Package 'AHMbook'

June 9, 2020

Type Package

Title Functions and Data for the Book ``Applied Hierarchical Modeling
in Ecology'' Vols 1 and 2

Version 0.2.0

Date 2020-06-08

Depends R (>= 2.10)

Imports grDevices, graphics, methods, stats, utils, plotrix, raster,
sp, RandomFields, coda, unmarked (>= 0.12.2), mvtnorm, spdep

Suggests

BugReports <https://github.com/mikemeredith/AHMbook/issues>

Description Provides functions and data sets to accompany the two volume publication ``Applied Hi-
erarchical Modeling in Ecology: Analysis of distribution, abundance and species rich-
ness in R and BUGS'' by Marc Kéry and Andy Royle: volume 1 (2016, ISBN: 978-0-12-801378-
6) and volume 2 (2020, ISBN: 978-0-12-809585-0), <https://www.mbr-
pwrc.usgs.gov/pubanalysis/keryroylebook>.

License GPL (>= 3)

Copyright file COPYRIGHTS

NeedsCompilation no

URL <https://www.mbr-pwrc.usgs.gov/pubanalysis/keryroylebook/>,
<https://sites.google.com/site/appliedhierarchicalmodeling/home>

Encoding UTF-8

Language en-US

Author Marc Kéry [aut],
Andy Royle [aut],
Mike Meredith [aut, cre],
Urs Breitenmoser [dtc] (EurasianLynx),
Richard Chandler [ctb],
Bob Dorazio [ctb],
Evan Grant [dtc] (duskySalamanders),
Jon-Andren Henden [dtc] (Finnmark),

Roland Kays [dtc] (MesoCarnivores),
David King [dtc] (cswa),
Xavier Lambin [dtc] (waterVoles),
Jeremy Mizel [ctb, dtc] (treeSparrow),
Anja Molinari-Jobin [dtc] (EurasianLynx),
Réne-Jean Monneret [dtc] (FrenchPeregrines),
Arielle Parsons [dtc] (MesoCarnivores),
René Ruffinoni [dtc] (FrenchPeregrines),
Michael Schaub [ctb],
Rahel Solmann [ctb],
Nicolas Strebel [dtc],
Chris Sutherland [ctb, dtc] (waterVoles),
Mathias Tobler [ctb],
Fridolin Zimmermann [dtc] (EurasianLynx),
Aargau Biodiversity Monitoring Program (LANAG) [dtc],
British Ornithological Trust [dtc] (willowWarbler, UKmarbledWhite,
<https://www.bto.org/>),
Butterfly Conservation [dtc, cph] (UKmarbledWhite,
<https://butterfly-conservation.org>),
Centre for Ecology & Hydrology [dtc, cph] (UKmarbledWhite,
<https://www.ceh.ac.uk>),
Dutch Centre for Field Ornithology (Sovon) [dtc] (wagtail,
<https://www.sovon.nl>),
eMammal [dtc] (MesoCarnivores, <https://emammal.si.edu/>),
Groupe Pèlerin Jura [dtc] (FrenchPeregrines),
Hubbard Brook Ecosystem Study [dtc] (HubbardBrook,
<http://data.hubbardbrook.org/data/dataset.php?id=178>),
Joint Nature Conservation Committee (JNCC) [dtc, cph] (UKmarbledWhite,
<https://jncc.gov.uk/>),
North Carolina Museum of Natural Sciences [dtc] (mesoCarnivores),
North Carolina State University [dtc] (mesoCarnivores),
Progetto Lince Italia [dtc] (EurasianLynx,
<https://www.progettolinceitalia.it>),
Swiss Federal Statistical Office [dtc, cph] (BerneseOberland,
<http://www.bfs.admin.ch>),
Swiss Foundation KORA Carnivore Ecology and Wildlife Management (SCALP
project) [dtc] (EurasianLynx, <https://www.kora.ch>),
Swiss Ornithological Institute [dtc] (<https://www.vogelwarte.ch>),
Centre for Ecology & Hydrology [dtc, cph] (UKmarbledWhite,
<https://www.ceh.ac.uk>),
UK Butterfly Monitoring Scheme (UKBMS) [dtc, cph] (UKmarbledWhite,
<https://www.ceh.ac.uk/our-science/projects/uk-butterfly-monitoring-scheme>)

Maintainer Mike Meredith <mike@mmeredith.net>

Repository CRAN

Date/Publication 2020-06-09 13:00:19 UTC

R **topics documented:**

AHMbook-package   *Functions and data for the Book "Applied Hierarchical Modeling in Ecology" Volumes 1 and 2*

Description

Provides functions and data sets needed to run the code given in the two-volume publication, *Applied Hierarchical Modeling in Ecology: analysis of distribution, abundance and species richness in R and BUGS* by Marc Kéry and Andy Royle, Academic Press (Vol 1, 2016; Vol 2, 2020).

Details

The functions are listed by chapter below, where AHM1 refers to volume 1 and AHM2 to volume 2.

AHM1 - Chapter 1

sim.fn Simulate a Poisson point process and illustrate the fundamental relationships between intensity, abundance and occurrence (AHM1 - section 1.1)

AHM1 - Chapter 4

data.fn Simulate count data that are replicated in space and in time according to the binomial N-mixture model of Royle (2004) (this is for much simpler cases than is possible with function simNmix in Chapter 6 below) (AHM1 - 4.3)

AHM1 - Chapter 6

ppc.plot Plot results from posterior predictive checks in section AHM1 - 6.8, for a fitted binomial N-mixture model object with JAGS

simNmix Simulate count data and individual detection histories for binomial and multinomial mixture models respectively under a wide range of conditions (AHM1 - 6.9.3)

plot_Nmix_resi Do diagnostic plots for one binomial N-mixture model fitted with all three mixture distributions currently available in unmarked: Poisson, negative binomial and zero-inflated Poisson (AHM1 - 6.9.3)

map.Nmix.resi Produce a map of the residuals from a binomial N-mixture model (see Section AHM1 - 6.9.3)

simpleNmix Simulate count data under a very simple version of the binomial mixture model, with time for space substitution (AHM1 - 6.12)

playRN Play Royle-Nichols (RN) model: generate count data under the binomial N-mixture model of Royle (2004), then 'degrade' the data to detection/nondetection and fit the RN model using unmarked and estimate site-specific abundance (AHM1 - 6.13.1)

AHM1 - Chapter 7

instRemPiFun, crPiFun, crPiFun.Mb, MhPiFun Define the relationship between the multinomial cell probabilities and the underlying detection probability parameters (i.e., a pi function) in various designs (AHM1 - 7.8 and AHM2 - Chapter 2)

AHM1 - Chapter 8

sim.ldata Simulate data under a non-hierarchical line transect distance sampling model (AHM1 - 8.2.3)

sim.pdata Simulate data under a non-hierarchical point transect (= point count) distance sampling model (AHM1 - 8.2.5.1)

simHDS Simulate data under a hierarchical distance sampling protocol (line or point) (AHM1 - 8.5.1)

AHM1 - Chapter 9

simHDSg Simulate data under a hierarchical distance sampling (HDS) protocol with groups (AHM1 - 9.2.1)

simHDStr Simulate data under a time-removal/distance sampling design (AHM1 - 9.3.2)

simHDSopen Simulate open hierarchical distance sampling data (AHM1 - 9.5.4)

issj.sim Simulate data under the open distance sampling protocol for the Island Scrub Jays (AHM1 - 9.7.1)

sim.spatialDS Simulate data under a basic spatial distance sampling model (AHM1 - 9.8.3)

sim.spatialHDS Simulate data under a spatial hierarchical distance sampling model (AHM1 - 9.8.5)

AHM1 - Chapter 10

simOcc Simulate detection/nondetection data under static occupancy models under a wide range of conditions (AHM1 - 10.5)

sim3Occ Simulate detection/nondetection data under a static 3-level occupancy model (AHM1 - 10.10)

simOccttd Simulate 'timing data' under a static time-to-detection occupancy design (AHM1 - 10.12.1)

wigglyOcc Simulate detection/nondetection data under a static occupancy model with really wiggly covariate relationships in occupancy and detection probability (AHM1 - 10.14)

spline.prep Prepare input for BUGS model when fitting a spline for a covariate (AHM1 - 10.14)

AHM1 - Chapter 11

simComm Simulate detection/nondetection or count data under a community occupancy or abundance model respectively (AHM1 - 11.2)

AHM2 - Chapter 1

simNpC Simulate data on abundance (N), detection probability (p) and resulting counts (C) under a counting process with imperfect detection (AHM2 - 1.2)

simPOP Simulate count data under a demographic state-space, or Dial-Madsen, model (AHM2 - 1.7.1)

simPH Simulate count data with phenological curves within a year (AHM2 - 1.8.1)

graphSSM Plot trajectories of counts and latent abundance from a fitted Gaussian state-space model (AHM2 - 1.6.1)

AHM2 - Chapter 2

simDM0 Simulate count data from a Dail-Madsen model, no covariates (AHM2 - 2.5.1)

simDM Simulate count data from a Dail-Madsen model, with covariates (AHM2 - 2.5.5)

simMultMix Simulate "removal" data from a multinomial-mixture model (AHM2 - 2.7.1)

AHM2 - Chapter 3

simCJS Simulate individual capture history data under a Cormack-Jolly-Seber (CJS) survival model (AHM2 - 3.2.2)

ch2marray Convert capture history data to the m-array aggregation (AHM2 - 3.4.1)

AHM2 - Chapter 4

simDynocc Simulate detection/nondetection data under a dynamic occupancy model under a wide range of conditions (AHM2 - 4.4)

simDemoDynocc Simulate detection/nondetection data under a demographic dynamic occupancy model (AHM2 - 4.12)

AHM2 - Chapter 5

simDCM Simulate detection/nondetection data under a general dynamic community model (site-occupancy variant) (AHM2 - 5.2)

AHM2 - Chapter 7

valid_data Partial validation of simulated data with false positives (AHM2 - 7.6.2)

modSelFP A stop-gap function for modSel for fitted false positive occupancy models (AHM2 - 7.2.2)

AHM2 - Chapter 8

getLVcorrMat Compute the correlation matrix from an analysis of a latent variable occupancy or binomial N-mixture model (AHM2 - 8.4.2)

AHM2 - Chapter 9

simDynoccSpatial Simulate detection/nondetection data under a dynamic occupancy model with spatial covariate and spatial autocorrelation (AHM2 - 9.6.1.1)

simExpCorrRF Simulate data from a Gaussian random field with negative exponential correlation function (AHM2 - 9.2)

simOccSpatial Simulate detection/nondetection data under a spatial, static occupancy model for a real landscape in the Bernese Oberland (Swiss Alps) (AHM2 - 9.2)

simNmixSpatial Simulate counts under a spatial, static binomial N-mixture model for a real landscape in the Bernese Oberland (Swiss Alps) (AHM2 - 9.2)

AHM2 - Chapter 10

simPPe Simulate a spatial point pattern in a heterogeneous landscape and show aggregation to abundance and occurrence ('e' for educational version) (AHM2 - 10.2)

simDataDK Simulate data for an integrated species distribution model (SDM) of Dorazio-Koshkina (AHM2 - 10.6.1)

AHM2 - Chapter 11

DATA SETS

UKmarbledWhite Data from the UK Butterfly Monitoring Scheme (UKBMS) for Marbled White butterfly (AHM2 - 1.8.2)

wagtail Distance sampling data for Yellow Wagtails in The Netherlands (AHM1 - 9.5.3)

waterVoles Detection/nondetection data for the Mighty Water Vole of Scotland (AHM2 - 7.2.2)

wigglyLine Coordinates for a wiggly transect line (AHM2 - 11.9)

willowWarbler Capture-history (survival) data for Willow Warblers in Britain (AHM2 - 3.4.1)

UTILITY FUNCTIONS

zinit Generate starting values for fitting survival models (introduced in AHM2 - 3.2.3).

standardize Standardize covariates to mean 0, SD 1.

fitstats, fitstats2 Calculate fit-statistics used in parboot GOF tests throughout the book (eg, Sections AHM1 - 7.5.4, AHM1 - 7.9.3, AHM2 - 2.3.3)

e2dist Compute a matrix of Euclidean distances

image_scale Draw scale for image (introduced in chapter AHM1 - 9.8.3)

bigCrossCorr Report cross-correlations above a given threshold

Color_Ramps Color ramps for use with image or raster plots

Author(s)

Marc Kéry, Andy Royle, Mike Meredith

| BerneseOberland | *Landscape data for the Bernese Oberland around Interlaken, in the Alps of central Switzerland* |

Description

Spatially-referenced data on elevation, forest cover, and water cover at a 1km-sq resolution for a 50km x 50km region in Switzerland, the Bernese Oberland. This is a small subset of the Switzerland data in the unmarked package. See the Examples for a plot of the location.

Usage

```
data(BerneseOberland)
```

Format

A data frame with 2500 observations on the following 5 variables.

x Easting (m)

y Northing (m)

elevation a numeric vector (m)

forest a numeric vector (percent cover)

water a numeric vector (percent cover)

Details

Forest and water coverage (in percent area) was computed using the 1992-97 landcover dataset of the Swiss Federal Statistical Office (http://www.bfs.admin.ch). Median elevation (in meters) was computed using a median aggregation of the digital elevation model of the Swiss Federal Statistical Office.

x and y are the coordinates of the center of each 1km2 pixel.

The coordinate reference system intentionally not specified.

These data can only be used for non-profit projects. Otherwise, written permission must be obtained from the Swiss Federal Statistical Office

Source

Swiss Federal Statistical Office (http://www.bfs.admin.ch)

Examples

```
data(BerneseOberland)
str(BerneseOberland)


library(raster)
data(Switzerland, package="unmarked")
r1 <- with(Switzerland, rasterFromXYZ(data.frame(x = x, y = y, z = elevation)))
mapPalette1 <- colorRampPalette(c("grey", "yellow", "orange", "red"))
plot(r1, col = mapPalette1(100), axes = FALSE, box = FALSE, zlim = c(0, 4500),
  main = "Location of Bernese Oberland data set")
with(BerneseOberland, rect(min(x), min(y), max(x), max(y), lwd=2))
```

---

bigCrossCorr   *Cross correlations for MCMC output*

Description

A wrapper for `coda::crosscorr`, which calculates cross-correlations between posterior draws of parameters in Markov chain Monte Carlo output. When the output has hundreds of parameters, the matrix produced by `crosscorr` is unwieldy, and `bigCrossCorr` extracts only those greater than a given threshold.

Usage

```
bigCrossCorr(x, big = 0.6, digits = 3)
```

Arguments

x	an `mcmc` or `mcmc.list` object.
big	only values below -big or above +big will be returned
digits	the number of decimal places to return

Value

A data frame with 2 columns for the names of parameters and a 3rd column with the cross-correlation.

Author(s)

Mike Meredith

See Also

[crosscorr](#) in package **coda**.

---

ch2marray	*Converts capture-histories to an m-array*

Description

Converts capture-histories to an m-array for use in a Cormack-Jolly-Seber (CJS) model.

Usage

```
ch2marray(CH)
```

Arguments

CH	An individuals x time matrix of capture records, 1 if captured, 0 otherwise, no missing values.

Value

An m-array, a (years-1) x years matrix, where element [i, j] contains the number of individuals released in year i and recaptured in year j+1 (by definition no recaptures can occur in year 1). The last column contains the number of individuals released in year i and never recaptured.

Author(s)

Marc Kéry & Andy Royle, modified from code in Kéry and Schaub (2012).

References

Kéry and Schaub (2012) *Bayesian population analysis using WinBUGS - a hierarchical perspective*,
Academic Press.

Kéry, M. & Royle, J.A. (2020) *Applied Hierarchical Modeling in Ecology* AHM2 - 3.4.1

Examples

```
data(willowWarbler)
ch <- as.matrix(willowWarbler$birds[ , 1:11]) # extract capture-histories.
dim(ch)
ch2marray(ch)
```

Color Ramps   *Color Ramps*

Description

These create vectors of n contiguous colors for use in image and raster plots. They are similar to
heat.colors or terrain.colors but with extra arguments to give greater control of colors. Currently
included ramps are:

* sequential yellow-orange-red

* sequential light to dark green

* sequential light to dark gray

* diverging blue-yellow-red

* diverging green-brown

Sequential ramps go from light to dark, diverging ramps from dark through light to dark. All
these ramps are color-blind safe and based on color palettes from http://colorbrewer2.org/ by
Cynthia Brewer and colleagues.

Usage

```
rampYOR(n=5, range=1:9, bias=1, ...)
rampGreens(n=5, range=1:9, bias=1, ...)
rampGreys(n=5, range=1:9, bias=1, ...)
rampBYR(n=5, range=1:11, bias=1, ...)
rampGBr(n=5, range=1:11, bias=1, ...)
```

Arguments

n	the number of colors to be in the palette.
range	the range of colors from the underlying palette to include: each ramp is based on a palette of 9 (if sequential) or 11 (if diverging) colors; range allows the extreme colors to be excluded.
bias	a positive number: values >1 give more widely spaced colors at the high end.
...	additional arguments passed to colorRampPalette.

Value

Return a vector of color values in hexadecimal format.

Author(s)

Mike Meredith.

Examples

```
# uses the built-in volcano data set
require(grDevices) # for colours
require(graphics)
image(volcano, col=rampYOR(225), main="Sequential yellow-orange-red")
image(volcano, col=rampBYR(225), main="Divergent blue-yellow-red")
image(volcano, col=rampGBr(225), main="Divergent green-brown")
points(runif(500), runif(500), pch=16, cex=0.7) # add points
title(sub="Points are hard to see on the darkest colors", line=2)
image(volcano, col=rampGBr(225, range=2:10), main="Green-brown without darkest colors")
points(runif(500), runif(500), pch=16, cex=0.7)

# Try with a raster
require(raster)
r <- raster(system.file("external/test.grd", package="raster"))
plot(r, main="Default colors") # default is rev(terrain.colors(225))
plot(r, col=rampGBr(225), main="Divergent green-brown")
plot(r, col=rampGBr(225, bias=3), main="Divergent green-brown, bias=3")
plot(r, col=rampGBr(225, bias=0.5), main="Divergent green-brown, bias=0.5")
```

---

crestedTit	*Count data from the Swiss Breeding Bird Survey MHB for crested tits from 1999 to 2015*

Description

The Swiss breeding bird survey ("Monitoring Häufige Brutvögel" MHB) has monitored the populations of 150 common species since 1999. The MHB sample consists of 267 1-km squares that are laid out as a grid across Switzerland. Fieldwork is conducted by about 200 skilled birdwatchers annually, most of them volunteers. Avian populations are monitored using a simplified territory mapping protocol (also called spot-mapping West of the Atlantic), where each square is surveyed up to three times during the breeding season (only twice above the tree line around 2000 m elevation). Surveys are conducted along a transect that does not change over the years and is 4-6 kms in length on average.

The data frame `crestedTit` has the data for Crested Tits *Parus cristatus* from 1999 to 2015. There are some missing values: see Details.

Usage

```
data("crestedTit")
```

Format

crestedTit is a data frame with 267 rows and 131 columns:

coordx, coordy the x and y coordinates of the center of the quadrat; the coordinate reference system intentionally not specified.

elev the mean elevation of the quadrat, in meters.

forest percentage forest cover

nsurveys the number of replicate surveys *planned* in the quadrat; above the tree-line 2, otherwise 3.

1999 to 2016 total number of territories observed for the year.

date991 to date163 Julian date of the survey, 1 = 1st January; the first 2 digits in the column name indicate the year and the 3rd digit the survey.

dur991 to dur163 duration of the survey, in minutes; the first 2 digits in the column name indicate the year and the 3rd digit the survey.

Details

Missing values in the date array indicate that the corresponding survey was not carried out.

On 26 occasions when surveys were carried out, the duration was not recorded, resulting in additional NAs in the dur array.

A new method for recording breeding territories was introduced in 2004, but the old protocol was in use at some sites until 2013. Surveys with the old protocol have the counts shown as NA in the count array.

See also Chapter 6 in Kéry & Royle (2016) for further description of the survey and the data it produces.

Source

Swiss Ornithological Institute

References

Kéry & Royle (2016) *Applied Hierarchical Modeling in Ecology* AHM1 - 6.

Kéry & Royle (2020) *Applied Hierarchical Modeling in Ecology* AHM2 - 1.

Examples

```
data(crestedTit)
str(crestedTit)

# Prepare objects needed for section AHM2 - 1.3 and following
C <- as.matrix(crestedTit[, 6:23]) # matrix of counts
year <- 1999:2016
datetmp <- as.matrix(crestedTit[, 24:77])  # matrix of Julian dates
datefull <- array(datetmp, c(nrow(datetmp), 3, ncol(datetmp)/3))
    # site x rep x year array
durtmp <- as.matrix(crestedTit[, 78:131])  # matrix of survey durations
```

```
durfull <- array(durtmp, c(nrow(durtmp), 3, ncol(durtmp)/3))
    # site x rep x year array

# Get mean date and duration of survey for each site and year
mdate <- apply(datefull, c(1,3), mean, na.rm=TRUE)
mdate[is.nan(mdate)] <- NA  # Replace NaN with NA
mdur <- apply(durfull, c(1,3), mean, na.rm=TRUE)
mdur[is.nan(mdur)] <- NA

# For Sec 1.5, we need standardised covariates with missing values imputed
elev.sc <- as.numeric(scale(crestedTit$elev))    # site elevation
forest.sc <- as.numeric(scale(crestedTit$forest)) # site forest cover

mean.date <- mean(mdate, na.rm=TRUE)          # mean survey date per site
sd.date <- sd(mdate, na.rm=TRUE)
mdate.sc <- (mdate - mean.date) / sd.date
mdate.sc[is.na(mdate.sc)] <- 0                # mean imputation

mean.dur <- mean(mdur, na.rm=TRUE)            # mean survey duration per site
sd.dur <- sd(mdur, na.rm=TRUE)
mdur.sc <- (mdur - mean.dur) / sd.dur
mdur.sc[is.na(mdur.sc)] <- 0                  # mean imputation

# For sections 1.6 and 1.7, we remove sites with no crested tits recorded,
#   or recorded in only one year:
nzero <- apply(C == 0, 1, sum, na.rm=TRUE) # number of zero years per site
sel <- nzero <= 1  # TRUE if site has 2 or more years of non-zero data
```

---

crossbillAHM   *Data from the Swiss Breeding Bird Survey MHB for European Cross-*
                          bill from 2001 to 2012

Description

The Swiss breeding bird survey ("Monitoring Häufige Brutvögel" MHB) has monitored the populations of 150 common species since 1999. The MHB sample consists of 267 1-km squares that are laid out as a grid across Switzerland. Fieldwork is conducted by about 200 skilled birdwatchers annually, most of them volunteers. Avian populations are monitored using a simplified territory mapping protocol (also called spot-mapping West of the Atlantic), where each square is surveyed up to three times during the breeding season (only twice above the tree line around 2000 m elevation). Surveys are conducted along a transect that does not change over the years and is 4-6 kms in length on average.

The data frame crossbillAHM has the data for European Crossbill (*Loxia curvirostra*) from 2001 to 2012.

A variant of this data set for 1999 to 2007 only is included in package **unmarked**.

Usage

```
data("crossbillAHM")
```

Format

crossbillAHM is a data frame with 267 rows and 77 columns:

coordx, coordy the x and y coordinates of the center of the quadrat; the coordinate reference system intentionally not specified.

elev the mean elevation of the quadrat, m.

forest percentage forest cover

nsurveys the number of replicate surveys *planned* in the quadrat; above the tree-line 2, otherwise 3.

det011 to det123 1 if the species was detected in the quadrat, 0 otherwise; NA if the corresponding survey was not carried out; the first 2 digits indicate the year and the 3rd digit the survey.

date011 to date123 Julian date of the survey, 1 = 1st January; the first 2 digits indicate the year and the 3rd digit the survey; NA if the corresponding survey was not carried out.

Source

Swiss Ornithological Institute

References

Kéry, M & Royle, J.A. (2020) *Applied Hierarchical Modeling in Ecology* AHM2 - 4.

Examples

```
data(crossbillAHM)
str(crossbillAHM)

# Extract data as site x survey x year arrays
ytmp <- as.matrix(crossbillAHM[, 6:41]) # matrix of detections
y <- array(ytmp, c(nrow(ytmp), 3, ncol(ytmp)/3))
datetmp <- as.matrix(crossbillAHM[, 42:77])  # matrix of Julian dates
date <- array(datetmp, c(nrow(datetmp), 3, ncol(datetmp)/3))
```

---

cswa   *Point count and spot-mapping data for Chestnut-sided Warbler*

Description

Chestnut-sided Warblers (*Setophaga pensylvanica*)in 38 "wildlife openings" in the White Mountain National Forest were surveyed using point counts and spot-mapping during June and July 2004 as part of Richard Chandler's MS thesis. Counts are for birds detected by song at each point. Two of the larger openings had more than one point, giving 43 points in all. Surveys lasted 10 mins and were divided into intervals of (0–2], (2–5], and (5–10] min. Each point was surveyed 3 times, except for one point surveyed only twice. The spot-mapping data come from intensive surveys of each patch recording individual sightings and nest locations, and using counter-singing to separate adjacent territories.

Usage

```
data("cswa")
```

Format

cwsa is a list with 3 components:

counts   a points x intervals x occasions array of counts; NAs correspond to the missing survey.

covs   a data frame with rows for 43 points and the following columns:

- time1 : time of start of first survey, hr; NA for the missing survey.
- time2 : same for second survey.
- time3 : same for third survey.
- date1 : date of first survey, NA for the missing survey.
- date2 : date of second survey.
- date3 : date of third survey.
- obs1 : observer ID for first survey.
- obs2 : observer ID for second survey.
- obs3 : observer ID for third survey.
- patchArea : patch area, ha.
- plotArea : size of point count area, ha. Most were 50-m radius but some were smaller because they overlapped adjacent forests which weren't the focus of the study.
- woodHt : woody vegetation height, m.
- woodCov : proportion covered by woody vegetation.

spotMaps   a data frame with rows for 38 patches and the following columns:

- CSWA : estimated population of chestnut-sided warblers in each patch obtained from spot-mapping.
- parea : patch area, ha.

Source

David King and colleagues.

References

Chandler, R.B., King, D.I., & Chandler, C.C. (2009) Effects of management regime on the abundance and nest survival of shrubland birds in wildlife openings in northern New England, USA. *Forest Ecology and Management*, 258, 1669-1676.

Chandler, R.B., Royle, J.A., & King, D.I. (2011) Inference about density and temporary emigration in unmarked populations. *Ecology*, 92, 1429-1435.

Kéry, M. & Royle, J.A. (2020) *Applied Hierarchical Modeling in Ecology* AHM2 - 2.

Examples

```
data(cswa)
str(cswa)

# Replace the NAs in the covariates with the mean from other surveys:
covs <- cswa$covs
covs$time1[is.na(covs$time1)] <- mean(covs$time1, na.rm=TRUE)
covs$date1[is.na(covs$date1)] <- mean(covs$date1, na.rm=TRUE)
# Now use covs instead of cswa$covs
```

---

data.fn   *Simulate count data under a binomial N-mixture model*

Description

Function to simulate point counts replicated at M sites during J occasions. Population closure
is assumed for each site. Expected abundance may be affected by elevation (elev), forest cover
(forest) and their interaction. Expected detection probability may be affected by elevation, wind
speed (wind) and their interaction. Used in AHM1 to illustrate how a data set under some specific
model of interest can be simulated.

Usage

```
data.fn(M = 267, J = 3, mean.lambda = 2, beta1 = -2, beta2 = 2, beta3 = 1,
    mean.detection = 0.3, alpha1 = 1, alpha2 = -3, alpha3 = 0, show.plot = TRUE)
```

Arguments

M	Number of spatial replicates (sites)
J	Number of temporal replicates (occasions)
mean.lambda	Mean abundance at value 0 of abundance covariates
beta1	Main effect of elevation on abundance
beta2	Main effect of forest cover on abundance
beta3	Interaction effect on abundance of elevation and forest cover
mean.detection	Mean detection prob. at value 0 of detection covariates
alpha1	Main effect of elevation on detection probability
alpha2	Main effect of wind speed on detection probability
alpha3	Interaction effect on detection of elevation and wind speed
show.plot	if TRUE, plots of the data will be displayed; set to FALSE if you are running many simulations

Value

A list with the input arguments and the following additional elements:

elev	Scaled elevation (a vector of length M)
forest	Scaled forest cover (a vector of length M)
wind	Scaled wind speed (an M x J matrix)
lambda	Expected number of individuals at each site (a vector of length M)
N	Realized number of individuals at each site (a vector of length M)
p	Probability of detection for each survey (an M x J matrix)
C	The number of detections for each survey (an M x J matrix)
Ntotal	Total abundance, sum(N)
psi.true	True occupancy in sample
summaxC	Sum of max counts (all sites)
psi.obs	Observed occupancy in sample

Note

The colors used for points in some of the plots indicate different temporal replicates.

Author(s)

Marc Kéry & Andy Royle

References

Kéry, M. & Royle, J.A. (2016) *Applied Hierarchical Modeling in Ecology* AHM1 - 4.3.

Examples

```
# Generate a simulated data set with default arguments and look at the structure:
tmp <- data.fn()
str(tmp)

str(data.fn(J = 2))            # Only 2 surveys
str(data.fn(J = 1))            # No temporal replicate
str(data.fn(M = 1, J = 100))   # No spatial replicates, but 100 counts
str(data.fn(beta3 = 1))        # With interaction elev-wind on p
str(data.fn(M = 267, J = 3, mean.lambda = 2, beta1 = -2, beta2 = 2, beta3 = 1,
  mean.detection = 1))         # No obs. process (i.e., p = 1, perfect detection)
str(data.fn(mean.lambda = 50))   # Really common species
str(data.fn(mean.lambda = 0.05)) # Really rare species
```

| dragonflies | *Fictional data for dragonflies* |

Description

A toy data set with fictional data for 9 individuals of the dragonfly *Onychogomphus uncatus* from 3 populations in the Spanish Pyrenees.

Usage

```
data(dragonflies)
```

Format

The format is seven vectors of length 9:

pop a factor indicating which population the individual was drawn from.

sex a factor indicating the sex of each individual.

wing wingspan.

body body length.

mites number of mites (ectoparasites) counted.

color proportion of body yellow as opposed to black.

damage the number of wings (out of 4) damaged.

Source

Fictitious data.

References

Kéry, M. & Royle, J.A. (2016) *Applied Hierarchical Modeling in Ecology* AHM1 - 3.1

Examples

```
data(dragonflies)
lm(wing ~ pop + body)
```

duskySalamanders	*Age-classified counts of dusky salamanders from two mid-Atlantic states in the USA*

Description

Northern dusky salamanders (*Desmognathus fuscus*) were sampled in 12 headwater streams in Maryland and the District of Columbia, USA, annually in June and July from 2005 to 2012. Most streams were sampled at two locations separated by at least 100 m along the stream reach for a total of 21 survey locations. At each location, an observer walked two 15-m transects along the stream-bed and turned cover objects, 6 cm in the water and on the bank within 2 m of the stream. Individuals < 35mm snout-to-vent length were classed as juveniles, including both first-year and second-year juveniles. The data are counts of adults and juveniles.

Usage

```
data("duskySalamanders")
```

Format

duskySalamanders is four-dimensional (21 x 7 x 2 x 2) array of counts, with dimensions:

site a numerical site identifier.

year the year of the survey, 2005 to 2012.

age Juveniles or Adults.

survey June or July.

Source

Evan Grant and colleagues.

References

Zipkin, E.F. et al (2014) Modeling structured population dynamics using data from unmarked individuals, *Ecology*, 95(1) 22-29.

Kéry, M. & Royle, J.A. (2020) *Applied Hierarchical Modeling in Ecology* AHM2 - 2.9.2

Examples

```
data(duskySalamanders)
str(duskySalamanders)
```

---

e2dist   *Calculates the pair-wise distances between two sets of points*

Description

Calculates the Euclidian distance between each of the points defined by the coordinates in 'x' and each of those in 'y'.

Usage

```
e2dist(x, y = NULL)
```

Arguments

x   a 2-column matrix or data frame with the x and y coordinates of a set of points, or a length 2 vector with the coordinates of a single point.

y   an optional 2-column matrix or data frame, or a length 2 vector, with the x and y coordinates of a second set of points; if NULL, y is taken to be the same as x.

Value

A `nrows(x) x nrows(y)` matrix with the pair-wise distances.

Author(s)

Andy Royle

Examples

```
pts1 <- expand.grid(x = 1:5, y = 6:8)
pts2 <- cbind(x=runif(5, 1, 5), y=runif(5, 6, 8))
require(graphics)
plot(pts1)
points(pts2, pch=19, col='red')
e2dist(x=pts1, y=pts2)
# with a vector argument:
vec <- colMeans(pts2)
e2dist(vec, pts2)
# with y = NULL:
e2dist(x=pts2)
```

EurasianLynx	*Data for Eurasian lynx from Switzerland and Italy (only Alps).*

Description

The data are observations of Eurasian lynx (*Lynx lynx*) confirmed by experts and by the general public (and not confirmed by experts), summarized to 10 x 10 km grid and for 3 occasions during a 'winter' (Nov-Dec, Jan-Feb, Mar-Apr). The observations are classified as 'certain' or 'uncertain', and only the latter are assumed to be contaminated with false positives (but both are subject to false-negative errors). This data set covers the Alps in Switzerland and Italy from 1994 to 2016.

Usage

```
data("EurasianLynx")
```

Format

EurasianLynx is a data frame with 43,332 rows corresponding to observations, and 10 columns:

type   factor: 'certain' or 'uncertain'.

site.nr   site (10 x 10 km cell) identifier.

y.1, y.2, y.3   detection (1)/non-detection (0) for the three occasions.

Year   the year of the observation.

Cntry   the country, 'Italy' or 'Switzerland'.

xcoord   the x coordinate of the cell, km E of the origin.

ycoord   the y coordinate of the cell, km N of the origin.

forest   the percentage forest cover in the cell.

Source

The Foundation KORA (SCALP Project) and the Progetto Lince Italia (Anja Molinari-Jobin, Urs Breitenmoser, Fridolin Zimmermann).

References

Molinari-Jobin, A., Kéry, M., Marboutin, E., Molinari, P., Koren, I., Fuxjäger, C., Breitenmoser-Würsten, C., Wölfl, S., Fasel, M., Kos, I., Wölfl, M., & Breitenmoser, U. (2012) Monitoring in the presence of species misidentification: the case of the Eurasian lynx in the Alps. *Animal Conservation*, 15, 266-273.

Kéry, M. & Royle, J.A. (2020) *Applied Hierarchical Modeling in Ecology* AHM2 - 7.3.2 and 9.6.1.3.

Examples

```
data(EurasianLynx)
lynx <- EurasianLynx # Shorter name
str(lynx)

# Create additional columns needed for the analysis in 'unmarked':
lynx$occ.1 <- 1
lynx$occ.2 <- 2
lynx$occ.3 <- 3
lynx$sYear <- standardize(lynx$Year)
str(lynx)
```

Finnmark   *Data from surveys of birds in Finnmark in NE Norway*

Description

A total of 37 plots were placed spanning the existing variation in extent and fragmentation of willow thickets. Each plot was visited 3-5 times in early July each year from 2005 to 2008. Birds observed within 100m during a 15 min period were recorded.

Usage

```
data("Finnmark")
```

Format

Finnmark is a list with 4 elements:

species   a data frame with a row for each species and the following columns:

- species : the English name.
- latin : the Latin name.
- assemblage : the guild to which each species belongs.

sites   a data frame with a row for each plot and the following columns:

- region : a factor, 3 levels, Ilford, Komag, and Vestre Jakobselv.
- catchment : a factor, 11 levels.
- plot : a factor, the alphanumeric ID of each plot.
- plotnr : the ID number of each plot.
- area : the percentage of a 400 x 400m quadrat centered on the sampling point covered by tall willow thickets.
- edge : the length (m) of the edge of the willow thickets in the quadrat.
- height : the mean height of willows at 4 measuring points.
- density : a measure of thicket density at 4 measuring points.

counts   a sites x replicates x years x species array of counts

timeOfDay   a sites x replicates x years array with the time of day that the survey was carried out.

Source

Data generously provided by John-André Henden.

References

Henden, J.-A., Yoccoz, N.G., Ims, R.A., Langeland, K. (2013) How Spatial Variation in Areal Extent and Configuration of Labile Vegetation States Affect the Riparian Bird Community in Arctic Tundra. PLoS ONE 8(5): e63312.

Kéry, M. & Royle, J.A. (2020) *Applied Hierarchical Modeling in Ecology* AHM2 - 5.

Examples

```
data(Finnmark)
str(Finnmark)

# Create the objects needed for the analysis:
# Extract the numeric covariates and standardise
str(Finnmark$sites)
scov <- scale(Finnmark$sites[, 5:8])
str(scov)

# Convert the 'counts' array to detection/nondetection data:
y <- Finnmark$counts > 0
storage.mode(y) <- "integer"
str(y)

# Get the guild information for each species
guild <- Finnmark$species$assemblage

# Standardise the timeOfDay and replace NAs with 0
tod <- with(Finnmark, (timeOfDay - mean(timeOfDay, na.rm=TRUE))/ sd(timeOfDay, na.rm=TRUE))
tod[is.na(tod)] <- 0
str(tod)
```

---

fitstats	*Functions to return fit statistics*

Description

`fitstats` produces sum of squared errors, Chi-squared statistic and Freeman-Tukey statistic used in parboot GOF tests throughout book, starting with AHM1 - 7.5.4. `fitstats2` produces the above, plus corresponding statistics based on total detection frequency across replicate surveys; see AHM1 - 7.9.3.

Usage

```
fitstats(fm)

fitstats2(fm)
```

Arguments

fm   A fitted model object as returned by **unmarked**.

Value

For `fitstats`, a named vector of length 3 with sum of squared errors (SSE), Chi-squared statistic (Chisq) and Freeman-Tukey (freemanTukey).

For `fitstats2`, a named vector of length 6 with the same information plus corresponding statistics based on numbers(SSE.n), (Chisq.n) and (freemanTukey.n).

Author(s)

Marc Kéry & Andy Royle

References

Kéry & Royle (2016) *Applied Hierarchical Modeling in Ecology* AHM1 - 7.5.4 and 7.9.3.

FrenchPeregrines   *Data for observations of peregrines from the French Jura mountains*

Description

The data are detection/nondetection data of the Peregrine Falcon (*Falco peregrinus*) from the French Jura between 1964 and 2016 for 284 cliff sites (= territories, or sites in the context of a site-occupancy model) where a pair had been detected at least once in this period. A large proportion of sites are visited multiple times per year, but unfortunately only the aggregate results are available in each year, i.e., whether a pair was detected at least once, or never.

Usage

```
data("FrenchPeregrines")
```

Format

`FrenchPeregrines` is a data frame with 284 rows and 56 columns:

site cliff (or site) identifier.

department factor, the administrative area (Ain, Jura or Doubs).

height factor, height of the cliff, low, medium, or tall.

yr1964 to yr2016 detection histories for each year: 1 if a pair of peregrines was detected during at least one survey, 0 if no pair was detected, NA if no survey carried out in that year.

Source

Groupe Pèlerin Jura (Réne-Jean Monneret, René Ruffinoni, and colleagues)

References

Kéry, M. & Royle, J.A. (2020) *Applied Hierarchical Modeling in Ecology* AHM2 - 4.11.

Examples

```
data(FrenchPeregrines)
str(FrenchPeregrines)
# Extract the capture history data:
ch <- as.matrix(FrenchPeregrines[, 4:56])
dim(ch)
range(ch, na.rm=TRUE)
```

---

getLVcorrMat   *Compute results of a Joint Species Distribution Model (JSDM)*

Description

Compute pairwise residual correlation in occurrence or abundance in a Joint Species Distribution Model (JSDM) with latent variables.

getLVcorrMat computes the correlation matrix in residual occurrence or abundance in an latent-variable multi-species occupancy or N-mixture model which has been fit using MCMC (eg, in JAGS or WinBUGS). See Tobler et al. (2019) and Kéry & Royle (2020) chapter 8.

getEcorrMat computes the correlation matrix for the coefficients of the observed environmental covariates.

Usage

```
getLVcorrMat(lv.coef, type=c("occupancy", "Nmix"), stat=mean)

getEcorrMat(beta, stat=mean)
```

Arguments

lv.coef	MCMC chains for the coefficients of the latent variables, typically from the sims.list of a model fit; an iterations x species x latent variables array.
beta	MCMC chains for the coefficients of the environment variables (excluding the intercept), typically from the sims.list of a model fit; an iterations x species x environmental variables array.
type	Indication of whether the model fit was for occupancy data (with a probit link) or an N-mixture model based on count data.
stat	The function used to summarize the MCMC chains for the correlations; if stat = NULL. the full array with the MCMC chains is returned

Value

The relevant correlation matrix, species x species; if stat = NULL, an array, iterations x species x species, with the MCMC chains.

Author(s)

Mathias Tobler.

References

Tobler, M. et al. (2019) Joint species distribution models with species correlations and imperfect detection. *Ecology*, 100(8), e02754.

Kéry, M. & Royle, J.A. (2020) *Applied Hierarchical Modeling in Ecology* AHM2 - 8.4.2 and 8.5.4.

---

graphSSM   *Plot the results for a Gaussian state-space model*

Description

Plot the observed time-series and the estimated population trajectories for a multivariate Gaussian state-space model (SSM).

Usage

```
graphSSM(ssm, C)
```

Arguments

ssm   An object of class `jagsUI` with the results of fitting the SSM model.

C   The original count data: an individuals x year matrix of counts, no missing values.

Value

No output, the function is called for plotting.

Author(s)

Marc Kéry & Andy Royle, adapted for multivariate SSMs from the SSM graphing function `graph.ssm` in Chapter 5 in Kéry & Schaub (2012).

References

Kéry, M. & Royle, J.A. (2020) *Applied Hierarchical Modeling in Ecology* Section 1.6

Kéry, M. & Schaub, M. (2012) *Bayesian population analysis using WinBUGS - a hierarchical perspective*, Academic Press.

Kéry, M. & Royle, J.A. (2020) *Applied Hierarchical Modeling in Ecology* AHM2 - 1.6.

greenWoodpecker	*Count data from the Swiss Breeding Bird Survey for Green Woodpeckers from 2004 to 2017*

Description

The Swiss breeding bird survey ("Monitoring Häufige Brutvögel" MHB) has monitored the populations of 150 common species since 1999. The MHB sample consists of 267 1-km squares that are laid out as a grid across Switzerland. Fieldwork is conducted by about 200 skilled birdwatchers annually, most of them volunteers. Avian populations are monitored using a simplified territory mapping protocol (also called spot-mapping West of the Atlantic), where each square is surveyed up to three times during the breeding season (only twice above the tree line around 2000 m elevation). Surveys are conducted along a transect that does not change over the years and is 4-6 kms in length on average.

The data frame greenWoodpecker has count data for Green Woodpeckers *Picus viridis* from 2004 to 2017. There are some missing values: see Details.

Usage

```
data("greenWoodpecker")
```

Format

greenWoodpecker is a data frame with 267 rows and 146 columns:

x, y the x and y coordinates of the center of the quadrat; the coordinate reference system intentionally not specified.

nsurveys the number of replicate surveys *planned* in the quadrat; above the tree-line 2, otherwise 3.

elevation the mean elevation of the quadrat, m.

forest percentage forest cover

route.length the length of the route

X2004.1.count to X2017.3.count number of territories observed for each survey; 2004.1 indicates the first survey in 2004.

X2004.1.date to X2017.3.date Julian date of the survey, 1 = a weekend near 15 April.

X2004.1.time to X2017.3.time time spent in the quadrat during the survey, mins.

X2004.observer to X2017.observer numerical code identifying the observer.

Details

Missing values in the count and date columns indicate that the corresponding survey was not carried out.

On 26 occasions when surveys were carried out, the time taken was not recorded, resulting in additional NAs in the time columns.

Source

Swiss Ornithological Institute

References

Kéry, M. & Royle, J.A. (2020) *Applied Hierarchical Modeling in Ecology* AHM2 - 2.2.

Examples

```
data(greenWoodpecker)
str(greenWoodpecker)
```

HubbardBrook   *Warbler point count data from the Hubbard Brook Experimental Forest*

Description

Three replicated point counts of 10 min each, out to a maximum distance of 100 m, were conducted in each spring between 1999 and 2018 at a total of 373 locations in the Hubbard Brook Experimental Forest in New Hampshire. This data set contains the data for the following 13 species:

- AMRE - American Redstart
- BAWW - Black-and-white Warbler
- BHVI - Blue headed Vireo
- BLBW - Blackburnian Warbler
- BLPW - Blackpoll Warbler
- BTBW - Black-throated Blue Warbler
- BTNW - Black-throated Green Warbler
- CAWA - Canada Warbler
- MAWA - Magnolia Warbler
- MYWA - Myrtle Warbler
- NAWA - Nashville Warbler
- OVEN - Ovenbird
- REVI - Red-eyed Vireo

Usage

```
data("HubbardBrook")
```

Format

HubbardBrook is a list with 4 elements:

counts a sites x replicates x years x species array of counts, the number of unique individuals detected within 50m of the point count location.

sitecov a data frame with rows for 373 sites and the following columns:

- PlotID : a numeric site identifier.
- Longitude : the longitude of the point (WGS84); 2 sites have missing data.
- Latitude : the latitude of the point (WGS84); 2 sites have missing data.
- Elev : the elevation of the point, m.
- Aspect : aspect of the sample location (radians).
- Slope : slope of the sample location.
- Forest : general forest cover type of sample location.

dates a sites x replicates x years array of ordinal day of count (1 Jan = 1); NA entries occur when surveys were not carried out.

times a sites x replicates x years array with the start time of the survey, hours after midnight.

Source

Rodenhouse N.L. & Sillett, T.S. (2019) Valleywide Bird Survey, Hubbard Brook Experimental Forest, 1999-2016 (ongoing). Environmental Data Initiative. <https://doi.org/10.6073/pasta/faca2b2cf2db9d415c39b695cc7fc21 Dataset accessed 2020-01-07.

References

Betts, M.G., Rodenhouse, N.L., Sillett, T.S., Doran, P.J. & Holmes, R.T. (2008). Dynamic occupancy models reveal within-breeding season movement up a habitat quality gradient by a migratory songbird. *Ecography* 31:592–600.

Goetz, S.J., Steinberg, D., Betts, M.G., Holmes, R.T., Doran, P.J., Dubayah, R., & Hofton, M. (2010). Lidar remote sensing variables predict breeding habitat of a Neotropical migrant bird. *Ecology* 91:1569–1576.

Van Tatenhove, A., Filiberti, E., Sillett, T.S., Rodenhouse, N.L. & Hallworth, M. T. (2019). Climate-related distribution shifts of migratory songbirds and sciurids in the White Mountain National Forest. *Forests* 10:84.

Kéry, M. & Royle, J.A. (2020) *Applied Hierarchical Modeling in Ecology* AHM2 - 8.2.

Examples

```
data(HubbardBrook)
str(HubbardBrook)
```

---

image_scale   *Helper function to draw scale for image*

Description

Plots a scale in the right margin of a plot, typically an image plot.

Usage

```
image_scale(z, col, x, y = NULL, size = NULL, digits = 2, labels = c("breaks", "ranges"),
    cex.legend=1)
```

Arguments

z	the values plotted in the image, the z argument to `image`
col	a vector of colors, use the same as in the image
x	the x coordinate of the left edge of the scale bar; or a list with components x and y, each of length 2, giving the x and y coordinates of the edges of the scale bar.
y	the y coordinate of the bottom edge of the scale bar, ignored if x is a list
size	the size of the boxes making up the scale bar, a length 2 vector with the width and height, or a scalar, in which case width = height; ignored if x is a list
digits	the number of decimal places to display
labels	if "breaks", the dividing lines in the scale bar are labeled, if "ranges", each box is labeled with its range; may be abbreviated
cex.legend	the magnification to be used for axis annotation

Value

None, used for its plotting side effect.

Note

This function appears in the text as `image.scale`; renamed here to avoid confusion with generic `image` functions.

Author(s)

Andy Royle.

References

Royle, J.A., Chandler, R.B., Sollmann, R., & Gardner, B. (2014) *Spatial capture-recapture* Elsevier.

Examples

```
# uses the built-in volcano data set
require(grDevices) # for colours
require(graphics)
par(mar = c(3,3,3,6))  # make the right margin wide enough
image(t(volcano)[ncol(volcano):1,], col=terrain.colors(12))
image_scale(volcano, col=terrain.colors(12))
# Try placing the scale bar on the left
par(mar = c(3,8,3,1))  # make the left margin wide enough
image(t(volcano)[ncol(volcano):1,], col=terrain.colors(12))
image_scale(volcano, col=terrain.colors(12), x= -0.28, digits=0, cex.legend=1.2)
# Trial and error needed to get the x value right.
```

---

issj.sim   *Simulate open distance sampling data for the Island Scrub Jays*

Description

Function to simulate open distance sampling data for the Island Scrub Jays, based on Sollmann et al (2015).

Usage

```
issj.sim(B, db, lam, sigma, phi, gamma, npoints, nyrs, nbsize = -1.02)
```

Arguments

B	Radius of the circle sampled; a site is a circle of radius B around a point.
db	Distance categories; a vector of cut points from 0 to B inclusive.
lam	Expected abundance per site, a vector of length nsites or an nsites x 1 matrix.
sigma	Scale parameter of the half-normal detection function at each site, a vector of length nsites or an nsites x 1 matrix.
phi	Survival probability
gamma	Recruitment rate
npoints	Number of sites where point counts are conducted.
nyrs	Number of years
nbsize	Size parameter for the negative binomial distribution used to generate individual counts per site for year 1.

Value

A list with the following elements:

NcList	A list with one element per year, with distances of all animals from the point.
detList	A list with one element per year, a npoints x nbands matrix with the number of animals detected in each distance category for each point.

N	The (true) number of animals at each point for each year, a `nsites x nyears` matrix.
cell	The site IDs where point counts are conducted.
y	`npoints x nyears` matrix of total detections
dclass	a vector with the distance class for each animal detected
site	a corresponding vector with the site for each animal detected
nsite	the number of sites in the study area
lam, phi, gamma, sigma	
	the values of the corresponding arguments

Author(s)

Marc Kéry & Andy Royle, based on Sollmann et al (2015)

References

Sollmann, R., Gardner, B., Chandler, R.B., Royle, J.A., Sillett, T.S. (2015) An open population hierarchical distance sampling model. *Ecology* 96, 325-331.

Kéry, M. & Royle, J.A. (2016) *Applied Hierarchical Modeling in Ecology* AHM1 - 9.7.1.

Examples

```
# A toy example with just 20 sites
set.seed(2015)
tmp <- issj.sim(B = 300,
    db = c(0,50, 100, 150, 200, 250, 300),
    lam = c(3.01, 7.42, 20.51, 1.60, 0.42, 3.42, 8.24, 0.66, 0.32, 0.39, 0.46, 0.52,
      0.63, 0.36, 4.93, 0.47, 2.07, 0.42, 0.48, 0.47),
    sigma = c(110, 91, 70, 114, 135, 101, 88, 130, 133, 134, 134, 135, 131, 135, 100,
      135, 110, 135, 134, 135),
    phi = 0.6, gamma = 0.35,
    npoints = 15, nyrs = 4)
str(tmp)
# Compare the number detected with the true numbers present
with(tmp, cbind(y, N[cell, ]))
```

---

map.Nmix.resi	*Mapping of residuals*

Description

Produces a map of the mean residuals from an N-mixture model fit by function pcount in **unmarked**. Used in AHM1 - 6.9.3 to produce maps of Switzerland with the residuals for each site.

Usage

```
map.Nmix.resi(fm, x, y)
```

Arguments

fm	the fitted model object
x	x coordinates of each site
y	y coordinates of each site

Value

None. Used for its plotting side effects.

Note

In previous versions, the defaults were x = tits$coordx and y = tits$coordy, but those defaults only worked if the data object tits was in the workspace. To run the code on page 263 of AHM1, you now need to specify the coordinates, eg, map.Nmix.resi(fm5,x = tits$coordx,y = tits$coordy).

Author(s)

Marc Kéry & Andy Royle

References

Kéry, M. & Royle, J.A. (2016) *Applied Hierarchical Modeling in Ecology* AHM1 - 6.9.3.

MesoCarnivores	*Camera trap data for Bobcat, Coyote and Red Fox*

Description

Camera trap (detection/nondetection) data for Bobcat (*Lynx rufus*), Coyote (*Canis latrans*) and Red Fox (*Vulpes vulpes*) from six Mid-Atlantic states in the eastern United States analyzed by Rota et al (2016).

Usage

```
data("MesoCarnivores")
```

Format

MesoCarnivores is a list with 4 elements:

bobcat, coyote, redfox 1/0 detection data for the respective species: matrices with rows for 1437 sites x 3 replicates, where each replicate corresponds to 1 week of observations.

sitecov a data frame with rows for 1437 sites and the following columns:

- Dist_5km : the proportion of disturbed land in the surrounding 5km.
- HDens_5km : housing density in the surrounding 5km.

- Latitude : the latitude of the point, units??.
- Longitude : the longitude, units??.
- People_site : number of people photographed at a site divided by 1000.
- Trail : 1 if the camera was located on a trail, 0 otherwise.

Source

Data courtesy of eMammal, Roland Kays, Arielle Parsons, and their group at the North Carolina Museum of Natural Sciences and North Carolina State University.

References

Rota, C.T., Ferreira, M.A.R., Kays, R.W., Forrester, T.D., Kalies, E.L., McShea, W.J., Parsons, A.W., & Millspaugh, J.J. (2016) A multispecies occupancy model for two or more interacting species. *Methods in Ecology and Evolution*, 7, 1164-1173.

Kéry, M. & Royle, J.A. (2020) *Applied Hierarchical Modeling in Ecology* AHM2 - 8.2.

Examples

```
data(MesoCarnivores)
str(MesoCarnivores)
```

MHB2014   *Data from the Swiss Breeding Bird Survey MHB 2014*

Description

The Swiss breeding bird survey ("Monitoring Häufige Brutvögel" MHB) has monitored the populations of 150 common species since 1999. The MHB sample consists of 267 1-km squares that are laid out as a grid across Switzerland. Fieldwork is conducted by about 200 skilled birdwatchers annually, most of them volunteers. Avian populations are monitored using a simplified territory mapping protocol (also called spot-mapping West of the Atlantic), where each square is surveyed up to three times during the breeding season (only twice above the tree line around 2000 m elevation). Surveys are conducted along a transect that does not change over the years and is 4-6 kms in length on average.

The list MHB2014 has the full survey data for the year 2014.

Usage

```
data("MHB2014")
```

Format

MHB2014 is a list with 5 elements:

species a data frame with row for 158 species, including 13 species not recorded in the year 2014, and the following columns:

- specid : a numeric species ID based on phylogeny.
- latabb : a 6-letter abbreviation of the Latin name.
- engname : the English name.
- latname : the Latin name.
- body.length : body length in cm.
- body.mass : body mass in g.
- wing.span : wing span in cm.

sites a data frame with rows for 267 1x1 km quadrat, including 1 quadrat not surveyed in 2014, and the following columns:

- siteID : an alphanumeric site identifier.
- coordx : the x coordinate of the center of the quadrat; the coordinate reference system intentionally not specified.
- coordy : the y coordinate of the center of the quadrat.
- elev : the mean elevation of the quadrat in m.
- rlength : the length of the route walked in the quadrat in km.
- nsurvey : the number of replicate surveys *planned* in the quadrat; above the tree-line 2, otherwise 3.
- forest : percentage forest cover.
- obs14 : identifying number of the observer.

counts a sites x replicates x species array of counts

date a sites x replicates matrix with Julian dates of the surveys, 1 April = 1

dur a sites x replicates matrix with the duration of each survey, mins

Note

Section 11.3 of the AHM1 book has code to read in data from a CSV file, "MHB_2014.csv". This is a huge file, because the site data are repeated for all 158 species and the species data are repeated for all 267 sites. The MHB2014 list has all the same data, but in a more compact format. See Examples for ways to generate the objects used in the book from the list.

Source

Swiss Ornithological Institute

References

Kéry, M. & Royle, J.A. (2016) *Applied Hierarchical Modeling in Ecology* AHM1 - 11.3.

Examples

```
data(MHB2014)
str(MHB2014)

# Create the objects at foot of p.644:
( nsite <- nrow(MHB2014$sites) )    # number of sites in Swiss MHB
nrep <- 3                           # maximum number of replicate surveys per season
( nspec <- nrow(MHB2014$species) )  # 158 species occur in the 2014 data
# Check the dimensions of the 'count' array:
dim(MHB2014$count) == c(nsite, nrep, nspec)

# Create the detection/nondetection matrix 'y':
y <- MHB2014$count > 0        # this is logical, convert to integer
storage.mode(y) <- "integer" # don't use 'as.integer', that strips out dimensions and names
str(y)

# Pull out and check the data for common chaffinch, p.645:
head(tmp <- y[, , "Common Chaffinch"])
tail(tmp)
```

---

modSelFP   *Model selection results from a list of 'unmarkedFitOccuFP' models*

Description

The **unmarked** functions fitList and modSel do not currently work for 'unmarkedFitOccuFP' models, so this is a stop-gap function for use until this capability is added. It will be removed from **AHMbook** once it is included in **unmarked**.

Usage

```
modSelFP(mod.list)
```

Arguments

mod.list   A list of fitted model objects of class 'unmarkedFitOccuFP' as returned by **unmarked**.

Value

A data frame with a row for each model in the input list and columns for number of parameters, AIC, deltaAIC, AIC weight and cumulative AIC weight.

Author(s)

Chris Sutherland.

References

Kéry, M. & Royle, J.A. (2020) *Applied Hierarchical Modeling in Ecology* AHM2 - 7.2.2.

---

piFuns   *Compute multinomial cell probabilities*

Description

These functions are used internally in the multinomial-Poisson models multinomPois and gmultmix in package **unmarked** to calculate the multinomial cell probabilities from the "raw" detection probabilities. See piFuns. These functions are customized for the examples in chapter 7 of AHM1 and only allow for 3 capture occasions.

Usage

```
instRemPiFun(p)

crPiFun(p)

crPiFun.Mb(p)

MhPiFun(p)
```

Arguments

p   a numeric matrix with rows for each site and 3 columns, see Details.

Details

instRemPiFun defines the relationship between the multinomial cell probabilities and the underlying detection probability parameters in a removal design with 3 sampling periods of unequal length, specifically 2, 3 and 5 minutes. The columns of p give the detection probabilities per unit time (ie, per minute) for each site and sampling period. See AHM1 Section 7.7. This is the same as the function defined in the Examples section of piFuns in **unmarked**.

crPiFun defines a pi function for capture-recapture design with 3 capture occasions. The columns of p give the detection probabilities for the three occasions. See AHM1 Section 7.8.5. NOTE that this is not the same as the custom crPiFun defined in Section 7.9.1.

crPiFun.Mb defines a pi function for capture-recapture design with 3 capture occasions with a behavioral response. Detection probabilities do not vary with time. Column #1 of p gives the probability of detection for animals not previously detected; column #3 gives detection probability after the first detection; column #2 is ignored. See AHM1 Section 7.8.2.

MhPiFun defines a pi function for a model with individual detection heterogeneity, modeled as a random effect with a logit-normal distribution. Column #1 of p gives the mean probability of detection; p[1,2] is a value in [0, 1] which controls the scale parameter for the normal distribution; other entries are ignored. See Section 7.8.3.

Value

A matrix with a row for each site corresponding to the rows of p and...

...for instRemPiFun, a column for each detection occasion with the multinomial probability.

...for other functions, a column for each detection history; for 3 detection occasions these are 001, 010, 011, 100, 101, 110, 111.

Author(s)

Richard Chandler, Andy Royle

References

Kéry, M. & Royle, J.A. (2016) *Applied Hierarchical Modeling in Ecology* AHM1 - 7.

Examples

```
# Capture probs for 5 sites, with 3 survey periods
( p <- matrix(0.4, nrow=5, ncol=3) )
# Removal model
instRemPiFun(p)
# The corresponding obsToY matrix:
matrix(1, 2, 3)

# Capture-recapture model
crPiFun(p)
# The corresponding obsToY matrix:
matrix(1, 3, 7)

# Capture-recapture model with behavioural response
( pMb <- cbind(rep(0.7, 5), NA, 0.3) )
crPiFun.Mb(pMb)
# The corresponding obsToY matrix:
matrix(1, 3, 7)

# Capture-recapture model with heterogeneity
pMh <- cbind(rep(0.4, 5), NA, NA)
pMh[1, 2] <- 0.3
pMh
MhPiFun(pMh)
# The corresponding obsToY matrix:
matrix(1, 3, 7)
```

---

playRN   *Function to play Royle-Nichols (RN) model*

Description

Function generates replicated count data under the binomial N-mixture model of Royle (2004), then 'degrades' the counts to detection/nondetection and fits the Royle-Nichols (RN) model (Royle & Nichols 2003) using **unmarked** and estimates site-specific abundance.

Usage

```
playRN(M = 267, J = 3, mean.abundance = 1, mean.detection = 0.3,
    show.plots = TRUE, verbose = TRUE)
```

Arguments

M	The number of sites.
J	The number of visits to each site.
mean.abundance	Expected abundance at each site.
mean.detection	Expected detection at each survey at each site.
show.plots	choose whether to show plots or not. Set to FALSE when using function in simulations.
verbose	if FALSE, output to the Console will be suppressed.

Value

A list with the following elements:

nsites	The number of sites, equal to M in the input
nvisits	The number of visits, equal to J in the input
coef	A named vector of coefficients for the linear models for expected number and detection probability
slope	Slope of the regression of the estimated number on the true number; 1 if the model is perfect

Author(s)

Marc Kéry & Andy Royle

References

Royle, J.A. & Nichols, J.D. (2003) Estimating abundance from repeated presence-absence data or point counts, *Ecology*, 84, 777-790.

Royle, J.A. (2004) N-mixture models for estimating population size from spatially replicated counts, *Biometrics*, 60, 108-115.

Kéry, M. & Royle, J.A. (2016) *Applied Hierarchical Modeling in Ecology* AHM1 - 6.13.1.

Examples

```
# Run a simulation with the default arguments and look at the results:
playRN()


# Execute the function using various settings
playRN(M = 100, J = 3, mean.abundance = 0.1)  # Increasing abundance
playRN(M = 100, J = 3, mean.abundance = 1)
playRN(M = 100, J = 3, mean.abundance = 5)
```

```
playRN(M = 100, J = 3, mean.detection = 0.3)  # Increasing detection
playRN(M = 100, J = 3, mean.detection = 0.5)
playRN(M = 100, J = 3, mean.detection = 0.7)
playRN(M = 100, J = 20)                        # More visits
playRN(M = 1000, J = 3)                        # More sites
```

---

plot_Nmix_resi   *Produce some residual plots*

Description

Function does diagnostic plots for one binomial N-mixture model fitted with all three mixture distributions currently available in package **unmarked**: Poisson, negative binomial and zero-inflated Poisson. For each, fitted values vs. observed data and residuals vs. fitted values are plotted.

Usage

```
plot_Nmix_resi(fmP, fmNB, fmZIP)
```

Arguments

fmP	Fitted model object for Poisson distribution.
fmNB	Fitted model object for negative binomial distribution.
fmZIP	Fitted model object for zero-inflated Poisson distribution.

Value

None, used for its plotting effect.

Note

This function appears in the text as plot.Nmix.resi; renamed here to avoid confusion with generic plot functions.

Author(s)

Marc Kéry & Andy Royle

References

Kéry, M. & Royle, J.A. (2016) *Applied Hierarchical Modeling in Ecology* AHM1 - 6.9.3.

---

ppc.plot *Plot results from posterior predictive check*

Description

Function plots results from posterior predictive check in AHM1 section 6.8 for a fitted model object with JAGS.

Usage

```
ppc.plot(fm)
```

Arguments

fm   The fitted model object

Value

None, used for its plotting side effect.

Author(s)

Marc Kéry & Andy Royle

References

Kéry, M. & Royle, J.A. (2016) *Applied Hierarchical Modeling in Ecology* AHM1 - 6.8.

---

sim.fn *Simulate a Poisson point process*

Description

Simulates animal or plant locations in space according to a homogenous Poisson process. This process is characterized by the intensity, which is the average number of points per unit area. The resulting point pattern is then discretized to obtain abundance data and presence/absence (or occurrence) data. The discretization of space is achieved by choosing the cell size. It is used in AHM1 Section 1.1 to help to understand the relationship between point patterns, abundance data and occurrence data (also called presence/absence or distribution data). For a similar and somewhat more sexy version of this function, see `simPPe`.

Usage

```
sim.fn(quad.size = 10, cell.size = 1, intensity = 1, show.plot = TRUE)
```

Arguments

`quad.size`	The length of each side of the quadrat (in arbitrary units)
`cell.size`	The length of each side of the cells into which the quadrat is divided. The ratio of quad.size to cell.size must be an integer.
`intensity`	The average number of points (animals or plants) per unit area.
`show.plot`	If TRUE, the results are plotted. Set to FALSE when running many simulations.

Value

A list with the values of the arguments and the following additional elements:

`exp.N`	Expected population size in quadrat
`breaks`	boundaries of grid cells
`n.cell`	Number of cells in the quadrat
`mid.pt`	Cell mid-points
`M`	Realized population size in quadrat
`u1`	x coordinate of each individual
`u2`	y coordinate of each individual
`N`	The number of individuals in each cell (a vector of length n.cell)
`z`	Presence/absence (1/0) in each cell (a vector of length n.cell)
`psi`	Proportion of cells occupied, ie, the species is present.

Author(s)

Marc Kéry and Andy Royle

References

Kéry, M. & Royle, J.A. (2016) *Applied Hierarchical Modeling in Ecology* AHM1 - 1.1.

Examples

```
# Generate a simulated data set with default arguments and look at the structure:
tmp <- sim.fn()
str(tmp)

# Effect of grain size of study on abundance and occupancy (intensity constant)
tmp <- sim.fn(quad.size = 10, cell.size = 1, intensity = 0.5)
tmp <- sim.fn(quad.size = 10, cell.size = 2, intensity = 0.5)
tmp <- sim.fn(quad.size = 10, cell.size = 5, intensity = 0.5)
tmp <- sim.fn(quad.size = 10, cell.size = 10, intensity = 0.5)

# Effect of intensity of point pattern (intensity) on abundance and occupancy
tmp <- sim.fn(intensity = 0.1) # choose default quad.size = 10, cell.size = 1
tmp <- sim.fn(intensity = 1)
tmp <- sim.fn(intensity = 5)
tmp <- sim.fn(intensity = 10)
```

---

sim.ldata *Simulation of distance sampling data.*

Description

Simulates non-hierarchical line transect data under conventional distance sampling (CDS). It subjects N individuals to sampling, and then retains the value of distance from transect only for individuals that are captured.

Usage

```
sim.ldata(N = 200, sigma = 30, show.plot = TRUE)
```

Arguments

N	number of individuals along transect with distance u(-100, 100)
sigma	scale parameter of half-normal detection function
show.plot	choose whether to show plots or not. Set to FALSE when using function in simulations.

Value

A list with the following elements:

N	the number of individuals along the transect
sigma	scale parameter of half-normal detection function
xall	distance from the transect line for all N individuals
x	absolute distance from the transect line for those individuals detected

Author(s)

Marc Kéry & Andy Royle

References

Kéry, M. & Royle, J.A. (2016) *Applied Hierarchical Modeling in Ecology* AHM1 - 8.2.3.

Examples

```
# Simulate a data set with the default arguments and look at the structure of the output:
tmp <- sim.ldata()
str(tmp)
```

---

sim.pdata   *Simulate non-hierarchical point transect (= point count) data*

Description

Function simulates coordinates of individuals on a square with a count location at the center point.

Usage

```
sim.pdata(N = 1000, sigma = 1, B = 3, keep.all = FALSE, show.plot = TRUE)
```

Arguments

N	total population size in the square
sigma	scale of half-normal detection function
B	circle radius; the data are simulated over a square of side 2 * B, but individuals outside the circle of radius B are not detected.
keep.all	if TRUE, the data for all individuals are returned; if FALSE, only for individuals detected.
show.plot	choose whether to show plots or not. Set to FALSE when using function in simulations.

Value

A list with the following elements:

N	total population size in the square
sigma	scale of half-normal detection function
B	circle radius
u1, u2	the x and y coordinates of each of the individuals
d	the distance of each individual from the center of the circle
y	a 0/1 indicator showing whether each individual is detected or not, a vector of length N
N.real	the realized number of individuals within the circle of radius B

Author(s)

Marc Kéry & Andy Royle

References

Kéry, M. & Royle, J.A. (2016) *Applied Hierarchical Modeling in Ecology* AHM1 - 8.2.5.1.

Examples

```
# Simulate a data set with the default arguments and look at the structure of the output:
tmp <- sim.pdata()
str(tmp)
```

---

sim.spatialDS	*Simulates data for a basic spatial distance sampling model*

Description

Generates data with the following steps:

1. Simulate a spatially correlated habitat covariate (x) over a grid of pixels covering a square.

2. Distribute the population of N individuals over the square with probability of location in a pixel related to the covariate.

3. Decide which individuals are detected using a distance sampling model with an observer at the center of the square, with either a half normal or a logit detection function. (Note that all the individuals in the square can be detected.)

4. If keep.all = FALSE, return the locations of only the individuals detected.

To recreate the data sets used in the book with R 3.6.0 or later, include sample.kind="Rounding" in the call to set.seed. This should only be used for reproduction of old results.

Usage

```
sim.spatialDS(N = 1000, beta = 1, sigma = 1, keep.all = FALSE, B = 3,
  model=c("logit", "halfnorm"), lambda = B/3, useHabitat, show.plot=TRUE)
```

Arguments

N	total population size in the square
beta	coefficient of SOMETHING on spatial covariate x
sigma	scale parameter of detection function
keep.all	if TRUE, the data for all individuals are returned; if FALSE, only for individuals detected.
B	distance from the observer to the side of the square. This is usually set so that the probability of detection of individuals outside the square is negligable, eg, B = 3*sigma.
model	The detection function used, can be "logit" or "halfnorm": see Details.
lambda	The scale parameter for the spatially autocorrelated Habitat covariate.
useHabitat	If the output from a previous simulation is provided, the same Habitat covariate will be used (and lambda will be ignored).
show.plot	choose whether to show plots or not. Set to FALSE when using function in simulations.

Details

The "logit" detection function is `2*plogis(-d^2/(2*sigma^2))`, which corresponds to the detection model implemented in `unmarked::pcount.spHDS`.

Value

A list with the values of the input arguments and the following additional elements:

u1	x coordinate of each animal
u2	y coordinate of each animal
d	distance of each animal from the center of the circle
pixel.id	the pixel in which each animal is located, the row number in `grid`
y	indicator of detection of each animal, a vector of length N
N.real	the number of animals inside the circle of radius B
Habitat	Value of the spatially correlated habitat covariate, a 900 x 1 matrix
grid	Coordinates of the center of each pixel, a dataframe with 900 rows and 2 columns

If `keep.all = FALSE` (the default), only the animals detected are included in `u1,u2,d,pixel.id`.

Note

Kéry & Royle (2016, p.535 and discussion p.540) and earlier versions of AHMbook included a hazard rate detection function. This is problematic because the detection probability at distance zero is less than 1 ($p(0) < 1$) and should not be used. It is replaced here with the `logit` detection function, which does have `p(0) = 1`.

Author(s)

Marc Kéry & Andy Royle

References

Kéry, M. & Royle, J.A. (2016) *Applied Hierarchical Modeling in Ecology* AHM2 - 9.8.3.

Examples

```
# Generate data with the default arguments and look at the structure:
tmp <- sim.spatialDS()
str(tmp)

# Generate date with model = "logit" and analyse the data with unmarked::pcount.spatialHDS
# RNGkind(sample.kind = "Rounding") # run this for R >= 3.6.0
set.seed(1234)
tmp <- sim.spatialDS(model="logit")
# Plot shows a large area of good habitat west of the observer with many animals detected
str(tmp)  # 272 animals detected out of 850 inside the circle (N.real)

# Fit some models with unmarked
if(require(unmarked)) {
```

```
    # Get the count of animals detected in each pixel
    pixel.count <- tabulate(tmp$pixel.id, nbins=nrow(tmp$grid))
    # Centre the Habitat covariate
    Habitat <- tmp$Habitat - mean(tmp$Habitat)
    # Create a detection covariate: distance between observer and pixel center
    dist <- with(tmp, sqrt((grid[,1]-B)^2 + (grid[,2]-B)^2))
    # Construct an unmarkedFrame
    umf <- unmarkedFramePCount(y=cbind(pixel.count),
        siteCovs=data.frame(dist=dist, Habitat=Habitat))
    summary(umf)

    (fm0 <- pcount.spHDS(~ -1 + I(dist^2) ~ 1, umf, K=20))
    (fm1 <- pcount.spHDS(~ -1 + I(dist^2) ~ Habitat, umf, K=20))
    # The model with Habitat has much lower AIC
    # Get an estimate of the total population in the square (true is N = 1000)
    sum(predict(fm1, type='state')[, 1])
}
```

---

sim.spatialHDS   *Simulates data for a hierarchical spatial distance sampling model*

Description

Generates data for distance sampling from spatially-replicated point transects, with density dependent on a spatially correlated habitat covariate. For each point count, the procedure is:

1. Simulate the habitat covariate over a grid of pixels covering a square.

2. Distribute the population of individuals over the square with probability of location in a pixel related to the covariate.

3. Decide which individuals are detected using a distance sampling model with an observer at the center of the square, with a half normal detection function. (Note that individuals outside the circle of radius B can be detected.)

The locations and detection status of individuals at all sites are collated and returned, except for individuals at sites when none are detected.

To recreate the data sets used in the book with R 3.6.0 or later, include sample.kind="Rounding" in the call to set.seed. This should only be used for reproduction of old results.

Usage

```
sim.spatialHDS(lam0 = 4, sigma = 1.5, B = 3, nsites = 100,
  beta1 = 1, npix = 20, show.plots = 3)
```

Arguments

lam0	the expected number of individuals in the square of side = 2*B if the habitat covariate had no effect (ie, beta1 = 0).
sigma	scale parameter of the half-normal detection function

B	distance from the observer to the side of the square. This is usually set so that the probability of detection of individuals outside the square is negligable, eg, B = 3*sigma.
nsites	number of sites
beta1	the size of the effect of the habitat covariate on the number of individuals in a pixel.
npix	the number of pixels along each dimension of the square: the entire grid has npix x npix pixels.
show.plots	the number of sites for which plots should be displayed.

Value

A list with the following components:

data	a matrix with columns for siteID, the coordinates of each individual (u1,u2), distance from the center of the square (d) and detection status, 0/1 (y), and rows for each individual, except that individuals at sites where no individuals were detected are NOT included, and those sites are represented by NAs
B	the radius of the circle; the data are simulated over a square of side 2*B
Habitat	a matrix with the values of the Habitat covariate for each pixel at each site
grid	a data frame with the coordinates of each pixel (same for all sites)
N	the realized number of individuals in the square at each site
nsites	the number sites

Author(s)

Marc Kéry & Andy Royle

References

Kéry, M. & Royle, J.A. (2016) *Applied Hierarchical Modeling in Ecology* AHM1 - 9.8.5.

Examples

```
# Generate data with the default arguments and look at the structure:
tmp <- sim.spatialHDS()
str(tmp)
```

---

sim3Occ	*Simulate detection/nondetection data for static 3-level occupancy models*

Description

Function generates 3-level occupancy data with possibility of site-specific random variation at every level, "time effects" at the middle and the lower levels and effects of one distinct covariate at each level.

Usage

```
sim3Occ(nunits = 100, nsubunits = 5, nreps = 3,
  mean.psi = 0.8, beta.Xpsi = 1, sd.logit.psi = 0,
 mean.theta = 0.6, theta.time.range = c(-1, 1), beta.Xtheta = 1, sd.logit.theta = 0,
  mean.p = 0.4, p.time.range = c(-2, 2), beta.Xp = -1, sd.logit.p = 0,
  show.plot = TRUE, verbose = TRUE)
```

Arguments

nunits	Number of main units (large quadrats)
nsubunits	Number of subunits (nested subsamples within each main unit)
nreps	Number of replicate surveys in every subunit
mean.psi	Mean large-scale, unit-level occupancy probability (psi)
beta.Xpsi	effect on psi of covariate A (at main unit level)
sd.logit.psi	SD of logit(psi), unstructured site variation in psi
mean.theta	Mean small-scale (subunit) occupancy probability (theta)
theta.time.range	
	range of theta 'intercepts' for subunits
beta.Xtheta	effect on theta of covariate B (at subunit level)
sd.logit.theta	SD of logit(theta), unstructured site variation in theta
mean.p	Mean per-survey detection probability
p.time.range	range of p 'intercepts' for replicates
beta.Xp	effect on p of covariate C (unit by subunit by replicate)
sd.logit.p	SD of logit(p)
show.plot	choose whether to show plots or not. Set to FALSE when using function in simulations.
verbose	if TRUE, output will be written to the Console.

Value

A list with the values of the input arguments and the following additional elements:

theta.time.effect
                Simulated time effect on theta, a vector of length nsubunits

p.time.effect   Simulated time effect on p, a vector of length nreps

p   Detection probabiliy, a nunits x nsubunits x nreps array

z   Occupancy indicator for main units, a nunits x 1 array of 0/1

a   Occupancy indicator for subunits, a nunits x nsubunits array of 0/1

y   Detection array, a nunits x nsubunits x nreps array of 0/1

sum.z   True number of occupied main units

obs.sum.z   Observed number of occupied main units

sum.z.a   Number of units with >=1 occupied, surveyed subunit

covA   Simulated covariate A, a vector of length nunits

covB   Simulated covariate B, a nunits x nsubunits matrix

covC   Simulated covariate C, a nunits x nsubunits x nreps array

Author(s)

Marc Kéry & Andy Royle

References

Kéry, M. & Royle, J.A. (2016) *Applied Hierarchical Modeling in Ecology* AHM1 - 10.10.

Examples

```
# Generate data with the default arguments and look at the structure:
tmp <- sim3Occ()
str(tmp)

# 'Null' model (model 1)
str(data <- sim3Occ(nunits = 100, nsubunits = 5, nreps = 3, mean.psi = 0.8,
  beta.Xpsi = 0, sd.logit.psi = 0, mean.theta = 0.6, theta.time.range = c(0, 0),
  beta.Xtheta = 0, sd.logit.theta = 0, mean.p = 0.4, p.time.range = c(0,0),
  beta.Xp = 0, sd.logit.p = 0))

# No covariate effects, no random variability (model 2)
str(data <- sim3Occ(nunits = 100, nsubunits = 5, nreps = 3, mean.psi = 0.8,
  beta.Xpsi = 0, sd.logit.psi = 0, mean.theta = 0.6, theta.time.range = c(-1, 1),
  beta.Xtheta = 0, sd.logit.theta = 0, mean.p = 0.4, p.time.range = c(-2,2),
  beta.Xp = 0, sd.logit.p = 0))

# All covariate effects, but no random variability (model 3)
str(data <- sim3Occ(nunits = 100, nsubunits = 5, nreps = 3, mean.psi = 0.8,
  beta.Xpsi = 1, sd.logit.psi = 0, mean.theta = 0.6, theta.time.range = c(-1, 1),
  beta.Xtheta = 1, sd.logit.theta = 0, mean.p = 0.4, p.time.range = c(-2,2),
```

```
      beta.Xp = -1, sd.logit.p = 0))

  # Most complex model with all effects allowed for by sim function (model 4)
  str(data <- sim3Occ(nunits = 100, nsubunits = 5, nreps = 3, mean.psi = 0.8,
    beta.Xpsi = 1, sd.logit.psi = 1, mean.theta = 0.6, theta.time.range = c(-1, 1),
    beta.Xtheta = 1, sd.logit.theta = 1, mean.p = 0.4, p.time.range = c(-2,2),
    beta.Xp = -1, sd.logit.p = 1))
```

---

simCJS   *Simulate individual capture-histories under a Cormack-Jolly-Seber*
                                (CJS) survival model

Description

Function generates individual capture-histories under a CJS model with possibly time-dependent
parameters. The number of values for interval-specific survival (phi) and time-specific detection (p)
must be ensured to be equal to the number of occasions (n.occ) minus 1.

Usage

```
  simCJS(n.occ = 6, n.marked = 20, phi = 0.7, p = 0.4, show.plot = TRUE)
```

Arguments

n.occ	The number of occasions (eg, years).
n.marked	The number of individuals marked per year except for the last year; scalar, or a vector of length (n.occ-1).
phi	The apparent survival probability between years; scalar, or a vector of length (n.occ-1).
p	The recapture probability for all except the first year; scalar, or a vector of length (n.occ-1).
show.plot	Choose whether to show plots or not. Set to FALSE when using function in simulations.

Value

A list with the values of the arguments entered and the following additional elements:

z	n.ind x n.occ matrix, 1 if the individual is alive and in the study area, 0 otherwise.
ch	n.ind x n.occ matrix, 1 if the individual is captured, 0 otherwise.
f	n.ind vector, occasion the individual was first captured, and marked.
n.ind	scalar, total number of individuals marked.
n.alive	n.occ vector, number of animals alive and in the study area each year.

Author(s)

Marc Kéry & Andy Royle, based on code written by Michael Schaub for Chapter 7 of Kéry & Schaub (2012).

References

Kéry, M. & Schaub, M. (2012) *Bayesian population analysis using WinBUGS - a hierarchical perspective*, Academic Press.

Kéry, M. & Royle, J.A. (2020) *Applied Hierarchical Modeling in Ecology* AHM2 - 3.2.2.

