Mean-Reverting and Trending Portfolios

## Strategy Overview

This document outlines a trading strategy by construction a portfolio, consisting of US equities, such that the historical portfolio value shows a mean-reverting or trending pattern. The strategy is back-tested in Python and implemented in C++ utilising Interactive Brokers (IBKR) API infrastructure to obtain real time market data and execution.

## Strategy Details

### Stock Selection

55 US Equity stocks are selected and 10 are selected for portfolio constructions. The 10 stocks are selected using Principal Component Analysis (PCA) and select the 10 stocks with the lowest variance. Selecting the lowest variance ensure a smooth portfolio values time series with minimal noise to show the mean-reverting or trend pattern.

Several five stocks combinations are selected from the 10 stocks, where each combination represent a unique portfolio.

### Portfolio Calibration

Each stock in a portfolio is assign a weight[[1]](#footnote-2), where each weight represent the amount of stocks to hold. In this part of the modelling, I have assume that the portfolio is able to hold a fraction of stocks. The stocks amount will be converted to an integer amount in the back-testing process, which will depend on the investment amount and stock price.

The calibration involved smoothing the historical portfolio value time series using Savitzky-Golay filter with window size of 31 and polyorder of 10. The filter is used to smooth the time series while preserving features such as peak, height, width, and position. This smoothing model is suitable for low noise time series, which I have chosen using the PCA method.

The smoothed time series are then calibrated to Ornstein-Ublenbeck (OU) process using method-of-moment. Method-of-moment involved matching the empirical moments of the observed data to the theoretical moments of OU process.

The calibration is performed by maximising the OU speed of reversion (theta) and minimising the volatility. In C++, I use the NLOpt library and utilising the COBLYA first, then SLSQP optimisation methods if the first method failed to achieve convergence. COBLYA (Constrained Optimisation by Linear Approximation) is a local optimiser and derivative-free, which approximates the objective and constraint functions using linear models built from sampled point. SLSQP (Sequential Least Squares Quadratic Programming) method is also a local optimiser and gradient-based, which solves a series of quadratic approximations to the objective and constraints using gradient information.

The calibration assigns weights to each stock in each portfolio.

### Generating Signals

## Future Improvements

* Calibrating the hurst, half-life, sma multiplication using Gaussian Mixture Model
* Using different mean reverting model. i.e., CIR, Ho-Lee model

1. The total weight in each portfolio sums to 1. [↑](#footnote-ref-2)