Texas 2022 Analysis

Kirsten Sheehy

2024-05-16

Overview

The following script cleans and analyzes data from Texas 2022. Fish were collected by Kirsten Sheehy and Jon Aguiñaga. Behavioral data and fish lengths were extracted from videos and photos by Nishika Raghavan. Parasite data were collected by Dr. Jessica Stephenson's lab.

Packages to Load

```
library(dplyr)
library(readr)
library(tidyr)
library(tibble)
library(lubridate)
library(tidyverse)
library(ggplot2)
library(lme4)
library(pscl)
```

```
## Warning: package 'pscl' was built under R version 4.3.2
```

```
library(MASS)
library(lmtest)
library(here)
library(knitr)
library(DHARMa)
```

```
## Warning: package 'DHARMa' was built under R version 4.3.3
```

```
library(glmmTMB)
```

```
## Warning in checkDepPackageVersion(dep_pkg = "TMB"): Package version inconsistency detected.
## glmmTMB was built with TMB version 1.9.6
## Current TMB version is 1.9.10
## Please re-install glmmTMB from source or restore original 'TMB' package (see '?reinstalling' for mor
```

```
library(performance)
```

```
## Warning: package 'performance' was built under R version 4.3.3
```

```
library(emmeans)
```

```
## Warning: package 'emmeans' was built under R version 4.3.2
```

Raw Data

```
parasite_data <- read.csv(here("data", "copy_RAW_parasite_data_20230428.csv"))
length_data <- read.csv(here("data", "copy_MERGE_NRkas_TexasFishMeasurements_20250702.csv"))
boris_data <- read.csv(here("data", "copy_RAW_Texas_BORISdata_20250617.csv"))
id_data <- read.csv(here("data", "copy_RAW_trial_ID_data_completeonly_20240220.csv"))
```

Tidy Data

Parasite Data

```
# rename columns to be consistent across data sets
parasite_data <- parasite_data %>% dplyr::rename(
  fish.id = fish.id,
  site.id = collection.site
)

# change collection.date and dissection.date to a date format (YYYY-MM-DD)
parasite_data$collection.date <- as.Date(parasite_data$collection.date,
  format = "%m/%d/%Y"
)
# some dates in the dissection date column are formatted M/D/YY or MM/DD/YY instead of MM/DD/YYYY.
# this code finds and fixes those years to match the YYYY format before we use as.Date.
parasite_data$dissection.date <- parse_date_time(parasite_data$dissection.date, orders = c("mdy", "dmy"
# this gets rid of the time stamps
parasite_data$dissection.date <- as.Date(parasite_data$dissection.date,
  format = "%m/%d/%Y"
)

# change site names from abbreviation (WES, BR) to full (Weslaco, Brownsville)
parasite_data$site.id <- gsub("WES", "Weslaco", parasite_data$site.id)
parasite_data$site.id <- gsub("BR-OP", "Brownsville", parasite_data$site.id)

# note: the 'OP' in Brownsville stands for 'overpass'. We explored several sites
# in Brownsville, but only used the ones from the overpass for this study, so I
# simplified the name to just 'Brownsville'.

# checking and fixing data structure
str(parasite_data)
```

```
## 'data.frame':    256 obs. of  14 variables:
##  $ collection.date: Date, format: NA "0022-08-06" ...
##  $ dissection.date: Date, format: NA "2023-02-16" ...
##  $ site.id        : chr  "" "Weslaco" "" "Weslaco" ...
##  $ fish.id        : chr  "" "P.formosa" "" "P.formosa" ...
##  $ species        : chr  "" "formosa" "" "formosa" ...
##  $ sex            : chr  "" "F" "" "F" ...
##  $ sex.label      : chr  "" "not listed" "" "not listed" ...
##  $ sex.species    : chr  "" "formosaF" "" "formosaF" ...
##  $ tremr          : int  NA 0 NA 0 NA 0 NA 1 NA 1 ...
##  $ treml          : int  NA 0 NA 1 NA 0 NA 0 NA 2 ...
##  $ unk            : int  NA 0 NA 0 NA 2 NA 3 NA 0 ...
##  $ totalpara      : int  NA 0 NA 1 NA 2 NA 4 NA 3 ...
##  $ notes          : chr  "" "well preserved" "" "" ...
```

```
##  $ label.notes    : chr  "" "" "" "no fish number or sex on the label with the specimen " ...
```

```
parasite_data$site.id <- as.factor(parasite_data$site.id)
parasite_data$fish.id <- as.factor(parasite_data$fish.id)
parasite_data$species <- as.factor(parasite_data$species)
parasite_data$sex <- as.factor(parasite_data$sex)
parasite_data$sex.label <- as.factor(parasite_data$sex.label)
parasite_data$sex.species <- as.factor(parasite_data$sex.species)
str(parasite_data)
```

```
## 'data.frame':    256 obs. of  14 variables:
##  $ collection.date: Date, format: NA "0022-08-06" ...
##  $ dissection.date: Date, format: NA "2023-02-16" ...
##  $ site.id        : Factor w/ 3 levels "","Brownsville",..: 1 3 1 3 1 3 1 3 1 3 ...
##  $ fish.id        : Factor w/ 122 levels "","P.formosa",..: 1 2 1 2 1 2 1 2 1 3 ...
##  $ species        : Factor w/ 3 levels "","formosa","latipinna": 1 2 1 2 1 2 1 2 1 3 ...
##  $ sex            : Factor w/ 3 levels "","F","M": 1 2 1 2 1 2 1 2 1 NA ...
##  $ sex.label      : Factor w/ 4 levels "","F","M","not listed": 1 4 1 4 1 4 1 4 1 4 ...
##  $ sex.species    : Factor w/ 5 levels "","formosaF",..: 1 2 1 2 1 2 1 2 1 5 ...
##  $ tremr          : int  NA 0 NA 0 NA 0 NA 1 NA 1 ...
##  $ treml          : int  NA 0 NA 1 NA 0 NA 0 NA 2 ...
##  $ unk            : int  NA 0 NA 0 NA 2 NA 3 NA 0 ...
##  $ totalpara      : int  NA 0 NA 1 NA 2 NA 4 NA 3 ...
##  $ notes          : chr  "" "well preserved" "" "" ...
##  $ label.notes    : chr  "" "" "" "no fish number or sex on the label with the specimen " ...
```

```
# remove blank rows (empty rows between each entry in original sheet from Jessica, probably for ease of
parasite_data <- parasite_data %>%
  tidyr::drop_na(collection.date)
```

