Package 'forestSAS'

May 28, 2026

Type Package

Title Forest Spatial Structure Analysis Systems

Version 2.0.5

Depends R (>= 3.5.0)

Imports ggimage,reshape2,utils,spatstat.geom, spatstat.random

Suggests ggplot2, shiny,spatstat,spatstat.data

Description Recent years have seen significant interest in neighborhood-based structural parameters that effectively represent the spatial characteristics of tree populations and forest communities, and possess strong applicability for guiding forestry practices. This package provides valuable information that enhances our understanding and analysis of the fine-scale spatial structure of tree populations and forest stands. Reference: Yan L, Tan W, Chai Z, et al (2019) <doi:10.13323/j.cnki.j.fafu(nat.sci.).2019.03.007>.

License GPL-2

LazyData TRUE

Encoding UTF-8

NeedsCompilation no

Author Zongzheng Chai [aut, cre] (ORCID: <https://orcid.org/0000-0002-0530-0040>)

Maintainer Zongzheng Chai <chaizz@126.com>

Repository CRAN

Date/Publication 2026-05-28 06:20:02 UTC

Contents

forestSAS-package 2
addmark.ppp 4
buffer 5
crowding 6
differ 7
dominance 8
expandedge 9

forestSAS_simapp . . . 11
fsasN4 . . . 11
ggrosechart . . . 12
ideal . . . 14
list_to_matrix . . . 15
mingling . . . 15
nnangle . . . 16
nnid . . . 17
nnIndex . . . 18
nnoverlap . . . 21
opt_spastr . . . 22
pv . . . 23
rebuild.ppp . . . 25
shrinkedge . . . 27
simtreecom . . . 28
spastr . . . 29
storeydvd . . . 30
tree.ppp . . . 31
treecom_example . . . 32
treedata . . . 33
uniform.angle . . . 34

Index **36**

`forestSAS-package` *Forest Spatial Structure Analysis Systems*

Description

Recent years have seen significant interest in neighborhood-based structural parameters that effectively represent the spatial characteristics of tree populations and forest communities, and possess strong applicability for guiding forestry practices. This package provides valuable information that enhances our understanding and analysis of the fine-scale spatial structure of tree populations and forest stands. Reference: Yan L, Tan W, Chai Z, et al (2019) <doi:10.13323/j.cnki.j.fafu(nat.sci.).2019.03.007>.

Details

Forest structure commonly refers to a distribution pattern of tree attributes within a forest ecosystem. Similarly, tree population structure describes the distribution characteristics of individuals of conspecifics within a community, and the spatial structure of a tree population is largely determined by the relationships among neighboring groups of trees.

Author(s)

Zongzheng Chai, chaizz@126.com

References

None

Examples

```
library(spatstat)
data(tree.ppp)
##Get the tree attributies of nearest neighbour
nnindices<-nnIndex(tree.ppp,N=4,
            smark=c("sp.code","dbh.cm","storey",
               "crownwid.m","group","biomass.kg",
               "quality","x","y"),buffer=FALSE)

#Species mingling
M<-fsasN4(nnindices$nnsp.code,match.fun=mingling)
M
#Stand storey differation degree
H<-fsasN4(nnindices$nnstorey,match.fun=differ)
H
#Tree successional degree
S<-fsasN4(nnindices$nngroup,match.fun=ideal,para="Climax")
S
#Tree quality ideal state
Q<-fsasN4(nnindices$nnquality,match.fun=ideal,
          para=c("Excellent","Good"))
Q
#Tree corwding degree
C<-fsasN4(nnoverlap(nnindices$nncrownwid.m,
         nnindices$nndist),match.fun=crowding)
C
#Tree uniform angle index
W<-fsasN4(nnangle(nnindices$nndist,nnindices$nnx,
           nnindices$nny)$nnangle,
          match.fun=uniform.angle,para=72)
W
#Tree dominance
U<-fsasN4(nnindices$nndbh.cm,match.fun=dominance)
U
#Tree biomass dominance
B<-fsasN4(nnindices$nnbiomass.kg,match.fun=dominance)
B

##Compute the structrue heterogeneity of index
M.pv<-pv(M$result$index,optm=1)
M.pv
H.pv<-pv(H$result$index,optm=1)
H.pv
S.pv<-pv(S$result$index,optm=1)
S.pv
Q.pv<-pv(Q$result$index,optm=1)
Q.pv
C.pv<-pv(C$result$index,optm=0.5)
C.pv
W.pv<-pv(W$result$index,optm=0.5)
W.pv
U.pv<-pv(U$result$index,optm=0)
```

```
U.pv
B.pv<-pv(B$result$index,optm=0)
B.pv

##Compute total forest saptial structrue heterogeneity
#based on the average of indices preference value
IAVE<-pv(index=c(M$meanI,H$meanI,S$meanI,Q$meanI,
                      C$meanI,W$meanI,U$meanI,B$meanI),
              optm=c(1,1,1,1,0.5,0.5,0,0))
IAVE

##Compute total forest saptial structrue heterogeneity
#based on the preference value of indices average
IPVE=mean(M.pv,H.pv,S.pv,Q.pv,C.pv,W.pv,U.pv,B.pv)
IPVE
```

---

addmark.ppp *Add marks for a point pattern*

Description

Add marks for a point pattern

Usage

```
addmark.ppp(X, add.mark, add.name = "storey")
```

Arguments

X	A point pattern (object of class "ppp").
add.mark	Marks need added in the point pattern
add.name	Names of added marks (add.mark)

Value

A point pattern added marks

Author(s)

Zongzheng Chai, chaizz@126.com

References

None

Examples

```
library(spatstat.data)
data(finpines)
####Dividing the stories
finpines.storey<-storeydvd(finpines$marks$height,storeynum=6)
finpines.storey

####Computing the storey differation
##Add the storey mark for finepines
finpines.addstorey<-addmark.ppp(finpines,
                                add.mark=finpines.storey$heightdata[,2:3],
                                add.name=c("interval","storey" ))
finpines.addstorey
```

`buffer` *Identify the buffer zone in the point pattern*

Description

To eliminate edge effects and improve the accuracy of the Calculation,such as analyzing the forest spatial structure parameters, we always established a buffer zone around the plot. In the statistical analysis, only the trees in the reduced window are used as reference trees, and the individual trees in the buffer zone are only considered to be the nearest neighbors of the trees in the reduced window. This edge correction can individually evaluate each tree to determine whether all n nearest neighbors are truly located within the plot.

