Package ‘rsdepth’

October 14, 2022

Type Package

Title Ray Shooting Depth (i.e. RS Depth) Functions for Bivariate Analysis

Version 0.1-22

Date 2022-04-18

Author Mudassir Shabbir [aut, cre], Nabil Mustafa [aut], Saurabh Ray [aut]

Maintainer Mudassir Shabbir <mudassir.shabbir@gmail.com>

Description Ray Shooting Depth functions are provided for bivariate analysis. This mainly includes functions for computing the bivariate depth as well as RS median. Drawing functions for depth bags are also provided.

Depends R (>= 2.4.0)

License GPL-2

NeedsCompilation yes

Repository CRAN

Date/Publication 2022-04-25 10:30:02 UTC

R topics documented:

 centroid .............................................................. 2
 convexhull ............................................................ 3
 drawcompletegraph .................................................. 4
 getbag ................................................................. 5
 inflate ................................................................. 6
 rsdepth ................................................................. 7
 rsmed ................................................................. 8
 rsplot ................................................................. 9
 rsrings ............................................................ 10
 rstinterval ....................................................... 11

Index 13
centroid

Centroid of a convex polygon

Description

Computes Centroid of a convex polygon in plane.

Usage

`centroid(x, y=NULL,...)`

Arguments

- `x` The data as a matrix, data frame or list. If it is a matrix or data frame, then each row is viewed as one bivariate observation. If it is a list, all components must be numerical vectors of equal length (coordinates of observations).
- `y` The data as a matrix, data frame or list. If it is a matrix or data frame, then each row is viewed as one bivariate observation. If it is a list, all components must be numerical vectors of equal length (coordinates of observations).
- `...` For future use.

Details

In dimension 2, calculates centroid of a convex polygon.

Value

Returns with respect to data set, the centroid point in plane.

Author(s)

Nabil Mustafa, Saurabh Ray, and Mudassir Shabbir.

References


See Also

`inflate`

Examples

```R
## calculation of centroid of a random pointset
z = matrix(rnorm(24),nc=2)
x = centroid(z)
```
convexhull

Convex Hull of a pointset

Description
Convex Hull of a pointset in plane.

Usage
convexhull(x, y=NULL,...)

Arguments
x
The data as a matrix, data frame or list. If it is a matrix or data frame, then each row is viewed as one bivariate observation. If it is a list, all components must be numerical vectors of equal length (coordinates of observations).

y
The data as a matrix, data frame or list. If it is a matrix or data frame, then each row is viewed as one bivariate observation. If it is a list, all components must be numerical vectors of equal length (coordinates of observations).

...
For future use.

Details
In dimension 2, calculates Convex Hull of a pointset.

Value
Returns with respect to data set, ordered set of points on the convex hull.

Author(s)
Nabil Mustafa, Saurabh Ray, and Mudassir Shabbir.

References

See Also
inflate

Examples
## calculation of centroid of a random pointset
z = matrix(rnorm(24),nc=2)
x = convexhull(z)
**drawcompletegraph**  
*Draws Complete Graph of a pointset*

**Description**  
Draws Complete Graph of a pointset in plane.

**Usage**  
```r
drawcompletegraph(x, y=NULL, startcanvas=TRUE, ...)
```

**Arguments**

- `x`  
The data as a matrix, data frame or list. If it is a matrix or data frame, then each row is viewed as one bivariate observation. If it is a list, all components must be numerical vectors of equal length (coordinates of observations).

- `y`  
The data as a matrix, data frame or list. If it is a matrix or data frame, then each row is viewed as one bivariate observation. If it is a list, all components must be numerical vectors of equal length (coordinates of observations).

- `startcanvas`  
A boolean value to let the function whether there is already a plot that we want to use or create a new canvas. Be default set to TRUE.

- `...`  
For future use.

**Details**  
In dimension 2, draws complete graph on a pointset.

**Value**  
Returns nothing.

**Author(s)**  
Nabil Mustafa, Saurabh Ray, and Mudassir Shabbir.

**References**  

**See Also**

- `inflate`

**Examples**

```r
# calculation of centroid of a random pointset
z = matrix(rnorm(24),nc=2)
x = drawcompletegraph(z)
```
Description

Computes the Ray Shooting depth ISO of a point with respect to a bivariate data set.

Usage

getbag(x, y=NULL, factorsecondbag=2,...)

Arguments

x
The data as a matrix, data frame or list. If it is a matrix or data frame, then each row is viewed as one bivariate observation. If it is a list, all components must be numerical vectors of equal length (coordinates of observations).

y
The data as a matrix, data frame or list. If it is a matrix or data frame, then each row is viewed as one bivariate observation. If it is a list, all components must be numerical vectors of equal length (coordinates of observations).

factorsecondbag
Factor of the second bag. Takes integer values. By default set to 2.

