ResearchProject

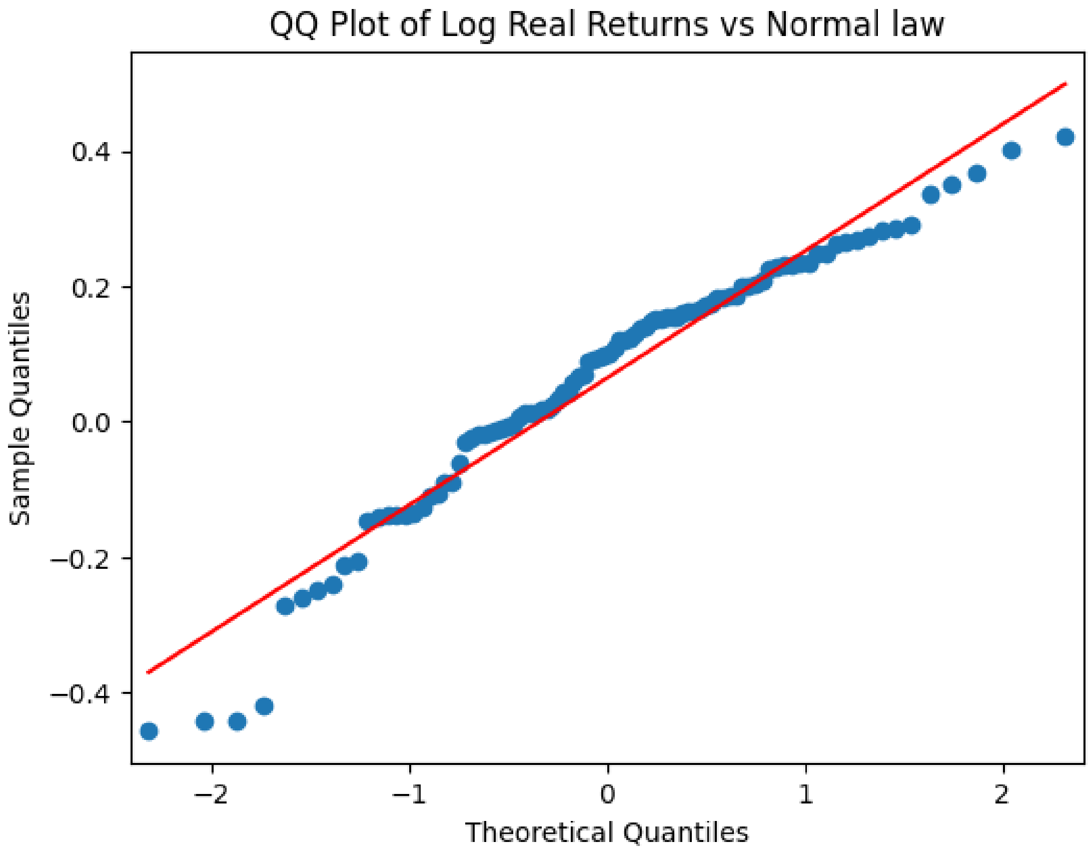
October 28, 2024

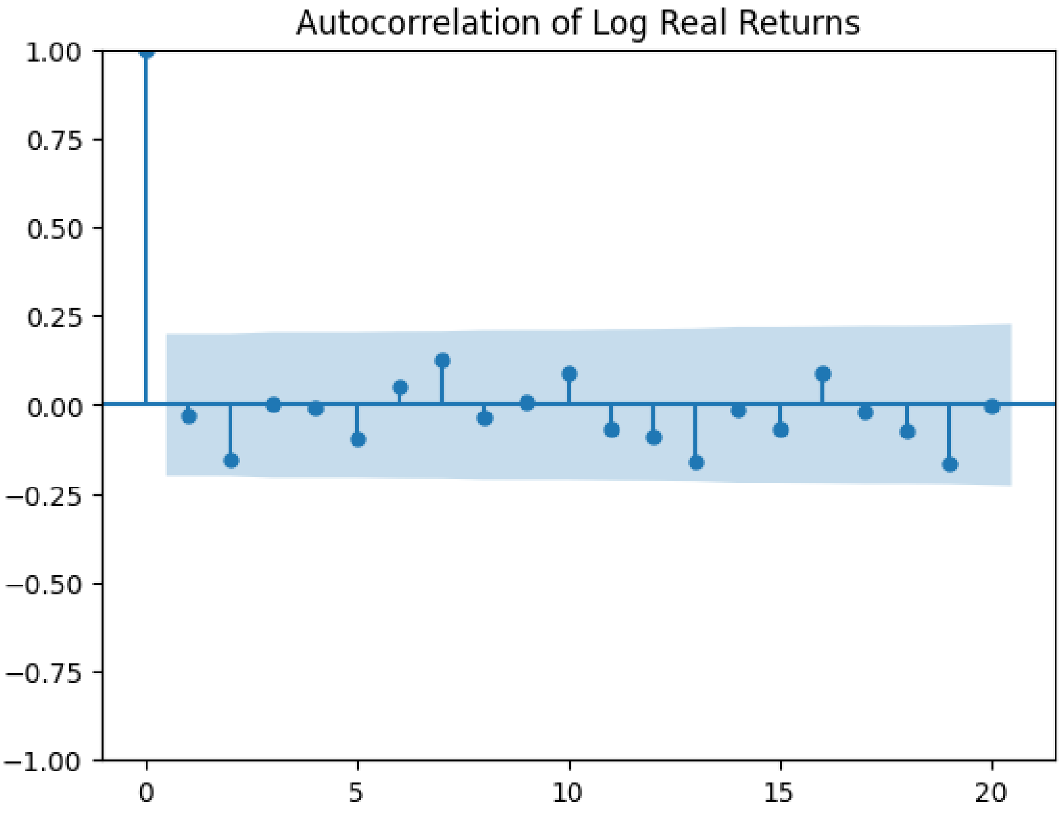
[12]:

