Relative Abundance of Food Resources for Rufa Red Knot in the Cape Shore region of Delaware Bay

Elizabeth Bouchard

4/29/2021

# Introduction

Delaware Bay is renowned for the ecological interaction between spawning horseshoe crabs (*Limulus polyphemus*) and migrating rufa red knots (*Calidris canutus rufa*) (Figure 1). However, climate change is expected to increase timing asynchronies between these two annual cycles. These temporal mismatches have implications for the population persistence of the threatened rufa red knot. In order to inform effective species management efforts in the face of anthropogenic pressure, we must understand the distribution and availability of potential red knot prey items throughout the tidal flats.



Figure 1. Horseshoe Crab and Rufa Red Knot

The rufa red knot is a robin-sized shorebird that has one of the longest migrations in the entire animal kingdom. These federally threatened birds migrate between their wintering grounds (as far south as Tierra del Fuego, the southernmost tip of South America) and breeding grounds in the central Canadian Arctic (Figure 2). Delaware Bay is the principal stopover habitat for this subspecies, where birds stop to rest and refuel before completing their journey to the Arctic each spring.



Figure 2. Spring Migration Route of the Rufa Red Knot Showing a Key Stopover Site in Delaware Bay

Delaware Bay also hosts the world’s largest spawning population of horseshoe crabs, which produce an abundance of eggs along the high tide line (Figure 3). Horseshoe crab eggs are a lipid-rich, easily digestible resource that many species rely on. For 2-3 weeks in May, the rufa red knot stops in Delaware Bay, where they eat massive quantities of horseshoe crab eggs. Declines in horseshoe crab abundance since the 1990s, however, are associated with a rapid decline of red knots, leading to a listing as ‘threatened’ under the US Endangered Species Act in 2015. Although both populations have stabilized following the implementation of horseshoe crab management, climate change is expected to modify this predator-prey interaction by threatening the temporal and spatial synchronicity of red knot and horseshoe crab migrations. In Delaware Bay, the final stopover before birds reach the Arctic, peak spawning and bird migration both typically occur during the third or fourth week of May. Horseshoe crabs migrate to the beaches to spawn in May and June, during which red knots arrive and take advantage of the abundant food resources. If the timing of these events are off, then red knots may not obtain adequate resources to complete their migration to the Arctic. Addressing this potential threat is necessary to understand the factors underpinning this complex ecological interaction, as well as its consequences for population management.



Figure 3. Horseshoe Crabs Eggs on Beach Slope in Cape Shore

The horseshoe crab-red knot predator-prey relationship has been studied extensively in Delaware Bay since the 1980s. Although the importance of horseshoe crab eggs in the red knot diet has been well established, there has been little to no research on alternative prey resources in Delaware Bay. Red knots are known to eat polychaete worms, clams, and snails in their southern wintering grounds. These types of organisms are also present in Delaware Bay. If alternative prey species are available and abundant in Delaware Bay, red knots may rely more heavily on alternate prey resources when horseshoe crab eggs are scarce. Thus, understanding the distribution and relative abundance of these alternative prey species in Delaware Bay is also critical for anticipating the impacts of climate change on the rufa red knot population.

My thesis is part of a broader project investigating the distributions of horseshoe crab eggs and foraging rufa red knots in relation to oyster farms in the Cape Shore region of Delaware Bay. Preliminary field sampling was conducted in June 2020 with the goal of informing the planning process for the larger field study, which is set to begin intensive field sampling in 2021. The objectives of this preliminary effort were:

1. To determine sampling methodology (i.e. how to sample and where to sample),
2. To get a sense of what macroinvertebrate organisms are in the sediment, and
3. To improve our understanding of the distribution of horseshoe crab eggs and other potential red knot prey resources in the tidal flats.

The analysis presented in this report focuses on a narrow question using a subset of the preliminary data. For this final paper, I assessed the distribution of prey resources in relation to sandbars. **The specific research question targeted in this analysis is: Are there differences in prey abundance on the inshore and offshore sides of the sandbar?** This question is important because it provides baseline information about the red knot prey distribution, which has implications for where we might expect to find red knots foraging. It also provides information that will inform our sampling design. For example, if egg abundances are different on the inshore vs. the offshore side of the sandbar, then we may need to collect samples on both sides. However, if there is no difference, then perhaps we can collect fewer samples and focus our sampling efforts one side or the other.

# Methods

##Study Area

The fieldwork associated with this project was conducted at Rutgers Cape Shore Research Laboratory (CSL; Figure 4), which is located about 2 hours south of New Brunswick, New Jersey. This facility is situated in the Cape Shore region of Delaware Bay and provides the opportunity to observe the remarkable ecological phenomenon between horseshoe crabs and rufa red knots. Horseshoe crabs typically lay nests on the beaches in this region between April and June. Rufa red knot use this region as a stopover site during their northward migration, typically visiting for two to three weeks in May. The study area is characterized by tidal flats with sandbars and sloughs, the water-filled areas between sandbars at low tide.



Figure 4. Rutgers Cape Shore Laboratory is Located on Delaware Bay About Two Hours South of New Brunswick, New Jersey

## Field Sampling

Sediment core samples were collected in the tidal flats at the CSL on June 2, 2020. Sediment core samples were 6.5 cm in diameter by 5cm in height. The preliminary sampling effort was focused on the second sandbar, which was about 90ft from the high tide line (Figure 5). Sediment core samples were collected at four sites along the sandbar: A, B, C, and D (Figure 6). They were each spaced out by ~40m. At each of these four sites, one sediment core sample was collected on both the inshore and offshore sides of the sandbar every 15 minutes as the tide went out. The inshore and offshore sampling points are indicated by the orange stars. Samples were captured at six time points, for a total of 48 sediment cores. Samples were frozen at Rutgers Haskin Shellfish Research Laboratory prior to analysis.



Figure 5. A Sandbar in the Tidal Flats

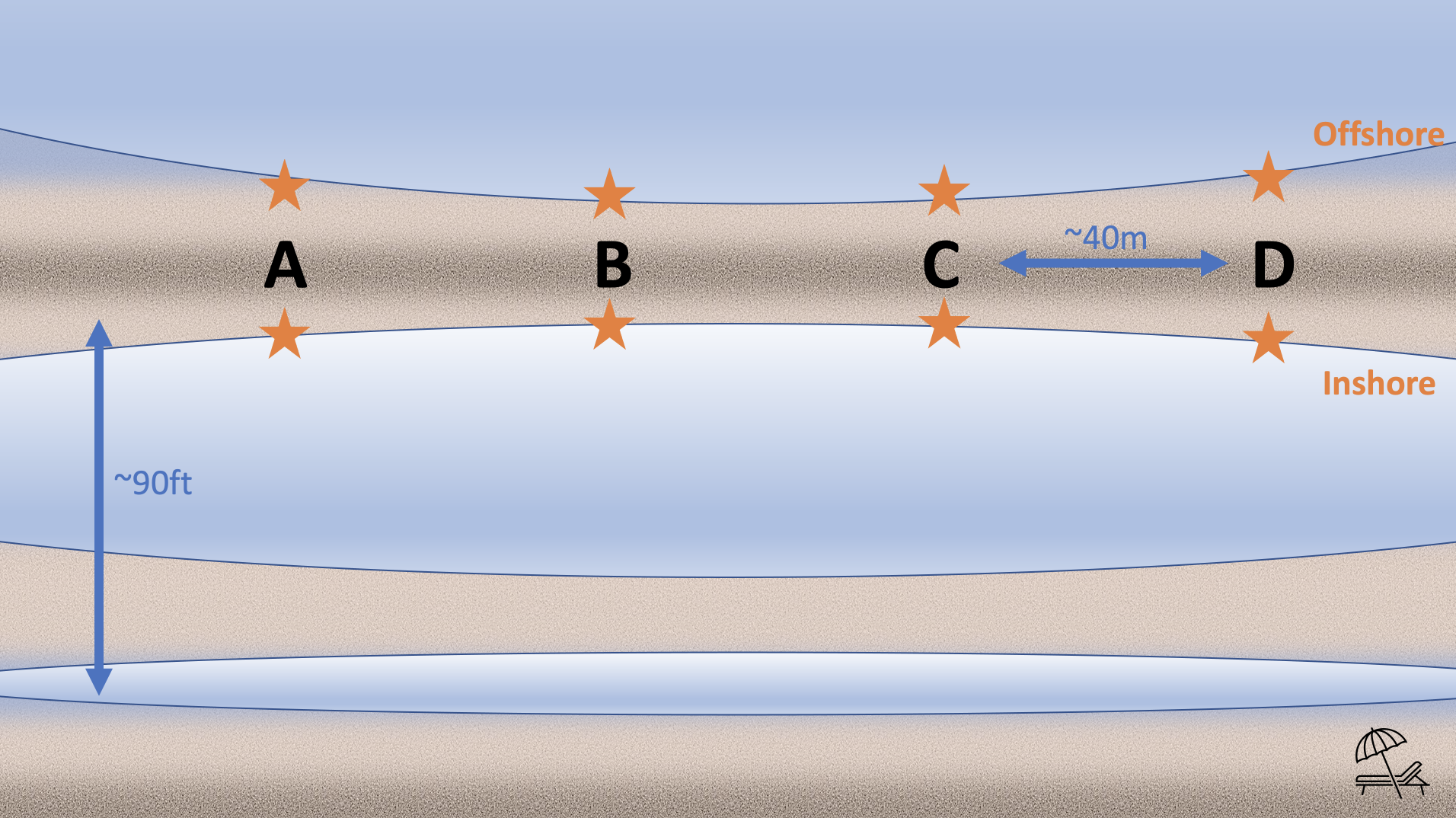


Figure 6. Sampling Design

## Laboratory Processing

Samples were thawed and rinsed through sediment sieves in order to sort particles into three size classes: 2mm, 1mm, and 0.5mm. For each sample, the contents of each sieve were transferred to separate petri dishes. These samples were examined under the dissecting microscope. All organisms were identified and the number of individual organisms per prey type was counted in each sample.

## Data Analysis Overview

I decided to use three variables for this analysis: sandbar site, the side of the sandbar, and abundance. In order to account for many zero counts in the dataset, I combined the abundance data for horseshoe crab eggs, worms, clams, and gastropods into one variable called prey abundance, assuming that these organisms are likely prey resources for rufa red knot. Note that the variables and prey types have different names in the R code and my report. In the R code, I refer to the sandbar site as “Point\_on\_Sandbar”, the side of the sandbar as “Inshore\_or\_Offshore”, and abundance as “Abundance” or “TotAbundance.” In the code, I refer to snails as “Gastropod”, worms as “Annelid”, and horseshoe crab eggs as “HSC\_eggs.”

To approach this analysis, I started by tidying the data. Then, I plotted the data. Next, I determined the appropriate model structure by running various models and checking diagnostics. To run the models, I used the lme4 and glmmTMB packages and I checked assumptions primarily using the DHARMa package. Next, I compared my alternative model to the null model, interpreted the results, and finally plotted the fitted model onto the data.

