import pandas as pd

import datetime as dt

import numpy as np

import matplotlib.pyplot as plt

import scipy.stats

import statistics

import statsmodels.api as sm

from scipy.stats import skew, skewtest, norm

from scipy.stats import kurtosis, skewnorm

import scipy.stats as stats

from scipy.stats import boxcox

from statsmodels.tsa.stattools import adfuller

from statsmodels.graphics import tsaplots

from statsmodels.graphics.tsaplots import plot\_acf

from statsmodels.graphics.tsaplots import plot\_pacf

import seaborn as sns

import yfinance as yf

import warnings

warnings.simplefilter("ignore")

plt.style.use('ggplot')

mean = amzn\_close.mean()

std = amzn\_close.std()

x = np.linspace(mean - 3\*std, mean + 3\*std, 100)

plt.figure(figsize=(8, 5))

sns.distplot(amzn\_close, hist=False, label="Empirical distribution of Amazon close prices")

plt.plot(x, scipy.stats.norm.pdf(x, mean, std), label="Gaussian distribution of Amazon close prices", color='k')

# Add Mean

plt.axvline(x=statistics.mean(amzn\_close), color='y')

plt.text(statistics.mean(amzn\_close), 0.001, "Mean", rotation=90, color='y')

# Add Median

plt.axvline(x=statistics.median(amzn\_close), color='b')

plt.text(statistics.median(amzn\_close), 0.002, "Median", rotation=90, color='b')

# Add mode

plt.axvline(x=statistics.mode(amzn\_close), color='g')

plt.text(statistics.mode(amzn\_close), 0.003, "Mode", rotation=90, color='g')

plt.legend(loc='upper center', bbox\_to\_anchor=(0.5, 1.05), ncol=2, fancybox=True, shadow=True)

plt.show()

mean = statistics.mean(amzn\_close)

median = statistics.median(amzn\_close)

mode = statistics.mode(amzn\_close)

print('\n mean: ', mean, '\n median: ', median, '\n mode: ',mode)

amzn\_skew = skew(amzn\_close, axis=0, bias=True)

print('skewness: ', amzn\_skew)

amzn\_close.isna().sum()

log\_amzn\_close = np.log(meta\_close)

log\_amzn\_close.head()

mean = log\_amzn\_close.mean()

std = log\_amzn\_close.std()

x = np.linspace(mean - 3\*std, mean + 3\*std, 100)

plt.figure(figsize=(8, 5))

sns.distplot(log\_amzn\_close, hist=False, label="Distribution of Log AMZN close prices")

plt.plot(x, scipy.stats.norm.pdf(x, mean, std), label="Gaussian distribution of Log AMZN close prices", color='k')

# Add Mean

plt.axvline(x=statistics.mean(log\_amzn\_close), color='y')

plt.text(statistics.mean(log\_amzn\_close), 0.001, "Mean", rotation=90, color='y')

# Add Median

plt.axvline(x=statistics.median(log\_amzn\_close), color='b')

plt.text(statistics.median(log\_amzn\_close), 0.002, "Median", rotation=90, color='b')

# Add mode

plt.axvline(x=statistics.mode(log\_amzn\_close), color='g')

plt.text(statistics.mode(log\_amzn\_close), 0.003, "Mode", rotation=90, color='g')

plt.legend(loc='upper center', bbox\_to\_anchor=(0.5, 1.05), ncol=2, fancybox=True, shadow=True)

plt.show()

log\_skew = skew(log\_amzn\_close, axis=0, bias=True)

print('before log transform skewness: ', amzn\_skew)

print('after log transform skewness: ', log\_skew)

bc\_amzn\_close, bc\_lambda = boxcox(amzn\_close)

print('best lambda parameter in box-cox method: ', bc\_lambda)

mean = bc\_amzn\_close.mean()

std = bc\_amzn\_close.std()

x = np.linspace(mean - 3\*std, mean + 3\*std, 100)

plt.figure(figsize=(8, 5))

sns.distplot(bc\_amzn\_close, hist=False, label="Distribution of box-cox AMZN close prices")

plt.plot(x, scipy.stats.norm.pdf(x, mean, std), label="Gaussian distribution of box-cox AMZN close prices", color='k')

# Add Mean

plt.axvline(x=statistics.mean(bc\_amzn\_close), color='y')

plt.text(statistics.mean(bc\_amzn\_close), 0.0001, "Mean", rotation=90, color='y')

