The Stata Journal

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.

The *Stata Journal* is indexed and abstracted by *CompuMath Citation Index*, *Current Contents/Social and Behavioral Sciences*, *RePEc: Research Papers in Economics*, *Science Citation Index Expanded* (also known as *SciSearch*), *Scopus*, and *Social Sciences Citation Index*.

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

http://www.stata-journal.com

Subscriptions are available from StataCorp, 4905 Lakeway Drive, College Station, Texas 77845, telephone 979-696-4600 or 800-STATA-PC, fax 979-696-4601, or online at

http://www.stata.com/bookstore/sj.html

Subscription rates listed below include both a printed and an electronic copy unless otherwise mentioned.

U.S. and Canada		Elsewhere	
Printed & electronic		**Printed & electronic**	
1-year subscription	$115	1-year subscription	$145
2-year subscription	$210	2-year subscription	$270
3-year subscription	$285	3-year subscription	$375
1-year student subscription	$ 85	1-year student subscription	$115
1-year institutional subscription	$345	1-year institutional subscription	$375
2-year institutional subscription	$625	2-year institutional subscription	$685
3-year institutional subscription	$875	3-year institutional subscription	$965
Electronic only		**Electronic only**	
1-year subscription	$ 85	1-year subscription	$ 85
2-year subscription	$155	2-year subscription	$155
3-year subscription	$215	3-year subscription	$215
1-year student subscription	$ 55	1-year student subscription	$ 55

Back issues of the *Stata Journal* may be ordered online at

http://www.stata.com/bookstore/sjj.html

Individual articles three or more years old may be accessed online without charge. More recent articles may be ordered online.

http://www.stata-journal.com/archives.html

The *Stata Journal* is published quarterly by the Stata Press, College Station, Texas, USA.

Address changes should be sent to the *Stata Journal*, StataCorp, 4905 Lakeway Drive, College Station, TX 77845, USA, or emailed to sj@stata.com.

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

The Stata Journal (2015)
15, Number 1, pp. 301–308

Generating nonnegatively correlated binary random variates

Minxing Chen
MD Anderson Cancer Center
Houston, TX
mchen3@mdanderson.org

Abstract. In this article, I discuss the methods for generating nonnegatively correlated binary random variates. I provide a new command, `rbinary`, with examples showing how the command can be used.

Keywords: st0382, rbinary, correlated binary random data, Monte Carlo simulation, drawnorm, multivariate normal distribution

1 Introduction

Correlated binary data are frequently encountered in clinical trials (for example, with or without certain disease symptoms) with repeated measures, system reliability analysis (for example, a machine fails or not), or the segmentation of marketing data (in which only two values, yes and no or 1 and 0, are used by customers to respond to questionnaires). Simulating these types of data has many important applications. For example, as suggested by Park, Park, and Shin (1996), simulated binary random variates with specified mean structures and correlation structures can be used to evaluate the small- or finite-sample properties of estimators in the generalized estimating equation; they can also be used in the analysis of system reliability in which the system comprises dependent components. When analytical evaluation of system performance is difficult, one needs to generate correlated binary variables for a Monte Carlo study. Correlated binary data are also found in the segmentation of marketing data, in which customer responses to questionnaires are usually binary (such as yes or no). Artificial data with known mean and correlation structure can be constructed to mimic situations from the real world (Dolnicar et al. 1998) and thus provide a valuable tool for the analysis of segmentation data (Leisch, Weingessel, and Hornik 1998). Finally, simulated correlated binary distribution is also useful for power analysis, such as calculating the power to detect the difference between pretreatment and posttreatment or treatment versus the control group in binary clinical data.

Unfortunately, there is not a standard procedure in the popular statistical software, including Stata, that can be used to directly simulate these types of data. In this article, I will present a straightforward and computationally fast method to achieve this goal. In section 2, after discussing different approaches for generating correlated binary variables, I focus on an appealing algorithm proposed by Emrich and Piedmonte (1991) and demonstrate how to generate desired correlated binary data step by step. In section 3, I provide detailed syntax for the new command `rbinary` and provide some examples of its use.

