CQF Exam 1

Viranca Balsingh

In partial fulfilment of the Certificate in Quantitative Finance
As offered by Fitch Learning under the supervision of Dr.Paul Wilmott

August 15, 2024

Problem 1

Given the investment universe:

Asset	μ	σ	w
A	0.05	0.07	w_1
B	0.07	0.28	w_2
C	0.15	0.25	w_3
D	0.22	0.31	w_4

$$\text{Corr} = \begin{pmatrix} 1 & 0.4 & 0.3 & 0.3 \\ 0.4 & 1 & 0.27 & 0.42 \\ 0.3 & 0.27 & 1 & 0.5 \\ 0.3 & 0.42 & 0.5 & 1 \end{pmatrix}$$

And given the objective function and constraint:

$$\text{argmin}_w \quad \frac{1}{2} w' \Sigma w \quad \text{s.t.} \quad w'1 = 1$$

The derivation of the analytical solution for optimal allocations w* is found by setting up the Lagrangian for the objective function as shown in Equation 1.1. Where λ is the Lagrange multiplier.

$$L(w, \lambda) = \frac{1}{2} w' \Sigma w - \lambda(w'1 - 1) \tag{1.1}$$

Then by partially differentiating L with respect to w and λ and setting them equal to 0 to find the inflection points of L, we get Equation1.2 and 1.3, respectively.

$$\frac{\partial L}{\partial w} = \Sigma w - \lambda 1 = 0 \implies \Sigma w = \lambda 1 \tag{1.2}$$

$$\frac{\partial L}{\partial \lambda} = -(w'1 - 1) = 0 \implies w'1 = 1 \tag{1.3}$$

Solving Equation 1.2 for w gives Equation 1.4.

$$w = \lambda \Sigma^{-1} 1 \tag{1.4}$$

Substituting that into Equation 1.3 gives Equation 1.5.

$$(\lambda \Sigma^{-1} 1)' 1 = 1 \implies \lambda(1' \Sigma^{-1} 1) = 1 \implies \lambda = \frac{1'}{\Sigma^{-1} 1} \tag{1.5}$$

Now plugging Equation 1.5 into Equation 1.4 we finally get the analytical solution for w^* as shown in 1.6.

$$w^* = \lambda \Sigma^{-1} 1 = \frac{\Sigma^{-1} 1}{1' \Sigma^{-1} 1} \tag{1.6}$$

b)

The snippet of code below first defines all the variables as given in the problem statement. Then calculates the covariance matrix and its inverse. Finally, the portfolio weights are calculated by using the formula analytically derived in part a.

```
import numpy as np

# Given data
mu = np.array([0.05, 0.07, 0.15, 0.22])
sigma = np.array([0.07, 0.28, 0.25, 0.31])
corr = np.array([
    [1, 0.4, 0.3, 0.3],
    [0.4, 1, 0.27, 0.42],
    [0.3, 0.27, 1, 0.5],
    [0.3, 0.42, 0.5, 1]
])

# Calculate covariance matrix
cov_matrix = np.outer(sigma, sigma) * corr

# Calculate inverse of the covariance matrix
cov_matrix_inv = np.linalg.inv(cov_matrix)

# Calculate the weights for the Global Minimum Variance Portfolio
ones = np.ones(len(mu))
w_star = cov_matrix_inv @ ones / (ones.T @ cov_matrix_inv @ ones)

print(w_star)
```

The final weights are numerically computed to be:

$$w^* = \begin{pmatrix} w_1 \\ w_2 \\ w_3 \\ w_4 \end{pmatrix} = \begin{pmatrix} 1.0431 \\ -0.0413 \\ 0.0060 \\ -0.0078 \end{pmatrix}$$

Problem 2

a)
Using the same investment universe as in Problem 1. But subject to the following objective function, budget constraint and target return, respectively:

$$\arg\min_{w} \frac{1}{2} w'\Sigma w \quad \text{subject to} \quad w'1 = 1, \quad w'\mu = m$$

Once again setting up the lagrangian now using both λ and γ:

$$L(w, \lambda, \gamma) = \frac{1}{2} w'\Sigma w - \lambda(w'1 - 1) - \gamma(w'\mu - m)$$

Then taking the partial derivative w.r.t. w, λ and γ and setting to zero gives the following equations:

$$\frac{\partial L}{\partial w} = \Sigma w - \lambda 1 - \gamma \mu = 0 \implies \Sigma w = \lambda 1 + \gamma \mu \tag{2.1}$$

$$\frac{\partial L}{\partial \lambda} = -(w'1 - 1) = 0 \implies w'1 = 1 \tag{2.2}$$

$$\frac{\partial L}{\partial \gamma} = -(w'\mu - m) = 0 \implies w'\mu = m \tag{2.3}$$

Solving Equation 2.1 for w leads to Equation 2.4.

$$w = \lambda \Sigma^{-1} 1 + \gamma \Sigma^{-1} \mu \tag{2.4}$$

Substituting Equation 2.4 into Equation 2.2 and 2.3 leads to the following system of equations:

$$\begin{cases} \lambda(1'\Sigma^{-1}1) + \gamma(\mu'\Sigma^{-1}1) = 1 \\ \lambda(\mu'\Sigma^{-1}1) + \gamma(\mu'\Sigma^{-1}\mu) = m \end{cases} \tag{2.5}$$

Which when solved for λ and γ gives:

$$\lambda = \frac{(\mu'\Sigma^{-1}\mu)(1'\Sigma^{-1}1) - (\mu'\Sigma^{-1}1)^2}{m(\mu'\Sigma^{-1}\mu) - (\mu'\Sigma^{-1}1)} \tag{2.6}$$

$$\gamma = \frac{(\mu'\Sigma^{-1}\mu)(1'\Sigma^{-1}1) - (\mu'\Sigma^{-1}1)^2}{(\mu'\Sigma^{-1}1) - m(1'\Sigma^{-1}1)} \tag{2.7}$$

Finally substituting Equation 2.6 and 2.7 into Equation 2.4 results in Equation 2.8.

$$w^* = \lambda\Sigma^{-1}1 + \gamma\Sigma^{-1}\mu \tag{2.8}$$

Using the stress multipliers 1x, 1.3x, 1.8x on the correlation matrix results in the following matrices, respectively:

$$\begin{bmatrix} 1 & 0.4 & 0.3 & 0.3 \\ 0.4 & 1 & 0.27 & 0.42 \\ 0.3 & 0.27 & 1 & 0.5 \\ 0.3 & 0.42 & 0.5 & 1 \end{bmatrix} \tag{2.9}$$

$$\begin{bmatrix} 1 & 0.52 & 0.39 & 0.39 \\ 0.52 & 1 & 0.351 & 0.546 \\ 0.39 & 0.351 & 1 & 0.65 \\ 0.39 & 0.546 & 0.65 & 1 \end{bmatrix} \tag{2.10}$$

$$\begin{bmatrix} 1 & 0.72 & 0.54 & 0.54 \\ 0.72 & 1 & 0.486 & 0.756 \\ 0.54 & 0.486 & 1 & 0.9 \\ 0.54 & 0.756 & 0.9 & 1 \end{bmatrix} \tag{2.11}$$

Now using 2.12 and performing the matrix calculation in python as follows:

$$\sigma_\Pi = \sqrt{\mathbf{w}'\Sigma\mathbf{w}} \tag{2.12}$$

```python
import numpy as np

# Define the original correlation matrix and stress factors
corr_matrix = np.array([
    [1, 0.4, 0.3, 0.3],
    [0.4, 1, 0.27, 0.42],
    [0.3, 0.27, 1, 0.5],
    [0.3, 0.42, 0.5, 1]
])

stress_factors = [1, 1.3, 1.8]
```

```
# Function to stress correlation matrix
def stress_corr_matrix(corr_matrix, factor):
    stressed_matrix = np.copy(corr_matrix)
    for i in range(stressed_matrix.shape[0]):
        for j in range(stressed_matrix.shape[1]):
            if i != j:
                stressed_matrix[i, j] = min(stressed_matrix[i, j] * factor, 0.99)
    return stressed_matrix

# Compute stressed correlation matrices
stressed_matrices = [stress_corr_matrix(corr_matrix, factor) for factor in
    stress_factors]

# Define the mean returns and target return m
mu = np.array([0.05, 0.07, 0.15, 0.22])
m = 0.07

# Compute the necessary terms for the Lagrange multipliers
ones = np.ones(len(mu))

# Compute the weights and portfolio risk for each stressed correlation matrix
def compute_portfolio_weights_and_risk(corr_matrix, mu, m):
    sigma = np.array([0.07, 0.28, 0.25, 0.31])
    cov_matrix = np.outer(sigma, sigma) * corr_matrix
    cov_matrix_inv = np.linalg.inv(cov_matrix)

    A = ones.T @ cov_matrix_inv @ ones
    B = ones.T @ cov_matrix_inv @ mu
    C = mu.T @ cov_matrix_inv @ mu

    lambda_ = (m * C - B) / (A * C - B**2)
    gamma = (A - m * B) / (A * C - B**2)

    w_star = lambda_ * (cov_matrix_inv @ ones) + gamma * (cov_matrix_inv @ mu)
    portfolio_risk = np.sqrt(w_star.T @ cov_matrix @ w_star)

    return w_star, portfolio_risk

# Calculate for each stressed correlation matrix
results = [compute_portfolio_weights_and_risk(matrix, mu, m) for matrix in
    stressed_matrices]

print(results)
```

This gives for Matrix 2.9 with stress factor 1:

Optimal Weights:

$$\mathbf{w}^* = \begin{pmatrix} -5.429 \\ -1.463 \\ 2.254 \\ 4.708 \end{pmatrix}$$

Portfolio Risk:

$$\sigma_\Pi = 1.618$$

This gives for Matrix 2.10 with stress factor 1.3:

Optimal Weights:

$$\mathbf{w}^* = \begin{pmatrix} -4.181 \\ -2.358 \\ 1.141 \\ 5.468 \end{pmatrix}$$

Portfolio Risk:

$$\sigma_\Pi = 1.581$$

This gives for Matrix 2.11 with stress factor 1.8:

Optimal Weights:

$$\mathbf{w}^* = \begin{pmatrix} 5.142 \\ -7.374 \\ -10.751 \\ 13.053 \end{pmatrix}$$

Portfolio Risk:

$$\sigma_\Pi = 0.796$$

Problem 3

a)

The sharpe ratio is given by:

$$S_C = \frac{\mu_C - R}{\sigma_C} \tag{3.1}$$

Where:

S_C is the Sharpe Ratio for investment C.
μ_C is the expected return of the investment C.
R is the risk-free rate.
σ_C is the standard deviation of the excess return of the investment C.

When considering different time periods, the standard deviation σ scales with the square root of time. For example, for an annual standard deviation, the daily standard deviation can be calculated as:

$$\sigma_{\text{daily}} = \frac{\sigma_{\text{annual}}}{\sqrt{252}} \tag{3.2}$$

Similarly, the monthly and quarterly standard deviations can be calculated as:

$$\sigma_{\text{monthly}} = \frac{\sigma_{\text{annual}}}{\sqrt{12}} \tag{3.3}$$

$$\sigma_{\text{quarterly}} = \frac{\sigma_{\text{annual}}}{\sqrt{4}} \tag{3.4}$$

Given the annualized Sharpe Ratio $SR_{\text{annual}} = 0.53$:
For daily Sharpe Ratio:

$$SR_{\text{daily}} = \frac{0.53}{\sqrt{252}} \approx 0.0334$$

For monthly Sharpe Ratio:

$$SR_{\text{monthly}} = \frac{0.53}{\sqrt{12}} \approx 0.1530$$

For quarterly Sharpe Ratio:

$$SR_{\text{quarterly}} = \frac{0.53}{\sqrt{4}} \approx 0.2650$$

The probability of loss can be found using the cumulative distribution function (CDF) of the standard normal distribution. Specifically, we want:

$$\Pr(P\&L < 0) = \Phi(-SR) \tag{3.5}$$

where Φ is the CDF of the standard normal distribution.

Plugging in the numbers and utilizing a normal distribution we get the following: For daily:

$$\Pr(P\&L_{\text{daily}} < 0) = \Phi(-0.0334) \approx 0.4867 \text{ or } 48.67\%$$

For monthly:

$$\Pr(P\&L_{\text{monthly}} < 0) = \Phi(-0.1530) \approx 0.4395 \text{ or } 43.95\%$$

For quarterly:

$$\Pr(P\&L_{\text{quarterly}} < 0) = \Phi(-0.2650) \approx 0.3959 \text{ or } 39.59\%$$

For annual:

$$\Pr(P\&L_{\text{annual}} < 0) = \Phi(-0.53) \approx 0.2981 \text{ or } 29.81\%$$

From this is seems that "Evaluating the P&L more frequently make it appear more risky than it actually is."

Problem 4

With the Nasdaq100 and sp500 data loaded into dataframes in Pandas, a couple columns will be added to each dataframe to help with the calculations. First of all, the daily returns are added using built *pct_change*() in Pandas functions. Then, the rolling standard deviation with a 21 day window is added by using *.rolling(window=21).std()* on the returns. Next, using the given formula, the analytical 10-day VaR is added. Additionally, another column is added with the returns shifted by 10 days, which represents the 10-day forward return. Lastly, another column is added which simply is True if the 10-day forward return is smaller than the analytical 10-day VaR, and False otherwise. This last column represents the breach condition for that given date.

```
import pandas as pd
import numpy as np

# Load the data from the CSV files
nasdaq_file_path = 'nasdaq100_2024.csv'
sp500_file_path = 'sp500_2024.csv'

nasdaq_data = pd.read_csv(nasdaq_file_path)
sp500_data = pd.read_csv(sp500_file_path)

# Convert the Date column to datetime
nasdaq_data['Date'] = pd.to_datetime(nasdaq_data['Date'])
sp500_data['Date'] = pd.to_datetime(sp500_data['Date'])

# Calculate the daily returns
nasdaq_data['Return'] = nasdaq_data['Closing Price'].pct_change()
sp500_data['Return'] = sp500_data['Closing Price'].pct_change()

# Calculate the rolling standard deviation (21 days)
nasdaq_data['RollingStdDev'] = nasdaq_data['Return'].rolling(window=21).std()
sp500_data['RollingStdDev'] = sp500_data['Return'].rolling(window=21).std()

# Calculate the Analytical VaR for a 10-day period using the given formula
factor = 2.33 # Factor for 99% confidence level (z-score)
nasdaq_data['VaR_10D'] = factor * nasdaq_data['RollingStdDev'] * np.sqrt(10)
sp500_data['VaR_10D'] = factor * sp500_data['RollingStdDev'] * np.sqrt(10)

# Calculate the 10-day forward return
nasdaq_data['Ret_10D'] = nasdaq_data['Closing Price'].shift(-10) /
   nasdaq_data['Closing Price'] - 1
sp500_data['Ret_10D'] = sp500_data['Closing Price'].shift(-10) /
   sp500_data['Closing Price'] - 1
```

```
# Identify breaches (when Ret_10D < VaR_10D)
nasdaq_data['Breach'] = nasdaq_data['Ret_10D'] < -nasdaq_data['VaR_10D']
sp500_data['Breach'] = sp500_data['Ret_10D'] < -sp500_data['VaR_10D']

# Drop rows with NaN values resulting from shifting operations
nasdaq_data_cleaned = nasdaq_data.dropna()
sp500_data_cleaned = sp500_data.dropna()
```

a)

