Package ‘dirdensity’

December 22, 2023

Title 3D Directional Density Models and Representation

Version 0.0.0.1

Author Maria Alonso-Pena [aut, cre],
Jose Ameijeiras-Alonso [aut]

Maintainer Maria Alonso-Pena <maria.alonsopena@ugr.es>

Description Computation of density functions defined on the surface of a sphere, a torus or a cylinder. It implements different models presented in Mardia and Jupp (1999) <doi:10.1002/9780470316979>, Ley and Verdebout (2017) <doi:10.1201/9781315119472> and Ameijeiras-Alonso and Ley (2022) <doi:10.1093/biostatistics/kxaa039>. The package also provides tools to represent such functions graphically, with 3D interactive plots.

License GPL-3

Imports rgl, Directional, DirStats, circular, movMF, misc3d

Encoding UTF-8

Roxygen list(markdown = TRUE)

RoxygenNote 7.2.3

NeedsCompilation no

R topics documented:

cyl.density.plot *cyl.density.plot*

Description

Function `cyl.density.plot` creates an interactive 3D representation of a given cylindrical density.

Usage

```
cyl.density.plot(x, fx, axis = TRUE)
```

Arguments

`x`	Matrix with three columns containing the points where the density is evaluated in cartesian coordinates.
`fx`	Vector containing the values of the density. The length must coincide with the number of rows of x.
`axis`	Logical; if TRUE, the axis are plotted.

Examples

```
N<-600
u = seq(0, 2 * pi, length.out = N)
v = seq(0,1, length.out = N)
ll<-expand.grid(u,v)
u2<-ll[,1]
v2<-ll[,2]
T<-diff(range(v))/4
x<-cbind(v2,  T * cos(u2), T * sin(u2))
fx<-dirdensity:::dms(u2, v2, mu=c(pi,0), kappa=5, sigma=0.5, nu=0, lambda=1)
cyl.density.plot(x,fx)
```

cyl.plot *cyl.plot*

Description

Function `cyl.plot` creates an interactive 3D representation of cylindrical data on the surface of a cylinder.

Usage

```
cyl.plot(
  x,
  col.points = "red",
  pch = 16,
  size = 4,
  axis = TRUE,
  zoom = 0.9,
  userMatrix = NULL,
  windowRect = NULL,
  range = NULL,
  T = NULL,
  col.cyl = "honeydew1",
  labelsx = NULL,
  labelsy = NULL
)
```

Arguments

`x`	Matrix with two columns containing the data in cylindrical coordinates, where the number of rows is the number of observations. The first column contains the azimuthal angle (in radians), while the second column contains the height.
`col.points`	Vector containing the mean direction of the von Mises-Fisher guide.
`pch`	Type of points to be plotted.
`size`	Numerical value containing the concentration of the von Mises-Fisher guide.
`axis`	Logical; if TRUE, axis are plotted.
`zoom`	Parameter controlling the zoom in the plot.
`userMatrix`	A 4 by 4 matrix describing user actions to display the scene. See par3d.
`windowRect`	A vector of four values indicating the left, top, right and bottom of the displayed window (in pixels). Applies to the whole device. See par3d.
`range`	Vector with two components giving the interval of the real-line where height of the cylinder should be displayed.
`T`	If plot=TRUE, parameter controlling the radius of the cylinder.
`col.cyl`	Character string indicating the desired color of the cylinder.
`labelsx`	Numeric vector containing the values to be displayed as labels for the circular component.
`labelsy`	Numeric vector containing the values to be displayed as labels for the linear component.

Examples

```
library(circular)
a1<-rvonmises(150,0,5)
a2<-runif(150,2,10)
x<-cbind(a1,a2)
cyl.plot(x)
```

`dAL.cyl` *dAL.cyl*

Description

Function `dAL.cyl` computes the density function of a Abe-Ley density and produces an interactive 3D representation of the density on the surface of a cylinder.

