Notes

packrat

Reproducible package manamement for R. Makes your R projects reproducible by making them

- **isolated**: installing other packages won't break your project
- **portable**: easy to share , even cross platform
- **reproducible**: because it records the exact versions of the packages your project depends on

Each project has its own package library.

Funny story: so I've known about packrat for a few years now. I've known what it does, and it made all the sense in the world. And yet I didn't use it. God knows what my problem is, but it was only last month, when I bought a new laptop, installed a new and different OS than my previous one and wanted to start working on a project I was working on in the spring, when it suddenly didn't work anymore. It was a big project with dozens of interconnected files and several dozen packages, and it was now broken. Now, I had already returned my old laptop to my employer so I had no way of recreating the package environment that I had there. And no easy way of figuring out which of the packages was the offending package that had been updated in the meantime.

And it made me realise how incredibly lucky I had been that the client—this was a consulting gig—the client was able to run the code on their systems without any problems, presumably because theey had the same outdated libraries installed as I had had on my old laptop. I was terribly lucky, but I learned my lesson anyway. So from then on I use packrat, actually preparing this talk was the first time I used it on a new project.

But I wanted to bring up this story also to demonstrate that it is often a slog, this reproducibility stuff. It's upfront work, and it's easy to put it off. Although in the end it turned out that packrat is ridiculously easy to setup, it is literally just one line of code, and if you're initializing it on an existing large project you'll have time to make yourself a quick cup of tea while it creates the snapshot of the working library environment, but I was quite embarassed once I realised I could have been doing this for years.. Instead I had to first have it go wrong and have the code break, before I came to my senses and started using this tool. It's normal, it's human. But it's also stupid. So don't be stupid like I was, it's a lot less painful to get on the reproduciblility bandwagon of your own accord than learning the hard way.

(Hong et al. 2015)

additional details

In addition to software there are other environmental variables that should be explicitly stated if they affect the results: " This information may include: operating system, specific packages, sub-routines, queries, files, libraries, scripts, service end-points, configurations, parameters or workflows" "this information should be described in the body of your paper, in the methods section, footnotes, acknowledgements or appendices.

style guide

text

References

Hong, Neil P Chue, Tom Crick, Ian P Gent, Lars Kotthoff, and Kenji Takeda. 2015. "Top Tips to Make Your Research Irreproducible." *arXiv Preprint arXiv:1504.00062*.

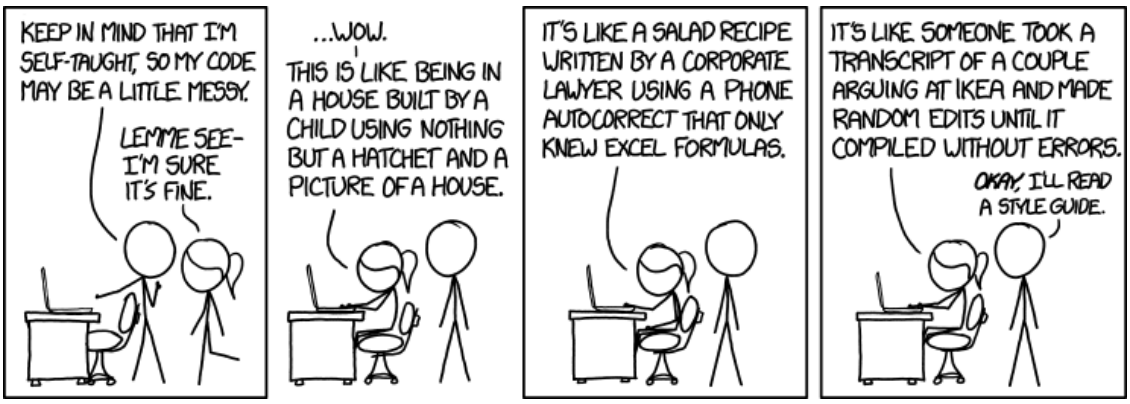

Figure 1: [xkcd: Code Quality [https://xkcd.com/1513/]