Temperature and Freshwater as Methods for Controlling *D. Vexillum* Biofouling

Payton Arthur, Hazel de Haas, Lauren Gill

27/11/2021

Contents

Abstract

Biofouling, the unwanted establishment of organisms on surfaces, impacts aquaculture facilities by decreasing the value of their products and causing expensive damages to their equipment. The biofouling tunicate *Didemnun vexillum* poses a notable threat to aquaculture given that it is an invasive species with strong competition abilities and is rapidly expanding its range. In this study, we seek to determine the impact of combining high-temperatures and freshwater treatments at different immersion times on *D. vexillum* as a method for controlling biofouling. We immersed *D. vexillum* in either freshwater or seawater at one of four different temperatures (12, 50, 70, and 90°C), for 60 or 120 seconds. We then analyzed the survival of the tunicate 3 weeks after treatment. We found that both 70°C and 90°C treatments successfully killed *D. vexillum* regardless of water type and immersion time. Therefore, to maximize the effectiveness of biofouling removal efforts while limiting the amount of time and energy used, we recommend aquaculture facilities should use 60 seconds 70°C seawater dips to control *D. vexillum* on their gear. Using this method to remove *D. vexillum* biofouling will help to decrease aquaculture gear damage, and reduce the spread of an invasive species.

Libraries

```
library(tidyverse)
library(cowplot)
library(patchwork)
library(ggplot2)
library(here)
library(tidyr)
library(performance)
library(DHARMa)
library(fitdistrplus)
library(gamlss)
library(FSA)
library(goft)
library(MASS)
library(ordinal)
```

Analyses and Graphs

Reading in Data

```
tunidata <- read.csv("tunicate_master.csv")
tunidata$temperature_c <- as.factor(tunidata$temperature_c)
tunidata$water_type <- as.factor(tunidata$water_type)
tunidata$exposure_time_s <- as.factor(tunidata$exposure_time_s)
```

Change in mean RGB values

First Check for normality, p-value = 0.3433 normal distribution!

```
shapiro.test(tunidata$X48hr_rgb)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  tunidata$X48hr_rgb
## W = 0.98249, p-value = 0.3433
```

Using a linear model for normal distribution - allows you to use random effects and nested effects

```
tunirgb <- tunidata %>%
  mutate(exposure_time_s = as.factor(exposure_time_s)) %>%
  mutate(temperature_c = as.factor(temperature_c)) %>%
  mutate(change_rgb = X48hr_rgb-initial_rgb)

modrgb <- lm(X48hr_rgb ~ exposure_time_s + water_type + temperature_c +
               exposure_time_s*water_type*temperature_c +
               (1|colony_id), family = gaussian,
```

```
            data = tunirgb)
summary(modrgb)
```

```
##
## Call:
## lm(formula = X48hr_rgb ~ exposure_time_s + water_type + temperature_c +
##     exposure_time_s * water_type * temperature_c + (1 | colony_id),
##     data = tunirgb, family = gaussian)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -25.420  -9.929   1.285   8.214  35.806
##
## Coefficients: (1 not defined because of singularities)
##                                                    Estimate Std. Error
## (Intercept)                                         151.174      6.417
## exposure_time_s120                                   15.362      9.075
## water_typeseawater                                   -1.261      9.075
## temperature_c50                                       4.087      9.075
## temperature_c70                                      24.380      9.075
## temperature_c90                                      14.826      9.075
## 1 | colony_idTRUE                                       NA         NA
## exposure_time_s120:water_typeseawater               -16.301     12.835
## exposure_time_s120:temperature_c50                   -7.030     12.835
## exposure_time_s120:temperature_c70                  -21.306     12.835
## exposure_time_s120:temperature_c90                  -16.871     12.835
## water_typeseawater:temperature_c50                    2.410     12.835
## water_typeseawater:temperature_c70                   -5.396     12.835
## water_typeseawater:temperature_c90                   -4.666     12.835
## exposure_time_s120:water_typeseawater:temperature_c50   3.402     18.151
## exposure_time_s120:water_typeseawater:temperature_c70  10.765     18.151
## exposure_time_s120:water_typeseawater:temperature_c90  26.347     18.151
##                                                    t value Pr(>|t|)
## (Intercept)                                         23.557  < 2e-16 ***
## exposure_time_s120                                   1.693  0.09537 .
## water_typeseawater                                  -0.139  0.88993
## temperature_c50                                      0.450  0.65402
## temperature_c70                                      2.686  0.00919 **
## temperature_c90                                      1.634  0.10723
## 1 | colony_idTRUE                                       NA       NA
## exposure_time_s120:water_typeseawater               -1.270  0.20864
## exposure_time_s120:temperature_c50                  -0.548  0.58580
## exposure_time_s120:temperature_c70                  -1.660  0.10180
## exposure_time_s120:temperature_c90                  -1.314  0.19337
## water_typeseawater:temperature_c50                   0.188  0.85164
## water_typeseawater:temperature_c70                  -0.420  0.67556
## water_typeseawater:temperature_c90                  -0.364  0.71739
## exposure_time_s120:water_typeseawater:temperature_c50  0.187  0.85191
## exposure_time_s120:water_typeseawater:temperature_c70  0.593  0.55522
## exposure_time_s120:water_typeseawater:temperature_c90  1.452  0.15150
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

```
## Residual standard error: 14.35 on 64 degrees of freedom
## Multiple R-squared:  0.2743, Adjusted R-squared:  0.1042
## F-statistic: 1.613 on 15 and 64 DF,  p-value: 0.09516
```

Since model is not influenced by random effects, take this out of the model (stepAIC does not work with random effects) and then reduce model

```
modrgb <- lm(X48hr_rgb ~ exposure_time_s + water_type + temperature_c + exposure_time_s*water_type*tempe
stepmodrgb <- stepAIC(modrgb, direction = "backward", trace = F)
formula(stepmodrgb)
```

```
## X48hr_rgb ~ water_type + temperature_c
```

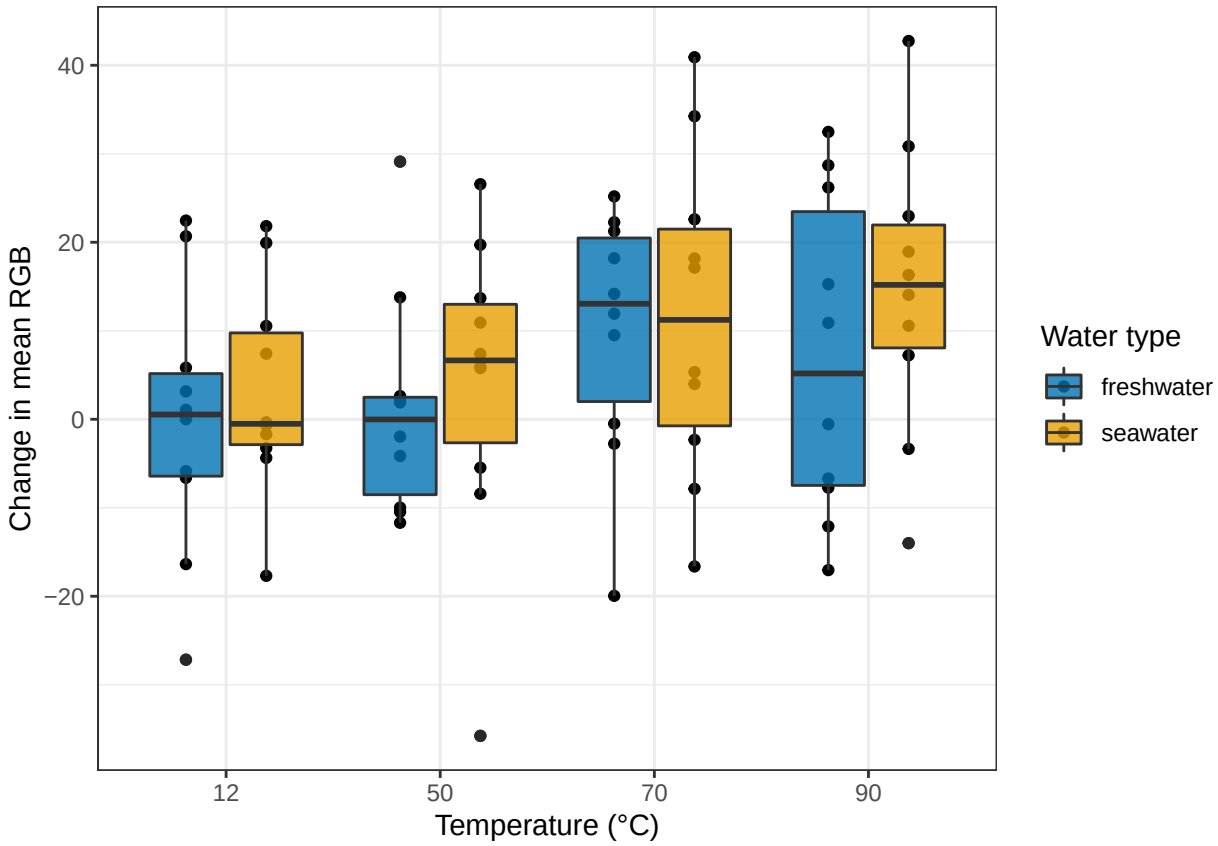
stepAIC has outputted its final, reduced model. Final p values for model output

```
newmodrgb <- lm(X48hr_rgb ~ water_type + temperature_c, data = tunirgb)
summary(newmodrgb)
```

```
##
## Call:
## lm(formula = X48hr_rgb ~ water_type + temperature_c, data = tunirgb)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -29.518  -9.107   0.730   8.664  36.323
##
## Coefficients:
##                    Estimate Std. Error t value Pr(>|t|)
## (Intercept)         157.279      3.518  44.710  < 2e-16 ***
## water_typeseawater   -6.260      3.146  -1.990  0.05026 .
## temperature_c50       2.627      4.450   0.590  0.55664
## temperature_c70      13.720      4.450   3.083  0.00286 **
## temperature_c90      10.645      4.450   2.392  0.01925 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 14.07 on 75 degrees of freedom
## Multiple R-squared:  0.1823, Adjusted R-squared:  0.1387
## F-statistic:  4.18 on 4 and 75 DF,  p-value: 0.004138
```

Graph

```
deltachangergb <- ggplot(tunirgb, aes(x = temperature_c, y = change_rgb,
                                      fill = water_type))+
  labs(x = "Temperature (°C)", y = "Change in mean RGB", fill = "Water type")+
#labs(title = "Temperature (°C)")+
  scale_fill_manual(values = c("#0072B2","#E69F00"))+
  geom_jitter(position = position_dodge(width=0.75))+
  geom_boxplot(alpha = 0.8)+
  theme_bw()
deltachangergb
```




```
#ggsave("delta-change-rbg.jpg")
```

Mold Cover

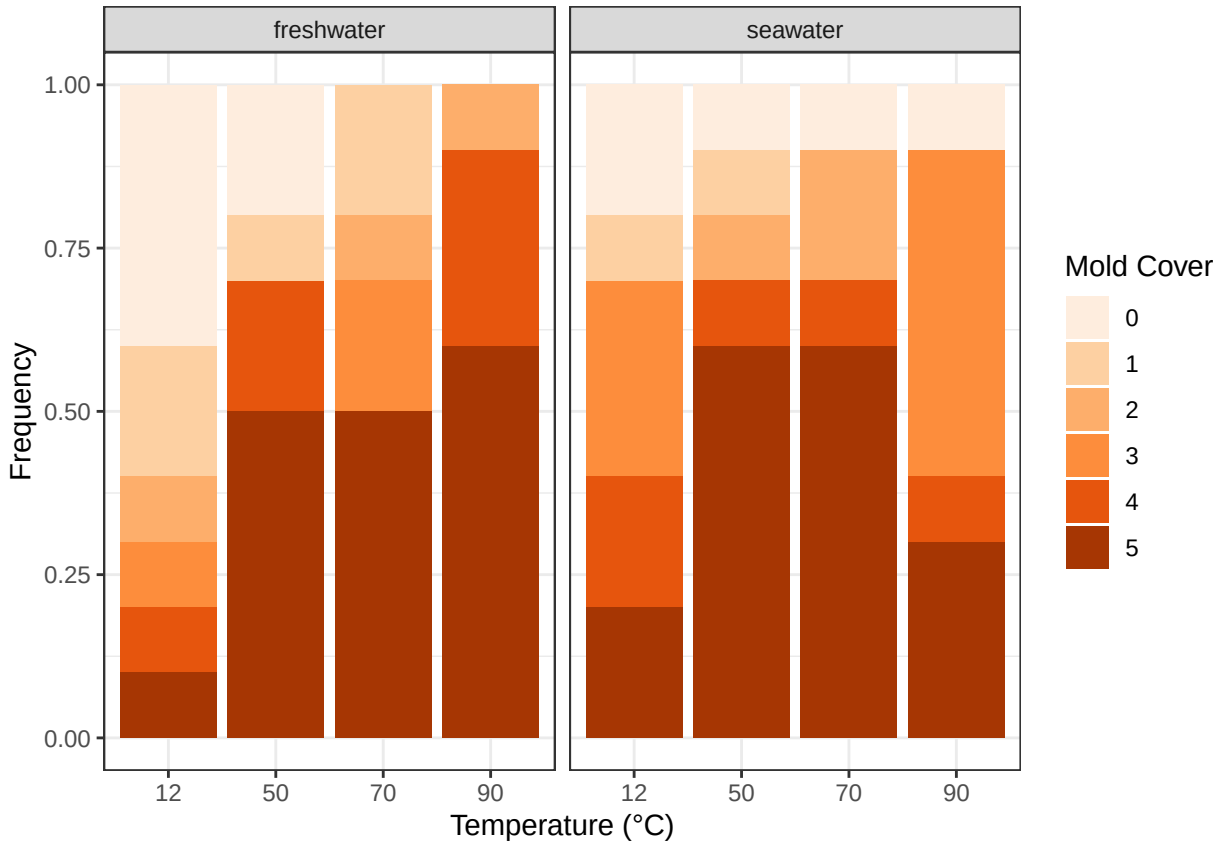
Displaying the data as a proportion of overall score with given mold cover scores

```
tuni_stacked = tunidata %>%
  group_by(mold_cover, temperature_c, water_type, exposure_time_s) %>%
  summarise(frequency = n())%>%
  mutate(temperature_c=  as.factor(temperature_c))
```

```
## `summarise()` has grouped output by 'mold_cover', 'temperature_c', 'water_type'. You can override usi
```

Stacked bar graphs showing mold cover

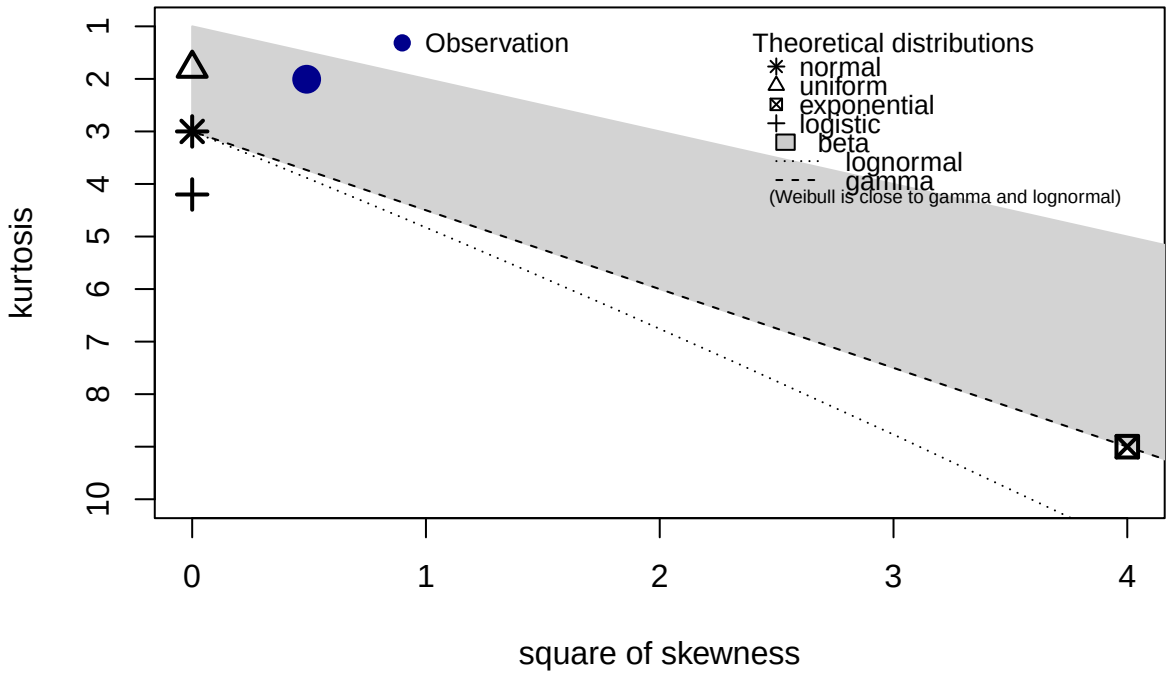
```
ggplot(tuni_stacked, aes(y = frequency, x = temperature_c,
                         fill = as.factor(mold_cover))) +
  geom_bar(stat = "identity", position = "fill") +
  facet_grid(.~water_type) +
  labs(x ="Temperature (°C)", y = "Frequency")+
  scale_fill_brewer(palette = "Oranges", direction=1) +
  labs(fill = "Mold Cover") +
  theme_bw()
```




```
#ggsave("mold-cover.jpg")
```

Examining mold cover data to find the best distribution

```
descdist(tunidata$mold_cover)
```




```
## summary statistics
## ------
## min:  0   max:  5
## median:  4
## mean:  3.325
## estimated sd:  1.854007
## estimated skewness:  -0.6997944
## estimated kurtosis:  2.006652
```

```
fit <- fitDist(mold_cover, data = tunidata, type = "realAll", try.gamlss = T)
```

```
##   |                                                                      |
##   Lapack routine dgesv: system is exactly singular: U[3,3] = 0
##   |                                                                      |==========
##   Lapack routine dgesv: system is exactly singular: U[4,4] = 0
##   |                                                                      |==========
##   Lapack routine dgesv: system is exactly singular: U[3,3] = 0
##   |                                                                      |==========
```

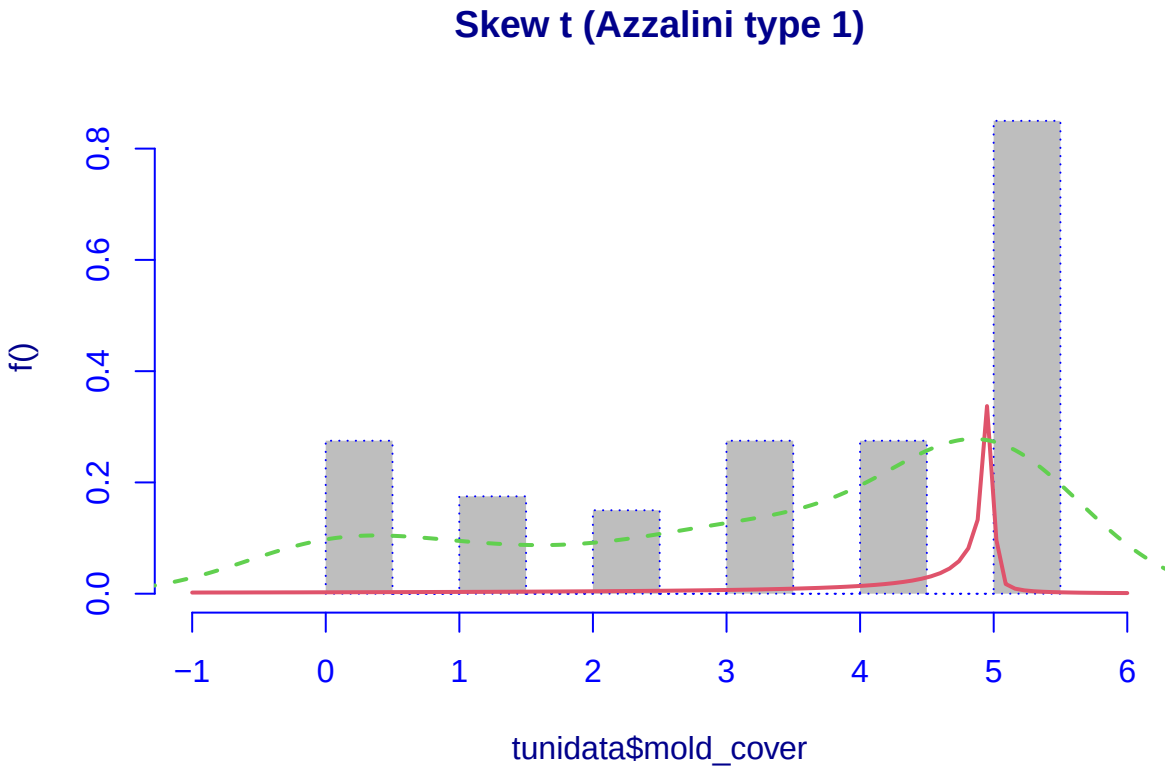
```
fit$fits
```

```
##        SHASHo       SHASHo2           ST1           JSUo           JSU
## -8.198516e+02 -7.527084e+02 -6.470003e+02 -3.786701e+02  9.443648e+00
##           ST5           ST3           SN2           SST           ST2
##  1.033464e+01  2.521045e+02  2.681226e+02  2.698089e+02  2.700913e+02
##            PE          SEP4           NET           PE2          EGB2
##  2.712465e+02  2.713103e+02  2.730814e+02  2.835127e+02  2.895608e+02
```

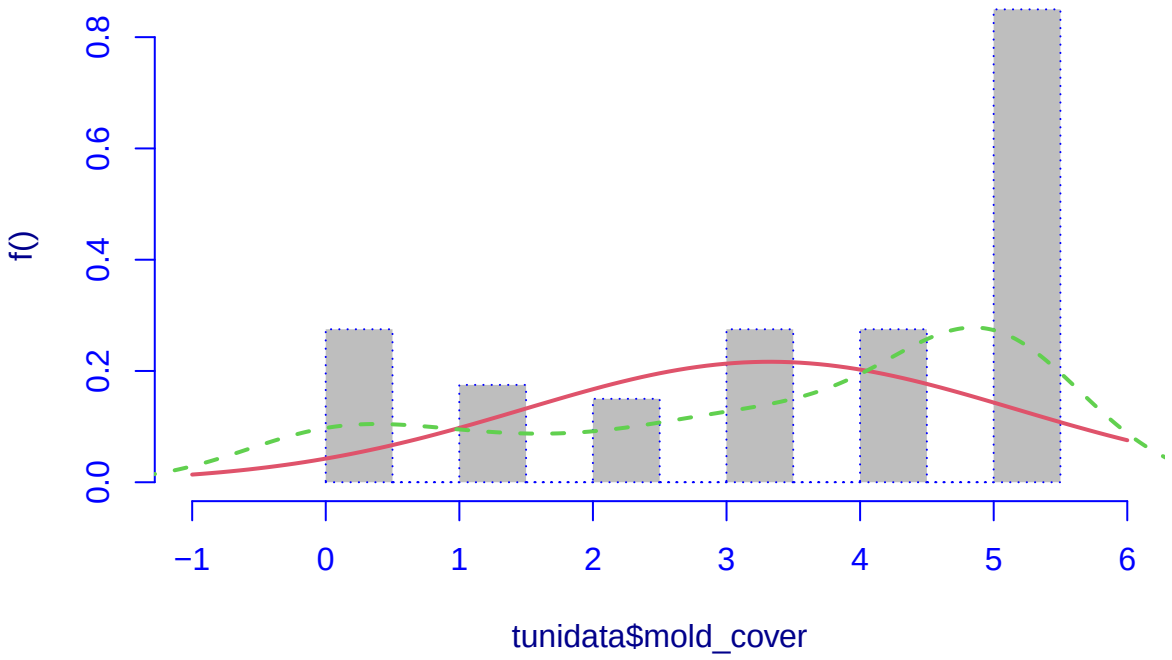
```
##            GU            NO           TF2            TF           SN1
##  3.112873e+02  3.287997e+02  3.307997e+02  3.307997e+02  3.307997e+02
##        exGAUS           ST4            LO            RG           EXP
##  3.308045e+02  3.327997e+02  3.343529e+02  3.492131e+02  3.542351e+02
##       PARETO2      PARETO2o          SEP2          SEP1            GT
##  3.562351e+02  3.562352e+02  3.452138e+04  1.185033e+05  6.144562e+08
```

Fitdist determined ST1 to be the best distribution, comparing it against a normal distribution to make sure AIC value is lower. AIC value of ST1 = -743.8 while AIC value of normal = 328

```
mST1 <- histDist(tunidata$mold_cover, "ST1", density = T,
                 main = "Skew t (Azzalini type 1)")
```




```
mNO <- histDist(tunidata$mold_cover, "NO", density = T, main = "Normal")
```




```
GAIC(mST1, mNO)
```

```
##      df       AIC
## mST1  4 -647.0003
## mNO   2  328.7997
```

Now creating a full model, reduced model is the same as full model. No factors are significant

```
tunirgb$colony_id <- as.factor(tunirgb$colony_id)
tunirgb$mold_cover <- as.factor(tunirgb$mold_cover)

ordinalmodfull <- clmm2(mold_cover ~ water_type + temperature_c + exposure_time_s +
                      exposure_time_s*water_type*temperature_c,
                      random = colony_id, data = tunirgb, Hess = TRUE)
summary(ordinalmodfull)
```

```
## Cumulative Link Mixed Model fitted with the Laplace approximation
##
## Call:
## clmm2(location = mold_cover ~ water_type + temperature_c + exposure_time_s +
##     exposure_time_s * water_type * temperature_c, random = colony_id,
##     data = tunirgb, Hess = TRUE)
##
## Random effects:
##                   Var   Std.Dev
## colony_id 0.08633119 0.2938217
##
## Location coefficients:
```

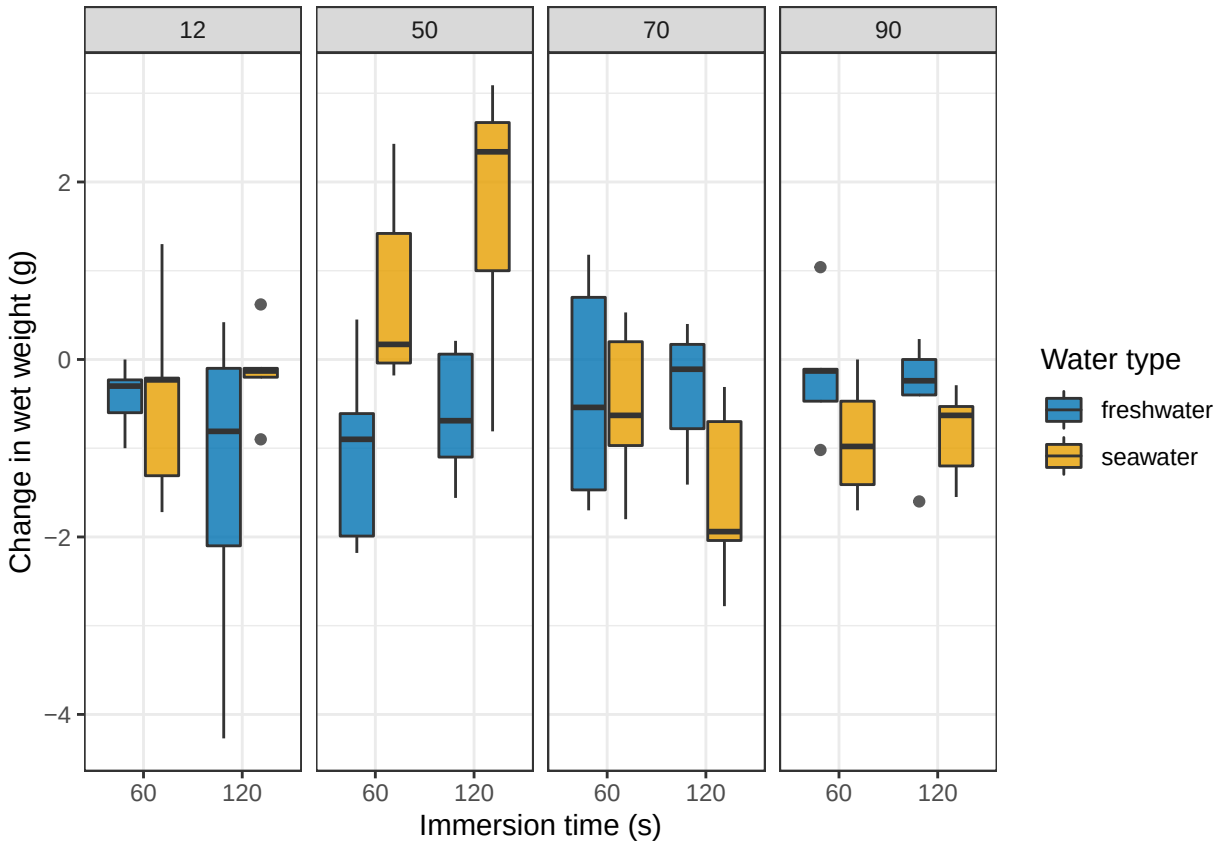
```
##                                                    Estimate Std. Error
## water_typeseawater                                  1.4322   1.0358
## temperature_c50                                     1.5210   1.0377
## temperature_c70                                     0.7898   0.5905
## temperature_c90                                     2.4114   1.1833
## exposure_time_s120                                 -1.3967   1.1751
## water_typeseawater:exposure_time_s120              -0.2186   1.5864
## temperature_c50:exposure_time_s120                  1.0243   1.7631
## temperature_c70:exposure_time_s120                  2.4574   1.4489
## temperature_c90:exposure_time_s120                  1.0751   1.7042
## water_typeseawater:temperature_c50                 -1.4843   1.5749
## water_typeseawater:temperature_c70                 -1.9368   1.0619
## water_typeseawater:temperature_c90                 -2.5154   1.6410
## water_typeseawater:temperature_c50:exposure_time_s120  1.0916   2.4101
## water_typeseawater:temperature_c70:exposure_time_s120 23.2430      NaN
## water_typeseawater:temperature_c90:exposure_time_s120 -0.1900   2.3047
##                                                    z value Pr(>|z|)
## water_typeseawater                                  1.3827 0.166767
## temperature_c50                                     1.4657 0.142722
## temperature_c70                                     1.3376 0.181027
## temperature_c90                                     2.0378 0.041566
## exposure_time_s120                                 -1.1886 0.234606
## water_typeseawater:exposure_time_s120              -0.1378 0.890380
## temperature_c50:exposure_time_s120                  0.5810 0.561255
## temperature_c70:exposure_time_s120                  1.6960 0.089881
## temperature_c90:exposure_time_s120                  0.6308 0.528142
## water_typeseawater:temperature_c50                 -0.9424 0.345968
## water_typeseawater:temperature_c70                 -1.8239 0.068175
## water_typeseawater:temperature_c90                 -1.5328 0.125328
## water_typeseawater:temperature_c50:exposure_time_s120  0.4529 0.650610
## water_typeseawater:temperature_c70:exposure_time_s120     NaN NA
## water_typeseawater:temperature_c90:exposure_time_s120 -0.0825 0.934289
##
## No scale coefficients
##
## Threshold coefficients:
##     Estimate Std. Error z value
## 0|1 -1.0939   0.7416    -1.4752
## 1|2 -0.3827   0.7253    -0.5277
## 2|3  0.0931   0.7269     0.1281
## 3|4  0.8816   0.7201     1.2242
## 4|5  1.6307   0.7279     2.2404
##
## log-likelihood: -112.6399
## AIC: 267.2799
## Condition number of Hessian: 3.058746e+12
```

```
#temperature_c90 p=0.041566
```

Change in weight after 48 hours

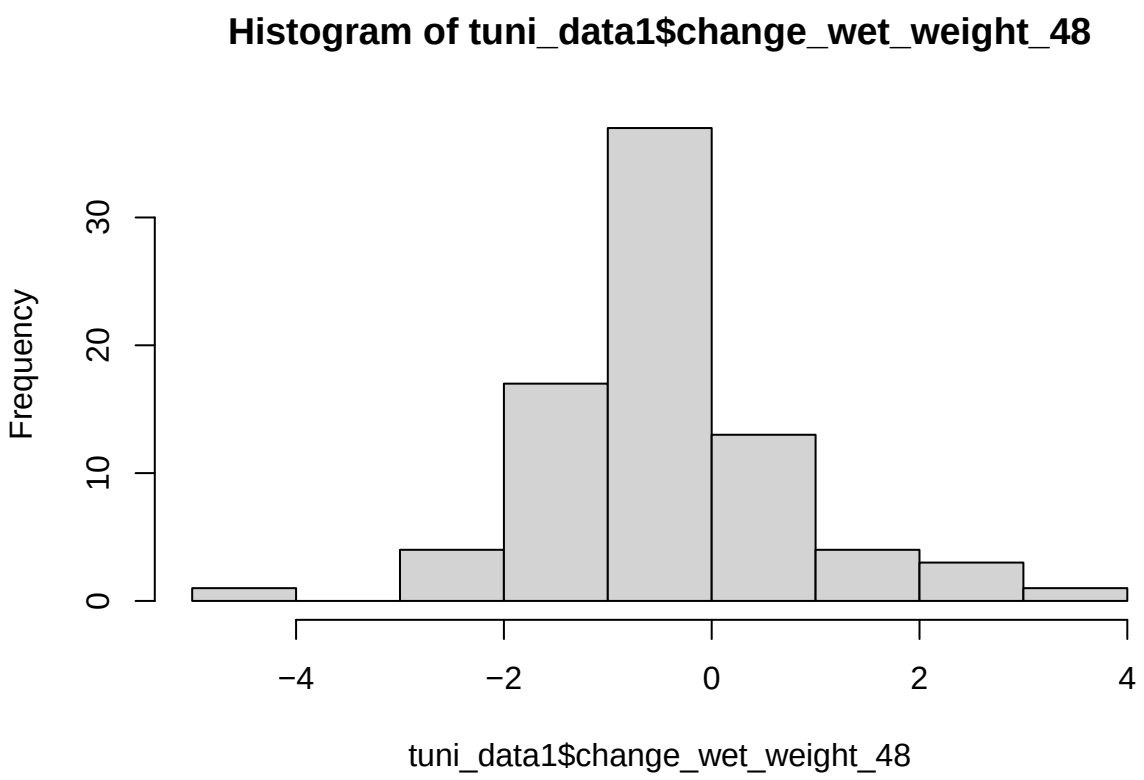
Data Visualization

```
#Making new columns for changes in weight
tuni_data1 <- tunidata %>%
  mutate(change_wet_weight_3wk = final_weight_g-post_acclimation_weight_g)
tuni_data1 <- tuni_data1 %>%
  mutate(change_wet_weight_48 = `X48hr_weight`-post_acclimation_weight_g)

p2 <- ggplot(tuni_data1, aes(x = exposure_time_s, y = change_wet_weight_48,
                             fill = water_type, ))+
  labs(x = "Immersion time (s)", y = "Change in wet weight (g)",
       fill = "Water type")+
  geom_boxplot(alpha = 0.8)+
  scale_fill_manual(values = c("#0072B2","#E69F00"))+
  theme_bw()

p3 <- p2 + facet_grid(cols = vars(temperature_c))
p3
```



Testing for Normality - p-value = 0.01824 so not a normal distribution

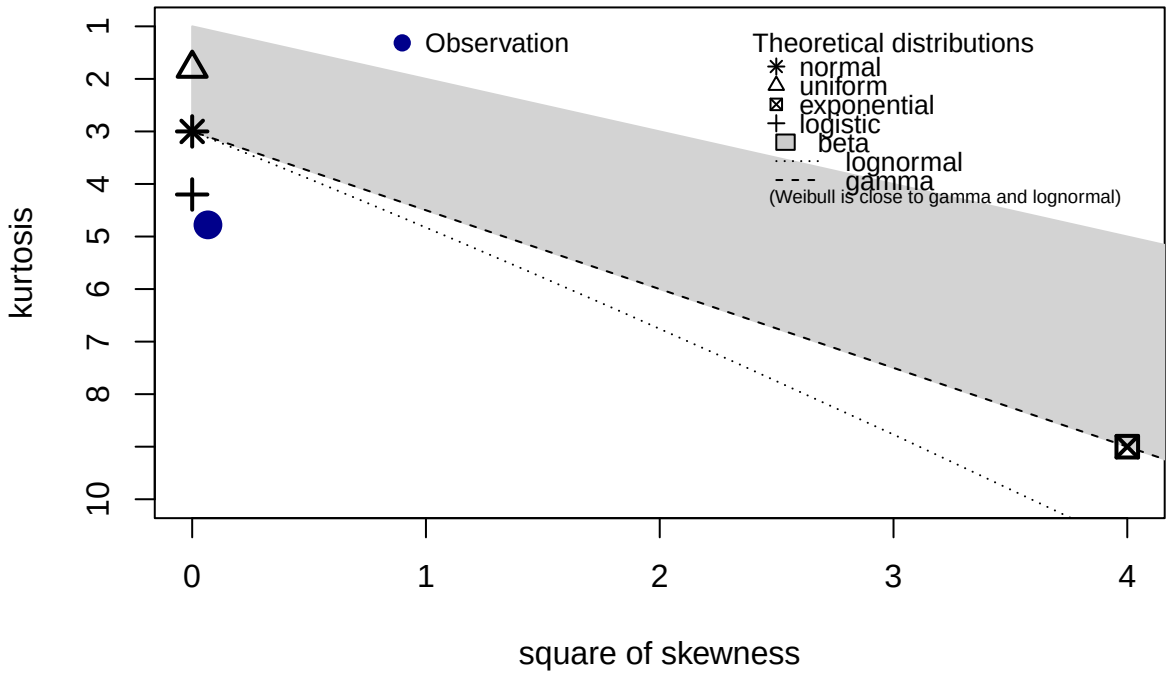
```
hist(tuni_data1$change_wet_weight_48)
```




```
shapiro.test(tuni_data1$change_wet_weight_48)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  tuni_data1$change_wet_weight_48
## W = 0.96216, p-value = 0.01824
```

```
#p-value = 0.01824, not normal
descdist(tuni_data1$change_wet_weight_48)
```




```
## summary statistics
## ------
## min:  -4.27   max:  3.09
## median:  -0.355
## mean:  -0.419375
## estimated sd:  1.189221
## estimated skewness:  0.2597728
## estimated kurtosis:  4.778191
```

```
#might be logistic
```

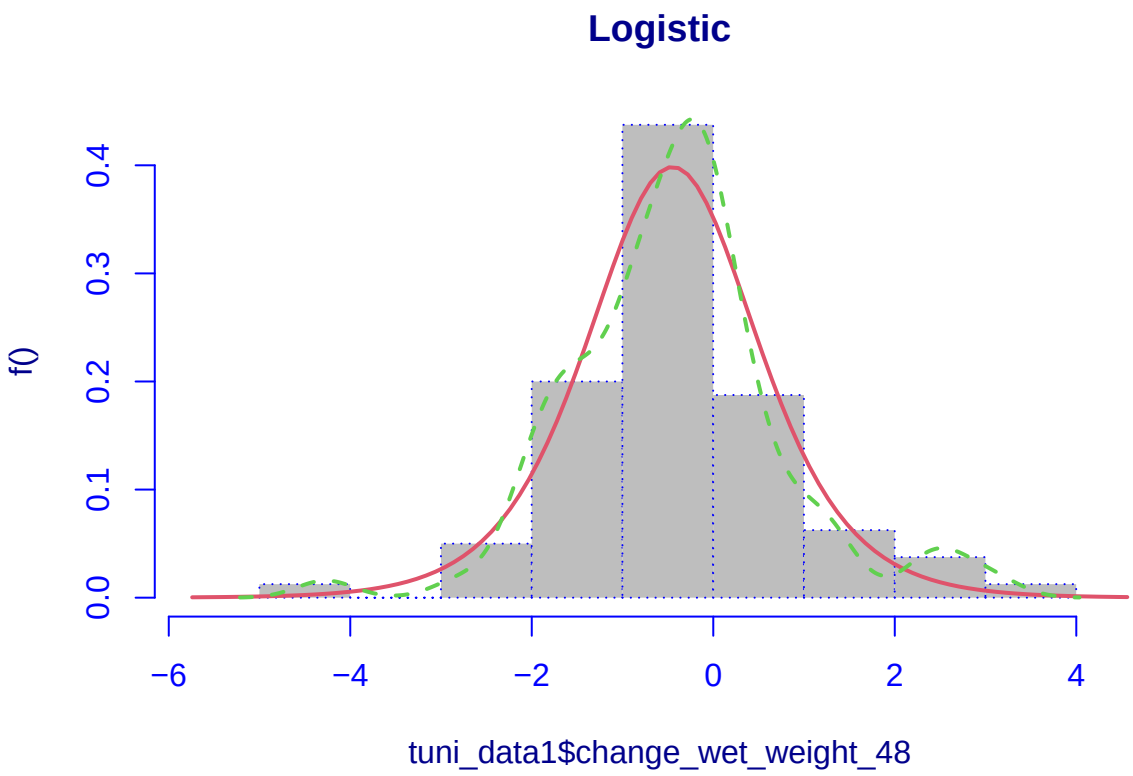
Distribution Fitting - used FitDist function - followed a logistic distribution

```
fitDist(change_wet_weight_48, data=tuni_data1, type="realAll", try.gamlss = T)
```

```
##   |                                                                      |
```

```
##
##  Degrees of Freedom for the fit: 2 Residual Deg. of Freedom   78
## Global Deviance:     247.968
##              AIC:     251.968
##              SBC:     256.732
```

```
#family=LO, logistic

mLOG_weight <- histDist(tuni_data1$change_wet_weight_48, "LO", density = T,
                        main = "Logistic")
```




```
GAIC(mLOG_weight)
```

```
## [1] 251.9677
```

GAMLSS Model

```
mod_weight <- gamlss(change_wet_weight_48 ~ water_type + temperature_c +
                     exposure_time_s + water_type*temperature_c*exposure_time_s,
                      family = LO, data = tuni_data1)
```

Model Selection - final formula for change

```
stepmodweight48 <- stepGAIC(mod_weight, direction = "backward", trace = F)
```

```
## Start:  AIC= 246.79
##  change_wet_weight_48 ~ water_type + temperature_c + exposure_time_s +
##      water_type * temperature_c * exposure_time_s
```