Usage

```
buffer(X, buf.xwid = 5, buf.ywid = 5)
```

Arguments

`X`	A point pattern (object of class "ppp").
`buf.xwid`	The width of buffer zone in the x coordinates of data points
`buf.ywid`	The width of buffer zone in the y coordinates of data points

Value

An object of class "ppp", which add a marks "zone" in the orginal point pattern

Author(s)

Zongzheng Chai, chaizz@126.com

References

Chai ZZ, Sun CL, Wang DX, Liu WZ, and Zhang CS.2016. Spatial structure and dynamics of predominant populations in a virgin old-growthoak forest in the Qinling Mountains, China. Scandinavian Journal of Forest Research. DOI: 10.1080/02827581.2016.1183703

Examples

```
library(spatstat.geom)
####Based on the simulated data####
# Creating a simulated point pattern
x <- runif(20)
y <- runif(20)
X <- ppp(x, y, c(0,1), c(0,1))
X
# Identifing the buffer zone and core zone in the point pattern
# The width of buffer zone in the x coordinates of data points is 0.2
# The width of buffer zone in the y coordinates of data points is 0.3
X_buf<-buffer(X,buf.xwid =0.2,buf.ywid=0.3)
X_buf
as.data.frame(X_buf)

####Based on the example data####
library(spatstat.data)
data(finpines)
finpines
# Marked planar point pattern: 126 points
# Mark variables: diameter, height
# window: rectangle = [-5, 5] x [-8, 2] metres

# Identifing the buffer zone and core zone in the finpines
# The width of buffer zone in the x coordinates of finpines is 2
# The width of buffer zone in the y coordinates of finpines is 3
finpines_buf<-buffer(finpines,buf.xwid =2,buf.ywid=3)
finpines_buf
as.data.frame(finpines_buf)
```

`crowding` *A forest spatial structure index characterizing crowding degree.*

Description

Analyze the crowding degree of a neighborhood unit according to the overlapping of the crown in spatial micro-environment which clearly define the crowding degree for a reference tree and its four nearest neighbors, and to some degree of the competition pressure on it as well.

Usage

```
crowding(x)
```

Arguments

`x`	A vector composed by 5 crowding degree value

Value

Result will retrun five values,0,0.25,0.5,0.75,1,which means none, one, two, three, or four overlap of nearest neighbors with reference tree,respectively.

Author(s)

Zongzheng Chai, chaizz@126.com

References

Hu YB, Hui GY. 2015. How to describe the crowding degree of trees based on the relationship of neighboring trees. Journal of Beijing Forestry University. 37(9):1-8.

Examples

```
overlap1<-c(-0.5,-0.4,-0.3,-0.2) ##negative means no overlap
overlap2<-c(-0.5,-0.4,-0.3,0.2)  ##positive means overlap
overlap3<-c(-0.5,-0.4,0.3,0.2)
overlap4<-c(-0.5,0.4,0.3,0.2)
overlap5<-c(0.5,0.4,0.3,0.2)

crowding1<-crowding(overlap1)
crowding1
crowding2<-crowding(overlap2)
crowding2
crowding3<-crowding(overlap3)
crowding3
crowding4<-crowding(overlap4)
crowding4
crowding5<-crowding(overlap5)
crowding5
```

---

differ *A forest spatial structure index characterizing differation degree of tree attributies.*

Description

Analyze the differation degree of tree attributies in a neighborhood unit according to the characteristics of the tree attributies,such as forest storey,speices,etc.in spatial micro-environment.which clearly define the differation degree of tree attributies for a reference tree and its four nearest neighbors.

Usage

```
differ(x)
```

Arguments

x	A vector composed by 5 tree attributies.

Value

Result will retrun five values,0,0.25,0.5,0.75,1,which means one, two, three, four or five different attributies of nearest neighbors for reference tree,respectively.

Author(s)

Zongzheng Chai, chaizz@126.com

References

None

Examples

```
##S1,S2,S3,S4,S5,represent 5 different forest stoeries
storey1<-c("S1","S1","S1","S1","S1")
storey2<-c("S1","S1","S1","S1","S2")
storey3<-c("S1","S1","S1","S3","S2")
storey4<-c("S1","S1","S4","S3","S2")
storey5<-c("S1","S5","S4","S3","S2")

differ1<-differ(storey1)
differ1
differ2<-differ(storey2)
differ2
differ3<-differ(storey3)
differ3
differ4<-differ(storey4)
differ4
differ5<-differ(storey5)
differ5
```

---

dominance *A forest spatial structure index characterizing tree dominance*

Description

The tree attribute dominance was proposed by Hui et al. (1998) to relate the relative dominance of a given tree to its species or silvicultural significance. It is defined as the proportion of the n nearest neighbours of a given reference tree which are bigger than the reference tree.

Usage

```
dominance(x)
```

Arguments

x	A vector composed by 5 tree attributes, such as dbh,biomass.

Value

Result will retrun five values,0,0.25,0.5,0.75,1,which means the propertion of the larger tree attributes in the 4 nearest neighbours than the reference tree.

Author(s)

Zongzheng Chai, chaizz@126.com

References

Gadow Kv, and Hui GY. 2002. Characterizing forest spatial structure and diversity. Proc of the SUFOR international workshop: Sustainable forestry in temperate regions. p 20-30.

Examples

```
dbh1<-c(5,4,4,4,4)
dbh2<-c(5,4,4,4,6)
dbh3<-c(5,4,4,7,6)
dbh4<-c(5,4,8,7,6)
dbh5<-c(5,9,8,7,6)

dominance1<-dominance(dbh1)
dominance1
dominance2<-dominance(dbh2)
dominance2
dominance3<-dominance(dbh3)
dominance3
dominance4<-dominance(dbh4)
dominance4
dominance5<-dominance(dbh5)
dominance5
```

---

expandedge *Expand the edges for the point pattern with replication.*

Description

When computing forest spatial structure indices, The trees near the edges of the study region are distorted because the outside is empty. Common solutions to this problem are not to use indices computed for trees near the edges, or (with rectangular regions) to attach translated copies, thus changing the topology into a torus.

Usage

```
expandedge(X, xwidth, ywidth,id = 1:X$n, marks=X$marks,type="com")
```

Arguments

X	A point pattern (object of class "ppp").
xwidth	Distance from the edges x coordinates,note xwidth is the half of your width wanted, because the width will expand both edges in the x coordinates.
ywidth	Distance from the edges y coordinates,note xwidth is the half of your width wanted, because the width will expand both edges in the y coordinates.
id	Specific identification number of the points in the point pattern.
marks	Marks of nearest neighbour in the point pattern.
type	Data format,com is data.frame and ppp is the ppp format in the spapstat package.

Details

When expanding for the full pattern, and using expandedge() with a positive xwidth and ywidth; conversely, shrinking for the full pattern, and using shrinkedge () with a negative xwidth and ywidth.

Value

Result returns original data (not a a point pattern with the same structure as trees), id is new id of points after edges expanded and old.id is orginal id of points. The pattern is first expanded by surrounding it with 8 shifted copies (the window must be rectangular). Then, the parts of the pattern that are at a distance less than xwidth or ywidth from an edge of the enlarged pattern are discarded. If xwidth or ywidth = 0, trees are returned unchanged.

Author(s)

Zongzheng Chai, chaizz@126.com

Examples

```
library(spatstat)
library(spatstat.data)
data(finpines)
finpines$window
# window: rectangle = [-5, 5] x [-8, 2] metres

# Expand the rectangle [-5,5]x[-8,2] to [-6,6]x[-9,3]
Expand.trees<- expandedge(finpines,xwidth=2,ywidth=1,id=1:126)
Expand.trees

# Show the changes by figures
opar<-par(mfrow=c(1,2))
plot(finpines$x,finpines$y)
text(finpines$x,finpines$y,1:126)
plot(Expand.trees$x,Expand.trees$y)
text(Expand.trees$x,Expand.trees$y,Expand.trees$old.id)
rect(-5,-8,5,2,border="red")
par(opar)
```

forestSAS_simapp *Shiny for forest spatial structure analysis based on simulated data*

Description

Shiny for forest spatial structure analysis based on simulated data.

Usage

```
forestSAS_simapp()
```

Details

Shiny for forest spatial structure analysis based on simulated data and visualizing rose charts using ggplot2.

Note

This function requires the **shiny** package, which is listed as a suggested dependency.

fsasN4 *A body function to analyze the forest saptial structure.*

Description

The function provided a easy flexible funciton body to analyze the forest saptial structure.

Usage

```
fsasN4(nnattri, match.fun, para = NULL)
```

Arguments

nnattri	attributies of nearest neighbour
match.fun	funcitons,such as "differ","dominance","ideal","uniform.angle","crowding","mingling",etc.
para	the parameter in the above match.fun

Value

result is the "nnmark" and,colum I is the sptial structre indices value Icount is the count distribution of I Ifreq is the frequency distribution of I meanI is the average value of I data is the final point pattern

Author(s)

Zongzheng Chai, chaizz@126.com

References

Gadow Kv, and Hui G. 2002. Characterizing forest spatial structure and diversity. Proc of the SUFOR international workshop: Sustainable forestry in temperate regions. p 20-30.

Examples

```
data(tree.ppp)
##Get the tree attributies of nearest neighbour for calculation
nnindices<-nnIndex(tree.ppp,N=4,
                   smark=c("sp.code","dbh.cm","storey","crownwid.m","group",
                           "biomass.kg","quality","x","y"),buffer=FALSE)

#Species mingling
M<-fsasN4(nnindices$nnsp.code,match.fun=mingling)
M
#Stand storey differation degree
H<-fsasN4(nnindices$nnstorey,match.fun=differ)
H
#Tree successional degree
S<-fsasN4(nnindices$nngroup,match.fun=ideal,para="Climax")
S
#Tree quality ideal state
Q<-fsasN4(nnindices$nnquality,match.fun=ideal,para=c("Excellent","Good"))
Q
#Tree corwding degree
C<-fsasN4(nnoverlap(nnindices$nncrownwid.m,nnindices$nndist),match.fun=crowding)
C
#Tree uniform angle index
W<-fsasN4(nnangle(nnindices$nndist,nnindices$nnx,nnindices$nny)$nnangle,
          match.fun=uniform.angle,para=72)
W
#Tree dominance
U<-fsasN4(nnindices$nndbh.cm,match.fun=dominance)
U
#Tree biomass dominance
B<-fsasN4(nnindices$nnbiomass.kg,match.fun=dominance)
B
```

`ggrosechart` *Draw a rose chart using the ggplot2 package.*

Description

Draw a rose chart using the ggplot2 package.

Usage

```
ggrosechart(value,index)
```

Arguments

value	Numeric vector, Numeric values for each sector of the rose chart (recommended range: 0 to 1). Length must match the number of categories.
index	character/factor vector,Category labels for each sector. Length must be identical to value.

Details

Calculates the number of categories and generates equal angles for polar coordinates.Constructs a data frame with angles, values, and category labels for ggplot2 visualization. Plots a circular bar chart and transforms it into a rose chart using coord_polar().Applies fixed axis limits (y-axis: 0-1), customized axis text, and clean theme settings for scientific publishing.

Value

Returns a ggplot object that can be printed, modified, or saved directly.

