

In [150...

```
import pandas as pd
import numpy as np
from sklearn.metrics import mean_squared_error

alg=pd.read_csv('AlgeriaExport.txt', header=None)

alg.head()
```

Out[150...

**0**

```
0 39.043173
1 46.244557
2 19.793873
3 24.684682
4 25.084059
```

In [152...

```
from statsmodels.tsa.api import SimpleExpSmoothing

fit=SimpleExpSmoothing(alg[0], initialization_method="estimated").fit()
optimal_alpha=fit.model.params["smoothing_level"]

def SimpleAlphaSmoothing(alpha, t_1, t):
    return alpha*t_1+(1-alpha)*t

def countMSE(alpha):
    reals=alg[0]
    comp=[alg[0][0]]
    ests=[alg[0][0]]
    for i in range(1, reals.size):
        est=SimpleAlphaSmoothing(alpha,reals[i-1],ests[i-1])
        ests.append(est)
        comp.append(reals[i])
    MSE=mean_squared_error(comp,ests)
    return MSE

lowestMSE=100*alg.size #something big enough to be overwritten in first iteration
for alpha in np.arange(0.01,1.0,0.01):
    if countMSE(alpha)<lowestMSE:
        lowestMSE=countMSE(alpha)
        best_alpha=alpha

print("Minimal MSE error for alpha: ",best_alpha," , optimal alpha value from statsmo
```

```
Minimal MSE error for alpha: 0.8400000000000001 , optimal alpha value from statsmod
el: 0.8397833478991826
```

In [ ]: