Using remote access to big datasets efficiently with Stata

2009 German Stata Users Group meeting, Bonn

Volker Lang, M.A. Dipl.-Vw.
Eberhard-Karls-University Tübingen
Institut of Sociology
v.lang@uni-tuebingen.de

26-06-2009
Using remote access to big datasets efficiently with Stata

1 Data used with remote access

**LIAB:** Linked-Employer-Employee-Data of the Institut for Employment Research (IAB) in Nuremberg (cf. Jacobebbinghaus, 2008; Alda et al., 2005)

- Longitudinal data of German firms and their employees covering the timespan between 1993 and 2006.
- Consists of waves of the IAB firm panel (“IAB-Betriebspanel”) and waves of the IAB employee sample (“IAB-Beschäftigtenstichprobe”).
1 Anonymisation of the output

- Typically remote access is implemented on survey or process generated data for privacy reasons. Therefore parts of the output are typically anonymised.

- In case of the LIAB every suppopulation of the data smaller then 20 observational units is blanked in the output submitted to the users. Since the LIAB data set contains a huge number of cases (see later) one typically only runs into problems with that when analysing very rare strata.

- For the same reason graphs are only submitted to the users if their are saved including the option `asis`. 
2 Using ado-files

- For security reasons the servers hosting the data are typically not directly connected to the internet. In case of the LIAB one sends his do-files to the FDZ of the IAB and they pass it on to the server.

- This implies that you cannot directly call and install ado-files from the internet. However on request it is possible to get them installed.

- I ran into problems with that when I tried to use the scheme lean with graphs and the package parmest.

- These problems can be solved through communication.
3 Size of the data

- Typically remote access data sets are huge. F.e. the LIAB contains information on about 2 million employees. Using them for event history analysis as in my case this can add up to more than 7 million job spells and data sets of 8-10 GB size when using episode splitting on this spells.

- Therefore running do-files on the whole data set can be very slow and sometimes even cause convergence problems with models.

- I ran into problems with that often in the beginning using `stcox` and `streg`.

- The obvious solution to that problem is using `sample` or `bsample` before estimating the models. That is fine but would not exploit one big advantage of huge data sets ..
3 Size of the data

- Huge data sets typically contain large case numbers even in rare strata.
- Drawing a random sample of such a dataset would typically reproduce the distribution of the original data but with relatively smaller case numbers.
- In the subsample absolute case numbers in rare strata can become so small that one runs into technical difficulties estimating models on them. (A problem similar to smaller data sets.)
- But with a huge data set there is an alternative: **Sampling equal sized strata.**
1 Sampling equal sized strata

Basically this means ..

1. Using the case number information which would be produced by a n-way cross-tabulation of the variables used for stratification.

2. Use a function to find the minimum case number in that matrix.

3. Plug that information into the sampler used.

I wrote a program called `samplegr` preforming these 3 steps. The user only has to specify the variables he wants to use for stratification.
2 samplegr

capture program drop samplegr

program samplegr, sortpreserve
syntax varlist [if] [in], [WITHreplacemenent]]
    marksample touse
    drop if `touse' != 1
    quietly {
        tempvar N
        bysort `varlist': generate long `N' = _N
        summarize `N', meanonly
        local minN = r(min)
        drop `N'
        if "`WITHreplacemenent'" == ""{
            sample `minN', count by(`varlist')
        }
        else {
            bsample `minN', strata(`varlist')
        }
    }
end
Using remote access to big datasets efficiently with Stata

1. Data used with remote access
2. Things to consider using remote access
3. Sampling from huge data sets
4. Discussion

3. Used on car data

```
. program samplegr, sortpreserve
1. syntax varlist [if] [in], [withreplacement seed(numlist integer > 0 max - 1)]
2. marksample touse
3. drop if touse != 1
4. quietly {
5.   tempvar N
6.   bysort varlist: generate long `N' = _N
7.   summarize `N', meanonly
8.   local minN = r(min)
9.   drop `N'
10.   if "withreplacement" == "" {
11.     if "seed" != "" {
12.         set seed `seed'
13.     }
14.     
15.     }
16.   else {
17.     if "seed" != "" {
18.         set seed `seed'
19.     }
20.     }
21.   bsample `minN', count by(`varlist')
22. }
23. }
24. end

: sysuse auto, clear
(1978 Automobile Data)

: recode rep78 (2/3 = 1) (4 = 2) (5 = 3)
(rep78: 67 changes made)

: sample rep78, foreign rep78 (2334)
(5 observations deleted)

: sort rep78 foreign

: list make rep78 foreign

<table>
<thead>
<tr>
<th>make</th>
<th>rep78</th>
<th>foreign</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dodge St. Regs</td>
<td>1</td>
<td>Domestic</td>
</tr>
<tr>
<td>Olds cutl Supr</td>
<td>1</td>
<td>Domestic</td>
</tr>
<tr>
<td>Renault Le Car</td>
<td>1</td>
<td>Foreign</td>
</tr>
<tr>
<td>Audi Fox</td>
<td>1</td>
<td>Foreign</td>
</tr>
<tr>
<td>Merc. XR-7</td>
<td>2</td>
<td>Domestic</td>
</tr>
<tr>
<td>Chev. Impala</td>
<td>2</td>
<td>Domestic</td>
</tr>
<tr>
<td>Datsun 810</td>
<td>2</td>
<td>Foreign</td>
</tr>
<tr>
<td>Honda Civic</td>
<td>2</td>
<td>Foreign</td>
</tr>
<tr>
<td>Dodge Colt</td>
<td>3</td>
<td>Domestic</td>
</tr>
<tr>
<td>Plyn. Champ</td>
<td>3</td>
<td>Domestic</td>
</tr>
<tr>
<td>Honda Accord</td>
<td>3</td>
<td>Foreign</td>
</tr>
<tr>
<td>Toyota Celica</td>
<td>3</td>
<td>Foreign</td>
</tr>
</tbody>
</table>
```
4 Setting the seed

**Problem:** For reasons I haven’t figured out yet `samplegr` doesn’t get the information of the global stored if a `set seed` command is used in the code.

To solve the problem I tried to plug the `set seed` command into the program using an option to be specified in the syntax by the user (see next slide). Unfortunately that doesn’t work out!
Using remote access to big datasets efficiently with Stata

1 Data used with remote access
2 Things to consider using remote access
3 Sampling from huge data sets
4 Discussion

5 samplegr with set seed

capture program drop samplegr

program samplegr, sortpreserve
syntax varlist [if] [in], [WITHreplacement seed(numlist integer > 0 max = 1)]
marksample touse
  drop if `touse' != 1
  quietly {
    tempvar N
    bysort `varlist': generate long `N' = _N
    summarize `N', meanonly
    local minN = r(min)
    drop `N'
    if "`withreplacement'" == "" {
      if "`seed'" != "" {
        set seed `seed'
      }
      sample `minN', count by(`varlist')
    }
    else {
      if "`seed'" != "" {
        set seed `seed'
      }
      bsample `minN', strata(`varlist')
    }
  }
end
4 Questions for discussion

1. Do you have any suggestions on how to fix the problem combining `samplegr` with `set seed`?

2. Do you have any suggestions on how to improve or extend the program in other ways?

3. Do you think that `samplegr` can be useful for other people?