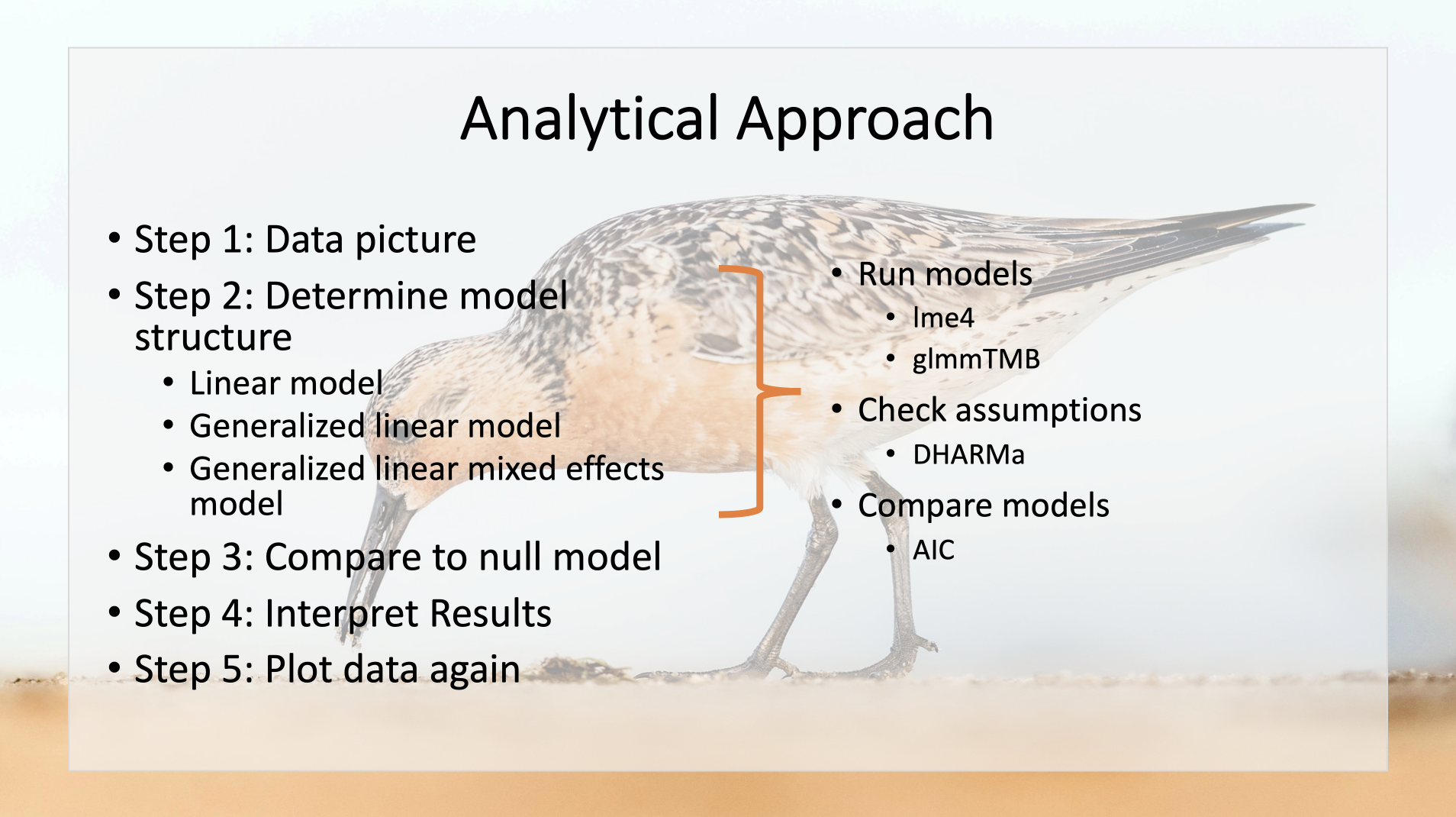


Diagram of Analytical Approach

# Results

##Tidying Data

First, I load the necessary libraries.

library(tidyverse)

## ── Attaching packages ─────────────────────────────────────── tidyverse 1.3.0 ──

## ✓ ggplot2 3.3.3 ✓ purrr 0.3.4  
## ✓ tibble 3.0.4 ✓ dplyr 1.0.2  
## ✓ tidyr 1.1.2 ✓ stringr 1.4.0  
## ✓ readr 1.4.0 ✓ forcats 0.5.0

## ── Conflicts ────────────────────────────────────────── tidyverse\_conflicts() ──  
## x dplyr::filter() masks stats::filter()  
## x dplyr::lag() masks stats::lag()

# tidyverse automatically loads all the packages below  
library(readr)   
library(tidyr)   
library(forcats)   
library(dplyr)   
library(ggplot2)   
library(stringr)   
  
library(ggfortify) # for autoplot  
library(lme4)

## Loading required package: Matrix

##   
## Attaching package: 'Matrix'

## The following objects are masked from 'package:tidyr':  
##   
## expand, pack, unpack

library(lmerTest)

##   
## Attaching package: 'lmerTest'

## The following object is masked from 'package:lme4':  
##   
## lmer

## The following object is masked from 'package:stats':  
##   
## step

require(glmmTMB) # for glmms

## Loading required package: glmmTMB

require(DHARMa) # for glm model evaluation

## Loading required package: DHARMa

## This is DHARMa 0.4.0. For overview type '?DHARMa'. For recent changes, type news(package = 'DHARMa') Note: Syntax of plotResiduals has changed in 0.3.0, see ?plotResiduals for details

Then, I import the complete dataset.

sandbar <- read\_csv("Final\_Project\_AEDA/Prelim\_Data\_Sandbar.csv")

##   
## ── Column specification ────────────────────────────────────────────────────────  
## cols(  
## .default = col\_double(),  
## Date\_Collected = col\_character(),  
## Date\_Processed = col\_character(),  
## Processor = col\_character(),  
## Point\_on\_Sandbar = col\_character(),  
## Inshore\_or\_Offshore = col\_character(),  
## Time\_Point = col\_character(),  
## Time = col\_time(format = ""),  
## Size\_Class = col\_character(),  
## Subsample = col\_character(),  
## Sample\_ID = col\_character(),  
## Subsample\_ID = col\_character(),  
## Notes = col\_character(),  
## Notes\_during\_Collection = col\_character(),  
## Notes\_during\_Processing = col\_character()  
## )  
## ℹ Use `spec()` for the full column specifications.

Now look at the data.

glimpse(sandbar)

## Rows: 192  
## Columns: 31  
## $ Date\_Collected <chr> "6/2/20", "6/2/20", "6/2/20", "6/2/20", "6/2/…  
## $ Date\_Processed <chr> "8/17/20", "8/17/20", "8/17/20", "8/17/20", "…  
## $ Processor <chr> "Jen", "Jen", "Jen", "Jen", "Elizabeth", "Eli…  
## $ Point\_on\_Sandbar <chr> "B", "B", "B", "B", "B", "B", "B", "B", "B", …  
## $ Inshore\_or\_Offshore <chr> "Offshore", "Offshore", "Offshore", "Offshore…  
## $ Time\_Point <chr> "T1", "T1", "T1", "T1", "T2", "T2", "T2", "T2…  
## $ Time <time> 10:10:00, 10:10:00, 10:10:00, 10:10:00, 10:2…  
## $ Size\_Class <chr> "2", "1", "0.5", "0.25", "2", "1", "0.5", "0.…  
## $ Subsample <chr> "1", "1", "1", "1/16", "1", "1", "1", "1/16",…  
## $ Sample\_ID <chr> "B-Offshore-T1", "B-Offshore-T1", "B-Offshore…  
## $ Subsample\_ID <chr> "B-Offshore-T1-2mm", "B-Offshore-T1-1mm", "B-…  
## $ Nematode <dbl> 1, 28, 356, 113, 0, 11, 292, 72, 0, 5, 156, 1…  
## $ Annelid <dbl> 0, 0, 0, 0, 0, 0, 2, 1, 0, 0, 1, 1, 0, 0, 1, …  
## $ Clam <dbl> 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, …  
## $ HSC\_egg <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, …  
## $ Gastropod <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, …  
## $ Copepod <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 2, 0, 0, 0, …  
## $ Ostracod <dbl> 0, 0, 0, 0, 0, 0, 0, 2, 0, 0, 0, 2, 0, 0, 0, …  
## $ Foraminifera <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, …  
## $ Empty\_oyster <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, …  
## $ Single\_clam\_valve <dbl> 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, …  
## $ Empty\_gastropod <dbl> 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, …  
## $ Empty\_ostracod <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, …  
## $ Exoskeleton <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, …  
## $ Isopod <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, …  
## $ Amphipod <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, …  
## $ `Unknown Star Organism` <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, …  
## $ Unknown <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, …  
## $ Notes <chr> NA, NA, "89\*4 nematodes; unknown org was an e…  
## $ Notes\_during\_Collection <chr> "Sample was wet. Birds were feeding at this s…  
## $ Notes\_during\_Processing <chr> NA, NA, "The unknown empty shell in 0.5mm is …

summary(sandbar)

## Date\_Collected Date\_Processed Processor Point\_on\_Sandbar   
## Length:192 Length:192 Length:192 Length:192   
## Class :character Class :character Class :character Class :character   
## Mode :character Mode :character Mode :character Mode :character   
##   
##   
##   
##   
## Inshore\_or\_Offshore Time\_Point Time Size\_Class   
## Length:192 Length:192 Length:192 Length:192   
## Class :character Class :character Class1:hms Class :character   
## Mode :character Mode :character Class2:difftime Mode :character   
## Mode :numeric   
##   
##   
##   
## Subsample Sample\_ID Subsample\_ID Nematode   
## Length:192 Length:192 Length:192 Min. : 0   
## Class :character Class :character Class :character 1st Qu.: 6   
## Mode :character Mode :character Mode :character Median : 53   
## Mean : 184   
## 3rd Qu.: 185   
## Max. :1638   
## NA's :87   
## Annelid Clam HSC\_egg Gastropod   
## Min. : 0.000 Min. : 0.0000 Min. : 0.000 Min. :0.0000   
## 1st Qu.: 0.000 1st Qu.: 0.0000 1st Qu.: 0.000 1st Qu.:0.0000   
## Median : 0.000 Median : 0.0000 Median : 0.000 Median :0.0000   
## Mean : 1.349 Mean : 0.5079 Mean : 1.222 Mean :0.0582   
## 3rd Qu.: 1.000 3rd Qu.: 0.0000 3rd Qu.: 0.000 3rd Qu.:0.0000   
## Max. :14.000 Max. :12.0000 Max. :101.000 Max. :4.0000   
## NA's :3 NA's :3 NA's :3 NA's :3   
## Copepod Ostracod Foraminifera Empty\_oyster   
## Min. :0.00000 Min. :0.0000 Min. : 0.000 Min. :0.000000   
## 1st Qu.:0.00000 1st Qu.:0.0000 1st Qu.: 0.000 1st Qu.:0.000000   
## Median :0.00000 Median :0.0000 Median : 0.000 Median :0.000000   
## Mean :0.06349 Mean :0.1481 Mean : 0.381 Mean :0.005291   
## 3rd Qu.:0.00000 3rd Qu.:0.0000 3rd Qu.: 0.000 3rd Qu.:0.000000   
## Max. :8.00000 Max. :3.0000 Max. :12.000 Max. :1.000000   
## NA's :3 NA's :3 NA's :3 NA's :3   
## Single\_clam\_valve Empty\_gastropod Empty\_ostracod Exoskeleton  
## Min. : 0.000 Min. :0.000 Min. :0.00000 Min. :0   
## 1st Qu.: 0.000 1st Qu.:0.000 1st Qu.:0.00000 1st Qu.:0   
## Median : 0.000 Median :0.000 Median :0.00000 Median :0   
## Mean : 1.487 Mean :0.254 Mean :0.02645 Mean :0   
## 3rd Qu.: 2.000 3rd Qu.:0.000 3rd Qu.:0.00000 3rd Qu.:0   
## Max. :15.000 Max. :7.000 Max. :4.00000 Max. :0   
## NA's :3 NA's :3 NA's :3 NA's :3   
## Isopod Amphipod Unknown Star Organism Unknown   
## Min. :0.00000 Min. :0 Min. :0.0 Min. :0.00000   
## 1st Qu.:0.00000 1st Qu.:0 1st Qu.:0.0 1st Qu.:0.00000   
## Median :0.00000 Median :0 Median :0.0 Median :0.00000   
## Mean :0.06842 Mean :0 Mean :0.1 Mean :0.05291   
## 3rd Qu.:0.00000 3rd Qu.:0 3rd Qu.:0.0 3rd Qu.:0.00000   
## Max. :2.00000 Max. :0 Max. :4.0 Max. :4.00000   
## NA's :2 NA's :2 NA's :2 NA's :3   
## Notes Notes\_during\_Collection Notes\_during\_Processing  
## Length:192 Length:192 Length:192   
## Class :character Class :character Class :character   
## Mode :character Mode :character Mode :character   
##   
##   
##   
##

