

Python programming for Finance

M203

Université Paris-Dauphine PSL

Individual Project Instructions

Goal: To apply the Python programming skills achieved during the course by creating a functional API to distribute data useful in a financial context.

Deadline: Friday, November 26, 2021, 23:59

Instructions

You will be graded based on you're the clarity and professionalism of your code as well as on the originality of your project.

1. Data collection (50%): The goal of the project is to collect and distribute data that could be used by an investor. The first step of the project is the data collection. You are expected to find and web-scrape data (any data) on the internet satisfying the following criteria (if you are not sure if your data satisfies the criteria send me an email):
 - a. There are no requirements on the frequency on the data (daily, weekly, monthly...).
 - b. Your data must contain at least 100 observations, but larger samples are encouraged. If instead of a time series you have cross-sectional data make sure you have at least 100 observations.
 - c. Data can be in any format; non-numeric data must be quantified using any methodology.

When looking for data, think of information that could be relevant for an investor (e.g., reddit posts for bitcoin investors).

- d. Obtain financial data of the type of investment that could be implemented with this data, you can use the YahooFinance API for this step.
- e. You need to provide Python code that web-scrapes all the data automatically. Write the Python code in individual .py files and not in

a Jupyter notebook. You can use notebooks to test your code while you write it.

2. Data Distribution (30%): Follow the lesson on creating your own API to distribute the data. Your API must be flexible enough to accommodate parameters such as the time interval you want the data from. Provide Python code that runs locally in order to test the API functionalities. Write the Python code in individual .py files and not in a Jupyter notebook. You can use notebooks to test your code while you write it.
3. (20%) Do a statistical analysis to test if your data can be helpful in an investment decision. You can implement any type of statistical analysis as well as financial backtests. Options include doing linear regressions, machine learning models, correlations, etc. Write the Python code in individual .py files and not in a Jupyter notebook. You can use notebooks to test your code while you write it.

Delivery: Upload all files in a compressed folder to Moodle, together with a text file called README with a short description of what each files does. I should be able to run everything, including 1) collecting 2) distributing and 3) analyzing the data.