Quantitative Portfolio Management

Assignment #4

Raman Uppal
EDHEC Business School

November 2023

Instructions for each assignment . . . I

- Assignment #1 should be done individually.
- The other assignments are to be done in groups of 4 or 5 students.
  - This means that groups of 1, 2, 3, 6, etc. are not allowed.
  - Diversity in groups is strongly encouraged
    (people from different countries, different genders, different finance knowledge, and different coding ability, etc.)

Instructions for each assignment . . . II

- Each assignment should be emailed as a Jupyter file
  - To `Raman.Uppal@edhec.edu`
  - The subject line of the email should be: "QPM: Assignment n," where $n = \{1, 2, \ldots, 8\}$.
  - Assignment n is due before Lecture n, where $n = \{1, 2, \ldots, 8\}$.
  - Assignments submitted late will not be accepted (grade $= 0$), so please do not email me assignments after the deadline.

Instructions for each assignment . . . III

- The Jupyter file should include the following (use Markdown):
  - Section "0" with information about your submission:
    - Line 1: QPM: Assignment n
    - Line 2: Group members: listed alphabetically by last name, where the last name is written in CAPITAL letters
    - Line 3: Any comments/challenges about the assignment
  - Section "k" where $k = \{1, 2, \ldots\}$.
    - First type Question k of Assignment n.
    - Then, below the question, provide your answer.
    - Your code should include any packages that need to be imported.

Initial step to prepare the data for this assignment

- The data we will be using is the same that we used for the previous assignment. For convenience, I have typed again the instructions.
  - Make sure you have already imported "pandas" and "yfinance."
  - Download from Wikipedia (or any other source) a table that lists the companies that comprise the S&P 500. (See "Helpful links" provided at the end of the assignment.)
  - From this table, extract the list of ticker symbols.
  - Set the start date and end date to be
    - start_date = "2000-01-01"
    - end_date = "2022-12-31"
  - Build a dataframe that contains the stock prices for the S&P 500 companies. (If there are errors for some company names, it is fine to ignore the company names with errors.)
  - Drop the columns that have only "NaN" entries.
  - Drop also the companies with more than 100 missing observations.

Questions for Assignment 4 . . . I

Q4.0 From the data that we used for the previous assignment, select the following 10 companies (these are the first 10 companies with no missing data):
"MMM","AOS","ABT","ADM","ADBE","ADP","AES","AFL","A","AKAM"

- So, our "new" dataset for this assignment will consist of monthly returns you had computed in the last assignment, but just for these 10 companies.
- To reduce the work required for this assignment, please assume that the risk-free rate of return is zero.

Questions for Assignment 4 ...II

Q4.1 Choose the estimation window to be $T^{\text{est}} = 60$ months of monthly returns. Call this the estimation sample. Use the estimation sample to compute the following two portfolio strategies:

a. mean-variance portfolio (MVP) without constraints on the size of the weight (assume that a risk-free rate is available, with the risk-free rate equal to zero);

b. global minimum variance (GMV) portfolio without constraints on the size of the weight.

Questions for Assignment 4 . . . III

Q4.2 Now use a rolling window of $T^{\text{est}} = 60$ months to estimate the portfolio weights for the two strategies listed above for each of the $T - T^{\text{est}}$ months. That is, repeat the calculations of the previous question for all the dates *after* the first 60 months.

Q4.3 Use the time-series of portfolios weights for each of the two portfolio strategies, to compute the out-of-sample portfolio returns. That is, for each of the two portfolio strategies that you estimate at each date t, compute its out-of-sample return in month $t + 1$.

Q4.4 Now, compute the Sharpe ratio of the out-of-sample returns for the two portfolio strategies. Which strategy has the higher Sharpe ratio?

Helpful hints

- Helpful links for information on downloading S&P 500 ticker symbols.
  - from Danny Groves
  - from GitHub
- Finally, please save the data you have downloaded and created for these ten companies because we will be using it again.

Discussion of Assignment 4: Initial setup

- We start by loading the libraries we will need.

Code to load required libraries

```
# Import libraries
import pandas as pd
import yfinance as yf
import numpy as np
import pandas_datareader as pdr
```

Discussion of Assignment: Q4.0

Q4.0 From the data that we used for the previous assignment, select the following 10 companies (these are the first 10 companies with no missing data):
"MMM","AOS","ABT","ADM","ADBE","ADP","AES","AFL","A","AKAM"

- So, our "new" dataset for this assignment will consist of monthly returns you had computed in the last assignment, but just for these 10 companies.
- To reduce the work required for this assignment, please assume that the risk-free rate of return is zero.

Code for Q4.0

Code to download data for required tickers

```python
# List of tickets for which we will download the data
tickers=["MMM","AOS","ABT","ADM","ADBE","ADP","AES","AFL","A","AKAM"]

# Set the start and end dates
start_date = "2000-01-01"
end_date = "2022-12-31"

# Create an empty dataframe
stock_prices = pd.DataFrame()

# Download the data
for ticker in tickers:
    price = yf.download(ticker,start=start_date,end=end_date)
    stock_prices[ticker] = price["Adj Close"]

# Change the index column to be the date
stock_prices.index = pd.to_datetime(stock_prices.index)
```

Code for Q4.0 (continued)

Construct returns from downloaded data

```
# Extract the stock price at the end of each month from downloaded data
month_stock_prices = stock_prices.resample("1M").last()
month_stock_prices.index = month_stock_prices.index.date

# Compute returns
month_return=np.log(month_stock_prices/month_stock_prices.shift(1))
month_return=month_return.dropna() # delete the first row - without data
```

Discussion of Assignment: Q4.1

Q4.1 Choose the estimation window to be $T^{\text{est}} = 60$ months of monthly returns. Call this the estimation sample. Use the estimation sample to compute the following two portfolio strategies:

a. mean-variance portfolio (MVP) without constraints on the size of the weight (assume that a risk-free rate is available, with the risk-free rate equal to zero);

b. global minimum variance (GMV) portfolio without constraints on the size of the weight.

