Alain F. Zuur and Elena N. Ieno

The World of Zero-Inflated Models
Volume 1: Using GLM

ISBN: 978-1-7399636-0-6
First published December 2021
First print

www.highstat.com

I dedicate this book to two little stars and one big star.
- Alain F. Zuur -

I dedicate this book to all families and friends who have lost loved ones as a result of this coronavirus pandemic. If there is one thing I have reaffirmed during this difficult time, it is my commitment to the essential teaching of the method of statistical analysis and respect for science.
- Elena N. Ieno -

Contents

Preface

Why this book?

In 2009 we published the book *Mixed Effects Models and Extensions in Ecology with R* (Zuur et al., 2009b). One chapter in this book was dedicated to the analysis of zero-inflated data using the `pscl` package (Jackman et al., 2020). This package can be used for the analysis of data sets with an excessive number of zeros in the response variable using zero-inflated generalised linear models (GLMs). That chapter has been cited quite often in the literature, underscoring the niche for zero-inflated models.

In 2012 we published a book dedicated solely to the analysis of zero-inflated models (Zuur et al., 2012). We implemented zero-inflated GLMs and also zero-inflated generalised linear mixed-effects models (GLMMs). For the zero-inflated GLMs we used again the `pscl` package, which is a pleasant package to work with. For the zero-inflated GLMMs, we had to use `glmmADMB` and WinBUGS as there were not many other options available at the time. Markov chain Monte Carlo simulation in WinBUGS requires complicated coding and is based on Bayesian statistics. And in 2012, `glmmADMB` was quite limited in what it could do with respect to zero-inflated GLMMs.

We then went on with zero-inflated GLMs and GLMMs in Zuur and Ieno (2016). Our intention was to write a second edition of the 2012 book, but we ended up with a (big) piece of text in which not a single line was taken from the 2012 book. Hence, it became a completely new book. We used JAGS instead of WinBUGS. JAGS is certainly more user friendly and is also cross-platform. But JAGS still requires a steep Bayesian learning curve and long computing time.

Since 2003 we have been teaching about 25 five-day statistics courses per year, all over the world. Since the very first course, we have been contacted by many course participants requesting help with statistical models that can cope with zero inflation and spatial correlation. And some participants even needed models for zero-inflated data with spatial-temporal correlation and non-linear covariate effects. We implemented such models in WinBUGS in Zuur et al. (2015). During this process, we even managed to burn a laptop due to overheating! We made some progress with these models using JAGS in Zuur

et al. (2015) and Zuur and Ieno (2016), but computing time for zero-inflated generalised additive models (GAMs) with spatial correlation was rather long.

The development of R-INLA with spatial GLMs (Rue et al., 2017) was a gift from heaven. It allowed us to implement zero-inflated GLMs and GAMs with spatial, and spatial-temporal dependency. We wrote two volumes in which we applied zero-inflated GLMs and GAMs with spatial, and spatial-temporal dependency (Zuur et al., 2017, Zuur and Ieno (2018)).

The only problem with R-INLA is that it requires a fair amount of (Bayesian) knowledge, and the approach is overkill in the event there is no spatial or spatial-temporal dependency. For this reason, we decided to write the text that is now in front of you. Our primary aim was to write a book that explains in simple language how to implement models for zero-inflated data and also for zero-inflated clustered data. By clustered data we mean multiple observations from the same site, classroom, patient, animal, etc. This type of data requires a linear mixed-effects model or a GLMM, but for zero-inflated data. Between 2018 and 2021 we also noticed that the `glmmTMB` package (Magnusson et al., 2021) became more stable, more user-friendly, and above all, the distributions required for zero-inflated models were implemented. This means that if you have data that require zero-inflated models, and also have dependency, then `glmmTMB` provides an easy-to-use platform for such models. We therefore decided to use the `glmmTMB` package throughout this book. The package is easy to use and is available for free within the R software.

Acknowledgements

We greatly appreciate the efforts of those who wrote R (R Core Team, 2020) and its many packages. This book would not have been possible without the efforts of the `glmmTMB` (Magnusson et al., 2021), `ggplot2` (Wickham et al., 2021) and `pscl` (Jackman et al., 2020) programmers. We hope that they will keep up the excellent work.

We are greatly indebted to all scientists who supplied data for this book. We thank Jan van Gils (van Gils et al., 2016) for providing the red knot data and Juan Timi for the Brazilian sandperch data (Timi et al., 2008). We also thank Nettleship (1972), Matula et al. (2015), Jackson-Ricketts et al. (2020) and Long (2017) for making their data available online.

Special thanks to Christine Andreasen for editing this book. We thank ir. Thierry Onkelinx for reviewing Appendix A

Data sets and R code used in this book

All data sets used in this book may be downloaded from `http://highstat.com/index.php/books`. All R code also may be downloaded from the website for this book. To open the ZIP files with R code, use the password <omitted in online pdf>.

Cover art

The cover drawing is by Jon Thompson (www.yellowbirdgallery.org). Mr Thompson was born in 1939 to Irish parents and has lived most of his life in Scotland. In the 1980s, he was drawn to the Orkney Islands. He is continually inspired by the landscape and bird life of Orkney. He has been creating bird art for 30 years in a variety of media, including drawing, painting, sculpture, and jewellery, never attempting to reproduce nature, but instead drawing parallels with it. A close-up view of a bird feather is all the inspiration he needs.

Alain F. Zuur
Newburgh, Scotland

Elena N. Ieno
Allicante, Spain

Acronyms

ACRONYM	TERM
3-D	three dimensional
AIC	Akaike information criterion
BIC	Bayesian information criterion
CI	confidence interval
CMP	Conway–Maxwell–Poisson
DHARMa	Diagnostics for HierArchical Regression Models
GAM	generalised additive model
GAMM	generalised additive mixed model
GLM	generalised linear model
GLMM	generalised linear mixed model
GP	generalised Poisson
KS	Kolmogorov–Smirnov test
LASSO	least absolute shrinkage and selection operator
LS	sagittal length
LT	total length
MCMC	Markov chain Monte Carlo
MSE	mean-square errors
NB	negative binomial
Q–Q	quantile–quantile
VIFs	variance inflation factors
ZAG	zero-altered Gamma
ZANB	zero-altered negative binomial
ZAP	zero-altered Poisson
ZIGP	zero-inflated generalised Poisson
ZINB	zero-inflated negative binomial
ZIP	zero-inflated Poisson as in ZIP model

1

Introduction

1.1 The world of zero-inflated models

The world of zero-inflated models is large and complex. The first layer of complexity is the type of data. In Volumes 1 to 3 we will analyse count data, continuous data, proportional data, density data, etc.