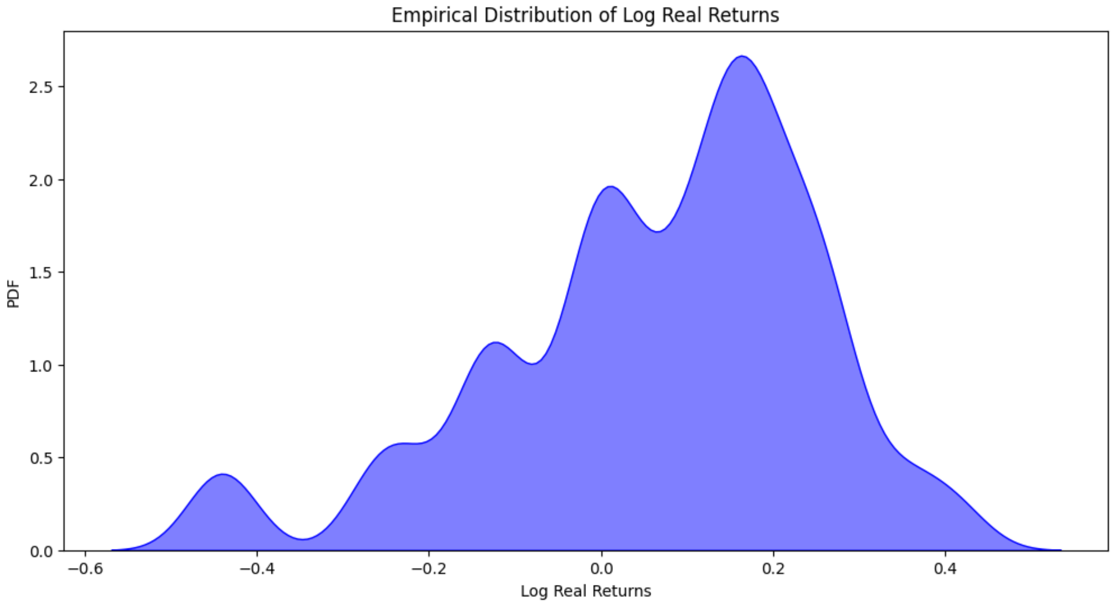
```
import pandas as pd
import numpy as np
import math
from statsmodels.graphics.gofplots import qqplot
from statsmodels.graphics.tsaplots import plot_acf
from matplotlib import pyplot as plt
import seaborn as sns

#Gathering data
cols = ['year','DecCPI']
CPIdata = pd.read_excel('DecCPIData.xlsx', names = cols)
cols = ['year', 'Div']
DividendsData = pd.read_excel('DividendDataS&P.xlsx', names = cols)
cols = ['date', 'ClosingPrice']
ClosingPriceData = pd.read_excel('Dec31S&Pdata.xlsx', names = cols)


CPIarray = CPIdata['DecCPI']


inflation = np.zeros(96)
#calculating c1/c0
for i in range (len(inflation)):
    inflation[i] = CPIarray[i]/CPIarray[i+1]


#Calculating s1+D/s0
closingPriceArray = ClosingPriceData['ClosingPrice']
dividendsArray = DividendsData['Div']
returns = np.zeros(96)
for j in range (len(returns)):
    returns[j] = ((closingPriceArray[j] + dividendsArray[j])/
 ↪closingPriceArray[j+1])


logRealReturn = np.zeros(96)
for k in range (len(logRealReturn)):
    logRealReturn[k] = math.log(returns[k]) - math.log(inflation[k])

logNominalReturn = np.zeros(96)
for l in range (len(logNominalReturn)):
    logNominalReturn[l] = math.log(returns[l])

qqplot(logRealReturn, line = 's')
plt.title('QQ Plot of Log Real Returns vs Normal law')
plt.show


plot_acf(logRealReturn)
plt.title('Autocorrelation of Log Real Returns')
plt.show

plt.figure(figsize =(12,6))
sns.kdeplot(logRealReturn, bw_adjust=0.5, fill=True, color='blue', alpha=0.5)
plt.title('Empirical Distribution of Log Real Returns')
plt.xlabel('Log Real Returns')
plt.ylabel('PDF')
plt.show

qqplot(logNominalReturn, line = 's')
plt.title('QQ Plot of Log Nominal Returns vs Normal Law')
plt.show

plot_acf(logNominalReturn)
plt.title('Autocorrelation of Log Nominal Returns')
plt.show

plt.figure(figsize =(12,6))
sns.kdeplot(logNominalReturn, bw_adjust=0.5, fill=True, color='blue', alpha=0.5)
plt.title('Empirical Distribution of Log Nominal Returns')
plt.xlabel('Log Nominal Returns')
plt.ylabel('PDF')
plt.show

year = 2023
yearDataFrame = np.zeros(96)
for l in range (len(yearDataFrame)):
    yearDataFrame[l] = year - l
excelExport = pd.DataFrame({
    'Year' : yearDataFrame,
    'Log Real Returns' : logRealReturn,
    'Log Nominal Returns' : logNominalReturn
})
print(excelExport)
```

