Investment & Portfolio Management - Assignment 1

Alifah Chusmarsyah

2025-11-14

Table of contents


```
import pandas as pd
```

Before we get into the data processing, we divide the data into 3 csv file to make it easier to conduct the analysis using Python.

```
#loading the dataset
Stock_price=pd.read_csv('Problem set_Stock Prices.csv')
Market_index=pd.read_csv('Problem set_Market Index.csv')
Risk_free=pd.read_csv('Problem set_Risk Free.csv')
```

Cleansing Part I - Cleansing Stock Price data

```
Stock_price.head()
```

	Date	Ticker	Company Name	Close Price	High Price	Low Price
0	2019-01	AAPL	APPLE INC	166.44	169.00	142.00
1	2019-02	AAPL	APPLE INC	173.15	175.87	165.93
2	2019-03	AAPL	APPLE INC	189.95	197.69	169.50
3	2019-04	AAPL	APPLE INC	200.67	208.48	188.38
4	2019-05	AAPL	APPLE INC	175.07	215.31	174.99

```
#checking whether the data type is already correct
print(Stock_price.dtypes)
```

```
Date             object
Ticker           object
Company Name     object
Close Price     float64
High Price      float64
Low Price       float64
dtype: object
```

```
#convert the data type to timestamp
Stock_price['Date'] = pd.to_datetime(Stock_price['Date'], errors='coerce')
Stock_price=Stock_price.drop(columns=['Company Name'])
```

```
#checking for any missing values
print(Stock_price.isnull().sum())
```

```
Date           0
Ticker         0
Close Price    0
High Price     0
Low Price      0
dtype: int64
```

```
#checking whether there is a negative stock price
if (Stock_price['Close Price'] < 0).any():
    print("Negative price detected in the dataset.")
else:
    print("NO Negative price detected in the dataset.")
```

```
NO Negative price detected in the dataset.
```

```
#calculate monthly returns based on closing price
Stock_price['Monthly Returns %'] = Stock_price.sort_values(['Ticker',
'Date']).groupby('Ticker')['Close Price'].pct_change() * 100
```

```
#see the top 2 data
Stock_price.head(2)
```

	Date	Ticker	Close Price	High Price	Low Price	Monthly Returns %
0	2019-01-01	AAPL	166.44	169.00	142.00	NaN
1	2019-02-01	AAPL	173.15	175.87	165.93	4.031483

```
#general statistical summary to understand our data
Stock_price.groupby("Ticker").agg(
    min_date=("Date","min"),
    max_date=("Date","max"),
    min_close=("Close Price", "min"),
    max_close=("Close Price", "max"),
    avg_close=("Close Price", "mean"))
```

Ticker	min_date	max_date	min_close	max_close	avg_close
AAPL	2019-01-01	2024-12-01	108.86	425.04	188.386944
AMZN	2019-01-01	2024-12-01	84.00	3507.07	1566.509722
GE	2019-01-01	2024-12-01	6.07	188.58	70.001389

```
Stock_price.groupby("Ticker").agg(
    min_daily_returns=("Monthly Returns %","min"),
    max_daily_returns=("Monthly Returns %","max"),
    avg_daily_returns=("Monthly Returns %","mean")
)
```

Ticker	min_daily_returns	max_daily_returns	avg_daily_returns
AAPL	−69.640504	18.863370	1.604891
AMZN	−95.582296	27.059599	0.532579
GE	−27.022059	713.976834	11.603734

The statistical summary above showed that the volatility of the close price is quite high with minimum and maximum difference more than 3000. The lowest daily returns is up to −90%. Therefore, it is necessary to conduct a deeper understanding these significant rate before the data are processed.

Based on the paper written by Maria and Zhuk, drop of stock prices can be attributed to split or reverse split premiums. In the paper, it is also stated that most of stock split in the United States will reflect to −50% stock price in the next day, or equal to 2:1. Since the companies in the dataset are under S&P and being traded in the United States, the data with a sudden significant change lower than −50% and more than 50% will be cleansed. The cleansing part will reflect the relevant news that justify the occurences.

```
#calculate how many days the daily return is lower than -50%
Stock_price['Split_Flag'] = Stock_price['Monthly Returns %'] <= -50
count_by_category = Stock_price.groupby('Ticker')['Split_Flag'].sum()

print(count_by_category)
```

```
Ticker
AAPL    1
AMZN    1
GE      0
Name: Split_Flag, dtype: int64
```

```
#drop split flag to make the visual on pdf better
Stock_price=Stock_price.drop(columns=['Split_Flag'])
```

```
#see the data detail for the stock price with daily returns less than 50%
Stock_price[Stock_price['Monthly Returns %'] <= -50]
```

	Date	Ticker	Close Price	High Price	Low Price	Monthly Returns %
19	2020-08-01	AAPL	129.04	131.00	107.8925	−69.640504
113	2022-06-01	AMZN	106.21	128.99	101.4300	−95.582296

```
#shows prior rows to check whether the split happened and the ratio. Then
cross check with relevant news to justify
selected_rows = Stock_price.iloc[[18, 19, 112, 113]]
selected_rows
```

	Date	Ticker	Close Price	High Price	Low Price	Monthly Returns %
18	2020-07-01	AAPL	425.04	425.66	356.5800	16.513158

	Date	Ticker	Close Price	High Price	Low Price	Monthly Returns %
19	2020-08-01	AAPL	129.04	131.00	107.8925	−69.640504
112	2022-05-01	AMZN	2404.19	2524.41	2025.2000	−3.276433
113	2022-06-01	AMZN	106.21	128.99	101.4300	−95.582296

```
#calculate how many days the daily return is more than 50%
Stock_price['Split_Flag'] = Stock_price['Monthly Returns %'] >= 50
count_by_category = Stock_price.groupby('Ticker')['Split_Flag'].sum()

print(count_by_category)
```

```
Ticker
AAPL    0
AMZN    0
GE      1
Name: Split_Flag, dtype: int64
```

```
#see the data detail for the stock price with daily returns more than 50%
Stock_price[Stock_price['Monthly Returns %'] >= 50]
```

