

COMBINING DATA FILES¹

Josh Klugman and Min Gong
Department of Sociology
Indiana University
April 11, 2005

Introduction

There are two commands in Stata that allow you to combine data files, **merge** (appropriate for panel data/longitudinal data) and **append** (suitable for pooled cross-sectional data). Distinguish carefully between these two commands. Merging data files refers to adding new variables to existing cases/observations/respondents, which increases the dataset's width. Appending data files means adding new cases/observations/respondents to existing observations.

We use the National Survey of Families and Households (NSFH) and General Social Survey (GSS) to explain the differences between the two commands. The first wave of NSFH were collected by interviewing respondents ages 19 or over from 13,007 households in 1987-88. Those respondents of the first wave sample were reinterviewed in 1992-94.² In this case, the **merge** command is appropriate if you want to combine the datasets. The General Social Surveys have been conducted since 1972. In each survey year, different respondents were interviewed on some of the same questions. (Each survey year also includes unique questions.) So the **append** command is suitable if you want to select data from certain years and create a pooled cross-sectional dataset. Remember, we have the combined GSS data on the G: drive if you want to use the data for all the years.

Merging Data Files

Before you do any merging commands, you need to make sure that all of the datasets you are merging are sorted by the match variable, the variable with values that should be unique for every case. Stata uses the match variable to keep track of cases when it's merging. For example, in the NSFH data, the match variables are "mcaseid" in the first wave; "id," "id1," etc. in the second wave. This means you need to go through each dataset you will merge, create a universal matching variable that has the same name and values, do the sort command, and save the dataset. In the NSFH data, if you use "mcaseid" as the universal variable, you need to go through the other datasets and rename the variables "id," "id1," etc. as "mcaseid." You can tell whether or not a dataset has been sorted by doing a **describe** command—the last line will tell you how the data is sorted. We have found that if you issue even simple analysis or data management commands after you do the sort, even if the order of the cases remains the same, Stata will consider the data unsorted. So, before you save the datasets, the last thing you should do is the *sort*.

Stata distinguishes between the master dataset and the using dataset. The master dataset is the dataset that is open in Stata when you run the merge command; the using dataset(s) are the files you merge to the dataset that is in memory. Whichever dataset you want to use as the "master" depends on your analytical purpose.

¹ We create this document to facilitate users to combine datasets in Stata. If you are still unclear after reading it, please see the Stata reference (release eight) for more detailed instructions.

² The NSFH website has preliminary data for wave III.

The basic syntax of the merge command is:

```
. merge [match variable] using filename
```

You can only run this command when the master dataset is opened in Stata.

Important options for merge are (you specify these options after the merge command above; they're placed after a comma):

keep(*varlist*)—the default is that all of the variables in the using datasets are merged with the master data; if you don't want this, the keep option will let you choose which variables in the using data you want to merge.

update replace—if the master and using dataset have variables with the same name, Stata's default is to toss out the variables in the using dataset. If you type **update**, Stata will replace missing values in the master data with nonmissing values in the using data (if the variable(s) with the missing data are also found in the using data). If you type **update replace**, Stata will overwrite the variables in the master data with the variables from the using data.³

nokeep—If your using data have cases which are not in the master data, Stata's default is to add these cases to the master data. If you want to drop those extra cases in the using data, type **nokeep**, and Stata will merge only those cases that are in both the master and using data.

It is extraordinarily important that you make sure the “merge” command worked correctly—that the cases in the using data were matched with the correct cases in the master data. To do this, when you do a merge, Stata will create a _merge variable which will let you know where each case came from and how it was affected by the merge. The _merge variable has 5 values.

- 1—the case originated in the master data; not found in the using data
- 2—the case originated in the using data; not found in the master data
- 3—the case was found in both the master and using data
- 4—the case was found in both the master and using data; missing values in the master were replaced with nonmissing values in the using data (if update option is specified).
- 5—the case was found in both the master and using data; nonmissing values in the master data were replaced with nonmissing values in the using data (if update replace is specified).

You should tab the _merge variable after the merge. If you get values other than 3, you need to make sure you know why that happened.

If you are merging multiple using datasets to one master dataset, for each merge after the first one you will have to tell Stata to change the name of the _merge variable (otherwise Stata will

³ Exception: if a case has a nonmissing value in the master data and a missing value in the using data—the value from the master data will be retained.

stop and tell you that you already have a variable named `_merge` in the data). The option to do this is:

```
_merge(varname)
```

Where *varname* is the new name of the `_merge` variable.

`nolabel`—If you use this option, you are telling Stata not to copy the value-label definitions from the using data into the master data in memory.⁴

Appending Data Files

Before you do any appending commands, you need to make sure that all the data files that you want to combine contain the same variables. If the first dataset in memory includes more variables than the other datasets you want to append, Stata will assign a missing value, “.”, to those variables for the appended cases. If the dataset in memory includes fewer variables than the other datasets you want to append, Stata will assign a missing value, “.”, to those variables for the dataset in memory. If you want to combine multiple data sets, you need to check all the variables in all the datasets before issuing the “append” command.

The basic syntax of the append command is:

```
. append using filename
```

The first dataset is the dataset that is open in Stata before you run the append command; the other dataset(s) are the files you append to the dataset that is in memory. The merge command described above is to add the width to your data. However, the append command is to add the length to your data.

Important options for append are (you specify these options after the append command above; they’re placed after a comma):

`keep(varlist)`—the default is that all of the variables in the appended datasets are kept. If you don’t want all of the variables, the keep option will let you choose which variables you want to keep.

`nolabel`—If you use this option, you are telling Stata not to copy the value-label definitions from the second dataset into the first dataset in memory.

It’s very important that you make sure that the “append” command worked correctly—that the sample size of the accumulative data is the sum of all the data sets that you appended. To do this, you can type `tab year` which will let you verify whether the combined dataset were the sum of the original datasets.

You are ready to open the Stata file that you combined. Congratulations!!

⁴ The example of the “nolabel” option is upon request.