TP DE SERIE TEMPORELLE

REALISE PAR MEKA MOISE CHRISTIAN JUNIOR 21T2561

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

In [2]:
```
FILEPATH="Datasets/SARS-Cov2-PCRdata.csv"
HORIZON=np.random.randint(1,100)
FEATURE="Tx"
SEP=";"
YLABEL="Taux de positivité"
XLABEL="Semaine Epidémiologique"
TITLE="Evolution par semane epidémiologique du taux de positivité \nau diagnosti
```

In [3]:
```
df=pd.read_csv(FILEPATH,sep=SEP)
```

In [4]:
```
df.head()
```

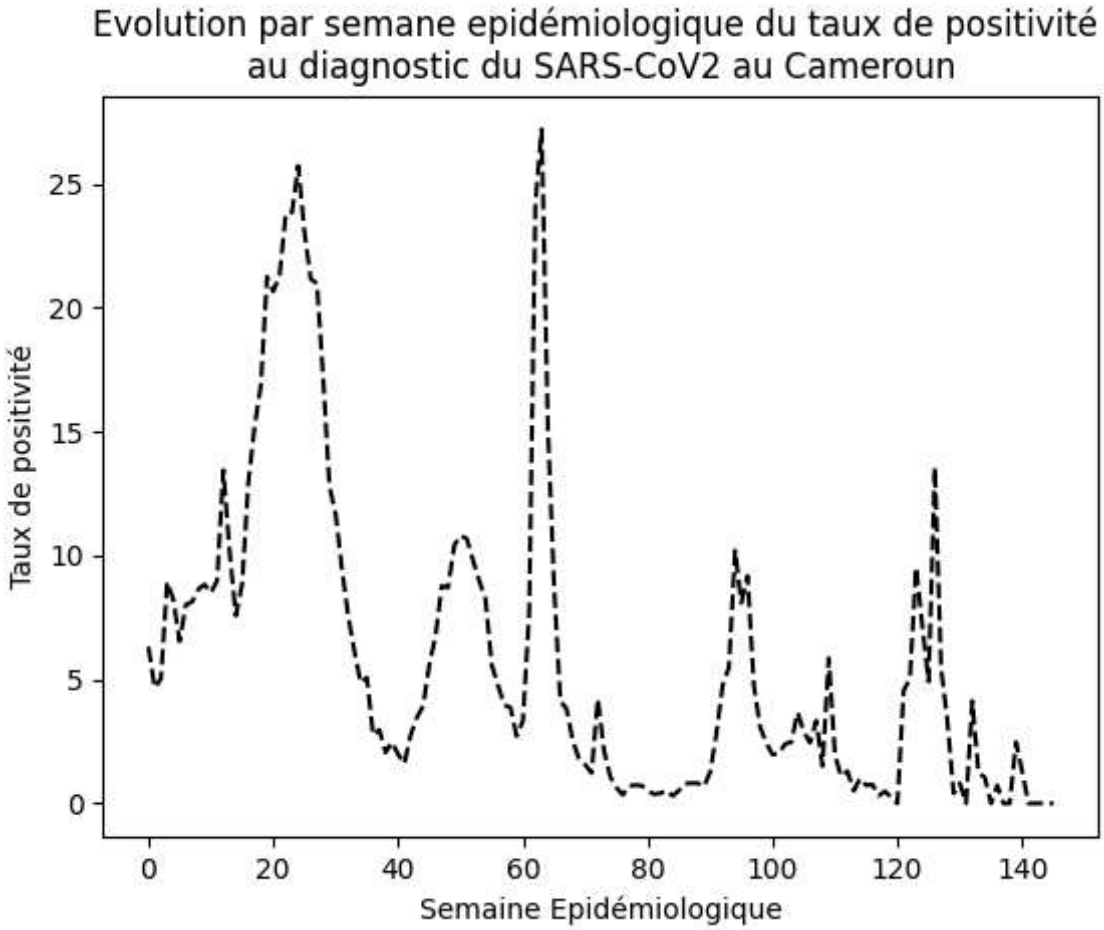
Out[4]:

	Year	Period	Full Period	Ech Analyse	Ech Positif	Tx
0	2020	S42	20-S42	380	24	6.315789
1	2020	S43	20-S43	4050	189	4.666667
2	2020	S44	20-S44	4682	235	5.019223
3	2020	S45	20-S45	3715	332	8.936743
4	2020	S46	20-S46	4167	342	8.207343

In [5]:
```
all_alpha=[0.001,0.005, 0.01, 0.05, 0.1, 0.3, 0.5, 0.9]
#all_alpha=np.linspace(0.001,0.9,10)
```

In [6]:
```
plt.plot(df[FEATURE], c="black", ls='--')
plt.xlabel(XLABEL)
plt.ylabel(YLABEL)
plt.title(TITLE)
```

