In [1]:

```python
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import statsmodels.formula.api as snf

import warnings
warnings.filterwarnings('ignore')
```

In [2]:

```python
airline_data = pd.read_csv('Airlines+Data.csv')
airline_data
```

Out[2]:

	Month	Passengers
0	Jan-95	112
1	Feb-95	118
2	Mar-95	132
3	Apr-95	129
4	May-95	121
...	...	...
91	Aug-02	405
92	Sep-02	355
93	Oct-02	306
94	Nov-02	271
95	Dec-02	306

96 rows × 2 columns

In [3]:

```python
airline_data.isna().sum()
```

Out[3]:

```
Month         0
Passengers    0
dtype: int64
```

In [4]:

```python
months=["Jan","Feb","Mar","Apr","May","Jun","Jul","Aug","Sep","Oct","Nov","Dec"]
```

In [5]:

```python
p=airline_data["Month"][0]
p[0:3]
```

Out[5]:

```
'Jan'
```

In [6]:

```python
airline_data["months"]=0
airline_data
```

Out[6]:

	Month	Passengers	months
0	Jan-95	112	0
1	Feb-95	118	0
2	Mar-95	132	0
3	Apr-95	129	0
4	May-95	121	0
...	...	...	...
91	Aug-02	405	0
92	Sep-02	355	0
93	Oct-02	306	0
94	Nov-02	271	0
95	Dec-02	306	0

96 rows × 3 columns

In [7]:

```python
for i in range(96):
    p=airline_data["Month"][i]
    airline_data["months"][i]=p[0:3]
```

In [8]:

```
month_dummies=pd.DataFrame(pd.get_dummies(airline_data["months"]))
month_dummies=month_dummies.iloc[:,[4,3,7,0,8,6,5,1,11,10,9,2]]
```

In [9]:

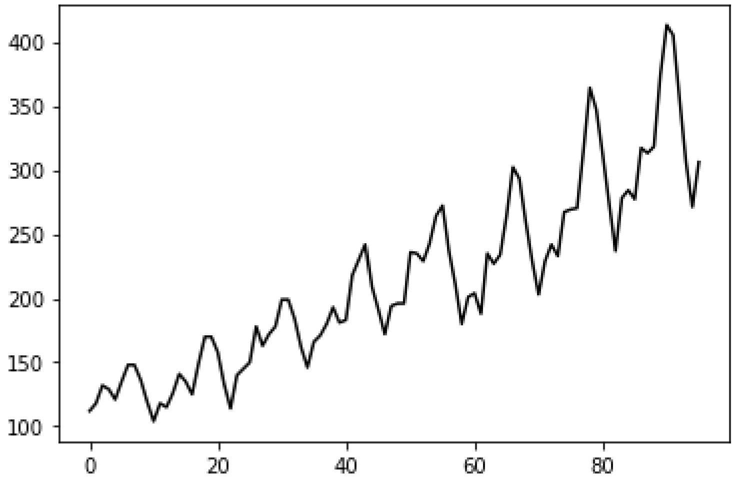
```
airline_data_1=pd.concat([airline_data,month_dummies],axis=1)
airline_data_1["t"]=np.arange(1,97)
airline_data_1["t_Squared"]=airline_data_1["t"]*airline_data_1["t"]
airline_data_1["log_passengers"]=np.log(airline_data["Passengers"])
```

In [10]:

```
######Time Plot############
airline_data_1.Passengers.plot(style="k")
```

Out[10]:

```
<AxesSubplot:>
```



In [11]:

```
#########Dividing the dataset into train & test############
Train=airline_data_1.head(80)
Test=airline_data_1.tail(16)
Test=Test.set_index(np.arange(1,17))
```

In [12]:

```
####Building the Linear model###########
Lin_model=snf.ols("Passengers~t",data=Train).fit()
Lin_pred=pd.Series(Lin_model.predict(pd.DataFrame(Test["t"])))
Lin_rmse=np.sqrt(np.mean((np.array(Test["Passengers"])-np.array(Lin_pred))**2))
```

In [13]:

```
Lin_rmse
```

Out[13]:

```
47.54262406772677
```

In [14]:

```
####Building the Exponential Model############
Exp_model=snf.ols("log_passengers~t",data=Train).fit()
Exp_pred=pd.Series(Exp_model.predict(pd.DataFrame(Test["t"])))
Exp_rmse=np.sqrt(np.mean((np.array(Test["Passengers"])-np.array(np.exp(Exp_pred)))**2))
```