2 Multivariate binary distribution

Stata has a built-in command, `drawnorm`, that can easily be used to draw a sample from a multivariate normal distribution with desired means and covariance matrix. Unfortunately, to my knowledge, no such built-in command or user-written command is available to do the same for binary distribution.

Several approaches of using an algorithm to generate binary random variates with desired marginal probabilities and correlation have been discussed in earlier literature. For example, the Bahadur model (Bahadur 1961) uses marginal probabilities, pairwise correlations, and higher-order moments to parameterize the multinomial distribution. For this model, we let the binary variable Y_{ij} indicate outcome j for subject i. We let $Z_{ij} = Y_{ij} - \mu_{ij}/\sqrt{\mu_{ij}(1-\mu_{ij})}$ and $z_{ij} = y_{ij} - \mu_{ij}/\sqrt{\mu_{ij}(1-\mu_{ij})}$, where y_{ij} is the observed value of Y_{ij}. Further, we let parameters r_{ijk} denote Pearson correlation coefficients, where $r_{jk} = \text{cor}(Y_j, Y_k) = E(Z_j Z_k)$, and $r_{jkl} = \text{cor}(Y_j, Y_k, Y_l) = E(Z_j Z_k Z_l), \ldots, r_{(1,\ldots,n)} = E(Z_1 Z_2 \ldots Z_n)$. Then the probability mass function, $\Pr(Y = y)$, is the product of the marginal probability $f_1(y)$ and the correction term $f_2(y)$, where $f_1(y) = \sum_{j=1}^{n} p_j^{y_j}(1-p_j)^{(1-y_j)}$ and $f_2(y) = 1 + \sum_{j<k} r_{jk} \times z_j \times z_k + \sum_{j<k<l} r_{jkl} \times z_j \times z_k \times z_l + \cdots + r_{1,\ldots,n} \times z_1 \times z_2 \ldots z_n$. However, this approach has drawbacks; for example, the correlations are constrained by the marginal probabilities, and it is computationally difficult to handle for higher-dimensional correlated binary random variates (Park, Park, and Shin 1996).

A more appealing algorithm proposed by Park, Park, and Shin (1996) is the approach that any Poisson random variable can be expressed as the sum of several other independent Poisson random variables, and if two Poisson random variables share a Poisson distribution component, they are nonnegatively correlated. For example, let $X_{(a)}$ denote a Poisson random variable with nonnegative mean a and assume Xs are mutually independent if they have different subscripts. Now, consider $k = 2$, and let $Y_1 = X_{1(a_{11}-a_{12})} + X_{3(a_{12})}$ and $Y_2 = X_{2(a_{22}-a_{12})} + X_{3(a_{12})}$. Y_1 and Y_2 will then follow Poisson distribution with means a_{11} and a_{22} and are also correlated because they share a common component, $X_{3(a_{12})}$. Further, if we set $z_i = I_0(Y_i)$, where $I_A(y) = 1$ if $y \in A$ and $I_A(y) = 0$ if $y \notin A$, then z_1 and z_2 will be binary random variates converted from the Poisson variables, which are also correlated. This approach is most suitable for generating two-dimensional binary random variates, but it is difficult to generate any correlated binary data with more than two dimensions.

In this article, I adopt an algorithm proposed by Emrich and Piedmonte (1991) that is both computationally convenient and easy to understand. This approach first generates a random multivariate normal distribution with the mean vector $\boldsymbol{\mu}$ and variance–covariance structure $\boldsymbol{\Sigma}$, which can be transformed into binary values by setting

$$Z_i = \begin{cases} = 1, & \text{if} \quad x_i > 0 \\ = 0, & \text{if} \quad x_i \leq 0 \end{cases}$$

This transformed multivariate binary distribution will have the mean vector $\mathbf{p}$ and correlation structure $\boldsymbol{\Sigma}_{\mathbf{b}}$ that we need.

For example, we could use this algorithm to generate random multivariate binary data with mean vector $\mathbf{p}$ and correlation matrix $\mathbf{\Sigma_b}$ defined as

$$\mathbf{p} = \begin{pmatrix} 0.05 \\ 0.10 \\ 0.15 \\ 0.20 \end{pmatrix}, \quad \mathbf{\Sigma_b} = \begin{pmatrix} 1 & 0.3 & 0.2 & 0.1 \\ 0.3 & 1 & 0.3 & 0.2 \\ 0.2 & 0.3 & 1 & 0.3 \\ 0.1 & 0.2 & 0.3 & 1 \end{pmatrix}$$

I demonstrate how to perform this in the following steps.

2.1 Step 1: Determine mean vector μ for multivariate normal distribution

The mean vector $\boldsymbol{\mu}$ is simply the inverse cumulative standard normal distribution of $\mathbf{p}$. For example, with the above binary distribution mean vector, the normal distribution mean vector $\boldsymbol{\mu}$ would be

$$\boldsymbol{\mu} = \begin{pmatrix} \text{invnormal}(0.05) = -1.645 \\ \text{invnormal}(0.10) = -1.282 \\ \text{invnormal}(0.15) = -1.036 \\ \text{invnormal}(0.20) = -0.842 \end{pmatrix}$$

2.2 Step 2: Determine correlation matrix Σ for normal distribution

It is important that we obtain the variance–covariance $\mathbf{\Sigma}$ of the normal distribution. With the Stata built-in command `drawnorm`, we can easily generate a random multivariate normal distribution with a specified mean vector and covariance matrix. However, we still need to find the connection between the covariance matrix $\mathbf{\Sigma_b}$ of the binary distribution and the covariance matrix $\mathbf{\Sigma}$ of the normal distribution.

By definition, the correlation coefficient r_{AB} of any two binary random variables A and B can be written as $r_{AB} = (p_{AB} - p_A p_B)/\sqrt{p_A q_A p_B q_B}$, where p_{AB} is the joint probability $P(A = 1, B = 1)$; p_A, p_B is the marginal probability of binary variable A and B; and q_A is equal to $(1 - p_A)$ and q_B equal to $(1 - p_B)$. The equation can be written as $p_{AB} = r_{AB}\sqrt{p_A q_A p_B q_B} + p_A p_B$.

If A and B are converted from two normal random variables X and Y, as described above, p_{AB} can then be related to the normal distribution by

$$p_{AB} = IP(X > 0, Y > 0) = IP\left(\overline{X} > -\mu_X, \overline{Y} > -\mu_Y\right) = L(-\mu_X, -\mu_Y, \rho) \longrightarrow$$
$$L(h, k, \rho) = IP\left(\overline{X} > h, \overline{Y} > k\right) = \int \int_{hk}^{\infty} \phi(x, y, \rho) dy dx$$

where ρ is the correlation coefficient between X and Y, and

$$\phi(x, y, \rho) = \frac{1}{2\pi(1 - \rho^2)} \exp\left\{-\frac{x^2 - 2\rho xy + y^2}{2(1 - \rho^2)}\right\}$$

With the desired correlation coefficient r_{AB} and the marginal probability of variable A and B, we can now obtain the joint probability p_{ij} for any pair of binary variables. With known p_{ij} and the equation $L(h,k,\rho) = IP(\overline{X} > h, \overline{Y} > k) = \int \int_{hk}^{\infty} \phi(x,y,\rho)dydx$, we can use numerical integration or Monte Carlo simulation to obtain the covariance matrix $\mathbf{\Sigma}$ of the normal distribution needed for $\mathbf{\Sigma_b}$. This results in

$$\mathbf{\Sigma} = \begin{pmatrix} 1 & \rho_{12} & \rho_{13} & \rho_{14} \\ \rho_{21} & 1 & \rho_{23} & \rho_{24} \\ \rho_{31} & \rho_{32} & 1 & \rho_{34} \\ \rho_{41} & \rho_{42} & \rho_{43} & 1 \end{pmatrix} = \begin{pmatrix} 1 & 0.6104 & 0.4599 & 0.2589 \\ 0.6104 & 1 & 0.5598 & 0.4014 \\ 0.4599 & 0.5598 & 1 & 0.5246 \\ 0.2589 & 0.4014 & 0.5246 & 1 \end{pmatrix}$$

To save time, the command `rbinary` uses presimulated joint binary probabilities that were previously produced by Friedrich Leisch (Leisch, Weingessel, and Hornik 1998) (for program R) via Monte Carlo simulation (within R function `simul.commonprob`).

2.3 Step 3: Draw a sample from the multivariate normal distribution

With the mean vector $\boldsymbol{\mu}$ and correlation matrix $\mathbf{\Sigma}$ that we determined in step 1 and step 2, we can use the Stata built-in command `drawnorm` to draw a sample, X, from the multivariate normal distribution.

2.4 Step 4: Transform normal distribution to binary distribution

We can easily transform the multivariate normally distributed sample drawn in step 3 to binary values by setting

$$Z_i = \begin{cases} = 1, & \text{if} \quad x_i > 0 \\ = 0, & \text{if} \quad x_i \leq 0 \end{cases}$$

This transformed multivariate binary Z should have the approximate mean vector $\boldsymbol{p}$ and the correlation structure $\mathbf{\Sigma_b}$ needed.

3 The rbinary command

3.1 Syntax

`rbinary` *newvarlist*`, means(`*vector*`) corr(`*matrix* | *vector*`) n(#)` [`seed(#)`]

3.2 Options

`means(`*vector*`)` specifies the mean of each variable, up to six variables. `means()` is required.

`corr(`*matrix*| *vector*`)` specifies the correlation matrix. `corr()` is required.

`n(#)` specifies the number of observations to be generated. `n()` is required.

`seed(#)` specifies the initial value of the random-number seed used by the command `drawnorm`.

3.3 Stored results

`rbinary` stores the following in `r()`:

Matrices

`r(mean)`	means vector that was used in `drawnorm`
`r(sigma)`	correlation matrix that was used in `drawnorm`

3.4 Examples

Here we generate a sample of 2,000 observations from a bivariate binary distribution with desired marginal probabilities at 0.05 and 0.10 and correlation coefficient at 0.3.

```
. rbinary x y, means(0.05,0.10) corr(1,0.3\0.3,1) n(2000) seed(12345)
. correlate
(obs=2000)
             |        x        y
-------------+------------------
           x |   1.0000
           y |   0.3227   1.0000
. summarize
    Variable |       Obs        Mean    Std. Dev.       Min        Max
-------------+--------------------------------------------------------
           x |      2000        .049     .215922         0          1
           y |      2000        .105    .3066301         0          1
```

Here we generate 4-dimensional binary data with 2,000 observations, with the following desired mean vector and correlation structure:

$$\boldsymbol{p} = \begin{pmatrix} 0.05 \\ 0.10 \\ 0.15 \\ 0.20 \end{pmatrix}, \quad \boldsymbol{\Sigma_b} = \begin{pmatrix} 1 & 0.3 & 0.2 & 0.1 \\ 0.3 & 1 & 0.3 & 0.2 \\ 0.2 & 0.3 & 1 & 0.3 \\ 0.1 & 0.2 & 0.3 & 1 \end{pmatrix}$$

```
. rbinary x1 x2 x3 x4, means(.05,.10,.15,.20)
> corr(1,.3,.2,.1\.3,1,.3,.2\.2,.3,1,.3\.1,.2,.3,1) n(2000) seed(12345)
. correlate
(obs=2000)
             |       x1       x2       x3       x4
-------------+------------------------------------
          x1 |   1.0000
          x2 |   0.3227   1.0000
          x3 |   0.1918   0.2980   1.0000
          x4 |   0.1104   0.2349   0.3361   1.0000

. summarize
    Variable |       Obs        Mean    Std. Dev.       Min        Max
-------------+--------------------------------------------------------
          x1 |      2000        .049     .215922         0          1
          x2 |      2000        .105    .3066301         0          1
          x3 |      2000        .143    .3501604         0          1
          x4 |      2000        .195    .3962998         0          1
```

In the second example, we get empirical probabilities and correlation of

$$\widehat{\boldsymbol{p}} = \begin{pmatrix} 0.049 \\ 0.105 \\ 0.143 \\ 0.195 \end{pmatrix}, \quad \widehat{\boldsymbol{\Sigma}_{\boldsymbol{b}}} = \begin{pmatrix} 1 & 0.3227 & 0.1918 & 0.1104 \\ 0.3227 & 1 & 0.2980 & 0.2349 \\ 0.1918 & 0.2980 & 1 & 0.3361 \\ 0.1104 & 0.2349 & 0.3361 & 1 \end{pmatrix}$$

3.5 Application

Suppose the smoking abstinence rate difference between two treatment groups is constant at each time point, with an increase in abstinence of 5% across adjacent time points moving from the 6- to 12- to 18-month follow-ups. Researchers want to calculate the power to detect a 10% difference on abstinence rate with a specified within-subject correlation structure. To achieve this goal, researchers should do the following: 1) use `rbinary` to simulate random correlated binary data for each group with specified sample size, mean, and correlation structure; 2) combine the groups' data; 3) run the command `xtgee` on the randomly generated combined data; 4) obtain the p-value for the overall contrast of groups; 5) perform steps 1–4 N times and save the p-values in a file with N observations. The estimated power for an alpha 0.05 test is simply the proportion of observations (out of N) for which the p-value is less than 0.05.

```
quietly {
  tempname pow
  tempfile result data
  postfile `pow´ sample p using `result´, replace
    *try different sample size per group below until desired level power reached
  local nsize=90
    *specify alpha level here, e.g. 0.05
  local alpha=0.05
    *specify # bootstrap replications
  local reps=500
    *set seed to replicate result
  local seed=10000
  local N=1
  while `N´<=`reps´ {
    clear
    *specify mean vector in means(), and within-subject correlation matrix ///
         in corr()
    rbinary t1 t2 t3, means(0.05,0.10,0.15) corr(1,.2,.1\.2,1,.2\.1,.2,1) ///
         n(`nsize´) seed(`seed´)
    generate group=1
    save `data´, replace

    rbinary t1 t2 t3, means(0.15,0.20,0.25) corr(1,.2,.1\.2,1,.2\.1,.2,1) ///
         n(`nsize´) seed(`seed´)
    generate group=2
    append using `data´
    generate id=_n

    reshape long t, i(id) j(time)
    xtgee t i.time i.group, i(id) t(time) family(binomial) link(logit) ///
         corr(unstructured) vce(robust) nolog
    contrast group, overall
    matrix p=r(p)
    scalar p=p[1,1]

    post `pow´ (`N´) (scalar(p))
    local N=`N´+1
    local seed=`seed´+1
  }
  postclose `pow´
  use `result´, clear
  count if p<`alpha´
  local num=r(N)
  noisily display "Power=" round(`num´/`reps´, 0.001)
}
```

To determine the sample size needed to achieve a desired level of power, we can adjust the values of `n()` until we have the desired power. For example, with the above mean and correlation structure and with a sample size of 90 per group, the above syntax yields power = 0.96. This indicates that we should be able to reject the null hypothesis (H_0: there is no smoking abstinence rate difference between the groups) 96% of the time when it is false. Further, through trial and error, we can determine that a sample size of 65 yields power = 0.804, which may be adequate for the researchers' purposes and provide a more efficient use of limited resources.

4 Acknowledgments

I gratefully acknowledge Bryan Fellman for his help with preparing the LaTeX file. I also thank Beibei Guo and Israel Christie for their help with the R program.

5 References

Bahadur, R. R. 1961. A representation of the joint distribution of responses to n dichotomus items. In *Studies in Item Analysis and Prediction*, ed. H. Solomon, 158–168. Stanford, CA: Stanford University Press.

Dolnicar, S., F. Leisch, A. Weingessel, C. Buchta, and E. Dimitriadou. 1998. A comparison of several cluster algorithms on artificial binary data scenarios from travel market segementation. Working Paper No. 7, SFB Adaptive Information Systems and Modelling in Economics and Management Science, Vienna University of Economics and Business Administration.

Emrich, L. J., and M. R. Piedmonte. 1991. A method for generating high-dimensional multivariate binary variables. *American Statistician* 45: 302–304.

Leisch, F., A. Weingessel, and K. Hornik. 1998. On the generation of correlated artificial binary data. Working Paper Series No. 13, SFB Adaptive Information Systems and Modeling in Economics and Management Science, Vienna University of Economics and Business Administration.

Park, C. G., T. Park, and D. W. Shin. 1996. A simple method for generating correlated binary variates. *American Statistician* 50: 306–310.

About the author

Minxing Chen is a senior statistical analyst in the Department of Biostatistics at the University of Texas MD Anderson Cancer Center.