Final Project
Due: Dec 13, 2017

- **Topic: Multi-factor Portfolio Trading Strategy (100 points)**

  Consider the problem of investing a total amount of $10,000,000 in a large universe of stocks. The Chinese stocks traded on the Shanghai Stock Exchange and Shenzhen Stock Exchange (over 2000 stocks) are chosen to be the entire universe.

  Suppose a subset of 1000 (or 1500) stocks are selected from the entire universe. At every rebalancing time point, a small group of 100 to 150 stocks are selected out of the 1000 (or 1500) stocks during the in-sample backtesting period based on the following criterion for long position.

  - **Stock selection criteria**
    - Criterion 1: The market capitalization needs to be no less than 500,000,000 (RMB)
    - Criterion 2: The average daily trading volume over the past 15 business-days needs to be no less than 1,000,000.
    - Criterion 3: Computing M-score based on a group of factors for each stock i which passes Criterion 1 and 2. Let $C^i(t)$ denote the price of stock i in period t. The group of factors are described below.
      - Factor F_1: Price to Book ratio (PB)
      - Factor F_2: Price to CashFlow ratio (PCF)
      - Factor F_3: Price to Earning ratio (PE)
      - Factor F_4: Price to Sales ratio: PS
      - Factor F_5: n-period momentum factor (PM): $PM^i(t,) = \ln \frac{C^i(t-1)}{C^i(t-n)}$; (n = 5)
      - Factor F_6: m-period reversion factor (PRev): $PR^i(t) = \ln \frac{C^i(t-m)}{C^i(t-1)}$; (m = 20)
      - Factor F_7: L-period log-return volatility (Vol): $Vol^i(t) = \sigma_L^i(r(t-1))$ where $\sigma_L^i(r(t-1))$ denotes the annualized standard deviation of the log-return of stock i over time window $[t-L, t-1]$.

      Choose a weight vector $w \equiv (w_1, w_2, \cdots, w_7)$. For example, $w = (0,0,0,0,0.5,0.5,0)$ or $w = (1/7, 1/7, 1/7, 1/7, 1/7, 1/7, 1/7)$. M-score for stock i is calculated as:

$$M_w^i(t) \equiv w_1 \cdot F_1^i(t) + w_2 \cdot F_2^i(t) + \cdots + w_7 \cdot F_7^i(t).$$

  - **Portfolio construction**
    - Choose a rebalance frequency to be every U periods. U can be set to a number between 1 and 20. This means while the M-scores are computed for all the periods from $t = 0, 1, 2, \cdots, T_N$, one can choose to rebalance the portfolio only at time points: $0, U, 2*U, 3*U, \cdots$ to avoid incurring excessive transaction costs.
    - At each time point t, select the best K stocks with highest M-score. Initially, choose $K = 100$.
    - Put 1% of total account value at time t into each of the 100 stocks.
    - Transaction cost is $TC_{cost} = 0.001$ of total dollar amount of transactions.
  - **Backtesting and Performance Evaluator** During the entire testing period from t_0 to t_N, calculate the annualized daily return of this factor-based portfolio strategy R_t and evaluate the performance of a given set of trading parameters by the following criteria:
    - Total profit/loss
    - Average daily return (annualized)
    - Sharpe Ratio
    - Maximum Drawdowns (MDD)

- **Data and Implementation Expectation**
  - Ticker universe and the fundamental data are provided as csv files in the Resources on $T-Squares$. **Note**: some of the tickers in the fundamental dataset have the suffix ".SH" which correspond to the security tickers ends with suffix ".ss" in the ticker universe file.
  - You are expected to download all the price data from Yahoo finance and save them in ./data folder in your project directory.
  - Your implementation shall contain at least the following classes.
    1. SecurityData Class: contain all the relevant time series data for a given universe of tickers.
    2. Strategy Class: generate buy/sell signals for a universe of securities and their corresponding data
    3. Portfolio Class: generate portfolio of securities based on the buy/sell signals generated in Strategy Class.
  - **Strategy input parameters**:
    - Score weight vector $w = (w_1, w_2, \cdots, w_7)$ for computing the M-score. This can be determined by running cross-sectional regression of daily (or weekly) return over the 7 factors.
    - n value for PM, m value for PR and L value for Vol.
    - Frequency U at which the portfolio is rebalanced.
- Pseudo Code
  - Divide the data into two periods: in-sample testing period from 0 to T_{test} and out-of-sample testing period from $T_{test}+1$ to T_{end}.
  - Loop: for time t from 0 to T_{test}
    - In each period t, apply the stock selection criteria to get a subset of N_t securities (N_t can vary with t).
    - Compute the M-score for the N_t securities.
    - Form the portfolio based on the strategy rule if $t = 0$. If $t > 0$ and t is a rebalancing time point, rebalance the portfolio based on the strategy rule (namely, buy stocks which become members of the top 100 M-score group and sell stocks which fall out of the top 100 M-score group.
    - Compute the profit and loss from the rebalancing and record the return in time t.
  - End of loop.
  - Vary trading input parameters to obtain the best performance of this trading strategy in time period from 0 to T_{test}.
  - Apply the trading parameters to time period $T_{test}+1$ to T_{end} to obtain the out-of-sample performance of the trading strategy with the best parameter set obtained using data in time period from 0 to T_{test}.
- **Beta neutral constraint implementation**
  - At every rebalance time t, use the values of 000300.ss as market index in the past 100 days to compute the beta of each stock with respect to this market index using the CAPM model.
  - Calculate the total beta of the 100 stocks with the highest M-scores at time t, denoted by $\beta_{port}(t)$. Add a short position of -1*$\beta_{port}(t)$ shares of 000300.ss to the portfolio of 100 stocks so that the overall portfolio has a 0-beta with respect to 000300.ss. Report the performance of this new portfolio and compare it with the original portfolio.
- **Deliverables**

  A final report (in Word or latex format, donnot submit pdf file) shall be submitted. The report is expected to contain the following parts.
  - Problem description.
  - Details on the implementation framework and code structure description
  - **Use data prior to 2014-10-31 as in-sample testing data and data from 2014-11-1 to 2015-7-31 as out-sample-test dataset**.

* The performance of the trading strategy under the best set of **strategy input parameters** and how the best trading parameter set is obtained.
* Use tables to illustrate the percentage of trades being winning/losing ones, the average profit/loss per trade, the annualized rate of return and the standard deviation of the daily return, and the Sharpe ratio for both in-sample testing period and the out-of-sample testing period separately. Use figures to show the cumulative growth rate of account value (with starting value being $1 and the distribution of the daily returns of the account value.

Submit all the program codes/documentations as ONE zip file containing the final report and a sub folder containing all the codes and input configuration file. In each folder, write a readme file to record the compiler and library you use and any other necessary information to run your codes, i.e. how to fill in your parameter configuration file. Use relative path when reading data file.

Also document the estimated time to run your program if you search all combinations of parameters to find the best trading parameters by brute force. If the running time is more than one hour, provide a relatively small test case of a parameter domain which contains the best trading parameter set.