	Date	Ticker	Close Price	High Price	Low Price	Monthly Returns %	Split_Flag
175	2021-08-01	GE	105.41	107.23	98.11	713.976834	True

According to the news on C Net Mihalcik (2022), there was a stock split for Amazon and Apple in August 2020 and June 2020. Due to this situation, we need to retrieve adjusted price so that we will not have a significant difference in stock prices. The news stated that AAPL experienced 4:1 stock split, therefore we need to divide the closing price by 4 before the split happened. Moreover, the stock split for AMZN is 20:1 and we will do the similar process for this ticker. On the other hand, it was observed that there is a significant change with more than 50% for GE. The news from General Electric Company (2021) stated that there was a reverse stock split with 1:8 in August 2021. Therefore, we will multiply the close price with 8 prior to the event of reverse stock split.

```
Stock_price['Date'] = pd.to_datetime(Stock_price['Date'], errors='coerce')
```

```
#calculating adjusted close price based on stocks split and reverse stocks
split
adjusted_close=[]
for i in range(len(Stock_price['Date'])):
    if Stock_price.loc[i, 'Ticker'] == "AAPL" and Stock_price.loc[i, 'Date'] <
```

```
pd.Timestamp('2020-08-01'):
        adjusted_close.append(Stock_price.loc[i, 'Close Price'] / 4)
    elif Stock_price.loc[i, 'Ticker'] == "AMZN" and Stock_price.loc[i, 'Date']
< pd.Timestamp('2022-06-01'):
        adjusted_close.append(Stock_price.loc[i, 'Close Price'] / 20)
    elif Stock_price.loc[i, 'Ticker'] == "GE" and Stock_price.loc[i, 'Date'] <
pd.Timestamp('2021-08-01'):
        adjusted_close.append(Stock_price.loc[i, 'Close Price'] * 8)
    else:
        adjusted_close.append(Stock_price.loc[i, 'Close Price'])
Stock_price['Adjusted Close']=adjusted_close
```

```
#checking the result for adjusted close price
Stock_price.iloc[172:176]
```

	Date	Ticker	Close Price	High Price	Low Price	Monthly Returns %	Split_Flag	Adjusted Close
172	2021-05-01	GE	14.06	14.40	12.72	7.164634	False	112.48
173	2021-06-01	GE	13.46	14.37	12.75	−4.267425	False	107.68
174	2021-07-01	GE	12.95	13.63	11.82	−3.789004	False	103.60
175	2021-08-01	GE	105.41	107.23	98.11	713.976834	True	105.41

```
#recalculating monthly returns based on adjusted close price
Stock_price['Final - Monthly Returns %'] = Stock_price.sort_values(['Ticker',
'Date']).groupby('Ticker')['Adjusted Close'].pct_change() * 100
```

```
#drop the Monthly Returns % data and fill NaN with 0 because the NaN is the
first data of the year therefore it can be filled with 0 since there wont be
any return %
Stock_price = Stock_price.drop(columns=['Monthly Returns %'])
Stock_price['Final - Monthly Returns %'] = Stock_price['Final - Monthly
Returns %'].fillna(0)
```

Cleansing Part II - Cleansing Market Index data

```
Market_index.head()
```

	Date	Level of the S&P 500 Index
0	2019-01	2704.10

	Date	Level of the S&P 500 Index
1	2019-02	2784.49
2	2019-03	2834.40
3	2019-04	2945.83
4	2019-05	2752.06

```
#checking whether the data type is already correct
print(Market_index.dtypes)
```

```
Date                          object
Level of the S&P 500 Index    float64
dtype: object
```

```
#convert the data type to timestamp
Market_index['Date'] = pd.to_datetime(Market_index['Date'], errors='coerce')
```

```
#checking for any missing values
print(Market_index.isnull().sum())
```

```
Date                          0
Level of the S&P 500 Index    0
dtype: int64
```

```
#checking whether there is a negative stock price
if (Market_index['Level of the S&P 500 Index'] < 0).any():
    print("Negative price detected in the dataset.")
else:
    print("NO Negative price detected in the dataset.")
```

```
NO Negative price detected in the dataset.
```

```
#recalculating monthly returns based on adjusted close price
Market_index['Market Monthly Returns %'] = Market_index['Level of the S&P 500
Index'].pct_change().fillna(0) * 100
Market_index.head(2)
```

	Date	Level of the S&P 500 Index	Market Monthly Returns %
0	2019-01-01	2704.10	0.000000

	Date	Level of the S&P 500 Index	Market Monthly Returns %
1	2019-02-01	2784.49	2.972893

Cleansing Part III - Cleansing Risk Free data

```
Risk_free.head()
```

	Date	Return on the T bill (in %)
0	2019-01	2.40
1	2019-02	2.43
2	2019-03	2.45
3	2019-04	2.43
4	2019-05	2.40

```
#checking whether the data type is already correct
print(Risk_free.dtypes)
```

```
Date                            object
Return on the T bill (in %)    float64
dtype: object
```

```
#convert the data type to timestamp
Risk_free['Date'] = pd.to_datetime(Risk_free['Date'], errors='coerce')
```

```
#checking for any missing values
print(Risk_free.isnull().sum())
```

```
Date                            0
Return on the T bill (in %)    0
dtype: int64
```

```
#checking whether there is a negative stock price
if (Risk_free['Return on the T bill (in %)'] < 0).any():
    print("Negative price detected in the dataset.")
else:
    print("NO Negative price detected in the dataset.")
```

```
NO Negative price detected in the dataset.
```

```
Risk_free['Date'].max()
#the number of period for the stocks and risk free returns data is the same
```

```
Timestamp('2024-12-01 00:00:00')
```

```
Risk_free['Returns']=(((1+(Risk_free['Return on the T bill (in
%)']/100))**(1/12))-1)*100
Risk_free.head(2)
```

	Date	Return on the T bill (in %)	Returns
0	2019-01-01	2.40	0.197833
1	2019-02-01	2.43	0.200279

Question a

When we are talking about portfolio, one of the most common questions is how to construct the optimal potfolio or the best risk-return combination. The variance or risk is tightly related to covariance, therefore, we need to take covariance into account when determining the best risk-return combination for a portfolio (Bodie, Kane, & Marcus, 2023). In this section, we will discuss Markowitz and Shrinkage method that can help us to determine appropriate covariance that is suitable for our dataset.

