Generalized linear models with R

Introduction to R - Day 3

Instructor: Selina Baldauf

Freie Universität Berlin - Theoretical Ecology

2021-08-01 (updated: 2023-02-26)

Linear models with transformed response variable

When to transform?

Transformation of the response variable Y can help with **potential violations of the model assumptions**:

- residuals non-normally distributed
  - skewed distribution
  - outliers
- trends in the residuals
- non-constant variance

Example

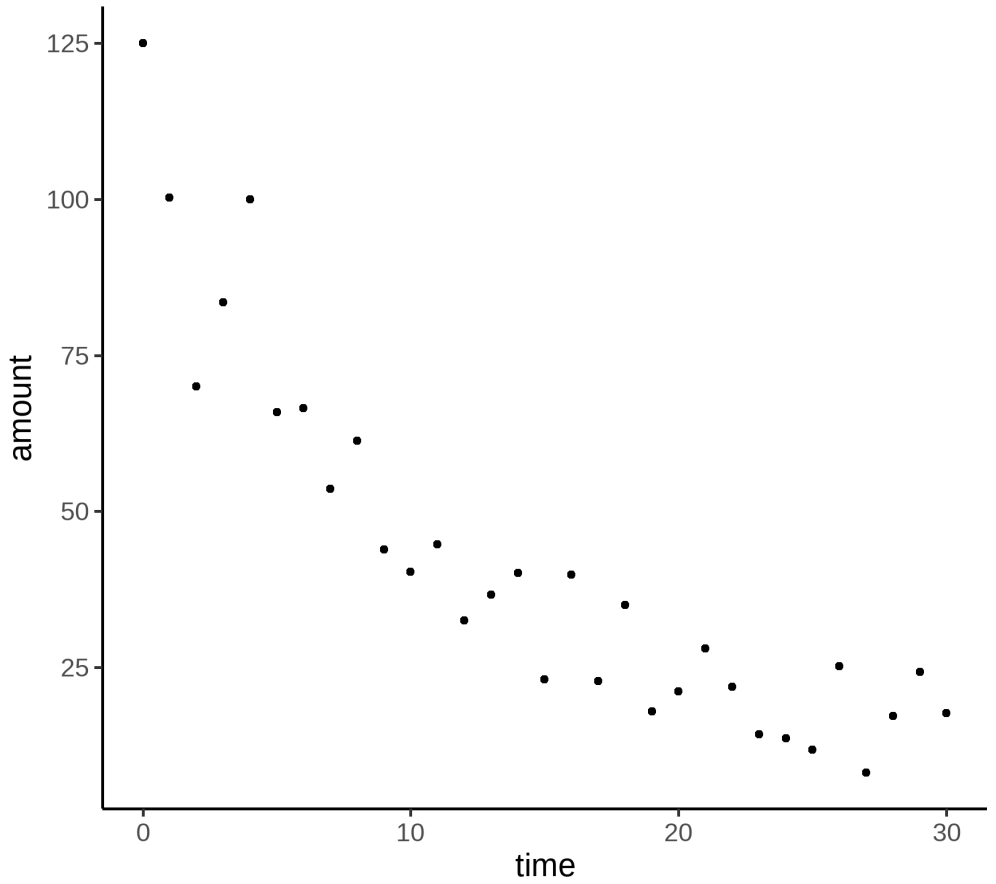
- Non-linear function $y = a * e^{b*x}$
- Transformed to a linear model: $log(y) = log(a) + b * x$
- Common natural processes: exponential growth and decay

Example

Data set `decay` on the decay of soil organic matter over time.

Two variables:

- `amount` of organic matter
- `time`



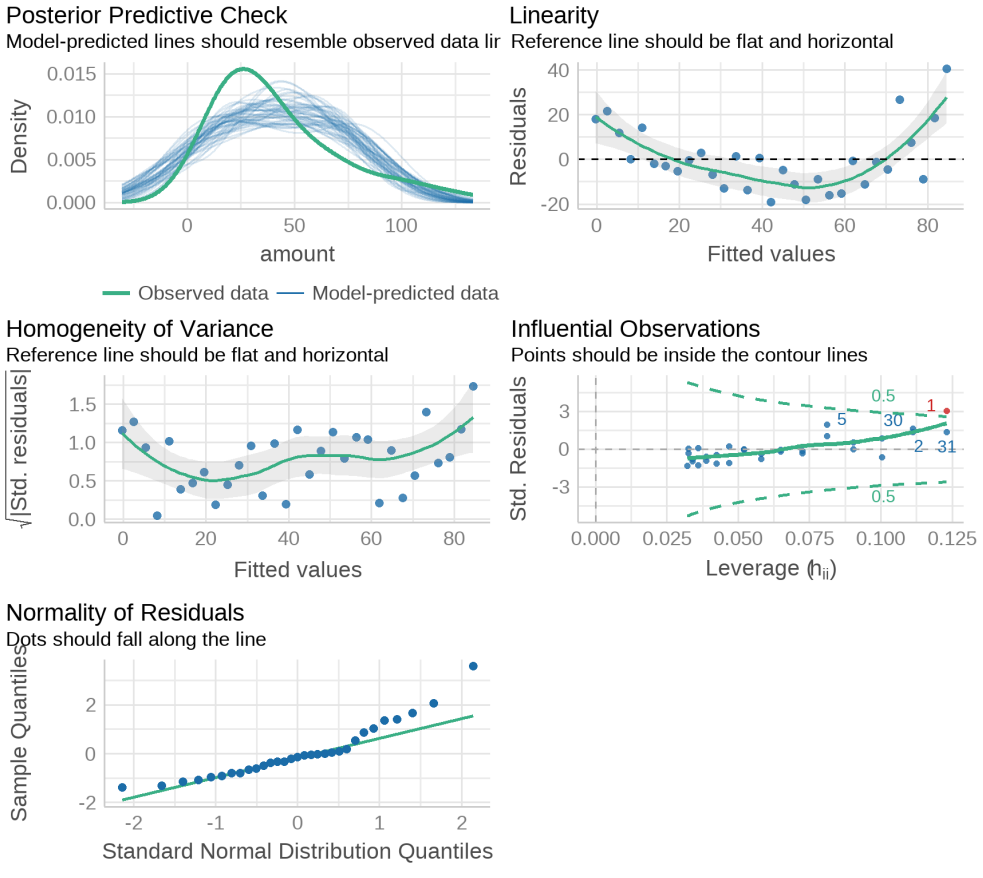
Example

First, let's fit a linear model to the untransformed data and look at the diagnostic plots:

```
mod1 <- lm(amount ~ time, data = decay)
performance::check_model(mod1)
```

Diagnostic plots look bad:

- pattern in the residuals
- skewed distribution



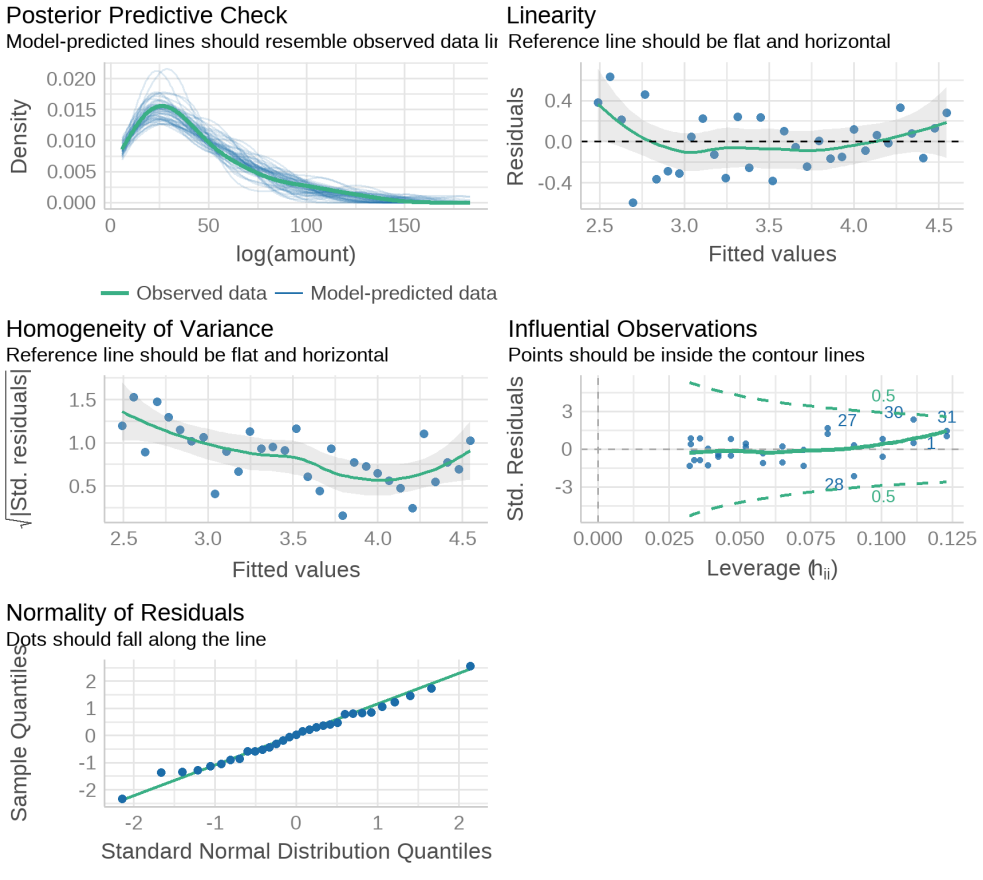
Exercise

Let's refit the model to transformed data (log-transformed response variable):

```
mod2 <- lm(log(amount) ~ time, data = decay)
performance::check_model(mod2)
```

Diagnostic plots look much better (though not perfect)

- no patterns anymore



Test for significant effects

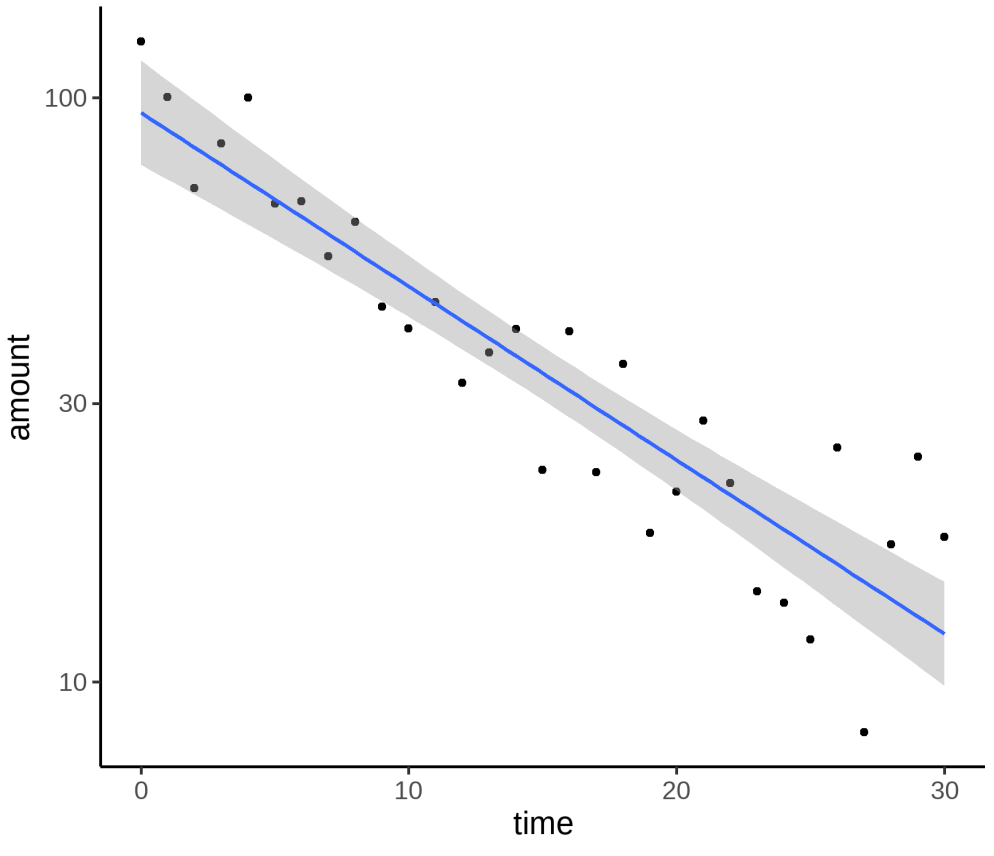
Is the effect of time on the amount of organic matter significant?

```
drop1(mod2, test = "F")
## Single term deletions
##
## Model:
## log(amount) ~ time
##        Df Sum of Sq    RSS     AIC F value    Pr(>F)
## <none>               2.372 -75.678
## time    1    11.646 14.018 -22.602  142.39 1.038e-12 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Plot the model

To plot the results of the model **on the transformed scale**, we can just use `geom_smooth` in combination with `scale_y_log10`

```
ggplot(decay, aes(x = time, y = amount)) +
  geom_point() +
  scale_y_log10() +
  geom_smooth(method = "lm")
```

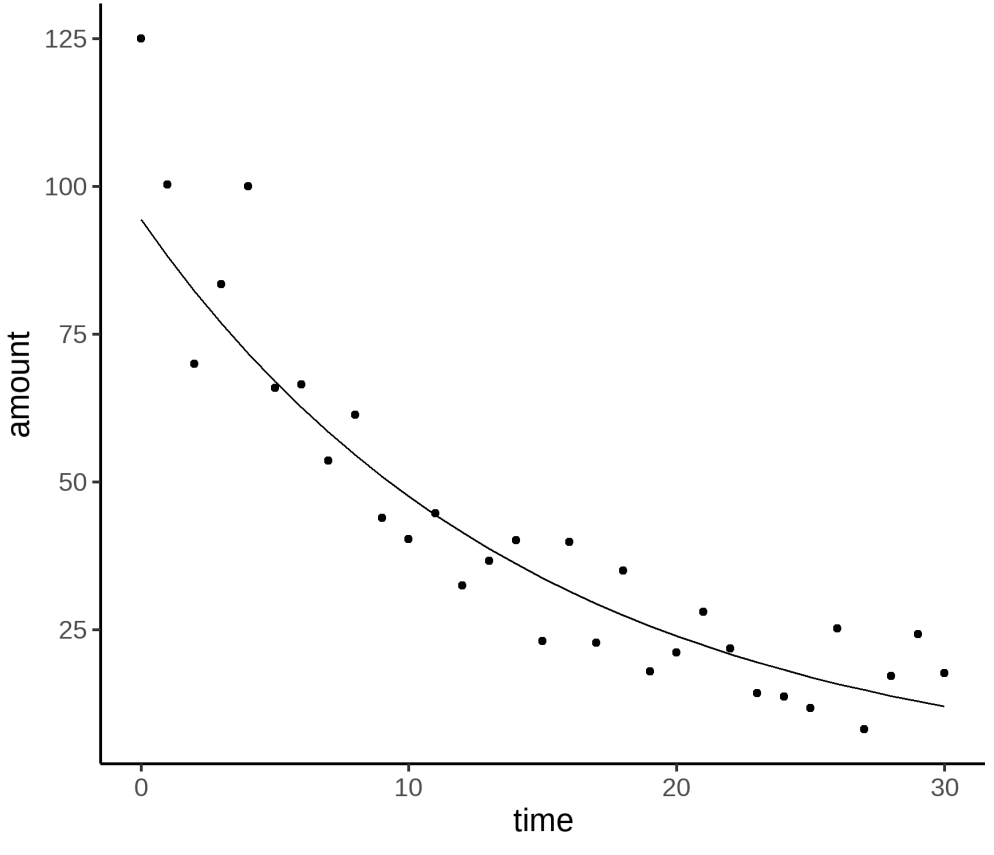


Plot the model

However, we mostly want to plot the model **on the scale of the original data**. For this, we need to use `predict`

```
# step 1: create some data to predict from
pred_data <- tibble(time = 0:30)
# step 2: make the prediction
amount_pred <- predict(mod, newdata =
pred_data)
# step 3: backtransform to original scale and
add to prediction data
pred_data$amount <- exp(amount_pred)
# step 4: add model to plot
ggplot(decay, aes(x= time, y=amount))+
  geom_point()+
  geom_line(data = pred_data)
```

- The trick here is to **backtransform** the predictions using `exp` as the inverse function of `log`



Transformations step by step

1. Plot raw data
2. Fit a model to untransformed data
3. Check diagnostic plots
4. Apply a transformation
5. Re-fit model to transformed data
6. Check diagnostic plots again
7. Plot raw data and add model predictions based on the transformed data
    - Backtransform predictions appropriately:
    - $log(y) <=> exp(y)$
    - $sqrt(y) <=> y^2$

Now you

Task 3: Linear models with transformations (75 min)

Find the task description here

GLMs background

Assumptions of linear models:

- Normal distribution of residuals
    - errors and response not bounded
- Constant variance of residuals
    - no relationship between mean and variance of y

There are types of data that (generally) violate these assumptions.

GLMs background

Count data

- are bounded (cannot be < 0)
- variance increases with the mean
  → response described by a ***poisson distribution***

Proportion data

- percentages: bounded between 0 and 1
- variance highest at p = 0.5, lower towards p = 0 and 1
  → response described by a ***binomial distribution*** (# successes in N trials with probability p)

Binary data

- Only 0 and 1
- variance highest at p = 0.5, lower towards p = 0 and 1
  → response described by a ***bernoulli distribution*** (binomial distribution with N=1)

GLMs background

Components of a GLM

- Linear predictor $\eta_i = \beta_0 + \sum \beta_j * X_{i,j}$
  - like a linear model: linear combination of the effects of the predictors X_j
- Error structure
  - non-normal distribution e.g. skewed, bounded, non-negative (e.g. Poisson distribution)
- Link function $g(\mu_i) = \eta_i$
  - relates the expected response values μ_i to the linear predictor
  - A linear model is also a glm with a gaussian link function (identity `=`)
  - predicted values: require inverse link function!

Parameter estimation in GLMS

Linear models

- Ordinary Least Squares (OLS)
- Minimizing the residual sum of squares to find parameters
- This method only works if linera model assumptions are not violated

GLMs

- Maximizing the likelihood (ML)
- **Likelihood:** Probability of the data *given* the model
- **Likelihood function:** How does the likelihood depend on the model parameters? → Maximum: ML parameter estimates of the GLM

Binomial distribution

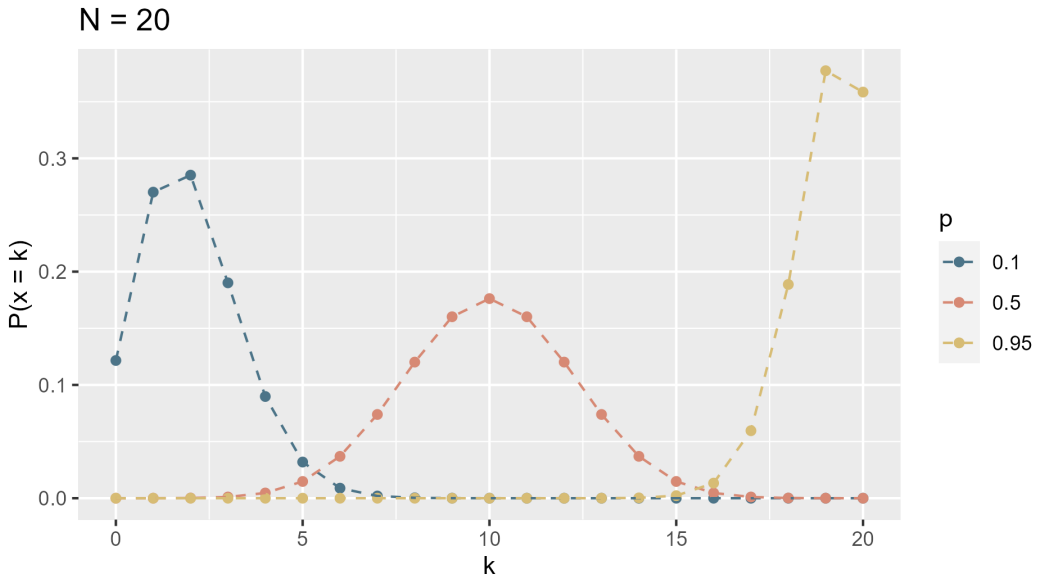
- Frequency distribution of the number of successes of an event with success probability p and N trials.
  - E.g. No of "six" with 10 dice: $p = 0.5, N = 10$

$$f(k) = \binom{N}{k} p^k * (1-p)^{N-k}$$

$$E(k) = N * p$$

$$Var(k) = N * p * (1-p)$$

with k being the number of "successes"



Bernoulli distribution as special case with N = 1 (e.g. coin flip).

Binomial and binary GLMs

- logit link function: $logit(p_i) = ln(\frac{p_i}{1-p_i}) = \eta_i$
- inverse link function: $p_i = \frac{e^{\eta_i}}{1+e^{\eta_i}}$
- binomial error distribution: $\epsilon \sim B(p_i, N = n)$

Poisson distribution

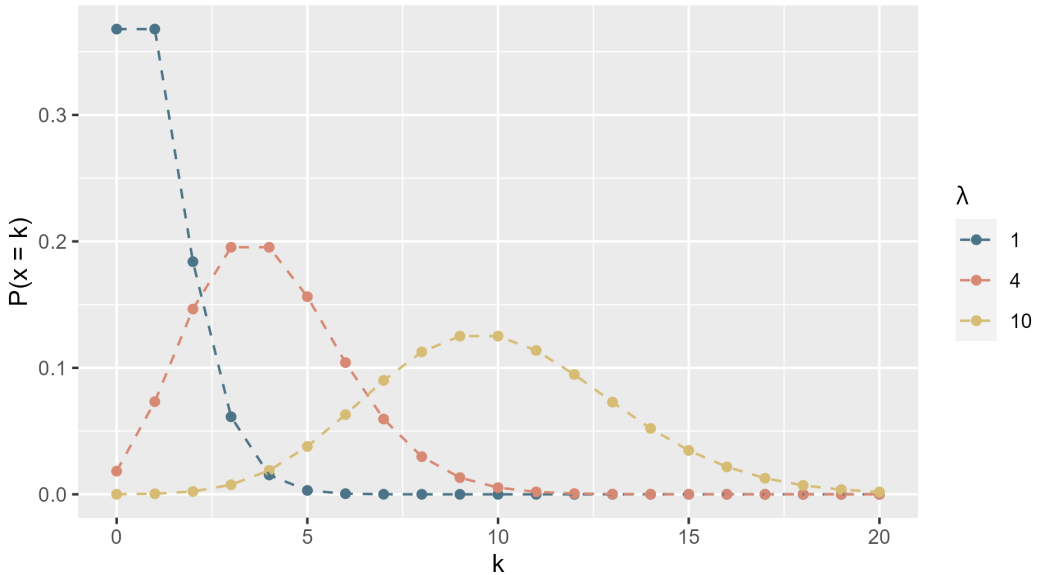
- Frequency distribution of counts of rare events
  - E.g. Bird count in forests

$$f(k) = \frac{e^{-\lambda}\lambda^k}{k!}$$

$$E(k) = \lambda$$

$$Var(k) = \lambda$$

with k being the number of occurrences



Poisson GLMs

- log link function: $ln(\lambda_i) = \eta_i$
- inverse link function: $\lambda_i = e^{\eta_i}$
- poisson error distribution: $\epsilon \sim P(\lambda_i)$

GLMs in R

Let's look a poisson GLM as example

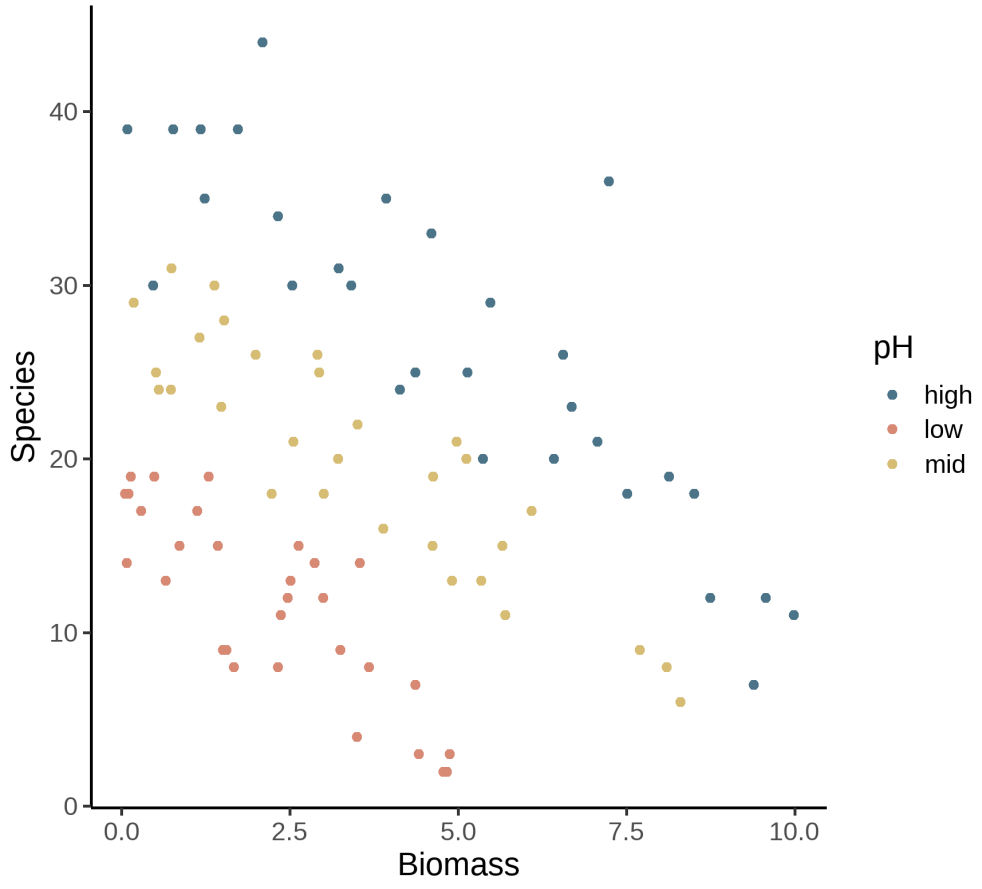
The data

Data set: Species diversity in plots with different soil pH

- Response variable:
  - number of plant species
- Predictor variables:
  - Plot biomass
  - soil pH

```
specdat
```

```
## # A tibble: 90 × 3
##    pH    Biomass Species
##    <chr>   <dbl>   <dbl>
## 1 high    0.469      30
## 2 high    1.73       39
## 3 high    2.09       44
## 4 high    3.93       35
## # … with 86 more rows
```



Fit the model

Question: How do biomass and soil pH influence species richness?

Fit the most complex model with interaction to describe our hypothesis:

```
mod1 <- glm(Species ~ Biomass + pH + Biomass:pH,
            data = specdat,
            family = "poisson")
```

- same structure as `lm`
- add family argument to specify the error distribution
  - by default, R will pick the corresponding link function for the error distribution.
  - you could also be explicit about link function with `family = poisson(link = "log")`
  - for more information have a look at `?family`

Summary table

```
summary(mod1)
##
## Call:
## glm(formula = Species ~ Biomass + pH + Biomass:pH, family = "poisson",
##     data = specdat)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.4978  -0.7485  -0.0402   0.5575   3.2297
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept)    3.76812    0.06153  61.240  < 2e-16 ***
## Biomass       -0.10713    0.01249  -8.577  < 2e-16 ***
## pHlow         -0.81557    0.10284  -7.931 2.18e-15 ***
## pHmid         -0.33146    0.09217  -3.596 0.000323 ***
## Biomass:pHlow -0.15503    0.04003  -3.873 0.000108 ***
## Biomass:pHmid -0.03189    0.02308  -1.382 0.166954
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 452.346  on 89  degrees of freedom
## Residual deviance:  83.201  on 84  degrees of freedom
## AIC: 514.39
##
```

Summary table

- Coefficients are on scale of link function → cannot be directly interpreted on the scale of the data
- For glms: deviance as measure of goodness-of-fit (for lms it was sum of squared residuals)

Hypothesis tests in GLMs

- for linear models: F-tests
  - $F = \frac{\frac{SSR}{df(mod1)}}{\frac{SSR}{dfmod2}}$
  - in R: `drop1(mod1, test = "F")`
- for generalized linear models: likelihood-ratio (LR) test
  - $LR = deviance(mod1) - deviance(mod2)$
  - LR follows a χ^2 distribution
  - in R: `drop1(mod1, test = "Chisq")`

Hypothesis tests in GLMs

Test the significant of the predictors in our model with

```
drop1(mod1, test = "Chisq")
## Single term deletions
##
## Model:
## Species ~ Biomass + pH + Biomass:pH
##            Df Deviance    AIC   LRT  Pr(>Chi)
## <none>          83.201 514.39
## Biomass:pH  2   99.242 526.43 16.04 0.0003288 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

If we remove the interaction between biomass and pH from the model, the deviance increases significantly

→ There is a significant effect or biomass and pH on species richness.

Plot the model

Option 1: Use predict function

```
# step 1: creat some data to predict from
pred_dat <- expand_grid(
  Biomass = seq(
    from = min(specdat$Biomass),
    to = max(specdat$Biomass),
    length.out = 200),
  pH = unique(specdat$pH)
)
```

```
# step 2: predict species with the new data
# careful: add type = response
pred_dat$Species <- predict(mod1, newdata = pred_dat,
                            type = "response")
```

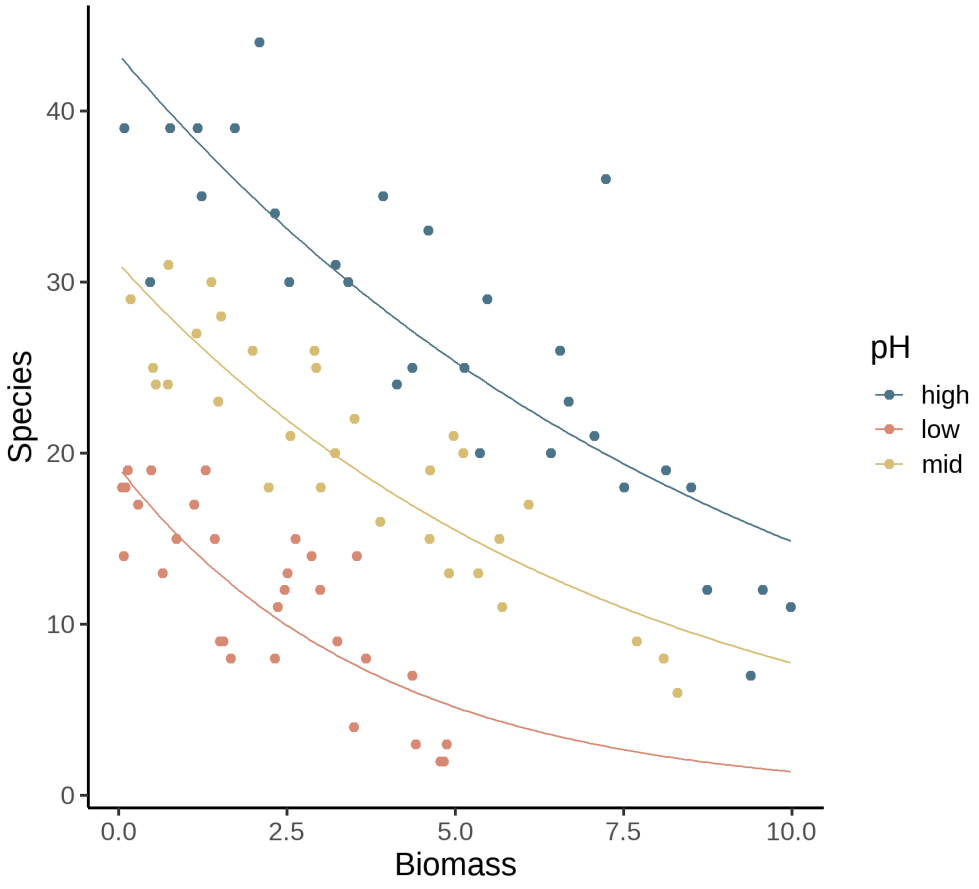
With `type = response` we predict **on the scale of the response variable** (i.e. the data) and not on the scale of the link function

Plot the model

Option 1: Use predict function

```
# define some colors
cols <- c( "#4C7488","#D78974", "#D7BC74")

# step 3: Add model predictions to the plot
ggplot(specdat, aes(
  y = Species,
  x = Biomass,
  color = pH
)) +
  geom_point() +
  geom_line(data = pred_dat) +
  scale_color_manual(values = cols)
```

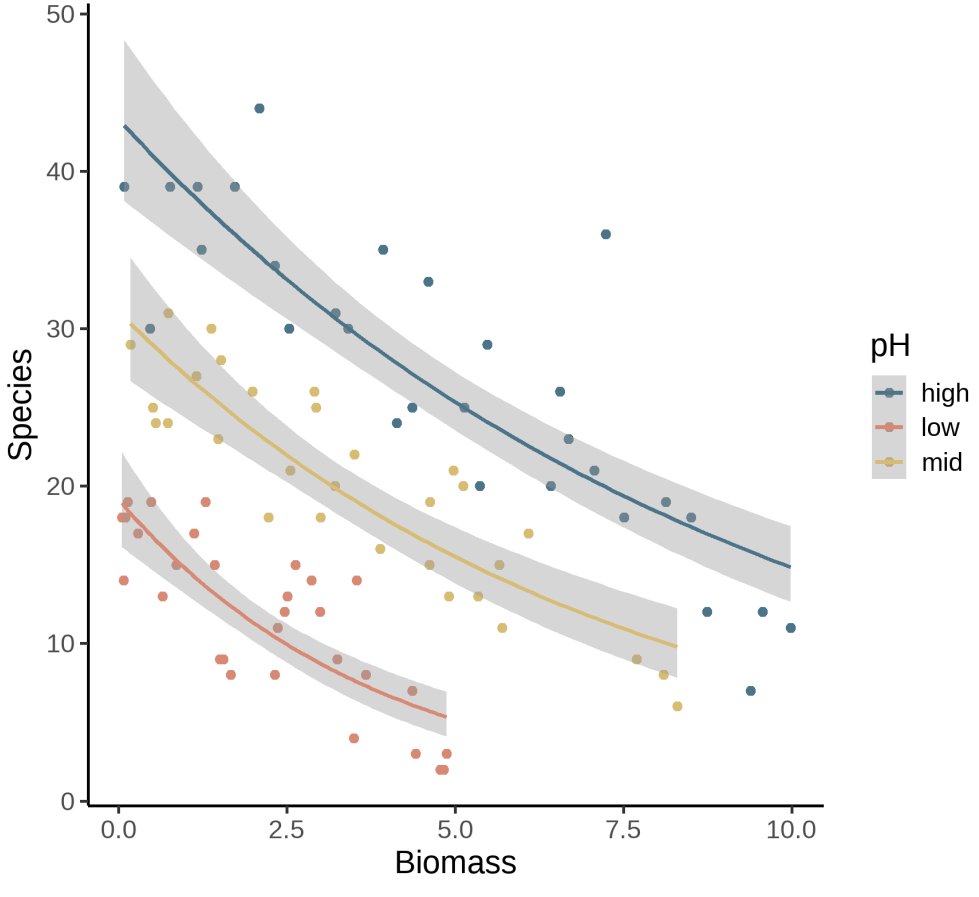


Plot the results

Option 2: Use `geom_smooth`

```
ggplot(specdat, aes(y = Species, x = Biomass,
color = pH)) +
  geom_point() +
  geom_smooth(
    method = "glm",
    method.args = list(
      family = "poisson"
    )
  )
```

- careful: `geom_smooth` always uses most complex model with all interactions
  - in this case: we want exactly that



GLM with binary data

Example

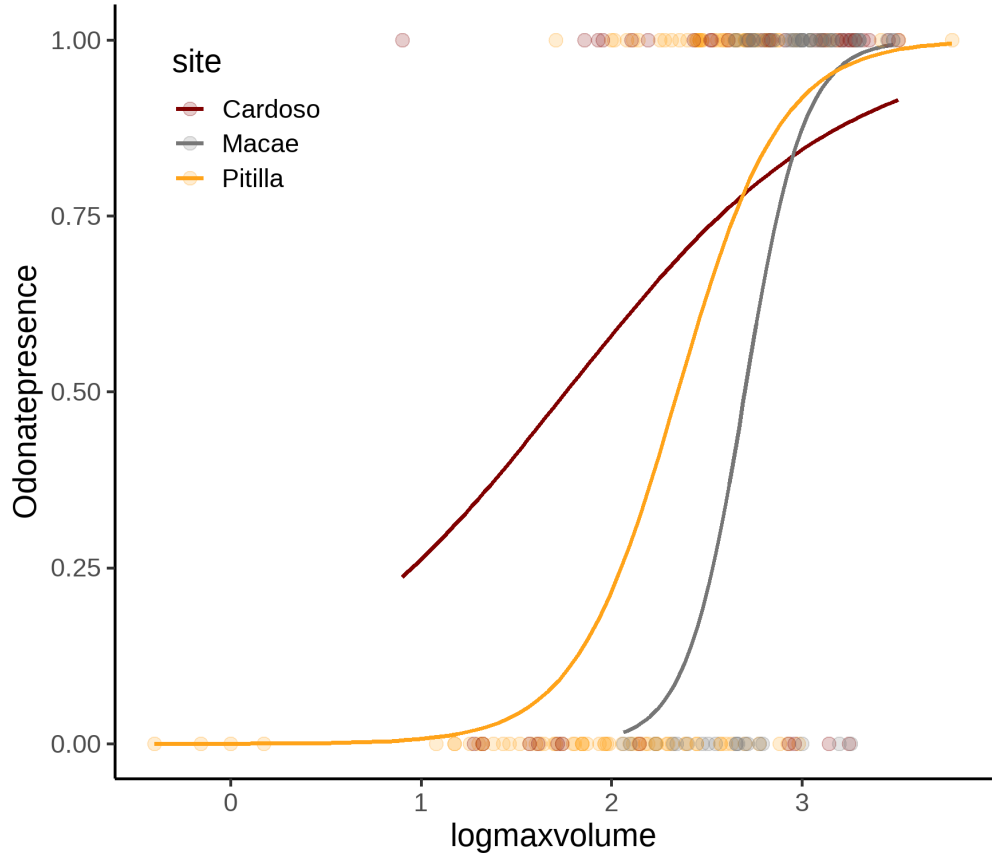
Odonates in water filled bromeliads in Costa Rica.

Variables:

- `site`: location
- `logmaxvolume`: bromeliad size
- `Odonatepresence`: Odonates present (1) or not (0)

Question: How does bromeliad size influence the presence of Odonates?

```
mod <- glm(Odonatepresence ~ logmaxvolume +
site, data = bromeliad, family = "binomial")
```



Petermann et al., *Ecology* (2015)

Now you

Task 3: Poisson and binary glms with R

Find the task description href="../03_tasks_controller.html#poisson-and-binary-glm">here

GLMs with proportion data

Let's look at an example

The data

Winter mortality of black-tailed prairie dogs

Question:

How do

- successful mating
- achievement of a certain minimum weight before winter
- hibernation

predict the *probability of survival & death* in winter?

By I, Brocken Inaglory, CC BY-SA 3.0, Link

The data

Winter mortality of black-tailed prairie dogs

```
pdogs
```

```
## # A tibble: 8 × 5
##   mated hibernation min.weight n.tot n.death
##   <chr> <chr>       <chr>      <dbl>   <dbl>
## 1 No    Yes         Yes           60       5
## 2 Yes   Yes         Yes           17       2
## 3 No    No          Yes            8       1
## # … with 5 more rows
```

- 3 categorical predictors: `mated`, `hibernation`, `min.weight`
- total no. of animals (`n.tot`)
- number of dead animals (`n.death`)

→ What is the response variable here?

The response variable

Winter mortality of black-tailed prairie dogs

Response variable: **proportion** of surviving (or dead) animals

→ Response has to be calculated first:

```
pdogs <- pdogs %>%
  mutate(
    n.surv = n.tot - n.death,
    p.surv = n.surv / n.tot,
    p.dead = 1 - p.surv
  )
```

By I, Brocken Inaglory, CC BY-SA 3.0, Link

The response variable

The response variables (`p.dead` or `p.surv` depending on the question) are now added to the data

```
## # A tibble: 8 × 8
##   mated hibernation min.weight n.tot n.death n.surv p.surv p.dead
##   <chr> <chr>       <chr>      <dbl>   <dbl>  <dbl>  <dbl>  <dbl>
## 1 No    Yes         Yes           60       5     55  0.917 0.0833
## 2 Yes   Yes         Yes           17       2     15  0.882 0.118
## 3 No    No          Yes            8       1      7  0.875 0.125
## 4 Yes   No          Yes            2       0      2  1     0
## 5 No    Yes         No           187      35    152  0.813 0.187
## # … with 3 more rows
```

But what could be the problem with that?

→ loss of information in proportions!

- 2 survivors / 4 total = 50%
- 100 survivors / 200 total = 50% **But** 100 out of 200: more information than 2 out of 4

→ data points with higher total number need more weight in model fitting

Fit the model

Option 1: Give **proportion of success** as response AND **total number** as weights

```
mod1 <- glm(
  formula = p.surv ~ ..., # choose predictors according to your hypothesis
  data = pdogs,
  family = "binomial",
  weights = n.tot
)
```

- `family = "binomial"` by default uses logit link function

Fit the model

Option 2: Give a **table with two columns** as response:

- No of successes
- No of failures

```
mod1 <- glm(
  cbind(n.surv, n.death) ~ ...,
  data = pdogs,
  family = "binomial"
)
```

Now you

Task 4: Binomial glms with proportion data

Find the task description here