#### **append** — Append datasets

DescriptionQuick startMenuSyntaxOptionsRemarks and examplesReferenceAlso see

### Description

append appends Stata-format datasets stored on disk to the end of the dataset in memory. If any *filename* is specified without an extension, .dta is assumed.

Stata can also join observations from two datasets into one; see [D] merge. See [U] 23 Combining datasets for a comparison of append, merge, and joinby.

# **Quick start**

Append mydata2.dta to mydata1.dta with no data in memory append using mydata1 mydata2

Same as above, but with mydata1.dta in memory append using mydata2

Same as above, and generate newv to indicate source dataset append using mydata2, generate(newv)

Same as above, but do not copy value labels or notes from mydata2.dta append using mydata2, generate(newv) nolabel nonotes

Only keep v1, v2, and v3 from mydata2.dta append using mydata2, keep(v1 v2 v3)

## Menu

Data > Combine datasets > Append datasets

## Syntax

append using *filename* [*filename* [...]] [, options]

You may enclose *filename* in double quotes and must do so if *filename* contains blanks or other special characters.

options	Description
generate(newvar)	newvar marks source of resulting observations
keep(varlist)	keep specified variables from appending dataset(s)
<u>nol</u> abel	do not copy value-label definitions from dataset(s) on disk
nonotes	do not copy notes from dataset(s) on disk
force	append string to numeric or numeric to string without error

# Options

- generate (*newvar*) specifies the name of a variable to be created that will mark the source of observations. Observations from the master dataset (the data in memory before the append command) will contain 0 for this variable. Observations from the first using dataset will contain 1 for this variable; observations from the second using dataset will contain 2 for this variable; and so on.
- keep(varlist) specifies the variables to be kept from the using dataset. If keep() is not specified, all variables are kept.

The varlist in keep(varlist) differs from standard Stata varlists in two ways: variable names in varlist may not be abbreviated, except by the use of wildcard characters, and you may not refer to a range of variables, such as price-weight.

- nolabel prevents Stata from copying the value-label definitions from the disk dataset into the dataset in memory. Even if you do not specify this option, label definitions from the disk dataset never replace definitions already in memory.
- nonotes prevents notes in the using dataset from being incorporated into the result. The default is to incorporate notes from the using dataset that do not already appear in the master data.
- force allows string variables to be appended to numeric variables and vice versa, resulting in missing values from the using dataset. If omitted, append issues an error message; if specified, append issues a warning message.

## **Remarks and examples**

The disk dataset must be a Stata-format dataset; that is, it must have been created by save (see [D] save).

#### Example 1

We have two datasets stored on disk that we want to combine. The first dataset, called even.dta, contains the sixth through eighth positive even numbers. The second dataset, called odd.dta, contains the first five positive odd numbers. The datasets are

```
. use even
(6th through 8th even numbers)
. list
       number
                  even
             6
  1.
                    12
  2.
             7
                    14
  З.
             8
                    16
. use odd
(First five odd numbers)
. list
       number
                  odd
  1.
             1
                    1
             2
  2.
                    3
             3
                    5
  З.
  4.
             4
                    7
             5
                    9
  5.
```

We will append the even data to the end of the odd data. Because the odd data are already in memory (we just used them above), we type append using even. The result is

. append using even

. list

	number	odd	even
1.	1	1	
2. 3.	3	5	•
4. 5.	4 5	7 9	•
6.	6		12
7. 8.	7 8	:	14 16

Because the number variable is in both datasets, the variable was extended with the new data from the file even.dta. Because there is no variable called odd in the new data, the additional observations on odd were forward-filled with *missing* (.). Because there is no variable called even in the original data, the first observations on even were back-filled with missing.

### Example 2

The order of variables in the two datasets is irrelevant. Stata always appends variables by name:

. use https://www.stata-press.com/data/r19/odd1 (First five odd numbers)					
. describe					
Contains dat Observation Variable	ta from http ns: es:	ps://www.st 5 2	ata-press.	com/data/r19/odd1.dta First five odd numbers 9 Jan 2024 08:41	
Variable name	Storage type	Display format	Value label	Variable label	
odd number	float float	%9.0g %9.0g		Odd numbers	
Sorted by: r	number				
. describe ι	using https	://www.stat	a-press.co	m/data/r19/even	
Contains dat Observation Variable	ta ns: es:	3 2		6th through 8th even numbers 9 Jan 2024 08:43	
Variable name	Storage type	Display format	Value label	Variable label	
number even	byte float	%9.0g %9.0g		Even numbers	

Sorted by: number

. append using https://www.stata-press.com/data/r19/even

. list

	odd	number	even
1. 2. 3. 4. 5.	1 3 5 7 9	1 2 3 4 5	
6. 7. 8.		6 7 8	12 14 16

The results are the same as those in the first example.

When Stata appends two datasets, the definitions of the dataset in memory, called the *master* dataset, override the definitions of the dataset on disk, called the *using* dataset. This extends to value labels, variable labels, characteristics, and date-time stamps. If there are conflicts in numeric storage types, the more precise storage type will be used regardless of whether this storage type was in the *master* dataset or the *using* dataset. If a variable is stored as a string in one dataset that is longer than in the other, the longer str# storage type will prevail. If a variable is stored as a strI in one dataset and a str# in another dataset, the strL storage type will prevail.

4

### Technical note

If a variable is a string in one dataset and numeric in the other, Stata issues an error message unless the force option is specified. If force is specified, Stata issues a warning message before appending the data. If the using dataset contains the string variable, the combined dataset will have numeric missing values for the appended data on this variable; the contents of the string variable in the using dataset are ignored. If the using dataset contains the numeric variable, the combined dataset will have empty strings for the appended data on this variable; the contents of the numeric variable in the using dataset are ignored.

#### Example 3

Because Stata has five numeric variable types—byte, int, long, float, and double—you may attempt to append datasets containing variables with the same name but of different numeric types; see [U] 12.2.2 Numeric storage types.

Let's describe the datasets in the example above:

	describe	using	https:/	1	www.stata-press	.com,	/data/	′r19/	odd	ł
--	----------	-------	---------	---	-----------------	-------	--------	-------	-----	---

Contains dat Observation Variable	a 1s: es:	5 2		First five odd numbers 9 Jan 2024 08:50
Variable name	Storage type	Display format	Value label	Variable label
number odd	float float	%9.0g %9.0g		Odd numbers
Sorted by: . describe u	sing https	://www.stat	a-press.co	m/data/r19/even
Contains dat Observation Variable	a ns: es:	3 2	1	6th through 8th even numbers 9 Jan 2024 08:43
Variable name	Storage type	Display format	Value label	Variable label
number even	byte float	%9.0g %9.0g		Even numbers
Sorted by: n . describe u	number nsing https	://www.stat	a-press.co	m/data/r19/oddeven
Contains dat Observation Variable	a ns: es:	8 3		First five odd numbers 9 Jan 2024 08:53
Variable name	Storage type	Display format	Value label	Variable label
number odd even	float float float	%9.0g %9.0g %9.0g		Odd numbers Even numbers

Sorted by:

The number variable was stored as a float in odd.dta but as a byte in even.dta. Because float is the more precise storage type, the resulting dataset, oddeven.dta, had number stored as a float. Had we instead appended odd.dta to even.dta, number would still have been stored as a float:

use https://www.stata-press.com/data/r19/even, clear 6th through 8th even numbers)								
. append us (variable <b>n</b>	append using https://www.stata-press.com/data/r19/odd variable number was byte, now float to accommodate using data's values)							
. describe Contains data from https://www.stata-press.com/data/r19/even.dta Observations: 8 6th through 8th even numbers Variables: 3 9 Jan 2024 08:43								
Variable name	Storage type	Display format	Value label	Variable label				
number even odd	float float float	%9.0g %9.0g %9.0g		Even numbers Odd numbers				

Sorted by:

Note: Dataset has changed since last saved.

#### Example 4

Suppose that we have a dataset in memory containing the variable educ, and we have previously given a label variable educ "Education Level" command so that the variable label associated with educ is "Education Level". We now append a dataset called newdata.dta, which also contains a variable named educ, except that its variable label is "Ed. Lev". After appending the two datasets, the educ variable is still labeled "Education Level". See [U] **12.6.2 Variable labels**.

#### Example 5

Assume that the values of the educ variable are labeled with a value label named educlbl. Further assume that in newdata.dta, the values of educ are also labeled by a value label named educlbl. Thus there is one definition of educlbl in memory and another (although perhaps equivalent) definition in newdata.dta. When you append the new data, you will see the following:

```
. append using newdata
label educlbl already defined
```

If one label in memory and another on disk have the same name, append warns you of the problem and sticks with the definition currently in memory, ignoring the definition in the disk file.

4

4

### Technical note

When you append two datasets that both contain definitions of the same value label, the codings may not be equivalent. That is why Stata warns you with a message like "label educlbl already defined". If you do not know that the two value labels are equivalent, you should convert the value-labeled variables into string variables, append the data, and then construct a new coding. decode and encode make this easy:

```
. use newdata, clear
```

- . decode educ, gen(edstr)
- . drop educ
- . save newdata, replace
- . use basedata
- . decode educ, gen(edstr)
- . drop educ
- . append using newdata
- . encode edstr, gen(educ)
- . drop edstr

#### See [D] encode.

You can specify the nolabel option to force append to ignore all the value-label definitions in the incoming file, whether or not there is a conflict. In practice, you will probably never want to do this.

#### Example 6

Suppose that we have several datasets containing the populations of counties in various states. We can use append to combine these datasets all at once and use the generate() option to create a variable identifying from which dataset each observation originally came.

. use https://www.stata-press.com/data/r19/capop

. list

	county	pop
1.	Los Angeles	9878554
2.	Orange	2997033
3.	Ventura	798364

. append using https://www.stata-press.com/data/r19/ilpop

- > https://www.stata-press.com/data/r19/txpop, generate(state)
- . label define statelab 0 "CA" 1 "IL" 2 "TX"
- . label values state statelab

. list

	county	pop	state
1.	Los Angeles	9878554	CA
2. 3. ⊿	Ventura	798364	CA
4. 5.	DeKalb	103729	IL IL
6.	Will	673586	IL
7. 8.	Brazos Johnson	152415 149797	TX TX
9.	Harris	4011475	TX

### Video example

How to append files into a single dataset

## Reference

Chatfield, M. D. 2015. precombine: A command to examine  $n \ge 2$  datasets before combining. Stata Journal 15: 607–626.

# Also see

- [D] cross Form every pairwise combination of two datasets
- [D] joinby Form all pairwise combinations within groups
- [D] merge Merge datasets
- [D] save Save Stata dataset
- [D] use Load Stata dataset
- [U] 23 Combining datasets

Stata, Stata Press, Mata, NetCourse, and NetCourseNow are registered trademarks of StataCorp LLC. Stata and Stata Press are registered trademarks with the World Intellectual Property Organization of the United Nations. StataNow is a trademark 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.

4