```
summary(stepmodweight48)
```

```
## ******************************************************************
## Family:  c("LO", "Logistic")
##
## Call:  gamlss(formula = change_wet_weight_48 ~ water_type +
##     temperature_c + water_type:temperature_c, family = LO,
##     data = tuni_data1, trace = FALSE)
##
## Fitting method: RS()
##
## ------------------------------------------------------------------
## Mu link function:  identity
## Mu Coefficients:
##                                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)                        -0.6259     0.2951  -2.121  0.03741 *
## water_typeseawater                  0.3364     0.4134   0.814  0.41849
## temperature_c50                    -0.1969     0.4225  -0.466  0.64260
## temperature_c70                     0.2711     0.4315   0.628  0.53192
## temperature_c90                     0.3656     0.4030   0.907  0.36745
## water_typeseawater:temperature_c50  1.7036     0.6363   2.677  0.00922 **
## water_typeseawater:temperature_c70 -1.0045     0.6109  -1.644  0.10453
## water_typeseawater:temperature_c90 -0.9499     0.5662  -1.678  0.09778 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## ------------------------------------------------------------------
## Sigma link function:  log
## Sigma Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.60404    0.09281  -6.509 9.23e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## ------------------------------------------------------------------
## No. of observations in the fit:  80
## Degrees of Freedom for the fit:  9
##       Residual Deg. of Freedom:  71
##                       at cycle:  3
##
## Global Deviance:     221.8059
##             AIC:     239.8059
##             SBC:     261.2441
## ******************************************************************
```

```
#water_typeseawater:temperature_c50 p-value = 0.00922 significant

formula(stepmodweight48)
```

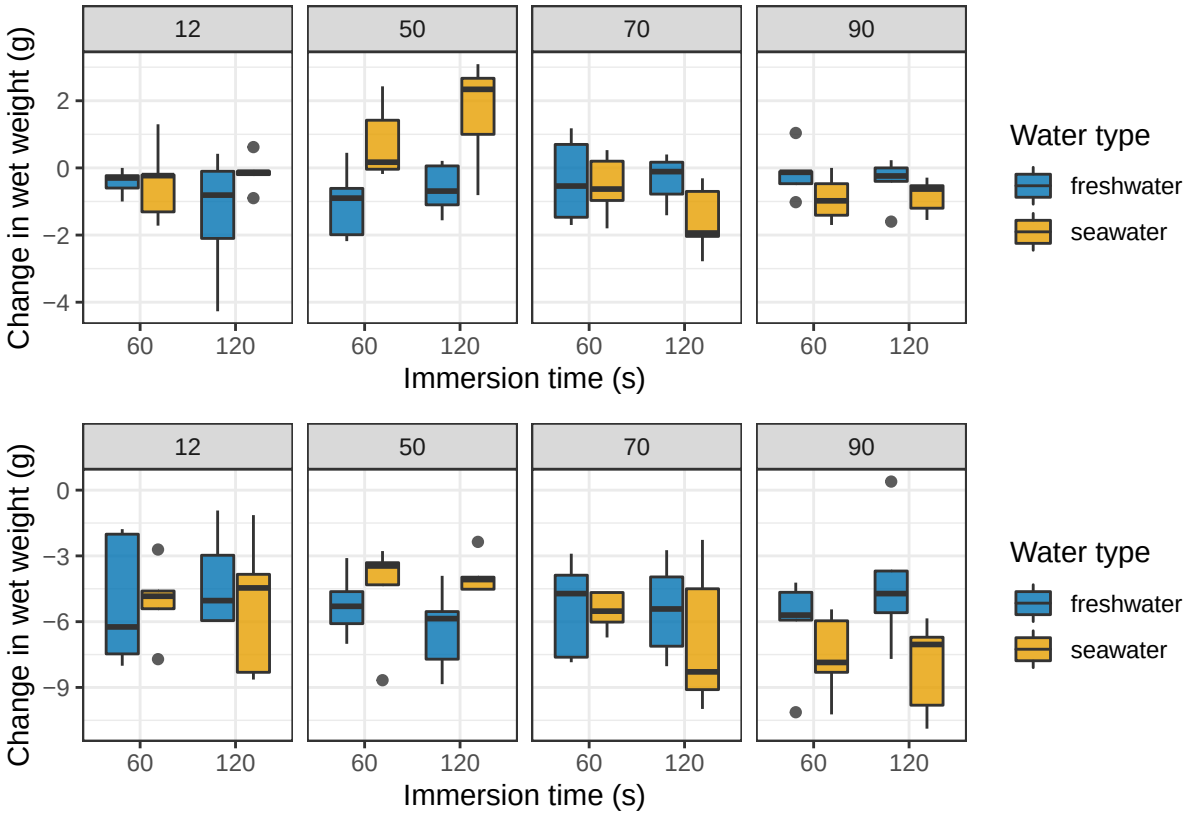
```
## change_wet_weight_48 ~ water_type + temperature_c + water_type:temperature_c
```

```
#change_wet_weight_48 ~ water_type + temperature_c + water_type:temperature_c
```

Change in Weight over 3 Weeks (compare to post-acclimation)

Data Visualization

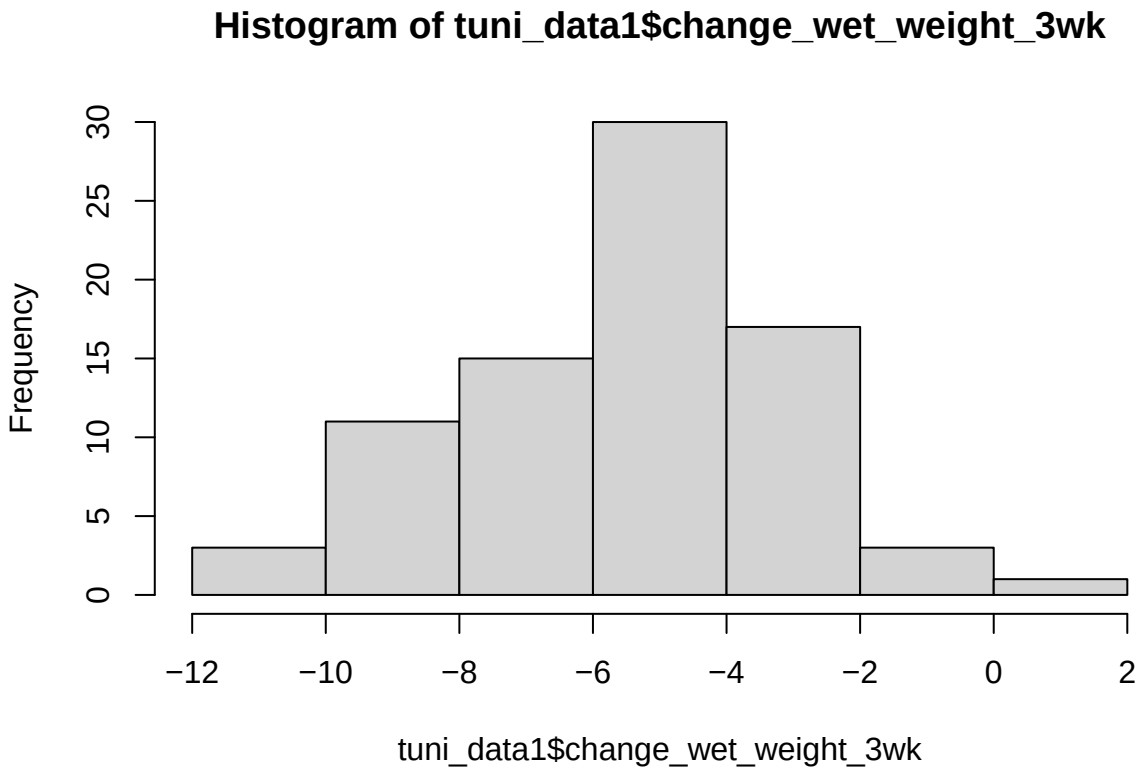
```
p <- ggplot(tuni_data1, aes(x = exposure_time_s, y = change_wet_weight_3wk,
                            fill = water_type))+
  labs(x = "Immersion time (s)", y = "Change in wet weight (g)",
       fill = "Water type")+
#labs(title = "Temperature (°C)")+
  geom_boxplot(alpha = 0.8)+
  scale_fill_manual(values = c("#0072B2","#E69F00"))+
  theme_bw()

p4 <- p + facet_grid(cols = vars(temperature_c))

p3 + p4 + plot_layout(ncol=1)
```




```
#ggsave("change-in-wet-weight.jpg")
```

Testing for Normality

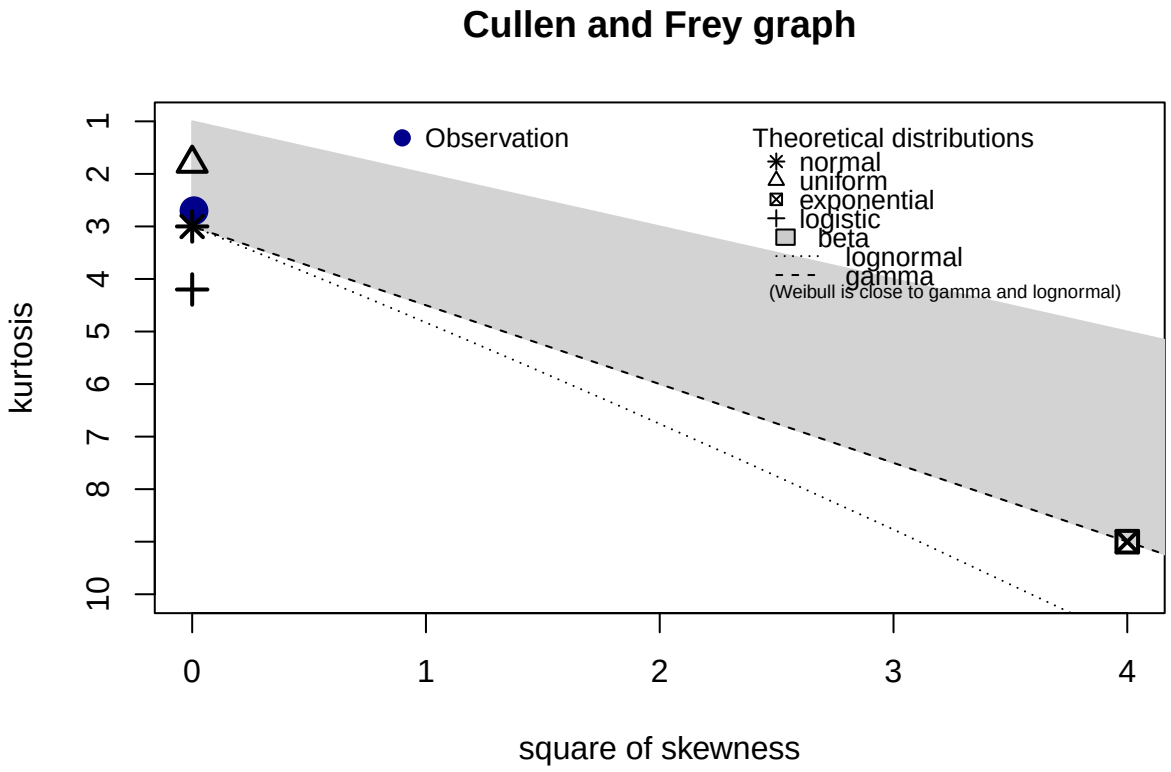
```
hist(tuni_data1$change_wet_weight_3wk)
```