Author(s)

Zongzheng Chai, chaizz@126.com

Examples

```
library(spatstat)
data(treecom_example)
head(treecom_example)
treecom_spastr<-spastr(X_df=treecom_example,
              xrange=c(0,100),yrange=c(0,100),
              xwidth=10,ywidth=10,
              buf.xwid =10, buf.ywid = 10,
              smark=c("spe","storey","dbh","cw","x","y"))
treecom_spastr

library(ggplot2)
p1<-ggrosechart(value=treecom_spastr$Index_pv,
                index=c("M.pv","H.pv","W.pv","C.pv","U.pv"))+
  annotate("text", x=-Inf, y=Inf,color="red", parse=TRUE,
           hjust=-0.8, vjust=-0.5,label=paste("italic(omega)==",0.465),
           size=6)
print(p1)
####0.465 is the value of treecom_spastr$Omega

p2<-ggrosechart(value=c(1,1,1,1,1),
                index=c("M.pv","H.pv","W.pv","C.pv","U.pv"))+
  annotate("text", x=-Inf, y=Inf,color="red", parse=TRUE,
           hjust=-2, vjust=-0.5,label=paste("italic(omega)==",1.000),
           size=6)
print(p2)
```

ideal *A forest spatial structure index characterizing distribution of ideal state for tree attributies.*

Description

Analyze the distribution of ideal state for tree attributies in a neighborhood unit according to the characteristics of the tree attributies,such as tree health,climax species distribution,etc.in spatial micro-environment.which clearly define the ideal degree of tree attributies for a reference tree and its four nearest neighbors.

Usage

```
ideal(x, para)
```

Arguments

x	A vector composed by 5 tree attributies.
para	Ideal state of tree attributies.

Value

Result will retrun five values,0,0.25,0.5,0.75,1,which means none,one, two, three or four ideal tree attributies of nearest neighbors for reference tree,respectively.

Author(s)

Zongzheng Chai, chaizz@126.com

References

None

Examples

```
health1<-c("poor","poor","poor","poor")
health2<-c("poor","poor","poor","excellent")
health3<-c("poor","poor","good","excellent")
health4<-c("poor","excellent","good","excellent")
health5<-c("good","excellent","good","excellent")

ideal1<-ideal(health1,para=c("good","excellent"))
ideal1
ideal2<-ideal(health2,para=c("good","excellent"))
ideal2
ideal3<-ideal(health3,para=c("good","excellent"))
ideal3
ideal4<-ideal(health4,para=c("good","excellent"))
ideal4
```

```
ideal5<-ideal(health5,para=c("good","excellent"))
ideal5
```

`list_to_matrix` *Converting list to matrix*

Description

Converting the data with class list to the data with class matrix

Usage

```
list_to_matrix(x, item = NA)
```

Arguments

`x`	Data with class list
`item`	After converting list to matrix, using NA to pad out the blanks within matrix

Value

Data with class matrix

Author(s)

Zongzheng Chai,chaizz@126.com

Examples

```
# Creating a list data "datalist", and covert "datalist" to a matrix data "datamat"
datalist<-list(dataA=1:5,dataB=1:10,dataC=1:15,dataD=1:20)
datamat<-list_to_matrix(datalist)
datamat
```

`mingling` *A forest spatial structure index characterizing tree diversity*

Description

The tree attribute mingling describes the species variety in the vicinity of a given reference tree and has been defined as the proportion of the n nearest neighbours that do not belong to the same species

Usage

```
mingling(x)
```

Arguments

x	A vector composed by 5 tree species

Value

Result will retrun five values,0,0.25,0.5,0.75,1,which means the propertion of the same tree species with refercence tree in the 4 nearest neighbours

Author(s)

Zongzheng Chai, chaizz@126.com

References

Hui GY, Zhao X, Zhao Z, Gadow Kv. 2011. Evaluating tree species spatial diversity based on neighborhood relationships. Forest Sci 57:292-300

Examples

```
spname1<-c("sp1","sp1","sp1","sp1","sp1")
spname2<-c("sp1","sp1","sp1","sp1","sp2")
spname3<-c("sp1","sp1","sp1","sp3","sp2")
spname4<-c("sp1","sp1","sp4","sp3","sp2")
spname5<-c("sp1","sp5","sp4","sp3","sp2")

mingling1<-mingling(spname1)
mingling1
mingling2<-mingling(spname2)
mingling2
mingling3<-mingling(spname3)
mingling3
mingling4<-mingling(spname4)
mingling4
mingling5<-mingling(spname5)
mingling5
```

---

nnangle *Identify the angle among nearest neighbours*

Description

Describes the degree of regularity in the spatial distribution of n trees that are nearest to a reference tree. Moving clockwise around reference tree,the angle was obtained between two adjacent neighbours.

Usage

```
nnangle(nndist, nnx, nny)
```

Arguments

nndist	Distance of nearest neighbour
nnx	x coordinate of nearest neighbour
nny	y coordinate of nearest neighbour

Value

The angles obtained between two adjacent neighbours.

Author(s)

Zongzheng Chai, chaizz@126.com

References

None

Examples

```
data(tree.ppp)
NNcoord<-nnIndex(tree.ppp,N=4,smark=c("x","y"),buffer=FALSE)
NNangle<-nnangle(NNcoord$nndist,NNcoord$nnx,NNcoord$nny)
NNangle
```

---

nnid *Find the id of nearest neighbour for each point.*

Description

Find the id of nearest neighbour for each point in a point pattern by specific numbers or circular neighborhood.

Usage

```
nnid(X, N = NULL, R = NULL, id, exclude = TRUE)
```

Arguments

X	A marked point pattern (object of class "ppp").
N	Specific number of points in the neighborhood, the value is always assinged 4.
R	Specific circular neighborhood, the value is always assinged 5.
id	Specific identification number of each row in the point pattern.
exclude	Wether including id of core points, if exclude is TRUE the result return the data removed the id of core points, if exclude is FALSE the result return the data with the id of core points.

Value

id of nearest neighbour for each point

Author(s)

Zongzheng Chai, chaizz@126.com

Examples

```
library(spatstat)
library(spatstat.data)
data(finpines)
#Find the id of the nearest neighbour in a certain numbers and without id
finpines.rmidN4<-nnid(finpines,id=paste("T",1:finpines$n),N=4)
finpines.rmidN4

#Find the id of the nearest neighbour in a certain numbers with id
finpines.idN4<-nnid(finpines,id=paste("T",1:finpines$n),N=4,exclude=FALSE)
finpines.idN4

#Find the id of the nearest neighbour within a certain radius without id
data(tree.ppp)
finpines.rmidR0.5<-nnid(tree.ppp,id=paste("T",1:41),R=0.5)
finpines.rmidR0.5

#Find the id of the nearest neighbour within a certain radius with id
finpines.idR0.5<-nnid(tree.ppp,id=paste("T",1:41),R=0.5,exclude=FALSE)
finpines.idR0.5
```

---

nnIndex *Get the marks of nearest neighbour in the point pattern*

Description

Get the marks of nearest neighbour in the point pattern

Usage

```
nnIndex(X, id = 1:(X$n), smark = NULL, N = NULL,
        R = NULL, rm.id = NULL, add.X = NULL,
        add.id = paste("add", 1:(add.X$n), sep = ""),
        buffer = FALSE, buf.xwid = 5, buf.ywid = 5,
        exclusion = FALSE)
```

Arguments

X	A marked point pattern (object of class "ppp").
id	Specific identification number of each row in the point pattern.
smark	Selected marks to find the marks of nearest neighbour.
N	Specific number of points in the neighborhood, the value is always assinged 4.
R	Specific circular neighborhood, the value is always assinged 5.
rm.id	Needed removed the specific rows in the point pattern, refer to the id.
add.X	Another point pattern need added in the point pattern.
add.id	Specific identification number of each row in the added point pattern.
buffer	if buffer is TRUE,show the all of data with buffer and core zone. if buffer is FALSE,only show the data in the core zone.
buf.xwid	The width of buffer zone in the x coordinates of data points.
buf.ywid	The width of buffer zone in the y coordinates of data points.
exclusion	Wether including the data of the buffer zone, if exclude is TRUE the result return the data remove the data of the buffer zone, and only the data of the core zone;if exclude is FALSE the result return the data with buffer and core zone.

Details

Given a marked point pattern dataset X this function computes, for each desired location y, the mark attached to the point of X that is nearest to y. The desired locations y can be either a pixel grid or the point pattern X itself.see also the function "nnmark" in the package spatstat

Value

Multiple "nnmarks" are the selected marks (smark) of nearest neighbour nnid is the id of nearest neighbour nndist is the distance of nearest neighbour data is the final point pattern

Author(s)

Zongzheng Chai, chaizz@126.com

See Also

The function "nnmark" in the package spatstat

Examples

```
library(spatstat)
library(spatstat.data)
library(spatstat.geom)
data(finpines)
#### Based on specific number (N=4) of nearest neighbour####
## Basic usage
nndN<-nnIndex(finpines,id=paste("T",1:126,sep=""),N=1,
              smark="diameter",buffer=TRUE,buf.xwid =2,buf.ywid=3)
```

```
nndN
nndhN<-nnIndex(finpines,id=paste("T",1:126,sep=""),N=4,
       smark=c("diameter","height"),buffer=TRUE,buf.xwid =2
       ,buf.ywid=3)
nndhN

# Only the points in the core zone (Removed the points in the buffer zone)
nndN_core<-nnIndex(finpines,id=paste("T",1:126,sep=""),N=4,
              smark="diameter",buffer=FALSE,buf.xwid =2,buf.ywid=3)
nndN_core
nndhN_core<-nnIndex(finpines,id=paste("T",1:126,sep=""),N=4,
         smark=c("diameter","height"),buffer=FALSE,buf.xwid =2,
         buf.ywid=3)
nndhN_core

## Remove some points, and then identify the marks of Nearest Neighbour
nndN.rm_core<-nnIndex(finpines,id=paste("T",1:126,sep=""),
                    rm.id=c("T1","T3","T8","T9","T59","T60","T120"),
            N=4,smark="diameter",buffer=FALSE,buf.xwid =2,buf.ywid=3)
nndN.rm_core

## add some points, and then identify the marks  of Nearest Neighbour
add.x=c(-2,-1,0,2,4)
add.y=c(-4,-3,-6,0,1)
add.marks=data.frame(diameter=c(2.0,3.0,4.0,5.0,6.0),
                     height=c(2.5,3.5,4.5,5.5,6.5))
add.Xdata=ppp(x=add.x,y=add.y,marks=add.marks,c(-5,5),c(-8,2))

nndN.add_core<-nnIndex(finpines,id=paste("T",1:126,sep=""),
                       add.X=add.Xdata,
                       add.id=paste("NT",1:5,sep=""),
          N=4,smark="diameter",buffer=FALSE,buf.xwid =2,buf.ywid=3)
nndN.add_core

#### Based on specific circular (R=5) of nearest neighbour####
## Basic usage
nndR<-nnIndex(finpines,id=paste("T",1:126,sep=""),R=0.5,
              smark="diameter",buffer=TRUE,buf.xwid =2,buf.ywid=3)
nndR
nndhR<-nnIndex(finpines,id=paste("T",1:126,sep=""),R=0.5,
              smark=c("diameter","height"),buffer=TRUE,
               buf.xwid =2,buf.ywid=3)
nndhR

# Only the points in the core zone (Removed the points in the buffer zone)
nndR_core<-nnIndex(finpines,id=paste("T",1:126,sep=""),R=0.5,
            smark="diameter",buffer=FALSE,buf.xwid =2,buf.ywid=3)
nndR_core
nndhR_core<-nnIndex(finpines,id=paste("T",1:126,sep=""),R=0.5,
            smark=c("diameter","height"),buffer=FALSE,
                    buf.xwid =2,buf.ywid=3)
nndhR_core
```

```
## Remove some points, and then identify the marks  of Nearest Neighbour
nndR.rm_core<-nnIndex(finpines,id=paste("T",1:126,sep=""),
                  rm.id=c("T1","T3","T8","T9","T59","T60","T120"),
          R=0.5,smark="diameter",buffer=FALSE,buf.xwid =2,buf.ywid=3)
nndR.rm_core

## add some points, and then identify the marks  of Nearest Neighbour
nndR.add_core<-nnIndex(finpines,id=paste("T",1:126,sep=""),
                       add.X=add.Xdata,
                       add.id=paste("NT",1:5,sep=""),
          R=0.5,smark="diameter",buffer=FALSE,buf.xwid =2,buf.ywid=3)
nndR.add_core
```

---

nnoverlap *Crown overlap among nearest neighbour*

Description

Identify wheter crown overlap among nearest neighbour.