Length Data

NOTE: Nishika redid these measurements in QuPath. For each image, there are three measurements: standard, total, and one labeled as the name of the file/fish.id. The file/fish.id is just the measurement we used to set the scale. Nishika measured a centimeter on the ruler in each photo, then set those pixels to equal 10000. This means that for every 10000, we have 1cm. This checks out with the measurements in the file (e.g. 30328.4 = 3.03cm). This also checks out with going back and eyeballing some measurements from random photos. Standard length is from the mouth of the fish to the caudal peduncle. Total length is from the mouth to the tip of the tail.

```
# renaming the columns
length_data <- length_data %>% dplyr::rename(
  file.name = Image,
  length = Length.µm
)

# need to pivot wider so that the fish.id, standard, and total length measurements are in their own col
length_data <- length_data %>%
  mutate(fish.id = if_else(str_detect(Name, "_"), Name, NA_character_)) %>%
  fill(fish.id) %>% # Fill down the fish.id so each standard/total gets its fish
  filter(Name != fish.id) %>% # Remove the rows where Name == fish.id (we already stored them)
  pivot_wider(names_from = Name, values_from = length) # Reshape wider

# split the fish.id column into site.id, date, fish.id, and sex
length_data <- length_data %>%
  tidyr::separate_wider_delim(fish.id,
```

```
    delim = "_",
    names = c(
      "site.id",
      "date.collected",
      "fish.id",
      "sex"
    ),
    too_few = "align_start"
  )

# make the fish.id, site.id, and sex format match the rest of the data
length_data$fish.id <- gsub("pf", "PF", length_data$fish.id)
length_data$fish.id <- gsub("pl", "PL", length_data$fish.id)
length_data$site.id <- gsub("wes", "Weslaco", length_data$site.id)
length_data$site.id <- gsub("brop", "Brownsville", length_data$site.id)
length_data$site.id <- gsub("br-op", "Brownsville", length_data$site.id)
length_data$sex <- gsub("f", "F", length_data$sex)
length_data$sex <- gsub("m", "M", length_data$sex)

# change length and date.collected column names
length_data$standard.length <- length_data$standard
length_data$total.length <- length_data$total
length_data$collection.date <- length_data$date.collected
length_data <- length_data %>%
  dplyr::select(
    -standard,
    -total,
    -date.collected
  )

# formatting collection.date column
length_data$collection.date <- gsub("aug", "-08-", length_data$collection.date)
# two dates are backwards (-08-11 instead of 11-08-), fixing them here
length_data$collection.date <- gsub("-08-11", "11-08-", length_data$collection.date)
length_data$collection.date <- paste0(length_data$collection.date, "2022")
length_data$collection.date <- as.Date(length_data$collection.date, format = "%d-%m-%Y")

# checking and fixing data structure
str(length_data)
```

```
## tibble [133 x 7] (S3: tbl_df/tbl/data.frame)
##  $ file.name      : chr [1:133] "IMG_20220807_155019762.jpg" "IMG_20220807_155214400.jpg" "IMG_20220
##  $ site.id        : chr [1:133] "Brownsville" "Brownsville" "Brownsville" "Brownsville" ...
##  $ fish.id        : chr [1:133] "PL-1" "PL-02" "PF-01" "PL-03" ...
##  $ sex            : chr [1:133] "F" "M" "F" "F" ...
##  $ standard.length: num [1:133] 24986 32053 25353 48183 25212 ...
##  $ total.length   : num [1:133] 31793 35514 29732 53749 29804 ...
##  $ collection.date: Date[1:133], format: "2022-08-06" "2022-08-06" ...
```

```
length_data$file.name <- as.factor(length_data$file.name)
length_data$site.id <- as.factor(length_data$site.id)
length_data$fish.id <- as.factor(length_data$fish.id)
length_data$sex <- as.factor(length_data$sex)
```

ID Data

just need to rename the columns to match other datasets

```
id_data <- id_data %>% dplyr::rename(
  video.id = video.ID,
  fish.id = fish.ID,
  site.id = site.ID,
  trial.id = trial.ID,
  batch.id = batch.ID
)
```

Boris Data

First, I tidy the raw data. I rename columns and remove unnecessary ones.

```
# remove unnecessary columns (largely meta data and unused features in BORIS)
boris_data <- boris_data %>% dplyr::select(
  -Observation.date, # this is just the day processed in BORIS
  -Description,
  -FPS,
  -Behavioral.category,
  -Modifiers,
  -Comment.start,
  -Comment.stop
)

# rename columns (to match up across data)
boris_data <- boris_data %>% dplyr::rename(
  pool = Subject,
  trial.length = Total.length,
  start = Start..s.,
  stop = Stop..s.,
  duration = Duration..s.
)

# split Media.file into columns to extract file name (could also use
# Observation.id, but figured this would help avoid typos made in Boris)
boris_data <- boris_data %>% tidyr::separate_wider_delim(Media.file,
  delim = "/",
  names = c(
    "file1",
    "file2",
    "file3",
    "file4",
    "file5",
    "video.id"
  ),
  too_few = "align_end"
)

# remove the excess filepath columns
boris_data <- boris_data %>% dplyr::select(
  -file1,
  -file2,
```

```
  -file3,
  -file4,
  -file5
)

# video.id (from the file path split above) is the file name of the recording.
# It decomposes into the site ID, trial number, batch, and date recorded. The
# following code duplicates the column, then splits the information in video.id
# into separate columns.
boris_data$video.id.split <- boris_data$video.id
boris_data <- boris_data %>% tidyr::separate_wider_delim(video.id.split,
  delim = "_",
  names = c(
    "site.id",
    "trial.id",
    "batch.id",
    "trial.date"
  )
)

# remove the file type from the trial.date column
boris_data$trial.date <- gsub(".mov", "", boris_data$trial.date)

# change trial.date from (YYYYMMDD) to a date (YYYY-MM-DD)
boris_data$trial.date <- as.Date(boris_data$trial.date, format = "%Y%m%d")

# remove 'trial' from the data entries for trial.ID
boris_data$trial.id <- gsub("trial", "", boris_data$trial.id)
boris_data$trial.id <- gsub("trail", "", boris_data$trial.id) # had to find a few with a typo in the fi

# remove 'pool' from data in pool column
boris_data$pool <- gsub("Pool ", "", boris_data$pool)

# change site ID from abbreviations to full name
# note: doing them in this order is important
boris_data$site.id <- gsub("Wes", "Weslaco", boris_data$site.id)
boris_data$site.id <- gsub("WES", "Weslaco", boris_data$site.id)
boris_data$site.id <- gsub("BR1", "Brownsville", boris_data$site.id)
boris_data$site.id <- gsub("BR2", "Brownsville", boris_data$site.id)
boris_data$site.id <- gsub("BR", "Brownsville", boris_data$site.id)
# note: there are three entry types for Brownsville: BR, BR1, and BR2
# need to revisit lab notebook to confirm, but I believe BR1 and BR2
# are the two sides of the garage (i.e. the two cameras)
```

Now that the columns are all formatted correctly, I need to pull out the behaviors from the Behavior column into their own, separate columns.

```
# start by duplicating the 'Behavior' column twice. This will be used to extract start and stop times o
boris_data <- boris_data %>%
  dplyr::mutate(behavior.start = Behavior)

boris_data <- boris_data %>%
  dplyr::mutate(behavior.stop = Behavior)
```

```
# then pivot_wider with names from behavior.start and values from start
boris_data_wide <- boris_data %>%
  tidyr::pivot_wider(
    names_from = behavior.start,
    values_from = start,
    names_prefix = "start."
  )

# do the same with stop
boris_data_wide <- boris_data_wide %>%
  tidyr::pivot_wider(
    names_from = behavior.stop,
    values_from = stop,
    names_prefix = "stop."
  )

# now, I need to get the duration of each behavior using the start and stop times
boris_data_wide <- boris_data_wide %>%
  tidyr::pivot_wider(
    names_from = Behavior,
    values_from = duration,
    names_prefix = "duration."
  )

# I'll remove stop_Startle and duration_Startle because these are 'points' not 'states' and do not have
boris_data_wide <- boris_data_wide %>% dplyr::select(
  -stop.Startle,
  -duration.Startle
)
```

Now, I need to join the ID_data and boris_data_wide datasets.

```
# I need a column in both ID_data and boris_data_wide to join by
# I'll create a new column that merges the file name (which already includes
# site, trial, and batch) with pool # for both data sets

boris_data_wide$merge.id <- paste(
  boris_data_wide$video.id,
  boris_data_wide$pool
)

id_data$merge.id <- paste(
  id_data$video.id,
  id_data$pool
)

boris_data_merge <- boris_data_wide %>%
  merge(id_data, by = "merge.id")
```

