Replicating the Russell 1000 Index with Top 25 Constituents

Tiko Nikvashvili

1 Abstract

To replicate the Russell 1000 index, the project utilized the top 25 index constituents and implemented several methodologies, including Market Cap Weighting, which served as benchmark, simple Linear Regression, Ridge Regression, Tracking Error Minimization, Factor-Based Optimization, and Iteratively Reweighted L_1. The ridge approach achieved the highest replication accuracy with an annualized tracking error of 2.95% and R^2 of 0.96 during the test period, outperforming simple market cap weighting (5.01% tracking error and R^2 of 0.89).

2 Data Collection & Cleaning

Historical price data was sourced from Yahoo Finance using the `yfinance` Python package, covering the training period from 1/1/2023-3/31/2024 (15 months of data for training was chosen to balance between having sufficient data and capturing recent market conditions), and the testing period during April 2024. Fama-French daily factor data was sourced via `pandas_datareader`. Data was aligned to the NYSE trading calendar, and missing data points were forward-filled if found. Data quality checks were conducted, but no stale or negative prices were found. Outliers were identified using a z-score threshold ($|z| > 5$) and visually inspected as seen in Figure 1, after which they were not winsorized since they appeared to be genuine moves.

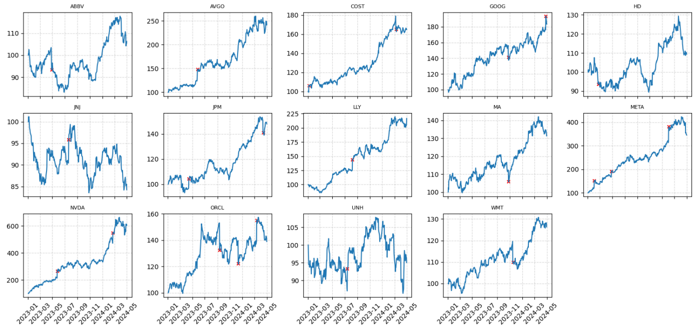

Figure 1: Time-series plots with detected outliers marked in red.

Daily simple returns were calculated as:

$$r_t = \frac{P_t}{P_{t-1}} - 1.$$

3 Methodology & Modeling Decisions

I implemented five weighting methodologies on top of the marketcap weighted benchmark described in more detail below, each tuned via 3-fold time series cross-validation on the training set to select hyperparameters minimizing average out-of-sample tracking error across folds. This approach ensures robustness to time-varying market conditions while preventing overfitting. The final models were trained on the entire training set using the optimal hyperparameters and the best weighting methodology was again chosen using the validation set, after which, the performance of the best weigthing methodology and the marketcap weighting benchmark was evaluated on the test period. I used tracking error as our evaluation metric:

Tracking Error (TE): Annualized volatility of active returns:

$$\text{TE} = \sqrt{\frac{1}{T}\sum_{t=1}^{T}\left(r_{p,t} - r_{\text{idx},t}\right)^2} \times \sqrt{252}.$$

but also report the r-squared as an additional metric. For each method, weights were constrained to be non-negative and sum to one.

3.1 Market-Cap Benchmark

The benchmark portfolio assigns weights proportional to each stock's market capitalization as of March 31, 2024:

$$w_i = \frac{\text{Market Cap}_i}{\sum_{j=1}^{25} \text{Market Cap}_j}$$

This approach serves as our baseline, representing the simplest implementation strategy.

3.2 Linear Regression

This method uses ordinary least-squares regression to estimate weights that best replicate the index returns:

$$\min_{w \in R^N} \sum_{t=1}^{T} \left(r_t^{\text{idx}} - w^\top r_t\right)^2, \quad \text{s.t.} \quad \sum_{i=1}^{N} w_i = 1, \quad w_i \geq 0 \quad \forall\, i.$$

Benefit of this approach is that it directly minimizes squared tracking error and is simple to implement and explain.

3.3 Ridge Regression

Ridge regression adds an L2 penalty term to the simple linear regression objective function, addressing potential multicollinearity among stock returns which I saw when looking at return correlations among the 25 constituents as seen in correlation matrix below. The regularization term also reduces estimation noise.

$$\min_{w \in R^N} \sum_{t=1}^{T} \left(r_t^{\text{idx}} - w^\top r_t\right)^2 + \lambda \left\|w\right\|_2^2$$

Subject to the same non-negativity and normalization constraints as linear regression. The regularization parameter α was optimized via cross-validation, the optimal value was $\alpha = 1.0e - 04$.

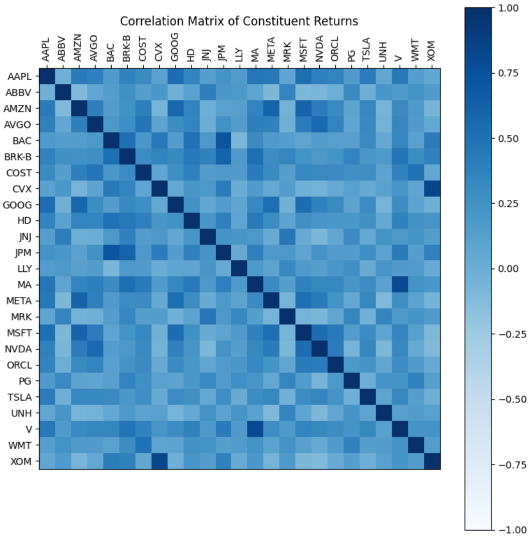

Figure 2: Correlation matrix of the 25 constituent returns over the training period.

3.4 Tracking Error Minimization

This approach directly minimizes the variance of tracking error:

$$\min_{w \in R^N} w^\top \Sigma\, w \; - \; 2\, w^\top \Sigma_{\text{idx}} \quad \text{s.t.} \quad \sum_{i=1}^{N} w_i = 1, \quad w_i \geq 0 \quad \forall\, i,$$

where Σ is the covariance matrix of component returns and Σ_{idx} the vector of covariances between each component and the index.

3.5 Factor-Based Optimization

This method attempts to match the Russell 1000's factor exposures while minimizing idiosyncratic risk:

$$\min_{w \in R^N} \left\| \mathbf{B}\, w - \beta^{\text{idx}} \right\|_2^2 \;+\; \gamma \sum_{t=1}^{T} \left(r_t^{\text{idx}} - w^\top r_t \right)^2, \quad \text{s.t.} \quad \sum_{i=1}^{N} w_i = 1, \quad w_i \geq 0 \quad \forall\, i.$$

Where $\mathbf{B} \in R^{K \times N}$ contains stock-factor loadings and $\beta^{\text{idx}} \in R^K$ the index's factor betas. γ is a penalty parameter controlling the trade-off between tracking error and factor exposure matching. Cross-validation determined the optimal value of $\gamma = 10$. Benefit of this appraoch is that it collapses 25-dimensional problem to a low-rank factor space.

3.6 Iteratively Reweighted L (IRL1)

This approach promotes sparsity in the weight vector while maintaining tracking performance, potentially reducing implementation complexity: At iteration k, solve

$$\min_{w \in R^N} \tfrac{1}{2}\, w^\top P\, w \;+\; \left(c_0 + \lambda\, u^{(k)} \right)^\top w, \quad \text{s.t.} \quad \sum_{i=1}^{N} w_i = 1, \quad w_i \geq 0 \quad \forall\, i,$$

where

$$P = 2\, R^\top R, \quad c_0 = -2\, R^\top r^{\text{idx}}, \quad u_i^{(k)} = \frac{1}{|w_i^{(k)}| + \varepsilon}.$$

Iterate until $\|w^{(k+1)} - w^{(k)}\|_2$ is below tolerance. λ controls the strength of the sparsity penalty. Cross-validation identified the optimal value of $\lambda = 0.0001$.

4 April 2024 Performance Comparison

All series normalized to 100 at the close of April 1, 2024. The best-performing weighting methodology on the validation set was Ridge Regression which was then evaluated on the test set and outperformed the benchmark as seen in the figure below.

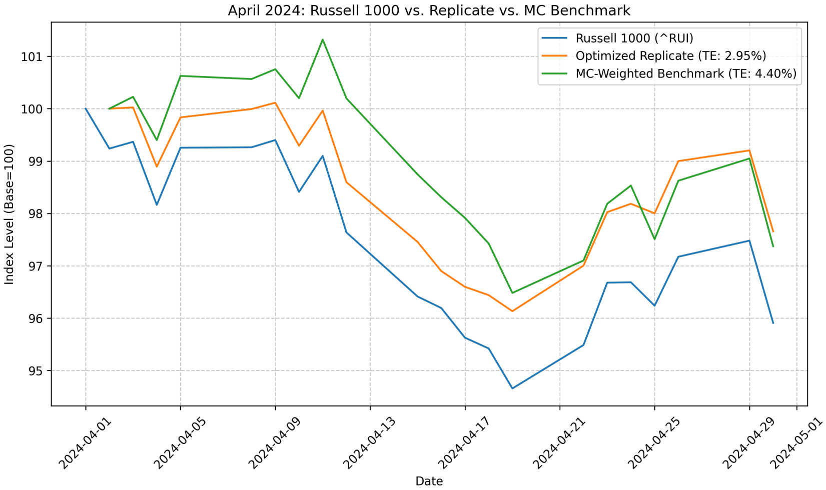

Figure 3: April 2024: Russell 1000 vs. optimized replicate vs. market-cap benchmark.

5 Conclusion

I found that sparse and regularized portfolio construction methods can closely replicate the Russell 1000 with only 25 constituents. Ridge regression performed the best which offers a balance of accuracy and stability. Future improvements can include evaluating alternative lookback window sizes (beyond the 15 months chosen here), sector constraints, and checking robustness across different time periods.