AREMA Model

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

In [2]:
```python
df_airline=pd.read_csv('airline_passengers.csv')
df_airline.head()
```

Out[2]:

	Month	Thousands of Passengers
0	1949-01	112.0
1	1949-02	118.0
2	1949-03	132.0
3	1949-04	129.0
4	1949-05	121.0

In [3]:
```python
df_airline.isnull().sum()
```

Out[3]:
```
Month                      0
Thousands of Passengers    1
dtype: int64
```

In [4]:
```python
df_airline.tail()
```

Out[4]:

	Month	Thousands of Passengers
140	1960-09	508.0
141	1960-10	461.0
142	1960-11	390.0
143	1960-12	432.0
144	International airline passengers: monthly tota...	NaN

In [5]:
```python
df_airline.dropna(axis=0,inplace=True)
```

In [6]:
```python
df_airline.isnull().sum()
```

Out[6]:
```
Month                      0
Thousands of Passengers    0
dtype: int64
```

In [7]:
```python
df_airline.info()
```
```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 144 entries, 0 to 143
Data columns (total 2 columns):
 #   Column                   Non-Null Count  Dtype
---  ------                   --------------  -----
 0   Month                    144 non-null    object
 1   Thousands of Passengers  144 non-null    float64
dtypes: float64(1), object(1)
memory usage: 3.4+ KB
```

In [8]:
```python
df_airline['Month']=pd.to_datetime(df_airline['Month'])
```

In [9]:
```python
df_airline.info()
```
```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 144 entries, 0 to 143
Data columns (total 2 columns):
 #   Column                   Non-Null Count  Dtype
---  ------                   --------------  -----
 0   Month                    144 non-null    datetime64[ns]
 1   Thousands of Passengers  144 non-null    float64
dtypes: datetime64[ns](1), float64(1)
memory usage: 3.4 KB
```

In [10]:
```python
df_airline.head()
```

Out[10]:

	Month	Thousands of Passengers
0	1949-01-01	112.0
1	1949-02-01	118.0
2	1949-03-01	132.0
3	1949-04-01	129.0
4	1949-05-01	121.0

In [11]:
```python
df_airline.set_index('Month',inplace=True)
```

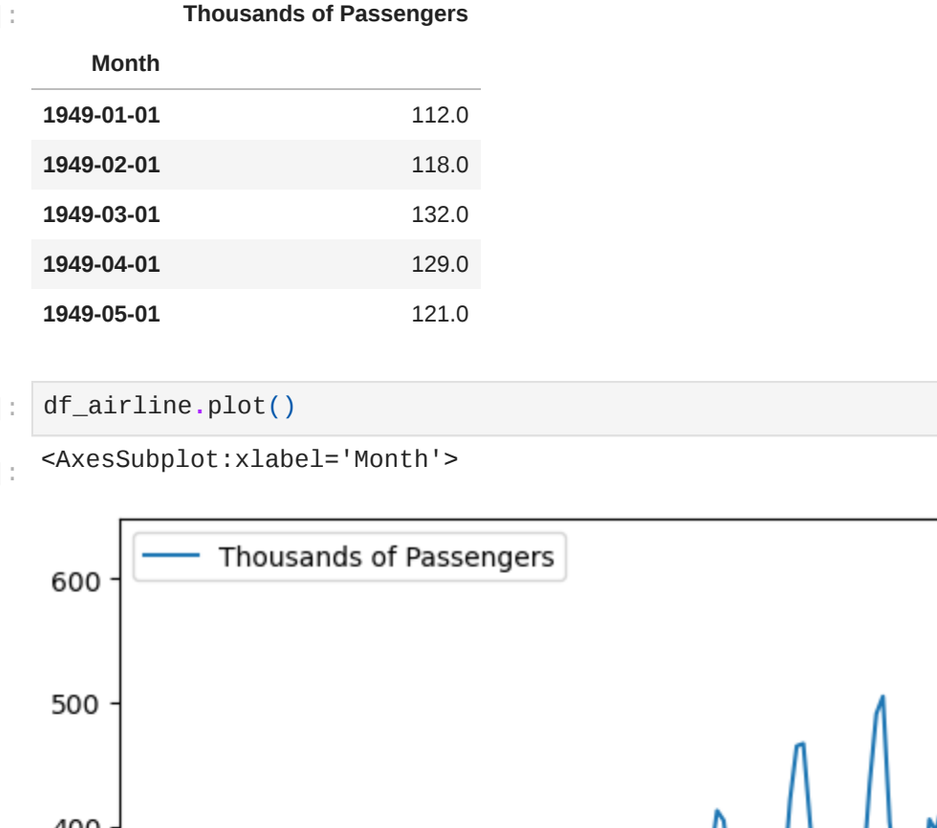
In [12]:
```python
df_airline.head()
```

