frac{1}{89} + 53 \times \frac{0}{89} + \cdots + 68 \times \frac{1}{68}\right) = 60.37$$

- **Variance**. Rappelons que les fréquences $f_i = \frac{n_i}{n} = \frac{n_i}{89}$

$$Var(x) = \sum_{i=1}^{k}(X_i - \bar{x})^2 f_i = \sum_{i=1}^{k}(X_i - \bar{x})^2 \frac{n_i}{n}$$
$$= \left((52 - 69.37)^2 \frac{1}{89} + \cdots + (68 - 69.37)^2 \frac{1}{89}\right) = 9.941$$

- **Ecart-Type**: La formule est donnée par

$$\sigma_x = \sqrt{Var(x)} = \sqrt{9.941} = 3.15$$

- **Coefficient de variation**: La formule est donnée par

$$\frac{\sigma_x}{\bar{X}} = \frac{3.15}{60.37} = 0.052$$

```
n=89
k=17
mean = np.sum(data_t['Taille']*data_t['Effectifs']/n)
var = np.sum((data_t['Taille']-mean)**2*data_t['Effectifs'])/n
print('Moyenne=', mean)
print("Variance =", var)
print("Ecar_Type= ", np.sqrt(var))
print("Coefficient de variation = ", np.sqrt(var)/mean)
```

```
Moyenne= 60.37078651685393
Variance = 9.941169044312584
Ecar_Type=  3.1529619478059967
Coefficient de variation =  0.05222661703977921
```

Question 2

Pour le calcul des quantiles on va rajouter à notre table les fréquences et les fréquences cumulées pour le calcul soit plus rapide.

```
data_t["Frequences"] = data_t["Effectifs"]/n
data_t["Frequences cumulées"]= np.cumsum(data_t["Frequences"])
print(data_t)
print('-'*70)
```

```
    Taille  Effectifs  Frequences  Frequences cumulées
1       52          1    0.011236             0.011236
2       53          0    0.000000             0.011236
3       54          3    0.033708             0.044944
4       55          2    0.022472             0.067416
5       56          3    0.033708             0.101124
6       57          8    0.089888             0.191011
7       58          7    0.078652             0.269663
8       59         11    0.123596             0.393258
9       60         11    0.123596             0.516854
10      61         10    0.112360             0.629213
11      62          6    0.067416             0.696629
12      63         14    0.157303             0.853933
13      64          6    0.067416             0.921348
14      65          3    0.033708             0.955056
15      66          2    0.022472             0.977528
16      67          1    0.011236             0.988764
17      68          1    0.011236             1.000000
----------------------------------------------------------------------
```

- **Mediane**: On observe à partir de quelle valeur la fréquence cumulée dépasse $\alpha = 0.5$, ici c'est à partir de la 9ème valeur donc
$$M = 60$$
- **D_1**: On observe à partir de quelle valeur la fréquence cumulée dépasse $\alpha = 0.1$, ici c'est à partir de la 5ème valeur donc
$$D_1 = 56$$
- **Q_1**: On observe à partir de quelle valeur la fréquence cumulée dépasse $\alpha = 0.25$, ici c'est à partir de la 7ème valeur donc
$$Q_1 = 58$$
- **Q_3**: On observe à partir de quelle valeur la fréquence cumulée dépasse $\alpha = 0.75$, ici c'est à partir de la 12ème valeur donc
$$Q_3 = 63$$
- **D_9**: On observe à partir de quelle valeur la fréquence cumulée dépasse $\alpha = 0.9$, ici c'est à partir de la 13ème valeur donc
$$D_9 = 64$$

Partie II

On comment par préparer le tableau de donné

- On regroupe les données dans les classes

$$\begin{array}{ccc}]50,57.5], &]57.5,60.5], &]60.5,63.5],]63.5,70] \\ ]y_0,y_1], &]y_1,y_2], &]y_2,y_3],]y_3,y_4] \end{array}$$

