vl list — List conte	nts of variable lists	3			
Description Stored result	Quick start s Also see	Syntax	Options	Remarks and examples	

Description

vl list shows the contents of variable lists when given names of variable lists. When given names of variables, it shows the variable lists to which each variable belongs.

vl dir shows the names of all variable lists.

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

Quick start

Show the contents of all variable lists

vllist

Show the contents of the system-defined variable list vlcategorical

vl list vlcategorical

Show the contents of the user-defined variable list myfav

vl list myfav

Show the variable lists to which x1-x100 belong

vllist (x1-x100)

Show the variable lists to which every numeric variable belongs v1 list (*)

Show the contents of all system-defined variable lists vl list, system

Show the contents of all user-defined variable lists

vllist, user

Show the contents of all variable lists, and show the minimum value, maximum value, and number of nonmissing values for each variable

vl list, minimum maximum observations

Show the contents of all variable lists, ordered by variable list and then alphabetically by variable name vllist, sort

Show the variable lists to which every numeric variable belongs, ordered alphabetically by variable name and then by variable list

vllist(*), sort

Syntax

Show the contents of variable lists

vl list [vlnamelist] [, options]

Show the variable lists to which variables belong

vllist (varlist) [, options]

Show names of all variable lists

vldir[, system user]

vlnamelist is a list of variable-list names.

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

options	Description	
system	show only system-defined variable lists	
user	show only user-defined variable lists	
<u>min</u> imum	show minimum value of each variable	
<u>max</u> imum	show maximum value of each variable	
<u>obs</u> ervations	show number of nonmissing observations of each variable	
sort	order by variable list and then alphabetically by variable name when <i>vlnamelist</i> is specified; order alphabetically by variable name and then by variable list when (<i>varlist</i>) is specified	
strok	allow string variables when (varlist) is specified	
nolstretch	do not stretch the width of the table to accommodate long names	

collect is allowed with vl list and vl dir; see [U] 11.1.10 Prefix commands.

Options

system specifies that only system-defined variable lists be shown. By default, both system-defined and user-defined variable lists are shown.

user specifies that only user-defined variable lists be shown.

minimum specifies that the minimum value of each variable be displayed.

maximum specifies that the minimum value of each variable be displayed.

observations specifies that number of nonmissing observations of each variable be displayed.

sort specifies that the listing be sorted. When *vlnamelist* is specified, the listing is ordered by variable list and then alphabetically by variable name. By default in this case, variables are listed in the order in which they were added to the variable list.

When (*varlist*) is specified, the listing is ordered alphabetically by variable name and then by variable list. By default in this case, variables are listed in the order in which they appear in *varlist*.

strok specifies that string variables be included in the listing when (*varlist*) is specified. By default, specifying string variables in *varlist* gives an error message. Specifying strok prevents this error message and lists any string variables.

nolstretch specifies that the width of the table not be automatically widened to accommodate long variable and variable-list names. When nolstretch is specified, names are abbreviated to make the table width no more than 79 characters. The default, lstretch, is to automatically widen the table up to the width of the Results window. To change the default, use set lstretch off.

Remarks and examples

vllist produces two types of listings. The first lists by variable-list name and then by variable name. The second is the reverse; it lists by variable name and then by variable-list name.

Typing

. vl list

produces the first type of listing. This listing is useful when you want to see the contents of each variable list.

Typing

. vl list (*)

or

. vl list (x1-x100)

produces the second type of listing. This listing is useful when you want to see all variable lists to which a variable belongs.

System-defined variable lists are disjoint, so a variable can only belong to one of them. There is no such restriction on user-defined variable lists. Variables can belong to more than one user-defined variable list.

Typing

. vl dir

shows all the variable lists, both system-defined and user-defined. The options system and user work with both vllist and vldir to restrict the output accordingly.

Example 1: Showing the contents of variable lists

We show examples using Stata's automobile dataset because it has only a small number of variables and the output will not be too lengthy.

```
. sysuse auto
(1978 automobile data)
```

We run vl set with the option nonotes to suppress the notes at the end of the output.

. vl set, nonotes

	Macro's contents	
Macro	# Vars	Description
System		
\$vlcategorical	2	categorical variables
\$vlcontinuous	2	continuous variables
<pre>\$vluncertain</pre>	7	perhaps continuous, perhaps categorical variables
\$vlother	0	all missing or constant variables

Let's list the contents of the variable lists.

. vl list

Variable	Macro	Values	Levels
rep78	<pre>\$vlcategorical</pre>	integers >=0	5
foreign	<pre>\$vlcategorical</pre>	0 and 1	2
headroom	\$vlcontinuous	noninteger	
gear_ratio	<pre>\$vlcontinuous</pre>	noninteger	
price	<pre>\$vluncertain</pre>	integers >=0	74
mpg	<pre>\$vluncertain</pre>	integers >=0	21
trunk	<pre>\$vluncertain</pre>	integers >=0	18
weight	<pre>\$vluncertain</pre>	integers >=0	64
length	<pre>\$vluncertain</pre>	integers >=0	47
turn	<pre>\$vluncertain</pre>	integers >=0	18
displacement	<pre>\$vluncertain</pre>	integers >=0	31

We decide to treat all the variables in vluncertain as continuous, so we move them to vlcontinuous. Then we run vl dir to confirm that vluncertain is empty.

. vl move vluncertain vlcontinuous note: 7 variables specified and 7 variables moved.

Macro	# Added/Removed
<pre>\$vlcategorical</pre>	0
\$vlcontinuous	7
<pre>\$vluncertain</pre>	-7
\$vlother	0

. vl dir

	Macro's contents		
Macro	# Vars	Description	
System			
<pre>\$vlcategorical</pre>	2	categorical variables	
\$vlcontinuous	9	continuous variables	
<pre>\$vluncertain</pre>	0	perhaps continuous, perhaps categorical variables	
\$vlother	0	all missing or constant variables	

Let's create two user-defined variable lists.

. vl create power = (gear_ratio weight displacement)
note: \$power initialized with 3 variables.
. vl create other = (price turn length)
note: \$other initialized with 3 variables.

Let's do a listing ordered by variable list. We specify options to see the minimum and maximum values and the number of nonmissing observations for each variable.

Variable	Macro	Values	Levels	Min	Max	Obs
rep78	<pre>\$vlcategorical</pre>	integers >=0	5	1	5	69
foreign	<pre>\$vlcategorical</pre>	0 and 1	2	0	1	74
headroom	\$vlcontinuous	noninteger		1.5	5	74
gear_ratio	<pre>\$vlcontinuous</pre>	noninteger		2.19	3.89	74
price	<pre>\$vlcontinuous</pre>	integers >=0	74	3291	15906	74
mpg	<pre>\$vlcontinuous</pre>	integers >=0	21	12	41	74
trunk	<pre>\$vlcontinuous</pre>	integers >=0	18	5	23	74
weight	<pre>\$vlcontinuous</pre>	integers >=0	64	1760	4840	74
length	<pre>\$vlcontinuous</pre>	integers >=0	47	142	233	74
turn	<pre>\$vlcontinuous</pre>	integers >=0	18	31	51	74
displacement	<pre>\$vlcontinuous</pre>	integers >=0	31	79	425	74
gear_ratio	\$power	noninteger		2.19	3.89	74
weight	\$power	integers >=0	64	1760	4840	74
displacement	\$power	integers >=0	31	79	425	74
price	\$other	integers >=0	74	3291	15906	74
turn	\$other	integers >=0	18	31	51	74
length	\$other	integers >=0	47	142	233	74
	1					

. vl list, minimum maximum observations

Specifying (*) means that we want a listing ordered by variable name.

. vl list (*)

Variable	Macro	Values	Levels
price	\$vlcontinuous	integers >=0	74
price	\$other	integers >=0	74
mpg	<pre>\$vlcontinuous</pre>	integers >=0	21
mpg	not in vluser		21
rep78	<pre>\$vlcategorical</pre>	integers >=0	5
rep78	not in vluser		5
headroom	<pre>\$vlcontinuous</pre>	noninteger	
headroom	not in vluser		
trunk	<pre>\$vlcontinuous</pre>	integers >=0	18
trunk	not in vluser		18
weight	<pre>\$vlcontinuous</pre>	integers >=0	64
weight	\$power	integers >=0	64
length	<pre>\$vlcontinuous</pre>	integers >=0	47
length	\$other	integers >=0	47
turn	<pre>\$vlcontinuous</pre>	integers >=0	18
turn	\$other	integers >=0	18
displacement	<pre>\$vlcontinuous</pre>	integers >=0	31
displacement	\$power	integers >=0	31
gear_ratio	<pre>\$vlcontinuous</pre>	noninteger	
gear_ratio	\$power	noninteger	
foreign	<pre>\$vlcategorical</pre>	0 and 1	2
foreign	not in vluser		2

Variables are listed multiple times showing all the variable lists to which each belongs. We can restrict the listing to user-defined variable lists.

. vl list (*), user

Variable	Macro	Values	Levels
price	\$other	integers >=0	74
mpg	not in vluser		21
rep78	not in vluser		5
headroom	not in vluser		
trunk	not in vluser		18
weight	\$power	integers >=0	64
length	\$other	integers >=0	47
turn	\$other	integers >=0	18
displacement	\$power	integers >=0	31
gear_ratio	\$power	noninteger	
foreign	not in vluser	0	2

See the lines "not in vluser"? They are omitted if you run vl list, user.

Let's use vl substitute with factor-variable operators to create interactions between the variables in the system-defined variable list, vlcategorical, and the variables in our user-defined variable list, mycontinuous.

. vl substitute indepvars = i.vlcategorical##c.(power other)

The factor-variable list indepvars shows up when we run vl dir.

	Macro's contents		
Macro	# Vars	Description	
System			
<pre>\$vlcategorical</pre>	2	categorical variables	
\$vlcontinuous	9	continuous variables	
<pre>\$vluncertain</pre>	0	perhaps continuous, perhaps categorical variables	
\$vlother	0	all missing or constant variables	
User		-	
\$power	3	variables	
\$other	3	variables	
<pre>\$indepvars</pre>		factor-variable list	

Factor-variable lists do not work with vllist. But you can display their contents because variable lists are global macros. You can list the contents of a variable list by typing

. display "\$indepvars"

i.rep78 i.foreign gear_ratio weight displacement price turn length i.rep78#c.gear_r

> atio i.rep78#c.weight i.rep78#c.displacement i.rep78#c.price i.rep78#c.turn i.rep

> 78#c.length i.foreign#c.gear_ratio i.foreign#c.weight i.foreign#c.displacement i.

> foreign#c.price i.foreign#c.turn i.foreign#c.length

Stored results

```
vl list stores the following in r():
```

```
Scalars
```

	r(k)	number of variables listed
	r(k_system)	number of variables listed in system-defined variable lists
	r(k_not_system)	number of variables listed not in system-defined variable lists
	r(k_vlcategorical)	number of variables listed in vlcategorical
	r(k_vlcontinuous)	number of variables listed in vlcontinuous
	r(k_vluncertain)	number of variables listed in vluncertain
	r(k_vlother)	number of variables listed in vlother
	r(k_vldummy)	number of variables listed in vldummy when defined
	r(k_user)	number of variables listed in user-defined variable lists
	r(k_not_user)	number of variables listed not in user-defined variable lists
	r(k_vlusername)	number of variables listed in vlusername
	r(k_string)	number of string variables listed when strok specified
a	cros	
	r(vlsysnames)	names of all system-defined variable lists
	r(vlusernames)	names of all user-defined variable lists

vl dir stores the following in r():

Scalars

Μ

```
r(k_system)
                             number of variables in system-defined variable lists
                             number of variables in vlcategorical
    r(k_vlcategorical)
                             number of variables in vlcontinuous
    r(k_vlcontinuous)
                             number of variables in vluncertain
    r(k_vluncertain)
    r(k_vlother)
                             number of variables in vlother
                             number of variables in vldummy when defined
    r(k_vldummy)
    r(k_user)
                             number of variables in user-defined variable lists
                             number of variables in vlusername
    r(k_vlusername)
Macros
    r(vlsysnames)
                             names of system-defined variable lists
                             names of user-defined variable lists
    r(vlusernames)
```

Also see

- [D] vl Manage variable lists
- [D] vl create Create and modify user-defined variable lists
- [D] vl drop Drop variable lists or variables from variable lists
- [D] vl rebuild Rebuild variable lists
- [D] vl set Set system-defined variable lists

Stata, Stata Press, and Mata are registered trademarks of StataCorp LLC. Stata and Stata Press are registered trademarks with the World Intellectual Property Organization of the United Nations. StataNow and NetCourseNow are trademarks of StataCorp LLC. Other brand and product names are registered trademarks or trademarks of their respective companies. Copyright © 1985–2025 StataCorp LLC, College Station, TX, USA. All rights reserved.



For suggested citations, see the FAQ on citing Stata documentation.