Package 'probhat'

March 5, 2019

Title Generalized Kernel Smoothing

Version 0.1.1

Date 2019-02-28

License GPL (>= 2)

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

Author Abby Spurdle

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

Description Computes nonparametric probability distributions (probability density functions, cumulative distribution functions and quantile functions) using kernel smoothing. Supports univariate, multivariate and conditional distributions, and weighted data (possibly useful mixed with fuzzy clustering or frequency data). Also, supports empirical continuous cumulative distribution functions and their inverses, and random number generation.

Depends graphics, stats

Imports barsurf

Suggests intoo, bivariate, mvtnorm, fclust, moments, scatterplot3d

NeedsCompilation no

Repository CRAN

Date/Publication 2019-03-05 22:20:04 UTC

R topics documented:

rd_1_sbc *Simplified Bell Curves*

Description

Kernel PDF and CDF.

Usage

```
sbc.cdf (x)
sbc.pdf (x)
```

Arguments

x	Numeric vector.

Details

Refer to the vignette for more information.

Examples

```
sbc.pdf (0)
```

rd_2_nonparametric *Nonparametric Models*

Description

Construct nonparametric probability distributions (that use kernel smoothing), empirical continuous cumulative distribution functions and their inverses, and regression-like models. Note that the objects are functions, which means that they can be evaluated.

Usage

```
#univariate
nppdfuv (x, spline = TRUE,
    kernel.pdf = sbc.pdf, kernel.cdf = NA,
    nc = 30, smoothness = 0.65, bw, w = NA)
npcdfuv (x, spline = TRUE,
    kernel.pdf = NA, kernel.cdf = sbc.cdf,
    nc = 30, smoothness = 0.65, bw, w = NA)
npcdfuv.inverse (x, kernel.pdf = NA, kernel.cdf = sbc.cdf,
    nc = 30, bw, smoothness = 0.65, w = NA)

#multivariate
```

```
nppdfmv (x, kernel.pdf = sbc.pdf, kernel.cdf = sbc.cdf,
    smoothness = 0.65, bw, w = NA)
npcdfmv (x, kernel.pdf = sbc.pdf, kernel.cdf = sbc.cdf,
    smoothness = 0.65, bw, w = NA)
chained.npcdfmv.inverse (
    x, kernel.pdf = sbc.pdf, kernel.cdf = sbc.cdf,
    smoothness = 0.65, bw, w = NA)

#conditional
nppdfc (model, x, spline = TRUE,
    kernel.pdf = sbc.pdf, kernel.cdf = sbc.cdf,
    nc = 30, smoothness = 0.65, bw, w = NA)
npcdfc (model, x, spline = TRUE,
    kernel.pdf = sbc.pdf, kernel.cdf = sbc.cdf,
    nc = 30, smoothness = 0.65, bw, w = NA)
npcdfc.inverse (model, x, kernel.pdf = sbc.pdf, kernel.cdf = sbc.cdf,
    nc = 30, smoothness = 0.65, bw, w = NA)

#empirical
eccdf (x, w = NA)
eccdf.inverse (x, w = NA)

#regression-like
reglike (model, x, kernel.pdf = sbc.pdf, kernel.cdf = sbc.cdf,
    nc = 30, nc.npc = nc, smoothness = 0.65, bw, w = NA)
```

Arguments

model	For conditional distributions, either a numeric vector giving the conditions, or a formula object. For regression-like models, a formula object. If it's a numeric vector, then it should have length m - 1, for the first m - 1 variables.
x	For univariate distributions (including empirical continuous cumulative distribution functions and their inverses), x should be a numeric vector. Otherwise, x should be a matrix with n rows and m columns. Where rows represent data points (or observations) and columns represent variables.
spline	If true (the default), use splines as intermediate models.
kernel.pdf	A function giving the kernel PDF.
kernel.cdf	A function giving the kernel CDF.
nc	The number of control points in the spline.
nc.npc	In regression-like models, the number of control points used in the conditional distributions.
smoothness	A smoothness parameter of length one or m.
bw	A bandwidth parameter of length one or m. If provided, the smoothness parameter is ignored. If missing, it's computed from the smoothness parameter.
w	A numeric vector of weights that sum to approximately one.

Details

Refer to the vignette for more information.

Examples

```
data (trees)
nppdfuv.f = nppdfuv (trees$Height)
nppdfuv.f (80)
```

`rd_3_marginal` *Marginal Distributions*

Description

Constructs a marginal object, which is a list of univariate distributions.

Usage

```
marginal (npmv)
```

Arguments

npmv	An nppdfmv or npcdfmv object.

Details

Refer to the vignette for more information.

Examples

```
data (trees)
trees = as.matrix (trees)
nppdfmv.f = nppdfmv (trees)
muv = marginal (nppdfmv.f)
muv [[1]]
```

rd_4_most_plots *Most Plots*

Description

Plots of most objects in this package.

Usage

```
#univariate
## S3 method for class 'nppdfuv'
plot(x, with.points = FALSE, nmax = 2000, ...)
## S3 method for class 'npcdfuv'
plot(x, with.points = FALSE, nmax = 2000, ...)
## S3 method for class 'npcdfuv.inverse'
plot(x, ...)
## S3 method for class 'nppdfuv'
lines(x, ...)
## S3 method for class 'npcdfuv'
lines(x, ...)
## S3 method for class 'npcdfuv.inverse'
lines(x, ...)

#conditional
## S3 method for class 'nppdfc'
plot(x, ...)
## S3 method for class 'npcdfc'
plot(x, ...)
## S3 method for class 'npcdfc.inverse'
plot(x, ...)
## S3 method for class 'nppdfc'
lines(x, ...)
## S3 method for class 'npcdfc'
lines(x, ...)
## S3 method for class 'npcdfc.inverse'
 lines(x, ...)

#empirical
## S3 method for class 'eccdf'
plot(x, with.points = FALSE, ...)
## S3 method for class 'eccdf.inverse'
plot(x, with.points = FALSE, ...)
## S3 method for class 'eccdf'
lines(x, ...)
## S3 method for class 'eccdf.inverse'
lines(x, ...)
```

```
#regression-like
## S3 method for class 'reglike'
plot(x, with.points = FALSE, ...)
## S3 method for class 'reglike'
lines(x, ...)
```

Arguments

x	An nppdfuv, npcdfuv, npcdfuv.inverse, nppdfc, npcdfc, npcdfc.inverse, eccdf, eccdf.inverse or reglike object.
with.points	If true, plot the data points.
nmax	If n <= nmax, plot all the data points. Otherwise, plot a random sample of the data points, with sample size = nmax.
...	Other arguments.

Details

Refer to the vignette for more information.

Examples

```
data (trees)
nppdfuv.f = nppdfuv (trees$Height)
plot (nppdfuv.f, TRUE)
```

---

rd_5_plots_bivariate *Plots of Bivariate Distributions*

Description

Plots of bivariate nonparametric distributions.

Usage

```
## S3 method for class 'nppdfmv'
plot(x, use.plot3d = FALSE,
    xlab, ylab, npoints = 30, ..., all = FALSE)
## S3 method for class 'npcdfmv'
plot(x, use.plot3d = FALSE,
    xlab, ylab, npoints = 30, ...)
## S3 method for class 'chained.npcdfmv.inverse'
plot(x, ...)
```

Arguments

x	An nppdfmv or npcdfmv object.
use.plot3d	If true, use a 3d plot. Otherwise, use a contour plot.
xlab	The x axis label.
ylab	The y axis label.
npoints	The number of grid points in each direction.
all	If true, plot all combinations.
...	Other arguments.

Details

Refer to the vignette for more information. Note that only bivariate nppdfmv and bivariate npcdfmv objects (with m = 2) can be plotted. Other objects produce error messages.

Examples

```
data (trees)

#matrix with height and volume
trees.sub = as.matrix (trees)[,2:3]

nppdfmv.f = nppdfmv (trees.sub)
plot (nppdfmv.f)
```

---

rd_6_plots_marginal *Plots of Marginal Distributions*

Description

Plots a marginal object.

Usage

```
## S3 method for class 'marginal'
plot(x, nrow, ncol, with.points = FALSE, ...)
```

Arguments

x	A marginal object.
nrow	Number of rows.
ncol	Number of columns.
with.points	If true, plot the data points.
...	Other arguments.

Details

Refer to the vignette for more information. Note that nrow and ncol may be omitted. However, the plot layout may be suboptimal.

Examples

```
data (trees)
trees = as.matrix (trees)
nppdfmv.f = nppdfmv (trees)
muv = marginal (nppdfmv.f)
plot (muv, 3, 1, TRUE)
```

---

rd_7_combinatoric_probability

Combinatoric Probability

Description

Compute probabilities (including multivariate probabilities) by evaluating CDFs, combinatorically.

Usage

```
comb.prob (cdf.f, a, b)
```

Arguments

cdf.f	A CDF.
a	A vector (or matrix) of lower limits. If a matrix, then each row represents one region and each column represents one variable.
b	Same as a, except upper limits.

Details

Refer to the vignette for more information.

Examples

```
data (trees)
trees = as.matrix (trees)
npcdfmv.f = npcdfmv (trees)

summary (trees)

#approximate first and third quartiles
#marginal probabilities are approximately 0.5 each
a = c (11.05, 72, 19.40)
b = c (15.25, 80, 37.30)
```

```
#however, multivariate probability is not necessarily close to 0.5 ^ 3
comb.prob (npcdfmv.f, a, b)
```

---

rd_8_stats *Statistics and Random Number Generation*

Description

Compute the mean and mode from nonparametric PDFs, including conditional distributions. Also, generate random numbers (or random samples) using inversion sampling.

Usage

```
npmean (nppdfuv.f)
npmode (nppdfuv.f, include.boundaries = TRUE,
    all = FALSE, warning = FALSE)
nprng (npcdf.f.inverse, n)
```

Arguments

nppdfuv.f	An nppdfuv object.
npcdf.f.inverse	An npcdfuv.inverse or chained.npcdfmv.inverse object.
include.boundaries	Include the boundary points, if they're modal points, which isn't possible without modifying the splines.
all	Return all modal points.
warning	Produce a warning, if zero or multiple modal points.
n	Number of random numbers.

Details

Refer to the vignette for more information.

Examples

```
data (trees)
npmode (nppdfuv (trees$Height) )
```

Index