  Il y a donc $\ell = 5$ classes. Chaque classe a un effectif n_i qu'on peut calculer en sommant les effectifs des modalités x_i dans la classe.

```
#II
bins = [50, 57.5, 60.5, 63.5, 70]  # Define bin ranges
labels = [']50,57.5]', ']57.5,60.5]', ']60.5,63.5]', ']63.5,70]']  # Labels for bins
data_t['class'] = pd.cut(data_t["Taille"], bins=bins,labels=labels,right=True)
data=data_t.groupby('class',as_index=False).agg('sum')
data=data.drop(columns=['Taille','Frequences cumulées'])
data["Frequences cumulées"]=np.cumsum(data['Frequences'])
print(data_t)
print('-'*50)
print('Nouvelle Table')
print('-'*50)



print(data)
```

```
    Taille  Effectifs  Frequences  Frequences cumulées        class
1       52          1    0.011236             0.011236    ]50,57.5]
2       53          0    0.000000             0.011236    ]50,57.5]
3       54          3    0.033708             0.044944    ]50,57.5]
4       55          2    0.022472             0.067416    ]50,57.5]
5       56          3    0.033708             0.101124    ]50,57.5]
6       57          8    0.089888             0.191011    ]50,57.5]
7       58          7    0.078652             0.269663  ]57.5,60.5]
8       59         11    0.123596             0.393258  ]57.5,60.5]
9       60         11    0.123596             0.516854  ]57.5,60.5]
10      61         10    0.112360             0.629213  ]60.5,63.5]
11      62          6    0.067416             0.696629  ]60.5,63.5]
12      63         14    0.157303             0.853933  ]60.5,63.5]
13      64          6    0.067416             0.921348    ]63.5,70]
14      65          3    0.033708             0.955056    ]63.5,70]
15      66          2    0.022472             0.977528    ]63.5,70]
16      67          1    0.011236             0.988764    ]63.5,70]
17      68          1    0.011236             1.000000    ]63.5,70]
--------------------------------------------------
Nouvelle Table
--------------------------------------------------
         class  Effectifs  Frequences  Frequences cumulées
0    ]50,57.5]         17    0.191011             0.191011
1  ]57.5,60.5]         29    0.325843             0.516854
2  ]60.5,63.5]         30    0.337079             0.853933
3    ]63.5,70]         13    0.146067             1.000000
C:\Users\Jalal\AppData\Local\Temp\ipykernel_15464\623278393.py:5: FutureWarning: The default of observed=False is deprecated and wi
  data=data_t.groupby('class',as_index=False).agg('sum')
```

- On calcule le centre des classes $c_i = \frac{y_{i-1}+y_i}{2}$ pour chaque classe

Question 1

On calcule la moyenne, la variance, l'écart-type et le coefficient de variation.

- **Moyenne**. On utilise la formule suivante qui se base le centre des classes c_i et les effectifs de chaque classe n_i (où la fréquence $f_i = \frac{n_i}{n}$)

$$\begin{aligned} \bar{x}_c &= \sum_{i=1}^{\ell} c_i \frac{n_i}{n} = \sum_{i=1}^{\ell} c_i f_i \\ &= (53.75 \times 0.19 + 59 \times 0.32 + 62 \times 0.33 + 66.75 \times 0.14) = 60.14 \end{aligned}$$

- **Variance**.

$$Var(x) = \sum_{i=1}^{\ell} (c_i - \bar{x}_c)^2 f_i = 15.77$$

- **Ecart-Type**: La formule est donnée par

$$\sigma_x = \sqrt{Var(x)} = \sqrt{15.77} = 3.97$$

- **Coefficient de variation**: La formule est donnée par

$$\frac{\sigma_x}{\bar{X}} = \frac{3.97}{60.14} = 0.066$$

```
#II.1
bins=np.array(bins)
centre_bins= (bins[1:]+bins[:-1])/2
```

```
data['Centre']=centre_bins
mean = np.sum(data['Centre']*data['Frequences'])
var = np.sum((data['Centre']-mean)**2 * data['Frequences'])

print(data)
print('Moyenne=',mean)
print('Variance=',var)
```

```
         class  Effectifs  Frequences  Frequences cumulées  Centre
0    ]50,57.5]         17    0.191011             0.191011   53.75
1  ]57.5,60.5]         29    0.325843             0.516854   59.00
2  ]60.5,63.5]         30    0.337079             0.853933   62.00
3    ]63.5,70]         13    0.146067             1.000000   66.75
Moyenne= 60.140449438202246
Variance= 15.771004292387325
```

