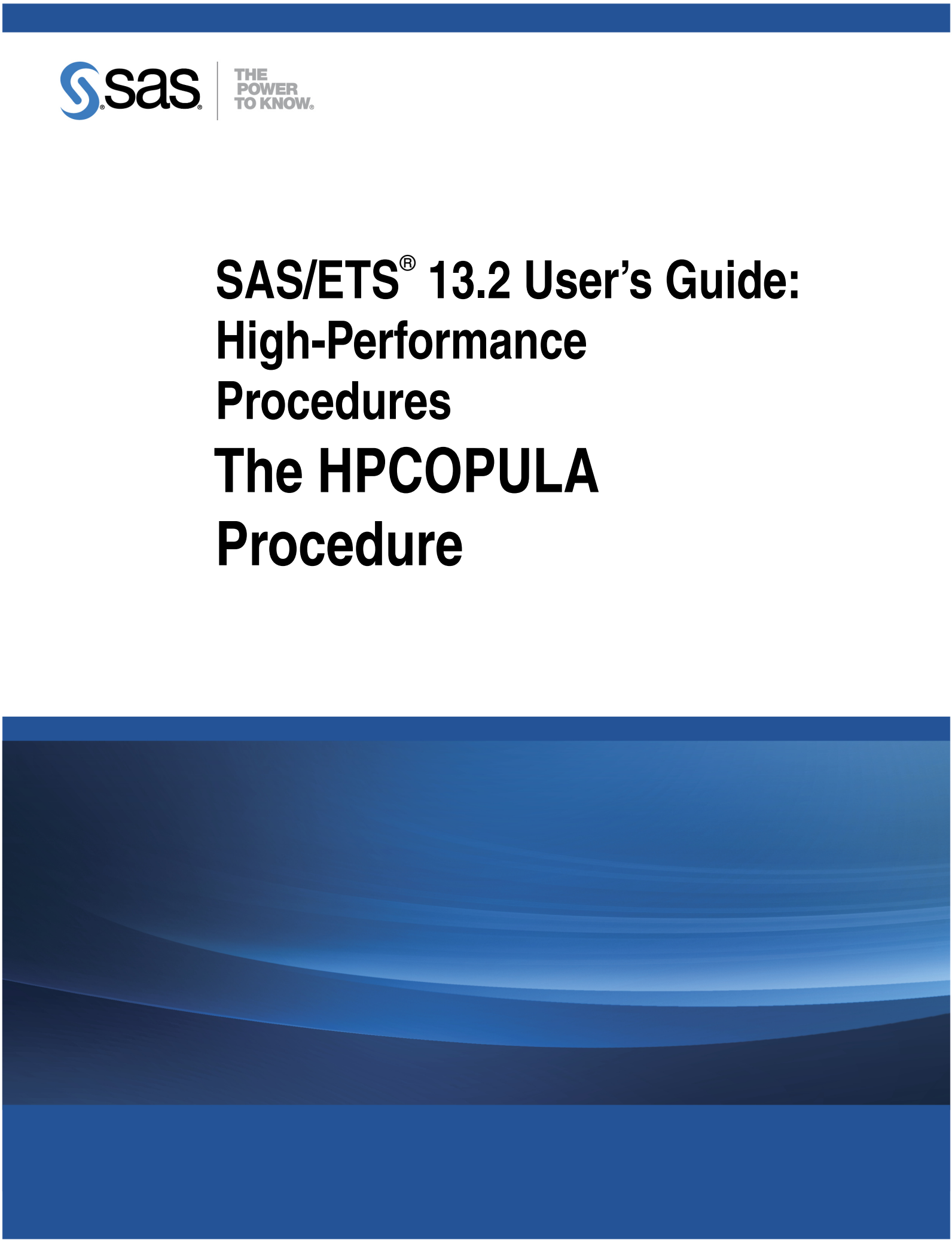

SAS/ETS® 13.2 User's Guide: High-Performance Procedures

The HPCOPULA Procedure

The correct bibliographic citation for this manual is as follows: SAS Institute Inc. 2014. *SAS/ETS® 13.2 User's Guide: High-Performance Procedures*. Cary, NC: SAS Institute Inc.

SAS/ETS® 13.2 User's Guide: High-Performance Procedures

SAS Institute Inc., SAS Campus Drive, Cary, North Carolina 27513.

August 2014

SAS provides a complete selection of books and electronic products to help customers use SAS® software to its fullest potential. For more information about our offerings, visit **support.sas.com/bookstore** or call 1-800-727-3228.

Chapter 5
The HPCOPULA Procedure

Contents

Overview: HPCOPULA Procedure

The HPCOPULA procedure is a high-performance version of the SAS/ETS COPULA procedure, which simulates data from a specified copula. Unlike the COPULA procedure, which can be run only on an individual workstation, the HPCOPULA procedure takes advantage of a computing environment in which the optimization task can be distributed to one or more nodes. In addition, each node can use one or more threads to perform the optimization on its subset of the data. When several nodes are used and each node uses several threads to carry out its part of the work, the result is a highly parallel computation that provides a dramatic gain in performance.

You can use the HPCOPULA procedure to read and write data in distributed form and perform analyses either in single-machine mode or in distributed mode. For more information about the execution mode of SAS High-Performance Analytics procedures, see the section "Processing Modes" on page 10 in Chapter 3, "Shared Concepts and Topics."

The HPCOPULA procedure is specifically designed to operate in the high-performance distributed environment. By default, PROC HPCOPULA performs computations in multiple threads.

PROC HPCOPULA Features

The HPCOPULA procedure enables you to simulate a specified copula, and it supports the following types of copulas:

- normal copula
- *t* copula
- Archimedean copulas:
  - Clayton copula
  - Frank copula
  - Gumbel copula

Getting Started: HPCOPULA Procedure

This example illustrates the use of PROC HPCOPULA. The data are daily returns on several major stocks. The main purpose of this example is to simulate from the joint distribution of stock returns a new sample of a specified size, provided that the parameter estimates of the copula model that is used are available.

In the following statements, the DEFINE statement specifies a normal copula named COP, and the CORR= option specifies that the data set Estimates be used as the source for the model parameters. The NDRAWS=1000000 option in the SIMULATE statement generates one million observations from the normal copula. The OUTUNIFORM= option specifies the name of the SAS data set to contain the simulated sample that has uniform marginal distributions. The PERFORMANCE statement requests that the analytic computations use two nodes in the distributed computing environment and two threads in each node. Note that this syntax does not require the DATA= option.

```
/* Copula simulation of uniforms */
proc hpcopula;
   var ret_ibm ret_msft ret_bp ret_ko ret_duk;
   define cop normal (corr = estimates);
   simulate cop / ndraws     = 1000000
                  outuniform = simulated_uniforms;
   PERFORMANCE nodes=2 nthreads=2 details;
run;
```

The simulated data are contained in the new SAS data set, Simulated_Uniforms.

Syntax: HPCOPULA Procedure

The following statements are available in the HPCOPULA procedure:

PROC HPCOPULA *options* **;**
VAR *variables* **;**
DEFINE *name copula-type* < *(parameter-value-options . . .)* > **;**
SIMULATE < *copula-name-list* > */ options* **;**

Functional Summary

Table 5.1 summarizes the statements and options that the HPCOPULA procedure uses.

Table 5.1 PROC HPCOPULA Functional Summary

Description	Statement	Option
Data Set Options		
Specifies the input data set that contains the correlation matrix for elliptical copulas	DEFINE	CORR=
Declaring the Role of Variables		
Specifies the names of the variables to use in copula fitting or in simulation	VAR	
Copula Simulation Options		
Specifies the random sample size	SIMULATE	NDRAWS=
Specifies the random number generator seed	SIMULATE	SEED=
Output Control Options		
Specifies the output data set to contain the random samples from the simulation with uniform marginal distribution	SIMULATE	OUTUNIFORM=

PROC HPCOPULA Statement

PROC HPCOPULA ;

The PROC HPCOPULA statement invokes the HPCOPULA procedure.

DEFINE Statement

DEFINE *name copula-type* < *(parameter-value-options ...)* > **;**

The DEFINE statement specifies the relevant information about the copula that is used for the simulation. You can specify the following arguments:

name specifies the name of the copula definition. You can be use this *name* later in the SIMULATE statement.

copula-type specifies the type of copula. You must specify one of the following copula types, which are described in the section "Details: HPCOPULA Procedure" on page 114:

NORMAL fits the normal copula.

