Session 2: Intro to frequentist models in R

Natalie Z. Kerr & Brian Lovett

October 11, 2020

Contents

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 likelihood ratio tests for nested models and Akaike Information Criteria (AIC). 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 hurdle models
library(lme4) # for mixed models

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 five 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

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

1. Intro to linear models

First, we will cover running linear models (LMs) for normally-distributed (or Gaussian distributed) data. 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, and analysis of covariance (ANCOVA).

We will use Brown hare (*Lepus europaeus*) data over 17 years (1992-2008) at 56 sites in 8 regions of Switzerland for most of our examples today. 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 that contains examples of both R and WinBUGS code.

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

```
##   no site     region    site2 area elevation landuse year count1 count2
## 1  1 AG01 CH.Central Reusstal 2.23       384  arable 1992     NA     NA
## 2  2 AG01 CH.Central Reusstal 2.23       384  arable 1993     NA     NA
## 3  3 AG01 CH.Central Reusstal 2.23       384  arable 1994     NA     NA
## 4  4 AG01 CH.Central Reusstal 2.23       384  arable 1995      6      4
## 5  5 AG01 CH.Central Reusstal 2.23       384  arable 1996      7      5
## 6  6 AG01 CH.Central Reusstal 2.23       384  arable 1997      3      3
##   mean.density
## 1           NA
## 2           NA
## 3           NA
## 4     2.690583
## 5     3.139013
## 6     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.

```r
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.

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

```
## [1] 4.079368e-16
```

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

```
## [1] 4.736569
```

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

```
## [1] 0.1466952
```

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

```
## [1] 0.8058354
```

One-way ANOVA

Here, 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.

```r
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
```

```r
Anova(dens2) # analysis of variance table, not to be confused with the model
```

```
## Anova Table (Type II tests)
##
## Response: mean.density
##           Sum Sq  Df F value    Pr(>F)
## landuse    623.7   1   45.64 3.072e-11 ***
## Residuals 9224.7 675
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The lm() summary also contains two more outputs:

- *Multiple R-squared:* determined how well your model fits your data (aka coefficient of determination), it subtracts the residual error (i.e., variance in the predictor) from the variance in Y (i.e., variance in the response). Here, only 6.3% of the landuse explains the variance in mean density of hares.

- *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.

Here, we used what is referred to as "effect parameterization". The dummy variable (or *Intercept*, β_0) is arable land, and the *landusegrass* is difference to grassland relative to the dummy variable (or *slope*, β_1). The mean density in the arable land is 5.3317186 and the mean density in the grassland is 3.1885457, i.e., $y = \beta_0 + \beta_1 x$ where x is either "1" for grassland or "0" for arable.

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

```
##      no site region site2 area elevation landuse year count1 count2
## 105 105 BE03   Aare  <NA> 1.12       557   grass 1994      1     NA
## 106 106 BE03   Aare  <NA> 1.12       557   grass 1995      1      1
## 109 109 BE03   Aare  <NA> 1.12       557   grass 1998      1      1
## 111 111 BE03   Aare  <NA> 1.12       557   grass 2000      2      1
## 120 120 BE04   Aare  <NA> 2.77       532  arable 1992      5      7
## 121 121 BE04   Aare  <NA> 2.77       532  arable 1993      8     10
## 122 122 BE04   Aare  <NA> 2.77       532  arable 1994      1      6
## 123 123 BE04   Aare  <NA> 2.77       532  arable 1995      9      8
## 126 126 BE04   Aare  <NA> 2.77       532  arable 1998      6      8
## 128 128 BE04   Aare  <NA> 2.77       532  arable 2000     21      8
## 133 133 BE04   Aare  <NA> 2.77       532  arable 2005      1      2
##     mean.density
## 105    0.8928571
## 106    0.8928571
## 109    0.8928571
## 111    1.7857143
## 120    2.5270758
## 121    3.6101083
## 122    2.1660650
## 123    3.2490975
## 126    2.8880866
## 128    7.5812274
## 133    0.7220217
```

```
model.matrix(dens2)[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
```

```
effects.par <- lm(mean.density ~ 1 + landuse, data = hares)
summary(effects.par) # same model as dens2
```

```
##
## Call:
## lm(formula = mean.density ~ 1 + 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
```

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)
summary(means.par) # removes dummy variable
```

```
##
## Call:
## lm(formula = mean.density ~ -1 + 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|)    
## landusearable   5.3317     0.1672   31.89   <2e-16 ***
## landusegrass    3.1885     0.2696   11.83   <2e-16 ***
## ---
## 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:  0.6316, Adjusted R-squared:  0.6305 
## F-statistic: 578.5 on 2 and 675 DF,  p-value: < 2.2e-16
```

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

```
##      no site region site2 area elevation landuse year count1 count2
## 105 105 BE03   Aare  <NA> 1.12       557   grass 1994      1     NA
## 106 106 BE03   Aare  <NA> 1.12       557   grass 1995      1      1
## 109 109 BE03   Aare  <NA> 1.12       557   grass 1998      1      1
## 111 111 BE03   Aare  <NA> 1.12       557   grass 2000      2      1
## 120 120 BE04   Aare  <NA> 2.77       532  arable 1992      5      7
## 121 121 BE04   Aare  <NA> 2.77       532  arable 1993      8     10
## 122 122 BE04   Aare  <NA> 2.77       532  arable 1994      1      6
## 123 123 BE04   Aare  <NA> 2.77       532  arable 1995      9      8
## 126 126 BE04   Aare  <NA> 2.77       532  arable 1998      6      8
## 128 128 BE04   Aare  <NA> 2.77       532  arable 2000     21      8
## 133 133 BE04   Aare  <NA> 2.77       532  arable 2005      1      2
##     mean.density
## 105    0.8928571
## 106    0.8928571
## 109    0.8928571
## 111    1.7857143
## 120    2.5270758
## 121    3.6101083
## 122    2.1660650
## 123    3.2490975
## 126    2.8880866
## 128    7.5812274
## 133    0.7220217
```

```r
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: Using the built-in “ToothGrowth” R dataset, run an ANOvA to estimate tooth growth for three dosages of vitamin C. Run two models using the effects and means parameterization approach.

> data(“ToothGrowth”) # to load data

ToothGrowth data set contains 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 (a form of vitamin C, VC).

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)
summary(dens4)
```

```
##
## Call:
## lm(formula = mean.density ~ region + landuse, data = hares)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -6.9006 -2.3263 -0.5892  1.5693 16.9740
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)         4.9530     0.2800  17.691  < 2e-16 ***
## regionBaselland    -1.4586     0.6401  -2.279   0.0230 *
## regionCH.Central   -1.9954     0.4585  -4.352 1.56e-05 ***
## regionCH.E         -0.5713     0.4984  -1.146   0.2521
## regionCH.N          0.3032     0.4727   0.641   0.5215
## regionCH.SW         4.6676     0.4951   9.428  < 2e-16 ***
## regionCH.W          0.1661     0.4322   0.384   0.7009
## regionRhone        -1.1703     0.6531  -1.792   0.0736 .
## landusegrass       -0.8074     0.3998  -2.019   0.0439 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.342 on 668 degrees of freedom
##   (275 observations deleted due to missingness)
## Multiple R-squared:  0.2424, Adjusted R-squared:  0.2333
## F-statistic: 26.71 on 8 and 668 DF,  p-value: < 2.2e-16
```

```
Anova(dens4)
```

```
## Anova Table (Type II tests)
##
## Response: mean.density
##           Sum Sq  Df F value  Pr(>F)
## region    1763.2   7 22.5504 < 2e-16 ***
## landuse     45.5   1  4.0778 0.04385 *
## Residuals 7461.5 668
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

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

```
##
## Call:
## lm(formula = mean.density ~ -1 + region + landuse, data = hares)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -6.9006 -2.3263 -0.5892  1.5693 16.9740
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## regionAare          4.9530     0.2800  17.691  < 2e-16 ***
## regionBaselland     3.4944     0.5979   5.845 7.94e-09 ***
## regionCH.Central    2.9576     0.3850   7.682 5.59e-14 ***
## regionCH.E          4.3817     0.4498   9.742  < 2e-16 ***
## regionCH.N          5.2561     0.3809  13.800  < 2e-16 ***
## regionCH.SW         9.6206     0.4083  23.562  < 2e-16 ***
## regionCH.W          5.1190     0.3293  15.545  < 2e-16 ***
## regionRhone         3.7827     0.5973   6.333 4.40e-10 ***
## landusegrass       -0.8074     0.3998  -2.019   0.0439 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.342 on 668 degrees of freedom
##   (275 observations deleted due to missingness)
## Multiple R-squared:  0.702,  Adjusted R-squared:  0.698
## F-statistic: 174.8 on 9 and 668 DF,  p-value: < 2.2e-16
```

Here, we will run a two-way ANOVA to evaluate the interactive effects of both vitamin C dosage and supplement type on tooth growth in guinea pigs.

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

```
##
## Call:
## lm(formula = len ~ supp + factor(dose) + supp:factor(dose), data = ToothGrowth)
##
## Residuals:
##    Min     1Q Median     3Q    Max
##  -8.20  -2.72  -0.27   2.65   8.27
```

```
##
## Coefficients:
##                        Estimate Std. Error t value Pr(>|t|)    
## (Intercept)              13.230      1.148  11.521 3.60e-16 ***
## suppVC                   -5.250      1.624  -3.233  0.00209 ** 
## factor(dose)1             9.470      1.624   5.831 3.18e-07 ***
## factor(dose)2            12.830      1.624   7.900 1.43e-10 ***
## suppVC:factor(dose)1     -0.680      2.297  -0.296  0.76831    
## suppVC:factor(dose)2      5.330      2.297   2.321  0.02411 *  
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.631 on 54 degrees of freedom
## Multiple R-squared:  0.7937, Adjusted R-squared:  0.7746 
## F-statistic: 41.56 on 5 and 54 DF,  p-value: < 2.2e-16
```

```
Anova(grow1)
```

```
## Anova Table (Type II tests)
##
## Response: len
##                    Sum Sq Df F value    Pr(>F)    
## supp               205.35  1  15.572 0.0002312 ***
## factor(dose)      2426.43  2  92.000 < 2.2e-16 ***
## supp:factor(dose)  108.32  2   4.107 0.0218603 *  
## Residuals          712.11 54                      
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

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

```
##
## Call:
## lm(formula = len ~ supp * factor(dose), data = ToothGrowth)
##
## Residuals:
##    Min     1Q Median     3Q    Max 
##  -8.20  -2.72  -0.27   2.65   8.27 
##
## Coefficients:
##                        Estimate Std. Error t value Pr(>|t|)    
## (Intercept)              13.230      1.148  11.521 3.60e-16 ***
## suppVC                 -5.250      1.624  -3.233  0.00209 **
## factor(dose)1           9.470      1.624   5.831 3.18e-07 ***
## factor(dose)2          12.830      1.624   7.900 1.43e-10 ***
## suppVC:factor(dose)1   -0.680      2.297  -0.296  0.76831
## suppVC:factor(dose)2    5.330      2.297   2.321  0.02411 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.631 on 54 degrees of freedom
## Multiple R-squared:  0.7937, Adjusted R-squared:  0.7746
## F-statistic: 41.56 on 5 and 54 DF,  p-value: < 2.2e-16
```

Linear regression

Here, we will use the brown hare dataset to run a linear regression evaluating whether mean density of brown hares changes linearly with year in the grassland sites.

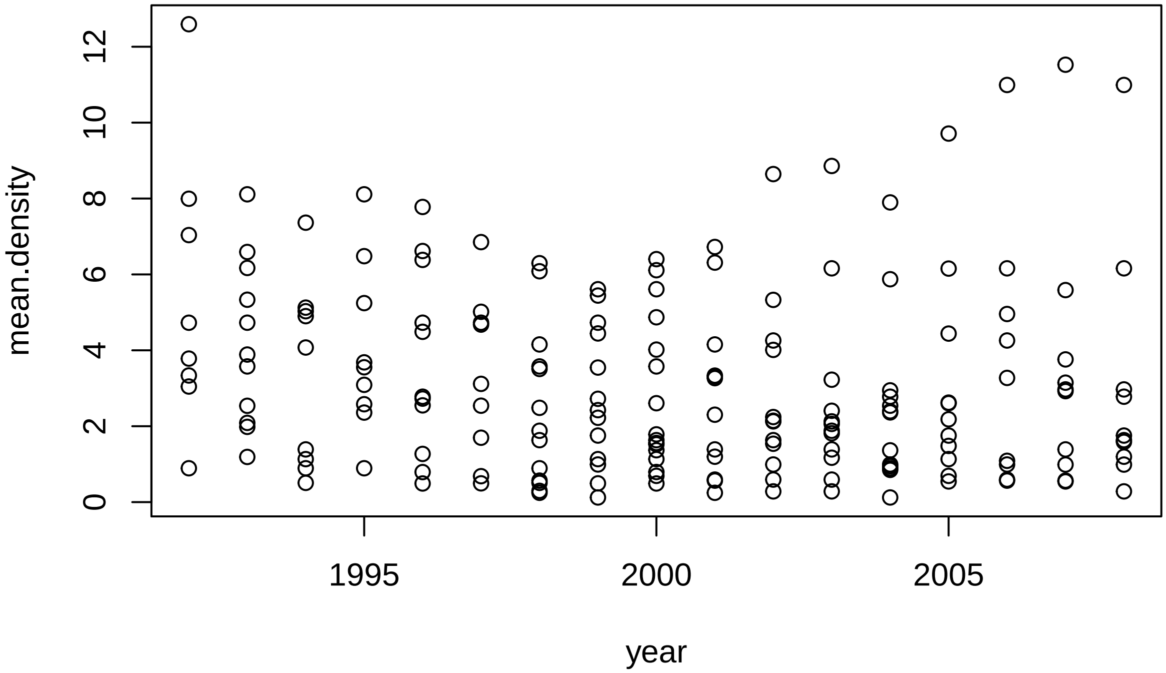
```
hares.grass <- subset(hares, landuse == "grass")

dens6 <- lm(mean.density ~ year, data = hares.grass)
summary(dens6)
```

```
##
## Call:
## lm(formula = mean.density ~ year, data = hares.grass)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.1798 -1.8019 -0.6822  1.2456  8.8889
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 164.76923   78.51067   2.099   0.0372 *
## year         -0.08078    0.03925  -2.058   0.0410 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.489 on 186 degrees of freedom
##   (84 observations deleted due to missingness)
## Multiple R-squared:  0.02227,    Adjusted R-squared:  0.01701
## F-statistic: 4.236 on 1 and 186 DF,  p-value: 0.04098
```

```
Anova(dens6)
```

```
## Anova Table (Type II tests)
##
## Response: mean.density
##            Sum Sq  Df F value  Pr(>F)
## year        26.24   1  4.2357 0.04098 *
## Residuals 1152.36 186
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
with(subset(hares, landuse == "grass"), plot(year, mean.density))
```



Mean density of brown hares does not seem to change linearly with year. Instead, there seems to be a drop in mean density around the year 2000.

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.

```r
dens7 <- lm(mean.density ~ year + I(year*year), data = hares.grass)
summary(dens7)
```

```
##
## Call:
## lm(formula = mean.density ~ year + I(year * year), data = hares.grass)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.9509 -1.8790 -0.5422  1.4058  8.2739
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)     9.176e+04  3.422e+04   2.682  0.00799 **
## year           -9.168e+01  3.422e+01  -2.679  0.00805 **
## I(year * year)  2.290e-02  8.555e-03   2.677  0.00810 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.449 on 185 degrees of freedom
##   (84 observations deleted due to missingness)
## Multiple R-squared:  0.05872,    Adjusted R-squared:  0.04854
## F-statistic:  5.77 on 2 and 185 DF,  p-value: 0.003707
```

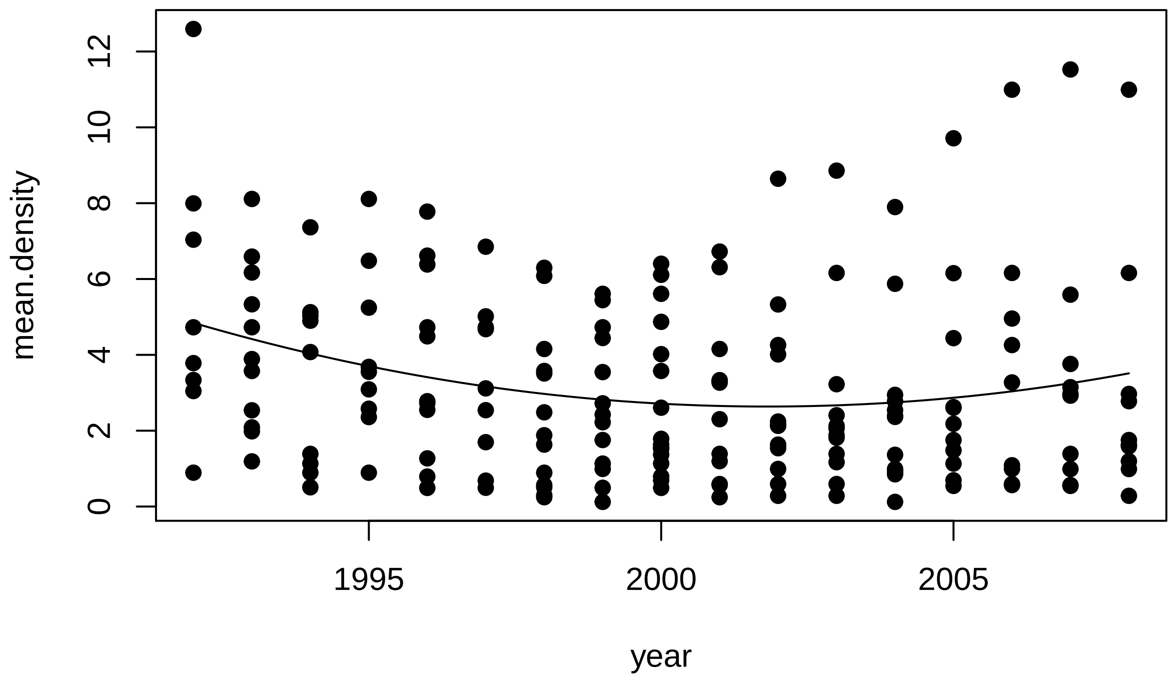
```r
Anova(dens7)
```

```
## Anova Table (Type II tests)
##
## Response: mean.density
##                 Sum Sq  Df F value   Pr(>F)
## year             43.04   1  7.1774 0.008048 **
## I(year * year)   42.97   1  7.1647 0.008103 **
## Residuals      1109.40 185
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

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).

We can predict values across the observed elevations, then plot our model predicted values.

```r
year.vals <- seq(min(hares.grass$year, na.rm = T), max(hares.grass$year, na.rm = T), len
pred.vals <- coef(dens7)[1] + coef(dens7)[2]*year.vals + coef(dens7)[3]*year.vals*year.v

with(hares.grass, plot(year, mean.density, pch = 19))
points(year.vals, pred.vals, type = "l")
```



Practice example 2: Using the built-in "ToothGrowth" R dataset, run linear regression to determine whether tooth growth changes linearly with vitamin C dosage.

```r
tg1 <- lm(len ~ dose, data = ToothGrowth)
```

ANCOVA

Here, we are running an analysis of covariance (ANCOVA) to evaluate the interactive effects of landuse and year on mean density of brown hares. Run two models with a linear term for elevation and one with a quadratic term.

```r
dens8 <- lm(mean.density ~ year*landuse, data = hares)
summary(dens8)
```

```
##
## Call:
## lm(formula = mean.density ~ year * landuse, data = hares)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -5.5404 -2.5285 -0.8794  1.5791 16.8569
##
## Coefficients:
##                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)       -150.47614   70.35486  -2.139   0.0328 *
## year                 0.07790    0.03517   2.215   0.0271 *
## landusegrass       315.24536  135.82951   2.321   0.0206 *
## year:landusegrass   -0.15868    0.06791  -2.337   0.0197 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.684 on 673 degrees of freedom
##   (275 observations deleted due to missingness)
## Multiple R-squared:  0.07275,    Adjusted R-squared:  0.06862
## F-statistic:  17.6 on 3 and 673 DF,  p-value: 5.205e-11
```

```r
Anova(dens8)
```

```
## Anova Table (Type II tests)
##
## Response: mean.density
##              Sum Sq  Df F value    Pr(>F)
## year            18.7   1  1.3785   0.24078
## landuse        622.7   1 45.8879 2.734e-11 ***
## year:landuse    74.1   1  5.4600   0.01975 *
## Residuals     9131.9 673
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
dens9 <- lm(mean.density ~ year*landuse + I(year*year)*landuse, data = hares)
summary(dens9)
```

```
##
## Call:
## lm(formula = mean.density ~ year * landuse + I(year * year) *
##     landuse, data = hares)
##
```

```
## Residuals:
##     Min      1Q  Median      3Q     Max 
## -5.9994 -2.4712 -0.8414  1.6214 17.0969 
## 
## Coefficients:
##                               Estimate Std. Error t value Pr(>|t|)
## (Intercept)                  4.477e+04  3.238e+04   1.383    0.167
## year                        -4.484e+01  3.238e+01  -1.385    0.167
## landusegrass                 4.699e+04  6.071e+04   0.774    0.439
## I(year * year)               1.123e-02  8.095e-03   1.387    0.166
## year:landusegrass           -4.684e+01  6.071e+01  -0.771    0.441
## landusegrass:I(year * year)  1.167e-02  1.518e-02   0.769    0.442
## 
## Residual standard error: 3.675 on 671 degrees of freedom
##   (275 observations deleted due to missingness)
## Multiple R-squared:  0.07976,    Adjusted R-squared:  0.0729 
## F-statistic: 11.63 on 5 and 671 DF,  p-value: 8.542e-11
```

```
Anova(dens9)
```

```
## Anova Table (Type II tests)
## 
## Response: mean.density
##                        Sum Sq  Df F value    Pr(>F)    
## year                     60.9   1  4.5091   0.03408 *  
## landuse                 687.8   2 25.4615 2.196e-11 ***
## I(year * year)           61.0   1  4.5142   0.03398 *  
## year:landuse              8.0   1  0.5951   0.44071    
## landuse:I(year * year)    8.0   1  0.5911   0.44226    
## Residuals              9063.0 671                      
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Practice example 3: From the infected individuals only, run an ANCOVA evaluating the interactive effects of sex and kidney fat index (KFI) on the estimated number of *Elaphostrongylus cervi* parasites.