Out[12]:

Month	Thousands of Passengers
1949-01-01	112.0
1949-02-01	118.0
1949-03-01	132.0
1949-04-01	129.0
1949-05-01	121.0

In [13]:
```python
df_airline.plot()
```

Out[13]: <AxesSubplot:xlabel='Month'>



In [14]:
```python
from statsmodels.tsa.stattools import adfuller
```

In [15]:
```python
#Ho: It is non stationary
#H1: It is stationary

def adf_test(series):
    result=adfuller(series)
    print('p- value: {}'.format(result[1]))
    if result[1] <= 0.05:
        print("strong evidence against the null hypothesis, reject the null hypothesis")
    else:
        print("weak evidence against null hypothesis, indicating it is non-stationary ")
```

In [16]:
```python
adf_test(df_airline['Thousands of Passengers'])
```
```
p- value: 0.9918802434376409
weak evidence against null hypothesis, indicating it is non-stationary
```

In [17]:
```python
## Use Techniques Differencing
df_airline['Passengers First Difference']=df_airline['Thousands of Passengers']-df_airline['Thousands of Passengers'].shift(1)
```

In [18]:
```python
df_airline.head()
```

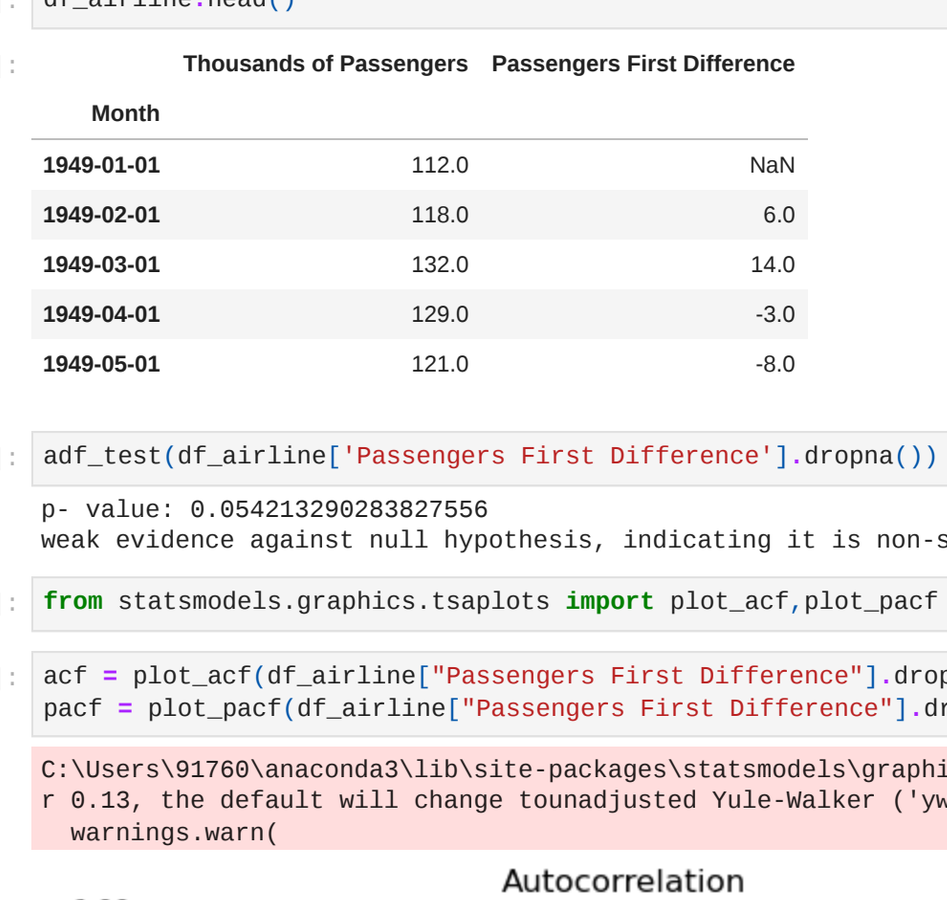
Out[18]:

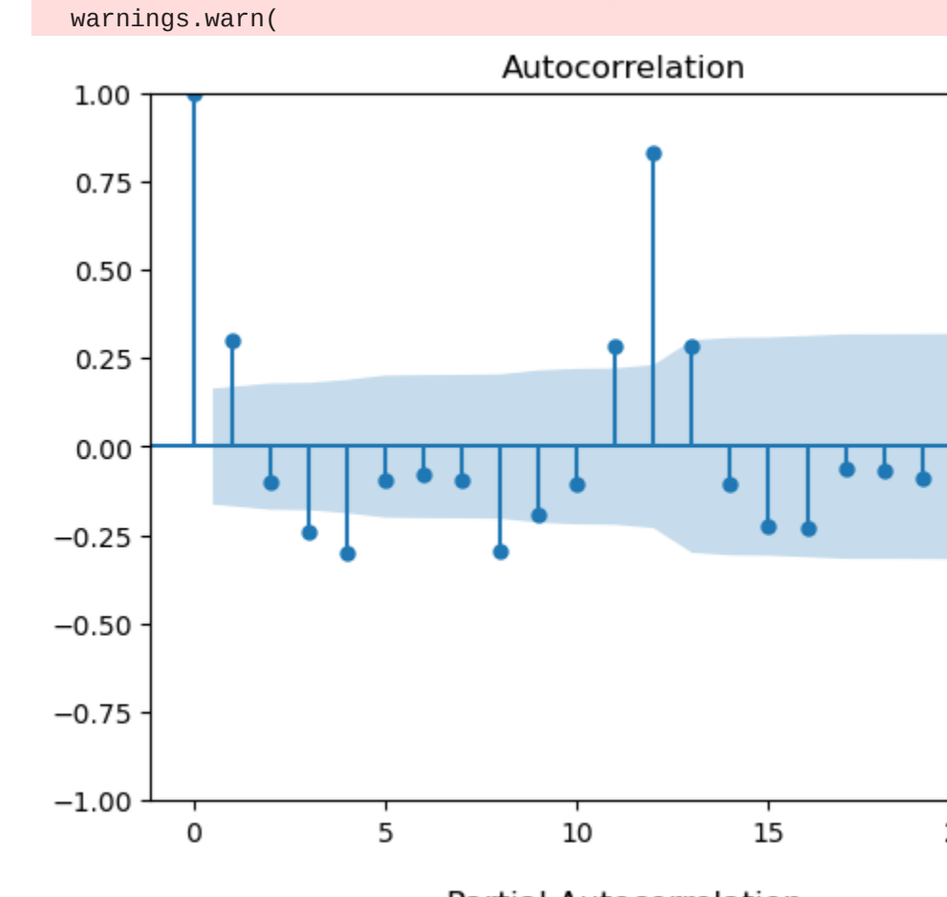
Month	Thousands of Passengers	Passengers First Difference
1949-01-01	112.0	NaN
1949-02-01	118.0	6.0
1949-03-01	132.0	14.0
1949-04-01	129.0	-3.0
1949-05-01	121.0	-8.0

In [19]:
```python
adf_test(df_airline['Passengers First Difference'].dropna())
```
```
p- value: 0.054213290283827556
weak evidence against null hypothesis, indicating it is non-stationary
```

In [20]:
```python
from statsmodels.graphics.tsaplots import plot_acf,plot_pacf
```

In [21]:
```python
acf = plot_acf(df_airline["Passengers First Difference"].dropna())
pacf = plot_pacf(df_airline["Passengers First Difference"].dropna())
```
```
C:\Users\91760\anaconda3\lib\site-packages\statsmodels\graphics\tsaplots.py:348: FutureWarning: The default method 'yw' can produce PACF values outside of the [-1,1] interval. After 0.13, the default will change tounadjusted Yule-Walker ('ywm'). You can use this method now by setting method='ywm'.
  warnings.warn(
```





In [22]:
```python
### split train and test data
df_airline
```

Out[22]:

Month	Thousands of Passengers	Passengers First Difference
1949-01-01	112.0	NaN
1949-02-01	118.0	6.0
1949-03-01	132.0	14.0
1949-04-01	129.0	-3.0
1949-05-01	121.0	-8.0
...	...	...
1960-08-01	606.0	-16.0
1960-09-01	508.0	-98.0
1960-10-01	461.0	-47.0
1960-11-01	390.0	-71.0
1960-12-01	432.0	42.0

144 rows × 2 columns

In [23]:
```python
from datetime import datetime,timedelta
train_dataset_end=datetime(1955,12,1)
test_dataset_end=datetime(1960,12,1)
```

In [24]:
```python
train_data=df_airline[:train_dataset_end]
test_data=df_airline[train_dataset_end+timedelta(days=1):test_dataset_end]
```

In [25]:
```python
##prediction
pred_start_date=test_data.index[0]
pred_end_date=test_data.index[-1]
```

In [26]:
```python
pred_start_date
```

Out[26]: Timestamp('1956-01-01 00:00:00')

In [27]:
```python
pred_end_date
```

Out[27]: Timestamp('1960-12-01 00:00:00')

