

Proportional Scaling of Fish Abundance with Coral Number (Additive Model Only)

Adrian Stier

April 01, 2025

Introduction

Fish counts are discrete, non-negative data that are often modeled using count distributions. In this analysis, we test whether fish abundance scales proportionally with the number of corals by using observed count data from a one-coral treatment to predict counts for treatments with 3 and 6 corals. Because fish counts may exhibit overdispersion relative to a Poisson model, we model the counts with a negative binomial (NB) distribution.

The NB model is characterized by a mean (μ) and a dispersion parameter (often called “size”). We estimate the dispersion parameter from the one-coral data as follows:

$$\text{size}_1 = \frac{\bar{Y}_1^2}{\text{Var}(Y_1) - \bar{Y}_1}.$$

We then use a parametric bootstrap to quantify the uncertainty in the predicted means. In each bootstrap replicate, we simulate replicate one-coral counts from the NB model—now using 30 replicates per bootstrap sample—to obtain a more robust estimate of the one-coral mean. Under the additive variance model, the mean for k corals is simply k times the one-coral mean, and the NB size parameter is scaled additively as $k \times \text{size}_1$.

Observed Data for 1 Coral

The observed fish counts in the one-coral treatment are:

6, 10, 20.

```
# Observed fish counts for 1 coral
observed_1_coral <- c(6, 10, 20)
n1_obs <- length(observed_1_coral)

# Compute basic statistics for the observed one-coral data
mean_1_coral <- mean(observed_1_coral)
var_1_coral <- var(observed_1_coral)
sd_1_coral <- sqrt(var_1_coral)

cat("1-coral mean:", round(mean_1_coral, 2), "\n")
```

```
## 1-coral mean: 12
```

```
cat("1-coral variance:", round(var_1_coral, 2), "\n")
```

```
## 1-coral variance: 52
```

```
# Estimate the negative binomial dispersion parameter (size)
size1 <- mean_1_coral^2 / (var_1_coral - mean_1_coral)
cat("Estimated NB size for 1 coral:", round(size1, 2), "\n")
```

```
## Estimated NB size for 1 coral: 3.6
```

Expected Means for 3 and 6 Corals

Under proportional scaling, the expected mean fish counts are:

$$\bar{Y}_3 = 3\bar{Y}_1 \quad \text{and} \quad \bar{Y}_6 = 6\bar{Y}_1.$$

```
mean_3_expected <- 3 * mean_1_coral
mean_6_expected <- 6 * mean_1_coral

cat("Expected mean (3 corals):", round(mean_3_expected, 2), "\n")
```

```
## Expected mean (3 corals): 36
```

```
cat("Expected mean (6 corals):", round(mean_6_expected, 2), "\n")
```

```
## Expected mean (6 corals): 72
```

Parametric Bootstrapping with the Negative Binomial Model

The negative binomial model is used as the basis for our parametric bootstrap. This model accommodates overdispersion common in ecological count data. The bootstrap procedure is as follows: 1. Simulating One-Coral Data: For each bootstrap replicate, we simulate data from the NB distribution using the estimated parameters from the observed one-coral data. Here, we generate 30 replicate counts per bootstrap replicate (i.e. $(n_1^{\text{sim}} = 30)$) to enhance the precision of the one-coral mean estimate. 2. Calculating the Mean: We compute the mean of these 30 simulated counts to obtain a bootstrap estimate of the one-coral mean. 3. Scaling for Multiple Corals (Additive Model): Under the additive variance assumption: • The predicted mean for (k) corals is (k) times the bootstrap one-coral mean. dispersion parameter is scaled as

$$(k \times \text{size}_1)$$

We use these scaled parameters to simulate the predicted fish counts for 3 and 6 corals.

```
# Increase the number of simulated replicates for the one-coral treatment
n1_sim <- 30

# Parametric bootstrap: simulate 30 replicate one-coral counts per bootstrap sample
bootstrap_means_1 <- replicate(B, mean(rnbinom(n1_sim, size = size1, mu = mean_1_coral)))

# For the additive model, counts from k independent corals have:
# Mean = k * (one-coral mean from bootstrap)
# Size = k * size1
```

```

pred_3_add <- rnbino(B, size = 3 * size1, mu = 3 * bootstrap_means_1)
pred_6_add <- rnbino(B, size = 6 * size1, mu = 6 * bootstrap_means_1)

# Calculate 95% Confidence Intervals for the predicted means
ci_3_add <- quantile(pred_3_add, probs = c(0.025, 0.975))
ci_6_add <- quantile(pred_6_add, probs = c(0.025, 0.975))

cat("3-coral (Additive) 95% CI:",
    paste0("[", round(ci_3_add[1], 2), ", ", round(ci_3_add[2], 2), "]\n"))

```

```
## 3-coral (Additive) 95% CI: [15, 66]
```

```

cat("6-coral (Additive) 95% CI:",
    paste0("[", round(ci_6_add[1], 2), ", ", round(ci_6_add[2], 2), "]\n"))

```

```
## 6-coral (Additive) 95% CI: [39, 115]
```

Comparison with Observed Data for 3 and 6 Corals

Suppose we have the following observed data for the multi-coral treatments: • 3 Corals: (25, 30, 40, 35) • 6 Corals: (60, 70, 80, 90)

