

Download the Google stock data

```
In [30]: # -*- coding: utf-8 -*-
"""
Created on Tue Sep 7 16:02:02 2021

@author: epinsky
"""

# -*- coding: utf-8 -*-
"""
Created on Mon Nov 5 14:37:29 2018

@author: epinsky
"""

# install yfinance version 0.1.62
# !pip install yfinance==0.1.62
# run this !pip install pandas_datareader
from pandas_datareader import data as web
import os
import pandas as pd
import yfinance as yf

def get_stock(ticker, start_date, end_date, s_window, l_window):
    try:
        yf.pdr_override()
        df = yf.download(ticker, start=start_date, end=end_date)
        # can use this as well df = web.get_data_yahoo(ticker, start=start_date, end=end_date)
        df['Return'] = df['Adj Close'].pct_change()
        df['Return'].fillna(0, inplace=True)
        df['Date'] = df.index
        df['Date'] = pd.to_datetime(df['Date'])
        df['Month'] = df['Date'].dt.month
        df['Year'] = df['Date'].dt.year
        df['Day'] = df['Date'].dt.day
        for col in ['Open', 'High', 'Low', 'Close', 'Adj Close']:
            df[col] = df[col].round(2)
        df['Weekday'] = df['Date'].dt.day_name()
        df['Week Number'] = df['Date'].dt.strftime('%U')
        df['Year Week'] = df['Date'].dt.strftime('%Y-%U')
        df['Short MA'] = df['Adj Close'].rolling(window=s_window, min_periods=1).mean()
        df['Long MA'] = df['Adj Close'].rolling(window=l_window, min_periods=1).mean()
        col_list = ['Date', 'Year', 'Month', 'Day', 'Weekday',
                    'Week Number', 'Year Week', 'Open',
                    'High', 'Low', 'Close', 'Volume', 'Adj Close',
                    'Return', 'Short MA', 'Long MA']
        num_lines = len(df)
        df = df[col_list]
        print('read ', num_lines, ' lines of data for ticker: ', ticker)
        return df
    except Exception as error:
        print(error)
        return None

try:
    ticker="GOOGL"
    input_dir = os.getcwd()
    output_file = os.path.join(input_dir, ticker + '.csv')
    df = get_stock(ticker, start_date='2016-01-01', end_date='2020-12-31',
                  s_window=14, l_window=50)
    df.to_csv(output_file, index=False)
    print('wrote ' + str(len(df)) + ' lines to file: ' + output_file)
except Exception as e:
    print(e)
    print('failed to get Yahoo stock data for ticker: ', ticker)
```

```
[*****100*****] 1 of 1 completed
read 1258 lines of data for ticker: GOOGL
wrote 1258 lines to file: C:\Users\Lee\iCloudDrive\Document\Boston University\CS677 DS with Python\Homework\Assignment_1\Part_2_portfolio_analysis\My Code\GOOGL.csv
```

Open "SPY.csv" file

```
In [31]: # -*- coding: utf-8 -*-
"""
Created on Mon Nov 5 14:37:29 2018

@author: epinsky
this scripts reads your ticker file (e.g. MSFT.csv) and
constructs a list of lines
"""
import os

ticker='SPY'
input_dir = os.getcwd()
ticker_file = os.path.join(input_dir, ticker + '.csv')

try:
    with open(ticker_file) as f:
        lines = f.read().splitlines()
        print('opened file for ticker: ', ticker)
        """
        your code for assignment 1 goes here
        """

except Exception as e:
    print(e)
    print('failed to read stock data for ticker: ', ticker)
```

opened file for ticker: SPY

```
In [32]: # Show the first 5 rows of the table
spyTable = lines
for i in range(5):
    print(spyTable[i])
```

```
Date,Year,Month,Day,Weekday,Week Number,Year Week,Open,High,Low,Close,Volume,Adj Close,Return,Short_MA,Long_MA
2016-01-04,2016,1,4,Monday,01,2016-01,200.49,201.03,198.59,201.02,222353500,181.32,0.0,181.32,181.32
2016-01-05,2016,1,5,Tuesday,01,2016-01,201.4,201.9,200.05,201.36,110845800,181.63,0.0016912824647046243,181.47
5,181.475
2016-01-06,2016,1,6,Wednesday,01,2016-01,198.34,200.06,197.6,198.82,152112600,179.34,-0.01261422378233823,180.7
6333333333332,180.76333333333332
2016-01-07,2016,1,7,Thursday,01,2016-01,195.33,197.44,193.59,194.05,213436100,175.04,-0.023991350126892685,179.
33249999999998,179.33249999999998
```

```
In [33]: # How many data do we have
spyDataLen = len(spyTable) - 1
print('we have {} data'.format(spyDataLen))
```

we have 1258 data

```
In [34]: # Type structure of the table
print("The spyTable is a {} of {}".format(type(spyTable), type(spyTable[0])))
```

The spyTable is a <class 'list'> of <class 'str'>

```
In [35]: # Split the row 1 data, which are labels
# Turn a string into a list of string
lables = spyTable[0]
lables = lables.split(",")
print(lables)

# Know the index of label 'Return'
# Know the index of label 'year'
returnIndex = lables.index("Return")
yearIndex = lables.index('Year')
print("\nThe index of label 'Return' is {}".format(returnIndex))
print("\nThe index of label 'Year' is {}".format(yearIndex))
```

```
['Date', 'Year', 'Month', 'Day', 'Weekday', 'Week Number', 'Year Week', 'Open', 'High', 'Low', 'Close', 'Volum
e', 'Adj Close', 'Return', 'Short_MA', 'Long_MA']
```

The index of label 'Return' is 13

The index of label 'Year' is 1

```
In [36]: # Collect all 'Return' data from the spyTable
spyReturnList = []
for line in spyTable[1:]:
    line = line.split(',')
    returnn = line[returnIndex]
    spyReturnList.append(float(returnn))

# Check the amount of the return data is correct
# Double check the second day data is the same with the csv file
print(len(spyReturnList))
print(spyReturnList[1])
```

1258
0.0016912824647046243

Open "GOOGL.csv" file

```
In [37]: # -*- coding: utf-8 -*-
"""
Created on Mon Nov 5 14:37:29 2018

@author: epinsky
this scripts reads your ticker file (e.g. MSFT.csv) and
constructs a list of lines
"""
import os

ticker='GOOGL'
input_dir = os.getcwd()
ticker_file = os.path.join(input_dir, ticker + '.csv')

try:
    with open(ticker_file) as f:
        lines = f.read().splitlines()
        print('opened file for ticker: ', ticker)
        """
        your code for assignment 1 goes here
        """

except Exception as e:
    print(e)
    print('failed to read stock data for ticker: ', ticker)
```

opened file for ticker: GOOGL

```
In [38]: # Show the first 5 rows of the table
gooTable = lines
for i in range(5):
    print(gooTable[i])
```

```
Date,Year,Month,Day,Weekday,Week Number,Year Week,Open,High,Low,Close,Volume,Adj Close,Return,Short_MA,Long_MA
2016-01-04,2016,1,4,Monday,01,2016-01,762.2,762.2,747.54,759.44,3369100,759.44,0.0,759.44,759.44
2016-01-05,2016,1,5,Tuesday,01,2016-01,764.1,769.2,755.65,761.53,2260800,761.53,0.002752063163317553,760.485,76
0.485
2016-01-06,2016,1,6,Wednesday,01,2016-01,750.37,765.73,748.0,759.33,2410300,759.33,-0.002888936906483508,760.1,
760.1
2016-01-07,2016,1,7,Thursday,01,2016-01,746.49,755.31,735.28,741.0,3156600,741.0,-0.0241397406898135,755.325,7
55.325
```

```
In [39]: # how many data do we have
gooDataLen = len(gooTable) - 1
print('we have also {} data, match with "SPY" data'.format(gooDataLen))
```

we have also 1258 data, match with "SPY" data

```
In [40]: # Collect all 'Return' data from the spyTable
gooReturnList = []
for line in gooTable[1:]:
    line = line.split(',')
    returnn = line[returnIndex]
    gooReturnList.append(float(returnn))

# Check the amount of the return data is correct
# Double check the second day data is the same with the csv file
print(len(gooReturnList))
print(gooReturnList[1])
```

1258
0.002752063163317553

Portfolio Analysis

Now we have extracted both 'Return' data from both csv file Then, I am going to build the portfolio

```
In [41]: # Locate the index of each year
# First, extract the 'Year' data from raw data, and turn them into a list
yearList = []
yearIndexList = []
for line in spyTable[1:]:
    line = line.split(',')
    year = line[yearIndex]
    yearList.append(int(year))

# Determine the index of each year in the list
for i in range(5):
    yearIndexList.append(yearList.index(2016 + i))
yearIndexList.append(len(spyReturnList))

# Show the list
print(yearIndexList)
```

