M2 272 - Bloomberg: data retrieval & applications

Celia Fseil
Université Paris-Dauphine

December 31, 2023

Introduction

When it comes to quant investment strategies, putting an idea to practice brings its lot of hurdles both in terms of feasibility and implementation. The goal of this project is to extract a research finding from a paper and turn into into an investable strategy.

In groups of two, you will be choosing a paper from the list of pre-selected publications of varying degrees of complexity.

1 Part 1 - Exploratory work

In the first part of the project, you will focus on the idea generation aspect of it. You will study the paper, perhaps look into additional literature, explore the availability of the data on Bloomberg and come up with an implementation plan we will be discussing together in February.

The exact data source used in the paper is most likely unavailable to you. A key aspect of the project is to find a way to extract data as exhaustively as possible using the Bloomberg terminal and the API. A first step is to go through the various fields made available through <**FLDS**>. A second step is to try to find terminal functions that may contain the data you are looking for and attempt a manual export to see if there are any fields you might have missed in the first step of the exploration. Finally, you may reach out to the helpdesk to ensure you have not missed valuable information elsewhere.

2 Part 2 - Implementation

The second part of the project will consist in building a backtesting tool framework for quantitative investment strategies in Python.

The project will contain six parts:
1- Retrieving the necessary data
2- Constructing the signal
3- Implementing the strategy
4- Implementing a weighting scheme (optimized or other)
5- Computing the track record and the risk measures
6- Producing all files required for the portfolio update/upload in PORT

In this second part, you will distinguish the research code (the one that will allow you to decide on what your strategy will be) from the production code (the one that will implement the strategy). Both programs will draw on the aforementioned six parts which ought to be as generic as possible to allow for re-use for other projects.

3 Deadlines

First meeting : around the end of February. An email will be sent out ahead.
Final handout deadline : TBC/05 11:59 pm to send to fseil_celia@hotmail.fr
Defense date : Compulsory defense sessions will be carried out in May. If these sessions are to happen on Teams, you will be receiving a link to choose some time slots.

4 Grading - subject to modification

- **Code quality and clarity** : organization, variable names, comments
- **Results accuracy** : this is the most important part of the project. Make use of PORT to run some sanity checks (e.g. by comparing the performance you compute with the one computed by PORT)
- **Defense presentation** : you are expected to present your work the same way you would to a co-worker (pro-tip: a small chart detailing the code structure would be helpful). The answers to the questions raised should be well argumented.
- **Creativity** : this type of project is by essence a work of creativity so do not hold back if you believe you have an idea that can improve the strategy

5 Data

The data you will need for this project will be retrieved using the Bloomberg Python API. You will build upon our work done during the two course sessions to create a BLP API class module with all the relevant fuctions.

A key challenge to this project is to take into account the fact that Bloomberg works with

quotas and limits of data and that you should only request what you need / do not already have.

6 Optimization

Depending on the nature of your project, there might be need for weights optimization. Equal weighting and volatility scaling will be the bare minimum. If need there be, you can implement other optimized allocation schemes: minimum variance, max Sharpe ratio, max diversification, risk parity etc. Do not use pre-built packages for this purpose. Look into the scipy.optimize minimize function.

7 Interface

Also dependent on the nature of the project, you should allow the user to choose the inputs. At the very least, this will be done via the python shell. For those who wish to go the extra mile, a small Gui (in tkinter for example) will be appreciated .

Input examples :

- The investment universe (e.g. ticker of an index)
- The backtest period (e.g. a start date)
- The rebalancing frequency (e.g. every 20 business day, end of month etc.)
- The level of transaction costs
- Any other input that are relevant to your project

Mandatory outputs :

- Figures:
  - Overall performance
  - Annualized performance
  - Daily, monthly and annualized volatility
  - Sharpe ratio
  - Maximum Drawdown
  - Historical VaR
- Charts
  - The track record
- All necessary outputs for the PORT update

8 Code organization

You are free to organize your code whichever way you deem most appropriate. Bear in mind that the clarity of your program and the pertinence of your choices will be evaluated.

9 Papers

A link to a Google form containing the papers list will be sent out to you in a separate email.

Miscellaneous

You can send me your questions by email. Describe the problem encountered in the body of the mail as precisely as possible. I cannot guarantee an answer after 01/05.
Good luck!