vl drop — Drop variable lists or variables from variable lists

Description Quick start Syntax Options Remarks and examples Also see

Description

`vl drop` *vlusername* deletes user-defined variable lists.

`vl drop` *vlsysname* zeros system-defined variable lists. They still exist but are empty.

`vl drop (`*varlist*`)` removes variables from all variable lists.

`vl clear` deletes all variable lists and removes all traces of the `vl` system.

For an introduction to the `vl` commands, see [D] **vl**.

Quick start

Delete the user-defined variable list `myfav`

```
vl drop myfav
```

Zero the system-defined variable list `vluncertain`

```
vl drop vluncertain
```

Drop the variables `x1` and `x2` from all variable lists

```
vl drop (x1 x2)
```

Same as above, but only drop them from user-defined variable lists

```
vl drop (x1 x2), user
```

Delete all variable lists and all traces of the `vl` system

```
vl clear
```

Delete all user-defined variable lists

```
vl clear, user
```

Delete all system-defined variable lists and the stored variable statistics

```
vl clear, system
```

Syntax

Drop variable lists

```
vl drop vlnamelist [ , system user ]
```

Drop variables from variable lists

```
vl drop (varlist) [ , system user ]
```

Clear all variable lists

```
vl clear [ , system user ]
```

vlnamelist is a list of variable-list names.

`(_all)` or `(*)` can be used to specify all numeric variables in the dataset.

Options

`system` when specified with `vl drop (varlist)`, drops the variables in *varlist* only from system-defined variable lists. By default, variables are dropped from all variable lists, both system-defined and user-defined.

When specified with `vl clear`, only the system-defined variable lists are deleted. By default, both the system-defined and user-defined variable lists are deleted, and all traces of the `vl` system are gone.

`user` when specified with `vl drop (varlist)`, drops the variables in *varlist* only from user-defined variable lists.

When specified with `vl clear`, only the user-defined variable lists are deleted.

Remarks and examples

When given one or more names of user-defined variable lists, `vl drop` deletes them. That is, typing

```
. vl drop myname
```

deletes the user-defined variable list `myname`. It is as if `myname` was never created. A new variable list called `myname` can now be created using `vl create`.

When given one or more names of system-defined variable lists, `vl drop` zeros them. That is, typing

```
. vl drop vluncertain
```

zeros the system-defined variable list `vluncertain`. It still exists but is empty. A single system-defined variable list cannot be deleted.

All system-defined variable lists can be deleted using

```
. vl clear, system
```

All system-defined variable lists are now gone. Also deleted are the stored variable statistics, namely, the number of levels, minimum and maximum values, and the number of nonmissing observations. It is as if `vl set` was never run.

Typing

```
. vl clear
```

deletes all variable lists and all traces of the `vl` system.

Typing

```
. vl drop (varlist)
```

removes the variables in *varlist* from all variable lists.

Say we only want to remove variable `x8` from the user-defined variable list `mylist`. To do this, we type

```
. vl modify mylist = mylist - (x8)
```

Note the parentheses around `x8`; see [D] **vl create**.

Say you want to remove variable `x8` from the system-defined variable list `vlcategorical`. System-defined variable lists are disjoint, so a variable is only in one of them. Thus, we can remove it by typing

```
. vl drop (x8), system
```

Rather than drop it, we could have moved it to the system-defined variable list `vlother`.

```
. vl move (x8) vlother
```

See [D] **vl set**.

Also see

[D] **vl** — Manage variable lists

[D] **vl create** — Create and modify user-defined variable lists

[D] **vl list** — List contents of variable lists

[D] **vl rebuild** — Rebuild variable lists

[D] **vl set** — Set system-defined variable lists