joinby — Form all pairwise combinations within groups

Description

joinby joins, within groups formed by *varlist*, observations of the dataset in memory with *filename*, a Stata-format dataset. By *join* we mean to form all pairwise combinations. If *filename* is specified without an extension, .dta is assumed.

If *varlist* is not specified, joinby takes as *varlist* the set of variables common to the dataset in memory and in *filename*.

Observations unique to one or the other dataset are ignored unless unmatched() specifies differently. Whether you load one dataset and join the other or vice versa makes no difference in the number of resulting observations.

If there are common variables between the two datasets, however, the combined dataset will contain the values from the master data for those observations. This behavior can be modified with the update and replace options.

Quick start

Form pairwise combinations of observations from mydata1.dta in memory with those from mydata2.dta using all common variables and drop unmatched observations

    joinby using mydata2

Same as above, but join on v1, v2, and v3

    joinby v1 v2 v3 using mydata2

Same as above, but include unmatched observations only from mydata2.dta and add _merge indicating whether the variable was in both datasets or only the using dataset

    joinby v1 v2 v3 using mydata2, unmatched(using)

Same as above, but include unmatched observations only from mydata1.dta

    joinby v1 v2 v3 using mydata2, unmatched(master)

Same as above, but name the variable indicating the source of the observation newv

    joinby v1 v2 v3 using mydata2, unmatched(master) _merge(newv)

Replace missing data in mydata1.dta with values from mydata2.dta

    joinby v1 v2 v3 using mydata2, update

Replace missing and conflicting data in mydata1.dta with values from mydata2.dta

    joinby v1 v2 v3 using mydata2, update replace

Menu

Data > Combine datasets > Form all pairwise combinations within groups

Syntax

> joinby [*varlist*] using *filename* [, *options*]

options	Description

Options

When observations match:
update	replace missing data in memory with values from *filename*
replace	replace all data in memory with values from *filename*

When observations do not match:
<u>unmat</u>ched(<u>none</u>)	ignore all; the default
<u>unmat</u>ched(<u>both</u>)	include from both datasets
<u>unmat</u>ched(<u>master</u>)	include from data in memory
<u>unmat</u>ched(<u>using</u>)	include from data in *filename*
_merge(*varname*)	*varname* marks source of resulting observation; default is _merge
<u>nolabel</u>	do not copy value-label definitions from *filename*

varlist may not contain strLs.

Options

Options

update varies the action that joinby takes when an observation is matched. By default, values from the
   master data are retained when the same variables are found in both datasets. If update is specified,
   however, the values from the using dataset are retained where the master dataset contains missing.

replace, allowed with update only, specifies that nonmissing values in the master dataset be replaced
   with corresponding values from the using dataset. A nonmissing value, however, will never be re-
   placed with a missing value.

unmatched(none | both | master | using) specifies whether observations unique to one of the datasets
   are to be kept, with the variables from the other dataset set to missing. Valid values are

none	ignore all unmatched observations (default)
both	include unmatched observations from the master and using data
master	include unmatched observations from the master data
using	include unmatched observations from the using data

_merge(*varname*) specifies the name of the variable that will mark the source of the resulting obser-
   vation. The default name is _merge(_merge). To preserve compatibility with earlier versions of
   joinby, _merge is generated only if unmatched is specified.

nolabel prevents Stata from copying the value-label definitions from the dataset on disk into the dataset
   in memory. Even if you do not specify this option, label definitions from the disk dataset do not replace
   label definitions already in memory.

Remarks and examples

The following, admittedly artificial, example illustrates `joinby`.

▷ Example 1

We have two datasets: `child.dta` and `parent.dta`. Both contain a `family_id` variable, which identifies the people who belong to the same family.

```
. use https://www.stata-press.com/data/r19/child
(Data on Children)
. describe
Contains data from https://www.stata-press.com/data/r19/child.dta
 Observations:              5                  Data on Children
    Variables:              4                  11 Dec 2024 21:08
───────────────────────────────────────────────────────────────────────
Variable       Storage   Display    Value
    name          type    format    label     Variable label
───────────────────────────────────────────────────────────────────────
family_id        int      %8.0g               Family ID number
child_id         byte     %8.0g               Child ID number
x1               byte     %8.0g
x2               int      %8.0g
───────────────────────────────────────────────────────────────────────
Sorted by: family_id
. list
```

	family~d	child_id	x1	x2
1.	1025	3	11	320
2.	1025	1	12	300
3.	1025	4	10	275
4.	1026	2	13	280
5.	1027	5	15	210

```
. use https://www.stata-press.com/data/r19/parent
(Data on Parents)
. describe
Contains data from https://www.stata-press.com/data/r19/parent.dta
 Observations:              6                  Data on Parents
    Variables:              4                  11 Dec 2024 03:06
───────────────────────────────────────────────────────────────────────
Variable       Storage   Display    Value
    name          type    format    label     Variable label
───────────────────────────────────────────────────────────────────────
family_id        int      %8.0g               Family ID number
parent_id        float    %9.0g               Parent ID number
x1               float    %9.0g
x3               float    %9.0g
───────────────────────────────────────────────────────────────────────
Sorted by:
```

```
. list, sep(0)
```

	family~d	parent~d	x1	x3
1.	1030	10	39	600
2.	1025	11	20	643
3.	1025	12	27	721
4.	1026	13	30	760
5.	1026	14	26	668
6.	1030	15	32	684

We want to join the information for the parents and their children. The data on parents are in memory, and the data on children are posted at https://www.stata-press.com.

```
. joinby family_id using https://www.stata-press.com/data/r19/child
. describe
Contains data
 Observations:                   8                      Data on Parents
     Variables:                  6
```

Variable name	Storage type	Display format	Value label	Variable label
family_id	int	%8.0g		Family ID number
parent_id	float	%9.0g		Parent ID number
x1	float	%9.0g		
x3	float	%9.0g		
child_id	byte	%8.0g		Child ID number
x2	int	%8.0g		

```
Sorted by:
     Note: Dataset has changed since last saved.
. list, sepby(family_id) abbrev(12)
```

	family_id	parent_id	x1	x3	child_id	x2
1.	1025	11	20	643	3	320
2.	1025	11	20	643	4	275
3.	1025	11	20	643	1	300
4.	1025	12	27	721	1	300
5.	1025	12	27	721	3	320
6.	1025	12	27	721	4	275
7.	1026	13	30	760	2	280
8.	1026	14	26	668	2	280

1. family_id of 1027, which appears only in child.dta, and family_id of 1030, which appears only in parent.dta, are not in the combined dataset. Observations for which the matching variables are not in both datasets are omitted.

2. The x1 variable is in both datasets. Values for this variable in the joined dataset are the values from `parent.dta`—the dataset in memory when we issued the `joinby` command. If we had `child.dta` in memory and `parent.dta` on disk when we requested `joinby`, the values for x1 would have been those from `child.dta`. Values from the dataset in memory take precedence over the dataset on disk.

◁

Acknowledgment

`joinby` was written by Jeroen Weesie of the Department of Sociology at Utrecht University, The Netherlands.

References

Baum, C. F. 2016. *An Introduction to Stata Programming*. 2nd ed. College Station, TX: Stata Press.

Mazrekaj, D., and J. Wursten. 2021. Stata tip 142: joinby is the real merge m:m. *Stata Journal* 21: 1065–1068.

Also see

[D] **append** — Append datasets

[D] **cross** — Form every pairwise combination of two datasets

[D] **fillin** — Rectangularize dataset

[D] **merge** — Merge datasets

[D] **save** — Save Stata dataset

[U] **23 Combining datasets**