Now we tidy the merged data.

```
# remove duplicate columns
boris_data_merge <- boris_data_merge %>% dplyr::select(
  -merge.id,
  -video.id.y,
```

```
  -pool.y,
  -site.id.y,
  -trial.id.y,
  -batch.id.y,
  -trial.date.y
)

# rename columns to get rid of .x and .y appendages
boris_data_merge <- boris_data_merge %>% dplyr::rename(
  video.id = video.id.x,
  pool = pool.x,
  site.id = site.id.x,
  trial.id = trial.id.x,
  batch.id = batch.id.x,
  trial.date = trial.date.x
)

# add column for species from fish.ID
boris_data_merge <- boris_data_merge %>%
  mutate(species = fish.id)

boris_data_merge <- boris_data_merge %>%
  separate_wider_delim(species,
    delim = "-",
    names = c(
      "species",
      "junk.num"
    )
  )

boris_data_merge <- boris_data_merge %>% dplyr::select(-junk.num)

# filling the 'start.startle' column based on fish and trial ID. This way every row has the startle tim
boris_data_merge <- boris_data_merge %>%
  group_by(fish.id, trial.id) %>%
  fill(start.Startle, .direction = "downup")
```

Now, I need to standardize the trial times. When Nishika and I were observing, we sometimes recorded behaviors for longer than the prescribed 15 minutes. The following code finds the earliest behavior observation (either start.hiding or start.open) and then cuts off any observations after 15 minutes.

```
# new columns with the earliest open and hiding value per fish per trial
boris_data_merge <- boris_data_merge %>%
  group_by(fish.id, trial.id) %>%
  mutate(
    earliest.open = min(start.Open, na.rm = TRUE),
    earliest.hiding = min(start.Hiding, na.rm = TRUE)
  )
```

```
## Warning: There were 33 warnings in `mutate()`.
## The first warning was:
## i In argument: `earliest.open = min(start.Open, na.rm = TRUE)`.
## i In group 15: `fish.id = "PF-28"`, `trial.id = "3"`.
## Caused by warning in `min()`:
## ! no non-missing arguments to min; returning Inf
```

```
## i Run `dplyr::last_dplyr_warnings()` to see the 32 remaining warnings.
```

```r
# for some fish, there was no earliest.open or earliest.hiding time (i.e. they were only in the open or
boris_data_merge$earliest.open <- as.character(boris_data_merge$earliest.open)
boris_data_merge <- boris_data_merge %>%
  mutate(earliest.open = na_if(earliest.open, "Inf"))
boris_data_merge$earliest.open <- as.numeric(boris_data_merge$earliest.open)

boris_data_merge$earliest.hiding <- as.character(boris_data_merge$earliest.hiding)
boris_data_merge <- boris_data_merge %>%
  mutate(earliest.hiding = na_if(earliest.hiding, "Inf"))
boris_data_merge$earliest.hiding <- as.numeric(boris_data_merge$earliest.hiding)

# creates a trial cutoff time by taking the earliest behavior time (either open or closed)
# and adding 1200 seconds (20 minutes) to it
boris_data_merge <- boris_data_merge %>%
  group_by(fish.id, trial.id) %>%
  mutate(
    trial.end = pmin(earliest.open, earliest.hiding, na.rm = TRUE) + 1200
  )

# now, I need to remove all observations per fish per trial that exceed this cutoff time
boris_data_cutoff <- boris_data_merge %>%
  group_by(fish.id, trial.id) %>%
  filter(start.Open <= trial.end |
    start.Hiding <= trial.end)

# now, I need to create an 'end cap' value to replace any 'stop' behaviors
# basically, I need to close the observation (like in Boris)

# this also means I'll need to change the 'duration' columns, which are automatically
# exported from Boris.

# replace stop.Open with trial cutoff if higher than cutoff
boris_data_cutoff <- boris_data_cutoff %>%
  group_by(fish.id, trial.id) %>%
  mutate(stop.Open = if_else(stop.Open > trial.end, trial.end, stop.Open))

# replace stop.Hiding with trial cutoff if higher than cutoff
boris_data_cutoff <- boris_data_cutoff %>%
  group_by(fish.id, trial.id) %>%
  mutate(stop.Hiding = if_else(stop.Hiding > trial.end, trial.end, stop.Hiding))

# now, recalculate duration based on new end times
boris_data_cutoff <- boris_data_cutoff %>%
  group_by(fish.id, trial.id) %>%
  mutate(duration.Open = stop.Open - start.Open)

boris_data_cutoff <- boris_data_cutoff %>%
  group_by(fish.id, trial.id) %>%
  mutate(duration.Hiding = stop.Hiding - start.Hiding)
```

I know that I won't be using the third trial since most fish didn't get there, so I'm removing that data now.

```r
# removing 3rd trial since most fish didn't get three
boris_data_cutoff <- boris_data_cutoff %>%
  filter(trial.id == "1" |
    trial.id == "2")
```

Creating some summary data

```r
boris_data_summary <- boris_data_cutoff %>%
  group_by(fish.id, trial.id, site.id, species) %>%
  summarise(
    total.time.hiding = sum(duration.Hiding, na.rm = TRUE),
    total.time.open = sum(duration.Open, na.rm = TRUE),
    total.behavior.switches = n()
  )
```

```
## `summarise()` has grouped output by 'fish.id', 'trial.id', 'site.id'. You can
## override using the `.groups` argument.
```

Merging with length data

```r
# there are two fish in the length data that need more specific names due to an error in how we initial

# first, I need to change fish.id to a character
length_data$fish.id <- as.character(length_data$fish.id)
length_data$fish.id[length_data$fish.id == "PF-08" & length_data$site.id == "Brownsville"] <- "PF-08BR"

# changing fish.id back to a factor
length_data$fish.id <- as.factor(length_data$fish.id)

# making a simpler version of the length data
length_data_simple <- length_data[, c("fish.id", "standard.length", "total.length")]
```

Merge all data into a single dataframe.

```r
all_data <- boris_data_summary %>%
  left_join(length_data_simple, by = "fish.id")
```

```
## Warning in left_join(., length_data_simple, by = "fish.id"): Detected an unexpected many-to-many rela
## i Row 37 of `x` matches multiple rows in `y`.
## i Row 23 of `y` matches multiple rows in `x`.
## i If a many-to-many relationship is expected, set `relationship =
##   "many-to-many"` to silence this warning.
```

```r
all_data <- all_data %>%
  left_join(parasite_data, by = "fish.id")
```

```
## Warning in left_join(., parasite_data, by = "fish.id"): Detected an unexpected many-to-many relations
## i Row 79 of `x` matches multiple rows in `y`.
## i Row 12 of `y` matches multiple rows in `x`.
## i If a many-to-many relationship is expected, set `relationship =
##   "many-to-many"` to silence this warning.
```

Tidy up column names.

```r
# remove duplicate columns
all_data <- all_data %>% dplyr::select(
  -site.id.y,
  -species.y
```

```
)

all_data <- all_data %>% dplyr::rename(
  site.id = site.id.x,
  species = species.x
)
```

