

# 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.

# Stata tip 80: Constructing a group variable with specified group sizes 

Martin Weiss
Department of Economics
Tübingen University
Tübingen, Germany
martin.weiss@uni-tuebingen.de
Quite often, Stata users wish to construct a variable denoting group membership with different group sizes. This could be part of a simulation study in which a discrete variable with a certain distribution is required. Two cases can be distinguished: one in which the desired group sizes should be hit exactly and one in which group size should vary randomly around the desired proportion. In either case, group membership is to be determined randomly. This problem differs from grouping on the basis of one or more existing categories, which usually is best accomplished through egen, group() (see [D] egen and Cox [2007]).

The first goal can be achieved by obtaining uniformly distributed random numbers with Stata's runiform() function (see Buis [2007] for an expanded discussion of the usefulness of uniform random numbers) and sorting on these numbers. Because this uniform random variable is of no interest as such, we construct it as a temporary variable. The groups are subsequently determined in one fell swoop by conditioning on the position of an observation after the random sorting. The way to achieve this is to use an expression, as described in [U] 13 Functions and expressions, featuring the system variable $\_$n (see [U] 13.4 System variables (_variables)), which denotes the running number of the observation. Say that you want groups with proportions $50 \%, 40 \%$, and $10 \%$ in a sample of 10,000 : whenever $\_\mathrm{n}$ is smaller than or equal to 5,000 , the expression inrange ( $n, 1,5000$ ) evaluates to 1 and the other two evaluate to zero. Thus the entire sum evaluates to 1 .

```
. clear
. set obs 10000
obs was 0, now 10000
. set seed 12345
. tempvar aux
. generate byte `aux`=runiform()
. sort 'aux*
. generate byte group = inrange(_n,1,5000)*1+
> linrange(_n,5001,9000)*2+
. tabulate group
\begin{tabular}{r|rrr} 
• tabulate group \\
group & Freq. & Percent & Cum. \\
\hline 1 & 5,000 & 50.00 & 50.00 \\
2 & 4,000 & 40.00 & 90.00 \\
3 & 1,000 & 10.00 & 100.00 \\
\hline Total & 10,000 & 100.00 &
\end{tabular}
```

The set seed command is used to make the draws from runiform(), and thus the division into groups, reproducible (for further information, see $[R]$ set seed). This practice is also maintained in the examples below.

A feasible way to achieve the second goal is the use of nested cond() functions, as introduced in Kantor and Cox (2005):

```
. clear
. set obs 10000
obs was 0, now 10000
. set seed 12345
. generate byte group = cond(runiform()<0.5, 1, cond(runiform()<0.8, 2, 3))
. tabulate group
\begin{tabular}{r|rrr} 
group & Freq. & Percent & Cum. \\
\hline 1 & 4,947 & 49.47 & 49.47 \\
2 & 4,048 & 40.48 & 89.95 \\
3 & 1,005 & 10.05 & 100.00 \\
\hline Total & 10,000 & 100.00 &
\end{tabular}
```

The outer cond () function takes care of the $50 \%$ group, and the inner cond() function splits the remaining $50 \%$ in the ratio 4 to 1 so that the end result coincides with the user's intention. The two-separate-draws from runiform() are not related in any way.

This process can be shortened further with the help of the lesser-known irecode() function (see [D] functions for the full array of functions).

```
. clear
. set obs 10000
obs was 0, now 10000
. set seed 12345
. generate byte group = irecode(runiform(), 0, 0.5, 0.9, 1)
. tabulate group
\begin{tabular}{r|rrr} 
group & Freq. & Percent & Cum. \\
\hline 1 & 4,957 & 49.57 & 49.57 \\
2 & 4,049 & 40.49 & 90.06 \\
3 & 994 & 9.94 & 100.00 \\
\hline Total & 10,000 & 100.00 &
\end{tabular}
```

The irecode() function takes $n+1$ arguments. The first argument is evaluated against the remaining $n$ arguments. If it is smaller than the second, " 0 " is returned; if it is larger than the second but smaller than the third, " 1 " is returned; and so on. Because the draws from runiform() must lie between 0 and 1, any draw will be larger than the second argument, and thus " 0 " will never be returned. Approximately $50 \%$ of the time, the draw will lie between the second and third argument, and the results will be " 1 ", and so on. The desired proportions for the groups must be translated into differences between the arguments of the irecode() function: to get the $40 \%$ group, for instance, the third and fourth argument must be $0.9-0.5=0.4$ units apart.

As a final thought, consider the slight difference between the group sizes in the second and third example - although we did set seed to " 12345 " in both cases. As mentioned above, in the second example, two draws from the uniform distribution are conducted, one for each cond () function. The irecode() function in the third example, however, is fed one draw from runiform(). The function compares this one draw with the cutpoints provided by the user and assigns group membership accordingly.

## References

Buis, M. L. 2007. Stata tip 48: Discrete uses for uniform(). Stata Journal 7: 434-435.
Cox, N. J. 2007. Stata tip 52: Generating composite categorical variables. Stata Journal 7: 582-583.

Kantor, D., and N. J. Cox. 2005. Depending on conditions: A tutorial on the cond() function. Stata Journal 5: 413-420.

