Package 'bivariate'

December 10, 2018

Title Bivariate Probability Distributions

Version 0.3.0

Date 2018-12-10

License GPL (>= 2)

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

Author Abby Spurdle

URL https://sites.google.com/site/asrpws

Description Contains convenience functions for constructing and plotting bivariate probability distributions (probability mass functions, probability density functions and cumulative distribution functions). Currently, supports uniform (discrete and continuous), binomial, Poisson, normal, bimodal and kernel distributions.

Depends graphics

Imports barsurf, mvtnorm, KernSmooth

Suggests intoo, empirical, moments, MASS

NeedsCompilation no

Repository CRAN

Date/Publication 2018-12-10 09:20:06 UTC

R topics documented:

`bimodal` *Bimodal Distributions*

Description

Bimodal distributions. Refer to the vignette for better examples.

Usage

```
bmbvpdf (mean.X1, mean.Y1, variance.X1, variance.Y1,
    mean.X2, mean.Y2, variance.X2, variance.Y2,
    w=0.5)
bmbvcdf (mean.X1, mean.Y1, variance.X1, variance.Y1,
    mean.X2, mean.Y2, variance.X2, variance.Y2,
    w=0.5)
```

Arguments

`mean.X1`	Numeric scalar, giving the mean of the first X component.
`mean.Y1`	Numeric scalar, giving the mean of the first Y component.
`variance.X1`	Numeric scalar, giving the variance of the first X component.
`variance.Y1`	Numeric scalar, giving the variance of the first Y component.
`mean.X2`	Numeric scalar, giving the mean of the second X component.
`mean.Y2`	Numeric scalar, giving the mean of the second Y component.
`variance.X2`	Numeric scalar, giving the variance of the second X component.
`variance.Y2`	Numeric scalar, giving the variance of the second Y component.
`w`	Numeric scalar in the interval (0, 1), giving the weight of the first component.

Value

These functions return functions.

See Also

bmbvpdf.f, bmbvcdf.f

Examples

```
#construct some bimodal distributions
bmbvpdf.f = bmbvpdf (3.5, 0, 1, 1, 6.5, 0, 1, 1)
bmbvcdf.f = bmbvcdf (3.5, 0, 1, 1, 6.5, 0, 1, 1)
```

binomial *Binomial Distributions*

Description

Binomial distributions. Refer to the vignette for better examples.

Usage

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

Arguments

n	Integer scalar, giving the number of trials.
p.X	Numeric scalar, giving the probability of the first event.
p.Y	Numeric scalar, giving the probability of the second event.

Value

These functions return functions.

See Also

bnbvpmf.f, bnbvcdf.f

Examples

```
#construct some binomial distributions
bnbvpmf.f = bnbvpmf (10, 0.5, 0.5)
bnbvcdf.f = bnbvcdf (10, 0.5, 0.5)
```

kernel *Kernel Distributions*

Description

Probability density functions and cumulative distribution functions resulting from kernel smoothing. Refer to the vignette for better examples.

Usage

```
kbvpdf (x, y, bw.X, bw.Y)
kbvcdf (x, y, bw.X, bw.Y)
```

Arguments

x	The x values.
y	The y values.
bw.X	Numeric scalar, giving the X bandwidth.
bw.Y	Numeric scalar, giving the Y bandwidth.

Value

These functions return functions. However, currently they can't be evaluated.

Examples

```
#construct some kernel distributions
#(based on example in KernSmooth package)
data ("geyser", package="MASS")
kbvpdf.f = kbvpdf (geyser$duration, geyser$waiting, 0.7, 7)
kbvcdf.f = kbvcdf (geyser$duration, geyser$waiting, 0.7, 7)
```

---

methods *Methods*

Description

Bivariate probability distribution methods.

Usage

```
## S3 method for class 'dubvpmf'
plot(x, xlim, ylim, ...)
## S3 method for class 'dubvcdf'
plot(x, xlim, ylim, ...)
## S3 method for class 'bnbvpmf'
plot(x, xlim, ylim, ...)
## S3 method for class 'bnbvcdf'
plot(x, xlim, ylim, ...)
## S3 method for class 'pbvpmf'
plot(x, xmax, ymax, ...)
## S3 method for class 'pbvcdf'
plot(x, xmax, ymax, ...)
## S3 method for class 'cubvpdf'
plot(x, use.plot3d=FALSE, npoints=20,
    xlim, ylim, ..., all=FALSE)
## S3 method for class 'cubvcdf'
plot(x, use.plot3d=FALSE, npoints=20,
    xlim, ylim, ...)
## S3 method for class 'nbvpdf'
```

```
plot(x, use.plot3d=FALSE, npoints=20,
    xlim, ylim, ..., all=FALSE)
## S3 method for class 'nbvcdf'
plot(x, use.plot3d=FALSE, npoints=20,
    xlim, ylim, ...)
## S3 method for class 'bmbvpdf'
plot(x, use.plot3d=FALSE, npoints=30,
    xlim, ylim, ..., all=FALSE)
## S3 method for class 'bmbvcdf'
plot(x, use.plot3d=FALSE, npoints=30,
    xlim, ylim, ...)
## S3 method for class 'kbvpdf'
plot(x, use.plot3d=FALSE, npoints=30,
    xlim, ylim, ..., all=FALSE)
## S3 method for class 'kbvcdf'
plot(x, use.plot3d=FALSE, npoints=30,
    xlim, ylim, ...)
```

