AgEcon SEARCH
RESEARCH IN AGRICULTURAL & APPLIED ECONOMICS

The World's Largest Open Access Agricultural & Applied Economics Digital Library

This document is discoverable and free to researchers across the globe due to the work of AgEcon Search.

Help ensure our sustainability.

Give to AgEcon Search

AgEcon Search
http://ageconsearch.umn.edu
aesearch@umn.edu

*Papers downloaded from **AgEcon Search** may be used for non-commercial purposes and personal study only. No other use, including posting to another Internet site, is permitted without permission from the copyright owner (not AgEcon Search), or as allowed under the provisions of Fair Use, U.S. Copyright Act, Title 17 U.S.C.*

No endorsement of AgEcon Search or its fundraising activities by the author(s) of the following work or their employer(s) is intended or implied.

The Stata Journal

Editor
H. Joseph Newton
Department of Statistics
Texas A & M University
College Station, Texas 77843
979-845-8817; FAX 979-845-6077
jnewton@stata-journal.com

Editor
Nicholas J. Cox
Department of Geography
Durham University
South Road
Durham City DH1 3LE UK
n.j.cox@stata-journal.com

Associate Editors

Christopher F. Baum, Boston College
Rino Bellocco, Karolinska Institutet, Sweden and Univ. degli Studi di Milano-Bicocca, Italy
A. Colin Cameron, University of California–Davis
David Clayton, Cambridge Inst. for Medical Research
Mario A. Cleves, Univ. of Arkansas for Medical Sciences
William D. Dupont, Vanderbilt University
Charles Franklin, University of Wisconsin–Madison
Allan Gregory, Queen's University
James Hardin, University of South Carolina
Ben Jann, ETH Zürich, Switzerland
Stephen Jenkins, University of Essex
Ulrich Kohler, WZB, Berlin
Jens Lauritsen, Odense University Hospital
Stanley Lemeshow, Ohio State University
J. Scott Long, Indiana University
Thomas Lumley, University of Washington–Seattle
Roger Newson, Imperial College, London
Marcello Pagano, Harvard School of Public Health
Sophia Rabe-Hesketh, University of California–Berkeley
J. Patrick Royston, MRC Clinical Trials Unit, London
Philip Ryan, University of Adelaide
Mark E. Schaffer, Heriot-Watt University, Edinburgh
Jeroen Weesie, Utrecht University
Nicholas J. G. Winter, University of Virginia
Jeffrey Wooldridge, Michigan State University

Stata Press Production Manager Lisa Gilmore
Stata Press Copy Editor Deirdre Patterson

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, fileservers, 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*, electronic version (ISSN 1536-8734) is a publication of Stata Press. Stata and Mata are registered trademarks of StataCorp LP.

The Stata Journal (2007)
7, Number 4, pp. 582–583

Stata tip 52: Generating composite categorical variables

Nicholas J. Cox
Department of Geography
Durham University
Durham City, UK
n.j.cox@durham.ac.uk

If you have two or more categorical variables, you may want to create one composite categorical variable that can take on all the possible joint values. The canonical example for Stata users is given by cross-combinations of `foreign` and `rep78` in the `auto` data. Setting aside missings, `foreign` takes on values of 0 and 1, and `rep78` takes on values of 1, 2, 3, 4, and 5. Hence there are ten possible joint values, which could be 0 and 1, 0 and 2, and so forth. As it happens, only eight occur in the data. If we add the value labels attached to `foreign`, we have Domestic 1, Domestic 2, and so forth.

Writing the values like that raises the question of whether these cross-combinations will be better expressed as string variables or as numeric variables with value labels. On the whole, an integer-valued numeric variable with value labels defined and attached is the best arrangement for any categorical variable, but a string variable may also be convenient, especially if you are producing a kind of composite identifier.

A method often seen is to produce string variables with `tostring` (see [D] **destring**), for example,

```
. tostring foreign rep78, generate(Foreign Rep78)
. gen both = Foreign + Rep78
```

Naturally, there are endless minor variations on this method. A small but useful improvement is to insert a space or other punctuation:

```
. gen both = Foreign + " " + Rep78
```

However, this method is not especially good. `tostring` is really for correcting mistakes, whether attributable to human fault or to some software used before you entered Stata: some variable that should be string is in fact numeric. You need to correct that mistake. `tostring` is a safe way of doing that.

That intended purpose does not stop `tostring` being useful for things for which it was not intended, but there are two specific disadvantages to this method:

1. This method needs two lines, and you can do it in one. That is a little deal.
2. This method could lose information, especially for variables with value labels or with noninteger values. That is, potentially, a big deal.

The second point may suggest using `decode` instead, but my suggestions differ. A better method is to use `egen, group()`. See [D] **egen**.

© 2007 StataCorp LP

```
. egen both = group(foreign rep78), label
```

This command produces a new numeric variable, with integer values 1 and above, and value labels defined and attached. Particularly, note the `label` option, which is frequently overlooked.

This method has several advantages:

1. One line.
2. No loss of information. Observations that are identical on the arguments are identical on the results. Value labels are used, not ignored. Distinct noninteger values will also remain distinct.
3. The label is useful—indeed essential—for tables and graphs to make sense.
4. Efficient storage.
5. Extends readily to three or more variables.

Another fairly good method is to use `egen, concat()`.

```
. egen both = concat(foreign rep78), decode p(" ")
```

This command creates a string variable, so it is less efficient for data storage and is less versatile for graphics or modeling. Compared with `tostring`, the advantages are

1. One line.
2. You can mix numeric and string arguments. `concat()` will calculate what is needed.
3. You can use the `decode` option to use value labels on the fly.
4. You can specify punctuation as separator, here a blank.
5. Extends to three or more variables.