SMU
SINGAPORE MANAGEMENT UNIVERSITY

QF609 - Risk Analysis Final Project

VaR Portfolio Result Analysis

Group 4 - G1 MQF 2024

Our Team

Hong Yang
01502068

Merlyvia Mualim
01458745

Maureen Calista Chandra
01498515

Nethania Abigail
01520514

Nandini Agarwal
01518145

Phoo Pyae Hsu Myat
01520414

Wenyu Chang
01505913

Yohanes Alfredo Phoa
01520735

Table of Contents

Value at Risk (VaR)

Provides the maximum potential losses of a portfolio position which helps institutions or individuals to estimate the amount of cash needed to cover those risks within the specified amount of time.

Any realized positions that exceeds the predefined limit should be followed by adjustments on the portfolio position. This measure also applicable to any type of assets.

What are VaR calculation approaches ?

- VaR parameters
- Portfolio identification
- Risk factors of portfolio identification
- Joint distribution of risk factors and portfolio P&L

Data Analysis and Parameter Definition

1

1. Defining the VaR Components

- ***Time horizon***: 1-day, as of 30 Oct 2023
- ***Confidence level***: 95% ($\alpha = 0.05$)
- ***Reporting currency***: in $ amount

2. Identify the Objective Portfolio

Stocks (4)

AAPL | BAC | MSFT | F

Investment amount: **$1 mn / each**

Swap (1)

Notional $100 mn | 10Y Maturity | Cash flow 4.2% | Annual freq. payment

3. Identify the Risk Factors of Portfolio

the **variable inputs** to the pricing and the risk engines that influence the value of daily P&L and the portfolio investment risks...

Stocks

- **Prices** of the stock portfolio
- Daily returns of each stocks → using **relative change**
- Daily changes have **multivariate normal** distribution

SOFR Swap Payer

- **30 zero rates (PV01)*** of the SOFR curve
- Daily changes on zero rates → using **absolute change**
- Zero rates as the **discount factor** for price derivation and **forward rate** calculation

** 30 zero rates based on the standardized tenor of 1D, 1M, 2M, 3M, 6M, 9M, 1Y, 2Y, 3Y, 4Y, 5Y, 6Y, 7Y, 8Y, 9Y, 10Y, 11Y, 12Y, 13Y, 14Y, 15Y, 16Y, 17Y, 18Y, 19Y, 20Y, 25Y, 30Y, 35Y, 40Y*

4. Modelling the Joint Distribution of The Daily Risk Factor Changes

Issues to be addressed?

Few dates have **NaN values** for either stock price or the SOFR zero rate curve

Date	AAPL	MSFT	F	BAC	1D	1M	2M
2022-11-11	0.019085	0.016854	0.022316	0.007316	NaN	NaN	NaN
2023-04-07	NaN	NaN	NaN	NaN	0.000596	-0.000189	-0.001362
2023-06-19	NaN	NaN	NaN	NaN	0.000035	0.000088	0.000109
2023-10-09	0.008416	0.007792	0.005816	0.009164	NaN	NaN	NaN

What we do:

```
stocks_daily_return = stocks.pct_change().rename(columns={'AAPL Adj Close':'AAPL daily return',
                        'MSFT Adj Close':'MSFT daily return',
                        'F Adj Close':'F daily return',
                        'BAC Adj Close':'BAC daily return'}).dropna()
```

```
sofr_dailychange = sofr.diff().dropna()
```

dropna() total sample size = 250 data points

preserve only the available information without any estimation that might lead to bias

Calculate the mean and covariance matrix of multivariate normal distribution

Mean of daily changes

	Mean Return
AAPL	0.000408
MSFT	0.001498
F	-0.000855
BAC	-0.001243
1D	0.000056
1M	0.000057
2M	0.000058
3M	0.000052
6M	0.000035
9M	0.000026
1Y	0.000020
2Y	0.000008
3Y	0.000007
4Y	0.000009
5Y	0.000012
6Y	0.000014
7Y	0.000016

	Mean Return
8Y	0.000018
9Y	0.000019
10Y	0.000020
11Y	0.000021
12Y	0.000022
13Y	0.000022
14Y	0.000023
15Y	0.000023
16Y	0.000023
17Y	0.000024
18Y	0.000024
19Y	0.000025
20Y	0.000026
25Y	0.000028
30Y	0.000029
35Y	0.000030
40Y	0.000029

Covariance matrix of daily changes for all 34 risk factors

	AAPL	MSFT	F	BAC	1D	1M	2M	3M	6M	9M	...	15Y
AAPL	2.482670e-04	1.791690e-04	1.310566e-04	7.367619e-05	4.910597e-06	1.591467e-06	-4.590581e-07	-3.733748e-07	1.564113e-08	-1.864894e-07	...	-1.421053e-06
MSFT	1.791690e-04	3.137999e-04	1.321883e-04	6.866378e-05	4.919860e-06	1.745901e-06	-6.470670e-07	-7.024751e-07	-4.622727e-07	-8.154872e-07	...	-9.539776e-07
F	1.310566e-04	1.321883e-04	5.155809e-04	1.690935e-04	5.203595e-06	2.272617e-06	2.552793e-07	3.558528e-07	1.299235e-06	1.598028e-06	...	-1.316482e-06
BAC	7.367619e-05	6.866378e-05	1.690935e-04	2.682169e-04	4.080173e-06	1.865740e-06	5.137577e-07	1.119301e-06	3.085988e-06	4.006378e-06	...	9.070452e-07
1D	4.910597e-06	4.919860e-06	5.203595e-06	4.080173e-06	1.935126e-06	7.077145e-07	-1.187314e-07	-9.803333e-08	2.669587e-08	-4.624695e-08	...	-8.310360e-08
1M	1.591467e-06	1.745901e-06	2.272617e-06	1.865740e-06	7.077145e-07	3.561458e-07	1.514501e-08	-1.041441e-08	3.943683e-08	2.436693e-08	...	-1.491814e-08
2M	-4.590581e-07	-6.470670e-07	2.552793e-07	5.137577e-07	-1.187314e-07	1.514501e-08	1.102684e-07	8.931059e-08	8.496510e-08	1.142346e-07	...	5.177159e-08
3M	-3.733748e-07	-7.024751e-07	3.558528e-07	1.119301e-06	-9.803333e-08	-1.041441e-08	8.931059e-08	1.031211e-07	1.332033e-07	1.717226e-07	...	6.964595e-08
6M	1.564113e-08	-4.622727e-07	1.299235e-06	3.085988e-06	2.669587e-08	3.943683e-08	8.496510e-08	1.332033e-07	2.757106e-07	3.626435e-07	...	1.486499e-07
9M	-1.864894e-07	-8.154872e-07	1.598028e-06	4.006378e-06	-4.624695e-08	2.436693e-08	1.142346e-07	1.717226e-07	3.626435e-07	4.911163e-07	...	2.225188e-07
1Y	-3.032567e-07	-1.007544e-06	1.761577e-06	4.519336e-06	-6.285184e-08	2.020087e-08	1.223410e-07	1.871690e-07	4.053589e-07	5.568052e-07	...	2.755049e-07
2Y	-8.918564e-07	-1.453380e-06	1.368668e-06	4.529985e-06	-1.182200e-07	-5.891156e-09	1.167244e-07	1.812831e-07	4.061249e-07	5.789833e-07	...	4.033670e-07
3Y	-1.164694e-06	-1.495000e-06	8.057722e-07	3.705452e-06	-1.278113e-07	-1.778939e-08	9.819566e-08	1.530226e-07	3.473365e-07	5.045931e-07	...	4.411766e-07

Since our risk horizon is **1-day VaR** → build the joint distribution of **1-day** risk factor changes.

Joint distribution of risk factor values at end of VaR period = distribution of the portfolio P&L over the risk horizon

5. Modelling the Distribution of 1-day Portfolio P&L

Given the **portfolio value P&L over h-period...**

There are **m** underlying risk factors which **k** denotes the order of risk factors

$$L = P_h - P_0 = \underbrace{\sum_{k=1}^{m} \boxed{a^{(k)}} \cdot \Delta x_{0,h}^{(k)}}_{}$$

i.i.d *and* ***additive*** *over an h-day period and* ***linear function*** *of risk factor changes*

$a^{(k)}$ is the **sensitivity factor** or risk of the portfolio value in relative to the k-th risk factor

Hence, the weight factor might be different based on the asset types

Stocks

Sensitivity factors: Notional of stocks position

Reason: direct dollar exposure to the market price changes

Weight of 4 stocks: *np.array ([1000000 , 1000000, 1000000, 1000000*

SOFR Swap

Sensitivity factors: Partial PV01 of the SOFR Swap relative to the corresponding zero rates

Reason:

- PV01 tells the change in the price of the swap for a basis point (0.01%) change in the yield → swap price volatility
- Partial PV01 measures the effect of change/shocks in specific maturities of the yield curve

Steps to derive:

1. Calculate the **swap market value**
   → net present value of the fixed and floating cash flow payment

2. Calculate the **partial PV01**
   → new swap market value after shifting one particular tenor of zero rate at one time by 0.01% or 1 bps ; daily basis

② Parametric VaR Method Calculation

Parametric VaR

This model requires assumption that the distribution of the assets returns to be normally distributed. Known also as variance-covariance method.

Parametric will calculate VaR based on the standard deviation of the historical changes in the assets’ value movements and using the probability within the specified confidence level to find the value.

What are the benefits of Parametric VaR calculation approaches ?

- Ease of implementation and wide applicability
- Computationally efficient

The challenges ?

- High sensitivity of the parameter definition
- Inability to handle extreme or fat tail events

Calculate the Stocks and Swap Mean and Standard Deviation

Since parametric VaR assumes that the portfolio P&L over the risk horizon is **linear function** of the risk factor changes over the period...
We obtained the portfolio distribution based on the **mean and variance of the risk factors changes**

STOCKS

```
stocks_mean = stocks_daily_return.mean(axis = 0)
stocks_std = stocks_daily_return.std(axis = 0)
stocks_cov = stocks_daily_return.cov().to_numpy()

# calculate new prices for parametric VaR
Para_stock_mean = (stocks_mean * stocks_notional).sum()
Para_stock_std = np.sqrt(np.matmul(np.matmul(stocks_notional, stocks_cov), stocks_notional.T))
```

Mean
- Multiply the **mean return** of each stock by its corresponding **notional** amount (weights)
- Add products to calculate the overall expected return of the stock portfolio.

Standard Deviation
- Multiply the **stock notional vector** with **covariance matrix** → weighted covariance of the stocks
- Multiply with **transposed stock notional** → portfolio variance as a weighted sum of covariances
- Square root the result to convert variance into **standard deviation** → **volatility**

Stock Mean = $ 536.21

Stock Std = $ 53,662.35

SWAP

```
sofr_mean = sofr_delta.mean(axis = 0).to_numpy()
sofr_std = sofr_delta.std(axis = 0).to_numpy()
sofr_cov = (sofr_delta).cov(numeric_only = False).to_numpy()

# calculate new swap prices
Para_swap_mean = (pv01 * sofr_mean).sum()
Para_swap_std = np.sqrt(np.matmul(np.matmul(pv01, sofr_cov), pv01.T))
```

Mean
- Multiply **PV01** of each swap with corresponding **SOFR mean return**
- Sum it to aggregate the products → **expected price change** of swap portfolio

Standard Deviation
- Multiply the **PV01 by covariance matrix** → weighted covariance related to swap
- Multiply with **transposed PV01**, applying the weights again → total variance of swap portfolio
- Square root the result to convert variance into **standard deviation** → **volatility**

Swap Mean = $ 20,927.25

Swap Std = $ 579,882.90

Calculate 1-Day 95% Parametric VaR of Stock and Swap

The **threshold loss** value of the stock/swap with **5% probability** that the stock/swap loss will **EXCEED** this amount over a 1 day period

```
Para_stock_VaR = abs(stat.norm.ppf(0.05, loc=Para_stock_mean, scale=Para_stock_std))
Para_swap_VaR = abs(stat.norm.ppf(0.05, loc=Para_swap_mean, scale=Para_swap_std))
```

abs is required to ensure that the result is **positive**

a **Stock/Swap Expected Return (Mean)**
Ensures calculation is based on the stock/swap expected return, rather than assuming a standard normal distribution ($\mu = 0$)

b **Stock/Swap Risk (Standard Deviation)**
Standard deviation (σ) determines the volatility of the distribution.

Take the square root of variance to know the stock/swap standard deviation and ensures the probability calculations are based on the actual risk of the stock/swap (rather than assuming ($\sigma = 1$)

Parametric Stock of 1- day 95% VaR
$ 87,730.50

Parametric Swap of 1- day 95% VaR
$ 932,895.24

Calculate 1-Day 95% Parametric VaR of Portfolio

The **threshold loss** value of the stock/swap with **5% probability** that the portfolio's loss will **EXCEED** this amount over a 1 day period

```
Para_port_mean = Para_swap_mean + Para_stock_mean
Para_port_std = np.sqrt(Para_stock_std ** 2 + Para_swap_std ** 2)

Para_port_VaR = abs(stat.norm.ppf(0.05, loc=Para_port_mean, scale=Para_port_std))
```

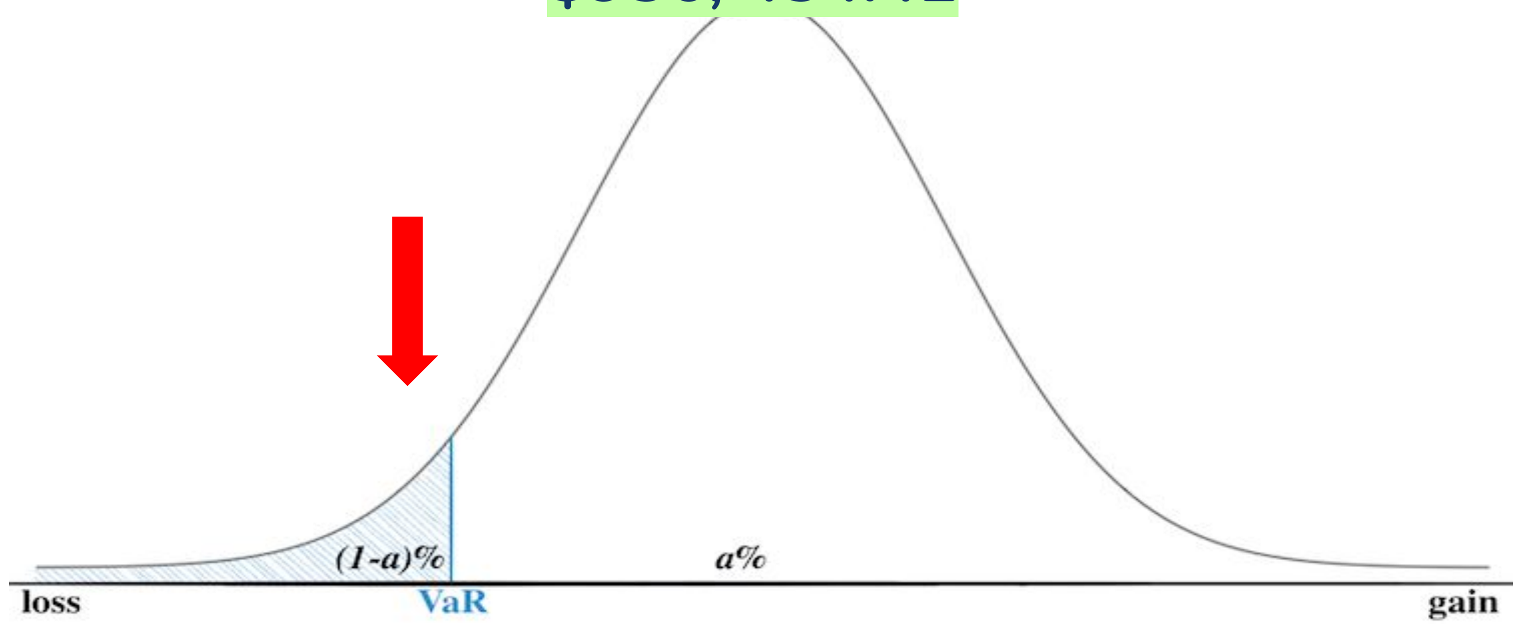
abs is required to ensure that the result is **positive**