Markowitz Method

According to Markowitz Portfolio Optimization, the best combination can be attained by finding the weight of how much we want to invest, so that we can minimize the value of variance. The formula of variance includes covariance therefore the minimizing the covariance will impact in minimizing the variance itself. Hence, the best risk-return can be obtained by solving the minimization of the covariance.

$$\begin{aligned} \min_{w \in \mathbb{R}^N} \quad & w^\top \Sigma_t^* w \\ \text{subject to} \quad & w^\top r_t^* = r, \\ & \sum_{i=1}^{N} w_i = 1 \end{aligned} \tag{1}$$

Based on (1), there are several important variables, such as

$$w_i = weight$$
$$\Sigma^* = covariance\ between\ two\ returns$$
$$r_t^* = true\ expected\ returns$$

In reality, true value or the population for covariance matrix and expected returns are unknown. Therefore, we can only use historical sample to estimate the real value of covariance matrix and expected returns. Covariance matrix for historical sample data is proven to be an unbiased estimator for the true covariance matrix (Agrawal, Roy, & Uhler, 2022).

Study conducted by Palantaleo, et al. stated that Markowitz method uses sample covariance in calculating the variance. However, there are two things that need to be considered when using Markowitz approach. First, this method is very sensitive to error of estimating returns, nonetheless, we can ignore this because the method focuses on global minimum variance when estimating covariance matrix (Ingersoll, 1987, as cited in Pantaleo, Tumminello, Lillo, & Mantegna, 2018). Although the minimum global variance is not on the efficient frontier, but study shows that diversification can be improved by risk reduction more than maximization of returns (Jorion, 1985 as cited in Pantaleo, Tumminello, Lillo, & Mantegna, 2018;Bodie, Kane, & Marcus, 2023). Secondly, it is extremely reactive to the number of assets (N) and the number of period (T) (Pantaleo, Tumminello, Lillo, & Mantegna, 2018).

Shrinkage Method

True to it's name, Shrinkage method reduces the sample covariance matrix S into a target matrix T to produce a more robust estimator Q for the covariance matrix,

$$Q = \alpha T + (1 - \alpha)S \quad (2)$$

There are various shrinkage method, but in this section, we will consider Shrinkage method to common covariance where the diagonal elements of T are average of variance for the sample. This impact in the minimization of covariance and variance for the samples (Ledoit & Wolf, 2003 as cited in Pantaleo, Tumminello, Lillo, & Mantegna, 2018)

Method Selection and Reasoning

Pantaleo, et al. studied about different covariance matrices, which includes Markowitz and Shrinkage method. The study take into account two things, the number of period (T) and presence of negative weight or short selling. The result shows that Markowitz model performed best with N less than T and with no short selling. (Pantaleo, Tumminello, Lillo, & Mantegna, 2018).

In the given dataset, the number of assets are three and the period is 72 months. Therefore, T > N and moreover, there is no mention of short selling presence. Hence, Markowitz or direct sample covariance is the best method in risk reduction for a portfolio when measured against Shrinkage method.

Note: The true expected returns, following [Agrawal, Roy, & Uhler, 2022;Bodie, Kane, & Marcus, 2023; Sharpe, 1994], hence the symbol for returns is standardised for all equations.

```
#we need to transform the data by pivoting prior to calculate the covariance
matrix
returns_stock_price = Stock_price.pivot(index='Date', columns='Ticker',
values='Final - Monthly Returns %')
returns_stock_price.head()
```

Ticker	AAPL	AMZN	GE
Date			
2019-01-01	0.000000	0.000000	0.000000
2019-02-01	4.031483	−4.590599	2.263780
2019-03-01	9.702570	8.593574	−3.849856
2019-04-01	5.643590	8.185877	1.801802
2019-05-01	−12.757263	−7.861325	−7.177974

```
#covariance matrix calculation for the stocks
cov_matrix = returns_stock_price.cov()
print(cov_matrix)
```

```
Ticker       AAPL       AMZN          GE
Ticker
AAPL    66.826077  49.225376   24.528222
AMZN    49.225376  79.288269   16.707211
GE      24.528222  16.707211  111.830618
```

Result and Analysis for Question a

```
#put the covariance matrix result into a table
cov_table = cov_matrix.stack().to_frame(name='Covariance
Value').rename_axis(index=['Ticker1','Ticker2']).reset_index()
cov_table
```

	Ticker1	Ticker2	Covariance Value
0	AAPL	AAPL	66.826077
1	AAPL	AMZN	49.225376
2	AAPL	GE	24.528222
3	AMZN	AAPL	49.225376
4	AMZN	AMZN	79.288269

	Ticker1	Ticker2	Covariance Value
5	AMZN	GE	16.707211
6	GE	AAPL	24.528222
7	GE	AMZN	16.707211
8	GE	GE	111.830618

Methodoly that we used to evaluate covariance matrix between AMZN, AAPL and GE is Markowitz method which utilise the sample covariance matrix. The reasoning behind this selection is the number of assets is lower than the the number of monthly period. Hence, compared to Shrinkage method, Markowitz method is more suitable for our dataset.

The result for Question a showed that: 1. Covariance between AAPL and AMZN is 49.22 2. Covariance between AAPL and GE is 24.53 3. Covariance between AMZN and GE is 16.71

In general, covariance shows how stock A reacts to the movement of stock B, which means if the covariance is positive then both stocks move together in the same direction. AAPL and AMZN has higher covariance compared to AAPL and GE, and AMZN and GE. This is reasonable since AAPL and AMZN is in the same industry, which is Information Technology, while GE is Aerospace industry. Stocks within the same industry can be affected by the same industry sector situation, therefore the movement of the stocks can be aligned.

Question b

Sharpe Ratio is one of the parameter to evaluate the performance of portfolio. The ratio considered excess returns and standard deviation of the excess returns. Hence, the higher the Sharpe Ratio means the higher the excess returns and the portfolio is more preferable.

$$D_t \equiv r_{pt} - r_{ft} \quad (3)$$

$$\overline{D} \equiv \frac{1}{T}\sum_{t=1}^{T} D_t \quad (4)$$

$$Sh \equiv \frac{\overline{D}}{\sigma_D} \quad (5)$$

(Iworiso, 2020;Sharpe, 1994). According to Sharpe (1994), annualised Sharpe Ratio is usually the standard form to compare with other investment techniques (Sharpe, 1994). Hence, we will calculate both monthly and annualised Sharpe Ratio.

```
#calculate standard deviation and variance for Portfolio A, B and C
import math
weights = [[33.33/100, 33.33/100, 33.34/100],[40/100,40/100,20/100],
[25/100,25/100,50/10]]
var_AAPL = returns_stock_price["AAPL"].var()
```

```
var_AMZN = returns_stock_price["AMZN"].var()
var_GE = returns_stock_price["GE"].var()

avg_return_AAPL = returns_stock_price["AAPL"].mean()
avg_return_AMZN = returns_stock_price["AMZN"].mean()
avg_return_GE = returns_stock_price["GE"].mean()

AAPL_AMZN = cov_table.iloc[1,2]
AAPL_GE = cov_table.iloc[2,2]
GE_AMZN = cov_table.iloc[7,2]
var_Portfolio=[]
std_Portfolio=[]
for i in range(len(weights)):
    var = (((weights[i][0])**2)*(var_AAPL))+(((weights[i]
[1])**2)*(var_AMZN))+(((weights[i][2])**2)*(var_GE))
    +(2*(weights[i][0])*(weights[i][1])*AAPL_AMZN)+(2*(weights[i]
[0])*(weights[i][2])*AAPL_GE)
    +(2*(weights[i][1])*(weights[i][2])*GE_AMZN)
    var_Portfolio.append(round(var,4))
    std_Portfolio.append(round(math.sqrt(var),4))
print('Standard Deviation for Portfolio A, B and C is ',std_Portfolio)
print('Variance for Portfolio A, B and C is ',var_Portfolio)
print('\n')
print('Variance for AAPL stock is ',round(var_AAPL,2))
print('Variance for AMZN stock is ',round(var_AMZN,2))
print('Variance for GE stock is ',round(var_GE,2))
print('\n')
print('Avg return for AAPL stock is ',round(avg_return_AAPL,2))
print('Avg return for AMZN stock is ',round(avg_return_AMZN,2))
print('Avg return for GE stock is ',round(avg_return_GE,2))
```

```
Standard Deviation for Portfolio A, B and C is  [5.3537, 5.2775, 52.9613]
Variance for Portfolio A, B and C is  [28.6623, 27.8515, 2804.8976]


Variance for AAPL stock is  66.83
Variance for AMZN stock is  79.29
Variance for GE stock is  111.83


Avg return for AAPL stock is  2.85
Avg return for AMZN stock is  1.69
Avg return for GE stock is  1.55
```

```
#create a new data for stocks combined so it only shows date, returns for each
tickers and RF
```

```
Stocks=Stock_price[['Date', 'Ticker', 'Final - Monthly Returns %']]
AAPL = Stocks[Stocks['Ticker']=='AAPL'].rename(columns={'Final - Monthly
Returns %':'AAPL Returns'}).drop('Ticker', axis=1)
AMZN = Stocks[Stocks['Ticker']=='AMZN'].rename(columns={'Final - Monthly
Returns %':'AMZN Returns'}).drop('Ticker', axis=1)
GE = Stocks[Stocks['Ticker']=='GE'].rename(columns={'Final - Monthly Returns
%':'GE Returns'}).drop('Ticker', axis=1)
Stocks_combined = (AAPL.merge(AMZN, on='Date', how='left').merge(GE,
on='Date', how='left').merge(Risk_free, on='Date', how='left')).drop('Return
on the T bill (in %)', axis=1).rename(columns={'Returns':'RF Returns'})
```

```
#calculate excess returns between portfolio and risk free returns in monthly
manners
weights = [[33.33/100, 33.33/100, 33.34/100],[40/100,40/100,20/100],
[25/100,25/100,50/100]]
Stocks_combined['Portofolio_A']=(Stocks_combined['AAPL Returns']*weights[0]
[0])+(Stocks_combined['AMZN Returns']*weights[0][1])+(Stocks_combined['GE
Returns']*weights[0][2])
Stocks_combined['Portofolio_B']=(Stocks_combined['AAPL Returns']*weights[1]
[0])+(Stocks_combined['AMZN Returns']*weights[1][1])+(Stocks_combined['GE
Returns']*weights[1][2])
Stocks_combined['Portofolio_C']=(Stocks_combined['AAPL Returns']*weights[2]
[0])+(Stocks_combined['AMZN Returns']*weights[2][1])+(Stocks_combined['GE
Returns']*weights[2][2])
Stocks_combined['A-RF']=Stocks_combined['Portofolio_A']-Stocks_combined['RF
Returns']
Stocks_combined['B-RF']=Stocks_combined['Portofolio_B']-Stocks_combined['RF
Returns']
Stocks_combined['C-RF']=Stocks_combined['Portofolio_C']-Stocks_combined['RF
Returns']

print('Avg Portofolio A returns in monthly is ',
Stocks_combined['Portofolio_A'].mean())
print('Avg Portofolio B returns in monthly is ',
Stocks_combined['Portofolio_B'].mean())
print('Avg Portofolio C returns in monthly is ',
Stocks_combined['Portofolio_C'].mean())
print('\n')
print('Excess Returns for Portfolio A and RF in monthly',
round(Stocks_combined['A-RF'].mean(),4))
print('Excess Returns for Portfolio B and RF in monthly',
round(Stocks_combined['B-RF'].mean(),4))
print('Excess Returns for Portfolio C and RF in monthly',
round(Stocks_combined['C-RF'].mean(),4))
```

```
Avg Portofolio A returns in monthly is  2.0296307862488514
Avg Portofolio B returns in monthly is  2.125516235702295
Avg Portofolio C returns in monthly is  1.9098817916540705


Excess Returns for Portfolio A and RF in monthly 1.831
Excess Returns for Portfolio B and RF in monthly 1.9268
Excess Returns for Portfolio C and RF in monthly 1.7112
```

Result and Analysis for Question b

```
#calculate monthly Sharpe Ratio
print('Sharpe Ratio for Portfolio A (monthly)',Stocks_combined['A-RF'].mean()/
Stocks_combined['A-RF'].std())
print('Sharpe Ratio for Portfolio B (monthly)',Stocks_combined['B-RF'].mean()/
Stocks_combined['B-RF'].std())
print('Sharpe Ratio for Portfolio C (monthly)',Stocks_combined['C-RF'].mean()/
Stocks_combined['B-RF'].std())
```

```
Sharpe Ratio for Portfolio A (monthly) 0.2626864362138037
Sharpe Ratio for Portfolio B (monthly) 0.27228548132470604
Sharpe Ratio for Portfolio C (monthly) 0.24181387035358473
```

```
#calculate annualised Sharpe Ratio
import numpy as np
print('Sharpe Ratio for Portfolio A
(annualised)',np.sqrt(12)*(Stocks_combined['A-RF'].mean()/Stocks_combined['A-
RF'].std()))
print('Sharpe Ratio for Portfolio B
(annualised)',np.sqrt(12)*(Stocks_combined['B-RF'].mean()/Stocks_combined['B-
RF'].std()))
print('Sharpe Ratio for Portfolio C
(annualised)',np.sqrt(12)*(Stocks_combined['C-RF'].mean()/Stocks_combined['C-
RF'].std()))
```

```
Sharpe Ratio for Portfolio A (annualised) 0.9099725079630182
Sharpe Ratio for Portfolio B (annualised) 0.943224575635475
Sharpe Ratio for Portfolio C (annualised) 0.8119528743335713
```

```
#from the result above, we know that when heavily weighted in GE, the Sharpe
Ratio is the lowest compared to other strategies.
#we can try to find one relevant news that can justify this condition,
#which is when the difference between the returns with its mean is the highest
diff=[]
```

```
rows=[]
for i in range(len(Stocks_combined)):
    diff.append(abs(Stocks_combined['GE Returns'].iloc[i]-Stocks_combined['GE
Returns'].mean()))
    rows.append(i)
print('Highest difference between returns to its mean is for GE stocks
',round(max(diff),2))
idx = diff.index(max(diff))
corresponding_value = rows[idx]
print('The highest difference between monthly returns and its mean for GE
stocks happened in ',Stocks_combined['Date'].iloc[corresponding_value])
```

```
Highest difference between returns to its mean is for GE stocks  35.65
The highest difference between monthly returns and its mean for GE stocks
happened in  2020-11-01 00:00:00
```

Result

The monthly Sharpe Ratio result for portfolio B is higher than other portfolios. Hence, it means that the excess return of portfolio B is greater than its risk, which can make portfolio B more desirable for investors. The least Sharpe Ratio is portfolio C where the porfolio is heavily weighted for GE. As we can see from the result above, GE has the lowest average of returns with the highest volatility. This indicates that there was a high jump in GE stocks hence it was more volatile compared to AMZN and AAPL. The lower value volatility for AMZN and AAPL may be the result of:

Company focused analysis

We tried to find one of the largest jump or absolute difference between returns and its mean, and the result was 35% more compared to its mean. This happened in November 2020, where market showed a positive reation towards GE announcement of split the company into three public companies that focus on energy, healthcare and aviation (General Electric Company, 2021).

Industry focused analysis

AAPL and AMZN are within the same industry, which is the Technology industry. On the other hand, GE is in the Aerospace industry. The result showed that when the portfolio is not heavily dependent on GE, the Sharpe Ratio tend to have better results. Moreover, the volatility for AAPL and AMZN are lower than GE. Hence, we can try to find relevant news or circumstances that may have contributed to this number of volatility.

Within the year 2019 to 2024, there has been at least three times where the Aerospace industry has turbulance situations, which are:

1. According to Axios, in 2019, Boeing decided to put their manufacturing into a stop production for 737 max after hundreds of incidents (Axios, 2019).

2. According to Financial Times, in 2024, GE had to postpone their production plan due to delayed deliveries of Leap engine (Financial Times, 2024).
3. The effect of Covid-19, where as stated by Guardian, more than 50% of revenue of Gatwick airport had plunged (Guardian, 2020).

Overall, although in company wise, there has been a very high jump in monthly returns due to positive reactions from the market to the split of companies, on the other hand, from industry perspective, the stocks for Aerospace industry had experienced several hard times which might have impacted the volatility for monthly returns of stocks in the industry.

Question c

Based on the problem set, we already have the optimal portfolio with its variance and expected returns. Moreover, the problem state also gave us the risk aversion for each investors. Now, we would like to calculate the how much each investors willing to put their money on risky portfolio.

In order for us to calculate the amount of money invested in risky assets or usually represented by y, we need to find the maximization of Utility function first. The formula for Utility function is

$$U = E[r] - \frac{1}{2}A\sigma^2 \quad (6)$$

The Utility function represents how much expected returns an investors will receive given their risk aversion and the variance. In order for us to determine y, we need to maximize the U through derivative since the U function second derivative determine that it is a concave function, hence the critical point is the maximum.

$$\max_y U = r_f + y\left(E\left[r_p\right] - r_f\right) - \frac{1}{2}Ay^2\sigma^2 \quad (7)$$

$$y^* = \frac{E\left[r_p\right] - r_f}{A\sigma^2}. \quad (8)$$

where

(Bodie, Kane, & Marcus, 2023). Hence, from the problem set, we can get that the risk aversion for investor X is 2, Y is 4 and Z is 1/5.

Result and Analysis for Question c

```
#calculate y
print('optimal y for investor X is ', round((0.1)/(2*0.3),2))
print('optimal y for investor Y is ', round((0.1)/(4*0.3),2))
print('optimal y for investor Z is ', round((0.1)/((1/5)*0.3),2))
```

```
optimal y for investor X is  0.17
optimal y for investor Y is  0.08
optimal y for investor Z is  1.67
```

Analysis

The result showed that investor Z invested more than 1 to risky portfolio which means investor Z allocate more than 100% of their wealth to risky assets. Investor Z borrow at risk free rate to be able to invest more on risky assets. The result reflected the risk aversion for investor Z where the risk aversion is < 0 which means that investor Z is willing to take more risk for lower returns.

On the other hand, both investor X and Y have risk aversion coefficient more than 1, and resulted in allocating less than 20% of their wealth to risky assets. This means that both investors are not risk lovers.

Question d

The question stated that it assumed the single factor APT holds. Hence, the data is true under APT assumptions and can be modeled as

$$r_i = r_f + \beta_i F + e_i \quad (9)$$

where
$E(r_i)$ = expected excess return on stock i
β_i = sensitivity (factor loading) of firm i
F = deviation of the common factor from its expected value
e_i = nonsystematic (idiosyncratic) component of returns

The beta here is a constant that measures how sensitive our stock to the market excess returns. Since this equation is a linear equation, then we can estimate beta using different approaches, such as beta-return relationships or OLS regressions. Both methods will produce the same value,

beta in probability will converge to

$$\beta_i = \frac{\text{Cov}\ (r_i, r_M)}{\sigma_M^2} \quad (10)$$

(Bodie, Kane, & Marcus, 2023; Wooldridge, 2025)

```
import statsmodels.api as sm
Stocks_combined['AAPL - Rf'] = Stocks_combined['AAPL Returns'] -
Stocks_combined['RF Returns']
Stocks_combined['M - Rf'] = Market_index['Market Monthly Returns %'] -
Stocks_combined['RF Returns']

X = sm.add_constant(Stocks_combined['M - Rf'])
y = Stocks_combined['AAPL - Rf']
```

```
model = sm.OLS(y, X).fit()

print(model.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:              AAPL - Rf   R-squared:                       0.588
Model:                            OLS   Adj. R-squared:                  0.582
Method:                 Least Squares   F-statistic:                     99.82
Date:                Fri, 14 Nov 2025   Prob (F-statistic):           4.15e-15
Time:                        02:05:35   Log-Likelihood:                -221.14
No. Observations:                  72   AIC:                             446.3
Df Residuals:                      70   BIC:                             450.8
Df Model:                           1
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const          1.3720      0.637      2.154      0.035       0.102       2.642
M - Rf         1.2661      0.127      9.991      0.000       1.013       1.519
==============================================================================
Omnibus:                        0.220   Durbin-Watson:                   1.449
Prob(Omnibus):                  0.896   Jarque-Bera (JB):                0.366
Skew:                          -0.112   Prob(JB):                        0.833
Kurtosis:                       2.733   Cond. No.                         5.14
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is
correctly specified.
```

```
Stocks_combined['AMZN - Rf'] = Stocks_combined['AMZN Returns'] -
Stocks_combined['RF Returns']
Stocks_combined['M - Rf'] = Market_index['Market Monthly Returns %'] -
Stocks_combined['RF Returns']

X = sm.add_constant(Stocks_combined['M - Rf'])
y = Stocks_combined['AMZN - Rf']

model = sm.OLS(y, X).fit()

print(model.summary())
```

```
                            OLS Regression Results
==============================================================================
```

```
Dep. Variable:              AMZN - Rf   R-squared:                       0.402
Model:                            OLS   Adj. R-squared:                  0.393
Method:                 Least Squares   F-statistic:                     47.05
Date:                Fri, 14 Nov 2025   Prob (F-statistic):           2.25e-09
Time:                        02:05:35   Log-Likelihood:                -240.45
No. Observations:                  72   AIC:                             484.9
Df Residuals:                      70   BIC:                             489.5
Df Model:                           1
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const          0.3461      0.833      0.416      0.679      -1.315       2.007
M - Rf         1.1365      0.166      6.859      0.000       0.806       1.467
==============================================================================
Omnibus:                        2.272   Durbin-Watson:                   1.902
Prob(Omnibus):                  0.321   Jarque-Bera (JB):                1.530
Skew:                           0.267   Prob(JB):                        0.465
Kurtosis:                       3.475   Cond. No.                         5.14
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is
correctly specified.
```

```
Stocks_combined['GE - Rf'] = Stocks_combined['GE Returns'] -
Stocks_combined['RF Returns']
Stocks_combined['M - Rf'] = Market_index['Market Monthly Returns %'] -
Stocks_combined['RF Returns']

X = sm.add_constant(Stocks_combined['M - Rf'])
y = Stocks_combined['GE - Rf']

model = sm.OLS(y, X).fit()

print(model.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                GE - Rf   R-squared:                       0.333
Model:                            OLS   Adj. R-squared:                  0.323
Method:                 Least Squares   F-statistic:                     34.90
Date:                Fri, 14 Nov 2025   Prob (F-statistic):           1.14e-07
Time:                        02:05:35   Log-Likelihood:                -256.76
No. Observations:                  72   AIC:                             517.5
Df Residuals:                      70   BIC:                             522.1
```

```
Df Model:                           1
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const          0.1136      1.044      0.109      0.914      -1.969       2.197
M - Rf         1.2277      0.208      5.908      0.000       0.813       1.642
==============================================================================
Omnibus:                        6.938   Durbin-Watson:                   1.525
Prob(Omnibus):                  0.031   Jarque-Bera (JB):               11.144
Skew:                          -0.205   Prob(JB):                      0.00380
Kurtosis:                       4.883   Cond. No.                         5.14
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is
correctly specified.
```

From the result of OLS regression, we get beta for portfolio A as 1.2, portfolio B as 1.07 and portfolio C as 1.11. Moreover, all the beta is statistically significant since all the p value is less than 0.05. We will do second regression to get next month expected returns by utilising the estimated beta. More detail explanation about beta will be carry out in the result analysis part of this question.

Gürtler, M., & Witowski, E. (2024) performed a research about forecasting secondary market CAT bonds with low volatility with various models. The result showed that a simple regression model AR(1) for out-of-sample data performed the best compared to other models. The data used by the researchers had a low volatility with only 11%. In addition to that, in AR(1) modeling, he only includes premium of the previous period. Before we get into the coding part, we will define the AR(1) by equation

$$\text{premi}_t = \alpha + \beta\,\text{premi}_{t-1} + \varepsilon_t \quad (11)$$

We will conduct the analysis using AR(1) and see whether our data satisfy the condition and perform well.

Note: The beta alpha, following [Gürtler, M., & Witowski, E., 2024;Bodie, Kane, & Marcus, 2023; Wooldridge, 2025], hence the symbol for parameters that is relevant with linear model (alpha and beta) is standardised for all equations.

```python
#AR(1) modelling
from statsmodels.tsa.stattools import adfuller

AR1_result = adfuller(Stocks_combined['M - Rf'])
print("ADF Statistic:", AR1_result[0])
print("p-value:", AR1_result[1])
```

```
ADF Statistic: -7.655435164374947
p-value: 1.744108696319389e-11
```

```
#forecast result for market excess returns
from statsmodels.tsa.ar_model import AutoReg
AR1 = AutoReg(Stocks_combined['M - Rf'], lags=1)
test = AR1.fit()
phi = test.params[1]

market_forecast = phi * Stocks_combined['M - Rf'].iloc[-1]
print("Forecasted market excess return:", market_forecast)
```

```
Forecasted market excess return: 0.4826067859713544
```

```
/var/folders/v4/03r5h2xx3gx3nqz6kb7d67w40000gn/T/
ipykernel_4026/3799561923.py:5: FutureWarning: Series.__getitem__ treating
keys as positions is deprecated. In a future version, integer keys will always
be treated as labels (consistent with DataFrame behavior). To access a value
by position, use `ser.iloc[pos]`
  phi = test.params[1]
```

Result and Analysis for Question d

```
#forecast expected excess returns for AAPL stocks using CAPM
FE_AAPL = (1.2661*0.4826067859713544)
FE_AMZN = (1.1365*0.4826067859713544)
FE_GE = (1.2277*0.4826067859713544)
print('Expected excess return for AAPL stocks next month is ', FE_AAPL)
print('Expected excess return for AMZN stocks next month is ', FE_AMZN)
print('Expected excess return for GE stocks next month is ', FE_GE)
```

```
Expected excess return for AAPL stocks next month is  0.6110284517183319
Expected excess return for AMZN stocks next month is  0.5484826122564443
Expected excess return for GE stocks next month is  0.5924963511370318
```

Beta result analysis

All the betas are statistically significant since all the p-value are less than 0.05. The beta for AAPL is 1.2661 and R-squared is around 55%. Hence, it means that around 55%, the excess returns variation for AAPl can be explained by the market excess return. The beta for AMZN is 1.1365 and R-squared is around 40%. Therefore, excess return variation for AMZN can be explained by the market excess return by 40%. Last but not least, beta for GE is 1.2277 and R-squared is 33%, which means the market excess return are able to explain the GE excess return by 33%. Overall,

the beta for AAPL is the highest among the other stocks, hence AAPL is more sensitive to the market movement compared to GE and AMZN.

Forecasting excess return result analysis

We are estimating next month excess return for each stocks using AR(1) upon reading the research result from Gürtler, M., & Witowski, E. (2024). There is a similarity with our dataset with their dataset, such as we only consider one independent variable. Although the variation of our stocks are much higher than their secondary market CAT bonds, the model shows that our data is stationary and satisfy the AR(1) well with p-value less than 0.5. The result also shows that AAPL has the highest next month axcess return with 1.98.

Question e

Identify a factor (or factor tracking portfolio) which would be useful to use in addition to the market return. The factor should be something other than SMB, f) HML, or WML, and you can find such a factor by consulting academic papers such as those we discussed in class. In light of the "factor zoo" problems studied by Harvey, Liu, and Zhu (2016), explain why you believe this factor's predictive power is statistically significant.

Result and Analysis for Question e

We choose DCG, which satisfy Harvey (2016), DCG is common factor and also can be intrepreted in the economy. Moreover, its also satisfy the statistics requirement. On Harvey (2016) it is stated that DCG has a t ratio higher than 5.0. Harvey mentioned that DCG they observed is from Yogo (2006). According to Yogo (2006), estimated of utility weight for DCG is SE = 0.089 for $\alpha = 0.827$, which is statistcally significant. This means that DCG can explain well the utility weight hence it can determine customers satisfaction and expected returns. Hence, DCG can represent a factor in systemmatic risk.

Question f

Give either a rational (risk-based) or behavioral explanation for your new factor's predictive power. Do you think the factor betas for Apple, Amazon, and GE would be positive or negative? Which of those three stocks do you think would have the highest or lowest betas with respect to your factor? Explain your answers.

Result and Analysis for Question f

Our portfolios are constructed using a combination of AAPL, AMZN and GE stocks. These companies represent different types of industries. Therefore, we need to find a factor that can help us to explain how risky circumstances affect returns accross multiple industries. We choose Durable Consumer Goods (GCD) which is considered as a Common factor by Hervey (2016). A common factor is not a factor that is specified for a specific sector (Harvey, 2016). Moreover, the the data for durable consumer goods can be accessed publicly, for this reason, investors are able to retrive the information easily.

In general, GCD covers how returns differ accross different type of sectors and how premium moves in an opposite direction to business cycle. This model examines durable and non durable household consumption and how elasticity of substitution affects the consumer utility and portfolio returns. The research shows that:

1. When durable goods consumption falls, then people tend to be more dissatisfied.
2. Small stocks are more sensitive to changes in durable consumption growth. Hence, small stocks tend to have a higer beta.
3. During recession or bad times, investors demand expected higher risk returns. Conversely, during good times, lower premiums are acceptable (Yogo, 2006).

Before we get into details on rational and behavioural explanation, we will provide brief information about each stocks and what they covers in the industry.

AAPL or Apple is a technology company that manufacture electronic goods, software and online services. Apple considered as a company that provides durable goods.

AMZN or Amazon is a company that sells software and e-commerce for various of products. Amazon provides durable and non-durable goods.

GE or General Electric is a company that focus on aerospace, healthcare and energy sector. Hence, the company does not provide durable goods.

Rational Explanation

We will look at several risk-based scenarios and how it might impact each our stocks.

A. Risk Scenario 1: During a recession or economic downturn

During an economic downturn, household tend to spend less on durable goods and focus more on spending for non durable goods. The impact for our stocks is:

1. AAPL have a higher and positive beta since it is a manufacture company that sells durable goods, therefore more sensitive to changes in durable consumption growth.
2. AMZN provides a combination of durable and non-durable growth, hence DCG might not have impact as it has to Apple. Positive beta but less sensitive compared to Apple.
3. GE is a manufacture company that focus more on business to business sales, hence it might be least impacted by DCG contrast to AAPL and AMZN. Possibly have a very low beta among others.

B. Risk Scenario 2: During a pandemic or unprecedented events related to healthcare crisis

1. Household might be more selective on luxury goods spending and focus more on healthcare products. Hence, AAPL have a positive beta since people will spend less on technology.
2. AMZN might experience lower positive beta since AMZN provides online retail and delivery for consumer goods. During pandemics, people rely more on online shopping and delivery, hence this circumstance might not have a huge impact on AMZN.

3. One of GE subsidiaries is a company that focus on manufacturing healthcare machines and products. Therefore, during this situation, GE might have a higher demand on their products from companies or businesses. This can have a positive impact to GE stocks and it moves the opposite direction with household spending, hence beta can be low or slightly negative.

C. Risk Scenario 3: Supply chain disruption

Previously, we have stated that GE had experienced a supply chain problems during 2019-2024. As a result, this risk scenario should be take into consideration to see how the situation can impact the beta and stock returns movement.

In general, if a supply chain problem occurs then it can impact the delivery and/or price of products. Accordingly, it can determine consumers interest in buying the products.

1. DCG beta for AAPL for this period can be higher since supply chain can create an increase in customers electronic goods that Apple sells.
2. The beta for AMZN might be less contrast to AAPL since they provide a mixture of durable and non durable product. Hence, although supply chain disruption can delay the delivery or an price rise but not all their products can be considered as durable household products.
3. GE may experience a turbulance in their stock prices because of supply chain problems but might not have a huge beta since most of their business are B2B.

Overall, the beta for DCG can be high or that company can be very sensitive to DCG factors if they provide durable products for household. Hence, AAPL and AMZN tend to have greater beta contrast to GE in most risky circumstances. However, if we compare AAPL and AMZN, AAPL will have a greater DCG factor since it provides full durable household electronic goods, while AMZN have more various options with a combination of both durable and non durable products.

References

1. Iannino, Maria Chiara and Zhuk, Sergey, Signaling through Timing of Stock Splits (October 21, 2021). Available at SSRN: https://ssrn.com/abstract=2587113 or http://dx.doi.org/10.2139/ssrn.2587113
2. CNET. (2022, June 6). Amazon stock split, retail giant's first in 23 years, takes effect. https://www.cnet.com/personal-finance/amazon-stock-split-retail-giants-first-in-23-years-takes-effect/
3. General Electric Company. (2021, July 30). GE completes one-for-eight reverse stock split [Press release]. https://www.ge.com/news/press-releases/ge-completes-one-for-eight-reverse-stock-split
4. Agrawal, R., Roy, U., & Uhler, C. (2022). Covariance matrix estimation under total positivity for portfolio selection. Journal of Financial Econometrics, 20(2), 367–389. https://doi.org/10.1093/jjfinec/nbaa018
5. Pantaleo, E., Tumminello, M., Lillo, F., & Mantegna, R. N. (2018). When do improved covariance matrix estimators enhance portfolio optimization? An empirical comparative study of nine estimators.

6. Bodie, Z., Kane, A., & Marcus, A. J. (2023). Investments (13th ed.). McGraw Hill.
7. Ingersoll, J. E. (1987). Theory of financial decision making. Savage, MD: Rowman & Littlefield.
8. Jorion, P. (1985). International portfolio diversification with estimation risk. Journal of Business, 58(3), 259–278.
9. Ledoit, O., & Wolf, M. (2003). Improved estimation of the covariance matrix of stock returns with an application to portfolio selection. Journal of Empirical Finance, 10(5), 603–621. https://doi.org/10.1016/S0927-5398(03)00007-0
10. Sharpe, W. F. (1994). The Sharpe ratio. The Journal of Portfolio Management, 21(1), 49–58. https://doi.org/10.3905/jpm.1994.409501
11. Iworiso, J. (2020). On the predictability of U.S. stock market using machine learning and deep learning techniques [Doctoral thesis, University of Essex].
12. General Electric Company. (2021, November 9). GE plans to form three public companies focused on growth sectors of aviation, healthcare, and energy [Press release]. https://www.ge.com/news/press-releases/ge-plans-to-form-three-public-companies-focused-on-growth-sectors-of-aviation#:~:text=Press%20Release-,GE%
%20November%209%2C%202021%20–,shaping%20the%20future%20of%20flight.
13. Freedman, A. (2019, December 17). Buckle up: Boeing 737 shutdown indicates turbulence ahead. Axios. https://www.axios.com/2019/12/17/boeing-737-shutdown-indicates-turbulence-ahead
14. Pfeifer, S., & Georgiadis, P. (2024, July 23). Rolls-Royce boss warns of prolonged supply chain strains. Financial Times. https://www.ft.com/content/b9c7b25a-42f4-4543-a197-feb475d7a05e
15. The Guardian. (2020, August 28). Gatwick airport hit by £343 m loss as passenger numbers fall. https://www.theguardian.com/uk-news/2020/aug/28/gatwick-airport-hit-by-343m-loss-as-passenger-numbers-fall-covid-19
16. Yogo, M. (2006). A consumption-based explanation of expected stock returns. Journal of Finance, 61(2), 539-580. https://doi.org/10.1111/j.1540-6261.2006.00848.x
17. Wooldridge, J. M. (2025). Introductory econometrics: A modern approach [PDF ebook]. Cengage.
18. Gürtler, M., & Witowski, E. (2024). Superior forecasting with simple AR(1) models in a low-volatility environment: Evidence from the CAT bond market. Journal of Asset Management, 26(3), 255–270. https://doi.org/10.1057/s41260-024-00379-8