Arguments

x	A dubvpmf, dubvcdf, etc object.
use.plot3d	False by default. If true use 3d plots.
npoints	The number of grid points in each direction. Increase for a smoother surface.
xlim	The x range for the plot.
ylim	The y range for the plot.
xmax	The x range max for the plot.
ymax	The y range max for the plot.
all	False by default. If true, plot a two by two grid of both the PDF and CDF using both contour and 3d plots.
...	Other arguments.

Examples

```
#plot a bivariate probability distribution
dubvpmf.f = dubvpmf (0, 1, 0, 1)
plot (dubvpmf.f)
```

---

normal *Normal Distributions*

Description

Normal distributions. Refer to the vignette for better examples.

Usage

```
nbvpdf (mean.X, mean.Y, variance.X, variance.Y, covariance)
nbvcdf (mean.X, mean.Y, variance.X, variance.Y, covariance)
```

Arguments

mean.X	Numeric scalar, giving the mean of X.
mean.Y	Numeric scalar, giving the mean of Y.
variance.X	Numeric scalar, giving the variance of X.
variance.Y	Numeric scalar, giving the variance of Y.
covariance	Numeric scalar, giving the covariance of X and Y.

Value

These functions return functions.

See Also

nbvpdf.f, nbvcdf.f

Examples

```
#construct some normal distributions
nbvpdf.f = nbvpdf (0, 0, 1, 1, 0)
nbvcdf.f = nbvcdf (0, 0, 1, 1, 0)
```

---

poisson *Poisson Distributions*

Description

Poisson distributions. Refer to the vignette for better examples and references.

Usage

```
pbvpmf (mean.X, mean.Y, covariance)
pbvcdf (mean.X, mean.Y, covariance)
```

Arguments

mean.X	Numeric scalar, giving the mean (and variance) of X.
mean.Y	Numeric scalar, giving the mean (and variance) of Y.
covariance	Numeric scalar, giving the covariance of X and Y.

Value

These functions return functions.

See Also

pbvpmf.f, pbvcdf.f

Examples

```
#construct some Poisson distributions
pbvpmf.f = pbvpmf (10, 10, 0)
pbvcdf.f = pbvcdf (10, 10, 0)
```

---

returned_functions *Returned Functions*

Description

Returned functions. Note that you can name your functions whatever you like. (In the vignette I've named them f and F). Also note that you don't call these functions, rather call dubvpmf(), dubvcdf(), etc and then call the functions that are returned.

Usage

```
dubvpmf.f (x, y)
dubvcdf.f (x, y)
bnbvpmf.f (x, y)
bnbvcdf.f (x, y)
pbvpmf.f (x, y)
pbvcdf.f (x, y)
cubvpdf.f (x, y)
cubvcdf.f (x, y)
nbvpdf.f (x, y)
nbvcdf.f (x, y)
bmbvpdf.f (x, y)
bmbvcdf.f (x, y)
```

Arguments

x	A vector of x values.
y	A vector of y values.

Examples

```
#construct and evaluate a discrete uniform distribution
dubvpmf.f = dubvpmf (0, 1, 0, 1)
dubvpmf.f (0.5, 0.5)
```

uniform *Discrete and Continuous Uniform Distributions*

Description

Discrete and continuous uniform distributions. Refer to the vignette for better examples.

Usage

```
dubvpmf (a.X, b.X, a.Y, b.Y)
dubvcdf (a.X, b.X, a.Y, b.Y)
cubvpdf (a.X, b.X, a.Y, b.Y)
cubvcdf (a.X, b.X, a.Y, b.Y)
```

Arguments

a.X	Numeric scalar, giving the a value of X.
b.X	Numeric scalar, giving the b value of X.
a.Y	Numeric scalar, giving the a value of Y.
b.Y	Numeric scalar, giving the b value of Y.

Value

These functions return functions.

See Also

dubvpmf.f, dubvcdf.f, cubvpdf.f, cubvcdf.f

Examples

```
#construct some discrete uniform distributions
dubvpmf.f = dubvpmf (0, 1, 0, 1)
dubvcdf.f = dubvcdf (0, 1, 0, 1)
```

Index