ⓐ **Portfolio's expected return (mean)**
Ensures calculation is based on the portfolio's expected return, rather than assuming a standard normal distribution ($\mu = 0$)

ⓑ **Portfolio's risk (standard deviation)**
Standard deviation (σ) determines the volatility of the distribution.

Take the square root of variance to know the portfolio's standard deviation and ensures the probability calculations are based on the actual risk of the portfolio (rather than assuming ($\sigma = 1$)

Parametric Portfolio 1- day 95% VaR:
$936, 434.42



Monte Carlo VaR

This model estimates the potential loss of a financial portfolio by simulating a large number of possible future scenarios.

It uses statistical techniques to model the joint distribution of risk factors (e.g., stock prices, interest rates, FX rates) and calculates the loss distribution of a portfolio over a defined time horizon at a given confidence level.

What are the benefits of Monte Carlo VaR calculation approaches ?

- Captures non-linearity (useful for derivatives).
- No restrictive assumptions about return distributions.

The challenges ?

- Computationally expensive.
- Requires a large number of simulations for accuracy.

Monte Carlo VaR Method Calculation

3

Pre-Simulation Processing for Full Revaluation

Steps to be done:

1. Generate Quasi-Random Samples

$$U_i \sim \mathcal{U}(0,1)$$

2. Convert to Standard Normal Variables

$$Z_i = \Phi^{-1}(U_i), \quad Z_i \sim \mathcal{N}(0, I)$$

3. Convert Correlation (ρ) to Covariance

$$\Sigma = D \cdot \rho \cdot D$$

4. Apply NumPy's Multivariate Normal

$$\Delta X = \mathrm{MVN}(\mu, \Sigma)$$

```
def simulate_correlated(dimension, power, corr_matrix, mean, std):
    """Generates correlated normal samples using NumPy's multivariate normal."""

    simulation = qmc.Sobol(d=dimension, scramble=False).random_base2(m=power)[1:]

    normal_samples = norm.ppf(simulation)

    cov_matrix = np.outer(std, std) * corr_matrix

    correlated_samples = np.random.multivariate_normal(mean, cov_matrix, size=normal_samples.shape[0])

    return correlated_samples
```

- **Dimension** = the number of risk factors
- **Power** = Adjustment factor to the distribution of simulated returns (e.g., for skewness and kurtosis control)

Given the simulation parameter...

Simulated STOCKS return ➡ $R_{\text{stocks}} = \mathrm{MVN}(\mu_{\text{stocks}}, \Sigma_{\text{stocks}})$

```
stock_simulated_returns = simulate_correlated(
    dimension=4, power=20,
    corr_matrix=stocks_daily_return.corr().to_numpy(),
    mean=stocks_mean, std=stocks_std
)
```

Simulated SOFR RATE changes ➡ $\Delta r_{\text{SOFR}} = \mathrm{MVN}(\mu_{\text{SOFR}}, \Sigma_{\text{SOFR}})$

```
sofr_simulated = simulate_correlated(
    dimension=len(sofr_delta.columns), power=20,
    corr_matrix=sofr_delta.corr(numeric_only=False).to_numpy(),
    mean=sofr_mean, std=sofr_std
)
```

Pre-Simulation Processing for Full Revaluation

Why we choose Multivariate not Cholesky decomposition?

1. Multivariate Normal (MVN) correctly applies both correlation and covariance.

- Uses the full **covariance matrix**: where D is the diagonal matrix of standard deviations.

$$\Sigma = D \cdot \rho \cdot D$$

- Ensures the **correct scaling of risk factor changes**.

2. Cholesky decomposition only applies correlation, not full covariance.

- Factorizes the **correlation matrix**, but does not inherently scale by standard deviation.

$$\Sigma = LL^T$$

- Can lead to **overestimated or inconsistent risk factor changes** if standard deviations are not explicitly applied.

Why was the Cholesky result 100K higher ?

- **Overestimated risk factor sensitivities** due to missing standard deviation scaling.
- **Correlation matrix alone does not represent full risk impact**—it needs to be combined with volatilities.

Full Revaluation Approach

STOCKS

Monte Carlo Full Revaluation Stock 1- day 95% VaR: **$87,735.51**

```
# apply latest stock price to the risk factor (stocks_daily_return) and get new stock prices
MC_FR_prices = latest_stock_price * (1 + stock_simulated_returns)

# multiply our shares and the new prices to get new portfolio
MC_FR_stocks_portfolio = share_holdings * MC_FR_prices
MC_FR_stocks_portfolio_changes = (MC_FR_stocks_portfolio - stocks_notional).sum(axis = 1)
```

a. Using the **simulated asset return distribution** to recalculate the new asset price based on the position **as of 30 Oct 2023**.

b. Calculate the **changes of new simulated portfolio** values with respect to the initial investment in stocks. The values using the shares owned (notional / price).

