Title

stata.com

duplicates — Report, tag, or drop duplicate observations

Syntax

Report duplicates

```
duplicates report [varlist] [if] [in]
```

List one example for each group of duplicates

```
duplicates examples [varlist] [if] [in] [, options]
```

List all duplicates

```
duplicates list [varlist] [if] [in] [, options]
```

Tag duplicates

```
duplicates tag [varlist] [if] [in], generate(newvar)
```

Drop duplicates

```
duplicates drop [if] [in]
```

```
duplicates drop varlist [if] [in], force
```

options	Description
Main	
compress	compress width of columns in both table and display formats
nocompress	use display format of each variable
fast	synonym for nocompress; no delay in output of large datasets
abbreviate(#)	abbreviate variable names to # characters; default is ab(8)
string(#)	truncate string variables to # characters; default is string(10)
Options	
table	force table format
display	force display format
header	display variable header once; default is table mode
noheader	suppress variable header
header(#)	display variable header every # lines
clean	force table format with no divider or separator lines
divider	draw divider lines between columns
separator(#)	draw a separator line every # lines; default is separator(5)
sepby(*varlist*)	draw a separator line whenever *varlist* values change
nolabel	display numeric codes rather than label values
Summary	
mean[(*varlist*)]	add line reporting the mean for each of the (specified) variables
sum[(*varlist*)]	add line reporting the sum for each of the (specified) variables
N[(*varlist*)]	add line reporting the number of nonmissing values for each of the (specified) variables
labvar(*varname*)	substitute Mean, Sum, or N for value of *varname* in last row of table
Advanced	
constant[(*varlist*)]	separate and list variables that are constant only once
notrim	suppress string trimming
absolute	display overall observation numbers when using by *varlist*:
nodotz	display numerical values equal to .z as field of blanks
subvarname	substitute characteristic for variable name in header
linesize(#)	columns per line; default is linesize(79)

Menu

Data > Data utilities > Manage duplicate observations

Description

duplicates reports, displays, lists, tags, or drops duplicate observations, depending on the subcommand specified. Duplicates are observations with identical values either on all variables if no *varlist* is specified or on a specified *varlist*.

duplicates report produces a table showing observations that occur as one or more copies and indicating how many observations are "surplus" in the sense that they are the second (third, . . .) copy of the first of each group of duplicates.

`duplicates examples` lists one example for each group of duplicated observations. Each example represents the first occurrence of each group in the dataset.

`duplicates list` lists all duplicated observations.

`duplicates tag` generates a variable representing the number of duplicates for each observation. This will be 0 for all unique observations.

`duplicates drop` drops all but the first occurrence of each group of duplicated observations. The word `drop` may not be abbreviated.

Any observations that do not satisfy specified `if` and/or `in` conditions are ignored when you use `report`, `examples`, `list`, or `drop`. The variable created by `tag` will have missing values for such observations.

Options

Options are presented under the following headings:

Options for duplicates examples and duplicates list

Main

`compress`, `nocompress`, `fast`, `abbreviate(#)`, `string(#)`; see [D] **list**.

Options

`table`, `display`, `header`, `noheader`, `header(#)`, `clean`, `divider`, `separator(#)`, `sepby(`*varlist*`)`, `nolabel`; see [D] **list**.

Summary

`mean[(`*varlist*`)]`, `sum[(`*varlist*`)]`, `N[(`*varlist*`)]`, `labvar(`*varname*`)`; see [D] **list**.

Advanced

`constant[(`*varlist*`)]`, `notrim`, `absolute`, `nodotz`, `subvarname`, `linesize(#)`; see [D] **list**.

Option for duplicates tag

`generate(`*newvar*`)` is required and specifies the name of a new variable that will tag duplicates.

Option for duplicates drop

`force` specifies that observations duplicated with respect to a named *varlist* be dropped. The `force` option is required when such a *varlist* is given as a reminder that information may be lost by dropping observations, given that those observations may differ on any variable not included in *varlist*.

Remarks and examples

stata.com

Current data management and analysis may hinge on detecting (and sometimes dropping) duplicate observations. In Stata terms, *duplicates* are observations with identical values, either on all variables if no *varlist* is specified, or on a specified *varlist*; that is, 2 or more observations that are identical on all specified variables form a group of duplicates. When the specified variables are a set of explanatory variables, such a group is often called a *covariate pattern* or a *covariate class*.

Linguistic purists will point out that duplicate observations are strictly only those that occur in pairs, and they might prefer a more literal term, although the most obvious replacement, "replicates", already has another statistical meaning. However, the looser term appears in practice to be much more frequently used for this purpose and to be as easy to understand.

Observations may occur as duplicates through some error; for example, the same observations might have been entered more than once into your dataset. In contrast, some researchers deliberately enter a dataset twice. Each entry is a check on the other, and all observations should occur as identical pairs, assuming that one or more variables identify unique records. If there is just one copy, or more than two copies, there has been an error in data entry.

Or duplicate observations may also arise simply because some observations just happen to be identical, which is especially likely with categorical variables or large datasets. In this second situation, consider whether `contract`, which automatically produces a count of each distinct set of observations, is more appropriate for your problem. See [D] **contract**.

Observations unique on all variables in *varlist* occur as single copies. Thus there are no surplus observations in the sense that no observation may be dropped without losing information about the contents of observations. (Information will inevitably be lost on the frequency of such observations. Again, if recording frequency is important to you, `contract` is the better command to use.) Observations that are duplicated twice or more occur as copies, and in each case, all but one copy may be considered surplus.

This command helps you produce a dataset, usually smaller than the original, in which each observation is *unique* (literally, each occurs only once) and *distinct* (each differs from all the others). If you are familiar with Unix systems, or with sets of Unix utilities ported to other platforms, you will know the `uniq` command, which removes duplicate adjacent lines from a file, usually as part of a pipe.

▷ Example 1

Suppose that we are given a dataset in which some observations are unique (no other observation is identical on all variables) and other observations are duplicates (in each case, at least 1 other observation exists that is identical). Imagine dropping all but 1 observation from each group of duplicates, that is, dropping the surplus observations. Now all the observations are unique. This example helps clarify the difference between 1) identifying unique observations before dropping surplus copies and 2) identifying unique observations after dropping surplus copies (whether in truth or merely in imagination). `codebook` (see [D] **codebook**) reports the number of unique values for each variable in this second sense.

Suppose that we have typed in a dataset for 200 individuals. However, a simple `describe` or `count` shows that we have 202 observations in our dataset. We guess that we may have typed in 2 observations twice. `duplicates report` gives a quick report of the occurrence of duplicates:

```
. use http://www.stata-press.com/data/r13/dupxmpl
. duplicates report
Duplicates in terms of all variables
```

copies	observations	surplus
1	198	0
2	4	2

Our hypothesis is supported: 198 observations are unique (just 1 copy of each), whereas 4 occur as duplicates (2 copies of each; in each case, 1 may be dubbed surplus). We now wish to see which observations are duplicates, so the next step is to ask for a `duplicates list`.

```
. duplicates list
Duplicates in terms of all variables
```

group:	obs:	id	x	y
1	42	42	0	2
1	43	42	0	2
2	145	144	4	4
2	146	144	4	4

The records for `id` 42 and `id` 144 were evidently entered twice. Satisfied, we now issue `duplicates drop`.

```
. duplicates drop
Duplicates in terms of all variables
(2 observations deleted)
```

◁

The `report`, `list`, and `drop` subcommands of `duplicates` are perhaps the most useful, especially for a relatively small dataset. For a larger dataset with many duplicates, a full listing may be too long to be manageable, especially as you see repetitions of the same data. `duplicates examples` gives you a more compact listing in which each group of duplicates is represented by just 1 observation, the first to occur.

A subcommand that is occasionally useful is `duplicates tag`, which generates a new variable containing the number of duplicates for each observation. Thus unique observations are tagged with value 0, and all duplicate observations are tagged with values greater than 0. For checking double data entry, in which you expect just one surplus copy for each individual record, you can generate a tag variable and then look at observations with tag not equal to 1 because both unique observations and groups with two or more surplus copies need inspection.

```
. duplicates tag, gen(tag)
Duplicates in terms of all variables
```

As of Stata 11, the `browse` subcommand is no longer available. To open duplicates in the Data Browser, use the following commands:

```
. duplicates tag, generate(newvar)
. browse if newvar > 0
```

See [D] **edit** for details on the `browse` command.

Acknowledgments

duplicates was written by Nicholas J. Cox of the Department of Geography at Durham University, UK, and coeditor of the *Stata Journal*, who in turn thanks Thomas Steichen of RJRT for ideas contributed to an earlier jointly written program (Steichen and Cox 1998).

References

Jacobs, M. 1991. dm4: A duplicated value identification program. *Stata Technical Bulletin* 4: 5. Reprinted in *Stata Technical Bulletin Reprints*, vol. 1, p. 30. College Station, TX: Stata Press.

Steichen, T. J., and N. J. Cox. 1998. dm53: Detection and deletion of duplicate observations. *Stata Technical Bulletin* 41: 2–4. Reprinted in *Stata Technical Bulletin Reprints*, vol. 7, pp. 52–55. College Station, TX: Stata Press.

Wang, D. 2000. dm77: Removing duplicate observations in a dataset. *Stata Technical Bulletin* 54: 16–17. Reprinted in *Stata Technical Bulletin Reprints*, vol. 9, pp. 87–88. College Station, TX: Stata Press.

Also see

[D] **codebook** — Describe data contents

[D] **contract** — Make dataset of frequencies and percentages

[D] **edit** — Browse or edit data with Data Editor

[D] **isid** — Check for unique identifiers

[D] **list** — List values of variables