Question 2

Le calcul des quantiles dans le cas des classes se fait a l'aide d'une formule d'intérpolation qui se base sur les fréquences cumulées des différentes classes. Si α designe l'ordre du quantile (par exemple $\alpha = 0.5$ pour la mediane, $\alpha = 0.1$ pour D_1...) alors on utilise la methode suivante **Methode d'interpolation**

- *Etape 1:* Identifier la classe a laquelle appartient le quantile. On observe la table et on selectionne la classe à partir de laquelle la fréquence cumulée est supérieure ou égale α. En d'autres terme on trouve un indice j pour lequel
$$F_{j-1} < \alpha, \quad F_j \geq \alpha$$
où F_j et la $j$ème fréquence cumulée Par exemple pour $\alpha = 0.5$ on va selectionner la $j = 2$ème classe.
- *Etape 2:* Calcul du quantile
$$Q_\alpha = y_{j-1} + \frac{\alpha - F_{j-1}}{F_j - F_{j-1}}(y_j - y_{j-1})$$
où . y_{j-1} est le bord gauche de la classe selectionnée et y_j est le bord droit. . F_{j-1} est la fréquence cumulée de la classe $j-1$ et F_j est la fréquence cumulée de la classe j.

Dans notre exemple $\alpha = 0.5$ on a que
$$m = Q_{0.5} = y_1 + \frac{0.5 - F_1}{F_2 - F_1}(y_2 - y_1) = 57.5 + \frac{0.5 - 0.191011}{0.516854 - 0.191011}(60.5 - 57.5) = 60.34$$
Pour le reste les formules sont similaires
$$D_1 = Q_{0.1} = y_0 + \frac{0.1 - F_0}{F_1 - F_0}(y_1 - y_0) = 50.0 + \frac{0.10 - 0}{0.191011 - 0}(57.5 - 50) = 53.92$$
$$Q_1 = Q_{0.25} = y_1 + \frac{0.25 - F_2}{F_2 - F_1}(y_2 - y_1) = 58.04$$
$$Q_3 = Q_{0.75} = y_2 + \frac{0.75 - F_2}{F_3 - F_2}(y_3 - y_2) = 62.575$$
$$D_9 = Q_{0.90} = y_2 + \frac{0.9 - F_2}{F_3 - F_2}(y_3 - y_2) = 65.55$$

```
#Mediane, Quartiles
def quantile_interpolation(bins,alpha,cumul_freq):
    j= (cumul_freq>=alpha).idxmax() if (cumul_freq>=alpha).any() else None
    if j!=0:
        return bins[j] + (alpha-cumul_freq[j-1])/(cumul_freq[j]-cumul_freq[j-1])*(bins[j+1]-bins[j])
    else:
        return bins[j] + (alpha)/(cumul_freq[j])*(bins[j+1]-bins[j])


bins=bins
cumul_freq=data['Frequences cumulées']
alpha_tab=[0.1,0.25,0.5,0.75,0.9]
for alpha in alpha_tab:
    temp = quantile_interpolation(bins,alpha,cumul_freq)
    print('Q_({})={}'.format(alpha,temp))
```

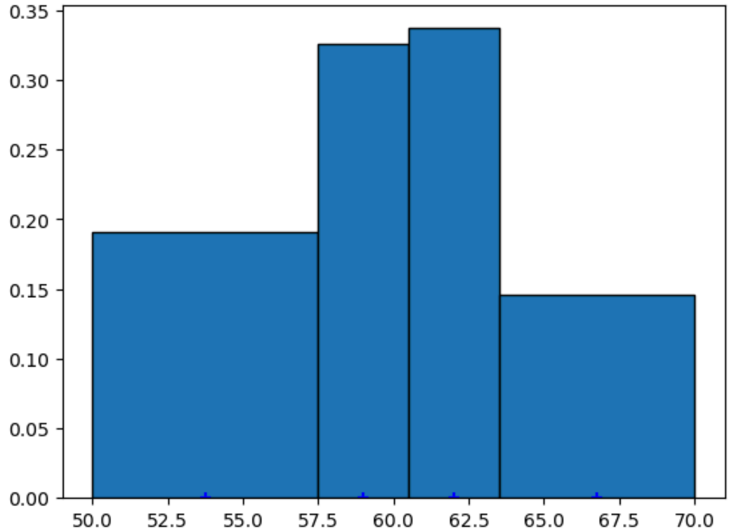
```
Q_(0.1)=53.9264705882353
Q_(0.25)=58.043103448275865
Q_(0.5)=60.3448275862069
Q_(0.75)=62.575
Q_(0.9)=65.55
```