In [15]:

```
Exp_rmse
```

Out[15]:

```
43.79373939334308
```

In [16]:

```
####Building the Quadratic Model##########
Quad_model=snf.ols("Passengers~t+t_Squared",data=Train).fit()
Quad_pred=pd.Series(Quad_model.predict(pd.DataFrame(Test[["t","t_Squared"]])))
Quad_rmse=np.sqrt(np.mean((np.array(Test["Passengers"])-np.array(Quad_pred))**2))
```

In [17]:

```
Quad_rmse
```

Out[17]:

```
43.65440369584248
```

In [19]:

```
#####Building the Additive  Seasonality#########
Add_sea=snf.ols("Passengers~Jan+Feb+Mar+Apr+May+Jun+Jul+Aug+Sep+Oct+Nov+Dec",data=Train).fit()
Add_sea_pred=pd.Series(Add_sea.predict(Test[["Jan","Feb","Mar","Apr","May","Jun","Jul","Aug","Sep","Oct","N
Add_sea_rmse=np.sqrt(np.mean((np.array(Test["Passengers"])-np.array(Add_sea_pred))**2))
```

In [20]:

```
Add_sea_rmse
```

Out[20]:

```
129.26647641443301
```

In [21]:
```
#####Building the Additive Quadratic Seasonality#########
Add_sea_Quad=snf.ols("Passengers~Jan+Feb+Mar+Apr+May+Jun+Jul+Aug+Sep+Oct+Nov+Dec+t+t_Squared",data=Train).f
Add_sea_Quad_pred=pd.Series(Add_sea_Quad.predict(Test[["Jan","Feb","Mar","Apr","May","Jun","Jul","Aug","Sep
Add_sea_Quad_rmse=np.sqrt(np.mean((np.array(Test["Passengers"])-np.array(Add_sea_Quad_pred))**2))
```

In [22]:
```
Add_sea_Quad_rmse
```

Out[22]:
```
23.910983570103003
```

In [23]:
```
#####Multiplicative Additive Seasonality#########
Mul_ad_sea=snf.ols("log_passengers~Jan+Feb+Mar+Apr+May+Jun+Jul+Aug+Sep+Oct+Nov+Dec",data=Train).fit()
Mul_ad_sea_pred=pd.Series(Mul_ad_sea.predict(Test[["Jan","Feb","Mar","Apr","May","Jun","Jul","Aug","Sep","O
Mul_ad_rmse=np.sqrt(np.mean((np.array(Test["Passengers"])-np.array(np.exp(Mul_ad_sea_pred)))**2))
```

In [24]:
```
Mul_ad_rmse
```

Out[24]:
```
135.32648414621056
```

In [25]:
```
#####Multiplicative Additive Quadratic Seasonality#########
Mul_ad_Quad_sea=snf.ols("log_passengers~Jan+Feb+Mar+Apr+May+Jun+Jul+Aug+Sep+Oct+Nov+Dec+t+t_Squared",data=T
Mul_ad_Quad_sea_pred=pd.Series(Mul_ad_Quad_sea.predict(Test[["Jan","Feb","Mar","Apr","May","Jun","Jul","Aug
Mul_ad_sea_rmse=np.sqrt(np.mean((np.array(Test["Passengers"])-np.array(np.exp(Mul_ad_Quad_sea_pred)))**2))
```

In [26]:
```
Mul_ad_sea_rmse
```

Out[26]:
```
23.08634854595413
```

In [27]:
```
#####Storing the values###########
data = {"MODEL":pd.Series(["Lin_rmse","Exp_rmse","Quad_rmse","Add_sea_rmse","Add_sea_Quad_rmse","Mul_ad_rms
table_rmse=pd.DataFrame(data)
```

In [28]:

```
table_rmse
```

Out[28]:

	MODEL	RMSE_Values
0	Lin_rmse	47.542624
1	Exp_rmse	43.793739
2	Quad_rmse	43.654404
3	Add_sea_rmse	129.266476
4	Add_sea_Quad_rmse	23.910984
5	Mul_ad_rmse	135.326484
6	Mul_ad_sea_rmse	23.086349

so Mul_ad_sea_rmse has the least value among the models prepared so far

In [ ]:

```

```