**Construct a long-short backtester**

The objective of this project is to write a class that will conduct a long-short backtest on a set of data.

The dataset is provided at a daily frequency and contains forward and backward return data along with factor/signal data. The dataset contains the following fields:

|  |  |
| --- | --- |
| **Field** | **Description** |
| date | The date to which the data corresponds |
| id\_security | The identifier of the security |
| fm\_1wd | The 1 week day forward return – i.e. the total return from t to t+1. |
| m\_1wd | The 1 week day backward return – i.e. the total return from t to t-1. |
| factor\_1 | The factor/signal value for factor 1. |
| factor\_2 | The factor/signal value for factor 2. |

**Simple Backtest**

The backtest is run by going long the top x% and short the bottom x% of securities on a given factor score at each date. So if x=0.2 we go long the top 20% and go short the bottom 20% of securities based upon their factor score on each date. We equal weight the long and short positions of the portfolio. That is to say if x = 0.2 and we are long 10 securities and short 10 securities then we will take a position of 1/10 in each long security and -1/10 in each short security. At each date we rebalance the portfolio and reweight the underlying security positions to be once again equally weighted.

This backtest is defined by two parameters:

1. Factor: (e.g. ‘factor\_1’ or ‘factor\_2’) this is the factor to be used in running the backtest.
2. Percentile: (e.g. 0.2) this is the percentile cut-off used in determining which securities are in the long and short portfolio.

The backtest should then output the periodic (daily) returns of the long, short and long-short portfolio.

**Extension**

As an extension calculate the turnover of the portfolio at each re-balance and adjust the daily returns for execution costs. The execution costs are applied as a percentage of the absolute notional turned over at each rebalance.

**Other**

It is preferred that you use the Python programming language to solve this task; however if you are not particularly familiar with the language then you may use another language of your choice.