By counting the breach column and comparing it to the overall size of the analytical VaR, 38 breaches are found for the Nasdaq 100, which equates to a breach percentage of 2.31%. Moreover, 55 breaches are found for the SP500, which equates to a breach percentage of 3.34%

```
# Count and percentage of breaches
breach_count = nasdaq_data_cleaned['Breach'].sum()
total_periods = len(nasdaq_data_cleaned)
breach_percentage = (breach_count / total_periods) * 100
```

b)

The breaches are marked by red crosses on the ticker chart of the nasdaq 100 in figure 4.1:

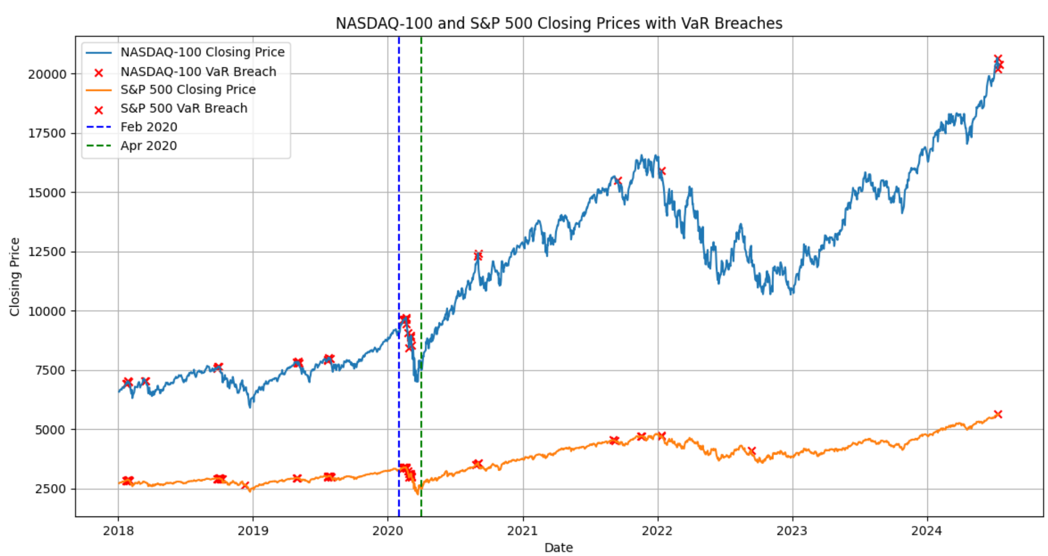


Figure 4.1: NASDAQ-100 Closing Prices with VaR Breaches.

c)

A complete list of the breaches for the Nasdaq 100 are listed in Table 4.1. And the breaches for the SP500 are listed in 4.2 and 4.3.

d)

Looking more closely at the breaches during February and April of 2020, as in Figure 4.2, it becomes apparent that there were substantially more breaches in this period for the SP500 than there were for the Nasdaq. In more detail, looking at the full list of breaches in the subsequent Tables, we see that the Nasdaq had twelve breaches compared to seventeen for the SP500. Based on that I would say that the SP500 has shown to be more risky.

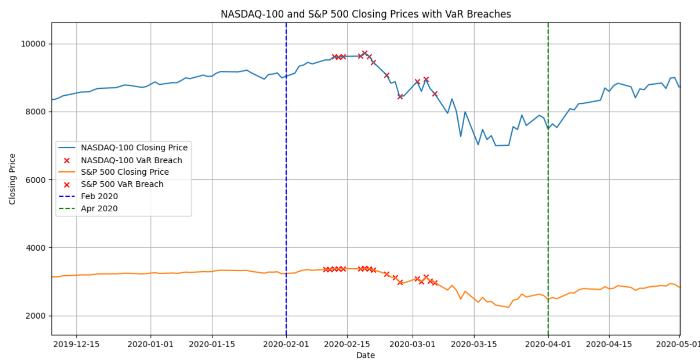


Figure 4.2: NASDAQ-100 Closing Prices with VaR Breaches.

The subsequent market correction period, depicted in 4.3, again shows more VaR breaches for the SP500 rendering it more risky again. An important dicussion note to add here is that this analysis did not consider the depth of the breaches, which could pose an alternative perspective on the riskyness of the breaches.

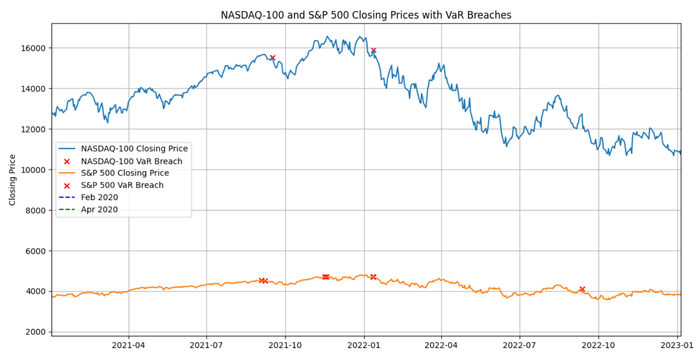


Figure 4.3: NASDAQ-100 Closing Prices with VaR Breaches.