SWAP

Monte Carlo Full Revaluation Swap 1- day 95% VaR: **$941,314.0**

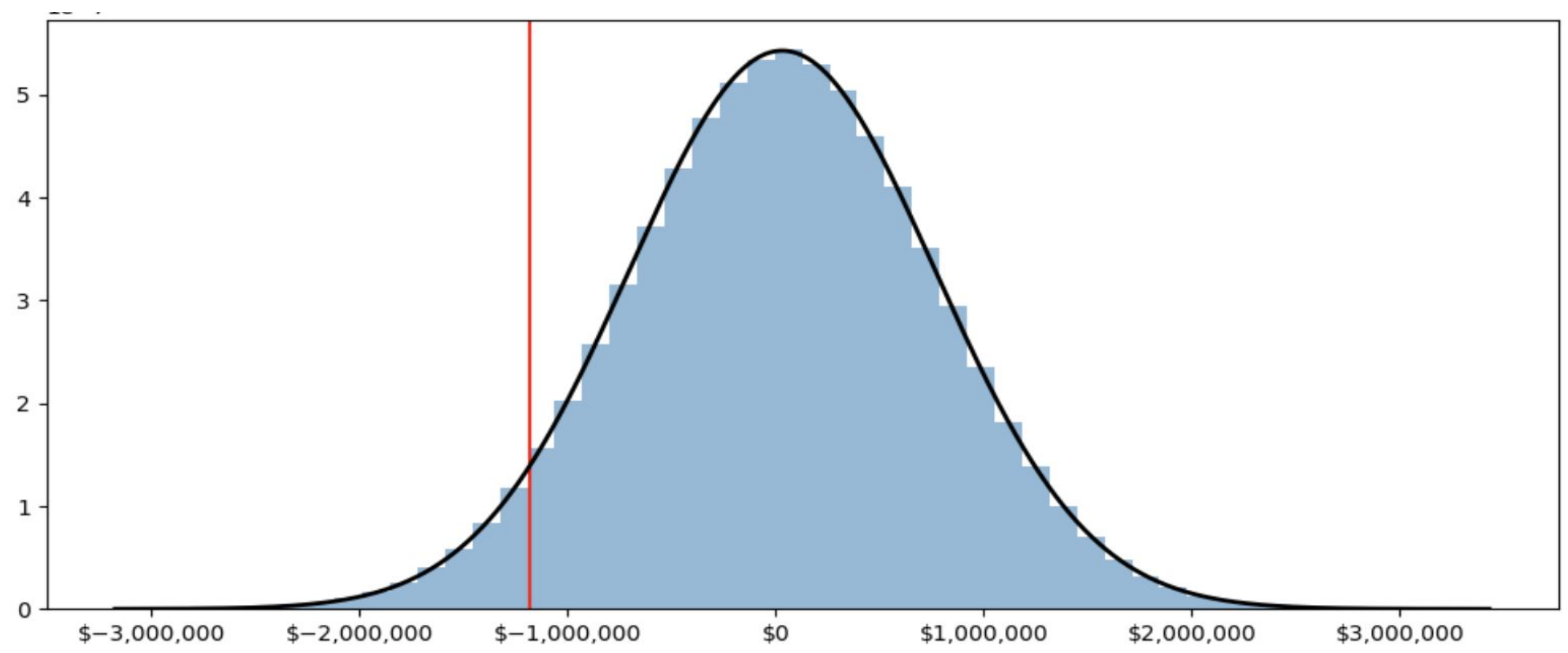
```
# apply historical rate changes to today's rates
new_MC_sofr_10 = pd.DataFrame(latest_sofr_rate + sofr_simulated)
new_MC_sofr_10_disc = compute_discount_factors(new_MC_sofr_10).to_numpy()

n = len(sofr_simulated)
MC_FR_payer = np.zeros(n)

# compute payer swap value for each risk factor
for i in range(n):
    MC_FR_payer[i] = PV_payer_swap(new_MC_sofr_10_disc[i])

# calculate change in value of swap
MC_FR_payer = MC_FR_payer - base_pv    # base_pv: swap initial portfolio
```

a. Recalculate the zero rate curve using the **simulated changes** in the zero rate and multiplied to the rate **as of 30 Oct 2023.**

b. Obtain the entire new swap portfolio value based on the **new SOFR zero rate curve** and calculate the changes of the swap value with respect to the initial value.

PORTFOLIO

```
# combine together stock and swap
MC_FR_portfolio_changes = MC_FR_stocks_portfolio_changes + MC_FR_payer

MC_FR_VaR = abs(np.percentile(MC_FR_portfolio_changes, VaR_percentile))

print(f"Monte Carlo Full Revaluation Portfolio 95% VaR: ${MC_FR_VaR:,.2f}")
plot_distribution(MC_FR_portfolio_changes, MC_FR_VaR, 'Historical VaR Full Revaluation Portfolio')
```



Monte Carlo Full Revaluation Portfolio 1-day 95% VaR: $944,164.63

Sensitivity Based Approach

What we do:

$$\mathrm{PnL}_i = \boxed{\sum_k a_k} \cdot \boxed{\Delta X_{i,k}}$$

a Sensitivity (exposure) to risk factor k

b Simulated risk factor change for scenario i and factor k

STOCKS

Monte Carlo Sensitivity- Based Stock 1- day 95% VaR: **$87,735.51**

```
MC_SB_stocks_changes = (stock_simulated_returns * stocks_notional).sum(axis=1)
```

Estimate the asset movement changes based on the **sensitivity factor** of the portfolio w.r.t to the **simulated changes in asset prices**

SWAP

Monte Carlo Sensitivity- Based Swap 1- day 95% VaR: **$935,853.06**

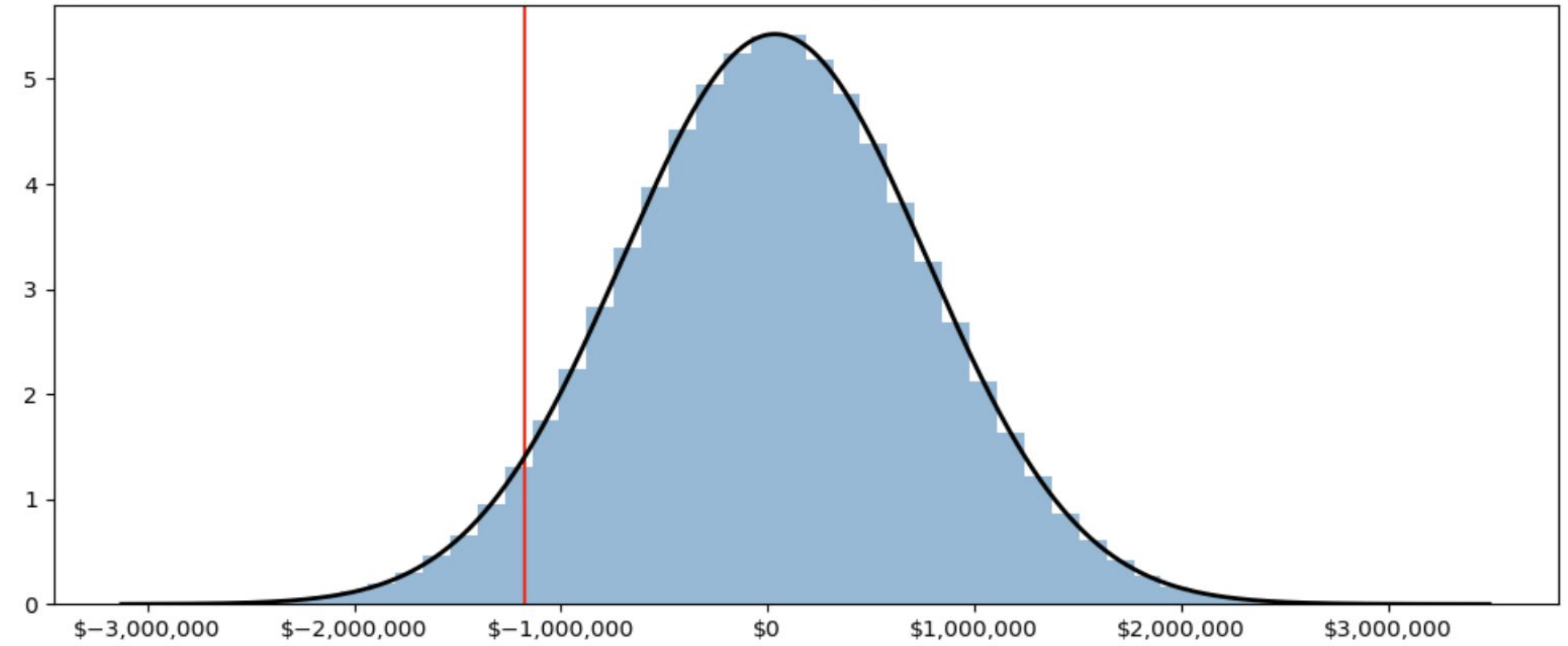
```
MC_SB_payer = (pv01 * sofr_simulated).sum(axis = 1).astype(float)
```

the change in value per basis point move

Estimate the new swap value based on the **sensitivity factor** of the swap value w.r.t to the **simulated changes in zero rate curves**

PORTFOLIO

```
# combine together stock and swap
MC_SB_portfolio_changes = MC_SB_stocks_changes + MC_SB_payer

MC_SB_VaR = abs(np.percentile(MC_SB_portfolio_changes, VaR_percentile))

print(f"Monte Carlo Sensitivity Based Portfolio 95% VaR: ${MC_SB_VaR:,.2f}")
plot_distribution(MC_SB_portfolio_changes, MC_SB_VaR, 'Monte Carlo VaR Sensitivity Based Portfolio')
```



Monte Carlo Sensitivity-Based Portfolio 1- day 95% VaR: **$938,834.54**

4

Historical VaR Method Calculation

Historical VaR

Historical VaR estimates risk using past real-world return joint distributions. Similar to Monte Carlo VaR method, this model can be performed using the risk-based (for linear products) and full revaluation (for exotic products) approach.