In [28]:
```python
test_data
```

Out[28]:

Month	Thousands of Passengers	Passengers First Difference
1956-01-01	284.0	6.0
1956-02-01	277.0	-7.0
1956-03-01	317.0	40.0
1956-04-01	313.0	-4.0
1956-05-01	318.0	5.0
1956-06-01	374.0	56.0
1956-07-01	413.0	39.0
1956-08-01	405.0	-8.0
1956-09-01	355.0	-50.0
1956-10-01	306.0	-49.0
1956-11-01	271.0	-35.0
1956-12-01	306.0	35.0
1957-01-01	315.0	9.0
1957-02-01	301.0	-14.0
1957-03-01	356.0	55.0
1957-04-01	348.0	-8.0
1957-05-01	355.0	7.0
1957-06-01	422.0	67.0
1957-07-01	465.0	43.0
1957-08-01	467.0	2.0
1957-09-01	404.0	-63.0
1957-10-01	347.0	-57.0
1957-11-01	305.0	-42.0
1957-12-01	336.0	31.0
1958-01-01	340.0	4.0
1958-02-01	318.0	-22.0
1958-03-01	362.0	44.0
1958-04-01	348.0	-14.0
1958-05-01	363.0	15.0
1958-06-01	435.0	72.0
1958-07-01	491.0	56.0
1958-08-01	505.0	14.0
1958-09-01	404.0	-101.0
1958-10-01	359.0	-45.0
1958-11-01	310.0	-49.0
1958-12-01	337.0	27.0
1959-01-01	360.0	23.0
1959-02-01	342.0	-18.0
1959-03-01	406.0	64.0
1959-04-01	396.0	-10.0
1959-05-01	420.0	24.0
1959-06-01	472.0	52.0
1959-07-01	548.0	76.0
1959-08-01	559.0	11.0
1959-09-01	463.0	-96.0
1959-10-01	407.0	-56.0
1959-11-01	362.0	-45.0
1959-12-01	405.0	43.0
1960-01-01	417.0	12.0
1960-02-01	391.0	-26.0
1960-03-01	419.0	28.0
1960-04-01	461.0	42.0
1960-05-01	472.0	11.0
1960-06-01	535.0	63.0
1960-07-01	622.0	87.0
1960-08-01	606.0	-16.0
1960-09-01	508.0	-98.0
1960-10-01	461.0	-47.0
1960-11-01	390.0	-71.0
1960-12-01	432.0	42.0

In [29]:
```python
## create a ARIMA model
# from statsmodels.tsa.arima_model import ARIMA
from statsmodels.tsa.arima.model import ARIMA
```

In [30]:
```python
train_data
```

Out[30]:

Month	Thousands of Passengers	Passengers First Difference
1949-01-01	112.0	NaN
1949-02-01	118.0	6.0
1949-03-01	132.0	14.0
1949-04-01	129.0	-3.0
1949-05-01	121.0	-8.0
...	...	...
1955-08-01	347.0	-17.0
1955-09-01	312.0	-35.0
1955-10-01	274.0	-38.0
1955-11-01	237.0	-37.0
1955-12-01	278.0	41.0

84 rows × 2 columns

In [31]:
```python
model_ARIMA=ARIMA(train_data['Thousands of Passengers'],order=(14,1,12))
```
```
C:\Users\91760\anaconda3\lib\site-packages\statsmodels\tsa\base\tsa_model.py:471: ValueWarning: No frequency information was provided, so inferred frequency MS will be used.
  self._init_dates(dates, freq)
C:\Users\91760\anaconda3\lib\site-packages\statsmodels\tsa\base\tsa_model.py:471: ValueWarning: No frequency information was provided, so inferred frequency MS will be used.
  self._init_dates(dates, freq)
C:\Users\91760\anaconda3\lib\site-packages\statsmodels\tsa\base\tsa_model.py:471: ValueWarning: No frequency information was provided, so inferred frequency MS will be used.
  self._init_dates(dates, freq)
```

In [32]:
```python
model_Arima_fit=model_ARIMA.fit()
```
```
C:\Users\91760\anaconda3\lib\site-packages\statsmodels\tsa\statespace\sarimax.py:966: UserWarning: Non-stationary starting autoregressive parameters found. Using zeros as starting parameters.
  warn('Non-stationary starting autoregressive parameters'
C:\Users\91760\anaconda3\lib\site-packages\statsmodels\tsa\statespace\sarimax.py:978: UserWarning: Non-invertible starting MA parameters found. Using zeros as starting parameters.
  warn('Non-invertible starting MA parameters found.'
C:\Users\91760\anaconda3\lib\site-packages\statsmodels\base\model.py:604: ConvergenceWarning: Maximum Likelihood optimization failed to converge. Check mle_retvals
  warnings.warn("Maximum Likelihood optimization failed to "
```

