Package ‘meteR’

June 27, 2016

Type Package
Title Fitting and Plotting Tools for the Maximum Entropy Theory of Ecology (METE)
Version 1.2
Date 2016-06-22
Author Andy Rominger, Cory Merow
Maintainer Cory Merow <cory.merow@gmail.com>
Description Fit and plot macroecological patterns predicted by the Maximum Entropy Theory of Ecology (METE).
License GPL-2
URL https://github.com/cmerow/meteR
BugReports https://github.com/cmerow/meteR/issues
VignetteBuilder knitr
LazyLoad yes
LazyData true
Depends R (>= 2.10)
Imports nleqslv, distr
Suggests testthat, knitr
RoxygenNote 5.0.1
NeedsCompilation no
Repository CRAN
Date/Publication 2016-06-27 17:22:44

R topics documented:

meteR-package ................................................................. 2
anbo ........................................................................ 3
arth ........................................................................... 4
downscaleSAR ............................................................... 5
## meteR-package

Analyses with the Maximum Entropy Theory of Ecology (METE)

<table>
<thead>
<tr>
<th>Function</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>ebar</td>
<td>6</td>
</tr>
<tr>
<td>empiricalSAR</td>
<td>7</td>
</tr>
<tr>
<td>ipd</td>
<td>8</td>
</tr>
<tr>
<td>logLik.meteDist</td>
<td>10</td>
</tr>
<tr>
<td>logLikZ</td>
<td>11</td>
</tr>
<tr>
<td>meteDist2Rank</td>
<td>12</td>
</tr>
<tr>
<td>meteESF</td>
<td>13</td>
</tr>
<tr>
<td>meteNu</td>
<td>15</td>
</tr>
<tr>
<td>metePhi</td>
<td>16</td>
</tr>
<tr>
<td>metePi</td>
<td>17</td>
</tr>
<tr>
<td>metePsi</td>
<td>18</td>
</tr>
<tr>
<td>meteSAR</td>
<td>19</td>
</tr>
<tr>
<td>meteSSF</td>
<td>21</td>
</tr>
<tr>
<td>meteTheta</td>
<td>22</td>
</tr>
<tr>
<td>mse</td>
<td>23</td>
</tr>
<tr>
<td>mseZ</td>
<td>24</td>
</tr>
<tr>
<td>plot.damuth</td>
<td>26</td>
</tr>
<tr>
<td>plot.meteDist</td>
<td>27</td>
</tr>
<tr>
<td>plot.meteRelat</td>
<td>28</td>
</tr>
<tr>
<td>plot.sar</td>
<td>29</td>
</tr>
<tr>
<td>predictESF</td>
<td>30</td>
</tr>
<tr>
<td>print.damuth</td>
<td>31</td>
</tr>
<tr>
<td>print.meteDist</td>
<td>32</td>
</tr>
<tr>
<td>print.meteESF</td>
<td>33</td>
</tr>
<tr>
<td>print.meteRelat</td>
<td>34</td>
</tr>
<tr>
<td>print.sar</td>
<td>34</td>
</tr>
<tr>
<td>residuals.meteDist</td>
<td>35</td>
</tr>
<tr>
<td>residuals.meteRelat</td>
<td>36</td>
</tr>
<tr>
<td>sad</td>
<td>37</td>
</tr>
<tr>
<td>sipd</td>
<td>38</td>
</tr>
<tr>
<td>spd</td>
<td>40</td>
</tr>
<tr>
<td>ssad</td>
<td>41</td>
</tr>
<tr>
<td>upscaleSAR</td>
<td>42</td>
</tr>
</tbody>
</table>

### Description

Fits and plots macroecological patterns predicted by the Maximum Entropy Theory of Ecology (METE)
Details

Package: meteR
Type: Package
Version: 1.0
Date: 2014-01-04
License: GPL-2

Author(s)

Andy Rominger, Cory Merow, John Harte

Maintainer: Cory Merow <cory.merow@gmail.com>

References


---

anbo

Community abundance data for a desert grassland (anza borrego)

Description

A dataset containing the community abundance data for plant species, as well as the locations of plots with respect to one another

Usage

anbo
Format

A data frame with 121 rows and 4 variables:

- **row**: plot coordinate
- **column**: plot coordinate
- **spp**: species ID
- **count**: number of individuals

---

**Arthropod community abundance data**

Description

A dataset containing the community abundance data for individuals, as well as their body mass.

Usage

```
arth
```

Format

A data frame with 547 rows and 3 variables:

- **spp**: species ID
- **count**: number of individuals
- **mass**: biomass

Source

**downscaleSAR**

| `downscaleSAR` | **Downscale the species area relationship (SAR) or endemics area relationship (EAR)** |

**Description**

Compute METE SAR by downscaling from some larger area \( A_0 \) to a smaller areas.

**Usage**

`downscaleSAR(x, A, A0, EAR = FALSE)`

**Arguments**

- `x`: an object of class meteESF
- `A`: numerical vector of areas (\( \leq A_0 \)) for which the METE prediction is desired
- `A0`: total study area
- `EAR`: logical. TRUE computes the endemics area relationship

**Details**

Downscaling is done non-iteratively (i.e. the SAD and SSAD are calculated based on state variables at the anchor scale \( A_0 \)) thus unlike the upscaling SAR function, downscaling can be computed for any arbitrary scale \( \leq A_0 \).

**Value**

an object of class `sar` inheriting from `data.frame` with columns `A` and `S` giving area and species richness, respectively

**Author(s)**

Andy Rominger <ajrominger@gmail.com>, Cory Merow

**References**


**See Also**

meteESF, meteSAR, empiricalSAR, upscaleSAR
Examples
data(anbo)
anbo.esf <- meteESF(spp=anbo$spp, abund=anbo$count)
anbo.thr.downscale <- downscaleSAR(anbo.esf, 2^((seq(-3, 4, length=7))), 16)
plot(anbo.thr.downscale)

## theoretical SARs from state variables only
thr.downscale <- downscaleSAR(meteESF(S0=40, N0=400), 2^((seq(-1,4,by=1))), 16)
thr.downscaleEAR <- downscaleSAR(meteESF(S0=40, N0=400), 2^((seq(-1,4,by=1))), 16, EAR=TRUE)
plot(thr.downscale, ylim=c(0, 40), col='red')
plot(thr.downscaleEAR, add=TRUE, col='blue')

**ebar**

*Relationship between mean metabolic rate (\( \bar{\epsilon} \)) and abundance*

Description

ebar calculates the relationship between average metabolic rate of a species and that species’ abundance. Also known as the Damuth relationship

Usage

ebar(x)

Arguments

x an object of class meteESF.

Details

See examples.

Value

An object of class meteRelaT. The object contains a list with the following elements.

- pred predicted relationship
- obs observed relationship

Author(s)

Andy Rominger <ajrominger@gmail.com>, Cory Merow

References

empiricalSAR

See Also

meteDist, sad.meteESF, metePsi

Examples

data(arth)
esf1 <- meteesf(spp=arth$spp, 
    abund=arth$count, 
    power=arth$mass^(.75), 
    minE=min(arth$mass^(.75)))
damuth <- ebar(esf1)

data(arth)
esf1 <- meteesf(spp=arth$spp, 
    abund=arth$count, 
    power=arth$mass^(.75), 
    minE=min(arth$mass^(.75)))
damuth <- ebar(esf1)

empiricalSAR  Empirical SAR or EAR

Description

computes observed SAR or EAR from raw data

Usage

empiricalSAR(spp, abund, row, col, x, y, Amin, A0, EAR = FALSE)

Arguments

  spp  vector of species identities
  abund numeric vector abundances associated with each record
  row   identity of row in a gridded landscape associated with each record, or desired number of rows to divide the landscape into
  col   identity of column in a gridded landscape associated with each record, or desired number of columns to divide the landscape into
  x     the x-coordinate of an individual if recorded
  y     the y-coordinate of an individual if recorded
  Amin  the smallest area, either the anchor area for upscaling or the desired area to downscale to
  A0    the largest area, either the area to upscale to or the total area from which to downscale
  EAR   logical, should the EAR or SAR be computed

Details

Currently only doublings of area are supported. There are several options for specifying areas. Either row and col or x and y must be provided for each data entry (i.e. the length of row and col or x and y must equal the length of spp and abund). If x and y are provided then the landscape is gridded either by specifying Amin (the size of the smallest grid cell) or by providing the number or desired rows and columns via the row and col arguments. If only row and col are provided these are taken to be the row and column identities of each data entry.
Value

an object of class sar inheriting from data.frame with columns A and S giving area and species richness, respectively

Author(s)

Andy Rominger <ajrominger@gmail.com>, Cory Merow

References


See Also

meteESF, meteSAR, downscaleSAR, upscaleSAR

Examples

data(anbo)
anbo.obs.sar <- empiricalsar(anbo$spp, anbo$count, anbo$row, anbo$col, Amin=1, A0=16)
plot(anbo.obs.sar)
anbo.obs.ear <- empiricalsar(anbo$spp, anbo$count, anbo$row, anbo$col, Amin=1, A0=16, EAR=TRUE)
plot(anbo.obs.ear)

## empirical SAR from simulated x, y data
anbo$x <- runif(nrow(anbo), 0, 1) + anbo$column
anbo$y <- runif(nrow(anbo), 0, 1) + anbo$row
metesar(anbo$spp, anbo$count, x=anbo$x, y=anbo$y, row=4, col=4)

ipd

Individual Power Distribution

Description

ipd.meteESF calculates the distribution $\Psi(e | N0, S0, E0)$, the distribution of metabolic rates across all individuals in a community

Usage

ipd(x, ...)

## S3 method for class 'meteESF'
ipd(x, ...)

Arguments

x an object of class meteESF.
...
additiona arguments to be passed to methods
Details

See examples.

Value

An object of class `metedist`. The object contains a list with the following elements.

- **data**: The data used to construct the prediction
- **d**: Density function
- **p**: Cumulative density function
- **q**: Quantile function
- **r**: Random number generator
- **La**: Vector of Lagrange multipliers
- **state.var**: State variables used to constrain entropy maximization
- **type**: Specifies the type of distribution is 'sad'

Author(s)

Andy Rominger <ajrominger@gmail.com>, Cory Merow

References


See Also

`metedist`, `sad.meteESF`, `metePsi`

Examples

```r
data(arth)
esf1 <- meteesf(spp=arth$spp, 
                abund=arth$count, 
                power=arth$mass^(N7UIII, 
                minE=min(arth$mass^(N7UIII))

ipd1 <- ipd(esf1)
```
logLik.meteDist

Compute log-likelihood of a meteDist object

Description

logLik.meteDist computes log-likelihood of a meteDist object

Usage

## S3 method for class 'meteDist'
logLik(object, ...)

Arguments

- object: a meteDist object
- ...: arguments to be passed

Details

Degrees of freedom are assumed to be equal to the number of Lagrange multipliers needed to specify the METE prediction. See Examples for usage.

Value

object of class logLik

Author(s)

Andy Rominger <ajrominger@gmail.com>, Cory Merow

References


See Also

sad, ssad, ipd, sipd

Examples

data(arth)
## object holding ecosystem structure function
esf1 <- meteESF(spp=arth$spp,
               abund=arth$count,
               power=arth$mass*(.75),
               minE=min(arth$mass*(.75))
## calculate individual power distribution and its likelihood
ipd1 <- ipd(esf1)
logLik(ipd1)
Compute log-likelihood z-score

Description

logLikZ.meteDist computes a log-likelihood z-score by simulation from a fitted METE distribution.

Usage

logLikZ(x, ...)

## S3 method for class 'meteDist'
logLikZ(x, nrep = 999, return.sim = FALSE, ...)

Arguments

x
a meteDist object

... arguments to be passed to methods

nrep
number of simulations from the fitted METE distribution

return.sim
logical; return the simulated likelihood values

Details

logLikZ.meteDist simulates from a fitted METE distribution (e.g. a species abundance distribution or individual power distribution) and calculates the likelihood of these simulated data sets. The distribution of these values is compared against the likelihood of the data to obtain a z-score, specifically $z = \left(\frac{\text{logLik}_{\text{obs}} - \text{mean(logLik}_{\text{sim}}))}{\text{sd(logLik}_{\text{sim}}))}\right)^2$. This value is squared so that it will be approximately Chi-squared distributed and a goodness of fit test naturally arises as $1 - \text{pchisq}(z, df=1)$.

Value

list with elements

z The z-score

sim nrep Simulated values (scaled by mean and sd as is the z-score) if return.sim=TRUE, NULL otherwise

Author(s)

Andy Rominger <ajrominger@gmail.com>, Cory Merow

References

See Also

mseZ.meteDist

Examples

data(arth)
## object holding ecosystem structure function
esf1 <- meteESF(spp=arth$spp,  
    abund=arth$count,  
    power=arth$mass*(.75),
    minE=min(arth$mass*(.75)))
## calculate individual power distribution
ipd1 <- ipd(esf1)
## calculate z-score, keeping all simulated log likelihoods for plotting
llz <- logLikZ(ipd1, nrep=100, return.sim=TRUE)

plot(density(llz$sim), xlim=range(c(llz$sim, llz$obs)),  
    xlab='scaled log(likelihood)^2', col='red')
abline(v=llz$z, lty=2)
legend('top', legend=c('data', 'simulated'), col=c('black', 'red'),  
    lty=c(1,1), bty='n')

Description

meteESF calculate the rank distribution of a meteDist object

Usage

meteDist2Rank(x)

Arguments

x meteDist object

Details

Extracts the predicted rank distribution from a meteDist object. This is effectively the quantile function of the distribution. Used, e.g., in plot.meteDist

Value

A vector of predicted quantiles, typically used to compare against data as in plot.meteDist

Author(s)

Andy Rominger <ajrominger@gmail.com>, Cory Merow
References


Examples

```r
data(arth)
esf1 <- meteesf(spp=arth$spp, 
                 abund=arth$count, 
                 power=arth$mass^(N7UII), 
                 mine=min(arth$mass^(N7UII)))
sad1 <- sad(esf1)
metedist2Rank(sad1)
```

**Description**

`meteesf` Calculates the “ecosystem structure function” $R(n, \epsilon)$ which forms the core of the Maximum Entropy Theory of Ecology

**Usage**

```r
meteesf(spp, abund, power, S0 = NULL, N0 = NULL, E0 = NULL, minE)
```

**Arguments**

- `spp` A vector of species names
- `abund` A vector of abundances
- `power` A vector of metabolic rates
- `S0` Total number of species
- `N0` Total number of individuals
- `E0` Total metabolic rate; defaults to `N0*1e6` if not specified or calculated from `power` to allow one to fit models that do not depend on metabolic rates
- `minE` Minimum possible metabolic rate

**Details**

Uses either data or state variables to calculate the Ecosystem Structure Function (ESF). `power` nor `E0` need not be specified; if missing an arbitrarily large value is assigned to `E0` ($N0*1e6$) such that it will minimally affect estimation of Lagrange multipliers. Consider using sensitivity analysis to confirm this assumption. Examples show different ways of combining data and state variables to specify constraints.
Value

An object of class `meteesf` with elements

data  The data used to construct the ESF
emin  The minimum metabolic rate used to rescale metabolic rates
La  Vector of Lagrange multipliers
La.info  Termination information from optimization procedure
state.var  State variables used to constrain entropy maximization
Z  Normalization constant for ESF

Author(s)

Andy Rominger <ajrominger@gmail.com>, Cory Merow

References


See Also

`metePi`

Examples

```r
## case where complete data available
esf1 <- meteesf(spp=arth$spp, 
  abund=arth$count, 
  power=arth$mass^(.75), 
  minE=min(arth$mass^(.75)))
esf1

## excluding metabolic rate data
esf2 <- meteesf(spp=arth$spp, 
  abund=arth$count)
esf2

## using state variables only
esf3 <- meteesf(S0=50, N0=500, E0=5000)
esf3
esf4 <- meteesf(S0=50, N0=500)
esf4
```
meteNu

Equation of the PMF for the METE species metabolic rate distribution

Description

meteNu is a low level function to calculate the value of \( \nu(e|N_0, S_0, E_0) \) (the distribution of metabolic rates/power across all species in a community) at the given value of \( e \); vectorized in \( e \).

Usage

meteNu(e, la1, la2, Z, S0, N0, E0)

Arguments

e 
the value (metabolic rate/power) at which to calculate \( \Psi \)

la1, la2 
Lagrange multipliers

Z 
partition function

S0 
Total number of species

N0 
Total number of individuals

E0 
Total metabolic rate

Details

Typically only used in spd.meteESF and not called by the user.

Value

numeric vector of length equal to length of \( e \)

Author(s)

Andy Rominger <ajrominger@gmail.com>, Cory Merow

References


See Also

spd.mete
Examples

```r
data(arth)
esf1 <- meteESF(spp=arth$spp,
               abund=arth$count,
               power=arth$mass^(.75),
               minE=min(arth$mass^(.75)))
meteNu(1:10,
       esf1$La[1],esf1$La[2],
       esf1$Z,esf1$state.var['S0'],
       esf1$state.var['N0'],
       esf1$state.var['E0'])
```

**metePhi**

*Equation of the METE species abundance distribution*

Description

`metePhi` returns the species abundance distribution (Φ(n)) predicted by METE; vectorized in n

Usage

`metePhi(n, la1, la2, Z, S0, N0, E0)`

Arguments

- `n` the value (number of individuals) at which to calculate `Φ`
- `la1, la2` Lagrange multipliers
- `Z` partition function
- `S0` Total number of species
- `N0` Total number of individuals
- `E0` Total metabolic rate

Details

See Examples

Value

numeric

Author(s)

Andy Rominger <ajrominger@gmail.com>, Cory Merow
References

See Also
sad.mete

Examples

```r
esf1 <- meteESF(spp = arth$spp, 
                 abund = arth$count, 
                 power = arth$mass^(.75), 
                 minE = min(arth$mass^(.75)))
meterphi(min(arth$mass^(.75)), 
         esf1$La[1], esf1$La[2], 
         esf1$Z, esf1$state.var['S0'], 
         esf1$state.var['N0'], 
         esf1$state.var['E0'])
```

---

**metePi**

*Equation of the PMF of the METE spatial species abundance distribution*

**Description**

`metePi` is a low level function that returns the spatial species abundance distribution \( P_i(n) \) predicted by METE; vectorized in \( n \)

**Usage**

`metePi(n, la, n0)`

**Arguments**

- `n` A vector giving abundances of each entry
- `la` The spatial Lagrange multiplier returned by `meteSSF`
- `n0` Total abundance in area \( A_0 \)

**Details**

See Examples

**Value**

A numeric vector giving the probability of each entry in \( n \)
Author(s)

Andy Rominger <ajrominger@gmail.com>, Cory Merow

References


See Also

metePi

Examples

metePi(0:10, 0.01, 100)

Description

metePsi is a low level function to calculate the value of \( \Psi(e|N_0, S_0, E_0) \) (the distribution of metabolic rates/power across all individuals in a community) at the given value of \( e \); vectorized in \( e \).

Usage

metePsi(e, la1, la2, Z, S0, N0, E0)

Arguments

e the value (metabolic rate/power) at which to calculate \( \Psi \)

la1, la2 Lagrange multipliers

Z partition function

S0 Total number of species

N0 Total number of individuals

E0 Total metabolic rate

Details

Typically only used in ipd.meteESF and not called by the user.

Value

numeric vector of length equal to length of \( e \)
Author(s)
Andy Rominger <ajrominger@gmail.com>, Cory Merow

References

See Also
ipd.mete

Examples
data(arth)
esf1 <- meteESF(spp=arth$spp, abund=arth$count, power=arth$mass^(.75), minE=min(arth$mass^(.75)))
metepsi(1:10, esf1$La[1],esf1$La[2], esf1$Z,esf1$state.var['S0'], esf1$state.var['N0'], esf1$state.var['E0'])

meteSAR

Compute METE species area relationship (SAR)

Description
Uses raw data or state variables to calculate METE SAR and EAR (endemics area relationship) as well as compute the observed SAR or EAR from data, if provided

Usage
meteSAR(spp, abund, row, col, x, y, S0 = NULL, N0 = NULL, Amin, A0, upscale = FALSE, EAR = FALSE)

Arguments
spp vector of species identities
abund numeric vector abundances associated with each record
row identity of row in a gridded landscape associated with each record, or desired number of rows to divide the landscape into
col identity of column in a gridded landscape associated with each record, or desired number of columns to divide the landscape into
x the x-coordinate of an individual if recorded
\( y \) the y-coordinate of an individual if recorded

sP total number of species

N0 total abundance

Amin the smallest area, either the anchor area for upscaling or the desired area to downscale to

A0 the largest area, either the area to upscale to or the total area from which to downscale

upscale logical, should upscaling or downscaling be carried out

EAR logical, should the EAR or SAR be computed

Details

Currently only doublings of area are supported. Predictions and comparison to data can be made via several options. If spp and abund are not provided then only theoretical predictions are returned without empirical SAR or EAR results. In this case areas can either be specified by providing Amin and A0 from which a vector of doubling areas is computed, or by providing row, col and A0 in which case row and col are taken to be the number of desired rows and columns used to construct a grid across the landscape. If data are provided in the form of spp and abund then either row and col or x and y must be provided for each data entry (i.e. the length of row and col or x and y must equal the length of spp and abund). If x and y are provided then the landscape is gridded either by specifying Amin (the size of the smallest grid cell) or by providing the number or desired rows and columns via the row and col arguments.

SARs and EARs can be predicted either interactively or non-iteratively. In the non-iterative case the SAD and SSAD (which are used to calculate the SAR or EAR prediction) are derived from state variables at one anchor scale. In the iterative approach state variables are re-calculated at each scale. Currently downscaling and upscaling are done differently (downscaling is only implemented in the non-iterative approach, whereas upscaling is only implemented in the iterative approach). The reason is largely historical (downscaling as originally done non-iteratively while upscaling was first proposed in an iterative framework). Future implementations in meteR will allow for both iterative and non-iterative approaches to upscaling and downscaling. While iterative and non-iterative methods lead to slightly different predictions these are small in comparison to typical ranges of state variables (see Harte 2011).

Value

an object of class meteRelat with elements

pred predicted relationship; an object of class sar
obs observed relationship; an object of class sar

Author(s)

Andy Rominger <ajrominger@gmail.com>, Cory Merow

References

meteSSF

See Also
sad, meteESF, metePi

Examples

## Not run:
data(anbo)

## using row and col from anbo dataset
anbo.sar1 <- metesar(anbo$spp, anbo$count, anbo$row, anbo$col, Amin=1, A0=16)
plot(anbo.sar1)

## using simulated x, y data
anbo.sar2 <- metesar(anbo$spp, anbo$count, x=anbo$x, y=anbo$y, row=4, col=4)
plot(anbo.sar2)

## using just state variable
thr.sar <- metesar(Amin=1, A0=16, S0=50, N0=500)

## End(Not run)

meteSSF

Description

meteSSF calculates the “spatial structure function” \( \Pi(n) \) (analogous to the ecosystem structure function). From the SSF the spatial abundance distribution can be calculated.

Usage

meteSSF(spp, sppID, abund, row, col, x, y, n0 = sum(abund), A, A0)

Arguments

spp A vector of species names
sppID A character giving the name of the desired species (as it appears in ‘spp’)
abund A vector of abundances
row A vector of row IDs for each observation
col A vector of column IDs for each observation
x A vector of x coordinates for each observation
y A vector of y coordinates for each observation
n0 Total abundance in area A0
A The area at which abundances were recorded
A0 Total study area
Details
Uses either data or state variables to calculate the Spatial Structure Function (SSF). Uses internal code to determine when computation-saving approximations can be safely made.

Value
An object of class meteSSF with elements
- data: The data used to construct the SSF
- La: Vector of Lagrange multipliers
- La.info: Termination information from optimization procedure
- state.var: State variables used to constrain entropy maximization

Author(s)
Andy Rominger <ajrominger@gmail.com>, Cory Merow

References

See Also
metePi

Examples
```r
data(anbo)
## calculate SSF Pi
pi1 <- meteSSF(anbo$spp, 'crrc', anbo$count, row=anbo$row, col=anbo$column, A=1, A0=16)
pi1
```

### meteTheta

*Equation of the PMF for the METE Intra-specific metabolic rate distribution*

Description
Distribution of metabolic rates over individuals within a species of abundance n0

Usage
```r
meteTheta(e, n, la2)
```
mse

Arguments

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>e</td>
<td>Metabolic rate</td>
</tr>
<tr>
<td>n</td>
<td>Number of individuals in species</td>
</tr>
<tr>
<td>la2</td>
<td>Lagrange multiplier (lambda_2) as obtained from meteESF</td>
</tr>
</tbody>
</table>

Value

numeric vector of length equal to length of e

Author(s)

Andy Rominger <ajrominger@gmail.com>, Cory Merow

References


See Also

metePsi, ipd
Other Theta: sipd, sipd.meteESF

mse Computes mean squared error for rank or cdf

Description

mse.meteDist computes mean squared error for rank or cdf between METE prediction and data

Usage

mse(x, ...)

## S3 method for class 'meteDist'
mse(x, type = c("rank", "cumulative"), relative = TRUE, log = FALSE, ...)

Arguments

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>x</td>
<td>a meteDist object</td>
</tr>
<tr>
<td>...</td>
<td>arguments to be passed to methods</td>
</tr>
<tr>
<td>type</td>
<td>'rank' or 'cumulative'</td>
</tr>
<tr>
<td>relative</td>
<td>logical; if true use relative MSE</td>
</tr>
<tr>
<td>log</td>
<td>logical; if TRUE calculate MSE on logged distribution. If FALSE use arithmetic scale.</td>
</tr>
</tbody>
</table>
**Details**

See Examples.

**Value**

numeric; the value of the mean squared error.

**Author(s)**

Andy Rominger <ajrominger@gmail.com>, Cory Merow

**References**


**See Also**

 mseZ.meteDist

**Examples**

```r
data(arth)
esf1 <- meteESF(spp=arth$spp,
  abund=arth$count,
  power=arth$mass^(.75),
  minE=min(arth$mass^(.75)))
sad1 <- sad(esf1)
mse(sad1, type='rank', relative=FALSE)
ebar1 <- ebar(esf1)
mse(ebar1)
```

---

**mseZ**

*Compute z-score of mean squared error*

**Description**

mseZ.meteDist Compute z-score of mean squared error

**Usage**

```r
mseZ(x, ...)
```

## S3 method for class 'meteDist'
mseZ(x, nrep, return.sim = TRUE, type = c("rank",
  "cumulative"), relative = TRUE, log = FALSE, ...)
```
Arguments

- **x**: a *meteDist* object
- **...**: arguments to be passed to methods
- **nrep**: number of simulations from the fitted METE distribution
- **return.sim**: logical; return the simulated likelihood values
- **type**: either "rank" or "cumulative"
- **relative**: logical; if true use relative MSE
- **log**: logical; if TRUE calculate MSE on logged distribution. If FALSE use arithmetic scale

Details

`mseZ.meteDist` simulates from a fitted METE distribution (e.g. a species abundance distribution or individual power distribution) and calculates the MSE between the simulated data sets and the METE prediction. The distribution of these values is compared against the MSE of the data to obtain a z-score in the same way as `logLikZ`; see that help document for more details.

Value

list with elements

- **z**: The z-score
- **sim**: nrep Simulated values

Author(s)

Andy Rominger <ajrominger@gmail.com>, Cory Merow

References


See Also

`logLikZ`

Examples

```r
esf1 = meteESF(spp = arth$spp, 
    abund = arth$count, 
    power = arth$mass^(4/3), 
    minE = min(arth$mass^(4/3)))

sad1 = sad(esf1)

mseZ(sad1, nrep = 100, type = 'rank', return.sim = TRUE)
```
plot.damuth  

Plot the relationship between abundance and metabolic rate, i.e. objects of class damuth

Description

Plot abundance-metabolic rate relationship with flexibility to adjust plotting parameters

Usage

## S3 method for class 'damuth'
plot(x, add = FALSE, ...)

Arguments

- `x`: an object of class damuth
- `add`: logical; should new damuth object be added to current plot or made its own plot
- `...`: arguments passed to `plot`

Details

see examples

Author(s)

Andy Rominger <ajrominger@gmail.com>, Cory Merow

References


See Also

empiricalSAR, downscaleSAR, upscaleSAR, meteSAR

Examples

data(arth)
esf1 <- meteESF(arth$spp, arth$count, arth$mass^0.75)
ebar1 <- ebar(esf1)
plot(ebar1)
Description

plot.meteDist plots both the theoretical prediction and data for a metedist object using either a rank or cumulative distribution plot.

Usage

```r
## S3 method for class 'metedist'
plot(x, ptype = c("cdf", "rad"), th.col = "red",
     lower.tail = TRUE, add.legend = TRUE, add.line = FALSE, ...)
```

Arguments

- `x`: a metedist object
- `ptype`: type of plot; either "cdf" or "rad"
- `th.col`: line color of theoretical prediction
- `lower.tail`: logical; choose TRUE to highlight differences between data and theory at low abundance; choose FALSE to highlight differences at high abundance.
- `add.legend`: logical; add a legend
- `add.line`: add the curve for a fitted model to the existing plot
- `...`: arguments to be passed to `plot`

Details

plot.meteDist automatically extracts the prediction and data (if used in meteESF) from the metedist object. Additional plotting arguments can be passed to `...`.

Author(s)

Andy Rominger <ajrominger@gmail.com>, Cory Merow

References


See Also

sad, ipd, ssad, sipd, print.meteDist
Examples

```r
data(arth)
esf1 <- meteESF(spp=arth$spp,
abund=arth$count,
power=arth$mass^(.75),
minE=min(arth$mass^(.75)))
ipd1 <- ipd(esf1)
plot(ipd1)
plot(ipd1, ptype='rad')
```

Description

`plot.meteRelat` plots both the theoretical prediction and data for a `meteRelat` object.

Usage

```r
## S3 method for class 'meteRelat'
plot(x, add.legend = TRUE, th.col = "red", ...)
```

Arguments

- `x`: a `meteRelat` object
- `add.legend`: logical; add a legend
- `th.col`: line color of theoretical prediction
- `...`: arguments to be passed to `plot`

Details

`plot.meteRelat` automatically extracts the prediction and data (if used in `meteESF`) from the `metedist` object. Additional plotting arguments can be passed to `...`.

Author(s)

Andy Rominger <ajrominger@gmail.com>, Cory Merow

References


See Also

`meteSAR`
Examples

data(anbo)
anbo.sar <- meteSAR(anbo$spp, anbo$count, anbo$row, anbo$col, Amin=1, A0=16)
plot(anbo.sar)

plot.sar

Plot the species abundance distribution (SAR), i.e. objects of class sar

Description

Plot species or endemics area relationship with flexibility to adjust plotting parameters

Usage

## S3 method for class 'sar'
plot(x, add = FALSE, ...)

Arguments

x an object of class SAR made with
add logical; should new sar object be added to current plot or made its own plot
... arguments passed to plot

Details

see examples

Author(s)

Andy Rominger <ajrominger@gmail.com>, Cory Merow

References


See Also

empiricalSAR, downscaleSAR, upscaleSAR, meteSAR

Examples

data(anbo)
anbo.obs.sar <- empiricalSAR(anbo$spp, anbo$count, anbo$row, anbo$col, Amin=1, A0=16)
plot(anbo.obs.sar)
predictESF

Description

predict predicts the probabilities for given combinations of abundance and energy from the “ecosystem structure function” $R(n, \epsilon)$.

Usage

predictESF(esf, abund, power)

Arguments

- `esf`: A fitted object of class `meteESF`
- `abund`: A vector of abundances
- `power`: A vector of metabolic rates

Details

Uses a fitted object of class `meteESF` and user supplied values of abundance and power to predict values of the ESF.

Value

A data.frame with abundance, power, and the predicted value of the ESF.

Author(s)

Andy Rominger <ajrominger@gmail.com>, Cory Merow

References


See Also

`meteESF`

Examples

