A Data Scientific Approach to Equity Backtesting Research

Masters in Data Science (STA5079W) Statement of Research Interest
University of Cape Town, South Africa
Prepared by : Riaz J Arbi
Supervised by: Associate Professor Tim Gebbie

15 February, 2018

Contents

Abstract

Contemporary research into the cross-sectional variation in stock returns is fraught with replication challenges. This makes it difficult to validate important results and hampers the advance of knowledge in the field. This project addresses some of these challenges by fitting the backtesting research process to a data scientific workflow. By making extensive use of standard open source statistical programming languages the authors provide an environment in which total replication is trivial and common statistical errors are avoided by default.

Keywords

backtest, data workflow, overfitting, historical simulation, replication

---

1 Hypothesis

There are major replicability issues surrounding academic research into the cross-sectional variation in stock returns.

1. These issues can be largely mitigated by adhering to a data-scientific approach to data analysis.
2. Robustness of results in the field of stock return research can be can be significantly improved by accounting for the risk of overfitting as the number of trials increases.
3. The general availability of a *minimally sufficient analysis environment* should make the replication of results in the field more robust, easier to admit into the established knowledge base and speed up iterative knowledge creation through significantly shortening the time required for downstream researchers to build on principal authors' work.

2 Motivation

2.1 Summary of Replication Challenges in Backtesting

Academic research into anomalies in the cross section of stock returns is beset by a replication crisis. A large proportion of the existing anomalies literature yields statistically insignificant findings when replicated.

Errors in analysis can have three causes.

1. Bad data
2. Faulty reasoning
3. Buggy implementation

The objective of a replicator is to verify that the result obtained was not spurious. Accordingly, they would like to verfiy that none of the above errors have occurred. At present, an auditor wishing to replicate a typical study compares their summary statistics to original papers to verify results. If the replication results do not match the original results, the auditor does not have enough information to distinguish between these errors. This means that correct reasoning could be rejected because of bad data or buggy implementation.

Errors arising from code bugs or poor data preprocessing are unforced errors, and their effects can largely be mitigated by the sharing of the full codebase that constitutes the data analysis workflow. This project aims to reduce the likelihood of errors by implementing a freely-available platform for automated data preprocessing and implementation of backtests.

2.2 Benefits the Proposed Implementation

2.2.1 Preprocessing Transparency

A plain-text, procedural codebase that collects, transforms, cleans and prepares data enables replicators to walk through the data flow in a stepwise fashion, verifying that no bugs exist and that best practices are adhered to.

2.2.2 Backtesting Transparency

The availability of live-code documents such as Rmarkdown and Jupyter notebook collapse the analysis and report generation steps of research into a single step. Elsevier-formatted papers can be easily written in these environments in a fashion that allows the embedding of the analysis in the source document. Replicators can work through the source document in a stepwise fashion to ensure that no faulty reasoning or buggy implementation occurs.

2.2.3 Iteration Speed

The existence of a freely available, semi-automated environment means that researchers can use a common baseline for conducting original research. Deviations from that baseline can be easily flagged by using differential analysis on the codebase. Simply put, if the base implementation has no processing errors, no code bugs and provides a baseline of statistical tools for robust results, then a replicator can take that for granted and focus on the deviations from the forked codebase.

2.2.4 Programmatic Replication

If an original paper is written using this codebase, then replication is trivial. A replicator simply verifies the final part of the codebase, where the actual backtest resides, and then runs the code on their local computer using their own raw data. This process is amenable to complete automation, where a replicator can automate the replication process using author-provided source code.

2.2.5 Daemonization

Since the codebase will be written in generally avaiaible free programming languages code can be scheduled and lightly modified to run in an automated fashion. Research can be quickly put into production for out-of-sample live analysis and verification. New knowledge can immediately be put into production by investment teams without fear that reverse-engineering of an academic paper may introduce bugs.

3 Proposed Literature Review

Below we include a short list of well-known papers that will provide the starting point for a comprehensive literature review.

1. A survey of several well-known backtests and the documentation of their workflow

- E. F. Fama and MacBeth (1973)
- E. Fama and French (1992)
- Daniel and Titman (1997)
- Hou, Xue, and Zhang (2017)

2. Issues with reproducibility in science in general and the field in particular

- Stodden et al. (2013)
- J. P. A. Ioannidis (2005)
- Brodeur et al. (2016)
- Harvey and Liu (2014)
- Munafò et al. (2017)

3. Statistical errors in backtests

- Lopez de Prado (2013)
- D. H. Bailey et al. (2014)
- Hou, Xue, and Zhang (2017)

4. General programming best practices.

- Martin (2009)

4 Aims and Objectives

This project is firmly rooted in the meta of finance research. The objective is not to validate whether particular anomalies in the cross-sectional variation of stock returns exists. Rather, it is to outline and implement a system wherein researchers can investigate these questions in a statistically rigorous manner.

1. Survey of current academic backtest methods (see the github dissertation repository)

- A critique of the challenges around replicability because of lack of documentation.

- A critique of the challenges around validity because of poor statistical methods (IS/OOS).
- Discussion on how these challenges can be mitigated using standard data science tools.

2. Documentation of an working demonstration system that mitigates these challenges (see the github code repository).

- Source code along with README documentation of every phase of the project will be released to a public repository under the AGPLv3 license.
- A strong copyleft license was chosen because a fundamental pronciple of this project is the facilitation of replication. Insofar that code sharing is necessary for replication, any researcher that uses this code should make their derivative work avaialable for replication.

3. An original replication case study which makes use of the demonstration system to replicate a widely cited academic paper in the field (see the github replication example).

5 Data Requirements Specification

All stages of the data collection and transformation will be transparently documented and source code will be made avaialble on a GitHub hosted repository.

All raw data will be programmatically extracted from a Bloomberg Terminal using the Excel Add-In. Validation of this data will be conducted by randomly selecting ten financial reports from the universe and cross-checking the contents of those financials against the raw data.

All data cleaning and interpolation will be done using the dplyr and tidyr packages in the R statistical computing language.

6 Systems Requirements Specification

6.1 Hardware Requirements

- A computer running an x86 processor
- 50gb of hard drive space
- At least 8gb or RAM

6.2 Software Requirements and Packages Used

- Access to a Bloomberg Terminal with Excel installed
- A research machine running
  - Ubuntu 16.04 LTS
  - Jupyter Server
  - RStudio Server
  - Python 3.5
  - R 3.4.3
  - Nextcloud 12 or later for transferring data from the Bloomberg Terminal to the research ma

6.3 Software Development Framework, Configuration Control and Version Control

The primary objective of this dissertation is the creation of a script-based workflow that improves the equity backtesting research process. The complete codebase, as well as the dissertation and demonstration case will be made avaialble on a GitHub repository. The code will be released under an AGPLv3 license.

The data will not be released due to data vendor licensing constraints. However, macro-enabled Excel workbooks that programmatically query the Bloomberg Excel Add-In for relevant data will be made available in the public repository.

Version control of all project deliverables will be managed using the Git version control system. Commits will be pushed regularly to the publically available GiHub repository to ensure timeous backups of the working paper and codebase.

7 Project Milestone Deliverables

Date	Milestone	Status
October 2017	Set up server with necessary dependencies	Complete
November 2017	Build Bloomberg Excel VBA Workbook to scrape data	Complete
December 2018	First Pass Literature Review	In Progess
January 2018	Scrape Bloomberg terminal for data	Complete
January 2018	Merge raw files into single csv files	Complete
February 2018	Document Codebase to Date	In Progress
February 2018	Write Project Proposal	In Progress
February 2018	Register for Academic Year	In Progress
February 2018	Clean, join and interpolate data	In Progress
February 2018	Transform raw data into Sqlite file	In Progress
March 2018	Build out code to control for backtesting biases (look-ahead, survivorship etc)	Not Started
April 2018	Perform a case study backtest	Not Started
May 2018	Debug, refactor, refine	Not Started
May 2018	Re-document Codebase to Date	Not Started
June 2018	Add additional data sources: iNet, Datastream	Not Started
July 2018	Replicate case study backtest on alternative data and compare differential results	Not Started
August 2018	Create second backtest and document workflow steps and benchmark timing	Not Started
August 2018	Wrap Code Documentation, proposal and findings into dissertation	Not Started

References

Bailey, David H., Jonathan M. Borwein, Marcos López de Prado, and Qiji Jim Zhu. 2014. "Pseudo-Mathematics and Financial Charlatanism: The Effects of Backtest Overfitting on Out-of-Sample Performance." *Notices of the AMS* 61 (5): 458–71. doi:10.2139/ssrn.2308659.

Brodeur, Abel, Mathias Lé, Marc Sangnier, and Yanos Zylberberg. 2016. "Star Wars: The Empirics Strike Back." *American Economic Journal: Applied Economics* 8 (1): 1–32. doi:10.1257/app.20150044.

Daniel, Kent, and Sheridan Titman. 1997. "Evidence on the Characteristics of Cross Sectional

Variation in Stock Returns." *The Journal of Finance* 52 (1): 1. doi:10.2307/2329554.

Fama, E., and K. French. 1992. "The Cross-Section of Expected Stock Returns." doi:10.2307/2329112.

Fama, Eugene F., and James D. MacBeth. 1973. "Risk, Return, and Equilibrium: Empirical Tests." *Journal of Political Economy* 81. The University of Chicago Press: 607–36. doi:10.2307/1831028.

Harvey, Campbell R, and Yan Liu. 2014. "Evaluating Trading Strategies." *The Journal of Portfolio Management* 40 (5): 108–18. doi:10.3905/jpm.2014.40.5.108.

Hou, Kewei, Chen Xue, and Lu Zhang. 2017. "Replicating anomalies." *NBER Working Papers*, no. No. 23394. doi:10.2139/ssrn.2190976.

Ioannidis, John P. A. 2005. "Why Most Published Research Findings Are False." *PLoS Medicine* 2 (8). Public Library of Science: e124. doi:10.1371/journal.pmed.0020124.

Lopez de Prado, Marcos. 2013. "The Probability of Back-Test Over-Fitting." *SSRN Electronic Journal*. doi:10.2139/ssrn.2308682.

Martin, Robert C. 2009. *Clean Code: A Handbook of Agile Software Craftsmanship*. First. Boston: Prentice Hall.

Munafò, Marcus R, Brian A Nosek, Dorothy V.M. Bishop, Katherine S Button, Christopher D Chambers, Nathalie Percie Du Sert, Uri Simonsohn, Eric Jan Wagenmakers, Jennifer J Ware, and John P.A. Ioannidis. 2017. "A manifesto for reproducible science." doi:10.1038/s41562-016-0021.

Stodden, V, D H Bailey, J Borwein, R J Leveque, W Rider, and W Stein. 2013. "Setting the Default to Reproducible Reproducibility in Computational and Experimental Mathematics." In *ICERM Workshop*, 19. http://www.davidhbailey.com/dhbpapers/icerm-report.pdf.