

# BamfieldDSProject2021

Carter Burtlake and Jules Jones

25/11/2021

```
library(readxl)
library(ggpubr)
library(ggplot2)
library(dplyr)
library(performance)
library(agricolae)
library(scales)
library(viridis)
library(plotrix)
library(tidyverse)
library(fitdistrplus)
library(LambertW)
library(gamlss)
library(FSA)
library(formatR)
```

## Read In

```
DSDData <- read.csv("LittorinaAbundanceData_DS2021.csv")
```

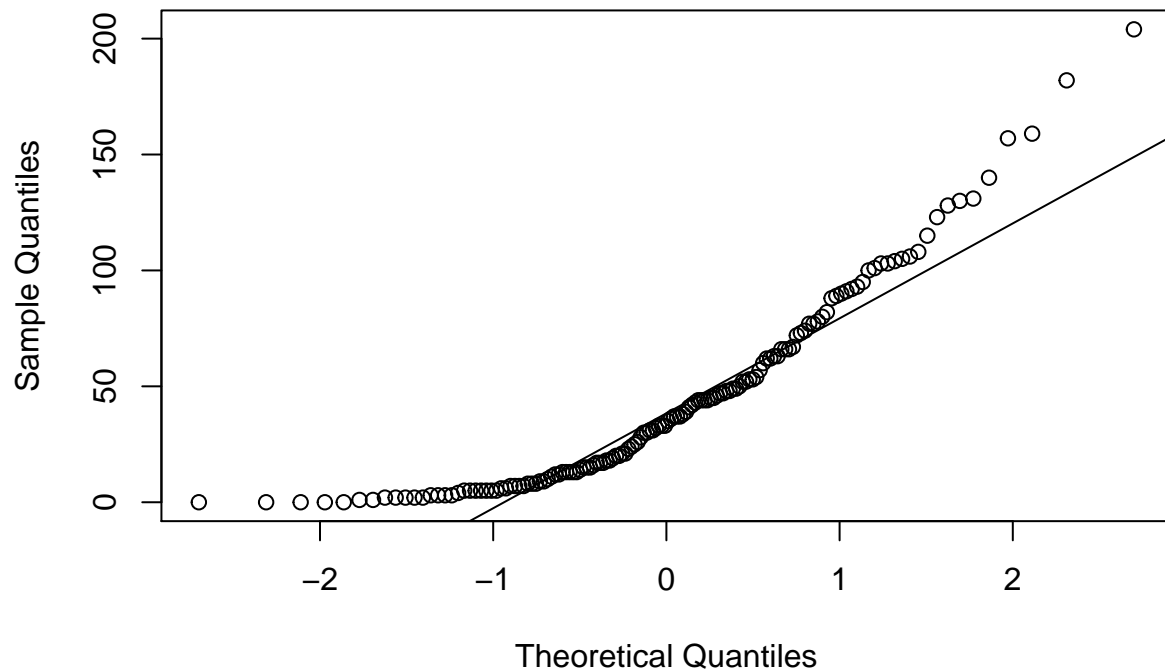
---

```
##### Distribution fitting###
```

```
test_no_percent_cover <- DSDData %>%
  dplyr::select(-notes) %>%
  dplyr::select(-percent_cover_barnacles) %>%
  na.omit() %>%
  mutate(wave_exposure = as.factor(wave_exposure), treatment = fct_relevel(treatment,
    "Control", "Mass_Mortality", "Cleared"))
# filter out unnecessary data for tests, change factor level of
# wave_exposure data, and remove NA's to allow functions to run below

# checking assumptions
qqnorm(DSDData$total)
qqline(DSDData$total)
```

