6 Managing memory

Contents

6.1 Memory-size considerations

Stata works with a copy of data that it loads into memory. Memory allocation is automatic.

Stata automatically sizes itself up and down as your session progresses. Stata obtains memory from the operating system and draws no distinction between real and virtual memory. Virtual memory is memory that resides on disk that operating systems supply when physical memory runs short. Virtual memory is slow but adequate in cases when you have a dataset that is too large to load into real memory. If you wish to limit the maximum amount of memory Stata can use, you can set `max_memory`; see [D] **memory**. If you use the Linux operating system, we strongly suggest you set `max_memory`; see *Serious bug in Linux OS* in [D] **memory**.

6.2 Compressing data

Stata stores data in memory. The `compress` command reduces the amount of memory required to store the data without loss of precision or any other disadvantages; see [D] **compress**. Typing `compress` every so often is a good idea.

`compress` works by examining the values you have stored and changing the data types of variables when that can be done without loss of precision. For instance, you may have a variable stored as `float` but that records only integer values between −127 and 100. `compress` would change the storage type of that variable to `byte` and save 3 bytes per observation. If you had 100 variables like that, the savings would be 300 bytes per observation, and if you had 3,000,000 observations, the total savings would be nearly 900 megabytes.

6.3 Setting maxvar

If you get the error message "no room to add more variables", r(901), do not jump to the conclusion that you have exceeded Stata's capacity.

`maxvar` specifies the maximum number of variables you can use. The default setting depends on whether you are using Stata/MP, Stata/SE, or Stata/IC. To determine the current setting, type `query memory` at the Stata prompt.

If you use Stata/MP, you can reset this maximum number to 120,000. If you use Stata/SE, you can reset this maximum number to 32,767. Set `maxvar` to more than you need—at least 20 more than you need but not too much more than you need. Figure that each 10,000 variables consumes roughly 0.5 megabytes of memory.

You reset maxvar using the `set maxvar` command,

`set maxvar #` [`, permanently`]

where 2,048 $\leq$ # $\leq$ 120,000, depending on your flavor of Stata. You can reset `maxvar` repeatedly during a session. If you specify the `permanently` option, you change `maxvar` not only for this session but also for future sessions. Each additional 10,000 variables specified with `set maxvar` requires Stata to set aside roughly 1.3 megabytes of memory for variable names, not including the data stored in those variables.

6.4 Setting matsize

You may issue an estimation command and obtain the error message "matsize too small", r(908). Stata uses matrices in making many calculations. `matsize` specifies the maximum size of those matrices in terms of (roughly speaking) the number of estimated coefficients. The default value of `matsize` is 400. `matsize` can be set to any value between 10 and 11,000, inclusive. The command is

`set matsize #` [`, permanently`]

where 10 $\leq$ # $\leq$ 11,000, depending on your flavor of Stata.

Increasing `matsize` increases Stata's memory consumption:

matsize	memory use
400	1.254M
800	4.950M
1,600	19.666M
3,200	78.394M
6,400	313.037M
11,000	924.080M

The table above understates the amount of memory Stata will use. The table was derived under the assumption of one matrix and eleven vectors. If two matrices are required, the numbers above would be nearly doubled.

If you use a 32-bit computer, you likely will be unable to set `matsize` to 11,000. A value of 11,000 would require nearly 1 gigabyte per matrix. The total memory consumption most 32-bit operating systems will grant to Stata is 2 gigabytes, so if you had two matrices, there would be no memory left for data or for Stata's code!

You should not `set matsize` larger than is necessary. Doing so will at best waste memory and at worst slow Stata down or prevent Stata from having enough memory for other tasks. If you receive the error message "matsize too small", increase `matsize` only as much as is necessary to eliminate the error message.

6.5 The memory command

The memory command will show you the major components of Stata's memory footprint.

```
. use http://www.stata-press.com/data/r15/regsmpl
(NLS Women 14-26 in 1968)

. memory
```

Memory usage

	used	allocated
data	856,020	67,108,864
strLs	0	0
data & strLs	856,020	67,108,864
data & strLs	856,020	67,108,864
var. names, %fmts, ...	4,436	67,327
overhead	2,081,344	1,082,136
Stata matrices	0	0
ado-files	9,429	9,429
stored results	0	0
Mata matrices	0	0
Mata functions	0	0
set maxvar usage	2,164,426	2,164,426
other	3,282	3,282
grand total	4,114,061	70,435,464

See [D] **memory**.