**Market Risk Analytics Project: Implementation Guide / User Manual**

Table of Contents

[1. System Requirements 2](#_Toc199819866)

[2. Project Setup 2](#_Toc199819867)

[2.1. Clone the Repository (if applicable) 2](#_Toc199819868)

[2.2. Install Dependencies 2](#_Toc199819869)

[2.3. Data Preparation 3](#_Toc199819870)

[3. Running the Script 3](#_Toc199819871)

[4. Configuration Parameters 3](#_Toc199819872)

[5. Interpreting the Output 4](#_Toc199819873)

[5.1. Data Information 4](#_Toc199819874)

[5.2. Market Risk Metrics 4](#_Toc199819875)

[5.3. VaR Model Backtesting (Kupiec's Proportion of Failures Test) 4](#_Toc199819876)

[5.4. CVaR Model Backtesting (Average Exceedance Test) 5](#_Toc199819877)

[5.5. Scenario Analysis & Stress Testing 5](#_Toc199819878)

[5.6. FRTB Considerations 5](#_Toc199819879)

[6. Troubleshooting 5](#_Toc199819880)

This document provides instructions for setting up the environment, running the market\_risk\_dashboard.py script, and understanding its output. It is intended for users who wish to execute the market risk analysis and review the results.

# 1. System Requirements

* **Operating System:** Windows, macOS, or Linux.
* **Python:** Python 3.7 or higher.
* **Package Manager:** pip (usually comes with Python installation).

# 2. Project Setup

To get started, follow these steps:

## 2.1. Clone the Repository (if applicable)

If your project is hosted on GitHub, clone it to your local machine:

git clone https://github.com/YourUsername/MarketRiskAnalyticsProject.git

cd MarketRiskAnalyticsProject

(Replace YourUsername and MarketRiskAnalyticsProject with your actual GitHub username and repository name.)

If you downloaded the files directly, simply navigate to the project's root directory in your terminal.

## 2.2. Install Dependencies

The project relies on several Python libraries. It's recommended to create a virtual environment to manage dependencies:

1. **Create a virtual environment (optional but recommended):**
2. python -m venv venv
3. **Activate the virtual environment:**
   * **On Windows:**
   * .\venv\Scripts\activate
   * **On macOS/Linux:**
   * source venv/bin/activate
4. **Install the required packages:**
5. pip install -r requirements.txt

The requirements.txt file should contain:

pandas

numpy

scipy

openpyxl

## 2.3. Data Preparation

* **Excel File:** Ensure your historical portfolio data is available in an Excel file named Historical\_Portfolio.xlsx.
* **Sheet Name:** The relevant data must be in a sheet named portfolio within Historical\_Portfolio.xlsx.
* **Column Names:** The sheet must contain a Date column (for the index) and columns for the closing prices of your assets (e.g., Cipla\_close, TATACON\_close, BEL\_close). **Ensure column names are exact and case-sensitive.**
* **Placement:** Place the Historical\_Portfolio.xlsx file in the data/ directory within your project structure.

# 3. Running the Script

Navigate to the src/ directory where market\_risk\_dashboard.py is located (or ensure your terminal is in the root directory and specify the full path to the script).

Execute the script using Python:

python src/market\_risk\_dashboard.py

The script will print its output directly to the console.

# 4. Configuration Parameters

You can modify key parameters directly within the market\_risk\_dashboard.py file under the --- Configuration --- section:

* **excel\_file\_path**: Path to your historical data Excel file.
* **sheet\_name**: Name of the sheet containing the data within the Excel file.
* **confidence\_level**: The confidence level for VaR and ES calculations (e.g., 0.99 for 99%, 0.95 for 95%).
* **num\_simulations**: Number of simulations for the Monte Carlo VaR/CVaR approach (e.g., 10000). A higher number increases accuracy but also computation time.

