Introduction

Welcome to Algorithmic Trading Club @ TMBA! This is your first homework assignment. The purpose of this assignment is to get you familiar with the basics of Python for financial data analysis and constructing trading strategies. You don't need to write any report / slides after finishing this assignment, it's just for you to get familiar with how to build a backtest system for single-asset CTA trading strategies.

Instructions

In the folder `src/`, you can see four Python files:

- `main.py`
- `utils.py`
- `backtest.py`
- `__init__.py`

Where `main.py` is the main file for you to execute the backtest system. You can run the backtest system by executing the following command in the terminal:

```
~/HWs> python -m src.main
```

In `utils.py`, you can find alot of useful functions for you to use in the backtest system, such as `load_data()` for loading the data, `plot_dd()` for plotting the drawdown of your strategy, etc. In `backtest.py`, you can find functions for you to implement trading indicators and trading strategies.

Requirements

To start programming, you need to install the following packages if you haven't done so. I have wrote them into the `requirements.txt` file, you can just type the following command in the terminal to install them. A good habit is to create a virtual environment before installing the packages to prevent packages version conflicts.

```
~/HWs> python3 -m venv your_venv_name
~/HWs> source your_venv_name/bin/activate

(your_venv_name) ~/HWs> pip3 install -r requirements.txt
```

After these steps, you are ready to start programming!

Tasks

I have already implemented a simple moving average crossover strategy in `backtest.py`. You have two choices to complete this assignment:

- **Not familiar with Python and CLI** [1]**:** You can first understand the code and the backtest structure, and then try to run the code in your local environment successfully.
- **Familiar with Python and CLI:** You can try to implement a "Moving Average Crossover" strategy based on the current structure (Or other trading strategies if you want to do so).

Hints

Moving Average Crossover

To implement the moving average crossover strategy, you need to first calculate one more moving average line, be aware that two moving average lines need to have different periods (different `k`).

```
get_moving_average(data, args.k1)
get_moving_average(data, args.k2)
```

To generate the trading signals, you can refer to the following links:

- https://trendspider.com/learning-center/moving-average-crossover-strategies/
- https://www.oanda.com/bvi-ft/lab-education/dictionary/golden-cross_death-cross/

CLI Execution

Type the following command in the terminal to run the backtest system:

```
~/HWs> python -m src.main --k1 10 --k2 20
```

Supplementary Materials – `ArgumentParser`

`ArgumentParser` is a class in the `argparse` module that allows you to parse command-line arguments. You can use it to add arguments to your script, and then parse them. Here is an example of how to use it:

[1] Command Line Interface

```
"""
-*- encoding: utf-8 -*-
main.py

Make sure to first `pip install argparse`
"""
from argparse import ArgumentParser

def parse_args():
    parser = ArgumentParser()
    parser.add_argument("--k", type=int)
    return parser.parse_args()

if __name__ == "__main__":
    args = parse_args()
    print(args.k)
```

And you can run the script with the following command:

```
~/HWs> python main.py --k 10
```

it will print out 10.

Want to know more?

- The source code of this homework & the full version of the backtest system can be found in my GitHub repo – Quant-Finance
- Coding Style Manual wrote by myself. A good coding style can make you easier to find a job in quantitative trading industry / tech companies.
- Interview experience of Kronos Summer Analyst. If you're interested in what the interview process of a quant trading firm looks like, you can check this out.

Any questions, feel free to ask me on Discord / Facebook.