

Combining Datasets in Stata

Thomas Elliott

January 31, 2013

Often, you will find yourself with two or more datasets, or data files, that you wish to combine into one data file. Stata provides a couple ways to combine datasets.

1 Appending Data

Appending data means you have two files of the same data, just with different cases. For example, say you have time series data (in which each case is a year), and one file (`yearly1.dta`) contains 1900-1950, and another file (`yearly2.dta`) contains 1951-2000. To combine these two files in Stata, you use the `append` command. First, load one of the files into Stata, then append the second:

```
use yearly1.dta
```

```
append using yearly2.dta
```

If both datasets have the exact same variables, then Stata will simply add all the cases to the end of the current dataset. If `yearly1` had variables that `yearly2` did not have, then Stata will create those variables for the cases in `yearly2` and set them all to missing. Likewise, if `yearly2` had variables that `yearly1` did not have, then Stata would create those variables for `yearly1` and set them all to missing.

2 Merging Data

More often than needing to append data, you'll want to merge data. Say, for example, you have your time series data with years 1900-2000, containing variables on a variety of measures, named `yearly`. You also have a separate data file (`crimerate`) with crime rates for the years 1920-1995 and you want to merge them into your time series dataset. To do so, you need to use the `merge` command. The `merge` command designates a master dataset, the one loaded in memory when you run the command, and a using dataset, the one you specify in the `merge` command. To use the `merge` command, you first must decide the nature of your merge, which can be one of the following:

- **one to one:** (1:1) each observation in the master dataset will be matched with only one observation in the using dataset, and each observation in the using dataset will be matched with only one observation in the master dataset. Essentially, you won't need multiple copies of any observations to complete the merge. For example, if you have simple time-series data,

with one observation for each year, then merging additional data will be on a one-to-one basis - you don't have multiple copies of the same year.

For example, say you have two files containing data in the following tables.

year	births	deaths	counts
1950	23	12	2
1960	24	16	8
1970	34	13	5

year	jobs	unemployment	rate
1950	458	7.8	12.5
1960	563	8.5	45.3
1970	512	9.3	32.1

Merging this data will be on a one-to-one basis, one line in the first table matches up with one and only one line in the second table. We merge the two tables with the following command:

```
use yearly1.dta
```

```
merge 1:1 year using yearly2.dta
```

Which would result in the following dataset:

year	births	deaths	counts	jobs	unemployment	rate
1950	23	12	2	458	7.8	12.5
1960	24	16	8	563	8.5	45.3
1970	34	13	5	512	9.3	32.1

- **many to one:** (`m:1`) multiple observations in the master dataset will be matched to a single observation in the using dataset. Observations in the using dataset will match multiple observations in the master dataset, and will be copied for each one. Observations in the master dataset will match only one observation in the using dataset. An example of this is when your master dataset contains individual level data and your using dataset contains state level data. Multiple individuals will live in the same state, so each will need its own copy of the state-level data from the using dataset.

For example, say you have two files with the data in the following tables:

id	age	education	state
1	23	12	TX
2	43	16	TX
3	82	10	CA
4	24	16	CA
5	34	18	CA
6	38	15	NY

state	population	crime rate	percent urban
TX	26	508.2	75.4
CA	38	503.8	89.7
NY	19	401.8	82.7

Merging this data will be on a many to one basis. Many rows in the master data will match up with one row in the using data. The command for this would be:

```
use individuals.dta
```

```
merge m:1 state using states.dta
```

Which would result in the following dataset:

id	age	education	state	population	crime rate	percent urban
1	23	12	TX	26	508.2	75.4
2	43	16	TX	26	508.2	75.4
3	82	10	CA	38	503.8	89.7
4	24	16	CA	38	503.8	89.7
5	34	18	CA	38	503.8	89.7
6	38	15	NY	19	401.8	82.7

- **one to many:** (1:m) multiple observations in the using dataset will be matched to a single observation in the master dataset. Observations in the master dataset will match multiple observations in the using dataset, and will be copied for each one. Observations in the using dataset will match only one observation in the master dataset. An example of this is when your master dataset contains school-level data and your using dataset contains student-level data. Multiple students will match to the same school, so you need to make copies of the master data for each student.

For example, say you have two files containing the following data:

school	population	percent free lunch	avg SAT
A	2400	12	1200
B	1200	23	1500
C	600	8	1400

studentID	math score	verbal score	school
1	95	85	A
2	75	80	A
3	80	70	B
4	90	95	B
5	70	75	C
6	80	80	C

We need to merge this data on a one to many basis - one row from the master dataset will match many rows in the using dataset. The command for this merge is:

```
use schools.dta
```

```
merge m:1 school using students.dta
```

Which will result in the following dataset:

school	population	% free lunch	avg SAT	studentID	math score	verbal score
A	2400	12	1200	1	95	85
A	2400	12	1200	2	75	80
B	1200	23	1500	3	80	70
B	1200	23	1500	4	90	95
C	600	8	1400	5	70	75
C	600	8	1400	6	80	80

- **many to many:** (m:m) multiple observations in the master dataset will be matched to multiple observations in the using dataset. This is rarely used and involves fairly complicated data.

Note that the variable on which you are merging must appear with the same name in both datasets. It is not possible to merge on a variable that has different names in the two files.

Merging a dataset creates a new variable called `_merge` that takes on a value for each case depending on the results of the merge:

Table 1: `_merge` values

<code>_merge</code>	Definition
1	observation appeared in master only
2	observation appeared in using only
3	observation appeared in both
4	observation appeared in both, missing values updated
5	observation appeared in both, conflicting non missing values

This lets you diagnose the results of the merging, Be warned: if you are merging multiple datasets, you will get an error if `_merge` already exists in your dataset.

Say you have the following data from the first example above, slightly modified:

year	births	deaths	counts
1950	23	12	2
1960	24	16	8
1970	34	13	5

year	jobs	unemployment	rate
1950	458	7.8	12.5
1960	563	8.5	45.3
1970	512	9.3	32.1
1980	534	8.9	41.6

If you merge the data, you have an extra row in the using dataset that won't match. You can change how Stata handles these cases, but the default behavior is to create a new row in the dataset with missing values for the variables in the master dataset:

year	births	deaths	counts	jobs	unemployment	rate	<code>_merge</code>
1950	23	12	2	458	7.8	12.5	3
1960	24	16	8	563	8.5	45.3	3
1970	34	13	5	512	9.3	32.1	3
1980	.	.	.	534	8.9	41.6	2

Stata does the same thing for unmatched rows contained in the master dataset - the row will have missing values for the variables merged in from the using dataset. These situations are where the `_merge` variable comes in handy - it makes it easy to identify cases that were not matched in the merge.