Inspect data

First, some 'dummy checks' to make sure the data make sense

```
# fish.IDs
unique(id_data$fish.id) # 68 unique IDs
```

```
##  [1] "PF-08BR" "PL-25"   "PF-25B"  "PF-26"   "PF-27"   "PF-28"   "PF-30"
##  [8] "PF-31"   "PF-32"   "PF-33"   "PF-35"   "PF-36"   "PF-37"   "PF-38"
## [15] "PF-39"   "PF-40"   "PF-46"   "PF-60"   "PF-61"   "PF-62"   "PF-65"
## [22] "PL-02"   "PL-05"   "PL-07"   "PL-08"   "PL-09"   "PL-11"   "PL-12"
## [29] "PL-13"   "PL-15"   "PL-20"   "PL-21"   "PL-22"   "PL-26"   "PL-28"
## [36] "PL-30"   "PL-31"   "PL-32"   "PL-34"   "PL-35"   "PL-37"   "PL-38"
## [43] "PL-39"   "PL-41"   "PL-43"   "PL-44"   "PL-45"   "PL-46"   "PL-47"
## [50] "PL-48"   "PL-49"   "PL-50"   "PL-51"   "PL-52"   "PL-53"   "PL-54"
## [57] "PL-55"   "PL-56"   "PL-57"   "PL-58"   "PL-59"   "PL-60"   "PL-61"
## [64] "PL-62"   "PL-63"   "PL-64"   "PL-65"   "PL-66"
```

```
unique(boris_data_summary$fish.id) # 68 unique IDs
```

```
##  [1] "PF-08BR" "PF-25B"  "PF-26"   "PF-27"   "PF-28"   "PF-30"   "PF-31"
##  [8] "PF-32"   "PF-33"   "PF-35"   "PF-36"   "PF-37"   "PF-38"   "PF-39"
## [15] "PF-40"   "PF-46"   "PF-60"   "PF-61"   "PF-62"   "PF-65"   "PL-02"
## [22] "PL-05"   "PL-07"   "PL-08"   "PL-09"   "PL-11"   "PL-12"   "PL-13"
## [29] "PL-15"   "PL-20"   "PL-21"   "PL-22"   "PL-25"   "PL-26"   "PL-28"
## [36] "PL-30"   "PL-31"   "PL-32"   "PL-34"   "PL-35"   "PL-37"   "PL-38"
## [43] "PL-39"   "PL-41"   "PL-43"   "PL-44"   "PL-45"   "PL-46"   "PL-47"
## [50] "PL-48"   "PL-49"   "PL-50"   "PL-51"   "PL-52"   "PL-53"   "PL-54"
## [57] "PL-55"   "PL-56"   "PL-57"   "PL-58"   "PL-59"   "PL-60"   "PL-61"
## [64] "PL-62"   "PL-63"   "PL-64"   "PL-65"   "PL-66"
```

```
unique(parasite_data$fish.id)
```

```
##   [1] P.formosa   P.latipinna PF-01       PF-02       PF-04       PF-05
##   [7] PF-07       PF-08BR     PF-09       PF-10       PF-11       PF-12
##  [13] PF-13       PF-14       PF-15       PF-16       PF-17       PF-18
##  [19] PF-19       PF-20       PF-21       PF-22       PF-23       PF-24
##  [25] PF-25B      PF-26       PF-27       PF-28       PF-29       PF-3
##  [31] PF-30       PF-31       PF-32       PF-33       PF-34       PF-35
##  [37] PF-36       PF-37       PF-38       PF-39       PF-40       PF-41
##  [43] PF-42       PF-43       PF-44       PF-45       PF-46       PF-47
##  [49] PF-49       PF-50       PF-51       PF-52       PF-53       PF-54
##  [55] PF-55       PF-56       PF-57       PF-58       PF-59       PF-6
##  [61] PF-60       PF-61       PF-62       PF-63       PF-64       PF-65
##  [67] PL-01       PL-02       PL-04       PL-05       PL-07       PL-08
##  [73] PL-09       PL-10       PL-11       PL-12       PL-13       PL-14
##  [79] PL-15       PL-16       PL-17       PL-18       PL-19       PL-20
##  [85] PL-22       PL-25       PL-26       PL-27       PL-28       PL-29
##  [91] PL-30       PL-31       PL-32       PL-34       PL-35       PL-36
```

```
##  [97] PL-37       PL-38       PL-40       PL-41       PL-42       PL-43
## [103] PL-44       PL-46       PL-47       PL-48       PL-49       PL-50
## [109] PL-51       PL-52       PL-53       PL-54       PL-55       PL-56
## [115] PL-57       PL-58       PL-60       PL-62       PL-63       PL-64
## [121] PL-66
## 122 Levels:  P.formosa P.latipinna PF-01 PF-02 PF-04 PF-05 PF-07 ... PL-66
```

```
# 121, but 2 are just 'P.formosa' and 'P.latipina' because these are how fish were labeled if they didn
unique(length_data$fish.id) # 128, but this is probably ok. We photographed way more fish for length th
```

```
##   [1] PL-1    PL-02   PF-01   PL-03   PF-02   PL-04   PF-3    PF-04   PL-05
##  [10] PF-05   PF-06   PL-06   PF-07   PF-08   PL-07   PL-08   PL-09   PL-10
##  [19] PL-11   PL-12   PL-13   PL-14   PF-08BR PF-09   PF-10   PF-11   PF-12
##  [28] PF-13   PF-14   PF-15   PF-16   PF-17   PF-18   PF-19   PF-20   PF-21
##  [37] PL-15   PL-16   PF-22   PL-17   PF-23   PF-24   PL-18   PL-19   PF-25
##  [46] PF-25B  PF-26   PL-20   PF-27   PF-28   PF-29   PF-30   PF-31   PF-32
##  [55] PL-21   PF-33   PL-28   PL-29   PF-35   PL-27   PL-26   PL-30   PL-31
##  [64] PF-36   PF-37   PF-38   PL-32   PL-33   PF-39   PL-34   PL-35   PL-36
##  [73] PL-38   PF-40   PL-37   PF-34   PL-39   PL-40   PF-41   PL-41   PL-42
##  [82] PL-43   PL-44   PF-42   PF-43   PF-44   PF-45   PL-45   PF-46   PF-47
##  [91] PL-46   PL-47   PF-48   PL-48   PL-49   PF-49   PF-50   PF-51   PF-52
## [100] PF-53   PL-50   PF-54   PF-55   PF-56   PF-57   PF-58   PF-59   PL-51
## [109] PL-52   PL-53   PF-60   PL-54   PF-61   PL-55   PL-56   PF-62   PF-63
## [118] PL-57   PF-64   PL-58   PL-59   PL-60   PL-61   PL-62   PL-63   PL-64
## [127] PL-65   PL-66   PF-65   PL-22   PL-25
## 131 Levels: PF-01 PF-02 PF-04 PF-05 PF-06 PF-07 PF-08 PF-08BR PF-09 ... PL-66
```

```
unique(all_data$fish.id) # 68 unique IDs
```

```
##  [1] "PF-08BR" "PF-25B"  "PF-26"   "PF-27"   "PF-28"   "PF-30"   "PF-31"
##  [8] "PF-32"   "PF-33"   "PF-35"   "PF-36"   "PF-37"   "PF-38"   "PF-39"
## [15] "PF-40"   "PF-46"   "PF-60"   "PF-61"   "PF-62"   "PF-65"   "PL-02"
## [22] "PL-05"   "PL-07"   "PL-08"   "PL-09"   "PL-11"   "PL-12"   "PL-13"
## [29] "PL-15"   "PL-20"   "PL-21"   "PL-22"   "PL-25"   "PL-26"   "PL-28"
## [36] "PL-30"   "PL-31"   "PL-32"   "PL-34"   "PL-35"   "PL-37"   "PL-38"
## [43] "PL-39"   "PL-41"   "PL-43"   "PL-44"   "PL-45"   "PL-46"   "PL-47"
## [50] "PL-48"   "PL-49"   "PL-50"   "PL-51"   "PL-52"   "PL-53"   "PL-54"
## [57] "PL-55"   "PL-56"   "PL-57"   "PL-58"   "PL-59"   "PL-60"   "PL-61"
## [64] "PL-62"   "PL-63"   "PL-64"   "PL-65"   "PL-66"
```

