**Portfolio Optimization Analysis and Algorithm Explanation**

# Introduction

This document provides an overview of the algorithms and methodologies used in the portfolio optimization script. The script fetches historical data for specified assets, calculates key portfolio metrics, and optimizes the portfolio to maximize the Sharpe Ratio.

# 1. Historical Data Analysis

The script fetches historical price data for the specified assets and calculates daily returns. The daily returns are then used to calculate annualized portfolio metrics.

# 2. Portfolio Metrics Calculation

## 2.1 Annualized Return

Formula: R = (∏ (1 + rt))^(252/T) - 1

Explanation: The annualized return is calculated by compounding the daily returns over the period and adjusting it to reflect annual returns, assuming 252 trading days per year.

## 2.2 Annualized Volatility

Formula: σ = √252 × std(rt)

Explanation: The annualized volatility is derived from the standard deviation of daily returns, scaled by the square root of 252 (the approximate number of trading days in a year).

## 2.3 Sharpe Ratio

Formula: Sharpe Ratio = (R - Rf) / σ

Explanation: The Sharpe Ratio measures the excess return per unit of risk, where R is the annualized return, Rf is the risk-free rate (assumed to be 0.01 or 1% in the script), and σ is the annualized volatility.

# 3. Portfolio Optimization

## 3.1 Optimization Objective

The primary objective in the optimization step is to maximize the Sharpe Ratio of the portfolio, which is equivalent to maximizing the risk-adjusted return.

## 3.2 Optimization Methodology

Algorithm: scipy.optimize.minimize with the Sequential Least Squares Programming (SLSQP) method.

Objective Function: Negative Sharpe Ratio (to be minimized, since SLSQP minimizes functions by default).

Example Function:

def max\_sharpe\_ratio(weights, returns, risk\_free\_rate=0.01):  
 p\_metrics = calculate\_portfolio\_metrics(weights, returns, risk\_free\_rate)  
 return -p\_metrics['sharpe\_ratio']

## 3.3 Constraints

The constraints include:

1. The sum of the weights must equal 1 (fully invested portfolio).

2. Each weight must be between 0 and 1 (no short selling or leverage).

Example Constraints:

constraints = [{'type': 'eq', 'fun': lambda x: np.sum(x) - 1}]  
bounds = tuple((0, 1) for \_ in range(num\_assets))

# 4. Process Overview

1. Fetch Data: The script retrieves historical price data for the assets.

2. Calculate Metrics: The daily returns are computed, followed by the calculation of annualized return, volatility, and the Sharpe Ratio based on user-defined weights.

3. Optimize Portfolio: The optimize\_portfolio function determines the weights that maximize the Sharpe Ratio using the SLSQP optimization method.

# 5. Summary

The combination of these algorithms allows for a comprehensive analysis of past performance and the construction of an optimized portfolio that maximizes expected risk-adjusted returns based on historical data. It is important to remember that these calculations are based on historical data and assumptions, and they do not guarantee future performance.