Date	ClosingPrice	Return	VaR 10D	Ret_10D
2018-01-22	6906.279785	0.010528	0.046126	-0.059418
2018-01-24	6919.350098	-0.006334	0.048624	-0.048752
2018-01-25	6916.299805	-0.000441	0.048462	-0.088226
2018-01-26	7022.970215	0.015423	0.050159	-0.086899
2018-01-29	6988.319824	-0.004934	0.051942	-0.066464
2018-03-14	7040.979980	-0.000785	0.077467	-0.082399
2018-03-16	7019.950195	-0.001567	0.073911	-0.089617
2018-09-26	7563.089844	-0.000013	0.053075	-0.068569
2018-09-27	7629.569824	0.008790	0.054881	-0.087232
2018-09-28	7627.649902	-0.000252	0.051344	-0.061676
2018-10-01	7645.450195	0.002334	0.051367	-0.075441
2019-04-29	7839.040039	0.001579	0.032851	-0.065685
2019-04-30	7781.459961	-0.007345	0.035990	-0.048780
2019-05-03	7845.729980	0.015752	0.039940	-0.043597
2019-05-06	7794.089844	-0.006582	0.042052	-0.053552
2019-07-22	7905.120117	0.008962	0.051390	-0.061913
2019-07-23	7954.560059	0.006254	0.051903	-0.054464
2019-07-24	8010.609863	0.007046	0.052538	-0.057263
2019-07-29	7989.080078	-0.003476	0.049473	-0.053498
2020-02-12	9613.200195	0.010017	0.075407	-0.122387
2020-02-13	9595.700195	-0.001820	0.074925	-0.118164
2020-02-14	9623.580078	0.002905	0.074789	-0.077476
2020-02-18	9629.799805	0.000646	0.073969	-0.107511
2020-02-19	9718.730469	0.009235	0.074621	-0.079172
2020-02-20	9627.830078	-0.009353	0.076974	-0.099313
2020-02-21	9446.690430	-0.018814	0.084188	-0.097002
2020-02-24	9079.629883	-0.038856	0.106053	-0.124631
2020-02-27	8436.669922	-0.049256	0.131240	-0.139038
2020-03-02	8877.980469	0.049180	0.156055	-0.209237
2020-03-04	8949.280273	0.041281	0.174664	-0.198239
2020-03-06	8530.339844	-0.016297	0.177289	-0.180069
2020-09-01	12292.860352	0.015041	0.078231	-0.085030
2020-09-02	12420.540039	0.010386	0.078642	-0.107853
2021-09-16	15515.910156	0.000799	0.047696	-0.053254
2022-01-12	15905.099609	0.003849	0.106531	-0.119584
2024-07-10	20675.380859	0.010872	0.050068	-0.079466
2024-07-11	20211.359375	-0.022443	0.065747	-0.068317
2024-07-15	20386.880859	0.002724	0.063116	-0.065110

Table 4.1: VaR breaches for the Nasdaq 100

Date	ClosingPrice	Return	VaR 10D	Ret_10D
2018-01-22	2832.969971	0.008067	0.031085	-0.064960
2018-01-23	2839.129883	0.002174	0.030548	-0.050716
2018-01-24	2837.540039	-0.000560	0.030997	-0.054935
2018-01-25	2839.250000	0.000603	0.030754	-0.090957
2018-01-26	2872.870117	0.011841	0.033357	-0.088177
2018-01-29	2853.530029	-0.006732	0.036907	-0.069223
2018-01-30	2822.429932	-0.010899	0.043115	-0.056508
2018-01-31	2823.810059	0.000489	0.041353	-0.044330
2018-09-26	2905.969971	-0.003289	0.027279	-0.041394
2018-09-27	2914.000000	0.002763	0.027599	-0.063703
2018-09-28	2913.979980	-0.000007	0.026036	-0.050395
2018-10-01	2924.590088	0.003641	0.025533	-0.059427
2018-10-02	2923.429932	-0.000397	0.025561	-0.038828
2018-10-03	2925.510010	0.000712	0.025336	-0.039754
2018-10-04	2901.610107	-0.008169	0.028494	-0.045778
2018-10-05	2885.570068	-0.005528	0.029334	-0.040820
2018-10-08	2884.429932	-0.000395	0.029084	-0.044567
2018-10-09	2880.340088	-0.001418	0.029053	-0.048484
2018-12-10	2637.719971	0.001762	0.100148	-0.108662
2019-04-29	2943.030029	0.001072	0.029677	-0.044566
2019-04-30	2945.830078	0.000951	0.028676	-0.037823
2019-07-22	2985.030029	0.002829	0.035079	-0.046998
2019-07-23	3005.469971	0.006847	0.036370	-0.041158
2019-07-24	3019.560059	0.004688	0.036577	-0.044901
2019-07-26	3025.860107	0.007388	0.034747	-0.035431
2019-07-29	3020.969971	-0.001616	0.034991	-0.045770
2019-07-31	2980.379883	-0.010886	0.038042	-0.046900
2020-02-10	3352.090088	0.007326	0.061210	-0.066788
2020-02-11	3357.750000	0.001688	0.060838	-0.071882
2020-02-12	3379.449951	0.006463	0.060715	-0.118567
2020-02-13	3373.939941	-0.001630	0.060730	-0.124401
2020-02-14	3380.159912	0.001844	0.060730	-0.085774
2020-02-18	3370.290039	-0.002920	0.059886	-0.108869