It assumes that past market behavior can be a good predictor for the future market behavior.

What are the benefits of Historical VaR calculation approaches ?

- Simple and model-free (no normality assumption)
- Captures actual market behavior (fat tails, skewness)

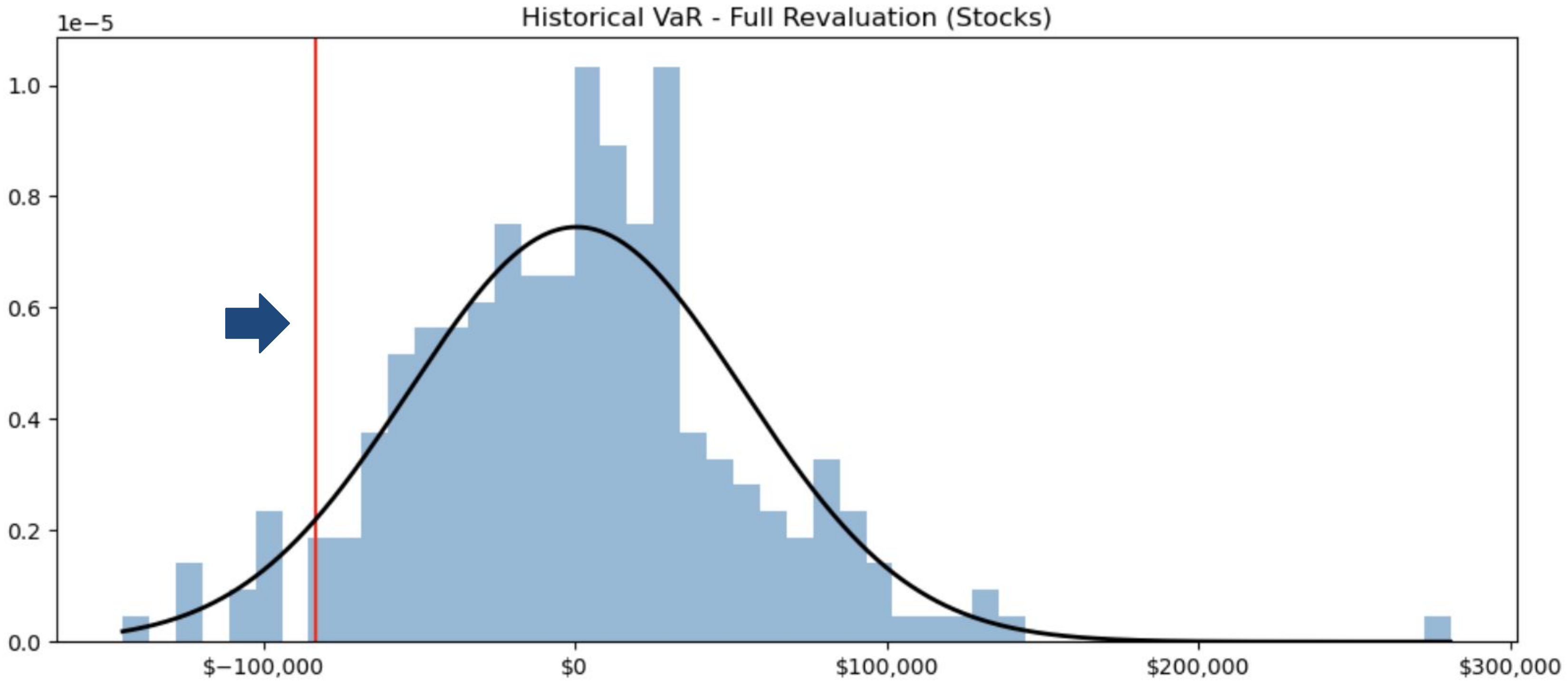
The challenges ?

- Relies on past data (not adaptive to new risks)
- Can be unstable if history doesn't capture extreme events

Full Revaluation Approach

STOCKS

Historical Full Revaluation Stock 1- day 95% VaR: **$ 82,934.71**



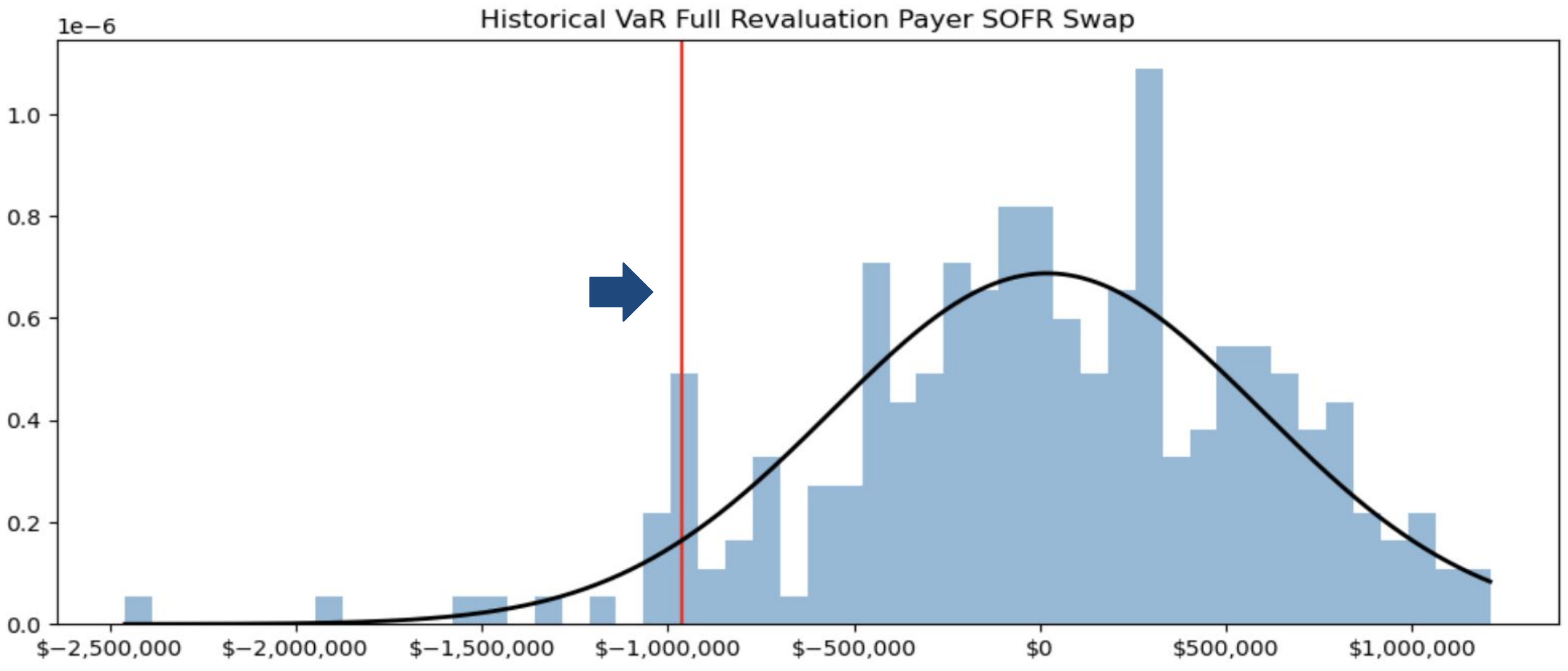

```
# apply latest stock price to the risk factor (stocks_daily_return) and get new stock prices
hist_FR_prices = latest_stock_price * (1 + stocks_daily_return)

# multiply our shares and the new prices to get new portfolio
hist_FR_stocks_portfolio = share_holdings * hist_FR_prices
hist_FR_stocks_portfolio_changes = (hist_FR_stocks_portfolio - stocks_notional).sum(axis = 1)

hist_FR_stocks_VaR = abs(np.percentile(hist_FR_stocks_portfolio_changes, VaR_percentile))
```