This example uses epidemiological data from Vicente et al. (2006) that took observations on wild boar and red deer reared on a number of estates in Spain. Here, the dataset contains information on the red deer only. 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), and the presence (0, 1) of Tuberculosis were taken. These data are used in examples from the Zuur book.

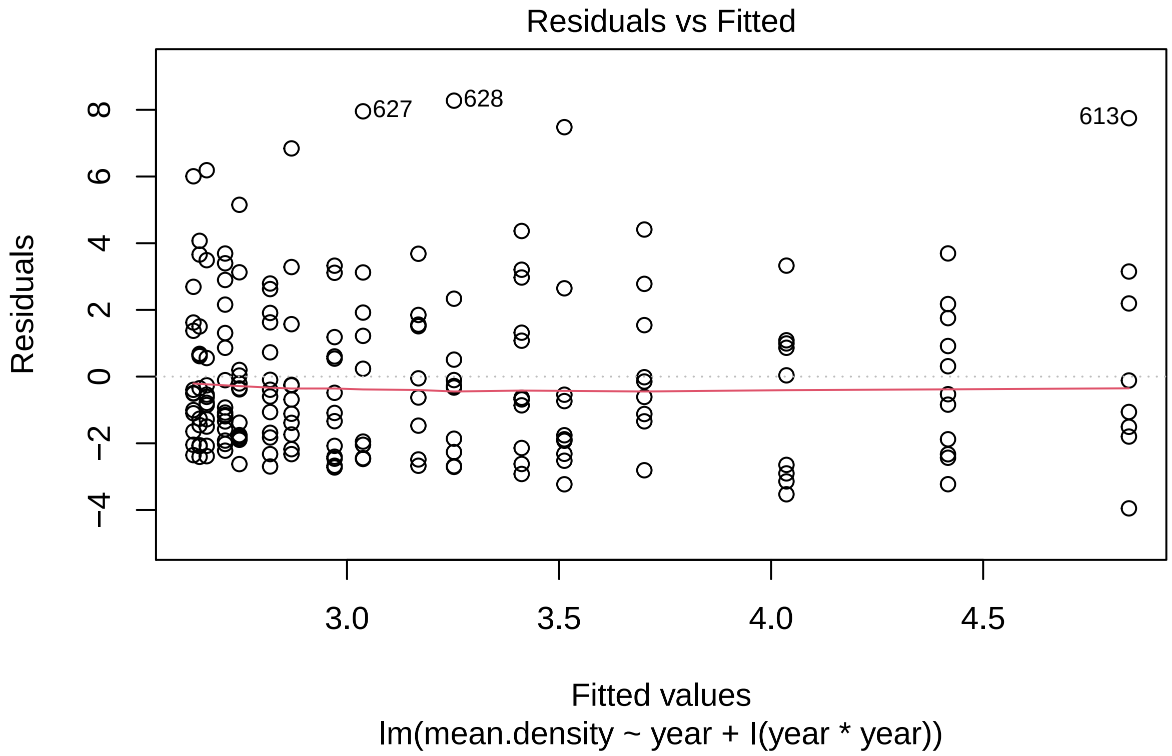
```r
deer <- read.table(here::here("Session 2", "Data", "Deer.txt"), header = T)

ec1 <- lm(Ecervi ~ KFI*Sex, data = subset(deer, Ecervi > 0))
```

Assumptions of normality

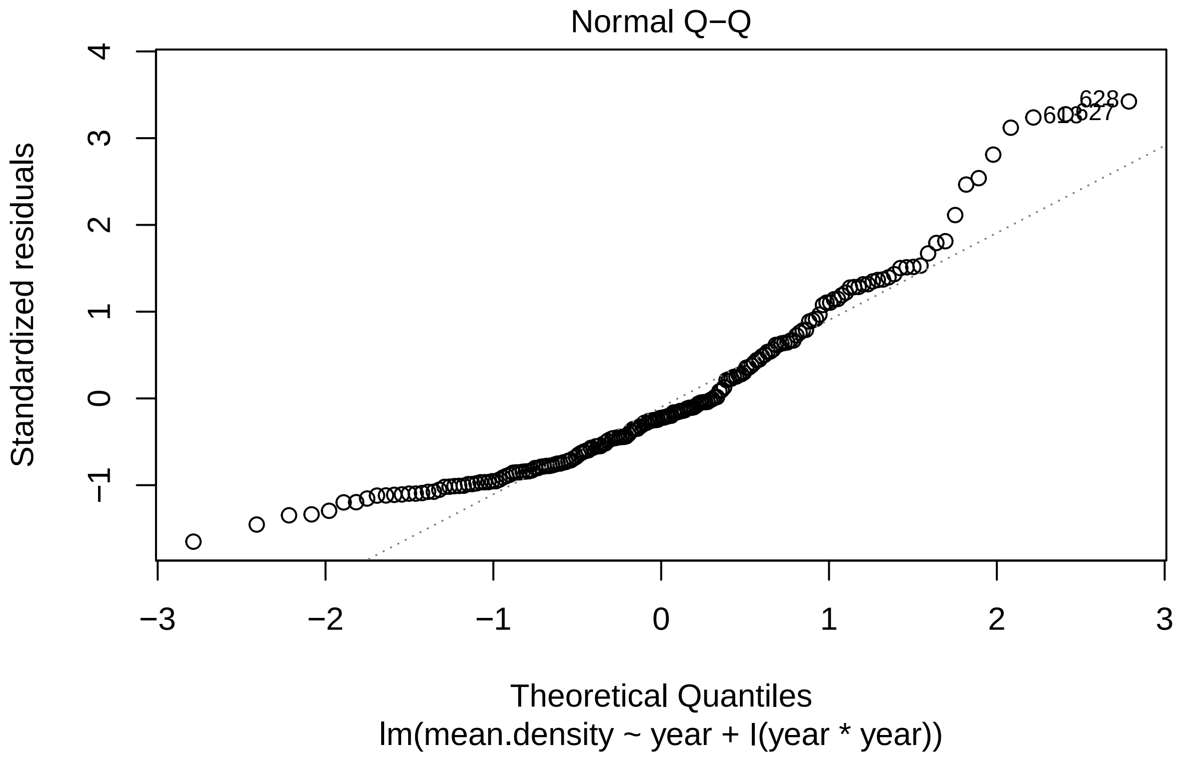
Before continuing with this statistical distribution, we always want to test for the assumptions of normality on our response variable. A linear regression has four assumptions: 1. *Linearity of the data:* linear relationship between x and y. (residuals vs fitted values) 2. *Normality of residuals:* residual errors are normally-distributed. (QQ plot) 3. *Homogeneity of residuals variance (i.e., homoscedasicity):* 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:

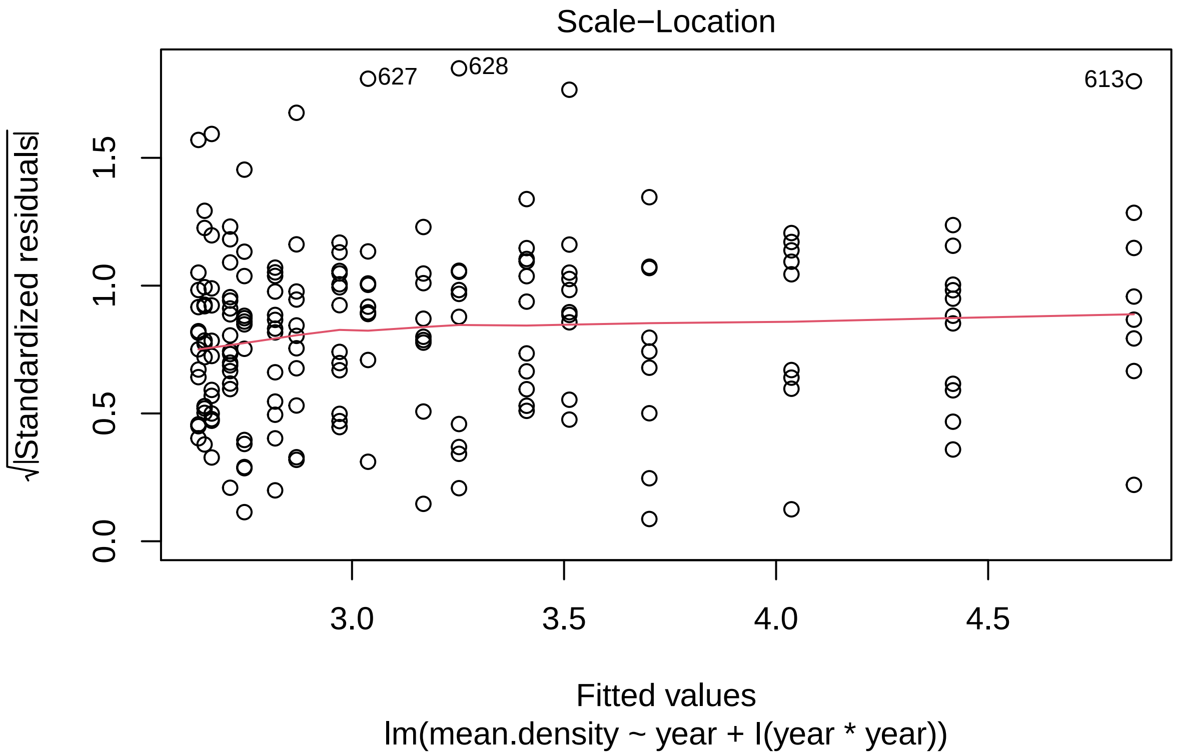
```r
dens7 <- lm(mean.density ~ year + I(year*year), data = hares.grass)

plot(dens7, 1) # Residuals vs fitted - red line is flat meaning linear relationsip
```



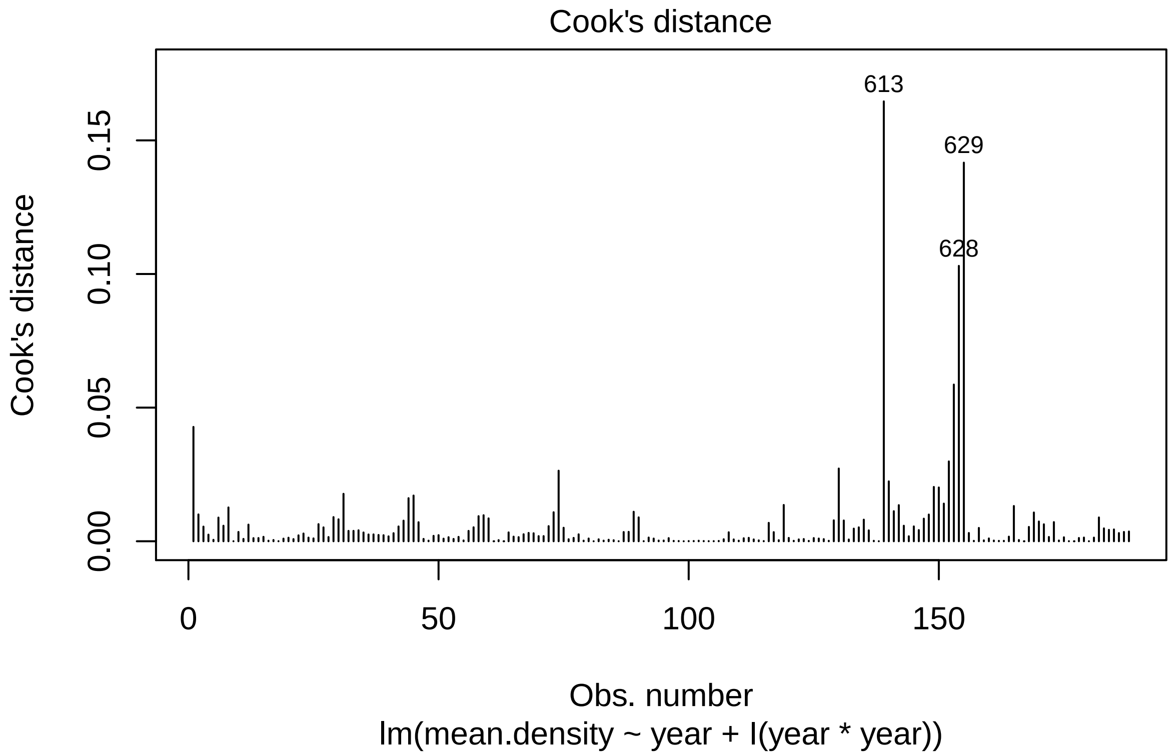
```r
plot(dens7, 2) # QQ plot - examine whether residuals are normally-distributed
```



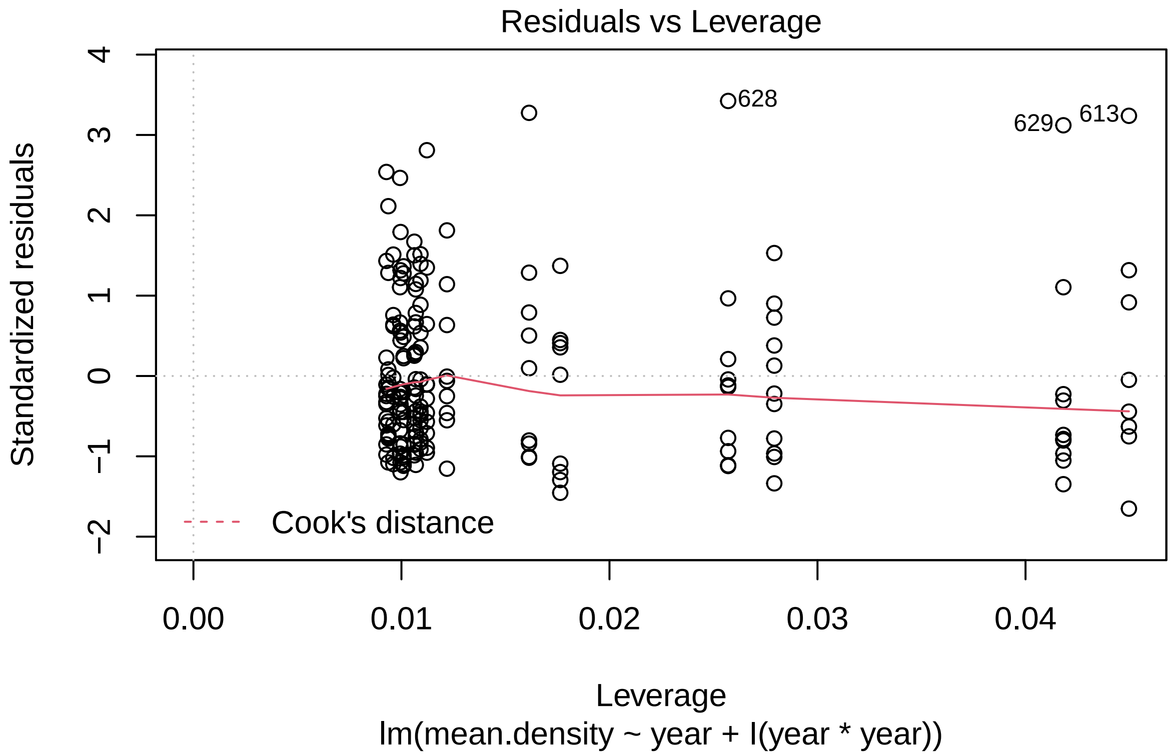
```r
plot(dens7, 3) # Scale-location - homogeneity of variance of residuals (homoscedasicit
```



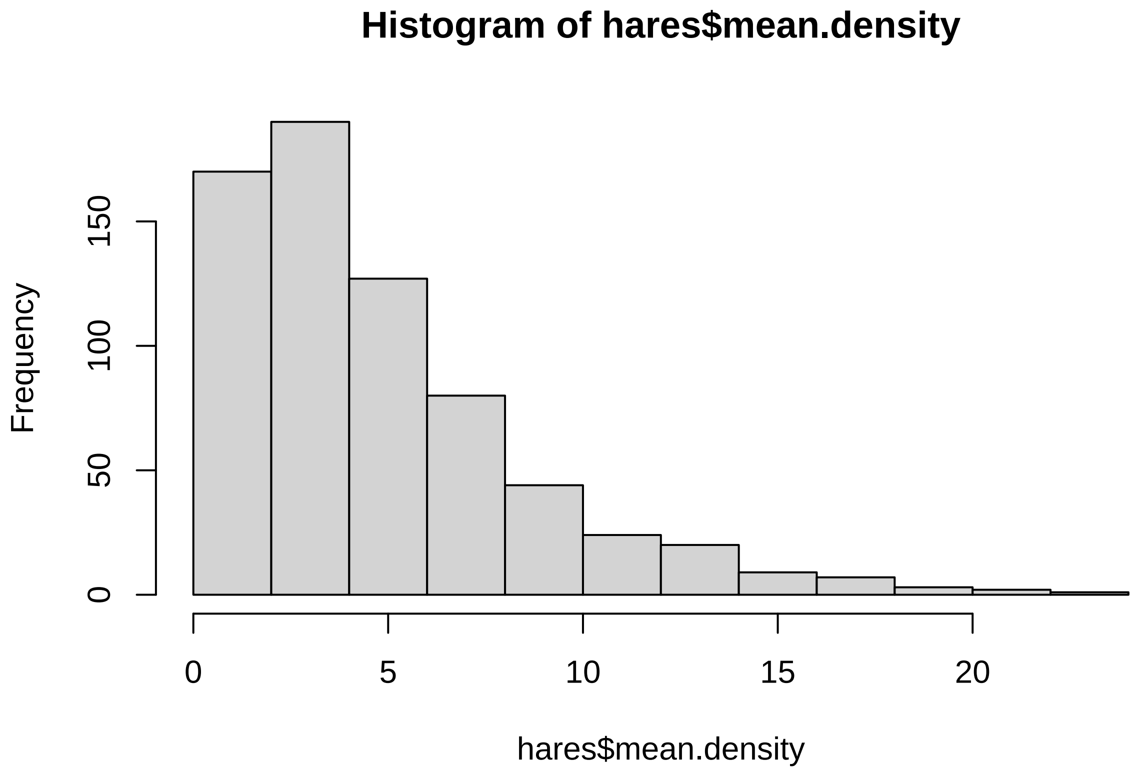
```r
plot(dens7, 4) # Cook's distance - identify extreme values and their obs. number
```



```
plot(dens7, 5) # Residuals vs leverage - for identifying influential cases or extreme
```



