Standards with Jupyter

In the original grant proposal for Project Jupyter, written by Brian Granger and Fernando Perez, the founders explain that "the core problem [they were] trying to solve is *the collaborative creation of reproducible computational narratives that can be used across a wide range of audiences and contexts*". Let's break that down into the main pieces that define a notebook with those standards.

1. A notebook should be collaborative.
2. A notebook should be reproducible.
3. A notebook should be a computational narrative.

Collaborative

I define a repository as *collaborative* if it has any forks and/or open issues, meaning people are actively working on the same repository or suggesting changes to the owner. A repository is *watched* if it is not collaborative, but has at least one stargazer or watcher. Even if individuals aren't collaborating on these repositories, they are viewed by other GitHub users. An *isolated* repository is neither collaborative nor watched; no one besides the owner is working on or viewing the repository.

With these definitions, I found that 17.5% of repositories are collaborative, 41.3% are watched, and 41.2% are isolated. On average, a collaborative repository has 11.76 forks (median = 1) and 1.98 issues (median = 0).

There is a higher ratio of words of markdown to lines of code among collaborative notebooks (mean = 8.2), followed by watched notebooks (6.3) and isolated notebooks (5.7). Further, collaborative and in sight repositories are more likely to have a description than an isolated repository (58% compared to 48%). The more collaborative a repository is, the larger it tends to be and the more notebooks it tends to hold.

Reproducible

A notebook is most reproducible if it can be run top to bottom without having to edit code or the order of cells. Some detectible things that impede the user's ability to do this include errors, functions or variables used before they're used, packages used before they're imported, and attempts to access local files.

More detail on these calculations can be found in *Analyzing 4 Million Jupyter Notebooks*, but I'll reiterate the results here. 5.54% of notebooks that could be parsed with the Python abstract syntax tree call user-defined functions before the function definition, 2.72% use variables before defining them, and 1.12% use packages before importing them. In *API Design*, I found that 11.77% of notebooks have at least one error.

If the owner ran cells out of order, it is uncertain whether errors would occur when running top to bottom (though the notebook is not necessarily unreproducible). I calculated that 29% of notebooks with at least one cell with output were run out of order. Errors tend to be more frequent in these notebooks, supporting the idea that cell execution order can contribute to reproducibility.

For a notebook to be reproducible, the data must be accessible. Sometimes data is provided in the same repository as the notebook, but sometimes it may be created in the notebook or in another file in the directory, downloaded with a bash `!wget` command, or linked in the README, in markdown cells, or in comments. It's hard to check all of these possibilities with the information we have access to, but we can check the code for full paths (in the form `/Users/user_name/...`, `C:/user_name/...`, `/home/...`, or `~/...`). 6.82% of notebooks attempt to access a file in this way.

Hypothetically, if everything is in order, there are no errors, and no full path is used, a notebook should be able to run top to bottom without having to edit or rearrange any code. Given this definition, 30.85% of notebooks are reproducible.

Computational Narrative

A notebook is a computational narrative if there is context to explain the purpose, process, and results. Variables considered when judging the narrative quality of a notebook are the ratio of words of markdown to lines of code, whether the notebook has been renamed, the number of comments per cell, and whether the repository has a description.

4.0% of notebooks have no code. Among notebooks with code, the median is 86 lines of code (mean 145.7). 27.8% of notebooks have no markdown text. Among notebooks with markdown, the median is 194 words (mean 550.9). 54.13% of repositories have descriptions. On average, there are 6.36 words of markdown for each line of code. Further, there are an average of 18.47 comments per notebook (1.0 comments per cell). I also check if the owner renamed the notebook from its default `Untitled[number].ipynb` title, and 1.8% of notebooks are still untitled.

Modeling Standards

After inspecting them individually, I was interested to see how these factors are related. Specifically, I was looking to model the level of *attention* a notebook gets using features related to the content of the notebook as predictors. The four factors that go into attention (number of issues, forks, stargazers, and watchers) are the same as we used to look at collaboration. Here, we are looking at the actual values, not just if it is "at least one". In a test to see whether the four attention factors were associated with the content, reproducibility, or whether the notebook is a computational narrative, I found strong evidence that they are independent.

This was a surprising finding because, as mentioned earlier, I found that collaborative notebooks (at least one fork or issue) tend to have a higher markdown to code ratio and are more likely to have descriptions. However, this was found when simply differentiating between collaborative, watched, and isolated. The lack of association found here might be because of how spread out the data is when considering the actual counts. Some collaborative repositories have only one fork, while others have thousands. In earlier analysis, I lumped all of those repositories together, but here I *want* to distinguish between a little bit of attention and a lot of attention.

Still hopeful, I created three possible measures of "attention" using the four related factors. For each measure, I applied multivariate linear regression with a training set of 80% and a test set of 20%. The first metric was the product of the four variables. The second metric was the first principal component of the four variables–this explained 72.5% of the variance in attention. The final metric was the sum of the four variables, an approximate measure of the number of views a repository has. All regression attempts resulted in train and test R-squared values under 0.00001, meaning almost none of the variance in any attention metric was explained by the predictors.

In the end, there is no evidence that attention is associated with the content of the notebook, features related to reproducibility, or whether a notebook is a computational narrative.

Sources

1. Granger, Brain and Perez, Fernando. *Project Jupyter: Computational Narratives as the Engine of Collaborative Data Science*. 7 July 2015. https://blog.jupyter.org/project-jupyter-computational-narratives-as-the-engine-of-collaborative-data-science-2b5fb94c3c58.