

# MIE 1622H: Final Exam Project

Dr. Oleksandr Romanko

April 6, 2021

---

**Due:** April 15, 2021, not later than 6:00 pm.

Use **Python** for MIE 1622H Final Exam Project.

**You should hand in:**

- Your report (up to 3-page, excluding graphs) and your Python code.
  - Compress your report, all of your Python code files and output files into a file **StudentID.zip** (which can be uncompressed by 7-zip (<http://www.7-zip.org>) software under Windows), and submit it via Quercus portal no later than 6:00 pm on April 15.
- 

## Question 1

File `Daily_closing_prices.csv` contains prices of 20 stocks for years 2019-2020. Your portfolio consists of 100 MSFT, 200 AAPL and 500 IBM shares and you hold it from the beginning of January 2019 till the end of December 2020.

To compute VaR of the portfolio of stocks we need to compute daily losses for a portfolio, i.e., add daily monetary loss for each stock and use that sum in the calculations of VaR.

To calculate 10-day VaR we need to use moving window method. To compute 10 day loss we compute loss for the first 10 trading days, then loss for second 10 trading days, etc. For example, for the given data, we compute portfolio loss from January 2, 2019 to January 16, 2019 as value of the portfolio on January 2 minus value of the portfolio on January 16. Then we compute loss of the portfolio for January 3, 2019 - January 17, 2019 in the same way as described above, then loss for January 4, 2019 - January 18, 2019, etc.

### Part 1

Compute VaR and CVaR for your portfolio value at quantile level 95% for 1-day and 10-day time horizons:

- From historical scenarios for years 2019-2020;
- If we assume Normal distribution of losses (mean and standard deviation are computed from the historical scenarios for years 2019-2020 for each stock). For the Normal distribution model, compute means and covariances for each stock from historical scenarios and then compute portfolio mean and variance as we have done for Markowitz model.

Plot a histogram of the distribution of losses in portfolio value, indicate VaR and CVaR levels for two time horizons. In the report include two figures (for 1-day and 10-day results) and answer the question below.

Is the following true for the given dataset? Please explain your answer:

- $\text{VaR}(10 \text{ day}) = 10 * \text{VaR}(\text{one day})$
- $\text{CVaR}(10 \text{ day}) = 10 * \text{CVaR}(\text{one day})$

## Part 2

Compute 1-day 95% VaR for 100 MSFT, 200 AAPL and 500 IBM stocks separately using historical scenarios and print your results.

Is the following true? Explain:  $\text{VaR}(\text{Portfolio}) = \text{VaR}(\text{MSFT}) + \text{VaR}(\text{AAPL}) + \text{VaR}(\text{IBM})$

Would the answer change if we used Normal distribution model to calculate VaR?

## Question 2

You are given historical data for the time period from 2 January 2019 till 31 December 2020 in file `Daily_closing_prices.csv`.

### Part 1

Compute and plot in the (expected daily return, std deviation of daily return) space:

- Efficient frontier of risky assets under no-short-sales constraint;
- Minimum variance portfolio of risky assets under no-short-sales constraint;
- Maximum return portfolio of risky assets under no-short-sales constraint;
- Equally-weighted ( $1/N$ ) portfolio of risky assets;
- Initial portfolio from Assignment 1;
- Maximum Sharpe ratio portfolio under no-short-sales constraint;
- Risk-free asset;
- Efficient frontier of all assets including risk-free asset, if shorting of risk-free asset is allowed but shorting of maximum Sharpe ratio portfolio is prohibited (tangent line).

Assume risk free annual rate of 2.5%.

Clearly label all portfolios and frontiers on your plot.

You are allowed to use both CPLEX native interface and CPLEX via CVXPY interface for your computations.

### Part 2

Compute and plot in the (expected daily return, std deviation of daily return) space:

- Efficient frontier of risky assets under no-short-sales constraint;
- Individual risky assets;
- Randomly generated 1000 portfolios under no-short-sales restriction.

Use appropriate distribution for generating random numbers for portfolio weights and explain why you have chosen that particular form of the distribution. Normalize portfolio weights to sum up to one after you generated random numbers.

## Question 3

File `Daily_closing_prices.csv` contains prices of 20 stocks for years 2019-2020. In this question you are given the historical factor returns for the Fama–French three-factor model corresponding to the period 02–Jan–2019 to 31–Dec–2020 (`Daily_FF_factors.csv`). This includes the daily risk-free rate. Note that the risk-free rate is not a factor, but we can use it together with the asset data to compute the observed asset *excess* returns. In other words you must compute the asset returns, and then subtract the risk-free rate.

We can use factor models to explain the rates of return of our assets. In this question you will need to implement the Fama–French (FF) three-factor model described below.

The FF model is a multi-factor that explains the excess return of our assets. This ‘excess’ refers to the return in excess of the risk-free rate. According to the FF model, the excess returns of asset  $i$  can be described as follows

$$r_i - r_f = \alpha_i + \beta_{im}(f_m - r_f) + \beta_{is}f_s + \beta_{iv}f_v + \varepsilon_i$$

where  $r_i$  is the asset return,  $r_f$  is the risk-free rate,  $\alpha_i$  is the intercept from regression,  $\beta_{im}$  is the market factor loading,  $(f_m - r_f)$  is the market excess return factor,  $\beta_{is}$  is the size factor loading,  $f_s$  is the size factor,  $\beta_{iv}$  is the value factor loading,  $f_v$  is the value factor, and  $\varepsilon_i$  is the residual from regression.

The FF model is a linear regression model. The model should be calibrated by using ordinary least squares (OLS) regression. In other words, we can find the regression coefficients through OLS.

### Part 1

Find the factor loadings  $\beta = [\beta_{im}, \beta_{is}, \beta_{iv}]$  for each of the twenty asset using data in 2019, in 2020, and over the two years. Calculate the  $R^2$  value of all stocks from the FF model.

- Report fitted loadings,  $R^2$  value for all assets using three historic periods;
- Compare the  $R^2$  score in 2019 and 2020, suggest possible reason for the difference;
- If some investor select the Fama–French three-factor model to model asset return during COVID-19, would the performance of the model be affected? Explain your reason.

### Part 2

Suppose the three factors in the Fama–French three-factor model follow the following distribution:  $f = [f_m - r_f, f_s, f_v] \sim \mathcal{N}(\mathbf{0}, \Sigma)$ .

$$\Sigma = \begin{pmatrix} 2.6999 & 0.1351 & 0.4893 \\ 0.1351 & 0.5005 & 0.2170 \\ 0.4893 & 0.2170 & 1.4790 \end{pmatrix}$$

Your portfolio consists of 100 MSFT, 200 AAPL and 500 IBM shares (same as in Question 1) and you hold it from the beginning of January 2019 till the end of December 2020. You may use the risk free return given in `Daily_FF_factors.csv`. Compute 1-day 95% VaR for 100 MSFT, 200 AAPL and 500 IBM stocks separately and for the whole portfolio using Monte Carlo simulations, justify your selection of number of scenarios and steps, and report

your results. Compare the results with Part 2 in Question 1, analyze the reasons for the differences.

Is the following true? Explain:  $\text{VaR}(\text{Portfolio}) = \text{VaR}(\text{MSFT}) + \text{VaR}(\text{AAPL}) + \text{VaR}(\text{IBM})$

Would the answer change if we used Normal distribution model to calculate VaR?

## Report

In your report (up to 3-page, excluding graphs) describe how you answered questions. Include all required plots and discussions. Use only CPLEX optimization solver in your Python code (CVXPY library is allowed).