Question 5

On affiche ici l'histogramme des fréquences :

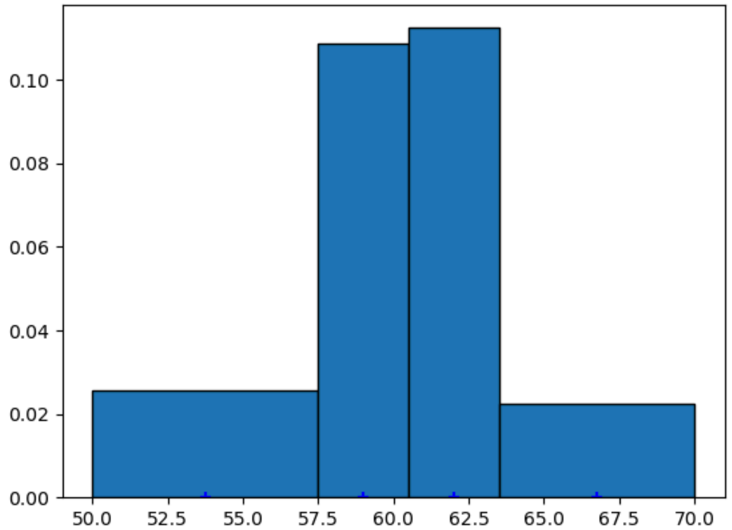
- En abscisse : les classes
- En ordonnée : la fréquence (pas la fréquence cumulée) de ces classes

```
#Histogram des fréquences
plt.figure()
bins_widths = np.diff(bins)
plt.bar(bins[:-1],data['Frequences'],width = bins_widths, edgecolor = 'black',align='edge')
plt.scatter(data['Centre'],np.zeros(4),marker='+',color='b',alpha=1)
plt.show()
```



On calcule maitenant les densité de fréquences $df_i = \frac{f_i}{a_i}$ où $a_i = y_i - y_{i-1}$ est l'amplitude des classes.

```
#densités de frequences
amplitudes= np.diff(bins)
freq_density= data['Frequences']/amplitudes

data['densité fréquence']= freq_density

plt.figure()
bins_widths = np.diff(bins)
plt.bar(bins[:-1],data['densité fréquence'],width = bins_widths, edgecolor = 'black',align='edge')
plt.scatter(data['Centre'],np.zeros(4),marker='+',color='b',alpha=1)
plt.show()
```



