Package 'bivariate'

September 13, 2020

Title Bivariate Probability Distributions

Version 0.6.0

Date 2020-09-13

License GPL (>= 2)

Maintainer Abby Spurdle <spurdle.a@gmail.com>

Author Abby Spurdle

URL https://sites.google.com/site/spurdlea/r

Description Convenience functions for plotting/evaluating bivariate (uniform, binomial, Poisson, categorical, normal and bimodal) distributions, trivariate (normal and Dirichlet) distributions, bivariate kernel density estimates and bivariate empirical cumulative distribution functions. Supports their probability mass functions (PMFs), probability density functions (PDFs) and cumulative distribution functions (CDFs), generally where applicable.

Imports intoo, barsurf, mvtnorm, KernSmooth

Suggests misc3d, probhat, MASS

Contact Primary <spurdle.a@gmail.com>, Secondary <spurdle.a@aol.com>

NeedsCompilation no

Repository CRAN

Date/Publication 2020-09-13 13:10:21 UTC

R topics documented:

`11_PD_uniform` *Uniform Distributions*

Description

Bivariate uniform distributions, both discrete and continuous (on closed intervals).

Usage

```
#discrete
dubvpmf (
    a.X, b.X, #first variable
    a.Y, b.Y) #second variable
dubvcdf (
    a.X, b.X,
    a.Y, b.Y)

#continuous
cubvpdf (
    a.X, b.X,
    a.Y, b.Y)
cubvcdf (
    a.X, b.X,
    a.Y, b.Y)
```

Arguments

`a.X, b.X`	Integers (in the discrete case) or numeric values (in the continuous case), giving the min/max possible values of X.
`a.Y, b.Y`	Integers (in the discrete case) or numeric values (in the continuous case), giving the min/max possible values of Y.

Value

Self-referencing function objects.

Refer to Mockup Function Objects.

References

Refer to the vignette for an overview, references, theoretical background and better examples.

See Also

Binomial, Poisson and Categorical
For other probability distributions of discrete random variables.

Normal, Bimodal, Dirichlet and Nonparametric
For other probability distributions of continuous random variables.

Main Plotting Functions

Density Matrices

Examples

```
f <- dubvpmf (
    1, 4, #first variable
    1, 4) #second varaible

1 / (4 * 4)

plot (f)
f (1, 1)
```

`12_PD_binomial` *Binomial Distributions*

Description

Bivariate binomial distributions.

Usage

```
bnbvpmf (p.X, p.Y, n=1)
bnbvcdf (p.X, p.Y, n=1)
```

Arguments

`p.X, p.Y`	Numeric values (between zero and one), giving the probabilities of the first and second events.
`n`	Positive integer, giving the number of trials.

Value

Self-referencing function objects.

Refer to Mockup Function Objects.

References

Refer to the vignette for an overview, references, theoretical background and better examples.

See Also

Uniform
For uniform distributions.

Poisson and Categorical
For other probability distributions of discrete random variables.

Normal, Bimodal, Dirichlet and Nonparametric
For other probability distributions of continuous random variables.

Main Plotting Functions

Density Matrices

Examples

```
f <- bnbvpmf (0.5, 0.5, 10)

plot (f)
f (5, 5)
```

`13_PD_poisson` *Poisson Distributions*

Description

Bivariate Poisson distributions.

Usage

```
pbvpmf (lambda.1, lambda.2, lambda.3)
pbvcdf (lambda.1, lambda.2, lambda.3)

pbvpmf.2 (mean.X, mean.Y, cov)
pbvcdf.2 (mean.X, mean.Y, cov)
```

Arguments

`lambda.1, lambda.2, lambda.3`	Positive numeric values, giving the first, second and third lambda parameters.
`mean.X, mean.Y`	Suitable numeric values, giving the mean of X and Y. Note that their means equal their variances.
`cov`	Suitable numeric value, giving the covariance of X and Y.

Value

Self-referencing function objects.

Refer to Mockup Function Objects.

References

Refer to the vignette for an overview, references, theoretical background and better examples.

See Also

Uniform
For uniform distributions.

Binomial and Categorical
For other probability distributions of discrete random variables.

Normal, Bimodal, Dirichlet and Nonparametric
For other probability distributions of continuous random variables.

Main Plotting Functions

Density Matrices

Examples

```
f <- pbvpmf (10, 10, 0)

plot (f)
f (5, 5)
```

14_PD_categorical *Categorical Distributions*

Description

Bivariate categorical distributions.

Usage

```
gbvpmf (p)
```

Arguments

p	Numeric matrix of probabilities (or frequencies), preferably with row and column names.

Value

Self-referencing function objects.

Refer to Mockup Function Objects.

References

Refer to the vignette for an overview, references, theoretical background and better examples.

Note that the probhat package supports categorical distributions, and provides more tools.

See Also

Uniform
For uniform distributions.

Binomial and Poisson
For other probability distributions of discrete random variables.

Normal, Bimodal, Dirichlet and Nonparametric
For other probability distributions of continuous random variables.

Main Plotting Functions

Density Matrices

Examples

```
library (barsurf)

h <- matrix (sample (1:24), 4, 6)
rownames (h) <- LETTERS [1:4]
colnames (h) <- letters [1:6]

f <- gbvpmf (h)

plot (f)
plot (f, FALSE)

h [2, 1:4] / sum (h)
f (2, 1:4)
f ("B", c ("a", "b", "c", "d") )
```

15_PD_normal *Normal Distributions*

Description

Bivariate and trivariate normal distributions.

Usage

```
#normal bivariate (PDF and CDF), sd/cor
nbvpdf (mean.X=0, mean.Y=0, sd.X=1, sd.Y=1, cor=0)
nbvcdf (mean.X=0, mean.Y=0, sd.X=1, sd.Y=1, cor=0)

#normal bivariate (PDF and CDF), var/cov
nbvpdf.2 (mean.X=0, mean.Y=0, var.X=1, var.Y=1, cov=0)
nbvcdf.2 (mean.X=0, mean.Y=0, var.X=1, var.Y=1, cov=0)

#normal trivariate (PDF only)
ntvpdf (mean.X=0, mean.Y=0, mean.Z=0,
    sd.X=1, sd.Y=1, sd.Z=1,
```

```
    cor.XY=0, cor.XZ=0, cor.YZ=0)
ntvpdf.2 (mean.X=0, mean.Y=0, mean.Z=0,
    var.X=1, var.Y=1, var.Z=1,
    cov.XY=0, cov.XZ=0, cov.YZ=0)
```

Arguments

`mean.X, mean.Y, mean.Z`
: Numeric values, giving the means of X, Y and Z.

`sd.X, sd.Y, sd.Z`
: Positive numeric values, giving the standard deviations of X, Y and Z.

`var.X, var.Y, var.Z`
: Positive numeric values, giving the variances of X, Y and Z.

`cor`
: Numeric value, giving the correlation between X and Y.

`cov`
: Numeric value, giving the covariance between X and Y.

`cor.XY, cor.XZ, cor.YZ`
: Numeric values, giving the pairwise correlations between X, Y and Z.

`cov.XY, cov.XZ, cov.YZ`
: Numeric values, giving the pairwise covariances between X, Y and Z.

Details

Note that the resulting covariance matrix needs to be a positive definite matrix.
Also, in the bivariate case, both sd/variance parameters need to be positive, and the absolute correlation needs to be less than one.

Note that the misc3d package needs to be installed and loaded, in order to plot the trivariate normal distribution.

Value

Self-referencing function objects.

Refer to Mockup Function Objects.

References

Refer to the vignette for an overview, references, theoretical background and better examples.

See Also

Uniform
For uniform distributions.

Binomial, Poisson and Categorical
For other probability distributions of discrete random variables.

Bimodal, Dirichlet and Nonparametric
For other probability distributions of continuous random variables.

Main Plotting Functions

Random Numbers
For bivariate and trivariate normal random numbers.

Density Matrices

Examples

```
f <- nbvpdf ()

plot (f)
f (0, 0)
```

16_PD_bimodal *Bimodal Distributions*

Description

Bivariate bimodal distributions.

Usage

```
bmbvpdf (
    mean.X1, mean.Y1, sd.X1, sd.Y1,
    mean.X2, mean.Y2, sd.X2, sd.Y2)
bmbvcdf (
    mean.X1, mean.Y1, sd.X1, sd.Y1,
    mean.X2, mean.Y2, sd.X2, sd.Y2)

bmbvpdf.2 (
    mean.X1, mean.Y1, var.X1, var.Y1,
    mean.X2, mean.Y2, var.X2, var.Y2)
bmbvcdf.2 (
    mean.X1, mean.Y1, var.X1, var.Y1,
    mean.X2, mean.Y2, var.X2, var.Y2)
```

Arguments

`mean.X1, mean.Y1`
: Numeric values, giving the means of the first X and Y components.