Table 4.2: VaR breaches for the SP500 part one

Date	ClosingPrice	Return	VaR 10D	Ret_10D
2020-02-19	3386.149902	0.004706	0.060018	-0.075611
2020-02-20	3373.229980	-0.003816	0.060231	-0.103548
2020-02-21	3337.750000	-0.010518	0.062916	-0.109469
2020-02-24	3225.889893	-0.033514	0.083054	-0.148588
2020-02-26	3116.389893	-0.003779	0.091929	-0.120335
2020-02-27	2978.760010	-0.044163	0.112038	-0.167224
2020-03-02	3090.229980	0.046039	0.138381	-0.227847
2020-03-03	3003.370117	-0.028108	0.142391	-0.157883
2020-03-04	3130.120117	0.042203	0.159433	-0.233863
2020-03-05	3023.939941	-0.033922	0.164963	-0.203228
2020-03-06	2972.370117	-0.017054	0.164193	-0.224552
2020-08-27	3484.550049	0.001673	0.038616	-0.041205
2020-09-01	3526.649902	0.007525	0.037766	-0.040027
2020-09-02	3580.840088	0.015366	0.042497	-0.062508
2021-09-03	4535.430176	-0.000335	0.035291	-0.039180
2021-09-07	4520.029785	-0.003396	0.036017	-0.036690
2021-11-16	4700.899902	0.003865	0.031399	-0.039963
2021-11-18	4704.540039	0.003385	0.031009	-0.035308
2022-01-11	4713.069824	0.009160	0.071960	-0.077049
2022-01-12	4726.350098	0.002818	0.070364	-0.084598
2022-09-12	4110.410156	0.010584	0.096345	-0.110785
2024-07-10	5633.910156	0.010208	0.027885	-0.036703

Table 4.3: VaR breaches for the SP500 part two

Problem 5

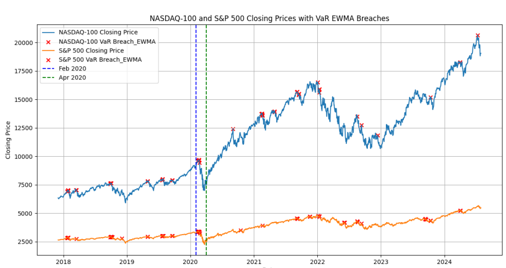
a)

Starting with the same derived dataframes as in Chapter 4, I will add the EWMA σ^2_{t+1} as an additional column, based on Equation 5.1. Where $\lambda = 0.72$ and the variance at t0 is equal to the variance of the overall variance. Then recalculating the forward 10-day return as before and finally identifying breaches. This is shown in the following code:

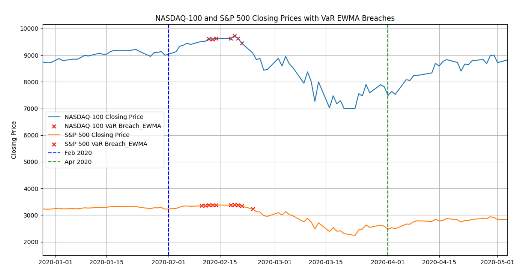
$$\sigma^2_{t+1|t} = \lambda \sigma^2_{t|t-1} + (1 - \lambda) r^2 \tag{5.1}$$

The resulting EWMA breach count is 50 for the Nasdaq, 3.05% of the total and 62 for the SP500, 3.78% of the total. The EWMA are plotted on the ticker in Figure 5.1a and 5.1b. A full list of breaches is presented in Table 5.1, 5.2, 5.3 and 5.4.

Regarding the effect of λ on the smoothness of EWMA-predicted volatility: The closer λ is to 1, the more weight the model assigns to past volatility as compared to more recent volatility. This makes the volatlity smoother as sudden fluctuations are not as heavily adjusted for as when λ is closer to 0.



(a) NASDAQ-100 & SP500 Closing Prices with EWMA VaR Breaches.



(b) NASDAQ-100 & SP500 Closing Prices with EWMA VaR Breaches.