Usage

```
nnoverlap(nncrown, nndist)
```

Arguments

nncrown	Crown width of nearest neighbour.
nndist	Distance of nearest neighbour.

Value

Crown overlap among nearest neighbour,the value is positive means overlap, value is negative means no overlap.

Author(s)

Zongzheng Chai, chaizz@126.com

Examples

```
data(tree.ppp)
NNcrown<-nnIndex(tree.ppp,N=4,smark="crownwid.m",buffer=FALSE)
NNoverlap<-nnoverlap(NNcrown$nncrownwid.m,NNcrown$nndist)
NNoverlap
```

opt_spastr *Optimal forest spatial structure indices.*

Description

Optimal forest spatial structure indices.

Usage

```
opt_spastr(X_df,inter=100,pop=0.2,
                  smark=c("spe","storey","dbh","cw","x","y"),
                  xrange=c(0,100),yrange=c(0,100),
                  xwidth=5,ywidth=5,
                  buf.xwid = 5, buf.ywid = 5)
```

Arguments

X_df	Forest community data (object of class "data.frame")
inter	Inter quanitity.
pop	Cutting intensity.
smark	Selected marks to compute forest spatial structure indices.
buf.xwid	The width of buffer zone in the x coordinates of data points.
buf.ywid	The width of buffer zone in the y coordinates of data points.
xrange	Range of X axes
yrange	Range of Y axes
xwidth	Length of buffer zone of X axes
ywidth	Length of buffer zone of Y axes

Details

NULL

Value

NULL

Author(s)

Zongzheng Chai, chaizz@126.com

Examples

```
library(spatstat)
data(treecom_example)
head(treecom_example)
treecom_opt<-opt_spastr(X_df=treecom_example,inter=5,pop=0.1,
              xrange=c(0,100),yrange=c(0,100),
              xwidth=10,ywidth=10,
              buf.xwid =10, buf.ywid = 10,
              smark=c("spe","storey","dbh","cw","x","y"))
treecom_opt
```

---

pv *Preference value*

Description

A descriptive index to analyze the forest saptial structure indices

Usage

```
pv(index, optm)
```

Arguments

index	Actual value of forest spatial structure indices
optm	Optimal value of forest spatial structure indices

Value

Result of forest spatial structure heterogeneity assessment

Author(s)

Zongzheng Chai, chaizz@126.com

References

None

Examples

```
data(tree.ppp)
##Get the tree attributies of nearest neighbour
nnindices<-nnIndex(tree.ppp,N=4,
            smark=c("sp.code","dbh.cm","storey",
               "crownwid.m","group","biomass.kg",
               "quality","x","y"),buffer=FALSE)

#Species mingling
```

```
M<-fsasN4(nnindices$nnsp.code,match.fun=mingling)
M
#Stand storey differation degree
H<-fsasN4(nnindices$nnstorey,match.fun=differ)
H
#Tree successional degree
S<-fsasN4(nnindices$nngroup,match.fun=ideal,para="Climax")
S
#Tree quality ideal state
Q<-fsasN4(nnindices$nnquality,match.fun=ideal,
          para=c("Excellent","Good"))
Q
#Tree corwding degree
C<-fsasN4(nnoverlap(nnindices$nncrownwid.m,
         nnindices$nndist),match.fun=crowding)
C
#Tree uniform angle index
W<-fsasN4(nnangle(nnindices$nndist,nnindices$nnx,
           nnindices$nny)$nnangle,
          match.fun=uniform.angle,para=72)
W
#Tree dominance
U<-fsasN4(nnindices$nndbh.cm,match.fun=dominance)
U
#Tree biomass dominance
B<-fsasN4(nnindices$nnbiomass.kg,match.fun=dominance)
B

##Compute the structrue heterogeneity of index
M.pv<-pv(M$result$index,optm=1)
M.pv
H.pv<-pv(H$result$index,optm=1)
H.pv
S.pv<-pv(S$result$index,optm=1)
S.pv
Q.pv<-pv(Q$result$index,optm=1)
Q.pv
C.pv<-pv(C$result$index,optm=0.5)
C.pv
W.pv<-pv(W$result$index,optm=0.5)
W.pv
U.pv<-pv(U$result$index,optm=0)
U.pv
B.pv<-pv(B$result$index,optm=0)
B.pv

##Compute total forest saptial structrue heterogeneity
#based on the average of indices preference value
IAVE<-pv(index=c(M$meanI,H$meanI,S$meanI,Q$meanI,
                        C$meanI,W$meanI,U$meanI,B$meanI),
              optm=c(1,1,1,1,0.5,0.5,0,0))
IAVE
```

```
##Compute total forest saptial structrue heterogeneity
#based on the preference value of indices average
IPVE=mean(M.pv,H.pv,S.pv,Q.pv,C.pv,W.pv,U.pv,B.pv)
IPVE
```

---

rebuild.ppp *Reestablishing an object of class "ppp"*

Description

Reestablishing an object of class "ppp" by adding or removing some data from orginal point pattern (object of class "ppp")

Usage

```
rebuild.ppp(X, id = 1:(X$n), rm.id = NULL, add.X = NULL,
          add.id = paste("add", 1:(add.X$n), sep = ""))
```

Arguments

X	A point pattern (object of class "ppp").
id	Specific identification number of each row in the point pattern.
rm.id	Needed removed the specific rows in the point pattern, refer to the id.
add.X	Another point pattern need added in the point pattern.
add.id	Specific identification number of each row in the added point pattern.

Details

An object of class "ppp" describing a point pattern in the two-dimensional plane (see ppp.object), which was Reestablished by adding or removing some data from orginal point pattern

Value

A new point pattern after reestablished by adding or removing some data from orginal point pattern

Author(s)

Zongzheng Chai, chaizz@126.com

See Also

R function ppp in the package "spatstat"

Examples

```
library(spatstat.geom)
####Based on the simulated data####
# Creating a simulated point pattern
x <- runif(20)
y <- runif(20)
X <- ppp(x, y, c(0,1), c(0,1))
X

# Adding a identification number (ID) for each data point in the point pattern
# the following are equivalent
X1<-rebuild.ppp(X)
X1<-rebuild.ppp(X,id=1:X$n)
X1<-rebuild.ppp(X,id=1:20)
X1
as.data.frame(X1)

# Adding a identification number (ID) for each data point in the point pattern, and
# Removing the assinged points according to the ID.
X2<-rebuild.ppp(X,id=1:20,rm.id=1:5)
X2
as.data.frame(X2)
# Adding some points into orginal point pattern
newx <- runif(10)
newy <- runif(10)
newX <- ppp(newx, newy, c(0,1), c(0,1))
XY<-rebuild.ppp(X,add.X=newX,add.id=paste("new",1:10,sep=""))
XY
as.data.frame(XY)

####Based on the example data####
library(spatstat.data)
data(finpines)
finpines
# Marked planar point pattern: 126 points
# Mark variables: diameter, height
# window: rectangle = [-5, 5] x [-8, 2] metres

# Adding a identification number (ID) for each tree in the finpines
# finpines$n
# [1] 126
finpines_id<-rebuild.ppp(finpines,id=paste("T",1:126,sep=""))
finpines_id
head(as.data.frame(finpines_id))

# Removing the trees with id "T1","T3","T8","T9","T59","T60","T120"
finpines_rm<-rebuild.ppp(X=finpines,id=paste("T",1:126,sep=""),
                         rm.id=c("T1","T3","T8","T9","T59","T60","T120"))
finpines_rm
as.data.frame(finpines_rm)

# Adding some trees data into finpines
```

```
add.x=c(-2,-1,0,2,4)
add.y=c(-4,-3,-6,0,1)
add.marks=data.frame(diameter=c(2.0,3.0,4.0,5.0,6.0),
                height=c(2.5,3.5,4.5,5.5,6.5))
add.Xdata=ppp(x=add.x,y=add.y,marks=add.marks,c(-5,5),c(-8,2))

finpines_add<-rebuild.ppp(X=finpines,id=paste("T",1:126,sep=""),
                          add.X=add.Xdata,add.id=paste("NT",1:5,sep=""))
finpines_add
as.data.frame(finpines_add)
```

`shrinkedge` *Shrink the edges for the point pattern*

Description

Sometimes, the window of point pattern is too large, we can shrink its edges to the wanted window of point pattern

Usage

```
shrinkedge(X, xwidth, ywidth, id)
```

Arguments

`X`	A point pattern (object of class "ppp").
`xwidth`	Distance from the edges x coordinates,note xwidth is the half of your width wanted, because the width will shrink both edges in the x coordinates.
`ywidth`	Distance from the edges y coordinates,note xwidth is the half of your width wanted, because the width will shrink both edges in the y coordinates.
`id`	Specific identification number of the points in the point pattern.

Details

When expanding for the full pattern, and using expandedge() with a positive xwidth and ywidth; conversely, shrinking for the full pattern, and using shrinkedge () with a negative xwidth and ywidth,the parts of the pattern that are at a distance less than -width from an edge are discarded.

Value

Result returns original data (not a point pattern with the same structure as trees), id is new id of points after edges expanded and old.id is orginal id of points. If xwidth or ywidth = 0, trees are returned unchanged.

Author(s)

Zongzheng Chai, chaizz@126.com

Examples

```
library(spatstat.data)
data(finpines)
finpines$window
# window: rectangle = [-5, 5] x [-8, 2] metres

#Shrink the rectangle [-5,5]x[-8,2] to [-3,3]x[-5,-1]
shrink.trees<- shrinkedge(finpines,xwidth=2,ywidth=3,id=1:126)
shrink.trees

# Show the changes by figures
opar<-par(mfrow=c(1,2))
plot(finpines$x,finpines$y)
text(finpines$x,finpines$y,1:126)
rect(-3,-5,3,-1,border="red")
plot(shrink.trees$x,shrink.trees$y)
text(shrink.trees$x,shrink.trees$y,shrink.trees$old.id)
par(opar)
```

`simtreecom` *Simulated a forest community data.*

Description

Simulated a forest community data.

Usage

```
simtreecom(size,nspe,nspe_max,xrange,
           yrange,dbhrange,type="com",lambda=1)
```

Arguments

`size`	Pool of species
`nspe`	Number of species
`nspe_max`	Maximum number of species
`xrange`	Range of X axes
`yrange`	Range of Y axes
`dbhrange`	Range of DBH
`type`	Data type, "ppp" or "com"
`lambda`	Degree of coordinate aggregation

Value

Data frame of forest community data

Author(s)

Zongzheng Chai, chaizz@126.com

References

NULL

Examples

```
treecom<-simtreecom(size=10,nspe=10,nspe_max=36,xrange=c(0,100),
          yrange=c(0,100),dbhrange=c(5,50),type="com",lambda=1)
treecom
```

---

spastr *Computing forest spatial structure indices.*

Description

Computing forest spatial structure indices.

Usage

```
spastr(X_df,smark=c("spe","storey","dbh","cw","x","y"),
                 buffer = FALSE,
                 xrange=c(0,100),yrange=c(0,100),
                 xwidth=5,ywidth=5,
                 buf.xwid = 5, buf.ywid = 5,
                 exclusion = FALSE)
```

Arguments

X_df	Forest community data (object of class "data.frame")
smark	Selected marks to compute forest spatial structure indices.
buffer	if buffer is TRUE,show the all of data with buffer and core zone. if buffer is FALSE,only show the data in the core zone.
buf.xwid	The width of buffer zone in the x coordinates of data points.
buf.ywid	The width of buffer zone in the y coordinates of data points.
xrange	Range of X axes
yrange	Range of Y axes
xwidth	Length of buffer zone of X axes
ywidth	Length of buffer zone of Y axes
exclusion	Wether including the data of the buffer zone, if exclude is TRUE the result return the data remove the data of the buffer zone, and only the data of the core zone;if exclude is FALSE the result return the data with buffer and core zone.

Details

NULL

Value

NULL

Author(s)

Zongzheng Chai, chaizz@126.com

Examples

```
library(spatstat)
data(treecom_example)
head(treecom_example)
treecom_spastr<-spastr(X_df=treecom_example,
              xrange=c(0,100),yrange=c(0,100),
              xwidth=10,ywidth=10,
              buf.xwid =10, buf.ywid = 10,
              smark=c("spe","storey","dbh","cw","x","y"))
treecom_spastr
```

---

storeydvd *Dividing the forest storeies*

Description

Forest overstory can be divided the many stories by dividing the range of tree height.

Usage

```
storeydvd(height, minH = 2.5, maxH = max(height),
      storeynum = 6, include.lowest = TRUE, right = TRUE)
```

Arguments

height	Data of tree height
minH	Minimum value of height
maxH	Maximum value of height
storeynum	Number of storey
include.lowest	Logical, indicating if an "[i]" equal to the lowest (or highest, for right=FALSE) "breaks" value should be included. See the function "cut".
right	Logical, indicating if the intervals should be closed on the right (and open on the left) or vice versa.See the function "cut".

Value

heightfreq is the frequency of tree height heightdata is the result after divided the storey

Author(s)

Zongzheng Chai, chaizz@126.com

References

None

Examples

```
library(spatstat)
data(finpines)
finpinesdata<-as.data.frame(finpines)
####Dividing the stories
finpines.storey<-storeydvd(finpinesdata$height,storeynum=6)
finpines.storey

####Computing the storey differation
##Add the storey mark for finepines
finpines.addstorey<-addmark.ppp(finpines,
              add.mark=finpines.storey$heightdata[,2:3],
                    add.name=c("interval","storey" ))
finpines.addstorey
##Compute the storeies of nearest neighbour in the point pattern
finpines.nnstorey<-nnIndex(finpines.addstorey,smark="storey",N=4,
                           buf.xwid =2,buf.ywid=3)
##Compute the stoery differation
finpines.H<-fsasN4(finpines.nnstorey$nnstorey,match.fun=differ)
finpines.H
```

---

tree.ppp *Sample data for analizing the forest spatial structure.*

Description

A sample data of field survey, to help us understanding the method of forest spatial structure analysis.class is "ppp".

Usage

```
data("tree.ppp")
```

Details

The function "ppp" in the package spatstat.geom

References

None

Examples

```
data(tree.ppp)
```

`treecom_example` *Example data for analizing the forest community.*

Description

A example data of field survey, to help us understanding the method of forest spatial structure analysis.

Usage

```
data("treecom_example")
```

Format

A data frame

`x` x coordinates of trees

`y` y coordinates of trees

`id` Specific identification number of trees

`spe` Specific code of tree species in the point pattern

`dbh` Diameter at breast height (DBH),1.3 m. unit:cm

`ht` Tree height,unit:m

`cw` Crown width,unit:m

`hcb` height of crown base,unit:m

`volume` volume of individual trees,unit:m3

`quality` Quality of individual trees

`storey` forest storey

References

None

Examples

```
data(treecom_example)
treecom_example
```

`treedata` *Sample data for analizing the forest spatial structure.*

Description

A sample data of field survey, to help us understanding the method of forest spatial structure analysis.

Usage

```
data("treedata")
```

Format

A data frame with 41 observations on the following 11 variables.

`tree.id` Specific identification number of trees in the point pattern

`x` x coordinates of trees

`y` y coordinates of trees

`sp.code` Specific code of tree species in the point pattern

`dbh.cm` Diameter at breast height (DBH),1.3 m. unit:cm

`h.m` Tree height,unit:m

`storey` forest storey

`crownwid.m` Crown width,unit:m

`group` Tree group, which can be divided exotic,pioneer,transtional,climax,5 types

`biomass.kg` Tree biomass,unit:kg

`quality` Quality of individual trees

References

None

Examples

```
data(treedata)
```

uniform.angle *A forest spatial structure index characterizing tree spatial distribution patterns*

Description

The uniform angle index (UAI) is used to characterize the spatial distribution of a forest community or of individual tree species within that community.By gradually comparing the included 4 angles with the standard angle (72).

Usage

```
uniform.angle(x, para)
```

Arguments

x	Moving clockwise around reference tree,4 angle were obtained among adjacent neighbours
para	standard angle (72)

Value

Result will retrun five values,0,0.25,0.5,0.75,1,which means none, one, two, three, or four angles are smaller than 72, respectively.

Author(s)

Zongzheng Chai, chaizz@126.com

References

Zhao ZH, Hui GY, Hu YB, Wang HX, Zhang GQ, and von Gadow K. 2014. Testing the significance of different tree spatial distribution patterns based on the uniform angle index. Canadian Journal of Forest Research 44:1419-1425.

Examples

```
angle1<-c(80,80,80,80)
angle2<-c(80,80,80,70)
angle3<-c(80,80,60,70)
angle4<-c(80,50,60,70)
angle5<-c(40,50,60,70)

uai1<-uniform.angle(angle1,para=72)
uai1
uai2<-uniform.angle(angle2,para=72)
uai2
uai3<-uniform.angle(angle3,para=72)
```

```
uai3
uai4<-uniform.angle(angle4,para=72)
uai4
uai5<-uniform.angle(angle5,para=72)
uai5
```

Index

addmark.ppp, 4

buffer, 5

crowding, 6

differ, 7
dominance, 8

expandedge, 9

forestSAS *(*forestSAS-package*)*, 2
forestSAS-package, 2
forestSAS_simapp, 11
fsasN4, 11

ggrosechart, 12

ideal, 14

list_to_matrix, 15

mingling, 15

nnangle, 16
nnid, 17
nnIndex, 18
nnoverlap, 21

opt_spastr, 22

pv, 23

rebuild.ppp, 25

shrinkedge, 27
simtreecom, 28
spastr, 29
storeydvd, 30

tree.ppp, 31
treecom_example, 32
treedata, 33

uniform.angle, 34