```r
## case where complete data available
esf1 <- meteESF(spp=arth$spp,
                 abund=arth$count,
                 power=arth$mass^(.75),
                 minE=min(arth$mass^(.75)))
predictESF(esf1),
```
print.damuth

abund=c(10,3),
power=c(.01,3))

print.damuth

Description

S3 method for class damuth

Usage

## S3 method for class 'damuth'
print(x, ...)

Arguments

x
an object of class damuth

...arguments to be passed to methods

Details

See Examples

Value

Returns the object silently

Author(s)

Andy Rominger <ajrominger@gmail.com>, Cory Merow

Examples

data(arth)
esf1 <- meteESF(arth$spp, arth$count, arth$mass^0.75)
ebar1 <- ebar(esf1)
print(ebar1)
print.meteDist

Print summaries of metedist objects

Description

S3 method for class meteDist

Usage

## S3 method for class 'meteDist'
print(x, ...)

Arguments

x a metedist object (e.g. from ipd.mete or sad.mete)
...

Details

Prints state variables and lagrange multipliers

Value

The metedist object is returned invisibly

Author(s)

Andy Rominger <ajrominger@gmail.com>, Cory Merow

References


Examples

data(arth)
esf1 <- meteESF(spp=arth$spp, 
    abund=arth$count, 
    power=arth$mass^(.75), 
    minE=min(arth$mass^(.75)))

ipd1 <- ipd(esf1)
ipd1
print.meteESF

Description
print.meteESF prints an object of class meteESF

Usage
## S3 method for class 'meteESF'
print(x, ...)

Arguments
x an object of class meteESF
... arguments to be passed

Details
See Examples

Value
x silently

Author(s)
Andy Rominger <ajrominger@gmail.com>, Cory Merow

References

Examples

data(arth)
esf1 <- meteESF(spp=arth$spp,
abund=arth$count,
power=arth$mass^(.75),
minE=min(arth$mass^(.75))
)
print(esf1)
esf1 # alternatively...
print.metereLat

Description
S3 method for class metereLat

Usage
## S3 method for class 'metereLat'
print(x, ...)

Arguments
x an object of class metereLat
... arguments to be passed to methods

Value
x silently

Author(s)
Andy Rominger <ajrominger@gmail.com>, Cory Merow

References

print.sar

Description
S3 method for class sar

Usage
## S3 method for class 'sar'
print(x, ...)

Arguments
x an object of class sar
... arguments to be passed to methods
residuals.meteDist  

Details

See Examples

Value

Returns the object silently

Author(s)

Andy Rominger <ajrominger@gmail.com>, Cory Merow

Examples

data(anbo)
anbo.sar <- meteSAR(anbo$spp, anbo$count, anbo$row, anbo$col, Amin=1, A0=16)
print(anbo.sar)
anbo.sar # alternatively

residuals.meteDist  Compute residuals between METE predictions and data of a meteDist object

Description

residuals.meteDist computes residuals between METE predictions and data of a meteDist object

Usage

## S3 method for class 'meteDist'
residuals(object, type = c("rank", "cumulative"),
          relative = TRUE, log = FALSE, ...)

Arguments

object  a meteDist object

type  'rank' or 'cumulative'

relative  logical; if true use relative MSE

log  logical; if TRUE calculate MSE on logged distribution. If FALSE use arithmetic scale.

...  arguments to be passed to methods

Details

See Examples. Typically not called directly by the user and rather used for calculating the mean square error with mse.meteDist. If type='rank' returned value will be of length equal to number of observations (e.g. number of species in case of SAD) but if type='cumulative' returned value will be of length equal to number of unique observations (e.g. number of unique abundances in case of SAR).
Value

A numeric vector giving residuals for each data point.

Author(s)

Andy Rominger <ajrominger@gmail.com>, Cory Merow

References


See Also

mse.meteDist

Examples

data(arth)
esf1 <- meteESF(spp=arth$spp,
    abund=arth$count,
    power=arth$mass^(N7UII),
    minE=min(arth$mass^(N7UII))
)sad1 <- sad(esf1)
residuals(sad1)

residuals.meteRelat  Compute residuals between METE predictions and date of a meteRelat object

Description

residuals.meteRelat computes residuals between METE predictions and data of a meteDist object.

Usage

## S3 method for class 'meteRelat'
residuals(object, ...)

Arguments

object  a meteRelat object
...    arguments to be passed

Details

See Examples. Typically not called directly by the user and rather used for calculating the mean square error with mse.meteRelat.
Value

a numeric vector giving residuals for each data point

Author(s)

Andy Rominger <ajrominger@gmail.com>, Cory Merow

References


See Also

mse.meteDist

Examples

data(arth)
esf1 <- meteESF(spp=arth$spp,  
  abund=arth$count,  
  power=arth$mass^(N7UII),  
  minE=min(arth$mass^(N7UIII))  
esbar1 <- esbar(esf1)  
residuals(ebar1)
Value

An object of class meteDist. The object contains a list with the following elements.

- **data**: The data used to construct the prediction
- **d**: density function
- **p**: cumulative density function
- **q**: quantile function
- **r**: random number generator
- **La**: Vector of Lagrange multipliers
- **state.var**: State variables used to constrain entropy maximization
- **type**: Specifies the type of distribution is ‘sad’

Author(s)

Andy Rominger <ajrominger@gmail.com>, Cory Merow

References


See Also

metePhi

Examples

data(arth)
esf1 <- meteESF(spp=arth$spp,
abund=arth$count,
power=arth$mass^(.75),
minE=min(arth$mass^(.75)))
sad1 <- sad(esf1)
sad1
sad1$r(20)
sad1$q(seq(0, 1, length=10))

sipd

Generic method to obtain the species-level individual power distribution (SIPD)

Description

Extract species level individual power distribution from ESF object and return object inheriting from meteDist. This distribution (Theta) describes the distribution of metabolic rates across the individuals of a species with n individuals.
Usage

sipd(x, ...)

## S3 method for class 'meteESF'
sipd(x, sppID, n, ...)

Arguments

x       An object of class meteESF (i.e. the fitted distribution $R(n, c)$)

...      arguments to be passed to methods

sppID    the name or index of the species of interest as listed in the spp argument passed
to meteESF

n        integer. Alternatively can extract METE prediction by indicating number of
individuals in the species

Details

If n is provided then only the theoretical prediction is returned (because data from multiple species
could map to the same n). Thus if data and prediction are desired use sppID.

Value

An object of class metedist. The object contains a list with the following elements.

data  The data used to construct the prediction
density function

cumulative density function

quantile function

random number generator

La  Vector of Lagrange multipliers

state.var  State variables used to constrain entropy maximization

type  Specifies the type of distribution is ‘sad’

Author(s)

Andy Rominger <ajrominger@gmail.com>, Cory Merow

References

Oxford University Press.

See Also

sad.meteESF, ipd.meteESF, metePsi

Other Theta: meteTheta
Species Power Distribution

**Description**

`spt.meteESF` calculates the distribution \( n(e | N_0, S_0, E_0) \), the distribution of average metabolic rates across for each species in a community.

**Usage**

```r
spt(x)
```

```r
## S3 method for class 'meteESF'
spd(x)
```

**Arguments**

- `x`: an object of class `meteESF`.

**Details**

See examples.

**Value**

An object of class `meteDist`. The object contains a list with the following elements.

- `data`: The data used to construct the prediction
- `d`: density function
- `p`: cumulative density function
- `q`: quantile function
- `r`: random number generator
- `La`: Vector of Lagrange multipliers
- `state.var`: State variables used to constrain entropy maximization
- `type`: Specifies the type of distribution is 'sad'

**Examples**

```r
data(arth)
esf1 <- meteESF(spp=arth$spp,
                 abun=arth$count,
                 power=arth$mass^{(.75)},
                 minE=min(arth$mass^{(.75)}))
sipd1 <- sipd(esf1, sppID=5)
sipd1
```
Species Spatial Abundance Distribution

Description

Species Spatial Abundance Distribution

Usage

ssad(x)

## S3 method for class 'meteSSF'
ssad(x)

Arguments

x An objects of class meteSSF; i.e. the spatial structure function II(n)

Author(s)

Andy Rominger <ajrominger@gmail.com>, Cory Merow

References

Examples

```r
data(anbo)
p1 <- meteSSF(anbo$spp, 'crr', anbo$count, row=anbo$row, col=anbo$col, A=1, A0=16)
plot(ssad(p1))
```

Description

Based on information at an anchor scale \((A0)\) calculate predicted species area relationship at larger scales

Usage

```r
upscaleSAR(x, A0, Aup, EAR = FALSE)
```

Arguments

- **x**: an object of class meteESF
- **A0**: the anchor scale at which community data are available.
- **Aup**: the largest area to which to upscale
- **EAR**: logical. TRUE computes the endemics area relationship; currently not supported

Details

Currently only doublings of area are supported and only the SAR (not EAR) is supported. Upscaling works by iteratively solving for the constraints \((S\) and \(N\) at larger scales) that would lead to the observed data at the anchor scale. See references for more details on this approach.

Value

an object of class `sar` inheriting from `data.frame` with columns `A` and `S` giving area and species richness, respectively

Author(s)

Andy Rominger <ajrominger@gmail.com>, Cory Merow

References


See Also

`meteESF`, `meteSAR`, `empiricalSAR`, `downscaleSAR`
Examples

data(anbo)
anbo.sar <- meteSAR(anbo$spp, anbo$count, anbo$row, anbo$col, Amin=1, A0=16)
anbo.sar
plot(anbo.sar, xlim=c(1, 2^10), ylim=c(5, 50), log='xy')

## get upscaled SAR and add to plot
anbo.esf <- meteESF(spp=anbo$spp, abund=anbo$count) # need ESF for upscaling
anbo.sarUP <- upscaleSAR(anbo.esf, 16, 2^10)
plot(anbo.sarUP, add=TRUE, col='blue')
Index

*Topic METE,
ebar, 6
ipd, 8
meteESF, 13
meteSSF, 21
predictESF, 30
print.meteESF, 33
sad, 37
spd, 40

*Topic MaxEnt,
ebar, 6
ipd, 8
meteESF, 13
predictESF, 30
print.meteESF, 33
sad, 37
spd, 40

*Topic MaxEnt
meteSSF, 21

*Topic datasets
anbo, 3
arth, 4

*Topic ecosystem
ebar, 6
ipd, 8
meteESF, 13
predictESF, 30
print.meteESF, 33
sad, 37
spd, 40

*Topic function
ebar, 6
ipd, 8
meteESF, 13
predictESF, 30
print.meteESF, 33
sad, 37
spd, 40

*Topic lagrange
ebar, 6
ipd, 8
meteESF, 13
meteSSF, 21
predictESF, 30
print.meteESF, 33
sad, 37
spd, 40

*Topic manip
meteNu, 15
metePhi, 16
metePsi, 18

*Topic multiplier,
ebar, 6
ipd, 8
meteESF, 13
meteSSF, 21
predictESF, 30
print.meteESF, 33
sad, 37
spd, 40

*Topic structure
ebar, 6
ipd, 8
meteESF, 13
predictESF, 30
print.meteESF, 33
sad, 37
spd, 40
anbo, 3
arth, 4
downscaleSAR, 5
ebar, 6
empiricalSAR, 7
ipd, 8
logLik.meteDist, 10
INDEX

logLikZ, 11
meteDist2Rank, 12
meteESF, 13
meteNu, 15
metePhi, 16
metePi, 17
metePsi, 18
meteR (meteR-package), 2
meteR-package, 2
meteSAR, 19
meteSSF, 21
meteTheta, 22, 39
mse, 23
mseZ, 24
plot.damuth, 26
plot.meteDist, 27
plot.meteRelat, 28
plot.sar, 29
predictESF, 30
print.damuth, 31
print.meteDist, 32
print.meteESF, 33
print.meteRelat, 34
print.sar, 34
residuals.meteDist, 35
residuals.meteRelat, 36
sad, 37
sipd, 23, 38
sipd.meteESF, 23
spd, 40
ssad, 41
upscaleSAR, 42