The Stata Journal (2005)
5, Number 1, pp. 43–45

Stata at 20: a personal view

Patrick Royston

I bought my first copy of Stata in May 1989, and it was version 2.05. I still use Stata most days, but now it is version 8.2. Why has it survived and prospered so long in my statistical armory?

I still have my 1989 shipment of Biturbo Stata version 2.05 and Stage 1.0 Graphics Editor, on five 3.5-inch, 720 Kb floppies. On opening at random my copy of the version 2.05 reference manual (copyright March 1989 © Computing Resource Center, Los Angeles, California), the page fell open at page 539, a summary of something called `Stat.Kit`. I recalled then the excitement of the arrival of "kits"; they were the forerunners of the now familiar and quintessential ado-files. In those days, there were 4 such kits (`Stat.Kit`, `Graph.Kit`, `Data.Kit`, `Survive.Kit`), and they provided a variety of programs in each relevant area. You typed, for example, `run Stat.Kit`, and the associated 17 programs were loaded into memory. After that you used them exactly as one now does an ado-file.

On reflection, Stata has a few key features that I value above all. First, I will illustrate one of these features in a little detail, and later I will briefly mention the others. This feature is *backwards compatibility*, which in Stata lingo is known as "version control". As a simple experiment designed to test version control, I wrote and ran in Stata 8.2 the following do-file:

```
/*
Experiment to show version control.
Patrick Royston.
*/
* Set up version
version 2.05
set logtype text
log using stata.log, replace
* clear out all ado-file support
quietly forvalues j = 1/7 {
adopath -1
}
* Load Stat.Kit
run stat.kit
* Load old auto data
use auto
* Create mpg + uniform() * 5
gen mpg2 = mpg + uniform() * 5
* Perform equal-variance t-test
ttest mpg = mpg2
log close
```

What I am doing here is to set Stata 8.2 to version 2.05, remove *all* ado-file support, load `Stat.Kit` and the classic `auto.dta` dataset, generate a new variable `mpg2` from `mpg` with on average about 2.5 mpg greater values, and run an unpaired t-test comparing `mpg` with `mpg2`. Here is the output as stored in `stata.log`:

```
--------------------------------------------------------------------------------
       log:  e:\tsj\stata20\stata.log
  log type:  text
 opened on:  28 Dec 2004, 10:03:35
. * clear out all ado-file support
. quietly forvalues j=1/7 {
. * Load Stat.Kit
. run stat.kit
Loading Stat.Kit Release 2.05 Copyright (c) 1986-1989 by ==C=R=C==
All rights reserved.
The following new commands are now available:
   blogit         genstd         kwallis        signrank       ttest
   bprobit        glogit         means          signtest
   dbeta          gprobit        ranksum        spearman
   genrank        ksmirnov       regdw          teststd
See help Stat.Kit.
. * Load old auto data
. use auto
(1978 Automobile Data)
. * Create mpg + uniform() * 5
. gen mpg2 = mpg + uniform() * 5
. * Perform equal-variance t-test
. ttest mpg = mpg2
    Variable |       Obs        Mean    Std. Dev.       Min        Max
-------------+----------------------------------------------------------
         mpg |        74     21.2973    5.785503         12         41
        mpg2 |        74    23.68922    5.945047   14.43152   44.18587
Test: means of mpg and mpg2 are equal (assuming equal variances)
 Difference = -2.3919258
t-statistic = -2.48 with 146 d.f.
 Prob > |t| = 0.0143
. log close
       log:  e:\tsj\stata20\stata.log
  log type:  text
 closed on:  28 Dec 2004, 10:03:35
--------------------------------------------------------------------------------
```

Lo and behold, Stata 8.2 runs correctly under version 2.05 control using 2.05 program code and data. This to my mind is impressive. How many other statistical packages would stand up to such a severe test? For me, version control is an apparently unglamorous property of Stata that shows StataCorp's strong commitment to the practical needs of its users, and is one reason why Stata still has a rosy future. I dread to think how many lines of C program code sit behind the scenes supporting the innocent-looking `version #` command. Surely this is virtuoso computer programming of a high order?

Other essential features of Stata I particularly value are its graphics, its simple but elegant command syntax (which in essence has not changed since version 2.05), its seamless programmability via ado-files, and last, but by no means least, its excellent documentation. I do recall that having fast, high-resolution graphics was in 1989 an important selling point and a key determinant of why I bought Stata. The fact that formulas are given in the manual even for quite complex mathematical calculations such as the partial likelihood in a Cox regression model has been immensely useful, and StataCorp even now continues to pay attention to improving this aspect of the product. I could add other far-sighted innovations that were even rather shocking at the time of their introduction, such as increasing levels of integration with the Internet for updating Stata and for installing add-on packages. Certainly the list-server Statalist has played a useful role in the development of user awareness, although I myself don't find much time to access it nowadays. One must also mention the *Stata Journal* and its predecessor, the *Stata Technical Bulletin*. No serious package can be without such a publication.

Is Stata now perfect? Of course not. Areas in which I would like to see improvements include documentation of graphics and GUI dialog programming, greater searchability and flexibility of the help system (I have lost count of the number of times I have trawled through the help system in search of some obscure graphics option), acceleration of drawing a graph, and a facility to make it easier to write help files in Stata's markup language, SMCL. A graphics editor would be nice too; my 1989 shipment included Stage 1.0, the first and only edition of the Stata graphics editor, a fine program that has not yet been replaced.

Above all, though, StataCorp's encouragement of users to get involved and StataCorp's habit of valuing and responding to their subsequent efforts have been some of the most delightful and productive aspects of Stata. If that interaction is ever lost, Stata will probably die.

About the Author

Patrick Royston is a medical statistician of 25 years of experience, with a strong interest in biostatistical methodology and in statistical computing and algorithms. At present, he works in clinical trials and related research issues in cancer. Currently he is focusing on problems of model building and validation with survival data, including prognostic factors studies, on parametric modeling of survival data, and on novel trial designs.