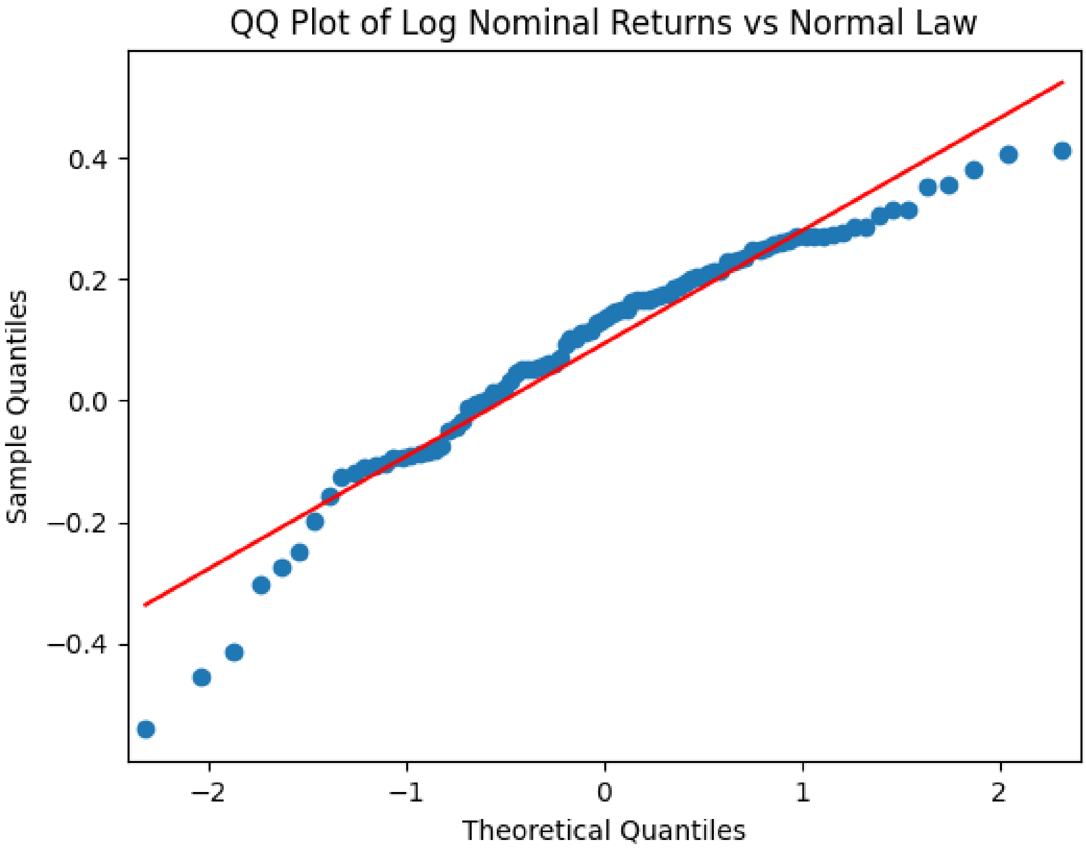
```
excelExport.to_excel('S&P500LogRealReturns.xlsx', index = False)
```

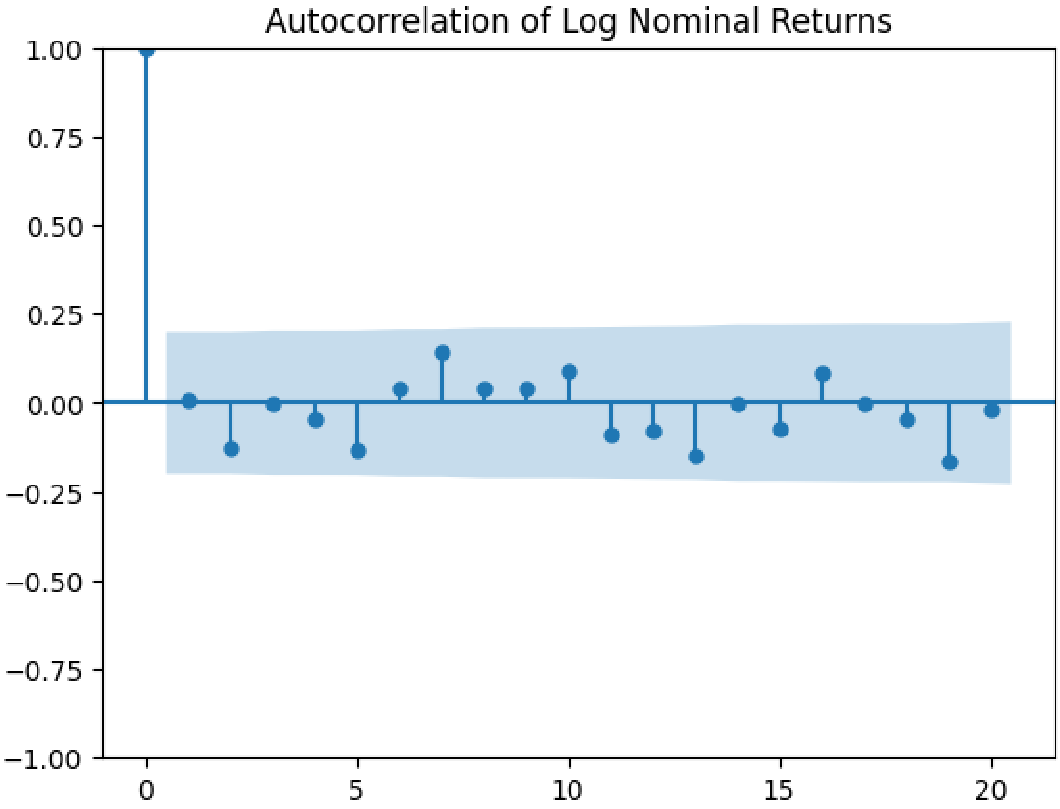
```
      Year  Log Real Returns  Log Nominal Returns
0   2023.0          0.198628             0.231599
1   2022.0         -0.261470            -0.198924
2   2021.0          0.182872             0.250871
3   2020.0          0.152519             0.166047
4   2019.0          0.248969             0.271563
..     …                 …                    …
91  1932.0          0.018258            -0.090151
92  1931.0         -0.442130            -0.539928
93  1930.0         -0.207243            -0.273333
94  1929.0         -0.088409            -0.082578
95  1928.0          0.367170             0.355542

[96 rows x 3 columns]
```

