```python
# -*- coding: utf-8 -*-
"""
Created on Wed Jan 19 08:32:58 2022

@author: miriajm
"""

#Install Pandas-datareader if you haven't already - Mine was installed through Anaconda
#pip install pandas_datareader
#pip install yfinance
#pip install plotly

#Import required packages
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import numpy as np
from sklearn.metrics import mean_squared_error

#Adjust the limits on rows and columns for this analysis
pd.set_option('display.max_columns', None)
pd.set_option('display.max_rows', None)

#import Bitcoin pricing from csv
data = pd.read_csv("C:/Users/miriajm/OneDrive - Oklahoma A and M System/5743 BAN/Spring 2022/Week 2/BTC-USD.csv")
print(data.head())

#selecting specific columns
"""
The inner square brackets define a Python list with column names,
whereas the outer brackets are used to select the data from a pandas DataFrame as seen in the previous example.
"""
tsdata=data[["Date", "Close"]]
tsdata.index = pd.to_datetime(tsdata['Date'], format='%m/%d/%Y')

#plotting time series data
sns.set()
plt.ylabel('BTC Closing Price')
plt.xlabel('Date')
plt.xticks(rotation=45)
plt.plot(tsdata.index, tsdata['Close'], )


#Splitting data into Train and Validate
train = tsdata[tsdata.index < pd.to_datetime("11/01/2021", format='%m/%d/%Y')]
validate = tsdata[tsdata.index > pd.to_datetime("11/01/2021", format='%m/%d/%Y')]

plt.plot(train.index, train['Close'], color = "black")
plt.plot(validate.index, validate['Close'],  color = "red")
plt.ylabel('BTC Closing Price')
plt.xlabel('Date')
plt.xticks(rotation=45)
plt.title("Train/Validate split for BTC Data")
plt.show()


#ARMA model
from statsmodels.tsa.statespace.sarimax import SARIMAX

#input
y = train['Close']

#model
ARMAmodel = SARIMAX(y, order = (1, 0, 1))
```

```
65
66  #fit model
67  ARMAmodel = ARMAmodel.fit()
68
69  #generate predictions
70  y_pred = ARMAmodel.get_forecast(len(validate.index))
71  y_pred_df = y_pred.conf_int(alpha = 0.05)
72  y_pred_df["Predictions"] = ARMAmodel.predict(start = y_pred_df.index[0], end =
    y_pred_df.index[-1])
73  y_pred_df.index = validate.index
74  y_pred_out = y_pred_df["Predictions"]
75
76  #plot results
77  plt.plot(train.index, train['Close'], color = "black")
78  plt.plot(validate.index, validate['Close'],  color = "red")
79  plt.plot(y_pred_out, color='green', label = 'Predictions')
80  plt.ylabel('BTC Closing Price')
81  plt.xlabel('Date')
82  plt.xticks(rotation=45)
83  plt.title("Train/Validate split for BTC Data")
84  plt.legend()
85  plt.show()
86
87  #Evaluate Model Performance
88  arma_rmse = np.sqrt(mean_squared_error(validate["Close"].values,
    y_pred_df["Predictions"]))
89  print("RMSE: ",arma_rmse)
90
91
92
93
94
95
96
```