This all makes sense. The number of unique fish IDs from my notebook match up with the boris data. There are more parasite fish IDs because we sent Jessica fish that didn't necessarily go through trials. Same for fish lengths, more were photographed than made it to trials.

```
# video.ID
unique(boris_data_cutoff$video.id) # 28
```

```
##  [1] "BR_trial1_01_20220808.mov"  "BR_trial1_02_20220808.mov"
##  [3] "BR_trial2_01_20220808.mov"  "BR_trial2_02_20220808.mov"
##  [5] "BR1_trial1_01_20220810.mov" "BR1_trial1_02_20220810.mov"
##  [7] "BR1_trial2_01_20220810.mov" "BR1_trial2_02_20220810.mov"
##  [9] "BR2_trial1_01_20220810.mov" "BR2_trial1_02_20220810.mov"
## [11] "BR2_trial2_01_20220810.mov" "BR2_trial2_02_20220810.mov"
## [13] "WES_trial1_01_20220808.mov" "Wes_trial1_01_20220812.mov"
## [15] "WES_trial1_02_20220808.mov" "WES_trial1_02_20220812.mov"
## [17] "WES_trial1_03_20220808.mov" "WES_trial1_03_20220812.mov"
## [19] "WES_trial1_04_20220812.mov" "WES_trial1_05_20220812.mov"
```

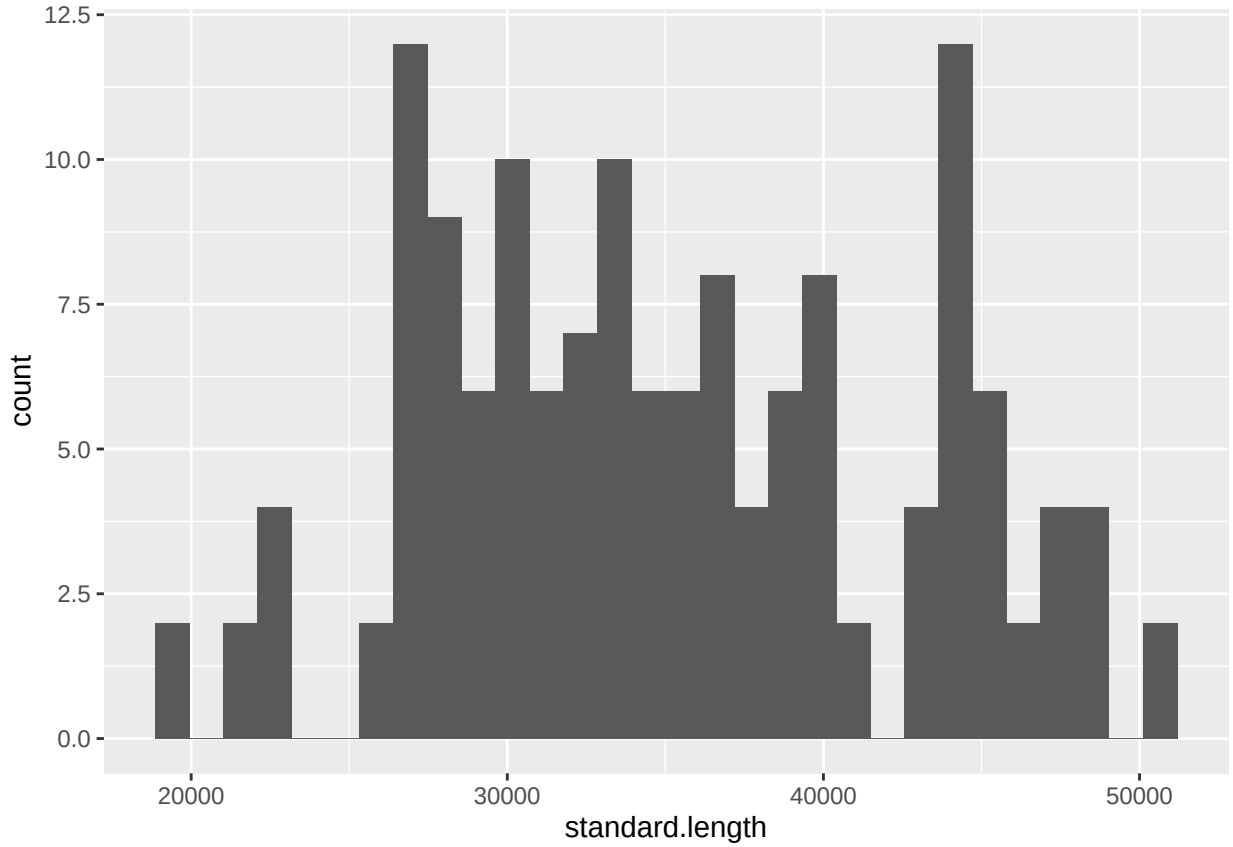
```
## [21] "WES_trial2_01_20220808.mov" "WES_trial2_01_20220812.mov"
## [23] "WES_trial2_02_20220808.mov" "WES_trial2_02_20220812.mov"
## [25] "WES_trial2_03_20220808.mov" "WES_trial2_03_20220812.mov"
## [27] "WES_trial2_04_20220812.mov" "WES_trial2_05_20220812.mov"
```

```
unique(id_data$video.id) # 36
```

```
##  [1] "BR_trial1_01_20220808.mov"  "BR_trial3_01_20220808.mov"
##  [3] "BR_trial2_01_20220808.mov"  "WES_trial1_01_20220808.mov"
##  [5] "WES_trial2_01_20220808.mov" "WES_trial3_02_20220809.mov"
##  [7] "WES_trial1_02_20220808.mov" "WES_trial2_02_20220808.mov"
##  [9] "WES_trial1_03_20220808.mov" "WES_trial2_03_20220808.mov"
## [11] "WES_trial3_01_20220809.mov" "BR1_trial1_01_20220810.mov"
## [13] "BR1_trial2_01_20220810.mov" "BR2_trial1_01_20220810.mov"
## [15] "BR2_trial2_01_20220810.mov" "BR2_trial1_02_20220810.mov"
## [17] "BR2_trial2_02_20220810.mov" "BR2_trial3_02_20220810.mov"
## [19] "BR2_trial3_01_20220810.mov" "BR1_trial3_01_20220810.mov"
## [21] "BR1_trial1_02_20220810.mov" "BR1_trial2_02_20220810.mov"
## [23] "WES_trial1_04_20220812.mov" "WES_trial2_04_20220812.mov"
## [25] "WES_trial1_05_20220812.mov" "WES_trial2_05_20220812.mov"
## [27] "BR_trial1_02_20220808.mov"  "BR_trial2_02_20220808.mov"
## [29] "BR_trial3_02_20220808.mov"  "BR1_trial3_02_20220810.mov"
## [31] "WES_trial1_02_20220812.mov" "WES_trial2_02_20220812.mov"
## [33] "WES_trial1_03_20220812.mov" "WES_trial2_03_20220812.mov"
## [35] "Wes_trial1_01_20220812.mov" "WES_trial2_01_20220812.mov"
```

This also makes sense. We have fewer video IDs in the boris data because we removed trial 3.

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



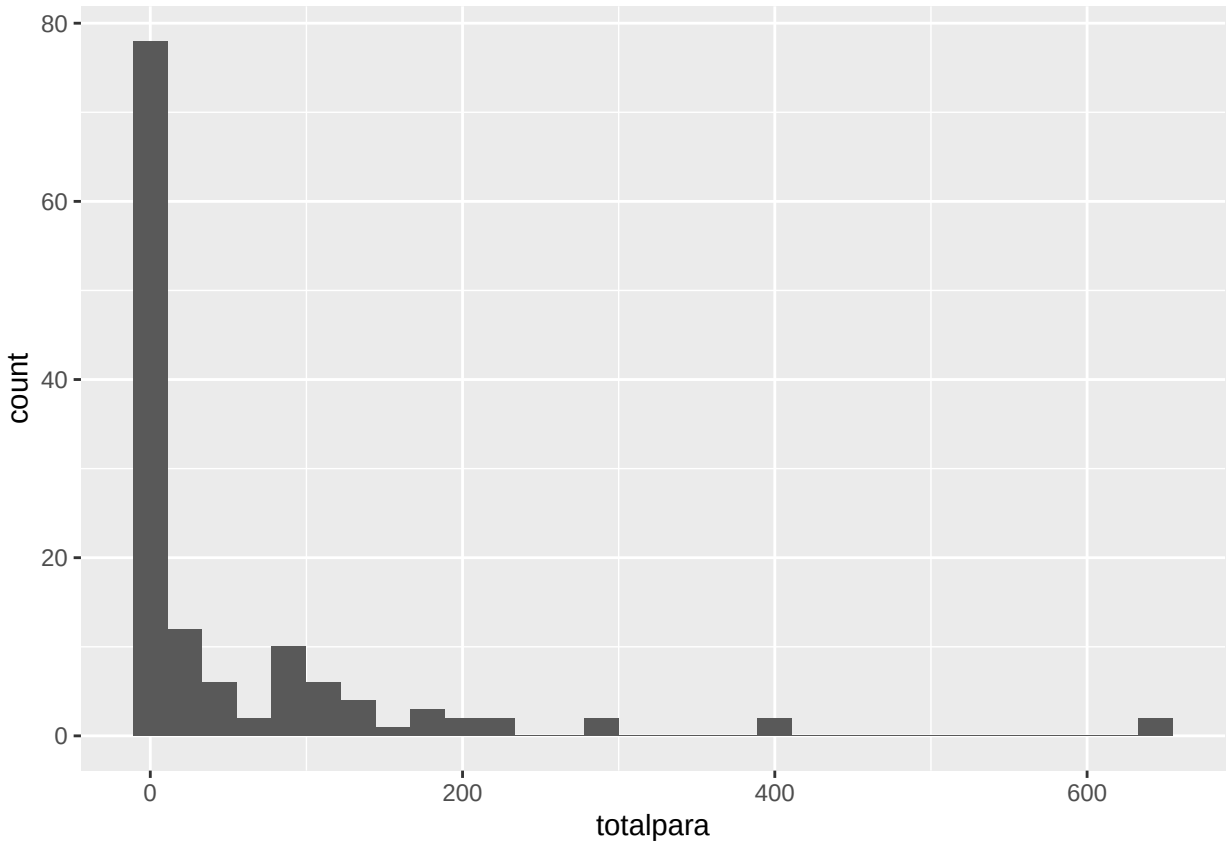
Seems like a fairly normal distribution for length! Some mongo fish were over 5cm!

Let's take a look at the parasite data.

```
# parasite data
parasite_hist <- all_data %>%
  ggplot(mapping = aes(totalpara)) +
  geom_histogram()
parasite_hist
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

```
## Warning: Removed 12 rows containing non-finite values (`stat_bin()`).
```

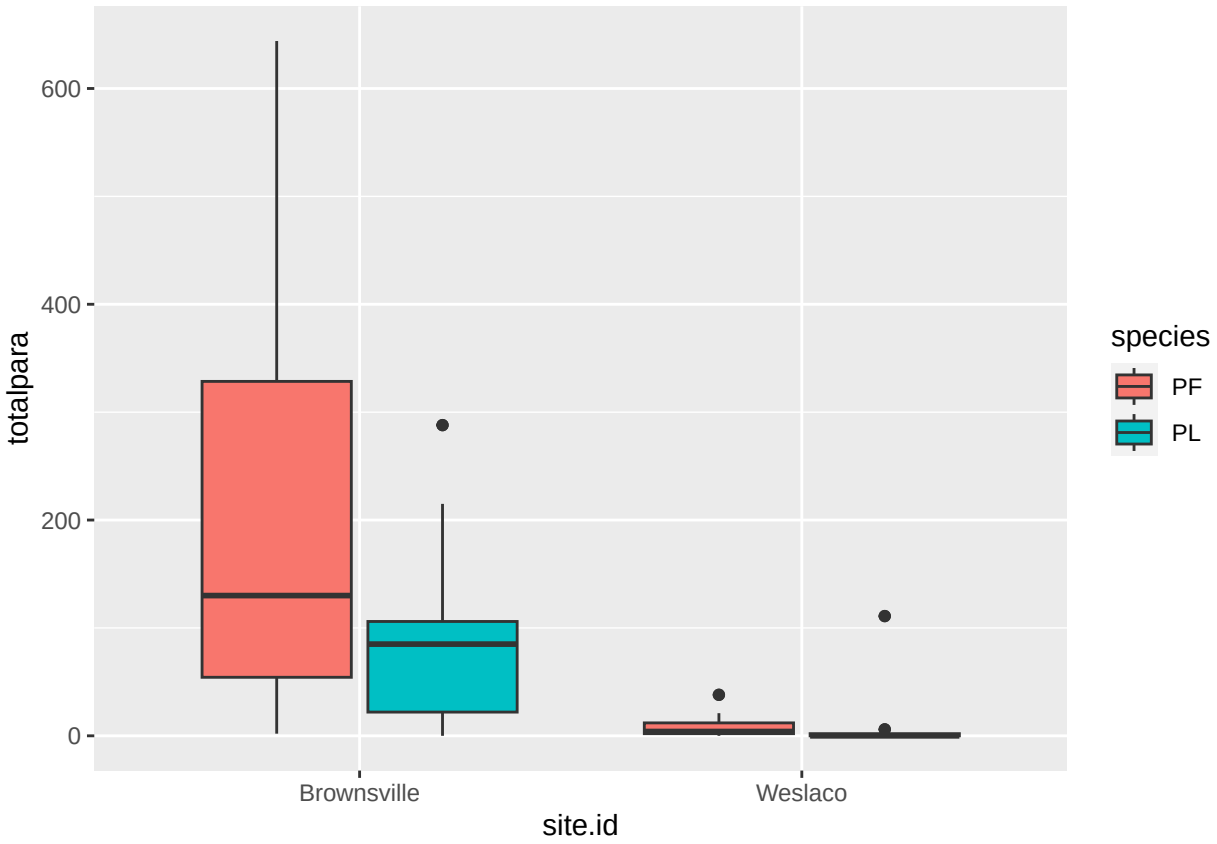


Not a normal distribution. No obvious pattern here, besides a lot of zeros. Should check to see how this matches up with notes in the parasite data about specimen quality).

Let's look at the parasites by species and site.

