Toolbox API Assignment

Language: Python (>3.8)

Packages: Pandas, Numpy, Matplotlib

Description:

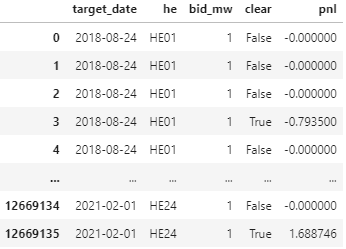
**Design a module named “backtest” to support the internal trading team. The backtest module should be a child class of the famous Pandas package(https://pandas.pydata.org/) so that people can use the instance with familiar grammar and handy methods from Pandas.**

Backtest is a very important step when designing strategies for trading and it could require a lot of tedious work when traders/researchers want to come to some solid conclusions. In order to make the research/trading activities more efficient, a handy tool is required.

A sample dataset for the testing purpose can be found in Github Repo page.

Task:

One crucial step in backtest is to report the results. Assume you have a standard input dataframe which contains the following information.



Columns:

Target\_date: datetime/str(%Y-%m-%d)/pd.Timestamp

He: str/int

Bid\_mw: int/float

Clear: int/bool

Pnl: float

Here targer\_date and he are the time information of the backtest result. Bid\_mw is the size of each trading activity and clear indicates whether the bid is cleared in the market or not(True means the bid is cleared, which suggests we successfully trade something in the market. False means we can’t make a deal). Pnl stands for profit and loss is how much money we made/lost for this trading activity.

1. Write a function named report under class Backtest to report the following information in the format of dataframe. This function should be treated as property.

|  |  |
| --- | --- |
| **Stat** | **Definition** |
| Start date | First date that the backtest could have traded |
| End date | Last date that the backtest could have traded |
| % of Possible Volume Bid | The number of MWh bid divided by the number it would have bid if it bid every hour. |
| % of Hours Bid | # of distinct hours bid / # of hours in backtest |
| Average Bid Size (when Bidding in an Hour) | total volume bid / # of distinct hours bid |
| % of Volume Bid Cleared | total volume cleared / total volume bid |
| % of Hours Bid Cleared | # of distinct hours cleared / # of distinct hours bid |
| Average Clear Size (when Clearing in an Hour) | total volume cleared / # of distinct hours |
| Profit per MWh | total profit / total volume cleared |
| Win % | The sum of all MWh cleared with positive profit / total volume cleared |
| Profit per MWh without Best 1% of Days | total profit (excluding 1% of most profitable days, rounding up on # of days) / total volume cleared (excluding 1% of most profitable days, rounding up on # of days) |
| Expected Daily Bid | total MWh bid / days in backtest |
| Expected Daily Profit | total profit / days in backtest |
| Max 2-Day Loss | minimum observed profit (largest negative profit) over all 2-consecutive-day periods in backtest |

Hint: use pandas methods to make it faster and simpler.

1. Write a function named plot to plot cumulative pnl of the backtest period. It should give us clear label information on the plot and x-axis should have time information on it.

Hint: some tutorial can be found at documentation of Matplotlib.