

Package ‘ecostatscale’

May 6, 2021

Title Statistical Scaling Functions for Ecological Systems

Version 1.0

Description

Implementation of the scaling functions presented in "General statistical scaling laws for stability in ecological systems" by Clark et al in Ecology Letters <DOI:10.1111/ele.13760>. Includes functions for extrapolating variability, resistance, and resilience across spatial and ecological scales, as well as a basic simulation function for producing time series, and a regression routine for generating unbiased parameter estimates. See the main text of the paper for more details.

Imports mvtnorm, stats, graphics, deSolve

License GPL-3

Encoding UTF-8

Date 2021-05-05

RoxygenNote 7.1.1

NeedsCompilation no

Author Adam Clark [aut, cre] (<<https://orcid.org/0000-0002-8843-3278>>)

Maintainer Adam Clark <adam.tclark@gmail.com>

Repository CRAN

Date/Publication 2021-05-06 08:20:09 UTC

R topics documented:

df0	2
df_col	2
df_col_loss	3
ecostatscale	3
res_scale	4
sd_scale	5
symdyn	6
symdynN	7
var_scale	9
xt	10
xt2fun	11
xtN	12

Index**13**

df0 *Simulate deterministic dynamics of a N competing species*

Description

Helper function for symdynN, used to simulate dynamics between disturbance events.

Usage

```
df0(time, state, pars)
```

Arguments

time	time step - ignored, but required for consistency with ode function
state	vector of current states
pars	parameter list, including matrix A with full interaction matrix (and growth rates along the diagonal)

Value

rate of change for each species

df_col *Simulate deterministic dynamics of N patches with dispersal*

Description

Helper function for symdynN, used to simulate dynamics between disturbance events. Note that K is set to 1 for all species.

Usage

```
df_col(time, state, pars)
```

Arguments

time	time step - ignored, but required for consistency with ode function
state	vector of current states
pars	parameter list, including matrix A with full interaction matrix (and growth rates along the diagonal), and a value Ifrac, which is the dispersal rate (D in the main text), and Ksim, which is the carrying capacity

Value

rate of change for each species

df_col_loss	<i>Simulate deterministic dynamics of N patches with dispersal and loss</i>
-------------	---

Description

Helper function for symdynN, used to simulate dynamics between disturbance events. Note that K is set to 1 for all species.

Usage

```
df_col_loss(time, state, pars)
```

Arguments

time	time step - ignored, but required for consistency with ode function
state	vector of current states
pars	parameter list, including matrix A with full interaction matrix (and growth rates along the diagonal), and a value Ifrac, which is the dispersal rate (D in the main text), and Ksim, which is the carrying capacity, and Iloss which is the loss rate.

Value

rate of change for each species

ecostatscale	<i>ecostatscale: Statistical Scaling Functions for Ecological Systems</i>
--------------	---

Description

Implementation of the scaling functions presented in "General statistical scaling laws for stability in ecological systems" by Clark et al. 2021 in Ecology Letters (DOI: 10.1111/ele.13760). Includes functions for extrapolating variability, resistance, and resilience across spatial and ecological scales, as well as a basic simulation function for producing time series, and a regression routine for generating unbiased parameter estimates. See the main text of the paper for more details.

Author

Adam Clark

Source

Clark et al. (2021). General statistical scaling laws for stability in ecological systems. Ecology Letters. DOI:10.1111/ele.13760.

Examples

```

# simulate a time series:
?symdyn # one species one patch
?symdynN # multiple species or patches

# get unbiased parameter estimates:
?xt2fun

#variance scaling function:
?var_scale

# resistance scaling function:
?sd_scale

# resilience scaling function:
?res_scale

```

res_scale	<i>Resilience scaling function</i>
-----------	------------------------------------

Description

Extrapolate resilience observed at the scale of a single spatial or ecological scale b (e.g. a patch or species) to a larger scale, B (e.g. functional group or landscape).

Usage

```

res_scale(
  mvar_b,
  murho_b_abundance,
  mucov_b_abundance = NULL,
  msd_b,
  murho_b_disturbance,
  mucov_b_disturbance = NULL,
  b = 1,
  B,
  lambda
)

```

Arguments

mvar_b	Mean abundance variance observed at scale b
murho_b_abundance	Mean Pearson correlation coefficient of abundance values observed at scale b
mucov_b_abundance	Mean covariance of abundance values observed at scale b . Ignored unless murho_b_abundance is NULL Defaults to NULL.
msd_b	Mean disturbance standard deviation observed at scale b

murho_b_disturbance	Mean Pearson correlation coefficient of disturbances observed at scale b
mucov_b_disturbance	Mean covariance of disturbances observed at scale b. Ignored unless murho_b_abundance is NULL Defaults to NULL.
b	Size of observed scale. Defaults to 1.
B	Larger scale being extrapolated to (e.g. total number of species, or size of patch B relative to b)
lambda	Mean disturbance frequency.

Value

Extrapolated median resilience at scale of M species.

Examples

```
# extrapolate from scale of 1 species to 10 species
res_scale(mvar_b = 0.25, murho_b_abundance = -0.034, msd_b = sqrt(0.1),
          murho_b_disturbance = 0, B = 30, lambda=1)

# plot relationship for groups of 1 to 30 species
plot(1:30, res_scale(mvar_b = 0.25, murho_b_abundance = -0.034, msd_b = sqrt(0.1),
                   murho_b_disturbance = 0, B = 1:30, lambda=1),
     xlab="ecological scale", ylab="resilience, r", type="b")
```

sd_scale

Sigma scaling function

Description

Extrapolate disturbance standard deviation observed at spatial or ecological scale b to a different scale, B (inversely related to resistance). Equivalent to Eq.7b in the main text.

Usage

```
sd_scale(msd_b, murho_b, mucov_b = NULL, b = 1, B)
```

Arguments

msd_b	Mean disturbance standard deviation observed at scale b.
murho_b	Mean Pearson correlation coefficient of disturbances observed at scale b, calculated as mucov_b/mvar_b. If NULL, mucov_b is used instead.
mucov_b	Mean covariane of disturbances observed at scale b. Ignored if mrho_b is not NULL. Defaults to NULL.
b	Size of observed scale. Defaults to 1.
B	Size of desired scale for extrapolation.

Value

Extrapolated disturbance standard deviation at scale B.

Examples

```
#extrapolate from scale of 1 to 10 - e.g. from a 1m2 patch to a 10m2 patch
sd_scale(msd_b = 1, murho_b = 0.5, b = 1, B = 10)
```

symdyn

Simulate time series for a single species in a single patch

Description

Function for simulating dynamics from Eq.1 in the main text.

Usage

```
symdyn(
  r,
  f,
  d,
  d_sd,
  sf,
  tmax,
  stochd = TRUE,
  stocht = TRUE,
  as.matrix = FALSE,
  oscillate_dist = FALSE
)
```

Arguments

r	per-capita growth rate (r in Eq.1)
f	the waiting time (or average waiting time) between disturbance events (equal to 1/lambda in Eq.1)
d	mean size of disturbance function (mu in Eq.1)
d_sd	standard deviation of disturbance function (sigma in Eq.1)
sf	waiting time between sampling events
tmax	the time series length to be simulated
stochd	a logical variable, indicating whether disturbance size should be stochastic - otherwise, all disturbances are of magnitude d - defaults to TRUE
stocht	a logical variable, indicating whether waiting time between disturbance events should be stochastic - otherwise, waiting time is always f - defaults to TRUE
as.matrix	indicates whether results should be returned as matrix (potentially faster for some applications) - defaults to FALSE
oscillate_dist	a logical variable indicating whether the sign of the disturbance should oscillate between positive and negative - ignored if stochd==TRUE - defaults to FALSE

Value

a matrix or data.frame with columns for sampling times, abundances, and number of disturbances for each time interval

Examples

```
# see xt2fun
```

symdynN

Simulate time series for N species or patches

Description

Function for simulating dynamics from Eq.2-3 in the main text.

Usage

```
symdynN(  
  r,  
  amu,  
  asd,  
  f,  
  d,  
  d_sd,  
  d_cov,  
  N,  
  sf,  
  tmax,  
  stochd = TRUE,  
  stocht = TRUE,  
  as.matrix = FALSE,  
  amax = 0,  
  amin = -Inf,  
  Ifrac = NULL,  
  Iloss = NULL,  
  dffun = df0,  
  fallout = FALSE,  
  xstart = NULL,  
  Ksim = 1  
)
```

Arguments

r	per-capita growth rate (r in Eq.2-3)
amu	the mean interaction strength
asd	standard deviation used to generate interaction strengths

f	the waiting time (or average waiting time) between disturbance events (equal to $1/\lambda$ in Eq.2-3)
d	mean size of disturbance function (μ in Eq.2-3)
d_sd	standard deviation of disturbance function (σ in Eq.5)
d_cov	the covariance for generating disturbances
N	number of species or patches
sf	waiting time between sampling events
tmax	the time series length to be simulated
stochd	a logical variable, indicating whether disturbance size should be stochastic - otherwise, all disturbances are of magnitude d - defaults to TRUE
stocht	a logical variable, indicating whether waiting time between disturbance events should be stochastic - otherwise, waiting time is always f - defaults to TRUE
as.matrix	indicates whether results should be returned as matrix (potentially faster for some applications) - defaults to FALSE
amax	the maximum value allowed for interaction coefficients - defaults to zero
amin	the minimum value allowed for interaction coefficients - defaults to -Inf
Ifrac	dispersal rate (D in Eq. 2) - defaults to NULL (i.e. no dispersal)
Iloss	loss rate from dispersal that falls outside of the patch - defaults to NULL
dffun	the function handed to the ODE solver - should be df_col for spatial simulations, and df0 for multi-species simulations - defaults to df0
fullout	a logical, determining whether the full output or just a summary is returned - defaults to fullout
xstart	optional vector of starting abundances - defaults to NULL (i.e. no values)
Ksim	carrying capacities - defaults to 1

Value

a matrix or data.frame with columns for sampling times, abundances, and number of disturbances for each time interval

Examples

```
### Example 1: 10 patches
r<-1 #rate of recovery
d<-(0) #mean size of disturbance (mu in text)
d_sd<-sqrt(0.1) #SD of disturbances (sigma in text)
f<-1 #average time between disturbances (1/lambda in text)
sf<-0.1 #sampling interval
tmax<-120 #maximum time for simulation
d_cov<-d_cov0<-(d_sd)^2/2 #covariance in disturbance size among patches

xtNpatches<-symdynN(r = r, amu=0, asd=0, f=f, d=d,
  d_sd=d_sd, d_cov=d_cov, N=10,
  sf=sf, tmax=tmax, Ifrac=0, dffun = df_col)
```



```

### Example 2: 30 species
r<-1 #rate of recovery
d<-(0) #mean size of disturbance (mu in text)
d_sd<-sqrt(0.1) #SD of disturbances (sigma in text)
f<-1 #average time between disturbances (1/lambda in text)
sf<-0.1 #sampling interval
tmax<-120 #maximum time for simulation
d_cov<-0 #covariance in disturbances among species
amu<-(-r/2) #average interaction coefficient
asd<-0.1 #standard deviation of interaction coefficient

xtNsp<-symdynN(r = r, amu=amu, asd=asd, f=f, d=d,
               d_sd=d_sd, d_cov=d_cov, N=30,
               sf=sf, tmax=tmax)

```

var_scale

Variance scaling function

Description

Extrapolate variance observed at spatial or ecological scale b to a different scale, B . Equivalent to Eq.7a in the main text.

Usage

```
var_scale(mvar_b, murho_b, mucov_b = NULL, b = 1, B)
```

Arguments

mvar_b	Mean variance of abundance values observed at scale b .
murho_b	Mean Pearson correlation coefficient of abundance values observed at scale b , calculated as $\text{mucov}_b/\text{mvar}_b$. If NULL, mucov_b is used instead.
mucov_b	Mean covariance of abundances observed at scale b . Ignored if mrho_b is not NULL. Defaults to NULL.
b	Size of observed scale. Defaults to 1.
B	Size of desired scale for extrapolation.

Value

Extrapolated variance at scale B .

Examples

```

# extrapolate from scale of 1 to 10 - e.g. from a 1m2 patch to a 10m2 patch
var_scale(mvar_b = 1, murho_b = 0.5, b = 1, B = 10)

# example with 100 simulated species
nsp<-100 # number of species
var_b<-1 # species-level abundance variance
cov_b<-(-0.01) # between-specie abundance covariance
# note - if nsp is large, cov_b must be near zero
# this is because, e.g. many variables cannot all be
# simultaneously negatively correlated

# make a covariance matrix based on var_b and cov_b
sigmamat<-diag(nsp)*var_b+(1-diag(nsp))*cov_b
# simulate 1000 observations of 100 species
sim_x<-rmvnorm::rmvnorm(n=1e3, mean = rep(0,100), sigma = sigmamat)

# calculate mean variance, covariance, and correlation from sim_x
cvmat<-cov(sim_x)
mvar_b<-mean(diag(cvmat))
mucov_b<-mean(cvmat[row(cvmat)!=col(cvmat)])
murho_b<-mucov_b/mvar_b

# test function vs. observation
# note - answers match exactly
var(rowSums(sim_x))
var_scale(mvar_b, murho_b = murho_b, b=1, B=100)

```

xt

Simulate deterministic dynamics of a single species

Description

Helper function for symdyn, used to simulate dynamics between disturbance events.

Usage

```
xt(t0, t1, B0, r)
```

Arguments

t0	initial time step
t1	desired time step
B0	initial abundance
r	relative growth rate

Value

predicted value of x at time t1

Examples

```
# see xt2fun
```

```
xt2fun Unbiased stability paramter estimation
```

Description

Function for solving for stability paramter values from observed time series. Equivalent to Eq.5 in the main text.

Usage

```
xt2fun(x0, r, d, d_sd, dt, ndist)
```

Arguments

x0	value of x^2 at time t ($x(t)$ in Eq.5)
r	per-capita growth rate (r in Eq.5)
d	mean size of disturbance function (μ in Eq.5)
d_sd	standard deviation of disturbance function (σ in Eq.5)
dt	time step (i.e. time between x_0 and x_1) - can be a vector of the same length as x_0 , or a number if all time steps are of equal length
ndist	number of disturbances in each time step (equivalent to $p(t+\tau)$ in Eq.5) - must be same length as x_0

Value

predicted value of x^2 at time $t+dt$

Examples

```
# simulate dynamics, with r=1, d=0, and d_sd=0.1
xtout<-symdyn(r=1, f=1, d=0, d_sd=0.1, sf=0.1, tmax=100)

# abundance in current time step
x0<-xtout$state[1:(nrow(xtout)-1)]
# abundance at t+1
x1<-xtout$state[2:nrow(xtout)]

dt<-diff(xtout$time)
ndist<-xtout$disturbed[-1]

# fit model - note square root transform of response variable,
# and log transform of parameter values

mod<-nls(sqrt(x1^2)~sqrt(xt2fun(x0, r=exp(lr), d=0, d_sd=exp(ld_sd), dt, ndist)),
        start=c(lr=log(1), ld_sd=log(0.1)))
exp(coef(mod)) # model estimates
```

`xtN`*Simulate deterministic dynamics*

Description

Helper function for `symdynN`. Simulate an ODE given parameters, starting value, and times.

Usage

```
xtN(t0, t1, B0, odepars, dffun, nsteps = 2)
```

Arguments

<code>t0</code>	initial time step
<code>t1</code>	desired time step
<code>B0</code>	initial abundance
<code>odepars</code>	parameter list
<code>dffun</code>	function for calculating derivatives
<code>nsteps</code>	number of time steps to return - defaults to 2

Value

a matrix of species abundances

Index

df0, [2](#)
df_col, [2](#)
df_col_loss, [3](#)

ecostatscale, [3](#)

res_scale, [4](#)

sd_scale, [5](#)
symdyn, [6](#)
symdynN, [7](#)

var_scale, [9](#)

xt, [10](#)
xt2fun, [11](#)
xtN, [12](#)