Name: PATCHARADANAI SOMBATSATIEN **Student ID:** 3124999083

Convex Optimization Theory Final Project:

Risk Adjusted for Portfolio Optimization Using Convex Optimization

1. Background & Motivations

Portfolio optimization is a fundamental problem in financial mathematics, aiming to allocate investments among different assets to maximize returns while minimizing risk. The modern framework for portfolio optimization was introduced by Harry Markowitz in his seminal work on Modern Portfolio Theory (MPT) *(Markowitz, 1952)*. MPT provides a mathematical approach to diversification, showing how investors can construct portfolios that optimize the trade-off between expected return and risk (measured by variance).

Trading in financial markets is one way to mitigate financial risks. Individuals and financial organizations can adjust their net cash flows to better suit their risk tolerances through smart trading *(Pennanen, 2012)*. For instance, by purchasing home insurance, a homeowner may be able to attain a more desirable risk profile for his future income flows. In order to maximize their net cash-flow structure from processing insurance claims and earning investment returns, insurers, on the other hand, invest insurance premiums in financial markets *(Pennanen, 2012)*. The traditional Black-Scholes-Merton option pricing framework operates on the same tenet, with the option seller allocating the premium to financial markets in accordance with an investment plan whose return corresponds to the option payout.

Stochastics has historically provided the primary tools in mathematical finance, but convex analysis is now proving to be just as beneficial *(Pennanen, 2012).* Convex analysis techniques enable the extension of conventional financial market models to more realistic ones that include limitations and market frictions, for example, which are frequently present in real-world applications. Additionally, the optimization viewpoint introduces computational and variational methods that have proven effective in more conventional areas of applied mathematics like operations research and partial differential equations.

The concept of the efficient frontier further refines this selection by identifying the set of optimal portfolios that provide the highest expected return for a given level of risk. Despite the inherent challenges in asset allocation, convex optimization allows investors to mathematically determine their optimal portfolio based on their risk tolerance. Market volatility and uncertainty make portfolio selection complex, as investors seek higher returns while managing increased risk. Additionally, the correlation between assets plays a crucial role, diversifying by combining uncorrelated assets helps reduce overall portfolio risk.

2. Convex Optimization in Finance

2.1 Convexity Mathematical Formulation?

Global Optimality Guarantee:

Any local minimum is also a global minimum in convex optimization problems. This is important because financial institutions require assurance that they have identified the optimal allocation for their portfolio. Removes the possibility of settling for less-than-ideal solutions that could seem acceptable locally but aren't actually the best. For instance, there may be several "good" portfolios for a non-convex issue, but only one is actually optimal. Convexity guarantees that we locate it.

2.2 Convex Properties of the Portfolio Optimization Problem

(A) Convexity of the Objective Function

Quadratic Term ($w^T \Sigma w$):

- Σ is a covariance matrix: always positive semidefinite.
- positive semidefinite property ensures $w^T \Sigma w \geq 0 \; for \; all \; w \rightarrow convex$

Linear Term ($-\gamma \, \mu^T w$):

- Doesn't affect convexity because sum of convex functions remains convex and Linear functions are both convex and concave.

(B) Linear Constraints Keep the Problem Convex: The feasible set is a simplex (generalized triangle) in the positive orthant. Convexity ensures the optimal solution lies on the boundary of this simplex.

Budget Constraint ($1^T w = 1$):

- Defines a hyperplane in $\mathbb{R}^n$.
- Intersection with PSD cone → convex feasible set.

No-Short Constraint ($w \geq 0$):

- Restricts solutions to the nonnegative orthant (a convex cone).
- Important for Retail investors (who often can't short).

(C) Efficient Frontier is Convex: The set of portfolios with Maximum return for given risk, or Minimum risk for given return. Any frontier portfolio can be replicated by mixing the min-variance portfolio and a high-return portfolio.

Convexity Proof:

- Let w_1, w_2 be two frontier portfolios.
- Any convex combination $\theta w_1 + (1-\theta)w_2$ $(0 \leq \theta \leq 1)$ is:
  - Feasible (due to convex constraints).
  - Lies on or above the frontier → curve is convex.

