

Generating Multivariate Normal Data by Using PROC IML

Lingling Han, University of Georgia, Athens, GA

1 Abstract

Methods of generating multivariate normal data are discussed, statistical and graphic methods are used to check the generated data sets. Codes of both non-macro version and macro version are provided, some SAS "tricks" are provided as well.

2 Introduction

In simulation studies in statistics, there are many situations that we need to generate data from a multivariate normal distribution. By multivariate normal data we mean joint observations of p variables $Y_1, Y_2, \dots, Y_p$, in which each individual variable by itself is normally distributed, the variables are mutually correlated, and come from a joint multivariate normal distribution. The procedure for generation of multivariate normal data is similar to the univariate case, that is, we can generate pairs of independent normals and then multiplied that pairs by the Cholesky square root of the desired variance-covariance matrix. One way to do that is to obtain the formula for the Cholesky square root of the variance-covariance matrix, and which is easy for bivariate normal data. However, it becomes complicated when p is large. An alternative method by using *PROC IML* can be used to accomplish the desired data easily by using matrix computations. Function *HALF* in *PROC IML* can be used to obtain the Cholesky square root of the desired variance-covariance matrix.

3 Data generation and the corresponding codes

3.1 Generate the bivariate normal data

As mentioned in introduction, obtaining the formula for the Cholesky square root of the desired variance-covariance matrix is easy for bivariate normal data, we introduced it as a special case for the multivariate normal data when $p = 2$.

Suppose that we want to obtain that

$$\begin{pmatrix} Y_1 \\ Y_2 \end{pmatrix} \sim N \left(\begin{pmatrix} \mu_1 \\ \mu_2 \end{pmatrix}, \Sigma \right),$$

where the variance-covariance matrix Σ is

$$\Sigma = \begin{pmatrix} \sigma_1^2 & \rho\sigma_1\sigma_2 \\ \rho\sigma_1\sigma_2 & \sigma_2^2 \end{pmatrix},$$

then we can obtain $\Sigma^{1/2}$ as

$$\Sigma^{1/2} = \begin{pmatrix} \sigma_1 & 0 \\ \rho\sigma_2 & \sqrt{\sigma_2^2(1-\rho^2)} \end{pmatrix},$$

so $Y_1 = \mu_1 + \sigma_1 * rannor1$ and $Y_2 = \mu_2 + \rho * \sigma_2 * rannor1 + \sqrt{\sigma_2^2(1-\rho^2)} * rannor2$, where $rannor1$ and $rannor2$ are two independent random variables. Thus, we can use the following code to generate the bivariate normal data:

```
/* Generate the bivariate normal data */
data one;
  mean1=0;  *mean for y1;
  mean2=10; *mean for y2;
  sig1=2;   *SD for y1;
  sig2=5;   *SD for y2;
  rho=0.5;  *Correlation between y1 and y2;
  do i = 1 to 1000;
    r1 = rannor(1245);
    r2 = rannor(2923);
    y1 = mean1 + sig1*r1;
    y2 = mean2 + rho*sig2*r1+sqrt(sig2**2-sig2**2*rho**2)*r2;
    output;
  end;
keep y1 y2;
*proc print;
run;
```

Such that Y_1 and Y_2 are bivariate normally distributed.

3.2 Generate the multivariate normal data by using PROC IML

As discussed, obtaining the formula for the Cholesky square root of the desired variance-covariance matrix is complicated when p is large. Next, we will introduce an alternative method by using *PROC IML*.

In matrix notation the random variable is expressed as a vector $\mathbf{Y}' = (Y_1, Y_2, \dots, Y_p)$ which has a multivariate normal distribution with mean vector $\mu' = (\mu_1, \mu_2, \dots, \mu_p)$ and variance-covariance matrix

$$\Sigma = \begin{pmatrix} \sigma_1^2 & \sigma_{12} & \dots & \sigma_{1p} \\ \sigma_{12} & \sigma_2^2 & \dots & \sigma_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ \sigma_{p1} & \sigma_{p2} & \dots & \sigma_p^2 \end{pmatrix},$$

each $Y_i (i = 1, 2, \dots, p)$ has a $N(\mu_i, \sigma_i^2)$ distribution, the covariance of Y_i and Y_j is σ_{ij} , and the correlation between Y_i and Y_j is $\rho_{ij} = \frac{\sigma_{ij}}{\sigma_i \sigma_j}$. Now suppose $Z_i (i = 1, 2, \dots, p)$ are independent and have a $N(0, 1)$ distribution. Then the vector $\mathbf{Z}' = (Z_1, Z_2, \dots, Z_p)$ has mean vector $\mathbf{0}' = (0, 0, \dots, 0)$, and the covariance between Z_i and Z_j is 0. Therefore the variance-covariance matrix for $\mathbf{Z}'$ is the identity matrix.

If A is a symmetric $p \times p$ matrix, then the statement $T = HALF(A)$ returns T as an upper-triangular $p \times p$ matrix such that $T'T = A$, where A must be positive-definite, otherwise the *HALF* function will cause an error which stops the program. This function is now applied to the variance-covariance matrix of $\mathbf{Y}$.

If $T = HALF(\Sigma)$, then the transformation $Y = \mu + T'Z$ results in a random vector $\mathbf{Y}$ with the desired multivariate normal distribution. Because each Y is a linear combination of normal variates, $\mathbf{Y}$ is normally distributed. To check that the mean and variance-covariance are correct, the expected value of Y is $E(Y) = E(\mu + T'Z) = \mu + 0 = \mu$, and the variance of Y is

$$\begin{aligned}
Var(Y) &= E[(Y - E(Y))(Y - E(Y))'] \\
&= E[(T'Z)(T'Z)'] \\
&= E[T'ZZ'T] \\
&= T'E[ZZ']T \\
&= T'IT \\
&= T'T \\
&= \Sigma
\end{aligned}$$

If instead, what we have is a correlation matrix R , the variance-covariance matrix Σ can be obtained from R by forming a diagonal matrix, D , whose elements are the standard deviation of each Y . The function to do this in *PROC IML* is *DIAG*. Then we pre-multiply and post-multiply R by D to obtain Σ :

$$DRD' = \begin{pmatrix} \sigma_1 & 0 & \dots & 0 \\ 0 & \sigma_2 & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \dots & 0 & \sigma_p \end{pmatrix} \times \begin{pmatrix} 1 & \rho_{12} & \dots & \rho_{1p} \\ \rho_{12} & 1 & \ddots & \vdots \\ \vdots & \ddots & \ddots & \rho_{p-1,p} \\ \rho_{1p} & \dots & \rho_{p-1,p} & 1 \end{pmatrix} \times \begin{pmatrix} \sigma_1 & 0 & \dots & 0 \\ 0 & \sigma_2 & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \dots & 0 & \sigma_p \end{pmatrix} = \Sigma.$$

3.2.1 Code of generating the multivariate normal data by using PROC IML - nonmacro version

The program in this subsection is a non-macro version program which generates 1000 observations of the variables Y_1, Y_2, Y_3 , beginning with the correlation matrix R and a vector of means $\mu = (\mu_1, \mu_2, \mu_3)'$ and standard deviations $\sigma = (\sigma_1, \sigma_2, \sigma_3)'$ read instream as variables using a *CARDS* statement.

```

/* Generate the multivariate normal data in SAS/IML */
/* non-macro version */
data MVN_par; /* data for the parameter for the multivariate normal data */
input r1 r2 r3 means vars;
cards;
  1.0 -0.5  0.9    100  2
 -0.5  1.0 -0.7    200  5
  0.9 -0.7  1.0    300 10
;
proc iml;
  use MVN_par;
  read all var {r1 r2 r3} into R;
  read all var {means}    into mu;
  read all var {vars}     into sigma;
  p = ncol(R);
  diag_sig = diag( sigma );
  DRD = diag_sig * R * diag_sig';
  U = half(DRD);
  do i = 1 to 1000;

```

```

        z = rannor( j(p,1,1234));
        y = mu + U' * z;
        yprime = y';
        yall = yall // yprime;
    end;
    varnames = { y1 y2 y3 };
    create my_MVN from yall (|colname = varnames|);
    append from yall;
    quit;
proc print data=my_MVN;
run;

```

3.2.2 Code of generating the multivariate normal data by using PROC IML - macro version

The program in this subsection is a macro version program which is good for any dimension multivariate normal data with variance-covariance matrix and means as macro arguments.

```

%macro mvn(varcov=, means=, n=, myMVN=);
/* arguments for the macro:
    1. covcov: data set for variance-covariance matrix
    2. means:   data set for mean vector
    3. n:       sample size
    4. myMVN:   output data set name */
proc iml;
    use &varcov;                /* read in data for variance-covariance matrix */
    read all into sigma;
    use &means;                 /* read in data for means */
    read all into mu;
    p = nrow(sigma);           /* calculate number of variables */
    n = &n;
    l = t(half(sigma));        /* calculate cholesky root of cov matrix */
    z = normal(j(p,&n,1234));   /* generate nvars*samplesize normals */
    y = l*z;                   /* premultiply by cholesky root */
    yall = t(repeat(mu,1,&n)+y); /* add in the means */
    varnames = { y1 y2 y3 };
    create &myMVN from yall (|colname = varnames|);
    append from yall;
    quit;
%mend mvn;
data means1;
input x @@;
cards;
100 200 300
;
run;
data varcov1;
input x1-x3;
cards;

```

```

4 -5 18
-5 25 -35
18 -35 100
;
run;
%mvn(varcov=varcov1, means=means1, n=1000, myMVN=my_MNV)
proc print data=my_MNV;
run;

```

3.3 Statistical and graphic methods to check the generated multivariate normal data

Several procedures can be used to illustrate that the data sets generated have the desired properties, including statistical and graphic methods. For example, means, standard deviation and normality test for each variable (PROC UNIVARIATE), pairwise correlations (PROC CORR), bar charts of each variable (PROC GCHART), and pairwise plots (PROC GPLOT). Follows are the codes to check the generated data from previous section.

```

proc univariate normal noprint data=test1;
var y1 y2 y3;
output out=new mean=avg1 avg2 avg3 std=std1 std2 std3 probn=prob1 prob2 prob3;
run;
proc print data=new;
run;
proc corr data=test1 noprint outp=mycorr;
var y1 y2 y3;
run;
proc print data=mycorr;
run;
goptions reset=all;
symbol1 value=dot cv=blue height=0.5 width=2;
proc gplot data=test1;
plot y1*y2 y2*y3 y1*y3;
run;
proc gchart data=test1;
vbar y1 y2 y3;
run;

```

4 Conclusion

This paper discussed the methods of generation of data from multivariate normal distribution, the function *HALF* in *PROC IML* can be easily used to obtain the Cholesky square root of the variance-covariance matrix, and then obtain the desired multivariate normal data. The codes are useful for any dimensional multivariate normal data, as long as you have the parameters for the distribution, you can use the macro to obtain the data you want. If the sample size for the data set are large, use "Batch Submit with SAS" can be faster and obtain the results sooner.

5 Acknowledgments

The author wishes to thank Dr. Nancy Lyons, who kindly shared some of the ideas, and supported the development of the work in this paper.

6 Contact Information

Lingling Han
University of Georgia
Athens, GA 30605
Work Phone: 706-542-3314
Email: llhan@stat.uga.edu