### Python Code to be Completed

```
# Optional Starter Code for Final Exam Project

# not mandatory to be used, you can start with your own codes

# Import libraries
import pandas as pd
import numpy as np
from os import path
import matplotlib.pyplot as plt

# Import other libraries
# Insert your code here #

# Read Daily Prices

# CSV file with price data
input_file_prices = 'Daily_closing_prices.csv'
if path.exists(input_file_prices):
    print('\nReading daily prices datafile - {}\n'.format(input_file_prices))
    fid = pd.read_csv(input_file_prices)
    # instrument tickers
    tickers = list(fid.columns)[1:]
    # time periods
    dates = fid['Date']
    data_prices = fid.values[:,1:]
else:
    print("No such file '{}'.format(input_file_prices), file=sys.stderr)

# Convert dates into array [year month day]
def convert_date_to_array(datestr):
    temp = [int(x) for x in datestr.split('/')]
    return [temp[-1], temp[0], temp[1]]

dates_array = np.array(list(fid['Date'].apply(convert_date_to_array)))

# Question 1

# Specify quantile level for VaR/CVaR
alf = 0.95

# Number of assets in universe
Na = data_prices.shape[1]

# Number of historical scenarios
Ns = data_prices.shape[0]

# Positions in the portfolio
positions = np.array([100, 0, 0, 0, 0, 0, 0, 0, 200, 500, 0, 0, 0, 0, 0, 0, 0, 0, 0]).reshape(Na,1)
```

```

##### Insert your code here #####

#print('Historical 1-day VaR %4.1f%% = %6.2f,      Historical 1-day CVaR %4.1f%% = %6.2f\n'%
#      (100*alf, VaR1, 100*alf, CVaR1))
#print('      Normal 1-day VaR %4.1f%% = %6.2f,      Normal 1-day CVaR %4.1f%% = %6.2f\n'%
#      (100*alf, VaR1n, 100*alf, CVaR1n))
#print('Historical 10-day VaR %4.1f%% = %6.2f,      Historical 10-day CVaR %4.1f%% = %6.2f\n'%
#      (100*alf, VaR10, 100*alf, CVaR10))
#print('      Normal 10-day VaR %4.1f%% = %6.2f,      Normal 10-day CVaR %4.1f%% = %6.2f\n'%
#      (100*alf, VaR10n, 100*alf, CVaR10n))

# Question 2

# Annual risk-free rate for years 2019-2020 is 2.5%
r_rf = 0.025

# Compute means and covariances for Question 2 (2019 and 2020)
cur_returns = data_prices[1:,:] / data_prices[:Ns-1,:] - 1
cur_returns = cur_returns

# Expected returns for Question 2
mu = np.mean(cur_returns, axis=0).reshape(cur_returns.shape[1],1)
# Covariances for Question 2
Q = np.cov(cur_returns.astype(float).T)

##### Insert your code here #####

# Question 3

# Import FF models data from 2019-01-03 to 2020-12-31
input_file_factors = 'Daily_FF_factors.csv'
if path.exists(input_file_factors):
    print('\nReading daily FF factors datafile - {} \n'.format(input_file_factors))
    ff_data = pd.read_csv(input_file_factors)
    factors_name = list(ff_data.columns)[1:-1]
    rf = ff_data['RF'][1:]
    factors = ff_data.values[1:,1:4]
else:
    print("No such file '{}'.format(input_file_factors), file=sys.stderr)

# Indices helpful to separate data
# dates[day_ind_start_2019] = '01/02/2019',
# dates[day_ind_end_2019] = '12/31/2019',
# dates[day_ind_start_2020] = '01/02/2020',
# dates[day_ind_end_2020] = '12/31/2019'.
day_ind_start_2019 = 0
day_ind_end_2019 = 251
day_ind_start_2020 = 252
day_ind_end_2020 = 504

data_prices_2019 = data_prices[day_ind_start_2019:day_ind_end_2019+1]
data_prices_2020 = data_prices[day_ind_start_2020:day_ind_end_2020+1]

##### Insert your code here #####

```