SOCIETY OF ACTUARIES

Article from:

Risk Management

March 2008 – Issue 12

A Toy Copula[1] ERM Model in R[2]

by Steve Craighead

A simple ERM model entails setting up various risk sub-models and creating a dependency relationship between these risks. After this is accomplished all one needs to do is to simulate for a given number of trials (say 1,000) and aggregate the dependent risk values. From these aggregated results one can then either determine VaR or conditional tail expectation (CTE) at a specific percentile. later in this article, we will construct our ERM toy model, which will demonstrate how to link statistical sub-models with copula dependency models using the R copula package.

The R copula package models the Frank, Gumbel and Clayton copulas within the Archimedean family, as well as, the multivariate normal and Student-t copulas. The Archimedean copulas are limited to dimensions less than seven since the resultant multivariate probability density function (PDF) is not available due to difficulty in symbolically differentiating the associated multivariate cumulative distribution function (CDF).

In brief, the steps that the reader needs to follow are:

A. Specify the copula that will model the dependence.
B. Specify the multivariate distribution using the copula defined in step A by using the mvdc() function. You must also define the marginal distributions associated with each risk. You will also specify the marginal parameters which will be discussed below.
C. Supply the marginal data associated with each risk.
D. Fit the model.
E. Examine the results.

Before we actually create the toy ERM model, we need to discuss some basics in R.

continued on page 14 ◗

[1] This entire article assumes that the reader is acquainted with both the concept of copulas in the modeling of dependence in ERM modeling and how to use the R language with statistical modeling. However, for further information on copulas, you may read the excellent survey article "Understanding Relationships Using Copulas" by Frees and Valdez in the *NAAJ* Volume 2, Number 1 of 1998. If you prefer to experiment with copulas in Excel instead of R, please refer to the workbook and articles created by Sam Cox and Don Behan on *http://www.behan.ws.*

[2] The R language is very popular open source statistical modeling environment, which you can obtain from the Web Page *http://cran.r-project.org.* If you are interested in building the toy model in this article, you will need to download the base R Binary for the operating system of your choice (Windows, Linux or MAC). Also, you will need to download the contributed copula package. You can do this initially when you download the base system, or you can use the package installation facility when you start R.

Steve Craighead, ASA, MAAA, is an actuarial consultant at Towers Perrin in Atlanta, Ga. He can be reached at *steven.craighead@towersperrin.com.*

A Toy Copula[1] ERM Model in R[2]

◗ continued from page 13

The R language has many different univariate distributions available for modeling the marginals. A statistical distribution in R uses three separate functions for modeling (with the possibility of a fourth). For example, if you wanted to model a univariate normal distribution, you would use either the dnorm, pnorm, qnorm (and rnorm) functions. The dnorm function models the density, the pnorm models the distribution (CDF) and the qnorm is the quantile function, which is the inverse function of the CDF. The fourth function is rnorm, which can be used to generate random normal variates. Other example distributions functions are dt, pt, qt and rt for the Student-t, and dexp, pexp, qexp and rexp for the exponential distribution. Each of these functions requires specific model parameters. The normal distribution has the model parameters mean and sd for the mean and standard deviation. The Student-t distribution requires the parameter df for the degrees of freedom. The exponential distribution uses the rate parameter.

One must also load the copula package after starting R. This is done in Windows by choosing the copula package when using the Load Package option under the Packages option on the command list at the top of R.

When carrying out step B (see page 13), the mvdc() function creates a multivariate distribution object in R. This function has three major inputs. The first is the copula. The second is a list of the specific marginals and the third is a list containing lists of associated marginal parameters.

For instance, say you want to model a bivariate CDF using the Gumbel copula (with parameter of three) and the first marginal is a normal distribution with a mean of 10 and a standard deviation of two and the second marginal is an exponential with a rate of two.

A. First, You need to use the command

```
gmb<-gumbelCopula(3, dim = 2)
```

to create the copula object. The symbol "<-" is used as the assignment operator. The "dim=2" assures that we are creating a bivariate CDF. You can specify up to six dimensions within the R Copula library.

B. Next you specify the bivariate distribution by using the command:

```
myCDF<- mvdc(gmb, c("norm","exp"), list
(list(mean=10,sd=2),list(rate=2)))
```

Notice the first parameter in mvdc is the gmb object defined in step A. The second parameter is a generic list of the two marginal distributions (produced using the c() function). The third parameter is much more complex, where the list() function is used three times. Here we have a list made of two separate lists. The first of these two separate lists are the parameters that model the normal marginal as discussed above. The second of these lists contains the parameter required to model the exponential marginal. The list() function is not as generic as the c() function, but the mvdc

function expects the marginal parameter defined in this fashion.

C. Next you need to supply the risk data associated with each marginal. Since this is a toy, we will actually use the myCDF multivariate distribution as defined to generate the data (a bit incestuous, but okay for this demonstration). Here you will use the rmvdc() function which will generate random variates from myCDF. Use the command

```
x <- rmvdc(myCDF,1000)
```

to generate 1000 samples from your bivariate CDF. You could also supply your own data x as in an R matrix with 1000 rows and two columns. For instance, if you were able to use identical scenarios and produce separate various risk values based on these scenarios, and collect these together into an R matrix, you could then design your copula multivariate distribution and use the fitting algorithm to fit your actual data.

D. Next, take the data in the matrix x and determine the best parameters in myCDF by using the fitting function fitMvdc() (which uses maximum likelihood). This function requires the x matrix, the myCDF object and a generic c() list of initial guesses to the parameters. Note that the first parameter in this list will be the parameter associated with the Gumbel copula model (which we assumed as 3). The remaining values are associated with the parameters of the marginals. Suppose that you use the command:

```
Fitted<-fitMvdc(x, myCDF, c(3,9,1,1))
```

Here the 3 is the Gumbel parameter, the 9 is the initial guess for the normal mean, 1 for the normal sd parameter and 1 for the exponential rate parameter. Here starting values were different than the original model, so that in Step E, we can see the goodness of fit. Note: you may get a warning message stating that there are so many separate warnings and that you should enter the command "warnings()" to display these. These warnings arise when the fitting algorithm moves the parameter estimates outside the acceptable values. At this time, you may disregard these.

E. Once the model fitting has completed, you can examine the results of the fit by just issuing the command "Fitted." This will display to the screen the results. For instance the following is the results obtained by the author. Your results may vary due to the random values that were generated in matrix x.

```
> Fitted
The ML estimation is based on
1000 observations.
Margin 1:
            Estimate    Std. Error
m1.mean   10.157765   0.05885397
m1.sd      1.984802   0.03243711
Margin 2:
            Estimate    Std. Error
m2.rate    1.893372   0.05849316
Copula:
            Estimate    Std. Error
param      2.819898   0.08826639
The maximized loglikelihood is  -1787.369
The convergence code is 0
>
```

Observe both the relative parameters and their standard error, and how close the estimates are to the original parameters originally set up.

Also, one can examine the contour plot of the bivariate density function by using the following commands:

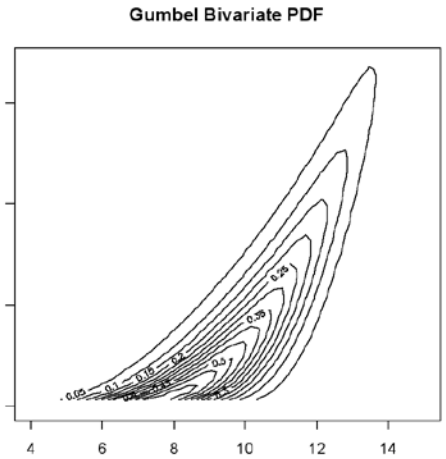
continued on page 16 ◗

A Toy Copula[1] ERM Model in R[2]

◗ continued from page 15

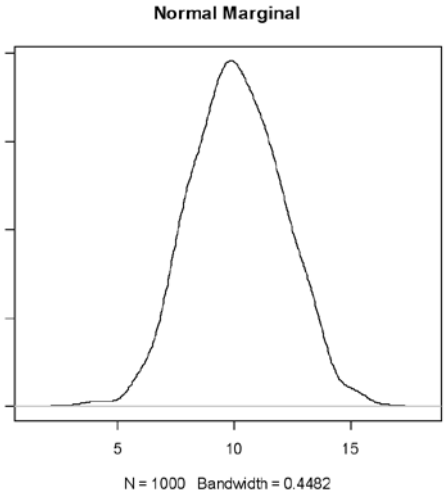
```
contour(myCDF,dmvdc,xlim=c(4,15),
ylim=c(0,1.7))
title("Gumbel Multivariate PDF")
```

These produce the following graph:



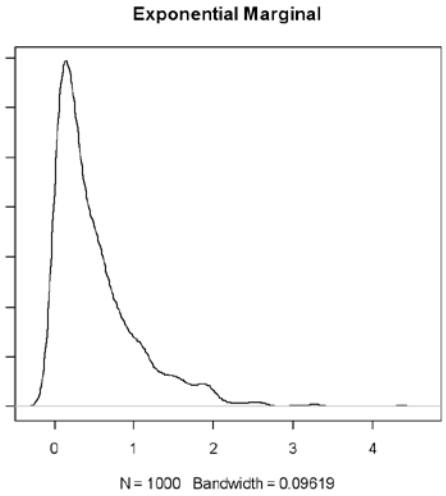
One obtains the graph of the normal marginal by using these commands:

```
K <- density(x[,1]) #x[,1] gets risk 1 results
K$Call <- "Normal Marginal"
plot(K)
```



To obtain the exponential marginal, use these commands:

```
K <- density(x[,2]) #x[,2] obtains risk 2
K$Call <- "Exponential Marginal"
plot(K)
```

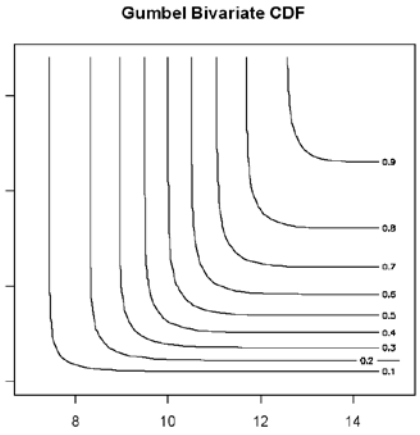


Note how that the PDF is not symmetric, which is a characteristic of the Gumbel Copula since it is only defined in the right tail.

Also, one can examine the contour plot of the bivariate cumulative function by using the following commands:

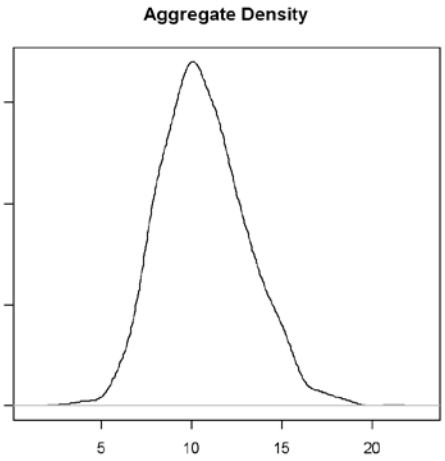
```
contour(myCDF,pmvdc,xlim=c(7,15),ylim=
c(0,1.7))
title("Gumbel Bivariate CDF")
```

This produces the following graph:



Of course, if you add up the random samples, you will have the total dependent risks! By using these following commands, R will display the density of the total risks:

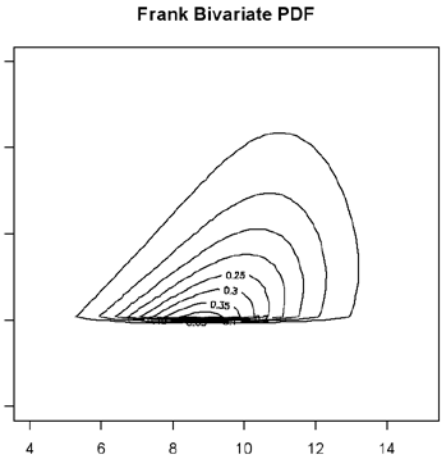
```
Y<- apply(x,1,"sum") #sum across columns
of x
K<-density(Y)
K$Call <- "Aggregate Risks"
plot(K)
```



With just the above commands you may examine other copulas such as the Frank copula using the commands:

```
frank <- frankCopula(param=2,dim=2)
myCDF<-mvdc(frank,c("norm","exp"),list
(list(mean=10,sd=2),list(rate=2)))
contour(myCDF,dmvdc,xlim=c(4,15),ylim=
c(-0.5,1.5))
title("Frank Bivariate PDF")
```

You will obtain the following density contour:



Other things that you can do, is to increase the number of dimensions and use other marginals. Also, you can use the rmvdc() command as we did in step C above and simulate your dependency multivariate distribution.

Though this toy model only uses two risks, you can use up to six risks, by changing the definition of the copula as well as specifying the different sub-models. Hopefully, the examples above are explicit enough to let you do various what-ifs on your own data. ✦

Though this toy model only uses two risks, you can use up to six risks, by changing the definition of the copula as well as specifying the different sub-models.

Bibliography

R Development Core Team (2006). R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. ISBN 3-900051-07-0, *http://www.R-project.org*.