AcF 633 - Python Programming for Data Analysis

Manh Pham

Group Project 1

16th February 2023 noon/12pm to 2nd March 2023 noon/12pm (UK time)

This assignment contains one question worth 100 marks and constitutes 15% of the total marks for this course.

You are required to submit to Moodle **a single .zip folder** containing a single Jupyter Notebook .ipynb file (preferred) and/or Python script .py files and supporting .csv files (e.g. input data files, if any), together with a **signed group coversheet**. The **name of this folder must be your group number** (e.g. Group1.zip, where Group 1 is your group).

In your main script, either Jupyter Notebook .ipynb file or Python .py file, you do **not** have to retype the question for each task. However, you must clearly label which task (e.g. 1.1, 1.2, etc) your subsequent code is related to, either by using a markdown cell (for .ipynb files) or by using the comments (e.g. `#1.1` or ```` ```1.1``` ```` for .py files).

Your submission .zip folder must be submitted electronically via Moodle by the **2nd March 2022 noon/12pm (UK time)**. Only one of the group members needs to submit the work for your group. Late submissions (without pre-approved extensions) will receive a zero mark.

Good Luck!

Question 1:

The S&P100 index is a **market capitalization-weighted** index of 100 leading stocks listed in the US stock exchanges. The csv data file 'SP100-Feb2023.csv' lists the constituents of the S&P100 index as of 8 February 2023 with the following information:

- Ticker: Company's stock symbol or ticker
- Company: Company's name
- Sector: Sector in which the company belongs
- Market Value: Company's market capitalization

Import the data file to an object called "`Index`" in Python and perform the following tasks.

Task 1: Descriptive Analysis of S&P100 index (Σ = **25 marks**)

1.1: How many unique sectors are there in the S&P100 index? Print the following statement: 'There are ... unique sectors in the S&P100 index, namely ...', where the first '...' is the number of unique sectors, and the second '...' contains the names of the sectors alphabetically ordered and separated by commas. (**2 marks**)

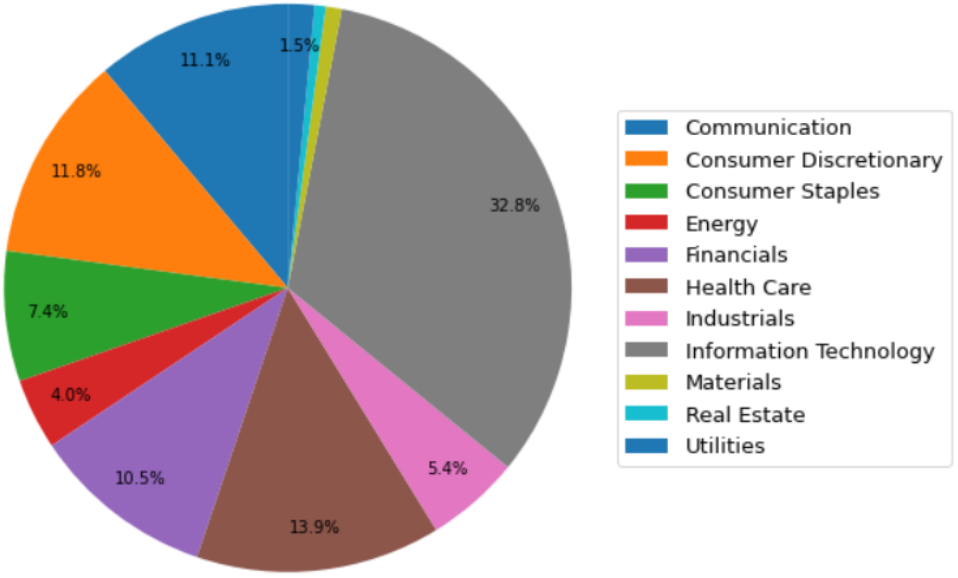
1.2: Write code to create a dictionary with keys being the unique sectors in the S&P100 index sorted in alphabetical order, and values being lists of tickers, also alphabetically ordered, in each sector.
Hint: An example of a key-value pair of the required dictionary is `'Materials': ['DOW', 'LIN']`. (**3 marks**)

1.3: Modify code in **Task 1.2** to create a dictionary with keys being the unique, alphabetically sorted sectors in the SP100 index, and values being tuples of two elements: the first being the number of tickers in each sector, and the second being the list of alphabetically ordered tickers in each sector.
Hint: An example of a key-value pair of the required dictionary is `'Materials': (2,['DOW', 'LIN'])`. (**3 marks**)