a. Using the **historical asset return distribution** to recalculate the new asset price based on the position as of 30 Oct 2023.

b. Multiply with no of shares owned each stocks (e.g., notional / latest share price) to get the estimated **historical portfolio position** and the **return distribution** used for VaR calc.

SWAP

Historical Full Revaluation Swap 1- day 95% VaR: **$ 962,541.68**




```
# apply historical rate changes to today's rates
new_sofr_10 = pd.DataFrame(latest_sofr_rate + sofr_delta.to_numpy())
# discount factors for each maturities
new_sofr_10_disc = compute_discount_factors(new_sofr_10).to_numpy()

n = len(sofr_delta)
hist_FR_payer = np.zeros(n)

# compute payer swap value for each risk factor
for i in range(n):
    hist_FR_payer[i] = PV_payer_swap(new_sofr_10_disc[i])

# calculate change in value of swap
hist_FR_payer = hist_FR_payer - base_pv    # base_pv: swap initial portfolio
hist_FR_payer_VaR = abs(np.percentile(hist_FR_payer, 5))
```

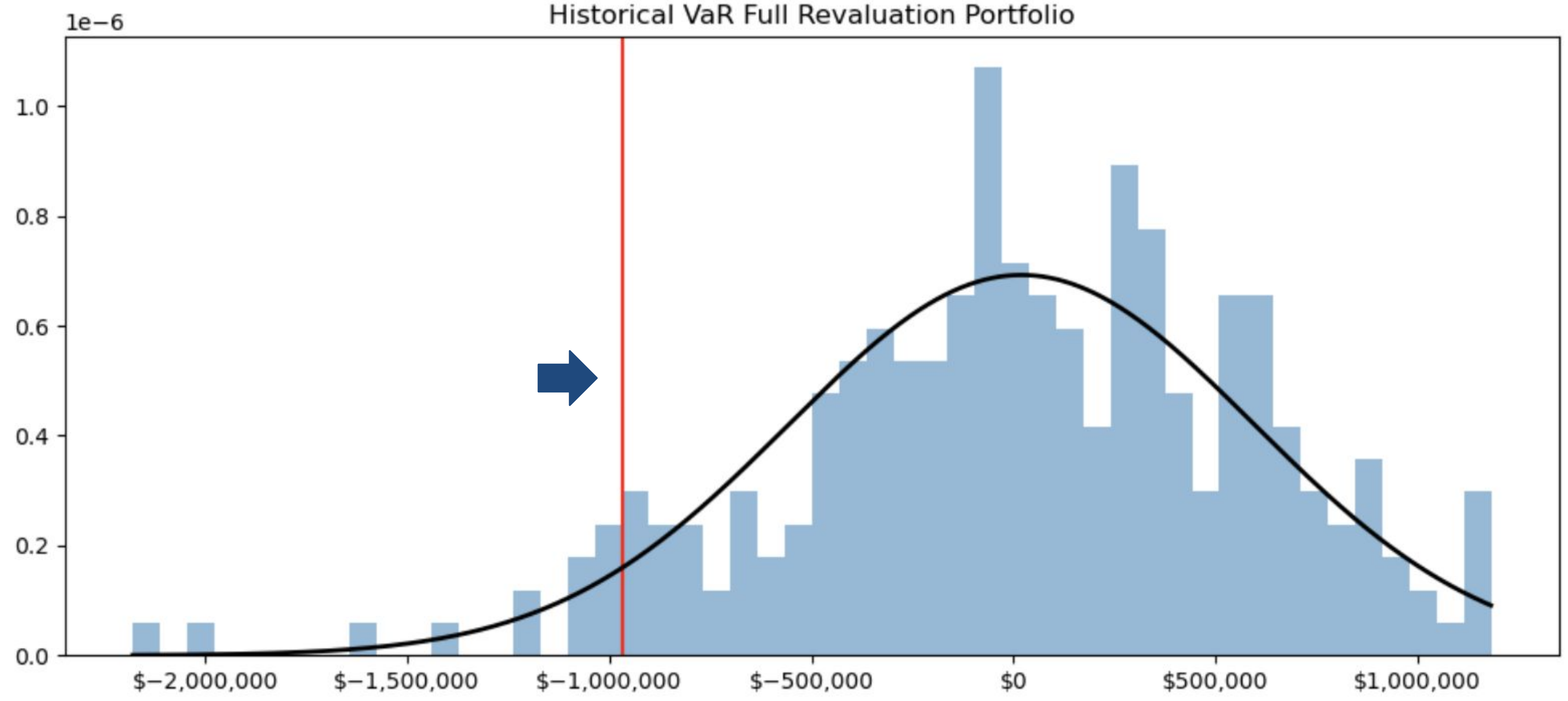
a. Recalculate the entire new swap portfolio value based on the **new SOFR zero rate curve** using the **historical zero rate return** distribution

b. Using the **changes of the new swap** price, obtain the 5th percentile of the distribution

Full Revaluation Approach

PORTFOLIO

```
# combine together stock and swap
hist_FR_portfolio_changes = hist_FR_stocks_portfolio_changes + hist_FR_payer

hist_FR_VaR = abs(np.percentile(hist_FR_portfolio_changes, VaR_percentile))
```



Historical Full Revaluation Portfolio 1-day 95% VaR: $ 966,173.22

What are the important components for the application of full revaluation approach?

Daily historical risk factors data — Time series data as the basis estimation of the past behavior pattern. Long length of time period is recommended to capture different market regime.

