TP1 SERIES TEMPS

Réaliser par MEKA MOISE CHRISTIAN JUNIOR 21T2561

Il s'agit de l'implémentation du TP1 sur le dataset des passagers d'une agence de voyage

```python
In [1]: import pandas as pd
        import numpy as np
        import matplotlib.pyplot as plt
```

```python
In [2]: FILEPATH="AirPassengers.csv"
        extension=FILEPATH.split(".")[-1]
        SEP=","
        if extension=="csv":
            df=pd.read_csv(FILEPATH, sep=SEP)
        else:
            df=pd.read_excel(FILEPATH, index_col=0)
```

```python
In [3]: extension
```

```
Out[3]: 'csv'
```

```python
In [4]: df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 144 entries, 0 to 143
Data columns (total 2 columns):
 #   Column       Non-Null Count  Dtype
---  ------       --------------  -----
 0   Month        144 non-null    object
 1   #Passengers  144 non-null    int64
dtypes: int64(1), object(1)
memory usage: 2.4+ KB
```

```python
In [5]: df.head()
```

Out[5]:

	Month	#Passengers
0	1949-01	112
1	1949-02	118
2	1949-03	132
3	1949-04	129
4	1949-05	121

```python
In [6]: metrique="#Passengers"
        period="Month"
```

```python
In [7]: df[metrique].describe()
```

```
Out[7]:  count    144.000000
         mean     280.298611
         std      119.966317
         min      104.000000
         25%      180.000000
         50%      265.500000
         75%      360.500000
         max      622.000000
         Name: #Passengers, dtype: float64
```

```
In [8]:  df=df.dropna()
```

Question 1

```
In [9]:  #moyenne
         np.mean(df[metrique])
```

```
Out[9]:  280.2986111111111
```

```
In [10]:  #varaince
          np.std(df[metrique])**2
```
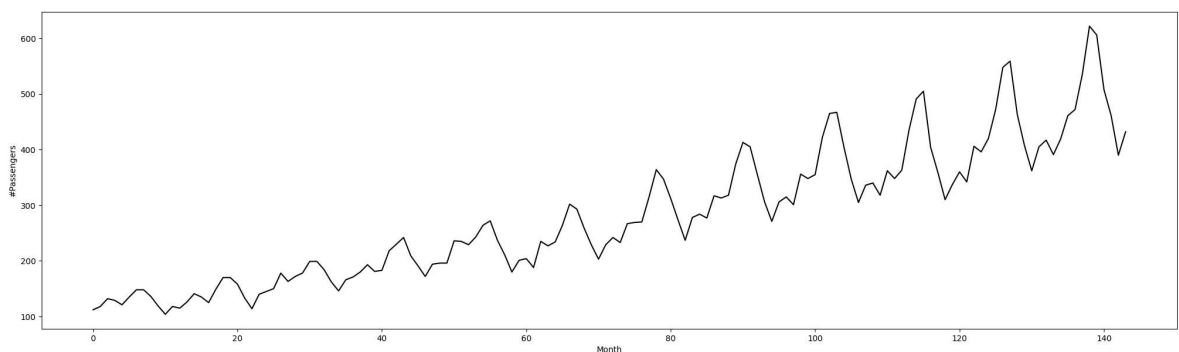
```
Out[10]:  14291.973331404317
```

```
In [11]:  #ecart-type
          np.std(df[metrique])
```

```
Out[11]:  119.54904153277147
```