Next, I need to manipulate the data to make columns for the total number of gastropods and clams for each subsample. Right now we have separate columns for: “Gastropod”, “Empty\_gastropod”, “Clam” and “Single\_clam\_valve”. “Gastropod” means that the snail shell contained tissues at the time of sample processing and “Empty\_gastropod” means that the snail did not contain tissues. We assume that all snails were alive at the time the sample was collected and that sieving the sample and flushing it with water removed the tissues. We assume the same for “Clam” and “Single\_clam\_valve,” the second of which was a single disarticulated clam valve. First, we have to make a row-wise data frame because, although dyplyr is good at performing operations over columns, it’s much more tricky over rows.

sandbar <- sandbar %>% rowwise(Subsample\_ID)

The next step is to make a “Gastropod” column that is the sum of “Gastropod” and “Empty\_gastropod” columns. This will estimate the number of gastropods that were in the sample upon collection. We are assuming that all gastropods (whether or not they contain tissues) were alive at the time of sample collection.

summary(sandbar)

## Date\_Collected Date\_Processed Processor Point\_on\_Sandbar   
## Length:192 Length:192 Length:192 Length:192   
## Class :character Class :character Class :character Class :character   
## Mode :character Mode :character Mode :character Mode :character   
##   
##   
##   
##   
## Inshore\_or\_Offshore Time\_Point Time Size\_Class   
## Length:192 Length:192 Length:192 Length:192   
## Class :character Class :character Class1:hms Class :character   
## Mode :character Mode :character Class2:difftime Mode :character   
## Mode :numeric   
##   
##   
##   
## Subsample Sample\_ID Subsample\_ID Nematode   
## Length:192 Length:192 Length:192 Min. : 0   
## Class :character Class :character Class :character 1st Qu.: 6   
## Mode :character Mode :character Mode :character Median : 53   
## Mean : 184   
## 3rd Qu.: 185   
## Max. :1638   
## NA's :87   
## Annelid Clam HSC\_egg Gastropod   
## Min. : 0.000 Min. : 0.0000 Min. : 0.000 Min. :0.0000   
## 1st Qu.: 0.000 1st Qu.: 0.0000 1st Qu.: 0.000 1st Qu.:0.0000   
## Median : 0.000 Median : 0.0000 Median : 0.000 Median :0.0000   
## Mean : 1.349 Mean : 0.5079 Mean : 1.222 Mean :0.0582   
## 3rd Qu.: 1.000 3rd Qu.: 0.0000 3rd Qu.: 0.000 3rd Qu.:0.0000   
## Max. :14.000 Max. :12.0000 Max. :101.000 Max. :4.0000   
## NA's :3 NA's :3 NA's :3 NA's :3   
## Copepod Ostracod Foraminifera Empty\_oyster   
## Min. :0.00000 Min. :0.0000 Min. : 0.000 Min. :0.000000   
## 1st Qu.:0.00000 1st Qu.:0.0000 1st Qu.: 0.000 1st Qu.:0.000000   
## Median :0.00000 Median :0.0000 Median : 0.000 Median :0.000000   
## Mean :0.06349 Mean :0.1481 Mean : 0.381 Mean :0.005291   
## 3rd Qu.:0.00000 3rd Qu.:0.0000 3rd Qu.: 0.000 3rd Qu.:0.000000   
## Max. :8.00000 Max. :3.0000 Max. :12.000 Max. :1.000000   
## NA's :3 NA's :3 NA's :3 NA's :3   
## Single\_clam\_valve Empty\_gastropod Empty\_ostracod Exoskeleton  
## Min. : 0.000 Min. :0.000 Min. :0.00000 Min. :0   
## 1st Qu.: 0.000 1st Qu.:0.000 1st Qu.:0.00000 1st Qu.:0   
## Median : 0.000 Median :0.000 Median :0.00000 Median :0   
## Mean : 1.487 Mean :0.254 Mean :0.02645 Mean :0   
## 3rd Qu.: 2.000 3rd Qu.:0.000 3rd Qu.:0.00000 3rd Qu.:0   
## Max. :15.000 Max. :7.000 Max. :4.00000 Max. :0   
## NA's :3 NA's :3 NA's :3 NA's :3   
## Isopod Amphipod Unknown Star Organism Unknown   
## Min. :0.00000 Min. :0 Min. :0.0 Min. :0.00000   
## 1st Qu.:0.00000 1st Qu.:0 1st Qu.:0.0 1st Qu.:0.00000   
## Median :0.00000 Median :0 Median :0.0 Median :0.00000   
## Mean :0.06842 Mean :0 Mean :0.1 Mean :0.05291   
## 3rd Qu.:0.00000 3rd Qu.:0 3rd Qu.:0.0 3rd Qu.:0.00000   
## Max. :2.00000 Max. :0 Max. :4.0 Max. :4.00000   
## NA's :2 NA's :2 NA's :2 NA's :3   
## Notes Notes\_during\_Collection Notes\_during\_Processing  
## Length:192 Length:192 Length:192   
## Class :character Class :character Class :character   
## Mode :character Mode :character Mode :character   
##   
##   
##   
##

sandbar <- mutate(sandbar, Gastropods = sum(c(Gastropod, Empty\_gastropod)))

Then, we repeat the above step for the “Ostracod” and “Empty\_ostracod” columns.

sandbar <- mutate(sandbar, Ostracods = sum(c(Ostracod, Empty\_ostracod)))

We make a “Clam” column that is the sum of current “Clam”, “Empty\_oyster”, and half of “Single\_clam\_valve”. This column will estimate the number of clams that were in the sample upon collection. We are assuming all single full valves that we found during processing were live clams at the time of sample collection. First, we divide “Single\_clam\_valve” by 2 in order to estimate the numbers of single valves that might have been full clams at the time of sample collection.

sandbar <- mutate(sandbar, Single\_clam\_valve = (Single\_clam\_valve/2))  
sandbar <- mutate(sandbar, Clams = sum(c(Clam, Empty\_oyster, Single\_clam\_valve)))  
sandbar <- mutate(sandbar, Clams = ceiling(Clams)) #Round up everything that ends with .5 to the nearest integer (whole number). This helps with AIC.

Now we are ready to prepare and tidy the data. I start by getting rid of columns of duplicate raw data (“Gastropod”, “Empty\_gastropod”, “Clam”, “Single\_clam\_valve”, “Empty\_oyster”, “Ostracod”, “Empty\_ostracod”) and notes to prevent confusion.

tidysandbar <- select(sandbar, -Clam, -Gastropod, -Single\_clam\_valve, -Empty\_gastropod, -Empty\_oyster, -Ostracod, -Empty\_ostracod, -Notes, -Notes\_during\_Collection, -Notes\_during\_Processing)  
tidysandbar <- gather(tidysandbar, Taxa, Abundance, c(12:24))

Next, I check the variable types using glimpse(), change the appropriate variables to factors, and check my work.

glimpse(tidysandbar)

## Rows: 2,496  
## Columns: 13  
## $ Date\_Collected <chr> "6/2/20", "6/2/20", "6/2/20", "6/2/20", "6/2/20",…  
## $ Date\_Processed <chr> "8/17/20", "8/17/20", "8/17/20", "8/17/20", "9/30…  
## $ Processor <chr> "Jen", "Jen", "Jen", "Jen", "Elizabeth", "Elizabe…  
## $ Point\_on\_Sandbar <chr> "B", "B", "B", "B", "B", "B", "B", "B", "B", "B",…  
## $ Inshore\_or\_Offshore <chr> "Offshore", "Offshore", "Offshore", "Offshore", "…  
## $ Time\_Point <chr> "T1", "T1", "T1", "T1", "T2", "T2", "T2", "T2", "…  
## $ Time <time> 10:10:00, 10:10:00, 10:10:00, 10:10:00, 10:25:00…  
## $ Size\_Class <chr> "2", "1", "0.5", "0.25", "2", "1", "0.5", "0.25",…  
## $ Subsample <chr> "1", "1", "1", "1/16", "1", "1", "1", "1/16", "1"…  
## $ Sample\_ID <chr> "B-Offshore-T1", "B-Offshore-T1", "B-Offshore-T1"…  
## $ Subsample\_ID <chr> "B-Offshore-T1-2mm", "B-Offshore-T1-1mm", "B-Offs…  
## $ Taxa <chr> "Nematode", "Nematode", "Nematode", "Nematode", "…  
## $ Abundance <dbl> 1, 28, 356, 113, 0, 11, 292, 72, 0, 5, 156, 120, …

tidysandbar <- mutate(tidysandbar, Point\_on\_Sandbar=factor(Point\_on\_Sandbar))  
tidysandbar <- mutate(tidysandbar, Inshore\_or\_Offshore=factor(Inshore\_or\_Offshore))  
tidysandbar <- mutate(tidysandbar, Time\_Point=factor(Time\_Point))  
tidysandbar <- mutate(tidysandbar, Size\_Class=factor(Size\_Class))  
tidysandbar <- mutate(tidysandbar, Sample\_ID=factor(Sample\_ID))  
tidysandbar <- mutate(tidysandbar, Subsample\_ID=factor(Subsample\_ID))  
tidysandbar <- mutate(tidysandbar, Taxa=factor(Taxa))  
#check my work  
glimpse(tidysandbar)

## Rows: 2,496  
## Columns: 13  
## $ Date\_Collected <chr> "6/2/20", "6/2/20", "6/2/20", "6/2/20", "6/2/20",…  
## $ Date\_Processed <chr> "8/17/20", "8/17/20", "8/17/20", "8/17/20", "9/30…  
## $ Processor <chr> "Jen", "Jen", "Jen", "Jen", "Elizabeth", "Elizabe…  
## $ Point\_on\_Sandbar <fct> B, B, B, B, B, B, B, B, B, B, B, B, B, B, B, B, B…  
## $ Inshore\_or\_Offshore <fct> Offshore, Offshore, Offshore, Offshore, Offshore,…  
## $ Time\_Point <fct> T1, T1, T1, T1, T2, T2, T2, T2, T3, T3, T3, T3, T…  
## $ Time <time> 10:10:00, 10:10:00, 10:10:00, 10:10:00, 10:25:00…  
## $ Size\_Class <fct> 2, 1, 0.5, 0.25, 2, 1, 0.5, 0.25, 2, 1, 0.5, 0.25…  
## $ Subsample <chr> "1", "1", "1", "1/16", "1", "1", "1", "1/16", "1"…  
## $ Sample\_ID <fct> B-Offshore-T1, B-Offshore-T1, B-Offshore-T1, B-Of…  
## $ Subsample\_ID <fct> B-Offshore-T1-2mm, B-Offshore-T1-1mm, B-Offshore-…  
## $ Taxa <fct> Nematode, Nematode, Nematode, Nematode, Nematode,…  
## $ Abundance <dbl> 1, 28, 356, 113, 0, 11, 292, 72, 0, 5, 156, 120, …

summary(tidysandbar)

## Date\_Collected Date\_Processed Processor Point\_on\_Sandbar  
## Length:2496 Length:2496 Length:2496 A:624   
## Class :character Class :character Class :character B:624   
## Mode :character Mode :character Mode :character C:468   
## D:780   
##   
##   
##   
## Inshore\_or\_Offshore Time\_Point Time Size\_Class   
## Extra-bird : 39 T1 :390 Length:2496 <0.25:156   
## Extra-Onshore: 39 T2 :390 Class1:hms 0.25 :312   
## Extra-Tread : 39 T3 :390 Class2:difftime 0.5 :676   
## Extra-Tread10: 39 T4 :390 Mode :numeric 1 :676   
## Inshore :1170 T5 :390 2 :676   
## Offshore :1170 T6 :390   
## NA's:156   
## Subsample Sample\_ID Subsample\_ID   
## Length:2496 D-Inshore-T1: 65 A-Extra-bird-NA-0.5mm : 13   
## Class :character D-Inshore-T2: 65 A-Extra-bird-NA-1mm : 13   
## Mode :character D-Inshore-T3: 65 A-Extra-bird-NA-2mm : 13   
## D-Inshore-T4: 65 A-Extra-Onshore-NA-0.5mm: 13   
## D-Inshore-T5: 65 A-Extra-Onshore-NA-1mm : 13   
## D-Inshore-T6: 65 A-Extra-Onshore-NA-2mm : 13   
## (Other) :2106 (Other) :2418   
## Taxa Abundance   
## Amphipod : 192 Min. : 0.000   
## Annelid : 192 1st Qu.: 0.000   
## Clams : 192 Median : 0.000   
## Copepod : 192 Mean : 8.537   
## Exoskeleton : 192 3rd Qu.: 0.000   
## Foraminifera: 192 Max. :1638.000   
## (Other) :1344 NA's :120

I need to fix the time variable. Right now I have two time columns. Time\_Point is categorical and Time is the time that the sample was collected in the field. I want to create a new column in the dataframe with a time variable (Time\_simple) that standardizes time (i.e. 0 min, 15 min, 30 min, and so on) as a numeric variable.

tidysandbar <- mutate(tidysandbar, Time\_simple = Time\_Point)  
levels(tidysandbar$Time\_simple)[levels(tidysandbar$Time\_simple)=="T1"] <- "0"  
levels(tidysandbar$Time\_simple)[levels(tidysandbar$Time\_simple)=="T2"] <- "15"  
levels(tidysandbar$Time\_simple)[levels(tidysandbar$Time\_simple)=="T3"] <- "30"  
levels(tidysandbar$Time\_simple)[levels(tidysandbar$Time\_simple)=="T4"] <- "45"  
levels(tidysandbar$Time\_simple)[levels(tidysandbar$Time\_simple)=="T5"] <- "60"  
levels(tidysandbar$Time\_simple)[levels(tidysandbar$Time\_simple)=="T6"] <- "75"  
glimpse(tidysandbar) #Time\_simple is currently a factor. We want to make it time.

