

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 erleichtern die Pakete MuMIn (Bartón 2024) und emmeans (Lenth 2025) den Umgang mit Modellselektion und 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"))
```

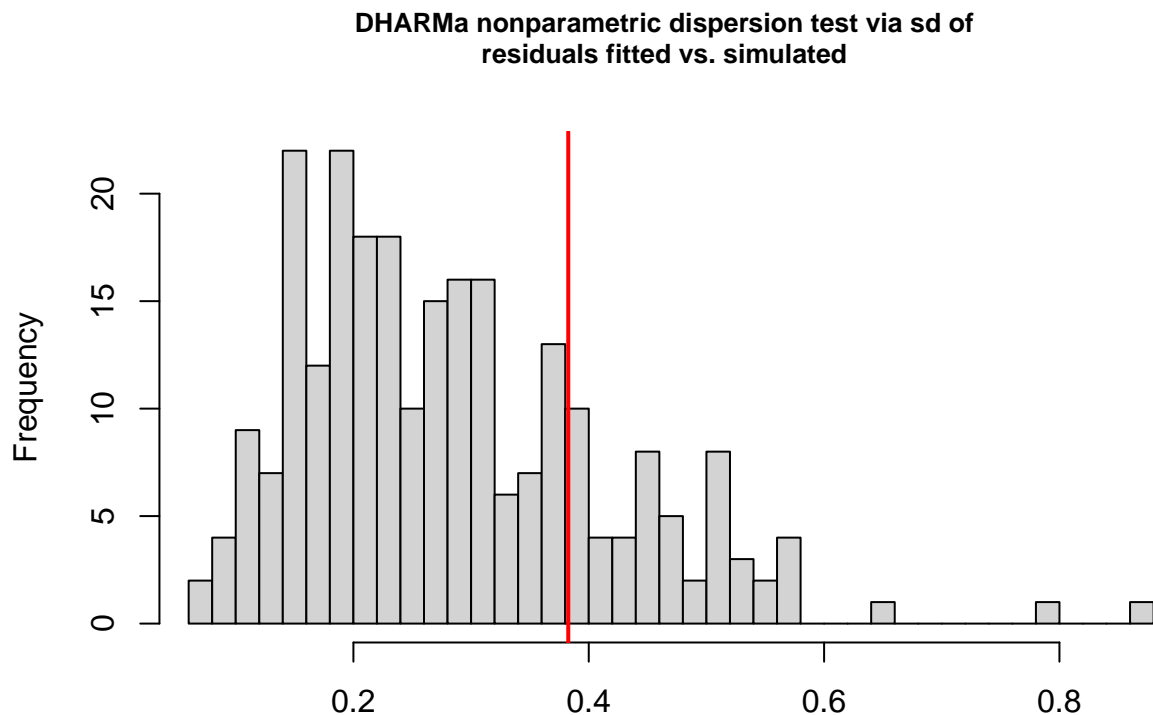
Generalisiertes lineares Modell

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. Dazu nutzen wir den Verteilungstest aus dem R-Paket DHARMA (Hartig 2024) und prüfen auf sogenannte Zero-inflation (d.h., zu viele 0en im Datensatz) mithilfe des R-Paketes performance (Lüdecke et al. 2021). Mittels `simulationOutput` kann man den Verteilungstest auch grafisch darstellen. Hierbei werden automatisch Achsenbeschriftungen erstellt, diese sind daher in Englisch.

```
testDispersion(glm.lv) # kein signifikanter Unterschied, Modell ist ok
```



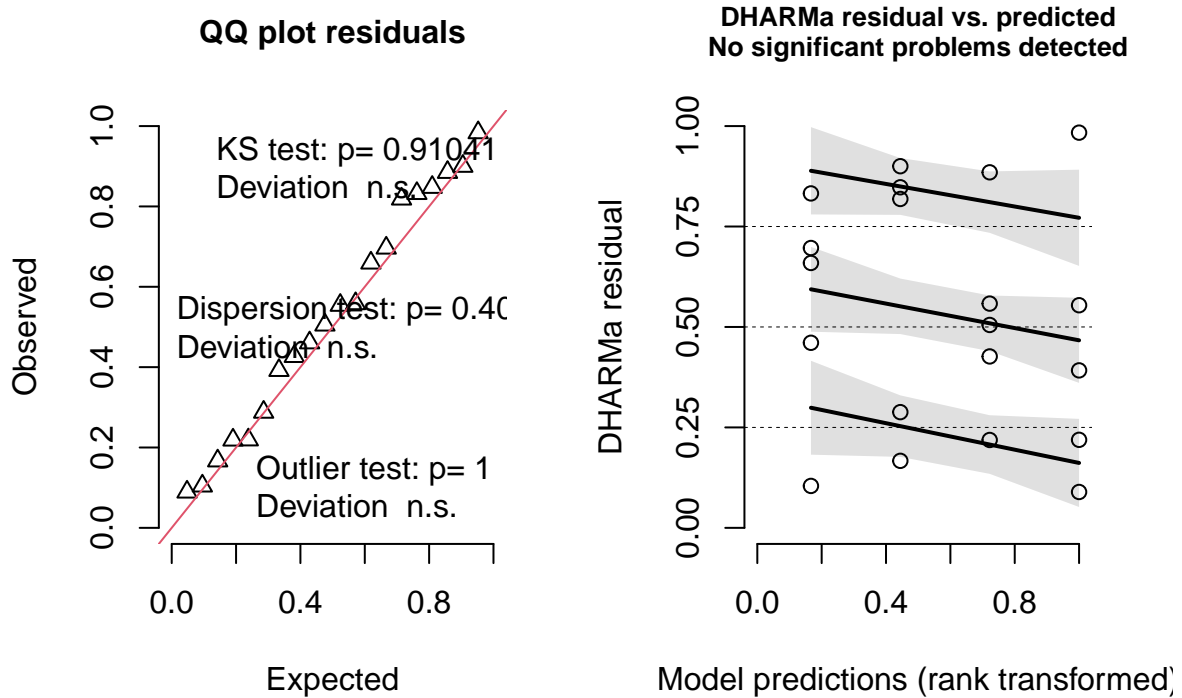
```
##
## DHARMA nonparametric dispersion test via sd of residuals fitted vs.
## simulated
##
## data: simulationOutput
## dispersion = 1.3559, p-value = 0.408
## alternative hypothesis: two.sided
```

```
check_zeroinflation(glm.lv) # keine Zero-inflation, Modell ist ok
```

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

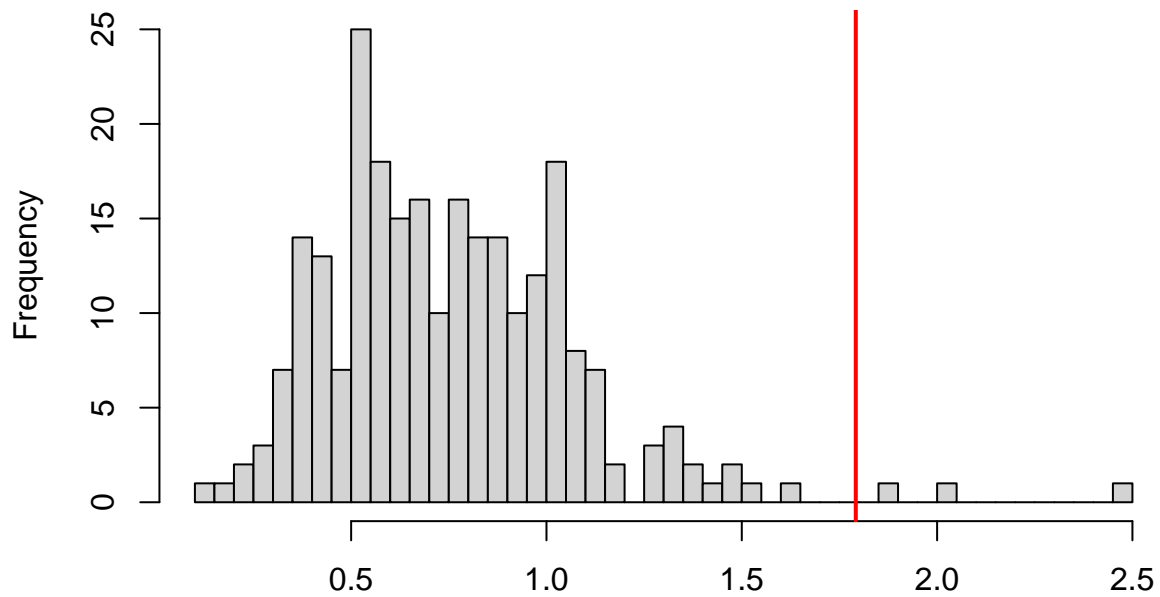
```
simulationOutput <- simulateResiduals(fittedModel = glm.lv, plot = T)
```

DHARMA residual



```
testDispersion(glm.tc) # signifikanter Unterschied, Verteilung falsch
```

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



Simulated values, red line = fitted model. p-value (two.sided) = 0.024

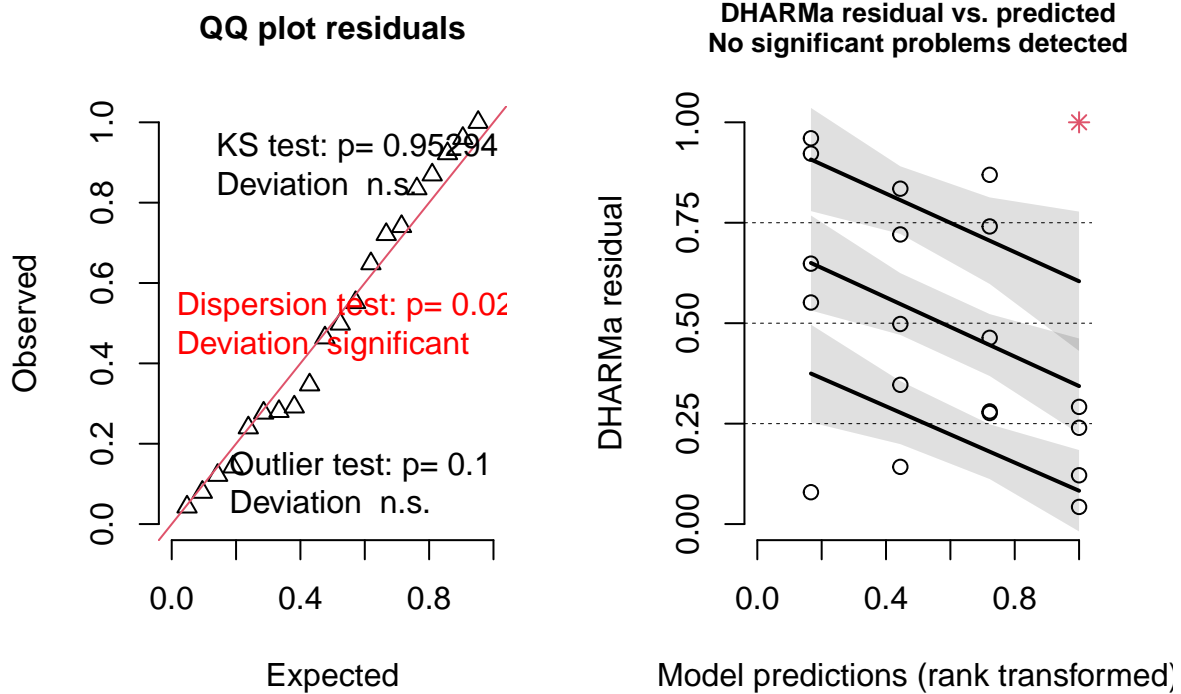
```
##
## DHARMA nonparametric dispersion test via sd of residuals fitted vs.
## simulated
##
## data: simulationOutput
## dispersion = 2.372, p-value = 0.024
## alternative hypothesis: two.sided
```

```
check_zeroinflation(glm.tc) # keine Zero-inflation, Modell ist ok
```

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

