
# Messung von Marktrisiken unter Verwendung von Copulafunktionen

## Eine empirische Studie für den Schweizer Aktienmarkt

GAUSS™-Bibliothek für Copulafunktionen – Handbuch

von

Manrico Glauser

Freiburg 2003

Nachfolgend wird eine in der Programmierumgebung GAUSS™ der Firma Aptech Systems, Inc. realisierte Programmbibliothek zur Schätzung und Simulation von verschiedenen Copulafunktionen beschrieben. Die Verwendung setzt GAUSS™ ab Version 4.0 und das Zusatzpaket Constrained Maximum Likelihood ab Version 2.0 voraus, das Algorithmen für die numerische Maximum-Likelihood-Schätzung unter Berücksichtigung von Parameterrestriktionen bereitstellt.

Bei den betrachteten Copulafunktionen handelt es sich um

- die Normal-Copula,
- die $t$-Copula,
- die Farlie-Gumbel-Morgenstern-Copula,
- die Gumbel-Copula,
- die Kimeldorf-Sampson-Copula und
- die so genannte Nelsen-Copula.

Die 121 Prozeduren umfassende Programmbibliothek für Copulafunktionen enthält zunächst Routinen, welche direkt die Copulafunktionen betreffen und in GAUSS™ durch den Aufruf

```
» library copulas
```

aktiviert werden. Weiter sind die benötigten Hilfsprozeduren enthalten, die durch den Aufruf

```
» library copulautils
```

zu aktivieren sind.

Die Darstellung der Prozeduren ist in Englisch abgefasst und wird in dem Stil präsentiert, der bei der Beschreibung von in GAUSS™ geschriebenen Routinen üblicherweise gepflegt wird. Dabei wird zunächst der Zweck einer Prozedur kurz umrissen und es wird die Syntax für den Prozeduraufruf angegeben. Anschliessend werden Input und Output beschrieben, wobei es sich, wie in GAUSS™ vorausgesetzt, jeweils um Matrizen handelt. Optional folgen Beispiele, Bemerkungen und Verweise auf verwandte Prozeduren. Den Abschluss bildet jeweils die Angabe der Quelldatei und der Bibliothek, in der die entsprechende Prozedur zu finden ist. Die Prozeduren sind nachfolgend in alphabetischer Reihenfolge dargestellt, wobei jede auf einer eigenen Seite beschrieben wird.

Die Prozedur `rchisquare` zur Generierung von $\chi^2$-verteilten Zufallszahlen konnte aus der von Schlittgen und Noack zur Verfügung gestellten Programmbibliothek DISTRIB übernommen werden.[1]

[1] Die Programmbibliothek DISTRIB kann unter der Adresse `http://www.american.edu/academic.depts/cas/econ/gaussres/pdf/distrib.zip` [Stand 2003-08-08] heruntergeladen werden.

## binarylossfunc

Purpose: Provides the number of losses in a sample that are larger than the corresponding VaR estimates as well as the samplesize.

Format: `{noofexceed,total} = binarylossfunc(VaR,loss);`

Input:

| | |
|---|---|
| VaR | TxL matrix, VaR's for L different confidence levels. |
| loss | Tx1 vector, losses. |

Output:

| | |
|---|---|
| noofexceed | 1xL vector, number of losses larger than corresponding VaR. |
| total | scalar, number of comparisons. |

Example:

```
V1 = {10,11,10,12,10,13,11,14,15,10,12};
V2 = {10,11,10,12,10,13,11,14,17,13,15};
VaR = V1 ~ V2;
loss = {10,10,9,11,9,11,10,15,16,12,14};
{noofexceed,total} = binarylossfunc(VaR,loss);

noofexceed = 4.0000000        1.0000000
total = 11.000000
```

Source: backtestenv.src

Library: copulautils.lcg

## chi2n

Purpose: Computes the Chi-Square test for normality with mean and variance unknown. The null hypothesis H_0 is: x ~ N(mu,sigma^2). Since mu and sigma^2 have to be estimated from data, the degrees of freedom are reduced by 2.

Format: {teststat,Pvalue} = chi2n(x);

Input: x TxN matrix, data.

Output: teststat 1xN vector, values of test statistic.

Pvalue 1xN vector, P-values.

Example:
```
u = rndn(1000,1);
v = rndu(1000,1);
x = u ~ v;
{teststat,Pvalue} = chi2n(x);

teststat = 5.0240000        147.93600

Pvalue =   0.9746610        5.3664852e-025
```

See also: kolsmiLilliefors

Source: distrtest.src

Library: copulautils.lcg

## decimaltobinary

```
Purpose:    Converts decimal to binary numbers.

Format:     b = decimaltobinary(d);

Input:      d       Tx1 vector, decimal numbers.

Output:     b       TxM matrix, binary numbers.

Example:    b = decimaltobinary(1|2|3|4);

                0.0000000       0.0000000       1.0000000
            b = 0.0000000       1.0000000       0.0000000
                0.0000000       1.0000000       1.0000000
                1.0000000       0.0000000       0.0000000

Source:     dectobin.src

Library:    copulautils.lcg
```

## effectpfloss

Purpose: Computes the loss of a portfolio of different assets between two different points in time.

Format: epfl = effectpfloss(assetweights,earlyprice,lateprice);

Input:

| | |
|---|---|
| assetweights | 1xN vector, number of every type of asset in the portfolio. |
| earlyprice | 1xN vector, former prices. |
| lateprice | 1xN vector, latter prices. |

Output:

| | |
|---|---|
| epfl | scalar, portfolioloss. |

Example:

```
assetweights = {1 1};
earlyprice = {50 100};
lateprice = {40 120};
epfl = effectpfloss(assetweights,earlyprice,lateprice);

epfl = -10.000000
```

Source: simenv.src

Library: copulautils.lcg

## empdist

```
Purpose:    Provides univariate emprical distribution functions
            of given data.

Format:     ed = empdist(x);

Input:      x       TxN matrix, data.

Output:     ed      TxN matrix, empirical distribution functions.

Remarks:    The TxN datamatrix corresponds to N vectors of
            T realisations each. The TxN matrix ed corresponds
            to N empirical distribution functions.

Example:    x = {1 2,
                 1 2,
                 1 5,
                 4 5,
                 2 3,
                 3 1};
            ed = empdist(x);

            ed = 0.50000000       0.50000000
                 0.50000000       0.50000000
                 0.50000000       1.0000000
                 1.0000000        1.0000000
                 0.66666667       0.66666667
                 0.83333333       0.16666667

 See also:    empdistinv

Source:     empdist.src

Library:    copulautils.lcg
```

## empdistinv

```
Purpose:    Computes the inverses of empirical distribution
            functions (quantiles), given specified probabilities.

Format:     xn = empdistinv(x,u);

Input:      x       TxN matrix, original data that leaded to the
                    empirical distribution functions.

            u       KxN matrix, probabilities.

Output:     xn      KxN matrix, inverses of empirical distribution
                    functions.

Example:    x = {1 2,
                 1 2,
                 1 5,
                 4 5,
                 2 3,
                 3 1};

            u = {0.50000000       0.50000000,
                 0.50000000       0.50000000,
                 0.50000000       1.0000000 ,
                 1.0000000        1.0000000 ,
                 0.66666667       0.66666667,
                 0.83333333       0.16666667} ;

            xn = empdistinv(x,u);

            xn = 1.0000000        2.0000000
                 1.0000000        2.0000000
                 1.0000000        5.0000000
                 4.0000000        5.0000000
                 2.0000000        3.0000000
                 3.0000000        1.0000000

 See also:    empdist

Source:     empdist.src

Library:    copulautils.lcg
```

## ES

```
Purpose:    Provides the Expected Shortfall.

Format:     e = ES(loss,confidence);

Input:      loss            Tx1 vector, losses.

            confidence      1xL vector, confidence levels.

Output:     e               1xL vector, Expected Shortfall's.

Example:    loss = {10,10,9,11,9,11,10,15,16,12,14};
            confidence = {0.9 0.8 0.7};
            e = ES(loss,confidence);

            e = 16.000000        15.500000        15.000000

Source:     simenv.src

Library:    copulautils.lcg
```

## fgmcopak

Purpose: Computes the value of a coefficient that is needed by the procedure fgmcopmlnum to simulate realisations from the reduced FGM copula.