In [33]:
```python
test_data
```

Out[33]:

Month	Thousands of Passengers	Passengers First Difference
1956-01-01	284.0	6.0
1956-02-01	277.0	-7.0
1956-03-01	317.0	40.0
1956-04-01	313.0	-4.0
1956-05-01	318.0	5.0
1956-06-01	374.0	56.0
1956-07-01	413.0	39.0
1956-08-01	405.0	-8.0
1956-09-01	355.0	-50.0
1956-10-01	306.0	-49.0
1956-11-01	271.0	-35.0
1956-12-01	306.0	35.0
1957-01-01	315.0	9.0
1957-02-01	301.0	-14.0
1957-03-01	356.0	55.0
1957-04-01	348.0	-8.0
1957-05-01	355.0	7.0
1957-06-01	422.0	67.0
1957-07-01	465.0	43.0
1957-08-01	467.0	2.0
1957-09-01	404.0	-63.0
1957-10-01	347.0	-57.0
1957-11-01	305.0	-42.0
1957-12-01	336.0	31.0
1958-01-01	340.0	4.0
1958-02-01	318.0	-22.0
1958-03-01	362.0	44.0
1958-04-01	348.0	-14.0
1958-05-01	363.0	15.0
1958-06-01	435.0	72.0
1958-07-01	491.0	56.0
1958-08-01	505.0	14.0
1958-09-01	404.0	-101.0
1958-10-01	359.0	-45.0
1958-11-01	310.0	-49.0
1958-12-01	337.0	27.0
1959-01-01	360.0	23.0
1959-02-01	342.0	-18.0
1959-03-01	406.0	64.0
1959-04-01	396.0	-10.0
1959-05-01	420.0	24.0
1959-06-01	472.0	52.0
1959-07-01	548.0	76.0
1959-08-01	559.0	11.0
1959-09-01	463.0	-96.0
1959-10-01	407.0	-56.0
1959-11-01	362.0	-45.0
1959-12-01	405.0	43.0
1960-01-01	417.0	12.0
1960-02-01	391.0	-26.0
1960-03-01	419.0	28.0
1960-04-01	461.0	42.0
1960-05-01	472.0	11.0
1960-06-01	535.0	63.0
1960-07-01	622.0	87.0
1960-08-01	606.0	-16.0
1960-09-01	508.0	-98.0
1960-10-01	461.0	-47.0
1960-11-01	390.0	-71.0
1960-12-01	432.0	42.0

In [34]:
```python
##prediction
pred_start_date=test_data.index[0]
pred_end_date=test_data.index[-1]
print(pred_start_date)
print(pred_end_date)
```
```
1956-01-01 00:00:00
1960-12-01 00:00:00
```

In [35]:
```python
pred=model_Arima_fit.predict(start=pred_start_date,end=pred_end_date)
residuals=test_data['Thousands of Passengers']-pred
```