Double-cliquez (ou appuyez sur Entrée) pour modifier

```
import numpy as np
```

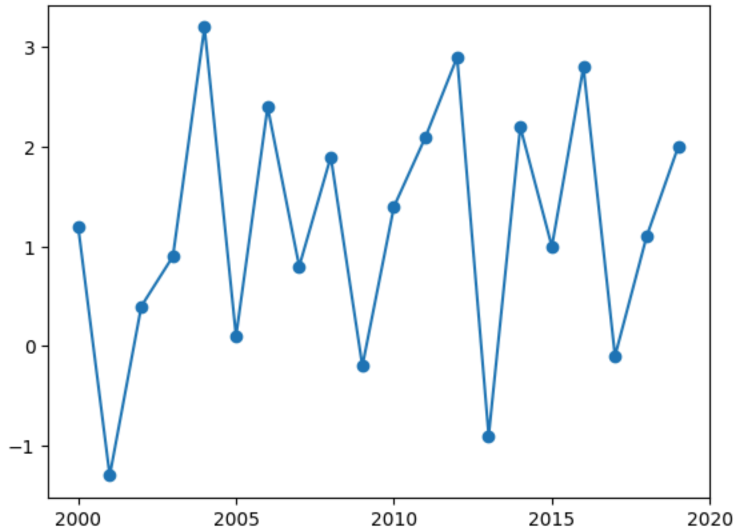
Exercice 1.3

```
import numpy as np
import pandas as pd

# Data
array=[1.2,-1.3,0.4,0.9,3.2,0.1,2.4,0.8,1.9,-0.2,1.4,2.1,2.9,-0.9,2.2,1.0,2.8,-0.1,1.1,2.0]

# Define bins
bins = np.array([-2, -1, 0, 1, 2, 3, 4])

# Create IntervalIndex with right-closed bins
interval_index = pd.IntervalIndex.from_breaks(bins, closed='right')

# Manually count occurrences in each bin
counts = [((array > left) & (array <= right)).sum() for left, right in zip(bins[:-1], bins[1:])]

# Create DataFrame
data_temperature = pd.DataFrame({'Effectif': counts}, index=interval_index)
data_temperature['Frequence'] = data_temperature['Effectif'] / np.sum(data_temperature['Effectif'])
data_temperature['Frequences cumulées'] = np.cumsum(data_temperature['Frequence'])

# Display result
print(data_temperature)
```

```
          Effectif  Frequence  Frequences cumulées
(-2, -1]         1       0.05                 0.05
(-1, 0]          3       0.15                 0.20
(0, 1]           5       0.25                 0.45
(1, 2]           5       0.25                 0.70
(2, 3]           5       0.25                 0.95
(3, 4]           1       0.05                 1.00
```

```
#plot array
plt.figure()
plt.plot(np.arange(2000,2020,1),array,marker='o')
plt.xticks([2000,2005,2010,2015,2020])
```

```
([<matplotlib.axis.XTick at 0x79d2d32c4950>,
  <matplotlib.axis.XTick at 0x79d2d67de710>,
  <matplotlib.axis.XTick at 0x79d2d2815550>,
  <matplotlib.axis.XTick at 0x79d2d3382150>,
  <matplotlib.axis.XTick at 0x79d2d300a310>],
 [Text(2000, 0, '2000'),
  Text(2005, 0, '2005'),
  Text(2010, 0, '2010'),
  Text(2015, 0, '2015'),
  Text(2020, 0, '2020')])
```

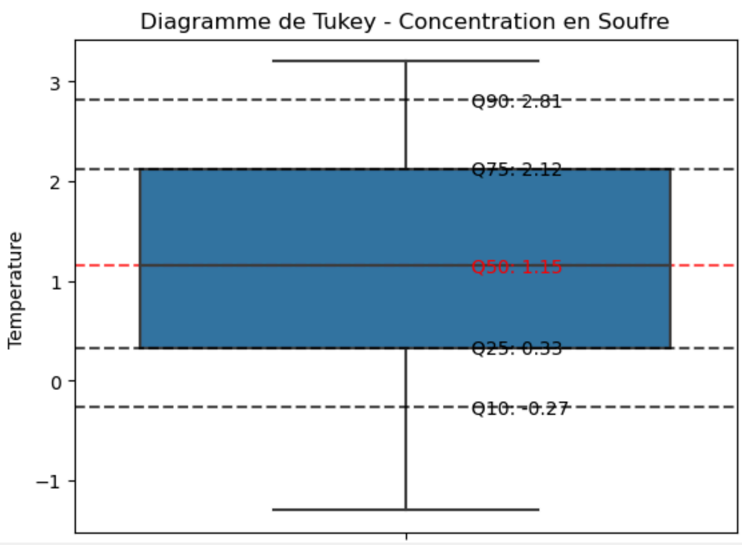



```
def quantile_interpolation(bins,alpha,cumul_freq):
    j= (cumul_freq>=alpha).idxmax() if (cumul_freq>=alpha).any() else None
    if j!=0:
        return bins[j] + (alpha-cumul_freq[j-1])/(cumul_freq[j]-cumul_freq[j-1])*(bins[j+1]-bins[j])
    else:
        return bins[j] + (alpha)/(cumul_freq[j])*(bins[j+1]-bins[j])

bins=bins

cumul_freq=pd.Series(data_temperature['Frequences cumulées']).reset_index(drop=True)
bins=bins
alpha_tab=[0.10,0.25,0.5,0.75,0.9]
```