# Add Median

plt.axvline(x=statistics.median(bc\_amzn\_close), color='b')

plt.text(statistics.median(bc\_amzn\_close), 0.0002, "Median", rotation=90, color='b')

# Add mode

plt.axvline(x=statistics.mode(bc\_amzn\_close), color='g')

plt.text(statistics.mode(bc\_amzn\_close), 0.0003, "Mode", rotation=90, color='g')

plt.legend(loc='upper center', bbox\_to\_anchor=(0.5, 1.05), ncol=2, fancybox=True, shadow=True)

plt.show()

print("Skew before box cox Transformation: %f" % skew(amzn\_close))

print("Skew after box cox Transformation: %f" % skew(bc\_amzn\_close))

start\_date = dt.date(2010, 1, 1)

end\_date = dt.date(2023, 10, 31)

AAPL\_data = yf.download(tickers = "AAPL" ,start = start\_date, end = end\_date)

AAPL\_data.head()

AAPL\_data["Volume"].plot(subplots=True, figsize=(10,12), color="blue")

plt.title('Volumes of Apple stock from 2010 to 2023')

plt.show()

AAPL\_data["Close"].plot(subplots=True, figsize=(10,12), color="blue")

plt.title('Close prices of Apple stock from 2010 to 2023')

plt.show()

adf = adfuller(AAPL\_data["Volume"])

print("p-value of volumes of AAPL shares: {}".format(float(adf[1])))

adf = adfuller(AAPL\_data["Close"])

print("p-value of close price of Apple shares: {}".format(float(adf[1])))

AAPL\_data["diff\_Close"] = AAPL\_data["Close"].diff().fillna(0)

adf = adfuller(AAPL\_data["diff\_Close"])

print("p-value of lagged difference of Apple close prices: {}".format(float(adf[1])))

AAPL\_data["diff\_Close"].plot();

plot\_acf(AAPL\_data["Volume"]);

plot\_pacf(AAPL\_data["Volume"]);

plot\_acf(AAPL\_data["diff\_Close"]);

plot\_pacf(AAPL\_data["diff\_Close"]);

import yfinance as yf

from scipy.stats import jarque\_bera

import pandas as pd

ticker = "META"

start\_date = "2020-09-01"

end\_date = "2023-10-31"

stock\_data = yf.download(ticker, start=start\_date, end=end\_date)

stock\_data['Daily\_Return'] = stock\_data['Adj Close'].pct\_change().dropna()

returns = stock\_data['Daily\_Return'].dropna()

statistic, p\_value = jarque\_bera(returns)

print(f"Jarque-Bera Test Statistic: {statistic}")

print(f"P-value: {p\_value}")

alpha = 0.05

if p\_value < alpha:

print("The null hypothesis is rejected. The data does not follow a normal distribution.")

else:

print("Fail to reject the null hypothesis. The data follows a normal distribution.")

import numpy as np

import matplotlib.pyplot as plt

from scipy.stats import kurtosis

from datetime import datetime

# Calculate excess kurtosis

excess\_kurt = kurtosis(returns, fisher=True)

# Plot the returns distribution

plt.figure(figsize=(10, 6))

plt.hist(returns, bins=50, density=True, alpha=0.75)

plt.title("Distribution of META Stock Returns")

plt.xlabel("Daily Returns")

plt.ylabel("Frequency")

plt.show()

# Display excess kurtosis

excess\_kurt

import numpy as np

import matplotlib.pyplot as plt

import yfinance as yf

from scipy.stats import kurtosis, boxcox

from scipy.special import inv\_boxcox

# Box-Cox Transformation

returns\_boxcox, lambda\_value = boxcox(returns + 1) # Adding 1 to handle zero or negative values

# Plot the transformed data

plt.figure(figsize=(10, 6))

plt.hist(returns\_boxcox, bins=50, density=True, alpha=0.75)

plt.title("Distribution of Box-Cox Transformed META Stock Returns")

plt.xlabel("Transformed Daily Returns")

plt.ylabel("Frequency")

plt.show()

# Inverse Box-Cox Transformation (if needed)

returns\_inverse\_boxcox = inv\_boxcox(returns\_boxcox, lambda\_value)

# Calculate excess kurtosis

excess\_kurt = kurtosis(returns, fisher=True)