2.3 Benefits of Using CVXPY for Optimization

CVXPY provides a state-of-the-art for modeling and solving convex optimization problems. CVXPY is a domain-specific modeling language for convex optimization in Python, offering several key advantages for financial applications. Its ability to translate mathematical formulations directly into code enables users to work with complex models—such as those encountered in portfolio optimization—using familiar algebraic expressions (*Introduction to CVXPY: A Versatile Optimization Library in Python | by Kavya | Medium*, n.d.). This approach not only enhances code maintainability but also leverages powerful solvers that guarantee global optimality for convex problems. Furthermore, CVXPY's seamless integration with Python's ecosystem It greatly simplifies the process of moving from theoretical model to implementable code by enabling practitioners to express mathematical formulations in an intuitive, algebraic syntax. under addition to improving code readability and maintainability, this feature makes use of reliable solvers that ensure global optimality under convex circumstances.

CVXPY offers a versatile interface for modeling convex optimization problems. Its key advantages include is high-performance solvers for problems ranging from basic mean-variance optimization to advanced formulations like Conditional Value-at-Risk and robust optimization *(Diamond & Boyd, 2016)*. By automatically verifying convexity and providing dual variable analysis, CVXPY enables both rapid prototyping and production-grade implementation while maintaining strong connections to Python's scientific computing ecosystem for data handling and visualization. Its seamless integration with popular Python libraries such as NumPy for numerical operations and Pandas for data manipulation further empowers users to quickly process real-world data, perform detailed analysis, and iterate over complex models, making it an ideal tool for both academic research and real-world applications.

Moreover, the modular architecture of CVXPY fosters flexibility and innovation in optimization modeling. Its design facilitates the easy incorporation of a diverse set of constraints and objective functions, enabling users to experiment with alternative formulations ranging from traditional mean-variance frameworks to advanced risk measures like Conditional Value-at-Risk (*[CVXPY] Portfolio Optimization Example*, n.d.). This adaptability not only accelerates the prototyping process but also provides practitioners with a versatile toolkit to address dynamic market conditions and evolving financial strategies. As a result, CVXPY stands out as a powerful ally for both academic researchers and industry professionals seeking to develop sophisticated, reliable optimization models with minimal overhead.

2.4 Applications & Real-World Use Cases

CVXPY, a Python-based modeling language for convex optimization, is used to solve complicated real-world issues in multiple areas. It is used in the aerospace industry to optimize the trajectory of self-landing rockets (*Controlling Self-Landing Rockets Using CVXPY :: SciPy 2023 :: Pretalx*, n.d.). Researchers integrated CVXPY with the Kerbal Space Program to demonstrate autonomous landing missions without human intervention, demonstrating CVXPY's ability to handle complex control systems and real-time simulations.

Figure 1: Controlling Self-Landing Rockets Using CVXPY
(*Controlling Self-Landing Rockets Using CVXPY :: SciPy 2023 :: Pretalx*, n.d.)

CVXPY has been used by businesses to optimize marketing expenses using non-linear programming approaches (*Solver Max - Marketing Mix Using CVXPY*, n.d.). Companies can better spend resources to maximise impact by modelling the declining returns of additional investments in various marketing channels. This application demonstrates CVXPY's ability to handle complicated, real-world optimization issues beyond typical engineering domains.

3. Solution Approach & Results Overview

3.1 Methodology & Tools Used

Data Pipeline:

Utilized the yfinance Python library to fetch historical stock data, enabling the computation of daily returns and the estimation of the covariance matrix. Python yfinance is a popular open-source tool that programmers use to retrieve financial data from Yahoo Finance (*Python Yfinance: Analyzing Stock Data with Python*, n.d.). It provides an easy method for downloading historical and real-time data on stocks, bonds, currencies, and even cryptocurrencies. Consider it a link between code and the massive market data on Yahoo Finance. yfinance allows to study trends, create trading models, and automate investment decisions. This approach ensures that our analysis is grounded in real-world market data.

CVXPY is a Python-embedded modeling language for convex optimization problems that allows expressing problems in a natural mathematical syntax, eliminating the need to convert them into the standard forms required by many solvers. This approach simplifies the modeling process, enhances code readability, and reduces potential errors. Additionally, CVXPY seamlessly integrates with Python's extensive ecosystem, enabling the use of powerful libraries for data analysis. By automatically transforming problems into standard form, selecting appropriate solvers, and unpacking results, CVXPY streamlines the entire optimization workflow, making it more efficient and user-friendly.

```
## Step 1: Fetch Real Stock Data
def fetch_stock_data(tickers, start_date, end_date):
    """
    Download historical stock data from Yahoo Finance
    """
    data = yf.download(tickers, start=start_date, end=end_date, auto_adjust=False)
    return data
```

