Final Report
FALL 2022
SP OBJ ORIENTD PROG 22:839:614:41
Backtesting Quantitative Trading Strategy on Digital Currencies
By Lau, Ming Him-ml1762 and Tai, Chun Yi-ct726

Financial Findings:

- Data description
  - Bitcoin BTC/USD pulled from Bloomberg Terminal
  - OHLC historical price data of 4 sets with different time intervals.
  - Daily data: from 11/01/2017 to 10/31/2022 5 years of data
  - 10-min data: 05/01/2022 to 10/31/2022 half year of data
  - 5-min data: same as 10-min data
  - min data: same as 10-min data

- Trading method:
  - Rule-based trading strategy with 100,000 USD initial capital, whenever signal appears, all of the portfolio capital will be used to buy/sell the underlying.
  - Optimization is done by grid search looking for parameter set that meet a threshold Sharpe ratio and generate the highest total profit. With Sharpe ratio = $\frac{Mean(interval\ profit)}{Standard\ Deviation(interval\ profit)}\sqrt{t}$ where t is the annualization factor for example, daily data t = 365

- RSI Strategy:
  - Basic information about RSI: https://www.investopedia.com/terms/r/rsi.asp
  - When RSI is higher than a preset sell sign value e.g., 70, it indicates overbuy situation and it's a sell signal that we will short sell the underlying and when RSI come back from over buy zone to mid-line 50, we will stop gain/loss, vice versa for buy signal.
  - We find the results not very appealing in the daily data, but it performed pretty well in the minutes datasets. Results are shown below:

```
RSI strategy backtesting for BTC_data_daily.csv:

  Total Profit  Sharpe Ratio    RSI period       RSI buy      RSI sell
     -89760.51         -0.49         15.00         31.00         66.00


RSI strategy backtesting for BTC_data_10-min.csv:

  Total Profit  Sharpe Ratio    RSI period       RSI buy      RSI sell
      41378.78          1.58         14.00         28.00         65.00


RSI strategy backtesting for BTC_data_5-min.csv:

  Total Profit  Sharpe Ratio    RSI period       RSI buy      RSI sell
      50031.69          1.84         21.00         33.00         66.00


RSI strategy backtesting for BTC_data_1-min.csv:

  Total Profit  Sharpe Ratio    RSI period       RSI buy      RSI sell
      15419.17          1.27         20.00         21.00         80.00
```

We decided to reverse the signals and we can see below, we buy when we see a sell signal, and sell when we see a buy signal. And results are shown below:

```
//update position
if (curr_position == 0){
  if (signal == Signal::BUY){sell(timeTick);}
  else if (signal == Signal::SELL){buy(timeTick);}
}
else if (curr_position < 0){
  if (signal == Signal::BUY){sell(timeTick);}
  else if (signal == Signal::HOLD){stop(timeTick);}
}
else if (curr_position > 0){
  if (signal == Signal::SELL){buy(timeTick);}
  else if (signal == Signal::HOLD){stop(timeTick);}
}
```

We can see huge improvement and much better profit results in the daily data as this is more like a trend strategy instead of a mean-reversion strategy.
And overall, RSI works better in shorter-time-interval data like in the 1-min data and relatively bad in longer time period.

```
RSI strategy backtesting for BTC_data_daily.csv:

  Total Profit  Sharpe Ratio    RSI period       RSI buy      RSI sell
    1904943.65          0.74         14.00         34.00         60.00


RSI strategy backtesting for BTC_data_10-min.csv:

  Total Profit  Sharpe Ratio    RSI period       RSI buy      RSI sell
      36751.61          2.56         15.00         29.00         72.00


RSI strategy backtesting for BTC_data_5-min.csv:

  Total Profit  Sharpe Ratio    RSI period       RSI buy      RSI sell
      42133.55          1.37         14.00         40.00         60.00


RSI strategy backtesting for BTC_data_1-min.csv:

  Total Profit  Sharpe Ratio    RSI period       RSI buy      RSI sell
     662154.85          7.71         15.00         39.00         60.00
```

- MACD Strategy:
  - Basic info about MACD: https://www.investopedia.com/terms/m/macd.asp
  - Besides RSI, we also tested for another trend strategy using MACD.
  - Buy signals are generated if both MACD and Signal lines are above 0 and MACD > Signal line, vice versa. Below are the backtesting results:

```
MACD strategy backtesting for BTC_data_daily.csv:

  Total Profit  Sharpe Ratio     MACD Fast     MACD Slow   MACD Preiod
    1942303.73          0.73         13.00         32.00          9.00


MACD strategy backtesting for BTC_data_10-min.csv:

  Total Profit  Sharpe Ratio     MACD Fast     MACD Slow   MACD Preiod
      23808.20          0.97         20.00         35.00         21.00


MACD strategy backtesting for BTC_data_5-min.csv:

  Total Profit  Sharpe Ratio     MACD Fast     MACD Slow   MACD Preiod
      78064.11          2.34         12.00         35.00         12.00


MACD strategy backtesting for BTC_data_1-min.csv:

  Total Profit  Sharpe Ratio     MACD Fast     MACD Slow   MACD Preiod
     103867.24          3.06         10.00         34.00         16.00
```

We also tried to relieve the condition of both lines > 0 (or < 0) that we buy when MACD line > Signal line and sell when MACD line < Signal line. In this strategy, we basically do not stop gain or loss. We either full-load buy or full-load sell.

```
MACD strategy backtesting for BTC_data_daily.csv:

  Total Profit  Sharpe Ratio     MACD Fast     MACD Slow   MACD Preiod
    3412879.85          0.40         13.00         32.00          9.00


MACD strategy backtesting for BTC_data_10-min.csv:

  Total Profit  Sharpe Ratio     MACD Fast     MACD Slow   MACD Preiod
     114025.68          1.95         10.00         27.00         15.00


MACD strategy backtesting for BTC_data_5-min.csv:

  Total Profit  Sharpe Ratio     MACD Fast     MACD Slow   MACD Preiod
      84959.43          1.79         14.00         28.00         11.00


MACD strategy backtesting for BTC_data_1-min.csv:

  Total Profit  Sharpe Ratio     MACD Fast     MACD Slow   MACD Preiod
     287314.13          5.49         11.00         32.00          9.00
```

We can see the total profit of the second MACD strategy is somewhat better than the first one because of the high volatility of the BTC

OOP Findings

- Abstraction: we used header files and classes to practice data and control abstraction, which made the program more readable, easier to use, and less complex.
- Encapsulation: at first, we declared member variables in base class as protected, but this may cause some problems (fragile program, inconsistent state). So, we finally practiced encapsulation by building setters and getters.
- Inheritance: the class inheritance is so convenient that we only need to define a new signal function and fine tune some functions of the base class. We can reuse all the functions and methods defined in the base class.
- Polymorphism: for virtual functions, we found that if the parameter for the derived class function (same name) are different from the base class, we can just overload it instead of overriding it. In other words, we can also achieve polymorphism through the use of function overloading.
- Powerful way to develop: Imagine in the trading world, there are many strategies. In each strategy, there are different way to implement, different algorithm to compute indicators, and various parameters for grid search. It's not an easy thing to develop. Yet through OOP, we can architect strategy one by one in an organized and elegant way.

Further Aspects for Improvement

- We did not include bid-ask spread, transaction free and short-selling cost In the backtesting class, which makes the model unrealistic and the actual profit and Sharpe ratio in real world adopting those strategies would be significantly lower than our results.
- Everytime we see buy/sell signals, we go full-load buy or sell which makes our portfolio extremely risky and building a logic to determine the bet size should be a good improvement to the risk/reward payoff. We can also add stop loss/time stop (stop the position after certain time periods no matter gain or loss) to further define our strategy.
- The Sharpe ratio we used has a annualization factor t and when time period is short the t gets very large which can distort the Sharpe ratio and we may choose another performance indicators for optimization.
- In designing the class structure, we overlooked the importance of data storage that we need to use a for-loop calculating the trading activities every time tick and maybe we should have made the model fully vectorized like DataFrame in Pandas in Python.