Package ‘moult’

May 9, 2026

Version 2.3.1

Date 2022-08-29

Title Models for Analysing Moult in Birds

Author Birgit Erni <`birgit.erni@uct.ac.za`>. Based on models developed by Underhill and Zucchini (1988, 1990).

Maintainer Birgit Erni <`birgit.erni@uct.ac.za`>

Imports Formula, Matrix

Description Functions to estimate start and duration of moult from moult data, based on models developed in Underhill and Zucchini (1988, 1990).

License GPL-2

LazyLoad yes

LazyData yes

Suggests knitr, rmarkdown

VignetteBuilder knitr

NeedsCompilation no

Repository CRAN

Date/Publication 2022-08-30 09:00:02 UTC

Contents

`moult-package` *Methods for Estimating Start and Duration of Moult in Birds*

Description

Functions for estimating the duration and mean start date for avian moult data, based on maximum likelihood.

Details

Package:	moult
Type:	Package
Version:	2.0.0
Date:	2016-05-03
License:	GPL-2
LazyLoad:	yes

Author(s)

Birgit Erni

References

Erni, B., Bonnevie, B. T., Oschadleus, H.-D., Altwegg, R. and Underhill, L. G. (2013) moult: An R package to analyze moult in birds. Journal of Statistical Software, **52**(8), 1–23. doi:10.18637/jss.v052.i08

Underhill, L. G. and Zucchini, W. (1988) A model for avian primary moult. Ibis **130**, 358–372.

Underhill, L. G. and Zucchini, W. and Summers, R. W. (1990) A model for avian primary moult-data types based on migration strategies and an example using the Redshank *Tringa totanus*. Ibis **132**, 118–123.

`confint.moult` *Conficence Intervals for Moult Parameters*

Description

Calculate normal theory and bootstrap percentile intervals for moult parameters. Also returns bootstrap covariance matrix and standard error estimates for parameters.

Usage

```
## S3 method for class 'moult'
confint(object, parm, level = 0.95, ..., B = 1000, add.plot = TRUE)
```

Arguments

`object`	model output returned by call to moult.
`parm`	currently ignored (should be a vector of numbers or names, specifying for which parameters to calculate confidence intervals).
`level`	confidence level.
`...`	additional arguments for plot.
`B`	integer: number of bootstrap samples to take.
`add.plot`	logical: should scatterplot of bootstrapped parameters be added?

Details

End date is calculated as *mean start date + duration*. Half-date is calculated as *mean start date + 0.5 * duration*.

Value

Returns a list with elements:

`bootstrap.distribution`	$B \times p$ matrix of bootstrapped parameters.
`bootstrap.percentile.ci`	Table of bootstrap percentile intervals for all parameters. In addition, confidence intervals for half-date and end date (see details) are given.
`bootstrap.vcov`	Bootstrap estimates for variances and covariances between all parameters.
`bootstrap.SE`	Bootstrap estimates of parameter standard errors.

Author(s)

Birgit Erni `<birgit.erni@uct.ac.za>`

Examples

```
data(sanderlings)
m2 <- moult(MIndex ~ Day, data = sanderlings)
## Not run: confint(m2, B = 100) # increase B for better precision
```

`date2days` *Convert date to day count*

Description

Convert date mm/dd/yyyy to days since the 1st of startmonth, starting with days = 1 for the 1st day of startmonth.

Usage

```
date2days(date, dateformat, startmonth)
```

Arguments

`date`	date character string with format as specified in dateformat
`dateformat`	string specifying format of date, e.g. "mm/dd/yyyy", "yyyy-dd-mm", "dd–mm" etc.
`startmonth`	integer between 1 and 12 indicating month from which to start counting.

Value

Returns an integer = number of days since 1st of startmonth, starting with 1 = 1st of startmonth.

Author(s)

Bo T. Bonnevie

Examples

```
date2days("01/01/2010", "dd/mm/yyyy", 1)
date2days("01-01-2010", "dd-mm-yyyy", 2)
date2days("2008/06/01", "yyyy/mm/dd", 8)    # year has no effect
```

`dfbeta.moult` *Influence Statistics for Moult Parameters*

Description

Calculates dfbeta (change in coefficients) and dfbetas (scaled by standard error) for moult parameters.

Usage

```
## S3 method for class 'moult'
dfbeta(model, ...)
```

Arguments

`model`	a model object returned by `moult`.
`...`	further arguments.

Details

Both dfbeta (absolute change in coefficients) and dfbetas (change in coefficients scaled by standard error of coefficient) are returned.

$$dfbetas_i = \frac{\hat{b} - \hat{b}_i}{SE(\hat{b}_i}),$$

where the $\hat{b}_i$ estimate is obtained with observation i removed.