The second layer of complexity is that, for each of these data types, we have multiple options for choosing a statistical distribution. For count data we will discuss the Poisson, negative binomial (NB), generalised Poisson (GP) and Conway–Maxwell–Poisson (CMP) distributions. For continuous data we will apply the Tweedie distribution and the zero-altered Gamma (ZAG) approach. For proportional data we will use the binomial and the beta distributions.

The third layer of complexity is pseudo-replication. We may have multiple observations from the same site, animal, person, etc. This brings us within the world of linear mixed-effects models and generalised linear mixed models (GLMMs). This is the topic of Volume 2.

The fourth layer of complexity is that some covariates may have a non-linear effect, which may require generalised additive models (GAMs). If your data sets requires zero-inflated GAM or zero-inflated generalised additive mixed models (GAMMs), then Volume 3 of this series will help you analyse your data. If on top of this, you also have spatial, temporal, or spatial-temporal dependency, then there is no escape from R-INLA. Note that with spatial dependency, we assume that you have 50+ spatial locations. And temporal dependency becomes relevant if you have 15+ measurements over time. Zero-inflated spatial and spatial-temporal models are discussed in Zuur and Ieno (2018).

1.2 Volumes 1, 2 and 3

The text before you is Volume 1 of *The World of Zero-Inflated Models*. We will discuss models for count data and continuous data with an excessive number of zeros. All models are extensions of generalised linear models (GLMs). In Volume 2, we will analyse count data, continuous data and proportional data using GLMMs. Hence, we will extend the models from Volume 1 with random intercepts and random slopes. In Volume 3, we will extend the models from Volumes 1 and 2 towards GAMs and GAMMs. A schematic overview of Volumes 1 – 3 is presented in Figure 1.1.

Volume 1 can be read as a stand-alone. Volume 2 assumes that you have read Volume 1, and Volume 3 is a continuation of Volume 2.

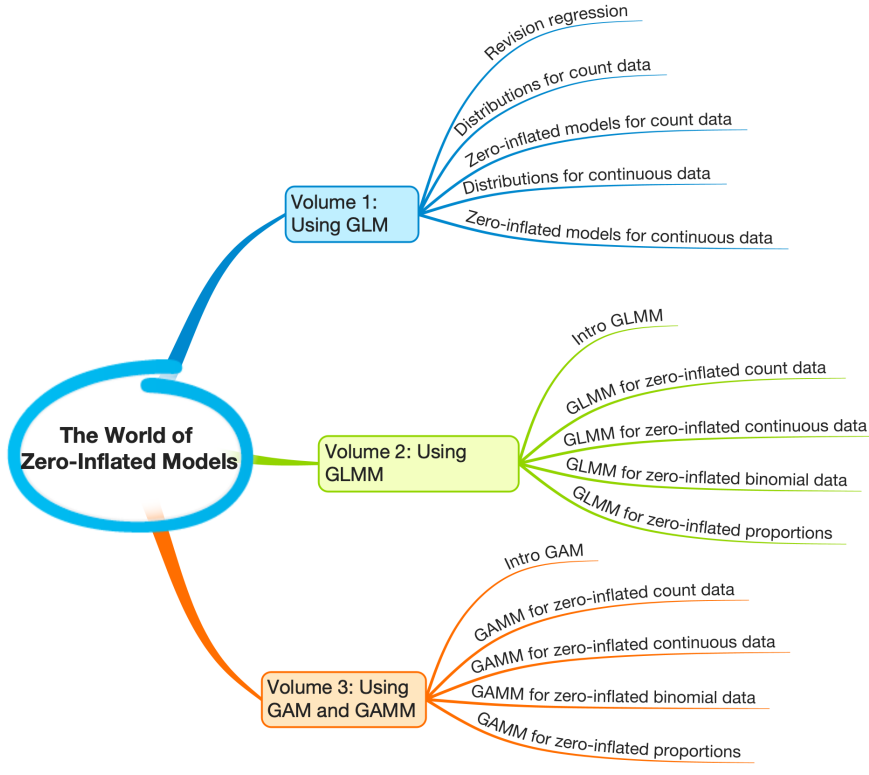

FIGURE 1.1: Outline of Volumes 1, 2 and 3.

1.3 Outline of this book (Volume 1)

In Chapter 2 we revise data exploration and multiple linear regression using red knot data. Stable isotope ratios of nitrogen in animal tissues are modelled as a function of 3 covariates. This chapter serves as a blueprint for all other chapters in the sense that it shows the general outline of a statistical analysis.

Chapter 3 starts with a revision of the Poisson distribution and the Poisson GLM for the analysis of count data. We use a small puffin data set. We also introduce the NB GLM and two relatively unknown, but useful, members of the family, namely the GP GLM and the CMP GLM. Surprisingly, the latter two models tend to perform better than the NB GLM in the case of overdispersion. The latter two can also be used to deal with underdispersion. Most models are fitted with the `glmmTMB` package in R. Model validation tools are explained, and the concept of simulating data from a model (to verify whether it complies with all assumptions of the model) is introduced. We first do the simulation steps ourselves, then quickly migrate to the `DHARMa` package, which is rapidly gaining popularity.

In Chapter 4 we introduce zero-inflated models for count data, and these are executed with the `glmmTMB` package. We start with a basic introduction using simulated data, and discuss zero-inflated Poisson (ZIP), zero-inflated NB (ZINB), zero-inflated generalised Poisson (ZIGP) and zero-inflated CMP (ZICMP) models. We then apply them all on the puffin data set.

In Chapter 5 we analyse data on parasites in Brazilian sandperch. Such data nearly always bring you within zero-inflation territory. Now that we are familiar with Poisson, NB, GP, CMP models, and their zero-inflated cousins, it is time to learn how we can manoeuvre among them. How do we decide to apply an NB GLM or a ZIP model? In this chapter, we will keep the binary part of the model simple.

Chapter 6 is about ZIGP models. Data on mistletoe tree infections are used. The ZIGP models contain covariates in both the count and binary parts of the model.