1.4: Add a column called "`Weighting`" to the "`Index`" object that computes the weight (in percentages, rounded to 3 decimal places) of each company in the SP100 index. (**3 marks**)

1.5: Write code to find the company having the largest index weight and the one with the smallest weight. Print the following statements:
Company ... (ticker ..., sector ...) has the largest index weight of ...%.
Company ... (ticker ..., sector ...) has the smallest index weight of ...%.
The range of the weights is ...%. (**3 marks**)

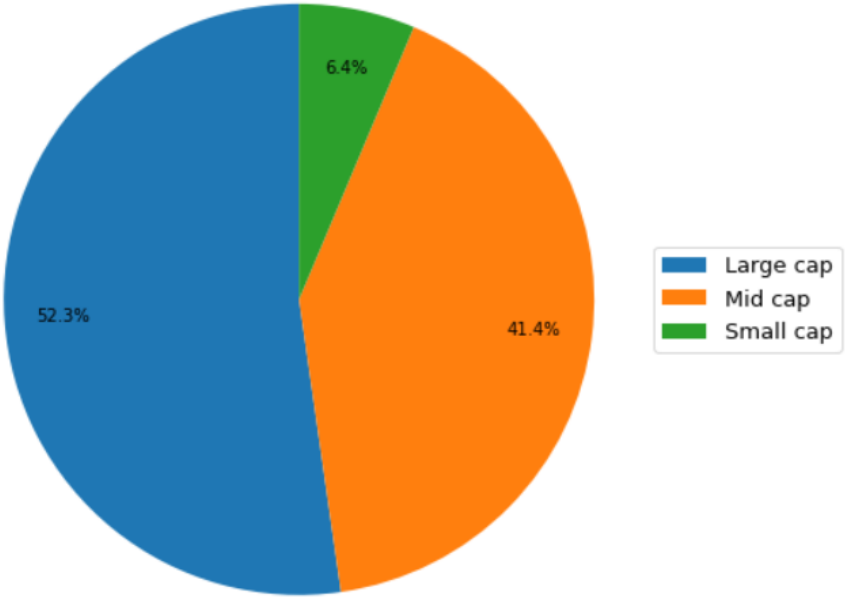
1.6: Write code to produce the following pie chart that shows the S&P100 index weighting by sectors. Only show the percentage weighting labels for sectors whose weights are larger than 1%.



Print the following statement:
Sector ... has the largest index weight of ...%, and Sector has the smallest index weight of ...%. (**6 marks**)

1.7: Write code to classify the S&P100 companies into 3 groups: "Large cap" when market value $\geq$ 100m, "Mid cap" when 100m $>$ market value $\geq$ 30m, and "Small cap" when market value $<$ 30m. Add a column called "`Group`" to the "`Index`" object to capture this grouping information. (**2 marks**)

1.8: Write code to produce the following pie chart that shows the S&P100 index weighting by company grouping. (**3 marks**)



Task 2: Portfolio Allocation (Σ = **35 marks**)

2.1: Using the order of your group number (e.g. 1, 2, 3, etc.) as a random seed, draw a random sample of 5 stocks (i.e. tickers) from the S&P100 index excluding stocks ABBV, AVGO, CHTR, DOW, GM, KHC, META, PYPL and TSLA.[1] Sort the stocks in alphabetical order, and then import daily Adjusted Close (Adj Close) prices for the 5 stocks between 01/01/2009 and 31/12/2022 from Yahoo Finance. Compute the **simple** daily returns for the stocks and drop days with NaN returns. (**3 marks**)

2.2: Create a data frame to summarize key statistics (including sample size, mean, standard deviation, minimum, quartiles, maximum, skewness, kurtosis, Jarque-Bera statistic, Jarque-Bera pvalue and Normality) for the daily returns of the five stocks over the above sample period. Jarque-Bera statistic is the statistic for the Jarque-Bera normality test that has the formula $JB = \frac{T}{6}\left(\widehat{S}^2 + \frac{(\widehat{K}-3)^2}{4}\right)$, where T is the sample size, $\widehat{S}$ and $\widehat{K}$ are sample skewness and kurtosis of data, respectively. Under the null hypothesis that data is normally distributed, the JB statistic follows a χ^2 distribution with 2 degrees of freedom. Jarque-Bera pvalue is the pvalue of the JB statistic under this χ^2 distribution. Normality captures the conclusion of the Jarque-Bera test - whether daily returns of a stock are normally distributed (Yes) or not (No), based on a 5% significance level.

Your data frame should look similar to the one below, but for the five stocks in your sample.

	COP	TMUS	V	WBA	WMT
count	3523.0000	3523.0000	3523.0000	3523.0000	3523.0000
mean	0.0007	0.0008	0.0010	0.0004	0.0004
std	0.0220	0.0247	0.0172	0.0179	0.0124
min	-0.2484	-0.3659	-0.1355	-0.1434	-0.1138
25%	-0.0097	-0.0098	-0.0074	-0.0081	-0.0054
50%	0.0004	0.0010	0.0013	0.0002	0.0005
75%	0.0111	0.0115	0.0092	0.0093	0.0063
max	0.2521	0.3678	0.1500	0.1260	0.1171
skewness	0.0215	-0.4014	0.2248	-0.1502	0.0941
kurtosis	17.2552	39.8125	11.5402	10.1748	17.8259
Jarque-Bera	29830.0901	199020.3531	10735.9093	7569.8301	32271.1326
JB pvalue	0.0000	0.0000	0.0000	0.0000	0.0000
Normality	No	No	No	No	No

(**5 marks**)

2.3: Using and/or modifying function `get_efficient_frontier()` from the file `Eff_Frontier_functions.py` on Moodle, construct and plot the Efficient Frontier for the five stocks based on **optimization** based on data over the above period. In your code, define an equally spaced range of expected portfolio

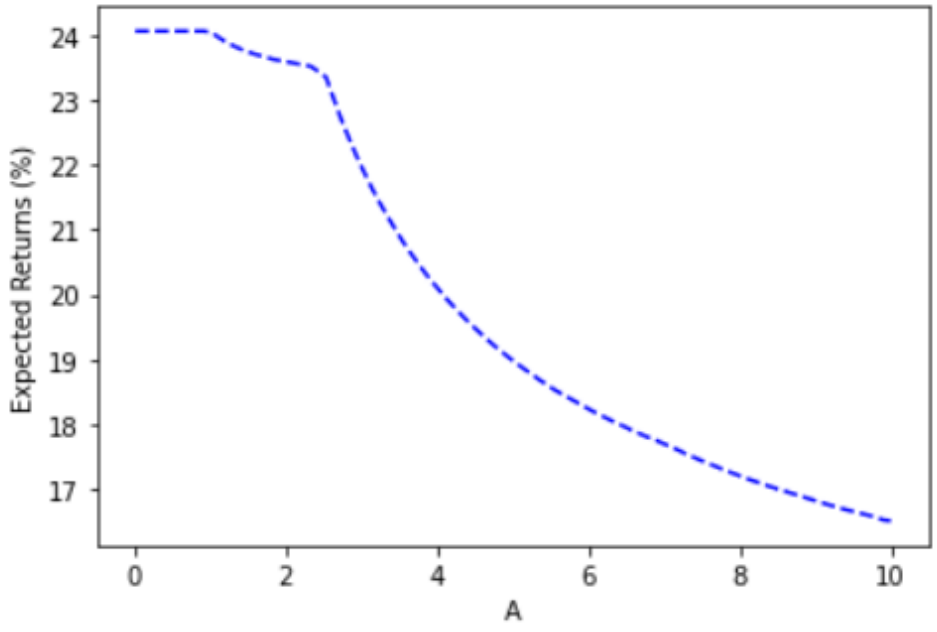
[1] These stocks only started trading after late 2009.

return targets with 2000 data points. Mark and label the locations of the five stocks in the Efficient Frontier plot. Also mark and label the locations of the Global Minimum Variance portfolio and the portfolio with the largest Sharpe ratio, assuming the annualized risk-free rate is 0.005 (or 0.5%).[2] **(6 marks)**

2.4: What are the return, volatility, Sharpe ratio and stock weights of the portfolio with the largest Sharpe ratio? Write code to answer the question and store the result in a Pandas Series object called `LSR_port` capturing the above statistics in percentages. Use the words 'return', 'volatility', 'Sharpe ratio', and stock tickers (in alphabetical order) to set the index of `LSR_port`. **(4 marks)**

