Session 2: Intro to frequentist models in R

Natalie Z. Kerr & Brian Lovett

October 21, 2020

## Workshop description

Here, we will cover how to run linear models (LMs; e.g., Gaussian/Normal distribution), generalized linear models (GLMs; e.g., binomial, Poisson for count data), and mixed models (LMMs/GLMMs; e.g., fixed vs. random effects). We will also expand on models using Poisson-distributed data by evaluating how to deal with common issues such as when your count data are over-dispersed/under-dispersed (e.g., Poisson vs. Negative Binomial vs. Conway-Maxwell-Poisson), when counts should be represented as rates (e.g., counts per unit time using Poisson offsets), and when your count data are zero-inflated (e.g., 0-inflated regression vs hurdle models). Finally, we will finish the session on appropriate ways to run model selection techniques for finding a winning model from a set of candidate models, using Akaike Information Criteria (AIC) or likelihood ratio tests (LRT) for nested models. This course will not be a statistics course, so people will need to be familiar with most of these models.

## Packages required

Before running the following code, please open the R Project for the EntSoc R Webinar series in the main folder (“EntSoc\_R-Webinar\_2020.Rproj”).

Here, is a list of packages required for this R course. You will need to install these prior to the class, either install from the “packages” panel in R studio or using the function below.

install.package("")

Once installed, we can run these packages in advance. I will inform you whenever we are running a function from each of these packages throughout this session.

library(here) # for navigation among folders  
library(tidyverse) # for all tidyverse packages  
  
library(pscl) # for zero-inflated regression models   
library(glmmTMB) # for both zero-inflated and hurdle models  
library(lme4) # for mixed models  
  
library(boot) # testing assumptions of the Gamma distribution  
library(car) # for likelihood ratio tests/marginal hypothesis testing  
library(lmtest) # for likelihood ratio tests  
library(bbmle) # for AIC

## Workshop topics

Today, we will cover six main topics in this workshop:

1. Linear models
2. Model selection approaches
3. Generalized linear models
4. Common issues with Poisson-distributed data
5. Mixed models
6. Intro to simulating data

We will cover alternative statistical distributions to the Normal and Poisson distributions, when encountering common issues with these data.

For background reading that is required for this course, I would highly recommend reading:

* “Ecological models and data in R” by *Ben Bolker*
* “Model Selection and Inference: A Practical Information-Theoretic Approach” by *Kenneth Burnham* and *David Anderson*
* Some of the first chapters (1-6) of “Program MARK: A Gentle Introduction” by *Gary White* and *Evan Cooch* that provides a good introduction to probability and maximum likelihood estimation.

I have listed some other useful resources at the end of this rmarkdown file. Not always one way of describing something will make sense to you, so often it takes finding the right teacher or resource for these concepts to click.

### 1. Intro to linear models

First, we will cover running linear models (LMs) for data that follows a normal (or Gaussian) distribution. Linear models can be used to carry out:

* Single stratum analysis of variance (i.e., intercept-only models)
* Analysis of variance (ANOVA, i.e., differences among groups)
* Regression
* Analysis of covariance (ANCOVA)

… that we will cover today. For these models, we will use Brown hare (*Lepus europaeus*) data over 17 years (1992-2008) at 56 sites in 8 regions of Switzerland for many of our examples today. These sites vary in area, elevation, and belong to two habitat types (arable and grassland). Mean density is the count1 of hares offset by the area of the site (i.e., mean.density = count1/area). These data are used in the [2010 Marc Kery book](https://www.mbr-pwrc.usgs.gov/software/kerybook/) that contains examples of both R and WinBUGS code.

hares <- read.table(here::here("Session 2", "Data", "hares.txt"), header = T)  
head(hares)

## no site region area elevation landuse year count1 count2 mean.density  
## 1 1 AG01 CH.Central 2.23 384 arable 1992 NA NA NA  
## 2 2 AG01 CH.Central 2.23 384 arable 1993 NA NA NA  
## 3 3 AG01 CH.Central 2.23 384 arable 1994 NA NA NA  
## 4 4 AG01 CH.Central 2.23 384 arable 1995 6 4 2.690583  
## 5 5 AG01 CH.Central 2.23 384 arable 1996 7 5 3.139013  
## 6 6 AG01 CH.Central 2.23 384 arable 1997 3 3 1.345291

#### Intercept-only models

First, we will run a single stratum analysis of variance (aka “intercept-only”) model to estimate the mean density of Brown hares across Switzerland.

dens1 <- lm(mean.density ~ 1, data = hares)  
summary(dens1)

##   
## Call:  
## lm(formula = mean.density ~ 1, data = hares)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -4.6137 -2.7488 -0.9541 1.5672 17.3565   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 4.7366 0.1467 32.29 <2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 3.817 on 676 degrees of freedom  
## (275 observations deleted due to missingness)

Interpreting the summary output:

* *Call:* model formula.
* *Residuals:* difference between the observed response values and model predicted values. Mean should be zero.
* *Coefficients:*
  + Model estimate
  + SE of model estimate
  + t-value (SDs our estimate is from 0)
  + P-value (i.e., probability of observing a value equal or larger than *t*, i.e., is our model estimate is significantly different from 0?)
* *Residual Standard Error:* average amount that the response will deviate from our model estimate.

mean(dens1$residuals) # mean of the residuals is close to zero

## [1] 4.079368e-16

mean(hares$mean.density, na.rm = T) # Mean

## [1] 4.736569

sd(hares$mean.density, na.rm = T)/sqrt(nrow(subset(hares, !is.na(mean.density)))) # SE = sd/sqrt(n)

## [1] 0.1466952

summary(dens1)$sigma / summary(dens1)$coefficient[1] # 80% percentage error

## [1] 0.8058354

#### One-way ANOVA

Next, we will run a one-way analysis of variance (ANOVA) model to estimate mean density of brown hares in the two habitat types: arable vs. grassland.

dens2 <- lm(mean.density ~ landuse, data = hares)  
summary(dens2)

##   
## Call:  
## lm(formula = mean.density ~ landuse, data = hares)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -5.1593 -2.6117 -0.9307 1.7159 16.7613   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 5.3317 0.1672 31.893 < 2e-16 \*\*\*  
## landusegrass -2.1432 0.3172 -6.756 3.07e-11 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 3.697 on 675 degrees of freedom  
## (275 observations deleted due to missingness)  
## Multiple R-squared: 0.06333, Adjusted R-squared: 0.06194   
## F-statistic: 45.64 on 1 and 675 DF, p-value: 3.072e-11

The lm() summary also contains three more outputs:

* *Multiple R-squared:* determines how well your model fits your data (aka the coefficient of determination).
* *Adjusted R-squared:* provides the same information, but adjusts the multiple R-squared value by the number of variables in your model.
* *F-statistic:* checks that at least one of your coefficients in your model is nonzero.

summary(dens2)$r.squared # only 6.3% of landuse explains the variance in mean density

## [1] 0.06333232

TSS <- sum((hares$mean.density-mean(hares$mean.density, na.rm = T))^2, na.rm = T) # total sum squares  
RSS <- sum(dens2$residuals^2) # residual sum squares (difference between y values and the two means)  
(TSS-RSS)/TSS # r-squared

## [1] 0.06333232

For this model, we used what is referred to as “effect parameterization”. The dummy variable (or *Intercept*, ) is arable land, and the *landusegrass* is difference to grassland relative to the dummy variable (or *slope*, ).

coef(dens2) # Intercept (i.e., arable estimate) and slope (i.e., difference between arable and grassland)

## (Intercept) landusegrass   
## 5.331719 -2.143173

The mean density in the arable land is 5.3317186 and the mean density in the grassland is 3.1885457, i.e., where *x* is either “1” for grassland or “0” for arable.

subset(hares, !is.na(mean.density))[70:80,]

## no site region area elevation landuse year count1 count2 mean.density  
## 105 105 BE03 Aare 1.12 557 grass 1994 1 NA 0.8928571  
## 106 106 BE03 Aare 1.12 557 grass 1995 1 1 0.8928571  
## 109 109 BE03 Aare 1.12 557 grass 1998 1 1 0.8928571  
## 111 111 BE03 Aare 1.12 557 grass 2000 2 1 1.7857143  
## 120 120 BE04 Aare 2.77 532 arable 1992 5 7 2.5270758  
## 121 121 BE04 Aare 2.77 532 arable 1993 8 10 3.6101083  
## 122 122 BE04 Aare 2.77 532 arable 1994 1 6 2.1660650  
## 123 123 BE04 Aare 2.77 532 arable 1995 9 8 3.2490975  
## 126 126 BE04 Aare 2.77 532 arable 1998 6 8 2.8880866  
## 128 128 BE04 Aare 2.77 532 arable 2000 21 8 7.5812274  
## 133 133 BE04 Aare 2.77 532 arable 2005 1 2 0.7220217

model.matrix(dens2)[70:80,] # each row is an observation used to find MLE for landuse

## (Intercept) landusegrass  
## 105 1 1  
## 106 1 1  
## 109 1 1  
## 111 1 1  
## 120 1 0  
## 121 1 0  
## 122 1 0  
## 123 1 0  
## 126 1 0  
## 128 1 0  
## 133 1 0

effects\_par <- lm(mean.density ~ 1 + landuse, data = hares)   
summary(effects\_par)$coefficients # same model as dens2, formula assumes "~ 1 +"

## Estimate Std. Error t value Pr(>|t|)  
## (Intercept) 5.331719 0.1671745 31.893138 7.712778e-137  
## landusegrass -2.143173 0.3172381 -6.755723 3.071940e-11

However, we can also use the “means parameterization” approach for our model structure, where each group is *not* in reference to the “dummy” variable.

means\_par <- lm(mean.density ~ -1 + landuse, data = hares) # removes dummy variable  
summary(means\_par)$coefficients

## Estimate Std. Error t value Pr(>|t|)  
## landusearable 5.331719 0.1671745 31.89314 7.712778e-137  
## landusegrass 3.188546 0.2696159 11.82625 1.846049e-29

model.matrix(effects\_par)[70:80,] # each row is an observation used to find MLE

## (Intercept) landusegrass  
## 105 1 1  
## 106 1 1  
## 109 1 1  
## 111 1 1  
## 120 1 0  
## 121 1 0  
## 122 1 0  
## 123 1 0  
## 126 1 0  
## 128 1 0  
## 133 1 0

model.matrix(means\_par)[70:80, ]

## landusearable landusegrass  
## 105 0 1  
## 106 0 1  
## 109 0 1  
## 111 0 1  
## 120 1 0  
## 121 1 0  
## 122 1 0  
## 123 1 0  
## 126 1 0  
## 128 1 0  
## 133 1 0

**Practice example 1:** We will be using the built-in “ToothGrowth” R dataset. These data from an experiment studying the effect of vitamin C on tooth growth in 60 Guinea pigs. Each animal received one of three dose levels of vitamin C (0.5, 1, and 2 mg/day) by one of two delivery methods: orange juice (OJ) or ascorbic acid (VC).

?ToothGrowth

attach(ToothGrowth)   
head(ToothGrowth)

## len supp dose  
## 1 4.2 VC 0.5  
## 2 11.5 VC 0.5  
## 3 7.3 VC 0.5  
## 4 5.8 VC 0.5  
## 5 6.4 VC 0.5  
## 6 10.0 VC 0.5

With these data, run an ANOVA to estimate tooth growth for the three dosages of vitamin C. Run two models using the effects and means parameterization approach.

*Hint:* you may want to use str() to look at the structure of the dataset.

#### Two-way ANOVA

Here, we will run a two-way ANOVA *without an interactive effect* of both landuse and region on mean density of Brown hares.

dens4 <- lm(mean.density ~ region + landuse, data = hares)  
coef(dens4)

## (Intercept) regionBaselland regionCH.Central regionCH.E   
## 4.9529579 -1.4585568 -1.9953734 -0.5712674   
## regionCH.N regionCH.SW regionCH.W regionRhone   
## 0.3031590 4.6676222 0.1660742 -1.1703031   
## landusegrass   
## -0.8073997

dens5 <- lm(mean.density ~ -1 + region + landuse, data = hares)  
coef(dens5) # mildly more comprehensible

## regionAare regionBaselland regionCH.Central regionCH.E   
## 4.9529579 3.4944011 2.9575845 4.3816905   
## regionCH.N regionCH.SW regionCH.W regionRhone   
## 5.2561169 9.6205801 5.1190322 3.7826548   
## landusegrass   
## -0.8073997

By removing the dummy variable for region (“dens5”), we can see that each region has an estimate for mean density of brown hares in arable land. To obtain a grassland estimate for each region, you will still need to add landusegrass: rcoef(dens5)[9] to each region estimate.

coef(dens5)[1:8] # estimate for arable land for all regions

## regionAare regionBaselland regionCH.Central regionCH.E   
## 4.952958 3.494401 2.957585 4.381691   
## regionCH.N regionCH.SW regionCH.W regionRhone   
## 5.256117 9.620580 5.119032 3.782655

coef(dens5)[1:8] + coef(dens5)[9] # estimate for grassland for all regions

## regionAare regionBaselland regionCH.Central regionCH.E   
## 4.145558 2.687001 2.150185 3.574291   
## regionCH.N regionCH.SW regionCH.W regionRhone   
## 4.448717 8.813180 4.311633 2.975255

Next, we will run a two-way ANOVA *with interactive effects* of both vitamin C dosage and supplement type on tooth growth in guinea pigs.

grow1 <- lm(len ~ supp + factor(dose) + supp:factor(dose), data = ToothGrowth)  
coef(grow1)

## (Intercept) suppVC factor(dose)1   
## 13.23 -5.25 9.47   
## factor(dose)2 suppVC:factor(dose)1 suppVC:factor(dose)2   
## 12.83 -0.68 5.33

grow2 <- lm(len ~ supp \* factor(dose), data = ToothGrowth) # same model as grow1   
coef(grow2)

## (Intercept) suppVC factor(dose)1   
## 13.23 -5.25 9.47   
## factor(dose)2 suppVC:factor(dose)1 suppVC:factor(dose)2   
## 12.83 -0.68 5.33

**Practice example 2:** For this practice example, we will be using the data are from [Lampe et al. 2013](https://doi.org/10.1111/1365-2435.12215). Grasshopper nymphs were relocated from seven roadside (n = 112) and five non-roadside (n = 65) habitats (i.e., “Origin”) to a laboratory. Half from each habitat were reared in either noisy or quiet conditions (i.e., “Treatment”) for a full two-by-two factorial design. Then, the adult males were weighed (i.e., “BodyMass”) and their courtship songs (i.e., “LocMax”) recorded.

ghp <- read.csv(here::here("Session 2", "Data", "GrasshopperSong.csv"))  
head(ghp)

## Site Individual Origin Treatment LocMax BodyMass  
## 1 Asch AS\_lei\_la\_0001 non-roadside noisy 7193 0.084  
## 2 Asch AS\_lei\_la\_0001 non-roadside noisy 7239 0.084  
## 3 Asch AS\_lei\_la\_0001 non-roadside noisy 7358 0.084  
## 4 Asch AS\_Lei\_la\_0002 non-roadside noisy 8958 0.086  
## 5 Asch AS\_Lei\_la\_0002 non-roadside noisy 8452 0.086  
## 6 Asch AS\_Lei\_la\_0002 non-roadside noisy 8958 0.086

Using the grasshopper song data, run a two-way ANOVA looking at the effects of origin population (i.e., non-roadside or roadside) and rearing treatment (i.e., noisy vs quiet) on male Bow-winged grasshopper song.

#### Simple linear regression

A simple linear regression is an approach for modelling the relationship between a single continuous predictor variable on a continuous response variable , as follows:

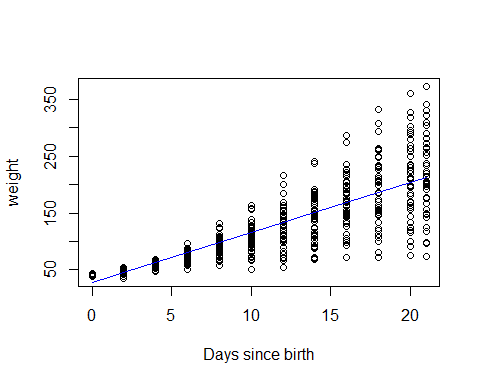
where the model describes a line with some intercept and y-intercept $\_{1}, such that the linear function provides a “best-fitting” line between the two variables.

Here, we will use the built-in dataset from an experiment evaluating the effects of diet on early growth of chicks to run a simple linear regression. Chicks were given one of four diets and weighed each day since birth (“Time”) for approximately 21 days.

cw1 <- lm(weight ~ Time, data = ChickWeight)  
summary(cw1)

##   
## Call:  
## lm(formula = weight ~ Time, data = ChickWeight)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -138.331 -14.536 0.926 13.533 160.669   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 27.4674 3.0365 9.046 <2e-16 \*\*\*  
## Time 8.8030 0.2397 36.725 <2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 38.91 on 576 degrees of freedom  
## Multiple R-squared: 0.7007, Adjusted R-squared: 0.7002   
## F-statistic: 1349 on 1 and 576 DF, p-value: < 2.2e-16

pred.lin <- coef(cw1)[1] + coef(cw1)[2]\*0:21  
with(ChickWeight, plot(Time, weight, xlab = "Days since birth"))  
points(0:21, pred.lin, type = "l", col = "blue")

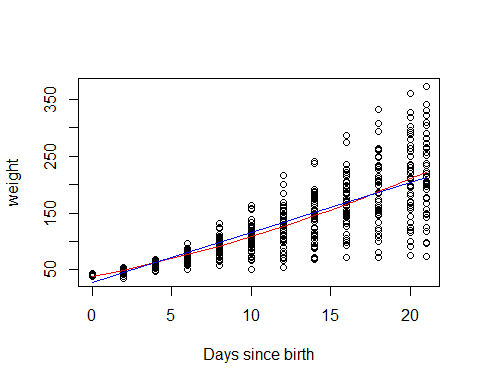


We can also run *n*-degree polynomial linear functions. I have choosen to only run a second-degree polynomial relationship (e.g., quadratic function) to evaluate whether there is an intermediate elevation that has the highest mean density of brown hares.

cw2 <- lm(weight ~ Time + I(Time\*Time), data = ChickWeight)  
summary(cw2)

##   
## Call:  
## lm(formula = weight ~ Time + I(Time \* Time), data = ChickWeight)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -147.952 -12.507 0.518 11.126 151.048   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 38.13394 4.08288 9.340 < 2e-16 \*\*\*  
## Time 5.45963 0.89962 6.069 2.34e-09 \*\*\*  
## I(Time \* Time) 0.15684 0.04071 3.852 0.00013 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 38.45 on 575 degrees of freedom  
## Multiple R-squared: 0.7083, Adjusted R-squared: 0.7073   
## F-statistic: 698 on 2 and 575 DF, p-value: < 2.2e-16

pred.quad <- coef(cw2)[1] + coef(cw2)[2]\*0:21 + coef(cw2)[3]\*0:21\*0:21  
with(ChickWeight, plot(Time, weight, xlab = "Days since birth"))  
points(0:21, pred.quad, type = "l", col = "red") # quadratic function  
points(0:21, pred.lin, type = "l", col = "blue") # linear function



Note that we need to use the function I() for the quadratic term, which treats the variable “as is” rather than an interaction between two variables (as seen in the two-way ANOVA with an interaction).

**Practice example 3:** Using the built-in “ToothGrowth” R dataset, run linear regression to determine whether tooth growth changes linearly with vitamin C dosage. Then, plot the raw data and model predicted values.

#### ANCOVA

An analysis of covariance (ANCOVA) combines both categorical (e.g., ANOVA) and continuous (e.g., linear regression) covariates to predict some response variable.

Using the ChickWeight dataset, we will run an ANCOVA to evaluate the interactive effects of diet and time on chick weights.

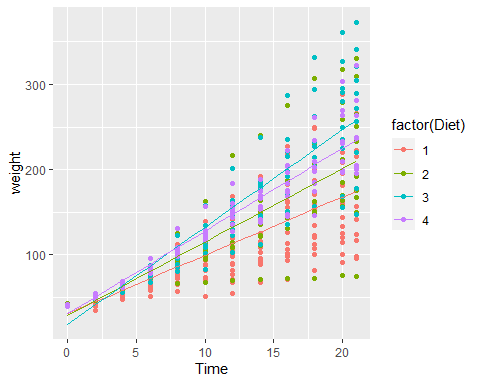
cw3 <- lm(weight ~ factor(Diet)\*Time, data = ChickWeight)  
coef(cw3)

## (Intercept) factor(Diet)2 factor(Diet)3 factor(Diet)4   
## 30.9309803 -2.2973848 -12.6806551 -0.1388608   
## Time factor(Diet)2:Time factor(Diet)3:Time factor(Diet)4:Time   
## 6.8417972 1.7673391 4.5810738 2.8725684

chick.dat <- data.frame(expand.grid(Time = 0:21, Diet = factor(c(1:4))))  
chick.dat$pred <- predict(cw3, newdata = chick.dat, type = "response") # use the predict() function  
head(chick.dat)

## Time Diet pred  
## 1 0 1 30.93098  
## 2 1 1 37.77278  
## 3 2 1 44.61457  
## 4 3 1 51.45637  
## 5 4 1 58.29817  
## 6 5 1 65.13997

ggplot(ChickWeight, aes(x = Time, y = weight, col = factor(Diet))) + geom\_point() +  
 geom\_line(chick.dat, mapping = aes(x = Time, y = pred, col = factor(Diet)))



**Practice example 4:** Using the built-in data on ToothGrowth, run an ANCOVA to evaluate the interactive effects of supplement type and dose on chick weights. Then, plot these data.

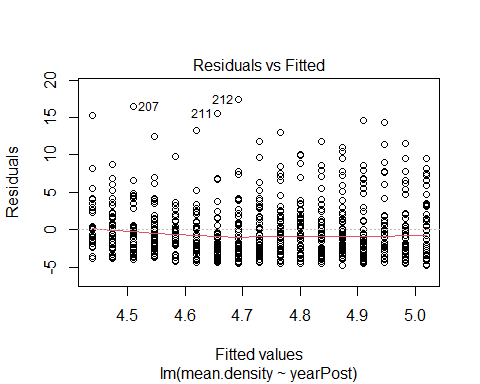
#### Assumptions of normality

Before continuing with this statistical distribution, we always want to test for the assumptions of normality on our response variable. A linear regression has four assumptions:

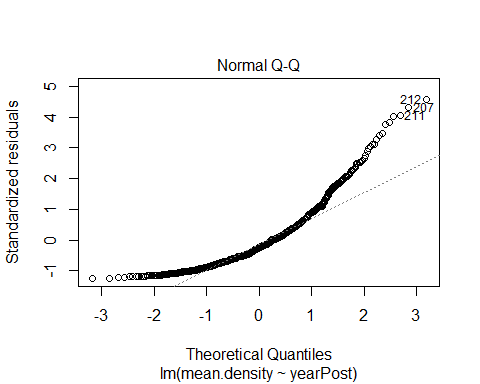
1. *Linearity of the data:* linear relationship between *x* and *y*. (residuals vs fitted values)
2. *Normality of residuals:* residual errors are normally-distributed. (QQ plot)
3. *Homogeneity of residuals variance (i.e., homoscedasticity):* constant variance of the residuals. (scale-location plot)
4. *Independence of residuals error terms:* depends on what may be dependent on your residuals.

We can test for relevant assumptions using diagnostic plots:

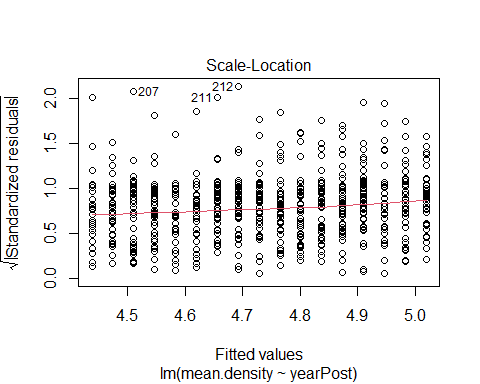
hares$yearPost <- hares$year - 1991 # rescale year to converge on an estimate, scale(year) also works  
dens6 <- lm(mean.density ~ yearPost, data = hares)  
  
plot(dens6, 1) # Residuals vs fitted - red line is flat meaning linear relationsip



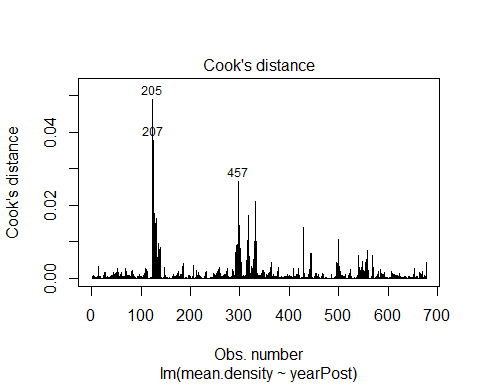
plot(dens6, 2) # QQ plot - examine whether residuals are normally-distributed



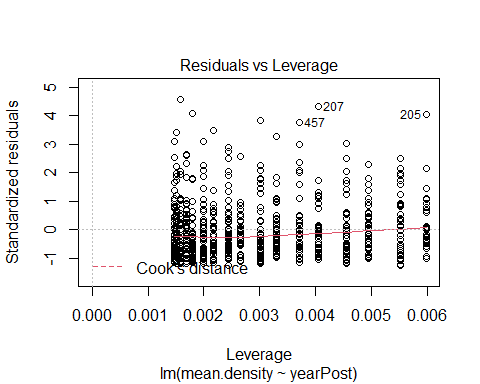
plot(dens6, 3) # Scale-location - homogeneity of variance of residuals (homoscedasticity) if horizontal line with equal spread



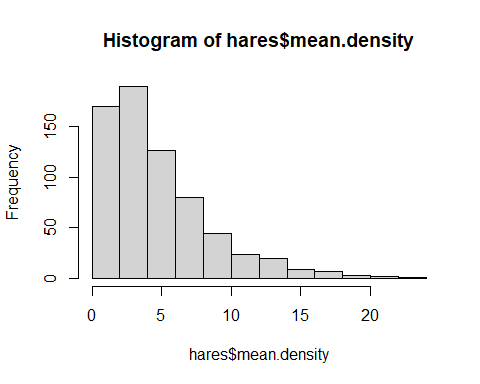
plot(dens6, 4) # Cook's distance - identify extreme values and their obs. number



plot(dens6, 5) # Residuals vs leverage - for identifying influential cases or extreme values

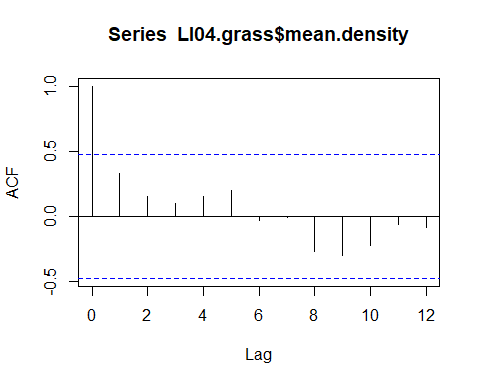


hist(hares$mean.density) # left-skewed

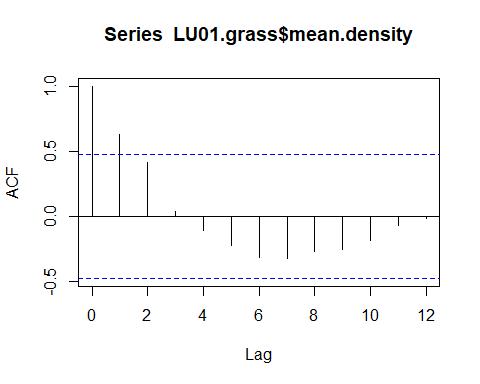


It is difficult to do a general test for dependence of the residual error term. You will need to know why your residual error term may be dependent: either residuals can correlate with another variable (e.g., check residuals-fitted plot) or residuals can correlate with a nearby residual (e.g., autocorrelation in time series data).

LI04.grass <- subset(hares, site == "BE12" & !is.na(mean.density))  
acf(LI04.grass$mean.density)



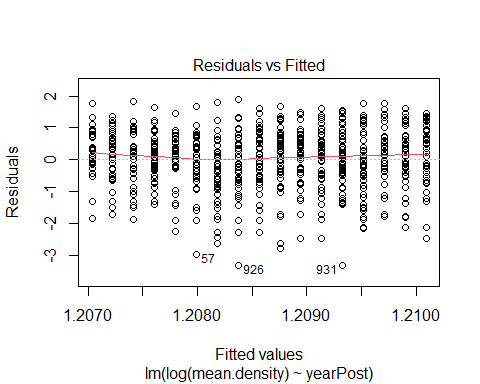
LU01.grass <- subset(hares, site == "GE02" & !is.na(mean.density))  
acf(LU01.grass$mean.density)



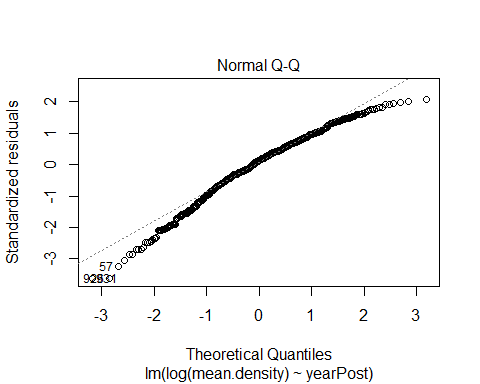
Our residual error terms seem to be independent, particularly from temporal autocorrelation. Regardless, our data do not meet all the assumptions of a linear regression (i.e., normality of residuals since data are negatively (left) skewed and bound by 0 to ).

First, we can use the most common approach when our data do not met the assumptions of normality, which is to transform our response variable, *y*. There are many ways to transform your data (e.g., log, square-root, arcsine). However, we will only cover the log transformation of our response variable, *y* (i.e., log(*y*) is normal given *x*):

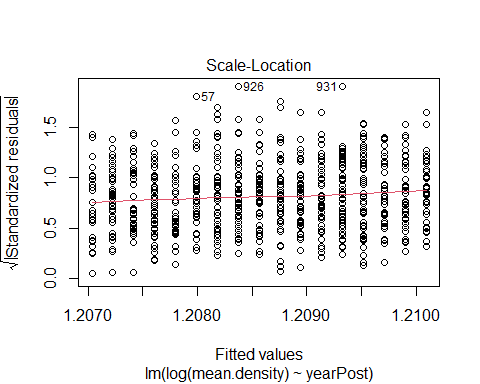
dens7 <- lm(log(mean.density) ~ yearPost, data = hares)  
  
plot(dens7, 1) # linear relationship between x and y (good)



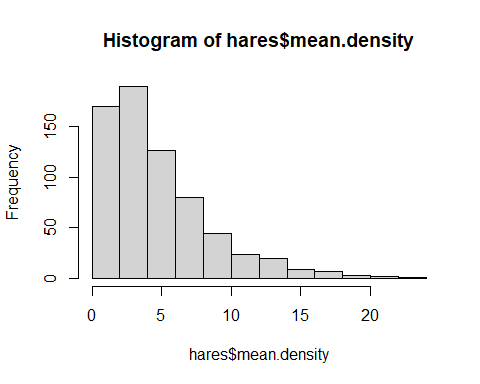
plot(dens7, 2) # residuals are not normality-distributed, mild right-skew (ok)



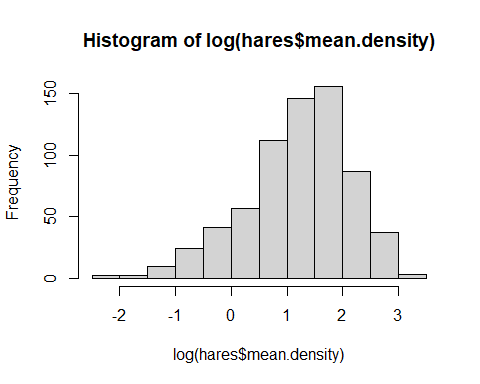
plot(dens7, 3) # mild constant variance of the residuals (ok)



hist(hares$mean.density) # negatively/left skewed

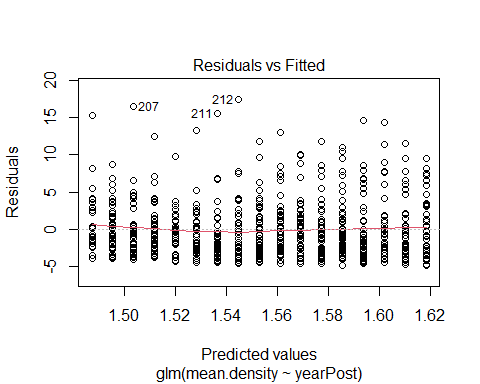


hist(log(hares$mean.density)) # slightly positively/right skewed

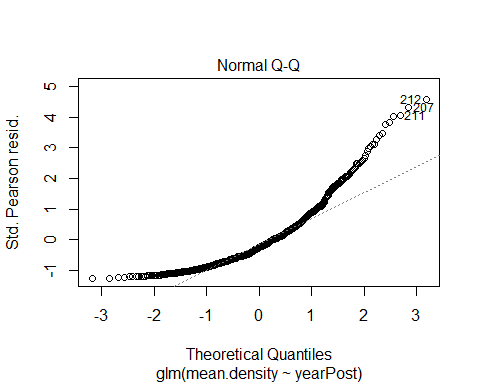


Instead of transforming our response to fit a statistical distribution (i.e., log(*y*) is normal given *x*), we can choose a statistical distribution that fits our data. First, we might want to try the log-link Gaussian distribution (i.e., mean of log(*y*) responses linearly to *x*), such that:

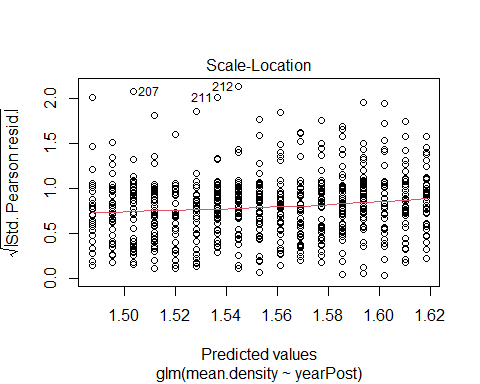
dens8 <- glm(mean.density ~ yearPost, family = gaussian(link = "log"), data = hares) # glm() instead of lm()  
  
plot(dens8, 1) # relationship between x and y seems linearly (good)



plot(dens8, 2) # residuals are not normality-distributed, left-skewed (bad)



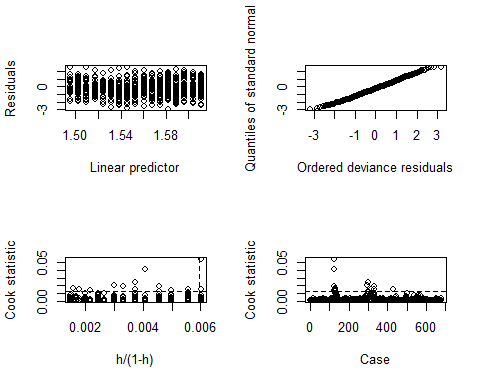
plot(dens8, 3) # constant variance of the residuals (ok)



Compared to log-transformation of the *y* values, the log-link Gaussian does not seem to perform better when checking the assumptions of a linear regression.

We could also use the Gamma distribution that allows for non-negative, skewed, continuous data that are bound by 0 to . The Gamma distribution (either log Gamma or inverse Gamma) assumes heavier tailed/skewed distribution, particularly the inverse Gamma distribution.

dens9 <- glm(mean.density ~ yearPost, family = Gamma(link = "log"), data = hares)  
  
gamma.diag <- boot::glm.diag(dens9)   
boot::glm.diag.plots(dens9, gamma.diag)



The plot(dens9) does not provide the correct diagnostic plots for the Gamma distribution. The “boot” package allows us to assess the relevant assumptions for the Gamma distribution. The Gamma distribution (or any distribution with a scale term) does not need to worry about heteroscedasticity.

**Practice example 5:** Test the assumptions of normality for the simple linear regression model that was run in code chunk #17. Then, re-run the model with a Gamma distribution and evaluate the assumptions.

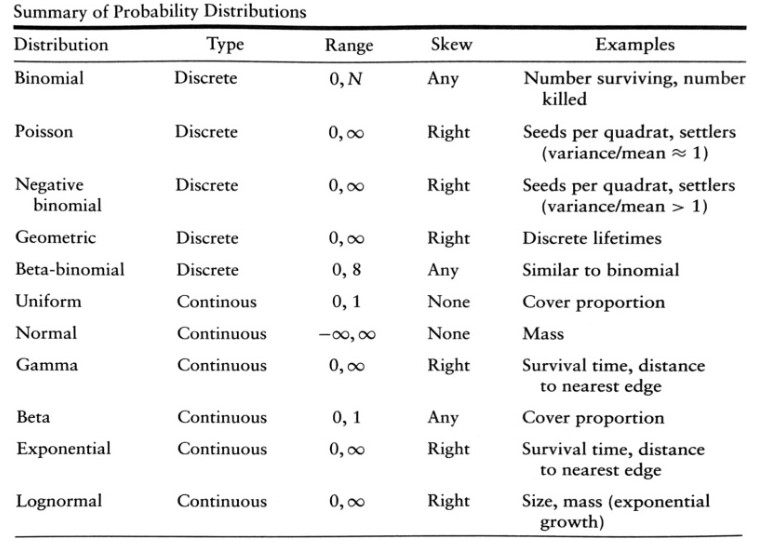
cw1 <- lm(weight ~ Time, data = ChickWeight)

### 2. Model selection

Here, I will outline the general procedure for model selection in two parts: selecting the appropriate probability distribution and selecting variables in your model.

#### Select the probability distribution

1. Based on first principles



Source: B. Bolker (2008) Ecological Models and Data in R, Princeton University Press, Princeton, New Jersey, USA

1. Based on analysis of data

We could also select the appropriate probability distribution from analyzing our data. To recap, we ran four models to evaluate whether mean density of brown hares changes linearly with year. We can use Akaike Information Criteria (AIC) to rank our candidate models:

where *k* is the number of parameters in the model and *LL* is the log likelihood of the model. AIC decreases with increasing model fit (*LL*) but penalizes for the number of parameters. The lower the AIC, the better the model. Here, we can use the ICtab function from the bbmle package.

# from testing our assumptions  
dens7 <- lm(mean.density ~ yearPost, data = hares)  
dens8 <- lm(log(mean.density) ~ yearPost, data = hares)   
dens9 <- glm(mean.density ~ yearPost, family = gaussian(link = "log"), data = hares)  
dens10 <- glm(mean.density ~ yearPost, family = Gamma(link = "log"), data = hares)  
  
ICtab(dens7, dens8, dens9, dens10, base = T) # log transformation wins

## AIC dAIC df  
## dens8 1800.9 0.0 3   
## dens10 3391.6 1590.7 3   
## dens9 3738.4 1937.5 3   
## dens7 3738.5 1937.6 3

# from practice 5  
cw1 <- lm(weight ~ Time, data = ChickWeight)  
cw1\_lg <- glm(weight ~ Time, family = Gamma(link = "log"), data = ChickWeight)  
  
ICtab(cw1, cw1\_lg) # gamma wins

## dAIC df  
## cw1\_lg 0.0 3   
## cw1 497.2 3

The common rule would be to choose the most parsimonious model within 2 dAIC of the winning model.

**Note:** If multiple models fall witin 2 dAIC of the winning model, you can choose to do something referred to as “model averaging”. Model averaging is not ideal for model with different probability distributions. It would be better for averaging multiple winning models, such a linear and quadratic function since the model falls somewhere between those functions. The *bbmle::ICtab()* function has an argument for computing IC weights that allows for model averaging. For more information on model averaging, see Anderson and Burham 2002.

#### Selecting variables

For selecting variables in your model, you can do model selection in two ways: Akaike Information Criteria (AIC) for competing models or likelihood ratio test (LRT) of nested models.

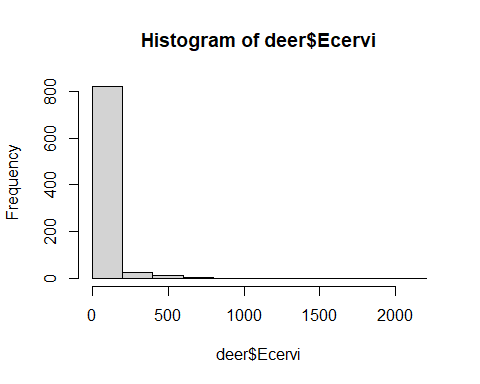
1. AIC analyses of competing models

We can also use AIC values to select a winning model based on a set of candidate models with different covariates.

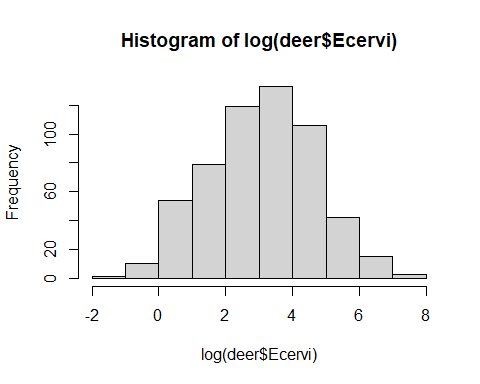
Here, we are using epidemiological data from [Vicente et al. 2006](https://doi.org/10.1051/vetres:2005044) that took parasite loads in red deer reared on a number of estates in Spain. Observations on red deer were taken at different farms, months, year (0-5 are 2000-2005, 99 is 1999), and sexes (1 - Male, 2 - Female). For each observation, the length of the animal (LCT, length of head-body), kidney-fat index (KFI), the number of *Elaphostrongylus cervi* parasites (Ecervi) per grams of wet feces, and the presence (0, 1) of Tuberculosis were taken.

Here, we might evaluate whether infection intensity of *Elaphostrongylus cervi* parasites in red deer correlated better with length of the animal (LCT) or kidney fat index (KFI).

deer <- read.table(here::here("Session 2", "Data", "Deer.txt"), header = T)  
  
hist(deer$Ecervi)



hist(log(deer$Ecervi))



ec1 <- lm(log(Ecervi + 0.01) ~ KFI, data = deer)  
ec2 <- lm(log(Ecervi + 0.01) ~ LCT, data = deer)  
  
ICtab(ec1, ec2) # KFI is better than LCT at explaining parasite load

## dAIC df  
## ec1 0.0 3   
## ec2 856.4 3

1. LRT and “marginal hypothesis testing” of a saturated model.

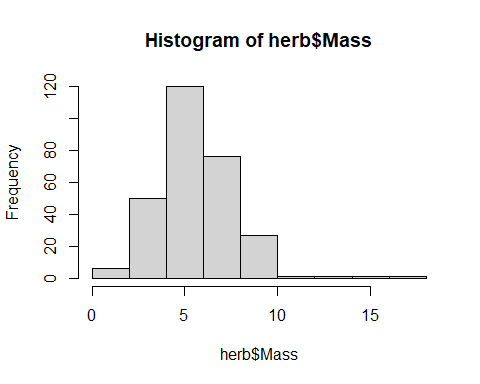
For nested models, we should use “marginal null hypothesis testing” of a saturated model to determine significant effects of nested terms.

For example, we might want to evaluate whether Taylor Checkerspot caterpillars reared on two host plant (i.e., Indian paintbrush and English Plantain) that were either sprayed or not sprayed with herbicide treatment affected their mass.

herb <- read.csv(here::here("Session 2", "Data", "HerbicideCaterpillars.csv"))  
head(herb)

## Species Host Treat Mass  
## 1 TCB Castilleja Control 4.53  
## 2 TCB Castilleja Control 3.67  
## 3 TCB Castilleja Control 5.84  
## 4 TCB Castilleja Control 2.00  
## 5 TCB Castilleja Control 4.25  
## 6 TCB Castilleja Control 5.44

hist(herb$Mass)



sat\_mod <- glm(Mass ~ Host + Treat + Host:Treat, family=gaussian, data = herb)   
car::Anova(sat\_mod)

## Analysis of Deviance Table (Type II tests)  
##   
## Response: Mass  
## LR Chisq Df Pr(>Chisq)   
## Host 155.777 1 < 2e-16 \*\*\*  
## Treat 6.200 1 0.01278 \*   
## Host:Treat 0.168 1 0.68177   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

add\_mod <- glm(Mass ~ Host + Treat, family=gaussian, data = herb)   
host\_mod <- glm(Mass ~ Host, family=gaussian, data = herb)  
treat\_mod <- glm(Mass ~ Treat, family=gaussian, data = herb)   
  
lmtest::lrtest(add\_mod, sat\_mod)

## Likelihood ratio test  
##   
## Model 1: Mass ~ Host + Treat  
## Model 2: Mass ~ Host + Treat + Host:Treat  
## #Df LogLik Df Chisq Pr(>Chisq)  
## 1 4 -537.33   
## 2 5 -537.25 1 0.1705 0.6797

lrtest(host\_mod, add\_mod)

## Likelihood ratio test  
##   
## Model 1: Mass ~ Host  
## Model 2: Mass ~ Host + Treat  
## #Df LogLik Df Chisq Pr(>Chisq)   
## 1 3 -540.44   
## 2 4 -537.33 1 6.2162 0.01266 \*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

lrtest(treat\_mod, add\_mod)

## Likelihood ratio test  
##   
## Model 1: Mass ~ Treat  
## Model 2: Mass ~ Host + Treat  
## #Df LogLik Df Chisq Pr(>Chisq)   
## 1 3 -600.07   
## 2 4 -537.33 1 125.48 < 2.2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

### 3. Intro to generalized linear models

Generalized linear models (GLMs) connect a mean of the response to its predictors in a linear way through “*link functions*”. Therefore, it produces coefficients of a linear relationship on the link function scale. Instead of transforming data to fit a normal distribution, such as the log transformation:

GLMs allows data to follow alternative error distributions, such as a Poisson error distribution on a natural log link function scale:

where the log of mean is a linear function of *x*.

#### Binomial distribution

Before explaining the Binomial distribution, we should explain the Bernoulli distribution. The Bernoulli distribution describes the probability of getting a single event in a single trial (or a particular sequence of results in some number of trials), whereas the binomial distribution describes the probability of getting *k* events out of *N* unordered trials, if each trial has a Bernoulli distribution with probability *p* of an event. Therefore, a binomial distribution with only one trial is a Bernoulli distribution.

* **Bernouli trial** - if we tossed a coin six times and we got five heads, we could represent this as one trial where we got 4 successes and 2 failures, where the probability of getting heads is 0.67 (=4/6). The number of trials *N* is 1.
* **Binomial trial** - if we tossed a coin six times, we got heads (1), tails (0), heads (1), heads (1), tails (0), and heads (1). If each toss was treated as an independent Bernoulli trial, the number of trials *N* is 6 and the *k* events is 4.

Below, we will use generalized linear models to estimate a response that is binomially-distributed. A binomial response can either be at the individual-level (i.e., presence/absence where each “0” or “1” is an independent Bernoulli trial) or at the group-level (e.g., *k* events and *N* trials).

**Individual-level response**

For our first example, we will be using data from a laboratory diapause experiment to estimate the probability of entering diapause as a function of photoperiod. Mustard white butterfly (*Pieris oleracea*) larvae were reared in five photoperiod treatments, then I reported whether each individual had survived (1 for survival, 0 for died) and whether they diapaused (1 for entering diapause, 0 for eclosing) conditioned on surviving. These data were used in [Kerr et al. 2020](https://doi.org/10.1111/gcb.14959).

diap <- read.csv(here::here("Session 2", "Data", "MW\_DiapausingExp.csv"))  
head(diap)

## Treatment Sex Surv Diapause  
## 1 11.5 female 1 1  
## 2 11.5 male 1 1  
## 3 11.5 male 0 NA  
## 4 12.5 female 1 1  
## 5 12.5 female 1 1  
## 6 12.5 female 1 1

We can run a logit-link GLM to estimate the probability of diapausing conditioned on survival as a function of photoperiod, as follows:

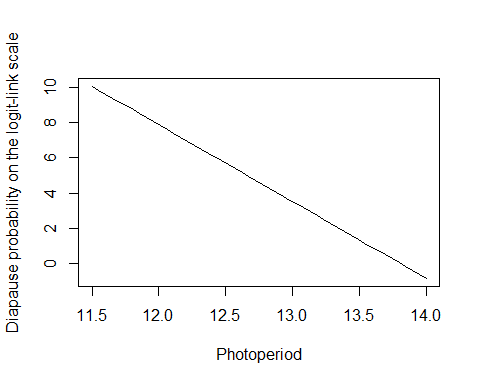
pd1 <- glm(Diapause ~ Treatment, family = binomial, data = subset(diap, Sex == "female"))  
Anova(pd1)

## Analysis of Deviance Table (Type II tests)  
##   
## Response: Diapause  
## LR Chisq Df Pr(>Chisq)   
## Treatment 104.49 1 < 2.2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

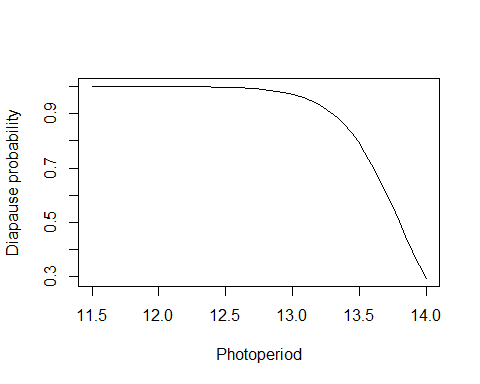
summary(pd1)

##   
## Call:  
## glm(formula = Diapause ~ Treatment, family = binomial, data = subset(diap,   
## Sex == "female"))  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -2.66078 0.00913 0.08165 0.24268 1.56334   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) 60.4946 9.8336 6.152 7.66e-10 \*\*\*  
## Treatment -4.3834 0.7154 -6.127 8.93e-10 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 216.53 on 211 degrees of freedom  
## Residual deviance: 112.04 on 210 degrees of freedom  
## (15 observations deleted due to missingness)  
## AIC: 116.04  
##   
## Number of Fisher Scoring iterations: 7

light.vals <- seq(min(diap$Treatment), max(diap$Treatment), length.out = 50)  
pd.logit <- coef(pd1)[1] + coef(pd1)[2]\*light.vals # logit-link scale estimates  
pd.pred <- plogis(coef(pd1)[1] + coef(pd1)[2]\*light.vals) # back-transformed estimates  
  
plot(light.vals, pd.logit, type = "l", ylab = "Diapause probability on the logit-link scale", xlab = "Photoperiod")



plot(light.vals, pd.pred, type = "l", ylab = "Diapause probability", xlab = "Photoperiod")



manual.pred <- exp(coef(pd1)[1] + coef(pd1)[2]\*light.vals) / (1 + exp(coef(pd1)[1] + coef(pd1)[2]\*light.vals))  
tail(pd.pred)

## [1] 0.5610065 0.5054010 0.4496616 0.3951582 0.3431419 0.2946374

tail(manual.pred)

## [1] 0.5610065 0.5054010 0.4496616 0.3951582 0.3431419 0.2946374

First, the summary output includes two more outputs for a GLM:

* *Null deviance:* is the 2(LL(Saturated Model) - LL(Null Model)), where the saturated model assumes each data point has its own parameter (*n* parameters) and the null assumes one parameter for all data (1 parameter).
* *Residual deviance:* is the 2(LL(Saturated Model - LL(Proposed Model)) and refers the goodness-of-fit of the proposed model, where your data can be explained by *p* parameters and an intercept term.

To compare your null with the proposed, you can calculate the chi-squared value () and the degrees of freedom ().

**Group-level response**

For our second example, we will be using data from an overwintering experiment to estimate the overwinter survival of diapausing pupa of the mustard white butterfly. Each bug dorm had a fall count of diapausing pupa (*N*, trials) and a spring count of the number of emerging butterflies (*k*, events). In other words, we evaluate the number of success as the number of emerging butterflies (i.e., *k*) and the number of failures as the number of non-emerging butterflies (i.e., *N - k*).

We can run an intercept-only logit-link GLM to estimate the overwinter survival of the mustard white butterfly, as follows:

surv <- read.csv(here::here("Session 2", "Data", "MW\_OverwinterSurv.csv"))  
head(surv)

## Dorm Fall.count Spring.count  
## 1 1 3 1  
## 2 3 20 0  
## 3 6 8 2  
## 4 7 2 0  
## 5 11 11 0  
## 6 13 11 2

ws1 <- glm(cbind(Spring.count, Fall.count - Spring.count) ~ 1, family = binomial, data = surv)  
summary(ws1)

##   
## Call:  
## glm(formula = cbind(Spring.count, Fall.count - Spring.count) ~   
## 1, family = binomial, data = surv)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -2.3610 -1.0870 -0.6373 0.9063 1.9614   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) -1.900 0.268 -7.089 1.35e-12 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 28.474 on 19 degrees of freedom  
## Residual deviance: 28.474 on 19 degrees of freedom  
## AIC: 48.119  
##   
## Number of Fisher Scoring iterations: 5

plogis(coef(ws1)) # 13% probability of surviving the winter

## (Intercept)   
## 0.1300813

exp(coef(ws1)) / (1 + exp(coef(ws1)))

## (Intercept)   
## 0.1300813

I think the group-level binary response has many potential applications:

For example, I have used it to estimate the trap color of Venus flytraps based on crude categorization of color along a spectrum from green to dark red.



Figure 2: Trap color of Venus flytraps from four populations (or eight sub-sites) across their range

The colors of traps were one of seven categories ranging from green, green-pink, pink, pink-red, red, red-dark red, and dark red. The total number of trials was 6, where green was 0 successes + 6 failures and dark red was 6 successes + 0 failures.

The following are three Bernoulli trials for three plants:

cbind(0, 6) # green cbind(3, 3) # pink-red cbind(6, 0) # dark red

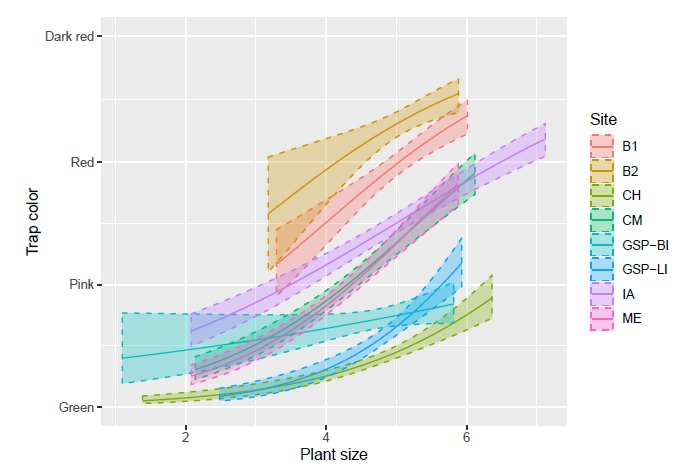


Figure 3. Trap color of Venus flytraps from four populations (or eight sub-sites) across their range

Another example could be estimating the day of spring emergence for a diapausing insect, where *N* trials is the 365 days of a year (assuming not a leap year) and *k* events is the number of days into the year when it emerged. In other words, the number of successes would be days from January 1st that it took to emerge and the number of failures would be the remaining days of the year. Then, you would get some probability (into the year) that an insect would emerge from diapause. You could multiple that probability estimate by the number of days in a year to convert back into “days until spring emergence”.

This binary “group-level” format has many potential ecological applications.

**Practice example 6:** Using the deer data, run a logit-link GLM to estimate the probability that a red deer will have the parasite *Elaphostrongylus cervi* given its kidney fat index (KFI) and sex. Run marginal hypothesis testing to evaluate the interactive effects of these two covariates.

#### Poisson distribution

Here, we will be exploring running log-link GLMs using the Poisson error distribution. In simple terms, the Poisson distribution is used for discrete count data that are bound by 0 to .

In more complex terms, the Poisson distribution is like the binomial distribution in that it has a large “unknown” number of trials *N* and a small probability of an event , where is the *k* discrete events (or mean count estimate). The probability of given number of events can be thought of as the probability of counting an individual that occurs over a fixed interval of time or space.

Here, we have counts of wolves in each year from 1982 to 2012 across five US states.

wolves <- read.csv(here::here("Session 2", "Data", "NRMwolves.csv"))  
head(wolves)

## year num.wolves MT.wolves WY.wolves ID.wolves OR.wolves WA.wolves  
## 1 1982 8 8 NA NA NA NA  
## 2 1983 6 6 NA NA NA NA  
## 3 1984 6 6 NA NA NA NA  
## 4 1985 13 13 NA NA NA NA  
## 5 1986 15 15 NA NA NA NA  
## 6 1987 10 10 NA NA NA NA

First, we will run a Poisson GLM to evaluate whether the total number of wolves has changed over a 10-year period from 1994-2003.

with(wolves[13:22,], plot(year, num.wolves))  
  
wv10a <- glm(num.wolves ~ year, family = poisson(link = "identity"), data = wolves[13:22,])  
coef(wv10a) # year is the change in the # of wolves per year

## (Intercept) year   
## -141848.51720 71.15512

wv10b <- glm(num.wolves ~ year, family = poisson(link = "log"), data = wolves[13:22,])  
coef(wv10b)

## (Intercept) year   
## -486.6604133 0.2463317

exp(coef(wv10b)[2]) # slope is the population growth rate

## year   
## 1.279324

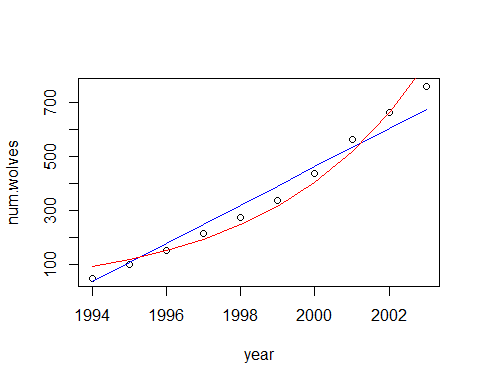
wolves$num.wolves[13] # number of wolves in 1994

## [1] 48

exp(coef(wv10b)[2]) \* wolves$num.wolves[14] # estimated number of wolves in 1995

## year   
## 129.2117

with(wolves[13:22,], plot(year, num.wolves))  
points(1994:2003, coef(wv10a)[1] + coef(wv10a)[2]\*1994:2003, type = "l", col = "blue") # linear growth  
points(1994:2003, exp(coef(wv10b)[1] + coef(wv10b)[2]\*1994:2003), type = "l", col = "red") # unlimited expontential growth



Second, we will evaluate whether the total number of wolves has changed over the full 30-year period from 1983 to 2012.

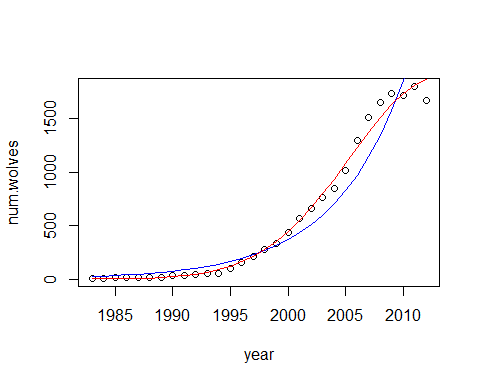
with(wolves[2:31,], plot(year, num.wolves))  
  
wv30a <- glm(num.wolves ~ year, family = poisson(link = "log"), data = wolves[2:31,]) # unlimited exp growth   
coef(wv30a)

## (Intercept) year   
## -318.2374555 0.1620725

wv30b <- glm(num.wolves ~ year + I(year\*year), family = poisson(link = "log"), data = wolves[2:31,]) # quadratic function model  
coef(wv30b)

## (Intercept) year I(year \* year)   
## -3.369141e+04 3.347835e+01 -8.314798e-03

with(wolves[2:31,], plot(year, num.wolves))  
points(1983:2012, exp(coef(wv30a)[1] + coef(wv30a)[2]\*1983:2012), type = "l", col = "blue")   
points(1983:2012, exp(coef(wv30b)[1] + coef(wv30b)[2]\*1983:2012 + coef(wv30b)[3]\*1983:2012\*1983:2012), type = "l", col = "red")



**Practice example 7:** Using the “Bombus\_eggs.rds” R object file, run a Poisson GLM to estimate the number of new eggs laid in bumblebee colonies (*Bombus vosnesenskii*) as a function of days of the experiment (“Days”). Evaluate whether a model with a linear or quadratic function better fits these data. Then, plot the raw data with model predicted values from both models.

readRDS() # for loading R object files

### 4. Common issues with count data

Here, we will cover how to deal with three common issues that you may encounter with Poisson-distributed data:

1. When counts should be represented as rates
2. When count data are over- or under-dispersed for the Poisson distribution variance
3. When count data are zero-inflated

#### … when counts should be rates

A common issue in ecology is when each observation of count data are not always equally in represented in space or time. For example, you may find yourself in a situation where each observation is collected over different lengths of time or space.

Here, we will use the [Audubon Christmas bird count](www.christmasbirdcount.org) 2013 data on Northern Flickers across New England. Since observation periods differ, we want to model our counts are rates (# per hour of observation).

flickers <- read.csv(here::here("Session 2", "Data", "NE\_flickers.csv"))  
head(flickers)

## Code Name Latitude Longitude hours Count  
## 1 CTBA Barkhamsted 41.9123 -72.9884 72.50 0  
## 2 CTEW Edwin Way Teale Trail Wood 41.7966 -71.9274 48.50 0  
## 3 CTGS Greenwich-Stamford 41.0826 -73.6138 182.50 53  
## 4 CTHA Hartford 41.7660 -72.6727 198.00 0  
## 5 CTLH Litchfield Hills 41.7703 -73.2724 69.95 0  
## 6 CTLS Lakeville-Sharon 41.9449 -73.4399 49.50 0

nf1 <- glm(Count ~ 1, family = poisson, data = flickers)  
nf1\_offset <- glm(Count ~ 1, offset = log(hours), family = poisson, data = flickers)  
ICtab(nf1, nf1\_offset)

## dAIC df  
## nf1\_offset 0.0 1   
## nf1 569.1 1

exp(coef(nf1\_offset)) # 0.07 birds per hour

## (Intercept)   
## 0.07294809

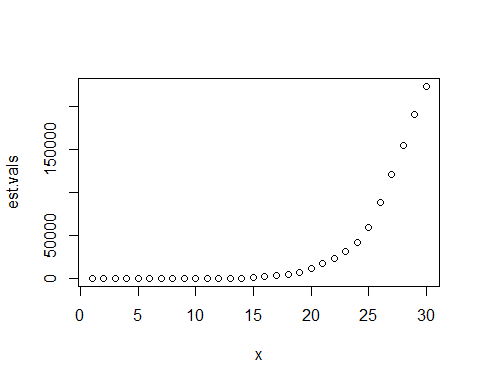
Why “offset = log(hours)” in our model? Well, the Poisson offset can be represented algebraically as follows:

which can be separated out as follows:

To estimate counts, the formula can be rearranged as follows:

We can also use offsets for running a process error model for evaluating population dynamics, where the counts in the current time step are offset by the previous time step. Therefore, you get an estimated growth rate for each time step of the series.

wv30a <- glm(num.wolves ~ year, family = poisson(link = "log"), data = wolves[2:31,]) # observation error model  
  
x = 1:30  
est.vals <- wolves$num.wolves[2]  
for(i in 2:30){   
 est.vals[i] <- wolves$num.wolves[i-1] \* exp(coef(wv30a)[2])^x[i] # simple discrete exp growth, N(t+x) = N(t) \* r^x  
}  
plot(x, est.vals) # exponential growth



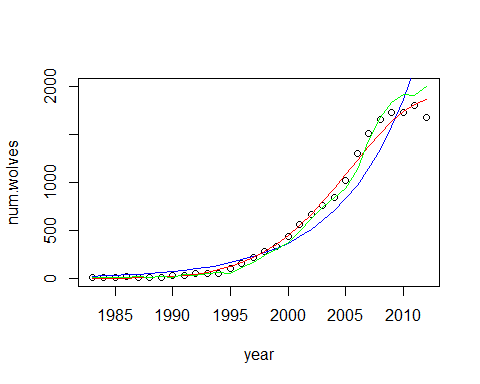
wv30b <- glm(num.wolves ~ year + I(year\*year),   
 family = poisson(link = "log"), data = wolves[2:31,]) # quadratic function model  
  
wv30c <- glm(num.wolves[2:31] ~ 1, offset = log(num.wolves[1:30]),   
 family = poisson(link = "log"), data = wolves) # process error model  
exp(coef(wv30c)) # growth rate from each time step

## (Intercept)   
## 1.108238

wolves$num.wolves[2:31] \* exp(coef(wv30c))

## [1] 6.649428 6.649428 14.407095 16.623571 11.082380 15.515333  
## [7] 13.298857 36.571856 32.138903 52.087188 60.953093 53.195426  
## [13] 111.932043 168.452183 236.054704 304.765463 373.476221 484.300026  
## [19] 623.938020 734.761824 843.369153 937.569387 1130.402807 1440.709459  
## [25] 1676.764163 1834.133966 1920.576533 1909.494153 1999.261435 1855.190489

with(wolves[2:31,], plot(year, num.wolves, ylim = c(0, 2000)))  
points(1983:2012, exp(coef(wv30a)[1] + coef(wv30a)[2]\*1983:2012), type = "l", col = "blue") # observation error  
points(1983:2012, exp(coef(wv30b)[1] + coef(wv30b)[2]\*1983:2012 + coef(wv30b)[3]\*1983:2012\*1983:2012), type = "l", col = "red")  
points(1983:2012, predict(wv30c, type = "response"), type = "l", col = "green") # process error



ICtab(wv30a, wv30b, wv30c) # quadratic model wins

## dAIC df  
## wv30b 0.0 3   
## wv30c 86.2 1   
## wv30a 1250.3 2

**Practice example 8:** Similarly, we could also model the brown hare counts *per area* as a Poisson offset instead of log(mean.density) using the normal distribution. Do you find similar estimates for mean density for the two landuse types when using Poisson distribution compared to log-transformed data?

#### … when data are over-dispersed

A common issue when fitting Poisson models is overdispersion, i.e., when count data has more variability than expected for a Poisson distribution. For Poisson models, data are less likely to be underdispersed for this given distribution. However, this is still likely to happen for biological data.

To test for overdispersion, the residual deviance should be equal to the degrees of freedom. In other words, the ratio of residual deviance/degrees of freedom should be equal to 1. If the ratio is greater than 1, then your data is more dispersed than the Poisson error distribution permits. If the ratio is less than 1, then your data are less dispersed than the Poisson error distribution.

Let’s test whether Northern Flicker counts are overdispersed, using the “intercept-only” model from the Poisson offset example #1.

nf1\_offset <- glm(Count ~ 1, offset = log(hours), family = poisson, data = flickers) # model above  
summary(nf1\_offset)

##   
## Call:  
## glm(formula = Count ~ 1, family = poisson, data = flickers, offset = log(hours))  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -5.375 -3.283 -2.691 -1.757 11.576   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) -2.61801 0.04089 -64.02 <2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for poisson family taken to be 1)  
##   
## Null deviance: 1567.4 on 101 degrees of freedom  
## Residual deviance: 1567.4 on 101 degrees of freedom  
## AIC: 1681.8  
##   
## Number of Fisher Scoring iterations: 6

nf1\_offset$deviance/nf1\_offset$df.residual # overdispersed

## [1] 15.51898

Let’s test whether hare counts using Poisson offset model was overdispersed.

dens12 <- glm(count1 ~ 1, offset = log(area), family = poisson, data = hares) # model above  
summary(dens12)

##   
## Call:  
## glm(formula = count1 ~ 1, family = poisson, data = hares, offset = log(area))  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -8.348 -3.321 -1.175 2.026 12.752   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) 1.482511 0.007233 205 <2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for poisson family taken to be 1)  
##   
## Null deviance: 10492 on 665 degrees of freedom  
## Residual deviance: 10492 on 665 degrees of freedom  
## (286 observations deleted due to missingness)  
## AIC: 13628  
##   
## Number of Fisher Scoring iterations: 5

dens12$deviance/dens12$df.residual

## [1] 15.7779

In both cases, these data are overdispersed for the Poisson distribution.

There are multiple ways to deal with overdispersed count data:

* Negative binomial distribution (family = nbinom1 or nbinom2)
* Conway-Maxwell-Poisson distribution (family = compois)
* [Observation-level random effects](https://peerj.com/articles/616/)

Here, we will use the negative binomial distribution, which is the most common approach for overdispersion. A negative binomial distribution is essentially the average of different Poisson distributions with different means. Unlike the Poisson distribution where the mean is equal to the limits, the variance of the negative binomial can be either linear (nbinom1) or quaratic (nbinom2) parameterization of the mean. Therefore, a negative binomial mean closer to its limit -> will resemble a Poisson distribution. But closer to the bound of 0, it allows for greater variance in relation to its mean.

Let’s re-run the intercept-only Poisson offset model for Northern Flicker counts using a negative binomial distribution using glmmTMB function in the glmmTMB package by Ben Bolker.

nf2 <- glmmTMB(Count ~ 1, offset = log(hours), family = nbinom1, data = flickers)  
nf3 <- glmmTMB(Count ~ 1, offset = log(hours), family = nbinom2, data = flickers)  
ICtab(nf2, nf3) # variance is linearly parameterized to its mean

## dAIC df  
## nf2 0.0 2   
## nf3 9.5 2

summary(nf1\_offset)$coefficients # Poisson variance

## Estimate Std. Error z value Pr(>|z|)  
## (Intercept) -2.618007 0.04089262 -64.02151 0

summary(nf2)$coefficients$cond # NB allows for greater variance

## Estimate Std. Error z value Pr(>|z|)  
## (Intercept) -2.618007 0.3574455 -7.324214 2.403021e-13

#### … when data are under-dispersed

Undispersion is far less common than overdispersion, but it can be often encountered in ecology with species that have small clutch/litter sizes. For example, when a bird may only lay up to 6 eggs per clutch.

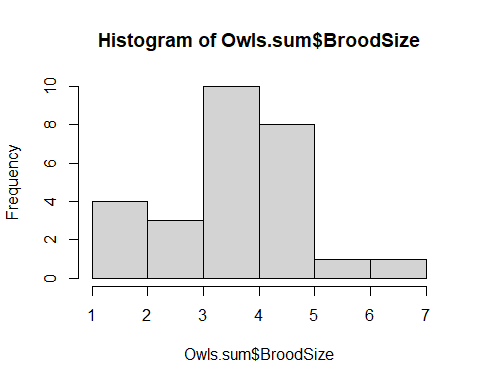
Here, we have a dataset on begging behaviour of nestling barn owls that may have an example of underdispersed count data from package glmmTMB.

?Owls

Roulin and Bersier (2007) looked at nestlings’ response to the provisioning parent under two food treatments. Using microphones inside and a video camera outside the nests, studying vocal begging behaviour when the parents bring prey for 27 nests. We use ‘sibling negotiation,’ defined as the number of calls by the nestlings in the 30-second interval immediately prior to the arrival of a parent, divided by the number of nestlings. Data were collected between 21.30 hours and 05.30 hours on two consecutive nights. The variable ArrivalTime indicates the time at which a parent arrived at the perch with prey.

Using these data, we want to estimate the brood size of barn owls and evaluate whether these data are underdispersed.

attach(Owls)  
Owls.sum <- Owls %>% group\_by(Nest) %>% distinct(BroodSize)   
  
hist(Owls.sum$BroodSize)



bs1 <- glm(BroodSize ~ 1, family = poisson, data = Owls.sum)  
summary(bs1) # on the log-link scale

##   
## Call:  
## glm(formula = BroodSize ~ 1, family = poisson, data = Owls.sum)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -1.81192 -0.27973 -0.01846 0.46189 1.33404   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) 1.39551 0.09578 14.57 <2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for poisson family taken to be 1)  
##   
## Null deviance: 12.274 on 26 degrees of freedom  
## Residual deviance: 12.274 on 26 degrees of freedom  
## AIC: 101.06  
##   
## Number of Fisher Scoring iterations: 4

exp(coef(bs1)) # back-transformed scale, approximately 4 nestlings per nest

## (Intercept)   
## 4.037037

summary(bs1)$deviance/summary(bs1)$df.residual # underdispersed, < 1

## [1] 0.4720701

To address underdispersed count data, the most appropriate distribution would be the Conway-Maxwell-Poisson distribution that adds a parameter to the Poisson distribution to account for either underdispersion or overdispersion.

bs2 <- glmmTMB(BroodSize ~ 1, family = compois, data = Owls.sum)  
ICtab(bs1, bs2)

## dAIC df  
## bs2 0.0 2   
## bs1 5.4 1

The “quasipoisson” distribution (family = quasipoisson) can also be used for underdispersed count data.

**Practice example 9:** Using the bumblebee dataset from practice example #7, run a Poisson GLM for new eggs laid as a function of resource treatment and evaluate whether these data are underdispersed or overdispersed for a Poisson distribution. Then, re-run a model with an appropriate probability distribution and evaluate whether the covariate of Treatment improves model fit.

#### … when counts are zero-inflated

Here, we will run two different models when you encounter many zeros in your count data (i.e., zero-inflated count data). We can take two potential approaches: zero-inflated regression or hurdle model. The first assumes that not all zeros are “true” zeros and that some are part of the Poisson process. This can commonly occur due to observation error. For example, your count data may be number of butterflies seen, but you are not sure that not seeing an individual means that there were actually no individuals present. For the zero-inflated models, the binomial process may determine whether a location is actually suitable habitat and the count process may represent the quality of the suitable habitat. However, remember that a count of 0 may not necessarily mean that it is a not suitable habitat.

A zero-inflated Poisson model is required for this process, since not all zeros are true. Therefore, a zeroinflated model evaluates the nonzero and zero process together by evaluating the probability that a zero comes from the main “nonzero” distribution vs. the binomial distribution (i.e., an excess zero).

You can implement zero-inflated models using zeroinfl function in the pscl package.

pscl::zerinfl(formula = y ~ x\_count | x\_zero)

However, we will be using the glmmTMB package to run both zero-inflated and hurdle models, since this package has additional probability distributions not in pscl and it also allows for both these models to have mixed-effects, which we will discuss later.

ZIP1 <- glmmTMB(Count ~ 1, offset = log(hours), ziformula = ~1, family = "nbinom1", data = flickers)  
summary(ZIP1)

## Family: nbinom1 ( log )  
## Formula: Count ~ 1  
## Zero inflation: ~1  
## Data: flickers  
## Offset: log(hours)  
##   
## AIC BIC logLik deviance df.resid   
## 314.4 322.3 -154.2 308.4 99   
##   
##   
## Overdispersion parameter for nbinom1 family (): 20.7   
##   
## Conditional model:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) -1.6833 0.2069 -8.137 4.06e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Zero-inflation model:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) 0.8205 0.2995 2.74 0.00615 \*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

summary(ZIP1)$coefficients

## $cond  
## Estimate Std. Error z value Pr(>|z|)  
## (Intercept) -1.683347 0.2068816 -8.136766 4.059764e-16  
##   
## $zi  
## Estimate Std. Error z value Pr(>|z|)  
## (Intercept) 0.8205285 0.2994806 2.739839 0.006146936  
##   
## $disp  
## NULL

ZIP2 <- glmmTMB(Count ~ 1, offset = log(hours), ziformula = ~Latitude, family = "nbinom1", data = flickers) # Same as ZiP2  
lrtest(ZIP1, ZIP2) # significant effect of latitude on zero process

## Likelihood ratio test  
##   
## Model 1: Count ~ 1  
## Model 2: Count ~ 1  
## #Df LogLik Df Chisq Pr(>Chisq)   
## 1 3 -154.22   
## 2 4 -150.09 1 8.2432 0.004091 \*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

summary(ZIP2)$coefficients

## $cond  
## Estimate Std. Error z value Pr(>|z|)  
## (Intercept) -1.650126 0.1881545 -8.770061 1.785701e-18  
##   
## $zi  
## Estimate Std. Error z value Pr(>|z|)  
## (Intercept) -26.3782065 10.0536801 -2.623736 0.008697107  
## Latitude 0.6364322 0.2355161 2.702287 0.006886428  
##   
## $disp  
## NULL

ZiP3 <- glmmTMB(Count ~ Latitude, offset = log(hours), ziformula = ~., family = "nbinom1", data = flickers)  
summary(ZiP3)$coefficients

## $cond  
## Estimate Std. Error z value Pr(>|z|)  
## (Intercept) 28.2685819 6.8672159 4.116455 3.847452e-05  
## Latitude -0.7089281 0.1638307 -4.327199 1.510172e-05  
##   
## $zi  
## Estimate Std. Error z value Pr(>|z|)  
## (Intercept) -14.2387685 11.8687106 -1.199690 0.2302599  
## Latitude 0.3506068 0.2802133 1.251214 0.2108564  
##   
## $disp  
## NULL

For hurdle models, the nonzero and zero processes are modelled separately, and therefore, we assume all zeros are true zeros. The glmmTMB package also allows for hurdle models for count data using the “truncated” families (i.e., family = truncated\_poisson). Note that use of the Poisson family would indicate a zero-inflated model. Here, we are using the owl dataset that is included in glmmTMB package.

table(Owls$SiblingNegotiation)

##   
## 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19   
## 156 24 43 29 26 26 40 30 22 17 22 16 23 17 27 13 12 12 11 9   
## 20 22 23 24 25 26 28 31 32   
## 4 5 5 2 1 2 1 2 2

h\_mod\_pois <- glmmTMB(SiblingNegotiation ~ FoodTreatment, ziformula = ~FoodTreatment,   
 family = truncated\_poisson(link = "log"), data = Owls)  
h\_mod\_nbin1 <- glmmTMB(SiblingNegotiation ~ FoodTreatment, ziformula = ~FoodTreatment,   
 family = truncated\_nbinom1(link = "log"), data = Owls)  
h\_mod\_nbin2 <- glmmTMB(SiblingNegotiation ~ FoodTreatment, ziformula = ~FoodTreatment,   
 family = truncated\_nbinom2(link = "log"), data = Owls)  
h\_mod\_compois <- glmmTMB(SiblingNegotiation ~ FoodTreatment, ziformula = ~FoodTreatment,   
 family = truncated\_nbinom2(link = "log"), data = Owls)  
ICtab(h\_mod\_pois, h\_mod\_nbin1, h\_mod\_nbin2, h\_mod\_compois) # do we need to account for overdispersion?

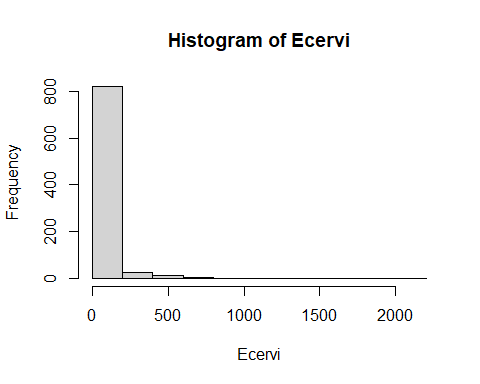
## dAIC df  
## h\_mod\_nbin1 0.0 5   
## h\_mod\_nbin2 3.9 5   
## h\_mod\_compois 3.9 5   
## h\_mod\_pois 791.2 4

summary(h\_mod\_nbin1)

## Family: truncated\_nbinom1 ( log )  
## Formula: SiblingNegotiation ~ FoodTreatment  
## Zero inflation: ~FoodTreatment  
## Data: Owls  
##   
## AIC BIC logLik deviance df.resid   
## 3381.1 3403.0 -1685.5 3371.1 594   
##   
##   
## Overdispersion parameter for truncated\_nbinom1 family (): 3.95   
##   
## Conditional model:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) 2.24752 0.04404 51.04 <2e-16 \*\*\*  
## FoodTreatmentSatiated -0.19661 0.07691 -2.56 0.0106 \*   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Zero-inflation model:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) -1.8101 0.1608 -11.256 < 2e-16 \*\*\*  
## FoodTreatmentSatiated 1.3957 0.2020 6.908 4.92e-12 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

For the deer data, let’s assume that method for detecting *Elaphostrongylus cervi* parasites in red deer is very accurate and all zeros are true zeros. In other words, there are no potential false negatives (not detecting parasite, when it’s present). Here, let’s evaluate whether the probability of infection and the infection intensity (i.e., the number of parasites per gram of wet feces) is dependent on both kidney fat index (KFI).

with(deer, hist(Ecervi)) # zero-inflated



deer$PEcervi <- deer$Ecervi  
deer$PEcervi[deer$PEcervi > 0] <- 1  
  
h\_zero <- glm(PEcervi ~ Sex, family = binomial, data = deer)  
Anova(h\_zero)

## Analysis of Deviance Table (Type II tests)  
##   
## Response: PEcervi  
## LR Chisq Df Pr(>Chisq)   
## Sex 11.151 1 0.0008397 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

h\_nzero <- glm(log(Ecervi) ~ KFI, family = gaussian, data = subset(deer, PEcervi == 1))  
Anova(h\_nzero) # no interaction

## Analysis of Deviance Table (Type II tests)  
##   
## Response: log(Ecervi)  
## LR Chisq Df Pr(>Chisq)   
## KFI 9.4023 1 0.002167 \*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

non\_hurdle <- glm(log(Ecervi + 0.01) ~ Sex\*KFI, family = gaussian, data = deer)  
Anova(non\_hurdle)

## Analysis of Deviance Table (Type II tests)  
##   
## Response: log(Ecervi + 0.01)  
## LR Chisq Df Pr(>Chisq)   
## Sex 16.4605 1 4.967e-05 \*\*\*  
## KFI 0.1323 1 0.7160   
## Sex:KFI 4.1884 1 0.0407 \*   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

# Is the hurdle model better?  
AIC(non\_hurdle)

## [1] 3665.097

2\*(length(coef(h\_zero)) + length(coef(h\_nzero))) - 2\*(logLik(h\_zero) + logLik(h\_nzero)) # AIC = 2k - 2LL

## 'log Lik.' 2647.028 (df=2)

**Practice example 10**: The Eastern bluebird data was also collected across New England from the [Audubon Christmas bird count](www.christmasbirdcount.org). Similar to the Northern Flickers, the Eastern Bluebirds have breeding grounds in the northern half of the United States, but all year grounds from Massachusetts down to Texas and exclusive wintering grounds in Texas and northern Mexico.

bluebirds <- read.csv(here::here("Session 2", "Data", "bluebirds.csv"))  
head(bluebirds)

## Code Latitude Longitude hours count  
## 1 CTBA 41.9123 -72.9884 72.50 127  
## 2 CTEW 41.7966 -71.9274 48.50 72  
## 3 CTGS 41.0826 -73.6138 182.50 60  
## 4 CTHA 41.7660 -72.6727 198.00 31  
## 5 CTLH 41.7703 -73.2724 69.95 122  
## 6 CTLS 41.9449 -73.4399 49.50 65

Using the bluebirds data, choose whether a zeroinflated model, hurdle model, or GLM of bird counts with an offset of observation hours is more appropriate for these data. Include a covariate of both latitude on the zero (i.e., probability that it has migrated) process of the models.

### 5. Mixed models

Here, we evaluated models with only parameters with fixed or non-random effects (i.e., fixed group means). Fixed effects are generally the covariates of interest. Here, we will run models with covariates that evaluate both *fixed* and *random* effects called “**mixed models**” or “**mixed effects models**”. These are a form of hierarchical models where the fixed effects are no longer assumed to be independent, but they come from a second distribution with some mean and variance of some “hyperparameter/s”. For example, a simple fixed-effects ANOVA evaluating group means can be written as follows:

where is the mean of response variable for population *j* and is the random deviance of *i* from its population mean . The random deviance of from each group mean comes from a normal distribution:

where the mean is 0 and the variance is (remember that standard error of the mean estimate is the square root of the variance).

For a random-effects ANOVA, the paramaters come from a second distribution with some mean and variance :

**NOTE:** a random-effects model rarely contains less than than five populations/groups, since estimating variance with sample size less than five will not result in very precise estimates.

A common way to distinguish fixed from random effects is respectively based on what is the subject of interest vs some random “latent” variable that may result in greater variance in your model estimates but is *not* of interest.

#### Random-coefficient models

Kery 2010 (CH 12) outlines four potential random-coefficient models:

1. Only intercepts are random, but all slopes are identical for all “random” groups.
2. Only slopes are random, but all intercepts are identical for all “random” groups. (*this model is not sensible model in most circumstances*)
3. Both intercepts and slopes are random, but they are independent.
4. Both intercepts and slopes are random, and there is a correlation between them.

Below, we will run all four combinations of random-coefficient models for the in-built dataset on Salamanders using lme4::glmer(). The Salamander dataset contains counts of salamander across sites that were either affected by mountain top coal mining (“mined”). The potential covariates of interest affecting counts are species of salamander (“spp”), cover objects in the stream (“cover”), days since precipitation (DOP), water temperature (Wtemp), and day of year (DOY).

attach(Salamanders)  
head(Salamanders)

## site mined cover sample DOP Wtemp DOY spp count  
## 1 VF-1 yes -1.4423172 1 -0.5956834 -1.22937861 -1.497003 GP 0  
## 2 VF-2 yes 0.2984104 1 -0.5956834 0.08476529 -1.497003 GP 0  
## 3 VF-3 yes 0.3978806 1 -1.1913668 1.01417627 -1.294467 GP 0  
## 4 R-1 no -0.4476157 1 0.0000000 -3.02335795 -2.712216 GP 2  
## 5 R-2 no 0.5968209 1 0.5956834 -0.14434533 -0.686860 GP 2  
## 6 R-3 no 1.3428470 1 0.5956834 -0.01466007 -0.686860 GP 1

Our models will contain fixed effects of day of year (DOY) on the presence of salamanders. First, we will run a binomial GLM without any random effects.

Salamanders$Presence <- Salamanders$count  
Salamanders$Presence[Salamanders$Presence > 1] <- 1  
  
sm0 <- glm(Presence ~ DOY, family = binomial, data = Salamanders)  
Anova(sm0)

## Analysis of Deviance Table (Type II tests)  
##   
## Response: Presence  
## LR Chisq Df Pr(>Chisq)   
## DOY 3.6403 1 0.0564 .  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

First, we will run a random-intercept model (probably the most commonly used random-coefficient model) that allows the model estimate for the intercept to come from a second distribution that estimates a mean and variance from intercepts from each species.

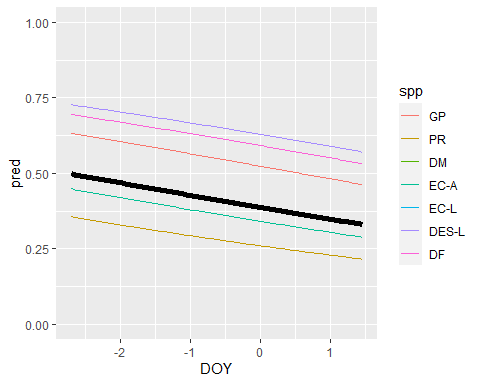
sm1 <- glmer(Presence ~ DOY + (1 | spp), family = binomial, data = Salamanders)  
summary(sm1)

## Generalized linear mixed model fit by maximum likelihood (Laplace  
## Approximation) [glmerMod]  
## Family: binomial ( logit )  
## Formula: Presence ~ DOY + (1 | spp)  
## Data: Salamanders  
##   
## AIC BIC logLik deviance df.resid   
## 835.8 849.2 -414.9 829.8 641   
##   
## Scaled residuals:   
## Min 1Q Median 3Q Max   
## -1.1802 -0.9066 -0.5072 1.0167 2.3645   
##   
## Random effects:  
## Groups Name Variance Std.Dev.  
## spp (Intercept) 0.3691 0.6076   
## Number of obs: 644, groups: spp, 7  
##   
## Fixed effects:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) -0.45926 0.24503 -1.874 0.0609 .  
## DOY -0.16646 0.08433 -1.974 0.0484 \*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Correlation of Fixed Effects:  
## (Intr)  
## DOY 0.011

ranef(sm1)$spp

## (Intercept)  
## GP 0.09401095  
## PR -1.04578323  
## DM 0.37141759  
## EC-A -0.65876778  
## EC-L 0.37141759  
## DES-L 0.52802873  
## DF 0.37141759

# Plotting each intercept  
DOY.vals <- seq(min(Salamanders$DOY), max(Salamanders$DOY), length.out = 100)  
re\_list1 <- list()  
for(i in 1:length(unique(Salamanders$spp))){  
 re\_list1[[i]] <- data.frame(spp = rep(unique(Salamanders$spp)[i], 100), DOY = DOY.vals,   
 int = rep(ranef(sm1)$spp[i,1], 100),   
 slope = rep(fixef(sm1)[2], 100))  
}  
re\_dat1 <- do.call("rbind", re\_list1)  
re\_dat1$pred <- plogis(re\_dat1$int + re\_dat1$slope\*re\_dat1$DOY) # y = B0 + B1x  
  
re\_mod1 <- data.frame(DOY = DOY.vals, pred = plogis(fixef(sm1)[1] + fixef(sm1)[2]\*DOY.vals))  
   
ggplot(re\_dat1, aes(x = DOY, y = pred, col = spp)) + geom\_line() +  
 geom\_line(re\_mod1, mapping = aes(x = DOY, y = pred, col = NA), col = "black", size = 2) +   
 scale\_y\_continuous(limits = c(0, 1))



Second, we will run a random-slope model (note that this is often not a sensible model) that allows the model estimate for the slope only to come from a second distribution that estimates a mean and variance from slopes from each spp.

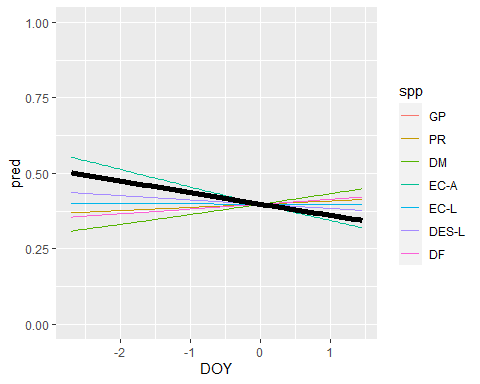
sm2 <- glmer(Presence ~ DOY + (0 + DOY | spp), family = binomial, data = Salamanders)  
summary(sm2)

## Generalized linear mixed model fit by maximum likelihood (Laplace  
## Approximation) [glmerMod]  
## Family: binomial ( logit )  
## Formula: Presence ~ DOY + (0 + DOY | spp)  
## Data: Salamanders  
##   
## AIC BIC logLik deviance df.resid   
## 867.7 881.1 -430.8 861.7 641   
##   
## Scaled residuals:   
## Min 1Q Median 3Q Max   
## -1.0880 -0.8211 -0.7503 1.2000 1.4875   
##   
## Random effects:  
## Groups Name Variance Std.Dev.  
## spp DOY 0.03113 0.1764   
## Number of obs: 644, groups: spp, 7  
##   
## Fixed effects:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) -0.4146 0.0811 -5.113 3.18e-07 \*\*\*  
## DOY -0.1570 0.1057 -1.486 0.137   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Correlation of Fixed Effects:  
## (Intr)  
## DOY 0.024

ranef(sm2)$spp

## DOY  
## GP 0.044387593  
## PR 0.044354882  
## DM 0.143815374  
## EC-A -0.232693717  
## EC-L -0.005089487  
## DES-L -0.058581025  
## DF 0.067178689

# Plotting each intercept  
re\_list2 <- list()  
for(i in 1:length(unique(Salamanders$spp))){  
 re\_list2[[i]] <- data.frame(spp = rep(unique(Salamanders$spp)[i], 100), DOY = DOY.vals,   
 int = rep(fixef(sm2)[1], 100),   
 slope = rep(ranef(sm2)$spp[i,1], 100))  
}  
  
re\_dat2 <- do.call("rbind", re\_list2)  
re\_dat2$pred <- plogis(re\_dat2$int + re\_dat2$slope\*re\_dat2$DOY)  
  
re\_mod2 <- data.frame(DOY = DOY.vals, pred = plogis(fixef(sm2)[1] + fixef(sm2)[2]\*DOY.vals))  
  
ggplot(re\_dat2, aes(x = DOY, y = pred, col = spp)) + geom\_line() +  
 geom\_line(re\_mod2, mapping = aes(x = DOY, y = pred, col = NA), col = "black", size = 2) +   
 scale\_y\_continuous(limits = c(0, 1))



Third, we will run a random-intercept and -slope model that allows the model estimates for both intercept and slopes to come from another distribution that estimates a mean and variance from intercepts and slopes from each spp.

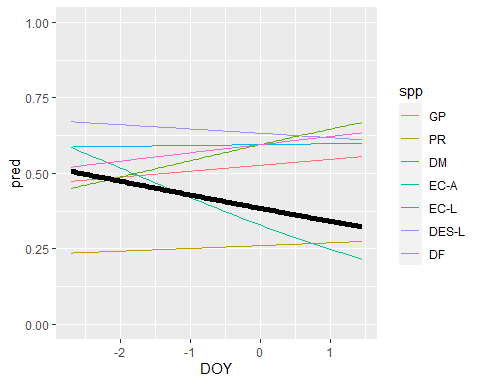
sm3 <- glmer(Presence ~ DOY + (1 | spp) + (0 + DOY | spp), family = binomial, data = Salamanders)  
summary(sm3)

## Generalized linear mixed model fit by maximum likelihood (Laplace  
## Approximation) [glmerMod]  
## Family: binomial ( logit )  
## Formula: Presence ~ DOY + (1 | spp) + (0 + DOY | spp)  
## Data: Salamanders  
##   
## AIC BIC logLik deviance df.resid   
## 835.6 853.5 -413.8 827.6 640   
##   
## Scaled residuals:   
## Min 1Q Median 3Q Max   
## -1.2584 -0.9054 -0.4650 1.0375 2.3975   
##   
## Random effects:  
## Groups Name Variance Std.Dev.  
## spp (Intercept) 0.39184 0.6260   
## spp.1 DOY 0.06065 0.2463   
## Number of obs: 644, groups: spp, 7  
##   
## Fixed effects:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) -0.4724 0.2521 -1.874 0.061 .  
## DOY -0.1840 0.1275 -1.443 0.149   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Correlation of Fixed Effects:  
## (Intr)  
## DOY 0.018

ranef(sm3)$spp

## (Intercept) DOY  
## GP 0.1076772 0.07848642  
## PR -1.0423741 0.04858182  
## DM 0.3860041 0.21715918  
## EC-A -0.7159298 -0.39153518  
## EC-L 0.3854927 0.01204268  
## DES-L 0.5436764 -0.06212241  
## DF 0.3859276 0.11170560

# Plotting each intercept  
re\_list3 <- list()  
for(i in 1:length(unique(Salamanders$spp))){  
 re\_list3[[i]] <- data.frame(spp = rep(unique(Salamanders$spp)[i], 100), DOY = DOY.vals,   
 int = rep(ranef(sm3)$spp[i,1], 100),   
 slope = rep(ranef(sm3)$spp[i,2], 100))  
}  
re\_dat3 <- do.call("rbind", re\_list3)  
re\_dat3$pred <- plogis(re\_dat3$int + re\_dat3$slope\*re\_dat3$DOY)  
  
re\_mod3 <- data.frame(DOY = DOY.vals, pred = plogis(fixef(sm3)[1] + fixef(sm3)[2]\*DOY.vals))  
  
ggplot(re\_dat3, aes(x = DOY, y = pred, col = spp)) + geom\_line() +  
 geom\_line(re\_mod3, mapping = aes(x = DOY, y = pred, col = NA), col = "black", size = 2) +   
 scale\_y\_continuous(limits = c(0, 1))



Finally, we will run a random-intercept and -slope model with a correlation between the coefficients that allows the model estimates for both intercept and slopes to come from another distribution that estimates a mean and variance from intercepts and slopes from each spp.

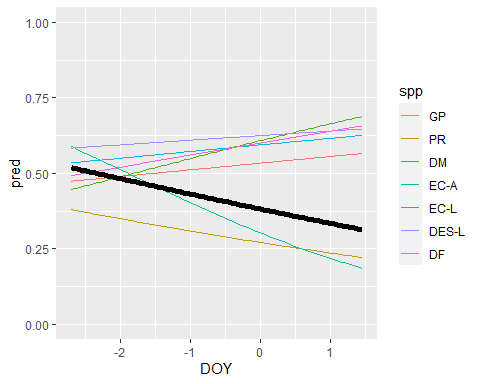
sm4 <- glmer(Presence ~ DOY + (1 + DOY | spp), family = binomial, data = Salamanders)  
summary(sm4)

## Generalized linear mixed model fit by maximum likelihood (Laplace  
## Approximation) [glmerMod]  
## Family: binomial ( logit )  
## Formula: Presence ~ DOY + (1 + DOY | spp)  
## Data: Salamanders  
##   
## AIC BIC logLik deviance df.resid   
## 835.5 857.8 -412.7 825.5 639   
##   
## Scaled residuals:   
## Min 1Q Median 3Q Max   
## -1.1351 -0.9185 -0.4679 1.0250 2.6872   
##   
## Random effects:  
## Groups Name Variance Std.Dev. Corr  
## spp (Intercept) 0.40819 0.6389   
## DOY 0.07042 0.2654 0.76  
## Number of obs: 644, groups: spp, 7  
##   
## Fixed effects:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) -0.4815 0.2570 -1.874 0.061 .  
## DOY -0.2058 0.1337 -1.539 0.124   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Correlation of Fixed Effects:  
## (Intr)  
## DOY 0.567

ranef(sm4)$spp

## (Intercept) DOY  
## GP 0.1360998 0.08876791  
## PR -0.9893844 -0.18413320  
## DM 0.4395231 0.24221032  
## EC-A -0.8342862 -0.43967336  
## EC-L 0.3821675 0.09006816  
## DES-L 0.5113961 0.06423783  
## DF 0.4101393 0.16394280

# Plotting each intercept and slope  
re\_list4 <- list()  
for(i in 1:length(unique(Salamanders$spp))){  
 re\_list4[[i]] <- data.frame(spp = rep(unique(Salamanders$spp)[i], 100),   
 DOY = DOY.vals,   
 int = rep(ranef(sm4)$spp[i,1], 100),   
 slope = rep(ranef(sm4)$spp[i,2], 100))  
}  
re\_dat4 <- do.call("rbind", re\_list4)  
re\_dat4$pred <- plogis(re\_dat4$int + re\_dat4$slope\*re\_dat4$DOY)  
  
re\_mod4 <- data.frame(DOY = DOY.vals, pred = plogis(fixef(sm4)[1] + fixef(sm4)[2]\*DOY.vals))  
  
ggplot(re\_dat4, aes(x = DOY, y = pred, col = spp)) + geom\_line() +  
 geom\_line(re\_mod4, mapping = aes(x = DOY, y = pred, col = NA), col = "black", size = 2) +   
 scale\_y\_continuous(limits = c(0, 1))



To evaluate which random-coefficient model wins, we can use AIC values to rank these models.

ICtab(sm0, sm1, sm2, sm3, sm4, base = T) # random-intercept model

## AIC dAIC df  
## sm4 835.5 0.0 5   
## sm3 835.6 0.1 4   
## sm1 835.8 0.3 3   
## sm0 866.7 31.2 2   
## sm2 867.7 32.2 3

**Practice example 11:** For this practice example, we will be using the built-in dataset on grouse ticks from the lme4 package (see ?grouseticks). This dataset contains counts of ticks (TICKS) on the heads of red grouse chicks sampled in the field from different broods (BROOD).

attach(grouseticks)  
head(grouseticks)

## INDEX TICKS BROOD HEIGHT YEAR LOCATION cHEIGHT  
## 1 1 0 501 465 95 32 2.759305  
## 2 2 0 501 465 95 32 2.759305  
## 3 3 0 502 472 95 36 9.759305  
## 4 4 0 503 475 95 37 12.759305  
## 5 5 0 503 475 95 37 12.759305  
## 6 6 3 503 475 95 37 12.759305

We want you to evaluate the effects of height above sea level (using “cHEIGHT”) on the number of ticks on Red Grouse chicks (TICKS) *conditioned on having ticks* using random-intercept model for brood number. Since this is a conditional vital rate, we assume that all ticks are detectable and that these data can be subseted for values greater than 0. We expect you to evaluate the correct error distribution for these data, and then find the winning model using model selection.

#### Nested random effects

Nested random effects are when each sample of one group is contained entirely within a single unit of another group. The common example for nested groups would be that students who are assigned to a particular classroom are nested within that classroom.

Here, we know that each of the 56 sites are nested within the eight regions of Switzerland that were censused for brown hares from 1992 to 2008. Here, we will run a nested random-intercept model.

dens\_re1 <- lmer(log(mean.density) ~ yearPost + (1 | region/site), data = hares)  
summary(dens\_re1)

## Linear mixed model fit by REML ['lmerMod']  
## Formula: log(mean.density) ~ yearPost + (1 | region/site)  
## Data: hares  
##   
## REML criterion at convergence: 1123.8  
##   
## Scaled residuals:   
## Min 1Q Median 3Q Max   
## -4.1577 -0.5086 0.0309 0.6289 3.5523   
##   
## Random effects:  
## Groups Name Variance Std.Dev.  
## site:region (Intercept) 0.59981 0.7745   
## region (Intercept) 0.06682 0.2585   
## Residual 0.22715 0.4766   
## Number of obs: 677, groups: site:region, 56; region, 8  
##   
## Fixed effects:  
## Estimate Std. Error t value  
## (Intercept) 1.167050 0.149353 7.814  
## yearPost -0.005867 0.004152 -1.413  
##   
## Correlation of Fixed Effects:  
## (Intr)  
## yearPost -0.254

ranef(dens\_re1)

## $`site:region`  
## (Intercept)  
## AG01:CH.Central -0.05842063  
## AG02:CH.Central 0.08364256  
## AG03:CH.Central 0.16960042  
## AG04:CH.Central -1.19459514  
## BE01:Aare -0.26042731  
## BE02:Aare -0.33709338  
## BE03:Aare -1.07550250  
## BE04:Aare -0.17830184  
## BE05:Aare -0.74192031  
## BE06:Aare 0.14157600  
## BE08:CH.W -0.26247600  
## BE09:CH.W 0.75464018  
## BE12:CH.W 1.62986620  
## BE19:Aare 0.57287272  
## BE20:Aare 0.62564613  
## BE23:Aare -0.91888334  
## BE24:Aare 0.22038043  
## BE25:Aare 0.30249139  
## BE27:Aare -0.09835289  
## BE28:Aare -0.66850156  
## BL02:Baselland 0.06762409  
## BL04:Baselland 0.57641701  
## BL05:Baselland -1.42193756  
## BR02:CH.W -0.52054755  
## BR03:CH.W -1.47531895  
## BR04:CH.W -0.42467699  
## GE01:CH.SW 0.97195119  
## GE02:CH.SW 0.75035374  
## GE03:CH.SW 1.02575712  
## LI04:CH.E -0.67315032  
## LI05:CH.E -1.08869201  
## LI15:CH.E -0.37780363  
## LU01:CH.Central 0.08389330  
## LU07A:CH.Central -0.23928683  
## LU07B:CH.Central -0.21652982  
## SG06:CH.E 0.58641192  
## SG07:CH.E 1.04520584  
## SG08:CH.E 0.09202055  
## SG09:CH.E 0.40455499  
## SH03:CH.N 0.20469902  
## SH04:CH.N 0.64147779  
## SH05:CH.N -0.71522068  
## SH07:CH.N 0.55965618  
## SH1\_2:CH.N 0.27298231  
## SO01:Aare 0.94005670  
## SO02:Aare 1.20524487  
## SO03:Aare 0.66010019  
## TG06B:CH.E 0.17929602  
## TG07:CH.E -1.06998388  
## VD01:CH.SW 0.07485527  
## VD02:CH.W -0.49424157  
## VD04:CH.W 0.28888268  
## VD05:Rhone 0.21580685  
## VS02:Rhone 0.69417389  
## VS04:Rhone -1.99148371  
## ZH06:CH.N 0.46121089  
##   
## $region  
## (Intercept)  
## Aare 0.04337776  
## Baselland -0.08665815  
## CH.Central -0.15280781  
## CH.E -0.10049902  
## CH.N 0.15872422  
## CH.SW 0.31447475  
## CH.W -0.05613165  
## Rhone -0.12048011  
##   
## with conditional variances for "site:region" "region"

dens\_re2 <- lmer(log(mean.density) ~ yearPost + (1 | region) + (1|region:site), data = hares) # same model  
ranef(dens\_re2)

## $`region:site`  
## (Intercept)  
## Aare:BE01 -0.26042731  
## Aare:BE02 -0.33709338  
## Aare:BE03 -1.07550250  
## Aare:BE04 -0.17830184  
## Aare:BE05 -0.74192031  
## Aare:BE06 0.14157600  
## Aare:BE19 0.57287272  
## Aare:BE20 0.62564613  
## Aare:BE23 -0.91888334  
## Aare:BE24 0.22038043  
## Aare:BE25 0.30249139  
## Aare:BE27 -0.09835289  
## Aare:BE28 -0.66850156  
## Aare:SO01 0.94005670  
## Aare:SO02 1.20524487  
## Aare:SO03 0.66010019  
## Baselland:BL02 0.06762409  
## Baselland:BL04 0.57641701  
## Baselland:BL05 -1.42193756  
## CH.Central:AG01 -0.05842063  
## CH.Central:AG02 0.08364256  
## CH.Central:AG03 0.16960042  
## CH.Central:AG04 -1.19459514  
## CH.Central:LU01 0.08389330  
## CH.Central:LU07A -0.23928683  
## CH.Central:LU07B -0.21652982  
## CH.E:LI04 -0.67315032  
## CH.E:LI05 -1.08869201  
## CH.E:LI15 -0.37780363  
## CH.E:SG06 0.58641192  
## CH.E:SG07 1.04520584  
## CH.E:SG08 0.09202055  
## CH.E:SG09 0.40455499  
## CH.E:TG06B 0.17929602  
## CH.E:TG07 -1.06998388  
## CH.N:SH03 0.20469902  
## CH.N:SH04 0.64147779  
## CH.N:SH05 -0.71522068  
## CH.N:SH07 0.55965618  
## CH.N:SH1\_2 0.27298231  
## CH.N:ZH06 0.46121089  
## CH.SW:GE01 0.97195119  
## CH.SW:GE02 0.75035374  
## CH.SW:GE03 1.02575712  
## CH.SW:VD01 0.07485527  
## CH.W:BE08 -0.26247600  
## CH.W:BE09 0.75464018  
## CH.W:BE12 1.62986620  
## CH.W:BR02 -0.52054755  
## CH.W:BR03 -1.47531895  
## CH.W:BR04 -0.42467699  
## CH.W:VD02 -0.49424157  
## CH.W:VD04 0.28888268  
## Rhone:VD05 0.21580685  
## Rhone:VS02 0.69417389  
## Rhone:VS04 -1.99148371  
##   
## $region  
## (Intercept)  
## Aare 0.04337776  
## Baselland -0.08665815  
## CH.Central -0.15280781  
## CH.E -0.10049902  
## CH.N 0.15872422  
## CH.SW 0.31447475  
## CH.W -0.05613165  
## Rhone -0.12048011  
##   
## with conditional variances for "region:site" "region"

**Practice example 12:** For the grasshopper dataset used in example #2, re-run the two-way ANOVA with interaction of Origin and Treatment on male grasshopper song with nested random-intercept effects of site and individual. Using AIC, compare this model with no random effects and only random-intercept effects of Site to find the winning model.

gh1 <- glmmTMB(LocMax ~ Origin\*Treatment, family = gaussian, data = ghp)

What does this winning model suggest about the variance in male grasshopper songs across sites and individuals?

### 6. Introduction to simulating data

Here, this is a very brief introduction to simulating data from three types of distributions: the normal distribution, binomial distribution, and Poisson distribution.

I will introduce the density function (e.g., dnorm()), distribution function (e.g., pnorm()), quantile function (e.g., qnorm()) and random generation function (e.g., rnorm()) for these distributions.

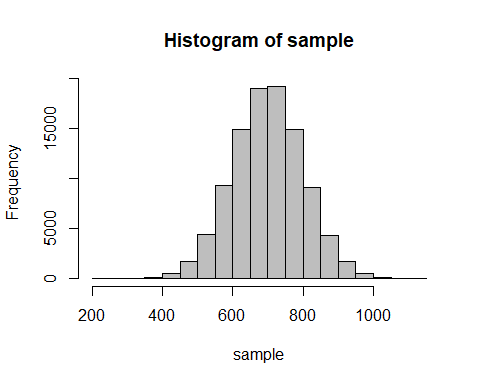
#### Normal distribution

First, we will sample from the normal distribution

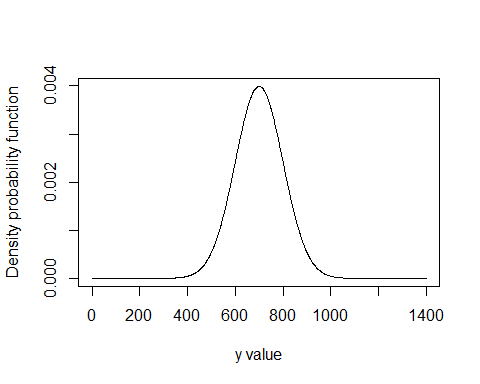
set.seed(1)  
  
n <- 100000 # sample size  
mu <- 700 # mean body mass of male pelegrines   
sd <- 100 # SD of body size  
  
sample <- rnorm(n = n, mean = mu, sd = sd)  
head(sample) # vector of randomly generated numbers

## [1] 637.3546 718.3643 616.4371 859.5281 732.9508 617.9532

par(mfrow = c(1,1))  
hist(sample, col = "grey")



plot(0:1400, dnorm(x = 0:1400, mean = mu, sd = sd), type = "l",   
 ylab = "Density probability function", xlab = "y value")



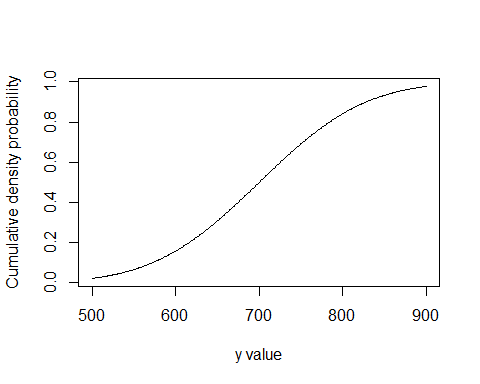
sum(dnorm(x = 0:1400, mean = mu, sd = sd))

## [1] 1

pnorm(q = 650, mean = mu, sd = sd, lower.tail = T) # cumulative density function

## [1] 0.3085375

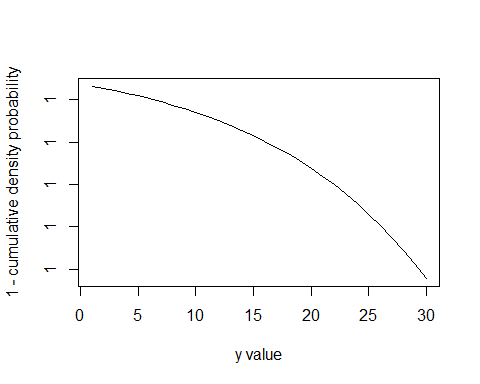
plot(500:900, pnorm(q = 500:900, mean = mu, sd = sd), type = "l",   
 ylab = "Cumulative density probability", xlab = "y value")



pnorm(q = 650, mean = mu, sd = sd, lower.tail = F)

## [1] 0.6914625

plot(x, pnorm(q = x, mean = mu, sd = sd, lower.tail = F), type = "l",   
 ylab = "1 - cumulative density probability", xlab = "y value")



qnorm(p = 0.90, mean = mu, sd = sd) # what is the 90% percentile of male pelegrine masses?

## [1] 828.1552

qnorm(p = 0.10, mean = mu, sd = sd) # what is the 10% percentile of male pelegrine masses?

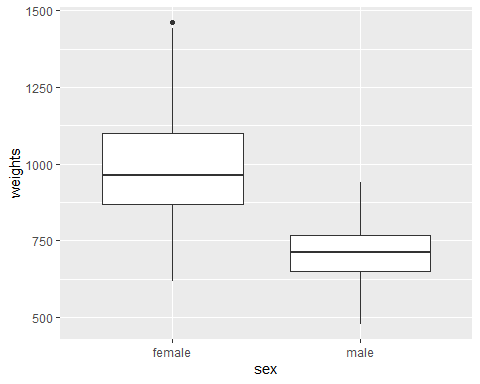
## [1] 571.8448

qnorm(p = 0.90, mean = mu, sd = sd, lower.tail = F)

## [1] 571.8448

Now, we will simulate two linear models: ANOVA and linear regression.

set.seed(1)  
  
# Differences between means (e.g., ANOVA)  
b0 = 700 # mean male pelegrine  
b1 = 300 # females weigh approximate 1 kg, so a difference of 300 g from males.   
male.sd <- 100  
fem.sd <- 200  
  
sex <- rep(c("male", "female"), each = 100)  
  
eps <- c(rnorm(n = 100, mean = 0, sd = male.sd), # male sd  
 rnorm(n = 100, mean = 0, sd = fem.sd)) # female sd  
weights <- b0 + b1\*(sex == "female") + eps  
  
bird.dat <- data.frame(sex, weights)  
ggplot(bird.dat, aes(x = sex, y = weights)) + geom\_boxplot()



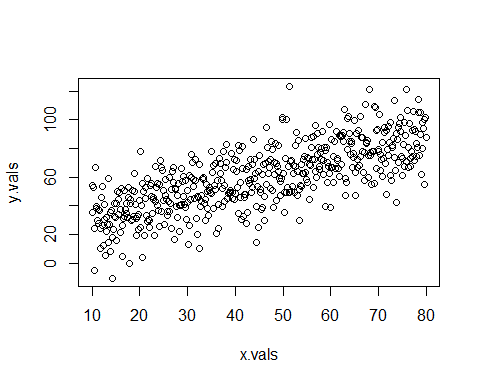
bird.lm <- lm(weights ~ sex, data = bird.dat)  
summary(bird.lm) # same se of the mean

##   
## Call:  
## lm(formula = weights ~ sex, data = bird.dat)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -375.31 -82.60 -11.63 72.73 469.16   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 992.44 14.96 66.33 <2e-16 \*\*\*  
## sexmale -281.55 21.16 -13.31 <2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 149.6 on 198 degrees of freedom  
## Multiple R-squared: 0.4721, Adjusted R-squared: 0.4694   
## F-statistic: 177.1 on 1 and 198 DF, p-value: < 2.2e-16

coef(bird.lm)

## (Intercept) sexmale   
## 992.4384 -281.5496

# Linear regression  
nB1 <- 20 # intercept  
nB0 <- 0.9 # slope  
  
x.vals <- seq(10, 80, length.out = 500)   
error <- rnorm(500, mean = 0, sd = 15)  
  
y.vals <- nB1 + nB0 \* x.vals + error  
plot(x.vals, y.vals)



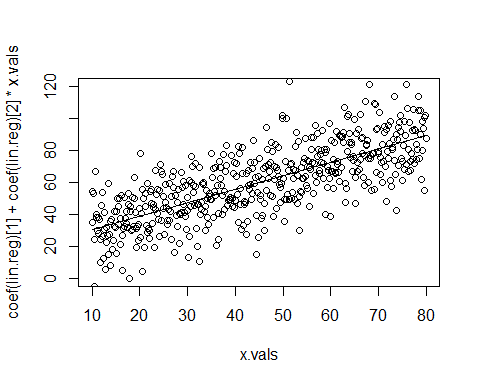
lin.reg <- lm(y.vals ~ x.vals)  
Anova(lin.reg)

## Anova Table (Type II tests)  
##   
## Response: y.vals  
## Sum Sq Df F value Pr(>F)   
## x.vals 145198 1 586.95 < 2.2e-16 \*\*\*  
## Residuals 123194 498   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

coef(lin.reg) # very close to our defined intercept and slope

## (Intercept) x.vals   
## 22.0209274 0.8416266

plot(x.vals, coef(lin.reg)[1] + coef(lin.reg)[2]\*x.vals, type = "l", ylim = c(0, 120))  
points(x.vals, y.vals)



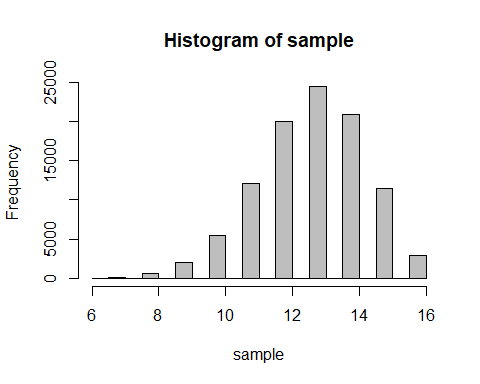
#### Binomial distribution

Second, we will learn to sample from the binomial distribution.

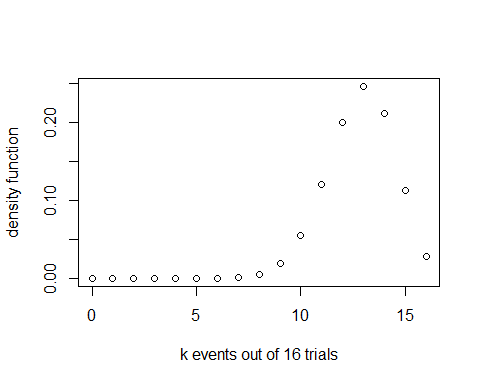
n <- 100000 # Sample size  
N <- 16 # Number of individuals  
p <- 0.8 # Probability of surviving  
sample <- rbinom(n = n, size = N, prob = p)  
head(sample)

## [1] 14 12 9 13 13 11

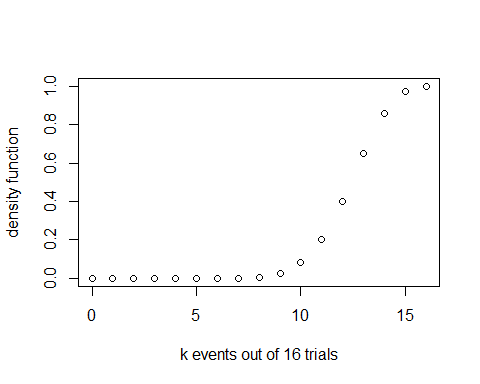
hist(sample, col = "grey")



plot(0:16, dbinom(x = 0:16, size = N, prob = p), ylab = "density function", xlab = "k events out of 16 trials")



plot(0:16, pbinom(q = 0:16, size = N, prob = p), ylab = "density function", xlab = "k events out of 16 trials")



pbinom(q = 11, size = N, prob = p) # probability of getting 11 or less success out of 16 trials

## [1] 0.2017546

pbinom(q = 11, size = N, prob = p, lower.tail = F) # probability of getting 11 or more

## [1] 0.7982454

qbinom(p = 0.9, size = N, prob = p) # 90% percentile

## [1] 15

qbinom(p = 0.1, size = N, prob = p) # 10% percentile

## [1] 11

Here, we will simulating binomial models.

set.seed(1)  
  
# Difference between means  
bin.B0 <- qlogis(0.8) # survival in control group on logit-scale is approx. 0.8  
bin.B1 <- qlogis(0.2) - qlogis(0.8) # survival in treatment group on logit-scale is 0.2  
  
treatment <- sample(c("control","spray"), size = 600, replace = TRUE)  
bin.y <- bin.B0 + bin.B1\*(treatment == "spray")  
  
p <- plogis(bin.y) # back-transformed probabilities for each 100 observations  
summary(p)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 0.200 0.200 0.800 0.508 0.800 0.800

y <- rbinom(n = 600, size = 1, prob = p) # size of trial is 1   
bin.means <- glm(y ~ treatment, family = binomial)  
coef(bin.means)

## (Intercept) treatmentspray   
## 1.504077 -2.774540

plogis(coef(bin.means)[1])

## (Intercept)   
## 0.8181818

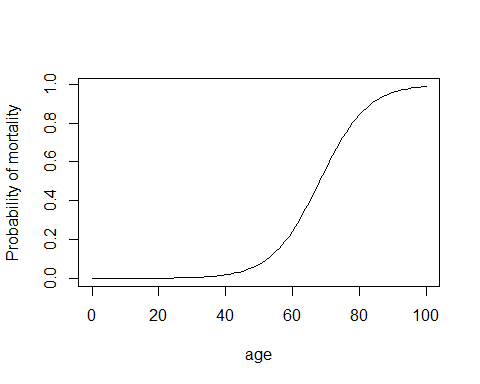
plogis(sum(coef(bin.means)))

## [1] 0.2191781

# Binomial regression  
binr.B0 <- -9  
binr.B1 <- 0.13  
  
age <- runif(1000, min = 0, max = 100)  
bin.probs <- plogis(binr.B0 + binr.B1\*age)  
  
br.y <- rbinom(n = 1000, size = 1, prob = bin.probs) # size of trial is 1   
bin.reg <- glm(br.y ~ age, family = binomial)  
coef(bin.reg)

## (Intercept) age   
## -9.5998542 0.1409979

plot(0:100, plogis(coef(bin.reg)[1] + coef(bin.reg)[2]\*0:100), type = "l",  
 ylab = "Probability of mortality", xlab = "age")



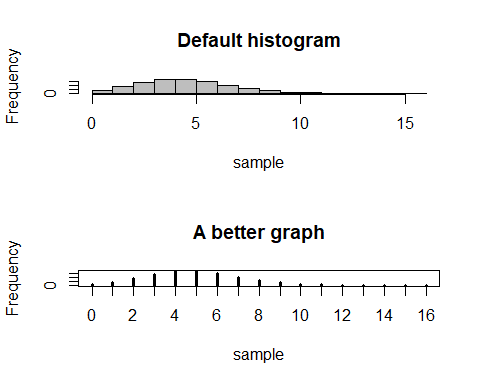
#### Poisson distribution

Third, we will learn how to sample from the Poisson distribution.

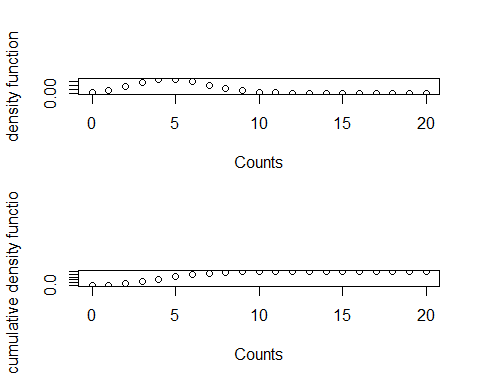
n <- 100000 # Sample size  
lambda <- 5 # Average # individuals per sample  
sample <- rpois(n = n, lambda = lambda)  
head(sample)

## [1] 6 12 5 9 3 5

par(mfrow = c(2,1))  
hist(sample, col = "grey", main = "Default histogram")  
plot(table(sample), main = "A better graph", lwd = 3, ylab = "Frequency")



plot(0:20, dpois(x = 0:20, lambda = 5), ylab = "density function", xlab = "Counts")  
  
plot(0:20, ppois(q = 0:20, lambda = 5), ylab = "cumulative density function", xlab = "Counts")



ppois(10, lambda = 5, lower.tail = T) # probability of getting a value 10 or less

## [1] 0.9863047

ppois(10, lambda = 5, lower.tail = F) # probability of getting a value 10 or more

## [1] 0.01369527

qpois(0.9, lambda = 5) # the 90% percentile is 8

## [1] 8

qpois(0.1, lambda = 5) # the 10% percentile is 2

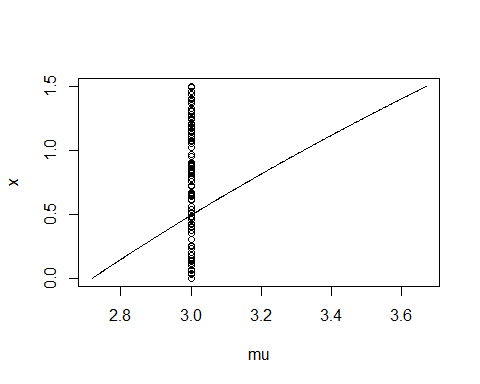
## [1] 2

Finally, we will learn to simulate Poisson models.

set.seed(1)  
  
# Differences between two Poisson means  
beta0 <- log(7) # mean of 7  
beta1 <- log(7 - 3) # mean of 10  
  
site <- sample(c("1","2"), size = 500, replace = TRUE)  
y\_sim <- rpois(n = 500, lambda = exp(beta0 + beta1 \* (site == "2")))  
  
pois.mod <- glm(y\_sim ~ site, family = poisson)  
Anova(pois.mod)

## Analysis of Deviance Table (Type II tests)  
##   
## Response: y\_sim  
## LR Chisq Df Pr(>Chisq)   
## site 3466.6 1 < 2.2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

# Poisson regression   
beta0 <- 1  
beta1 <- 0.2  
  
x <- seq(0, 1.5, length.out = 500) # ran  
  
mu <- exp(beta0 + beta1 \* x) # line of best fit  
y <- rpois(n=500, lambda=mu)  
plot(mu, x, type = "l")  
points(y, x)



pois.reg <- glm(y ~ x, family = poisson)  
Anova(pois.reg)

## Analysis of Deviance Table (Type II tests)  
##   
## Response: y  
## LR Chisq Df Pr(>Chisq)   
## x 14.341 1 0.0001525 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

summary(pois.reg)

##   
## Call:  
## glm(formula = y ~ x, family = poisson)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -2.7435 -0.8269 -0.0903 0.6515 2.9346   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) 0.99886 0.05199 19.212 < 2e-16 \*\*\*  
## x 0.21809 0.05767 3.782 0.000156 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for poisson family taken to be 1)  
##   
## Null deviance: 596.11 on 499 degrees of freedom  
## Residual deviance: 581.76 on 498 degrees of freedom  
## AIC: 1999.8  
##   
## Number of Fisher Scoring iterations: 5

The following distributions have their relative random generation functions:

* Bernoulli distribution, rbernoulli()
* Negative binomial distribution, rnbinom()
* Uniform distribution, runif()
* Gamma distribution, rgamma()
* Conway-Maxwell-Poisson distribution, mpcmp::rcomp() in package mpcmp
* Beta distribution, rbeta()
* Exponential distribution, rexp()

## Other frequentist models

This is just the surface of the many frequentist models you could use to analyze your data. For example, we did not delve into the world of multivariate statistics for understanding which set of response variables explains the most variation in your data, such as Principal Component Analysis (PCA), Linear discriminant Analysis (LCA), or Partial Least Square Regression (PLSR).

Two good resources for multivariate statistics would be “*A Primer of Ecological Statistics*” by Gotelli and Ellison (Ch 12) and the first chapter of “*A Beginner’s Guide to Generalized Additive Models with R*” by Zuur, which is also a good introduction to GAMs where your linear predictor depends on some smooth function (rather than parametric function) of some covariate.

If you want to explore simulating multivariate data, [Dr. Eric Scott](https://www.ericrscott.com/) developed a R package called [Holodeck](https://github.com/Aariq/holodeck) for simulating multivariate data.

## Sources

Here are a list of additional resources for statistics behind these models and more example R code:

* Bolker, B. (2008) *Ecological models and data in R.* Princeton University Press, Princeton, New Jersey, USA. (<https://ms.mcmaster.ca/~bolker/emdbook/>)
* Burnham, K. P. and Anderson, D. R. (2002) *Model Selection and Inference: A Practical Information-Theoretic Approach.* Second Edition. Springer, New York, New York, USA.
* Gotelli, N.J. and Ellison, A.M. (2012) *A Primer for Ecological Statistics*. Second edition, Sinauer Associates Publishers.
* Kery, M. (2010) *Introduction to WinBUGS for Ecologists.* Academic Press, Burlinton, Massachusetts, USA.
* White G. C. , and Cooch E. (2005). *Program MARK: A Gentle Introduction.* 4th edn. (<http://www.phidot.org/software/mark/docs/book/>.)
* Zuur, A.F, E.N. Ieno, and E. Meesters. (2009) *A Beginner’s Guide to R.* Springer, New York, New York, USA.
* Zuur, A.F. (2012) *A Beginner’s guide to Generalized Additive Models with R*. Highland Statistics Ltd, Newburgh, NY.