# moult: An R Package to Analyse Moult in Birds

*Birgit Erni, Bo T. Bonnevie, Hans-Dieter Oschadleus, Res Altwegg, Les G. Underhill*

*20 December 2017*

The original version of this manuscript was published in the *Journal of Statistical Software* (Erni et al. 2013). This vignette is kept up to date with the package.

#### Abstract

Moult is the process by which birds replace their feathers. It is a costly process in terms of energy and reduced flight ability but necessary for the maintenance of the plumage and its functions. Because birds generally avoid to moult while engaged with other energy demanding activities such as breeding and migration, the analysis of moult data gives insight into how birds fit this life stage into the annual cycle, on time constraints in the annual cycle, and on the effects of environmental variables on the timing of moult. The analysis of moult data requires non-standard statistical techniques. More than 20 years ago Underhill and Zucchini developed a likelihood approach for estimating duration, mean start date and variation in start date of a population of moulting birds. However, use of these models has been limited, mainly due to the lack of user-friendly software. The **moult** package for R implements the Underhill-Zucchini models, allowing the user to specify moult models in a regression type formula. In addition the functions allow the moult parameters (duration, and mean and variation in start date) to depend on explanatory variables. We here describe the package, give a brief summary of the theory and illustrate the models on two datasets included in the package.

## Introduction

Many bird ringers collect data on the progression of moult. Moult data can provide insights into the energy allocation under time constraints involved in the annual cycle of different bird populations, and these data are relatively easy to collect compared to the timing of breeding and migration (Newton 2009). However, the analysis of moult data has been slowed initially by the lack of appropriate statistical methods to analyse these, and after Underhill and Zucchini (1988) developed their likelihood models (referred to as UZ models henceforth), by the lack of software that could run these models in a convenient user-friendly way.

Here we show how to fit the UZ moult models using the R package **moult**, available from the Comprehensive R Archive Network (CRAN) at http://cran.r-project.org/package=moult. We briefly summarise the theory underlying the moult models as developed by Underhill and Zucchini (1988) and extended in Underhill et al. (1990). We describe the implementation in R (R Core Team 2017) and illustrate the use of the models using some examples.

## Moult data

The standard accepted method of collecting moult data on primary feathers is to score each primary feather as 0 (old), 5 (new), or scores 1 to 4 for intermediate stages of feather development (Ginn and Melville 1983). These individual moult scores (collected for either nine, ten or, rarely, 11 primaries, depending on the species) are then summed to give an overall moult score between 0 and 45 (for 9 primaries) or between 0 and 50 (for 10 primaries).

The aim of analysing moult data is to understand the timing of moult, so that the parameters of interest are the start and duration of moult, and a measure of the within-population variation in start date.

Fitting simple linear regression models to moult data is not appropriate (Pimm 1976; Summers, Swann, and Nicoll 1983; Underhill 1985; Underhill and Zucchini 1988); they give biased estimates for the start and duration of moult. Underhill and Zucchini (1988) developed models that specifically address the issues
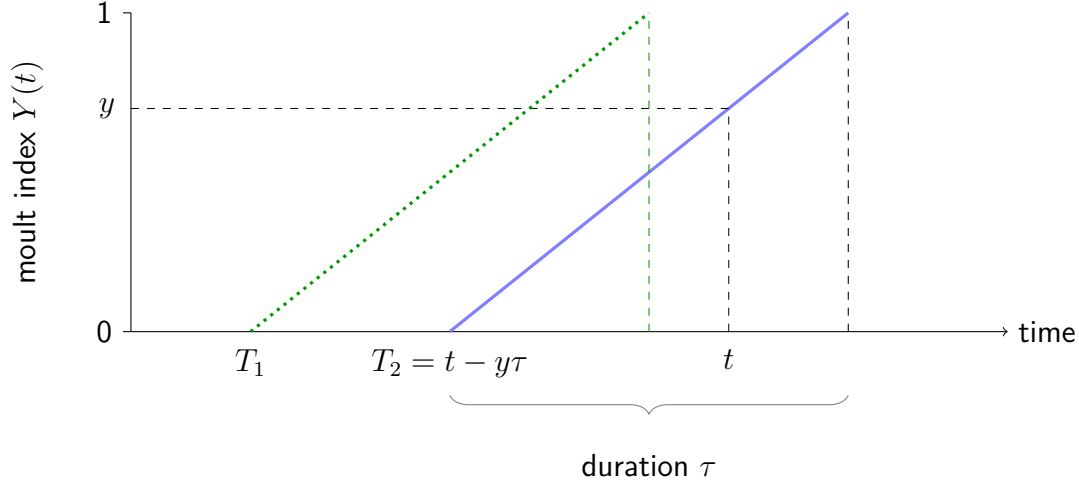
Figure 1: Illustration of UZ model assumptions and parameters. With a constant rate of change in moult index, the individual indicated by the solid line has completed a proportion $y$ of its moult at time $t$ and thus must have started moult at time $t - y\tau$, where $\tau$ is the duration of moult $= 1/$slope of line. The individual indicated by the dotted line has completed moult at time $t$, thus its starting time $T_1$ must have been prior to $t - \tau$.

of moult data using a likelihood approach. The main issue is that moult has three clearly defined stages (pre-moult, in moult, post-moult), so that moult scores cannot be assumed to be normally distributed.

## The Underhill and Zucchini (UZ) moult models

The UZ models use four key assumptions to construct a likelihood for the moult data. The assumptions are as follows. 1) The moult index increases linearly over time, i.e. a constant rate of change in moult index, for each individual (Figure 1). This moult index can be the original sum of moult scores, or some transformation of these to make the index linear. For example, for waders Summers et al. (1983) proposed using 'proportion of feather mass grown' assuming that feather mass is produced at a constant rate. It is convenient to scale this index to range from 0 to 1. 2) The duration of moult, and hence the rate of change in moult index, is the same for every individual. Modelling the duration of moult as a function of covariates allows us to relax this assumption. 3) Date of start of moult of individuals is normally distributed about a population mean start date. Assumption (3) could be changed, for example to Gamma, however this has not been implemented in package **moult** so far. 4) The individual observations are independent.

Figure 1 shows the hypothetical trajectories of two individuals. If both individuals were caught at time $t$, the green individual (dotted line) has completed moult and will have moult index $= 1$; the blue individual is in moult with observed moult index $y$. Of interest is the time at which these individuals have started, $T_1$ and $T_2$ say, respectively, and their duration of moult ($\tau$ for both). To model moult in the population, we assume that for individual $i$ starting date $T_i \sim N(\mu, \sigma^2)$, with $\mu$ and $\sigma$ the mean start date and standard deviation in start date, respectively, for the population.

