```
import pandas as pd
import numpy as np
from statsmodels.tsa.stattools import adfuller
import statsmodels
from statsmodels.graphics.tsaplots import plot_acf,plot_pacf
import matplotlib.pyplot as plt
import statsmodels.api as sm
from statsmodels.tsa.arima.model import ARIMA

data = pd.read_csv('C:\\Users\\PandaBas\\Desktop\\Personal\\Teaching\\Time_Series\\monthly_agg_sales.csv')
data.head()


# train test split

train = data.head(42)
test = data.tail(6)


# Stationarity check

result = adfuller(train.Sales.dropna())
print('ADF Statistic: %f' % result[0])
print('p-value: %f' % result[1])


# Create time series objects
train.set_index('month',inplace = True)
test.set_index('month',inplace = True)


# create lag variables

train['lag1'] = train['Sales'].shift(1)


# check stationarty for lag

result = adfuller(train.lag1.dropna())
print('ADF Statistic: %f' % result[0])
print('p-value: %f' % result[1])


# Plot ACF and PACF

fig = plt.figure(figsize=(12,8))
ax1 = fig.add_subplot(211)
fig = plot_acf(train['lag1'].dropna(),lags=15,ax=ax1)
```

```
ax2 = fig.add_subplot(212)
fig = plot_pacf(train['lag1'].dropna(),lags=15,ax=ax2)


# Fit the model

model = ARIMA(train.Sales, order=(1,0,1))
model_fit = model.fit()
print(model_fit.summary())


# make predictions
predictions = model_fit.forecast(6)

# Check accuracy
print(f'ARIMA Model Test Data MSE: {np.mean((predictions.values -
test.Sales)**2):.3f}')
print(f'ARIMA Model Test Data MAPE: {np.mean(np.abs(test.Sales -
predictions.values)/test.Sales )*100:.3f}')
```