Examples

```
# Run with the default arguments and look at the structure of the output:
set.seed(123)
tmp <- simCJS()
str(tmp)
colSums(tmp$ch)
```

---

simComm	*Simulate community occupancy or community abundance data*

Description

Simulate detection/nondetection or count data, respectively, under a community occupancy or abundance model with random species effects for psi/lambda and p (both including effects of one covariate, 'habitat' for psi/lambda and 'wind speed' for p) (introduced in AHM1 - 11.2)

Usage

```
simComm(type = c("det/nondet", "counts"), nsites = 30, nreps = 3, nspecies = 100,
  mean.psi = 0.25, sig.lpsi = 1, mu.beta.lpsi = 0, sig.beta.lpsi = 0,
  mean.lambda = 2, sig.loglam = 1, mu.beta.loglam = 1, sig.beta.loglam = 1,
  mean.p = 0.25, sig.lp = 1, mu.beta.lp = 0, sig.beta.lp = 0, show.plot = TRUE)
```

Arguments

type	choose whether you want to simulate detection/nondetection ("det/nondet") data or count data ("counts").
nsites	number of sites
nreps	number of replicate samples (occasions or repeated measurements)
nspecies	total number of species in the area that is sampled by these sites (regional species pool)
mean.psi	community mean of occupancy probability over all species in community (probability scale)
sig.lpsi	community standard deviation of logit(occupancy probability intercept)

mu.beta.lpsi	community mean of the effects of 'habitat' covariate on occupancy probability
sig.beta.lpsi	community standard deviation of the effects of 'habitat' covariate on occupancy probability
mean.lambda	community mean of expected abundance over all species in superpopulation
sig.loglam	community standard deviation of log(lambda intercept)
mu.beta.loglam	community mean of the effects of 'habitat' covariate on log(lambda)
sig.beta.loglam	
	community standard deviation of the effects of 'habitat' covariate on expected abundance
mean.p	community mean of detection probability over all species in superpopulation (probability scale)
sig.lp	community standard deviation of logit(detection probability intercept)
mu.beta.lp	community mean of the effects of 'wind' covariate on detection probability
sig.beta.lp	community standard deviation of the effects of 'wind' covariate on detection probability
show.plot	choose whether to show plots or not. Set to FALSE when using function in simulations.

Details

Function simulates data from repeated sampling of a metacommunity (or spatially structured community) according the model of Dorazio & Royle (JASA, 2005) for type = "det/nondet" (this is the default) or under the model of Yamaura et al. (2012) for type = "counts".

Occupancy probability (psi) or expected abundance (lambda) can be made dependent on a continuous site covariate 'habitat', while detection probability can be made dependent an observational covariate 'wind'. Both intercept and slope of the two log-linear or logistic regressions (for occupancy or expected abundance, respectively, and for detection) are simulated as draws from a normal distribution with mean and standard deviation that can be selected using function arguments.

Specifically, the data are simulated under the following linear models:

(1) for type = "det/nondet" **(i.e., community occupancy)**

```
(occupancy (psi) and detection (p) for site i, replicate j and species k)
 psi[i,k] <- plogis(beta0[k] + beta1[k] * habitat[i]        Occupancy
 p[i,j,k] <- plogis(alpha0[k] + alpha1[k] * wind[i,j]        Detection
```

(2) for type = "counts" **(i.e., community count)**

```
(exp. abundance (lambda) and detection (p) for site i, rep. j and species k)
 lambda[i,k] <- exp(beta0[k] + beta1[k] * habitat[i]        E(N)
 p[i,j,k] <- plogis(alpha0[k] + alpha1[k] * wind[i,j]        Detection
```

Species-specific heterogeneity in intercepts and slopes is modelled by up to four independent normal distributions (note: no correlation between the intercepts as in Dorazio et al. (2006) or Kéry & Royle (2008))

(1) for `type = "det/nondet"` **(i.e., community occupancy)**

```
beta0 ~ dnorm(logit(mean.psi), sig.lpsi)        Mean and SD of normal distribution
 beta1 ~ dnorm(mu.beta.lpsi, sig.beta.lpsi)
 alpha0 ~ dnorm(logit(mean.p), sig.lp)
 alpha1 ~ dnorm(mu.beta.lp, sig.beta.lp)
```

(2) for `type = "counts"` **(i.e., community count)**

```
beta0 ~ dnorm(log(mean.lambda), sig.loglam)        Mean and SD of normal distribution
beta1 ~ dnorm(mu.beta.loglam, sig.beta.loglam)
alpha0 ~ dnorm(logit(mean.p), sig.lp)
alpha1 ~ dnorm(mu.beta.lp, sig.beta.lp)
```

Value

A list with the arguments supplied and the following additional elements:

(1) for `type = "det/nondet"` **(i.e., community occupancy)**

psi	psi[i,k] occupancy probability for site i and species k
p	p[i,j,k] detection probability for site i, occasion j and species k
z	z[i,k] true presence/absence for site i and species k
z.obs	z.obs[i,k] observed presence/absence for site i and species k
y.all	y.all[i,j,k] detection/nondetection data for site i, occasion j and species k for all species
y.obs	y.obs[i,j,k] detection/nondetection data for site i, occasion j and species k for observed species
y.sum.all	detection frequency for all species
y.sum.obs	detection frequency for observed species
Ntotal.fs	finite sample (or conditional) species richness
Ntotal.obs	observed species richness
S.true	true number of species occurring at each site
S.obs	observed number of species occurring at each site

(2) for `type = "counts"` **(i.e., community count)**

lambda	lambda[i,k] expected number of individuals for site i and species k
p	p[i,j,k] detection probability for site i, occasion j and species k
N	N[i,k] true number present for site i and species k
y.all	y.all[i,j,k] number observed for site i, occasion j and species k for all species
y.obs	y.obs[i,j,k] number observed for site i, occasion j and species k for observed species
ymax.obs	ymax.obs[i,k] maximum number observed for site i and species k
Ntotal.fs	finite sample (or conditional) species richness
Ntotal.obs	observed species richness

Author(s)

Marc Kéry & Andy Royle, with community occupancy model code partly based on code by Richard Chandler.

References

Dorazio, R.M. & Royle, J.A. (2005) Estimating size and composition of biological communities by modeling the occurrence of species. *J American Statistical Association*, 100, 389-398.

Dorazio, R.M., et al (2006) Estimating species richness and accumulation by modeling species occurrence and detectability. *Ecology* 87, 842-854.

Kéry, M. & Royle, J.A. (2008) Hierarchical Bayes estimation of species richness and occupancy in spatialy replicated surveys. *J. Appl. Ecol.* 45, 589-598.

Yamaura, Y., et al. (2012) Biodiversity of man-made open habitats in an underused country: a class of multispecies abundance models for count data. *Biodiversity and Conservation* 21, 1365-1380.

Kéry, M. & Royle, J.A. (2016) *Applied Hierarchical Modeling in Ecology* AHM1 - 11.2.

Examples

```
# Default arguments:
str(simComm())

# Some possibly interesting settings of the function
data <- simComm(nsites = 267, nspecies = 190, mean.psi = 0.25, sig.lpsi = 2,
  mean.p = 0.12, sig.lp = 2) # similar to Swiss MHB
data <- simComm(mean.psi = 1)        # all species occur at every site
data <- simComm(mean.p = 1)          # no measurement error (perfect detection)

# Effect of spatial sample size (nsites) on species richness in sample (Ntotal.fs)
data <- simComm(nsites=50, nspecies = 200) # 1-3 are usually missed in sample
data <- simComm(nsites=30, nspecies = 200) # 4-6 usually missed
data <- simComm(nsites=10, nspecies = 200) # around 30 typically missed
```

---

simDataDK	*Simulate data for an integrated species distribution model (SDM) of Dorazio-Koshkina*

Description

The function generates a population represented as a point pattern in a heterogeneous landscape and simulates data from two different sources: (1) opportunistic presence-only data, and (2) replicate counts or detection/nondetection data in randomly-placed quadrats. This is the scenario for the integrated models described by Dorazio (2014) and Koshkina et al. (2017). The former assumes counts as data source (2) while the latter assume detection/nondetection data.

A Poisson point pattern (PPP) with intensity a function of a covariate X and intercept and coefficient beta is simulated on a discrete (pixel-based) approximation of a continuous landscape.

This PPP is first thinned with a pixel-wise thinning probability controlled by a covariate W and coefficients alpha, and second, with a landscape-wise random drop-out process to produce a first data set of presence-only kind.

A second data set is simulated by imagining replicated counts conducted in randomly-selected quadrats within the landscape. Detection of individuals is imperfect, with probability of detection controlled by the covariate W and coefficients gamma. These counts can be quantized to detection/nondetection data for use in a model as in Koshkina et al. (2017).

For simDataDK1 animals are limited to one individual per pixel; this is not the case for simDataDK.

To recreate the data sets used in the book with R 3.6.0 or later, include sample.kind="Rounding" in the call to set.seed. This should only be used for reproduction of old results.

Usage

```
simDataDK(sqrt.npix = 100, alpha = c(-1,-1), beta = c(6,0.5),
  drop.out.prop.pb = 0.7, quadrat.size = 4, gamma = c(0,-1.5),
  nquadrats = 250, nsurveys = 3, show.plot = TRUE)

simDataDK1(sqrt.npix = 100, alpha = c(-1,-1), beta = c(6,0.5),
  drop.out.prop.pb = 0.7, quadrat.size = 4, gamma = c(0,-1.5),
  nquadrats = 250, nsurveys = 3, show.plot = TRUE)
```

Arguments

sqrt.npix   number of pixels along each side of square state space (the 'landscape'); the total number of pixels is then npix = sqrt.npix^2. For simDataDK1 this limits the population size to 1 per pixel.

alpha   coefficients for the relationship: logit(b) = alpha[1] + alpha[2] * W, where b is the sampling detection bias in the presence-only observations.

beta   coefficients for the relationship: log(lambda) = beta[1] + beta[2] * X, where lambda is the intensity of the Poisson point process. If the values of beta result in very large numbers of animals, an error will occur.

drop.out.prop.pb
                 proportion of presence-only points at the end that are discarded.

quadrat.size   length of the side of quadrats for conducting replicate counts in *pixel* units; note that sqrt.npix / quadrat.size must yield an integer.

gamma   coefficients for the relationship: logit(p) = gamma[1] + gamma[2] * W, where p is the probability of detecting an individual during the count surveys in the quadrats.

nquadrats   the number of quadrats selected for the count survey.

nsurveys   the number of replicate counts in each quadrat.

show.plot   if TRUE, summary plots are displayed.

Value

A list with the values of the input arguments and the following additional elements:

npix	the number of pixels in the landscape
s.area	the area of the whole landscape = 4
s.loc	2-column vector with the location of each pixel
xcov	values of the 'X' (intensity) covariate
wcov	values of the 'W' (detection) covariate
N.ipp	true number of individuals in the landscape
pixel.id.ipp	pixel ID for each individual in the population
loc.ipp	coordinates for each individual in the population
pTrue.ipp	probability of detection for each individual for presence-only data
pixel.id.det	pixel ID for each individual detected opportunistically
N.det	number of detections
loc.det	coordinates of each individual detected opportunistically
pcount	probability of detection during count surveys, varies by quadrat
fullCountData	matrix with rows for each quadrat, columns for ID, x and w coords, true N, and 3 replicate counts
countData	as above, but rows for quadrats sampled only
s	a Raster Stack with layers for 'X', 'W', and number in each pixel, 'n'
squad	a Raster Stack corresponding to the quadrats, with mean 'X' and 'W' and true abundance, 'N'

Author(s)

Marc Kéry, Andy Royle & Mike Meredith, based on the code written by Dorazio (2014) and adapted by Koshkina et al. (2017).

References

Dorazio, R.M. (2014) Accounting for imperfect detection and survey bias in statistical analysis of presence-only data. *Global Ecology and Biogeography*, 23, 1472-1484.

Koshkina, V., Wang, Y., Gordon, A., Dorazio, R.M., White, M., & Stone, L. (2017) Integrated species distribution models: combining presence-background data and site-occupany data with imperfect detection. *Methods in Ecology and Evolution*, 8, 420-430.

Kéry, M. & Royle, J.A. (2020) *Applied Hierarchical Modeling in Ecology* AHM2 - 10.

Examples

```
# Run the function with default values and look at the output
str(tmp <- simDataDK(), 1)  # use str(., max.level=1) to limit the amount of output.

str(tmp <- simDataDK(show.plot=FALSE), 1)  # no plots

str(tmp <- simDataDK(sqrt.npix = 500), 1)  # much larger landscape

str(tmp <- simDataDK(alpha = c(-1,1)), 1)  # positive effect of W on bias rate parameter b
```

```
str(tmp <- simDataDK(beta = c(6, 0.5)), 1) # lower density

str(tmp <- simDataDK(drop.out.prop = 0), 1)# No final uniform thinning ("drop out")

str(tmp <- simDataDK(beta = c(6, 1)), 1)   # steeper gradient of habitat suitability
```

| simDCM | *Simulate detection/nondetection data under a general dynamic com-* |
| | *munity (site-occupancy) model* |

Description

Function to simulate detection/nondetection data under a general dynamic community (= dynamic, multi-species site-occupancy) model, including:

* annual variation in the probabilities of patch persistence, colonization and detection is specified by the bounds of a uniform distribution.

* species heterogeneity around the community means is specified by the SD of a normal distribution and expressed on the logit scale

* one covariate is allowed per parameter (site covariate for psi1, site-year covariate for phi and gamma and site-year-rep covariate for p). Each covariate is allowed to differ among species again according to a logit-normal model of heterogeneity.

* additional detection heterogeneity at the site- or the occasion level, with the possibility of a temporal trend in this heterogeneity over years. E.g., an annual trend in detection heterogeneity at the site or the occasion level is specified by the value in the first and the last year. Hence, trend.sd.site = c(0, 1) will result in a linear trend in the magnitude of site-level heterogeneity in detection from 0 in the first year to 1 in the last year, with interpolation for the years in between.

* additional detection heterogeneity that among occasions according to a quadratic effect of occasion number (to model the typical 'phenological curve' of an insect species for instance).

These last two types of detection heterogeneity are not (yet) allowed to be species-specific.

Usage

```
simDCM(nspecies = 50, nsites = 100, nsurveys = 3, nyears = 10,
  mean.psi1 = 0.4, sig.lpsi1 = 1, mu.beta.lpsi1 = 0, sig.beta.lpsi1 = 0,
  range.mean.phi = c(0.8, 0.8), sig.lphi = 1, mu.beta.lphi = 0,
  sig.beta.lphi = 0, range.mean.gamma = c(0.2, 0.2), sig.lgamma = 1,
  mu.beta.lgamma = 0, sig.beta.lgamma = 0, range.mean.p = c(0.5, 0.5),
  sig.lp = 1, mu.beta.lp = 0, sig.beta.lp = 0, range.beta1.survey = c(0, 0),
  range.beta2.survey = c(0, 0), trend.sd.site = c(0, 0),
  trend.sd.survey = c(0, 0), show.plot = TRUE, verbose = TRUE)
```

Arguments

nspecies	number of species (typically called N in AHM book)
nsites	number of sites (M).
nsurveys	number of replicate surveys within a year (J).
nyears	number of years (T).
mean.psi1	average (across all species in the community) of the intercept of occupancy probability in first year.
sig.lpsi1	sd of the normal distribution from which species-specific occupancy intercepts are drawn (centered on logit(mean.psi1)), on logit scale.
mu.beta.lpsi1	community mean of the coefficients of the covariate in probabilities of initial occupancy: the probability-scale mean of the normal distribution from which the species-specific coefficients are drawn.
sig.beta.lpsi1	sd of the normal distribution from which species-specific slopes are drawn (centered on mu.beta.lpsi1).
range.mean.phi	bounds of uniform distribution from which the average (across species) annual intercept of persistence is drawn.
sig.lphi	sd of the normal distribution from which species-specific persistence intercepts are drawn (centered on logit(mean.phi), which are year-specific), on logit scale.
mu.beta.lphi	community mean of the coefficients of the covariate in probabilities of persistence: the probability-scale mean of the normal distribution from which the species-specific coefficients are drawn.
sig.beta.lphi	sd of the normal distribution from which species-specific persistence slopes are drawn (centered on mu.beta.lphi).
range.mean.gamma	
	bounds of uniform distribution from which the average (across species) annual intercept of colonization is drawn.
sig.lgamma	sd of the normal distribution from which species-specific colonization intercepts are drawn (centered on logit(mean.gamma), which are year-specific), on logit scale.
mu.beta.lgamma	community mean of the coefficients of the covariate in probabilities of colonization: the probability-scale mean of the normal distribution from which the species-specific coefficients are drawn.
sig.beta.lgamma	
	sd of the normal distribution from which species-specific colonization slopes are drawn (centered on mu.beta.lgamma).
range.mean.p	bounds of uniform distribution from which the average (across species) annual intercept of p is drawn.
sig.lp	sd of the normal distribution from which species-specific detection intercepts are drawn (centered on logit(mean.p), which are year-specific), on logit scale.
mu.beta.lp	community mean of the coefficients of the covariate in probabilities of detection: the probability-scale mean of the normal distribution from which the species-specific coefficients are drawn.

sig.beta.lp	sd of the normal distribution from which species-specific detection slopes are drawn (centered on mu.beta.lp).
range.beta1.survey	
	the range of the annual variation in the linear effect of survey (i.e., of surveys 1-12 if nsurveys = 12) on the product of availability and detection.
range.beta2.survey	
	the range of the annual variation in the quadratic effect of survey (i.e., of surveys 1-12 if nsurveys = 12) on the product of availability and detection.
trend.sd.site	sd of normal distribution to model logit-normal noise in p at the site level in the first and the last year of the simulation, with values for intermediate years interpolated linearly.
trend.sd.survey	
	sd of normal distribution to model logit-normal noise in p at the occasion level, in the first and the last year, with values for intermediate years interpolated linearly.
show.plot	if TRUE, plots are produced. Set this to FALSE when running many simulations.
verbose	if TRUE, output will be written to the Console.

Value

A list with the values of the input arguments and the following additional elements:

Xpsi1	Site covariate for psi1, a nsites x 1 matrix
Xphi	Yearly-site covariate for phi, a nsites x nyears matrix
Xgamma	Yearly-site covariate for gamma, a nsites x nyears matrix
Xp	Observation covariate for p, a nsites x nsurveys x nyears array
beta0.lpsi	initial (logit-scale) occupancy intercept for each species in the community, a vector of length nspecies
beta1.lpsi	initial (log-scale) occupancy slope on Xpsi1 for each species in the community, a vector of length nspecies
psi	occupancy probability per site, year and species, a nsites x nyears x nspecies array
mean.phi	mean persistence (across species) intercept for each interval, a vector of length (nyears - 1)
mean.gamma	mean colonization (across species) intercept for each interval, a vector of length (nyears - 1)
eps.lphi	additive species effects in logit(phi) intercept, a vector of length nspecies
eps.lgamma	additive species effects in logit(gamma) intercept, a vector of length nspecies
beta0.lphi	logit-scale persistence intercepts for each species in community, a nspecies x (nyears - 1) matrix
beta0.lgamma	logit scale colonization intercepts for each species in the community, a nspecies x (nyears - 1) matrix
beta1.lphi	slope of logit(phi) on Xphi for each species in the community, a vector of length nspecies

beta1.lgamma	slope of logit(gamma) on Xgamma for each species in the community, a vector of length nspecies
phi	probability of persistence for each site, yearly interval and species, a nsites x (nyears-1) x nspecies array
gamma	probability of colonization for each site, yearly interval and species, a nsites x (nyears-1) x nspecies array
mean.p	mean detection (across species) intercept for each year, a vector of length nyears
eps.lp	additive species effects in logit(p) intercept, a vector of length nspecies
beta0.lp	species- and site-specific intercepts in the linear predictor for p, a nspecies x nyears matrix
beta1.lp	species specific slopes of logit(p) on Xp, a vector of length nspecies
beta1	linear effect of the occasion number on detection probability, a vector of length nyears
beta2	quadratic effect of the occasion number on detection probability, a vector of length nyears
sd.site	standard deviation of the zero-mean normal distribution from which additional, site-level detection heterogeneity is simulated, a vector of length nyears
sd.survey	standard deviation of the zero-mean normal distribution from which additional, occasion-level detection heterogeneity is simulated,, a vector of length nyears
eps1	additive site random effects tht generate unstructured site-level detection heterogeneity, a vector of length nsites
eps2	additive occasion random effects tht generate unstructured site-level detection heterogeneity, a vector of length nsurveys
n.occ	Number of occupied sites, a nyears x nspecies matrix
psi.fs	Finite-sample occupancy proportion, a nyears x nspecies matrix
mean.psi	Average psi over sites, a nyears x nspecies matrix
z.obs	Observed value of z matrix, a nsites x nyears x nspecies array
n.occ.obs	Observed number of occupied sites, a nyears x nspecies matrix
psi.obs	Observed occupancy (finite sample), a nyears x nspecies matrix
nyears.pres	Number of years when species present, a vector of length nspecies
nspecies.pres	Number of species ever present, scalar
nyears.det	Number of years when species detected, a vector of length nspecies
nspecies.det	Number of species ever detected, scalar
z	True value of z matrix (ie, presence/absence), a nsites x nyears x nspecies array
p	Probability of detection, a nsites x nsurveys x nyears x nspecies array
y	Observed detection history, a nsites x nsurveys x nyears x nspecies array of 0/1

Author(s)

Marc Kéry & Andy Royle

References

Kéry, M. & Royle, J.A. (2020) *Applied Hierarchical Modeling in Ecology* AHM2 - 5.2.

Examples

```
# Simulate a data set with the default arguments and look at the structure of the output:
tmp <- simDCM()
str(tmp)


# Default arguments, without plots
str(data <- simDCM(show.plot = FALSE))
# More examples:
str(data <- simDCM(nspecies = 200)) # More species (looks great)
str(data <- simDCM(nspecies = 1))   # A single species (ha, works !)
str(data <- simDCM(nsites = 267))   # More sites
str(data <- simDCM(nsites = 1))     # A single site
str(data <- simDCM(nsurveys = 10))  # More visits
str(data <- simDCM(nyears = 25))    # More years
str(data <- simDCM(nyears = 2))     # Just two years
try(data <- simDCM(nyears = 1))     # A single year ... error

# No species heterogeneity in parameters of initial occupancy
str(data <- simDCM(sig.lpsi1 = 0, sig.beta.lpsi1 = 0))
# No species heterogeneity in parameters of persistence
str(data <- simDCM(sig.lphi = 0, sig.beta.lphi = 0))
# No species heterogeneity in parameters of colonisation
str(data <- simDCM(sig.lgamma = 0, sig.beta.lgamma = 0))
# No species heterogeneity in parameters of detection
str(data <- simDCM(sig.lp = 0, sig.beta.lp = 0))
# No annual variation in rates
str(data <- simDCM(range.mean.phi = c(0.8, 0.8), range.mean.gamma = c(0.3, 0.3),
  range.mean.p = c(0.6, 0.6)))

# Function arguments that lead to much structure (no zero arguments)
str(data <- simDCM(nspecies = 200, nsites = 267, nsurveys = 3, nyears = 25,
  mean.psi1 = 0.4, sig.lpsi1 = 3, mu.beta.lpsi1 = 1, sig.beta.lpsi1 = 3,
  range.mean.phi = c(0.5, 1), sig.lphi = 3, mu.beta.lphi = 1,
  sig.beta.lphi = 3, range.mean.gamma = c(0, 0.5),
  sig.lgamma = 3, mu.beta.lgamma = -1, sig.beta.lgamma = 3,
  range.mean.p = c(0.1, 0.9), sig.lp = 3, mu.beta.lp = 1, sig.beta.lp = 3,
  range.beta1.survey = c(-2, -0.5), range.beta2.survey = c(0, 2),
  trend.sd.site = c(0, 0), trend.sd.survey = c(0, 0), show.plot = TRUE))

# Not every occurring species will be detected
set.seed(1)
str(data <- simDCM(nspecies = 200, nsites = 20, nsurveys = 2, nyears = 10,
  mean.psi1 = 0.1, sig.lpsi1 = 5,
  range.mean.phi = c(0.3, 0.3), sig.lphi = 5,
  range.mean.gamma = c(0.1, 0.1), sig.lgamma = 5,
  range.mean.p = c(0.1, 0.1), sig.lp = 5) )
```

```
# Pull out data from species 5
ysp5 <- data$y[,,,5]

# Pull out data from year 1
yyr1 <- data$y[,,1,]
```

| simDemoDynocc | *Simulate data under a demographic dynamic occupancy model* |

Description

Function to simulate detection/nondetection data under a variant of the demographic occupancy (or 'local survival') model of Roth & Amrhein (2010). Data are simulated in an 'unconditional' manner, i.e., for each site from first to last year. All parameters can be made year-dependent by specification of a range within which annual values will be drawn from uniform distributions.

Usage

```
simDemoDynocc(nsites = 100, nyears = 10, nvisits = 5, psi1 = 0.6,
    range.phi = c(0.2, 0.9), range.r = c(0, 0.4), range.p = c(0.1, 0.9),
    show.plot=TRUE)
```

Arguments

nsites	Number of sites.
nyears	Number of years (or 'seasons', as they are somewhat confusingly often also called in the occupancy literature).
nvisits	Number of replicate surveys (= occasions) within a year.
psi1	occupancy probability in first year.
range.phi	bounds of uniform distribution from which annual local probability of persistence is randomly drawn.
range.r	bounds of uniform distribution from which annual local probability of colonization is randomly drawn.
range.p	bounds of uniform distribution from which annual probability of detection is randomly drawn.
show.plot	If TRUE, plots of results are displayed; set to FALSE if running many simulations.

Value

A list with the values of the arguments input and the following additional elements:

phi	persistence for each interval, a vector of length nyears - 1
r	colonization for each interval, a vector of length nyears - 1

p	detection probability for each year, a vector of length nyears
z	true occurrence state, a nsites x nyears matrix of 0/1
y	the observed detection history, a nsites x nvisits x nyears array
f	year of first detection, a vector of length nsites
nocc.true	the true number of occupied sites, a vector of length nyears
nocc.true	the observed number of occupied sites, a vector of length nyears

Author(s)

Marc Kéry & Andy Royle

References

Roth, T. & Amrhein, V. (2010), Estimating individual survival using territory occupancy data on unmarked animals. *Journal of Applied Ecology*, 47: 386-392.

Kéry, M. & Royle, J.A. (2020) *Applied Hierarchical Modeling in Ecology* AHM2 - 4.

Examples

```
# Generate data with the default arguments and look at the structure:
str(data <- simDemoDynocc() )              # Implicit defaults
str(data <- simDemoDynocc(psi1 = 1))       # All sites initially occupied
str(data <- simDemoDynocc(nsites = 1000))  # Plenty more sites
str(data <- simDemoDynocc(nyears = 100))   # Plenty more years
str(data <- simDemoDynocc(nvisits = 20))   # Plenty more visits
str(data <- simDemoDynocc(range.phi = c(0.8, 0.8))) # Constant survival
str(data <- simDemoDynocc(range.phi = c(0.2,0.3), range.r = c(0,0.2))) # Decline
str(data <- simDemoDynocc(range.phi = c(0.8,1), range.r = c(0.5,0.7))) # Increase
str(data <- simDemoDynocc(nvisits = 1))      # Single visit
str(data <- simDemoDynocc(range.p = c(1,1))) # Perfect detection
```

---

simDM	*Simulate count data under the dynamic N-mixture model of Dail-Madsen*

Description

Simulation for multiple-visit data (from pcountOpen help file). `simDM0` has no covariates while `simDM` allows for covariates. Both functions assume constant time intervals between primary periods.

Usage

```
simDM0(nsites = 50, nsurveys = 3, nyears = 5,
  lambda = 4, gamma = 1.5, phi = 0.8, p = 0.7, show.plots=TRUE)

simDM(nsites = 50, nsurveys = 3, nyears = 5,
  mean.lambda = 4, mean.gamma.rel = 0.5,
  mean.phi = 0.8, mean.p = 0.7,
  beta.lam = 1, beta.gamma = 1, beta.phi = -1, beta.p = -1,
  show.plots=TRUE)
```

Arguments

nsites	number of sites.
nsurveys	number of replicate (secondary) samples within period of closure.
nyears	number of primary samples: years, seasons etc.
lambda	Initial expected abundance.
gamma	recruitment rate.
phi	apparent survival rate.
p	detection probability.
mean.lambda	Initial expected abundance at cov.lam = 0.
mean.gamma.rel	recruitment rate at cov.gamma = 0.
mean.phi	apparent survival rate at cov.phi = 0.
mean.p	detection probability at cov.p = 0.
beta.lam	the slope of parameter lambda (link transformed) on the cov.lam covariate
beta.gamma	the slope of parameter gamma (link transformed) on the cov.gamma covariate
beta.phi	the slope of parameter phi (link transformed) on the cov.phi covariate
beta.p	the slope of parameter p (link transformed) on the cov.p covariate
show.plots	if TRUE, plots are produced. Set this to FALSE when running many simulations.

Value

For simDM0, a list with the values of the input arguments and the following additional elements:

N	true number of individuals, nsites x nyears
S	number of survivors, nsites x (nyears-1)
R	number of recruits, nsites x (nyears-1)
y	number detected, nsites x nyears x nsurveys
yy	number detected as a matrix, nsites x (nyears*nsurveys)

simDM0 has the following additional elements:

cov.lam	covariate for lambda generated from Uniform(-1, 1), nsites vector
cov.gamma	covariate for gamma generated from Uniform(-1, 1), nsites vector
cov.phi	covariate for phi generated from Uniform(-1, 1), nsites vector
cov.p	covariate for p generated from Uniform(-1, 1), nsites x nyears x nsurveys
ccov.p	cov.p formatted as a matrix, nsites x (nyears*nsurveys)

Author(s)

Marc Kéry & Andy Royle, based in part on code written by Richard Chandler.

References

Dail, D. & Madsen, L. (2011) Models for estimating abundance from repeated counts of an open metapopulation. *Biometrics*, 67, 577-587.

Kéry, M. & Royle, J.A. (2020) *Applied Hierarchical Modeling in Ecology* AHM2 - 2.5.

Examples

```
# Simulate a data set with the default arguments and look at the structure of the output:
tmp0 <- simDM0()
str(tmp0)

tmp <- simDM()
str(tmp)
```

---

simDSM   *Simulate line transect data for density surface modeling*

Description

The function generates a population represented as a binomial point pattern in a heterogeneous landscape with density a function of the covariate Habitat. Data for multiple line transect surveys using a wiggly transect are then simulated, and the pixel IDs for the activity centers of detected individuals returned.

To recreate the data sets used in the book with R 3.6.0 or later, include sample.kind="Rounding" in the call to set.seed. This should only be used for reproduction of old results.

Usage

```
simDSM(X, Ntotal = 400, sigma = 0.65, beta1 = 1.0,
    nsurveys = 2, xlim = c(-0.5, 3.5), ylim = c(-0.5, 4.5), show.plot = TRUE)
```

Arguments

X	a 2-column matrix with coordinates of regularly spaced points along the transect line; see Examples.
Ntotal	the true total number of individuals in the study area.
sigma	scale parameter for the half-normal detection function.
beta1	coefficient for the relationship between the Habitat covariate and population density.
nsurveys	the number of replicate surveys along the transect.
xlim, ylim	the extent of the (rectangular) study area
show.plot	if TRUE, summary plots are displayed.

Value

A list with the values of the input arguments and the following additional elements:

Habitat	a vector for the habitat covariate for each pixel
Habgrid	a 2-column matrix with the coordinates of center of each pixel
nPix	the number of pixels in the study area
N	true number of activity centers in each pixel
U	a 2-column matrix with the locations of each activity center
nind	the number of individuals detected at least once
pixel	a nind x nsurvey matrix with the pixel ID for the activity center or NA if the individual was not detected on the survey

Author(s)

Marc Kéry, Andy Royle & Mike Meredith

References

Kéry, M. & Royle, J.A. (2020) *Applied Hierarchical Modeling in Ecology* AHM2 - 11.10.

Examples

```
# Run the function with default values and look at the output
library(AHMbook)
data(wigglyLine)
points <- sp::SpatialPoints( wigglyLine )
sLine <- sp::Line(points)
regpoints <- sp::spsample(sLine, 100, type = "regular")
str(simDSM(X = regpoints@coords))

# Generate the data set used in AHM2 11.10
# RNGkind(sample.kind = "Rounding") # run this for R >= 3.6.0
set.seed(2027, kind = "Mersenne-Twister")
tmp <- simDSM(X = regpoints@coords) # Produces Fig 11.15 in the book
```

---

simDynocc	*Simulate detection/nondetection data under a wide variety of non-spatial dynamic occupancy models*

Description

Function to simulate detection/nondetection data under a general dynamic site-occupancy model, including:

* Annual variation in the probabilities of patch persistence, colonization and detection is specified by the bounds of a uniform distribution.

* One covariate is allowed to affect each parameter: a site covariate for psi1, site-by-year covariates for phi and gamma, and an observational covariate for p. Covariates are generated internally from uniform(-2, 2) distributions.

* Additional heterogeneity among sites in persistence and colonization or both.

* Additional detection heterogeneity at the site-, the survey, or the site-by-survey level, with the possibility of a temporal trend in this heterogeneity over the years. E.g., an annual trend in detection heterogeneity at the site or the survey level is specified by the first and second value, which correspond to the heterogeneity in the first and the last year. Hence, trend.sd.site = c(0, 1) will result in a linear trend in the magnitude of site heterogeneity in detection from 0 in the first year to 1 in the last year.

* Additional detection heterogeneity that varies over the survey (= occasion) according to a quadratic effect of occasion number (to model the typical phenology of an insect species for instance).

* Simulation of data under a BACI (before-after-control-impact) design, where some event happens in a given year and *reduces* phi or gamma by a stated percentage (only reductions, no increases allowed!)

Usage

```
simDynocc(nsites = 250, nyears = 10, nsurveys = 3, year.of.impact = NA,
  mean.psi1 = 0.4, beta.Xpsi1 = 0,
  range.phi = c(0.5, 1), sd.lphi.site = 0, impact.phi = 0, beta.Xphi = 0,
  range.gamma = c(0, 0.5), sd.lgamma.site = 0, impact.gamma = 0, beta.Xgamma = 0,
  sd.lphi.lgamma.site = 0,
  range.p = c(0.1, 0.9), beta.Xp = 0,
  range.beta1.survey = c(0, 0), range.beta2.survey = c(0, 0),
  trend.sd.site = c(0, 0), trend.sd.survey = c(0, 0),
  trend.sd.site.survey = c(0, 0), show.plot = TRUE)
```

Arguments

nsites   Number of sites.

nyears   Number of years (or 'seasons').

nsurveys   Number of replicate surveys (= occasions) within a year.

year.of.impact Year in which an impact happens that can affect phi and gamma (for BACI design), NA if no impact occurs; for the BACI design, nyears must be greater than 2, and year of impact cannot be the first or last year.

mean.psi1   average occupancy probability in first year.

beta.Xpsi1   coefficient of environmental covariate in probability of initial occupancy.

range.phi   bounds of uniform distribution from which annual probability of persistence is randomly drawn.

sd.lphi.site   SD of random site effect on persistence on the logit scale drawn from a normal distribution with mean zero.

impact.phi   negative effect in percent on annual phi (e.g., impact.phi = 20 means a 20% reduction in phi); ignored if year.of.impact = NA.

beta.Xphi   coefficients of environmental covariate in probability of persistence.

range.gamma	bounds of uniform distribution from which annual probability of colonization is randomly drawn.
sd.lgamma.site	SD of random site effect on colonization on the logit scale drawn from a normal distribution with mean zero.
impact.gamma	negative effect in percent on annual gamma (e.g., impact.gamma = 20 means a 20% reduction in gamma); ignored if year.of.impact = NA.
beta.Xgamma	coefficient of environmental covariate in probability of colonization.
sd.lphi.lgamma.site	
	SD of random site effect on persistence AND colonization on the logit scale drawn from a normal distribution with mean zero.
range.p	bounds of uniform distribution from which annual probability of detection is randomly drawn.
beta.Xp	coefficients of environmental covariate in probability of detection.
range.beta1.survey	
	bounds of the uniform distribution from which the annual variation in the linear effect of the survey occasion (i.e., of survey 1-12 with nsurveys = 12) on detection (= product of availability and perceptibility) is randomly drawn.
range.beta2.survey	
	the same for the quadratic effect of survey occasion.
trend.sd.site	initial and final values of sd of normal distribution to model logit-normal noise in p at the site level; a linear trend is assumed over time; if the two values are the same, a constant value is assumed.
trend.sd.survey	
	initial and final values of sd of normal distribution to model logit-normal noise in p *only* at the 'survey' level; if they are different, a linear trend is assumed over time.
trend.sd.site.survey	
	initial and final values of sd of normal distribution to model logit-normal noise in p at the site/year/survey = 'survey' level; if they are different, a linear trend is assumed over time.
show.plot	If TRUE, plots of results are displayed; set to FALSE if running many simulations.

Value

A list with the values of the arguments input and the following additional elements:

impact	a 0/1 vector of length (nyears-1) indicating if an impact applies to the interval
BACI.effect.phi	
	reduction in persistence due to impact, a vector of length nyears - 1
BACI.effect.gamma	
	reduction in colonization due to impact, a vector of length nyears - 1
beta1	linear effect of occasion on the product of availability and detection, a vector of length nyears
beta2	quadratic effect of occasion on the product of availability and detection, a vector of length nyears

mean.phi	mean persistence for each interval before application of any BACI effect, a vector of length nyears - 1
mean.gamma	mean colonization for each interval before application of any BACI effect, a vector of length nyears - 1
mean.p	mean detection probability for each year, a vector of length nyears
psi	annual occupancy for each site, a nsites x nyears matrix
mean.psi	average occupancy over sites, a vector of length nyears
n.occ	number of occupied sites, a vector of length nyears
psi.fs	finite-sample occupancy proportion, a vector of length nyears
psi.app	apparent occupancy over sites, a vector of length nyears
z	true occurrence state, a nsites x nyears matrix of 0/1
phi	persistence, a nsites x nyears-1 matrix
gamma	colonization, a nsites x nyears-1 matrix
p	detection probability, a nsites x nsurveys x nyears array
y	the observed detection history, a nsites x nsurveys x nyears array
Xpsi1	covariate affecting initial occupancy, a vector of length nsites
Xphi	covariate affecting persistence, a nsites x nyears matrix
Xgamma	covariate affecting colonization, a nsites x nyears matrix
Xp	covariate affecting probability of detection, a nsites x nsurveys x nyears array
eps.lphi.site	site random effects on persistence, a vector of length nsites
eps.lgamma.site	site random effects on colonization, a vector of length nsites
eps.lphi.lgamma.site	site random effects on persistence and colonization, a vector of length nsites
eps1	site random effects on detection, a vector of length nsites
eps2	the survey random effects on detection, a vector of length nsurveys
eps3	the site/survey/year random effects on detection, a nsites x nsurveys x nyears array

Author(s)

Marc Kéry & Andy Royle

References

Kéry, M. & Royle, J.A. (2020) *Applied Hierarchical Modeling in Ecology* AHM2 - 4.

Examples

```
# Generate data with the default arguments and look at the structure:

tmp <- simDynocc()
str(tmp)

# no annual variation in the parameters
str(data <- simDynocc(nsites = 250, nsurveys = 3, nyears = 10, mean.psi1 = 0.6,
    range.phi = c(0.7, 0.7), range.gamma = c(0.3, 0.3), range.p = c(0.5, 0.5)))
# a fully time-dependent model (with p constant within each primary period)
str(data <- simDynocc(mean.psi1 = 0.6, range.phi = c(0.5, 0.8),
   range.gamma = c(0.1, 0.5), range.p = c(0.1, 0.9)) )
# a time-constant model with four different covariates affecting the four parameters
str(data <- simDynocc(mean.psi1 = 0.6, beta.Xpsi1 = 1,
  range.phi = c(0.6, 0.6), beta.Xphi = 2, range.gamma = c(0.3, 0.3),
  beta.Xgamma = 2, range.p = c(0.2, 0.2), beta.Xp = -2) )
# seasonal variation in detection probability
str(data <- simDynocc(nsurveys = 12, mean.psi1 = 0.6,
  range.phi = c(0.6, 0.6), range.gamma = c(0.3, 0.3),
  range.p = c(0.5, 0.5), range.beta1.survey = c(-0.3, 0.4),
  range.beta2.survey = c(0, -0.7)) )
# now both yearly variation and effects of all covariates (including survey)
str( data <- simDynocc(mean.psi1 = 0.6, beta.Xpsi1 = 1,
   range.phi = c(0.6, 1), beta.Xphi = 2, range.gamma = c(0, 0.2),
   beta.Xgamma = 2, range.p = c(0.1, 0.9), beta.Xp = -2,
   range.beta1.survey = c(-0.4, 0.5), range.beta2.survey = c(0, -0.8)) )

# To add detection heterogeneity at the site level, you can do this:
str(data <- simDynocc(trend.sd.site = c(3, 3)) ) # No time trend
str(data <- simDynocc(trend.sd.site = c(1, 3)) ) # With time trend

# To add detection heterogeneity at the level of the survey, you can do this:
str(data <- simDynocc(trend.sd.survey = c(3, 3)) ) # No time trend
str(data <- simDynocc(trend.sd.survey = c(1, 3)) ) # With time trend

# To add detection heterogeneity at the level of the individual visit, you can do this:
str(data <- simDynocc(trend.sd.site.survey = c(3, 3)) ) # No trend
str(data <- simDynocc(trend.sd.site.survey = c(1, 3)) ) # With trend

# To simulate data under a BACI design, where an impact happens in year 10
str(data <- simDynocc(nsites = 250, nsurveys = 3, nyears = 20, year.of.impact = 10,
   impact.phi = 80, impact.gamma = 50) )

# And data where there is no detection error (i.e., with p = 1):
str( data <- simDynocc(range.p = c(1, 1)) )
```

---

simDynoccSpatial	*Simulate data under a dynamic occupancy model with spatial auto-correlation*

Description

Function to simulate detection/nondetection data under a general dynamic site-occupancy model with autocorrelation, including:

* Annual variation in the probabilities of patch persistence, colonization and detection is specified by the bounds of a uniform distribution.

* One covariate is allowed to affect a parameter: a site covariate for psi1, site-by-year covariates for phi and gamma, and an observational covariate for p. Covariates are generated internally from uniform(-2, 2) distributions.

* Additional heterogeneity among sites in persistence and colonization or both.

* Additional detection heterogeneity at the site or survey level, with the possibility of a temporal trend in this heterogeneity over the years. E.g., an annual trend in detection heterogeneity at the site or the survey level is specified by the first and second value, which correspond to the heterogeneity in the first and the last year. Hence, trend.sd.site = c(0, 1) will result in a linear trend in the magnitude of site heterogeneity in detection from 0 in the first year to 1 in the last year.

* A single, spatially structured covariate for habitat suitability may affect all parameters via coefficient beta.XAC; for a biologically reasonable way, choose coefficients with the same sign for all 4 (mediated by underlying density). That spatial covariate is simulated as a Gaussian random field with negative exponential correlation function with 'range parameter' theta.XAC.

* Autologistic effects (beta.Xautolog) in persistence and colonization probability can be chosen, which fits a logistic regression of these parameters on the proportion of occupied neighboring cells (in a queen's or 2nd order neighborhood) during the previous time step.

* Additional detection heterogeneity can be introduced at the site- or the individual survey level, with the possibility of a temporal trend in this heterogeneity. For instance, an annual trend in detection heterogeneity at the site or the survey level is specified by the value in the first and the last year. Hence, trend.sd.site = c(0, 1) will result in a linear trend in the magnitude of site heterogeneity in detection from 0 in the first year to 1 in the last year.

Usage

```
simDynoccSpatial(side = 50, nyears = 10, nsurveys = 3,
     mean.psi1 = 0.4, beta.Xpsi1 = 0,
     range.phi = c(0.8, 0.8), beta.Xphi = 0,
     range.gamma = c(0.1, 0.1), beta.Xgamma = 0,
     range.p = c(0.4, 0.4), beta.Xp = 0,
     theta.XAC = 5000, beta.XAC = c(0, 0, 0, 0), beta.Xautolog = c(0, 0),
     trend.sd.site = c(0, 0), trend.sd.survey = c(0, 0),
     seed.XAC = NA, seed = NULL, ask.plot = TRUE, verbose = TRUE)
```

Arguments

side	side length of square simulation area; the number of sites, or cells, M = side^2.
nyears	Number of years (or 'seasons').
nsurveys	Number of replicate surveys (= occasions) within a year.
mean.psi1	average occupancy probability in first year.
beta.Xpsi1	coefficient of environmental covariate in probability of initial occupancy.

range.phi	bounds of uniform distribution from which annual probability of persistence is randomly drawn.
beta.Xphi	coefficient of environmental covariate in probability of persistence.
range.gamma	bounds of uniform distribution from which annual probability of colonization is randomly drawn.
beta.Xgamma	coefficient of environmental covariate in probability of colonization.
range.p	bounds of uniform distribution from which probability of detection is randomly drawn.
beta.Xp	coefficient of environmental covariate in probability of detection.
theta.XAC	'range parameter' of a covariate with exponential spatial correlation (i.e., a Gaussian random field is used as an environmental covariate); must be > 0.
beta.XAC	vector of coefficients of that field for the 4 model parameters: psi1, phi, gamma, and p (in that order).
beta.Xautolog	vector of coefficients of autologistic covariate in the following order: persistence (phi), colonization (gamma); the autocovariate is computed at every season as the proportion of occupied cells in a queen's neighborhood around each cell.
trend.sd.site	initial and final values of sd of normal distribution to model logit-normal noise in p at the site level; a linear trend is assumed over time; if the two values are the same, a constant value is assumed.
trend.sd.survey	
	initial and final values of sd of normal distribution to model logit-normal noise in p at the 'survey' level; if they are different, a linear trend is assumed over time.
seed.XAC	the seed to be used for simulation of the spatially structured covariate for habitat suitability.
seed	the seed to be used for simulation of values apart from the spatially structured covariate for habitat suitability; using the same value for seed.XAC with different values for seed allows generation of different data sets with the same habitat suitability covariate.
ask.plot	If TRUE, pause between plots of results; set to FALSE if running many simulations.
verbose	if TRUE, output will be written to the Console.

Value

A list with the values of the arguments input and the following additional elements:

M	scalar, total number of pixels in the study area, side^2
grid	2-column matrix, x and y coordinates of each pixel
amatrix	M x M matrix, [i,j] = 1 if cells i and j are neighbors, 0 otherwise
Xpsi1	side x side matrix, value of covariate affecting initial occupancy
Xphi	side x side x nyears array, value of covariate affecting persistence
Xgamma	side x side x nyears array, value of covariate affecting colonization

Xp	side x side x nsurveys x nyears array, value of covariate affecting detection
XAC	side x side matrix, the spatially correlated covariate
Xauto	side x side x nyears array, the autocovariate, the proportion of neighboring cells occupied
Xautoobs	side x side x nyears array, the observed autocovariate, the proportion of neighboring cells where the species was detected
sd.site	vector nyears, year-specific values of SD of Gaussian random site effects in p
sd.survey	vector nyears, year-specific values of SD of Gaussian random survey effects in p
mean.phi	vector nyears-1, mean persistence for each interval
mean.gamma	vector nyears-1, mean colonization for each interval
mean.p	vector nyears, mean detection probability for each year
psi	side x side x nyears array, probability of occupancy for each site and year
mean.psi	vector nyears, mean occupancy over cells for each year
psi.app	vector nyears, apparent occupancy, proportion of cells where species detected
z	side x side x years array, true occupancy state
zobs	side x side x years array, observed occupancy state
nocc	vector nyears, true number of occupied cells
nocc.obs	vector nyears, number of cells where the species was detected
phi	side x side x nyears-1 array, probability of persistence in each interval between years
gamma	side x side x nyears-1 array, probability of colonization in each interval between years
p	side x side x nsurveys x nyears array, probability of detection
y	side x side x nsurveys x nyears array, the observed detection history
umf	an unmarked data frame object with the simulated data

Author(s)

Marc Kéry & Andy Royle

References

Kéry, M. & Royle, J.A. (2020) *Applied Hierarchical Modeling in Ecology* AHM2 - 9.

Examples

```
# Generate data with the default arguments and look at the structure:
tmp <- simDynoccSpatial()
str(tmp)

  # rather slow
# Large grid, seed.XAC = 88 gives a cool pattern
str(tmp <- simDynoccSpatial(side = 100, seed.XAC = 88))
```

```
# Longer series and no browsing through plots
str(tmp <- simDynoccSpatial(nyears = 100, ask.plot = FALSE))

# Dynamic system (high variance between years)
str(tmp <- simDynoccSpatial(range.phi = c(0, 0.8), range.gamma = c(0, 0.6)))

# Time-variation in all potentially time-varying parameters
str(tmp <- simDynoccSpatial(range.phi = c(0.5, 1), range.gamma = c(0, 0.5),
  range.p = c(0.1, 0.9)))

# Strong spatial autocorrelation via a spatially structured covariate
str(tmp <- simDynoccSpatial(beta.XAC = c(1, 1, 1, 1), beta.Xautolog = c(0, 0)))

# Strong spatial autocorrelation via a spatially structured covariate
str(tmp <- simDynoccSpatial(beta.XAC = c(1, 1, 1, 1), beta.Xautolog = c(2, 2)))

# Strong spatial autocorrelation via a spatially structured covariate
# but now effect of that spatial field only on colonization and persistence
str(tmp <- simDynoccSpatial(beta.XAC = c(0, 1, 1, 0), beta.Xautolog = c(0, 0)))

# Can it ever go extinct ???
str(tmp <- simDynoccSpatial(side = 20, nyears = 20, mean.psi1 = 0.01,
  range.phi = c(0.1, 0.1), range.gamma = c(0.1, 0.1), beta.Xautolog = c(0, 0)))

# Can it ever go extinct ???   -------> Yes
str(tmp <- simDynoccSpatial(side = 20, nyears = 10, mean.psi1 = 0.01,
  range.phi = c(0.3, 0.3), range.gamma = c(0, 0), beta.Xautolog = c(0, 0)), seed = 24)

# Evolution of number of occupied quads
apply(tmp$z, 3, sum)
# [1] 2 0 0 0 0 0 0 0 0 0
```

---

simExpCorrRF   *Create a Gaussian random field with negative exponential correlation*

Description

Function creates Gaussian random field with negative exponential correlation and visualizes correlation and random field.

Usage

```
simExpCorrRF(variance = 1, theta = 1, size = 50, seed = NA, show.plots = TRUE)
```

Arguments

variance   variance of field.

theta	parameter governing spatial correlation (=1/phi) ("large theta means high correlation") Note that RMexp (which is used internally) is specified in terms of phi = 1/theta.
size	number of pixels along each side of the square site.
seed	used to generate reproducible output.
show.plots	if TRUE, the result will be displayed.

Value

A list with the values of the input arguments and the following additional elements:

| field | the random field variable, a vector of length size^2 |
| grid | the grid corresponding to field |

Author(s)

Marc Kéry & Andy Royle

References

Kéry, M. & Royle, J.A. (2020) *Applied Hierarchical Modeling in Ecology* AHM2 - 9.

Examples

```
# Run the function with default values and look at the output
str(tmp <- simExpCorrRF())
```

| simHDS | *Simulate data under hierarchical distance sampling protocol (line or point)* |

Description

Function simulates hierarchical distance sampling (HDS) data under either a line or a point transect protocol.

Usage

```
simHDS(type=c("line", "point"), nsites = 100, mean.lambda = 2, beta.lam = 1,
  mean.sigma = 1, beta.sig = -0.5, B = 3, discard0 = TRUE, show.plot = TRUE)
```

Arguments

type	type of transect, "line" or "point".
nsites	Number of sites (spatial replication)
mean.lambda	the expected value of lambda when the habitat covariate = 0; the intercept of the log-linear regression for lambda is log(mean.lambda).
beta.lam	the slope of the log-linear regression for lambda on a habitat covariate.
mean.sigma	the expected value of the scale parameter of the half-normal detection function when the wind speed = 0; the intercept of the log-linear regression for sigma is log(mean.sigma).
beta.sig	the slope of log-linear regression of scale parameter of the half-normal detection function on wind speed
B	the strip half-width or circle radius
discard0	If TRUE, subset to sites at which individuals were captured. You may or may not want to do this depending on how the model is formulated so be careful.
show.plot	choose whether to show plots or not. Set to FALSE when using function in simulations.

Value

A list with the values of the arguments entered and the following additional elements:

data	simulated distance sampling data: a matrix with a row for each individual detected and 5 columns: site ID, status (1 if captured), x and y coordinates (NA for line transects), distance from the line or point; if discard0 = FALSE, sites with no detections will appear in the matrix with NAs in columns 2 to 5.
habitat	simulated habitat covariate
wind	simulated detection covariate
N	simulated number of individuals at each site
N.true	for point counts, the simulated number of individuals within the circle sampled

Author(s)

Marc Kéry & Andy Royle

References

Kéry, M. & Royle, J.A. (2016) *Applied Hierarchical Modeling in Ecology* AHM1 - 8.5.1.

Examples

```
# Simulate a data set with the default arguments and look at the structure of the output:
set.seed(123)
tmp <- simHDS()
str(tmp)
head(tmp$data)
```

```
tmp <- simHDS("point", discard0=FALSE)
str(tmp)
head(tmp$data, 10)
```

---

simHDSg   *Simulate data under HDS protocol with groups*

Description

Simulates hierarchical distance sampling (HDS) data for groups under either a line or a point transect protocol and using a half-normal detection function (Buckland et al. 2001).

Usage

```
simHDSg(type = c("line", "point"), nsites = 100, lambda.group = 0.75,
  alpha0 = 0, alpha1 = 0.5,
  beta0 = 1, beta1 = 0.5, B = 4, discard0 = TRUE, show.plot = TRUE)
```

Arguments

type	The type of distance transect, either "line" or "point".
nsites	Number of sites (spatial replication)
lambda.group	Poisson mean of group size
alpha0	intercept of log-linear model relating sigma of the half-normal detection function to group size
alpha1	slope of log-linear model relating sigma of the half-normal detection function to group size
beta0	intercept of log-linear model relating the Poisson mean of the number of groups per unit area to habitat
beta1	slope of log-linear model relating the Poisson mean of the number of groups per unit area to habitat
B	strip half width or the radius of the circle
discard0	whether to discard or keep the data from sites with nobody detected
show.plot	choose whether to show plots or not. Set to FALSE when using function in simulations.

Value

A list with the values of the arguments entered and the following additional elements:

data	simulated distance sampling data: a matrix with a row for each group detected and 6 columns: site ID, status (1 if captured), x and y coordinates (NA for line transects), distance from the line or point, group size; if discard0 = FALSE, sites with no detections will appear in the matrix with NAs in columns 2 to 6.
habitat	simulated habitat covariate

N	simulated number of groups at each site
N.true	for point counts, the simulated number of groups within the circle sampled
groupsize	group size for each of the groups observed

Author(s)

Marc Kéry & Andy Royle

References

Buckland, S.T., et al (2001) *Introduction to distance sampling: estimating abundance of biological populations*. Oxford University Press, Oxford, UK.

Kéry, M. & Royle, J.A. (2016) *Applied Hierarchical Modeling in Ecology* AHM1 - 9.2.1.

Examples

```
# Run with the default arguments and look at the structure of the output:
set.seed(123)
tmp <- simHDSg()
str(tmp)
head(tmp$data)

str(simHDSg(type = "line"))     # Defaults for line transect data
str(simHDSg(type = "point"))    # Defaults for point transect data
str(simHDSg(lambda.group = 5))  # Much larger groups
str(simHDSg(lambda.group = 5, alpha1 = 0)) # No effect of groups size on p
```

---

simHDSopen	*Simulate open hierarchical distance sampling data*

Description

Simulates distance sampling data from a multi-season (or multi-year) model, incorporating habitat and detection covariates, temporary emigration, and a trend in abundance or density.

Usage

```
simHDSopen(type=c("line", "point"), nsites = 100,
  mean.lam = 2, beta.lam = 0, mean.sig = 1, beta.sig = 0,
  B = 3, discard0 = TRUE, nreps = 2, phi = 0.7, nyears = 5, beta.trend = 0)
```

Arguments

type	the transect protocol, either "line" or "point" .
nsites	Number of sites (spatial replication)
mean.lam	intercept of log-linear regression of expected lambda on a habitat covariate
beta.lam	slope of log-linear regression of expected lambda on a habitat covariate

mean.sig	intercept of log-linear regression of scale parameter of half-normal detection function on wind speed
beta.sig	slope of log-linear regression of scale parameter of half-normal detection function on wind speed
B	strip half width, or maximum distance from the observer for point counts
discard0	Discard sites at which no individuals were captured. You may or may not want to do this depending on how the model is formulated so be careful.
nreps	the number of distance sampling surveys within a period of closure in a season (or year)
phi	the availability parameter
nyears	the number of seasons (typically years)
beta.trend	loglinear trend of annual population size or density

Value

A list with the values of the arguments entered and the following additional elements:

data	simulated distance sampling data: a list with a component for each year, each itself a list with a component for each replicate; this is a matrix with a row for each individual detected and 5 columns: site ID, status (1 if captured), x and y coordinates (NA for line transects), distance from the line or point; if discard0 = FALSE, sites with no detections will appear in the matrix with NAs in columns 2 to 5.
habitat	simulated habitat covariate, a vector of length nsites
wind	simulated detection covariate, a nsites x nreps x nyears array
M.true	simulated number of individuals, a nsites x nyears matrix
K	= nreps
Na	the number of individuals available for detection, a nsites x nreps x nyears array
Na.real	for point counts, the number of individuals available for detection within the circle sampled, a nsites x nreps x nyears array

Note

For "point" the realized density is [(area of circle) /(area of square)]*lambda

Author(s)

Marc Kéry & Andy Royle

References

Kéry, M. & Royle, J.A. (2016) *Applied Hierarchical Modeling in Ecology* AHM1 - 9.5.4.1.

Examples

```
set.seed(123)
tmp <- simHDSopen() # Generate data with default parameters
str(tmp)
head(tmp$data[[1]][[1]])

tmp <- simHDSopen("point")
str(tmp)
head(tmp$data[[1]][[1]])

tmp <- simHDSopen(discard0=FALSE)
str(tmp)
head(tmp$data[[1]][[1]])
```

---

simHDStr	*Simulate HDS time-removal or double-observer data*

Description

A general function for simulating hierarchical distance sampling (HDS) data combined with a time-removal (with 3 removal periods) or double-observer protocol, either for a line or a point transect protocol and with method = "removal" or method = "double". Also produces plots of the output.

Usage

```
simHDStr(type = c("line", "point"), method=c("removal", "double"), nsites = 200,
  lambda.group = 1, alpha0 = 0, alpha1 = 0, beta0 = 1, beta1 = 0.5,
  p.avail = 0.75, K = 3, p.double = c(0.4, 0.6),
  B = 3, discard0 = FALSE, show.plot = TRUE)
```

Arguments

type	The type of distance transect, either "line" or "point".
method	Is the method time-removal ("removal") or double-observer ("double")
nsites	Number of sites (spatial replication)
lambda.group	Poisson mean of group size
alpha0	intercept of log-linear model relating sigma of half-normal detection function to group size
alpha1	slope of log-linear model relating sigma of half-normal detection function to group size
beta0	intercept of log-linear model relating the Poisson mean of the number of groups per unit area to habitat
beta1	slope of log-linear model relating the Poisson mean of the number of groups per unit area to habitat
p.avail	overall availability probability (phi in text)

K	number of removal periods (of equal length)
p.double	detection probability for first and second observer
B	strip half width or radius of the circle
discard0	whether to discard or keep the data from sites with nobody detected
show.plot	choose whether to show plots or not. Set to FALSE when running many simulations.

Value

A list with the values of the arguments and the following additional elements:

data	simulated distance sampling data: a matrix with a row for each group detected and 7 columns: site ID, status (1 if captured), x and y coordinates (NA for line transects), distance from the line or point, group size, the time interval of removal or capture history; if discard0 = FALSE, sites with no detections will appear in the matrix with NAs in columns 2 to 7.
habitat	simulated habitat covariate
M	simulated number of groups at each site
M.true	for point counts, the simulated number of groups within the circle sampled
params	a vector with the input arguments

Author(s)

Marc Kéry & Andy Royle

References

Kéry, M. & Royle, J.A. (2016) *Applied Hierarchical Modeling in Ecology* AHM1 - 9.3.2 (time removal) and 9.4.1 (double observer).

Examples

```
# Run with the default arguments and look at the structure of the output:
set.seed(123)
tmp <- simHDStr() # default: line, removal
str(tmp)
head(tmp$data)
tmp <- simHDStr("point", method="double")
str(tmp)
head(tmp$data)
```

---

simMultMix   *Simulate the multinomial-mixture model*

Description

Simulation of "removal" data from a multinomial-mixture model.

Usage

```
simMultMix(nsites = 100, nsurveys = 3, nyears = 4,
  lambda = 3, theta = 0.5, p = 0.3)
```

Arguments

nsites	number of sites.
nsurveys	number of replicate (secondary) samples
nyears	number of primary samples: years, seasons etc.
lambda	expected local population size.
theta	availability, the proportion of the population available for detection.
p	detection probability.

Value

A list with the values of the input arguments and the following additional elements:

M	true local population size, vector length nsites
N	true number available for detection, vector length nsites
y	number detected, nsites x nyears x nsurveys
y2d	number detected as a matrix, nsites x (nyears*nsurveys)

Author(s)

Marc Kéry & Andy Royle, adapting code in documentation for gmultmix by Richard Chandler.

References

Chandler, R.B., Royle, J.A. & King. D.I. (2011) Inference about density and temporary emigration in unmarked populations. *Ecology* 92:1429-1435.

Kéry, M. & Royle, J.A. (2020) *Applied Hierarchical Modeling in Ecology* AHM2 - 2.7.1.

Examples

```
# Simulate a data set with the default arguments and look at the structure of the output:
tmp <- simMultMix()
str(tmp)
```

---

simNmix   *Simulate data for binomial and multinomial mixture models*

Description

This very general function generates single-season count data under variants of the binomial N-mixture model of Royle (2004) and of the multinomial N-mixture model of Royle et al (2007).

Usage

```
simNmix(nsites = 267, nvisits = 3, mean.theta = 1, mean.lam = 2, mean.p = 0.6,
  area = FALSE, beta1.theta = 0, beta2.theta = 0, beta3.theta = 0,
  beta2.lam = 0, beta3.lam = 0, beta4.lam = 0, beta3.p = 0, beta5.p = 0,
  beta6.p = 0, beta.p.survey = 0, beta.p.N = 0, sigma.lam = 0, dispersion = 10,
  sigma.p.site = 0, sigma.p.visit = 0, sigma.p.survey = 0, sigma.p.ind = 0,
  Neg.Bin = FALSE, open.N = FALSE, show.plots = TRUE, verbose = TRUE)
```

Arguments

nsites	number of sites
nvisits	number of visits per site
mean.theta	proportion of sites that can have non-zero abundance in principle: suitability model for zero-inflation
mean.lam	Expected abundance at the average value of all abundance covariates (and ignoring random site effects): abundance model
mean.p	Expected detection at the average value of all detection covariates (and ignoring all random effects): detection model
area	determines area of sites (A), defaults to A=1 (i.e., all identical), but you can supply a vector of site areas of length nsites instead.
beta1.theta	coefficient of site covariate 1 in suitability model
beta2.theta	coefficient of site covariate 2 in suitability model
beta3.theta	coefficient of site covariate 3 in suitability model
beta2.lam	coefficient of site covariate 2 in abundance model
beta3.lam	coefficient of site covariate 3 in abundance model
beta4.lam	coefficient of site covariate 4 in abundance model
beta3.p	coefficient of site covariate 3 in detection model
beta5.p	coefficient of site covariate 5 in detection model
beta6.p	coefficient of site covariate 6 in detection model
beta.p.survey	coefficient of survey ('observational') covariate on p
beta.p.N	coefficient of centered local population size ($\log(N+1)$) in detection model (i.e., coef. for density-dependent detection prob.)
sigma.lam	"Overdispersion SD" in linear predictor of abundance

dispersion	'size' or extra-Poisson dispersion of Negative binomial
sigma.p.site	"Overdispersion SD" in linear predictor of detection coming from random site effects
sigma.p.visit	"Overdispersion SD" in linear predictor of detection coming from random visit effects
sigma.p.survey	"Overdispersion SD" in linear predictor of detection coming from random site-by-survey effects
sigma.p.ind	"Overdispersion SD" in linear predictor of detection coming from random site-by-individual effects
Neg.Bin	if FALSE, any overdispersion in abundance is modeled by a Poisson log-normal; if TRUE, abundance overdispersion is modeled by adoption of a Negative binomial distribution for latent N
open.N	if TRUE, data are simulated under one specific form of an open population, where N in the first occasion is drawn from the specified mixture distribution and for all further occasions j, we have N_ij ~ Poisson(N_i(j-1)). With open.N = TRUE, we must have sigma.p.ind = 0, show.plot = FALSE and nvisits >1.
show.plots	if TRUE, plots of the data will be displayed; set to FALSE if you are running many simulations.
verbose	if TRUE, output will be written to the Console.

Details

Data are simulated at the level of each individual and individual-specific detection heterogeneity can be included. As a side-effect, individual-specific detection histories are generated and hence, data are also be simulated under the corresponding multinomial N-mixture model.

Broadly, the function can generate data under this most general model:

'Suitability' (zero-inflation) ~ cov1 + cov2 + cov3

Abundance ~ offset + cov2 + cov3 + cov4 + overdispersion

Detection ~ cov3 + cov5 + cov6 + survey.covariate + $\log(N+1)$ + eps.site + eps.visit + eps.survey + eps.individual

Overdispersion in abundance is modeled either as a Poisson-log-normal with a normal random site effect in lambda or with a Negative binomial with mean lambda and a 'size', or dispersion, parameter. Variable site areas can be specified to affect abundance as in an offset.

Abundance can be zero-inflated (this is the 'suitability' model). Note that the zero-inflation parameter is called theta here (in unmarked it is called psi). mean.phi is the probability that a site is suitable (i.e., 1 minus the expected proportion of sites with structural zero abundance.

Site covariate 2 can affect both suitability and abundance, while covariate 3 may affect all three levels. Hence, the function permits to simulate the case where a single site covariate affects different levels in the process (e.g., abundance and detection) in opposing directions (as for instance in Kéry, Auk, 2008)

Density-dependent detection can be modeled as a logistic-linear effect of local abundance (centered and $\log(x+1)$ transformed). Overdispersion in detection is modeled via normal random effects (the eps terms above) specific to sites, visits, surveys or individuals.

Effects of covariates and random-effects factors are modeled as additive on the link scale (log for abundance and logit for suitability and detection).

Data may be generated under one specific open-population model when argument 'open.N' is set to TRUE.

Value

A list with the arguments input and the following additional elements:

nobs	The total number of visits
site.cov	An nsites x 6 matrix of values for 6 site covariates
survey.cov	An nsites x nvisits matrix of values for a survey covariate
log.lam	Linear predictor of PLN abundance model including random effects, a vector of length nsites
s	Site suitability indicator, a vector of length nsites
N	Number of individuals at each site, a vector of length nsites
p	Probability of detection, an array with dimensions sites occupied x visits x max(N)
DH	Detection history (1/0), an array with dimensions sites occupied x visits x max(N)
N.open	Number of individuals at each site for open model, a sites x visits matrix
C	Summary of DH: number of individuals detected for each site and visit
eta.lam	Random site effects in lambda, a vector of length nsites, zero if Neg.Bin == TRUE
eta.p.site	Random site effects in p, a vector of length nsites
eta.p.visit	Random visit effects in p, a vector of length nvisits
eta.p.survey	Random survey (= site-by-survey) effects in p, a nsites x nvisits matrix
eta.p.ind	Random individual (= site-by-ind) effects in p (NOT site-ind-visit !), a nsites x max(N) matrix
odcN	Naive overdispersion measure (var/mean) for true abundance (N)
odcC	Naive overdispersion measure (var/mean) for observed counts (C)
Ntotal	Total abundance summed over all sites
summax	The sum of maximum counts over all sites

Author(s)

Marc Kéry & Andy Royle

References

Royle, J.A. (2004) N-mixture models for estimating population size from spatially replicated counts. *Biometrics* 60, 108-115.

Royle, J.A., et al (2007) Hierarchical spatial models of abundance and occurrence from imperfect survey data. *Ecological Monographs*, 77, 465-481.

Kéry, M. (2008) Estimating abundance from bird counts: binomial mixture models uncover complex covariate relationships, *Auk* 125, 336-345

Kéry, M. & Royle, J.A. (2016) *Applied Hierarchical Modeling in Ecology* AHM1 - 6.5.

Examples

```
# Generate data with the default arguments and look at the structure:
tmp <- simNmix()
str(tmp)


str(data <- simNmix())                    # Null data-generating model
str(data <- simNmix(mean.theta = 0.60)) # ZIP with 40% structural zeroes
str(data <- simNmix(sigma.lam = 1))     # Poisson-lognormal (PLN) mixture
str(data <- simNmix(Neg.Bin = TRUE))    # Negative-binomial mixture
str(data <- simNmix(mean.theta = 0.6, sigma.lam = 1))  # Zero-inflated PLN
str(data <- simNmix(mean.theta = 0.6, Neg.Bin = TRUE)) # Zero-infl. NegBin
str(data <- simNmix(mean.p = 1))        # Perfect detection (p = 1)
str(data <- simNmix(mean.theta = 0.6, mean.p = 1))     # ZIP with p = 1
str(data <- simNmix(sigma.lam = 1, mean.p = 1))        # PLN with p = 1
```

---

simNmixSpatial	*Simulate replicated counts under a spatial, static binomial N-mixture model*

Description

Simulates replicated counts under a spatial, static binomial N-mixture model for a semi-realistic landscape in a square of 50x50 km in the Bernese Oberland around Interlaken, Switzerland. Unit of the data simulation is a 1km2 quadrat, hence, there are 2500 units. It uses the data set BerneseOberland, which has covariates for elevation and forest cover.

For abundance, the function allows you to specify a quadratic effect of elevation. Then, a Gaussian spatial random field (s) with negative exponential correlation function is simulated using simExpCorrRF; you can set the variance and the range (scale) parameter theta. Basically, the larger the value of theta.RF, the bigger are the 'islands' simulated in the random field. The abundance in each quadrat (i) is built up via the following linear predictor:

```
lambda[i] <-exp(beta0 + beta[1] * elev[i] + beta[2] * elev[i]^2 + s[i])
N[i] ~ Poisson(lambda[i])
```

Replicated counts are simulated as usual under a binomial observation model, and detection probability is allowed to vary by one site and one observational covariate: respectively quadrat forest cover (in the BerneseOberland data set), and wind-speed, which is invented data. Counts at each site (i) and for each occasion (j) are then produced according to the following model:

```
 p[i,j] <-plogis(alpha0 + alpha[1] * forest[i] + alpha[2] * wind[i,j])
C[i,j] ~ Binomial(N[i],p[i,j])
```

Finally, we assume that only a subset of the 2500 quadrats is surveyed. Hence, we allow you to choose the number of quadrats that are surveyed and these will then be randomly placed into the landscape. We then assume that counts will only be available for these surveyed quadrats, i.e., counts from all non-surveyed quadrats will be NA.

Usage

```
simNmixSpatial(nsurveys = 3, mean.lambda = exp(2), beta = c(2, -2),
  mean.p = 0.5, alpha = c(-1, -1), sample.size = 500, variance.RF = 1, theta.RF = 10,
    seeds = c(10, 100), truncN = 6, show.plots=TRUE, verbose = TRUE)
```

Arguments

nsurveys	number of surveys per quadrat.
mean.lambda	expected number of individuals at a quadrat with mean values of all covariates; beta0 is log(mean.lambda).
beta	vector of length 2, the linear and quadratic coefficients for the effect of elevation on abundance.
mean.p	Expected detection at the average value of all detection covariates (and ignoring all random effects): detection model
alpha	vector of length 2, the coefficients for the effect of forest and wind on detection.
sample.size	the number of quadrats surveyed.
variance.RF	variance of the random field.
theta.RF	parameter governing spatial correlation (=1/phi); the larger the value of theta.RF, the bigger are the 'islands' simulated in the random field.
seeds	vector of length 2; random seeds used for the random field and the selection of quadrats surveyed respectively.
truncN	used to limit the range of values of N displayed in some plots.
show.plots	if TRUE, plots of the data will be displayed; set to FALSE if you are running many simulations.
verbose	if TRUE, output will be written to the Console.

Value

A list with the arguments input and the following additional elements:

xcoord, ycoord	The x and y coordinates from the BerneseOberland data set.
elevation	The elevation covariate from the BerneseOberland data set.
forest	The forest cover covariate from the BerneseOberland data set.
elevationS	The elevation covariate standardized to mean 0, SD 1.
forestS	The forest cover covariate standardized to mean 0, SD 1.
wind	The wind covariate, generated internally.
field	The spatially-autocorrelated covariate, generated internally.
beta0	Intercept for the abundance model, equal to log(mean.lambda).
lam	Expected abundance in each quadrat.
N	Number of individuals at each site.
Ntotal	Total number of individuals in the study area.
p	probability of detection for each survey.

y	number of individuals detected in each quadrat.
surveyed.sites	
	the IDs of the quadrats surveyed.
yobs	number of individuals detected in each quadrat surveyed, NA for quadrats not surveyed.

Author(s)

Marc Kéry & Andy Royle

References

Kéry, M. & Royle, J.A. (2020) *Applied Hierarchical Modeling in Ecology* AHM2 - 11.

Examples

```
# Generate data with the default arguments and look at the structure:
str(dat <- simNmixSpatial())
str(dat <- simNmixSpatial(show.plots=FALSE))

# More surveys
str(dat<- simNmixSpatial(nsurveys = 10))

# Minimal number of surveys is 1
str(dat<- simNmixSpatial(nsurveys = 1))

# Much more common species
str(dat<- simNmixSpatial(mean.lambda = exp(4)))

# Only negative linear effect of elevation
str(dat<- simNmixSpatial(beta = c(-2, 0)))

# No effect of elevation at all
str(dat<- simNmixSpatial(beta = c(0, 0)))

# Perfect detection (p = 1)
str(dat<- simNmixSpatial(mean.p = 1))

# No effect of forest cover on detection
str(dat<- simNmixSpatial(alpha = c(0, -1)))

# No effect of wind speed on detection
str(dat<- simNmixSpatial(alpha = c(-1, 0)))

# Sample only 100 quadrats
str(dat<- simNmixSpatial(sample.size = 100))

# Sample all 2500 quadrats
str(dat<- simNmixSpatial(sample.size = 2500))

# Larger variance of the multivariate Gaussian Random variable in the random field
#  (this will increase the effect of the field on abundance and counts)
```

```
str(dat<- simNmixSpatial(variance.RF = 10))

# No spatial autocorrelation (Variant 1: set variance to 0)
str(dat<- simNmixSpatial(variance.RF = 0))

# No spatial autocorrelation (Variant 2: set theta very close to 0,
#  but not quite 0, otherwise function breaks)
str(dat<- simNmixSpatial(theta.RF = 0.0001))

# Larger value of theta.RF gives larger 'islands'
str(dat<- simNmixSpatial(theta.RF = 100))

# Even larger value of theta.RF gives even larger 'islands'
str(dat<- simNmixSpatial(theta.RF = 1000))

# Truncate abundance in final plots to presence/absence
str(dat<- simNmixSpatial(truncN = 0.5))

# Essentially do not truncate abundance in final plots
str(dat<- simNmixSpatial(truncN = 70))
```

---

simNpC	*Generate counts from a single population observed over T years under a binomial observation process*

Description

Generates counts from a single population observed over T years and which can be observed with or without imperfect detection. The goal is to focus on what happens with relative-abundance inference when temporal patterns in abundance are confounded with temporal patterns in detection probability. Hence, we can simulate a stable population or one with linear increase or decrease with specified start and end points, and around which there is Poisson noise. The observed counts are Binomial outcomes with a detection probability which can similarly be chosen to be constant or change linearly over time.

Usage

```
simNpC(T = 20, expN = c(100, 75), dp = c(0.5, 0.5), show.plot = TRUE)
```

Arguments

T	The length of the time series.
expN	The expected abundance at start and end of period, linear trend.
dp	The detection probability at start and end of period, linear trend.
show.plot	Choose whether to show plots or not. Set to FALSE when using function in simulations.

Value

A list with the values of the arguments entered and the following additional elements:

lambda   T vector, expected abundance for each year.

p   T vector, detection probability (dp) for each year.

N   T vector, realized abundance.

C   T vector, observed counts.

Author(s)

Marc Kéry & Andy Royle

References

Kéry, M. & Royle, J.A. (2020) *Applied Hierarchical Modeling in Ecology* AHM2 - 1.2.

Examples

```
# Run with the default arguments and look at the structure of the output:
set.seed(123)
tmp <- simNpC()
str(tmp)
tmp$C
```

---

simOcc   *Simulate data for static occupancy models under wide range of conditions*

Description

Function to simulate static occupancy measurements replicated at M sites during J occasions. Population closure is assumed for each site. Expected occurrence may be affected by elevation (elev), forest cover (forest) and their interaction. Expected detection probability may be affected by elevation, wind speed (wind) and their interaction.

Usage

```
simOcc(M = 267, J = 3, mean.occupancy = 0.6, beta1 = -2, beta2 = 2, beta3 = 1,
  mean.detection = 0.3, time.effects = c(-1, 1),
  alpha1 = -1, alpha2 = -3, alpha3 = 0, sd.lp = 0.5,
  b = 2, show.plot = TRUE)
```

Arguments

M	Number of spatial replicates (sites)
J	Number of temporal replicates (occasions)
mean.occupancy	Mean occurrence at value 0 of occurrence covariates
beta1	Main effect of elevation on occurrence
beta2	Main effect of forest cover on occurrence
beta3	Interaction effect on occurrence of elevation and forest cover
mean.detection	Mean detection prob. at value 0 of detection covariates
time.effects	bounds (on logit scale) for uniform distribution from which time effects gamma will be drawn
alpha1	Main effect of elevation on detection probability
alpha2	Main effect of wind speed on detection probability
alpha3	Interaction effect on detection of elevation and wind speed
sd.lp	standard deviation of random site effects (on logit scale)
b	constant value of 'behavioral response' leading to 'trap-happiness' (if b > 0) or 'trap shyness' (if b < 0)
show.plot	if TRUE, plots of the data will be displayed; set to FALSE if you are running many simulations

Value

A list with the values of the input arguments and the following additional elements:

gamma	The time effects, a vector of length J
eps	Individual random effects, a vector of length M
elev	Elevation, a vector of length M
forest	Forest cover, a vector of length M
wind	wind speed, a M x J matrix
psi	Probability of occurrence, a vector of length M
z	Realized occurrence (0/1), a vector of length M
p	probability of capture, possibly with a behavioral effect, a M x J matrix
p0	probability of capture when not captured on previous occasion, a M x J matrix
p1	probability of capture when captured on previous occasion, a M x J matrix
y	simulated capture history, a M x J matrix
sumZ	True number of occupied sites
sumZ.obs	Number of sites observed to be occupied
psi.fs.true	True proportion of occupied sites in sample (sumZ/N)
psi.fs.obs	Proportion of sites observed to be occupied (sumZ.obs/N)

Author(s)

Marc Kéry & Andy Royle

References

Kéry, M. & Royle, J.A. (2016) *Applied Hierarchical Modeling in Ecology* AHM1 - 10.8.

Examples

```
# Generate data with the default arguments and look at the structure:
tmp <- simOcc()
str(tmp)

# Simplest possible occupancy model, with constant occupancy and detection
str(simOcc(mean.occ=0.6, beta1=0, beta2=0, beta3=0, mean.det=0.3, time.effects=c(0, 0),
  alpha1=0, alpha2=0, alpha3=0, sd.lp=0, b=0))
# psi = 1 (i.e., species occurs at every site)
str(simOcc(mean.occ=1))

# p = 1 (i.e., species is always detected when it occurs)
str(simOcc(mean.det=1))

# Other potentially interesting settings include these:

str(simOcc(J = 2))                  # Only 2 surveys
str(simOcc(M = 1, J = 100))         # No spatial replicates, but 100 measurements
str(simOcc(beta3 = 1))              # Including interaction elev-wind on p
str(simOcc(mean.occ = 0.96))        # A really common species
str(simOcc(mean.occ = 0.05))        # A really rare species
str(simOcc(mean.det = 0.96))        # A really easy species
str(simOcc(mean.det = 0.05))        # A really hard species
str(simOcc(mean.det = 0))           # The dreaded invisible species
str(simOcc(alpha1=-2, beta1=2))     # Opposing effects of elev on psi and p
str(simOcc(J = 10, time.effects = c(-5, 5))) # Huge time effects on p
str(simOcc(sd.lp = 10))             # Huge (random) site effects on p
str(simOcc(J = 10, b = 0))          # No behavioural response in p
str(simOcc(J = 10, b = 2))          # Trap happiness
str(simOcc(J = 10, b = -2))         # Trap shyness
```

---

simOccSpatial	*Simulate replicated detection/nondetection data under a spatial, static occupancy model*

Description

Simulates replicated detection/nondetection data under a spatial, static occupancy model for a semi-realistic landscape in a square of 50x50 km in the Bernese Oberland around Interlaken, Switzerland.

Unit of the data simulation is a 1km2 quadrat, hence, there are 2500 units. It uses the data set [BerneseOberland](), which has covariates for elevation and forest cover.

For occupancy, the function allows you to specify a quadratic effect of elevation. Then, a Gaussian spatial random field (s) with negative exponential correlation function is simulated using [simExpCorrRF](); you can set the variance and the range (scale) parameter theta. Basically, the larger the value of theta.RF, the bigger are the 'islands' simulated in the random field. The abundance in each quadrat (i) is built up via the following linear predictor:

```
psi[i] <-qlogis(beta0 + beta[1] * elev[i] + beta[2] * elev[i]^2 + s[i])
z[i] ~ Bernoulli(psi[i])
```

Replicated detection/nondetection data are simulated as usual under a Bernoulli observation model, and detection probability is allowed to vary by one site and one observational covariate: respectively quadrat forest cover (in the BerneseOberland data set), and wind-speed, which is invented data. Counts at each site (i) and for each occasion (j) are then produced according to the following model:

```
p[i,j] <-plogis(alpha0 + alpha[1] * forest[i] + alpha[2] * wind[i,j])
y[i,j] ~ Bernoulli(z[i] * p[i,j])
```

Finally, we assume that only a subset of the 2500 quadrats is surveyed. Hence, we allow you to choose the number of quadrats that are surveyed and these will then be randomly placed into the landscape. We then assume that counts will only be available for these surveyed quadrats, i.e., detection/non-detection data from all non-surveyed quadrats will be replaced with NA.

Usage

```
simOccSpatial(nsurveys = 3, mean.psi = 0.6, beta = c(2, -2),
 mean.p = 0.4, alpha = c(-1, -1), sample.size = 500, variance.RF = 1, theta.RF = 10,
  seeds = c(10, 100), show.plots = TRUE, verbose = TRUE)
```

Arguments

nsurveys	number of surveys per quadrat.
mean.psi	probability of occupancy at a quadrat with mean values of all covariates; beta0 is qlogis(mean.psi).
beta	vector of length 2, the linear and quadratic coefficients for the effect of elevation on occupancy.
mean.p	Expected detection at the average value of all detection covariates (and ignoring all random effects): detection model
alpha	vector of length 2, the coefficients for the effect of forest and wind on detection.
sample.size	the number of quadrats surveyed.
variance.RF	variance of the random field.
theta.RF	parameter governing spatial correlation (=1/phi); the larger the value of theta.RF, the bigger are the 'islands' simulated in the random field.
seeds	vector of length 2; random seeds used for the random field and the selection of quadrats surveyed respectively.
show.plots	if TRUE, plots of the data will be displayed; set to FALSE if you are running many simulations.
verbose	if TRUE, output will be written to the Console.

Value

A list with the arguments input and the following additional elements:

xcoord, ycoord	The x and y coordinates from the `BerneseOberland` data set.
elevation	The elevation covariate from the `BerneseOberland` data set.
forest	The forest cover covariate from the `BerneseOberland` data set.
elevationS	The elevation covariate standardized to mean 0, SD 1.
forestS	The forest cover covariate standardized to mean 0, SD 1.
wind	The wind covariate, generated internally.
field	The spatially-autocorrelated covariate, generated internally.
alpha0	Intercept for the detection model, equal to `qlogis(mean.p)`.
beta0	Intercept for the abundance model, equal to `qlogis(mean.psi)`.
z	True occupancy in each quadrat.
trueNocc	True number of occupied sites.
obsNocc	Number of sites where the species was detected.
true_psi_fs	
obs_psi_fs	
p	probability of detection for each survey.
y	detected/non-detected data for each quadrat.
surveyed.sites	
	the IDs of the quadrats surveyed.
yobs	detected/non-detected data for each quadrat surveyed, NA for quadrats not surveyed.

Author(s)

Marc Kéry & Andy Royle

References

Kéry, M. & Royle, J.A. (2020) *Applied Hierarchical Modeling in Ecology* AHM2 - 9.2.

Examples

```
# Generate data with the default arguments and look at the structure:
str(dat <- simOccSpatial())
str(dat <- simOccSpatial(show.plots=FALSE))

# More surveys
str(dat<- simOccSpatial(nsurveys = 10))

# Minimal number of surveys is 1
str(dat<- simOccSpatial(nsurveys = 1))

# A truly ubiquitous species
```

```
str(dat <- simOccSpatial(mean.psi = 1))

# Only negative linear effect of elevation
str(dat <- simOccSpatial(beta = c(2, 0)))

# No effect of elevation at all (see effects of spatial field now clearly)
str(dat <- simOccSpatial(beta = c(0, 0)))

# Perfect detection (p = 1)
str(dat <- simOccSpatial(mean.p = 1))

# No effect in detection of forest cover
str(dat <- simOccSpatial(alpha = c(0, -1)))

# No effect in detection of wind speed (see neatly forest effect now)
str(dat <- simOccSpatial(alpha = c(-1, 0)))

# Sample only 100 quadrats
str(dat <- simOccSpatial(sample.size = 100))

# Sample all 2500 quadrats
str(dat <- simOccSpatial(sample.size = 2500))

# Larger variance of the multivariate Gaussian Random variable in the random field
#  (this will increase the effect of the field on occupancy and detection)
str(dat <- simOccSpatial(variance.RF = 10))

# No spatial autocorrelation (Variant 1: set variance to 0)
str(dat <- simOccSpatial(variance.RF = 0))

# No spatial autocorrelation (Variant 2: set theta very close to 0,
#  but not quite 0, otherwise function breaks)
str(dat <- simOccSpatial(theta.RF = 0.0001))

# Larger value of theta.RF gives larger 'islands'
str(dat <- simOccSpatial(theta.RF = 100))

# Even larger value of theta.RF gives even larger 'islands'
str(dat <- simOccSpatial(theta.RF = 10000))
```

simOccttd	*Simulate time-to-detection occupancy data (single visits)*

Description

Function simulates time-to-detection occupancy design data under model of Garrard et al. (2008), also see Bornand et al. (2014)

Usage

```
simOccttd(M = 250, mean.psi = 0.4, mean.lambda = 0.3,
   beta1 = 1, alpha1 = -1, Tmax = 10, show.plot = TRUE, verbose = TRUE)
```

Arguments

M	Number of sites
mean.psi	intercept of occupancy probability
mean.lambda	intercept of Poisson rate parameter
beta1	slope of continuous covariate B on logit(psi)
alpha1	slope of continuous covariate A on log(lambda)
Tmax	maximum search time (in arbitrary units, which are same as response), response will be censored at Tmax
show.plot	choose whether to show plots or not. Set to FALSE when running many simulations.
verbose	if TRUE, output will be written to the Console.

Value

A list with the values of the arguments input and the following additional elements:

covA	Simulated values of covariate A, a vector of length M
covB	Simulated values of covariate B, a vector of length M
psi	Probability of occurrence at each site, a vector of length M
z	Realized occurrence at each site, a 0/1 vector of length M
ttd.temp	Uncensored simulated time-to-detection at each site, a vector of length M
ttd	Censored simulated time-to-detection at each site, a vector of length M
d	Censoring indicator, a 0/1 vector of length M
sum.z	Total number of sites occupied
n.obs	Total number of sites where the species was observed

Author(s)

Marc Kéry & Andy Royle

References

Garrard, G.E., Bekessy, S.A., McCarthy, M.A., & Wintle, B.A. (2008) When have we looked hard enough? A novel method for setting minimum survey effort protocols for flora surveys. *Austral Ecology*, 33, 986-998.

Bornand, C.N., Kéry, M., Bueche, L., & Fischer, M. (2014) Hide and seek in vegetation: time-to-detection is an efficient design for estimating detectability and occurrence. *Methods in Ecology and Evolution*, 5, 433-442.

Kéry, M. & Royle, J.A. (2016) *Applied Hierarchical Modeling in Ecology* AHM1 - 10.12.1.

Examples

```
# Generate data with the default arguments and look at the structure:
tmp <- simOccttd()
str(tmp)
```

---

simPH   *Generate counts under a variant of a 'phenomenological model'*

Description

Function generates (insect) counts under a variant of a 'phenomenological model' of Dennis et al. (2016). Interannual population model is exponential population growth, with Poisson initial abundance governed by initial.lambda and annually varying growth rate (or productivity parameter) gamma. Within-year dynamics is described by a Gaussian curve with date of mean flight period mu (site- and year-specific) and length of flight period sigma (only year-specific). Counts are made subject to a detection probability (p), which varies randomly according to a uniform distribution for every single count. Counts are plotted for up to 16 populations only (but can be simulated for any number).

Usage

```
simPH(npop = 18, nyears = 17, nreps = 10, date.range = 1:150, initial.lambda = 300,
  gamma.parms = c(0, 0.3), mu.range = c(50, 80),  sigma.range = c(10, 20),
  p.range = c(0.4, 0.6), show.plot = TRUE)
```

Arguments

npop	The number of populations.
nyears	The number of years (seasons).
nreps	The number of surveys per year.
date.range	The dates over which surveys may be conducted.
initial.lambda	The Poisson mean of initial relative population size.
gamma.parms	The mean and SD of lognormal interannual productivity.
mu.range	The range of date of peak flight period (varies by site and year).
sigma.range	The range of sigma of Gaussian phenology curve (varies by year only).
p.range	The range of detection probabilities (varies by site, year and visit).
show.plot	Choose whether to show plots or not. Set to FALSE when using function in simulations.

Value

A list with the values of the arguments entered and the following additional elements:

gamma	(nyears-1) vector, change in abundance for each interval between years.
n	site x year matrix, true relative abundance.
mu	site x year matrix, mean date of the flight period.
sigma	nyears vector, half-length of flight period.
date	site x year x reps array, dates of the surveys.
a	site x year x reps array, phenology term.
lambda	site x year x reps array, expected counts.
p	site x year x reps array, probability of detection.
C	site x year x reps array, simulated counts.

Author(s)

Marc Kéry & Andy Royle

References

Dennis, E.B., et al (2016) Dynamic models for longitudinal butterfly data, *Journal of Agricultural, Biological and Environmental Statistics* 21, 1-21.

Kéry & Royle (2020) *Applied Hierarchical Modeling in Ecology* AHM2 - 1.8.1.

Examples

```
# Run with the default arguments and look at the structure of the output:
set.seed(123)
tmp <- simPH()
str(tmp)
summary(tmp$C)
```

---

simpleNmix	*Simulate N-mixture data under a time-for-space substitution design*

Description

A simple function to simulate data under binomial N-mixture model where you have a single site that is surveyed over 'nyears' primary sampling periods ('seasons', 'years'), within each of which there are 'nreps' secondary samples.

Usage

```
simpleNmix(nyears = 12, nreps = 4, beta0 = 2, beta1 = 0.1,
  alpha0 = 0.5, alpha1 = -0.1, alpha2 = 1, show.plot = TRUE)
```

Arguments

nyears	Number of primary sampling periods.
nreps	Number of secondary samples within each primary period.
beta0	the intercept of a log-linear model of expected abundance (lambda).
beta1	the Time coefficient of a log-linear model for lambda.
alpha0	the intercept of a logit-linear model for detection (p).
alpha1	the Time coefficient of a logit-linear model for detection (p).
alpha2	the coefficient of a survey-specific covariate such as temperature (temp).
show.plot	choose whether to show plots or not. Set to FALSE when using function in simulations.

Value

A list with the values of the arguments input and the following additional elements:

N	The realized number of individuals at each primary season, a vector of length nyears
C	The number of individuals counted at each survey, a nyears x nreps matrix.
Time	The Time covariate, a vector of length nyears.
temp	The temperature covariate, a nyears x nreps matrix.
p	The probability of detection, a nyears x nreps matrix.

Author(s)

Marc Kéry & Andy Royle

References

Kéry, M. & Royle, J.A. (2016) *Applied Hierarchical Modeling in Ecology* AHM2 - 6.12.

Examples

```
# Simulate a data set with the default arguments and look at the structure of the output:
tmp <- simpleNmix()
str(tmp)
```

---

simPOP   *Simulate data for a demographic state-space model*

Description

Simulate multiple time-series of counts under a pure Markov model (with exponential population model) or under an extended Markov model with exponential-plus-random-immigration population model; see Sollmann et al.(2015). Default is Markov model, setting sd.rho to a value greater than 0 changes to extended Markov and sets the amount of random immigration.

Usage

```
simPOP(M = 100, T = 10, mean.lam = 3, beta.lam = 0, sd.log.lam = 0,
  mean.gamma = 1.0, beta.gamma = 0, sd.log.gamma.site = 0,
  sd.log.gamma.time = 0, sd.log.gamma.survey = 0, sd.rho = 0,
  mean.p = 0.6, beta.p = 0, sd.logit.p.site = 0, sd.logit.p.time = 0,
  sd.logit.p.survey = 0, show.plot = TRUE)
```

Arguments

M	The number of sites.
T	The number of years.
mean.lam	The mean abundance for year 1.
beta.lam	The covariate coefficient for lambda.
sd.log.lam	The over-dispersion in lambda.
mean.gamma	The mean population growth rate.
beta.gamma	The covariate coefficient for gamma.
sd.log.gamma.site	
	SD of random site effects for gamma.
sd.log.gamma.time	
	SD of random time effects for gamma.
sd.log.gamma.survey	
	SD of random survey (site+time) effects for gamma.
sd.rho	The random immigration term.
mean.p	The mean detection probability.
beta.p	The covariate coefficient for p.
sd.logit.p.site	
	SD of random site effects for p on the logit scale.
sd.logit.p.time	
	SD of random time effects for p on the logit scale.
sd.logit.p.survey	
	SD of random survey (site+time) effects for p on the logit scale.
show.plot	Choose whether to show plots or not. Set to FALSE when using function in simulations.

Value

A list with the values of the arguments entered and the following additional elements:

Xsite1	M vector, site covariate affecting initial abundance (lambda).
Xsiteyear1	M x T matrix, yearly site covariate affecting recruitment (gamma).
Xsiteyear2	M x T matrix, yearly site covariate affecting detection (p).
eps.N	M vector, site over-dispersion at t = 1.
lambda	M vector, abundance in year 1.
eps.gamma.site	
	M vector, random site effect for gamma.
eps.gamma.time	
	T vector, random time effect for gamma.
eps.gamma.survey	
	M x T matrix, random survey effect for gamma.
gamma	M x T matrix, population growth rate.
rho	(T-1) vector, immigration rate.
eps.p.site	M vector, random site effect for detection.
eps.p.time	T vector, random time effect for detection.
eps.p.survey	M x T matrix, random survey effect for detection.
p	M x T matrix, detection probability.
N	M x T matrix, true population.
C	M x T matrix, simulated counts.
zeroNyears	scalar, sum(N == 0).
Nextinct	scalar, number of sites where N == 0 at time T.
extrate	scalar, proportion of sites where N == 0 at time T.
sumN	T vector, total population in each year.
gammaX	(T-1) vector, realized population growth rate.

Author(s)

Marc Kéry & Andy Royle

References

Sollmann, R. et al. (2015) An open-population hierarchical distance sampling model *Ecology* 96, 325-331.

Kéry, M. & Royle, J.A. (2020) *Applied Hierarchical Modeling in Ecology* AHM2 - 1.7.1.

Examples

```
# Run with the default arguments and look at the structure of the output:
set.seed(123)
tmp <- simPOP()
str(tmp)
head(tmp$C)
```

---

simPPe   *Simulate a spatial point pattern in a heterogeneous landscape*

Description

The function simulates a spatial point pattern in a heterogeneous landscape simulated on a square landscape. The study area ('core') is simulated inside the larger landscape that includes a buffer. The size of the core is defined by the lscape.size minus twice the buffer.

There is one habitat covariate X that affects the intensity of the points. X is spatially structured with negative-exponential spatial autocorrelation; the parameters of the field can be chosen to create large 'islands' of similar values or no 'islands' at all, in which case the field is spatially unstructured.

The intensity of STATIC points (e.g. animal activity centers) may be inhomogeneous and affected by the coefficient beta, which is the log-linear effect of X.

To recreate the data sets used in the book with R 3.6.0 or later, include sample.kind="Rounding" in the call to set.seed. This should only be used for reproduction of old results.

Usage

```
simPPe(lscape.size = 150, buffer.width = 25, variance.X = 1, theta.X = 10,
  M = 250, beta = 1, quads.along.side = 6)
```

Arguments

lscape.size	size (width = height) of the square landscape, including core and buffer.
buffer.width	width of buffer around core study area.
variance.X	variance of Gaussian random field (covariate X).
theta.X	scale parameter of correlation in field (must be >0).
M	expected number of activity centers in core area.
beta	coefficient of the habitat covariate.
quads.along.side	
	number of quadrats along the side of the core area; the total number of quadrats will be quads.along.side^2, thus indirectly defining the quadrat area.

Value

A list with the values of the input arguments and the following additional elements:

core	range of x and y coordinates in the 'core'
M2	number of ACs in the total landscape, including the buffer
grid	coordinates of the center of each pixel
pixel.size	length of side of each pixel
size.core	the width=height of the core area
prop.core	the proportion of the landscape inside the core
X	matrix of covariate values for each pixel
probs	matrix of probabilities of an AC being inside each pixel (sums to 1)
pixel.id	the ID of the pixel for each AC
u	2-column matrix, coordinate of each AC
nsite	number of quadrats
quad.size	width = height of each quadrat
breaks	boundaries of the quadrats
mid.pt	mid-points of the quadrats
lambda_pp	intensity of point pattern (ACs per unit area)
Nac	site-specific abundance of ACs
zac	site-specific occurrence (0/1) of ACs
E_N	average realized abundance per quadrat
E_z	average realized occupancy per quadrat

Author(s)

Marc Kéry & Andy Royle.

References

Kéry, M. & Royle, J.A. (2020) *Applied Hierarchical Modeling in Ecology* AHM2 - 10.

Examples

```
# Nice plot (produces the really nice Fig. 10-2 in the book)
# RNGkind(sample.kind = "Rounding") # run this for R >= 3.6.0
set.seed(117, kind="Mersenne-Twister")
str(dat <- simPPe(lscape.size = 200, buffer.width = 25, variance.X = 1,
  theta.X = 70, M = 200, beta = 1, quads.along.side = 6))


# Smaller study area, fewer individuals (M)
str(dat <- simPPe(lscape.size = 24, buffer.width = 2, variance.X = 1,
  theta.X = 10, M = 50, beta = 1, quads.along.side = 6))
```

```
# Stronger habitat heterogeneity (variance.X): more aggregation
str(dat <- simPPe(lscape.size = 24, buffer.width = 2, variance.X = 10,
  theta.X = 10, M = 50, beta = 1, quads.along.side = 6))

# Longer habitat gradient (theta.X)
str(dat <- simPPe(lscape.size = 24, buffer.width = 2, variance.X = 1,
  theta.X = 100, M = 250, beta = 1, quads.along.side = 6))

# No habitat variability (variance.X): homogeneous point process
str(dat <- simPPe(lscape.size = 24, buffer.width = 2, variance.X = 0,
  theta.X = 10, M = 100, beta = 1, quads.along.side = 6))
# No habitat preference (beta): homogeneous point process
str(dat <- simPPe(lscape.size = 24, buffer.width = 2, variance.X = 1,
  theta.X = 10, M = 100, beta = 0, quads.along.side = 6))
# Habitat heterogeneity at very small scale (theta.X)
str(dat <- simPPe(lscape.size = 1000, buffer.width = 20, variance.X = 1,
  theta.X = 0.001, M = 250, beta = 1, quads.along.side = 6))

str(simPPe(M = 1))      # Often produces no point at all
str(simPPe(M = 10))
str(simPPe(M = 100))
str(simPPe(M = 1000))

str(simPPe(M = 20, quads.along.side = 100)) # Lots of small sites
str(simPPe(M = 20, quads.along.side = 10))
str(simPPe(M = 20, quads.along.side = 5))
str(simPPe(M = 20, quads.along.side = 1))   # whole study area is 1 site
```

---

simSpatialDSline   *Simulate data for a standardized line transect*

Description

This simulates line transect distance sampling data with a spatial distribution of objects in a heterogeneous landscape where the parameter beta controls the effect of habitat. Habitat is simulated according to a Gaussian random field model defined within the function. Uses a half normal detection model (if perp = TRUE) or a Gaussian hazard model (perp = FALSE).

To recreate the data sets used in the book with R 3.6.0 or later, include sample.kind="Rounding" in the call to set.seed. This should only be used for reproduction of old results.

Usage

```
simSpatialDSline(N=1000, beta = 1, sigma=0.25, alpha0 = -2, W=1/2, L = 4, perp=FALSE)
```

Arguments

N	total population size in the rectangle
beta	coefficient of spatial covariate x for the density model.

sigma	scale of half-normal detection function
alpha0	intercept of the hazard function.
W	half-width of the rectangle, which extends W each side of the transect line.
L	length of the transect.
perp	if TRUE, data are simulated for a traditional distance sampling model with perpendicular distances; if FALSE (the default) a model with 'forward distance' data, ie, the distance from the observer to the animal on first detection.

Value

A list with the values of the input arguments and the following additional elements:

delta	the distance between pixel centers (spatial resolution of the raster
grid	2-column matrix with x/y coordinates of all pixels
Habitat	value of habitat covariate for each pixel
Habraster	a Raster object with the habitat covariate
u1, u2	x and y coordinates for all the animals in the population
traps	2-column matrix of trap locations

If perp = TRUE we have

data	a 2-column matrix with x and y coordinates of each animal captured.
pixel	pixel ID for each animal captured.

and if perp = FALSE we have

data	a matrix with rows for each animal captured and columns for trap of first capture, distance from trap to animal, and x and y coordinates of the animal.
pbar	probability that each animal is the population is captured at least once
pixel	pixel ID for each animal captured.

Author(s)

Marc Kéry & Andy Royle

References

Kéry, M. & Royle, J.A. (2020) *Applied Hierarchical Modeling in Ecology* AHM2 - 11.

Examples

```
# Run the function with default values and look at the output
str(tmp <- simSpatialDSline(), 1)  # use str(., max.level=1) to limit the amount of output.
```

simSpatialDSte   *Simulate data for replicate line transect surveys with temporary emigration*

Description

This simulates line transect distance sampling data with a spatial distribution of objects in a heterogeneous landscape where the parameter beta controls the effect of habitat. Multiple sample occasions are simulated and temporary emigration is allowed (parameter phi). Habitat is simulated according to a Gaussian random field model defined within the function. Uses a half normal detection model (if perp = TRUE) or a Gaussian hazard model (perp = FALSE).

Usage

```
simSpatialDSte(nsites=28, dim=10, beta=1, lam0=2.5, nsurveys=4, sigma=3,
  phi=0.6, theta=2, show.plots=3)
```

Arguments

nsites	number of sites
dim	number of pixels along each side of the square site
beta	the effect of habitat on the number of individuals in a pixel.
lam0	expected population size at each site
nsurveys	the number of replicate surveys
sigma	scale of half-normal detection function
phi	probability an individual is available for detection, ie, not temporarily emigrated.
theta	exponential correlation in the spatial covariate.
show.plots	the number of sites for which plots should be displayed.

Value

A list with the values of the input arguments and the following additional elements:

npixels	the number of pixels in each site (= dim^2)
B	distance from line to edge of square (= dim/2)
M	true number of individuals at each site
d	perpendicular distance of each pixel from the line
Habitat	pixels x sites matrix, value of habitat covariate for each pixel
y	sites x pixels x surveys array, number of animals detected
Counts	sites x surveys matrix, number of animals detected (summed over pixels)

Author(s)

Mizel et al (2018) Appendix S1, based in turn on Kéry & Royle (2016).

References

Kéry, M. & Royle, J.A. (2016) *Applied Hierarchical Modeling in Ecology*.

Mizel, J.D., Schmidt, J.H., & Lindberg, M.S. (2018) Accommodating temporary emigration in spatial distance sampling models. *Journal of Applied Ecology*, 55, 1456-1464.

Kéry, M. & Royle, J.A. (2020) *Applied Hierarchical Modeling in Ecology* AHM2 - 11.

Examples

```
# Run the function with default values and look at the output
str(tmp <- simSpatialDSte())
```

---

spline.prep	*Prepare input for BUGS model when fitting a spline for a covariate*

Description

Function chooses knots and creates design matrices for fixed and random-effects parts of a spline model for a chosen covariate. Based on code by Crainiceanu et al. (2005) and Zuur et al. (2012). Allows you to choose number of knots or else uses it by the rule given in Crainiceanu et al. (2005). Prepares fixed part of covariate as a quadratic polynomial.

Usage

```
spline.prep(cov, nknot = NA)
```

Arguments

cov	the covariate, a numeric vector
nknot	optional, number of knots

Value

A list with the following elements:

cov	The input covariate
knots	The values of the knots
X	The fixed-effects design matrix
Z	The random-effects design matrix

Author(s)

Marc Kéry & Andy Royle

References

Crainiceanu, C.M., Ruppert, D., & Wand, M.P. (2005) Bayesian analysis for penalized spline regression using WinBUGS. *Journal of Statistical Software*, 14.

Zuur, A.F., Saveliev, A.A., Ieno, E.N. (2012) *Zero-inflated Models and Generalized Linear Mixed Models with R*. Highlands Statistics

Kéry, M. & Royle, J.A. (2016) *Applied Hierarchical Modeling in Ecology* AHM1 - 10.14.

---

spottedWoodpecker	*Data for observations of Middle Spotted Woodpecker in Switzerland*

Description

The data set is based on checklists which volunteers submit from their bird-watching trips and which were then summarized by day and site (1 km2 quadrat) such that we have the number of surveys (= checklists) per quadrat and day, and the number among these on which a Middle-Spotted Woodpecker was recorded, during 162 days (Julian days 51–212, corresponding to 20 February–31 July). There are data from a total of 144,517 recorded surveys on 116,204 day/quadrat combinations during 26 years from 1545 1 km2 quadrats in which the species was ever recorded in Switzerland since 1985.

Usage

```
data("spottedWoodpecker")
```

Format

spottedWoodpecker is a data frame with 116,204 rows and 7 columns:

site quadrat identifier.

coordx, coordy the x and y coordinates of the center of the quadrat; the coordinate reference system intentionally not specified.

year, jdate the year and Julian date of the observations.

y the number of surveys during which the species was detected.

nsurveys the number of recorded surveys in the quadrat on that day.

Source

Database of Swiss Ornithological Institute (courtesy of Nicolas Strebel).

References

Kéry, M. & Royle, J.A. (2020) *Applied Hierarchical Modeling in Ecology* AHM2 - 4.11.2.

Examples

```
data(spottedWoodpecker)
str(spottedWoodpecker)
```

---

standardize | *Scaling and centering of vectors, matrices and arrays*

Description

Maps a numeric variable to a new object with the same dimensions. standardize is typically used to standardize a covariate to mean 0 and SD 1. standardize2match is used to standardize one object using the mean and SD of another; it is a wrapper for standardize(x,center=mean(y),scale=sd(y)).

Usage

```
standardize(x, center = TRUE, scale = TRUE)
standardize2match(x, y)
```

Arguments

x, y	a numeric vector, matrix or multidimensional array; NA and NaN are allowed; Inf and -Inf will produce all-NaN output if either center or scale is TRUE.
center	either a logical or a numeric value of length 1.
scale	either a logical or a numeric value of length 1.

Details

standardize differs from [scale](#) by (1) accepting multidimensional arrays but not data frames; (2) *not* standardizing column-wise but using a single value to center or to scale; (3) if x is a vector, the output will be a vector (not a 1-column matrix). If each column in the matrix represents a different variable, use scale not standardize.

Centering is performed before scaling.

If center is numeric, that value will be subtracted from the whole object. If logical and TRUE, the mean of the object (after removing NAs) will be subtracted.

If scale is numeric, the whole object will be divided by that value. If logical and TRUE, the standard deviation of the object (after removing NAs) will be used; this may not make sense if center = FALSE.

Value

A numeric object of the same dimensions as x with the standardized values. NAs in the input will be preserved in the output.

For the default arguments, the object returned will have mean approximately zero and SD 1. (The mean is not exactly zero as scaling is performed after centering.)

Author(s)

Mike Meredith, after looking at the code of base::scale.

Examples

```
# Generate some fake elevation data:
elev <- runif(100, min=100, max=500)
mean(elev) ; sd(elev)
str( e <- standardize(elev) )
mean(e) ; sd(e)

# Standardize so that e=0 corresponds to exactly 300m and +/- 1 to
#   a change of 100m:
e <- standardize(elev, center=300, scale=100)
mean(e)
mean(elev) - 300
range(e)
range(elev) - 300

# Generate data matrix for survey duration for 3 surveys at 10 sites
dur <- matrix(round(runif(30, 20, 60)), nrow=10, ncol=3)
d <- standardize(dur)
mean(d) ; sd(d)

# Standardize new data to match the mean and SD of 'dur'
(new <- seq(20, 60, length.out=11))
standardize2match(new, dur)

# compare with base::scale
dx <- base::scale(dur)
colMeans(dx) ; apply(dx, 2, sd)
colMeans(d) ; apply(d, 2, sd)
# Don't use 'standardize' if the columns in the matrix are different variables!
```

SwissAtlasHa *Data for 1ha quadrats from the Swiss Breeding Bird Atlas*

Description

A 1ha-scale subset of the count data from the most recent Swiss Breeding Bird Atlas (Knaus et al. 2018). The data consists of replicate counts of birds from 2,318 1ha quadrats for 78 passerine species. Each quadrat is surveyed up to three times during the breeding season (only twice above the tree line).

Usage

```
data("SwissAtlasHa")
```

Format

SwissAtlasHa is a list with 3 elements:

counts   a sites x replicates x species array of counts

sites   a data frame with rows for 2318 1ha quadrats and the following columns:

- x : the x coordinate of the center of the quadrat; the coordinate reference system intentionally not specified.
- y : the y coordinate of the center of the quadrat.
- elevation : the mean elevation of the quadrat, m.
- slope : the slope.
- north : the aspect, with due south = -1, due north = 1.
- forest : proportion of forest cover.
- nsurvey : the number of replicate surveys carried out in the quadrat.

date   a sites x replicates matrix with Julian dates of the surveys, 1 January = 1

Source

Swiss Ornithological Institute

References

Knaus, P., S. Antoniazza, S. Wechsler, J. Guélat, M. Kéry, N. Strebel, & T. Sattler (2018) Brutvogelatlas 2013–2016. Bestandsentwicklung der Brutvögel der Schweiz und des Fürstentums Liechtensteins (Swiss Breeding Bird Atlas 2013–2016). Schweizerische Vogelwarte, Sempach.

Kéry, M. & Royle, J.A. (2020) *Applied Hierarchical Modeling in Ecology* AHM2 - 8.4.2.

Examples

```
data(SwissAtlasHa)
str(SwissAtlasHa)
# display the species names:
dimnames(SwissAtlasHa$counts)[[3]]
```

| SwissEagleOwls | *Territory-level, multi-state detection/nondetection data for Eagle Owls in Switzerland* |

Description

Multi-state detection/nondetection data stem from more or less opportunistic surveys of a total of 274 territories of the Eurasian Eagle Owl (*Bubo bubo*), the largest owl in the world, between 2007 and 2016 (10 years). Sites are all over Switzerland and represent the complete list of sites where a territorial Eagle Owl was ever detected during this period. For site/year combinations with more than 20 observations, 20 were randomly selected for inclusion.

The data set recognizes four observed states: 'nondetection', 'detection of single bird', 'detection of a pair without breeding evidence' and 'detection of a pair with confirmed breeding'; these being the classical criteria of bird atlases for possible, probable or confirmed breeding. For the analysis in the book, the last two are aggregated into a single state 'Pair'.

Usage

```
data("SwissEagleOwls")
```

Format

`SwissEagleOwls` is a list with 2 elements:

obs   a data frame with a row for each of the 5974 observations and the following columns:

- site_name : a numeric site ID.
- year : the year of the observation.
- jdate : the Julian date of the observation (1 = Jan 1st).
- y : the observed state: 0 = species not detected; 1 = detection of a single bird; 2 = detection of a pair without evidence of breeding; 3 = detection of a pair with evidence of breeding.

sites   a data frame with rows for 274 1x1km quadrats containing the sites, and the following columns:

- site_name : the numeric site ID, corresponding to site_name in the obs data frame.
- region : codes for six regions of Switzerland.
- elev : the elevation of the center of the quadrat, m.
- forest : proportion of the quadrat covered by forest.
- slope : the slope of the quadrat in degrees.
- roads : total road length in the quadrat, m.

Source

Swiss Ornithological Institute

References

Kéry, M. & Royle, J.A. (2020) *Applied Hierarchical Modeling in Ecology* AHM2 - 6.

Examples

```
data("SwissEagleOwls")
str(SwissEagleOwls)
```

SwissMarbledWhite   *Data for Swiss Marbled White butterflies*

Description

Detection/nondetection data for the Marbled White butterfly (*Melanargia galathea*) collected in the canton of Aargau, Switzerland, as part of the Aargau Biodiversity Monitoring Program (LANAG). 519 study plots were each sampled once every five years from 1998 to 2010 with 11 visits each year, resulting in 1337 observed detection histories.

Usage

```
data("SwissMarbledWhite")
```

Format

A data frame with 1337 rows and the following columns:

site identification number for site

year the year of the record

Day1, Day2, Day3, Day4, Day5, Day6, Day7, Day8, Day9, Day10, Day11 Julian date of the survey, 1 = 1st April. NA indicates that the date of the survey was not recorded; see Examples for a way to impute dates.

y1, y2, y3, y4, y5, y6, y7, y8, y9, y10, y11 1 if the species was recorded during the survey, 0 otherwise.

Source

Biodiversity Monitoring Program in the canton of Aargau, Switzerland (Courtesy Isabelle Flöss, Abteilung Landschaft und Gewässer des Kanton Aargau). See Appendix C of Roth et al. (2014).

https://figshare.com/articles/Appendix_C_Field_methods_and_analyses_of_the_butterfly_surveys_of_the_Aargau_Biodiversity_Monitoring_Program_/3559965/1

References

Roth, T., Strebel, N. and Amrhein, V. (2014), Estimating unbiased phenological trends by adapting site-occupancy models. *Ecology*, 95: 2144-2154

Kéry, M. & Royle, J.A. (2020) *Applied Hierarchical Modeling in Ecology* AHM2 - 4.13.

See Also

UKmarbledWhite.

Examples

```
data(SwissMarbledWhite)
str(SwissMarbledWhite)

# Impute values for the missing dates:
smw <- SwissMarbledWhite
for(t in 3:13) {
  smw[is.na(smw[, t]), t] <- mean(smw[, t], na.rm=TRUE)
}
summary(smw)
```

SwissSquirrels	*Data for Red Squirrels in Switzerland from the Swiss breeding bird survey MHB*

Description

A file with detection/nondetection data for the Red Squirrel (*Sciurus vulgaris*)in 265 1 km2 survey quadrats in Switzerland for 2007, together with covariates. See Examples for code to load the data.

Format

The file SwissSquirrels.txt is a tab-delimited text file with 265 rows and the following columns:

- spec.name : the species name.
- coordx : the x coordinate of the center of the quadrat; the coordinate reference system intentionally not specified.
- coordy : the y coordinate of the center of the quadrat.
- ele : the mean elevation of the quadrat, m.
- route : the length of the route walked in the quadrat, km.
- forest : percentage forest cover.
- det071, det072, det073 : 1/0 detection data for 3 survey occasions in 2007.
- date071, date072, date073 : Julian date for the 3 survey occasions (1 = 1st April).
- dur071, dur072, dur073 : duration of the 3 survey occasions (mins).

Source

Swiss Ornithological Institute

References

Kéry, M. & Royle, J.A. (2016) *Applied Hierarchical Modeling in Ecology* AHM1 - 10.9.

Examples

```
# To read in the text file:
fn <- file.path(find.package("AHMbook"), "extdata", "SwissSquirrels.txt")
data <- read.table(fn, header = TRUE)
str(data)
```

SwissTits   *Data from the Swiss Breeding Bird Survey for 6 species of tits*

Description

The Swiss breeding bird survey ("Monitoring Häufige Brutvögel" MHB) has monitored the populations of 150 common species since 1999. The MHB sample consists of 267 1-km squares that are laid out as a grid across Switzerland. Fieldwork is conducted by about 200 skilled birdwatchers annually, most of them volunteers. Avian populations are monitored using a simplified territory mapping protocol (also called spot-mapping West of the Atlantic), where each square is surveyed up to three times during the breeding season (only twice above the tree line around 2000 m elevation). Surveys are conducted along a transect that does not change over the years and is 4-6 kms in length on average.

The list SwissTits has the data for six species of tits from 2004 to 2013. There are some missing values: see Details.

Usage

```
data("SwissTits")
```

Format

SwissTits is a list with 5 elements:

species   a data frame with a row for each species and the following columns:

1. specid : a numeric species ID based on phylogeny.
2. latname : a 6-letter abbreviation of the Latin name.
3. name : the English name.

sites   a data frame with a row for each 1x1 km quadrat and the following columns:

1. siteID : an alphanumeric site identifier.
2. coordx : the x coordinate of the center of the quadrat; the coordinate reference system intentionally not specified.
3. coordy : the y coordinate of the center of the quadrat.
4. AQ : an identifier for the 10km x 10km block called "Atlas Quadrat" within which the site falls.
5. AQ.coordx : the x coordinate of the center of the AQ.
6. AQ.coordy : the y coordinate of the center of the AQ.

7. elev : the mean elevation of the quadrat, m.

8. rlength : the length of the route walked in the quadrat, km.

9. forest : percentage forest cover.

counts   a sites x replicates x years x species array of counts

date   a sites x replicates x years array with Julian dates of the surveys, 1 April = 1

dur   a sites x replicates x years array with the duration of each survey, mins

Details

Missing values in the date array indicate that the corresponding survey was not carried out.

On 26 occasions when surveys were carried out, the duration was not recorded, resulting in additional NAs in the dur array.

A new method for recording breeding territories was introduced in 2004, but the old protocol was in use at some sites until 2013. Surveys with the old protocol have the counts shown as NA in the count array.

Note

Sections 6.9.1 and 6.13.1 of the AHM1 book have code to read in data from a CSV file, "SwissTits_mhb_2004_2013.csv". The SwissTits list has all the same data in a more compact format. See Examples for ways to generate the objects used in the book from the list.

Source

Swiss Ornithological Institute

References

Kéry, M. & Royle, J.A. (2016) *Applied Hierarchical Modeling in Ecology* AHM1 - 6.9.1 and 6.13.1.

Examples

```
data(SwissTits)
str(SwissTits)

# Section 6.9.1, foot of p.257 and top of p.258:
# -------------
y0 <- SwissTits$counts[, , '2013', 'Great tit']
( NA.sites <- which(rowSums(is.na(y0)) == 3) ) # Unsurveyed sites
y <- y0[-NA.sites, ]                # Drop them from the count data
tits <- SwissTits$sites[-NA.sites, ] # Also drop from the site covariates
str(y)
# Get date and duration data for 2013, without the NA.sites rows:
date <- SwissTits$date[-NA.sites, , '2013']
dur <- SwissTits$dur[-NA.sites, , '2013']

# Section 6.13.1, p.303
# --------------
# Get the count data for 2013 (all species)
```

```
y0 <- SwissTits$count[, , '2013', ]
str(y0)
# We keep the sites with count data, remove those with 3 NAs
# See which sites have counts in 2013 for (say) Great tits:
keep <- which(rowSums(is.na(y0[, , "Great tit"])) != 3)
length(keep)
y <- y0[keep, , ]
# Get the covariate data for the 'keep' sites
elev <- SwissTits$sites$ele[keep]
route <- SwissTits$sites$rlength[keep]
forest <- SwissTits$sites$forest[keep]
date <- SwissTits$date[keep, , '2013']  # Survey date
dur <- SwissTits$dur[keep, , '2013']    # Survey duration

# Degrade counts to detection/nondetection data
y3DRN <- y
y3DRN[y3DRN > 1] <- 1
str(y3DRN)
# Final detail...
( spec.names <- paste0(SwissTits$species$name, "s") )
```

---

treeSparrow   *American Tree Sparrow data from Alaska*

Description

Mizel et al. (2018) describe the _data on American Tree Sparrows (*Spizella arborea*) that were collected in May and June 2016 at Noatak National Preserve, Alaska. The study area was a 10.8 x 9.6 km rectangle divided up into 288 plots of 600 m x 600 m, of which 150 had some coverage by a transect running through them. Routes were usually chosen to intersect the centroid of the cell and deflected by 0 or 90 degrees when required to move into an adjacent cell. In some cases, a transect was shifted to avoid terrain features. Distance measurements were truncated at 250m from the observer. Each visited plot was surveyed an average of three times (max = 5). Each plot was gridded to a 30-m resolution and spatial attributes were associated with each of these pixels. This resulted in an average of 328 pixels per plot within the buffered transect. Bird locations were assigned to the nearest pixel centroid.

Usage

```
data("treeSparrow")
```

Format

treeSparrow is a list with 3 elements:

surveyData   a data frame with rows for 466 surveys and the following columns:

- Site : a numeric site identifier, 1 to 150
- Visit : a numeric visit identifier, 1 to 5

- Survey : a numeric survey identifier, 1 to 466
- count : the number of tree sparrows detected.
- juldate : the Julian date of the survey, day 1 = 1 Jan.
- effort : time spent in each plot during each visit (in hours)
- reltime : number of hours since 0200 (there is no sunrise during the survey period in the Arctic)
- Noise : background noise (combination of wind, creek, and mosquito-associated noise): 0 = no noise, 1 = slight, 2 = noticeable reduction in hearing, 3 = prohibitive noise
- Observer2 : Observed ID

obs a data frame with rows for 325 birds observed and the following columns:

- Pixel : a numeric pixel identifier, 1 to 49250
- Site : a numeric site identifier, 1 to 150
- Visit : a numeric visit identifier, 1 to 5
- NEAR_DIST : distance between the bird and the transect line
- SurveyID : a numeric survey identifier, 1 to 466
- ind : an index identifying the individual within the survey.

pixels a data frame with rows for 49250 pixels and the following columns:

- pixID : a numeric pixel identifier, 1 to 49250
- Site : a numeric site identifier, 1 to 150
- X : x-coordinate of the pixel
- Y : y-coordinate of the pixel
- NEAR_DIST : distance between the center of the pixel and the transect line
- NDVI : Normalized Difference Vegetation Index
- elev : elevation of the center of the pixel, m.

Source

Mizel et al. (2018)

References

Mizel, J.D., Schmidt, J.H. and Lindberg, M.S. (2018) Accommodating temporary emigration in spatial distance sampling models. *Journal of Applied Ecology*, 55(3), pp.1456-1464.

Kéry, M. & Royle, J.A. (2020) *Applied Hierarchical Modeling in Ecology* AHM2 - 11.8.4.

Examples

```
data(treeSparrow)
str(treeSparrow)
```

---

ttdPeregrine   *Time-to-detection data for peregrines from the French Jura mountains*

Description

Between 7 and 9 March 2015, 38 breeding cliffs were visited up to 3 times to survey for Peregrine
Falcons (*Falco peregrinus*). Observation duration varied from 3 to 95 minutes. The time to detection
was recorded for each bird seen. If no birds were seen, the time was entered as NA.

Usage

```
data("ttdPeregrine")
```

Format

A data frame with 70 rows and the following columns:

Date   a factor with 3 levels giving the date

DayNumber   the number of the day: 1, 2, or 3

SiteNumber   identification number of the site

Start.hour, Start.minute   the time of starting the search

End.hour, End.minute   the time of ending the search

MinOfDay   the time of the start of the search, minutes after 06:00

Tmax   the duration of the search, in minutes

ttd   the time to detection, minutes; NA if no birds were seen during the search

sex   the sex of the birds seen; NA if no birds were seen

Source

Based on the field work of one of the authors.

References

Kéry, M. & Royle, J.A. (2016) *Applied Hierarchical Modeling in Ecology* AHM1 - 10.12.

Examples

```
# Using the built-in data set instead of the TXT file for the example in Section
#  10.12 of the book is easy, as the format is the same as the file.

?ttdPeregrine  # check the description of the data
data(ttdPeregrine)
# Instead of data <- read.table("ttdPeregrine.txt", header = TRUE, sep = "\t") do:
data <- ttdPeregrine

# Then continue with the rest of the analysis on p.618
```

UKmarbledWhite   *Data for UK Marbled White butterflies*

Description

Data from the UK Butterfly Monitoring Scheme (UKBMS) for the Marbled White butterfly (*Melanargia galathea*), a fairly widespread species that flies in a single generation per year in the UK.

Most survey sites are chosen opportunistically and represent fixed transects of variable and often unknown length, which are walked between two and 26 times per year starting in early April. Butterflies are counted within an imaginary 5m x 5m 'moving box' along the transects.

We restrict the data to sites where the marbled white was ever recorded, the years 1991–2015 and those 80 sites where counts took place in at least 20 of these 25 years.

Usage

```
data("UKmarbledWhite")
```

Format

A data frame with 9651 rows and the following columns:

C   count, the number of butterflies recorded

site   identification number for site

year   the year of the record

north   approximate northing of the site in the British National Grid, in meters

date   Julian date of the survey, 1 = 1st January

Source

UK Butterfly Monitoring Scheme (UKBMS) data © copyright and database right Butterfly Conservation, the Centre for Ecology and Hydrology, British Trust for Ornithology, and the Joint Nature Conservation Committee, 2018. The UKBMS is indebted to all volunteers who contribute data to the scheme.

https://www.ceh.ac.uk/our-science/projects/uk-butterfly-monitoring-scheme

References

Kéry, M. & Royle, J.A. (2020) *Applied Hierarchical Modeling in Ecology* AHM2 - 1.8.2.

See Also

SwissMarbledWhite.

Examples

```
data(UKmarbledWhite)
str(UKmarbledWhite)
```

valid_data	*Validates simulated observational data against data-generating values.*

Description

Where species identifications (from photos, recorded calls, etc) is doubtful, it is sometimes feasible for a subset of the data to be validated by experts. Chambert et al (2017) used simulations to investigate the effectiveness of such partial validation. Where a data set with false positives has been simulated and the true status is known, valid_data compares a subset with truth and returns the number of detections checked per site and the number checked and found to be valid. The function attempts to validate equal numbers of detections at each site (not equal proportions) subject to the actual number of detections and the total number to be validated; if the number of detections is small, all will be validated, otherwise a random sample is used; see Examples.

Usage

```
valid_data(N, tp, n.valid, prop.valid=FALSE)
```

Arguments

N	a vector with the number of detections for each site, including true positives and false positives.
tp	a vector with the number of true positive detections for each site.
n.valid	the number of detections to be validated (if prop.valid=FALSE) or the proportion of detections to be validated (if prop.valid=TRUE).
prop.valid	logical; determines how n.valid is interpreted.

Value

A list with components:

n	a vector with the number of detections checked for each site.
k	a vector with the number of detections checked and found to be valid (true positives) for each site.

Author(s)

Mike Meredith.

References

Chambert, T., Waddle, J.H., Miller, D.A.W., Walls, S.C., & Nichols, J.D. (2017) A new framework for analysing automated acoustic species-detection data: occupancy estimation and optimization of recordings post-processing. *Methods in Ecology and Evolution*, 9, 560-570.

Kéry, M. & Royle, J.A. (2020) *Applied Hierarchical Modeling in Ecology* AHM2 - 7.6.2.

Examples

```
# Generate some data for 100 sites:
set.seed(42, kind="Mersenne-Twister")
z <- rbinom(100, 1, 0.7)  # z[i] == 1 if species present at site 1, 0 otherwise
tp <- rpois(100, 3*z)     # Number of true detections, 0 if species not present
N <- tp + rpois(100, 0.5) # Add false positives

# Validate a subset of 150 detections:
out <- valid_data(N, tp, 150)
head(tmp <- cbind(z=z, tp=tp, N=N, n=out$n, k=out$k), 10)
colSums(tmp)

# Plot the number validated vs all detections:
graphics::sunflowerplot(N, out$n, xlab="Number of detections at each site",
  ylab="Number validated")
  # For sites with <= 2 detections, all are validated; otherwise at least 2 are validated,
  #  with a 3rd draw from randomly selected sites to bring the total to 150.
```

---

wagtail   *Data for Dutch wagtails*

Description

The Dutch Centre for Field Ornithology (Sovon) monitored grassland birds in Flevoland between April and mid-July 2011. 235 points were surveyed on up to 4 occasions; observations were divided into distance classes 50m wide, and the number of observations in each class recorded. These data are for the Yellow Wagtail (*Motacilla flava*).

Usage

```
data("wagtail")
```

Format

wagtail is a list with 8 elements:

potato for each point, the percentage of the area which are potato fields.

grass for each point, the percentage of the area under permanent grassland.

lscale for each point, an index of whether the landscape is open (0) or closed (100).

date a points x occasions matrix, the Julian date of the survey.

hour a points x occasions matrix, the hour of the survey.

breaks the boundaries between the distance classes; birds more than 300m from the point were not included.

Y a matrix of counts, with a row for each site; columns 1 to 6 give the counts in the distance classes for the 1st survey occasion, columns 7 to 12 for the 2nd occasion, and so on (this is the format required for unmarkedFrameGDS).

rep a points x occasions character matrix with the occasion number (this is used as a categorical variable in the analysis).

Source

Dutch Centre for Field Ornithology (SOVON)

References

Kéry, M. & Royle, J.A. (2016) *Applied Hierarchical Modeling in Ecology* AHM1 - 9.5.3.

Examples

```
data(wagtail)
str(wagtail)
```

---

waterVoles   *Data for the Mighty Water Voles of Scotland, UK.*

Description

Habitat patches suitable for Water Voles (*Arvicola amphibius*) were surveyed for vole latrines (fecal deposits used for territory marking), used to indicate presence. It is suspected that some of the detections recorded may be false positives. The data set covers 114 patches surveyed up to 3 times per year from 2009 to 2011.

Usage

```
data("waterVoles")
```

Format

waterVoles is a data frame with 332 rows corresponding to surveys, and 5 columns:

Patch an alphanumeric site identifier.

y1, y2, y3 detection (1)/non-detection (0).

Year the year of the survey.

Source

Chris Sutherland, Xavier Lambin and colleagues.

References

Sutherland, C., Elston, D.A., & Lambin, X. (2012) Multi-scale processes in metapopulations: effects of stage structure, rescue effect and correlated extinctions. *Ecology*, 93, 2465-2473.

Sutherland, C.S., Elston, D.A., & Lambin, X. (2014) A demographic, spatially explicit patch occupancy model of metapopulation dynamics and persistence. *Ecology*, 95, 3149-3160.

Kéry, M. & Royle, J.A. (2020) *Applied Hierarchical Modeling in Ecology* AHM2 - 7.2.2.

Examples

```
data(waterVoles)
str(waterVoles)
```

---

wigglyLine	*Coordinates for a wiggly line transect*

Description

Coordinates for a wiggly line transect used as an example in Kéry & Royle (2020) chapter 11.9.

Usage

```
data("wigglyLine")
```

Format

wigglyLine is a 2-column matrix of x and y coordinates.

Source

Made-up data.

References

Kéry, M. & Royle, J.A. (2020) *Applied Hierarchical Modeling in Ecology* AHM2 - 11.

Examples

```
data(wigglyLine)
str(wigglyLine)
plot(wigglyLine, type='l')
```

---

wigglyOcc	*Simulate static occupancy data*

Description

Function to generate a static occupancy data set with really wiggly covariate relationships in occupancy and detection probability.

To recreate the data sets used in the book with R 3.6.0 or later, include sample.kind="Rounding" in the call to set.seed. This should only be used for reproduction of old results.

Usage

```
wigglyOcc(seed = 1, show.plot = TRUE, verbose = TRUE)
```

Arguments

`seed`	Seed for random number generator
`show.plot`	choose whether to show plots or not. Set to FALSE when using function in simulations.
`verbose`	if TRUE, output will be written to the Console.

Value

A list with the following elements:

`M`	Number of sites
`J`	Number of replicate surveys
`Xsite`	Simulated site covariate, a vector of length M
`Xsurvey`	Simulated survey covariate, a M x J matrix
`psi`	Occupancy probability, a vector of length M
`z`	Realized occupancy, a 0/1 vector of length M
`p`	Detection probability, a M x J matrix
`y`	detection history, a M x J matrix of 0/1

Author(s)

Marc Kéry & Andy Royle

References

Kéry, M. & Royle, J.A. (2016) *Applied Hierarchical Modeling in Ecology* AHM1 - 10.14.

Examples

```
# Generate data with the default arguments and look at the structure:
tmp <- wigglyOcc()
str(tmp)
```

---

willowWarbler	*Constant effort site (CES) data for British Willow Warblers*

Description

British Willow Warblers (Phylloscopus trochilus) caught at a total of 193 ringing (or banding) stations in England, Wales and Scotland. The willow warbler is a small (8–10 g) migratory species that used to be extremely widespread and common throughout Britain, but whose populations have suffered severe declines since the mid-90s. We analyse data from 11 years (1986–1996) before the major population decline.

Usage

```
data("willowWarbler")
```

Format

willowWarbler is a list with 4 elements:

birds a data frame with rows for 10551 birds and the following columns:

1. 1986-1996 : capture history, 1 if the bird was captured that year, 0 otherwise.
2. cesID : numerical code for the Constant Effort Site where the bird was caught.

cells a data frame with rows for 9667 5x5 km cells covering the whole of Britain, and the following columns:

1. lon, lat : easting and northing of the center of the cell.
2. gdd : mean growing degree days (GDD: the sum of daily mean temperatures above 5.5C).
3. blockID : the ID of the 25x25 km block into which the cell falls.

CES a data frame with rows for each of the Constant Effort Sites, and the following columns:

1. cesx, cesy : approximate easting and northing of the site.
2. blockID : the ID of the 25x25 km block into which the site falls.
3. cellID : the ID of the 5x5 km cell into which the site falls.

blocks a data frame with rows for 495 25x25 km blocks covering the whole of Britain, and the following columns:

1. blockX, blockY : easting and northing of the center of the block.

Source

British Ornithological Trust (BTO)

References

Kéry, M. & Royle, J.A. (2020) *Applied Hierarchical Modeling in Ecology* AHM2 - 3.4.

Examples

```
data(willowWarbler)
str(willowWarbler)
attach(willowWarbler)
ch <- as.matrix(birds[ , 1:11]) # extract capture histories.
```

---

zinit   *Starting values for survival analysis in JAGS/BUGS*

Description

Generates a matrix for use as starting values for the survival indicator in a JAGS or BUGS model for survival analysis, traditionally designated z.

Usage

```
zinit(CH)
```

Arguments

CH   An individuals x time matrix of capture records, 1 if captured, 0 otherwise, no
                     missing values.

Value

An individuals x time matrix with 1 in each row after the first capture; all other elements NA, including the first capture occasion.

Author(s)

Marc Kéry & Andy Royle

References

Kéry, M. & Royle, J.A. (2020) *Applied Hierarchical Modeling in Ecology* AHM2 - 3.2.3.

Examples

```
# Generate a fake capture history
( ch <- matrix(rbinom(60, 1, 0.5), 6, 5) )
zinit(ch)
```

Index