How used are user-released commands? Introducing `sscount`

Tim P. Morris, Babak Choodari-Oskooei

11 Sep 2015

MRC Clinical Trials Unit at UCL

INTRODUCTION

INTRODUCTION

I am going to present a command in the larger context of thinking about the dissemination and impact of our work.

Some of us develop new methods; some of us use them; some do both.

If a new method improves on existing methods then hopefully it will be used. Are new methods used?

INTRODUCTION

> *'…new statistical methods are underused. Encouraging statistical methodologists to develop a knowledge translation plan for improved dissemination and uptake, placing greater value on the role of the statistical collaborator in research, and providing software alongside new methods may improve the use of newly developed statistical methods.'*

Eleanor Pullenayegum *et al*. Knowledge translation in biostatistics: a survey of current practices, preferences, and barriers to the dissemination and uptake of new statistical methods. *Statistics in Medicine* 2015 (ePub ahead of print).

FROM METHOD TO UPTAKE: BAD EXAMPLES

Cox DR. Regression models and life tables. *Journal of the Royal Statistical Society, series B* 1972; 34:187–220.

Rubin DB. Multiple imputations in sample surveys: a phenomenological Bayesian approach to nonresponse. *Proceedings of the Survey Research Methods Section* 1978; 20–28.

Rosenbaum PR, and DB Rubin. The central role of the propensity score in observational studies for causal effects. *Biometrika* 1983; 70(1): 41–55.

WRITING AND RELEASING COMMANDS

A great feature of Stata is that you can write your own commands that can be invoked in the same way as factory commands. (I didn't realise until recently that this is not as simple in other software.)

It's also really easy to 'release' a command to other users to download, thanks to **`net from`**, **`ssc install`** and **`Kit Baum`**. (I didn't realise until recently that this is not as simple in other software.)

WRITING AND RELEASING COMMANDS

StataCorp won’t just implement your method and include it in the next version (ask Mike Kenward & James Roger).

If *you* want a method used, *you* need to implement it.

Implementing a method is a core part of its development, but often unrecognised. Writing a Stata command takes time, as does writing a paper for Stata Journal.

Funders want evidence of the impact of our work.

IMPACT

For a peer-reviewed article, citations give a soft marker of:

- INTEREST

IMPACT

For a peer-reviewed article, citations give a soft marker of:

- INTEREST
- RELEVANCE

IMPACT

For a peer-reviewed article, citations give a soft marker of:

- INTEREST
- RELEVANCE
- IDIOCY

IMPACT

For a peer-reviewed article, citations give a soft marker of:

- INTEREST
- RELEVANCE
- IDIOCY

– all of which mean **IMPACT**.

IMPACT

Commands themselves can be cited but people generally don’t know or forget.

How can we measure the impact of a command?

ssc hot

The **ssc hot** command tells you the most downloaded user-written packages. It does this for a recent month, usually about two months previous.

You can use the **author** option for navel gazing.

```
. ssc hot, author(Morris)
```

SSC HITS

But maybe you want to know the number of hits for other months. It would be interesting to know about total hits, and to see how things go over time.

The **ssc hot** command basically says:

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

Here, ___ is a number that specifies the month. You can find out which month using something like

```
. display %tm 600
```

SSC HITS

The earliest file on repec.org is numbered 570…

SSC HITS

The earliest file on repec.org is numbered 570…

```
. display %tmm_CY 570
```

SSC HITS

The earliest file on repec.org is numbered 570…

```
. display %tmm_CY 570
Jul 2007
```

SSC HITS

The earliest file on repec.org is numbered 570…

```
. display %tmm_CY 570
Jul 2007
```

…so you can download these files starting from July 2007 and look at the total hits or uptake over time.

SSC HITS

You're probably thinking,
'That's cool, I'm going to check ***`cmd_name`***.'

SSC HITS

You're probably thinking,
'That's cool, I'm going to check ***`cmd_name`***.'

DON'T PANIC!

A new command, **`ssccount`**, is available on SSC. It downloads and appends the datasets.

Optionally specify author/s and/or package/s and plot hits over time.

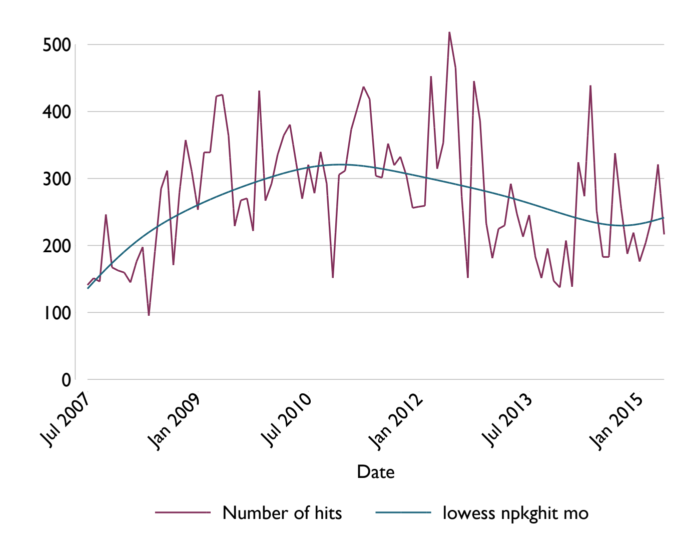
Syntax of `ssccount`

```
ssccount [ , from(first_month) to(last_month)
    author(author_name) clear graph
    package(pkg_name)
    saving(filename, replace) ]
```

EXAMPLE: ice

```
. ssccount , from(2007m7) to(2015m7)
>    author(Royston) graph package(ice)
```

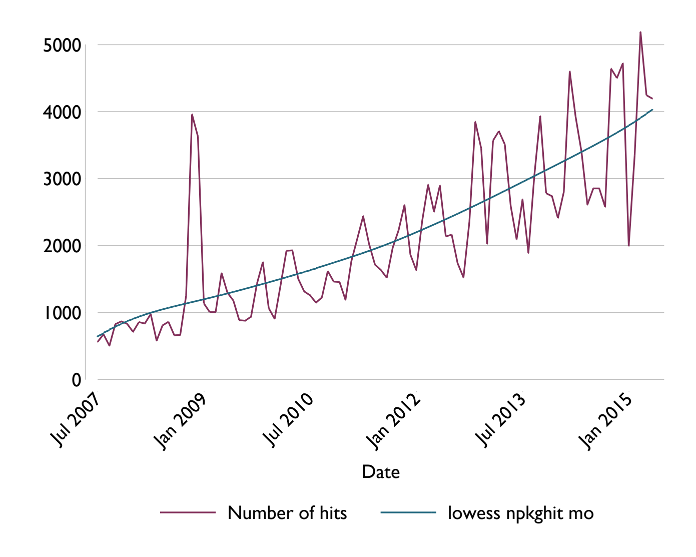
EXAMPLE: `ice`


500
400
300
200
100
0
Jul 2007
Jan 2009
Jul 2010
Jan 2012
Jul 2013
Jan 2015
Date
Number of hits
lowess npkghit mo

EXAMPLE: psmatch2

```
. ssccount , graph package(psmatch2)
```

EXAMPLE: `psmatch2`



Cautionary remarks

If you update a package, people running **`adoupdate`** will give you new hits.

Not all package types get counted (e.g. graph schemes).

This is arguably not very interesting for utility commands.

I wouldn't risk running this:

```
forval i = 1/100000 {
    ssc install cmd_name
    ssc uninstall cmd_name
}
```

SUMMARY I

Funders want evidence of impact.

If you write programs, encourage people to cite them. You can encourage citation in help files; proper documentation is important (help files, advice on use, case studies).

Citations may be negative; software downloads should be at worst apathetic. It's useful to have an idea of how many people are using your commands.

SUMMARY II

Waste countless happy hours looking at hits for your packages and wondering who is downloading them, what they use it for, and if they are using it wrong.

FOR REDDITORS

TL;DR: `ssccount` is like `ssc hot` but over time.

ACKNOWLEDGEMENTS

Max Parmar for telling us he wanted to know about this for the Unit.

Kit Baum for explaining how to get the datasets used by `ssc hot`.

Patrick Royston for comments.

How used are user-released commands? Introducing `ssccount`

Tim P. Morris, Babak Choodari-Oskooei

11 Sep 2015

MRC Clinical Trials Unit at UCL

DISCUSSION POINTS

How do you approach ‘validation’ of your program? Simulations, comparison with existing packages?

What responsibility should/do user-authors take for programs they release?