# 5. Interpreting the Output

The script's output is structured into several sections:

## 5.1. Data Information

* **Original Data Head/Info/Description:** Shows the first few rows, data types, and summary statistics of your loaded price data.
* **Daily Returns Head/Info:** Shows the first few rows and information about the calculated daily percentage returns.
* **Portfolio Weights:** Indicates the weighting scheme used (currently equal weighting).
* **Portfolio Returns Statistics:** Displays the mean and standard deviation of the daily portfolio returns.

## 5.2. Market Risk Metrics

* **Historical VaR & CVaR:** Non-parametric risk measures derived directly from historical data.
  + **VaR:** The maximum expected loss at the specified confidence level based on past performance.
  + **CVaR (Expected Shortfall):** The expected loss given that the VaR threshold has been breached (i.e., the average of the worst losses).
* **Parametric (Delta-Normal) VaR & CVaR:** Risk measures assuming a normal distribution of returns.
  + **VaR:** Calculated using the portfolio's mean, standard deviation, and a Z-score.
  + **CVaR:** Derived from the normal distribution's properties.
* **Monte Carlo VaR & CVaR:** Risk measures derived from simulated future returns.
  + **VaR:** The specified percentile of the simulated loss distribution.
  + **CVaR:** The average of simulated losses worse than the Monte Carlo VaR.

## 5.3. VaR Model Backtesting (Kupiec's Proportion of Failures Test)

* **Total Observations (T):** Total number of historical return periods used for backtesting.
* **Observed VaR Exceptions (N):** Number of times actual losses exceeded the VaR.
* **Expected VaR Exceptions:** The number of exceptions the model *should* produce given the confidence level.
* **Observed Proportion of Failures (p\_hat):** N / T.
* **Kupiec's LR\_POF Test Statistic:** The calculated test statistic.
* **Critical Value:** The threshold from the Chi-squared distribution.
* **Conclusion:** States whether the model is rejected (observed exceptions are significantly different from expected) or not rejected.

## 5.4. CVaR Model Backtesting (Average Exceedance Test)

* **Number of VaR violations:** How many times the VaR threshold was breached.
* **Average Actual Loss on VaR Violation Days:** The average of the actual losses on the days when VaR was exceeded.
* **Predicted Historical CVaR:** The CVaR value calculated by the model.
* **Qualitative CVaR Backtest:** A simple comparison indicating if the model might be underestimating or overestimating tail losses based on the average actual exceedance.

## 5.5. Scenario Analysis & Stress Testing

* **Current Asset Prices / Portfolio Value:** The starting point for the stress tests.
* **Stress Test Results:** For each defined scenario (e.g., "2008 Global Financial Crisis"), it displays:
  + **Stressed Portfolio Value:** The portfolio value after applying the scenario's shocks.
  + **Loss Under Stress:** The absolute dollar loss under the scenario.
  + **Percentage Loss:** The percentage loss relative to the current portfolio value.

## 5.6. FRTB Considerations

* This section provides conceptual notes on how the implemented metrics relate to FRTB requirements, including discussions on ES, Backtesting, Stress Testing, Non-Modellable Risk Factors (NMRF), Default Risk Charge (DRC), and P&L Attribution (PLA). It highlights the project's alignment and areas for further development for full regulatory compliance.

# 6. Troubleshooting

* **"Error: The file 'Historical\_Portfolio.xlsx' was not found."**: Ensure the Excel file is in the data/ directory relative to your script, or provide the full, correct path.
* **"Index Date invalid (sheet: portfolio)"**: Verify that your Excel sheet named portfolio has a column explicitly named Date (case-sensitive) and that it's correctly formatted as dates.
* **NaN values in returns**: This is normal for the first row. If NaNs persist, check for non-numeric data or missing prices in your original Excel file.
* **"No VaR violations observed, cannot perform CVaR backtesting"**: This happens if your historical data period did not experience losses worse than your chosen VaR confidence level. This is a good sign for your portfolio's historical performance but limits the CVaR backtest.

This guide should help you effectively use and understand your Market Risk Analytics project.