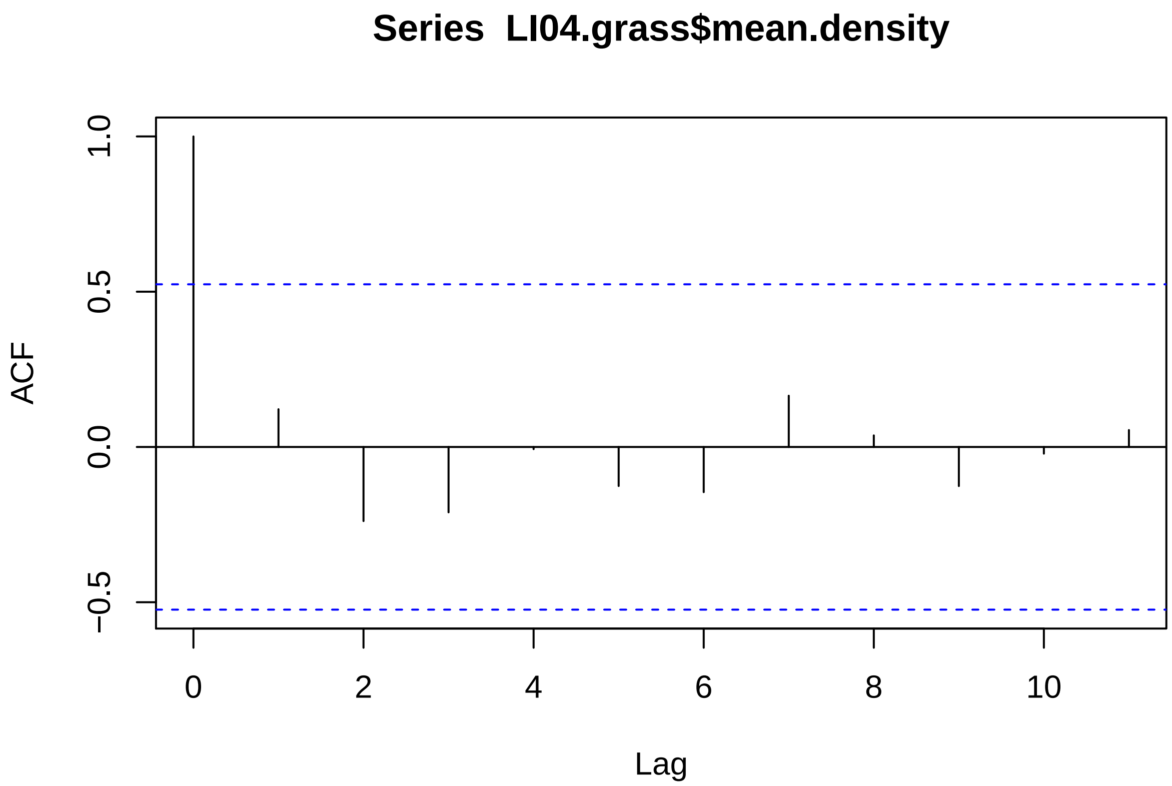
```r
hist(hares$mean.density)
```



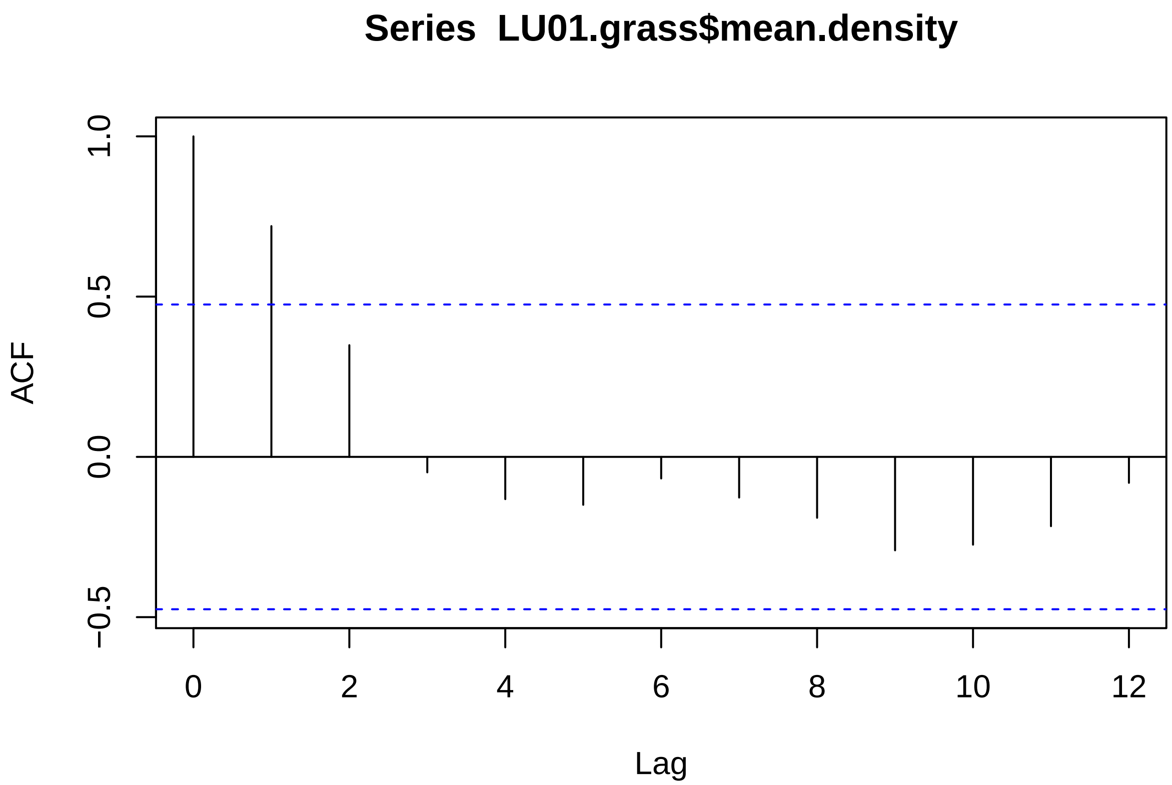
It is difficult to do a general test for dependence of the residual error term. You will need to know why your residual error terms may be dependent: either residuals can correlate with another variable (e.g., check residuals-fitted plot) or residuals can correlate with a nearby residual (e.g., autocorrelation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

View(hares.grass)

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



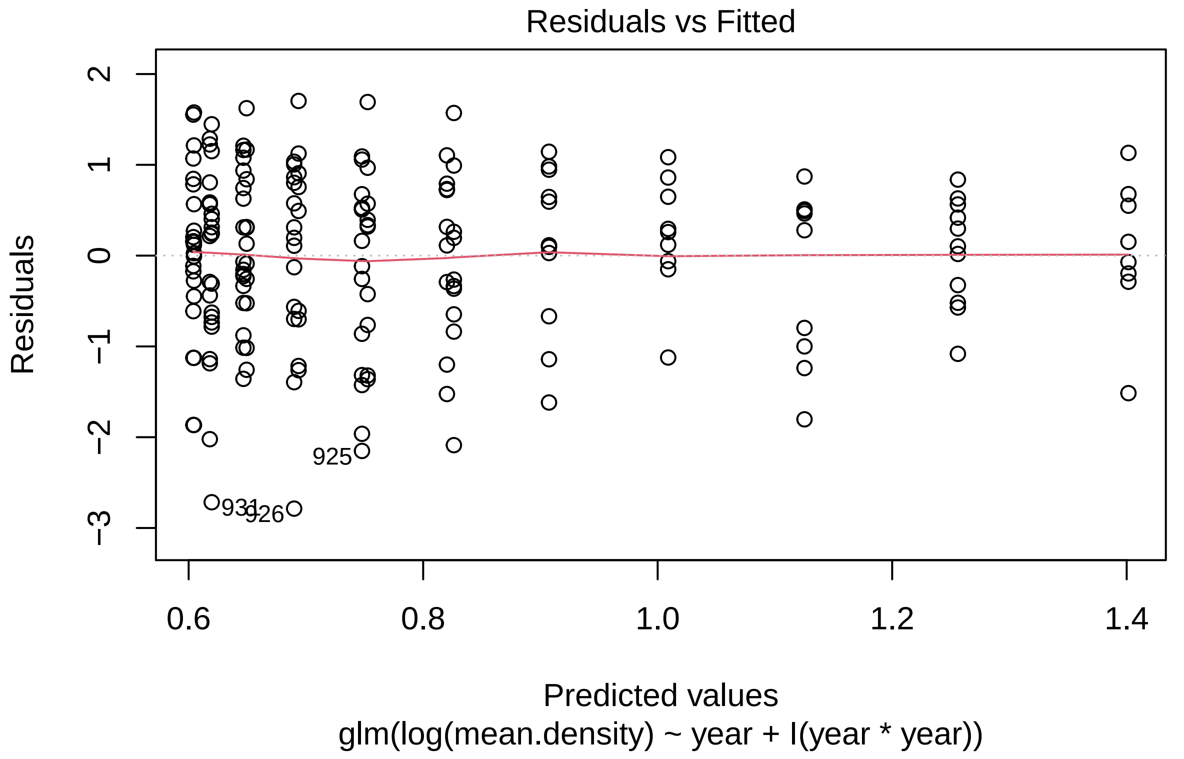
```
LU01.grass <- subset(hares.grass, site == "LU01" & !is.na(mean.density))
acf(LU01.grass$mean.density)
```



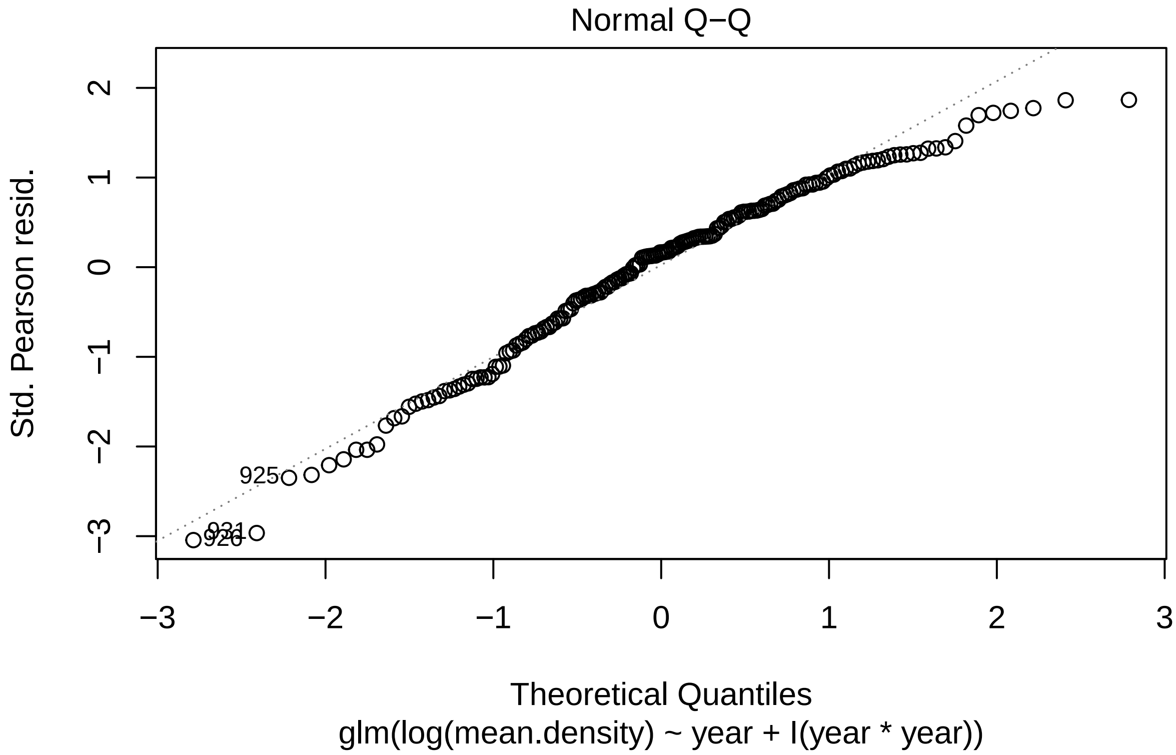
In any case, we can see that these data are negatively (left) skewed and bound by 0 to ∞. First, we can use the most common approach when our data do not met the assumptions of normality, which is to transform our response variable, y. There are many ways to transform your data (e.g., log, square-root, arcsine). However, we will only cover the log transformation of our response variable, y (i.e., $\log(y)$ is normal given x):

$$log(y_i) = \beta_0 + \beta_1 x_i$$

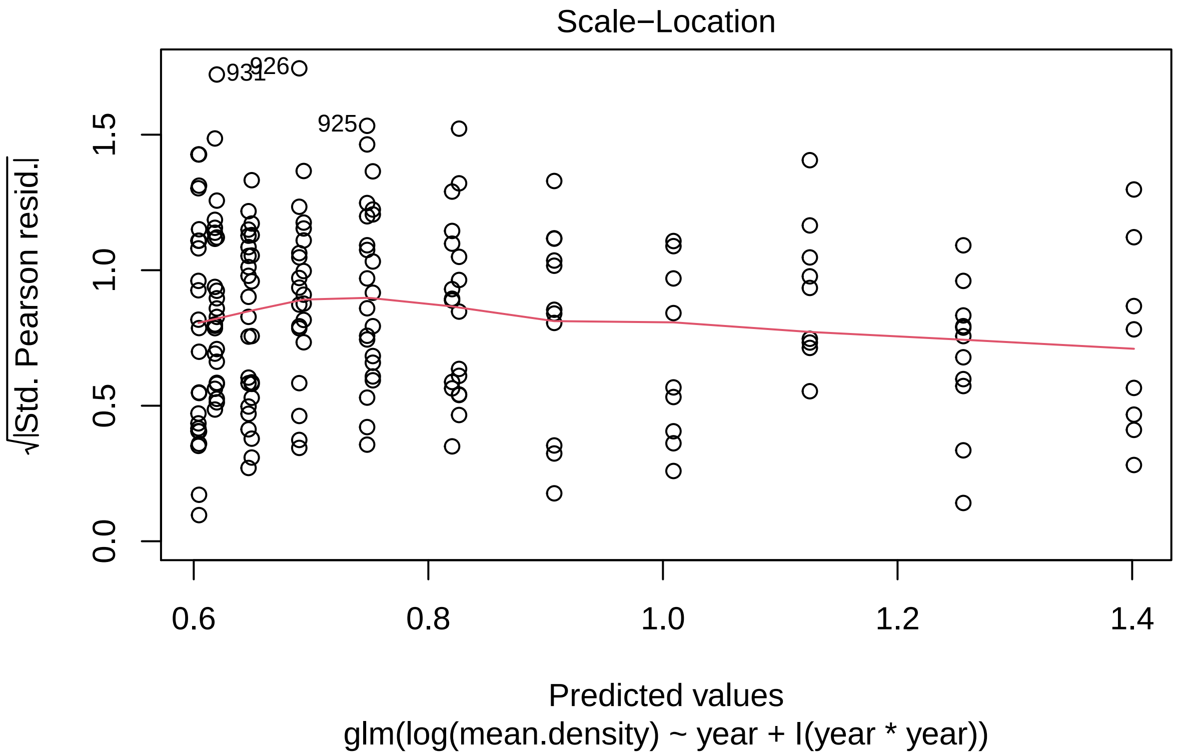
```
dens10 <- glm(log(mean.density) ~ year + I(year*year), family = gaussian, data = hares.g

plot(dens10, 1) # linear relationship between x and y (good!)
```



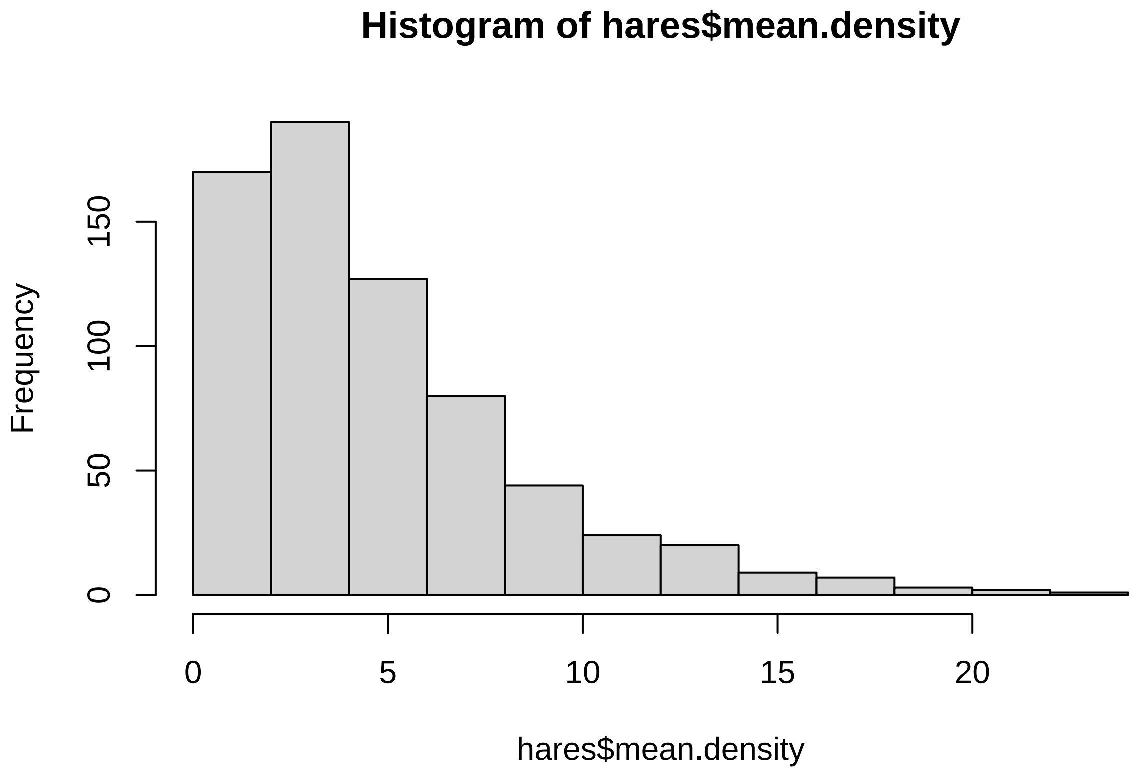
```
plot(dens10, 2) # residuals are not normality-distributed, mild right-skew (ok)
```



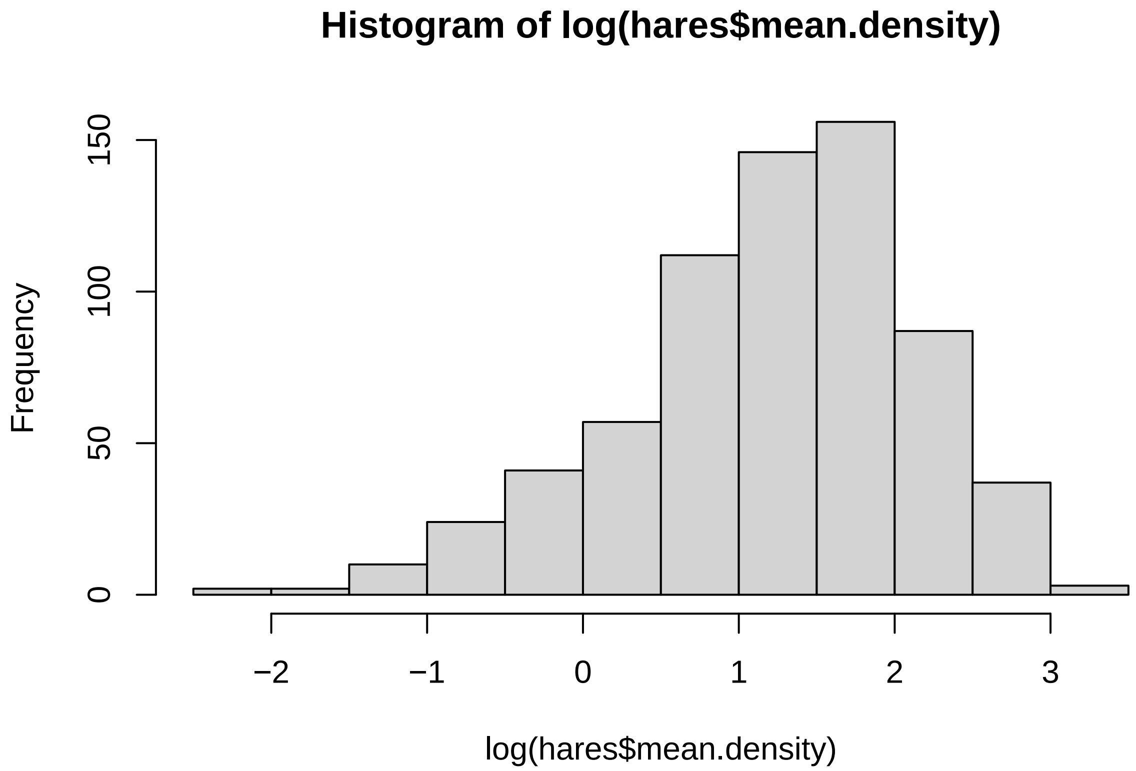
```r
plot(dens10, 3) # mild constant variance of the residuals (ok)
```



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



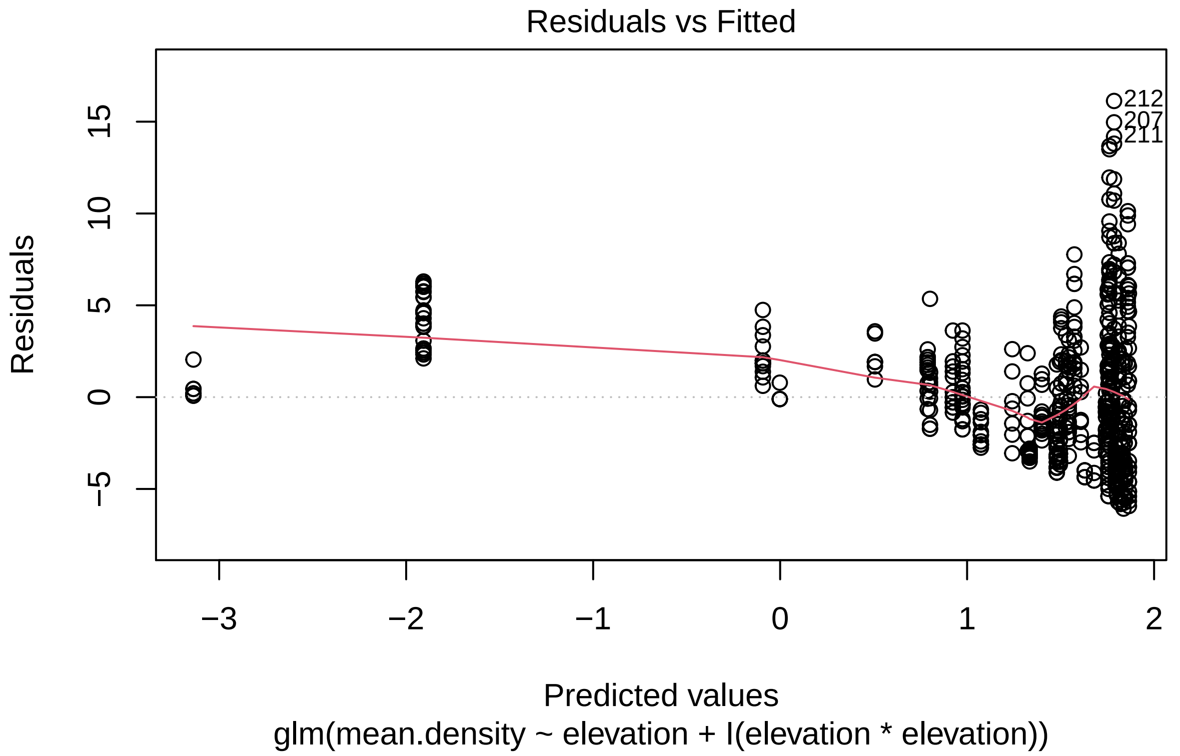
```
hist(log(hares$mean.density)) # positively skewed/right skewed
```