`sd.X1, sd.Y1`
: Positive numeric values, giving the standard deviations of the first X and Y components.

`var.X1, var.Y1`
: Positive numeric values, giving the variances of the first X and Y components.

`mean.X2, mean.Y2`
: Numeric values, giving the means of the second X and Y components.

`sd.X2, sd.Y2`
: Positive numeric values, giving the standard deviations of the second X and Y components.

`var.X2, var.Y2`
: Positive numeric values, giving the variances of the second X and Y components.

Value

Self-referencing function objects.

Refer to Mockup Function Objects.

References

Refer to the vignette for an overview, references, theoretical background and better examples.

See Also

Uniform
For uniform distributions.

Binomial, Poisson and Categorical
For other probability distributions of discrete random variables.

Normal, Dirichlet and Nonparametric
For other probability distributions of continuous random variables.

Main Plotting Functions

Density Matrices

Examples

```
f <- bmbvpdf (3.5, 0, 1, 1, 6.5, 0, 1, 1)

plot (f, axes = c (TRUE, FALSE), ref.arrows=FALSE, xat = c (3.5, 6.5) )
f (3.5, 0)
```

17_PD_dirichlet *Trivariate Dirichlet Distributions*

Description

Trivariate Dirichlet distributions.

Usage

```
dtvpdf (alpha.X, alpha.Y, alpha.Z)
```

Arguments

`alpha.X, alpha.Y, alpha.Z`
Positive numeric values, giving the alpha parameters.

Value

Self-referencing function objects.

Refer to Mockup Function Objects.

References

Refer to the vignette for an overview, references, theoretical background and better examples.

See Also

Uniform
For uniform distributions.

Binomial, Poisson and Categorical
For other probability distributions of discrete random variables.

Normal, Bimodal and Nonparametric
For other probability distributions of continuous random variables.

Main Plotting Functions

Density Matrices

Examples

```
f <- dtvpdf (1.05, 1.05, 1.05)

plot (f)
f (0.3, 0.3, 0.4)
```

18_PD_kernel_and_empirical

Nonparametric Distributions

Description

Bivariate kernel density estimates and bivariate empirical cumulative distribution functions.

Usage

```
#kernel density estimates
kbvpdf (x, y, xbw, ybw)

#ecdf
ebvcdf (x, y)
```

Arguments

x, y	Numeric vectors, of x and y values.
xbw, ybw	Optional numeric values, giving the x and y bandwidths.

Details

Default xlab and ylab labels are taken for deparsed x and y names.

Bandwidth parameters are optional, however, the default values are likely to be sub-optimal.

Kernel density estimates use KernSmooth::bkde2D.

Value

Self-referencing function objects.

Refer to Mockup Function Objects.

Note that you can't evaluate the function representing kernel density estimates.
(However, the bvmat function can be used to compute density matrices).

References

Refer to the vignette for an overview, references, theoretical background and better examples.

Note that the probhat package provides more tools for kernel smoothing.

See Also

Uniform
For uniform distributions.

Binomial, Poisson and Categorical
For other probability distributions of discrete random variables.

Normal, Bimodal and Dirichlet
For other probability distributions of continuous random variables.

Main Plotting Functions

Density Matrices

Examples

```
########################################
#kernel density estimates
########################################
data ("geyser", package="MASS")
attach (geyser)

#adapted from the KernSmooth package
fh <- kbvpdf (duration, waiting, 0.7, 7)

plot (fh,, TRUE)
plot (fh, TRUE, z.axis=TRUE, ref.arrows=FALSE)

detach (geyser)

########################################
#ECDF
########################################
attach (trees)

Fh <- ebvcdf (Height, Volume)

plot (Fh)
plot (Fh, FALSE)
```

```
Fh (median (Height), mean (Volume) )

detach (trees)
```

21_mockup_function_objects
Mockup Function Objects

Description

Hard-coded functions, representing (runtime) function objects.

DO NOT CALL THESE FUNCTIONS.
CALL A CONSTRUCTOR, WHICH SHOULD RETURN A FUNCTION OBJECT.

Usage

```
#PMF/PDF and CDF
#(all distributions, except the exceptions below)
mf.f (x, y)
mf.F (x, y)

#Categorical PMF
mf.gf (a, b)

#Trivariate Normal PDF
mf.f3 (x, y, z)

#Dirichlet PDF
mf.wf (x, y, z = 1 - x - y, ..., log=FALSE)

#ECDF
mf.Fh (x, y)
```

Arguments

`x, y, z`	Vectors giving the evaluation bins/points. In discrete distributions, they should be integers. In continuous distributions they should be numerics. Currently, missing values, non-finite values and zero-length vectors are not allowed.
`a, b`	Integer or character vectors, giving the evaluation bins. (Same as x and y above, except that strings are allowed).
`log`	Logical, if true, return log-transformed densities.
`...`	Ignored.

Details

DO NO CALL THESE FUNCTIONS.
CALL A CONSTRUCTOR, WHICH SHOULD RETURN A FUNCTION OBJECT.

Uniform and binomial distributions allow evaluation values outside the supported region.
(PMFs/PDFs return zero mass/density, and CDFs return the corresponding values).

Poisson evaluation values need to be non-negative.

Categorical string evaluation values need to match group names, taken from the parameter matrix.

Categorical integer evaluation values, need to have:
x in [1, ngx], where ngx is the number of x groups (and rows of the parameter matrix).
y in [1, ngy], where ngy is the number of y groups (and columns of the parameter matrix).

In Dirichlet distributions, z is optional.
And the three variables should be in the interval (0, 1) and sum to one.

Note that kbvpdf objects (often labelled fh), can't be evaluated.
(However, the bvmat function can be used to compute density matrices).

Also note that the help pages, linked to in the see also section, have better examples.

Value

PMFs return a numeric vector, giving mass.
P (X = x, Y = y)

PDFs return a numeric vector, giving density.

CDFs return a numeric vector, giving cumulative probability.
P (X <= x, Y <= y)

References

Refer to the vignette for an overview, references, theoretical background and better examples.

See Also

Uniform
For uniform distributions.

Binomial, Poisson and Categorical
For other probability distributions of discrete random variables.

Normal, Bimodal, Dirichlet and Nonparametric
For other probability distributions of continuous random variables.

Examples

```
f <- nbvpdf ()

x <- -3:3
y <- 0
density <- f (x, y)

cbind (x, y, density)
```

31_new_generics *New Generic Functions*

Description

New generic functions for plotting function objects, generating random numbers and computing density matrices.

Usage

```
bvplot (...)
bvrng (...)
bvmat (...)
```

Arguments

...	Arguments for specific methods.

Details

Do not call bvplot.
Call plot instead.

References

Refer to the vignette for an overview, references, theoretical background and better examples.

See Also

Main Plotting Functions

Random Numbers

Density Matrices

32_supporting_methods *Print and Plot Methods*

Description

Print and plot methods for bv (bivariate) and tv (trivariate) objects.

Usage

```
## S3 method for class 'bv'
print(x, ...)
## S3 method for class 'tv'
print(x, ...)

## S3 method for class 'bv'
plot(x, ...)
## S3 method for class 'tv'
plot(x, ...)
```

Arguments

x	A bv object.
...	Other arguments for intoo::object.summary.

Details

The print methods call the intoo::object.summary function.

The plot methods call bvplot, which then call one of the bvplot methods.

References

Refer to the vignette for an overview, references, theoretical background and better examples.

Examples

```
f <- nbvpdf ()

print (f)
plot (f)
```

41_main_plotting_functions

Plot Methods

Description

Plot methods for bv (bivariate) and tv (trivariate) objects.

DO NOT CALL bvplot, CALL plot.

Note that the misc3d package needs to be installed and loaded, in order to plot the trivariate normal distribution.

