

Programming in Economics & Finance II, Final Project

Niccolò Lecce, Marco Gasparetti, Alessandro Dodon

Introduction

Risk management is central to finance, yet many practitioners still rely on inefficient tools like spreadsheets. As programming becomes essential in financial workflows, there is a growing need for accessible, efficient solutions to compute Value-at-Risk (VaR) and other risk metrics.

We introduce a Python package that automates the full VaR estimation pipeline—from portfolio creation to risk metrics, visualization, backtesting, and natural language interpretation using LLMs. Focused on equity portfolios and simple models, the package prioritizes speed, clarity, and ease of use—ideal for both professionals and retail investors seeking intuitive, modern risk tools.

Project Plan

xxx

Project Diary

xxx

Theoretical Foundation

Basic VaR Models

Value-at-Risk at confidence level α is the smallest positive number z_α —the $(1 - \alpha)$ -quantile of the return distribution such that:

$$\Pr(r_t < -z_\alpha) = 1 - \alpha.$$

We start with models that keep volatility constant over time. In the non-parametric (historical) approach:

$$\text{VaR}_\alpha^{\text{hist}} = -\text{quantile}_{1-\alpha}(r_t), \quad \text{VaR}_\alpha^{\text{normal}} = z_\alpha \sigma,$$

and for a portfolio this naturally extends to the portfolio-normal form. Expected shortfall is evaluated as:

$$\text{ES}_{\text{hist}} = -\mathbb{E}[r_t \mid r_t < -z_\alpha], \quad \text{ES}_{\text{normal}} = \sigma \frac{\phi(z_\alpha)}{1 - \alpha}, \quad \text{ES}_t = \sigma \frac{f_{t\nu}(z_\alpha)}{1 - \alpha} \frac{\nu + z_\alpha^2}{\nu - 1},$$

where $\phi(\cdot)$ and $f_{t\nu}(\cdot)$ denote the corresponding density functions.

All VaR and ES estimates scale with the holding period as \sqrt{h} . For any univariate model, multiplying by wealth W turns the measure into monetary terms; omitting W leaves it as

a percentage. These constant-volatility benchmarks prepare the ground for more advanced risk models.

Extreme Value Theory

Technically still a parametric model, EVT is more sophisticated than the methods discussed above. We implement it with the Peaks-Over-Threshold (POT) approach: once a high threshold u (e.g., the 99th-percentile loss) is chosen, a Generalized Pareto Distribution (GPD) is fitted to the excesses above u .

VaR and ES are then derived analytically from the fitted GPD parameters. This allows for more accurate estimation of rare, extreme losses. The formulas used are:

$$\widehat{\text{VaR}}_\alpha = u + \frac{\hat{\beta}}{\hat{\xi}} \left[\left(\frac{n}{n_u} (1 - \alpha) \right)^{-\hat{\xi}} - 1 \right], \quad \widehat{\text{ES}}_\alpha = \frac{\widehat{\text{VaR}}_\alpha + \hat{\beta} - \hat{\xi}u}{1 - \hat{\xi}},$$

where $\hat{\xi}$ and $\hat{\beta}$ are the estimated shape and scale parameters, n is the sample size, and n_u is the number of exceedances.

Volatility Models

We have implemented the most popular univariate volatility models.

The GARCH(1,1) model is specified as:

$$\sigma_t^2 = \omega + \alpha r_{t-1}^2 + \beta \sigma_{t-1}^2, \quad r_t = \sigma_t z_t,$$

where z_t are i.i.d. standardized shocks. Value-at-Risk is then computed as:

$$\text{VaR}_{t,\alpha} = -\hat{\sigma}_t z_\alpha.$$

Future volatility can be forecasted analytically using GARCH(1,1).

The τ -step ahead forecast and cumulative T -step forecast are:

$$\begin{aligned} \mathbb{E}[\sigma_{t+\tau}^2] &= \text{VL} + (\alpha + \beta)^\tau (\sigma_t^2 - \text{VL}), \\ \mathbb{E}[\sigma_{t,T}^2] &= \text{VL} \left(T - 1 - \frac{(\alpha + \beta)(1 - (\alpha + \beta)^{T-1})}{1 - (\alpha + \beta)} \right) + \sigma_t^2 \frac{1 - (\alpha + \beta)^T}{1 - (\alpha + \beta)}, \end{aligned}$$

which naturally allow multi-period VaR estimation.