## Rows: 2,496  
## Columns: 14  
## $ Date\_Collected <chr> "6/2/20", "6/2/20", "6/2/20", "6/2/20", "6/2/20",…  
## $ Date\_Processed <chr> "8/17/20", "8/17/20", "8/17/20", "8/17/20", "9/30…  
## $ Processor <chr> "Jen", "Jen", "Jen", "Jen", "Elizabeth", "Elizabe…  
## $ Point\_on\_Sandbar <fct> B, B, B, B, B, B, B, B, B, B, B, B, B, B, B, B, B…  
## $ Inshore\_or\_Offshore <fct> Offshore, Offshore, Offshore, Offshore, Offshore,…  
## $ Time\_Point <fct> T1, T1, T1, T1, T2, T2, T2, T2, T3, T3, T3, T3, T…  
## $ Time <time> 10:10:00, 10:10:00, 10:10:00, 10:10:00, 10:25:00…  
## $ Size\_Class <fct> 2, 1, 0.5, 0.25, 2, 1, 0.5, 0.25, 2, 1, 0.5, 0.25…  
## $ Subsample <chr> "1", "1", "1", "1/16", "1", "1", "1", "1/16", "1"…  
## $ Sample\_ID <fct> B-Offshore-T1, B-Offshore-T1, B-Offshore-T1, B-Of…  
## $ Subsample\_ID <fct> B-Offshore-T1-2mm, B-Offshore-T1-1mm, B-Offshore-…  
## $ Taxa <fct> Nematode, Nematode, Nematode, Nematode, Nematode,…  
## $ Abundance <dbl> 1, 28, 356, 113, 0, 11, 292, 72, 0, 5, 156, 120, …  
## $ Time\_simple <fct> 0, 0, 0, 0, 15, 15, 15, 15, 30, 30, 30, 30, 45, 4…

tidysandbar <- mutate(tidysandbar, Time\_simple=as.numeric(as.character(Time\_simple))) #Make Time\_simple numeric to represent the number of minutes since tide.  
glimpse(tidysandbar) #Check my work. It worked!

## Rows: 2,496  
## Columns: 14  
## $ Date\_Collected <chr> "6/2/20", "6/2/20", "6/2/20", "6/2/20", "6/2/20",…  
## $ Date\_Processed <chr> "8/17/20", "8/17/20", "8/17/20", "8/17/20", "9/30…  
## $ Processor <chr> "Jen", "Jen", "Jen", "Jen", "Elizabeth", "Elizabe…  
## $ Point\_on\_Sandbar <fct> B, B, B, B, B, B, B, B, B, B, B, B, B, B, B, B, B…  
## $ Inshore\_or\_Offshore <fct> Offshore, Offshore, Offshore, Offshore, Offshore,…  
## $ Time\_Point <fct> T1, T1, T1, T1, T2, T2, T2, T2, T3, T3, T3, T3, T…  
## $ Time <time> 10:10:00, 10:10:00, 10:10:00, 10:10:00, 10:25:00…  
## $ Size\_Class <fct> 2, 1, 0.5, 0.25, 2, 1, 0.5, 0.25, 2, 1, 0.5, 0.25…  
## $ Subsample <chr> "1", "1", "1", "1/16", "1", "1", "1", "1/16", "1"…  
## $ Sample\_ID <fct> B-Offshore-T1, B-Offshore-T1, B-Offshore-T1, B-Of…  
## $ Subsample\_ID <fct> B-Offshore-T1-2mm, B-Offshore-T1-1mm, B-Offshore-…  
## $ Taxa <fct> Nematode, Nematode, Nematode, Nematode, Nematode,…  
## $ Abundance <dbl> 1, 28, 356, 113, 0, 11, 292, 72, 0, 5, 156, 120, …  
## $ Time\_simple <dbl> 0, 0, 0, 0, 15, 15, 15, 15, 30, 30, 30, 30, 45, 4…

Now I need to exclude the taxa from the dataset that will not be used in this analysis. I will do so by checking levels of Taxa and then filtering out the unnecessary taxa and making a new dataframe. These unnecessary taxa (or groups of organisms) include: Exoskeleton, Unknown, and Unknown Star Organism. I will exclude nematodes as well because we did not count them for all samples and they are unlikely food resource for red knots. I excluded the <0.25 mm and 0.25 mm size classes because we did not process them for all samples. Lastly, I removed the extra samples because they are not part of this sampling design and they are unique: they have no replicates.

levels(tidysandbar$Taxa)

## [1] "Amphipod" "Annelid" "Clams"   
## [4] "Copepod" "Exoskeleton" "Foraminifera"   
## [7] "Gastropods" "HSC\_egg" "Isopod"   
## [10] "Nematode" "Ostracods" "Unknown"   
## [13] "Unknown Star Organism"

tidiersandbar <- filter(tidysandbar, Taxa != "Exoskeleton" & Taxa != "Unknown" & Taxa != "Unknown Star Organism")  
tidiersandbar <- filter(tidysandbar, Taxa != "Nematode" & Taxa != "Exoskeleton" & Taxa != "Unknown" & Taxa != "Unknown Star Organism")  
tidiersandbar <- filter(tidiersandbar, Size\_Class != "<0.25" & Size\_Class != "0.25")  
tidiersandbar <- tidiersandbar %>%  
 filter(!str\_detect(Inshore\_or\_Offshore, "Extra"))   
#Check data again  
glimpse(tidiersandbar)

## Rows: 1,296  
## Columns: 14  
## $ Date\_Collected <chr> "6/2/20", "6/2/20", "6/2/20", "6/2/20", "6/2/20",…  
## $ Date\_Processed <chr> "8/17/20", "8/17/20", "8/17/20", "9/30/20", "9/30…  
## $ Processor <chr> "Jen", "Jen", "Jen", "Elizabeth", "Elizabeth", "E…  
## $ Point\_on\_Sandbar <fct> B, B, B, B, B, B, B, B, B, B, B, B, B, B, B, B, B…  
## $ Inshore\_or\_Offshore <fct> Offshore, Offshore, Offshore, Offshore, Offshore,…  
## $ Time\_Point <fct> T1, T1, T1, T2, T2, T2, T3, T3, T3, T4, T4, T4, T…  
## $ Time <time> 10:10:00, 10:10:00, 10:10:00, 10:25:00, 10:25:00…  
## $ Size\_Class <fct> 2, 1, 0.5, 2, 1, 0.5, 2, 1, 0.5, 2, 1, 0.5, 2, 1,…  
## $ Subsample <chr> "1", "1", "1", "1", "1", "1", "1", "1", "1", "1",…  
## $ Sample\_ID <fct> B-Offshore-T1, B-Offshore-T1, B-Offshore-T1, B-Of…  
## $ Subsample\_ID <fct> B-Offshore-T1-2mm, B-Offshore-T1-1mm, B-Offshore-…  
## $ Taxa <fct> Annelid, Annelid, Annelid, Annelid, Annelid, Anne…  
## $ Abundance <dbl> 0, 0, 0, 0, 0, 2, 0, 0, 1, 0, 0, 1, 1, 0, 2, 0, 0…  
## $ Time\_simple <dbl> 0, 0, 0, 15, 15, 15, 30, 30, 30, 45, 45, 45, 60, …

summary(tidiersandbar)

## Date\_Collected Date\_Processed Processor Point\_on\_Sandbar  
## Length:1296 Length:1296 Length:1296 A:324   
## Class :character Class :character Class :character B:324   
## Mode :character Mode :character Mode :character C:324   
## D:324   
##   
##   
##   
## Inshore\_or\_Offshore Time\_Point Time Size\_Class   
## Extra-bird : 0 T1:216 Length:1296 <0.25: 0   
## Extra-Onshore: 0 T2:216 Class1:hms 0.25 : 0   
## Extra-Tread : 0 T3:216 Class2:difftime 0.5 :432   
## Extra-Tread10: 0 T4:216 Mode :numeric 1 :432   
## Inshore :648 T5:216 2 :432   
## Offshore :648 T6:216   
##   
## Subsample Sample\_ID Subsample\_ID   
## Length:1296 A-Inshore-T1: 27 A-Inshore-T1-0.5mm: 9   
## Class :character A-Inshore-T2: 27 A-Inshore-T1-1mm : 9   
## Mode :character A-Inshore-T3: 27 A-Inshore-T1-2mm : 9   
## A-Inshore-T4: 27 A-Inshore-T2-0.5mm: 9   
## A-Inshore-T5: 27 A-Inshore-T2-1mm : 9   
## A-Inshore-T6: 27 A-Inshore-T2-2mm : 9   
## (Other) :1134 (Other) :1242   
## Taxa Abundance Time\_simple   
## Amphipod :144 Min. : 0.0000 Min. : 0.0   
## Annelid :144 1st Qu.: 0.0000 1st Qu.:15.0   
## Clams :144 Median : 0.0000 Median :37.5   
## Copepod :144 Mean : 0.4259 Mean :37.5   
## Foraminifera:144 3rd Qu.: 0.0000 3rd Qu.:60.0   
## Gastropods :144 Max. :32.0000 Max. :75.0   
## (Other) :432

Based on my understanding of the literature, I think that the most likely prey resources for rufa red knot in Delaware Bay are: horseshoe crab eggs, clams, gastropods, and annelids. For my analysis, I am going to exclude all of the other taxa and organize these potential prey items into a single group called “Prey.”

sandbar\_prey <- tidiersandbar[(tidiersandbar$Taxa=="HSC\_egg") | (tidiersandbar$Taxa=="Annelid") | (tidiersandbar$Taxa=="Clams")   
 | (tidiersandbar$Taxa=="Gastropods"),]

Next, I explore the data. Of the four types of prey items considered in the analysis, annelids are most abundance (233), followed by clams (170), horseshoe crab eggs (90), and gastropods (23).

sandbar\_prey %>%  
 group\_by(Taxa) %>%  
 summarize(TotAbundance = sum(Abundance))

## `summarise()` ungrouping output (override with `.groups` argument)

## # A tibble: 4 x 2  
## Taxa TotAbundance  
## <fct> <dbl>  
## 1 Annelid 233  
## 2 Clams 186  
## 3 Gastropods 23  
## 4 HSC\_egg 90

Finally, I made the dataframe to be used in our analysis by removing size class. When size class is included in the dataset, each row is a subsample. By removing this column from the dataset, each row is now a single sample.

