Package 'bivariate'

February 26, 2020

Title Bivariate Probability Distributions

Version 0.5.0

Date 2020-02-27

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 constructing, plotting and evaluating bivariate probability distributions, including their probability mass/density functions and cumulative distribution functions. Supports uniform (discrete and continuous), binomial, Poisson, categorical, normal, bimodal and Dirichlet (trivariate) distributions, and kernel smoothing and empirical cumulative distribution functions.

Imports intoo, barsurf, mvtnorm, KernSmooth

Suggests probhat, MASS

NeedsCompilation no

Repository CRAN

Date/Publication 2020-02-26 21:50:02 UTC

R topics documented:

`01_uniform` *Uniform Distributions*

Description

Bivariate uniform distributions, both discrete and continuous.

Usage

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

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

Arguments

`a.X, a.Y`	Integers (in the discrete case) or numeric values (in the continuous case), giving the lower bounds of X and Y.
`b.X, b.Y`	Integers (in the discrete case) or numeric values (in the continuous case), giving the upper bounds of X and Y.

Value

Self-referencing functions of the form:

function (x, y) { ... }

Where x and y are integer-valued vectors (in the discrete case) or numeric vectors (in the continuous case).

References

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

Examples

```
f <- dubvpmf (1, 1, 4, 4)

f (1, 1)
```

02_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	Integer, giving the number of trials.

Value

Self-referencing functions of the form:

function (x, y) {...}

Where x and y are integer-valued vectors.

References

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

Examples

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

f (4, 4)
```

03_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`	Numeric values, giving the first, second and third lambda parameters.
`mean.X, mean.Y`	Numeric values, giving the mean of X and Y. Note that their means equal their variances.
`cov`	Numeric value, giving the covariance of X and Y.

Value

Self-referencing functions of the form:

function (x, y) { ... }

Where x and y are integer-valued vectors.

References

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

Examples

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

f (4, 4)
```

`04_categorical` *Categorical Distributions*

Description

Bivariate categorical distributions.

Usage

```
cbvpmf (p)
```

Arguments

`p`	Numeric matrix of probabilities (or frequencies), preferably with row and column names. Increasing rows correspond to increasing x values and increasing columns correspond to increasing y values.

Details

Note that sometimes, text labels may be missing.

This is likely to be fixed in the near future.

Value

Self-referencing functions of the form:

function (x, y) { ... }

Where x and y are integer-valued or character vectors.

References

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

Examples

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

f <- cbvpmf (h)

f (1, 2)
```

`05_normal` *Normal Distributions*

Description

Bivariate normal distributions.

Usage

```
nbvpdf (mean.X, mean.Y, sd.X, sd.Y, cor)
nbvcdf (mean.X, mean.Y, sd.X, sd.Y, cor)

nbvpdf.2 (mean.X, mean.Y, var.X, var.Y, cov)
nbvcdf.2 (mean.X, mean.Y, var.X, var.Y, cov)
```

Arguments

`mean.X, mean.Y`	Numeric values, giving the means of X and Y.
`sd.X, sd.Y`	Numeric values, giving the standard deviations of X and Y.
`var.X, var.Y`	Numeric values, giving the variances of X and Y.
`cor`	Numeric value, giving the correlation of X and Y.
`cov`	Numeric value, giving the covariance of X and Y.

Value

Self-referencing functions of the form:

function (x, y) { ... }

Where x and y are numeric vectors.

References

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

Examples

```
f <- nbvpdf (0, 0, 1, 1, 0)

f (0, 0)
```

`06_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`	Numeric values, giving the standard deviations of the first X and Y components.
`var.X1, var.Y1`	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` Numeric values, giving the standard deviations of the second X and Y components.

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

Value

Self-referencing functions of the form:

function (x, y) {...}

Where x and y are numeric vectors.

References

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

Examples

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

f (0, 0)
```

`07_dirichlet` *Trivariate Dirichlet Distributions*

Description

Trivariate Dirichlet distributions.

Usage

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

Arguments

`alpha.X, alpha.Y, alpha.Z`
Numeric values, giving the alpha parameters, greater than zero.

Value

Self-referencing functions of the form:

function (x, y, z, log=FALSE) {...}

Where x, y and z are numeric vectors, in the interval (0, 1), which sum (across) to one.

Note that a small error (<= 0.001) is allowed, given possible floating point errors.

References

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

Examples

```
f <- dtvpdf (1, 1, 1)

f (0.3, 0.3, 0.4)
```

`08_kernel_and_empirical`
Nonparametric Distributions

Description

Bivariate kernel density estimates and bivariate empirical cumulative distribution functions.

Usage

```
#kernel density estimates
kbvpdf (x, y, bw.X, bw.Y)

#ecdf
ebvcdf (x, y)
```

Arguments

`x, y`	Numeric values of x and y values.
`bw.X, bw.Y`	Numeric values, giving the X and Y bandwidths.

Value

Self-referencing functions of the form:

function (x, y) {...}

Where x and y are numeric vectors.

Note that you can't evaluate the function representing kernel density estimates.

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.

Examples

```
x <- rnorm (20)
y <- rnorm (20)
fh <- ebvcdf (x, y)

fh (0, 0)
```

09_print_method *Print Method*

Description

Print method for bv (bivariate) objects.

Usage

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

Arguments

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

Details

This method calls the intoo::object.summary function.

Examples

```
f <- dubvpmf (1, 1, 4, 4)

print (f)
```

10_plot_methods *Plot Methods*

Description

Plot methods for bv (bivariate) objects.

Usage

```
#discrete uniform
## S3 method for class 'dubvpmf'
plot(x, plot.3d=FALSE, ..., xlim, ylim, all=FALSE)
## S3 method for class 'dubvcdf'
plot(x, plot.3d=FALSE, ..., xlim, ylim)

#binomial
## S3 method for class 'bnbvpmf'
plot(x, plot.3d=FALSE, ..., xlim, ylim, all=FALSE)
## S3 method for class 'bnbvcdf'
```

```
plot(x, plot.3d=FALSE, ..., xlim, ylim)

#poisson
## S3 method for class 'pbvpmf'
plot(x, plot.3d=FALSE, ..., xlim, ylim, all=FALSE)
## S3 method for class 'pbvcdf'
plot(x, plot.3d=FALSE, ..., xlim, ylim)

#categorical
## S3 method for class 'cbvpmf'
plot(x, plot.3d, ..., labels=TRUE)

#continuous uniform
## S3 method for class 'cubvpdf'
plot(x, plot.3d=FALSE, ..., xlim, ylim, n=20, all=FALSE)
## S3 method for class 'cubvcdf'
plot(x, plot.3d=FALSE, ..., xlim, ylim, n=20)

#normal
## S3 method for class 'nbvpdf'
plot(x, plot.3d=FALSE, ..., xlim, ylim, n=30, all=FALSE)
## S3 method for class 'nbvcdf'
plot(x, plot.3d=FALSE, ..., xlim, ylim, n=30)

#bimodal
## S3 method for class 'bmbvpdf'
plot(x, plot.3d=FALSE, ..., xlim, ylim, n=40, all=FALSE)
## S3 method for class 'bmbvcdf'
plot(x, plot.3d=FALSE, ..., xlim, ylim, n=40)

#dirichlet
## S3 method for class 'dtvpdf'
plot(x, plot.3d=FALSE, ..., n=30, log=FALSE)

#kernel density estimate
## S3 method for class 'kbvpdf'
plot(x, plot.3d=FALSE, ..., xlim, ylim, n=30,
    points=TRUE, point.color="#00000030")

#ecdf
## S3 method for class 'ebvcdf'
plot(x, plot.3d=FALSE, ..., as.smooth)

#additional plotting functions
#(called by the ebvcdf method, above)
plot_ebvcdf_step (x, plot.3d=FALSE, ...,
    steps=TRUE, points=TRUE, point.color="#00000030")
plot_ebvcdf_smooth (x, plot.3d=FALSE, ...,
```

```
    xlim, ylim, n=30)
```

Arguments

x	A bv (bivariate) object.
plot.3d	Logical value, if false, a 2D plot, if true, a 3D plot.
n	Integer, the number of grid points in each x and y direction. Note that in this package, "n", has more than one meaning.
xlim, ylim	The x and y ranges for the plot. Note that currently, the plot method for Poisson distributions, uses zero as the lower limits, regardless of the xlim and ylim values.
labels	Logical value, if true, include numeric labels.
log	Logical value, if true, plot the log density.
as.smooth	Logical value, if true, plot the probability distribution like a (continuous) surface. Defaults to false, if there's forty data points or less.
steps	Logical value, if true, plot the steps. Ignored if plot.3d is true.
points	Logical value, if true, plot the data points. Ignored if plot.3d is true.
point.color	String, the color for data points. Ignored if plot.3d is true.
all	Logical value, if true, plot a two by two grid of both the PMF/PDF and CDF using both 2D and 3D plots.
...	Other arguments for plotting functions in the barsurf package.

Details

These methods use the barsurf package, which contains plotting functions.

Note that the plot method for ebvcdf objects (bivariate ECDFs) calls either the plot_ebvcdf_step or plot_ebvcdf_smooth functions.

Examples

```
f <- dubvpmf (1, 1, 4, 4)

plot (f)
```

Index