```
sp_parasite_box <- all_data %>%
  ggplot(mapping = aes(
    fill = species,
    x = site.id,
    y = totalpara
  )) +
  geom_boxplot()
sp_parasite_box
```

```
## Warning: Removed 12 rows containing non-finite values (`stat_boxplot()`).
```

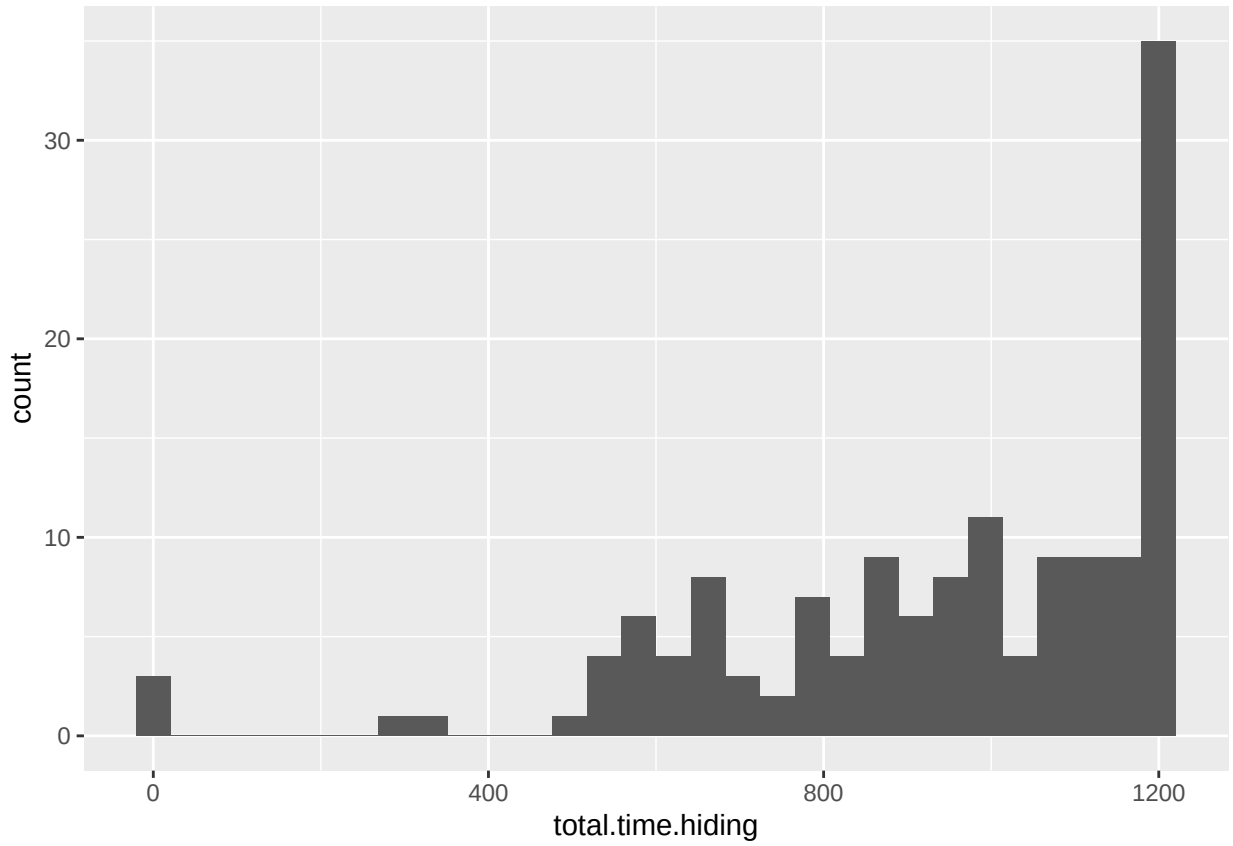


Ok, so there is a clear pattern of more parasites in Brownsville, generally. It also seems like there may be more parasites on Amazons in both sites, but we'll see what the stats say.

Now, let's take a look at the shape of the behavior data.

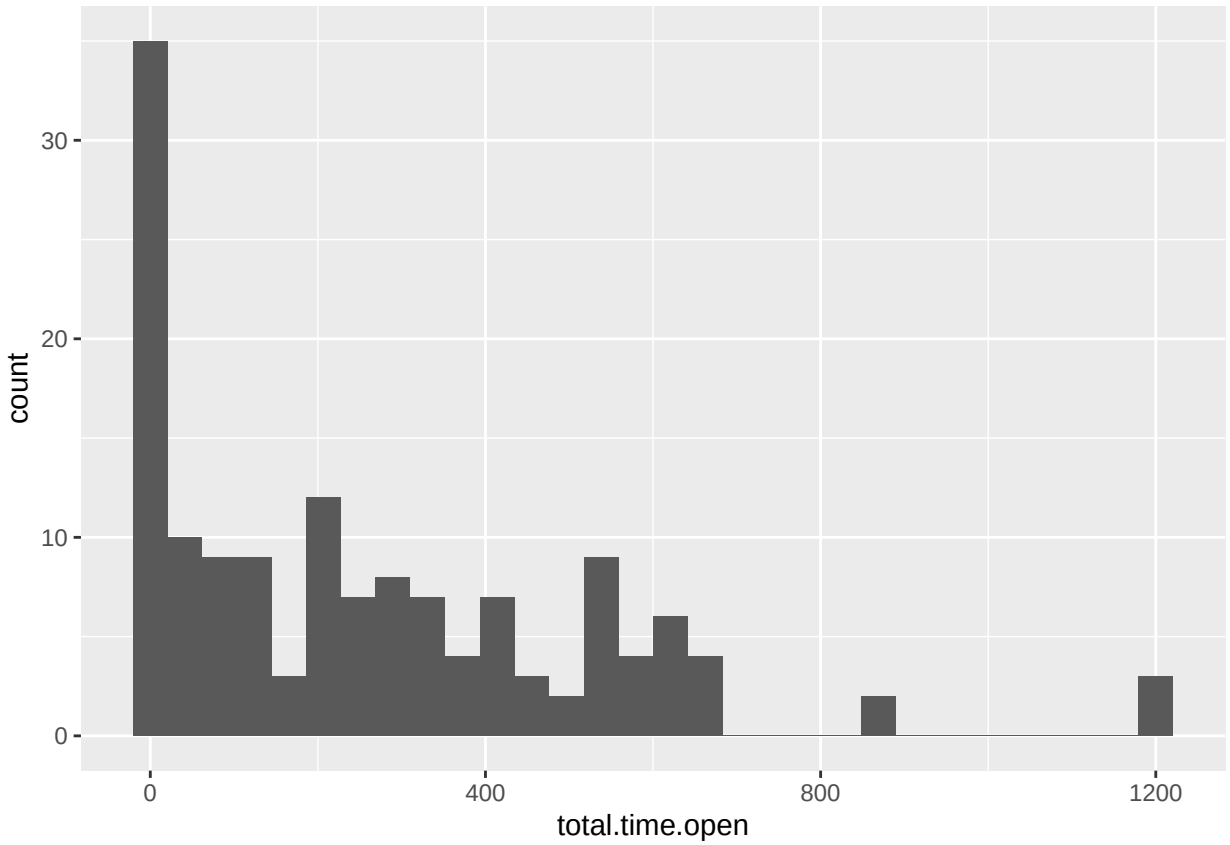
```
# boris_data, distributions
hiding_hist <- all_data %>%
  ggplot(mapping = aes(total.time.hiding)) +
  geom_histogram()
hiding_hist
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```




```
swimming_hist <- all_data %>%
  ggplot(mapping = aes(total.time.open)) +
  geom_histogram()
swimming_hist
```

