Package 'colorednoise'

January 30, 2018

Type Package

Title Simulate Temporally Autocorrelated Population Time Series

Version 0.0.2

Description Temporally autocorrelated populations are correlated in their vital rates (growth, death, etc.) from year to year. It is very common for populations, whether they be bacteria, plants, or humans, to be temporally autocorrelated. This poses a challenge for stochastic population modeling, because a temporally correlated population will behave differently from an uncorrelated one.
This package provides tools for simulating populations with white noise (no temporal autocorrelation), red noise (positive temporal autocorrelation), and blue noise (negative temporal autocorrelation). The algebraic formulation for autocorrelated noise comes from Ruokolainen et al. (2009) <doi:10.1016/j.tree.2009.04.009>. The simulations are based on an assumption of an asexually reproducing population, but it can also be used to simulate females of a sexually reproducing species.

Depends R (>= 3.3.0)

Imports dplyr (>= 0.7.3), purrr (>= 0.2.3), stats (>= 3.3.2), Rcpp (>= 0.12.13)

License GPL-3

Encoding UTF-8

LazyData true

BugReports http://github.com/japilo/colorednoise/issues

RoxygenNote 6.0.1

Suggests ggplot2 (>= 2.2.1), knitr (>= 1.17), rmarkdown (>= 1.6), testthat (>= 1.0.2)

VignetteBuilder knitr

LinkingTo Rcpp

NeedsCompilation yes

Author Julia Pilowsky [aut, cre], Elizabeth Crone [ctb]

Maintainer Julia Pilowsky <jap2178@caa.columbia.edu>

Repository CRAN

Date/Publication 2018-01-30 19:02:33 UTC

R topics documented:

`autocorrelation` *Estimate the Temporal Autocorrelation of a Numeric Vector*

Description

A wrapper for the `acf` function from the stats package that extracts only the temporal autocorrelation at a lag of one timestep (which is the type of temporal autocorrelation that this package simulates).

Usage

```
autocorrelation(x, na.action = na.omit)
```

Arguments

`x`	A numeric vector.
`na.action`	The way the function will handle NAs in the vector. Set to `na.pass` by default.

Value

A single numeric value: the estimate of the temporal autocorrelation with a lag of 1.

Examples

```
rednoise <- raw_noise(timesteps = 50, mu = 0.5, sigma = 0.2, phi = 0.3)
autocorrelation(rednoise)
```

`autocorr_sim` *Simulate Temporally Autocorrelated Populations for Every Combination of Parameters*

Description

Essentially a loop of `timeseries`, this function simulates a population with temporally autocorrelated vital rates for every combination of parameters you specify, with as many replicates as desired. It also estimates the sample mean survival and fertility for each simulated population. Please be advised that this function can be very computationally intensive if you provide many possible parameter values and/or ask for many replicates.

Usage

```
autocorr_sim(timesteps, start, survPhi, fecundPhi, survMean, survSd, fecundMean,
  fecundSd, replicates)
```

Arguments

`timesteps`	The number of timesteps you want to simulate. Individuals are added and killed off every timestep according to the survival and fertility rates. Can be a scalar or a vector of values to loop over.
`start`	The starting population size. Can be a scalar or vector.
`survPhi`	The temporal autocorrelation of survival. 0 is white noise (uncorrelated), positive values are red noise (directly correlated) and negative values are blue noise (inversely correlated). Can be a scalar or a vector.
`fecundPhi`	The temporal autocorrelation of fecundity. As above.
`survMean`	The mean survival from timestep to timestep. Must be a value between 0 (all individuals die) and 1 (all individuals live). Can be a scalar or a vector.
`survSd`	The standard deviation of the survival from timestep to timestep. Must be a value between 0 and 1. Can be a scalar or a vector.
`fecundMean`	The mean fertility: mean offspring produced by each individual per timestep. Can be a scalar or a vector.
`fecundSd`	The standard deviation of the fertility. Can be a scalar or a vector of values.
`replicates`	How many replicates you would like of each possible combination of parameters.

Value

A list of data frames, each with fourteen variables: timestep, newborns (new individuals added this timestep), survivors (individuals alive last year who survived this timestep), population (total individuals alive), growth (the increase or decrease in population size from last year), estimated survival in the timestep, estimated fecundity in the timestep, and the seven parameters used to generate the simulation.

Examples

```
survival_range <- autocorr_sim(timesteps = 30, start = 200, survPhi = 0.3, fecundPhi = 0.1,
                               survMean = c(0.2, 0.3, 0.4, 0.5, 0.6), survSd = 0.5,
                               fecundMean = 1.1, fecundSd = 0.5, replicates = 50)
head(survival_range[[1]])
```

`raw_estim` *Estimate Mean, SD, and Autocorrelation of Sample Noise.*

Description

This function estimates the temporal autocorrelation of a vector of random numbers, as well as the sample mean and standard deviation. Try feeding it the output of `raw_noise`.

Usage

```
raw_estim(noise)
```

Arguments

`noise`	The vector of random numbers.

Value

A labeled vector with the sample mean, sample SD, and sample autocorrelation.

Examples

```
rednoise <- raw_noise(timesteps = 30, mu = 0.5, sigma = 0.2, phi = 0.3)
raw_estim(rednoise)
```

`raw_estim_loop` *Simulation and Estimation of Colored Noise*

Description

This function simulates generates sets of temporally autocorrelated random numbers for every possible combination of parameter values you specify, then estimates the mean, standard deviation, and autocorrelation of each set of random numbers so generated. Internally, the function does the same thing as `raw_noise_loop`, but instead of outputting the random numbers, it outputs measures of each set of random numbers.

Usage

```
raw_estim_loop(timesteps, mu, sigma, phi, replicates)
```

Arguments

timesteps	How many timesteps you want in each set. Can be scalar or vector.
mu	The mean of the temporally autocorrelated random numbers. Can be scalar or vector.
sigma	The standard deviation of the temporally autocorrelated random numbers. Can be scalar or vector.
phi	The temporal autocorrelation. 0 is white noise (uncorrelated), positive values are red noise (correlated) and negative values are blue noise (inversely correlated). Can be scalar or vector.
replicates	How many replicates you would like of each possible combination of parameters.

Value

A data frame with one row for each set of random numbers. The variables are mean, SD, autocorrelation, and the four parameters used to generate the random numbers.

Examples

```
estimates <- raw_estim_loop(timesteps=c(5:10), mu=c(0.2, 0.5), sigma=c(0.2, 0.5),
                            phi=c(0, 0.1), replicates=10)
head(estimates)
```

---

raw_noise *Generate Autocorrelated Noise*

Description

This function generates temporally autocorrelated random numbers with a mean, standard deviation, and autocorrelation you specify.

Usage

```
raw_noise(timesteps, mu, sigma, phi)
```

Arguments

timesteps	The number of temporally autocorrelated random numbers (one per timestep) you want.
mu	The mean of the temporally autocorrelated random numbers.
sigma	The standard deviation of the temporally autocorrelated random numbers.
phi	The temporal autocorrelation. 0 is white noise (uncorrelated), positive values are red noise (directly correlated) and negative values are blue noise (inversely correlated).

Value

A vector of temporally autocorrelated random numbers.

Examples

```
rednoise <- raw_noise(timesteps = 30, mu = 0.5, sigma = 0.2, phi = 0.3)
rednoise
```

`raw_noise_loop` *Generate Autocorrelated Noise for Every Combination of the Given Parameters.*

Description

This function generates sets of temporally autocorrelated random numbers for every possible combination of parameter values you specify. Essentially a loop of `raw_noise` that outputs a list. All parameters can be given as single values or vectors of values.

Usage

```
raw_noise_loop(timesteps, mu, sigma, phi, replicates)
```

Arguments

`timesteps`	How many timesteps you want in each set. Can be scalar or vector.
`mu`	The mean of the temporally autocorrelated random numbers. Can be scalar or vector.
`sigma`	The standard deviation of the temporally autocorrelated random numbers. Can be scalar or vector.
`phi`	The temporal autocorrelation. 0 is white noise (uncorrelated), positive values are red noise (correlated) and negative values are blue noise (inversely correlated). Can be scalar or vector.
`replicates`	How many replicates you would like of each possible combination of parameters.

Value

A list of vectors of temporally autocorrelated random numbers. Each element in the list is named with the combination of parameters that generated it.

Examples

```
loop <- raw_noise_loop(timesteps=c(5:10), mu=c(0.2, 0.5), sigma=c(0.2, 0.5),
                       phi=c(0, 0.1), replicates=10)
loop[[1]]
```

`timeseries` *Simulated Time Series of an Unstructured Temporally Autocorrelated Population*

Description

This function simulates an unstructured population with temporally autocorrelated vital rates (survival and fertility). In other words, this function will show you the dynamics over time of a population whose survival and fertility is stochastic, but also correlated to the survival and fertility in the previous year, respectively. The assumptions of the simulation are that the population is asexually reproducing or female-only, survival and fertility are the same at all ages / stages, and that individuals continue to be reproductively capable until they die.

Usage

```
timeseries(start, timesteps, survPhi, fecundPhi, survMean, survSd, fecundMean,
  fecundSd)
```

Arguments

`start`	The starting population size.
`timesteps`	The number of timesteps you want to simulate. Individuals are added and killed off every timestep according to the survival and fertility rates. In ecological applications, timesteps are usually years, but theoretically they can be any length of time.
`survPhi`	The temporal autocorrelation of survival. 0 is white noise (uncorrelated), positive values are red noise (directly correlated) and negative values are blue noise (inversely correlated).
`fecundPhi`	The temporal autocorrelation of fecundity. As above.
`survMean`	The mean survival from timestep to timestep. Must be a value between 0 (all individuals die) and 1 (all individuals live).
`survSd`	The standard deviation of the survival from timestep to timestep. Must be a value between 0 and 1.
`fecundMean`	The mean fertility: mean offspring produced by each individual per timestep.
`fecundSd`	The standard deviation of the fertility.

Details

Be advised that not all combinations of values will work. If you set survival and fertility unrealistically high, the population size will tend toward infinity and the simulation will fail because the numbers are too large to handle. Use your common sense as a demographer / population biologist.

Value

A data frame with four variables: timestep, population (total individuals alive at the start of the timestep), newborns (new individuals born this timestep), and survivors (individuals who survive this timestep).

Examples

```
series1 <- timeseries(start = 20, timesteps = 10, survPhi = 0.7, fecundPhi = -0.1, survMean = 0.6,
survSd = 0.52, fecundMean = 1.2, fecundSd = 0.7)
head(series1)
```

Index