simplesand <- sandbar\_prey %>%  
 group\_by(Point\_on\_Sandbar, Inshore\_or\_Offshore, Time\_simple) %>%  
 summarize(TotAbundance = sum(Abundance))

## `summarise()` regrouping output by 'Point\_on\_Sandbar', 'Inshore\_or\_Offshore' (override with `.groups` argument)

## Data Picture

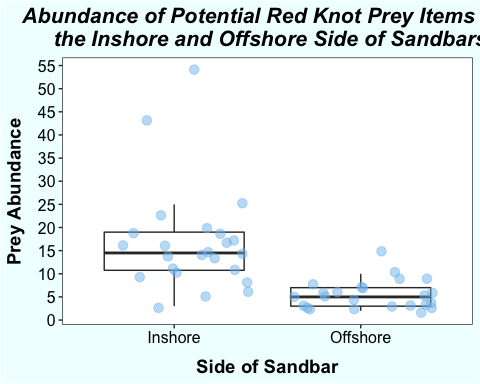
Reminder: What variables am I working with?

summary(simplesand)

## Point\_on\_Sandbar Inshore\_or\_Offshore Time\_simple TotAbundance   
## A:12 Extra-bird : 0 Min. : 0.0 Min. : 2.00   
## B:12 Extra-Onshore: 0 1st Qu.:15.0 1st Qu.: 4.75   
## C:12 Extra-Tread : 0 Median :37.5 Median : 8.50   
## D:12 Extra-Tread10: 0 Mean :37.5 Mean :11.08   
## Inshore :24 3rd Qu.:60.0 3rd Qu.:15.00   
## Offshore :24 Max. :75.0 Max. :54.00

This is the preliminary boxplot of my data. The blue points are the data points. As you can see, it looks like prey abundance may be greater on the inshore side of the sandbar. The Median abundance on the inshore side is high (14.5) compared to offshore side (5) of sandbar. I also noticed that there is greater spread on the inshore side of the sandbar, suggesting that prey availability may be more variable on the inshore side.

ggplot(simplesand, aes(x=Inshore\_or\_Offshore, y=TotAbundance))+  
 geom\_boxplot(outlier.shape = NA) +  
 geom\_jitter(alpha=0.5,color = "skyblue2",size=3) +  
 theme\_bw() +  
 ggtitle('Abundance of Potential Red Knot Prey Items on \n the Inshore and Offshore Side of Sandbars') +  
 xlab("Side of Sandbar") +   
 ylab("Prey Abundance") +  
 theme(plot.title = element\_text(color="black", size=16, face="bold.italic", hjust = 0.5),  
 axis.title.x = element\_text(color="black", size=14, face="bold", margin = margin(t = 10, r = 0, b = 0, l = 0)),  
 axis.title.y = element\_text(color="black", size=14, face="bold", margin = margin(t = 0, r = 10, b = 0, l = 0)),  
 axis.text.x = element\_text(color="black", size=12),  
 axis.text.y = element\_text(color="black", size=12),  
 panel.background = element\_rect(fill = "white"),  
 panel.grid.major = element\_blank(),  
 panel.grid.minor = element\_blank(),  
 plot.background = element\_rect(fill = "azure")) +  
 scale\_y\_continuous(breaks=seq(0,55,5))



I also calculate the center of the distributions, as referenced in the above distribution of the figure.

In <- subset(simplesand, Inshore\_or\_Offshore == "Inshore")  
median(In$TotAbundance) #14.5

## [1] 14.5

mean(In$TotAbundance) #16.75

## [1] 16.75

Off <- subset(simplesand, Inshore\_or\_Offshore == "Offshore")  
median(Off$TotAbundance) #5

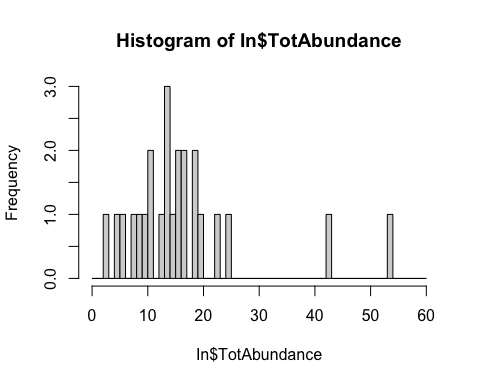
## [1] 5

mean(Off$TotAbundance) #5.4166667

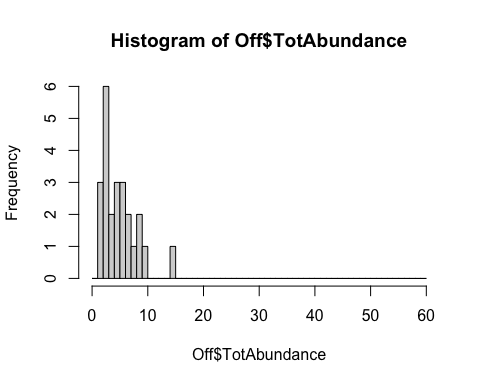
## [1] 5.416667

Next, I visualize the data in another way. I look at a histogram to examine the distribution of prey abundance on the inshore and offshore sides of the sandbar separately. They confirm the above findings.

In <- subset(simplesand, Inshore\_or\_Offshore == "Inshore")  
hist(In$TotAbundance,breaks=seq(0,60,1))



Off <- subset(simplesand, Inshore\_or\_Offshore == "Offshore")  
hist(Off$TotAbundance,breaks=seq(0,60,1))



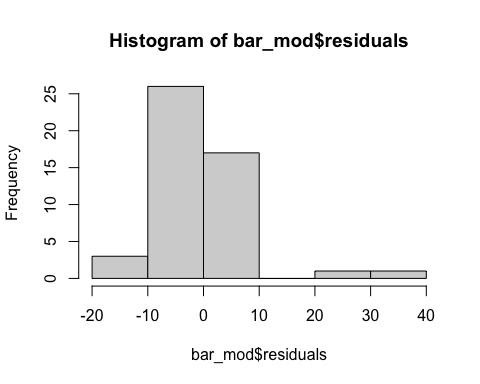
## Determine Model Structure

The next step is to determine the appropriate model structure. I tested model structure in the following order:

* Linear model
* Generalized linear model with poisson distribution
* Generalized linear mixed effects model with poisson distribution
* Generalized linear mixed effects model with negative binomial distribution

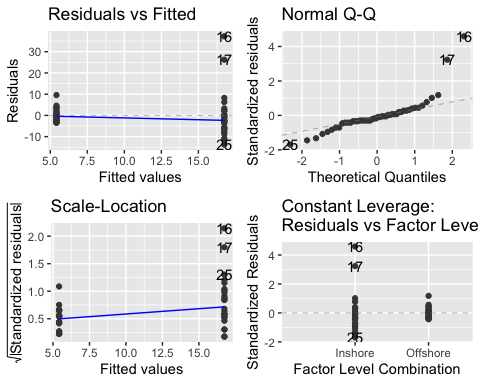
First, I run a linear regression model with prey abundance modeled as a function of sandbar side. As a reminder, sandbar side refers to the inshore vs. offshore side of the sandbar. Then, I check the diagnostics using autoplot and look at a histogram of the residuals.

bar\_mod <- lm(TotAbundance ~ Inshore\_or\_Offshore, data = simplesand)  
hist(bar\_mod$residuals)



autoplot(bar\_mod)

## Warning: `arrange\_()` is deprecated as of dplyr 0.7.0.  
## Please use `arrange()` instead.  
## See vignette('programming') for more help  
## This warning is displayed once every 8 hours.  
## Call `lifecycle::last\_warnings()` to see where this warning was generated.

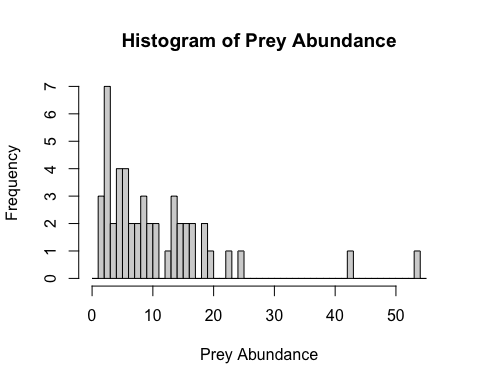
 This data does not meet the assumptions of a linear model. Looking at the diagnostics with autoplot you see that the data does not meet the assumptions of a linear model. For example, the qq plot shows that the residuals are not normally distributed. Instead, they are right-skewed as shown by the histogram.

Here is my interpretation of the autoplot output:

* Residuals vs. Fitted: This checks the assumption that a line is an appropriate fit for our data. There is no clear pattern in the residuals, but there is much greater scatter at right, suggesting that a line is not appropriate.
* Normal Q-Q: This plot indicates that the assumption that residuals are normally distributed is not met. The plot indicates that the residuals are skewed to the right.
* Scale-Location: This plot suggests that the data violates the assumption that variance is similar across the range of X values. There is evidence of heteroscedasticity because the variance of residuals is greater at the right side of the plot.
* Residuals vs. Leverage: There may be some influential data points on the inshore side of the sandbar.

Next, I look at the distribution of the abundance data to get a sense of what error distribution might be most appropriate.

hist(simplesand$TotAbundance,breaks=seq(0,55,1),main="Histogram of Prey Abundance", xlab = "Prey Abundance")



mean(simplesand$TotAbundance) #11.08333

## [1] 11.08333

max(simplesand$TotAbundance)

## [1] 54

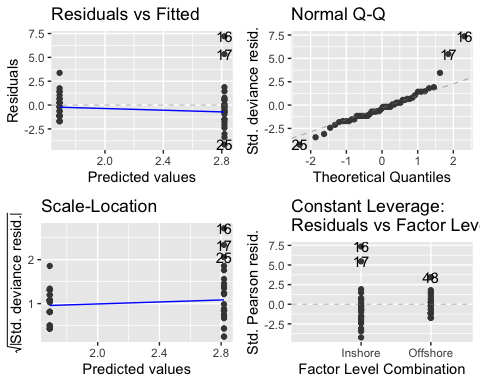
min(simplesand$TotAbundance) #2

## [1] 2

I decide that a Poisson distribution might be a better choice because 1) we are working with count data, 2) it peaks near zero, and 3) it is right skewed. It has a relatively low mean of about 11 prey items.

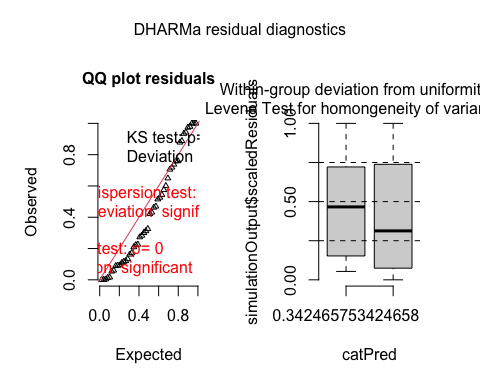
Next, I run a generalized linear model with side of sandbar as the predictor and I use a poisson distribution. I check the diagnostics plots with autoplot() again.

bar\_glm1 <- glm(TotAbundance ~ Inshore\_or\_Offshore, data = simplesand, family = poisson)  
autoplot(bar\_glm1)

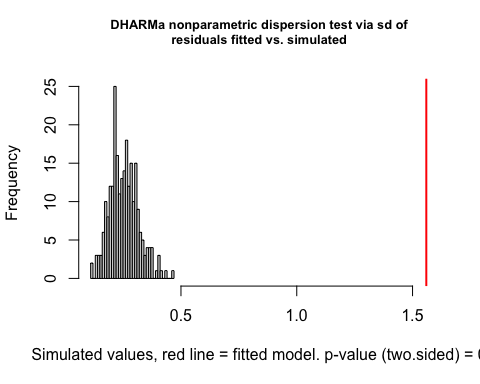
 The autoplot diagnostics results do not look better than those of the linear model, suggesting that this model violates the model assumptions.