Specific scenario-based — Revaluate the entire portfolio value based on a specific scenario over all horizon period. For the given case, we use the market situation **as of 30 Oct 2023.**

P&L Distribution — Capturing the distribution of historical P&L changes after the revaluation, by comparing with the initial portfolio value. Using scaling factor for longer horizon period

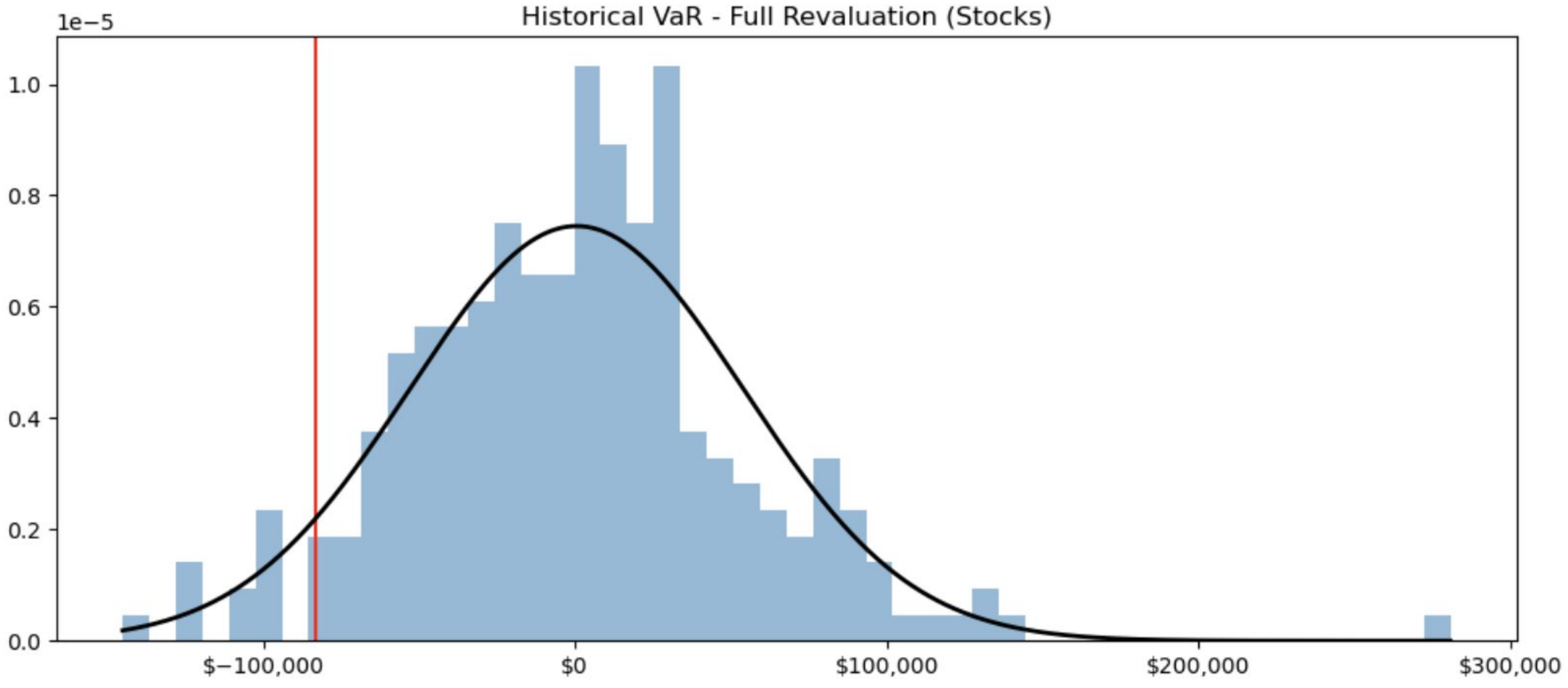
Key consideration

Since it is revaluate based on specific scenario could be **time intensive** process...
→ use the **delta** as sensitivity factor of portfolio value w.r.t to changes in risk factor

Sensitivity Based Approach

STOCKS

Historical Sensitivity-Based Stocks 1- day 95% VaR: $ 82,934.71



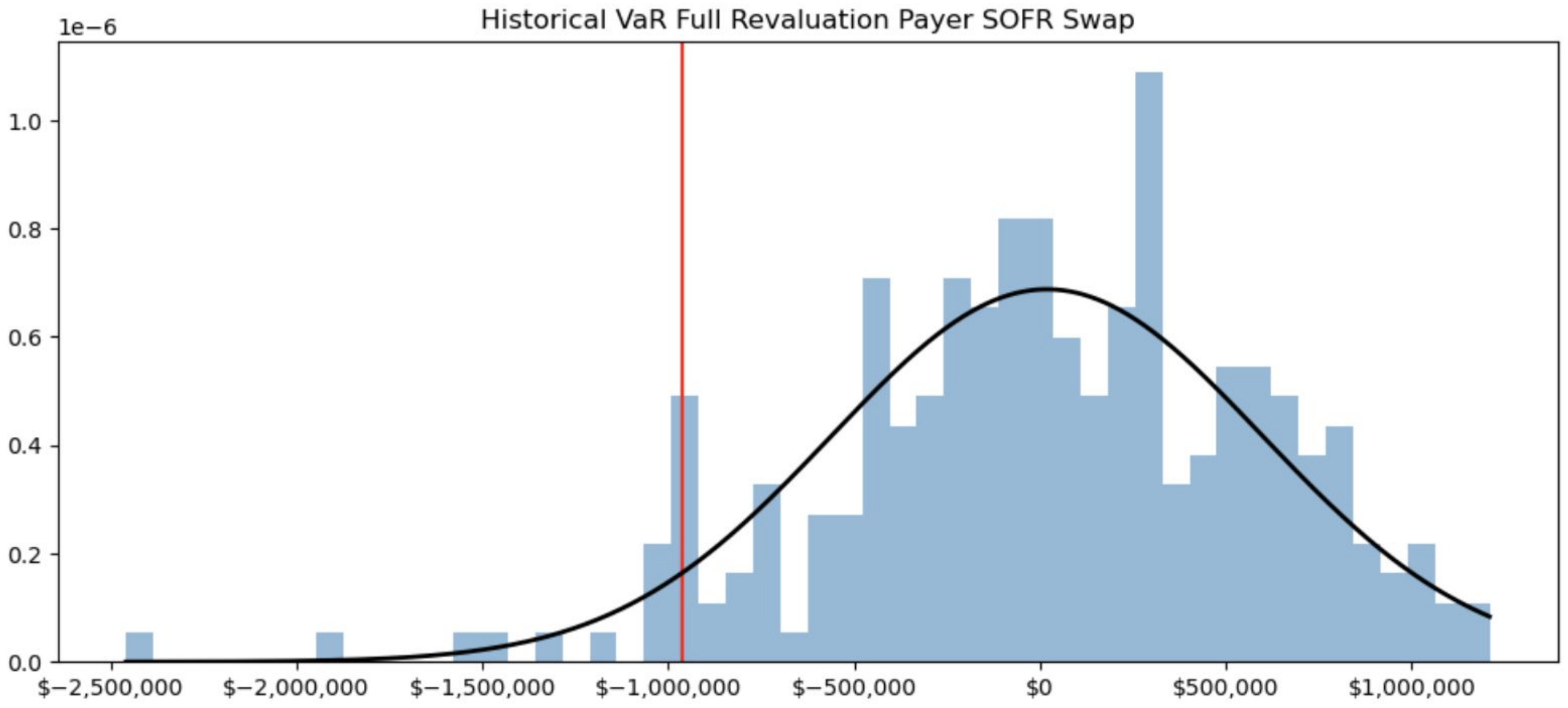

```
# multiply our return to notionals and sum all the 4 stocks together daily (250 cols)
hist_SB_stocks_changes = (stocks_daily_return * stocks_notional).sum(axis=1)

hist_SB_stocks_VaR = abs(np.percentile(hist_SB_stocks_changes, VaR_percentile))
```

a

Using the **historical asset return distribution** to simulate **future asset movement** by considering the sensitivity of the asset values w.r.t to the changes in the underlying risk factors.