We calculate the observed means and the t-based 95% confidence intervals.

```

# Observed data for 3 and 6 corals
observed_3_coral <- c(25, 30, 40, 35)
observed_6_coral <- c(60, 70, 80, 90)

mean_3_obs <- mean(observed_3_coral)
mean_6_obs <- mean(observed_6_coral)

n3_obs <- length(observed_3_coral)
n6_obs <- length(observed_6_coral)

sd_3_obs <- sd(observed_3_coral)
sd_6_obs <- sd(observed_6_coral)

t_crit_3 <- qt(1 - alpha/2, n3_obs - 1)
t_crit_6 <- qt(1 - alpha/2, n6_obs - 1)

se_3_obs <- sd_3_obs / sqrt(n3_obs)
se_6_obs <- sd_6_obs / sqrt(n6_obs)

ci_3_obs <- c(mean_3_obs - t_crit_3 * se_3_obs,
              mean_3_obs + t_crit_3 * se_3_obs)
ci_6_obs <- c(mean_6_obs - t_crit_6 * se_6_obs,
              mean_6_obs + t_crit_6 * se_6_obs)

cat("Observed 3-coral mean:", round(mean_3_obs, 2),
    "; 95% CI: [", round(ci_3_obs[1], 2), ", ", round(ci_3_obs[2], 2), "]\n")

```

```
## Observed 3-coral mean: 32.5 ; 95% CI: [ 22.23 , 42.77 ]
```

```
cat("Observed 6-coral mean:", round(mean_6_obs, 2),
    "; 95% CI: [", round(ci_6_obs[1], 2), ", ", round(ci_6_obs[2], 2), "]\n")
```

```
## Observed 6-coral mean: 75 ; 95% CI: [ 54.46 , 95.54 ]
```

Species-Level Abundance Analysis Using a t-Based Approach

When modeling fish counts on individual corals using a negative binomial (NB) distribution, we sometimes encounter a problem: if the baseline (treatment 1) data for a species have very low variance or only a few replicates, the NB-based bootstrap may return a zero-width (or collapsed) confidence interval for the expected counts in the multi-coral treatments. In such cases, the computed NB dispersion parameter becomes unstable or even undefined, resulting in identical lower and upper confidence bounds.

An alternative approach is to use a t-(or z-) based scaling. If we assume that fish counts on individual corals are independent, then:

- The expected mean for k corals is

$$k\mu_1$$

- The variance of the total is

$$k\sigma_1^2$$

, so the standard error scales as:

$$SE_k = \sqrt{k}, \sigma_1.$$

Thus, for a 95% confidence interval for k corals, we have:

$$CI_k = k\mu_1 \pm t_{1-\alpha/2, df} \sqrt{k}, \sigma_1,$$

where $df = n_1 - 1$ is the degrees of freedom from the one-coral data and the significance level is given by

$$\alpha$$

.

Variance and Confidence Interval Scaling

The t-based method offers a robust alternative when the NB approach is unreliable due to low variance or few replicates. Below is the complete R code for computing the t-based confidence intervals for 3 and 6 corals.

```
# ----- Compute One-Coral Summary Statistics -----
# Observed fish counts for 1 coral
observed_1_coral <- c(6, 10, 20)
n1 <- length(observed_1_coral) # Number of replicates for 1 coral
mean_1 <- mean(observed_1_coral) # Mean for one coral (\mu_1)
sd_1 <- sd(observed_1_coral) # Standard deviation for one coral (\sigma_1)

# Display one-coral summary statistics
cat("One-coral mean:", round(mean_1, 2), "\n")
```

```
## One-coral mean: 12
```

```
cat("One-coral standard deviation:", round(sd_1, 2), "\n")
```

```
## One-coral standard deviation: 7.21
```

```

# ----- Define Multi-Coral Treatments -----
# Define the number of corals for which we want to scale the data
k_values <- c(3, 6)

# Degrees of freedom from the one-coral data (n1 - 1)
df <- n1 - 1

# Significance level and corresponding t critical value for a 95% CI
alpha <- 0.05
t_crit <- qt(1 - alpha/2, df)

# ----- Compute t-Based Confidence Intervals -----
# For each treatment (k corals), calculate the scaled mean, standard error, and CI.
ci_results <- lapply(k_values, function(k) {
  scaled_mean <- k * mean_1          # Expected mean for k corals: k\mu_1
  se_scaled <- sqrt(k) * sd_1        # Scaled standard error: \sqrt{k}\sigma_1
  lower_bound <- scaled_mean - t_crit * se_scaled # Lower bound of 95% CI
  upper_bound <- scaled_mean + t_crit * se_scaled # Upper bound of 95% CI
  list(k = k, mean = scaled_mean, lower = lower_bound, upper = upper_bound)
})

# Print the results for each multi-coral treatment
for (res in ci_results) {
  cat("For", res$k, "corals:\n")
  cat("  Scaled Mean =", round(res$mean, 2), "\n")
  cat("  95% CI = [", round(res$lower, 2), ", ", round(res$upper, 2), "]\n")
}

```

```

## For 3 corals:
##   Scaled Mean = 36
##   95% CI = [ -17.74 , 89.74 ]
## For 6 corals:
##   Scaled Mean = 72
##   95% CI = [ -4 , 148 ]

```

Mathematical Summary • Mean Scaling:

$$\mu_k = k\mu_1.$$

- Variance Scaling:

$$\sigma_k^2 = k\sigma_1^2 \implies SE_k = \sqrt{k}\sigma_1.$$

- t-Based 95% Confidence Interval for k Corals:

$$CI_k = k\mu_1 \pm t_{1-\alpha/2, df} \sqrt{k}\sigma_1.$$

Comparison and Interpretation

Using both the NB-based bootstrap and the t-based scaling provides a robust evaluation of the proportional scaling assumption. The NB-based approach is powerful for handling overdispersion; however, when the one-coral data are sparse or exhibit low variability, the t-based method ensures that the confidence intervals do not collapse. Comparing the two sets of intervals helps assess the reliability of the NB dispersion parameter estimate and the overall model assumptions.