In [36]:
```python
pred
```

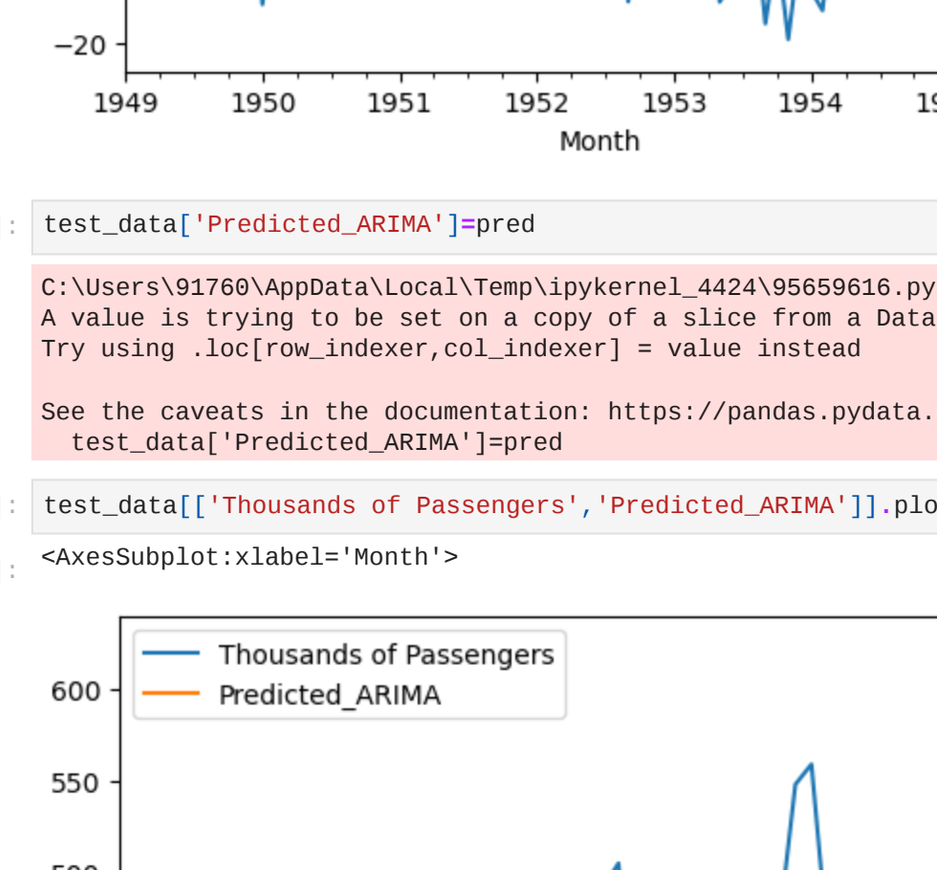
Out[36]:
```
1956-01-01    284.639205
1956-02-01    278.054575
1956-03-01    301.226591
1956-04-01    306.836632
1956-05-01    308.442140
1956-06-01    351.804401
1956-07-01    396.650587
1956-08-01    376.299066
1956-09-01    342.971653
1956-10-01    308.373345
1956-11-01    276.844533
1956-12-01    310.552293
1957-01-01    320.215422
1957-02-01    308.218373
1957-03-01    328.019315
1957-04-01    335.409903
1957-05-01    340.386959
1957-06-01    382.563300
1957-07-01    427.324345
1957-08-01    403.524115
1957-09-01    372.095659
1957-10-01    339.659961
1957-11-01    312.370056
1957-12-01    341.675937
1958-01-01    351.631652
1958-02-01    337.127370
1958-03-01    351.671542
1958-04-01    363.087079
1958-05-01    369.050302
1958-06-01    411.983312
1958-07-01    454.103594
1958-08-01    429.640760
1958-09-01    398.010438
1958-10-01    369.626886
1958-11-01    345.005550
1958-12-01    371.003830
1959-01-01    380.297198
1959-02-01    363.786451
1959-03-01    373.782443
1959-04-01    388.558038
1959-05-01    396.097221
1959-06-01    438.708806
1959-07-01    478.691191
1959-08-01    453.408654
1959-09-01    422.200623
1959-10-01    397.313394
1959-11-01    375.808218
1959-12-01    397.957026
1960-01-01    406.919297
1960-02-01    388.048292
1960-03-01    394.692404
1960-04-01    411.929508
1960-05-01    421.581576
1960-06-01    463.052825
1960-07-01    501.071497
1960-08-01    475.247341
1960-09-01    444.546088
1960-10-01    423.205228
1960-11-01    404.522195
1960-12-01    423.071924
Freq: MS, Name: predicted_mean, dtype: float64
```

In [37]:
```python
residuals
```

Out[37]:
```
Month
1956-01-01     -0.639205
1956-02-01     -1.054575
1956-03-01     15.773409
1956-04-01      6.163368
1956-05-01      9.557860
1956-06-01     22.195599
1956-07-01     16.349413
1956-08-01     28.700934
1956-09-01     12.028347
1956-10-01     -2.373345
1956-11-01     -5.844533
1956-12-01     -4.552293
1957-01-01     -5.215422
1957-02-01     -7.218373
1957-03-01     27.980685
1957-04-01     12.590097
1957-05-01     14.613041
1957-06-01     39.436700
1957-07-01     37.675655
1957-08-01     63.475885
1957-09-01     31.904341
1957-10-01      7.340039
1957-11-01     -7.370056
1957-12-01     -5.675937
1958-01-01    -11.631652
1958-02-01    -19.127370
1958-03-01     10.328458
1958-04-01    -15.087079
1958-05-01     -6.050302
1958-06-01     23.016688
1958-07-01     36.896406
1958-08-01     75.359240
1958-09-01      5.989562
1958-10-01    -10.626886
1958-11-01    -35.005550
1958-12-01    -34.003830
1959-01-01    -20.297198
1959-02-01    -21.786451
1959-03-01     32.217557
1959-04-01      7.441962
1959-05-01     23.902779
1959-06-01     33.291194
1959-07-01     69.308809
1959-08-01    105.591346
1959-09-01     40.799377
1959-10-01      9.686606
1959-11-01    -13.808218
1959-12-01      7.042974
1960-01-01     10.080703
1960-02-01      2.951708
1960-03-01     24.307596
1960-04-01     49.070492
1960-05-01     50.418424
1960-06-01     71.947175
1960-07-01    120.928503
1960-08-01    130.752659
1960-09-01     63.453912
1960-10-01     37.794772
1960-11-01    -14.522195
1960-12-01      8.928076
dtype: float64
```