Code for Q4.1

Code for unconstrained portfolios: MVP and GMV

```
# Optimal weights of MVP (tangency) portfolio without constraints:
def MVP_w(returns):
    ones = np.ones(len(returns.columns)) # 10 companies in this case
    mu = returns.mean()
    var_cov = returns.cov()
    numerator = np.linalg.inv(var_cov) @ mu
    denominator = ones.T @ np.linalg.inv(var_cov) @ mu
    optimal_weight = numerator/denominator
    return optimal_weight

# Optimal weights of GMV portfolio without constraints:
def GMV_w(returns):
    ones = np.ones(len(returns.columns)) #10 companies in this case
    var_cov = returns.cov()
    numerator = np.linalg.inv(var_cov) @ ones
    denominator = ones.T @ np.linalg.inv(var_cov) @ ones
    optimal_weight = numerator/denominator
    return optimal_weight
```

Discussion of Assignment: Q4.2

Q4.2 Now use a rolling window of $T^{\text{est}} = 60$ months to estimate the portfolio weights for the two strategies listed above for each of the $T - T^{\text{est}}$ months. That is, repeat the calculations of the previous question for all the dates *after* the first 60 months.

Code for Q4.2: Mean-variance portfolio

Code for rolling-window analysis of MVP

```
# MVP portfolio:
MVP_weights = pd.DataFrame(index = month_return.index, columns =
     month_return.columns)
for i in month_return.index[60:]: # start from the 61st month
    start_date = i - pd.DateOffset(months = 60)
    end_date = i-pd.DateOffset(months = 1)
    start_date = pd.to_datetime(start_date).date()
    end_date = pd.to_datetime(end_date).date()
    df = month_return.loc[start_date:end_date]
    MVP_weights.loc[i] = MVP_w(df) # calculate MVP for each rolling
     window

MVP_weights = MVP_weights.dropna()
```

Code for Q4.2: Global minimum-variance portfolio

Code for rolling-window analysis of GMV

```
# GMV portfolio:
GMV_weights = pd.DataFrame(index = month_return.index, columns =
     month_return.columns)
for i in month_return.index[60:]:#start from the 61th month
    start_date = i - pd.DateOffset(months = 60)
    end_date = i - pd.DateOffset(months = 1)
    start_date = pd.to_datetime(start_date).date()
    end_date = pd.to_datetime(end_date).date()
    df = month_return.loc[start_date:end_date]
    GMV_weights.loc[i] = GMV_w(df) # calculate GMV for each rolling
     window

GMV_weights=GMV_weights.dropna()
```

Discussion of Assignment: Q4.3

Q4.3 Use the time-series of portfolios weights for each of the two portfolio strategies, to compute the out-of-sample portfolio returns. That is, for each of the two portfolio strategies that you estimate at each date t, compute its out-of-sample return in month $t+1$.

Code for computing portfolio return at date $t+1$

```
# Because Rf is assumed to be 0, the portfolio return equals the return
    on the portfolio of risky assets

# Return on MVP portfolio without constraints:
MVP_return = (MVP_weights * month_return.iloc[60:]).sum(axis = 1)

# Return on GMV portfolio without constraints:
GMV_return = (GMV_weights * month_return.iloc[60:]).sum(axis = 1)
```

Discussion of Assignment: Q4.4

Q4.4 Now, compute the Sharpe ratio of the out-of-sample returns for the two portfolio strategies. Which strategy has the higher Sharpe ratio?

Code to compute the Sharpe ratio

```
def sharpe_ratio(m_return):                      # intput is monthly return
    Rf=0

    m_mean_return = m_return.mean()      # monthly return mean
    m_vol = m_return.std()               # monthly return volatility

    y_mean_return = m_mean_return * 12   # transform to yearly mean
    y_vol = m_vol * np.sqrt(12)          # transform to yearly volatility

    SR = (y_mean_return-Rf)/y_vol

    return SR
```

Code and final output for Q4.4 (continued)

Code for printing the output

```
# Sharpe ratio for MVP portfolio:
print(f'Annualized Sharpe ratio of MVP portfolio is
 {sharpe_ratio(MVP_return)}.')

# Sharpe ratio for GMV portfolio:
print(f'Annualized Sharpe Ratio of GMV portfolio is
 {sharpe_ratio(GMV_return)}.')
```

- The final output we get from the above print statement is:

```
Annualized Sharpe ratio of MVP portfolio is -0.14067884090356617.
Annualized Sharpe Ratio of GMV portfolio is 0.691813497486071.
```

- From the above result, we conclude that, in the absence of constraints, the GMV portfolio outperforms the mean-variance portfolio – at least for the companies and sample period considered.

End of assignment