

Generating Multivariate Random Associated Data

Part 1

From time-to-time a researcher needs to develop a script or an application to collect and analyze data. They may also need to test their application under a variety of scenarios prior to data collection. However, because the data has not been collected yet it is necessary to create test data. Creating continuous data is relatively simple and is fairly straight forward using the Cholesky (pronounced kol-eh-ski) decomposition. This approach takes an original X variable and uses the Cholesky transformation to create a new, correlated, Y variable. To make things simple and straight forward this example will generate data from the a random normal distribution $N(0,1)$.

The reason this approach is so useful is that that correlation structure can be specifically defined. The scripts can be used to create many different variables with different correlation structures. The method to transform the data into correlated variables is seen below using the correlation matrix R .

$$R = \begin{bmatrix} 1 & 0.8 & 0.2 \\ 0.8 & 1 & 0.7 \\ 0.2 & 0.7 & 1 \end{bmatrix} \quad (1)$$

Once the correlation matrix is set the researcher takes the Cholesky decomposition of the correlation matrix. Multiplying the Cholesky decomposition of the correlation matrix by the data matrix the resulting matrix is a transformed dataset with the specified correlation.

$$W = [Cholesky(R)][X] \quad (2)$$

The R code from below will generate a correlation matrix of

$$R = \begin{bmatrix} 1 & 0.7997999 & 0.1998661 \\ 0.7997999 & 1 & 0.7000217 \\ 0.1998661 & 0.7000217 & 1 \end{bmatrix} \quad (3)$$

```
R <- matrix(cbind(1,.80,.2,
                  .80,1,.7,
                  .2,.7,1),nrow=3);
U <- t(chol(R));
nvars <- dim(U)[1];
numobs <- 100000;
set.seed(1);
random.normal <- matrix(rnorm(nvars*numobs,0,1), nrow=nvars, ncol=numobs);
X <- U %*% random.normal;
newX <- t(X);
raw <- as.data.frame(newX);
orig.raw <- as.data.frame(t(random.normal));
names(raw) <- c("response","predictor1","predictor2");

cor(raw);

plot(head(raw, 100));
plot(head(orig.raw,100));
```