

Minha função

Função precip.abund

Help da Função

precip.abund

package:-

R Documentation

Plotar gráficos de abundância e precipitação de dados coletados e simulados

Description:

A função `precip.abund` calcula índice de abundância através da divisão do número de indivíduos por quantidade de água coletada, usando esse índice para plotar um gráfico que mostra a distribuição desse índice por quantidade de precipitação (mM) por coleta. Ela gera também um gráfico com as mesmas características já descritas, porém, com os dados misturados, simulando assim diferentes situações com os dados já conhecidos.

Usage:

```
precip.abund(dados,L=TRUE)
```

Arguments:

`dados` Data.frame com quatro colunas: primeira com o número da coleta, segunda com a quantidade de água coletada, terceira com o número de indivíduos coletados, quarta com a precipitação acumulada (período escolhido pelo usuário).

`L` Argumento para transformação da quantidade de água coletada de mililitro para litro. O argumento deve ser `=TRUE` quando o usuário quiser realizar a transformação.

Details:

Caso o usuário queira realizar a transformação com o argumento $L=T$, a quantidade de água deve estar obrigatoriamente em mililitro. Caso $L=F$, o esperado é que os valores já estejam em litro.

Para que o produto final da função seja correto, as colunas do `data.frame` devem estar dispostas exatamente como está na explicação do argumento `dados`, pois há o cálculo de índices e indexação de colunas do `data.frame`.

Ao rodar a função, serão emitidas mensagens para lembrar ao usuário quais valores devem estar em cada coluna do `data.frame`.

Value:

A função retorna dois gráficos.

O primeiro tem como valor nas barras a precipitação por coleta e os pontos e linhas são

os índices de abundância calculados e plotados por coleta.

O segundo tem como valor nas barras a precipitação simulada por coleta e os pontos e linhas são

os índices de abundância simulados calculados por coleta.

Warning:

Caso a entrada de dados não seja em formato `data.frame`, a função é interrompida e é gerada uma mensagem avisando que a entrada de dados não está no formato correto.

Author(s):

Amanda Alves Camargo

amandacamargo@usp.br

See Also:

As funções `sample()` e `plot()` podem ajudar o usuário a entender o retorno da função, principalmente com relação ao segundo gráfico.

Examples:

```
##Criando objeto com o número das coletas  
coletas<- rep(1:12,each=10)  
##Criando objeto com a quantidade de água coletada  
agua<- seq(from=100,to=350,by=50)
```

```
agual<-rep(agua,20)
##Criando objeto com o número de indivíduos coletados
individuos<-rep(0:30,4)
##Criando objeto com os valores de precipitação
precip<-seq(from=70,to=180,by=10)
precip1<-rep(precip,10)
#Montando o data.frame com os objetos criados
dados<-data.frame(coletas,agual,individuos,precip1)
#Rodando a função
precip.abund(dados,L=T)

##Criando vetor de números aleatórios
vetor<-c(1,3,6,8,90,4,5)
##Rodando a função
precip.abund(vetor,L=T)
#Ao rodar, o usuário receberá a mensagem de erro avisando que a entrada de
dados não é um data.frame
```

Código da Função

```
#####Script da Função: Precipitação e Abundância (precip.abund)#####
#Amanda Alves Camargo
#amandacamargo@usp.br

precip.abund<-function(dados,L=TRUE) #a função simabund possui dois
argumentos: dados, que deve ser obrigatoriamente um data.frame e L, que
deverá ser TRUE caso o usuário deseje converter o volume de água (mL), em
litro.0 default da função é L=TRUE.
{
  if(class(dados)!="data.frame") #indica que se a classe de dados inseridos
não for um data.frame
  {
stop("o objeto não é um data.frame")} #a função não rodará e aparecerá a
seguinte mensagem:"o objeto não é um data.frame", caso a entrada de dados
não seja um data.frame

  if(L==TRUE) #se o argumento litro for verdadeiro
{litro<-c(dados[,2]/1000)} #a função pegará a segunda coluna do data.frame
(que deve obrigatoriamente ter os valores das quantidades de água em mL) e
dividir por 1000, para assim dar o valor transformado em litros
else #caso contrário
{litro=dados[,2]} #manter os valores originais da coluna 2, todas as
linhas
{message("se litro=TRUE a quantidade de água deve estar em ml no seu
data.frame")} #avisa o usuário que se litro for verdadeiro, para que
atransformação seja feita de forma correta, a quantidade de água deve estar
em mL no data.frame
{message("verifique se a coluna 2 é a que contém as quantidade de água")}
```

#avisa o usuário que a coluna 2 do data.frame deve conter obrigatoriamente os valores das quantidades de água

```
indice<-c(dados[,3]/litro) #cria o objeto indice, que possuirá o índice originado da divisão do número de indivíduos na coluna 3, pela quantidade de água que está no objeto litro
{message("verifique se a coluna 3 é a que contém o número de indivíduos")}
#avisa o usuário que a coluna 3 do data.frame de conter obrigatoriamente os valores de número de indivíduos
precip<-c(dados[,4]) #cria um objeto que contém um vetor com os valores de precipitação, que estão na coluna 4 do data.frame
{message("verifique se a coluna 4 é a que contém os valores de precipitação")} #avisa o usuário que a coluna 4 do data.frame de conter obrigatoriamente os valores de precipitação
dados2<-cbind(dados,indice) #cria um objeto,adiciona uma coluna nova com os valores do objeto indice aos dados originais
coleta<-c(dados[,1]) #cria um objeto que contém um vetor com os valores dos dados de coleta, que estão na coluna 1
{message("verifique se a coluna 1 é a que contém a informação da coleta de dados")} ##avisa o usuário que a coluna 1 do data.frame de conter obrigatoriamente as informações de coleta
fun1<-tapply(X=indice,INDEX=coleta,FUN=mean) #cria um objeto,indexa os valores da coluna indice, por coleta, e calcula a média desses valores
fun2<-tapply(X=precip,INDEX=coleta,FUN=mean) #cria um objeto,indexa os valores da coluna de precipitação, por coleta, e calcula a média desses valores
par(mfrow=c(1,2)) #indica que os gráficos plotados devem estar em apenas 1 linha, mas em 2 colunas (ou seja,um gráfico ao lado do outro)
```

```
par(mar=c(5,4,5,4)) #define os valores dos parâmetros das margens superior e inferior, direita e esquerda
```

```
plot1<-barplot(fun2, col='white', xlab="Coletas",ylab="Precipitação (mM)",axes=FALSE) #cria objeto, plota o gráfico de barras que contém os valores do objeto fun2, a cordas barras deve ser branca, o eixo x de nome "Coletas", o eixo y de nome "Precipitação (mM)",os valores de eixo não devem ser plotados
```

```
axis(2,at=seq(0,500,50)) #o eixo y deve ir de 0 a 500, com intervalo de 50
```

```
par(new=TRUE)#o próximo plot deve ser feito no mesmo gráfico, não em um gráfico novo
```

```
plot2<-plot(fun1, col='red',axes=FALSE,ann=FALSE,pch=16) #cria um objeto, esse novo gráfico, que será postado no anterior, plota os valores do objeto fun1, na cor vermelha,os valores de eixo não devem ser plotados,os dados devem ser plotados em pontos, no modelo 16
```

```
axis(4) #cria o eixo para os valores do plot 2 no lado direito do gráfico
```

```
mtext(text="Índice Larvário",side=4, line=2.2) #define que o eixo 4 deve conter o nome "Índice Larvário", e que a distância do nome para o eixo deve ser 2.2

lines(fun1,col="red") #liga os pontos do objeto fun1 plotados através de uma linha vermelha

amostra<-sample(litro) #cria um objeto, mistura os valores do objeto litro

amostra2<-sample(dados[,3]) #cria um objeto, mistura os valores da coluna 3 dos dados (que deve conter o número de indivíduos)
amostra3<-sample(dados[,4]) #cria um objeto, mistura os valores da coluna 4 dos dados (que deve conter os valores de precipitação)
indice2<-amostra2/amostra #cria um objeto, dá o valor do índice que é obtido ao se dividir o objeto que contém os valores misturados de indivíduos pelo objeto que contém os valores de litro misturados
dados3<-cbind(dados2,amostra,amostra2,amostra3,indice2) #adiciona os objetos com os valores misturados ao objeto de dados original
fun3<-tapply(X=indice2,INDEX=coleta,FUN=mean) #cria um objeto,indexa os valores da coluna indice2, por coleta, e calcula a média desses valores
fun4<-tapply(X=amostra3,INDEX=coleta,FUN=mean) #cria um objeto,indexa os valores da coluna amostra3, por coleta, e calcula a média desses valores

par(mar=c(5,4,5,4)) #define os valores dos parâmetros das margens superior e inferior, direita e esquerda

plot3<-barplot(fun4, col='white', xlab="Coletas",ylab="Precipitação Simulada (mM)",axes=FALSE) #cria objeto, plota o gráfico de barras que contém os valores do objeto fun4, a cor das barras deve ser branca, o eixo x de nome "Coletas", o eixo y de nome "Precipitação Simulada (mM)",os valores de eixo não devem ser plotados

axis(2,at=seq(0,500,50)) #o eixo y deve ir de 0 a 500, com intervalo de 50

par(new=TRUE) #o próximo plot deve ser feito no mesmo gráfico, não em um gráfico novo

plot4<-plot(fun3, col='red',axes=FALSE,ann=FALSE,pch=16) #cria um objeto, esse novo gráfico, que será postado no anterior, plota os valores do objeto fun3, na cor vermelha,os valores de eixo não devem ser plotados,os dados devem ser plotados em pontos, no modelo 16
axis(4) #cria o eixo para os valores do plot 2 no lado direito do gráfico

mtext(text="Índice Larvário Simulado",side=4, line=2.2) #define que o eixo 4 deve conter o nome "Índice Larvário Simulado", e que a distância do nome para o eixo deve ser 2.2

lines(fun3,col="red") #liga os pontos do objeto fun1 plotados através de uma linha vermelha

plotfinal<-c(plot1,plot2,plot3,plot4) #cria um vetor com os objetos que
```

contém os 4 diferentes gráficos que são gerados na função

```
return(plotfinal) #retorna o objeto final que contém os 4 gráficos gerados na função  
}
```

```
#####Fim da função#####
```

Documentação da Função

[help_da_funcao.txt](#)

[script_funcao.r](#)

[script_exemplo.r](#)

[outro_exemplo.csv](#)

From:
<http://ecor.ib.usp.br/> - **ecoR**

Permanent link:
http://ecor.ib.usp.br/doku.php?id=05_curso_antigo:r2017:alunos:trabalho_final:amandacamargo:minha_funcao

Last update: **2020/08/12 06:04**