Portfolio Construction and Risk Management

Anton Vorobets

Latest sneak peek: https://ssrn.com/abstract=4807200

Crowdfunding page: https://igg.me/at/pcrm-book

Latest updates: https://antonvorobets.substack.com

Preface

This book is written based on the need to bridge quantitative analysis and investment management in practice, particularly focusing on portfolio construction and risk management. The approach can be characterized as being quantamental, i.e., a mix of quantitative analysis and human and machine interaction. It is based on the philosophy that some things are really hard for machines and statistical models to do, because these usually require a significant amount of stationary data to work well, while humans are better capable of quickly developing an understanding based on little or imperfect information. Hence, most people likely have the highest probability of being successful using a quantamental approach to investment management, but all the methods can be used for fully systematic investing.

The book attempts to keep the mathematics practically rigorous while referring to proofs of very technical results instead of replicating them. It does not attempt to theorize or quantify human behavior like some finance and economics academics do with utility theory. Conventional utility theory has poor empirical support and limits the analysis to methods that have very little to do with real-world market behavior. In relation to investor preferences, our hypothesis is that investors characterize investment risk as large losses, not squared deviations from the mean. It is very hard to reject this hypothesis in practice. We will in particular focus on the Conditional Value-at-Risk (CVaR) investment risk measure, because it has many nice properties and is easy to interpret.

The goal is to keep the book concise and to the point. It does not spend time on introducing general mathematical concepts, because these are readily available in other great books. Hence, it is a prerequisite that you understand linear algebra, calculus including convex constrained optimization with Lagrange multipliers, probability theory, multivariate time series, and machine learning/artificial intelligence/econometrics/statistics. Finally, we will use Python and in particular the fortitudo.tech[1] package for the code examples.

This book is cohesive in the sense that it gives you a complete investment risk and analysis framework. The approach is scientific as it tries to address many of the complex nuances of real-world investment markets. It is based on my experiences managing multi-asset portfolios for institutional clients and my regular dialogues with institutional asset managers about the problems they experience in practice. The book will naturally draw on my articles[2], while presenting the framework in a more coherent way to make it perfectly clear how it should be used from start to finish including its subtle nuances.

[1] Available at https://github.com/fortitudo-tech/fortitudo.tech
[2] Available at https://ssrn.com/author=2738420

I omit aspects related to signal extraction and alpha generation. You should rather see the framework from this book as the general principles for a good investment calculator that allows you to get the most out of your investment risk budget and, hence, generate portfolio construction alpha. Alpha related to security selection and market timing is so elusive that I would trade it myself and not tell anyone how I did it.

This book is very different from the current mainstream academic finance and economics books that continue to spend time on theories and methods like utility theory, CAPM, mean-variance, and Black-Litterman. I think these methods do more harm than good in practice due to their highly unrealistic market assumptions. However, they are still being taught mainly due to academic and commercial vested interests. We will occasionally use mean-variance to build intuition in the idealized case, but it is never recommended to actually use mean-variance to manage portfolios in practice. The elliptical distribution assumption is too oversimplifying to represent the dynamics of real-world markets well. All methods presented in this book operate directly on fully general Monte Carlo distributions with associated joint probability vectors.

The book is written for investment practitioners who employ a scientific approach to investment management that potentially combines data with human discretion. By scientific, I mean constantly testing the theories and methods against real-world market behavior. Readers are generally encouraged to critically examine the content of this book and suggest improvements that will make the framework even better for skillfully navigating investment markets in practice. Hypothetical issues or introduction of constraints due to theories with poor empirical support will not be considered.

The book and supporting material may eventually be provided free of charge online, while crowdfunding makes it possible to write it. During writing, all campaign contributors beyond a threshold amount will have access to a private GitHub repository where the latest version of this book will be available in addition to the accompanying Python code. I will also answer questions in the forum of this repository. Top 10 contributors by the time this book is finished will have their name written in this preface (if they wish) in addition to getting three months access to an institutional-grade implementation of the investment framework described in this book. Hence, backing this project gives you access to a community that is actively learning the content simultaneously with you. It is a great opportunity to master the theory and methods before most others.

The book will occasionally use proprietary software for some of the case studies and examples. I decided to do this in order to show readers some of the more advanced analysis that they can perform using the methods from this book, although the accompanied code cannot be provided. In most situations, the code will be available, and it is an integral part of studying the content of this book, while readers should not expect the accompanying code to be of production quality. The accompanying code is given without any warranty. The author cannot be made liable for any potential damages stemming from using the code.

To get access to the book and the accompanying Python code, you can support the crowdfunding campaign at: **https://igg.me/at/pcrm-book**

It doesn't matter how beautiful your theory is. If it disagrees with experiment, it's wrong.

- Richard Feynman

Acknowledgments

The creation of this book is made possible through crowdfunding contributions at:

https://igg.me/at/pcrm-book

Contributors have often shared their feedback on the content, making it better and easier to understand for all readers. All contributions are highly appreciated, while some people have contributed significant amounts and allowed their names to be shared in this section. While the book is being written, the below lists will be updated and expanded. When the book is finished sometime in Q1 2025, the final top 10 list will be revealed. Until then, the top 10 contributors section will include the necessary contribution amount to enter the top 10 as of the date listed in the section. Top 10 contributors will get access to exploring an institutional-grade implementation of the framework and really test out their newly acquired knowledge in practice. Potential ties among top 10 contributors are settled on a first come first served basis, giving early contributors an advantage.

Top 10 Contributors

As of January 10, 2025, your contribution must be more than 100 euro to enter the current top 10.

Platinum Contributors

Charlotte Hansen, Hitesh Sundeha.

Gold Contributors

Silver Contributors

Bronze Contributors

Matteo Nobile, Roger McIntosh, Mark Brezina, Veliko Dinkov Donchev, Irina Johansen, Pedro Jorge Feijoo Videira e Castro, Hans-Peter Schrei, Zarko Stefanovski, Radu Briciu, and four anonymous contributors.

Contents

Chapter 1

Introduction and Overview

1.1 Book and Investment Framework Overview

The investment framework in this book is centered around a Monte Carlo simulation given by the matrix $R \in \mathbb{R}^{S \times I}$ and associated joint scenario probability vector $p = (p_1, p_2, \ldots, p_S)^T \in \mathbb{R}^S$ with $\sum_{s=1}^{S} p_s = 1$ and $p_s \in (0, 1]$. This market representation allows us to work with fully general joint investment distributions. If we want to introduce some (subjective) market views or stress-tests, we generally do it by adjusting the prior probability vector p into the posterior probability vector $q \in \mathbb{R}^S$.

The columns of R represent samples from the marginal distribution of price, return, or factor i, $i = 1, 2, \ldots, I$. The rows of R represent samples from the joint distribution of the prices, returns, and factors. Hence, the core elements of the investment framework are

$$R = \begin{pmatrix} R_{1,1} & R_{1,2} & \cdots & R_{1,I} \\ R_{2,1} & R_{2,2} & \cdots & R_{2,I} \\ \vdots & \vdots & \ddots & \vdots \\ R_{S,1} & R_{S,2} & \cdots & R_{S,I} \end{pmatrix}, \quad p = \begin{pmatrix} p_1 \\ p_2 \\ \vdots \\ p_S \end{pmatrix}, \quad \text{and} \quad q = \begin{pmatrix} q_1 \\ q_2 \\ \vdots \\ q_S \end{pmatrix}. \tag{1.1.1}$$

Our first task is to generate realistic prior simulations of the market R based on observed historical market data $D \in \mathbb{R}^{T \times N}$. Realistic simulations are characterized by their ability to capture the stylized market facts presented in Chapter 2 and ideally additional subtle nuances of real-world investment markets. Another important point is that analysis and optimization methods must be able to handle the fully general simulations R and associated probability vectors p and q. Mean-variance analysis clearly fails in this regard, because it reduces the market to an elliptical distribution and only allows for linear and cross-sectionally constant dependencies through the covariance matrix.

Note that the market representation in (1.1.1) and most of the analysis in this book are one-period in nature, while we do not make any assumptions about living in a one-period world. Hence, there is an implicit horizon in the matrix R. Sometimes, we will be explicit and write R_h, $h = 1, 2, \ldots, H$, as in Chapter 3 about market simulation, while maintaining the more concise notation throughout most of the book. For dynamic investment strategies, it is important to simulate entire paths, while their cumulative returns can still be analyzed in a meaningful way for some horizon $h \in \{1, 2, \ldots, H\}$.

When generating Monte Carlo simulations of the market, the focus will be on generating simulations of factors and then translating their simulations into instrument price and return simulations. We will distinguish between two kind of factors: risk factors and market factors. A risk factor is defined as something that goes directly into the pricing function of an instrument, for example, US government bond zero-coupon interest rates are risk factors for US government bonds. A market factor is something that might have a statistical effect on the price or return of an instrument, for example, Purchasing Managers' Index (PMI) numbers, but does not directly enter into the pricing function of an instrument. Risk factors are explored much more carefully in Chapter 4 about instrument pricing.

Once we have good simulations of prices, returns, and factors, we probably want to incorporate adjustments based on information that might not be available in the historical data. We call this (subjective) market views. We might also want to examine how our simulations behave in adverse market scenarios, which we call stress-testing. No matter which of the two cases it is, the fundamental method we use is called Entropy Pooling (EP), introduced by Meucci (2008a). EP minimizes the relative entropy (Kullback–Leibler divergence) between the prior probability vector p and the posterior probability vector q subject to linear constraints on the scenario probabilities, i.e.,

$$q = \underset{x}{\operatorname{argmin}} \left\{ x^T (\ln x - \ln p) \right\}$$

subject to

$$Gx \leq h \quad \text{and} \quad Ax = b.$$

Note that we use $\ln x$ to mean the logarithm of each element of x, and that $\leq$ and $=$ are used to denote element-wise (in)equalities. Entropy Pooling can be used in many sophisticated ways, see for example Vorobets (2021) and Vorobets (2023), that we will carefully explore in Chapter 5.

After implementing the market views and stress-testing, we likely want to optimize our portfolios subject to various investment constraints and include portfolio optimization parameter uncertainty, see Kristensen and Vorobets (2024) and Vorobets (2024). Portfolio optimization is the topic of Chapter 6, where we will focus on CVaR optimization with risk targets and risk budgets in addition to the ability to seamlessly handle derivatives instruments by separating relative market values $v \in \mathbb{R}^I$ from relative exposures $e \in \mathbb{R}^I$, see Vorobets (2022a) and Vorobets (2022b).

Chapter 7 is about general risk and return analysis, with a particular focus on the nuances of tail risk hedging and management. The general idea is that the methods in the first six chapters will allow us to build well-diversified portfolios, which we can additionally equip with tail risk hedges if we wish. Outright tail risk hedging often comes at a significant negative (volatility) risk premium. Hence, tail risk hedges should ideally be designed with the objective of giving a positive return when key diversification assumptions fail, but outright hedging can be considered on a tactical basis.

Chapter 8 contains a summary of the investment risk and analysis framework introduced in this book in addition to a comparison with old methods such as mean-variance and Black-Litterman (BL). In Chapter 8, we will see that in the best case the old framework is a simple subset of the new, while BL contains so many questionable aspects that it does not even produce a correct updating of the CAPM prior to a posterior distribution when returns are normally distributed. For this reason, it is never recommended to use the old framework in practice, and especially not the BL model.

The rest of this chapter presents core principles for reasoning about market states, structural breaks, and time-conditioning in addition to the essence of investment and risk management. These principles will be important to keep in mind throughout the book and should apply to the majority of investment markets and strategies. Very niche investment markets or strategies might have characteristics that are so unique and different from the majority that they fall outside the scope of a general portfolio construction and risk management book like this one.

1.2 Market States, Structural Breaks, and Time-Conditioning

We start this section with the dreaded coin flipping analysis. Not because we want to study the probability theory behind coin flips and the associated Bernoulli and binomial distributions. Also not because coin flipping is a good representation of investment markets, but because it allows us to introduce the important concepts of market states and structural breaks in a way that is easy to understand. These concepts will be important for thinking about real-world investment and risk management, where the sheer complexity might overshadow the essence of the points made in this section.

As an investment professional, you might have experienced a situation where someone you know asked you for investment advice along the lines of "what about company X? Should I buy it now, will it go up?". Many people seem to think that what investment managers do is to predict and time the realizations of investment markets. Some investment managers trading in very special markets or on very special information might, but the vast majority do not, simply because it is impossible.

Many people find the answer about market realizations not being predictable unsatisfactory, while they fully understand that no one can predict the realization of the next coin flip or draw of lottery numbers. In fact, if someone claimed to be able to do that, most people would immediately distrust that claim. It is clear to most that the best we can do when it comes to coin flips and lotteries is to make probabilistic predictions. Yet the additional complexity of investment markets fools many people into believing that they have discovered a pattern for the next period's realization. As an inexperienced investor, you might even think that you have this ability yourself, but with time you will realize that consistently calling the realization of investment markets is not possible.

So, is careful investment analysis and risk management a complete waste of time? Not quite. It is just important to understand that we can only add value in a probabilistic sense, which we will carefully explore in the next section. In this section, we will continue to focus on coin flips to introduce the important concepts of market states, structural breaks, and time-conditioning.

Without going into too much mathematical formality, let us consider a coin C that can come out heads with probability $p \in [0, 1]$ and tails with probability $1 - p$. We can flip this coin over a discrete time-period $t = 1, 2, \ldots, T$ and denote its realization at each step $c_t \in \{\text{heads}, \text{tails}\}$. Let us assume that you have \$100 that you can invest in various strategies involving the coin, giving you the opportunity of increasing the value of your initial \$100. If it is a "fair" coin with $p = 0.5$, the expected value of engaging in this activity is \$0. You might still want to participate in the coin flipping just for the thrill of it and design strategies that fix the probability of losing all your money during the time-period to some sufficiently low number.

Let us now imagine that the coin is biased such that $p = 0.9$, making it obvious that you can expect to earn money by systematically betting on heads. If you are not careful, you can still lose all your money as there is no guarantee that the coin will come out heads. Hence, it is still impossible for you to call the realization of the coin c_t at any time t. How you decide to invest in this coin will to a large extent be a function of your risk willingness. The highest expected return will be achieved by betting all your money on heads at each point in time $t = 1, 2, \ldots, T$, while it also gives you the highest probability of losing all your money. A reasonable suggestion would be to find a strategy where the trade-off between the expected return and the probability of losing all your money is the best. While we are not going to spend time on that in this book, the interested reader can study this further.

Continuing with the coin flipping, let us imagine a situation where $p = 0.1$. In this case, betting on tails will obviously be associated with a positive expected return. However, let us say that you are constrained to only betting on heads, i.e., you are "long-only" heads, while you can adjust the size of your bets before each coin flip at time t. If the coin is in a constant state with $p = 0.1$, while you are only allowed to bet on heads, it only make sense for you to participate if you are a real thrill seeker. However, if the coin switches between $p \in \{0.1, 0.5, 0.9\}$, you still have the opportunity of generating a positive expected return if you are able to infer the value of p and size your investments accordingly.

To formalize the above a bit more, let us introduce a state variable $z_t \in \{z_{t,1}, z_{t,2}, z_{t,3}\}$, $t = 0, 1, \ldots, T$, and imagine that states changes according to the following transition probability matrix

$$\mathcal{T} = \begin{pmatrix} 0.9 & 0.1 & 0.0 \\ 0.1 & 0.8 & 0.1 \\ 0.0 & 0.1 & 0.9 \end{pmatrix}.$$

Each row in the transition matrix represents the probability $\mathcal{T}_{ij}$ for transitioning from a state i to a state j. This evolution of states is known as a Markov chain, because the next state z_{t+1} only depends on the current state z_t.

A stationary probability vector π can be computed for the transition matrix $\mathcal{T}$, i.e., a row vector π so that $\pi\mathcal{T} = \pi$. In our particular example, the transition matrix is called doubly stochastic, because both its rows and columns sum to 1. For doubly stochastic matrices, the stationary distribution will be uniform, i.e., $\pi = (1/3, 1/3, 1/3)$ in our case. Readers can verify that this is indeed the case by performing the matrix multiplication $\pi\mathcal{T}$. Here, by the stationary distribution we mean what the Markov chain will produce over the long run. Hence, if the time period T is sufficiently long, we expected the three different states to occur equally frequently regardless of the initial state.

Readers who are interested in exploring the characteristics of the above Markov chain further are encouraged to examine the accompanying code to this section. The code contains a function for simulating this Markov chain with some elementary analysis illustrating that it behaves as expected. The important realization one has to make is that there is stochasticity in the state, which determines the probability of the coin coming up heads p, and finally there is stochasticity in the outcome of the coin. This is an important conceptual separation. In practice, we would only observe outcomes of the coin and have to estimate both the transition matrix $\mathcal{T}$ and the associated heads probabilities p for each state. If we are only good at estimating one of them and very bad at estimating the other, our strategy is unlikely to be successful.

In the previous paragraphs, a lot new terminology has been vaguely introduced. To maintain our focus, we will not delve deeper into Markov chains and simply refer the interest readers to Norris (1997). This might be unsatisfactory for some readers, but we do not want the mathematical formality to overshadow the main points. The objective of the above presentation was to introduce the concept of market states, if we for a second imagine that the coin flipping is a market that someone would allow us to invest in. It was also to introduce the concept of short-term and long-term behavior, by showing that the current state helps us to infer the likely next state, while it does very little to help us infer the state many time-steps into the future. You can see this in practice in the accompanying code to this section.

The analysis so far has been very nice and simple, with a very well-behaved coin having constant states and state transition probabilities $\mathcal{T}$. Now let us imagine that a sudden appearance of new state $p \in [0, 1] \notin \{0.1, 0.5, 0.9\}$ can happen at a random point in time, causing a fundamental change in the transition matrix $\mathcal{T}$. Or maybe one of the existing states stops appearing all together. No matter which case it is, we will refer to any changes to the transition matrix $\mathcal{T}$ as a structural break.

Structural breaks pose additional challenges because we cannot simply lean back and enjoy the profits of our strategy that might have been very successful historically at inferring the state z_t and sizing investment into the heads outcome accordingly. We must constantly monitor the performance of our investment strategy and adapt it to the new reality. If we insist on using fully systematic strategies, we have to rely on having sufficient data showing us that a structural break has occurred. This will probably lead to a period of lower expected returns and a higher probability of losing money, with the investment strategy being suboptimal until we adjust it.

If we are allowed to adjust our strategy based on other information, for example, someone reliable communicating to us that a structural break has occurred, we have the opportunity to adapt before we see realizations of the structural break in the data. In real-world markets, such announcements could, for example, be a central bank communicating a new forward guidance on their monetary policy. The ability to mix historical data with forward-looking adjustments based on qualitative inputs is what makes the quantamental investing approach appealing.

For real-world investment markets, prices, returns, and factors have much more complex distributions than the Bernoulli distribution of the coin, and the states are much more abstract than the probability of heads. It is probably also very hard to find true stationarity in the data, because markets are constantly evolving. But the concepts of market states and structural breaks will still be valuable for us when thinking about investment markets, and you are surely going to hear investment practitioners talk about these things. At the very least, we are going to use these concepts in Section 3.2.1 about time- and state-dependent resampling methods for market simulation. Note also that practitioners sometimes refer to states as regimes.

Since real-world investment markets are governed by much more complex states, we are unlikely to be able to capture all necessary state-dependent information at any given point in time by relying solely on interpretable state variables. For example, we might use the VIX index and the slope of the interest rate curve as state variables for real-world markets, but there is likely still much more that we need to condition on to fully capture the market state. Hence, we probably need a residual state variable that we can condition on.

Time-conditioning has historically been a frequently used way to capture market state. Popular examples are the exponentially weighted moving average (EWMA) and generalized autoregressive conditional heteroskedasticity (GARCH) models. The hypothesis is that recent data tells us more about the immediate future than old data. While it is probably true that markets tomorrow are more similar to markets today than 30 years ago, it is a strong assumption that time-conditioning alone will capture all the necessary information. In this book, we will view time-conditioning as a way to capture residual information that our other state variables are potentially unable to capture.

Here comes another real-world blow. All the perspectives related to market state, structural breaks, and time-conditioning might be completely wrong for how investment markets actually behave. While we can have full control over the coin flipping experiment, we have no control over the distributions of investment markets. These perspectives are simply the ones that we currently have the mathematical machinery to handle, so our analysis will naturally be biased by them. Hence, when we analyze the stylized market facts in Chapter 2, it will be through this lens. The same goes for our simulation approaches in Chapter 3. We are, however, not stipulating that market distributions can be fully characterized by their state, or that market states behave according to a Markov chain. The hypothesis is rather that something along these lines is approximately correct, where we are at least not introducing constraints that obviously violate the characteristics of real-world markets.

1.3 The Essence of Investment and Risk Management

In this section, we move a bit closer to investment markets by starting to talk about equities and bonds instead of a coin. We will still remain in a highly hypothetical world by assuming that the return distributions of equities and bonds are jointly normal and therefore fully characterized by the first two moments, allowing us to focus on just the mean vector μ and the covariance matrix Σ. As we will see in Chapter 2, this is quite far from reality, while it is the setting for the original analysis of investment risk and return introduced by Markowitz (1952).

Similarly to the previous section, let us assume that the return distributions for the next period have various states, which we call risk off, base case, and risk on. Our first goal is to infer which state we believe is most likely and decide on a portfolio for the next period. As with the coin's outcomes, investment outcomes are only predictable in a probabilistic sense, and the states are stochastic. We will not write out a transition matrix for the states in this section, but you can imagine that something along those lines determines the market state.

We will examine how the optimal portfolio is in all of the states while noting that we must eventually select just one portfolio. Using the Entropy Pooling method presented in Chapter 5 and Section 5.3 will allow us to weight the different cases according to their probability of occurrence to arrive at a single distribution that we can optimize over. We note that the naive approach of simply weighting the individual portfolios according to the probability of their corresponding state occurrence does not guarantee mean-risk optimality over the final weighted distribution. From a portfolio optimization with parameter uncertainty perspective, this weighting can still make sense to reduce the variance of the optimal exposure estimates, see Kristensen and Vorobets (2024), Vorobets (2024), and Chapter 6 on portfolio optimization.

Let us assume that we characterize the states of bonds and equities by the following assumptions on the mean vector and covariance matrix:

$$\mu_{off} = \begin{pmatrix} 0.05 \\ 0.06 \end{pmatrix} \quad \text{and} \quad \Sigma_{off} = \begin{pmatrix} 0.08^2 & -0.2 \cdot 0.08 \cdot 0.3 \\ \bullet & 0.3^2 \end{pmatrix},$$
$$\mu = \begin{pmatrix} 0.03 \\ 0.1 \end{pmatrix} \quad \text{and} \quad \Sigma = \begin{pmatrix} 0.1^2 & 0 \\ \bullet & 0.25^2 \end{pmatrix},$$
$$\mu_{on} = \begin{pmatrix} 0.02 \\ 0.125 \end{pmatrix} \quad \text{and} \quad \Sigma_{on} = \begin{pmatrix} 0.1^2 & 0.2 \cdot 0.1 \cdot 0.2 \\ \bullet & 0.2^2 \end{pmatrix},$$

where we use $\bullet$ to indicate that the covariance matrix is symmetric.

Let us say that our investment mandate requires us to maintain a volatility level of 12.5%, while our portfolio must be long-only and allow us to invest only in cash instruments, i.e., $e_i \geq 0$ and $\sum_{i=1}^{I=2} e_i = 1$. Note that we use the derivatives framework notation from Vorobets (2022a) and Section 6.1 throughout this book. In this case, the relative market values v are equal to a vector of ones, so exposures will correspond to portfolio weights, but this will not always be the case.

The portfolio optimization problem can be solved easily in this case using second-order cone programming, see Boyd and Vandenberghe (2004) and the accompanying Python code to this section. The optimal portfolios for the three different states are given in Table 1.1.

	Risk off	Base	Risk on
Bonds	58.10%	55.12%	46.43%
Equities	41.90%	44.88%	53.57%

Table 1.1: Optimal bond/equity portfolios with a 12.5% volatility target.

While we can optimize portfolios for each state, we must decide on one portfolio to invest in. It is, however, still interesting to examine what happens to the portfolio in each of the different cases that we have imagined. For example, what happens to the base case portfolio in the risk off case. For risk management purposes, we would then ask if it is a risk that we are comfortable with, or if we should attempt to reduce or eliminate it through an adjustment of the portfolio or with tail risk hedging, as presented in Section 7.3.

Figure 1.3.1 shows the return distributions of the optimal portfolios from Table 1.1 in the risk off state. It is clear that if we decide to invest in the optimal portfolio for the base case, it will be suboptimal in the risk off case. Even worse is the optimal portfolio for the risk on state, which performs a lot worse than the risk off optimal portfolio. Considerations like these are what we will define as risk management as opposed to general diversification.

One could argue that the risk-adjusted performance loss in the risk off case is not substantial for the base case optimal portfolio, while the risk on portfolio exposes us to significant losses if we end up in the risk off case. Hence, unless we strongly believe that the risk on or risk off case is very likely, we should probably choose the base case portfolio. An important consideration is also how large the risk overshoot will be for the risk on portfolio if we are in a risk off state, something that we will examine much more carefully in Section 6.3.

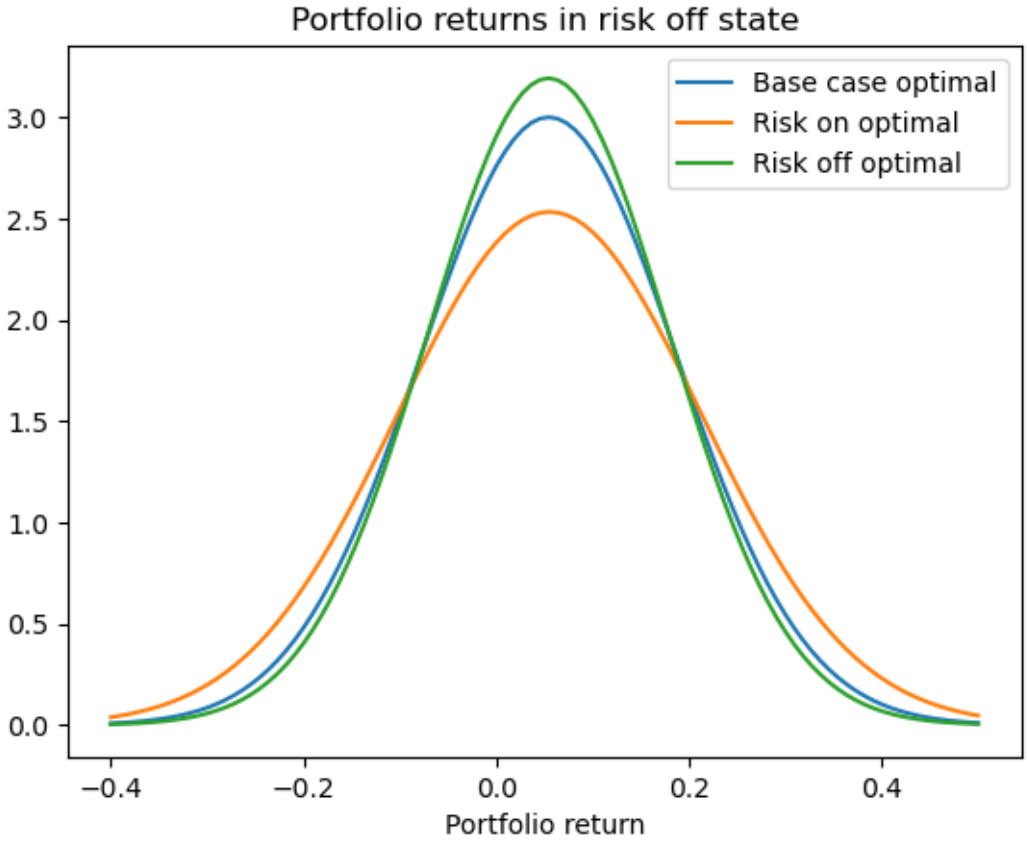

Figure 1.3.1: Portfolio return distributions in the risk off state.

Diversification, e.g., building a portfolio of both bonds and equities can of course also be seen as a form of risk management, but we will define risk management more specifically as a careful analysis of adverse scenarios and assessment of how to deal with them in this book. Diversification can be defined in many different ways, while we will loosely think about it as the ratio between the sum of the standalone risks of the individual exposures and actual portfolio risk, i.e.,

$$d = \frac{\sum_{i=1}^{I} \mathcal{R}(e_i)}{\mathcal{R}(e)},$$

where $\mathcal{R}(e_i)$ is the risk of the individual exposures, while $\mathcal{R}(e)$ is the risk of the portfolio.

The risk measure $\mathcal{R}$ can for example be CVaR, variance, or VaR. For coherent risk measures, see Artzner et al. (1999), it holds that

$$\mathcal{R}(e) \leq \sum_{i=1}^{I} \mathcal{R}(e_i), \tag{1.3.1}$$

which implies that $d \geq 1$ when the portfolio risk $\mathcal{R}(e)$ is positive, which will usually be the case. This property is formally referred to subadditivity and sometimes called the diversification principle. VaR is not a coherent risk measure, because it does not respect this diversification principle. VaR also has

several other issues, which is why CVaR is becoming the preferred investment tail risk measure for both market makers and investment managers.

Readers are encouraged to examine the accompanying Python code to this section to see how the computations have been performed for the optimal portfolios from Table 1.1 and the portfolio return plot from Figure 1.3.1. In addition, there are joint return plots for the three different states, while Figure 1.3.2 shows the joint return plots for the base and risk on states.

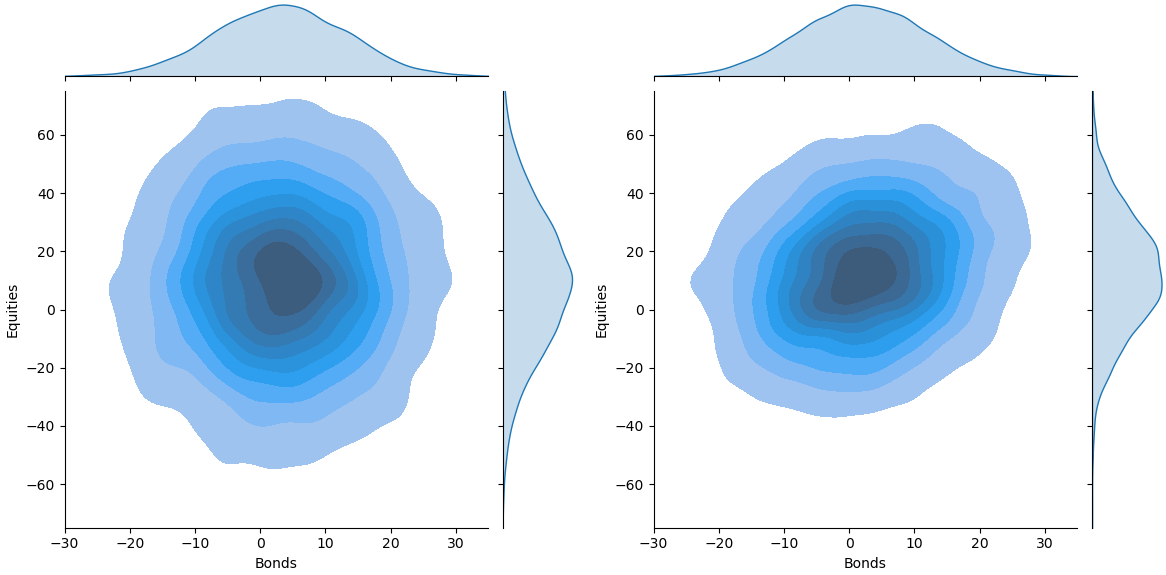

Figure 1.3.2: Joint return plots for bonds and equities in the base and risk on state.

Readers are also encouraged to spend some time thinking about what the graphs in Figure 1.3.2 show as we will be looking at graphs similar to these throughout the book. Think, for example, about which of the two graphs shows a positive correlation between bonds and equities, in which graph equities have the highest variance, and what it means that the graph is darker and lighter in different areas. In Figure 1.3.2, we have purposefully simulated the distributions to generate the Monte Carlo market simulation matrix R from (1.1.1). The central graph is a contour plot of the joint distribution, while the two axes show the marginal distributions.

Finally, take a moment to reflect on the content in this chapter. Although the perspectives have been introduced in highly oversimplified cases, the concepts of market states, structural breaks, time-conditioning, and the essence of investment and risk management will carry over to more complex cases and analysis. These concepts have simply been presented in the most basic cases to make them perfectly clear in this chapter. As we increase the complexity of the market simulation and analysis, they might become less apparent, but they will be fundamentally the same.

In the rest of this book, we will start focusing more on tail risk and risk budgeting in addition to risk contribution analysis. In some cases, we will also analyze the market simulations directly either through joint plots similar to 1.3.2 or other conditional perspectives. All this will show that once we abandon the focus on the covariance matrix, we can perform much deeper and more meaningful analysis of investment markets.

Chapter 2

Stylized Market Facts

This chapter presents some stylized market facts, i.e., typical statistical characteristics of historical multi-asset investment data. Most of these characteristics should be well-known to experienced investment managers, but they will be presented from perspectives that are perhaps new to them. For people with no or very little practical experience with investment markets, this chapter will be essential to understand how gross the mean-variance oversimplification is. Hence, mean-variance should not be used to manage investments in practice, despite it being widely promoted by some academics and other people who have a reputational or commercial vested interest in the approach.

This chapter uses daily historical data for 10 US equity indices (S&P 500 and nine sector indices), three US treasury yields (the 13 week, 10 year, and 30 year yields), and the VIX index since December 1998. This gives us 6,493 daily observations including a tech bubble burst, a financial crisis, a COVID crisis, and several monetary easing and tightening cycles. Hence, the data should be sufficient to extract some interesting historical insights about investment markets characteristics.

We start with Section 2.1 about the risk and return trade-off, which is the tendency for riskier instruments to give a higher expected return. This is quite intuitive, because most people are risk averse by nature. The risk and return trade-off was first formalized by Markowitz (1952) and Sharpe (1964) with the introduction of the academically celebrated capital asset pricing model (CAPM). As we will see in this chapter, reality is much more nuanced than the CAPM predicts.

Section 2.2 continues with an illustration of how there is a tendency for market risk to cluster over time, i.e., periods with high market volatility tend to be followed by periods with continued high volatility and vice versa. This shows that risk is somewhat predictable, while expected returns are more challenging to predict, and outcome prediction is probably close to impossible with publicly available information as explained in Section 1.2.

Section 2.3 illustrates the volatility risk premium, which is the tendency for option implied volatility to be higher than subsequent realized volatility. This empirically reveals investors' preferences for avoiding losses. Section 2.4 contains additional insights that are usually conveniently ignored by some finance and economics academics, but that any investment manager should be familiar with and incorporate into their portfolio construction and risk management. Section 2.5 summarizes the consequences of the stylized market facts for investment modeling and analysis. Finally, Section 2.6 presents a traditional backtest of CVaR and variance optimization and explains why it is naive.

2.1 Risk and Return Trade-Off

The foundation for quantitative analysis of the trade-off between risk and return was laid by Markowitz (1952) and Sharpe (1964). The CAPM model states that

$$\mathbb{E}[R_i] = R_f + \beta_i (\mathbb{E}[R_m] - R_f),$$

where

$$\beta_i = \frac{\mathrm{Cov}(R_i, R_m)}{\mathrm{Var}(R_m)} = \rho_{i,m} \frac{\sigma_i}{\sigma_m}.$$

Here, R_i is the return of instrument i, R_f is a risk-free rate, and R_m is the return of "the market", for example, a broad equity index such as S&P 500 for an equity application of the model. The CAPM model has been empirically rejected on many occasions, and many other factors that attempt to explain the expected returns of investments have been introduced, for example, by Fama and French (1992).

The CAPM is a direct consequence of the mean-variance framework, which this book has already underlined builds on fundamentally flawed assumptions about the market and investor preferences. Hence, it is no surprise that the CAPM fails empirically. However, many articles and books have been written about CAPM and mean-variance, so there are significant vested interests in maintaining the relevance of this work among some academics as well as technology and course providers. But this is of course a poor reason for using these methods to manage your own or other peoples' money.

Markowitz (1959) already recognized that the focus should rather be on the investment downside, i.e., avoiding losses. However, with the computational technology available in 1959, this was an unthinkable problem to solve in practice. In fact, estimating a covariance matrix was perceived as computationally intensive and in many cases infeasible. Luckily, we no longer live in the 1950s and have now discovered fast and stable algorithms for solving fully general CVaR optimization problems. Hence, we are no longer technologically constrained in a way that makes tail risk analysis based on realistic market simulations practically infeasible.

Below, we compute some statistics and analyze the daily historical data of S&P 500 and nine US sector equity indices. The indices are selected based on data availability to maximize the number of observations, so newer sectors like Real Estate are not included. However, the main conclusions are unlikely to change due to an inclusion of these sectors.

We first start with some descriptive statistics for the daily returns in Table 2.1. We immediately note that the daily average return is quite low but positive for all indices. The daily risk numbers look quite high compared to the average return, but we must remember that returns compound exponentially, while risk does not. This is evident from Table 2.2 and Table 2.3 that show monthly and yearly return statistics by setting the parameter $H = 21$ and $H = 252$ trading days, respectively.

We note that the kurtosis generally decreases as the horizon increases. However, it is important not to fool oneself, because the decrease in kurtosis is often combined with an increase in negative skewness. Hence, when we just look at the statistics from the table, it is important to have the combination of skewness and kurtosis in mind and remember that these statistics do not tell us the full story about how the distribution actually looks. Skewness and kurtosis will be analyzed more carefully in Section 2.4, in addition to other visually interesting aspects of the equity return distributions.

Index	Mean	Volatility	Skewness	Kurtosis	90%-CVaR
Materials	0.044%	1.502%	-0.020	9.469	2.735%
Energy	0.048%	1.821%	-0.247	13.885	3.265%
Financial	0.038%	1.817%	0.313	17.526	3.163%
Industrial	0.043%	1.338%	-0.163	10.649	2.477%
Technology	0.048%	1.633%	0.273	10.139	3.036%
Consumer Staples	0.031%	0.964%	-0.097	10.945	1.756%
Utilities	0.036%	1.224%	0.207	14.821	2.225%
Health Care	0.040%	1.130%	-0.021	12.033	2.062%
Consumer Discretionary	0.046%	1.428%	-0.226	8.958	2.644%
S&P 500	0.032%	1.221%	-0.153	12.863	2.261%

Table 2.1: Daily return statistics for US equity indices.

Index	Mean	Volatility	Skewness	Kurtosis	90%-CVaR
Materials	0.859%	6.174%	-0.513	6.429	11.766%
Energy	0.948%	7.517%	-0.541	8.456	13.610%
Financial	0.682%	6.824%	-0.341	9.502	13.157%
Industrial	0.874%	5.725%	-0.782	7.713	10.999%
Technology	0.928%	6.586%	-0.468	5.191	13.118%
Consumer Staples	0.632%	3.700%	-0.713	6.476	7.376%
Utilities	0.731%	4.878%	-0.923	7.487	9.477%
Health Care	0.796%	4.470%	-0.563	6.579	8.594%
Consumer Discretionary	0.935%	5.986%	-0.534	6.575	11.597%
S&P 500	0.613%	4.725%	-0.980	7.597	9.474%

Table 2.2: Monthly return statistics for US equity indices.

Index	Mean	Volatility	Skewness	Kurtosis	90%-CVaR
Materials	9.481%	19.089%	0.009	4.791	35.347%
Energy	11.041%	26.531%	0.201	3.192	45.323%
Financial	7.888%	23.743%	0.040	5.225	43.570%
Industrial	9.859%	19.233%	-0.130	4.700	37.393%
Technology	11.543%	25.382%	-0.588	3.260	52.990%
Consumer Staples	7.593%	11.225%	-0.694	3.862	24.169%
Utilities	8.326%	15.527%	-0.796	3.822	33.803%
Health Care	9.160%	12.983%	-0.074	3.187	23.285%
Consumer Discretionary	10.797%	19.305%	-0.016	4.200	36.436%
S&P 500	7.041%	16.677%	-0.524	3.857	33.881%

Table 2.3: Yearly return statistics for US equity indices.

We continue with a visualization of the trade-off between risk and return in Figure 2.1.1, where the historical daily 90%-CVaR is plotted against the historical average return from Table 2.1. From Figure 2.1.1, we see that there is some approximate trade-off between tail risk and return. For a more careful analysis, we could analyze individual stocks and consider whether some instruments have safe haven or convexity properties that justify a lower return similar to options presented in Section 2.3.

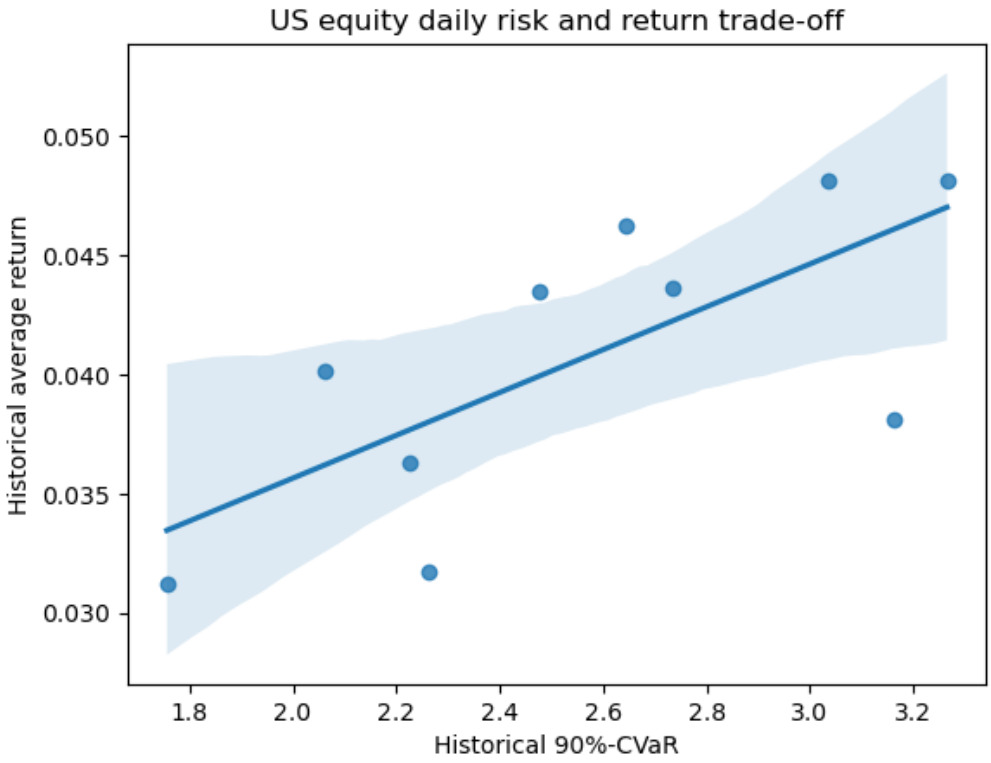

Figure 2.1.1: Historical daily US index average return and 90%-CVaR relationship from Table 2.1.

2.2 Risk Clustering

In the previous Section 2.1, we focused on analyzing the cross-sectional properties of US equity returns. In this section, we will look at the time series properties. The most important point here is that there is a tendency for risk to cluster in the sense that periods with volatile markets tends to be followed by periods with continued high volatility and vice versa. The word volatility is used in a broader sense in this section to mean variation in prices and risk factors rather than their standard deviations.

We start by simply examining the historical time series of the daily S&P 500 return in Figure 2.2.1. Interested readers can examine the accompanying code and create this plot for the other indices. From Figure 2.2.1, we note that the historical returns do not seem to be identically and independently distributed (iid), because we clearly see a clustering of periods with high/low market volatility.

The interesting question is whether the time series dependency is in the return directly or in its higher moments, i.e., whether it is the expected return or the expected risk that is predictable. Figure 2.2.2 illustrates the historical absolute return time series, where it becomes clear that it is the risk that is predictable. Readers can further examine the accompanying code and see regression plots, where they can see that the correlation between the absolute returns is positive, while the correlation between daily returns is close to zero and statistically insignificant. As the risk clustering result is well-known and quite easy to see, we will not spend more time on it and instead focus on more subtle results in Section 2.4.

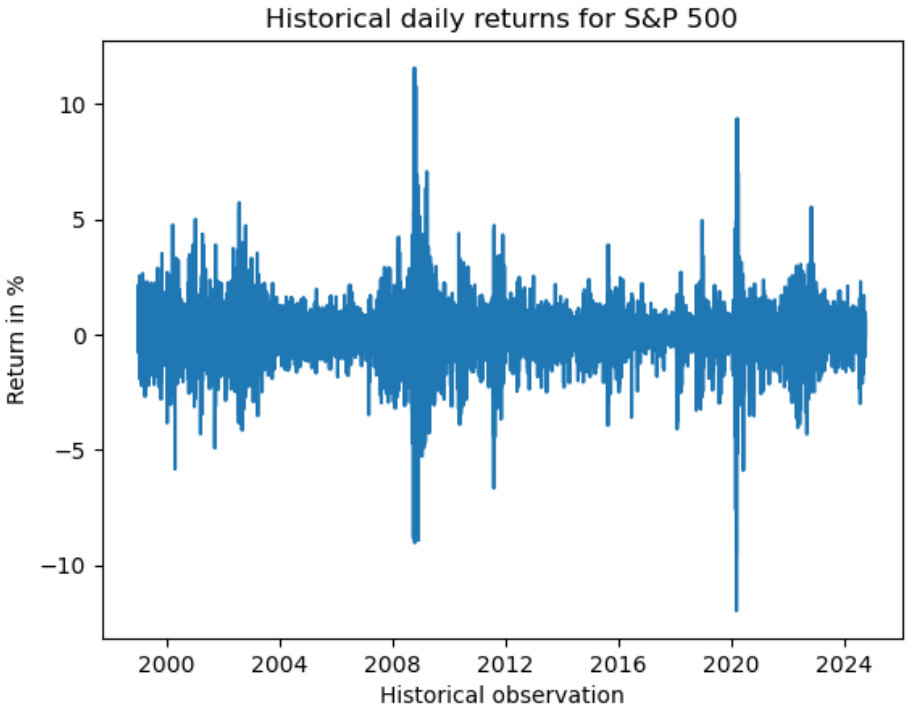

Figure 2.2.1: Historical daily returns for S&P 500.

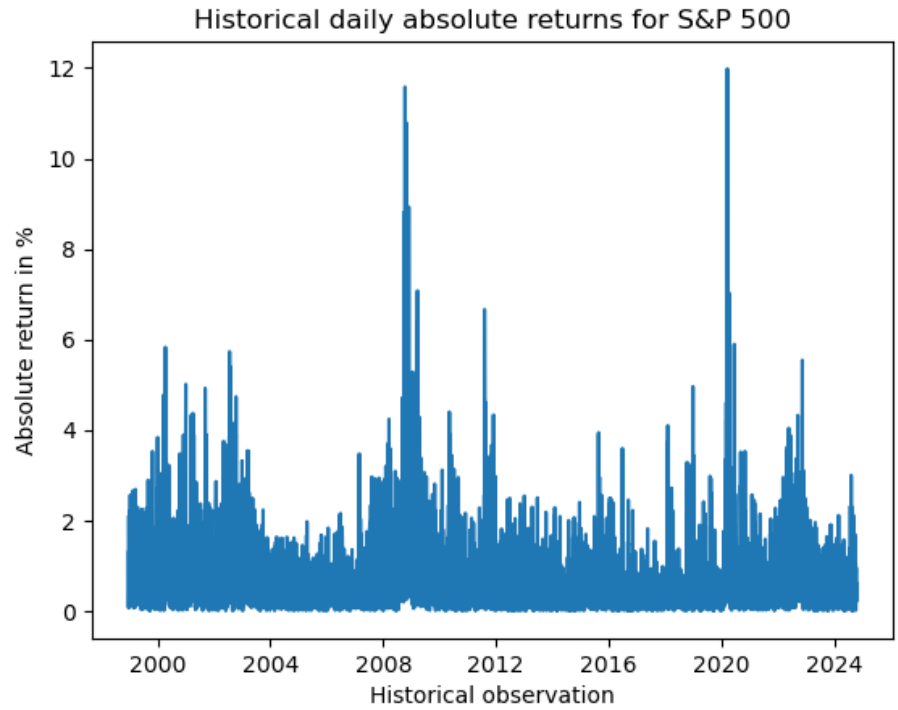

Figure 2.2.2: Historical daily absolute returns for S&P 500.

2.3 Volatility Risk Premium

The volatility risk premium refers to the fact that option implied volatility $\sigma_{implied}$ tends to be higher than subsequent realized volatility $\sigma_{realized}$. To understand what this means, this section briefly introduces European-style options for readers who are completely unfamiliar with derivatives. We start by noting that European refers to the exercise properties of the options, not geography. For a more detailed introduction to options and other derivatives, see Hull (2021).

The two most common options types are put and call options. A put/call option gives the option holder the right, but not the obligation, to sell/buy an underlying security at a predetermined strike price K. Hence, a put option's payoff at expiry is $\max(K - S_T, 0)$, while a call option's payoff is $\max(S_T - K, 0)$, where S_T is the value of the underlying security at the option's expiry time T. The payoff profiles for put and call options are shown in Figure 2.3.1 as a function of the underlying value S_T. As we see from Figure 2.3.1, the option payoff introduces convexity, i.e., we win something if the market moves in the direction we want and do not lose anything, besides the initial option price, if it moves in the opposite direction.

Options have many characteristics that are similar to insurance contracts. For example, the payoff of an insurance on your house or car is very similar to a put option payoff, if you think of your house or car as the underlying of the insurance contract. There is obvious value in such insurance contracts, so they have a premium associated with them. In this regard, it is important to separate between exposure/notional, which you can think of as the value of your house or car, and option market value, which you can think of as the price you pay for the insurance. See Section 6.1 for more precise definitions of exposures and (relative) market values.

Figure 2.3.1: Option payoff profiles for strike $K = 100$.

There are various ways in which market makers price options, but they most commonly use stochastic differential equations (SDEs) as presented in Section 3.2.3. The most important principle is that the prices they quote should not allow for arbitrage, which is a risk-free profit opportunity. SDEs offer nice guarantees when it comes to no-arbitrage, but they are usually a poor description of actual high-dimensional market behavior as explained in Section 3.2.3.

Although each market maker has their own pricing models and quote prices that probably do not allow arbitrage, there is a market convention to quote option prices in terms of the implied volatility $\sigma_{T,K}$, which is basically a maturity T, strike K, and underlying level S_t normalized price for the option. The implied volatility is calculated using the famous Black and Scholes (1973) formula, which assumes that the underlying security follows a geometric Brownian motion, see Section 3.2.3.

In relation to options' implied volatilities, there is a popular index known as the VIX, commonly referred to as "the fear index", which is designed to be a replication of a variance swap strike σ_{strike} based on listed options' strikes and expiries. A variance swap is a derivative that allows investors to speculate directly on the implied variance of some underlying and has payoff $\left(\sigma^2_{realized} - \sigma^2_{strike}\right) N_{var}$, where N_{var} is the (variance) notional/exposure of the variance swap contract. We can think of VIX as being the approximate σ_{strike} for a one-month variance swap. The difference is that the variance swap is an over-the-counter (OTC) instrument that is not traded on exchanges, so the strike σ_{strike} can be calculated more precisely when we are not limited to exchange traded options.

Figure 2.3.2 illustrates the historical relationship between S&P 500 and the difference between realized volatility and one-month lagged VIX. This helps us understand the approximate behavior of a variance swap contract and leads to a historical volatility risk premium estimate of 3.64%.

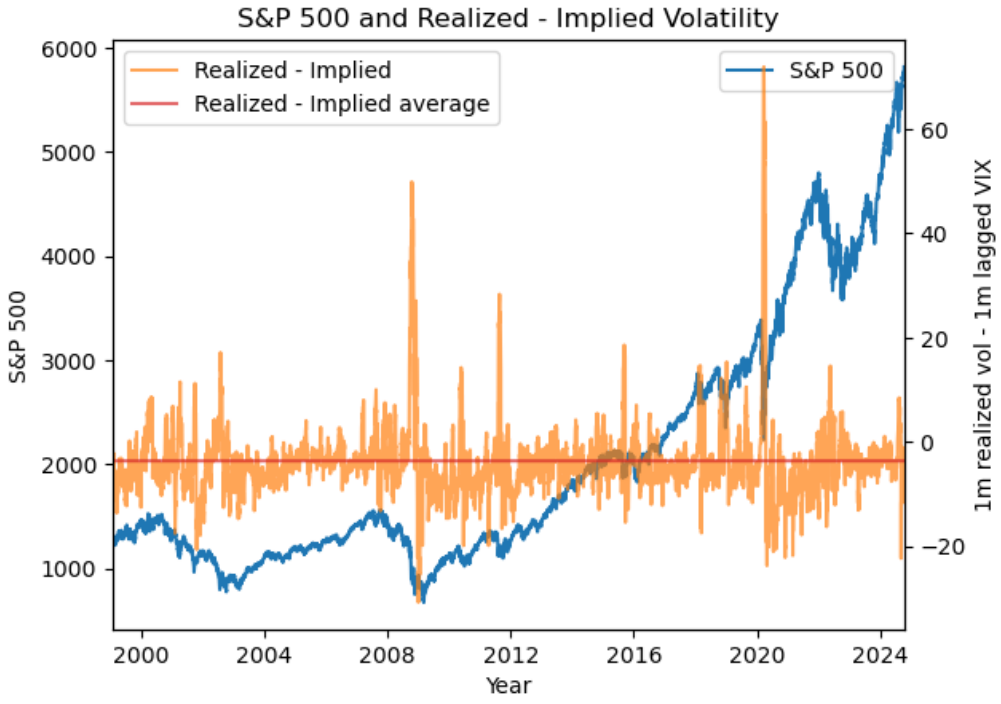

Figure 2.3.2: Historical relationship between S&P 500 and realized - implied volatility.

2.4 Skewness, Kurtosis, and Other Interesting Insights

This chapter has so far focused on the equity time series and statistics. This section will incorporate some multi-asset perspectives and examine skewness and kurtosis further. All these characteristics will be important for us to keep in mind when modeling investment markets in Chapter 3 and subsequently analyzing and optimizing portfolios as presented in chapters 5-7.

We start by visualizing what the skewness and kurtosis numbers from Section 2.1 actually look like. Looking just at the statistics makes it hard to get a sense of how the distributions are, because the same statistics can lead to very different distributional shapes. It is also important to keep the combination of skewness and kurtosis in mind. For example, a distribution that has slightly more kurtosis than the normal distribution but significant negative skewness can have very significant left tails. A good example of this is the Technology index, where the yearly kurtosis of 3.26 might seem like it is "close to a normal distribution", but the combination with a significant negative skewness of -0.588 can introduce some very significant left tails as shown in Figure 2.4.1. Readers can use the accompanying code to create graphs similar to Figure 2.4.1 for the other equity indices and verify that they are skewed and fat-tailed in complex ways.

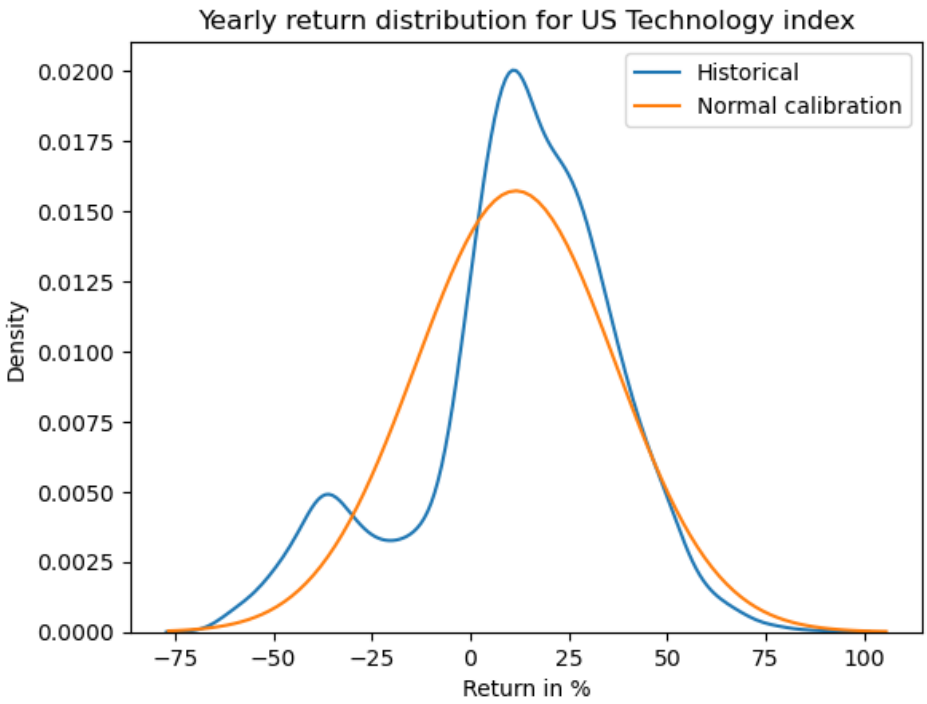

Figure 2.4.1: Yearly return distribution for the US Technology equity index.

From Figure 2.4.1, we can conclude that investment return distributions can be skewed and fat-tailed in complex ways, i.e., it is not just an adjustment of some analytically known distribution that allows us to approximate their empirical properties well. Another important point is that if we use oversimplified analytical distributions, such as the normal or t-distribution, we are not only introducing a large approximation error for one investment in our portfolio but likely for all of them.

We can also conclude that just looking at statistics such as the mean, variance, skewness, kurtosis, and correlation can be misleading, because they can effectively remove many of the nuances of investment distributions. Hence, it is important to actually visually look at the data to understand its characteristics. Once we do that, it becomes immediately clear that the elliptical distribution assumption inherent to mean-variance analysis is grossly oversimplifying for the marginal distributions.

Another important point about elliptical distributions, which many people tend to forget, is that they only allow for linear and cross-sectionally constant dependencies. However, this is also a gross oversimplification of investment market dependencies. Not only for portfolios that contain nonlinear derivatives such as put and call options, where the violation is obvious, but also for plain vanilla portfolios that contain cash equity indices and bonds. For example, if we look at the 10% worst yearly historical scenarios for S&P 500, aligning with the 90%-CVaR focus from Section 2.1, we can compute the correlation matrix in these scenarios and the remaining 90% of scenarios. This is done in Table 2.4 and Table 2.5 below, where we clearly see a significant change in the correlations, e.g., column 4.

	0	1	2	3	4	5	6	7	8	9
0, Materials	100.0	54.8	89.7	88.5	-42.7	54.4	56.0	62.5	85.6	83.2
1, Energy	54.8	100.0	60.8	77.1	-10.9	52.3	65.0	52.4	23.2	63.1
2, Financial	89.7	60.8	100.0	91.7	-40.4	45.8	53.7	72.5	80.1	85.9
3, Industrial	88.5	77.1	91.7	100.0	-30.6	65.1	71.6	68.2	72.8	87.7
4, Technology	-42.7	-10.9	-40.4	-30.6	100.0	-35.0	-35.0	4.7	-29.6	5.4
5, C. Staples	54.4	52.3	45.8	65.1	-35.0	100.0	89.4	7.0	45.3	40.8
6, Utilities	56.0	65.0	53.7	71.6	-35.0	89.4	100.0	12.7	40.2	44.6
7, Health Care	62.5	52.4	72.5	68.2	4.7	7.0	12.7	100.0	58.4	81.7
8, C. Discretionary	85.6	23.2	80.1	72.8	-29.6	45.3	40.2	58.4	100.0	76.7
9, S&P 500	83.2	63.1	85.9	87.7	5.4	40.8	44.6	81.7	76.7	100.0

Table 2.4: Correlation matrix for the 10% worst S&P 500 scenarios.

	0	1	2	3	4	5	6	7	8	9
0, Materials	100.0	43.4	65.9	83.7	50.5	47.3	32.4	51.3	70.8	76.2
1, Energy	43.4	100.0	34.5	44.0	3.3	16.4	34.9	9.1	5.0	29.1
2, Financial	65.9	34.5	100.0	82.8	42.9	51.3	31.6	61.4	71.2	80.1
3, Industrial	83.7	44.0	82.8	100.0	57.2	55.6	36.9	57.8	75.6	88.0
4, Technology	50.5	3.3	42.9	57.2	100.0	14.6	4.4	51.1	60.7	82.3
5, C. Staples	47.3	16.4	51.3	55.6	14.6	100.0	58.8	55.3	52.7	50.0
6, Utilities	32.4	34.9	31.6	36.9	4.4	58.8	100.0	20.6	17.3	30.7
7, Health Care	51.3	9.1	61.4	57.8	51.1	55.3	20.6	100.0	69.2	72.5
8, C. Discretionary	70.8	5.0	71.2	75.6	60.7	52.7	17.3	69.2	100.0	83.0
9, S&P 500	76.2	29.1	80.1	88.0	82.3	50.0	30.7	72.5	83.0	100.0

Table 2.5: Correlation matrix for the 90% best S&P 500 scenarios.

It should be quite obvious that if we build our portfolios based on the assumption that marginal return distributions do not have fat left tails such as in Figure 2.4.1, and that dependencies behave in the same way in the tail risk scenarios as in the other scenarios, this is a recipe for disaster. We note once again that the correlation statistic gives us a very crude understanding of the dependencies. Many

important nuances are probably missing from just looking at the correlations, similar to the issues from looking just at skewness and kurtosis statistics. Readers are encouraged to use the accompanying code to explore the data further.

Next, we perform an analysis similar to the above but for multi-asset daily log returns. First, we compute constant maturity zero-coupon bond prices, see Section 4.1, from the interest rate data, and then compute correlation statistics for the daily return series conditional on the VIX index change being above or below the 90th percentile. Table 2.6 and Table 2.7 shows the correlation matrices, while Figure 2.4.2 shows the daily 2y bond returns plotted against daily S&P 500 returns.

We can draw several conclusions from these daily return tables and joint return figure. First of all, from Figure 2.4.2 we see that VIX spikes and adverse S&P 500 scenarios tend to occur simultaneously, which agrees with our conclusion from Figure 2.3.2. Second, we can see that there has historically been a lower correlation between bond and equities in these market stress scenarios, as evident by the lower correlations in Table 2.6.

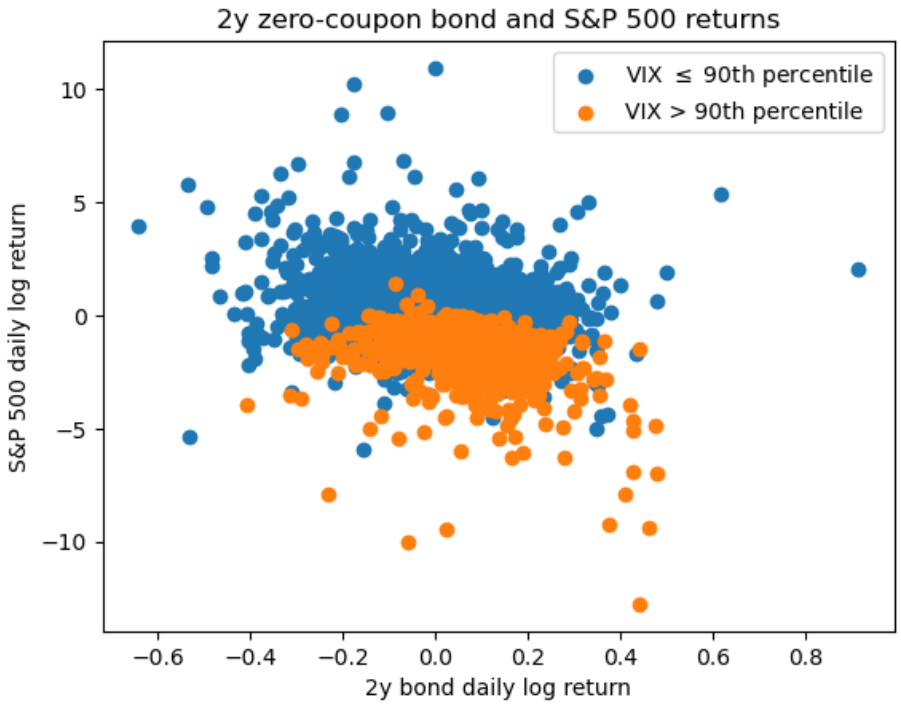

Figure 2.4.2: 2y zero-coupon bond and S&P 500 daily log returns.

	0	1	2	3	4
0, S&P 500	100.0	-25.3	-32.9	-32.6	-46.1
1, 13w bond	-25.3	100.0	29.8	17.9	8.4
2, 2y bond	-32.9	29.8	100.0	93.5	19.4
3, 30y bond	-32.6	17.9	93.5	100.0	18.2
4, VIX	-46.1	8.4	19.4	18.2	100.0

Table 2.6: Correlation matrix conditional on daily VIX changes being above the 90th percentile.

	0	1	2	3	4
0, S&P 500	100.0	-5.4	-19.7	-19.6	-65.8
1, 13w bond	-5.4	100.0	18.2	13.7	1.6
2, 2y bond	-19.7	18.2	100.0	92.9	13.6
3, 30y bond	-19.6	13.7	92.9	100.0	13.4
4, VIX	-65.8	1.6	13.6	13.4	100.0

Table 2.7: Correlation matrix conditional on daily VIX changes being below the 90th percentile.

2.5 Consequences for Modeling and Analysis

Although this chapter has performed only a brief analysis of historical investment data, it has made it clear that the elliptical distribution assumption inherent to mean-variance analysis is a gross oversimplification of reality. Reducing the market to a mean vector μ and a covariance matrix Σ effectively removes many of the important nuances that make the difference between good and bad portfolio construction.

Below is a summary of the most important conclusions:

1. Return distributions do not follow nice bell-shaped curves. They are skewed and fat-tailed in complex ways.

2. Cross-sectional asset dependencies are not just linear and constant, even for plain vanilla instruments such as stocks and bonds.

3. There is a tendency for risk to cluster, i.e., periods with high market volatility tend to be followed by periods with continued high volatility and vice versa.

4. There exists a volatility risk premium, which indicates that investors are willing to pay a premium above fair value for convexity and, hence, perceive risk as losses instead of all deviations from the mean.

So, when we model investment markets in Chapter 3, we must use methods that are capable of capturing all of the above characteristics, which a mean vector μ and a covariance matrix Σ is not capable of. In general, distributional statistics can be misleading and hide many important nuances of investment markets data. Therefore, we must focus on generating market scenarios as represented by the simulation matrix R and associated joint probability vector p, see Section 1.1.

As we will see in the following chapters, generating fully general simulated market paths and analyzing them is much harder than estimating a mean vector $\hat{\mu}$ and a covariance matrix $\hat{\Sigma}$ and subsequently analyzing these using some variation of mean-variance. However, if something is obvious and easy, it is also unlikely to produce better results than the market. The complexity and more careful attention to detail is what allows us to outperform the average investor.

A final important point is that market simulation and analysis go hand in hand. It does not make a big difference if we generate very good market simulation scenarios R and subsequently reduce them to a mean vector and covariance matrix for mean-variance analysis, because this effectively removes the important nuances that allow us to build portfolios in a clever way.

2.6 Naive CVaR and Variance Optimization Backtesting

While the empirical analysis and arguments given in this chapter should be sufficient to make any logically thinking person abandon the use of mean-variance for investment management in practice, requests are occasionally made about historical backtests "proving" that CVaR produces better results than variance. This section provides such a backtest and explains why it is naive.

We can logically conclude that mean-CVaR analysis is more meaningful given that investors have an aversion to losses, as the volatility risk premium from Section 2.3 illustrates, and the fact that mean-CVaR will coincide with mean-variance in most practical cases when the mean-variance assumptions are satisfied, see Vorobets (2022b). Hence, mean-variance can be thought of as a simple subset of mean-CVaR under highly oversimplified and unrealistic market assumptions. These are some of the valid arguments for transitioning to mean-CVaR, in addition to the fact that CVaR analysis gives meaningful insights for fully general joint distributions.

The remainder of this section proceeds to perform a standard expanding window backtest, which is frequently produced and shared in various empirical studies as well as shown to investment clients. We use the equity data that is described in the introductory section of this chapter and analyzed throughout. We start with some descriptive statistics of the quarterly returns in Table 2.8.

Index	Mean	Volatility	Skewness	Kurtosis	90%-CVaR
Materials	2.50%	9.92%	-0.48	5.99	18.89%
Energy	2.79%	12.38%	-0.22	5.35	22.66%
Financial	1.98%	11.45%	-0.07	9.35	22.41%
Industrial	2.54%	9.27%	-0.62	5.77	18.62%
Technology	2.73%	11.01%	-0.62	4.24	22.76%
C. Staples	1.85%	5.81%	-0.59	4.49	11.61%
Utilities	2.12%	7.61%	-0.62	4.80	15.10%
Health Care	2.31%	6.79%	-0.58	4.22	13.15%
C. Discretionary	2.65%	9.45%	-0.29	5.08	18.20%
S&P 500	1.77%	7.64%	-0.79	5.69	15.69%

Table 2.8: Quarterly US equity index return statistics.

Not surprisingly, we see from Table 2.8 that the quarterly returns are also skewed and fat-tailed in complex ways as well as very far from being normally distributed. All of this is similar to the daily, monthly, and yearly returns presented in Section 2.1.

We next proceed to the backtest, which is designed in the following way. As we have roughly 25 years of historical data (assuming 252 trading days in a year), we use approximately the first 10 years as the initial in-sample data and expand the in-sample period with 63 days at each iteration, which corresponds to a yearly quarter. This gives us a total of $15 \cdot 4 = 60$ quarters where we minimize the CVaR and variance of a long-only portfolio having the 10 US equity indices from Table 2.8 as its investment universe. We rebalance the portfolios to these optimized exposures every quarter. Readers can find all the details in the accompanying code to this section and adjust the backtest parameters.

The historical performance of the CVaR and variance optimizations are given in Figure 2.6.1, while the historical outperformance of the CVaR portfolio is given in Figure 2.6.2.

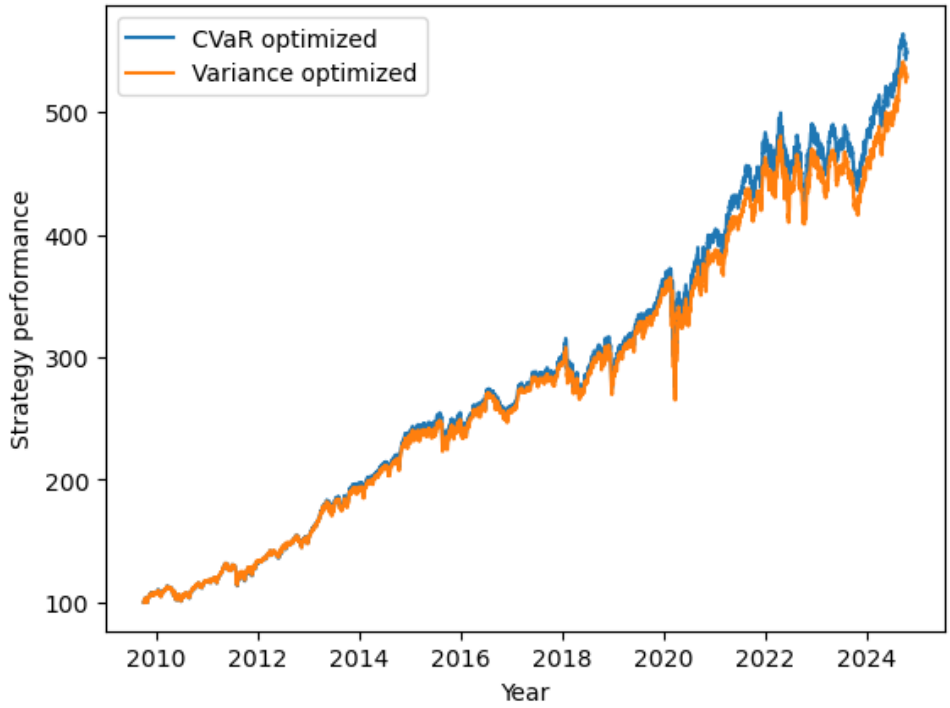

Figure 2.6.1: CVaR and variance optimized historical performance.

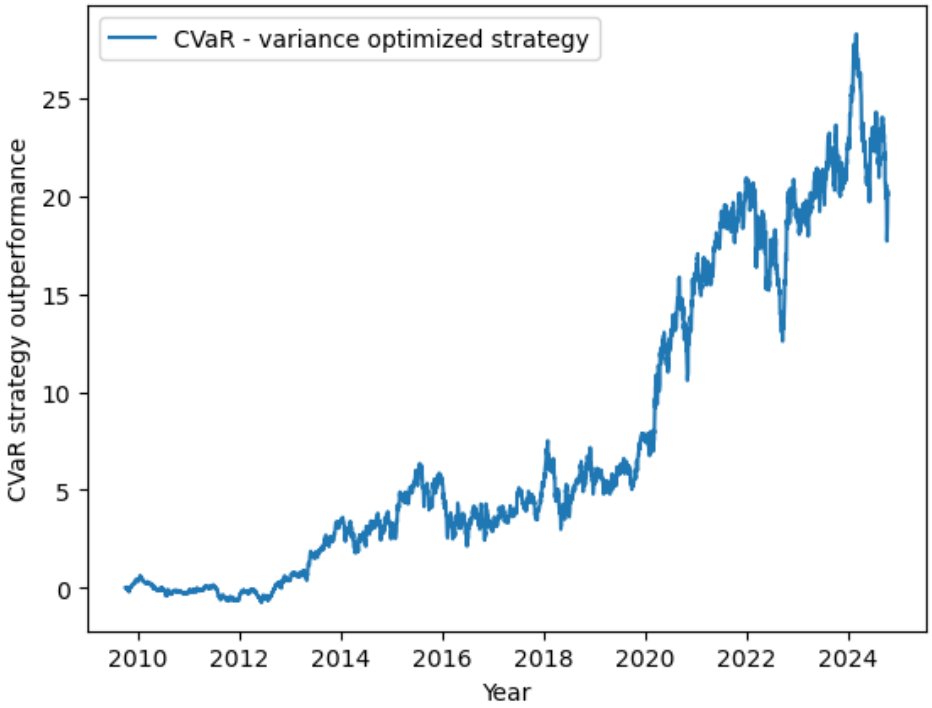

Figure 2.6.2: CVaR vs variance historical outperformance.

What can we conclude from Figure 2.6.1 and Figure 2.6.2? That CVaR is a strictly better risk measure, which gives better performance over time and especially outperforms during market sell-offs? It certainly seems like it if we use the backtest and its approximately 25% cumulative outperformance. Some people would happily make that conclusion and often do that when presenting new risk measures or pitching investment strategies.

However, the only thing we really can conclude is that we are able to find a backtest configuration where the CVaR optimized portfolio has performed best historically. It is almost certain that if we adjust the backtest configuration, we could find one where the variance optimized portfolio has better performance. Hence, drawing generalized conclusions based on one historical realization and backtest, especially without knowing how we decided to perform exactly this backtest, is dangerous.

We can, however, conclude that there are cases where CVaR optimized portfolios behave as expected. So, there are some indications that our logical reasoning and practice coincide. Making the conclusion that CVaR optimized portfolios will always generate better performance is however premature, and we probably will not be able to draw that conclusion at any point. The best we can hope for is that CVaR optimized portfolios probably lead to more desirable performance characteristics given the empirical market facts and investor preferences for avoiding large losses.

Section 3.5 presents better approaches for designing backtests that will likely allow us to make generalized conclusions with more confidence. These backtests use synthetically generated market paths that preserve the main characteristics of the historical data, while giving us new paths to validate our strategies. Backtests like the one presented in this section will surely be able to fool novice investors, and some people willingly use them to do that. However, this book is focused on methods and approaches that are likely to give you better tail risk-adjusted performance in practice, because reality always eventually catches up if we are not careful in our logic and analysis.

Chapter 3

Market Simulation

This chapter focuses on generating realistic future paths R_h, $h = 1, 2, ..., H$, for the factors, prices, and returns that are relevant for our investments and portfolios. By realistic, we mean simulations that are capable of capturing the stylized facts presented in Chapter 2 in addition to other subtle nuances. The chapter will introduce several simulation approaches; one based on resampling and some based on generative machine learning methods such as variational autoencoders (VAEs) and generative adversarial networks (GANs).

The simulations use observed historical time series $D \in \mathbb{R}^{T \times N}$, where T represents the number of historical observations, while N represents the number of time series. We can think about the data as being aligned by days, while the frequency can be different. We note that there might be additional challenges associated with extreme frequencies, e.g., very short or long horizons. For short horizons such as intraday data, the analysis might be significantly affected by market microstructure elements, while longer horizons simply give us fewer observations, making it potentially harder to learn the patterns of the data.

Investment time series have many unique challenges simply due to the constantly changing nature of investment markets, but also due to the need to capture the dependencies both in the cross-section and across time. The many structural breaks and few constraints on the possible future outcomes make it one of the most challenging problems, especially considering the potentially high-dimensional nature of the time series that we want to simulate future paths for.

The simulation approach based on historical data lends itself to the trivial critique that history will never repeat itself. While this is true, it will in most cases be hard to argue for discarding the historical data of the markets and risk factors that we are trying to model. It is also important to underline that we make no assumptions about history repeating itself, because this is obviously false. What we instead try to do is to generate new synthetic samples that preserve the characteristics of historical observations while giving us entirely new paths for the future behavior of investment markets.

In the "quantamental" spirit of this book, we can make discretionary adjustments based on information that is not available in the historical data by using the Sequential Entropy Pooling method, thoroughly presented in Chapter 5. The causal and predictive framework from Section 5.4 specifically allows us to hypothesize about future relationships that have not been present in historical data while respecting the core characteristics of our simulations R_h, $h = 1, 2, \ldots, H$.

3.1 Stationary Transformations

The first step is to transform the observed historical time series $D \in \mathbb{R}^{T \times N}$ into stationary data $ST \in \mathbb{R}^{\tilde{T} \times \tilde{N}}$. Formally, we say that a stochastic process is stationary if its unconditional joint probability distribution does not change when shifted in time. For more on time series stationarity, see Hamilton (1994). Loosely speaking, we can think of stationary data as having some repeatable statistical patterns that we are able to learn from. For more on statistical learning, see Hastie, Tibshirani, and J. Friedman (2009).

The harsh reality of investment markets is that they are probably not stationary in the strict theoretical sense due to constant structural breaks, presented in Section 1.2. Hence, the best we can hope for is likely approximate stationarity and repeatability. But just because the market simulation problem is really hard, and we are aware of the limitations, it does not mean that we cannot learn something useful from historical data. We must just not fool ourselves into believing that we have estimated the "correct" model that market realizations are generated from. This model probably does not exist in reality. Hence, we are looking for models and methods that allow us to capture the complexities of investment markets and are likely to be useful in the future. Reducing the market to a covariance matrix and a mean vector fails in relation to capturing these complexities.

A natural question is: why do we bother transforming the data into something stationary and not just use the raw data? This is simply due to the fact that most statistical models work best on data that is stationary. So, we can see it as a part of preprocessing similar to data normalization. Some models might perform well without any preprocessing, but it is more the exception than the rule. Hence, the general idea is that we transform the raw data into something stationary, project these stationary transformations into the future, and then compute the desired market simulations R_h, $h = 1, 2, \ldots, H$. A subtle nuance of this approach is that we must be able to recover the quantities that we are actually interested in from the simulated stationary transformations. For most cases, this is not a problem, but it is an important aspect to keep in mind.

We can, without loss of generality, focus on simulating market risk factors as defined in Section 1.1, because the prices and returns of our investable instruments are functions of these risk factors, see Chapter 4 on instrument pricing. Sometimes a stationary transformation will be very closely related to the P&L that we are interested in simulating. For example, for equities the log return is a good candidate for a stationary transformation that we can use for projection. With the risk factor simulation at hand, it should be straightforward for us to transform them into the price or return simulations that we want to analyze for investment and risk management purposes.

To understand what we are trying to achieve with our approach, consider for a moment an $AR(1)$ process

$$X_t = \varphi_0 + \varphi_1 X_{t-1} + \varepsilon_t,$$

with $\varepsilon_t \overset{iid}{\sim} \mathcal{N}\left(0, \sigma^2\right)$. If $|\varphi_1| < 1$, we call the process stationary. If $\varphi_1 = 1$, we call the process a random walk. If we imagine that we know the values φ_0 and φ_1, we can easily transform the raw data observations X_t, $t = 0, 1, 2, \ldots, T$, into noise observations ε_t in the following way

$$\varepsilon_t = X_t - \varphi_0 - \varphi_1 X_{t-1}$$

and focus on just estimating σ.

With an estimate $\hat{\sigma}$ of σ and some initial value x_0, we can simulate the $AR(1)$ process by simply sampling iid from the normal distribution with mean zero and variance $\hat{\sigma}^2$. In this realization lies an important point. Even if the original time series observations X_t are highly dependent over time, it is possible to perform some transformations where the fundamental uncertainty is iid, which is the case for ε_t. Some people struggle to understand how one can transform originally correlated data into something that is iid. The $AR(1)$ process gives a simple and easy to understand example. These ideas will be particularly important to keep in mind in Section 3.2.2 when variational autoencoders (VAEs) are introduced, because the encoder's purpose will be to estimate parameters that transform the data into something that has an iid Gaussian distribution, while the decoder's objective is to reverse this transformation and reintroduce the time series dependencies.

The $AR(1)$ process can be stationary or non-stationary. In the non-stationary case with $\varphi_1 = 1$, we can transform the process to a stationary one by difference, i.e., work with

$$\Delta X_t = X_t - X_{t-1} = \varphi_0 + \varepsilon_t,$$

from which it follows that $\Delta X_t \overset{iid}{\sim} \mathcal{N}\left(\varphi_0, \sigma^2\right)$. Investment price time series, for example, the S&P 500 and STOXX 50 equity indices are considered to be non-stationary processes. Hence, we must perform some differencing to transform this data into something stationary, for example, the log return.

Even in cases where we have stationary data, for example, interest rate or implied volatility time series, we might want to perform additional transformations to make the data even nicer for us to work with. By nicer, we mean something that comes closer to being iid, because iid processes like ΔX_t in the example above simplify the estimation for us. Highly persistent processes with φ_1 close to 1 can be challenging for many statistical time series models if they are applied to the raw series.

Real-world investment data usually does not follow simple process like the $AR(1)$ process above. Hence, hoping that we can transform the raw data into something that is iid without overfitting to the particular historical sample is probably a bit naive. If we had data that was iid through time, we could simply focus on estimating the cross-sectional dependencies. However, since this is not the case, we still need models that are capable of capturing both cross-sectional and time series dependencies, but we are trying to make the latter as simple as possible to make it easier for our models to learn.

Throughout this chapter, we will work with the time series simulation that follows with the fortitudo.tech Python package. This is simply because it contains the common type of data that we have for investment markets, i.e., prices, interests rates and spreads, as well as implied volatility surfaces. It is hard to find real-world data for these quantities that can be distributed with the book, so we must rely on simulated time series. However, the transformation and analysis performed in this chapter can be applied to the real-world data that readers have licenses for. Real-world data will simply have more complex dynamics than the stochastic differential equation (SDE) simulation that follows with the fortitudo.tech Python package. See Section 3.2.3 for more perspectives on SDEs and the no-arbitrage condition.

The accompanying code to this section contains suggestions for the transformations we can perform for equity price series, interest rates and spreads, as well as volatility surfaces. These transformations are fairly simple and easy for us to invert. Hence, they satisfy the recovery requirement.

The figures below show the characteristics of the stationary transformations, starting with a comparison between the raw equity index time series in Figure 3.1.1 and its log return stationary transformation in Figure 3.1.2. It is clear from Figure 3.1.2 that the data looks like something which is closer to being iid, or at the very least less persistent than the raw time series in Figure 3.1.1. However, it is also clear that we have not removed all time series dependencies, as there appears to be some clustering in the magnitude of the stationary transformations in 3.1.2. Hence, our statistical simulation models must still be able to capture both the cross-sectional and time series dependencies.

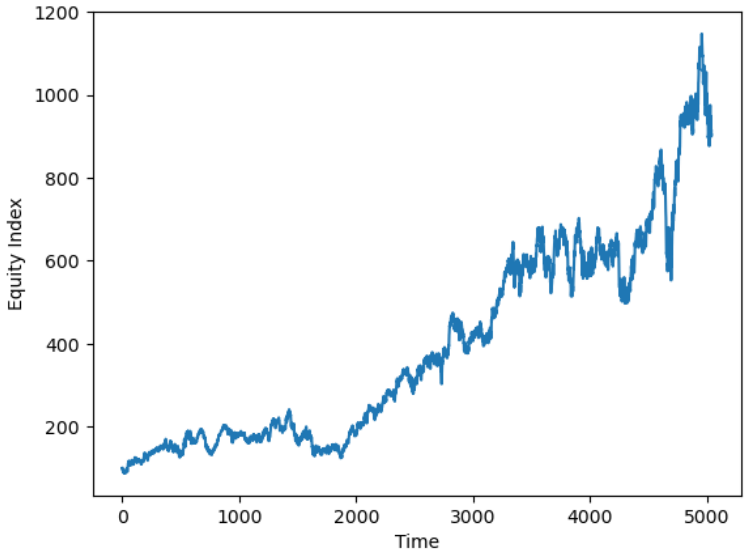

Figure 3.1.1: Raw equity index time series.

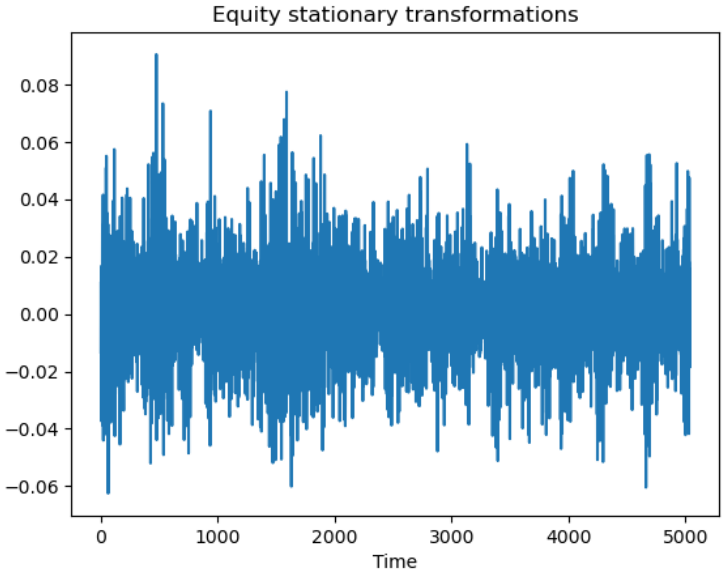

Figure 3.1.2: Equity index stationary transformation time series.

The remaining figures in this section show the stationary transformation time series for zero-coupon interest rates, implied volatilities, and credit spreads. What we see from these figures is that they from a statistical point of view are quite similar to the stationary transformation of the equity index in Figure 3.1.2. In this realization lies a key point. We do not care about which asset class or risk factors we are working with. The only thing that matters is that we are able to transform it into something that has nice properties for simulating future paths. The label we put on the data does not matter for our generative models. It is only the statistical properties of the data that matters.

Figure 3.1.3: Zero-coupon interest rates stationary transformations.

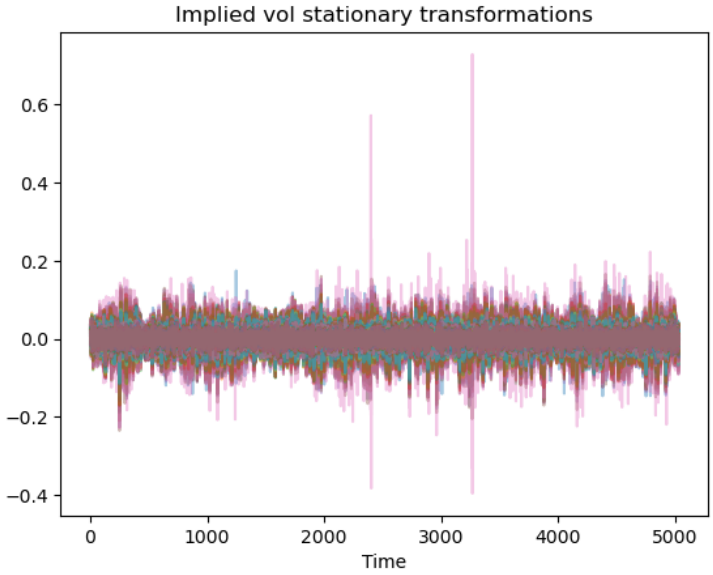

Figure 3.1.4: Implied volatility stationary transformations.

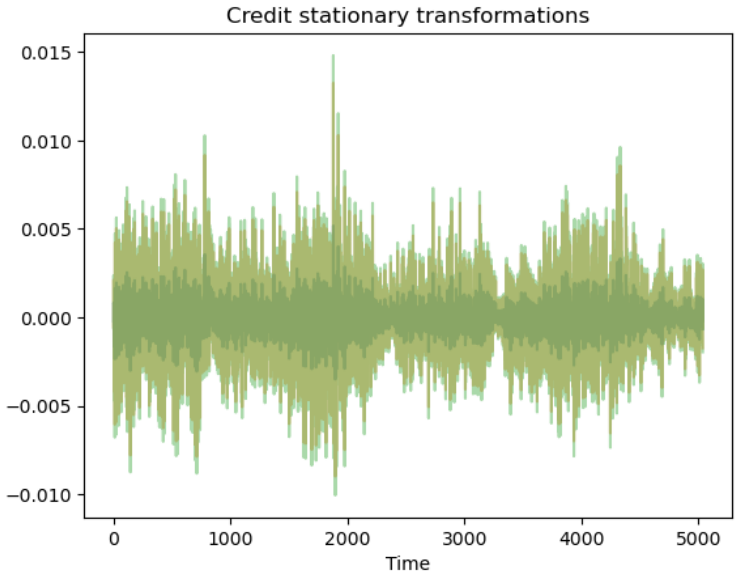

Figure 3.1.5: Credit spread stationary transformations.

You can find the details of the stationary transformations in the accompanying code to this section, where you will also see how the various graphs have been generated. It is important to underline that the stationary transformations we propose are just examples. You might be able to come up with other stationary transformations that work even better. The important point is that you keep the objective in mind, i.e., making the data easier for statistical models to work with and generate paths from. It is also important that you are able to transform the stationary transformations to the risk factors that you are actually interested in without loss of information. In most cases, the reverse transformation does not cause any issues, but it is an important aspect to keep in mind.

Note also that the stationary transformations $ST \in \mathbb{R}^{\tilde{T} \times \tilde{N}}$ can have other dimensions than the historical time series $D \in \mathbb{R}^{T \times N}$. For example, we might have $\tilde{T} = T - 1$ if we have performed differencing to achieve stationarity. Similarly, we might have performed some transformations that reduce or increase the number of time series such that $N \neq \tilde{N}$. In the examples that we will work with in this book, we will only have a reduction in the time series dimension due to differencing and thus $\tilde{T} = T - 1$ as well as $N = \tilde{N}$, but you should not be limited by this in practice.

3.2 Projection of Stationary Transformations

This section introduces several time series simulation approaches; one based on bootstrap resampling and some based on generative machine learning methods. The idea is to apply these approaches to the historical stationary transformations ST from Section 3.1, potentially after some additional preprocessing. Common for both approaches is their focus on direct future path generation rather than, for example, estimation of the next day's covariance matrix like GARCH-type models. As already discussed, reducing the market risk to a covariance matrix is very oversimplifying.

When deciding between the resampling method or the generative machine learning methods, it

is important to understand that resampling methods usually have less capability when it comes to capturing highly complex time series dependencies, while they are excellent at capturing the cross-sectional dependencies no matter how complex they are. The opposite holds for generative machine learning methods, which are very capable when it comes to capturing time series dependencies, but suffer from the well-known curse of dimensionality when it comes to the cross-sectional dependencies. There will be more perspectives on these points in the sections that cover each of the approaches.

Synthetic data generation has received a lot of attention in recent years with, for example, image and text generation. Contrary to these applications, investment time series have a lot less structure. For example, when we are generating an image, there are certain limitations on the pixels and the image resolution. When we are generating text, there are certain grammatical rules and a finite number of words in the dictionary that we can choose from. We would not worry about pixels suddenly behaving in structurally new ways or the rules of the language fundamentally changing in important ways without us knowing. Even with the additional structure, image and text generation problems are by no means trivial, which is evident by the large industries that exist around solving these problems.

Recently, video and music generation has also received increased attention. Investment time series probably have more in common with these applications, while still being distinct due to the constant structural breaks. Another challenge with investment data is that we are trying to estimate both the cross-sectional distribution as well as the time series dependencies using just one historical realization for each time series. This creates challenges both when it comes to estimation and evaluation. For example, when evaluating our distribution forecast for the next time step, we only have one realization to assess the quality of our distributional forecast. The potentially high-dimensional nature of our joint distributional forecasts can additionally complicate these aspects.

By now, it should be clear to readers that the market simulation problem is immensely challenging. However, this does not mean that we cannot do better than doing nothing and relying solely on qualitative considerations when building our portfolios. We also want to avoid using methods and making assumptions that are in obvious disagreement with observed historical data. It is for example very dangerous to make the assumption that investment return distributions do not have fat tails, are not skewed in complex ways, and that the dependencies are only linear and cross-sectionally constant. These are the assumptions that we would make if we restrict ourselves to just focusing on a mean vector and a covariance matrix, which is why we avoid doing that in the sections below.

3.2.1 Time- and State-Dependent Resampling

Imagine for a second that we were able to find stationary transformations for real-world market data such that the transformed time series presented in Section 3.1 were independently and identically distributed (iid) over time. Making the additional idealized assumption that there are no structural breaks such that the historical data distribution is representative of the future, we could simply re-sample these stationary transformations and combine them in new ways to generate new, previously unseen, paths. This would be equivalent to a high-dimensional resampling of observations of ε_t from the $AR(1)$ from Section 3.1. The elegant feature of this approach would also be that the cross-sectional dependencies among the time series can be arbitrary complex. We would not need to estimate them with parametric models but still be able to generate new paths that preserve these arbitrarily complex

dependencies.

As already discussed, it is unlikely that we will be able to find stationary transformations such that the transformed data is iid across time without overfitting to the particular historical sample. This realization has lead to the development of bootstrap methods that attempt to capture the remaining time series dependencies. Most popularly, the block bootstrap that samples several time periods at a time. For more details on resampling methods with dependent data, see Lahiri (2003). The structural breaks are not something that we have control of. The strategy here will be to have simulation methods that are reactive enough to capture them within reasonable time or use a discretionary Entropy Pooling adjustment as presented in Chapter 5.

This section introduces a new time series bootstrapping method, which is presented for the first time in this book and is coined Fully Flexible Resampling. It builds on a clever use of the Entropy Pooling method, introduced in Section 1.1 and thoroughly presented in Chapter 5. More specifically, it is a generalization of the approach introduced by Meucci (2013), where we include an implicit Markov chain, see Section 1.2, that allows us to simulate new paths for horizons $H \geq 1$ and potentially condition on a different initial state than the current. The introduction of this Markov chain gives us the flexibility to capture some of the time series dependencies that are remaining after our stationary transformations from Section 3.1.

The starting point of our simulation is the stationary transformations $ST \in \mathbb{R}^{\tilde{T}\times\tilde{N}}$, having $\tilde{T}$ observations. When we resample these observations, the number of time series $\tilde{N}$ is not important. We can have markets that are arbitrarily high-dimensional. If the historical data is a good representation of the cross-sectional dependencies that we can expect in the future, we will be able to capture these with our resampling approach no matter how complex they are. In fact, when we perform our resampled simulation, we can just sample the historical time series indices and store them as our simulations. Based on these indices, we can sample the stationary transformations that we need for a particular purpose. This is also a useful feature in the case where we work with very high-dimensional markets, where we are perhaps not able to store all of our historical data in memory at once.

By default, the historical observations carry equal sample weight $p_t = \frac{1}{\tilde{T}}$ for $t = 1, 2, \ldots, \tilde{T}$. However, there is nothing restricting us from changing these sample weights, besides the natural constraint that $\sum_t^{\tilde{T}} p_t = 1$, and $p_t \in [0, 1]$ for all t. As an example, we can assign equal probability to all scenarios in a subset of the historical observations or implement some decay in the sample weight that we assign to historical scenarios, giving a higher weight to recent observations. A common suggesting for the latter is exponential decay with half life parameter τ, i.e.,

$$p_t^{exp} \propto e^{-\frac{\ln 2}{\tau}\left(\tilde{T}-t\right)}, \quad t \in \left\{1, 2, \ldots \tilde{T}\right\}, \tag{3.2.1}$$

where the "proportional to" symbol $\propto$ is used to indicate that the scenario probabilities must be normalized such that $\sum_t^{\tilde{T}} p_t^{exp} = 1$. We will refer to the exponentially decaying probabilities (3.2.1) as time-conditioning.

To introduce state-conditioning, think about the VIX index introduced in Chapter 2. We could think of this as a state variable for the market, with periods of low, medium, and high implied volatility. It is indeed commonly thought of as such by investment practitioners and sometimes called the "fear index". If we define some values VIX_{high} and VIX_{low} for which we consider the VIX index to be,

respectively, high or low, we can assign equal probability to all historical scenarios having a VIX value equal to or higher than VIX_{high} and define this as an "elevated implied volatility" state. We could then define additional two states with VIX below VIX_{low}, and VIX between VIX_{low} and VIX_{high}. These are elementary examples of state-conditioning.

The most elementary way of introducing state-dependence into resampling is then to estimate a state transition probability matrix as presented in Section 1.2. We can estimate these transition probabilities by simply counting how many times the VIX index has historically transitioned from one state to another and then normalize, such that each row of the transition matrix sums to one. The resampling would then happen according to the following procedure: use the current state to sample the next state according to the transition probability matrix $\mathcal{T}$ and then sample a historical joint observation of the stationary transformations conditional on the new state. This procedure would probably help us capture some of the time series dependencies that remain in the stationary transformations, certainly more than an iid bootstrap would, but it contains no time-conditioning like in (3.2.1), and it uses just one state variable.

The Fully Flexible Resampling method introduced in this section allows us to elegantly combine time- and state-conditioning including several arbitrarily complex state variables. This section focuses on presenting the method with a code example, while a more detailed mathematical analysis of the method's properties will be presented in a separate forthcoming article in order to keep the focus on the essence in this book. The method essentially uses Entropy Pooling in the way proposed by Meucci (2013), while the innovation here is to generalize the approach to H-step-ahead simulations instead of being restricted just the next period as in the original article.

Letting z_t denote the value of a state variable such as the VIX index, we start by formally defining purely state-conditioned scenario probabilities as

$$p_t^{crisp} \propto \begin{cases} 1 & \text{if } z_t \in R\left(z^\star\right), \\ 0 & \text{otherwise.} \end{cases} \tag{3.2.2}$$

In the above, $R\left(z^\star\right)$ is a symmetric range around the value $z^\star$ in the sense that

$$\sum_{\{t|z_t\in[\underline{z},z^\star]\}} p_t = \frac{\alpha}{2} = \sum_{\{t|z_t\in[z^\star,\bar{z}]\}} p_t,$$

where p_t, $t = 1, 2, \ldots, \tilde{T}$, are the uniform empirical scenario probabilities and $\alpha \in [0, 1]$ is some probability.

This section will proceed to define the Fully Flexible Resampling method for one state variable, such as the VIX index, while Section 3.2.1.1 presents how to condition on multiple state variables. The fundamental resampling approach is independent of the number of state variables. Additional state variables simply introduce more states and hence more state probability vectors for us to sample from.

For one state variable, let us imagine that we partition the historical realizations into J ranges such

as (3.2.2) by defining $J-1$ partitioning values v_j such that $v_j < v_{j+1}$ for $j = 1, 2, \dots J-1$, i.e.,

$$R\left(z_j^\star\right) = \begin{cases} z_t \leq v_j & \text{for } j = 1, \\ v_{j-1} < z_t \leq v_j & \text{for } j = 2, \dots, J-1, \\ v_{j-1} < z_t & \text{for } j = J. \end{cases}$$

The partitioning values v_j could, for example, be determined by some desired percentiles such as 25% and 75%.

The next step is to compute J scenario probability vectors q_j that each combine the time-conditioning from (3.2.1) with the state-conditioning from (3.2.2) for the ranges $R\left(z_j^\star\right)$. We do this using Entropy Pooling with p_t^{exp} as the prior together with the views

$$\begin{aligned} \sum_{t=1}^{\tilde{T}} x_t z_t &= \mu_j, \\ \sum_{t=1}^{\tilde{T}} x_t z_t^2 &\leq \mu_j^2 + \sigma_j^2, \end{aligned}$$

where

$$\begin{aligned} \mu_j &= \sum_{t \in \left\{t | z_t \in R\left(z_j^\star\right)\right\}} p_t^{crisp} z_t, \\ \sigma_j^2 &= \sum_{t \in \left\{t | z_t \in R\left(z_j^\star\right)\right\}} p_t^{crisp} z_t^2 - \mu_j^2. \end{aligned}$$

Meucci (2013) shows this approach corresponds to a mixture of an exponential decay prior (3.2.1) and a smoother kernel version of the crisp probabilities (3.2.2) with optimal bandwidth and center. More technical details can be found in Meucci (2013) and the forthcoming article about the Fully Flexible Resampling method. See also Chapter 5 for a deeper presentation of Entropy Pooling than in the introductory Section 1.1.

With an initial state $j = j_0$ and the probability vectors q_j at hand, we can generate S paths using the Fully Flexible Resampling method with following procedure:

1. Sample a historical scenario $t \in \left\{1, 2, \dots, \tilde{T}\right\}$ according to the scenario probabilities q_j.
2. Update the state j, so it corresponds to the state of the historical scenario t from 1.
3. Repeat 1. and 2. H times.

At first, it might be unclear how the above procedure results in a Markov chain as introduced in Section 1.2, because we do not explicitly sample a state before resampling the historical scenarios. After the initial state, the state sampling is implicit through the scenario probability vectors q_j. Based on these, we can calculate a state transition probability matrix $\mathcal{T}$ if we want. The case study in this section illustrates how you can apply the method to the stationary transformations from Section 3.1, using

the one month at-the-money forward (ATMF) implied volatility as the state variable. You can find all details in the accompanying code to this section.

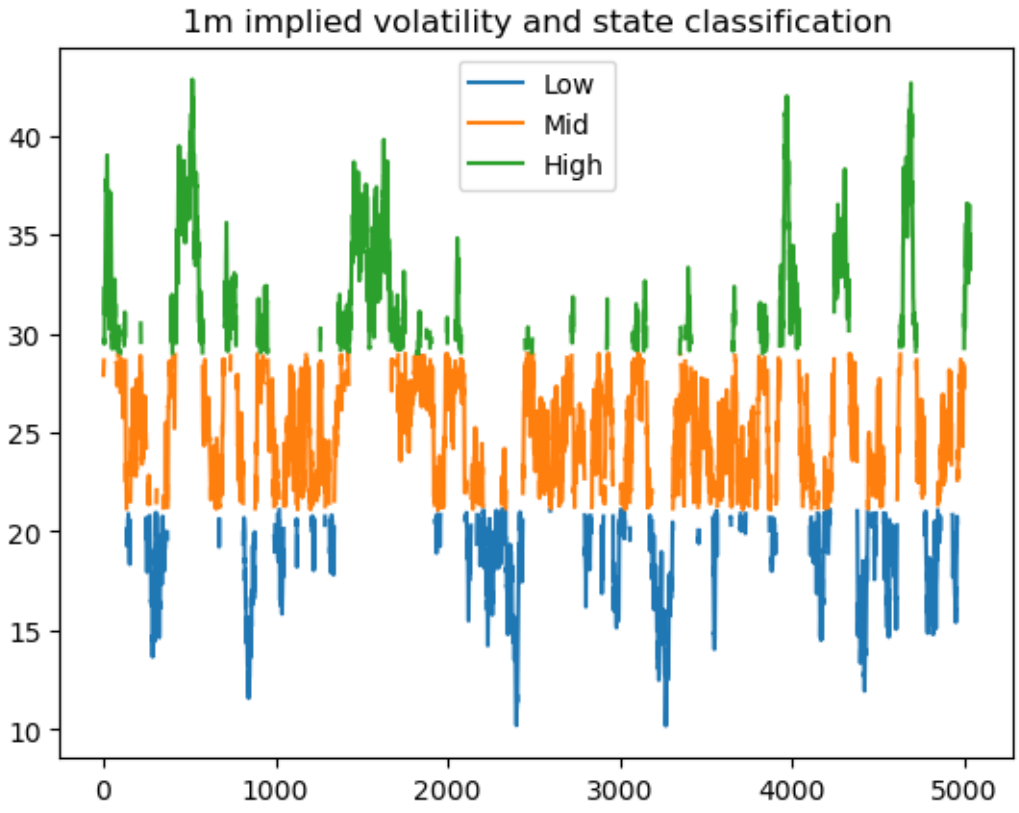

Figure 3.2.1: 1m ATMF implied volatility and state classification.

Figure 3.2.1 shows the historical realizations of the 1m ATMF implied volatility together with its state classification based on 25% and 75% percentiles. Based on the state classification, we can calculate μ_j and σ_j that we will use as Entropy Pooling view values to compute q_j against the exponential decay prior (3.2.1), which Figure 3.2.2 shows with a half life parameter $\tau = \tilde{T}/2$.

Figure 3.2.3 shows the Entropy Pooling posterior probability vectors q_j for each of the state classifications from Figure 3.2.1. The exponential decay for these probability vectors is evident from the figure. It is also clear that we have more scenarios in the mid implied volatility state with each scenario having a lower average probability than for the high or low implied volatility states. The scenario probabilities of the high and low implied volatility states are naturally concentrated on fewer historical observations.

We note that in this case study we used a state variable which is fairly straightforward to interpret, but we are not limited to state variables with an economic interpretation, although this will probably be preferred by other stakeholders. We can use arbitrarily complex state variable as long as their values can be partitioned in a meaningful way. In this case, we have an intuitive idea of what it means that the implied volatility is higher or lower, but that might also hold for more complex state variables.

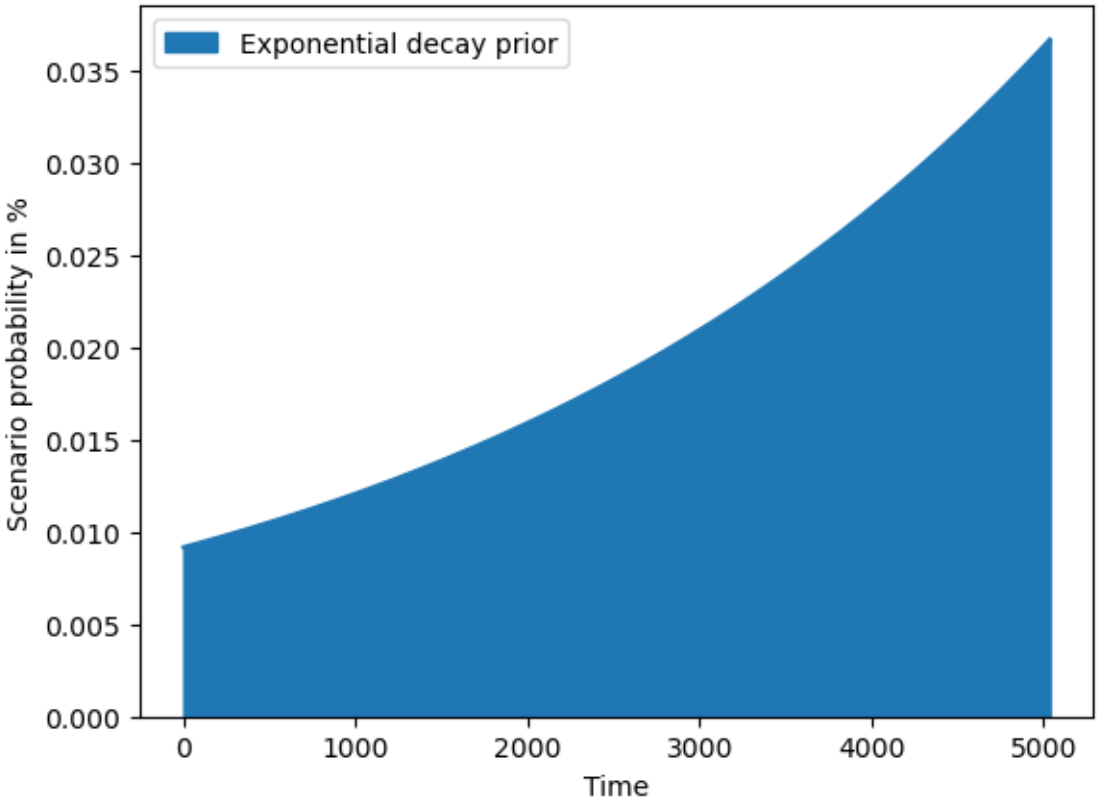

Figure 3.2.2: Exponential decay prior.

We use the probability vectors from Figure 3.2.3 to perform a resampled simulation $H = 21$ days into the future, using the Fully Flexible Resampling procedure from this section. Since the stationary transformations for the equity index are just the log returns, it is easy for us to transform these into the returns that we would use for investment and risk analysis. Hence, Figure 3.2.4 shows the one month return distribution for the equity index conditional on the initial state. The reverse transformation and simulation for implied volatilities, interest rates, and credit spreads will be presented in Section 3.3.

Based on the scenario probability vectors q_j, which are illustrated in Figure 3.2.3, we can compute the implicit state transition probability matrix $\mathcal{T}$ of the resampling Markov chain. For this case study, it is given by

$$\mathcal{T} = \begin{pmatrix} 0.902 & 0.098 & 0.000 \\ 0.043 & 0.923 & 0.035 \\ 0.000 & 0.110 & 0.890 \end{pmatrix}.$$

We note again that we do not explicitly sample the state before resampling a historical observation. After the initial state conditioning, the state and historical scenario is sampled jointly.

The 1m implied volatility state variable in this case study is probably not sufficient to capture the time series dependencies in the data. Hence, Section 3.2.1.1 presents how we can condition on multiple state variables, but the fundamental resampling procedure remains the same.

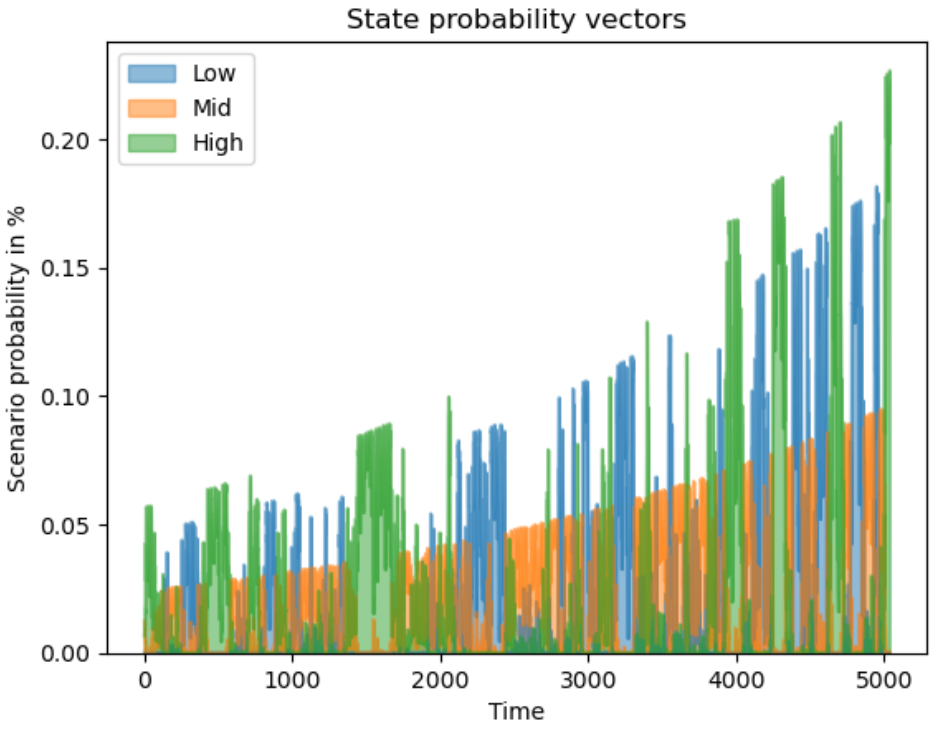

Figure 3.2.3: Fully Flexible Resampling probability vectors q_j.

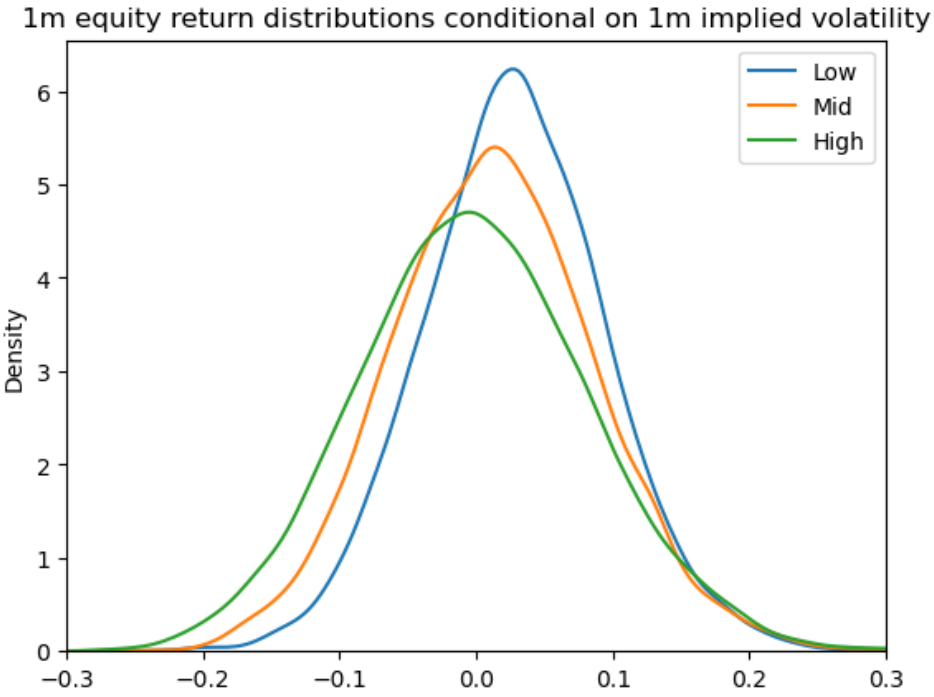

Figure 3.2.4: One month equity return distributions conditional on initial state.

3.2.1.1 Multiple State Variables

Even if we have multiple state variables, the Fully Flexible Resampling procedure is the same as in Section 3.2.1. We just need to do a bit more work when computing the historical scenario probability vectors q_j, $j = 1, 2, \ldots, J$, and define partitioning values for the M state variables, i.e., we add an $m \in \{1, 2, \ldots, M\}$ subscript to the ranges

$$R\left(z^{\star}_{i,m}\right) = \begin{cases} z_{t,m} \leq v_{i,m} & \text{for } i = 1, \\ v_{i-1,m} < z_{t,m} \leq v_{i,m} & \text{for } i = 2, \ldots, I_m - 1, \\ v_{i-1,m} < z_{t,m} & \text{for } i = I_m. \end{cases}$$

The number of state probability vectors q_j then becomes $J = \prod_{m=1}^{M} I_m$.

The first step is the same as in Section 3.2.1, but it must be repeated for all M state variables, i.e., we must compute posterior probability vectors $q_{i,m}$ using Entropy Pooling with p_t^{exp} as the prior and the views

$$\begin{aligned} \sum_{t=1}^{\tilde{T}} x_t z_{t,m} &= \mu_{i,m}, \\ \sum_{t=1}^{\tilde{T}} x_t z_{t,m}^2 &\leq \mu_{i,m}^2 + \sigma_{i,m}^2, \end{aligned}$$

where

$$\begin{aligned} \mu_{i,m} &= \sum_{t \in \left\{t | z_t \in R\left(z^{\star}_{i,m}\right)\right\}} p_t^{crisp} z_{t,m}, \\ \sigma_{i,m}^2 &= \sum_{t \in \left\{t | z_t \in R\left(z^{\star}_{i,m}\right)\right\}} p_t^{crisp} z_{t,m}^2 - \mu_{i,m}^2. \end{aligned}$$

Note that the above results in $\sum_{m=1}^{M} I_m$ probability vectors $q_{i,m}$ for the M individual state variables. These probability vectors need to be combined through a proper weighting, so the state conditioning happens jointly over all M state variables.

The method used for combining the individual probability vectors $q_{i,m}$ to achieve joint state conditioning is the one suggested by Meucci (2013). However, in this book it is recommended to use linear mixing as in Section 5.3, because this usually results in a higher effective number of scenarios (ENS), given in equation (5.1.2), for the final state probability vectors q_j, which is arguably a desirable feature for a resampling method where we want to use the available data as effectively as possible. See Section 5.1 for a definition and explanation of ENS.

With multiple state variables, we have I_m probability vectors $q_{1,m}, q_{2,m}, \ldots, q_{I_m,m}$ for each state variable $m = 1, 2, \ldots, M$. We must combine these over the Cartesian product $\mathcal{I} = \mathcal{I}_1 \times \mathcal{I}_2 \times \cdots \times \mathcal{I}_M$ with $\mathcal{I}_m = \{1, 2, \ldots, I_m\}$ into J probability vectors q_j. For each m we define the mapping $f_m : \{1, 2, \ldots, J\} \to \mathcal{I}_m$ such that $f_m(j) = i_m$ gives us the index i_m of state variable m for scenario j. We must then determine weights $w_{i_m,m}$ for the vectors $q_{i_m,m}$ that satisfy the constraints $w_{i_m,m} \geq 0$ and

$\sum_{m=1}^{M} w_{i_m,m} = 1$. The final state probability vector q_j is then given by

$$q_j = \sum_{m=1}^{M} w_{i_m,m} q_{i_m,m} \quad \text{for } j \in \{1, 2, \ldots, J\}.$$

With an overall understanding of what we are trying to achieve, we continue to the procedure of computing $w_{i_m,m}$. As already argued, vectors $q_{i_m,m}$ with a higher effective number of scenarios (ENS) are preferable from a data efficiency perspective. Hence, vectors $q_{i_m,m}$ with a higher ENS should be given a higher weight $w_{i_m,m}$. The other consideration comes from giving a lower weight to redundant state variables in the sense that they contain very little extra information compared to the other state variables. Such state variables should be given a lower weight $w_{i_m,m}$. Hence, if we let $D_{i_m,m}$ be a soon to be defined diversity indicator, we determine the weights $w_{i_m,m}$ using the following formula

$$w_{i_m,m} = \frac{ENS_{i_m,m} D_{i_m,m}}{\sum_{m=1}^{M} ENS_{i_m,m} D_{i_m,m}}.$$

The effective number of scenarios is defined in equation (5.1.2), so we focus on defining $D_{i_m,m}$ using the notation from this book. For any pair of individual probability vectors $(q_{i_m,m}, q_{i_{\tilde{m}},\tilde{m}})$, we first define the Bhattacharyya coefficient as $b_{m,\tilde{m}} = \sum_t^{\tilde{T}} \left(q_{t,i_m,m} q_{t,i_{\tilde{m}},\tilde{m}}\right)^{1/2}$, compute the Hellinger distance $d_{m,\tilde{m}} = \sqrt{1 - b_{m,\tilde{m}}}$, and finally compute the diversity score as

$$D_{i_m,m} = \frac{1}{M-1} \sum_{\tilde{m} \neq m} d_{m,\tilde{m}}.$$

The justification for the diversity score is given by Meucci (2013). While the author finds that this leads to meaningful weights $w_{i_m,m}$, readers are encouraged to explore other meaningful alternatives based on the same fundamental objectives.

The case study in this section uses the same data as Section 3.2.1 and adds the slope of the interest rate curve, defined as the difference between the 1y and 10y zero-coupon bond yield, as an additional state variable. For the slope, we define just one value $v_{1,2} = 0$, i.e., the states are whether the slope of the interest rate curve is positive or inverted. Hence, $I_1 = 3$ for the 1m implied volatility state variable as in the previous section, while $I_2 = 2$ for the slope state variable. This gives us a total of $J = 3 \cdot 2 = 6$ states with $M = 2$ vectors in each state that need to be combined into q_j based on $ENS_{i_m,m}$ and $D_{i_m,m}$.

This section shows the results of the analysis similar to Section 3.2.1, but it is left as an exercise for readers to test their understanding by replicating the graphs below. We start with Figure 3.2.5 and Figure 3.2.6 that illustrate the state variables together with their state classifications. We note that Figure 3.2.5 shows the same 1m implied volatility time series, but it is now classified into six states instead of three. Finally, Figure 3.2.7 shows the six Fully Flexible Resampling probability vectors q_j that we would use for resampled simulation.

From Figure 3.2.7, we note that there are few states where the interest rate curve is inverted. Hence, the state conditioning becomes significant on the historical scenarios where we have an inverted interest rate curve, clearly distorting the exponential decay prior.

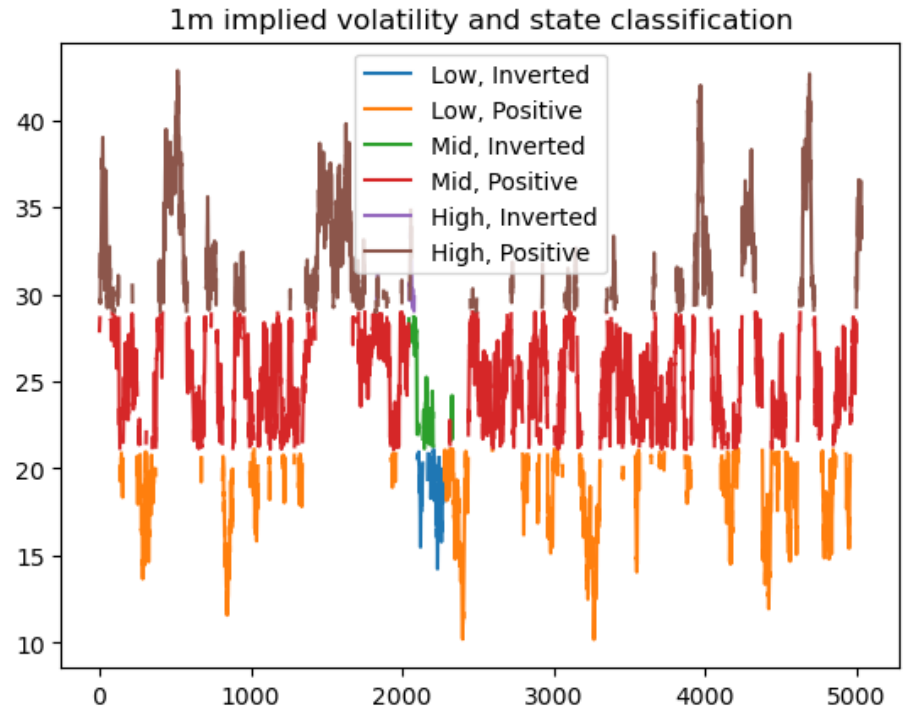

Figure 3.2.5: 1m ATMF implied volatility and state classification.

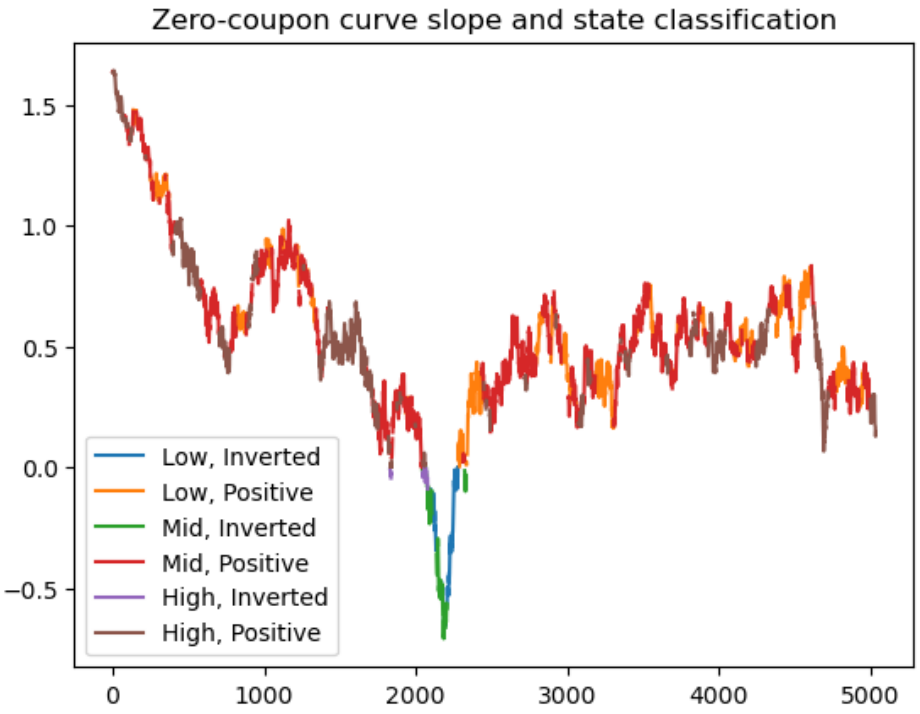

Figure 3.2.6: Slope of interest rate curve and state classification.

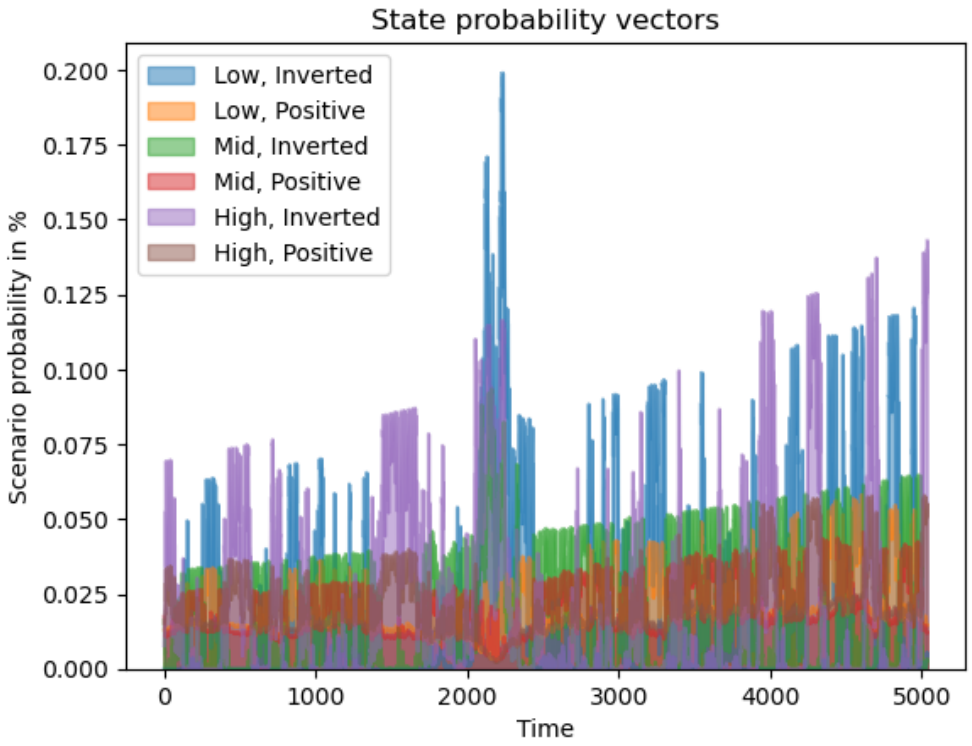

Figure 3.2.7: Fully Flexible Resampling probability vectors q_j.

3.2.2 Generative Machine Learning

This section presents generative modeling of investment markets based on new approaches such as variational autoencoders (VAEs) and generative adversarial nets (GANs). Besides these methods using other mathematical concepts than the Fully Flexible Resampling method introduced in Section 3.2.1, the objective remains the same. Resampling methods have the advantage that they are capable of capturing arbitrarily complex cross-sectional dependencies, while they require time- and state-conditioning to capture the time series dependencies that remain in the stationary transformations from Section 3.1. On the other hand, the generative machine learning methods are very capable of capturing time series dependencies when properly implemented, but they might suffer from the usual curse of dimensionality problem as the cross-sectional dimension increases.

VAEs and GANs in their pure forms are arguably harder to interpret than the Fully Flexible Resampling method, which is likely a reason for cautiousness among investment practitioners. However, the field is still in its infancy, so we might come up with good ideas for how to use these methods in combination with some interpretative overlay. In some cases, the performance of generative machine learning methods might also simply be so good that we decide to accept the lack of interpretability and simply think about it as statistical pattern recognition. The author's hypothesis is that in the same way that we have large language models (LLMs), we will have large market models (LMMs) in the future. We are currently not quite there, but we will see in the subsections below that we can already use VAEs to solve relevant problems in a better way than conventional alternatives.

3.2.2.1 Variational Autoencoders (VAEs)

To understand variational autoencoders (VAEs), it is helpful to understand autoencoders (AEs) first. AEs are essentially a dimensionality reduction technique similar to the more well-known principal components analysis (PCA). The main differences are that PCA transforms the raw data in a linear way, while autoencoders can also perform nonlinear transformations. Another difference is that PCA orders factors from the ones that explain the most of the variance in the data to the least. AEs do not do that.

As a reminder of how principal components work, let us consider the historical data $D \in \mathbb{R}^{T \times N}$ and compute a demeaned version across the time series dimension $\bar{D} \in \mathbb{R}^{T \times N}$. The principal components are then defined as

$$F = \bar{D}W \in \mathbb{R}^{T \times N},$$

where $W \in \mathbb{R}^{N \times N}$ is a transformation matrix which ensures that all columns of F are orthogonal to each other, i.e., have zero correlation. Furthermore, the principal component factors are ordered such that F_i explains a larger proportion of the variance in the data than F_j when $i < j$, with F_i and F_j denoting some columns i and j in the matrix F. There is a lot of literature on PCA and software that solves the problem of finding the matrix W, see for example James et al. (2023), so we will not spend time on it here and instead focus on relating PCA to AEs.

If we want to reconstruct the raw demeaned data, we can naturally do it through

$$FW^{-1} = \bar{D} \in \mathbb{R}^{T \times N}.$$

Hence, we can think of the matrix W as a linear encoding of the raw data $\bar{D}$ into the principal component factors F, while W^{-1} is a linear decoding of the principals components into the raw data $\bar{D}$. The dimensionality reduction comes from the fact that we can decide to use only some of the principal components to analyze and reconstruct our data, see James et al. (2023).

When it comes to AEs, the encoding into factors is performed through a general function

$$f(D) = F \in \mathbb{R}^{T \times \tilde{N}},$$

where $\tilde{N} < N$ typically, and the function $f(D)$ is a neural network. The same is true for the subsequent decoding function

$$g(F) = \tilde{D}.$$

Note that $\tilde{D}$ will contain a reconstruction loss when $\tilde{N} < N$, while $\tilde{D} = D$ when $\tilde{N} \geq N$. This is similar to principal components when we do not use all of them to reconstruct the data.

The variational autoencoder was introduced by Kingma and Welling (2013). Instead of reducing the data to a deterministic representation, VAEs typically parameterize the encoder and decoder such that the factors are fitted to a normal distribution with a diagonal covariance matrix, i.e.,

$$f(D) = F \sim \mathcal{N}\left(\mu, \operatorname{diag}\left(\sigma^2\right)\right),$$

where $\mu \in \mathbb{R}^{\tilde{N}}$ and $\sigma^2 \in \mathbb{R}_+^{\tilde{N}}$. The details of how this is done using what is called the "reparametrization

trick" and a mean squared error objective combined with a Kullback–Leibler divergence term can be found in Kingma and Welling (2019).

Once we have estimated the relevant VAE parameters, we no longer need the encoder $f(D)$ for market simulation. We can just sample from the normal distribution with mean μ and covariance diag (σ^2) and subsequently input these samples into the decoder $g(F)$, which will generate new synthetic data samples that have properties similar to the historical data D, assuming that we have specified and estimated our VAE model properly.

VAEs are usually straightforward to train, but they are very hard to implement in a way that properly handles statefulness in training, evaluation, and simulation given the current deep learning technology. This leads to a common misunderstanding that VAEs are not capable of handling data that is correlated over time. Some people also struggle to understand how the iid sampling from $\mathcal{N}(\mu, \text{diag}(\sigma^2))$ can produce correlated synthetic samples, but it is in the exact same way as the $AR(1)$ process presented in Section 3.1, where the noise term ε_t is also iid Gaussian, while the actual $AR(1)$ process has obvious autocorrelation. Properly implemented and stateful neural network layers are in fact very capable of capturing time series dependencies.

Compared to the Fully Flexible Resampling method introduced in Section 3.2.1, VAEs in their raw form are naturally harder to interpret. Hence, investment managers have so far been reluctant to use them for market simulation, while it might be possible to come up with architectures that introduce an interpretative or causal overlay. One practical area where VAEs have been used is data imputation, for example, if we have market data from various countries or regions where the holidays are different. Previously, such imputations were typically done with regression models, PCA, or more simple imputation methods such as replacement with the mean or last observation carried forward (LOCF).

Below is a case study where we compare the performance of LOCF, PCA, and a VAE to impute missing data in the multi-asset simulated data introduced in Section 3.1. The case study randomly removes 10% of the data and imputes it using the three different methods. It uses proprietary implementations of these methods, while the VAE algorithm is spelled out in McCoy, Kroon, and Auret (2018). The VAE has a very simple architecture with one layer of 16 simple RNN nodes for both the encoder and decoder. It is perhaps possible to get even better performance by improving the architecture, but this example shows that properly implemented VAEs, which correctly handle statefulness, are capable of producing good results with narrow and simple architectures. The challenge is to get these implementations right for the tabular time series data that we work with.

The mean squared errors for LOCF, PCA, and VAE are, respectively, 0.916, 0.601, and 0.379, so we see that the VAE approach produces significantly better data imputations than the two other methods. Figure 3.2.8 illustrates the last year of imputed and real data for the equity index, while Figure 3.2.9 illustrates the deviations from the true value over the entire period for the indices that are missing. From Figure 3.2.9, we can visually get a sense of the gain from using VAEs. Note that the magnitude of the differences in Figure 3.2.9 increases simply due to the equity index increasing.

Since imputation methods like PCA and regressions are typically applied in a statistical pattern recognition way, the lack of interpretative overlay for VAEs seems to be acceptable to investment managers. For the actual market simulation, methods like the Fully Flexible Resampling from Section

3.2.1 seems to be preferred, at least until generative machine learning methods can clearly prove that they produce much better future market simulations.

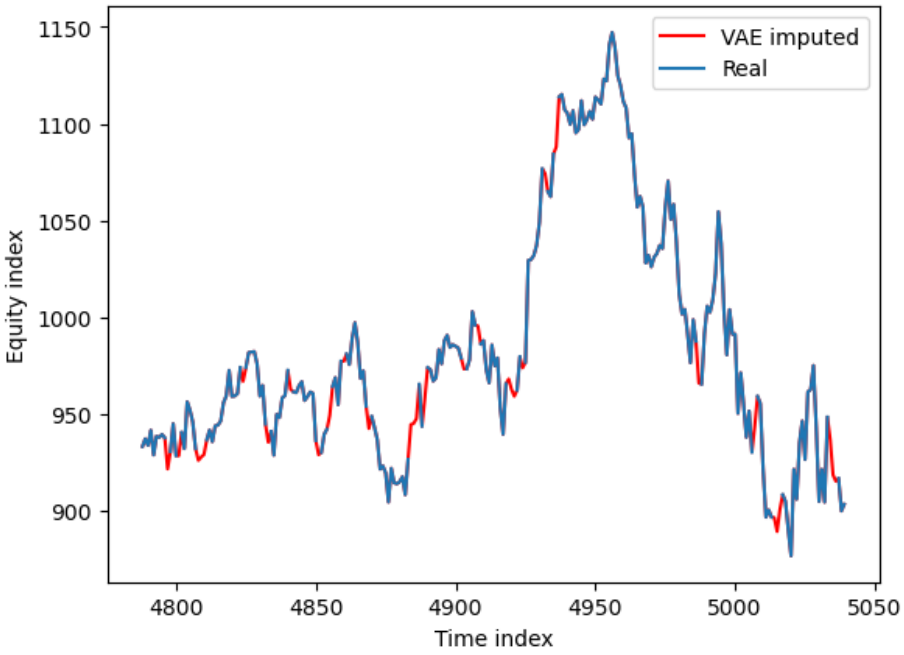

Figure 3.2.8: Equity index series and imputed values.

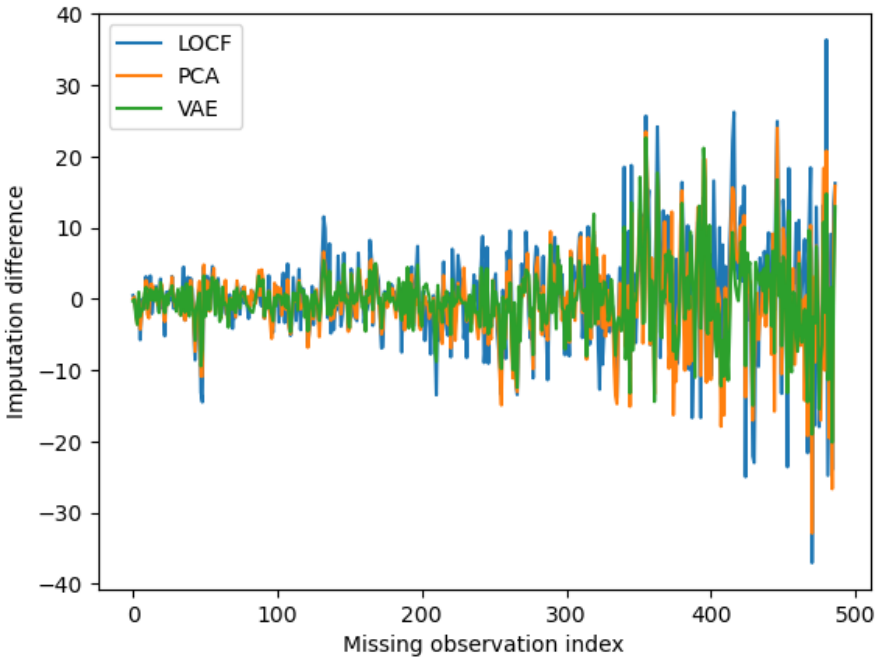

Figure 3.2.9: Imputation difference for the equity index.

3.2.2.2 Generative Adversarial Networks (GANs)

Generative Adversarial Networks (GANs) were initially introduced by Goodfellow et al. (2014) and are fundamentally different from VAEs, while the synthetic data generation objective is the same. In a standard GAN model, there are two neural networks competing with each other. The first one is a generator $\mathcal{G}(z)$, which typically takes a normally distributed noise vector $z \sim \mathcal{N}(0, \text{diag}(1))$ as input, while the second is a discriminator $\mathcal{D}(\mathcal{G}(z), D)$, which takes both the original data D and the simulated data $\mathcal{G}(z)$ as input.

The idea is to train these two networks in a way where the generator becomes better at generating synthetic samples $\mathcal{G}(z)$ that the discriminator cannot distinguish from the original data D. While this sounds natural and is the essence of what we are trying to do, several practical issues can occur when we train GAN models, for example, mode collapse. Mode collapse is a situation where the generator learns to generate excellent data points in a specific area of the distribution and continues to generate these samples without exploring the entire distribution of the historical data. A method to alleviate issues with mode collapse is to use minibatch discrimination layers, see Salimans et al. (2016).

Similar to VAEs, tabular time series introduce several additional complexities. Hence, it is crucial that statefulness is handled correctly in GAN training, evaluation, and simulation. There have also been specific time series GAN architectures proposed, for example, by Yoon, Jarrett, and Van der Schaar (2019) that essentially combine the VAE and GAN architectures and generate samples in the latent VAE space, such that we are no longer constrained to it being iid normal as in Section 3.2.2.1. In summary, it is still very early days for generative machine learning methods applied to investment time series, and there is a significant exploration barrier due to the implementation complexities. Naive copy/pasting of code from other applications usually does not work.

3.2.3 Perspectives on No-Arbitrage and Stochastic Differential Equations

Stochastic differential equations (SDEs) are arguably one of the most used modeling approaches in quantitative finance, which used to be all about derivatives pricing. While SDEs are useful when it comes to guaranteeing no arbitrage for derivative prices, few people believe that they are a good representation of real-world market behavior. The most well-known SDE in quantitative finance is probably the geometric Brownian motion used in the Black and Scholes (1973) option pricing model

$$dX_t = \mu X_t dt + \sigma X_t dW_t,$$

where $W_t \sim \mathcal{N}(0, t)$ is called a Wiener process or Brownian motion.

The geometric Brownian motion results in price and return distributions that are log-normal. This is arguably a bit better than the normal distribution, which assigns some probability to losses below 100% for plain vanilla cash instruments like stock and bonds, but it is still not sufficient to capture the complex distributions of real-world markets that are presented in Chapter 2. It is also highly unrealistic that the drift μ and volatility σ are constant through time and thus deterministic.

Several extensions to the Black-Scholes model have been proposed, including local volatility and stochastic volatility models. While these introduce additional real-world elements, they are still quite restrictive when it comes to capturing the complex distributions and dependencies of real-world invest-

ment markets. Perhaps even more importantly, they are very restrictive when it comes to capturing the potentially very high-dimensional dependencies of investment markets. For example, if we generalize the geometric Brownian motion to a stochastic volatility model formulation

$$dX_t = \mu X_t dt + \sqrt{v_t} X_t dW_t,$$
$$dV_t = \alpha_t dt + \beta_t dB_t,$$

the additional dependencies between these processes are usually introduced through a correlation between the Brownian motions W_t and B_t that drive the fundamental uncertainty. This is still too restrictive.

In summary, stochastic differential equations are good no-arbitrage calibrating machines that allow market makers to price derivatives on typically a single underlying such as S&P 500 and STOXX 50, but they are usually too restrictive when it comes to approximating both the marginal and joint distributions of investment markets. The author is not familiar with any successful use of SDEs for the risk modeling of high-dimensional markets for investment applications, but he has witnessed many failed attempts. The joint calibration of these models becomes increasingly complex as the dimension increases, and there can be many additional restrictions on, for example, the dynamics of interest rates.

The approaches presented in this book for market simulation are much more focused on being able to capture the complex shapes and dependencies observed in investment markets. This is usually much more important for investment managers. We also note that the issues related to arbitrage are mostly related to derivative instruments, where there must be a logical consistency between the underlying and the derivative. While this aspect is enforced in our focus on simulating risk factors, we cannot rule out that the implied volatility surfaces we simulate allow for arbitrage. These simulated arbitrage opportunities are unlikely to be tradeable in practice, so they mainly create issues if we have investment algorithms that specifically look for them in the simulated data.

A final perspective is that if our historical market data does not contain arbitrage and our simulations models are good at capturing the features of our data, our simulations should be essentially arbitrage-free. This is indeed also what early results seem to indicate for synthetic data generation using machine learning methods as presented in Section 3.2.2. We also note that the stationary transformation preprocessing in Section 3.1 is very flexible. Hence, it might be possible to perform these transformations in a way that ensures no arbitrage for specific instruments. Providing joint no-arbitrage guarantees for very general simulations is, to the author's knowledge, very hard.

We end this section by noting that there are no issues with arbitrage for common simulations of instruments like stocks, because all dependencies, both historical and simulated, are statistical. Hence, it is quite unlikely that we randomly end up with a simulation where there is a guaranteed risk-free profit, unless we perhaps simulate the price of the same stock traded on different exchanges. If we do something like that, we must give the simulations extra attention.

If you are a market maker interested in understanding the joint risks of your trading books, the methods presented in this book can still be very helpful. You just have to make sure that you do not use the simulations to give prices to clients if they are not guaranteed to be arbitrage-free. An understanding of the joint risks across trading books is indeed something that can add significant value to market makers and is often lacking currently.

3.3 Computing Simulated Risk Factors

As already highlighted throughout this chapter, the fundamental idea of the simulation approach is:

1. Transform the raw historical investment time series data $D \in \mathbb{R}^{T \times N}$ into stationary data $ST \in \mathbb{R}^{\tilde{T} \times \tilde{N}}$.
2. Generate S future paths $\tilde{ST} \in \mathbb{R}^{S \times \tilde{N} \times H}$ for the stationary transformations.
3. Compute simulated risk factors paths $\tilde{R} \in \mathbb{R}^{S \times \tilde{I} \times H}$ by transforming the stationary transformations to risk factors.
4. Generate the final market simulation $R = R_h \in \mathbb{R}^{S \times I}$, $h \in \{1, 2, \ldots, H\}$, by using the simulated risk factors for instrument pricing as presented in Chapter 4.

An important point, which was already made in Section 3.1, is that it should be easy for us to recover the risk factor simulations $\tilde{R} \in \mathbb{R}^{S \times \tilde{I} \times H}$ in step 3 from the simulated stationary transformations paths $\tilde{ST} \in \mathbb{R}^{S \times \tilde{N} \times H}$ from step 2. For most transformations such as differencing, this is usually not an issue. However, for fractionally differenced time series it is nontrivial to recover the original time series in general. Hence, although fractional differencing might be useful for some investment applications, it creates some challenges for common market simulation approaches as presented in this book. One can imagine that there exist other stationary transformations that make it hard to recover the original time series, either due to numerical issues or simply because it is challenging to do in full generality as with fractional differencing.

In Section 3.2.1, we already saw how to recover the quantity of interest for an equity instrument, which is usually the return. It is quite easy to compute the return based on simulated log return paths. In this section, we will continue with the recovery of the risk factors for government bonds, options, and credit bonds. These risk factors are respectively, the simulated zero-coupon curve, implied volatility surface, and credit spread. Once we have simulations for these quantities, we can compute the instrument P&L for government bonds, options, and credit bonds as presented in Chapter 4.

We let the accompanying code show how the risk factor recovery is performed for the simulated stationary transformations from Section 3.2.1 instead of writing it out mathematically. This is similar to 3.2.1, where the accompanying code shows how the stationary transformations are computed. In brief, the stationary transformations for the equity index and implied volatility surface is log changes, while the stationary transformations for government bonds and credit spreads can be interpreted as log changes in constant maturity discount factors. For government bonds, this interpretation is strictly correct, while for credit spreads it is simply the same transformation performed to the spread. One could have combined the zero-coupon curve and the credit spread for joint projection of this quantity. We leave it to the interested reader to try this out and once again underline that the suggestions from this chapter are just examples. Readers have full flexibility in relation to using other stationary transformations if they are more suitable for their particular case.

Figure 3.3.1 shows the government bond zero-coupon curve simulation at the one-month horizon resulting from the stationary transformations simulation using the Fully Flexible Resampling method from Section 3.2.1. Figure 3.3.2 shows a simulated implied volatility surface. Finally, Figure 3.3.3 shows a hypothetical credit spread with multiple maturities.

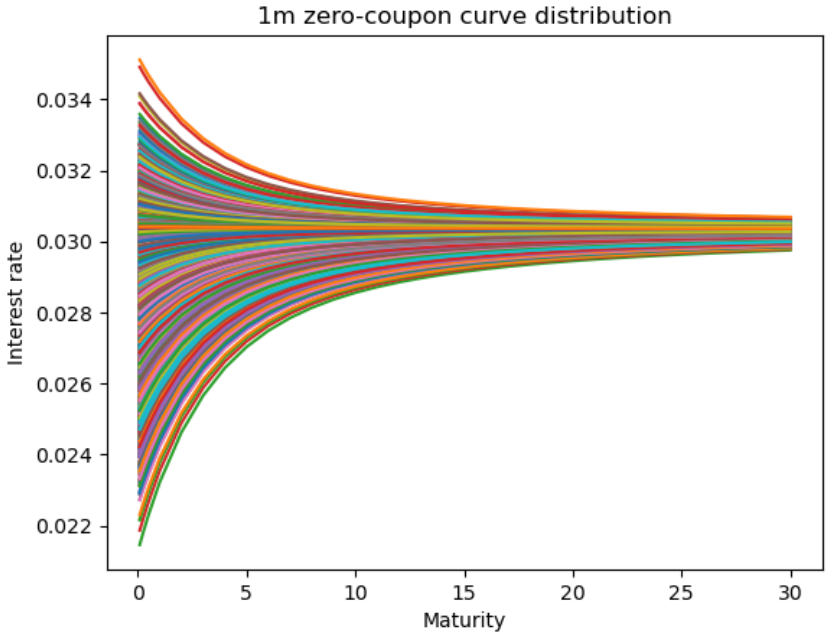

Figure 3.3.1: Simulated zero-coupon curve at the one-month horizon.

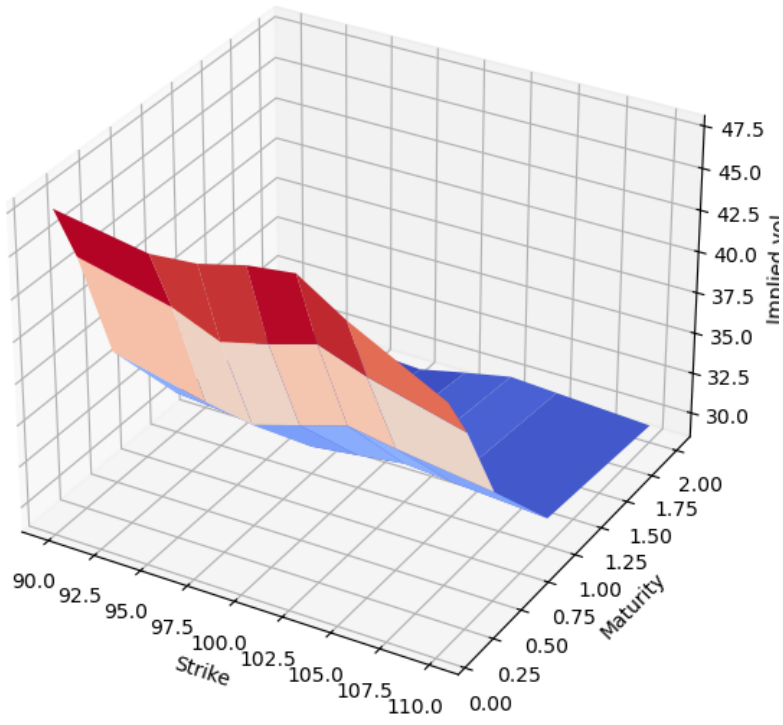

Figure 3.3.2: A simulated implied volatility surface at the one-month horizon.

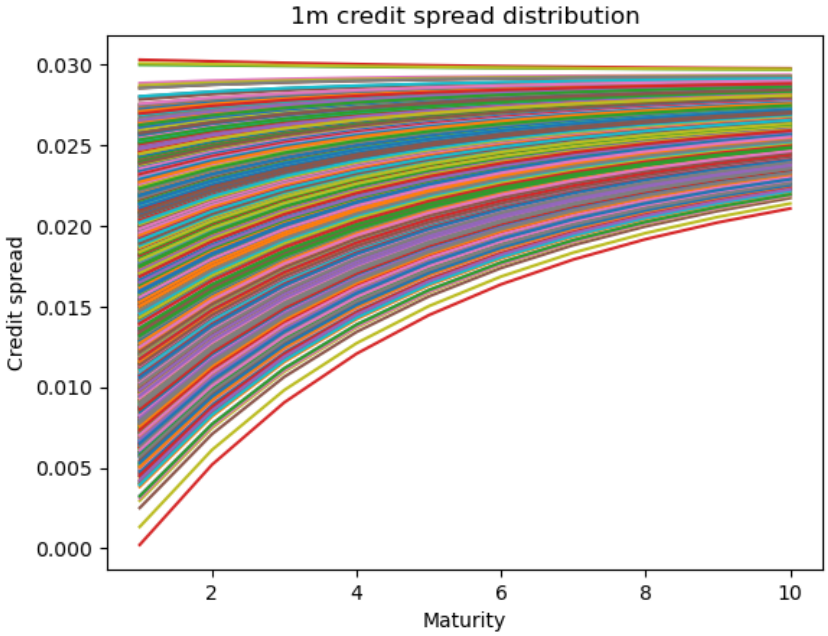

Figure 3.3.3: Simulated credit spreads at the one-month horizon.

We note that the case study in this section uses the Fully Flexible Resampling method in a quite elementary way with just one state variable given by the one-month at-the-money forward (ATMF) implied volatility. This state variable is probably insufficient to capture all of the the time series dependencies in the data, which is evident from the quite erratic simulated implied volatility surface in Figure 3.3.2. Readers can adjust the accompanying code to see other implied volatility simulations. Figure 3.3.2 shows the simulation for $s = 100$, while there are in total $S = 10,000$ simulations.

In practice, we should probably condition on more state variables as presented in Section 3.2.1.1 or use a more complex state variable that better summarizes the state of the data than the one-month ATMF implied volatility. An example of a more complex state variable could be a statistical summary of several state variables, extracted through principle components analysis (PCA) or other similar methods. There is also nothing that prevents us from combining machine learning methods as presented in Section 3.2.2 with the Fully Flexible Resampling method. Once we understand the individual foundations of the methods, we can be creative in how we combine them.

3.4 Simulation Evaluation

3.5 Better CVaR and Variance Optimization Backtesting

This section repeats the CVaR and variance optimization analysis from Section 2.6, where we performed a historical backtest. We concluded that it can be dangerous to draw generalized conclusions from historical backtests, because we have just one realization that is susceptible to overfitting. For this reason, the vast majority of historical backtests that we see probably suffer from overfitting.

Truly realistic backtests are non-trivial and would require us to, for example, properly incorporate transaction costs, risk budgets as presented in Section 6.3, and portfolio optimization parameter uncertainty as presented in Section 6.4. Software capable of handling all these aspects is well beyond the scope of the code that can be provided with this book. Hence, the backtest in this section is still not going to be at the level required for real-world applications. However, the backtesting approach is going to be exactly how it is recommended for real-world applications.

The main issue with historical backtests is that we only have one realization. If we can generate good synthetic paths that capture the main characteristics of the historical data, which the approach and methods from this chapter are designed to do, and generate as many plausible paths as we deem necessary and feasible, we can overcome this issue. While this sounds straightforward in theory, it is not easy or perfect in practice. Generating realistic simulations for high-dimensional markets is challenging, even if we have access to methods like the Fully Flexible Resampling (FFR) and good implementations of time series VAEs and GANs. We still need to properly calibrate these models using good state variables, neural network architectures, and potentially clever training tricks for GANs.

This sections and its accompanying code will illustrate the described backtesting procedure using the FFR method with VIX as the state variable, i.e., the example mentioned in Section 3.2.1. We will use the same percentiles as we did for the simulated multi-asset data in Section 3.2.1, i.e., 25% and 75% to define the states "low", "mid", and "high" VIX. Since the FFR and CVaR optimization implementations are not of production quality and therefore quite slow, we generate $S = 100$ joint simulations R for each quarter instead of using an expanding window as in Section 2.6. In practice, it is recommended to use around $S = 10,000$ joint simulations. We also generate $\tilde{S} = 100$ additional paths for the out-of-sample period that we can use to compare to the historical portfolio realization, i.e., we generate a distribution for the CVaR and variance optimized portfolio paths.

The main results are presented in Figure 3.5.1 and Figure 3.5.2 below. We first start with Figure 3.5.1, which shows the historical performance of CVaR and variance optimized portfolios. Although CVaR again comes out on top, we certainly still could have found other backtest and simulation configurations where the opposite would have been the case. The most interesting result is perhaps that we only use $S = 100$ scenarios for the 90%-CVaR optimization. This gives us just 10 scenarios below the VaR, which we use to determine the portfolio. One of the excuses for not doing the harder work and performing CVaR analysis is often that "we do not have enough data to have good estimates of the tail risks". While more data is almost always preferable, this case study shows that CVaR optimized portfolios can give meaningful results even without a high number of joint scenarios S. However, we should be careful about generalizing this conclusion. It might be driven by our particular data and case study design, i.e., there might not be a lot of diversification between the 10 equity indices neither in the tails nor on average, so the portfolios are largely determined by the indices with the lowest standalone risks.

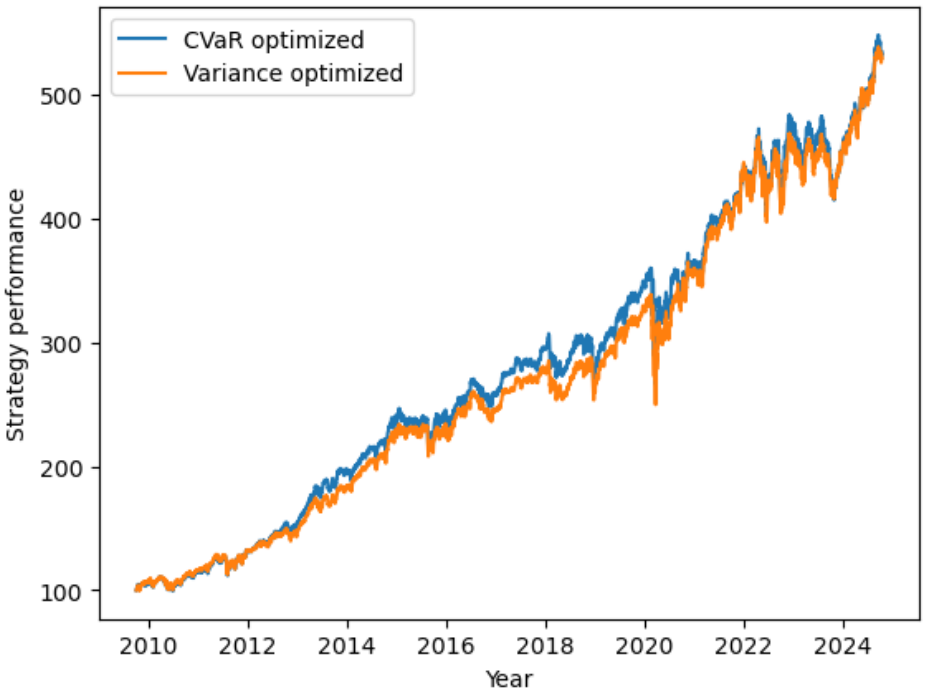

Figure 3.5.1: Historical performance of CVaR and variance strategies with simulated joint scenarios.

In Figure 3.5.2, we see some synthetic paths for the CVaR optimized portfolios. We first note that the historical realization is within this synthetic path distribution, which is good, but it seems that the synthetic paths are somewhat more optimistic than the historical realizations. Hence, just conditioning on VIX with the values that we used seems to be insufficient to capture the market state. Readers can adjust the accompanying code to introduce, for example, exponential decay in the prior probabilities as in equation (3.2.1), refine the conditioning with other values, or use multiple state variables as presented in Section 3.2.1.1.

Readers are encouraged to carefully examine the accompanying code to this section. First and foremost to make sure that they understand the suggested backtesting procedure, which is designed to overcome the issues with having just one historical realization. It is important to understand that this procedure also can give a false sense of security if our simulations R are an inaccurate representation of the historical market characteristics.

There is of course also the danger that the historical data we have is an inaccurate representation of the future, but historical backtests arguably suffer even more from this. Hence, this book generally recommends using a quantamental approach to investment management by combining the simulations based on historical data with more forward looking information using the Sequential Entropy Pooling (SeqEP) method presented in Chapter 5. We note that the methods presented in Chapter 5 can be used in a fully systematic way, although this book does not present such case studies beyond the use of the Fully Flexible Resampling method in this section. Systematic use of the methods can be seen as the base case for quantamental investors, defining the foundations of their investment processes.

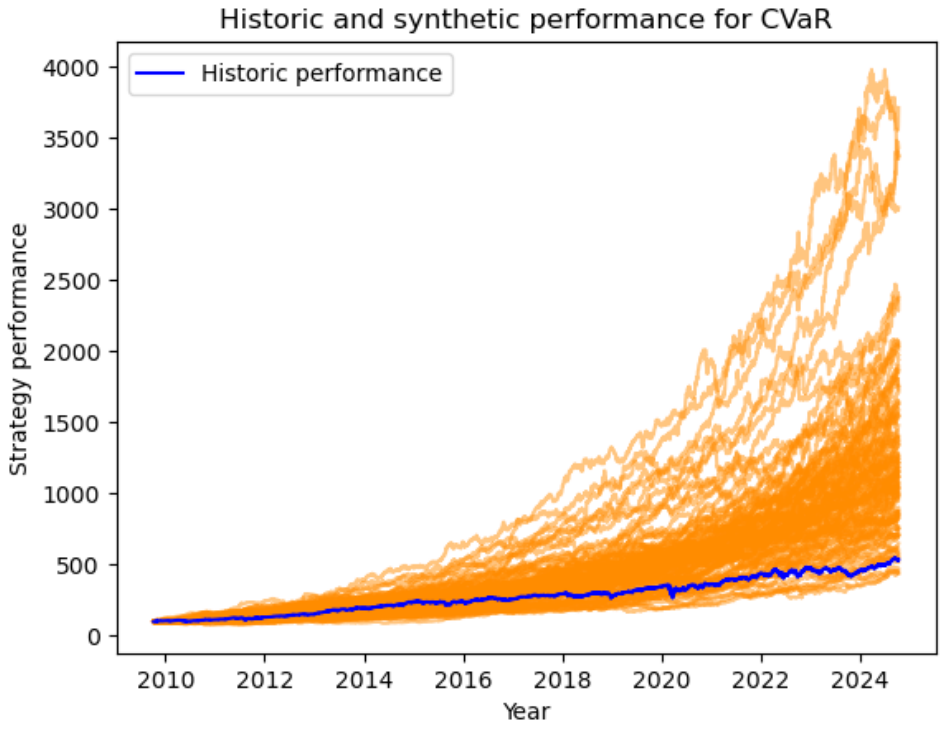

Figure 3.5.2: Historical and synthetic performance for the CVaR optimized strategy.

Chapter 4

Instrument Pricing

This chapter presents the final market modeling step before the joint market scenarios R, introduced in Section 1.1, can be used for (subjective) views, stress-testing, optimization, and general risk and return analysis. As explained in Section 3.3, we can focus on simulating future paths R_h, $h \in \{1, 2, \ldots, H\}$, for the risk factors that are relevant for our portfolios and subsequently price all the instruments that we are interested in. The fully general joint simulation of factors and instrument (relative) P&L then allows us to perform very deep investment analysis as presented in chapters 5 to 8.

While this book uses the popular risk factor terminology, these factors can also rightfully be called pricing factors, because they enter directly into the pricing functions of investment instruments. The factor model literature is vast and generally focuses on linear factor models, initially popularized by Ross (1976). This chapter will instead focus on pricing functions, but readers are free to approximate these by linear models if they deem it sufficient to price the instruments they invest it. For example, a derivative instrument's P&L can be approximated by a linear combination of partial derivatives commonly referred to as "Greeks", see Section 4.4. It is generally recommended to perform full pricing and not ignore the residual, unless this is practically infeasible for some reason.

This chapter presents a general framework for investment instrument valuation including some common examples and perspectives on illiquid alternatives. While this framework should apply to most investment instruments, some might be so unique that they require an entirely different approach or a combination of the methods presented in this chapter, for example, callable bonds that are a combination of a bond and a call option.

Section 4.1 gives a presentation of bond pricing and their risk factors, i.e., zero-coupon interest rates, yields to maturity, breakeven inflation, and credit spreads. Section 4.2 presents equity pricing, which to a large extend depends on the investment manager's desired level of granularity and sometimes requires additional factor model estimation. Section 4.3 demystifies derivative instruments that are sometimes perceived as being very different from plain vanilla instruments such as stocks and bonds. The perspectives on derivatives valuation in Section 4.3 combined with the portfolio management framework for derivatives presented in Section 6.1 should illustrate that derivatives only have few extra characteristics compared to other investment instruments. Section 4.4 presents dynamic strategies and a put option delta hedging case study. Section 4.5 contains some general perspectives and principles for illiquid alternatives. Finally, Section 4.6 is a multi-asset pricing case study.

4.1 Bond Pricing

This section gives an introduction to bond pricing. While there exist many different kinds of bonds, their common characteristics is that they have coupon payments c_t that need to be properly discounted to calculate the bond's current price P. The intention of this section is not to present an exhaustive list of bond instruments, but to describe the most common types of bonds and give readers a general understanding of the principles that are used to price them. Readers should then be able to apply these principles to the particular bonds that are relevant for their portfolios.

4.1.1 Nominal Bonds

Nominal bonds are the most common type of bonds, for example, government bonds. The payoff profile of government bonds is usually that of a bullet bond, which is characterized by having coupon payments with some fixed frequency and paying back the borrowed amount, also known as the principal, together with the last coupon payment. For example, a three-year bullet bond with a coupon of 3%, yearly payments, and a principal of 100 would have the following cash flows: 3, 3, and 103.

There are many other repayment profiles for bonds, but this section will focus on bullet bonds and zero-coupon bonds, which are the repayment profiles of US government bonds, the biggest bond market in the world. US government bonds are classified into three categories: bills, notes, and bonds. Bills are short-term investments that mature in one year or less and have a zero-coupon bond repayment profile, where the principal is paid back at maturity without any coupon. Notes and bonds have semiannual payments and are structured as bullet bonds.

Two natural questions arise: how do we price these bonds, and who would want to invest in a zero-coupon bond that does not pay any coupon? The answer to the last question comes from the fact that a lower coupon will be reflected in a lower price, i.e., a US government bond with a 2% coupon is generally not considered to be worse than a bond with a 3% coupon, because it has a lower price.

To understand bond pricing more formally, let us imagine a bond having N coupon payments c_t for $t \in \{t_1, t_2, \ldots, t_N\}$, and that we have access to the associated zero-coupon interest rates r_t. The price of the bond is then given by

$$P = \sum_{t \in \{t_1, t_2, \ldots, t_N\}} (1 + r_t)^{-t} c_t + (1 + r_{t_N})^{-t_N} C.$$

The term $(1 + r_t)^{-t} = D_t$ is know as the discount factor for time t. D_t is the price of a zero-coupon bond with principal set to one. In the above pricing formula, C is the principal which is paid together with the last coupon payment at time t_N.

If we imagine that we have access to the zero-coupon interest rates r_t for all sufficiently long maturities t, we call this collection of interest rates the zero-coupon yield curve. Actual zero-coupon bonds usually have maturity of up to one year. We can extract zero-coupon bond yields directly from these, while we have to estimate them based on traded coupon bonds for longer maturities, see Munk (2011). There also exist Separate Trading of Registered Interest and Principal of Securities (STRIPS) instruments, which partition the coupon bond into zero-coupon bonds. If these are available to you as an investor, you can also extract zero-coupon yields from these.

Another common rate is the yield to maturity, which is defined as the common yield y_{t_N} applied to all repayments, i.e.,

$$P = \sum_{t \in \{t_1, t_2, \ldots, t_N\}} (1 + y_{t_N})^{-t} c_t + (1 + y_{t_N})^{-t_N} C.$$

For zero-coupon bonds, the yield to maturity y_{t_N} and zero-coupon interest rate r_t are the same, because the bond only has one payment that pays back the principal. For coupon bonds, it is only in the uncommon scenario where we have a flat interest rate curve until the maturity of the bond that the two will be the same.

While the yield to maturity y_{t_N} can be interesting and useful for bond investors, who use it to analyze bonds in various ways, it is usually the zero-coupon curve, r_t with $t \in \{t_1, t_2, \ldots, t_N\}$, that we are trying to simulate for risk purposes. If we can generate good simulations for zero-coupon curves of, for example, US treasuries, we can price any US government bond using these simulations and consequently generate future P&L scenarios.

In practice, we probably have to generate scenarios for some key maturities of the zero-coupon interest rate curve and interpolate between these points if it is necessary to price the bonds we are interested in. There exist several interpolation methods, for example, spline interpolation or Nelson-Siegel(-Svensson) parametrization. We will not go into details with these in this book but refer interested readers to Munk (2011) or other freely available resources.

4.1.2 Inflation-Linked Bonds

Inflation-linked bonds (ILBs), also conversationally known as linkers, differ from nominal bonds due to their link to inflation and inflation expectations. They have the special feature that their principal and coupon payments are adjusted by an inflation index such as the Consumer Price Index (CPI). The price of an inflation-linked bond is given by

$$P = \sum_{t \in \{t_1, t_2, \ldots, t_N\}} D_t \bar{c}_t + D_{t_N} \bar{C},$$

where

$$\bar{c}_t = \frac{CPI_t}{CPI} c_t \quad \text{and} \quad \bar{C} = \frac{CPI_{t_N}}{CPI} C$$

are the inflation-adjusted coupons and principal, with CPI being the current level of the price index.

The difference between the yield on nominal bonds and inflation-linked bonds with the same maturity is known as the breakeven inflation, while the yield on inflation-linked bonds is called the real rate. Inflation-linked bonds are usually not traded as much as nominal bonds, which can complicate estimation of the real yield curve directly from bond prices. However, there exists a fairly liquid market for inflation swaps, which trades many different maturities and allows us to estimate the real rate curve by subtracting the inflation swap strike from the nominal bond curves.

Inflation-linked bonds typically also have a feature where the principal is protected against deflation, i.e., if $\frac{CPI_{t_N}}{CPI} < 1$ then $\bar{C}$ is set equal to C. Note that this usually only applies to the principal and not the coupons. Hence, this feature should be incorporated into our P&L modeling of ILBs, while we

can perform the same stationary transformations to the breakeven inflation as we do to interest rates and the credit spreads below.

4.1.3 Credit Bonds

Credit bonds differ from nominal government bonds and inflation-linked bonds, because they have a default probability associated with them. For example, a company that is not able to pay back its debt or a country borrowing in a foreign currency. While a country's bonds in its own currency should in principle not be able to default, as the country can print more money to repay this debt, there have been several historical instances of countries defaulting on debt in their own currency. It is also important to keep in mind that the country's currency can be worth very little due to inflation if they decide to print money to repay the debt.

Since credit bonds are considered to have a nonnegligible default probability associated with them, investors require an additional compensation compared to government bonds. This additional compensation s_{t_N} is usually measured against the yield to maturity y_{t_N} on a bond with the same time to expiry as the credit bond, i.e., the price of a credit bond with coupons c_t, $t \in \{t_1, t_2, \ldots, t_N\}$, is given by

$$P = \sum_{t \in \{t_1, t_2, \ldots, t_N\}} (1 + y_{t_N} + s_{t_N})^{-t} c_t + (1 + y_{t_N} + s_{t_N})^{-t_N} C.$$

The quantity s_{t_N} is called the credit spread for maturity t_N. It is a function of the bond's default probability and the potential recovery payment in the case of a default. Credit bond investors might carefully analyze these quantities to determine if they believe the current credit spread s_{t_N} represents these risks appropriately. For our multi-asset risk modeling purposes, simulating the credit spread s_{t_N} will usually be sufficient. As we have seen in Section 3.1, we can apply the same stationary transformations to credit spreads as we do to zero-coupon interest rates. The interpretability of a log return of the constant maturity zero-coupon bond just disappears in this case, but interpretability is not a core objective of the stationary transformations. We just want something that is easier for our statistical models, such as the ones presented in Section 3.2, to project into the future.

4.2 Equity Pricing

The main difference between equities and bonds is that equity instruments usually do not have any promises about coupon payments. Companies can still pay dividends or buy back shares, but they are usually not contractually obliged to do so, and there are often legal regulations on when a company can pay out dividends.

This section will present two types of equity pricing models; fundamental models that find the price by discounting future cash flows with an equity risk premium on top of a risk-free zero-coupon curve and factor models, which try to explain the equity return as a function of several characteristics. Fundamental models use the same principles as bonds and are quite similar to credit bond pricing, with the main difference being that the future cash flows are stochastic and in principle can continue in perpetuity. While factor models are most frequently used, we start with fundamental models while the bond pricing is still fresh in mind.

We note that both fundamental and factor models require additional estimation and assumptions compared to direct simulation of the equity return and price. The market simulation methods from Section 3.2 do not require that you impose these fundamental or factor structures on the equity instruments. They are simply presented here in case you want to, and because they are commonly seen and talked about in practice.

4.2.1 Fundamental Models

Fundamental equity models have a pricing equation that is quite similar to bonds, with the main difference being that we are trying to find the present value of a stochastic cash flow that potentially runs in perpetuity. The pricing equation usually looks something like this

$$P = \sum_{t \in \{t_1, t_2, \ldots, t_N\}} (1 + r_t + p)^{-t} d_t + (1 + r_{t_N+1} + p)^{-T_N+1} \frac{d_{t_N+1}}{r^\star + p - g},$$

where p is the equity risk premium, i.e., the extra compensation investors require for bearing equity risk, $r^\star$ is some long-term estimate of the risk-free rate, d_t are the dividend payments, and g is a long-term dividend growth rate estimate. The last term is the present value of a perpetuity with an assumed constant growth rate g.

Fundamental stock picking investors usually apply this dividend discount model, or some residual operating earnings variant of it, to reformulated accounting numbers, see Penman (2012). These investor will then input a risk premium p for the particular stock and see if the price is above or below the current market price. On the other hand, multi-asset investors will usually apply this model to an index like the S&P 500 and solve for the risk premium to assess how attractive the index return is and use it as a signal. The author has experience with both of these applications of the model.

Since the model builds on many forecasts and assumptions, it is probably a good idea to jointly sample these assumptions and get a distribution for the price of the stock or index. This is also our objective when it comes to market simulation. The challenge comes from the fact that we must perform a joint simulation of the dividend paths and other parameters together with the potentially very high-dimensional market we are trying to model. This can be very challenging in practice if we want to use the dividend discount model for many stocks or indices, and it is not something that the author has seen done before for a high-dimensional market simulation.

4.2.2 Factor Models

Due to the infeasibility of using dividend discount models in high dimensions, equity returns $R_{i,t}$ are usually assumed to follow a factor model. The general form of a factor model is

$$R_{i,t} = f_i(F_t) + \varepsilon_{i,t}, \tag{4.2.1}$$

where $F_t \in \mathbb{R}^N$ are the factor realizations, and $\varepsilon_{i,t}$ is a residual for stock i at time t. As with so many other things in finance and economics, the usual factor model formulation is linear and given by

$$R_{i,t} = \alpha_i + \beta_i^T F_t + \varepsilon_{i,t}, \tag{4.2.2}$$

where $\alpha \in \mathbb{R}$ is an intercept that allows us to assume that the residual has zero mean without loss of generality, and $\beta \in \mathbb{R}^N$ are the "factor loadings". The linear formulation implies that

$$\mathbb{E}\left[R_{i,t}\right] = \alpha_i + \beta_i^T \mathbb{E}\left[F_t\right]. \tag{4.2.3}$$

From the linear expected return formulation, we can immediately conclude that the CAPM model from Section 2.1 is a linear factor model with one factor $F_t = R_m - R_f$. Other examples of linear factor models are the APT introduced by Ross (1976) and the Fama-French model introduced by Fama and French (1992).

The expected return expression for a linear factor model (4.2.3) tempts some people to ignore the residual in (4.2.1) when performing market simulation. This makes it easier to compute marginal risk contributions and assign them all to specific factors as presented in Section 7.1, but it potentially ignores very important characteristics of the residual, so this approach is generally not recommended in this book.

The presentation of linear factor models in this section is very short and sweet. There are many details and formalities which are purposefully skipped. Some people might even say that CAPM or the APT does not fall into their definition of linear factor models. We will not delve into these details as they are not essential for our purposes and simply conclude that the expected return expression has a linear factor model form. For more details on linear factor models, see Meucci (2014).

4.3 Demystifying Derivatives

Derivatives have a tendency to be treated as alien instruments by investment managers or the "buy-side". The opposite is true for market makers or the "sell-side", first introduced in Section 3.2.3, who trade derivatives and are very familiar with their pricing and dynamics. However, sell-side traders tend to talk about derivatives P&L and exposure in monetary terms, for example, $100,000, while buy-side investors tend to think about percentage returns. These differences tend to confuse people, which is why a portfolio management framework for derivative instruments was first publicly documented by Vorobets (2022a), see also Section 6.1.

Besides the differences in market values and exposures, derivatives have the unique characteristic that they cannot be separated from their underlying. Hence, our market simulation as well as views, stress-testing, and optimization must be consistent as explained in Section 6.4.2 regarding derivatives portfolio optimization with parameter uncertainty. We start by underlining this point in the next section about forwards and futures.

4.3.1 A Note on Forwards/Futures

A forward/futures contract gives us a linear exposure to the price changes in an underlying asset such as the S&P 500 index. The forward price with time to expiry T and discrete dividends is given by

$$F_T = S_0 e^{r_T T} - \sum_{t \in \{t_1, t_2, \ldots, t_N\}} d_t e^{r_t (T-t)},$$

where S_0 is the current underlying "spot" price, while r_t and d_t are the risk-free zero-coupon interest rates and dividends, which have been previously introduced. In the forward formula, we note that $t \leq T$, i.e., we only discount dividends that are payable during the life of the forward contract. Forward/futures strikes are often conveniently computed using a dividend yield q instead of the discrete dividends. This reduces the formula to

$$F_T = S_0 e^{(r_T - q)T}. \tag{4.3.1}$$

For most forward/futures contracts, the dividends will be close to known during the life of the contract, unless they have a very long time to maturity. These can therefore be extracted from market expectations. If we still want to simulate dividends uncertainty, it is probably a good idea to at least anchor them in the market expectations. Besides that, we see that the forward contract has the underlying and interest rates as risk factors, which we have methods for simulating and pricing as presented in the previous chapters and sections.

Lets now turn to FX forwards, which are commonly used to hedge currency risk. Consider, for example, the EUR/USD spot exchange rate at $S_{EUR/USD} = 1.05$, which means that one euro can be exchanged for 1.05 US dollars. We can simulate this spot exchange rate by performing the same stationary transformations as we do with equity indices and implied volatility surfaces. However, if we also simulate the USD and EUR risk-free interest rate curves, we actually also automatically simulate the forward FX prices, because these are functions of the current spot and the interest rate differential. For example, for a one-year EUR/USD forward contract the forward rate is given by

$$F_{EUR/USD} = S_{EUR/USD} \frac{(1 + r_{1,EUR})}{(1 + r_{1,USD})},$$

where $r_{1,EUR}$ and $r_{1,USD}$ are the one-year risk-free interest rates in EUR and USD, respectively.

4.3.2 The Underlying as a Risk Factor

As already illustrated for forward/futures, the underlying such as the S&P 500 index is a risk factor for the derivative contract, while it is also an instrument that we can invest in. Once we realize this, it becomes clear that derivatives are not some special type of alien instruments, but simply inseparable from the underlying in the same way that bond prices are inseparable from their zero-coupon interest rate curve r_t, $t \in \{t_1, t_2, \ldots, t_N\}$.

Let us next consider plain vanilla European-style put and call options, first introduced in Section 2.3. As explained in Section 2.3, option prices are usually quoted in terms of implied volatility, which we can perceive as a time, underlying, and maturity normalized price of the option. So instead of quoting option prices for S&P 500 and STOXX 50 in index points, the implied volatility makes it easier for us to compare the option prices for these two indices.

When we talk about implied volatility, it is understood that the famous Black and Scholes (1973) model is used to invert the option prices. In practice, it is often the Black (1976) formula that is used, because it expresses the option prices as a function of the forward price F_T from equation (4.3.1), which we can see is a function of the underlying spot price S_0. The Black (1976) formula for European-style

call and put options with strike K is given by

$$c\left(S_0, T, K, \sigma_{T,K}, r_T, q\right) = e^{-r_T T}\left[F_T N\left(d_1\right) - K N\left(d_2\right)\right],$$
$$p\left(S_0, T, K, \sigma_{T,K}, r_T, q\right) = e^{-r_T T}\left[K N\left(-d_2\right) - F_T N\left(-d_1\right)\right],$$

where $N\left(\cdot\right)$ is the normal distribution density while

$$d_1 = \frac{1}{\sigma_{T,K}\sqrt{T}}\left[\ln\left(\frac{F_T}{K}\right) + \frac{1}{2}\sigma_{T,K}^2 T\right] \quad \text{and} \quad d_2 = d_1 - \sigma_{T,K}\sqrt{T}.$$

As we see from the above formula, the main risk factors for European-style put and call options are the underlying spot S_0, the risk-free rates r_T, and the implied volatilities $\sigma_{T,K}$, assuming that the constant dividend yield q is close to known.

If we compute the changes in option prices using the Black (1976) formula, we can view this as a nonlinear factor model similar to (4.2.1) with the residual $\varepsilon_{i,t} = 0$ for all t, in which case we can rightfully call it a pricing function. If we decide to approximate the changes in option prices by a Taylor expansion using option "Greeks", i.e., partial derivatives with respect to the price inputs, we can view this as a linear factor model similar to (4.2.2), in which case $\varepsilon_{i,t} \neq 0$ in general. It is up to you as an investment and risk manager to decide which approach you use, but it is generally recommended not to ignore the residual and perform full pricing whenever it is feasible, because many important nuances that can significantly affect our portfolio construction can be hidden in the residual.

There exist many other derivatives than European-style put and call options, for example, American-style options that allow us to exercise the right to sell/buy during the life of the option. Another example is variance swaps, which were first introduced in Section 2.3. These can be priced either through a replicating portfolio of options or by direct Monte Carlo simulation. The important point is simply that we do what we can to keep the pricing consistent by using the risk factor simulation correctly. It is clearly out of scope for this book to go through all the details of every known derivative instrument, so it is up to the reader to use the principles from this chapter to appropriately simulate the (relative) P&L of the derivative instruments that are relevant for their portfolio. To the author's knowledge, these principles should be sufficient to price any derivative with a satisfactory accuracy for investment management purposes.

4.4 Dynamic Strategies and a Delta Hedging Case Study

As already mentioned in the introductory Section 1.1, the market simulation framework, which focuses on generating risk factor paths for each horizon $h \in \{1, 2, \ldots, H\}$, allows us to simulate the P&L of dynamic strategies that are rebalanced using some rule. This could, for example, be trend following strategies that are commonly implemented by commodity trading advisers (CTAs). Trend following strategies are typically implemented with forward/futures contracts, presented in Section 4.3.1, using proprietary trend signals to identify when the underlying asset is trending upwards or downwards. Although the trend following investment strategy is clearly path dependent, it is still meaningful to analyze its cumulative P&L at some specific horizon h.

As trend following strategies rely on some signal, which is beyond the scope of this book, we will instead use another commonly interesting dynamic strategy, which is the delta hedging of an option. Let us imagine that we have a portfolio consisting of an equity index and a one month put option with an at-the-money forward strike K. The intention of the put option is clearly to limit the downside of the portfolio. If we statically hold the option to maturity, we can just look at the one month horizon and compute the option's P&L at expiry given by $\max(K - S_T, 0) - p_0$, with S_T being the value of the underlying index after one month, and p_0 being the initial option price. So if $K - S_T \leq p_0$, the put option has been unprofitable over the one month horizon.

An alternative to the buy and hold strategy is to delta hedge the put option, i.e., trade forwards/futures so that the derivative portfolio's P&L is not sensitive to first-order changes in the underlying index. If the market movements end up being very volatile during the one month period, the delta hedging strategy can actually provide a positive P&L at the end of the horizon, even if $K \leq S_T$. To understand this, we can decompose the put option's P&L using its "Greeks", i.e.,

$$dp = \frac{\partial p}{\partial t}dt + \frac{\partial p}{\partial S}dS + \frac{\partial p}{\partial \sigma}d\sigma + \frac{\partial p}{\partial r}dr + \frac{1}{2}\frac{\partial^2 p}{\partial S^2}dS^2 + \varepsilon. \tag{4.4.1}$$

Formula (4.4.1) is the typical decomposition with $\frac{\partial p}{\partial t} = \Theta$ called theta, $\frac{\partial p}{\partial S} = \Delta$ called delta, $\frac{\partial p}{\partial \sigma} = \mathcal{V}$ called vega, $\frac{\partial p}{\partial r} = \rho$ called rho, and $\frac{\partial^2 p}{\partial S^2} = \Gamma$ called gamma. Note that the formula is presented here for a put option, but it might as well have been a call option, having different values for the "Greeks". It is also possible to approximate the option P&L with higher order "Greeks" and make the residual ε close to zero. Hence, this is an alternative to the full Black and Scholes (1973) pricing presented in Section 4.3.2.

We can also decompose the P&L of a portfolio trading a put option and employing a delta hedging strategy using a forward/futures contract. In that case, the expression will be the following

$$d\Pi = \frac{\partial \Pi}{\partial t}dt + \frac{\partial \Pi}{\partial \sigma}d\sigma + \frac{\partial \Pi}{\partial r}dr + \frac{1}{2}\frac{\partial^2 \Pi}{\partial S^2}dS^2 + \varepsilon,$$

where Π denotes the portfolio value. Note that $\frac{\partial p}{\partial S} = 0$ due to our delta hedging strategy. If we hold the put option to expiry, we cannot realize the potential gains from increases in the implied volatility, so the net effect of vega, $\frac{\partial \Pi}{\partial \sigma}$, will be zero. Rho, $\frac{\partial \Pi}{\partial r}$, is typically very small in magnitude, so the main drivers of a delta hedged option's cumulative P&L, which is held to expiry, will be the terms $\frac{\partial \Pi}{\partial t}dt$ and $\frac{1}{2}\frac{\partial^2 \Pi}{\partial S^2}dS^2$. For a long position in a put option, theta $\frac{\partial \Pi}{\partial t}$ will be negative (the value of the option decreases as time to expiry decreases), so whether the delta hedged strategy is profitable over the one month horizon or not depends mainly on the cumulative magnitude of $\frac{1}{2}\frac{\partial^2 \Pi}{\partial S^2}dS^2$ and $\frac{\partial \Pi}{\partial t}dt$.

The case study to this section illustrates the effect of delta hedging and compares the P&L of the different option strategies at the one week, two week, three week, and one month horizons. We use the risk factor simulation from Section 3.2.1. Since we only have data for one month implied volatility and strikes from -10% to $+10\%$, we make some simplifying assumptions when computing the option's P&L simulation. In real-world applications, we should use the implied volatilities that corresponds to the correct strike, while the main focus of this example is to illustrate how dynamic strategies can be simulated and analyzed at different horizons. Readers can see the details in the accompanying code.

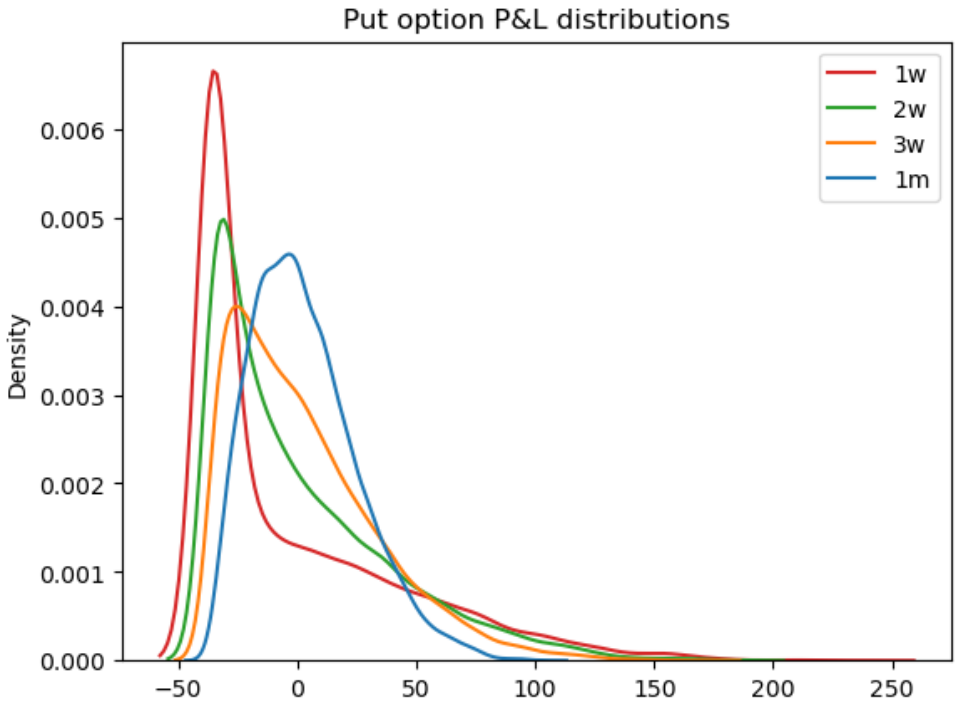

Figure 4.4.1: One month ATMF put P&L at different horizons.

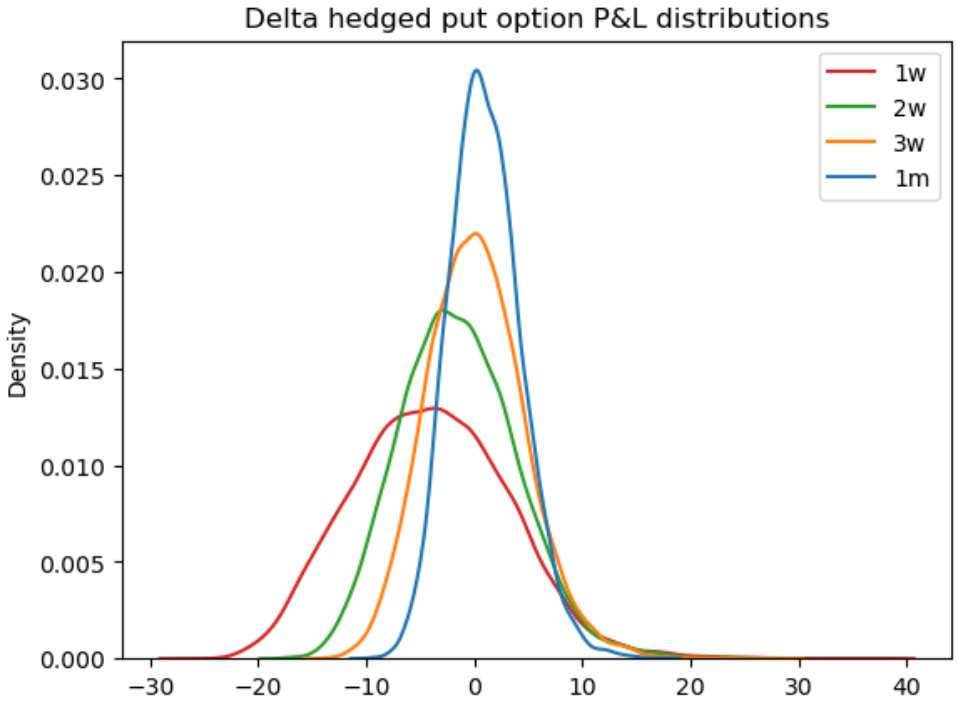

Figure 4.4.2: One month ATMF delta hedged put P&L at different horizons.

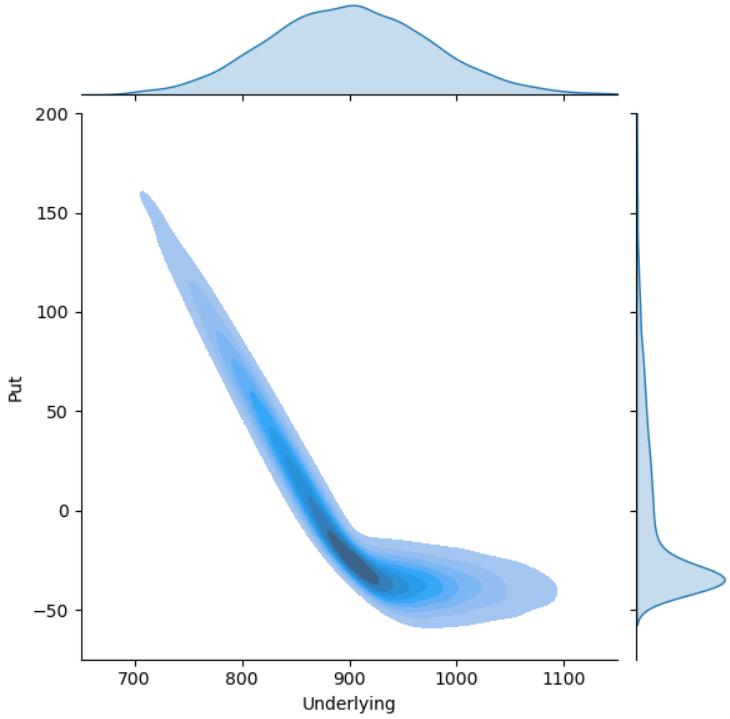

Figure 4.4.3: Joint plot for underlying asset and one month put P&L.

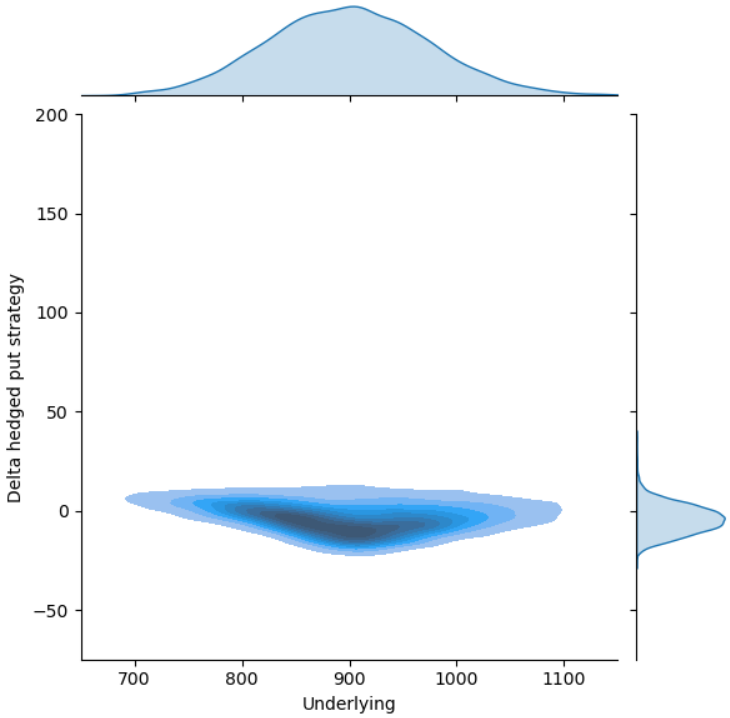

Figure 4.4.4: Joint plot for underlying asset and one month delta hedged put strategy P&L.

From the figures above, we see how the options P&L for the buy and hold strategy eventually converges to $\max(K - S_T, 0) - p_0$, while the delta hedged strategy offers some convexity at the end of the one month period. From the last figure, we clearly see that the delta hedging has been fairly successful. We note however that it is not perfect, both due to the frequency of the hedging, which is one day, and due to some of the simplifying assumptions that have been made in the pricing and estimation of Δ using the Black and Scholes (1973) formula. For more information about option "Greeks" and delta hedging, see Hull (2021).

4.5 Illiquid Alternatives

Illiquid alternatives such as private equity, real estate, and infrastructure create additional challenges due to their infrequent price updates. These investments are usually "marked-to-model" with some predetermined frequency such as every quarter. The mark-to-model pricing is often regulated and based on accounting principles. Hence, we cannot easily simulate these instruments using high-frequency data because we have none. Another issue is that the mark-to-model prices often arrive several months after the end of the quarter and can therefore have some look-ahead bias built into them. Due to these aspects, it is common to observe autocorrelation in illiquid alternatives returns that usually underestimate the actual risk of the investment. In principle, we can only calculate trustworthy returns once these exposures have been closed down.

For alternatives that have a liquid counterpart like equities, the market modeling approach often involves a linear factor model mapping, as presented in Section 4.2.2. For example, assuming that the exposure corresponds to some element of the general equity market combined with a small cap bias. Particular investments can also be modeled in greater detail, for example, a fund investing into European technology companies can be mapped to a European technology index. Simple approaches along these lines are often used in practice, because they seem to somewhat align with the guidelines that regulators provide.

The approximate modeling of illiquid alternatives might seem unsatisfactory, but we should remember that these investments are strategic in nature. It is simply not possible or feasible from a transaction costs perspective to trade these frequently. Hence, for portfolio construction illiquid alternatives are often treated as exposures that are already there, with the initial allocation often determined based on a largely qualitative analysis. Once the allocation has been made, it makes sense for asset allocators to account for their approximate risk when determining the liquid part of the investment strategy.

For alternative investments where there is no liquid equivalent, there seems to be no way around trying to model the investment from fundamental perspectives, i.e., use models similar to the dividend discount model presented in Section 4.2.1. Note that we usually have access to good data for the risk-free rates, which we can simulate using the methods from Chapter 3, and that it is hard to imagine an asset that is completely independent of the funding conditions. Hence, it is the modeling of the risk premium and the future payments jointly with the rest of the market that is challenging. Such challenges are also present for more liquid alternatives like individual hedge funds. The author is not familiar with good and general models for such investments, which probably must be handled on an ad hoc basis. Simply mapping the hedge fund exposures to some hedge fund index usually does not

give an accurate representation of the actual risk and return characteristics.

4.6 Multi-Asset Pricing Case Study

This section shows you how to use the equity, government bond, and option pricing formulas from this chapter applied to the risk factor simulation from Section 3.3. This is simply to give you a concrete example of how these fairly simple formulas work. We note that for the equity, we use the simplest factor model version and simulate the returns directly. For the government bonds and options, we apply the formulas from this chapter directly. It is left as an exercise for readers to compute the return simulations for the corporate bond using the credit spread simulation. Similar to Section 4.4, we have to make some simplifying assumptions for the option pricing due to the data that is available to us. All computations are given in the accompanying code to this section, so readers can see all the details.

We start by illustrating the one month joint price simulation for the equity index and three month at-the-money-forward (ATMF) put and call options having the index as the underlying in Figure 4.6.1.

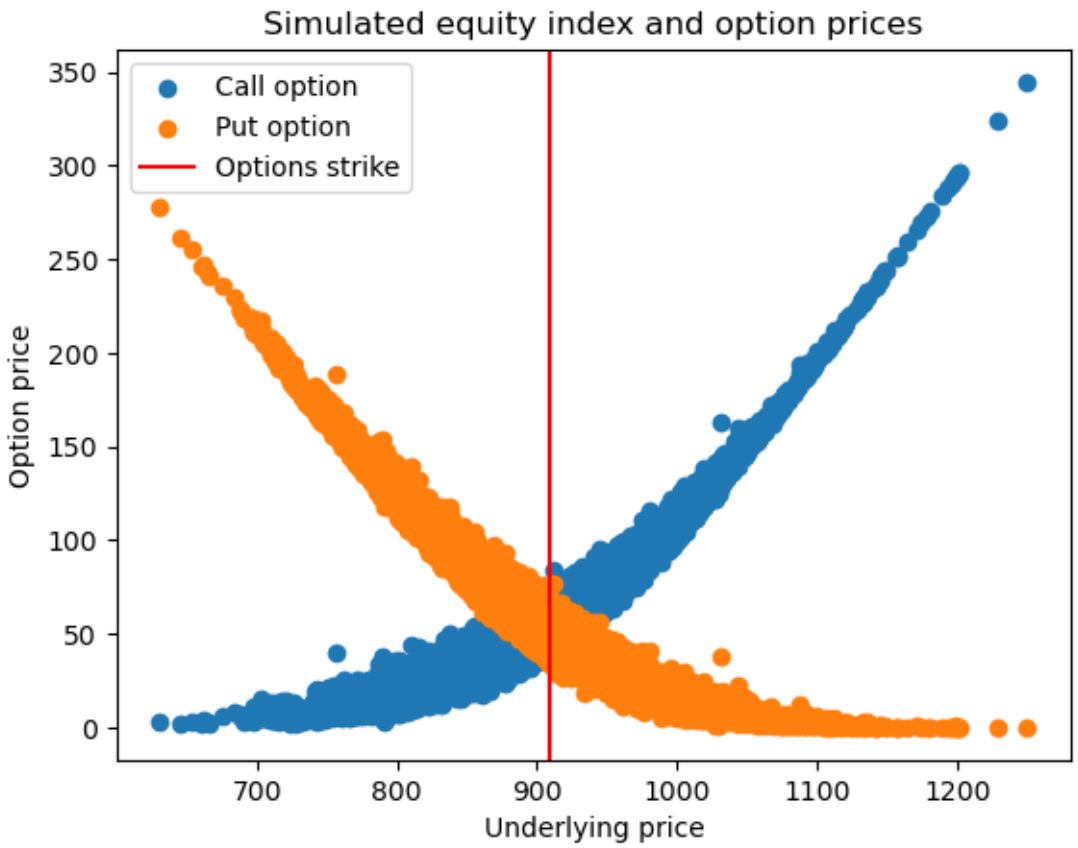

Figure 4.6.1: Equity and options P&L.

We note that the price in Figure 4.6.1 is given in index points. Readers can see the accompanying code for an example of how to convert these prices to relative P&Ls for the options, using the portfolio

management framework for derivatives from Section 6.1, and returns for the equity index. It is also important to remember that there is a joint simulation of the implied volatilities and interest rates for the options. Hence, it is not just the underlying index that affects the option prices.

Finally, we show a joint histogram plot for the equity and bond returns in Figure 4.6.2. The bond return has been computed using the formula from Section 4.1.1 with simulated zero-coupon interest rates. We note from Figure 4.6.2 that the state of the data is such that one-month correlation between bonds and equities is positive. We also note that the bond risk is very low compared to the equity risk. Finally, it is important to keep in mind that the time series simulation which follows with the fortitudo.tech Python package is not necessarily a realistic representation of investment market behavior, because it is generated using stochastic differential equations presented in Section 3.2.3.

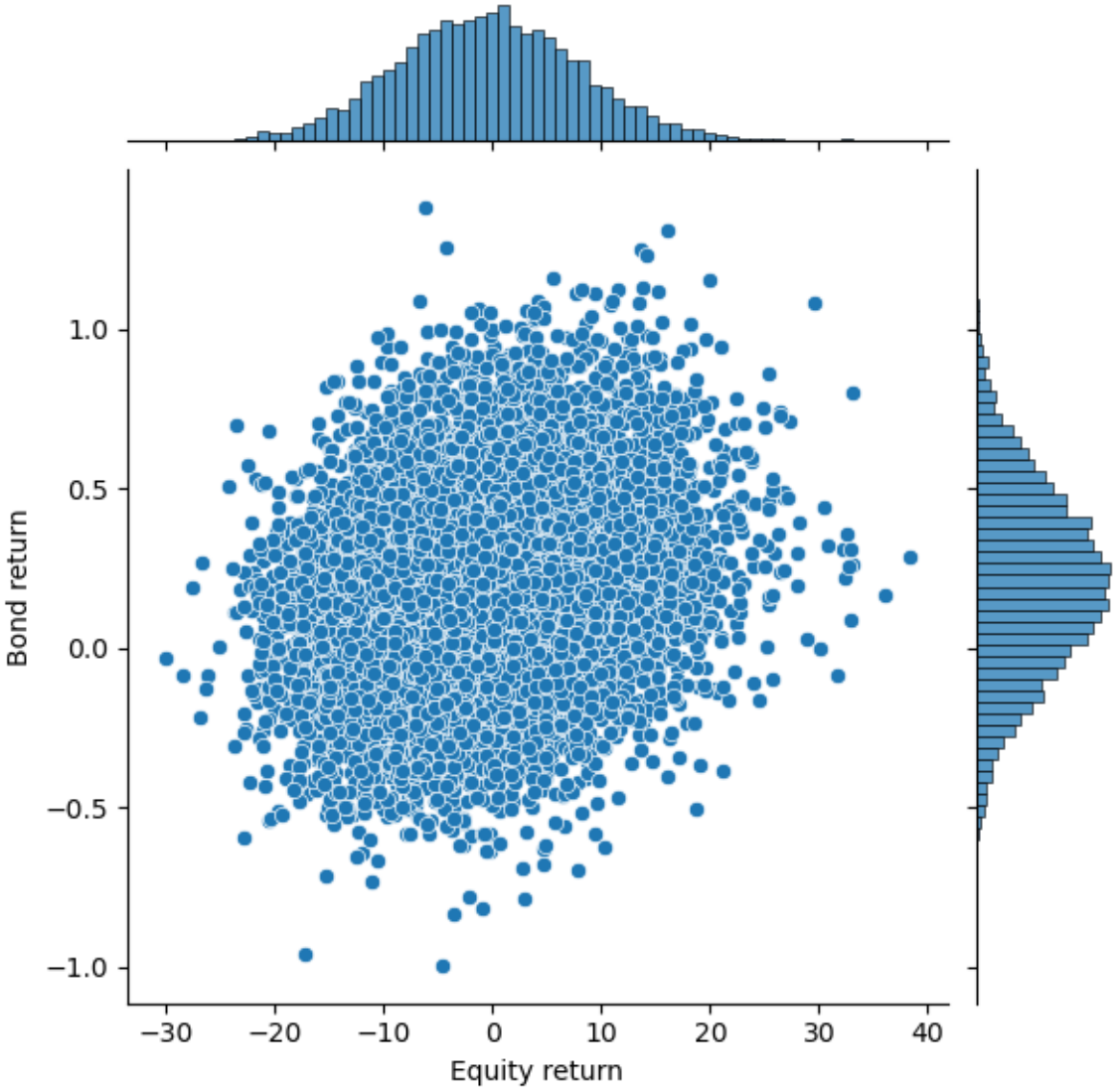

Figure 4.6.2: Joint equity and bond return histogram.

Chapter 5

Market Views and Stress-Testing

This chapter gives a comprehensive presentation of the Entropy Pooling (EP) method, introduced by Meucci (2008a) and refined by Vorobets (2021), as well as its applications in causal and predictive market views and stress-testing as introduced by Vorobets (2023). While all this content is in principle publicly available already, it seems to be challenging for most people to understand. Hence, this chapter fills in the gaps in one place with a careful treatment and unified notation.

Entropy Pooling is a way to update the prior probability vector p, associated with the market Monte Carlo distribution R, by inputting information about the desired update. Some people like to think about Entropy Pooling as a generalization of the Black-Litterman (BL) model without all the oversimplifying normal distribution and CAPM assumptions, additionally avoiding the questionable engineering with the τ parameter. I think EP is so much more than that, so it is not doing it justice to make this comparison, which you will hopefully also discover after reading this chapter.

Section 5.1 presents the basic version of Entropy Pooling, which at its core is a relative entropy (Kullback–Leibler divergence) minimization between the prior probability vector p and the posterior probability vector q subject to linear constraints on the posterior probabilities. The section also presents some of the common views specifications, ranking views, and the nuances of VaR and CVaR views.

Section 5.2 presents the Sequential Entropy Pooling (SeqEP) refinement introduced by Vorobets (2021), which improves on many of the limitations imposed by the requirement that views are specified as linear constraints on the posterior probabilities and usually gives much better results. The sequential approach can also solve practically interesting problems that the original approach simply cannot.

Section 5.3 presents aspects related to view confidences and multiple users or states, which is presented in the appendix of Meucci (2008a) but is still a largely unexplored area of the EP method. View confidences give additional nuances to the method, and user weights or state probabilities allow us to input potentially conflicting views that are aggregated into one posterior distribution.

Section 5.4 uses the multiple states results and presents the Causal and Predictive Market Views and Stress-Testing framework from Vorobets (2023). This framework combines Entropy Pooling with a causal Bayesian network layer on top to generate joint causal views and their associated joint probabilities, which allow us to compute a single posterior distribution that incorporates the causality assumptions from the Bayesian network.

5.1 Entropy Pooling (EP)

As mentioned in the introductory Section 1.1, Entropy Pooling (EP) solves the relative entropy (Kullback–Leibler divergence) minimization problem subject to linear constraints on the posterior probabilities

$$q = \underset{x}{\operatorname{argmin}} \left\{ x^T \left(\ln x - \ln p \right) \right\} \tag{5.1.1}$$

subject to

$$Gx \leq h \quad \text{and} \quad Ax = b.$$

The first natural question that arises is: why minimize the relative entropy? The short answer is because it has good properties for our updating problem, similar to the mean squared error having good properties for linear regression. However, this explanation is probably insufficient for most people from an intuitive perspective, which is arguably important for the adoption of a new method when nontechnical people ask them to explain it.

To start building Entropy Pooling intuition, we must first be clear about what we intuitively will not be able to explain. We will not be able to intuitively explain the actual value of the relative entropy, but we can transform it to the effective number of scenarios given in (5.1.2) below, which has a nice intuitive interpretation as the probability mass concentration over the S joint Monte Carlo samples in R.

The relative entropy represents a statistical distance between two distributions p and q, while it is not a mathematical metric because it is asymmetric and does not satisfy the triangle inequality. It can be interpreted as the expected excess surprise from using the distribution q instead of p. We are now approaching the essence of what we are doing when we are minimizing the relative entropy. We are minimizing the spuriosity while updating our prior distribution p to the posterior distribution q.

In relation to the spuriosity, we probably want to avoid assigning all probability to one scenario s and get a degenerate posterior market distribution q. Unless we have it as an actual view or stress-test, we probably also want to avoid introducing dependencies that are completely opposite of what we have in our prior simulation. For example, if we have two equity indices that are highly dependent, we usually want a stress-test on one of them to affect the other, see the example with S&P 500 and STOXX 50 in Section 5.2 below.

Besides operating on fully general Monte Carlo distributions R, taking the potentially very complex dependencies into account is where the true power of Entropy Pooling comes from. For example, it is just as easy to stress-test a portfolio containing only S&P 500 as a portfolio containing S&P 500 and thousands of derivatives on S&P 500. Entropy Pooling essentially makes a prediction on how other instruments and factors are expected to behave under the posterior distribution q. As we will see in Section 6.2.1 below, there is no need to reprice European put and call options when we perform Entropy Pooling stress-testing. Their posterior P&L distribution is automatically given to us.

With all of the above in mind, it hopefully becomes clear that Entropy Pooling is so much more than the BL model, which relies on the oversimplifying and empirically rejected normal distribution and CAPM assumptions. It sounds nice that "equilibrium expected returns" can be extracted using the BL model, but since these have very little to do with reality, they probably do not add any actual investment value and might even be detrimental.

In summary, Entropy Pooling is a theoretically sound method for implementing market views and stress-testing fully general Monte Carlo distributions R. It helps us predict what will happen to all instruments that we invest in and all factors that our portfolios are exposed to. It will ensure logical consistency in our derivatives P&L. As we will see in Section 5.3 below, Entropy Pooling also handles view confidences and state probabilities in a much more natural, probabilistic way than the BL model with its τ parameter, see Meucci (2008b) for an explanation of the issues and paradoxes. Hence, there is no reason to continue using BL when fast and stable Entropy Pooling implementations are freely available. If you for some reason want to use the CAPM prior, you can still do that to simulate R while getting all the Entropy Pooling benefits when implementing views and stress-tests.

To explore Entropy Pooling further from a mathematical perspective, we start by noticing that if all elements of the prior probability vector p are equal to $\frac{1}{S}$, the second term of (5.1.1) reduces to $\ln S$, which does not affect the solution, i.e., in the uniform prior probability case we can reduce the problem to

$$q = \underset{x}{\operatorname{argmin}} \left\{ x^T \ln x \right\} = \underset{x}{\operatorname{argmax}} \left\{ -x^T \ln x \right\} = \underset{x_1, x_2, \ldots, x_S}{\operatorname{argmax}} - \sum_{s=1}^{S} x_s \ln x_s.$$

The expression $-\sum_{s=1}^{S} x_s \ln x_s$ is known as the entropy, which we are trying to maximize when the prior probability vector p is uniform.

In most practical cases, the prior probability vector p will be uniform, while we have the opportunity to specify any valid prior probability vector with strictly positive elements that sum to one. Hence, Entropy Pooling will in most practical cases correspond to entropy maximization subject to linear constraints on the posterior distribution. Many more technical details are given about the properties of relative entropy minimization by Caticha and Giffin (2006), who also show that it corresponds to a generalization of Bayesian updating in the sense that information about the posterior distribution is given by constraints instead of the usual data. Caticha and Giffin (2006) also explain that entropy does not require any interpretation in this situation. It just has good properties for updating the distribution when additional information is given about the moments of the posterior distribution.

There is generally a lot of information about the relative entropy (Kullback–Leibler divergence), which is a quantity used in many different fields of mathematics and statistics. Readers who are interested in exploring this further must be warned that the content can be confusing due to the order of the distributions in (5.1.1). Sometimes, the objective is formulated equivalently as $-x^T (\ln p - \ln x)$, while in other cases the formulation remains the same but the prior and posterior order is switched for computational convenience. Finally, there are situations where the prior is implicitly assumed to be uniform, in which case we have seen that relative entropy minimization corresponds to entropy maximization.

A useful way of assessing how much the views and stress-tests deviate from the prior is called the effective number of scenarios, introduced by Meucci (2012a) and given by the exponential of the posterior probability entropy

$$\hat{S} = \exp\left\{ -\sum_{s=1}^{S} q_s \ln q_s \right\}. \tag{5.1.2}$$

The idea is that the effective number of scenarios is $\hat{S} = 1$ if all probability mass is given to one scenario, while it is $\hat{S} = S$ when scenario probabilities are uniform and equal to $\frac{1}{S}$. Note that we use

the convention that $q_s \ln q_s = 0$ for $q_s = 0$. It is often convenient to compute the relative effective number of scenarios $\hat{s} = \frac{\hat{S}}{S} \in (0, 1]$. We note that the relative effective number of scenarios is just the exponential of the negative relative entropy when prior probabilities are uniform. Hence, in a uniform prior case, a lower relative entropy gives a higher effective number of scenarios, which makes intuitive sense as this indicates that the posterior distribution is close to the uniform prior distribution.

The second natural Entropy Pooling questions is: what are the matrices and vectors G, h, A, and b? The short answer is that G and A contain functions of the Monte Carlo simulation R from (1.1.1), while h and b contain constraint values for these functions. From an intuitive investment perspective, this might not help you much, so we improve on that with examples below.

Consider for a moment how you would implement a constraint on the posterior expected value of some price, return, or factor $i = 1, 2, \ldots, I$. If we let R_i denote column i from the matrix R containing the market Monte Carlo simulation, it should be easy to convince yourself that the prior expected value is given by $R_i^T p = \mu_i$. If we want this value to change for the posterior distribution to $\tilde{\mu}_i$, we must implement the constraint that $R_i^T x = \tilde{\mu}_i$, which can be done through the matrix A and vector b. More generally, we can write

$$R_i^T x \gtreqless \tilde{\mu}_i,$$

using $\gtreqless$ to indicate that the view can be an equality or inequality in one of the two directions, with inequality views being implemented through G and h.

Questions related to Entropy Pooling's limitations are more subtle. For example, we cannot implement views that are not feasible for the Monte Carlo market simulation in R, i.e., if we want to implement a view that the expected value of some return should be 10% while all our simulations are below 10%, we cannot do that. It is hard to definitely say if this is a bug or a feature, because if R does not contain all the possible scenarios that we believe can occur in reality, it is a prior problem rather than a posterior problem. The prior problem should be fixed by better quality simulations using, for example, the approaches from Chapter 3. Limiting ourselves to the scenarios in R allows us to avoid the potentially computationally expensive repricing of derivatives and other instruments after implementing risk factor views, so in that sense it is a feature.

A significant limitation of the EP method is, however, that views must be specified as linear constraints on the posterior probabilities. Ideally, we would want to be able to solve the problem with fully general constraints $\mathcal{G}(x) \leq h$ and $\mathcal{A}(x) = b$. Although we can specify nonlinear parameter views with linear constraints on the posterior probabilities, we cannot specify them in full generality because we must fix some parameters to be able to specify specific view types. For example, consider the variance view

$$\left(R_i^T \odot R_i^T\right) x - \left(R_i^T x\right)^2 \gtreqless \tilde{\sigma}_i^2,$$

which is clearly nonlinear in the posterior probabilities x, where we use $\odot$ to denote the element-wise Hadamard product.

Note that we in the specification of views will use a concise programming broadcast notation, where we subtract scalars from vectors. We use this notation to replicate the way that these views would be implemented in most programming languages. From a strictly mathematical perspective, readers can imagine that there is a conforming S-dimensional vector of ones multiplied with scalar values such as

$\left(R_i^T x\right)^2$ that we do not want to constantly replicate.

To specify a variance view with linear constraints on the posterior probabilities, we must fix the second term to some value $\tilde{\mu}_i$ and specify the variance view through the two constraints

$$R_i^T x = \tilde{\mu}_i \quad \text{and} \quad \left(R_i^T \odot R_i^T\right) x - \tilde{\mu}_i^2 \gtreqless \tilde{\sigma}_i^2.$$

There is naturally the philosophical question of whether one can have a variance view without having a mean view. While we will not delve into this, we note that it is a fact that we would be able to implement the variance view with nonlinear functions on the posterior probabilities without specifying a mean view. Hence, as we will clearly see later in Section 5.2, the linear constraints pose a significant limitation. The original suggestion by Meucci (2008a) is to fix the mean to the prior, i.e., $\tilde{\mu}_i = R_i^T p = \mu_i$, while we note that this imposes an implicit view about the mean staying the same.

5.1.1 Solving the EP Problem

The Entropy Pooling problem formulation (5.1.1) is a convex problem with linear constraints. Hence, the problem has one unique solution that is optimal, see Boyd and Vandenberghe (2004). The problem's Lagrangian function is

$$\mathcal{L}(x, \lambda, \nu) = x^T (\ln x - \ln p) + \lambda^T (Gx - h) + \nu^T (Ax - b),$$

where λ and ν are Lagrange multipliers. Meucci (2008a) shows that the solution is given by

$$x(\lambda, \nu) = \exp\left\{\ln p - \iota - G^T \lambda - A^T \nu\right\}, \tag{5.1.3}$$

with ι being an S-dimensional vector of ones. The solution (5.1.3) illustrates that positivity constraints on scenario probabilities $x \geq 0$ are automatically satisfied and can therefore be omitted.

The original/primal problem formulation is potentially very high-dimensional as it is a function of the number of scenarios S. On the other hand, the Lagrange dual function

$$\mathcal{G}(\lambda, \nu) = \mathcal{L}(x(\lambda, \nu), \lambda, \nu)$$

is only a function of the Lagrange multipliers λ and ν and therefore has dimension equal to the number of views in addition to a Lagrange multiplier for the requirement that posterior probabilities sum to one. Meucci (2008a) therefore proposes to solve the dual problem given by

$$(\lambda^\star, \nu^\star) = \underset{\lambda \geq 0, \nu}{\operatorname{argmax}}\ \mathcal{G}(\lambda, \nu)$$

and subsequently recover the solution to the original/primal problem (5.1.1) by computing

$$q = x(\lambda^\star, \nu^\star).$$

A fast and stable Python implementation for solving the Entropy Pooling problem is freely available in the open-source packages entropy-pooling and fortitudo.tech.

5.1.2 Common Views Specifications and Ranking Views

This section presents views specifications that are commonly used in practice. In particular, mean, volatility, skewness, kurtosis, and correlation views. The convenient feature of these views is also that it is easy to implement ranking views once we understand how views on these parameters are specified directly. The list of views in this chapter is by no means exhaustive. It is simply the views that I have seen being used the most in practice. Once you understand how to specify these views and the limitations the linear constraints impose, you are encouraged to experiment with other types of views.

Before we start writing out the views specifications, we start by defining classes of views. This will probably seem abstract to you at first, but it is actually quite simple once you grasp it. Understanding this view classification will be essential for understanding the sequential Entropy Pooling refinement in Section 5.2, which solves many of the issues imposed by the linear constraints requirement and usually gives significantly better results.

We have already seen that some views can be naturally specified through linear constraints without loss of generality, for example, views on the mean. However, views on variances require us to fix the mean in order for us to be able to implement them as linear constraints on the posterior probabilities. With this in mind, let $\mathcal{C}_i$, $i \in \{0, 1, 2, ...\}$, denote the class of parameters that require i other parameters from some or all of the classes $\mathcal{C}_j$, $j = 0, 1, .., i-1$, to be fixed in order to be formulated as linear constraints on the posterior probabilities.

Finally, let $\bar{\mathcal{C}}$ denote the class of parameters that can be formulated as linear constraints on the posterior probabilities but do not belong to any $\mathcal{C}_i$. Hence, $\mathcal{C} = \left\{\mathcal{C}_0, \mathcal{C}_1, \mathcal{C}_2, ..., \bar{\mathcal{C}}\right\}$ is the class of all parameters that can be formulated as linear constraints on the posterior probabilities, with or without fixing other parameters.

Many commonly interesting parameters can be characterized by the $\mathcal{C}_i$ classes. For example, as we have already seen, means belong to $\mathcal{C}_0$, and variances belong to $\mathcal{C}_1$ (mean fixed). Below, we will see that skewness and kurtosis belong to $\mathcal{C}_2$ (mean and variance fixed), and correlations belong to $\mathcal{C}_4$ (two means and two variances fixed), while CVaR views belong to the residual class $\bar{\mathcal{C}}$.

Looking at the linear constraints $Gx \leq h$ and $Ax = b$, we see that a particular Entropy Pooling parameter view is formulated as

$$f(R)\, x \gtreqless c,$$

where $c \in \mathbb{R}$ is a constant, and $f : \mathbb{R}^{S \times I} \to \mathbb{R}^S$ is a function. The elements of the vector $x \in \mathbb{R}^S$ represents scenario probabilities. Hence, we must always include the constraint

$$\sum_{s=1}^{S} x_s = \iota^T x = 1,$$

which we implement through the equality constraints matrix A and vector b. As we saw from the solution to EP problem (5.1.3), the positivity requirement for scenario probabilities $x \geq 0$ will automatically be satisfied, so we do not need to include it in our constraints.

As we have already seen, a view on the expected value of price, return, or factor i is the easiest to implement as

$$R_i^T x \gtreqless \tilde{\mu}_i,$$

Note that most other views require multiple constraints in order for them to be formulated as linear constraints in a logically consistent way. For example, a view on the variance requires an equality view on the mean in addition to a view on the variance using the mean view value, i.e.,

$$R_i^T x = \tilde{\mu}_i \quad \text{and} \quad \left(R_i^T \odot R_i^T\right) x - \tilde{\mu}_i^2 \gtreqless \tilde{\sigma}_i^2.$$

If we do not fix the mean with a view on the expected value i, there is no guarantee that we are subtracting the right constant in the variance view. The interested reader is encouraged to try this out and calculate the posterior variance to verify that it usually becomes incorrect without the mean view. It is generally good practice to compute posterior statistics using the posterior probability vector q to verify that views have been implemented correctly.

With a good understanding of mean and variance views, we proceed to skewness, kurtosis, and correlation views. These views require us to fix means and variances, so we will assume that these have been fixed through the constraints

$$R_i^T x = \tilde{\mu}_i \quad \text{and} \quad \left(R_i^T \odot R_i^T\right) x - \tilde{\mu}_i^2 = \tilde{\sigma}_i^2.$$

The skewness view is then simply specified as

$$\left(\frac{R_i^T - \tilde{\mu}_i}{\tilde{\sigma}_i}\right)^3 x \gtreqless \tilde{\gamma}_i.$$

The kurtosis view is similarly specified as

$$\left(\frac{R_i^T - \tilde{\mu}_i}{\tilde{\sigma}_i}\right)^4 x \gtreqless \tilde{\kappa}_i.$$

Finally, correlation views between i and j are specified as

$$\frac{\left(R_i^T - \tilde{\mu}_i\right) \odot \left(R_j^T - \tilde{\mu}_j\right)}{\tilde{\sigma}_i \tilde{\sigma}_j} x \gtreqless \tilde{\rho}_{ij}.$$

The astute reader might have recognized that views on parameters are simply specified as their definition for discrete distributions, which the Monte Carlo market simulation R represents. We note that we make no assumptions on the actual market distribution being discrete, but any samples R from this distribution will be. It is up to the reader what they believe is a sufficiently good approximation, but sample sizes of $S = 10,000$ seem to be sufficient for most practical purposes, while keeping the computation time unnoticeable. Readers who are interested in seeing a practical example of all these view types being implemented are encouraged to see the case study of Vorobets (2021).

So how would we implement a ranking view? Simply by subtracting two view specifications from each other. For example, for views on the expected value

$$R_i^T x - R_j^T x = \left(R_i - R_j\right)^T x \gtreqless 0.$$

It can often be convenient to multiple one of the terms by a scalar a and specify views such as

$$(R_i - aR_j)^T x \leq 0.$$

This allows us to implement views such as $\mu_i \leq 2\mu_j$, i.e., that the expected value of i is at most $a = 2$ times larger than the expected value of j, assuming that $\mu_j > 0$.

We will not write out the ranking views for variances, skewness, kurtosis, and correlations, but these are also simply subtracting one view specification from another potentially scaled view specification on the same parameter type. It is a good exercise for readers to try this out on their own and compute the posterior values to see whether they have understood this concept or not. Ranking views can of course also be specified across different parameters, for example, that the expected value of i is half the variance of j. While this is a technical possibility, I have not seen it being done in practice yet.

With some practice, this section will hopefully give readers the understanding they need to explore Entropy Pooling views on their own. Readers are generally encouraged to share their experience and views ideas. Practically interesting use cases might be added to this section over time.

5.1.3 VaR and CVaR Views

Having an understanding of the basics of Entropy Pooling views specifications, we are now ready to delve into VaR and CVaR views. While these views are also nothing more than linear constraints on the posterior probabilities, they have some characteristics that make them special. For example, to the author's knowledge it is not possible to easily implement ranking views on these parameters.

For general CVaR views, there is the additional complexity that we a priori do not know the number of scenarios below the VaR. If we have an equality VaR view, that is not an issue, but in the general case we must develop an algorithm for finding an optimal number of scenarios below a yet undetermined VaR value, which makes general CVaR views quite complicated to implement in a fast and stable way.

Meucci, Ardia, and Keel (2011) were the first to analyze VaR and CVaR Entropy Pooling views, while they end up proposing a solution that introduces a determistic grid, because their algorithm for CVaR views with fully general Monte Carlo distributions R does not perform well in practice. We will present VaR and CVaR views in this section without going into details about algorithms that solve the general CVaR view problem, because it is a highly specialized, complex, and proprietary implementation. However, you will be presented for the challenges and of course have access to the solution proposed by Meucci, Ardia, and Keel (2011).

VaR views are fairly straightforward to implement. Let us say that we have a α-VaR view given by the value $\tilde{v}$ for return i. Next, we identify the scenarios $s \in \{1, 2, \ldots, S\}$ where the return simulations R_i are below this α-VaR value and define the row vector $a_{i,\alpha} = (a_1, a_2, \ldots, a_S)$ with elements

$$a_s = \begin{cases} 0 & \text{if} \quad R_{s,i} > -\tilde{v} \\ 1 & \text{if} \quad R_{s,i} \leq -\tilde{v}. \end{cases} \tag{5.1.4}$$

We can then simply add $a_{i,\alpha}$ to the matrix G or A as well as $1 - \alpha$ to the vectors h and b, depending

on whether the view is an inequality or equality view. Hence, for VaR views we simply identify the scenarios where the losses are greater than the VaR view value $\tilde{v}$ and assign them a total probability of probability $1-\alpha$. Since VaR views do not require any parameters to be fixed, VaR views belong to the class $\mathcal{C}_0$. Note that we use the convention that the VaR view value $\tilde{v}$ is specified as a loss, meaning that an α-VaR value of 10% assigns $1-\alpha$ of the probability mass to the scenarios where the elements of R_i are below -10%. We will use the same convention for CVaR views below.

CVaR views introduce an additional complexity when we do not have an equality VaR view, because we do not a priori know the optimal number of scenarios below the VaR, which we can adjust the probabilities of to achieve the desired CVaR view value $\tilde{cv}$. By the optimal number of scenarios below the still undetermined VaR value $\bar{v}$, we naturally mean the number of scenarios that gives us the lowest relative entropy while satisfying the CVaR view.

Once we have a fast and numerically stable algorithm to find the optimal number of scenarios below the VaR, CVaR views become straightforward to implement simultaneously with an equality VaR view implemented through the constraints $a_{i,\alpha}$ in A and $1-\alpha$ in b, replacing $\tilde{v}$ with $\bar{v}$ when computing a_s using (5.1.4). In the presentation of CVaR views below, we will assume that this has been done, so that the VaR is fixed to some value $\bar{v}$, which we did not have a view on.

Assuming that we know the number of scenarios below the α-VaR value $\bar{v}$, α-CVaR views on return i with a view value $\tilde{cv}$ can be formulated as

$$a_{i,\alpha}x = (1-\alpha) \quad \text{and} \quad \left(R_i^T \odot a_{i,\alpha}\right) x \gtreqless -(1-\alpha)\,\tilde{cv}.$$

To understand why this view combination gives us the desired α-CVaR value, let us define $\mathcal{CV} \subseteq \{1,2,\ldots,S\}$ as the set of indices below the α-VaR value $\bar{v}$. Hence, the sample CVaR is then given by

$$\mathbb{E}\left[R_i | R_i \leq -\bar{v}\right] = \frac{\sum_{s\in\mathcal{CV}} R_{s,i}x_s}{\sum_{s\in\mathcal{CV}} x_s} = \frac{\left(R_i^T \odot a_{i,\alpha}\right) x}{a_{i,\alpha}x} \gtreqless \frac{-(1-\alpha)\,\tilde{cv}}{(1-\alpha)} = -\tilde{cv}.$$

We note again that we define the α-CVaR value as a loss, which is why we have a minus in front of the view value $\tilde{cv}$.

As we do not know the number of scenarios below the VaR value a priori, CVaR views belong to the residual class $\bar{\mathcal{C}}$. If we have a VaR view, CVaR could in principle be considered a $\mathcal{C}_1$ view, but as we will see in Section 5.2 this would impose unnecessary implicit constraints, so it is never recommended to classify CVaR views as belonging to $\mathcal{C}_1$.

With CVaR views being classified as $\bar{\mathcal{C}}$, we can shed some more light on what intuitively happens when we compute the optimal number of scenarios below the VaR. Imagine a situation where we have implemented all views except the CVaR view for i, and that we do not have a VaR view for i. The brute-force solution is to loop through all possible configurations of scenarios below the VaR value, which there are in principle S of, and then select the solution with the lowest relative entropy that also satisfies the CVaR view. This is obviously a quite slow way of finding the optimal solution, so we must build algorithms that make good initial guesses and quickly identify the optimal number of scenarios without getting stuck or being too sensitive to the numerical issues that are inherent to practical implementations.

To further understand the issues with developing algorithms for finding the optimal number of

scenarios below the VaR value, see Meucci, Ardia, and Keel (2011). Going into details with these algorithms is beyond the scope of this book and would derail our focus. Interested readers can study the original article by Meucci, Ardia, and Keel (2011) to see a mathematical formalization of how such algorithms can be designed, while we note that their proposed algorithm for fully general Monte Carlo simulations R is not stable enough in practice.

Although we do not go further into detail with general algorithms for implementing the CVaR views, there is an example with a view implementing a 50% increase in the 90%-VaR and 90%-CVaR for daily S&P 500 returns. This gives readers a practical example of how VaR and CVaR views are eventually implemented. We also assess which effect the combined VaR and CVaR view for S&P 500 has on the STOXX 50 daily return distribution. Figure 5.1.1 shows the results. Readers are encouraged to examine the accompanying code to this section for more details and see how the VaR and CVaR views are implemented and validated to verify that they really understand the constraints above.

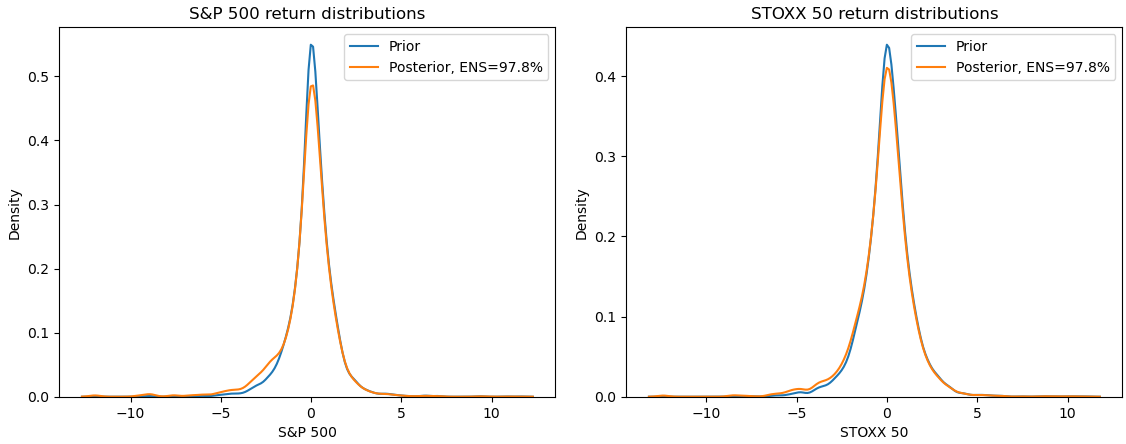

Figure 5.1.1: Daily return distributions including 90%-VaR and 90%-CVaR views for S&P 500.

As with almost all aspects of CVaR analysis for fully general distributions, it is significantly harder to implement CVaR Entropy Pooling views. However, the analysis of fully general distributions and their tail risks also gives us insights that are simply not possible to get using conventional methods based on the mean-variance oversimplification of the market. When we stress-test and analyze scenarios of historical or market simulations, we are really starting to look into the complex nuances of investment markets. Our imagination truly becomes the most limiting factor. Hence, although this section has given you some examples of practical use cases and views specifications, you should not be limited by these. Once you understand the Entropy Pooling technology, you might get ideas that go well beyond what is shown in this and the following sections.

A final interesting use case of CVaR views is to implement them directly on a portfolio's return and analyze how the marginal risk contributions change, both due to the higher standalone CVaR values but also due to changes in diversification properties. See Figure 5.1.2 for the results of such an analysis with a simple log-normal prior return simulation, where we have implemented a 50% increase in the 90%-CVaR of a portfolio containing 25% equities, 20% alternatives, and the rest in government and corporate bonds. This example is generated using a proprietary implementation.

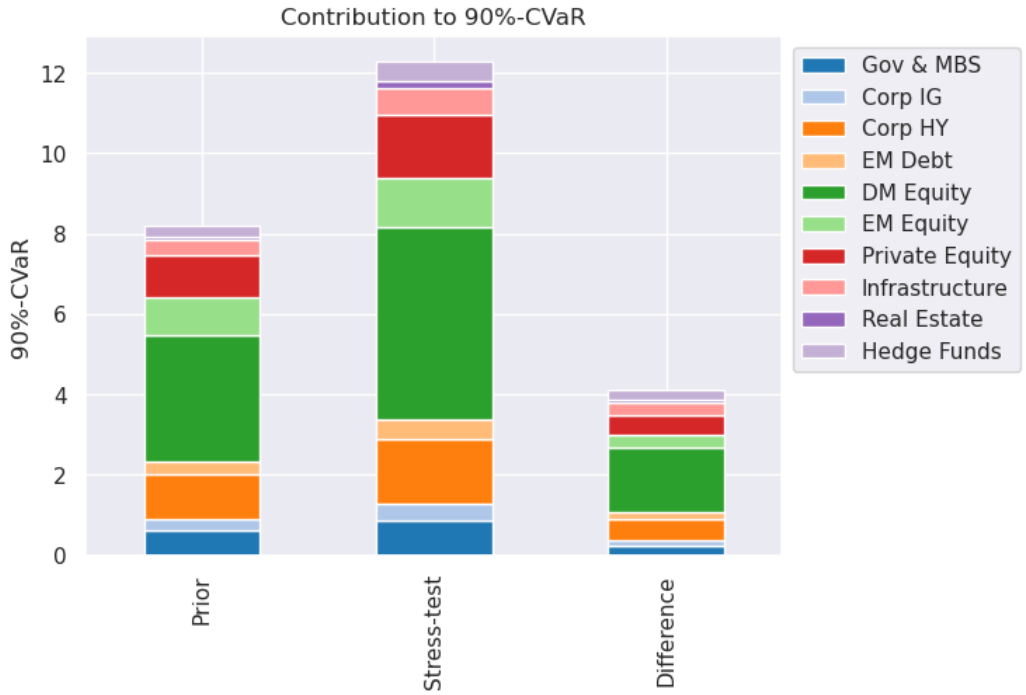

Figure 5.1.2: Portfolio CVaR view.

5.2 Sequential Entropy Pooling (SeqEP)

As mentioned in Section 5.1 above, the requirement that views are specified as linear constraints on the posterior probabilities imposes significant limitations on the EP method. However, this requirement is necessary for us to be able to solve the problem quickly and reliably for high-dimensional market simulations R and their associated prior probability vector p. If we could solve the problem in a fast and stable way using general constraints on the posterior probabilities $\mathcal{G}(x) \leq h$ and $\mathcal{A}(x) = b$, we would do that. Unfortunately, we are not able to do that with the current technology.

The biggest issue stemming from the linear constraints requirement is that we must fix some parameter from the class $\mathcal{C}_i$ to be able to specify a view on a parameter from the class $\mathcal{C}_j$, with $i < j$. The issue stems from the fact that a view for the class $\mathcal{C}_i$ parameter might significantly affect the value of other parameters from the class $\mathcal{C}_i$. For example, if we have a view on the mean of STOXX 50, this view might significantly affect the mean of S&P 500. However, if we also want to specify a variance view for S&P 500, we must fix the mean of S&P 500. The original suggestion of always using the prior mean can therefore impose a potentially strong implicit view.

To overcome these practical limitations and solve additional practically interesting problems, Vorobets (2021) introduces heuristic sequential Entropy Pooling algorithms that sequentially process views according to their class characterizations $\mathcal{C}_i$. The fundamental hypothesis is that views that belong to the class $\mathcal{C}_i$ affect other parameters that belong to the class $\mathcal{C}_i$ more than parameters that belong to the class $\mathcal{C}_j$, with $i \neq j$. For example, that views on expected value of a affect the expected value of b more than views on the variance of a affect the expected value of b.

This fundamental hypothesis is also what practitioners generally experience, while there is no guarantee that it is always true. It is perhaps also possible to generate simulations that specifically exploit the design of the sequential heuristics, but for simulations that resemble real-world market behavior and views on commonly interesting parameters, the assumption seem to hold most of the time.

We now proceed to define the sequential heuristic algorithms, closely following some sections from Vorobets (2021). A particular set of views $\mathcal{V}$ can be partitioned in a similar way to the view class $\mathcal{C}$, i.e., $\mathcal{V} = \left\{\mathcal{V}_0, \mathcal{V}_1, ..., \mathcal{V}_I, \bar{\mathcal{V}}\right\}$ with each $\mathcal{V}_i$ being the set of views on parameters that belong to $\mathcal{C}_i$, and $\bar{\mathcal{V}}$ being the set of views on parameters that belong to $\bar{\mathcal{C}}$. The main idea of the sequential heuristics is to process views according to this partition, carry forward the updated parameters θ_i, $i = 0, 1, ..., I$, and use them to set fixed values when specifying the views in $\mathcal{V}_j$, $j = i+1, i+2, ..., I$, and $\bar{\mathcal{V}}$. More specifically, EP is sequentially applied to the sequence of views with increasing cardinality given by $\mathcal{V}^0 = \{\mathcal{V}_0\}, \mathcal{V}^1 = \{\mathcal{V}_0, \mathcal{V}_1\}, ..., \mathcal{V} = \left\{\mathcal{V}_0, \mathcal{V}_1, ..., \mathcal{V}_I, \bar{\mathcal{V}}\right\}$. Note that the final set in this sequence contains all views $\mathcal{V}$, so the final posterior probabilities are guaranteed to satisfy all views, assuming that the views are feasible for the scenarios in R of course.

With the partitioning of the views established, the remaining question is which prior probability to use in the sequential processing. There are two natural choices in this regard. One is to use the original prior probability vector p, while the other is to use the updated posterior probabilities $q_0, q_1, ..., q_I$ associated with the updated parameters $\theta_0, \theta_1, ..., \theta_I$. This choice is exactly the difference between the two heuristics. Algorithm 1 (H1) uses the original probability vector p in all iterations, while Algorithm 2 (H2) uses the updated posterior probabilities $q_0, q_1, ..., q_I$, except the first iteration where p is used.

The two heuristics usually lead to similar final posterior probabilities q, but H1 is slightly better when measured by the relative entropy, as each relative entropy minimization step is against the original probability vector p, while H2 is usually slightly faster. Hence, H2 can be used if computation time is a crucial factor, while H1 is recommended for all other purposes.

Before presenting the sequential heuristics, some additional definitions must be established. For convenience, we define $q_{-1} = p$ and $\theta_{-1} = \theta_{prior}$. By $EP\left(\mathcal{V}^i, \theta, r\right)$ we mean that the EP method presented in Section 5.1 is applied to the set of views $\mathcal{V}^i$ using the parameter values in θ as fixed values when necessary and $r \in \mathbb{R}^S$ as the prior probability vector. Finally, $f\left(R, r\right)$ denotes the function that computes updated parameter values.

The two sequential heuristics are given by Algorithm 1 (H1) and Algorithm 2 (H2) below.

Algorithm 5.1 (H1)

for $i \in \{0, 1, ..., I\}$
  if $\mathcal{V}_i \neq \emptyset$, compute $q_i = EP\left(\mathcal{V}^i, \theta_{i-1}, p\right)$ and $\theta_i = f\left(R, q_i\right)$
  else $q_i = q_{i-1}$ and $\theta_i = \theta_{i-1}$
if $\bar{\mathcal{V}} \neq \emptyset$, compute $q = EP\left(\mathcal{V}, \theta_I, p\right)$
else $q = q_I$
return q

Algorithm 5.2 (H2)

for $i \in \{0, 1, ..., I\}$
  if $\mathcal{V}_i \neq \emptyset$, compute $q_i = EP\left(\mathcal{V}^i, \theta_{i-1}, q_{i-1}\right)$ and $\theta_i = f\left(R, q_i\right)$
  else $q_i = q_{i-1}$ and $\theta_i = \theta_{i-1}$
if $\bar{\mathcal{V}} \neq \emptyset$, compute $q = EP\left(\mathcal{V}, \theta_I, q_I\right)$
else $q = q_I$
return q

In the last five years, where I have worked with the sequential Entropy Pooling heuristics quite extensively, there has only been one extreme instance where the original heuristic of always using the prior value has given a better result. In all other cases, the sequential heuristics have given significantly better results. Readers are encouraged to test this out in practice on their own data and compare the performance gains through the relative entropy or effective number of scenarios as well as a visual assessment of the view/stress-test.

Below is a very simple case study using the H1 heuristic on daily S&P 500 and STOXX 50 data. The prior statistics are given in Table 5.1, while the posterior statistics for, respectively, the original Entropy Pooling heuristic and H1 are given in Table 5.2 and Table 5.3. We implement views on STOXX 50 expected return and the volatility of S&P 500. The relative effective number of scenarios is 85% for the original heuristic and 89.4% for H1. For a more advanced case study involving skewness, kurtosis, and correlations views, see Vorobets (2021).

	Mean	Volatility	Skewness	Kurtosis
S&P 500	0.0373%	1.2650%	-0.2522	15.0594
STOXX 50	0.0133%	1.3990%	-0.0936	10.6756

Table 5.1: Prior statistics for S&P 500 and STOXX 50 daily returns.

	Mean	Volatility	Skewness	Kurtosis
S&P 500	0.0373%	**1.5180%**	0.2786	14.0392
STOXX 50	**-0.6862%**	1.8115%	-1.8402	9.4778

Table 5.2: Posterior statistics for S&P 500 and STOXX 50 daily returns with original EP heuristic.

	Mean	Volatility	Skewness	Kurtosis
S&P 500	-0.3630%	**1.5180%**	-2.1073	13.5475
STOXX 50	**-0.6862%**	1.8322%	-2.2120	12.2508

Table 5.3: Posterior statistics for S&P 500 and STOXX 50 daily returns with H1 EP heuristic.

From the tables above, we clearly see the constraint that the original Entropy Pooling heuristic imposes by fixing the mean of S&P 500, which is marked with an underscore in Table 5.2. Views are marked with bold. Figure 5.2.1 shows the prior and posterior distributions for both S&P 500 and STOXX 50. A visual inspection reveals that the sequential heuristic H1 also gives results that look nicer and more realistic, without sudden kinks in unexpected areas of the distribution.

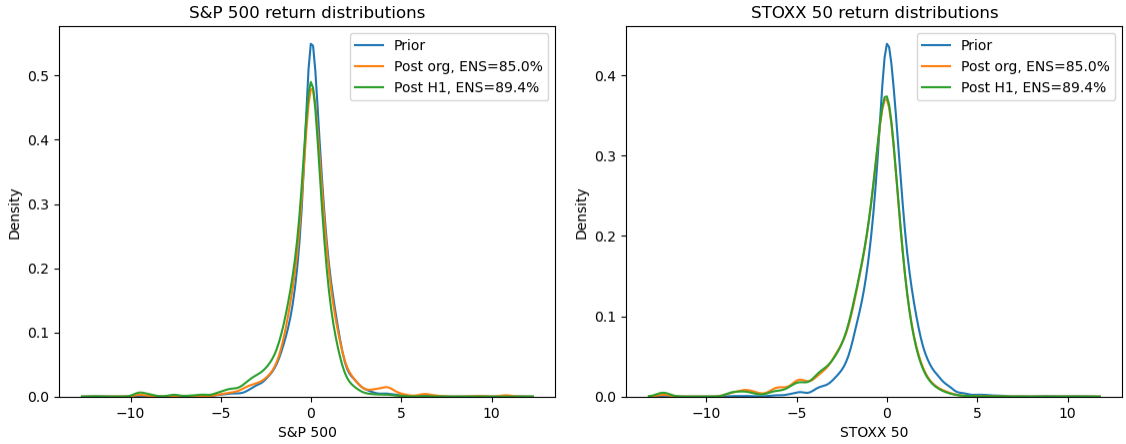

Figure 5.2.1: Prior and posterior distributions for S&P 500 and STOXX 50.

To understand the reasoning behind the sequential algorithms, the idea for both is to only fix parameters when we absolutely have to according to the class definitions $\mathcal{C} = \{\mathcal{C}_0, \mathcal{C}_1, \mathcal{C}_2, ..., \bar{\mathcal{C}}\}$. Hence, we allow views for the classes $\mathcal{C}_i$ to affect the parameters of the classes $\mathcal{C}_j$, $i < j$, for as long as we can. The difference between H1 and H2 is then whether we anchor the sequential updates in the latest intermediate posterior probability q_i or the prior p. You can think of H1 as looking into the future and then taking the information back to the prior to look further into the future, while H2 takes a step in the right direction and continues from there. As each iteration of H1 is against the prior probability, it is natural that it tends to give the best results, while the author currently cannot provide a proof that it will always be the case. Readers are encouraged to explore the methods and share their experiences.

With the above principles in mind, it hopefully becomes clearer why we characterized CVaR as a parameter belonging to $\bar{\mathcal{C}}$. There is simply no need for us to process it as a $\mathcal{C}_1$ parameter. In the algorithms that we use to find the optimal number of scenarios below the VaR, we quite clearly are likely to get a lower relative entropy if we allow views before $\bar{\mathcal{C}}$ to affect the final value of VaR before implementing the CVaR view.

The sequential algorithms do not only give us better results with lower relative entropies and higher effective number of scenarios. They also give us distributions that look more realistic. See for example the accompanying code to this section and another example in Figure 5.2.2 below. The example in Figure 5.2.2 implements several multi-asset views and compares the result of the original Entropy Pooling heuristic, which always fixes parameters to their prior values, and the H1 heuristic. These computations are performed using a proprietary implementation, so the accompanying code is not provided, but the views are exactly the same in the two cases. We simply see that the H1 heuristic gives significantly better results and more realistic-looking distributions.

There are some additional important benefits to the sequential heuristics. For example, they allow us to solve problems where we have a ranking view that must increase the expected value of a to make it higher than the expected value of b, and then implement views on higher moments for one or both of the assets. This is not possible with the original approach. It is also convenient that means are updated automatically, for example, in the case where we have views on two assets' returns and a view on the variance of a basket of the two. See Vorobets (2021) for more perspectives.

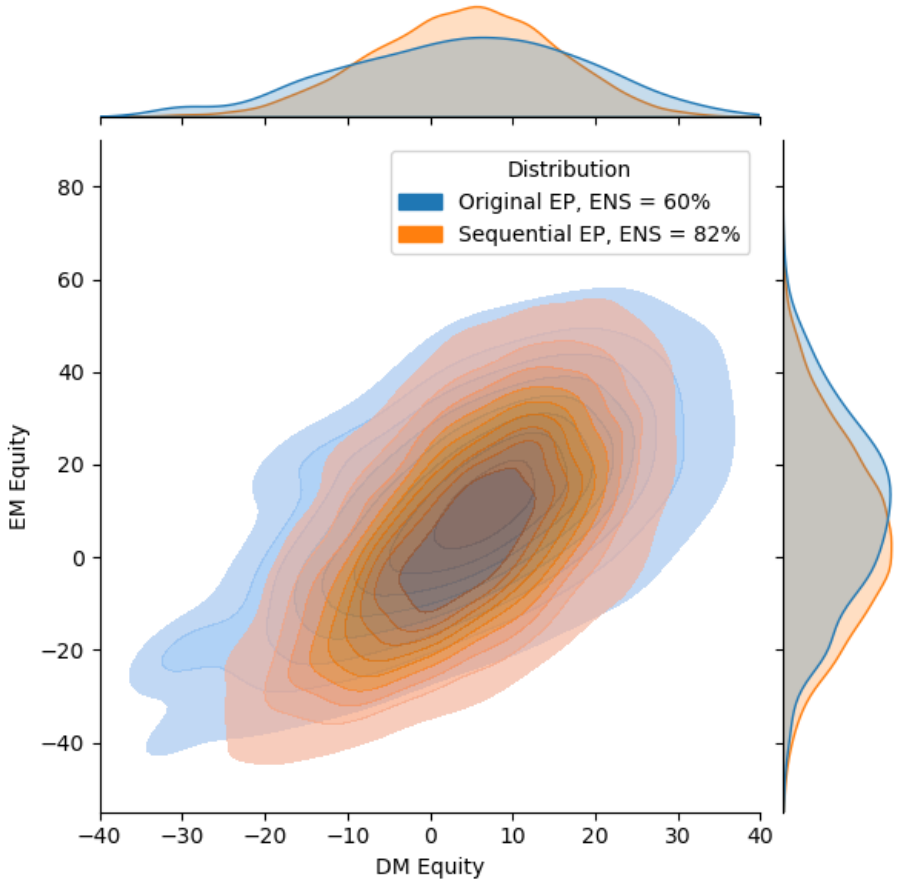

Figure 5.2.2: DM and EM equity posterior distributions.

5.3 View Confidences and Multiple Users or States

Until now, we have assumed that you have full confidence in your Entropy Pooling views and stress-tests. While this sounds like a strong assumption, it is actually how Entropy Pooling is used most of the time in practice. People simply want to see what will happen to the market given their views, with an understanding that it is all uncertain including the prior Monte Carlo distribution R. View confidences, however, add an additional nuance to the Entropy Pooling method, and they are treated in a natural probabilistic sense. The general framework is presented in the appendix of Meucci (2008a), while we will present it in a slightly more simple way as well as include additional perspectives.

While view confidences are not very frequently used in practice, probabilities of different parameter values or market states are used quite frequently. This perspective will be foundational for the Causal and Predictive Market Views and Stress-Testing framework in Section 5.4 below. We will make a separation between these two perspectives. To be precise, view confidences $c \in (0, 1]$ are specified over a logically consistent set of views. Weights assigned to individual users and probabilities of different states $u \in [0, 1]$ can be specified over potentially conflicting set of views. It is probably still confusing to you what the difference is, so there are illustrative examples below.

In the simplest case, you have a set of views V and some common confidence c in these views. Hence, it is natural to allocate this confidence to the posterior probability vector $q_{\mathcal{V}}$ and the rest to the prior, so the final posterior becomes

$$q = cq_{\mathcal{V}} + (1 - c)\, p.$$

Note that we require that the total confidence sums to 1. We cannot logically be more than 100% confident in our views.

What happens when we have multiple views with multiple confidences? For example, that the yearly expected return of S&P 500 should be 10% with 70% confidence, and that the yearly expected return of STOXX 50 should be 12% with 90% confidence. What does that mean? Obviously, we cannot just allocate 70% confidence to one posterior probability vector q_{V_1} and 90% to another posterior probability vector q_{V_2}. If we think more carefully about it, it means that with 70% confidence you believe that the expected return of S&P 500 should be 10% and the expected return of STOXX 50 should be 12%. With additionally $90\% - 70\% = 20\%$ confidence, you believe that the expected return on STOXX 50 should be 12%. And finally, you allocate the last $1 - 90\% = 10\%$ to the prior. Ordering and partitioning views in this way is what we will define as view confidence.

Note that we use the notation V_i instead of $\mathcal{V}_i$ to distinguish between partitioning of views according to the sequential Entropy Pooling heuristics and view sets in general. To define how view confidence is handled in general, imagine that we have a set of views $V = \{V_1, V_2, ..., V_I\}$ and their associated confidences $c = \{c_1, c_2, \dots c_I\}$, ordered from lowest confidence to highest confidence, i.e., $c_i \leq c_j$ for $i < j$. Let us then define $\bar{V}_i = \{V_j | j \geq i\}$, e.g., $\bar{V}_1 = V = \{V_1, V_2, ..., V_I\}$ and $\bar{V}_2 = \{V_2, V_3, ..., V_I\}$. Hence, the posterior probability vector with view confidences become

$$q = \sum_{i=1}^{I} (c_i - c_{i-1})\, q_i + (1 - c_I)\, p, \tag{5.3.1}$$

where $q_i = EP\left(\bar{V}_i, p\right)$ and $c_0 = 0$ for convenience. Note that $EP\left(\bar{V}_i, p\right)$ can be Entropy Pooling with the original heuristic or one of the sequential heuristics H1 or H2 from Section 5.2. The general formula (5.3.1) might seem complicated at first, but it is the same principle as describe in the simple case above with $c_1 = 70\%$ and $c_2 = 90\%$. It is a good exercise to verify that you can see this.

To conclude view confidences, we again underline that it must be a set of logically consistent views. For example, we cannot be 50% confident that a parameter is equal to 10 and 50% confident that it is equal to 12. We can, however, believe that there is a 50% probability that the parameter is equal to 10 and 50% probability that the parameter is equal to 12. This is assigning probabilities to states or weights to different users with conflicting views, which Entropy Pooling also can handle.

We can think of view confidences as the probabilities we assign within one user's views or one state, while the other probabilities and weights are assigned across states or users. This distinguishing will be important in Section 5.4 below, where we can define views with multiple confidences for each state and have conflicting view values across states. Handling of probabilities assigned to users and states is quite simple. We just need to make sure that they are positive and sum to one. How we determine these probabilities and view confidences is however more complex, but we have full flexibility.

5.4 Causal and Predictive Market Views and Stress-Testing

This section presents the Causal and Predictive Market Views and Stress-Testing framework from Vorobets (2023). Contrary to the article, you now have a better understanding of Entropy Pooling view confidences and the state probability weighting of posterior probability vectors. The causal and predictive framework is essentially a combination of Bayesian networks (BNs) and Entropy Pooling, where the Bayesian network acts as a causal joint view generator that additionally produces the state probabilities for each posterior probability vector through the joint view probabilities. EP is then used in the usual way to project each of the joint views over the market simulation represented by the matrix R. Hence, the framework naturally combines and leverages the strengths of the two methods.

The idea of using Bayesian networks for causal market analysis gained traction after the introduction by Rebonato and Denev (2011), while the idea of combining Bayesian networks with Entropy Pooling was first introduced by Meucci (2012b). Contrary to the framework introduced by Meucci (2012a), which discretizes the Monte Carlo simulation R into bins and applies EP to multivariate histograms, the framework in this book does not impose such limitations. Instead, we work with the fully general market representation (1.1.1).

Rebonato and Denev (2014) give a careful treatment of the thoughts behind building Bayesian networks for investment analysis, which have inspired the basic use cases of the framework in this book. However, the book's framework can be used in many creative ways that go well beyond the original perspectives, see Vorobets (2023) for several case studies. The presentation of the framework will naturally draw quite heavily on Vorobets (2023), while hopefully being easier to understand given the deeper understand of Entropy Pooling, view confidences, and state probability weighting.

5.4.1 An Introduction to Bayesian Networks

We will start with a high-level introduction to Bayesian networks that will be sufficient for our purposes, but it might be beneficial to find some introductory literature for readers who are completely unfamiliar with graphs and BNs. A BN is a directed acyclic graph (DAG) that represents a set of variables and their conditional dependencies. Each node represents a variable and each directed edge (arrow) represents a conditional dependency. If two nodes are not connected by a directed path, they are said to be conditionally independent.

Figure 5.4.1 shows a simple BN. In this BN, we say that A, B, and F are root nodes, while E, D, and F are leaf nodes. We call a node root if it has no incoming edges, while we call a node leaf if it has no outgoing edges. Note that a node can be both a root and a leaf node, which is the case for node F in this example. It follows immediately that any leaf node is conditionally independent of any other leaf node, because leaf nodes only have incoming edges. Additionally, we say that a node X is a parent of the child node Y if X has an outgoing edge to Y. For example, in Figure 5.4.1 B and A are parent nodes for C. Similarly, A is a parent node for the child node D, and C is a parent node for child node E.

The conditional dependence between two variables in a BN does not necessarily represent a causal relationship, but that is the usual assumption, and we will make it in this framework as well. An important realization is that causality is almost always an assumption in investment applications and

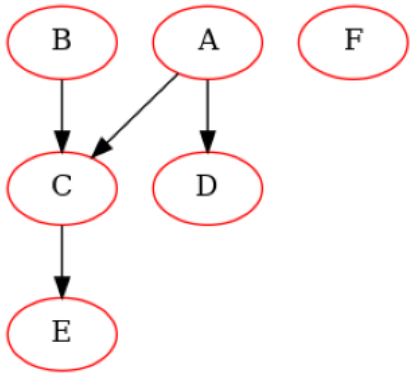

Figure 5.4.1: Simple Bayesian network illustration.

social sciences. It is usually non-trivial to prove causality, and it is also the hardest part of estimating BNs. Once the conditional dependencies are specified, estimating the probabilities of a discrete BN is straightforward. In this framework, we suggest that you specify both the conditional dependencies and probabilities for the BN. This gives a high level of flexibility, allowing you to hypothesize about and impose future causal relationships that are not necessarily strongly present in historical data. However, a BN where both the causal structure and probabilities are estimated based on historical data will work just as well in this framework.

Specifying the probability tables of a discrete BN might seem as a daunting task at first, but it is usually easier than one expects if the task is approached in a structured way. The following four step procedure works well in practice:

1. Specify the relevant variables (nodes) of the BN,
2. Specify the causal relationships (edges/arrows) between the variables,
3. Infer the size of the conditional probability tables,
4. Populate the conditional probability tables one at a time.

In all of these steps, it of course helps to have an elegant implementation of the BN technology to keep track of and manage all the information. Especially in step 3, it is convenient to let the tables be auto-generated. For step 4, it is also convenient to be able to specify probability ranges or leaving probabilities unspecified with some method helping you to estimate a probability based on the defined BN structure. A suggestion is to use maximum entropy for inferring missing values as described by Corani and Campos (2015).

With a high-level understanding of BNs, the question remains how to use them for investment analysis. The idea is well-described in Rebonato and Denev (2014), so it is only briefly summarized here: we want the relevant assets, returns, or factors to correspond to the leaf nodes of the BN. For example, for an asset allocation investor, the leaf nodes E, D, and F in Figure 5.4.1 could be variables like real rate, inflation, and risk premium, while the other nodes are variables that causally affect the distribution of the real rate, inflation, and risk premium, see the case study in Section 5.4.4 below.

Letting $X = (X_1, X_2, \ldots, X_N) \in \mathbb{N}^N$ denote the N-dimensional vector of random variables/nodes in a discrete BN, the joint probability can be computed using the well-known chain rule factorization

$$\mathbb{P}(X_1, X_2, \ldots, X_N) = \prod_{i=1}^{N} \mathbb{P}(X_i | pa(X_i)),$$

where $\mathbb{P}(X_i | pa(X_i))$ denotes the probability of X_i conditional on its parents. Since root nodes do not have any parents, $\mathbb{P}(X_i | pa(X_i)) = \mathbb{P}(X_i)$.

Letting $\mathcal{LN} \subseteq \{1, 2, ..., N\}$ denote the leaf node indices, the framework in this article mostly focuses on the joint distribution of the leaf nodes X_i, $i \in \mathcal{LN}$, given by

$$\mathbb{P}(\{X_i | i \in \mathcal{LN}\}) = \prod_{i \in \mathcal{LN}} \mathbb{P}(X_i | pa(X_i)). \tag{5.4.1}$$

The total number of joint leaf node probabilities is given by

$$J = \prod_{i \in \mathcal{LN}} S_i,$$

where S_i is the number of states for node $i \in \mathcal{LN}$.

5.4.2 Integrating Entropy Pooling

The presentation so far has brought us to a point where we understand how to specify a discrete Bayesian network. These networks can be interesting to analyze on their own, but they first become really interesting when we are able to project the discrete joint states of the leaf nodes over general Monte Carlo simulations R. This is what we use Entropy Pooling for.

It is strongly recommended to use the sequential Entropy Pooling refinements as presented in Section 5.2 above. However, it is important to underline that the framework also works with the original EP heuristic of always using prior parameter values when necessary, i.e., there is no reliance on any aspects of the sequential refinements from Section 5.2. They just usually give significantly better results.

We will denote the joint leaf node probabilities from (5.4.1) by p_j, $j = 1, 2, \ldots, J$, and use them as weights for the associated EP posterior probability vectors. These joint leaf node probabilities are natural weight candidates for posterior EP probability vectors because $p_j \in [0, 1]$ and $\sum_{j=1}^{J} p_j = 1$, which implies that the sum of the elements in $q = \sum_{j=1}^{J} p_j q_j \in \mathbb{R}^S$ is one for any set of valid EP posterior probability vectors $q_j \in \mathbb{R}^S$. The EP posterior probability vectors q_j are computed using EP for each of the joint leaf node events. Formally, each of the leaf node states $x_i = \{x_{i,1}, x_{i,2}, ..., x_{i,S_i}\}$ is mapped into EP views $v_i = \{v_{i,1}, v_{i,2}, ..., v_{i,S_i}\}$, $i \in \mathcal{LN}$, and combined using a Cartesian product over all v_i into J joint leaf node EP views.

The above weighting is an implementation of the state probability weighting approach. Specifically, the joint leaf node probabilities correspond to the u's from Section 5.3. Hence, the BN effectively helps us to generate these weights. View confidences c are directly specified for each leaf node state and therefore reflected in the posterior probability vectors q_j, $1, 2, \ldots, J$.

5.4.3 Additional Perspectives

The framework is introduced with one variable for each leaf node, but it is actually much more flexible than that because we can use fully general EP views for each of the leaf node states. The only limitation is that a leaf node cannot contain views that contradict the views from another leaf node. For example, you cannot have a state in one leaf node with a view that some variable should be equal to 10, while you have a view in another leaf node that this variable should be equal to 20. You must implement these views using the same node with two different states to ensure logical consistency in the joint views. This is the only restriction.

It can be tempting to interpret the framework in the following way: the causality comes from the BN, while the predictiveness comes from EP. This is a reasonable way to think about the framework, but it is strictly speaking not correct. It is clear that the BN defines causal relationships, this is one of its main features. However, there is also an element of predictiveness when one conditions on realizations of the variables, making some events more and less likely. Similarly with EP, where the predictiveness aspect is clear, while there can also be causal elements in the market simulation, for example, interest rates causing changes in bond prices.

It is important to realize that assuming the causal relationships in the BN introduces an EP view, even when we do not condition on realization of relevant variables. How much this view deviates from the prior can be assessed by the relative entropy between the prior and unconditional posterior. If the prior is uniform in scenario probabilities, the (relative) effective number of scenarios can also be used to assess how concentrated scenario probabilities are.

Although the BN defines a causal structure, it can also be used to answer non-causal questions. For example, if one has defined a BN where a central bank's decision depends on inflation and employment, one can condition on the central bank being hawkish and compute the probability of high inflation or low unemployment, i.e., answering the question of how the distribution of these variables should be in order for the central bank to be hawkish. These results can then be compared to what is implied by the market and used for taking positions.

A full implementation of the framework does not only have a good interface for BNs combined with maximum entropy for estimating partially specified probabilities, but also an integration of sequential EP presented in Section 5.2 with support for rank views, view confidences, and CVaR views. Building such an implementation is a very daunting task that should probably only be attempted by the most determined. However, once successful the framework allows investment and risk managers to perform sophisticated market views and stress-testing analysis at a level that is way beyond current standards.

For case studies using the framework presented in this section, see Vorobets (2023). There is a simple Bayesian network example available as open-source code using the fortitudo.tech Python package. There is even a video walkthrough of the article and the accompanying code. This book will purposefully not replicate more from the article than what has already been done, except present some perspectives on asset allocation cases in Section 5.4.4 below, which are interesting and easy to relate to for most people.

We conclude this section by stating that the framework is incredibly flexible and powerful in the hands of skilled quantamental investment practitioners, who are usually very excited about its possibilities.

5.4.4 Asset Allocation Case Study

A popular case that most investment managers can relate to is an analysis of the macroeconomy and translation into portfolio P&L. Specifically, consider the Bayesian network given in Figure 5.4.2. This is a plain vanilla application of the framework, where each leaf node consist of a key macro risk factor. Many asset allocation investors think about the risk in their portfolios from a perspective similar to this one.

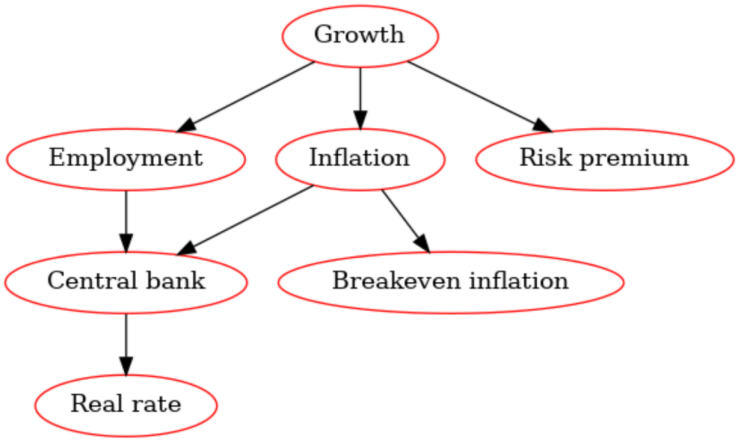

Figure 5.4.2: Asset allocation Bayesian network.

Figure 5.4.3 shows the result of a stagflation and rate hike stress-test using this Bayesian network for a low risk portfolio and its tail risk hedge. The case uses a proprietary implementation and is therefore not available in the accompanying code.

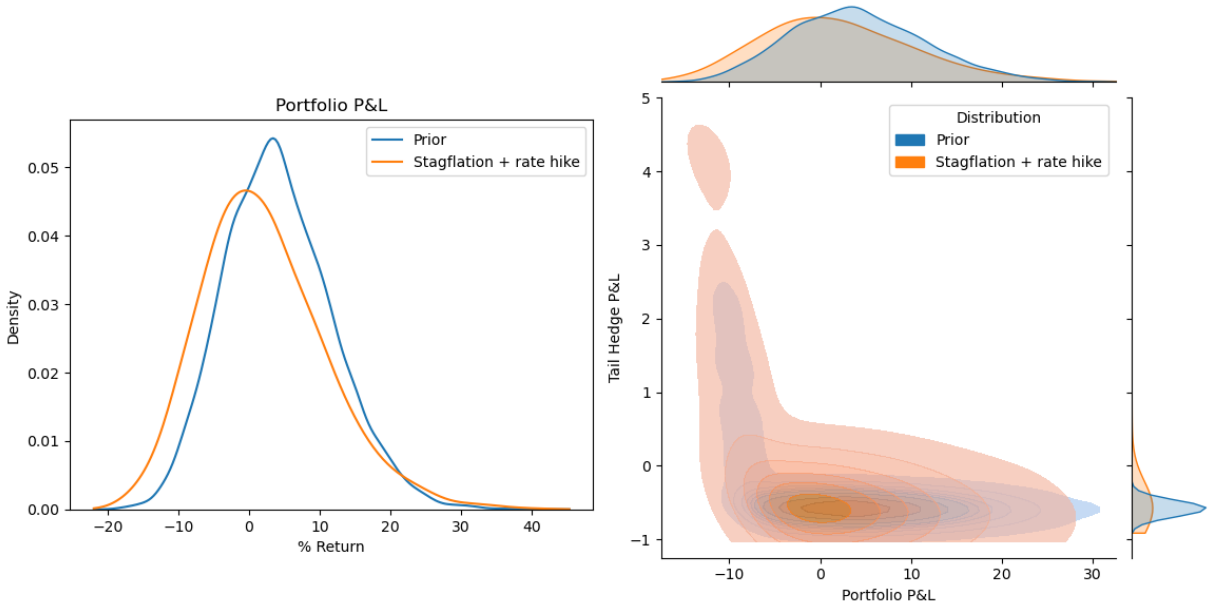

Figure 5.4.3: Portfolio and tail hedge prior and posterior P&L.

Chapter 6

Portfolio Optimization

This chapter focuses on portfolio optimization, in particular CVaR portfolio optimization for fully general Monte Carlo simulations R and associated probability vectors p and q, i.e., the starting point presented in (1.1.1). An elegant aspect of CVaR optimization in practice is that it operates directly on the market simulations R and the associated probability vectors p and q. Hence, it does not matter how complex our simulations or market views and stress-tests are, CVaR will give us meaningful results.

Section 6.1 presents the portfolio management framework that we work with throughout the book, first documented by Vorobets (2022a). By making a separation between relative exposures e and relative market values v, we can handle derivative instruments as easily as plain vanilla cash instruments such as stocks and bonds.

Section 6.2 compares CVaR optimization to the traditional variance optimization, showing how CVaR optimization problems can be solved with linear programming and giving an understanding of why CVaR optimization is much harder to implement in practice than variance optimization. However, fast and stable algorithms that make CVaR optimization practically feasible exist.

Section 6.3 presents portfolio optimization problems with multiple risk targets, in particular a risk target for overall portfolio risk $\mathcal{R}_{target}$ and a risk target for deviations from a benchmark portfolio $\mathcal{R}_{TE}$. The joint optimization of these two risk targets introduce important trade-offs between the risk of the benchmark and the risk of the deviations from the benchmark. This analysis is initially presented for variance optimization, because it makes it easy for us to build intuition by representing diversification with the correlation, while the results are eventually generalized to CVaR.

Section 6.4 introduces parameter uncertainty into the portfolio optimization problem. We focus in particular on resampled optimization with the perspectives introduced by Kristensen and Vorobets (2024). The section introduces a new Exposure Stacking generalization, which is coined Resampled Portfolio Stacking. While derivatives are easy to handle in general portfolio management, they introduce significant complexities when it comes to portfolio optimization with parameter uncertainty. Conveniently, Entropy Pooling helps us solve these problems as explained by Vorobets (2024).

Section 6.5 introduces a new method for intelligent portfolio rebalancing, which is based on the Resampled Portfolio Stacking perspectives and presented for the first time in this book. Portfolio rebalancing is an essential part of portfolio management, but it is usually handled in an ad hoc manner without a framework for thinking about the rebalancing problem. This book suggests an improvement.

6.1 Exposures and Relative Market Values

The most well-known portfolio in investment management is perhaps the long-only portfolio characterized by the constraints $\sum_{i=1}^{I} w_i = 1$ and $w_i \geq 0$, with w_i representing the weight of the total portfolio value invested in asset i. There are no issues with this characterization for simple portfolios consisting of only cash instruments and in fact being long-only. However, modern portfolios are increasingly utilizing derivatives, making the traditional framework insufficient.

Not much attention has been given to generalize the traditional framework to portfolios that invest in derivative instruments. As a consequence, practitioners usually try to treat derivative portfolios in the same way as portfolios investing in cash instruments only. A key quantity in this attempt is the offsetting cash position, introduced to make everything sum to one by offsetting the leverage introduced by the derivative instruments.

This book argues that although the offsetting cash approach might seem like a natural first guess by trying to somehow replicate the derivative exposure, it is in fact unnecessarily complicated. It is much easier and fairly straightforward to treat the derivative positions as they are by properly separating exposure/notional and market value/price. The exposures are then used for all aspects of portfolio management, while prices are used for elementary bookkeeping, see Vorobets (2022a) for more perspectives.

To understand the need for separating between exposure/notional and market value/price, consider a futures contract. The futures contract has initial market value of zero, but a positive or negative exposure depending on the direction of the position. It follows immediately that it is meaningless to use the market value of the futures contract as a measure of exposure, because it will appear as if the contract has no effect on the portfolio's future P&L distribution. Hence, it is important to realize that it is the exposure that determines the portfolio's future P&L distribution, while the market value is used for elementary bookkeeping.

The usual self-financing constraint found in, e.g., portfolio optimization problems reads something like

$$\sum_{i=1}^{I} w_i = \iota^T w = 1, \tag{6.1.1}$$

with ι being an I-dimensional vector of ones. With the above reasoning in mind, this book argues that the self-financing constraint should be more accurately expressed as

$$\sum_{i=1}^{I} v_i e_i = v^T e = 1, \tag{6.1.2}$$

with v being an I-dimensional vector of relative market values (the value of the derivative instrument with exposure/notional set to one), and e being an I-dimensional vector of exposures relative to portfolio value.

For plain vanilla cash instruments like stocks and bonds, market value and exposure are the same. Hence, $v = \iota$ and $e = w$. This illustrates how (6.1.1) is in fact a special case of (6.1.2). Now imagine that we are allowed to invest in at-the-money forward European put and call options. The price of these options with one year to expiry, zero interest rate, zero dividends, implied volatility of 15%, and

a notional of $E_i = 100$ is approximately $V_i = 5.98$, see the accompanying code to this section. So what is v_i in this case? That is easily calculated as $v_i = \frac{V_i}{E_i} = \frac{5.98}{100} = 0.0598$.

The vector of relative market values v makes it easy for us to calculate the value V_{pf} of any portfolio characterized by the exposures vector $E \in \mathbb{R}^I$, i.e.,

$$v^T E = V_{pf}. \tag{6.1.3}$$

To compare (6.1.3) to (6.1.2), we multiply (6.1.2) by V_{pf} to get

$$v^T e V_{pf} = V_{pf}. \tag{6.1.4}$$

From (6.1.3) and (6.1.4), it follows that $eV_{pf} = E \Leftrightarrow e = \frac{E}{V_{pf}}$. This illustrates precisely how e is the vector of exposures relative to portfolio value V_{pf}.

So far, exposure has not been explicitly defined. The recommendation is to use the notional for both derivative and cash instruments. If the notional is not directly given in the term sheet of the derivative instrument, the necessary information (e.g. contract unit) for computing the notional is. For plain vanilla cash instruments, notional is simply equal to the market value/price. Hence, there is no need for complicated computations when exposure is measured by the notional.

Another benefit of using the notional is that it will often correspond to the cash equivalent of the position. For example, an equity futures contract with a notional exposure of \$100 gives the same exposure to price movements as a \$100 cash position in the underlying, making it easy to work in conceptually similar ways for these instruments. Conclusively, notional is often the best conceptual approximation to the conventional portfolio weight, and it is easy to compute.

The most important nuance introduced by this portfolio management framework is the clear separation of portfolio weights given by $v_i e_i$ and portfolio exposures given by e_i. Hence, the sum of the weights is still required to be one as in the conventional framework (6.1.1), while there are no a priori restrictions on the sum of exposures. This underlines the main point that weights are used only for elementary bookkeeping, while exposures are relevant for the future portfolio P&L distribution.

Since we are usually interested in optimizing the portfolio's return and decompose its risk, it is interesting to examine how the return equation looks for this case. Relative P&L over the next period is naturally defined as $\Delta v_i = \frac{V_i - V_{i,0}}{E_{i,0}}$, with $V_{i,0}$ being the initial value and $E_{i,0}$ the initial exposure of position i. Summing over the products of relative P&L Δv_i and relative exposures $e_i = \frac{E_{i,0}}{V_{pf}}$, it follows that the portfolio percentage return is given by

$$r_{PF} = \sum_{i=1}^{I} \Delta v_i e_i = \sum_{i=1}^{I} \frac{V_i - V_{i,0}}{E_{i,0}} \frac{E_{i,0}}{V_{pf}} = \frac{1}{V_{pf}} \sum_{i=1}^{I} V_i - V_{i,0}. \tag{6.1.5}$$

Note that (6.1.5) is exactly how portfolio return is conventionally computed for cash portfolios. Hence, instead of percentage returns and portfolio weights, we can use the relative P&L Δv_i and exposures e_i relative to portfolio value V_{pf} for portfolio optimization and risk decomposition.

For realistic optimization and rebalancing applications, the self-financing constraint (6.1.2) should

include transaction costs that are payable at the time of the trade. In that case, (6.1.2) becomes

$$v^T e + TC\,(e - e_0) = 1,$$

where $TC\,(e - e_0)$ is the transaction cost function (implicitly normalized by portfolio value V_{pf}) for the turnover $e - e_0$, with $e_0 \in \mathbb{R}^I$ being the vector of initial exposures relative to portfolio value.

Using the traditional cash-based framework in (6.1.1), long-short portfolios are usually characterized by the constraint

$$\sum_{i=1}^{I} w_i = 0.$$

However, it is rarely the case (if ever) in practice that one is allowed to take margin free risk, e.g., borrow a stock to sell it in the market and use all of the proceeds to take a long position. Usually, the portfolio must be collateralized in some way. The value of the collateral/margin is thus the value of the portfolio, and the long-short constraint is therefore more correctly implemented by the requirement that

$$\sum_{i \in LS} e_i = 0,$$

where $LS \subsetneq \{1, 2, ..., I\}$ is the subset of instruments that can be used for the long-short exposures.

6.2 CVaR vs Variance Optimization

Investors are rarely worried about their returns being too high or their portfolio's return distribution being skewed too much to the upside. Both of these properties should indeed be desirable under normal circumstances where there are no strange incentives that affect investor preferences for higher risk-adjusted returns. The best examples of positive skewness being a desirable property are perhaps put and call options, where investors have historically been willing to pay a volatility risk premium above the fair value for a long position in these instruments, see Section 2.2. However, mean-variance optimization treats upside and downside deviations as equally undesirable, clearly going against this important principle.

Rockafellar and Uryasev (2000) were the first to introduce a practical method for solving CVaR portfolio optimization problems. They show that solutions can be found using linear programming for fully general Monte Carlo distributions like (1.1.1). While linear programming sounds nice and easy, we will see in Section 6.2.1 below that it is nontrivial to get good performance, and that the original formulation is probably too slow and unstable for practically relevant CVaR optimization, which includes resampled parameter uncertainty as presented in Section 6.4 below. However, fast and stable algorithms that exploit the structure of the problem exist.

Proposition 1 of Rockafellar and Uryasev (2000) allows us to determine when CVaR and variance optimization coincide. If the return distribution is elliptical (for example normal or t-distribution), results will always coincide when the expected return constraint is binding. Additionally, mean-CVaR optimization results always coincide with mean-variance optimization when demeaned portfolio returns are used and instrument returns are jointly normal, see Vorobets (2022b) for more information and

a case study verifying this result. Hence, you lose nothing from using CVaR in the oversimplified textbook case, while you gain a lot when it comes to real-world investment analysis.

If we use demeaned returns when computing CVaR, it becomes more comparable to other deviation measures, e.g., variance and (lower semi-)absolute deviation. Since most portfolios in practice have a positive expected return, it will also give us more conservative risk estimates, which in practical cases is arguably better than underestimating the risk. Readers can freely decide what they believe is best for their purposes, while a formal treatment of the differences between these two choices is given by Rockafellar, Uryasev, and Zabarankin (2006).

Besides giving meaningful results for fully general Monte Carlo distributions (1.1.1) and focusing on investment tail risk, what are some other benefits of using CVaR? First of all, contrary to VaR, CVaR is a coherent risk measure, respecting the diversification principle in (1.3.1), which is arguably a desirable feature for an investment risk measure, see Artzner et al. (1999). From a less technical perspective, it becomes harder to hide significant risks below the VaR value because the mean is sensitive to outliers. Hence, CVaR is steadily overtaking VaR to become the preferred investment tail risk measure among both market makers and investment managers.

While all of the above should be sufficient to stop using variance and start using CVaR for investment practitioners, there is an additional important aspect to CVaR. It is much easier to interpret than variance for investment clients and other nontechnical stakeholders. For example, if you ask a regular person which risk they can tolerate as the expected loss in the worst year out of 10 (90%-CVaR), it is probably much easier for them to answer this question than which level of expected squared deviations from the mean they can tolerate.

As an investment practitioner, you might have developed some sense of what 5% and 10% yearly volatility looks like, but regular people do not have this sense. The ease of interpretability is very important for nontechnical people on boards and asset management clients. Finally, if you tell a regular person that your portfolio optimization method focuses as much on minimizing positive deviations from the mean as negative deviations, you probably will see someone who looks very strangely at you and tells you not to do that.

It is mainly finance and economics academics, or people who otherwise feel reputationally invested in the mean-variance system they have used for the past many years, who continue to defend mean-variance analysis despite its obvious and severe deficiencies. One of the seemingly scientific arguments that these people sometimes present is that "CVaR implies risk neutrality below the VaR value". This argument is based on a utility theory definition of risk aversion, which in itself is known to be a very poor representation of how people actually behave.

That utility theory is a good representation of reality seems to be a rumor among finance and economics academics, which they take for granted without being able to share any studies that actually show this. For a careful treatment of the issues with utility theory, see D. Friedman et al. (2014). Interestingly, D. Friedman et al. (2014) conclude that people seem to behave more according to a linear utility function subject to constraints, which fully agrees with practical CVaR optimization and analysis.

If utility theory was a good representation of how people actually behave, almost anyone who was presented with the CVaR risk measure should feel a deep sense of unease by the "risk neutrality

below the VaR". However, people's reaction tend to be quite the opposite, where they feel that it represents their preferences for avoiding losses quite well. If you tell them that minimizing the upside is a byproduct of your optimization method, you will probably see them become very uneasy. Hence, the reality is that mean-variance optimization does not represent what people actually want to do. Interestingly, Markowitz (1952) already acknowledged this and argued that the focus should be on the downside, but this was practically unthinkable with the technology that was available at the time.

Perhaps even more intriguing, Harry Markowitz told in an interview that he did not use mean-variance to manage his own portfolio. He used the $1/N$ heuristic. While we will rely on heuristics when it comes to resampled portfolio optimization in Section 6.4, the methods and framework recommended in this book are actually used to manage the author's own money.

We will not delve more into utility theory or the attempts to justify the continued use of mean-variance analysis. The author's hypothesis is that it is mainly driven by reputational investments in this theory and the ease of implementing mean-variance optimization compared to mean-CVaR optimization. We will not rely on any aspects related to utility theory and simply note that CVaR seems to be a good representation of the investment risk that people want to avoid. For a more detailed comparison between variance and CVaR as investment risk measures, see Vorobets (2022b).

6.2.1 Solving CVaR Problems

A practical way to solve CVaR problems was first introduced by Rockafellar and Uryasev (2000), while equivalent formulations of the risk-adjusted return objectives and general scenario probability weighted Monte Carlo distributions (1.1.1) are analyzed by Krokhmal, Palmquist, and Uryasev (2002). The CVaR theory and the justification for solving the problem using linear programming through a discretization and linearization are quite mathematically advanced. It is intentionally not replicated here. We simply focus on the problem formulation using the notation from this book.

Focusing initially on the case where we want to minimize α-CVaR, potentially subject to an expected return constraint, the linear programming problem is given by

$$(VaR^\star, y^\star, e^\star) = \underset{VaR,y,e}{\operatorname{argmin}}\ VaR + \frac{1}{1-\alpha} p^T y$$

subject to

$$\begin{aligned} y_s &\geq -R_s e - VaR && \forall s \in \{1, 2, \ldots, S\}, \\ y_s &\geq 0 && \forall s \in \{1, 2, \ldots, S\}, \\ \mu_{target} &\geq \mu^T e, \\ v^T e &= 1, \\ e &\in \mathcal{E}. \end{aligned}$$

In the above, $y = (y_1, y_2, \ldots, y_S)^T$ is a vector of auxiliary variables, while R_s represents row s from the Monte Carlo simulation R, and p is the associated joint probability vector from (1.1.1). Finally, $\mu = R^T p$ is the vector of expected relative P&L's, while v is the vector of relative market values introduced in Section 6.1. We use $\mathcal{E}$ to represent the set of linear (in)equality constraints on the

portfolio exposures e, forming a convex polyhedron.

To focus on the essence of the problem, we do not include transaction costs, which require an additional introduction of auxiliary variables $e^+ \geq 0$ and $e^- \geq 0$, representing the buys and sells, in addition to the constraints $e = e_0 + e^+ - e^-$ with $e^+e^- = 0$, and an extension of the self-financing constraint to $v^T e + TC\,(e - e_0) = 1$ as explained in Section 6.1. We would also need to subtract the transaction costs from the expected return as explained by Krokhmal, Palmquist, and Uryasev (2002).

From the CVaR optimization problem, we immediately notice that it requires an introduction of S auxiliary variables y_s, for $s = 1, 2, \ldots, S$, in addition to $2S$ extra constraints. Hence, even though it can be formulated as a linear programming problem, it is potentially very high-dimensional and introduces a trade-off between computation time and approximation quality due to the discretization and linearization of the objective function. In practice, the original formulation of the CVaR optimization problem is significantly slower and less stable than the traditional mean-variance optimization. This fact is especially inconvenient as portfolio optimization in practice almost always needs to include parameter uncertainty as presented in Section 6.4 below.

Using the original formulation, solving CVaR problems subject to CVaR constraints is as straightforward as solving problems with expected return constraints. Krokhmal, Palmquist, and Uryasev (2002) show that these problems simply require us to change the optimization objective to

$$(VaR^\star, y^\star, e^\star) = \underset{VaR,y,e}{\operatorname{argmin}} \; -\mu^T e = \underset{VaR,y,e}{\operatorname{argmax}} \; \mu^T e$$

and add the CVaR formulation to the constraints

$$VaR + \frac{1}{1-\alpha} p^T y \leq CVaR_{target},$$

while of course removing the expected return constraint from our initial formulation above.

Krokhmal, Palmquist, and Uryasev (2002) show that we solve equivalent problems in both cases, so it does not matter which method we use for the same combination of risk and return. This is a feature that we will use in Section 6.3.1 below. While solving the problem with a single CVaR target is straightforward, albeit slow and potentially unstable, solving the problem with multiple CVaR targets becomes much more complicated. Imagine for example if we had two CVaR targets that would require $2S$ auxiliary variables and $4S$ constraints to solve the problem. Whatever issues we have with speed and stability from just one CVaR target are guaranteed to be amplified in this case.

The most sophisticated investment managers usually want to solve CVaR optimization problems with multiple risk constraints as presented in Section 6.3 below. Fortunately, there exist fast and stable algorithms to solve CVaR problems. These are, however, hard to discover and very complex to implement. Due to their proprietary nature, they cannot be shared in this book, while a fast and semi-stable version of an algorithm solving the problem with a return target can be found in the fortitudo.tech Python package.

It is left as an exercise for readers to really test out whether they understand the CVaR problem formulation and solve a problem for a portfolio with a 5% return target and 25% individual upper bounds on cash instruments as well as -50% to 50% bounds for derivative instruments, i.e., solving the prior and posterior optimization problems from Vorobets (2022a). See the accompanying code to

this section, which gives you results for these problems using the fortitudo.tech package.

6.3 Risk Budgeting and Tracking Error Constraints

The easiest way to generate a higher return is just to take more risk. For example, instead of a futures position with a notional of \$100, you increase the exposure to \$200. That does not require skillful investment management, only a bit more margin. It is especially easy to take more "market beta" risk, so it is not something that would be worth paying an asset manager for.

On the other hand, building portfolios with a good balance between tail risk and return requires a lot of skill and is worth paying for. Throughout this book, we assume that you as an investment manager are determined to maximize the risk-adjusted return and not simply maximize the return as many bonus schemes unfortunately incentivize. Hence, good portfolio construction is focused on maximizing the expected return subject to risk constraints.

Following Vorobets (2022b), we will start with an analysis of the portfolio variance, because it is easy to understand and implement, while going into more details than the article and having an Entropy Pooling case study. The starting point of our analysis is a decomposition of the portfolio return into a return on a benchmark/strategic/long-term portfolio and a tracking error/tactical/short-term portfolio

$$r_{PF} = r_{BM} + r_{TE}. \tag{6.3.1}$$

This is usually how most investment managers think about the risk and return of their portfolios, while it can of course be partitioned in many other ways. To make it easier for us to analyze the main ideas, we stick to the usual decomposition (6.3.1).

The return decomposition above implies that the portfolio variance can be expressed as

$$\sigma_{PF}^2 = \sigma_{BM}^2 + \sigma_{TE}^2 + 2\rho_{BM,TE}\sigma_{BM}\sigma_{TE}. \tag{6.3.2}$$

A meaningful risk budgeting workflow is to initially set the benchmark/long-term portfolio (in many cases it is externally given to the investment manager) and then risk budget using σ_{PF} and σ_{TE}. An important but perhaps trivial observation is that once the benchmark risk σ_{BM} is fixed, there are only two degrees of freedom among the remaining parameters σ_{PF}, σ_{TE}, and $\rho_{BM,TE}$.

Several interesting insights can be extracted from analyzing (6.3.2). For example:

1. The risk contribution from the tracking error portfolio is amplified by $\rho_{BM,TE}$ since $\frac{\partial \sigma_{PF}^2}{\partial \sigma_{TE}} = 2\sigma_{TE} + 2\rho_{BM,TE}\sigma_{BM}$,
2. A tracking error portfolio reduces the overall risk σ_{PF} when $\sigma_{TE}^2 + 2\rho_{BM,TE}\sigma_{BM}\sigma_{TE} < 0 \Leftrightarrow \frac{\sigma_{TE}}{\sigma_{BM}} < -2\rho_{BM,TE}$, which can only happen when $\rho_{BM,TE} < 0$.

The practical implication of 1. is that tracking error underestimation affects portfolio risk σ_{PF} more adversely the higher the correlation $\rho_{BM,TE}$ is, i.e., you are more exposed to risk overshoots due to estimation error related to σ_{TE}, e.g., if $\sigma_{TE} = 3\%$ while you estimate $\hat{\sigma}_{TE} = 2\%$. Figure 6.3.1 below illustrates this for various correlations $\rho_{BM,TE}$ and tracking errors σ_{TE}. This figure clearly illustrates that the risk overshoots increase with the correlation. See the accompanying code to this section for

the details of how these computations have been performed, and think about how the graph will look for $\rho_{BM,TE} = 1$. You can use the code to validate your intuition.

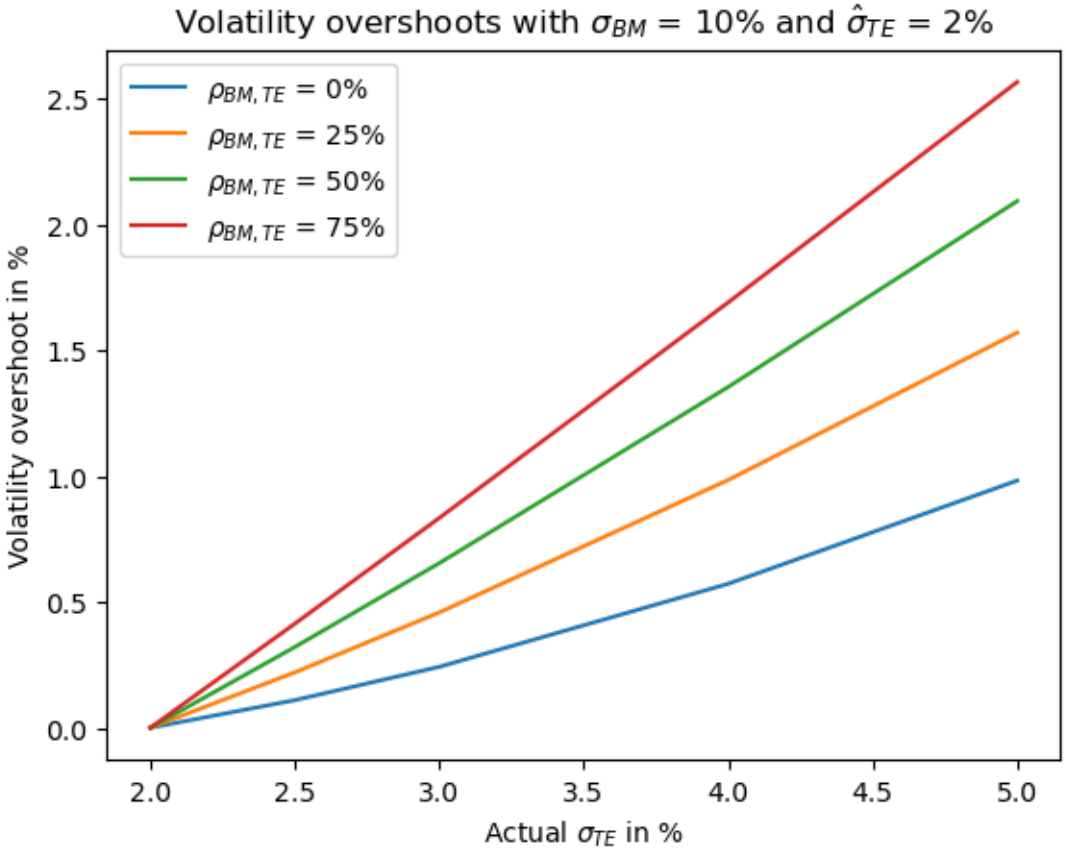

Figure 6.3.1: Risk overshoot for $\sigma_{BM} = 10\%$, $\hat{\sigma}_{TE} = 2\%$, and various $\rho_{BM,TE}$.

There is a slightly more subtle second-order effect from the fact that you are probably also more likely to underestimate the benchmark risk σ_{BM} simultaneously with underestimating the tracking error risk σ_{TE} if the two portfolios have a high correlation $\rho_{BM,TE}$. Figure 6.3.2 below repeats the analysis from Figure 6.3.1 using Entropy Pooling to stress-test the tracking error σ_{TE} for various correlation levels. We clearly see that the risk overshoot becomes significantly higher not only due to the ceteris paribus effect from a higher tracking error than we have estimated, but also due to a higher benchmark risk. Hence, maintaining a low correlation $\rho_{BM,TE}$ between the benchmark and tracking error portfolios is a very important part of skillful portfolio construction in practice.

The practical implication of 2. is that there is a trade-off between standalone tracking error risk σ_{TE} and the correlation $\rho_{BM,TE}$, i.e., a lower correlation $\rho_{BM,TE}$ allows for a higher standalone tracking error σ_{TE} without increasing the overall portfolios risk σ_{PF}. Hence, diversification can happen within and between the benchmark and tracking error portfolios, with $\rho_{BM,TE}$ being a proxy for the degree of diversification between the two portfolios. Note also that the trade-off between σ_{TE} and $\rho_{BM,TE}$ depends on the ratio $\frac{\sigma_{TE}}{\sigma_{BM}}$. This observation explains why the impact of FX hedging on overall portfolio

risk is more significant for a low risk portfolio of short-term investment grade bonds than a high risk portfolio of equities.

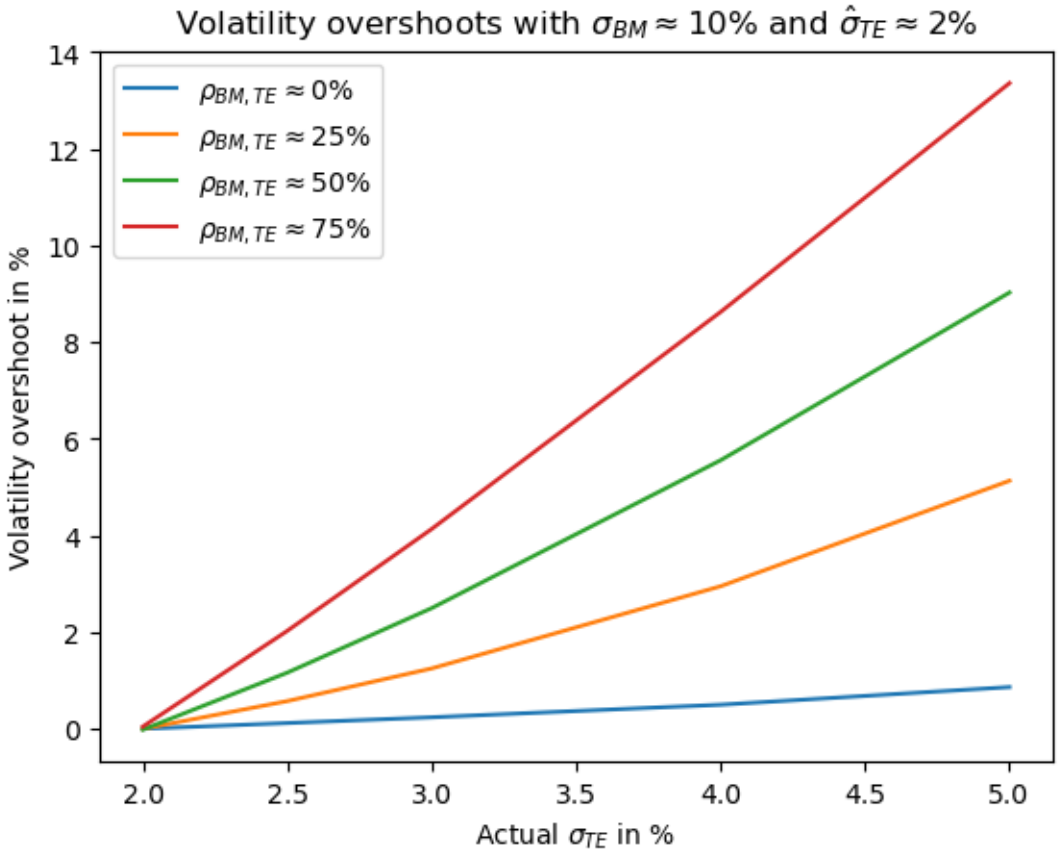

Figure 6.3.2: Risk overshoot with EP for $\sigma_{BM} \approx 10\%$, $\hat{\sigma}_{TE} \approx 2\%$, and various $\rho_{BM,TE}\sigma_{BM}$.

To understand the FX hedging argument, let σ_{BM} represent the risk of US equities or investment grade bonds. Let us assume that $\sigma_{BM} = 20\%$ for equities and $\sigma_{BM} = 5\%$ for bonds. As a foreign EUR investor, you have the opportunity to hedge the USD risk of these bonds or not. In that case, the tracking error portfolio consists of EURUSD. Let us assume that its risk is $\sigma_{TE} = 10\%$, which is the same no matter if we invest in equities or bonds. From the relation $\sigma_{TE}^2 + 2\rho_{BM,TE}\sigma_{BM}\sigma_{TE} \leq 0 \Leftrightarrow \frac{\sigma_{TE}}{\sigma_{BM}} \leq -2\rho_{BM,TE}$, we can see that for bonds we must require that $\rho_{BM,TE} = -1$ if the USD risk is not to increase the risk of the portfolio, while we only require $\rho_{BM,TE} = -0.25$ for equities. It is quite obvious that we are more likely to have a correlation of $\rho_{BM,TE} = -0.25$ or lower than $\rho_{BM,TE} = -1$. Hence, FX affects the risk of low risk bond portfolios more than higher risk equity portfolios.

CVaR is a more complex measure of risk that in the general case does not follow nice relationships like (6.3.2). For CVaR, the best we can do to replicate the form of (6.3.2) is

$$CVaR\left(PF\right) = CVaR\left(BM\right) + CVaR\left(TE\right) + f\left(BM, TE\right), \tag{6.3.3}$$

where $f(BM, TE) \leq 0$ is the diversification term. It follows that $f(BM, TE) \leq 0$ from the subadditivity property of CVaR, i.e., $CVaR(X + Y) \leq CVaR(X) + CVaR(Y)$, see Artzner et al. (1999) and Rockafellar, Uryasev, and Zabarankin (2006).

Note that the analysis of the volatility tracking error σ_{TE} and the correlation between the tracking error portfolio and benchmark $\rho_{BM,TE}$ can be generalized to the CVaR risk measure. With the portfolio and benchmark CVaR risks $CVaR(PF)$ and $CVaR(BM)$ fixed, there is again a trade-off between the diversification between the benchmark and tracking error portfolio $f(BM, TE)$ and the standalone tracking error portfolio risk $CVaR(TE)$. In summary, a better diversification represented by a lower $f(BM, TE)$ allows for a higher standalone risk $CVaR(TE)$ for the tracking error portfolio. For CVaR, the two terms are just more abstract and flexible than standard deviations and correlations.

Although it is only variance that we can decompose nicely into σ_{BM}, σ_{TE}, and $\rho_{BM,TE}$, the general principles are likely to hold for more complex risk measures such as CVaR, because the decomposition (6.3.3) will hold for any coherent risk measure $\mathcal{R}$. Hence, we must not be limited by the convenience of variance, because we pay a high price in terms of building our portfolios based on highly unrealistic market assumptions in that case.

Let us generally define portfolio optimization problems with risk targets and tracking error/tactical/short-term risk budgets as

$$e^\star = \underset{e}{\operatorname{argmax}}\ \mu^T e$$

subject to

$$\begin{aligned} \mathcal{R}(e) &\leq \mathcal{R}_{target}, \\ \mathcal{R}(e - e_{BM}) &\leq \mathcal{R}_{TE}, \\ v^T e &= 1, \\ e &\in \mathcal{E}. \end{aligned}$$

For some investment risk measure $\mathcal{R}$, e.g., CVaR or variance. As we have seen in the analysis above, this formulation introduces an implicit constraint on the diversification term $f(BM, TE)$.

The main takeaway from this section is that the two risk constraints introduce important trade-offs between the benchmark risk and the tracking error risk. It is important that we constantly take these dependencies into account and avoid potentially significantly underestimating the risk in our portfolio as shown in Figure 6.3.2. While it will be shown in Section 6.3.1 below how you can solve the risk budget optimization problem for variance, the same analysis as well as the subsequent Entropy Pooling stress-testing of the tracking error risk can be performed for CVaR. Readers are encouraged to test their understanding using the CVaR formulation from Section 6.2.1 and the CVaR Entropy Pooling views from Section 5.1.3 to perform this analysis.

A final comment is that we can of course also underestimate the risk of the benchmark. In the typical case where the benchmark represents broad "market beta" exposure, a high correlation between the benchmark and tracking error portfolios implies that the tracking error portfolio is simply more "market beta". It is almost equivalent to increasing the exposure of the futures position introduced in the first paragraph of this section. If the tracking error portfolio is simply characterized by systematically having more benchmark exposure, it should arguably become a part of the benchmark and not be

considered investment alpha.

6.3.1 Validating Portfolio Optimization Solvers

Having established that it is the problem with an overall risk target $\mathcal{R}_{target}$ and a risk budget for short-term deviations from a benchmark portfolio $\mathcal{R}_{TE}$ that is interesting from a portfolio construction perspective, we want to be able to validate that we are solving these problems correctly. This section and the accompanying code will give you the general principles for how you validate that and show you how to solve the problem for the variance risk measure using second-order cone programming, which is fairly straightforward compared to CVaR optimization.

Let us imagine that we have a solver which can solve the portfolio optimization problem for a return target μ_{target}. Perhaps it is a straightforward implementation like quadratic programming for mean-variance optimization, which we have potentially also validated against other implementations and believe is correct with a high probability. For this particular section and its accompanying code, we will use the mean-variance implementation from the fortitudo.tech package to compute the optimal long-only portfolio for a $\mu_{target} = 5\%$ return target and 25% individual upper bounds for the multi-asset instruments that follow with the package.

So how do we use the optimal exposure results $e^\star$ of the target return optimization to validate a target risk optimization? We simply compute the portfolio volatility $\sigma^\star$ for the optimal portfolio exposures $e^\star$ and use that as a target for the problem formulation that maximizes the expected return. These are equivalent problem formulations, so we can compare their results for the optimal exposures $e^\star$. Furthermore, we can compute the optimal expected return $\mu^\star$ and validate that it is indeed equal to $\mu_{target} = 5\%$. Hence, the validation of the target risk solver is straightforward to understand and hopefully implement.

Validating solvers that have both a risk target $\mathcal{R}_{target}$ and a tracking error target $\mathcal{R}_{TE}$ is a bit more challenging, while still conceptually fairly easy to understand. First, we must fix some benchmark exposure e_{BM} and introduce auxiliary variables e_{TE} through the constraint $e_{TE} = e - e_{BM}$. We can then implement risk target constraints directly on e_{TE} in the same way that we implement risk constraints on the overall portfolio risk. To validate that the dual risk solver gives us the correct results, we can compute the tracking error risk $\sigma^\star_{TE} = \mathcal{R}\left(e^\star - e_{BM}\right)$ using the optimal exposures $e^\star$ and use this as the target tracking error risk $\mathcal{R}_{TE}$ in our dual risk optimization.

The requirement that you have access to a portfolio optimization solver with a target return is in practice not that strict. This is the usual formulation, so for most risk measures you should be able to find an implementation that you can compare your own against, or simply be very careful with your own implementation and validate it in other ways. For the CVaR risk measure, you can use the fortitudo.tech package for target return and efficient frontier optimizations.

The accompanying code to this section illustrates the above procedure for variance using a second-order cone solver. You are encouraged to repeat the exercise for fully general Monte Carlo distributions and CVaR optimization after finalizing the CVaR optimization exercise from Section 6.2.1. Table 6.1 shows the results of the exercise for the variance risk measure. You can use the structure of the accompanying code to repeat this for CVaR, starting with a 5% return target. Make sure that you validate the CVaR solvers both for uniform and fully general scenario probabilities. The results for

both cases are given in the accompanying code to Section 6.2.1. Note that this is still a very basic use case, while solving the problem with advanced constraints and transaction costs is more challenging.

	Target return	Target risk	Target risk and tracking error	Benchmark
Gov & MBS	0.00	0.00	0.00	10.00
Corp IG	0.00	0.00	0.00	10.00
Corp HY	0.00	0.00	0.00	10.00
EM Debt	18.39	18.39	18.39	10.00
DM Equity	0.00	0.00	0.00	10.00
EM Equity	0.00	0.00	0.00	10.00
Private Equity	6.61	6.61	6.61	10.00
Infrastructure	25.00	25.00	25.00	10.00
Real Estate	25.00	25.00	25.00	10.00
Hedge Funds	25.00	25.00	25.00	10.00
Return target	**5.00**			
Volatility		**6.69**	**6.69**	
Tracking error			**3.18**	

Table 6.1: Mean-variance optimized portfolios using different objectives.

The numbers in bold from Table 6.1 indicate which constraints were used to optimize the portfolio. Hence, we started with a target return constraint, then a target risk constraint, and finally a target risk and tracking error constraint. You can find all the details in the accompanying code to this section.

Note that in Table 6.1 above, we optimized a portfolio with a return target first and then took the risk and tracking error targets as given. In practice, we would usually specify risk targets and tracking errors and then take the optimal expected return as given. Tracking error constraints can in practice contribute to alleviating the inherent portfolio optimization issues related to concentrated portfolios, especially when combined with transactions costs, but they do not solve all the issues. Therefore, portfolio optimization in practice must almost always take parameter uncertainty into account, which we do in the next section.

6.4 Parameter Uncertainty and Resampled Portfolio Stacking

This section introduces resampled parameter uncertainty into the portfolio optimization problem using the perspectives from Kristensen and Vorobets (2024). While the foundation is the same including a recap of the Exposure Stacking method, this section introduces new ways of using the Exposure Stacking objective and includes a case study that shows you how you can use the method to combine CVaR optimizations with parameter uncertainty and different α-CVaR levels in Section 6.4.3 below.

As we have seen in the previous sections, portfolio optimization problems are highly sensitive to parameter estimates or more generally the market model input. Some practitioners are so skeptical that they do not use portfolio optimization methods at all, at least explicitly. Most people still attempt to build mean-risk optimal portfolios, albeit in implicit or heuristic ways.

Resampled portfolio optimization is a heuristic first introduced by Michaud and Michaud (1998), where B market model or parameter samples are generated. For each of these samples, we optimize

portfolios with the same constraints and similar risk levels to compute a sample of optimal exposures $e_b^\star \in \mathbb{R}^I$, $b \in \{1, 2, \ldots, B\}$. Afterwards, we compute a weighted average of the optimal exposures to get the final resampled portfolio

$$e_w^\star = \sum_{b=1}^{B} w_b e_b^\star, \tag{6.4.1}$$

with $w_b \geq 0$ and $\sum_{b=1}^{B} w_b = 1$.

The original suggestion by Michaud and Michaud (1998) is to set $w_b = \frac{1}{B}$ and align the portfolio risk through the portfolio's index on the efficient frontier. A desirable aspect of the resampled approach is that it is highly flexible, while it initially did not have much justification. Fundamental perspectives justifying the resampled approach are presented by Kristensen and Vorobets (2024), while also introducing other ways of aligning the portfolio risk with a new method for determining the sample weights w_b.

To understand the new stacking methods, it is important to view the portfolio optimization problem from the right perspective. We consider a risk-adjusted return objective f, which is a function of the market model (R, p) with $R \in \mathbb{R}^{S \times I}$ being a matrix of joint market scenarios and $p \in \mathbb{R}^S$ being an associated scenario probability vector from (1.1.1). Additionally, f is a function of the optimization constraints $\mathcal{E}$ and the portfolio exposures $e \in \mathbb{R}^I$. First, consider the case where f determines the expected return, while constraints on the risk are included in $\mathcal{E}$. The optimal portfolio exposures $e^\star$ are then given by

$$e^\star = \underset{e}{\operatorname{argmax}}\ f\left(R, p, \mathcal{E}, e\right).$$

It is also possible to work from the perspective of minimizing risk with a return target. In such cases, the optimal portfolio exposures are given by

$$e^\star = \underset{e}{\operatorname{argmin}}\ g\left(R, p, \mathcal{E}, e\right),$$

where g is a function that determines the risk, while the return target is included in the constraints $\mathcal{E}$. As we noted in Section 6.3.1 above, the two perspectives solve equivalent risk-adjusted return problems.

Kristensen and Vorobets (2024) argue that the parameter uncertainty issue can be analyzed from the same perspective as statistical learning models using the bias-variance trade-off. Specifically, treating the market model or parameters as data and the optimal portfolio exposures as estimates, with the particular mean-risk optimization method being an estimator. From this perspective, the traditional mean-risk portfolio optimization estimators have no bias but a high variance due to their sensitivity to the market model and parameter values. On the other hand, resampled versions of these estimators have some bias but a lower variance.

To explore resampled portfolio optimization from a bias-variance perspective, we define the optimal exposure for sample $b \in \{1, 2, \ldots, B\}$ by

$$e_b^\star = \underset{e}{\operatorname{argmax}}\ f\left(R_b, p_b, \mathcal{E}, e\right) = \underset{e}{\operatorname{argmin}}\ g\left(R_b, p_b, \mathcal{E}, e\right).$$

Note that in this definition, we are optimizing over samples (R_b, p_b) of the market model. For the mean-variance objective, this is simply estimates of mean vectors μ_b and covariance matrices Σ_b. For

the mean-CVaR objective, this can be entirely new joint market scenarios R_b and associated probability vectors p_b, with p_b possibly generated using Sequential Entropy Pooling as described in Chapter 5. In their case study, Kristensen and Vorobets (2024) show that it is usually the mean vectors that affect the efficient exposures $e_b^\star$ the most.

How do we analyze the risk-adjusted objective from a bias-variance perspective? We must first understand in which sense the traditional portfolio optimization methods such as mean-variance and mean-CVaR are unbiased. We call a mean-risk estimator unbiased if it correctly estimates $e^\star$, i.e., if it correctly estimates the optimal portfolio exposures given the market model (R, p). Clearly, the resampled estimator (6.4.1) is not unbiased in the general case.

Hence, when we perform resampled optimization, we are purposefully introducing bias to reduce the variance of the final optimal exposures estimate $e_w^\star$. If we weight the sample estimates equally as originally suggested, we are just hoping that the increase in bias will reduce the variance more. Although this is also what people experience in practice, we do not have any guarantees that the bias-variance trade-off is minimized by an equally weighted average, so it is worth exploring whether we can do better.

Direct minimization of the vector mean squared error

$$MSE\left(e_w^\star\right) = \mathbb{E}\left[\left\|\sum_{b=1}^{B} w_b e_b^\star - e^\star\right\|_2^2\right]$$

of the resampled estimator (6.4.1) is not interesting given that the solution will simply try to find a linear combination of the vectors $e_b^\star$ which is as close as possible to $e^\star$ with respect to the Euclidean norm $\|\cdot\|_2$. Instead, we focus on multivariate stacking objectives having a form similar to the vector mean squared error.

Stacked generalization, see Wolpert (1992) and Breiman (1996), has received increased attention in recent years due to its versatility and potential for improving out-of-sample performance. Hence, we analyze the resampled portfolio optimization problem from this perspective. A natural, albeit slightly naive, starting point for the objective function is to ensure that the Euclidean norm of the difference between $e_w^\star$ and each $e_b^\star$ is, on average, as small as possible, i.e., solve the problem

$$w^\star = \underset{w}{\operatorname{argmin}} \frac{1}{B}\sum_{k=1}^{B}\left\|e_k^\star - \sum_{b=1}^{B} w_b e_b^\star\right\|_2^2. \tag{6.4.2}$$

However, as shown in Appendix A of Kristensen and Vorobets (2024), the optimal solution to (6.4.2) always yields a vector of equal sample weights $w_b^\star = \frac{1}{B}$ for all $b \in \{1, 2, \ldots, B\}$, corresponding to the traditional resampling method.

Instead of (6.4.2), Kristensen and Vorobets (2024) define an objective function based on the ideas of stacked regression and cross-validation. Let $\mathcal{B} = \{1, 2, \ldots, B\}$ be the set of sample indices and suppose that we partition $\mathcal{B}$ into L nonempty sets $\mathcal{K}_1, \mathcal{K}_2, \ldots, \mathcal{K}_L$ for some $L \in \{2, 3, \ldots, B\}$. For any choice of $l \in \{1, 2, \ldots, L\}$, we consider the set of exposure vectors $\{e_k \mid k \in \mathcal{K}_l\}$ as a validation set, find sample weights w_b for the remaining $B - |\mathcal{K}_l|$ exposure vectors, and calculate the average difference ε_l

between the weighted exposure and the validation set exposures, i.e., we compute

$$\varepsilon_l = \frac{1}{|\mathcal{K}_l|} \sum_{k \in \mathcal{K}_l} \left\| e_k^\star - \sum_{b \notin \mathcal{K}_l} w_b e_b^\star \right\|_2^2 .$$

If we repeat the analysis using each of the sets $\mathcal{K}_1, \mathcal{K}_2, \ldots, \mathcal{K}_L$ as our validation set indices, we can compute the L-fold cross-validation estimate as the average $\varepsilon = \frac{1}{L} \sum_{l=1}^{L} \varepsilon_l$, see James et al. (2023). Thus, instead of using the naive objective function (6.4.2), we wish to find the vector of sample weights $w^\star$ that minimizes the cross-validation estimate ε, i.e., solve the problem

$$w^\star = \underset{w}{\operatorname{argmin}} \frac{1}{L} \sum_{l=1}^{L} \left(\frac{1}{|\mathcal{K}_l|} \sum_{k \in \mathcal{K}_l} \left\| e_k^\star - \sum_{b \notin \mathcal{K}_l} w_b e_b^\star \right\|_2^2 \right). \tag{6.4.3}$$

Kristensen and Vorobets (2024) show in Appendix B that (6.4.3) can be formulated as a quadratic programming problem and therefore solved in a fast and stable way. Note that (6.4.3) allows us to stack exposures both across different market model samples (R_b, p_b) to incorporate parameter uncertainty as well as different efficient portfolio estimators. For example, one could combine mean-variance, mean-CVaR, or even mean-CVaR with different CVaR levels such as 90%, 95%, and 97.5%, see the case study in Section 6.4.3.

The original suggestion to stack based on portfolio exposures $e_b^\star$ is based on the risk alignment considerations from Section 2.1 in Kristensen and Vorobets (2024). The case study in the accompanying code to this section confirms that the portfolios stacked on the portfolio mean or volatility indeed drift quite significantly in relation to risk and return compared to the original resampled frontier. Table 6.2 shows the results for a repeated analysis from Kristensen and Vorobets (2024) with a different seed and including portfolio mean, volatility, and mean / volatility stacked resampled portfolios for $L = 2$.

	Resampled	Exposure	Mean	Volatility	Mean / Volatility	Frontier
Gov & MBS	0.12	0.00	0.00	0.00	0.00	0.00
Corp IG	0.00	0.00	0.00	0.00	0.00	0.00
Corp HY	0.00	0.00	0.00	0.00	0.00	0.00
EM Debt	4.88	0.00	0.00	14.00	0.00	0.00
DM Equity	0.48	0.00	0.00	0.00	0.00	0.00
EM Equity	0.07	0.00	0.00	0.00	0.00	0.00
Private Equity	18.07	13.92	25.60	33.69	25.60	20.41
Infrastructure	34.35	46.49	32.14	17.17	32.14	40.49
Real Estate	16.36	10.49	18.01	8.39	18.01	11.06
Hedge Funds	25.66	29.11	24.25	26.75	24.25	28.04
Mean	6.34	6.25	7.02	7.29	7.02	6.71
Vol	8.96	8.78	10.26	11.26	10.26	9.61

Table 6.2: Mean-variance optimal exposures for $L = 2$ as well as portfolio return and volatility.

From Table 6.2, we immediately note the significant drift in the risk and return of the resampled portfolios that are stacked based on portfolio mean, volatility, and mean / volatility. It is also interest-

ing that the portfolios based on mean and mean / volatility coincide in this case. The issues with drift in risk and return properties seem to diminish once we set $L = 10$, see Table 6.3. Hence, the number of folds parameter L acts as a hyperparameter for the resampled stacking methods. It seems to affect the various objectives differently, where low values such as $L = 2$ work well for Exposure Stacking while higher values such as $L = 10$ are required for stacking based on portfolio mean and volatility.

	Resampled	Exposure	Mean	Volatility	Mean / Volatility	Frontier
Gov & MBS	0.12	0.00	0.06	0.05	0.03	0.00
Corp IG	0.00	0.00	0.00	0.00	0.00	0.00
Corp HY	0.00	0.00	0.00	0.00	0.00	0.00
EM Debt	4.88	3.35	4.33	6.82	4.60	0.00
DM Equity	0.48	0.00	0.45	0.36	0.48	0.00
EM Equity	0.07	0.00	0.05	0.05	0.05	0.00
Private Equity	18.07	17.97	20.15	22.71	20.72	20.41
Infrastructure	34.35	36.07	33.17	30.33	33.19	40.49
Real Estate	16.36	16.16	14.48	14.57	14.98	11.06
Hedge Funds	25.66	26.46	27.31	25.12	25.95	28.04
Mean	6.34	6.38	6.52	6.65	6.56	6.71
Vol	8.96	9.00	9.30	9.60	9.39	9.61

Table 6.3: Mean-variance optimal exposures for $L = 10$ as well as portfolio return and volatility.

For more details about the above computations, see the accompanying code to this section. Since these method are still very new, there is generally still a need to extensively explore them to see when they work well. The same is true for the Resampled Portfolio Stacking methods that are introduced for the first time in this book in the sections below. A Python function for solving (6.4.3) is provided in the accompanying code, so it is easy to explore these methods further. The code also allows you to change the L parameter and see the consequences. Finally, the code contains out-of-sample return, risk, and risk-adjusted return distributions graphs similar to Kristensen and Vorobets (2024).

6.4.1 Return and Risk Stacking

While the Exposure Stacking method maintains the overall risk of the portfolio and correctly focuses on the portfolio exposures e rather than portfolio weights, it treats all exposures as equally important even if their standalone risks or risk contributions are very different. For example, an exposure deviation of \$100 in low risk money market investment will be treated in the same way as a high risk equities exposure. However, the high risk equities exposure likely has a significantly larger effect on the portfolios risk and return characteristics. This section therefore introduces marginal Risk and Return Stacking, which is (6.4.3) applied to the marginal risk contributions $\mathcal{R}^{'}(e) \odot e \in \mathbb{R}^{I}$ and marginal return contributions $\mu \odot e \in \mathbb{R}^{I}$, or even the marginal risk-adjusted returns that consist of the individual ratios of the two $(\mu \odot e) \oslash \left(\mathcal{R}^{'}(e) \odot e\right) = \mu \oslash \mathcal{R}^{'}(e) \in \mathbb{R}^{I}$, with $\oslash$ denoting the element-wise Hadamard division. See Section 7.1 for more information on these vectors.

This section introduces the new methods by repeating many parts of the case study from Kristensen and Vorobets (2024) for mean-variance optimization with mean uncertainty and Exposure, Return, Risk, and Return / Risk Stacking. It illustrates out-of-sample results similar to the case study from

	Resampled	Exposure	Return	Risk	Return / Risk	Frontier
Gov & MBS	0.12	0.00	0.00	0.00	0.00	0.00
Corp IG	0.00	0.00	0.00	0.00	0.00	0.00
Corp HY	0.00	0.00	0.00	0.00	0.00	0.00
EM Debt	4.88	0.00	0.00	1.36	0.43	0.00
DM Equity	0.48	0.00	0.00	0.00	0.00	0.00
EM Equity	0.07	0.00	0.00	0.00	0.00	0.00
Private Equity	18.07	13.92	21.90	24.90	18.42	20.41
Infrastructure	34.35	46.49	38.89	37.44	38.54	40.49
Real Estate	16.36	10.49	12.66	9.23	15.60	11.06
Hedge Funds	25.66	29.11	26.55	27.07	27.01	28.04
Mean	6.34	6.25	6.81	7.03	6.50	6.71
Vol	8.96	8.78	9.80	10.25	9.21	9.61

Table 6.4: Mean-variance optimal exposures for $L = 2$ with Exposure, Return, Risk, and Return / Risk Stacking.

Kristensen and Vorobets (2024) for $L \in \{2, 5, 20, B = 1000\}$. While this gives us many different insights, there are still many other interesting analyses that can be performed for the new methods. Readers are encouraged to explore these on their own and share their experiences. All the details for the computations related to this section can be found in the accompanying code. The rest of this section focuses on showing the simulation case study results.

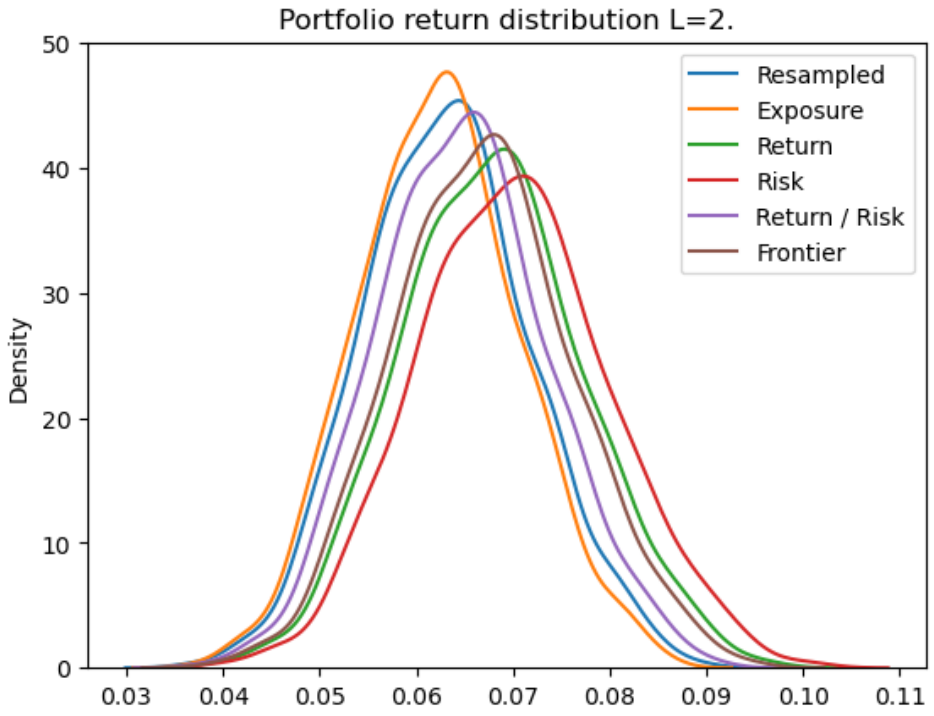

Figure 6.4.1: Out-of-sample portfolio return distribution for $L = 2$.

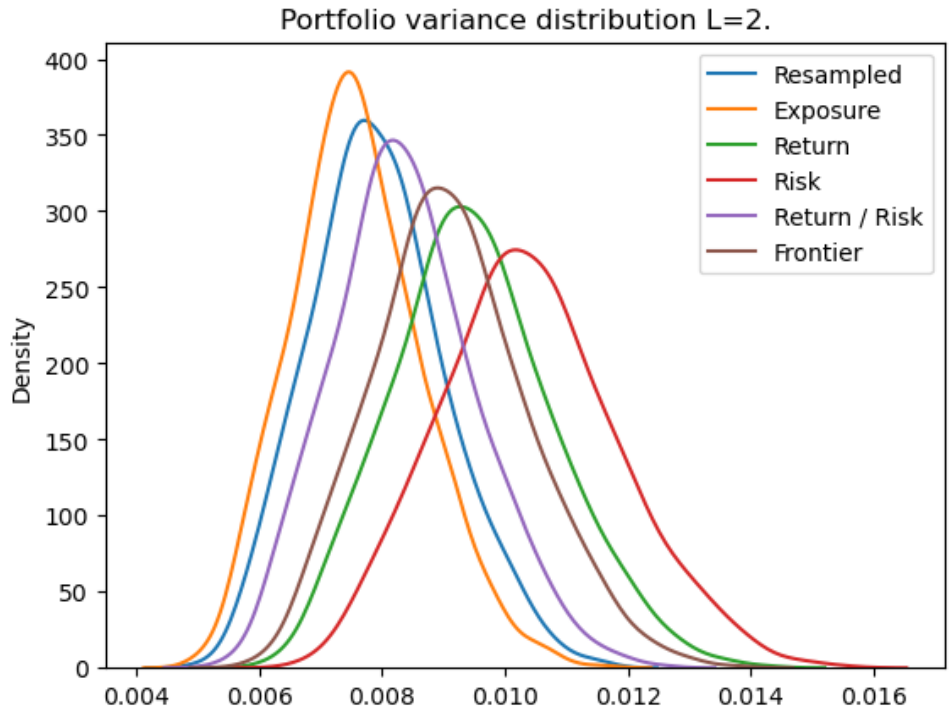

Figure 6.4.2: Out-of-sample portfolio variance distribution for $L = 2$.

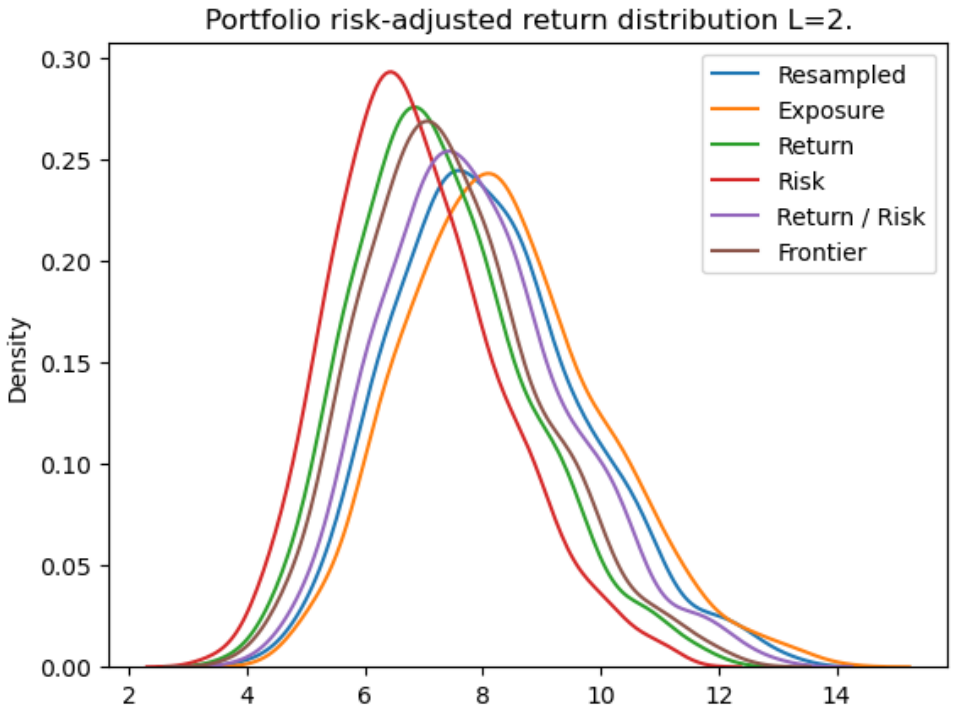

Figure 6.4.3: Out-of-sample portfolio return / variance distribution for $L = 2$.

	Resampled	Exposure	Return	Risk	Return / Risk	Frontier
Gov & MBS	0.12	0.00	0.01	0.00	0.00	0.00
Corp IG	0.00	0.00	0.00	0.00	0.00	0.00
Corp HY	0.00	0.00	0.00	0.00	0.00	0.00
EM Debt	4.88	1.27	5.21	1.85	2.25	0.00
DM Equity	0.48	0.00	0.52	0.00	0.00	0.00
EM Equity	0.07	0.00	0.06	0.00	0.00	0.00
Private Equity	18.07	17.76	18.26	20.33	16.19	20.41
Infrastructure	34.35	38.11	34.12	34.81	38.16	40.49
Real Estate	16.36	15.62	16.29	15.55	20.57	11.06
Hedge Funds	25.66	27.25	25.53	27.47	22.82	28.04
Mean	6.34	6.42	6.35	6.59	6.26	6.71
Vol	8.96	9.07	8.99	9.40	8.78	9.61

Table 6.5: Mean-variance optimal exposures for $L = 5$ with Exposure, Return, Risk, and Return / Risk Stacking.

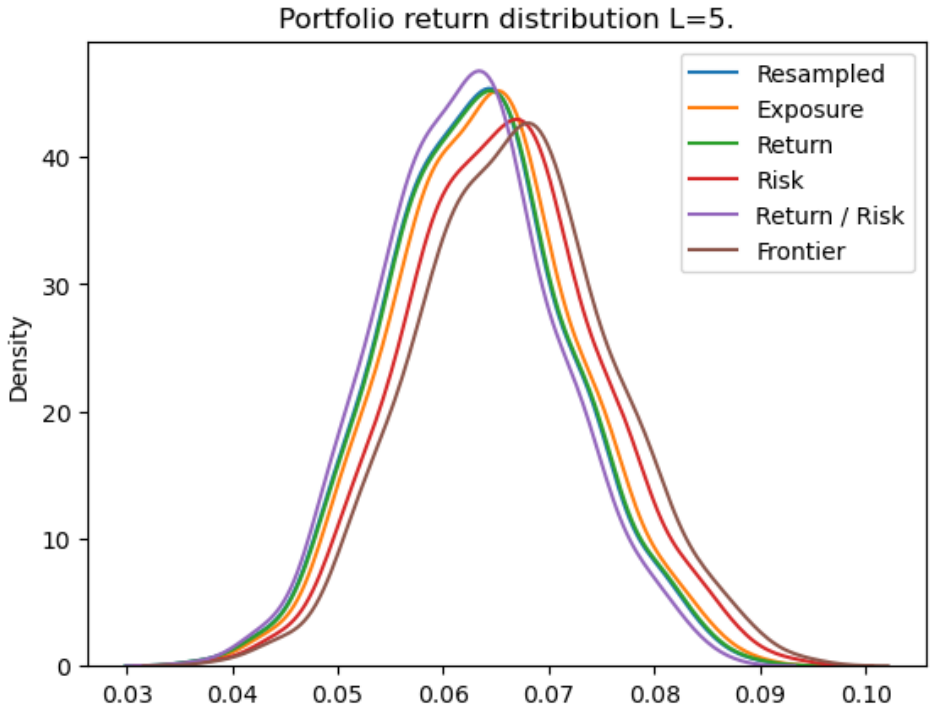

Figure 6.4.4: Out-of-sample portfolio return distribution for $L = 5$.

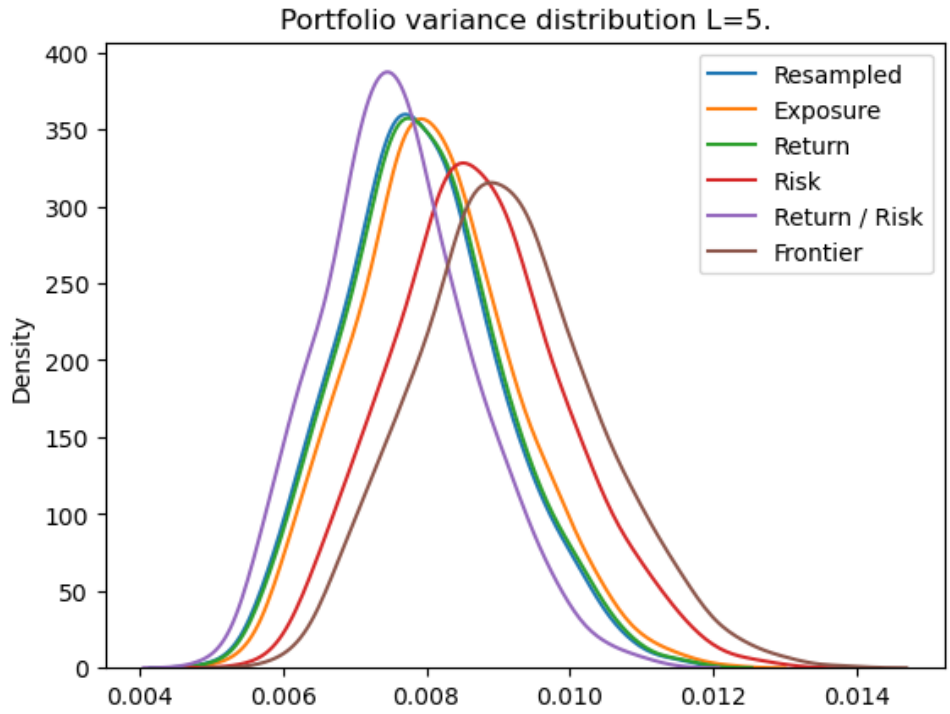

Figure 6.4.5: Out-of-sample portfolio variance distribution for $L = 5$.

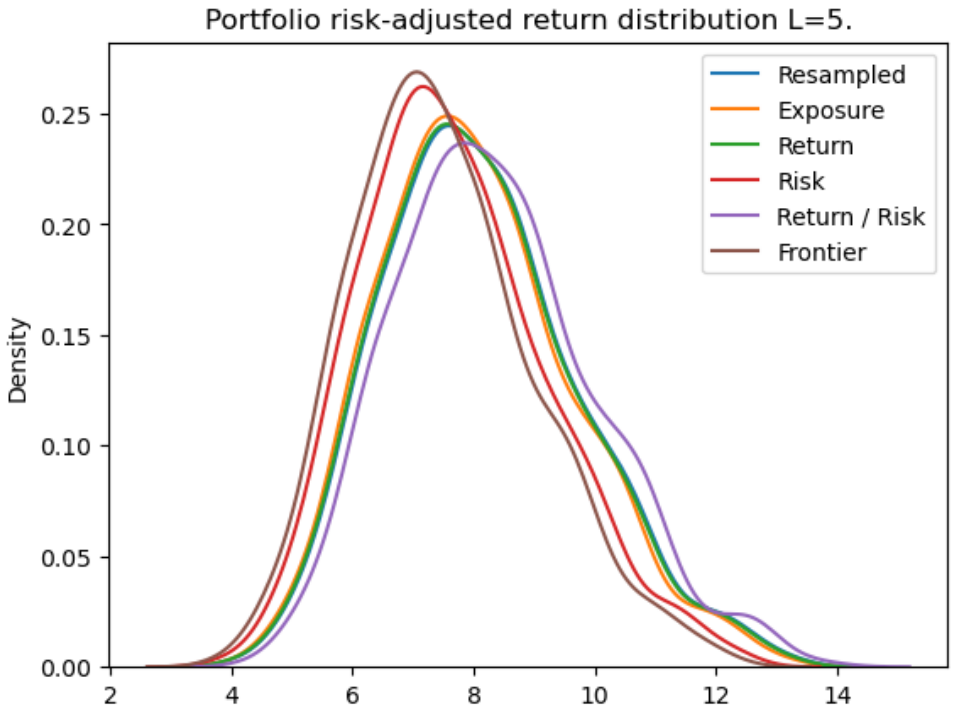

Figure 6.4.6: Out-of-sample portfolio return / variance distribution for $L = 5$.

	Resampled	Exposure	Return	Risk	Return / Risk	Frontier
Gov & MBS	0.12	0.00	0.00	0.00	0.00	0.00
Corp IG	0.00	0.00	0.00	0.00	0.00	0.00
Corp HY	0.00	0.00	0.00	0.00	0.00	0.00
EM Debt	4.88	4.21	5.18	5.59	3.92	0.00
DM Equity	0.48	0.00	0.48	0.41	0.04	0.00
EM Equity	0.07	0.00	0.01	0.01	0.00	0.00
Private Equity	18.07	18.06	17.99	18.10	14.83	20.41
Infrastructure	34.35	35.24	34.36	34.25	42.86	40.49
Real Estate	16.36	16.35	16.64	16.32	21.35	11.06
Hedge Funds	25.66	26.13	25.33	25.32	17.00	28.04
Mean	6.34	6.36	6.33	6.33	6.16	6.71
Vol	8.96	8.97	8.94	8.94	8.65	9.61

Table 6.6: Mean-variance optimal exposures for $L = 20$ with Exposure, Return, Risk, and Return / Risk Stacking.

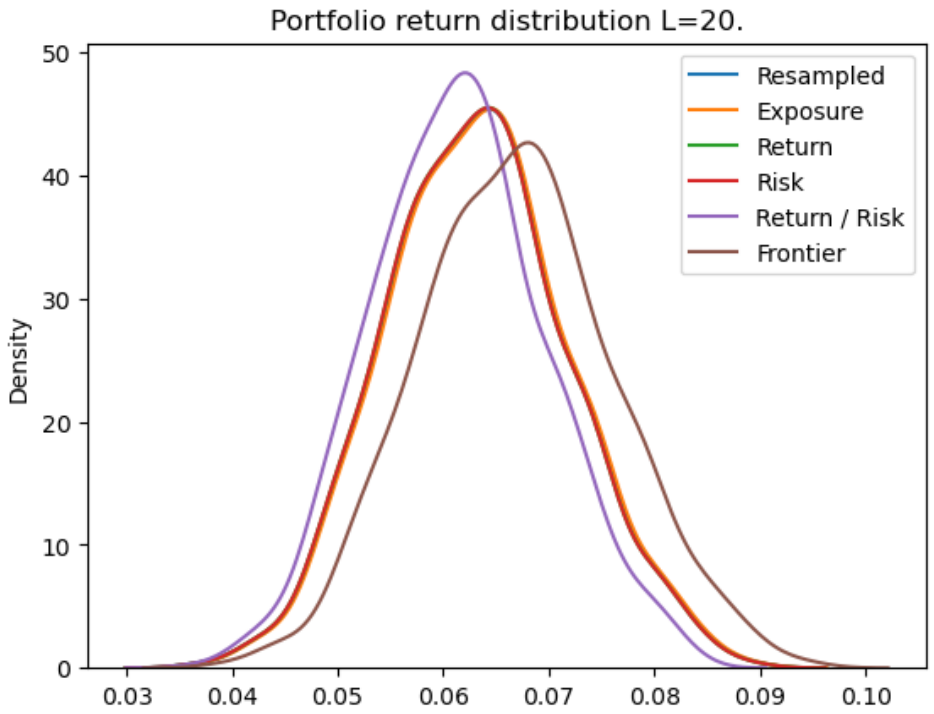

Figure 6.4.7: Out-of-sample portfolio return distribution for $L = 20$.

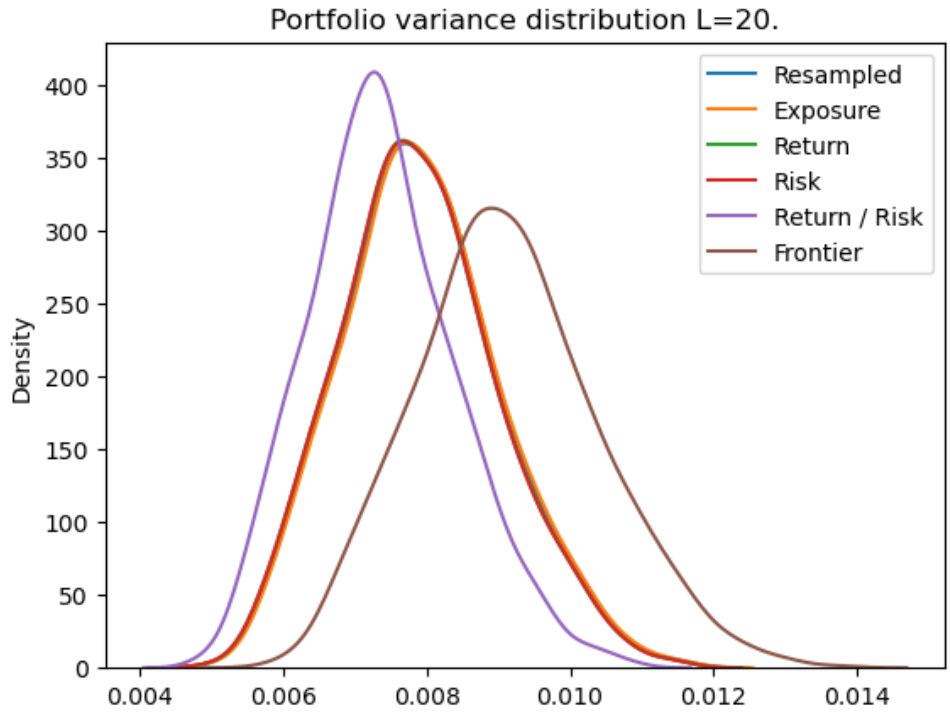

Figure 6.4.8: Out-of-sample portfolio variance distribution for $L = 20$.

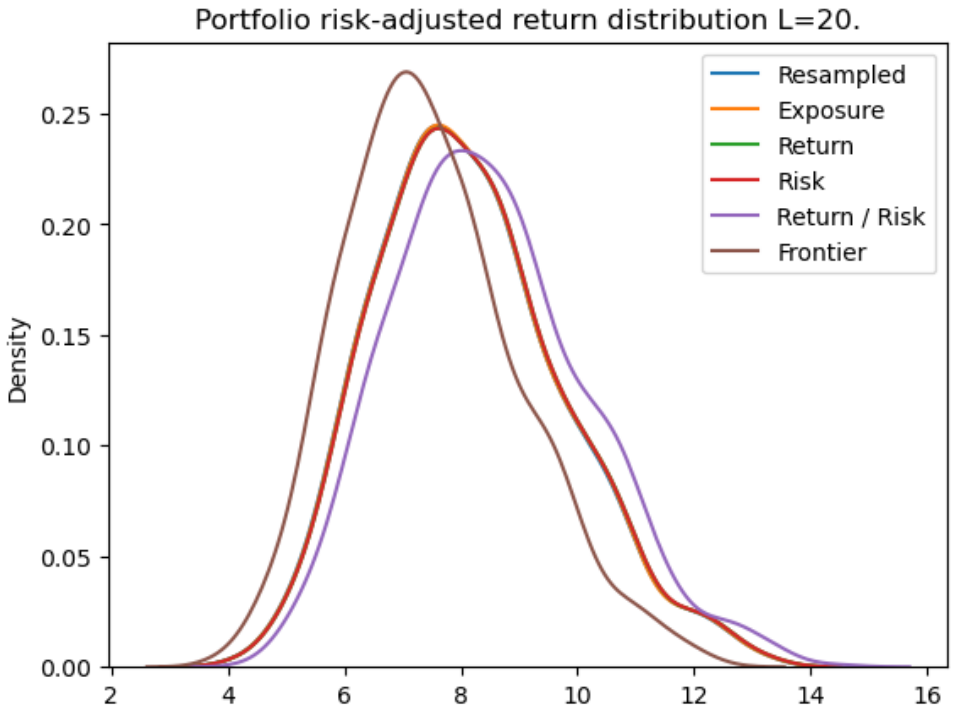

Figure 6.4.9: Out-of-sample portfolio return / variance distribution for $L = 20$.

	Resampled	Exposure	Return	Risk	Return / Risk	Frontier
Gov & MBS	0.12	0.12	0.00	0.00	0.03	0.00
Corp IG	0.00	0.00	0.00	0.00	0.00	0.00
Corp HY	0.00	0.00	0.00	0.00	0.00	0.00
EM Debt	4.88	4.86	4.41	4.32	3.46	0.00
DM Equity	0.48	0.47	0.49	0.49	0.45	0.00
EM Equity	0.07	0.11	0.07	0.08	0.02	0.00
Private Equity	18.07	18.07	19.00	18.22	17.57	20.41
Infrastructure	34.35	34.36	35.11	35.53	38.34	40.49
Real Estate	16.36	16.36	16.03	16.39	18.29	11.06
Hedge Funds	25.66	25.66	24.90	24.98	21.85	28.04
Mean	6.34	6.34	6.44	6.38	6.36	6.71
Vol	8.96	8.97	9.14	9.03	8.99	9.61

Table 6.7: Mean-variance optimal exposures for $L = B = 1000$ with Exposure, Return, Risk, and Return / Risk Stacking.

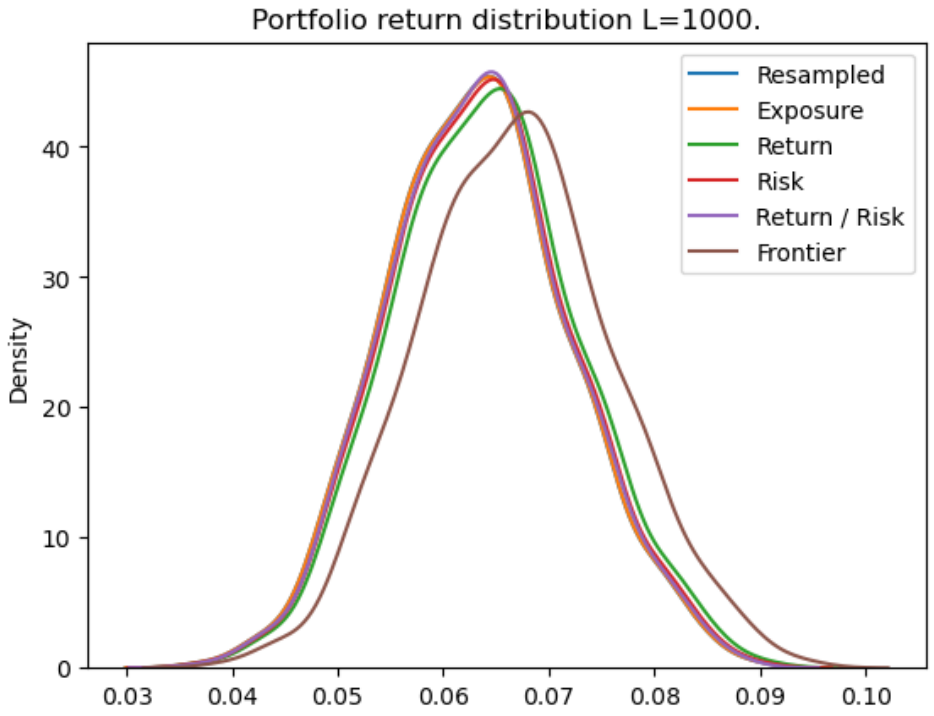

Figure 6.4.10: Out-of-sample portfolio return distribution for $L = B = 1000$.

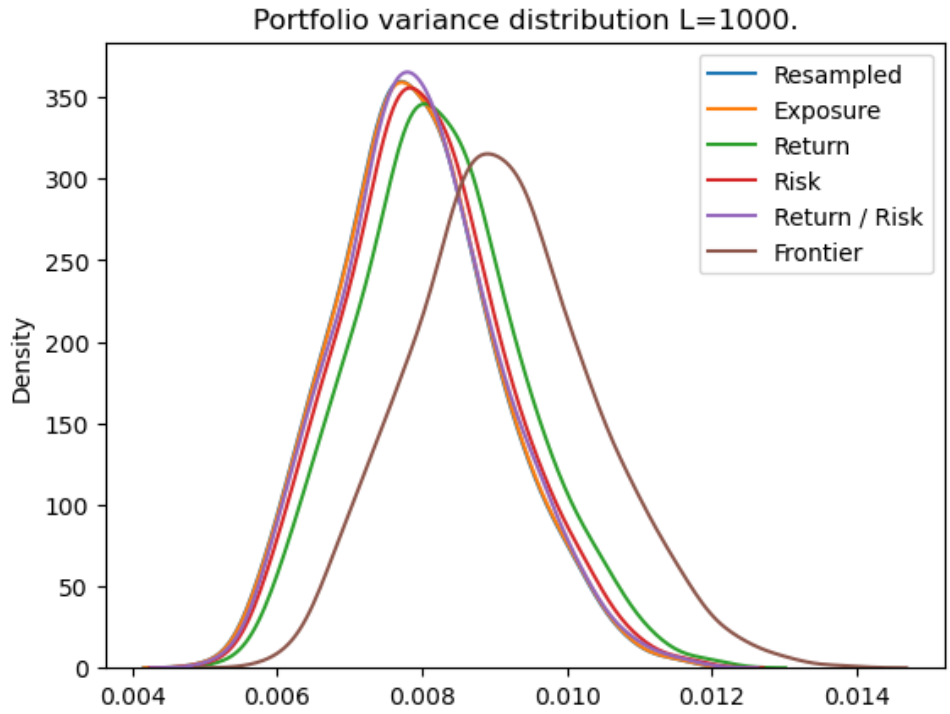

Figure 6.4.11: Out-of-sample portfolio variance distribution for $L = B = 1000$.

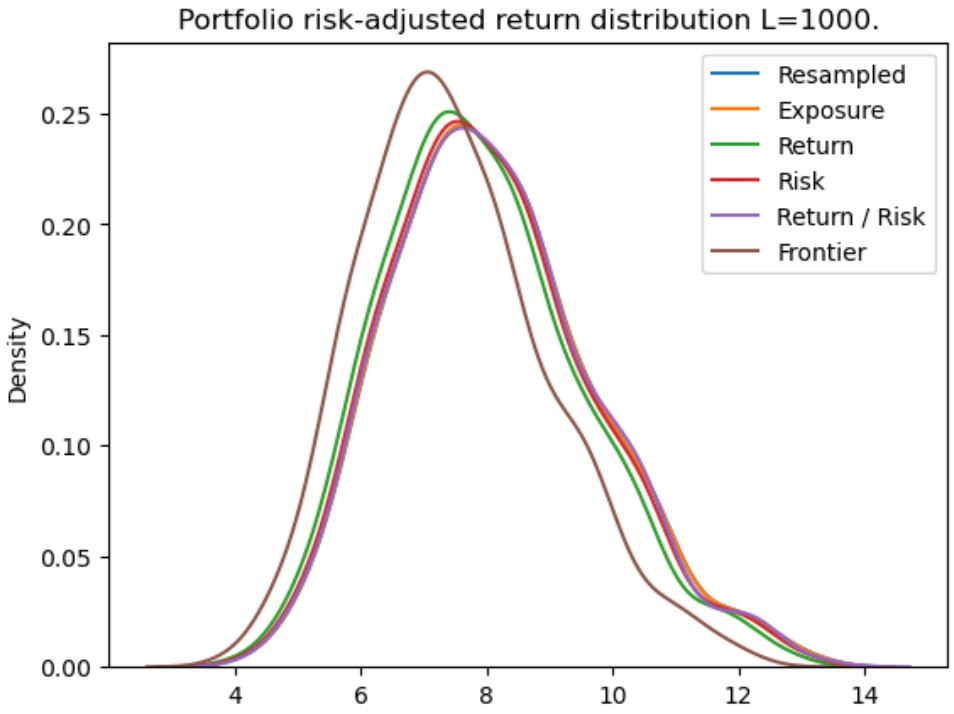

Figure 6.4.12: Out-of-sample portfolio return / variance distribution for $L = B = 1000$.

From the above figures and tables, we can conclude that L serves as a hyperparameter, and that the different resampled portfolio optimization approaches react differently to this hyperparameter. Stacking based on exposures seems to perform well for low values of L, while stacking based on risk and return require a higher L to not significantly drift in terms of portfolio risk and return. Once L is increased towards the number of resamples B, all approaches converge to similar results with only minor deviations.

It is important to underline that the case studies in this chapter still do not allow us to make definite conclusions. The results might be affected by the particular normally distributed market simulation as well as potential numerical instability in the estimation of the sample weights w_b. Hence, the new methods must still be carefully tested on practical cases to assess the magnitude of their potential risk-adjusted gains. It might also be meaningful to combine the various objectives either into one vector or through some form of weighting.

This section is concluded by coining all approaches that solve the objective in (6.4.3) with some resampled target $x_b^\star$ that is not necessarily the exposures $e_b^\star$ as Resampled Portfolio Stacking. The figures below compares the out-of-sample results for $L \in \{2, 5, 20, B = 1000\}$ and Return / Risk stacking.

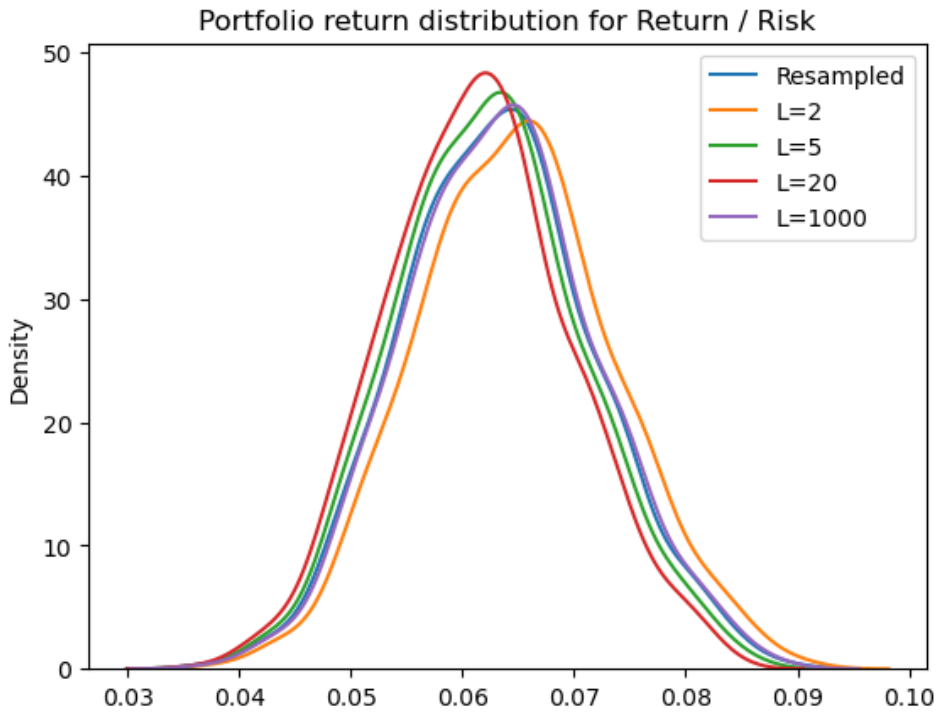

Figure 6.4.13: Out-of-sample portfolio return distribution for Return / Risk stacking.

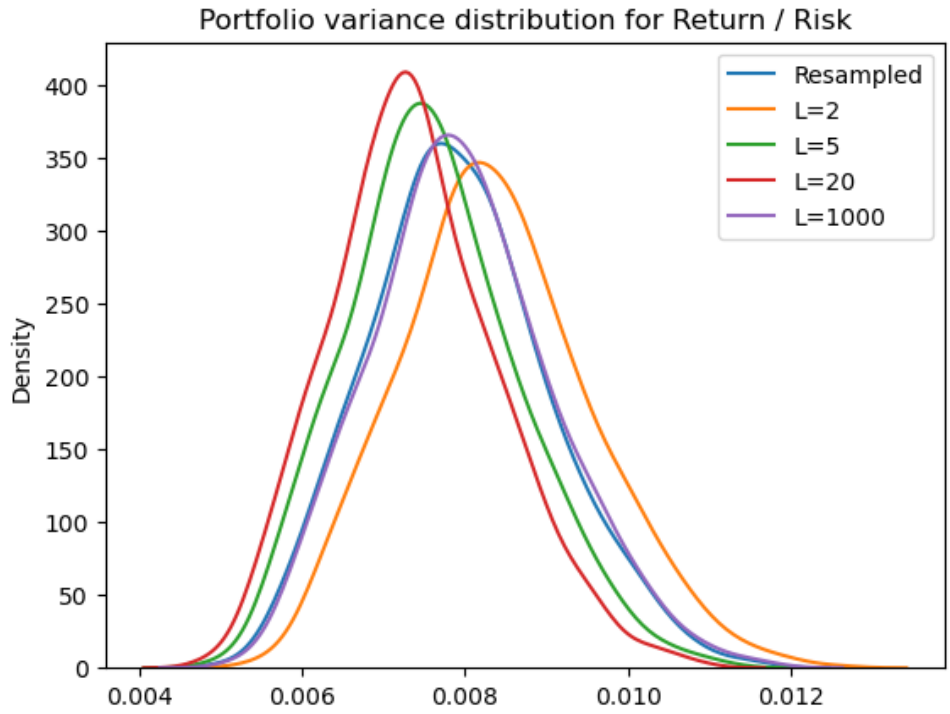

Figure 6.4.14: Out-of-sample portfolio variance distribution for Return / Risk stacking.

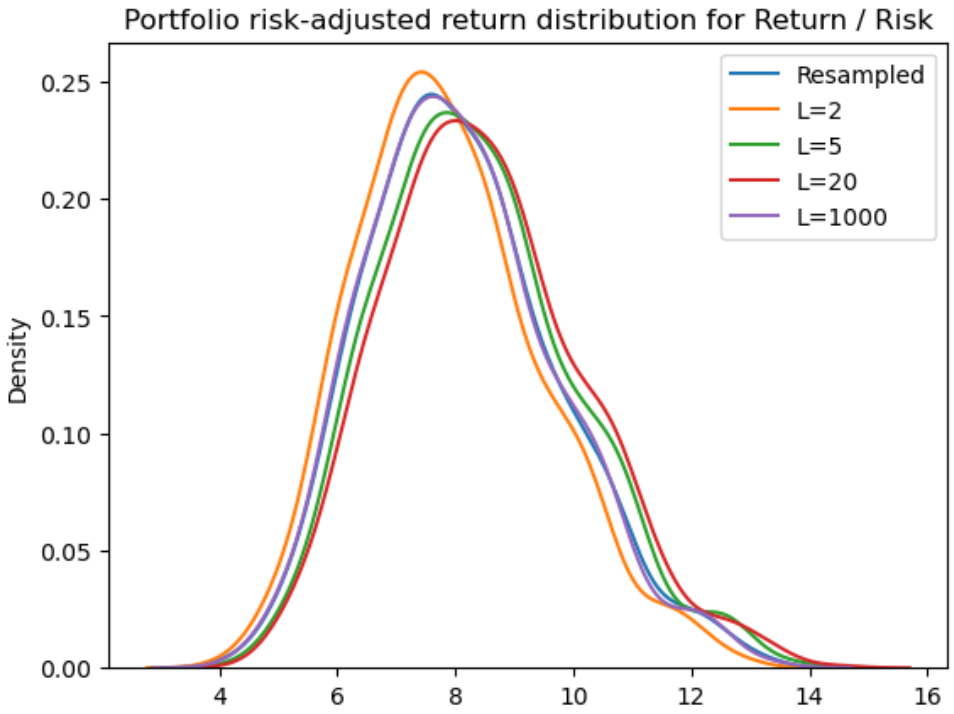

Figure 6.4.15: Out-of-sample portfolio return / variance distribution for for Return / Risk stacking.

6.4.2 Derivatives and Risk Factor Parameter Uncertainty

As we have seen in Section 6.1, derivatives introduce only a simple extra layer of complexity in general portfolio management that requires us to separate between relative market values $v \in \mathbb{R}^I$ and relative exposures $e \in \mathbb{R}^I$, see also Vorobets (2022a) for the original documentation.

For portfolio optimization with parameter uncertainty, derivatives introduce significantly more complexity as we must ensure that the derivatives P&L is consistent with the parameter uncertainty we introduce in the underlying and risk factors such as implied volatilities. For example, the expected P&L of a European option at expiry is fully determined by the P&L of the underlying. Hence, we cannot introduce separate parameter uncertainty into the underlying and the derivative instrument if we want to maintain logical consistency.

While Resampled Portfolio Stacking allows us to introduce parameter uncertainty by generating both new market simulations R_b and scenario probability vectors p_b for $b = 1, 2, \ldots, B$, we probably want to avoid the costly simulation and pricing of derivatives associated with generating R_b for each sample. Hence, we fix $R_b = R$ for all samples and introduce parameter uncertainty into the derivative instruments by adjusting p_b with Entropy Pooling.

An elegant feature of the Entropy Pooling method is that it allows us to avoid the potentially costly repricing of derivative instruments, see the case study in Vorobets (2022a). The special aspect of derivatives is that they are inseparable from the underlying and the derivative's other risk factors. Hence, parameter uncertainty related to derivatives P&L should be introduced through parameter uncertainty in the underlying and the derivative's other risk factors. This is exactly what the approach introduced by Vorobets (2024) does.

For each sample $b = 1, 2, \ldots, B$, the algorithm is:

1. Introduce parameter uncertainty into the non-derivative instruments and risk factors.
2. Compute Entropy Pooling posterior probability vectors p_b using the new parameters as views.
3. Compute CVaR optimal exposures $e_b^\star$ using R and p_b.
4. Compute the final optimal exposure of the resampled estimator $e_w^\star$ using Resampled Portfolio Stacking to compute sample weights w_b.

Using CVaR instead of variance becomes especially crucial for option portfolios that clearly have nonlinear dependencies, which the covariance matrix cannot handle. However, even for plain vanilla cash instruments such as stocks and bonds, the mean-variance assumptions are likely to be severely violated, see Chapter 2. Hence, it is always recommended to use CVaR with fully general Monte Carlo distributions (1.1.1) in practice.

The procedure can also be used to generate parameter uncertainty for risk factor distributions in general. For example, imagine a portfolio of US government bonds where you want to introduce uncertainty into the 10 year zero-coupon yield or a set of key interest rates. This is likely a better approach than introducing, for example, mean uncertainty into all of the bonds. An elegant feature of the framework is that it also assists us in estimating what happens to, for example, the implied volatility when we introduce uncertainty in the underlying. We don't have to specify this manually if we don't want to.

6.4.3 Multiple CVaR Levels Case Study

This section uses the constraints and simulation from Vorobets (2022a) and Vorobets (2024) to illustrate how Exposure Stacking can be applied across different optimization methods as suggested by Kristensen and Vorobets (2024). In particular, optimization is performed for 90%, 95%, and 97.5%-CVaR for a multi-asset portfolio containing derivatives. The risk is aligned by selecting the middle portfolio on the efficient frontier. The resampled efficient exposures are shown in Table 6.8 below.

	90%	95%	97.5%	Combined
Gov & MBS	0.00	0.10	0.56	0.02
Corp IG	0.00	0.00	0.00	0.00
Corp HY	0.00	0.00	0.00	0.00
EM Debt	12.85	11.73	10.71	11.91
DM Equity	0.85	11.08	18.76	9.84
EM Equity	0.00	0.00	0.00	0.00
Private Equity	11.56	7.76	5.41	8.25
Infrastructure	24.23	20.44	17.22	21.06
Real Estate	20.46	16.59	14.12	17.01
Hedge Funds	25.00	25.00	25.00	25.00
Put 90	-50.00	-50.00	-49.63	-50.00
Put 95	-9.20	14.44	21.36	9.16
Put ATMF	50.00	50.00	50.00	50.00
Call ATMF	10.39	32.73	43.35	29.56
Call 105	50.00	50.00	50.00	50.00
Call 110	50.00	50.00	50.00	50.00

Table 6.8: CVaR resampled efficient portfolios using 3-fold Exposure Stacking.

Readers can find all the details in the accompanying code and are encouraged to explore the results further. We note that in most practical applications, investment managers will be constrained by a particular α-CVaR with a target for the overall portfolio risk and tracking error risk as explained in Section 6.3. However, it is meaningful to put CVaR risk limits across multiple α values, for example, $\alpha \in \{90\%, 95\%, 97.5\%\}$ as in this case study.

We note that for CVaR optimization, the case study has used a lower number of resamples B simply because the problem takes longer time to solve, even with a semi-fast and semi-stable algorithm compared to the traditional linear programming presented in Section 6.2.1. We also use the efficient frontier to align the risk, because this is the only semi-fast CVaR implementation available to us. In practice, it is advised to use a target risk $\mathcal{R}_{target}$ and target tracking error $\mathcal{R}_{TE}$ problem formulation, but fast and stable solutions to these problems for CVaR are not freely available.

An example using the target risk $\mathcal{R}_{target}$ and target tracking error $\mathcal{R}_{TE}$ will be given for mean-variance optimization in Section 6.5 below. We have already seen how to solve this problem using second-order cone programming in the accompanying code to Section 6.3.1. Although the dual risk CVaR optimization problem is really hard to solve in a fast and stable way, which is crucial for resampled portfolio optimization, it is practically feasible, while the algorithms are highly specialized and require very careful implementation.

6.4.4 Perspectives on Resampled Portfolio Stacking Targets

In the previous sections, we have used various stacking targets. To define them formally, we define the stacking target x as

$$x = t\left(e, R, p\right), \tag{6.4.4}$$

where $e \in \mathbb{R}^I$ are portfolio exposures, while R and p are from the market representation with fully general Monte Carlo distributions and associated joint scenario probabilities from (1.1.1). We use p in (6.4.4), but the probability vector might as well be an Entropy Pooling posterior probability vector q.

Which target (6.4.4) is the most meaningful depends on the application. The reader has full flexibility in terms of determining which target makes sense for them. It is generally recommended to use targets that include the marginal risk contributions presented in Section 7.1, because these take into account the differences in standalone risk as well as potentially complex diversification interactions. We note that the risk contributions do not necessarily have to stem from the exposures e, they could also be from risk factors.

The choice of stacking target (6.4.4) is by no means trivial and should be done with care. For example, in Section 6.4.1 we have used $x = \mu \oslash \mathcal{R}^{'}\left(e\right)$, the ratio between marginal return and risk contributions. The astute reader might have noticed that this is not mathematically defined when the marginal risk contribution $\frac{\partial \mathcal{R}(e)}{\partial e_i}$ is zero. We could still technically do it from a computational perspective due to the way real numbers are stored in computer memory. Although it did not result in any issues in our case studies, it might lead to issues in other cases.

Given the above issues with the ratio between marginal risk and return contributions, it is worth considering whether the target that includes these quantities should be formulated as a combined vector of the two or as the difference between the elements. With these formulations, there will be no conceptual mathematical issues or practical issues from dividing by a number that is very close to zero. However, it of course introduces some implicit weighting, depending on the magnitude of the marginal risk and return contributions.

When we optimize over multiple risk measures, or the same risk measure with different hyperparameters like in Section 6.4.3, it introduces some additional nuances to the stacking objective. While we used Exposure Stacking in Section 6.4.3, we could in principle have used a combined vector containing the marginal risk contributions to α-CVaR with $\alpha \in \{90\%, 95\%, 97.5\%\}$. The interested reader is encouraged to perform this analysis. With multiple risk measures, we could also decide that there is one which is more important to us than the others and use the marginal risk contributions from this risk measure as the Resampled Portfolio Stacking target. The book's general recommendation is to stick to CVaR, because it has nice properties.

From the case study in Section 6.4.1, we noticed that the number of folds hyperparameter L affects the various targets x differently. Hence, when using a new target it is especially important to examine how it is affected by this parameter. It is generally recommended that L is determined by its out-of-sample performance by evaluating the results on new samples as done in Section 6.4.1. Finally, we underline that the Resampled Portfolio Stacking approach is still very new. Hence, the magnitude of the risk-adjusted return gains must still be assessed by extensive practical use of the method, which readers are encouraged to explore in creative ways.

6.5 Portfolio Rebalancing

Portfolio rebalancing is often approached in an ad hoc way without a clear way of thinking about it. Having introduced Resampled Portfolio Stacking in Section 6.4.1, we have a natural way of measuring similarities between portfolios with the Euclidean norm, which can be applied to any suitable target $x_b^\star$. For example, we can measure the distance between portfolios based on their marginal risk contributions $\mathcal{R}^{'}(e) \odot e$ or a combined vector of the marginal return and risk contributions $\left(\mu \odot e, \mathcal{R}^{'}(e) \odot e\right) \in \mathbb{R}^{2I}$. Note that we purposefully avoid using the ratio between the marginal risks and returns $\mu \oslash \mathcal{R}^{'}(e)$ due to the potential issues with this target, presented in Section 6.4.4.

When we introduce market model or parameter uncertainty into the portfolio optimization problem, we are generating a distribution of optimal portfolio exposures $e_b^\star$ for the particular constraints, risk measure, and risk targets. We can use these sample exposures $e_b^\star$ to generate a distribution for an Euclidean norm metric, allowing us to assess the distance between our current portfolio e_0 and a desired portfolio e_{target} that we are considering rebalancing towards.

The above reasoning is not entirely new as these thoughts were initially introduced by Michaud and Michaud (1998). However, the original resampled approach is focused on mean-variance optimization and the efficient frontier, aligning the portfolio risks through the index on the efficient frontier. The method introduced in this sections goes well beyond that and works for general risk measures $\mathcal{R}(e)$ as well as portfolio optimization with a portfolio risk target $\mathcal{R}_{target}$, a tracking error target $\mathcal{R}_{TE}$, or both. Risk and return alignment through the index on the efficient frontier is of course also still possible.

The resampled rebalancing target (6.4.4) can simply be the exposures e, following the original Exposure Stacking suggestion presented in Section 6.4. However, it can also be the marginal return contributions $\mu \odot e$, the marginal risk contributions $\mathcal{R}^{'}(e) \odot e$, or a combined vector of the marginal return and risk contributions $\left(\mu \odot e, \mathcal{R}^{'}(e) \odot e\right) \in \mathbb{R}^{2I}$ as suggested in this section. All targets align with the general Resampled Portfolio Stacking perspectives introduced in Section 6.4.1, while only the last two follow the marginal risk recommendation from Section 6.4.4.

We formally define a resampled test statistic as

$$\bar{x}_0 = \left\| x_{target} - x_0 \right\|_2$$

and its resampled distribution as

$$\bar{x}_b = \left\| x_{target} - x_b \right\|_2, \quad b \in \{1, 2, \ldots, B\},$$

where $x_{target} = t(e_{target}, R, p)$, $x_0 = t(e_0, R, p)$, and $x_b = t(e_b, R, p)$.

The idea is then to perform B resampled portfolio optimizations with risk constraints defined by e_{target}, i.e., setting $\mathcal{R}_{target} = \mathcal{R}(e_{target})$ and $\mathcal{R}_{TE} = \mathcal{R}(e_{target} - e_{BM})$ as well as the portfolio's usual exposure constraints $e \in \mathcal{E}$. This will give us a distribution for the optimal Euclidean norm deviations $\bar{x}_b$ with the particular constraints and market model/parameter uncertainty. We can then test the null hypothesis that we do not need to rebalance by simply counting the number of times that $\bar{x}_b \geq \bar{x}_0$ and divide this by B. If the proportion of Euclidean norm deviations $\bar{x}_b$ above $\bar{x}_0$ is high, this indicates that our current exposures e_0 are not far from the target exposures e_{target}. If the proportion is sufficiently

low, for example at 5%, this indicates that our current exposures e_0 are far from the target exposures e_{target}. The above approach is coined Resampled Portfolio Rebalancing, and the proportion of $\bar{x}_b < \bar{x}_0$ the Resampled Rebalancing Probability.

The convenient feature of the Euclidean norm is that it clearly measures the deviation from our target in the sense that a higher Euclidean norm represents a larger deviation. If we measured the deviations directly by, for example, the portfolio's CVaR, which can be both negative and positive, we do not have this uniformity in interpretation. Other measures of deviation can of course also be used, while it is important that they preserve the uniformity in interpretation that allows us to do one-sided tests. The original suggestion by Michaud and Michaud (1998) is to use the volatility tracking error, but this is related to their focus on mean-variance optimization, which we are not constrained to. Note also that we can operate with multiple targets x_b and their joint distribution if we want.

Continuing along the lines of the analysis from Section 6.4.4, we focus on a combined vector of the marginal return and risk contributions, i.e., $x = \left(\mu \odot e, \mathcal{R}^{'}(e) \odot e\right)$. The accompanying code to this section shows you how to perform the rebalancing test for mean-variance optimization, because it is easy and fast with an implementation of the dual risk objective already available from Section 6.3.1. In practice, it is always recommended not to reduce the market model to a mean vector and covariance matrix as well as optimize over CVaR as argued throughout the book and in Section 6.2.

We use the portfolio from Table 6.1 in Section 6.3.1 as the current allocation e_0, while using a target portfolio e_{target} with an overall volatility of 7% and a tracking error of 2% to the equally weighted benchmark e_{bm}. We use $B = 1,000$ resamples and $N = 100$ samples to introduce uncertainty into the means. See Table 6.9 for an overview of the portfolios that we work with.

	Target	Current	Benchmark
Gov & MBS	3.19	0.00	10.00
Corp IG	0.00	0.00	10.00
Corp HY	0.52	0.00	10.00
EM Debt	25.00	18.39	10.00
DM Equity	3.25	0.00	10.00
EM Equity	0.00	0.00	10.00
Private Equity	10.38	6.61	10.00
Infrastructure	16.01	25.00	10.00
Real Estate	16.66	25.00	10.00
Hedge Funds	25.00	25.00	10.00

Table 6.9: Portfolio overview for the rebalancing case study.

Figure 6.5.1 shows the sampling histogram of $\bar{x}_b$ as well as the test statistic $\bar{x}_0$. We note that this depends on the target x_b, the risk measure $\mathcal{R}$, the portfolio exposure constraints $\mathcal{E}$, and the market model/parameter uncertainty introduced by the samples. When we include the marginal risks in the target x, we take into account the different diversification effects present in the portfolios. Hence, the new rebalancing framework is very flexible for assessing the distance between the target exposures e_{target} and the current exposures e_0, and it works for fully general parameter uncertainty.

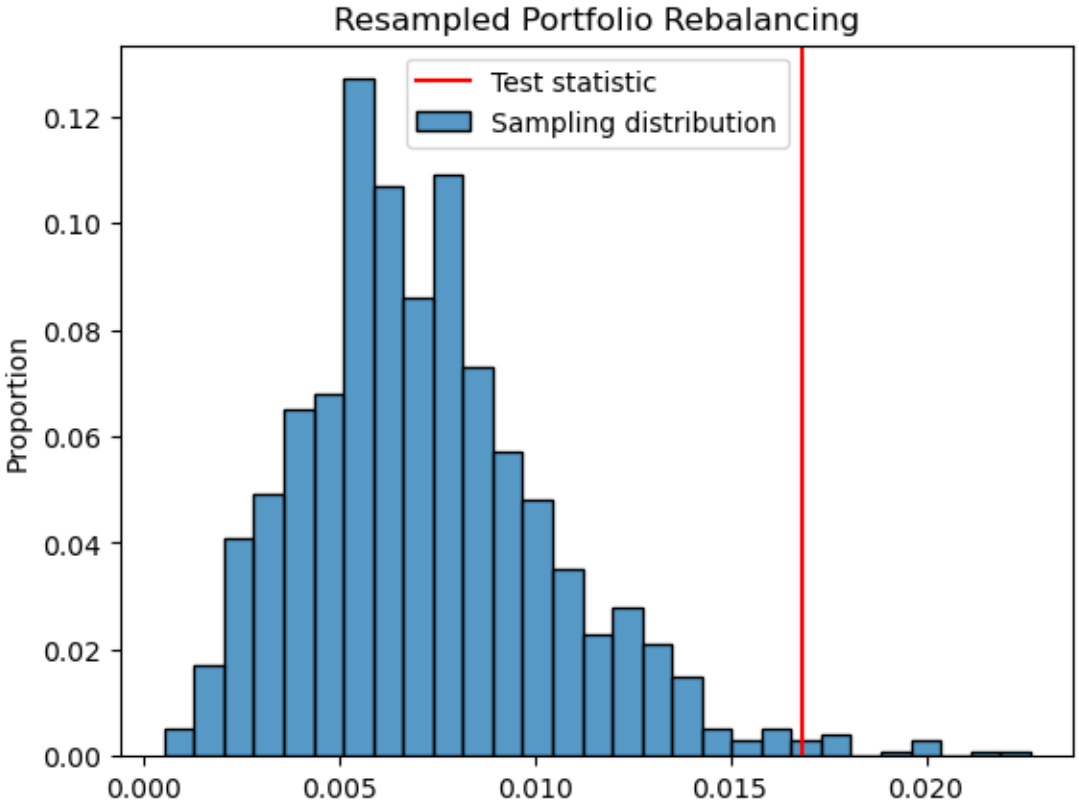

Figure 6.5.1: Sampling distribution for $\bar{x}_b$ and the test statistic $\bar{x}_0$.

The histogram in Figure 6.5.1 is a convenient way of visualizing how the Resampled Portfolio Rebalancing test is performed. In practice, it is sufficient to look at the proportion where $\bar{x}_b \geq \bar{x}_0$. We call this proportion the Resampled Portfolio Rebalancing p-value. In this particular case study, the p-value is 1.1%, while the rebalancing probability is 98.9%, see the accompanying code to this section for all computation details.

While we have so far focused on the investable portfolio exposures e, there is nothing preventing us from analyzing the portfolio's risk and return from a risk factor perspective. Hence, we can apply the Resampled Portfolio Rebalancing approach to risk factor contributions. This, however, requires that we are able to perform such a risk factor decomposition of the portfolio's risk and return. More perspectives on this will be given in Chapter 7 below.

Chapter 7

Risk and Return Analysis

This chapter presents fundamental methods and perspectives for general risk and return analysis. While these methods are frequently used by practitioners, they are often ignored by academics, who seem to think that portfolio construction is mainly about estimating covariance matrices and performing mean-variance optimization.

As mentioned in the previous Chapter 6 about portfolio optimization, many practitioners are in fact so skeptical about portfolio optimization that they do not perform it explicitly and mostly rely on a risk allocation exercise using the marginal contributions to risk as presented in Section 7.1. The risk allocation exercise can naturally be combined with the views and stress-testing methods presented in Chapter 5. As we have seen in Section 6.3, the Sequential Entropy Pooling method can also naturally be used to stress-test the diversification assumptions for such risk budgeting exercises.

While many investment managers claim to only do risk budgeting, the reality is that expected return assumptions have a tendency to sneak themselves into the allocation in implicit ways. So the problem of estimating expected returns is not ignored in practice, these estimates are just rarely fed into a mean-variance optimizer and used for actual investment management. As discussed throughout this book, mean-variance has many shortcomings that quickly show up and lead to undesired outcomes in practice. It is hard to separate practitioners' skepticism towards mean-variance from the skepticism towards portfolio optimization in general. As Chapter 6 shows, there exist practically feasible solutions for handling fully general parameter uncertainty issues, and practitioners frequently use some variant of the resampled portfolio optimization approach.

Risk parity approaches that distribute the risk contribution evenly from the individual investments or some factor representation have been promoted a lot in recent years. However, the implicit assumption is that the marginal return contributions also happen to be the same for each instrument. This is probably rarely true in reality, and many practitioners are not even able to implement risk parity portfolios given their investment constraints. Hence, risk parity is often more theoretical than practical. If you have a portfolio containing very complex investment strategies with strong alpha signals where the expected return estimation and market modeling might be extremely challenging, it might make sense to allocate roughly the same amount of risk to each strategy. In all other cases, it is recommended not to ignore the expected returns and perform portfolio optimization including parameter uncertainty as presented in Chapter 6.

7.1 Marginal Risk and Return Contributions

Marginal risks are defined as the gradient vector of the investment risk measure $\mathcal{R}$ with respect to the individual exposures e, i.e.,

$$\mathcal{R}'(e) = \begin{pmatrix} \frac{\partial \mathcal{R}(e)}{\partial e_1} \\ \frac{\partial \mathcal{R}(e)}{\partial e_2} \\ \vdots \\ \frac{\partial \mathcal{R}(e)}{\partial e_I} \end{pmatrix} \in \mathbb{R}^I.$$

For investment risk measures that are homogeneous of degree one, it holds that

$$\mathcal{R}(e) = \sum_{i=1}^{I} \frac{\partial \mathcal{R}(e)}{\partial e_i} e_i = \mathcal{R}'(e)^T e, \tag{7.1.1}$$

see Meucci (2007). That the risk measure is homogeneous of degree one simply means that if we double all the individual exposures, the risk will also double. This will hold for the investment risk measures that we consider, in particular CVaR and variance.

We call the elements

$$\mathcal{R}'(e) \odot e = \left(\frac{\partial \mathcal{R}(e)}{\partial e_i} e_i \right)_i \in \mathbb{R}^I \tag{7.1.2}$$

the marginal risk contributions. It is this quantity that we use in the Risk Stacking approach from Section 6.4.1.

For variance, the marginal risks can be computed simply as

$$\mathcal{R}'(e) = \frac{\Sigma e}{\sqrt{e^T \Sigma e}}.$$

There is also a slightly more complicated formula for general CVaR marginal risks based on the Monte Carlo market simulation R and associated probability vectors p and q from 1.1.1, but it is generally fine to compute the partial derivatives numerically, i.e.,

$$\mathcal{R}'(e) \approx \left(\frac{\mathcal{R}(e_{i,0} + \Delta) - \mathcal{R}(e_{i,0})}{\Delta} \right)_i$$

for some small $\Delta > 0$, e.g., $\Delta = 0.000001$. A convenient feature of the numerical approach is that you only have to know how to compute the risk of a portfolio. You do not have to know how to derive the analytical marginal risk contribution formula for that particular risk measure. It also ensures consistency in the way the risk is computed, e.g., when it comes to interpolation methods for VaR and CVaR.

Marginal relative returns are simply given by the mean vector

$$\mu \in \mathbb{R}^I,$$

while the marginal return contributions are given by

$$\mu \odot e = (\mu_i e_i)_i \in \mathbb{R}^I.$$

It should be easy to convince yourself that this is indeed the case. In a rebalancing or portfolio optimization application with transaction costs, we probably want to account for the transaction cost as well when computing the marginal return contributions of the portfolios we are trading towards.

It might also be interesting to analyze the marginal risk-adjusted returns defined as

$$(\mu \odot e) \oslash \left(\mathcal{R}^{'}(e) \odot e\right) = \mu \oslash \mathcal{R}^{'}(e) \in \mathbb{R}^{I},$$

where $\oslash$ denotes the element-wise Hadamard division. We note that the marginal risk-adjusted returns do not follow the decomposition from (7.1.1), and that they are only defined for the elements where $\frac{\partial \mathcal{R}(e)}{\partial e_i} \neq 0$ for $i \in \{1, 2, \ldots, I\}$. It is worth computing this quantity to assess if the ratio between expected marginal return and risk contribution is particularly attractive for some exposure i. It is also interesting to assess these ratios for risk parity investors to evaluate whether they have included sources of risk in their portfolio with a very low contribution to the expected return.

Another example of how marginal risk contribution analysis can be combined with the views and stress-testing from Chapter 5 is a case where we implement a CVaR stress-test for developed market equities instead of the portfolio, as we do in Section 5.1.3, and assess its effect on the marginal risk contributions. The case study in this section uses a proprietary implementation, but it is a good exercise for readers to try to replicate it with an Entropy Pooling CVaR stress-test, a computation of the portfolio CVaR, and a CVaR risk decomposition as in equation (7.1.2).

Figure 7.1.1 shows the stress-tests, where we have increased the 90%-CVaR of developed market equities by 50%. Figure 7.1.1 also illustrates the effect from the CVaR stress-test on corporate high-yield bonds. We note that Entropy Pooling not surprisingly predicts more tail risk for corporate high yield as a consequence of the increased equity tail risk. Table 7.1 shows the portfolio that we compute the marginal risk contributions (7.1.2) for as well as their posterior and prior values. From Table 7.1, we note that many interesting interaction effects happen. For example, the 90%-CVaR contribution from equities increases significantly more than 50%. This effect stems from the fact that other assets with a positive correlation with equities also have an increase in CVaR, which makes the DM equity allocation even less diversifying. The important point here is that the marginal risk contributions (7.1.2) include several complex interaction effects. Figure 7.1.2 visualizes them.

	Exposure	Prior 90%-CVaR	Stress-test 90%-CVaR
Gov & MBS	30.00%	0.62%	0.66%
Corp IG	10.00%	0.27%	0.27%
Corp HY	10.00%	1.11%	1.73%
EM Debt	5.00%	0.31%	0.45%
DM Equity	22.00%	3.18%	6.01%
EM Equity	3.00%	0.92%	1.37%
Private Equity	5.00%	1.05%	1.83%
Infrastructure	5.00%	0.38%	0.78%
Real Estate	5.00%	0.07%	0.15%
Hedge Funds	5.00%	0.30%	0.65%
Sum	100.00%	8.21%	13.90%

Table 7.1: Overview of the portfolio exposures and 90%-CVaR contributions.

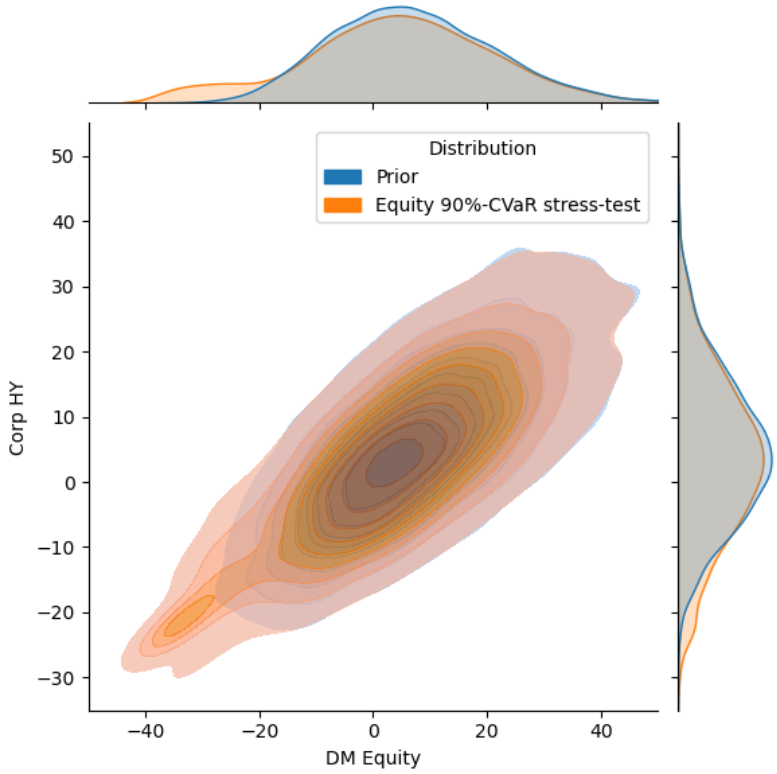

Figure 7.1.1: Joint distribution for DM Equity and Corp HY after a 90%-CVaR stress-test.

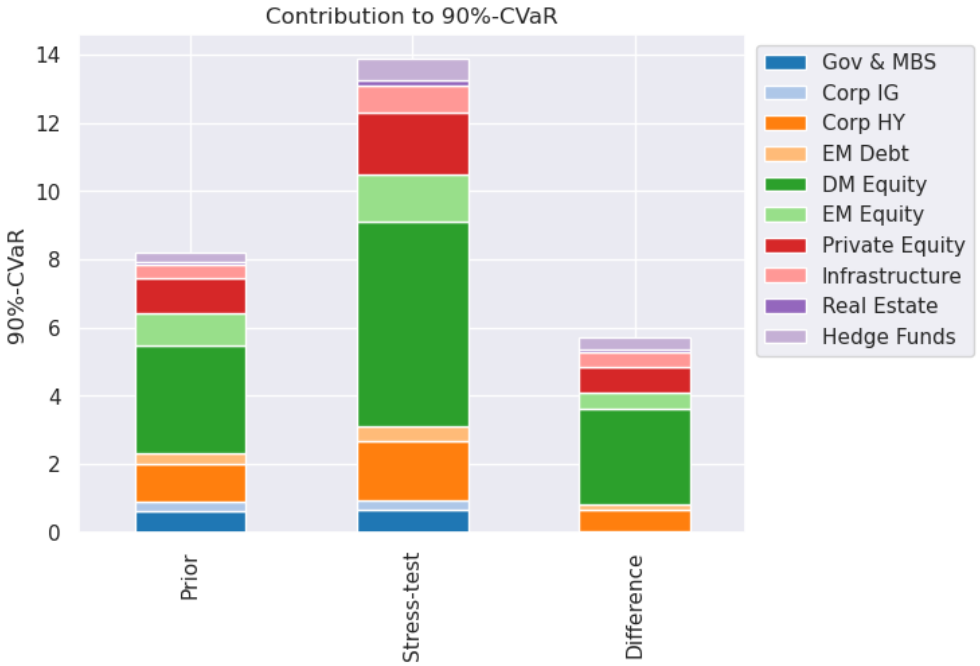

Figure 7.1.2: Comparison of the marginal contributions to 90%-CVaR.

7.2 Market Views vs Stress-Tests

The terms market views and stress-tests are frequently used by investment practitioners, although there is usually no precise definition. We can loosely define views as minor adjustments to the market model, while stress-tests are specific adverse scenarios that we want to examine. While both are meaningful to analyze from a (marginal) risk and return perspective as presented in Section 7.1, stress-tests are probably what we want to build tail risk hedges for as presented in Section 7.3.

In this section, we will see some examples of what we can rightfully call market views, and what should be characterized as a stress-test. We keep the market simulation very simple with a log-normal simulation for equities and bonds as well as introduce a stress-test for the classical 60/40 portfolio. The stress-test example will also illustrate a case where the Sequential Entropy Pooling algorithms from Section 5.2 are capable of solving practically interesting problems that the original suggestion from Meucci (2008a) of always using the prior value when necessary cannot.

We make the following assumption for the log-return of bonds and equities:

$$\mu = \begin{pmatrix} 0.04 \\ 0.10 \end{pmatrix} \quad \text{and} \quad \Sigma = \begin{pmatrix} 0.1^2 & -0.3 \cdot 0.1 \cdot 0.2 \\ \bullet & 0.2^2 \end{pmatrix}.$$

Figure 7.2.1 shows a joint plot for this prior log-normal distribution, see the accompanying code for how the simulation and plot was created.

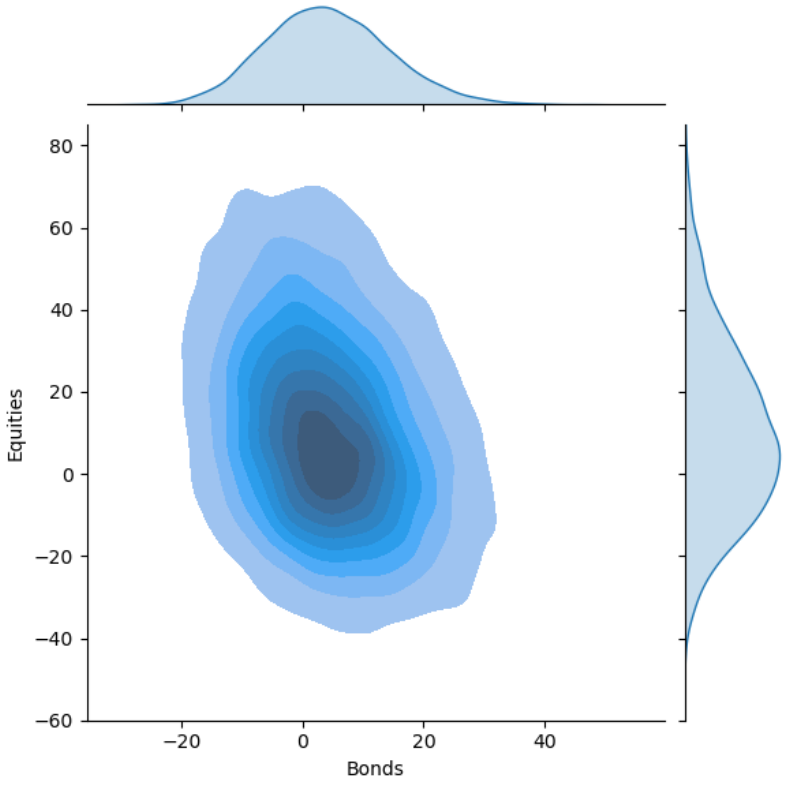

Figure 7.2.1: Prior log-normal simulation for bonds and equities.

We first implement a view on equities, setting the expected return to 7.5% and the volatility to 27.5%. This can rightfully be called a market view, because the adjustment is not that significant, which is evident by a low relative entropy of 7.08% and a high effective number of scenarios of 93.16%, see equation (5.1.2). You can see how these views have been implemented and the statistics computed in the accompanying code to this section.

Next, we implement an interesting stress-test using Sequential Entropy Pooling from Section 5.2. We first calculate the 90%-CVaR of the 60/40 equity/bond portfolio to be 11.68%. We implement this as the expected return for the portfolio as a $\mathcal{C}_0$ view. After the first step, we update the means and volatilities of bonds and equities and finally implement the $\mathcal{C}_4$ correlation view that it should be equal to 30%.

Sequential Entropy Pooling is needed for the stress-test because we must update the means and volatilities of bonds and equities before we implement the correlation view. If we kept them at their old values when implementing the correlation view, this would lead to logical inconsistencies or, at the very least, very different volatilities in the correlation view. Hence, this practically relevant tail risk stress-test is not possible to implement with the original Entropy Pooling heuristic that simply fixes parameters to their prior values when necessary.

Figure 7.2.2 shows the joint prior and posterior distribution for the stress-test view. So, this is a case where we are in the left tail of our portfolio's return distribution and diversification fails, i.e., a tail risk scenario if there ever was one. Perspectives on how to analyze and work with tail risk hedging are given in Section 7.3 below. Figure 7.2.2 is generated using a proprietary implementation, but readers are encouraged to replicate the plot in the accompanying code. Some initial code for the prior distribution is provided as a starting point.

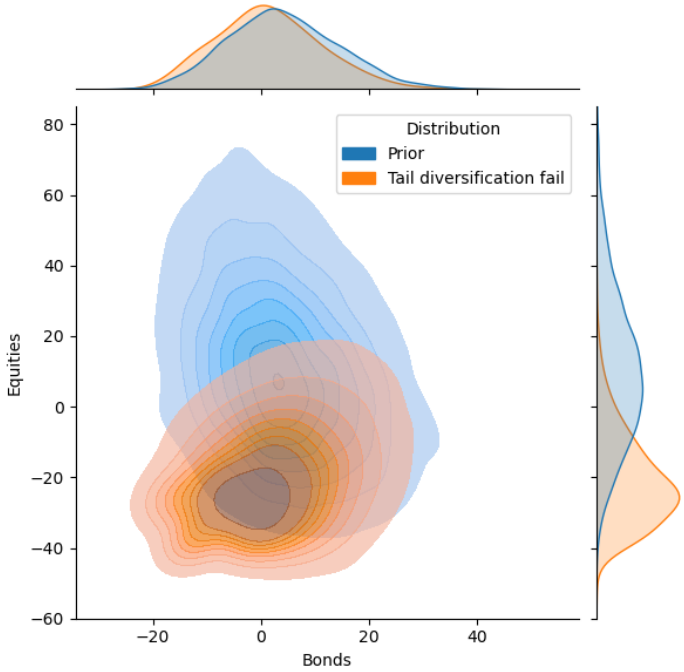

Figure 7.2.2: Prior and posterior distribution for the equity and bonds stress-test.

7.3 Tail Risk Hedging

As Section 2.3 about the volatility risk premium shows, outright tail risk hedging using instruments that are guaranteed to introduce downside convexity, such as options, is usually expensive. This is due to investors' risk aversion to significant loss scenarios. The counterparts that provide such loss protection also have an aversion to these events and therefore require a compensation beyond the break-even value for providing us with such downside convex payoffs. Hence, a strategic allocation to put options on your portfolio will probably be a significant performance drag. If that is not the case for your particular portfolio, you can go ahead and outright hedge the downside strategically, but it is unlikely to hold true in reality. If you happen to be particularly worried about the downside over some short period of time, you can of course tactically consider equipping your portfolio with an outright tail risk hedge, being aware that this is a very challenging timing task.

Due to the volatility risk premium, many investors combine a generally well-diversified portfolio with strategies that probably provide some downside convexity, for example, investing in trend-following strategies, as introduced in Section 4.4. It is important to note that these are not guaranteed to provide a positive payoff during a market sell-off, so the hedge is statistical and indirect in this case.

Another approach to tail risk hedging, which this book generally recommends, is to consider the scenarios where we are in the left tail of the portfolio's return distribution and diversification fails, as in the last example of Section 7.2. There can be many such scenarios, and they are unique to individual portfolios. Hence, providing general rules for how to identify them is challenging. There might, however, exist general mathematical methods for how to isolate the diversification factors in a portfolio and subsequently stress-tests these directly using Sequential Entropy Pooling. Investment managers usually have a good sense of which diversification fail scenarios they are worried about, so these can readily be implemented using Sequential Entropy Pooling like in Section 7.2.

If we are only worried about one tail risk scenario, perhaps what we believe is the most severe one, we focus on building a tail risk for this particular scenario. We can then use a state probability $c \in [0, 1]$ as presented in Section 5.3 to combine the tail risk scenario probability vector q_{tail} with the prior probability p or some other base case posterior probability q_{base}. This gives us a posterior probability $q = cq_{tail} + (1 - c)\, q_{base}$ that we can use for the final optimization of the portfolio including an appropriately sized tail risk hedge strategy.

If we are worried about N different tail risk scenarios, we can conveniently use Bayesian networks as presented in Section 5.4 to keep track of the state probabilities of each of these tail risk scenarios $c_n \in [0, 1]$ for $n \in \{1, 2, \ldots, N\}$ with $\sum_{n=1}^{N} c_n = 1$. In this case, we create one node called "tail risk scenario" and assign probabilities to the N different scenarios. It is up to you as portfolio manager to figure out how to set these tail risk scenario probabilities, while you have full flexibility.

It is probably hard to come up with good rules for determining the tail risk scenarios in general, but if you want to introduce more structure for determining the tail risk scenario probabilities, you can add the key risk factors on top of the tail risk scenario node, similar to the market case example in Vorobets (2023). Figure 7.3.1 illustrates such a Bayesian network where real rates, inflation, and growth affects the tail risk scenario probability. This allows us to also condition on the realization of key risk factors and assess how it affects our tail risk hedging strategy. We can then use the final q_{tail} posterior probability vector to build the tail risk strategy.

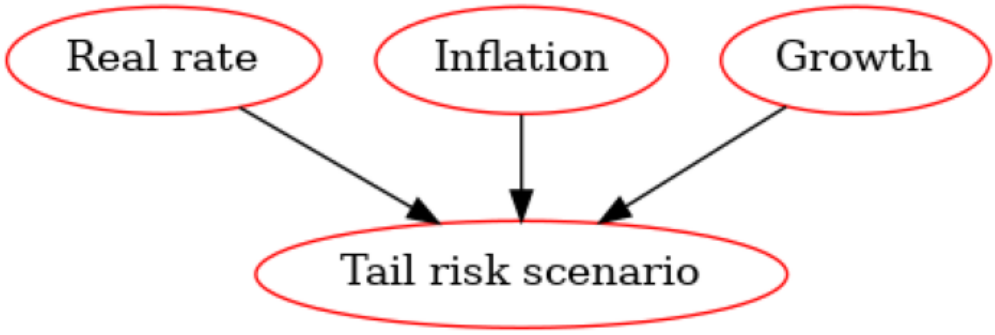

Figure 7.3.1: Bayesian net with real rate, inflation, and growth affecting tail risk scenario probability.

In summary, the objective of tail risk hedging is to get as much downside convexity as possible for the lowest price. In relation to evaluating the performance of the tail risk hedging, a put option on the portfolio's return with some out-of-the-money strike is a natural benchmark. This put option probably does not trade in the market, so you either have to price it yourself or get a counter part to quote a price. The put option is likely an upper bound on the convexity that you can expect from a good tail risk hedging strategy, while it should also be an upper bound on the performance drag that you experience from the tail risk hedge. If your tail risk hedging strategy ends up costing more than the put option with a worse payoff in the tail risk scenarios, then this is obviously a bad outcome, especially considering that the tail risk hedging strategy probably requires a lot more investment engineering.

A final perspective on tail risk hedging is that if you struggle to find good hedges that bring your level of tail risk to a satisfactory level after you have build a generally well-diversified portfolio, you should probably consider setting the strategic risk level lower for the portfolio. This might be a better and easier solution to achieve a better risk-adjusted return at a satisfactory level of risk. Some people also talk about the perspective that the tail risk allows you to take on more risk in the rest of the portfolio. While this statement depends on the perspective, it is important to avoid simply canceling the tail risk hedge by implementing close to opposite trades in an other layer of the portfolio. In the end, such an approach will probably just accumulate higher trading costs while remaining more or less the same from a risk factor exposure perspective.

Chapter 8

Summary and Comparison with the Variance-Based Approach

This final chapter summarizes the key points of the book as well as essential parts of the new investment framework. It makes it clear how the new framework and methods are upgrades to the current mainstream standard with (co)variance-based analysis and optimization. For each mainstream method, including covariance-based risk, the Black-Litterman model, and mean-variance, the book suggests an improvement. As explained in Section 8.1, the new approach is strictly better from a logical perspective. It is, however, much harder to implement a production-quality version of it, but it is practically feasible with current technology.

While the book has used mean-variance to illustrate investment concepts and principles, it has hopefully made it clear that mean-variance grossly oversimplify the portfolio construction problem. Hence, mean-variance should not be used to manage portfolios in practice. Underestimating the marginal tail risks of instrument P&L distributions and assuming that dependencies are cross-sectionally constant and linear is a clear recipe for disaster. If you use this approach, reality will eventually catch up to you and reveal the excessive tail risks that your portfolios have probably been exposed to.

A natural question is: if the new investment framework is strictly better, why is everyone not using it already? There are multiple reasons for this. The first one being that most people still do not know about it, or have a sufficiently good understanding of the methods to use them. Then comes the very significant implementation complexities. Finally, there are more subtle reasons related to human nature with many commercial and reputational interests formed around covariance matrix estimation, Black-Litterman, and mean-variance. While these methods are simple enough for most people to understand, the ease of use also eliminates any potential for portfolio construction alpha. It is simply too easy to replicate for mom-and-pop investors.

People who have commercial or reputational vested interests in the old methods do not want you to know about better alternatives. Some of them proactively engage in trying to limit your knowledge about the framework and methods presented in this book. A final subtle nuance is that some people simply do not like to admit that what they have been doing for a long time is in fact not very beneficial. Hence, there is significant nonscientific resistance and a bias to maintain the status quo.

If you do not directly benefit from maintaining the status quo, it is in your best interest to transition to methods that are going to help you get a better risk-adjusted return. In the long run, it is in the best interest of everyone to transition to something better than the old approach. Remember that investment markets are relentless feedback machines in the sense that they will simply let you know whether you managed your investments in a good or bad way. You will not even know why you did good or bad. You might have been lucky, or you might have been skillful. However, to increase your probability of being successful you have to infer reality well and build portfolios in a clever way based on this reality. This is what this book tries to help you do.

8.1 Comparison to Black-Litterman and Mean-Variance

We once again note that our market representation with fully general Monte Carlo simulations $R \in \mathbb{R}^{S \times I}$ and associated joint scenario probability vectors $p \in \mathbb{R}^S$ allows us to use elliptical distributions such as the normal or t-distribution. Hence, all the methods presented in this book will work for these distributions. The issue is rather that these distributions have very little to do with real-world market behavior as shown in Chapter 2.

Let us on the other hand imagine that we have more realistic simulations R and possibly associated Entropy Pooling posterior probability vectors q. If we use mean-variance analysis in that case, we are effectively removing many of the important nuances by reducing the market model to a covariance matrix and a mean vector. So, there is not much value in realistic market simulations if the subsequent analysis ignores many of the crucial nuances of these simulations. Hence, the market simulation and the subsequent optimization and analysis go hand in hand.

As shown in Section 6.2.1, practical applications of CVaR optimization and analysis operate directly on the market simulations R and associated probability vectors p or q. Hence, CVaR analysis and optimization will give meaningful results no matter how complex the market simulations and views/stress-tests are. Proposition 1 from Rockafellar and Uryasev (2000) shows that CVaR and variance optimization will coincide when return distributions are fully characterized by a mean vector and covariance matrix, and the expected return constraint is binding. The latter is usually the case in most practical cases. When CVaR is computed with demeaned P&L and return distributions happen to be elliptical, CVaR and variance optimization will always give the same results, see Vorobets (2022b). So, CVaR analysis gives the same results as mean-variance when the mean-variance assumptions are satisfied. Hence, mean-variance can be thought of as a small simple subset of mean-CVaR under the highly hypothetical elliptical distribution assumption. An elegant feature is that we do not have to make an assessment of whether the mean-variance assumptions are satisfied or not, mean-CVaR will automatically give us the correct results if they are.

The Black-Litterman (BL) model relies on additional assumptions on top of mean-variance by calibrating the prior model with an assumed risk aversion parameter and requiring an additional confidence parameter τ, which leads to several logical inconsistencies, see Meucci (2008b). Hence, the foundation of the BL model is also highly unrealistic, and it requires some questionable engineering to work. On the other hand, Entropy Pooling has a theoretically sound justification and allows us to easily implement views on higher moments as well as VaR and CVaR. Furthermore, view confidences

are handled in a natural probabilistic way as explained in Section 5.3. Entropy Pooling even has an analytical solution in the normally distributed case, see Meucci (2008a). So once again, there is no reason to continue using the old BL model when we have access to the new Entropy Pooling method.

In summary, there are no logical reasons to continue using BL and mean-variance when Entropy Pooling and mean-CVaR implementations are practically feasible. The continued justification of the old methods is based on irrational arguments or commercial and reputational vested interests. As an ambitious investment manager, you must not let these biases affect your investment performance.

8.2 Topics for Future Research

While the framework and methods proposed in this book are a major leap forward compared to the current standard with (co)variance-based risk, Black-Litterman, and mean-variance, it would be naive to think that they are end-of-civilization technologies. This section contains a list of suggested topics for future research that readers are encouraged to explore, possibly in collaboration with the author. The focus is initially on the methods introduced in this book and then subsequently on more general topics.

The first obvious study is further empirical analysis of the Resampled Portfolio Stacking approach introduced in Chapter 6, i.e., examining when it works well and the magnitude of the risk-adjusted return gains. While studying this, it would be interesting to examine the effect of various targets $x_b = t(e_b, R, p)$, how they are affected by the number of folds parameter L, and which cross-validation procedures work well for determining the optimal number of folds $L^\star$. It is also worth considering whether the Resampled Portfolio Stacking objective (6.4.3) can be further improved, but this is arguably more complicated analytical work.

Continuing with the Resampled Portfolio Stacking perspectives, it would be interesting to study optimization algorithms that can be combined with the Resampled Portfolio Rebalancing distributions from Section 6.5. For example, rebalance the portfolio towards the target exposures e_{target} subject to constraints on the Resampled Rebalancing Probability. A simple starting point is to optimize the portfolio with a CVaR tracking error budget given by $CVaR(e_{target} - e) \leq \tilde{\mathcal{R}}_{TE}$ and then examine which Resampled Rebalancing Probability the optimal exposures $e^\star$ lead to. The tracking error budget $\tilde{\mathcal{R}}_{TE}$ can, for example, be determined by using $x_b = CVaR(e_b - e_0, R, p)$ as a Resampled Portfolio Rebalancing target and then picking a value for $\tilde{\mathcal{R}}_{TE}$ that yields the desired Resampled Rebalancing Probability. This tracking error constraint allows us to find an optimal trade-off between risk and return in practical cases with transaction costs that might make full rebalancing too costly.

As explained in Section 5.2, the Sequential Entropy Pooling refinement is introduced to improve on the original Entropy Pooling heuristic of always using prior values when necessary to implement views as linear constraints on posterior probabilities. If we could solve the relative entropy minimization problem efficiently and in a stable way subject to general nonlinear constraints $\mathcal{G}(x) \leq h$ and $\mathcal{A}(x) = b$ on the posterior probabilities, we would do that. With the current state of optimization technology, this seems to be infeasible. There is also the issue that some interesting views might not be convex. Hence, we would have to solve the problem with potentially non-convex constraints on the posterior probabilities for a very large number of variables S. When we use Sequential Entropy Pooling to

formulate various views and stress-tests, convexity is guaranteed by the linear constraints, and solutions can be found in a fast and stable way.

Another interesting Entropy Pooling research area is to generalize the method to operate on fully general multi-period market simulations as presented in Chapter 3. Meucci and Nicolosi (2016) have initialized work on what they call Dynamic Entropy Pooling, but they unfortunately focus on just Ornstein-Uhlenbeck processes, which suffer from the issues presented in Section 3.2.3 when it comes to realistic market simulation. Hence, more research needs to be done to generalize the Entropy Pooling method for general multi-period market simulations $R_h \in \mathbb{R}^{S \times I}$, $h \in \{1, 2, \ldots, H\}$.

It is of course also interesting to extend the CVaR analysis to multiple periods, while the risk budgeting exercise from Section 6.3 is a practically good heuristic. It is important to note that this book is multi-period in its market simulation approach, because we obviously live in a multi-period world and want to be able to calculate the cumulative P&L of dynamic strategies. Few investment managers contest that. However, investment managers' appetite for multi-period optimization seems to be quite low. The standard is still one-period analysis with mean-variance, while some introduce the risk budgeting with tracking error constraints from Section 6.3 to handle signals with various horizons in a heuristic way. The value of multi-period optimization is probably greater for high-frequency investors, who usually also have a better understanding of their trading costs and market impact. For most other investment managers, one-period optimization with risk budgets and proportional transaction cost estimates seems to be preferred. A focus on fully general distributions and CVaR tail risks is a major upgrade to mean-variance in that case. The multi-period optimization reservations are caused by investment managers' skepticism about forecasting paths for expected returns and transaction costs, which might exaggerate the issues with mean uncertainty presented in Section 6.4.

As stated in Section 3.2.2 about generative machine learning for investment market simulation, this field is still in its infancy with many opportunities for significant improvements. The author's hypothesis is that in the same way that we have large language models (LLMs), we will have large market models (LMMs) in the future. While investment market generation is a very different field from text generation, the fundamental approaches can be the same, but the data preprocessing and architectures must necessarily be different. The author's hypothesis is that although there is a limited number of historical investment time series observations, a lot of transfer learning can happen in the cross-section, i.e., that we can learn something about the dynamics of the US government bond curve by also training the model on the dynamics of the German government bond curve. Indicator variables that classify the type of data might also be useful in this context. Finally, it would be interesting to examine a combination between generative machine learning methods from Section 3.2.2 and the Fully Flexible Resampling method introduced in Section 3.2.1, potentially getting the best of both worlds in terms of capturing very complex time series and cross-sectional dependencies.

In summary, while the framework from this book significantly improves on many of the core issues with the current investment risk and analysis standard, there are still many interesting research topics to explore. Not only does this research still need to be done, but these topics are in many cases very broad and, hence, outside the scope of this book. This book already contains many new results and perspectives for portfolio construction and risk management, which can be explored in many creative ways. Research-oriented readers are encouraged to work on solving these problems.

Bibliography

Artzner, P. et al. (1999). "Coherent Measures of Risk". *Mathematical Finance* 9.3, pp. 203–228.

Black, F. (1976). "The Pricing of Commodity Contracts". *Journal of Financial Economics*, pp. 167–179.

Black, F. and M. Scholes (1973). "The Pricing of Options and Corporate Liabilities". *Journal of Political Economy* 81.3.

Boyd, Stephen and Lieven Vandenberghe (2004). *Convex Optimization*. Cambridge University Press.

Breiman, Leo (1996). "Stacked Regressions". *Machine Learning* 24, pp. 49–64.

Caticha, A. and A. Giffin (2006). "Updating Probabilities". *Bayesian Inference and Maximum Entropy Methods in Science and Engineering Conference*.

Corani, G. and C. de Campos (2015). "A Maximum Entropy Approach to Learn Bayesian Networks from Incomplete Data". *Interdisciplinary Bayesian Statistics. Springer Proceedings in Mathematics & Statistics*. URL: https://doi.org/10.1007/978-3-319-12454-4_6.

Fama, E. F. and K. R. French (1992). "The Cross-Section of Expected Stock Returns". *Journal of Finance, American Finance Association* 47.2, pp. 427–465.

Friedman, D. et al. (2014). *Risky Curves: On the Empirical Failure of Expected Utility*.

Goodfellow, Ian J. et al. (2014). "Generative Adversarial Networks". *arXiv*. URL: https://arxiv.org/abs/1406.2661.

Hamilton, James D. (1994). *Time Series Analysis*. Princeton University Press.

Hastie, T., R. Tibshirani, and J. Friedman (2009). *The Elements ofStatistical Learning*. Springer.

Hull, J (2021). *Options, Futures, and Other Derivatives*. 11th ed. Pearson Education Limited.

James, Gareth et al. (2023). *An Introdution to Statistical Learning with Applications in Python*. Springer.

Kingma, D. P. and M. Welling (2013). "Auto-Encoding Variational Bayes". *arXiv*. URL: https://arxiv.org/abs/1312.6114.

Kingma, D. P. and M. Welling (2019). "An Introduction to Variational Autoencoders". *Foundations and Trends in Machine Learning*.

Kristensen, L. and A. Vorobets (2024). "Portfolio Optimization and Parameter Uncertainty". *SSRN*. URL: https://ssrn.com/abstract=4709317.

Krokhmal, P., J. Palmquist, and S. Uryasev (2002). "Portfolio Optimization with Conditional Value-at-Risk Objective and Constraints". *Journal of Risk* 4, pp. 43–68.

Lahiri, S. N. (2003). *Resampling Methods for Dependent Data*. Springer Series in Statistics.

Markowitz, H. (1952). "Portfolio Selection". *The Journal of Finance* 7.1, pp. 77–91.

Markowitz, H. (1959). *Portfolio Selection: Efficient Diversification of Investments*. Yale University Press.

McCoy, J., S. Kroon, and L. Auret (2018). "Variational Autoencoders for Missing Data Imputation with Application to a Simulated Milling Circuit". *IFAC-PapersOnLine* 51.21, pp. 141–146.

Meucci, A. (2007). "Risk Contributions from Generic User-Defined Factors". *The Risk Magazine*, pp. 84–88. URL: https://ssrn.com/abstract=930034.

Meucci, A. (2008a). "Fully Flexible Views: Theory and Practice". *Risk* 21.10, pp. 97–102. URL: https://ssrn.com/abstract=1213325.

Meucci, A. (2008b). "The Black-Litterman Approach: Original Model and Extensions". URL: https://ssrn.com/abstract=1117574.

Meucci, A. (2012a). "Effective Number of Scenarios in Fully Flexible Probabilities". *GARP Risk Professional, February 2011*, pp. 32–35. URL: https://ssrn.com/abstract=1971808.

Meucci, A. (2012b). "Stress-Testing with Fully Flexible Causal Inputs". *Risk*. URL: https://ssrn.com/abstract=1721302.

Meucci, A. (2013). "Estimation and Stress-Testing via Time- and Market-Conditional Flexible Probabilities". *SSRN*. URL: https://ssrn.com/abstract=2312126.

Meucci, A. (2014). "Linear Factor Models: Theory, Applications and Pitfalls". *SSRN*.

Meucci, A., D. Ardia, and S. Keel (2011). "Fully Flexible Extreme Views". *Journal of Risk* 14.2, pp. 39–49. URL: https://ssrn.com/abstract=1542083.

Meucci, A. and M. Nicolosi (2016). "Dynamic Portfolio Management with Views at Multiple Horizons". *Applied Mathematics and Computation* 274, pp. 495–518.

Michaud, R. and R. Michaud (1998). *Efficient Asset Management: A practical Guide to Stock Portfolio Optimization and Asset Allocation*. Oxford University Press.

Munk, C. (2011). *Fixed Income Modelling*. Oxford University Press.

Norris, J. R. (1997). *Markov Chains*. Cambridge University Press.

Penman, S. (2012). *Financial Statement Analysis and Security Valuation*. 5th. McGraw-Hill Education.

Rebonato, R. and A. Denev (2011). "Coherent Asset Allocation and Diversification in the Presence of Stress Events". *Journal of Investment Management*. URL: https://ssrn.com/abstract=1824207.

Rebonato, R. and A. Denev (2014). *Portfolio Management under Stress: A Bayesian-Net Approach to Coherent Asset Allocation*. Cambridge University Press.

Rockafellar, R. T. and S. Uryasev (2000). "Optimization of Conditional Value-at-Risk". *Journal of risk* 2, pp. 21–42.

Rockafellar, R. T., S. Uryasev, and M. Zabarankin (2006). "Generalized Deviations in Risk Analysis". *Finance and Stochastics* 10.1, pp. 51–74.

Ross, S. (1976). "The Arbitrage Theory of Capital Asset Pricing". *Journal of Economic Theory* 13, pp. 341–360.

Salimans, T. et al. (2016). "Improved Techniques for Training GANs". *Advances in Neural Information Processing Systems*.

Sharpe, W. F. (1964). "Capital Asset Prices: A Theory of Market Equilibrium Under Conditions of Risk". *The Journal of Finance* 19.3, pp. 425–442.

Vorobets, A. (2021). "Sequential Entropy Pooling Heuristics". *SSRN*. URL: https://ssrn.com/abstract=3936392.

Vorobets, A. (2022a). "Portfolio Management Framework for Derivative Instruments". *SSRN*. URL: https://ssrn.com/abstract=4217884.

Vorobets, A. (2022b). "Variance for Intuition, Cvar for Optimization". *SSRN*. URL: https://ssrn.com/abstract=4034316.

Vorobets, A. (2023). "Causal and Predictive Market Views and Stress-Testing". *SSRN*. URL: https://ssrn.com/abstract=4444291.

Vorobets, A. (2024). "Derivatives Portfolio Optimization and Parameter Uncertainty". *SSRN*. URL: https://ssrn.com/abstract=4825945.

Wolpert, David H. (1992). "Stacked Generalization". *Neural Networks* 5, pp. 241–259.

Yoon, J., D. Jarrett, and M. Van der Schaar (2019). "Time-Series Generative Adversarial Networks". *Advances in Neural Information Processing Systems*.