## Normal Q-Q Plot



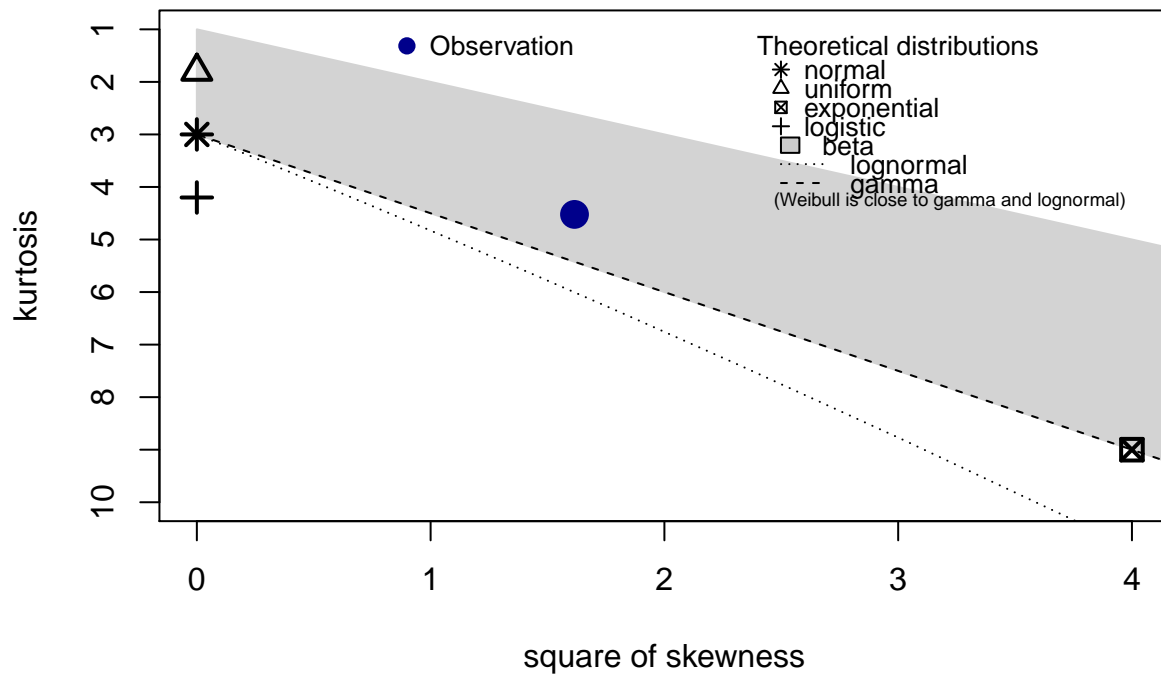
```
# visualizing normality: looks non-normal  
shapiro.test(DSData$total)
```

```
##  
## Shapiro-Wilk normality test  
##  
## data: DSData$total  
## W = 0.8765, p-value = 1.341e-09
```

```
# p= 1.341e-09, the distribution is significantly different from a  
# normal distribution
```

```
descdist(test_no_percent_cover$total)
```

## Cullen and Frey graph



```
## summary statistics
## -----
## min: 0    max: 204
## median: 34
## mean: 44.29167
## estimated sd: 41.74412
## estimated skewness: 1.27103
## estimated kurtosis: 4.522202
```

```
fitDist(total, data = test_no_percent_cover, type = "counts", try.gamlss = T)
```

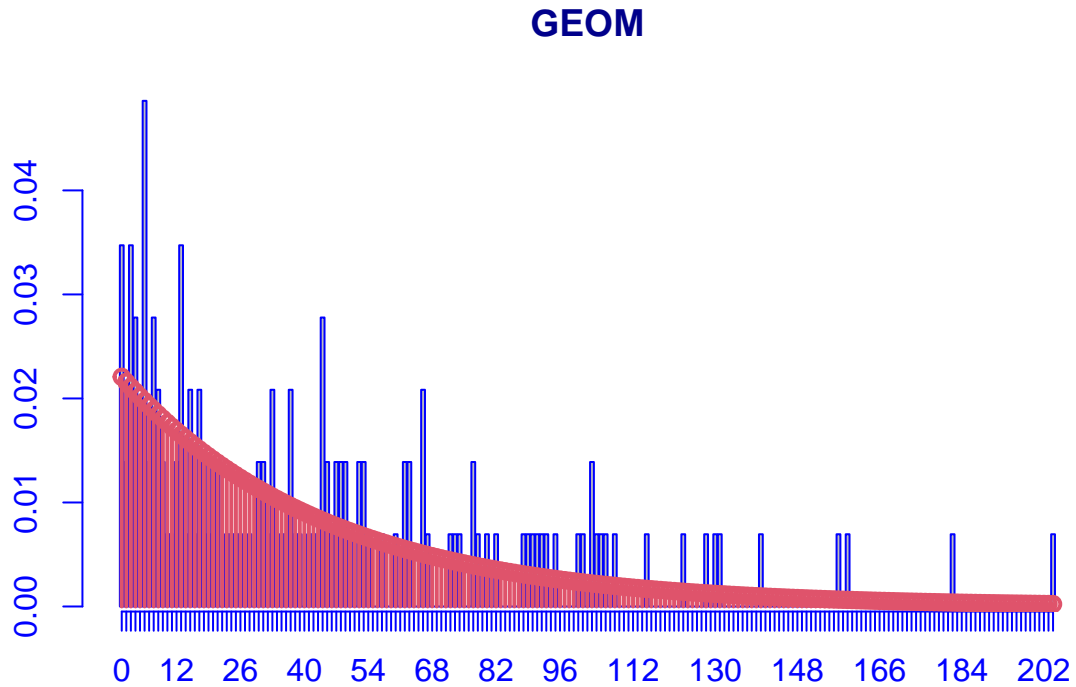
```
## |
## Lapack routine dgesv: system is exactly singular: U[2,2] = 0
## |
```

```
##
## Family: c("GEOM", "Geometric")
## Fitting method: "nlminb"
##
## Call: gamlssML(formula = y, family = DIST[i])
##
## Mu Coefficients:
## [1] 3.791
##
## Degrees of Freedom for the fit: 1 Residual Deg. of Freedom 143
```

```
## Global Deviance:    1382.98
##                   AIC:    1384.98
##                   SBC:    1387.95
```