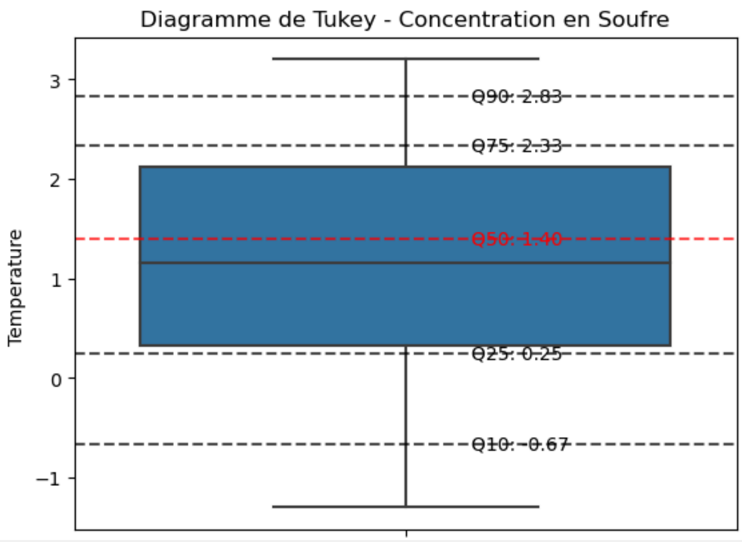
```
quantiles=[]
for alpha in alpha_tab:
    temp = quantile_interpolation(bins,alpha,cumul_freq)
    print('Q_{}={}'.format(alpha,temp))
    quantiles.append(temp)
```

```
Q_0.1=-0.6666666666666667
Q_0.25=0.19999999999999996
Q_0.5=1.2
Q_0.75=2.2
Q_0.9=2.8000000000000003
```

```
# Afficher le diagram de Tuckey avec les quantiles
plt.figure('Tuckey Diagram')
sns.boxplot(y=array)
plt.title("Diagramme de Tukey - Concentration en Soufre")
plt.ylabel("Temperature ")


for i in range(5):
    q=alpha_tab[i]
    value=np.quantile(array,q)
    plt.axhline(y=value, color="red" if q == 0.50 else "black", linestyle="--", alpha=0.7)
    plt.text(0.1, value, f"Q{int(q*100)}: {value:.2f}", va="center", ha="left", fontsize=10, color="red" if q == 0.50 else "black")
```




```
# Afficher le diagram de Tuckey avec les quantiles
plt.figure('Tuckey Diagram')
sns.boxplot(y=array)
plt.title("Diagramme de Tukey - Concentration en Soufre")
plt.ylabel("Temperature ")


for i in range(5):
    q=alpha_tab[i]
    value=quantiles[i]
    plt.axhline(y=value, color="red" if q == 0.50 else "black", linestyle="--", alpha=0.7)
    plt.text(0.1, value, f"Q{int(q*100)}: {value:.2f}", va="center", ha="left", fontsize=10, color="red" if q == 0.50 else "black")
```




```
#Moyenne classe, ecartype et coefficient de variation


def moyenne_classe(bins,freq):
    c = (bins[1:] + bins[:-1])/2
    return np.sum(c*freq)

def variance_classe(bins,freq):
    c = (bins[1:] + bins[:-1])/2
    m = moyenne_classe(bins,freq)
    return np.sum((c-m)**2*freq)

bins=np.array(bins)
freq=data_temperature['Frequence']

m_c = moyenne_classe(bins,freq)
v_c= variance_classe(bins,freq)
e_t= np.sqrt(variance_classe(bins,freq))
coeff_var= e_t/m_c
print('Moyenne_c=',m_c)
print('Variance_c=',v_c)
print('Ecart-type', e_t)

print('Coefficient de variation= ', coeff_var )
```

```
Moyenne_c= 1.15
Variance_c= 1.6275
Ecart-type 1.2757350822173072
Coefficient de variation=  1.1093348541020065
```

```
#Diagrame de fréquences
amplitudes= np.diff(bins)
print(data_temperature)
freq_density= data_temperature['Frequence']/amplitudes

data_temperature['densité fréquence']= freq_density


plt.figure()
bins_widths = np.diff(bins)
plt.bar(bins[:-1],data_temperature['densité fréquence'],width = bins_widths, edgecolor = 'black',align='edge')
plt.show()
```

	Effectif	Frequence	Frequences cumulées
(-2, -1]	1	0.05	0.05
(-1, 0]	3	0.15	0.20
(0, 1]	5	0.25	0.45
(1, 2]	5	0.25	0.70
(2, 3]	5	0.25	0.95
(3, 4]	1	0.05	1.00