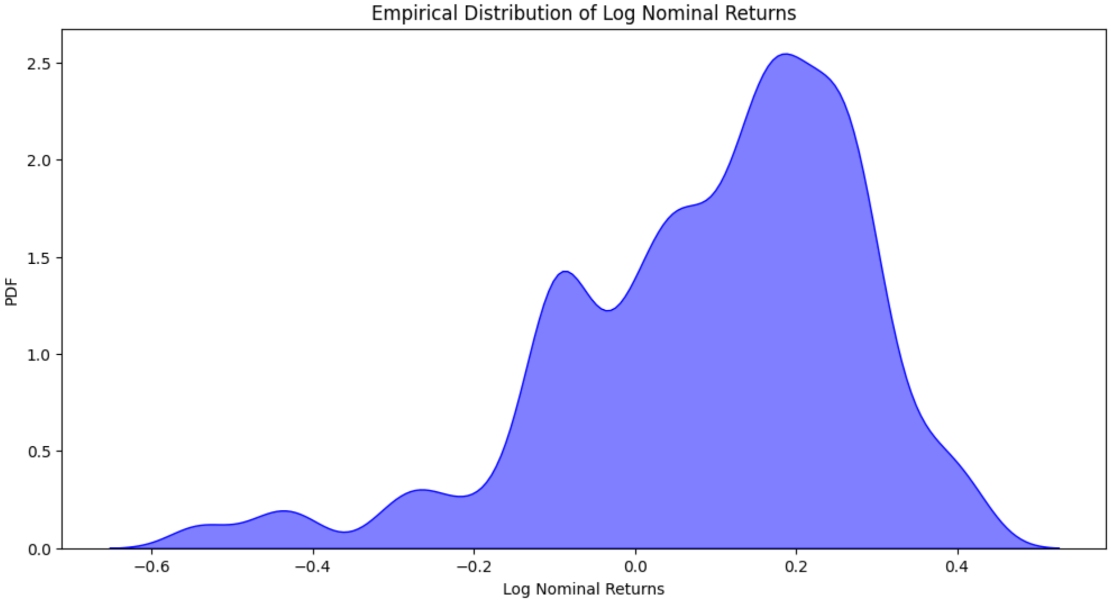



Autocorrelation of Log Real Returns
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.00
0
5
10
15
20


Empirical Distribution of Log Real Returns
PDF
2.5
2.0
1.5
1.0
0.5
0.0
−0.6
−0.4
−0.2
0.0
0.2
0.4
Log Real Returns


QQ Plot of Log Nominal Returns vs Normal Law
Sample Quantiles
0.4
0.2
0.0
−0.2
−0.4
−2
−1
0
1
2
Theoretical Quantiles


Autocorrelation of Log Nominal Returns
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.00
0
5
10
15
20


Empirical Distribution of Log Nominal Returns
PDF
2.5
2.0
1.5
1.0
0.5
0.0
−0.6
−0.4
−0.2
0.0
0.2
0.4
Log Nominal Returns

[ ]: