

Stock portfolio and market index ex post performance comparison

December 21, 2018

University of St. Gallen

Skills: Programming with Advanced Computer Languages

Authors:

Doga Alkan:
Loris Roggo:
Jan-Marius Hüweler:

Student ID:

15 – 611 – 098
16 – 616 – 914
15 – 616 – 154

Contents

1	Task and programming procedure	1
2	Code	1
3	Example: outline	6
4	Example: input and output	7

1 Task and programming procedure

This program lets the user choose a portfolio consisting of stocks from Yahoo Finance to evaluate the hypothetical performance over an ex post time period of choice and compares it to the performance of the S&P500 market index. The S&P 500 is a stock market index that tracks the stocks of 500 large-cap U.S. companies. It represents the stock market's performance by reporting the risks and returns of the biggest companies. Investors use it as the benchmark of the overall market, to which all other investments are compared. The S&P 500 tracks the market capitalization of the companies in its index. Market cap is the total value of all shares of stock a company has issued. It's calculated by multiplying the number of shares issued by the stock price. A company that has a market cap of 100 billion USD receives 10 times the representation as a company whose market cap is 10 billion USD. It captures 80 percent of the market cap of the stock market.

The user can select, which stocks in which quantity he wants to include in his own stock portfolio. The code is structured by getting access to the stock data from Yahoo Finance and to construct a code to build the stock portfolio and calculate its cumulative returns (as well as for the S&P 500). The code is corrected for different potential input mistakes and returns error messages in those cases so the user can revise the input again. Finally the portfolio cumulative returns are plotted with the cumulative returns of the market index over a desired time period to allow for better visualization.

2 Code

```
In [2]: #In this program, we use yahoo finance. To run yahoo finance,  
        #you need to install the tool in terminal/command.  
        #Install fix_yahoo_finance using pip:  
        # $ pip install fix_yahoo_finance --upgrade --no-cache-dir  
  
import pandas as pd  
from pandas_datareader import data as pdr  
import fix_yahoo_finance as yf  
yf.pdr_override()  
import math  
import pandas as pd  
import numpy as np  
import matplotlib.pyplot as plt  
%matplotlib inline
```

```

#first, we define the functions needed

#function to calculate the cumulative daily returns of a stock or index
def cum_daily_returns(a):
    adj_daily_close= a['Adj Close']
    adj_daily_returns = adj_daily_close / adj_daily_close.shift(1) -1
    cum_adj_daily_returns = (1+adj_daily_returns).cumprod()
    #return a dataframe
    cum_adj_daily_returns = pd.DataFrame(cum_adj_daily_returns)
    return cum_adj_daily_returns

#loop to multiply the stock price with the quantity entered
def add_quantity(ticker_company):
    portfolio_value = pd.Series()
    #change quantity variable for each loop
    y = 0
    #loop to multiply the stock price with the right quantity
    for i in ticker_company:
        #get the financial data for the portfolio
        port_data = pdr.get_data_yahoo(i, start_date, end_date)
        e = pd.DataFrame(port_data)
        #get the number in the serie named quantity
        quant_num = int(diff_quant.iloc[y])
        #multiply the adj close from everyday with the quantity
        e = e['Adj Close'].mul(quant_num)
        #fill the Nan values with 0 to be able to add the series
        portfolio_value = portfolio_value.add(e, fill_value = 0)
        #increase y, to get the next row in the serie with the quantities
        y = y + 1
    #create a dataframe with the portfolio value
    portfolio_value = pd.DataFrame(portfolio_value, columns = ['Adj Close'])
    return portfolio_value

#function to get the highest return stock
def highest_return(stocks):
    port_data = pdr.get_data_yahoo(stocks, start_date, end_date)
    port_data = pd.DataFrame(port_data)
    #drop companies that were not listed on the stock market at the start date
    port_data = port_data.dropna(axis = 1)
    port_data = (port_data['Adj Close'].iloc[-1] /
                 port_data['Adj Close'].iloc[0] -1) * 100
    #get the highest number in the serie
    high_return_comp = port_data.idxmax()
    per_return = port_data.get(high_return_comp)
    return "The stock with the highest return during this "\
    "period is %s with %.2f%%" % (high_return_comp,per_return)

def lowest_return(stocks): #to get the lowest return stock

```

```

port_data = pdr.get_data_yahoo(stocks, start_date, end_date)
port_data = pd.DataFrame(port_data)
#drop companies that were not listed on the stock market at the start date
port_data = port_data.dropna(axis = 1)
port_data = (port_data['Adj Close'].iloc[-1] /
              port_data['Adj Close'].iloc[0] - 1) * 100
high_return_comp = port_data.idxmin()
per_return = port_data.get(high_return_comp)
return "The stock with the lowest return during this "\
"period is %s with %.2f%%" % (high_return_comp, per_return)

#step 1: onboarding
print ("This program lets you choose a portfolio consisting of stocks\
from Yahoo Finance to evaluate the hypothetical performance over \
an ex post time period of choice and compares it to the performance \
of the S&P500 market index. You can select, which stocks in which \
quantity you want to include in your portfolio.")

#step 2: insert transactions into program
#we create a dataframe with ticker and quantity as column names
col_names = ['ticker', 'quantity']
portfolio = pd.DataFrame(columns = col_names)

#after you finish the program, you can start to build another portfolio
#using a while loop
run_again = ("yes")
while run_again == ("yes"):

    question2 = ("no")

    while question2 == ("no"):
        print ("First you need to enter your portfolio.")

        portfolio = portfolio [0:0]
        question = ("yes")

        #try if the ticker entered exist in yahoo finance and try if
        #you entered a number for the quantity
        while True:
            portfolio = portfolio.append(pd.Series(
                [input("What's the ticker of the first stock? "),
                 input("How many of those stocks do you have in your portfolio? ")] ,
                index=portfolio.columns ), ignore_index=True)
            a = portfolio['ticker'].tolist()
            b = portfolio['quantity'].tolist()
            a = a[-1]

```

```

b = b[-1]
try:
    w = pdr.get_data_yahoo(a, start = "1950-01-01",
                           end = "2018-01-01")

    b = int(b)
    break
except ValueError:
    portfolio = portfolio.drop(portfolio.index[-1])
    print("Error! The ticker you entered does "\
          "not exist or you did not enter a number "\
          "for the quantity; please try again.")
    continue

while question != ("no"):
    question = input("Do you want to add another stock? (yes/no)")

    if question == "yes":
        #try if the ticker entered exist in yahoo finance and try if
        #you entered a number for the quantity
        while True:
            portfolio = portfolio.append(pd.Series(
                [input("What's the ticker of this stock? "),
                 input("How many of those stocks do you have " \
                       "in your portfolio? ")],
                index=portfolio.columns ), ignore_index=True)
            a = portfolio['ticker'].tolist()
            b = portfolio['quantity'].tolist()
            a = a[-1]
            b = b[-1]
            try:
                w = pdr.get_data_yahoo(a, start = "1950-01-01",
                                       end = "2018-01-01")

                b = int(b)
                break
            except ValueError:
                portfolio = portfolio.drop(portfolio.index[-1])
                print("Error! The ticker you entered does not exist "\
                      "or you did not enter a number for the quantity; "\
                      "please try again.")
                continue

        elif question == "no":
            print ("Alright, please check if the following table " \
                  "of your transactions is correct.")
            print (portfolio)

            question2 = input ("Is this portfolio table correct? (yes/no) ")

```

```

        if question2 == ("yes"):
            print ("Perfect, let's go on!")

        elif question2 == ("no"):
            print ("Alright, we'll have to start over!")

    else:
        print ("You didn't type yes or no, please revise.")

#step 3: insert performance time period dates into program
print ("Now define the time period you want to " \
"calculate your portfolio performance for.")
start_date = input("What's the start date? (YYYY-MM-DD) ")
end_date = input("What's the end date? (YYYY-MM-DD) ")

#step 4: program provides information about users portfolio

#create a list with the different tickers of the companies
num_stocks = portfolio['ticker'].tolist()
#create a list out of column quantity
diff_quant= portfolio['quantity'].tolist()
#create a serie based on the quantities
diff_quant = pd.Series(diff_quant)
#rename the column
diff_quant.columns = ['quantity']

#portfolio is based at 100 to be compared with the index
#extract the financial data of S&P500
sp500 = pdr.get_data_yahoo('^GSPC',start_date,end_date)
sp500 = sp500['Adj Close']
sp500 = pd.DataFrame(sp500)

#the cumulative daily returns of portfolio are calculated
#and compared to the return of the index
z = add_quantity(num_stocks)
#call the function to calculate cumulative daily returns
cum_index = cum_daily_returns(sp500)
cum_port = cum_daily_returns(z)

#plot the graph and change the legend, labs and title
plt.figure(figsize =(13.5,9))
plt.plot(cum_index, color = 'orange')
plt.plot(cum_port, color = 'blue')
plt.legend(['S&P500','Portfolio'], fontsize = 15)
plt.xlabel('Time Period', fontsize = 15)
plt.ylabel('Cumulative returns', fontsize = 15)
plt.title("Ex post stock portfolio/market index comparison",
          fontsize = 20)

```

```

plt.show()

#if user enters only one stock then there will be no distinction between the
#highest and lowest returns; if user entered multiple stocks then the code
#computes the highest and lowest return of the time period
if len(num_stocks) == 1:
    port_data = pdr.get_data_yahoo(num_stocks, start_date, end_date)
    port_data = pd.DataFrame(port_data)
    port_data = (port_data['Adj Close'].iloc[-1] /
                 port_data['Adj Close'].iloc[0] - 1) * 100
    #convert stocks into a string to avoid having brackets in the result
    stock = ''.join(num_stocks)
    print("You only entered one stock. During this "\
          "period %s had %.2f%% return" % (stock, port_data))
else:
    print(highest_return(num_stocks))
    print(lowest_return(num_stocks))

#compute the returns for the S&P 500 over the time period
sp500_returns = (sp500['Adj Close'].iloc[-1] /
                 sp500['Adj Close'].iloc[0] - 1) * 100
print('During this period, S&P 500 had %.2f%% return' % sp500_returns)

#compute the returns for stock portfolio over the time period
portfolio_returns = (z['Adj Close'].iloc[-1] /
                    z['Adj Close'].iloc[0] - 1) * 100
print('During this period, your portfolio had %.2f%% return' % portfolio_returns)

#question if the user wants to repeat the whole construction process again
run_again = input("Would you like to start over? (yes/no)")

if run_again == ("yes"):
    print ("Alright, let's start over!")

elif run_again == ("no"):
    print ("Hope you enjoyed our program. Goodbye.")

```

3 Example: outline

The following code output displays the example output of the written code for a stock portfolio consisting of 21 Apple stocks, 12 IBM stocks, 23 Tesla stocks and 4 Google stocks over a time period from January 2016 to January 2018. This portfolio is then compared with the S&P 500 market index in a plot and the according returns for the portfolio and the index are computed. In the computed example the stock with the highest return in the portfolio was Apple with a return of 67.99% and IBM the stock with the lowest return of 21.37%. This can be compared to the S&P 500 index with a return of 30.81% over the examined period. In contrast the chosen stock portfolio would have significantly performed better over the time period with a return of 37.23%.

4 Example: input and output

What's the ticker of the first stock? AAPL
How many of those stocks do you have in your portfolio? 21

[*****100%*****] 1 of 1 downloaded

Do you want to add another stock? (yes/no) yes
What's the ticker of this stock? IBM
How many of those stocks do you have in your portfolio? 12

[*****100%*****] 1 of 1 downloaded

Do you want to add another stock? (yes/no) yes
What's the ticker of this stock? TSLA
How many of those stocks do you have in your portfolio? 23

[*****100%*****] 1 of 1 downloaded

Do you want to add another stock? (yes/no) yes
What's the ticker of this stock? GOOG
How many of those stocks do you have in your portfolio? 4

[*****100%*****] 1 of 1 downloaded

Do you want to add another stock? (yes/no) no

Alright, please check if the following table of your transactions is correct.

	ticker	quantity
0	AAPL	21
1	IBM	12
2	TSLA	23
3	GOOG	4

Is this portfolio table correct? (yes/no) yes

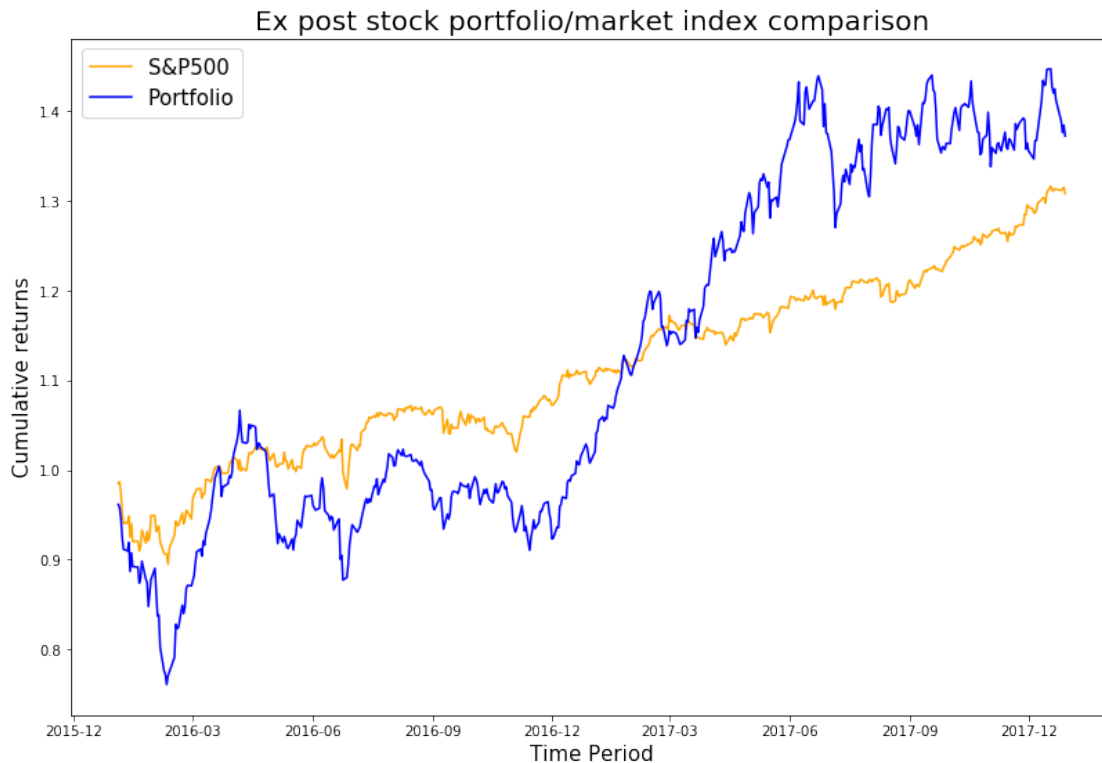
Perfect, let's go on!

Now define the time period you want to calculate your portfolio performance for.

What's the start date? (YYYY-MM-DD) 2016-01-01

What's the end date? (YYYY-MM-DD) 2018-01-01

```
[*****100%*****] 1 of 1 downloaded
[*****100%*****] 1 of 1 downloaded
[*****100%*****] 1 of 1 downloaded
[*****100%*****] 1 of 1 downloaded
[*****100%*****] 1 of 1 downloaded
```



```
[*****100%*****] 4 of 4 downloaded
The stock with the highest return during this period is AAPL with 66.99%
[*****100%*****] 4 of 4 downloaded
The stock with the lowest return during this period is IBM with 21.37%
During this period, S&P 500 had 30.81% return
During this period, your portfolio had 37.23% return
```

Would you like to start over? (yes/no) no

Hope you enjoyed our program. Goodbye.