Package ‘ASMbook’

May 6, 2026

Version 1.0.2

Date 2024-07-29

Title Functions for the Book ``Applied Statistical Modeling for Ecologists''

Depends R (>= 3.4.0)

Imports coda, lattice, MASS, stats, TMB

Suggests roxygen2

Description Provides functions to accompany the book ``Applied Statistical Modeling for Ecologists'' by Marc Kéry and Kenneth F. Kellner (2024, ISBN: 9780443137150). Included are functions for simulating and customizing the datasets used for the example models in each chapter, summarizing output from model fitting engines, and running custom Markov Chain Monte Carlo.

License GPL-3

NeedsCompilation no

Encoding UTF-8

URL `https://shop.elsevier.com/books/applied-statistical-modelling-for-ecologists/kery/978-0-443-13715-0`

BugReports `https://github.com/kenkellner/ASMbook/issues`

RoxygenNote 7.3.2

Author Marc Kéry [aut],
Ken Kellner [cre, aut]

Maintainer Ken Kellner `<contact@kenkellner.com>`

Repository CRAN

Date/Publication 2024-07-31 10:20:27 UTC

Contents

`demoMCMC` *Fit a Poisson GLM with MCMC*

Description

This is a demo function that fits a Poisson GLM with one continuous covariate to some data (y, x) using a random-walk Metropolis Markov chain Monte Carlo algorithm.

Usage

```
demoMCMC(
  y,
  x,
  true.vals = c(2.5, 0.14),
  inits = c(0, 0),
  prior.sd.alpha = 100,
  prior.sd.beta = 100,
  tuning.params = c(0.1, 0.1),
  niter = 10000,
  nburn = 1000,
  quiet = TRUE,
```

```
  show.plots = TRUE
)
```

Arguments

`y`	A vector of counts, e.g., y in the Swiss bee-eater example
`x`	A vector of a continuous explanatory variable, e.g. year x in the bee-eaters
`true.vals`	True intercept and slope if known (i.e., when run on simulated data)
`inits`	Initial values in the MCMC algorithm for alpha, beta
`prior.sd.alpha`	SD of normal prior for alpha
`prior.sd.beta`	SD of normal prior for beta
`tuning.params`	SD of the Gaussian proposal distributions for alpha, beta
`niter`	Total chain length (before burnin)
`nburn`	Burn-in length
`quiet`	Logical, suppress console output
`show.plots`	Logical, should diagnostic plots be shown?

Value

A list containing input settings, acceptance probabilities and MCMC samples.

Author(s)

Marc Kéry

Examples

```
# Load the real data used in the publication by Mueller (Vogelwarte, 2021)

# Counts of known pairs in the country 1990-2020
y <- c(0,2,7,5,1,2,4,8,10,11,16,11,10,13,19,31,
      20,26,19,21,34,35,43,53,66,61,72,120,102,159,199)
year <- 1990:2020     # Define year range
x <- (year-1989)      # Scaled, but not centered year as a covariate
x <- x-16             # Now it's centered

oldpar <- par(no.readonly = TRUE)
par(mfrow = c(1, 2), mar = c(5,5,5,2), cex.lab = 1.5, cex.axis = 1.5, cex.main = 1.5)
plot(table(y), xlab = 'Count (y)', ylab = 'Frequency', frame = FALSE,
   type = 'h', lend = 'butt', lwd = 5, col = 'gray20', main = 'Frequency distribution of counts')
plot(year, y, xlab = 'Year (x)', ylab = 'Count (y)', frame = FALSE, cex = 1.5,
     pch = 16,  col = 'gray20', main = 'Relationship y ~ x')
fm <- glm(y ~ x, family = 'poisson')        # Add Poisson GLM line of best fit
lines(year, predict(fm, type = 'response'), lwd = 3, col = 'red', lty = 3)


# Execute the function with default function args
# In a real test you should run more iterations
```

```
par(mfrow = c(1,1))
str(tmp <- demoMCMC(niter=100, nburn=50))

# Use data created above
par(mfrow = c(1,1))
str(tmp <- demoMCMC(y = y, x = x, niter=100, nburn=50))


par(oldpar)
```

`getMLE` *Print Estimates, Standard Errors, and 95% Wald-type Confidence Intervals From optim Output*

Description

Print Estimates, Standard Errors, and 95% Wald-type Confidence Intervals From optim Output

Usage

```
getMLE(opt, dig = 3)

get_MLE(opt, dig = 3)
```

Arguments

`opt`	Object resulting from a call to `optim`
`dig`	Number of decimal places to use when printing

Value

A matrix of parameter estimates, standard errors, and 95% Wald-type confidence intervals.

Author(s)

Marc Kéry, Ken Kellner

`nimbleSummary` *Summarize MCMC Samples in an mcmc.list Object Created by NIMBLE*

Description

Summarize MCMC Samples in an mcmc.list Object Created by NIMBLE

Usage

```
nimbleSummary(samples, params = NULL)

nimble_summary(samples, params = NULL)
```

Arguments

`samples`	An object of class `mcmc.list`
`params`	An optional list of the parameter names used to sort the output

Value

A data frame of summary information for each saved parameter

Author(s)

Ken Kellner

`simDat102` *Simulate data for Chapter 10.2: Linear mixed-effects model*

Description

Simulate mass ~ length regressions in 56 populations of snakes with random population effects for intercepts and slopes. There is no correlation between the intercept and slope random variables.

Usage

```
simDat102(
  nPops = 56,
  nSample = 10,
  mu.alpha = 260,
  sigma.alpha = 20,
  mu.beta = 60,
  sigma.beta = 30,
  sigma = 30
)
```

Arguments

`nPops`	Number of populations
`nSample`	Samples from each population
`mu.alpha`	Mean of random intercepts
`sigma.alpha`	SD of random intercepts
`mu.beta`	Mean of random slopes
`sigma.beta`	SD of random slopes
`sigma`	Residual standard deviation

Value

A list of simulated data and parameters.

`nPops`	Number of populations
`nSample`	Number of samples per population
`mu.alpha`	Mean of random intercepts
`sigma.alpha`	SD of random intercepts
`mu.beta`	Mean of random slopes
`sigma.beta`	SD of random slopes
`sigma`	Residual SD
`pop`	Indicator for population number
`orig.length`	Snake body length, not standardized
`lengthN`	Snake body length, standardized
`alpha`	Random intercepts
`beta`	Random slopes
`eps`	Residuals
`mass`	Simulated body mass for each snake

Author(s)

Marc Kéry

Examples

```
library(lattice)
str(dat <- simDat102())      # Implicit default arguments
xyplot(dat$mass ~ dat$lengthN | dat$pop, xlab = 'Length', ylab = 'Mass',
       main = 'Realized mass-length relationships', pch = 16, cex = 1.2,
       col = rgb(0, 0, 0, 0.4))

# Fewer populations, more snakes (makes patterns perhaps easier to see ?)
str(dat <- simDat102(nPops = 16, nSample = 100))
xyplot(dat$mass ~ dat$lengthN | dat$pop, xlab = 'Length', ylab = 'Mass',
       main = 'Realized mass-length relationships
```

```
       (default random-coefficients model)',
       pch = 16, cex = 1.2, col = rgb(0, 0, 0, 0.4))

# Revert to random intercept model (and less residual variation), fewer pops
# and more snakes. Increased sigma.alpha to emphasize the random intercepts part
str(dat <- simDat102(nPops = 16, nSample = 100, sigma.alpha = 50, sigma.beta = 0, sigma = 10))
xyplot(dat$mass ~ dat$lengthN | dat$pop, xlab = 'Length', ylab = 'Mass',
       main = 'Realized mass-length relationships (random-intercepts model)',
       pch = 16, cex = 1.2, col = rgb(0, 0, 0, 0.4))

# Revert to random-effects one-way ANOVA model, only random intercepts, but zero slopes
str(dat <- simDat102(nPops = 16, nSample = 100, sigma.alpha = 50,
                     mu.beta = 0, sigma.beta = 0, sigma = 10))
xyplot(dat$mass ~ dat$lengthN | dat$pop, xlab = 'Length', ylab = 'Mass',
       main = 'Realized mass-length relationships
       (one-way ANOVA model with random pop effects)',
       pch = 16, cex = 1.2, col = rgb(0, 0, 0, 0.4))

# Revert to simple linear regression (= no effects of pop on either intercepts or slopes)
str(dat <- simDat102(nPops = 16, nSample = 100, sigma.alpha = 0, sigma.beta = 0, sigma = 10))
xyplot(dat$mass ~ dat$lengthN | dat$pop, xlab = 'Length', ylab = 'Mass',
       main = 'Realized mass-length relationships
       (de-facto a simple linear regression now)',
       pch = 16, cex = 1.2, col = rgb(0, 0, 0, 0.4))

# Revert to "model-of-the-mean": no effects of either population or body length
str(dat <- simDat102(nPops = 16, nSample = 100, sigma.alpha = 0, mu.beta = 0,
                     sigma.beta = 0, sigma = 10))
xyplot(dat$mass ~ dat$lengthN | dat$pop, xlab = 'Length', ylab = 'Mass',
       main = 'Realized mass-length relationships
       ("model-of-the-mean", no effects of pop or length)',
       pch = 16, cex = 1.2, col = rgb(0, 0, 0, 0.4))
```

`simDat105` *Simulate data for Chapter 10.5: Linear mixed-effects model with correlation between intercepts and slopes*

Description

Simulate mass ~ length regressions in 56 populations of snakes with random population effects for intercepts and slopes. Note that now there is a correlation between the intercept and slope random variables.

Usage

```
simDat105(
  nPops = 56,
  nSample = 10,
  mu.alpha = 260,
```

```
  sigma.alpha = 20,
  mu.beta = 60,
  sigma.beta = 30,
  cov.alpha.beta = -50,
  sigma = 30
)
```

Arguments

`nPops`	Number of populations
`nSample`	Samples from each population
`mu.alpha`	Mean of random intercepts
`sigma.alpha`	SD of random intercepts
`mu.beta`	Mean of random slopes
`sigma.beta`	SD of random slopes
`cov.alpha.beta`	Covariance between alpha and beta
`sigma`	Residual standard deviation

Value

A list of simulated data and parameters.

`nPops`	Number of populations
`nSample`	Number of samples per population
`mu.alpha`	Mean of random intercepts
`sigma.alpha`	SD of random intercepts
`mu.beta`	Mean of random slopes
`sigma.beta`	SD of random slopes
`cov.alpha.beta`	Covariance betwen alpha and beta
`sigma`	Residual SD
`pop`	Indicator for population number
`orig.length`	Snake body length, not standardized
`lengthN`	Snake body length, standardized
`ranef.matrix`	Random effects matrix
`alpha`	Random intercepts
`beta`	Random slopes
`eps`	Residuals
`mass`	Simulated body mass for each snake

Author(s)

Marc Kéry

Examples

```
library(lattice)
str(dat <- simDat105())      # Implicit default arguments
xyplot(dat$mass ~ dat$lengthN | dat$pop, xlab = 'Length', ylab = 'Mass',
       main = 'Realized mass-length relationships', pch = 16, cex = 1.2,
       col = rgb(0, 0, 0, 0.4))

# Fewer populations, more snakes (makes patterns perhaps easier to see)
str(dat <- simDat105(nPops = 16, nSample = 100))
xyplot(dat$mass ~ dat$lengthN | dat$pop, xlab = 'Length', ylab = 'Mass',
       main = 'Realized mass-length relationships (random-coef model
       intercept-slope correlation)',
       pch = 16, cex = 1.2, col = rgb(0, 0, 0, 0.4))

# Revert to simpler random-coefficient model without correlation between intercepts and slopes
# (that means to set to zero the covariance term)
str(dat <- simDat105(nPops = 16, nSample = 100, cov.alpha.beta = 0))
xyplot(dat$mass ~ dat$lengthN | dat$pop, xlab = 'Length', ylab = 'Mass',
       main = 'Realized mass-length relationships
       (random-coefficients model without correlation)',
       pch = 16, cex = 1.2, col = rgb(0, 0, 0, 0.4))

# Revert to even simpler random-intercepts model without correlation between intercepts and slopes
# (that means to set to zero the covariance term and the among-population variance of the slopes)
# Note that sigma.beta = 0 and non-zero covariance crashes owing to non-positive-definite VC matrix
str(dat <- simDat105(nPops = 16, nSample = 100, sigma.alpha = 50, sigma.beta = 0,
    cov.alpha.beta = 0, sigma = 10))
xyplot(dat$mass ~ dat$lengthN | dat$pop, xlab = 'Length', ylab = 'Mass',
       main = 'Realized mass-length relationships\n(random-intercepts model)',
       pch = 16, cex = 1.2, col = rgb(0, 0, 0, 0.4))

# Revert to random-effects one-way ANOVA model, only random intercepts, but zero slopes
str(dat <- simDat105(nPops = 16, nSample = 100, sigma.alpha = 50, mu.beta = 0, sigma.beta = 0,
    cov.alpha.beta = 0, sigma = 10))
xyplot(dat$mass ~ dat$lengthN | dat$pop, xlab = 'Length', ylab = 'Mass',
       main = 'Realized mass-length relationships
       (one-way ANOVA model with random pop effects)',
       pch = 16, cex = 1.2, col = rgb(0, 0, 0, 0.4))

# Revert to simple linear regression (= no effects of pop on either intercepts or slopes)
str(dat <- simDat105(nPops = 16, nSample = 100, sigma.alpha = 0, sigma.beta = 0,
    cov.alpha.beta = 0, sigma = 10))
xyplot(dat$mass ~ dat$lengthN | dat$pop, xlab = 'Length', ylab = 'Mass',
       main = 'Realized mass-length relationships
       (this is de-facto a simple linear regression now)',
       pch = 16, cex = 1.2, col = rgb(0, 0, 0, 0.4))

# Revert to "model-of-the-mean": no effects of either population or body length
str(dat <- simDat105(nPops = 16, nSample = 100, sigma.alpha = 0, mu.beta = 0, sigma.beta = 0,
    cov.alpha.beta = 0, sigma = 10))
xyplot(dat$mass ~ dat$lengthN | dat$pop, xlab = 'Length', ylab = 'Mass',
       main = 'Realized mass-length relationships
```

```
        ("model-of-the-mean" with no effects of pop or length)',
        pch = 16, cex = 1.2, col = rgb(0, 0, 0, 0.4))
```

`simDat11` *Simulate data for Chapter 11: Comparing two groups of Poisson counts*

Description

Generate counts of hares in two areas with different landuse

Usage

```
simDat11(nSites = 30, alpha = log(2), beta = log(5) - log(2))
```

Arguments

`nSites`	Number of sites
`alpha`	Intercept
`beta`	Slope for land use

Value

A list of simulated data and parameters.

`nSites`	Number of sites
`alpha`	Intercept
`beta`	Slope for land use
`y`	Simulated hare counts

Author(s)

Marc Kéry

Examples

```
str(dat <- simDat11())      # Implicit default arguments

# Revert to "Poisson model-of-the-mean"
# (Increase sample size to reduce sampling variability)
str(dat <- simDat11(nSites = 1000, beta = 0))
```

`simDat122` *Simulate data for Chapter 12.2: Overdispersed counts*

Description

Generate counts of hares in two landuse types when there may be overdispersion relative to a Poisson

Usage

```
simDat122(nSites = 50, alpha = log(2), beta = log(5) - log(2), sd = 0.5)
```

Arguments

`nSites`	Number of sites
`alpha`	Intercept
`beta`	Slope for land use
`sd`	Standard deviation for overdispersion

Value

A list of simulated data and parameters.

`nSites`	Number of sites
`alpha`	Intercept
`beta`	Slope for land use
`sd`	Standard deviation for overdispersion
`C_OD`	Simulated hare counts with overdispersion
`C_Poisson`	Simulated hare counts without overdispersion

Author(s)

Marc Kéry

Examples

```
str(dat <- simDat122())      # Implicit default arguments

# Much greater OD to emphasize patterns (also larger sample size)
str(dat <- simDat122(nSites = 100, sd = 1))

# Revert to "Poisson model-of-the-mean" (i.e., without an effect of landuse type)
str(dat <- simDat122(nSites = 100, beta = 0, sd = 1))
```

`simDat123` *Simulate data for Chapter 12.3: Zero-inflated counts*

Description

Generate counts of hares in two landuse types when there may be zero-inflation (this is a simple general hierarchical model, see Chapters 19 and 19B in the book)

Usage

```
simDat123(nSites = 50, alpha = log(2), beta = log(5) - log(2), psi = 0.2)
```

Arguments

`nSites`	Number of sites
`alpha`	Intercept
`beta`	Slope for land use
`psi`	Zero inflation parameter (probability of structural 0)

Value

A list of simulated data and parameters.

`nSites`	Number of sites
`alpha`	Intercept
`beta`	Slope for land use
`psi`	Zero inflation parameter
`w`	Indicator that count is not a structural 0
`C`	Simulated hare counts with zero inflation

Author(s)

Marc Kéry

Examples

```
str(dat <- simDat123())      # Implicit default arguments

# Drop zero inflation (and make sample sizes bigger)
str(dat <- simDat123(nSites = 1000, psi = 0))    # Note 0 % of the sites have structural zeroes now

# Half of all sites have structural zeroes
str(dat <- simDat123(nSites = 1000, psi = 0.5))

# Revert to "model-of-the-mean" without zero inflation
# 0 % of the sites have structural zeroes
```

```
str(dat <- simDat123(nSites = 1000, beta = 0, psi = 0))

# Revert to "model-of-the-mean" with zero inflation
# 50 % of the sites have structural zeroes
str(dat <- simDat123(nSites = 1000, beta = 0, psi = 0.5))
```

`simDat124` *Simulate data for Chapter 12.4: Counts with offsets*

Description

Generate counts of hares in two landuse types when study area size A varies and is used as an offset

Usage

```
simDat124(nSites = 50, alpha = log(2), beta = log(5) - log(2))
```

Arguments

`nSites`	Number of sites
`alpha`	Intercept
`beta`	Slope for land use

Value

A list of simulated data and parameters.

`nSites`	Number of sites
`alpha`	Intercept
`beta`	Slope for land use
`A`	Site areas
`C`	Simulated hare counts

Author(s)

Marc Kéry

Examples

```
str(dat <- simDat124())      # Implicit default arguments
str(dat <- simDat124(nSites = 1000, beta = 0)) # "Model-of-the-mean" without effect of landuse
str(dat <- simDat124(nSites = 100, alpha = log(2), beta = -2)) # Grassland better than arable
```

`simDat13` *Simulate data for Chapter 13: Poisson ANCOVA*

Description

Simulate parasite load ~ size regressions in 3 populations of goldenring dragonflies

Usage

```
simDat13(nPops = 3, nSample = 100, beta.vec = c(-2, 1, 2, 4, -2, -5))
```

Arguments

`nPops`	Number of populations
`nSample`	Number of samples per population
`beta.vec`	Vector of regression coefficients

Value

A list of simulated data and parameters.

`nPops`	Number of populations
`nSample`	Number of samples per population
`beta`	Vector of regression coefficients
`x`	Indicator for population number
`pop`	Population name (factor)
`orig.length`	Wing length, non-centered
`wing.length`	Wing length, centered
`load`	Simulated parasite loads

Author(s)

Marc Kéry

Examples

```
str(dat <- simDat13())      # Implicit default arguments

# Revert to main-effects model with parallel lines on the log link scale
str(dat <- simDat13(nSample = 100, beta.vec = c(-2, 1, 2, 4, 0, 0)))

# Same with less strong regression coefficient
str(dat <- simDat13(nSample = 100, beta.vec = c(-2, 1, 2, 3, 0, 0)))

# Revert to simple linear Poisson regression: no effect of population (and less strong coefficient)
str(dat <- simDat13(nSample = 100, beta.vec = c(-2, 0, 0, 3, 0, 0)))
```

```
# Revert to one-way ANOVA Poisson model: no effect of wing length
# (Choose larger sample size and greater differences in the intercepts to better show patterns)
str(dat <- simDat13(nSample = 100, beta.vec = c(-1, 3, 5, 0, 0, 0)))

# Revert to Poisson "model-of-the-mean": no effects of either wing length or population
# Intercept chosen such that average parasite load is 10
str(dat <- simDat13(nSample = 100, beta.vec = c(log(10), 0, 0, 0, 0, 0)))
mean(dat$load)        # Average is about 10
```

`simDat14` *Simulate data for Chapter 14: Poisson GLMM*

Description

Simulate count ~ year regressions in 16 populations of red-backed shrikes

Usage

```
simDat14(
  nPops = 16,
  nYears = 30,
  mu.alpha = 3,
  sigma.alpha = 1,
  mu.beta = -2,
  sigma.beta = 0.6
)
```

Arguments

`nPops`	Number of populations
`nYears`	Number of years sampled in each population
`mu.alpha`	Mean of random intercepts
`sigma.alpha`	SD of random intercepts
`mu.beta`	Mean of random slopes
`sigma.beta`	SD of random slopes

Value

A list of simulated data and parameters.

`nPops`	Number of populations
`nYears`	Number of years sampled
`mu.alpha`	Mean of random intercepts
`sigma.alpha`	SD of random intercepts

`mu.beta`	Mean of random slopes
`sigma.beta`	SD of random slopes
`pop`	Population index
`orig.year`	Year values, non-scaled
`year`	Year values, scaled to be between 0 and 1
`alpha`	Random intercepts
`beta`	Random slopes
`C`	Simulated shrike counts

Author(s)

Marc Kéry

Examples

```
library(lattice)
str(dat <- simDat14())
xyplot(dat$C ~ dat$orig.year | dat$pop, ylab = "Red-backed shrike counts", xlab = "Year", pch = 16,
       cex = 1.2, col = rgb(0, 0, 0, 0.4),
       main = 'Realized population trends\n(random-coefficients model)') # works

# Revert to random intercept model. Increased sigma.alpha to emphasize the random intercepts part
str(dat <- simDat14(nPops = 16, sigma.alpha = 1, sigma.beta = 0))
xyplot(dat$C ~ dat$orig.year | dat$pop, ylab = "Red-backed shrike counts", xlab = "Year",
       pch = 16, cex = 1.2, col = rgb(0, 0, 0, 0.4),
       main = 'Realized population trends (random-intercepts model)')

# Revert to random-effects one-way Poisson ANOVA model: random intercepts, but zero slopes
str(dat <- simDat14(nPops = 16, sigma.alpha = 1, mu.beta = 0, sigma.beta = 0))
xyplot(dat$C ~ dat$orig.year | dat$pop, ylab = "Red-backed shrike counts", xlab = "Year",
       pch = 16, cex = 1.2, col = rgb(0, 0, 0, 0.4),
       main = 'Realized population trends
       (random-effects, one-way Poisson ANOVA model)')

# Revert to simple log-linear Poisson regression (no effects of pop on intercepts or slopes)
str(dat <- simDat14(nPops = 16, sigma.alpha = 0, sigma.beta = 0))
xyplot(dat$C ~ dat$orig.year | dat$pop, ylab = "Red-backed shrike counts",
       xlab = "Year", pch = 16, cex = 1.2, col = rgb(0, 0, 0, 0.4),
       main = 'Realized population trends\n(simple log-linear Poisson regression)')

# Revert to Poisson "model-of-the-mean": no effects of either population or body length
str(dat <- simDat14(nPops = 16, sigma.alpha = 0, mu.beta = 0, sigma.beta = 0))
xyplot(dat$C ~ dat$orig.year | dat$pop, ylab = "Red-backed shrike counts",
       xlab = "Year", pch = 16, cex = 1.2, col = rgb(0, 0, 0, 0.4),
       main = 'Realized population trends\n(Poisson "model-of-the-mean")')
```

---

simDat15 *Simulate data for Chapter 15: Comparing two groups of binomial counts*

Description

Generate presence/absence data for two gentian species (Bernoulli variant)

Usage

```
simDat15(N = 50, theta.cr = 12/50, theta.ch = 38/50)
```

Arguments

`N`	Number of sites
`theta.cr`	Probability of presence for cross-leaved gentian
`theta.ch`	Probability of presence for chiltern gentian

Value

A list of simulated data and parameters.

`N`	Number of sites
`theta.cr`	Probability for cross-leaved gentian
`theta.ch`	Probability for chiltern gentian
`y`	Simulated presence/absence data
`species.long`	Species indicator (longform), 1 = chiltern
`C`	Aggregated presence/absence data
`species`	Species indicator for aggregated data
`chiltern`	Effect of chiltern (difference in species intercepts)

Author(s)

Marc Kéry

Examples

```
str(dat <- simDat15())      # Implicit default arguments

# Revert to "Binomial model-of-the-mean"
# (Increase sample size to reduce sampling variability)
str(dat <- simDat15(N = 100, theta.cr = 40/100, theta.ch = 40/100))
```

simDat16 *Simulate data for Chapter 16: Binomial ANCOVA*

Description

Simulate Number black individuals ~ wetness regressions in adders in 3 regions

Usage

```
simDat16(nRegion = 3, nSite = 10, beta.vec = c(-4, 1, 2, 6, 2, -5))
```

Arguments

`nRegion`	Number of regions
`nSite`	Number of sites per region
`beta.vec`	Vector of regression coefficients

Value

A list of simulated data and parameters.

`nRegion`	Number of regions
`nSite`	Number of sites per region
`beta`	Vector of regression coefficients
`x`	Indicator for region number
`region`	Region name (factor)
`wetness`	Wetness covariate
`N`	Number of adders captured at each site
`C`	Number of black adders captured at each site

Author(s)

Marc Kéry

Examples

```
str(dat <- simDat16())      # Implicit default arguments

# Revert to main-effects model with parallel lines on the logit link scale
# (also larger sample size to better see patterns)
str(dat <- simDat16(nSite = 100, beta.vec = c(-4, 1, 2, 6, 0, 0)))

# Same with less strong logistic regression coefficient
str(dat <- simDat16(nSite = 100, beta.vec = c(-4, 1, 2, 3, 0, 0)))

# Revert to simple logit-linear binomial regression: no effect of pop (and weaker coefficient)
```

```
str(dat <- simDat16(nSite = 100, beta.vec = c(-4, 0, 0, 3, 0, 0)))

# Revert to one-way ANOVA binomial model: no effect of wetness
# (Choose greater differences in the intercepts to better show patterns)
str(dat <- simDat16(nSite = 100, beta.vec = c(-2, 2, 3, 0, 0, 0)))

# Revert to binomial "model-of-the-mean": no effects of either wetness or population
# Intercept chosen such that average proportion of black adders is 0.6
str(dat <- simDat16(nSite = 100, beta.vec = c(qlogis(0.6), 0, 0, 0, 0, 0)))
mean(dat$C / dat$N)        # Average is about 0.6
```

`simDat17` *Simulate data for Chapter 17: Binomial GLMM*

Description

Simulate Number of successful pairs ~ precipitation regressions in 16 populations of woodchat shrikes

Usage

```
simDat17(
  nPops = 16,
  nYears = 10,
  mu.alpha = 0,
  mu.beta = -2,
  sigma.alpha = 1,
  sigma.beta = 1
)
```

Arguments

`nPops`	Number of populations
`nYears`	Number of years sampled in each population
`mu.alpha`	Mean of random intercepts
`mu.beta`	Mean of random slopes
`sigma.alpha`	SD of random intercepts
`sigma.beta`	SD of random slopes

Value

A list of simulated data and parameters.

`nPops`	Number of populations
`nYears`	Number of years sampled

`mu.alpha`	Mean of random intercepts
`sigma.alpha`	SD of random intercepts
`mu.beta`	Mean of random slopes
`sigma.beta`	SD of random slopes
`pop`	Population index
`precip`	Precipitation covariate values
`alpha`	Random intercepts
`beta`	Random slopes
`N`	Number of shrike pairs at each site
`C`	Number of successful shrike pairs at each site

Author(s)

Marc Kéry

Examples

```
library(lattice)
str(dat <- simDat17())    # Implicit default arguments (DOES NOT PRODUCE PLOT FOR SOME REASON)
xyplot(dat$C/dat$N ~ dat$precip | dat$pop, ylab = "Realized woodchat shrike breeding success ",
     xlab = "Spring precipitation index", main = "Realized breeding success", pch = 16, cex = 1.2,
      col = rgb(0, 0, 0, 0.4))

# Revert to random intercept model. Increased sigma.alpha to emphasize the random intercepts part
str(dat <- simDat17(nPops = 16, sigma.alpha = 1, sigma.beta = 0))
xyplot(dat$C/dat$N ~ dat$precip | dat$pop, ylab = "Realized woodchat shrike breeding success ",
       xlab = "Spring precipitation index",
       main = "Realized breeding success (random-intercepts model)",
      pch = 16, cex = 1.2, col = rgb(0, 0, 0, 0.4))

# Revert to random-effects one-way binomial ANOVA model: random intercepts, but zero slopes
str(dat <- simDat17(nPops = 16, sigma.alpha = 1, mu.beta = 0, sigma.beta = 0))
xyplot(dat$C/dat$N ~ dat$precip | dat$pop, ylab = "Realized woodchat shrike breeding success ",
       xlab = "Spring precipitation index",
       main = "Realized breeding success (random-effects,
       one-way binomial ANOVA model)",
       pch = 16, cex = 1.2, col = rgb(0, 0, 0, 0.4))

# Revert to simple log-linear binomial (i.e., logistic) regression
#   (= no effects of pop on either intercepts or slopes)
str(dat <- simDat17(nPops = 16, sigma.alpha = 0, sigma.beta = 0))
xyplot(dat$C/dat$N ~ dat$precip | dat$pop, ylab = "Realized woodchat shrike breeding success ",
       xlab = "Spring precipitation index",
       main = "Realized breeding success\n(simple logistic regression model)",
       pch = 16, cex = 1.2, col = rgb(0, 0, 0, 0.4))

# Revert to binomial "model-of-the-mean": no effects of either population or precipitation
str(dat <- simDat17(nPops = 16, sigma.alpha = 0, mu.beta = 0, sigma.beta = 0))
xyplot(dat$C/dat$N ~ dat$precip | dat$pop, ylab = "Realized woodchat shrike breeding success ",
```

```
     xlab = "Spring precipitation index",
     main = "Realized breeding success (binomial 'model-of-the-mean')",
    pch = 16, cex = 1.2, col = rgb(0, 0, 0, 0.4))
```

---

simDat18 *Simulate data for Chapter 18: model selection*

Description

Simulate counts of rattlesnakes in Virginia

Usage

```
simDat18(
  nSites = 100,
  beta1.vec = c(1, 0.2, 0.5, 1, -1),
  ncov2 = 50,
  beta2.vec = rnorm(50, 0, 0),
  show.plot = TRUE
)
```

Arguments

`nSites`	Sample size (number of snakes)
`beta1.vec`	Values of log-linear intercept and coefs of rock, oak, and chip (linear and squared), in this order
`ncov2`	Number of 'other' covariates
`beta2.vec`	Values of coefs of the 'other' covariates (all continuous). All at zero by default
`show.plot`	Switch to turn on or off plotting. Set to 'FALSE' when running sims

Value

A list of simulated data and parameters.

`nSites`	Sample size
`rock`	Rock covariate vector
`oak`	Oak covariate vector
`chip1`	Chip covariate vector
`chip2`	Chip^2 covariate vector
`Xrest`	Array of "other" covariate values
`beta1.vec`	Parameter values for intercept, rock, oak, chip, chip^2
`ncov2`	Number of "other" covariates
`beta2.vec`	Vector of coefficient values for "other" covariates
`C`	Simulated rattlesnake counts

Author(s)

Marc Kéry

Examples

```
str(dat <- simDat18())  # With default arguments

#### First variant of data simulation: beta1.vec is identical, beta2.vec is not
# Variant B: execute when you want to play with a small data set
set.seed(18)
trainDat <- simDat18(nSites = 50, beta1.vec = c(1, 0.2, 0.5, 1, -1), ncov2 = 10,
                     beta2.vec = rnorm(10, 0, 0.1), show.plot = TRUE)
testDat <- simDat18(nSites = 50, beta1.vec = c(1, 0.2, 0.5, 1, -1), ncov2 = 10,
                    beta2.vec = rnorm(10, 0, 0.1), show.plot = TRUE)
# Note how relatively different the two realizations of the SAME process are

#### Second variant of data simulation: both beta1.vec and beta2.vec are identical

# Variant B: execute when you want to play with a small data set
set.seed(18)
beta2.vec <- rnorm(10, 0, 0.1)
trainDat <- simDat18(nSites = 50, beta1.vec = c(2, 0.2, 0.5, 1, -1), ncov2 = 10,
                     beta2.vec = beta2.vec, show.plot = TRUE)
testDat <- simDat18(nSites = 50, beta1.vec = c(2, 0.2, 0.5, 1, -1), ncov2 = 10,
                    beta2.vec = beta2.vec, show.plot = TRUE)
# Note how relatively different the two realizations of the SAME process are
```

`simDat19` *Simulate data for Chapter 19: Occupancy model*

Description

Simulate detection/nondetection data of Chiltern gentians

Usage

```
simDat19(
  nSites = 150,
  nVisits = 3,
  alpha.occ = 0,
  beta.occ = 2,
  alpha.p = 0,
  beta.p = -3
)
```

Arguments

`nSites`	Number of sites
`nVisits`	Number of replicate visits per site
`alpha.occ`	Occupancy intercept
`beta.occ`	Occupancy slope
`alpha.p`	Detection probability intercept
`beta.p`	Detection probability slope

Value

A list of simulated data and parameters.

`nSites`	Number of sites
`nVisits`	Number replicate visits per site
`alpha.occ`	Occupancy intercept
`beta.occ`	Occupancy slope
`alpha.p`	Detection probability intercept
`beta.p`	Detection probability slope
`humidity`	Humidity covariate
`occ.prob`	Probability of occupancy at each site
`z`	True occupancy state at each site
`true_Nz`	True number of occupied sites
`lp`	Linear predictor for detection
`p`	Probability of detection at each site
`y`	Simulated detection/non-detection data
`obs_Nz`	Observed number of occupied sites

Author(s)

Marc Kéry

Examples

```
str(dat <- simDat19())             # Implicit default arguments
str(dat <- simDat19(nSites = 150, nVisits = 3, alpha.occ = 0, beta.occ = 2,
  alpha.p = 0, beta.p = -3))       # Explicit default arguments
str(dat <- simDat19(nSites = 500)) # More sites
str(dat <- simDat19(nVisits = 1))  # Single-visit data
str(dat <- simDat19(nVisits = 20)) # 20 visits, will yield cumulative detection prob of about 1
str(dat <- simDat19(alpha.occ = 2))# Much higher occupancy
str(dat <- simDat19(beta.occ = 0)) # No effect of humidity on occupancy
str(dat <- simDat19(beta.p = 3))   # Positive effect of humidity on detection
str(dat <- simDat19(beta.p = 0))   # No effect of humidity on detection
```

`simDat19B` *Simulate data for Chapter 19B: Binomial N-mixture Model*

Description

Function simulates replicated count data as used in the bonus Chapter 19B in the ASM book. Abundance, detection and count data simulation is undertaken for the approximate elevation of the actual sample of survey sites in the Swiss breeding bird survey "Monitoring Häufige Brutvögel" (MHB). Note there is no nSites argument, since this is given in the MHB survey at 267.

Usage

```
simDat19B(
  nVisits = 3,
  alpha.lam = -3,
  beta1.lam = 8.5,
  beta2.lam = -3.5,
  alpha.p = 2,
  beta.p = -2,
  show.plot = TRUE
)
```

Arguments

`nVisits`	Number of occasions, or visits per site
`alpha.lam`	Intercept of the regression of log expected abundance on scaled elevation
`beta1.lam`	Linear effect of scaled elevation on log expected abundance
`beta2.lam`	Quadratic effect of scaled elevation on log expected abundance
`alpha.p`	Intercept of the regression of logit detection on scaled elevation
`beta.p`	Linear effect of scaled elevation on logit detection
`show.plot`	Show plot of simulated output?

Value

A list of simulated data and parameters.

`nSites`	Number of sites
`nVisits`	Number of visits to each site
`alpha.lam`	Abundance intercept
`beta1.lam`	Linear effect of elevation on abundance
`beta2.lam`	Quadratic effect of elevation on abundance
`alpha.p`	Detection intercept
`beta.p`	Linear effect of elevation on detection

`mhbElev`	Elevation covariate values
`mhbElevScaled`	Scaled elevation covariate values
`lambda`	Expected abundance for each site
`N`	True abundance at each site
`p`	Detection probability at each site
`C`	Observed repeated counts at each site
`nocc.true`	True number of occupied sites
`nocc.app`	Apparent number of occupied sites
`psi`	True proportion of occupied sites
`psi.app`	Apparent proportion of occupied sites
`opt.elev.true`	Optimal elevation value
`totalN.true`	True total population size
`totalN.app`	Apparent total population size

Author(s)

Marc Kéry

Examples

```
# With implicit default function argument values
str(simDat19B())

# With explicit function argument values
str(simDat19B(nVisits = 3, alpha.lam = -3, beta1.lam = 8.5, beta2.lam = -3.5,
    alpha.p = 2, beta.p = -2, show.plot = TRUE))

# No plots
str(simDat19B(show.plot = FALSE))

# More visits
str(simDat19B(nVisits = 10))

# A single visit at each site
str(simDat19B(nVisits = 1))

# Greater abundance
str(simDat19B(alpha.lam = 0))

# Much rarer abundance
str(simDat19B(alpha.lam = -5))

# No quadratic effect of elevation on abundance (only a linear one)
str(simDat19B(beta2.lam = 0))

# No effect of elevation at all on abundance
str(simDat19B(beta1.lam = 0, beta2.lam = 0))
```

```
# Higher detection probability (intercept at 0.9)
str(simDat19B(alpha.p = qlogis(0.9), beta.p = -2))

# Higher detection probability (intercept at 0.9) and no effect of elevation
str(simDat19B(alpha.p = qlogis(0.9), beta.p = 0))

# Perfect detection (p = 1)
str(simDat19B(alpha.p = 1000))

# Positive effect of elevation on detection probability (and lower intercept)
str(simDat19B(alpha.p = -2, beta.p = 2))
```

`simDat20` *Simulate data for Chapter 20: Integrated model*

Description

Simulate three count datasets under different data collection conditions

Usage

```
simDat20(
  nsites1 = 500,
  nsites2 = 1000,
  nsites3 = 2000,
  mean.lam = 2,
  beta = -2
)
```

Arguments

`nsites1`	Number of sites in regular count dataset
`nsites2`	Number of sites in zero-truncated count dataset
`nsites3`	Number of sites in detection/non-detection dataset
`mean.lam`	Mean site abundance
`beta`	Slope for elevation covariate

Value

A list of simulated data and parameters.

`nsites1`	Number of sites in regular count dataset
`nsites2`	Number of sites in zero-truncated count dataset
`nsites3`	Number of sites in detection/non-detection dataset
`mean.lam`	Mean site abundance

`beta`	Slope for elevation covariate
`C1`	Simulated regular counts from dataset 1
`C2`	Simulated regular counts from dataset 2
`C3`	Simulated regular counts from dataset 3
`ztC2`	Simulated zero-truncated counts from dataset 2
`y`	Simulated detection/non-detection data from dataset 3

Author(s)

Marc Kéry

Examples

```
str(dat <- simDat20())            # Implicit default arguments

# Revert to an 'integrated Poisson/binomial model-of-the-mean': no effect of elevation on abundance
str(dat <- simDat20(nsites1 = 500, nsites2 = 1000, nsites3 = 2000, mean.lam = 2, beta = 0))
```

`simDat4` *Simulate data for Chapter 4: Model of the mean*

Description

Simulate body mass measurements for n peregrine falcons from a normal distribution with population mean = 'mean' and population sd = 'sd'

Usage

```
simDat4(n = 10, mean = 600, sd = 30)
```

Arguments

`n`	The sample size
`mean`	Population mean
`sd`	Population standard deviation

Value

A list of simulated data and parameters.

`n`	Sample size
`mean`	Population mean
`sd`	Population SD
`y`	Simulated peregrine mass measurements

Author(s)

Marc Kéry

Examples

```
str(dat <- simDat4())          # Implicit default arguments
str(dat <- simDat4(n = 10^6))  # More than the world population of peregrines
str(dat <- simDat4(n = 10, mean = 900, sd = 40))  # Simulate 10 female peregrines
```

`simDat5` *Simulate data for Chapter 5: Simple linear regression*

Description

Simulate percent occupancy population trajectory of Swiss Wallcreepers from a normal distribution. Note that other choices of arguments may lead to values for x and y that no longer make sense in the light of the story in Chapter 5 (i.e., where y is a percentage), but will still be OK for the statistical model introduced in that chapter.

Usage

```
simDat5(n = 16, a = 40, b = -0.5, sigma2 = 25)
```

Arguments

`n`	The sample size
`a`	Value for the intercept
`b`	Value for the slope
`sigma2`	Value for the residual variance

Value

A list of simulated data and parameters.

`n`	Sample size
`a`	Intercept
`b`	Slope
`sd`	Residual SD
`y`	Simulated wallcreeper occupancy probabilities

Author(s)

Marc Kéry

Examples

```
str(dat <- simDat5())        # Implicit default arguments
str(dat <- simDat5(b = 0))   # Stable population (this is a de-facto "model-of-the-mean")
str(dat <- simDat5(b = 0.5)) # Expected increase
```

`simDat62` *Simulate data for Chapter 6.2: Two groups with equal variance*

Description

Simulate wingspan measurements in female and male peregrines with equal variance.

Usage

```
simDat62(n1 = 60, n2 = 40, mu1 = 105, mu2 = 77.5, sigma = 2.75)
```

Arguments

`n1`	The sample size of females
`n2`	The sample size of males
`mu1`	The population mean males
`mu2`	The population mean females
`sigma`	The standard deviation for both groups

Value

A list of simulated data and parameters.

`n1`	Female sample size
`n2`	Male sample size
`mu1`	Female mean
`mu2`	Male mean
`beta`	Difference in wingspan mean between sexes
`sigma`	Standard deviation for both groups
`x`	Indicator variable for sex, 1 = male
`y`	Simulated wingspan data

Author(s)

Marc Kéry

Examples

```
str(dat <- simDat62())      # Implicit default arguments
str(dat <- simDat62(n1 = 1000, n2 = 10000)) # Much larger sample sizes

# Revert to "model-of-the-mean" (with larger sample size)
str(dat <- simDat62(n1 = 10000, n2 = 10000, mu1 = 105, mu2 = 105))
```

`simDat63` *Simulate data for Chapter 6.3: Two groups with unequal variance*

Description

Simulate wingspan measurements in female and male peregrines with unequal variance.

Usage

```
simDat63(n1 = 60, n2 = 40, mu1 = 105, mu2 = 77.5, sigma1 = 3, sigma2 = 2.5)
```

Arguments

`n1`	The sample size of females
`n2`	The sample size of males
`mu1`	The population mean males
`mu2`	The population mean females
`sigma1`	The standard deviation for females
`sigma2`	The standard deviation for males

Value

A list of simulated data and parameters.

`n1`	Female sample size
`n2`	Male sample size
`mu1`	Female mean
`mu2`	Male mean
`beta`	Difference in wingspan mean between sexes
`sigma1`	Standard deviation for females
`sigma2`	Standard deviation for males
`x`	Indicator variable for sex, 1 = male
`y`	Simulated wingspan data

Author(s)

Marc Kéry

Examples

```
str(dat <- simDat63())            # Implicit default arguments
str(dat <- simDat63(sigma1 = 5, sigma2 = 1)) # Very unequal variances

# Much larger sample sizes and larger difference in residual variation
str(dat <- simDat63(n1 = 10000, n2 = 10000, sigma1 = 5, sigma2 = 2))

# Revert to model with homoscedasticity
str(dat <- simDat63(n1 = 10000, n2 = 10000, sigma1 = 5, sigma2 = 5))

# Revert to "model-of-the-mean" (with larger sample size)
str(dat <- simDat63(n1 = 10000, n2 = 10000, mu1 = 105, mu2 = 105, sigma1 = 5, sigma2 = 5))
```

`simDat72` *Simulate data for Chapter 7.2: ANOVA with fixed effects of population*

Description

Simulate snout-vent length measurements of nSample smooth snakes in each of nPops populations Data are simulated under the assumptions of a model with fixed effects of populations

Usage

```
simDat72(nPops = 5, nSample = 10, pop.means = c(50, 40, 45, 55, 60), sigma = 5)
```

Arguments

`nPops`	Number of populations
`nSample`	Samples from each population
`pop.means`	Vector of mean length for each population
`sigma`	Value for the residual standard deviation

Value

A list of simulated data and parameters.

`nPops`	Number of populations
`nSample`	Number of samples per population
`pop.means`	Population means
`sigma`	Residual SD
`pop`	Indicator for population number
`eps`	Simulated residuals
`y`	Simulated lengths

Author(s)

Marc Kéry

Examples

```
str(dat <- simDat72())      # Implicit default arguments

# More pops, fewer snakes in each
str(dat <- simDat72(nPops = 10, nSample = 5, pop.means = runif(10,20,60)))

# Revert to "model-of-the-mean" (larger sample size to minimize sampling variability)
str(dat <- simDat72(nSample = 1000, pop.means = rep(50, 5), sigma = 5))
```

`simDat73` *Simulate data for Chapter 7.3: ANOVA with random effects of population*

Description

Simulate snout-vent length measurements of nSample smooth snakes in each of nPops populations. Data are simulated under the assumptions of a model with random effects of populations

Usage

```
simDat73(nPops = 10, nSample = 12, pop.grand.mean = 50, pop.sd = 3, sigma = 5)
```

Arguments

`nPops`	Number of populations
`nSample`	Samples from each population
`pop.grand.mean`	Mean of population means (hyperparameter)
`pop.sd`	Standard deviation of population means (hyperparameter)
`sigma`	Value for the residual standard deviation

Value

A list of simulated data and parameters.

`nPops`	Number of populations
`nSample`	Number of samples per population
`pop.grand.mean`	Mean of population means
`pop.sd`	SD of population means
`sigma`	Residual SD
`pop`	Indicator for population number
`pop.means`	Simulated population means
`eps`	Simulated residuals
`y`	Simulated lengths

Author(s)

Marc Kéry

Examples

```
str(dat <- simDat73())      # Implicit default arguments
# More pops, more snakes in each, more among-population variability
str(dat <- simDat73(nPops = 20, nSample = 30, pop.sd = 8))

# Revert to "model-of-the-mean" (larger sample size to minimize sampling variability)
str(dat <- simDat73(nSample = 1000, pop.sd = 0, sigma = 5))
```

`simDat8` *Simulate data for Chapter 8: Two-way ANOVA*

Description

Simulate wing length measurements of mourning cloak butterflies with two factors (habitat and population) including their interaction if so wished (simulation under a fixed-effects model)

Usage

```
simDat8(
  nPops = 5,
  nHab = 3,
  nSample = 12,
  baseline = 40,
  pop.eff = c(-10, -5, 5, 10),
  hab.eff = c(5, 10),
  interaction.eff = c(-2, 3, 0, 4, 4, 0, 3, -2),
  sigma = 3
)
```

Arguments

`nPops`	Number of populations
`nHab`	Number of habitats
`nSample`	Samples from each population-habitat combination
`baseline`	Grand mean length
`pop.eff`	Population effects, should be nPops - 1 values
`hab.eff`	Habitat effects, should be nHab - 1 values
`interaction.eff`	Interaction effects, should be (nPops-1)*(nHab-1) values
`sigma`	Value for the residual standard deviation

Value

A list of simulated data and parameters.

`nPops`	Number of populations
`nSample`	Number of samples per population
`baseline`	Grand mean length
`pop.eff`	Population effects
`hab.eff`	Habitat effects
`interaction.eff`	Interaction effects
`sigma`	Residual SD
`all.eff`	All effects
`pop`	Indicator for population number
`hab`	Indicator for habitat number
`eps`	Simulated residuals
`wing`	Simulated wing lengths

Author(s)

Marc Kéry

Examples

```
str(dat <- simDat8())      # Implicit default arguments (for the model with interactions)
# Model with main effects only (and very large sample size; to minimize sampling error
# and clarify structure of main effects in plot)
str(dat <- simDat8(nSample = 1000, interaction.eff = c(0,0,0,0, 0,0,0,0)))
str(dat <- simDat8(nSample = 10000, interaction.eff = rep(0, 8))) # same, even larger sample size

# Revert to one-way ANOVA model with only effects of pop (with much larger sample size)
str(dat <- simDat8(nSample = 10000, pop.eff = c(-10, -5, 5, 10),
    hab.eff = c(0, 0), interaction.eff = rep(0, 8)))  # note no effect of habitat

# Revert to one-way ANOVA model with only effects of hab
str(dat <- simDat8(nSample = 10000, pop.eff = c(0, 0, 0, 0),
    hab.eff = c(5, 10), interaction.eff = rep(0, 8)))  # note no effect of pop

# Revert to "model-of-the-mean"
str(dat <- simDat8(nSample = 10000, pop.eff = c(0, 0, 0, 0),
    hab.eff = c(0, 0), interaction.eff = rep(0, 8)))  # note no effect of pop nor of h
```

---

simDat9 *Simulate data for Chapter 9: ANCOVA or general linear model*

Description

Simulate mass ~ length regressions in 3 populations of asp vipers

Usage

```
simDat9(
  nPops = 3,
  nSample = 10,
  beta.vec = c(80, -30, -20, 6, -3, -4),
  sigma = 10
)
```

Arguments

`nPops`	Number of populations
`nSample`	Samples from each population
`beta.vec`	Vector of regression parameter values
`sigma`	Value for the residual standard deviation

Value

A list of simulated data and parameters.

`nPops`	Number of populations
`nSample`	Number of samples per population
`beta.vec`	Regression parameter values
`sigma`	Residual SD
`x`	Indicator for population number
`pop`	Population name (factor)
`lengthC`	Centered body length for each viper
`mass`	Simulated body mass for each viper

Author(s)

Marc Kéry

Examples

```
# Implicit default arguments (with interaction of length and pop)
str(dat <- simDat9())

# Revert to main-effects model with parallel lines
str(dat <- simDat9(beta.vec = c(80, -30, -20, 6, 0, 0)))

# Revert to main-effects model with parallel lines
# (larger sample size to better show patterns)
str(dat <- simDat9(nSample = 100, beta.vec = c(80, -30, -20, 6, 0, 0)))

# Revert to simple linear regression: no effect of population
# (larger sample size to better show patterns)
str(dat <- simDat9(nSample = 100, beta.vec = c(80, 0, 0, 6, 0, 0)))

# Revert to one-way ANOVA model: no effect of body length
# (larger sample size to better show patterns)
str(dat <- simDat9(nSample = 100, beta.vec = c(80, -30, -20, 0, 0, 0)))

# Revert to "model-of-the-mean": no effects of either body length or population)
str(dat <- simDat9(nSample = 100, beta.vec = c(80, 0, 0, 0, 0, 0)))
```

`tmbSummary` *Summarize Output from TMB*

Description

Summarize output from TMB by point estimate (MLE), standard error (SE), and 95% Wald-type confidence intervals (CIs).

Usage

```
tmbSummary(tmbObject, dig = NULL)

tmb_summary(tmbObject, dig = NULL)
```

Arguments

`tmbObject`	A TMB object created by `MakeADFun` that has been optimized (e.g. with `optim`)
`dig`	Number of decimal places to use in output

Value

A matrix of parameter estimates, standard errors, and 95% Wald-type confidence intervals.

Author(s)

Ken Kellner

Index