Reproducible Research in the Digital Humanities:
A Checklist with Methods for Implementation in Python

Joseph Flanagan[1][0000-0001-8238-1828]

[1] University of Helsinki, 00014 University of Helsinki, Finland

Joseph.flanagan@helsinki.fi

Keywords: Reproducible Research, Open Science, Project Management

One of the hallmarks of scientific research is that its results are reproducible. While it is not often presented in such terms, the humanities (or at least that corner of it that has stressed interpretation as its focus) has long been concerned with ensuring what we might call "textualist" reproducibility. After all, the extensive use of quotation along with the documentation of those quotations serves the purpose of making the author's evidence transparent, if not reproducible: all the primary evidence for the author's interpretation is contained within the book or article (in the form of the specified examples) and the providence of that evidence has been documented (in the form of a bibliography). You might disagree about the author's interpretations of specific examples, or the appropriateness of the specific examples, or the larger interpretative pattern the author places those examples within, but at least all the author's cards on the table.

Now compare the situation described above with traditional empirical research before the so-called reproducibility crisis brought the problem to the forefront. Readers (and reviewers) often have no access to the raw data, or the decisions that determined how the data was annotated or classified, or the decisions that determined what data would be included or excluded from analysis, or the specific statistical procedures the were used in the analysis, or the countless other steps that would be needed to take the reader from the raw data to the tables, figures, and statistical summaries found in the paper. Compared with the traditional humanities, almost everything in a so-called "empirical" paper has to be taken on faith. And if the reproducibility crisis has shown us anything, it is that that faith has been misplaced, not because researchers are dishonest (although some of them are), nor because researchers are incompetent (although some of them might be), nor because researchers have plenty of incentives to cut corners or weigh the scales to get the results they know they "should have got" (although all researchers have those incentives). It's because empirical research is really difficult and messy.

As computational practices (or, at the very least, computer-assisted methodologies) become more commonplace across the humanities, researchers will increasingly need to address the problems of reproducibility that researchers in the sciences and the social sciences face. My presentation will outline some of the issues researchers in the digital

humanities need to think about along with some strategies for implementing them. First, I will define reproducibility rather narrowly as the ability for an independent researcher to generate the exact same results using the same data as those found in the original paper (reproducibility in this sense is sometimes also referred to as *repeatability* or *replicability*). While this notion of reproducibility is perhaps less important than related forms (say, the robustness of an approach to different theoretical assumptions or the ability of a model to get the "same" results with different data), it is still an important concern, as the reproducibility of a prior study in the narrow sense is necessary before we can assess its robustness or generalizability. After all, we need some means of resolving any conflicts that arise between an original study and latter studies that suggest its findings are not robust or generalizable.

Having defined what I mean by reproducibility, I will then present a means by which it can be achieved. Here I will suggest that reproducibility is a dual process involving documentation and narrative, and that we should rely upon machine-readable code for documentation and narrative exposition for interpretation. Unlike, say, Jupyter notebooks, which mix computation and narrative in a single environment, I will suggest the need to keep the two environments distinct, with Jupyter notebooks providing narrative exposition of the computational process rather than the documentation of those processes and the software environment in which those processes take place. My argument will be that computer-assisted methodologies are at least partly a form of software engineering, and that we should adopt such best practices in software engineering as the use of version control, some form of testing, installation instructions, and dependency management if we want our work to be reproducible by others (or even our future self).

To illustrate what reproducible science as software engineering looks like, I will demonstrate Cookiecutter NLP, an open-source project I am working on that extends Cookiecutter Data Science, a boilerplate project structure that can be used for transparent and reproducible research. When complete, Cookiecutter NLP will provide users with a means of generating a project that

1. has a logical and modularized project structure
2. automates the downloading, preprocessing, modelling, and visualization of data with `make` files
3. tracks changes with Git
4. manages the software stack with Docker
5. automates tests with Travis CI

Admittedly, there are still numerous challenges, technological as well as cultural, that reproducible research faces. I will conclude with a consideration of these challenges and how they might be overcome.

References

1. Cookiecutter Data Science Homepage, https://drivendata.github.io/cookiecutter-data-science/, last accessed 2019/02/10.