The *Stata Journal* publishes reviewed papers together with shorter notes or comments, regular columns, book reviews, and other material of interest to Stata users. Examples of the types of papers include 1) expository papers that link the use of Stata commands or programs to associated principles, such as those that will serve as tutorials for users first encountering a new field of statistics or a major new technique; 2) papers that go “beyond the Stata manual” in explaining key features or uses of Stata that are of interest to intermediate or advanced users of Stata; 3) papers that discuss new commands or Stata programs of interest either to a wide spectrum of users (e.g., in data management or graphics) or to some large segment of Stata users (e.g., in survey statistics, survival analysis, panel analysis, or limited dependent variable modeling); 4) papers analyzing the statistical properties of new or existing estimators and tests in Stata; 5) papers that could be of interest or usefulness to researchers, especially in fields that are of practical importance but are not often included in texts or other journals, such as the use of Stata in managing datasets, especially large datasets, with advice from hard-won experience; and 6) papers of interest to those who teach, including Stata with topics such as extended examples of techniques and interpretation of results, simulations of statistical concepts, and overviews of subject areas.

For more information on the *Stata Journal*, including information for authors, see the web page

http://www.stata-journal.com

The *Stata Journal* is indexed and abstracted in the following:

- CompuMath Citation Index®
- RePEc: Research Papers in Economics
- Science Citation Index Expanded (also known as SciSearch®)

**Copyright Statement**: The *Stata Journal* and the contents of the supporting files (programs, datasets, and help files) are copyright © by StataCorp LP. The contents of the supporting files (programs, datasets, and help files) may be copied or reproduced by any means whatsoever, in whole or in part, as long as any copy or reproduction includes attribution to both (1) the author and (2) the *Stata Journal*.

The articles appearing in the *Stata Journal* may be copied or reproduced as printed copies, in whole or in part, as long as any copy or reproduction includes attribution to both (1) the author and (2) the *Stata Journal*.

Written permission must be obtained from StataCorp if you wish to make electronic copies of the insertions. This precludes placing electronic copies of the *Stata Journal*, in whole or in part, on publicly accessible web sites, file servers, or other locations where the copy may be accessed by anyone other than the subscriber.

Users of any of the software, ideas, data, or other materials published in the *Stata Journal* or the supporting files understand that such use is made without warranty of any kind, by either the *Stata Journal*, the author, or StataCorp. In particular, there is no warranty of fitness of purpose or merchantability, nor for special, incidental, or consequential damages such as loss of profits. The purpose of the *Stata Journal* is to promote free communication among Stata users.

The *Stata Journal* (ISSN 1536-867X) is a publication of Stata Press. Stata and Mata are registered trademarks of StataCorp LP.
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:
```

```
```
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:

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.
. codebook foreign

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>foreign</td>
<td>(unlabeled)</td>
</tr>
</tbody>
</table>

- 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.