Aalto University
School of Science

Final Report

MS-E2177, Seminar on Case Studies in Operations Research
May 30, 2022

Lauri Jokinen	lauri.jokinen@aalto.fi	(project manager)
Christian Segercrantz	christian.segercrantz@aalto.fi	
Ilmari Vauhkonen	ilmari.vauhkonen@aalto.fi	

Client: Aktia Life Insurance

Contents

1 Introduction **2**

2 Background **3**
- 2.1 Assets and scenarios 3
- 2.2 Views 6
- 2.3 Entropy minimization 7
- 2.4 Converting the views to linear constraints 9
- 2.5 Markowitz portfolio optimization 9

3 Data and methods **13**
- 3.1 Input data 13
- 3.2 Views 13
- 3.3 Example Portfolios 14
  - 3.3.1 Example Portfolio 1 14
  - 3.3.2 Example Portfolio 2 14
- 3.4 Algorithm 15

4 Results **16**
- 4.1 The posterior distributions 16
- 4.2 Optimal portfolio allocation 18
  - 4.2.1 Example portfolio 1 18
  - 4.2.2 Example portfolio 2 18

5 Discussion **21**

A Views filling instructions **23**

B Variance approximation **24**

C Stacked portfolio graphs **25**

1 Introduction

In investing, one has two major tasks: simultaneously trying to achieve increase in portfolio value (maximize returns) and protect the portfolio from large sudden declines in value (minimize the risk). These two objectives are however connected, e.g., to increase the expected returns, one must take additional risk. The trade-off between expected returns and risk is chosen by the investor.

In reality, there is a great number of interrelated assets available for trading: stocks, bonds, commodities, real estate, derivatives, and so on, which can be related to each other in some way. One could pose a question: would it be possible to take advantage of the dependencies between assets to reduce risks of the portfolio while still maintaining the desired level of returns?

A seminal milestone in portfolio optimization was laid by Harry Markowitz in 1952 [1]. His model considers the expected returns and volatilities of assets, as well as the covariances between different assets. This model helps to find the portfolio of assets that has minimum possible volatility while still yielding the desired level of expected returns.

The Markowitz theory does not account for external factors that are not visible in the asset sphere included in the model, as it assumes constant expected returns, volatility, and cross-asset correlation. For example, changing inflation or economic activity cannot be expressed in the model just by changing parameters, but every single asset parameter should be adjusted. For example, inflation affects various industries in a different way (e.g. raw material producers versus end product manufacturers), so changes in inflation could affect not only the expected returns and volatilities of the assets but also correlations between asset returns. Also, the optimal portfolio in the Markowitz model is very sensitive to input data: with tiny changes in data, the model may suggest reinvesting large amounts of money for a tiny improvement in returns.

An advancement in portfolio optimization was the Black-Litterman model which deals with so-called *views* [2]. The views are typically relative relations between two assets, and they have uncertainty factors assigned to them. Now, the market data is processed and filtered in a way that the views hold and we optimize the portfolio based on this data. The Black-Litterman model also deals with the mean and variance of returns, and the uncertainties of the views are assumed to be normally distributed. The Black-Litterman model has a closed-form solution for the optimal portfolio and is very popular today.

The two models have been criticized because they assume that the portfolio's risk is the variance of returns. This means, e.g., that both greater and lower returns are marked as risks, while only the lower returns should be avoided. [3] Thus, an investor would benefit from being able to accommodate their definitions of risk. For more precise portfolio optimization we should consider non-parametric distributions. A non-parametric distribution captures a more precise picture of the market and it also handles outliers better.

In his paper, Meucci proposes a new method, called *Entropy Pooling*, which is based on the Black-Litterman method but generalizes it to non-parametric distributions. [4] The paper introduces the steps of forming a posterior distribution from the prior market distribution that takes the views into account, but disrupts the prior distribution as little as possible. This is accomplished by minimizing the entropy between the two distributions. The optimal portfolio is solved numerically by Bayesian optimization and the Monte Carlo method.

However, the model has its drawbacks. For example, with extremely rare events, the posterior distribution is formed with a small amount of data which ultimately leads to inaccurate results. All the models discussed here are based on historical data, i.e., they do not forecast the market.

Portfolio optimization is a very much studied topic, and so it is very difficult to create portfolios that would outperform the existing ones. However, we may have information (views) that help build better-performing portfolios. It is important to address the importance of the methods of how the views are gathered from experts. When responding to certain types of questions, humans are prone to show strong biases. Also, experts have different systematic biases that need to be tested and calibrated.

The client organization, Aktia Life Insurance, is interested in applying an Entropy Pooling approach to its portfolio to manage its assets. For the client, a Python library performing the Entropy Pooling and Markowitz optimizations to find an optimal asset portfolio (in terms of expected returns and volatility) is created. This report focuses on the mathematical background of the procedure, its advantages, and drawbacks, as well as the results obtained by the implemented program.

The rest of the work is structured as follows: Section 2 goes through the main concepts and mathematics used in constructing the scenarios and views and then proceeds to explain the entropy minimization and Markowitz portfolio optimization methodologies. Section 3 introduces the data used in this project and the implemented algorithm in more detail. The results are discussed in Section 4, and Section 5 concludes.

2 Background

This section introduces the mathematical theory behind the entire Entropy Pooling and portfolio optimization process. First, the concepts of scenarios and views are presented, and both their meaning and notation are explained. Then, the idea of using Entropy Pooling to adjust the probability distribution of the scenarios is shown. Finally, the Markowitz portfolio optimization procedure, adapted for using the scenario data and probabilities, is described. Figure 1 shows a simple flow chart of the program.

2.1 Assets and scenarios

Scenarios denote possible future changes in the value of *factors*, $f \in F$, which resemble various indicators of a macroeconomic situation. In this setting, we assume F to be finite and discrete. So, one scenario $s \in S$ is the set $\{d_s^f : f \in F\}$ that contains the changes in the values of all factors that occur (more precisely, are expected to occur) simultaneously. Note the units of d: for stock indexes, the unit is price changes in percentages, for bonds, absolute change in the yield (must be scaled by $1/100$ to get percentage points instead of base points), and for other indicators (such as unemployment rate), the change in percentage points.

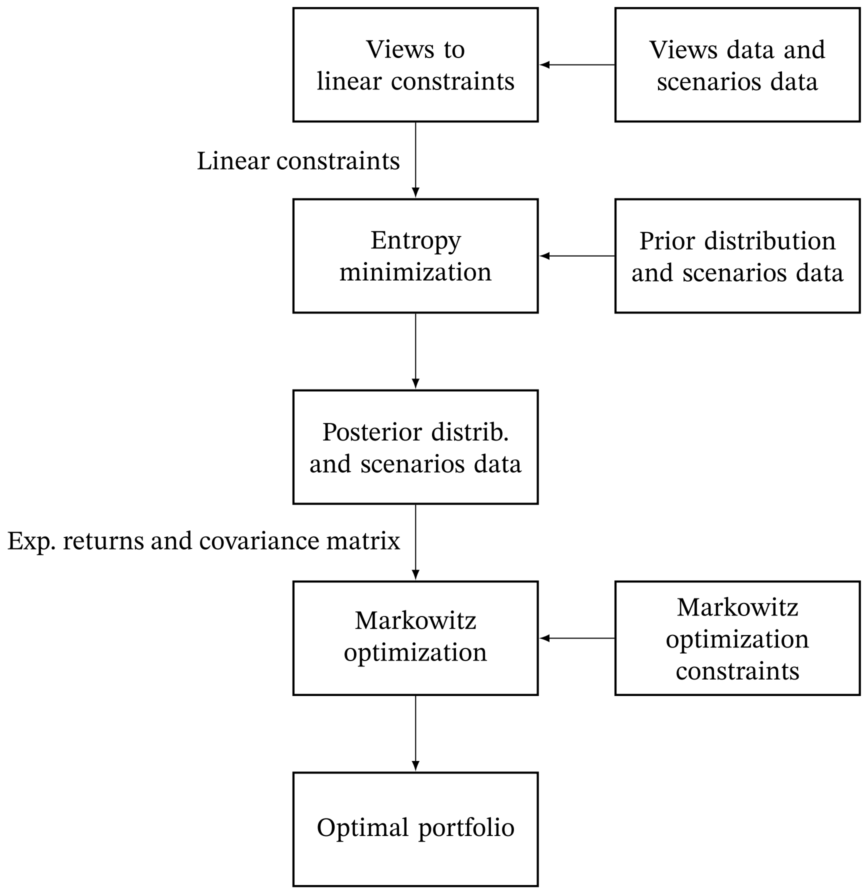


Figure 1: A flow chart of the program. On the right-hand side, we have data provided by the user, and on the left, the data is processed.

Table 1: An example of a factor scenario matrix (a subset of the data used in this project).

Scenario	European Equities	EUR/USD FX Rate	US 10-Year Treasury Yield	Eurozone Core Inflation
1	5.42	0.28	6.58	0.40
2	−2.53	−1.90	−10.78	0.00
3	4.67	−0.19	6.48	0.30
4	−3.29	−1.94	17.85	0.50
5	2.18	−0.51	8.65	0.30

The factor scenario matrix is defined as follows,

$$S_F = \begin{bmatrix} d_1^1 & d_1^2 & \dots & d_1^{|F|} \\ d_2^1 & d_2^2 & \dots & d_2^{|F|} \\ \vdots & \vdots & \ddots & \vdots \\ d_{|S|}^1 & d_{|S|}^2 & \dots & d_{|S|}^{|F|} \end{bmatrix} \in \mathbb{R}^{|S|\times|F|}, \tag{1}$$

where $|\cdot|$ represents the cardinality of the set. An exemplary subset of the real, much larger factor scenario matrix is in Table 1. Each row represents one scenario, so this matrix represents five future states of the world. In, say, future state (scenario) 3, the European equities would rise 4.67 %, the EUR/USD exchange rate would fall 0.19 %, the US 10-year treasury yield would rise 6.48 basis points (corresponding to 0.0648 percentage points), and the Eurozone core inflation measure would rise 0.3 percentage points.

To create the asset-specific return scenarios for the Markowitz optimization phase, one can use the information on the assets' sensitivities to changes in the values of the factors. The linearly approximated sensitivity of the value V of asset $i \in N$ to a factor f is called the delta of the asset,

$$\Delta_i^f = \frac{\partial V_i}{\partial f}. \tag{2}$$

Thus, the delta is simply the derivative of the asset value with respect to the factor. Note that the asset may be sensitive to many factors simultaneously. Thus, we list the sensitivities of all assets, $i \in N$, to all factors in a so-called delta matrix of dimension $\mathbb{R}^{|F| \times |N|}$:

$$\Delta = \begin{bmatrix} \Delta_1^1 & \Delta_2^1 & \dots & \Delta_{|N|}^1 \\ \Delta_1^2 & \Delta_2^2 & \dots & \Delta_{|N|}^2 \\ \vdots & \vdots & \ddots & \vdots \\ \Delta_1^{|F|} & \Delta_2^{|F|} & \dots & \Delta_{|N|}^{|F|} \end{bmatrix}. \tag{3}$$

Table 2 shows an example of an asset delta matrix for four factors and three assets. The factors are as in Table 1. The sensitivity of Nokia stock with respect to the European Equities Index (a proxy for the market) can be understood as its CAPM beta. The changes in the value of a USD cash position are inversely proportional to the changes in the FX Rate (rising EUR/USD rate implies a depreciation of the dollar). The relative bond price sensitivity to changes in the yield is obtained as the negative of the modified duration. Modified duration is defined as $-(\partial V/\partial y)/V$, where V is the bond value and y is the yield, so it is the negative of the percentage change of the bond value with respect to changes in yield. Also note, that in this linear approximation, none of the assets are directly dependent on Eurozone inflation rates.

