

Onlinebeispiel 8.4. Einfluss einer Vegetationsentfernung auf die Molchlarven in Auengewässern

Kapitel 8.3.1 aus Henle, K., A. Grimm-Seyfarth & B. Gruber: Erfassung und Analyse von Tierpopulationen. Ulmer Verlag

Annegret Grimm-Seyfarth

22.04.2025

In diesem Beispiel zeigen wir eine einfache Vorher-Nachher-Kontrolle-Effekt-Analyse (engl. Before-After-Control-Impact Assessment, oder kurz BACI-Analyse) in R. Für klassische Regressionsanalysen mittels `lm` und `glm` sind keine weiteren R-Pakete nötig, jedoch für die `glm.nb` Modelle das Paket MASS (Venables & Ripley 2002). Allerdings erleichtert das R-Paket MuMIn (Bartón 2024) den Umgang mit Modellselektion und das R-Paket emmeans (Lenth 2025) die Interpretation von Interaktionen. Für die Modelldiagnostik nutzen wir die Pakete DHARMA (Hartig 2024) und performance (Lüdtke et al. 2021). Außerdem nutzen wir das Paket ggplot2 für Visualisierungen (Wickham 2016). Als Datensatz nutzen wir ein Larvenmonitoring im Leipziger Auwald aus zwei aufeinanderfolgenden Jahren, in dem ein Gewässer zwischen den Erfassungen von Vegetation freigeschnitten wurde.

```
# check.packages function: install and load multiple R packages.
# Function from: https://gist.github.com/smithdanielle/9913897
check.packages <- function(pkg){
  new.pkg <- pkg[!(pkg %in% installed.packages()[, "Package"])]
  if (length(new.pkg))
    install.packages(new.pkg, dependencies = TRUE, type = "source")
  sapply(pkg, require, character.only = TRUE)
}

# benötigte R pakete
pakete <- c("MASS", "MuMIn", "emmeans", "ggplot2", "DHARMA", "performance")

# Prüfe und installiere
check.packages(pakete)
```

| | | | | | | |
|----|------|-------|---------|---------|--------|-------------|
| ## | MASS | MuMIn | emmeans | ggplot2 | DHARMA | performance |
| ## | TRUE | TRUE | TRUE | TRUE | TRUE | TRUE |

Daten einladen

Im Leipziger Auwald leben viele Amphibien in den temporären Gewässern (Lachen). Im Zuge eines Renaturierungsprojektes stehen die Larven von Kamm- (*Triturus cristatus*) und Teichmolchen (*Lissotriton vulgaris*) im Fokus. Im Herbst 2023 wurde ein Gewässer im Zuge einer Pflegemaßnahme für eine Oberleitung komplett von Vegetation freigeschnitten. Wir wollen nun die Auswirkungen auf den Reproduktionserfolg

der beiden Molcharten testen. Dazu nutzen wir neben dem Einflussgewässer noch ein weiteres Gewässer in naher Umgebung, das jedoch weit genug weg ist, um von der Maßnahme nicht betroffen zu sein.

Beide Gewässer wurden jeweils 2023 und 2024 im Juni für fünf aufeinanderfolgende Tage mittels Minnow Traps befangen. Da jeweils die gleiche Fallenzahl genutzt wurde (6 Fallen, jeweils 2 Öffnungen), können die Daten direkt als relative Häufigkeiten verglichen werden. Andernfalls hätte man einen Index berechnen müssen, der die Anzahl Fallen bzw. Fallenöffnungen und Fangtage berücksichtigt.

Zunächst laden wir die Daten ein:

```
UlmerBuch::beispiel.pfad() #Pfad zu den Beispieldaten
```

```
## Der Pfad zu den Beispieldaten wurde gesetzt auf:  
## C:/Users/grimm/AppData/Local/Programs/R/R-4.4.3/library/UlmerBuch/extdata
```

```
BAdata <- read.csv2("Burgaue_Larven.csv")  
head(BAdata)
```

```
##   Saison      Date      Group Time LV_larvae_Numbers_of_Individual  
## 1      3 15.06.2023 Control1 Before                               1  
## 2      3 15.06.2023 Treatment Before                               1  
## 3      3 16.06.2023 Control1 Before                               1  
## 4      3 16.06.2023 Treatment Before                               1  
## 5      3 17.06.2023 Control1 Before                               1  
## 6      3 17.06.2023 Treatment Before                               2  
##   TC_larvae_Numbers_of_Individual  
## 1                                     2  
## 2                                     1  
## 3                                     2  
## 4                                     1  
## 5                                     1  
## 6                                     0
```

Das Kontrollgewässer wurde hier als “Control1”, das Einflussgewässer als “Treatment” benannt (Parameter Group). Als Zeit gilt 2023 als “Before” und 2024 als “After” (Parameter Time).

Da in R standardmäßig alle Faktoren alphanumerisch sortiert werden, müssen wir für die Zeit die Sortierung ändern.

```
BAdata$Time <- factor(BAdata$Time, levels = c("Before", "After"))
```

Generalisiertes lineares Modell

Da wir hier als Ergebnisvariable Zähldaten verwenden, sind die Residuen in unserem Modell Poissonverteilt. Wir bauen nun die beiden Modelle mit der Interaktion aus Group und Time, dargestellt durch ein *.

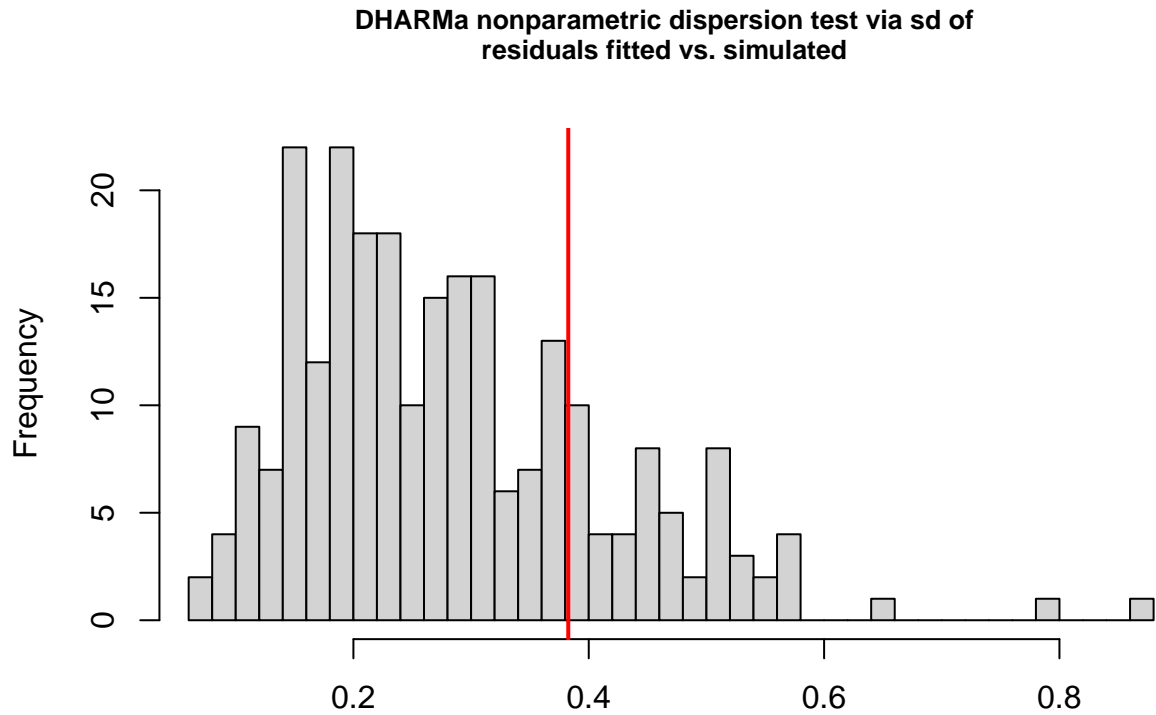
```
# Teichmolch  
glm.lv <- glm(LV_larvae_Numbers_of_Individual ~ Group * Time,  
              data = BAdata, family = "poisson")  
# Kammolch  
glm.tc <- glm(TC_larvae_Numbers_of_Individual ~ Group * Time,  
              data = BAdata, family = "poisson")
```

Bevor wir weitermachen, müssen wir noch prüfen, dass die Modelle tatsächlich valide sind, also die Poissonverteilung wirklich zutrifft. Dazu nutzen wir den DHARMA-Verteilungstest aus dem R-Paket DHARMA (Hartig 2024). Der Hintergrund ist, dass eine Poissonverteilung dadurch charakterisiert ist, dass der Mittelwert der Varianz (als der Streuung) entspricht. Aber in echten Daten ist das selten so: manchmal sind die Daten zu stark gestreut (= Overdispersion, z. B. weil in manchen Teichen extrem viele Molche gezählt werden und in anderen sehr wenige), und manchmal sind sie zu wenig gestreut (= Underdispersion, z. B. weil die Zahlen „zu regelmäßig“ sind). Ein Modell, das die Streuung falsch einschätzt, kann zu falschen Tests und p-Werten führen. Der nichtparametrische Verteilungstest vergleicht dann: Wie groß ist die Standardabweichung der Residuen in den echten Daten, verglichen mit der Streuung in den simulierten Daten? Wenn sie ungefähr gleich ist, passt die Poissonverteilung. Wenn die echten Residuen viel stärker schwanken, liegt Overdispersion vor. Wenn die echten Residuen viel gleichmäßiger sind, liegt Underdispersion vor. Das Ergebnis ist ein p-Wert: nicht signifikant ($p > 0.05$) bedeutet, das Modell erklärt die Streuung gut, signifikant ($p < 0.05$) bedeutet, das Modell passt nicht zur Streuung (Über- oder Underdispersion). In letzterem Fall müssen wir eine andere Verteilung nehmen (z. B. Negative-Binomial statt Poisson). Mittels `simulationOutput` kann man den Verteilungstest auch grafisch darstellen. Hierbei werden automatisch Achsenbeschriftungen erstellt, diese sind daher in Englisch. Man bekommt die vier Standarddiagnosen (Plots + Tests): Quantile-Plot / QQ-Plot der Residuen (prüft, ob die simulierten Residuen gleichverteilt sind); ein Histogramm der Residuen (visuelle Prüfung: Ist die Verteilung der Residuen flach (Uniform)?); Residuals vs. Predicted (Verteilungstest, prüft nach Over- oder Underdispersion); und Residuals vs. Regressors (zeigt, ob Residuen mit den erklärenden Variablen oder der Zeit korreliert sind, dient zum Erkennen von fehlenden Prädiktoren oder Autokorrelation). Das praktische daran ist, dass Auffälligkeiten immer rot dargestellt werden.

Weiterhin prüfen wir auf sogenannte Zero-inflation (d. h., zu viele Nullen im Datensatz) mithilfe des R-Paketes `performance` (Lüdtke et al. 2021). Zu viele Nullen entstehen, wenn in manchen Gewässern keine Individuen beobachtet wurden. Ein „normales“ Poisson- oder Negative-Binomial-Modell geht aber nur von einer bestimmten Anzahl Nullen aus (basierend auf der erwarteten Verteilung). Wenn viel mehr Nullen in den Daten stecken, als das Modell erlaubt, spricht man von Zero-Inflation. Der Zero-Inflation-Test schaut nach: Wie viele Nullen hat mein Modell erwartet? Wie viele Nullen haben die echten Daten tatsächlich? Dann wird statistisch getestet, ob die Differenz größer ist, als man durch Zufall erwarten würde. Wenn das Modell ungefähr gleich viele Nullen erwartet wie beobachtet, passt die Verteilung. Wenn die echten Daten viel mehr Nullen enthalten als das Modell, sprechen wir von einer signifikanten Zero-Inflation. In seltenen Fällen gibt es auch zu wenige Nullen, dann passt das Modell ebenfalls nicht gut. Bei Zero-Inflation einer Poissonverteilung kann man hier ebenfalls zunächst die Negativ-Binomialverteilung prüfen, und wenn das Problem weiter besteht, müssen spezielle Zero-Inflated-Modelle oder das Hurdle-Modell verwendet werden, das Nullen gesondert modelliert.

Führen wir die Tests nach Art getrennt nacheinander durch:

```
# Teichmolch
# Verteilungstest
testDispersion(glm.lv)
```

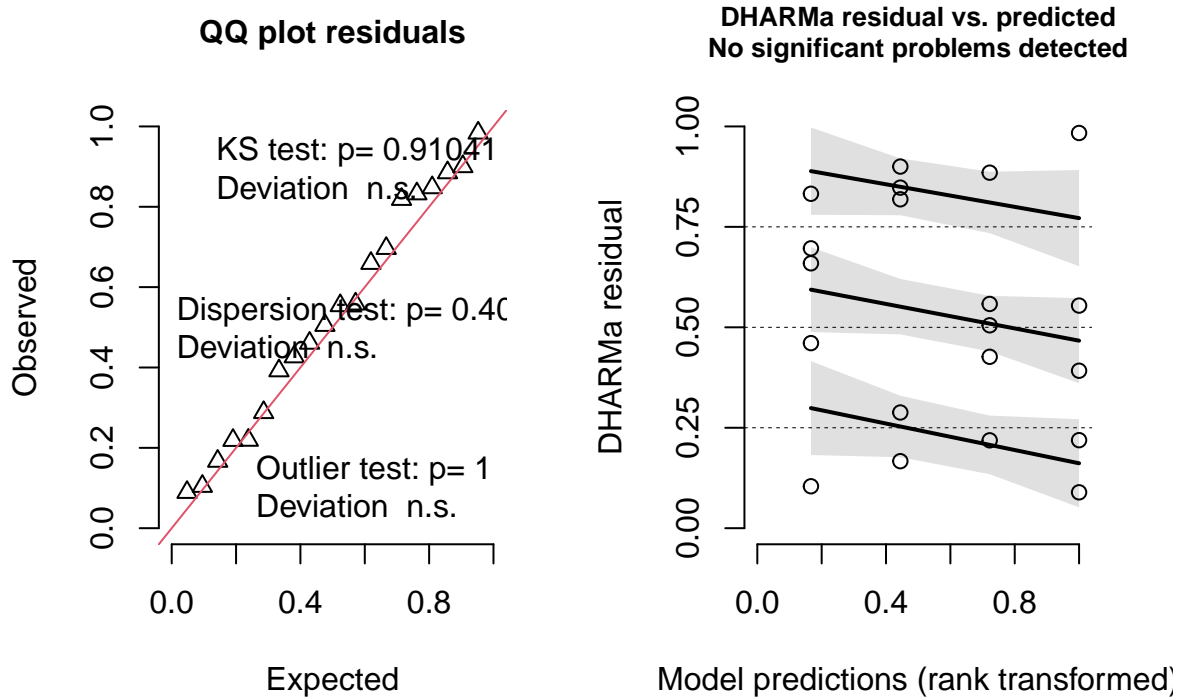


```
##
## DHARMA nonparametric dispersion test via sd of residuals fitted vs.
## simulated
##
## data: simulationOutput
## dispersion = 1.3559, p-value = 0.408
## alternative hypothesis: two.sided
```

```
# kein signifikanter Unterschied, Modell ist ok
```

```
# Grafische Darstellung der Verteilungstests
simulationOutput <- simulateResiduals(fittedModel = glm.lv, plot = T)
```

DHARMA residual



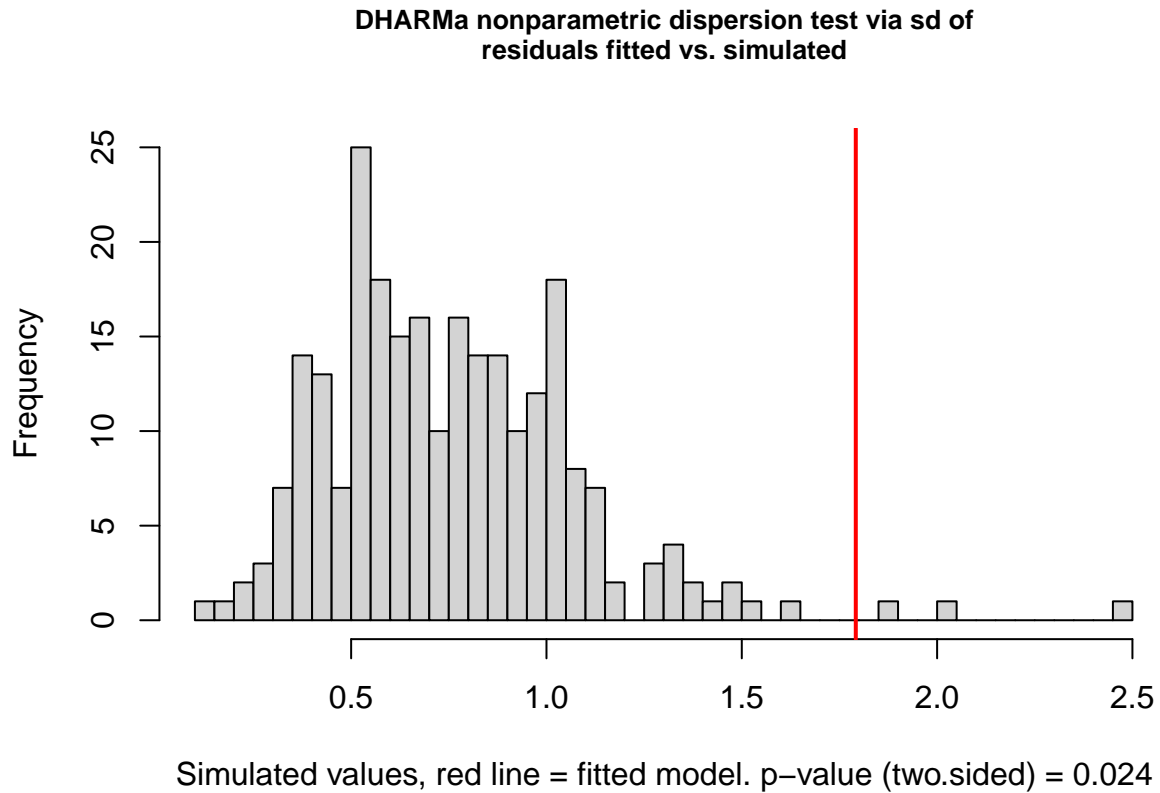
```
# keine signifikanten Unterschiede, Modell ist ok
```

```
# Test auf Zero-inflation
check_zeroinflation(glm.lv)
```

```
## # Check for zero-inflation
##
##   Observed zeros: 3
##   Predicted zeros: 4
##   Ratio: 1.33
```

```
# keine Zero-inflation, Modell ist ok
```

```
# Kammolch
# Verteilungstest
testDispersion(glm.tc)
```

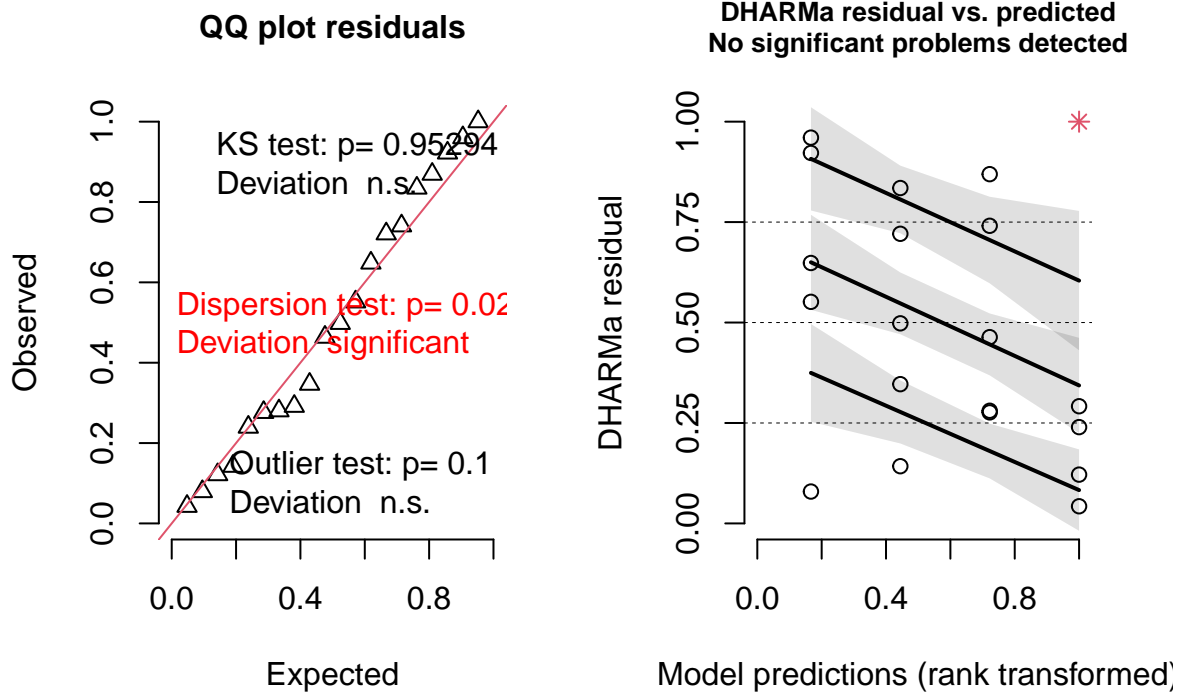


```
##
## DHARMA nonparametric dispersion test via sd of residuals fitted vs.
## simulated
##
## data: simulationOutput
## dispersion = 2.372, p-value = 0.024
## alternative hypothesis: two.sided
```

```
# signifikanter Unterschied, Verteilung falsch
```

```
# Grafische Darstellung der Verteilungstests
simulationOutput <- simulateResiduals(fittedModel = glm.tc, plot = T)
```

DHARMA residual



```
# Dispersionstest signifikant, Verteilung falsch
```

```
# Test auf Zero-inflation
check_zeroinflation(glm.tc)
```

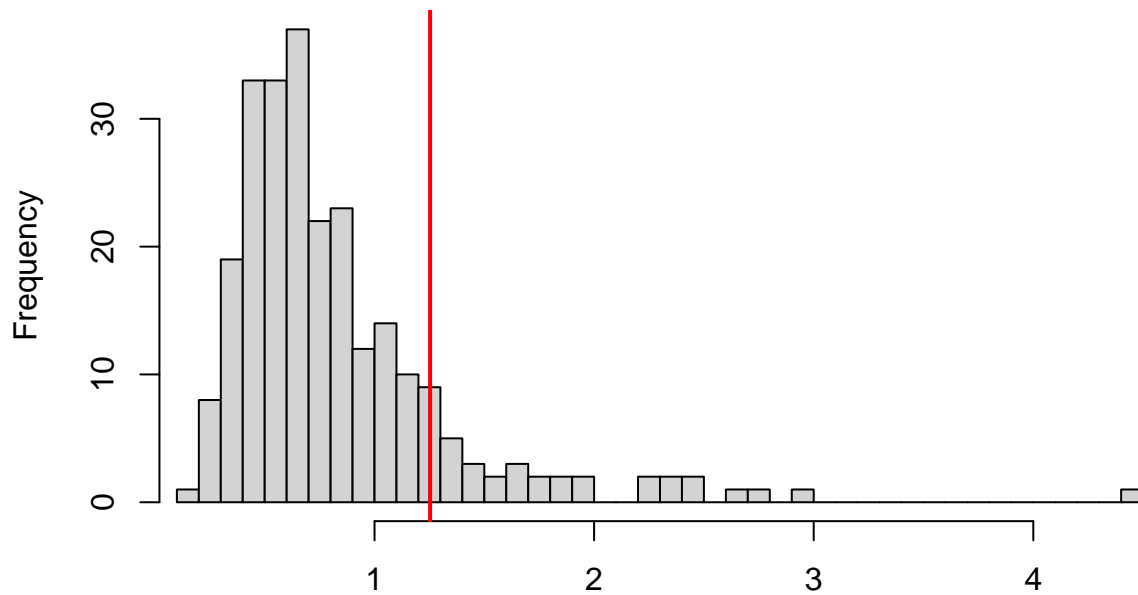
```
## # Check for zero-inflation
##
##   Observed zeros: 7
##   Predicted zeros: 8
##   Ratio: 1.14
```

```
# keine Zero-inflation
```

Für den Teichmolch ist das Modell unauffällig, die Poissonverteilung ist also korrekt. Für den Kammmolch gibt es jedoch eine signifikante Abweichung hinsichtlich der Verteilung ($p = 0,024$). Daher nutzen wir alternativ zur Poisson-Verteilung die Negativ-Binomialverteilung. Diese erstellen wir hiermit und prüfen gleich im Anschluss erneut die Verteilung sowie die Zero-Inflation:

```
glm.tc <- glm.nb(TC_larvae_Numbers_of_Individual ~ Group * Time,
  data = BAdata)
# Verteilungstest
testDispersion(glm.tc)
```

**DHARMA nonparametric dispersion test via sd of
residuals fitted vs. simulated**



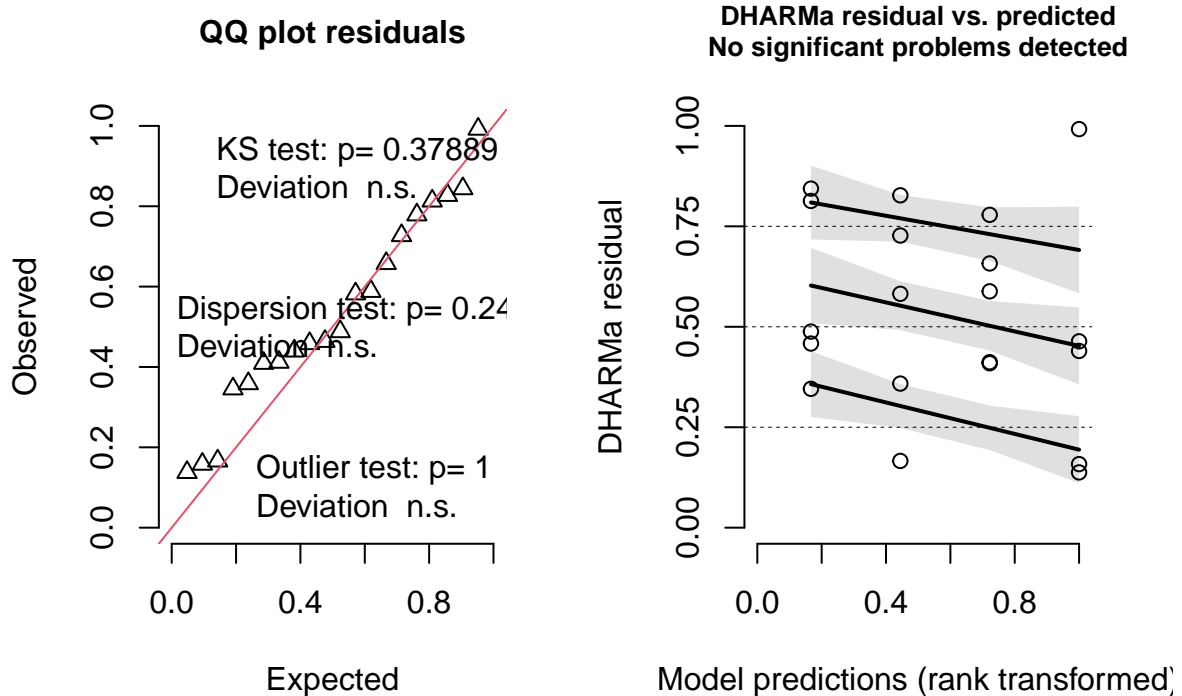
Simulated values, red line = fitted model. p-value (two.sided) = 0.248

```
##
## DHARMA nonparametric dispersion test via sd of residuals fitted vs.
## simulated
##
## data: simulationOutput
## dispersion = 1.5106, p-value = 0.248
## alternative hypothesis: two.sided
```

kein signifikanter Unterschied, Modell ist ok


```
# Grafische Darstellung der Verteilungstests
simulationOutput <- simulateResiduals(fittedModel = glm.tc, plot = T)
```

DHARMA residual



```
# keine signifikanten Unterschiede, Modell ist ok
```

```
# Test auf Zero-inflation
check_zeroinflation(glm.tc)
```

```
## # Check for zero-inflation
##
##   Observed zeros: 7
##   Predicted zeros: 9
##   Ratio: 1.26
```

```
# keine Zero-inflation, Modell ist ok
```

Nun gibt es keine weiteren Abweichungen, das Modell für den Kammolch ist ebenfalls ok.

Schauen wir uns nun die Analysen der Arten nun im Detail an.

Teichmolch

Vergleichen wir zunächst die Modelle mit jeweils linearen Kombinationen der Parameter. Dazu nutzen wir die `dredge` Funktion aus dem R-Paket `MuMIn`. Eine Einführung in Modellvergleiche findet sich in Kapitel 9.3 des Buches.

```
# Modellvergleich
```

```
glm.lv.mod <- dredge(glm.lv, options(na.action = "na.fail"))
glm.lv.mod
```

```
## Global model call: glm(formula = LV_larvae_Numbers_of_Individual ~ Group * Time,
##      family = "poisson", data = BAdata)
## ---
## Model selection table
##      (Int) Grp Tim Grp:Tim df   logLik  AICc delta weight
## 4 -0.68960  +   +           3 -33.412  74.3  0.00  0.716
## 8 -0.22310  +   +           + 4 -32.756  76.2  1.85  0.283
## 2  0.09531  +           2 -42.316  89.3 15.01  0.000
## 3  0.26240      +           2 -44.962  94.6 20.31  0.000
## 1  1.04700           1 -53.866 110.0 35.63  0.000
## Models ranked by AICc(x)
```

Wir sehen, dass die lineare Kombination aus Group und Time besser abschneidet als die Interaktion. Allerdings ist das Modell mit Interaktion nur 1,85 AICc Punkte schlechter - ein Einfluss der Intervention (Freischneidung der Fläche) kann also nicht ausgeschlossen werden.

Schauen wir uns die Schätzwerte der beiden besten Modelle an.

```
get.models(glm.lv.mod, subset = 1:2)
```

```
## $'4'
##
## Call:  glm(formula = LV_larvae_Numbers_of_Individual ~ Group + Time +
##      1, family = "poisson", data = BAdata)
##
## Coefficients:
##      (Intercept)  GroupTreatment      TimeAfter
##          -0.6896         1.4307         1.2192
##
## Degrees of Freedom: 19 Total (i.e. Null);  17 Residual
## Null Deviance:      60.44
## Residual Deviance: 19.54      AIC: 72.82
##
## $'8'
##
## Call:  glm(formula = LV_larvae_Numbers_of_Individual ~ Group + Time +
##      Group:Time + 1, family = "poisson", data = BAdata)
##
## Coefficients:
##      (Intercept)      GroupTreatment      TimeAfter
##          -0.2231         0.8109         0.5596
## GroupTreatment:TimeAfter
##          0.8541
##
## Degrees of Freedom: 19 Total (i.e. Null);  16 Residual
## Null Deviance:      60.44
## Residual Deviance: 18.22      AIC: 73.51
##
## attr("rank")
```

```
## function (x)
## do.call("rank", list(x))
## <environment: 0x0000022a8ad9f7e0>
## attr("call")
## AICc(x)
## attr("class")
## [1] "function"      "rankFunction"
## attr("beta")
## [1] "none"
```

Im Treatment-Gewässer gab es mehr Teichmolchlarven, ebenso gab es 2024 mehr als 2023.

Schauen wir noch nach möglicher Signifikanz der Parameter

```
drop1(glm.lv, test="Chi")
```

```
## Single term deletions
##
## Model:
## LV_larvae_Numbers_of_Individual ~ Group * Time
##           Df Deviance   AIC    LRT Pr(>Chi)
## <none>           18.222 73.511
## Group:Time  1    19.535 72.824 1.3128  0.2519
```

```
drop1(glm(LV_larvae_Numbers_of_Individual ~ Group + Time,
          data = BAdat, family = "poisson"),
      test="Chi")
```

```
## Single term deletions
##
## Model:
## LV_larvae_Numbers_of_Individual ~ Group + Time
##           Df Deviance   AIC    LRT  Pr(>Chi)
## <none>           19.535 72.824
## Group    1    42.635 93.924 23.100 1.538e-06 ***
## Time     1    37.344 88.632 17.808 2.443e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Während die Gesamtinteraktion nicht signifikant ist, sind es die Einzelparame-ter ohne Interaktion. Wenn es einen Interaktionseffekt (also Einfluss der Intervention) gibt, ist dieser schwächer als die Effekte der einzelnen Parameter.

Interaktionseffekte

Die einzelnen Interaktionseffekte sind im Modell oben schwer zu interpretieren. Einfacher können wir sie uns auch mithilfe des `emmeans`-Paketes anschauen. Die Funktion `emmeans` berechnet die sogenannten Estimated Marginal Means (= erwartete Mittelwerte), also die Mittelwerte nach dem Modell, nicht die rohen Mittelwerte aus den Daten. Das funktioniert folgendermaßen: Dein Modell wurde in einer bestimmten Link-Funktion geschätzt. Bei einem Poisson-GLM ist das die log-Link:

$$\log(\mu) = \alpha + \beta \cdot x$$

Das Modell arbeitet also auf der Log-Skala. Die Funktion `emmeans` berechnet die erwarteten Werte auf dieser Modellskala (= Logarithmus der erwarteten Anzahl). Deshalb siehst du die Ergebnisse erstmal auf der Log-Skala. Die Log-Skala hat den Vorteil, dass Unterschiede in den Gruppen zu Additionen werden (linearer Zusammenhang). Auf der Originalskala wären es Multiplikationen (Exponent). Schauen wir uns das an:

```
l.em <- emmeans(glm.lv, ~ Group * Time)
l.em

## Group      Time    emmean      SE df asymp.LCL asymp.UCL
## Control1 Before -0.223 0.500 Inf   -1.2031    0.757
## Treatment Before  0.588 0.333 Inf   -0.0655    1.241
## Control1 After  0.336 0.378 Inf   -0.4043    1.077
## Treatment After  2.001 0.164 Inf    1.6793    2.324
##
## Results are given on the log (not the response) scale.
## Confidence level used: 0.95
```

Wir sehen die Gruppe (`Control1` vs. `Treatment`) und die entsprechende Zeit (`Before` und `After`), sowie die für die entsprechende Kombination berechneten Mittelwerte auf der Log-Skala (Spalte `emmean`). Im `Treatment`-Gewässer waren also auch 2023 (`Before`) bereits mehr Teichmolchlarven als im `Control1`-Gewässer, jedoch ist der Unterschied zwischen den Gewässern in 2024 (`After`) deutlich stärker zu sehen. In beiden Gewässern gab es 2024 allerdings mehr Teichmolchlarven als 2023, weshalb der mögliche positive Effekt der Pflegemaßnahme maskiert sein könnte.

Die geschätzten Mittelwerte können natürlich auch zurücktransformiert werden:

```
l.em <- as.data.frame(l.em)
l.em$emmean_original <- exp(l.em$emmean)
l.em

## Group      Time    emmean      SE df asymp.LCL asymp.UCL emmean_original
## Control1 Before -0.2231436 0.5000000 Inf -1.2031255 0.7568384          0.8
## Treatment Before  0.5877867 0.3333333 Inf -0.0655347 1.2411080          1.8
## Control1 After  0.3364722 0.3779637 Inf -0.4043230 1.0772675          1.4
## Treatment After  2.0014800 0.1643990 Inf  1.6792639 2.3236961          7.4
##
## Results are given on the log (not the response) scale.
## Confidence level used: 0.95
```

Hier sehen wir den Unterschied noch deutlich stärker (Spalte `emmean_original`), im `Treatment`-Gewässer waren nach der Pflegemaßnahme deutlich mehr Teichmolchlarven (letzter Wert) als im `Control1`-Gewässer oder im Jahr vor der Pflegemaßnahme.

Grafische Darstellung

Stellen wir die Rohdaten auch noch einmal direkt gegenüber. Achtung, hier handelt es sich nicht wie zuvor um Schätzwerte.

```
ggplot(BAdata, aes(x = Group, y = LV_larvae_Numbers_of_Individual,
                  fill = Time, group = interaction(Group, Time))) +
  geom_boxplot() +
```

```

labs(
  x = "",
  y = "Larvenzahl",
  title = "Veränderung der Teichmolchlarven vor und nach dem Eingriff",
  fill = "Zeit"
) +
theme_classic() +
scale_fill_manual(
  values = c("Before" = "yellowgreen", "After" = "seagreen") # Anpassung der Farben
) +
theme(
  text = element_text(family = "Times"),

  # Achsentitel
  axis.title.x = element_text(size = 18,color = "black"),
  axis.title.y = element_text(size = 18,color = "black"),

  # Achsenbeschriftungen
  axis.text.x = element_text(size = 18, color = "black"),
  axis.text.y = element_text(size = 18,color = "black"),

  # Titel
  plot.title = element_text(size = 14, face = "bold", hjust = 0.5,color = "black"),

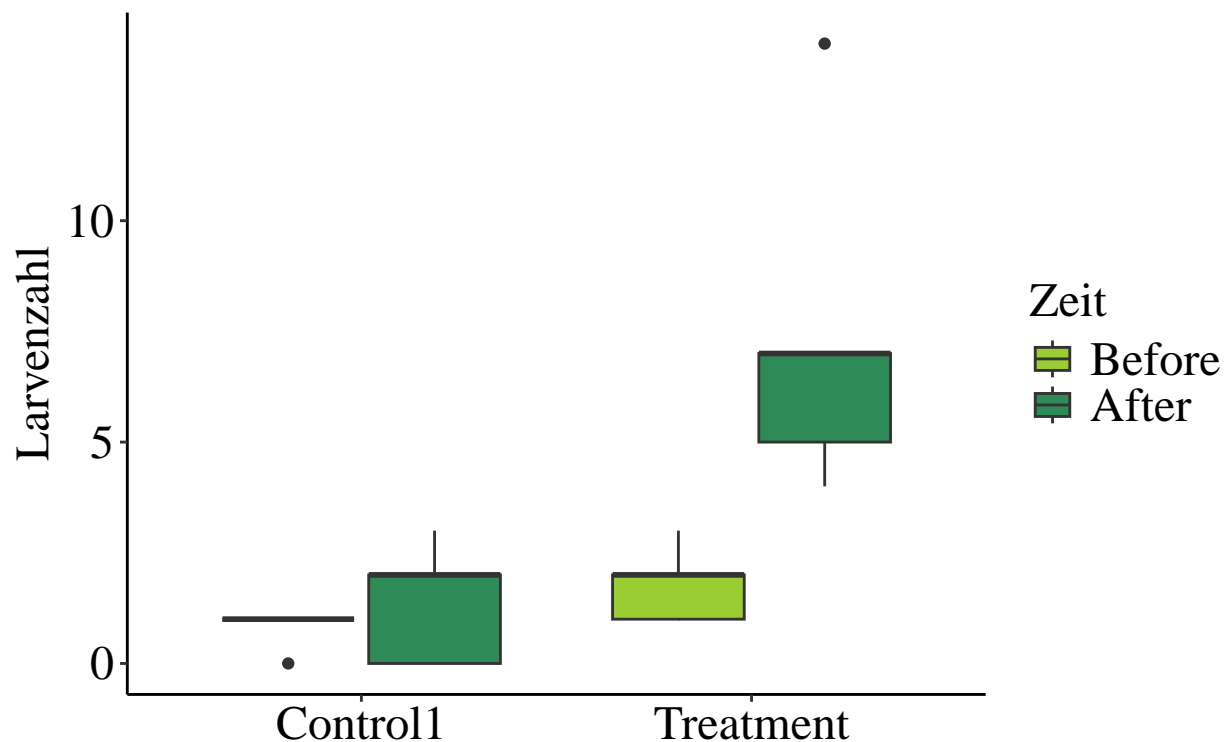
  #legend.key.size = unit(1.5, "cm"),

  # Ränder
  #plot.margin = margin(t = 50, r = 20, b = 20, l = 20),

  # Legende
  legend.title = element_text(size = 18,color = "black"),
  legend.text = element_text(size = 18,color = "black"),
  legend.position = "right"
)

```

Veränderung der Teichmolchlarven vor und nach dem Eingriff



Dieses Ergebnis deckt sich grundsätzlich mit den Modellergebnissen: im **Treatment**-Gewässer waren nach der Pflegemaßnahme deutlich mehr Teichmolchlarven als im **Control1**-Gewässer oder im Jahr vor der Pflegemaßnahme. Achtung: in einem Boxplot werden keine Mittelwerte, sondern Mediane dargestellt, weshalb der Zahlenwert weiter von den Mittelwerten aus `emmeans` abweichen kann.

Kammmolch

Vergleichen wir zunächst die Modelle mit jeweils linearen Kombinationen der Parameter. Dazu nutzen wir die `dredge` Funktion aus dem R-Paket `MuMIn`.

```
# Modellvergleich
glm.tc.mod <- dredge(glm.tc, options(na.action = "na.fail"))
glm.tc.mod
```

```
## Global model call: glm.nb(formula = TC_larvae_Numbers_of_Individual ~ Group * Time,
##   data = BAdat, init.theta = 2.336382242, link = log)
## ---
## Model selection table
##      (Int) Grp Tim Grp:Tim df  logLik AICc delta weight
## 1  0.13980      2      2 -29.532 63.8  0.00  0.579
## 2 -0.10540    +      3 -29.218 65.9  2.17  0.196
## 3  0.09531      +      3 -29.520 66.5  2.77  0.145
## 4 -0.10510    +    +      4 -29.218 69.1  5.33  0.040
## 8  0.33650    +    +    +      5 -27.414 69.1  5.34  0.040
## Models ranked by AICc(x)
```

Wir sehen, dass es hier eigentlich keinen Einfluss der Intervention oder der Zeit gibt, da das zweitbeste Modell bereits mehr als 2 AICc-Punkte dahinter liegt.

Schauen wir uns dennoch die Schätzwerte der drei besten Modelle an.

```
get.models(glm.tc.mod, subset = c(1:3))
```

```
## $'1'
##
## Call: glm.nb(formula = TC_larvae_Numbers_of_Individual ~ 1, data = BAdata,
##   init.theta = 1.476495851, link = log)
##
## Coefficients:
## (Intercept)
##      0.1398
##
## Degrees of Freedom: 19 Total (i.e. Null);  19 Residual
## Null Deviance:      19.52
## Residual Deviance: 19.52      AIC: 63.06
##
## $'2'
##
## Call: glm.nb(formula = TC_larvae_Numbers_of_Individual ~ Group + 1,
##   data = BAdata, init.theta = 1.583007986, link = log)
##
## Coefficients:
## (Intercept)  GroupTreatment
##      -0.1054      0.4418
##
## Degrees of Freedom: 19 Total (i.e. Null);  18 Residual
## Null Deviance:      20.02
## Residual Deviance: 19.38      AIC: 64.44
##
## $'3'
##
## Call: glm.nb(formula = TC_larvae_Numbers_of_Individual ~ Time + 1,
##   data = BAdata, init.theta = 1.48050514, link = log)
##
## Coefficients:
## (Intercept)  TimeAfter
##      0.09531      0.08701
##
## Degrees of Freedom: 19 Total (i.e. Null);  18 Residual
## Null Deviance:      19.54
## Residual Deviance: 19.51      AIC: 65.04
##
## attr("rank")
## function (x)
## do.call("rank", list(x))
## <environment: 0x0000022a8fc8cf90>
## attr("call")
## AICc(x)
## attr("class")
## [1] "function"      "rankFunction"
```

```
## attr("beta")
## [1] "none"
```

Im Treatment-Gewässer gab es insgesamt über beide Jahre tendenziell mehr Kammolchlarven. Der Zeiteffekt ist nur schwach ausgeprägt, es gibt aber ebenfalls den Trend, dass 2024 mehr Kammolchlarven vorhanden waren als 2023.

Schauen wir noch nach möglicher Signifikanz der Parameter

```
drop1(glm.tc, test="Chi")
```

```
## Single term deletions
##
## Model:
## TC_larvae_Numbers_of_Individual ~ Group * Time
##           Df Deviance   AIC    LRT Pr(>Chi)
## <none>           18.157 62.828
## Group:Time  1    22.011 64.682 3.8539  0.04963 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
drop1(glm.nb(TC_larvae_Numbers_of_Individual ~ Group + Time,
             data = BAdata),
      test="Chi")
```

```
## Single term deletions
##
## Model:
## TC_larvae_Numbers_of_Individual ~ Group + Time
##           Df Deviance   AIC    LRT Pr(>Chi)
## <none>           19.381 64.437
## Group    1    19.994 63.049 0.61246  0.4339
## Time     1    19.381 62.437 0.00000  0.9992
```

In diesem Modell ist die Gesamtinteraktion signifikant, während die Einzelparame-ter (eines Modelles ohne Interaktion) es nicht wären. Der Likelihood-Ratio-Test zeigt also doch einen möglichen Interaktionseffekt und damit einen Effekt der Maßnahme.

Interaktionseffekte

Die einzelnen Interaktionseffekte sind im Modell oben schwer zu interpretieren. Einfacher können wir sie uns auch mithilfe des emmeans-Paketes anschauen. Die Funktion **emmeans** berechnet die sogenannten Estimated Marginal Means (= erwartete Mittelwerte), also die Mittelwerte nach dem Modell, nicht die rohen Mittelwerte aus den Daten. Wie das funktioniert, wurde beim Teichmolch bereits erklärt.

```
l.em <- emmeans(glm.tc, ~ Group * Time)
l.em
```

```
## Group    Time    emmean    SE df asymp.LCL asymp.UCL
## Control1 Before  0.336 0.478 Inf   -0.600    1.273
## Treatment Before -0.223 0.579 Inf   -1.359    0.912
## Control1 After -0.916 0.765 Inf   -2.416    0.584
```



```
## Treatment After 0.693 0.431 Inf -0.151 1.538
##
## Results are given on the log (not the response) scale.
## Confidence level used: 0.95
```

Wir sehen die Gruppe (Control1 vs. Treatment) und die entsprechende Zeit (Before und After), sowie die für die entsprechende Kombination berechneten Mittelwerte auf der Log-Skala (Spalte `emmean`). Während 2023 (Before) im Treatment-Gewässer weniger Kammolchlarven als im Kontrollgewässer waren (mit nur geringem Unterschied), befanden sich nach der Maßnahme (After) im Treatment-Gewässer deutlich mehr Kammolchlarven, im Kontrollgewässer deutlich weniger als im Vorjahr. Die Unterschiede sind jedoch nicht sehr groß.

Die geschätzten Mittelwerte können natürlich auch zurücktransformiert werden:

```
l.em <- as.data.frame(l.em)
l.em$emmean_original <- exp(l.em$emmean)
l.em
```

```
## Group      Time      emmean      SE df  asymp.LCL asymp.UCL emmean_original
## Control1 Before 0.3364722 0.4779744 Inf -0.6003405 1.2732849      1.4
## Treatment Before -0.2231436 0.5793120 Inf -1.3585743 0.9122872      0.8
## Control1 After -0.9162907 0.7652466 Inf -2.4161466 0.5835651      0.4
## Treatment After 0.6931472 0.4308160 Inf -0.1512367 1.5375310      2.0
##
## Results are given on the log (not the response) scale.
## Confidence level used: 0.95
```

Hier sehen wir den Unterschied noch deutlich stärker (Spalte `emmean_original`), im Treatment-Gewässer waren nach der Pflegemaßnahme deutlich mehr Kammolchlarven (letzter Wert) als im Control1-Gewässer oder im Jahr vor der Pflegemaßnahme.

Grafische Darstellung

Stellen wir die Rohdaten auch noch einmal direkt gegenüber. Achtung, hier handelt es sich nicht wie zuvor um Schätzwerte.

```
ggplot(BAdata, aes(x = Group, y = TC_larvae_Numbers_of_Individual,
                    fill = Time, group = interaction(Group, Time))) +
  geom_boxplot() +
  labs(
    x = "",
    y = "Larvenzahl",
    title = "Veränderung der Kammolchlarven vor und nach dem Eingriff",
    fill = "Zeit"
  ) +
  theme_classic() +
  scale_fill_manual(
    values = c("Before" = "yellowgreen", "After" = "seagreen") # Anpassung der Farben
  ) +
  theme(
    text = element_text(family = "Times"),

    # Achsentitel
```

```

axis.title.x = element_text(size = 18,color = "black"),
axis.title.y = element_text(size = 18,color = "black"),

# Achsenbeschriftungen
axis.text.x = element_text(size = 18, color = "black"),
axis.text.y = element_text(size = 18,color = "black"),

# Titel
plot.title = element_text(size = 14, face = "bold", hjust = 0.5,color = "black"),

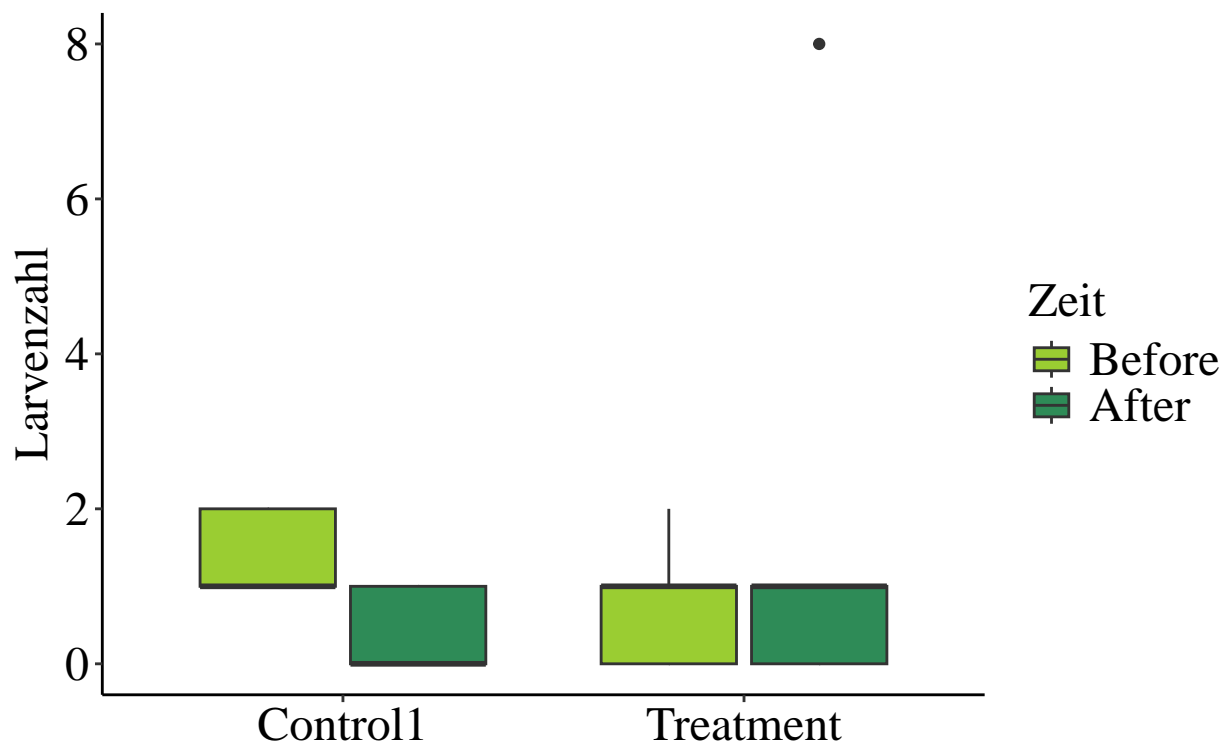
#legend.key.size = unit(1.5, "cm"),

# Ränder
#plot.margin = margin(t = 50, r = 20, b = 20, l = 20),

# Legende
legend.title = element_text(size = 18,color = "black"),
legend.text = element_text(size = 18,color = "black"),
legend.position = "right"
)

```

Veränderung der Kammolchlarven vor und nach dem Eingriff



Schauen wir uns nur die Boxplots an (also die Mediane der Rohdaten, nicht die geschätzten Mittelwerte der Interaktionen), sehen wir, dass die Kammolchlarven von 2023 nach 2024 eigentlich zurückgegangen sind, wie im Kontrollgewässer sichtbar. Im Gewässer mit der Pflegemaßnahme sind sie jedoch nicht zurückgegangen, was darauf schließen lässt, dass die Maßnahme für die Kammolche ebenfalls positiv war. Insgesamt hatte die Freischneidung also auf die Molchlarven beider Arten einen positiven Effekt.

Bei Analysen mit mehreren Gebieten vor und nach dem Eingriff mit jeweils mehreren Beobachtungen sollte man ggf. statt eines (generalisierten) linearen Modells ein gemischtes (generalisiertes) lineares Modell nutzen, bei dem das Gebiet als sogenannter **Random Intercept** eingegeben wird.

Literaturverzeichnis

- Bartoń, K. (2024): MuMIn: Multi-model inference. R package version 1.48.9/r534, <https://R-Forge.R-project.org/projects/mumin/>
- Hartig, F. (2024): DHARMA: Residual diagnostics for hierarchical (multi-level / mixed) regression models. R package version 0.4.7, <https://CRAN.R-project.org/package=DHARMA>
- Lenth, R. (2025): emmeans: Estimated marginal means, aka least-squares means. R package version 1.11.0, <https://CRAN.R-project.org/package=emmeans>
- Lüdtke, L., Ben-Shachar, M.S., Patil, I., Waggoner, P., Makowski, D. (2021): performance: An R package for assessment, comparison and testing of statistical models. — Journal of Open Source Software 6(60): 3139.
- Venables, W.N., Ripley, B.D. (2002): Modern applied statistics with S. Fourth Edition. — Springer, New York.
- Wickham, H. (2016): ggplot2: Elegant graphics for data analysis. — Springer-Verlag New York.