The Stata Journal (2009)
9, Number 1, pp. 166–168

Stata tip 73: append with care!

Christopher F. Baum
Department of Economics
Boston College
Chestnut Hill, MA
baum@bc.edu

The `append` command is a useful tool for data management. Most users are aware that they should be careful when appending datasets in which variable names differ; for instance, `PRICE` in one dataset with `price` in another will lead to both variables appearing in different columns of the combined dataset. But one perhaps lesser-known feature of `append` is worth noting. What if the *names* of the variables in the two datasets are the same, but their *data types* differ? If that is the case, then the order in which you combine the datasets may matter and can even lead to different retained contents in the combined dataset. This is particularly dangerous (as I recently learned!) when a variable is held as numeric in one dataset and string in another.

Let's illustrate this feature with `auto.dta`. You may know that the `foreign` variable is a 0/1 indicator variable (0 for domestic, 1 for foreign) with a value label. Let's create two datasets from `auto.dta`: the first with only domestic cars (`autodom.dta`) and the second with only foreign cars (`autofor.dta`). In the former dataset, we will leave the `foreign` variable alone. It is numeric and will be zero for all observations. In the second dataset, we create a string variable named `foreign`, containing `foreign` for each observation.

```
. sysuse auto
(1978 Automobile Data)
. drop if foreign
(22 observations deleted)
. save autodom
file autodom.dta saved
. sysuse auto
(1978 Automobile Data)
. drop if !foreign
(52 observations deleted)
. rename foreign nondom
. generate str foreign = "foreign" if nondom
. save autofor
file autofor.dta saved
```

Let's pretend that we are unaware that the same variable name, `foreign`, has been used to represent numeric content in one dataset and string content in the other, and let's use `append` to combine them. We `use` the domestic dataset and `append` the foreign dataset:

dm0044

```
. use autodom
(1978 Automobile Data)
. append using autofor
(note: foreign is str7 in using data but will be byte now)
(label origin already defined)
. describe foreign
              storage  display     value
variable name   type   format      label      variable label
-------------------------------------------------------------------------------
foreign         byte   %8.0g       origin     Car type
. codebook foreign
-------------------------------------------------------------------------------
foreign                                                                Car type
-------------------------------------------------------------------------------

                  type:  numeric (byte)
                 label:  origin

                 range:  [0,0]                        units:  1
         unique values:  1                        missing .:  22/74

            tabulation:  Freq.   Numeric  Label
                            52         0  Domestic
                            22         .
```

Notice that `append` produced the following message:

```
(note: foreign is str7 in using data but will be byte now)
```

This message indicates that the `foreign` variable will be numeric in the combined dataset. The contents of the string variable in the using dataset have been lost, as you can see from the `codebook` output. Twenty-two cases are now classified as missing.

But quite a different outcome is forthcoming if we merely reverse the order of combining the datasets. It usually would not matter in what order we combined two or more datasets. After all, we could always use `sort` to place them in any desired order. But if we first `use` the foreign dataset and `append` the domestic dataset, we receive the following results:

```
. use autofor, clear
(1978 Automobile Data)
. append using autodom
(note: foreign is byte in using data but will be str7 now)
(label origin already defined)
. describe foreign
              storage  display     value
variable name   type   format      label      variable label
-------------------------------------------------------------------------------
foreign         str7   %9s
```

```
. codebook foreign
------------------------------------------------------------------------------
foreign                                                            (unlabeled)
------------------------------------------------------------------------------

                  type:  string (str7)

         unique values:  1                        missing "":  52/74

            tabulation:  Freq.  Value
                            52  ""
                            22  "foreign"
```

Again we receive the fairly innocuous message:

```
(note: foreign is byte in using data but will be str7 now)
```

Unfortunately, this message may not call your attention to what has happened in the `append`. Because the data type of the first dataset rules, the string variable is unchanged, and the numeric values in the `using` dataset are discarded. As the codebook shows, the variable `foreign` is now missing for all domestic cars.

You may be used to the notion, with commands like `merge`, that the choice of master and using datasets matters. You may not be as well aware that `append`'s results may also be sensitive to the order in which files are appended. You should always take heed of the messages that `append` produces when data types are altered. If the `append` step changes only the data type of a numeric variable to allow for larger contents (for instance, `byte` to `long` or `float` to `double`) or extends the length of a string variable to allow for longer strings, no harm is done. But `append` does not highlight the instances, such as those we have displayed above, where combining string and numeric variables with the same name causes the total loss of one or the other's contents. In reality, that type of data type alteration deserves a warning message, or perhaps even an error. Until and unless such changes are made to this built-in Stata command, `append` with care.