Usage

```
dAL.cyl(
  mu,
  lambda,
  beta,
  alpha,
  kappa,
  rangex,
  T = NULL,
  plot = TRUE,
  axis = TRUE,
  orientation = "horiz"
)
```

Arguments

`mu`	Circular location parameter.
`lambda`	Skewness parameter (must belong to (-1,1)).
`beta`	Linear dispersion (has to be positive).
`alpha`	Shape parameter (has to be positive).
`kappa`	Circular concentration, which acts also as dependence parameter (greater or equal than zero).
`rangex`	Vector with two components giving the interval of the real-line where linear part of the density should be evaluated.
`T`	If plot=TRUE, parameter controlling the radius of the cylinder.
`plot`	Logical; if TRUE, the 3D plot is produced.
`axis`	Logical; if TRUE, axis are plotted.
`orientation`	If plot=TRUE, character string indicating the orientation of the cylinder. Must be "horiz" or "vert".

Value

An list containing the following components:

`fx`	The estimated values of the density.
`eval.points`	The points where the estimated density was evaluated.

References

Abe, T. and Ley, C. (2017) 'A tractable, parsimonious and flexible model for cylindrical data, with applications', Econometrics and Statistics, 4, 91-104.

Examples

```
al<-dAL.cyl(mu=0,lambda=-0.5,kappa=1,beta=2,alpha=3,orientation="vert",rangex=c(0,3),T=1.5)
```

dcos.torus *dcos.torus*

Description

Function `dcos.torus` computes the density function of a bivariate cosine model and produces an interactive 3D representation of the density on the surface of a cylinder.

Usage

```
dcos.torus(mu, kappa, rho, plot = TRUE, axis = TRUE)
```

Arguments

`mu`	Vector with two components representing the mean direction of each marginal model.
`kappa`	Vector with two components representing the concentration of each marginal model.
`rho`	Numeric element containing the correlation between the two components.
`plot`	Logical; if TRUE, the 3D plot is produced.
`axis`	Logical; if TRUE, axis are plotted.

Value

An list containing the following components:

`fx`	The estimated values of the density.
`eval.points`	The points where the estimated density was evaluated.

References

Mardia, K. V., Taylor, C. C. & Subramaniam, G. K. (2007), ‘Protein bioinformatics and mixtures of bivariate von Mises distributions for angular data’, Biometrics 63, 505–512.

Examples

```
co<-dcos.torus(mu=c(-pi/16,pi/4), kappa=c(4,5), rho=3)
```

dJW.cyl *dJW.cyl*

Description

Function dJW.cyl computes the density function of a Jones-Wehrly density and produces an interactive 3D representation of the density on the surface of a cylinder.

Usage

```
dJW.cyl(
  mu,
  lambda,
  kappa,
  rangex,
  T = NULL,
  plot = TRUE,
  axis = TRUE,
  orientation = "horiz"
)
```

Arguments

mu	Circular location parameter.
lambda	Linear dispersion (positive).
kappa	Dependence parameter (has to be less than lambda).
rangex	Vector with two components giving the interval of the real-line where linear part of the density should be evaluated.
T	If plot=TRUE, parameter controlling the radius of the cylinder.
plot	Logical; if TRUE, the 3D plot is produced.
axis	Logical; if TRUE, axis are plotted.
orientation	If plot=TRUE, character string indicating the orientation of the cylinder. Must be "horiz" or "vert".

Value

An list containing the following components:

fx	The estimated values of the density.
eval.points	The points where the estimated density was evaluated.

References

Johnson, R. A. & Wehrly, T. E. (1978), ‘Some angular-linear distributions and related regression models’, J. Amer. Statist. Assoc. 73, 602–606.

Examples

```
jw<-dJW.cyl(mu=pi/2,lambda=2,kappa=1.25,orientation="horiz",rangex=c(0,5))
```

dkent.sph *dkent.sph*

Description

Function dkent.sph computes the density function of a Kent density of dimension 2 and produces an interactive 3D representation of the density on the surface of a sphere.

Usage

```
dkent.sph(mu, kappa, beta, G, plot = TRUE, axis = TRUE)
```

Arguments

mu	Vector with three components giving the mean of the density in cartesian coordinates.
kappa	Parameter controlling the concentration of the density.
beta	Parameter controlling the ovalness of the density.
G	A 3 x 3 matrix with first column equal to the mean direction. The second and third columns are the major and minor axes respectively.
plot	Logical; if TRUE, the 3D plot is produced.
axis	Logical; if TRUE, axis are plotted.