2.5: Paul, a mean-variance optimizer, is interested in the five stocks in your sample. He has an expected utility function of the form $U(R_p) = E(R_p) - 0.5A\sigma_p^2$, where R_p and σ_p^2 are respectively the return and variance of the portfolio p, and A is Paul's risk-aversion coefficient. Assume A = 4. Also assume that Paul does **not** have access to a risk-free asset (i.e. he **cannot** lend or borrow money at the risk-free rate) and he would like to invest all of his wealth in the five stocks in your sample. How much, in percentages of his wealth, should Paul invest in each of the stocks in your sample to maximize his expected utility? Write code to answer the question and store the result in a Pandas Series object called `Paul_port` respectively capturing the return, volatility, Sharpe ratio and the stock weights of Paul's portfolio. Set the index of `Paul_port` correspondingly as in **Task 2.4**. **(5 marks)**

2.6: Paul is interested in knowing how the expected return of his optimal (i.e. maximum-utility) portfolio changes with his risk-aversion level A, when he does **not** have access to a risk-free asset. Write code to produce a similar plot to the following for your stock sample, setting A between 0 and 10. What can you conclude from the plot?



(5 marks)

[2]This equals the average of the risk-free rates over the sample period.

2.7: Now suppose that Paul has access to a risk-free asset and can borrow and lend money at the risk-free rate. In this case, he will choose the efficient portfolio with the largest Sharpe ratio in **Task 2.4** as his optimal risky portfolio and will divide his wealth between this optimal portfolio and the risk-free asset to maximize his expected utility. He could also borrow money (i.e. have a negative weight on the risk-free asset, which is assumed to be capped at -100%; that is, the maximum amount that he can borrow is equal to his wealth) to invest more in the risky assets. What will be his portfolio compositions in this case, assuming that his risk-aversion coefficient $A = 4$? Write code to answer the question and store the result in a Pandas Series object called `Paul_port_rf` respectively capturing the return, volatility, Sharpe ratio, the stock weights and risk-free asset weight of Paul's portfolio. Set the index of `Paul_port_rf` correspondingly. (**7 marks**)

Task 3: Factor models (Σ = **25 marks**)

3.1: Denote P be the portfolio formed by combining the five stocks in your sample using **equal** weights. Compute the returns of the portfolio P over the considered time period from 01/01/2009 to 31/12/2022. (**3 marks**)

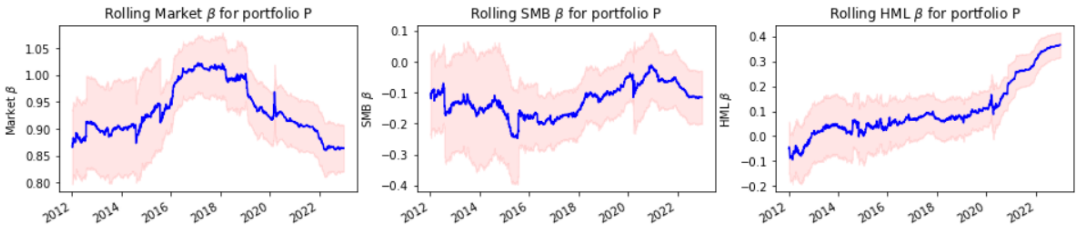
3.2: Using data from the Fama-French dataset, estimate a Fama-French three-factor model for portfolio P over the above period. Test if portfolio P possesses any abnormal returns that cannot be explained by the three-factor model. (**4 marks**)

3.3: Conduct the White test for the absence of heteroskedasticity in the residuals of the above factor model and draw your conclusion using a 5% significance level. (**3 marks**)

3.4: Conduct the Breusch-Godfrey test for the absence of serial correlation up to order 10 in the residuals of the above factor model and draw your conclusion using a 5% significance level. (**3 marks**)

3.5: Based on results in the above two tasks, update the Fama-French three-factor regression model and re-assess your conclusion on the pricing of portfolio P according to the three-factor model in **Task 3.2**. (**3 marks**)

3.6: Compute the 3-year rolling window β estimates of the Fama-French three factors for portfolio P over the sample period. That is, for each day, we compute β loadings for the three factors using the past 3-year data (including data on that day). Plot a 3×1 subplot figure similar to the following for your stock sample, showing the rolling window β estimates of the three factors, together with 95% confidence bands. Provide brief comments. (**9 marks**)



(Σ = **15 marks**)

Task 4: These marks will go to programs that are well structured, intuitive to use (i.e. provide sufficient comments for me to follow and are straightforward for me to run your code), generalisable (i.e. they can be applied to different sets of stocks, different utility functions for Paul with minimal adjustments/changes to the code) and elegant (i.e. code is neat and shows some degree of efficiency).