To be thorough and consistent with the following models, I also check the diagnostics of the GLM using the DHARMa package.

res\_glmbar1 <- simulateResiduals(fittedModel = bar\_glm1, plot = F)  
plot(res\_glmbar1)



testDispersion(res\_glmbar1)



##   
## DHARMa nonparametric dispersion test via sd of residuals fitted vs.  
## simulated  
##   
## data: simulationOutput  
## dispersion = 6.1393, p-value < 2.2e-16  
## alternative hypothesis: two.sided

For the DHARMa residual diagnostics, the lefthand plot is a qq-plot that detects deviations from the expected distribution. Ideally, the black dots will line up with the red line. The righthand figure plots residuals against the predicted values.

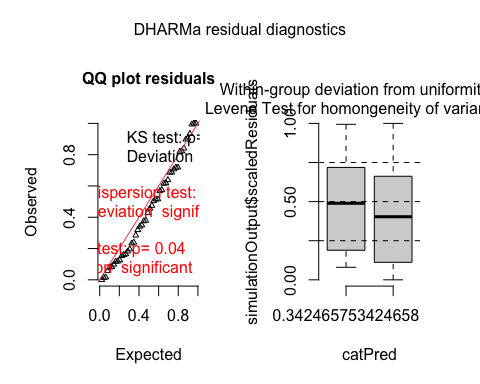
The lefthand QQ plots for our model shows there is high overdispersion, which means that the data is more variable than the distribution of the model. Also, the DHARMa nonparametric dispersion test yields a dispersion value of 6.1 with a p-value of less than 2.2e-16. This value is the ratio of the observed versus the mean simulated variance, and since we have a significant ratio greater than 1, we have evidence of overdispersion.

To account for overdispersion, I add sandbar site to the model as a random intercept. I decide to use site as a random effect because multiple samples were collected from each of four locations on the sandbar, so I expect these samples might be more similar in prey abundance. Now we are working with a generalized linear mixed effects model with a poisson distribution.

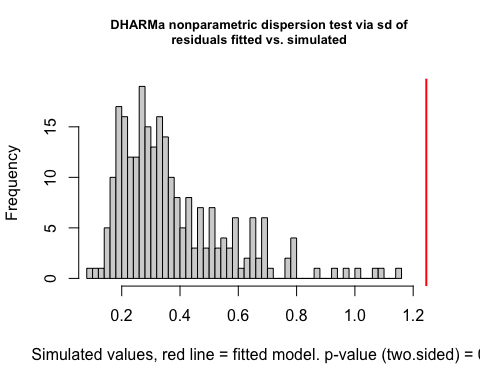
bar\_glm2 <- glmer(TotAbundance ~ Inshore\_or\_Offshore + (1|Point\_on\_Sandbar), data = simplesand, family = poisson) #random intercept model  
summary(bar\_glm2)

## Generalized linear mixed model fit by maximum likelihood (Laplace  
## Approximation) [glmerMod]  
## Family: poisson ( log )  
## Formula: TotAbundance ~ Inshore\_or\_Offshore + (1 | Point\_on\_Sandbar)  
## Data: simplesand  
##   
## AIC BIC logLik deviance df.resid   
## 353.8 359.4 -173.9 347.8 45   
##   
## Scaled residuals:   
## Min 1Q Median 3Q Max   
## -2.6587 -1.2721 -0.2918 0.8408 6.7418   
##   
## Random effects:  
## Groups Name Variance Std.Dev.  
## Point\_on\_Sandbar (Intercept) 0.06927 0.2632   
## Number of obs: 48, groups: Point\_on\_Sandbar, 4  
##   
## Fixed effects:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) 2.7849 0.1412 19.72 <2e-16 \*\*\*  
## Inshore\_or\_OffshoreOffshore -1.1289 0.1007 -11.21 <2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Correlation of Fixed Effects:  
## (Intr)  
## Inshr\_r\_OfO -0.174

res\_glmbar2 <- simulateResiduals(fittedModel = bar\_glm2, plot = F)  
plot(res\_glmbar2)



testDispersion(res\_glmbar2)



##   
## DHARMa nonparametric dispersion test via sd of residuals fitted vs.  
## simulated  
##   
## data: simulationOutput  
## dispersion = 3.322, p-value < 2.2e-16  
## alternative hypothesis: two.sided

As shown in the DHARMa residual diagnostics, this model is an improvement from the generalized linear model, but there is still evidence of overdispersion. In addition to the QQ plot, the DHARMa nonparametric dispersion test yields a dispersion value of 3.3 with a p-value of less than 2.2e-16, indicating overdispersion.

Next, I address the overdispersion problem by using the negative binomial distribution instead of poisson. I fit a generalized linear mixed effects model with a negative binomial distribution. Again, abundance was the response variable, the side of the sandbar was the explanatory variable, and sandbar site was a random intercept.

bar\_glm3 <- glmmTMB(TotAbundance ~ Inshore\_or\_Offshore + (1|Point\_on\_Sandbar), data = simplesand, family = nbinom1)

## Warning in Matrix::sparseMatrix(dims = c(0, 0), i = integer(0), j =  
## integer(0), : 'giveCsparse' has been deprecated; setting 'repr = "T"' for you  
  
## Warning in Matrix::sparseMatrix(dims = c(0, 0), i = integer(0), j =  
## integer(0), : 'giveCsparse' has been deprecated; setting 'repr = "T"' for you  
  
## Warning in Matrix::sparseMatrix(dims = c(0, 0), i = integer(0), j =  
## integer(0), : 'giveCsparse' has been deprecated; setting 'repr = "T"' for you

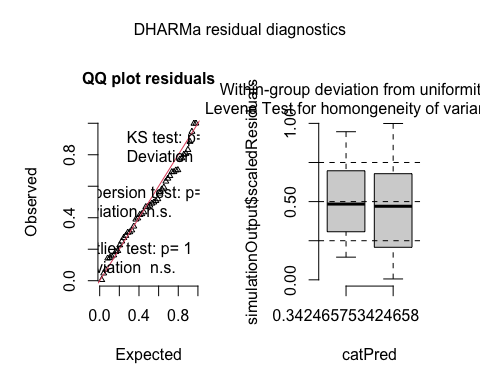
summary(bar\_glm3)

## Family: nbinom1 ( log )  
## Formula: TotAbundance ~ Inshore\_or\_Offshore + (1 | Point\_on\_Sandbar)  
## Data: simplesand  
##   
## AIC BIC logLik deviance df.resid   
## 304.3 311.7 -148.1 296.3 44   
##   
## Random effects:  
##   
## Conditional model:  
## Groups Name Variance Std.Dev.  
## Point\_on\_Sandbar (Intercept) 0.02543 0.1595   
## Number of obs: 48, groups: Point\_on\_Sandbar, 4  
##   
## Overdispersion parameter for nbinom1 family (): 2.35   
##   
## Conditional model:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) 2.7732 0.1219 22.750 < 2e-16 \*\*\*  
## Inshore\_or\_OffshoreOffshore -1.0019 0.1753 -5.715 1.09e-08 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

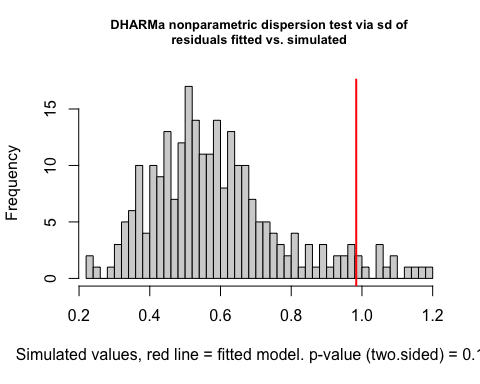
res\_glmbar3 <- simulateResiduals(fittedModel = bar\_glm3, plot = T)

## Warning in Matrix::sparseMatrix(dims = c(0, 0), i = integer(0), j =  
## integer(0), : 'giveCsparse' has been deprecated; setting 'repr = "T"' for you  
  
## Warning in Matrix::sparseMatrix(dims = c(0, 0), i = integer(0), j =  
## integer(0), : 'giveCsparse' has been deprecated; setting 'repr = "T"' for you  
  
## Warning in Matrix::sparseMatrix(dims = c(0, 0), i = integer(0), j =  
## integer(0), : 'giveCsparse' has been deprecated; setting 'repr = "T"' for you  
  
## Warning in Matrix::sparseMatrix(dims = c(0, 0), i = integer(0), j =  
## integer(0), : 'giveCsparse' has been deprecated; setting 'repr = "T"' for you  
  
## Warning in Matrix::sparseMatrix(dims = c(0, 0), i = integer(0), j =  
## integer(0), : 'giveCsparse' has been deprecated; setting 'repr = "T"' for you  
  
## Warning in Matrix::sparseMatrix(dims = c(0, 0), i = integer(0), j =  
## integer(0), : 'giveCsparse' has been deprecated; setting 'repr = "T"' for you

plot(res\_glmbar3)



testDispersion(res\_glmbar3)



##   
## DHARMa nonparametric dispersion test via sd of residuals fitted vs.  
## simulated  
##   
## data: simulationOutput  
## dispersion = 1.6665, p-value = 0.104  
## alternative hypothesis: two.sided

As shown by the DHARMa residual diagnostics plots, this model does not exhibit overdispersion. The residuals line up well with the red line of the expected distribution in the lefthand qqplot. This is confirmed by the DHARMa nonparametric dispersion test which yields a dispersion value of 1.67 and a p-value of 0.104. The righthand figure shows that there are no simulation outliers. Also, the empirical quantiles (solid lines) and theoretical quantiles (dashed lines) appear reasonably similar, suggesting that the expected quantiles do not significantly deviate.

I also check for zero-inflation (testZeroInflation). Zero-inflation is a type of overdispersion in which there are more zeros in your observed data than expected under the fitted model. First, I check for zero-inflation by using DHARMa. This test compares the observed versus the expected number of zeros. Another way to check for zero-inflation is to fit a model with zero-inflation and compare it to the model without zero-inflation. I did this as well.

testZeroInflation(res\_glmbar3)



##   
## DHARMa zero-inflation test via comparison to expected zeros with  
## simulation under H0 = fitted model  
##   
## data: simulationOutput  
## ratioObsSim = 0, p-value = 0.52  
## alternative hypothesis: two.sided

zeromod2 <- glmmTMB(TotAbundance ~ Inshore\_or\_Offshore + (1|Point\_on\_Sandbar), ziformula = ~1, data = simplesand, family = nbinom1)

## Warning in Matrix::sparseMatrix(dims = c(0, 0), i = integer(0), j =  
## integer(0), : 'giveCsparse' has been deprecated; setting 'repr = "T"' for you  
  
## Warning in Matrix::sparseMatrix(dims = c(0, 0), i = integer(0), j =  
## integer(0), : 'giveCsparse' has been deprecated; setting 'repr = "T"' for you  
  
## Warning in Matrix::sparseMatrix(dims = c(0, 0), i = integer(0), j =  
## integer(0), : 'giveCsparse' has been deprecated; setting 'repr = "T"' for you

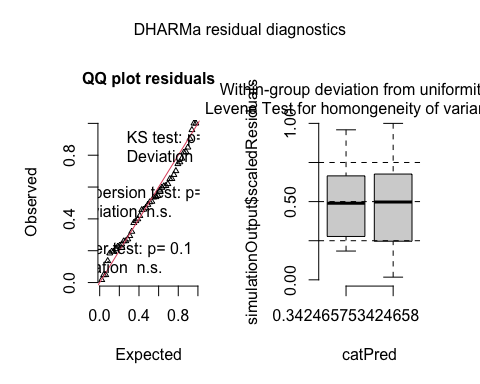
summary(zeromod2)

## Family: nbinom1 ( log )  
## Formula: TotAbundance ~ Inshore\_or\_Offshore + (1 | Point\_on\_Sandbar)  
## Zero inflation: ~1  
## Data: simplesand  
##   
## AIC BIC logLik deviance df.resid   
## 306.3 315.6 -148.1 296.3 43   
##   
## Random effects:  
##   
## Conditional model:  
## Groups Name Variance Std.Dev.  
## Point\_on\_Sandbar (Intercept) 0.02543 0.1595   
## Number of obs: 48, groups: Point\_on\_Sandbar, 4  
##   
## Overdispersion parameter for nbinom1 family (): 2.35   
##   
## Conditional model:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) 2.7732 0.1219 22.750 < 2e-16 \*\*\*  
## Inshore\_or\_OffshoreOffshore -1.0019 0.1753 -5.715 1.09e-08 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Zero-inflation model:  
## Estimate Std. Error z value Pr(>|z|)  
## (Intercept) -21.51 6780.30 -0.003 0.997

simulationz <- simulateResiduals(fittedModel = zeromod2)

## Warning in Matrix::sparseMatrix(dims = c(0, 0), i = integer(0), j =  
## integer(0), : 'giveCsparse' has been deprecated; setting 'repr = "T"' for you  
  
## Warning in Matrix::sparseMatrix(dims = c(0, 0), i = integer(0), j =  
## integer(0), : 'giveCsparse' has been deprecated; setting 'repr = "T"' for you  
  
## Warning in Matrix::sparseMatrix(dims = c(0, 0), i = integer(0), j =  
## integer(0), : 'giveCsparse' has been deprecated; setting 'repr = "T"' for you  
  
## Warning in Matrix::sparseMatrix(dims = c(0, 0), i = integer(0), j =  
## integer(0), : 'giveCsparse' has been deprecated; setting 'repr = "T"' for you  
  
## Warning in Matrix::sparseMatrix(dims = c(0, 0), i = integer(0), j =  
## integer(0), : 'giveCsparse' has been deprecated; setting 'repr = "T"' for you  
  
## Warning in Matrix::sparseMatrix(dims = c(0, 0), i = integer(0), j =  
## integer(0), : 'giveCsparse' has been deprecated; setting 'repr = "T"' for you

plot(simulationz)



#compare AIC for models with and without zero-inflation  
AIC(bar\_glm3,zeromod2)

## df AIC  
## bar\_glm3 4 304.2522  
## zeromod2 5 306.2522

There is no evidence of zero-inflation. For the zero-inflation test, we get a value of 0 and a p-value of 0.52, which indicates that the observed data has less zeros than expected. In other words, we do not have evidence of zero-inflation. A comparison of the models with and without zero-inflation shows that they receive similar support, as the deltaAIC is 2. However, the model without zero-inflation is the better model. Also, it is worthwhile to not that my data has no zeros because I added all four prey taxa together.

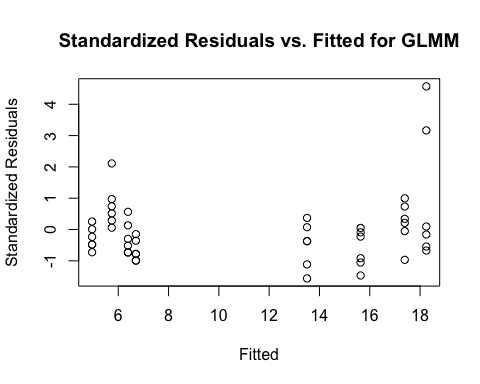
Next, I check for heteroscedasticity. I plot the standardized residuals against the fitted values. It looks decent because the variance is relatively consistent across the x-axis. I also plot the standardized residuals against the predictor variable, which confirms that there is similar spread in the residuals for inshore vs. offshore.

res\_bar\_glm3 <- residuals(bar\_glm3, type='pearson')

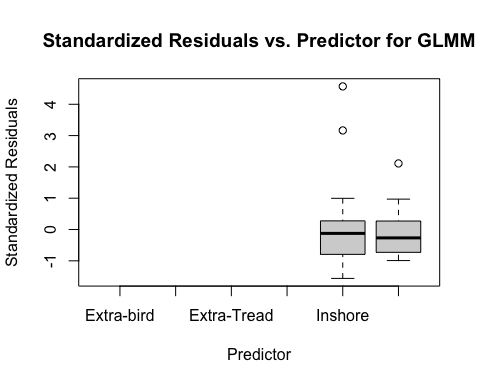
## Warning in Matrix::sparseMatrix(dims = c(0, 0), i = integer(0), j =  
## integer(0), : 'giveCsparse' has been deprecated; setting 'repr = "T"' for you  
  
## Warning in Matrix::sparseMatrix(dims = c(0, 0), i = integer(0), j =  
## integer(0), : 'giveCsparse' has been deprecated; setting 'repr = "T"' for you  
  
## Warning in Matrix::sparseMatrix(dims = c(0, 0), i = integer(0), j =  
## integer(0), : 'giveCsparse' has been deprecated; setting 'repr = "T"' for you  
  
## Warning in Matrix::sparseMatrix(dims = c(0, 0), i = integer(0), j =  
## integer(0), : 'giveCsparse' has been deprecated; setting 'repr = "T"' for you  
  
## Warning in Matrix::sparseMatrix(dims = c(0, 0), i = integer(0), j =  
## integer(0), : 'giveCsparse' has been deprecated; setting 'repr = "T"' for you  
  
## Warning in Matrix::sparseMatrix(dims = c(0, 0), i = integer(0), j =  
## integer(0), : 'giveCsparse' has been deprecated; setting 'repr = "T"' for you

plot(res\_bar\_glm3~fitted(bar\_glm3),main="Standardized Residuals vs. Fitted for GLMM", xlab = "Fitted", ylab = "Standardized Residuals")

## Warning in Matrix::sparseMatrix(dims = c(0, 0), i = integer(0), j =  
## integer(0), : 'giveCsparse' has been deprecated; setting 'repr = "T"' for you  
  
## Warning in Matrix::sparseMatrix(dims = c(0, 0), i = integer(0), j =  
## integer(0), : 'giveCsparse' has been deprecated; setting 'repr = "T"' for you  
  
## Warning in Matrix::sparseMatrix(dims = c(0, 0), i = integer(0), j =  
## integer(0), : 'giveCsparse' has been deprecated; setting 'repr = "T"' for you



plot(res\_bar\_glm3~simplesand$Inshore\_or\_Offshore,main="Standardized Residuals vs. Predictor for GLMM", xlab = "Predictor", ylab = "Standardized Residuals")



**After performing these diagnostics tests, I decided that this model structure better fits the data and I decided to use the GLMM with the negative binomial family for model selection.**

##Compare to Null Model

The next step is to fit the corresponding null model. I choose the null model because I am asking the question: **Are there differences in prey abundance on the inshore vs. offshore side of sandbars?** Then, I compared the models using AIC. The deltaAIC is greater than 10, thus providing substantially more evidence in support of the alternative model (AIC 304) than the null model (AIC 328). I decide to use the generalized linear mixed effects model with a negative binomial distribution and with abundance as the response variable, the side of the sandbar as the explanatory variable, and sandbar site as a random intercept.

bar\_glm5 <- glmmTMB(TotAbundance ~ 1 + (1|Point\_on\_Sandbar), data = simplesand, family = nbinom1)

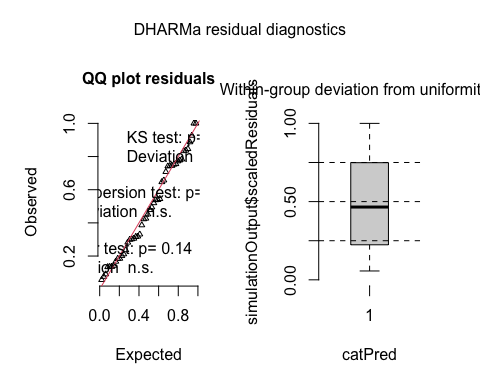
## Warning in Matrix::sparseMatrix(dims = c(0, 0), i = integer(0), j =  
## integer(0), : 'giveCsparse' has been deprecated; setting 'repr = "T"' for you  
  
## Warning in Matrix::sparseMatrix(dims = c(0, 0), i = integer(0), j =  
## integer(0), : 'giveCsparse' has been deprecated; setting 'repr = "T"' for you  
  
## Warning in Matrix::sparseMatrix(dims = c(0, 0), i = integer(0), j =  
## integer(0), : 'giveCsparse' has been deprecated; setting 'repr = "T"' for you

summary(bar\_glm5)

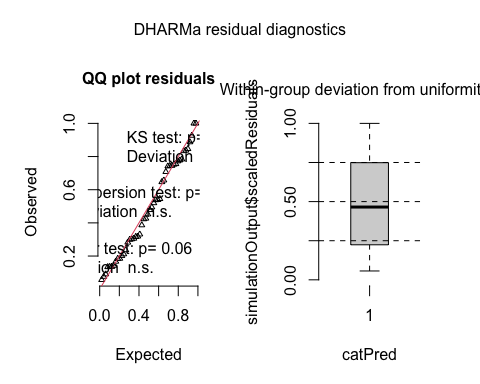
## Family: nbinom1 ( log )  
## Formula: TotAbundance ~ 1 + (1 | Point\_on\_Sandbar)  
## Data: simplesand  
##   
## AIC BIC logLik deviance df.resid   
## 328.3 333.9 -161.1 322.3 45   
##   
## Random effects:  
##   
## Conditional model:  
## Groups Name Variance Std.Dev.   
## Point\_on\_Sandbar (Intercept) 8.489e-10 2.914e-05  
## Number of obs: 48, groups: Point\_on\_Sandbar, 4  
##   
## Overdispersion parameter for nbinom1 family (): 5.5   
##   
## Conditional model:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) 2.4054 0.1105 21.76 <2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

res\_glmbar5 <- simulateResiduals(fittedModel = bar\_glm5, plot = T)

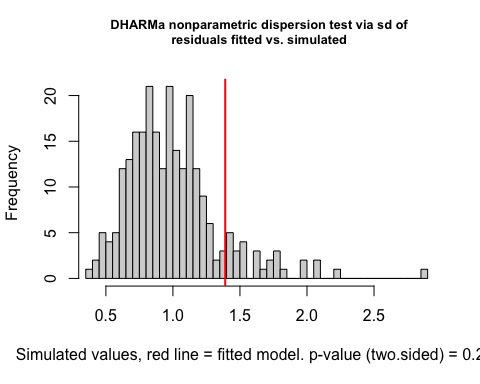
## Warning in Matrix::sparseMatrix(dims = c(0, 0), i = integer(0), j =  
## integer(0), : 'giveCsparse' has been deprecated; setting 'repr = "T"' for you  
  
## Warning in Matrix::sparseMatrix(dims = c(0, 0), i = integer(0), j =  
## integer(0), : 'giveCsparse' has been deprecated; setting 'repr = "T"' for you  
  
## Warning in Matrix::sparseMatrix(dims = c(0, 0), i = integer(0), j =  
## integer(0), : 'giveCsparse' has been deprecated; setting 'repr = "T"' for you  
  
## Warning in Matrix::sparseMatrix(dims = c(0, 0), i = integer(0), j =  
## integer(0), : 'giveCsparse' has been deprecated; setting 'repr = "T"' for you  
  
## Warning in Matrix::sparseMatrix(dims = c(0, 0), i = integer(0), j =  
## integer(0), : 'giveCsparse' has been deprecated; setting 'repr = "T"' for you  
  
## Warning in Matrix::sparseMatrix(dims = c(0, 0), i = integer(0), j =  
## integer(0), : 'giveCsparse' has been deprecated; setting 'repr = "T"' for you



plot(res\_glmbar5)



testDispersion(res\_glmbar5)



##   
## DHARMa nonparametric dispersion test via sd of residuals fitted vs.  
## simulated  
##   
## data: simulationOutput  
## dispersion = 1.3937, p-value = 0.232  
## alternative hypothesis: two.sided

#Compare models.  
AIC(bar\_glm3,bar\_glm5)

## df AIC  
## bar\_glm3 4 304.2522  
## bar\_glm5 3 328.2952

## Interpret Results

Finally, I interpret the selected model.

summary(bar\_glm3)

## Family: nbinom1 ( log )  
## Formula: TotAbundance ~ Inshore\_or\_Offshore + (1 | Point\_on\_Sandbar)  
## Data: simplesand  
##   
## AIC BIC logLik deviance df.resid   
## 304.3 311.7 -148.1 296.3 44   
##   
## Random effects:  
##   
## Conditional model:  
## Groups Name Variance Std.Dev.  
## Point\_on\_Sandbar (Intercept) 0.02543 0.1595   
## Number of obs: 48, groups: Point\_on\_Sandbar, 4  
##   
## Overdispersion parameter for nbinom1 family (): 2.35   
##   
## Conditional model:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) 2.7732 0.1219 22.750 < 2e-16 \*\*\*  
## Inshore\_or\_OffshoreOffshore -1.0019 0.1753 -5.715 1.09e-08 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

exp(-1.0019) #0.3671811

## [1] 0.3671811

This model indicates that the side of the sandbar has a strong effect on prey abundance with a p-value less than 0.001. The slope for the side of the sandbar is -1.0019 (on the log scale), indicating that the natural log of prey abundance for the offshore side is on average 1.0019 less than that of the inshore side. When I account for the log link, the slope of the model is 0.3671811, indicating that prey abundance for the offshore side of the sandbar is on average a factor of about 0.37 times less than the prey abundance on the inshore side of the sandbar.