We also support EGARCH (logs variance, capturing leverage), GJR-GARCH (adds an indicator for negative shocks), and APARCH (introduces a power term for flexible tail dynamics), all with maximum-likelihood estimation under Normal, Student- t , GED, or Skew- t innovations.

For simpler benchmarks we include ARCH(p) and two rolling estimators, Moving Average (MA) and Exponentially Weighted Moving Average (EWMA):

$$\sigma_t^{2,\text{ARCH}} = \omega + \sum_{i=1}^p \alpha_i r_{t-i}^2, \quad \sigma_t^{2,\text{MA}} = \frac{1}{n} \sum_{i=1}^n r_{t-i}^2, \quad \sigma_t^{2,\text{EWMA}} = \lambda \sigma_{t-1}^2 + (1 - \lambda) r_{t-1}^2.$$

ES for volatility models becomes:

$$\text{ES}_{t,\alpha} = -\hat{\sigma}_t \mathbb{E}[z_t \mid z_t < z_\alpha].$$

All our volatility models use a semi-empirical approach for the VaR computation, first calculating the empirical distribution of innovations with their respective volatility estimate and then using the required percentile of those innovations as z .

Time-Varying Correlation Models

We now introduce several models to analyze an entire portfolio and not just single assets. We implement the two simplest time-varying correlation models: moving average (MA) and exponentially weighted moving average (EWMA). These serve as natural multivariate extensions of the corresponding univariate volatility models.

Because a semi-empirical approach is considerably harder in this setting, we rely on the normal distribution to obtain the needed quantiles.

Given the monetary position vector x_t and the time-varying covariance matrix Σ_t estimated by the model, VaR is calculated as:

$$\text{VaR}_t = z_\alpha \cdot \sqrt{x_t^\top \Sigma_t x_t}.$$

More advanced models like DCC-GARCH or VEC(1,1) are not easily supported in Python and are therefore not included.

Expected Shortfall in this setting is computed using the analytical formula under the normal distribution assumption.

Factor Models

We implement the Sharpe model and Fama-French three factor models to estimate portfolio VaR and ES. These models use asset betas with respect to systematic risk factors, allowing us to infer portfolio-level risk from exposures to common sources of variation.

For the Sharpe Model, we assume that each asset's return r_i follows the linear factor structure:

$$r_i = \alpha_i + \beta_i r_m + \varepsilon_i,$$

where r_m is the return on the market portfolio, β_i is the asset's sensitivity to the market factor, and ε_i is a zero-mean idiosyncratic shock uncorrelated with r_m and other residuals. The market return r_m has variance σ_m^2 , and $\text{Var}(\varepsilon_i) = \sigma_{\varepsilon_i}^2$.

Let $w = (w_1, \dots, w_N)^\top$ be the vector of monetary portfolio weights. Then the total portfolio variance becomes:

$$\sigma_p^2 = \left(\sum_{i=1}^N w_i \beta_i \right)^2 \sigma_m^2 + \sum_{i=1}^N w_i^2 \sigma_{\varepsilon_i}^2.$$

VaR and ES are computed under the normality assumption:

$$\text{VaR}_{t,\alpha} = z_\alpha \cdot \sigma_p, \quad \text{ES}_{t,\alpha} = \sigma_p \cdot \frac{\phi(z_\alpha)}{1 - \alpha}.$$

We also extend the previous model by considering three risk factors: market excess return (Mkt-RF), size (SMB), and value (HML). Each asset's excess return is modeled as:

$$r_i - r_f = \alpha_i + \beta_{i,1} \cdot \text{Mkt-RF} + \beta_{i,2} \cdot \text{SMB} + \beta_{i,3} \cdot \text{HML} + \varepsilon_i,$$

where r_f is the risk-free rate, and ε_i is the idiosyncratic error. We estimate factor loadings $\beta_{i,k}$ through OLS regression.