Question 2

```
In [12]:  plt.figure(figsize=[25,7])
          plt.plot(df[metrique], color='black',)
          plt.xlabel(period)
          plt.ylabel(metrique)
          plt.show()
```
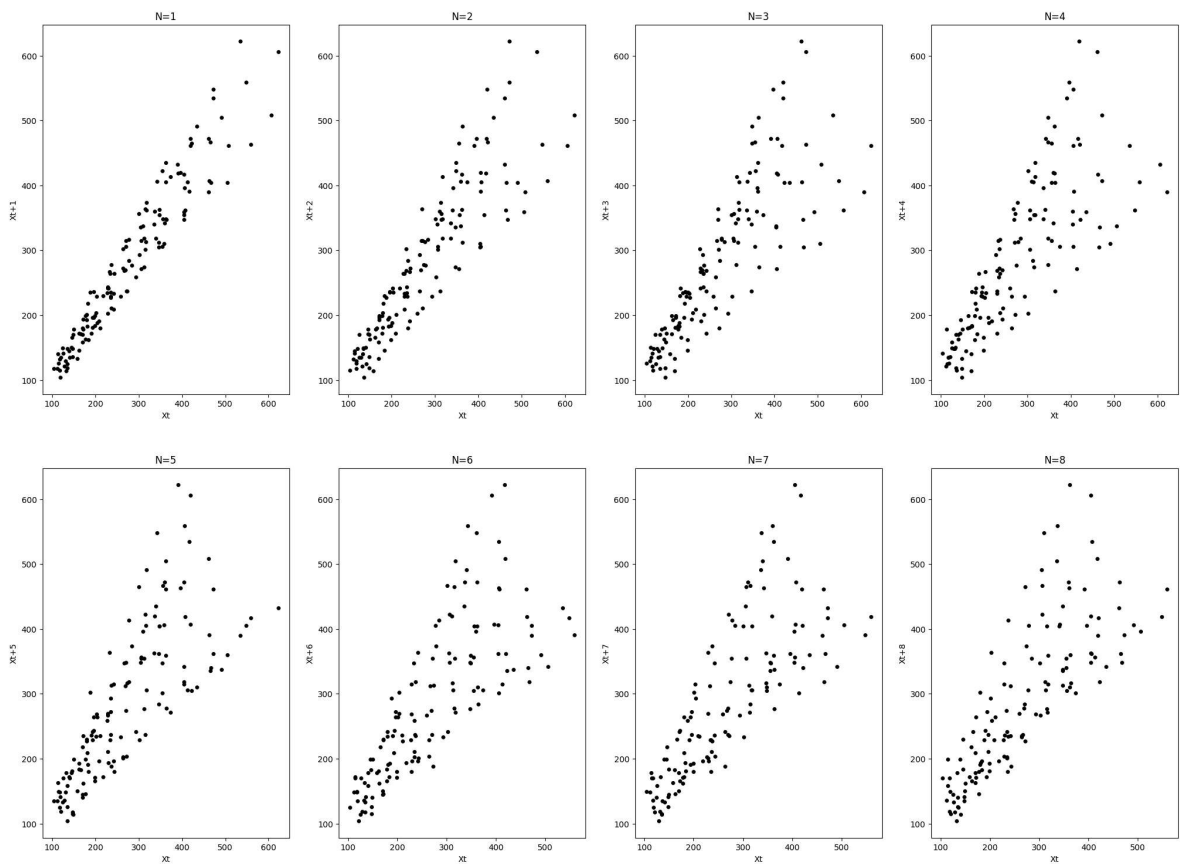

Question 3

```
In [13]:  ###NX
          plt.figure(figsize=[25,18])
          for N in range(1,9):
              plt.subplot(2,4,N)
```

```
    debut=0
    fin=df.shape[0]
    plt.scatter(df[metrique][debut:fin-N], df[metrique][debut+N:fin], color='bla
    plt.xlabel("Xt")
    plt.ylabel(f"Xt+{N}")
    plt.title(f"N={N}")
plt.show()
```


Question 4

```
In [14]: #Calcul l'auto-Covariance empirique
         def auto_cov(data, K, moy):
             debut=0
             fin=len(data)
             Xt=data[debut:fin-K]
             Xt_k=data[debut+K:fin]

             cov=0
             for i in range(fin-K):
                 cov+=(Xt[i]-moy)*(Xt_k[i]-moy)
             return cov/(fin-K)
```

```
In [15]: def auto_cor(data, K):
             moy=np.mean(data)
             cov_0=auto_cov(data, 0, moy)
             cov_K=auto_cov(data, K, moy)
             return cov_K/cov_0
```

```
In [16]: from tqdm import tqdm
         auto_cor_all=list()
         data=list(df[metrique])
```

```
for i in tqdm(range(1, 51)):
    auto_cor_all.append(auto_cor(data, i))
```

100%|██|
████| 50/50 [00:00<00:00, 3342.50it/s]

In [17]:
```
indexes=[i for i in range(1,51)]
plt.bar(indexes,auto_cor_all, align='edge', width= 0.25, color='black')
plt.xlabel("Ordre")
plt.ylabel("Auto-Correlation")
```

Out[17]: Text(0, 0.5, 'Auto-Correlation')