Generalized Linear Models

Melissa Guzman

March 16, 2016

Etherpad:

https://public.etherpad-mozilla.org/p/404GLMs

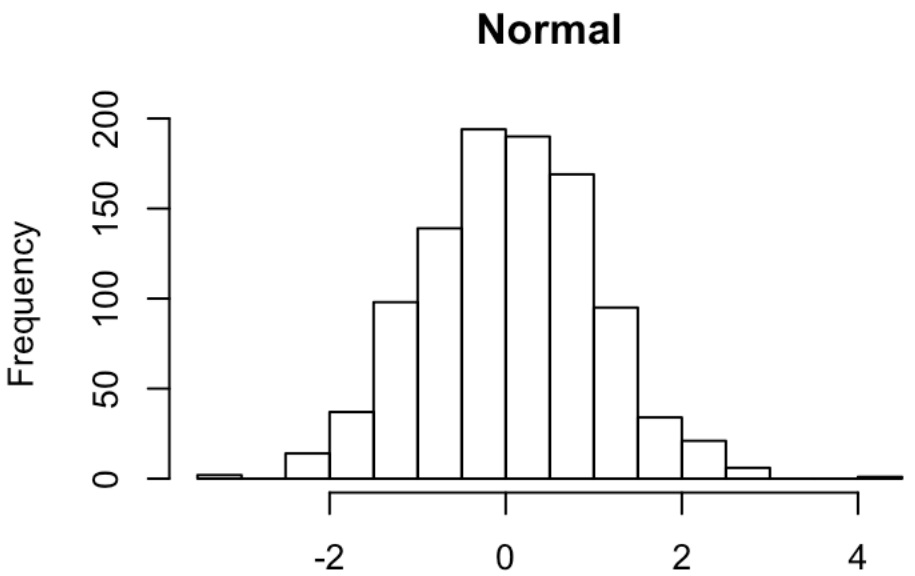
Link to all material:

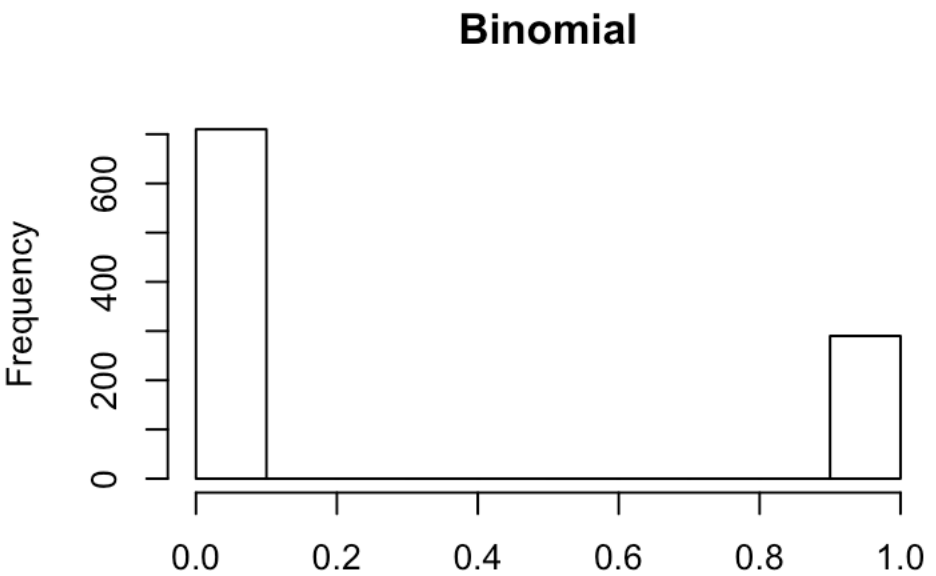
https://github.com/lmguzman/GLMs

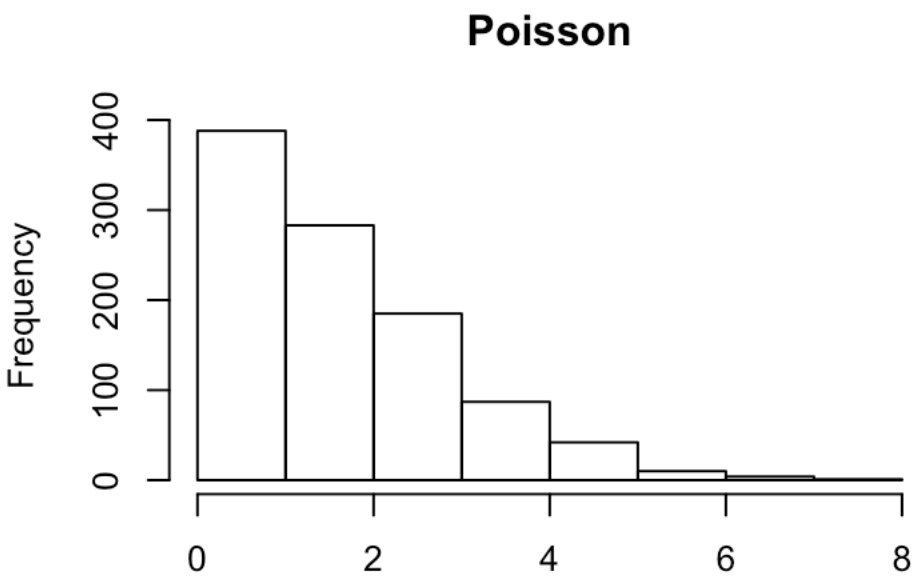
What is a frequency distribution?

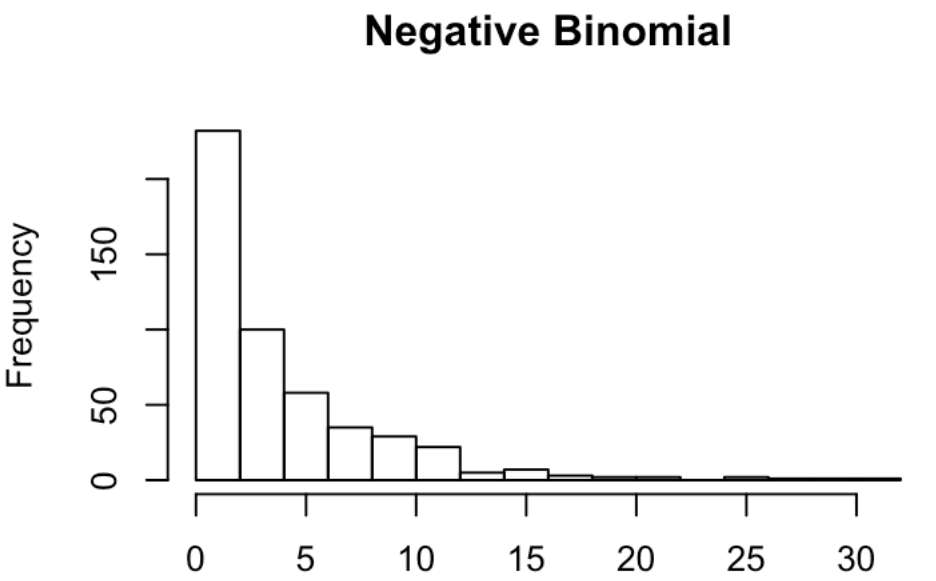
What frequency distributions can you remember?

Frequency Distributions

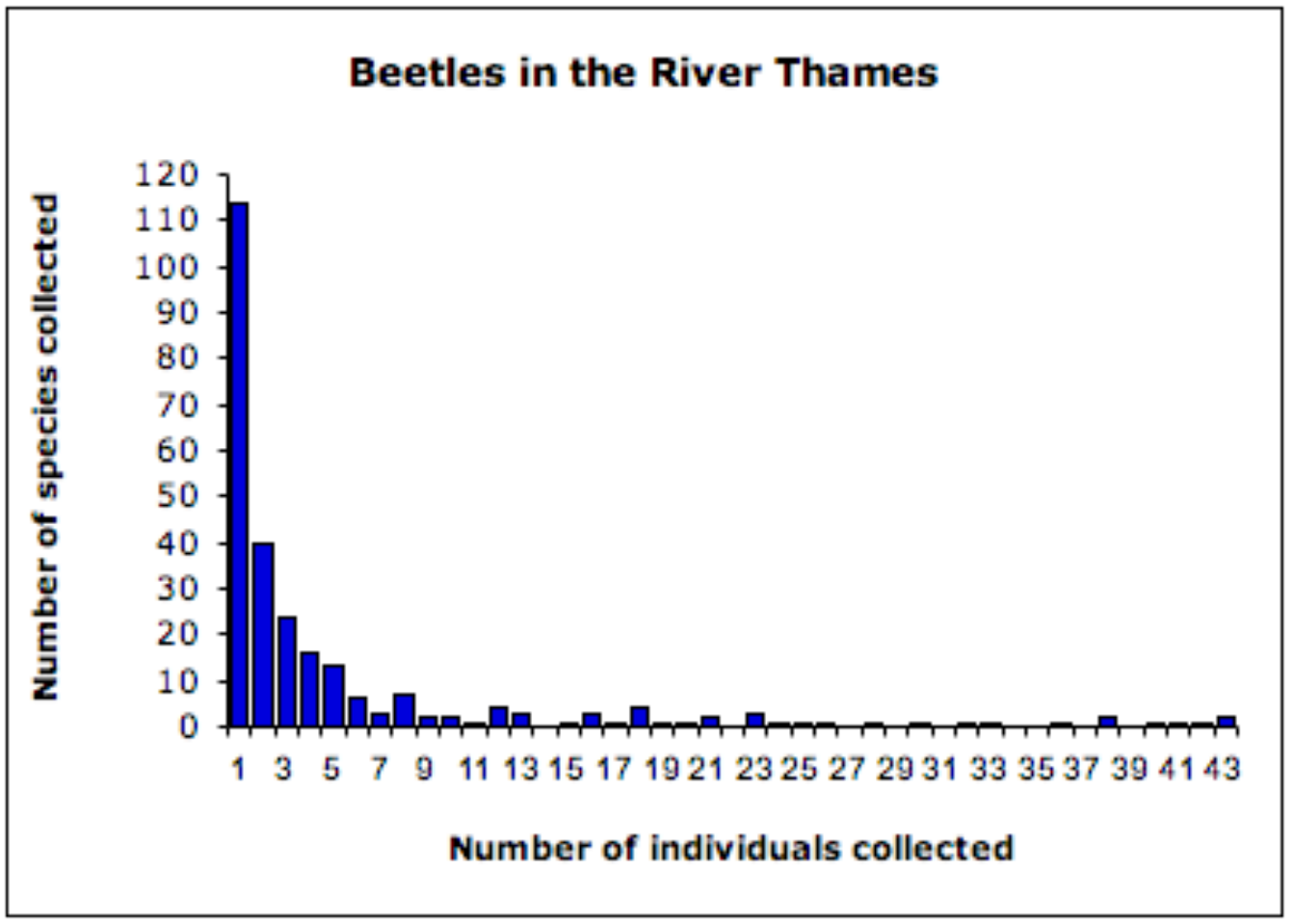








Why do we care?



Relative species abundance of beetles sampled from the river Thames collected by C.B. Williams (1964). (Magurran 2004)

Learning Objectives

By the end of this lesson the students will:

- Differentiate and categorize GLMs vs ANOVAS and regressions
- Identify the different components of GLMs
- Select the most appropriate GLM
- Carry out a GLM

Generalised Linear Models (GLM)

Linear Models (LM)

"Errors" are normally distributed

Multiple regression

ANOVA

t-tests

"Errors" are not normally distributed

e.g.

- **Binomial**
- **Negative binomial**
- **Poisson**
- **etc.**

LO: Differentiate and categorize GLMs vs ANOVAS and regressions

When do we use them?

Used when the *residuals* from a linear model are not-normal

Whilst it is the error distribution (the residuals) that is important, the distribution of the dependent variable has a strong influence on this.

General formula

$$y = g(\beta_0 + \beta_1 X_1 + \beta_2 X_2 + \ldots + \beta_i X_i) + \varepsilon$$

X are your explanatory variables

y your response variable

β are the coefficients

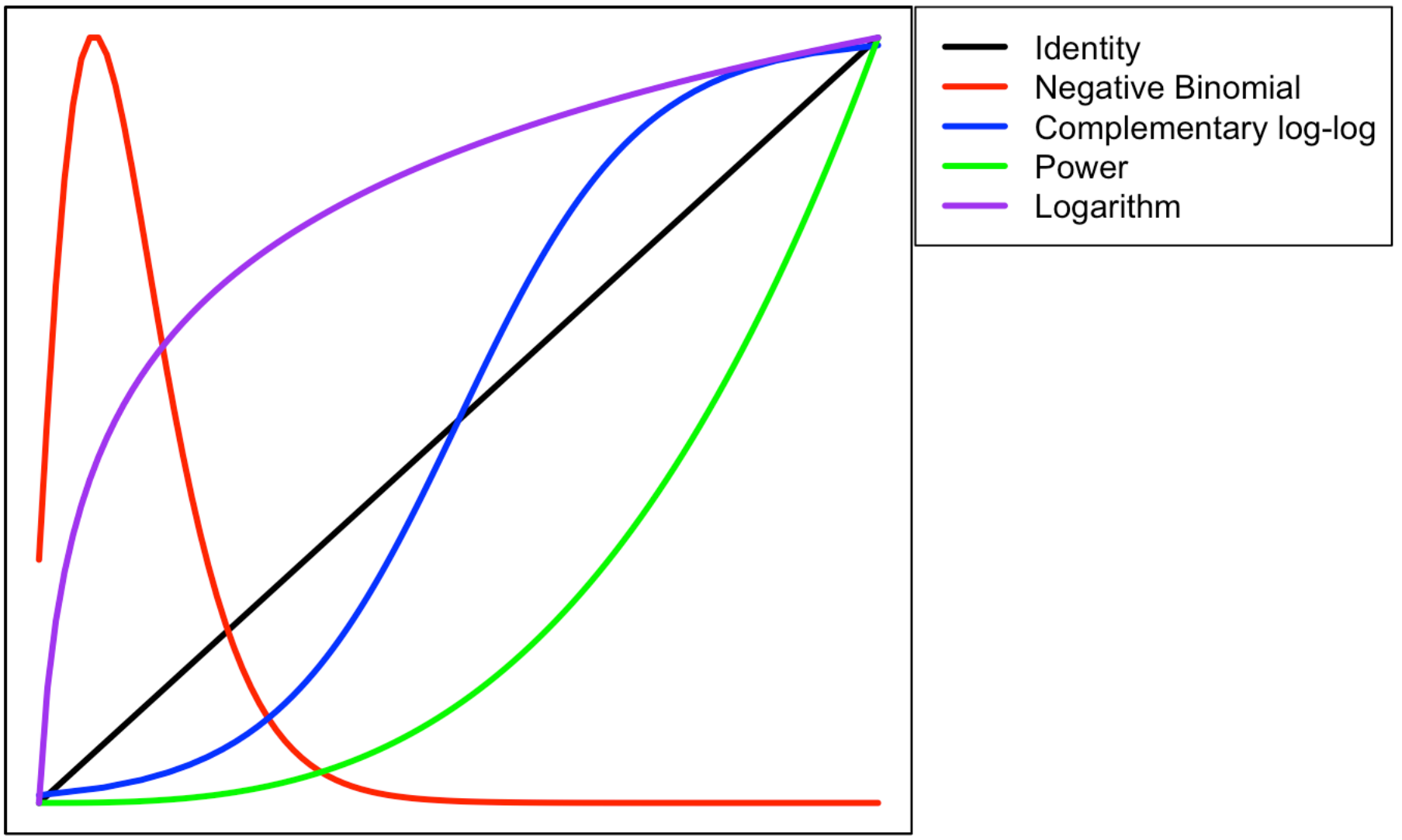
ε is the error term

Where g is a function (called the link function) which transforms each value of y in relation to the linear predictors (the variables and their coefficients) i.e. the link function transforms the dependent variable within the model.

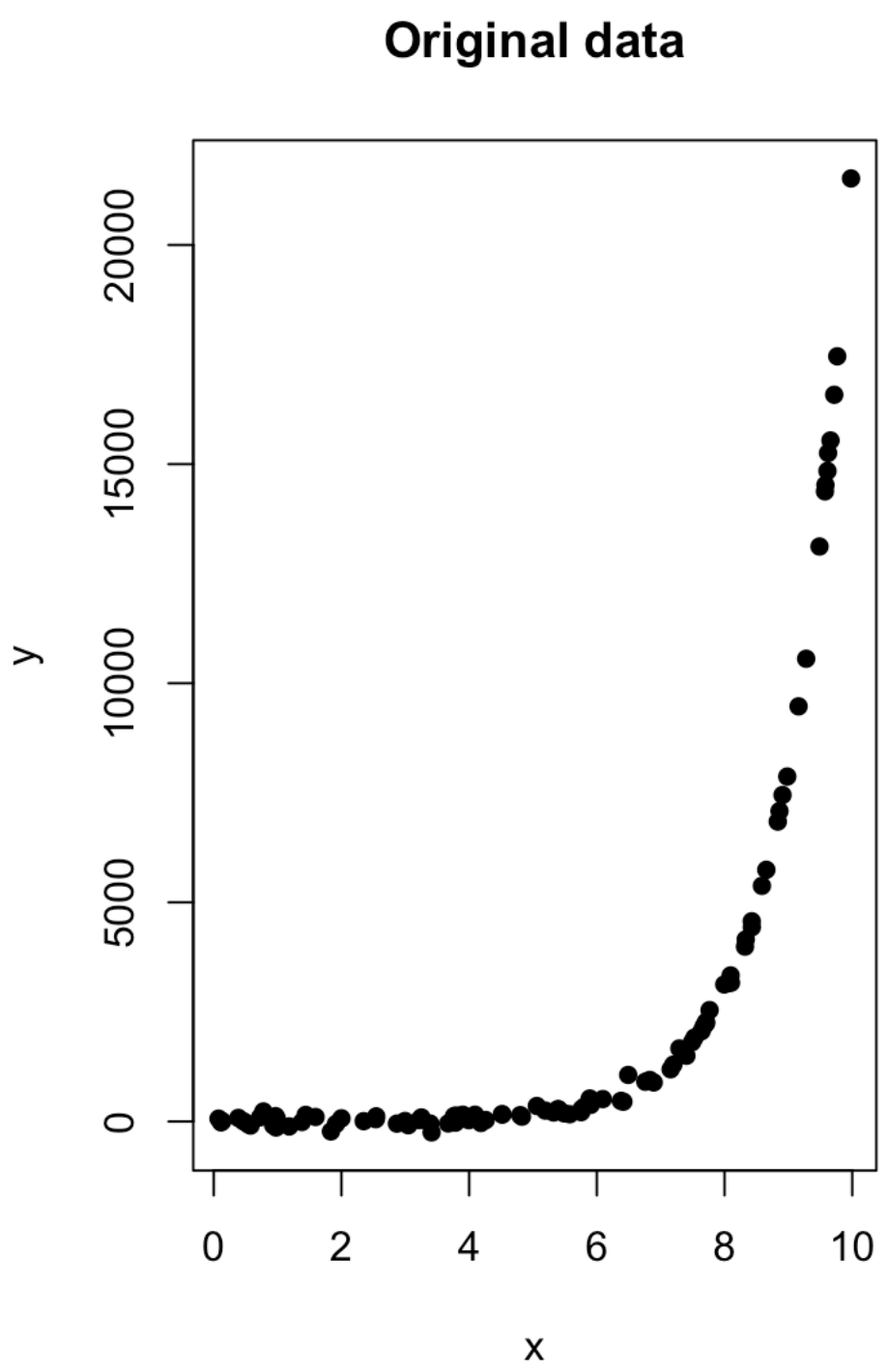
LO: Identify the different components of GLMs