T fits the *t* copula.

CLAYTON fits the Clayton copula.

FRANK fits the Frank copula.

GUMBEL fits the Gumbel copula.

parameter-value-options
specify the input parameters that are used to simulate the specified copula. These options must be appropriate for the type of copula specified. You can specify the following *parameter-value-options*:

CORR=*SAS-data-set*
specifies the data set that contains the correlation matrix to use for elliptical copulas. If the correlation matrix is valid but its elements are not submitted in order, then you must provide the variable names in the first column of the matrix, and these names must match the variable names in the VAR statement. See Output 5.1.1 for an example of a correlation matrix input in this form. If the correlation matrix elements are submitted in order, the first column of variable names is not required. You can use this option for normal and *t* copulas.

DF=*value*
specifies the degrees of freedom. You can use this option for *t* copulas.

THETA=*value*
specifies the parameter value for the Archimedean copulas.

The DEFINE statement is used with the SIMULATE statement.

SIMULATE Statement

SIMULATE < *copula-name-list* >/*options* **;**

The SIMULATE statement simulates data from a specified copula model. The copula name specification is the name of a defined copula as specified by *name* in the DEFINE statement.

NDRAWS=*integer*
specifies the number of draws to generate for this simulation. By default, NDRAWS=100.

OUTUNIFORM=*SAS-data-set*
specifies the output data set to contain the result of the simulation in uniform margins. You can use this option when MARGINALS=UNIFORM or MARGINALS=EMPIRICAL. If MARGINALS=EMPIRICAL, then this option enables you to obtain the samples that are simulated from the joint distribution specified by the copula, where all marginal distributions are uniform. The data are not created if you do not specify this option.

SEED=*integer*
specifies the seed for generating random numbers for the simulation. If you do not provide the seed, a random number is used as the seed.

PERFORMANCE Statement

PERFORMANCE < *performance-options* > **;**

The PERFORMANCE statement specifies *performance-options* to control the multithreaded and distributed computing environment and requests detailed performance results of the HPCOPULA procedure. You can also use the PERFORMANCE statement to control whether the HPCOPULA procedure executes in SMP or MPP mode. You can specify the following *performance-options*:

DETAILS
requests a table that shows a timing breakdown of the PROC HPCOPULA steps.

NODES=*n*
specifies the number of nodes in the distributed computing environment, provided that the data are not processed alongside the database.

NTHREADS=*n*
specifies the number of threads for analytic computations and overrides the SAS system option THREADS | NOTHREADS. If you do not specify the NTHREADS= option, PROC HPCOPULA creates one thread per CPU for the analytic computations.

For more information about the PERFORMANCE statement, see the section "PERFORMANCE Statement" on page 36 in Chapter 3, "Shared Concepts and Topics."

VAR Statement

VAR *variables* **;**

The VAR statement specifies the variable names in the input data set that is specified by the DATA= option in the PROC HPCOPULA statement. The subset of variables in the data set is used for the copula models in the FIT statement. If there is no input data set, the VAR statement creates the list of variable names for the SIMULATE statement.

Details: HPCOPULA Procedure

Sklar's Theorem

The copula models are tools for studying the dependence structure of multivariate distributions. The usual joint distribution function contains the information both about the marginal behavior of the individual random variables and about the dependence structure between the variables. The copula is introduced to decouple the marginal properties of the random variables and the dependence structures. An m-dimensional *copula* is a joint distribution function on $[0, 1]^m$, where all marginal distributions are standard uniform. The common notation for a copula is $C(u_1, \ldots, u_m)$.

The Sklar (1959) theorem shows the importance of copulas in modeling multivariate distributions. The first part of the theorem states that a copula can be derived from any joint distribution functions, and the second part asserts the opposite: that any copula can be combined with any set of marginal distributions to result in a multivariate distribution function. The theorem follows:

- Let F be a joint distribution function, and let $F_j, j = 1, \ldots, m$, be the marginal distributions. Then there exists a copula $C : [0, 1]^m \rightarrow [0, 1]$ such that

  $$F(x_1, \ldots, x_m) = C(F_1(x_1), \ldots, F_m(x_m))$$

  for all $x_1, \ldots, x_m$ in $[-\infty, \infty]$. Moreover, if the margins are continuous, then C is unique; otherwise C is uniquely determined on $\mathrm{Ran}F_1 \times \ldots \times \mathrm{Ran}F_m$, where $\mathrm{Ran}F_j = F_j([-\infty, \infty])$ is the range of F_j.
- The converse is also true. That is, if C is a copula and $F_1, \ldots, F_m$ are univariate distribution functions, then the multivariate function that is defined in the preceding equation is a joint distribution function with marginal distributions $F_j, j = 1, \ldots, m$.

Dependence Measures

There are three basic types of dependence measures: linear correlation, rank correlation, and tail dependence. Linear correlation is given by

$$\rho \equiv \mathrm{corr}(X, Y) = \frac{\mathrm{cov}(X, Y)}{\sqrt{\mathrm{var}(X)}\sqrt{\mathrm{var}(Y)}}$$

The linear correlation coefficient contains very limited information about the joint properties of the variables. A well-known property is that zero correlation does not imply independence, whereas independence implies zero correlation. In addition, there are distinct bivariate distributions that have the same marginal distribution and the same correlation coefficient. These results suggest that caution must be used in interpreting the linear correlation.

Another statistical measure of dependence is rank correlation, which is nonparametric. For example, Kendall's tau is the covariance between the sign statistics $X_1 - \tilde{X}_1$ and $X_2 - \tilde{X}_2$, where $(\tilde{X}_1, \tilde{X}_2)$ is an independent copy of (X_1, X_2):

$$\rho_\tau \equiv E[\mathrm{sign}(X_1 - \tilde{X}_1)(X_2 - \tilde{X}_2)]$$

The sign function (sometimes written as sgn) is defined as

$$\text{sign}(x) = \begin{cases} -1 & \text{if} x \leq 0 \\ 0 & \text{if} x = 0 \\ 1 & \text{if} x \geq 0 \end{cases}$$

Spearman's rho is the correlation between the transformed random variables:

$$\rho_S(X_1, X_2) \equiv \rho(F_1(X_1), F_2(X_2))$$

The variables are transformed by their distribution functions so that the transformed variables are uniformly distributed on $[0, 1]$. The rank correlations depend only on the copula of the random variables and are indifferent to the marginal distributions. Like linear correlation, rank correlation has its limitations. In particular, different copulas result in the same rank correlation.

A third measure, tail dependence, focuses on only part of the joint properties between the variables. Tail dependence measures the dependence when both variables have extreme values. Formally, they can be defined as the conditional probabilities of quantile exceedances. There are two types of tail dependence:

- Upper tail dependence is defined as

  $$\lambda_u(X_1, X_2) \equiv \lim_{q->1^-} P(X_2 > F_2^{-1}(q) | X_1 > F_1^{-1}(q))$$

  when the limit exists and $\lambda_u \in [0, 1]$. Here F_j^{-1} is the quantile function (that is, the inverse of the CDF).
- Lower tail dependence is defined symmetrically.

Normal Copula

Let $u_j \sim U(0, 1)$ for $j = 1, \ldots, m$, where $U(0, 1)$ represents the uniform distribution on the $[0, 1]$ interval. Let Σ be the correlation matrix, where $m(m-1)/2$ parameters satisfy the positive semidefiniteness constraint. The normal copula can be written as

$$C_\Sigma(u_1, u_2, \ldots u_m) = \mathbf{\Phi}_\Sigma\Big(\Phi^{-1}(u_1), \ldots \Phi^{-1}(u_m)\Big)$$

where Φ is the distribution function of a standard normal random variable and $\mathbf{\Phi}_\Sigma$ is the m-variate standard normal distribution with mean vector 0 and covariance matrix Σ. That is, the distribution $\mathbf{\Phi}_\Sigma$ is $N_m(0, \Sigma)$.

Simulation

For the normal copula, the input of the simulation is the correlation matrix Σ. The normal copula can be simulated by the following steps, in which $\boldsymbol{U} = (U_1, \ldots, U_m)$ denotes one random draw from the copula:

1. Generate a multivariate normal vector $\boldsymbol{Z} \sim N(0, \Sigma)$, where Σ is an m-dimensional correlation matrix.
2. Transform the vector $\boldsymbol{Z}$ into $\boldsymbol{U} = (\Phi(Z_1), \ldots, \Phi(Z_m))^T$, where Φ is the distribution function of univariate standard normal.

The first step can be achieved by Cholesky decomposition of the correlation matrix $\Sigma = LL^T$, where L is a lower triangular matrix with positive elements on the diagonal. If $\tilde{\boldsymbol{Z}} \sim N(0, I)$, then $L\tilde{\boldsymbol{Z}} \sim N(0, \Sigma)$.

Student's *t* copula

Let $\Theta = \{(\nu, \Sigma) : \nu \in (1, \infty), \Sigma \in \mathbb{R}^{m \times m}\}$, and let t_ν be a univariate t distribution with ν degrees of freedom.

The Student's t copula can be written as

$$C_\Theta(u_1, u_2, \ldots u_m) = \boldsymbol{t}_{\nu,\Sigma}\left(t_\nu^{-1}(u_1), t_\nu^{-1}(u_2), \ldots, t_\nu^{-1}(u_m)\right)$$

where $\boldsymbol{t}_{\nu,\Sigma}$ is the multivariate Student's t distribution that has a correlation matrix Σ with ν degrees of freedom.

Simulation

The input parameters for the simulation are (ν, Σ). The t copula can be simulated by the following steps:

1. Generate a multivariate vector $\boldsymbol{X} \sim t_m(\nu, 0, \Sigma)$ that follows the centered t distribution with ν degrees of freedom and correlation matrix Σ.
2. Transform the vector $\boldsymbol{X}$ into $\boldsymbol{U} = (t_\nu(X_1), \ldots, t_\nu(X_m))^T$, where t_ν is the distribution function of univariate t distribution with ν degrees of freedom.

To simulate centered multivariate t random variables, you can use the property that $\boldsymbol{X} \sim t_m(\nu, 0, \Sigma)$ if $\boldsymbol{X} = \sqrt{\nu/s}\boldsymbol{Z}$, where $\boldsymbol{Z} \sim N(0, \Sigma)$ and the univariate random variable $s \sim \chi^2_\nu$.

Archimedean Copulas

Overview of Archimedean Copulas

Let function $\phi : [0, 1] \to [0, \infty)$ be a strict Archimedean copula generator function, and suppose that its inverse ϕ^{-1} is completely monotonic on $[0, \infty)$. A strict generator is a decreasing function $\phi : [0, 1] \to [0, \infty)$ that satisfies $\phi(0) = \infty$ and $\phi(1) = 0$. A decreasing function $f(t) : [a, b] \to (-\infty, \infty)$ is completely monotonic if it satisfies

$$(-1)^k \frac{d^k}{dt^k} f(t) \geq 0, k \in \mathbb{N}, t \in (a, b)$$

An Archimedean copula is defined as follows:

$$C(u_1, u_2, \ldots, u_m) = \phi^{-1}\left(\phi(u_1) + \cdots + \phi(u_m)\right)$$

The Archimedean copulas available in the HPCOPULA procedure are the Clayton copula, the Frank copula, and the Gumbel copula.

Clayton Copula

Let the generator function $\phi(u) = \theta^{-1}\left(u^{-\theta} - 1\right)$. A Clayton copula is defined as

$$C_\theta(u_1, u_2, \ldots, u_m) = \left[\sum_{i=1}^{m} u_i^{-\theta} - m + 1\right]^{-1/\theta}$$

where $\theta > 0$.

Frank Copula

Let the generator function be

$$\phi(u) = -\log\left[\frac{\exp(-\theta u) - 1}{\exp(-\theta) - 1}\right]$$

A Frank copula is defined as

$$C_\theta(u_1, u_2, \ldots, u_m) = \frac{1}{\theta}\log\left\{1 + \frac{\prod_{i=1}^{m}[\exp(-\theta u_i) - 1]}{[\exp(-\theta) - 1]^{m-1}}\right\}$$

where $\theta \in (-\infty, \infty)\backslash\{0\}$ for $m = 2$ and $\theta > 0$ for $m \geq 3$.

Gumbel Copula

Let the generator function $\phi(u) = (-\log u)^\theta$. A Gumbel copula is defined as

$$C_\theta(u_1, u_2, \ldots, u_m) = \exp\left\{-\left[\sum_{i=1}^{m}(-\log u_i)^\theta\right]^{1/\theta}\right\}$$

where $\theta > 1$.

Simulation

Suppose that the generator of the Archimedean copula is ϕ. Then the simulation method that uses a Laplace-Stieltjes transformation of the distribution function is given by Marshall and Olkin (1988), where $\tilde{F}(t) = \int_0^\infty e^{-tx} dF(x)$:

1. Generate a random variable V that has the distribution function F such that $\tilde{F}(t) = \phi^{-1}(t)$.
2. Draw samples from the independent uniform random variables $X_1, \ldots, X_m$.
3. Return $\boldsymbol{U} = (\tilde{F}(-\log(X_1)/V), \ldots \tilde{F}(-\log(X_m)/V))^T$.

The Laplace-Stieltjes transformations are as follows:

- For the Clayton copula, $\tilde{F} = (1 + t)^{-1/\theta}$, and the distribution function F is associated with a gamma random variable that has a shape parameter of θ^{-1} and a scale parameter of 1.
- For the Gumbel copula, $\tilde{F} = \exp(-t^{1/\theta})$, and F is the distribution function of the stable variable $\mathrm{St}(\theta^{-1}, 1, \gamma, 0)$, where $\gamma = [\cos(\pi/(2\theta))]^\theta$.

- For the Frank copula where $\theta > 0$, $\tilde{F} = -\log\{1 - \exp(-t)[1 - \exp(-\theta)]\}/\theta$, and F is a discrete probability function $P(V = k) = (1 - \exp(-\theta))^k/(k\theta)$. This probability function is related to a logarithmic random variable that has a parameter value of $1 - e^{-\theta}$.

For more information about simulating a random variable from a stable distribution, see Theorem 1.19 in Nolan (2010). For more information about simulating a random variable from a logarithmic series, see Chapter 10.5 in Devroye (1986).

For a Frank copula where $m = 2$ and $\theta < 0$, the simulation can be done through conditional distributions as follows:

1. Draw independent v_1, v_2 from a uniform distribution.
2. Let $u_1 = v_1$.
3. Let $u_2 = -\frac{1}{\theta}\log\left(1 + \frac{v_2(1-e^{-\theta})}{v_2(e^{-\theta v_1}-1)-e^{-\theta v_1}}\right)$.

OUTUNIFORM= Data Sets

The number of columns and the names of columns in OUTUNIFORM= data sets match the number and names of the *variables* in the VAR statement.

Examples: HPCOPULA Procedure

Example 5.1: Simulating Default Times

Suppose the correlation structure that is required for a normal copula function is already known. For example, the correlation structure can be estimated from the historical data on default times in some industries, but this estimation is not within the scope of this example. The correlation structure is saved in a SAS data set called Inparm. The following statements and their output in Output 5.1.1 show that the correlation parameter is set at 0.8:

```
proc print data = inparm;
run;
```

Output 5.1.1 Copula Correlation Matrix

Obs	Y1	Y2
1	1.0	0.8
2	0.8	1.0

The following statements use PROC HPCOPULA to simulate the data:

```
option set=GRIDHOST="&GRIDHOST";
```

```
 option set=GRIDINSTALLLOC="&GRIDINSTALLLOC";


/* simulate the data from bivariate normal copula */
proc hpcopula;
   var Y1-Y2;
   define cop normal (corr=inparm);
   simulate cop /
            ndraws     = 1000000
            seed       = 1234
            outuniform = normal_unifdata;
   PERFORMANCE nodes=4 nthreads=4 details
               host="&GRIDHOST" install="&GRIDINSTALLLOC";
run;
```

The VAR statement specifies the list of variables that contains the simulated data. The DEFINE statement assigns the name COP and specifies a normal copula that reads the correlation matrix from the Inparm data set. The SIMULATE statement refers to the COP label that is defined in the VAR statement and specifies several options: the NDRAWS= option specifies a sample size, the SEED= option specifies 1234 as the random number generator seed, and the OUTUNIFORM=NORMAL_UNIFDATA option names the output data set to contain the result of simulation in uniforms. The PERFORMANCE statement requests that the analytic computations be performed on four nodes in the distributed computing environment and four threads on each node. Output 5.1.2 shows the run time of this particular simulation experiment.

Output 5.1.2 Run-Time Performance

Performance Information	
Host Node	<< your grid host >>
Install Location	<< your grid install location >>
Execution Mode	Distributed
Number of Compute Nodes	4
Number of Threads per Node	4

Procedure Task Timing		
Task	**Seconds**	**Percent**
Simulation of Model	0.07	0.20%
Writing of output data	32.79	99.80%

The following DATA step transforms the variables from zero-one uniformly distributed to nonnegative exponentially distributed with parameter 0.5 and adds three indicator variables to the data set: SURVIVE1 and SURVIVE2 are equal to 1 if company 1 or company 2, respectively, has remained in business for more than three years, and SURVIVE is equal to 1 if both companies survived the same period together.

```
/* default time has exponential marginal distribution with parameter 0.5 */
data default;
   set normal_unifdata;
   array arr{2} Y1-Y2;
   array time{2} time1-time2;
   array surv{2} survive1-survive2;
   lambda = 0.5;
```

```
   do i=1 to 2;
      time[i] = -log(1-arr[i])/lambda;
      surv[i] = 0;
      if (time[i] >3) then surv[i]=1;
   end;
   survive = 0;
   if (time1 >3) && (time2 >3) then survive = 1;
run;
```

The first analysis step is to look at correlations between survival times of the two companies. You can perform this step by using the CORR procedure as follows:

```
proc corr data = default pearson kendall;
   var time1 time2;
run;
```

Output 5.1.3 shows the output of this code. The output contains some descriptive statistics and two measures of correlation: Pearson and Kendall. Both measures indicate high and statistically significant dependence between the life spans of the two companies.

Output 5.1.3 Default Time Descriptive Statistics and Correlations

The CORR Procedure

2 Variables:	time1	time2

Simple Statistics

Variable	N	Mean	Std Dev	Median	Minimum	Maximum
time1	1000000	2.00042	1.99724	1.38664	1.78961E-6	28.39277
time2	1000000	2.00190	2.00064	1.38787	2.24931E-6	30.50949

Pearson Correlation Coefficients, N = 1000000
Prob > |r| under H0: Rho=0

	time1	time2
time1	1.00000	0.76950 <.0001
time2	0.76950 <.0001	1.00000

Kendall Tau b Correlation Coefficients, N = 1000000
Prob > |tau| under H0: Tau=0

	time1	time2
time1	1.00000	0.58998 <.0001
time2	0.58998 <.0001	1.00000

The second and final step is to empirically estimate the default probabilities of the two companies. This is done by using the FREQ procedure as follows:

```
proc freq data=default;
   table survive survive1-survive2;
run;
```

The results are shown in Output 5.1.4.

Output 5.1.4 Probabilities of Default

The FREQ Procedure

survive	Frequency	Percent	Cumulative Frequency	Cumulative Percent
0	852314	85.23	852314	85.23
1	147686	14.77	1000000	100.00

survive1	Frequency	Percent	Cumulative Frequency	Cumulative Percent
0	776565	77.66	776565	77.66
1	223435	22.34	1000000	100.00

survive2	Frequency	Percent	Cumulative Frequency	Cumulative Percent
0	776382	77.64	776382	77.64
1	223618	22.36	1000000	100.00

Output 5.1.4 shows that the empirical default probabilities are 78% and 78%. Assuming that these companies are independent yields the probability estimate that both companies default during the period of three years as 0.75*0.78=0.59 (61%). Comparing this naive estimate with the much higher actual 85% joint default probability illustrates that neglecting the correlation between the two companies significantly underestimates the probability of default.

References

Devroye, L. (1986), *Non-uniform Random Variate Generation*, New York: Springer-Verlag. URL http://luc.devroye.org/rnbookindex.html

Marshall, A. W. and Olkin, I. (1988), "Families of Multivariate Distributions," *Journal of the American Statistical Association*, 83, 834–841.

Nolan, J. P. (2010), *Stable Distributions: Models for Heavy Tailed Data*, Boston: Birkhäuser.

Sklar, A. (1959), "Fonctions de répartition à *n* dimensions et leurs marges," *Publications de l'Institut de Statistique de L'Université de Paris*, 8, 229–231.

Subject Index

Syntax Index