... For future use.

Details

In dimension 2, calculates ray shooting depth of a given point with respect to the point set. Time complexity of the simple algorithms implemented is $O(n \log n)$. ISO

Value

Returns with respect to data set pt, the number of line segments interested by a ray from , minimum over all rays. ISO

Author(s)

Nabil Mustafa, Saurabh Ray, and Mudassir Shabbir.

References


See Also

rsdepth
## Calculation of RS Depth

```
# calculation of RS depth
z = matrix(rnorm(24), nc=2)
x = getbag(z)
```

### Description

Inflates a convex polygon

### Usage

```
inflate(x, y=NULL, factor=2, ...)
```

### Arguments

- **x**: The data as a matrix, data frame or list. If it is a matrix or data frame, then each row is viewed as one bivariate observation. If it is a list, all components must be numerical vectors of equal length (coordinates of observations).

- **y**: The data as a matrix, data frame or list. If it is a matrix or data frame, then each row is viewed as one bivariate observation. If it is a list, all components must be numerical vectors of equal length (coordinates of observations).

- **factor**: An integer by default set to 2.

- **...**: For future use.

### Details

In dimension 2, inflates a convex polygon

### Value

Returns nothing.

### Author(s)

Nabil Mustafa, Saurabh Ray, and Mudassir Shabbir.

### References


### See Also

`convexhull`
Examples

```r
## calculation of centroid of a random pointset
z = matrix(rnorm(24),nc=2)
x = convexhull(z)
y= inflate(x)
```

rsdepth

RS Depth calculation

Description

Computes the Ray Shooting depth of a point with respect to a bivariate data set.

Usage

```r
rsdepth(pt,q, ...)
```

Arguments

- `q`: Numerical vector whose depth is to be calculated. Data needs to be 2-dimensional.
- `pt`: The data as a matrix, data frame or list. If it is a matrix or data frame, then each row is viewed as one bivariate observation. If it is a list, all components must be numerical vectors of equal length (coordinates of observations).
- `...`: For future use.

Details

In dimension 2, calculates ray shooting depth of a given point with respect to the point set. Time complexity of the simple algorithms implemented is $O(n \log n)$.

Value

Returns the exact depth of bivariate point $q$ with respect to data set $pt$, the number of line segments interested by a ray from $q$, minimum over all rays.

Author(s)

Nabil Mustafa, Saurabh Ray, and Mudassir Shabbir.

References


See Also

rsmed
Examples

```r
## calculation of RS depth
z = matrix(rnorm(24), nc=2)
x = matrix(rnorm(2), nc=2)
rsdepth(z, x)
```

### Description
Computes the Ray Shooting median of a bivariate data set.

### Usage

```r
rsmed(pt, eps=c(0), ...)
```

### Arguments

- `pt` The data as a matrix, data frame or list. If it is a matrix or data frame, then each row is viewed as one multivariate observation. If it is a list, all components must be numerical vectors of equal length (coordinates of observations). Only 2-dimensional data is acceptable in this version.
- `eps` eps is an optional parameter used for approximating a median in case of large data sets. It takes value of a real between 0 and 1 and is by default set to 0 which means no approximation is used if eps is not given.
- `...` Reserved for future use.

### Details
Finds out the an arbitrary point among the Ray Shooting median set of given point set. Current uses the brute-force algorithm on all $O(n^4)$ possible points in the arrangement of all possible lines in complete graph on pt. For each point $O(n \log n)$ is used to find out depth so overall complexity of this algorithm is $O(n^5 \log n)$. When approximation parameter is provided then algorithm tries to approximate by finding median of a uniform sample subset of pt of size $1/\text{eps}^2 \log(1/\text{eps})$. If this constant is more than the size of original set then eps value is ignored and exact median is calculate on original point set.

### Value
A point in two dimension is returned as a single row two column vector

### Author(s)
Nabil Mustafa, Saurabh Ray, and Mudassir Shabbir.
rsplot

References


See Also

rdepth for depth function

Examples

## RS median of a two-dimensional data set

```r
set.seed(617)
z <- matrix(rnorm(120), nc = 2)
rsmed(z, eps = 0.2)
```

rsplot

Ray Shooting depth Bag

Description

Computes the Ray Shooting depth ISO of a point with respect to a bivariate data set.

Usage

```r
rsplot(x, y = NULL, factorsecondbag = 2, mring = T, ...)
```

Arguments

- **x**: The data as a matrix, data frame or list. If it is a matrix or data frame, then each row is viewed as one bivariate observation. If it is a list, all components must be numerical vectors of equal length (coordinates of observations).
- **y**: The data as a matrix, data frame or list. If it is a matrix or data frame, then each row is viewed as one bivariate observation. If it is a list, all components must be numerical vectors of equal length (coordinates of observations).
- **factorsecondbag**: Factor for second bag set to 2 by default.
- **mring**: Boolean value set to TRUE by default.
- **...**: For future use.

Details

In dimension 2, calculates ray shooting depth of a given point with respect to the point set. Time complexity of the simple algorithms implemented is $O(n \log n)$. ISO

Value

Returns with respect to data set pt, the number of line segments interested by a ray from , minimum over all rays. ISO
### rsrings

**Bivariate RS Rings**

Computes the Ray Shooting rings of a bivariate data set.

#### Usage

```r
rsrings(pt, numofrings=c(5), clr=FALSE,...)
```

#### Arguments

- **pt**: The data as a matrix, data frame or list. If it is a matrix or data frame, then each row is viewed as one multivariate observation. If it is a list, all components must be numerical vectors of equal length (coordinates of observations). Only 2-dimensional data is acceptable in this version.
- **numofrings**: Total number of rings expected.
- **clr**: Boolean for whether colors are used or not.
- **...**: Reserved for future use.

#### Details

Finds out the an arbitrary point among the Ray Shooting median set of given point set. Current uses the brute-force algorithm on all $O(n^4)$ possible points in the arrangement of all possible lines in complete graph on `pt`. For each point $O(n \log n)$ is used to find out depth so overall complexity of this algorithm is $O(n^5 \log n)$. When approximation parameter is provided then algorithm tries to approximate by finding median of a uniform sample subset of `pt` of size $1/\varepsilon^2 \log(1/\varepsilon)$. If this constant is more than the size of original set then `eps` value is ignored and exact median is calculate on original point set.
**Value**

Number of rings returned

**Author(s)**

Nabil Mustafa, Saurabh Ray, and Mudassir Shabbir.

**References**


**See Also**

`rsdepth` for depth function

**Examples**

```r
## RS median of a two-dimensional data set
set.seed(617)
zz <- matrix(rnorm(60), nc = 2)
rsrings(zz)
```

---

**rstinteval**

*Bivariate RS Rings*

**Description**

Computes the Ray Shooting rings of a bivariate data set.

**Usage**

`rstinteval(pt, beta=c(0.90), sampleSize=c(250), M=c(50), clr=FALSE, ...)`

**Arguments**

- **pt**
  
  The data as a matrix, data frame or list. If it is a matrix or data frame, then each row is viewed as one multivariate observation. If it is a list, all components must be numerical vectors of equal length (coordinates of observations). Only 2-dimensional data is acceptable in this version.

- **beta**
  
  beta is a parameter between 0 and 1 determines the accuracy of the interval. Set to 0.90 by default.

- **sampleSize**
  
  Size of the sample data set.

- **M**
  
  Size of test data set.

- **clr**
  
  Clear the canvas before use or not. Boolean and set to FALSE by default.

- **...**
  
  Reserved for future use.
Details
This function creates a two dimension generalization of confidence intervals of data. A bag that contains beta fraction of data points is constructed.

Value
Should not return anything

Author(s)
Nabil Mustafa, Saurabh Ray, and Mudassir Shabbir.

References

See Also
rsdepth for depth function

Examples
```r
## RS median of a two-dimensional data set
set.seed(617)
zz <- matrix(rnorm(600), nc = 2)
rstinterval(zz)
```
Index

* Centroid of a convex polygon
  centroid, 2
* Convex Hull of a pointset
  convexhull, 3
* Draws Complete Graph of a pointset
  drawcompletegraph, 4
* RS Bag
  getbag, 5
  rsplot, 9
* RS depth
  rsdepth, 7
* bivariateCentroid
  centroid, 2
* bivariateConvexhull
  convexhull, 3
drawcompletegraph, 4
  inflate, 6
* bivariateISO
  getbag, 5
  rsplot, 9
* bivariate
  rsdepth, 7
  rsmed, 8
  rsrings, 10
  rstinterval, 11
* inflates a convex polygon
  inflate, 6
* multivariate
  rsdepth, 7
  rsmed, 8
  rsrings, 10
  rstinterval, 11
* nonparametric
  rsdepth, 7
  rsmed, 8
  rsrings, 10
  rstinterval, 11
* robust
  rsdepth, 7