In [38]:
```python
model_Arima_fit.resid.plot()
```

Out[38]: <AxesSubplot:xlabel='Month'>

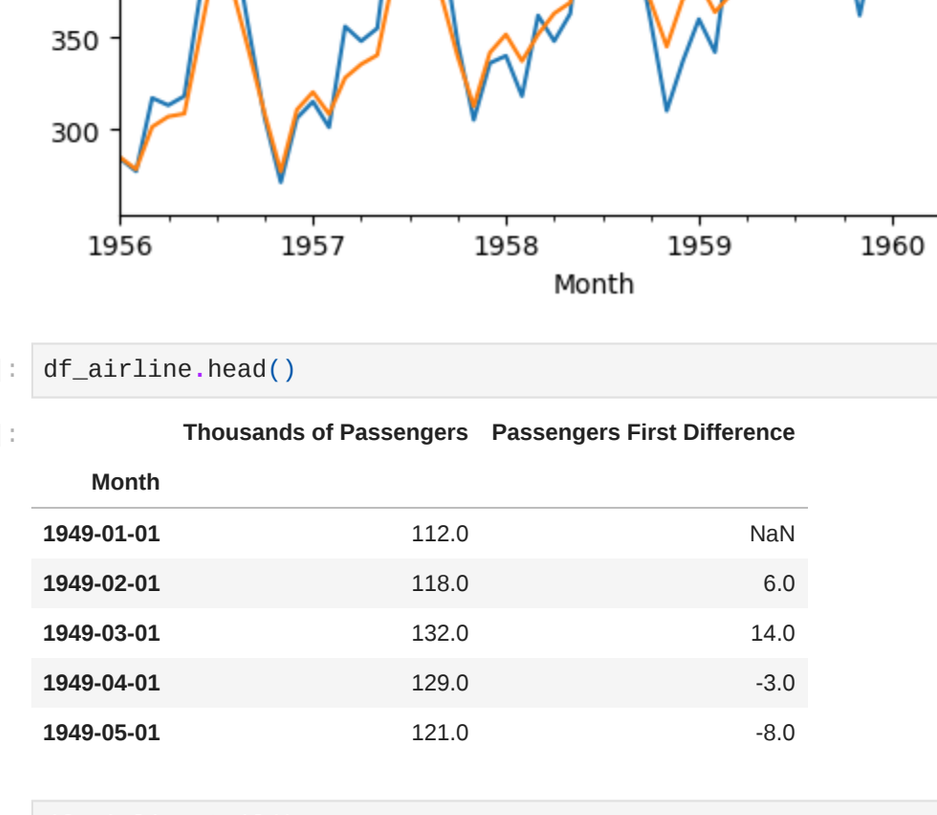


In [39]:
```python
test_data['Predicted_ARIMA']=pred
```
```
C:\Users\91760\AppData\Local\Temp\ipykernel_4424\95659616.py:1: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  test_data['Predicted_ARIMA']=pred
```

In [40]:
```python
test_data[['Thousands of Passengers','Predicted_ARIMA']].plot()
```

Out[40]: <AxesSubplot:xlabel='Month'>



In [41]:
```python
df_airline.head()
```

Out[41]:

Month	Thousands of Passengers	Passengers First Difference
1949-01-01	112.0	NaN
1949-02-01	118.0	6.0
1949-03-01	132.0	14.0
1949-04-01	129.0	-3.0
1949-05-01	121.0	-8.0

In [42]:
```python
df_airline.tail()
```

Out[42]:

Month	Thousands of Passengers	Passengers First Difference
1960-08-01	606.0	-16.0
1960-09-01	508.0	-98.0
1960-10-01	461.0	-47.0
1960-11-01	390.0	-71.0
1960-12-01	432.0	42.0

In [43]:
```python
from pandas.tseries.offsets import DateOffset
future_dates=[df_airline.index[-1]+ DateOffset(months=x)for x in range(0,25)]
```

In [44]:
```python
future_datest_df=pd.DataFrame(index=future_dates[1:],columns=df_airline.columns)
```

In [45]:
```python
future_datest_df.head()
```

Out[45]:

	Thousands of Passengers	Passengers First Difference
1961-01-01	NaN	NaN
1961-02-01	NaN	NaN
1961-03-01	NaN	NaN
1961-04-01	NaN	NaN
1961-05-01	NaN	NaN