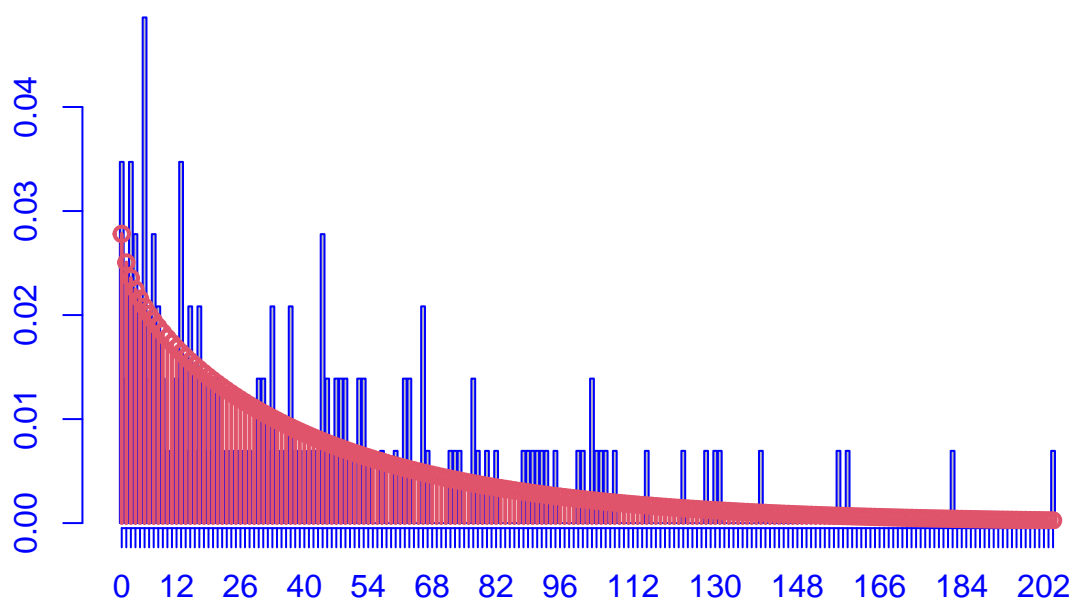
```
# Suggests distribution is best fit to Geometric family
```

```
total_geom <- histDist(test_no_percent_cover$total, "GEOM", density = T,
  main = "GEOM")
```



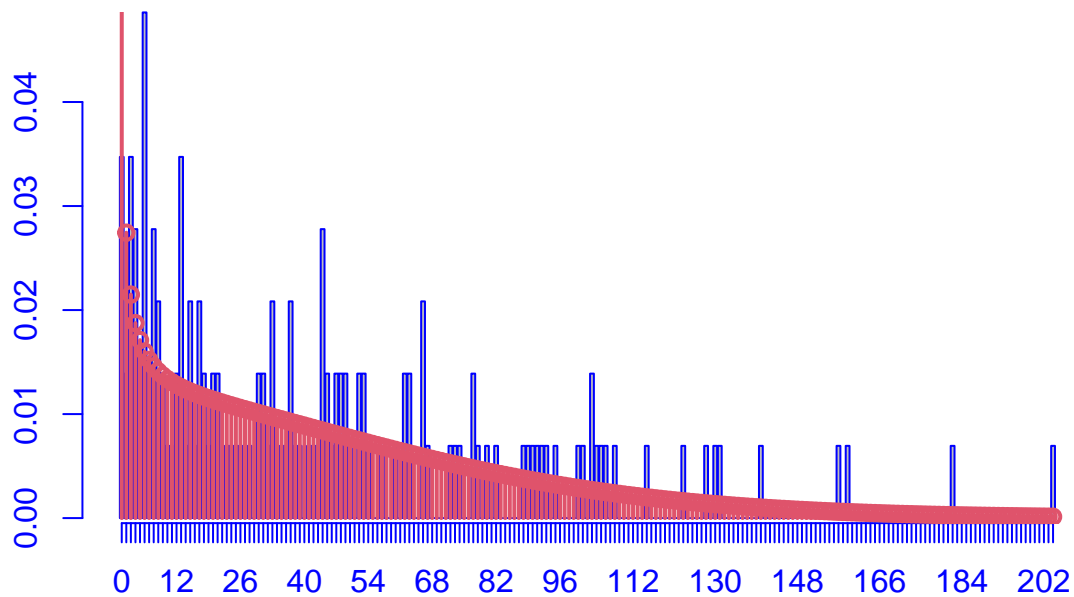
```
total_NBF <- histDist(test_no_percent_cover$total, "NBF", density = T,
  main = "NegBin")
```

## NegBin



```
total_dpoi <- histDist(test_no_percent_cover$total, "DP0", density = T,  
  main = "DPoisson")
```

## DPoisson



*# All seem to fit the distribution well*

```
GAIC(total_geom, total_NBF, total_dpoi)
```

```
##           df      AIC
## total_geom 1 1384.976
## total_dpoi  2 1387.394
## total_NBF   3 1388.408
```

*# Lowest AIC is Geometric (1384.976)*

*##### Models without summaries###*

```
Geometric.full.mod <- gamlss(total ~ treatment + survey + treatment * survey +
  random(as.factor(site)), family = GEOM(), data = test_no_percent_cover,
  control = gamlss.control(n.cyc = 100))
step.test_no_percent_cover_geom <- stepGAIC(Geometric.full.mod, direction = "backward",
  trace = F)
formula(Geometric.full.mod) # get the final formula
step.test_no_percent_cover_geom$anova
# look at the terms as they were taken out in a step-wise fashion

DoublePoisson.full.mod <- gamlss(total ~ treatment + survey + treatment *
  survey + random(as.factor(site)), family = DPO(), data = test_no_percent_cover,
```

```

control = gamlss.control(n.cyc = 100))
step.test_no_percent_cover_DP <- stepGAIC(DoublePoisson.full.mod, direction = "backward",
trace = F)
formula(DoublePoisson.full.mod) # get the final formula
step.test_no_percent_cover_DP$anova
# look at the terms as they were taken out in a step-wise fashion

NegativeBinomial.full.mod <- gamlss(total ~ treatment + survey + treatment *
survey + random(as.factor(site)), family = NBF(), data = test_no_percent_cover,
control = gamlss.control(n.cyc = 100))
step.test_no_percent_cover_NBF <- stepGAIC(NegativeBinomial.full.mod, direction = "backward",
trace = F)
formula(NegativeBinomial.full.mod) # get the final formula
step.test_no_percent_cover_NBF$anova
# look at the terms as they were taken out in a step-wise fashion

# Wave exposure final model
NegativeBinomial.wave_exposure.mod <- gamlss(total ~ treatment * survey *
wave_exposure + random(as.factor(site)), family = NBF(), data = test_no_percent_cover,
control = gamlss.control(n.cyc = 200))
step.test_no_percent_cover_NBF2 <- stepGAIC(NegativeBinomial.wave_exposure.mod,
direction = "backward", trace = F)
formula(NegativeBinomial.wave_exposure.mod) # get the final formula
step.test_no_percent_cover_NBF2$anova
# look at the terms as they were taken out in a step-wise fashion

```

```
##### Summaries of above Models###
```

```
summary(step.test_no_percent_cover_geom) # get a summary of final model above
```

```

## *****
## Family:  c("GEOM", "Geometric")
##
## Call:  gamlss(formula = total ~ treatment + survey + treatment *
##          survey + random(as.factor(site)), family = GEOM(),
##          data = test_no_percent_cover, control = gamlss.control(n.cyc = 100))
##
## Fitting method: RS()
##
## -----
## Mu link function:  log
## Mu Coefficients:
##
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)      3.57796    0.39588   9.038 1.55e-15 ***
## treatmentMass_Mortality -0.17551    0.55278  -0.318  0.7514
## treatmentCleared      0.97692    0.52717   1.853  0.0661 .
## survey              0.08415    0.18303   0.460  0.6464
## treatmentMass_Mortality:survey 0.22216    0.25556   0.869  0.3862
## treatmentCleared:survey -1.11786    0.24624  -4.540 1.24e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## -----

```

```

## 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: 144
## Degrees of Freedom for the fit: 9.82065
## Residual Deg. of Freedom: 134.1793
## at cycle: 4
##
## Global Deviance: 1273.505
## AIC: 1293.146
## SBC: 1322.311
## *****

# Summary of Geometric distribution fit to gamlss model above

summary(step.test_no_percent_cover_DP) # get a summary of final model above

## *****
## Family: c("DPO", "Double Poisson")
##
## Call: gamlss(formula = total ~ treatment + survey + treatment *
## survey + random(as.factor(site)), family = DPO(),
## data = test_no_percent_cover, control = gamlss.control(n.cyc = 100))
##
## Fitting method: RS()
##
## -----
## Mu link function: log
## Mu Coefficients:
##
## Estimate Std. Error t value Pr(>|t|)
## (Intercept) 3.97655 0.19277 20.628 < 2e-16 ***
## treatmentMass_Mortality -0.16744 0.26096 -0.642 0.5222
## treatmentCleared 1.62151 0.36700 4.418 2.04e-05 ***
## survey -0.01340 0.08958 -0.150 0.8814
## treatmentMass_Mortality:survey 0.21860 0.11811 1.851 0.0664 .
## treatmentCleared:survey -1.65181 0.26002 -6.353 3.10e-09 ***
## ---
## 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) 2.4939 0.1293 19.29 <2e-16 ***
## ---
## Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## -----
## 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: 144