SWAP

Historical Sensitivity-Based SOFR Swap 1- day 95% VaR: $ 956,909.67




```
# compute payer swap value for each risk factor using PV01 and sum it along the row (daily like stocks)
hist_SB_payer = (pv01 * sofr_delta.to_numpy()).sum(axis = 1).astype(float)

hist_SB_payer_VaR = abs(np.percentile(hist_SB_payer, VaR_percentile))
```

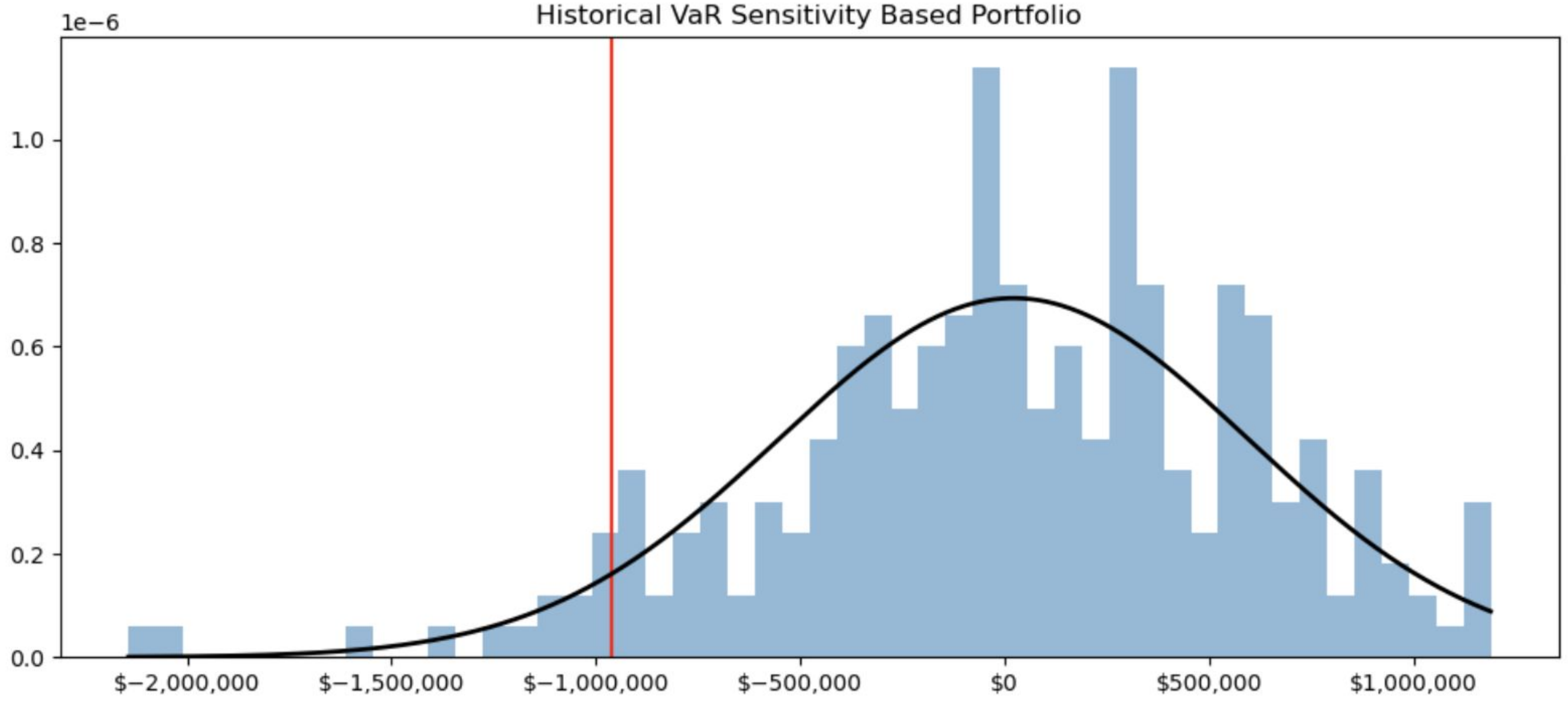
b

Using the **historical zero rate curve changes distribution,** multiply by the PV01 that measures the sensitivity of the swap values w.r.t to the changes in the underlying risk factors. In this case, sensitivity captures changes of zero rate curve **as of 30 Oct 2023.**

Sensitivity Based Approach

PORTFOLIO

```
# combine together stock and swap
hist_SB_portfolio_changes = hist_SB_stocks_changes + hist_SB_payer

hist_SB_VaR = abs(np.percentile(hist_SB_portfolio_changes, VaR_percentile))
```



Historical Sensitivity Based Portfolio 1- day 95% VaR: **$ 960,687.72**

What are the important components for the application of sensitivity based approach?

- **Daily historical risk factors data**: Time series data as the basis estimation of the past behavior pattern. Long length of time period is recommended to capture different market regime.
- **Sensitivity estimation**: It approximates the changes in the portfolio value using the single factor of risk sensitivity measures. Could be first or second order sensitivity based on the asset types
- **Faster computational process**: The process takes faster than the full-revaluation approach since it assumes that sensitivity remains constant for small market moves

Key consideration

Since it estimates only based on the **sensitivity** of portfolio to the risk factor changes...
→ may **not accurate** if large market shock exists, which the sensitivity itself also changes (e.g., gamma effect)

Indicator	Parametric	Monte Carlo	Historical
Application	• Assume returns is normally distributed • Returns should be **independent, linear** and **stable** risk factors over time • Works well for **linear portfolios** (stocks, bonds)	• Uses **historical returns** to simulate infinite possible future return movement in determining future value • Use **repeated random sampling** process • Use in **complex risk modeling**	• Use **historical returns** to estimate the future return movement in determining future value • Only use **sorted** historical returns and specified **confidence interval** • **Captures real-world return distribution**
Advantages	• Variables are **easy to be obtained** • Computational **efficient** (requires only mean and standard deviation)	• Model **complex instruments** and capture **nonlinear risks** • **No strict** assumptions on return distribution	• Accounts for **skewness and fat tails in asset returns** • **Simple to implement:** based on past actual market data than model assumptions
Disadvantages	• **High sensitivity** on the parameter input • **Underestimate risk** for portfolios with fat tails and non-linear instruments like options • **Inadequate** for handling **extreme events** which could alter the risk variance/covariance matrix over the time	• **Computationally expensive,** require many simulations • Difficult to implement and interpret • Could lead to **higher sampling variability** that does not decrease as number trials ↑	• **Inadequate data points** to model may lead inaccurate risk calculation if past crises are not in the dataset • Relies on historical data, may **not represent future risks** (e.g. new market conditions)
Best Use Case	**Large financial institutions (risk management, regulatory reporting, and portfolio risk measurement)** Suitable for asset returns that behaved in the well and stability markets	**Financial services & banking, insurance, energy & commodities, engineering & manufacturing, health care** Suitable for assessing non-linear relationships or complex pricing products, and portfolios with derivatives	**Banks and hedge funds** Suitable for portfolio cases that contains large number assets and non-linear products, operating in high uncertainties industry

Results of Portfolio 1-day 95% VaR

Parametric	Monte Carlo	Historical
$ 936, 434.42	Full-revaluation: $944, 164.63 Sensitivity based: $ 938, 834.54	Full-revaluation : $966, 713.22 Sensitivity based : $ 960, 687.72

Why historical VaR has the highest value?

Historical VaR directly relies on actual past market data, capturing extreme market movements that Parametric and Monte Carlo may not fully reflect.

Key reasons:

1. **Non-normal return distribution**
2. **Inclusion of market crashes or volatility spikes**
3. **Empirical nature vs. model based estimates**
4. **Market conditions at different time periods**

Why does the value of full-revaluation is greater than sensitivity-based?

Full-revaluation is more conservative because it incorporates real-world, complex scenarios that linear sensitivity analysis might overlook. This leads to a higher risk estimate, providing a more robust buffer against market uncertainties

Key factors:

1. **Non-linear instruments**
2. **Market volatility**
3. **Historical extremes**
4. **Compounded effects of correlation**

QF609 - Risk Analysis Final Project

Thank you

Group 4 - G1 MQF 2024