# Risk-Adjusted Portfolio Optimization Using Convex Optimization

**Name:** PATCHARADANAI SOMBATSATIEN

**ID**：3124999083

**Course:** Optimization Theory and Its Applications in Signal Processing and Communications

# Contents

西安交通大學
XI'AN JIAOTONG UNIVERSITY

# 01

# Background & Motivations

1 PART

# Introduction to Portfolio Optimization

Portfolio optimization addresses how to distribute investments across various assets to achieve the highest possible returns for a given level of risk.

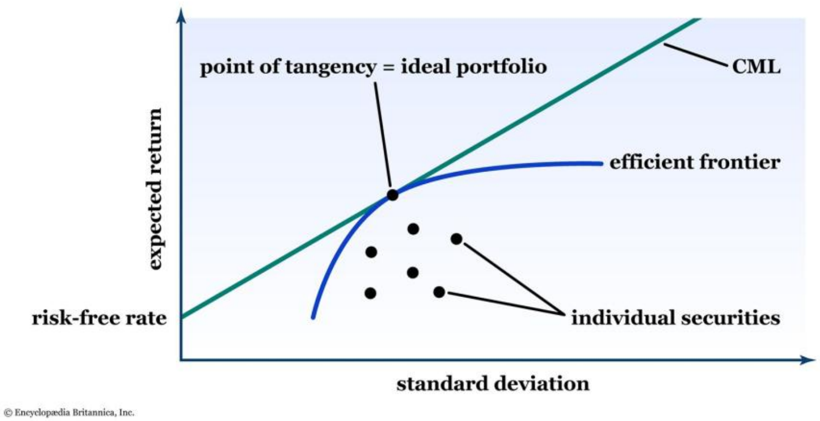
In 1952 Harry Markowitz’s Modern Portfolio Theory (MPT) introduced a mathematical framework for this task, demonstrating how diversification enables investors to balance expected gains against the variability of returns.

*Key Concept of MPT:*

i. MPT is the framework to maximize their overall returns within an acceptable level of risk

ii. MPT can be useful to investors trying to construct efficient and diversified portfolios

Capital market line (CML) and the efficient frontier

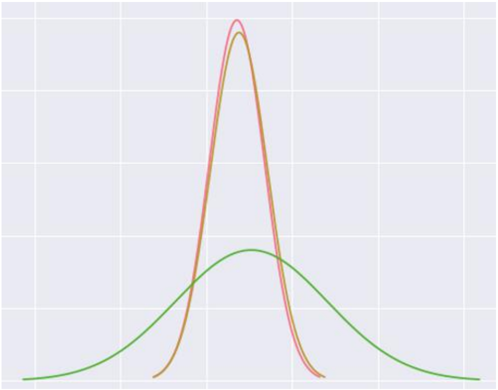



*from: https://www.britannica.com/money/modern-portfolio-theory-explained*

1
PART

# Why Convex Optimization in Finance?

- ❖ Convexity ensures any local minimum is the global minimum, guaranteeing the optimal portfolio.
- ❖ Robust to nonlinearities and non-smoothness via strong duality.
- ❖ Efficient algorithms enable scalable solutions.



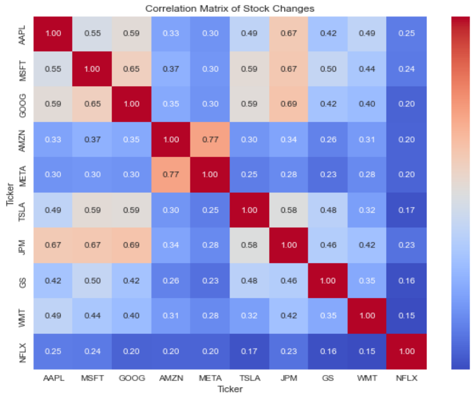
*Key Concept of project:*

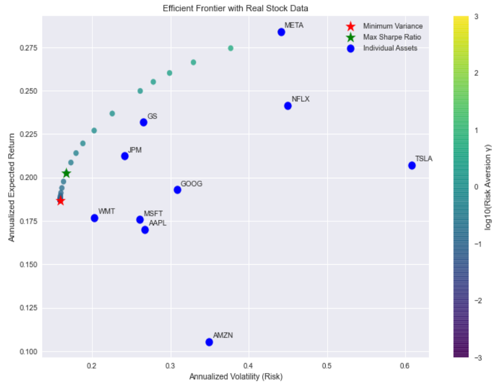
i. Formulate portfolio selection as convex quadratic program

ii. Maximize their overall returns within an acceptable level of risk

iii. Incorporates asset correlations for optimal diversification

iv. Visualize and extract the information from stocks data



# Mathematical Formulation

The portfolio optimization using the Markowitz mean-variance model, which relies on simplified, static estimates of μ and Σ and incorporates no-short-selling constraints that, although practically necessary, can restrict theoretical optimality.

**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,
$\mu$ = vector of expected returns,
γ = risk parameter (higher γ → more risk-averse).
This is a quadratic program (QP)

# Convexity Properties of the Objective Function

**Quadratic Term ($\mathbf{w}^T\Sigma\mathbf{w}$):**

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

**Linear Term ($-\gamma\ \boldsymbol{\mu}^T\mathbf{w}$):**

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

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

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

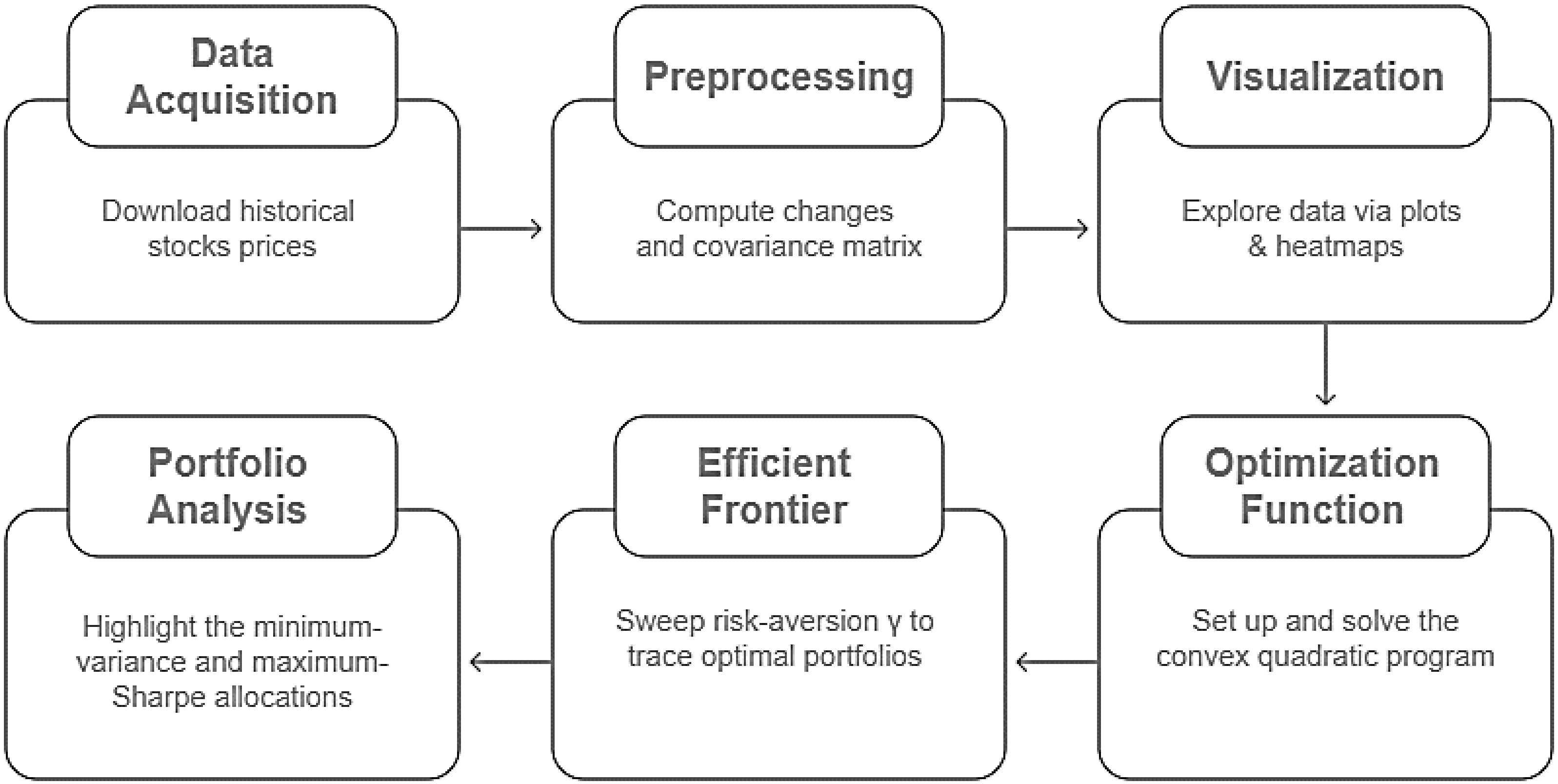
**No-Short Constraint ($w \geq 0$):**

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

西安交通大學
XI'AN JIAOTONG UNIVERSITY

# 02 Implementation

# Implementation Pipeline

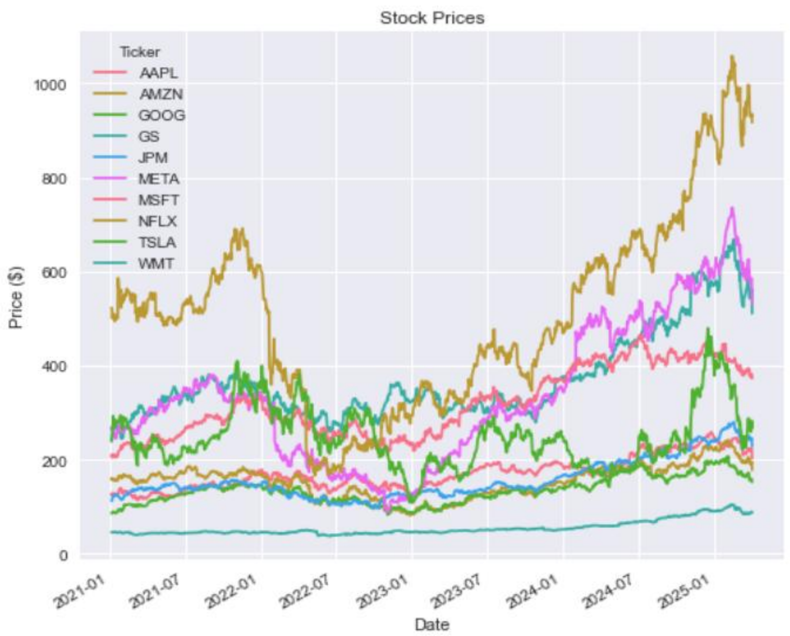


2 PART

# Data Acquisition

We utilized the ***yfinance*** Python library to fetch historical stock data, enabling the calculation of daily returns and the estimation of the covariance matrix. As a widely used open-source tool, yfinance provides seamless access to historical and real-time data on stocks, bonds, currencies, and cryptocurrencies from Yahoo Finance.
It acts as a bridge between Python code and large-scale market data, allowing users to analyze trends, build trading models, and automate investment strategies.
This ensures that our analysis is firmly rooted in real-world financial data.

```python
## 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
```



2 PART

# Optimization Function

The portfolio optimization problem was formulated using the Markowitz mean-variance framework, assuming static estimates of expected returns (μ) and the covariance matrix (Σ) a simplifying assumption that, while limiting theoretical optimality, is practical in real-world applications. Constraints such as no short-selling, which restricts portfolio weights to be non-negative (w ≥ 0), and a budget constraint ensuring full investment (weights summing to one), were incorporated to reflect realistic trading conditions.

Using CVXPY, a Python library for convex optimization, the objective was set to minimize portfolio variance while achieving a specified target return, enabling a clear and efficient implementation of the model.

```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
    }
```

2 PART

# Efficient Frontier

To visualize the trade-off between risk and return, we generated the efficient frontier by varying the risk-aversion parameter (γ) within the Markowitz mean-variance framework. Each point on the frontier represents an optimized portfolio that achieves the maximum expected return for a specific level of risk.

This curve provides valuable insights into the relationship between portfolio volatility and expected performance, helping investors make informed decisions based on their individual risk tolerance. By plotting the frontier, we highlight the boundary of achievable portfolios, distinguishing optimal asset allocations from suboptimal ones.

```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()
```

# 2 PART Portfolio Analysis

Within the efficient frontier, two key portfolios were identified for further analysis. The Maximum Sharpe Ratio Portfolio represents the portfolio with the highest risk-adjusted return, maximizing the Sharpe ratio by balancing excess return against volatility. This portfolio is ideal for investors seeking the most efficient use of risk.

In contrast, the Minimum Variance Portfolio focuses purely on risk reduction, identifying the asset mix with the lowest possible volatility. It offers a conservative investment strategy suited for risk-averse individuals, emphasizing stability over return maximization. Together, these portfolios exemplify distinct investment priorities within the efficient frontier framework.

```python
# Prepare treemap data
def prepare_treemap_data(weights, threshold=0.03):
    w = weights.copy()
    w[w < threshold] = 0  # Group small weights
    w = w[w > 0]
    others = 1 - w.sum()
    if others > 0:
        w['Others'] = others
    return w.sort_values(ascending=False)

min_vol_treemap = prepare_treemap_data(frontier_df.loc[min_vol_idx, 'weights'])
max_sharpe_treemap = prepare_treemap_data(frontier_df.loc[max_sharpe_idx, 'weights'])

# Plot treemaps
fig, (ax1, ax2) = plt.subplots(1, 2, figsize=(16,8))

min_vol_treemap.plot(kind='pie', autopct='%1.1f%%',
                    ax=ax1, startangle=90)
ax1.set_title('Minimum Variance Portfolio (Diversified Allocation)')

max_sharpe_treemap.plot(kind='pie', autopct='%1.1f%%',
                       ax=ax2, startangle=90)
ax2.set_title('Maximum Sharpe Portfolio (Concentrated Allocation)')

plt.suptitle('Optimal Portfolio Weight Distribution Patterns')
plt.tight_layout()
plt.show()
```

西安交通大學
XI'AN JIAOTONG UNIVERSITY

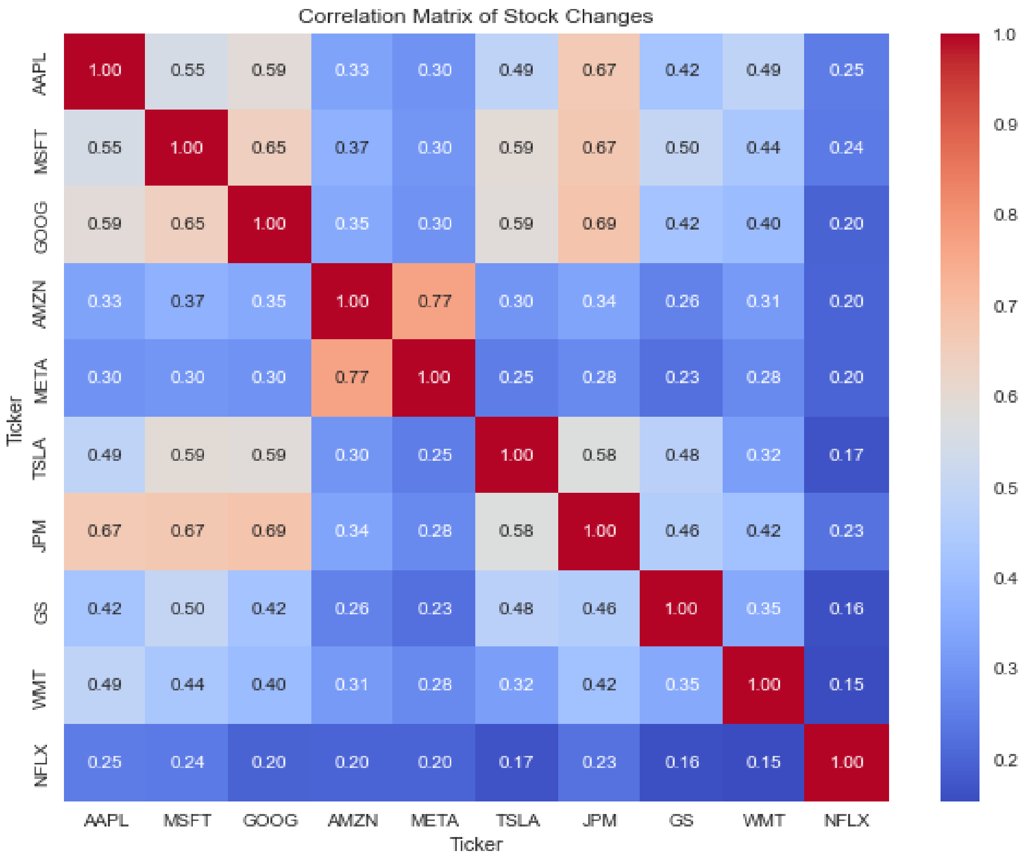
# Implemented Result & Analysis

# 03

3 PART

# The covariance matrix Σ

**Asset correlations drive risk reduction:**
low or negative correlations let portfolios achieve lower volatility than any single holding. The covariance matrix Σ encodes these relationships, bending the efficient frontier and underscoring diversification's power in boosting risk-adjusted returns.

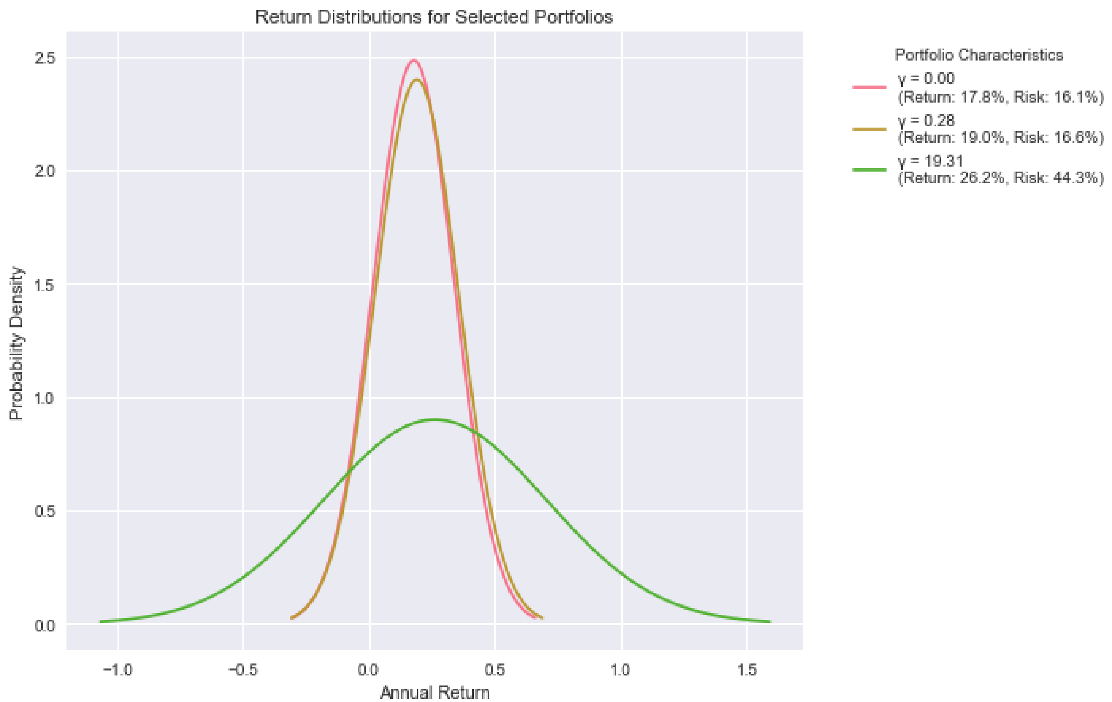


3 PART

# Sensitivity Analysis

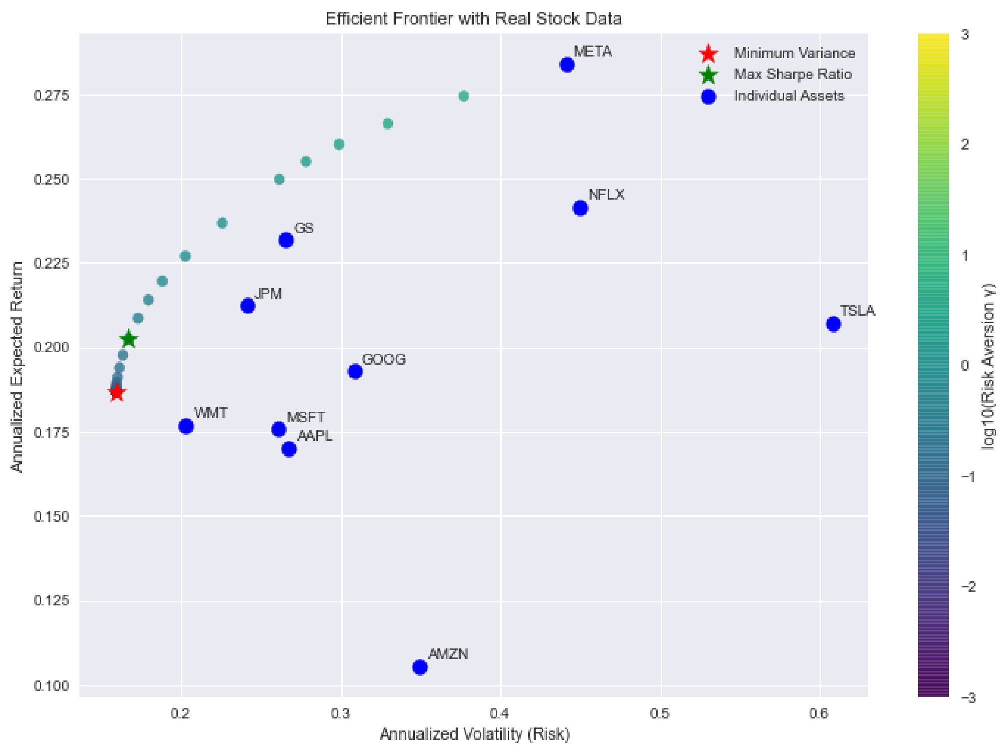
Portfolio weights are extremely sensitive to expected. Return inputs can reshape the maximum-Sharpe portfolio's top holdings to stabilize estimates. In contrast, the minimum-variance portfolio, driven primarily by Σ, is more resilient to return misestimation, making it attractive for risk-averse investors.

These results reaffirm that, despite its elegance, mean-variance optimization demands careful input handling to prevent unstable or counterintuitive allocations.



# Efficient Frontier with Real Stock Data

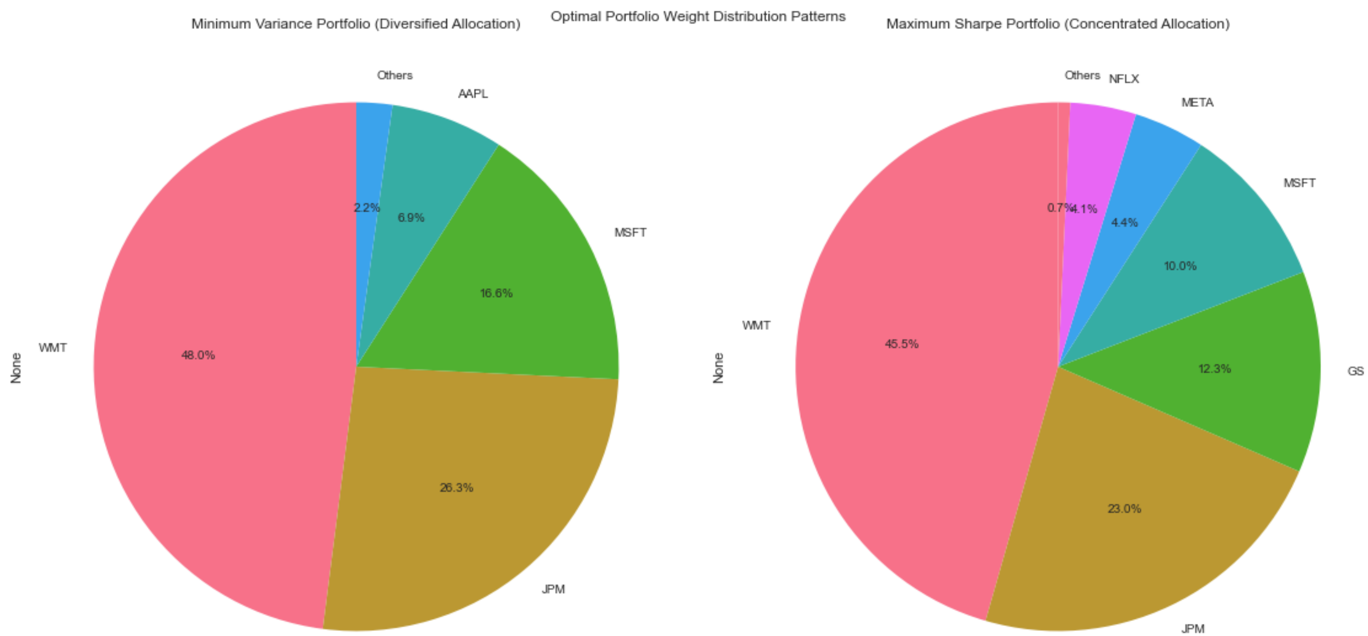
The efficient frontier is the Pareto-optimal locus of portfolios solving the convex program. Convexity of the feasible set (a no-short-sales polytope) and positive–definiteness of Σ guarantee a unique global solution at each $\gamma$. Its characteristic parabola arises from the quadratic risk term, and the “knee” pinpoints the optimal trade-off between risk and return, illustrating how diversification via asset combinations outperforms single-asset investments.



3 PART

# Weight Allocation Patterns

The maximum-Sharpe portfolio solves a convex program that concentrates 60–80% of capital in 3–5 assets with high μ/σ ratios and low covariances while thinly allocating the remainder to enhance diversification; its heavy concentration, however, hinges on precise return estimates. In contrast, the minimum-variance portfolio, which ignores μ and focuses solely on Σ, yields more balanced, stable weights across low-volatility and negatively correlated assets, achieving 70–80% of the possible risk reduction under the same no-short-sale constraints.



西安交通大學
XI'AN JIAOTONG UNIVERSITY

# 04

# Key Achievements

4 PART

# Key Achievements

i. Framework Implementation: Translated Markowitz mean-variance model into a convex QP using CVXPY, guaranteeing global optimal, and Established a Python pipeline (Data Acquisition → Preprocessing → CVXPY → Analysis).

ii. Efficient Frontier Construction: Computed and plotted the Pareto-optimal curve over real stock data, illustrating risk–return trade-offs.

iii. Optimal Portfolio Identification: Derived both maximum-Sharpe and minimum-variance portfolios, demonstrating distinct concentration.

iv. Sensitivity & Robustness Analysis: Quantified the impact of $\mu$ and $\Sigma$ perturbations (±10% return shocks), highlighting the need for robust or Bayesian techniques.

v. Covariance Insights: Showcased how $\Sigma$'s structure bends the frontier and drives diversification benefits across asset mixes.

vi. Practical Constraints Handling: Modeled no-short-sale and budget constraints, reflecting realistic investment limits and regulatory requirements.

# Thank you