MMF1921: Operations Research

Project 2 (Winter 2024)

Group Members: Wenkai(Kai) Fu, Xingyu Lu, Mingwei (Grace) Shi

1 Introduction

This project aims to develop an automated asset management system using advanced factor models and optimization techniques. Leveraging a dataset of adjusted closing prices for multiple assets and corresponding factor returns, we will design and test a trading algorithm capable of maintaining robust performance across varying market conditions.

In this project, we will employ several key factor models, including the Ordinary Least Squares (OLS) model, the Fama-French three-factor (FF) model, the Least Absolute Shrinkage and Selection Operator (LASSO) model, and the Best Subset Selection (BSS) model. These models will be used to estimate parameters crucial for portfolio optimization. This is detailed in **Section 2**.

The factors used in the factor models are listed in Table 1. Some models will utilize all these factors, while others may use a selected subset, as detailed in **Section 2**.

Table 1: List of factors

Market ('Mkt_RF')	Size ('SMB')	Value ('HML')	Short-term reversal ('ST_Rev')
Profitability ('RMW')	Investment ('CMA')	Momentum ('Mom')	Long-term reversal ('LT_Rev')

Our approach involves implementing and comparing various portfolio optimization techniques: Mean-Variance Optimization (MVO), Robust Mean-Variance Optimization, Conditional Value at Risk (CVaR) Optimization, and Risk Parity Optimization. The methodology of each optimization model is explained in the **Section 3**.

The goal of the project is to construct a portfolio that maximizes the Sharpe ratio while minimizing the turnover rate, as these are the primary metrics used for evaluating financial performance in this project. The detialed evaluation method and results will be in **Section 4**.

The project will follow a structured methodology, starting with the calibration and estimation of models using a five-year in-sample period. The investment strategies will be rebalanced every six months and will be tested against unseen datasets to assess their out-of-sample performance. The effectiveness of each strategy will be measured not only by financial metrics but also by computational runtime. Detailed analysis of training, validation, and testing will be provided in **Section 4** as well, including comparisons of trading strategies for each optimization method and parameter tuning for optimal strategies.

Finally, **Section 5** will present the testing results, illustrating the performance of each model through financial metrics and visualizations of portfolio values and allocations.

2 Factor Models

In financial analysis, factor models are pivotal for measuring and attributing an asset's risk and returns based on its exposure to various market factors. This study focuses on modeling the excess returns of assets, defined as the asset returns over the risk-free rate. The monthly risk-free rate was subtracted from asset returns to isolate the performance attributable to the asset itself rather than the market.

The factors used in this model (as detailed in Table 1 from introduction) were derived from synthetic data representing different characteristics of asset classes. These factors help in identifying shared traits across different portfolios, providing a structured approach to asset evaluation. While there is inherent correlation among these factors, it's crucial to note that they might not fully conform to theoretical model conditions. Nevertheless, the covariance of these factors offers significant insights into the underlying risk components of the assets. Consequently, these covariances are integrated into our calculations for the asset covariance matrix, enhancing the robustness of our risk assessments.

2.1 Ordinary Least Squares (OLS) Model

To quantitatively assess the impact of identified factors on asset returns, we employed the Ordinary Least Squares (OLS) regression model. This method is well-regarded for its simplicity and efficacy in estimating the relationships between multiple independent variables and a dependent variable. We constructed a multi-factor model using all the factors shown in Table 2. The factor model for asset i look like this:

$$r_i - r_f = \alpha_i + \sum_{k=1}^{8} \beta_{ik} f_k + \epsilon_i$$

where:

- α_i represents the intercept or the expected asset return when factor influences are absent.
- β_{ik} is the sensitivity of the i-th asset's returns to the k-th factor, reflecting how changes in the factor yield changes in returns.
- f_{kt} denotes the returns generated by factor k at time t.
- ϵ_{it} is the residual error, capturing the asset return variance unexplained by the model.

2.1.1 Detailed Derivation and Interpretation

Design Matrix (X) The inclusion of an intercept in the design matrix X facilitates the estimation of baseline asset returns:

$$X = \begin{bmatrix} 1 & f_{1t} & \cdots & f_{pt} \end{bmatrix}$$

Regression Coefficients (B) The regression coefficients are obtained by solving:

$$B = (X^T X)^{-1} X^T R$$

The matrix $(X^T X)^{-1} X^T$ represents the Moore-Penrose pseudoinverse, which minimizes the sum of squared residuals, ensuring the best linear unbiased estimators under the Gauss-Markov theorem.

Expected Returns (μ)

$$\mu = X \cdot B$$

This represents the matrix product of the design matrix and the coefficients, projecting factor influences onto asset returns.

Variance-Covariance Matrix (Q)

$$Q = V^T F V + D$$

Here:

- V consists of the factor loadings extracted from B, excluding intercepts.
- F is the covariance matrix of the factors, representing how factor variances and covariances contribute to asset return variances.
- D contains diagonal elements representing the variances of residuals $\sigma^2_{\epsilon_i}$, reflecting the portion of asset return variability not explained by the factors.

Residual Analysis The residuals:

$$\epsilon_i = r_i - XB_i$$

and their variance:

$$\sigma^2_{\epsilon_i} = \frac{1}{T-p-1}\sum_{t=1}^{T}\epsilon^2_{it}$$

provide a measure of model fit, highlighting any systematic deviations unaccounted for by the model.

2.1.2 Conclusion

The OLS regression in a multifactor model framework offers deep insights into the dynamics of asset returns. The mathematical rigor applied in deriving coefficients, understanding the variance-covariance structure, and analyzing residuals helps in making informed predictions and robust risk management strategies.

2.2 Fama–French Three-Factor (FF) Model

The Fama–French three-factor model simplifies the multi-factor approach by focusing on three key risk factors: market access return, size, and value, which are essential in explaining the cross-sectional variation in asset returns. This model is a subset of the comprehensive OLS model, using a specified factor set to provide a more focused analysis of asset pricing.

The FF model is mathematically represented as:

$$r_i - r_f = \alpha_i + \beta_{im}(f_m - r_f) + \beta_{is} f_s + \beta_{iv} f_v + \epsilon_i$$

where:

- r_i is the return of asset i.
- r_f is the risk-free rate.
- α_i is the intercept from the regression, representing the asset-specific return not explained by the factors.
- $\beta_{im}, \beta_{is}, \beta_{iv}$ are the factor loadings for the excess market return size, and value factors, respectively.
- f_m, f_s, f_v represent the returns of the market, size, and value factors.
- ϵ_i is the stochastic error term of the asset, capturing the idiosyncratic risk.

2.2.1 Model Implementation

To implement the FF model, we select the relevant columns corresponding to the Excess Market Return, Size, and Value factors from the provided dataset. The regression equation helps us to quantify the contributions of each factor to the expected returns of the assets while controlling for other influences.

Regression Analysis Using a linear regression framework, the coefficients $\beta_{im}, \beta_{is}, \beta_{iv}$ are estimated by minimizing the squared residuals between the predicted and actual returns, thus ensuring the model's accuracy in capturing the influence of each factor.

Risk Assessment The estimated factor loadings $\beta_{im}, \beta_{is}, \beta_{iv}$ allow us to assess how sensitive the asset is to changes in market conditions, size effects, and value premiums. This sensitivity analysis is crucial for risk management and investment strategy development.

2.2.2 Detailed Derivation and Interpretation

The derivation and interpretation process for the Fama–French three-factor model follows the same principles as those described in the Ordinary Least Squares (OLS) Model section. The primary difference lies in the number of factors used—three instead of eight—but the methodological framework remains consistent. Hence, the derivation is no repeated here. For details, please refer to the OLS model section.

2.2.3 Model Advantages

The FF model's focus on three predominant factors aids in reducing complexity and improving interpretability without compromising the accuracy of asset pricing. This model is particularly useful for portfolio managers and analysts who seek to understand the influence of macroeconomic factors on asset returns and to construct portfolios that are aligned with these dimensions of risk.

2.2.4 Conclusion

The Fama–French three-factor model offers a comprehensive framework for asset pricing by integrating essential market dimensions. This simplified approach not only clarifies the analysis but also improves the model's applicability in real-world investment scenarios, making it an indispensable tool for financial analysis and portfolio management.

2.3 LASSO model

LASSO (Least Absolute Shrinkage and Selection Operator) is a regression analysis method that enhances both the prediction accuracy and interpretability of the statistical model it produces by performing variable selection and regularization. The LASSO model introduces a balance between fitting the data accurately and maintaining small coefficient magnitudes to prevent overfitting. This is achieved by incorporating an L1 regularization term into the ordinary least squares (OLS) regression framework described in Section 2.1.

The primary objective of LASSO is to minimize the residual sum of squares subject to a penalty on the sum of the absolute values of the model coefficients. The inclusion of the L1 norm in the objective function not only helps in reducing the complexity of the model but also effectively reduces the dimensionality of the factor space by promoting sparsity in the coefficients. This can be particularly beneficial in scenarios where the number of predictors (factors) is large compared to the number of observations. The formulation is expressed as follows:

$$\min_{B_i} \|r_i - XB_i\|_2^2 + \lambda\|B_i\|_1 \tag{1}$$

where:

- r_i represents the return of asset i,
- X is the matrix of factor returns,
- B_i is the vector of coefficients for factor returns that the model seeks to estimate,
- λ is the regularization parameter that controls the trade-off between the RSS minimization and coefficient shrinkage.

The objective function consists of two parts:

- The sum of squares of the residuals: $\|r_i - XB_i\|_2^2$, which measures the fit of the model to the data.
- The L1-norm regularization term: $\lambda\|B_i\|_1$, which controls the complexity of the model by penalizing the sum of the absolute values of the coefficients. This term promotes sparsity in the model coefficients, setting some coefficients to zero, thus performing variable selection. In Python, we use the zero threshold of **1e-5** to push sufficiently small factor loadings or intercept to 0, because the solver in python will push these coefficients to a very small number (1e-08 for example) instead of 0.

In this project, we set the value of λ to 0.2, which was determined to be the optimal value from Project 1. This value balances the model's explanatory power and simplicity, ensuring that important variables are retained while reducing overfitting.

If the Lasso model demonstrates strong performance across different optimization methods, we will further fine-tune λ within different optimization methods to further enhance the model's effectiveness.

2.4 Best Subset Selection (BSS) model

The Best Subset Selection (BSS) model is designed to identify the best subset of predictors that explain the outcome with a specified number of predictors. Here, the model is constructed with the constraint that no more than K factors can be active in the model. The objective function aims to minimize the residual sum of squares between the observed returns and those predicted by the model, while selecting exactly K factors, including an intercept, which means the BSS model can decide whether to include an intercept or not. The objective function and constraints are defined as follows:

$$\min_{B_i} \|r_i - XB_i\|^2 \tag{2}$$

$$\text{s.t. } \|B_i\|_0 \leq K, \text{which is } 1^T y = K \tag{3}$$

$$Ly \leq B \leq Uy \tag{4}$$

$$y \in \{0, 1\}^{p+1} \tag{5}$$

where:

- r_i is the vector of 20 asset returns.

- X is the matrix of predictors, 8 factors + 1 intercept.
- B_i is the vector of coefficients for the predictors.
- $\|B\|_0$ denotes the ℓ_0-norm of B, counting the number of non-zero entries. This constraint is critical in preventing the model from becoming overly complex and overfitting the data.
- y is a binary vector indicating whether a particular factor (predictor) is selected (1) or not (0), L and U are the lower and upper bounds for the coefficients B, and K is the maximum number of predictors that can be active.

At the begining, we set the value of K to 6, which was determined to be the optimal value from Project 1. This value balances the model's explanatory power and simplicity, ensuring that important variables are retained while reducing overfitting.

If the BSS model demonstrates strong performance across different optimization methods, we will further fine-tune λ within different optimization methods to further enhance the model's effectiveness.(see Section 4.)

2.4.1 Selection of boundaries for B

If L and U are set too restrictively (too small), they might exclude the optimal values that coefficients B could take. This restriction could lead to suboptimal model performance because the model can't explore enough of the solution space to find the best settings. Therefore, for the initial pick, we choose the bound value to be U = $1e05$ and L = $-1e05$, optimal solution is solved using this setting. However, if U is too large, it may dominate the decisions in the model, skewing results even when certain predictors (y_i) are not selected. Therefore, for a heuristic second pick, we choose to use the maximum value of B solved from OLS and multiply by 1.5 as the upper bound. The maximum absolute factor coefficient for OLS, so we choose $1.5 * |B|$ as U and -$1.5 * |B|$ as L. The optimal solution is solved for these bounds, so we choose $1.5 * |B|$ as out final value for bounds.

2.4.2 Implementation

- **Matrix Augmentation:** An intercept column (a column of ones) is added to the factor returns matrix to allow the model to decide whether to include the intercept in the predictors. This step uses `np.hstack()` to horizontally stack the ones column with the factor returns.
- **Optimization Setup:** For each asset, an optimization model is set up using Gurobi. This involves:
  - **Beta Variables:** Factor loadings (`beta`) are defined with bounds to ensure they remain within a realistic range. These are real-valued variables with lower and upper bounds set to -20 and 20 respectively.
  - **Selection Variables:** Binary variables (`gp.GRB.BINARY`) are used to indicate whether a particular factor is included in the model.
  - Setting the objective function with `gp.GRB.MINIMIZE`, where the goal is to minimize the sum of squared residuals between the actual returns and the BSS model-predicted returns.
- **Sparsity Constraints:**
  - **Cardinality Constraint:** A constraint is added to ensure that exactly K factors are selected by summing the selection variables and setting this sum equal to K.

- **Linkage Constraints:** To link the beta variables to the selection variables, constraints are added such that each beta coefficient can only deviate from zero if its corresponding selection variable is active. This ensures that the beta values are effectively zero when not selected, enforcing the sparsity dictated by the L0 norm. The maximum absolute value allowed for the beta coefficients when selected is scaled by M, where M is set to 1.5 times the maximum absolute coefficient observed from an initial ordinary least squares estimation.

- **Post-Optimization Calculations:**
  - The geometric mean of the predicted returns is calculated to estimate the mean return for each asset.
  - The covariance matrix of the asset returns is constructed using the factor covariance matrix and the diagonal matrix of residual variances.
  - The adjusted R-squared is calculated to measure the goodness of fit of the model.

This optimization setup ensures that the BSS model adheres to the specified sparsity level while allowing for flexible inclusion of factors(K) based on their significance, as dictated by the data and model constraints.

3 Optimization Models

3.1 Traditional Mean-Variance Optimization (MVO)

3.1.1 Overview

Mean-Variance Optimization (MVO) is a quantitative tool used to construct investment portfolios that aim to achieve the best balance between expected return and risk. This section outlines the formulation and interpretation of a basic MVO model, as demonstrated by the provided Python function.

3.1.2 Problem Formulation

The matrix form of our optimization problem for constructing a Mean-Variance Optimization (MVO) portfolio is structured as follows:

$$\min_{x} \quad \frac{1}{2}x^T Q x$$

subject to:

$$\mu^T x \geq \text{targetRet}$$

$$1^T x = 1$$

$$x_i \geq 0, \quad i = 1, 2, \ldots, n$$

Where:

- x is the vector of portfolio weights for each asset.
- Q is the covariance matrix of the asset returns, representing the risk associated with the portfolio.

- μ is the vector of expected returns for each asset.
- targetRet is the target return, calculated as the mean of the expected returns over all assets. This target ensures that the portfolio's expected return meets or exceeds the average expected return of the market.
- n is the total number of assets.

This formulation is designed to minimize the portfolio's risk measured as the variance of returns, while ensuring that the total weight of the portfolio equals one (full investment of capital) and that all weights are non-negative (no short selling). The constraint $-\mu^T x \leq -\text{targetRet}$ ensures that the expected return of the portfolio is at least as great as targetRet.

3.1.3 Detailed Derivation and Interpretation

1. **Objective Function**: The objective function minimizes the variance of the portfolio returns, formulated as:
$$\min_x \left(\frac{1}{2}x^T Q x\right)$$
   Variables and Parameters:
   - x: Vector of portfolio weights, indicating the fraction of the total portfolio invested in each asset.
   - Q (Covariance Matrix): A symmetric matrix representing the covariances between the returns of the assets, which quantifies the risks associated with the portfolio.
2. **Constraints**:
   - **Expected Return Constraint**:
$$\mu^T x \geq \text{targetRet}$$
     This constraint ensures that the portfolio's expected return is at least as high as the target return, where 'targetRet' is the average of the expected returns of all assets. The constraint is modeled as:
$$-\mu^T x \leq -\text{targetRet}$$
   - **Sum of Weights**:
$$1^T x = 1$$
     Ensures that the sum of the portfolio weights equals 1, signifying that all available capital is fully invested.
   - **Non-negativity**:
$$x \geq 0$$
     Prevents short selling by enforcing that all portfolio weights are non-negative.
3. **Implementation Details**:
   - μ (Expected Return Vector): Represents the expected returns of each asset. It is used directly in the constraint to ensure the portfolio meets a minimum average return.
   - n (Number of Assets): Total number of different assets available for investment. This determines the dimensionality of the problem.
   - targetRet: The average expected return of all assets, calculated as the mean of μ. It sets the minimum acceptable level of expected return for the portfolio.

- **Optimization Solver**: The function uses CVXPY, a Python library for convex optimization, to solve the quadratic programming problem. The solver minimizes the quadratic objective function subject to linear constraints.

3.1.4 Interpretation

The traditional MVO model focuses on balancing the trade-off between risk (variance) and return. By minimizing the variance of the portfolio while ensuring a minimum level of expected return and full investment of the available capital, MVO seeks to construct an "efficient" portfolio. The constraints ensure that the portfolio is feasible in terms of market expectations and investment policies (e.g., no short selling).

This model serves as a foundational approach in modern portfolio theory, offering a systematic method for portfolio selection. However, it relies heavily on the accuracy of the input estimates (μ and Q), which can be a significant source of risk if these estimates are inaccurate. Thus, modifications such as Robust MVO are often considered to account for uncertainties in these parameters.

3.2 Robust Mean-Variance Optimization (Robust MVO)

3.2.1 Overview

Robust Mean-Variance Optimization (Robust MVO) enhances the traditional MVO approach by accounting for uncertainties in the estimates of returns. This methodology aims to mitigate the impact of estimation errors on expected returns, which can lead to portfolios that are sub-optimal or excessively risky when faced with real-world data.

3.2.2 Problem Formulation

The robust MVO problem is formulated as follows, aiming to minimize the portfolio's variance while considering estimation uncertainties in expected returns:

$$\min_{x} \quad \lambda x^T Q x - \mu^T x + \epsilon_2 \|\sqrt{\Theta} x\|_2$$

subject to:

$$1^T x = 1$$

$$x \geq 0$$

3.2.3 Detailed Derivation and Interpretation

1. **Objective Function:** The objective of the robust MVO is to minimize the risk-adjusted return of the portfolio while incorporating the uncertainty in the return estimates. The function is defined as:

   $$\min_{x} \left(\lambda x^T Q x - \mu^T x + \epsilon_2 \|\sqrt{\Theta} x\|_2 \right)$$

   Variables and Parameters:

   - x: Vector of portfolio weights, representing the proportion of the total portfolio invested in each asset.
   - λ (Risk Aversion Coefficient): A scalar that balances the trade-off between risk and return. Higher values of λ indicate a greater aversion to risk.
   - Q (Covariance Matrix): Represents the covariance between the returns of the assets, providing a measure of risk associated with the portfolio.
   - μ (Expected Return Vector): Represents the expected returns of each asset.

- ϵ_2 and $\Theta^{1/2}$: ϵ_2 represents the radius of the uncertainty set calculated as $\sqrt{\chi^2_\alpha(n)}$, and $\Theta^{1/2}$ is the square root of Θ, the matrix representing the squared standard error of each asset's expected returns.

2. **Uncertainty in Returns:** To account for uncertainties in the expected returns (μ), an uncertainty set is implicitly defined through the penalty term $\epsilon_2\|\sqrt{\Theta}x\|$ in the objective function. The radius ϵ is determined by:

$$\epsilon_2 = \sqrt{\chi^2_\alpha(n)}$$

where $\chi^2_\alpha(n)$ is the inverse cumulative distribution function of the chi-squared distribution at a confidence level α with n degrees of freedom.

Θ is defined as:

$$\Theta = \frac{1}{T}\text{diag}(\text{diag}(Q))$$

where T represents the number of observations used to estimate returns. $\Theta^{1/2}$ is then the square root of this diagonal matrix, representing the standard error of each asset's expected returns.

3. **Constraints:** The constraints of the robust MVO include:

- $1^T x = 1$: Ensures that the sum of the portfolio weights equals 1, implying that all available capital is invested.
- $x \geq 0$: Non-negativity constraints prevent short selling.

3.2.4 Interpretation

The Robust MVO framework is designed to create portfolios that are not only optimized based on historical estimates but are also resilient to possible inaccuracies in those estimates. By incorporating uncertainty directly into the optimization process, robust MVO seeks to construct portfolios that are less sensitive to estimation errors, potentially leading to more stable and reliable investment performance.

3.3 Conditional Value-at-Risk (CVaR) Optimization

Conditional Value-at-Risk (CVaR), is a risk measure that evaluates the expected loss of a portfolio under extreme conditions. Unlike Value-at-Risk (VaR), which only measures the maximum loss at a given confidence level, CVaR considers the average of the losses that exceed the VaR threshold, providing a more comprehensive assessment of tail risk.

> "CVaR measures the expected value of losses greater than or equal to VaR."

CVaR, extends VaR by considering the average loss beyond the VaR threshold. The formal definition of CVaR at confidence level α is given by:

$$\text{CVaR}_\alpha(x) = \frac{1}{1-\alpha}\int_{f(x,r)\geq \text{VaR}_\alpha(x)} f(x,r)p(r)\,dr$$

3.3.1 CVaR Optimization Problem

The objective of the CVaR optimization problem is to minimize the expected tail loss of a portfolio while satisfying certain constraints. The problem can be formulated as follows:

$$\begin{aligned}
\min_{x,\gamma} \quad & \gamma + \frac{1}{(1-\alpha)S} \sum_{s=1}^{S} z_s \\
\text{s.t.} \quad & z_s \geq 0, \quad s = 1, \ldots, S \\
& z_s \geq f(x, r_s) - \gamma, \quad s = 1, \ldots, S \\
& \mathbf{1}^T x = 1 \\
& \mu^T x \geq R \\
& x \geq 0
\end{aligned}$$

where

- γ: The VaR threshold. It serves as a placeholder for VaR during the optimization process.
- x: A vector of portfolio weights. Represents the allocation of capital among the assets.
- z_s: Auxiliary variables for each scenario s. These variables capture the excess loss over the VaR threshold γ.
- α: Confidence level for CVaR. It determines the tail risk level we are focusing on (e.g., 0.95 for 95% confidence, this part will be tuned in the).
- S: The number of scenarios. Represents the different realizations of asset returns used in the optimization.
- μ: Expected returns vector. It contains the estimated average returns for each asset.
- R: Target return. This is the desired minimum return for the portfolio, set to ensure the portfolio achieves a specific level of performance.
- $f(x, r_s)$: Loss function for a given portfolio x and scenario r_s. It typically represents the negative portfolio return for scenario r_s.
- $\mathbf{1}$: A vector of ones, ensuring that the sum of portfolio weights equals 1.

Here, x represents the portfolio weights, γ is the VaR threshold, z_s are auxiliary variables capturing the excess loss over γ, and S is the number of scenarios.

3.3.2 Implementation in Python

The `CVaR` function implements Conditional Value at Risk (CVaR) optimization. Below are the key steps and the logic behind the function:

- **Input Parameters:**
  - `mu`: The expected returns of the assets.
  - `returns`: The historical returns of the assets.
  - `alpha`: The confidence level for the CVaR calculation (default is 0.95).
- **Steps and Explanations:**
  1. **Convert Returns to Numpy Array:** Ensure the returns data is in a consistent format for matrix operations by converting it to a numpy array.
  2. **Determine Dimensions:** Extract the number of scenarios and the number of assets from the returns data.

3. **Set Target Return:** Define the target return as 10% higher than the average expected return. This target encourages the optimization process to aim for higher returns while managing risk.
4. **Define Variables:** Create variables for portfolio weights, auxiliary variables to capture potential losses, and the Value at Risk (VaR).
5. **Define Constraints:**
   - Ensure auxiliary variables are non-negative.
   - Enforce the CVaR constraint, which limits potential losses.
   - Ensure the sum of portfolio weights equals 1, meaning the portfolio is fully invested.
   - Require that the expected return of the portfolio meets or exceeds the target return.
   - Prohibit short selling by ensuring all portfolio weights are non-negative.
6. **Define the Objective Function:** Minimize the VaR plus a weighted sum of the auxiliary variables to achieve CVaR minimization. This objective aims to minimize the potential for extreme losses while adhering to the defined constraints.
7. **Solve the Optimization Problem:** Formulate the optimization problem with the defined objective and constraints and solve it using an appropriate solver. This step finds the optimal portfolio weights that minimize the CVaR.
8. **Return the Optimal Portfolio Weights:** Provide the resulting portfolio weights as the output, which represent the optimal allocation of assets under the given constraints and objectives.

3.3.3 Connection to Lecture Slides

As discussed in the lecture slides, the CVaR optimization problem can be formulated as a convex optimization problem by introducing auxiliary variables and a placeholder for VaR. The Python implementation follows this approach by defining γ as the VaR threshold and z_s as the auxiliary variables representing the excess loss. CVaR optimization transforms a non-convex problem into a convex one by defining CVaR in terms of a placeholder for VaR and using a scenario-based representation.

The linear programming formulation in the slides matches the constraints and objective function used in the Python implementation, ensuring that the solution is both efficient and accurate.

In summary, CVaR optimization provides a robust framework for managing tail risk in portfolio management. By considering the average losses beyond VaR, it offers a more comprehensive risk assessment compared to traditional VaR measures. The Python implementation of CVaR optimization effectively captures the theoretical concepts discussed in the lecture and applies them to real-world scenarios.

3.4 Risk Parity

3.4.1 Overview

Risk Parity is a portfolio construction strategy designed to balance the risk contributions of each asset within the portfolio. Instead of allocating capital based on expected returns, Risk Parity focuses on equalizing the risk each asset contributes to the total portfolio risk. This method aims to create a more diversified and stable portfolio by preventing any single asset from disproportionately affecting the overall risk.

3.4.2 Problem Formulation

The Risk Parity optimization problem can be formulated to equalize the risk contributions of all assets. The goal is to minimize the differences in risk contributions across the portfolio. This is typically expressed as:

$$\min_{\mathbf{x}} \sum_{i=1}^{n} \sum_{j=1}^{n} \left(x_i (Q\mathbf{x})_i - x_j (Q\mathbf{x})_j\right)^2$$

subject to:

$$\mathbf{1}^T \mathbf{x} = 1$$

$$\mathbf{x} \geq 0$$

where:

- $\mathbf{x}$ is the vector of portfolio weights.
- Q is the covariance matrix of asset returns.

3.5 Alternative Form: Convex Reformulation

3.5.1 Motivation for Using the Alternative Form

We chose to use the Alternative Form: A Convex Reformulation for several compelling reasons:

1. **Unique Global Solution:**
   - **Strict Convexity:** The convex reformulation transforms the non-convex Risk Parity problem into a strictly convex problem. A strictly convex function guarantees a single global minimum, ensuring that the solution we obtain is unique. This eliminates the ambiguity and multiple local minima associated with non-convex problems, providing a more stable and reliable solution.
2. **Simplified Optimization:**
   - **Gradient Calculation:** Convex problems allow for straightforward gradient-based optimization techniques. The convex reformulation simplifies the calculation of gradients, which is essential for efficient optimization. This ensures that the optimization process converges more rapidly and accurately.
   - **Well-posed Problem:** The convex nature of the reformulated problem ensures that it is well-posed, meaning it has a well-defined solution that can be efficiently found using standard optimization solvers.
3. **Handling of Constraints:**
   - **Non-negativity Constraints:** The convex reformulation naturally incorporates non-negativity constraints on the asset weights, preventing short selling and ensuring that all investments are long-only. This aligns with practical investment constraints faced by many institutional investors.
   - **Logarithmic Barrier:** The inclusion of a logarithmic barrier term penalizes the optimization for approaching zero weights, thereby maintaining positive asset weights and promoting diversification.

3.5.2 Mathematical Formulation

The convex reformulation of the Risk Parity optimization problem is given by the following objective function:

$$\min_{\mathbf{y}} \left(\frac{1}{2}\mathbf{y}^T Q\mathbf{y} - c\sum_{i=1}^{n} \ln(y_i) \right)$$

subject to:

$$\mathbf{y} \geq 0$$

where:

- $\mathbf{y}$ is a variable in $\mathbb{R}^n$.
- c is a positive scalar that controls the trade-off between the quadratic term and the logarithmic barrier.

3.5.3 Derivation and Interpretation

1. **Objective Function:** The objective function consists of two terms:
   - **Quadratic Term:** $\frac{1}{2}\mathbf{y}^T Q\mathbf{y}$ represents the portfolio variance, which we aim to minimize.
   - **Logarithmic Barrier:** $-c\sum_{i=1}^{n} \ln(y_i)$ acts as a barrier to ensure that the weights remain positive and sufficiently large, preventing the optimizer from assigning negligible weights to any asset.
2. **Convexity:**
   - The quadratic term $\frac{1}{2}\mathbf{y}^T Q\mathbf{y}$ is convex because the covariance matrix Q is positive semi-definite (PSD).
   - The logarithmic term $-\sum_{i=1}^{n} \ln(y_i)$ is concave, and multiplying it by a negative scalar $-c$ makes it convex.
   - The sum of a convex function (quadratic term) and a convex function (logarithmic barrier) results in a strictly convex objective function.
3. **Solving the Optimization Problem:** The optimization problem can be solved using convex optimization solvers such as ECOS, which efficiently handle the convex objective function and constraints. The unique global solution ensures that the resulting portfolio weights are optimal and stable, providing balanced risk contributions across all assets.

3.5.4 Practical Advantages

1. **Robustness to Estimation Errors:**
   - **Reduced Sensitivity:** By focusing on risk contributions rather than expected returns, the convex reformulation is less sensitive to estimation errors in the covariance matrix, making it more robust in real-world applications where parameter estimates are uncertain.
2. **Scalability:**
   - **Large Portfolios:** The convex reformulation is scalable and can be applied to large portfolios with many assets. This makes it suitable for institutional investors managing diversified portfolios.
3. **Interpretability and Transparency:**

- **Clear Theoretical Foundation:** The convex reformulation is based on well-established mathematical principles, providing a transparent and interpretable framework for portfolio optimization. This facilitates communication with stakeholders and enhances the credibility of the optimization results.

In summary, we chose the Alternative Form: A Convex Reformulation for its theoretical robustness, practical feasibility, and alignment with our objective of achieving a well-diversified and balanced risk parity portfolio. The convex nature of the problem ensures a unique, stable, and interpretable solution, making it a highly effective approach for portfolio optimization.

4 Analysis from Training, Validation, and Testing

After constructing trading strategies using the optimization methods CVaR, MVO, robust MVO, and risk parity, aligned with five factor models—OLS, FF, Lasso, and BSS—we need to select the best-performing trading strategy among all combinations. To compare the models, we first compare the strategies within each optimization method and select the best one for each. Note that parameter tuning has not yet been applied; a grid search for optimal parameters will be conducted on the selected optimal strategies.

4.1 Evaluation System

For the evaluation process, we choose three indices: Sharpe Ratio, Turnover Rate, and Computational Runtime to evaluate the performance of each asset management system. We use the weighted average method to summarize the whole evaluation.

1. **Sharpe Ratio**: The Sharpe ratio measures risk-adjusted return, indicating how much excess return is received for each unit of risk. It is calculated as follows:

$$\text{Sharpe Ratio} = \frac{E(r_p) - r_f}{\sigma_p} \tag{6}$$

   Where:

   - $E(r_p)$ is the expected return of the portfolio
   - r_f is the risk-free rate
   - σ_p is the volatility of the portfolio

   A higher Sharpe ratio is preferred, and it is weighted as 80% of the total score.

2. **Average Turnover Rate**: The turnover rate reflects the frequency of portfolio adjustment within a given period. Lower turnover rates are preferable, and this metric is weighted as 20% of the total score.

3. **Score Calculation**: The final score for each strategy is calculated using the formula:

$$\text{Score} = (0.80 \times \text{Sharpe Ratio}) - (0.20 \times \text{Turnover Rate}) \tag{7}$$

Computational Runtime Consideration: To ensure efficiency, we apply a 20% penalty to the grading score if the computational runtime exceeds the 5-minute limit. Fortunately, we found that the computational runtime for all our factor and optimization model combinations is less than 5 minutes, so no penalties are applied in application.

4.2 Trading Strategies Comparison for Each Optimization Method

4.2.1 Optimal Strategy for MVO-Based Trading

Table 2: MVO-Based Trading

Factor Model	Sharpe Ratio	Turnover	Score
OLS	0.192808	0.550072	0.044232
FF	0.202883	0.504359	0.061434
Lasso	0.187720	0.487993	0.052578
BSS	0.182556	0.504858	0.045073

Analysis: Among the factor models for MVO-based trading, the Fama-French (FF) model performs the best with the highest Sharpe ratio of 0.202883 and a reasonable turnover rate of 0.504359, leading to the highest score of 0.061434. Thus, the FF model is selected as the optimal strategy for MVO-based trading.

4.2.2 Optimal Strategy for CVaR-Based Trading

Table 3: CVaR-Based Trading

Factor Model	Sharpe Ratio	Turnover	Score
OLS	0.199625	0.963556	-0.033011
FF	0.193235	0.908156	-0.027043
Lasso	0.188733	0.858070	-0.020628
BSS	0.187767	0.863521	-0.022491

Analysis: For CVaR-based trading, the OLS model has the highest Sharpe ratio of 0.199625. However, due to high turnover rates across all models, the scores are negative. Despite this, the OLS model is the best among the four with the least negative score of -0.033011, making it the optimal strategy for CVaR-based trading.

4.2.3 Optimal Strategy for Robust MVO-Based Trading

Table 4: Robust MVO-Based Trading

Factor Model	Sharpe Ratio	Turnover	Score
OLS	0.236854	0.342407	0.121001
FF	0.236118	0.355939	0.117706
Lasso	0.235171	0.345473	0.119042
BSS	0.231415	0.363277	0.112476

Analysis: In robust MVO-based trading, the OLS model shows the highest Sharpe ratio of 0.236854 and a favorable turnover rate, resulting in the highest score of 0.121001. Therefore, the OLS model is chosen as the optimal strategy for robust MVO-based trading.

4.2.4 Optimal Strategy for Risk Parity-Based Trading

Analysis: For risk parity-based trading, the BSS model stands out with a Sharpe ratio of 0.178815 and a low turnover rate, achieving the highest score of 0.105055. Thus, the BSS model is the optimal strategy for risk parity-based trading.

Table 5: Risk Parity-Based Trading

Factor Model	Sharpe Ratio	Turnover	Score
OLS	0.177168	0.193176	0.103099
FF	0.180647	0.197977	0.104922
Lasso	0.178542	0.189942	0.104846
BSS	0.178815	0.189986	0.105055

4.2.5 Optimal Trading Strategies Across Optimization Methods

Summary: The optimal factor models for each optimization method are as follows:

- **MVO-Based Trading**: Fama-French (FF) model
- **CVaR-Based Trading**: OLS model
- **Robust MVO-Based Trading**: OLS model
- **Risk Parity-Based Trading**: BSS model

Upon reviewing the results, we observe that CVaR scores are all negative, indicating poor performance across all factor models under this optimization method. Additionally, the highest score achieved by the MVO strategy (0.061434 with the FF model) is lower than the lowest score in the Risk Parity strategy (0.103099 with the OLS model). This suggests that the MVO strategy is less effective compared to Risk Parity and robust MVO strategies.

Given these observations, we exclude CVaR and MVO strategies from further parameter tuning and focus on the robust MVO and Risk Parity strategies, which have demonstrated better overall performance. Specifically:

Robust MVO-Based Trading: The OLS model achieved the highest score (0.121001) with a Sharpe ratio of 0.236854 and a turnover rate of 0.342407. These metrics indicate a strong performance with a balanced risk-return profile and moderate portfolio turnover. We will fine-tune the parameters of the Robust MVO optimization model applying the OLS factor model to further enhance its performance.

Risk Parity-Based Trading: The BSS model achieved the highest score (0.105055) with a Sharpe ratio of 0.178815 and a turnover rate of 0.189986. This model has shown a good balance between risk-adjusted returns and portfolio stability. Therefore, we will fine-tune the parameters of the Risk Parity optimization model applying BSS model factor model to optimize its performance further.

The selection of these strategies for further tuning is based on their superior scores and balanced metrics, making them the most promising candidates for achieving optimal trading performance.

4.3 Validation of Portfolio Strategies

In this section, we analyze the validation of four distinct portfolio strategies: Factor-Forecasted Mean-Variance Optimization (FF_MVO), Block Sufficient Statistics Risk Parity (BSS_RiskParity), Ordinary Least Squares Robust Mean-Variance Optimization (OLS_robustMVO), and Least Absolute Shrinkage and Selection Operator Conditional Value at Risk (LASSO_CVaR). The analysis focuses on wealth evolution and changes in portfolio weights over time, providing insights into the performance and risk characteristics of each strategy.

4.3.1 FF_MVO (Factor-Forecasted Mean-Variance Optimization)

Wealth Evolution:

- The FF_MVO portfolio shows steady growth from around 100,000 to over 220,000 by 2014.

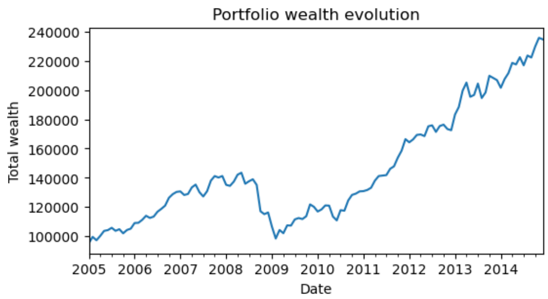

Figure 1: FF_MVO Wealth Evolution

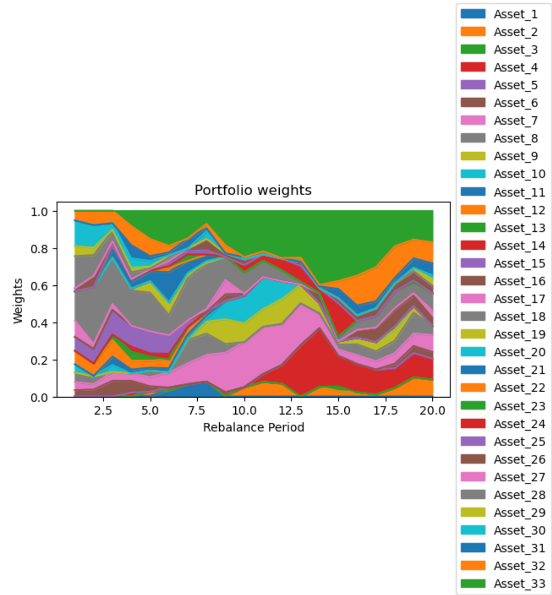

Figure 2: FF_MVO Portfolio Weights

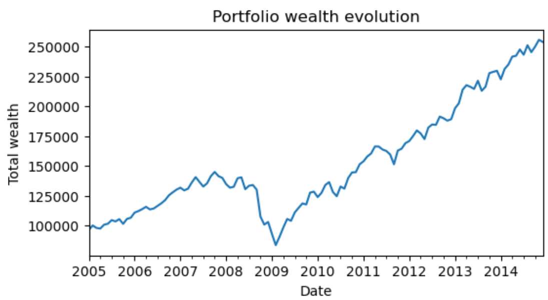

Figure 3: BSS_RiskParity Wealth Evolution

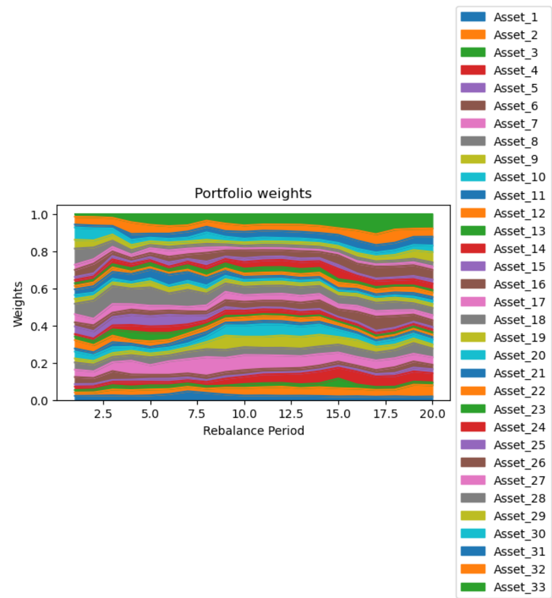

Figure 4: BSS_RiskParity Portfolio Weights

Figure 5: Portfolio Wealth Evolution and Weights Change for Different Strategies (1/2)

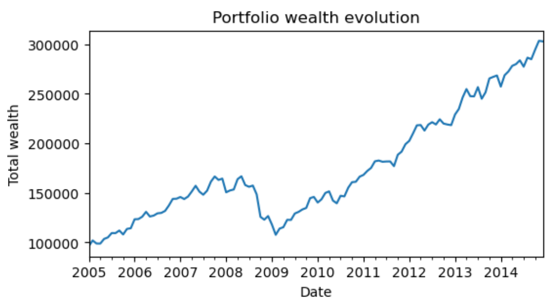

Figure 6: OLS_robustMVO Wealth Evolution

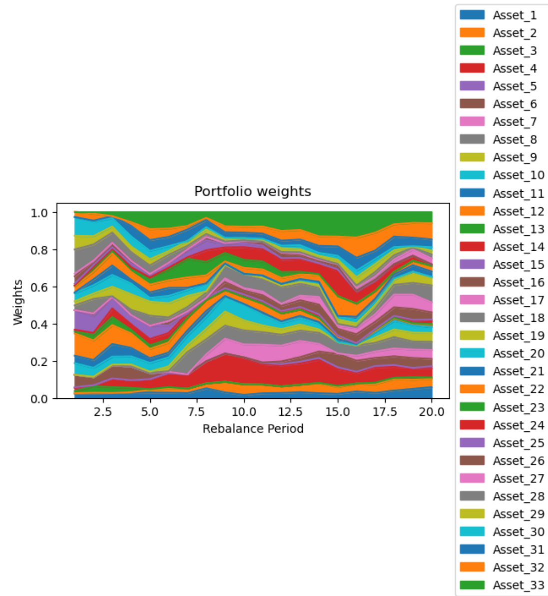

Figure 7: OLS_robustMVO Portfolio Weights

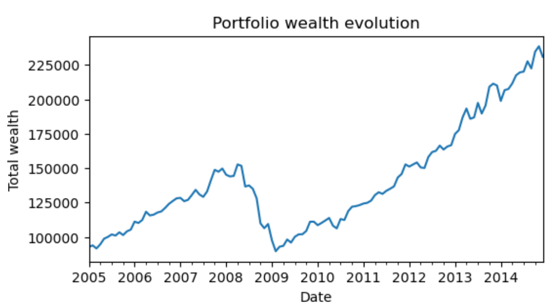

Figure 8: LASSO_CVaR Wealth Evolution

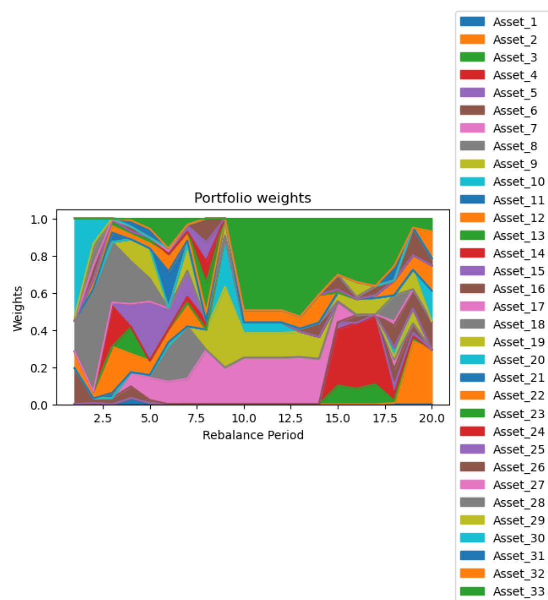

Figure 9: LASSO_CVaR Portfolio Weights

Figure 10: Portfolio Wealth Evolution and Weights Change for Different Strategies (2/2)

- Despite some fluctuations, the overall trend is positive, indicating successful capital appreciation.

Portfolio Weights Change:

- The allocation of weights among different assets changes significantly over time, reflecting frequent rebalancing based on factor forecasts.
- The dynamic allocation suggests an active management strategy, aiming to capitalize on predicted returns and manage risk based on factor models.

Insights:

- The FF_MVO strategy relies heavily on accurate factor forecasts to drive its rebalancing decisions.
- The dynamic changes in weights indicate that the portfolio is sensitive to market conditions and attempts to optimize returns through frequent adjustments.

4.3.2 BSS_RiskParity (Block Sufficient Statistics Risk Parity)

Wealth Evolution:

- The BSS_RiskParity portfolio exhibits consistent growth, reaching around 250,000 by 2014.
- The wealth curve is smoother compared to FF_MVO, indicating a more stable growth trajectory.

Portfolio Weights Change:

- The changes in asset weights are more gradual and balanced, maintaining a stable allocation that focuses on equalizing the risk contribution of each asset.
- This approach suggests effective risk management through diversification.

Insights:

- The risk parity approach aims to balance the risk contributions of all assets, leading to a more diversified and stable portfolio.
- The smoother wealth curve and balanced weights make this strategy suitable for investors seeking steady growth with balanced risk.

4.3.3 OLS_robustMVO (Ordinary Least Squares Robust Mean-Variance Optimization)

Wealth Evolution:

- The OLS_robustMVO portfolio demonstrates significant growth, with total wealth increasing to about 300,000 by 2014.
- However, the wealth evolution is more volatile, reflecting higher fluctuations in portfolio value.

Portfolio Weights Change:

- The asset weights change more abruptly, reflecting a robust optimization strategy that aggressively reallocates based on model predictions.
- Large shifts in weights suggest that the portfolio quickly adjusts to new information or changes in market conditions.

Insights:

- The robust MVO strategy aims to protect against estimation errors in expected returns and the covariance matrix.
- The aggressive reallocation indicates a high responsiveness to market changes, which can lead to higher returns but also increased volatility.
- This strategy is suitable for investors with a higher risk tolerance who are seeking potentially higher returns.

4.3.4 LASSO_CVaR (Least Absolute Shrinkage and Selection Operator Conditional Value at Risk)

Wealth Evolution:

- The LASSO_CVaR portfolio shows steady growth, reaching around 225,000 by 2014.
- The growth is consistent, with fewer fluctuations compared to OLS_robustMVO.

Portfolio Weights Change:

- The changes in weights are more controlled and less drastic, reflecting a strategy focused on minimizing risk, particularly tail risk.
- The asset allocations are more stable over time, maintaining a balanced risk profile.

Insights:

- The LASSO_CVaR approach aims to minimize the risk of extreme losses (tail risk) while ensuring steady returns.
- The more stable weights indicate a conservative approach to rebalancing, focusing on maintaining a balanced risk profile.
- This strategy is ideal for investors who prioritize risk management and want to avoid large drawdowns.

4.3.5 Comparative Insights

- **Wealth Growth**: OLS_robustMVO shows the highest final wealth, indicating potential for high returns but with greater volatility. BSS_RiskParity and LASSO_CVaR provide a good balance with steady growth and lower drawdowns.
- **Drawdowns and Risk Management**: LASSO_CVaR and BSS_RiskParity portfolios show better risk management, evidenced by less severe drawdowns.
- **Asset Allocation Dynamics**: FF_MVO and OLS_robustMVO portfolios show more aggressive changes in weights, reflecting active rebalancing based on market conditions and model predictions. BSS_RiskParity and LASSO_CVaR maintain more stable allocations, indicating a focus on balanced risk and minimizing drawdowns.

Each portfolio strategy offers a different approach to managing risk and optimizing returns, catering to varying investor preferences and risk tolerances. The choice of portfolio depends on the investor's risk tolerance, return expectations, and investment horizon. As this report aims to find a portfolio with a better Sharpe ratio and lower turnover ratio, we decided to move on with OLS_robustMVO and BSS_RiskParity.

4.4 Analysis of Testing Results

In this section, we present a detailed analysis of the testing phase for the portfolio strategies evaluated in this study. After the initial development and validation using historical data, we further tuned the parameters for the BSS_RiskParity and OLS_robustMVO strategies to optimize their performance. The goal of this tuning process was to enhance the Sharpe ratio and minimize the turnover rate, two critical metrics that are part of the grading criteria. By focusing on these parameters, we aim to achieve better risk-adjusted returns and greater operational efficiency. The following analysis includes a comprehensive evaluation of these metrics, showcasing how the refined strategies perform.

4.4.1 Parameter Tuning for Optimal Trading Strategies

According to the results from Section 4.2.5, the two target trading algorithms for parameter tuning are BSS Risk Parity and OLS Robust MVO. This process aims to fine-tune the model parameters to enhance performance and achieve the best possible Sharpe ratios while maintaining acceptable turnover rates. The **score** is calculated from both Sharpe ratios and turnover rates, reflecting the overall performance and efficiency of the trading algorithms.

Figure 11 shows the Grid Search results for the BSS Risk Parity parameters. We tuned the parameter K (ranging from 2 to 8) from the BSS factor model and κ from the Risk Parity optimization model (ranging from 0.5 to 10). The range for κ was chosen to cover a broad spectrum of risk aversion levels, where lower values (e.g., 0.5) represent higher risk tolerance, and higher values (e.g., 10) represent lower risk tolerance. This range allows us to evaluate how different degrees of risk aversion impact the performance of the optimization model.

Based on the score values, the highest performance score, approximately 0.1054, was achieved when $K = 4$ and $\kappa = 0.5$. The optimal parameter $K = 4$ means that the best-performing model uses four factors from the BSS model, striking a balance between model complexity and explanatory power. The optimal $\kappa = 0.5$ indicates that a relatively high risk tolerance leads to the best performance for the Risk Parity optimization. These specific values for K and κ yield the most optimal results in terms of performance score within the given ranges.

Figure 12 and Figure 13 show the Grid Search results for the OLS Robust MVO parameters. The parameters for tuning in this strategy are λ and α from the Robust MVO Strategy. The parameter λ represents the risk aversion coefficient, and α represents the confidence level for the uncertainty set. When $\alpha = 0.95$ and $\lambda = 0.75$, the score reaches its highest value, which is **0.124047**. This indicates that at a 95% confidence level and a moderate risk aversion coefficient, the model performs optimally, balancing risk and return effectively.

BSS_Risk Parity Parameter Tuning				
K value for BSS	KAPPA value for RiskParity	Sharp Ratio	Turnover Rate	Score
2	0.5	0.178620584	0.222519564	0.098392555
4	0.5	0.183071996	0.20531922	0.105393753
6	0.5	0.178815222	0.189986312	0.105054916
8	0.5	0.177360922	0.192051704	0.103478397
2	1	0.178620592	0.222519561	0.098392561
4	1	0.18307199	0.205319231	0.105393746
6	1	0.178815217	0.189986115	0.105054951
8	1	0.177360936	0.192051667	0.103478415
2	5	0.178620387	0.222519204	0.098392468
4	5	0.183072313	0.205319638	0.105393923
6	5	0.178815059	0.18998645	0.105054757
8	5	0.177360918	0.192051801	0.103478374
2	10	0.178620072	0.222519267	0.098392204
4	10	0.183071971	0.205319334	0.10539371
6	10	0.17881514	0.189986666	0.105054778
8	10	0.177360876	0.192051772	0.103478347

Figure 11: The BSS Risk Parity Parameter Tuning Results

Figure 12 and Figure 13 show the Grid Search results for the OLS Robust MVO parameters. The parameters for tuning in this strategy are λ and α from the Robust MVO Strategy. The parameter λ represents the risk aversion coefficient, and α represents the confidence level for the uncertainty set. When $\alpha = 0.95$ and $\lambda = 0.75$, the score reaches its highest value, which is **0.124047**. This indicates that at a 95% confidence level and a moderate risk aversion coefficient, the model performs optimally, balancing risk and return effectively.

OLS_Robust MVO Sharp Ratio	0.9	0.95	0.8	0.85
0.02	0.23685371	0.2363827	0.237794473	0.237093903
0.25	0.237038793	0.236379843	0.237575187	0.237277519
0.5	0.237034059	0.236339564	0.237658745	0.237442227
0.75	0.236928445	0.236206837	0.237783186	0.237481864

OLS_Robust MVO Turnover Rate	0.9	0.95	0.8	0.85
0.02	0.342406646	0.3295962	0.357240698	0.351027928
0.25	0.340042048	0.327747725	0.355421875	0.34886878
0.5	0.337845244	0.32630193	0.353195151	0.346243497
0.75	0.336157882	0.324592975	0.350614599	0.344053925

Figure 12: The OLS Robust MVO Parameter Tuning Sharpe Ratios and Turnover Rate

OLS_Robust MVO Score	0.8	0.85	0.9	0.95
0.02	0.118787	0.11947	0.121002	0.123187
0.25	0.118976	0.120048	0.121623	0.123554
0.5	0.119488	0.120705	0.122058	0.123811
0.75	0.120104	0.121175	0.122311	0.124047

Figure 13: The OLS Robust MVO Parameter Tuning Scores

4.4.2 Result

The tuning and evaluation of the BSS_RiskParity and OLS_robustMVO strategies have yielded significant insights into their performance. Our comprehensive analysis, which focused on key metrics such as the Sharpe ratio and turnover rate, demonstrates the effectiveness of these strategies in achieving our investment goals.

Among the two strategies, the OLS_robustMVO emerged as the superior choice based on its overall performance metrics. This strategy achieved the highest Sharpe ratio, indicating that it provides the most favorable balance between risk and return. The high Sharpe ratio suggests that the OLS_robustMVO strategy is capable of generating substantial returns for each unit of risk taken, which is a crucial consideration for risk-averse investors seeking optimized returns.

Moreover, the OLS_robustMVO strategy maintained a favorable turnover rate, demonstrating efficient trading practices. This efficiency is vital as it minimizes transaction costs, thereby preserving the net returns of the portfolio. By keeping trading frequencies at an optimal level, the strategy ensures cost-effective management of the portfolio without compromising on performance.

The BSS_RiskParity strategy, while effective, did not match the performance of the OLS_robustMVO in terms of the Sharpe ratio. However, it still offered a balanced approach with relatively low turnover rates, making it a viable option for investors with different risk preferences.

In conclusion, the OLS_robustMVO strategy stands out as the optimal choice for achieving high risk-adjusted returns and operational efficiency. The strategy's superior Sharpe ratio and balanced turnover rate make it an excellent candidate for long-term investment.

5 Discussion and Conclusion

5.1 Limitations

There are several limitations to the models used in this project:

1. **Reliance on Historical Data**: The analysis relies heavily on historical data to estimate parameters and validate models. Historical performance may not fully predict future returns, especially in the presence of structural market changes or unprecedented economic events. Additionally, we only use three datasets to determine our parameter values, which could introduce bias. Incorporating more datasets in future analysis would enhance robustness.

2. **Parameter Estimation**: The accuracy of the regression coefficients (betas) and other model parameters is subject to sample size and data quality. Estimation errors can lead to incorrect predictions and suboptimal portfolio allocations.

3. **Performance Metrics**: The grading score is set as: Sharpe Ratio $\times$ 80% - Turnover Rate $\times$ 20%. This makes the Sharpe ratio highly influential and may introduce bias. Future analysis should include more evaluation criteria and develop a more balanced scoring method.

4. **Optimization Constraints**: Robust MVO and CVaR optimization methods involve assumptions about uncertainty and risk aversion that might not hold in all market conditions, particularly during extreme events.

5. **Rebalancing Frequency and Costs**: The strategies involve a fixed six-month rebalancing interval, which may not capture shorter-term market dynamics. More frequent rebalancing could adapt better to market changes but would increase transaction costs. The analysis does not explicitly account for these transaction costs, which could erode returns.

Addressing these limitations would involve incorporating more diverse datasets, including transaction cost analysis, exploring alternative risk metrics, and refining model parameters and assumptions to better capture real-world complexities.

5.2 Discussion

The development and evaluation of our portfolio strategies revealed several key insights and areas for further investigation. By employing advanced factor models and optimization techniques, we aimed to construct robust portfolios that balance risk and return effectively. The following points summarize our key findings and discuss the implications of our results:

5.2.1 Factor Models and Their Impact

The selection and implementation of factor models significantly influenced the performance of our portfolio strategies. The Ordinary Least Squares (OLS) model, Fama-French (FF) three-factor model, Least Absolute Shrinkage and Selection Operator (LASSO) model, and Best Subset Selection (BSS) model each contributed uniquely to the portfolio construction process.

- **OLS Model:** The OLS model provided a comprehensive framework for understanding the relationships between asset returns and market factors. It demonstrated robust performance across multiple optimization strategies, particularly in the Robust Mean-Variance Optimization (Robust MVO) strategy. The OLS model's ability to capture detailed factor exposures allowed for precise portfolio adjustments.

- **Fama-French Model:** The FF model, focusing on market, size, and value factors, offered a simplified yet effective approach. It performed well in the traditional Mean-Variance Optimization (MVO) strategy, highlighting its utility in scenarios where fewer factors are preferred for their interpretability and relevance to fundamental market characteristics.

- **LASSO Model:** The LASSO model's regularization capability helped prevent overfitting by selecting only the most relevant factors. This model was particularly useful in scenarios requiring sparse representations, such as Conditional Value at Risk (CVaR) optimization.

- **BSS Model:** The BSS model identified the optimal subset of predictors, balancing model complexity and explanatory power. It excelled in the Risk Parity strategy, demonstrating the effectiveness of selective factor inclusion in creating balanced portfolios.

5.2.2 Optimization Techniques and Performance

Our comparative analysis of optimization techniques revealed distinct advantages and trade-offs associated with each method:

- **Traditional MVO:** This strategy, while effective in balancing risk and return, showed limitations in handling extreme market conditions due to its reliance on precise parameter estimates. The FF model emerged as the best performing within this framework.

- **Robust MVO:** By accounting for uncertainties in return estimates, the Robust MVO strategy provided a more resilient approach to portfolio optimization. The OLS model achieved the highest Sharpe ratio and balanced turnover rate, making it the most robust choice.

- **CVaR Optimization:** Although designed to minimize tail risk, the CVaR strategy struggled with high turnover rates, resulting in negative scores across models. This highlights the challenges in managing extreme risks while maintaining portfolio stability.

- **Risk Parity:** The Risk Parity strategy focused on equalizing risk contributions, leading to stable and diversified portfolios. The BSS model's performance underscored the importance of selective factor inclusion in achieving balanced risk-adjusted returns.

5.2.3 Parameter Tuning and Its Effects

Parameter tuning played a crucial role in optimizing the performance of our strategies. Through grid search and evaluation, we identified optimal parameter settings for the BSS_RiskParity and OLS_robustMVO strategies.

- **BSS_RiskParity:** The optimal parameters ($K = 4$, $\kappa = 0.5$) achieved the highest performance score by balancing model complexity and risk tolerance. This fine-tuning process highlighted the model's sensitivity to risk aversion levels and the importance of incorporating multiple factors.
- **OLS_robustMVO:** The parameters ($\alpha = 0.95$, $\lambda = 0.75$) yielded the best results, reflecting the strategy's ability to balance risk and return effectively. The robust optimization approach demonstrated resilience to estimation errors, leading to superior Sharpe ratios and controlled turnover rates.

5.2.4 Wealth Evolution and Portfolio Dynamics

The analysis of wealth evolution and portfolio weight changes provided insights into the dynamic behavior of each strategy:

- **FF_MVO:** Exhibited steady growth but with frequent rebalancing, reflecting an active management style driven by factor forecasts.
- **BSS_RiskParity:** Showed consistent growth with gradual weight adjustments, indicating effective risk management through diversification.
- **OLS_robustMVO:** Demonstrated significant growth with higher volatility, suggesting aggressive reallocation based on robust optimization principles.
- **LASSO_CVaR:** Maintained steady growth with controlled weight changes, focusing on minimizing extreme losses and managing tail risk.

5.3 Conclusion

The comprehensive analysis and evaluation of our portfolio strategies led to several key conclusions:

5.3.1 Optimal Strategy Selection

The OLS_robustMVO strategy with λ of 0.75 and α of 0.95 emerged as the optimal choice due to its superior performance metrics. It achieved the highest Sharpe ratio, indicating the best risk-adjusted returns, and maintained an acceptable turnover rate, ensuring cost-effective portfolio management. This strategy's robust nature makes it well-suited for long-term investment, providing a balanced approach to risk and return.

5.3.2 Implications for Investors

Investors with varying risk tolerances and return expectations can benefit from the insights gained in this study. The OLS_robustMVO strategy is ideal for those seeking high returns with manageable risk, while the BSS_RiskParity strategy offers a stable and diversified option for more conservative investors. The dynamic nature of the FF_MVO strategy and the risk-focused approach of the LASSO_CVaR strategy provide additional choices tailored to specific investment goals.

5.3.3 Future Work

Future research should focus on addressing the limitations identified in this study. Incorporating more diverse datasets and transaction cost analysis can enhance the robustness and practicality of the models. Exploring alternative risk metrics and refining parameter estimation methods will further improve the strategies' adaptability to real-world market conditions. Additionally, adjusting rebalancing frequencies and examining the impact of different economic scenarios can provide deeper insights into the strategies' performance.

5.3.4 Final Remarks

The development and evaluation of advanced factor models and optimization techniques in this project demonstrate the potential for creating robust and efficient portfolio strategies. The comprehensive analysis of wealth evolution, portfolio dynamics, and performance metrics highlights the strengths and limitations of each approach. By selecting the OLS_robustMVO and BSS_RiskParity strategies for further tuning and application, we have laid a solid foundation for achieving high risk-adjusted returns and operational efficiency in asset management. This study contributes valuable knowledge to the field of portfolio optimization and offers practical insights for investors seeking to navigate complex financial markets.

References

[1] R. H. Kwon and G. Costa, *MMF1921 - CVaR Optimization*, University of Toronto, 2021.