```
shapiro.test(tuni_data1$change_wet_weight_3wk)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  tuni_data1$change_wet_weight_3wk
## W = 0.98952, p-value = 0.7646
```

```
#p-value = 0.7646 thus follows a normal distribution!
descdist(tuni_data1$change_wet_weight_3wk)
```




```
## summary statistics
## ------
## min:  -10.88   max:  0.39
## median:  -5.43
## mean:  -5.55025
## estimated sd:  2.36077
## estimated skewness:  -0.08817917
## estimated kurtosis:  2.700809
```

Distribution Fitting

```
fitDist(change_wet_weight_3wk, data = tuni_data1, type = "realline", try.gamlss = T)
```

```
##   |                                                                      |
##   Lapack routine dgesv: system is exactly singular: U[3,3] = 0
##   |                                                                      |=================
##   Lapack routine dgesv: system is exactly singular: U[3,3] = 0
##   |                                                                      |==================
##   Lapack routine dgesv: system is exactly singular: U[4,4] = 0
##   |                                                                      |==================
##   Lapack routine dgesv: system is exactly singular: U[4,4] = 0
##   |                                                                      |==================
##   Lapack routine dgesv: system is exactly singular: U[4,4] = 0
##   |                                                                      |==================
##   Lapack routine dgesv: system is exactly singular: U[3,3] = 0
##   |                                                                      |==================
##   Lapack routine dgesv: system is exactly singular: U[3,3] = 0
```

```
##
## Family:  c("NO", "Normal")
## Fitting method: "nlminb"
##
## Call:  gamlssML(formula = y, family = NO)
##
## Mu Coefficients:
## [1]  -5.55
## Sigma Coefficients:
## [1]  0.8527
##
##  Degrees of Freedom for the fit: 2 Residual Deg. of Freedom   78
## Global Deviance:     363.462
##             AIC:     367.462
##             SBC:     372.226
```

```
#this also gives normal distribution
```

Linear Model

```
mod_changeweight <- lm(change_wet_weight_3wk ~
                         exposure_time_s + water_type +
                         temperature_c+
                         exposure_time_s*water_type*temperature_c,
                       family = gaussian, data = tuni_data1)
```

Model Selection

```
#Model Selection
step.mod_changeweight <- stepAIC(mod_changeweight, direction = "backward",
                                 trace = F)
summary(step.mod_changeweight)
```

```
##
## Call:
## lm(formula = change_wet_weight_3wk ~ water_type + temperature_c +
##     water_type:temperature_c, data = tuni_data1, family = gaussian)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.9350 -1.6263  0.1905  1.5020  5.5850
##
## Coefficients:
##                                    Estimate Std. Error t value Pr(>|t|)
## (Intercept)                         -4.6370     0.7025  -6.601 5.97e-09 ***
## water_typeseawater                  -0.5290     0.9935  -0.532    0.596
## temperature_c50                     -1.1630     0.9935  -1.171    0.246
## temperature_c70                     -0.7870     0.9935  -0.792    0.431
## temperature_c90                     -0.5580     0.9935  -0.562    0.576
## water_typeseawater:temperature_c50   2.1290     1.4050   1.515    0.134
## water_typeseawater:temperature_c70  -0.2180     1.4050  -0.155    0.877
## water_typeseawater:temperature_c90  -2.0850     1.4050  -1.484    0.142
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.221 on 72 degrees of freedom
## Multiple R-squared:  0.193,  Adjusted R-squared:  0.1145
## F-statistic:  2.46 on 7 and 72 DF,  p-value: 0.0255
```

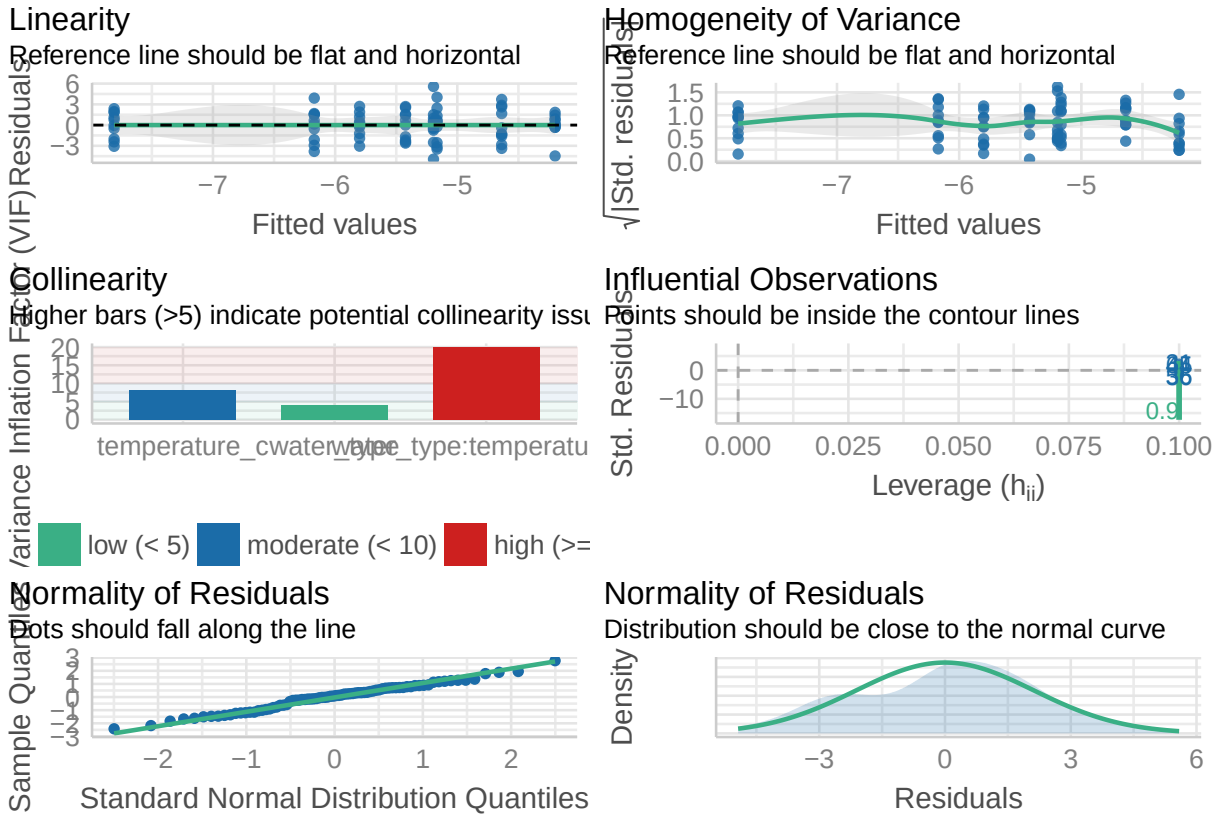
```
formula(step.mod_changeweight)
```

```
## change_wet_weight_3wk ~ water_type + temperature_c + water_type:temperature_c
```

```
#change_wet_weight_3wk ~ water_type + temperature_c + water_type:temperature_c
```

Check Model

```
check_model(step.mod_changeweight)
```




```
#check model when water_type:temperature_c was present and had major
#collinearity issues, thus removed water_type:temperature and check model
#again and collinearity issues were solved
```

New Model

```
mod_changeweightv2 <- lm(change_wet_weight_3wk ~ water_type + temperature_c,
                         family = gaussian, data = tuni_data1)
summary(mod_changeweightv2)
```

```
##
## Call:
## lm(formula = change_wet_weight_3wk ~ water_type + temperature_c,
##     data = tuni_data1, family = gaussian)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.1362 -1.6128  0.4318  1.5524  6.6058
##
## Coefficients:
##                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)          -4.6152     0.5773  -7.995 1.22e-11 ***
## water_typeseawater   -0.5725     0.5163  -1.109   0.2711
## temperature_c50      -0.0985     0.7302  -0.135   0.8931
## temperature_c70      -0.8960     0.7302  -1.227   0.2237
## temperature_c90      -1.6005     0.7302  -2.192   0.0315 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.309 on 75 degrees of freedom
## Multiple R-squared:  0.09168,    Adjusted R-squared:  0.04324
## F-statistic: 1.893 on 4 and 75 DF,  p-value: 0.1205
```

```
#significant p-value for temperature_c90 (p=value = 0.0315)
```

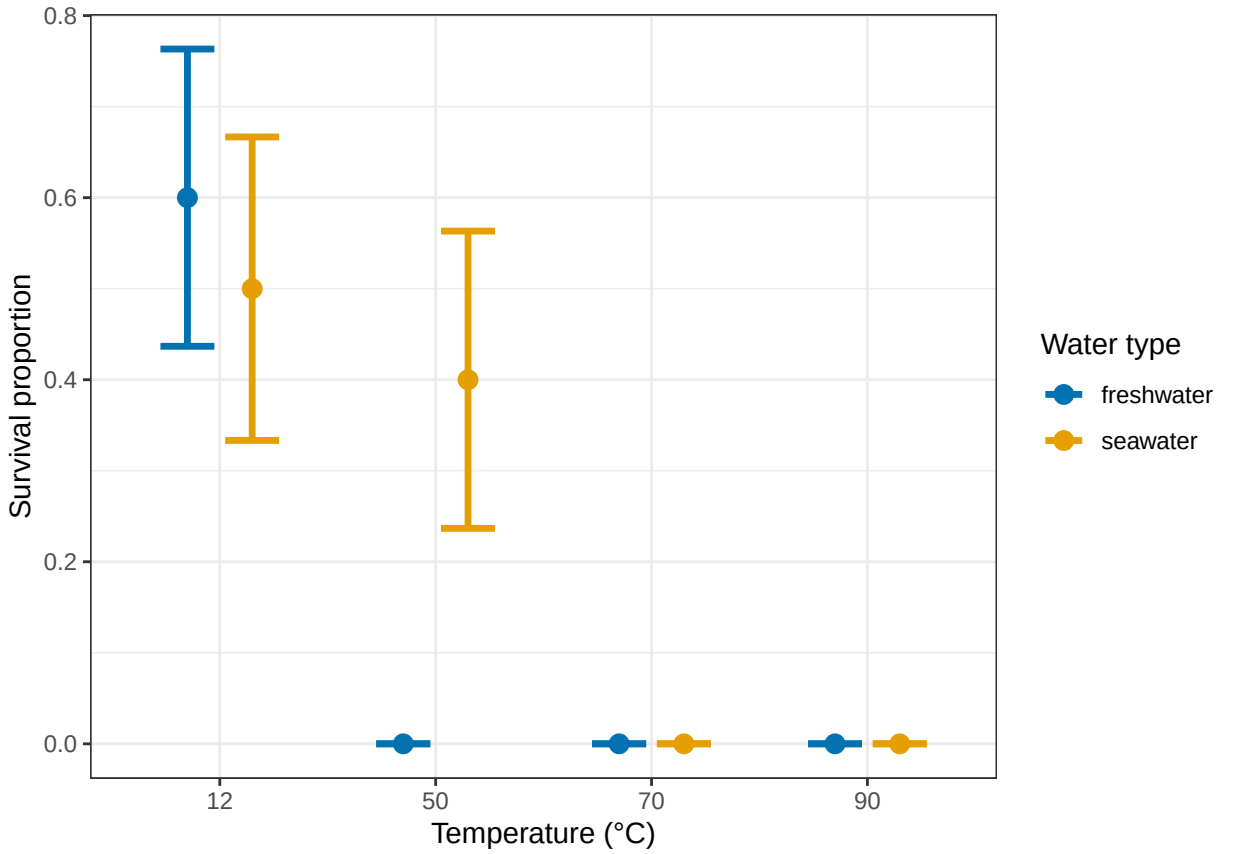
Survival

Visualizing Data