Out[6]:
```
Text(0.5, 1.0, 'Evolution par semane epidémiologique du taux de positivité \nau
diagnostic du SARS-CoV2 au Cameroun')
```



In [7]:
```
def predict_simple_expo_lissage(data, alpha, taille, horizon=1):
    results=[]
    for i in range(taille-horizon):
        if i==0:
            results.append((1-alpha)*data[i])
        else:
            tmp=(1-alpha)*data[i]+alpha*results[i-1]
            results.append(tmp)
    return results
```

In [8]:
```
alpha=all_alpha[-1]
res=predict_simple_expo_lissage(df[FEATURE], alpha, len(df[FEATURE]))
```

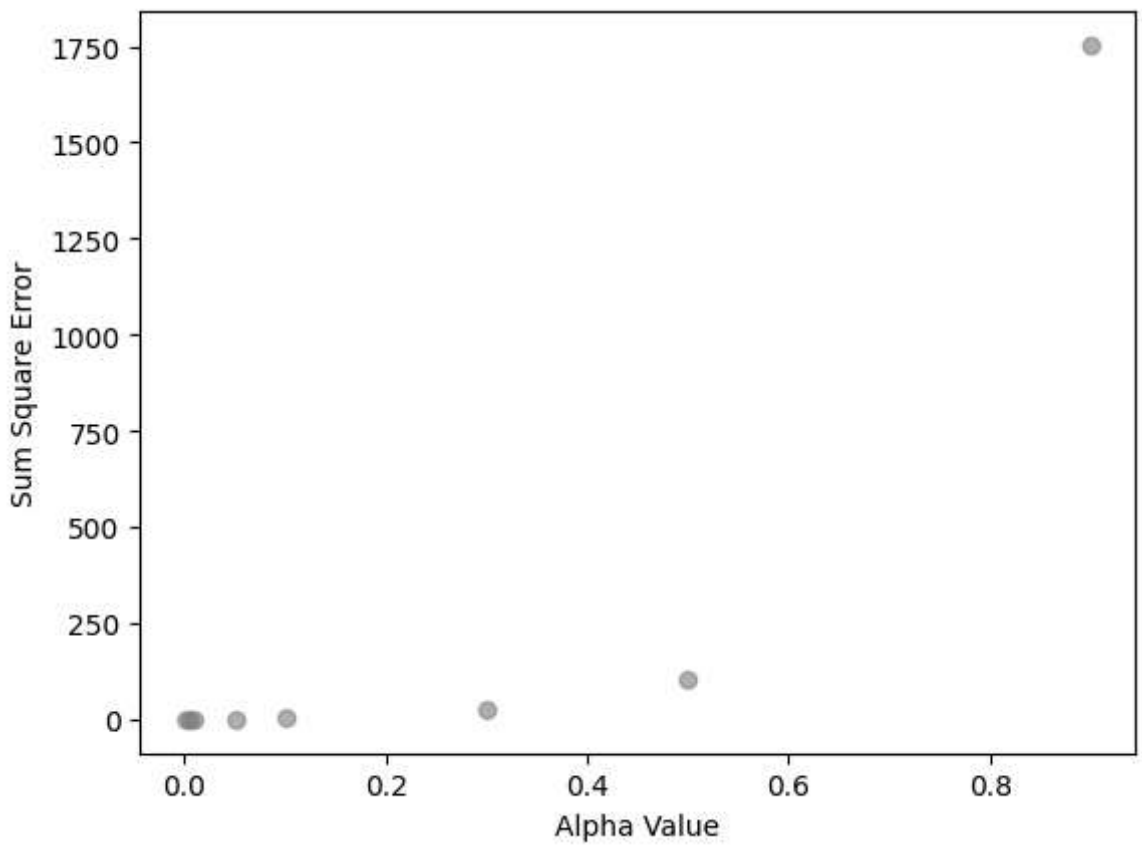
In [9]:
```
all_results={}
for alpha in all_alpha:
    all_results[f"alpha_{alpha}"]=predict_simple_expo_lissage(df[FEATURE], alpha
```

In [10]:
```
def sum_square_error(real, predic):
    result=real-predic
    result=result**2
    return np.sum(result)
```

