Project
FIM 549

FINANCIAL RISK MANAGEMENT

SYMBOL – WELLS FARGO & CO. (WFC)

Submitted By: Mehul Mehta

Submitted To: Dr. Wei Chen

Question 1

Preliminary Data Preparation

The data of the stock prices are downloaded from "Yahoo Finance" for Tickers: "SPY" (S&P 500) and "WFC" (Wells Fargo & Co.).

Submitting data and code in good standing – working condition
The Jupyter file is also submitted with the Report.

Question 2

PART 1) Report the GARCH (1,1) model parameters, i.e., omega, alpha and beta along with p-statistics for the selected symbol and 'SPY'.

Defining GARCH Model

The GARCH model, or Generalized Autoregressive Conditionally Heteroscedastic model, was developed by doctoral student Tim Bollerslev in 1986. 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 GARCH(1,1) is defined as

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

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

1. Firstly, extract the Wells Fargo & Co (ticker = WFC) and S&P 500 (ticker = SPY) data pertaining to the required dates by using Yahoo Finance.
2. After extracting the data, calculate the log returns of the close values for Wells Fargo (WFC) and S&P 500(SPY).
3. 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 in vol='GARCH' and assign lag parameters (p,q).

After using the above steps,

Results

GARCH(1,1) for SPY is summarized below.

Parameters	Coefficient	Standard Error	P value
ω	0.0531	2.230e-02	1.723e-02
α	0.2438	4.849e-02	4.961e-07
β	0.7388	3.672e-02	4.635e-90

GARCH(1,1) for WFC is summarized below.

Parameters	Coefficient	Standard Error	P value
ω	0.2458	0.156	0.116
α	0.2238	0.132	8.906e-02
β	0.7349	0.128	9.850e-09

Snapshot of the Summary of GARCH(1,1) for SPY

```
Iteration:      7,   Func. Count:     38,   Neg. LLF: 1070.6765842455266
Optimization terminated successfully    (Exit mode 0)
            Current function value: 1070.676584245651
            Iterations: 7
            Function evaluations: 38
            Gradient evaluations: 7
```

Out[16]:

Zero Mean - GARCH Model Results

Dep. Variable:	log_returns	**R-squared:**	0.000
Mean Model:	Zero Mean	**Adj. R-squared:**	0.001
Vol Model:	GARCH	**Log-Likelihood:**	-1070.68
Distribution:	Normal	**AIC:**	2147.35
Method:	Maximum Likelihood	**BIC:**	2161.23
		No. Observations:	755
Date:	Sun, Apr 11 2021	**Df Residuals:**	755
Time:	19:14:21	**Df Model:**	0

Volatility Model

	coef	std err	t	P>|t|	95.0% Conf. Int.
omega	0.0531	2.230e-02	2.382	1.723e-02	[9.409e-03,9.683e-02]
alpha[1]	0.2438	4.849e-02	5.028	4.961e-07	[0.149, 0.339]
beta[1]	0.7388	3.672e-02	20.123	4.635e-90	[0.667, 0.811]

Covariance estimator: robust

Snapshot of the Summary of GARCH(1,1) for WFC

```
Iteration:      7,   Func. Count:     39,   Neg. LLF: 1529.3299580027308
Optimization terminated successfully    (Exit mode 0)
            Current function value: 1529.32995273475
            Iterations: 9
            Function evaluations: 46
            Gradient evaluations: 9
```

]:

Zero Mean - GARCH Model Results

Dep. Variable:	log_returns	**R-squared:**	0.000
Mean Model:	Zero Mean	**Adj. R-squared:**	0.001
Vol Model:	GARCH	**Log-Likelihood:**	-1529.33
Distribution:	Normal	**AIC:**	3064.66
Method:	Maximum Likelihood	**BIC:**	3078.54
		No. Observations:	755
Date:	Sun, Apr 11 2021	**Df Residuals:**	755
Time:	19:23:58	**Df Model:**	0

Volatility Model

	coef	std err	t	P>|t|	95.0% Conf. Int.
omega	0.2458	0.156	1.574	0.116	[-6.031e-02, 0.552]
alpha[1]	0.2238	0.132	1.700	8.906e-02	[-3.417e-02, 0.482]
beta[1]	0.7349	0.128	5.733	9.850e-09	[0.484, 0.986]

PART 2) Long Term Volatilities of selected symbol (WFC) and 'SPY'

We know that,

$$\boldsymbol{\omega = \gamma V_L}$$

Where V_L is the long-term volatility.

And,

$$\boldsymbol{\gamma + \alpha + \beta = 1}$$

Therefore, using the above equations,

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

Therefore, we use the values of alpha, beta and omega to calculate the long-term volatility.

The long-term volatilities of 'WFC' and 'SPY' are,

Long Term Volatility (VL)

```
#Defining the parameters for SPY from the GARCH model.
alpha_spy=result_spy_garch.params["alpha[1]"]
beta_spy=result_spy_garch.params["beta[1]"]
omega_spy=result_spy_garch.params["omega"]/10**4

#Defining the parameters for WFC from the GARCH model.
alpha_wfc=result_wfc_garch.params["alpha[1]"]
beta_wfc=result_wfc_garch.params["beta[1]"]
omega_wfc=result_wfc_garch.params["omega"]/10**4

#Long term volatility of stock
gamma_wfc = 1-alpha_wfc-beta_wfc
VL_wfc = np.sqrt(omega_wfc/gamma_wfc)
print("The long term volatility of WFC is:", VL_wfc*100,"%")

#Long term volatility of index
gamma_spy = 1-alpha_spy-beta_spy
VL_spy = np.sqrt((omega_spy/gamma_spy))
print("The long term volatility of SPY is:", VL_spy*100,"%")
```

```
The long term volatility of WFC is: 2.437581176997433 %
The long term volatility of SPY is: 1.7471299812907712 %
```

PART 3) On Jan 4th, 2021 what are the volatilities of your symbol and SPY?

To calculate the required volatilities of the stock and the index on Jan 4th, 2021, the following steps were followed:

1. The GARCH(1,1) model is used to forecast the the 1 day volatilities on Jan 4th , 2021. In the below code, horizon =1 means that we have taken 1 day as our input parameters.
2. As the market was closed on January 1st, January 2nd and January 3rd, therefore, we can use the GARCH(1,1) model of both the symbol and the index as on Dec 31st , 2020.

Volatility on 2021-01-04

```
#Volatility of Stock on 2021-01-04 (WFC)
vol_wfc = math.sqrt(result_wfc_garch.forecast(horizon=1).variance.iloc[-1])
print("The Volatility of WFC on 2021-01-04 is:",vol_wfc)

#Volatility of Index on 2021-01-04 (SPY)
vol_spy = math.sqrt(result_spy_garch.forecast(horizon=1).variance.iloc[-1])
print("The Volatility of SPY on 2021-01-04 is:",vol_spy)
```

```
The Volatility of WFC on 2021-01-04 is: 1.764343204541756
The Volatility of SPY on 2021-01-04 is: 0.6357065411000443
```

Question 3

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

Correlation Function

The correlation coefficient can be used to summarize the strength of the linear relationship between two data samples.

Steps followed

1. Python has an inbuilt function called **.corr()** which automatically gives the correlation between two different datasets. In our case, we calculated the correlation between
2. The mathematics behind the function **.corr()** is defined below.

Method to calculate correlation

The Pearson's correlation coefficient is calculated as the covariance of the two variables divided by the product of the standard deviation of each data sample.

The formula for correlation is,

$$\rho = \frac{\mathbf{E(WFC \times SPY) - E(WFC)E(SPY)}}{\mathbf{STD(WFC)STD(SPY)}}$$

Parameter	Results
Correlation	0.734476239266421

Correlation between the returns of WFC and SPY

```
corr_spy_wfc = {'SPY Close':spy['log_returns'], 'WF Close':wfc['log_returns']}
corr_spy_wfc = pd.DataFrame(data=corr_spy_wfc)
corr_spy_wfc.dropna(inplace=True)
print("The correlation between the returns of WFC and SPY is:",corr_spy_wfc.corr()['SPY Close']['WF Close'])
```

```
The correlation between the returns of WFC and SPY is: 0.734476239266421
```

Question 4

PART A

Definition of Value at Risk

Value at risk (VaR) is a statistic that measures and quantifies the level of financial risk within a firm, portfolio or position over a specific time frame. This metric is most commonly used by investment and commercial banks to determine the extent and occurrence ratio of potential losses in their institutional portfolios.

Definition of Expected Shortfall

Conditional Value at Risk (CVaR), also known as the **expected shortfall**, is a risk assessment measure that quantifies the amount of tail risk an investment portfolio has. CVaR is derived by taking a weighted average of the "extreme" losses in the tail of the distribution of possible returns, beyond the value at risk (VaR) cutoff point. Conditional value at risk is used in portfolio optimization for effective risk management.

Historical Simulation

The fundamental assumption of the Historical Simulations methodology is that you base your results on the past performance of your portfolio and make the assumption that the past is a good indicator of the near-future.

Steps followed to calculated One Day 99% VaR and One Day 99% Expected Shortfall.

1. Ticker used in the code is 'WFC' and an investment of $1 million is assumed.
2. The data is extracted using Yahoo Finance pertinent to the mentioned time period.
3. Multiple scenarios were generated i.e. For each scenario, the dollar change in the value of the portfolio between today and tomorrow is calculated. For instance, for scenario 1, the close price of day 1/day 0 is calculated, for scenario 2, the close price of day 2/day 1 is calculated and so on.
4. For each scenario, we calculated the loss by subtracting the values of the portfolio from the investment.
5. The losses are then arranged in the descending order.
6. We then extract the top '1' percent values from the losses. The length of the dataset is 756, therefore, we will take one percent as the 7^{th} value. This value represents the one-day 99% VaR.
7. Similarly, to calculate the one-day 99% ES, the average of top 1 percent losses is calculated.

The above mentioned procedure is used to calculate the VaR and Expected Shortfall for Portfolio A, B and C.

Portfolio	Value at Risk	Expected Shortfall
A	78392.24	117516.68
B	48688.16	73370.95

C	119220.94	187569.89

Diversification Benefits

- Portfolio diversification benefits when a firm or investor invest in various assets whose values do not rise and fall in perfect harmony. Because of this imperfect correlation, the risk of a diversified portfolio is smaller than the weighted average risk of its constituent assets. In term of Value at Risk (VaR), portfolio VaR is smaller than the sum of its constituent VaRs because VaR is a sub additive risk measure.
- Diversification Benefit = VaR(A) + VaR(B) – VaR(C) = 7859.46

Portfolio A - Calculating one day 99% VaR using Historical Simulation

```python
#investment in Portfolio A is 1 Million.
investment=1000000

# Create a portfolio of WFC
ticker = ['WFC']

# Download closing prices
data = pdr.get_data_yahoo(ticker, start="2018-01-01", end="2020-12-31")['Close']
data['returns'] = (data['WFC'])/(data['WFC'].shift(1))

#Generating scenarios
generating_scenarios = data['returns']*data['WFC'][-1]
generating_scenarios.dropna(inplace=True)
generating_scenarios

#A represents the value of portfolio under different scenarios
A = (generating_scenarios*investment)/data['WFC'][-1]

#Calculating the loss
loss_A = (investment-A)
loss_A.sort_values(inplace=True, ascending=False)

#Calculating the 1 Day 99% Value at Risk (VaR)
VaR_99_A = loss_A.head(10)[6]

#Calculating the 1 Day 99% Expected Shortfall (ES)
ES_99_A = np.mean(loss_A[:6])
ES_99_A

print("For Portfolio A")
print("1 Day 99% Value at Risk (VaR) =", VaR_99_A)
print("1 Day 99% Expected Shortfall (ES) =", ES_99_A)
```

```
For Portfolio A
1 Day 99% Value at Risk (VaR) = 78392.24220238079
1 Day 99% Expected Shortfall (ES) = 117516.68188552333
```

Portfolio B - Calculating one day 99% VaR using Historical Simulation

```
#investment in Portfolio B is 1 Million.
investment=1000000

# Create a portfolio of SPY
ticker = ['SPY']

# Download the closing prices
data = pdr.get_data_yahoo(ticker, start="2018-01-01", end="2020-12-31")['Close']
data['returns'] = (data['SPY'])/(data['SPY'].shift(1))

#Generating scenarios
generating_scenarios = data['returns']*data['SPY'][-1]
generating_scenarios.dropna(inplace=True)

#B represents the value of portfolio under different scenarios
B = (generating_scenarios*investment)/data['SPY'][-1]

#Calculating the loss
loss_B = (investment-B)
loss_B.sort_values(inplace=True, ascending=False)

#Calculating the 1 Day 99% Value at Risk (VaR)
VaR_99_B = loss_B.head(10)[6]

#Calculating the 1 Day 99% Expected Shortfall (ES)
ES_99_B = np.mean(loss_B[:6])
ES_99_B

print("For Portfolio B")
print("1 Day 99% Value at Risk (VaR) =", VaR_99_B)
print("1 Day 99% Expected Shortfall (ES) =", ES_99_B)
```

```
For Portfolio B
1 Day 99% Value at Risk (VaR) = 48688.16981634952
1 Day 99% Expected Shortfall (ES) = 73370.94789777206
```

Portfolio C - Calculating one day 99% VaR using Historical Simulation ¶

```
#investment 1 Million each in SPY and WFC.
investment=1000000

# Create a portfolio of a mixture of SPY and WFC
ticker = ['SPY','WFC']

# Download the closing prices
data = pdr.get_data_yahoo(ticker, start="2018-01-01", end="2020-12-31")['Close']
data['Returns_WFC'] = (data['WFC'])/(data['WFC'].shift(1))
data['Returns_SPY'] = (data['SPY'])/(data['SPY'].shift(1))

#Generating scenarios
generating_scenarios = pd.DataFrame(data=data[['Returns_WFC','Returns_SPY']])
generating_scenarios['wfc'] = data['Returns_WFC']*data['WFC'][-1]
generating_scenarios['spy'] = data['Returns_SPY']*data['SPY'][-1]
generating_scenarios.dropna(inplace=True)
generating_scenarios = generating_scenarios[['wfc','spy']]

#C represents the value of portfolio under different scenarios
C = (generating_scenarios['wfc']*investment)/data['WFC'][-1]+(generating_scenarios['spy']*investment)/data['SPY'][-1]

#Calculating the loss
investment=2000000
loss_C = (investment-C)
loss_C.sort_values(inplace=True, ascending=False)

#Calculating the 1 Day 99% Value at Risk (VaR)
VaR_99_C = loss_C.head(10)[6]

#Calculating the 1 Day 99% Expected Shortfall (ES)
ES_99_C = np.mean(loss_C[:6])
ES_99_C

print("For Portfolio C")
print("1 Day 99% Value at Risk (VaR) =", VaR_99_C)
print("1 Day 99% Expected Shortfall (ES) =", ES_99_C)
```

```
For Portfolio C
1 Day 99% Value at Risk (VaR) = 119220.93645872525
1 Day 99% Expected Shortfall (ES) = 187569.89675263126
```

PART B - Using Basel (1996) method back testing the one-day 99% VaR of portfolio (A) (hint: you can either scale to $1 million or just use the unit share price.)

Back Testing

Back testing is used to determine the accuracy of a VaR model. Backtesting involves the comparison of the calculated VaR measure to the actual losses (or gains) achieved on the portfolio. A backtest relies on the level of confidence that is assumed in the calculation.

Backtest

```
ticker=['WFC']
investment=1000000
data = pdr.get_data_yahoo(ticker, start="2018-01-01", end="2020-12-31")['Close']
data_backtest= pdr.get_data_yahoo(ticker, start="2020-03-01", end="2021-02-28")['Close']
no_of_share=investment/data.loc['2020-12-31']['WFC']
change_in_price=[]
for i in range(len(data_backtest)-1):
    change_in_price.append(no_of_share*(data_backtest['WFC'][i+1]-data_backtest['WFC'][i]))
change_in_price

exceed = [ i for i in change_in_price if i > VaR_99_A]
print("The number of times the actual VaR exceeded the calculated VaR: ",len(exceed),"times")
```

```
The number of times the actual VaR exceeded the calculated VaR:  6 times
```

Conclusion

The above results tell us that the actual VaR exceeded the calculate VaR 6 times. Therefore, using only VaR to calculate the loss in a portfolio is not a good strategy. Instead VaR must be used with other quantitative methods like Expected Shortfall to calculate the max loss that might occur in the portfolio.

PART C - Does the backtesting result surprise you? Please discuss why your VaR works/fails to work.

Yes, I was surprised to see that how the concept of VaR failed. The concept of VaR was established to calculate how much a set of investments might lose (with a given probability), given normal market conditions, in a set time period such as a day ,**however,** the concept of VaR failed to its own definition.

The calculation of VaR did not work because the method assumes that the returns are normally distributed for assets and that the portfolios do not have skewness or excess kurtosis. Therefore, using unrealistic return distributions as inputs can lead to underestimating the real risk with VAR. Therefore, I believe Value at Risk should be one little piece in the risk management process and it must be complemented with other tools, especially those taking care of that 1% worst case area, which VAR virtually ignores.

Question 5

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

VAR summarizes the predicted maximum loss (or worst loss) over a target horizon within a given confidence interval.

Steps followed to calculated One Day 99% VaR.

1. Let us assume that the portfolio returns follows a normal distribution.
2. Calculate the standard deviation and the correlation between the stock 'WFC' and the index 'SPY'.
3. After following step2, calculate the standard deviation of the portfolio.
4. Calculate the z-value corresponding to one percentile in normal distribution
5. Multiply the z value with the volatility of portfolio C to obtain the one-day 99% VaR.

Portfolio C - Calculating one day 99% VaR using Normal Distribution

In [45]:

```
# Create our portfolio of equities
investment=1000000
ticker = ['SPY','WFC']
mean=0

# Download closing prices
data = pdr.get_data_yahoo(ticker, start="2018-01-01", end="2020-12-31")['Adj Close']

# Calculating the log return
data['Returns_SPY'] = np.log(data['SPY'])/(data['SPY'].shift(1))
data['Returns_WFC'] = np.log(data['WFC'])/(data['WFC'].shift(1))

# Calculating the standard deviation of the returns of SPY and WFC
std_spy=np.std(data['Returns_SPY'])*investment
std_wfc=np.std(data['Returns_WFC'])*investment

#Calculating Z value using normal distribution.
Z_value = norm.ppf(.99)

#Calculating the correlation
correlation = data.corr()['SPY']['WFC']
correlation

#The standard deviation of the one-day change in the value of the portfolio
std_portfolio = np.sqrt((std_spy**2)+(std_wfc**2)+(2*correlation*std_spy*std_wfc))

#Calculating the VaR
VaR_99 = mean + std_portfolio*Z_value
print("1 Day 99% Value at Risk (VaR) using Normal Distribution is =", VaR_99)
```

```
1 Day 99% Value at Risk (VaR) using Normal Distribution is = 46908.23274609298
```

PART B - 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?

Method to calculate VaR for Portfolio C	VaR Value
Historical Approach	119220.94
Normal Distribution	46908.23

Conclusion:

The VaR from Historical Simulation approach was greater than VaR from Normal distribution approach. The reason that VaR from Normal distribution approach is low because we assume that the asset returns are i.i.d. normally distributed. However, in reality, there is empirical evidence that the financial data is not normally distributed and exhibits properties of skewness or kurtosis.

The presence of skewness and kurtosis has already violated the condition of normality. Further the combination of negative skewness and positive kurtosis indicates that there is a high probability of a large negative return than estimated under normal distribution. So, if we are using data that has heavy tails, and still using the standard distribution assumption for calculating VaR, then our VaR will underestimate the actual risk of the portfolio. Hence, in our case, VaR of normal distribution is lower than from historical approach.

Question 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

Definition of RAROC

- 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

To calculate Risk-adjusted returns on capital (RAROC),

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

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

Using the above steps to calculate the RAROC for Portfolio A, B and C

Portfolio	Gain or Loss	RAROC
A	-15904.56	-20.28%
B	-13613.98	-27.96%
C	-29518.54	-24.75

Therefore, Portfolio B has the best Risk-adjusted returns on capital (RAROC). One of the reason that RAROC for Portfolio B is better than others is that the 99% VaR of Portfolio B is the lowest (when calculated using historical simulation approach).

Risk-Adjusted Return On Capital (RAROC) for Portfolio A

```
investment=1000000
# Create our portfolio of equities
ticker = ['WFC']

# Download closing prices
data = pdr.get_data_yahoo(ticker, start="2018-01-01", end="2021-01-04")['Close']
Price_wfc_1 = data.loc['2020-12-31']['WFC']
Price_wfc_2 = data.loc['2021-01-04']['WFC']
No_of_share = investment/Price_wfc_1
Change_in_price_wfc = Price_wfc_2 - Price_wfc_1
gain_or_loss = Change_in_price_wfc*No_of_share

RAROC_A=(gain_or_loss/VaR_99_A)*100
print("The actual profit and loss on Jan 4th, 2021 for Portfolio A is: ", gain_or_loss)
print("The Risk-Adjusted Return On Capital (RAROC)  for Portfolio A is: ",RAROC_A,"%")
```

```
The actual profit and loss on Jan 4th, 2021 for Portfolio A is:  -15904.557236005381
The Risk-Adjusted Return On Capital (RAROC)  for Portfolio A is:  -20.28843261677028 %
```

Risk-Adjusted Return On Capital (RAROC) for Portfolio B

```
investment=1000000
# Create our portfolio of equities
ticker = ['SPY']

# Download closing prices
data = pdr.get_data_yahoo(ticker, start="2018-01-01", end="2021-01-04")['Close']
Price_spy_1 = data.loc['2020-12-31']['SPY']
Price_spy_2 = data.loc['2021-01-04']['SPY']
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/VaR_99_B)*100
print("The actual profit and loss on Jan 4th, 2021  for Portfolio B is: ", gain_or_loss)
print("The Risk-Adjusted Return On Capital (RAROC)  for Portfolio B is: ",RAROC_B,"%")
```

```
The actual profit and loss on Jan 4th, 2021  for Portfolio B is:  -13613.983822125
The Risk-Adjusted Return On Capital (RAROC)  for Portfolio B is:  -27.96158465901796 %
```

Risk-Adjusted Return On Capital (RAROC) for Portfolio C

```python
investment=1000000 #Investing 1 Million in both SPY and WFC
# Create our portfolio of equities
ticker = ['SPY','WFC']

# Download closing prices
data = pdr.get_data_yahoo(ticker, start="2018-01-01", end="2021-01-04")['Close']
Price_spy_1 = data.loc['2020-12-31']['SPY']
Price_spy_2 = data.loc['2021-01-04']['SPY']

#Calculating number of share of SPY
No_of_share = investment/Price_spy_1
Change_in_price_spy = Price_spy_2 - Price_spy_1

#Calculating if there is a gain or loss
gain_or_loss_spy = Change_in_price_spy*No_of_share

#Calculating change in price
Price_wfc_1 = data.loc['2020-12-31']['WFC']
Price_wfc_2 = data.loc['2021-01-04']['WFC']

#Calculating number of share of WFC
No_of_share = investment/Price_wfc_1
Change_in_price_wfc = Price_wfc_2 - Price_wfc_1

#Calculating if there is a gain or loss
gain_or_loss_wfc = Change_in_price_wfc*No_of_share
Total_gain_or_loss = gain_or_loss_spy+gain_or_loss_wfc

#Calculating RAROC for Portfolio C
RAROC_C=(Total_gain_or_loss/VaR_99_C)*100
print("The actual profit and loss on Jan 4th, 2021  for Portfolio C is: ", Total_gain_or_loss)
print("The Risk-Adjusted Return On Capital (RAROC)  for Portfolio C is: ",RAROC_C,"%")
```

```
The actual profit and loss on Jan 4th, 2021  for Portfolio C is:  -29518.54105813038
The Risk-Adjusted Return On Capital (RAROC)  for Portfolio C is:  -24.759527927672178 %
```