Usage

```
########################################
#discrete uniform
########################################
## S3 method for class 'dubvpmf'
bvplot(f, in3d=TRUE, ..., xlim, ylim, all=FALSE)
## S3 method for class 'dubvcdf'
bvplot(F, in3d=TRUE, ..., xlim, ylim)

########################################
#binomial
########################################
## S3 method for class 'bnbvpmf'
bvplot(f, in3d=TRUE, ..., xlim, ylim, all=FALSE)
## S3 method for class 'bnbvcdf'
bvplot(F, in3d=TRUE, ..., xlim, ylim)

########################################
#poisson
########################################
## S3 method for class 'pbvpmf'
bvplot(f, in3d=TRUE, ..., xlim, ylim, all=FALSE)
## S3 method for class 'pbvcdf'
bvplot(F, in3d=TRUE, ..., xlim, ylim)

########################################
#continuous uniform
########################################
## S3 method for class 'cubvpdf'
bvplot(f, in3d=TRUE, ..., xlim, ylim, all=FALSE, n=20)
## S3 method for class 'cubvcdf'
bvplot(F, in3d=TRUE, ..., xlim, ylim, n=20)

########################################
#normal
#(bivariate and trivariate)
########################################
## S3 method for class 'nbvpdf'
bvplot(f, in3d=TRUE, ..., xlim, ylim, all=FALSE, n=30)
## S3 method for class 'nbvcdf'
bvplot(F, in3d=TRUE, ..., xlim, ylim, n=30)

## S3 method for class 'ntvpdf'
bvplot(f, iso=TRUE, ..., xlim, ylim, zlim)

########################################
#bimodal
########################################
```

```
## S3 method for class 'bmbvpdf'
bvplot(f, in3d=TRUE, ..., xlim, ylim, all=FALSE, n=40)
## S3 method for class 'bmbvcdf'
bvplot(F, in3d=TRUE, ..., xlim, ylim, n=40)

#######################################
#categorical, dirichlet, kernel, ecdf
#######################################
## S3 method for class 'gbvpmf'
bvplot(f, in3d=TRUE, data, ...)

## S3 method for class 'dtvpdf'
bvplot(f, in3d=TRUE, ..., log=FALSE, n=30)

## S3 method for class 'kbvpdf'
bvplot(fh, in3d=TRUE, data = (fh %$% "n" <= 2000), ...,
    xlim, ylim, n=30, point.color="#00000030")

## S3 method for class 'ebvcdf'
bvplot(Fh, in3d=TRUE, ..., reg = (Fh %$% "n" > 40) )

#######################################
#additional plotting functions
#(called by the ebvcdf method, above)
#######################################
ebvcdf_plot_reg (Fh, in3d=TRUE, ..., xlim, ylim, n=30)

ebvcdf_plot_step_2d (Fh, data=TRUE, steps=data, ...,
    point.color="#00000030", line.color="#000000", border.color="#808080",
    main.colff = opt.litmus.fit (), rim.colff=rim.litmus.fit)

ebvcdf_plot_step_3d (Fh, ...,
    top.color = opt.top.color (), side.color = opt.side.color (),
    rim.color="#D0D0D0")
```

Arguments

`f, F, fh, Fh`	A function representing a PMF/PDF or CDF, from this package. Refer to the see also section.
`in3d`	Logical value, if true, a 3d plot, if false, a 2d plot.
`xlim, ylim, zlim`	The x, y and z ranges for the plot.
`all`	Logical value, if true, plot a two by two array of both the PMF/PDF and CDF using both 2d and 3d plots. Note that by default, ref.arrows (reference arrows in 3d plots) will be true, regardless of global options. And some plotting arguments are ignored.

`n` Integer vector of length one or two, the number of grid bins/points in each x and y direction.
(Except for Dirichlet distributions, where it needs to be scalar).

Note that, n, is also an attribute of kbvpdf and ebvcdf objects, representing sample size.

`iso` Logical value, if true, plot isosurfaces.

`log` Logical value, if true, plot the log density.

`reg` Logical value, if true, plot the probability distribution as a continuous surface, evaluated over a regularly-spaced grid.
(Refer to details).

`data` Logical value, if true, plot the data points/labels.
Ignored, if in3d is true.

`steps` Logical value, if true, plot the steps.

`point.color, line.color, border.color`
String, the color for points, lines and borders.

`top.color, side.color, rim.color`
Strings, giving the bar top, bar side and rim bar colors.

`main.colff, rim.colff`
Litmus-fitting functions.
Refer to the barsurf package.

Note that the default for rim.colff is rim.litmus.fit, which is defined in this package.

`...` Other arguments for plotting functions in the barsurf package.
Refer to the details section.

