Mixed Effects Models using nlme example - badgers

NES8010 Data Analaysis and Modelling

February 2021

# Introduction

A basic straight line regression is based on the equation:

where is the response variable, the explanatory variable, is the intercept of the fitted regression line, and the gradient of the line.

To extend this to account for a known grouping in the data we can specify a model that allows the slope and intercept of the regression line to vary for different groups.

Mixed models can be random intercepts model, random slope models or random slopes and intercepts models. The syntax for a mixed-effects model is:

A mixed model includes both ‘fixed effects’ and ‘random effects’. Random effects are categories or factors, that we want to control for the effects of ( rather than specifically being interested in their influence).

There are several packages/libraries available in R to model mixed effects (a.k.a random effects multi-level models or heirarchical models). The most common packages include ‘nlme’ and ‘lme4’. Different researchers prefer using different packages, as the packages have different features and some feature analyses not availble in others. Packages will achieve the same (or very similar) results and the most notable differences are in the syntax of the model specification.

# Badger Characteristics

## Dataset

The dataset are of badger characteristics, measured at postmotem. The badgers were collected from across SW England over a number of years. We are interested to determine if we can describe the weight of badgers in relation to the other covariates provided. The variables include Area and Group (Codes to determine where the animal came from), Year (Year of Post mortem), Age (adult/juvenile) , Weight (g), Length (cm) Sex (M/F) of the individual a variable called tooth to be used as a proxy for Age.

We are aware that individuals can only belong to one Area and Group and want to include these data structures in our models

# Analysis

First, read in the data. Ensure you are either using RProj or have the correct filepath to where your data are stored using the Session> set working >directory> choose working directory from the Rstudio menus.

The code in these notes uses the here package and the data are in the *Data* directory. The Code first checks that the packages we need for the exercise are loaded to our workspace.

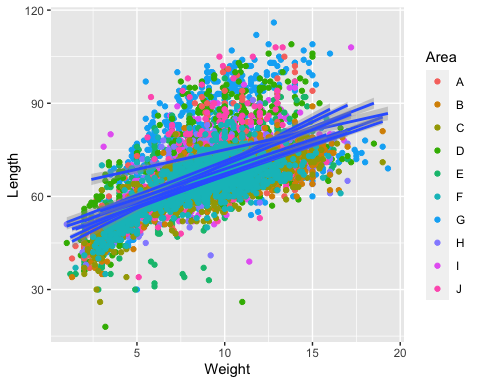
wants <- c("ggfortify", "here", "nlme", "lattice","ggplot2")  
has <- wants %in% rownames(installed.packages())  
if(any(!has)) install.packages(wants[!has])  
  
  
library(here)

badger<- read.csv(here("Data", "bad4.csv"))  
  
library(nlme)  
library(ggplot2)  
library(lattice)

## Warning: package 'lattice' was built under R version 4.0.2

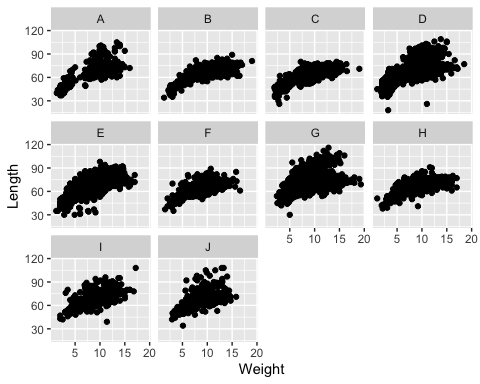
The length-weight relationship of the badgers can be compared between all Areas by plotting the relationship and highlighting the Areas by colour

#ggplot  
  
ggplot(badger, aes( Weight,Length, group= Area)) +   
 geom\_point(aes (colour=Area))+  
 geom\_smooth(method = "lm")



or this can be plotted more clearly using facets (gridded plots).

#ggplot  
  
ggplot(badger, aes( Weight,Length, group= Area)) +   
 geom\_point()+  
 facet\_wrap(.~Area)



We could run separate analyses, and fit a separate model for each Area, but remember badgers also belong to different Groups within each area so that could mean running hundreds of seperate models, each with only a few badgers in it - perhaps too few to have a reliable regression model. It would be better if we can include all of the data in a single model but account for the grouping structures.

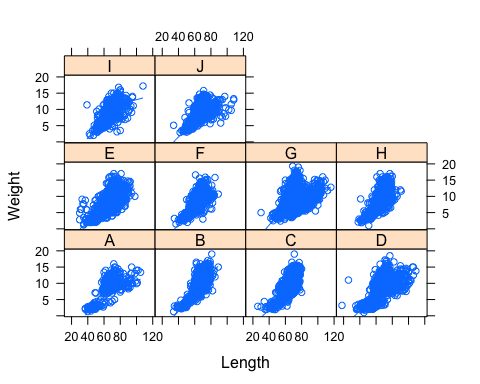
The mixed effect model allows all of the data to be used to model the relationship between length and weight, while accounting for correlations between data coming from the same Area and Group. This allows means the model has to estimate fewer parameters.

Using the nlme package the grouping structure can be specified using the groupedData function.

# group data   
# The model requires grouping to be specified - so set group structure  
badgerG<-groupedData(Weight~Length|Area,badger)   
  
#badgerList<- lmList(Weight~Length|Area,badger)

The nlme package and its grouping structure can be visualised using the package/ library called lattice which was available to visualise data hierarchies before ggplot was developed. Compare the ggplot and grid facets with the xyplot from lattice

xyplot(Weight~Length|Area, data=badger, type=c("p","smooth"))



The mixed model is compiled using the lme function rather than the glm function. As there are some *NA* values in the dataset na.action is set to na.exclude.

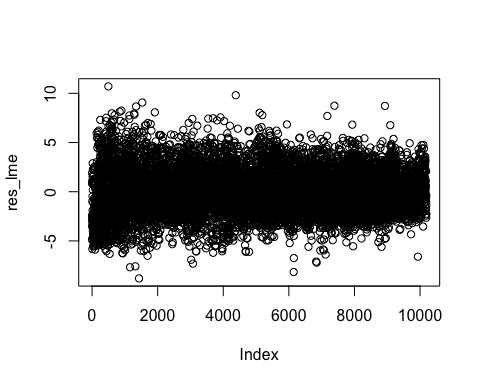
model\_lme<- lme(Weight~Length + Sex , random = ~1|Area, data = badgerG, na.action=na.exclude)  
summary(model\_lme)

## Linear mixed-effects model fit by REML  
## Data: badgerG   
## AIC BIC logLik  
## 45217.44 45253.57 -22603.72  
##   
## Random effects:  
## Formula: ~1 | Area  
## (Intercept) Residual  
## StdDev: 0.5103364 2.22983  
##   
## Fixed effects: Weight ~ Length + Sex   
## Value Std.Error DF t-value p-value  
## (Intercept) -3.927761 0.22554120 10154 -17.41483 0  
## Length 0.191253 0.00231943 10154 82.45674 0  
## SexM 0.355987 0.04461908 10154 7.97836 0  
## Correlation:   
## (Intr) Length  
## Length -0.685   
## SexM -0.026 -0.094  
##   
## Standardized Within-Group Residuals:  
## Min Q1 Med Q3 Max   
## -3.9505022 -0.6856327 -0.0612243 0.6327131 4.7995185   
##   
## Number of Observations: 10166  
## Number of Groups: 10

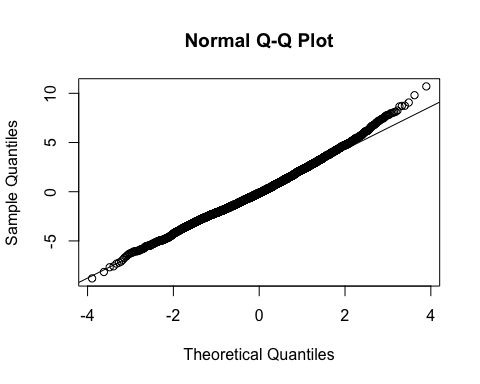
The summary command details the influence of both the fixed and random effects. The variation explained by the random effect can be calculated as *proportion explained = variance/ (variance+residual)*.

The residuals can be plotted in a number of different ways to visualise the model assumptions.

res\_lme=residuals(model\_lme)  
plot(res\_lme)

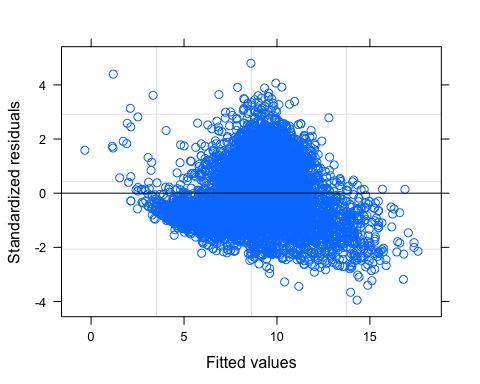


qqnorm(res\_lme)  
qqline(res\_lme)

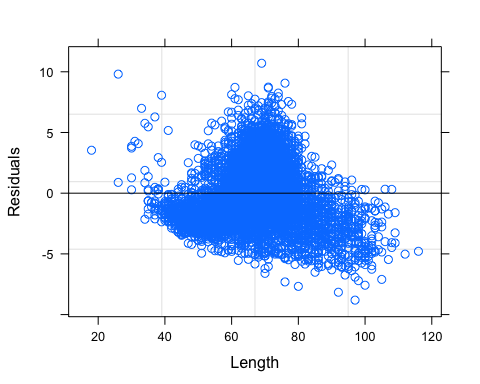


The random effects can also be visualised.

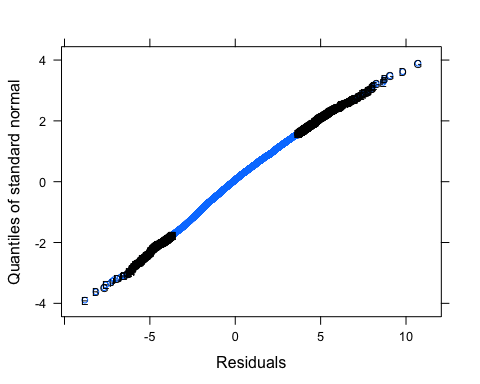
plot(model\_lme)



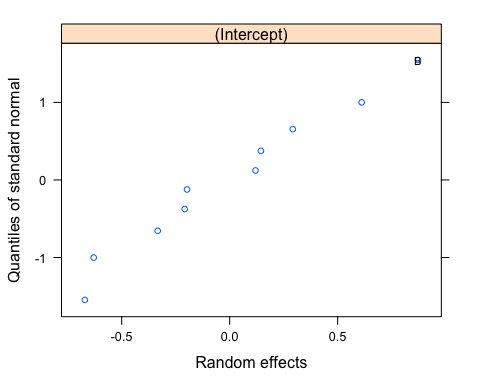
#same as  
plot(model\_lme, resid(.) ~Length, abline =0)



qqnorm(model\_lme, ~resid(.), id=0.10, cex=0.7)



qqnorm(model\_lme, ~ranef(.), id=0.10, cex=0.7)



plot(ranef(model\_lme))

