The Stata Journal (2016)
16, Number 1, pp. 88–95

Quantifying the uptake of user-written commands over time

Babak Choodari-Oskooei
Hub for Trials Methodology Research
MRC Clinical Trials Unit
University College London
London, UK
b.choodari-oskooei@ucl.ac.uk

Tim P. Morris
Hub for Trials Methodology Research
MRC Clinical Trials Unit
University College London
and
Department of Medical Statistics
London School of Hygiene and Tropical Medicine
London, UK
tim.morris@ucl.ac.uk

Abstract. A major factor in the uptake of new statistical methods is the availability of user-friendly software implementations. One attractive feature of Stata is that users can write their own commands and release them to other users via Statistical Software Components at Boston College. Authors of statistical programs do not always get adequate credit, because programs are rarely cited properly. There is no obvious measure of a program's impact, but researchers are under increasing pressure to demonstrate the impact of their work to funders. In addition to encouraging proper citation of software, the number of downloads of a user-written package can be regarded as a measure of impact over time. In this article, we explain how such information can be accessed for any month from July 2007 and summarized using the new `ssccount` command.

Keywords: dm0086, ssccount, SSC, impact

1 Introduction

Many statisticians are paid to develop new methods, but implementing methods in software is not always recognized as a key part of this activity. A published article detailing a new method is citeable, and citations can be tracked, providing funders and bosses with a measure of interest or relevance. There is no equivalent to an impact factor or H-index for programs, which often go uncited by users. It is thus harder to demonstrate the value of time spent writing and testing programs. However, there are other indicators that can be used to demonstrate impact (Brueton et al. 2014).

We regard the release of programs as an important factor in the uptake of new methods (Pullenayegum et al. 2016). Historically, this appears to be supported by the following:

- The Cox model was originally published in 1972 (Cox 1972), but it was not widely used until implementations in Fortran by Richard Peto and colleagues and Kalbfleisch and Prentice (1980).
- Multiple imputation was first conceived in 1978 (Rubin 1978) followed by a period of theoretical developments (Rubin 1987), but the widespread use now seen (Rezvan, Lee, and Simpson 2015) did not occur until the release of the R package `mice` (van Buuren and Oudshoorn 2000) and the Stata package `ice` (Royston 2004).
- Propensity-score matching was originally proposed in 1983 (Rosenbaum and Rubin 1983) and has gradually been applied more and more since the turn of the millennium, thanks in part to programs such as `psmatch2` (Leuven and Sianesi 2003).

Each new Stata release adds commands implementing recent methods, but it would be unreasonable to expect StataCorp to keep on top of all the methodological developments in statistics and implement them. Rather, the onus falls on methodologists to implement their own methods and promote the software. Having written a program, a user can share it easily: a package of files can be submitted to the Statistical Software Components (SSC) repository at Boston College, and it can then be downloaded by others by typing `ssc install` *pkg_name* in Stata's command line.

The `ssc hot` command returns the number of downloads in the previous month for most user-written packages on SSC. Many users might not know that they can obtain the datasets that this command is based on for any month dating back to July 2007. These monthly datasets can then be linked.

In this article, we describe how to obtain data on monthly hits, and we introduce the `ssccount` command, which downloads the datasets for a specified time window. `ssccount` allows specification of certain packages and authors of interest, and it provides a graph plotting downloads over time. The number of downloads over time provides a useful—though imperfect—picture of how much a program is used, provided it has been released on SSC. The `ssccount` command is thus one way for Stata programmers to demonstrate the uptake of their packages and evaluate the value of the time spent writing them.

2 Methods

In this section, we discuss how to obtain the number of downloads of user-written statistical packages, which can be regarded as a soft measure of impact, and we introduce the `ssccount` command, which can be used for this purpose.

2.1 Statistics regarding uptake

The SSC archive is a well-known repository for user-written commands. The host site, RePEc services, tallies the individual file downloads whenever a user issues `ssc install` *pkg_name* to Stata. Typing `ssc hot` produces a list of the 10 (by default) most downloaded packages for the previous month. This list consists of the top 10 rows of data from a file containing the downloads for all packages. A file is created for each month and stored in the SSC archive, which goes back to July 2007.

Stata users can access this information from Stata by submitting the following command:

```
. use http://repec.org/docs/sschotPxxx.dta
```

In this command, *xxx* corresponds to Stata's monthly calendar (for example, *xxx* = `570` is the "Stata internal form" value for July 2007 [typing `display %tm 570` returns `2007m7`]). So to obtain the file containing the number of package downloads in July 2007, you replace *xxx* with `570` in the above command. The number of package "hits" (downloads) reported can be noninteger because some users might have downloaded only some of the files in a package. Some packages consist of many files, and not all are updated each time.

The number of hits must be interpreted cautiously for these reasons:

1. The statistics appear to be limited to packages containing user-written ado-files. For example, graph schemes are not counted.
2. The data do not distinguish between the first download and the downloads of an update.
3. If a user downloads a command to two computers (say, one at work and one at home), this is counted as two hits.

Clearly, the precise number of hits should not be relied on too heavily—there is potential for commands to look more impressive by releasing many incremental updates instead of a fully developed first version—but the information is useful.

On the other hand, citations in peer-reviewed articles are widely used as a measure of "impact", but they have their own pitfalls. Simple citation counts are agnostic to whether citations were for positive, negative, or neutral reasons. Although we cannot tell precisely what the spirit of a software download was, it seems plausible that downloads are mainly for positive reasons.

3 The ssccount command

The `ssccount` command downloads datasets detailing monthly downloads of user-written commands from SSC for specified authors and packages, and it optionally plots the results.

3.1 Syntax

The syntax for the `ssccount` command is

`ssccount` [`,` `from(`*month*`)` `to(`*month*`)` `author(`*author_name*`)` `clear` `fillin(`#`)` `graph` `package(`*pkg_name*`)` `saving(`*filename*`, replace)`]

where *month* is a calendar month in Stata's `%tm` format.

3.2 Options

`from(`*month*`)` specifies the earliest month of data to download. This must be entered in Stata's `%tm` format (for example, January 2011 is specified by `2011m1`). Specifying a month before July 2007 (`2007m7`) will return an error because this is before records began. The default is `from(2007m7)`.

`to(`*month*`)` specifies the latest month of data to download. As with `from()`, this must be entered in Stata's `%tm` format (for example, January 2011 is specified by `2011m1`). Specifying a month before July 2007 (`2007m7`) will return an error. The default is the current month minus three months, which helps users avoid trying to download datasets that do not yet exist, though one further month may be available. (Users can check the latest available month by typing `ssc hot`.)

`author(`*author_name*`)` specifies the name of the author whose packages are of interest. The names on SSC packages can be inconsistent. You do not have to get it exactly right, as long as the name used contains what you specify in `author()`. The option is not case sensitive, so specifying `author(bloggs)` is the same as `author(BLOGGS)` or anything in between, like `author(BlOgGs)`.

`clear` specifies that the data in memory will be cleared. If `saving()` or `clear` is not specified and you have data in memory, `ssccount` will exit with an error.

`fillin(`#`)` calls the `fillin` command (see [D] **fillin**). This option is used with plots when more than one author or package has been specified. It creates missing months to form a rectangular dataset and fills each one with # hits. Filling as missing (`.`) is allowed. The default is to not fill anything.

`graph` draws a simple graph of the month-by-month hits using `twoway line` and overlays a smoothed trend using `lowess`. If the data contain multiple authors or packages, the graphs will be drawn by author and package.

`package(`*pkg_name*`)` specifies the name of the package of interest. This may be useful if an author has written multiple packages but a user is interested in one in particular. It can also be helpful if the author's name is a substring of one or more other authors' names.

`saving(`*filename*`, replace)` specifies the downloaded data be saved as *filename*`.dta`.

3.3 Examples

To download the data on downloads (hits) for all SSC packages and save them to a file called `allhits.dta`, type

```
. ssccount, saving(allhits, replace)
Looking to download 99 months of SSC files (Jul 2007 to Sep 2015)
..................................................................................
> ..................
file allhits.dta saved
```

This will append the various files; the appended dataset will be stored in `allhits.dta`.

Next, we look at the downloads of Royston's (2004) `ice` command over time. The package was first released as `ice` in April 2005 (after its earlier incarnations as `mvis` and, briefly, `mice`). As noted earlier, the records in SSC begin in July 2007. Here is the command:

```
. ssccount, from(2007m7) to(2015m9) author(Royston) graph package(ice)
> saving(icehits, replace)
Looking to download 99 months of SSC files (Jul 2007 to Sep 2015)
.................................................................................
> ..................
file icehits.dta saved
```

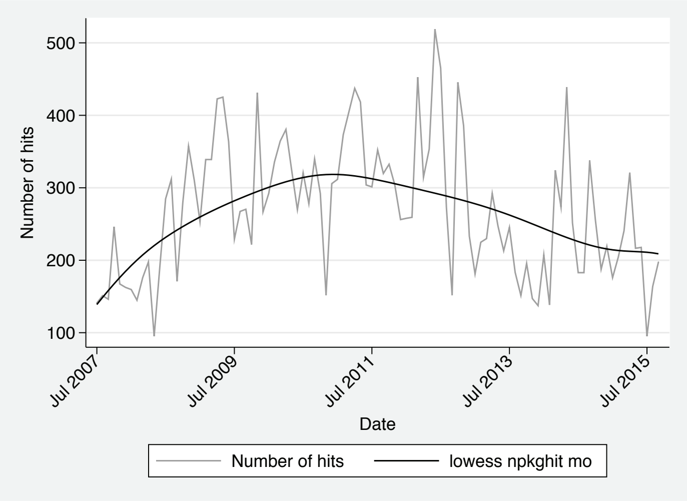

Figure 1. Plot showing the number of hits for `ice`, July 2007 to September 2015. Gray line: number of hits recorded each month; black curve: lowess-smoothed trend.

Here we have downloaded data for all packages from July 2007 to September 2015, and we kept the data if the author's name contains `Royston` and the package is named `ice`. The resulting data are saved to `icehits.dta`, and the graph shown in figure 1 is produced.

Note the reduction in hits from the end of 2012. This is presumably due to the release of `mi impute chained` by StataCorp; users were likely directed to use `mi impute chained` instead of `ice` because of the reassurance that comes with using an official Stata command. Further development of `ice` then became less necessary, so updates were less frequent. There is a surprising sharp spike in `ice` hits during 2014 despite no updates at the time. We speculate that the rise was due to an article critiquing multiple imputation by predictive mean matching (Morris, White, and Royston 2014), which praised the `ice` implementation and noted the serious shortcomings of `mi impute pmm`.

As a further example, we look at the uptake of the `psmatch2` command. We use `allhits.dta`, which we previously downloaded. Downloading the datasets afresh is a slow process.

```
. use allhits, clear

. keep if lower(package) == "psmatch2"
(180,402 observations deleted)

. sort mo

. twoway (line npkghit mo, lcolor(gs10)) (lowess npkghit mo, lp(l)),
> ylabel(#6,format(%9.0f) angle(0)) xlabel(,angle(45)) yscale(r(0 .))
> ytitle("Number of hits")
```

Figure 2 demonstrates that the `psmatch2` command is much used, and, unlike `ice`, its use continues to increase despite the release of Stata's official `teffects` command.

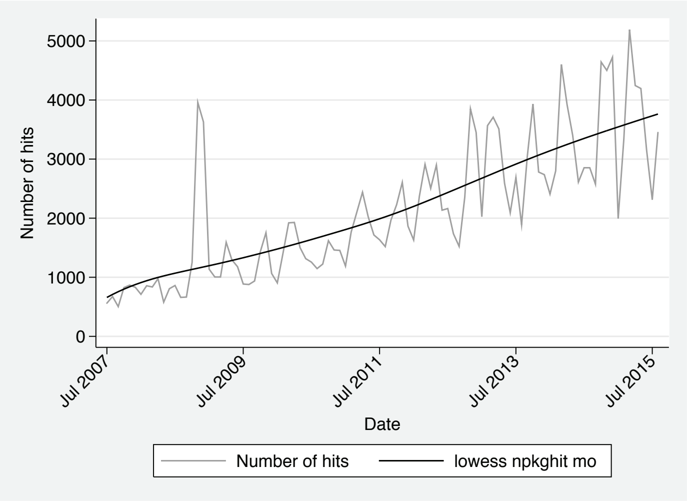

Figure 2. Plot of the number of hits for `psmatch2`, July 2007 to September 2015. Gray line: number of hits recorded each month; black curve: lowess-smoothed trend.

4 Closing remarks

Accessible software for implementing new statistical methods is obviously an important factor in the uptake of new methods. We have introduced a command, `ssccount`, that counts the monthly downloads of user-written packages stored in the SSC archive. The program provides useful information on the extent of the use of such packages.

Some authors of commands put their packages on only personal or corporate websites, or they do this in addition to putting packages on SSC. The ability to keep track of downloads makes the option of releasing packages exclusively on the SSC archive attractive. We hope the `ssccount` command is helpful for highlighting the packages with the greatest uptake over time.

5 Acknowledgments

We thank Kit Baum for helping us understand where the datasets containing downloads are stored. We are grateful to Patrick Royston and Roger Newson for their comments on a draft and earlier versions of the program and to Stephen Evans for information on the history of the Cox model. We also thank the associate editor and an anonymous reviewer for their useful comments on the earlier version of this article. This work was supported by the UK Medical Research Council (MRC) London Hub for Trials Methodology Research grant number MC_EX_G0800814 (510636, MQEL).

6 References

Brueton, V. C., C. L. Vale, B. Choodari-Oskooei, R. Jinks, and J. F. Tierney. 2014. Measuring the impact of methodological research: A framework and methods to identify evidence of impact. *Trials* 15: 464.

Cox, D. R. 1972. Regression models and life-tables. *Journal of the Royal Statistical Society, Series B* 34: 187–220.

Kalbfleisch, J. D., and R. L. Prentice. 1980. *The Statistical Analysis of Failure Time Data*. New York: Wiley.

Leuven, E., and B. Sianesi. 2003. psmatch2: Stata module to perform full Mahalanobis and propensity score matching, common support graphing, and covariate imbalance testing. Statistical Software Components S432001, Department of Economics, Boston College. https://ideas.repec.org/c/boc/bocode/s432001.html.

Morris, T. P., I. R. White, and P. Royston. 2014. Tuning multiple imputation by predictive mean matching and local residual draws. *BMC Medical Research Methodology* 14: 75.

Pullenayegum, E. M., R. W. Platt, M. Barwick, B. M. Feldman, M. Offringa, and L. Thabane. 2016. Knowledge translation in biostatistics: A survey of current practices, preferences, and barriers to the dissemination and uptake of new statistical methods. *Statistics in Medicine* 35: 805–818.

Rezvan, P. H., K. J. Lee, and J. A. Simpson. 2015. The rise of multiple imputation: A review of the reporting and implementation of the method in medical research. *BMC Medical Research Methodology* 15: 30.

Rosenbaum, P. R., and D. B. Rubin. 1983. The central role of the propensity score in observational studies for causal effects. *Biometrika* 70: 41–55.

Royston, P. 2004. Multiple imputation of missing values. *Stata Journal* 4: 227–241.

Rubin, D. B. 1978. Multiple imputations in sample surveys: A phenomenological Bayesian approach to nonresponse. In *Proceedings of the Survey Research Methods Section of the American Statistical Association*, 20–34.

———. 1987. *Multiple Imputation for Nonresponse in Surveys.* New York: Wiley.

van Buuren, S., and C. G. M. Oudshoorn. 2000. *Multivariate Imputation by Chained Equations: MICE V1.0 User's Manual.* Leiden, The Netherlands: Netherlands Organization for Applied Scientific Research.

About the authors

Babak Choodari-Oskooei is a statistician in the Hub for Trials Methodology Research at the MRC Clinical Trials Unit at University College London. He has a particular interest in survival analysis, clinical trials methodology, and research impact.

Tim P. Morris is a medical statistician interested in statistical methods to improve the design and analysis of randomized trials and meta-analyses and in the use of simulation studies. He is a Stata enthusiast.