```
simulationOutput <- simulateResiduals(fittedModel = glm.tc, plot = T)
```

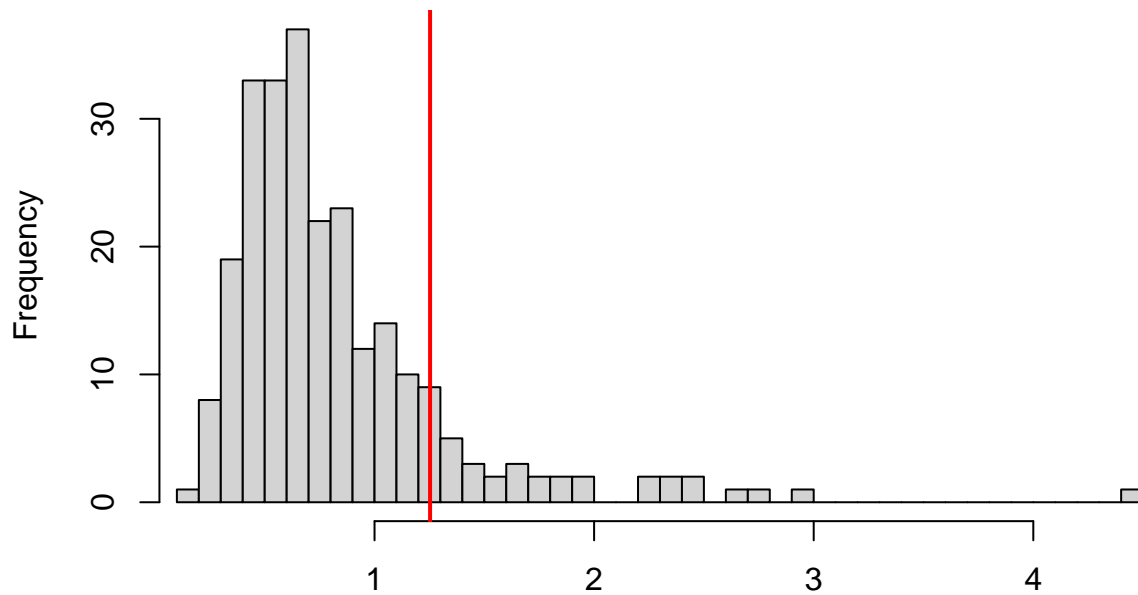
DHARMA residual



Für den Teichmolch ist das Modell unauffällig, für den Kammmolch gibt es jedoch eine signifikante Abweichung hinsichtlich der Verteilung ($p = 0,024$). Daher nutzen wir alternativ zur Poisson-Verteilung die Negativbinomialverteilung.

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

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



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

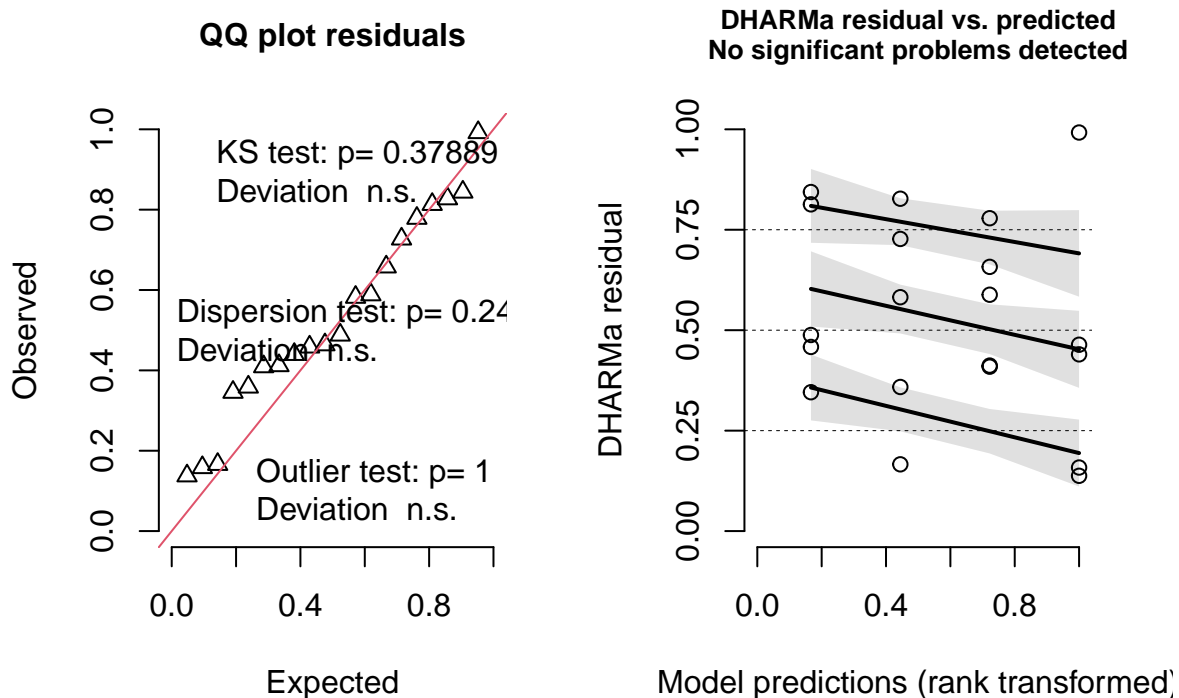
```
##
## DHARMA nonparametric dispersion test via sd of residuals fitted vs.
## simulated
##
## data: simulationOutput
## dispersion = 1.5106, p-value = 0.248
## alternative hypothesis: two.sided
```

```
check_zeroinflation(glm.tc) # keine Zero-inflation, Modell ist ok
```

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

```
simulationOutput <- simulateResiduals(fittedModel = glm.tc, plot = T)
```

DHARMA residual



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

Schauen wir uns nun die 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 = BAdat)
##
## 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 = BAdat)
##
## 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: 0x000001366aa763f0>
## 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 können wir uns auch mithilfe des emmeans-Paketes anschauen.

```
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
```

Im Treatment-Gewässer waren also auch 2023 mehr Larven als im Control1-Gewässer, jedoch ist der Unterschied 2024 deutlich stärker zu sehen. In beiden Gewässern gibt es 2024 allerdings mehr Larven als 2023, weshalb der mögliche positive Effekt der Pflegemaßnahme maskiert sein könnte.

Graphische Darstellung

```

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

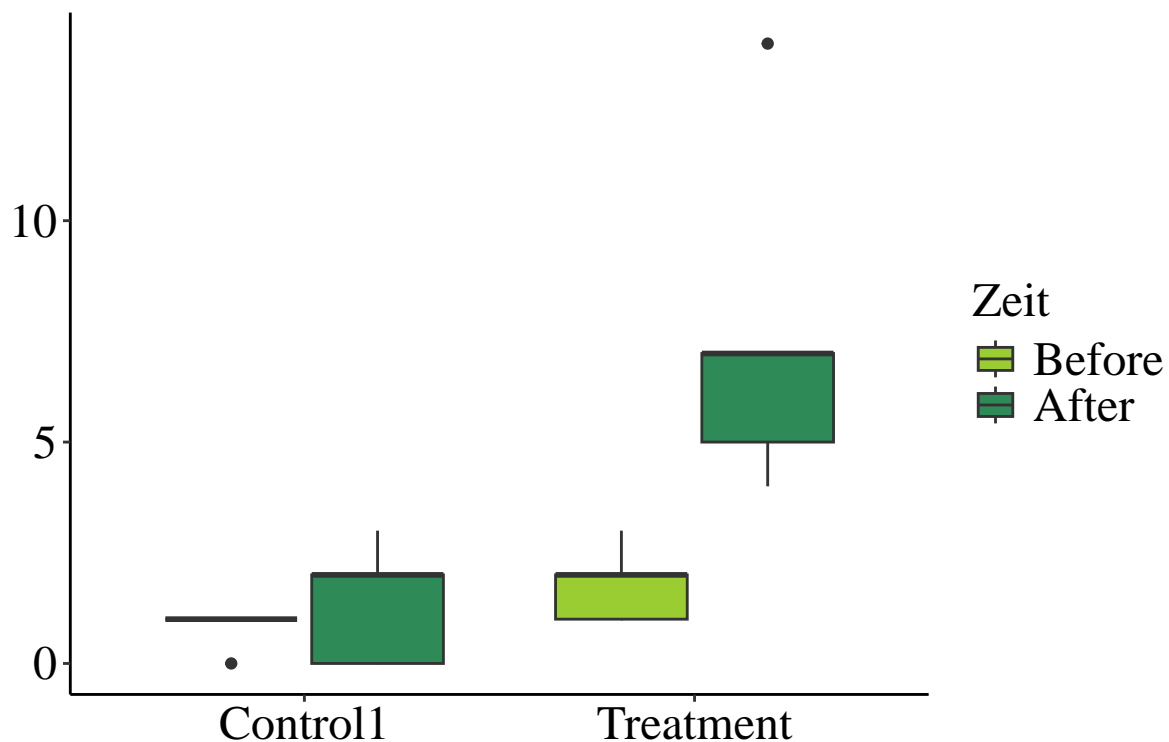
    #legend.key.size = unit(1.5, "cm"),

    # 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 mit den Modellergebnissen. Achtung: in einem Boxplot werden keine Mittelwerte, sondern Mediane dargestellt, weshalb der Zahlenwert nicht mit denen aus emmeans überein stimmen kann.

Kammolch

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 -0.10540  +
## 3  0.09531    +
## 4 -0.10510  +  +
## 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 = BAdat,
##   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 = BAdat, 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 = BAdat, 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: 0x0000013671d7dc80>
## attr("call")
## AICc(x)
## attr("class")
## [1] "function"      "rankFunction"
## attr("beta")
## [1] "none"
```

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

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 Einzelparameter (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 können wir uns auch mithilfe des `emmeans` Paketes anschauen.

```
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
```

Während 2023 im **Treatment**-Gewässer weniger Kammmolchlarven als im Kontrollgewässer waren (mit nur geringem Unterschied), befanden sich nach der Maßnahme im **Treatment**-Gewässer deutlich mehr Kammmolchlarven, im Kontrollgewässer deutlich weniger als im Vorjahr. Die Unterschiede sind jedoch nicht sehr groß.

Graphische Darstellung

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

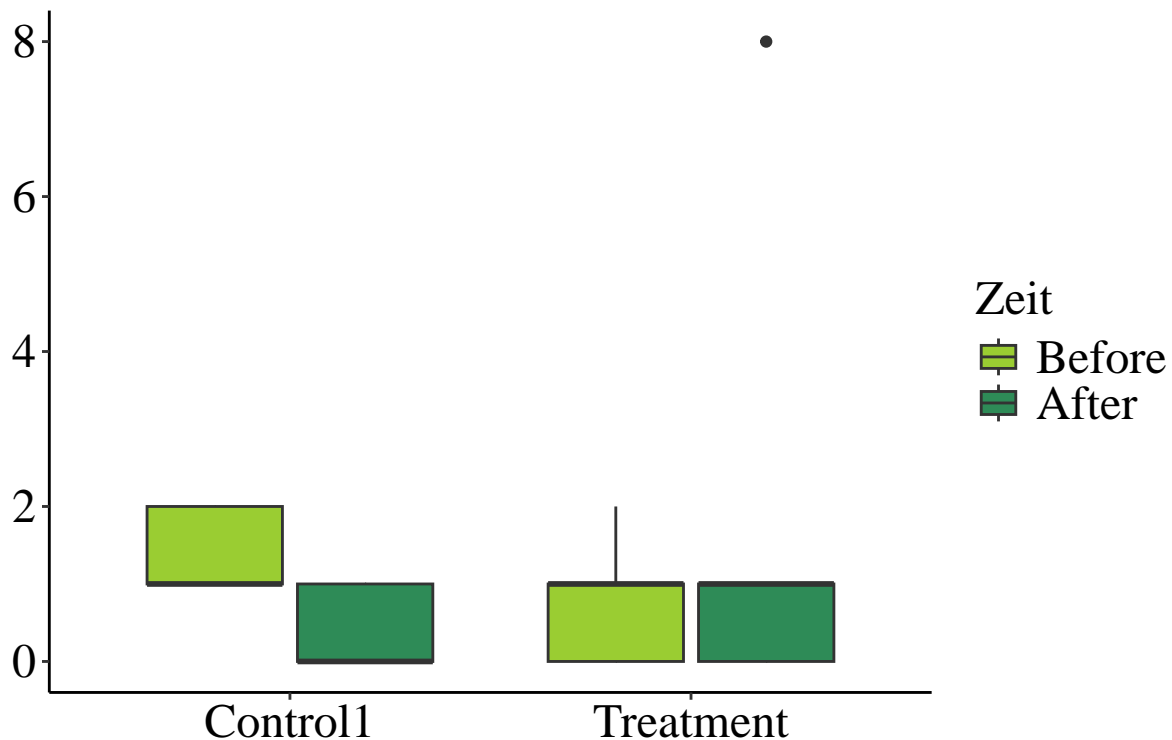
    # Titel
    plot.title = element_text(size = 14, face = "bold", hjust = 0.5,color = "black"),

    #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, nicht die 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 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.