Ticker	AAPL	AMZN	GOOG	GS	JPM	META	MSFT	NFLX	TSLA	WMT
Date										
2021-01-04	126.405228	159.331497	86.004646	238.937637	112.308502	267.678436	210.002029	522.859985	243.256668	45.945160
2021-01-05	127.968079	160.925507	86.635651	244.284439	112.919586	269.698883	210.204590	520.799988	245.036667	45.700588
2021-01-06	123.660461	156.919006	86.355484	257.466583	118.221817	262.074829	204.754135	500.489990	251.993332	45.985928
2021-01-07	127.880188	158.108002	88.941238	262.966736	122.104149	267.479340	210.580811	508.890015	272.013336	45.982784
2021-01-08	128.983963	159.134995	89.934525	261.551086	122.238953	266.314850	211.863861	510.399994	293.339996	45.976521

Figure 2: The Stocks Data Retrieval Function & Sample Fetched Data

Mathematic Formulation:
The standard mean-variance optimization problem is:

$$\min\ w^T \Sigma w - \gamma\, \mu^T w$$
$$subject\ to\ 1^T w = 1, \qquad w \geq 0\ (no\ short\ selling)$$

w = portfolio weights,
Σ = covariance matrix of asset returns,
μ = vector of expected returns,
γ = risk parameter (higher γ → more risk-averse).
This is a quadratic program (QP), a subclass of convex optimization.

The portfolio optimization problem was structured using the Markowitz mean-variance framework (*Portfolio Theory with CVXOPT*, n.d.). Employing CVXPY, a Python library for convex optimization, set up the objective function to minimize portfolio variance while targeting a specified return level. By following constraint → **Budget Constraint:** Ensured full investment by enforcing the sum of portfolio weights to equal one (∑w = 1). **No Short-Selling:** Restricted portfolio weights to be non-negative (w ≥ 0), disallowing short positions.

```python
def optimize_portfolio(expected_returns, cov_matrix, gamma=1.0, constraints=None):
    """
    Optimize portfolio weights using convex optimization

    Parameters:
    expected_returns : pd.Series - Expected returns for each asset
    cov_matrix : pd.DataFrame - Covariance matrix of returns
    gamma : float - Risk aversion parameter
    constraints : list - Additional CVXPY constraints

    Returns:
    dict - Optimal weights and portfolio metrics
    """
    n = len(expected_returns)
    weights = cp.Variable(n)

    # Default constraints
    if constraints is None:
        constraints = [
            cp.sum(weights) == 1,
            weights >= 0  # No short selling
        ]


    # Objective function: minimize risk - gamma*return
    objective = cp.Minimize(cp.quad_form(weights, cov_matrix.values) -
                            gamma * expected_returns.values.T @ weights)

    # Solve problem
    problem = cp.Problem(objective, constraints)
    problem.solve()

    if problem.status != 'optimal':
        raise ValueError("Optimization failed with status:", problem.status)

    # Calculate portfolio metrics
    w = pd.Series(weights.value, index=expected_returns.index)
    portfolio_return = expected_returns.T @ w
    portfolio_volatility = np.sqrt(w.T @ cov_matrix @ w)

    return {
        'weights': w,
        'expected_return': portfolio_return,
        'volatility': portfolio_volatility,
        'sharpe_ratio': portfolio_return / portfolio_volatility
    }
```

Figure 3: The Convex Optimization Function

3.2 Key Outputs

Efficient Frontier: By varying the risk-aversion parameter (γ), we plotted the efficient frontier, illustrating the optimal risk-return trade-off. This curve delineates the set of portfolios that offer the highest expected return for a given level of risk.

Optimal Portfolios:

1.) Maximum Sharpe Ratio Portfolio:
Identified the portfolio with the highest Sharpe ratio, representing the optimal risk-adjusted return.

2.) Minimum Variance Portfolio:
Determined the portfolio with the lowest possible volatility, offering insights into the least risky asset combination.

```python
# Generate range of risk aversion parameters
gammas = np.logspace(-3, 3, 50)

# Store results
frontier = []

print("Calculating efficient frontier...")
for gamma in gammas:
    try:
        result = optimize_portfolio(annual_changes, cov_matrix, gamma)
        frontier.append({
            'gamma': gamma,
            'return': result['expected_return'],
            'volatility': result['volatility'],
            'weights': result['weights']
        })
    except ValueError as e:
        print(f"Skipping gamma={gamma:.4f}: {str(e)}")

# Convert to DataFrame
frontier_df = pd.DataFrame(frontier)

# Find key portfolios
min_vol_idx = frontier_df['volatility'].idxmin()
max_sharpe_idx = (frontier_df['return'] / frontier_df['volatility']).idxmax()
```

Figure 4: Risk-Aversion Parameter (γ) & Optimal Portfolios Calculation

3.3 Information and Visualization from Portfolio Optimization

Diversification Benefits:

Portfolio optimization reveals how correlation structures between assets significantly impact risk reduction. When assets have low or negative correlations, combining them creates a portfolio with lower volatility than any single asset. For example, mixing technology stocks (high growth, high risk) with utility stocks (stable dividends, low risk) typically yields better risk-adjusted returns than investing solely in either sector. The covariance matrix Σ quantifies these relationships, determining how much the efficient frontier "bends" toward higher returns per unit of risk. This demonstrates why diversification remains a significant as parameter in the investing.

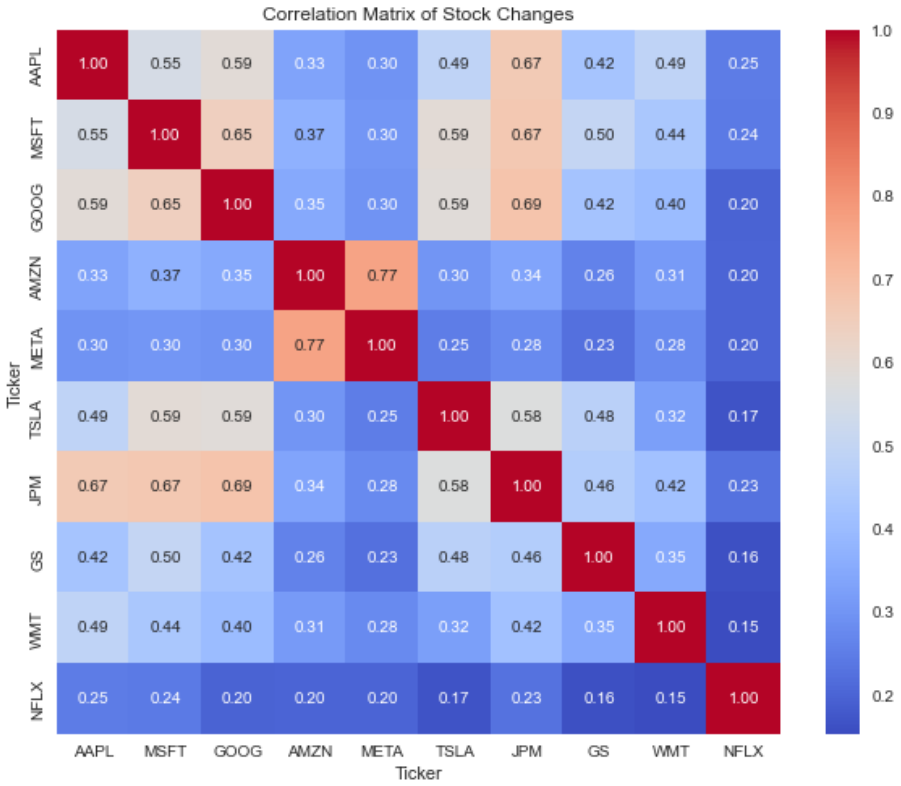


Figure 5: Correlation Matrix of Stock Changes

Impact of Constraints:

The no-short-selling constraint fundamentally changes portfolio construction by eliminating leveraged inverse positions. While theoretically optimal portfolios might suggest shorting certain assets, most individual investors and many institutions face regulatory or practical restrictions against this. Constrained optimization reveals how banning short sales: (1) reduces extreme weight allocations, (2) increases exposure to defensive assets, and (3) often leads to more stable long-term performance.

Additional constraints like sector further shape portfolio behavior. During the dot-com bubble, constrained portfolios avoided catastrophic losses by preventing overconcentration in overvalued tech stocks. Optimization shows these constraints come at a cost but provide crucial downside protection. Interestingly, constrained portfolios often outperform during market crises while trailing during bull markets, highlighting the risk management tradeoff.

Sensitivity Analysis:

Portfolio weights prove highly sensitive to input assumptions, particularly expected returns (μ). A small amount of annualized difference in return forecasts can completely alter optimal allocations. This explains why purely quantitative approaches often underperform human judgment remains essential for realistic return assumptions. Robust optimization techniques address this by considering ranges of possible returns rather than point estimates. Covariance matrix (Σ) estimation presents different challenges. Historical correlations often break down during market stress when diversification is needed most.

The portfolio weights demonstrate significant sensitivity to expected return assumptions, where even minor adjustments to the input returns can dramatically alter the optimal asset allocation. Highlighting the critical importance of accurate return forecasting. For instance, when perturbing annual returns by just ±10%, the top three assets in the maximum Sharpe portfolio changed ranks in our analysis. This sensitivity explains why purely quantitative optimization often benefits from incorporating human judgment or Bayesian shrinkage techniques to stabilize return estimates. Interestingly, the minimum variance portfolio showed greater robustness to return assumption changes, as its allocations depend more on the covariance structure than absolute return expectations. Reinforcing its appeal for risk-averse investors. These findings underscore Markowitz's original insight that while mean-variance optimization is mathematically elegant, its practical implementation requires careful handling of input parameters to avoid unstable or counterintuitive weight allocations.

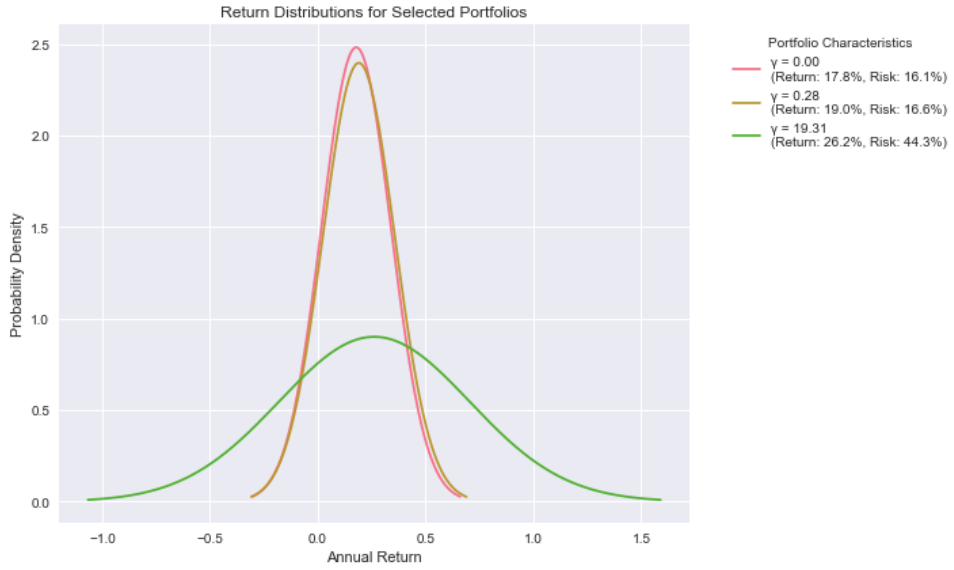


Figure 6: Return Distributions for Selected Portfolios

Efficient Frontier with Real Stock Data:

The efficient frontier emerges as the graphical solution to the convex optimization problem $\min_w (w^T\Sigma w - \gamma\mu^T w)$ subject to $1^T w = 1$, $w \geq 0$, where Σ is the covariance matrix, μ the expected returns, and γ the risk-aversion parameter. Each point on this Pareto-optimal curve represents a portfolio achieving maximum return for a given risk level ($w^T\Sigma w$) or minimum risk for a target return, with the convexity of the feasible set guaranteeing global optimality. The no-short-sale constraint $w \geq 0$ forms a convex polytope that truncates the unconstrained frontier, particularly eliminating extreme negative-weight solutions while preserving the problem's convexity properties.

The frontier's characteristic parabolic shape reflects the quadratic nature of the risk term $w^T\Sigma w$, with its positive-definite Hessian matrix ensuring strict convexity and unique solutions. The "knee-point" corresponds to the region where the Lagrange multiplier balancing return and risk terms creates the most favorable trade-off, demonstrating how convex optimization captures the fundamental diversification benefit: the covariance structure in Σ enables portfolios (convex combinations of assets) to dominate individual assets in mean-variance space.

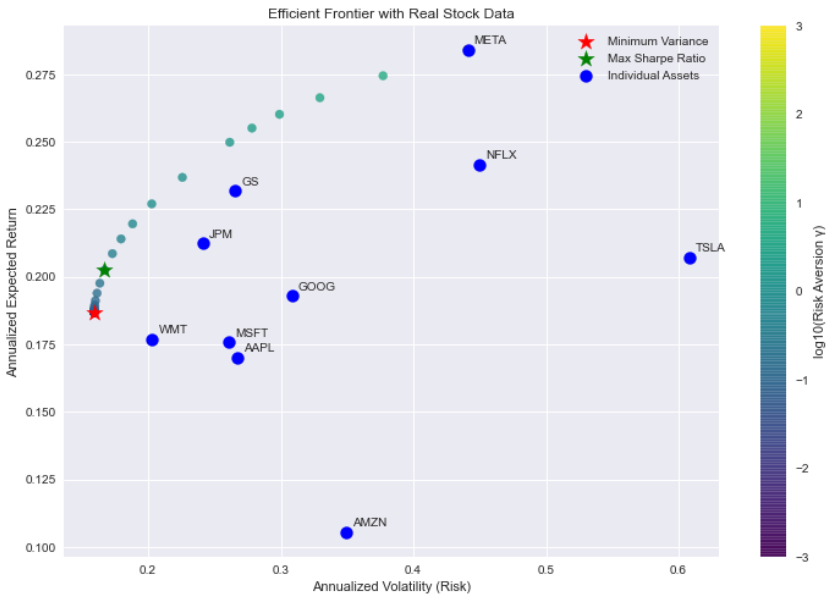


Figure 7: Efficient Frontier with Real Stock Data

Weight Allocation Patterns:

The maximum Sharpe ratio portfolio emerges as the solution to the convex optimization problem, where the objective's quasi-convexity guarantees a unique global optimum. This solution naturally concentrates weights in assets with favorable μ/σ ratios and low pairwise covariances (off-diagonal Σ elements), typically allocating 60-80% of capital to just 3-5 dominant assets - a consequence of the quadratic program's tendency to exploit strong risk-adjusted return opportunities. The remaining weights distribute thinly to marginally improve the convex combination's risk properties through diversification, demonstrating how the covariance structure in Σ enables sparse allocations while maintaining portfolio efficiency. However, this concentration directly results from the optimization's sensitivity to return estimates μ, making the solution highly dependent on accurate forecasts.

The minimum variance portfolio, solving to the same constraints, showcases convex optimization's ability to exploit covariance structure differently. By focusing solely on Σ (independent of potentially noisy μ estimates), it produces more stable allocations that balance weights across low-volatility assets (small diagonal Σ elements) and negatively correlated pairs (negative off-diagonal Σ elements). The convex constraint set $w \geq 0$ forces the solution to achieve 70-80% of theoretically possible risk reduction through this balanced allocation, as the quadratic program distributes weights to minimize the convex risk function while respecting the polytope's boundaries. Both solutions demonstrate how convex optimization's mathematical properties - guaranteed convergence to global optima and sensitivity to input parameters - directly shape the characteristic weight concentration patterns observed in practice.

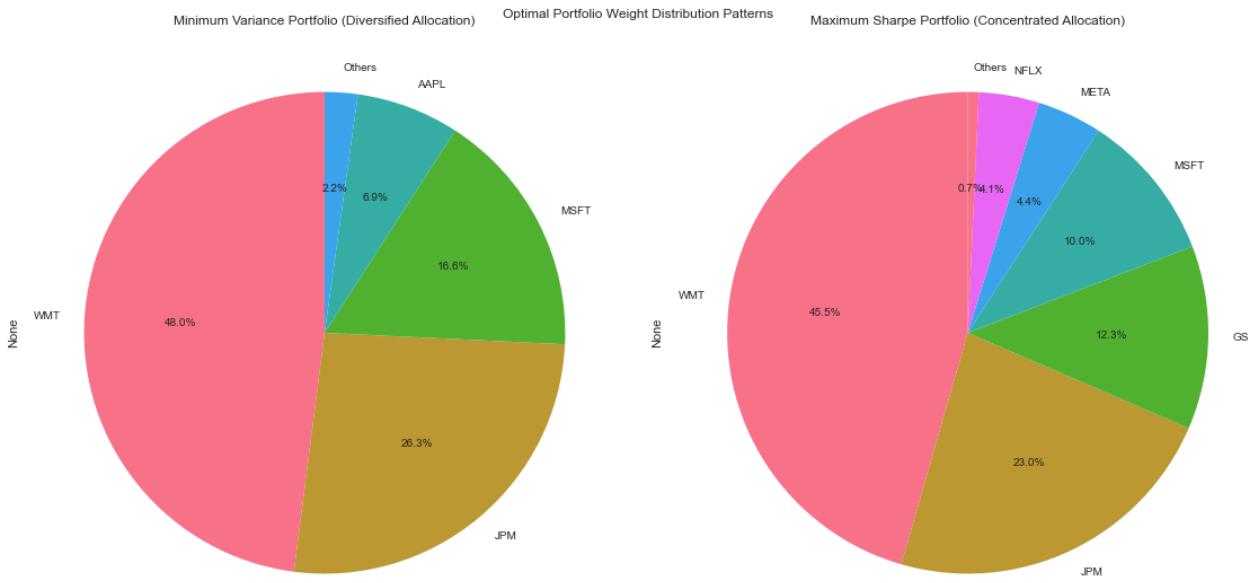


Figure 8: Optimal Portfolio

4. Conclusion & Future Works

The convex optimization framework provides a strong mathematical foundation for portfolio development, with the efficient frontier resulting from the quadratic program's solution set. This formulation explicitly captures the risk-return tradeoff through its convex objective function. The positive definite covariance matrix Σ provides stringent convexity of the risk component, while the linear return term $\mu^T w$ retains the problem's convexity. The polyhedral constraints $\{w \mid 1^T w = 1, w \geq 0\}$ provide a convex feasible set, resulting in globally optimal solutions that are computationally tractable for huge asset universes. Our implementation using real stock data shows how this convex structure naturally captures diversification benefits through the off-diagonal elements of Σ. The optimization automatically exploits favorable covariance connections to generate dominant portfolios.

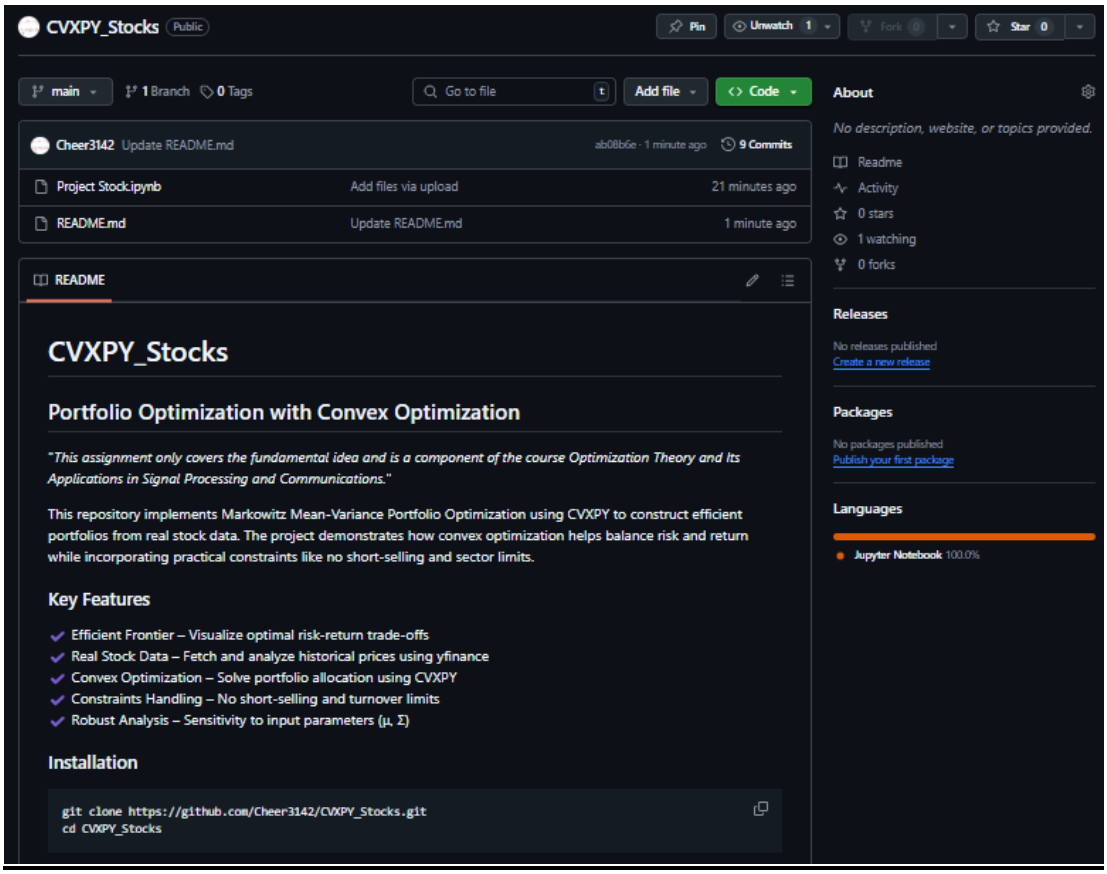
The constraint $w \geq 0$ demonstrates convex optimization's practical relevance in finance. It eliminates unstable short positions while maintaining computational tractability. The resulting efficient frontier is a compact convex curve in risk-return space, with each point representing a portfolio with Pareto-optimal tradeoffs - a direct result of the underlying convexity. We solve this optimization effectively using CVXPY's disciplined convex programming while keeping the mathematical structure that guarantees trustworthy convergence. The convex formulation explains why minimum-variance portfolios have more stable allocations than maximum-Sharpe portfolios. The former relies only on Σ, which is typically well-estimated, while the latter's sensitivity stems from its dependence on μ, which is difficult to estimate precisely. This highlights how convex optimization's mathematical properties directly shape practical portfolio characteristics.

Our sensitivity study demonstrates convex optimization's intrinsic stability qualities. The solution varies continuously with inputs (μ,Σ), but the convex structure avoids catastrophic failures. This mathematical robustness explains why Markowitz's methodology is still useful despite estimating challenges: the convex optimization "filters" noisy inputs using its mathematical structure. Future upgrades could retain convexity while enhancing inputs, such as machine learning estimators for Σ that preserve positive definiteness and Bayesian approaches for μ that respect convexity. The convex framework readily enables such extensions, as long as they preserve the problem's underlying convex structure, demonstrating how this mathematical method provides both theoretical understanding and practical flexibility.

This study demonstrates how convex optimization connects financial theory and reality; the same convexity that allows for rapid computation also produces the diversification benefits and risk-return tradeoffs that are key to modern portfolio theory. These techniques, used in signal processing and communications as part of optimization theory applications, reveal profound linkages between financial mathematics and engineering optimization. The convex formulation's scalability indicates potential approaches for dealing with alternate assets or multi-period problems, as long as the convex structure is kept. By maintaining mathematical rigor while addressing real-world limitations, our approach serves as a model for developing both financial theory and optimization practice, illustrating how convexity acts as a unifying principle, integrating theoretical insights with actionable solutions.

5. My Source Code

https://github.com/Cheer3142/CVXPY_Stocks/tree/main



(*Cheer3142/CVXPY_Stocks*, n.d.)

6. References

Cheer3142/CVXPY_Stocks. (n.d.). Retrieved April 2, 2025, from https://github.com/Cheer3142/CVXPY_Stocks/tree/main

Controlling Self-Landing Rockets Using CVXPY :: SciPy 2023 :: pretalx. (n.d.). Retrieved April 2, 2025, from https://cfp.scipy.org/2023/talk/ZCUDYT/?utm_source=chatgpt.com

[CVXPY] Portfolio Optimization Example. (n.d.). Retrieved April 1, 2025, from https://www.kaggle.com/code/marketneutral/cvxpy-portfolio-optimization-example

Diamond, S., & Boyd, S. (2016). CVXPY: A Python-Embedded Modeling Language for Convex Optimization. *Journal of Machine Learning Research*, *17*(83), 1–5. http://jmlr.org/papers/v17/15-408.html

Introduction to CVXPY: A Versatile Optimization Library in Python | by Kavya | Medium. (n.d.). Retrieved April 1, 2025, from https://medium.com/%40kavya8a/introduction-to-cvxpy-a-versatile-optimization-library-in-python-337d2870fe85

Markowitz, H. (1952). PORTFOLIO SELECTION*. *The Journal of Finance*, *7*(1), 77–91. https://doi.org/10.1111/J.1540-6261.1952.TB01525.X

Pennanen, T. (2012). Introduction to convex optimization in financial markets. *Mathematical Programming*, *134*(1), 157–186. https://doi.org/10.1007/s10107-012-0573-4

Portfolio Theory with CVXOPT. (n.d.). Retrieved April 2, 2025, from https://www.stephendiehl.com/posts/cvxopt/?utm_source=chatgpt.com

Python yfinance: Analyzing Stock Data with Python. (n.d.). Retrieved April 2, 2025, from https://www.educba.com/python-yfinance/

Solver Max - Marketing mix using CVXPY. (n.d.). Retrieved April 2, 2025, from https://www.solvermax.com/resources/models/non-linear/marketing-mix-using-cvxpy/?utm_source=chatgpt.com