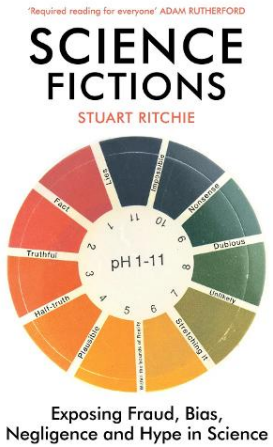
Making your work replicable

James Brusey

25 March 2022

Motivation

- There is a replication crisis in science today
- There are plenty of incentives to publish more papers
- There are few incentives to publish better papers
- Penalties for negligence, bias, and fraud are minor even if offenders are found out



Our institutions are implicated

- Publishers charge outrageous fees for essentially running a web / archive service
- Universities continue to promote staff based on questionable metrics
- Professors teach their PhD students to continue to 'game' the system
- Reviewers reject replication studies as not 'sufficiently novel'
- Academics collude in citation cartels to bump up each others citation ranking

The public are starting not to trust academics

- Surprisingly, academics are still respected
- Public trust is not a given and should not be taken for granted
- We (academics) have a responsibility to fix things
  - let's start by improving replicability of our research

Ideas for improving replicability

These ideas are focused on the *analysis* rather than the experimental work itself.

Please stop using Word and Excel

- An old version of Excel caused a statistical analysis error during the Covid pandemic
  - but why were they using Excel?
- An analysis of genomics research shows that many studies have fallen prey to MARCH1 gene being altered by autocorrect in Excel
- There are many reasons why you should not use Word but the number one reason is that it will stop you from automating parts of your research—you will tend to be relying on cutting and pasting in figures and tables rather than auto-generating them. *Convert away before it is too late.*

Use the command line and GNU Make

- Analysis ends up having several steps
  - combining multiple data-sets into one
  - cleaning up NA entries
  - removing junk entries
  - summarising data to produce a table
  - producing a graph

Method for using Make

- Each step should be performed with a command or script (e.g., gnuplot)
- Form multiple steps into a pipeline with GNU Make
- Alongside much on-line sources, also see Data Science at the Command Line `https://datascienceatthecommandline.com/`
- Python tabulate library can be used to convert a CSV to a LaTeX table.
- In your LaTeX file, use `\input` to include those files

Example—generating data

For example, say we have a script to generate some data `a.csv`, `b.csv`, `c.csv` called `gen.py`

```
import pandas as pd
import numpy as np

SZ=(20,)

df = pd.DataFrame(np.random.randint(0, 10, size=SZ), columns=["value"])
df.to_csv("a.csv", index=False)
df = pd.DataFrame(np.random.normal(0, 1, size=SZ), columns=["value"])
df.to_csv("b.csv", index=False)
df = pd.DataFrame(np.random.normal(5, 3, size=SZ), columns=["value"])
df.to_csv("c.csv", index=False)
```

Example—combine data

We might then have another script `comb.py` to combine them.

```
import pandas as pd
import numpy as np

newframe = {}
for f in ["a", "b", "c"]:
    newframe[f] = pd.read_csv(f"{f}.csv")["value"]

df = pd.DataFrame(newframe)
df.to_csv("all.csv", index=False)
```

Example—table

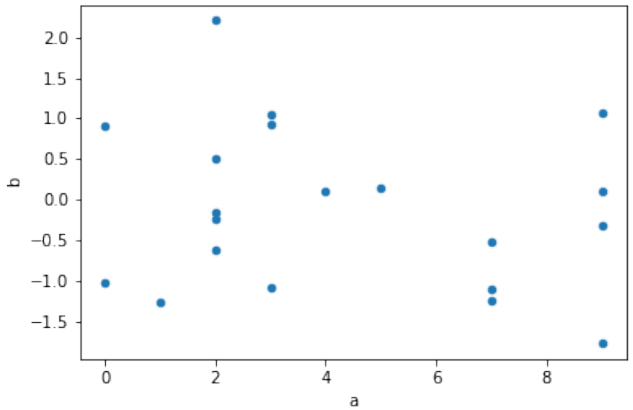
We can produce a table using python tabulate in a script called `maketable.py`

```
import pandas as pd
import numpy as np
import tabulate

df = pd.read_csv("all.csv")
result = pd.melt(df).groupby("variable").agg(["mean", "std"])

with open("result.tex", "w") as f:
    print(
        tabulate.tabulate(result, tablefmt="latex", headers=["Class", "mean",
        ↪  "std"]),
        file=f,
    )
```

Example—graph

Finally, we might use `graph.py` to plot a versus `b` (ok, this is not a very meaningful graph!)

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
df = pd.read_csv("all.csv")
df.plot(x="a", y="b", kind="scatter")
plt.savefig("graph.png")
```



Example—LaTeX doc

Naturally, we need a LaTeX document:

```
\documentclass{article}
\usepackage{siunitx}
\usepackage{graphicx}
\title{My great article}
\author{James Brusey}
\begin{document}
\maketitle
\section{Introduction}
Blah blah blah.
\section{Results}
\input{result}
\begin{figure}
  \includegraphics{graph.png}
  \caption{A scatter plot}
\end{figure}
\end{document}
```

Example—Makefile

Finally, we tie everything together with a `Makefile`

```
article.pdf: article.tex result.tex graph.png
	pdflatex article.tex

graph.png: all.csv graph.py
	python graph.py

result.tex: all.csv maketable.py
	python maketable.py

all.csv: a.csv b.csv c.csv comb.py
	python comb.py

a.csv: gen.py
	python gen.py
```

Using RStudio

- RStudio allows you to put all the steps into a notebook form
- The result can be exported to a LaTeX document
- Best for R but difficult to format for a paper
- A great resource for R and the tidyverse is R for Data Science `https://r4ds.had.co.nz/`
- You can also use Pandoc separately from RStudio

RStudio example

```
---
title: "Example rmarkdown document"
date: "24/03/2022"
author:
- James Brusey
- Ann Other Author
documentclass: scrartcl
classoption: twoside
geometry: false
subtitle: false
output:
  pdf_document:
    includes:
      in_header: header.tex
---

# Introduction

This is a sample markdown document.

I can make a new paragraph using a blank line and a numbered list just with:
1. this item
2. this other item
3. and so forth
```

Use Jupyter Notebook

- Jupyter notebook supports Python and several other languages
- As with Rstudio, can produce LaTeX by combining code, graphs, and markdown
- There are no easy options for changing the document class, so not really a viable option for writing papers

Use Emacs org-mode

- Org mode is a powerful editing environment that comes with Emacs
- Org mode documents are similar to Rmarkdown (or pandoc) with easy formatting instructions
- More flexible than Rmarkdown
- Easy to change document class
- Can include many different programming languages within the one document

Further reading

1. I thoroughly recommend Science Fictions Ritchie [2020]
2. John Kitchin has a nice article on embedding data into PDFs. Kitchin [2015]
3. He also has a youtube describing org mode for research `https://youtu.be/1-dUkyn_fZA`

John R. Kitchin. Examples of Effective Data Sharing in Scientific Publishing. *ACS Catal.*, 5(6):3894–3899, June 2015. doi: 10.1021/acscatal.5b00538. URL `https://doi.org/10.1021/acscatal.5b00538`.

Stuart Ritchie. *Science Fictions: Exposing Fraud, Bias, Negligence and Hype in Science*. Random House, 2020.