

## A Simple backtester

### Requirements:

- pyfolio-reloaded
- pandas
- numpy

This is a simple backtester that measures performance of a series of trades. To run the backtester, download the folder *backtest* and copy it to a location of your choice. The folder has four folders:

- *../backtest/data* has trade files. The trades files are in CSV format with four columns --Symbol, StartDate, EndDate, Trade
- *../backtest/backtest* has the Python code
- *../backtest/result* is folder where log file and output files are generated
- *../backtest/test\_data* has data for unit tests

Backtester needs to set up correct paths. Change the `top_dir` in *../backtest/backtest/backtest/utils/path.py*. Backtester can run in two modes, one using config file, and other using command line options. To run using config options use:

```
>>../backtest/backtest/backtest/run_backtest.py --config_file <config_file_path>
```

Default config is in *../backtest/backtest/backtest/config.cfg*

To run using default config do the following

```
>>../backtest/backtest/backtest/run_backtest.py --trade_file <trade_file_path>
```

Accepted arguments are :

*--aum (default \$100M)*

*--maxTradeADV (default 5 %, pass 10 to change it to 10%)*

*--maxDollarTrade (default \$10M)*

*--trade\_file*

*--write\_stats (default True) write backtest stats to the result directory*

*--pyfolio\_analysis (default True) create full tear sheet using Pyfolio analysis*

```
>>../backtest/backtest/backtest/run_backtest.py --trade_file <trade_file_path> --maxDollarTrade 5000000
```

The backtester generates a tear sheet using pyfolio. It also generates four csv files in *../backtest/data* folder -- *Full\_stats\*.csv*, *Portfolio\*.csv*, *PL\*.csv*, *Yearly\_Stats\*.csv*.

Notes:-

1. The backtester assumes that trades are executed on tradedate and execution price is average of OHLC for that day. I have used average of OHLC as a proxy for VWAP. The execution slippage to avg\_OHLC is assumed to be zero. It would not be very difficult to add a simple VWAP slippage model to the backtest and make it more realistic.
2. The backtester is generic and can run for any period and any set of securities as long as it has the price volume data for the period. Currently it downloads five years of data from yahoo/finance.
3. There is a script `..\backtest\backtest\data\create_trades.py` to generate random trades.
4. The returns calculated by backtester may not match with pyfolio. I use AUM to calculate returns. The dollar PL should match.