Let B be the $N \times 3$ matrix of estimated factor loadings, Σ_f the 3×3 sample covariance matrix of the factors, and Σ_ε the diagonal matrix of residual variances $\sigma_{\varepsilon_i}^2$. The total portfolio variance is:

$$\sigma_p^2 = w^\top (B \Sigma_f B^\top + \Sigma_\varepsilon) w.$$

Once the portfolio variance is known, we compute:

$$\text{VaR}_{t,\alpha} = z_\alpha \cdot \sigma_p, \quad \text{ES}_{t,\alpha} = \sigma_p \cdot \frac{\phi(z_\alpha)}{1 - \alpha}.$$

Specific Portfolio Metrics

We compute several (specific) portfolio-level Value-at-Risk (VaR) measures using monetary positions $x_t = (x_{1,t}, \dots, x_{N,t})^\top$, the covariance matrix Σ , and a time horizon of h days.

The quantile z_α corresponds to the desired confidence level and is always derived from the standard normal distribution, as the underlying assumption is that asset returns (or their innovations) are normally distributed.

The asset-normal parametric portfolio VaR is defined as:

$$\text{VaR}_t = z_\alpha \cdot \sqrt{x_t^\top \Sigma x_t} \cdot \sqrt{h}.$$

Assuming all asset returns are perfectly correlated ($\rho = 1$) and there are no short positions, the undiversified portfolio VaR is simply the sum of the individual asset VaRs:

$$\text{UVaR}_t = \sum_{i=1}^N \text{VaR}_{i,t}.$$

The marginal VaR, representing the sensitivity of portfolio VaR to a small change in position $x_{i,t}$, is given by:

$$\Delta \text{VaR}_{i,t} = \text{VaR}_t \cdot \frac{(\Sigma x_t)_i}{x_t^\top \Sigma x_t}.$$

It measures the change in total VaR from adding one extra unit of asset i .

The component VaR is:

$$\text{CVaR}_{i,t} = x_{i,t} \cdot \Delta \text{VaR}_{i,t},$$

which captures the absolute contribution of asset i to total portfolio VaR.

The relative component VaR is:

$$\text{RCVaR}_{i,t} = \frac{\text{CVaR}_{i,t}}{\text{VaR}_t},$$

interpreted as the percentage share of total VaR attributable to asset i .

For a vector a representing changes in the portfolio allocation, the incremental VaR is:

$$\text{IVaR}_t = \Delta \text{VaR}_t^\top \cdot a,$$

which estimates the change in total VaR resulting from shifting the portfolio by a .

Each of these portfolio-level VaR metrics can be complemented with a corresponding Expected Shortfall (ES), computed under the normality assumption.

Simulation Methods

Simulation methods are powerful to compute the VaR and ES of portfolios with more complex instruments, such as options, which have a non-linear payoff.

We implement a one-day parametric Monte Carlo simulation approach to estimate VaR and ES. Under the assumption of multivariate normality, we simulate future returns as:

$$\begin{aligned} \mu &= \mathbb{E}[r_t], \quad \Sigma = \text{Cov}(r_t), \quad L = \text{Cholesky}(\Sigma), \\ r^{(i)} &= \mu + L z^{(i)}, \quad z^{(i)} \sim \mathcal{N}(0, I), \\ S^{(i)} &= S_0 \circ (1 + r^{(i)}), \end{aligned}$$

where \circ denotes element-wise multiplication.

The simulated P&L of the portfolio is calculated as:

$$\Delta P^{(i)} = w^\top (S^{(i)} - S_0) + \sum_{j=1}^{N_{\text{opt}}} q_j \left(C(S_j^{(i)}, \tau_j') - C_{j,0} \right),$$

where q_j is the number of contracts for option j , $C(\cdot)$ is the Black-Scholes option price, and $\tau_j' = \max(T_j - \frac{1}{252}, 0)$ is the adjusted time to maturity.