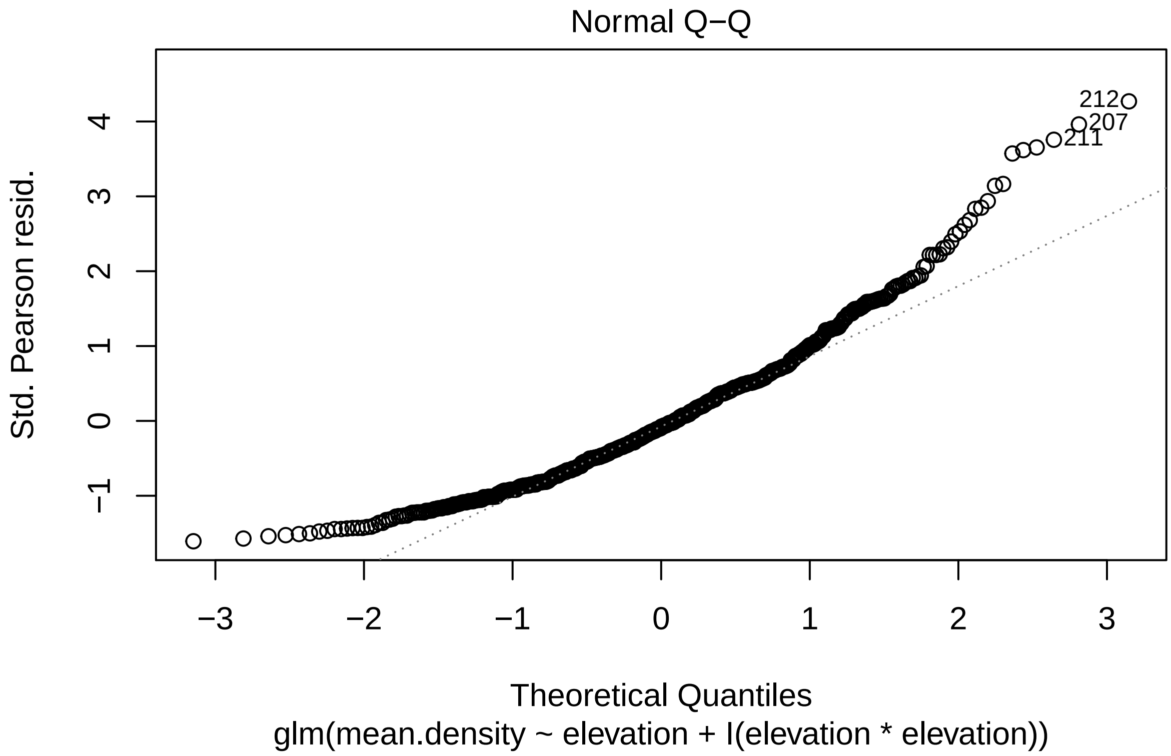
Instead of transforming our response to fit a statistical distribution (i.e., $\log(y)$ is normal given x), we can choose a statistical distribution that fits our data. First, we might want to try the log-link Gaussian distribution (i.e., mean of $\log(y)$ responses linearly to x), such that:

$$ln(\mu) = \beta_0 + \beta_1 x$$

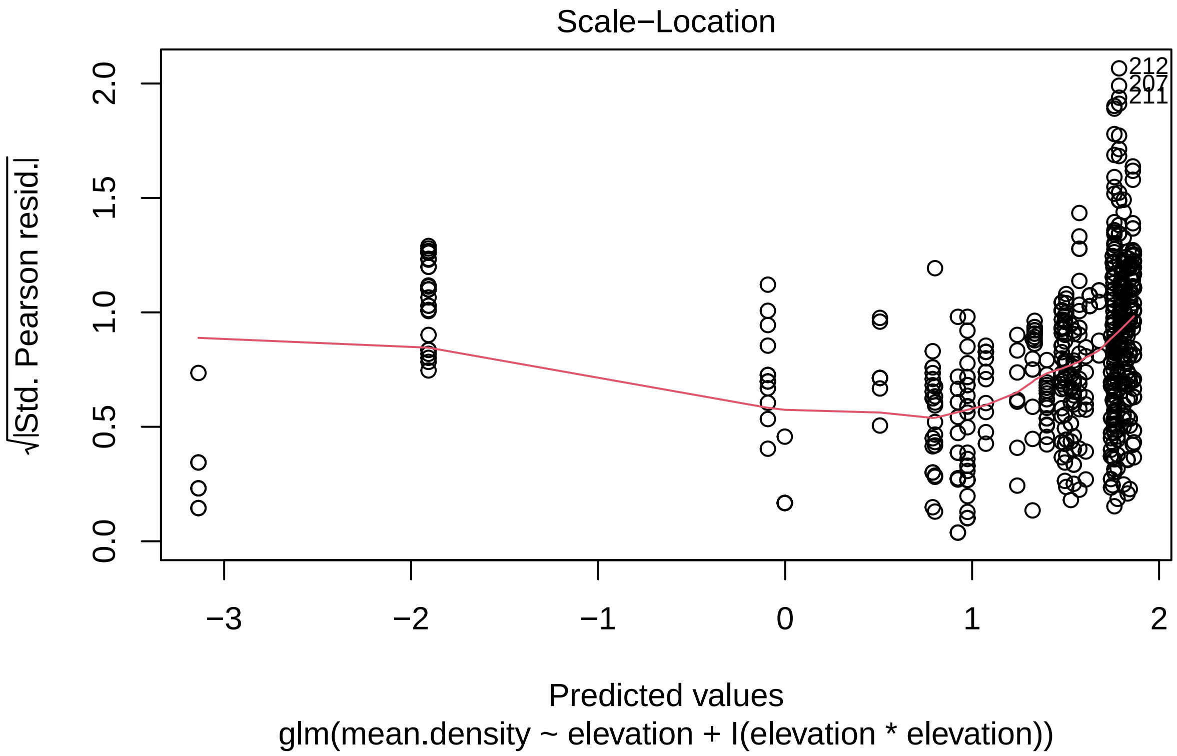
```
dens11 <- glm(mean.density ~ elevation  + I(elevation*elevation), family = gaussian(link

plot(dens11, 1) # relationship between x and y is not linearly (bad)
```



```r
plot(dens11, 2) # residuals are not normality-distributed, left-skewed (bad)
```



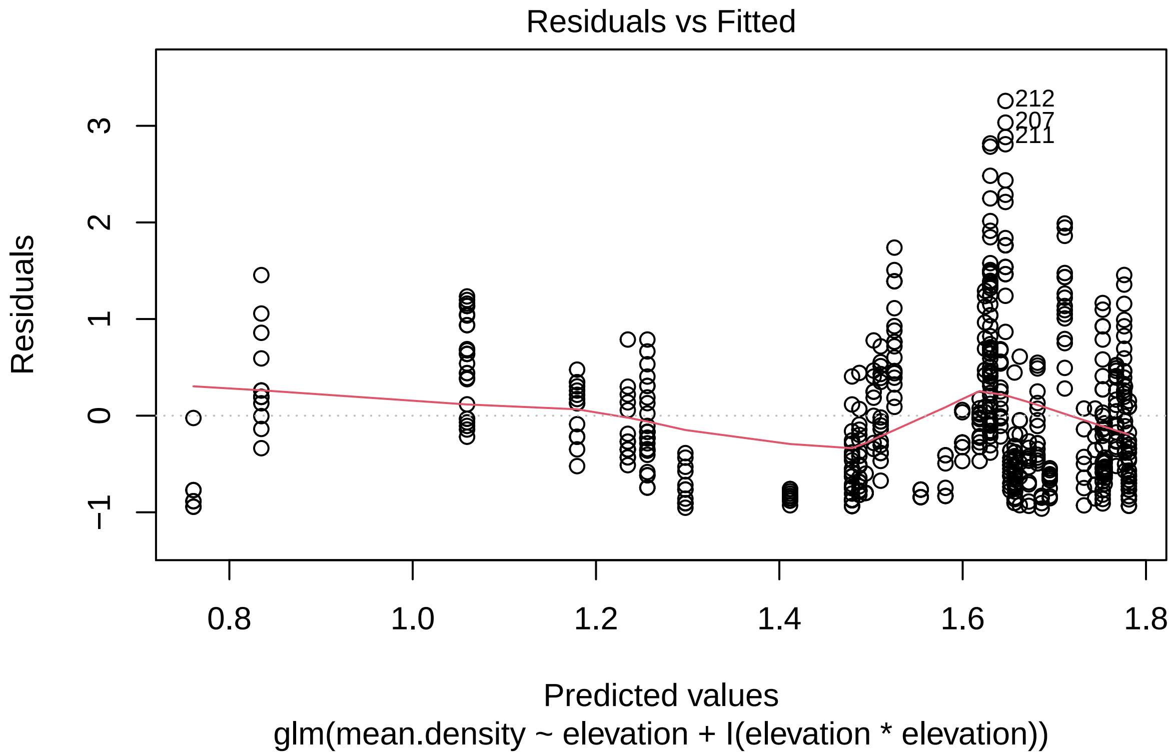
```
plot(dens11, 3) # hetereoscedasicity should exist, since the variance changes with the
```



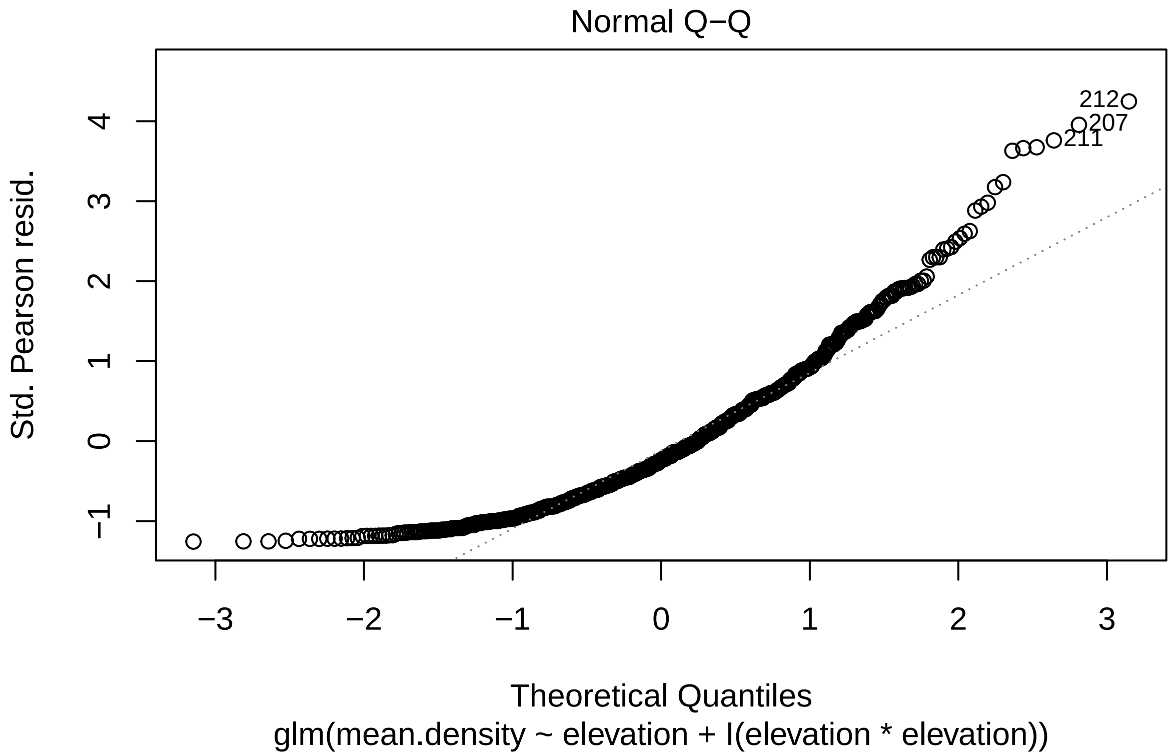
such as the Gamma distribution (i.e., y is normal given x on either the log- or inverse-link function scale) or the log-link Gaussian, whereby:

Since our data are bound at zero, we can use an alternative statistical distribution that allows for non-negative, skewed, continuous data that are bound by 0 to infinity. The Gamma distribution (either log Gamma or inverse Gamma) assumes heavier tailed/skewed distribution, particularly the inverse Gamma distribution.

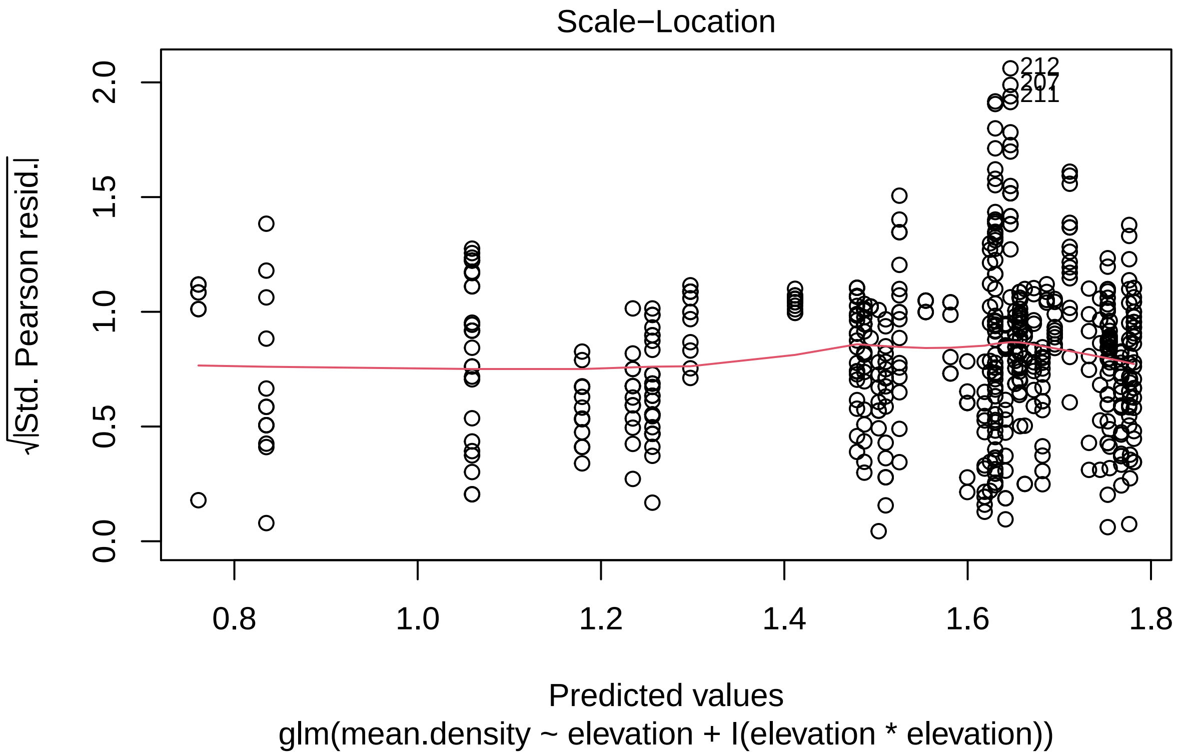
```
dens11 <- glm(mean.density ~ elevation  + I(elevation*elevation), family = Gamma(link =
plot(dens11, 1)
```



```
plot(dens11, 2) # address skew
```



```
plot(dens11, 3) # hetereoscedasicity should exist, since the variance changes with the
```

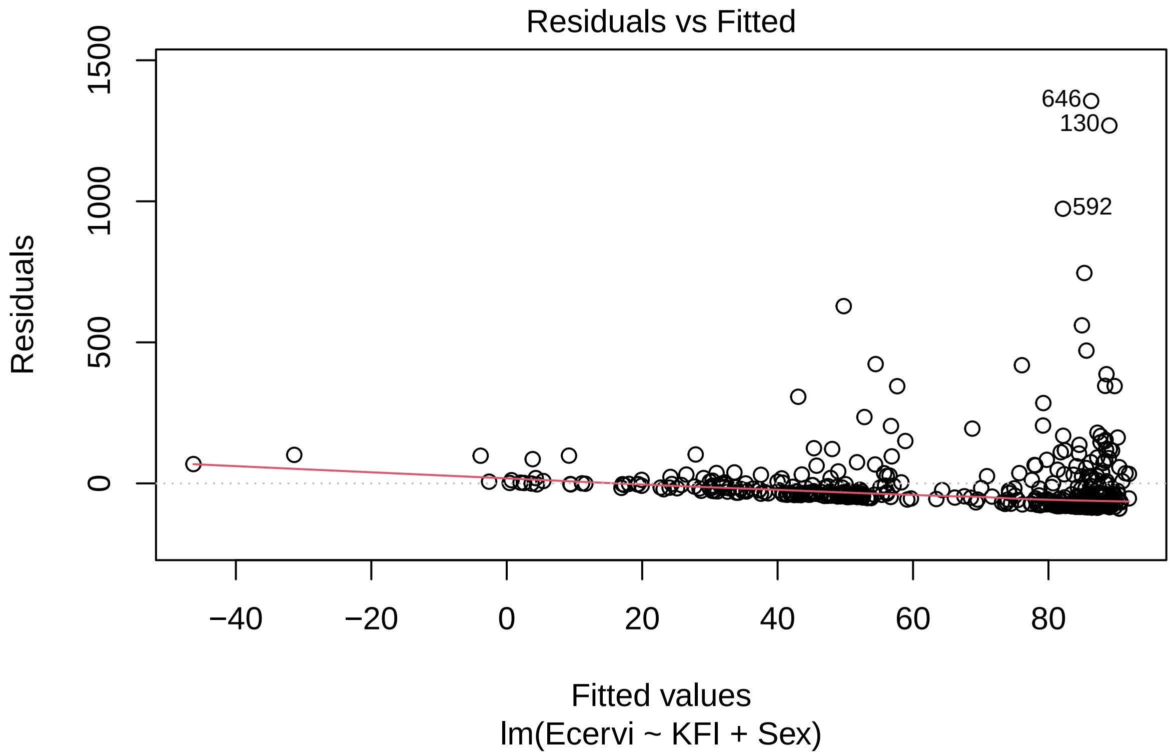


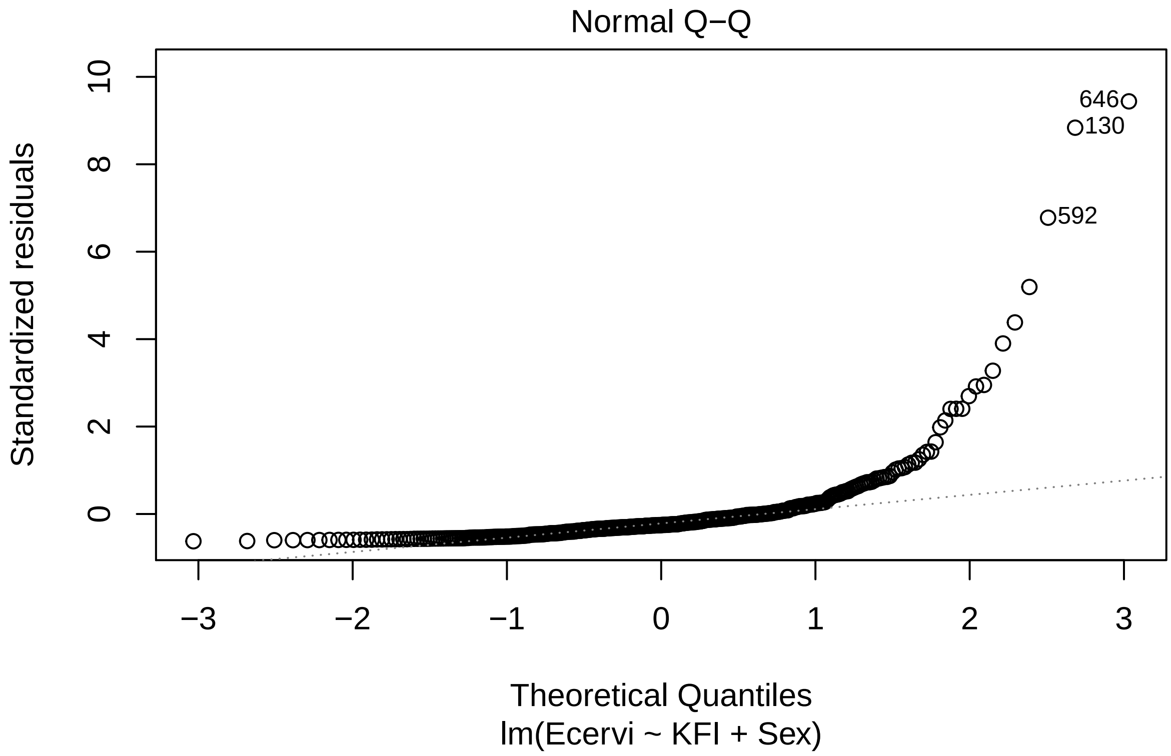
The Gamma distribution (or any distribution with a scale term) does not need to worry about the same assumptions of a linear regression, except the deviance residuals should be normal and hetereoscadicity should be observed.

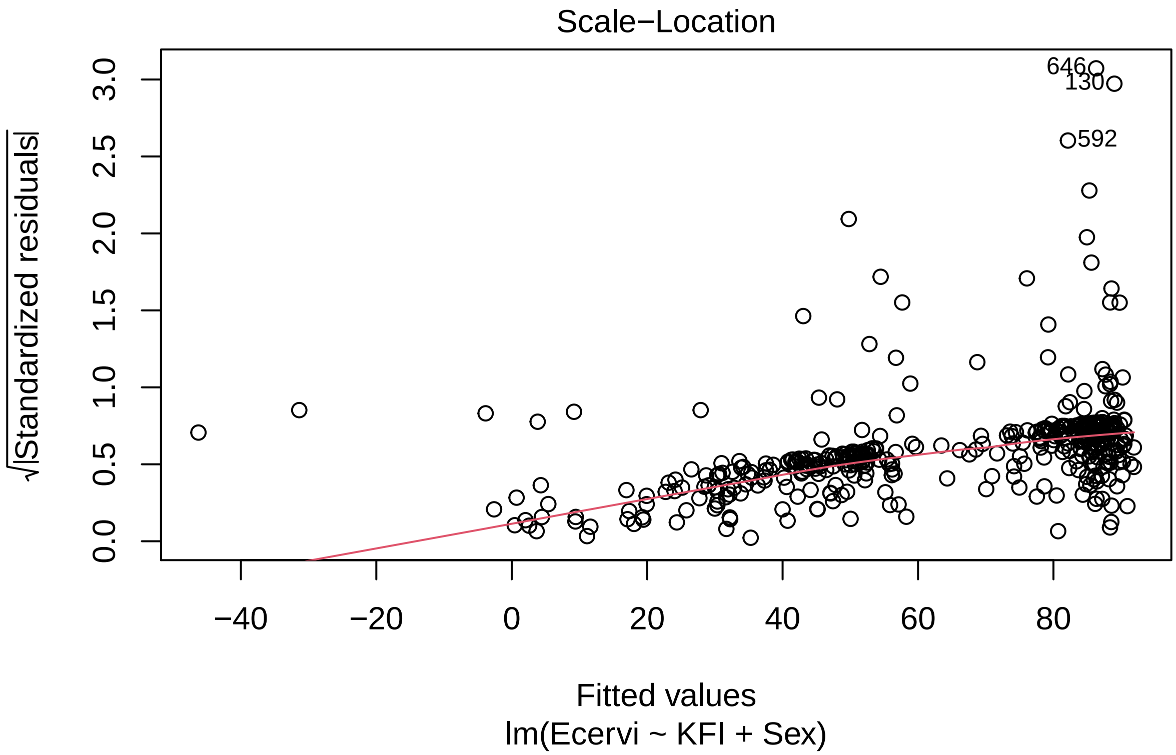
```
dens11 <- glm(mean.density ~ elevation  + I(elevation*elevation), family = gaussian(link
```

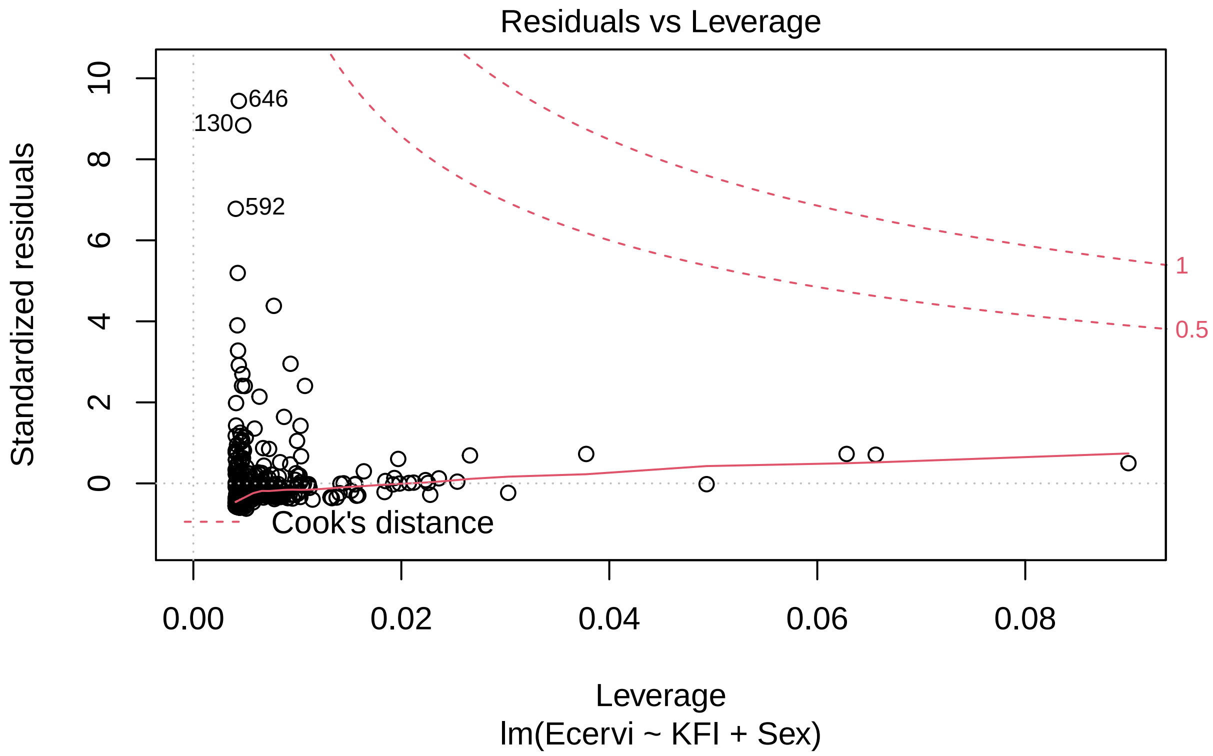
Practice example 4: Test the assumptions of normality for the model run for practice 3 (i.e., model without an interactive term). Then, find the appropriate approach to deal with any assumptions that are not met.

```
ec2 <- lm(Ecervi ~ KFI + Sex, data = subset(deer, Ecervi > 0))
plot(ec2)
```

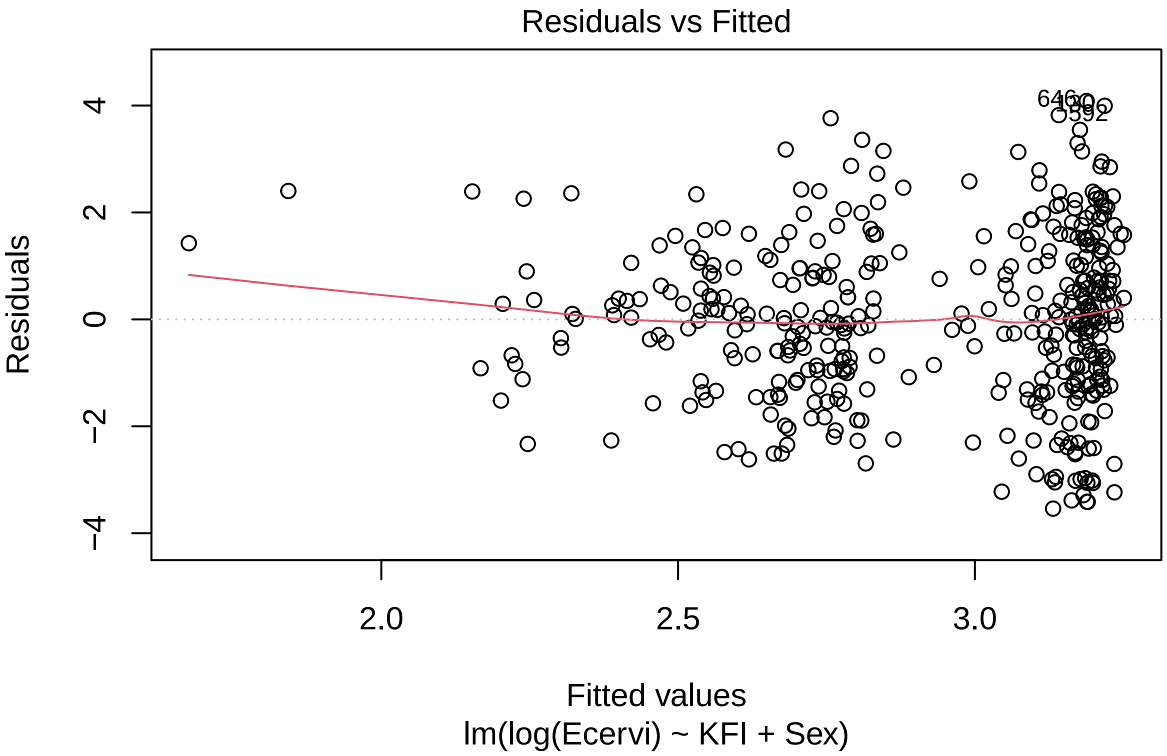

Residuals vs Fitted
Residuals
1500
1000
500
0
−40
−20
0
20
40
60
80
Fitted values
lm(Ecervi ~ KFI + Sex)
646
130
592

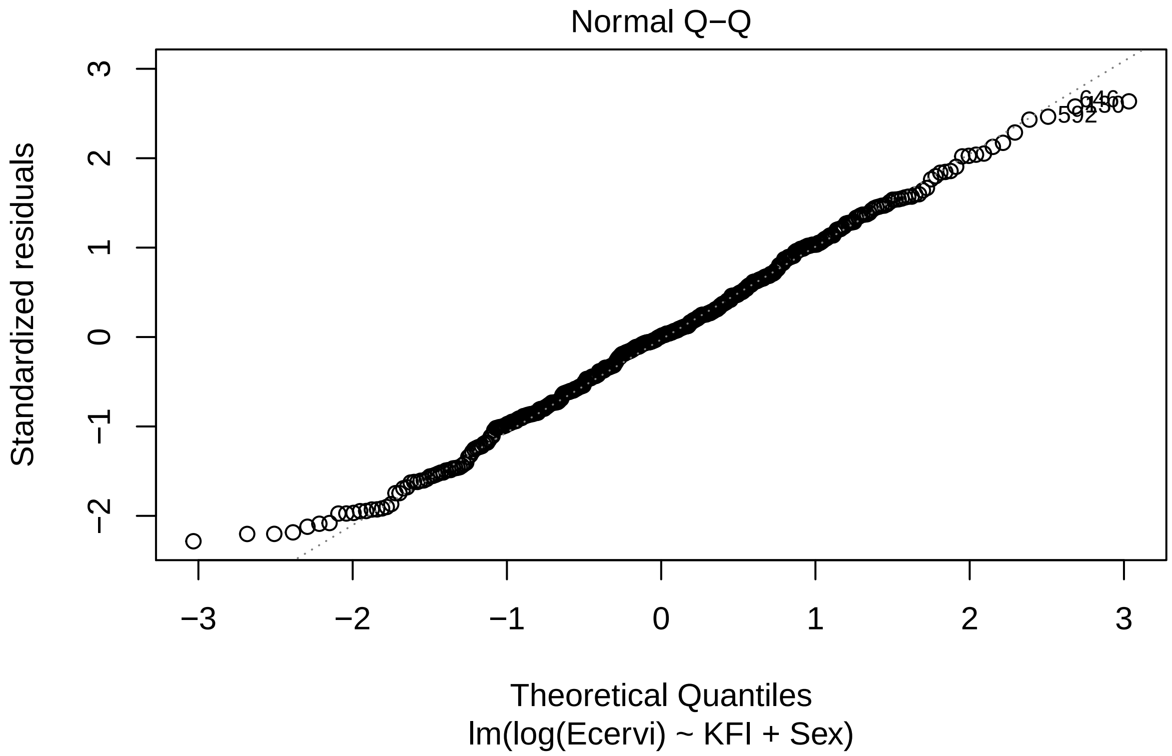

Normal Q−Q
Standardized residuals
0
2
4
6
8
10
646
130
592
−3
−2
−1
0
1
2
3
Theoretical Quantiles
lm(Ecervi ~ KFI + Sex)

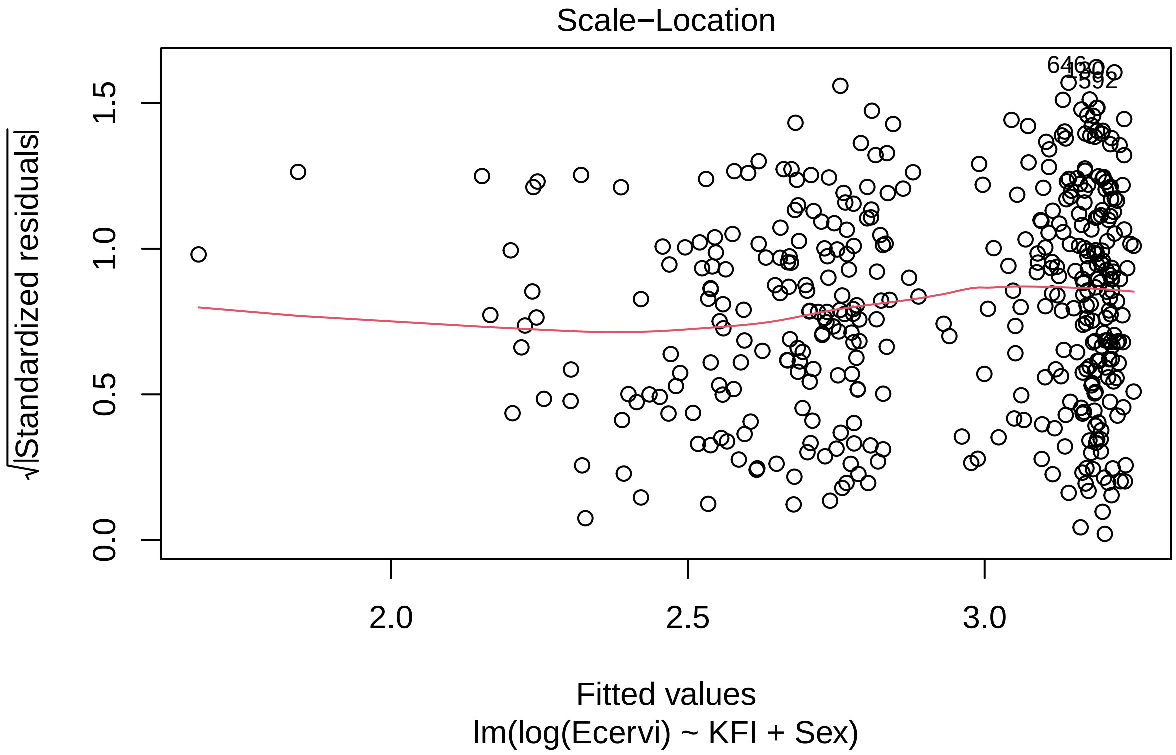

Scale−Location
√|Standardized residuals|
0.0
0.5
1.0
1.5
2.0
2.5
3.0
−40
−20
0
20
40
60
80
646
130
592
Fitted values
lm(Ecervi ~ KFI + Sex)

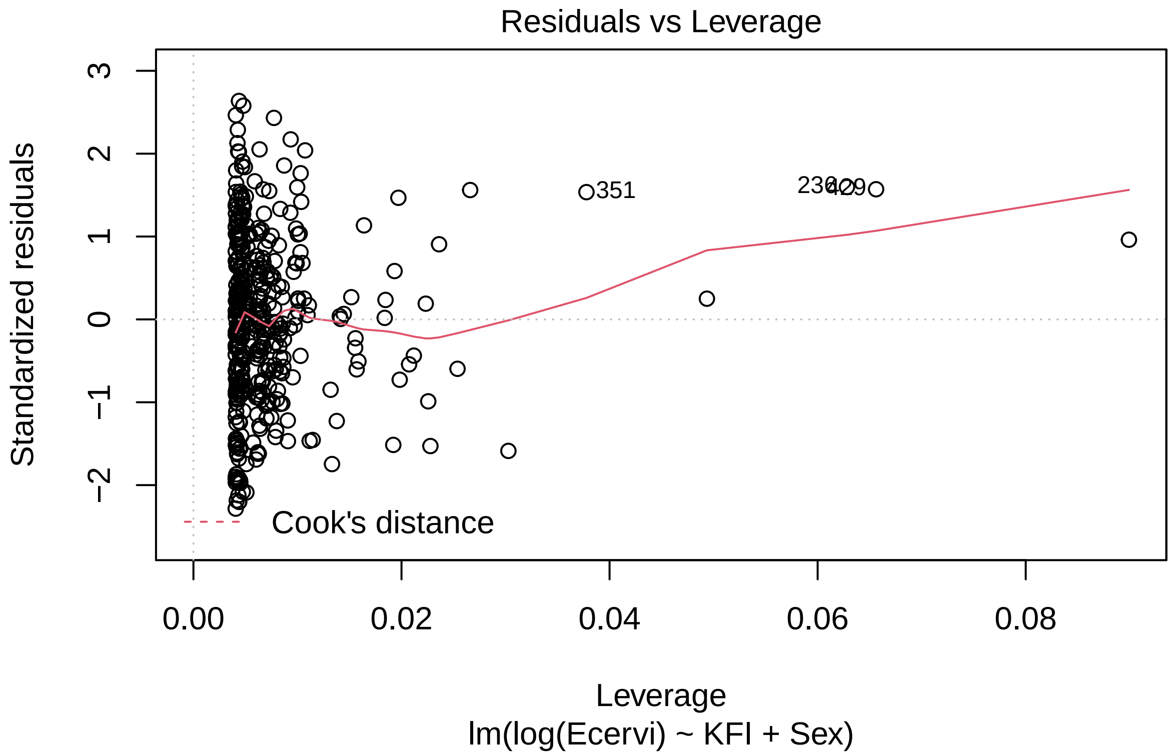


```r
ec3 <- lm(log(Ecervi) ~ KFI  + Sex, data = subset(deer, Ecervi > 0))
plot(ec3)
```

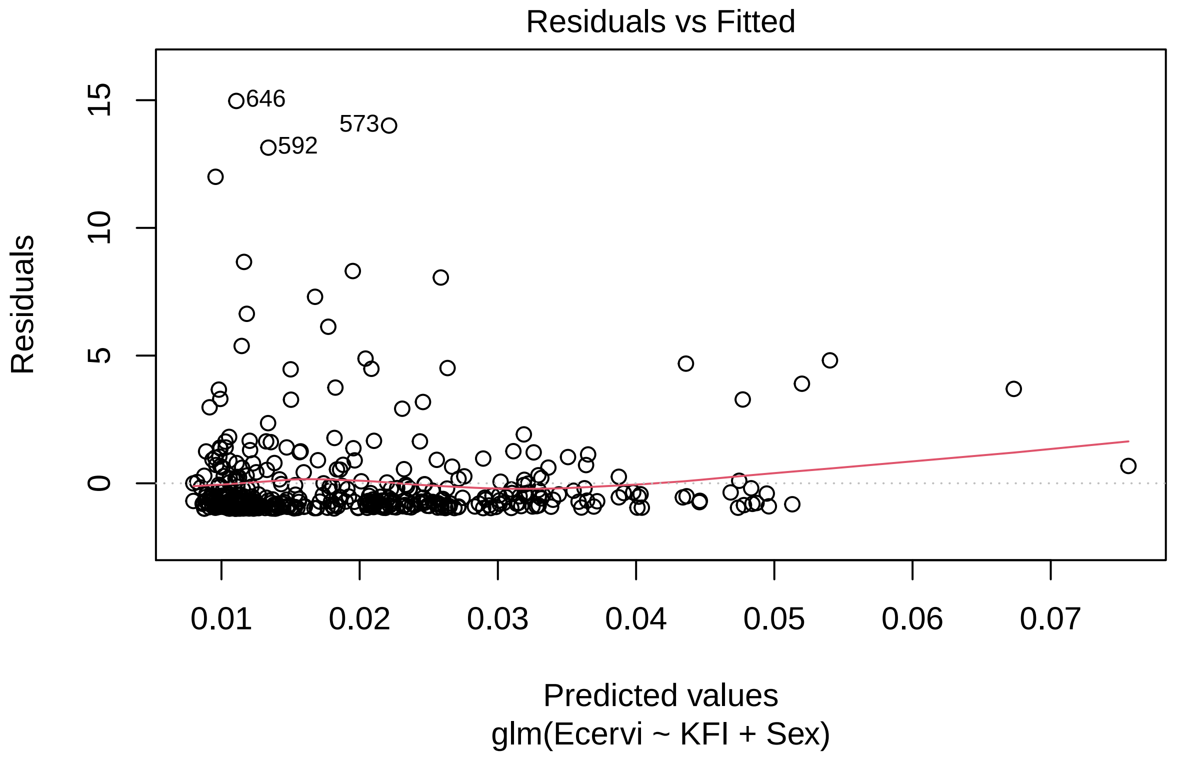

Residuals vs Fitted
Residuals
4
2
0
−2
−4
646
150
592
2.0
2.5
3.0
Fitted values
lm(log(Ecervi) ~ KFI + Sex)

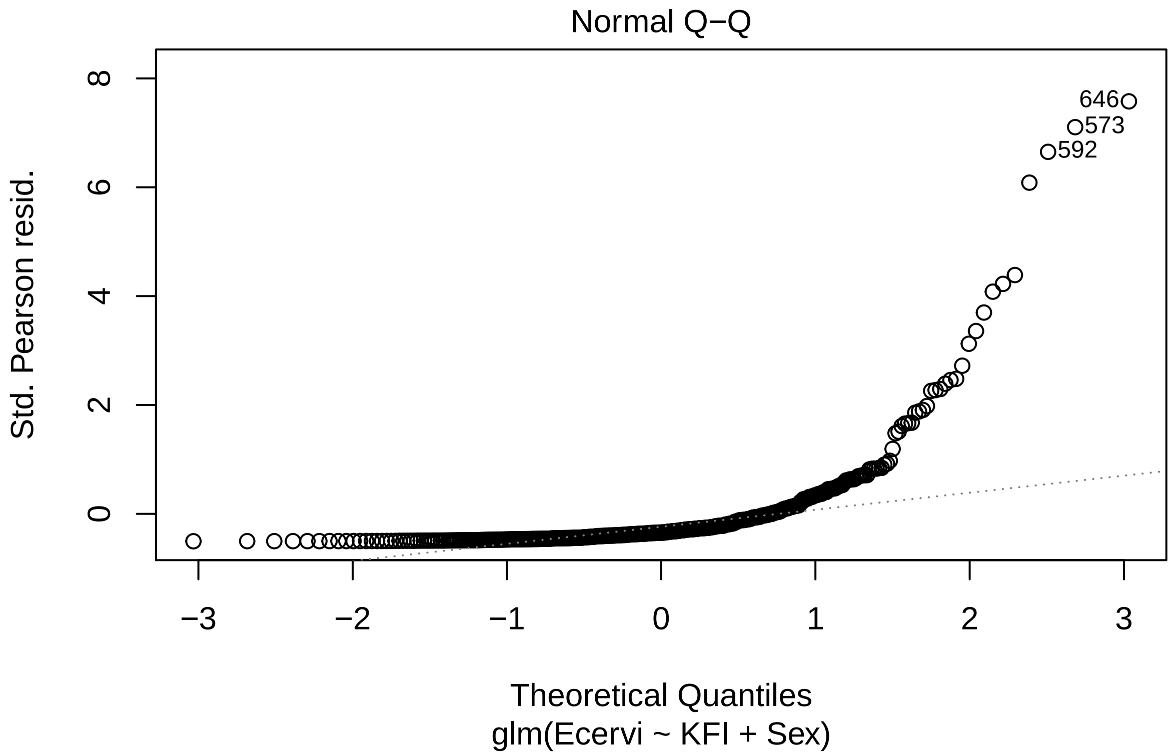

Normal Q−Q
Standardized residuals
3
2
1
0
−1
−2
646
450
592
−3
−2
−1
0
1
2
3
Theoretical Quantiles
lm(log(Ecervi) ~ KFI + Sex)

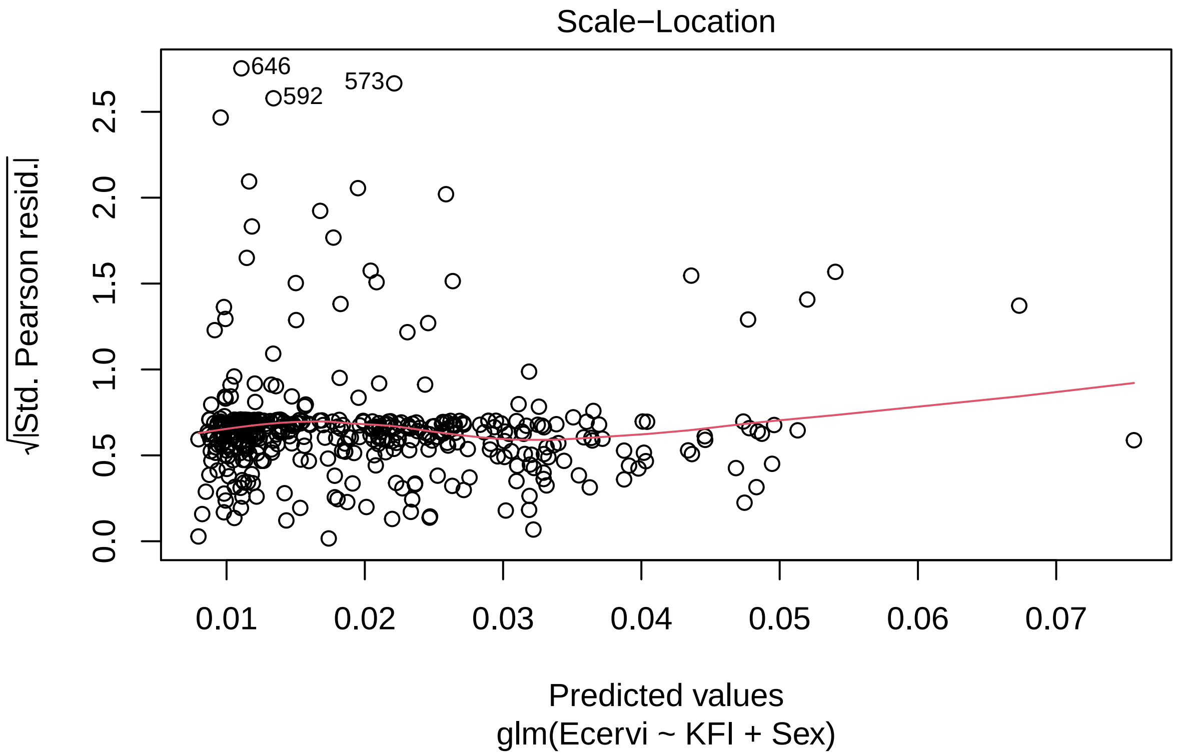

Scale−Location
√|Standardized residuals|
0.0
0.5
1.0
1.5
2.0
2.5
3.0
646
1590
592
Fitted values
lm(log(Ecervi) ~ KFI + Sex)

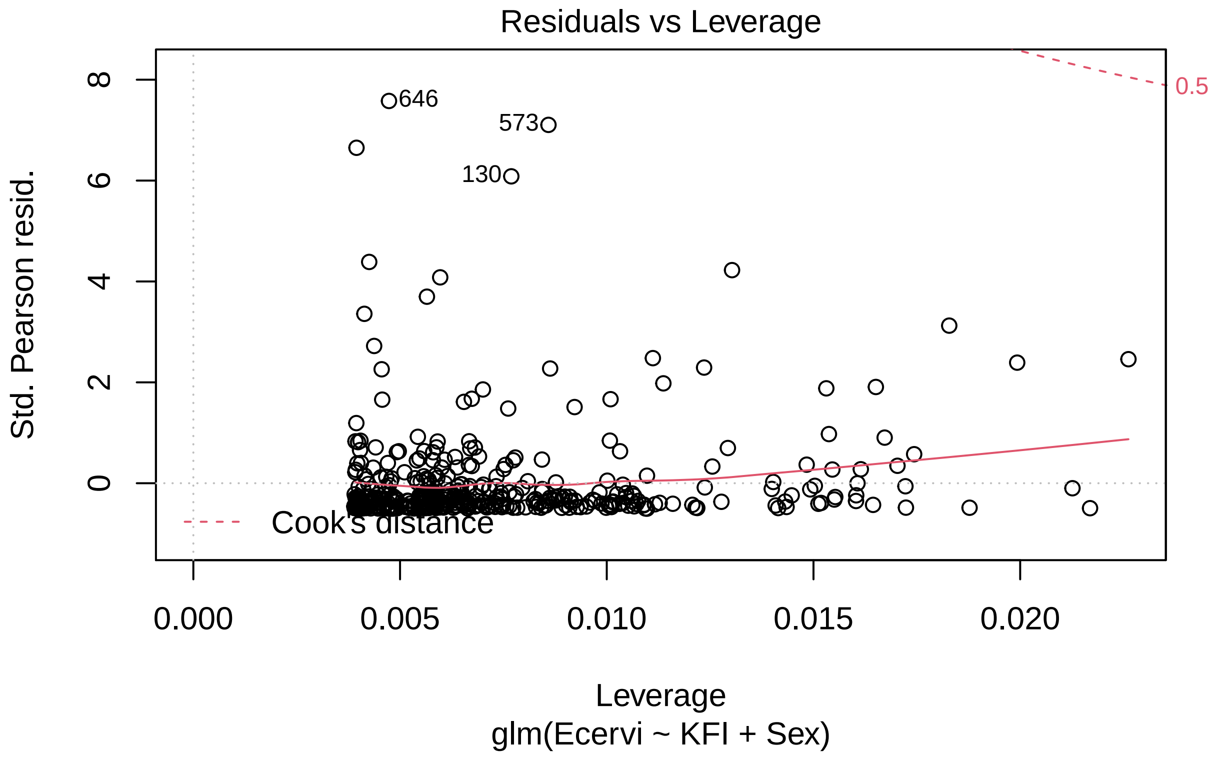


```r
ec4 <- glm(Ecervi ~ KFI + Sex, family = Gamma, data = subset(deer, Ecervi > 0))
plot(ec4)
```


Residuals vs Fitted
Residuals
15
10
5
0
646
573
592
0.01
0.02
0.03
0.04
0.05
0.06
0.07
Predicted values
glm(Ecervi ~ KFI + Sex)


Normal Q−Q
Std. Pearson resid.
0
2
4
6
8
646
573
592
−3
−2
−1
0
1
2
3
Theoretical Quantiles
glm(Ecervi ~ KFI + Sex)


Scale−Location
646
592
573
√|Std. Pearson resid.|
0.0
0.5
1.0
1.5
2.0
2.5
0.01
0.02
0.03
0.04
0.05
0.06
0.07
Predicted values
glm(Ecervi ~ KFI + Sex)



2. Model selection

Here, I will outline the general procedure for model selection in two parts:

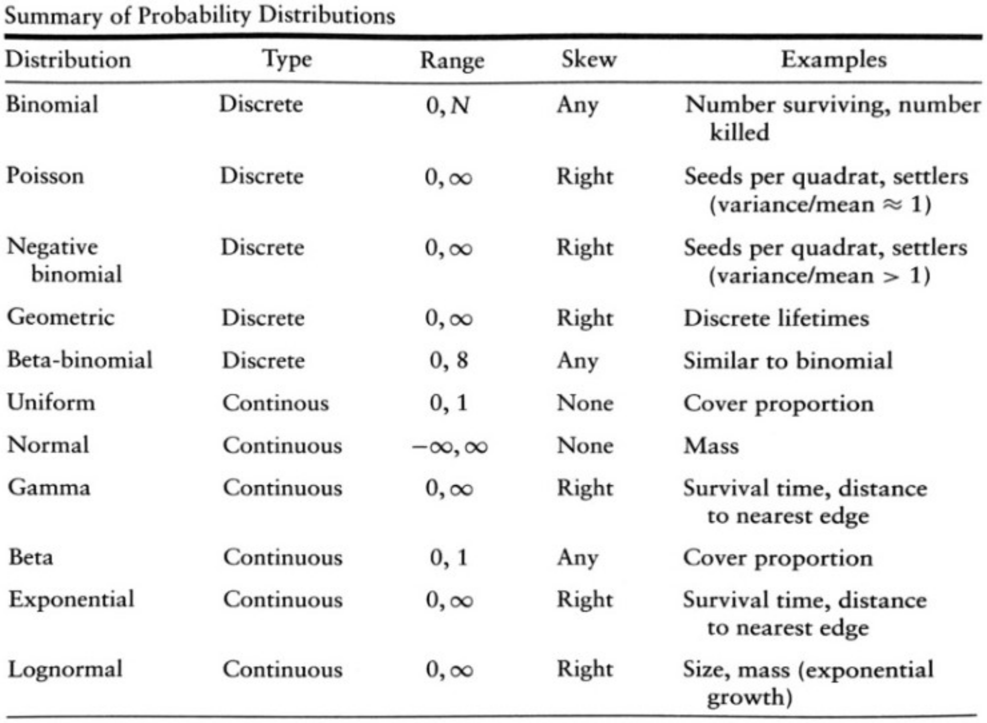
1. Select probability distribution:

a. 1st Principles

b. Analysis of data

Here, we could select our probability distribution

2. Selecting variables



Summary of Probability Distributions

Distribution	Type	Range	Skew	Examples
Binomial	Discrete	$0, N$	Any	Number surviving, number killed
Poisson	Discrete	$0, \infty$	Right	Seeds per quadrat, settlers (variance/mean $\approx$ 1)
Negative binomial	Discrete	$0, \infty$	Right	Seeds per quadrat, settlers (variance/mean > 1)
Geometric	Discrete	$0, \infty$	Right	Discrete lifetimes
Beta-binomial	Discrete	0, 8	Any	Similar to binomial
Uniform	Continous	0, 1	None	Cover proportion
Normal	Continuous	$-\infty, \infty$	None	Mass
Gamma	Continuous	$0, \infty$	Right	Survival time, distance to nearest edge
Beta	Continuous	0, 1	Any	Cover proportion
Exponential	Continuous	$0, \infty$	Right	Survival time, distance to nearest edge
Lognormal	Continuous	$0, \infty$	Right	Size, mass (exponential growth)

Figure 1: caption

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,

$$\mu_{ln(y)} = \beta_0 + \beta_1 x_1 + ...$$

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

$$ln(\mu_y) = \beta_0 + \beta_1 x_1 + ...$$

Binomial proccesses

Here, 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) or at the group-level (e.g., k events and N trials).

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. I put eggs from the mustard white butterfly (*Pieris oleracea*) in five photoperiod treatments, then measured at the individual-level whether they survived (1 for survival, 2 for died) and whether they diapaused (e.g., 1 for entering diapause, 0 for eclosing) conditioned on surviving. These data were used in Kerr et al. 2020.

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

```
diap <- read.csv(here::here("Session 2", "Data", "Diapausing_example.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
```

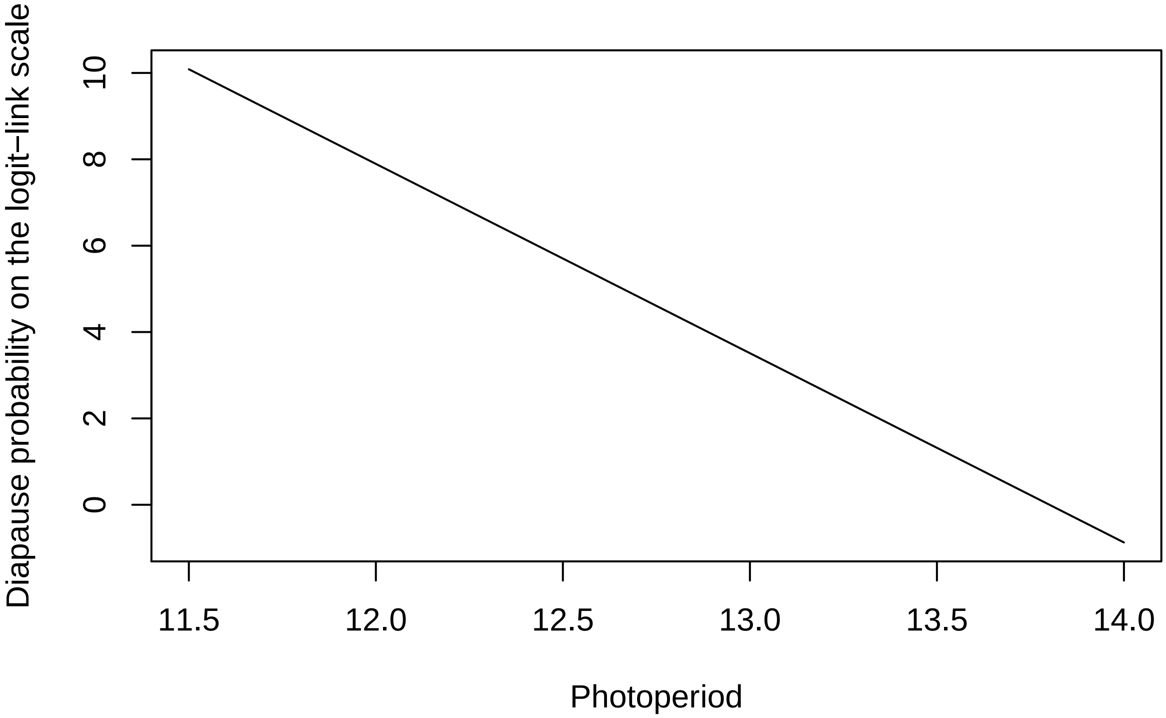
```
pd1 <- glm(Diapause ~ Treatment, family = binomial, data = subset(diap, Sex == "female")
summary(pd1)
```

```
##
## 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
```

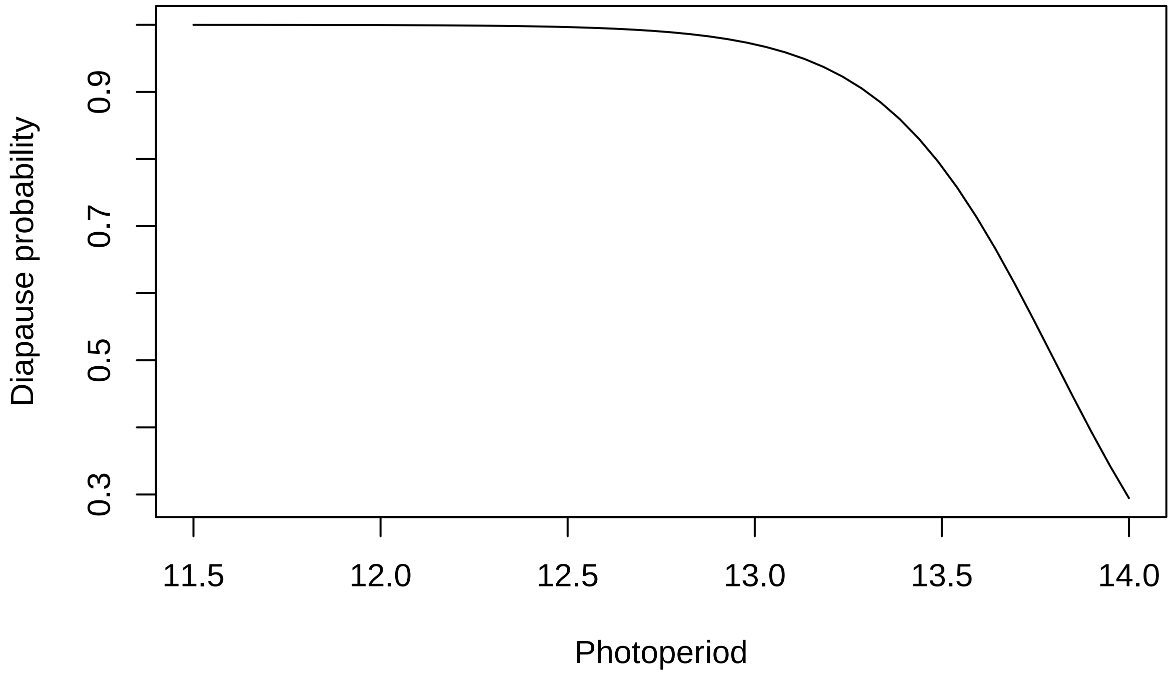
```r
Anova(pd1)
```

```
## 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
```

```r
light.vals <- seq(min(diap$Treatment), max(diap$Treatment), length.out = 50)
pd.logit <- coef(pd1)[1] + coef(pd1)[2]*light.vals
pd.pred <- exp(coef(pd1)[1] + coef(pd1)[2]*light.vals)/(1 + exp(coef(pd1)[1] + coef(pd1

plot(light.vals, pd.logit, type = "l", ylab = "Diapause probability on the logit-link sc
```



```r
plot(light.vals, plogis(pd.logit), type = "l", ylab = "Diapause probability", xlab = "Ph
```



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 $\chi^2 = nulldeviance - residualdeviance$ and the degrees of freedom $dfProposed - dfNull$.

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).

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", "OverwinterSurv_example.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 =
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
```

Practice example 5: Using the deer data, run a logit-link GLM to estimate the probability that a red deer will have a parasite given its kidney fat index (KFI) and sex.

```
deer$P.Ecervi <- deer$Ecervi
deer$P.Ecervi[deer$P.Ecervi > 0] <- 1

pec1 <- glm(P.Ecervi ~ KFI*Sex, family = binomial, data = deer)
```

Count data

Here, we will be exploring running log-link GLMs using the Poisson error distribution. 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"))
```

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

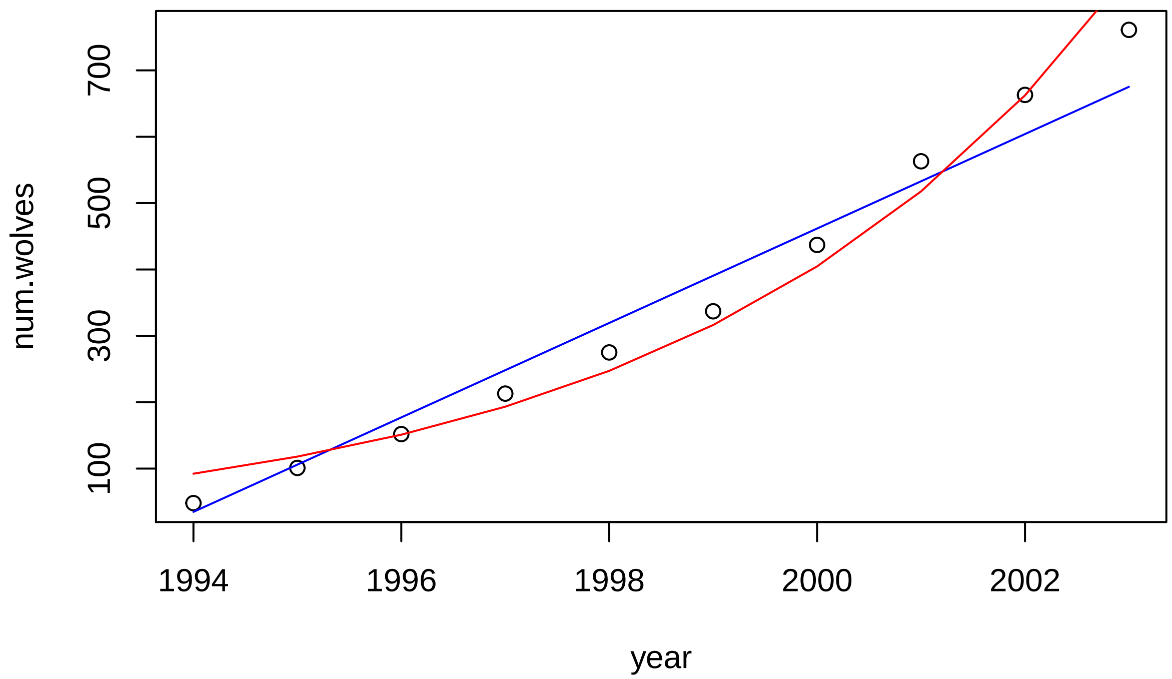
```
with(wolves[13:22,], plot(year, num.wolves))

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,])
exp(coef(wv10b)) # slope is the population growth rate
```

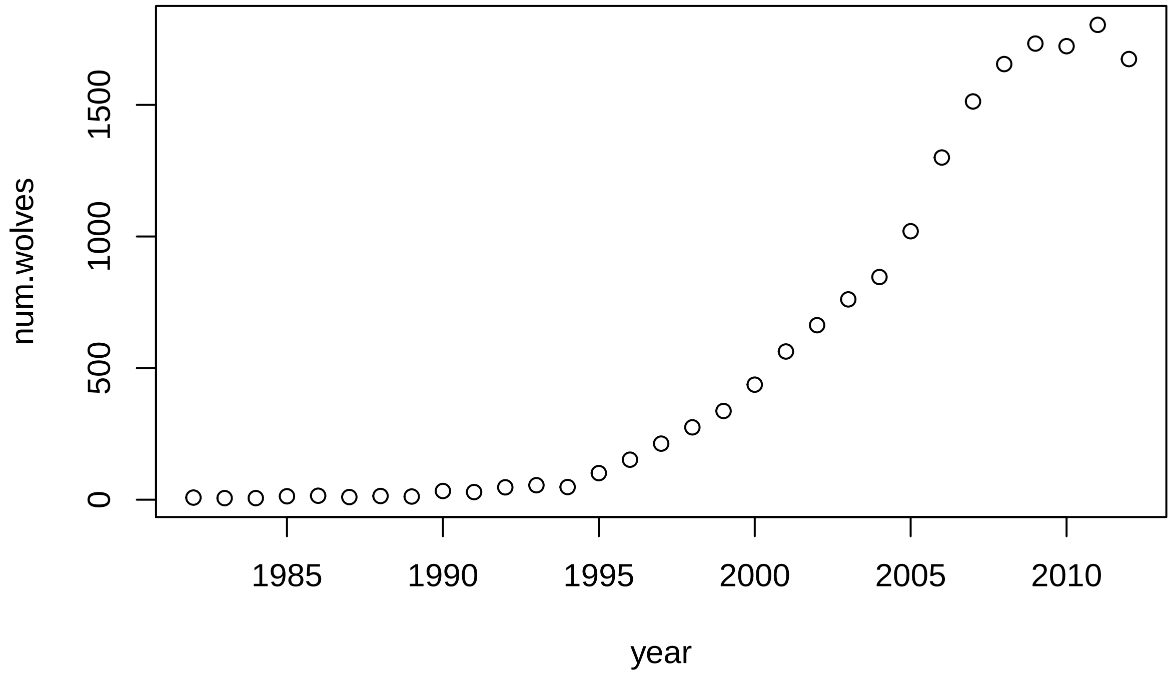
```
##   (Intercept)          year
## 4.426576e-212  1.279324e+00
```

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



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

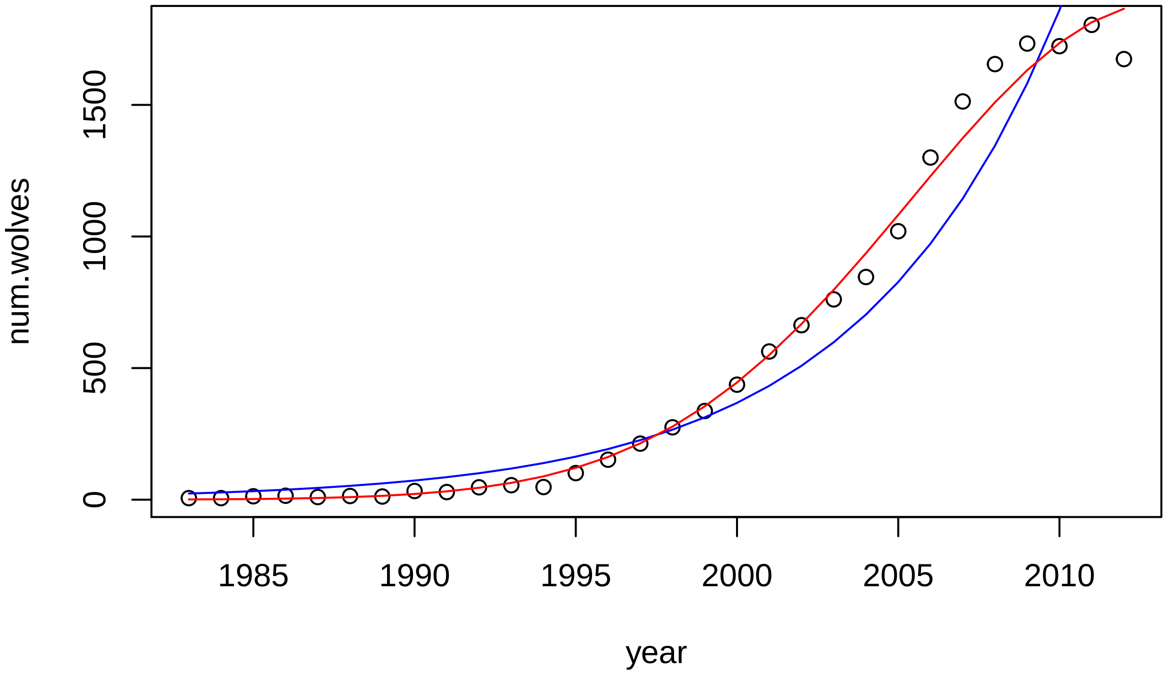
```r
with(wolves, plot(year, num.wolves))
```



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

```
##  (Intercept)         year
## -318.2374555    0.1620725
```

```
wv30b <- glm(num.wolves ~ year + I(year*year), family = poisson(link = "log"), data = wo

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



Practice example 6:

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, and (3) when count data are zero inflated.

… when counts should be rates

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

Here, we will use the Audubon Christmas bird count 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
```

```r
nf1 <- glm(Count ~ 1, offset = log(hours), family = poisson, data = flickers)
summary(nf1)
```

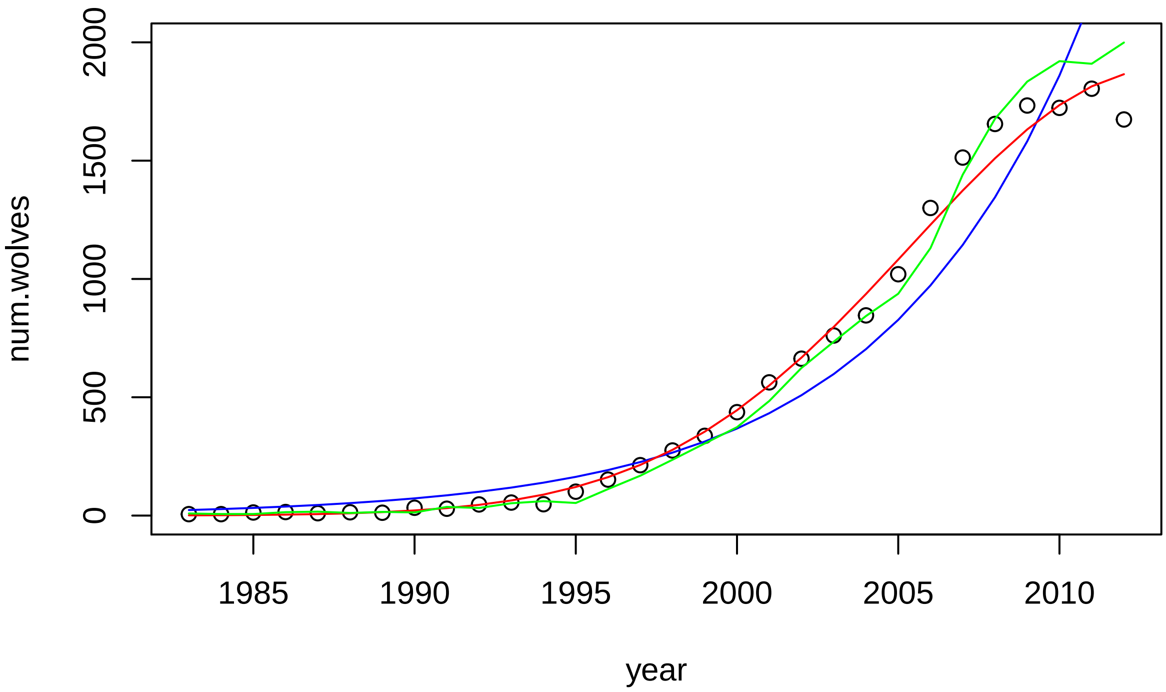
```
##
## Call:
## glm(formula = Count ~ 1, family = poisson, data = flickers, offset = log(hours))
##
## Deviance Residuals:
##    Min      1Q  Median      3Q     Max
## -5.375  -3.283  -2.691  -1.757  11.576
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -2.61801    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
```

```r
exp(coef(nf1)) # 0.07 birds per hour
```

```
## (Intercept)
##  0.07294809
```

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

```
wv30c <- glm(num.wolves[2:31] ~ 1, offset = log(num.wolves[1:30]), family = poisson(lin

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 = "blu
points(1983:2012, exp(coef(wv30b)[1] + coef(wv30b)[2]*1983:2012 + coef(wv30b)[3]*1983:2
points(1983:2012, predict(wv30c, type = "response"), type = "l", col = "green") # proce
```



Practice example 7: Similarly, we could also model the brown hare counts per area as a Poisson offset instead of mean.density using the normal, lognormal, or Gamma distribution. Do you find similar estimates for mean density for the two landuse types when using Poisson distribution compared to lognormal distribution?

```
dens12 <- glm(count1 ~ 1, offset = log(area), family = poisson, data = hares)
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
```

```
dens13 <- glm(mean.density ~ 1, family = gaussian(link = "log"), data = hares)

exp(coef(dens12)) # 4.4 brown hares per hectare
```

```
## (Intercept)
##    4.403988
```

... when data are over-dispersed or under-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 flicker counts are overdispersed, using the "intercept-only" model from the Poisson offset example #1.

```
summary(nf1)
```

```
##
## 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$deviance/nf1$df.residual # overdispersed
```

```
## [1] 15.51898
```

There are multiple ways to deal with overdispersed count data, such as observation-level random effects, Conway-Maxwell-Poisson distribution, or the negative binomial error distribution. Here, we will use the negative binomial distribution, which is the most common approach for overdispersion.

```
nf2 <- glmmTMB(Count ~ 1, offset = log(hours), family = nbinom2, data = flickers)
summary(nf2)$coefficients$cond # allows for greater variance
```

```
##              Estimate Std. Error   z value     Pr(>|z|)
## (Intercept) -2.893406  0.3595499 -8.047302 8.463933e-16
```

```
summary(nf1)$coefficients
```

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

Undispersion is more common 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. Roulin and Bersier (2007) looked at nestlings' response to the presence of the mother and the father. 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.

Here, we want to evaluate whether food treatment impacts the brood size of barn owls.

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

bs1 <- glm(BroodSize ~ FoodTreatment, family = poisson, data = owls)
summary(bs1)
```

```
##
## Call:
## glm(formula = BroodSize ~ FoodTreatment, family = poisson, data = owls)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.9861  -0.2262  -0.1583   0.3166   1.1794
##
## Coefficients:
##                         Estimate Std. Error z value Pr(>|z|)
## (Intercept)              1.46441    0.02688  54.479   <2e-16 ***
## FoodTreatmentSatiated  0.03287    0.03904   0.842      0.4
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 191.03  on 598  degrees of freedom
## Residual deviance: 190.32  on 597  degrees of freedom
## AIC: 2182.1
##
## Number of Fisher Scoring iterations: 4
```

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

```
## [1] 0.3187986
```

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

```r
bs2 <- glmmTMB(BroodSize ~ FoodTreatment, family = compois, data = owls)
```

Practice example X:

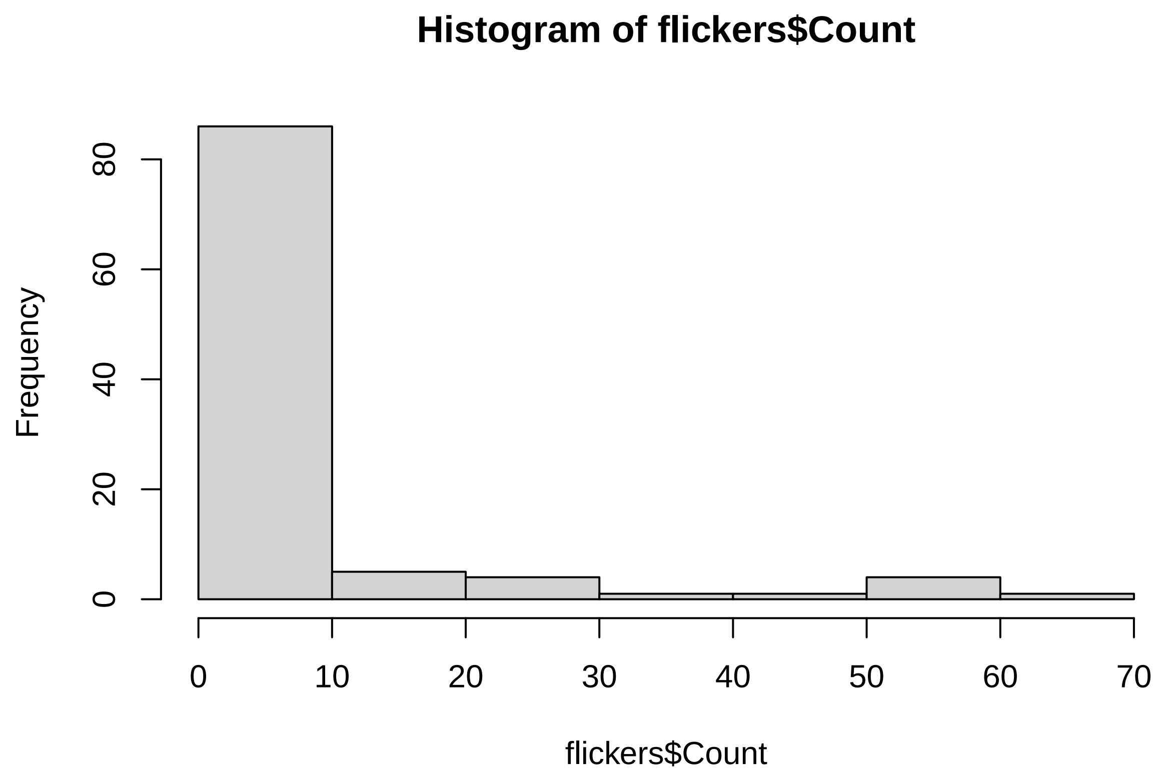
… when counts are zero-inflated

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).

> pscl::zerinfl(y ~ x_count | x_zero)

```r
hist(flickers$Count) # excess zeros
```



```
ZiP1 <- zeroinfl(Count ~ 1 | 1, offset = log(hours), dist = "negbin", data = flickers)
summary(ZiP1)
```

```
##
## Call:
## zeroinfl(formula = Count ~ 1 | 1, data = flickers, offset = log(hours),
##     dist = "negbin")
##
## Pearson residuals:
##     Min      1Q  Median      3Q     Max
## -0.3980 -0.3932 -0.3896 -0.3703  4.1963
##
## Count model coefficients (negbin with log link):
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -1.5615     0.2009  -7.771 7.76e-15 ***
## Log(theta)    0.1100     0.3954   0.278    0.781
##
## Zero-inflation model coefficients (binomial with logit link):
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)    1.033      0.245   4.215  2.5e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Theta = 1.1163
## Number of iterations in BFGS optimization: 7
## Log-likelihood: -157.2 on 3 Df
```

```
ZiP2 <- zeroinfl(Count ~ Latitude | 1, offset = log(hours), dist = "negbin", data = flic
summary(ZiP2)
```

```
##
## Call:
## zeroinfl(formula = Count ~ Latitude | 1, data = flickers, offset = log(hours),
##     dist = "negbin")
##
## Pearson residuals:
##     Min      1Q  Median      3Q     Max
## -0.5115 -0.4968 -0.4223 -0.2140  3.9332
##
## Count model coefficients (negbin with log link):
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  36.2123     6.8787   5.264 1.41e-07 ***
## Latitude     -0.8959     0.1627  -5.506 3.67e-08 ***
## Log(theta)    0.9399     0.4046   2.323   0.0202 *
##
## Zero-inflation model coefficients (binomial with logit link):
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)   0.8865     0.2584    3.43 0.000603 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Theta = 2.5598
## Number of iterations in BFGS optimization: 19
## Log-likelihood: -145.1 on 4 Df
```

```
Anova(ZiP2)
```

```
## Analysis of Deviance Table (Type II tests)
##
## Response: Count
##          Df  Chisq Pr(>Chisq)
## Latitude  1 30.317  3.668e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
ZiP3 <- zeroinfl(Count ~ Latitude | Latitude, offset = log(hours), dist = "negbin", data
```

Note that the 'glmmTMB' package allows for both zeroinflated mixed models and hurdle mixed models.

```
ZiP4 <- glmmTMB(Count ~ Latitude, offset = log(hours), ziformula = ~1, family = "nbinom2
summary(ZiP4)$coefficients
```

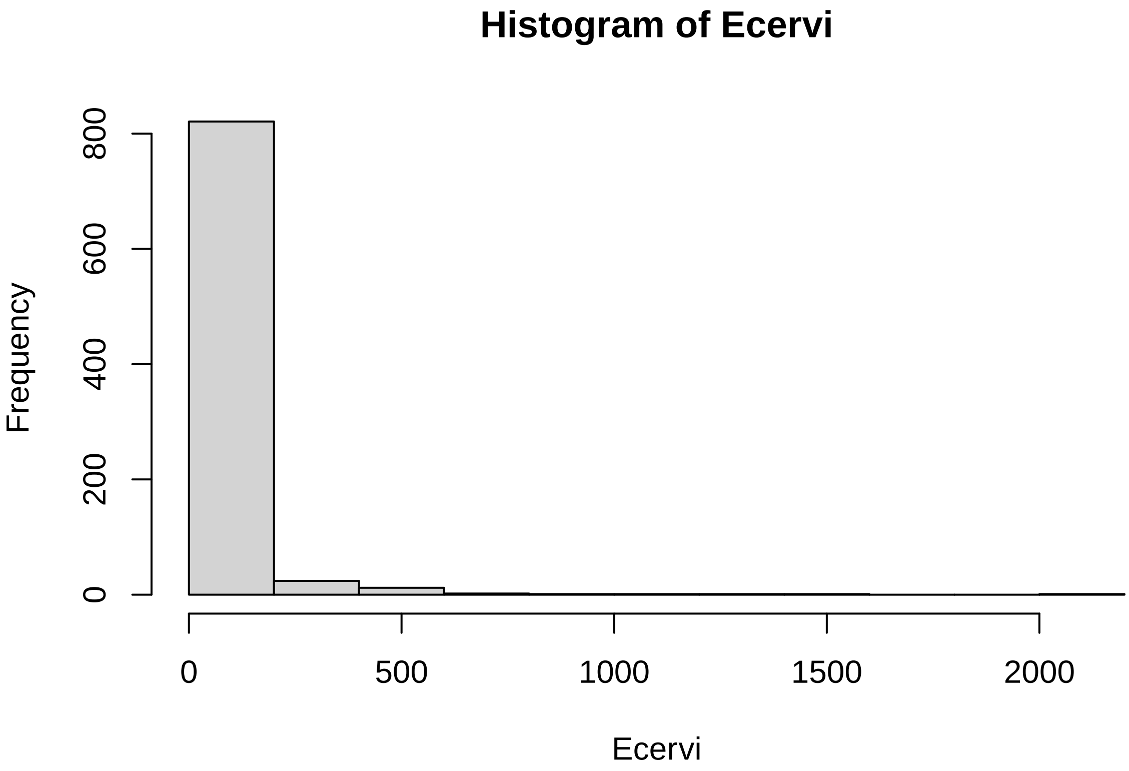
```
## $cond
##               Estimate Std. Error   z value     Pr(>|z|)
## (Intercept) 36.2124071  6.8780654  5.264912 1.402567e-07
## Latitude    -0.8959298  0.1626989 -5.506675 3.656741e-08
##
## $zi
##              Estimate Std. Error  z value     Pr(>|z|)
## (Intercept) 0.8864842  0.2584243 3.430344 0.0006028174
##
## $disp
## NULL
```

```
ZiP5 <- glmmTMB(Count ~ Latitude, offset = log(hours), ziformula = ~., family = "nbinom2
summary(ZiP5)$coefficients
```

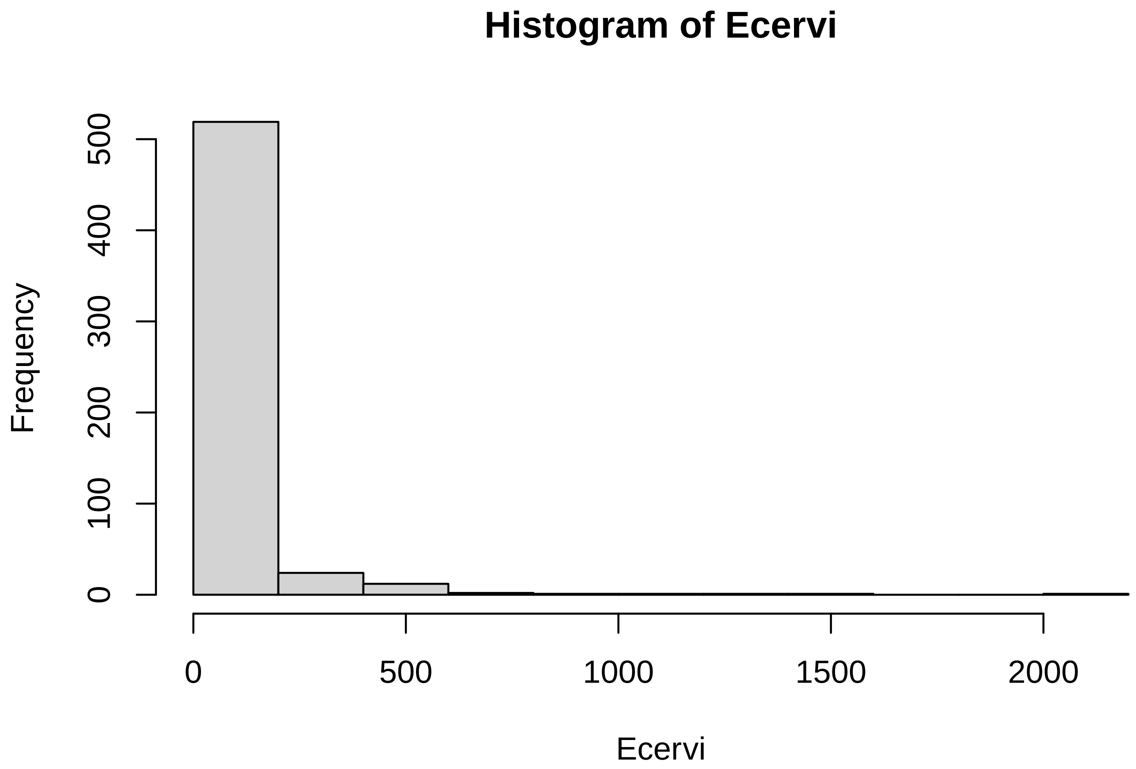
```
## $cond
##               Estimate Std. Error   z value     Pr(>|z|)
## (Intercept) 31.7812147  6.2946235  5.048946 4.442540e-07
## Latitude    -0.7902148  0.1488927 -5.307276 1.112756e-07
##
## $zi
##                Estimate Std. Error   z value   Pr(>|z|)
## (Intercept) -22.3363235  9.9557076 -2.243570 0.02486010
## Latitude      0.5447831  0.2337981  2.330143 0.01979857
##
## $disp
## NULL
```

For hurdle models, the nonzero and zero processes are modelled separately, and therefore, we assume all zeros are true zeros. 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. Here, let's evaluate whether the probability of infection dependent on sex and the infection intensity is dependent on kidney fat index (KFI).

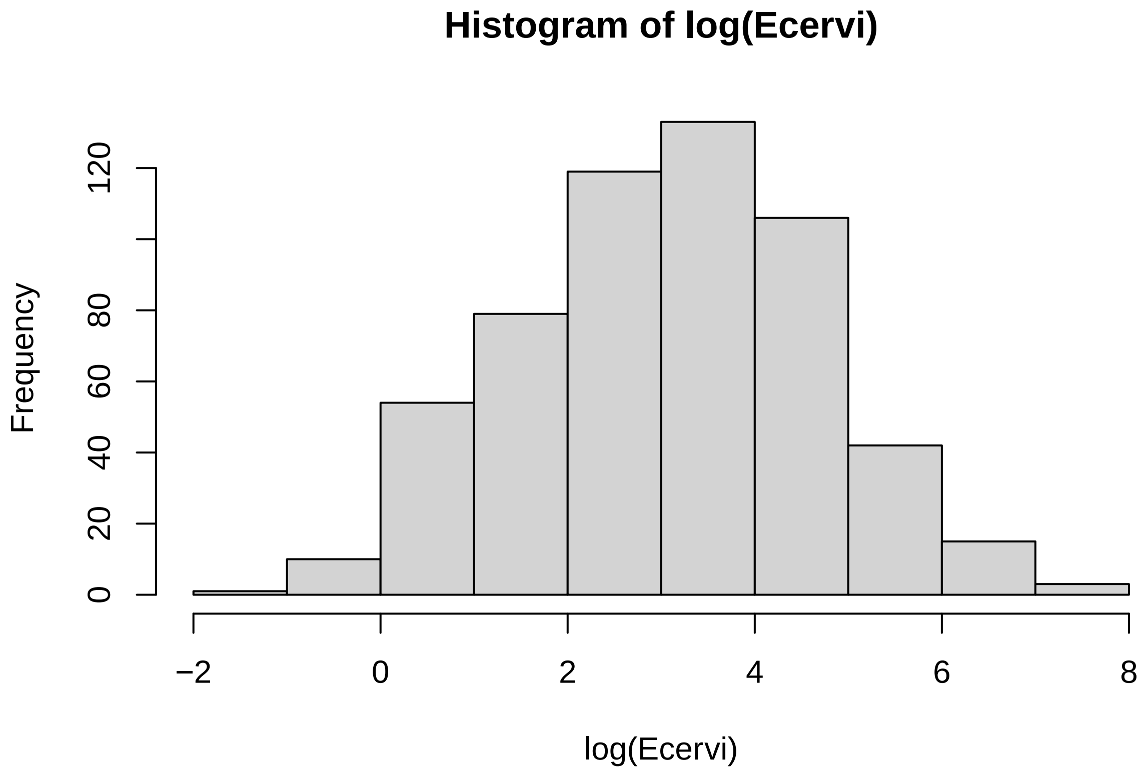
```r
with(deer, hist(Ecervi)) # zero-inflated
```



```r
with(subset(deer, P.Ecervi == 1), hist(Ecervi))
```



```r
with(subset(deer, P.Ecervi == 1), hist(log(Ecervi)))
```



```r
h_zero <- glm(P.Ecervi ~ Sex, family = binomial, data = deer)
Anova(h_zero)
```

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

```r
coef(h_zero)
```

```
## (Intercept)         Sex
##   1.3319519  -0.4826007
```

```r
h_nzero <- glm(log(Ecervi) ~ KFI, family = gaussian, data = subset(deer, P.Ecervi == 1))
Anova(h_nzero)
```

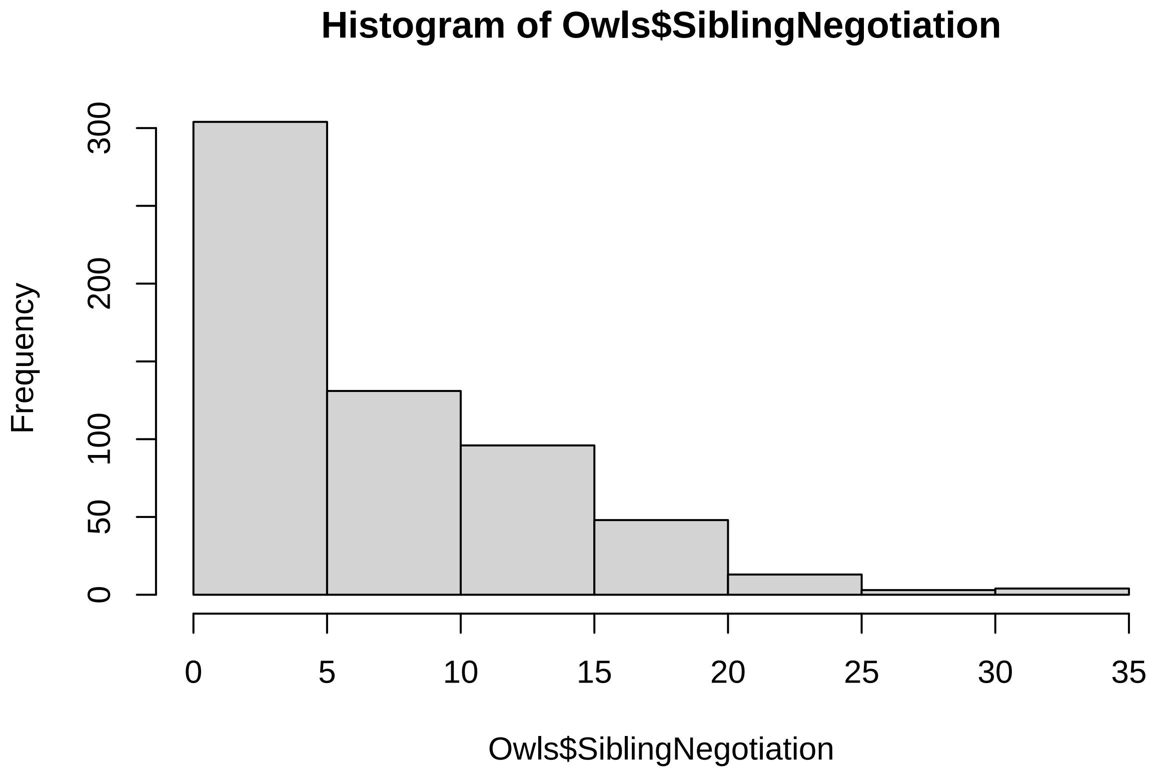
```
## Analysis of Deviance Table (Type II tests)
##
## Response: log(Ecervi)
##     LR Chisq Df Pr(>Chisq)
## KFI   9.4023  1   0.002167 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
summary(h_nzero)
```

```
##
## Call:
## glm(formula = log(Ecervi) ~ KFI, family = gaussian, data = subset(deer,
##     P.Ecervi == 1))
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -3.3964  -1.1184   0.0061   1.0151   4.2076
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  3.178997   0.113517  28.005  < 2e-16 ***
## KFI         -0.008300   0.002707  -3.066  0.00231 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for gaussian family taken to be 2.443036)
##
##     Null deviance: 1034.4  on 415  degrees of freedom
## Residual deviance: 1011.4  on 414  degrees of freedom
##   (146 observations deleted due to missingness)
## AIC: 1556.1
##
## Number of Fisher Scoring iterations: 2
```

A Gamma distribution cannot be overdispersed. The glmmTMB package also allows for hurdle models for count data using the "truncated_poisson"

```
hist(Owls$SiblingNegotiation)
```



```r
#h_mod <- glmmTMB(SiblingNegotiation ~ BroodSize, family = truncated)
```

Practice example 9: Similar to the Northern Flickers, the Eastern bluebird data was also collected across New England from the Audubon Christmas bird count. Eastern Bluebirds have breeding grounds in the northern half of the United States, but all year grounds south of Massachusetts down to Texas and exclusive wintering grounds in Texas and northern Mexico.

Using the "bluebird.csv" data, choose whether a zeroinflated or hurdle model of bird counts with an offset of observation hours is more appropriate for these data. Evaluate both latitude on the zero (i.e., probability that it has migrated) and nonzero (i.e., # of all-year resident birds) process parts.

5. Mixed models

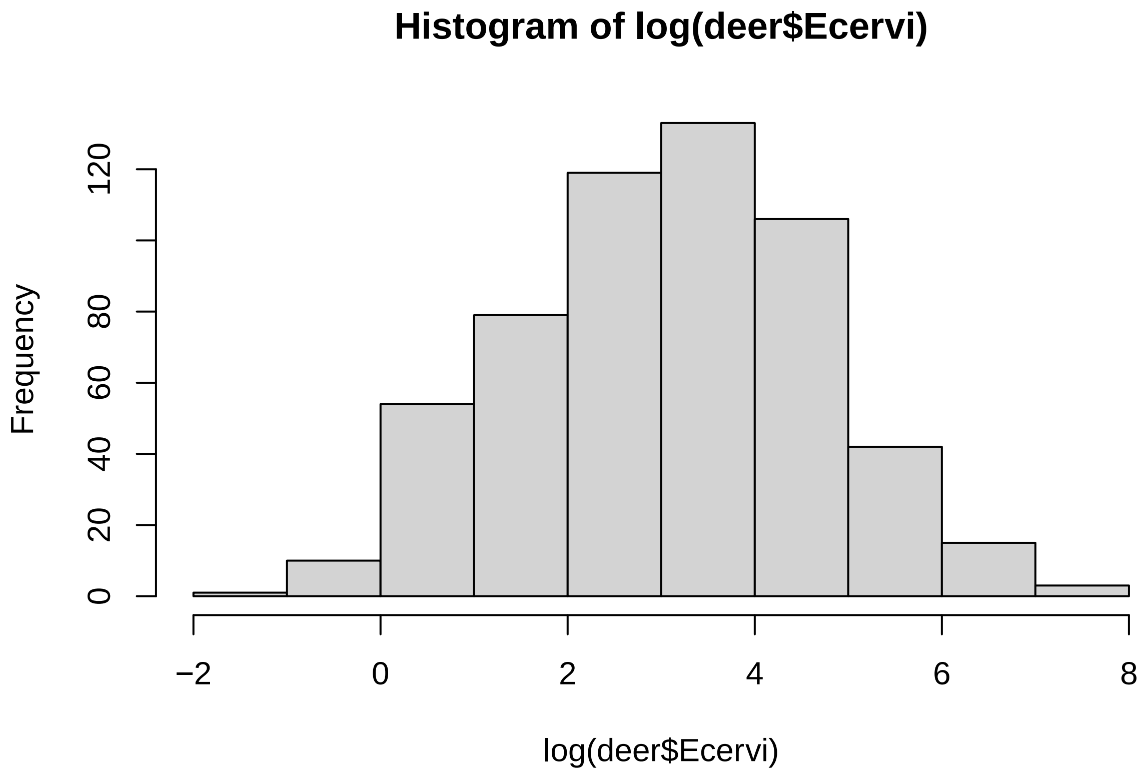
Here, we evaluated models with fixed effects. Fixed effects are the covariates of interest. Here, we will incorporate models that evaluate both fixed and random effects called "**mixed models**".

Random-coefficient models

First, we will look at the

Here, we will run mixed models for the deer data when looking only at the

```
hoppers <- read.csv(here::here("Session 2", "Data", "GrasshopperSong.csv"))
hist(log(deer$Ecervi))
```



```
#ri1 <- glmmTMB(log(Ecervi) ~ KFI + (1|Farm), family = gaussian, data = subset(deer, !
#summary(ri1)
```

```
#ri2 <- glmmTMB(log(Ecervi) ~ KFI + (0 + KFI |Farm), family = gaussian, data = deer)
#summary(ri2)
```

```
#ri3 <- glmmTMB(log(Ecervi) ~ KFI + (1|Farm) + (0 + KFI |Farm), family = gaussian, dat
```

```
#ri4 <- glmmTMB(log(Ecervi) ~ KFI + (0 + KFI |Farm), family = gaussian, data = deer)
```

Practice example 10:

Nested random effects

Practice example 11:

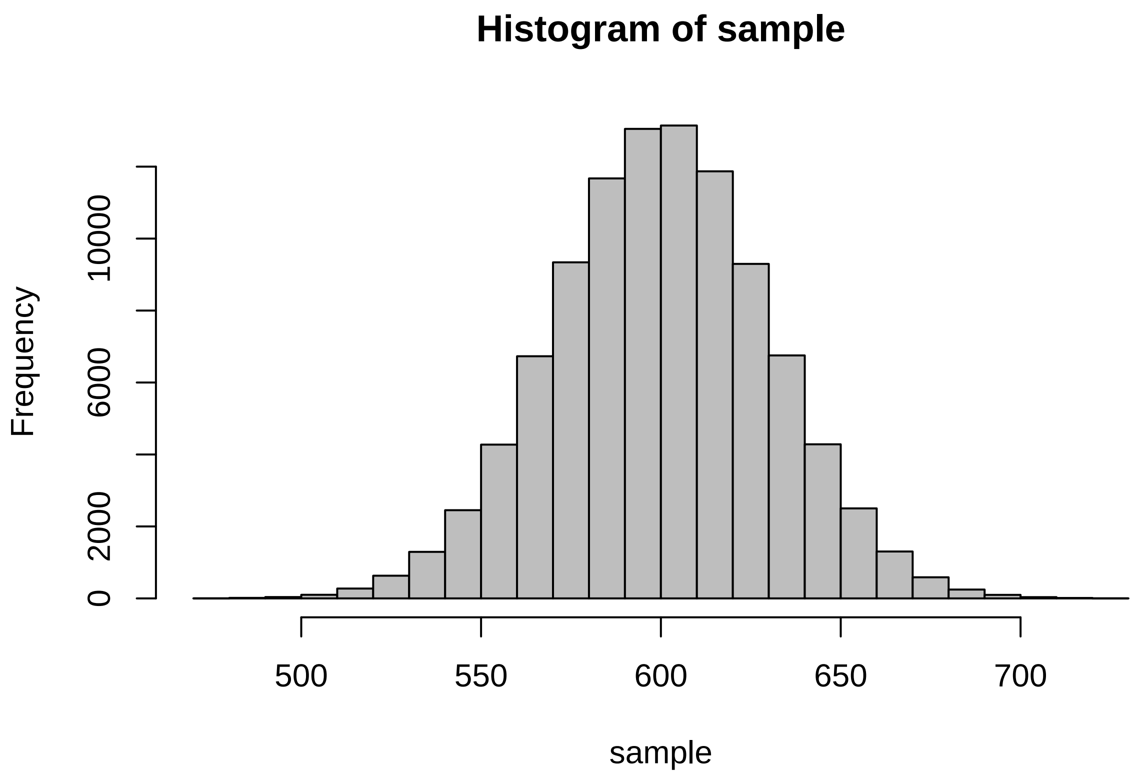
6. Intro to simulating data

Normal distribution and how to run linear models in R.

```
n <- 100000 # sample size
mu <- 600 # mean body mass of male pelegrines
sd <- 30 # SD of body size

sample <- rnorm(n = n, mean = mu, sd = sd)
head(sample) # vector of randomly generated numbers
```

```
## [1] 551.8464 591.8167 581.2525 607.5042 526.3999 557.4761
```

```
hist(sample, col = "grey")
```



```
dnorm(x = 650, mean = mu, sd = sd)
```

```
## [1] 0.003315905
```

Sources

- Marc Kery (2010) Introduction to WinBUgS for Ecologists. Academic Press, Burlinton.
- Zuur, A.F, E.N. Ieno, and E. Meesters. (2009) A Beginner's Guide to R.