Hurdle models for count data are discussed in Chapter 7 using dolphin sighting data. In a hurdle model we perform 2 analyses. First, the sighting abundances are converted into absence/presence data, and a Bernoulli GLM is applied. Then the zero counts are set to NA (or dropped), and a truncated Poisson (or NB) GLM is applied. In the third step, the two components are combined to calculate the expected values of the hurdle model. Chapter 7 is relatively long as it contains many topics that may be relevant: Bernoulli GLM, quasi-separation, truncated Poisson and NB distributions, and zero-altered Poisson (ZAP) and zero-altered NB (ZANB) models.

In the last 2 chapters of this volume, we discuss models for the analysis of continuous data with an excessive number of zeros. Biomass of lobsters are analysed using Tweedie GLMs in Chapter 8, and a ZAG model is applied on the same data in Chapter 9. The ZAG is a hurdle model for continuous data. Our recommendation is to opt for the Tweedie GLM approach.

Figure 1.2 shows a schematic outline of this volume.

FIGURE 1.2: Outline of Volume 1.

1.4 E-access to our 2012 book

Although the WinBUGS part of our 2012 book is out of date, the book still contains a wealth of information on zero-inflated GLMs using `pscl`. Readers who have purchased Volumes 1, 2 or 3 of *The World of Zero-Inflated Models* can request free electronic access to our 2012 book (Zuur et al., 2012). If you email Dr. Alain F. Zuur at highstat@highstat.com[1], and quote the reference 'deleted in online version' then you will be given a link to a protected pdf file that can be opened with Adobe Digital Edition (ADE). Note that no support or refund will be given in the event your computer cannot open a protected pdf file in ADE. We will only provide one link per person. Please see `http://highstat.com/index.php/order-books-or-e-books` for general in-formation on our E-books.

1.5 Decision map

Figure 1.3 shows a decision map for most of the techniques that are used in this book. You can either have a look at this figure now or first read the book and then use this figure for the analysis of your own data.

If the response variable consists of counts, then go to step 1. If the counts can theoretically not obtain the value of zero, and the observed values are relatively small, then consider zero-truncated models. If the data consist of zeros and positive counts, then start with a Poisson GLM. In the event of overdispersion, figure out why there is overdispersion (e.g. the presence of outliers, non-linear relationships, missing interactions, dependency that has not been included, etc.). If overdispersion is due to the variance being larger than the mean, then consider applying the GP, NB or CMP GLMs. Only if the Poisson GLM cannot cope with the excessive number of zeros, should you then apply a ZIP GLM. If this model is overdispersed, then figure out why there is overdispersion (e.g. the presence of outliers, non-linear relationships, missing interactions, dependency that has not been included, etc.). Only if overdispersion is due to the variance being larger than the mean, then consider applying the ZIGP, ZINB or ZICMP GLMs. Never start the analysis of your data with these models!

If abundance data are analysed, and if abundance is defined as numbers divided by, for example, sampling effort, then use the log of the denominator

[1] `mailto:highstat@highstat.com`

(sampling effort) as a covariate, or as an offset. See Zuur et al. (2014) for how to deal with an offset variable in a GLM. Alternatively, use the denominator itself (and not its log) as a covariate. Note that in all these cases the response variable is the numbers (nominator in the abundances). If the abundance data are only available as a continuous variable, then go to option 3.

If the response variable is continuous (e.g. biomass or density), then do not apply a model for count data. If the data are negative, zero and positive, then apply a linear regression model. If the data contain zeros and positive data, then the Tweedie GLM is an option and so is the linear regression model (provided that it does not have negative fitted values). If the data are strictly positive, then the Gamma GLM is an option. The inverse Gaussian GLM (not discussed in this volume) is also an option, and so is the Tweedie GLM. The Tweedie GLM, as implemented in `glmTMB`, assumes that theoretically there can be zeros. If the data contain zeros and continuous data, then the Tweedie GLM is a good option. The ZAG GLM can also be applied, provided that a fair number of observations are absent, and also present. The reason for this is that the ZAG GLM consists of multiple models, and one of the models is a Bernoulli GLM that is applied on absence/presence data. Such a model is likely to give numerical problems (quasi-separation issues) when 90 observations equal 0, and 10 observations have values larger than 0. The zero-altered log-normal model is an alternative to the ZAG model, but it is not discussed in this volume.

If your data are proportional or binomial, then see Volume 2, in which the (zero-inflated) binomial and beta GLMMs are introduced. If you have repeated measurements (hierarchical data, clustered data, short time series), then see Volume 2, in which we discuss (zero-inflated) GLMMs. If you have spatial, temporal or spatial-temporal dependency, then see our two INLA volumes (Zuur et al., 2017; Zuur and Ieno, 2018). If you have non-linear relationships, zero-inflation problems and repeated measurements, then see Volume 3, in which we use GAMs and GAMMs for such data.

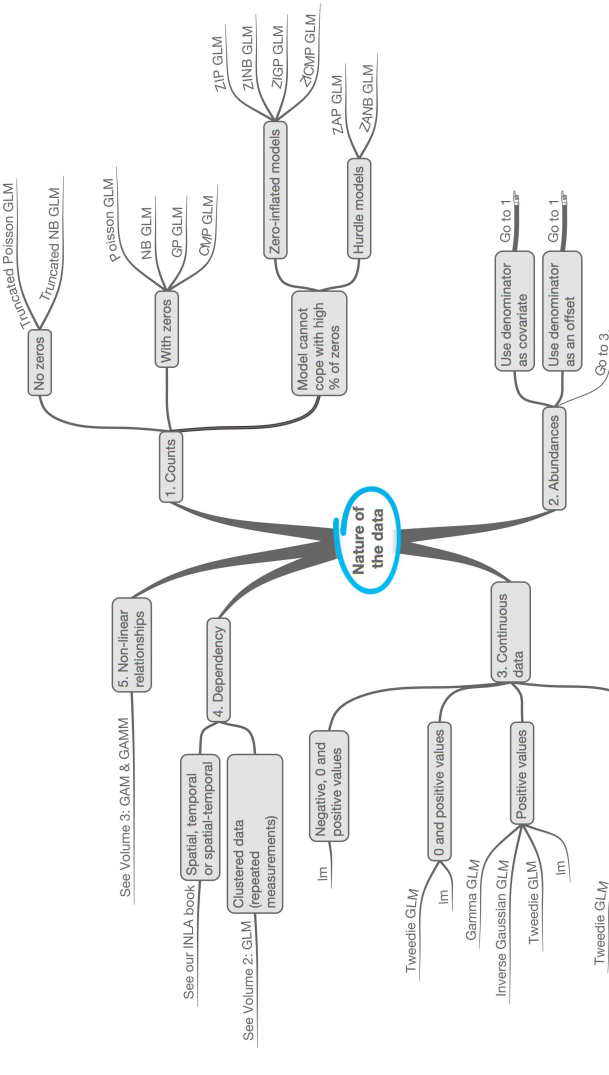

FIGURE 1.3: Decision map. A high-resolution colour version is available on the website for this book.

2

Linear regression applied on red knot data

In this chapter we will review data exploration and multiple linear regression. The reason for doing this is that the analyses that will be applied in later chapters will follow similar steps. It is not our intention to explain data exploration and multiple linear regression in detail. For an introduction to data exploration we refer the reader to Zuur et al. (2010) or Ieno and Zuur (2015). Linear regression is explained in detail in Quinn and Keough (2002), Montgomery et al. (2012) or Zuur et al. (2007).

We assume familiarity with basic R coding. Even if you have not worked with R before, it is relatively easy to learn R while working through the chapters of this book. For an introduction to R, see Dalgaard (2008) or Zuur et al. (2009a).

2.1 Red knot

The red knot *Calidris canutus canutus* experienced globally unrivaled warming rates at its high-Arctic breeding grounds and produced smaller offspring with shorter bills during summers with early snow melt (van Gils et al., 2016).

As a result, the red knot eats fewer deeply buried bivalve prey and more shallowly buried seagrass rhizomes at their wintering grounds in the tropics. This bill length–dependent diet shift may be explained by the depth distribution of *Loripes lucinalis*, the preferred energy-content prey that only long-billed knots incorporate in their diet. The authors suggest that seasonal migrants experience reduced fitness and therefore lower survival rates due to a changing climate.

In this chapter, we will use part of the data that was analysed in van Gils et al. (2016). Red knots were caught using mist-nets. Upon capture, each bird was aged, body measurements were made and a small blood sample was taken. The birds were ringed before release.

The analysis of stable isotope ratios of carbon and nitrogen in animal tissues has become accepted and commonly applied in ecological studies, thanks to its simplicity and the fact that isotope ratios reflect diet over a relatively long time. The technique is widely used in the study of the feeding ecology of higher oceanic predators such as marine mammals, large pelagic fish, turtles and birds. We will analyse the nitrogen isotope ratio values d15N that were taken from the blood samples of the birds.

2.2 Importing the data

We first import the data with the `read.table` function. We assume that the data are in the directory `Data` inside the working directory (use `setwd` to set the working directory). Character strings are automatically imported as categorical variables (`stringsAsFactors = TRUE`), and `NA` is used for missing values. The text file uses points for decimal separation (`dec = "."`).

```
RK <- read.table("./Data/RedKnots.txt",
                 dec = ".",
                 na.strings = "NA",
                 stringsAsFactors = TRUE,
                 header = TRUE)
```

We have 474 observations in the `RK` object. Each of these observations contains measurements for 8 variables from a specific bird. The following variables are available in the `RK` object.

```
names(RK)
```

```
## [1] "Ring"   "Age"    "d13C"   "d15N"   "Year"   "Wing"
## [7] "Bill"   "Tarsus"
```

The response variable is `d15N`. The variables `Wing`, `Bill` and `Tarsus` are morphometric variables and are measured on the continuous scale. `Age` is a categorical variable with values `adult` and `juvenile`. The variable `Ring` identifies

the bird. Finally, `Year` is an integer, but because we only have data from 2 years, we will treat it as a categorical covariate.

We load a series of packages and also our support file `HighstatLibV13.R`. Some of these packages are not part of the base installation of R so they must be installed manually. The `ggplot2` is such a package. It can be installed with the `install.packages("ggplot2", dependencies = TRUE)` command.

```
library(lattice)     #Multipanel graphs
library(mgcv)        #Smoothing
library(ggplot2)     #Multipanel graphs
library(plyr)        #Data manipulation
library(GGally)      #Multipanel graphs
source("./Data/HighstatLibV13.R") #Our support file
```

2.3 Data exploration

We will apply data exploration following the protocol presented in Zuur et al. (2010). The aim of data exploration is to detect potential outliers, determine collinearity (i.e. relationships between covariates), identify potential pseudo-replication problems (e.g. spatial correlation, temporal correlation, repeated measurements from the same station), determine the number of zeros in the response variable and try to identify which follow-up analysis should be applied (i.e. what type of models should be applied). Data exploration should not be used to formulate hypotheses, but merely to check the quality of the data, get a feeling for the data and determine the appropriate strategy for the data analysis.

2.3.1 Missing values

We use the following command to determine if there are missing values in the data.

```
colSums(is.na(RK))
```

```
##   Ring    Age   d13C   d15N   Year   Wing   Bill Tarsus
##      0      0      0      0      0     34      0      0
```

Note that the covariate `Wing` has 34 missing values. We can do one of two things. The first option is to remove the 34 missing values from the variable `Wing`, but this will also delete 34 rows from the data set. The second option is to wait until we investigate which covariates are collinear with one another. Perhaps `Wing` is collinear with another covariate that does not have any missing values. This approach may avoid deleting 34 observations from the object `RK`. We will go with the second option as it is likely that collinearity will be an issue here.

2.3.2 Outliers

We first focus on the presence of extremely large and extremely small values as such observations may dominate the statistical analyses. Common tools for this are boxplots and Cleveland dotplots. The advantage of boxplots is that most people know how to interpret them, but at times they can also be misleading. Cleveland dotplots are less known, but, in our opinion, more useful. In this section, we will use multi-panel Cleveland dotplots; see Figure 2.1. In a Cleveland dotplot the value of a variable is plotted along the x-axis, and the sequence number (i.e. the row number as imported from the data file) is plotted along the y-axis. The main purpose of this graph is to see whether there are any points sticking out on the right-hand side (these are extremely large values) or on the left-hand side (these are extremely small values). 'Sticking out' is defined as 'being different from the majority of the observations'. If such points are present, then we first need to check whether these values are typing errors. If this is not the case, then a data transformation may be considered (though this should be avoided if possible).

The dotplots in Figure 2.1 indicate that there is one bird with a relatively large d15N value. It is an option to remove this bird from the analysis, but we decided to retain it.

We used the function `MyDotplot` from our support file `HighstatLibV13.R` to create the Cleveland dotplots.

```
MyVar <- c("d15N", "Wing", "Bill", "Tarsus", "Year")
Mydotplot(RK[,MyVar])
```

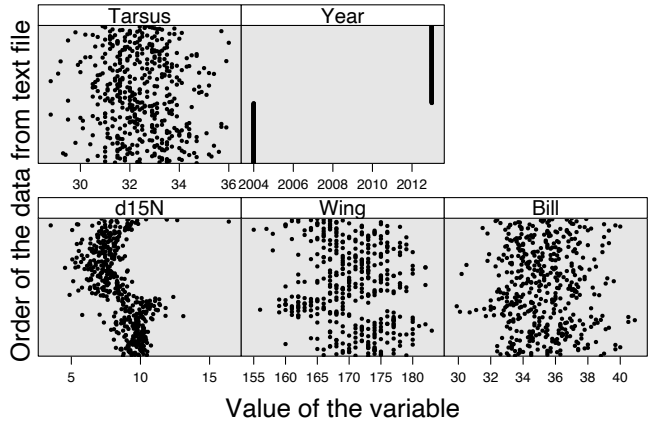

FIGURE 2.1: Cleveland dotplots of all continuous covariates and the response variable. The x-axis shows the value of a variable, and the y-axis shows the row number from the data file. Of prime interest is whether there are any odd data values.

2.3.3 Categorical covariates

There are two categorical covariates in the data set. In this subsection we briefly discuss both of them. We will start with `Year`. We have data from 2 years. The numbers below show the sampled number of red knots in each year and indicate that the data are reasonably well balanced with respect to this covariate.

```
RK$fYear <- factor(RK$Year)
table(RK$fYear)
```

```
##
## 2004 2013
##  207  267
```

The covariate `Age` also has 2 levels: adult and juvenile. The numbers below indicate that we have more adults than juveniles. We decided to set the juveniles as the baseline level as it produces results that are easier to interpret from a biological point of view.

```
RK$fAge <- factor(RK$Age, levels = c("juvenile", "adult" ))
table(RK$fAge)
```

```
##
## juvenile    adult
##      112      362
```

Applying the `table` function on both categorical covariates shows how many observations per age and year combination we have. Note that the data are slightly unbalanced with this respect. This is important information because we will use an interaction between age and year in the model. A rule of thumb is to have about 15–20 observations per regression parameter. The smallest group contains 42 observations, hence we can still apply a model with an interaction term between `fAge` and `fYear`.

```
table(RK$fAge, RK$fYear)
```

```
##
##            2004 2013
##   juvenile   70   42
##   adult     137  225
```

Finally, we want to know how many observations we have per bird. If we have more than one, then this indicates pseudoreplication, and a mixed-effects modelling approach would be the starting point. The numerical output of `table(RK$Ring)` is not presented here, but it shows that every observation is from a different bird. Formulated differently, we indeed have one observation per bird.

2.3.4 Collinearity

Collinearity is defined as relationships between covariates. We deliberately did not use the phrase 'correlation between covariates' as any type of relationship (also non-linear) is collinearity (and this will not be picked up by a correlation coefficient). In a linear regression model, collinearity increases the standard errors of the regression parameters, and this inflates p-values. Formulated differently, correlation between covariates makes the parameters less significant. There are various tools to identify collinearity, e.g. Pearson correlation coefficients, scatterplots, variance inflation factors and principal component analysis biplots. See Zuur et al. (2010) for a detailed discussion and examples.

To investigate collinearity between the 3 continuous and 2 categorical covariates, we plot multipanel scatterplots and conditional boxplots using the `ggpairs` from the `GGally` package (Schloerke et al., 2021); see Figure 2.2. It shows scatterplots, Pearson correlation coefficients, conditional boxplots, histograms, density curves and even the information from the `table` function applied on 2 factors. That is a lot of information. The `ggpairs` help file shows how to omit some of these graphs. For a small set of covariates, like here, this function produces an interesting graph. For a larger number of covariates it may not be that useful.

The scatterplots and correlation coefficients indicate that the covariates `Wing`, `Bill` and `Tarsus` are slightly collinear. We do not feel comfortable using these 3 morphometric variables in the same model. Note that `Age` is also collinear with `Wing`. The paper uses `Bill` as the main morphometric variable.

```
MyVar2 <- c("Wing", "Bill", "Tarsus", "Age", "fYear")
ggpairs(RK[,MyVar2])
```

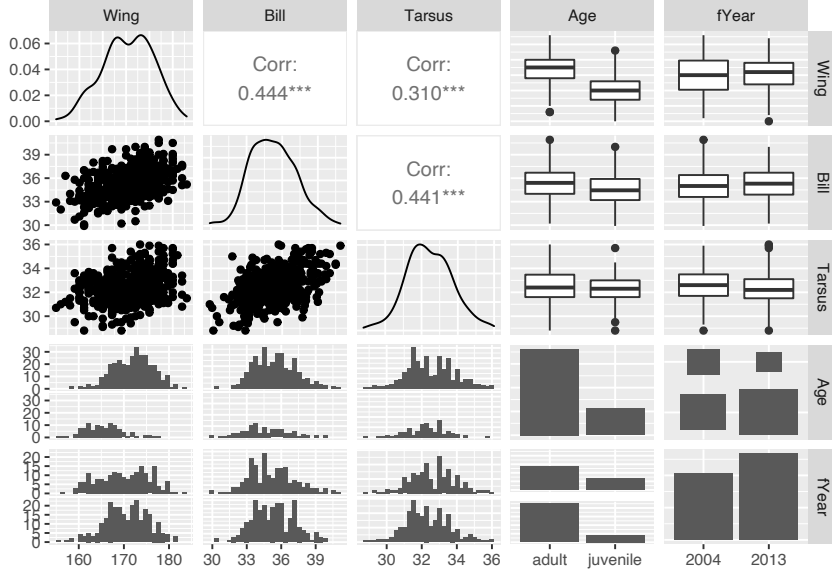

FIGURE 2.2: Pairplot and conditional boxplots of the covariates.

2.3.5 Relationships

In this subsection we will plot the response variable `d15N` versus each covariate and add a linear regression line. We will use the `ggplot2` function (Wickham

et al., 2021) to plot the multipanel scatterplots; see Figure 2.3. In 2013, the adults seem to have lower d15N values, and the slope for the adults seems to be more negative. We will need a statistical model to see whether these differences are important.

```
p <- ggplot(data = RK, aes(y = d15N, x = Bill))
p <- p + geom_point()
p <- p + geom_smooth(method = "lm", se = FALSE,
                     aes(group = Age, col = Age))
p <- p + xlab("Bill length") + ylab("d15N")
p <- p + theme(text = element_text(size=15))
p <- p + facet_grid(.~fYear)
p
```

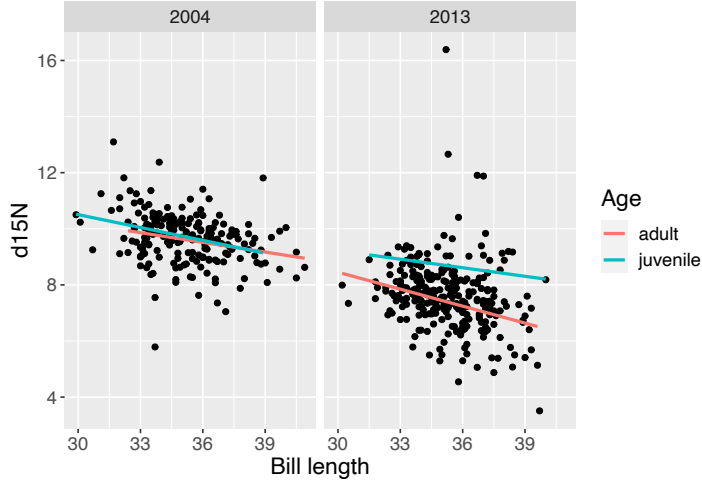

FIGURE 2.3: Multiple scatterplots showing the relationship between d15N and bill length for each Year and Age level.

To better visualise the interaction between the factors we execute the following `ggplot` code. The graph in Figure 2.4 emphasises that `d15N` for adults in 2013 is lower.

```
p <- ggplot(data = RK, aes(y = d15N, x = fAge))
p <- p + geom_boxplot()
p <- p + xlab("Age") + ylab("d15N")
p <- p + theme(text = element_text(size=15))
```

```
p <- p + facet_grid(.~fYear)
p
```

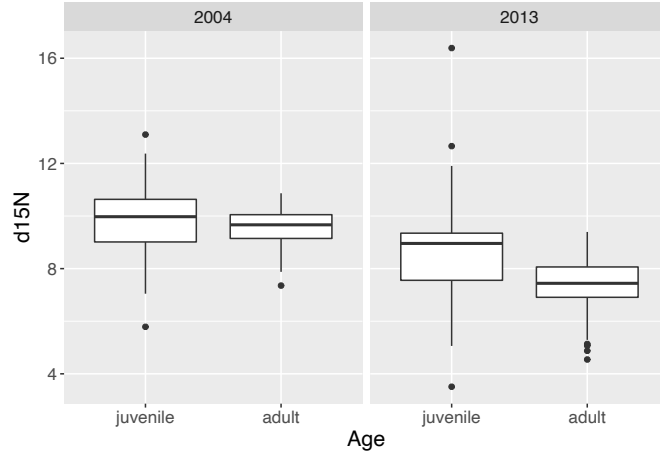

FIGURE 2.4: Boxplot of d15N conditional on Year and Age.

2.4 Model formulation

The response variable `d15N` is defined on the continuous scale, and we will therefore start with a normal distribution. We write this distributional assumption as follows.

$$\text{d15N}_i \sim \text{Normal}(\mu_i, \sigma^2) \tag{2.1}$$

The d15N_i is the isotope composition value from bird i, and we assume that it is normally distributed with the mean μ_i and variance σ^2. The mean μ_i is modelled as a function of covariates. We write this as follows.

$$\mu_i = \text{Covariates}_i \tag{2.2}$$

For the 'Covariates' part, we will use all main terms, 2-way interactions and the 3-way interaction. We actually should have formulated the underlying biological question first, before showing this model. Let us do that now. We want to know whether the relationship between `d15N` and `Bill` size differs

between the 2 years and 2 age classes. This question guides us to the model with the 3-way interaction term. Because the 3-way interaction is included, we must include all 2-way interactions and also the main terms.

2.5 Applying the multiple linear regression model

2.5.1 Executing the model in R

We apply the multiple linear regression model in R with the `lm` function. The notation `Bill * fYear * fAge` in the `lm` function below means that all main terms, all 2-way interactions and the 3-way interaction are included.

```
M1 <- lm(d15N ~ Bill * fYear * fAge, data = RK)
options(width = 70, show.signif.stars = FALSE)
print(summary(M1), digits = 2)
```

```
##
## Call:
## lm(formula = d15N ~ Bill * fYear * fAge, data = RK)
##
## Residuals:
##    Min     1Q Median     3Q    Max
##  -4.73  -0.54   0.07   0.55   7.69
##
## Coefficients:
##                           Estimate Std. Error t value Pr(>|t|)
## (Intercept)                 15.090      2.315     6.5    2e-10
## Bill                        -0.153      0.068    -2.3     0.02
## fYear2013                   -2.819      3.544    -0.8     0.43
## fAgeadult                   -1.426      2.828    -0.5     0.61
## Bill:fYear2013               0.051      0.102     0.5     0.61
## Bill:fAgeadult               0.038      0.082     0.5     0.65
## fYear2013:fAgeadult          3.657      4.139     0.9     0.38
## Bill:fYear2013:fAgeadult    -0.138      0.118    -1.2     0.24
##
## Residual standard error: 1 on 466 degrees of freedom
## Multiple R-squared:  0.55,   Adjusted R-squared:  0.55
## F-statistic:  82 on 7 and 466 DF,  p-value: <2e-16
```

Note that the 3-way interaction is not significant at the 5% level. Dropping covariates or interaction terms because they are not significant is a controversial topic. The main options to proceed are as follows.

1. Leave the model as it is. We have answered the biological question.
2. Apply model selection using the Akaike Information criterion (AIC) or any of its cousins.
3. Define 10–15 models *a priori* and apply the information-theoretic approach.
4. Use p-values to drop non-significant terms (one at a time).
5. Only apply model selection on the interaction terms, but keep all main terms in the model (even if they are not significant).

Other options exists, e.g. cross-validation, least absolute shrinkage and selection operator (LASSO), ridge regression, etc. We will not discuss the pros and cons of these approaches in detail as many books and scientific publications cover these topics in great detail; see, for example, Burnham and Anderson (2002) or Harrison et al. (2018), and a very large number of publications in between. All that we mention here is that option 1 is a sensible approach, provided that there is no strong collinearity between the covariates and if the model does not contain too many interactions (otherwise consider option 5). Option 2 is heavily criticised by many scientists, yet the majority of scientific publications use it. Option 3 is the 'must-do' approach required by many journals. Option 4 is deemed by many to be the worst possible thing that you can do. We will keep it simple here, and apply backwards selection in a moment. But before applying backwards selection, we first investigate whether the model complies with all assumptions. This process is called model validation.

2.5.2 Model validation

As a first step in this process we obtain the (standardised) residuals and fitted values.

```
E1 <- rstandard(M1)
F1 <- fitted(M1)
```

One of the underlying assumptions of the linear regression model is homogeneity of variance. We can verify this by plotting the residuals versus the fitted values; see Figure 2.5. Homogeneity of variance implies that the vertical variation in the residuals should be similar along the horizontal axis. In this case there is no strong reason to question this assumption.

```
plot(x = F1, y = E1, xlab = "Fitted values", ylab = "Residuals")
abline(h = 0, lty = 2)
```

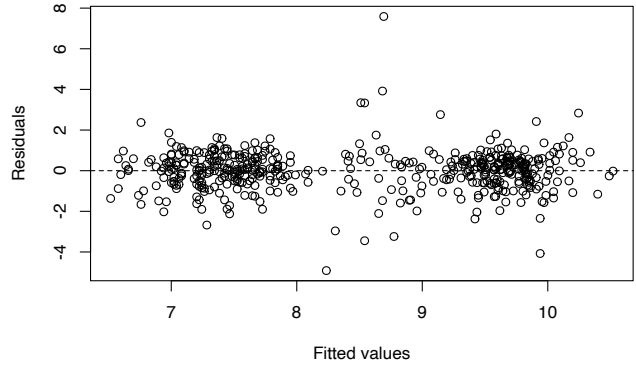

FIGURE 2.5: Residuals plotted versus fitted values for the multiple linear regression model.

Residuals are assumed to be normal, identical and independently distributed with mean 0 and variance σ^2. In simple language this means that the residuals should not contain any information. One way to verify this is to plot the residuals versus each covariate in the model, and also versus each covariate not in the model. If any of these graphs show a clear non-linear (or linear) pattern, then we are violating the independence assumption. If this happens, then we need to improve the model. If we do not do this, then the estimated parameters may be biased. If the graph with residuals versus a covariate in the model shows a non-linear pattern, then extending the model by allowing for non-linear covariate effects is an option. If the graph with residuals versus a covariate that is not in the model shows a pattern, then that covariate should be included in the model.

The graphs in Figure 2.6 do not indicate any serious problems. To aid the visual interpretation, we also added a smoothing curve. Note that 0 is always within the 95% confidence intervals of these smoothers, indicating that there are no strong residual patterns present. If you are not sure, or do not feel confident about visually judging these graphs, then it is also an option to model the residuals with a generalised additive model (GAM) in which one of these covariates is used as a smoother. If the GAM indicates that the smoother is non-linear, then it is time to improve the multiple linear regression model.

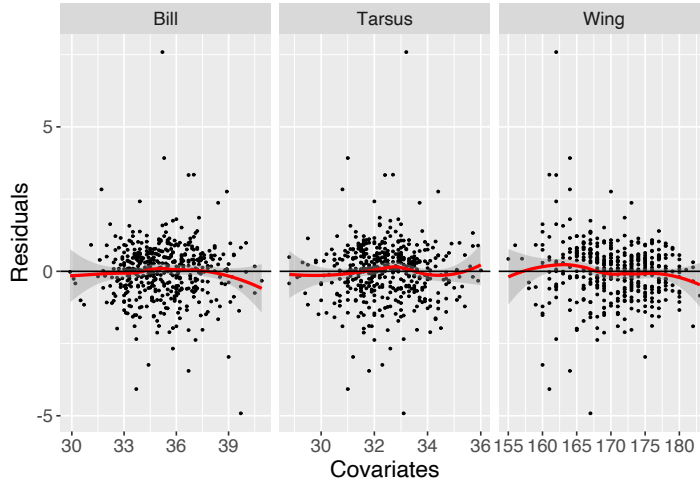

FIGURE 2.6: Multipanel scatterplots showing residuals versus all the continuous covariates that are in, and not in, the model. A LOESS smoother was added to aid visual interpretation.

To create this graph we first put the standardised residuals inside the `RK` data frame, and we then use our `MyMultipanel.ggp2` function, which is in our support file.

```
MyVar <- c("Bill", "Tarsus", "Wing")
RK$E1 <- E1
MyMultipanel.ggp2(Z = RK,
                  varx = MyVar,
                  vary = "E1",
                  ylab = "Residuals",
                  addSmoother = TRUE,
                  addRegressionLine = FALSE,
                  addHorizontalLine = TRUE)
```

We also need to plot the residuals versus the categorical covariates, age and year. These graphs are not presented here, but there are no clear patterns (e.g. heterogeneity).

2.5.3 Model selection

As discussed in Subsection 2.5.1 we will apply classical backwards model selection using the AIC. The advantage of a backwards selection, as opposed

to a forward selection, is that we start with a saturated model and gradually decrease the complexity. This procedure is implemented in R in the `step` function, and it is executed with the following function.

```
step(M1)
```

The full results are not shown here, but they indicate that a model with all main terms and the interaction between age and year is the best model, as judged by the AIC. We execute this model below.

```
M2 <- lm(d15N ~ Bill + fYear + fAge + fYear:fAge, data = RK)
```

At this stage model validation should be applied again. Because only non-important components were removed from the model, we expect to find similar results as in Subsection 2.5.2. We leave it as an exercise for the reader to verify this.

2.6 Model interpretation

In Subsection 2.5.2 we applied model validation on the full model. We did not find any clear reason to reject the model. In Subsection 2.5.3 classical model selection was applied, and we ended up with a model that contains all main terms and an interaction between age and year. In this subsection we will provide an interpretation of this model.

We start with the numerical output.

```
print(summary(M2), digits = 2)
```

```
##
## Call:
## lm(formula = d15N ~ Bill + fYear + fAge + fYear:fAge, data = RK)
##
## Residuals:
##    Min     1Q Median     3Q    Max
##  -4.50  -0.54   0.08   0.55   7.68
##
```

```
## Coefficients:
##                      Estimate Std. Error t value Pr(>|t|)
## (Intercept)            15.191      0.882    17.2   <2e-16
## Bill                   -0.156      0.026    -6.1    2e-09
## fYear2013              -1.001      0.203    -4.9    1e-06
## fAgeadult              -0.079      0.156    -0.5      0.6
## fYear2013:fAgeadult    -1.228      0.233    -5.3    2e-07
##
## Residual standard error: 1 on 469 degrees of freedom
## Multiple R-squared:  0.55,   Adjusted R-squared:  0.55
## F-statistic: 1.4e+02 on 4 and 469 DF,  p-value: <2e-16
```

The results indicate that we have a negative bill length effect, and a strong interaction between age and year. Both are significant at the 5% level. The model explains 55% of the total variation in `d15N`. The data are from 2 years and 2 age classes. This means that we can write 4 equations for the expected values μ_i; see Equation (2.3).

$$\begin{aligned}
\mu_i &= 15.19 - 0.16 \times \text{Bill}_i && \text{for 2004 and juvenile} \\
\mu_i &= 15.19 - 0.16 \times \text{Bill}_i - 0.08 && \text{for 2004 and adult} \\
\mu_i &= 15.19 - 0.16 \times \text{Bill}_i - 1.00 && \text{for 2013 and juvenile} \\
\mu_i &= 15.19 - 0.16 \times \text{Bill}_i - 1.00 - 0.08 - 1.22 && \text{for 2013 and adult}
\end{aligned} \tag{2.3}$$

In addition to presenting a table with the estimated values, standard errors, t-values and p-values in a paper or report, we strongly suggest also including a graphical presentation of the fitted model. We will now show how to do this.

We need to create covariate values for which we can sketch the fitted values. The model contains bill length, year and age; hence we need to specify a data frame that contains bill length, year and age values. We will call this data frame `MyData`. We will use the `predict` function to obtain predicted values and standard errors for these covariate values. To avoid doing predictions for bill length values that are not observed in a certain year and age level, we use the `ddply` function from the `plyr` package (Wickham, 2020). It takes the data in `RK`, which is then broken up into pieces according to the levels of `fYear` and `fAge`. In each piece the `summarise` function from the same package is applied; it takes the minimum and maximum value of `Bill` in each piece and creates 25 values between these 2 extremes.

```
MyData <- ddply(RK,
                .(fYear, fAge), summarise,
                 Bill = seq(from = min(Bill),
```

```
                              to = max(Bill),
                              length = 25))
```

The same results can be obtained with the `dplyr` package, which is the successor of the `plyr` package. However, this requires the use of 'pipes', and we find these slightly confusing to work with. For details, see `https://blog.rstudio.com/2014/01/17/introducing-dplyr/`.

We now have 100 ($= 25 \times 2 \times 2$) artificial covariate values for which we want to make predictions. We can use the `predict` function for this.

```
P2 <- predict(M2, newdata = MyData, se = TRUE)
```

The object `P2` contains predicted d15N values for the covariate values in `MyData`. For plotting purposes, it is easier to have these predicted values inside the `MyData` object. The R code below does that. We also calculate the 95% confidence intervals for the fitted values and add these to `MyData`.

```
MyData$mu    <- P2$fit                       #Predicted values
MyData$selow <- P2$fit - 1.96 * P2$se.fit    #Lower bound
MyData$seup  <- P2$fit + 1.96 * P2$se.fit    #Upper bound
head(MyData)
```

```
##   fYear     fAge   Bill       mu     selow     seup
## 1  2004 juvenile 29.900 10.53261 10.212053 10.85317
## 2  2004 juvenile 30.275 10.47419 10.165759 10.78262
## 3  2004 juvenile 30.650 10.41577 10.118768 10.71277
## 4  2004 juvenile 31.025 10.35735 10.070996 10.64370
## 5  2004 juvenile 31.400 10.29893 10.022352 10.57551
## 6  2004 juvenile 31.775 10.24051  9.972741 10.50828
```

The `MyData` data frame now contains a set of covariate values for bill length, year and age, and matching predicted values and the boundaries of the 95% confidence intervals. We make a `ggplot2` graph with the observed bill length, year, age and d15N data, and superimpose the predicted values; see Figure 2.7. The R code for this graph is given below. To obtain the 4 lines in one graph we define a new variable `YearAge` that defines the 4 combinations of year and age class, and we use this for colour and groups.

```
RK$YearAge     <- factor(paste(RK$Year, RK$Age, sep = " "))
MyData$YearAge <- factor(paste(MyData$fYear,
                               MyData$fAge, sep = " "))
p <- ggplot()
p <- p + geom_point(data = RK,
                    aes(y = d15N, x = Bill, col = YearAge),
                    shape = 16,
                    size = 1)
p <- p + xlab("Bill length") + ylab("d15N")
p <- p + theme(text = element_text(size=15)) + theme_bw()
p <- p + geom_line(data = MyData,
                   aes(x = Bill , y = mu,
                       group = YearAge, col = YearAge))
p <- p + geom_ribbon(data = MyData,
                     aes(x = Bill,
                         ymax = seup,
                         ymin = selow,
                         group = YearAge,
                         fill = YearAge),
                     alpha = 0.5)
p
```

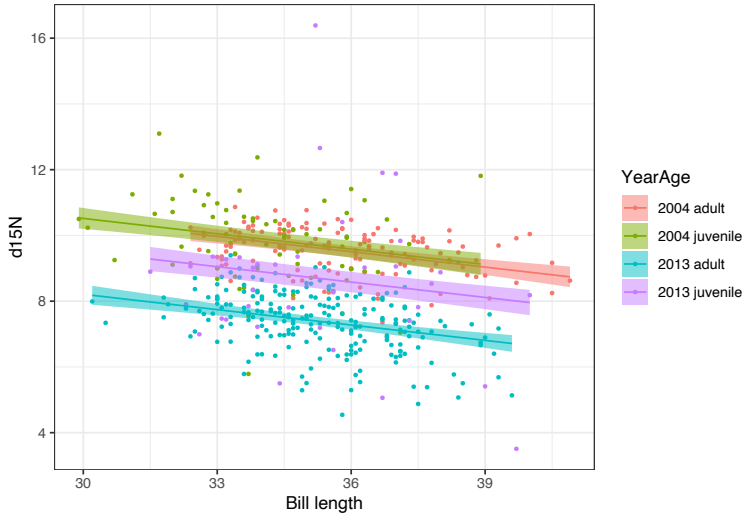

FIGURE 2.7: Results of the linear regression model. The lines represent the fitted values, and the shaded areas around these lines represent 95% confidence intervals.

The predicted values in the figure indicate that `d15N` decreases for larger `Bill` size. Adults in 2013 had the lowest `d15N` values. In 2004, both adults and juveniles had higher `d15N` values than in 2014.