Given the above model for start and duration of moult, it is then possible to calculate the probability for individual $i$ to have a moult index $y_i$ at time $t_i$:

$$Pr(y_i) = Pr(T_i = t_i - y_i\tau)$$

2

In other words, the probability that the individual has started moult at time $T_i = t_i - y_i\tau$. The probability of moult index $y_i$ at time $t_i$ is approximated by the normal density $f(t_i - y_i\tau|\mu, \sigma^2)$. In newer versions of the moult package, this probability is not approximated by the density anymore, but, more correctly, by the integral $\int_{y_i-e}^{y_i+e} t_i - y_i\tau dy_i$, where $e$ represents the precision of the moult indices, with default 0.02. The probabilities for all individuals are combined into a likelihood, and parameters $\mu$, $\tau$ and $\sigma^2$ are estimated by maximising this likelihood. For a more detailed description of the UZ models we refer the reader to Underhill and Zucchini (1988).

## Explanatory variables

As in other statistical models it is often of interest to see how parameters change as functions of covariates. For example, the duration of moult could depend on the distance a population needs to migrate, as the time spent on migration restricts the time left for other activities (Newton 2009). Another example would be different mean start dates for different geographical populations of a species.

Previously, software for the UZ models did not include the option of modelling parameters as functions of explanatory variables, or did so only in a limited way (Underhill, Serra, and Brandao 2008). The **moult** package allows the three parameters $\mu$, $\tau$ and $\sigma$ to be modelled as functions of explanatory variables. There is a restriction: $\sigma$ can be modelled as a function of only a single categorical variable, resulting in a different variance estimate for each group specified by the categorical variable; no continuous covariates are currently allowed for $\sigma$.

## Fitting moult models in R

The UZ moult models can be fitted with the function `moult()`. This function was designed to be similar in structure to the standard regression type functions in R (such as `lm()` and `glm()`). In particular, we obtained many ideas for coding from the `hurdle()` function in package **pscl** (Zeileis, Kleiber, and Jackman 2008). The call to moult has the following structure:

```
moult(formula, data, start = NULL, type = 2)
```

Only the `formula` argument is required, the others are optional, with default values if not specified. `formula` can have five parts, of which the first two (`moult.index` and `days`) must always be provided:

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

On the left-hand-side of the $\sim$ the vector of moult indices (between 0 and 1, one for each individual) is given, the first part after the $\sim$ requires a vector of corresponding days on which these individuals were observed (number of days since, e.g., 1 July). The remaining three parts are optional, but are used to model any of the three parameters as functions of covariates. Their order is important: the first part specifies the covariates for the duration of moult, the second part covariates for the mean start date of moult, and the third a covariate for the standard deviation in start date. These covariates can overlap, and quadratic terms, interactions, etc. can be added to either part, using the standard R notation for model formulae (Chambers and Hastie 1992).

This is best demonstrated by a few examples. Taking `colony` to be a hypothetical (categorical) covariate, assuming that observations came from individuals caught in two or more different colonies, we can fit the following models.

```
m1 <- moult(m.index ~ days)
m2 <- moult(m.index ~ days | colony)
m3 <- moult(m.index ~ days | 1 | colony)
m4 <- moult(m.index ~ days | colony | colony)
m5 <- moult(m.index ~ days | colony | colony | colony)
```

In `m1` duration, mean start date and standard deviation in start date are assumed the same for all colonies; in `m2` only duration differs between colonies, while start dates are assumed equal for all colonies; in `m3` only mean start date differs between colonies; in `m4` a different start date and a different duration are estimated for each colony; in `m5` all three parameters are assumed to differ between colonies. Similar covariates such as sex or year, could be used, or continuous covariates such as average temperature or total rainfall during the breeding season. The covariate for the standard deviation in start date of moult is restricted to be a single categorical variable.

Starting values for parameter estimation can be specified through the `start` argument. If not specified `moult()` will calculate its own starting values by using regression models of time on moult index.

The `type` argument allows the specification of different types of moult data as defined by Underhill and Zucchini (1988) and Underhill et al. (1990). The data type should depend on which stages of moult were representatively sampled. For type 1 and type 2 data it is required that pre-moult, in-moult and post-moult individuals are all equally likely to have been sampled, i.e. sampling probability is independent of moult stage; type 2 data requires the moult scores, type 1 only the categories pre-moult, in moult, post-moult. Type 3 data requires only that individuals in moult are a representative sample of the population. For type 4 data the sample must be representative of the part of the population in moult and post-moult, and for type 5 of the part of the population pre-moult and in moult (Underhill, Zucchini, and Summers 1990).

`moult()` constructs the likelihood and makes a call to R's `optim()` function to maximise the log-likelihood, using `optim`'s 'BFGS' algorithm.

# Examples

## Sanderlings

The `sanderlings` data (Underhill and Zucchini 1988) are included with the **moult** package (Figure 2). The moult scores in `MIndex` are proportions of feather mass grown. `Day` contains the number of days since July 1.

```
library("moult")
head(sanderlings)
```

```
R>   Day MIndex
R> 1 117  0.000
R> 2 117  0.000
R> 3 117  0.000
R> 4 117  0.000
R> 5 117  0.000
R> 6 117  0.011
```

```
pdf("sanderling1.pdf")
plot(MIndex ~ Day, data = sanderlings, pch = 19,
     xlab = "days since July 1", las = 1,
     ylab = "proportion of feather mass grown")
dev.off()
```

To demonstrate the `moult()` function, we will fit three models, one for each of the data types 1 to 3 (see Section 5). Even though the `sanderlings` data contains individuals not in moult, depending on `type`, the function will discard non-moulting individuals (type 3) or the actual moult scores, only keeping the information `pre-moult'`,in moult', or 'post-moult' (type 1).

```
m1 <- moult(MIndex ~ Day, data = sanderlings, type = 1)

m2 <- moult(MIndex ~ Day, data = sanderlings)
summary(m2)
```
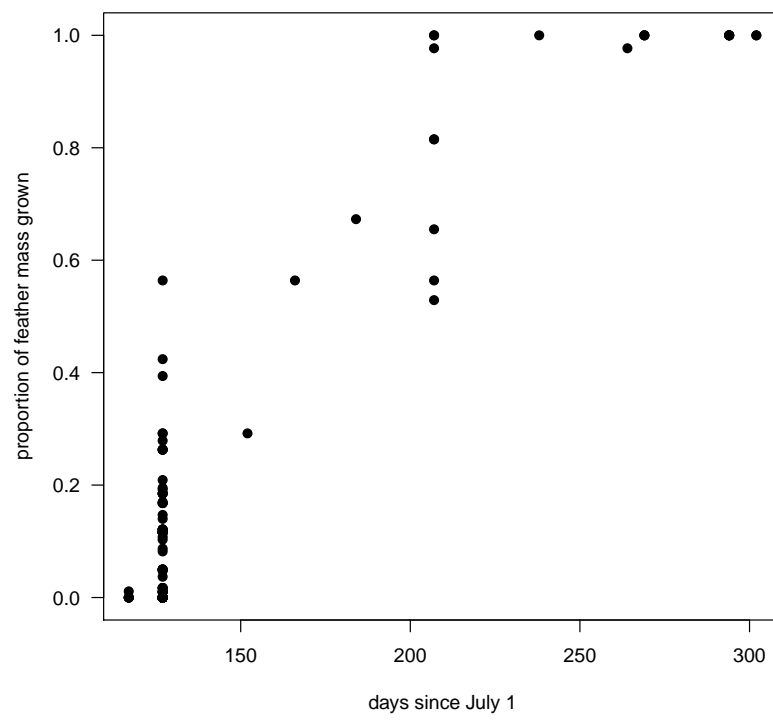
Figure 2: Sanderling moult scores versus time.

```
R>
R> Call:
R> moult(formula = MIndex ~ Day, data = sanderlings)
R>
R>
R> Duration coefficients:
R>            Estimate Std. Error
R> intercept.1    96.08       6.902
R>
R> Mean start date coefficients:
R>          Estimate Std. Error
R> intercept    131.4       2.096
R>
R> Coefficients for standard deviation in start date:
R>            Estimate Std. Error
R> (Intercept)    19.18       6.169
R>
R> Log-likelihood:  -255 on 3 Df
```

```r
m3 <- moult(MIndex ~ Day, data = sanderlings, type = 3)
summary(m3)
```

```
R>
R> Call:
R> moult(formula = MIndex ~ Day, data = sanderlings, type = 3)
R>
R>
R> Duration coefficients:
R>             Estimate Std. Error
R> intercept.1    99.44      18.53
R>
R> Mean start date coefficients:
R>          Estimate Std. Error
R> intercept    126.4       8.817
R>
R> Coefficients for standard deviation in start date:
R>            Estimate Std. Error
R> (Intercept)    17.46       6.187
R>
R> Log-likelihood: -151.6 on 3 Df
```

Estimates obtained for `m3` differ only slightly from those in Underhill and Zucchini (1988); the value obtained for the log-likelihood is different to that in Underhill and Zucchini (1988) because we don't approximate the probabilities of individual observations by densities, but use an integral to approximate the probability of observations. The log-likelihood values in the outputs from models `m1`, `m2` and `m3` are not comparable, because different subsets or transformations of the observations (moult indices) were used in these three models.

In model `m2` mean start date was estimated as day number 131.4 (1 = 1 July). We can convert this to a date as follows.

```r
format(as.Date(coef(m2, "mean"), origin = "2012-06-30"), "%d %b")
```

```
R> intercept
R>  "08 Nov"
```

6

To obtain the intercept and slope for the moult trajectory line for a given duration and start date (Figure 1) we write a function which converts duration and mean start date to intercept and slope:

```
durationmean2ab <- function(duration, mean)
  { ab <- c(- mean / duration, 1 / duration)
    names(ab) <- c("intercept", "slope")
    return(ab)
  }

uz1 <- durationmean2ab(coef(m1, "duration"), coef(m1, "mean"))
```

The last line calculates slope and intercept of the moult trajectory line for an individual starting moult on the mean start date estimated in model m1 (Figure 3).

**Fitted values and predictions**

predict() can be used on a moult object to predict the population average moult score on a specified day, a frequency table of number of birds at different stages of moult, or the probability of being in a specified stage of moult on a specified day. Moult stages can be categorised into intervals, e.g., pre-moult (moult index 0), moult index between 0 and 0.1, between 0.1 and 0.2, .... The interval length can be changed through the intervals argument in the predict function.

In the following code we predict, according to model m2, for each observation day the number of birds out of those observed that should be in moult stages $0, (0, 0.25), [0.25, 0.5), [0.5, 0.75), [0.75, 1), 1$.

```
day <- unique(sanderlings$Day)
p1 <- predict(m2, newdata = data.frame(day), intervals = 0.25)
nn <- as.numeric(table(sanderlings$Day))
p1$M * nn
```

| R> | | 0 | 0 - 0.25 | 0.25 - 0.5 | 0.5 - 0.75 | 0.75 - 1 | 1 |
|---|---|---|---|---|---|---|---|
| R> | 117 | 4.638 | 1.224 | 0.132 | 0.006 | 0.000 | 0.000 |
| R> | 127 | 79.650 | 45.900 | 8.910 | 0.405 | 0.000 | 0.000 |
| R> | 152 | 0.141 | 0.429 | 0.353 | 0.073 | 0.004 | 0.000 |
| R> | 166 | 0.036 | 0.255 | 0.468 | 0.217 | 0.025 | 0.001 |
| R> | 184 | 0.003 | 0.065 | 0.338 | 0.439 | 0.144 | 0.012 |
| R> | 207 | 0.000 | 0.032 | 0.576 | 2.808 | 3.448 | 1.144 |
| R> | 238 | 0.000 | 0.000 | 0.001 | 0.035 | 0.256 | 0.709 |
| R> | 264 | 0.000 | 0.000 | 0.000 | 0.001 | 0.028 | 0.972 |
| R> | 269 | 0.000 | 0.000 | 0.000 | 0.000 | 0.045 | 2.955 |
| R> | 294 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 5.000 |
| R> | 302 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 2.000 |

The argument newdata in predict() must contain one column for dates, and one for each explanatory variable used in the model. In this case predict will return estimated probabilities for any bird being in moult stage $i$ on the given day; rows sum to one. When multiplied by the number of birds observed on the particular day (nn here), the expected number of birds in each moult stage on that day is obtained (see Table 6 in Underhill and Zucchini 1988). If the argument newdata is not supplied, predict will return estimates of the expected moult score for each of the observed dates.

```
days2 <- seq(70, 310, by = 10)
p2 <- predict(m2, newdata = data.frame(days2))
p2$M * 100
```

The above code predicts the proportion of birds in each moult stage for days 70 to 310 (multiplied by 100 to obtain percentages) (see Table 7 in Underhill and Zucchini 1988). See Section 6.3 for obtaining predictions with covariates.

## Probit analysis

Rothery and Newton (2002) realised that one could obtain estimates for mean start date and standard deviation using a probit model (a generalized linear model with a binomial distribution and a probit link function), referred to as the RN method henceforth. The distributional assumptions (normality) for the start of moult are equivalent in the UZ and RN models. What differs in the RN method is that a second probit analysis is required for the finishing date, but no assumptions are made about the functional form of change in moult index. Thus in the RN method it is possible to obtain starting and finishing distributions with different standard deviations. Average moult duration is estimated as the difference between mean finishing and mean start dates. The RN method uses the same data twice, once to estimate the start and once to estimate the end of moult. Development of the RN model was probably in part motivated by the lack of user-friendly software for fitting UZ models since they are conceptually similar to the UZ type 1 model. Below we demonstrate how the RN probit models can be fitted using R for the `sanderlings` data.

To estimate mean start date in the probit model, the moult scores are transformed to a binary variable with categories `have not started moult' (0), and `have started moult' (1). Individuals with completed moult are scored as 1. Then a probit model is fitted to these scores.

```
sanderlings$started <- ifelse (sanderlings$MIndex > 0, 1, 0)
m.start <- glm(started ~ Day, family = binomial(link = "probit"),
               data = sanderlings)
```

```
R> Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```
summary(m.start)
```

```
R>
R> Call:
R> glm(formula = started ~ Day, family = binomial(link = "probit"),
R>     data = sanderlings)
R>
R> Deviance Residuals:
R>     Min      1Q   Median      3Q      Max
R> -1.0253  -1.0253  -0.5637   1.3376   1.9586
R>
R> Coefficients:
R>             Estimate Std. Error z value Pr(>|z|)
R> (Intercept) -10.63391    6.47880  -1.641    0.101
R> Day           0.08192    0.05110   1.603    0.109
R>
R> (Dispersion parameter for binomial family taken to be 1)
R>
R>     Null deviance: 227.13  on 163  degrees of freedom
R> Residual deviance: 187.99  on 162  degrees of freedom
R> AIC: 191.99
R>
R> Number of Fisher Scoring iterations: 10
```

This probit model models the probability of having started moult as a function of time (Day here). `glm` issues a warning message here as a result of 'Day' being able to predict the response (started or not) almost perfectly (see, Venables and Ripley 2002, e.g. pg. 197), but is not serious here. The slope and intercept obtained have to be transformed as follows to obtain mean and standard deviation estimates for the start of moult (Rothery and Newton 2002).

```
cfs <- as.numeric(coef(m.start))
sigma.hat1 <- 1 / cfs[2]
mu.hat1 <- - cfs[1] / cfs[2]
```

```
c(mu.hat1, sigma.hat1)
```

R> [1] 129.8156  12.2077

```
format(as.Date(mu.hat1, origin = "2012-06-30"), "%d %b")
```

R> [1] "06 Nov"

To estimate mean and standard deviation for date of completion of moult the above is repeated but now moult scores are transformed to 1 if moult is completed, and 0 otherwise.

```
sanderlings$finished <- ifelse (sanderlings$MIndex < 1, 0, 1)
m.end <- glm(finished ~ Day, family = binomial(link = "probit"),
             data = sanderlings)

cfs <- as.numeric(coef(m.end))
sigma.hat2 <- 1 / cfs[2]
mu.hat2 <- - cfs[1] / cfs[2]
duration <- mu.hat2 - mu.hat1
c(duration, mu.hat2, sigma.hat2)
```

R> [1] 100.57376 230.38933  29.73703

```
format(as.Date(mu.hat2, origin = "2012-06-30"), "%d %b")
```

R> [1] "15 Feb"

```
nr1 <- durationmean2ab(duration, mu.hat1)
```

The last line above calculates the intercept and slope of the average moult trajectory.

Mean start date and mean completion date can also be estimated with a single model. This, however, requires the assumption of equal standard deviations in starting and finishing dates, and an additional indicator variable is required ($x$ here), indicating whether we are considering starting date or finishing date.

```
scores <- c(sanderlings$started, sanderlings$finished)
x <- c(rep(0, times = length(sanderlings$started)),
       rep(1, times = length(sanderlings$finished)))
ddays <- rep(sanderlings$Day, times = 2)

m.both <- glm(scores ~ ddays + x, family = binomial(link = "probit"))
summary(m.both)
```

```
R>
R> Call:
R> glm(formula = scores ~ ddays + x, family = binomial(link = "probit"))
R>
R> Deviance Residuals:
R>     Min       1Q    Median       3Q       Max
R> -2.15506  -0.79254  -0.01047   0.00065   1.80419
R>
R> Coefficients:
R>             Estimate Std. Error z value Pr(>|z|)
R> (Intercept) -5.021635   1.269464  -3.956 7.63e-05
R> ddays        0.037670   0.009944   3.788 0.000152
R> x           -3.630548   1.035830  -3.505 0.000457
R>
R> (Dispersion parameter for binomial family taken to be 1)
```

```
R>
R>     Null deviance: 389.28  on 327  degrees of freedom
R> Residual deviance: 204.51  on 325  degrees of freedom
R> AIC: 210.51
R>
R> Number of Fisher Scoring iterations: 9
```

```r
cfs <- as.numeric(coef(m.both))
sigma.hat <- 1 / cfs[2]
mu.hat <- - cfs[1] / cfs[2]
mean.finish <- - (cfs[1] + cfs[3]) / cfs[2]
duration <- - cfs[3] / cfs[2]
c(duration, mu.hat, mean.finish, sigma.hat)
```

```
R> [1]  96.37801 133.30638 229.68439  26.54641
```

```r
format(as.Date(c(mu.hat, mean.finish), origin = "2012-06-30"), "%d %b")
```

```
R> [1] "10 Nov" "14 Feb"
```

```r
nr2 <- durationmean2ab(duration, mu.hat)
```

```r
pdf("probitplots.pdf")
plot(MIndex ~ Day, data = sanderlings, pch = 16, cex = 0.5,
     xlab = "days since July 1", ylab = "moult index (PFMG)")

abline(uz1, lwd = 2, col = "grey")
abline(nr1, lty = 2, lwd = 2)
abline(nr2, lty = 3, col = "red", lwd = 3)
legend(220, 0.3, lty = c(1, 2, 3), lwd = c(2, 2, 3),
       col = c("grey", "black", "red"), bty = "n",
       legend = c("UZ type 1", "NR separate", "NR combined"))
dev.off()
```

Because the NR method duplicates the data, the analysis is based on $2n$ non-independent data points. The log-likelihood of the RN2 model and that of the UZ1 model are therefore not comparable. Figure 3 compares estimates for timing and duration of moult obtained from the 2 NR models and the UZ model with type 1 data. UZ1 and NR2 give the same estimates. NR1 (with separate estimates for start and finish) estimates start of moult to be slightly earlier, and duration slightly longer.


## Weavers

Moult in South African weavers has been intensely studied by Oschadleus (2005). To demonstrate the use of covariates in moult models, the **moult** package includes a dataset on Southern Masked Weavers (*Ploceus velatus*), all caught in the Western Cape province of South Africa. In the Western Cape this species starts to breed during late winter, with the main breeding season starting only once the large cold fronts, typical for winter in this Mediterranean climate, have stopped. Like many other species it begins moult only after breeding has been completed. Here we will investigate the effect of September (southern hemisphere spring) rainfall on the timing of moult, assuming that total September rainfall (mm) can be taken as a measure of breeding suitability, predicting later start of moult with more September rain. The data contains observations from years 1988 to 2005, and from adult birds only.
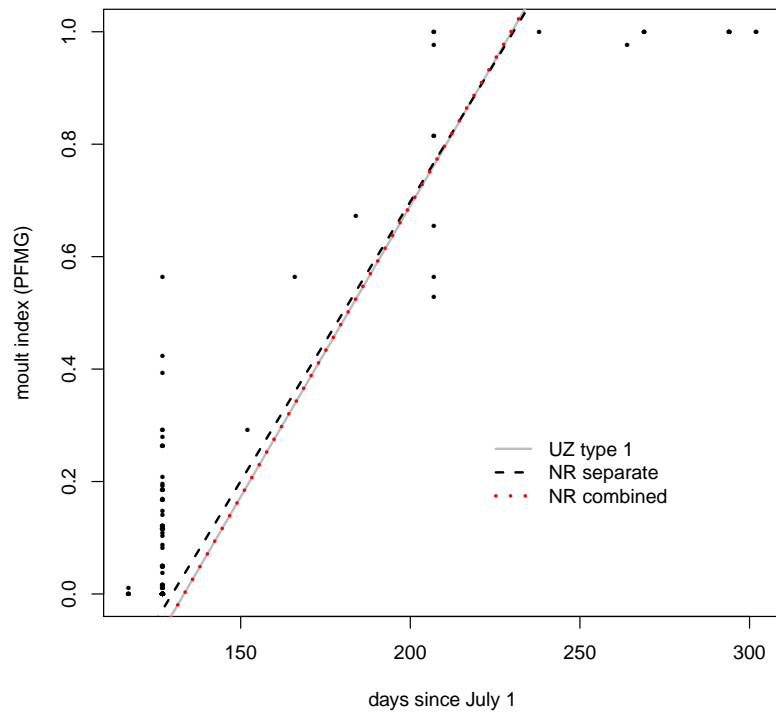
10

Figure 3: Observed moult scores (converted to 'proportion of feather mass grown') in sanderlings versus date. The lines represent estimated moult trajectories of a bird starting to moult on the mean start date using various models: UZ type 1 (solid grey line), NR with separate analyses for start and completion (dashed black line), and NR with combined analysis for start and completion (dotted red line, on top of grey line).

**Preliminaries**

In R reading in and converting from raw individual-feather moult scores can be tricky. If the moult scores for all primary feathers are kept as a single character string (e.g., of the form 5444320000), one can ensure that the column of moult scores is read in as a character vector by using the argument `colClasses` in `read.table()`. R might otherwise convert these to scientific notation or eliminate leading zeroes.

We first convert from individual feather scores to the "proportion of feather mass grown" using function `ms2pfmg()`. The proportion of feather mass grown is calculated by summing, over all feathers, the product of the proportion of feather grown and the relative mass of the feather (Underhill and Zucchini 1988). Individual feather masses for Southern Masked Weavers are given below (Craig et al. 2001). We use moult scores from the first nine primaries because the tenth is very small in this species. `date2days` converts dates to days since 1 August (= day 1). This `startmonth` is generally chosen as one in which no birds are in active moult.

```
data("weavers")
head(weavers)
```

```
R>          RDate Sex Year       Moult
R> 1 1999-07-24   2 1999 5555555555
R> 2 1999-07-24   1 1999 5555555555
R> 3 2000-05-19   1 2000 5555555555
R> 4 2002-10-21   1 2002 5555555555
R> 5 2003-11-19   2 2003 5555555555
R> 6 2004-12-03   4 2004 5555555555
```

```
if (is.numeric(weavers$Moult)) {
  scores <- format(weavers$Moult, scientific = FALSE, trim = TRUE)
} else {
  scores <- weavers$Moult
}

mscores <- substr(scores, 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)

## convert to days since 1 Aug, if earlier, convert to
## number of days since previous 1 Aug
weavers$day <-
  ifelse(as.Date(weavers$RDate) < as.Date(paste(weavers$Year, "-08-01", sep = "")),
         as.Date(weavers$RDate) - as.Date(paste(weavers$Year - 1, "-07-31", sep = "")),
         as.Date(weavers$RDate) - as.Date(paste(weavers$Year, "-07-31", sep = "")))

## same, but date2days will be removed from future version
weavers$day <- date2days(weavers$RDate, dateformat = "yyyy-mm-dd",
                          startmonth = 8)
```

**Two simple models**

In many moult data sets individuals coded with all new feathers and/or all old feathers appear throughout the year. This is also the case in our weaver data set (Figure 4). It becomes impossible to obtain a sensible estimate for the standard deviation in start of moult (see output from model `weav.t2` below). To avoid this problem, one could change all 55555555 at the start to 000000000, and all 000000000 at the end to 555555555 (moult complete). In practice, it can be difficult to decide how close to the moult period this adjustment

should be made. When the ratio of non-moulting birds compared to moulting birds is higher than expected, sub-sampling of non-moulting birds can improve estimates [Bonnevie (2010)}. Here, we will only consider birds actively moulting (data type 3).

```
## type 2 data
weav.t2 <- moult(pfmg ~ day, data = weavers)
summary(weav.t2)
```

```
R>
R> Call:
R> moult(formula = pfmg ~ day, data = weavers)
R>
R>
R> Duration coefficients:
R>           Estimate Std. Error
R> intercept.1    83.46      2.971
R>
R> Mean start date coefficients:
R>          Estimate Std. Error
R> intercept     181.5      2.839
R>
R> Coefficients for standard deviation in start date:
R>           Estimate Std. Error
R> (Intercept)    146.7      23.05
R>
R> Log-likelihood: -7460 on 3 Df
```

```
## type 3 data
weav.t3 <- moult(pfmg ~ day, data = weavers, type = 3)
summary(weav.t3)
```

```
R>
R> Call:
R> moult(formula = pfmg ~ day, data = weavers, type = 3)
R>
R>
R> Duration coefficients:
R>            Estimate Std. Error
R> intercept.1    84.48      11.95
R>
R> Mean start date coefficients:
R>           Estimate Std. Error
R> intercept     152.7      6.427
R>
R> Coefficients for standard deviation in start date:
R>            Estimate Std. Error
R> (Intercept)    31.88      8.326
R>
R> Log-likelihood: -2675 on 3 Df
```

```
## mean moult trajectory
traj.t3 <- durationmean2ab(coef(weav.t3, "duration"), coef(weav.t3, "mean"))
```

