Package 'fMultivar'

September 29, 2009

Version 2100.76

Revision 4277

Date 2009-09-28

Title Multivariate Market Analysis

Author Diethelm Wuertz and many others, see the SOURCE file

Depends R (>= 2.4.0), methods, sn, timeDate, timeSeries, fBasics

Suggests RUnit, tcltk

Maintainer Rmetrics Core Team <Rmetrics-core@r-project.org>

Description Environment for teaching "Financial Engineering and Computational Finance"

NOTE SEVERAL PARTS ARE STILL PRELIMINARY AND MAY BE CHANGED IN THE FUTURE. THIS TYPICALLY INCLUDES FUNCTION AND ARGUMENT NAMES, AS WELL AS DEFAULTS FOR ARGUMENTS AND RETURN VALUES.

LazyLoad yes

LazyData yes

License GPL (>= 2)

URL http://www.rmetrics.org

Repository CRAN

Date/Publication 2009-09-29 18:44:17

R topics documented:

`BivariateBinning` *Square and Hexagonal Data Binning*

Description

A collection and description of functions which allow to create histograms due to sqaure and hexagonal binning.

Bivariate Binning Functions:

`squareBinning`	does a square binning of data points,
`hexBinning`	does a hexagonal binning of data points

Usage

```
squareBinning(x, y = NULL, bins = 30)
hexBinning(x, y = NULL, bins = 30)

## S3 method for class 'squareBinning':
plot(x, col = heat.colors(12), addPoints = TRUE,
    addRug = TRUE, ...)
## S3 method for class 'hexBinning':
plot(x, col = heat.colors(12), addPoints = TRUE,
    addRug = TRUE, ...)
```

Arguments

`addPoints`	a logical flag, should the center of mass points added to the plot?
`addRug`	a logical flag, should a rug representation be added to the plot, for details see the function `rug`.
`bins`	an integer specifying the number of bins.
`col`	color map like for the `image` function.
`x, y`	[squareBinning][hexBinning] - either two numeric vectors of equal length or if `y` is NULL, a list with entries

x, y, or named data frame with x in the first and y in the second column. Note, timeSeries objects are also allowed as input.
[plot] -
an object of class squareBinning or hexBinning.

... arguments to be passed.

Value

A list with three entries, x, y and z, specified by an oject of class squareBinning or hexBinning.

Note, the returned value, can be directly used by the persp() and contour 3D plotting functions.

Author(s)

Diethelm Wuertz for the Rmetrics R-port.

Examples

```
## squareBinning -
   sB = squareBinning(x = rnorm(1000), y = rnorm(1000))
   plot(sB)

## hexBinning -
   hB = hexBinning(x = rnorm(1000), y = rnorm(1000))
   plot(hB)
```

BivariateGridding *Bivariate Gridded Data Sets*

Description

A collection and description of functions which allow to generate bivariate gridded data sets.

Grid Data Functions:

gridData	generates a grid data set of class 'gridData',
persp	generates a perspective plot from a grid data set,
contour	generates a contour plot from a grid data set.

Usage

```
gridData(x = (-10:10)/10, y = x, z = outer(x, y, function(x, y) (x^2+y^2) ) )

## S3 method for class 'gridData':
persp(x, theta = -40, phi = 30, col = "steelblue",
    ticktype = "detailed", ...)
## S3 method for class 'gridData':
```

```
contour(x, addImage = TRUE, ...)
```

Arguments

`addImage`	[contour] - a logical flag indicating if an image plot should be underlayed to the contour level plot.
`x, y, z`	[gridData] - `x` and `y` are two numeric vectors of grid pounts and `z` is a numeric matrix or any other rectangular object which can be transformed by the function `as.matrix` into a matrix object.
`theta, phi, col, ticktype`	[persp] - tailored parameters passed the perspective plot function `persp`.
`...`	[contour][persp] - additional arguments to be passed to the perspectice and countour plot functions.

Value

`gridData` -
A list with at least three entries, `x`, `y` and `z`.

The returned values, can be directly used by the `persp.gridData()` and `contour.gridData` 3D plotting methods.

Author(s)

Diethelm Wuertz for the Rmetrics R-port,
H. Akima for the Fortran Code of the Akima spline interpolation routine.

Examples

```
## gridData -
   # Grid Data Set
   gD = gridData()
   persp(gD)
   contour(gD)
```

---

cauchy2d *Bivariate Cauchy Distribution*

Description

Density, distribution function, and random generation for the bivariate Cauchy distribution.

Usage

```
pcauchy2d(x, y = x, rho = 0)
dcauchy2d(x, y = x, rho = 0)
rcauchy2d(n, rho = 0)
```

Arguments

n	the number of random deviates to be generated, an integer value.
rho	the correlation parameter, a numeric value ranging between minus one and one, by default zero.
x, y	two numeric vectors defining the x and y coordinates.

Value

`pcauchy2d`
returns a two column matrix of probabilities for the bivariate Cauchy distribution function.

`dcauchy2d`
returns a two column matrix of densities for the bivariate Cauchy distribution function.

`rcauchy2d`
returns a two column matrix of random deviates generated from the bivariate Cauchy distribution function.

Author(s)

Adelchi Azzalini for the underlying pnorm2d function,
Diethelm Wuertz for the Rmetrics R-port.

References

Azzalini A., (2004); *The sn Package*; R Reference Guide available from www.r-project.org.

Venables W.N., Ripley B.D., (2002); *Modern Applied Statistics with S*, Fourth Edition, Springer.

Examples

```
## Bivariate Cauchy Density:
   x = (-40:40)/10
   X = grid2d(x)
   z = dcauchy2d(X$x, X$y, rho = 0.5)
   Z = list(x = x, y = x, z = matrix(z, ncol = length(x)))
   persp(Z, theta = -40, phi = 30, col = "steelblue")
```

`density2d` *Bivariate Density Tools*

Description

Grid generator, kernel density estimator, histogram counter, and integrator for bivariate distributions

Usage

```
grid2d(x = (0:10)/10, y = x)
density2d(x, y = NULL, n = 20, h = NULL, limits = c(range(x), range(y)))
hist2d(x, y = NULL, n = c(20, 20))
integrate2d(fun, error = 1.0e-5, ...)
```

Arguments

`error`	the error bound to be achieved by the integration formula. A numeric value.
`fun`	the function to be integrated. The first argument requests the x values, the second the y values, and the remaining are reserved for additional parameters. The integration is over the unit square "[0,1]\textasciicircum2".
`h`	a vector of bandwidths for `x` and `y` directions. Defaults to normal reference bandwidth.
`limits`	the limits of the rectangle covered by the grid.
`n`	n - an integer specifying the number of grid points in each direction. The default value is 20. [hist2D] - In this case `n` may be a scalar or a two element vector. The default value is 20. [rnorm2d] - the number of random deviates to be generated, an integer value.
`x, y`	two numeric vectors defining the x and y coordinates. [density2D][hist2D] - two vectors of coordinates of data. If `y` is NULL then `x` is assumed to be a two column matrix, where the first column contains the `x` data, and the second column the `y` data.
`...`	parameters passed to the function to be integrated.

Value

`grid2d`
returns a list with two vectors named `$x` and `$y` spanning the grid defined by the coordinate vectors `x` and `y`.

`density2d`
`hist2d`
returns a list with three elements `$x`, `$y`, and `$z`. `x` and `y` are vectors spanning the two dimensional

grid and z the corresponding matrix. The output can directly serve as input to the plotting functions image, contour and persp.

integrate2d
returns a list with the $value of the integral over the unit square [0,1]\textasciicircum2, an $error estimate and the number of grid $points used by the integration function.

Author(s)

W.N. Venables and B.D. Ripley for the underlying kde2d function,
Gregory R. Warnes for the underlying hist2d function,
Diethelm Wuertz for the Rmetrics R-port.

References

Azzalini A., (2004); *The sn Package*; R Reference Guide available from www.r-project.org.

Venables W.N., Ripley B.D., (2002); *Modern Applied Statistics with S*, Fourth Edition, Springer.

Warnes G.R., (2004); *The gregmisc Package*; R Reference Guide available from www.r-project.org.

---

elliptical2d *Bivariate Elliptical Densities*

Description

Density function for bivariate elliptical distributions.

Usage

```
delliptical2d(x, y = x, rho = 0, param = NULL, type = c("norm", "cauchy", "t",
    "logistic", "laplace", "kotz", "epower"), output = c("vector", "list"))
```

Arguments

output	output - a character string specifying how the output should be formatted. By default a vector of the same length as u and v. If specified as "list" then u and v are expected to span a two-dimensional grid as outputted by the function grid2d and the function returns a list with elements $x, y, and z which can be directly used for example by 2D plotting functions.
param	additional parameters to specify the bivariate density function. Only effective for the Kotz and Exponential Power distribution. For the Kotz distribution we can specify a numeric value, by default defined as param=c(r=sqrt(2)), and for the Exponential Power distribution a numeric vector, by default defined as param=c(r=sqrt(2)),s=1/2.
rho	the correlation parameter, a numeric value ranging between minus one and one, by default zero.
type	the type of the elliptical copula. A character string selected from: "norm", "cauchy", "t", "laplace", "kotz", or "epower".
x, y	two numeric vectors defining the x and y coordinates.

Value

`delliptical2d`
returns a two column matrix of densities for the selected bivariate elliptical distribution function.

Author(s)

Diethelm Wuertz for the Rmetrics R-port.

References

Azzalini A., (2004); *The sn Package*; R Reference Guide available from www.r-project.org.

Venables W.N., Ripley B.D., (2002); *Modern Applied Statistics with S*, Fourth Edition, Springer.

Examples

```
## Kotz' Elliptical Density:
   x = (-40:40)/10
   X = grid2d(x)
   z = delliptical2d(X$x, X$y, rho = 0.5, type = "kotz")
   Z = list(x = x, y = x, z = matrix(z, ncol = length(x)))
   persp(Z, theta = -40, phi = 30, col = "steelblue")
```

MultivariateDistribution

Multivariate Skew Normal and Student-t Distributions

Description

A collection and description of functions to compute multivariate densities and probabilities from skew normal and skew Student-t distribution functions. Furthermore, multivariate random daviates can be generated, and for multivariate data, the parameters of the underlying distribution can be estimated by the maximum log-likelihood estimation.

The functions are:

`dmvsnorm`	Multivariate Skew Normal Density,
`pmvsnorm`	Multivariate Skew Normal Probability,
`rmvsnorm`	Random Deviates from MV Skew Normal Distribution,
`dmvst`	Multivariate Skew Student Density,
`pmvst`	Multivariate Skew Student Probability,
`rmvst`	Random Deviates from MV Skew Student Distribution,
`mvFit`	Fits a MV Skew Normal or Student-t Distribution,
`print`	S3 print method for an object of class 'fMV',
`plot`	S3 Plot method for an object of class 'fMV',
`summary`	S3 summary method for an object of class 'fMV'.

These functions are useful for portfolio selection and optimization if one likes to model the data by multivariate normal, skew normal, or skew Student-t distribution functions.

Usage

```
dmvsnorm(x, dim = 2, mu = rep(0, dim), Omega = diag(dim), alpha = rep(0, dim))
pmvsnorm(q, dim = 2, mu = rep(0, dim), Omega = diag(dim), alpha = rep(0, dim))
rmvsnorm(n, dim = 2, mu = rep(0, dim), Omega = diag(dim), alpha = rep(0, dim))

dmvst(x, dim = 2, mu = rep(0, dim), Omega = diag(dim), alpha = rep(0, dim), df = 4)
pmvst(q, dim = 2, mu = rep(0, dim), Omega = diag(dim), alpha = rep(0, dim), df = 4)
rmvst(n, dim = 2, mu = rep(0, dim), Omega = diag(dim), alpha = rep(0, dim), df = 4)

mvFit(x, method = c("snorm", "st"), fixed.df = NA, title = NULL,
    description = NULL, trace = FALSE, ...)

## S4 method for signature 'fMV':
show(object)

## S3 method for class 'fMV':
plot(x, which = "ask", ...)
## S3 method for class 'fMV':
summary(object, which = "ask", doplot = TRUE, ...)
```

Arguments

description	[mvFit] - a character string, assigning a brief description to an "fMV" object.
doplot	a logical value, by default TRUE. Should a plot be generated and displayed?
dim	[*mvsnorm][*mvst] - the colum dimension of the matrix x. If x is specified as a vector, dim=1 must be set to one.
fixed.df	either NA, the default, or a numeric value assigning the number of degrees of freedom to the model. In the case that fixed.df=NA the value of df will be included in the optimization process, otherwise not.
method	[mvFit] - a string value specifying the method applied in the optimizing process. This can be either method="snorm" or method="st", in the first case the parameters for a skew normal distribution will be fitted and in the second case the parameters for a skew Student-t distribution.
mu, Omega, alpha, df	[*mvsnorm][*mvst] - the model parameters: mu a vector of mean values, one for each column, Omega the covariance matrix, alpha the skewness vector, and df the number of degrees of freedom which is a measure for the fatness of the tails (excess kurtosis).

	For a symmetric distribution `alpha` is a vector of zeros. For the normal distributions `df` is not used and set to infinity, `Inf`. Note that all columns assume the same value for `df`.
`n`	[rmvsnorm][rmvst] - number of data records to be simulated, an integer value.
`object`	[summary] - an object of class `fMV`.
`title`	[mvFit] - a character string, assigning a title to an `"fMV"` object.
`trace`	a logical, if set to `TRUE` the optimization process will be traced, otherwise not. The default setting is `FALSE`.
`which`	which of the five plots should be displayed? `which` can be either a character string, `"all"` (displays all plots) or `"ask"` (interactively asks which one to display), or a vector of 5 logical values, for those elements which are set `TRUE` the correponding plot will be displayed.
`x, q`	[*mvsnorm][*mvst][mvFit] - a numeric matrix of quantiles (returns) or any other rectangular object like a data.frame or a multivariate time series objects which can be transformed by the function `as.matrix` to an object of class `matrix`. If `x` is a vector, it will be transformed into a matrix object with one column. [plot][print] - An object of class `fMV`.
`...`	optional arguments to be passed to the optimization or plotting functions.

Details

These are "easy-to-use" functions which allow quickly to simulate multivariate data sets and to fit their parameters assuming a multivariate skew normal or skew Student-t distribution. The functions make use of the contributed R packages `sn` and `mtvnorm`.

For an extended functionality in modelling multivariate skew normal and Student-t distributions we recommend to download and use the functions from the original package `sn` which requires also the package `mtvnorm`.

The algorithm for the computation of the normal and Student-t distribution functions is described by Genz (1992) and (1993), and its implementation by Hothorn, Bretz, and Genz (2001).

The parameter estimation is done by the maximum log-likelihood estimation. The algorithm and the implemantation was done by Azzalini (1985-2003).

The multivariate skew-normal distribution is discussed in detail by Azzalini and Dalla Valle (1996); the `(Omega,alpha)` parametrization adopted here is the one of Azzalini and Capitanio (1999).

The family of multivariate skew-t distributions is an extension of the multivariate Student's t family, via the introduction of a shape parameter which regulates skewness; for a zero shape parameter the skew Student-t distribution reduces to the usual t distribution. When `df = Inf` the distribution reduces to the multivariate skew-normal one.

The plot facilities have been completely reimplemented. The S3 plot method allows for selective batch and interactive plots. The argument `which` takes care for the desired operation.

The contributed R package `mvtnorm` is required, the contributed R package `sn` is builtin, since it is not available on the Debian Software Server.

Value

[dp]mvsnorm
[dp]mvst
return a vector of density and probability values computed from the matrix x.

mvFit
returns a S4 object class of class "fASSETS", with the following slots:

@call	the matched function call.
@data	the input data in form of a data.frame.
@description	allows for a brief project description.
@fit	the results as a list returned from the underlying fitting function.
@method	the selected method to fit the distribution, either "snorm", or "st".
@model	the model parameters describing the fitted parameters in form of a list, model=list(mu, Omega, alpha, df.
@title	a title string.
@fit$dp	a list containing the direct parameters beta, Omega, alpha. Here, beta is a matrix of regression coefficients with dim(beta)=c(nrow(X), ncol(y)), Omega is a covariance matrix of order dim, alpha is a vector of shape parameters of length dim.
@fit$se	a list containing the components beta, alpha, info. Here, beta and alpha are the standard errors for the corresponding point estimates; info is the observed information matrix for the working parameter, as explained below.
@fit$optim	the list returned by the optimizer optim; see the documentation of this function for explanation of its components.

print
is the S3 print method for objects of class "fMV" returned from the function mvFit. If shows a summary report of the parameter fit.

plot
is the S3 plot method for objects of class "fMV" returned from the function mvFit. Five plots are produced. The first plot produces a scatterplot and in one dimension an histogram plot with the fitted distribution superimposed. The second and third plot represent a QQ-plots of Mahalanobis distances. The first of these refers to the fitting of a multivariate normal distribution, a standard statistical procedure; the second gives the corresponding QQ-plot of suitable Mahalanobis distances for the multivariate skew-normal fit. The fourth and fivth plots are similar to the previous ones, except that PP-plots are produced. The plots can be displayed in several ways, depending an the argument which, for details we refer to the arguments list above.

summary
is the S3 summary method for objects of class "fMV" returned from the function mvFit. The summary method prints and plots in one step the results as done by the print and plot methods.

Author(s)

Torsten Hothorn for R's `mvtnorm` package,
Alan Ganz and Frank Bretz for the underlying Fortran Code,
Adelchi Azzalini for R's `sn` package,
Diethelm Wuertz for the Rmetrics port.

References

Azzalini A. (1985); *A Class of Distributions Which Includes the Normal Ones*, Scandinavian Journal of Statistics 12, 171–178.

Azzalini A. (1986); *Further Results on a Class of Distributions Which Includes the Normal Ones*, Statistica 46, 199–208.

Azzalini A., Dalla Valle A. (1996); *The Multivariate Skew-normal Distribution*, Biometrika 83, 715–726.

Azzalini A., Capitanio A. (1999); *Statistical Applications of the Multivariate Skew-normal Distribution*, Journal Roy. Statist. Soc. B61, 579–602.

Azzalini A., Capitanio A. (2003); *Distributions Generated by Perturbation of Symmetry with Emphasis on a Multivariate Skew-t Distribution*, Journal Roy. Statist. Soc. B65, 367–389.

Genz A., Bretz F. (1999); *Numerical Computation of Multivariate t-Probabilities with Application to Power Calculation of Multiple Contrasts*, Journal of Statistical Computation and Simulation 63, 361–378.

Genz A. (1992); *Numerical Computation of Multivariate Normal Probabilities*, Journal of Computational and Graphical Statistics 1, 141–149.

Genz A. (1993); *Comparison of Methods for the Computation of Multivariate Normal Probabilities*, Computing Science and Statistics 25, 400–405.

Hothorn T., Bretz F., Genz A. (2001); *On Multivariate t and Gauss Probabilities in R*, R News 1/2, 27–29.

Examples

```
## rmvst -
   par(mfcol = c(3, 1), cex = 0.7)
   r1 = rmvst(200, dim = 1)
   ts.plot(as.ts(r1), xlab = "r", main = "Student-t 1d")
   r2 = rmvst(200, dim = 2, Omega = matrix(c(1, 0.5, 0.5, 1), 2))
   ts.plot(as.ts(r2), xlab = "r", col = 2:3, main = "Student-t 2d")
   r3 = rmvst(200, dim = 3, mu = c(-1, 0, 1), alpha = c(1, -1, 1), df = 5)
   ts.plot(as.ts(r3), xlab = "r", col = 2:4, main = "Skew Student-t 3d")

## mvFit -
   # Generate Grid Points:
   n = 51
   x = seq(-3, 3, length = n)
   xoy = cbind(rep(x, n), as.vector(matrix(x, n, n, byrow = TRUE)))
   X = matrix(xoy, n * n, 2, byrow = FALSE)
   head(X)
   # The Bivariate Normal Case:
```

```
Z = matrix(dmvsnorm(X, dim = 2), length(x))
par (mfrow = c(2, 2), cex = 0.7)
persp(x, x, Z, theta = -40, phi = 30, col = "steelblue")
title(main = "Bivariate Normal Plot")
image(x, x, Z)
title(main = "Bivariate Normal Contours")
contour(x, x, Z, add = TRUE)
# The Bivariate Skew-Student-t Case:
mu = c(-0.1, 0.1)
Omega = matrix(c(1, 0.5, 0.5, 1), 2)
alpha = c(-1, 1)
Z = matrix(dmvst(X, 2, mu, Omega, alpha, df = 3), length(x))
persp(x, x, Z, theta = -40, phi = 30, col = "steelblue")
title(main = "Bivariate Student-t Plot")
image(x, x, Z)
contour(x, x, Z, add = TRUE)
title(main = "Bivariate Student-t Contours")
```

`norm2d` *Bivariate Normal Distribution*

Description

Density, distribution function, and random generation for the bivariate normal distribution.

Usage

```
pnorm2d(x, y = x, rho = 0)
dnorm2d(x, y = x, rho = 0)
rnorm2d(n, rho = 0)
```

Arguments

`n`	the number of random deviates to be generated, an integer value.
`rho`	the correlation parameter, a numeric value ranging between minus one and one, by default zero.
`x, y`	two numeric vectors defining the x and y coordinates.

Value

`pnorm2d`
returns a two column matrix of probabilities for the bivariate normal distribution function.

`dnorm2d`
returns a two column matrix of densities for the bivariate normal distribution function.

`rnorm2d`
returns a two column matrix of random deviates generated from the bivariate normal distribution function.

Author(s)

Adelchi Azzalini for the underlying pnorm2d function,
Diethelm Wuertz for the Rmetrics R-port.

References

Azzalini A., (2004); *The sn Package*; R Reference Guide available from www.r-project.org.

Venables W.N., Ripley B.D., (2002); *Modern Applied Statistics with S*, Fourth Edition, Springer.

Examples

```
## Bivariate Normal Density:
   x = (-40:40)/10
   X = grid2d(x)
   z = dnorm2d(X$x, X$y, rho = 0.5)
   Z = list(x = x, y = x, z = matrix(z, ncol = length(x)))
   persp(Z, theta = -40, phi = 30, col = "steelblue")
```

---

t2d *Bivariate Student-t Distribution*

Description

Density, distribution function, and random generation for the bivariate Student-t distribution.

Usage

```
pt2d(x, y = x, rho = 0, nu = 4)
dt2d(x, y = x, rho = 0, nu = 4)
rt2d(n, rho = 0, nu = 4)
```

Arguments

n	the number of random deviates to be generated, an integer value.
nu	the number of degrees of freedom, a numeric value ranging between two and infinity, by default four.
rho	the correlation parameter, a numeric value ranging between minus one and one, by default zero.
x, y	two numeric vectors defining the x and y coordinates.

Value

`pt2d`
returns a two column matrix of probabilities for the bivariate Student-t distribution function.

`dt2d`
returns a two column matrix of densities for the bivariate Student-t distribution function.

`rt2d`
returns a two column matrix of random deviates generated from the bivariate Student-t distribution function.

Author(s)

Adelchi Azzalini for the underlying pnorm2d function,
Diethelm Wuertz for the Rmetrics R-port.

References

Azzalini A., (2004); *The sn Package*; R Reference Guide available from www.r-project.org.

Venables W.N., Ripley B.D., (2002); *Modern Applied Statistics with S*, Fourth Edition, Springer.

Examples

```
## Bivariate Student-t Density:
   x = (-40:40)/10
   X = grid2d(x)
   z = dt2d(X$x, X$y, rho = 0.5, nu =6)
   Z = list(x = x, y = x, z = matrix(z, ncol = length(x)))
   persp(Z, theta = -40, phi = 30, col = "steelblue")
```

Index