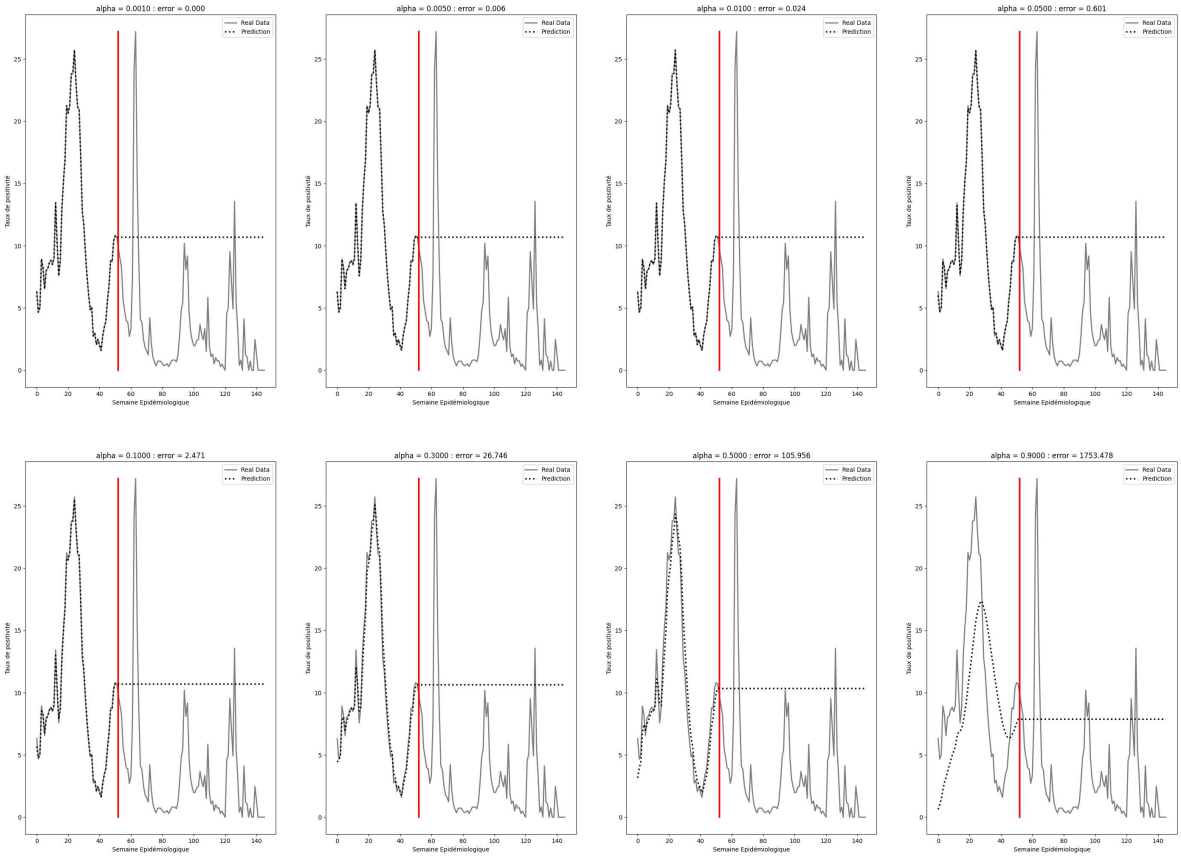
In [11]:
```
all_error={}
for key in all_results.keys():
    real=np.array(df[FEATURE])
    predic=np.array(all_results[key])
    all_error[key]=sum_square_error(real[:real.shape[0]-HORIZON],predic)
```

In [12]:
```
list_=[]
for key in all_error.keys():
    list_.append(all_error[key])
plt.scatter(all_alpha, list_, alpha=0.6, c="gray")
plt.xlabel("Alpha Value")
plt.ylabel("Sum Square Error")
```

Out[12]: Text(0, 0.5, 'Sum Square Error')



In [13]:
```
plt.figure(figsize=[35, 25])
for i in range(len(all_alpha)):
    plt.subplot(2, int(len(all_alpha)/2), i+1)
    plt.plot(df[FEATURE] , label='Real Data', c='gray', lw=2)
    m1=np.max(df[FEATURE])
    m2=np.min(df[FEATURE])
    x=len(df)-HORIZON
    y=np.linspace(m2,m1,100)
    x=np.ones(y.shape)*x
    plt.plot(x,y, c="red", lw=2.7, ls="-")
    last=all_results[f"alpha_{all_alpha[i]}"][-1]
    tmp=all_results[f"alpha_{all_alpha[i]}"]
    for j in range(HORIZON):
        tmp.append(last)
    plt.plot(tmp, label='Prediction', lw=2.5 , ls=':', c='black')
    alpha=f"alpha_{all_alpha[i]}"
    plt.title(f"alpha = {all_alpha[i]:.4f} : error = {all_error[alpha]:.3f}")
    plt.xlabel(XLABEL)
    plt.ylabel(YLABEL)
    plt.legend()
```


alpha = 0.0010 : error = 0.000
alpha = 0.0050 : error = 0.006
alpha = 0.0100 : error = 0.024
alpha = 0.0500 : error = 0.601
alpha = 0.1000 : error = 2.471
alpha = 0.3000 : error = 26.746
alpha = 0.5000 : error = 105.956
alpha = 0.9000 : error = 1753.478
Real Data
Prediction
Taux de positivité
Semaine Epidémiologique

In [ ]: