Append, Merge, and Collapse in Stata

This document will assist Stata users in learning when and how to use append, merge, collapse, and many of the options for each in Stata. Each of these commands offers a way for users to manipulate datasets. Append and merge are both way of combining datasets into larger datasets. Collapse can be helpful in many ways; for saving summary statistics of variables, changing the level of data, cleaning your data, etc.

Append

Append is used when you want to combine datasets that contain the same variables, but have different cases, thus, you are adding new rows to the dataset, but the number of columns should remain the same.

Here is an example of a basic append:

```
. use "/Users/tlg0309/Desktop/Soc561Programming/auto1"
(1978 Automobile Data)

. describe

Contains data from /Users/tlg0309/Desktop/Soc561Programming/au
> to1.dta
  obs:            51                          1978
                                                Automobile
                                                Data
 vars:            12                          29 Feb 2016 10:3
> 6
 size:         2,193                          (_dta has notes)
------------------------------------------------------------------------------
              storage   display    value
variable name   type    format     label      variable label
------------------------------------------------------------------------------
make            str18   %-18s                 Make and Model
price           int     %8.0gc                Price
mpg             int     %8.0g                 Mileage (mpg)
rep78           int     %8.0g                 Repair Record
                                              1978
headroom        float   %6.1f                 Headroom (in.)
trunk           int     %8.0g                 Trunk space (cu.
                                              ft.)
weight          int     %8.0gc                Weight (lbs.)
length          int     %8.0g                 Length (in.)
turn            int     %8.0g                 Turn Circle
                                              (ft.)
displacement    int     %8.0g                 Displacement
                                              (cu. in.)
gear_ratio      float   %6.2f                 Gear Ratio
foreign         byte    %8.0g       origin    Car type
------------------------------------------------------------------------------
Sorted by:  foreign
```

```
. append using "/Users/tlg0309/Desktop/Soc561Programming/auto2
> "
(label origin already defined)

. describe

Contains data from /Users/tlg0309/Desktop/Soc561Programming/au
> to1.dta
  obs:            74                          1978
                                                Automobile
                                                Data
 vars:            12                          29 Feb 2016 10:3
> 6
 size:         3,182                          (_dta has notes)
-------------------------------------------------------------------------------
              storage   display    value
variable name   type    format     label      variable label
-------------------------------------------------------------------------------
make            str18   %-18s                 Make and Model
price           int     %8.0gc                Price
mpg             int     %8.0g                 Mileage (mpg)
rep78           int     %8.0g                 Repair Record
                                              1978
headroom        float   %6.1f                 Headroom (in.)
trunk           int     %8.0g                 Trunk space (cu.
                                              ft.)
weight          int     %8.0gc                Weight (lbs.)
length          int     %8.0g                 Length (in.)
turn            int     %8.0g                 Turn Circle
                                              (ft.)
displacement    int     %8.0g                 Displacement
                                              (cu. in.)
gear_ratio      float   %6.2f                 Gear Ratio
foreign         byte    %8.0g       origin    Car type
-------------------------------------------------------------------------------
Sorted by:
     Note:  dataset has changed since last saved
```

Here you can see that the number of variables remains the same, while the number of observations increases from 51 to 74.

VARIABLE NAMES

Perhaps the most important thing to consider when appending data, is to make sure the variable names are consistent across datasets. If not, this will led to problems in the new appended dataset.

Here is an example showing how to check for mismatching variable names:

```
. use "/Users/tlg0309/Desktop/Soc561Programming/auto2"
(1978 Automobile Data)

. describe

Contains data from /Users/tlg0309/Desktop/Soc561Programming/auto2.dta
  obs:            23                          1978 Automobile Data
 vars:            12                          29 Feb 2016 10:37
 size:           989                          (_dta has notes)
-------------------------------------------------------------------------------
              storage   display    value
variable name   type    format     label      variable label
-------------------------------------------------------------------------------
make            str18   %-18s                 Make and Model
price           int     %8.0gc                Price
mpg             int     %8.0g                 Mileage (mpg)
rep78           int     %8.0g                 Repair Record 1978
headroom        float   %6.1f                 Headroom (in.)
trunk           int     %8.0g                 Trunk space (cu. ft.)
weight          int     %8.0gc                Weight (lbs.)
length          int     %8.0g                 Length (in.)
turn            int     %8.0g                 Turn Circle (ft.)
displacement    int     %8.0g                 Displacement (cu. in.)
gear_ratio      float   %6.2f                 Gear Ratio
foreign         byte    %8.0g       origin    Car type
-------------------------------------------------------------------------------
Sorted by:  foreign
```

In this first dataset, there are 23 observations, and 12 variables. Next I append the dataset with a dataset where the variable “price” is named “cost”.

```
. append using "/Users/tlg0309/Desktop/Soc561Programming/auto3"
(label origin already defined)

. describe

Contains data from /Users/tlg0309/Desktop/Soc561Programming/auto2.dta
  obs:            74                          1978 Automobile Data
 vars:            13                          29 Feb 2016 10:37
 size:         3,478                          (_dta has notes)
-------------------------------------------------------------------------------
              storage   display    value
variable name   type    format     label      variable label
-------------------------------------------------------------------------------
make            str18   %-18s                 Make and Model
price           int     %8.0gc                Price
mpg             int     %8.0g                 Mileage (mpg)
rep78           int     %8.0g                 Repair Record 1978
headroom        float   %6.1f                 Headroom (in.)
trunk           int     %8.0g                 Trunk space (cu. ft.)
weight          int     %8.0gc                Weight (lbs.)
length          int     %8.0g                 Length (in.)
turn            int     %8.0g                 Turn Circle (ft.)
displacement    int     %8.0g                 Displacement (cu. in.)
gear_ratio      float   %6.2f                 Gear Ratio
foreign         byte    %8.0g       origin    Car type
cost            float   %9.0g
-------------------------------------------------------------------------------
Sorted by:
     Note:  dataset has changed since last saved
```

Here we can see that the number of observations has increased from 23 to 74 as expected. However, the number of variables has also increased from 12 to 13. In the new appended data, variables "price" and "cost" are both describing the same data, but are included in separate variables.

Further, it is important that variables are describing the same data and are using the same measurement. For instance, if you have two variables in separate datasets both named "education," if one is measured in years of education, while the other is measured as a numerical categorical variable, Stata will not be able to recognize the difference between those two, which will lead to inaccurate data.

DATA TYPES

It is also important to consider the data type. If the data types for variables do not match one another, Stata will not allow you to append the datasets. For example, if the variable "price" is saved as a string variable in one dataset, but float in another, Stata will not allow you to append those two datasets with one another.

SORT

You do not need to sort the data before appending the data because the newly appended data will not be sorted by that variable. Because you are adding observations, no matching is involved.

SAVING

It is important to remember that these changes only take place in the working memory, thus in order to make sure you do not lose your newly appended dataset, you must save the dataset, preferably with a new name.

Merge

Merging is another way of combining datasets. Where the append command adds rows, or observations, merge adds columns, or variables. Thus, you would use merge when there are the same observations across datasets, with different variables, and you want to combine those datasets into one larger dataset.

ONE-TO-ONE MERGE

A 1-to-1 merge combines datasets that have identifiers for single observations in each dataset. In this type of merge, Stata combines the data based on those identifiers. In the next example, the variable "make" is unique to each observation, and present in each dataset.

Here is an example of a 1-to-1 merge:

```
. use "/Users/tlg0309/Desktop/Soc561Programming/autoA"
(1978 Automobile Data)

. describe

Contains data from /Users/tlg0309/Desktop/Soc561Programming/autoA.dta
  obs:            74                          1978 Automobile Data
 vars:             7                          29 Feb 2016 13:09
 size:         2,294                          (_dta has notes)
------------------------------------------------------------------------------
              storage   display    value
variable name   type    format     label      variable label
------------------------------------------------------------------------------
make            str18   %-18s                 Make and Model
weight          int     %8.0gc                Weight (lbs.)
length          int     %8.0g                 Length (in.)
turn            int     %8.0g                 Turn Circle (ft.)
displacement    int     %8.0g                 Displacement (cu. in.)
gear_ratio      float   %6.2f                 Gear Ratio
foreign         byte    %8.0g      origin     Car type
------------------------------------------------------------------------------
Sorted by:  foreign
```

Here I opened the master dataset, and then asked Stata to describe that data. You can see there are 74 observations and 7 variables. Next I merged using the merge 1:1 syntax, matching on the variable "make".

```
. merge 1:1 make using "/Users/tlg0309/Desktop/Soc561Programming/autoB"

    Result                           # of obs.
    -----------------------------------------
    not matched                             0
    matched                                74  (_merge==3)
    -----------------------------------------

. describe

Contains data from /Users/tlg0309/Desktop/Soc561Programming/autoA.dta
  obs:            74                          1978 Automobile Data
 vars:            13                          29 Feb 2016 13:09
 size:         3,256                          (_dta has notes)
-------------------------------------------------------------------------------
              storage   display    value
variable name   type    format     label      variable label
-------------------------------------------------------------------------------
make            str18   %-18s                 Make and Model
weight          int     %8.0gc                Weight (lbs.)
length          int     %8.0g                 Length (in.)
turn            int     %8.0g                 Turn Circle (ft.)
displacement    int     %8.0g                 Displacement (cu. in.)
gear_ratio      float   %6.2f                 Gear Ratio
foreign         byte    %8.0g       origin    Car type
price           int     %8.0gc                Price
mpg             int     %8.0g                 Mileage (mpg)
rep78           int     %8.0g                 Repair Record 1978
headroom        float   %6.1f                 Headroom (in.)
trunk           int     %8.0g                 Trunk space (cu. ft.)
_merge          byte    %23.0g      _merge
-------------------------------------------------------------------------------
Sorted by:  make
     Note:  dataset has changed since last saved
```

You can see that the merge was successful. There were 74 observations matched, and there are now 13 variables in the newly merged dataset.

Keep Option

You can also use the "keep(match)" option so that Stata only keeps the observations that match. In the example above, using this option wouldn't change the results of the merge because all of the observations matched. However, if there were non-matching observations, it would only keep the observations that matched. You can also specify the type of match you would want to keep. For instance, if you want to keep observations that were not matched from the master dataset, you could use the option "keep if _merge=1" or if you want to keep the observations that were not matched from the using dataset, you could use the option "keep if _merge=2".

MANY-TO-ONE MERGE

You would use a many-to-1 merge when you have a master dataset that has individual level data and using dataset that contains within-individual level data as well as a unique individual level identifier from the master data. The Stata merge pdf gives this example of adding regional level data to person-level data:

```
. merge m:1 region using filename
```

master + *using* = *merged result*

id	region	a
1	2	26
2	1	29
3	2	22
4	3	21
5	1	24
6	5	20

region	x
1	15
2	13
3	12
4	11

id	region	a	x	_merge
1	2	26	13	3
2	1	29	15	3
3	2	22	13	3
4	3	21	12	3
5	1	24	15	3
6	5	20	.	1
.	4	.	11	2

ONE-TO-MANY MERGE

The one-to-many merge is simply the reverse of a many-to-one merge. In this case the master dataset would have within-individual level data and the using data would contain individual level data that you want to add to the master dataset. The Stata merge pdf gives another example of adding person-level data, to regional level data:

```
. merge 1:m region using filename
```

master + *using* = *merged result*

region	x
1	15
2	13
3	12
4	11

id	region	a
1	2	26
2	1	29
3	2	22
4	3	21
5	1	24
6	5	20

region	x	id	a	_merge
1	15	2	29	3
1	15	5	24	3
2	13	1	26	3
2	13	3	22	3
3	12	4	21	3
4	11	.	.	1
5	.	6	20	2

MANY-TO-MANY

The Stata manual doesn't even cover this because it's such a bad idea. The observations are matched within equal values of the specified variables with the corresponding observations matching with one another. If there are not an equal number of observations then the last observation of the shorter dataset continues matching the rest of the observations of the longer dataset.

Collapse

The collapse command is helpful for saving summary statistics of your data, changing the level of analysis of your data, and for cleaning your data.

SUMMARY STATISTICS

Collapse is quick and useful way to obtain summary statistics about your data.

Using the basic syntax "collapse *varname"* collapses the entire data set into one variable that is equal to the mean of the specified variable. Here is an example:

```
. sysuse auto
(1978 Automobile Data)

. preserve

. describe

Contains data from /Applications/Stata/ado/base/a/auto.dta
  obs:            74                          1978 Automobile Data
 vars:            12                          13 Apr 2013 17:45
 size:         3,182                          (_dta has notes)
------------------------------------------------------------------------------
              storage   display    value
variable name   type    format     label      variable label
------------------------------------------------------------------------------
make            str18   %-18s                 Make and Model
price           int     %8.0gc                Price
mpg             int     %8.0g                 Mileage (mpg)
rep78           int     %8.0g                 Repair Record 1978
headroom        float   %6.1f                 Headroom (in.)
trunk           int     %8.0g                 Trunk space (cu. ft.)
weight          int     %8.0gc                Weight (lbs.)
length          int     %8.0g                 Length (in.)
turn            int     %8.0g                 Turn Circle (ft.)
displacement    int     %8.0g                 Displacement (cu. in.)
gear_ratio      float   %6.2f                 Gear Ratio
foreign         byte    %8.0g      origin     Car type
------------------------------------------------------------------------------
Sorted by:  foreign
```

```
. collapse mpg

. describe

Contains data
  obs:             1                          1978 Automobile Data
 vars:             1
 size:             4                          (_dta has notes)
-------------------------------------------------------------------------------
              storage   display    value
variable name   type    format     label      variable label
-------------------------------------------------------------------------------
mpg             float   %8.0g                 (mean) mpg
-------------------------------------------------------------------------------
Sorted by:
     Note:  dataset has changed since last saved

. tab mpg

 (mean) mpg |      Freq.     Percent        Cum.
------------+-----------------------------------
    21.2973 |          1      100.00      100.00
------------+-----------------------------------
      Total |          1      100.00

. restore
```

In this example I collapsed the auto data into one variable that equaled the mean of mpg. While this isn't particularly useful on it's own, you can run the collapse command on multiple variables and save it as a new dataset which can then be appended to the original, which can be useful for creating tables and graphs with the summary statistics of those variables.

Collapse can be used with options for a variety of summary statistics including mean, median, percentiles, minimum and maximum values, standard deviations, standard errors, etc.

LEVEL OF ANALYSIS

This command can also be useful for changing the level of analysis you are interested in. For example, if you have person-level data that are contained within family level data and you are interested in finding summary statistics at the family level, you can use the "by" option to do this.

Here is an example:

```
. collapse mpg, by(foreign)

. list
```

	foreign	mpg
1.	Domestic	19.8269
2.	Foreign	24.7727

This example shows the mean mpg for domestic cars as well as the mean mpg for foreign cars.