Format: ak = fgmcopak(theta,U);

Input: theta (N(N-1)/2)x1 vector, parameters.

U TxN matrix, values between 0 and 1.

Output: ak Tx1 vector.

See also: fgmcopmlnum

Source: fgmcop.src

Library: copulas.lcg

## fgmcopck

```
Purpose:    Returns the value of the density-function of the
            reduced FGM copula.

Format:     ck = fgmcopck(theta,U);

Input:      theta   (N(N-1)/2)x1 vector, parameters.

            U       TxN matrix, values between 0 and 1.

Output:     ck      Tx1 vector.

Source:     fgmcop.src

Library:    copulas.lcg
```

## fgmcopktausrho

Purpose: Returns two vectors of estimations of the parameters theta of the reduced FGM copula.
The first vector is derived from the estimation of Kendall's tau correlation matrix, the second from the estimation of Spearman's rho correlation matrix.

Format: `{kt,sr} = fgmcopktausrho(U);`

Input:

| | | |
|---|---|---|
| U | TxN matrix, values between 0 and 1. |

Output:

| | |
|---|---|
| kt | (N(N-1)/2)x1 vector, parameters. |
| sr | (N(N-1)/2)x1 vector, parameters. |

See also: fgmcopmlnum

Source: fgmcop.src

Library: copulas.lcg

## fgmcopktausrhogivkt

Purpose: Returns two vectors of estimations of the parameters theta of the reduced FGM copula.
The first vector is derived from the estimation of Kendall's tau correlation matrix, the second from the estimation of Spearman's rho correlation matrix. The Kendall's correlation matrix must be given.

Format: `{kt,sr} = fgmcopktausrhogivkt(U,ktau);`

Input:

| | | |
|---|---|---|
| | U | TxN matrix, values between 0 and 1. |
| | ktau | NxN matrix, Kendall's tau matrix. |

Output:

| | | |
|---|---|---|
| | kt | (N(N-1)/2)x1 vector, parameters. |
| | sr | (N(N-1)/2)x1 vector, parameters. |

See also: fgmcopktausrho

Source: fgmcop.src

Library: copulas.lcg

## fgmcoploglik

```
Purpose:    Returns the value of the log-likelihood-function
            of the reduced FGM copula.

Format:     l = fgmcoploglik(theta,U);

Input:      theta   (N(N-1)/2)x1 vector, parameters.

            U       TxN matrix of data.

Output:     l       Tx1 vector.

See also:   fgmcopmlnum, fgmcopck

Source:     fgmcop.src

Library:    copulas.lcg
```

## fgmcopmlnum

Purpose: Returns the parameter-vector theta of the reduced FGM copula estimated by the numerical maximum likelihood method.

Format: `theta = fgmcopmlnum(U);`

Input: `U` TxN matrix, values between 0 and 1.

Output: `theta` (N(N-1)/2)x1 vector, parameters.

See also: fgmcoploglik, fgmcopktausrho

Source: fgmcop.src

Library: copulas.lcg

## fgmcopsimu

Purpose: Simulates values from a N-dimensional reduced FGM copula with parameter-vector theta.

Format: `U = fgmcopsimu(T,theta);`

Input:

| | | |
|---|---|---|
| T | scalar. |
| theta | (N(N-1)/2)x1 vector, parameters. |

Output:

| | |
|---|---|
| U | TxN matrix, values between 0 and 1. |

See also: fgmcopak, fgmcopck, quadeq

Source: fgmcop.src

Library: copulas.lcg

## fgmcopsimugivenS

Purpose: Simulates values from a N-dimensional reduced FGM copula with parameter-vector theta. The needed random numbers must be given.

Format: `U = fgmcopsimugivenS(S,theta);`

Input:

| | |
|---|---|
| S | TxN matrix, U(0,1) random numbers. |
| theta | (N(N-1)/2)x1 vector, parameters. |

Output:

| | |
|---|---|
| U | TxN matrix, values between 0 and 1. |

See also: fgmcopsimu

Source: fgmcop.src

Library: copulas.lcg

## gumbelcopktaumultiparam

Purpose: Returns the order of variables and the parameter-vector theta of the Gumbel copula derived from the estimation of Kendall's tau correlation matrix.

Format: {idx,theta} = gumbelcopktaumultiparam(U);

Input: U TxN matrix, values between 0 and 1.

Output: idx Nx1 vector, order of the columns of U that provides from all admissible solutions the one with the smallest distance to the target value in the minimization problem to solve.

theta (N-1)x1 vector, parameters.

Source: gumbelcop.src

Library: copulas.lcg

## gumbelcopktaumultiparamgivkt

Purpose: Returns the order of variables and the parameter-vector theta of the Gumbel copula derived from the estimation of Kendall's tau correlation matrix. The Kendall's correlation matrix must be given.

Format: {idx,theta} = gumbelcopktaumultiparamgivkt(U,ktau);

Input:

| | |
|---|---|
| U | TxN matrix, values between 0 and 1. |
| ktau | NxN matrix, Kendall's tau matrix. |

Output:

| | |
|---|---|
| idx | Nx1 vector, order of the columns of U that provides from all admissible solutions the one with the smallest distance to the target value in the minimization problem to solve. |
| theta | (N-1)x1 vector, parameters. |

Source: gumbelcop.src

Library: copulas.lcg

## gumbelcopktauuniparam

```
Purpose:    Returns the parameter theta (scalar) of the Gumbel
            copula derived from the estimation of Kendall's
            tau correlation matrix.

Format:     theta = gumbelcopktauuniparam(U)

Input:      U       TxN matrix, values between 0 and 1.

Output:     theta   scalar, parameter.

Source:     gumbelcop.src

Library:    copulas.lcg
```

## gumbelcopktauuniparamgivkt

Purpose: Returns the parameter theta (scalar) of the Gumbel copula derived from the estimation of Kendall's tau correlation matrix. The Kendall's correlation matrix must be given.

Format: theta = gumbelcopktauuniparamgivkt(ktau)

Input: ktau NxN matrix, Kendall's tau matrix.

Output: theta scalar, parameter.

Source: gumbelcop.src

Library: copulas.lcg

## gumbelcoploglik2

```
Purpose:    Returns the value of the log-likelihood-function
            of the bivariate Gumbel copula.

Format:     l = gumbelcoploglik2(theta,U);

Input:      theta   scalar, parameter.

            U       Tx2 matrix of data.

Output:     l       Tx1 vector.

Source:     gumbelcop.src

Library:    copulas.lcg
```

## gumbelcoploglik3

```
Purpose:    Returns the value of the log-likelihood-function
            of the trivariate Gumbel copula with parameter
            theta (scalar).

Format:     l = gumbelcoploglik3(theta,U);

Input:      theta   scalar, parameter.

            U       Tx3 matrix of data.

Output:     l       Tx1 vector.

Source:     gumbelcop.src

Library:    copulas.lcg
```

## gumbelcoploglik3multiparam

```
Purpose:    Returns the value of the log-likelihood-function
            of the trivariate Gumbel copula with parameter-
            vector theta.

Format:     l = gumbelcoploglik3multiparam(theta,U);

Input:      theta   2x1 vector, parameters.

            U       Tx3 matrix of data.

Output:     l       Tx1 vector.

Source:     gumbelcop.src

Library:    copulas.lcg
```

## gumbelcoploglik4

```
Purpose:    Returns the value of the log-likelihood-function
            of the 4-variate Gumbel copula with parameter
            theta (scalar).

Format:     l = gumbelcoploglik4(theta,U);

Input:      theta   scalar, parameter.

            U       Tx4 matrix of data.

Output:     l       Tx1 vector.

Source:     gumbelcop.src

Library:    copulas.lcg
```

## gumbelcoploglik4multiparam

Purpose: Returns the value of the log-likelihood-function of the 4-variate Gumbel copula with parameter-vector theta.

Format: l = gumbelcoploglik4multiparam(theta,U);

Input: theta 3x1 vector, parameters.

U Tx4 matrix of data.

Output: l Tx1 vector.

Source: gumbelcop.src

Library: copulas.lcg

## gumbelcoploglik5

```
Purpose:    Returns the value of the log-likelihood-function
            of the 5-variate Gumbel copula with parameter
            theta (scalar).

Format:     l = gumbelcoploglik5(theta,U);

Input:      theta   scalar, parameter.

            U       Tx5 matrix of data.

Output:     l       Tx1 vector.

Source:     gumbelcop.src

Library:    copulas.lcg
```

## gumbelcoploglik5multiparam

Purpose: Returns the value of the log-likelihood-function of the 5-variate Gumbel copula with parameter-vector theta.

Format: l = gumbelcoploglik5multiparam(theta,U);

Input: theta 4x1 vector, parameters.

U Tx5 matrix of data.

Output: l Tx1 vector.

Source: gumbelcop.src

Library: copulas.lcg

## gumbelcopmlnummultiparam

Purpose: Returns the order of variables and the parameter-vector theta of the Gumbel copula estimated by the numerical maximum likelihood method.

Format: {idx,theta} = gumbelcopmlnummultiparam(U);

Input: U TxN matrix, values between 0 and 1.

Output: idx Nx1 vector, order of the columns of U that provides from all admissible solutions the one with the highest value of the log-likelihood-function.

theta (N-1)x1 vector, parameters.

Source: gumbelcop.src

Library: copulas.lcg

## gumbelcopmlnummultiparamgivkt

Purpose: Returns the order of variables and the parameter-vector theta of the Gumbel copula estimated by the numerical maximum likelihood method. The Kendall's correlation matrix must be given.

Format: {idx,theta} = gumbelcopmlnummultiparamgivkt(U,ktau)

Input:

| | |
|---|---|
| U | TxN matrix, values between 0 and 1. |
| ktau | NxN matrix, Kendall's tau matrix. |

Output:

| | |
|---|---|
| idx | Nx1 vector, order of the columns of U that provides from all admissible solutions the one with the highest value of the log-likelihood-function. |
| theta | (N-1)x1 vector, parameters. |

Source: gumbelcop.src

Library: copulas.lcg

## gumbelcopmlnumuniparam

```
Purpose:    Returns the parameter theta (scalar) of the Gumbel
            copula estimated by the numerical maximum likelihood
            method.

Format:     theta = gumbelcopmlnumuniparam(U);

Input:      U       TxN matrix, values between 0 and 1.

Output:     theta   scalar, parameter.

Source:     gumbelcop.src

Library:    copulas.lcg
```

## gumbelcopmlnumuniparamgivkt

Purpose: Returns the parameter theta (scalar) of the Gumbel copula estimated by the numerical maximum likelihood method. The Kendall's correlation matrix must be given.

Format: theta = gumbelcopmlnumuniparamgivkt(U,ktau);

Input: U TxN matrix, values between 0 and 1.

ktau NxN matrix, Kendall's tau matrix.

Output: theta scalar, parameter.

Source: gumbelcop.src

Library: copulas.lcg

## gumbelcopsimu

```
Purpose:    Simulates values from a N-dimensional Gumbel copula
            with parameter-vector theta.

Format:     U = gumbelcopsimu(T,theta);

Input:      T       scalar.

            theta   (N-1)x1 vector, parameters.

Output:     U       TxN matrix, values between 0 and 1.

Source:     gumbelcop.src

Library:    copulas.lcg
```

## gumbelcopsimugivenS

Purpose: Simulates values from a N-dimensional Gumbel copula with parameter-vector theta. The needed random numbers must be given.

Format: U = gumbelcopsimugivenS(S,theta);

Input: S TxN matrix, U(0,1) random numbers.

theta (N-1)x1 vector, parameters.

Output: U TxN matrix, values between 0 and 1.

Source: gumbelcop.src

Library: copulas.lcg

## gumbelinvKnum

Purpose: Provides the function K of the Gumbel copula in the form that is needed to calculate numerically the inverse of K.

Format: f = gumbelinvKnum(t);

Input: t Tx1 vector.

Output: f Tx1 vector.

Remarks: The known value y=K(t) must be given in a global variable named "yglob".

See also: invfunc, gumbelinvKstart

Source: gumbelcop.src

Library: copulas.lcg

## gumbelinvKstart

Purpose: Provides the startvalue to calculate numerically the inverse of K of the Gumbel copula.

Format: f = gumbelinvKstart(y);

Input: y Tx1 vector.

Output: f Tx1 vector.

Remarks: y denotes the known value of the function K.

See also: invfunc, gumbelinvKnum

Source: gumbelcop.src

Library: copulas.lcg

## gumbelinvphi

```
Purpose:    Computes the value of the inverse of the function
            phi of the Gumbel copula with parameter theta.

Format:     f = gumbelinvphi(t);

Input:      t   Tx1 vector.

Output:     f   Tx1 vector.

Remarks:    The parameter theta must be given in a global
            variable named "thetaglob".

Source:     gumbelcop.src

Library:    copulas.lcg
```

## gumbelK

```
Purpose:    Computes the value of the function K of the
            Gumbel copula with parameter theta.

Format:     f = gumbelK(t);

Input:      t   Tx1 vector.

Output:     f   Tx1 vector.

Remarks:    The parameter theta must be given in a global
            variable named "thetaglob".

Source:     gumbelcop.src

Library:    copulas.lcg
```

## gumbelphi

```
Purpose:    Computes the value of the function phi of the
            Gumbel copula with parameter theta.

Format:     f = gumbelphi(t);

Input:      t   Tx1 vector.

Output:     f   Tx1 vector.

Remarks:    The parameter theta must be given in a global
            variable named "thetaglob".

Source:     gumbelcop.src

Library:    copulas.lcg
```

## gumbelprimephi

Purpose: Computes the value of the first derivative of the function phi of the Gumbel copula with parameter theta.

Format: f = gumbelprimephi(t);

Input: t Tx1 vector.

Output: f Tx1 vector.

Remarks: The parameter theta must be given in a global variable named "thetaglob".

Source: gumbelcop.src

Library: copulas.lcg

## invfunc

Purpose: Computes numerically the inverse of a function.

Format: x = invfunc(y,&function,&start);

Input: y Tx1 vector, values of the function.

&function function to invert, form:

```
proc function(x);
    local y;
    y = varget("yglob");
    retp(f(x) -y);
endp;
```

&start function, calculates the startvalue in funtion of y, form:

```
proc start(x);
    retp(x);
endp;
```

Output: x Tx1 vector, values of the argument.

Example:

```
proc f(x);
    retp(x+1);
endp;

proc function(x);
    local y;
    y = varget("yglob");
    retp(f(x) -y);
endp;

proc start(x);
    retp(x);
endp;

invfunc(2,&function,&start);

1.0000000
```

Source: invfunc.src

Library: copulautils.lcg

## kendallstau

```
Purpose:    Computes Kendall's tau for two vectors.

Format:     t = kendallstau(x,y);

Input:      x   Tx1 vector.

            y   Tx1 vector.

Output:     t   scalar, Kendall's tau.

Example:    x={ 54.10  54.50  51.60  54.25 54.25 54.45
                53.10  53.20};
            y={ 157.50 158.50 168.00 165.75 160.00 160.00
                163.25 166.25};
            t = kendallstau(x',y');

            t = -0.50000000

See also:   kendallstaumatrix, spearmansrho, spearmansrhomatrix

Source:     kendallstau.src

Library:    copulautils.lcg
```

## kendallstaumatrix

```
Purpose:    Computes the Kendall's tau correlation matrix.

Format:     t = kendallstaumatrix(Z);

Input:      Z   TxN matrix.

Output:     t   symmetric NxN matrix with ones in the diagonal
                and the pairwise Kendall's tau in the non-
                diagonal.

Example:    x={ 54.10  54.50  51.60  54.25 54.25 54.45
                53.10  53.20};
            y={ 157.50 158.50 168.00 165.75 160.00 160.00
                163.25 166.25};
            Z=x'~y';
            t = kendallstaumatrix(Z);

            t =  1.0000000    -0.50000000
                -0.50000000    1.0000000

See also:   kendallstau, spearmansrho, spearmansrhomatrix

Source:     kendallstau.src

Library:    copulautils.lcg
```

## kimsamcopktaumultiparam

Purpose: Returns the order of variables and the parameter-vector theta of the Kimeldorf-Sampson copula derived from the estimation of Kendall's tau correlation matrix.

Format: {idx,theta} = kimsamcopktaumultiparam(U);

Input: U TxN matrix, values between 0 and 1.

Output: idx Nx1 vector, order of the columns of U that provides from all admissible solutions the one with the smallest distance to the target value in the minimization problem to solve.

theta (N-1)x1 vector, parameters.

Source: kimsamcop.src

Library: copulas.lcg

## kimsamcopktaumultiparamgivkt

Purpose: Returns the order of variables and the parameter-vector theta of the Gumbel copula derived from the estimation of Kendall's tau correlation matrix. The Kendall's correlation matrix must be given.

Format: {idx,theta} = kimsamcopktaumultiparamgivkt(U,ktau);

Input:

| | |
|---|---|
| U | TxN matrix, values between 0 and 1. |
| ktau | NxN matrix, Kendall's tau matrix. |

Output:

| | |
|---|---|
| idx | Nx1 vector, order of the columns of U that provides from all admissible solutions the one with the smallest distance to the target value in the minimization problem to solve. |
| theta | (N-1)x1 vector, parameters. |

Source: kimsamcop.src

Library: copulas.lcg

## kimsamcopktauuniparam

Purpose: Returns the parameter theta (scalar) of the Kimeldorf-Sampson copula derived from the estimation of Kendall's tau correlation matrix.

Format: theta = kimsamcopktauuniparam(U)

Input: U TxN matrix, values between 0 and 1.

Output: theta scalar, parameter.

Source: kimsamcop.src

Library: copulas.lcg

## kimsamcopktauuniparamgivkt

Purpose: Returns the parameter theta (scalar) of the Kimeldorf-Sampson copula derived from the estimation of Kendall's tau correlation matrix. The Kendall's correlation matrix must be given.

Format: theta = kimsamcopktauuniparamgivkt(ktau)

Input: ktau NxN matrix, Kendall's tau matrix.

Output: theta scalar, parameter.

Source: kimsamcop.src

Library: copulas.lcg

## kimsamcoploglik2

```
Purpose:    Returns the value of the log-likelihood-function
            of the bivariate Kimeldorf-Sampson copula.

Format:     l = kimsamcoploglik2(theta,U);

Input:      theta   scalar, parameter.

            U       Tx2 matrix of data.

Output:     l       Tx1 vector.

Source:     kimsamcop.src

Library:    copulas.lcg
```

## kimsamcoploglik3

```
Purpose:    Returns the value of the log-likelihood-function
            of the trivariate Kimeldorf-Sampson with parameter
            theta (scalar).

Format:     l = kimsamcoploglik3(theta,U);

Input:      theta   scalar, parameter.

            U       Tx3 matrix of data.

Output:     l       Tx1 vector.

Source:     kimsamcop.src

Library:    copulas.lcg
```

## kimsamcoploglik3multiparam

Purpose: Returns the value of the log-likelihood-function of the trivariate Kimeldorf-Sampson with parameter-vector theta.

Format: l = kimsamcoploglik3multiparam(theta,U);

Input: theta 2x1 vector, parameters.

U Tx3 matrix of data.

Output: l Tx1 vector.

Source: kimsamcop.src

Library: copulas.lcg

## kimsamcoploglik4

```
Purpose:    Returns the value of the log-likelihood-function
            of the 4-variate Kimeldorf-Sampson with parameter
            theta (scalar).

Format:     l = kimsamcoploglik4(theta,U);

Input:      theta   scalar, parameter.

            U       Tx4 matrix of data.

Output:     l       Tx1 vector.

Source:     kimsamcop.src

Library:    copulas.lcg
```

## kimsamcoploglik4multiparam

Purpose: Returns the value of the log-likelihood-function of the 4-variate Kimeldorf-Sampson with parameter-vector theta.

Format: l = kimsamcoploglik4multiparam(theta,U);

Input: theta 3x1 vector, parameters.

U Tx4 matrix of data.

Output: l Tx1 vector.

Source: kimsamcop.src

Library: copulas.lcg

## kimsamcoploglik5

```
Purpose:    Returns the value of the log-likelihood-function
            of the 5-variate Kimeldorf-Sampson with parameter
            theta (scalar).

Format:     l = kimsamcoploglik5(theta,U);

Input:      theta   scalar, parameter.

            U       Tx5 matrix of data.

Output:     l       Tx1 vector.

Source:     kimsamcop.src

Library:    copulas.lcg
```

## kimsamcoploglik5multiparam

Purpose: Returns the value of the log-likelihood-function of the 5-variate Kimeldorf-Sampson with parameter-vector theta.

Format: `l = kimsamcoploglik5multiparam(theta,U);`

Input:

| | | |
|---|---|---|
| theta | 4x1 | vector, parameters. |
| U | Tx5 | matrix of data. |

Output:

| | | |
|---|---|---|
| l | Tx1 | vector. |

Source: kimsamcop.src

Library: copulas.lcg

## kimsamcopmlnummultiparam

Purpose: Returns the parameter-vector theta of the Kimeldorf-Sampson copula estimated by the numerical maximum likelihood method.

Format: {idx,theta} = kimsamcopmlnummultiparam(U);

Input: U TxN matrix, values between 0 and 1.

Output: idx Nx1 vector, order of the columns of U that provides from all admissible solutions the one with the highest value of the log-likelihood-function.

theta (N-1)x1 vector, parameters.

Source: kimsamcop.src

Library: copulas.lcg

## kimsamcopmlnummultiparamgivkt

Purpose: Returns the parameter-vector theta of the Kimeldorf-Sampson copula estimated by the numerical maximum likelihood method. The Kendall's correlation matrix must be given.

Format: {idx,theta} = kimsamcopmlnummultiparamgivkt(U,ktau);

Input:

| | |
|---|---|
| U | TxN matrix, values between 0 and 1. |
| ktau | NxN matrix, Kendall's tau matrix. |

Output:

| | |
|---|---|
| idx | Nx1 vector, order of the columns of U that provides from all admissible solutions the one with the highest value of the log-likelihood-function. |
| theta | (N-1)x1 vector, parameters. |

Source: kimsamcop.src

Library: copulas.lcg

## kimsamcopmlnumuniparam

```
Purpose:    Returns the parameter theta (scalar) of the
            Kimeldorf-Sampson copula estimated by the
            numerical maximum likelihood method.

Format:     theta = kimsamcopmlnumuniparam(U);

Input:      U       TxN matrix, values between 0 and 1.

Output:     theta   scalar, parameter.

Source:     kimsamcop.src

Library:    copulas.lcg
```

## kimsamcopmlnumuniparamgivkt

```
Purpose:    Returns the parameter theta (scalar) of the
            Kimeldorf-Sampson copula estimated by the
            numerical maximum likelihood method. The
            Kendall's correlation matrix must be given.

Format:     theta = kimsamcopmlnumuniparamgivkt(U,ktau);

Input:      U       TxN matrix, values between 0 and 1.

            ktau    NxN matrix, Kendall's tau matrix.

Output:     theta   scalar, parameter.

Source:     kimsamcop.src

Library:    copulas.lcg
```

## kimsamcopsimu

```
Purpose:    Simulates values from a N-dimensional Kimeldorf-
            Sampson copula with parameter-vector theta.

Format:     U = kimsamcopsimu(T,theta);

Input:      T       scalar.

            theta   (N-1)x1 vector, parameters.

Output:     U       TxN matrix, values between 0 and 1.

Source:     kimsamcop.src

Library:    copulas.lcg
```

## kimsamcopsimugivenS

Purpose: Simulates values from a N-dimensional Kimeldorf-Sampson copula with parameter-vector theta. The needed random numbers must be given.

Format: U = kimsamcopsimugivenS(S,theta);

Input: S TxN matrix, U(0,1) random numbers.

theta (N-1)x1 vector, parameters.

Output: U TxN matrix, values between 0 and 1.

Source: kimsamcop.src

Library: copulas.lcg

## kimsaminvKnum

Purpose: Provides the function K of the Kimeldorf-Sampson copula in the form that is needed to calculate numerically the inverse of K.

Format: f = kimsaminvKnum(t);

Input: t Tx1 vector.

Output: f Tx1 vector.

Remarks: The known value y=K(t) must be given in a global variable named "yglob".

See also: invfunc, kimsaminvKstart

Source: kimsamcop.src

Library: copulas.lcg

## kimsaminvKstart

Purpose: Provides the startvalue to calculate numerically the inverse of K of the Kimeldorf-Sampson copula.

Format: f = kimsaminvKstart(y);

Input: y Tx1 vector.

Output: f Tx1 vector.

Remarks: y denotes the known value of the function K.

See also: invfunc, kimsaminvKnum

Source: kimsamcop.src

Library: copulas.lcg

## kimsaminvphi

Purpose: Computes the value of the inverse of the function phi of the Kimeldorf-Sampson copula with parameter theta.

Format: `f = kimsaminvphi(t);`

Input: `t` Tx1 vector.

Output: `f` Tx1 vector.

Remarks: The parameter theta must be given in a global variable named "thetaglob".

Source: kimsamcop.src

Library: copulas.lcg

## kimsamK

```
Purpose:    Computes the value of the function K of the
            Kimeldorf-Sampson copula with parameter theta.

Format:     f = kimsamK(t);

Input:      t   Tx1 vector.

Output:     f   Tx1 vector.

Remarks:    The parameter theta must be given in a global
            variable named "thetaglob".

Source:     kimsamcop.src

Library:    copulas.lcg
```

## kimsamphi

```
Purpose:    Computes the value of the function phi of the
            Kimeldorf-Sampson copula with parameter theta.

Format:     f = kimsamphi(t);

Input:      t   Tx1 vector.

Output:     f   Tx1 vector.

Remarks:    The parameter theta must be given in a global
            variable named "thetaglob".

Source:     kimsamcop.src

Library:    copulas.lcg
```

## kimsamprimephi

Purpose: Computes the value of the first derivative of the function phi of the Kimeldorf-Sampson copula with parameter theta.

Format: `f = kimsamprimephi(t);`

Input: `t` Tx1 vector.

Output: `f` Tx1 vector.

Remarks: The parameter theta must be given in a global variable named "thetaglob".

Source: kimsamcop.src

Library: copulas.lcg

## kolsmiLilliefors

```
Purpose:    Computes the Kolmogorov-Smirnov test for normality
            with mean and variance unknown. The null hypothesis
            H_0 is: x ~ N(mu,sigma^2). Since mu and sigma^2 have
            to be estimated from data, Lilliefors' critical
            values are used.

Format:     {teststat,critval,reject} = kolsmiLilliefors(x);

Input:      x           TxN matrix, data.

Output:     teststat    1xN vector, values of test statistic.

            critval     3x1 vector, Lilliefors' critical values
                        for levels of significance 0.01, 0.05,
                        0.1.

            reject      3xN matrix, 1 if H_0 is rejected, 0 if
                        H_0 is not rejected.

Example:    u = rndn(100,1);
            v = rndu(100,1);
            x = u ~ v;
            {teststat,critval,reject} = kolsmiLilliefors(x);

            teststat = 0.044044955      0.089616820

                       0.10310000
            critval =  0.08860000
                       0.08050000

                       0.00000000       0.0000000
            reject =   0.00000000       1.0000000
                       0.00000000       1.0000000

See also:   chi2n

Source:     distrtest.src

Library:    copulautils.lcg
```

## logreturn

```
Purpose:    Computes log-returns (geometric returns) from prices.

Format:     logret = logreturn(price,hp);

Input:      price   TxN matrix, prices.

            hp      scalar, holding period.

Output:     sret    (T-hp)xN matrix, log-returns.

Example:    price = {2293 2920,
                     2312 2917,
                     2332 2968};
            logret = logreturn(price,1);

            logret = 0.00825   -0.00103
                     0.00861    0.01733

Source:     return.src

Library:    copulautils.lcg
```

# LPM

```
Purpose:    Provides the LPM (lower partial moment) for a
            given threshold value and a given exponent.

Format:     l = LPM(profit,z,k);

Input:      profit      Tx1 vector, profits (= -losses).

            z           1xK vector, threshold values.

            k           1xK vector, exponents.

Output:     l           1xK vector, LPM's.

Example:    loss = {0, 2.5, -3.5};
            z = 0 ~ meanc(-loss);
            k = {2 2};
            l = LPM(-loss,z,k);

            l = 2.0833333        2.7129630

Source:     simenv.src

Library:    copulautils.lcg
```

## maxmultipletoloss

```
Purpose:    Provides the maximum loss in relation to VaR.

Format:     multiple = maxmultipletoloss(VaR,loss);

Input:      VaR             TxL matrix, VaR's for L
                            different confidence levels.

            loss            Tx1 vector, losses.

Output:     multiple        1xL vector, multiples.

Example:    V1 = {10,11,10,12,10,13,11,14,15,10,12};
            V2 = {10,11,10,12,10,13,11,14,17,13,15};
            VaR = V1 ~ V2;
            loss = {10,10,9,11,9,11,10,15,16,12,14};
            multiple = maxmultipletoloss(VaR,loss);

            multiple = 1.2000000        1.0714286

Source:     backtestenv.src

Library:    copulautils.lcg
```

## meanmultipletoloss

```
Purpose:    Provides the average uncovered loss in relation
            to VaR.

Format:     multiple = meanmultipletoloss(VaR,loss,confidence);

Input:      VaR             TxL matrix, VaR's for L
                            different confidence levels.

            loss            Tx1 vector, losses.

            confidence      1xL vector, confidence levels
                            that correspond to the VaR's.

Output:     multiple        1xL vector, multiples.

Example:    V1 = {10,11,10,12,10,13,11,14,15,10,12};
            V2 = {10,11,10,12,10,13,11,14,17,13,15};
            VaR = V1 ~ V2;
            loss = {10,10,9,11,9,11,10,15,16,12,14};
            confidence = {0.8 0.8};
            multiple = meanmultipletoloss(VaR,loss,confidence);

            multiple = 1.1460317        1.0042017

Source:     backtestenv.src

Library:    copulautils.lcg
```

# MRB

```
Purpose:    Provides the Mean Relative Bias. Compares different
            models with the mean of all models.

Format:     M = MRB(VaR);

Input:      VaR       TxL matrix, VaR's from different models.

Output:     M         1xL vector, MRB's for different models.

Example:    VaR = { 3 4 5,
                    2 4 5 };
            M = MRB(VaR);

            M = -0.35227273      0.045454545       0.30681818

Source:     backtestenv.src

Library:    copulautils.lcg
```

## MRBtocoverage

Purpose: Provides the Mean Relative Bias scaled to the desired level of coverage. Compares different models with the mean of all models.

Format: `Mscal = MRBtocoverage(VaR,loss,confidence);`

Input:

| | |
|---|---|
| VaR | TxL matrix, VaR's for L different confidence levels. |
| loss | Tx1 vector, losses. |
| confidence | 1xL vector, confidence levels that correspond to the VaR's. |

Output:

| | |
|---|---|
| Mscal | 1xL vector, scaled MRB's for different models. |

Example:

```
V1 = {10,11,10,12,10,13,11,14,15,10,12};
V2 = {10,11,10,12,10,13,11,14,17,13,15};
VaR = V1 ~ V2;
loss = {10,10,9,11,9,11,10,15,16,12,14};
confidence = {0.8 0.8};
Mscal = MRBtocoverage(VaR,loss,confidence);
```

```
Mscal = 0.036997038     -0.036997038
```

Source: backtestenv.src

Library: copulautils.lcg

## msqdtov

```
Purpose:    Transforms the diagonal and the upper triangle
            of a square matrix into a vector.

Format:     v = msqdtov(m);

Input:      m   NxN matrix.

Output:     v   ((N^2 + N) / 2)x1 vector.

Example:    m = {1 2 3,
                 2 4 5,
                 3 5 6};
            v = msqdtov(m);

                1
                2
            v = 3
                4
                5
                6

See also:   vtomsymd, msqtov, vtomsym

Source:     vmtomv.src

Library:    copulautils.lcg
```

## msqtov

```
Purpose:    Transforms the upper triangle of a square matrix
            into a vector.

Format:     v = msqtov(m);

Input:      m   NxN matrix.

Output:     v   ((N^2 - N) / 2)x1 vector.

Example:    m = {1 1 2 3,
                 1 1 4 5,
                 2 4 1 6,
                 3 5 6 1};
            v = msqtov(m);

                1
                2
            v = 3
                4
                5
                6

See also:   vtomsym, msqdtov, vtomsymd,

Source:     vmtomv.src

Library:    copulautils.lcg
```

## multipletocoverage

```
Purpose:    Provides the multiple that is needed for each VaR
            to attain the desired level of coverage.

Format:     multiple = multipletocoverage(VaR,loss,confidence);

Input:      VaR             TxL matrix, VaR's for L
                            different confidence levels.

            loss            Tx1 vector, losses.

            confidence      1xL vector, confidence levels
                            that correspond to the VaR's.

Output:     multiple        1xL vector, multiples.

Example:    V1 = {10,11,10,12,10,13,11,14,15,10,12};
            V2 = {10,11,10,12,10,13,11,14,17,13,15};
            VaR = V1 ~ V2;
            loss = {10,10,9,11,9,11,10,15,16,12,14};
            confidence = {0.8 0.8};
            multiple = multipletocoverage(VaR,loss,confidence);

            multiple = 1.0714286       0.94117647

Source:     backtestenv.src

Library:    copulautils.lcg
```

## nelsencopktaumultiparam

```
Purpose:    Returns the order of variables and the
            parameter-vector theta of the Nelsen copula
            derived from the estimation of Kendall's
            tau correlation matrix.

Format:     {idx,theta} = nelsencopktaumultiparam(U);

Input:      U       TxN matrix, values between 0 and 1.

Output:     idx     Nx1 vector, order of the columns of U
                    that provides from all admissible solutions
                    the one with the smallest distance to the
                    target value in the minimization problem
                    to solve.

            theta   (N-1)x1 vector, parameters.

Source:     nelsencop.src

Library:    copulas.lcg
```

## nelsencopktaumultiparamgivkt

Purpose: Returns the order of variables and the parameter-vector theta of the Nelsen copula derived from the estimation of Kendall's tau correlation matrix. The Kendall's correlation matrix must be given.

Format: {idx,theta} = nelsencopktaumultiparamgivkt(U,ktau);

Input:

| | |
|---|---|
| U | TxN matrix, values between 0 and 1. |
| ktau | NxN matrix, Kendall's tau matrix. |

Output:

| | |
|---|---|
| idx | Nx1 vector, order of the columns of U that provides from all admissible solutions the one with the smallest distance to the target value in the minimization problem to solve. |
| theta | (N-1)x1 vector, parameters. |

Source: nelsencop.src

Library: copulas.lcg

## nelsencopktauuniparam

```
Purpose:    Returns the parameter theta (scalar) of the Nelsen
            copula derived from the estimation of Kendall's
            tau correlation matrix.

Format:     theta = nelsencopktauuniparam(U)

Input:      U       TxN matrix, values between 0 and 1.

Output:     theta   scalar, parameter.

Source:     nelsencop.src

Library:    copulas.lcg
```

## nelsencopktauuniparamgivkt

Purpose: Returns the parameter theta (scalar) of the Nelsen copula derived from the estimation of Kendall's tau correlation matrix. The Kendall's correlation matrix must be given.

Format: theta = nelsencopktauuniparamgivkt(ktau)

Input: ktau NxN matrix, Kendall's tau matrix.

Output: theta scalar, parameter.

Source: nelsencop.src

Library: copulas.lcg

## nelsencoploglik2

```
Purpose:    Returns the value of the log-likelihood-function
            of the bivariate Nelsen copula.

Format:     l = nelsencoploglik2(theta,U);

Input:      theta   scalar, parameter.

            U       Tx2 matrix of data.

Output:     l       Tx1 vector.

Source:     nelsencop.src

Library:    copulas.lcg
```

## nelsencoploglik3

```
Purpose:    Returns the value of the log-likelihood-function
            of the trivariate Nelsen copula with parameter
            theta (scalar).

Format:     l = nelsencoploglik3(theta,U);

Input:      theta   scalar, parameter.

            U       Tx3 matrix of data.

Output:     l       Tx1 vector.

Source:     nelsencop.src

Library:    copulas.lcg
```

## nelsencoploglik3multiparam

Purpose: Returns the value of the log-likelihood-function of the trivariate Nelsen copula with parameter-vector theta.

Format: l = nelsencoploglik3multiparam(theta,U);

Input: theta 2x1 vector, parameters.

U Tx3 matrix of data.

Output: l Tx1 vector.

Source: nelsencop.src

Library: copulas.lcg

## nelsencoploglik4

Purpose: Returns the value of the log-likelihood-function of the 4-variate Nelsen copula with parameter theta (scalar).

Format: `l = nelsencoploglik4(theta,U);`

Input:

| | | |
|---|---|---|
| theta | scalar, parameter. |
| U | Tx4 matrix of data. |

Output:

| | |
|---|---|
| l | Tx1 vector. |

Source: nelsencop.src

Library: copulas.lcg

## nelsencoploglik4multiparam

Purpose: Returns the value of the log-likelihood-function of the 4-variate Nelsen copula with parameter-vector theta.

Format: `l = nelsencoploglik4multiparam(theta,U);`

Input:

| | | |
|---|---|---|
| theta | 3x1 | vector, parameters. |
| U | Tx4 | matrix of data. |

Output:

| | | |
|---|---|---|
| l | Tx1 | vector. |

Source: nelsencop.src

Library: copulas.lcg

## nelsencoploglik5

```
Purpose:    Returns the value of the log-likelihood-function
            of the 5-variate Nelsen copula with parameter
            theta (scalar).

Format:     l = nelsencoploglik5(theta,U);

Input:      theta   scalar, parameter.

            U       Tx5 matrix of data.

Output:     l       Tx1 vector.

Source:     nelsencop.src

Library:    copulas.lcg
```

## nelsencoploglik5multiparam

Purpose: Returns the value of the log-likelihood-function of the 5-variate nelsen copula with parameter-vector theta.

Format: `l = nelsencoploglik5multiparam(theta,U);`

Input:

| | | |
|---|---|---|
| theta | 4x1 | vector, parameters. |
| U | Tx5 | matrix of data. |

Output:

| | | |
|---|---|---|
| l | Tx1 | vector. |

Source: nelsencop.src

Library: copulas.lcg

## nelsencopmlnummultiparam

Purpose: Returns the parameter-vector theta of the Nelsen copula estimated by thenumerical maximum likelihood method.

Format: {idx,theta} = nelsencopmlnummultiparam(U);

Input: U TxN matrix, values between 0 and 1.

Output: idx Nx1 vector, order of the columns of U that provides from all admissible solutions the one with the highest value of the log-likelihood-function.

theta (N-1)x1 vector, parameters.

Source: nelsencop.src

Library: copulas.lcg

## nelsencopmlnummultiparamgivkt

Purpose: Returns the parameter-vector theta of the Nelsen copula estimated by the numerical maximum likelihood method. The Kendall's correlation matrix must be given.

Format: `{idx,theta} = nelsencopmlnummultiparamgivkt(U,ktau);`

Input:

| | |
|---|---|
| U | TxN matrix, values between 0 and 1. |
| ktau | NxN matrix, Kendall's tau matrix. |

Output:

| | |
|---|---|
| idx | Nx1 vector, order of the columns of U that provides from all admissible solutions the one with the highest value of the log-likelihood-function. |
| theta | (N-1)x1 vector, parameters. |

Source: nelsencop.src

Library: copulas.lcg

## nelsencopmlnumuniparam

Purpose: Returns the parameter theta (scalar) of the Nelsen copula estimated by the numerical maximum likelihood method.

Format: `theta = nelsencopmlnumuniparam(U);`

Input: `U` TxN matrix, values between 0 and 1.

Output: `theta` scalar, parameter.

Source: nelsencop.src

Library: copulas.lcg

## nelsencopmlnumuniparamgivkt

Purpose: Returns the parameter theta (scalar) of the Nelsen copula estimated by the numerical maximum likelihood method. The Kendall's correlation matrix must be given.

Format: theta = nelsencopmlnumuniparamgivkt(U,ktau);

Input: U TxN matrix, values between 0 and 1.

ktau NxN matrix, Kendall's tau matrix.

Output: theta scalar, parameter.

Source: nelsencop.src

Library: copulas.lcg

## nelsencopsimu

```
Purpose:    Simulates values from a N-dimensional Nelsen copula
            with parameter-vector theta.

Format:     U = nelsencopsimu(T,theta);

Input:      T       scalar.

            theta   (N-1)x1 vector, parameters.

Output:     U       TxN matrix, values between 0 and 1.

Source:     nelsencop.src

Library:    copulas.lcg
```

## nelsencopsimugivenS

```
Purpose:    Simulates values from a N-dimensional Nelsen copula
            with parameter-vector theta. The needed random
            numbers must be given.

Format:     U = nelsencopsimugivenS(S,theta);

Input:      S       TxN matrix, U(0,1) random numbers.

            theta   (N-1)x1 vector, parameters.

Output:     U       TxN matrix, values between 0 and 1.

Source:     nelsencop.src

Library:    copulas.lcg
```

## nelseninvK

```
Purpose:    Computes the value of the inverse of the function
            K of the Nelsen copula with parameter theta.

Format:     f = nelseninvK(t);

Input:      t   Tx1 vector.

Output:     f   Tx1 vector.

Remarks:    The parameter theta must be given in a global
            variable named "thetaglob".

Source:     nelsencop.src

Library:    copulas.lcg
```

## nelseninvphi

Purpose: Computes the value of the inverse of the function phi of the Nelsen copula with parameter theta.

Format: f = nelseninvphi(t);

Input: t Tx1 vector.

Output: f Tx1 vector.

Remarks: The parameter theta must be given in a global variable named "thetaglob".

Source: nelsencop.src

Library: copulas.lcg

## nelsenK

Purpose: Computes the value of the function K of the Nelsen copula with parameter theta.

Format: f = nelsenK(t);

Input: t Tx1 vector.

Output: f Tx1 vector.

Remarks: The parameter theta must be given in a global variable named "thetaglob".

Source: nelsencop.src

Library: copulas.lcg

## nelsenphi

```
Purpose:    Computes the value of the function phi of the
            Nelsen copula with parameter theta.

Format:     f = nelsenphi(t);

Input:      t   Tx1 vector.

Output:     f   Tx1 vector.

Remarks:    The parameter theta must be given in a global
            variable named "thetaglob".

Source:     nelsencop.src

Library:    copulas.lcg
```

## nelsenprimephi

Purpose: Computes the value of the first derivative of the function phi of the Nelsen copula with parameter theta.

Format: f = nelsenprimephi(t);

Input: t Tx1 vector.

Output: f Tx1 vector.

Remarks: The parameter theta must be given in a global variable named "thetaglob".

Source: nelsencop.src

Library: copulas.lcg

## normalcoploglik

Purpose: Returns the value of the log-likelihood-function of the normal copula.

Format: `l = normalcoploglik(r,U);`

Input:

| | |
|---|---|
| `r` | scalar or NxN matrix, correlation coefficient or correlation matrix. |
| `U` | TxN matrix of data. |

Output:

| | |
|---|---|
| `l` | scalar. |

See also: normalcopmlnum

Source: normalcop.src

Library: copulas.lcg

## normalcopmlana

Purpose: Returns the correlation matrix of the normal copula estimated by analytical maximum likelihood.

Format: R = normalcopmlana(U);

Input: U TxN matrix, values between 0 and 1.

Output: R NxN matrix, correlation matrix.

See also: normalcopmlnum

Source: normalcop.src

Library: copulas.lcg

## normalcopmlnum

Purpose: Returns the correlation matrix of the normal copula estimated by the numerical maximum likelihood method.

Format: R = normalcopmlnum(U);

Input: U TxN matrix, values between 0 and 1.

Output: R NxN matrix, correlation matrix.

See also: normalcoploglik, normalcopmlana

Source: normalcop.src

Library: copulas.lcg

## normalcopsimu

```
Purpose:    Simulates values from a N-dimensional normal copula
            with correlation matrix R.

Format:     U = normalcopsimu(T,R);

Input:      T   scalar.

            R   NxN matrix, correlation matrix.

Output:     U   TxN matrix, values between 0 and 1.

Source:     normalcop.src

Library:    copulas.lcg
```

## normalcopsimugivenZ

```
Purpose:    Simulates values from a N-dimensional normal copula
            with correlation matrix R. The needed random numbers
            must be given.

Format:     U = normalcopsimugivenZ(Z,R);

Input:      Z   TxN matrix, standardnormal random numbers.

            R   NxN matrix, correlation matrix.

Output:     U   TxN matrix, values between 0 and 1.

Source:     normalcop.src

Library:    copulas.lcg
```

## quadeq

```
Purpose:    Computes the two solutions of a quadratic equation
            ax^2 + bx + c = 0.

Format:     {x1,x2} = quadeq(a,b,c);

Input:      a       Tx1 vector, coefficients (different from 0).

            b       Tx1 vector, coefficients.

            c       Tx1 vector, coefficients.

Output:     x1      Tx1 vector, first solution for x.

            x2      Tx1 vector, second solution for x.

Example:    a = 1|1;
            b = -1|-1;
            c = (-12)|(-6);
            {x1,x2} = quadeq(a,b,c);

            x1 = 4.0000000
                 3.0000000

            x2 = -3.0000000
                 -2.0000000

Source:     quadeq.src

Library:    copulautils.lcg
```

## quantileorderedsample

```
Purpose:    Computes quantiles from data given specified
            probabilities (estimation from ordered sample).

Format:     y = quantileorderedsample(x,e);

Input:      x   Tx1 vector of data.

            e   Lx1 vector, quantile levels or probabilities.

Output:     y   Lx1 vector, quantiles.

Example:    x = {17,12,14,18,11,13,15,16,10};
            e = {0.8,0.9};
            y = quantileorderedsample(x,e);

            y = 17  18

Source:     quantilealternatives.src

Library:    copulautils.lcg
```

## rchisquare

Purpose: Generates random variates of the chi-square distribution.

Format: x = rchisquare(R,C,n)

Input:
- R scalar, number of rows of random variates.
- C scalar, number of columns of random variates.
- n scalar, degrees of freedom.

Output:
- x RxC matrix, random numbers.

Remarks:

Version 1.0 16/05/2000

Source: rchisquare.src

Library: copulautils.lcg

## resorter

```
Purpose:    Reverses the sorting of the columns of a matrix
            according to a given index.

Format:     Y = resorter(idx,X);

Input:      idx         Nx1 vector, index.

            X           TxN matrix, sorted data matrix.

Output:     Y           TxN matrix, original data matrix.

Example:    O = {1 2 3, 1 2 3};
            idx = {2,3,1};

            X = O[.,idx];
            X = 2.0000000        3.0000000        1.0000000
                2.0000000        3.0000000        1.0000000

            Y = resorter(idx,X);
            Y = 1.0000000        2.0000000        3.0000000
                1.0000000        2.0000000        3.0000000

Source:     resorter.src

Library:    copulautils.lcg
```

## simpfloss

Purpose: Computes the loss of a portfolio of different assets between the last known prices and simulated prices.

Format: spfl = simpfloss(assetweights,lastprice,simret);

Input:

| | | |
|---|---|---|
| assetweights | 1xN vector, number of every type of asset in the portfolio. |
| lastprice | 1xN vector, last known prices. |
| simret | TxN vector, simulated "future" returns. |

Output:

| | |
|---|---|
| spfl | Tx1 vector, simulated portfoliolosses. |

Example:

```
assetweights = {1 1};
lastprice = {50 100};
simret = {0.06 0.04,
          0.04 0.03,
          0.02 0.03};
spfl = simpfloss(assetweights,lastprice,simret);

       -7.0000000
spfl = -5.0000000
       -4.0000000
```

Source: simenv.src

Library: copulautils.lcg

## simpfretandloss

```
Purpose:    Computes the return and the loss of a portfolio of
            different assets between the last known prices and
            simulated prices.

Format:     {spfr,spfl} = simpfretandloss(assetweights,lastprice,
                          simret);

Input:      assetweights    1xN vector, number of every type of
                            asset in the portfolio.

            lastprice       1xN vector, last known prices.

            simret          TxN vector, simulated "future" returns.

Output:     spfr            Tx1 vector, simulated portfolioreturns.

            spfl            Tx1 vector, simulated portfoliolosses.

Example:    assetweights = {1 1};
            lastprice = {50 100};
            simret = {0.06 0.04,
                      0.04 0.03,
                      0.02 0.03};
            {spfr,spfl} = simpfretandloss(assetweights,lastprice,
                          simret);

                    0.046666667
            spfr =  0.033333333
                    0.026666667

                   -7.0000000
            spfl = -5.0000000
                   -4.0000000

Source:     simenv.src

Library:    copulautils.lcg
```

## spearmansrho

```
Purpose:    Computes Spearman's rho for two vectors.

Format:     r = spearmansrho(x,y);

Input:      x   Tx1 vector.

            y   Tx1 vector.

Output:     r   scalar, Spearman's rho.

Example:    x = {76 44 32 53 25 58 26 59 29 65};
            y = {122 67 68 101 42 59 118 79 83 89};
            r = spearmansrho(x',y');

            r = 0.36969697

See also:   spearmansrhomatrix, kendallstau, kendallstaumatrix

Source:     spearmansrho.src

Library:    copulautils.lcg
```

## spearmansrhomatrix

```
Purpose:    Computes a Spearman's rho correlation matrix.

Format:     r = spearmansrhomatrix(Z);

Input:      Z   TxN matrix.

Output:     r   symmetric NxN matrix with ones in the diagonal
                and the pairwise Spearman's rho in the non-
                diagonal.

Example:    x={ 54.10  54.50  51.60  54.25 054.25 054.45
                53.10  53.20};
            y={ 157.50 158.50 168.00 165.75 160.00 160.00
                163.25 166.25};
            Z=x'~y';
            r = spearmansrhomatrix(Z);

            r = 1.0000000       0.36969697
                0.36969697      1.0000000

See also:   spearmansrho, kendallstau, kendallstaumatrix

Source:     spearmansrho.src

Library:    copulautils.lcg
```

## sreturn

```
Purpose:    Computes arithmetic returns from prices.

Format:     sret = sreturn(price,hp);

Input:      price   TxN matrix, prices.

            hp      scalar, holding period.

Output:     sret    (T-hp)xN matrix, arithmetic returns.

Example:    price = {2293 2920,
                     2312 2917,
                     2332 2968};
            sret = sreturn(price,1);

            sret = 0.00829   -0.00103
                   0.00865    0.01748

Source:     return.src

Library:    copulautils.lcg
```

## tcopktau

Purpose: Returns the correlation matrix and the degree of freedom of the t-copula derived from the estimation of Kendall's tau correlation matrix.

Format: {R,nu}=tcopktau(U);

Input: U TxN matrix, values between 0 and 1.

Output: R NxN matrix, correlation matrix.

nu scalar, degree of freedom.

See also: tcopmlnum, tcopnormalapprox

Source: tcop.src

Library: copulas.lcg

## tcoploglik

```
Purpose:    Returns the value of the log-likelihood-function
            of the t-copula.

Format:     l = tcoploglik(Z,R,nu);

Input:      Z   TxN matrix, transformed data.

            R   NxN matrix, correlation matrix.

            nu  scalar, degrees of freedom.

Output:     l   scalar.

See also:   tcopmlnum

Source:     tcop.src

Library:    copulas.lcg
```

## tcoploglikscalrfull

```
Purpose:    Returns the value of the log-likelihood-function
            of the t-copula. The value of the degrees of
            freedom must be given in a global variable
            named "nuglob".

Format:     l = tcoploglikscalrfull(r,Z);

Input:      r   scalar, correlation coefficient.

            Z   Tx2 matrix, transformed data.

Output:     l   scalar.

See also:   tcopmlnum

Source:     tcop.src

Library:    copulas.lcg
```

## tcoploglikscalrgivennu

Purpose: Returns the value of the log-likelihood-function of a short form of the t-copula (only terms of t-copula containing the correlation matrix R). The value of the degrees of freedom must be given in a global variable named "nuglob".

Format: l = tcoploglikscalrgivennu(r,Z);

Input: r scalar, correlation coefficient.

Z Tx2 matrix, transformed data.

Output: l scalar.

See also: tcopmlnum

Source: tcop.src

Library: copulas.lcg

## tcopmlnum

Purpose: Returns the correlation matrix and the degree of freedom of the t-copula estimated by the numerical maximum likelihood method.

Format: {R,nu}=tcopmlnum(U);

Input: U TxN matrix, values between 0 and 1.

Output: R NxN matrix, correlation matrix.

nu scalar, degree of freedom.

See also: tcoploglik, tcoploglikscalrgivennu, tcoploglikscalrfull

Source: tcop.src

Library: copulas.lcg

## tcopsimu

Purpose: Simulates values from a N-dimensional t-copula with correlation matrix R and nu degrees of freedom.

Format: U = tcopsimu(T,R,nu);

Input: T scalar.

R NxN matrix, correlation matrix.

nu scalar, degrees of freedom.

Output: U TxN matrix, values between 0 and 1.

Source: tcop.src

Library: copulas.lcg

## tcopsimugivenZ

Purpose: Simulates values from a N-dimensional t-copula with correlation matrix R and nu degrees of freedom. The needed random numbers must be given.

Format: `U = tcopsimugivenZ(Z,R,nu);`

Input:

| | |
|---|---|
| Z | TxN matrix, standardnormal random numbers. |
| R | NxN matrix, correlation matrix. |
| nu | scalar, degrees of freedom. |

Output:

| | |
|---|---|
| U | TxN matrix, values between 0 and 1. |

Source: tcop.src

Library: copulas.lcg

## VaR

```
Purpose:    Provides the VaR for different confidence levels.

Format:     V = VaR(loss,confidence);

Input:      loss            Tx1 vector, losses.

            confidence      1xL vector, confidence levels.

Output:     V               1xL vector, VaR's.

Example:    loss = {-7, -5, -4};
            confidence = {0.99 0.5};
            V = VaR(assetweights,lastprice,simret,confidence);

            V = -4.0000000       -5.0000000

See also:   quantileorderedsample

Source:     simenv.src

Library:    copulautils.lcg
```

## vtomsym

```
Purpose:    Transforms a vector into a symmetric matrix with
            ones in the diagonal.

Format:     m = vtomsym(v);

Input:      v   ((N^2 - N) / 2)x1 vector.

Output:     m   symmetric NxN matrix with ones in the diagonal
                and the values of v in the non-diagonal.

Example:    v = {1,2,3,4,5,6};
            m = vtomsym(v);

                1 1 2 3
            m = 1 1 4 5
                2 4 1 6
                3 5 6 1

See also:   msqtov, vtomsymd, msqdtov

Source:     vmtomv.src

Library:    copulautils.lcg
```

## vtomsymd

```
Purpose:    Transforms a vector into a symmetric matrix.

Format:     m = vtomsymd(v);

Input:      v   ((N^2 + N) / 2)x1 vector.

Output:     m   symmetric NxN matrix. All the values of m,
                diagonal included, are given in v.

Example:    v = {1,2,3,4,5,6};
            m = vtomsymd(v);

                1 2 3
            m = 2 4 5
                3 5 6

See also:   msqdtov, vtomsym, msqtov

Source:     vmtomv.src

Library:    copulautils.lcg
```

## ww

```
Purpose:    Computes the Wald-Wolfowitz test statistic.
            The null hypothesis H_0 is: x's are iid.
            (see Mittelhammer, pp. 663)

Format:     z = ww(x);

Input:      x           TxN matrix, data.

Output:     z           1xN vector, values of test statistic.
                        Follows a standard normal distribution
                        under H_0.

Example:    x = {1.37,1.96,0.74,0.42,0.12,0.61,1.98,1.76,1.73,
                 3.32,1.44,2.46,0.34,2.31,2.14,2.11,2.84,2.47,
                 1.25,0.66,2.23,1.11,1.73,0.26,1.77,1.35,2.91,
                 0.93,1.50,2.72,1.73,0.59,0.36,0.24,2.68,0.30,
                 0.10,2.75,1.68,0.88};
            z = ww(x);

            z = 0.64072328

Source:     iidtest.src

Library:    copulautils.lcg
```