Mini Project Final

Jenny Petrova

April 2025

1 Executive Summary

This report investigates the expected behavior of a sample portfolio, P, invested in the top 400 stocks (sorted by market capitalization) comprising the S&P500 stock index 4. The weekly closing price for each stock is observed over a time period of 2 years (104 weeks), from 2023.04.12 to 2025.04.12. The portfolio's excess returns are determined with respect to the risk-free rate, and the (weekly) sample covariance matrix $S \in \mathbb{R}^{400\times 400}$ is constructed. We use a Principal Component Analysis (PCA) technique to construct a single factor market model. We extract the leading eigenvalue and the corresponding eigenvector of S to construct a single-factor sample covariance matrix, $\Sigma_{PCA} \in \mathbb{R}^{400\times 400}$. The portfolio holdings and the estimated portfolio variance are calculated for the minimum risk, fully invested portfolio, denoted as portfolio C. To improve on our choice of the leading sample eigenvector, we use "James-Stein for eigenvectors" (JSE) to obtain a shrinkage estimator for the leading eigenvector of S. We construct an improved single-factor market model $\Sigma_{JSE} \in \mathbb{R}^{400\times 400}$, and compute the optimized holdings and variance of the portfolio's expected excess returns. The performance of JSE in the context of this project is attributable to the high-dimension low sample size (HL) asymptotic regime of our data. The estimated values from both techniques are analyzed and compared, and the overall feasibility of this portfolio is assessed. We conclude with remarks regarding the level of risk of these investments.

2 Description of the Mathematics

Let p denote the number of observed stocks and let n denote the length of the period observed (in weeks).

2.1 Portfolio Expected Returns

Let P_t be the time t value of portfolio P. The weekly return of each stock i is calculated by

$$r_t^{(i)} = \frac{P_t^{(i)} - P_{t-1}^{(i)}}{P_{t-1}^{(i)}}, \quad 1 \le i \le p, \quad 1 \le t \le n. \tag{1}$$

The weekly *excess* return of each stock i is then found by taking

$$r_P^{(i)}(t) = r_t^{(i)} - r_F, \tag{2}$$

where r_F indicates the risk-free rate (in this scenario, r_F is set to a constant rate for ease of computation). We construct a matrix $X \in \mathbb{R}^{p\times n}$ of weekly excess return such that $r_P^{(i)}(t) \in X$.
Define the expected (average) weekly excess return for each stock i to be

$$f_P^{(i)} = \frac{1}{n}\sum_{t=1}^{n} r_P^{(i)}(t). \tag{3}$$

The de-meaned excess returns for each stock i are found by subtracting $f_P^{(i)}$ from the stock's weekly excess return, such that

$$\tilde{r}_P^{(i)}(t) = r_P^{(i)}(t) - f_P^{(i)}, \quad 1 \le i \le p, \quad 1 \le t \le n \tag{4}$$

for all $r_P^{(i)}(t) \in X$, where $\tilde{r}_P^{(i)}(t) \in Y$.

2.2 Covariance Matrix of Stock Returns

The (weekly) sample covariance matrix S is given by

$$S = \frac{YY^T}{n} \tag{5}$$

In this scenario, S is a singular matrix (not invertible). To proceed with our computations, we must first replace the sample covariance matrix S with a single factor model covariance matrix Σ_{PCA}. Let λ^2 be the leading eigenvalue of S and $\vec{h}$ be the corresponding unit eigenvector. Let I denote the identity matrix, such that $I \in \mathbb{R}^{p\times p}$. Let $tr(S)$ denote the trace of S (sum of the main diagonal entries of S). Compute the average of the non-zero eigenvalues less than λ^2 by

$$l^2 = \frac{tr(S) - \lambda^2}{n-1}. \tag{6}$$

Now we construct the single-factor sample covariance matrix

$$\Sigma_{PCA} = \left((\lambda^2 - l^2)\vec{h}\vec{h}^T\right) + \left(\frac{n}{p} l^2 \times I\right), \quad \Sigma_{PCA} \in \mathbb{R}^{p\times p}. \tag{7}$$

Since the matrix Σ_{PCA} is constructed using the leading eigenvalue of the matrix S and the corresponding eigenvector, Σ_{PCA} has the same leading eigenvector as S and is non-singular (i.e. invertible). We prove these results below.

Proof. Let Σ denote Σ_{PCA}. We want to show Σ is invertible. Take $\vec{x}^T \Sigma \vec{x}$, where $\vec{x}$ is a nonzero real column vector. We show

$$\begin{aligned} \vec{x}^T \, \Sigma \, \vec{x} &= \vec{x}^T \left[\left((\lambda^2 - l^2)\vec{v}\vec{v}^T\right) + \left(\frac{n}{p} l^2 \times I\right) \right] \vec{x} \\ &= (\lambda^2 - l^2)(\vec{x}^T \vec{v})(\vec{v}^T \vec{x}) + \frac{n}{p} l^2 (\vec{x}^T x) \\ &= (\lambda^2 - l^2)\langle x, v\rangle^2 + \frac{n}{p} l^2 ||\vec{x}||_2^2 \end{aligned}$$

The inner product of two real-valued vectors, $\langle x, v\rangle$ is a real-valued scalar. Therefore the squared value yields $\langle x, v\rangle^2 > 0$. Since all the eigenvalues of the sample covariance matrix S are positive, then the average of the nonzero eigenvalues less than λ^2 will be positive ($l^2 > 0$) and smaller than λ^2, therefore $(\lambda^2 - l^2) > 0$. From this argument, it also follows that $\frac{n}{p} l^2 > 0$ for $n > 0$, $p > 0$, and the squared norm of the nonzero vector $\vec{v}$ will be positive such that $||\vec{x}||_2^2 > 0$. Since each value in the above equation is nonzero and positive, we then have

$$\vec{x}^T \, \Sigma \, \vec{x} > 0.$$

Therefore Σ_{PCA} satisfies the conditions for a symmetric positive definite (SPD) matrix which is, by definition, invertible. For a SPD matrix, we also know that all the eigenvalues are real and strictly positive. This leads us to the second part of the proof, where we show Σ_{PCA} will have the same leading eigenvector as S. Let λ^2 be the leading eigenvalue of the sample covariance matrix S, and let $\vec{v}$ be the corresponding unit eigenvector. By definition of eigenvectors and eigenvalues, we know

$$S\vec{v} = \lambda^2 \vec{v}.$$

We want to show $\Sigma\vec{v} = \lambda^2 \vec{v}$. We solve to get

$$\begin{aligned} \Sigma\vec{v} &= \left[\left((\lambda^2 - l^2)\vec{v} \cdot \vec{v}^T\right) + \left(\frac{n}{p} l^2 \times I\right) \right] \vec{v} \\ &= (\lambda^2 - l^2)\vec{v}(\vec{v}^T \vec{v}) + \frac{n}{p} l^2 \vec{v} \\ &= (\lambda^2 - l^2)\vec{v} + \frac{n}{p} l^2 \vec{v} \\ &= (\lambda^2 - l^2 + \frac{n}{p} l^2)\vec{v}, \end{aligned}$$

where the norm of the unit eigenvector is $\vec{v}^T \vec{v} = ||\vec{v}||_2^2 = 1$. Therefore $\Sigma\vec{v} = (\lambda^2 - l^2 + \frac{n}{p} l^2)\vec{v}$ satisfies that $\vec{v}$ is also an eigenvector of Σ, where $\lambda^2 - l^2 + \frac{n}{p} l^2$ is the correspondign eigenvalue. Now, let $\vec{y}$ denote any other eigenvector of Σ. Since Σ is symmetric, then any two eigenvectors associated with different eigenvalues will be orthogonal, meaning $\vec{v}^T\vec{y} = 0$ for any $\vec{y}$. Then

$$\begin{aligned}\Sigma\vec{y} &= \left[\left((\lambda^2 - l^2)\vec{v}\cdot\vec{v}^T\right) + \left(\frac{n}{p}l^2 \times I\right)\right]\vec{y} \\ &= (\lambda^2 - l^2)\vec{v}(\vec{v}^T\vec{y}) + \frac{n}{p}l^2\vec{y} \\ &= \frac{n}{p}l^2\vec{v}.\end{aligned}$$

Therefore for any other eigenvector $\vec{y}$, the corresponding eigenvalue will be $\frac{n}{p}l^2$. We have shown $(\lambda^2 - l^2) > 0$, and thus is it true that

$$\lambda^2 - l^2 + \frac{n}{p}l^2 > \frac{n}{p}l^2,$$

meaning all eigenvalues corresponding to eigenvectors $\vec{y} \neq \vec{v}$ will be smaller than the eigenvalue corresponding to $\vec{v}$. Since $\vec{v}$ corresponds to the largest eigenvalue of Σ, it is therefore also the leading eigenvector of Σ. □

Following from the above proof, Σ_{PCA} is a sufficient estimator of the true covariance matrix.

We want to improve on our estimate of $\vec{h}$, with the JSE estimator $\vec{h}^{JSE}$. Again, we take λ^2 to be the leading eigenvalue of S, but we now wish to construct an *improved* leading eigenvector to enhance the results of the computations. We first scale l^2 (6), the average of the nonzero eigenvalues less than λ^2, by $1/p$. Define this to be

$$v^2 = \frac{1}{p}l^2. \tag{8}$$

The next step relies on an important result of JS estimation. Rather than using the sample average of the data, which may introduce higher squared error for large dimension sizes, the JS estimator lowers the aggregate squared error of a collection of averages by "shrinking" them toward a target (i.e. the collective average). We compute

$$s^2(h) = \frac{1}{p}\sum_{i=1}^{p}(\lambda h_i - \lambda m(h))^2, \tag{9}$$

a measure of the variation of the i-th component of h from the average of the components of h, denoted $m(h)$, scaled by the square root of the corresponding leading eigenvalue (λ^2) of S. Use (8) and (9) to compute the shrinkage constant, c^{JSE}:

$$c^{JSE} = 1 - \frac{v^2}{s^2(h)}. \tag{10}$$

Now we can define the JSE estimator h^{JSE} by shrinking h (i.e. the leading eigenvector of S) towards the average of its components $(m(h))$:

$$h^{JSE} = m(h)\mathbf{1} + c^{JSE}(h - m(h)\mathbf{1}). \tag{11}$$

Normalizing h^{JSE},

$$\hat{h}^{JSE} = \frac{h^{JSE}}{\|h^{JSE}\|} \tag{12}$$

we now have a new unit eigenvector to calculate the single-factor covariance matrix (7), where

$$\Sigma_{JSE} = \left((\lambda^2 - l^2)\hat{h}^{JSE}(\hat{h}^{JSE})^T\right) + \left(\frac{n}{p}l^2 \times I\right), \quad \Sigma_{JSE} \in \mathbb{R}^{p\times p}. \tag{13}$$

2.3 Portfolio Performance Metrics

We now proceed to compute several performance metrics for the constructed portfolio, using $\Sigma = \Sigma_{PCA}$ and $\Sigma = \Sigma_{JSE}$ respectively. The chosen matrix Σ is used to compute the holdings vector $\vec{h}_C$ for the minimum risk, fully invested portfolio C by

$$\vec{h}_C = \frac{\Sigma^{-1} \cdot \vec{e}}{\vec{e}^T \cdot \Sigma^{-1} \cdot \vec{e}} \tag{14}$$

where $\vec{e}$ is a $(1 \times p)$ vector of ones. The value of the expected excess returns of the fully invested portfolio C is

$$f_C = \vec{h}_C^T \cdot \vec{f} \tag{15}$$

where $\vec{f} = [f_P^{(1)}, f_P^{(2)}, \ldots, f_P^{(p)}]$ is the expected excess returns vector for all stocks in portfolio P. We compute the variance of portfolio C by

$$\sigma_C^2 = \vec{h}_C^T \cdot \Sigma \cdot \vec{h}_C, \tag{16}$$

and the standard deviation (i.e. risk) is simply

$$\sigma_C = \sqrt{\vec{h}_C^T \cdot \Sigma \cdot \vec{h}_C}. \tag{17}$$

The variance of each asset i is given by the diagonal entries of the covariance matrix Σ, where

$$[\sigma_1^2, \sigma_2^2, \ldots, \sigma_p^2] = diag(\Sigma). \tag{18}$$

Note that this portfolio is fully invested in risky assets (i.e. no cash assets). Any fully invested portfolio P will have $\sigma_P = \sigma_C$, so in context of this portfolio we may use these variable notations interchangeably.

3 Numerical Results

For our chosen sample portfolio, we observe the closing prices of $p = 400$ stocks over a period of $n = 104$ weeks. All stock (closing) price data is accessed using the 'STOCKHISTORY' command in Excel. The portfolio data is constructed in an Excel spreadsheet and exported into a Python program. All computations are run with Python 3.9.6 using the PyCharm IDE.

The weekly closing stock price data for each stock is obtained using the following form of the 'STOCKHISTORY' Excel command:

=STOCKHISTORY("AAPL","2023-4-12", "2025-4-12",1,1,1),

where each stock's corresponding ticker symbol is changed out in the first entry. In the formula, first "1" indicates "weekly" data, the second "1" indicates the inclusion of headers, and the third "1" indicates retrieving the "Close" prices for each stock. We retrieve the closing prices of stocks beginning in week 2023.04.12 (time $t = 0$) and ending in week 2025.04.12 (time $t = 104$).

3.1 Portfolio Returns

Using the obtained close prices, we calculate the returns from $t = 1$ to $t = 104$. Note that we began with $104 + 1$ initial weekly observations. When differencing the weekly closing prices of each stock, we do not have data prior to our initial time, $t = 0$. Therefore we calculate the returns from $t = 104$ to $t = 1$, and remove the $t = 0$ observation after performing the calculations. We proceed with 104 rows of observations weekly stock returns for 400 different stocks.

Subtracting the risk-free rate r_F from each cell generates a (104×400) matrix P of excess returns $r_P^{(i)}$ for each asset i at each time t (1). This data is exported into Python and the following Python commands are executed:

```
p = 400
n = 104

prices = pd.read_excel("snp500.xlsx", index_col=0)
returns = prices.pct_change(fill_method=None).dropna()
excess_returns = returns - 0.000993
```

We set the risk-free rate to $r_F = 0.000993$. This value is determined by taking the average of daily 3-Month Treasury Bill (T-Bill) rates, corresponding to our observed time period, obtained from public U.S. Department of Treasury records. These rates are often used by U.S.-based investors as a proxy for the true risk-free rate, since investors expect there to be no risk of default on obligations made by the government. The average annualized 3-Month T-Bill rate is found to be $r_{F_{\text{annual}}} \approx 0.516$. Dividing $.045/52 \approx 0.000993$ gives the average weekly rate. To compute the expected excess returns $f_P^{(i)}$ of each asset i (2), we transpose the table (i.e. our matrix P) and calculate the average (*expected* excess returns) for each row i in P. Therefore the vector of expected excess returns is

	Ticker	f_P
1	AAPL	0.000965
2	MSFT	0.001609
3	NVDA	0.015025
4	AMZN	0.005236
5	META	0.008522
6	BRK.B	0.004037
7	GOOGL	0.003775
8	AVGO	0.012255
9	GOOG	0.003844
10	TSLA	0.006074
$\vdots$	$\vdots$	$\vdots$
391	AMCR	-0.001705
392	APTV	-0.006419
393	ARE	-0.004758
394	AVY	-0.000828
395	BALL	-0.002281
396	BBY	-0.001959
397	BLDR	0.001597
398	CAG	-0.004305
399	CF	0.000081
400	CHRW	-0.001569

Based on the observed 104-week (two-year) data, there appears to be a relationship between expected excess returns and market cap (the aggregate market value of a company). Specifically, the top 10 stocks with the largest market caps are expected to have positive expected excess returns. This can likely be attributed to their market dominance and resilience against major economic fluctuations, making them more likely to guarantee stable revenue for investors. In contrast, the bottom 10 stocks (smallest market caps in our portfolio) tend to display negative expected excess returns, possibly due to their higher (relative) vulnerability to market volatility.

3.2 Covariance Matrix of Stock Returns

The expected (average) excess return for stock each i is subtracted from the stock's excess return value at each time t (4). This transforms matrix P into the de-meaned excess returns matrix Y:

```
expected_returns = excess_returns.mean(axis=1)
Y = excess_returns.sub(expected_returns, axis=0)
```

Then the weekly sample covariance matrix S is calculated by taking the inner product of Y (5) such that the following code

```
S = Y @ Y.T / 104
```

yields the desired (400×400) matrix. We find the trace, leading eigenvalue (λ^2) and corresponding unit eigenvector ($\vec{h}$) of S, to calculate l^2 (6):

```
trace_S = np.trace(S)
eigvals, eigvecs = np.linalg.eigh(S)
lambdaS = eigvals[-1]
h = eigvecs[:, -1]
l = (trace_S - lambdaS) / (n-1)
```

The above Python command returns the eigenvalues of the matrix in ascending order, so we choose the last eigenvalue (i.e. the largest eigenvalue) to be λ^2, and the corresponding eigenvector to be our $\vec{h}$. Using these values, we compute the single-factor covariance matrix Σ_{PCA} (7) by:

```
term1 = lambdaS - l
term2 = (n / p) * l
sigma = ((term1) * np.outer(h, h))
        + (term2 * np.eye(p))
```

Now, we apply JSE for eigenvectors to obtain the improved eigenvector estimator $\vec{h}^{JSE}$ (11). The following code is executed:

```
v = l / p
lambda_sqrt = np.sqrt(lambdaS)
m = np.average(h)
sum_var = 0
for i in range(len(h)):
    sum_var += ((lambda_sqrt * h[i])
                 - (lambda_sqrt * m)) ** 2
s = sum_var / p
c_JSE = 1 - (v / s)
h_JSE = (m * ones) + (c_JSE * (h - (m * ones)))
h_JSE = h_JSE / np.linalg.norm(h_JSE)
```

The shrinkage estimator $\vec{h}^{JSE}$ is now used to recompute the single-factor covariance matrix, Σ_{JSE} (13):

```
sigma_JSE = (term1 * np.outer(h_JSE, h_JSE))
            + (term2 * np.eye(p))
```

Note that we again use the same 'term1' and 'term2' variables used in the computations for Σ_{PCA}, and only change out the eigenvector in the equation.

3.3 Portfolio Performance Metrics

We find the holdings vector of excess returns $\vec{h}_C$ (14) for both single-factor covariance matrices we constructed (with and without JSE estimation). From now on, we include code with the general variable 'sigma,' since the same code is executed with Σ_{PCA} and Σ_{JSE} to generate portfolio performance results. We compare the output for both covariance matrices.

```
sigma_inv = np.linalg.inv(sigma)
h_C = sigma_inv @ ones / (ones.T @ sigma_inv @ ones)
```

The holdings vector $h_C^{(PCA)}$ yields 290 positive holdings percentages (indicating long positions in 290 of stocks) and 110 negative holdings (indicating short positions in the remaining 110 stocks). In contrast, the optimization-based holdings vector $h_C^{(JSE)}$ yields fewer positive holdings, indicating long positions in 277 of the stocks, and short positions in 123 of the stocks. Our largest long and short positions are approximately

	Ticker	Largest Long Position	Ticker	Largest Short Position
PCA	DG	0.013062	SMCI	-0.017710
JSE	DG	0.014262	SMCI	-0.020006

Observe that both the leading eigenvalue estimator and the JSE shrinkage estimator take the largest long and short positions in the same stocks (Dollar General Corporation and Super Micro Computer, Inc., respectively). The JSE estimator outputs a holdings vector with larger positive and lower negative holdings than the eigenvalue estimator, meaning the performance-optimized approach takes larger long and short positions in these stocks.

The value of the weekly expected excess return f_c (15) of the portfolio is computed by taking the inner product of the holdings vector $\vec{h}_C$ and excess returns vector $\vec{f}$.

```
portfolio_expected_returns = h_C.T @ expected_returns
```

We perform this computation for both methods. The annualized expected excess returns are found by scaling f_C by 52. The approximate results are

Expected Excess Returns	PCA	JSE
$f_C^{(weekly)}$	-0.001585	-0.001945
$f_C^{(annual)}$	-0.082446	-0.101158

The JSE shrinkage estimator approximates lower expected excess returns than the eigenvalue estimator, although both methods compute negative expected excess returns for our minimum risk, fully invested portfolio.

Lastly, we are interested in the variance and standard deviation of the fully invested portfolio C (16), as well as the variance and standard deviation of the excess returns for each individual stock i in our original portfolio P, stored in the diagonal entries of each sample covariance matrix Σ (18). We compute

```
portfolio_var = h_C.T @ V @ h_C
portfolio_std_dev = np.sqrt(portfolio_var)
stock_var = np.diag(V)
```

The resulting outputs, and the corresponding annualized values (scaled by 52), are displayed in the following tables:

Weekly Estimates	Portfolio C (PCA)	Portfolio C (JSE)
Variance (σ^2)	1.237263e-05	1.471901e-05
Standard Deviation (σ)	0.003517	0.003837

Annualized Estimates	Portfolio C (PCA)	Portfolio C (JSE)
Variance (σ^2)	0.000643	0.000765
Standard Deviation (σ)	0.025365	0.027666

The JSE shrinkage estimator is used to improve our estimates of the standard deviation of returns (i.e. the volatility) of the portfolio. The results of the JSE application show that, compared to the eigenvalue estimator, our portfolio is expected to be *more* volatile than previously expected. Therefore the standard PCA output underestimates the risk in the portfolio's investments.

We now examine the stocks with the top 10 highest variances (and hence highest standard deviations) of expected excess returns. The following are the most *volatile* stocks in our portfolio, per method:

Ticker	σ^2 (PCA)	σ (PCA)	Ticker	σ^2 (JSE)	σ (JSE)
SMCI	0.005214	0.07221	SMCI	0.004788	0.069197
PLTR	0.003704	0.060863	PLTR	0.003477	0.058964
MPWR	0.00312	0.05586	MPWR	0.002966	0.054459
MU	0.003112	0.055783	MU	0.002958	0.05439
CCL	0.002928	0.05411	CCL	0.002797	0.052884
VST	0.002898	0.053829	VST	0.00277	0.05263
KKR	0.002896	0.053818	KKR	0.002769	0.05262
INTC	0.002865	0.053523	INTC	0.002741	0.052354
URI	0.002815	0.053057	URI	0.002697	0.051935
DELL	0.002797	0.052885	DELL	0.002681	0.05178

We see that both methods estimate the same 10 companies to have the highest volatility of expected excess returns. We also examine the stocks with the bottom 10 lowest variances (and hence lowest standard deviations) of expected excess returns, for each method. Here, see a difference in the stocks expected to have the least volatility. The following are the most *stable* stocks in our portfolio, per method:

Ticker	σ^2 (PCA)	σ (PCA)	Ticker	σ^2 (JSE)	σ (JSE)
CAG	0.001149	0.033892	GIS	0.001148	0.033889
ED	0.00115	0.033915	MOH	0.00115	0.03391
CBOE	0.001151	0.033932	CAG	0.001155	0.03398
CNC	0.001152	0.033936	DG	0.001155	0.033992
MDLZ	0.001153	0.033953	ED	0.001159	0.034044
GIS	0.001154	0.033972	CBOE	0.001161	0.034078
KHS	0.001154	0.033975	CNC	0.001162	0.034086
UNH	0.001154	0.033978	MDLZ	0.001164	0.034117
HSY	0.001155	0.033984	KHC	0.001166	0.034153
DUK	0.001157	0.034021	UNH	0.001167	0.034157

The above results are used to construct visual representations of the data. We first observe the relationship between the level of risk associated with these assets and the portfolio holdings (h_C) for the minimum-risk fully invested portfolio C. We compare the relationship between risk and holdings for the leading eigenvalue estimator and the JSE estimator. A particularly interesting phenomenon is observed:

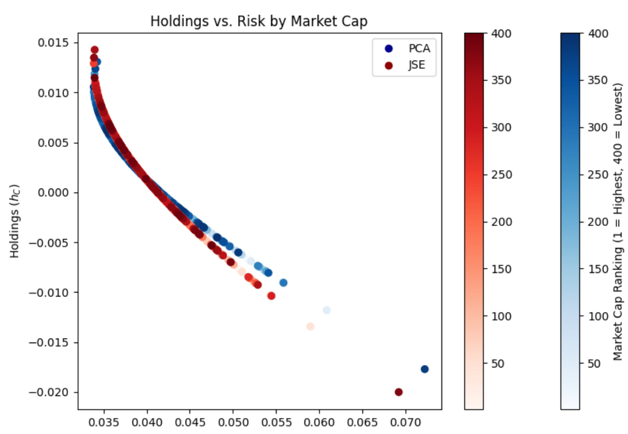

Figure 1: Comparing Asset Weights to Risk for PCA vs. JSE

There exists is a strong inverse linear relationship between risk and holdings!

Notice that many points with low risk (left side of x-axis) have higher (positive) holdings on the y-axis, while points with high risk (right side of x-axis) have lower or negative holdings. This suggests both of our examined strategies (eigenvector optimizers) are over-weighting (taking long positions on) safer, low-volatility stocks and under-weighting (taking short positions on) riskier stocks. The color scale is an added visual component. The larger-cap stocks in our portfolio tend to get a heavier weighting, while the smaller-cap stocks tend to get a lesser or negative weighting. For lower risk investments, we hold more assets, and the portfolio is more diversified. Ultimately, this plot makes sense. The observed allocation pattern suggest our chosen strategies minimize overall portfolio risk by concentrating holdings in more stable, large-cap securities, while using short positions to hedge exposures to volatile, smaller-cap stocks.

The next plots visualizes the annualized expected excess returns of each stock in our portfolio. Observe the following:

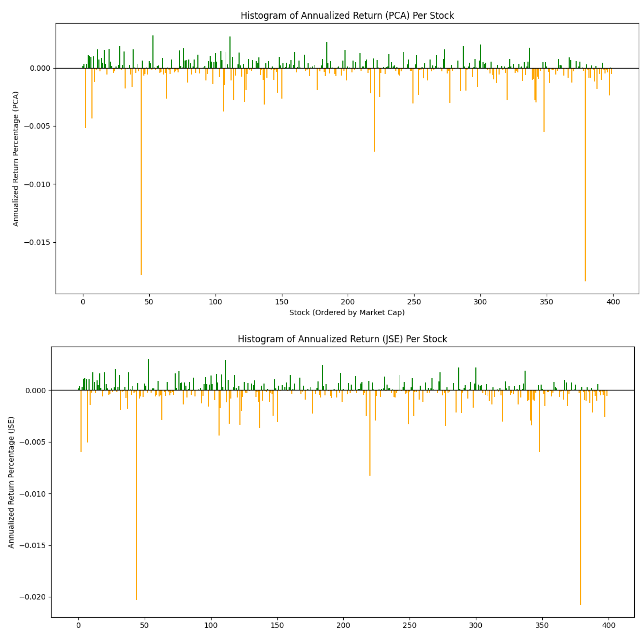

Figure 2: Annualized Expected Excess Returns of Assets

From the previously displayed tabulated data, we know the JSE estimator computations result in *lower* annualized portfolio expected excess returns overall. Visual comparison of both plots suggests no dramatic difference in the expected excess returns of each stock (between the two estimators used). However, this does not imply the JSE estimation strategy is ineffective! Small variations in individual stock returns, which might not stand out in these histograms, can still have a significant impact when aggregated into a weighted (and fully-invested) portfolio. The JSE method enhances our estimation strategy, while still maintaining the expected excess return pattern obtained from our original (leading-eigenvalue) estimator.

Our final plot compares the annualized standard deviation of the portfolio assets to the expected excess returns of the assets. We again compare these values for these metrics between both estimation strategies:

Figure 3: Comparing Annualized Risk to Expected Excess Returns of Assets

This plot allows us to discern the effect between the JSE and PCA estimators on the portfolio's performance metrics. The standard deviations of the assets measured with the JSE method (displayed in magenta) are higher than the standard deviations of the assets measured with the standard PCA method. The aggregated effect of these differences is displayed in the results of the fully-invested, minimum risk portfolio estimates. Both methods construct a diversified portfolio, such that the annualized portfolio risk is significantly lower than the annualized risk of the individual assets in the portfolio. The JSE method estimates a portfolio with slightly more negative returns and slightly higher risk, compared to the PCA method. Overall, we observe leading eigenvalue estimator *under*-estimates the level of potential risk in these investments and, as a result, under-estimates the expected negative returns of this portfolio.

4 Analysis of Results and Conclusion

The S&P500 is a stock index that tracks the performance of 500 of the largest publicly traded companies in the U.S. and is often regarded as the best overall measure of the U.S. stock market's performance. We measured weekly two-year data of the top 400 stocks (by market cap) that are tracked by the S&P500 index. As such, our constructed portfolio is entirely comprised of risky assets and we aim to improve on our valuation of the risk of these investments. Both estimators suggest this portfolio will under-perform, signaling a potential downturn in the market. Further, the JSE estimator suggests that the standard method for portfolio estimation will under-estimate the level of risk and expected loss of investing in our constructed portfolio. Given these results, investing in this portfolio may not be advisable, as it is expected to provide negative returns. An alternative investment option, based on these findings, would be in 'riskless assets' (such as U.S. treasuries securities), which offer a nearly guaranteed return (though, often smaller return) without the uncertainty associated with a risky portfolio.

Appendix A - Citations

[1] L.R. Goldberg, & A.N. Kercheval, James–Stein for the leading eigenvector, *Proc. Natl. Acad. Sci. U.S.A.* 120 (2) e2207046120, https://doi.org/10.1073/pnas.2207046120 (2023).

[2] Slickcharts. (n.d.). *S&P 500 ETF components.* Retrieved April 8, 2025, from https://www.slickcharts.com/sp500

[3] U.S. Department of the Treasury. Daily Treasury Bill Rates - 2025. https://home.treasury.gov/resource-center/data-chart-center/ interest-rates/TextView? Type=daily_treasury_bill_rates&field_tdr_date_value=2025, 2025. Accessed: 2025-04-10.

Appendix B - Complete Stock List

No.	Ticker	Company Name
1	AAPL	Apple Inc.
2	MSFT	Microsoft Corporation
3	NVDA	NVIDIA Corporation
4	AMZN	Amazon.com, Inc.
5	META	Meta Platforms, Inc.
6	BRK-B	Berkshire Hathaway Inc.
7	GOOGL	Alphabet Inc. (Class A)
8	AVGO	Broadcom Inc.
9	GOOG	Alphabet Inc. (Class C)
10	TSLA	Tesla, Inc.
11	JPM	JPMorgan Chase& Co.
12	LLY	Eli Lilly and Co.
13	V	Visa Inc.
14	XOM	Exxon Mobil Corporation
15	UNH	UnitedHealth Group Incorporated
16	MA	Mastercard Incorporated
17	COST	Costco Wholesale Corporation
18	NFLX	Netflix, Inc.
19	JNJ	Johnson & Johnson
20	PG	Procter & Gamble Co.
21	WMT	Walmart Inc.
22	ABBV	AbbVie Inc.
23	HD	Home Depot, Inc.
24	CVX	Chevron Corporation
25	BAC	Bank of America Corporation
26	KO	Coca-Cola Company
27	CRM	Salesforce, Inc.
28	CSCO	Cisco Systems, Inc.
29	PM	Philip Morris International
30	WFC	Wells Fargo & Co.
31	ABT	Abbott Laboratories
32	IBM	International Business Machines Corporation
33	MRK	Merck & Co., Inc.
34	ORCL	Oracle Corporation
35	MCD	McDonald's Corporation
36	LIN	Linde plc
37	GE	General Electric Company
38	PEP	PepsiCo, Inc.
39	T	AT&T Inc.
40	ACN	Accenture plc
41	TMO	Thermo Fisher Scientific Inc.
42	VZ	Verizon Communications Inc.
43	DIS	The Walt Disney Company
44	ISRG	Intuitive Surgical, Inc.
45	PLTR	Palantir Technologies Inc.
46	RTX	Raytheon Technologies Corporation
47	GS	Goldman Sachs Group, Inc.
48	QCOM	Qualcomm Incorporated
49	ADBE	Adobe Inc.
50	AMD	Advanced Micro Devices, Inc.
51	AMGN	Amgen Inc.
52	INTU	Intuit Inc.
53	NOW	ServiceNow, Inc.

No.	Ticker	Company Name
54	PGR	Progressive Corporation
55	TXN	Texas Instruments Incorporated
56	CAT	Caterpillar Inc.
57	SPGI	S&P Global Inc.
58	BKNG	Booking Holdings Inc.
59	UBER	Uber Technologies Inc.
60	AXP	American Express Company
61	BSX	Boston Scientific Corporation
62	NEE	NextEra Energy, Inc.
63	MS	Morgan Stanley
64	PFE	Pfizer Inc.
65	UNP	Union Pacific Corporation
66	BLK	BlackRock, Inc.
67	CMCSA	Comcast Corporation
68	GILD	Gilead Sciences, Inc.
69	HON	Honeywell International Inc.
70	C	Citigroup Inc.
71	COP	ConocoPhillips
72	DHR	Danaher Corporation
73	SCHW	Charles Schwab Corporation
74	TJX	The TJX Companies, Inc.
75	BA	Boeing Company
76	LOW	Lowe's Companies, Inc.
77	TMUS	T-Mobile US, Inc.
78	VRTX	Vertex Pharmaceuticals Incorporated
79	ADP	Automatic Data Processing, Inc.
80	BMY	Bristol-Myers Squibb Company
81	FI	Fleetcor Technologies, Inc.
82	SYK	Stryker Corporation
83	AMAT	Applied Materials, Inc.
84	DE	Deere & Company
85	MMC	Marsh & McLennan Companies, Inc.
86	MDT	Medtronic plc
87	PANW	Palo Alto Networks, Inc.
88	CB	Chubb Limited
89	ETN	Eaton Corporation plc
90	SBUX	Starbucks Corporation
91	PLD	Prologis, Inc.
92	ADI	Analog Devices, Inc.
93	AMT	American Tower Corporation
94	BX	Blackstone Inc.
95	ELV	Elevance Health, Inc.
96	ICE	Intercontinental Exchange, Inc.
97	INTC	Intel Corporation
98	MO	Altria Group, Inc.
99	MU	Micron Technology, Inc.
100	SO	Southern Company
101	CME	CME Group Inc.
102	DUK	Duke Energy Corporation
103	LMT	Lockheed Martin Corporation
104	LRCX	Lam Research Corporation
105	WELL	Welltower Inc.
106	CI	Cigna Corporation
107	CRWD	CrowdStrike Holdings, Inc.
108	KLAC	KLA Corporation
109	AJG	Arthur J. Gallagher & Co.
110	AON	Aon plc

No.	Ticker	Company Name
111	CVS	CVS Health Corporation
112	MCK	McKesson Corporation
113	MDLZ	Mondelez International, Inc.
114	WM	Waste Management, Inc.
115	ANET	Arista Networks, Inc.
116	APH	Amphenol Corporation
117	EQIX	Equinix, Inc.
118	MMM	3M Company
119	ORLY	O'Reilly Automotive, Inc.
120	SHW	Sherwin-Williams Company
121	UPS	United Parcel Service, Inc.
122	CL	Colgate-Palmolive Company
123	KKR	KKR & Co. Inc.
124	NKE	Nike, Inc.
125	PH	Parker-Hannifin Corporation
126	TDG	TransDigm Group Incorporated
127	TT	Trane Technologies plc
128	CDNS	Cadence Design Systems, Inc.
129	CTAS	Cintas Corporation
130	EOG	EOG Resources, Inc.
131	MCO	Moody's Corporation
132	MSI	Motorola Solutions, Inc.
133	NOC	Northrop Grumman Corporation
134	WMB	Williams Companies, Inc.
135	ZTS	Zoetis Inc.
136	APD	Air Products and Chemicals, Inc.
137	BDX	Becton Dickinson and Company
138	CEG	Constellation Energy Corporation
139	CMG	Chipotle Mexican Grill, Inc.
140	COF	Capital One Financial Corporation
141	GD	General Dynamics Corporation
142	ITW	Illinois Tool Works Inc.
143	PNC	PNC Financial Services Group, Inc.
144	PYPL	PayPal Holdings, Inc.
145	REGN	Regeneron Pharmaceuticals, Inc.
146	SNPS	Synopsys, Inc.
147	USB	U.S. Bancorp
148	APO	Apollo Global Management, Inc.
149	AZO	AutoZone, Inc.
150	BK	Bank of New York Mellon Corporation
151	DASH	DoorDash, Inc.
152	ECL	Ecolab Inc.
153	EMR	Emerson Electric Co.
154	FTNT	Fortinet, Inc.
155	HCA	HCA Healthcare, Inc.
156	OKE	ONEOK, Inc.
157	ROP	Roper Technologies, Inc.
158	TRV	The Travelers Companies, Inc.
159	ADSK	Autodesk, Inc.
160	AEP	American Electric Power Company, Inc.
161	ALL	The Allstate Corporation
162	CSX	CSX Corporation
163	FCX	Freeport-McMoRan Inc.
164	KMI	Kinder Morgan, Inc.
165	NEM	Newmont Corporation
166	SLB	Schlumberger Limited
167	ABNB	Airbnb, Inc.

No.	Ticker	Company Name
168	AFL	Aflac Incorporated
169	CARR	Carrier Global Corporation
170	FDX	FedEx Corporation
171	HLT	Hilton Worldwide Holdings Inc.
172	HWM	Howmet Aerospace Inc.
173	JCI	Johnson Controls International plc
174	MAR	Marriott International, Inc.
175	NSC	Norfolk Southern Corporation
176	PCAR	PACCAR Inc.
177	PSX	Phillips 66
178	RCL	Royal Caribbean Group
179	SPG	Simon Property Group, Inc.
180	TFC	Truist Financial Corporation
181	WDAY	Workday, Inc.
182	AIG	American International Group, Inc.
183	AMP	Ameriprise Financial, Inc.
184	CCI	Crown Castle International Corp.
185	COR	CoreSite Realty Corporation
186	CPRT	Copart, Inc.
187	D	Dominion Energy, Inc.
188	DLR	Digital Realty Trust, Inc.
189	EXC	Exelon Corporation
190	GM	General Motors Company
191	KMB	Kimberly-Clark Corporation
192	KVUE	KVUE, Inc.
193	MET	MetLife, Inc.
194	MPC	Marathon Petroleum Corporation
195	NXPI	NXP Semiconductors N.V.
196	O	Realty Income Corporation
197	PAYX	Paychex, Inc.
198	PSA	Public Storage
199	RSG	Republic Services, Inc.
200	TGT	Target Corporation
201	BKR	Baker Hughes Company
202	CMI	Cummins Inc.
203	CTVA	Corteva, Inc.
204	EW	Edwards Lifesciences Corporation
205	FAST	Fastenal Company
206	FICO	Fair Isaac Corporation
207	GWW	Grainger, Inc.
208	HES	Hess Corporation
209	KDP	Keurig Dr Pepper Inc.
210	KR	The Kroger Co.
211	MNST	Monster Beverage Corporation
212	MSCI	MSCI Inc.
213	OTIS	Otis Worldwide Corporation
214	PEG	Public Service Enterprise Group Incorporated
215	ROST	Ross Stores, Inc.
216	SRE	Sempra Energy
217	TRGP	Targa Resources Corp.
218	URI	United Rentals, Inc.
219	VLO	Valero Energy Corporation
220	VRSK	Verisk Analytics, Inc.
221	VST	Vistra Corp.
222	YUM	Yum! Brands, Inc.
223	ACGL	Arch Capital Group Ltd.
224	AME	A. O. Smith Corporation

No.	Ticker	Company Name
225	AXON	Axon Enterprise, Inc.
226	CBRE	CBRE Group, Inc.
227	CHTR	Charter Communications, Inc.
228	CTSH	Cognizant Technology Solutions Corporation
229	DFS	Discover Financial Services
230	DHI	D.R. Horton, Inc.
231	ED	Consolidated Edison, Inc.
232	ETR	Entergy Corporation
233	F	Ford Motor Company
234	FIS	FIS Global
235	GEHC	GE HealthCare Technologies Inc.
236	LHX	L3Harris Technologies, Inc.
237	PCG	Pacific Gas and Electric Company
238	PRU	Prudential Financial, Inc.
239	PWR	Quanta Services, Inc.
240	SYY	Sysco Corporation
241	XEL	Xcel Energy Inc.
242	A	Agilent Technologies, Inc.
243	CAH	Cardinal Health, Inc.
244	CSGP	CoStar Group, Inc.
245	DD	DuPont de Nemours, Inc.
246	EA	Electronic Arts Inc.
247	EBAY	eBay Inc.
248	EQT	EQT Corporation
249	EXR	Extra Space Storage Inc.
250	GIS	General Mills, Inc.
251	GLW	Corning Incorporated
252	GRMN	Garmin Ltd.
253	HIG	The Hartford
254	HUM	Humana Inc.
255	IDXX	IDEXX Laboratories, Inc.
256	IR	Ingersoll Rand Inc.
257	IT	Gartner, Inc.
258	LULU	Lululemon Athletica, Inc.
259	NDAQ	Nasdaq, Inc.
260	ODFL	Old Dominion Freight Line, Inc.
261	OXY	Occidental Petroleum Corporation
262	RMD	ResMed Inc.
263	TTWO	Take-Two Interactive Software, Inc.
264	VICI	VICI Properties Inc.
265	VMC	Vulcan Materials Company
266	WAB	Westinghouse Air Brake Technologies Corporation
267	WEC	WEC Energy Group, Inc.
268	WTW	WeightWatchers International, Inc.
269	AEE	Ameren Corporation
270	ANSS	ANSYS, Inc.
271	AVB	AvalonBay Communities, Inc.
272	AWK	American Water Works Company, Inc.
273	BR	Broadridge Financial Solutions, Inc.
274	BRO	Brown & Brown, Inc.
275	CHD	Church & Dwight Co., Inc.
276	CNC	Centene Corporation
277	DAL	Delta Air Lines, Inc.
278	DELL	Dell Technologies Inc.
279	DTE	DTE Energy Company
280	DXCM	Dexcom, Inc.
281	EFX	Equifax Inc.

No.	Ticker	Company Name
282	FANG	Diamondback Energy, Inc.
283	HPQ	HP Inc.
284	IP	International Paper Company
285	IQV	IQVIA Holdings Inc.
286	KHC	The Kraft Heinz Company
287	LEN	Lennar Corporation
288	MCHP	Microchip Technology Inc.
289	MLM	Martin Marietta Materials, Inc.
290	MPWR	Monolithic Power Systems, Inc.
291	MTB	M&T Bank Corporation
292	NUE	Nucor Corporation
293	PPL	PPL Corporation
294	ROK	Rockwell Automation, Inc.
295	STZ	Constellation Brands, Inc.
296	TSCO	Tractor Supply Company
297	VTR	Ventas, Inc.
298	XYL	Xylem Inc.
299	ADM	Archer-Daniels-Midland Company
300	ATO	Atmos Energy Corporation
301	CBOE	Cboe Global Markets, Inc.
302	CDW	CDW Corporation
303	CINF	Cincinnati Financial Corporation
304	CMS	CMS Energy Corporation
305	CNP	CenterPoint Energy, Inc.
306	CPAY	Cloudflare, Inc.
307	DOV	Dover Corporation
308	DOW	Dow Inc.
309	DRI	Darden Restaurants, Inc.
310	DVN	Devon Energy Corporation
311	EIX	Edison International
312	EQR	Equity Residential
313	ES	Eversource Energy
314	EXE	Exelon Corporation
315	FITB	Fifth Third Bancorp
316	FTV	Fortive Corporation
317	GDDY	GoDaddy, Inc.
318	GPN	Global Payments Inc.
319	HAL	Halliburton Company
320	HBAN	Huntington Bancshares Incorporated
321	HSY	Hershey Company
322	IRM	Iron Mountain Incorporated
323	K	Kellogg Company
324	KEYS	Keysight Technologies, Inc.
325	MTD	Mettler-Toledo International Inc.
326	NVR	NVR, Inc.
327	PPG	PPG Industries, Inc.
328	RJF	Raymond James Financial, Inc.
329	SBAC	SBA Communications Corporation
330	STE	Stryker Corporation
331	STT	State Street Corporation
332	TDY	Teledyne Technologies Incorporated
333	TPL	Texas Pacific Land Trust
334	TYL	Tyler Technologies, Inc.
335	UAL	United Airlines Holdings, Inc.
336	WAT	Waters Corporation
337	WBD	Warner Bros. Discovery, Inc.
338	WRB	W.R. Berkley Corporation

No.	Ticker	Company Name
339	WY	Weyerhaeuser Company
340	ZBH	Zimmer Biomet Holdings, Inc.
341	BAX	Baxter International Inc.
342	BIIB	Biogen Inc.
343	CCL	Carnival Corporation
344	CFG	Citizens Financial Group, Inc.
345	CLX	Clorox Company
346	COO	CooperCompanies, Inc.
347	CTRA	Coterra Energy, Inc.
348	DECK	Deckers Outdoor Corporation
349	DG	Dollar General Corporation
350	DGX	Quest Diagnostics Incorporated
351	ESS	Essex Property Trust, Inc.
352	EXPD	Expeditors International of Washington, Inc.
353	EXPE	Expedia Group, Inc.
354	FDS	FactSet Research Systems Inc.
355	FE	FirstEnergy Corporation
356	HPE	Hewlett Packard Enterprise Company
357	HUBB	Hubbell Incorporated
358	IFF	International Flavors & Fragrances Inc.
359	INVH	Invitation Homes Inc.
360	LDOS	Leidos Holdings, Inc.
361	LH	Laboratory Corporation of America Holdings
362	LII	Lennox International Inc.
363	LUV	Southwest Airlines Co.
364	LYB	LyondellBasell Industries N.V.
365	LYV	Live Nation Entertainment, Inc.
366	MAA	Mid-America Apartment Communities, Inc.
367	MKC	McCormick & Company, Incorporated
368	MOH	Molina Healthcare, Inc.
369	NI	NiSource Inc.
370	NRG	NRG Energy, Inc.
371	NTAP	NetApp, Inc.
372	NTRS	Northern Trust Corporation
373	ON	ON Semiconductor Corporation
374	PFG	Principal Financial Group, Inc.
375	PHM	PulteGroup, Inc.
376	PKG	Packaging Corporation of America
377	PODD	Insulet Corporation
378	PTC	PTC Inc.
379	RF	Regions Financial Corporation
380	SMCI	Super Micro Computer, Inc.
381	SNA	Snap-on Incorporated
382	STLD	Steel Dynamics, Inc.
383	STX	Seagate Technology Holdings PLC
384	SYF	Synchrony Financial
385	TROW	T. Rowe Price Group, Inc.
386	TSN	Tyson Foods, Inc.
387	ULTA	Ulta Beauty, Inc.
388	VRSN	Verisign, Inc.
389	WSM	Williams-Sonoma, Inc.
390	AKAM	Akamai Technologies, Inc.
391	AMCR	Amcor plc
392	APTV	Aptiv PLC
393	ARE	Alexandria Real Estate Equities, Inc.
394	AVY	Avery Dennison Corporation
395	BALL	Ball Corporation

No.	Ticker	Company Name
396	BBY	Best Buy Co., Inc.
397	BLDR	Builders FirstSource, Inc.
398	CAG	Conagra Brands, Inc.
399	CF	CF Industries Holdings, Inc.
400	CHRW	C.H. Robinson Worldwide, Inc.

Appendix C - Complete Code

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import matplotlib.colors as mcolors
import matplotlib.cm as cm
from matplotlib.lines import Line2D


p = 400
n = 104
ones = np.ones(p) # to be used in several computations

prices = pd.read_excel("snp500.xlsx", index_col=0) #
    weekly closing price data
returns = prices.pct_change(fill_method=None).dropna()
     # weekly returns
excess_returns = returns - 0.000993 # excess returns
excess_returns = excess_returns.T # (p x n) matrix
expected_returns = excess_returns.mean(axis=1) # de-
    meaned excess returns matrix Y
expected_returns_table = expected_returns.to_latex(
    index=True)
#print(expected_returns_table)
Y = excess_returns.sub(expected_returns, axis=0)
S = Y @ Y.T / n # sample covariance matrix S

trace_S = np.trace(S)
eigenvals, eigenvecs = np.linalg.eigh(S)
lambdaS = eigenvals[-1] # leading eigenvalue
h = eigenvecs[:, -1] # estimator (leading eigenvector)
l = (trace_S - lambdaS) / (n-1)
term1 = lambdaS - l
term2 = (n / p) * l
sigma = (term1 * np.outer(h, h)) + (term2 * np.eye(p))
     # single factor model covariance matrix sigma_PCA

eigenval1, eigenvec1 = np.linalg.eigh(sigma) # verify
    identical leading eigenvector
lambda1 = eigenval1[-1]


"---------------------------JSE FOR EIGENVECTORS
    ---------------------------"
```

```python
v = 1 / p
lambda_sqrt = np.sqrt(lambdaS)
m = np.average(h)
m_h = lambda_sqrt * m
sum = 0
for i in range(len(h)):
    sum += ((lambda_sqrt * h[i]) - m_h) ** 2
s = sum / p
c_JSE = 1 - (v / s)
h_JSE = (m * ones) + (c_JSE * (h - (m * ones)))
h_JSE = h_JSE / np.linalg.norm(h_JSE)
sigma_JSE = (term1 * np.outer(h_JSE, h_JSE)) + (term2 * np.eye(p)) # single factor model covariance matrix sigma_JSE


"---------------------------PORTFOLIO PERFORMANCE METRICS---------------------------"
# Metrics for sigma_PCA
sigma_inv = np.linalg.inv(sigma)
holdings = sigma_inv @ ones / (ones.T @ sigma_inv @ ones) # holdings for min var, fully invested portfolio C
#print(np.sum(np.array(holdings) > 0, axis=0)) # number of positive holdings
maxh = np.max(holdings)
max_id = np.argmax(holdings)
#print(maxh, max_id) # max holdings and index
minh = np.min(holdings)
min_id = np.argmin(holdings) # min holdings and index
#print(minh, min_id)

portfolio_expected_returns = holdings.T @ expected_returns
print("portfolio expected excess returns:", portfolio_expected_returns)
portfolio_var = holdings.T @ sigma @ holdings
#print("portfolio variance:", portfolio_var)
portfolio_std_dev = np.sqrt(portfolio_var)
#print("portfolio standard deviation:", portfolio_std_dev)
stock_var = np.diag(sigma)
stock_std_dev = np.sqrt(stock_var)

# annualized results (scaled by 52)
annualized_return = portfolio_expected_returns * 52
```

```
#print("annualized returns", annualized_return)
annualized_var = portfolio_var * 52
#print("annualized variance", annualized_var)
annualized_std_dev = portfolio_std_dev * np.sqrt(52)
#print("annualized standard deviation",
    annualized_std_dev)
annualized_stock_var = stock_var * 52
annualized_stock_std_dev = stock_std_dev * np.sqrt(52)


# Metrics for sigma_JSE
sigma_inv_JSE = np.linalg.inv(sigma_JSE)
holdings_JSE = sigma_inv_JSE @ ones / (ones.T @
    sigma_inv_JSE @ ones)
#print(np.sum(np.array(holdings_JSE) > 0, axis=0))
maxh_JSE = np.max(holdings_JSE)
max_id_JSE = np.argmax(holdings_JSE)
#print(maxh_JSE, max_id_JSE) # max holdings and index
minh_JSE = np.min(holdings_JSE)
min_id_JSE = np.argmin(holdings_JSE) # min holdings
    and index
#print(minh_JSE, min_id_JSE)

portfolio_expected_returns_JSE = holdings_JSE.T @
    expected_returns
#print("portfolio expected excess returns (JSE)",
    portfolio_expected_returns_JSE)
portfolio_var_JSE = holdings_JSE.T @ sigma_JSE @
    holdings_JSE
#print("portfolio variance (JSE)", portfolio_var_JSE)
portfolio_std_dev_JSE = np.sqrt(portfolio_var_JSE)
#print("portfolio standard deviation (JSE)",
    portfolio_std_dev_JSE)
stock_var_JSE = np.diag(sigma_JSE)
stock_std_dev_JSE = np.sqrt(stock_var_JSE)

# (JSE) annualized results (scaled by 52)
annualized_return_JSE = portfolio_expected_returns_JSE
     * 52
#print("annualized returns (JSE)",
    annualized_return_JSE)
annualized_var_JSE = portfolio_var_JSE * 52
#print("annualized variance (JSE)", annualized_var_JSE
    )
annualized_std_dev_JSE = portfolio_std_dev_JSE * np.
    sqrt(52)
```

```
#print("annualized standard deviation (JSE)", annualized_std_dev_JSE)
annualized_stock_var_JSE = stock_var_JSE * 52
annualized_stock_std_dev_JSE = stock_std_dev_JSE * np.sqrt(52)


# Sorting variance and standard deviation
# No JSE
sorted_indices1 = np.argsort(stock_var)
bottom10_indices1 = sorted_indices1[:10]
top10_indices1 = sorted_indices1[-10:][::-1]

# JSE
sorted_indices = np.argsort(stock_var_JSE)
bottom10_indices = sorted_indices[:10]
top10_indices = sorted_indices[-10:][::-1]
# print("\nTop 10 values and their indices:")
# for idx in top10_indices1:
#     print(f"{idx} & {round(stock_var[idx], 6)} & {round(stock_std_dev[idx], 6)}")
# print("JSE:")
# for idx in top10_indices:
#     print(f"{idx} & {round(stock_var_JSE[idx],6)} & {round(stock_std_dev_JSE[idx],6)}")
#
# print("\nBottom 10 values and their indices:")
# for idx in bottom10_indices1:
#     print(f"{idx} & {round(stock_var[idx], 6)} & {round(stock_std_dev[idx], 6)}")
# print("JSE:")
# for idx in bottom10_indices:
#     print(f"{idx} & {round(stock_var_JSE[idx],6)} & {round(stock_std_dev_JSE[idx],6)}")


"---------------------------PLOTS !!---------------------------"
# PLOT 1 -
row = prices.iloc[0]
new_df = pd.DataFrame([row])
for i in range(p):
    new_df.iloc[0, i] = i+1

x1 = stock_std_dev
y1 = holdings
```

```
x2 = stock_std_dev_JSE
y2 = holdings_JSE
z1 = new_df.iloc[0, :]
fig, ax = plt.subplots(figsize=(9, 6))
ax.scatter(x1, y1, label='PCA', c=z1, cmap='Blues')
ax.scatter(x2, y2, label='JSE', c=z1, cmap='Reds')
norm = mcolors.Normalize(vmin=z1.min(), vmax=z1.max())
sm = cm.ScalarMappable(norm=norm, cmap='Blues')
sm2 = cm.ScalarMappable(norm=norm, cmap='Reds')
sm.set_array([])
sm2.set_array([])
cbar = plt.colorbar(sm, ax=ax)
cbar2 = plt.colorbar(sm2, ax=ax)
cbar.set_label("Market Cap Ranking (1 = Highest, 400 = Lowest)".format(z1.min()))
legend_elements = [
    Line2D([0], [0], marker='o', color='w', label='PCA',
           markerfacecolor='darkblue', markersize=8),
    Line2D([0], [0], marker='o', color='w', label='JSE',
           markerfacecolor='darkred', markersize=8)
]
ax.legend(handles=legend_elements)
plt.xlabel('Risk ($\sigma$)')
plt.ylabel('Holdings ($h_C$)')
plt.title('Holdings vs. Risk by Market Cap')
plt.savefig('holdings_risk.png')
plt.show()

weight_PCA = holdings * expected_returns * 52
weight_JSE = holdings_JSE * expected_returns * 52
stock_indices = np.arange(p)
plt.figure(figsize=(12, 6))
colors_PCA = ['green' if x >= 0 else 'orange' for x in weight_PCA]
plt.bar(stock_indices, weight_PCA, color=colors_PCA)
plt.xlabel('Stock (Ordered by Market Cap)')
plt.ylabel('Annualized Return Percentage (PCA)')
plt.title('Histogram of Annualized Return (PCA) Per Stock')
plt.axhline(0, color='black', linewidth=1)
plt.tight_layout()
plt.savefig('hist_pca.png')
plt.show()
```

```
plt.figure(figsize=(12, 6))
colors_JSE = ['green' if x >= 0 else 'orange' for x in weight_JSE]
plt.bar(stock_indices, weight_JSE, color=colors_JSE)
plt.xlabel('Stock (Ordered by Market Cap)')
plt.ylabel('Annualized Return Percentage (JSE)')
plt.title('Histogram of Annualized Return (JSE) Per Stock')
plt.axhline(0, color='black', linewidth=1)
plt.tight_layout()
plt.savefig('hist_jse.png')
plt.show()


annualized_stock_returns = expected_returns * 52
plt.figure(figsize=(10, 7))
plt.scatter(
    annualized_stock_std_dev,
    annualized_stock_returns,
    color='magenta', alpha=0.6, edgecolors='none',
    label='Individual Stocks (PCA)'
)
plt.scatter(
    annualized_stock_std_dev_JSE,
    annualized_stock_returns,
    color='teal', alpha=0.6, edgecolors='none',
    label='Individual Stocks (JSE)'
)
plt.scatter(
    annualized_std_dev,
    annualized_return,
    color='red', marker='*', s=250, edgecolors='black',
    label='PCA Min-Var Portfolio'
)
plt.scatter(
    annualized_std_dev_JSE,
    annualized_return_JSE,
    color='blue', marker='*', s=250, edgecolors='black',
    label='JSE Min-Var Portfolio'
)
text_PCA = (
    f"PCA:\n"
    f"Return = {annualized_return:.4f}\n"
    f"Variance = {annualized_var:.4f}"
```

```
)
plt.annotate(
    text_PCA,
    (annualized_std_dev, annualized_return),
    textcoords="offset points",
    xytext=(10, 40),
    ha='left',
    color='red',
    fontsize=10,
    arrowprops=dict(arrowstyle="-", color='red')
)
text_JSE = (
    f"JSE:\n"
    f"Return = {annualized_return_JSE:.4f}\n"
    f"Variance = {annualized_var_JSE:.4f}"
)
plt.annotate(
    text_JSE,
    (annualized_std_dev_JSE, annualized_return_JSE),
    textcoords="offset points",
    xytext=(80, -40),
    ha='left',
    color='blue',
    fontsize=10,
    arrowprops=dict(arrowstyle="-", color='blue')
)
plt.title('Risk Return Profile with Excess Returns')
plt.xlabel('Annualized Standard Deviation (Risk)')
plt.ylabel('Annualized Excess Return')
plt.grid(True, linestyle='--', alpha=0.5)
plt.legend()
plt.tight_layout()
plt.savefig('scatter.png')
plt.show()
```