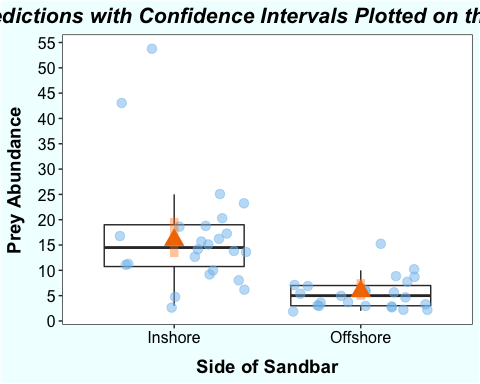
The random intercept is normally distributed with mean 0 and variance 0.02543. In other words, different sites have different prey abundances.

This finding matches our expectations based on our initial data pictures.

## Plot Data Again

Lastly, I plot the model prediction (shown by the orange triangles) back onto the data with confidence intervals for the fitted value of y (shown by the orange lines). The blue circles indicate the original data points.

newdat2 <- expand.grid(Inshore\_or\_Offshore=c("Inshore","Offshore"))  
#design matrix (fixed effects)  
mm2 <- model.matrix(delete.response(terms(bar\_glm3)),newdat2)  
#Linear predictor, back-transform this with the inverse link function  
newdat2$TotAbundance <- drop(mm2 %\*% fixef(bar\_glm3)[["cond"]])  
predvar2 <- diag(mm2 %\*% vcov(bar\_glm3)[["cond"]] %\*% t(mm2))  
newdat2$SE <- sqrt(predvar2)  
newdat\_transform2 <- mutate(newdat2, ci\_lwr = TotAbundance - (1.96 \* SE), ci\_upr = TotAbundance + (1.96 \* SE))  
newdat\_transform2 <- mutate(newdat\_transform2, TotAbundance = exp(TotAbundance), ci\_lwr = exp(ci\_lwr), ci\_upr = exp(ci\_upr))  
#plot  
pd <- position\_dodge(width = 0.4)  
g3 <- ggplot(simplesand, aes(x=Inshore\_or\_Offshore, y=TotAbundance))+  
 geom\_boxplot(outlier.shape = NA) + #hides outliers  
 geom\_jitter(alpha=0.5,color = "skyblue2",size=3)  
g4 <- g3 +  
 geom\_point(data=newdat\_transform2, shape=17, size=5, position=pd, color="darkorange2")   
#confidence intervals  
g5 <- g4 +   
 geom\_linerange(data=newdat\_transform2, aes(ymin=ci\_lwr, ymax=ci\_upr),  
 lwd=3, position=pd, color = "darkorange1", alpha = 0.4) + #alpha controls opacity  
 theme\_bw() +  
 ggtitle('Model Predictions with Confidence Intervals Plotted on the Original Data') +  
 xlab("Side of Sandbar") +   
 ylab("Prey Abundance") +  
 theme(plot.title = element\_text(color="black", size=16, face="bold.italic", hjust = 0.5),  
 axis.title.x = element\_text(color="black", size=14, face="bold", margin = margin(t = 10, r = 0, b = 0, l = 0)),  
 axis.title.y = element\_text(color="black", size=14, face="bold", margin = margin(t = 0, r = 10, b = 0, l = 0)),  
 panel.background = element\_rect(fill = "white"),  
 axis.text.x = element\_text(color="black", size=12),  
 axis.text.y = element\_text(color="black", size=12),  
 # panel.border = element\_blank(),  
 panel.grid.major = element\_blank(),  
 panel.grid.minor = element\_blank(),  
 plot.background = element\_rect(fill = "azure")) +  
 scale\_y\_continuous(breaks=seq(0,55,5))  
g5

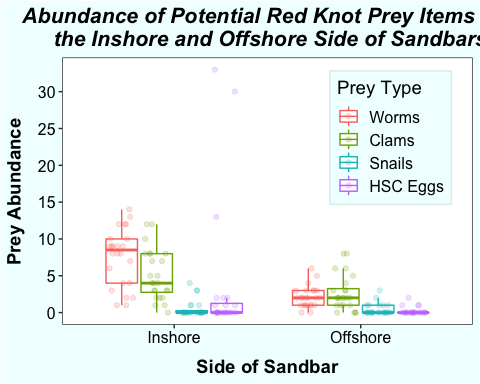
 As you can see, the model lines up quite well with what I would expect given the original data picture. I conclude that the side of the sandbar influences prey abundance.

But I have one further question: **Is this trend driven by any particular type of prey?**

To address this question, I plot the abundance of the four prey types (worms, clams, snails, and horseshoe crab eggs) against the inshore vs. offshore side of the sandbar. Based on this visual assessment, the trend appears to be driven by annelids, clams, and HSC eggs, which are more abundant on the inshore side of the sandbar. However, there are more snails on average on the offshore side of the sandbar.

sandbar\_prey %>% group\_by(Point\_on\_Sandbar, Inshore\_or\_Offshore, Time\_simple, Taxa) %>%  
 summarize(TotAbundance = sum(Abundance)) %>%  
 ggplot(data=, aes(x = Inshore\_or\_Offshore, y = TotAbundance, color=Taxa)) +  
 geom\_boxplot(outlier.shape = NA) +  
 geom\_point(alpha = 0.2, position = position\_jitterdodge(jitter.width = 1)) +  
 theme\_bw() +  
 ggtitle('Abundance of Potential Red Knot Prey Items on \n the Inshore and Offshore Side of Sandbars') +  
 xlab("Side of Sandbar") +   
 ylab("Prey Abundance") +  
 scale\_color\_discrete(name="Prey Type", labels = c("Worms", "Clams", "Snails", "HSC Eggs")) +  
 theme(plot.title = element\_text(color="black", size=16, face="bold.italic", hjust = 0.5),  
 axis.title.x = element\_text(color="black", size=14, face="bold", margin = margin(t = 10, r = 0, b = 0, l = 0)),  
 axis.title.y = element\_text(color="black", size=14, face="bold", margin = margin(t = 0, r = 10, b = 0, l = 0)),  
 axis.text.x = element\_text(color="black", size=12),  
 axis.text.y = element\_text(color="black", size=12),  
 legend.position = c(0.8,0.7),   
 legend.title = element\_text(size = 14),  
 legend.text = element\_text(color = "black", size = 12),  
 legend.background = element\_rect(fill = "azure", size=0.5, linetype="solid",   
 colour ="azure2"),  
 legend.key = element\_rect(fill="azure", color = NA),  
 panel.background = element\_rect(fill = "white"),  
 panel.grid.major = element\_blank(),  
 panel.grid.minor = element\_blank(),  
 plot.background = element\_rect(fill = "azure")) +  
 scale\_y\_continuous(breaks=seq(0,55,5))

## `summarise()` regrouping output by 'Point\_on\_Sandbar', 'Inshore\_or\_Offshore', 'Time\_simple' (override with `.groups` argument)

 I dig a little deeper by comparing the means for each prey type and the side of the sandbar.

sandbar\_prey %>% group\_by(Point\_on\_Sandbar, Inshore\_or\_Offshore, Time\_simple, Taxa) %>%  
 summarize(TotAbundance = sum(Abundance)) %>%  
 group\_by(Taxa, Inshore\_or\_Offshore) %>%  
 summarize(MeanAb = mean(TotAbundance))

## `summarise()` regrouping output by 'Point\_on\_Sandbar', 'Inshore\_or\_Offshore', 'Time\_simple' (override with `.groups` argument)

## `summarise()` regrouping output by 'Taxa' (override with `.groups` argument)

## # A tibble: 8 x 3  
## # Groups: Taxa [4]  
## Taxa Inshore\_or\_Offshore MeanAb  
## <fct> <fct> <dbl>  
## 1 Annelid Inshore 7.58   
## 2 Annelid Offshore 2.12   
## 3 Clams Inshore 5.08   
## 4 Clams Offshore 2.67   
## 5 Gastropods Inshore 0.542  
## 6 Gastropods Offshore 0.417  
## 7 HSC\_egg Inshore 3.54   
## 8 HSC\_egg Offshore 0.208

This further confirms the patterns observed in the above figure. Regardless of the side of the sandbar, snails (gastropod) were the least abundant prey type. On average, worms (annelid) were the most abundant prey type, although clams were slightly more abundant on the offshore side of sandbars. Generally, all prey types were more abundant on the inshore side of the sandbar.

# Discussion

This analysis shows that potential prey items for the rufa red knot are more abundant on the inshore side of sandbars compared to the offshore side in the Cape Shore region of Delaware Bay. This finding has implications for rufa red knot foraging, as well as our sampling design. I expect rufa red knots to spend more time foraging on the inshore side of sandbars during the ebbing tide. In our upcoming field surveys, I also expect to find a greater abundance of rufa red knots foraging on the inshore side of the sandbars. These results will inform our sediment sampling efforts this spring as we assess the distribution of horseshoe crab eggs and other prey items throughout the tidal flats in relation to oyster farms. Since prey abundances and potentially also prey composition are different on the inshore and offshore sides of sandbars, we will collect sediment core samples on both sides.

This analysis does not suggest what potential mechanisms may determine the distribution of prey items on the inshore and offshore sides of the sandbar. It is possible that the physical structure of the sandbar acts as a barrier, trapping meiofaunal organisms on the inshore side of the sandbar as the tide goes out. Alternatively, these meiofaunal organisms may prefer the inshore side of the sandbars as it could provide greater protection from wave action.

The analysis suggests that worms, clams, and horseshoe crabs eggs are more abundant on the inshore side of the sandbar. On average, however, snails are more abundant on the offshore side (0.4 snails) of the sandbar compared to the inshore side (0.5 snails). Although presumably all of the prey items considered in this analysis could be transported via wave action, snails are the only actively mobile organism. On the surface of the tidal flats, you are able to see the tracks that the snails make in the sediment. On the other hand, worms and clams burrow vertically in the sediment and the dispersal of horseshoe crab eggs is driven by wave action. Since snails are mobile scavengers, perhaps there are more snails on the offshore side of the sandbar because they follow the outgoing tide. Alternatively, the finding that there were more snails on the offshore side of the sandbar could be due to the fact that the sample size is low and many samples contained 0 snails.

It is important to note that this sampling effort is preliminary. The conclusions drawn from the resulting data are limited by several factors. First, due to the preliminary nature of the fieldwork, the sample size is low and the samples were collected from relatively few locations within the study site, restricting our ability to generalize beyond the sandbar of study. Additionally, rufa red knots had migrated north from Delaware Bay by the time these sediment samples were collected on June 2, 2020. Shorebirds may have depleted the prey resources in the study area and altered the distribution and relative abundance by the time of sample collection. Nevertheless, this sampling effort was useful for picking up likely trends at the sandbar-level and will inform our future sampling efforts.

In conclusion, we found that red knot prey items are more abundnant on the inshore side of sandbars than the offshore side. This finding informs our understanding of prey distribution throughout the tidal flats and will inform our future intensive sampling efforts. In addition, it helps us generate hypotheses about where we expect red knots to forage in the tidal flats.