**Start and duration depend on sex**

We now fit a model where we let start and duration of moult depend on sex.

```r
ssex <- ifelse(weavers$Sex == 1 | weavers$Sex == 3, 'male',
        ifelse(weavers$Sex == 2 | weavers$Sex == 4, 'female', NA))

weavers$ssex <- as.factor(ssex)

weav.mf <- moult(pfmg ~ day | ssex | ssex, data = weavers, type = 3)
summary(weav.mf)
```

```
R>
R> Call:
R> moult(formula = pfmg ~ day | ssex | ssex, data = weavers, type = 3)
R>
R>
R> Duration coefficients:
R>            Estimate Std. Error
R> intercept.1    54.78       10.862
R> ssexmale       34.94        9.164
R>
R> Mean start date coefficients:
R>           Estimate Std. Error
R> intercept   161.39        6.183
R> ssexmale    -10.67        5.802
R>
R> Coefficients for standard deviation in start date:
R>             Estimate Std. Error
R> (Intercept)     29.4        7.897
R>
R> Log-likelihood: -2411 on 5 Df
```

The coefficient `ssexmale` is an estimate of the difference between males and females, for both, duration and mean start date. Thus males seem to take longer to moult but start earlier (Figure 4). Underhill and Zucchini (1988) have noted that, especially for type 3 models, start and duration estimates are often highly negatively correlated.

```r
## females
fstart <- coef(weav.mf, "mean")[1]
fduration <- coef(weav.mf, "duration")[1]
female.traj <- durationmean2ab(fduration, fstart)

## males
mstart <- sum(coef(weav.mf, "mean")[1:2])
mduration <- sum(coef(weav.mf, "duration")[1:2])
male.traj <- durationmean2ab(mduration, mstart)
```

```r
pdf("weaverplot2.pdf")
plot(pfmg ~ day, pch = 16, cex = 0.5, xlab = "days since August 1",
     ylab = "proportion of feather mass grown", las = 1, col = "grey",
     data = weavers)
abline(traj.t3, lwd = 2)
abline(male.traj, col = "blue", lty = 2, lwd = 3)
abline(female.traj, col = "red", lty = 4, lwd = 3)
legend("bottomright", lty = c(1, 2, 4), lwd = 2, col = c(1, "blue", "red"),
```
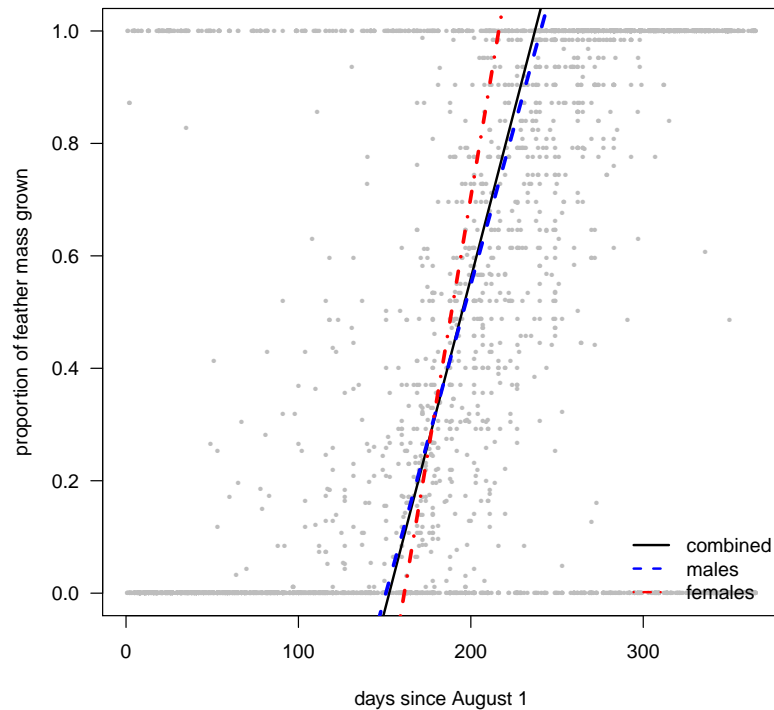
Figure 4: Observed moult scores (converted to 'proportion of feather mass grown') of Southern Masked Weavers versus date. The lines represent estimated moult trajectories of birds starting at the mean start date for all birds (solid black line), females (dotted-and-dashed red line), and males (dashed blue line).

```
        legend = c("combined", "males", "females"), bty = "n")
dev.off()
```

**Start of moult depends on year**

In the following example, year is defined as a factor, and the (mean) moult start date is allowed to vary between years, duration of moult is assumed constant.

```
weavers$year.f <- as.factor(weavers$Year)
m88y <- moult(pfmg ~ day | 1 | year.f, data = weavers, type = 3)
summary(m88y)
```

```
R>
R> Call:
R> moult(formula = pfmg ~ day | 1 | year.f, data = weavers, type = 3)
R>
R>
R> Duration coefficients:
R>             Estimate Std. Error
R> intercept.1    87.97      11.93
R>
R> Mean start date coefficients:
```

15

```
R>              Estimate Std. Error
R> intercept  147.41804        396.4
R> year.f1989 -28.95514        396.7
R> year.f1990 -23.34579        396.8
R> year.f1991 -15.34127        396.5
R> year.f1992  -5.11460        396.4
R> year.f1993  10.69091        396.4
R> year.f1994   5.53374        396.4
R> year.f1995   6.54268        396.4
R> year.f1996  21.93591        396.6
R> year.f1997   0.57239        396.5
R> year.f1998  10.34152        396.5
R> year.f1999   7.59767        396.4
R> year.f2000  -7.72685        396.4
R> year.f2001  -0.08902        396.4
R> year.f2002   5.46045        396.4
R> year.f2003  -4.37632        396.4
R> year.f2004   4.17356        396.4
R> year.f2005  17.16040        396.5
R>
R> Coefficients for standard deviation in start date:
R>              Estimate Std. Error
R> (Intercept)     31.81      8.299
R>
R> Log-likelihood: -2409 on 20 Df
```

```r
weavers$pfmg[weavers$Year == 1988]
```

```
R>  [1] 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000
R>  [8] 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000
R> [15] 0.0000000 0.0000000 0.0000000 0.9840706
```

With the above model the standard errors for the year effects could not be estimated, hence also the warning message. The problem seems to be year 1988, which had only a single individual in active moult. We next remove this year from the data frame. Using `data = weavers[weavers$Year > 1988]` here leads to problems, because this results in a factor with unused levels (level 1988), therefore we create a new data frame `weav89`.

```r
weav89 <- weavers[weavers$Year >= 1989, ]
weav89$year.f <- as.factor(weav89$Year)
m89y <- moult(pfmg ~ day | 1 | year.f, data = weav89, type = 3)
summary(m89y)
```

```
R>
R> Call:
R> moult(formula = pfmg ~ day | 1 | year.f, data = weav89, type = 3)
R>
R>
R> Duration coefficients:
R>            Estimate Std. Error
R> intercept.1    87.41      11.98
R>
R> Mean start date coefficients:
R>            Estimate Std. Error
R> intercept   128.022      17.26
R> year.f1990    1.392      24.68
```

```
R> year.f1991     5.165        18.85
R> year.f1992    15.112        17.48
R> year.f1993    29.139        17.91
R> year.f1994    25.209        16.88
R> year.f1995    27.274        16.84
R> year.f1996    36.837        23.81
R> year.f1997    19.013        20.10
R> year.f1998    28.330        19.86
R> year.f1999    27.065        18.46
R> year.f2000    10.668        18.50
R> year.f2001    19.334        17.48
R> year.f2002    25.212        17.34
R> year.f2003    15.405        16.91
R> year.f2004    23.129        16.86
R> year.f2005    34.719        18.81
R>
R> Coefficients for standard deviation in start date:
R>           Estimate Std. Error
R> (Intercept)    31.65       8.278
R>
R> Log-likelihood: -2409 on 19 Df
```

The differences in mean starting date between years seem large, however the standard errors are also large. Therefore it seems that there is no evidence for differences in starting dates between years. The year estimates are differences (in days) compared to the reference year 1989.

To obtain the fitted values of mean start date of moult for each year, we can use the `predict` function:

```
pred.year <- predict(m89y, predict.type = "start",
                     newdata = data.frame(year.f = as.factor(1989:2005)))
```

**Start of moult depends on rainfall**

We now model start of moult as a function of a continuous covariate September rainfall at the Cape Town Astronomical Observatory (South African Weather Service), first as a linear function, then as a quadratic function of rainfall.

```
rainSep <- c(44.8, 110.2, 24.3, 84.4, 73.1, 10.5, 32.4, 3.8, 91.2,
             8.1, 27.8, 112.4, 58.7, 111.5, 20, 66.3, 43, 14.5)
weavers$rain <- rainSep[match(weavers$Year, 1988:2005)]

m88r <- moult(pfmg ~ day | 1 | rain, data = weavers, type = 3)
m88r2 <- moult(pfmg ~ day | 1 | rain + I(rain^2), data = weavers, type = 3)
summary(m88r2)
```

```
R>
R> Call:
R> moult(formula = pfmg ~ day | 1 | rain + I(rain^2), data = weavers,
R>     type = 3)
R>
R>
R> Duration coefficients:
R>           Estimate Std. Error
R> intercept.1    86.67       11.98
R>
```

17

```
R> Mean start date coefficients:
R>             Estimate Std. Error
R> intercept 159.161681      7.621
R> rain       -0.325919      0.196
R> I(rain^2)   0.001862      0.002
R>
R> Coefficients for standard deviation in start date:
R>             Estimate Std. Error
R> (Intercept)    31.74      8.198
R>
R> Log-likelihood: -2416 on 5 Df
```

Comparing these models using Akaike's information criterion:

```
AIC(m88y, weav.t3, m88r, m88r2)
```

```
R>          df      AIC
R> m88y     20 4857.593
R> weav.t3   3 5355.586
R> m88r      4 4840.891
R> m88r2     5 4841.382
```

`weav.mf` had many missing observations due to unknown sex, and is thus not included in the AIC comparison. Out of the remaining models, the AIC selects model `weav.t3` (constant start date), which confirms the above conclusions that there is no clear evidence for differences between years, nor for a relationship between mean starting date and September rainfall (Figure 5).

```
pdf("weaverplot1.pdf")
par(mar = c(7, 9, 1, 1), mgp = c(5, 1, 0))
plot(1988:2005, c(NA, pred.year[, 1]), type = "b", lwd = 1, pch = 19,
     xlab = "", ylab = "mean start of moult",
     ylim = c(70, 210), las = 1, yaxt = "n")
abline(h = coef(weav.t3, "mean"), col = "grey")
upp <- pred.year[, 1] + 1.96 * pred.year[, 2]
lwr <- pred.year[, 1] - 1.96 * pred.year[, 2]
arrows(x0 = 1989:2005, y0 = lwr, y1 = upp, code = 3, angle = 90,
       length = 0.05)
ylab <- format(as.Date(seq(70, 210, by = 20), origin = "2012-07-31"), "%d %b")
axis(2, at = seq(70, 210, by = 20), labels = ylab, las = 1)
dev.off()
```

## Discussion

More than 20 years ago Underhill and Zucchini (1988) published an improved method to analyse moult in birds. These UZ models, and the probit models used by Rothery and Newton (2002), are still considered the best way to analyse moult. However, easy-to-use software to fit the UZ models was not widely available. The **moult** package for R aims to provide all functions required for the analysis of moult data in an easy-to-use format. Here, we have demonstrated its functionality. In addition to the original moult models published in Underhill and Zucchini (1988) and Underhill et al. (1990), in **moult** the duration, mean and standard deviation of start of moult parameters can be modelled as functions of explanatory variables. This will allow researchers to ask and answer important ecological questions, and can be seen as a further advantage of the UZ models: because duration and mean start date (and standard deviation in start date) are the parameters of the likelihood, it is straightforward to extend the likelihood to model these parameters as functions of covariates.
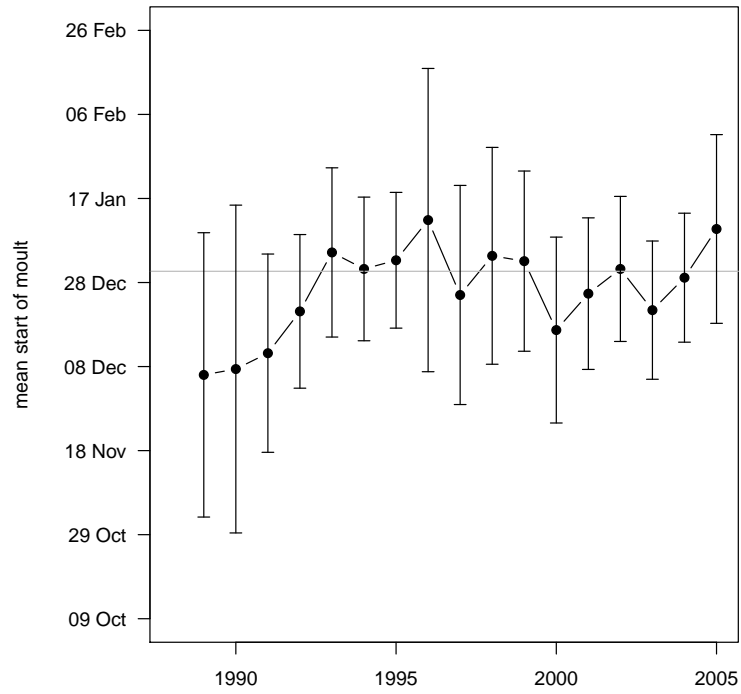
Figure 5: Estimated mean start dates of moult in Southern Masked Weavers, 1988-2005, from model with year effect (m89y, solid black line), and model with constant mean start date (weav.t3, grey line). Error bars are 95% confidence intervals.

The UZ moult models can be sensitive to certain types of data problems. One of these problems is individuals that start moulting markedly later or earlier than the bulk of the population. Such outliers can result in large estimates of the standard deviation parameter $\sigma$, and may even lead to non-convergence. A second, common, problem is the presence of zero (not started) or 1 (completed) moult scores throughout the season. In our experience, for example, with the weaver data in this manuscript, it is not straightforward to choose cut-off points to decide whether a bird not in moult has all old or all new feathers, and estimates can be sensitive to these cut-off points. Optimization of the log-likelihood can be sensitive to starting values, especially for standard deviation estimates, so it is advisable to try different starting values for complex models.

The critical assumption made by the UZ models of constant rate of change in moult index, or a linear increase in moult index over time, may not hold in reality (Summers, Swann, and Nicoll 1980; Newton and Rothery 2000). This assumption is not required when using the probit model (Rothery and Newton 2002), or the UZ models with type 1 data. Newton and Rothery (2000) compared the UZ models for data types 1 to 3 and found that type 1 models gave more accurate estimates of timing and duration of moult, especially when the linearity assumption was not met. The probit model and the UZ type 1 model are almost identical. The advantage of the UZ model is that, because start and duration of moult are the parameters being estimated directly, it is easier to add covariates.

# References

Bonnevie, B. T. 2010. "Balancing Moult Data by Subsampling Non-Moulting Birds Prior to Regression Analysis." *Ostrich* 81 (3): 265–68.

Chambers, J. M., and T. J. Hastie, eds. 1992. *Statistical Models in S*. London: Chapman & Hall.

Craig, A. J. F. K., P. E. Hulley, C. A. Whittington-Jones, and B. T. Bonnevie. 2001. "Flying Times and Flight Feathers: Patterns of Moult in Sympatric Seed-Eaters." *Ostrich Supplement 15* 72.

Erni, B., H.-D. Oschadleus, B.T. Bonnevie, R. Altwegg, and L. G. Underhill. 2013. "Moult: An R Package to Analyse Moult in Birds." *Journal of Statistical Software* 52 (8): 1–23. https://www.jstatsoft.org/article/view/v052i08.

Ginn, H. B., and D. S. Melville. 1983. *Moult in Birds. Bto Guide 19.* British Trust for Ornithology, Tring.

Newton, I. 2009. "Moult and Plumage." *Ringing & Migration* 24: 220–26.

Newton, I., and P. Rothery. 2000. "Timing and Duration of Moult in the Bullfinch *Pyrrhula Pyrrhula*: An Appraisal of Different Analytical Procedures." *Ibis* 142: 65–74.

Oschadleus, H.-D. 2005. "Patterns of Primary Moult in the Weavers, Ploceidae." PhD thesis, University of Cape Town.

Pimm, S. 1976. "Estimation of the Duration of Bird Molt." *Condor* 78: 550.

R Core Team. 2017. *R: A Language and Environment for Statistical Computing.* Vienna, Austria: R Foundation for Statistical Computing. https://www.R-project.org/.

Rothery, P., and I. Newton. 2002. "A Simple Method for Estimating Timing and Duration of Avian Primary Moult Using Field Data." *Ibis* 144: 526–28.

Summers, R. W., R. L. Swann, and M. Nicoll. 1980. "Unbending Moult Data." *Wader Study Group Bulletin* 30: 12–13.

———. 1983. "The Effects of Methods on Estimates of Primary Moult Duration in the Redshank *Tringa Totanus*." *Bird Study* 30: 149–56.

Underhill, L. G. 1985. "Estimating the Parameters for Primary Moult - a New Statistical Model." *Wader*

*Study Group Bulletin* 44: 27–29.

Underhill, L. G., and W. Zucchini. 1988. "A Model for Avian Primary Moult." *Ibis* 130: 358–72.

Underhill, L. G., L. Serra, and A. Brandao. 2008. "Progress with the Statistical Analysis of Primary Moult." In *Proceedings of the Xxiii International Ornithological Congress, Beijing, China. Acta Zoologica Sinica*, 52 Supplement:440–43.

Underhill, L. G., W. Zucchini, and R. W. Summers. 1990. "A Model for Avian Primary Moult Data Types Based on Migration Strategies and an Example Using the Redshank *Tringa Totanus*." *Ibis* 132: 118–23.

Venables, W. N., and B. D. Ripley. 2002. *Modern Applied Statistics with S*. 4th ed. Springer-Verlag.

Zeileis, A., C. Kleiber, and S. Jackman. 2008. "Regression Models for Count Data in R." *Journal of Statistical Software* 27 (8): 1–25. http://www.jstatsoft.org/v27/i08/.