Figure 5.1: Side-by-side comparison of NASDAQ-100 & SP500 Closing Prices with EWMA VaR Breaches.

```
# Initialize variables for EWMA calculation
lambda_value = 0.72
nasdaq_data['EWMA_Variance'] = 0.0
sp500_data['EWMA_Variance'] = 0.0
initial_variance_nasdaq = nasdaq_data['Return'].var()
initial_variance_sp500 = sp500_data['Return'].var()

# Set the first value of EWMA variance to the overall variance
nasdaq_data.loc[0, 'EWMA_Variance'] = initial_variance_nasdaq
sp500_data.loc[0, 'EWMA_Variance'] = initial_variance_sp500

# Calculate EWMA variance for the entire dataset
for t in range(1, len(nasdaq_data)):
    nasdaq_data.loc[t, 'EWMA_Variance'] = (lambda_value * nasdaq_data.loc[t-1,
        'EWMA_Variance'] +
                                           (1 - lambda_value) * nasdaq_data.loc[t,
                                               'Return']**2)

for t in range(1, len(sp500_data)):
    sp500_data.loc[t, 'EWMA_Variance'] = (lambda_value * sp500_data.loc[t-1,
        'EWMA_Variance'] +
                                          (1 - lambda_value) * sp500_data.loc[t,
                                              'Return']**2)

# Calculate the 10-day VaR using EWMA variance
nasdaq_data['VaR_10D_EWMA'] = z_score * np.sqrt(nasdaq_data['EWMA_Variance']) *
    np.sqrt(10)
sp500_data['VaR_10D_EWMA'] = z_score * np.sqrt(sp500_data['EWMA_Variance']) *
    np.sqrt(10)

# Recompute forward 10-day return for consistency
nasdaq_data['Forward_10D_Return'] = nasdaq_data['Closing Price'].shift(-10) /
    nasdaq_data['Closing Price'] - 1
sp500_data['Forward_10D_Return'] = sp500_data['Closing Price'].shift(-10) /
    sp500_data['Closing Price'] - 1

# Identify breaches where forward 10-day return < VaR (EWMA)
nasdaq_data['Breach_EWMA'] = nasdaq_data['Forward_10D_Return'] <
    -nasdaq_data['VaR_10D_EWMA']
sp500_data['Breach_EWMA'] = sp500_data['Forward_10D_Return'] <
    -sp500_data['VaR_10D_EWMA']
```

Date	ClosingPrice	Return	VaR 10D	Ret_10D
2018-01-22	6906.279785	0.010528	0.046126	-0.059418
2018-01-25	6916.299805	-0.000441	0.048462	-0.088226
2018-01-26	7022.970215	0.015423	0.050159	-0.086899
2018-01-29	6988.319824	-0.004934	0.051942	-0.066464
2018-03-14	7040.979980	-0.000785	0.077467	-0.082399
2018-03-15	7030.970215	-0.001422	0.077880	-0.063980
2018-03-16	7019.950195	-0.001567	0.073911	-0.089617
2018-09-26	7563.089844	-0.000013	0.053075	-0.068569
2018-09-27	7629.569824	0.008790	0.054881	-0.087232
2018-09-28	7627.649902	-0.000252	0.051344	-0.061676
2018-10-01	7645.450195	0.002334	0.051367	-0.075441
2018-10-02	7628.279785	-0.002246	0.051426	-0.046124
2018-10-03	7637.430176	0.001200	0.050998	-0.046979
2019-04-29	7839.040039	0.001579	0.032851	-0.065685
2019-04-30	7781.459961	-0.007345	0.035990	-0.048780
2019-07-22	7905.120117	0.008962	0.051390	-0.061913
2019-07-23	7954.560059	0.006254	0.051903	-0.054464
2019-07-24	8010.609863	0.007046	0.052538	-0.057263
2019-09-18	7888.560059	-0.000029	0.080707	-0.042818
2019-09-19	7901.790039	0.001677	0.079512	-0.033334
2020-02-12	9613.200195	0.010017	0.075407	-0.122387
2020-02-13	9595.700195	-0.001820	0.074925	-0.118164
2020-02-14	9623.580078	0.002905	0.074789	-0.077476
2020-02-18	9629.799805	0.000646	0.073969	-0.107511
2020-02-19	9718.730469	0.009235	0.074621	-0.079172
2020-02-20	9627.830078	-0.009353	0.076974	-0.099313
2020-02-21	9446.690430	-0.018814	0.084188	-0.097002
2020-09-02	12420.540039	0.010386	0.078642	-0.107853
2021-02-11	13734.349609	0.005791	0.093556	-0.060062
2021-02-16	13773.769531	-0.002457	0.092777	-0.051825

Table 5.1: Summary of Financial Data

Date	ClosingPrice	Return	VaR 10D	Ret_10D
2021-02-17	13699.709961	-0.005377	0.092221	-0.074190
2021-02-18	13637.509766	-0.004540	0.090817	-0.086050
2021-02-19	13580.780273	-0.004160	0.084281	-0.067174
2021-04-28	13901.620117	-0.004202	0.075817	-0.064740
2021-04-29	13970.200195	0.004933	0.074398	-0.061635
2021-09-03	15652.860352	0.003115	0.048052	-0.040930
2021-09-07	15675.759766	0.001463	0.046886	-0.041337
2021-09-14	15382.900391	-0.003343	0.049459	-0.039823
2021-09-15	15503.530273	0.007842	0.050726	-0.048417
2021-09-16	15515.910156	0.000799	0.047696	-0.053254
2022-01-03	16501.769531	0.011133	0.110714	-0.078235
2022-01-10	15614.429688	0.001426	0.107623	-0.093843
2022-01-11	15844.120117	0.014710	0.108858	-0.105488
2022-01-12	15905.099609	0.003849	0.106531	-0.119584
2022-08-18	13505.990234	0.002608	0.119738	-0.091172
2022-09-12	12739.719727	0.012029	0.119031	-0.116612
2022-12-13	11834.209961	0.010914	0.111160	-0.097587
2023-10-12	15184.099609	-0.003741	0.078896	-0.070767
2024-04-01	18293.199219	0.002110	0.070057	-0.032054
2024-07-10	20675.380859	0.010872	0.050068	-0.079466