In the optional plot of dfbetas, cutoff lines at $\pm 2/\sqrt{n}$ are added. These are the limits used in linear regression models, but there is no reason that the same limits are valid for moult models. Therefore these cutoff lines can only be used as very rough guidelines.

Value

`dfbeta`	$n \times p$ matrix with absolute change in coefficients when observation i is deleted.
`dfbetas`	$n \times p$ matrix with scaled change in coefficients when observation i is deleted.

Author(s)

Birgit Erni <`birgit.erni@uct.ac.za`>

References

Fox, J.D. (2020). *Regression Diagnostics: an Introduction.* 2nd edition. SAGE Publications.

See Also

`dfbeta.lm`

Examples

```
data(sanderlings)
m2 <- moult(MIndex ~ Day, data = sanderlings)
## Not run: dfbeta(m2)
```

`moult` *Models to analyse data of moult in birds*

Description

Estimate duration and mean start date of moult from moult score data by maximum likelihood. Covariates to model duration and start of moult are possible.

Usage

```
moult(formula, data, start = NULL, type = 2, method = "BFGS", fixed = NULL,
             fixed.val = NULL, prec = 0.02)

moult_alternative(formula, data, start = NULL, type = 2, method = "BFGS",
                  fixed = NULL, fixed.val = NULL, prec = 0.02)
```

Arguments

`formula`	symbolic description of the model, see details.
`start`	starting values for parameters.
`data`	an optional data frame containing the variables in the model. If not found in data, the variables are taken from the environment from which `moult` was called.
`type`	integer (one of 1,2,3,4,5) referring to type of moult data and consequently model to be fitted (see details).
`method`	optimisation algorithm, passed to `optim`.
`fixed`	logical vector specifying which parameter values to fix during optimization.
`fixed.val`	numeric vector with values for the fixed parameters.
`prec`	numeric, measurement precision of moult index (proportion of feather mass grown), default = 0.02.

Details

`formula` is specified in 5 parts:

`moult.index ~ days | x1 + x2 | y1 + y2 | z1 + z2`

where `moult.index` is a numerical vector with values between 0 and 1, `days` is a vector with corresponding day numbers on which moult indeces were observed. The next three parts contain explanatory variables for modelling duration, mean start date and standard deviation in start date, respectively. If no explanatory variables are wanted for duration, say, this can be specified by leaving a blank between the first and second vertical lines, or equivalently, inserting a 1 between the vertical lines (which means all durations will be assumed equal). Similarly for mean start date and standard deviation. The minimum formula must contain `moult.index ~ days`, which will assume the same duration, mean start date and standard deviation for all individuals.

`type` refers to the type of moult data available (see Underhill and Zucchini (1998) and Underhill, Zucchini and Summers (1990)).

type = 1 sample is representative of entire population (not yet moulting, in moult, and birds which have completed moult). For type 1 data, any value between 0 and 1 (> 0 and < 1) can be used as the moult index for birds in active moult. The value used does not matter, only the fact that they are in moult.

type = 2 (default) sample is representative of entire population (not yet moulting, in moult, and birds which have completed moult). Moult scores are required.

type = 3 sample is representative only of birds in moult. Individuals with moult scores 0 or 1 are ignored.

type = 4 sample is representative only of birds in moult and those that have completed moult. Individuals with moult scores 0 are ignored.

type = 5 sample is representative only of birds that have not started moult or that are in moult. Individuals with moult scores 1 are ignored.

To *fix parameters* `fixed` will be a logical vector, e.g. `fixed = c(F, F, T),fixed.val = 3.5` will fix the standard deviation in start date to exp(3.5) and only estimate the remaining two parameters.

`moult` uses the R function `optim` to minimise the negative log-likelihood.

Note: The standard deviation parameters are estimated on the log-scale. Thus the corresponding elements in the covariance matrix are also on the log-scale. Starting values for the standard deviation should also be supplied on the log-scale. Standard errors for the standard deviation parameters are on the scale of the standard deviation (in days), estimated using the delta method.

`moult_alternative` offers a different parameterization (still in testing!): the *duration of moult* parameter is as before, but instead of start of moult, the parameter estimated is the *halfway date*, i.e. the date at which 50% of individuals have completed 50% of moult. The standard deviation parameter now is for the standard deviation between indiviudals in reaching 50% of moult. Start and end of moult can be derived as halfway date - 0.5 x duration, + 0.5 x duration, respectively. This alternative parameterization attempts to reduce the problem of strong negative correlation between the previous parameters start of moult and duration, and gives a more robust estimate for timing of moult (Les Underhill, pers. comm., Jackson 2018).

Value

`coefficients`	parameter estimates split up into duration, mean start date and standard deviation of start date.
`loglik`	log-likelihood at parameter estimates.
`vcov`	variance covariance matrix for paramter estimates (duration, mean start date, SD(start date)).
`standard.errors`	vector of standard errors for parameter estimates, obtained from diagonal elements of vcov, see details (duration, mean start date, SD(start date)).
`type`	type of data assumed, see details.
`residuals`	vector of residuals: observed - fitted moult index.
`fitted.values`	a vector of fitted values (moult scores).
`n`	number of observations.
`df.residual`	residual degrees of freedom for fitted model.

`terms`	duration formula, mean formula, full formula.
`X`	model matrix with covariates.
`y`	observed response (moult index) values.
`Day`	observed observation days.
`mean.formula`	model formula for mean start date.
`duration.formula`	model formula for duration of moult.
`formula`	model formula for mean start and duration of moult.
`sd.formula`	grouping variable used to estimate standard deviations in mean start dates, different for each group.
`optim`	object returned by call to `optim`, which minimises negative log-likelihood.
`converged`	logical value indicating whether algorithm has converged or not.
`convergence.value`	value for convergence returned by optim, see `optim` for details.

Author(s)

Birgit Erni <`birgit.erni@uct.ac.za`>

References

Erni, B., Bonnevie, B. T., Oschadleus, H.-D., Altwegg, R. and Underhill, L. G. (2013) moult: An R package to analyze moult in birds. *Journal of Statistical Software*, **52**(8), 1–23. doi:10.18637/jss.v052.i08

Jackson, C. 2018. *The moult and migration strategies of Lesser Sand Plover, Greater Sand Plover and Terek Sandpiper.* PhD Thesis, University of Cape Town, South Africa.

Underhill, L. G. and Zucchini, W. (1988) A model for avian primary moult. *Ibis* **130**, 358–372.

Underhill, L. G. and Zucchini, W. and Summers, R. W. (1990) A model for avian primary moult-data types based on migration strategies and an example using the Redshank *Tringa totanus*. *Ibis* **132**, 118–123.

See Also

`predict.moult`, `ms2pfmg`

Examples

```
data(sanderlings)

m2 <- moult(MIndex ~ Day, data = sanderlings)
summary(m2)
```

---

ms2pfmg *Convert moult scores to proportion of feather mass grown*

Description

Convert moult scores obtained for single feathers into overall proportion of feather mass grown.

Usage

```
ms2pfmg(ms, fm, split = "")
```

Arguments

`ms`	vector of moult scores. Each moult score should be a character string of one individual's feather moult scores, each between 0 and 5, e.g. "555444000" if nine primaries are of interest. Half steps are allowed, in which case the separator between individual scores can be specified in `split`, e.g. "5 5 5 4.5 4 3.5 0.5 0 0", in which case `split = " "`.
`fm`	vector of relative feather mass, corresponding to each feather in ms.
`split`	character used to separate individual feather moult scores. Default is no space / separator between scores.

Details

`ms` will usually be a vector of 9 or 10 primary feather scores, but single feathers can be given, in which case `fm` is ignored. The method used here assumes that a moult score of 1 for any feather corresponds to 1/8th of the feather grown, 2 corresponds to 3/8th = 0.375, etc.. The proportion of feather mass grown is then a weighted sum over all feathers, with weights equal to the relative weight (compared to the total weight) of each feather (Underhill and Zucchini 1988).

Value

`ms2pfmg` returns a vector (same length as `ms`) of values between 0 and 1: proportion of total feather mass grown.

Author(s)

Bo T. Bonnevie, Birgit Erni

References

Underhill, L. G. and Zucchini, W. (1988) A model for avian primary moult. Ibis **130**, 358–372.

