**Backtesting for multiple asset prices and strategies**

**Negative numbers🡺 Long, Positive nembers🡺 Short**

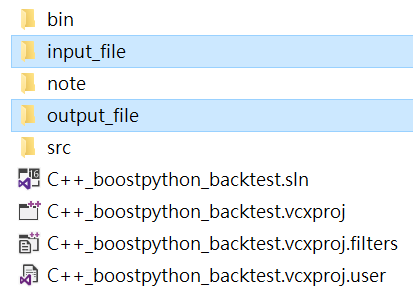
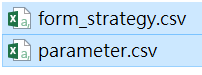
**Put the price and strategy data in form\_strategy.csv, and set parameters in parameter.csv at relative path of ./input\_file**

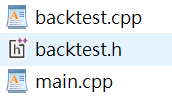
**Create folder output\_file for the output**

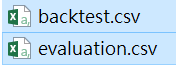
**After including backtest.h and linking backtest.cpp, use writeAllBacktest function in main.cpp to produce backtest.csv and evaluation.csv at relative path of ./output\_file**

fopen path of program

Source code







**Input file : ”form\_strategy.csv”, “parameter.csv” in relative path of ./input\_file**

**Format of form\_strategy.csv:**

first row:time,asset1,strategy1,asset2,strategy2,...

other rows: price and strategy data

number of asset and strategy should be same

**Format of parameter.csv:**

first row:eachTradeCost,percetageOfTradeVolumnCost,

dataFreq,riskFreeRate,investMoney

second row: data

dataFreq represents how many data points for each asset during a year

**Output file : ”backtest.csv”, “evaluation.csv” in relative path of ./output\_file**

**Format of backtest.csv:**

first row:time,transactionCost, realizedCPnL, unrealizedPnL, cumulativePnL, momentPnL

other rows: data

**Format of evaluation.csv:**

first row:start time, end time, eachTradeCost,percetageOfTradeVolumnCost,

dataFreq,riskFreeRate,investMoney

second row: corresponding data

fourth row: totalPnL, annualReturn, annualSTD, annualDownSTD,Sharpe Ratio,Sortino Ratio, Max DrawDown

fifth row: corresponding data

after fifth row: Win/Lose Analysis for all assets and strategies

**Name description**

**transactionCost:**

transaction cost at each moment = eachTradeCost+price\*strategy\*percetageOfTradeVolumnCost

**realizedCPnL:**

realized cumulative PnL based on price, strategy, transaction cost(include cost)

**unrealizedPnL:**

unrealized PnL at each moment(include cost)

**cumulativePnL:**

realizedCPnL+unrealizedPnL

**momentPnL:**

this moment’s cumulative PnL- previous moment cumulative PnL

**totalPnL:**

cumulative PnL at the end of time

**annualReturn:**

annual return=average(daily holding PnL)\*252/ invest money

**annualSTD:**

annual STD= STD of daily holding PnL\*sqrt(252)/ invest money

**annualDownSTD:**

Assume minimum acceptable return is equal to riskFreeRate

Use minimum acceptable return to calculate target average daily PnL

Use target average daily PnL to calculate downside STD of daily holding PnL

annual downside STD= downside STD of daily holding PnL\*sqrt(252)/ invest money

**Sharpe Ratio:**

(annualReturn- riskFreeRate)/annualSTD

**Sortino Ratio:**

(annualReturn- riskFreeRate)/annualDownSTD

**Max DrawDown:**

max drawdown of the strategy during entire period

**Win/Lose Analysis:**

Only consider realized win/lose and no markup or markdown situation

Following the order of assets in form\_strategy.csv

First Part: Time,Win/Lose,Amount

Second Part: no. of transaction,no. of win,total amount of win,no. of lose,total amount of lose,win rate

**Source code: “main.cpp”, “backtest.h”,”backtest.cpp”**

Functions are listed in backtest.h

The detail implementation and illustration are in backtest.cpp

How to use these functions are decribed in writeAllbacktest function at the end of backtest.cpp

Transfer all function into python function at the end of main.cpp

Extend the program?

copy backtest.h and backtest.cpp to get the useful function. After the copy, it is important to adjust right path at read/write functions at backtest.cpp for input files and output files. In Visual Studio, the path is relative to the file position of .sln or .vcxproj.