– Senior Thesis Written Proposal, 2017 –

Replicating and Verifying Submissions to the Fragile Families Challenge and *Socius* Journal

David Liu
Adviser: Professor Matthew Salganik

1. Motivation and Goal

The purpose of this project is to ensure the reproducibility and verifiability of all submissions made to the Fragile Families edition of the *Socius* journal. Building on efforts of other social science journals, such as the American Journal of Political Science and the American Economics Review [1], this project will support open-sourcing the data and code used to generate *Socius's* findings. The open sourcing process will render the results more transparent while also instilling greater confidence in the conclusions.

At the same time, the project will also focus on analyzing the costs of the open sourcing process as well as documenting best practices. The primary opponents to the open science movement argue that the verification process only delays scientific throughput, with the costs outweighing supposed negligible benefits [1]. Adding on previous efforts to quantify costs of the verification process, this project will document the time and resources necessary to open source the data behind *Socius*. In the process of navigating expected hurdles, the project will suggest additional recommendations for open-sourcing scientific findings, supplementing existing recommendations [3] [2].

2. Approach

2.1. Replicate Personal Fragile Families Submission

As a preliminary step, I will begin by setting up a virtual machine with all the packages and specifications needed to replicate my personal submission to the Fragile Families Challenge. Because I understand the intricacies of my submission, the focus this phase will be on configuring the VM to fulfill all the specifications necessary. A successful configuration will minimize the time and knowledge needed for others to replicate the virtual machine.

2.2. Generalize Replication Procedure for Socius Submissions

Once Challenge participants begin submitting code and data by the journal's October 16 deadline, I will incrementally verify that each submission is reproducible. While analyzing the submission code, I will assemble a VM that has all of the packages and software installed to run any of the submitted code. This process will primarily involved installing language-dependent (R, Python) packages.

[1] *https://www.insidehighered.com/blogs/rethinking-research/should-journals-be-responsible-reproducibility*

2.3. Verify Socius Results

Once the code is proven to be executable, the final phase will involve verifying the quantitative results drawn in each of the submissions. This final phase will likely involve communication with the authors to clarify any discrepancies between the code's output and the paper's contents. Throughout this process, I will document any missing information linking the workings of the code and the paper's findings; this way, readers of *Socius* will be able to replicate more easily.

3. Implementation

Below, I will briefly discuss the technologies that will be used in the replication and verification process.

Virtual Box + Vagrant: Because of personal familiarity, I will begin by replicating all of the submissions on a blank Virtual Box-hosted Unix machine. The goal will be to configure a `Vagrantfile` that contains all of the specifications for the virtual machine. With the Vagrantfile downloaded, users will need to only run the command `vagrant up` to instantiate a fully configured VM.

Docker + Amazon Machine Image (AMI): As a backup and/or supplement to the VirtualBox configuration, Docker and AMI will serve as alternatives. Both of these options are common in commercial software development stacks and are popular choices for creating distributable, packaged environments.

While the three technologies are relatively interchangeable, one of the learning objectives of the implementation will be comparing the services and their ease of use in the context of Open Science. For example, while creating Amazon Machine Images may be suitable for commercial software development, they may be less intuitive for smaller scale computational science projects.

4. Evaluation and Future Work

Evaluation in this project will be measuring the costs of open sourcing scientific data and code. As configuring the VM and navigating the submissions will be highly dependent on the submission, the project inherently carries a degree of variability. However, by establishing concrete benchmarks on the cost of open-sourcing, future researchers and editors alike will be able to better evaluate the merits of verifying journal submissions.

References

[1] J. Freese, "Replication standards for quantitative social science: Why not sociology?" *Sociological Methods & Research*, vol. 36, no. 2, pp. 153–172, 2007.

[2] G. K. Sandve, A. Nekrutenko, J. Taylor, and E. Hovig, "Ten simple rules for reproducible computational research," *PLoS computational biology*, vol. 9, no. 10, p. e1003285, 2013.

[3] V. Stodden, M. McNutt, D. H. Bailey, E. Deelman, Y. Gil, B. Hanson, M. A. Heroux, J. P. Ioannidis, and M. Taufer, "Enhancing reproducibility for computational methods," *Science*, vol. 354, no. 6317, pp. 1240–1241, 2016.