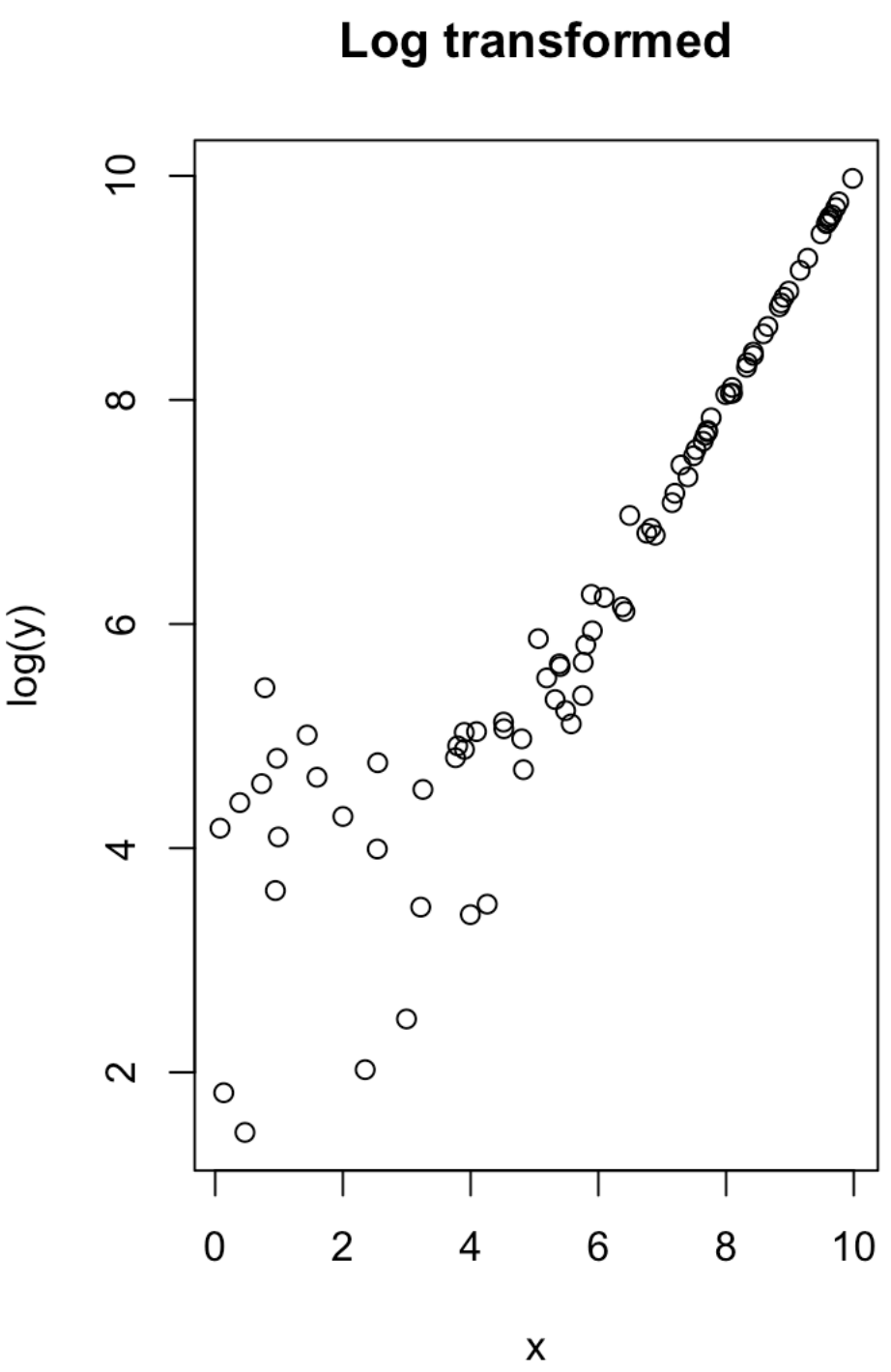
What is the link function?

Link Functions: defining the shape of the relationship between the dependent & independent variables.



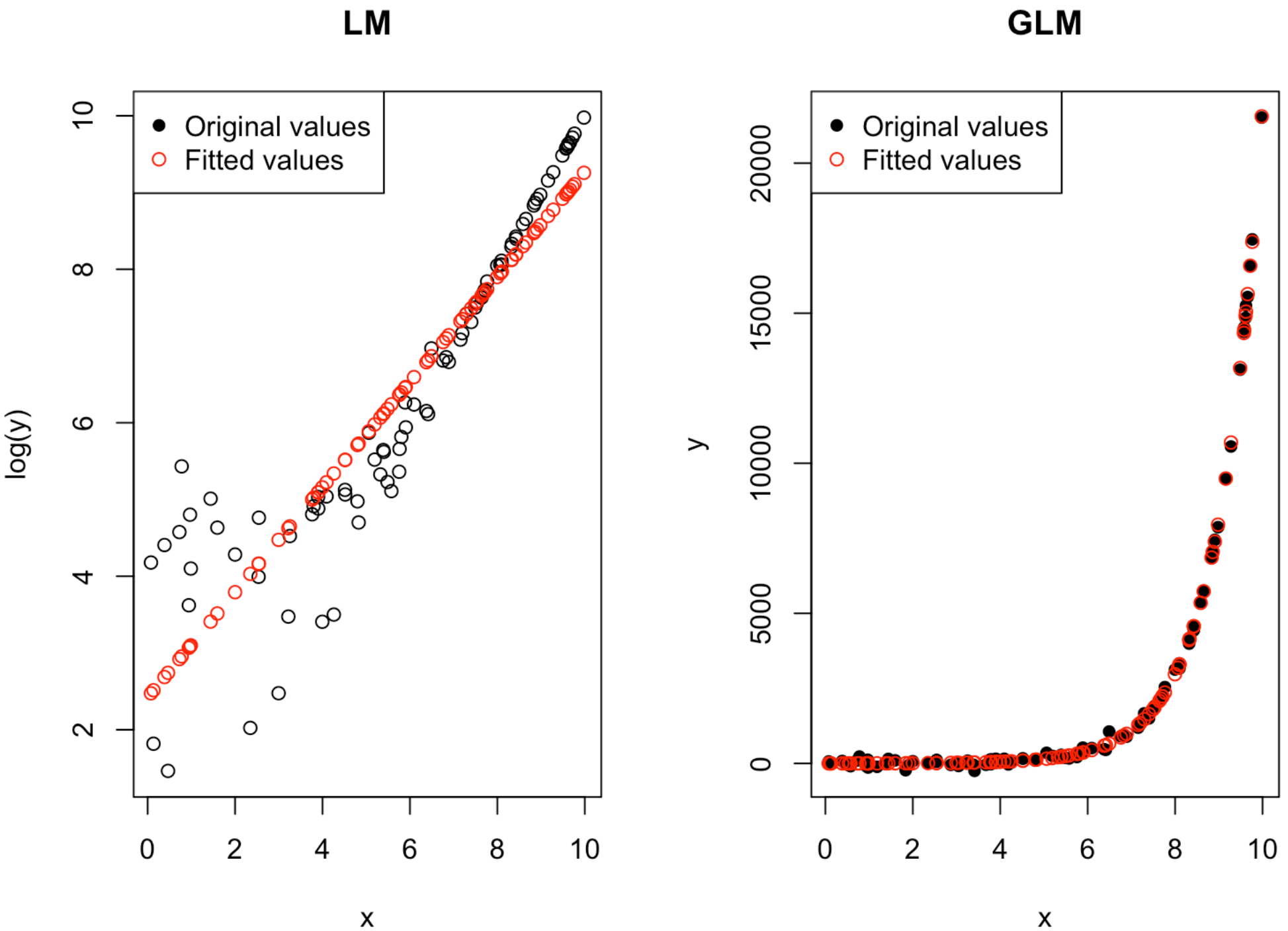
Why use a link function instead of transforming the data?





Let's fit some models to that data

```
model1 <- lm(log(y)~x)
model2 <- glm(y~x, family = gaussian (link = "log"), start=c(1, 1))
```



Using a link function is often better than simply transforming the dependent variable, because it doesn't simply attempt to make the variance constant. Instead, it fits a function that "adjusts" the variance as the data is being fitted.

What links should I use?

Families	Links
gaussian (c)	identity, log, inverse
inverse.gaussian (c)	$1/\mu^2$, inverse, identity, log
gamma (c)	inverse, identity, log
quasi (c)	logit, probit, cloglog, identity, inverse, $1/\mu^2$, sqrt, power
poisson (d)	log, identity, sqrt
quasipoisson (d)	identity, logit, probit cloglog
negative binomial (d)	log, sqrt, identity *use glm.nb function from MASS library*
binomial (d)	logit, probit, cauchit, log, cloglog
quasibinomial (d)	identity, logit, probit, cloglog

d = discrete, c = continous

First link is the default

So many options… How do I choose?

Start with the default link function for the family error term that you have chosen. E.g. for family = binomial, the "logit" is the default link function.

The "correct" combination of family and link function is arrived at by trial and error, comparing

- residual distributions (aiming for normal)
- adjusted R2 values (where relevant, higher is better)
- AIC values (lower is better) of different nested models

LO: Select the most appropriate GLM

Challenge

How do I do it in R?

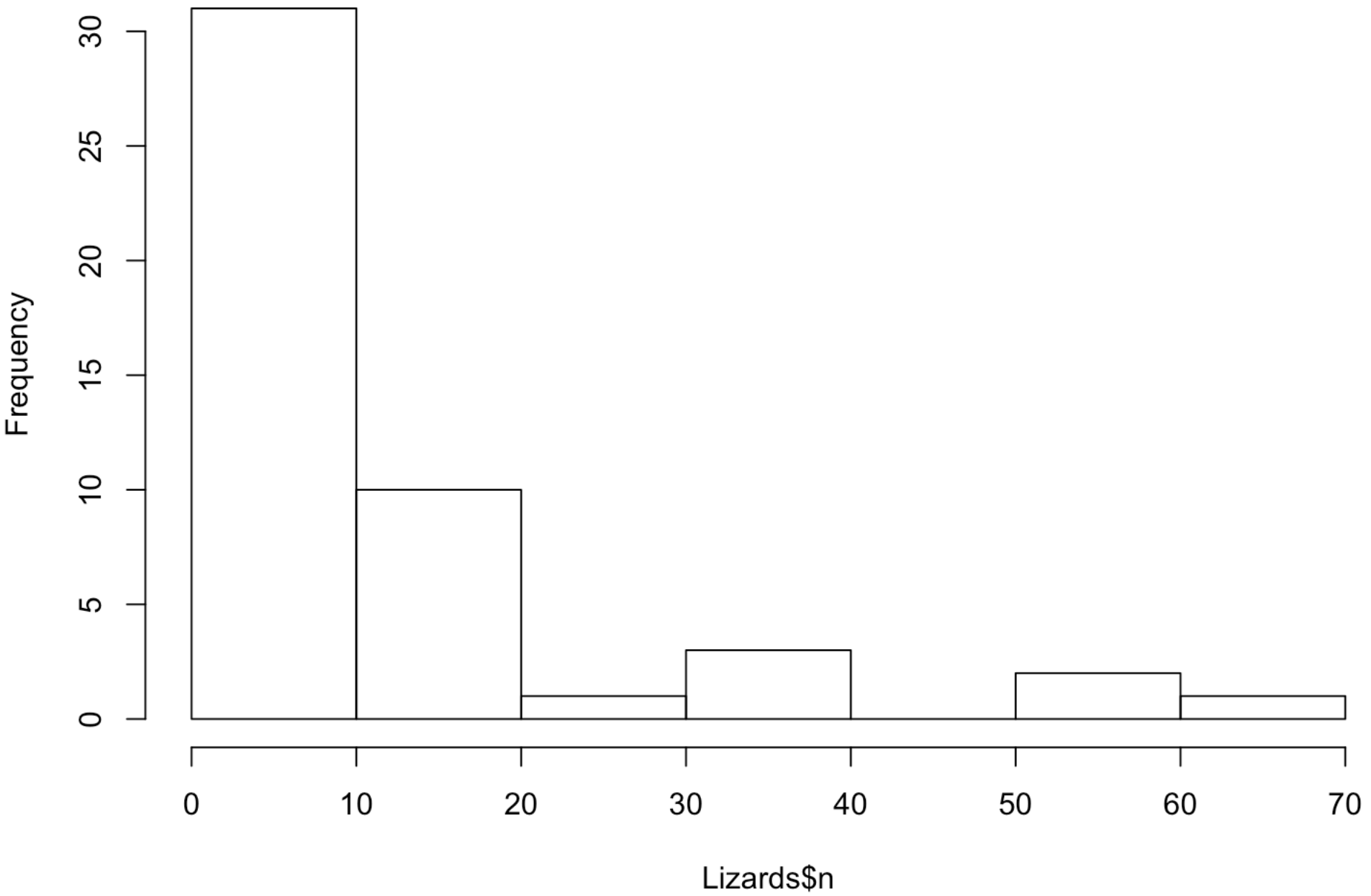
Let's look at some data first.

```
Lizards <- read.csv('lizards.txt', sep = "\t")
head(Lizards)
```

```
##    n   sun height  perch    time  species
## 1 20 Shade   High  Broad Morning opalinus
## 2 13 Shade    Low  Broad Morning opalinus
## 3  8 Shade   High Narrow Morning opalinus
## 4  6 Shade    Low Narrow Morning opalinus
## 5 34   Sun   High  Broad Morning opalinus
## 6 31   Sun    Low  Broad Morning opalinus
```

LO: Carry out a GLM

```
hist(Lizards$n)
```

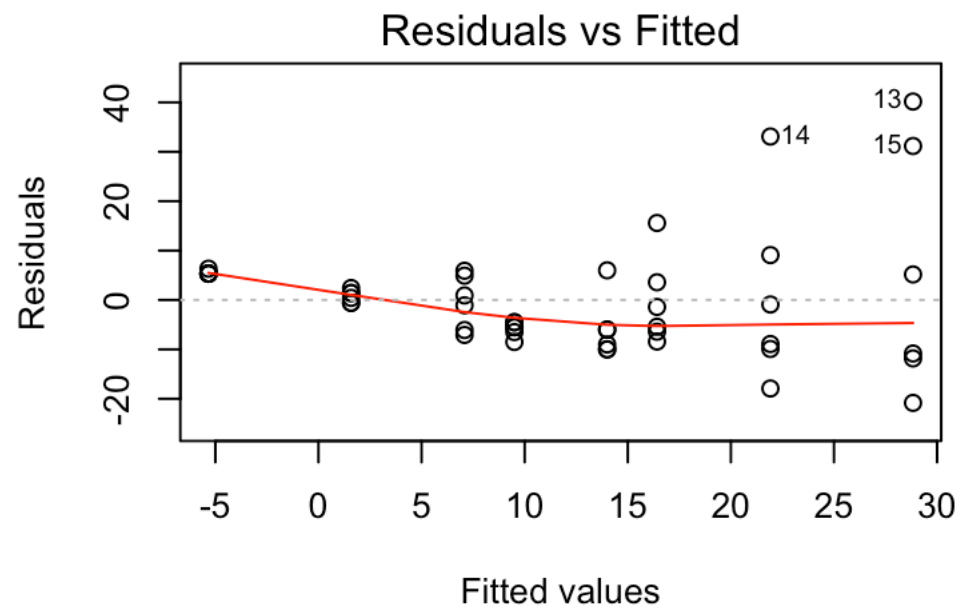


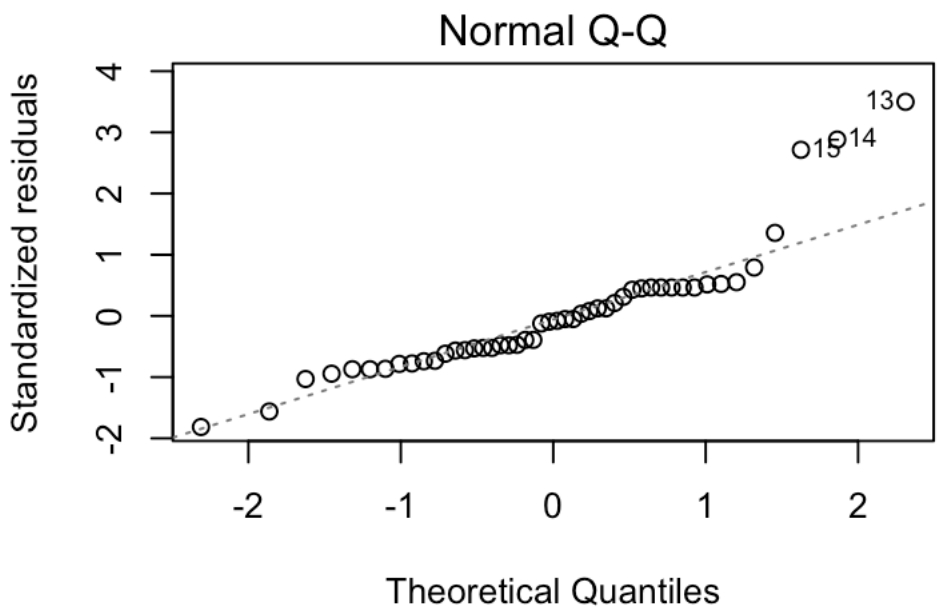
```r
LizardsModelLM <- lm(n ~ sun + height + species, data = Lizards)

LizardsModel <- glm(n ~ sun + height + species, family = poisson (link = log), data = Lizards)
```

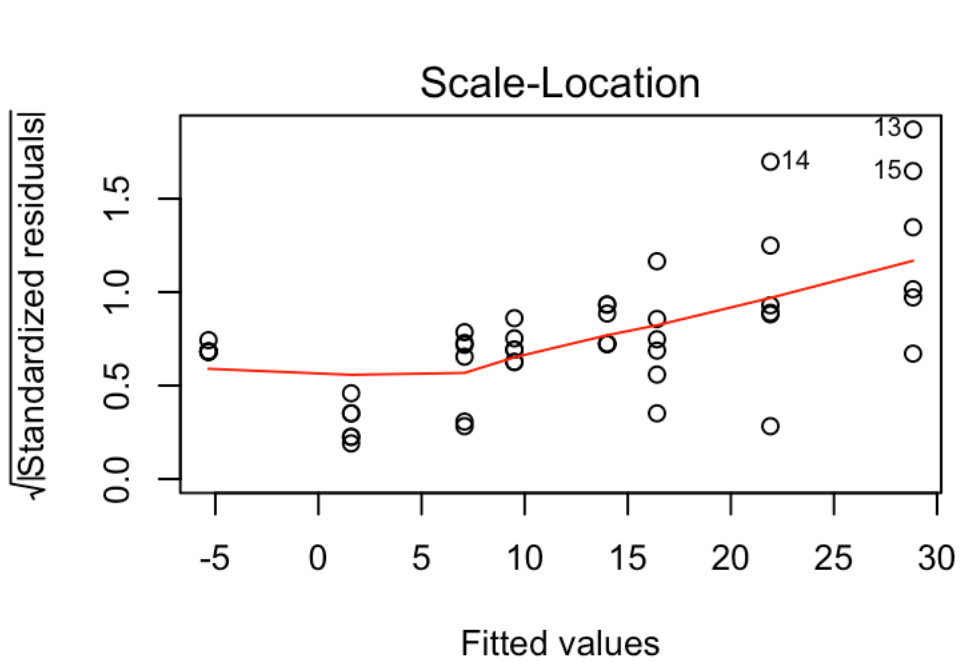
Check assumptions of the model

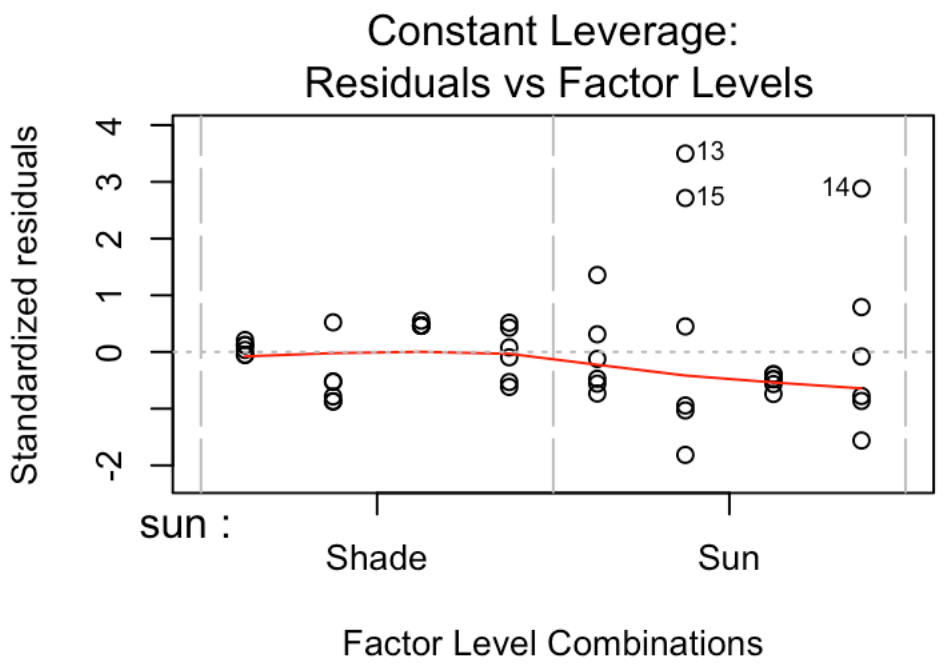
1. Distribution of the residuals

```
par(mfrow = c(2,2))
plot(LizardsModelLM)
```



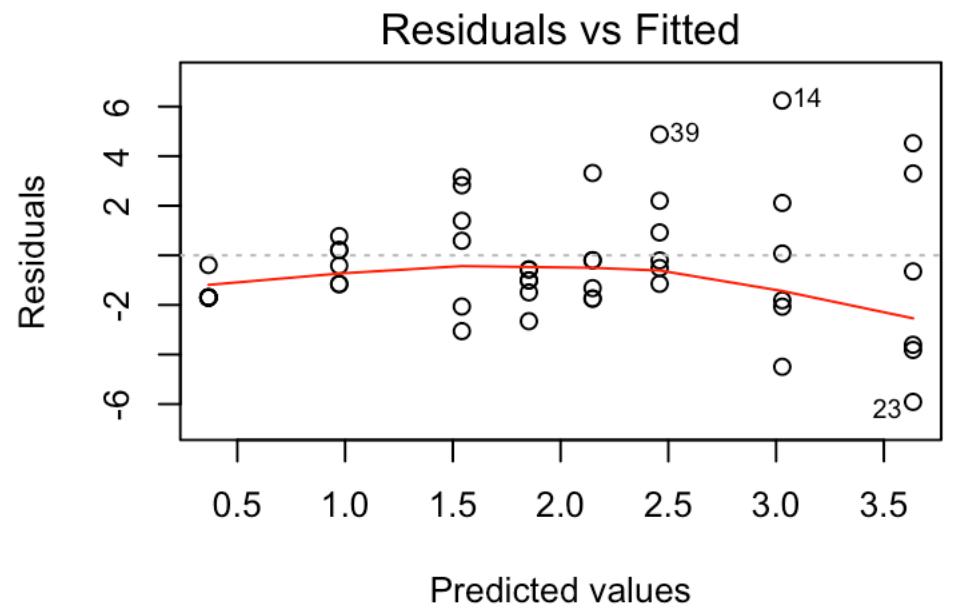


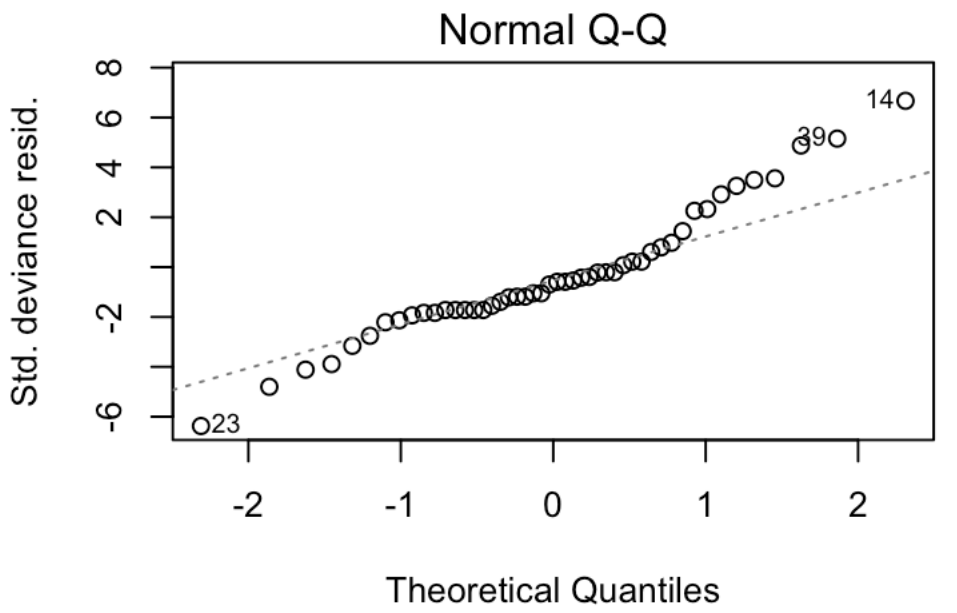


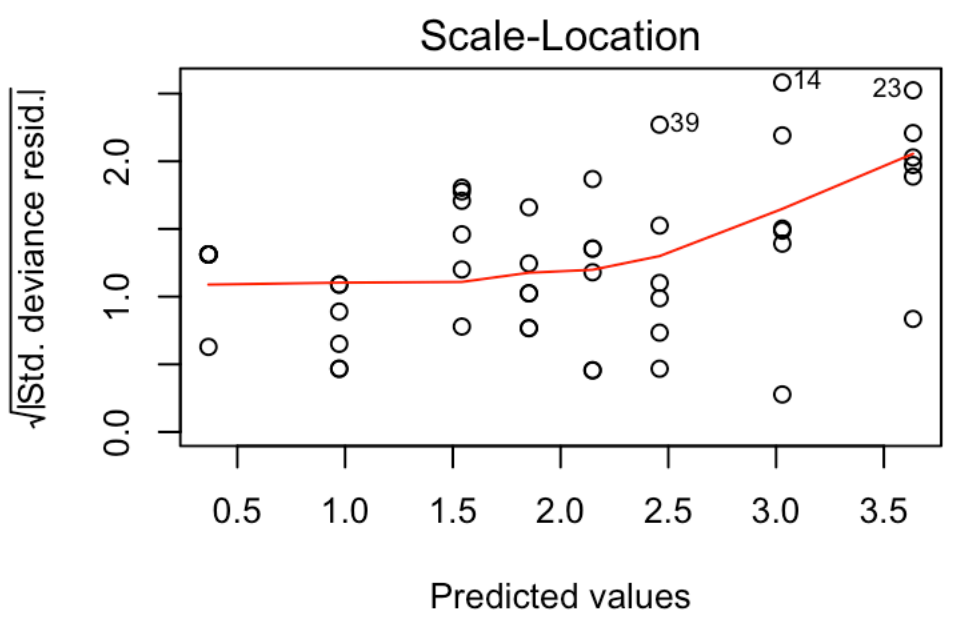


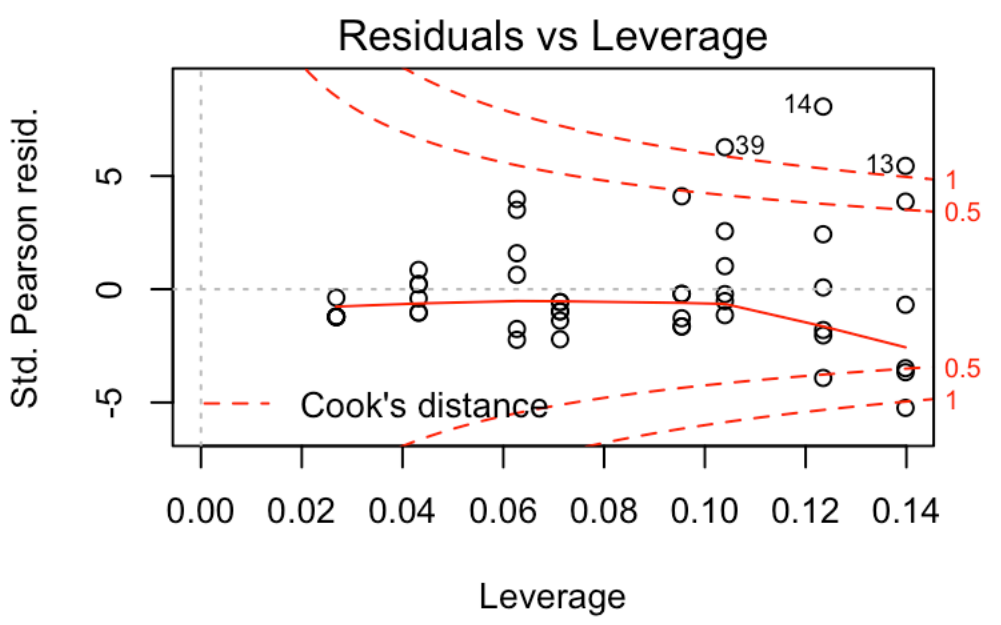
Let's see the GLM

```
par(mfrow = c(2,2))
plot(LizardsModel)
```









2. Overdispersion

In a poisson distribution, mean = variance

Overdispersion parameter θ = $\frac{\text{residual deviance}}{\text{residual degrees of freedom}}$

```
LizardsModel$deviance / LizardsModel$df.residual
```

```
## [1] 6.352926
```

If your model is overdispersed use the *quasi* family. In this case, quasipoisson

pseudo R^2 explained deviance

$$\text{Pseudo } R^2 = \frac{\text{null deviance} - \text{residual deviance}}{\text{null deviance}}$$

```
(LizardsModel$null.deviance - LizardsModel$deviance) / LizardsModel$null.deviance
```

```
## [1] 0.6210064
```

Model Summary

```
summary(LizardsModel)
```

```
##
## Call:
## glm(formula = n ~ sun + height + species, family = poisson(link = log),
##     data = Lizards)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -5.9107  -1.6984  -0.6075   0.6341   6.2438
##
## Coefficients:
##                Estimate Std. Error z value Pr(>|z|)
## (Intercept)     0.97277    0.12778   7.613 2.68e-14 ***
## sunSun          1.48684    0.10858  13.694  < 2e-16 ***
## heightLow      -0.60659    0.08812  -6.884 5.83e-12 ***
## speciesopalinus  1.17576    0.09919  11.853  < 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: 737.56  on 47  degrees of freedom
## Residual deviance: 279.53  on 44  degrees of freedom
## AIC: 450.78
##
## Number of Fisher Scoring iterations: 5
```

Challenge

Using the Boar csv, run the appropriate GLM.

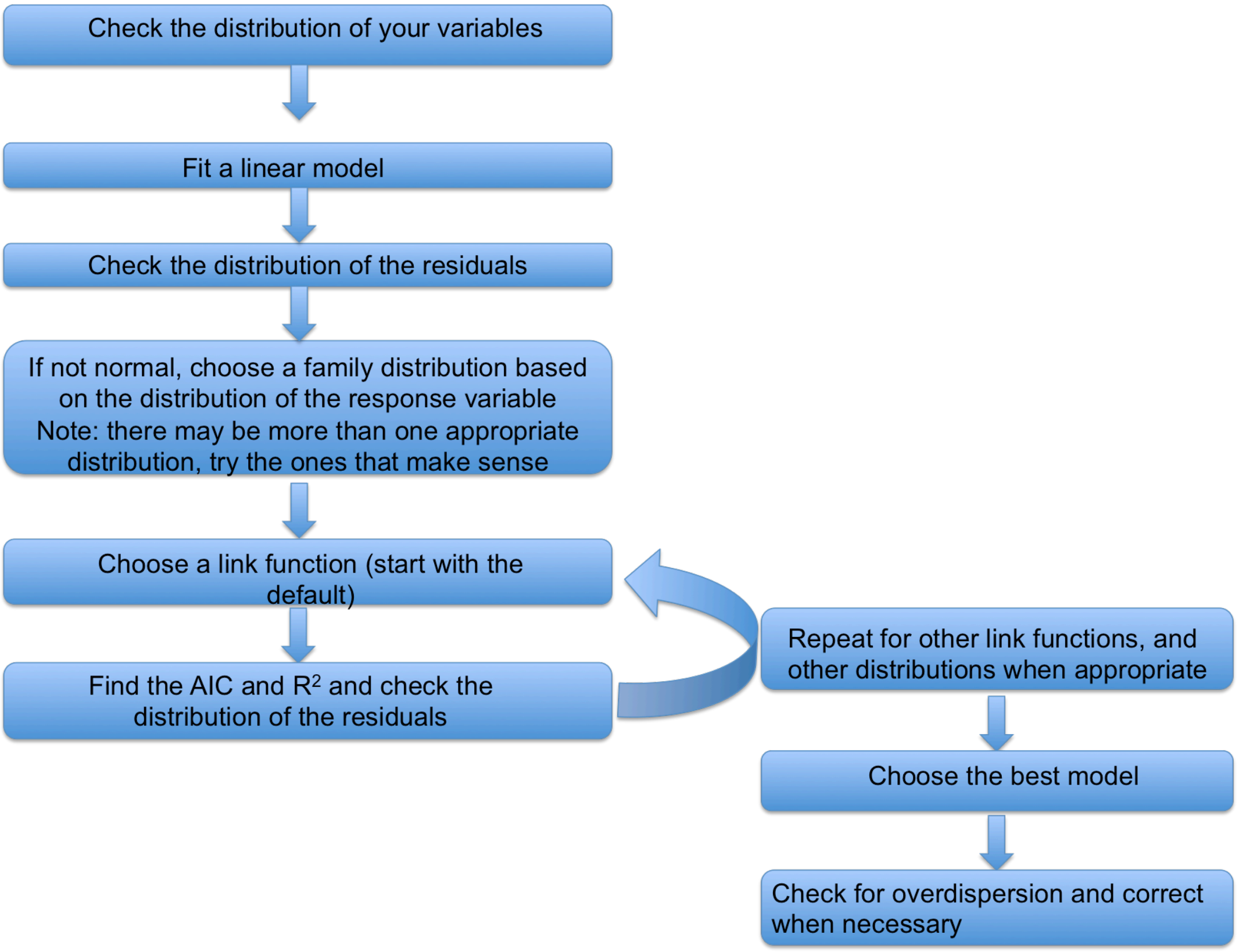
Make sure you check the model assumptions and all appropriate link functions

Learning Objectives

By the end of this lesson the students will:

- Differentiate and categorize GLMs vs ANOVAS and regressions
- Identify the different components of GLMs
- Select the most appropriate GLM
- Carry out a GLM

Summary


Check the distribution of your variables
Fit a linear model
Check the distribution of the residuals
If not normal, choose a family distribution based on the distribution of the response variable
Note: there may be more than one appropriate distribution, try the ones that make sense
Choose a link function (start with the default)
Find the AIC and R^2 and check the distribution of the residuals
Repeat for other link functions, and other distributions when appropriate
Choose the best model
Check for overdispersion and correct when necessary

Challenge 2 Solution

Look at the data first

```
Boar <- read.csv('Boar.csv')

head(Boar)
```

```
##    Tb sex age length
## 1   0   1   1   46.5
## 2   0   2   1   47.0
## 3   0   1   1   48.0
## 4   0   1   1   51.5
## 5   0   2   1   53.0
## 6   0   2   1   53.0
```

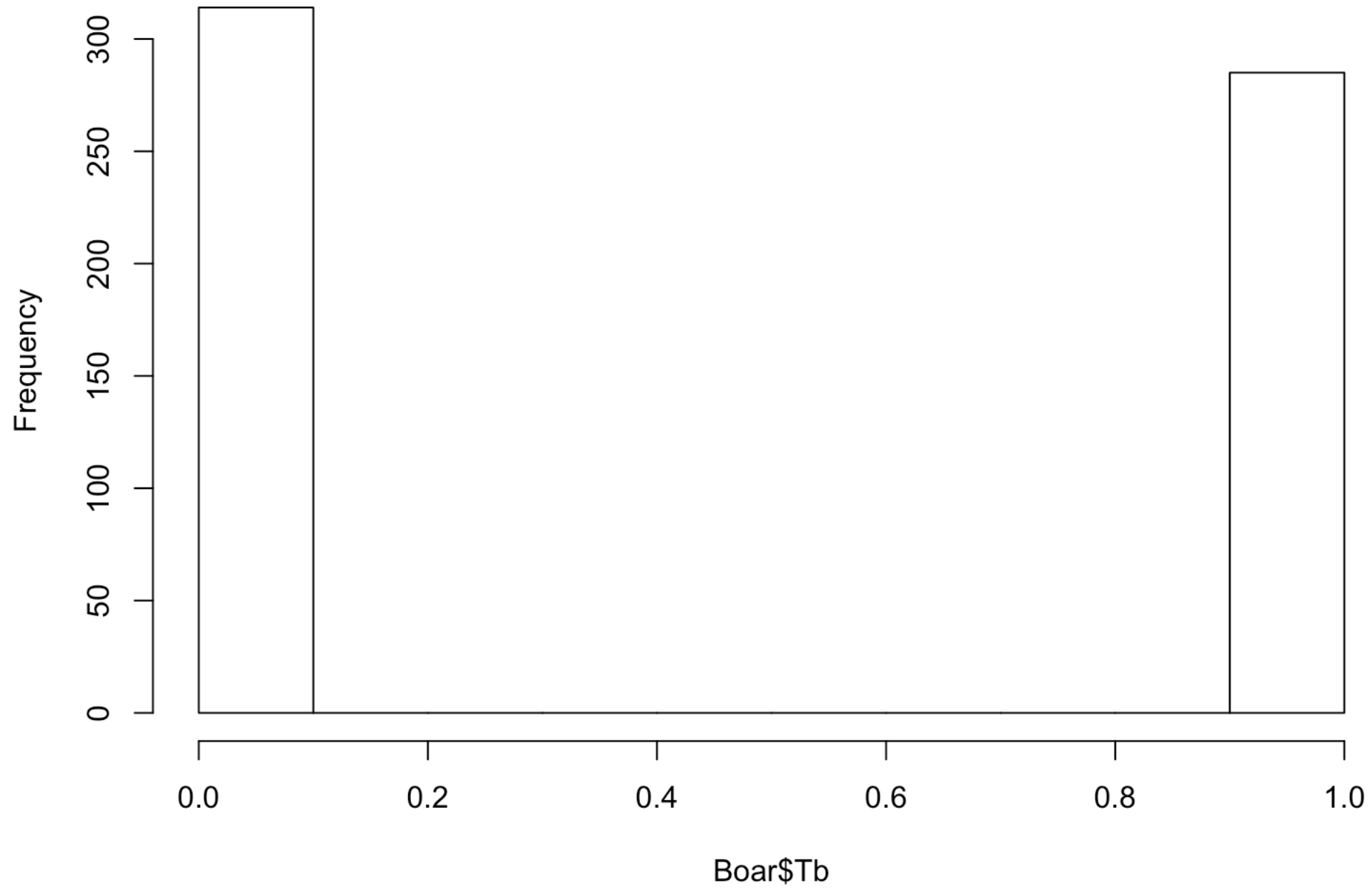
Tb is the response variable. Sex, age and length are explanatory variables.

```
summary(Boar)
```

```
##        Tb              sex              age            length     
##  Min.   :0.0000   Min.   :1.000   Min.   :1.000   Min.   : 46.5  
##  1st Qu.:0.0000   1st Qu.:1.000   1st Qu.:3.000   1st Qu.:106.0  
##  Median :0.0000   Median :2.000   Median :3.000   Median :121.0  
##  Mean   :0.4758   Mean   :1.582   Mean   :3.142   Mean   :116.8  
##  3rd Qu.:1.0000   3rd Qu.:2.000   3rd Qu.:4.000   3rd Qu.:129.5  
##  Max.   :1.0000   Max.   :2.000   Max.   :4.000   Max.   :165.0  
##  NA's   :58       NA's   :35      NA's   :7       NA's   :110
```

Let's check the distribion to see how it looks like.

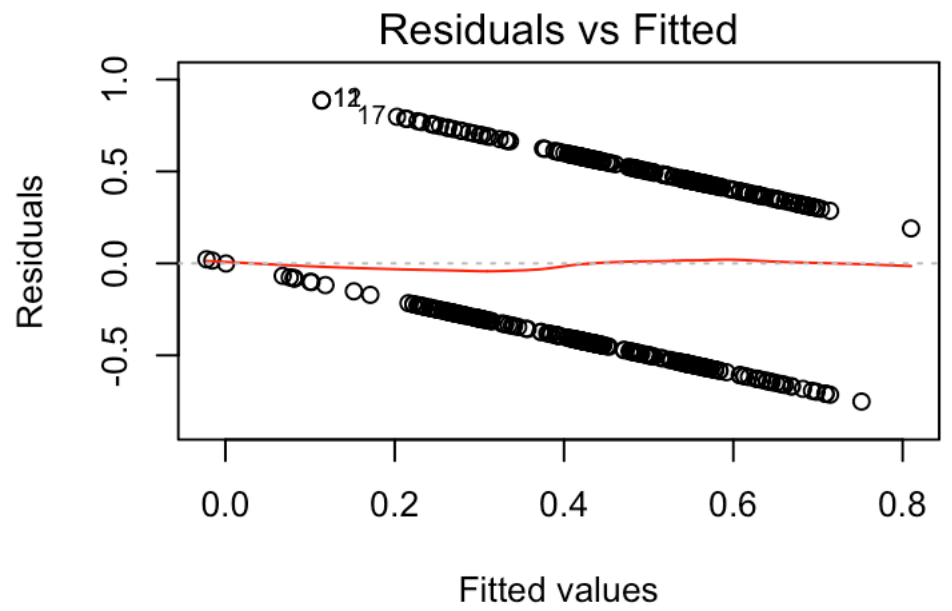
```
hist(Boar$Tb)
```

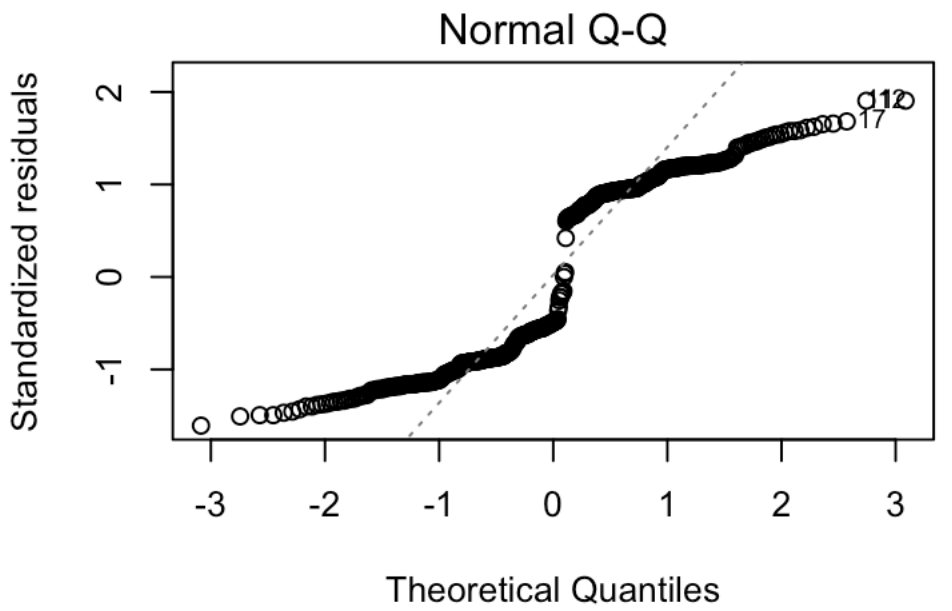


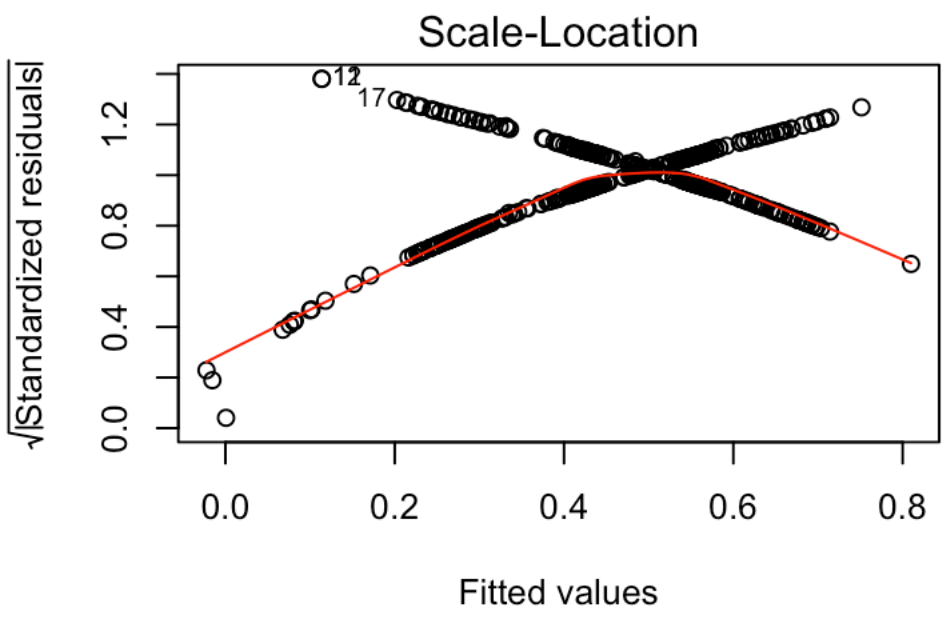
Looking at the summary and histogram, Tb is a categorical variables with two categories. Therefore we need to use the binomial distribution.

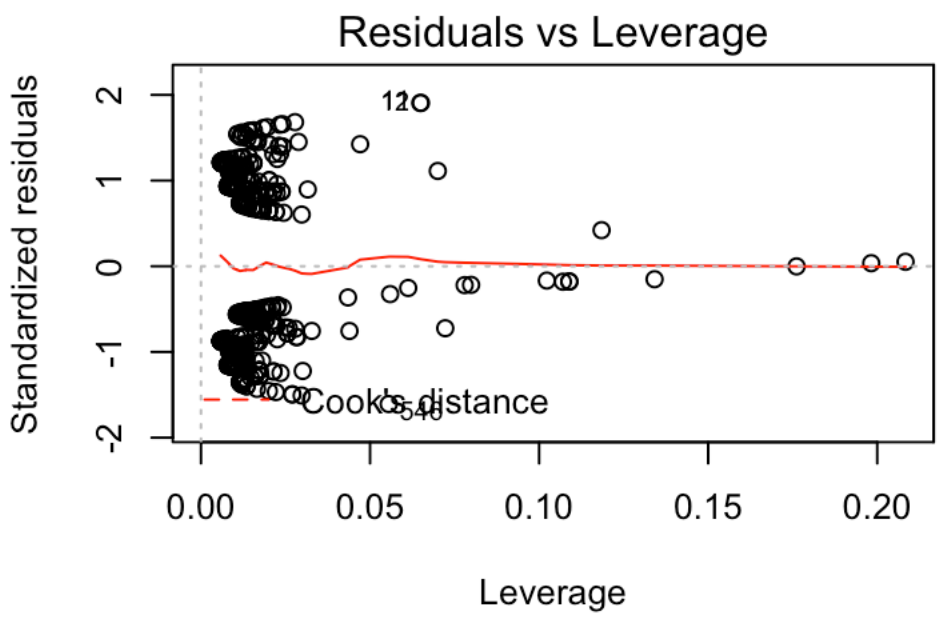
We can start with a linear model

```
LinearBoarModel <- lm(Tb ~ sex * age * length, data = Boar)
par(mfrow = c(2,2))
plot(LinearBoarModel)
```









The residuals don't look very normally distributed. Let's use a binomial family.

The default link functions for the binomial distribution are: logit, probit, cauchit, log, cloglog Let's start with the logit function

```
LogitBoarModel <- glm(Tb ~ sex * age * length, family = binomial (link = logit), data = Boar)
```

Let's look at the AIC and pseudo R^2

```
AIC(LogitBoarModel)
```

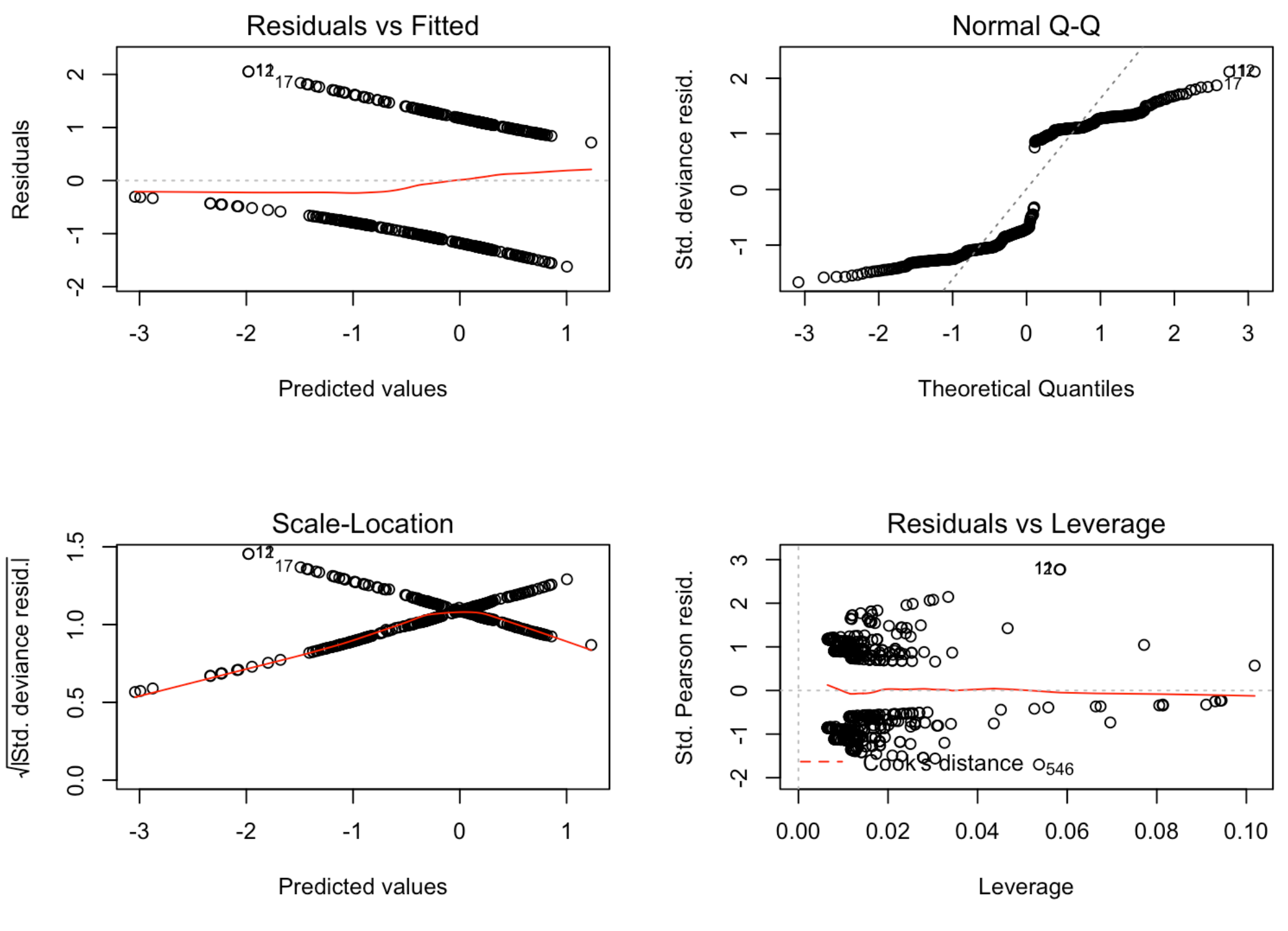
```
## [1] 653.3006
```

```
(LogitBoarModel$null.deviance - LogitBoarModel$deviance) / LogitBoarModel$null.deviance
```

```
## [1] 0.06451869
```

And the distribution of the residuals

```
par(mfrow = c(2,2))
plot(LogitBoarModel)
```



Look at all the other link functions and choose the one with the lowest AIC and better distribution of the residuals.

```
probitBoarModel <- glm(Tb ~ sex * age * length, family = binomial (link = probit), data = Boar)

cauchitBoarModel <- glm(Tb ~ sex * age * length, family = binomial (link = cauchit), data = Boar)

logBoarModel <- glm(Tb ~ sex * age * length, family = binomial (link = log), data = Boar)

cloglogBoarModel <- glm(Tb ~ sex * age * length, family = binomial (link = cloglog), data = Boar)

AIC(LogitBoarModel, probitBoarModel, cauchitBoarModel, logBoarModel, cloglogBoarModel)
```

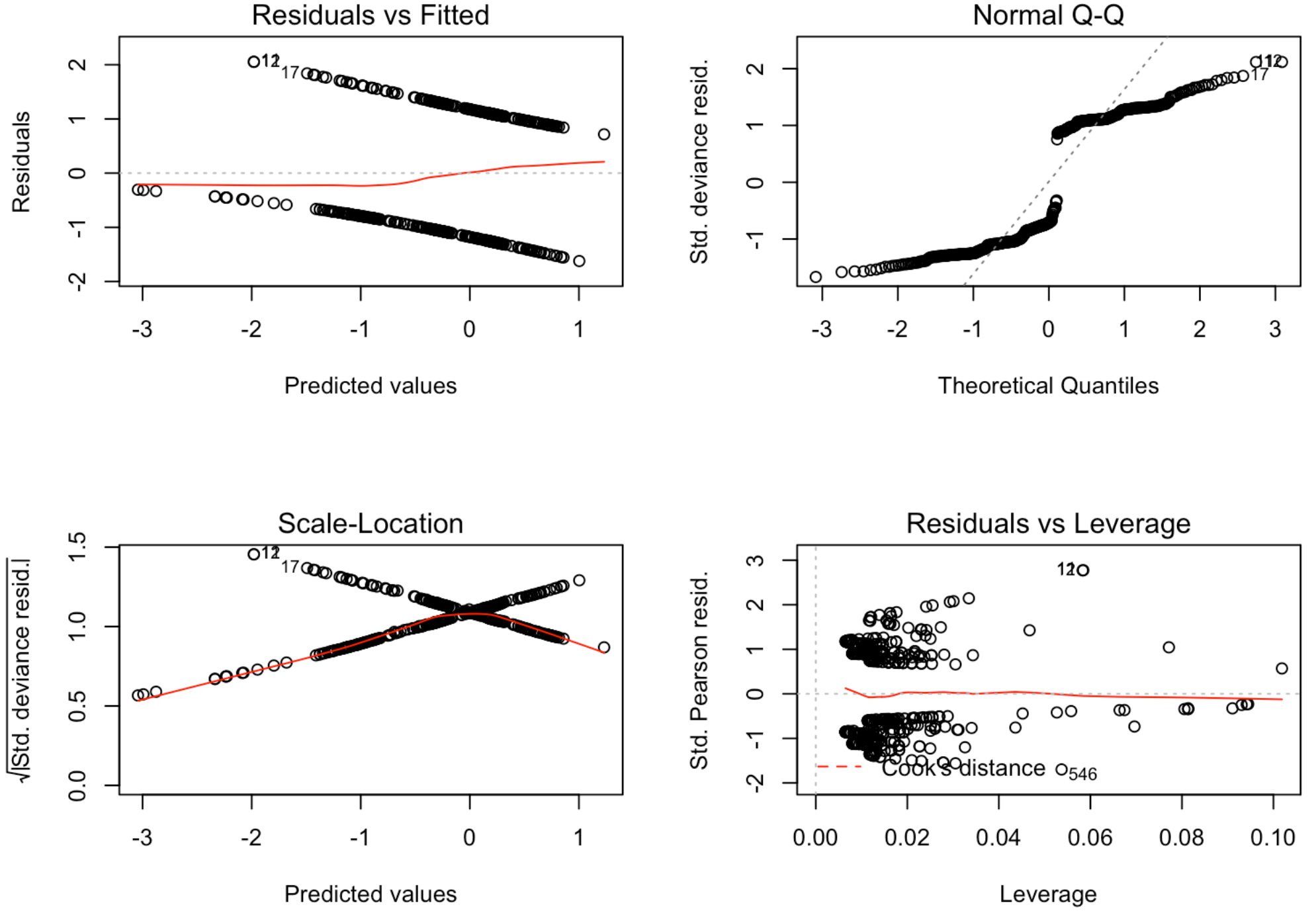
```
##                   df      AIC
## LogitBoarModel     8 653.3006
## probitBoarModel    8 653.3818
```

```
## cauchitBoarModel   8 652.2948
## logBoarModel       8 652.8498
## cloglogBoarModel   8 653.1425
```

Logit has the lowest AIC, so let's keep working with that

Let's check the distribution of the residuals.

```
par(mfrow = c(2,2))
plot(LogitBoarModel)
```



We also need to check for overdispersion

```
LogitBoarModel$deviance / LogitBoarModel$df.residual
```

```
## [1] 1.311318
```

θ is close to 1, so we can keep the binomial family.

We can keep refining the model using model simplification to choose a model we are happy with.