Assignment 7

In [1]:
```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
from tiingo import TiingoClient
import numpy as np
from datetime import date
import warnings
warnings.filterwarnings('ignore')
from dateutil.relativedelta import relativedelta
config = {}

config['session'] = True

config['api_key'] = "110ee73e29ec4269f49eb85cfb4b976ab8e73361"

client = TiingoClient(config)
```

In [14]:
```
def download_financial_data(ticker):
    fin_data = client.get_ticker_price(ticker,
                                       fmt='csv',
                                       startDate = date.today() - relativedelta(years=5),
                                       endDate = date.today(),
                                       frequency = 'daily')
    file_name = f"{ticker}.csv"
    with open(file_name,'w') as outfile:
        outfile.write(fin_data)
    print(f'{ticker}.csv created')
    return pd.read_csv(f"{ticker}.csv")
```

In [15]:
```
df = download_financial_data("SPY")
```

```
SPY.csv created
```

In [16]:
```
df.head()
```

Out[16]:

	date	close	high	low	open	volume	adjClose	adjHigh	adjLow	adjOpen	adjVolume	divCash	splitF
0	2017-05-31	241.440002	241.880005	240.639999	241.839996	91796000	221.138419	221.541424	220.405685	221.504779	91796000	0.0	
1	2017-06-01	243.360001	243.380005	241.639999	241.970001	68962000	222.896974	222.915296	221.321599	221.623853	68962000	0.0	
2	2017-06-02	244.169998	244.350006	243.080002	243.419998	88666100	223.638862	223.803734	222.640519	222.951926	88666100	0.0	
3	2017-06-05	243.990005	244.300003	243.759995	243.970001	44698800	223.474004	223.757936	223.263334	223.455682	44698800	0.0	
4	2017-06-06	243.210007	243.979996	243.119995	243.339996	50375400	222.759592	223.464837	222.677149	222.878651	50375400	0.0	

In [32]:
```
def Calculate_Statistics(df,ticker):
    df['Close_lag'] = df['close'].shift(periods=1)
    df['date'] = pd.to_datetime(df['date'])

    df['ret_daily'] = df['close'] / df['Close_lag']
    df['ret_pct'] = (df['ret_daily'] - 1.0) * 100

    df['Close_lag252'] = df['close'].shift(periods=252)
    df['ret_annual'] = df['close'] / df['Close_lag252']
    df['ret_daily_ln'] = np.log(df['ret_daily'])


    sd = df['ret_pct'].std(ddof=0)
    mean = df['ret_pct'].mean()
    print(f'68% of the daily returns for {ticker} will be between {mean - sd:.4f}% and {mean + sd:.4f}%')
    return df['ret_pct'].describe()
```

In [29]:
```
Calculate_Statistics(df,"SPY")
```

```
68% of the daily returns for SPY will be between -1.2134% and 1.3147%
```

Out[29]:
```
count    1259.000000
mean        0.050678
std         1.264559
min       -10.942373
25%        -0.370693
50%         0.086201
75%         0.627458
max         9.060327
Name: ret_pct, dtype: float64
```

In [30]:
```
df2 = download_financial_data("FB")
Calculate_Statistics(df2,"FB")
```

```
FB.csv created
68% of the daily returns for FB will be between -2.3639% and 2.4627%
```

Out[30]:
```
count    1259.000000
mean        0.049385
std         2.414257
min       -26.390093
25%        -0.971510
50%         0.100083
75%         1.270603
max        17.593598
Name: ret_pct, dtype: float64
```

In [33]:
```
df3 = download_financial_data("WE")
Calculate_Statistics(df3,"WE")
```

```
WE.csv created
68% of the daily returns for WE will be between -3.8994% and 3.9056%
```

Out[33]:
```
count    404.000000
mean       0.003108
std        3.907349
min      -22.312704
25%       -1.288059
50%        0.000000
75%        1.128745
max       20.287622
Name: ret_pct, dtype: float64
```