Table 5.2: Summary of Financial Data

Date	ClosingPrice	Return	VaR 10D	Ret_10D
2018-01-22	2832.969971	0.008067	0.031085	-0.064960
2018-01-23	2839.129883	0.002174	0.030548	-0.050716
2018-01-24	2837.540039	-0.000560	0.030997	-0.054935
2018-01-25	2839.250000	0.000603	0.030754	-0.090957
2018-01-26	2872.870117	0.011841	0.033357	-0.088177
2018-01-29	2853.530029	-0.006732	0.036907	-0.069223
2018-03-16	2752.010010	0.001703	0.065974	-0.061820
2018-09-26	2905.969971	-0.003289	0.027279	-0.041394
2018-09-27	2914.000000	0.002763	0.027599	-0.063703
2018-09-28	2913.979980	-0.000007	0.026036	-0.050395
2018-10-01	2924.590088	0.003641	0.025533	-0.059427
2018-10-02	2923.429932	-0.000397	0.025561	-0.038828
2018-10-03	2925.510010	0.000712	0.025336	-0.039754
2018-10-04	2901.610107	-0.008169	0.028494	-0.045778
2018-10-05	2885.570068	-0.005528	0.029334	-0.040820
2018-10-08	2884.429932	-0.000395	0.029084	-0.044567
2018-10-09	2880.340088	-0.001418	0.029053	-0.048484
2018-12-03	2790.370117	0.010941	0.087607	-0.087519
2019-04-29	2943.030029	0.001072	0.029677	-0.044566
2019-04-30	2945.830078	0.000951	0.028676	-0.037823
2019-07-22	2985.030029	0.002829	0.035079	-0.046998
2019-07-23	3005.469971	0.006847	0.036370	-0.041158
2019-07-24	3019.560059	0.004688	0.036577	-0.044901
2019-07-29	3020.969971	-0.001616	0.034991	-0.045770
2019-09-18	3006.729980	0.000343	0.064081	-0.039618
2019-09-19	3006.790039	0.000020	0.062202	-0.031981
2020-02-10	3352.090088	0.007326	0.061210	-0.066788
2020-02-11	3357.750000	0.001688	0.060838	-0.071882
2020-02-12	3379.449951	0.006463	0.060715	-0.118567
2020-02-13	3373.939941	-0.001630	0.060730	-0.124401
2020-02-14	3380.159912	0.001844	0.060730	-0.085774
2020-02-18	3370.290039	-0.002920	0.059886	-0.108869
2020-02-19	3386.149902	0.004706	0.060018	-0.075611
2020-02-20	3373.229980	-0.003816	0.060231	-0.103548
2020-02-21	3337.750000	-0.010518	0.062916	-0.109469
2020-02-24	3225.889893	-0.033514	0.083054	-0.148588
2020-10-16	3483.810059	0.000135	0.088939	-0.061384
2021-02-18	3913.969971	-0.004416	0.073435	-0.037175
2021-09-03	4535.430176	-0.000335	0.035291	-0.039180
2021-09-07	4520.029785	-0.003396	0.036017	-0.036690

Table 5.3: Summary of Financial Data

Date	ClosingPrice	Return	VaR 10D	Ret_10D
2021-09-08	4514.069824	-0.001319	0.036075	-0.026236
2021-11-16	4700.899902	0.003865	0.031399	-0.039963
2021-11-18	4704.540039	0.003385	0.031009	-0.035308
2022-01-04	4793.540039	-0.000630	0.073662	-0.054402
2022-01-10	4670.290039	-0.001441	0.072096	-0.067199
2022-01-11	4713.069824	0.009160	0.071960	-0.077049
2022-01-12	4726.350098	0.002818	0.070364	-0.084598
2022-06-02	4176.819824	0.018431	0.148273	-0.122114
2022-06-07	4160.680176	0.009523	0.130417	-0.096328
2022-08-16	4305.200195	0.001876	0.083631	-0.074106
2022-08-17	4274.040039	-0.007238	0.077291	-0.074646
2022-08-18	4283.740234	0.002270	0.077266	-0.073975
2022-08-19	4228.479980	-0.012900	0.080998	-0.071945
2022-09-12	4110.410156	0.010584	0.096345	-0.110785
2023-09-12	4461.899902	-0.005696	0.054960	-0.042217
2023-09-13	4467.439941	0.001242	0.054152	-0.043186
2023-09-14	4505.100098	0.008430	0.052292	-0.045593
2023-09-19	4443.950195	-0.002151	0.053156	-0.048268
2023-10-12	4349.609863	-0.006246	0.060535	-0.048827
2023-10-13	4327.779785	-0.005019	0.058533	-0.048618
2024-04-01	5243.770020	-0.002014	0.045002	-0.034698
2024-04-03	5211.490234	0.001091	0.045474	-0.036320

Table 5.4: Summary of Financial Data