[0, 252, 503, 754, 1006, 1258]

Define some useful function for convience

```
In [42]: class node:
    def __init__(self, final, mx, mn):
        self._final = final
        self._max = mx
        self._min = mn

    def finMaxMin(yearlyFund):
        fin = yearlyFund[-1]
        mx = max(yearlyFund)
        mn = min(yearlyFund)

        return node(fin, mx, mn)

    def comReturnFun(alpha, beta, security1, security2):
        comReturn = []
        for i in range(len(spyReturnList)):
            comReturn.append(alpha * security1[i] + beta * security2[i])
        return comReturn

    def fundEDayFun(initialFund, comReturn, yearIndexList):
        fund = [0]
        for i in range(5):
            temp = initialFund
            for r in comReturn[yearIndexList[i] : yearIndexList[i + 1]]:
                temp = temp * (1 + r)
            fund[i].append(temp)
        return fund
```

```
In [43]: # Parameter from the question
initialFund = 100
alphaList = [0,0.2,0.4,0.6,0.8,1]
```

```
In [44]: # Process the data
total = []
for alpha in alphaList:
    beta = 1 - alpha
    comReturn = comReturnFun(alpha, beta, spyReturnList, gooReturnList)
    fundFiveYear = fundEDayFun(initialFund, comReturn, yearIndexList)
    result = []
    for i in range(5):
        temp = finMaxMin(fundFiveYear[i])
        result.append(temp)
    total.append(result)
```

```
In [45]: # Print the data in the way we want
for i in range(5):
    print("{}".format(2016+i))
    for j in range(len(alphaList)):
        temp = total[j][i]
        print('Alpha= {:<3f> Final= {:6.2f} Max= {:6.2f} Min= {:6.2f}'.format(alphaList[j], temp.final, temp
        .max, temp.min))
    print()
```

```
2016
Alpha= 0 Final= 104.35 Max= 110.05 Min= 89.69
Alpha= 0.2 Final= 106.34 Max= 109.92 Min= 91.84
Alpha= 0.4 Final= 108.27 Max= 110.70 Min= 92.27
Alpha= 0.6 Final= 110.12 Max= 112.24 Min= 91.86
Alpha= 0.8 Final= 111.89 Max= 113.70 Min= 91.42
Alpha= 1 Final= 113.59 Max= 115.06 Min= 90.97
```

```
2017
Alpha= 0 Final= 138.71 Max= 142.88 Min= 106.36
Alpha= 0.2 Final= 139.07 Max= 142.55 Min= 108.18
Alpha= 0.4 Final= 139.21 Max= 141.99 Min= 109.87
Alpha= 0.6 Final= 139.11 Max= 141.19 Min= 111.49
Alpha= 0.8 Final= 138.79 Max= 140.17 Min= 113.02
Alpha= 1 Final= 138.24 Max= 138.93 Min= 114.45
```

```
2018
Alpha= 0 Final= 137.60 Max= 169.27 Min= 129.66
Alpha= 0.2 Final= 137.22 Max= 165.54 Min= 129.17
Alpha= 0.4 Final= 136.46 Max= 161.81 Min= 128.32
Alpha= 0.6 Final= 135.31 Max= 158.94 Min= 127.12
Alpha= 0.8 Final= 133.80 Max= 155.76 Min= 125.58
Alpha= 1 Final= 131.92 Max= 153.38 Min= 123.70
```

```
2019
Alpha= 0 Final= 176.37 Max= 179.40 Min= 135.03
Alpha= 0.2 Final= 177.16 Max= 179.72 Min= 134.55
Alpha= 0.4 Final= 177.23 Max= 179.30 Min= 133.68
Alpha= 0.6 Final= 176.57 Max= 178.14 Min= 132.45
Alpha= 0.8 Final= 175.19 Max= 176.27 Min= 130.85
Alpha= 1 Final= 173.12 Max= 173.70 Min= 128.91
```

```
2020
Alpha= 0 Final= 228.62 Max= 240.30 Min= 138.80
Alpha= 0.2 Final= 225.99 Max= 234.57 Min= 136.16
Alpha= 0.4 Final= 222.12 Max= 227.82 Min= 132.97
Alpha= 0.6 Final= 217.08 Max= 220.03 Min= 129.29
Alpha= 0.8 Final= 210.94 Max= 211.93 Min= 125.16
Alpha= 1 Final= 203.81 Max= 203.91 Min= 120.62
```