```
#making a column for the propotion of samples that survived per treatment
tunisurv<- tunidata%>%
  group_by(temperature_c, water_type)%>%
  summarize(proportion_survival=sum(survival)/10, sd=sd(survival), total= n(),
            SE = sd(survival)/sqrt(total))
```

```
## `summarise()` has grouped output by 'temperature_c'. You can override using the `.groups` argument.
```

```
# divided by 10 because each combination of
#temperature and water type has 10 data points

my_cols = c("freshwater" = "#0072B2", "seawater" = "#E69F00")

#plotting
pd <- position_dodge(width = 0.6)
ggplot(tunisurv, aes(x=temperature_c, y=proportion_survival, colour=water_type))+
  geom_point(aes(colour=water_type), position = pd, size=3)+
  xlab("Temperature (°C)")+
  ylab("Survival proportion")+
  geom_errorbar(aes(ymin=proportion_survival-SE, ymax=proportion_survival+SE,
                    width=0.5), size=1.2, position = pd)+
  theme_bw()+labs(colour="Water type")+
    scale_color_manual(values = my_cols)+
  scale_fill_manual(values = my_cols)
```




```
pd
```

```
## <ggproto object: Class PositionDodge, Position, gg>
##     compute_layer: function
##     compute_panel: function
##     preserve: total
##     required_aes:
##     setup_data: function
##     setup_params: function
##     width: 0.6
##     super:  <ggproto object: Class PositionDodge, Position, gg>
```

```
#ggsave("survivalproportion.jpg")
```

Testing for Normality

```
shapiro.test(tunidata$survival)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  tunidata$survival
## W = 0.47548, p-value = 2.047e-15
```

```
#p-value = 2.047e-15
```

Checking Distribution

```
fitDist(survival, data = tunidata, type = "binom", try.gamlss = T)
```

```
##   |                                                                                  |
##   system is computationally singular: reciprocal condition number = 5.31094e-21
##   |                                                                                  |==================
```

```
##
## Family:  c("BI", "Binomial")
## Fitting method: "nlminb"
##
## Call:  gamlssML(formula = y, family = BI)
##
## Mu Coefficients:
## [1]  -1.466
##
##  Degrees of Freedom for the fit: 1 Residual Deg. of Freedom   79
## Global Deviance:     77.2124
##             AIC:     79.2124
##             SBC:     81.5944
```

```
#family = BI (binomial)
```

GAMLSS model

```
mod_survival <- gamlss(survival ~ water_type + temperature_c + exposure_time_s + water_type*temperature_
                       family = BI, data = tunidata)
```

```
## GAMLSS-RS iteration 1: Global Deviance = 30.6187
## GAMLSS-RS iteration 2: Global Deviance = 30.6185
```

```
summary(mod_survival)
```

```
## ******************************************************************
## Family:  c("BI", "Binomial")
##
## Call:  gamlss(formula = survival ~ water_type + temperature_c +
##     exposure_time_s + water_type * temperature_c *
##     exposure_time_s + random(as.factor(colony_id)),
##     family = BI, data = tunidata)
##
## Fitting method: RS()
##
## ------------------------------------------------------------------
## Mu link function:  logit
## Mu Coefficients:
##                                                    Estimate Std. Error
```

```
## (Intercept)                                          -0.4979     0.9767
## water_typeseawater                                    0.9480     1.3928
## temperature_c50                                     -13.2564   378.1169
## temperature_c70                                     -13.2564   378.1889
## temperature_c90                                     -13.2564   378.1889
## exposure_time_s120                                    2.0913     1.5404
## water_typeseawater:temperature_c50                   11.2229   378.1200
## water_typeseawater:temperature_c70                   -0.9480   534.8400
## water_typeseawater:temperature_c90                   -0.9480   534.8400
## water_typeseawater:exposure_time_s120                -3.0393     2.0766
## temperature_c50:exposure_time_s120                   -2.0913   534.7300
## temperature_c70:exposure_time_s120                   -2.0913   534.8404
## temperature_c90:exposure_time_s120                   -2.0913   534.8404
## water_typeseawater:temperature_c50:exposure_time_s120  5.0728   534.7340
## water_typeseawater:temperature_c70:exposure_time_s120  3.0393   756.3722
## water_typeseawater:temperature_c90:exposure_time_s120  3.0393   756.3722
##                                                     t value Pr(>|t|)
## (Intercept)                                          -0.510    0.612
## water_typeseawater                                    0.681    0.499
## temperature_c50                                      -0.035    0.972
## temperature_c70                                      -0.035    0.972
## temperature_c90                                      -0.035    0.972
## exposure_time_s120                                    1.358    0.180
## water_typeseawater:temperature_c50                    0.030    0.976
## water_typeseawater:temperature_c70                   -0.002    0.999
## water_typeseawater:temperature_c90                   -0.002    0.999
## water_typeseawater:exposure_time_s120                -1.464    0.149
## temperature_c50:exposure_time_s120                   -0.004    0.997
## temperature_c70:exposure_time_s120                   -0.004    0.997
## temperature_c90:exposure_time_s120                   -0.004    0.997
## water_typeseawater:temperature_c50:exposure_time_s120  0.009    0.992
## water_typeseawater:temperature_c70:exposure_time_s120  0.004    0.997
## water_typeseawater:temperature_c90:exposure_time_s120  0.004    0.997
##
## ---------------------------------------------------------------------
## NOTE: Additive smoothing terms exist in the formulas:
##  i) Std. Error for smoothers are for the linear effect only.
## ii) Std. Error for the linear terms maybe are not accurate.
## ---------------------------------------------------------------------
## No. of observations in the fit:  80
## Degrees of Freedom for the fit:  19.80211
##       Residual Deg. of Freedom:  60.19789
##                       at cycle:  2
##
## Global Deviance:     30.61848
##             AIC:     70.2227
##             SBC:     117.3918
## *********************************************************************
```

Model Selection

```
step.modsurvival <- stepGAIC(mod_survival, direction = "backward", trace = F)
```

```
## Start:  AIC= 70.22
```

```
##  survival ~ water_type + temperature_c + exposure_time_s + water_type *
##     temperature_c * exposure_time_s + random(as.factor(colony_id))
```

```
summary(step.modsurvival)
```

```
## ******************************************************************
## Family:  c("BI", "Binomial")
##
## Call:  gamlss(formula = survival ~ temperature_c, family = BI,
##     data = tunidata, trace = FALSE)
##
## Fitting method: RS()
##
## ------------------------------------------------------------------
## Mu link function:  logit
## Mu Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)       0.2007     0.4495   0.446   0.6565
## temperature_c50  -1.5870     0.7173  -2.212   0.0299 *
## temperature_c70 -13.7667   197.3864  -0.070   0.9446
## temperature_c90 -13.7667   197.3864  -0.070   0.9446
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## ------------------------------------------------------------------
## No. of observations in the fit:  80
## Degrees of Freedom for the fit:  4
##       Residual Deg. of Freedom:  76
##                       at cycle:  2
##
## Global Deviance:     47.54175
##             AIC:     55.54175
##             SBC:     65.06986
## ******************************************************************
```

```
formula(step.modsurvival)
```

```
## survival ~ temperature_c
```

```
#survival ~ temperature_c
```

Kruskal-Wallis Test

```
kruskal.test(survival ~ temperature_c, data = tunidata)
```

```
##
##  Kruskal-Wallis rank sum test
##
## data:  survival by temperature_c
## Kruskal-Wallis chi-squared = 26.171, df = 3, p-value = 8.781e-06
```

```
#Kruskal-Wallis chi-squared = 26.171, df = 3, p-value = 8.781e-06
```

since temperature is the only explanatory variable - thus we can use Kruskal-Wallis Test to see p-values comparing temperatures to controls

```
dunnTest(survival ~ temperature_c, data = tunidata)
```

```
## Dunn (1964) Kruskal-Wallis multiple comparison

##   p-values adjusted with the Holm method.

##   Comparison        Z      P.unadj        P.adj
## 1    12 - 50 2.817892 4.834013e-03 1.933605e-02
## 2    12 - 70 4.428115 9.506008e-06 5.703605e-05
## 3    50 - 70 1.610224 1.073490e-01 3.220471e-01
## 4    12 - 90 4.428115 9.506008e-06 4.753004e-05
## 5    50 - 90 1.610224 1.073490e-01 2.146980e-01
## 6    70 - 90 0.000000 1.000000e+00 1.000000e+00
```

```
#Comparison        Z      P.unadj        P.adj
#1    12 - 50 2.817892 4.834013e-03 1.933605e-02*
#2    12 - 70 4.428115 9.506008e-06 5.703605e-05*
#3    50 - 70 1.610224 1.073490e-01 3.220471e-01
#4    12 - 90 4.428115 9.506008e-06 4.753004e-05*
#5    50 - 90 1.610224 1.073490e-01 2.146980e-01
#6    70 - 90 0.000000 1.000000e+00 1.000000e+00

#p-value for 12-90 and 12-70 is significant!
```

Attachment

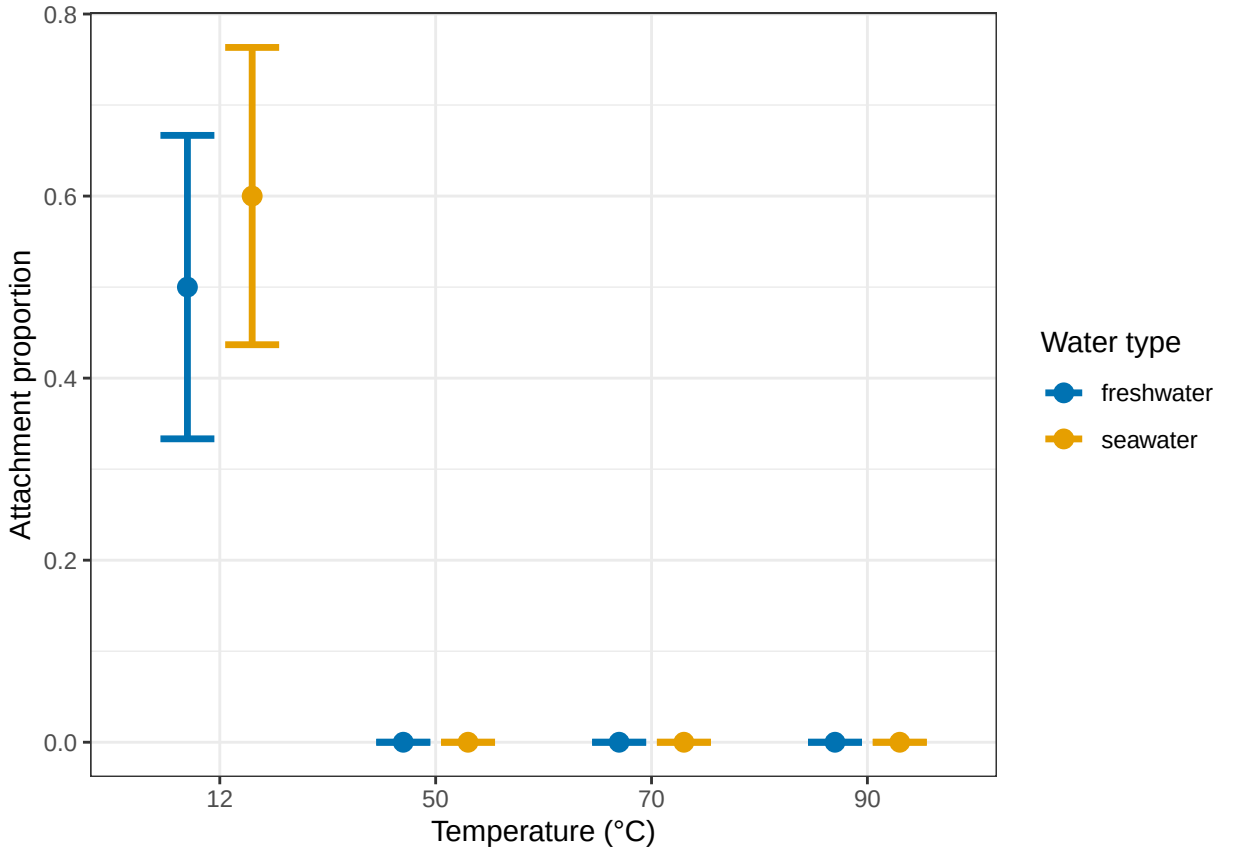
Data Visualization

```
#making a column for proportion of samples attached to the container
tuniattach<- tunidata%>%
  group_by(temperature_c, water_type)%>%
  summarize(proportion_attached=sum(attachment)/10, sd=sd(attachment),
            total= n(), SE = sd(attachment)/sqrt(total))
```

```
## `summarise()` has grouped output by 'temperature_c'. You can override using the `.groups` argument.
```

```
# divided by 10 because each temperature and water type combination has 10 data points


#plotting
ggplot(tuniattach, aes(x=temperature_c, y=proportion_attached, colour=water_type))+
  geom_point(aes(colour=water_type), position=pd, size=3)+
  xlab("Temperature (°C)")+
  ylab("Attachment proportion")+
  geom_errorbar(aes(ymin=proportion_attached-SE, ymax=proportion_attached+SE,
```

```
                  width=.5), size=1.2, position=pd)+
  theme_bw()+labs(colour="Water type")+
  scale_color_manual(values = my_cols)+
  scale_fill_manual(values = my_cols)
```




```
#ggsave("attachmentproportion.jpg")
```

Distribution Fitting

```
fitDist(attachment, data = tuni_data1, type = "binom", try.gamlss = T)
```

```
##   |                                                                      |
##   Lapack routine dgesv: system is exactly singular: U[1,1] = 0
##   |                                                                      |==================
```

```
##
## Family:  c("BI", "Binomial")
## Fitting method: "nlminb"
##
## Call:  gamlssML(formula = y, family = BI)
##
## Mu Coefficients:
## [1]  -1.836
##
```

```
##  Degrees of Freedom for the fit: 1 Residual Deg. of Freedom   79
## Global Deviance:     64.0639
##              AIC:     66.0639
##              SBC:     68.4459
```

```
#family = BI (binomial)
```

GAMLSS Model

```
mod <- gamlss(attachment ~ water_type + temperature_c + exposure_time_s + water_type*temperature_c*expos
```

```
## GAMLSS-RS iteration 1: Global Deviance = 26.9209
## GAMLSS-RS iteration 2: Global Deviance = 26.9206
```

```
summary(mod)
```

```
## ******************************************************************
## Family:  c("BI", "Binomial")
##
## Call:  gamlss(formula = attachment ~ water_type + temperature_c +
##     exposure_time_s + water_type * temperature_c *
##     exposure_time_s, family = BI, data = tuni_data1)
##
## Fitting method: RS()
##
## ------------------------------------------------------------------
## Mu link function:  logit
## Mu Coefficients:
##                                                      Estimate Std. Error
## (Intercept)                                           -0.4055     0.9129
## water_typeseawater                                     0.8109     1.2910
## temperature_c50                                      -13.1606   394.7864
## temperature_c70                                      -13.1606   394.7864
## temperature_c90                                      -13.1606   394.7864
## exposure_time_s120                                     0.8109     1.2910
## water_typeseawater:temperature_c50                    -0.8109   558.3026
## water_typeseawater:temperature_c70                    -0.8109   558.3026
## water_typeseawater:temperature_c90                    -0.8109   558.3026
## water_typeseawater:exposure_time_s120                 -0.8109     1.8257
## temperature_c50:exposure_time_s120                    -0.8109   558.3026
## temperature_c70:exposure_time_s120                    -0.8109   558.3026
## temperature_c90:exposure_time_s120                    -0.8109   558.3026
## water_typeseawater:temperature_c50:exposure_time_s120  0.8109   789.5523
## water_typeseawater:temperature_c70:exposure_time_s120  0.8109   789.5523
## water_typeseawater:temperature_c90:exposure_time_s120  0.8109   789.5523
##                                                      t value Pr(>|t|)
## (Intercept)                                           -0.444    0.658
## water_typeseawater                                     0.628    0.532
## temperature_c50                                       -0.033    0.974
## temperature_c70                                       -0.033    0.974
## temperature_c90                                       -0.033    0.974
## exposure_time_s120                                     0.628    0.532
```

```
## water_typeseawater:temperature_c50                     -0.001    0.999
## water_typeseawater:temperature_c70                     -0.001    0.999
## water_typeseawater:temperature_c90                     -0.001    0.999
## water_typeseawater:exposure_time_s120                  -0.444    0.658
## temperature_c50:exposure_time_s120                     -0.001    0.999
## temperature_c70:exposure_time_s120                     -0.001    0.999
## temperature_c90:exposure_time_s120                     -0.001    0.999
## water_typeseawater:temperature_c50:exposure_time_s120   0.001    0.999
## water_typeseawater:temperature_c70:exposure_time_s120   0.001    0.999
## water_typeseawater:temperature_c90:exposure_time_s120   0.001    0.999
##
## ------------------------------------------------------------------
## No. of observations in the fit:  80
## Degrees of Freedom for the fit:  16
##       Residual Deg. of Freedom:  64
##                       at cycle:  2
##
## Global Deviance:     26.92062
##             AIC:     58.92062
##             SBC:     97.03305
## ******************************************************************
```

Model Selection

```
step.mod <- stepAIC(mod, direction = "backward", trace = F)
```

```
## GAMLSS-RS iteration 1: Global Deviance = 26.9209
## GAMLSS-RS iteration 2: Global Deviance = 26.9206
## GAMLSS-RS iteration 1: Global Deviance = 26.9209
## GAMLSS-RS iteration 2: Global Deviance = 26.9206
## GAMLSS-RS iteration 1: Global Deviance = 26.9209
## GAMLSS-RS iteration 2: Global Deviance = 26.9206
## GAMLSS-RS iteration 1: Global Deviance = 27.1191
## GAMLSS-RS iteration 2: Global Deviance = 27.1189
## GAMLSS-RS iteration 1: Global Deviance = 27.3236
## GAMLSS-RS iteration 2: Global Deviance = 27.3233
## GAMLSS-RS iteration 1: Global Deviance = 27.526
## GAMLSS-RS iteration 2: Global Deviance = 27.5257
```

```
summary(step.mod)
```

```
## ******************************************************************
## Family:  c("BI", "Binomial")
##
## Call:  gamlss(formula = attachment ~ temperature_c, family = BI,
##     data = tuni_data1)
##
## Fitting method: RS()
##
## ------------------------------------------------------------------
## Mu link function:  logit
## Mu Coefficients:
```

```
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)       0.2007     0.4495   0.446    0.657
## temperature_c50 -13.7667   197.3881  -0.070    0.945
## temperature_c70 -13.7667   197.3881  -0.070    0.945
## temperature_c90 -13.7667   197.3881  -0.070    0.945
##
## ------------------------------------------------------------------
## No. of observations in the fit:  80
## Degrees of Freedom for the fit:  4
##       Residual Deg. of Freedom:  76
##                       at cycle:  2
##
## Global Deviance:     27.52571
##             AIC:     35.52571
##             SBC:     45.05381
## ******************************************************************
```

```
formula(step.mod)
```

```
## attachment ~ temperature_c
```

```
#attachment ~ temperature_c
```

Kruskie

```
kruskal.test(attachment ~ temperature_c, data = tunidata)
```

```
##
##  Kruskal-Wallis rank sum test
##
## data:  attachment by temperature_c
## Kruskal-Wallis chi-squared = 37.783, df = 3, p-value = 3.142e-08
```

```
#Kruskal-Wallis chi-squared = 37.783, df = 3, p-value = 3.142e-08
```

since temperature is the only explanatory variable - thus we can use Kruskal-Wallis Test to see p-values comparing temperatures to controls

```
dunnTest(attachment ~ temperature_c, data = tuni_data1)
```

```
## Dunn (1964) Kruskal-Wallis multiple comparison

##   p-values adjusted with the Holm method.

##   Comparison        Z      P.unadj        P.adj
## 1    12 - 50 5.018805 5.199384e-07 3.119630e-06
## 2    12 - 70 5.018805 5.199384e-07 2.599692e-06
## 3    50 - 70 0.000000 1.000000e+00 1.000000e+00
## 4    12 - 90 5.018805 5.199384e-07 2.079753e-06
## 5    50 - 90 0.000000 1.000000e+00 1.000000e+00
## 6    70 - 90 0.000000 1.000000e+00 1.000000e+00
```

```
#Comparison        Z      P.unadj        P.adj
#1    12 - 50 5.018805 5.199384e-07 3.119630e-06*
#2    12 - 70 5.018805 5.199384e-07 2.599692e-06*
#3    50 - 70 0.000000 1.000000e+00 1.000000e+00
#4    12 - 90 5.018805 5.199384e-07 2.079753e-06*
#5    50 - 90 0.000000 1.000000e+00 1.000000e+00
#6    70 - 90 0.000000 1.000000e+00 1.000000e+00

#p-value for 12-90, 12-70 and 12-50 are significant
```