In [11]:

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

In [12]:

```
import numpy as np
import pandas as pd
```

In [13]:

```
df = pd.read_csv('/content/Airline_Passangers.csv')
```

In [17]:

```
df.isnull().sum()
```

Out[17]:

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

In [18]:

```
df.shape
```

Out[18]:

```
(144, 2)
```

Building MA(q) model:

- **order = (0,0,1)**

In [19]:

```
order = (0,0,1)
```

In [20]:

```
from statsmodels.tsa.arima_model import ARIMA
```

In [28]:

```
model = ARIMA(df.Passengers[:50], order = order)
```

In [31]:

```
MA_model = model.fit()
MA_forecast = np.round(MA_model.predict(50,60),0)
```

In [32]:

```
MA_forecast
```

Out[32]:

```
50    186.0
51    160.0
52    160.0
53    160.0
54    160.0
55    160.0
56    160.0
57    160.0
58    160.0
```

```
59    160.0
60    160.0
dtype: float64
```

In [33]:

```
MA_model.summary2()
```

Out[33]:

Model:	ARMA	**BIC:**	460.5762
Dependent Variable:	Passengers	**Log-Likelihood:**	-224.42
Date:	2022-08-29 18:27	**Scale:**	1.0000
No. Observations:	50	**Method:**	css-mle
Df Model:	2	**Sample:**	0
Df Residuals:	48		0
Converged:	1.0000	**S.D. of innovations:**	21.316
No. Iterations:	15.0000	**HQIC:**	457.024
AIC:	454.8401		

	Coef.	Std.Err.	t	P>|t|	[0.025	0.975]
const	159.9704	5.3651	29.8170	0.0000	149.4550	170.4857
ma.L1.Passengers	0.7952	0.0828	9.6060	0.0000	0.6330	0.9575

	Real	Imaginary	Modulus	Frequency
MA.1	-1.2575	0.0000	1.2575	0.5000

In [34]:

```
df_MA = pd.DataFrame({'actual': df.Passengers[50:60], 'pred':MA_forecast})
```

In [35]:

```
df_MA
```

Out[35]:

	actual	pred
50	236.0	186.0
51	235.0	160.0
52	229.0	160.0
53	243.0	160.0
54	264.0	160.0
55	272.0	160.0
56	237.0	160.0
57	211.0	160.0
58	180.0	160.0
59	201.0	160.0
60	NaN	160.0

In [36]:

```
def get_mape(actual, pred):
  mape = np.round(np.mean(np.abs(100*(actual-pred)/actual)),2)
```

```
    return mape
```

In [39]:

```
MA_mape = get_mape(df_MA.actual[:9],df_MA.pred[:9])
MA_mape
```

Out[39]:

```
29.53
```

In [40]:

```
from sklearn.metrics import mean_squared_error
MA_rmse = np.round(np.sqrt(mean_squared_error(df_MA.actual[:9],df_MA.pred[:9])),2)
MA_rmse
```

Out[40]:

```
76.06
```

In [ ]: