

```

import pandas as pd
import numpy as np
import matplotlib.pyplot as plt

from statsmodels.tsa.stattools import adfuller
from statsmodels.tsa.arima.model import ARIMA
from statsmodels.graphics.tsaplots import plot_acf, plot_pacf

from sklearn.metrics import mean_absolute_error, mean_squared_error

df = pd.read_csv("airline_passenger_demand.csv")
df['Month'] = pd.to_datetime(df['Month'])
df.set_index('Month', inplace=True)

print(df.head())

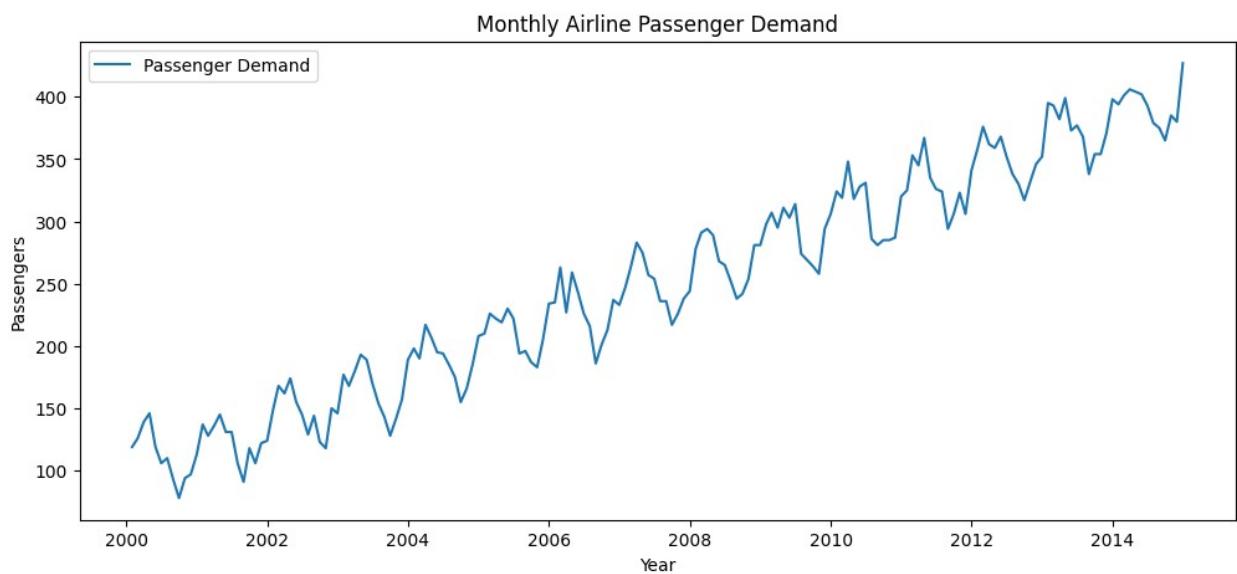
```

Passengers	
Month	Passengers
2000-01-31	119
2000-02-29	126
2000-03-31	139
2000-04-30	146
2000-05-31	119

```

plt.figure(figsize=(12,5))
plt.plot(df, label="Passenger Demand")
plt.title("Monthly Airline Passenger Demand")
plt.xlabel("Year")
plt.ylabel("Passengers")
plt.legend()
plt.show()

```



```

result = adfuller(df['Passengers'])

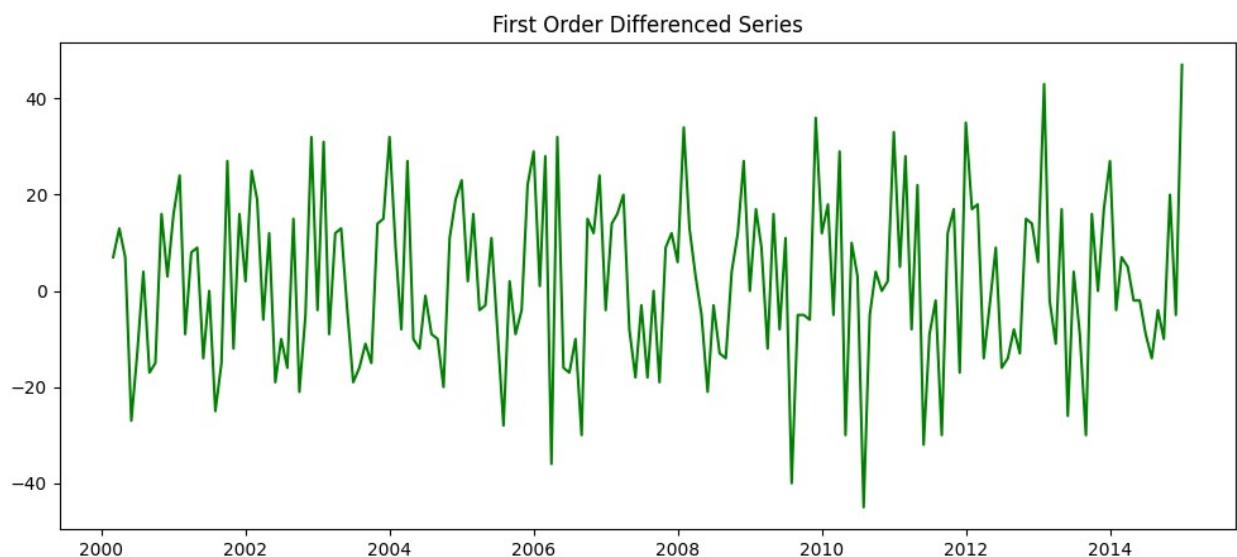
print("ADF Statistic:", result[0])
print("p-value:", result[1])

ADF Statistic: 0.6332633976013743
p-value: 0.9884074541056656

df_diff = df.diff().dropna()

plt.figure(figsize=(12,5))
plt.plot(df_diff, color="green")
plt.title("First Order Differenced Series")
plt.show()

```



```

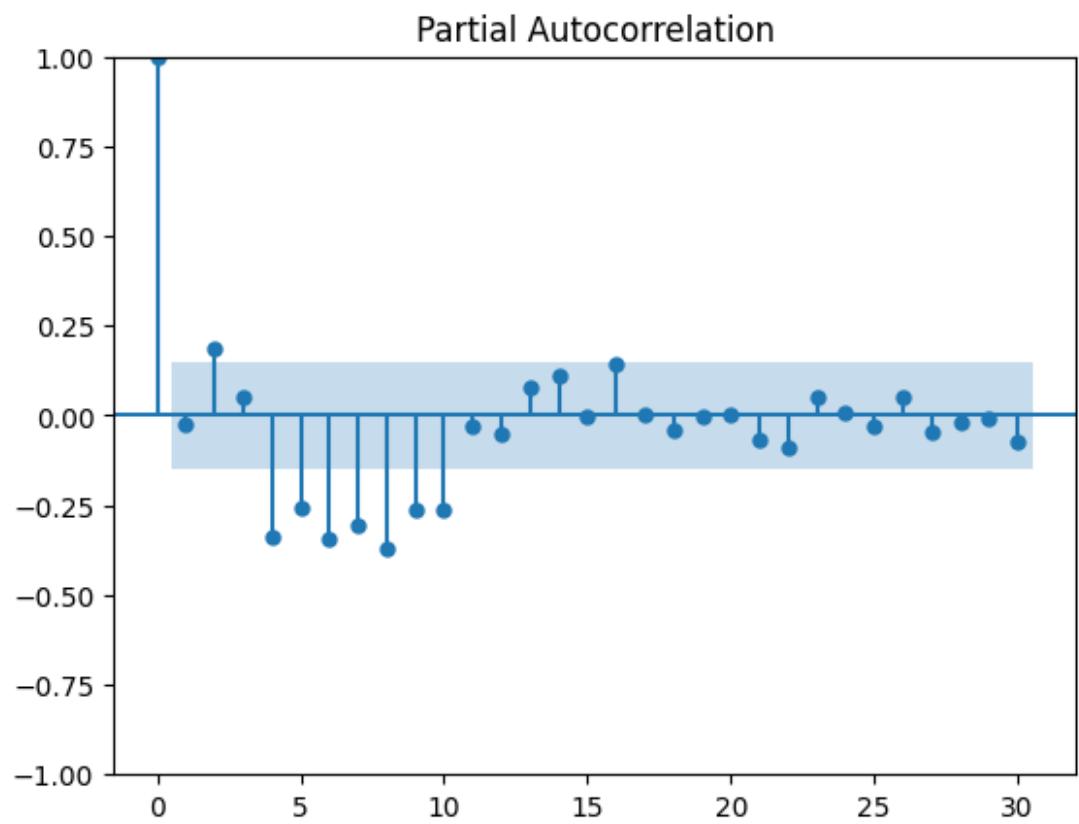
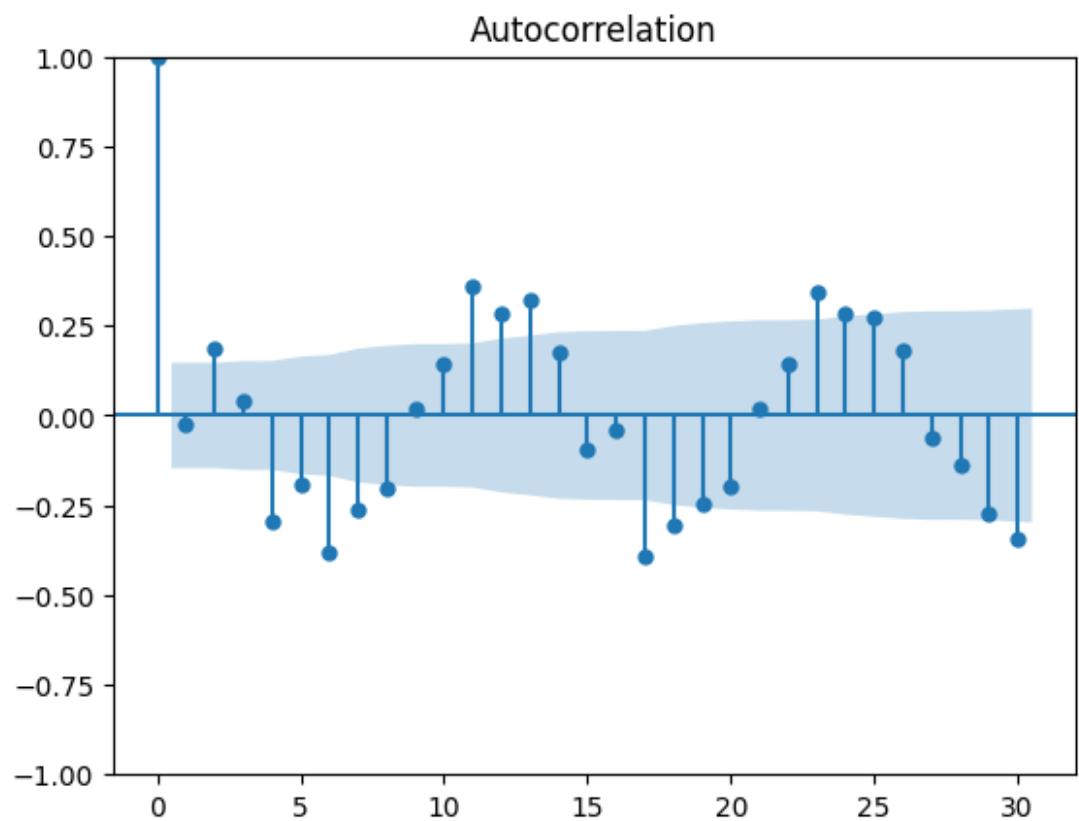
adf_diff = adfuller(df_diff['Passengers'])
print("p-value after differencing:", adf_diff[1])

p-value after differencing: 3.0120854952707844e-15

plot_acf(df_diff, lags=30)
plt.show()

plot_pacf(df_diff, lags=30)
plt.show()

```



```

train = df.iloc[:-12]
test = df.iloc[-12:]
model = ARIMA(train, order=(1,1,1))
model_fit = model.fit()

print(model_fit.summary())

```

C:\Users\revan\miniconda3\Lib\site-packages\statsmodels\tsa\base\tsa\_model.py:473: ValueWarning: No frequency information was provided, so inferred frequency ME will be used.  
self.\_init\_dates(dates, freq)  
C:\Users\revan\miniconda3\Lib\site-packages\statsmodels\tsa\base\tsa\_model.py:473: ValueWarning: No frequency information was provided, so inferred frequency ME will be used.  
self.\_init\_dates(dates, freq)  
C:\Users\revan\miniconda3\Lib\site-packages\statsmodels\tsa\base\tsa\_model.py:473: ValueWarning: No frequency information was provided, so inferred frequency ME will be used.  
self.\_init\_dates(dates, freq)

### SARIMAX Results

```

=====
=====
Dep. Variable:           Passengers    No. Observations:      168
Model:                 ARIMA(1, 1, 1)    Log Likelihood:   -714.343
Date:                 Thu, 05 Feb 2026   AIC:                1434.686
1Time:                  10:05:09        BIC:                1444.040
Sample:                01-31-2000     HQIC:               1438.482
- 12-31-2013

```

Covariance Type: opg

```

=====
=====
            coef    std err        z     P>|z|      [0.025
0.975]
-----
ar.L1      0.4039     1.778     0.227      0.820     -3.082
3.889
ma.L1     -0.3673     1.814    -0.202      0.840     -3.923
3.188
sigma2    304.0269    38.592     7.878      0.000     228.388
379.666

```

```

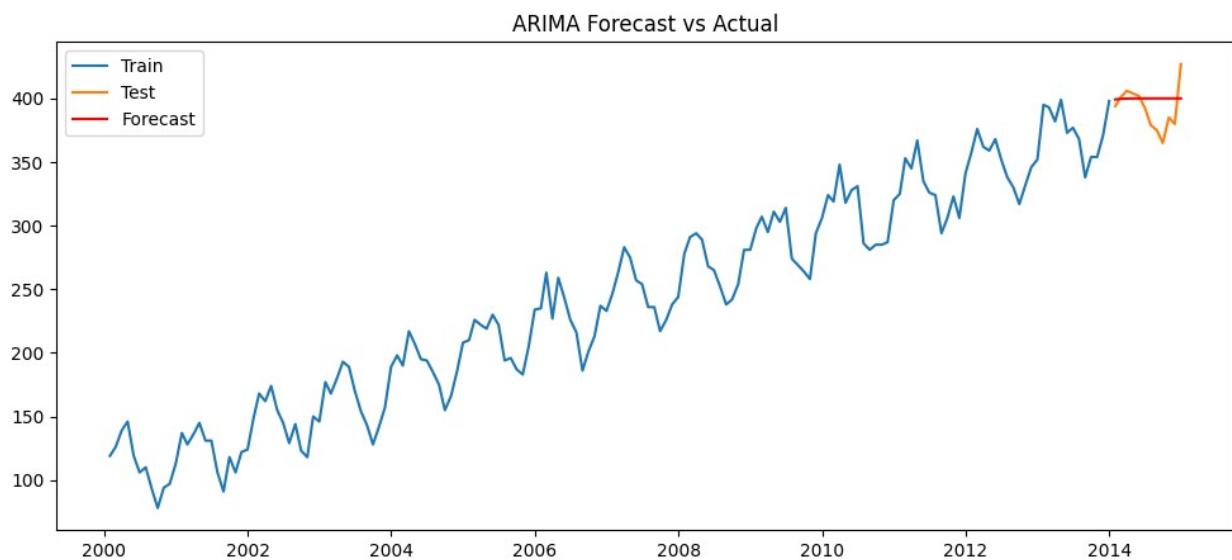
=====
=====
Ljung-Box (L1) (Q):          0.58   Jarque-Bera (JB):
1.58                         0.45   Prob(JB):
Prob(Q):                      0.45
Heteroskedasticity (H):      1.50   Skew:
-0.07                         0.13   Kurtosis:
Prob(H) (two-sided):        2.54
=====
```

Warnings:

```
[1] Covariance matrix calculated using the outer product of gradients
(complex-step).
```

```
forecast = model_fit.forecast(steps=12)
```

```
plt.figure(figsize=(12,5))
plt.plot(train, label="Train")
plt.plot(test, label="Test")
plt.plot(forecast, label="Forecast", color="red")
plt.legend()
plt.title("ARIMA Forecast vs Actual")
plt.show()
```



```
mae = mean_absolute_error(test, forecast)
rmse = np.sqrt(mean_squared_error(test, forecast))

print("MAE:", mae)
print("RMSE:", rmse)
```

```

MAE: 14.067341051009398
RMSE: 17.770626064047264

final_model = ARIMA(df, order=(1,1,1))
final_fit = final_model.fit()

future_forecast = final_fit.forecast(steps=24)

plt.figure(figsize=(12,5))
plt.plot(df, label="Historical Data")
plt.plot(future_forecast, label="Future Forecast", color="purple")
plt.legend()
plt.title("Future Airline Passenger Demand Forecast")
plt.show()

C:\Users\revan\miniconda3\Lib\site-packages\statsmodels\tsa\base\
tsa_model.py:473: ValueWarning: No frequency information was provided,
so inferred frequency ME will be used.
    self._init_dates(dates, freq)
C:\Users\revan\miniconda3\Lib\site-packages\statsmodels\tsa\base\
tsa_model.py:473: ValueWarning: No frequency information was provided,
so inferred frequency ME will be used.
    self._init_dates(dates, freq)
C:\Users\revan\miniconda3\Lib\site-packages\statsmodels\tsa\base\
tsa_model.py:473: ValueWarning: No frequency information was provided,
so inferred frequency ME will be used.
    self._init_dates(dates, freq)

```