```



```

## Degrees of Freedom for the fit: 10.84601
##      Residual Deg. of Freedom: 133.154
##              at cycle: 6
##
## Global Deviance:      1205.836
##      AIC:      1227.528
##      SBC:      1259.738
## *****

# Summary of Double Poisson distribution fit to gamlss model above

summary(step.test_no_percent_cover_NBF) # get a summary of final model above

## *****
## Family:  c("NBF", "NB Family")
##
## Call:  gamlss(formula = total ~ treatment + survey + treatment *
##      survey + random(as.factor(site)), family = NBF(),
##      data = test_no_percent_cover, control = gamlss.control(n.cyc = 100))
##
## Fitting method: RS()
##
## -----
## Mu link function:  log
## Mu Coefficients:
##
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)      3.67250    0.20543  17.877 < 2e-16 ***
## treatmentMass_Mortality -0.11799    0.27998  -0.421 0.674120
## treatmentCleared      1.25005    0.31969   3.910 0.000147 ***
## survey              0.04487    0.09449   0.475 0.635682
## treatmentMass_Mortality:survey 0.18853    0.12698   1.485 0.139992
## treatmentCleared:survey -1.29548    0.17233  -7.517 7.56e-12 ***
## ---
## 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)   0.9582    0.4661   2.056  0.0418 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## -----
## Nu link function:  log
## Nu Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)   0.33697    0.09155   3.681 0.000338 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## -----
## 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: 144
## Degrees of Freedom for the fit: 11.84602
## Residual Deg. of Freedom: 132.154
## at cycle: 52
##
## Global Deviance: 1200.754
## AIC: 1224.446
## SBC: 1259.627
## *****

# Summary of Negative Binomial distribution fit to gamlss model above
# (non-wave exposure version)

# Check AIC values for all three models run above
GAIC(NegativeBinomial.full.mod, DoublePoisson.full.mod, Geometric.full.mod)
```

```
## df AIC
## NegativeBinomial.full.mod 11.84602 1224.446
## DoublePoisson.full.mod 10.84601 1227.528
## Geometric.full.mod 9.82065 1293.146
```

```
# Negative Binomial distribution has lowest AIC value for the models
# (1224.446)
```

```
summary(step.test_no_percent_cover_NBF2) # get a summary of final model above
```

```
## *****
## Family: c("NBF", "NB Family")
##
## Call: gamlss(formula = total ~ treatment + survey + wave_exposure +
## random(as.factor(site)) + treatment:survey + survey:wave_exposure,
## family = NBF(), data = test_no_percent_cover, control = gamlss.control(n.cyc = 200),
## trace = FALSE)
##
## Fitting method: RS()
##
## -----
## Mu link function: log
## Mu Coefficients:
## Estimate Std. Error t value Pr(>|t|)
## (Intercept) 4.41510 0.21184 20.842 < 2e-16 ***
## treatmentMass_Mortality -0.09376 0.27148 -0.345 0.73036
## treatmentCleared 1.30062 0.31062 4.187 5.09e-05 ***
## survey -0.07711 0.09870 -0.781 0.43602
## wave_exposureSheltered -1.79610 0.24875 -7.220 3.50e-11 ***
## treatmentMass_Mortality:survey 0.17757 0.12318 1.442 0.15177
## treatmentCleared:survey -1.31845 0.16831 -7.833 1.29e-12 ***
## survey:wave_exposureSheltered 0.38467 0.11493 3.347 0.00106 **
## ---
## 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)  0.9068      0.4595   1.973  0.0505 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## -----
## Nu link function:  log
## Nu Coefficients:
##           Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.33401    0.09026    3.7 0.000313 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## -----
## 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:  144
## Degrees of Freedom for the fit:  10.00008
##      Residual Deg. of Freedom:  133.9999
##                      at cycle:  52
##
## Global Deviance:      1192.367
##           AIC:        1212.367
##           SBC:        1242.065
## *****

# Summary of Negative Binomial distribution fit to gamlss model above
# (wave exposure version)

# Check AIC values for new model compared to old ones
GAIC(NegativeBinomial.full.mod, DoublePoisson.full.mod, Geometric.full.mod,
      NegativeBinomial.wave_exposure.mod)

##                                df      AIC
## NegativeBinomial.wave_exposure.mod 14.00009 1218.520
## NegativeBinomial.full.mod           11.84602 1224.446
## DoublePoisson.full.mod               10.84601 1227.528
## Geometric.full.mod                   9.82065 1293.146

# Note: Negative Binomial distribution still has lowest AIC value for
# the models (1218.520) From models above

##### Graphs###

# Box plots!

Box_plot_total_snails = DSData %>%
  dplyr::select(-notes) %>%

```

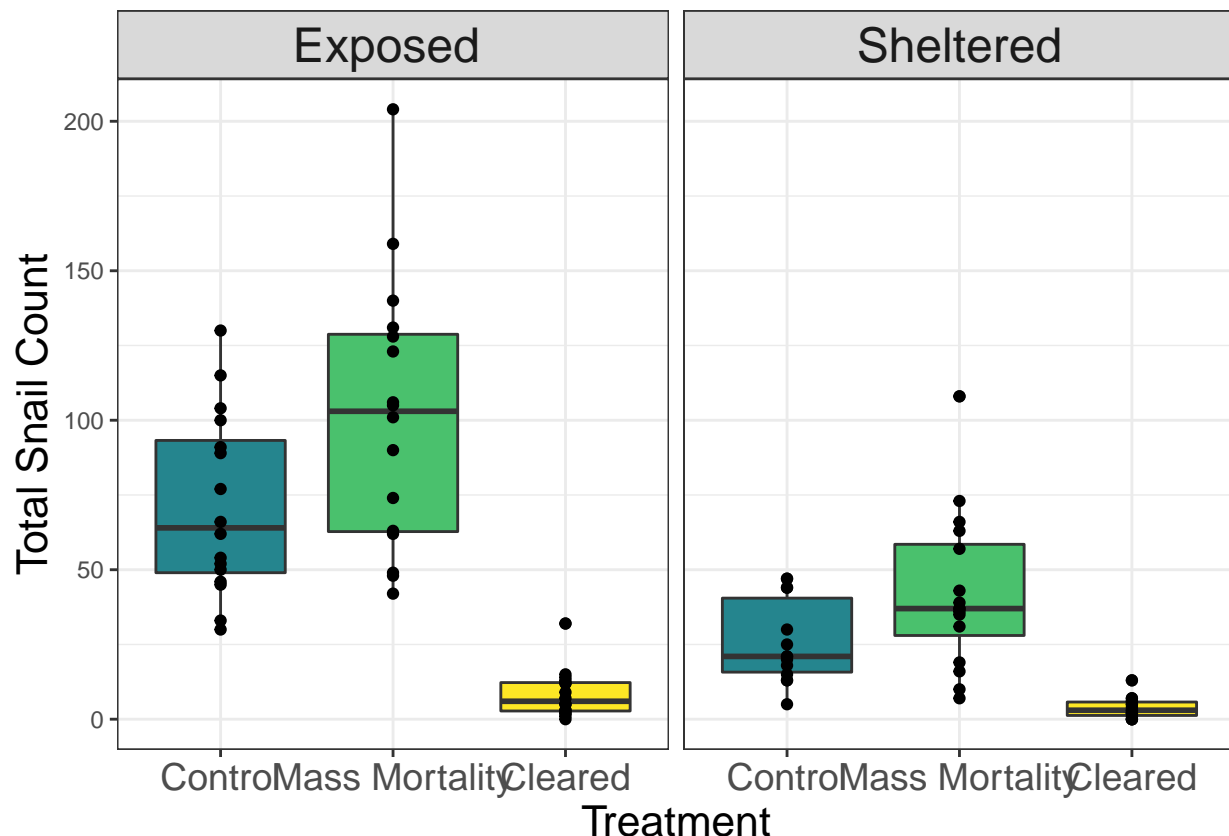
```

filter(survey == 2 | survey == 3) %>%
mutate(percentage_in = (in./total) * 100) %>%
mutate(treatment = as.factor(treatment)) %>%
mutate(treatment = fct_relevel(treatment, "Control", "Mass_Mortality",
                                "Cleared"))
# Filter the data for the plot and re-ordered the naming sequence to
# follow logical pattern

TotalSnailPlot <- ggplot(data = Box_plot_total_snails, aes(y = total, x = treatment,
  fill = treatment, color_palette(viridis))) + geom_boxplot() + geom_point() +
  xlab("Treatment") + ylab("Total Snail Count") + facet_grid(. ~ wave_exposure) +
  theme_bw() + theme(legend.position = "None", strip.text.x = element_text(size = 18),
    axis.title.x = element_text(size = 16), axis.text.x = element_text(size = 14),
    axis.title.y = element_text(size = 16)) + scale_fill_manual(values = c("#25858EFF",
    "#4AC16DFF", "#FDE725FF")) + scale_colour_manual(values = c("#25858EFF",
    "#4AC16DFF", "#FDE725FF"))

# plot and re-name legend to fit aesthetics
TotalSnailPlot + scale_x_discrete(labels = c(Control = "Control", Mass_Mortality = "Mass Mortality",
  Cleared = "Cleared"))

```



```

# Scatter plots

# With wave exposure (Figure 4)
exposed_scatter = DSData %>%

```

```

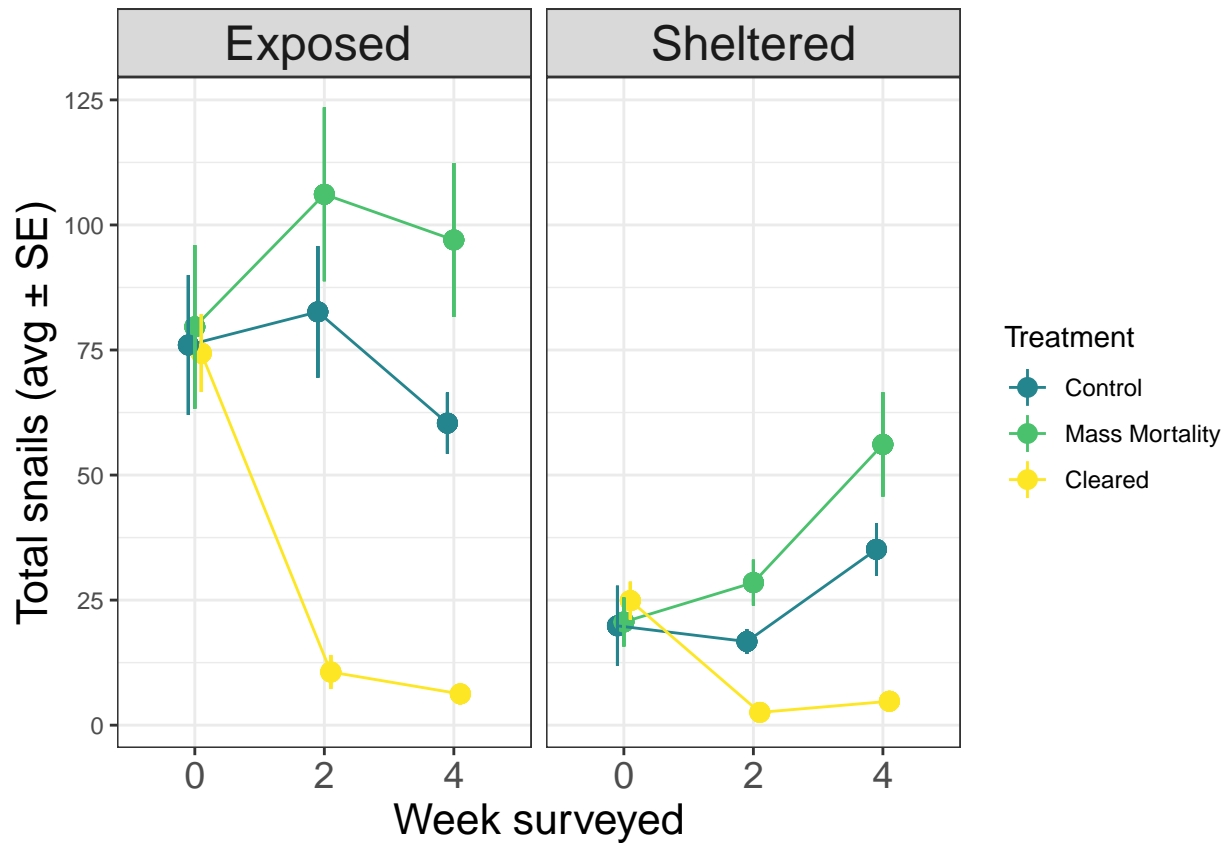
dplyr::select(-notes) %>%
dplyr::select(-c(percent_cover_barnacles)) %>%
mutate(percentage_in = (in./total) * 100) %>%
mutate(treatment = as.factor(treatment)) %>%
mutate(treatment = fct_relevel(treatment, "Control", "Mass_Mortality",
  "Cleared")) %>%
group_by(survey, treatment, wave_exposure) %>%
mutate(mean_total = mean(total), SE = std.error(total)) %>%
na.omit()

# Filtered data, received standard error, and re-ordered the naming
# sequence to follow logical pattern

exposure <- ggplot(exposed_scatter, aes(x = survey, y = mean_total, color = treatment,
  color_palette(viridis))) + geom_point(size = 3, position = position_dodge(width = 0.15)) +
  xlab("Week surveyed") + ylab("Total snails (avg ± SE)") + theme_bw() +
  geom_line(aes(group = treatment), position = position_dodge(width = 0.15)) +
  geom_linerange(aes(ymin = mean_total - SE, ymax = mean_total + SE),
    position = position_dodge(width = 0.15)) + facet_grid(. ~ wave_exposure) +
  theme(strip.text.x = element_text(size = 18), axis.title.x = element_text(size = 16),
    axis.text.x = element_text(size = 14), axis.title.y = element_text(size = 16)) +
  scale_x_discrete(name = "Week surveyed", limits = c("0", "2", "4")) +
  scale_colour_manual(values = c("#25858EFF", "#4AC16DFF", "#FDE725FF"),
    name = "Treatment", breaks = c("Control", "Mass_Mortality", "Cleared"),
    labels = c("Control", "Mass Mortality", "Cleared"))

# plot and re-name legend to fit aesthetics
exposure + scale_fill_discrete(name = "Treatment", labels = c("Control",
  "Mass mortality", "Cleared"))

```

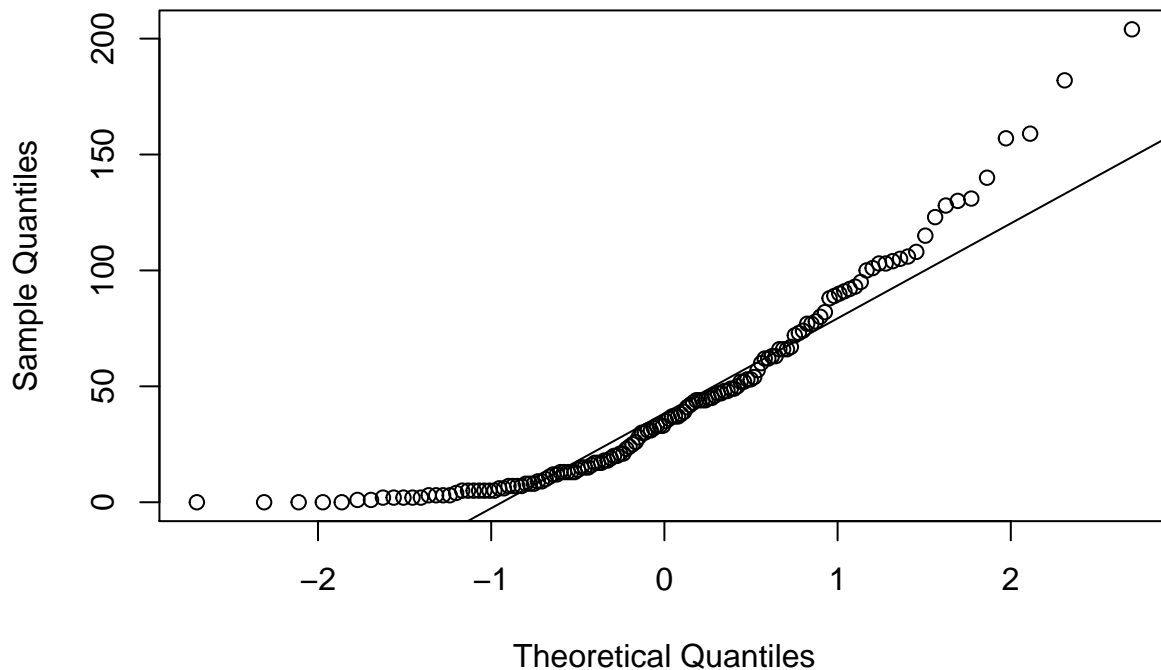


```
##### Percent Cover###

# Comparing exposures of barnacle cover at base level community
# structure
percent_cover <- DSData[c(0:48), ] %>%
  mutate(percent_cover_barnacles = as.numeric(percent_cover_barnacles)) %>%
  mutate(barnacle_cover = as.numeric(barnacle_cover))
# Change both percent_cover_barnacles and barnacle_cover to numeric
# and filter for percent cover prior to treatment as to compare sites

# Checking assumptions
qqnorm(DSData$total)
qqline(DSData$total)
```

## Normal Q-Q Plot



```
# visualizing normality: looks non-normal  
shapiro.test(percent_cover$percent_cover_barnacles)
```

```
##  
## Shapiro-Wilk normality test  
##  
## data: percent_cover$percent_cover_barnacles  
## W = 0.88694, p-value = 0.0002445
```

```
# p = 0.0002445, the distribution is significantly different from a  
# normal distribution Thus must use non-parametric anova
```

```
# Non-parametric anova  
kruskal.test(percent_cover_barnacles ~ site, data = percent_cover)
```

```
##  
## Kruskal-Wallis rank sum test  
##  
## data: percent_cover_barnacles by site  
## Kruskal-Wallis chi-squared = 35.694, df = 3, p-value = 8.689e-08
```

```
# p-value = 8.698x10^-8 so the barnacle cover differs across sites but  
# between which sites - Duns Test  
dunnTest(percent_cover_barnacles ~ site, data = percent_cover)
```

##	Comparison	Z	P.unadj	P.adj
## 1	Brady_point - Rance_island	4.0549733	5.014002e-05	2.005601e-04
## 2	Brady_point - Scotts_bay	-0.5688632	5.694490e-01	1.000000e+00
## 3	Rance_island - Scotts_bay	-4.6238365	3.767068e-06	2.260241e-05
## 4	Brady_point - Strawberry_point	3.7778349	1.581977e-04	4.745931e-04
## 5	Rance_island - Strawberry_point	-0.2771385	7.816738e-01	7.816738e-01
## 6	Scotts_bay - Strawberry_point	4.3466980	1.382023e-05	6.910114e-05

*# Dunn test proves Brady's Point is similar in cover to Scott's Bay*  
*# (p= 1.0) Dunn test proves Rance island is similar in cover to*  
*# Strawberry Point (p= 0.78) Dunn test proves to statistically differ*  
*# between all other sites*