Stata tip 41: Monitoring loop iterations

David A. Harrison
Intensive Care National Audit & Research Centre
London, UK
david.harrison@icnarc.org

If, like me, you have ever started a Stata program running and returned hours later to find it still running with no idea of whether it is actually getting anywhere, then you will be looking for a simple method to monitor your loops. The commands bootstrap and jackknife produce an attractive table of dots for just this purpose by using the undocumented command dots.

Any loop can easily be modified to report its progress by using dots:

```stata
nois _dots 0, title(Loop running) reps(100)
forvalues i = 1/100 {
    (main body of loop)
    nois _dots 'i' 0
}
Loop running (100)

1 2 3 4 5
.................................................. 50
.................................................. 100
```

The first dots command, called with the argument 0, sets up the graduated header line. The title and number of repetitions are optional. Further calls to dots take two arguments: the repetition number and a return code. The return code 0 (as used above) indicates a successful repetition, and a dot is displayed. Alternative return codes produce a green 's' (−1) or a red 'x' (1), 'e' (2), 'n' (3) or '?' (any other value).

Below is a more complicated example using a while loop. Here, the loop runs until 70 successes are achieved. For this contrived example, each iteration succeeds at random with 80% probability. Successes are reported with a dot (.) and failures with an x.

```stata
nois _dots 0, title(Looping until 70 successes...) local rep 1
local nsuccess 0
while 'nsuccess' < 70 {
    local fail = uniform() < .2
    local nsuccess = 'nsuccess' + ('fail' == 0)
    nois _dots 'rep++' 'fail'
}
Looping until 70 successes...

1 2 3 4 5
xx...x.x...xx...x.x...x..............x........ 50
.x..............x.x.x.x...x.......
```

© 2007 StataCorp LP  pr0030