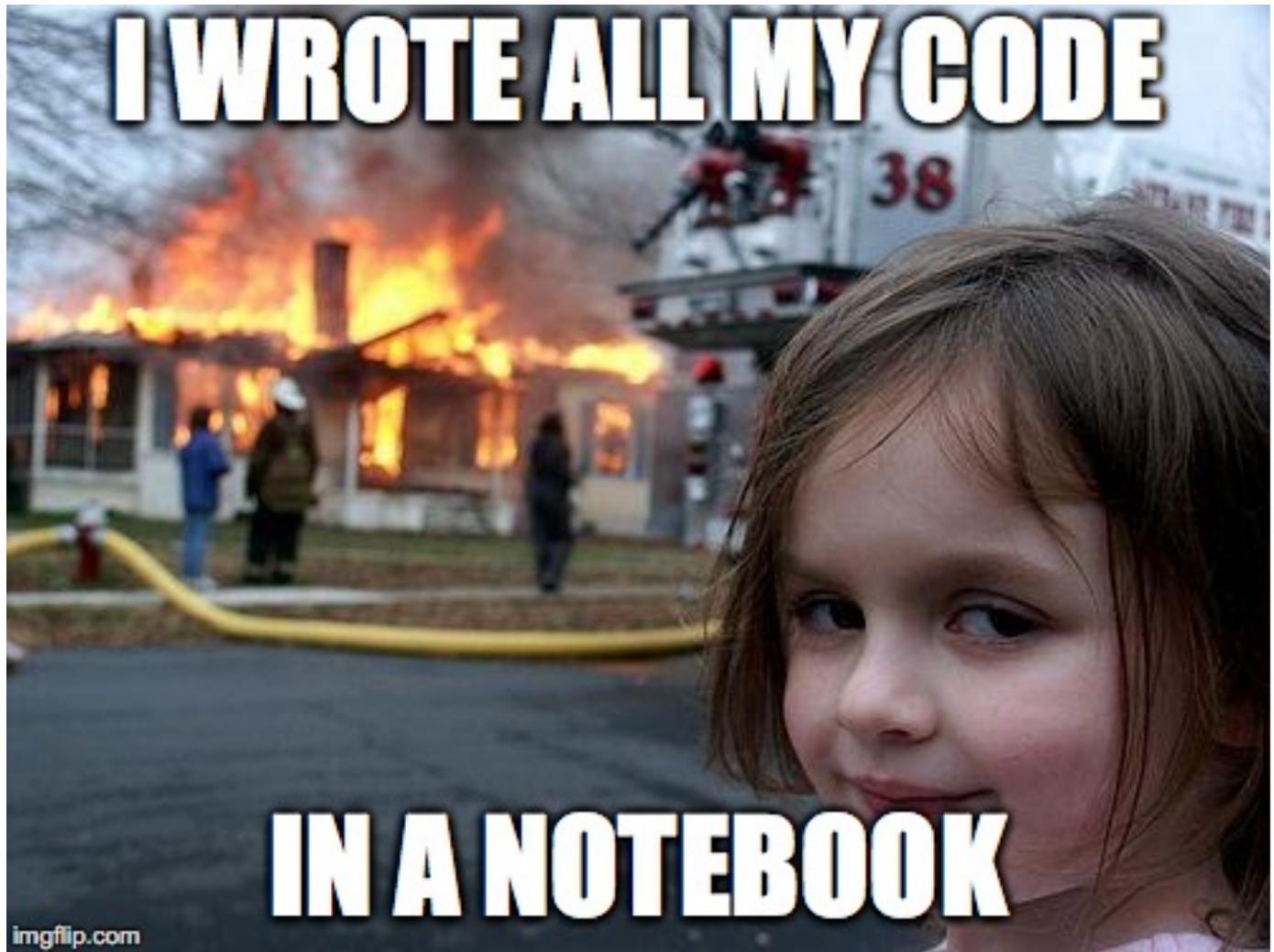


Terry's notebook (<https://civnb.info/terry/notebooks/private/>)

Tue 23 April 2024

20240423 anti-Jupyter talk (<https://civnb.info/terry/notebooks/private/20240423-anti-jupyter-talk.html>)

Posted by Terry Jones (<https://civnb.info/terry/notebooks/private/author/terry-jones.html>) in Rants (<https://civnb.info/terry/notebooks/private/category/rants.html>)



From these slides (https://docs.google.com/presentation/d/1n2RIMdmv1p25Xy5thJUhkKGvjtV-dkAIsUXP-AL4ffI/preview#slide=id.g3a428e2eb8_0_255).

Overall summary

I think using Jupyter (by itself) to try to do scientific / research computing is a *really bad idea*. And I'm really serious.

What do you gain?

You can quickly see things, especially graphics, on the screen without having to switch apps.

What do you lose?

But computational notebooks can also be confusing and foster poor coding practices. And they are difficult to share, collaborate on and reproduce. A 2019 study found that just 24% of 863,878 publicly available Jupyter notebooks on GitHub could be successfully re-executed, and only 4% produced the same results (J. F. Pimentel et al. in 2019 IEEE/ACM 16th International Conference on Mining Software Repositories (MSR) 507–517; IEEE, 2019).

From this Nature article (<https://www.nature.com/articles/d41586-021-01174-w>).

You don't have a proper editor. Can't search and replace, can't search properly, don't have all your editing skills and keystrokes, etc.

No one, including you, can import your code so it can be used elsewhere.

How to do a review on GitHub when a `.ipynb` file is all JSON?

It's very hard to write tests. Did anyone in the group ever actually do that? I doubt it. See testbook (<https://github.com/nteract/testbook>).

You can't easily grep your code.

You can't run a Python linter or formatter.

It incentivizes putting everything into one jumbled pile. This runs contrary to good practice and is like a throwback to a more primitive time. See ploobmer (<https://github.com/ploomber/ploomber>).

You can't put things in one notebook and import them in another. What if you write some code that you want in two places?

Cells are not automatically re-run, so it is easy to get into a broken state.

It incentivizes global variables, which is also contrary to best practice.

It is very easy to end up with a notebook that does not fully run or that when re-run does something different from the previous time you ran it (e.g., because you have changed a global variable).

You can't make readable diffs if you put a notebook into a version control system. But see nbdime (<https://github.com/jupyter/nbdime>).

Two people can't easily edit the same Jupyter file and have their edits resolved. If there is a conflict you have to resolve the Jupyter JSON file mess yourself.

You don't have the Python debugger.

Thoughts about what's going on here

Five steps back in order to take one forward. Then, try to recover all the lost things.

Reminds me a bit of the HTML-in-email pains of the 90s.

We already had ipython. What was the problem? Why go to all the effort to put python into the browser and make ipython-notebook? It's purely about aesthetics (presentation).

The various tools that have been developed to mitigate some of the above issues are all efforts to get us back to what we already had (e.g., regular diff, tests, code organized into several files, seeing filesystem changes).

Jupyter notebooks have a gravitational pull. If you start using them, you tend to keep using them and putting more and more stuff into the notebook. That makes it increasingly harder for you to get out of that world. And it also exerts a pull on other people - they need to start a notebook to see or run your code.

Having everything in one place is an advantage but also a disadvantage.

A compromise?

Get things working in a notebook, then save code to files and import it.

What I did to run the sars-2-anova notebook cleanly and to configure where its outputs went and to test them.

Make it possible to run the whole notebook from the command line. See papermill (<https://github.com/nteract/papermill>).

Site built using Pelican (<http://getpelican.com/>) • Theme based on VoidyBootstrap (<http://www.voidynullness.net/page/voidy-bootstrap-pelican-theme/>) by RKI (<http://www.robertiwancz.com/>)