Quantitative Trading Project

May 4, 2021

1 The Task

As part of our program of study, you (and 0, 1 or 2 group partners) will invent, run and analyze a quantitative trading strategy. Such a strategy has the following features:

- Investment in publicly traded (but not necessarily electronically traded) assets

- A "recipe" for evaluating attractiveness of potential trades that

    - depends on pre-specified classes of information, and
    - could be followed by anyone, given the recipe and data sources

- A "recipe" for investment sizes, and position entry and exit rules, that

    - may link position size to attractiveness
    - ensures risk stays within limits and capital remains bounded
    - makes reasonable assumptions about entry and exit liquidity

Elements of trading strategies include lead-lag relationships, market making, carry trades, pairs trading, factor investment, model reversion, merger arbitrage, "technical" signals and more.

Your trading strategy should have the following features:

- Significant dependence on at least one class of information that is *not* twitter, equity high-low-open-close prices, equity VWAPs, or equity

trading volumes[1], unless at relatively high frequency (600 seconds or less)

- Not essentially equivalent to Fama-French-Carhart factors

- As of at least some times in your simulation, simultaneously holds at least 5 distinct assets (different currencies, commodities, option maturities/strikes etc.)

- Produces a total of 40 trades or more over your entire analysis/simulation period, without excessive clustering

- Leverage, with reasonable capital and risk control assumptions

- An expression in either R or Python code

This project is due at 23:00:00 Chicago time on the Friday after our final official class time[2].

---

[1]Examples include debt ratios, option volatilities, commodity futures prices, bond rates, news items and more.

[2]Late submissions will be accepted at a 10% penalty until 11:30:00 the following morning and at a 50% penalty until 11:30:00 Sunday morning.

2 Analysis and Code

You will analyze your strategy, and provide[3]

- A "pitchbook" of presentation slides, as though intended to attract non-quants to allocating money to the strategy

- A paper in Jupyter notebook form. This will have a technical analysis of the study, as though intended to give quants an understanding of the strategy and its properties

  - reviewing academic literature germane to its design
  - reprising the essential timeline of your research into the strategy
  - giving the quantitative rationale behind the strategy
  - characterizing performance

- PDFs of 2 or more academic papers most relevant to your strategy's design

- Source code[4], data files (if not too large) and documentation necessary for running the strategy in a backtest/simulation

Both the pitchbook and the paper should be well-written with fair and balanced analysis of the attractions, shortfalls and risks of your strategy. Make note of and support decisions and judgment calls (for example, lookback periods or quantile choices). Your work should quantify important risks wherever possible, and explain unquantifiable but important risks clearly,

---

[3]Zip everything in a directory and submit exactly one file: your zipfile. Include the last names of all project participants as part of the zipfile name. Examples are

> Good:`Lee_Boonstra_Greco.zip`
> Bad:`Final_Project.zip`

Nonconforming filename styles such as the "Bad" example above, and submissions consisting of multiple files, are accepted at a 10% penalty to the grade.

Include full names and student ID numbers on the first section of both the pitchbook and technical study. Pitchbooks should be in PDF format. Technical studies must be Jupyter notebooks. Other formats are accepted at a 50% penalty to the grade. Missing student IDs are accepted at a 5% penalty to the grade.

Only one submission per group is necessary, but multiple submissions of the same file are happily accepted.

[4]At most a tiny amount of source code should appear in your technical paper. Instead, have your notebook import methods from a source code library module.

while underplaying or eliding unimportant risks. Consider benchmarks, available size, and trading costs.