In [46]:
```python
future_datest_df.tail()
```

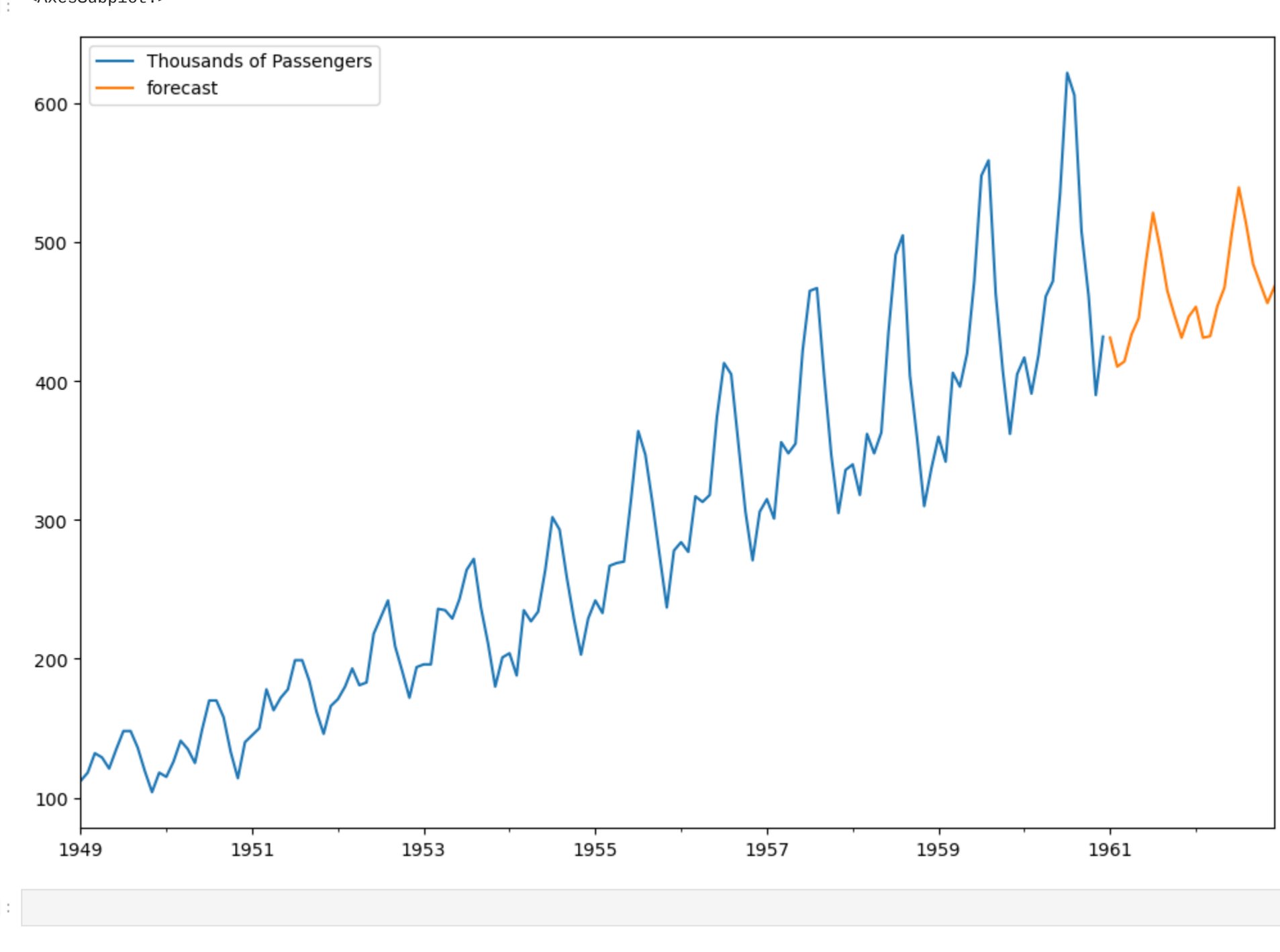
Out[46]:

	Thousands of Passengers	Passengers First Difference
1962-08-01	NaN	NaN
1962-09-01	NaN	NaN
1962-10-01	NaN	NaN
1962-11-01	NaN	NaN
1962-12-01	NaN	NaN

In [47]:
```python
future_df=pd.concat([df_airline,future_datest_df])
```

In [48]:
```python
future_df['forecast']=model_Arima_fit.predict(start='1961-01-01',end='1962-12-01')
future_df[['Thousands of Passengers', 'forecast']].plot(figsize=(12, 8))
```

Out[48]: <AxesSubplot:>



In [ ]: