Data Summary

- Asset Universe: CSI300
- Date Range: 2005-4 - 2023-8-4
- Data Used: All the data from jointquant
  - components.csv: CSI300 components, monthly data and back fill so that in the future month the weights is the same, ignore the weights change caused by the return, could be better handled if given daily data
  - history.csv: CSI300 history daily data
  - index_history.csv: CSI300 index daily data
  - industry_stock.csv: yearly data of the industry of each stock
  - market_cap.csv: CSI300 market cap data
  - factor folder: all the barra risk factor, data is already normalized. So when we calculate the factor returns, only do the cross sectional normalization. Only use the mean and std of the tickers that is in the index to do the normalization, but better way is to do it for the full stock universe.
- Inter Data Description:
  - expected_rets.csv: the expected returns calculated by the simple momentum signal, actually the momentum signal is not treated correctly as I add an opposite sign
  - factor_rets.csv: the factor rets from the cross sectional regression
  - residual_rets.csv: the residual rets from the cross sectional regression
  - weights_optimize.csv: the weights calculated by the mean variance optimization

Model Summary

- Cov Matrix:
  1. do the cross sectional WLS regression with $Y = ret$ and $X = barra, industry, countryfactor$ to get the factor rets and residual rets, weighted by square root of market cap, and also have a constraint of $\Sigma w_i f_i = 0$ for each sectors to avoid multicollinearity
  2. use the ewma to get the factor matrix, and $Cov = \beta^T Cov_{factor} \beta + Diag(residual\_vol)$
- Expected Returns:
  1. use the simple momentum signal of short term ret - long term ret,
  2. do the cross sectional normalization to get the signal
  3. use a rolling OLS where Y is the ret and X is the lagged signal to get the params and calculate the expected returns(flip it to make it has positive IC, but I think here the alpha is not the focus)
- Mean Variance Optimization:
  1. Use the $W_{i,t} = W_{i,t-1} + W_i^+ + W_i^-$ to convert to a quadratic optimization problem
  2. Use the cvxopt package to solve the quadratic problem daily to get the optimal weights based on the expected returns and cov matrix, the

objective is $w^T r - 0.5 * alpha * w^T \Sigma w$. alpha=5 is selected based on the book "Robust Portfolio Optimization and Management" to have a normally 2-4, and I would like to add more on the risk side

3. The constraint is in matrix form for the w_i, w_i^+, w_i^- as stated in optimize.py

Code Explanation

- calc_rets_and_cov.py: mainly for calculating the factor returns, and the asset cov matrix
- optimize.py: create the optimization problem both in matrix form and readable function to validate
- utils.py: some utility functions to load data and preprocess data
- process_factor.ipynb: main file to run the whole process of getting factor_rets, expected returns and optimize weights, and save the result to csv
- backtest.ipynb: use the bt and pyfolio to check the result of the weights, and also see how the weights break the constraints.

Hypothesis:

1. we could only trade the stocks which is not suspended next date and not in the high limit or low limit, for those stocks, set it to the previous weight for convenience. Here we actually could buy the low limit data and sell the high limit data, but I decide to remove it from tradable for simplicity and could improve the logic in the future.
2. we could only trade the stocks which is in the CSI300 index
3. when there is index rebalance, the turnover constraints is relaxed, and we are going to reduce all the removing stocks to 0. The constraint needs to be relaxed beacuse otherwise there could not have any weights that meets all the criteria. For the removal stock, we could have a better way to handle but setting it to 0 makes sure we alway invest in the index stocks.
4. when there is index rebalance, there could be some breach of deviation from index weight, as sometimes you cannot trade the stock because of the trading halt but they are deviate from the previous one.
5. For 990018.XSHG, I do not get the return series. So in the index weights, I just drop the ticker and scale other weights up to sum up to 1.

Backtest Result:

Please see backtest.ipynb for the backtest result. Overall this gives us a 0.67 SR compared with the 0.41 of index. Not considering the risk free return anywhere.

Future Development

1. Need to better handle the removal of the stock and index rebalance
2. Need to use the whole stock universe to get the factor rets