From the empirical distribution of $\{\Delta P^{(i)}\}_{i=1}^N$, we compute:

$$\text{VaR}_\alpha = -\text{Quantile}_\alpha(\Delta P), \quad \text{ES}_\alpha = -\mathbb{E}[\Delta P \mid \Delta P \leq -\text{VaR}_\alpha].$$

We also implement a version of Monte Carlo over multiple days, effectively serving as a way to forecast for VaR and ES. This is limited to equity only portfolios. To simulate risk over multiple days (h -day horizon) for pure equity portfolios, we model prices as following a geometric Brownian motion:

$$S_{t+1}^{(i)} = S_t^{(i)} \cdot \exp \left(\mu - \frac{1}{2} \sigma^2 + L z_t^{(i)} \right), \quad z_t^{(i)} \sim \mathcal{N}(0, I),$$

for $t = 0, \dots, h-1$, where L is the Cholesky factor of the covariance matrix.

The final P&L is:

$$\Delta P^{(i)} = \sum_{k=1}^N w_k (S_{h,k}^{(i)} - S_{0,k}).$$

The VaR and ES are computed from the empirical distribution of simulated $\Delta P^{(i)}$.

A non-parametric alternative to Monte Carlo are the class of Historical Simulations. We implement both the version with and without replacement. We simulate alternative return paths by re-applying historical return vectors to the current portfolio composition. For each scenario i , we define:

$$S^{(i)} = S_0 \circ (1 + r_{\pi(i)}),$$

where $r_{\pi(i)}$ is the i -th sampled return vector from the historical dataset, and $\pi(i)$ is either a fixed ordering (historical simulation) or a random draw (bootstrap simulation).

The corresponding P&L is:

$$\Delta P^{(i)} = w^\top (S^{(i)} - S_0).$$

Risk measures are then computed as:

$$\text{VaR}_\alpha = -\text{Quantile}_\alpha(\Delta P), \quad \text{ES}_\alpha = -\mathbb{E}[\Delta P \mid \Delta P \leq -\text{VaR}_\alpha].$$

This approach is fully non-parametric and accounts for fat tails and nonlinear dependence in historical returns. When bootstrap is enabled, multiple resamples ensure robustness to limited data windows.

Backtesting

Backtesting assesses whether the observed losses are consistent with the VaR predictions by checking the frequency and structure of violations (exceptions). The objective is to test whether the model is correctly calibrated.

We implement three standard tests.

The Kupiec test (unconditional coverage) evaluates if the number of observed exceptions N over T days matches the expected number under the model. If p is the failure probability, then under the null N follows a binomial distribution. The likelihood ratio test statistic is:

$$\text{LR}_{\text{uc}} = -2 \left\{ \ln \left[(1-p)^{T-N} p^N \right] - \ln \left[(1-\hat{p})^{T-N} \hat{p}^N \right] \right\}, \quad \hat{p} = \frac{N}{T}, \quad \text{LR}_{\text{uc}} \sim \chi_1^2.$$

The Christoffersen test (conditional coverage) tests the independence of exceptions by modeling their dynamics as a first-order Markov chain. It compares the likelihoods of transition counts under the null (independence) and alternative. The statistic is:

$$\text{LR}_{\text{c}} = -2(\ln L_0 - \ln L_1), \quad \text{LR}_{\text{c}} \sim \chi_1^2,$$

where L_0 is the likelihood under independence and L_1 under the alternative.

The joint test combines both:

$$\text{LR} = \text{LR}_{\text{uc}} + \text{LR}_{\text{c}} \sim \chi_2^2.$$

Rejecting the null in any of the tests suggests that the VaR model is either miscalibrated (too many or too few violations) or fails to capture time dependence in risk (e.g., volatility clustering).

Our back-testing framework covers every method except the simulation models. Each simulation delivers only a single-day (or multiple day) forecast; to back-test them we would

need to rerun a full rolling simulation at every step—a task that is both logically more involved and computationally heavy—so it lies outside the present scope.

Results and Applications

xxx

Lessons Learned

xxx

AI Acknowledgement

xxx

Link to GitHub Repository

xxx