Value

An list containing the following components:

fx	The estimated values of the density.
eval.points	The points where the estimated density was evaluated.

References

Kent John (1982). The Fisher-Bingham distribution on the sphere. Journal of the Royal Statistical Society, Series B, 44(1): 71–80. Mardia, K. V. & Jupp, P. E. (2000), Directional Statistics, Wiley, New York.

Examples

```
ken<-dkent.sph(mu=c(-1,0,0),kappa=5,beta=3,G=cbind(c(-1,0,0),c(0,1,0),c(0,0,1)))
```

dmixvMF.sph *dmixvMF.sph*

Description

Function dmixvMF.sph computes the density function of a mixture of von Mises-Fisher densities, of dimension 2, and produces an interactive 3D representation of the density on the surface of a sphere.

Usage

```
dmixvMF.sph(mu, kappa, probs, plot = TRUE, axis = TRUE)
```

Arguments

mu	Matrix with three columns and number of rows equal to the number of components in the mixture, giving the mean vectors for all components.
kappa	Vector of parameters controlling the concentration of each component.
probs	Vector of parameters controlling the mixing probabilities.
plot	Logical; if TRUE, the 3D plot is produced.
axis	Logical; if TRUE, axis are plotted.

Value

An list containing the following components:

fx	The estimated values of the density.
eval.points	The points where the estimated density was evaluated.

References

Kurt Hornik and Bettina Grun (2014). movMF: An R Package for Fitting Mixtures of von Mises-Fisher Distributions http://cran.r-project.org/web/packages/movMF/vignettes/movMF.pdf

Examples

```
mixvm<-dmixvMF.sph(mu=matrix(c(0,0,1,0,0,-1),ncol=3,byrow=TRUE),kappa=c(5,5),probs=c(0.6,0.4))
```

dMS.cyl *dMS.cyl*

Description

Function dMS.cyl computes the density function of a Mardia-Sutton density and produces an interactive 3D representation of the density on the surface of a cylinder.

Usage

```
dMS.cyl(
  mu,
  kappa,
  sigma,
  nu,
  lambda,
  rangex = NULL,
  T = NULL,
  plot = TRUE,
  axis = TRUE,
  orientation = "horiz"
)
```

Arguments

mu	Vector with two components: the first one is the circular mean and the second one is the linear mean.
kappa	Concentration parameter (greater or equal than zero).
sigma	Standard deviation (has to be positive).
nu	Parameter between -pi and pi.
lambda	Structure parameter; has to be greater or equal than zero
rangex	Vector with two components giving the interval of the real-line where linear part of the density should be evaluated.
T	If plot=TRUE, parameter controlling the radius of the cylinder.
plot	Logical; if TRUE, the 3D plot is produced.
axis	Logical; if TRUE, axis are plotted.
orientation	If plot=TRUE, character string indicating the orientation of the cylinder. Must be "horiz" or "vert".

Value

An list containing the following components:

fx	The estimated values of the density.
eval.points	The points where the estimated density was evaluated.

References

Mardia, K. V. & Sutton, T. W. (1978), ‘A model for cylindrical variables with applications’, J. Roy. Stat. Soc. Ser. B 40, 229–233.

Examples

```
ms<-dMS.cyl(mu=c(pi,0),kappa=5,sigma=0.5,nu=1,lambda=3,rangex=c(-1,6),orientation="vert")
```

dsin.torus *dsin.torus*

Description

Function `dsin.torus` computes the density function of a bivariate sine model and produces an interactive 3D representation of the density on the surface of a cylinder.

Usage

```
dsin.torus(mu, kappa, rho, plot = TRUE, axis = TRUE)
```

Arguments

`mu`	Vector with two components representing the mean direction of each marginal model.
`kappa`	Vector with two components representing the concentration of each marginal model.
`rho`	Numeric element containing the correlation between the two components.
`plot`	Logical; if TRUE, the 3D plot is produced.
`axis`	Logical; if TRUE, axis are plotted.

Value

An list containing the following components:

`fx`	The estimated values of the density.
`eval.points`	The points where the estimated density was evaluated.

References

Singh, H., Hnizdo, V. & Demchuk, E. (2002), ‘Probabilistic model for two dependent circular variables’, Biometrika 89, 719–723.

Examples

```
si<-dsin.torus(mu=c(-pi/16,pi/4), kappa=c(4,5), rho=3)
```

`dsinsk.torus` *dsinsk.torus*

Description

Function `dsinsk.torus` computes the density function of a sine-skewed toroidal distribution and produces an interactive 3D representation of the density on the surface of a cylinder.

Usage

```
dsinsk.torus(mu, kappa, rho, lambda, model = NULL, plot = TRUE, axis = TRUE)
```

Arguments

`mu`	Vector with two components representing the mean direction of each marginal model.
`kappa`	Vector with two components representing the concentration of each marginal model.
`rho`	Numeric element containing the correlation between the two components.
`lambda`	Vectorwith two components representing the skewness coefficients.
`model`	Character value indicating a predefined toroidal symmetric density. Implemented examples include model="sine", model="cosine", model="bwc".
`plot`	Logical; if TRUE, the 3D plot is produced.
`axis`	Logical; if TRUE, axis are plotted.

Value

An list containing the following components:

`fx`	The estimated values of the density.
`eval.points`	The points where the estimated density was evaluated.

References

Jose Ameijeiras-Alonso, Christophe Ley, Sine-skewed toroidal distributions and their application in protein bioinformatics, Biostatistics, Volume 23, Issue 3, July 2022, Pages 685–704.

Examples

```
ssk<-dsinsk.torus(mu=c(-pi/2,pi/8),kappa=c(5,5),rho=3,lambda=c(-0.9,0.1),model="sine")
```

dvMF.sph *dvMF.sph*

Description

Function dvMF.sph computes the density function of a von Mises-Fisher density of dimension 2 and produces an interactive 3D representation of the density on the surface of a sphere.

Usage

```
dvMF.sph(mu, kappa, plot = TRUE, axis = TRUE)
```

Arguments

mu	Vector with three components giving the mean of the density in cartesian coordinates.
kappa	Parameter controlling the concentration of the density.
plot	Logical; if TRUE, the 3D plot is produced.
axis	Logical; if TRUE, axis are plotted.

Value

An list containing the following components:

fx	The estimated values of the density.
eval.points	The points where the estimated density was evaluated.

References

Mardia, K. V. & Jupp, P. E. (2000), Directional Statistics, Wiley, New York.

Examples

```
vm<-dvMF.sph(c(1,0,0),5)
```

dwc.torus *dwc.torus*

Description

Function dcos.torus computes the density function of a bivariate wrapped Cauchy and produces an interactive 3D representation of the density on the surface of a cylinder.

Usage

```
dwc.torus(mu, kappa, rho, plot = TRUE, axis = TRUE)
```

Arguments

`mu`	Vector with two components representing the mean direction of each marginal model.
`kappa`	Vector with two components representing the concentration of each marginal model.
`rho`	Numeric element containing the correlation between the two components.
`plot`	Logical; if TRUE, the 3D plot is produced.
`axis`	Logical; if TRUE, axis are plotted.

Value

An list containing the following components:

`fx`	The estimated values of the density.
`eval.points`	The points where the estimated density was evaluated.

References

Kato, S. & Pewsey, A. (2015), 'A Möbius transformation-induced distribution on the torus', Biometrika 102, 359–370

Examples

```
wc<-dwc.torus(mu=c(pi/2,0), kappa=c(0.5,0.15), rho=0.7)
```

`sph.density.plot` *sph.density.plot*

Description

Function `sph.density.plot` creates an interactive 3D representation of a given spherical density.

Usage

```
sph.density.plot(x, fx, axis = TRUE)
```

Arguments

`x`	Matrix with three columns containing the points where the density is evaluated in cartesian coordinates.
`fx`	Vector containing the values of the density. The length must coincide with the number of rows of x.
`axis`	Logical; if TRUE, the axis are plotted.

Examples

```
# Constructing a grid on the sphere
N<-300
ele_grid<-seq(0,pi,length=N)
ori<-list()
ori[[1]]<-0
ori[[N]]<-0
for (i in 2:(N-1)){
  ori[[i]]<- seq(0,2*pi,length=floor(2*N*sin(ele_grid[i])))
}
lonxi<-as.numeric(lapply(ori,length))
both<-cbind(unlist(ori),rep(ele_grid,times=lonxi))
x<-DirStats::to_sph(both[,1], both[,2]) # conversion of coordinates
fx<-Directional::iagd(x,mu=c(1,0,0))
sph.density.plot(x,fx)
```

sph.plot *sph.plot*

Description

Function sph.plot creates an interactive 3D representation of spherical data on a sphere.

Usage

```
sph.plot(
  x,
  col.points = "red",
  pch = 16,
  size = 4,
  axis = TRUE,
  zoom = 0.8,
  windowRect = NULL,
  col.sph = "honeydew1"
)
```

Arguments

x Matrix containing the data in cartesian coordinates, where the number of columns must be 3 and the number of rows is the number of observations..

col.points Vector containing the mean direction of the von Mises-Fisher guide.

pch Type of points to be plotted.

size Numerical value containing the concentration of the von Mises-Fisher guide.

axis Logical; if TRUE, the axis are plotted.

zoom Parameter controlling the zoom in the plot.

windowRect aa.

col.sph Character string indicating the desired color of the sphere.

Examples

```
 library(movMF)
n<-200
mu<-matrix(c(0,0,1,0,0,-1),ncol=3,byrow=TRUE)
k<-c(7,2)
probs<-c(0.85,0.15)
x<-rmovMF(n,k*mu,alpha=probs)
sph.plot(x,2,4)
```

`torus.density.plot` *torus.density.plot*

Description

Function `torus.density.plot` creates an interactive 3D representation of a given toroidal density.

Usage

```
torus.density.plot(x, fx, axis = TRUE)
```

Arguments

`x`	Matrix with three columns containing the points where the density is evaluated in cartesian coordinates.
`fx`	Vector containing the values of the density. The length must coincide with the number of rows of x.
`axis`	Logical; if TRUE, the axis are plotted.

Examples

```
N<-600
u = seq(0, 2 * pi, length.out = N)
v = seq(0, 2 * pi, length.out = N)
ll<-expand.grid(u,v)
u2<-ll[,1]
v2<-ll[,2]
x<-cbind((1 + 0.5 * cos(v2)) * cos(u2), (1 + 0.5 * cos(v2)) * sin(u2), 0.5 * sin(v2))
fx<-dirdensity:::dbwc(u2,v2,mu1=pi,mu2=0,kappa1=0.5,kappa2=0.15,rho=0.5)
torus.density.plot(x, fx)
```

`torus.plot` *torus.plot*

Description

Function `torus.plot` creates an interactive 3D representation of toroidal data on the surface of a torus.

Usage

```
torus.plot(
  x,
  col.points = "red",
  pch = 16,
  size = 4,
  axis = TRUE,
  zoom = 0.7,
  userMatrix = NULL,
  windowRect = NULL,
  col.torus = "honeydew1",
  labelsx = NULL
)
```

Arguments

`x`	Matrix with two columns containing the data, where the number of rows is the number of observations. The first column contains the azimuthal ange, while the second column contains the polar angle (both in radians).
`col.points`	Vector containing the mean direction of the von Mises-Fisher guide.
`pch`	Type of points to be plotted.
`size`	Numerical value containing the concentration of the von Mises-Fisher guide.
`axis`	Logical; if TRUE, axis are plotted.
`zoom`	Parameter controlling the zoom in the plot.
`userMatrix`	A 4 by 4 matrix describing user actions to display the scene. See `par3d`.
`windowRect`	A vector of four values indicating the left, top, right and bottom of the displayed window (in pixels). Applies to the whole device. See `par3d`.
`col.torus`	Character string indicating the desired color of the torus.
`labelsx`	Numeric vector containing the values to be displayed as labels for the azimuth component.

Examples

```
library(circular)
a1<-rvonmises(150,0,5)
a2<-rvonmises(150,0,15)
x<-cbind(a1,a2)
torus.plot(x)
```

Index