A Data Scientific Approach to Equity Backtesting

Riaz Arbi[a]

[a] *University of Cape Town, South Africa*

Abstract

Abstract to be written here. The abstract should not be too long and should provide the reader with a good understanding what you are writing about. Academic papers are not like novels where you keep the reader in suspense. To be effective in getting others to read your paper, be as open and concise about your findings here as possible. Ideally, upon reading your abstract, the reader should feel he / she must read your paper in entirety.

Keywords: Equity Backtesting, Reproducible Research, Event-based Backtesting, R, RStudio

JEL classification

Table of Contents

Email address: (Riaz Arbi)

1. Introduction

Intro intro intro

Last paragraph

This paper is organised as follows. Section 2 briefly introduces the practice of backtesting of investment strategies in academia and industry, discusses the replication crisis in the academic literature, and enumerates common methodological issues that compromise the validity of research. Section 3 describes the goals and philosophy of the reproducible research movement and its links to the open source software movement, with a special focus on the R statistical programming language, the Tidy approach to data organization and the use of scripting to facilitate reprducibility and transparency in scientific research. Section 4 introduces a backtesting template that is roughly compatible with the structure of a research compendium - that is, a basic minimum viable unit of reproducible research.

2. Anomalies Research in Theory and Practice

Investigations into why the returns of some common stocks outperform others have a long history. Professional bodies of knowledge, often captured in heuristics or rules of thumb, have existed since at least the early 20th century. (**???**) collected the body of knowledge of one approach, value investing, into a single textbook in 1934.

As time has progressed, professionals and academics have sought to test these heuristics in a rigorous manner. A standard way to test whether a particular heuristic or rule of thumb may drive returns is to investigate whether an investor that followed the rule in the past would have significantly different results from an investor that did not. This process is known as backtesting. Several foundational papers of investment management (for instance, (**???**), (**???**)) use backtesting methods as a basis for their findings.

Unfortunately, backtesting is a very difficult operation to achieve correctly. A backtester has to ensure that their dataset is complete and free of error, which is difficult to know after the fact. For instance, the exact date on which the financial results of a company were released may not be known - end-of-year results are never released on the 31st of December, since those financials need to be computed and audited after the fact prior to release. The actual release date may be months later, and if the information contained in the financials drive returns, the timing of the impact of the release on the market be simply be unknowable. This problem (which is just an example of many) can be dealt with in a variety of ways, all of which require a judgment call from the researcher.

It is standard practice for a researcher to disclose all the judgment calls made during the backtest

process, but these disclosures are invariably incomplete. This makes validation by other researchers very difficult. At the extreme, independent validators may arrive at a completely different conclusion (Hou, Xue, and Zhang (2017)); with ambiguity on both sides it is unclear whether the discrepancy is due to differences in underlying data, prepearation methodologies, mathematical errors or some other factor unrelated to the question at hand.

The academic discipline of anomalies research (as this area is now known) has grown into a substantial corpus of contradictory claims, which have been mirrored in the business of investment management by competing investment philosopies, each with its own library of peer-reviewed papers ((**???**)). While markets may very well accommodate contradictory drivers of returns it is often difficult to discern whether differing claims are the result of geniune facts or the result of methodological differences.

Issues with anamalies research The relationship between the issues and difficulty replicating

2.1. Characteristics of an Academic Equity Backtest

Basically rip the guts out of the Replicating Anomalies paper - Poor documentaiton of data collection and munging - Ambiguity around actual operations done to organise, clean and drop data - No verifiability of error - Unable to detect p-hacking - Biases: look-ahead, survivorship, data mining - No transparency (replication, verifiability) - Backtest overfitting - de Prado, Bailey et al

2.2. Commercial and Open Source Backtesting Software Packages

- Survey: Proprietary, free web, free open source
- Prop are out because not transparent
- Open Source: See zipline, qstrader,
- Poor data provenance (yahoo)
- Absence of fundamental data
- Built for pairs and single stocks, not portfolios
- OOP makes transparency difficult (black box)
- Presupposes good data hygeine & pre-munging

2.3. The Replication Crisis

2.4. Common Biases and Methodological Errors

3. Reproducible Research and The R Programming Language

In recent years, as computational power and storage has become increasingly cheap and available, there have been persistent calls for a review of the way that scientific reseach is packaged and presented (Stodden et al. (2013), Koenker and Zeileis (2009)).

Although scientific research is an accretive process, involving verification and building upon earlier datasets and findings, the computational work done by researchers is often poorly documented, or absent (Stodden et al. (2013)). This introduces the risk that views or principles that are considered to be scientifically verified may, in fact, be false.

Gaps in documentation allow bad reasoning, calculation errors or spurious results to creep in to the corpus of knowledge of a discipline because peers are forced to take it on faith that a researcher has a proper understanding of underlying mathematical concepts, and that workings have been thouroughly cross-checked before release.

Worse still, because follow-up replication is difficult to impossible with poor documentation or data availability, incorrect beliefs can persist for years, or even decades (Munafò et al. (2017)). *P-hacking*, or the selection of data and finessing of results to conform to significance tests of 5% or lower, has resulted in a situation where a research finding is as likely to to be false it is to be true (J. P. A. Ioannidis (2005)).

3.1. Reproducible Research Terminology

In response to these critiques, a taxonomy of reproducibility has emerged, allowing peers to effectively categorize research.

This taxonomy allows us to dinstinguish reveiwability (which assumes mathematical competence and focuses on reasoning) from reproducibility (which allow the extension of the review into assessement of mathematical competence and quality of data).

Table 3.1: Common Terminology in Reproducible Science (Stodden et al. (2013))

Term	Description
Confirmable	Main conclusions can be attained independently
Reviewable	Descriptions of methods can be independently assessed and judged

Term	Description
Replicable	Tools are made available that would allow duplication of results
Auditable	Sufficient records exist (perhaps privately) to defend research
Reproducable	Auditable research is made openly available, including code and data
To Verify	To check that computer code correctly performs the intended operation
To Validate	To check that the results of a computation agree with observations

This paper uses reproducibility in the sense defined in 3.1, which is to say, it refers to research output being bundled and distributed with well documented, fully transparent code and data that allows a peer to fully review, replicate, audit and thereby confirm results of a body of work (Stodden et al. (2013)).

3.2. Distinguising Between Open Source ands Closed Source Software

In general, software is written in text files, by humans, in a particular programming language. This *source code* is *compiled* or *interpreted* by another program into binary machine code (which is not readable by humans) and distributed for execution. The software for the popular spreasheet program Microsoft Excel, for instance, would be written by a group of humans in a programming language. These text files would be compiled by a compiler program into a binary executable program, which is distributed to consumers who can execute the binary. When someone clicks the Excel icon on their computer to 'launch' the program, they are executing the binary.

Open source software is software for which the *source code* is made publically available. Microsoft Excel is closed source, and the source code is not made available. Source code can be read by humans, binary cannot. Changes to source code can be meaningfully interpreted by humans, because one can read the changes in plain text. Changes to binary code cannot be interpreted by humans, because the changes are simply alterations to a very long sequence of `0` and `1` digits. It is trivial to compile source code into binary code. It is extremely difficult to accurately decompile binary code into source code, and the tools are not readilty available ((**???**)).

3.3. The Relationship between Reproducible Research and Open Source Software

Open source software development principles exhibit properties which are useful to practitioners of reproducible research.

By definition, the text-based nature of source code makes all source code reproducible (and, by corollary, auditable). The implications of this are twofold. Firstly, a researcher using open source software for computation can be assured that the full software stack will be open to scrutiny. That is, at the limit, an auditor could inspect every single line of code from some arbitrary 'open starting point' up to verify that the computations are correct. This 'open starting point' is as the operating systme level for linux based machines, and from the application level up for Mac and Windows based machines.

Secondly, and perhaps more importantly, the open source software community has decades of experience in defining good practices for creating and maintaining large, open, verfiable and repeatable enviroments. These principles are often quoted in the folkloric 'Unix Way', which manifests in epithets such as "text is the universal interface" (Baum and Sirin (2002)), "each unit should do one thing and do it well", and "build things to talk to other things" ((**???**)). Reproducible research practice is built on the same principles - simple, readable text-based data and code where possible; clear separation between data, code and results; complete transparency coverage; and portability by design (see Baum and Sirin (2002), Bache and Wickham (2014), Wickham (2014)).

Historically, there has been close collaboration between the open source software and the free software communities. This means that, as a rule, most open source software is free or has a free analogue. The core tools of Data Science, for instance (such as the python and R programming languages, the Apache Foundation of big data processing packages and the Jupyter and RStudio interactive development environemnts), are all open source and free. This means that the *cost* of replication is not prohibitive: in general, reproducible research should be reproducible for free on commodity hardware.

3.4. The Suitability of R for Reproducible Research

The R programming language was originally conceived as a project to build out an open source statistical programming environment. In development since 1997, base R is now a mature ecosystem comprising a scriptable language, graphical rendering system and debugger (R Core Team (2018)). A large community of third party developers (some commercial, most free) have extended base R with over 13000 packages, including a fully-fledged interactive development environmnent (Rstudio), interactive dashboarding web server (Shiny) and bindings to LaTex for simple rendering of publication-ready LaTex documents from markdown (knitr and rmarkdown) (RStudio Team (2015)). Because of its roots in open source, it adheres to many open source software conventions. For instance-

- All source code is freely available
- All development work in R is done in plain text files, which are transparently human and machine readable.

- The R project has adopted the GNU General Public License verison 2, which "does not restrict anyone from making use of the program in a specific field of endeavor" (R Core Team (2018)). This free avaialability means that anybody with a commodity computer and an internet connection can install and use R.

Although the reproducible research community is principles-based and language-agnostic, the free, text-based nature of the R language have made it a good candidate for reproducible work ((**???**)). Marwick, Boettiger, and Mullen (2018) review the concept of a *research compendium*, and explore how research done in R can be packaged into compendia. They argue that a compendium is defined by three principles - 1. Files should be organised according to a prevailing standard so that others can immediately understand the structure of the project, and pipe the compendium to environments that expect the standard structure with no structural modifications. 2. Maintain a clear separation between data, method and output. Separation means that data is treated as read only, and that all steps in getting from source data to output is documented. Output should be treated as disposable, and rebuildable from the programmatic application of the method to the data. 3. Specify the computational environment that was used to conduct the research. This provides critical information to a replicator on the undelrying tooling needed to support the analysis.

These principles are intended to ensure that, given a certain input (the data) and a certain operation (the method), a deterministic output results (the analysis). With these details out of the way, an auditor can focus on verifying the correctness of the reasoning (i.e should we use this theory) and of the implementation (i.e are there any errors in the math).

The richness of the R ecosystem means that one can complete the full research life cycle from data loading to publishing without leaving the R ecosystem ((**???**)). Data can be loaded, cleaned, transformed, modeled and aggregated in R. The results can be written up in RStudio - in fact, the code and write-up can be combined into a single text-only document using the RMarkdown format - and exported to a wide range of academically accepted typsets and formats (such as Tex, Microsoft Word, Markdown and HTML). Because R is scriptable, R scripts can also be used in production environments, where results can be periodically recomputed and handed on to some other production process.

This property dramatically simplifies the complexity of building a compendium and the required skillset of a replicator. For instance, a single-platform research compendium can be assessed by anyone with knowledge of that single environment. Multi-platform compendia need replicators well-versed in each platform, which dramatically reduces the pool of potential replicators.

Upon publication, code versioning tools such as Git allow the public to view all future changes to the code or data transparently, and, if code changes over time, to assess the impact of those changes on the result.

3.5. Tidy Extensions to Base R

The open source and reproducible science principles of interoperability, readibility, modular code and common standards have been extended into the prcatice of data manipulation by Wickham (2014). The collection of R packages which fall under this banner are collectively known as the 'Tidyverse'. These packages provide a common, idiosyncratic syntax for data ingestion, munging, manipulation and visualization that are opinionated in their expectations about data structure and focused on enabling readable code (Wickham (2017)). According to Wickham, tidy data intuitively maps the meaning of a dataset to its structure -

1. Each variable forms a column.
2. Each observation forms a row.
3. Each type of observational unit forms a table.

Whether there is a single optimal data structure is up for debate. What is not up for debate is that that the adoption of a common standard for structures allows researchers to make assumptions about the meaning of data before engaging with the content, i.e simply knowing that data is 'tidy' means that a researcher knows that each column maps to a variable, and each row an observation, *prior to looking at the data*. Setting of standards enables the decoupling of processes - process (a) can produce some peice of analytical output *with no knowledge of what it will be used for*. As long as it is 'tidy', and process can consume that output on the understanding that is confirms to the 'tidy' standard and take the analysis further. In this way, the 'tidy data' concept push the Unix Way concept of 'build things that talk to other things' into the practice of structuring data for reproducible research.

3.6. The Potential for Reproducible Research to Address The Replication Crisis in Anomalies Research

- An argument that these approaches and tools will improve backtesting, both for academics and for practitioners
- A proviso that none of this replaces critical thinking or good data hygeine. Still need to get data, scrutinize it, make judgment calls etc etc.
- Reference the definition of a compendium

4. A Reference Implementation of a Data Scientific Equity Backtester in R

The problem of proprietary data, the use of the log, and the transformation into tidy data

Restructure code along the ines of a compendium Packaging Data analytical Work Reproducibly Using R (and Friends)

4.1. My documentation goes here

4.2. Features

- data versioning
- multi-source
- multi-index
- code version control
- clear procedural workflow
- event-driven
- tidy data
- pbo
- host of metrics
- bias avoidance

5. Conclusion

This paper has explored the various ways in which the lack of transparency in anomalies research makes it difficult to discern spurious results from geniune findings. This 'replication crisis' has strong analogues in other academic disciplines. We have argued that the 'reproducible science' response to this crisis in science at large has potential to address much of the issues that bedevil anomalies replication.

We have introduced a collection of R scripts, organised into a compendium, that can be used to conduct anomalies research in a transparent and reproducible way. These scripts utilize only free, open source software and to organize data along the lines of 'tidy' data. Using plain text computer code to collect, process, structure and analyse data represents a good approach to producing research that is easy to reproduce.

We avoid the problem of proprietary data distribution by including customizable data query scripts. These scripts query proprietary vendors and save the results in a standard way. Bundling these query scripts in a compendium enables a replicator to rebuild the dataset programatically and non-interactively from source.

This collection of scripts make it possible to test an investment algorithm against an index of stocks, where each stock comprises a set of daily observations of price data plus an arbitrary number of attributes. The scripts use the event-based backtest method (as opposed to vectorized methods) which make it easy to avoid look-ahead bias and to introduce non-standard data to the algorithm. Transaction cost and slippage modelling, while rudimentary, exist and can be refined.

Using this code base as a starting point should save a research a great deal of time in preparing stock data for backtesting, and the open source nature of the project ensures that any researcher can comb the operations for bugs or implement features that are not present. While it is not expected that this code base is necessary or sufficient for end-to-end backtesting, it represents a solid base for ongoing development.

Katzke (2017)

References

Bache, Stefan Milton, and Hadley Wickham. 2014. "magrittr: A Forward-Pipe Operator for R." https://cran.r-project.org/package=magrittr.

Baum, Christopher F., and Selcuk Sirin. 2002. "Why should you avoid using point-and-click method in statistical software packages?" http://fmwww.bc.edu/GStat/docs/pointclick.html.

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.

Katzke, N F. 2017. "Texevier: Package to create Elsevier templates for Rmarkdown." Stellenbosch, South Africa.

Koenker, Roger, and Achim Zeileis. 2009. "On reproducible econometric research." *Journal of Applied Econometrics*. doi:10.1002/jae.1083.

Marwick, Ben, Carl Boettiger, and Lincoln Mullen. 2018. "Packaging Data Analytical Work Reproducibly Using R (and Friends)." *American Statistician*. doi:10.1080/00031305.2017.1375986.

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.

R Core Team. 2018. “R: A Language and Environment for Statistical Computing.” Vienna, Austria. https://www.r-project.org/.

RStudio Team. 2015. “RStudio: Integrated Development Environment for R.” Boston, MA. http://www.rstudio.com/.

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.

Wickham, Hadley. 2014. “Tidy Data.” *Journal of Statistical Software*. doi:10.18637/jss.v059.i10.

———. 2017. “tidyverse: Easily Install and Load the ’Tidyverse’.” https://cran.r-project.org/package=tidyverse.