Details

DO NOT CALL bvplot, CALL plot.

GENERAL INFORMATION

These functions call functions in the barsurf package:

2d plotting functions call plot_dfield, plot_cfield or plot_tricontour.
(The categorical method, calls plot_matrix, which calls plot_dfield).

3d plotting functions call plot_bar, plot_surface or plot_trisurface.
Except for the trivariate normal method, which calls plotf_isosurface or plotf_cfield3.

Re-iterating, the misc3d package needs to be installed and loaded, in order to plot the trivariate normal distribution.

The barsurf package is built on top of the base graphics system, but uses a different color system.
You will need to refer to the documentation for that package to change surface and heatmap colors.

DEFAULTS

As of bivariate version 0.6.0, 3d plots are the default.

By default, 3d plots use reference arrows.
(This can be changed by first calling set.bs.options with ref.arrows=FALSE).

By default, 3d PMFs/PDFs have no z axis markup, but 3d CDFs do.
(This can be changed by setting the z.axis argument to true or false).

By default, 3d PMFs/PDFs have zlim equal to c (0, zmax).
And by default, 3d CDFs have zlim equal to c (0, 1).

Where in general, zmax will be the maximum value of the PMF/PDF, within the evaluation window. Exceptions include uniform distributions (which use zero to one, if the maximum function value is less than or equal to one) and Dirichlet distributions (which vertically center the surface, if all alpha parameters are equal to one).

By default, 2d PMFs/PDFs have unlabelled contour lines, and 2d CDFs have labelled contour lines at 0.2, 0.4, 0.6 and 0.8.
(The ncontours, contour.labels, fb and clabs arguments, can be used to change this).

CATEGORICAL DISTRIBUTIONS

Note that the 2d plot has a different orientation to other 2d plots in this package.

For optimal results, you should update the plot margins, before and after the calling the plot method.
(This can be done, using the matrix.margins function, from the barsurf package).

EMPIRICAL CUMULATIVE DISTRIBUTION FUNCTIONS

The plot method for ebvcdf objects calls either plot_ebvcdf_reg, plot_ebvcdf_step_2d or plot_ebvcdf_step_3d, depending on the in3d and reg arguments.

The plot_ebvcdf_step_2d and plot_ebvcdf_step_3d functions plot the ECDF as a step function.
(This could be regarded as an exact representation).

The plot_ebvcdf_reg function plots the ECDF as a continuous surface, evaluated over a regularly-spaced grid.
(This could be regarded as an approximation).

Note that setting reg=FALSE for large sample sizes, is probably a bad idea.
(But the continuous approximation improves as the sample size increases).

NOTES

Currently, continuous uniform PDFs are plotted incorrectly, if you set xlim/ylim ranges outside the supported region.

References

Refer to the vignette for an overview, references, theoretical background and better examples.

See Also

Uniform
For uniform distributions.

Binomial, Poisson and Categorical
For other probability distributions of discrete random variables.

Normal, Bimodal, Dirichlet and Nonparametric
For other probability distributions of continuous random variables.

Density Matrices

Examples

```
########################################
#PDF
########################################
f <- nbvpdf ()

plot (f)
plot (f, z.axis=TRUE, ref.arrows=FALSE)
plot (f, grid=FALSE, n=60)

plot (f, FALSE)
plot (f, FALSE, hcv=TRUE)

########################################
#CDF
########################################
F <- nbvcdf ()

plot (F)

plot (F, FALSE)
plot (F, FALSE, fb=0.5)

tempff <- function (x)
    barsurf::hot.and.cold.fit (x, t = c (0.45, 0.55) )

plot (F, gradient.shading=FALSE, colff=tempff, n=60)
```

42_random_numbers *Random Numbers*

Description

Generate random numbers from bivariate and trivariate normal distributions.

Usage

```
## S3 method for class 'nbvpdf'
bvrng(f, n=1, ...)
## S3 method for class 'ntvpdf'
bvrng(f, n=1, ...)
```

Arguments

f	A nbvpdf (bivariate normal) or ntvpdf (trivariate normal) object.
n	Integer, the number of random numbers.
...	Ignored.

Details

These functions call the rmvnorm function from the mvtnorm package.

Value

A two-column or three-column numeric matrix.

References

Refer to the vignette for an overview, references, theoretical background and better examples.

See Also

Normal
For bivarate and triviate normal distributions.

Examples

```
f <- nbvpdf ()

bvrng (f, 30)
```

`43_density_matrices` *Density Matrices*

Description

Compute density (and probability) matrices.

Usage

```
########################################
#(du) discrete uniform
#(bn) binomial
#(p)  poisson
########################################
## S3 method for class 'dubv'
bvmat(sf, xlim, ylim, ...)
## S3 method for class 'bnbv'
bvmat(sf, xlim, ylim, ...)
## S3 method for class 'pbv'
bvmat(sf, xlim, ylim, ...)

########################################
#(cu) continuous uniform
#(n)  normal
#(bm) bimodal
########################################
## S3 method for class 'cubv'
bvmat(sf, xlim, ylim, ..., n=10)
## S3 method for class 'nbv'
bvmat(sf, xlim, ylim, ..., n=10)
## S3 method for class 'bmbv'
bvmat(sf, xlim, ylim, ..., n=10)

########################################
#(g)  categorical
#(d)  dirichlet
#(k)  kernel
#(e)  empirical
########################################
## S3 method for class 'gbv'
bvmat(sf, ...)
## S3 method for class 'dtv'
bvmat(sf, ..., log=FALSE, n=10)
## S3 method for class 'kbv'
bvmat(sf, xlim, ylim, ..., n=10)
## S3 method for class 'ebv'
bvmat(sf, ..., reg=TRUE)

########################################
#additional functions
#(called by the ebv method, above)
########################################
ebvmat_reg (sf, xlim, ylim, ..., n=10)
ebvmat_step (sf, ..., extend=FALSE)
```

Arguments

sf	A suitable function object. Which here, represents one of the probability distributions from this package. Refer to the see also section.
xlim, ylim	A length-two vector for the evaluation window. For discrete distributions, this should be an integer vector, and for continuous distributions, this should be a numeric vector.
n	Integer vector of length one or two, number of grid points. (Except for Dirichlet distributions where it needs to be scalar).
log	Logical, if true, return log-transformed densities.
reg	Logical, if true, evaluate the ECDF over a regularly-spaced grid. In general, reg should be true, for large sample sizes.
extend	Logical vector or length one or two, if true, compute a rim (of probabilities) outside the observed values.
...	Ignored.

Details

These methods work for both PMFs/PDFs and CDFs, where applicable.

For example:
If bvmat is called with a normal density function it will return densities (from the density function), however, if called with a normal distribution function it will return the probabilities (from the distribution function).

Note that the ebv (empirical) method calls either the ebvmat_reg or ebvmat_step function, depending on the reg argument.

Value

A list with:

An fv component, which is a matrix, of densities or probabilities.

x and y components, which are the evaluation points.
(This length of x should equal the number of rows, and the length of y should equal the number of columns).

And exception to the above is ECDFs with reg=FALSE, where x and y are breakpoints.
(The length of x should be equal to the number of rows plus one, and the length of y should be equal to the number of columns plus one).

References

Refer to the vignette for an overview, references, theoretical background and better examples.

See Also

Uniform
For uniform distributions.

Binomial, Poisson and Categorical
For other probability distributions of discrete random variables.

Normal, Bimodal, Dirichlet and Nonparametric
For other probability distributions of continuous random variables.

Main Plotting Functions

Examples

```
data (geyser, package="MASS")
attach (geyser)

#adapted from the KernSmooth package
fh <- kbvpdf (duration, waiting, 0.7, 7)

bvmat (fh)

detach (geyser)
```

51_color_functions *Color Functions*

Description

Predefined litmus-fitting functions for the bivariate package.

Usage

```
gpd.litmus.fit (x)
rim.litmus.fit (x)
```

Arguments

x	A numeric vector.

Details

The gpd.litmus.fit and rim.litmus.fit functions call the litmus.fit and rainbow.litmus.fit functions, from the barsurf package, respectively.
Please refer to that package for details.

The rim litmus is for the (3d) rim of ECDFs.

The gpl litmus is for gold heatmaps, and is only used in the vignette.

References

Refer to the vignette for an overview, references, theoretical background and better examples.

Examples

```
f <- nbvpdf ()

s <- bvmat (f)
colf <- rim.litmus.fit (s$fv)

plot (colf)
```

Index