SAS/ETS® 14.3 User's Guide

The COPULA Procedure

This document is an individual chapter from *SAS/ETS® 14.3 User's Guide*.

The correct bibliographic citation for this manual is as follows: SAS Institute Inc. 2017. *SAS/ETS® 14.3 User's Guide*. Cary, NC: SAS Institute Inc.

SAS/ETS® 14.3 User's Guide

SAS Institute Inc., SAS Campus Drive, Cary, NC 27513-2414

September 2017

Chapter 10
The COPULA Procedure

Contents

Overview: COPULA Procedure

A multivariate distribution for a random vector contains a description of both the marginal distributions and their dependence structure. A copula approach to formulating a multivariate distribution provides a way to isolate the description of the dependence structure from the marginal distributions. A copula is a function that combines marginal distributions of variables into a specific multivariate distribution. All of the one-dimensional marginals in the multivariate distribution are the cumulative distribution functions of the factors. Copulas help perform large-scale multivariate simulation from separate models, each of which can be fitted using different, even nonnormal, distributional specifications.

The COPULA procedure enables you to fit multivariate distributions or copulas from a given sample data set. You can do the following:

- estimate the parameters for a specified copula type
- simulate a given copula
- plot dependent relationships among the variables

The following types of copulas are supported:

- normal copula
- *t* copula
- Clayton copula
- Gumbel copula
- Frank copula

Getting Started: COPULA Procedure

The following example illustrates the use of PROC COPULA. The data used are daily returns on several major stocks. The main purpose of this example is to estimate the joint distribution of stock returns and then simulate from this distribution a new sample of specified size.

Figure 10.1 shows the first 10 observations of the daily stock return data set.

Figure 10.1 First 10 Observations of Daily Returns

Obs	date	ret_msft	ret_ko	ret_ibm	ret_duk	ret_bp
1	01/03/2008	0.004182	0.010367	0.002002	0.003503	0.019114
2	01/04/2008	-0.027960	0.001913	-0.035861	-0.000582	-0.014536
3	01/07/2008	0.006732	0.023607	-0.010671	0.025611	0.017922
4	01/08/2008	-0.033435	0.004239	-0.024610	-0.002838	-0.016049
5	01/09/2008	0.029560	0.026680	0.007301	0.010814	-0.027078
6	01/10/2008	-0.003054	0.004441	0.016414	-0.001689	-0.004395
7	01/11/2008	-0.012255	-0.027346	-0.022546	-0.012408	-0.018473
8	01/14/2008	0.013958	0.008418	0.053857	0.003427	0.001166
9	01/15/2008	-0.011318	-0.010851	-0.010689	-0.017075	-0.040925
10	01/16/2008	-0.022587	-0.015021	-0.001955	0.002316	-0.021336

The following statements fit a normal copula to the returns data (with the FIT statement) and create a new SAS data set that contains parameter estimates of the model. The VAR statement specifies the list of variables, which in this case are the daily returns of five large company stocks.

```
/* Copula estimation */
proc copula data = returns;
   var ret_ibm ret_msft ret_bp ret_ko ret_duk;
   fit normal / outcopula=estimates;
run;
```

The first table in Figure 10.2 shows some general information about the copula fitting procedure: the number of observations, the name of the input data set, the type of model, and the correlation matrix.

Figure 10.2 Copula Estimation: Fit Summary and Correlation Matrix

The COPULA Procedure

Model Fit Summary	
Number of Observations	603
Data Set	WORK.RETURNS
Copula Type	Normal

Correlation Matrix

	ret_ibm	ret_msft	ret_bp	ret_ko	ret_duk
ret_ibm	1.0000	0.6232	0.5294	0.4725	0.4902
ret_msft	0.6232	1.0000	0.5229	0.5015	0.4567
ret_bp	0.5294	0.5229	1.0000	0.3980	0.4378
ret_ko	0.4725	0.5015	0.3980	1.0000	0.5283
ret_duk	0.4902	0.4567	0.4378	0.5283	1.0000

Next, the following statements restrict the data set to only those columns that contain correlation parameter estimates:

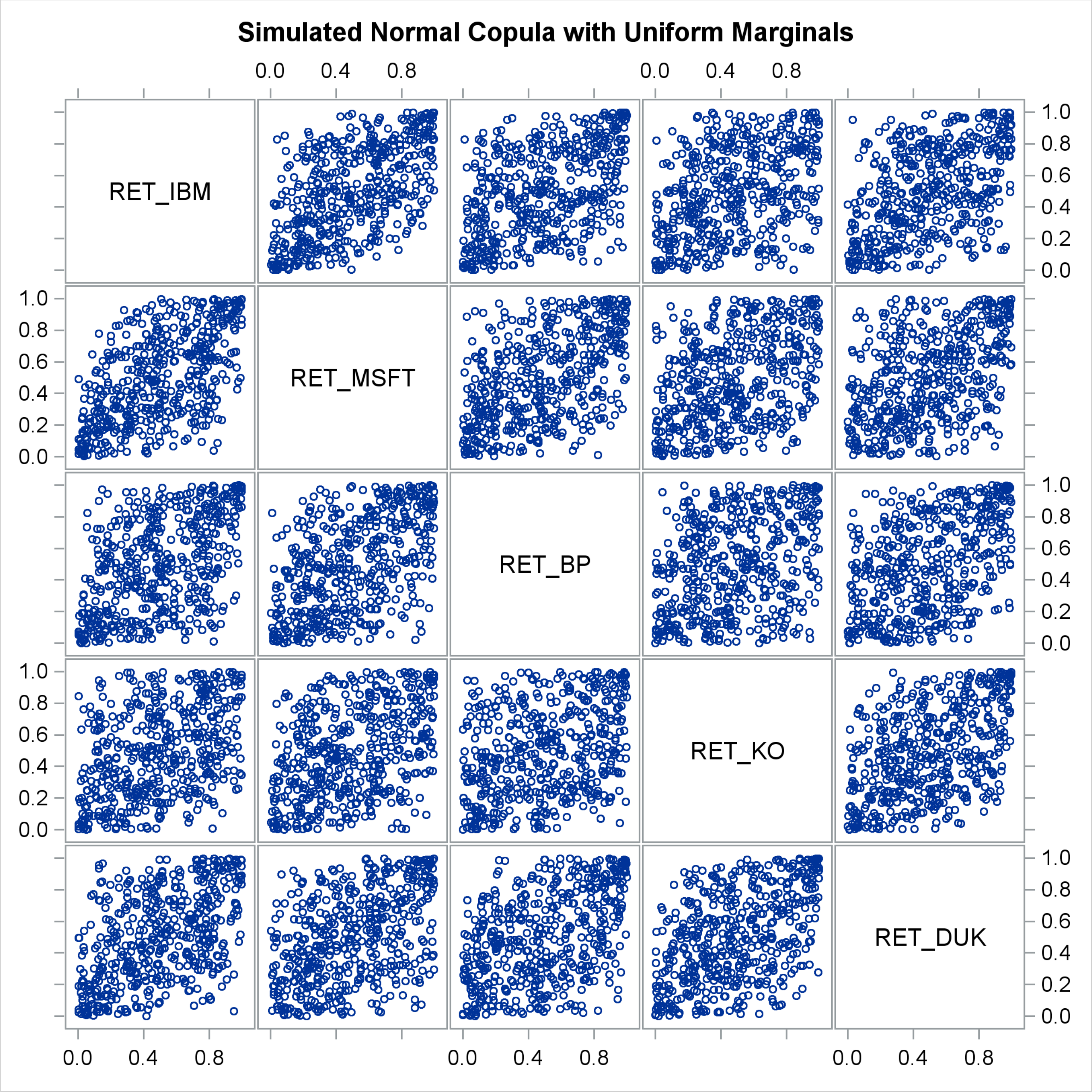
```
/* keep only correlation estimates */
data estimates;
   set estimates;
   keep ret_ibm ret_msft ret_bp ret_ko ret_duk;
run;
```

Then, 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=500 option in the SIMULATE statement generates 500 observations from the normal copula. The OUTUNIFORM= option specifies the name of SAS data set to contain the simulated sample with uniform marginal distributions. Note that this syntax does not require the DATA= option.

```
/* Copula simulation of uniforms */
proc copula;
   var ret_ibm ret_msft ret_bp ret_ko ret_duk;
   define cop normal (corr = estimates);
   simulate cop / ndraws     = 500
                  seed       = 1234
                  outuniform = simulated_uniforms
                  plots=(datatype=uniform);
run;
```

The simulated data are contained in the new SAS data set, Simulated_Uniforms. A scatter plot matrix of uniform marginals contained in the data set is shown in Output 10.3.

Figure 10.3 Simulated Data, Uniform Marginals



The preceding sequence of PROC COPULA usage—first fit, then simulate given estimated parameters—is a legitimate sequence but has a limitation in that the second COPULA call does not generate the sample according to the empirical distribution of the raw data. It generates only marginally uniform series.

In the following statements, the FIT statement fits a *t* copula to the returns data and at the same time simulates the sample according to empirical marginal distributions:

```
/* Copula estimation and simulation of returns */
proc copula data = returns;
   var ret_ibm ret_msft ret_bp ret_ko ret_duk;
   fit T;
   simulate / ndraws = 1000
              seed   = 1234
              out    = simulated_returns;
run;
```

The output of the statements is similar in structure to the output displayed in Figure 10.2 with the addition of parameter estimates and inference statistics that are specific to the copula model as shown in Figure 10.4. For a *t* copula, the degrees of freedom are displayed (as in Figure 10.4); for Archimedean copulas, the parameter “theta” is displayed; and for a normal copula, this table is not printed.

Figure 10.4 Copula Estimation: Specific Parameter Estimates

The COPULA Procedure

Parameter Estimates

Parameter	Estimate	Standard Error	t Value	Approx Pr > |t|
DF	3.659299	0.320810	11.41	<.0001

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

Syntax: COPULA Procedure

The COPULA procedure is controlled by the following statements:

PROC COPULA < **DATA=***SAS-data-set*> ;
VAR *variables* ;
DEFINE *name copula-type* < (*parameter-value-options . . .*)> ;
FIT *type* < **NAME=***name* > < **INIT=***(parameter-value-options)* > / *options* ;
BOUNDS *bound1* < , *bound2 . . .* > ;
SIMULATE < *copula-name-list* > / *options* ;
BY *variables* ;

Functional Summary

Table 10.1 summarizes the statements and options used with the COPULA procedure.

Table 10.1 PROC COPULA Functional Summary

Description	Statement	Option
Data Set Options		
Specifies the input data set	COPULA	DATA=
Specifies the input data set that contains the correlation matrix for elliptical copulas	DEFINE	CORR=
Specifies the input data set that contains the correlation matrix defined in Kendall's tau for elliptical copulas	DEFINE	KENDALL=
Specifies the input data set that contains the correlation matrix defined in Spearman's rho for elliptical copulas	DEFINE	SPEARMAN=
Specifies the degrees of freedom for *t* copulas	DEFINE	DF=
Specifies the parameter value for Archimedean copulas	DEFINE	THETA=
Specifies the hierarchy for hierarchical Archimedean copulas	DEFINE	HIERARCHY=
Declaring the Role of Variables		
Specifies the names of the variables to use in copula fitting or in simulation	VAR	
Specifies BY-group processing	BY	
Plotting Options		
Prints a summary iteration listing	FIT	ITPRINT
Suppresses the normal printed output	FIT	NOPRINT
Requests all printing options	FIT	PRINTALL
Suppresses the correlation matrix printed output	FIT	NOCORR
Printing Control Options		
Displays plots for fitted copulas	FIT	PLOTS=
Displays plots for simulated copulas	SIMULATE	PLOTS=
Optimization Process Control Options		
Sets boundary restrictions on parameters	BOUNDS	
Selects the iterative minimization method to use	FIT	METHOD=
Sets initial values for parameters	FIT	INIT=
Copula Estimation Options		
Specifies the marginal distribution of the individual variables	FIT	MARGINALS=

Table 10.1 *continued*

Description	Statement	Option
Copula Simulation Options		
Specifies the marginal distribution of the simulated variables	SIMULATE	MARGINALS=
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 fitted copula values	FIT	OUTCOPULA=
Specifies the output data set to contain pseudo-samples with the uniform marginal distribution	FIT	OUTPSEUDO=
Specifies the output data set to contain the random samples from the simulation	SIMULATE	OUT=
Specifies the output data set to contain the random samples from the simulation with uniform marginal distribution	SIMULATE	OUTUNIFORM=

PROC COPULA Statement

PROC COPULA < **DATA=***SAS-data-set*> **;**

The PROC COPULA statement has the following option:

DATA= < *libref.* >*SAS-data-set*

specifies the input data set used to estimate parameters for the FIT statement. When the procedure is used for simulation only, the input data set is not required to run the procedure. If you do not specify *libref*, then the Work library is used. Work is the default temporary library that is automatically defined by SAS at the beginning of each SAS session or job.

BOUNDS Statement

BOUNDS *bound1* < *, bound2 . . .* > **;**

The BOUNDS statement specifies the lower and upper bounds for the parameters. You can use this statement only when maximum likelihood estimation is used for the specified copula. Each bound is composed of parameters, constants, and inequality operators in the following format:

item operator item < *operator item operator item . . .* >

Each *item* is a constant, parameter, or list of parameters. Parameters associated with a regressor variable are referred to by the name of the corresponding regressor variable. Each *operator* is <, >, <=, or >=. The following example indicates that the lower and upper bounds for the parameter THETA are –5 and 10, respectively:

```
bounds -5 < THETA < 10;
```

If you do not specify bounds, the internal default values are used; the default values are described in the section "Details: COPULA Procedure" on page 530. For the normal and *t* copulas, the correlation matrix uses only the default parameter bounds, which are –1 and 1 for lower bound and upper bound, respectively.

BY Statement

BY *variables* **;**

The BY statement specifies groups in which separate FIT analyses for copula are performed. The *variables* must be present in the input data set and are excluded from the model fitting. The BY statement requires the VAR statement to be present.

DEFINE Statement

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

FIT *type* < **NAME=***name* >< **INIT=(***parameter-value-options* >**)** */options* **;**

The DEFINE statement specifies the relevant information of copula used for the simulation.

name specifies the name of the copula definition, which can be used later in the SIMULATE statement.

copula-type specifies one of the following types of copula:

CLAYTON specifies the Clayton copula.

FRANK specifies the Frank copula.

GUMBEL specifies the Gumbel copula.

HACCLAYTON specifies the hierarchical Clayton copula.

HACFRANK specifies the hierarchical Frank copula.

HACGUMBEL specifies the hierarchical Gumbel copula.

NORMAL specifies the normal copula.

T specifies the *t* copula.

These copula models are also described in the section "Details: COPULA Procedure" on page 530.

parameter-value-options

specify the input parameters used to simulate the specified copula. These options must be appropriate for the type of copula specified. You can specify the following 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. For an example of a correlation matrix input in this form, see Output 10.2.1. 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.

HIERARCHY=(*name=***(***HAC-specification* **)(THETA***=value* **))** (Experimental)

specifies the hierarchy for hierarchical Archimedean copulas. The argument usually consists of multiple specification lines, where each line specifies one copula in the hierarchy. The *name* can be user-defined symbols, with the exception of the copula at the top of the hierarchy, which must be named ROOT. The *HAC-specification* is a list of symbols that can be either defined copula names or variable names from the VAR statement, depending on whether the element of the copula is a variable or an inner copula in the hierarchy. For example, you can use the following code to define a hierarchical Archimedean copula, with the hierarchy shown in Figure 10.5:

```
var u1-u4;
define cop hacclayton hierarchy=(
root = (c1 c2)(theta=1)
c1 = (u1 u2)(theta=3)
c2 = (u3 u4)(theta=5));
```

Note that as long as the specification is valid, the order of the specification lines does not matter. In the previous example, you could first list $c1$ and $c2$, and then define $root$.

KENDALL=*SAS-data-set*

specifies the data set that contains the correlation matrix defined in Kendall's tau. 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. 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.

SPEARMAN=*SAS-data-set*

specifies the data set that contains the correlation matrix defined in Spearman's rho. 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. 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 copulas.

THETA=*value*

specifies the parameter value for Archimedean copulas.

The DEFINE statement is used with the SIMULATE statement. The FIT statement can also be used with the SIMULATE statement. The results of the FIT statement can be the input of the SIMULATE statement. Therefore, the SIMULATE statement can follow the FIT statement. If there is no FIT statement, then the DEFINE statement must precede the SIMULATE statement. However, you cannot use both the FIT and DEFINE statements in the same procedure.

FIT Statement

FIT *type* < **NAME=***name* >< **INIT=(***parameter-value-options* >**)** */options* **;**

The FIT statement estimates the parameters for a specified copula type.

type

specifies the type of the copula to be estimated, which is one of the following:

CLAYTON	fits the Clayton copula.
FRANK	fits the Frank copula.
GUMBEL	fits the Gumbel copula.
NORMAL	fits the normal copula.
T	fits the *t* copula.

INIT=(*parameter-value-options***)**

provides the initial values for the numerical optimization.

[*parameter-value-options*]

specify the input parameters that are used to initialize the specified copula. These options must be appropriate for the type of copula that you specify. You can specify the following 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. For an example of a correlation matrix input in this form, see Output 10.2.1. 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.

KENDALL=*SAS-data-set*

specifies the data set that contains the correlation matrix defined in Kendall's tau. 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. 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.

SPEARMAN=*SAS-data-set*

specifies the data set that contains the correlation matrix defined in Spearman's rho. 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. 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 copulas.

THETA=*value*

specifies the parameter value for Archimedean copulas.

For Archimedean copulas, the default initial values of the parameter are computed using the calibration method. The default initial value for the degrees-of-freedom parameter in the *t* copula is set to 2.0. The following statement shows an initialization for Student's *t* copula, where the Kendall's tau correlation matrix is stored in the corrmat data set and the DF is set to 2.5:

```
fit t init=(df=2.5 kandall=corrmat);
```

NAME=*name*

specifies an identifier for the fit, which is stored as an ID variable in the OUTCOPULA= data set.

You can specify the following *options* after a slash (/):

MARGINALS=UNIFORM | EMPIRICAL

specifies the marginal distribution of the individual variables. You can specify the following values:

EMPIRICAL uses the marginal empirical CDF to transform the data and uses the transformed data to fit the copula.

UNIFORM uses the input data without transformation to fit the copula.

METHOD=MLE | CAL

specifies the method used to estimate parameters. You can specify the following values:

CAL specifies the calibration method that uses the correlation matrix (only Kendall's tau is implemented in this procedure).

MLE represents canonical maximum likelihood estimation (CMLE) or maximum likelihood estimation (MLE).

For the *t* copula, if METHOD=CAL, then the correlation matrix is estimated using the calibration method with Kendall's tau and the degrees of freedom are estimated by the MLE. For the normal copula, only METHOD=MLE is supported and METHOD=CAL is ignored. By default for all copula types, METHOD=MLE.

OUTCOPULA=*SAS-data-set*

specifies the name of the output data set. Each fitted copula is written to the OUTCOPULA= data set. The data set is not created if this option is not specified.

OUTPSEUDO=*SAS-data-set*

specifies the output data set for saving the pseudo-samples with uniform marginal distributions. The pseudo-samples are obtained by transforming the individual variables of the original data with the empirical cumulative distribution functions (CDFs). The data set is not created if this option is not specified.

PLOTS<**(***global-plot-options***)**> **< =** *(specific-plot-options)***>**

controls the plots that are produced by the COPULA procedure. By default, PROC COPULA produces a scatter plot matrix for variables (that is, it displays a symmetric matrix plot with the variables that are specified in the VAR statement).

You can specify the following *global-plot-options*:

NVAR=ALL | *n*

specifies the maximum number of variables specified in the VAR statement to be displayed in the matrix plot. The NVAR=ALL option uses all variables that are specified in the VAR statement. By default, NVAR=5.

TAIL | CHI

requests that tail dependence plots (chi-plots) be plotted. If you specify this option with the UNPACK option on, PROC COPULA displays a chi-plot for each applicable pair of distinct variables that are specified in the VAR statement. If you specify this option without the UNPACK option, PROC COPULA displays a scatter plot matrix, the lower triangular section shows regular scatter plots between distinct pairs of variables that are specified in the VAR statement, the upper triangular section shows chi-plots for corresponding pairs of variables.

UNPACKPANEL | UNPACK

requests scatter plots for pairs of variables. If you specify this option, PROC COPULA displays a scatter plot for each applicable pair of distinct variables that are specified in the VAR statement.

You can specify the following *specific-plot-options*:

DATATYPE=ORIGINAL | UNIFORM | BOTH

requests the data type to be plotted. DATATYPE=ORIGINAL presents the data in their original marginal distribution; DATATYPE=UNIFORM shows the transformed data with uniform marginal distribution; and DATATYPE=BOTH plots both the original and uniform data types. If MARGINALS=UNIFORM, then the transformation is omitted and the DATATYPE= option is ignored.

NONE

suppresses all plots.

Printing Options

ITPRINT

prints a summary iteration listing.

NOCORR

suppresses the correlation matrix.

NOPRINT

suppresses all output.

PRINTALL

default option.

SIMULATE Statement

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

The SIMULATE statement simulates data from a specified copula model. The copula name specification can be either the name of a defined copula as specified by *name* in the DEFINE statement or the name of a fitted copula specified in the NAME= option in the FIT statement copula specification.

MARGINALS=UNIFORM | EMPIRICAL

specifies how the marginal distributions are computed. If MARGINALS=UNIFORM, then the samples are drawn from the copula distribution and marginal distributions are uniform.

MARGINALS=EMPIRICAL can be used to explicitly specify that the marginal distributions are empirical CDF computed from the DATA= option in the PROC COPULA statement.

If the MARGINALS= option is not specified in the SIMULATE statement, then the marginal distributions used in the simulation depend on whether a preceding FIT statement was used: If there is no FIT statement, the marginal distributions depend on whether the PROC COPULA statement includes a DATA= option. If there is a preceding FIT statement, then the marginal distributions from that fit are used. If there is no FIT statement and there is no DATA= option, then MARGINALS=UNIFORM.

NDRAWS=*integer*

specifies the number of draws to generate for this simulation. The default is 100.

OUT=*SAS-data-set*

specifies the output data set for the random samples from the simulation. This data set is the SAS data set in the OUTUNIFORM= option transformed by the inverse empirical CDF. This option is useful only when an input data exists and MARGINALS=EMPIRICAL. The data set is not created if this option is not specified.

OUTUNIFORM=*SAS-data-set*

specifies the output data set for the result of the simulation in uniforms. This option can be used when MARGINALS=UNIFORM or when MARGINALS=EMPIRICAL. If MARGINALS=EMPIRICAL, then this option enables you to obtain the samples simulated from the joint distribution specified by the copula, with all marginal distributions being uniform. The data are not created if this option is not specified.

PLOTS<**(***global-plot-options***)**> **<** **=** *(specific-plot-options)***>**

controls the plots that are produced by the COPULA procedure. By default, the PROC COPULA produces a scatter plot matrix for variables. You can specify any of the following *global-plot-options*:

NVAR=ALL | *n*

specifies the maximum number of variables specified in the VAR statement to be displayed in the matrix plot. The NVAR=ALL option uses all variables that are specified in the VAR statement. By default, NVAR=5.

TAIL | CHI

requests that tail dependence plots (chi-plots) be plotted. If you specify this option with the UNPACK option on, PROC COPULA displays a chi-plot for each applicable pair of distinct variables that are specified in the VAR statement. If you specify this option without the UNPACK option, PROC COPULA displays a scatter plot matrix, the lower triangular section shows regular scatter plots between distinct pairs of variables that are specified in the VAR statement, the upper triangular section shows chi-plots for corresponding pairs of variables.

UNPACKPANEL | UNPACK

requests scatter plots for pairs of variables. If you specify this option, PROC COPULA displays a scatter plot for each applicable pair of distinct variables that are specified in the VAR statement.

You can specify the following *specific-plot-options*:

DATATYPE=ORIGINAL | UNIFORM | BOTH

requests the data type to be plotted. DATA=ORIGINAL presents the data in their original marginal distribution; DATA=UNIFORM shows the transformed data with uniform marginal distribution; and DATA=BOTH plots both the original and uniform data types. If MARGINALS=UNIFORM, then the transformation is omitted and the DATA= option is ignored. If there are no input data, then the simulated data can only have uniform marginal distributions; in this case, the DATA= option is ignored.

DISTRIBUTION=PDF | CDF

requests distributional graphs for the case of two variables. DISTRIBUTION=PDF specifies that the theoretical probability density function is provided with both a contour plot and a surface plot. DISTRIBUTION=CDF requests the graph for the theoretical cumulative distribution function of the copula.

NONE

suppresses all plots.

SEED=*integer*

specifies the seed for generating random numbers for the simulation. If the seed is not provided, a random number is used as the seed.

VAR Statement

VAR *variables* **;**

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

Details: COPULA 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 $[0, 1]^m$ with all marginal distributions being standard uniform. The common notation for a copula is $C(u_1, \dots, u_m)$.

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

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

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

  for all $x_1, \dots, x_m$ in $[-\infty, \infty]$. Moreover, if the margins are continuous, then C is unique; otherwise C is uniquely determined on $\text{Ran}F_1 \times \cdots \times \text{Ran}F_m$, where $\text{Ran}F_j = F_j([-\infty, \infty])$ is the range of F_j.

- The converse is also true. That is, if C is a copula and $F_1, \dots, F_m$ are univariate distribution functions, then the multivariate function defined in the preceding equation is a joint distribution function with marginal distributions $F_j, j = 1, \dots, m$.

Dependence Measures

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

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

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

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

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

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

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

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, the rank correlations have their limitations. In particular, there are different copulas that result in the same rank correlation.

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

- The upper tail dependence, denoted λ_u, is

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

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

- The lower tail dependence is defined symmetrically.

Tail dependence is hard to detect by looking at a scatter plot of realizations of two random variables. One graphical way to detect tail dependence between two variables is by creating the chi plot of those two variables. The chi plot, as defined in Fisher and Switzer (2001), has characteristic patterns that depend on the dependence structure between the variables. The chi plot for the random variables X and Y is a scatter plot of the pairs (λ_i, χ_i) for each data point (x_i, y_i). λ_i is a measure of the distance of the data point (x_i, y_i) from the center of the data as measured by the median values of (x_i, y_i), and χ_i is a correlation coefficient between dichotomized values of X and Y. A positive λ_i means that x_i and y_i are either both large with respect to their median values or both small. A negative λ_i means that x_i or y_i is large with respect to its median, whereas the other value is small. Signs of tail dependence manifest as clusters of points that are significantly far from the χ axis around λ values of ± 1. If X and Y are uncorrelated, the χ values cluster around the λ axis.

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 with $m(m-1)/2$ parameters satisfying the positive semidefiniteness constraint. The normal copula can be written as

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

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 $L\tilde{\boldsymbol{Z}} \sim N(0, \Sigma)$.

Fitting

To fit a normal copula is to estimate the covariance matrix Σ from an input sample data set. Given a random sample $\boldsymbol{u}_i = (u_{i,1}, \ldots, u_{i,m})^\top$ where $i = 1, \ldots, n$, the log-likelihood function is

$$\begin{aligned} &\log L(\Sigma; \boldsymbol{u}_1, \ldots, \boldsymbol{u}_n) \\ &= \sum_{t=1}^{n} \log f_\Sigma(\Phi^{-1}(u_{t,1}), \ldots, \Phi^{-1}(u_{t,m})) - \sum_{t=1}^{n} \sum_{j=1}^{m} \log \phi(\Phi^{-1}(u_{t,j})) \end{aligned}$$

Here f_Σ is the joint density of the multivariate normal with mean zero and variance Σ, and ϕ is the univariate density of the standard normal distribution. Note that the second term is not related to the parameters Σ and, therefore, can be ignored during the optimization. The restriction that Σ is a correlation matrix is very inconvenient, and it is common practice to circumvent this problem by first assuming that Σ has the covariance form. Therefore, Σ can be estimated by

$$\widehat{\Sigma} = \frac{1}{n} \sum_{i=1}^{n} \zeta_i \zeta_i^T$$

where

$$\zeta_i = \left(\Phi^{-1}(u_{i,1}), \Phi^{-1}(u_{i,2}), \ldots, \Phi^{-1}(u_{i,m})\right)^T$$

This estimate is consistent with the form of a covariance matrix but not necessarily with the form of a correlation matrix. The approximation to the original MLE problem can be obtained using the normalizing operator defined as follows:

$$\begin{aligned} \Delta(\Sigma) &= \mathrm{diag}(\sigma_{11}^{1/2}, \ldots, \sigma_{mm}^{1/2}) \\ \mathcal{P}(\Sigma) &= (\Delta(\Sigma))^{-1} \Sigma (\Delta(\Sigma))^{-1} \end{aligned}$$

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 with 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 two steps:

1. Generate a multivariate vector $\boldsymbol{X} \sim t_m(\nu, 0, \Sigma)$ following 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 $s \sim \chi^2_\nu$.

Fitting

To fit a t copula is to estimate the covariance matrix Σ and degrees of freedom ν from a given multivariate data set. Given a random sample $\boldsymbol{u}_i = (u_{i,1}, \ldots, u_{i,m})^\top$, $i = 1, \ldots, n$ that has uniform marginal distributions, the log likelihood is

$$\log L(\nu, \Sigma; u_{i,1}, \ldots, u_{i,m})$$
$$= \sum_{i=1}^{n} \log g_{\nu,\Sigma}(t_\nu^{-1}(u_{i,1}), \ldots, t_\nu^{-1}(u_{i,m})) - \sum_{i=1}^{n}\sum_{j=1}^{m} \log g_\nu(t_\nu^{-1}(u_{i,j}))$$

where ν denotes the degrees of freedom of the t copula, $g_{\nu,\Sigma}$ denotes the joint density function of the centered multivariate t distribution with parameters (ν, Σ), t_ν is the distribution function of a univariate t distribution with ν degrees of freedom, Σ is a correlation matrix, and g_ν is the density function of univariate t distribution with ν degrees of freedom.

The log likelihood can be maximized with respect to the parameters $\theta = (\nu, \Sigma) \in \Theta$ using numerical optimization. If you allow the parameters in Σ to be such that Σ is symmetric and with ones on the diagonal, then the MLE estimate for Σ might not be positive semidefinite. In that case, you need to apply the adjustment to convert the estimated matrix to positive semidefinite, as shown by McNeil, Frey, and Embrechts (2005), Algorithm 5.55.

When the dimension of the data m increases, the numerical optimization quickly becomes infeasible. It is common practice to estimate the correlation matrix Σ by calibration using Kendall's tau. Then, using this fixed Σ, the single parameter ν can be estimated by MLE. By proposition 5.37 in McNeil, Frey, and Embrechts (2005),

$$\rho_\tau(U_i, U_j) = \frac{2}{\pi}\arcsin\rho_{ij}$$

where ρ_τ is the Kendall's tau and ρ_{ij} is the off-diagonal elements of the correlation matrix Σ of the t copula. Therefore, an estimate for the correlation is

$$\hat{\rho}_{ij} = \sin\left(\frac{1}{2}\pi\hat{\rho}^\tau_{i,j}\right)$$

where $\hat{\rho}$ and $\hat{\rho}^\tau$ are the estimates of the sample correlation matrix and Kendall's tau, respectively. However, it is possible that the estimate of the correlation matrix $\hat{\Sigma}$ is not positive definite. In this case, there is a standard procedure that uses the eigenvalue decomposition to transform the correlation matrix into one that is positive definite. Let Σ be a symmetric matrix with ones on the diagonal, with off-diagonal entries in $[-1, 1]$. If Σ is not positive semidefinite, use Algorithm 5.55 from McNeil, Frey, and Embrechts (2005):

1. Compute the eigenvalue decomposition $\Sigma = EDE^T$, where D is a diagonal matrix that contains all the eigenvalues and E is an orthogonal matrix that contains the eigenvectors.
2. Construct a diagonal matrix $\tilde{D}$ by replacing all negative eigenvalues in D by a small value $\delta > 0$.
3. Compute $\tilde{\Sigma} = E\tilde{D}E^T$, which is positive definite but not necessarily a correlation matrix.
4. Apply the normalizing operator $\mathcal{P}$ on the matrix $\tilde{\Sigma}$ to obtain the correlation matrix desired.

The log-likelihood function and its gradient function for a single observation are listed as follows, where $\boldsymbol{\zeta} = (\zeta_1, \ldots, \zeta_m)$, with $\zeta_j = t_\nu^{-1}(u_j)$, and g is the derivative of the $\log\Gamma$ function:

$$
\begin{aligned}
l = \log(c) &= -\frac{1}{2}\log(|\Sigma|) + \log\Gamma\left(\frac{\nu+m}{2}\right) + (m-1)\log\Gamma\left(\frac{\nu}{2}\right) - m\log\Gamma\left(\frac{\nu+1}{2}\right) \\
&\quad - \frac{\nu+m}{2}\log(1+\zeta^T\Sigma^{-1}\zeta/\nu) + \frac{\nu+1}{2}\sum_{j=1}^{m}\log\left(1+\frac{\zeta_j^2}{\nu}\right) \\
\frac{\partial l}{\partial \nu} &= \frac{1}{2}g\left(\frac{\nu+m}{2}\right) + \frac{m-1}{2}g\left(\frac{\nu}{2}\right) - \frac{m}{2}g\left(\frac{\nu+1}{2}\right) \\
&\quad - \frac{1}{2}\log(1+\zeta^T\Sigma^{-1}\zeta/\nu) + \frac{\nu+m}{2\nu^2}\frac{\zeta^T\Sigma^{-1}\zeta}{1+\zeta^T\Sigma^{-1}\zeta/\nu} \\
&\quad + \frac{1}{2}\sum_{j=1}^{m}\log(1+\zeta_j^2/\nu) - \frac{\nu+1}{2\nu^2}\sum_{j=1}^{m}\frac{\zeta_j^2}{1+\zeta_j^2/\nu} \\
&\quad - \frac{(\nu+m)}{\nu}\frac{\zeta^T\Sigma^{-1}(d\zeta/d\nu)}{1+\zeta^T\Sigma^{-1}\zeta/\nu} + \frac{\nu+1}{\nu}\sum\frac{\zeta_j(d\zeta_j/d\nu)}{1+\zeta_j^2/\nu}
\end{aligned}
$$

The derivative of the likelihood with respect to the correlation matrix Σ follows:

$$
\begin{aligned}
\frac{\partial l}{\partial \Sigma} &= -\frac{1}{2}(\Sigma^{-1})^T + \frac{\nu+m}{2}\frac{\Sigma^{-T}\zeta\zeta^T\Sigma^{-T}/\nu}{1+\zeta^T\Sigma^{-1}\zeta/\nu} \\
&= -\frac{1}{2}(\Sigma^{-1})^T + \frac{\nu+m}{2}\frac{\Sigma^{-T}\zeta\zeta^T\Sigma^{-T}}{\nu+\zeta^T\Sigma^{-1}\zeta}
\end{aligned}
$$

Archimedean Copulas

Overview of Archimedean Copulas

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

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

The Archimedean copulas available in the COPULA 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

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

with $\theta > 0$.

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

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

with $\theta \in (-\infty, \infty)\backslash\{0\}$ 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

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

with $\theta > 1$.

Simulation

Suppose the generator of the Archimedean copula is ϕ. Then the simulation method using the 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 with the distribution function F such that $\tilde{F}(t) = \phi^{-1}(t)$.
2. Draw samples from 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 with shape parameter θ^{-1} and scale parameter one.
- For the Gumbel copula, $\tilde{F} = \exp(-t^{1/\theta})$, and F is the distribution function of the stable variable $\mathrm{St}(\theta^{-1}, 1, \gamma, 0)$ with $\gamma = [\cos(\pi/(2\theta))]^\theta$.
- For the Frank copula with $\theta > 0$, $\tilde{F} = -\log\{1 - \exp(-t)[1 - \exp(-\theta)]\}/\theta$, 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 with parameter value $1 - e^{-\theta}$.

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 with $m = 2$ 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 $u_2 = -\frac{1}{\theta} \log\left(1 + \frac{v_2(1-e^{-\theta})}{v_2(e^{-\theta v_1}-1)-e^{-\theta v_1}}\right)$.

Fitting

One method to estimate the parameters is to calibrate with Kendall's tau. The relation between the parameter θ and Kendall's tau is summarized in Table 10.5 for the three Archimedean copulas.

Table 10.2 Calibration Using Kendall's Tau

Copula Type	τ	**Formula for** θ
Clayton	$\theta/(\theta+2)$	$2\tau/(1-\tau)$
Gumbel	$1 - 1/\theta$	$1/(1-\tau)$
Frank	$1 - 4\theta^{-1}(1 - D_1(\theta))$	No closed form

In Table 10.2, $D_1(\theta) = \theta^{-1} \int_0^\theta t/(\exp(t) - 1)dt$ for $\theta > 0$, and $D_1(\theta) = D_1(\theta) + 0.5\theta$ for $\theta < 0$. In addition, for the Frank copula, the formula for θ has no closed form. The numerical algorithm for root finding can be used to invert the function $\tau(\theta)$ to obtain θ as a function of τ.

Alternatively, you can use the MLE or the CMLE method to estimate the parameter θ given the data $\mathbf{u} = \{u_{i,j}\}$ and $i = 1, \ldots, n, j = 1, \ldots, m$. The log-likelihood function for each type of Archimedean copula is provided in the following sections.

Fitting the Clayton Copula

For the Clayton copula, the log-likelihood function is as follows (Cherubini, Luciano, and Vecchiato 2004, Chapter 7):

$$
\begin{aligned}
l = n\left[m\log(\theta) + \log\left(\Gamma\left(\frac{1}{\theta} + m\right)\right) - \log\left(\Gamma\left(\frac{1}{\theta}\right)\right)\right] - (\theta+1)\sum_{i,j}\log u_{ij} \\
-\left(\frac{1}{\theta} + m\right)\sum_i \log\left(\sum_j u_{ij}^{-\theta} - m + 1\right)
\end{aligned}
$$

Let $g(\cdot)$ be the derivative of $\log(\Gamma(\cdot))$. Then the first-order derivative is

$$
\begin{aligned}
\frac{dl}{d\theta} &= n\left[\frac{m}{\theta} + g\left(\frac{1}{\theta} + m\right)\frac{-1}{\theta^2} - g\left(\frac{1}{\theta}\right)\frac{-1}{\theta^2}\right] \\
&\quad - \sum_{i,j}\log(u_{ij}) + \frac{1}{\theta^2}\sum_i \log\left(\sum_j u_{ij}^{-\theta} - m + 1\right) \\
&\quad - \left(\frac{1}{\theta} + m\right)\sum_i \frac{-\sum_j u_{ij}^{-\theta}\log(u_{ij})}{\sum_j u_{ij}^{-\theta} - m + 1}
\end{aligned}
$$

The second-order derivative is

$$
\begin{aligned}
\frac{d^2 l}{d\theta^2} &= n\left\{\frac{-m}{\theta^2} + g'\left(\frac{1}{\theta} + m\right)\frac{1}{\theta^4} + g\left(\frac{1}{\theta} + m\right)\frac{2}{\theta^3} - g'\left(\frac{1}{\theta}\right)\frac{1}{\theta^4} - g\left(\frac{1}{\theta}\right)\frac{2}{\theta^3}\right\} \\
&\quad - \frac{2}{\theta^3}\sum_i \log\left(\sum_j u_{ij}^{-\theta} - m + 1\right) \\
&\quad + \frac{2}{\theta^2}\sum_i \frac{-\sum_j u_{ij}^{-\theta}\log u_{ij}}{\sum_j u_{ij}^{-\theta} - m + 1} \\
&\quad - \left(\frac{1}{\theta} + m\right)\sum_i \left\{\frac{\sum_j u_{ij}^{-\theta}(\log u_{ij})^2}{\sum_j u_{ij}^{-\theta} - m + 1} - \left(\frac{\sum_j u_{ij}^{-\theta}\log u_{ij}}{\sum_j u_{ij}^{-\theta} - m + 1}\right)^2\right\}
\end{aligned}
$$

Fitting the Gumbel Copula

A different parameterization $\alpha = \theta^{-1}$ is used for the following part, which is related to the fitting of the Gumbel copula. For the Gumbel copula, you need to compute $\phi^{-1(m)}$. It turns out that for $k = 1, 2, \ldots, m$,

$$\phi^{-1(k)}(u) = (-1)^k \alpha \exp(-u^\alpha) u^{-k+\alpha} \Psi_{k-1}(u^\alpha)$$

where Ψ_{k-1} is a function that is described later. The copula density is given by

$$\begin{aligned} c &= \phi^{-1(m)}(x) \prod_k \phi'(u_k) \\ &= (-1)^m \alpha \exp\left(-x^\alpha\right) x^{-k+\alpha} \Psi_{m-1}\left(x^\alpha\right) \prod_k \phi'(u_k) \\ &= (-1)^m f_1 f_2 f_3 f_4 f_5 \end{aligned}$$

where $x = \sum_k \phi(u_k)$, $f_1 = \alpha$, $f_2 = \exp(-x^\alpha), f_3 = x^{-k+\alpha}, f_4 = \Psi_{m-1}(x^\alpha)$, and $f_5 = (-1)^m \prod_k \phi'(u_k)$.

The log density is

$$\begin{aligned} l &= \log(c) \\ &= \log(f_1) + \log(f_2) + \log(f_3) + \log(f_4) + \log\left((-1)^m f_5\right) \end{aligned}$$

Now the first-order derivative of the log density has the decomposition

$$\frac{dl}{d\alpha} = \frac{1}{c}\frac{dc}{d\alpha} = \sum_{j=1}^{4} \frac{1}{f_j}\frac{df_j}{d\alpha} + \frac{d \sum_k \log\left(-\phi'(u_k)\right)}{d\alpha}$$

Some of the terms are given by

$$\begin{aligned} \frac{1}{f_1}\frac{df_1}{d\alpha} &= \frac{1}{\alpha} \\ \frac{1}{f_2}\frac{df_2}{d\alpha} &= -x^\alpha \log(x) - \alpha x^{\alpha-1}\frac{dx}{d\alpha} \\ \frac{1}{f_3}\frac{df_3}{d\alpha} &= \log(x) + (-k+\alpha) x^{-1}\frac{dx}{d\alpha} \end{aligned}$$

where

$$\frac{dx}{d\alpha} = \sum (-\log u_k)^{1/\alpha} \log(-\log u_k) \left(\frac{-1}{\alpha^2}\right)$$

The last term in the derivative of the $dl/d\alpha$ is

$$\begin{aligned}
\log\left(-\phi'(u_k)\right) &= \log\left(\frac{1}{\alpha}\left(-\log u_k\right)^{\frac{1}{\alpha}-1}\frac{1}{u_k}\right)\\
&= -\log\alpha - \log(u_k) + \left(\frac{1}{\alpha}-1\right)\log\left(-\log(u_k)\right)\\
\frac{d\sum_k \log\left(-\phi'(u_k)\right)}{d\alpha} &= \sum_{k=1}^{m} -\frac{1}{\alpha} - \frac{1}{\alpha^2}\log\left(-\log(u_k)\right)\\
&= -\frac{m}{\alpha} - \frac{1}{\alpha^2}\sum_{k=1}^{m}\log\left(-\log(u_k)\right)
\end{aligned}$$

Now the only remaining term is f_4, which is related to Ψ_{m-1}. Wu, Valdez, and Sherris (2007) show that $\Psi_k(x)$ satisfies a recursive equation

$$\Psi_k(x) = [\alpha(x-1)+k]\Psi_{k-1}(x) - \alpha x \Psi'_{k-1}(x)$$

with $\Psi_0(x) = 1$.

The preceding equation implies that $\Psi_{k-1}(x)$ is a polynomial of x and therefore can be represented as

$$\Psi_{k-1}(x) = \sum_{j=0}^{k-1} a_j(k-1,\alpha)\, x^j$$

In addition, its coefficient, denoted by $a_j(k-1,\alpha)$, is a polynomial of α. For simplicity, use the notation $a_j(\alpha) \equiv a_j(m-1,\alpha)$. Therefore,

$$f_4 = \Psi_{m-1}\left(x^\alpha\right) = \sum_{j=0}^{m-1} a_j(\alpha)\, x^{j\alpha}$$

$$\begin{aligned}
\frac{df_4}{d\alpha} &= \frac{d\Psi_{m-1}(x^\alpha)}{d\alpha}\\
&= \sum_{j=0}^{m-1}\left[\frac{da_j(\alpha)}{d\alpha}x^{j\alpha} + a_j(\alpha)\, x^{j\alpha}\log(x)\, j + a_j(\alpha)(j\alpha)\, x^{j\alpha-1}\frac{dx}{d\alpha}\right]
\end{aligned}$$

Fitting the Frank Copula

For the Frank copula,

$$\phi^{-1(k)}(u) = -\frac{1}{\theta}\Psi_{k-1}\left(\left(1+e^{-u}(e^{-\theta}-1)\right)^{-1}\right)$$

When $\theta > 0$, a Frank copula has a probability density function

$$\begin{aligned}
c &= \varphi^{-1(m)}(x)\prod_k \varphi'(u_k)\\
&= \frac{-1}{\theta}\Psi_{m-1}\left(\frac{1}{1+e^{-x}\left(e^{-\theta}-1\right)}\right)\prod_k \varphi'(u_k)
\end{aligned}$$

where $x = \sum_k \varphi\left(u_k\right)$.

The log likelihood is

$$\log c = -\log\left(\theta\right) + \log\left(\Psi_{m-1}\left(\frac{1}{1+e^{-x}\left(e^{-\theta}-1\right)}\right)\right) + \sum \log\left(\varphi'\left(u_k\right)\right)$$

Denote

$$y = \frac{1}{1+e^{-x}\left(e^{-\theta}-1\right)}$$

Then the derivative of the log likelihood is

$$\frac{d\log c}{d\theta} = -\frac{1}{\theta} + \frac{1}{\Psi_{m-1}\left(y\right)}\frac{d\Psi_{m-1}}{d\theta} + \sum_k \frac{1}{\varphi'\left(u_k\right)}\frac{d\varphi'\left(u_k\right)}{d\theta}$$

The term in the last summation is

$$\frac{1}{\varphi'\left(u_k\right)}\frac{d\varphi'\left(u_k\right)}{d\theta} = \frac{1}{\theta\left(1-e^{\theta u_k}\right)}\left[1-e^{\theta u_k}+\theta u e^{\theta u_k}\right]$$

The function Ψ_{m-1} satisfies a recursive relation

$$\Psi_k(x) = x(x-1)\Psi'_{k-1}(x)$$

with $\Psi_0(x) = x - 1$. Note that Ψ_{m-1} is a polynomial whose coefficients do not depend on θ; therefore,

$$\begin{aligned}
\frac{d\Psi_{m-1}}{d\theta} &= \frac{d\Psi_{m-1}}{dy}\frac{dy}{d\theta} \\
&= \frac{d\Psi_{m-1}}{dy}\left[\frac{dy}{d\theta} + \frac{dy}{dx}\frac{dx}{d\theta}\right] \\
&= \frac{d\Psi_{m-1}}{dy}\left[\frac{e^{-x}e^{-\theta}}{\left[1+e^{-x}\left(e^{-\theta}-1\right)\right]^2} + \frac{e^{-x}\left(e^{-\theta}-1\right)}{\left[1+e^{-x}\left(e^{-\theta}-1\right)\right]^2}\frac{dx}{d\theta}\right]
\end{aligned}$$

where

$$\begin{aligned}
\frac{dx}{d\theta} &= \sum_k \frac{d\varphi\left(u_k\right)}{d\theta} = \sum_k \left[-\frac{u_k e^{-\theta u_k}}{1-e^{-\theta u_k}} + \frac{e^{-\theta}}{1-e^{\theta}}\right] \\
&= \sum_k \left[-\frac{u_k}{e^{\theta u_k}-1} + \frac{1}{e^{\theta}-1}\right]
\end{aligned}$$

For the case of $m = 2$ and $\theta < 0$, the bivariate density is

$$\log c = \log(\theta(1-e^{-\theta})) - \theta(u_1+u_2) - \log((1-e^{-\theta}-(1-e^{-\theta u_1})(1-e^{-\theta u_2}))^2)$$

Hierarchical Archimedean Copula (HAC) (Experimental)

Adopting the notations of Savu and Trede (2010), let L denote the total level of hierarchies and let D denote the dimension of the HAC. There are n_l distinct copulas at each level $l, l = 1, \ldots, L$. These copulas are indexed by $(l, j), j = 1, \ldots, n_l$. At each level, there are also d_l variables, $0 \leq d_l \leq D$ and $\sum_l d_l = D$. In the first step, all the variables at the lowest level are grouped into n_1 subsets, each subset being an ordinary multivariate Archimedean copula

$$C_{1,j}(\boldsymbol{u}_{1,j}) = \phi_{1,j}^{-1}\left(\sum_{\boldsymbol{u}_{1,j}} \phi_{1,j}(\boldsymbol{u}_{1,j})\right), j = 1, \ldots, n_1$$

where $\phi_{1,j}$ is the generator of copula $C_{1,j}$, $\boldsymbol{u}_{1,j}$ denotes the variables that belong to copula $C_{1,j}$, and the sum $\sum_{u_{1,j}}$ is the sum over each variable in the subset $\boldsymbol{u}_{1,j}$. The copulas $C_{1,j}$ can be different Archimedean copulas for $j = 1, \ldots, n_1$. Then at the second level, the copulas $C_{1,j}$ that are derived in the first level are aggregated as if they are individual variables. Suppose there are n_2 copulas and d_2 variables,

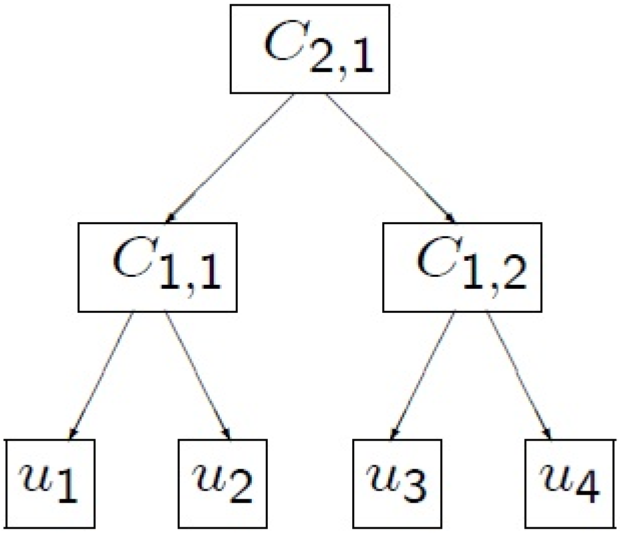
$$C_{2,j}(\boldsymbol{C}_{1,j}, \boldsymbol{u}_{2,j}) = \phi_{2,j}^{-1}\left(\sum_{\boldsymbol{C}_{1,j}} \phi_{2,j}(\boldsymbol{C}_{1,j}) + \sum_{\boldsymbol{u}_{2,j}} \phi_{2,j}(\boldsymbol{u}_{2,j})\right)$$

where $\phi_{2,j}$ denotes the generator of $C_{2,j}$ and $\boldsymbol{C}_{1,j}$ represents the subset of copulas in $C_{1,h}, h = 1, \ldots, n_1$, that is aggregated for copula $C_{2,j}$ for $j = 1, \ldots, n_2$. This structure continues until at level $l = L$ a single copula $C_{L,1}$ aggregates all the copulas at its previous level, $l = L - 1$.

A four-dimensional example that has total levels $L = 2$ and a structure shown in Figure 10.5 is defined as follows:

$$\begin{aligned} C_{2,1}(u_1, u_2, u_3, u_4) &= C_{2,1}\left(C_{1,1}(u_1, u_2), C_{1,2}(u_3, u_4)\right) \\ &= \phi_{2,1}^{-1}\left(\phi_{2,1} \circ \phi_{1,1}^{-1}\left(\phi_{1,1}(u_1) + \phi_{1,1}(u_2)\right) + \phi_{2,1} \circ \phi_{1,2}^{-1}\left(\phi_{1,2}(u_3) + \phi_{1,2}(u_4)\right)\right) \end{aligned}$$

Figure 10.5 Example Four-Dimensional Hierarchical Structure with Two Levels



Theorem 4.4 of McNeil (2008) states that the sufficient condition for a general hierarchical Archimedean structure to be a proper copula is that all appearing nodes of the form $\phi_{m,j} \circ \phi_{n,j}^{-1}$ have completely monotone derivatives. This condition places certain constraints on the copula parameters. In particular, if all the copulas in a hierarchical structure come from the Frank, Clayton, or Gumbel family, then $\theta_{m,j} \leq \theta_{n,j}$ for all j when $m < n$. Intuitively, this means that rank correlation must be increasing as you move down the hierarchical structure.

The hierarchical Archimedean copulas available in the COPULA procedure are the hierarchical versions of the Clayton, Frank, and Gumbel copulas.

Simulation

A slightly modified version of the recursive algorithm from McNeil (2008) works for all valid hierarchical structures that have Clayton, Frank, or Gumbel generators:

1. Start at $l = L$, and generate a random variable V with the distribution function F with Laplace transform $\phi_{L,1}^{-1}$.
2. For $l = L-1, \ldots, 1$, generate $u_{l,j}$ from its parent hierarchy. For $C_{l,j}$, recursively call this algorithm with the proper inner generators that correspond to the copula family.
3. Return $\boldsymbol{U} = (\phi_{L,1}^{-1}(-\log(u_1)/V), \ldots, \phi_{L,1}^{-1}(-\log(u_D)/V))^T$.

Let ϕ_1 be the outer generator and ϕ_2 the nested generator, and let θ_1 and θ_2 be the respective generator parameters. Let v be a draw from distribution function F with Laplace transform ϕ_1^{-1}. The inner copula generators $\phi_{12}(\cdot; v) = \exp(-v\phi_1 \circ \phi_2^{-1}(\cdot))$ and their corresponding Laplace transform distributions for the Clayton, Frank, and Gumbel family are summarized in Table 10.3.

Table 10.3 Inner Generators and Corresponding Distributions

Copula Type	$\phi_{12}(x; v)$	**Distribution with LT $\phi_{12}(\cdot; v)$**
Clayton	$\exp\left(v - v(1+x)^{\theta_1/\theta_2}\right)$	Tiled stable
Gumbel	$\exp(-vx^{\theta_1/\theta_2})$	Stable$\left(\frac{\theta_1}{\theta_2}, 1, \left(v\cos\frac{\theta_1\pi}{2\theta_2}\right)^{\theta_2/\theta_1}, 0\right)$
Frank	$\left(\frac{1}{1-e^{-\theta_1}}\left(1-\left(1-(1-e^{-\theta_2})\exp(-x)\right)^{\theta_1/\theta_2}\right)\right)^{v}$	No closed form

Note that when $\theta_1 = \theta_2$, the inner generators for the Clayton and Gumbel family both simplify to the generator of the independence copula, $\exp(-vx)$. For more information about simulating from the distribution with the Laplace transform given by the inner generator for the Frank family, see Hofert (2011). For more information about how to simulate from a tilted stable distribution, see McNeil (2008).

Canonical Maximum Likelihood Estimation (CMLE)

In the canonical maximum likelihood estimation (CMLE) method, it is assumed that the sample data $\boldsymbol{x}_i = (x_{i1}, x_{i2}, \ldots, x_{im})^\top$, $i = 1, \ldots, n$, have been transformed into uniform variates $\hat{\boldsymbol{u}}_i = (\hat{u}_{i1}, \ldots, \hat{u}_{im})$, $i = 1, \ldots, n$. One commonly used transformation is the nonparametric estimation of the CDF of the marginal distributions, which is closely related to empirical CDF,

$$\hat{u}_{i,j} = \hat{F}_{j,n}(x_{i,j})$$

where

$$\hat{F}_{j,n}(x) = \frac{1}{n+1}\sum_{i=1}^{n} I_{[x_{i,j} \le x]}$$

The transformed data $\hat{u}_{i,j}$ are used as if they had uniform marginal distributions; hence, they are called pseudo-samples. The function $\hat{F}_{j,n}$ is different from the standard empirical CDF in the scalar $1/(n+1)$, which is to ensure that the transformed data cannot be on the boundary of the unit interval $[0, 1]$. It is clear that

$$\hat{u}_{i,j} = \frac{1}{n+1}\operatorname{rank}(x_{i,j})$$

where $\operatorname{rank}(x_{i,j})$ is the rank among $i = 1, \ldots, n$ in increasing order.

Let $c(u_1, u_2, \ldots, u_m; \theta)$ be the density function of a copula $C(u_1, u_2, \ldots, u_m; \theta)$, and let θ be the parameter vector to be estimated. The parameter θ is estimated by maximum likelihood:

$$\hat{\theta} = \arg\max_{\theta\in\Theta}\sum_{i=1}^{n}\log c(\hat{u}_{i1}, \ldots, \hat{u}_{im}; \theta)$$

Exact Maximum Likelihood Estimation (MLE)

Suppose that the marginal distributions of vector elements $\boldsymbol{x}_i = (x_{i1}, x_{i2}, \ldots, x_{im})^\top$, $i = 1, \ldots, n$ are already known to be uniform. Then the parameter θ is estimated by exact maximum likelihood:

$$\hat{\theta} = \arg\max_{\theta \in \Theta} \sum_{i=1}^{n} \log c(x_{i1}, x_{i2}, \ldots, x_{im}; \theta)$$

Calibration Estimation

Instead of fitting the whole distribution as in MLE methods, you can directly use empirical estimates of distribution parameters. The unknown parameter that you want to estimate can be obtained by calibration using Kendall's tau. There exists a one-to-one map between the parameter at interest and Kendall's tau. Therefore, after you estimate the Kendall's tau, you can use the map to compute the parameter value. For example, the parameter matrix Σ in a t copula and the parameter θ in Archimedean copulas can be estimated in this manner. The most frequently used estimator of Kendall's tau is the rank correlation coefficient:

$$\hat{\rho}_\tau \left(X_i, X_j \right) = \left(\begin{array}{c} n \\ 2 \end{array} \right)^{-1} \sum_{1 \le t < s \le n} \text{sign} \left(\left(x_{t,i} - x_{s,i} \right) \left(x_{t,j} - x_{s,j} \right) \right)$$

The preceding formula is analogous to its population counterpart

$$\rho_\tau \left(X_i, X_j \right) = E \left[\text{sign} \left(\left(X_i - \tilde{X}_i \right) \left(X_j - \tilde{X}_j \right) \right) \right]$$

where $(\tilde{X}_i, \tilde{X}_j)$ has the same distribution but is independent of (X_i, X_j).

For Archimedean multivariate copulas there is only one parameter to estimate, τ (or its function θ), although for m variables there are $m(m-1)/2$ unique pairwise correlation coefficients. Denote the map from ρ_τ to θ by $\theta = \hat{\theta}(\rho_\tau)$. To aggregate the map, take simple arithmetic average:

$$\hat{\theta} = \frac{2}{m(m-1)} \sum_{1 \le i < j \le m} \hat{\theta} \left[\hat{\rho}_\tau \left(X_i, X_j \right) \right]$$

Nonlinear Optimization Options

PROC COPULA uses the nonlinear optimization (NLO) subsystem to perform nonlinear optimization tasks. In the PROC COPULA statement, you can specify nonlinear optimization options that are then passed to the NLO subsystem. For a list of all the nonlinear optimization options, see Chapter 6, "Nonlinear Optimization Methods."

Displayed Output

PROC COPULA produces displayed output described in the following sections.

Optimization Start and Resulting Parameter Estimates

If you specify the ITPRINT option in the PROC COPULA statement, PROC COPULA displays two tables, “Optimization Start Parameter Estimates” and “Optimization Results Parameter Estimates.” Each table contains the following information for each model parameter:

- parameter number
- parameter name
- parameter estimate
- gradient of the objective function at the initial parameter values

In addition to this information, the table “Optimization Start Parameter Estimates” contains the following columns:

- lower-bound constraint
- upper-bound constraint

The value of the objective function at the parameter values is displayed below each table.

Iteration History for Parameter Estimates

If you specify the ITPRINT option in the PROC COPULA statement, PROC COPULA displays a table that contains the following information for each iteration. Note that some information is specific to the model-fitting method chosen (for example, Newton-Raphson, trust region, or quasi-Newton method).

- iteration number
- number of restarts since the fitting began
- number of function calls
- number of active constraints at the current solution
- value of the objective function (–1 times the log-likelihood value) at the current solution
- change in the objective function from previous iteration
- value of the maximum absolute gradient element
- step size (for Newton-Raphson and quasi-Newton methods)
- slope of the current search direction (for Newton-Raphson and quasi-Newton methods)
- lambda (for trust region method)
- radius value at current iteration (for trust region method)

Model Fit Summary

The “Model Fit Summary” table contains the following information:

- number of observations used
- number of missing values in data set, if any
- data set name
- type of model that was fit
- log-likelihood value at solution
- maximum absolute gradient at solution
- number of iterations
- optimization method
- value of Akaike’s information criterion (AIC) at the solution (a smaller value indicates better fit)
- value of Schwarz Bayesian criterion (SBC) at the solution (a smaller value indicates better fit)

Below the “Model Fit Summary” table is a statement about whether the algorithm successfully converged.

Parameter Estimates

The “Parameter Estimates” table contains the estimates of the model parameters. For the normal copula, this table is not displayed because the only parameters are in the correlation matrix, which is displayed in the “Correlation Matrix” table. For the *t* copula, the parameter is the number of degrees of freedom; in the table it is called “DF.” For Archimedean copulas such as Clayton, Frank, and Gumbel, the parameter is called “theta.”

Correlation Matrix

The “Correlation Matrix” table contains the estimates of the model correlation matrix. This table is displayed only for elliptical copulas such as the normal and *t* copulas. Row and column names come from the list of variables defined in the VAR statement.

OUTCOPULA= Data Set

The OUTCOPULA= data set consists of several rows. The first row (with _TYPE_=‘PARM’) contains the parameter estimates in the model. For a *t* copula, the estimate is the number of degrees of freedom; for Archimedean copulas, the estimate is “theta.” The second row (with _TYPE_=‘STD’) contains the standard error for the parameter estimate in the model. These two rows do not appear for the normal copula.

If you use one of the elliptical copulas, *t* or normal, the rest of the data set contains the correlation matrix estimates. The correlation matrix appears in the observations with _TYPE_=‘CORR’, and the _VARIABLE_ column contains the parameter names.

If METHOD=MLE and the nonlinear optimization subsystem is used, a _STATUS_ column is created that contains a character variable that indicates whether the optimization process reached convergence or failed to converge:

- 0 indicates that the convergence was reached
- 1 indicates that the maximum number of iterations allowed was exceeded
- 2 indicates a failure to improve the function value
- 3 indicates a failure to converge for one of the following reasons:
  - The objective function or its derivatives could not be evaluated or improved.
  - Linear constraints are dependent.
  - The algorithm failed to return to feasible region.
  - The number of iterations is greater than prespecified.

OUTPSEUDO=, OUT=, and OUTUNIFORM= Data Sets

The OUTPSEUDO=, OUT=, and OUTUNIF= data sets contain the same number of columns as specified in the VAR statement. The names of the columns are taken from the same VAR statement list.

ODS Table Names

PROC COPULA assigns a name to each table it creates. You can use these names to denote the table when you use the Output Delivery System (ODS) to select tables and create output data sets. These names are listed in Table 10.4.

Table 10.4 ODS Tables Produced in PROC COPULA

ODS Table Name	Description	Option
ODS Tables Created by the FIT Statement		
ConvergenceStatus	Convergence status	Default
Correlation	Correlation matrix estimates	Default with elliptical copulas
KendallCorrelation	Kendall correlation matrix estimates	Default with elliptical copulas
SpearmanCorrelation	Spearman correlation matrix estimates	Default with normal copula
FitSummary	Summary of nonlinear estimation	Default
ParameterEstimates	Parameter estimates	Default
ConvergenceStatus	Convergence status	ITPRINT
InputOptions	Input options	ITPRINT
IterHist	Iteration history	ITPRINT
IterStart	Optimization start	ITPRINT

Table 10.4 *continued*

ODS Table Name	Description	Option
IterStop	Optimization results	ITPRINT
ParameterEstimatesResults	Parameter estimates	ITPRINT
ParameterEstimatesStart	Parameter estimates	ITPRINT
ProblemDescription	Problem description	ITPRINT

ODS Graph Names

Statistical procedures use ODS Graphics to create graphs as part of their output. ODS Graphics is described in detail in Chapter 21, "Statistical Graphics Using ODS" (*SAS/STAT User's Guide*).

Before you create graphs, ODS Graphics must be enabled (for example, with the ODS GRAPHICS ON statement). For more information about enabling and disabling ODS Graphics, see the section "Enabling and Disabling ODS Graphics" in that chapter.

The overall appearance of graphs is controlled by ODS styles. Styles and other aspects of using ODS Graphics are discussed in the section "A Primer on ODS Statistical Graphics" in that chapter.

PROC COPULA assigns a name to each graph it creates by using ODS. You can use these names to refer to the graphs when you use ODS. The names are listed in Table 10.5.

Table 10.5 ODS Graphics Produced by PROC COPULA

ODS Graph Name	Plot Description	Statement	PLOTS= Option

MatrixPlotOrig	Matrix panel of pairwise scatter plots of the original data	FIT	DATATYPE=BOTH,
			DATATYPE=ORIGINAL

MatrixPlotUnif	Matrix panel of pairwise scatter plots of the original data transformed into uniform marginals	FIT	DATATYPE=BOTH,
			DATATYPE=UNIFORM

MatrixPlotSOrig	Matrix panel of pairwise scatter plots of the simulated data	SIMULATE	DATATYPE=BOTH,
			DATATYPE=ORIGINAL

MatrixPlotSUnif	Matrix panel of pairwise scatter plots of the simulated data transformed into uniform marginals	SIMULATE	DATATYPE=BOTH,
			DATATYPE=UNIFORM

ScatterPlotOrig	Pairwise scatter plots of the original data	FIT	DATATYPE=BOTH UNPACK,
			DATATYPE=ORIGINAL UNPACK

Table 10.5 *continued*

ODS Graph Name	Plot Description	Statement	PLOTS= Option
ScatterPlotUnif	Pairwise scatter plots of the original data transformed into uniform marginals	FIT	DATATYPE=BOTH UNPACK, DATATYPE=UNIFORM UNPACK
ScatterPlotSOrig	Pairwise scatter plots of the simulated data	SIMULATE	DATATYPE=BOTH UNPACK, DATATYPE=ORIGINAL UNPACK
ScatterPlotSUnif	Pairwise scatter plots of the simulated data transformed into uniform marginals	SIMULATE	DATATYPE=BOTH UNPACK, DATATYPE=UNIFORM UNPACK
CdfContourPlot	Contour plot of theoretical bivariate CDF function	SIMULATE	DISTRIBUTION=CDF
CdfSurfacePlot	Surface plot of theoretical bivariate CDF function	SIMULATE	DISTRIBUTION=CDF
PdfContourPlot	Contour plot of theoretical bivariate PDF function	SIMULATE	DISTRIBUTION=PDF
PdfSurfacePlot	Surface plot of theoretical bivariate PDF function	SIMULATE	DISTRIBUTION=PDF
ChiPlotOrig	Tail dependence plot matrix with original data	FIT	
ChiPlotUnif	Tail dependence plot matrix with original data transformed into uniform marginals	FIT	
ChiPlotSOrig	Tail dependence plot matrix with simulated data	SIMULATE	
ChiPlotSUnif	Tail dependence plot matrix with simulated data transformed into uniform marginals	SIMULATE	
ChiPlot	Pairwise tail dependence plot of the data	FIT	UNPACK
ChiPlotS	Pairwise tail dependence plot of the simulated data	SIMULATE	UNPACK

Examples: COPULA Procedure

Example 10.1: Copula-Based VaR Estimation

Value-at-risk (VaR) has become a de facto standard in financial risk management. The purpose of this measure is to give some quantitative insight to the riskiness of an asset portfolio. This measure is expressed generically in the following terms: What is the probability of losing no more than given percentage of a portfolio in a certain period of time? Or, what are the maximum possible losses at a given confidence level? The most simple and clearly wrong answer to this question is to compute the empirical quantile of past portfolio returns. The problem of this approach is that it does not take into account the dynamic nature of asset returns, the possibility of changing distribution, time memory, and, most importantly, cross-sectional dependence between individual assets in the portfolio.

This simple example of VaR computation takes into account at least cross-sectional dependence of the data. The end result is the prediction of the next-day maximum possible loss on the portfolio of stocks.

This example uses the daily returns on large stocks such as IBM, Microsoft, British Petroleum, Coca-Cola, and Duke Energy. Output 10.1.1 shows the first 10 observations of the data.

Output 10.1.1 First 10 Observations of Daily Returns

Obs	date	ret_msft	ret_ko	ret_ibm	ret_duk	ret_bp
1	01/03/2008	0.004182	0.010367	0.002002	0.003503	0.019114
2	01/04/2008	-0.027960	0.001913	-0.035861	-0.000582	-0.014536
3	01/07/2008	0.006732	0.023607	-0.010671	0.025611	0.017922
4	01/08/2008	-0.033435	0.004239	-0.024610	-0.002838	-0.016049
5	01/09/2008	0.029560	0.026680	0.007301	0.010814	-0.027078
6	01/10/2008	-0.003054	0.004441	0.016414	-0.001689	-0.004395
7	01/11/2008	-0.012255	-0.027346	-0.022546	-0.012408	-0.018473
8	01/14/2008	0.013958	0.008418	0.053857	0.003427	0.001166
9	01/15/2008	-0.011318	-0.010851	-0.010689	-0.017075	-0.040925
10	01/16/2008	-0.022587	-0.015021	-0.001955	0.002316	-0.021336

The purpose of this exercise is to estimate one-day future losses of a stock portfolio. The simplest approach is to assume that the joint distribution of individual asset returns does not change with time. This might be close to the truth if only a small time interval is used. Then, a copula approach is used to estimate the joint distribution. Next, the new large sample of daily individual asset returns is simulated from the fitted joint distribution. These assets are then combined into a portfolio and its daily returns are computed. Finally, quantiles of simulated portfolio returns (which simply represent possible next-day losses of the portfolio) are examined.

The first step is to cut off a small number of past return observations as in the following SAS DATA step:

```
/* Keep only the last 250 observations of the data */
data returns;
   set returns nobs=observ;
   if (_N_ > observ-250);
run;
```

The following statements fit a *t* copula to the returns data and at the same time simulate the sample from the fitted joint distribution:

```
/* Copula estimation and simulation of returns */
proc copula data = returns;
   var ret_ibm ret_msft ret_bp ret_ko ret_duk;
   * fit T-copula to stock returns;
   fit T /
          marginals = empirical
          method    = MLE
          plots      = (datatype = both);
   * simulate 10000 observations;
   * independent in time, dependent in cross-section;
   simulate /
            ndraws = 10000
            seed   = 1234
            out    = simulated_returns
            plots(unpack) = (datatype = original);
run;
```

The first line of the COPULA procedure uses a VAR statement to specify the list of variables. In this example, these are daily returns of five large-company stocks.The next statement, FIT, requires some options. First, Student's *t* copula (T) is specified. After the slash, the MARGINALS=EMPIRICAL option specifies that an empirical distribution be fit. The choice of fitting method is MLE. The PLOTS=BOTH option requests that both original and transformed data graphs be organized into a symmetric panel.

Then, given the estimation results, the NDRAWS= option in the SIMULATE statement simulates 10,000 new observations for each asset return series. The SEED= option fixes the random number generator, the OUT= option specifies the name of SAS data set to contain the simulated sample, and the PLOT= option requests scatter plots of simulated returns in the original data scale.

The output of these statements is shown in Output 10.1.2.

Output 10.1.2 Copula Estimation

The COPULA Procedure

Model Fit Summary	
Number of Observations	250
Data Set	WORK.RETURNS
Copula Type	T
Log Likelihood	171.52667
Maximum Absolute Gradient	2.52364E-7
Number of Iterations	9
Optimization Method	Newton-Raphson
AIC	-321.05333
SBC	-282.31726

Parameter Estimates

Parameter	Estimate	Standard Error	t Value	Approx Pr > |t|
DF	6.713591	1.349789	4.97	<.0001

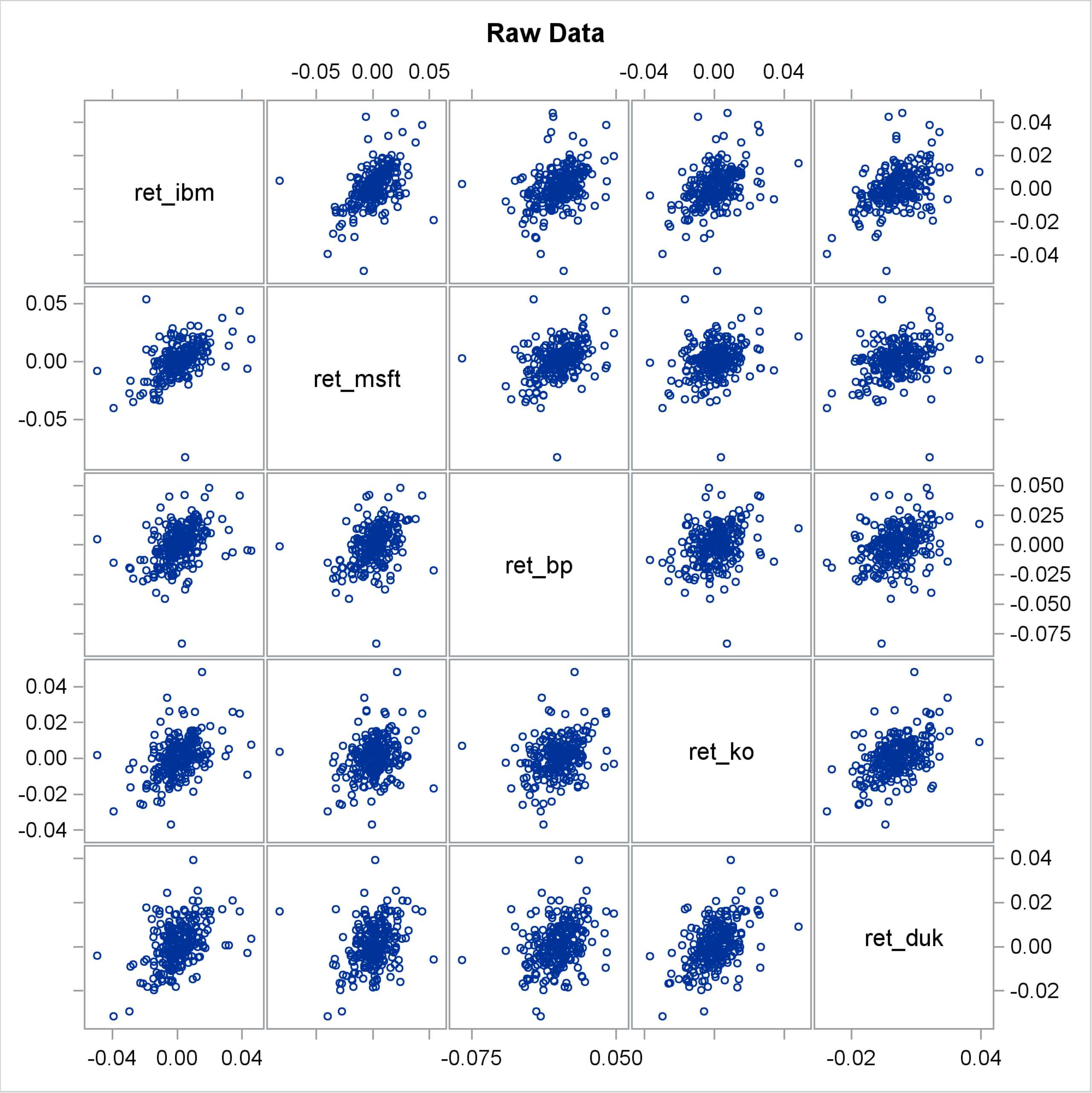
Correlation Matrix

	ret_ibm	ret_msft	ret_bp	ret_ko	ret_duk
ret_ibm	1.0000	0.5657	0.4663	0.4548	0.4740
ret_msft	0.5657	1.0000	0.4585	0.3234	0.3658
ret_bp	0.4663	0.4585	1.0000	0.3459	0.3576
ret_ko	0.4548	0.3234	0.3459	1.0000	0.4742
ret_duk	0.4740	0.3658	0.3576	0.4742	1.0000

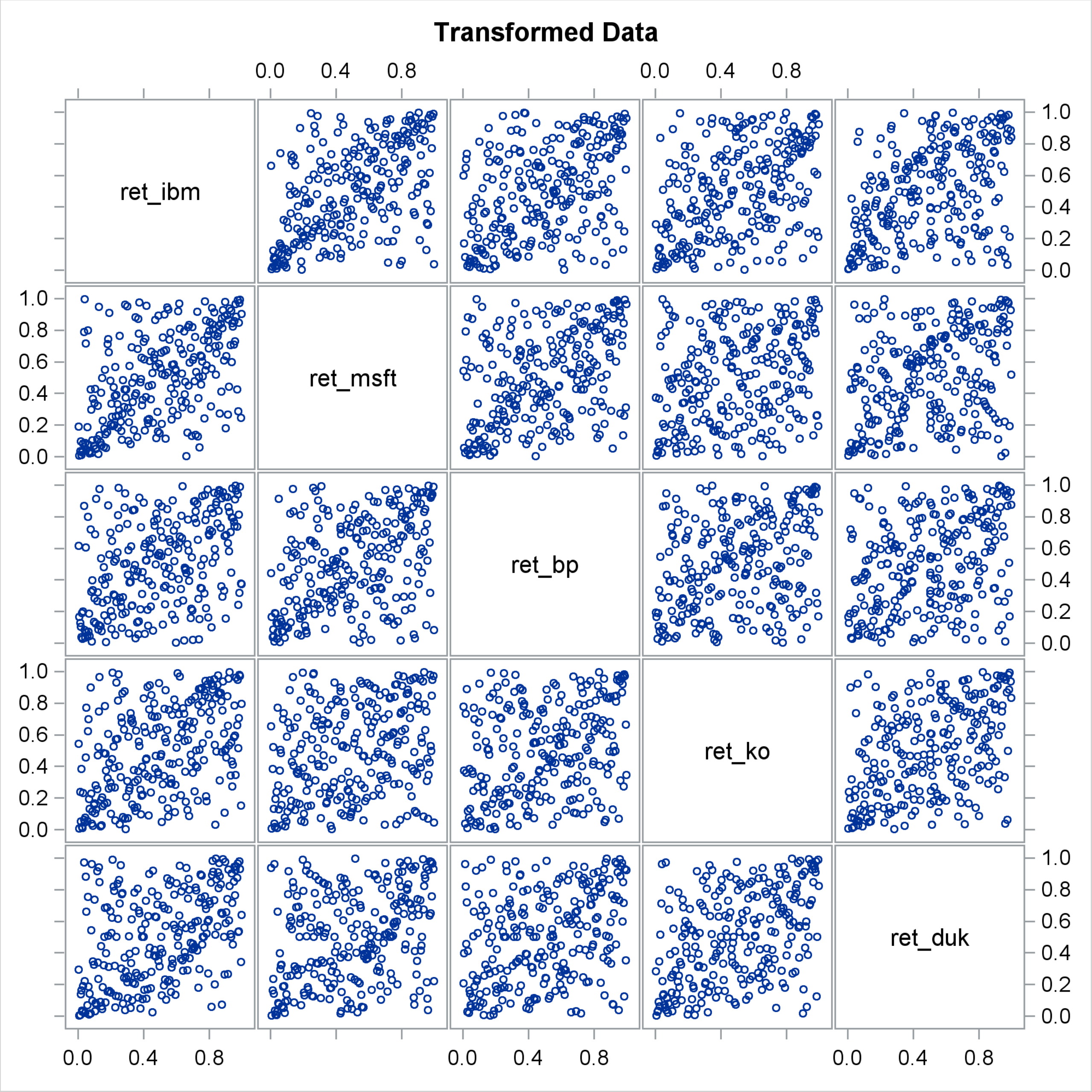
The first table in Output 10.1.2, “Model Fit Summary,” provides some general description of copula model estimation. The second table, “Parameter Estimates,” provides point estimates and inference on copula parameters. In this example the only parameter in this table is the number of degrees of freedom in the multivariate *t* distribution. The last table, “Correlation Matrix,” contains estimates of copula model parameters.

The graphical output of the preceding statements is in Output 10.1.3 and in Output 10.1.4.

Output 10.1.3 Original Data



Output 10.1.4 Original Data Transformed into Uniform Marginals



Note that in Output 10.1.3 the most elliptical scatter plot, between IBM and MSFT, indicates the strongest dependence. Similarly, in Output 10.1.4 those graphs that are denser along the diagonal indicate the same thing.

Now the equally weighted next day portfolio return is computed. Each individual return is transformed into nominal scale first, then all returns are added up with equal weights, and the result is transformed into a net return by subtracting one.

```
/* compute equally weighted portfolio return */
data port_ret (drop = i ret);
   set simulated_returns;
   array returns{5} ret_ibm ret_msft ret_bp ret_ko ret_duk;
   ret =0;
   do i =1 to 5;
      ret = ret+ 0.2*exp(returns[i]);
   end;
   port_ret = ret-1;
run;
```

The final step is to compute empirical quantiles of simulated daily portfolio return. This is done with the help of PROC UNIVARIATE in the following statements:

```
/* compute descriptive statistics */
/* quantile table will give Value-at-Risk estimates for the portfolio */
proc univariate data = port_ret;
   var port_ret;
run;
```

Output 10.1.5 shows that with 99% confidence the potential loss on an equally weighted portfolio over the next day does not exceed 2.6% (the number in table is multiplied by 100). You can also say that there is no more than a 5% chance of losing 1.5% of the portfolio value. These percentage measures are exactly the value-at-risk.

Output 10.1.5 Return Quantiles

The UNIVARIATE Procedure
Variable: port_ret

Quantiles (Definition 5)

Level	Quantile
100% Max	0.048144752
99%	0.026628638
95%	0.015538195
90%	0.011573915
75% Q3	0.005801203
50% Median	0.000688897
25% Q1	-0.004953729
10%	-0.010636996
5%	-0.014677061
1%	-0.026629715
0% Min	-0.052757770

Example 10.2: Simulating Default Times

Suppose the correlation structure required for a normal copula function is already given. For example, it can be estimated from the historic data on default times in some set of industries, but this stage is not in 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 10.2.1 show that the correlation parameter is set at 0.8:

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

Output 10.2.1 Copula Correlation Matrix

Obs	name	Y1	Y2
1	Y1	1.0	0.8
2	Y2	0.8	1.0

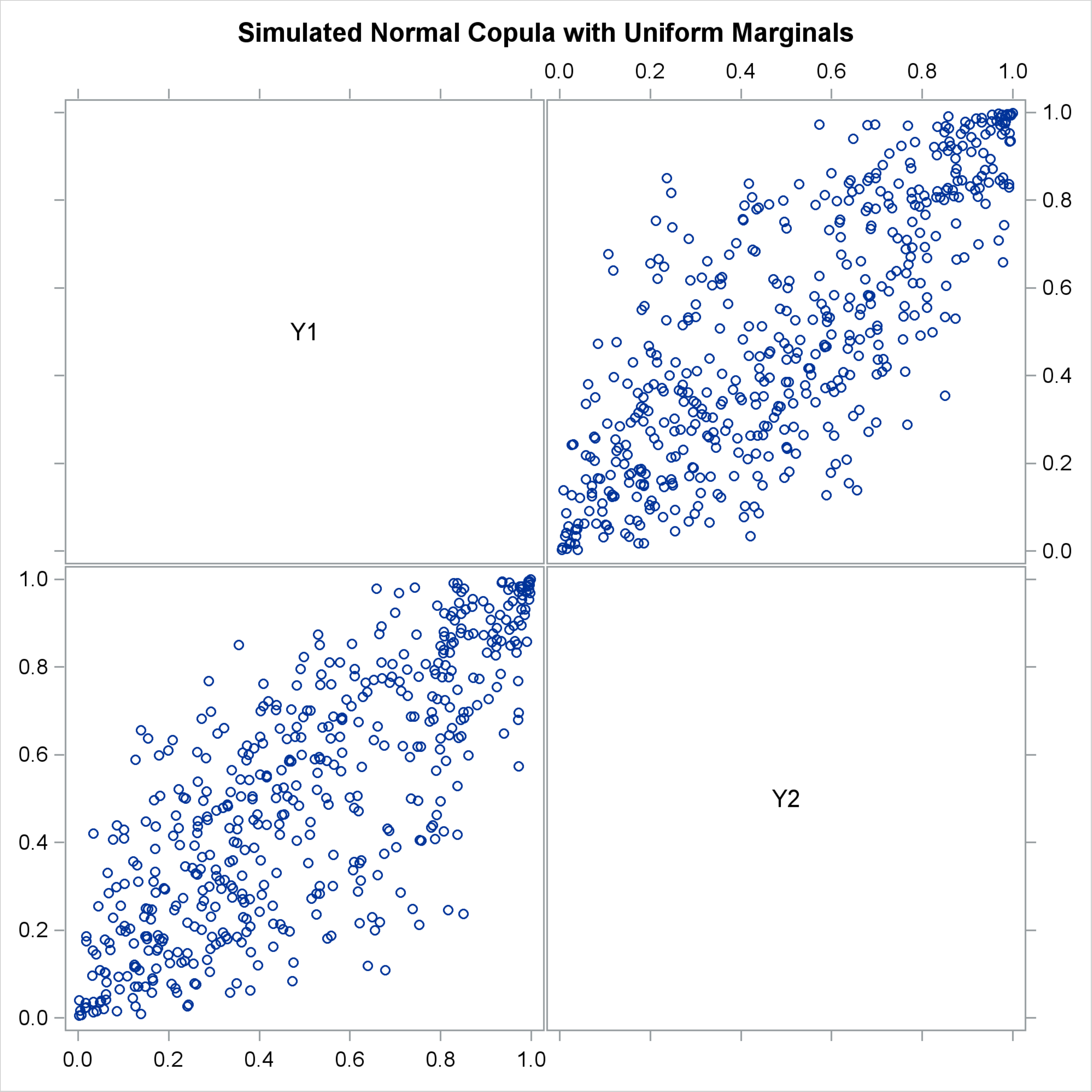
Now you use PROC COPULA to simulate the data. The VAR statement specifies the list of variables to contain simulated data. The DEFINE statement assigns the name COP and specifies a normal copula that reads the correlation matrix from the inparm data set.

The SIMULATE statement refers to the COP label defined in the VAR statement and specifies some options: the NDRAWS= option specifies a sample size, the SEED= option specifies 1234 as the random number generator seed, the OUTUNIFORM=NORMAL_UNIFDATA option names the output data set for the result of simulation in uniforms, and the PLOTS= option requests the matrix of data scatter plots and marginal distributions (DATATYPE=ORIGINAL) and theoretical cumulative distribution function contour and surface plots (DISTRIBUTION=CDF). Theoretical distribution graphs work only for the bivariate case.

```
/* simulate the data from bivariate normal copula */
proc copula ;
   var Y1-Y2;
   define cop normal (corr=inparm);
   simulate cop /
            ndraws     = 500
            seed       = 1234
            outuniform = normal_unifdata
            plots      = (datatype = original
                          distribution = cdf);
run;
```

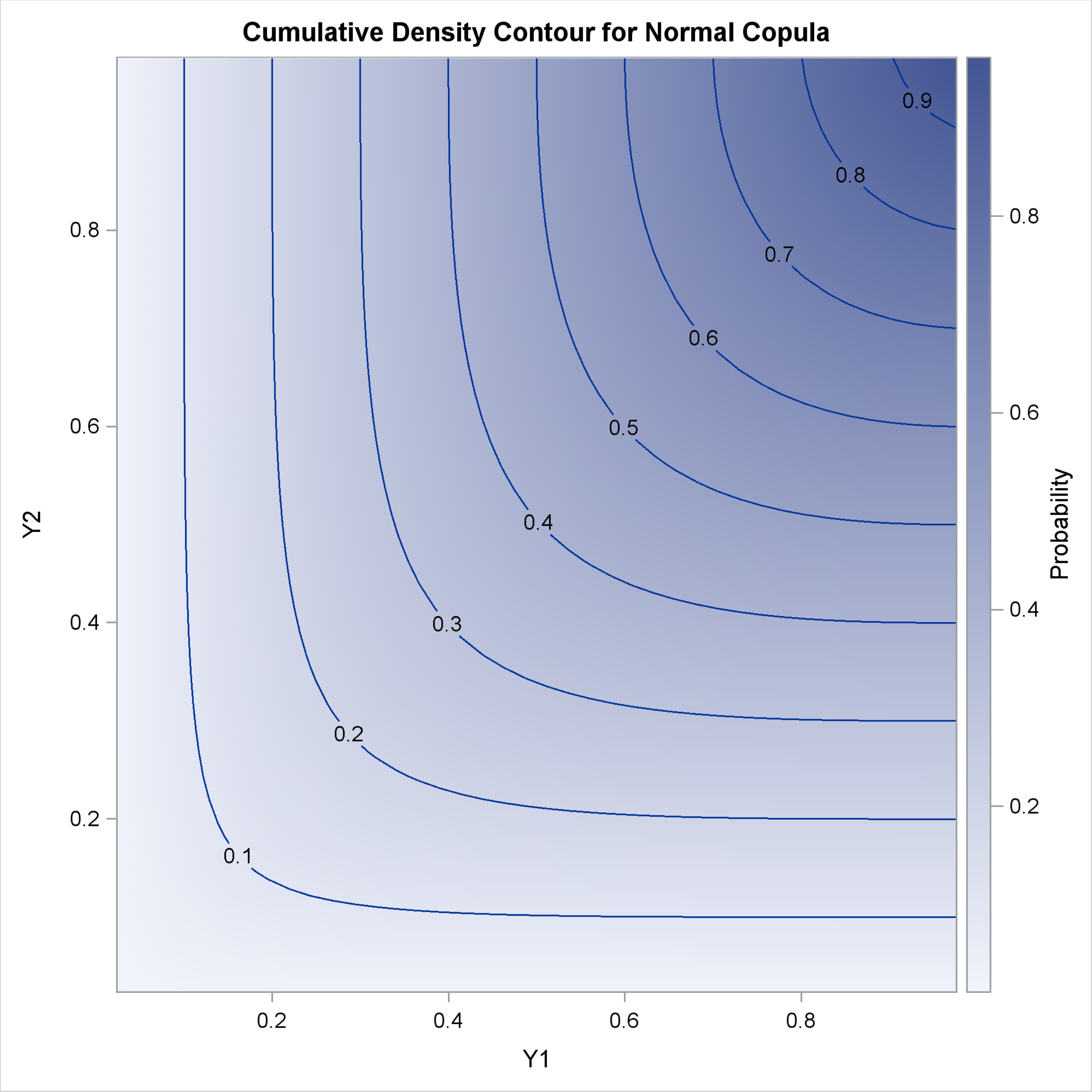
The graphical output is shown in Output 10.2.2 and in Output 10.2.3.

Output 10.2.2 Simulated Data, Uniform Marginals



Output 10.2.2 shows bivariate scatter plots of the simulated data. Also note that due to the high correlation parameter (0.8), the scatter plots are most dense around the 45 degree line, which indicates high dependence between the two variables.

Output 10.2.3 Joint Cumulative Distribution



Output 10.2.3 shows the theoretical CDF contour plot. If the correlation parameter were set to 0, then knowing copula properties you would expect perfectly parallel straight lines with the slope of –45 degrees. On the other hand, if the parameter were set to 1, you would expect perpendicular lines with corners lying on the diagonal.

The next DATA step transforms the variables from zero-one uniformly distributed to nonnegative exponentially distributed with parameter 0.5. Three indicator variables are added to the data set as well. SURVIVE1 and SURVIVE2 are equal to 1 if a respective company has remained in business for more than three years. 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;
```

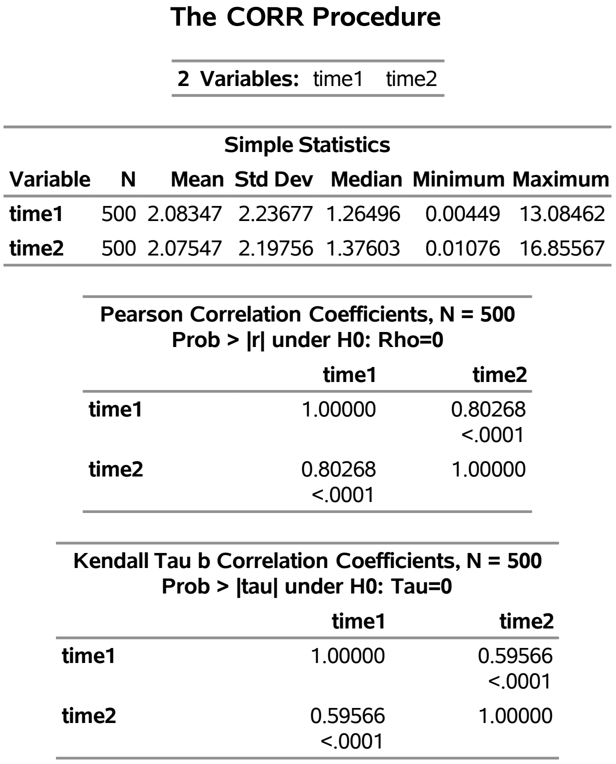
The first analysis step is to look at correlations between survival times of two companies. This step is performed with the following CORR procedure:

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

The output of this code is given in Output 10.2.4 and in Output 10.2.5.

Output 10.2.4 shows some descriptive statistics and two measures of correlation: Pearson and Kendall. Both of these measures indicate high and statistically significant dependence between life spans of two companies.

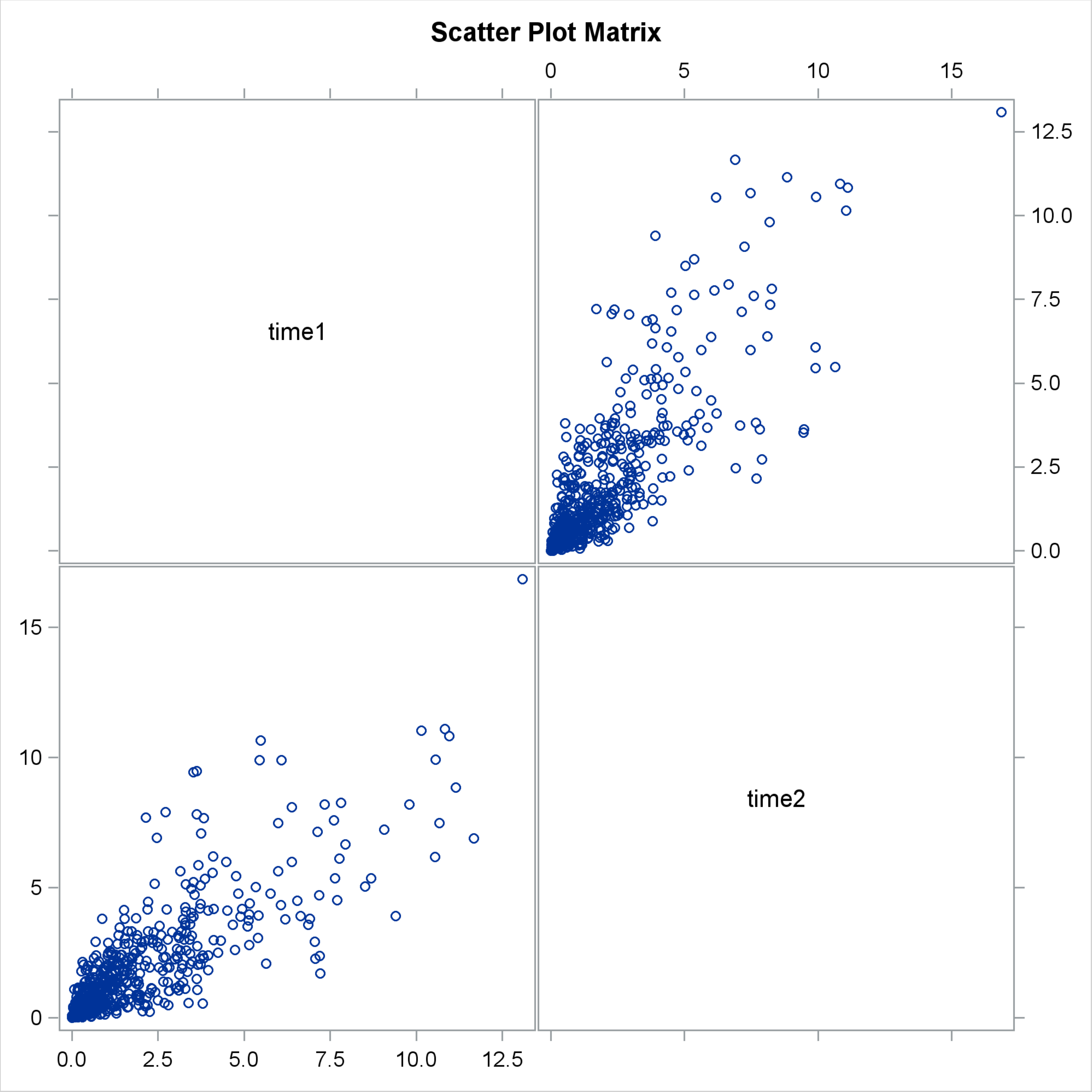
Output 10.2.4 Default Time Descriptive Statistics and Correlations



```
                        The CORR Procedure

                     2 Variables:  time1    time2

                          Simple Statistics
Variable    N     Mean  Std Dev    Median  Minimum  Maximum
time1     500  2.08347  2.23677   1.26496  0.00449  13.08462
time2     500  2.07547  2.19756   1.37603  0.01076  16.85567

        Pearson Correlation Coefficients, N = 500
               Prob > |r| under H0: Rho=0
                          time1          time2
time1                   1.00000        0.80268
                                        <.0001
time2                   0.80268        1.00000
                         <.0001

     Kendall Tau b Correlation Coefficients, N = 500
              Prob > |tau| under H0: Tau=0
                           time1          time2
time1                    1.00000        0.59566
                                         <.0001
time2                    0.59566        1.00000
                          <.0001
```

Output 10.2.5 shows marginal distributions and scatter plots of simulated data. Distributions are noticeably close to exponential and scatter plots show a high degree of dependence.

Output 10.2.5 Default Times



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

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

The result is shown in Output 10.2.6.

Output 10.2.6 Probabilities of Default

```
                 The FREQ Procedure

                                   Cumulative  Cumulative
survive  Frequency   Percent   Frequency      Percent
      0        415     83.00         415        83.00
      1         85     17.00         500       100.00

                                    Cumulative  Cumulative
survive1  Frequency   Percent   Frequency      Percent
       0        374     74.80         374        74.80
       1        126     25.20         500       100.00

                                    Cumulative  Cumulative
survive2  Frequency   Percent   Frequency      Percent
       0        390     78.00         390        78.00
       1        110     22.00         500       100.00
```

Output 10.2.6 shows that the empirical default probabilities are 75% and 78%. Assuming that these companies are independent gives the probability estimate of both companies defaulting during the period of three years as: 0.75*0.78=0.59 (59%). Comparing this naive estimate with the much higher actual 83% joint default probability illustrates that neglecting the correlation between the two companies significantly underestimates the probability of default.

References

Cherubini, U., Luciano, E., and Vecchiato, W. (2004). *Copula Methods in Finance*. Chichester, UK: John Wiley & Sons.

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

Fisher, N. I., and Switzer, P. (2001). "Graphical Assessment of Dependence: Is a Picture Worth 100 Tests?" *American Statistician* 55:233–239.

Galiani, S. S. (2003). "Copula Functions and Their Application in Pricing and Risk Managing Multiname Credit Derivative Products." Master's thesis, King's College London. `http://www.defaultrisk.com/pp_crdrv_41.htm`.

Genest, C., Ghoudi, K., and Rivest, L. P. (1995). "A Semiparametric Estimation Procedure of Dependence Parameters in Multivariate Families of Distributions." *Biometrika* 82:543–552.

Hofert, M. (2011). "Efficiently Sampling Nested Archimedean Copulas." *Computational Statistics and Data Analysis* 55:57–70.

Joe, H. (1997). *Multivariate Models and Dependence Concepts*. London: Chapman & Hall.

Joe, H., and Xu, J. (1996). *The Estimation Method of Inference Functions for Margins for Multivariate Models*. Technical Report 166, University of British Columbia.

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

McNeil, A., Frey, R., and Embrechts, P. (2005). *Quantitative Risk Management: Concepts, Techniques, and Tools*. Princeton, NJ: Princeton University Press.

McNeil, A. J. (2008). "Sampling Nested Archimedean Copulas." *Journal of Statistical Computation and Simulation* 78:567–581.

Mendes, B. V. M., de Melo, E. F. L., and Nelsen, R. B. (2007). "Robust Fits for Copula Models." *Communications in Statistics—Simulation and Computation* 36:997–1008.

Nelsen, R. B. (2006). *An Introduction to Copulas*. 2nd ed. New York: Springer.

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

Rüschendorf, L. (2009). "On the Distributional Transform, Sklar's Theorem, and the Empirical Copula Process." *Journal of Statistical Planning and Inference* 11:3921–3927.

Savu, C., and Trede, M. (2010). "Hierarchies of Archimedean Copulas." *Quantitative Finance* 10:295–304.

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.

Wu, F., Valdez, E., and Sherris, M. (2007). "Simulating from Exchangeable Archimedean Copulas." *Communications in Statistics—Simulation and Computation* 36:1019–1034.

Subject Index

Syntax Index