FIM 549 -FRA Project

SYMBOL:

AMERICAN EXPRESS (AXP)

Name: Karan Barde

Student ID: 200436222

Project Objective:

Use risk management techniques to perform calculation on the historical data of American Express (AXP) stock and the S&P Index (SPY) individually and in combination of a portfolio and then model the risk measures, analyze results and document them.

Submitting data and code in good standing – working condition

The report is submitted along with the python jupyter file in good working condition and used comments wherever necessary.

Things done in the project

- Estimated AXP and SPY (independently) log return volatility using GARCH (1,1) from data Jan 1, 2019 to Dec 31, 2021.
- Estimated the correlation between AXP and SPY.
- Estimated 99% VaR for AXP and SPY using historical simulation (equal weight) using data Jan 1, 2019 to Dec 31, 2021, for 3 independent portfolios A, B and C.
- Use the results of actual profit and loss on Jan 3rd, 2022 to assess and discuss which portfolio has the best risk adjusted performance, assuming 99% VaR is used for economic capital.

Preliminary Data Preparation:

The data of the stock prices are downloaded from “Yahoo Finance” for Tickers: “^GSPC” (S&P 500) and “AXP” (American Express). It was an excel spreadsheet (for raw data reporting) and can be taken within the code itself by using the ‘yfinance’ or ‘pandas_datareader’ packages in Python. I have also used the numpy and pandas package in python to load the data.

The GARCH Model (1,1)

The generalized autoregressive conditional heteroskedasticity (GARCH) process is an econometric term developed in 1982 by Robert F. Engle. It describes an approach to estimate volatility in financial markets. The goal of GARCH is to provide volatility measures for heteroscedastic time series data, much in the same way standard deviations are interpreted in simpler models.

The equation for the Garch(1,1) model is as shown

$$\sigma_n^2 = \gamma V_L + \alpha u_{n-1}^2 + \beta \sigma_{n-1}^2$$

where γ (gamma) is the weight assigned to VL, α (alpha) is the weight assigned to u2 n−1, and β(beta) is the weight assigned to σ2 n−1. Because the weights must sum to one, i.e.

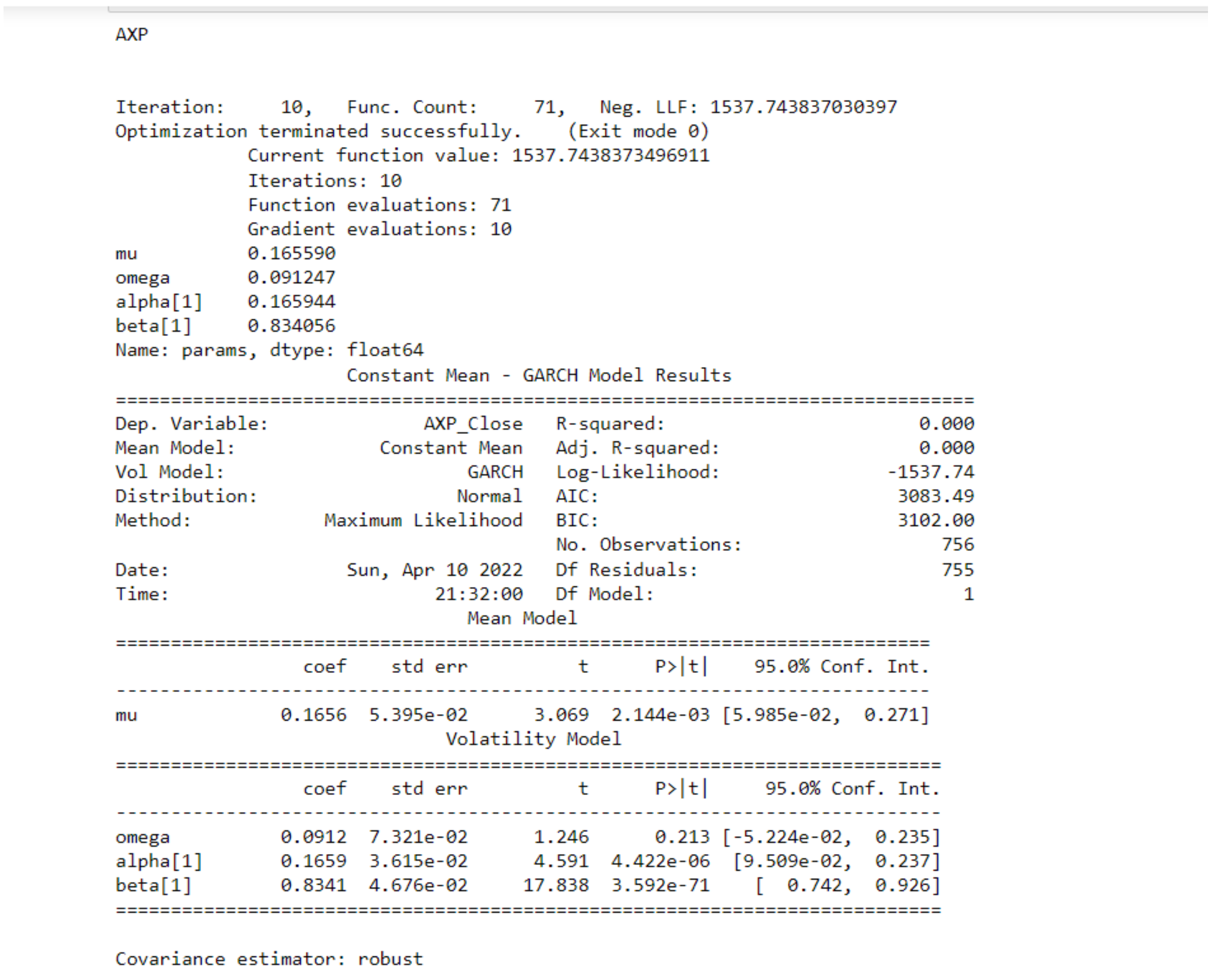
$$\gamma + \alpha + \beta = 1$$

Steps followed to get GARCH:

- Firstly, extract American Express (ticker = AXP) and S&P 500 (ticker = ^GSPC) data according to the required dates by using Yahoo Finance.
- After extracting the data, calculate the log returns of the close values for American Express (AXP)and S&P 500(^GSPC).
- Then use the arch_model() module from the ARCH package to build the GARCH (1,1) model. Moreover, in the model we need to specify a GARCH instead of ARCH model invol='GARCH' and assign lag parameters (1,1).

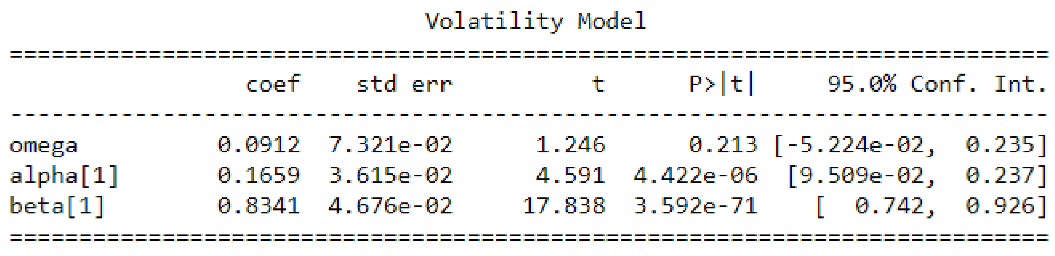
Results (covers question 1 and question 2.1)

Model for AXP



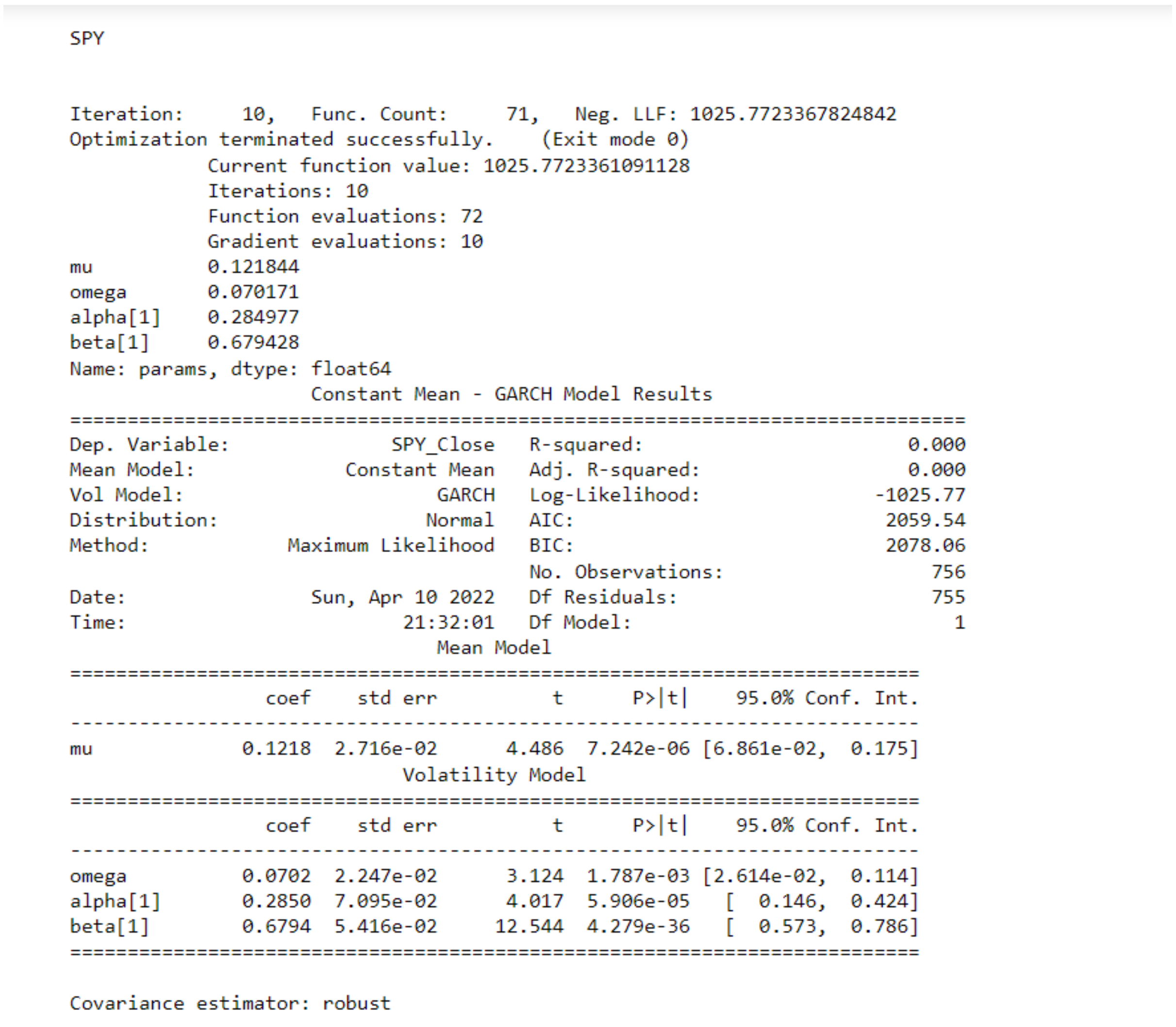
```
AXP


Iteration:     10,   Func. Count:     71,   Neg. LLF: 1537.743837030397
Optimization terminated successfully.    (Exit mode 0)
            Current function value: 1537.7438373496911
            Iterations: 10
            Function evaluations: 71
            Gradient evaluations: 10
mu          0.165590
omega       0.091247
alpha[1]    0.165944
beta[1]     0.834056
Name: params, dtype: float64
                     Constant Mean - GARCH Model Results
==============================================================================
Dep. Variable:              AXP_Close   R-squared:                       0.000
Mean Model:             Constant Mean   Adj. R-squared:                  0.000
Vol Model:                      GARCH   Log-Likelihood:               -1537.74
Distribution:                  Normal   AIC:                           3083.49
Method:            Maximum Likelihood   BIC:                           3102.00
                                        No. Observations:                  756
Date:                Sun, Apr 10 2022   Df Residuals:                      755
Time:                        21:32:00   Df Model:                            1
                                 Mean Model
==========================================================================
                 coef    std err          t      P>|t|     95.0% Conf. Int.
--------------------------------------------------------------------------
mu             0.1656  5.395e-02      3.069  2.144e-03 [5.985e-02,  0.271]
                              Volatility Model
===========================================================================
                 coef    std err          t      P>|t|      95.0% Conf. Int.
---------------------------------------------------------------------------
omega          0.0912  7.321e-02      1.246      0.213 [-5.224e-02,  0.235]
alpha[1]       0.1659  3.615e-02      4.591  4.422e-06  [9.509e-02,  0.237]
beta[1]        0.8341  4.676e-02     17.838  3.592e-71    [  0.742,  0.926]
===========================================================================

Covariance estimator: robust
```

GARCH (1,1) parameters for AXP is summarized as follows:



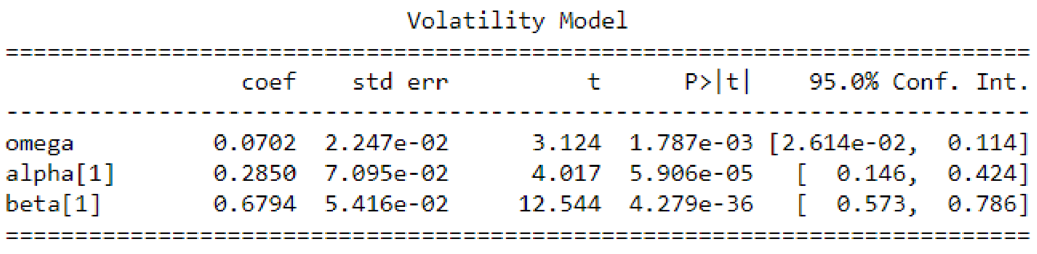
```
                              Volatility Model
===========================================================================
                 coef    std err          t      P>|t|      95.0% Conf. Int.
---------------------------------------------------------------------------
omega          0.0912  7.321e-02      1.246      0.213 [-5.224e-02,  0.235]
alpha[1]       0.1659  3.615e-02      4.591  4.422e-06  [9.509e-02,  0.237]
beta[1]        0.8341  4.676e-02     17.838  3.592e-71    [  0.742,  0.926]
===========================================================================
```

Model for S&P index



```
SPY


Iteration:     10,   Func. Count:     71,   Neg. LLF: 1025.7723367824842
Optimization terminated successfully.    (Exit mode 0)
            Current function value: 1025.7723361091128
            Iterations: 10
            Function evaluations: 72
            Gradient evaluations: 10
mu          0.121844
omega       0.070171
alpha[1]    0.284977
beta[1]     0.679428
Name: params, dtype: float64
                     Constant Mean - GARCH Model Results
==============================================================================
Dep. Variable:                SPY_Close   R-squared:                       0.000
Mean Model:               Constant Mean   Adj. R-squared:                  0.000
Vol Model:                        GARCH   Log-Likelihood:                -1025.77
Distribution:                    Normal   AIC:                            2059.54
Method:              Maximum Likelihood   BIC:                            2078.06
                                          No. Observations:                   756
Date:                  Sun, Apr 10 2022   Df Residuals:                       755
Time:                          21:32:01   Df Model:                             1
                                 Mean Model
==========================================================================
                 coef    std err          t      P>|t|    95.0% Conf. Int.
--------------------------------------------------------------------------
mu             0.1218  2.716e-02      4.486  7.242e-06 [6.861e-02,  0.175]
                              Volatility Model
==========================================================================
                 coef    std err          t      P>|t|    95.0% Conf. Int.
--------------------------------------------------------------------------
omega          0.0702  2.247e-02      3.124  1.787e-03 [2.614e-02,  0.114]
alpha[1]       0.2850  7.095e-02      4.017  5.906e-05   [ 0.146,  0.424]
beta[1]        0.6794  5.416e-02     12.544  4.279e-36   [ 0.573,  0.786]
==========================================================================

Covariance estimator: robust
```

GARCH(1,1) for SPY index is summarized as follows:



```
                              Volatility Model
==========================================================================
                 coef    std err          t      P>|t|    95.0% Conf. Int.
--------------------------------------------------------------------------
omega          0.0702  2.247e-02      3.124  1.787e-03 [2.614e-02,  0.114]
alpha[1]       0.2850  7.095e-02      4.017  5.906e-05   [ 0.146,  0.424]
beta[1]        0.6794  5.416e-02     12.544  4.279e-36   [ 0.573,  0.786]
==========================================================================
```

Question 2.2) What are the long-term volatilities of your symbol and the SPY based on the GARCH (1,1)? And Question 2.3) On Jan 3th, 2022 (first trading day of the year) what are the projected volatilities of your symbol and SPY?

Long term volatility

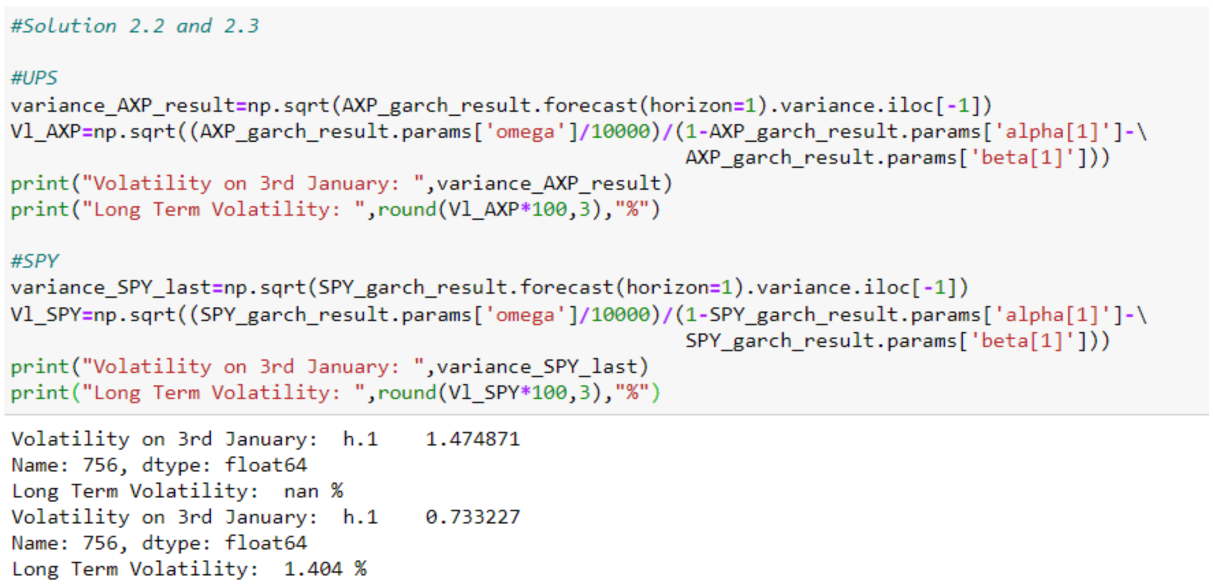
From the formulae:

$\boldsymbol{\omega} = \boldsymbol{\gamma V_L}$, where VL is the long term volatility and the equation $\boldsymbol{\gamma + \alpha + \beta = 1}$, we can get:

$$V_L = \frac{\omega}{1 - \alpha - \beta}$$

Using this equation we can calculate the long term volatilities.

V_L is long term variance rate



```
#Solution 2.2 and 2.3

#UPS
variance_AXP_result=np.sqrt(AXP_garch_result.forecast(horizon=1).variance.iloc[-1])
Vl_AXP=np.sqrt((AXP_garch_result.params['omega']/10000)/(1-AXP_garch_result.params['alpha[1]']-\
                                                         AXP_garch_result.params['beta[1]']))
print("Volatility on 3rd January: ",variance_AXP_result)
print("Long Term Volatility: ",round(Vl_AXP*100,3),"%")

#SPY
variance_SPY_last=np.sqrt(SPY_garch_result.forecast(horizon=1).variance.iloc[-1])
Vl_SPY=np.sqrt((SPY_garch_result.params['omega']/10000)/(1-SPY_garch_result.params['alpha[1]']-\
                                                         SPY_garch_result.params['beta[1]']))
print("Volatility on 3rd January: ",variance_SPY_last)
print("Long Term Volatility: ",round(Vl_SPY*100,3),"%")
```

```
Volatility on 3rd January:  h.1    1.474871
Name: 756, dtype: float64
Long Term Volatility:  nan %
Volatility on 3rd January:  h.1    0.733227
Name: 756, dtype: float64
Long Term Volatility:  1.404 %
```

Question 3) Estimate the correlation between the returns of your symbol and SPY, state the method/formula that your estimation is based on

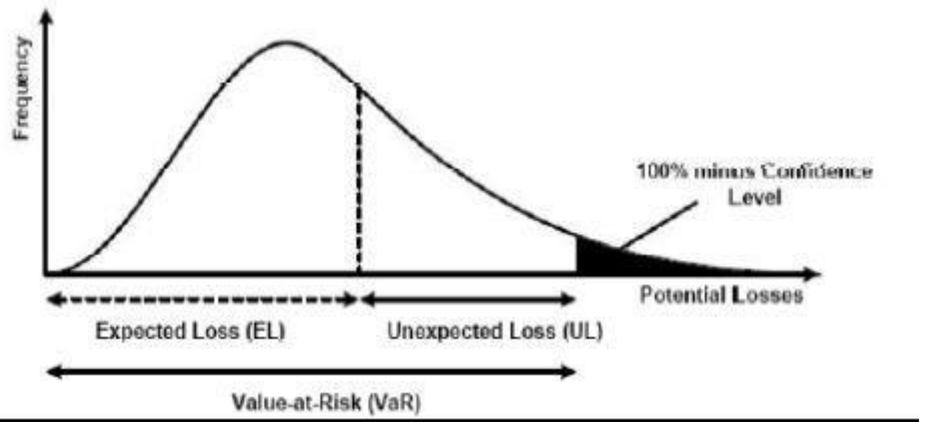
The correlation coefficient describes linear relationship between variables. Using the .corr() the correlation is calculated. The data has date, AXP stock price and SPY stock price. So, we first find the log return and then calculate the correlation.

```
#Solution 3
#Correlation of returns
np.log(data.drop(['Date'], axis = 1)/data.drop(['Date'], axis = 1).shift(1)).corr()
```

	AXP_Close	SPY_Close
AXP_Close	1.000000	0.757087
SPY_Close	0.757087	1.000000

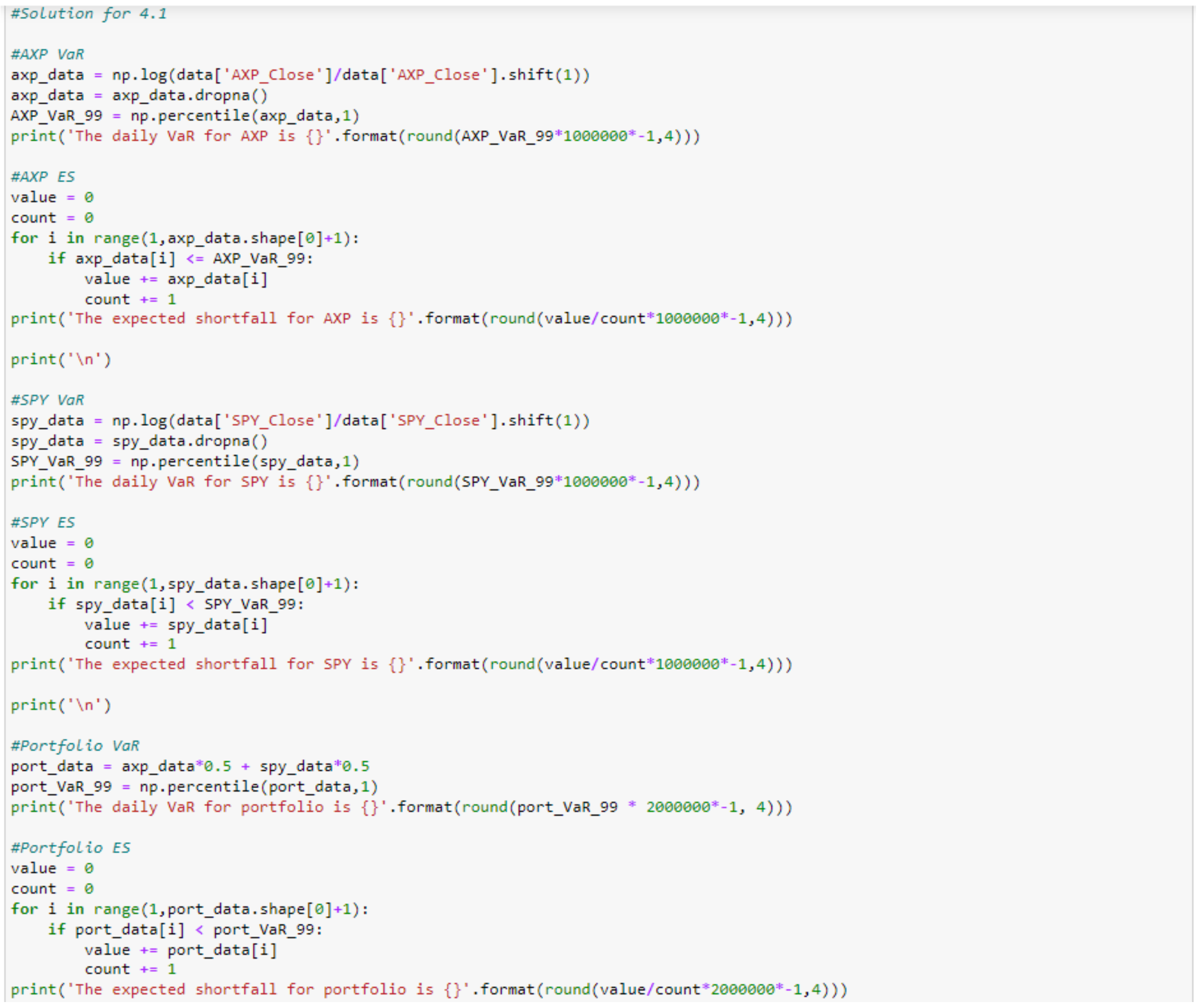
Question 4

Value-at-risk is a statistical measure of the riskiness of financial entities or portfolios of assets. Value at risk (VaR) is a statistic used to try and quantify the level of financial risk within a firm or portfolio over a specified time frame. VaR provides an estimate of the maximum loss from a given position or portfolio over a period of time, and you can calculate it across various confidence levels.

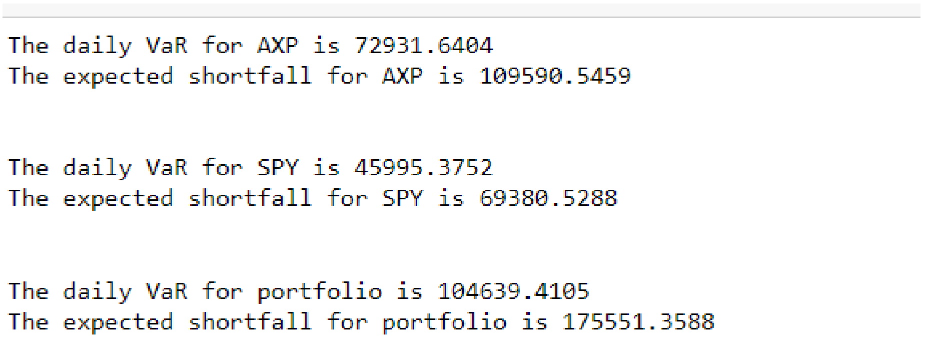


Expected Shortfall: Expected shortfall (ES) is a risk measure—a concept used in the field of financial risk measurement to evaluate the market risk or credit risk of a portfolio. The "expected shortfall at q% level" is the expected return on the portfolio in the worst. of cases. To calculate ES, we calculate the log returns for each date that is the daily log returns. The returns are arranged in ascending order. To calculate 99% VaR which is equal to 1 percentile of the value is calculated using np.percentile.

4.1) Report the one-day 99% (historical) VaR and ES of portfolios (A), (B), (C) with historical simulation. What is the diversification benefit in terms of VaR?



```
#Solution for 4.1

#AXP VaR
axp_data = np.log(data['AXP_Close']/data['AXP_Close'].shift(1))
axp_data = axp_data.dropna()
AXP_VaR_99 = np.percentile(axp_data,1)
print('The daily VaR for AXP is {}'.format(round(AXP_VaR_99*1000000*-1,4)))

#AXP ES
value = 0
count = 0
for i in range(1,axp_data.shape[0]+1):
    if axp_data[i] <= AXP_VaR_99:
        value += axp_data[i]
        count += 1
print('The expected shortfall for AXP is {}'.format(round(value/count*1000000*-1,4)))

print('\n')

#SPY VaR
spy_data = np.log(data['SPY_Close']/data['SPY_Close'].shift(1))
spy_data = spy_data.dropna()
SPY_VaR_99 = np.percentile(spy_data,1)
print('The daily VaR for SPY is {}'.format(round(SPY_VaR_99*1000000*-1,4)))

#SPY ES
value = 0
count = 0
for i in range(1,spy_data.shape[0]+1):
    if spy_data[i] < SPY_VaR_99:
        value += spy_data[i]
        count += 1
print('The expected shortfall for SPY is {}'.format(round(value/count*1000000*-1,4)))

print('\n')

#Portfolio VaR
port_data = axp_data*0.5 + spy_data*0.5
port_VaR_99 = np.percentile(port_data,1)
print('The daily VaR for portfolio is {}'.format(round(port_VaR_99 * 2000000*-1, 4)))

#Portfolio ES
value = 0
count = 0
for i in range(1,port_data.shape[0]+1):
    if port_data[i] < port_VaR_99:
        value += port_data[i]
        count += 1
print('The expected shortfall for portfolio is {}'.format(round(value/count*2000000*-1,4)))
```



```
The daily VaR for AXP is 72931.6404
The expected shortfall for AXP is 109590.5459


The daily VaR for SPY is 45995.3752
The expected shortfall for SPY is 69380.5288


The daily VaR for portfolio is 104639.4105
The expected shortfall for portfolio is 175551.3588
```

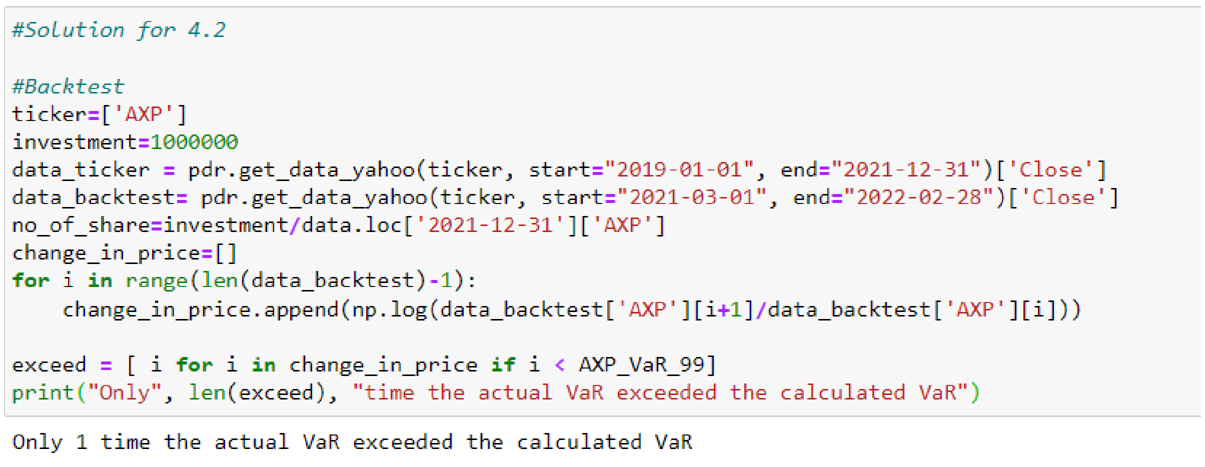
Diversification benefit of VAR:

To reduce the risk of portfolio, we diversify the portfolio with different asset class which reduces the risk of portfolio. Due to portfolio diversification, there is lower Var compared to portfolio with no diversification**.**

VaR(A) + VaR(B) – Var(C) = `72931.6404 + 45995.3752 - 104639.4105 = 14,287.60`

Question 4.2) Back test your estimated VaRs above using data March 1, 2021-Feb 28, 2022. Back test the one-day 99% VaR of portfolio (A) using Basel (1996) method (hint: you can either scale to $1 million or just use the unit share price.)

Backtesting is the process of comparing losses predicted by a value at risk (VaR) model to those actually experienced over the testing period. It is done to ensure that VaR models are reasonably accurate. For backtesting from 2021/03/01 to 2022/02/28, we checked the number of times the daily loss is more than VaR



```python
#Solution for 4.2

#Backtest
ticker=['AXP']
investment=1000000
data_ticker = pdr.get_data_yahoo(ticker, start="2019-01-01", end="2021-12-31")['Close']
data_backtest= pdr.get_data_yahoo(ticker, start="2021-03-01", end="2022-02-28")['Close']
no_of_share=investment/data.loc['2021-12-31']['AXP']
change_in_price=[]
for i in range(len(data_backtest)-1):
    change_in_price.append(np.log(data_backtest['AXP'][i+1]/data_backtest['AXP'][i]))

exceed = [ i for i in change_in_price if i < AXP_VaR_99]
print("Only", len(exceed), "time the actual VaR exceeded the calculated VaR")
```

```
Only 1 time the actual VaR exceeded the calculated VaR
```

4.3) Does the backtesting result surprise you? Why

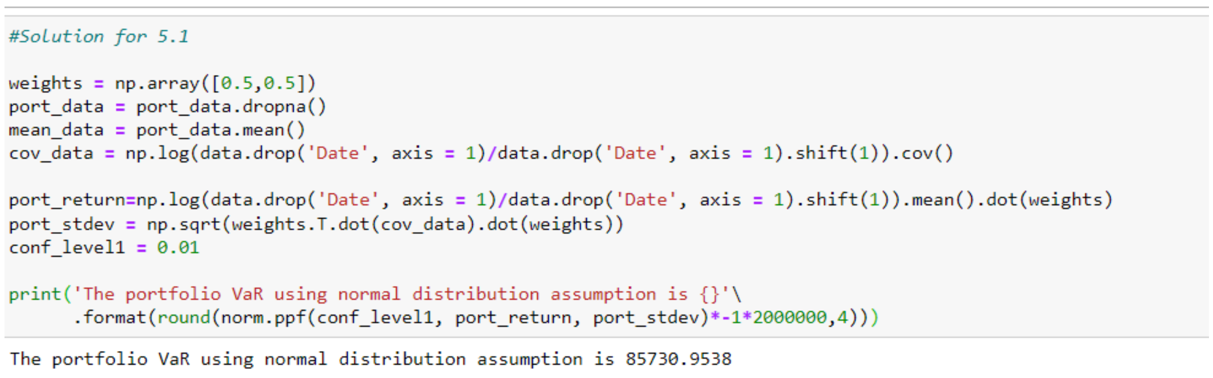
Yes the results are surprising as the markets have performed well over 1 year and it resulted in 1 day having daily loss less than Var.

4.4) Propose one method that may improve your VaR estimation and explain why your proposed method might be better than the plain vanilla historical simulation method?

One Method that can improve VAR estimation is using Monte Carlo Simulation.

- Monte Carlo account for nonlinear of portfolio values w.r.t. underlying risk factor and also incorporates fat tails and time varying volatilities
- The key difference between historical simulation and simulation Monte Carlo is that the historical simulation model carries out the simulation using the real observed changes in the market place over the last X periods (using historical market price data) to generate Y hypothetical portfolio profits or losses, whereas in the Monte Carlo simulation a random number generator is used to produce tens of thousands of hypothetical changes in the market. These are then used to construct thousands of hypothetical profits and losses on the current portfolio, and the subsequent distribution of possible portfolio profit or loss.

5.1) Use the volatility and correlation for your symbol and SPY and calculate one day 99% VaR for portfolio (C) using normal distribution assumption.



```
#Solution for 5.1

weights = np.array([0.5,0.5])
port_data = port_data.dropna()
mean_data = port_data.mean()
cov_data = np.log(data.drop('Date', axis = 1)/data.drop('Date', axis = 1).shift(1)).cov()

port_return=np.log(data.drop('Date', axis = 1)/data.drop('Date', axis = 1).shift(1)).mean().dot(weights)
port_stdev = np.sqrt(weights.T.dot(cov_data).dot(weights))
conf_level1 = 0.01

print('The portfolio VaR using normal distribution assumption is {}'\
      .format(round(norm.ppf(conf_level1, port_return, port_stdev)*-1*2000000,4)))
```

```
The portfolio VaR using normal distribution assumption is 85730.9538
```

5.2) Compare the 99% VaR for portfolio (C) from 4-1) and 5-1). Which approach leads to a larger VaR? What could be the reason for this difference?

The Var calculated in 4.1 is higher than that calculated in 5.1 using normal distribution as normal distribution considers thin tails and no skewness. But the var from the market is not normally distributed and has fatter tails.

6. Use the results above and the actual profit and loss on Jan 4th, 2021 to assess and discuss which portfolio (i.e. A, B, or C) has the best risk adjusted performance, assuming 99% VaR is used for economic capital.

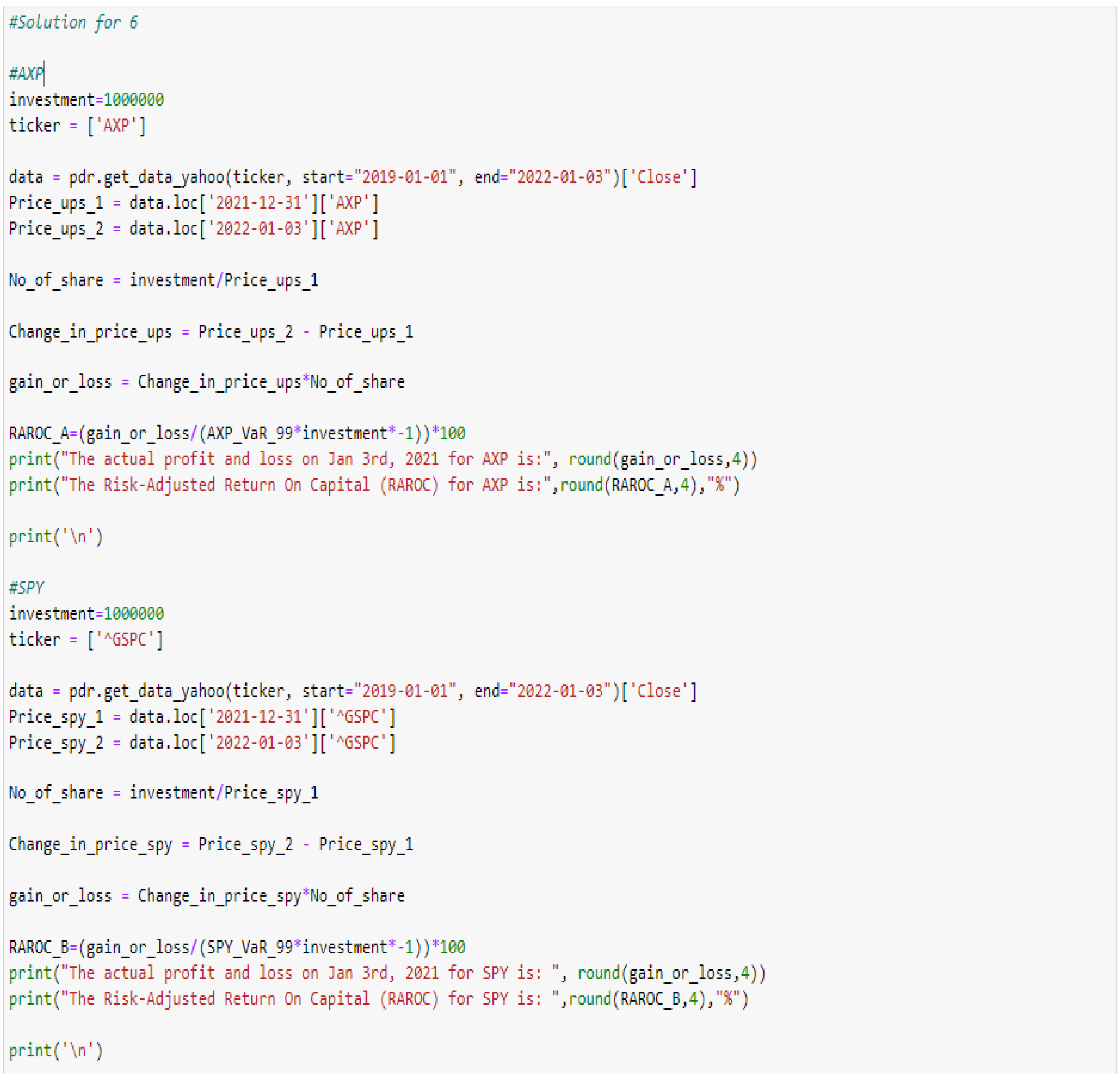
Using the results of actual profit and loss on Jan 3rd, 2022 , we discuss which portfolio has best risk adjusted performance assuming 99% VaR for economic capital.

Risk-adjusted return on capital (RAROC) is a risk-adjusted measure of the return on investment. It does this by accounting for any expected losses and income generated by capital, with the assumption that riskier projects should be accompanied by higher expected returns.

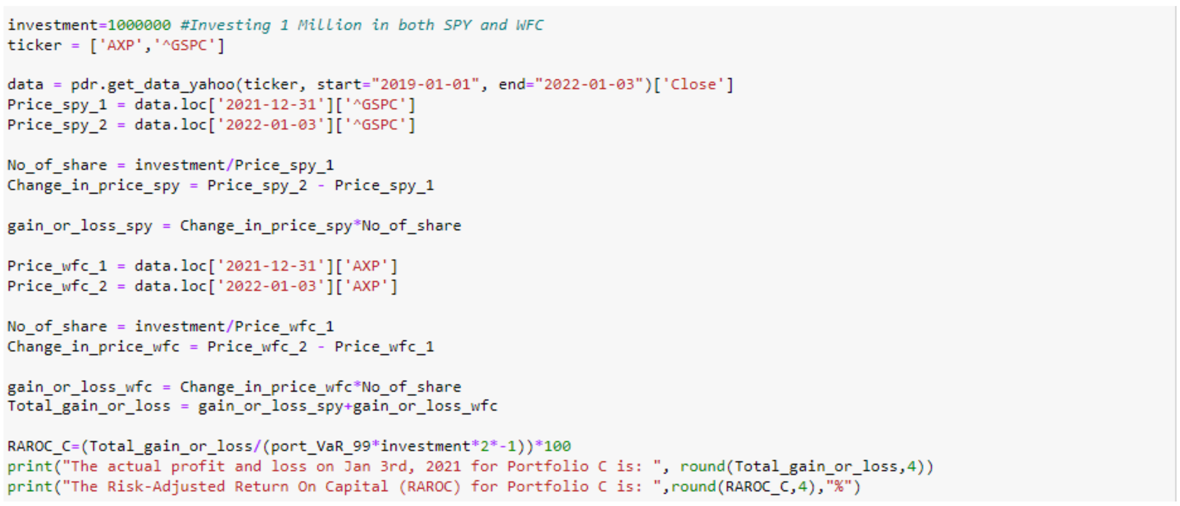
Steps:

- Download the stock price from yahoo finance for the given period.
- Find the change in stock price to understand if there is a loss or gain and multiply it with totalshares to calculate loss at portfolio level.
- Then we calculate RAROC using the following formula

$$\mathbf{RAROC} = \frac{\mathbf{Expected\ Return}}{\mathbf{99\ \%\ VAR}} \times \mathbf{100}$$

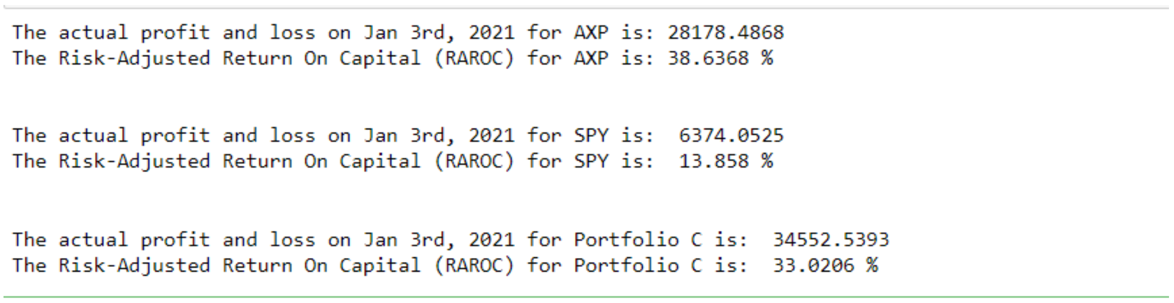


```python
#Solution for 6

#AXP
investment=1000000
ticker = ['AXP']

data = pdr.get_data_yahoo(ticker, start="2019-01-01", end="2022-01-03")['Close']
Price_ups_1 = data.loc['2021-12-31']['AXP']
Price_ups_2 = data.loc['2022-01-03']['AXP']

No_of_share = investment/Price_ups_1

Change_in_price_ups = Price_ups_2 - Price_ups_1

gain_or_loss = Change_in_price_ups*No_of_share

RAROC_A=(gain_or_loss/(AXP_VaR_99*investment*-1))*100
print("The actual profit and loss on Jan 3rd, 2021 for AXP is:", round(gain_or_loss,4))
print("The Risk-Adjusted Return On Capital (RAROC) for AXP is:",round(RAROC_A,4),"%")

print('\n')

#SPY
investment=1000000
ticker = ['^GSPC']

data = pdr.get_data_yahoo(ticker, start="2019-01-01", end="2022-01-03")['Close']
Price_spy_1 = data.loc['2021-12-31']['^GSPC']
Price_spy_2 = data.loc['2022-01-03']['^GSPC']

No_of_share = investment/Price_spy_1

Change_in_price_spy = Price_spy_2 - Price_spy_1

gain_or_loss = Change_in_price_spy*No_of_share

RAROC_B=(gain_or_loss/(SPY_VaR_99*investment*-1))*100
print("The actual profit and loss on Jan 3rd, 2021 for SPY is: ", round(gain_or_loss,4))
print("The Risk-Adjusted Return On Capital (RAROC) for SPY is: ",round(RAROC_B,4),"%")

print('\n')
```



```python
investment=1000000 #Investing 1 Million in both SPY and WFC
ticker = ['AXP','^GSPC']

data = pdr.get_data_yahoo(ticker, start="2019-01-01", end="2022-01-03")['Close']
Price_spy_1 = data.loc['2021-12-31']['^GSPC']
Price_spy_2 = data.loc['2022-01-03']['^GSPC']

No_of_share = investment/Price_spy_1
Change_in_price_spy = Price_spy_2 - Price_spy_1

gain_or_loss_spy = Change_in_price_spy*No_of_share

Price_wfc_1 = data.loc['2021-12-31']['AXP']
Price_wfc_2 = data.loc['2022-01-03']['AXP']

No_of_share = investment/Price_wfc_1
Change_in_price_wfc = Price_wfc_2 - Price_wfc_1

gain_or_loss_wfc = Change_in_price_wfc*No_of_share
Total_gain_or_loss = gain_or_loss_spy+gain_or_loss_wfc

RAROC_C=(Total_gain_or_loss/(port_VaR_99*investment*2*-1))*100
print("The Jan actual profit and loss on Jan 3rd, 2021 for Portfolio C is: ", round(Total_gain_or_loss,4))
print("The Risk-Adjusted Return On Capital (RAROC) for Portfolio C is: ",round(RAROC_C,4),"%")
```

Result:



```
The actual profit and loss on Jan 3rd, 2021 for AXP is: 28178.4868
The Risk-Adjusted Return On Capital (RAROC) for AXP is: 38.6368 %


The actual profit and loss on Jan 3rd, 2021 for SPY is:  6374.0525
The Risk-Adjusted Return On Capital (RAROC) for SPY is:  13.858 %


The actual profit and loss on Jan 3rd, 2021 for Portfolio C is:  34552.5393
The Risk-Adjusted Return On Capital (RAROC) for Portfolio C is:  33.0206 %
```

We can see that the stock only portfolio has the highest RAROC. This is because it has the least 99% one day Var. This means that stock portfolio is the most effective investment as it provides the best balance between risk and reward. Therefore, more resources and capital can be used to invest in stock portfolio.