Table 2: An example of the asset delta matrix.

Factor	Nokia Stock	USD Cash	US 10-Year Treasury Note
European Equities	0.7	0	0
EUR/USD FX Rate	0	–1	0
US 10-Y Tr. Yield	0	0	–10
Eurozone Inflation	0	0	0

In this case, the returns of the asset i with respect to changes in all factors in a scenario s is

$$r_s^i = \sum_{f \in F} d_s^f \Delta_i^f. \tag{4}$$

Note that this summation corresponds to the dot product of the row s of S_F and column i of Δ. Thus, the scenarios of all assets in all scenarios can therefore be represented very conveniently as a matrix product of S_F and Δ. We call the $|S|$-by-$|N|$ product matrix asset return matrix R,

$$R = S_F \Delta = \begin{bmatrix} r_1^1 & r_1^2 & \dots & r_1^{|N|} \\ r_2^1 & r_2^2 & \dots & r_2^{|N|} \\ \vdots & \vdots & \ddots & \vdots \\ r_{|S|}^1 & r_{|S|}^2 & \dots & r_{|S|}^{|N|} \end{bmatrix}. \tag{5}$$

The matrix product of Tables 1 and 2 is in Table 3. The table should be interpreted the same way as the factor scenario matrix, but now all elements are percentage returns: for example, in scenario 3, Nokia stock would rise 3.27 %, USD cash position would appreciate 0.19 %, and a 10-Year US T-Note would lose 0.65 % of its value.

Table 3: An example of an asset scenario matrix.

Scenario	Nokia Stock	USD Cash	US 10-year Treasury Note
1	3.79	–0.28	–0.66
2	–1.77	1.90	1.08
3	3.27	0.19	–0.65
4	–2.30	1.94	–1.79
5	1.53	0.51	–0.87

We must highlight that this is only a linear approximation, and in real life, the dependence between assets and factors may be nonlinear. Also, possible multicollinearity between the factors causes errors in the estimates of R. The responsibility of detecting and addressing multicollinearity is left to the user.

2.2 Views

In addition to the scenarios, the user can specify certain market characteristics he/she finds important. For example, the user may think that a given factor has some expected future value or volatility, or two factors exhibit a given level of correlation. These kinds of expectations, called *views*, could be e.g. reflected in current market prices of assets. Examples of views are:

1. The annual expected return of the European equities index is 5 percent.
2. The annual expected volatility of the European equities index is less than or equal to the annual volatility of Global equities minus 10 percent.
3. The correlation between European equities and global equities index returns is 0.9.

We allow various types of views: mean, volatility and correlation. Furthermore, we admit equality and inequality views, and absolute and relative views. The user inputs the views data in a spreadsheet format and the figures are given in the annual form. Detailed filling instructions and examples are provided in Appendix A. Even though we are keen on providing a user-friendly interface for the user, the user needs to be very careful when dealing with the views.

2.3 Entropy minimization

The fundamental idea of the entropy minimization approach is to use prior knowledge of the probability distribution of scenarios (prior distribution) and additional information about possible future outcomes (views) to find an adjusted probability distribution (posterior distribution) that deviates as little as possible from the prior one. The measure (or proxy) of deviation between the two distributions is called relative entropy, or Kullback-Leibler divergence [5]. The expression of relative entropy is frequently used, e.g., in the field of information theory. It can be expressed either in continuous or discrete form,

$$H_C(p,p') = \int_S p(s)\Big(\ln p(s) - \ln p'(s)\Big)\mathrm{d}s, \tag{6}$$

$$H(p,p') = \sum_{s\in S} p_s(\ln p_s - \ln p'_s). \tag{7}$$

So, the problem of finding the entropy-minimizing distribution p using the prior distribution p' and the views $Ap \leq b$ can be expressed as

$$\min_p \; H(p) \tag{8}$$

$$\text{s.t. } Ap \leq b, \tag{9}$$

$$p_s \geq 0, \; \forall\, s \in S, \tag{10}$$

where notation $H(p)$ highlights that the objective function is a function of the posterior distribution only. The inequality in $Ap \leq b$, applies to each row independently. The constraint equation $Ap \leq b$ must contain $\sum_{s\in S} p_s = 1$ to ensure that p satisfies the properties of a discrete probability distribution.

The complexity of this optimization problem depends on two factors: the number of scenarios ($|S|$) and the number of constraints ($|\Lambda|$ with Λ being the set of constraints). The number of scenarios is likely much higher than the number of constraints, thus making it beneficial to use a dual of the problem which reduces the number of decision variables from $|S|$ to $|\Lambda|$.

We start by forming the Lagrangian function, which combines the constraints and the original objective function,

$$L(p,\lambda) = H(p) + \lambda^\top(Ap - b) = p^\top(\ln p - \ln p') + \lambda^\top(Ap - b), \tag{11}$$

where $\ln x = [\,\ln x_1 \;\; \ln x_2 \;\; \cdots \;\; \ln x_S\,]^\top$.

The solution to the original problem is found by minimizing the Lagrangian function with respect to p and λ jointly. The function is now unrestricted with respect to p, but we have $\lambda \geq 0$, since the Lagrangian function penalizes the violation of the constraints $Ap - b \leq 0$, and any violation ($Ap - b > 0$) must increase the value of the objective (Lagrangian) function.

Now that the Lagrangian function is obtained, we transfer the problem into dual space. The dual objective function is $D(\lambda) = L(p^*(\lambda), \lambda)$, where,

$$p^*(\lambda) = \operatorname*{argmin}_p L(p,\lambda). \tag{12}$$

In other words, $D(\lambda)$, is the value of the Lagrangian when it is minimized with respect to p, with a fixed λ. So, the summation $H(p)$, can be represented more compactly as a dot product.

As derived in [4], the value of p minimizing the Lagrangian function is

$$p^*(\lambda) = \exp\{\ln p' - 1 - A^\top \lambda\}, \tag{13}$$

where the exponentiation is defined as $\exp\{x\} = [\exp\{x_1\} \quad \exp\{x_2\} \quad \cdots \quad \exp\{x_S\}]^\top$. Due to the property of exponential function $\exp\{x\} > 0,\ \forall x \in \mathbb{R}$, the constraint $p \geq 0$, is satisfied by default. Thus, the dual function itself becomes,

$$\begin{aligned} D(\lambda) &= p^{*\top}(\lambda)(\ln p^*(\lambda) - \ln p') + \lambda^\top (Ap^*(\lambda) - b) \\ &= \exp\{\ln p' - 1 - A^\top \lambda\}^\top \left(\ln \exp\{\ln p' - 1 - A^\top \lambda\} - \ln p' \right) \\ &\quad + \lambda^\top (A \exp\{\ln p' - 1 - A^\top \lambda\} - b), \end{aligned} \tag{14}$$

which simplifies to,

$$D(\lambda) = \exp\{\ln p' - 1 - A^\top \lambda\}^\top (-1 - A^\top \lambda) + \lambda^\top (A \exp\{\ln p' - 1 - A^\top \lambda\} - b). \tag{15}$$

Now, the dual optimization problem can be formulated as

$$\text{max. } D(\lambda) \tag{16}$$
$$\text{s.t. } \lambda \geq 0, \tag{17}$$

If the original problem is convex, the primal and dual problems have the same optimal point and value of the objective function. This is the case in our problem, as $H(p)$ is convex and the feasible set is also convex, due to linear constraints. Once an optimum λ^*, is obtained, the corresponding optimal value of the primal decision variable p^* is recovered using equation (13).

The dual problem can be solved with many optimization algorithms. Many of them require the gradient of $D(\lambda)$, which can be represented analytically,

$$\begin{aligned} \nabla D(\lambda) &= \nabla \left(p^{*\top} (\ln p^* - \ln p') + \lambda^\top (Ap^* - b) \right) \\ &= (\mathbf{1} + \ln p^*)^\top \nabla p^* - \ln p'^\top \nabla p^* + Ap^* + (A^\top \lambda)^\top \nabla p^* - b, \end{aligned} \tag{18}$$

where,

$$\nabla p^* = J(p^*) = -\text{diag}(\exp\{\ln p' - \mathbf{1} - A^\top \lambda\} A^\top, \tag{19}$$

is the Jacobian matrix of $p^*(\lambda)$ evaluated for λ. The equation above is not provided in [4], but plays a crucial part in the optimization algorithm.

The optimization problem will be solved using a truncated Newton (TNC) conjugate gradient algorithm which is an iterative algorithm that is able to solve high-dimensional nonlinear optimization problems [6]. Further consideration of a suitable algorithm is out of the scope of this project, but it is worth highlighting that the TNC algorithm allows for bounded decision variables, and it employs quadratic approximation of the objective function which usually implies relatively fast convergence.

2.4 Converting the views to linear constraints

The program converts the views data to optimization constraints which are used in entropy minimization. Furthermore, we only allow linear constraints. For views regarding mean values, the linear constraints are trivial to form. Consider a set of scenarios x_s of the value of an asset x, and weights p_s, where $s \in S$. Consider that we'd like to set the mean value of x to μ_x, which is provided by the user. This yields a linear equation,

$$\sum_{s \in S} p_s x_s = \mu_x,$$

which we can use with entropy minimization. A relative mean view between options x_s and y_s, $s \in S$, is rather easy to formulate to a linear form,

$$\sum_{s \in S} p_s x_s - \sum_{s \in S} p_s y_s = \mu_{xy} \quad \Longleftrightarrow \quad \sum_{s \in S} p_s (x_s - y_s) = \mu_{xy}.$$

However, problems arise with variance and covariance views,

$$\sum_{s \in S} p_s (x_s - \bar{x})^2 = \sigma_x^2,$$

where $\bar{x}$ is a weighted average,

$$\bar{x} = \sum_{s \in S} p_s x_s.$$

The equation is no longer linear, but note that by fixing $\bar{x}$, the equation becomes linear in p_s. The user may provide $\bar{x}$, in which case no approximations are needed. In the case where the user does not provide $\bar{x}$, we fix the value anyway, to a value that is close to the prior mean value of x_s, and satisfies all the views regarding mean values. We do a similar maneuver with correlation, for which both mean and variance are fixed.

The user defines the risk as standard deviation instead of variance. Relative volatility raises yet another problem, because squaring both sides,

$$\sqrt{s_x} - \sqrt{s_y} = \sigma_{xy} \implies s_x - 2\sqrt{s_x s_y} + s_y = \sigma_{xy}^2,$$

yields an additional term, $2\sqrt{s_x s_y}$, which is not present with absolute volatility. We approximate the term linearly in Appendix B, which yields the following approximation,

$$\sigma_{xy}^2 \approx \sum_{s \in S} p_s \left((x_s - \bar{x})^2 + (y_s - \bar{y})^2 - \frac{(x_s - \bar{x})^2 \sigma_y^2 + (y_s - \bar{y})^2 \sigma_x^2}{\sigma_x \sigma_y} \right),$$

where σ_x and σ_y are prior volatilities.

2.5 Markowitz portfolio optimization

To optimize the portfolio based on the posterior scenario probabilities, we use the Markowitz model, introduced by Harry Markowitz in 1952 [1]. The model is based on the following assumptions:

1. The risk of the portfolio is based on the variability of the returns of the portfolio.
2. The investor is risk-averse.

3. As a result, the investor's utility function is concave and increasing.
4. The investor favours increased consumption.
5. The analysis only takes into consideration a single period of investment.
6. The investor is rational.

With the model, we can, e.g., look for solutions that are optimal at some given risk level or return level. In this project, we minimize risk at a fixed minimum expected return level, which is defined by the user. However, the implementation can accommodate different strategies with fairly minor modifications to the code.

We model the problem as a quadratic optimization problem,

$$\begin{aligned} \text{min.} \quad & \mathbf{x}^\top \Sigma \mathbf{x} && (20) \\ \text{subject to:} \quad & \sum_{i=1}^{N} \mathbf{x}_i \leq 1 && (21) \\ & \mu^\top \mathbf{x} \geq \mu_0 && (22) \\ & \mathbf{x}_i \geq 0, \forall i, && (23) \end{aligned}$$

where $\mathbf{x}$ is a decision vector containing the resources allocated to each asset i. The vector $\mathbf{x}$ is also referred as *weights*. The vector μ contains the expected returns for each asset, Σ is the covariance matrix (or cross-correlation matrix) of the assets, and μ_0 is a minimum level of return, which is treated as a constant.

The objective function, see Equation (20), is the total variance of the portfolio's returns. Alternatively, we could minimize standard deviation (or, in this case, volatility) instead of variance. This yields an equivalent problem since taking a square root is a monotonic operation. Often, minimizing variance is less complex.

Equation (21) describes the constraint to keep the sum of the weights below one, i.e. not to use more resources than available (one would expect that all resources are used but we allow for this possibility as well). Equation 22 sets the minimum level of expected return. Since Equation (20) is a second-order equation and Equations (21)-(23) form half-spaces, the optimization problem is a convex problem. This important result ensures that a numerically found local minimum is a global minimum.

To solve the optimization problem, we use a quasi-Newton method, *Limited-memory Broyden–Fletcher–Goldfarb–Shanno method with box constraints* (L-BFGS-B) [7]. This method is natively implemented in SciPy which is convenient. Again, the testing of alternative optimization methods is out of this project's scope.

The model can be modified to accommodate additional constraints and modifications. Alterations considered in this project are listed below.

1. Short positions. This is achieved by either removing constraint (23) (short positions in all assets are allowed) or relaxing it as,

   $$\mathbf{x}_j \geq 0, \ \forall j \in J,$$

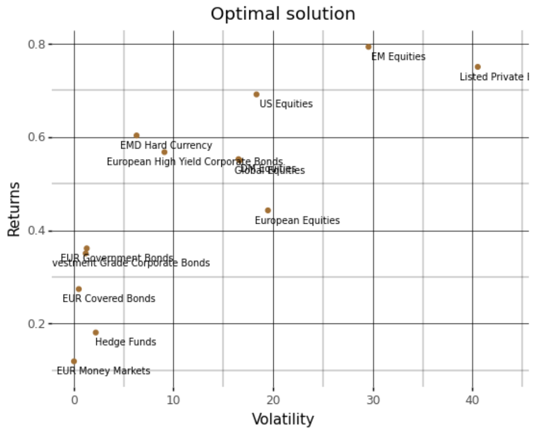
   where J is the set of asset for which short positions are not allowed.

2. Minimum and/or maximum position levels for individual assets or a subset of assets. This is represented as
$$\underline{c}^K \leq \sum_{i \in K} \mathbf{x}_i \leq \overline{c}^K,$$
where the set K represents the asset subset, and $\underline{c}^K$ and $\overline{c}^K$ are the given limits. With many such constraints, it is possible to form conflicting position requirements, which implies an infeasible optimization problem.

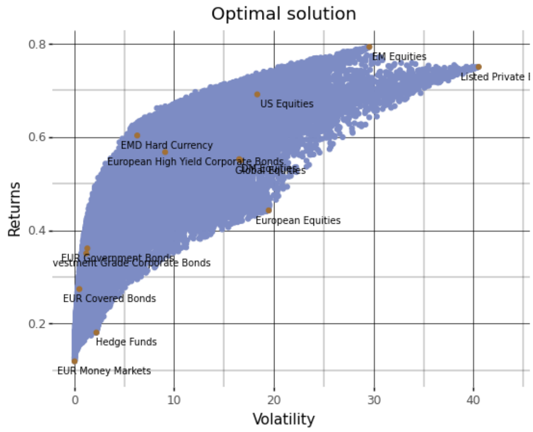
3. Including assets that do not require any capital invested at present time, such as swaps. Such assets can be included in the Markowitz model by altering the constraint (21) as follows,
$$\sum_{i \in \{1,\dots,N\} \setminus M} \mathbf{x}_i \leq 1,$$
where M represents the set of assets where no initial capital is invested.

The Markowitz optimization, or model, can also be viewed in a different way. The following steps lead to the same result as the above convex optimization problem. First, we take all the possible portfolios given our available assets, visible in 2a, an example is illustrated in Figure 2b. We then find those that have the lowest risk for each given return level. These portfolios define the efficient frontier, visible in Figure 2c. Finally, we choose a portfolio that satisfies our needs, e.g. at a certain return level. The final result, where the risk level μ_0 is set to 0.5, can be seen in Figure 2d.

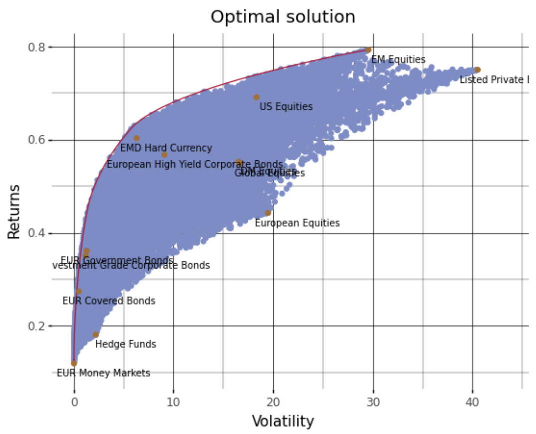
An important extension to the Markowitz mean-variance portfolio model is the inclusion of a risk-free asset that has zero volatility [8]. In this model, the optimal portfolios are convex combinations of the risk-free asset and an asset portfolio that maximizes the so-called Sharpe ratio. In this project, we do not consider risk-free investments. This is because the project's client is a bank and thus does not have the same freedom with risk-free assets compared to, e.g., a private person. Ultimately, a risk-free asset is a theoretical concept (or an approximation), because even the usual proxies used as risk-free investments (e.g., US government bonds) include some yield risk and thus their future value is not certain, in case an asset is not held all the way to its maturity.



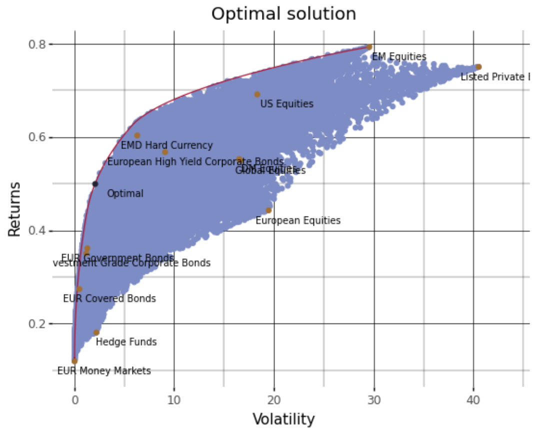
(a) The original assets plotted in the asset space.



(b) A blue cloud of dots representing the linear combinations of the assets.



(c) A red line, displaying the efficient frontier, added to the previous picture.



(d) The optimal portfolio plotted at some μ_0.

Figure 2: A series of plots, using a random scenario space, to describe the Markowitz optimization.

3 Data and methods

3.1 Input data

The code relies heavily on data given by the user. The user gives the following input matrices in Excel sheets: factor scenarios, views, asset deltas, and constraints for Markowitz optimization. In this project, the inputs are provided by the client. All of the provided data used in this project is simulated mock data, and it does not represent real historical data, real opinions nor expectations, or any sensitive information possessed by the client.

The factor scenario matrix provided by the client has dimension 252×36. That is, it includes 36 factors and 252 scenarios. The factors include major stock indexes (European, American, global, and emerging market equities), interest rates (US Treasury notes, German Bonds, European corporate bonds, swap rates, and the like), and major macroeconomic indicators such as EU and USA unemployment and inflation rates. The original data denotes monthly changes, but the values have been annualized by Aktia already.

3.2 Views

In total eight views were used to test their effect on the entropy-minimizing posterior distribution. The views are listed in Table 4, and an example input Excel sheet is shown in Table 5. In this problem setting, only inequality type constraints for expected values of changes in factors were considered. It was concluded that taking views regarding standard deviations or correlations of factors was not of great importance. Changes in posterior probabilities induced by enforcing a single view were considered a good approach for sensitivity analysis. In fact, trying out different views and their effect on the optimal portfolio allocation is one of the purposes of performing the entire Entropy Pooling procedure.

Table 4: The views that were used to find entropy-minimizing posterior distributions. The operator μ stands for expected value of change in factor value (absolute change or return depending on factor type). All the right-hand side numbers correspond to annualized returns or changes measures in percentage points (pp) or basis points (bp).

View name	Explanation		
Rates up	μ(Germany 10 Yr. Govt. Bond Yield)	$\geq$	2.5 bp
Rates down	μ(Germany 10 Yr. Govt. Bond Yield)	$\leq$	-2.5 bp
Equities up	μ(Global Equities)	$\geq$	1 %
Equities down	μ(Global Equities)	$\leq$	-1 %
Inflation up	μ(Eurozone Core Inflation)	$\geq$	0.1 pp
Inflation down	μ(Eurozone Core Inflation)	$\leq$	-0.1 pp
VIX up	μ(iVol US Equities)	$\geq$	1 pp
VIX down	μ(iVol US Equities)	$\leq$	-1 pp

Table 5: Example views Excel sheet, for the case *Equities up*.

* View on	* Risk factor 1	Risk factor 2 (applicable for corr)	* Operator	* Constant (alpha)
Mean	Global Equities	-	>	1 %

3.3 Example Portfolios

We next describe two example portfolios with different assets and additional constraints. These portfolios are later optimized using different distributions.

3.3.1 Example Portfolio 1

The first example portfolio used for analysis consists of three assets: a European government bond portfolio, a European investment-grade corporate bond portfolio, and a European 5v5 swap agreement.

The asset delta matrix used here is shown in Table 6. Note that the deltas with respect to bond indexes are not computed using the modified duration as was done in Table 2, as these two bond indexes are denoted in terms of total percentage returns instead of yield. (That is the case with US 10-year treasury notes. This is simply a peculiarity of the given data).

Table 6: The asset delta matrix of Example Portfolio 1. The column 'Other Factors' specifies that none of the assets has a nonzero delta with respect to any of the remaining 33 factors.

Factor	European Govt. Bonds	European Corp. Bonds	European 5v5 Swap	Other Factors
EUR Govt. Bonds	1	0	0	0
EUR Inv. Grade Corp. Bonds	0	1	0	0
10Y EUR SWAP	0	0	2	0
5Y EUR SWAP	0	0	−1	0

The following constraints were included to specify certain requirements that the portfolio manager might want to take into account:

1. The amount of capital invested in the European government bond portfolio must be between 0.5 and 1 million Euros.
2. The amount of capital invested in the European corporate bond portfolio must be between 0 and 0.5 million Euros.
3. The total amount of capital invested in both bond portfolios must be equal to one million Euros.
4. The notional capital of the 5v5 swap agreement must be between 0 and 100,000 Euros (1,000 Euros per basis point).

These constraints are represented in matrix form as, $\underline{b} \le Ax \le \bar{b}$, where,

$$A_1 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 1 & 0 \end{bmatrix}, \quad \underline{b}_1 = \begin{bmatrix} 500{,}000 \\ 0 \\ 0 \\ 1{,}000{,}000 \end{bmatrix}, \quad \bar{b}_1 = \begin{bmatrix} 1{,}000{,}000 \\ 500{,}000 \\ 100{,}000 \\ 1{,}000{,}000 \end{bmatrix}. \tag{24}$$

3.3.2 Example Portfolio 2

The second example portfolio to be tested consisted of 13 factors that could be understood as index assets, plus two derivatives contracts that have fixed positions. So, each 'normal' asset has a unit delta with respect to itself (when considered as factors), whereas all cross-deltas are zeros. The derivative assets have nonzero deltas with respect to multiple swap rates but not to other factors. These two assets do not require capital investment beforehand, but their future value depends on the changes in the swap rates.

The following constraints were used for the normal assets, already expressed in matrix form:

$$A_n = \begin{bmatrix} 1&0&0&0&0&0&0&0&0&0&0&0&0 \\ 0&1&0&0&0&0&0&0&0&0&0&0&0 \\ 0&0&1&0&0&0&0&0&0&0&0&0&0 \\ 0&0&0&1&0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&1&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&1&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&1&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&1&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&1&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0&1&0&0&0 \\ 0&0&0&0&0&0&0&0&0&0&1&0&0 \\ 0&0&0&0&0&0&0&0&0&0&0&1&0 \\ 0&0&0&0&0&0&0&0&0&0&0&0&1 \\ 1&1&1&1&1&1&1&1&1&1&1&1&1 \\ 1&1&1&1&1&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0&0&0&1&1 \end{bmatrix}, \underline{b}_n = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 83{,}400{,}000 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 417{,}000{,}000 \\ 0 \\ 0 \end{bmatrix}, \overline{b}_n = \begin{bmatrix} 41{,}700{,}000 \\ 41{,}700{,}000 \\ 41{,}700{,}000 \\ 41{,}700{,}000 \\ 41{,}700{,}000 \\ 417{,}000{,}000 \\ 417{,}000{,}000 \\ 417{,}000{,}000 \\ 125{,}100{,}000 \\ 41{,}700{,}000 \\ 62{,}550{,}000 \\ 41{,}700{,}000 \\ 41{,}700{,}000 \\ 417{,}000{,}000 \\ 41{,}700{,}000 \\ 41{,}700{,}000 \end{bmatrix}. \tag{25}$$

So, the constraints specify lower and upper limits for all positions, plus the total capital at hand (417 million EUR) as well as two constraints specifying minimum and maximum amounts than can be invested together in assets 1 – 5 and 12 – 13, respectively.

The positions in the derivatives are fixed (specifying that both of them denote a single specific derivatives contract), and they do not affect the position constraints of the other assets. The constraints of the entire portfolio can be expressed by augmenting the constraints specified earlier:

$$A_2 = \begin{bmatrix} A_n & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \underline{b}_2 = \begin{bmatrix} \underline{b}_n \\ 1 \\ 0 \end{bmatrix}, \overline{b}_2 = \begin{bmatrix} \overline{b}_n \\ 0 \\ 1 \end{bmatrix}, \tag{26}$$

where A_n, $\underline{b}_n$ and $\overline{b}_n$ are defined in (25).

3.4 Algorithm

The code was implemented using Python 3. To solve the optimization problems, packages `cvxopt` and `scipy.optimize` were used. A flow chart of the package is shown in Figure 1, and the rough structure of the code package is as follows:

1. Script file `main.ipynb`. It uses the function files to perform the entire Entropy Pooling procedure and prints the results and visualizations for the user.
2. Function file `views.py`, containing functions needed to upload factor scenario data and views specified by the users and to process this data into constraints used in entropy minimization.

3. Function file `entropy_minimizer.py`, containing the functions for performing the entropy minimization task.
4. Function file `markovitz_optimizer.py`, containing the functions for uploading asset delta matrix, computing the asset scenarios, and finally performing the Markowitz optimization and visualization of the results.

The code (not including the data) will be publicly available on GitHub. Please refer to the authors to get the Internet address.

4 Results

4.1 The posterior distributions

The posterior distributions are seen in Figure 3. The views are described in Table 4, and they are enforced one view at a time. In the Figure, we see that even a single view can produce highly polarized posterior weights, in which case the posterior relies on a very small data set. This means, that the further results computed with the posterior may be unreliable. In fact, the subsequent portfolio optimizations frequently failed when the *Rates down* view was used. The prior and posterior distributions of the view *Inflation up* are equal, as is seen in the Figure. This is because the prior distribution already satisfies the view condition.

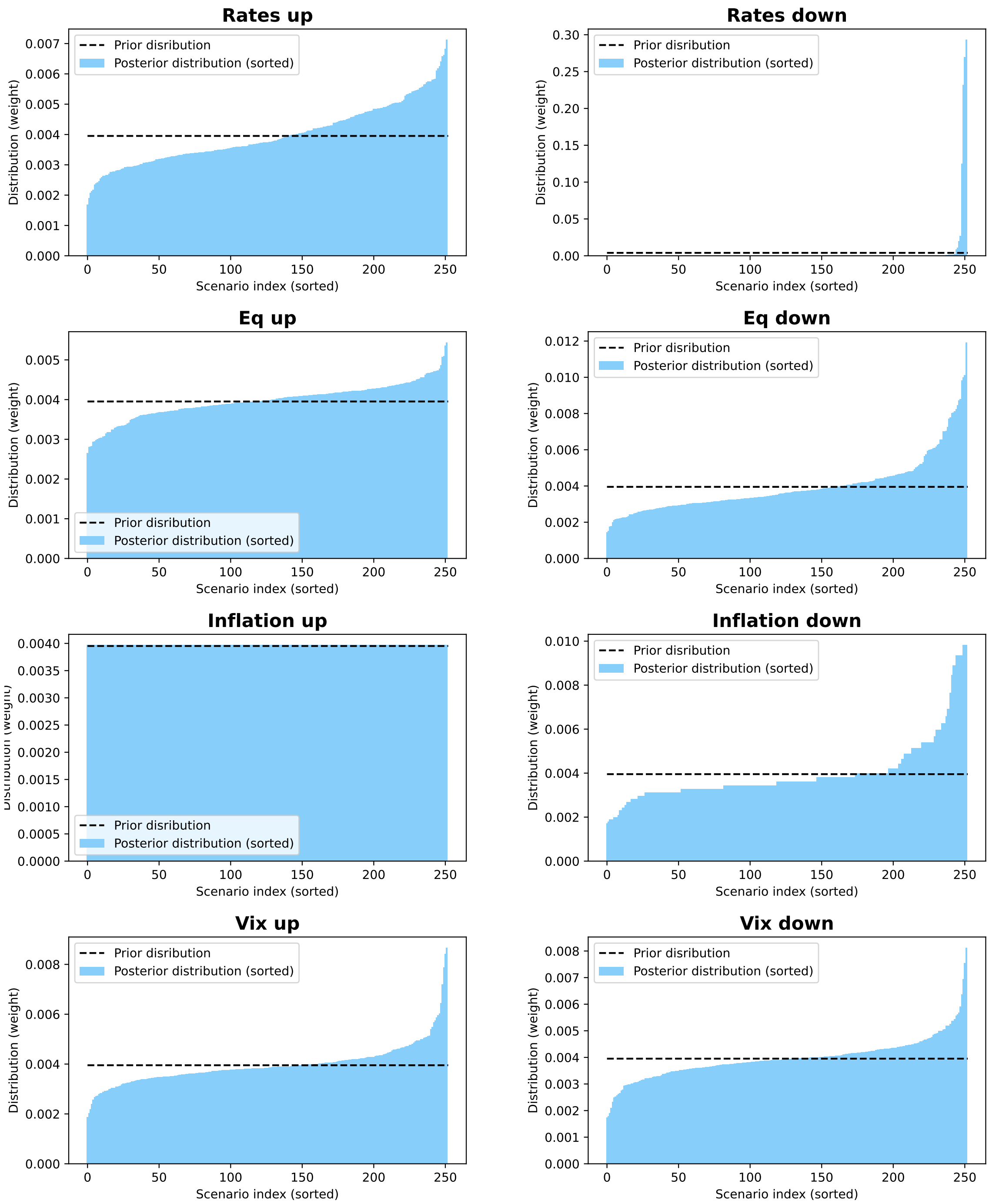


Figure 3: Sorted posterior distributions. The more the posterior deviates from the prior distribution, the fewer scenarios are 'active'. In the case *Rates down*, the posterior distribution is dominated by only a handful of scenarios.

4.2 Optimal portfolio allocation

4.2.1 Example portfolio 1

The expected returns and volatility of the optimal portfolio are shown in Figure 4. For this optimization task, the view *Equity up* was active. The optimal portfolio holdings are listed in Table 7. The expected return and volatility of the optimal portfolio are 0.36 % and 0.98 %, respectively. The magnitude of the numbers deviates from those often seen in textbook examples (tens of percents), but this is explained by the factor scenario data.

The properties of the optimal point become evident when it is compared with the convex combinations of the two bond assets (red curve). The swap thus enables the investor to achieve a level of volatility unattainable for portfolios consisting of just the two bond assets.

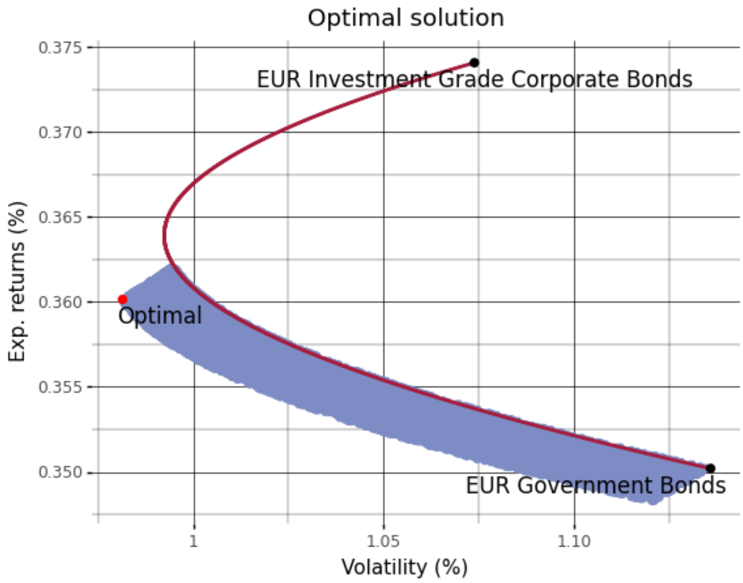


Figure 4: The volatility and expected returns of the optimal allocation (red point) plotted together with the individual assets excluding the swap (brown points). The blue point cloud represents all portfolios that satisfy the constraints (24), and the dark red curve represents the volatilities and returns of the two bond assets' convex combinations.

Table 7: The asset positions in the optimal portfolio of example 1.

Asset	Position (€)
EUR Govt. Bonds	500,000
EUR Inv. Grade Corp. Bonds	500,000
EUR 5v5 Swap	100,000

4.2.2 Example portfolio 2

Figure 5 shows the composition of the minimum volatility portfolios, for each view. The view *Rates down* was discarded, as the optimizer was not able to find any solutions with a positive expected return. The positions taken in hedge funds and listed private equity, change noticeably between different views. Otherwise, the portfolio compositions do not vary much, because of the constraints used in the Markowitz optimization. The volatilities and expected returns of the minimum variance portfolios of each view are shown in Table 8. Interestingly, the volatilities do not vary very much while there is a significant deviation between the expected returns.

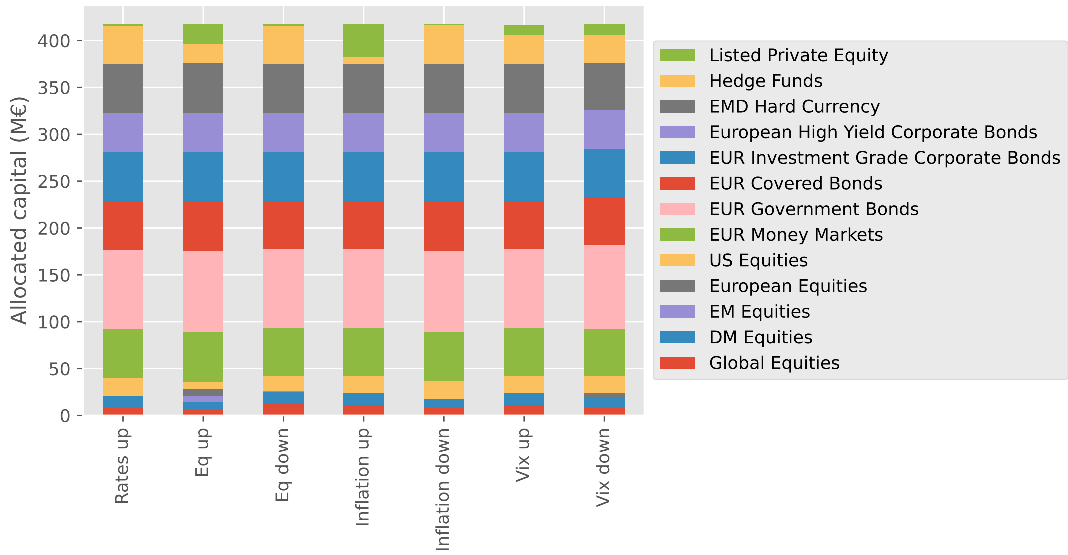


Figure 5: The asset compositions of the minimum variance portfolios, for each view. The view *Inflation up* yields the prior distribution, see Figure 3.

Table 8: The volatility and expected return of the minimum variance portfolios, when different views were enforced on the probability distribution. For reference, the values for the portfolio, optimized using the prior distribution, are also shown on the first row of the Table.

View	Volatility (€)	Expected return (€)
Prior	74,323	22,455
Rates up	65,764	10,351
Eq up	68,788	25,083
Eq down	72,441	10,380
Inflation up	74,323	22,455
Inflation down	64,052	23,742
Vix up	71,495	15,907
Vix down	65,687	25,515

The minimum variance portfolio, obtained using the view *Equity up*, is in Figure 6. Another portfolio was optimized with a pre-set minimum expected return of 2 million Euros, or around 0.48%.

The blue point cloud represents 5,000 randomly chosen feasible portfolios. We see that the constrained minimum variance portfolio does indeed have smaller volatility than any of the simulated sets. So, both optimal portfolios lie on the efficient frontier as supposed. It is noteworthy that one asset, Money Markets, has even smaller volatility. This is understandable, as the portfolio constraints prevent one from freely choosing any possible combination of the assets.

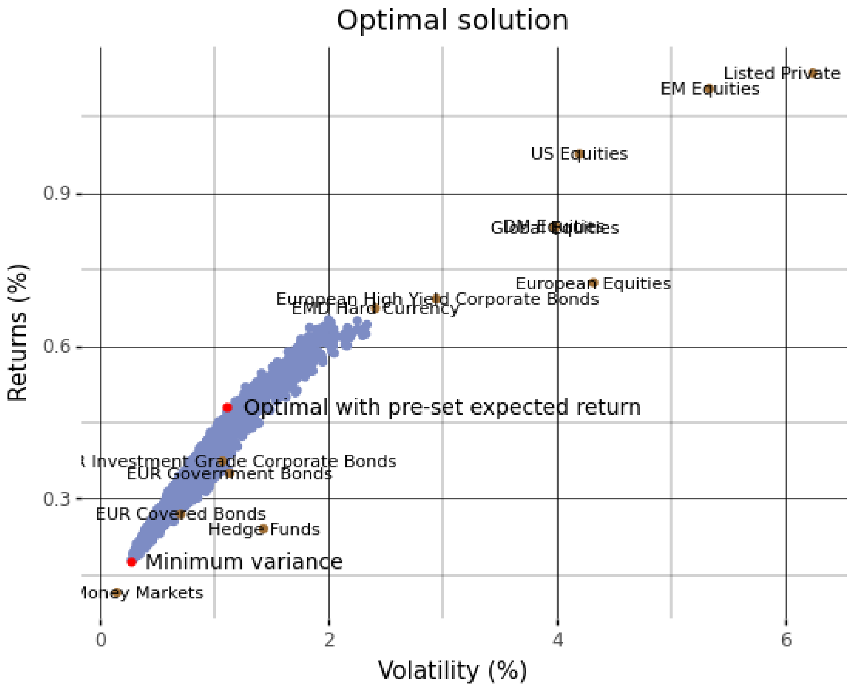


Figure 6: The volatility and expected return of two efficient portfolios (the red points). One is the minimum variance portfolio, and the other stands for an optimal portfolio with pre-set minimum expected return. The individual assets (not including the derivatives) are shown in brown. A randomly simulated set of 5,000 feasible portfolios is shown by the blue point cloud.

Figure 7 shows the portfolios on the efficient frontier in the case of the *Rates up* -view. The graph is quite stable across the frontier, with some fluctuation. Graphs for the remaining views are in Appendix C.

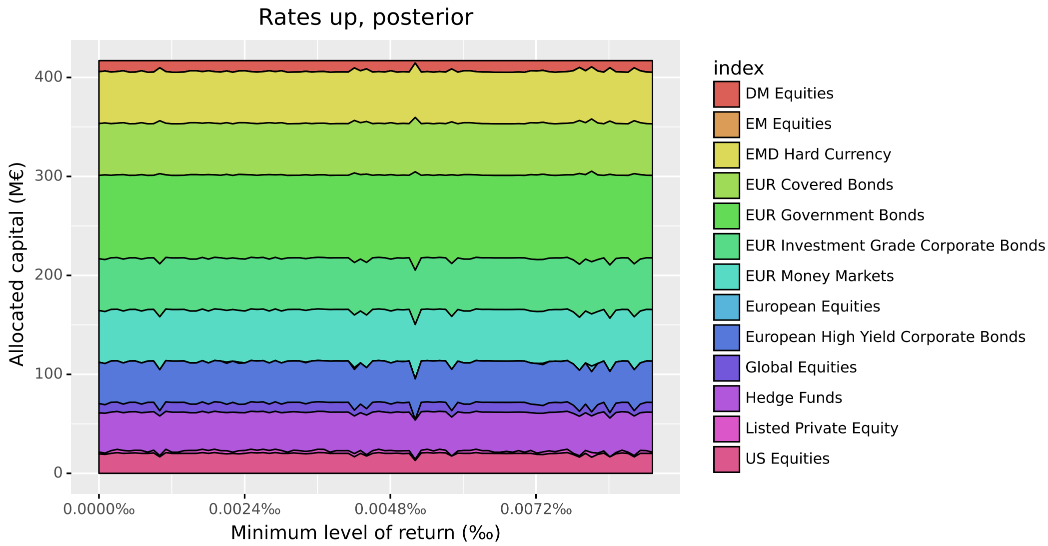


Figure 7: A stacked bar graph of portfolios at an efficient frontier, as a function of expected return.

5 Discussion

In this project, the Entropy Pooling approach was implemented for testing purposes for the client, Aktia Life Insurance. The implementation works and may provide precise and versatile asset handling. The example views and example portfolios produced different results providing a good overview.

Handling of a wide variety of different assets was successfully implemented in the program. However, an issue related to the factor scenarios, which was deemed to be out of the scope of this project, was the effect of the different units of the factor on the subsequent entropy minimization process. Namely, factors quoted in basis points, which might thus have much larger values than factors quoted in percentage returns or percentage points. It would be an interesting future extension to the implementation to study the effects of standardizing all the factor returns so that they have all zero mean and unit variance.

There is a very important caveat in the delta approach when constructing the asset-wise return scenarios. For example, if a portfolio consisting of several European assets is used, and all the asset returns are assumed to only depend on a single factor (*European Equities* in the input data used in this project), the asset-specific risk factors and differences in cross-asset correlations would be totally neglected. On the other hand, exhaustively modelling each asset's dependence on other factors likely makes the model overly complex and the user would consider it too tedious to update the delta matrix accurately. This is why one should conclude that this approach is best suited for portfolios consisting of asset indexes and other such instruments.

A known shortcoming of the Markowitz optimization theory is that the data (means, variances, and correlations of the asset returns) needs to be very precise. For most practical cases, the model is too sensitive. Instead, one could employ some more advanced assumptions on the behaviour of the assets, for example, a Bayesian approach to approximate the uncertainty of the parameters and, additionally, use higher moment knowledge [9]. We speculate, that together with the Entropy Pooling approach, one could achieve very good predictability.

One cannot emphasize too much the importance of accurate, reliable, and well-formatted input data in the results. Seemingly small errors, such as confusing bond yield units (percent versus basis point), or the notion of quoting bond yields instead of price returns, could hamper the results. Thus, it is crucial that the user understands these issues and makes sure that the data is exactly as required.

Even if the input data is correct and well gathered, the results may be unreliable, e.g., in the case of *Rates down*, see Figure 3. Information is always lost during the data processing, and the user must be aware of this fact. A numerical measure (or estimate) for 'active' points in the posterior distribution could be used to warn the user whenever the posterior distribution is dominated by just a few scenarios.

Relative entropy is a measure of dispersion, but not the only one used in mathematics. For example, Kolmogorov–Smirnov distance [10] could be used as well. An advantage of using entropy is the easy formulation of a dual optimization problem. Still, the choice of entropy seems rather arbitrary, and it might be hard to give any real-life justifications for choosing this measure over the others.

The conversion from views data to linear constraints turned out to be quite tedious and the results rely on approximations. The problem could be avoided by iterated optimization or with an optimization model that deals with specific non-linear constraints. Formulation of such a model is left for future research.

To summarize, the results obtained using the Entropy Pooling and Markowitz approaches should not be seen as the only viable approach when optimizing a portfolio. Instead, the methodology should be seen rather as a tool for exploring the effects of changing market scenarios and views on the Markowitz-optimal portfolio composition. The implemented approach is, however, an improvement over using more simpler models.

References

[1] Harry Markowitz. "Portfolio Selection". *The Journal of Finance* 7.1 (1952), pp. 77–91. ISSN: 00221082, 15406261. URL: `http://www.jstor.org/stable/2975974` (visited on 05/06/2022).

[2] Fischer Black and Robert Litterman. "Asset Allocation: Combining Investor Views With Market Equilibrium". *Goldman Sachs Fixed Income Research* 115 (1990), pp. 7–18.

[3] Harry Markowitz. *Portfolio Selection. Efficient Diversification of Investments*. Cowles Foundation for Research in Economics at Yale University, 1959. URL: `https://cowles.yale.edu/sites/default/files/files/pub/mon/m16-all.pdf`.

[4] Attilio Meucci. "Fully Flexible Views: Theory and Practice". *Fully Flexible Views: Theory and Practice, Risk* 21.10 (2008), pp. 97–102.

[5] James M Joyce. "Kullback-Leibler Divergence". *International Encyclopedia of Statistical Science*. Springer, 2011, pp. 720–722.

[6] Stephen G Nash. "A Survey of Truncated Newton Methods". *Journal of Computational and Applied Mathematics* 124.1-2 (2000), pp. 45–59.

[7] Ciyou Zhu et al. "Algorithm 778: L-BFGS-B: Fortran Subroutines for Large-Scale Bound-Constrained Optimization". *ACM Transactions on Mathematical Software (TOMS)* 23.4 (1997), pp. 550–560.

[8] William F Sharpe. "Mutual Fund Performance". *The Journal of Business* 39.1 (1966), pp. 119–138.

[9] Campbell R. Harvey et al. "Portfolio Selection With Higher Moments". *Quantitative Finance* 10.5 (2010), pp. 469–485. DOI: `10.1080/14697681003756877`. eprint: `https://doi.org/10.1080/14697681003756877`. URL: `https://doi.org/10.1080/14697681003756877`.

[10] Frank J Massey Jr. "The Kolmogorov-Smirnov Test for Goodness of Fit". *Journal of the American Statistical Association* 46.253 (1951), pp. 68–78.

A Views filling instructions

This section is copied from the code documentation.

In this code library, views can be given for mean, volatility and correlation. Views can be absolute or relative, and they can be equalities or inequalities. All numbers are given in annual units. Each view is filled as its own row, and the number of rows is only limited by Excel.

*Minor mathematical note: The **strict** inequality signs in the Excel act as ≤ and ≥ in the program.*

Mean values

Example: We want to set the mean of *Eurozone Core Inflation* to 1 % (annual). This is filled as follows,

* View on	* Risk factor 1	Risk factor 2 (applicable for corr)	* Operator	* Constant (alpha)	Multiplier (beta)	Risk factor 3	Risk factor 4 (applicable for corr)
Mean	Eurozone Core Inflation		=	0.01			

Note, that Excel will show 0.01 as 1 %, if percentage units are used. By leaving blank cells (or dash -), the program knows that we're dealing with an absolute view.

Example: *Eurozone Core Inflation* is at least 1 % (annual) greater than *US Core Inflation*. This is filled as follows,

* View on	* Risk factor 1	Risk factor 2 (applicable for corr)	* Operator	* Constant (alpha)	Multiplier (beta)	Risk factor 3	Risk factor 4 (applicable for corr)
Mean	Eurozone Core Inflation		>	0.01	1	US Core Inflation	

If a given multiplier is a dash (-), or the cell is left blank, the multiplier is interpreted as 1. The multiplier acts on the *Risk factor 3* (and *Risk factor 4* in the case of a correlation view). In mathematical terms,

$$\mu(\text{Risk factor 1}) - \beta\,\mu(\text{Risk factor 3}) = \alpha.$$

The equality sign can be changed to < or > if needed.

Volatility

Filling volatility views is analogous to filling mean values. The left-most column is changed from *Mean* to *Vol*. The volatilities are also filled with annual units.

Correlation

Only with correlation, are *Risk factor 2* and *Risk factor 4* used.

Example: The correlation between *Eurozone Core Inflation* and *US Core Inflation* is greater than 0.8. This is accomplished below,

* View on	* Risk factor 1	Risk factor 2 (applicable for corr)	* Operator	* Constant (alpha)	Multiplier (beta)	Risk factor 3	Risk factor 4 (applicable for corr)
Corr	Eurozone Core Inflation	US Core Inflation	>	0.8			

Possible errors

The following cases must be **satisfied**

- The view rows should not lead to contradictions (e.g., rows (*Eurozone Core Inflation*) = 1 %, and (*Eurozone Core Inflation*) = 2 % would lead to a contradiction)
- Volatility is always positive, or zero.
- Correlation only gets values from −1 to 1.
- With correlation view, (*Risk factor 1*) ≠ (*Risk factor 2*) and (*Risk factor 3*) ≠ (*Risk factor 4*).
- With any relative view, (*Risk factor 1*) ≠ (*Risk factor 3*) and with relative correlation, (*Risk factor 2*) ≠ (*Risk factor 4*).

B Variance approximation

Consider the following equation,

$$\sqrt{s_x} - \sqrt{s_y} = \sigma_{xy} \implies s_x - 2\sqrt{s_x s_y} + s_y = \sigma_{xy}^2,$$

where s_x and s_y are variances for assets x and y, respectively. The term σ_{xy} is a user defined constant. The variances, s_x and s_y, depend on the posterior distribution, p_s, $s \in S$. We'd like to approximate the right-hand-side equation linearly with respect to p_s, $s \in S$. We shall focus on term $\sqrt{s_x s_y}$. First, fix the weighted mean values $\bar{x}$ and $\bar{y}$, and define,

$$s_x = \sum_{s \in S} p_s (x_s - \bar{x})^2 = \sum_{s \in S} p_s a_s,$$

$$s_y = \sum_{s \in S} p_s (y_s - \bar{y})^2 = \sum_{s \in S} p_s b_s.$$

We evaluate partial derivatives with respect to p_s, $s \in S$,

$$\frac{\partial}{\partial p_s} \sqrt{s_x s_y} = \frac{1}{2\sqrt{s_x s_y}} \frac{\partial}{\partial p_s}(s_x s_y) = \frac{1}{2\sqrt{s_x s_y}} \left(s_x \frac{\partial s_y}{\partial p_s} + s_y \frac{\partial s_x}{\partial p_s} \right) = \frac{1}{2\sqrt{s_x s_y}} (s_x b_s + s_y a_s).$$

Let, σ_x and σ_y, be the standard deviations with prior data. Now we write a Taylor approximation, where p'_s, $s \in S$, is the prior distribution,

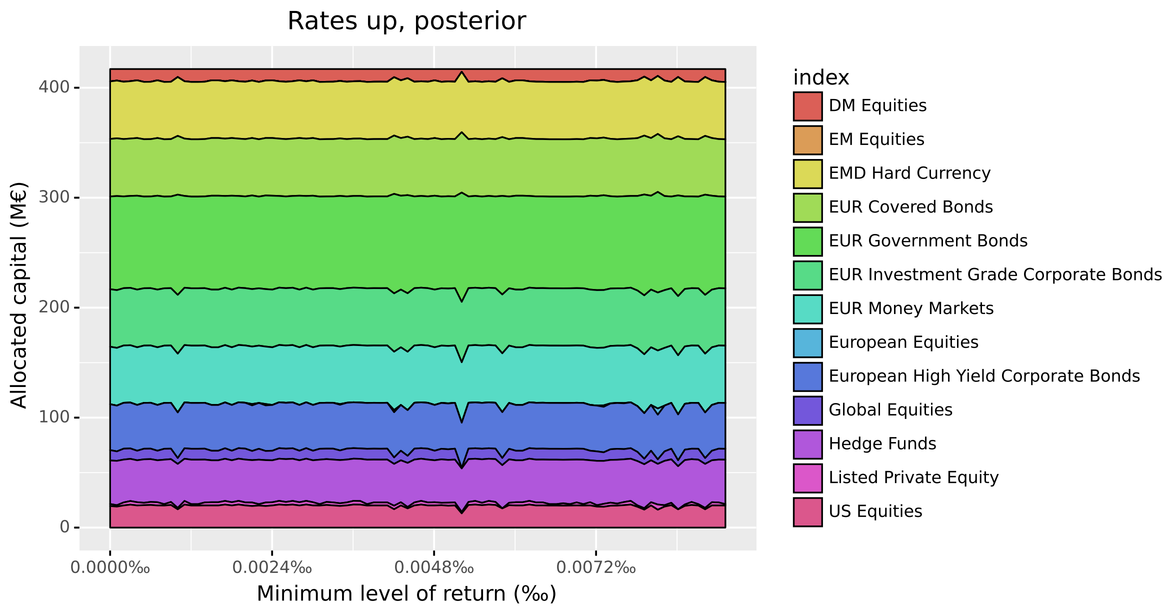
$$\begin{aligned}
\sqrt{s_x s_y} &\approx \left(\sqrt{s_x s_y}\right)\Big|_{s_x \leftarrow \sigma_x^2 \ ; \ s_y \leftarrow \sigma_y^2} + \sum_{s \in S} (p_s - p'_s) \left(\frac{\partial}{\partial p_s} \sqrt{s_x s_y} \right) \Big|_{s_x \leftarrow \sigma_x^2 \ ; \ s_y \leftarrow \sigma_y^2} \\
&= \sigma_x \sigma_y + \frac{1}{2\,\sigma_x \sigma_y} \sum_{s \in S} (p_s - p'_s)(a_s \sigma_y^2 + b_s \sigma_x^2) \\
&= \sigma_x \sigma_y + \frac{1}{2\,\sigma_x \sigma_y} \sum_{s \in S} p_s (a_s \sigma_y^2 + b_s \sigma_x^2) - \frac{1}{2\,\sigma_x \sigma_y} (\sigma_x^2 \sigma_y^2 + \sigma_y^2 \sigma_x^2) \\
&= \sum_{s \in S} p_s \frac{a_s \sigma_y^2 + b_s \sigma_x^2}{2\,\sigma_x \sigma_y}.
\end{aligned}$$

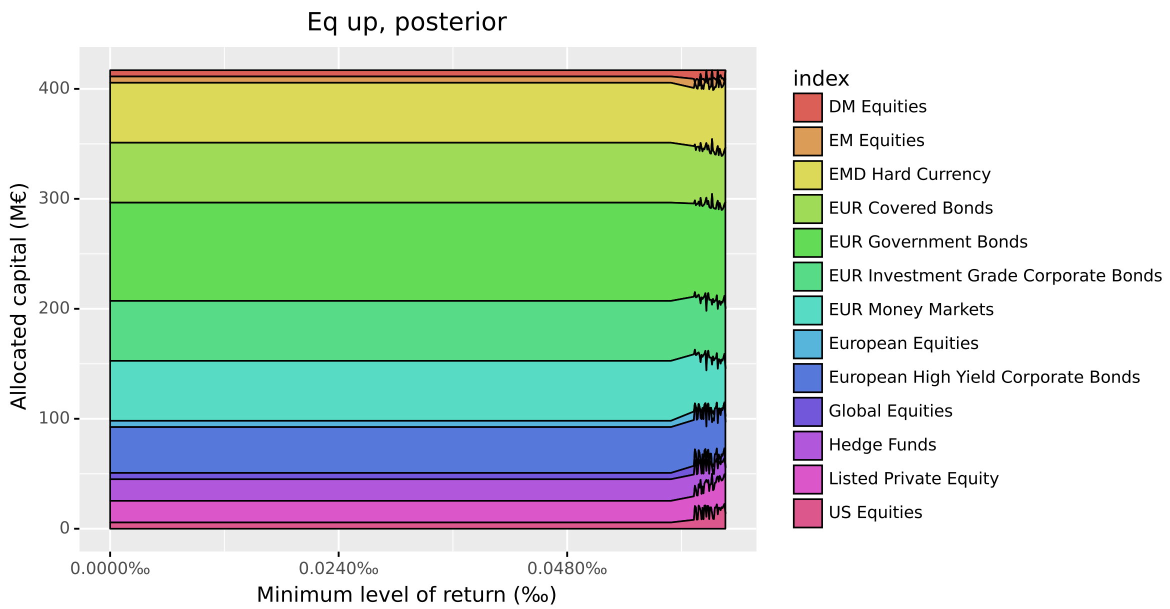
Finally, we have an approximation for σ_{xy}^2,

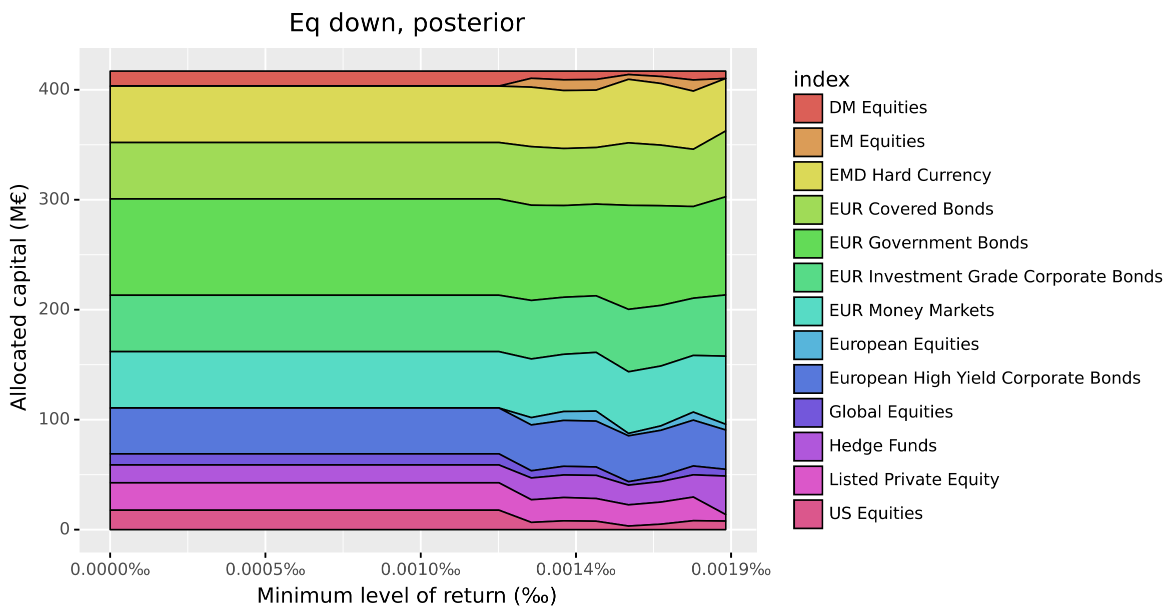
$$\sigma_{xy}^2 \approx \sum_{s \in S} p_s \left(a_s + b_s - \frac{a_s \sigma_y^2 + b_s \sigma_x^2}{\sigma_x \sigma_y} \right).$$

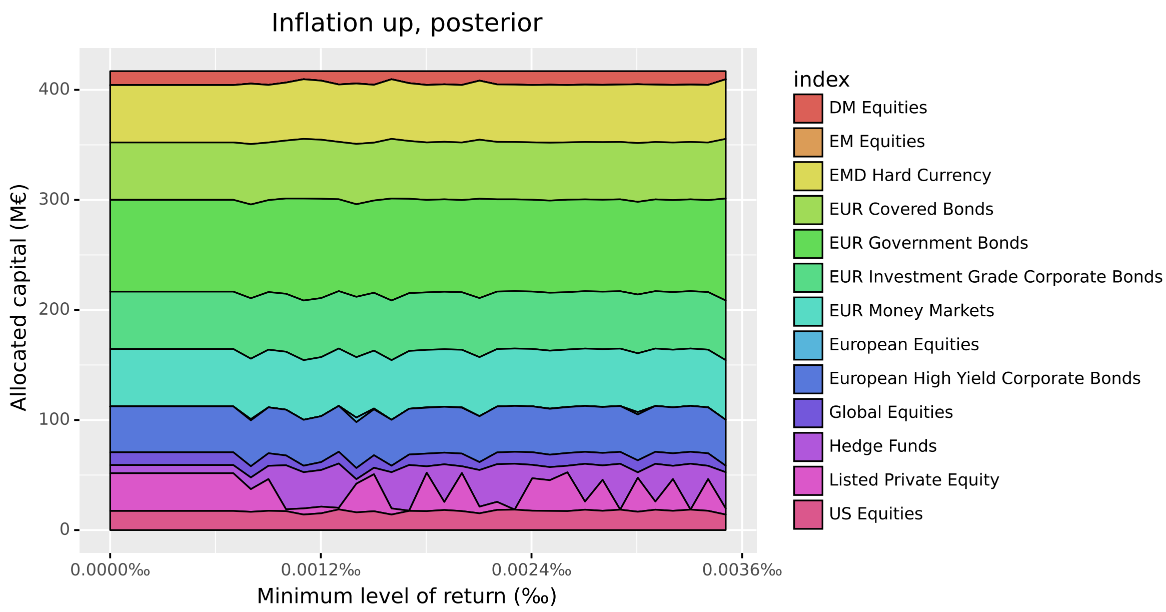
C Stacked portfolio graphs

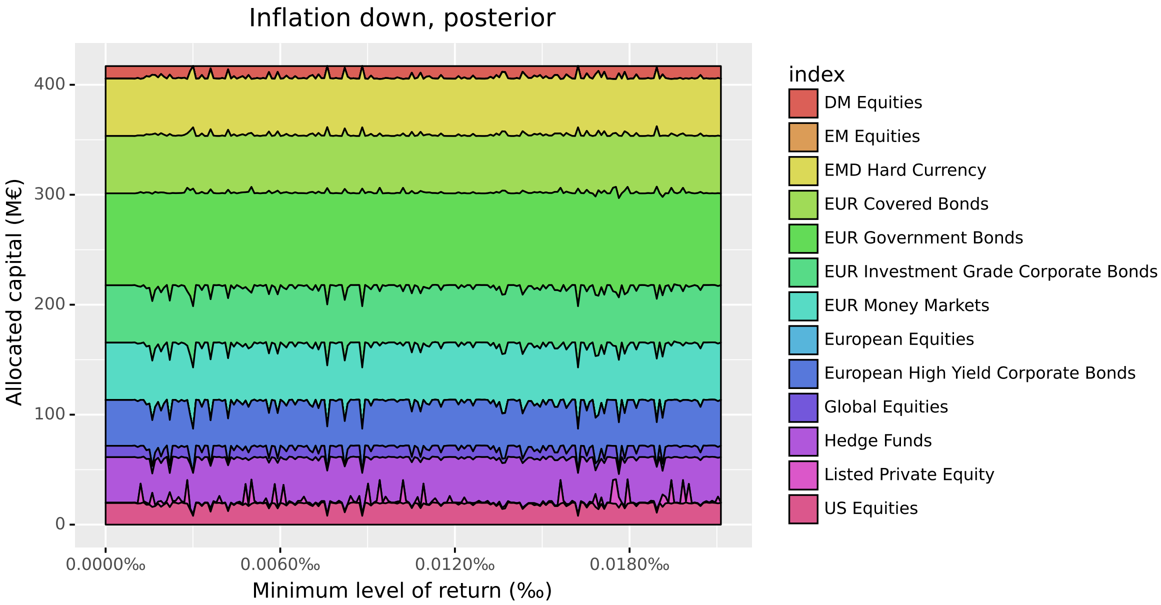
The portfolios at the efficient frontier, as a function of expected return. Here, we use the Markowitz constraints described in Chapter 3.3.2. Note that the case of the *Rates down* -view is discarded because the optimization did not yield feasible results with a positive expected return. Note, that the number of sampled portfolios varies. The graphs include the whole range of feasible and positive expected returns.



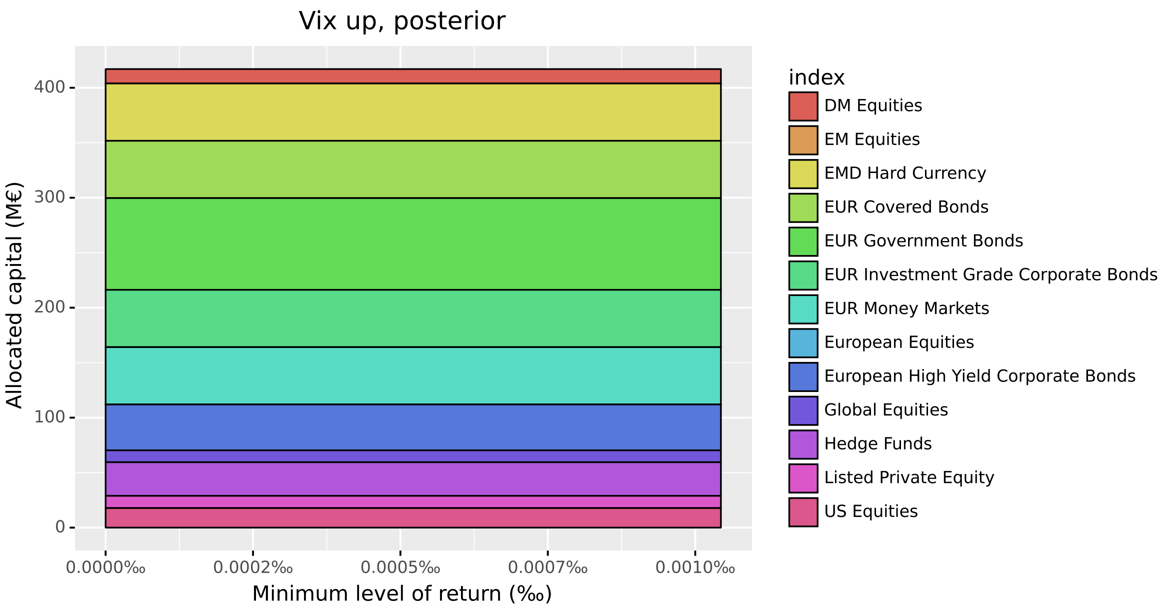



Eq down, posterior
Allocated capital (M€)
400
300
200
100
0
0.0000‰
0.0005‰
0.0010‰
0.0014‰
0.0019‰
Minimum level of return (‰)
index
DM Equities
EM Equities
EMD Hard Currency
EUR Covered Bonds
EUR Government Bonds
EUR Investment Grade Corporate Bonds
EUR Money Markets
European Equities
European High Yield Corporate Bonds
Global Equities
Hedge Funds
Listed Private Equity
US Equities

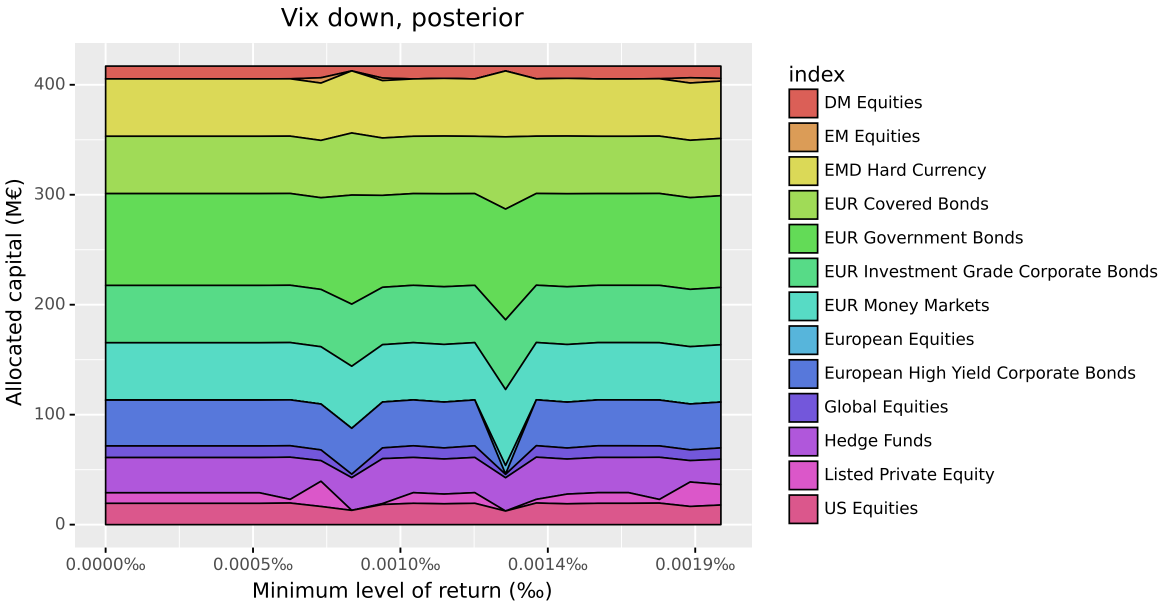

Inflation up, posterior
Allocated capital (M€)
400
300
200
100
0
0.0000‰
0.0012‰
0.0024‰
0.0036‰
Minimum level of return (‰)
index
DM Equities
EM Equities
EMD Hard Currency
EUR Covered Bonds
EUR Government Bonds
EUR Investment Grade Corporate Bonds
EUR Money Markets
European Equities
European High Yield Corporate Bonds
Global Equities
Hedge Funds
Listed Private Equity
US Equities



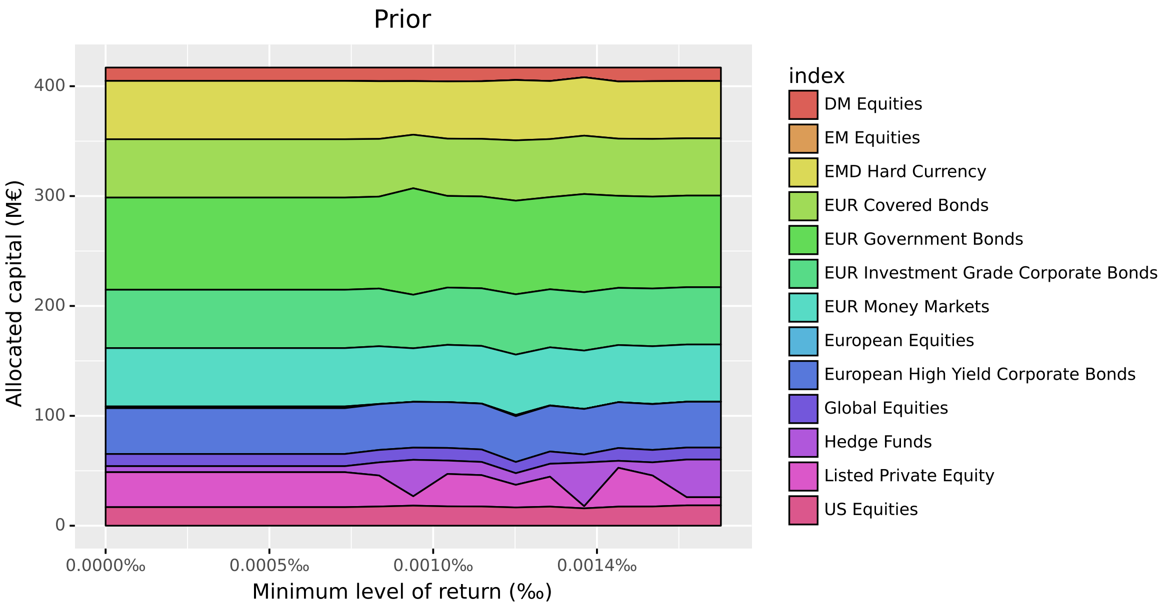
Inflation down, posterior
Allocated capital (M€)
0
100
200
300
400
0.0000‰
0.0060‰
0.0120‰
0.0180‰
Minimum level of return (‰)
index
DM Equities
EM Equities
EMD Hard Currency
EUR Covered Bonds
EUR Government Bonds
EUR Investment Grade Corporate Bonds
EUR Money Markets
European Equities
European High Yield Corporate Bonds
Global Equities
Hedge Funds
Listed Private Equity
US Equities



Vix up, posterior
Allocated capital (M€)
0
100
200
300
400
0.0000‰
0.0002‰
0.0005‰
0.0007‰
0.0010‰
Minimum level of return (‰)
index
DM Equities
EM Equities
EMD Hard Currency
EUR Covered Bonds
EUR Government Bonds
EUR Investment Grade Corporate Bonds
EUR Money Markets
European Equities
European High Yield Corporate Bonds
Global Equities
Hedge Funds
Listed Private Equity
US Equities



Vix down, posterior
Allocated capital (M€)
0
100
200
300
400
0.0000‰
0.0005‰
0.0010‰
0.0014‰
0.0019‰
Minimum level of return (‰)
index
DM Equities
EM Equities
EMD Hard Currency
EUR Covered Bonds
EUR Government Bonds
EUR Investment Grade Corporate Bonds
EUR Money Markets
European Equities
European High Yield Corporate Bonds
Global Equities
Hedge Funds
Listed Private Equity
US Equities



Prior
Allocated capital (M€)
0
100
200
300
400
0.0000‰
0.0005‰
0.0010‰
0.0014‰
Minimum level of return (‰)
index
DM Equities
EM Equities
EMD Hard Currency
EUR Covered Bonds
EUR Government Bonds
EUR Investment Grade Corporate Bonds
EUR Money Markets
European Equities
European High Yield Corporate Bonds
Global Equities
Hedge Funds
Listed Private Equity
US Equities

Self-assessment

1) How closely did the actual implementation of the project follow the initial project plan? Were there any major departures and, if so, what?

The very implementation of the Entropy Pooling approach, including views processing, entropy minimization, and Markowitz portfolio optimization was realized without deviations from the original objectives. The work followed the papers given as source material the way it was supposed at the beginning.

As the project progressed, the importance of correctly handling the input data gained much more significance than it was initially thought. Not only the mere format of the data was considered (e.g. the file format and order of columns and rows), but we also understood the many possibilities of error related to misunderstandings. As discussed in the text, confusing percentage returns, percentage points, percentage yields, and basis points is one of such issues. Another issue is that all the asset scenarios must have percentage returns as units, so special attention is required when quoting the deltas of bonds and other assets for which percentage returns of prices are not usually considered.

The suggestions of Aktia concerning the usability and scope of the project in terms of asset classes also caused some extensions to the original project scope. Namely, the Markowitz optimization had to be modified to include asset classes, such as swaps, that were originally not considered.

2) In what regard was the project successful?

The project clearly expanded the understanding of the team members in the field of portfolio optimization of financial assets. Especially the shortcomings of the aforementioned theories have become clear and shed light on the complexity of state-of-the-art portfolio theory.

In terms of working as a team, everybody participated actively in the work and the communication was frequent. Any problems were addressed swiftly. The team members got experience in working on a larger code-based project with multiple people working on different parts. Additional reflection on this can also be found in the next part.

The client is also pleased with our co-operation and results. The communication between the client and the team was effortless, but not very frequent due to clear objectives.

3) In what regard was it less so?

A team member decided not to continue the course with the team at an early stage, but the team managed to finish the project with the three remaining team members. Although this put more pressure on the rest of the team, it also lead to the remaining members learning more as the project was divided into larger parts between the team members.

Due to the nature of the project, its implementation, and its extent, the final project is mostly a proof-of-concept and may contain very serious flaws if implemented in a business environment. This is due to not having access to enough data to test a wide enough variety of different scenarios, inputs, and outcomes. Additionally, some additional features of the code were implemented at a late stage and, thus, there was not enough time to properly test the newest features.

4) What could have been done better, in hindsight? (you may analyze this question from the roles of the project team, the client, and the teacher(s))

This project was a valuable lesson on working in a team, including an external stakeholder. It became clear that the largest issues were not connected to the mathematical aspects, but to the program implementation. Namely, when the function tasks were divided between team members, it was not specified in which format everybody's functions take inputs and give outputs. This was a frequent source of confusion and frustration. The solution here could have been to together decide on the input and output format of each of the main functions and then go on to let each member decide how their internal part looks by themselves.

Communication regarding the course schedule could be clearer. Approximate deadlines were given at an early stage beforehand, but exact deadlines for both presentation and reports were only given in under a week before the deadlines themselves. This resulted in hasty reports and presentations. However, we did not miss any deadline, and we're pleased with the results.

The team should have scheduled a few more meetings with the client. During the first meeting, when the team had completed a somewhat working prototype, the client was able to specify requests and features. These lead to some confusion in the team that might have been easier to clear up with an in-person meeting with the client.

Some of the team members possessed domain knowledge, that is knowledge of finance and banking, but not all. This made understanding requests and context more challenging for those members not acquainted with the subject. Where the responsibility lies for acquiring the aforementioned knowledge is not clear, be it on the teacher, client, or the team itself. However, it would have been beneficial to be introduced to some of the more uncommon terms which were quite important for the project (from the point of view of someone who has not studied or worked in the domain).