Replication Guidelines for the *Quarterly Journal of Political Science*

December 11, 2014

The success of science depends critically on the ability of peers to interrogate published research in an effort not only to confirm its validity but also to extend its scope and probe its limitations. The *QJPS* replication policy is designed to ensure that ongoing peer review of published research is possible at low-cost.[1] With a firm commitment to progress in the social sciences, the *QJPS* requires that all empirical manuscripts accepted for publication be accompanied by an archive of replication files—hereafter known as the *replication package.* We require the replication package to be organized and documented in a particular fashion to allow other researchers to easily and transparently investigate and build upon the analyses in *QJPS* articles and research notes. Specialized knowledge should not be required to replicate analyses. Rather, the replication package should be sufficient to allow even beginning graduate students to conduct a replication.

It is not sufficient merely to provide the raw materials for replicating the results of a paper. Authors of published research must also provide their data and code in a way that allows a detailed, complete and easily interpretable record of how every result reported in a published paper (and its appendix) was generated. To meet this objective, authors of accepted manuscripts are expected follow the specific guidelines below, while or prior to submitting the final version of their forthcoming manuscript. Please note that while this list identifies many necessary components of a good replication package, the list may not be sufficient depending on the nature of the project. If the *QJPS* Editors and Replication Assistant determine that the replication package fails to satisfy the general principle of accessibility and amenability of research to peer review, the author(s) may be required to revise and resubmit their replication package. In extreme cases, failure to comply within a reasonable period of time may result in rescission of the invitation to publish. One goal of these guidelines is to make such an outcome highly improbable by clearly outlining expectations with regard to what it means for published findings to be easily replicable.

[1] The *QJPS* was the first political science journal to require pre-publication replication and online availability of all contributors' replication datasets and documentation. This document updates, clarifies, and codifies our guidelines based on the experiences of our contributing authors, editors, and a series of outstanding *QJPS* Replication Assistants: Alex Hirsch, Chris Stanton, Zach Peskowitz, and, most recently, Nick Eubank, who initiated and drafted this document.

1 Ease of Replication

By ease of replication, we mean that anyone who is interested in replicating a published article (hereafter, a "replicator") should be able to do so as follows:

1. Open a README.txt file in the root replication folder, and find a summary of all replication materials in that folder, including subfolders if any.
2. After installing any required software (see Section 5 on Software Dependencies) and setting a working directory according to directions provided in the README.txt file, the replicator should be able simply to open and run the relevant files to generate every result and figure in the forthcoming publication. This includes all results in print and/or online appendices.
3. Once the code has finished running, the replicator should be able easily to locate the output and to see where that output is reported in the paper's text, footnotes, figures, tables, or appendices.

2 README.txt File

All replication packages should prominently include a plain text file named README.txt. Authors should assume this is the first file a replicator will examine. The README.txt file should include, at a minimum:

1. **Table of Contents:** a brief description of every file in the replication folder.
2. **Notes for Each Table and Figure:** a short list of where replicators will find the code needed to replicate all parts of the publication.
3. **Base Software Dependencies:** a list of all software required for replication, including the version of software used by the author (e.g. Stata 11.1, R 2.15.3, 32bit Windows 7, OSX 10.9.4).
4. **Additional Dependencies:** a list of all libraries or added functions required for replication, as well as the versions of the libraries and functions that were used and the location from which those libraries and functions were obtained.

    R: the current R versions can be found by typing `R.Version()` and information on loaded libraries can be found by typing `sessionInfo()`.

    STATA: Stata does not specifically "load" extra-functions in each session, but a list of all add-ons installed on a system can be found by typing `ado dir`.

5. **Seed locations:** Authors are required to set seeds in their code for any analyses that employ randomness (e.g., simulations or bootstrapped standard errors. (For further discussion, see Section 7). The README.txt file should include a list of locations where seeds are set in the analyses so that replicators can find and change the seeds to check the robustness of the results.

3 Scope and Depth of Replication

3.1 Scope

Replication data and code must be provided for all results referenced in the paper. This includes not only all tables and figures in the primary manuscript, but also any results referred to in text in the body of the paper, results discussed in footnotes, and findings to be posted in online appendices.

3.2 Depth of Replication

The *QJPS* requires that every replication package include the code that computes the primary results of the paper. In other words, it is not sufficient to provide a file of pre-computed results along with the code that formats the results for LaTeX. Rather, the replication package must include everything that is needed to execute the statistical analyses or simulations that constitute the primary contribution of the paper. For example, if a paper's primary contribution is a set of regressions, then the data and code needed to produce those regressions must be included. If a paper's primary contribution is a simulation, then code for that simulation must be provided—not just a dataset of the simulation results. If a paper's primary contribution is a novel estimator, then code for the estimator must be provided. And, if a paper's primary contribution is theoretical and numeric simulation or approximation methods were used to provide the equilibrium characterization, then that code must be included.

3.3 Data

Although we do not necessarily require the submitted code to access the data if the data are publicly available (e.g., data from the National Election Studies, or some other data repository), we absolutely require that the dataset containing all of the original variables used in the analysis be included in the replication package. For the sake of transparency, the variables should be in their original, untransformed and unrecoded form, with code included that performs the transformations and recodings in the reported analyses. This allows replicators to assess the impact of transformations and recodings on the results.

3.3.1 Proprietary and Non-Public Data

If an analysis relies on proprietary or non-public data, please contact the *QJPS* Editors before or during your initial submission. Even when data cannot be released publicly, authors may be required to provide *QJPS* replicators access to data for replication prior to publication.

4 Working Directories

All code should be written so a replicator can set a single working directory at the top of the first file to be run and then have all subsequent code run smoothly by appropriately referencing the directory substructure.

5 Software Dependencies

Authors must include copies of all software (including both base software and add-on functions and libraries) used in the replication that is **NOT**:

1. R, Stata, Matlab, Java, Python, or ArcMap

> *Although please note that while authors do not need to provide installations of Java or Python, Java- and Python-based applications must be included.*

Including add-ons entails providing a full copy of the add-on software in the replication folder, as well as either (a) installation directions in the README.txt file or (b) code that installs the add-on from a file in the replication folder.

A command that installs the add-on from an online repository (like `install_packages("PACKAGE")` in R or `ssc install PACKAGE` in Stata) is not sufficient. Online software repositories provide easy access to the latest version of powerful add-ons, but because they always provide the most recent version of add-ons to users, the software provided in response to a given query actually changes over time. As replication may take place months or years after a paper has been published, these changes may lead to replication failures. Indeed, the QJPS has experienced replication failures due to add-on software updates on the CRAN server that occurred between when papers were submitted to the QJPS and when they were subject to in-house replication review.

Directions on including and installing copies of R packages from CRAN or Stata functions from SSC in a replication package can be found in Appendix A.

6 Transcription

The vast majority of replication errors are due to authors' mistakes in transcribing results from their statistical software to their documents. Therefore, we strongly discourage hand-copying results into documents. Instead, when possible, use software that directly exports results into LaTeX or Excel, such as `esttab`, `outtable`, `outreg2` for Stata, or `sweave` for R.

We recognize that not everyone is comfortable with these tools and that situations arise in which formatting concerns preclude their use. In such situations, authors should:

- Print out a copy of their paper,
- Re-generate all the results in their paper one by one using their replication files (and *only* their replication files),
- Walk through the printed document(s) making sure each result is matched to an analysis that is clearly labelled and created by output from the replication file.

Please pay special attention to rounding of the last digit. Experience affirms that authors are particularly prone to rounding errors during transcription.

7 Randomizations and Simulations

A large number of modern algorithms employ randomness in generating their results (e.g., the bootstrap). In these cases, replication requires both (a) ensuring that the *exact* results in the paper can be re-created, and (b) ensuring that the results in the paper are typical rather than cherry-picked outliers. To facilitate this type of analysis, authors should:

1. Set a random number generator seed in their code so it consistently generates the exact results in the paper;
2. Provide a note in the README.txt file indicating the location of all such commands, so replicators can remove them and test the representativeness of result.

In spite of these precautions, painstaking experience has shown that setting a seed is not always sufficient to ensure exact replication. For example, some libraries generate slightly different results on different operating systems (e.g. Windows versus OSX) and on different hardware architectures (e.g. 32-bit Windows 7 versus 64-bit Windows 7). To protect yourself from such surprises, you should test your software on multiple platforms, and document any resulting exceptions or complications in your README.txt file.

8 ArcGIS

Although we encourage authors to write replication code for their ArcGIS-based analyses using the ArcPy scripting utility, we recognize that most authors have yet to adopt this tool. For the time being, we will accept detailed, step-by-step instructions for replicating results via the ArcGIS Graphical User Interface (GUI).[2]

9 Recommended Pre-Submission Exercises

The following two suggestions follow from nearly a decade of *QJPS* replication experiences. They are not mandatory, but they are highly recommended.

1. **Test your files on a different computer, preferably with a different operating system:** Once you've prepared your replication code, email it to a different computer, unzip it, and run it. Code often contains small dependencies—things like unnoticed software dependencies or specific file locations—that go unnoticed until replication. Running code on a different computer often exposes these issues in a way that running the code on your own does not.
2. **Check every code-generated result against your final manuscript PDF:** The vast majority of replication problems emerge because authors either modified their code but failed to update their manuscript, or made an error while transcribing their results into their paper. Please print out a copy of your manuscript and check each result before submitting your final version of the manuscript and replication package.

[2] ArcPy is a Python-based tool for scripting in ArcGIS, providing ArcGIS projects with an analogue to .R files for R and .do files for Stata. The tool requires very little experience with Python. A tutorial on the utility is available from former *QJPS* Replication Assistant Nick Eubank (nickeubank at gmail.com).

10 Conclusion

Although these guidelines are more specific and more stringent than those of most or all other political science journals, they are consistent with our commitment to ensuring the replicability, hence the credibility, of *QJPS*-published research.

A Downloading and Installing Add-Ons

This Appendix describes how to download libraries from common online repositories—like CRAN and SSC—for inclusion in replication files.

A.1 Including R Add-Ons from CRAN

There are two steps to including R add-ons (often called libraries) in your replication package:

1. Download the appropriate materials from the CRAN repository;
2. Add code that installs the downloaded add-on from disk, rather than from the online servers.

These two steps are discussed in turn.

A.1.1 Downloading Libraries from CRAN

Source files for all add-ons that can be installed using the `install_packages()` command in R can be found on the CRAN project website. The CRAN website for most packages can be found by googling their names, or at: `http://cran.r-project.org/package=PACKAGENAME`. So for example, the "plyr" package can (as of November 2014) be found at `http://cran.r-project.org/package=plyr`.

The source file is labeled "Package source," and has the suffix .tar.gz (a format sometimes called a "tarball"). This file can be used to install the add-on on any platform. Save this file to your replication folder.

A.1.2 Load downloaded library in replication files

The add-on can now be installed from this file (instead of from the CRAN server) on any computer using the `install_packages()` command with the following options:

```
install.packages("FILEPATH_TO_PACKAGE/PACKAGENAME.tar.gz",
         repo = NULL, type="source")
```

A.2 Including Stata Add-Ons

Add-ons in Stata take the form of files with the suffix `.ado`, and are often also paired with files with the suffix `.hlp` or `.sthlp` that contains help documentation for the program.

A.2.1 Downloading SSC add-ons

Add-ons files on the SSC repository (add-ons usually installed via the `ssc install PACKAGENAME`) can be downloaded into the current working directory using the command:

```
ssc copy PACKAGENAME
```

A.2.2 .ado files from other services

Including add-ons that are not in the `ssc` repository is only slightly more difficult. When an add-on is installed in Stata (for example, using the `findit` command), it is placed in the Stata PLUS directory. To find your PLUS directory, use the command `sysdir`. To copy an add-on that you have already installed into your replication package, simply copy the relevant `.ado` file(s) (some add-ons have multiple .ado files) from the PLUS directory into your replication file (we recommend also including the associated help file, which will probably have the same name and the suffix `.hlp` or `.sthlp`, although this is not strictly necessary).

A.2.3 Installing .ado Files

Installation of an add-on in Stata is quite easy—simply copy the relevant `.ado` file(s) in your replication folder to Stata's PLUS directory. This can be accomplished with the following code:

```
copy "PATH_TO_DIRECTORY_WITH_ADDONS/addon.ado"  "‘c(sysdir_plus)’"
discard
```

If you also include a help file, you can also run the command:

```
copy "PATH_TO_DIRECTORY_WITH_ADDONS/addon.hlp"  "‘c(sysdir_plus)’", replace
```

or

```
copy "PATH_TO_DIRECTORY_WITH_ADDONS/addon.sthlp"  "‘c(sysdir_plus)’", replace
```

depending on the suffix on the relevant help file.