See Also

`predict.moult`, `moult`

Examples

```
## relative primary feather mass of the 10 primary feathers
## (as proportion of total feather mass) for Sanderlings
fm.sand <- c(0.0385, 0.0458, 0.0544, 0.0680, 0.0827, 0.1019,
             0.1199, 0.1417, 0.1604, 0.1867)

ms2pfmg(3, 0.2)             # single feather
ms2pfmg(3, 1)
ms2pfmg("5555500000", fm.sand)

## all feathers weighted equally
ms2pfmg("54321", c(1,1,1,1,1))

## with half scores
ms2pfmg("5 4.5 3 2.5 1", c(1,1,1,1,1), split = " ")

## moult scores for more than 1 bird
ms2pfmg(c("5,4.5,3,2.5,1", "5,3.5,3,2.5,2.5"), c(1,1,1,1,1), split = ",")
```

`predict.moult` *Predict method for moult models*

Description

Predict either the proportion of birds in a certain moult stage (as defined in intervals) on a specified day, the average moult score on a specified day, or start and/or duration of moult for given covariate patterns

Usage

```
  ## S3 method for class 'moult'
predict(object, newdata, predict.type = "prob", intervals = 0.1, ...)
```

Arguments

`object`	moult model objects
`newdata`	optional dataframe with explanatory variables for which to make predictions. The first column must contain the days (as used when fitting) for which to make predictions.
`predict.type`	specifies form of predictions, see details.
`intervals`	length of moult categories for which probabilities/proportions should be calculated. The default (= 0.1) will calculate the probability that a bird will fall in this moult category on the specified day for each of the following categories: 0, (0, 0.1), [0.1, 0.2), [0.2, 0.3), ..., [0.9, 1.0), 1.
`...`	other arguments passed to the predict method

Details

`predict.type` has the following options:

- **"response"** the average moult index (proportion of feather mass grown) for each of the days specified.
- **"prob"** default, the proportion of birds in each of the moult categories as defined by intervals is predicted.
- **"duration"** predicts the duration of moult for the covariate combinations defined in `newdata`, with standard errors. If `newdata` is not supplied, returns duration of baseline covariate set.
- **"start"** predicts the mean start date of moult for the covariate combinations defined in `newdata`, with standard errors. If `newdata` is not supplied, returns mean start date of baseline covariate set.
- **"both"** predicts both mean start date of moult and duration of moult for the covariate combinations defined in `newdata`, with standard errors. Also, covariance of duration and start date estimates is given.

Value

If `newdata` is missing, the expected moult scores at the observed days are returned.

If `predict.type = "response"`, the expected moult scores at the specified days are returned.

If `predict.type = "prob"` a matrix of predicted probabilities for being in each of the moult categories defined by intervals.

If `predict.type` equals "start" or "duration" or "both", the corresponding estimates (with standard errors) for each of the covariate patterns are returned.

Author(s)

`<birgit.erni@uct.ac.za>`

References

Erni, B., Bonnevie, B. T., Oschadleus, H.-D., Altwegg, R. and Underhill, L. G. (2013) moult: An R package to analyze moult in birds. Journal of Statistical Software, **52**(8), 1–23. `https://www.jstatsoft.org/v52/i08/`

Underhill, L. G. and Zucchini, W. (1988) A model for avian primary moult. *Ibis* **130**, 358–372.

Underhill, L. G. and Zucchini, W. and Summers, R. W. (1990) A model for avian primary moult-data types based on migration strategies and an example using the Redshank *Tringa totanus*. *Ibis* **132**, 118–123.

See Also

`moult`

Examples

```
data(weavers)

## convert moult scores to PFMG (proportion feather mass grown)
mscores <- substr(weavers$Moult, 1, 9)
feather.mass <- c(10.4, 10.8, 11.5, 12.8, 14.4, 15.6, 16.3, 15.7, 15.7)
weavers$pfmg <- ms2pfmg(mscores, feather.mass)

## day of year starting 1 August
weavers$day <- date2days(weavers$RDate, dateformat = "yyyy-mm-dd", startmonth = 8)
weavers$ssex <- ifelse(weavers$Sex == 1 | weavers$Sex == 3, "male",
  ifelse(weavers$Sex == 2 | weavers$Sex == 4, "female", NA))

mmf <- moult(pfmg ~ day | ssex | ssex, data = weavers, type = 3)
summary(mmf)

## predict duration and start of moult (then both) for males and females
ssex <- c("male", "female")
day <- 150

(p1 <- predict.moult(mmf, newdata = data.frame(day, ssex), predict.type = "duration"))
(p2 <- predict.moult(mmf, newdata = data.frame(day, ssex), predict.type = "start"))
(p3 <- predict.moult(mmf, newdata = data.frame(day, ssex), predict.type = "both"))
```

`sanderlings` *Sanderling Moult Data*

Description

This data set gives moult indices for 164 Sanderlings trapped on 11 days.

Usage

```
data(sanderlings)
```

Format

A data frame with 164 observations on the following 2 variables.

`Day` a numeric vector of day bird was measured, 1 = 1 July

`MIndex` a numeric vector of moult indices, 0 = bird has not started moult, 1 = bird has completed moult

Details

This data set gives moult indices for 164 Sanderlings trapped on 11 days in the southwestern Cape, South Africa, between October 1978 and April 1979. Day 1 = 1 July). Moult indices are a transformation of moult scores so that moult index increases linearly with time. See Underhill and Zucchini (1988) for details.

Source

Underhill and Zucchini (1998)

References

Underhill, L. G. and Zucchini, W. (1988) A model for avian primary moult. Ibis **130**, 358–372.

Examples

```
data(sanderlings)

## fit model of type 1 to data
m1 <- moult(MIndex ~ Day, data = sanderlings, type = 1)
summary(m1)

## model of type 2 (default)
m2 <- moult(MIndex ~ Day, data = sanderlings)
summary(m2)

## model of type 3
m3 <- moult(MIndex ~ Day, data = sanderlings, type = 3)
summary(m3)

## find intercept and slope of mean moult trajectory line
uza <- - coef(m2, "mean") / coef(m2, "duration")
uzb <- 1 / coef(m2, "duration")

## extract how many birds observed on each of the days
nn <- as.numeric(table(sanderlings$Day))
## extract days of observations
day <- unique(sanderlings$Day)

## probabilities of moult stages
## Table 6 in Underhill and Zucchini 1988
p1 <- predict(m2, newdata = data.frame(day))
p1$M * nn

## Table 7 in Underhill and Zucchini 1988
days2 <- seq(70, 310, by = 10)
p2 <- predict(m2, newdata = data.frame(days2))
p2$M * 100

p3 <- predict(m3, newdata = data.frame(day))
p3

## Comparison with regression models
MInd <- sanderlings$MIndex[sanderlings$MIndex > 0 &
                           sanderlings$MIndex < 1]
MTime <- sanderlings$Day[sanderlings$MIndex > 0 &
                         sanderlings$MIndex < 1]

lm1 <- lm(MTime ~ MInd)
```

```
lm1.int <- coef(lm1)[1]
lm1.slope <- coef(lm1)[2]

lm2 <- lm(MInd ~ MTime)

## regression of Index on Time
plot(MTime, MInd, pch = 19, cex=0.7)

## regression of Time on Index: gives better estimates
## for mean start day and duration of moult
abline(lm2, col = "blue", lwd = 2)
abline(-lm1.int / lm1.slope, 1 / lm1.slope, col = "orange", lwd = 2)
abline(uza, uzb, col = "red", lty = 2, lwd = 2)
```

`weavers` *Weaver Moult Data*

Description

Weaver moult data from the Western Cape, South Africa

Usage

```
data(weavers)
```

Format

A data frame with 7543 observations on the following 4 variables.

`RDate` a character vector with dates on which individuals were caught, format: yyyy-mm-dd.

`Sex` a numeric vector, 0 = unknown, 1 = male, 2 = female, 3 = possibly male, 4 = possibly female.

`Year` year in which individual was caught.

`Moult` a character vector with moult scores for individual primary feathers, either nine or ten, starting with innermost primary feather. 0: old feather, 5: new feather, 1 to 4, feathers at various stages of growth in between.

Source

Oschadleus, D. (2005). Patterns of primary moult in weavers. PhD Thesis. University of Cape Town.

SAFRING, URL: https://safring.birdmap.africa/

Examples

```
data(weavers)

mscores <- substr(weavers$Moult, 1, 9)

## convert moult scores to proportion of feather mass grown
feather.mass <- c(10.4, 10.8, 11.5, 12.8, 14.4, 15.6, 16.3, 15.7, 15.7)
weavers$pfmg <- ms2pfmg(mscores, feather.mass)

## days since 1 August
weavers$day <- date2days(weavers$RDate, dateformat = "yyyy-mm-dd", startmonth = 8)
ssex <- ifelse(weavers$Sex == 1 | weavers$Sex == 3, "male",
  ifelse(weavers$Sex == 2 | weavers$Sex == 4, "female", NA))
weavers$ssex <- as.factor(ssex)

## moult model with duration and mean start date depending on sex
mmf <- moult(pfmg ~ day | ssex | ssex, data = weavers, type = 3)
summary(mmf)

## predict duration and start of moult (then both) for males and females
ssex <- c("male", "female")
day <- 150
(p1 <- predict.moult(mmf, newdata = data.frame(day, ssex), predict.type = "duration"))
(p2 <- predict.moult(mmf, newdata = data.frame(day, ssex), predict.type = "start"))
(p3 <- predict.moult(mmf, newdata = data.frame(day, ssex), predict.type = "both"))
```

Index