Wall Risk Engine® Quant Guide

February 2014

Contents

List of Figures

1 Introducing Wall Risk Engine®

1.1 What is Wall Risk Engine®?

1.1.1 Overview

Wall Risk Engine® is a software component composed of 5 modules:

- Risk and expert modeling: estimating the risk (volatilities, correlations,...) and the performance (expected returns) of the components of your investment universe;
- Risk and performance analysis: analyzing (ex-ante and ex-post) the risk and performance properties of your financial instruments and portfolios;
- Robust portfolio optimization: designing tailored, dynamic and robust allocation strategies;
- Simulation and stress-tests: Simulating future values of financial instruments (simple and structured products) and portfolios, computing large-scale risk measures (VaR) based on Monte-Carlo approaches.
- Data preparation: outlier detection and data completion functionalities to prepare the data for the risk modeling and risk analysis.

1.1.2 Main features

Risk and performance modeling

This module provides advanced functionalities to build tailored risk and performance models, using both historical data and exogenous information such as anticipations, market-implied information or in-house scores. Monitoring and estimating the correlations (or covariances) across all asset classes is a very sensitive issue especially within a portfolio optimization or a pricing process. The risk models provided in Wall Risk Engine® rely on 3 main pillars:

- Computing reactive and dynamic historical estimations of the correlations and volatilities (or the variance/covariance matrix);
- Taking into account exogenous information (if any) about the correlations and the volatilities;

- Applying a relevant filter to ensure the robustness of the risk model (by removing the noise) and merge historical and exogenous information.

Risk and performance analysis

This module embeds a wide range of ex-ante and ex-post analysis designed to monitor the risk and performance of a given portfolio:

- Performance measures;
- Risk measures: Volatility, semi-volatility, tracking-error, robust VaR, drawdown,...;
- Ratios: Sharpe, Sortino, Information, Omega;
- 3rd and 4th moments of the distribution: skewness, kurtosis;
- Risk and performance contributions.

Robust portfolio optimization

This module provides optimization functions to design tailored portfolios:

- Minimization of the risk of the portfolio given some performance constraint;
- Risk Budgeting: maximization of the expected return of the portfolio given some volatility budget constraints;
- Market portfolio (with maximum Sharpe ratio);
- Off-the-shelf adaptive allocation model to build reactive and robust portfolios;

All these functionalities embed patented robust optimization techniques and provide reliable and pertinent portfolios, ensuring a smooth behaviour with respect to market changes. These allocation models are compatible with any type of linear constraints on the weights such as sectorial constraints, asset-level constraints.

Simulation and stress-testing

This module provides tools to:

- simulate financial instruments using factorial or dense simulation models,
- build scenario simulations based on exogenous factors,
- stress-test a financial instrument or a portfolio,
- compute risk measures (Value at Risk) on large-scale portfolios based on Monte-Carlo simulations.

Quant data preparation

This module provides functionalities to prepare the market data for the risk modeling and analysis:

- detecting outliers in multi-dimensional time series,
- completing historical time series with robust regression approaches.

2

Risk and Performance Modeling

2.1 Introduction

Estimating the risk and performance of the investment universe is a crucial and sensitive step in the design of an allocation strategy. This module provides functionalities to build tailored risk and performance models, combining historical estimations and exogenous information.

2.2 Covariance and correlation matrices

2.2.1 Empirical estimation: the standard approach

Sample Covariances and Correlations

Sample estimates are a naive starting point for covariance and correlation modeling. Wall Risk Engine® provides functionalities to compute the empirical covariance matrix of a basket of financial instruments. The sample covariance estimate between two random signals r_i and r_j with means $\bar{r_i}$ and $\bar{r_j}$ is defined as:

$$\sigma_{i,j}^2 = \frac{1}{n-1}\sum_{t=1}^{n}\left(r_{i,t}-\bar{r_i}\right)\left(r_{j,t}-\bar{r_j}\right)$$

and their empirical correlation is defined as the scaled covariance:

$$\rho_{i,j} = \frac{\sigma_{ij}}{\sqrt{\sigma_{ii}\sigma_{jj}}}$$

Exponentially-weighted approach

One way to capture the dynamic features of the covariances is to use an exponential moving average of the historical observations, where the latest observations carry the highest weight in the estimate. This approach has two important advantages over the equally weighted model. First, covariances react faster to shocks (a large return for instance) and second, the volatility declines exponentially as the amplitude of the shock lowers. The exponentially-weighted covariance between two random signals r_i and r_j with means $\bar{r_i}$ and $\bar{r_j}$ is defined as:

$$\sigma\exp_{ij}^2 = (1-\lambda)\sum_{t=1}^{n}\lambda^{t-1}\left(r_{i,t}-\bar{r_i}\right)\left(r_{j,t}-\bar{r_j}\right)$$

where the parameter λ $(0 < \lambda < 1)$ is the decay factor of the observations. $\lambda = 0.97$ is a commonly used value for the decay factor.
The exponentially weighted correlations $\rho\text{exp}_{i,j}$ are obviously derived from $\sigma\text{exp}_{i,j}$.
NB: the exponentially weighted functions should be used only for historical time series with more than 50 historical observation dates.

Related functions

The related functions of Wall Risk Engine® are:

Function	Description
WREmodelingCov	Sample variance/covariance matrix
WREmodelingCovExp	Exponentially-weighted variance/covariance matrix
WREmodelingCorr	Sample correlation matrix
WREmodelingGARCHCov	variance/covariance matrix estimation with a DCC approach

2.2.2 Dealing with short series and missing data

Missing values and different length of historical series

To deal with incomplete historical data (historical quotations of different lengths, asynchronous or missing data), Wall Risk Engine® provides a set of functions to compute the covariances and correlations pairwise using all and only the available information, meaning with shortened series. But in such a case the resulting matrix is often ill-conditioned and can even fail to be positive definite.
Even in case we have a backup matrix for missing covariance values, the global matrix may be ill-conditioned or have negative eigenvalues. It is then necessary to perform a reconditioning (with the calibration functions presented in the next paragraph) in order to ensure the required statistical properties.

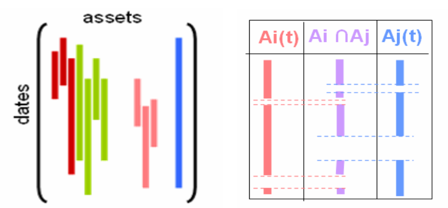


Figure 2.1: Historical quotations with missing data

Related functions

The related functions of Wall Risk Engine® are:

Function	Description
WREmodelingCovLack	Covariances with incomplete time series (holes)
WREmodelingCorrLack	Correlations with incomplete time series (holes)
WREmodelingCovLength	Covariances with different lengths of series
WREmodelingCorrLength	Correlations with different lengths of series

2.2.3 Adaptive risk modeling

Long vs. short estimation windows

Standard risk modeling approaches are based on long estimation windows (1 year or more) in order to ensure the required statistical properties of the historical estimations (the correlation matrix). The price to pay to ensure these good statistical properties is the lack of reactivity of the estimators during market turmoil. Indeed, **using long windows weakens the impact of recent market information**, especially when correlations are going down.

Shortening the estimation windows allows to build more reactive models, but introduces a sampling risk. In practice, the correlation matrix will embed noisy information and some sources of risk will be underestimated. This drawback is very dangerous when using the correlation matrix as an input of a portfolio optimization process, because the optimizer will be "fooled" by false portfolio with near-zero volatility and will produce unstable and irrelevant solutions.

The choice of the estimation windows depends also on the market conditions: in case of sudden market regime changes, the weight given to information from different past periods should automatically adapt. The difficulty lies in selecting the most relevant past information to achieve the right tradeoff between reactivity and robustness in the risk model.

Automatic historical data selection with the adaptive approach

To avoid the subjective choice of the estimation window, Raise Partner has developed an adaptive risk modeling approach. This method is based on the work of V. Spokoiny [MS04] on the volatilities: since then his work has been generalized to the correlations by several research teams (among which Raise Partner Quant team).

The adaptive approach consists in **detecting automatically regime changes** on the market and compute the optimal historical weights associated to the past periods. It differs from standard approaches (equally or exponentially weighted estimations introduced earlier in this chapter) in which the weights do not depend on market conditions.

Risk model pooling

Wall Risk Engine® allows you to combine several risk models (with both experts and quantitative inputs) based on the risk model pooling approach with a dynamic and automatic methodology attributing scores to the risk models depending on the market conditions.

The function WREmodelingCovAsema computes an adaptive risk model by dynamically combining covariance matrices coming from different estimation windows and optionally the current/latest one to include a Markovian effect.

The weights of the linear combination are the outputs of an optimization problem seeking to maximize the predictability of the volatility of the min-vol portfolio computed with the aggregated covariance matrix.

Related functions

The related functions of Wall Risk Engine® are:

Function	Description
WREmodelingAdaptiveCov	Adaptive estimation of the variance/covariance matrix
WREmodelingCovASEMA	adaptive risk model by dynamically combining covariance matrices

2.2.4 The role of risk model filtering

Why do we need to filter covariance matrices?

A big misunderstanding originates from the gap between two domains of the quantitative finance: the **descriptive statistics** (to explain the risk) and the **decision mathematics** (to allocate the risk). The same words are sometimes used in both contexts, such as the important notion of **robustness**, but do not mean the same thing. As a consequence, risk models which are acknowledged as robust in a descriptive risk analysis framework will not be suitable in a decision making context, and vice versa.

For example, **empirical estimations** might be the best estimates of the covariance matrices in a statistical approach, but using them in a portfolio optimization context can be highly dangerous. Indeed, statistical analysis shows that empirical covariance matrix estimators cannot capture properly the small contributions to risk, especially when using noisy market data. These small contributions to risk can be neglected when explaining the risk of a portfolio. Indeed, 95% or so of the information is sufficient to analyse/decompose the risk. Furthermore, such risk management approaches use large estimation windows (1 year or more) to compute the covariance matrix, which allows to reduce statistical errors.

But this is not the case in a **decision making framework** such as portfolio optimization. Indeed, to design a reactive (optimization-oriented) risk model and take the right allocation decision, we need to use shorter estimation windows, especially in times of highly volatile markets. **Random Matrix Theory** tells us that the lesser data we use, the more underestimated the small contributions to risk.

When optimizing a portfolio, underestimating these contributions (i.e. the smallest eigenvalues) can be very dangerous. Indeed, it implies that there exists a theoretical portfolio with a volatility close to zero! Indeed, The eigenvalue represents the variance of the portfolio defined by the corresponding normalized eigenvector. Hence, an eigenvalue close to zero implies that this portfolio has a volatility close to zero.

If this portfolio happens to have a positive expected return, then its Sharpe ratio will be $+\infty$, and the optimizer will be lured by this aberrant portfolio. In the end, this means that **the portfolio optimization process is essentially driven by noise** hence implying very large turnovers and operational risk at each reallocation.

When do we need to filter covariance matrices?

Matrix filtering is especially needed in situations where the specific structural properties of covariance matrices are destroyed:

- Heterogeneous data series (asynchronicity due to large and diversified investment universes, multi data-provider, several time zones);
- Aggregation of local information into a global risk model (estimation of local matrices by independent statistical approaches);
- Stress test of risk matrices (in a risk management framework);
- Combination of historical and exogenous information (ex. prior views or intervals constraints on your matrix).

Some random matrix theory background

Random Matrix Theory predicts a specific shape for the density of the eigenvalues of a purely random matrix. In the example below from [LLP99], we compare the density of a sample covariance matrix (the covariance matrix of the returns of the S&P500 index components) with the density of a random matrix. The figure 2.2 shows the predicted density shape for random matrices (in red) vs. the density of an empirical covariance matrix (in black):

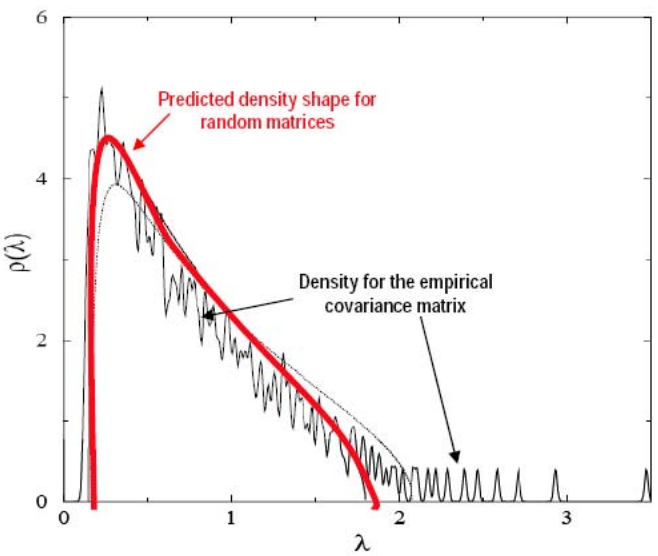


Figure 2.2: Compared eigenvalues densities (random matrix vs. sample matrix) from [LLP99]

This chart suggests that the eigenvalues of the sample matrix that are smaller than a certain threshold (around 2 in this example) might include much noise.

NB: we do not claim that this part of the density contains nothing but noise, that would be a shortcut. Indeed, every purely random matrix will have a density that fits this shape, but this does not mean that this is reciprocal: this chart cannot ensure that most of the information is noise. But in a conservative approach, we will consider that it might be.

Based on this hypothesis, only the upper part of the spectrum embeds reliable market information, as highlighted in figure 2.3:

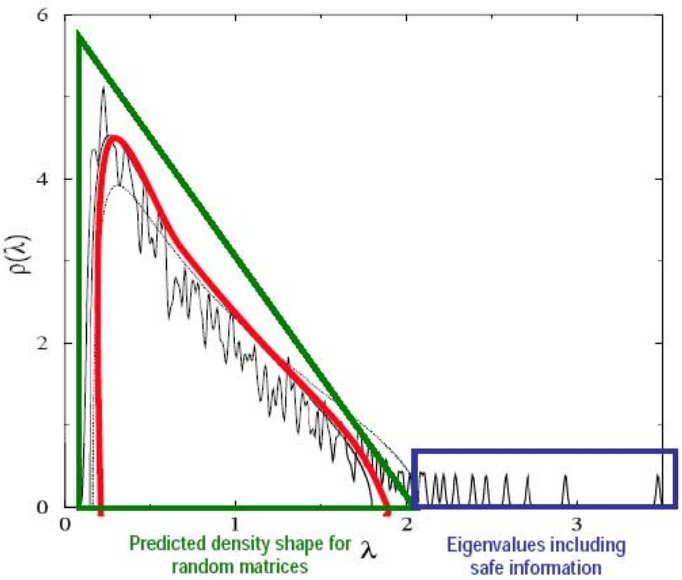


Figure 2.3: Noise vs. safe market information

2.2.5 The limits of standard eigenvalues correction

As explained in the previous paragraphs, using covariance matrices in a portfolio optimization framework requires a matrix correction process to filter the noise and distinguish it from pertinent market information.

A straightforward solution would be to simply increase the smallest eigenvalues to a specified threshold, using the Schur decomposition. This method would restore the semi-definite positivity property of the covariance matrix,but it does not allow to control explicitly both the deformation applied to the matrix and its conditioning.

2.2.6 Ledoit-Wolf matrix shrinkage

Ledoit and Wolf [LW03] propose a shrinkage method consisting in finding the right trade-off between bias and estimation error. The shrunken matrix is a linear interpolation of the sample covariance matrix S and a structured predefinite matrix F, i.e. $\Sigma = \alpha S + (1-\alpha)F$.

This method improves the conditioning of the covariance matrix if the structured estimate is already well conditioned and it has the nice property that there is an optimal explicit value of α that minimizes the residual $R = \Sigma - \alpha S - (1-\alpha)F$. A common choice for F is the identity matrix but doing so does not allow to control the deformation applied to the sample covariance matrix with explicit constraints.

Related functions

The related functions of Wall Risk Engine® are:

Function	Description
WREmodelingShrinkCov	Ledoit-Wolf shrinkage method

2.2.7 Robust, open and scalable matrix calibration with SDP approaches

Going beyond matrix reconditioning

Ensuring the good conditioning of covariance matrices is far below the expectations of Risk and Asset Management professionals who need to merge the historical estimation with exogenous information (market implied information, anticipations...). As a consequence, in addition to the requirements for reliable numerical computations, it is necessary to:

- preserve some internal blocks or a factorial structure (structural design),
- be able to add constraints such as minimal risk-levels on sub-market, or invariance of well-estimated sub-matrices.

The robust SDP matrix calibration methods [PMNP06] available in Wall Risk Engine® are a key component in this process, as they provide a framework embracing all of these requirement and therefore enhance the whole asset allocation and risk analysis process.

The SDP matrix filtering method provided in Wall Risk Engine® allows to build conservative and reactive risk model:

- **reactive**: capturing current trends by reducing the estimation windows and incorporating implied information;
- **conservative**: producing robust portfolio decisions by filtering the noise due to the lack/uncertainty of data.

We can compare this approach to the robust control model for a fighter aircraft: the model needs to react quickly but cautiously using very short-term information through a very conservative filtering scheme.

Our filtering method consists in taking the best candidate from the descriptive statistics world and making it cross towards the decision making world. In other words, we choose the PSD (Positive Semi-Definite) matrix which is the closest from the best candidate (the empirical covariance matrix for instance) and which verifies the set of required constraints.

This method allows us to:

- control the conditioning of the covariance matrix;
- control the deformation of the matrix with a set of constraints such as: preserving volatilities, giving prior views or confidence intervals on some part of the matrix, including market implied information.

Formulation of the SDP calibration

The Wall Risk Engine® matrix calibration routines are based on the resolution of pertinent optimization problems. These so-called SDP techniques [LEGL98] are useful to implement efficient and relevant risk models in a financial framework. They consist in finding the semi-definite positive (SDP) matrix which is the closest to the reference covariance matrix (empirical or block-assembled), once specified some calibration rules (which generate a set of linear constraints).

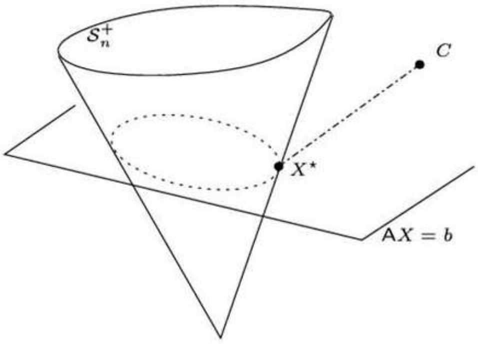


Figure 2.4: Constrained projection on the cone of positive semi-definite matrices

The SDP setting is especially adapted for covariance or correlation matrix calibration for two main reasons:

- the cone of semi-definite matrices captures the exact geometry of the set of correlation matrices,
- SDP problems have a key structural property : the convexity.

Properties of convex functions make them particularly suitable for modeling and aggregating risks. Among other properties, powerful analytical tools are available such as convex duality: convex duality not only enables advanced modeling approaches but also plays a key role in practical numerical implementation. The SDP problem is formulated as follows in the space $\mathcal{S}_n^+(\mathbb{R})$ of $n \times n$ symmetric matrices with positive eigenvalues equipped with the Frobenius norm[1]:

$$\begin{cases} \min_{X \in \mathcal{S}_n^+(\mathbb{R})} \parallel X - \Gamma \parallel^2 & \\ \langle A_{eq_i}, X \rangle = b_{eq_i}, \quad 1 \leq i \leq m_{eq} & \leftarrow \textbf{Block invariance constraints} \\ L_{ineq_i} \leq \langle A_{eq_i}, X \rangle \leq U_{ineq_i}, \quad 1 \leq i \leq m_{ineq} & \leftarrow \textbf{Interval constraints} \\ X - \alpha I_n \in \mathcal{S}_n^+(\mathbb{R}), \quad \alpha \geq 0 & \leftarrow \textbf{Minimum eigenvalue constraint} \end{cases}$$

[1]derived from the scalar product $\langle A, X \rangle = Tr\left(AX^\top\right)$

Mixing instantaneous market information and historical data
Investors have been using the implied market information (such as the VIX index) for a while to get the real-time market temperature: but how can investors use this instantaneous market information in a more systematic way to better apprehend their risk?

Risk models based on historical data, as advanced as they might be, are not able to detect short-term regime changes on the financial markets. Indeed, time-series have a smoothing effects on recent perturbations, which could lead to a dangerous under-estimation of the risk taken.

So why use historical data in the quantitative risk modeling approach? Historical information remains fundamental for a fine risk modeling and a dynamic estimation of the interdependences. Using the sole implied market information as an input for the risk estimation might give too much importance to localized picks of volatility, hence ignore the global trend.

Finding the right trade-off between instantaneous and historical information is the key: as simple as it may sound, this mixed approach requires matrix calibration tools to aggregate the information in a consistent and systematic way.

Expression of the correction level

The correction level is the parameter allowing to determine to what extent the covariance or correlation matrix should be filtered. The higher the correction level, the more over-estimated the risk of the underlying assets hence the more conservative the approach.
There are different ways to express the correction level for the covariance matrix calibration:

- The threshold of the minimum eigenvalue λ (all eigenvalues of the calibrated matrix should be greater than this threshold).

- The level of explained variance (x%): if we want to preserve x% (between 0% and 100%) of the explained variance when calibrating the matrix, then the eigenvalues that are contributing to this x% variance explication should be preserved. The level of explained variance is then translated into a minimum eigenvalue for the matrix calibration.

- An automatically-calibrated filtering level (α) between 0% and 100%: the bounds of this filtering level correspond to different levels of calibration depending on the nature (the level of risk) of the considered financial instruments. If this filtering level equals zero, then the matrix is left untouched. The larger the filtering level, the more conservative the approach, as a large filtering level corresponds to a high minimum eigenvalue for the matrix correction.

Related functions

There are 4 SDP calibration methods in Wall Risk Engine®:

Function	Constraints	Type of Matrix	Nature of the correction level
WREmodelingSDP	Linear	Correlation or covariance	minimum eigenvalue
WREmodelingSDPcorr	Linear	Correlation	minimum eigenvalue
WREmodelingSDPmodeler	Explicit or Block constraints	Correlation or covariance	minimum eigenvalue
WREmodelingCovFiltering	Linear	Correlation or covariance	filtering level between 0% and 100 %

The other related functions of Wall Risk Engine® are:

Function	Description
WREmodelingCalEpsVar	Computation of the minimum eigenvalue corresponding to a given level of explained variance

2.2.8 Structured filtering

General context

As seen in section 2.2.7 on page 17, one of the straight effects of the filtering is increasing the level of the lowest eigenvalues and consequently increasing the volatility of the less risky assets. This is not an issue while working on homogeneous universes, i.e. universes consisting of assets with similar volatility level.

When filtering a covariance matrix over a cross asset class universe, the fact of increasing the lowest volatility can have actual drawbacks. For instance, building a min-vol portfolio from such a risk model will lead to underweighting the low-volatility assets because they will be seen by the optimizer as riskier than they are.

A solution provided by the SDLS framework consists in filtering the global risk matrix by keeping the universe structure and risk levels: each homogeneous block will be filtered according to its specific risk and a global aggregated matrix will be computed to ensure the right mathematical properties (SDP).

This result can be achieved with either the WREmodelingSDLSmodeler function or more directly by the function WREmodelingStructuredFiltering.

An example: from local to global risk model

Context

One of the current challenges in risk modelling consists in building global risk models from local ones: from a set of local market risk forecasts and cross-market correlations, a global covariance matrix preserving local market estimations and restoring a positive semi-definite matrix must be computed.

It is not unusual that the global covariance matrix does not satisfy the required statistical properties because each local risk model is estimated by an independent approach. It may even fail to be positive semi-definite, meaning that it should be possible to construct a portfolio with negative volatility! The good mathematical properties of the global covariance matrix need to be restored by recalibrating the global market matrix while preserving as much economic information and/or strategic views as possible.

The difficulty stands in the aggregation of this information in a global unique matrix while:

- preserving each local model considered as the best local estimation;
- limiting the loss of information in the cross-market correlations.

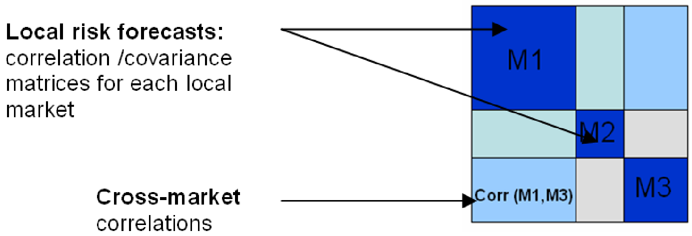


Figure 2.5: Risk aggregation across 3 different markets

Let Γ be the "raw" covariance matrix, computed pair-wise. The aggregation problem on p markets can be formulated as a SDP problem with positivity and block constraints:

$$\begin{cases} \min_{X \in \mathcal{S}_n^+(\mathbb{R})} \parallel X - \Gamma \parallel^2 \\ X_k = \Gamma_k \quad \forall k \in [1, p] \end{cases}$$

One possible implementation is described by the block diagram in figure 2.6.

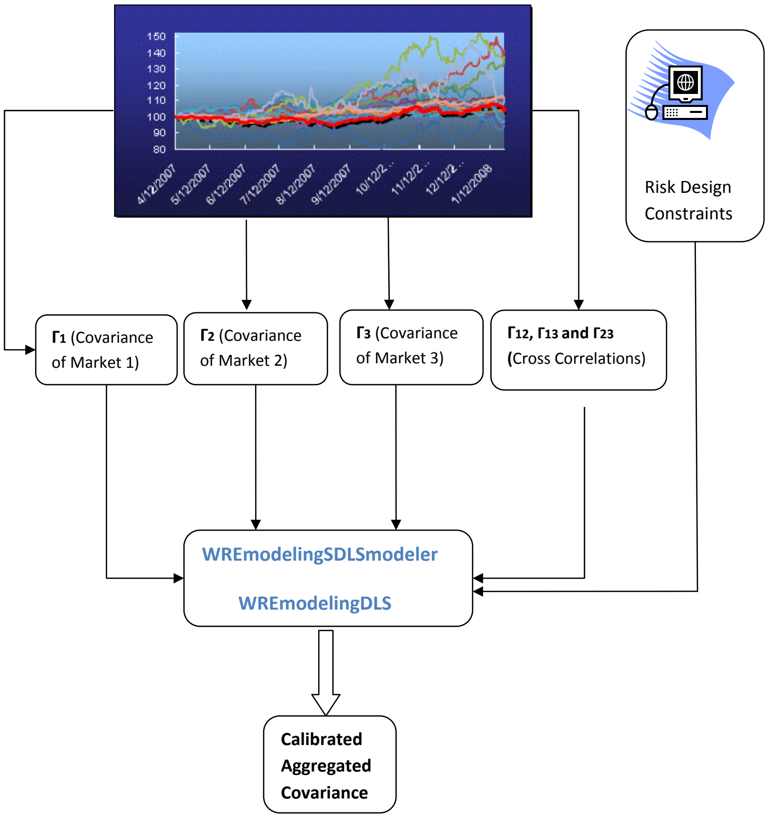


Figure 2.6: From local to global risk models

Practical example

In this example, we aggregate correlations from the American and the European stock markets, represented by a set of benchmarks (Eurostoxx50 and DJI), value indices (MSCI Europe and US value indices) and growth indices (MSCI Europe and US growth indices).

The initial block-built matrix is not semidefinite positive, because its smallest eigenvalue is negative:

	EU Market	EU Value	EU Growth	US Market	US Value	US Growth
EU Market	1	0,9548	0,9521	0,3189	0,4593	0,3272
EU Value	0,9548	1	0,8875	0,0958	0,2972	0,0575
EU Growth	0,9521	0,8875	1	0,3127	0,3845	0,4072
US Market	0,3189	0,0958	0,3127	1	0,9446	0,8473
US Value	0,4593	0,2972	0,3845	0,9446	1	0,8235
US Growth	0,3272	0,0575	0,4072	0,8473	0,8235	1

Figure 2.7: Initial correlation matrix

The SDP function of Wall Risk Engine® enables the ability to add any linear (equality or inequality) constraint. In the simulation below, we have forced the cross-correlations of the blue block in initial matrix to be point-wise greater than the local block represented in figure 2.8:

0,32	0,47
0,093	0,29

Figure 2.8: Inequality block constraint

With the SDP calibration function, we compute the semi definite positive matrix:

- which is the closest to the initial one,
- which preserves intra market blocks,
- which satisfies the required linear constraints.

The calibrated covariance matrix is given by:

	EU Market	EU Value	EU Growth	US Market	US Value	US Growth
EU Market	1,0000	0,9548	0,9521	0,3200	0,4700	0,3257
EU Value	0,9548	1,0000	0,8875	0,1051	0,2900	0,0676
EU Growth	0,9521	0,8875	1,0000	0,3054	0,3967	0,3992
US Market	0,3200	0,1051	0,3054	1,0000	0,9446	0,8473
US Value	0,4700	0,2900	0,3967	0,9446	1,0000	0,8235
US Growth	0,3257	0,0676	0,3992	0,8473	0,8235	1,0000

Figure 2.9: Corrected correlation matrix

The block constraint is respected, while other parts of the covariance matrix are calibrated so as to ensure the positive semi-definite property and meet the prescribed linear constraint at the same time.

Related functions

The related functions of Wall Risk Engine® are:

Function	Description
WREmodelingSDLSmodeler	Variance/covariance matrix calibration with equality and inequality constraints
WREmodelingStructuredFiltering	Variance/covariance matrix calibration with block constraints

2.3 Expected performance modeling

2.3.1 The key role of expected return models

Building an appropriate return model is one of the key steps of the allocation process, along with the risk model and the allocation model. Through the expression of expected performances on assets (or asset classes), the investment manager can feed his market anticipation into the allocation process and give a bias to the optimal allocation.

There are different ways of expressing views on the markets through the return model:

- Choosing the right estimation window for an historical trend approach: depending on the market conditions, how long should we go backwards in time to estimate the performance trend?
- Using an adaptive approach selecting the most relevant past information to compute the expected returns;
- Using explicit views, either directly as expected returns or through qualitative in-house grades (stars, discrete scale from -2 to 2, and others;)
- Using a view on the composition of the portfolio, and computing the corresponding implied returns: given the allocation strategy (Minimum-volatility or more complex strategies) and the risk model, what are the expected returns that would lead to this portfolio?

2.3.2 Momentum models

The performance of a portfolio is measured in terms of price changes. These variations can take a variety of forms such as absolute price change, relative price change, and log price change.

Simple net returns

Let NAV_t be the Net Asset Value of the asset at time t. The simple net return for the holding period $[t, t+h]$ is equal to:

$$R_t^h = \frac{\mathrm{NAV}_{t+h} - \mathrm{NAV}_t}{\mathrm{NAV}_t}$$

Continuously-compounded returns

Let NAV_t be the Net Asset Value of the asset at time t. The log-return function for the holding period $[t, t+h]$ is equal to:

$$LR_t^h = \log\left(\frac{\mathrm{NAV}_{t+h}}{\mathrm{NAV}_t}\right)$$

Related functions

The related functions of Wall Risk Engine® are:

Function	Description
WREmodelingReturns	Simple net returns (arithmetic)
WREmodelingReturnsLack	Simple net returns (arithmetic) with missing values
WREmodelingLogReturns	Log returns
WREmodelingLogReturnsLack	Log returns with missing values

2.3.3 Adaptive momentum model

Description of the adaptive approach

As introduced earlier in this chapter (Risk Modeling section), the adaptive momentum approach consists in **detecting automatically regime changes** on the market and computing the optimal historical weights associated to the past periods. It differs from standard approaches (equally or exponentially weighted estimations) in which the weights do not depend on market conditions.

Together with the adaptive risk model, this adaptive momentum model allows to build a dynamic investment strategy which is reactive with respect to regime changes on the market (see chapter 4).

Related functions

The related functions of Wall Risk Engine® are:

Function	Description
WREmodelingAdaptiveReturns	Adaptive performance model

2.3.4 Shrinkage methods for the estimation of expected returns

Shortfalls of the mean return estimator

When estimating the expected return of a single asset with no exogenous information, the mean return is proven to be the best estimate. But it has two major drawbacks:

- it is unstable;
- it is not the best estimate for the expected returns of a basket of assets.

The James-Stein estimator([BS90] and [Jor86]) is based on a shrinkage technique producing more robust[2] results. Furthermore, when estimating the returns of a basket of more than 3 financial instruments, it is proven that the JS model provides better estimates of the expected returns than the historical means.

In other words, the estimation error is reduced when estimating the expected returns of several assets at the same time with a shrinkage model than when estimating the expected returns of each asset independently.

Definition of the James-Stein model

Let

- X is a random variable (of dimension n) for which we want to estimate the expected value;
- $(X_i)_i$ for $i = 1...p$ observations of X with mean μ and covariance Γ;

[2] A statistical estimator is robust if minor changes in the inputs or in the model parameters do not impact the estimator significantly

- $\bar{\mu}$ the historical estimation of μ and L_2 the mean quadratic error of an estimate $\hat{\mu}$ of μ;
- L_2 the mean quadratic error of an estimator $\hat{\mu}$ of μ: $L_2(\mu;\hat{\mu}) = E_\mu ||\mu - \hat{\mu}||_2$;
- μ^0 a reference vector of dimension n.

The James-Stein estimator allows to compute an estimator of a vector μ of expected returns for a portfolio under the assumption that the mean vector of returns is normally iid distributed with known covariance matrix Σ, i.e. following the normal law $N(\bar{m}, \Sigma)$.

Let $\bar{r}$ be the sample mean obtained from a number T of observations of the returns

$$\bar{r} = \frac{1}{n}\sum_{i=1}^{T} r_i$$

For a given reference vector r_0, the James-Stein estimator is defined as a shrinkage of $\bar{r}$ by the formula:

$$\hat{\mu}(r) = (1-\omega)\bar{r} - \omega r_0 \bar{1}$$

where $\bar{1}$ is a vector of ones and ω is defined as:

$$\omega = \min\left[1, \frac{(n-2)/T}{(\bar{r} - r_0\bar{1})'\Sigma^{-1}(\bar{r} - r_0\bar{1})}\right].$$

Roughly speaking, the James-Stein estimator achieves uniformly lower risk than the Maximum Likelihood estimator even though it allows some increased risk for some components of the $\bar{r}$ vector.

Related functions

The related functions of Wall Risk Engine® are:

Function	Description
WREmodelingShrinkMeanJS	James-Stein shrinkage estimation of the expected returns

2.3.5 Predictive autoregressive signals (PAS)

Description of the PAS model

The momentum models expect the future returns to carry on their evolution based on the current trend. The autoregressive model tends to offset the momentum signals: if the returns of the asset have been recently above (resp. below) its average return, then the autoregressive model predicts expected returns to be below (resp. above) the average in the near future.

In the multidimensional autoregressive model, the expected prices of the n assets at time t can be deduced from their prices at time $t-1$ according to the following formula:

$$S_t = AS_{t-1} + Z_t$$

where

- S_t is the vector of prices of the n assets at time t;
- A is a $n \times n$ matrix (to be calibrated);
- Z_t is a 'noise' vector.

Wall Risk Engine® embeds a function to compute the autoregressive expected returns given a time horizon. These expected returns are obtained by simulating the process described above on the given future period.

Robust calibration of the PAS model

Calibrating the matrix A is a sensitive step. A standard estimate for A is the matrix that minimizes the distance between the price estimations implied by the model and the T observed market prices during a past period:

$$A = \arg\min_{A \in \mathbb{R}^{n \times n}} J(A)$$

where

$$J(A) = \sum_{t=1}^{T} ||S_t - AS_{t-1}||^2$$

This approach has two major drawbacks:

- The spectral radius $\rho(A)$(absolute value of the largest eigenvalue) can diverge if A is ill-conditionned (when using few observation points for the calibration);
- Negative prices can occur in this model!

As a consequence, we formulate the robust counterpart of the calibration problem as follows:

$$\begin{cases} \min_{A \in \mathbb{R}^{n \times n}} J(A) \\ A \in \mathbb{R}_+^{n \times n} \\ \rho(A) \leq \bar{\rho} \end{cases}$$

where $\bar{\rho}$ is the return of a virtual portfolio that would invest each day 100% on the best performing asset.

Combining PAS and momentum signals

Using autoregressive signals as such in the return model to offset the momentum views can be sensitive for several reasons:

- these simulated returns are unstable;
- their order of magnitude differs from the momentum signals so it is not straightforward to build a mixed momentum/autoregressive signal.

To build a mixed momentum/autoregressive portfolio in practise and get around the drawbacks mentioned above, we help our clients (through our RP Quant Advisory missions) to proceed as follows:

- **Step 1**: compute autoregressive weight signals, namely the output of an unconstrained long-short optimization using the autoregressive expected returns. These weight signals take into account cross-assets correlations and embed hedging information: for instance, it is possible to get a negative weight on an asset with a positive expected return, because this asset contributes to the diversification of the portfolio. As a consequence, these autoregressive weight signals are more stable through time and embed richer information than the autoregressive simulated returns alone.

- **Step 2**: compare these weights to the momentum-driven constrained weights, obtained by a constrained optimization based on the momentum expected returns. The order of magnitude of the two signals (autoregressive weight signals and momentum weight signals) is the same, hence it is natural to compare them.

In practice, the autoregressive weight signals can be used to reinforce or decrease the momentum-driven weights. We define a specific momentum/PAS combination process with each client to meet his specific needs.

Related functions

The related functions of Wall Risk Engine® are:

Function	Description
WREmodelingPASreturns	Predictive Autoregressive Signals for the expected returns

2.3.6 Implied returns

The fact that asset allocation methods are highly sensitive to changes in the input expected returns is well known by investment practitioners. An alternative approach is therefore to turn the problem around: instead of starting with a set of expected returns and computing optimal weights, implied returns are extracted from a given initial portfolio structure.

Given as inputs an initial portfolio (the Market Portfolio or any initial portfolio provided by the investor), the covariances between the assets and the Risk-free rate, the implied returns are computed by solving a reverse engineering problem for the Markowitz Allocation Model. These implied returns can be used as return forecasts for the portfolio optimization:

$$r^i_{implied} = riskAversion * \sum_{j=1}^{n} \Gamma(i,j) * \omega(j)$$

where:

- $riskAversion$ is the risk aversion coefficient, i.e. the marginal rate at which an investor is willing to sacrifice expected return in order to lower variance by one unit,

- $\Gamma(i,j)$ is the covariance between assets i and j,

- $\omega(j)$ is the weight of asset j in the input portfolio. In case the input portfolio is the market portfolio, $\omega(j)$ is the ratio of the market capitalization of asset j divided by the total market capitalization.

Investors, however, often have their own views about how the market is going to behave in the future. By comparing the implied returns with the explicit expected returns that the investors might have, the implied returns can be adapted in an iterative way and plugged into a rebalancing method.

Related functions

The related functions of Wall Risk Engine® are:

Function	Description
WREmodelingImpliedReturns	Implied returns

2.4 Robust Factorial Models

Wall Risk Engine® embeds regression functions allowing you to perform a robust regression on any exogenous factors, given a set of bounds constraints. The linear regression model is defined as:

$$Y_i = \beta^\top X_i + \epsilon_i$$

where

- β is a vector of parameters to be estimated given a set of bound constraints,
- Y_i and X_i are the vectors of the i^{th} observation,
- the errors ϵ_i are assumed to be normally and independently distributed with mean 0 and constant variance.

For instance, the linear regression can be used to regress asset prices on several macro-economic variables as in the **Arbitrage Pricing Theory**, developed by Ross in [Ros76]. The Wall Risk Engine® regression method embeds a matrix calibration to ensure the robustness of the regression coefficients.

Related functions

The related functions of Wall Risk Engine® are:

Function	Description
WREmodelingAPT	Robust Linear Regression
WREmodelingAPTconstraint	Constrained Linear Regression

2.5 Higher order moments modeling

The concept of a covariance matrix can be extended to higher moments such as the skewness and the kurtosis.

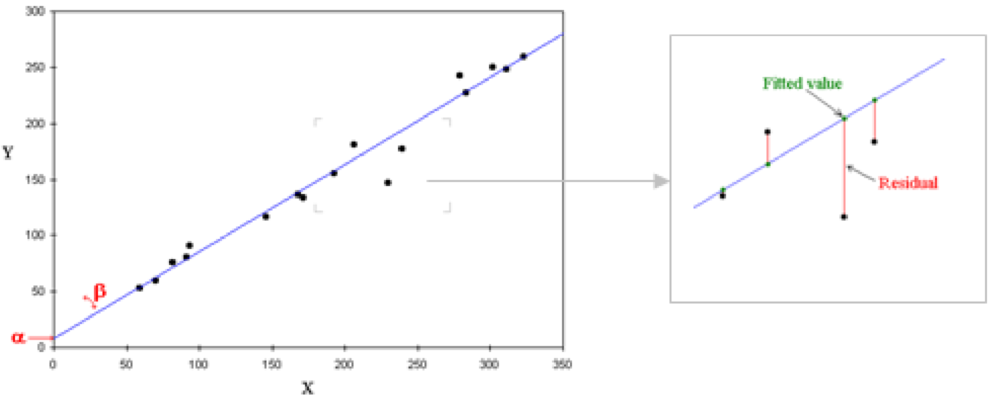


Figure 2.10: Illustration of an unconstrained linear regression of Y on X

2.5.1 Co-Kurtosis

Definition

The co-kurtosis is a statistical measure that calculates the peakedness of a variable's probability distribution in relation to another variable's peakedness. Everything else being equal, a higher co-kurtosis means that the first variable has a flatter probability distribution.

Co-kurtosis can be used as a supplement to the covariance calculation of risk estimation. The co-kurtosis can be calculated using an asset's historic price data as the first variable, and the market's historic price data as the second. For a risk-averse investor, a lower co-kurtosis is preferred, as the security's returns would not be much different from the market's returns (i.e. low beta).

Related functions

The related functions of Wall Risk Engine® are:

Function	Description
WREmodelingCoKurtosis	Co-kurtosis matrix

2.5.2 Co-Skewness

Definition

The co-skewness is a statistical measure that calculates the asymmetry of a variable's probability distribution relative to another variable's probability distribution asymmetry. Everything else being equal, a positive co-skewness means that the first variable's probability distribution is skewed to the right of the second variable's distribution.

Co-skewness can be used as a supplement to the covariance calculation of risk estimation. The co-skewness can be calculated using an asset's historic price data as the first variable, and the market's historic price data as the second. This provides an estimation of the asset's risk in relation to market risk.Investors prefer a positive coskewness because this represents a higher probability of extreme positive returns for the asset over market returns.

Related functions

The related functions of Wall Risk Engine® are:

Function	Description
WREmodelingCoSkewness	Co-skewness matrix

3 Risk and Performance Analysis

3.1 Overview

3.1.1 Advanced ex-ante and ex-post measures

The first step to a coherent and reliable risk or performance measure is the robust estimation of the variance-covariance matrix of the underlying assets.

All the functionalities of Wall Risk Engine® are based on a patented method for the estimation of interdependences, which ensures the reliability of the risk measures provided in this module and reduce the ex-ante/ex-post gap (cf Risk and Performance Modeling module to know more about the Wall Risk Engine® covariance matrix calibration method).

The Risk and Performance Analysis module provides a wide range of state-of-the-art functionalities designed for a tailored ex-ante and ex-post analysis of a portfolio:

- Variance and volatility measures: absolute and relative measures of the dispersion,
- Moments of the distribution: advanced analysis of the distribution,
- VaR measures : extreme risk measures for gaussian and non-gaussian returns,
- Loss and Drawdown measures,
- Performance measures,
- Performance to risk ratios,
- Risk and performance decomposition,
- Concentration and diversification measures.

This series of functions provides ex-ante and ex-post risk and performance measures of a portfolio.

On the one hand, the ex-ante measures provide an estimation of the risk/return profile of the portfolio based on the historical estimation of:

- the expected returns of the underlying assets,

- the variance-covariance matrix of the underlying assets.

On the other hand, the ex-post measures of a portfolio are based on the realized performances of the portfolio itself.

Note that for the ex-post measures including both performance and risk measures, the estimation window for the risk measures and the mean returns can either be the same (for the standard functions such as *WREanalysisExpostSharpeRatio*) or different (for the expert functions such as *WREanalysisExpostSharpeRatioX*). For instance, in the "expert" version of the Sharpe ratio, the ex-post volatility might be estimated on a 6-month estimation window, while the ex-post mean return might be estimated over the past 3 months.

3.1.2 Notations

In this section, we will refer to the following notations:

- ω is the vector of the weights of the portfolio,
- Γ is the variance-covariance matrix of the underlying assets,
- ρ is the vector of performances of the underlying assets (for the given horizon),
- Φ is the cumulative distribution function (CDF) of a standard normal random variable,
- r_f is the risk-free rate (for the given horizon),
- n is the size of the estimation window (number of historical data),
- $r_p(t)$for $t = 1, ..., n$ is the portfolio's historical returns,
- $r_p = (r_p(1), ..., r_p(n))$ is the vector of portfolio's ex-post returns (for n dates),
- For a given vector or matrix A,$A^\top$ is the transpose of A,
- $\bar{X}$ is the mean of the vector X over the period $t = 1, ..., n$,
- $\sigma^2(X)$ is the unbiased statistical estimation of the variance of the vector $X = (X(1), ..., X(n))$:

$$\sigma^2(X) = \frac{1}{n-1}\sum_{t=1}^{n}(X(t) - \bar{X})^2$$

3.2 Performance measures

The performance of a portfolio (or an asset) is measured in terms of mean return.
The ex-ante mean return of a portfolio is defined as:

$$\text{MeanReturn}_{\text{ex-ante}} = \rho^\top \omega$$

while the ex-post return of a portfolio (or any asset) over the period $t = 1, ..., n$ is given by:

$$\text{MeanReturn}_{\text{ex-post}} = \frac{1}{n}\sum_{t=1}^{n} r(t)$$

3.3 Ex-ante Risk

The dispersion of the returns (absolute, or relative with respect to a benchmark) is the simplest measure of the risk of a portfolio.

3.3.1 Variance

For a given time horizon, the ex-ante variance of a portfolio is given by:

$$\text{Variance}_{\text{ex-ante}} = \omega^\top \Gamma \omega$$

The unbiased ex-post variance of a portfolio is given by:

$$\text{Variance}_{\text{ex-post}} = \sigma^2\left(r_p\right) = \frac{1}{n-1}\sum_{t=1}^{n}\left(r_p(t) - \bar{r_p}\right)^2$$

3.3.2 Volatility

The ex-ante volatility (also referred to as standard deviation) is defined by:

$$\text{Volatility}_{\text{ex-ante}} = \sqrt{\omega^\top \Gamma \omega}$$

The ex-post volatility of a portfolio is defined as:

$$\text{Volatility}_{\text{ex-post}} = \sqrt{\text{Variance}_{\text{ex-post}}}$$

3.3.3 Tracking Error

The tracking error enables the comparison between the behaviors of a portfolio and a given benchmark. The lower the tracking error, the closer the portfolio and the benchmark are in terms of returns, so it is a pertinent risk measure for benchmarked allocation strategy.
The ex-ante tracking error is defined as the ex-ante standard deviation of the differences between the returns of the portfolio and the returns of the benchmark:

$$\text{TrackingError}_{\text{ex-ante}} = \sqrt{\begin{bmatrix}\omega^\top & -1\end{bmatrix}\begin{bmatrix}\Gamma & c_1 \\ {c_1}^\top & c_2\end{bmatrix}\begin{bmatrix}\omega^\top & -1\end{bmatrix}}$$

where c_1 is the vector of covariances between the components of the portfolio and the benchmark, and c_2 is the variance of the benchmark. The ex-post formulation of the tracking error is the actual tracking error over a past period. It is defined as the standard deviation of the difference between the return of the portfolio and the return of the benchmark:

$$\text{TrackingError}_{\text{ex-post}} = \sqrt{\sigma^2\left(r_p - r_b\right)}$$

where $r_b = (r_b(1), ..., r_b(n))$ is the vector of the benchmark's historical returns.

3.3.4 Downside Risk

The downside deviation isolates the negative portion of the volatility, i.e. the volatility of the returns that are smaller than a given threshold: $r_p(t) < \mu$. This risk measure is particularly interesting when dealing with sophisticated financial instruments that have an asymmetric distribution, such as guaranteed capital products.

$$\text{DownsideRisk}_{\text{ex-post}} = \sqrt{\frac{1}{m-1}\sum_{t=1}^{n}\left[(r_p(t)-\mu)^2\,\chi_{]-\infty,\mu]}\,(r_p(t))\right]}$$

where m is the number of dates between 1 and n such that $r_p(t) < \mu$. For $\mu = \bar{r_p}$, we get the downside volatility.

3.3.5 Definition of the (relative) Max Drawdown

$r^k_{i,j} = \frac{S^k_j - S^k_i}{S^k_i}$ is the arithmetic return from i to j for asset k. $r_{i,j}$ is the vector of returns of all assets. $S_i(\omega)$: price of the portfolio for ω fixed (see WREanalysisFutureValues)

$$DDmax(\omega) = \max_{i<j}\left(\frac{S_i(\omega) - S_j(\omega)}{S_i(\omega)}, 0\right) = \max_{i<j}(-r^t_{j,i}\omega, 0)$$

We obviously have :

$$0 \leqslant DDmax(\omega) \leqslant 1$$

3.3.6 Related Functions

The related functions of Wall Risk Engine® are:

Function	Description
WREanalysisExanteTE	Ex-ante tracking error
WREanalysisExanteVolatility	Ex-ante volatility
WREanalysisExpostDownside	Ex-post downside risk
WREanalysisExpostDownsideX	Ex-post downside risk with different estimation windows for risk and return
WREanalysisExpostSemiVolatility	Ex-post semi-volatility
WREanalysisExpostSemiVolatilityX	Ex-post semi-volatility with different estimation windows for risk and return
WREanalysisExpostTE	Ex-post tracking error
WREanalysisExpostTEX	Ex-post tracking error with different estimation windows for risk and return
WREanalysisExpostVolatility	Ex-post volatility
WREanalysisExpostVolatilityX	Ex-post volatility with different estimation windows for risk and return
WREanalysisExpostVolatilityExp	Ex-post exponentially weighted volatility
WREanalysisExpostVolatilityExpX	Ex-post exponentially weighted volatility with different estimation windows for risk and return

3.4 Advanced analysis of the distribution

When dealing with non-normally distributed returns, the 2 first moments (mean and variance) do not allow the capture of the shape of the distribution.

k^{th} central moment

For a given k, this function computes the k^{th} moment of a distribution $P(x)$ and is defined by:

$$\mu_k = E\left[(X - E(X))^k\right] = \int_\Omega (x-\mu)^k P(x)dx$$

3.4.1 Skewness (Asymmetry)

The skewness characterizes the degree of asymmetry of a distribution around its mean.
Note that:

- Gaussian returns have a 0-skewness.
- Negative (respectively positive) skewness implies asymmetry toward negative values (resp. positive) and indicates that the distribution tail is fatter for negative (resp. positive) values.
- A risk-averse investor does not like the negative skewness.

The skewness of a distribution is the normalized form of the third moment of the distribution. The skewness is therefore defined by:

$$\text{Skewness} = \frac{\mu_3}{(\mu_2)^{\frac{3}{2}}}$$

where μ_i is the i^{th} central moment of the distribution of returns.

3.4.2 Excess Kurtosis (Peakedness)

The kurtosis characterizes the peakedness or flatness of a given distribution and is defined as a normalized form of the fourth central moment of a distribution:

$$\text{Kurtosis} = \frac{\mu_4}{(\mu_2)^2}$$

where μ_i is the i^{th} central moment of the distribution of returns.
For a standard normal distribution,

$$\frac{\mu_4}{(\mu_2)^2} = 3$$

For this reason, we define the excess kurtosis as the relative peakedness or flatness of a given distribution compared to a normal distribution. The excess kurtosis is therefore defined by:

$$\text{ExcessKurtosis} = \frac{\mu_4}{(\mu_2)^2} - 3$$

so that the normal distribution has an excess kurtosis of 0.
When the excess kurtosis is positive the distribution is called leptokurtic, when negative it is referred as platykurtic, when null it is called mesokurtic.

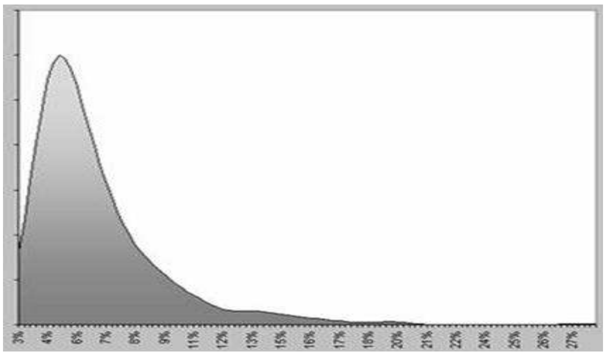


Figure 3.1: Non parametric estimation of probability density function

3.4.3 Density Estimation with Non Parametric Gaussian Kernel

This function computes the non parametric estimation of the distribution of returns by the Gaussian kernel method [MS05]:

$$f(y) = \frac{1}{nh} \sum_{t=1}^{n} K \left[\frac{y - r(t)}{h} \right]$$

where n is the number of returns (simulated or historical), h is the width of the estimation step, $r(t)$ is the t^{th} return, and K is the Gaussian kernel defined as:

$$K(u) = \frac{1}{\sqrt{2\pi}} \exp^{-\frac{u^2}{2}}$$

For example, consider a structured product with only 100 simulated returns for a ten-year maturity. The Gaussian kernel function produces the following smooth estimation of the distribution:

3.4.4 Related Functions

Function	Description
WREanalysisExanteKurtosis	Ex-ante kurtosis
WREanalysisExanteSkewness	Ex-ante skewness
WREanalysisExpostKurtosis	Ex-post kurtosis
WREanalysisExpostKurtosisX	Ex-post kurtosis with different estimation windows for risk and return
WREanalysisExpostSkewness	Ex-post skewness
WREanalysisExpostSkewnessX	Ex-post skewness with different estimation windows for risk and return
WREanalysisMoment	Estimation of the k^{th} moment $(k = 1, 2, 3, 4)$
WREanalysisGaussianKernel	Densitity estimation with Gaussian kernel method

3.4.5 VaR measures

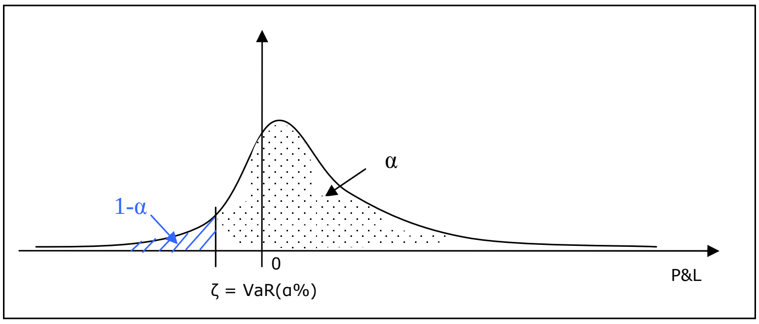


Figure 3.2: Illustration of the VaR on a distribution of PnLs

The Value-at-Risk (VaR) of a portfolio at the probability level α (VaR_α) is the lowest possible value such that the probability of achieving a return greater than VaR_α exceeds α:

$$\mathrm{VaR}_\alpha = \max\left\{\zeta | \mathrm{prob}(r > \zeta) \geq \alpha\right\}$$

where r is the return of the considered portfolio and $\alpha \in [0, 1]$.

Wall Risk Engine® provides 6 measures of VaR.

VaR measure	When to use it
Gaussian VaR	For normally-distributed returns
Conditional VaR	To measure extreme losses beyond VaR and work with a coherent risk measure
Max VaR	For a conservative upper bound of the VaR of non gaussian returns, when no information is available on the higher moments of the distribution
Modified VaR	For non Gaussian returns, when higher moments of the distribution (skewness and kurtosis) are known
Non parametric VaR	When no a priori information is available on the shape of the distribution
Delta-Gamma VaR	This function computes the short-term VaR of a portfolio containing non-linear instruments for which the greeks are known

These VaR can be estimated using:

- A **parametric** approach (ex-ante formulation),
- An **historical** approach (ex-post formulation),
- A **Monte-Carlo** approach (ex-post formulation, along with the simulation functionalities provided the Simulation Module of Wall Risk Engine®).

NB: The VaR measures computed in Wall Risk Engine® are expressed as returns. As a consequence, when negative, the VaR is to be interpreted as a risk of loss, and when positive it is a risk of limited gain.

Gaussian VaR

For normally-distributed returns, the ex-ante Gaussian VaR (for the probability level α) is given by:

$$\mathrm{NVaR}_\alpha^{ex-ante} = \rho^\top \omega - \Phi^{-1}(\alpha)\sqrt{\omega^\top \Gamma \omega}$$

and the ex-post Gaussian Value-at-Risk is given by:

$$\mathrm{NVaR}_\alpha^{ex-post} = \frac{1}{n}\sum_{t=1}^{n} r_p(t) - \sqrt{\frac{1}{n-1}\sum_{t=1}^{n}[r_p(t)-\bar{r_p}]^2}\Phi^{-1}(\alpha)$$

Gaussian Conditional VaR

An alternative measure of expected losses, with more attractive properties than the Value-at-Risk, is the Tail Value-at-Risk, which is also called Mean Excess Loss, Mean Shortfall, or Conditional VaR. The conditional Value at Risk is defined as the expected amount of loss below the VaR, which is why we always have:

$$\mathrm{NCVaR}_\alpha \leq \mathrm{NVaR}_\alpha$$

The **ex-ante Normal Conditional VaR** is given by:

$$\mathrm{NCVaR}_\alpha^{ex-ante} = \rho^\top \omega - \kappa(\alpha)\sqrt{\omega^\top \Gamma \omega}$$

where

$$\kappa(\alpha) = \left[(1-\alpha)\sqrt{2\pi}\exp^{\frac{\Phi^{-1}(\alpha)^2}{2}}\right]^{-1}$$

The Conditional Value at Risk has nicer properties than the standard Value at Risk:

- It is a coherent measure of risk (translation-invariant, sub-additive, positively homogeneous, monotonic with respect to the stochastic dominance),
- It is convex with respect to the portfolio positions ω.

The **ex-post Gaussian Conditional Value-at-Risk** is defined by:

$$\mathrm{NCVaR}_\alpha^{ex-post} = \frac{1}{n}\sum_{t=1}^{n} r_p(t) - \sqrt{\frac{1}{n-1}\sum_{t=1}^{n}[r_p(t)-\bar{r_p}]^2}\kappa(\alpha).$$

Max VaR

We can compute a lower bound of the Value-at-Risk based on the sole knowledge of the first two moments of the distribution. This lower bound provides a robust estimation of the ex-ante Value-at-Risk for non-gaussian returns.

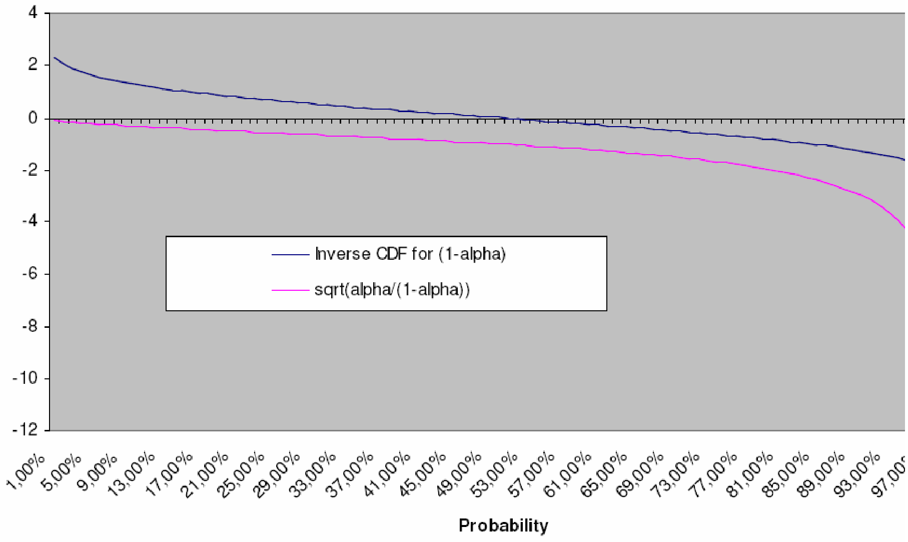


Figure 3.3: Inverse cumulative distribution function and its lower bound

Following Bertsimas and Popescu (2000), we can define the following lower bounds for ex-ante and ex-post Value-at-Risk as :

$$\mathrm{MaxVaR}_{\alpha}^{ex-ante} = \rho^{\top}\omega - \sqrt{\frac{\alpha}{1-\alpha}}\sqrt{\omega^{\top}\Gamma\omega}$$

and

$$\mathrm{MaxVaR}_{\alpha}^{ex-post} = \frac{1}{n}\sum_{t=1}^{n} r_p(t) - \sqrt{\frac{1}{n-1}\sum_{t=1}^{n}\left[r_p(t) - \bar{r_p}\right]^2}\sqrt{\frac{\alpha}{1-\alpha}}.$$

This bound is illustrated in the figure 3.3.

Modified VaR

When dealing with non Gaussian returns but with available information on the higher moments of the distribution, Wall Risk Engine® provides a modified version of the Value-at-Risk, based on a result from [CF37].

The modified VaR adjusts the traditional Gaussian VaR with the skewness and the kurtosis of the distribution. The modified VaR allows to compute an improved measure of the Value-at-Risk for distributions with asymmetry (positive or negative skewness) and fat tails (positive excess kurtosis).

The ex-ante modified VaR is defined as:

$$\mathrm{ModifiedVaR}_{\alpha}^{ex-ante} = \rho^{\top}\omega - \kappa(\alpha)\sqrt{\omega^{\top}\Gamma\omega}$$

where

$$\begin{aligned}\kappa(\alpha) = \Phi^{-1}(\alpha) &+ \frac{1}{6}\left[\left(\Phi^{-1}(\alpha)\right)^2 - 1\right] && \text{Skewness}_{\text{ex-ante}}\\ &+ \frac{1}{24}\left[\left(\Phi^{-1}(\alpha)\right)^3 - 3\Phi^{-1}(\alpha)\right] && \text{ExcessKurtosis}_{\text{ex-ante}}\\ &+ \frac{1}{36}\left[-2\left(\Phi^{-1}(\alpha)\right)^3 + 5\Phi^{-1}(\alpha)\right] && \text{Skewness}^2_{\text{ex-ante}}\end{aligned}$$

The ex-post modified VaR is defined as:

$$\text{ModifiedVaR}_{\alpha}^{ex-post} = \frac{1}{n}\sum_{t=1}^{n} r_p(t) - \kappa(\alpha)\sqrt{\frac{1}{n-1}\sum_{t=1}^{n}\left[r_p(t) - \bar{r_p}\right]^2}.$$

where

$$\begin{aligned}\kappa(\alpha) = \Phi^{-1}(\alpha) &+ \frac{1}{6}\left[\left(\Phi^{-1}(\alpha)\right)^2 - 1\right] && \text{Skewness}_{\text{ex-post}}\\ &+ \frac{1}{24}\left[\left(\Phi^{-1}(\alpha)\right)^3 - 3\Phi^{-1}(\alpha)\right] && \text{ExcessKurtosis}_{\text{ex-post}}\\ &+ \frac{1}{36}\left[-2\left(\Phi^{-1}(\alpha)\right)^3 + 5\Phi^{-1}(\alpha)\right] && \text{Skewness}^2_{\text{ex-post}}\end{aligned}$$

Note that:

- If the distribution is normal, the skewness and the excess kurtosis are equal to zero, which makes $\kappa(\alpha)$ to be equal to $\Phi^{-1}(\alpha)$.
- If a portfolio (especially a portfolio composed of hedge funds, private equity, technology stocks, emerging markets stocks,...) has a negative skewness and/or a positive excess kurtosis, the modified VaR will be higher than Gaussian Value-at-Risk.
- For a portfolio with negative skewness and/or a negative excess kurtosis the Modified VaR will be lower than Gaussian Value-at-Risk.

Non parametric VaR

When no information is available on the moments of the distribution, it is still possible to have a precise approximation of the VaR. The non parametric Value-at-Risk is based on the estimation of a distribution of returns by the non-parametric kernel method. We use a Gaussian kernel to estimate the density function:

$$f(y) = \frac{1}{nh}\sum_{t=1}^{n} K\left[\frac{y - r_p(t)}{h}\right]$$

where h is the width of the estimation step and K is the Gaussian kernel:

$$K(u) = \frac{1}{\sqrt{2\pi}}\exp^{-\frac{u^2}{2}}$$

The non parametric VaR is defined by:

$$\text{NPVaR}_{\alpha} = \max\left\{\zeta | \text{prob}(r > \zeta) \geq \alpha\right\}$$

where the probability is given by:

$$\text{prob}(r > \zeta) = \int_{\zeta}^{+\infty} f(u)du$$

Non parametric CVaR

Similarly to the non parametric VaR, the non parametric conditional Value-at-Risk is based on the estimation of a distribution of returns by the non-parametric kernel method, using a Gaussian kernel to estimate the density function.

Delta-Gamma VaR

This function computes the short-term VaR of a portfolio containing non-linear instruments for which the greeks [ZKR09] are known.

For a single instrument
Let

- S_t be the value of a basket of linear instruments at time t
- $V(S_t, t)$ be the time-t value of a non-linear instrument based on this underlying basket
- θ measures the sensitivity of the derivative price with respect to the value of the value of the underlying portfolio
- δ measures the sensitivity of the instrument price with respect to the price of its underlying: it is the derivative of the option price with respect to the underlying price.
- Γ measures the sensitivity of the delta measure with respect to changes in the underlying price. In other words, Gamma is the second derivative of the instrument price with respect to the underlying price.

This approach consists in approximating the value of a derivative instrument at time t by a Taylor expansion of the value of this derivative at time 0:

$$V(S_t, t) = V(S_0, 0) + \theta_t + \delta(S_t - S_0) + \frac{1}{2}(S_t - S_0)'\Gamma(S_t - S_0) + o()$$

Then we derive from this expansion a formulation for the returns:

$$\frac{V(S_t, t) - V(S_0, 0)}{V(S_0, 0)} = \frac{\theta_t}{V(S_0, 0)} + \frac{1}{2}\frac{(S_t - S_0)'\Gamma(S_t - S_0)}{V(S_0, 0)}$$

$$r_t = \frac{\theta t}{V(S_0, 0)} + \frac{\Delta^T diag(S_0)\xi_t}{V(S_0, 0)} + \frac{1}{2}\frac{\xi_t^T diag(S_0)\Gamma\xi_t diag(S_0)}{V(S_0, 0)}$$

Let's define the following relative values for the greeks:

$$\bar{\theta} = \frac{\theta t}{V(S_0, 0)}$$

$$\bar{\Delta} = \frac{\Delta diag(S_0)}{V(S_0, 0)}$$

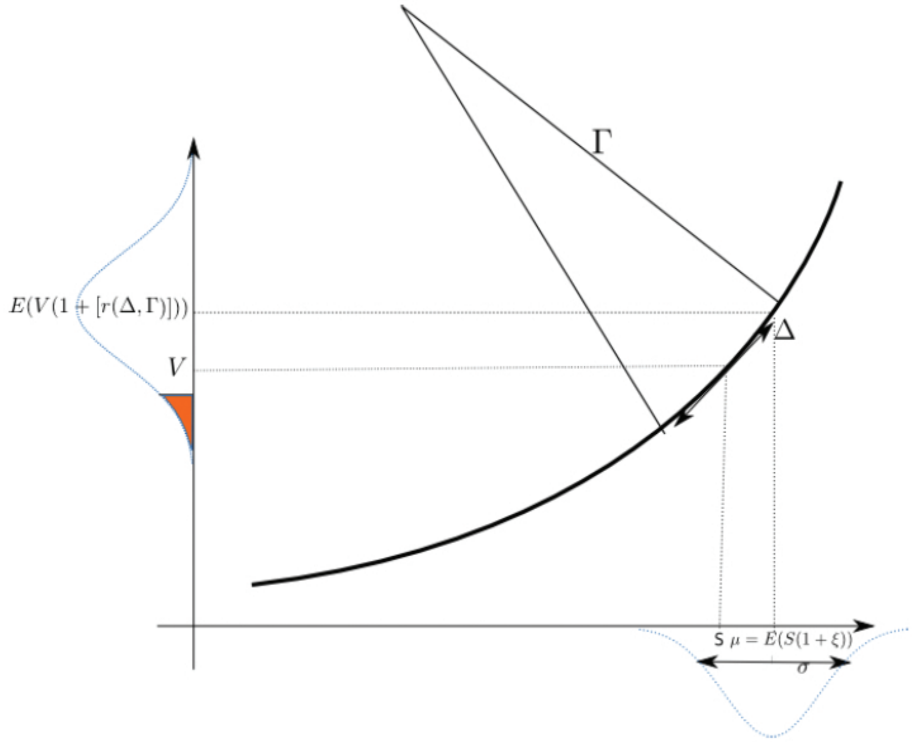


Figure 3.4: Delta Gamma VaR

$$\bar{\Gamma} = \frac{diag(S_0)\Gamma}{V(S_0, 0)}$$

Then we can write :

$$r_t = \bar{\theta} + \bar{\Delta}^T \xi_t + \frac{1}{2}\xi_t^T \bar{\Gamma} \xi_t$$

Hence the 2 first moments of the r_t variable are:

$$M_1 = E[r_t] = \bar{\theta} + \bar{\Delta}^T E[\xi_t] + \frac{1}{2}\langle \bar{\Gamma}; E[\xi_t \xi_t^T]\rangle$$

$$\begin{aligned} M_2 = E[r_t^2] \quad = \quad & \bar{\theta}^2 + \langle \bar{\Delta}\bar{\Delta}^T; E[\xi_t \xi_t^T]\rangle + \frac{1}{4}\langle \bar{\Gamma} \otimes \bar{\Gamma}; E[\xi_t^{\otimes 4}]\rangle \\ & + 2\bar{\theta}\bar{\Delta}^T E[\xi_t] + \bar{\theta}\langle \bar{\Gamma}; E[\xi_t \xi_t^T]\rangle + \langle \bar{\Gamma} \otimes \bar{\Delta}; E[\xi_t^{\otimes 3}]\rangle \end{aligned}$$

For a portfolio of instruments
We can generalize the previous formulation for a portfolio of m products with the same set of underlying instruments. Let

- ω (size m) the vector of weights of the portfolio;
- r_t the vector of size m containing the returns of the non linear instrument at time t;
- $\bar{\Delta}$ a $n \times m$ matrix containing the sensitivities of the prices of the portfolio components with respect to the underlying prices;

- $\bar{\Gamma}$ a $n \times n \times m$ array containing the second order derivatives of the prices of the portfolio components with respect to the underlying prices;
- $\bar{\Delta}(\omega) = \sum_i \omega_i \bar{\Delta}_i$ where $\bar{\Delta}_i$ is the i^{th} column of $\bar{\Delta}$;
- $\bar{\Gamma}(\omega) = \sum_i \omega_i \bar{\Gamma}_i$ where $\bar{\Gamma}_i$ is the i^{th} slice ($n \times n$ matrix) of $\bar{\Gamma}$.

We can write the return of the portfolio as :

$$r_t^T \omega = \bar{\theta}^T \omega + \bar{\Delta}(\omega)^T \xi_t + \frac{1}{2} \xi_t^T \bar{\Gamma}(\omega) \xi_t$$

The first and second-order moments for the returns of the portfolio can be written as:

$$M_1 = E[r_t^T \omega] = \bar{\theta}^T \omega + \bar{\Delta}(\omega)^T E[\xi_t] + \frac{1}{2} \langle \bar{\Gamma}(\omega); E[\xi_t \xi_t^T] \rangle$$

$$\begin{aligned} M_2 = E[(r_t^T \omega)^2] \quad = \quad & (\bar{\theta}^T \omega)^2 + \langle \bar{\Delta}(\omega) \bar{\Delta}(\omega)^T; E[\xi_t \xi_t^T] \rangle + \frac{1}{4} \langle \bar{\Gamma}(\omega) \otimes \bar{\Gamma}(\omega); E[\xi_t^{\otimes 4}] \rangle \\ & + 2\bar{\theta}^T \omega \bar{\Delta}(\omega)^T E[\xi_t] + \bar{\theta}^T \omega \langle \bar{\Gamma}(\omega); E[\xi_t \xi_t^T] \rangle + \langle \bar{\Gamma}(\omega) \otimes \bar{\Delta}(\omega); E[\xi_t^{\otimes 3}] \rangle \end{aligned}$$

Formulation of the Delta-Gamma VaR
Finally, for a given confidence level α, the ex-ante Delta-Gamma Value-at-Risk with a Gaussian hypothesis on the distribution of returns is given by:

$$\text{NDeltaGammaVaR}_\alpha^{ex-ante} = M_1 - \Phi^{-1}(\alpha) \sqrt{M_2 - M_1^2}$$

where Φ is the cumulative distribution function (CDF) of a standard normal random variable.

The robust version with no Gaussian hypothesis is defined as:

$$\text{RDeltaGammaVaR}_\alpha^{ex-ante} = M_1 - \kappa(\alpha) \sqrt{M_2 - M_1^2}$$

where

$$\kappa = \sqrt{\frac{\alpha}{1-\alpha}}$$

Related functions

Function	Description
WREanalysisExanteMaxVaR	Ex-ante upper bound of the VaR
WREanalysisExanteModifiedVaR	Ex-ante Modified VaR
WREanalysisExanteNormalCVaR	Ex-ante Gaussian Conditional VaR
WREanalysisExanteNormalVaR	Ex-ante Gaussian VaR
WREanalysisExpostMaxVaR	Ex-post upper bound of the VaR
WREanalysisExpostMaxVaRX	Ex-post upper bound of the VaR with different estimation windows for risk and return
WREanalysisExpostModifiedVaR	Ex-post Modified VaR
WREanalysisExpostModifiedVaRX	Ex-post Modified VaR with different estimation windows for risk and return
WREanalysisExpostNonParametricCVaR	Ex-post non parametric CVaR
WREanalysisExpostNonParametricCVaRX	Ex-post non parametric CVaR with different estimation windows for risk and return
WREanalysisExpostNonParametricVaR	Ex-post non parametric VaR
WREanalysisExpostNonParametricVaRX	Ex-post non parametric VaR with different estimation windows for risk and return
WREanalysisExpostNormalCVaR	Ex-post Gaussian Conditional VaR
WREanalysisExpostNormalCVaRX	Ex-post Gaussian Conditional VaR with different estimation windows for risk and return
WREanalysisExpostNormalVaR	Ex-post Gaussian VaR
WREanalysisExpostNormalVaRX	Ex-post Gaussian VaR with different estimation windows for risk and return
WREanalysisExanteDeltaGammaVaR	Ex-ante delta gamma VaR for portfolios containing options

3.4.6 Loss and drawdown measures

Gaussian shortfall probability

The shortfall probability is the probability that the return of the portfolio may fall below some given threshold return.
The **ex-ante gaussian shortfall** probability is given by:

$$\begin{aligned}\text{NormalShortfall}_{\text{ex-ante}} &= \Phi\left[\frac{\mu - \text{Return}_{\text{ex-ante}}}{\text{Volatility}_{\text{ex-ante}}}\right] \\ &= \Phi\left[\frac{\mu - \rho^\top \omega}{\sqrt{\omega^\top \Gamma \omega}}\right]\end{aligned}$$

The **ex-post** formulation is given by:

$$\text{NormalShortfall}_{\text{ex-post}} = \Phi\left[\frac{\mu - \text{Return}_{\text{ex-post}}}{\text{Volatility}_{\text{ex-post}}}\right]$$

where μ is the threshold return.

Maximum loss

The **ex-post maximum loss** is the minimum return cumulated from the beginning over a given time period. Given the portfolio historical returns time series, the ex-post maximum loss is defined by:

$$\begin{aligned}\text{MaxLoss}_{\text{ex-post}} &= \min\left[r_p(1), r_p(1)+r_p(2), ..., r_p(1)+r_p(2)+...+r_p(n)\right] \\ &= \min_{t=1,...,n}\left[\sum_{i=1}^{t} r_p(i)\right]\end{aligned}$$

Drawdown

The **drawdown** is defined as the largest drop from a peak to the end of a given time period. It captures a path-dependent feature of a time series which is not represented in the distribution of the returns.

Given the portfolio historical returns, the **ex-post drawdown** is expressed as a return and is given by:

$$\text{Drawdown}_{\text{ex-post}}([1,t]) = \min\left(\min_{u=1,...,t}\frac{S(u)-S(t)}{S(t)}, 0\right)$$

The **maximum drawdown** over the period $[1,T]$ is defined as the largest drop from a peak to a bottom in this period. It is expressed as a return and is defined as:

$$\text{MaxDrawdown}_{\text{ex-post}}([1,T]) = \min_{t=1,...,T}\text{Drawdown}_{\text{ex-post}}([1,t])$$

Related functions

Function	Description
WREanalysisNormalShortfall	Ex-ante Gaussian shortfall probability
WREanalysisExpostMaxLoss	Ex-post maximum loss
WREanalysisExpostMaxLossX	Ex-post maximum loss with different estimation windows for risk and return
WREanalysisExpostNormalShortfall	Ex-post Gaussian shortfall probability
WREanalysisExpostNormalShortfallX	Ex-post Gaussian shortfall probability with different estimation windows for risk and return

3.5 Ratios

3.5.1 Sharpe ratio

The Sharpe ratio is a risk-adjusted measure for diversified portfolios and represents the excess reward to variability ratio. The greater the Sharpe ratio, the better the portfolio, but the Sharpe ratio makes sense only if it is positive.

The **ex-ante Sharpe ratio** is given by:

$$\text{SharpeRatio}_{ex-ante} = \frac{\text{MeanReturn}_{\text{ex-ante}} - r_f}{\text{Volatility}_{\text{ex-ante}}}$$

The **ex-post Sharpe ratio** is given by:

$$\text{SharpeRatio}_{ex-post} = \frac{\text{MeanReturn}_{\text{ex-post}} - r_f}{\text{Volatility}_{\text{ex-post}}}$$

3.5.2 Modified Sharpe ratio

The Sharpe ratio is valid only if the assets are normally distributed. As soon as the portfolio is composed of technology stocks, distressed companies, hedge funds, high yield bonds, etc..., this ratio is no, because it does not take into account the higher order moments of the distribution, like the skewness and the kurtosis.

The modified Sharpe ratio is the ratio of the excess return divided by the Modified Value-at-Risk. Two portfolios with the same mean and the same volatility will be differentiated by their extreme losses. This is the advantage of working with the modified Value-at-Risk and Sharpe ratio.

The **ex-ante modified Sharpe ratio** is given by:

$$\text{ModifiedSharpeRatio}_{ex-ante} = \frac{\text{Return}_{\text{ex-ante}} - r_f}{\text{ModifiedVaR}_{\text{ex-ante}}}$$

The **ex-post modified Sharpe ratio** is given by:

$$\text{ModifiedSharpeRatio}_{ex-post} = \frac{\text{Return}_{\text{ex-post}} - r_f}{\text{ModifiedVaR}_{\text{ex-post}}}$$

3.5.3 Information ratio

The information ratio is a risk-adjusted measure of a portfolio out-performance with respect to a benchmark. Considering a benchmarked allocation strategy, it indicates how much the risk taken in disconnecting from the benchmark is remunerated.

The **ex-ante information ratio** of a portfolio with respect to a benchmark is defined as:

$$\text{InformationRatio}_{ex-ante} = \frac{\text{Return}_{\text{ex-ante}} - \bar{r_b}}{\text{TrackingError}_{\text{ex-ante}}}$$

The **ex-post information ratio** of a portfolio with respect to a benchmark is defined as:

$$\text{InformationRatio}_{ex-post} = \frac{\text{Return}_{\text{ex-post}} - \bar{r_b}}{\text{TrackingError}_{\text{ex-post}}}$$

where $\bar{r_b}$ is the mean historical return of the benchmark.

3.5.4 Sortino ratio

This measure is similar to the Sharpe ratio, except that it uses downside deviation as a denominator, whereas the Sharpe ratio uses standard deviation.

The **Sortino ratio** is the ratio of the excess return (compared to the risk-free rate) over the downside semi-volatility, so it measures the return to "bad" volatility. This ratio allows investors to assess risk in a better manner than simply looking at excess returns to total volatility, since the Sortino ratio considers how often the price of the security rises as opposed to how often it falls.

$$\text{SortinoRatio}_{ex-post} = \frac{\text{Return}_{\text{ex-post}} - r_f}{\text{DownsideVolatility}_{\text{ex-post}}}$$

3.5.5 Omega ratio

The Omega ratio uses all the information contained within the returns time series. It can be used to rank and evaluate portfolios unequivocally. All that is known about the risk and return of a portfolio is contained within this measure.

For a given threshold return r, the Omega ratio is the probability weighted ratio of gains to losses, relative to r:

$$\text{OmegaRatio}(r) = \frac{\int_r^{+\infty} (1 - F(x))\, dx}{\int_{-\infty}^{r} (F(x))\, dx}$$

where F is the cumulative distribution of returns.

As is illustrated in the figure below, it involves partitioning returns into loss and gain above and below a return threshold and then considering the probability weighted ratio of returns above and below the partitioning.

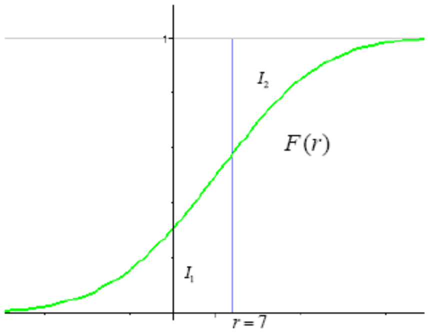


Figure 3.5: Cumulative distribution of the returns

In the graph above,

- The cumulative distribution has a mean return of 5.
- The loss threshold is at $r = 7$.

- I_2 is the area above the graph of F and to the right of 7.
- I_1 is the area under the graph of F and to the left of 7.
- The Omega ratio at $r = 7$ is the ratio of probability weighted gains, I_2, to probability weighted losses, I_1.

This measure is, in a rigorous mathematical sense, equivalent to the distribution of returns itself, rather than simply being an approximation to it. It therefore omits none of the information in the distribution and is as statistically significant as the returns series itself. It measures the combined effect of all the moments of the distributions, rather than the individual effects of any of them.

Note that the Omega ratio is strictly decreasing as a function of the threshold return and takes the value 1 at the portfolio mean return.

Both the **ex-ante and ex-post Omega ratios** are provided in this Wall Risk Engine®.

3.5.6 Related functions

Function	Description
WREanalysisExanteIR	Ex-ante information ratio
WREanalysisExanteModifiedSharpeRatio	Ex-ante modified Sharpe ratio
WREanalysisExanteOmega	Ex-ante Omega ratio
WREanalysisExanteSharpeRatio	Ex-ante Sharpe ratio
WREanalysisExpostIR	Ex-post information ratio
WREanalysisExpostIRX	Ex-post information ratio with different estimation windows for risk and return
WREanalysisExpostModifiedSharpeRatio	Ex-post modified Sharpe ratio
WREanalysisExpostModifiedSharpeRatioX	Ex-post modified Sharpe ratio with different estimation windows for risk and return
WREanalysisExpostOmega	Ex-post Omega ratio
WREanalysisExpostOmegaX	Ex-post Omega ratio with different estimation windows for risk and return
WREanalysisExpostSharpeRatio	Ex-post Sharpe ratio
WREanalysisExpostSharpeRatioX	Ex-post Sharpe ratio with different estimation windows for risk and return
WREanalysisExpostSortinoRatio	Ex-post Sortino ratio
WREanalysisExpostSortinoRatioX	Ex-post Sortino ratio with different estimation windows for risk and return

3.6 Risk decomposition

3.6.1 Overview

Risk and portfolio managers are always concerned with the level of risk implied by a given position in their portfolio. Wall Risk Engine® provides three types of risk decomposition methods:

- Marginal risk, which measures the impact of small changes in a position,

- Component risk, which allocates the total risk of the portfolio to the positions or books, based on the marginal risk measures.

These functions are implemented with a parametric approach.

3.6.2 Notations

In this section, we use the following notations:

- p is the number of underlying assets of the portfolio,
- ω_i is the weight of the asset i, with $\omega_i \geq 0$ and $\sum_{i=1}^{p} \omega_i = 1$,
- $\text{Risk}_{\text{Portfolio}}$ is the value of the considered risk measure for the global portfolio,
- Risk_i is the value of the considered risk measure for the asset i.

3.6.3 Volatility decomposition

Marginal volatility contribution

The marginal volatility measures the impact on the portfolio volatility of an infinitesimal change in a given position i. It is the partial derivative of the portfolio volatility with respect to the weight of the asset:

$$\text{MarginalVolatility}_i = \frac{\partial \text{Volatility}_{\text{Portfolio}}}{\partial \omega_i}$$

Component volatility

The component volatility provides the proportion of the portfolio volatility that can be attributed to each component of the portfolio:

$$\begin{aligned} \text{ComponentVolatility}_i &= \omega_i \text{MarginalVolatility}_i \\ &= \omega_i \frac{\partial \text{Volatility}_{\text{Portfolio}}}{\partial \omega_i} \end{aligned}$$

Related functions

Function	Description
WREanalysisVolatilityContribution	Marginal volatility contribution

3.6.4 Tracking-error decomposition

Marginal tracking error contribution

The marginal tracking error measures the impact on the portfolio volatility of an infinitesimal change in a given position i. It is the partial derivative of the portfolio tracking error with respect to the weight of the asset:

$$\text{MarginalTE}_i = \frac{\partial \text{TE}_{\text{Portfolio}}}{\partial \omega_i}$$

Component tracking error

The component tracking error provides the proportion of the portfolio tracking error that can be attributed to each component of the portfolio:

$$\begin{aligned}\text{ComponentTE}_i &= \omega_i \text{MarginalTE}_i \\ &= \omega_i \frac{\partial \text{TE}_{\text{Portfolio}}}{\partial \omega_i}\end{aligned}$$

Related functions

Function	Description
WREanalysisTrackingErrorContribution	Marginal tracking error contribution

3.6.5 VaR decomposition

Marginal Gaussian VaR contribution

The Marginal Gaussian VaR measures the impact on the portfolio volatility of an infinitesimal change in a given position i. It is the partial derivative of the portfolio VaR with respect to the weight of the asset:

$$\text{MarginalVaR}_i = \frac{\partial \text{VaR}_{\text{Portfolio}}}{\partial \omega_i}$$

Component Gaussian VaR

The Component VaR provides the proportion of the portfolio VaR that can be attributed to each of the components of the portfolio. It is a linear approximation of the Incremental VaR as it approximates the change in VaR if a portfolio component were to be deleted from (or added to) the portfolio.

The Component VaR of instrument i is defined by:

$$\begin{aligned}\text{ComponentVaR}_i &= \omega_i \text{MarginalVaR}_i \\ &= \omega_i \frac{\partial \text{VaR}_{\text{Portfolio}}}{\partial \omega_i}\end{aligned}$$

The component VaRs of the positions add up to the total VaR of the portfolio:

$$\text{VaR}_{\text{Portfolio}} = \sum_{i=1}^{p} \text{ComponentVaR}_i$$

This additive property of the component VaR has important applications in the allocation of risk to different units (desks, sectors, countries).

Related functions

Function	Description
WREanalysisNormalVaRContribution	Marginal Gaussian VaR contribution

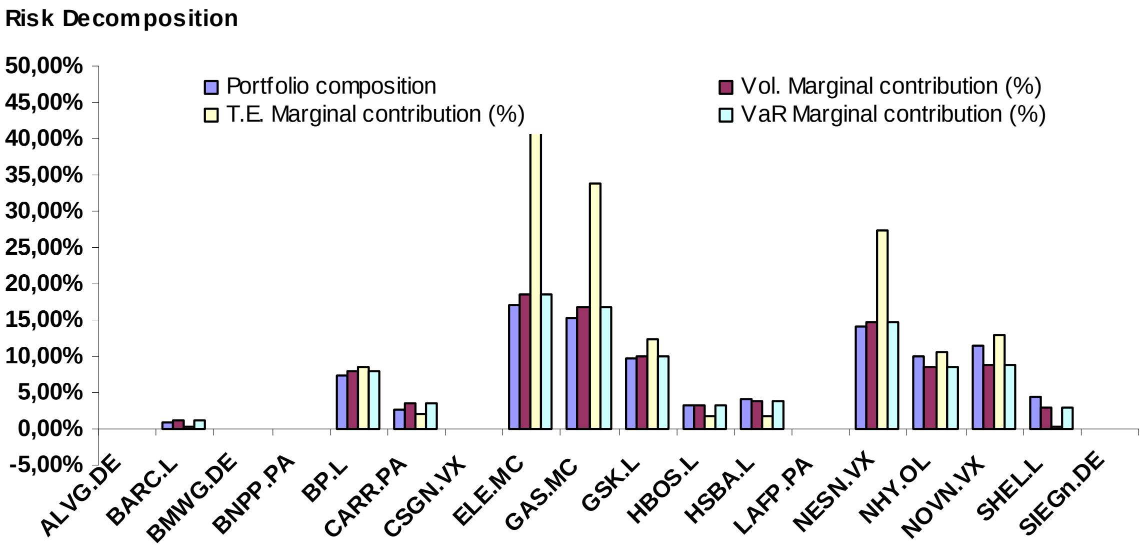


Figure 3.6: Example of marginal risk decomposition of a portfolio

3.7 Concentration and diversification measures

3.7.1 Ex-ante portfolio concentration measures

Measuring the diversification of an investment universe

Measuring the diversification of an investment universe is very useful to detect anomalies, such as highly correlated assets in the same investment universe. There are different ways to visualize and quantify this diversification:

- Measuring the diversification of the risk levels via the distribution of the volatilities;
- Analyzing the distribution of the correlations between the assets;
- Computing quantitative indicators of the diversification such as the intra-correlation, described below in this section.

Measuring the diversification of a portfolio

Wall Risk Engine® provide a series of ex-ante concentration and diversification measures of the portfolio:

- The concentration coefficient measures the concentration of the portfolio given the weights of the underlying components:

$$CC_{exante} = \frac{1}{\sum_{i=1}^{n} \omega_i^2}$$

  The concentration coefficient is between 1 and n, where n is the number of assets of the portfolio. The concentration coefficient equals n if the portfolio is equally-weighted and equals 1 if the whole portfolio is concentrated in one asset only.

- The intra-portfolio correlation ranges from -1 to 1 and measures the degree to which the various assets in a portfolio can be expected to perform in a similar way or not. A

measure of -1 means that the assets within the portfolio perform perfectly oppositely: whenever one asset goes up, the other goes down. A measure of 0 means that the assets fluctuate independently, i.e. that the performance of one asset cannot be used to predict the performance of the others. A measure of 1, on the other hand, means that whenever one asset goes up, so do the others in the portfolio. To eliminate diversifiable risk completely, one needs an intra-portfolio correlation of -1.
The intra-portfolio correlation is defined as follows:

$$IPC_{exante} = \frac{\sum_i \sum_{i \neq j} \omega_i \omega_j \rho_{ij}}{\sum_i \sum_{i \neq j} \omega_i \omega_j}$$

where ρ_{ij} is the correlation coefficient between assets i and j.

- The diversifiable risk eliminated, expressed as a percentage, is equivalent to the intra-portfolio correlation. The linear relationship between intra-portfolio correlation and diversifiable risk elimination is given in the table hereafter:

Intra-portfolio correlation	Diversifiable risk eliminated
-1	0%
-0.5	25%
0	50%
0.5	75%
1	100%

Related functions

Function	Description
WREanalysisExanteCC	Ex-ante concentration coefficient
WREanalysisExanteIPC	Ex-ante intra-portfolio correlation
WREanalysisExanteDRisk	Ex-ante diversifiable risk eliminated
WREanalysisExpostCorrCoef	Ex-post correlation coefficient
WREanalysisExpostCorrCoefX	Ex-post correlation coefficient with different estimation windows for risk and return

3.8 Backtesting functionalities

3.8.1 Backtesting principles

Backtesting an allocation strategy is a crucial step when designing a portfolio. It allows for the analysis of the behavior of the allocation strategy in given market conditions: if the portfolio had been created 2 years ago and reallocated on a monthly basis according to a given quantitative strategy, what would have been its risk/return profile?

When the weights of a portfolio are set and the portfolio is left untouched for a given period, the weights initially set at the allocation date will be changing over the time period (provided that the portfolio is not rebalanced) because the prices of its components are varying over time. Wall Risk Engine® provides 2 functions to help the user recompose the evolution of the performances of a portfolio:

- a function that recomposes the performances of a portfolio between two allocation dates of the backtesting (how will the portfolio perform between two allocation dates if it is not rebalanced?);
- a function that builds the simulated past values of a portfolio (what are the past performances of the portfolio that led to today's composition without any rebalancing?)

3.8.2 Related functions

Function	Description
WREanalysisFutureValues	Builds the values of a portfolio between 2 allocation dates of a back-test
WREanalysisPastValues	Builds the simulated past values of a portfolio

4 Robust Portfolio Allocation

4.1 Overview

4.1.1 Tailored portfolio allocation

The Asset Allocation Module of Wall Risk Engine® helps investors design quantitative investment strategies to achieve the best possible trade-off between risk and return, according to their investment profile:

- minimization of a quadratic risk measure under sectoral and global performance constraints,
- maximization of a utility function under sectoral and global quadratic risk constraints.

The formulation and the resolution of these portfolio optimization problems are based on the most recent developments in Modern Portfolio theory ([Lue98], [SS03] and [Meu05]) and Robust Optimization theory ([GI03] and [LEGO03]).
The methods provided in this module allow to build the following type of strategies:

Type of strategy	Objective	Constraint
Robust Risk Minimization	Min risk measure	Constraint on the expected return
Robust Risk Budgeting	Max Expected Return	Global and sectorial quadratic risk constraints

These quantitative strategies are designed to fit with any risk and performance model described in the Risk and Performance Modeling section of this documentation. By bringing the best of quantitative methods (robust optimization, dynamic risk and performance models) together with the investor's expertise (views on the trends, tailored strategy design), Wall Risk Engine®

is a powerful and flexible decision-aid solution for the invesment professionals.

Some of the allocation functions are available in two versions:

- without a risk-free asset (ex: $WREallocMV$);
- with a risk-free asset, in which case the asset is not considered when computing the covariance matrix of the investment universe (ex: $WREallocMVRfr$).

4.1.2 Notations

Throughout this section, we use the following notations:

- n is the size of the investment universe (the number of underlying components),
- $n_{sectors}$ is the number of user-defined sectors in the investment universe,
- ω is the vector of weights of the portfolio to be rebalanced (variables of the optimization problem),
- ω_{ini} is the vector of initial weights of the portfolio to be rebalanced,
- c_i is the transaction cost of one unit of the i^{th} underlying asset (transaction costs are supposed to be symmetric),
- Γ is the variance-covariance matrix of the underlying assets,
- $\beta(i)$ is the beta of the i^{th} underlying component,
- $minWeights(i)$ and $maxWeights(i)$ are the minimum and maximum bounds on the i^{th} underlying component $\forall i = 1, ..., n$,
- $minWeights_{sector}(i)$ and $maxWeights_{sector}(i)$ are the minimum and maximum bounds on the i^{th} sector $\forall i = 1, ..., n_{sectors}$.

4.2 Allocation models

4.2.1 Minimization of a risk measure under performance constraints

General formulation

This strategy is a generalization of the Markowitz portfolio selection approach (cf. [PMNP52] and [Mar59]). It allows to minimize any quadratic risk measure under any set of linear performance constraints:

$$\begin{cases} \min\limits_{\omega} \text{RiskMeasure}(\omega) \\ \text{subject to} \\ \quad \text{Performance constraints : Performance}(\omega) \geq \mu \\ \quad \text{Bound constraints : } l_i \leq \omega_i \leq u_i \forall i \in [1, n] \\ \quad \text{Sectorial constraints : } l_i^{\text{sector}} \leq \sum\limits_{i \in \text{sector}} \omega_i \leq u_i^{\text{sector}} \forall i \in [1, n] \\ \quad \text{Transaction cost constraint : } \sum\limits_i \left|\omega_i - \omega_i^{ini}\right| c_i \leq \text{TC}_{\max} \forall i \in [1, n] \\ \quad \text{Any equality or inequality user-defined linear constraint} \end{cases}$$

Examples

- Robust mean-variance allocation

$$\begin{aligned} \text{RiskMeasure}(\omega) &= \omega^\top \Gamma \omega \\ \text{Performance}(\omega) &= \text{Return}_{\text{ex-ante}}(\omega) \\ \mu &= \text{Target return} \end{aligned}$$

- Robust index tracking

$$\begin{aligned} \text{RiskMeasure}(\omega) &= \text{TrackingError}_{\text{ex-ante}}(\omega) \\ \text{Performance}(\omega) &= \text{Return}_{\text{ex-ante}}(\omega) - \text{Return}_{\text{ex-ante}}(\text{Benchmark}) \\ \mu &= \text{Target surperformance with respect to the benchmark} \end{aligned}$$

Related functions

Function	Description
WREallocIT	Index-tracking allocation
WREallocITRfr	Index-tracking allocation with risk-free rate
WREallocMV	Mean-variance allocation
WREallocMVRfr	Mean-variance allocation with risk-free rate
WREallocMVTC	Mean-variance allocation with transaction costs
WREconstSector	Modelization of sectorial constraints
WREmve	Mean-variance efficient frontier

4.2.2 Maximization of a utility function under risk constraints

General formulation

This strategy allows to maximize a linear utility function under any quadratic risk measure (such as a volatility or tracking error budget constraint):

$$\begin{cases} \max\limits_{\omega} \text{PerfMeasure}(\omega) \\ \text{subject to} \\ \quad \text{Risk constraints : RiskMeasure}(\omega) \leq \text{risk}_{\max} \\ \quad \text{Bound constraints : } l_i \leq \omega_i \leq u_i \forall i \in [1,n] \\ \quad \text{Sectorial constraints : } l_i^{\text{sector}} \leq \sum\limits_{i \in \text{sector}} \omega_i \leq u_i^{\text{sector}} \quad \forall i \in [1,n] \\ \quad \text{Any equality or inequality user-defined linear constraint} \end{cases}$$

Examples

- Robust volatility budgeting

$$\begin{aligned} \text{Performance}(\omega) &= \text{Return}_{\text{ex-ante}}(\omega) \\ & \quad \text{based on any performance model} \\ \text{RiskMeasure}(\omega) &= \text{Volatility}_{\text{ex-ante}}(\omega) \\ & \quad \text{at the global portfolio level or at the sectorial level} \end{aligned}$$

- Robust tracking error budgeting

$$\begin{aligned} \text{Performance}(\omega) &= \text{Return}_{\text{ex-ante}}(\omega) - \text{Return}_{\text{ex-ante}}(\text{Benchmark}) \\ & \quad \text{based on any performance model} \\ \text{RiskMeasure}(\omega) &= \text{TrackingError}_{\text{ex-ante}}(\omega) \\ & \quad \text{at the global portfolio level or at the sectorial level} \end{aligned}$$

Related functions

Function	Description
WREallocRiskBudgeting	Risk budgeting allocation function
WREallocRiskBudgetingRfr	Risk budgeting allocation function with risk-free rate
WREallocRiskBudgetingIT	Tracking error budgeting allocation function
WREconstSector	Modelization of sectorial constraints

4.2.3 Maximization of the Sharpe ratio

General formulation

This strategy allows to maximize the Sharpe ratio of a portfolio under a set of linear constraints:

$$\begin{cases} \max\limits_{\omega} \text{SharpeRatio}_{\text{ex-ante}}(\omega) \\ \text{subject to} \\ \quad \text{Bound constraints : } l_i \leq \omega_i \leq u_i \forall i \in [1,n] \\ \quad \text{Sectorial constraints : } l_i^{\text{sector}} \leq \sum\limits_{i \in \text{sector}} \omega_i \leq u_i^{\text{sector}} \quad \forall i \in [1,n] \\ \quad \text{Any equality or inequality user-defined linear constraint} \end{cases}$$

Related functions

Function	Description
WREallocSharpeRatio	Maximization of the Sharpe ratio
WREconstSector	Modelization of sectorial constraints

4.3 Non Gaussian Optimization

4.3.1 Higher moment optimization

The aim of this optimization function is to take higher moments into consideration at the problem formulation level. The problem to solve is the following:

$$\omega_* \leftarrow min_\omega \omega^T \Gamma \omega - \kappa M_3(\omega) + v M_4(\omega)$$
$$sc \begin{cases} \rho^T \omega \geq \mu \\ C^T \omega \leq b \\ C_{inf} \leq \omega \leq C_{sup} \end{cases}$$

Where:

- ω is the vector of portfolio weights
- ρ is the vector of expected returns of assets
- C and b define some additional linear constraints
- C_{inf} and C_{sup} are min weights and max weights constraints
- Γ is the calibrated covariance matrix of the assets returns
- $M_3(\omega)$ and $M_4(\omega)$ are the third and the fourth moments of the distribution of the returns of ω
- κ and v are positive coefficients weighting the importance of the optimization of each moments

The approach consists using convex relaxation techniques so that the bundle method can be used to solve the problem.

4.3.2 Drawdown optimization

The aim here is minimizing the maximal drawdown (MaxDD) of the portfolio.
The MaxDD is estimated over the past period taken for the estimation period as follows:

$MaxDD(\omega) = max_{1 \leq t \leq ndate} max_{1 \leq s \leq t} r_{ts}^T \omega$

Where:

- ω is the vector of portfolio weights
- $ndate$ is the size of the estimation period
- t and s are dates in the estimation period such as $s < t$

- r_{st} is the vector of assets returns between the dates s and t

The program to be solved is the following:

$$\omega_* \leftarrow min_\omega \omega^T \Gamma \omega$$

$$sc \begin{cases} \rho^T \omega \geq \mu \\ C^T \omega \leq b \\ C_{inf} \leq \omega \leq C_{sup} \\ MaxDD(\omega) \leq DDmax \end{cases}$$

Where:

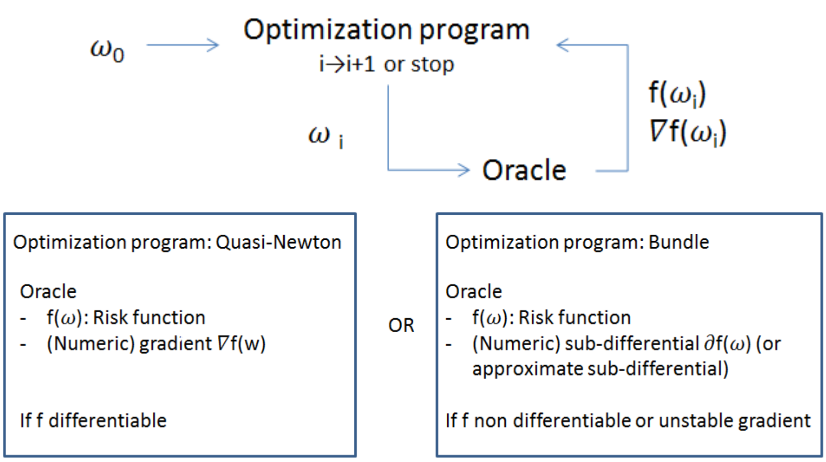
- ρ is the vector of expected returns of assets
- C and b define some additional linear constraints
- C_{inf} and C_{sup} are min weights and max weights constraints
- Γ is the calibrated covariance matrix of the assets returns

The approach consists in solving directly this convex non-differentiable problem using so-called bundle methods.

4.3.3 Customized utility function

More generally, the low level optimizers (Quasi Newton and Bundle) included in Wall Risk Engine allows to deal with any type of risk measure as soon as a way to make it convex can be found. The convexification methods are a field of recognized expertise of Raise Partner.

The principle of the methodology is summarized in the graph below:



Below are a few examples of customized utility functions.

Monte Carlo method

- $f : \omega \rightarrow \mathrm{VaR}_{MC}(\omega)$ computed with one of Raise Partner Monte Carlo method (Quasi-Monte Carlo, Gaussian Mixture - Monte Carlo)
- Gradient or sub-differential can be numerically computed with the same method than f
- Multi-period optimization can be dealt with this way: the optimization variable is then $J(\mathrm{W}_1, ..., \mathrm{W}_n)$, i.e. the set of allocations through time at given dates

Non Gaussian optimization based on the Bundle method [HO00]

The risk function should be convex (CVaR, regularized VaR, Max Drawdown).
Here, sub-differential can be numerically computed with the same method than the risk function (no explicit formulation is required), for instance Robust VaR or CVaR computation:

- Short term allocation: DeltaGamma (or Duration-Convexity) pricing
- Long term allocation: Full pricing

The method for running a robust optimization at the book level is described by the steps below:

- Step 1- Getting the distribution of the risk factors (Historical simulations or Monte Carlo simulations)
- Step 2- Getting the distribution of the lines of the book (via delta-gamma extension or full pricing)
- Step 3 - Getting a regularized distribution of the book
- Step 4 - Optimizing the tail of the distribution (Bundles method)

Multi-risk budget optimization based on the Bundle method

- Problem to solve:

$$\omega_* \leftarrow min_\omega TC\left(\omega^t, \omega^{t-1}\right)$$

$$s.t. \begin{cases} VaR_{bond}(\omega^t) \leq RiskBudget^t_{bond} \\ VaR_{FX}(\omega^t) \leq RiskBudget^t_{FX} \\ VaR_{equity}(\omega^t) \leq RiskBudget^t_{equity} \\ \omega_{min} \leq \omega^t \leq \omega_{max} \\ C\omega^t \leq b \end{cases}$$

- Feasibility ($C \neq \emptyset$): find a proof, i.e. a strictly feasible portfolio ω^t_0.
- Resolution: solve the optimization problem using ω^t_0.

4.4 Constraints

As mentioned in the formulations above, all the methods provided in the Wall Risk Engine® allocation module take into account the following user-defined constraints in addition to the risk and performance constraints embedded in these strategies:

- **Bound constraints**: lower and upper bounds for each underlying asset (shorting being allowed). These constraints are defined through the input vectors $minWeights$ and $maxWeights$ and expressed as percentages. For instance, for an unconstrained long-only strategy in a size-n universe, the input parameters will be set to:

$$\begin{aligned} minWeights(i) &= 0 \quad \forall i \in [1, ..., n] \\ maxWeights(i) &= 1 \quad \forall i \in [1, ..., n] \end{aligned}$$

  and the optimal weights ω will be computed given the constraints

$$minWeights(i) \leq \omega(i) \leq maxWeights(i) \quad \forall i \in [1, ..., n]$$

- **Sectoral weight constraints**: definition of any sectoral classification (geographical, industrial, etc...) with the associated weight constraints. An on-the-shelf function is embedded in this module to make the standard sectoral constraint modeling easier: the *WREallocConstSector* function. This function builds the input matrix C and the input vector b defining the linear constraints given the explicit input sectoral constraints:

$$minWeights_{sector}(i) \leq \sum_{j \in sector_i} \omega(j) \leq maxWeights_{sector}(i) \quad \forall i \in [1, ..., n_{sectors}]$$

- Beta constraints: definition of constraints on the beta at the portfolio or at the sectorial level. These constraints are expressed as linear constraints and formulated in the input parameters C and b (cf linear constraints for the definition of C and b). For instance, the beta constraint at the portfolio level can be written as:

$$\beta_{min} \leq \sum_{i}^{n} \beta(i)\omega(i) \leq \beta_{max}$$

- **Any other linear constraints**: definition of any linear constraints with respect to the weights of the portfolio. These constraints are defined through the input parameters C and b (cf. the following item regarding linear constraints):

$$\begin{aligned} {C_{eq}}^{\top}\omega &= b_{eq} \\ {C_{ineq}}^{\top}\omega &\leq b_{ineq} \end{aligned}$$

  where

  - n is the size of the universe,

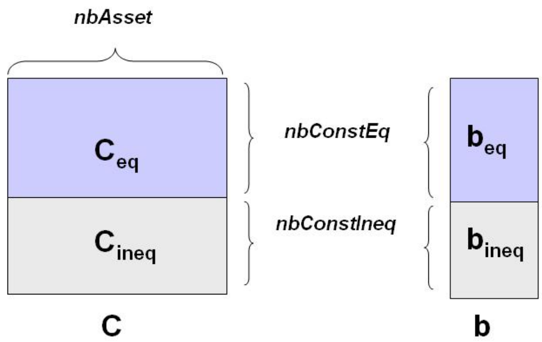


Figure 4.1: C and b input parameters

- ω is the vector of weights (size n),
- n_{eq} is the number of linear equality constraints and n_{ineq} the number of linear inequality constraints,
- C and b are defined as illustrated in the figure 4.1.

- **Turnover constraints** for each underlying component, sector, or at the global portfolio level, so as to limit the transactions in the rebalancing process. These constraints can be formulated as linear constraints and defined in the C and b parameters.

5 Simulation and Stress-testing

5.1 Overview

This module embeds Monte Carlo methods for the simulation of future values and pricing of financial instruments and for the computation of VaR on large portfolios.

These functionalities allow to:

- introduce any complex derivative product in an investment universe via the implementation of a payoff function,
- evaluate the impact of market conditions (macro-economic factors) on a product or a portfolio,
- measure the diversification and performance impact of any complex financial instrument (such as structured products) in an investment universe,,
- run fast Monte-Carlo simulation for VaR computation of large-scale portfolios.

5.2 Multi-dimensional Monte Carlo Simulation

5.2.1 Formulation

This module embeds a function which simulates a multi-dimensional geometric brownian motion (GBM) on a given horizon. A GBM (also called exponential Brownian motion) is a continuous-time stochastic process (cf [Duf88], [KS91] and [Gla04]) in which the logarithm of the randomly varying quantity follows a Brownian motion. In finance, it can be used to simulate a basket of correlated assets:

$$\frac{dS_t^i}{S_t^i} = \mu_i dt + \sigma_i dW_t^i$$

The simulations of the Wiener processes are discretized according to the Euler-Maruyama methodology [PEK99].

The equation has an analytic solution for an arbitrary initial value S_0^i:

$$S_t^i = S_0^i e^{\left(\mu_i - \frac{\sigma_i^2}{2}\right)t + dW_t^i}$$

The random variable $\ln\left(S_t^i / S_0^i\right)$ is normally distributed with mean $\left(\mu_i - \sigma_i^2/2\right)t$ and variance $\sigma_i^2 t$.

The inputs of this simulation function are the model parameters, which can be calibrated from market data (risk-neutral approach) or explicitly defined by the user in a stress-testing approach:

- the drifts μ_i of the assets (supposed constant in this model),
- the volatilities σ_i of their log-returns (supposed constant in this model),
- the correlations between the components of the multi-dimensional brownian motion dW_t.

The simulated scenarios can then be used to recompose the simulated path of a portfolio ω at any time t:

$$r_P(t) = \sum_{i=1}^{n} \omega_i(t) r_i(t)$$

where $r_P(t)$ is the return of the portfolio at time t, $r_i(t)$ is the return of the i^{th} asset at time t.

This simulation tool can be used to stress-test the portfolio by defining "what if" scenario hypothesis on any parameter (volatility, expected return, cross-asset correlations).

5.2.2 Related functions

Function	Description
WREsimulationMultiBrownianX	Multi-dimensional geometric Brownian motion simulation

5.3 Payoff simulation

5.3.1 Dense approach

Formulation

Wall Risk Engine® provides a range of functionalities to simulate the future payoffs of any financial instrument implemented by the user in a payoff function. The payoff function defined by the user is then plugged into the MC simulation function of Wall Risk Engine®.

NB: this functionality is only available in the C API of Wall Risk Engine®.
The input data can be one of the following:

- the inter-asset correlations, the drifts and the volatilities of the underlying assets,
- the historical prices of the underlying assets.

We suppose that the processes to be simulated follow a geometric Brownian motion (GBM) driven by the following Stochastic Differential Equation (SDE):

$$\frac{dS_t^i}{S_t^i} = \mu_i dt + \sigma_i dW_t^i$$

where the correlations between the components of the multi-dimensional standard Brownian motion are given by the correlation matrix C.

5.3.2 Factorial approach

Formulation

This function simulates future payoffs of any financial instrument implemented by the user in a payoff function, using a multi-factor diffusion model. The payoff function defined by the user is then plugged into the MC simulation function of Wall Risk Engine®.
NB: this functionality is only available in the C API of Wall Risk Engine®.

First, we define an explanatory model on the logarithm of returns (log-returns) of the n underlying assets. Let

- $S_i(t)$ be the price of the ith underlying asset at time t,
- $F_j(t)$ be the value of the jth factor at time t,
- $r_i(t) = \ln\left[S_i(t+\Delta t)/S_i(t)\right]$ be the log-return for a given horizon Δt of the i^{th} underlying asset at time t,
- $R_i(t) = \ln\left[F_i(t+\Delta t)/F_i(t)\right]$be the log-return for the same horizon of the j^{th} factor.

The multi-factor linear model is defined by:

$$\forall t, \forall i \in \{1,...,n\}, r_i(t) = \alpha_i + \sum_j \beta_{ij} R_j(t) + \epsilon_i(t)$$

where $E(\epsilon_i) = 0$, $\mathrm{Var}(\epsilon_i)$ is constant and ϵ_i is uncorrelated with the factor return and with other ϵ_j's.

The α_i and β_j are constant for all t. The α_i, β_j and the variances of ϵ_i can be estimated from the regression of the historical log-returns of the underlying assets on the historical log-returns of the factors, using a robust APT routine.

The simulation of a scenario for the factor (following GBM) implies dynamics for the log-returns of the underlying assets. Using the factor model, we get:

$$\forall t, \forall i \in \{1,...,n\}, \ln\left[\frac{S_i(t+\Delta t)}{S_i(t)}\right] \alpha_i + \sum_j \beta_{ij} \ln\left[\frac{F_j(t+\Delta t)}{F_j(t)}\right] + \epsilon_i(t)$$

Based on the simulated paths F_j for the factors we deduce simulated paths for the assets prices S_i using the previous relation.

5.3.3 Related functions

Function	Description
WREsimulationMonteCarloFact	Factorial multi-dimensional MC simulation
WREsimulationMonteCarloBase	Multi-dimensional MC with drifts and Choleski matrix as input
WREsimulationMonteCarloCorr	Multi-dimensional MC with drifts and correlations as inputs
WREsimulationMonteCarloHist	Mean-variance allocation with historical time series as inputs

5.4 Large-scale Monte-Carlo VaR

5.4.1 Introduction

To compute the Value-at-Risk of a portfolio of assets, we assume that the returns of the assets are normally distributed and we use the following factorial approach:

- Compute a set of principal factors that enables to explain a given level of variance;
- Run a Monte Carlo simulation to generate paths (shocks) for the underlying assets (Gaussian model with no residual error);
- Compute the Value-at-Risk of the portolio can be calculed using the simulated scenarios.

One true computational challenge is the size of the portfolio, especially when combined with a limited number of available quotations. Typically, we have to deal with dozens of historical quotations whereas the portfolio size goes from hundreds to thousands of positions. To overcome this dimension issue, we decompose the investment universe into submarkets that will enable to generate a block of a limited number of factors for each submarket. The combination of these blocks of factors when appropriately correlated will allow to explain the target level of variance.

5.4.2 Modeling

Given a portfolio of n assets of returns, we would like to use a reduced model of the form:

$$R_i = \beta^T F + \alpha + \epsilon \tag{5.1}$$

where F is the vector of q factors, β the loadings, α_i a constant equal to $E(R)$ and ϵ the residual modeling error. As input of the problem, we have an estimate of the mean m_i of the returns and the covariance matrix Σ. For ease of statement, we are going to use standardized variables $\tilde{R}$ and their sample correlation / covariance $\widehat{\Gamma}$ and perform all the computations on those standardized variables.

An important preliminary step is to make sure the correlation matrix is well-defined. That requirement is often not satisfied because the length of historical quotations is too short compared to the size of the portfolio. We solve a SDP problem to ensure the structural property of the matrix (cf the Risk Modeling chapter of this documentation):

$$\begin{aligned} &\min_{X \in S_+^n} ||X - \widehat{\Gamma}||_F \\ &X_{ii} = 1, \quad \lambda_{min}(X) = \lambda_m \end{aligned} \tag{5.2}$$

where S_+^n is the cone of SDP matrices. The solution of this problem will be referred to as the calibrated correlation matrix from now on.

5.4.3 Direct approach: factor model based on principal components

To generate the factors for the standardized variables, we compute a spectral decomposition of the calibrated correlation matrix Γ by writing

$$\Gamma = \tilde{E}\tilde{\Lambda}\tilde{E}^T \tag{5.3}$$

where $\tilde{E}$ is the matrix of eigenvectors and $\tilde{\Lambda}$ the diagonal matrix of associated eigenvalues ordered decreasingly.

We define the set principal of factors F by

$$\tilde{F} = \tilde{E}^T\tilde{R} \tag{5.4}$$

in such a way that, by construction, we have:

$$\begin{aligned} Cov(F) &= \tilde{E}^T cov(\tilde{R})\tilde{E} && (5.5)\\ &= \tilde{E}^T\Gamma\tilde{E} = \tilde{\Lambda}. && (5.6)\end{aligned}$$

5.4.4 Subspace Approach : from Local to Global Factors

To deal with a large-scale porfolio, typically 3000 to 5000 assets, we can divide it into submarkets (or blocks) and apply for each block a local PCA approach to generate principal factors that embed enough information and are independent within their submarket. The reconstruction of a global set of factors at the portfolio level can be done by taking into account the inter-block correlations.

To describe the block-approach, let us introduce a few notations.

A/ Denote by

- $\tilde{R}_1, \tilde{R}_2, ..., \tilde{R}_p$ the blocks of standardized returns with covariance/correlation $\tilde{\Gamma}_i$, cross-correlation $Cov(\tilde{R}_i, \tilde{R}_i) = Corr(\tilde{R}_i, \tilde{R}_i) = \tilde{\Gamma}_{ij}$;
- $\tilde{F}_1, \tilde{F}_2, ..., \tilde{F}_p$ the blocks of factors associated to the standardized returns, each block having a dimension q; by $\tilde{\Gamma}^F = Corr(\tilde{F}_1, \tilde{F}_2, ..., \tilde{F}_p)$ their correlation matrix and by $\tilde{\Sigma}^F$ their covariance matrix;
- $\tilde{E}_1, \tilde{E}_2, ..., \tilde{E}_p$ the blocks of the q first eigenvectors from the spectral factorization of the block-correlations and $\tilde{\Lambda}_1, \tilde{\Lambda}_2, ..., \tilde{\Lambda}_p$ the blocks of the corresponding principal q first eigenvaleues ordered decreasingly.

B/ Denote also by

- $R_1, R_2, ..., R_p$ the blocks of original returns with mean m_i, covariance Σ_i and cross-covariance Σ_{ij};
- $F_1, F_2, ..., F_p$ the blocks of factors associated to the original returns;
- $E_1, E_2, ..., E_p$ the blocks of the q first eigenvectors from the spectral factorization of the block-covariances Σ_i;
- $S_i = Diag(\Sigma_i)^{\frac{1}{2}}$ the volatility of the i^{th} block of returns ($R_i = m_i + S_i\tilde{R}_i$).

Now, we define the principal factors of the standardized returns by

$$\tilde{F}_i = \tilde{E}_i^T \tilde{R}_i \tag{5.7}$$

in such a way that we have by construction:

$$\begin{aligned} Cov(\tilde{F}_i) &= \tilde{E}_i^T Cov(\tilde{R}_i) \tilde{E}_i & (5.8)\\ &= \tilde{E}_i^T \Gamma_i \tilde{E}_i = \tilde{\Lambda}_i. & (5.9) \end{aligned}$$

Therefore, from the equality

$$Cov(\tilde{F}_i, \tilde{F}_j) = \tilde{E}_i^T Cov(\tilde{R}_i, \tilde{R}_j) \tilde{E}_j = \tilde{E}_i^T \Gamma_{ij} \tilde{E}_j \tag{5.10}$$

we can deduce that the correlation between the blocks of factors is given by

$$\begin{aligned} \tilde{\Gamma}_{ij} &= Corr(\tilde{F}_i, \tilde{F}_j) = \Sigma_i^{\frac{-1}{2}} Cov(\tilde{F}_i, \tilde{F}_j) \Sigma_j^{\frac{-1}{2}} \\ &= \tilde{\Lambda}_i^{\frac{-1}{2}} Cov(\tilde{F}_i, \tilde{F}_j) \tilde{\Lambda}_j^{\frac{-1}{2}} \\ &= \tilde{\Lambda}_i^{\frac{-1}{2}} \tilde{E}_i^T \Gamma_{ij} \tilde{E}_j \tilde{\Lambda}_j^{\frac{-1}{2}} & (5.11) \end{aligned}$$

Eventually, we may point out that using the system of reduced factors $\tilde{F}_i$, we can build a simulation model of the form

$$R_i = S_i \tilde{E}_i \tilde{F}_i + \epsilon_i \tag{5.12}$$

where the random variables ϵ_i are the redidual modeling errors.

5.4.5 Implementation

Here after, we describe the simulation model assuming that we can neglect the residual errors. In that case, assuming that the returns are Gaussian, the implementation of the simulation can be summarized in the following four steps:

Step 1 Divide the portfolio in p blocks.

Step 2 For each bloc i.

- **2a.** *Make sure that the correlation matrix is Semi-Definite Positive.* This is a critical step that requires to solve a Semi-Definite Programming (SDP) problem. This step is necessary because in practise the number of historical quotations is by far smaller that the size of the blocks and moreover other numerical procedures such as exponential decay or returns rollover introduce negative eigenvalues.
- **2b.** Compute the spectral decomposition of the correlation matrix;
- **2c.** Scale the eigenvalues of the correlation matrix by the volatilities of the returns;

Step 3 For each bloc i, choose between:

- **3a.** Either use directly the factors of the standardized returns following $N(0, \tilde{\Lambda})$.
- **3b.** Or scale again the eigenvectors by $\tilde{\Lambda}^{\frac{-1}{2}}$.

Step 4 At the Global Portfolio Level: To simulate the returns of the original variables, we have to correlate the different blocks using $\tilde{\Gamma}^F$, i.e.

4a. Initialize the blocks with random candidates independently generated and bundle them together to have one single block $\tilde{Z}_F$;

4b. Make sure that the factors correlation matrix is definite positive (which requires to solve a SDP problem);

4c. Compute the Choleski factor $\tilde{C}_F$ of the correlation matrix $\tilde{\Gamma}^F$, and apply $\tilde{C}_F$ to the matrix $\tilde{Z}_F$ of previously generated candidates to get the final set of factors with correlation diffusion of global factors $\tilde{\Gamma}^F$;

4d. Eventually add the mean m_i of the original returns to the simulated returns R_i.

5.4.6 Related functions

The related functions of Wall Risk Engine® are:

Function	Description
WREmodelingBlockVaR	Monte Carlo VaR Simulation by "Submarket Approach"
WREmodelingBlockXVaR	Monte Carlo VaR Simulation by "Submarket Approach" with a different number of factors for each market block
WREmodelingMonteCarloVaR	Monte Carlo VaR Simulation by "Direct Approach"

5.5 Examples

5.5.1 Portfolio optimization with a structured product

The simulation module, along with the allocation functionalities, is used in this example to analyze a structured product in an investment context: what would be the effect of this structured product in a classical investment universe? To what extent would it contribute to the diversification of the portfolio?

The initial universe considered here is composed of a macro-economic index (the MSCI) and a risk-free rate. The study takes place in a 10-years horizon allocation context:

- we simulate the macro-economic factor MSCI and the structured product until maturity,
- we compute the efficient frontier (using Wall Risk Engine® robust Markowitz allocation method) on the two following universes:
  - MSCI, risk-free rate,
  - MSCI, risk-free rate and the structured product,
- then we compare the properties of the optimal portfolios in those two universes (in term on performance and risk).

A first quantitative analysis is performed on the results of the simulation: we compare the simulated returns of the MSCI and the structured product, using Wall Risk Engine® Performance and Risk Analysis module. This analysis shows that the structured product has good properties compared to the MSCI equity Index (lower volatility and better Sharpe ratio):

	MSCI WORLD	Structured Product
Mean	9,57%	**7,17%**
Volatility	10,49%	**2,54%**
Sharpe ratio	0,626	**1,643**

We use the results of the simulations to compute correlations and perform optimal allocations (using robust Markowitz allocation method) for different target returns on the two universes described above. The figure hereafter represents the efficient frontier (built with the *allocMV* function of Wall Risk Engine®) of the investment universe with and without the structured product. It shows that the introduction of the structured products implies a noticeable return/risk ratio gain.

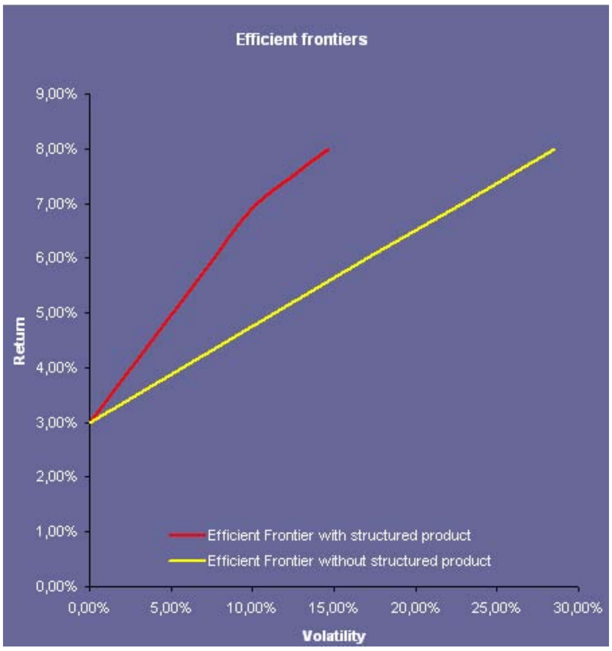


Figure 5.1: Impact of the structured product on the efficient frontier of the investment universe

5.5.2 Large-scale Monte-Carlo VaR for energy contracts

We have considered a Portfolio of 2858 Forward gas contracts and run the complete VaR computation process on a Linux Machine (Intel(R) Xeon5TM) CPU 3.00GHz / Cache size 2048 KB / Linux version 2.6.28-11 Ubuntu 4.3.3).

Using the block-approach for a 5-factor model, we have obtained the following computational times (all the way from the computation of correlations to the shock generation):

- Computation time for market segmentation in 29 blocks (submarkets) each block of dimension 100 assets : $<$ 1 minute;
- Computation time for market segmentation in 3 blocks (submarkets) each block of dimension 1000 assets : $<$ 15 minutes.

In comparison, the direct approach (with no block) for a 5 factors model takes 1h20mns from back to end.

Hereafter, we provide some display of the volatiliy error a porfolio of 2858 assets simulated with 5000 shocks.

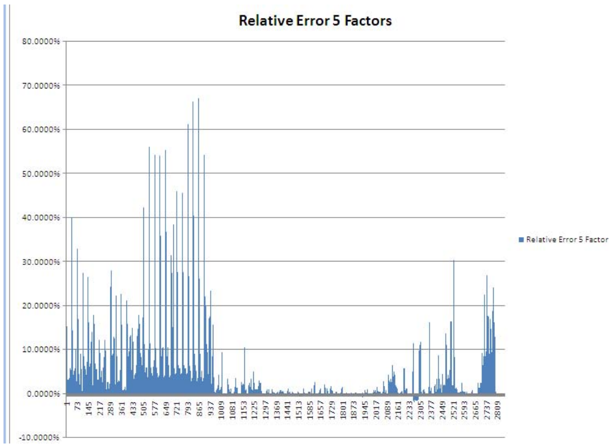


Figure 5.2: Relative Volatility Error for 5 factors

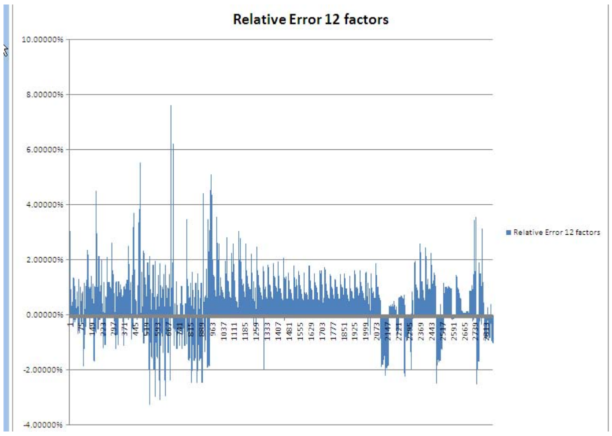


Figure 5.3: Relative Volatility Error for 12 factors

6 Quant Data Preparation (Available January 2014)

6.1 Univariate outlier detection

The method used here is a univariate method named MAD method[BG05], using the median and the Median Absolute Deviation (MAD) as measures of location and dispersion. It is known to be a robust method largely unaffected by the presence of extreme values of the data set (breakdown points of median and MAD are around 50% meaning that they are robust while the percentage of outliers is below 50%).

- $MED = median(x)$ is the mid of the series sorted in ascending order (the mean of two central observations if the series is even)
- $MAD = median(|x_i - MED| \ / \ i = 1, 2, ..., n)$

A measure of the outliers degree of contamination (outlyingness) of the data x_i is then determined by the following formula: $|x_i - median(x)|$.

The decision rule to determine if x_i is outlier is:

- If $|x_i - median(x)| \leq r \cdot MAD, x_i$ is not outlier
- If $|x_i - median(x)| > r \cdot MAD, x_i$ is outlier

where r is a conservative parameter.

This method is also known as an efficient (not time consuming) outliers detection method in one dimensional data.

6.2 Data cleaning

The EM algorithm

Wall Risk Engine® risk and performance modeling functionalities can deal with incomplete data series and build consistent and robust risk and performance models taking into account this lack of information (as explained above). But data completion can be useful in some cases. To recover the missing observations in a incomplete time series, this module embeds a state-of-the-art data completion method, the Expectation-Maximization (EM) algorithm.

The EM algorithm [Bor04] is an efficient iterative procedure to compute the Maximum Likelihood (ML) estimate in the presence of missing or hidden data. In ML estimation, we wish to estimate the model parameters for which the observed data are the most likely.

Each iteration of the EM algorithm consists of two processes:

- In the expectation step (or E-step), the missing data are estimated given the observed data and current estimate of the model parameters. This is achieved using the conditional expectation, explaining the choice of terminology.
- In the maximization step (or M-step), the likelihood function is maximized under the assumption that the missing data are known. The estimate of the missing data from the E-step are used instead of the actual missing data.

Convergence is ensured since the algorithm is guaranteed to increase the likelihood at each iteration.

Related functions

The related functions of Wall Risk Engine® are:

Function	Description
WREmodelingEM	Expectation-Maximization algorithm to recover missing observations

Bibliography

[BG05] Irad Ben-Gal. Outlier detection. In Maimon O. and Rockach L., editors, *Data Mining and Knowledge Discovery Handbook: A Complete Guide for Practitioners and Researchers*. Kluwer Academic Publishers, 2005. ISBN 0-387-24435-2.

[Bor04] S. Borman. The expectation maximization algorithm: a short tutorial, 2004.

[BS90] A. C. Brandwein and W. E. Strawderman. Stein estimation: the spherically symmetric case. *Statistical Science*, 5(3), 1990.

[CF37] E.A. Cornish and R.A. Fisher. Moments and cumulants in the specification of distributions. *Review of the International Statistical Institute*, pages 307–320, 1937. Reprinted in Fisher, R. A. Contributions to Mathematical Statistics. New York: Wiley, 1950.

[Duf88] D. Duffie. *Security Markets: Stochastic Models*. Academic Press, 1988.

[GI03] D. Goldfarb and G. Iyengar. Robust portfolio selection problems. *Math. Operations Research.*, 28(1):1–38, 2003.

[Gla04] P. Glasserman. *Monte Carlo Methods in Financial Engineering*. Springer, 2004.

[HO00] C. Helmberg and F. Oustry. Bundle methods to minimize the maximum eigenvalue function. pages 307–337, 2000.

[Jor86] P. Jorion. Bayes-stein estimation for portfolio analysis. *The Journal of Financial and Quantitative Analysis*, 21(3):279–292, 1986.

[KS91] I. Karatzas and S.E. Shreve. *Brownian Motion and Stochastic Calculus*. Springer, 2 edition, 1991.

[LEGL98] F. Oustry L. El Ghaoui and H. Lebret. Robust solutions to uncertain semidefinite programs. *SIAM J. Optimization*, 9(1), 1998.

[LEGO03] M. Oks L. El Ghaoui and F. Oustry. Worst-case value at risk and robust portfolio optimization: A conic programming approach. *Operations Research*, 51(4):543–556, 2003.

[LLP99] J. P. Bouchaud L. Laloux, P. Cizeau and M. Potters. Noise dressing of financial correlation matrices. *Physical Review Letter*, 83:1467 – 1470, 1999.

[Lue98] D. G. Luenberger. *Investment Science*. Oxford University Press, New York, 1998.

[LW03] O. Ledoit and M. Wolf. Improved estimation of the covariance matrix of stock returns with an application to portfolio selection. *Journal of Empirical Finance*, 10(5):603–621, 2003.

[Mar59] H. M. Markowitz. *Portfolio Selection: Efficient Diversification of Investments*. New York: John Wiley and Sons, 1959. reprinted by Yale University Press, 1970, ISBN 978-0300013726; 2nd ed. Basil Blackwell, 1991, ISBN 978-1557861085.

[Meu05] A. Meucci. *Risk and Asset Allocation*. Springer Finance, 2005.

[MS04] Danilo Mercurio and Vladimir Spokoiny. Statistical inference for time-inhomogeneous volatility models. 2004.

[MS05] Y. Malevergne and D. Somette. *Extreme Financial Risks: From Dependence to Risk Management*. Springer Finance, 2005.

[PEK99] E. Platen P. E. Kloeden. *Numerical Solution of Stochastic Differential Equations*. Springer, 1999.

[PMNP52] F. Oustry P. M. Ndiaye and V. Piolle. Portfolio selection. *The Journal of Finance*, 7(1):7791, March 1952.

[PMNP06] F. Oustry P. M. Ndiaye and V. Piolle. Semidefinite optimisation for global risk modelling. *Journal of Asset Management*, 7:142–153, July 2006.

[Ros76] S. Ross. The arbitrage theory of capital asset pricing. *Journal of Economic Theory*, 13(3), 1976.

[SS03] S. Stephen and A. Scowcroft. *Advances in portfolio construction and implementation*. Butterworth-Heinemann, 2003.

[ZKR09] Steve Zymler, Daniel Kuhn, and Berc Rustem. Worst-case value-at-risk of non-linear portfolios. Working Papers 017, COMISEF, 2009.

Copyright