Title

stata.com

constraint — Define and list constraints

Syntax

Define constraints

```
constraint [define] # [exp=exp | coeflist]
```

List constraints

```
constraint dir [numlist | _all]
```

```
constraint list [numlist | _all]
```

Drop constraints

```
constraint drop {numlist | _all}
```

Programmer's commands

```
constraint get #
```

```
constraint free
```

where *coeflist* is as defined in [R] **test** and # is restricted to the range 1–1,999, inclusive.

Menu

Statistics > Other > Manage constraints

Description

`constraint` defines, lists, and drops linear constraints. Constraints are for use by models that allow constrained estimation.

Constraints are defined by the `constraint` command. The currently defined constraints can be listed by either `constraint list` or `constraint dir`; both do the same thing. Existing constraints can be eliminated by `constraint drop`.

`constraint get` and `constraint free` are programmer's commands. `constraint get` returns the contents of the specified constraint in macro `r(contents)` and returns in scalar `r(defined)` 0 or 1—1 being returned if the constraint was defined. `constraint free` returns the number of a free (unused) constraint in macro `r(free)`.

Remarks and examples

stata.com

Using constraints is discussed in [R] **cnsreg**, [R] **mlogit**, and [R] **reg3**; this entry is concerned only with practical aspects of defining and manipulating constraints.

▷ Example 1

Constraints are numbered from 1 to 1,999, and we assign the number when we define the constraint:

```
. use http://www.stata-press.com/data/r13/sysdsn1
(Health insurance data)
. constraint 2 [indemnity]2.site = 0
```

The currently defined constraints can be listed by `constraint list`:

```
. constraint list
     2:  [indemnity]2.site = 0
```

`constraint drop` drops constraints:

```
. constraint drop 2
. constraint list
```

The empty list after `constraint list` indicates that no constraints are defined. Below we demonstrate the various syntaxes allowed by `constraint`:

```
. constraint 1 [Indemnity]
. constraint 10 [Indemnity]: 1.site 2.site
. constraint 11 [Indemnity]: 3.site
. constraint 21 [Prepaid=Uninsure]: nonwhite
. constraint 30 [Prepaid]
. constraint 31 [Insure]
. constraint list
     1:  [Indemnity]
    10:  [Indemnity]: 1.site 2.site
    11:  [Indemnity]: 3.site
    21:  [Prepaid=Uninsure]: nonwhite
    30:  [Prepaid]
    31:  [Insure]
. constraint drop 21-25, 31
. constraint list
     1:  [Indemnity]
    10:  [Indemnity]: 1.site 2.site
    11:  [Indemnity]: 3.site
    30:  [Prepaid]
. constraint drop _all
. constraint list
```

◁

❑ Technical note

The `constraint` command does not check the syntax of the constraint itself because a constraint can be interpreted only in the context of a model. Thus `constraint` is willing to define constraints that later will not make sense. Any errors in the constraints will be detected and mentioned at the time of estimation.

❑

References

Buis, M. L. 2012. Stata tip 108: On adding and constraining. *Stata Journal* 12: 342–344.

Weesie, J. 1999. sg100: Two-stage linear constrained estimation. *Stata Technical Bulletin* 47: 24–30. Reprinted in *Stata Technical Bulletin Reprints*, vol. 8, pp. 217–225. College Station, TX: Stata Press.

Also see

[R] **cnsreg** — Constrained linear regression