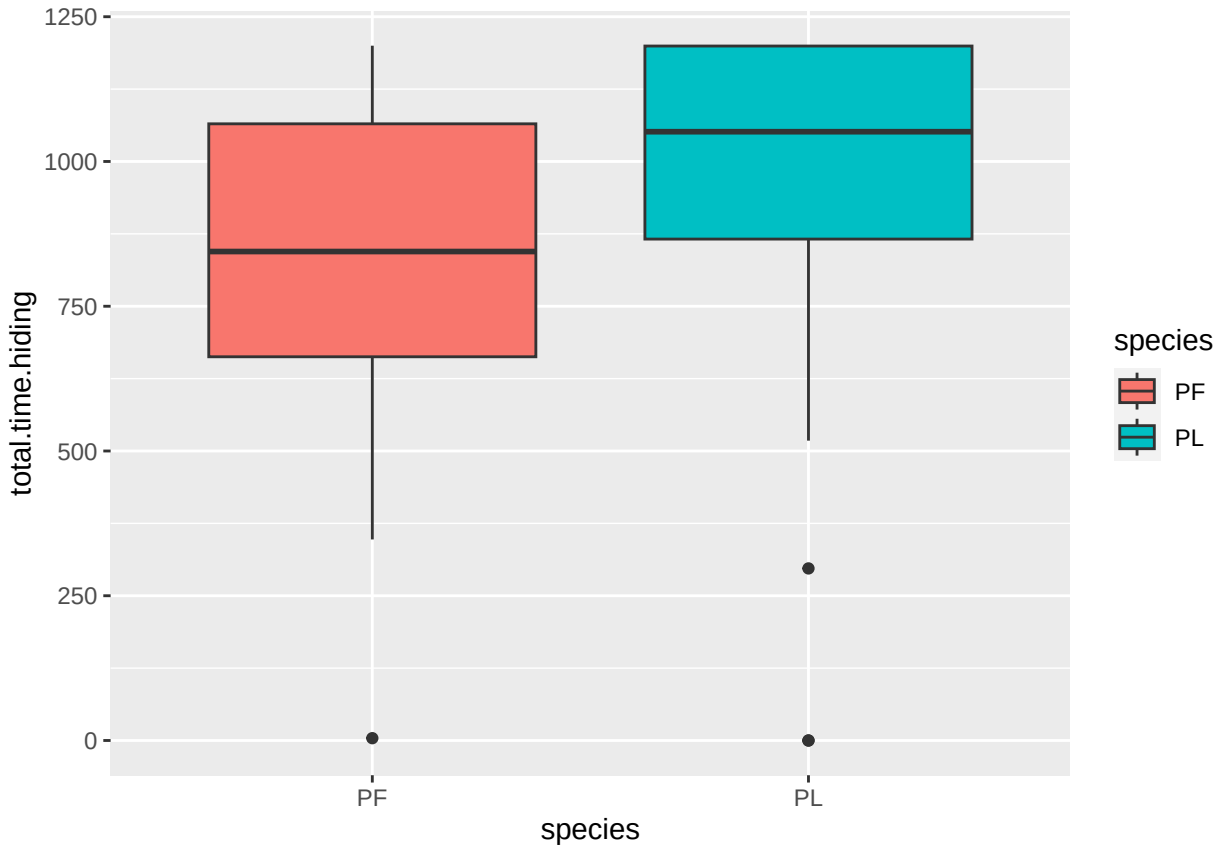
```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



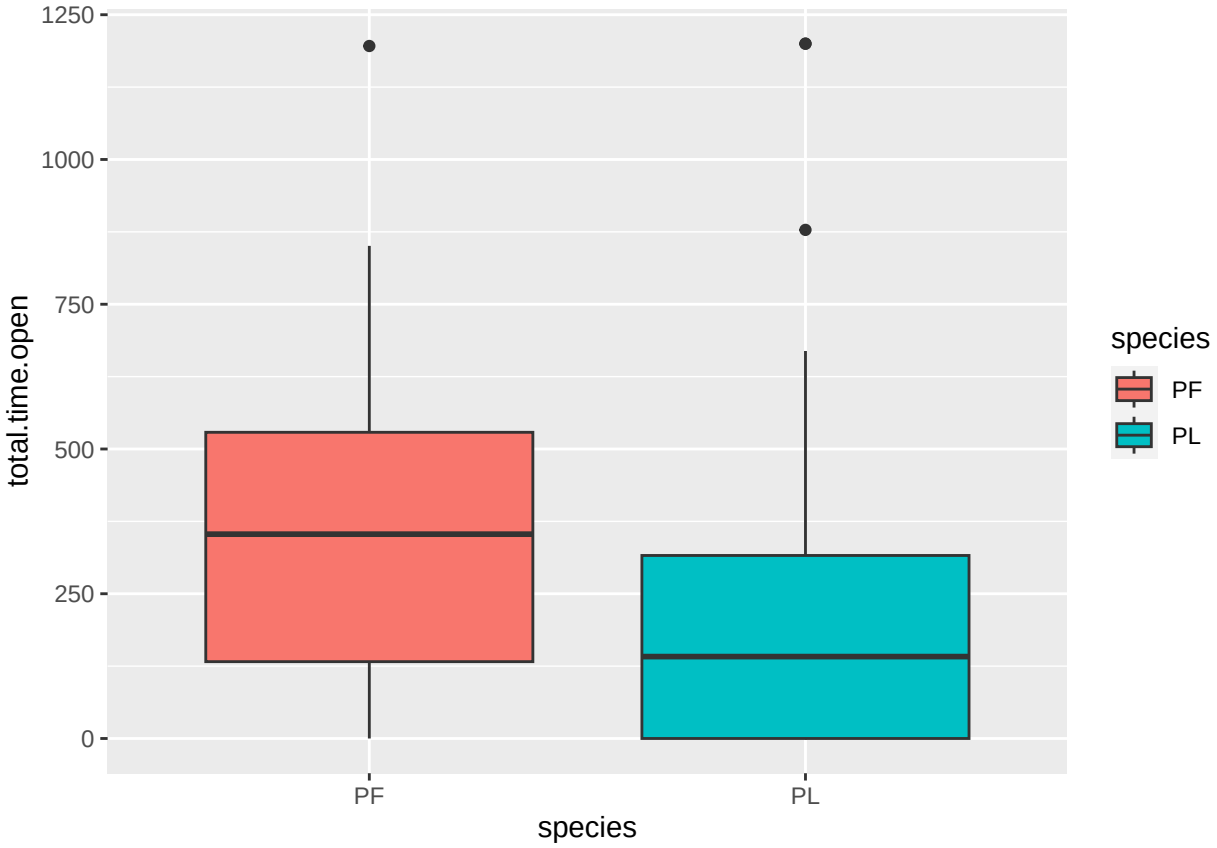
Ok, so it looks like we have a ceiling for total.time.hiding (i.e. lots of fish spent the whole trial hiding), with a right skew. For time in the open, we have a floor of zero, so a strong left skew (i.e. fish spent very little time out in the open). This indicates that many fish spent the entire, or most of the trial hiding.

Now, let's take a look at the open vs. hiding data by species.

```
# diff in hiding between species
species_hiding <- all_data %>%
  ggplot(mapping = aes(
    x = species,
    y = total.time.hiding,
    fill = species
  )) +
  geom_boxplot()
species_hiding
```




```
# diff in open between species
species_open <- all_data %>%
  ggplot(mapping = aes(
    x = species,
    y = total.time.open,
    fill = species
  )) +
  geom_boxplot()
species_open
```



Again, we'll see how the stats pan out, but it looks like Amazons might spend less time hiding and more time in the open than sailfins.

Models

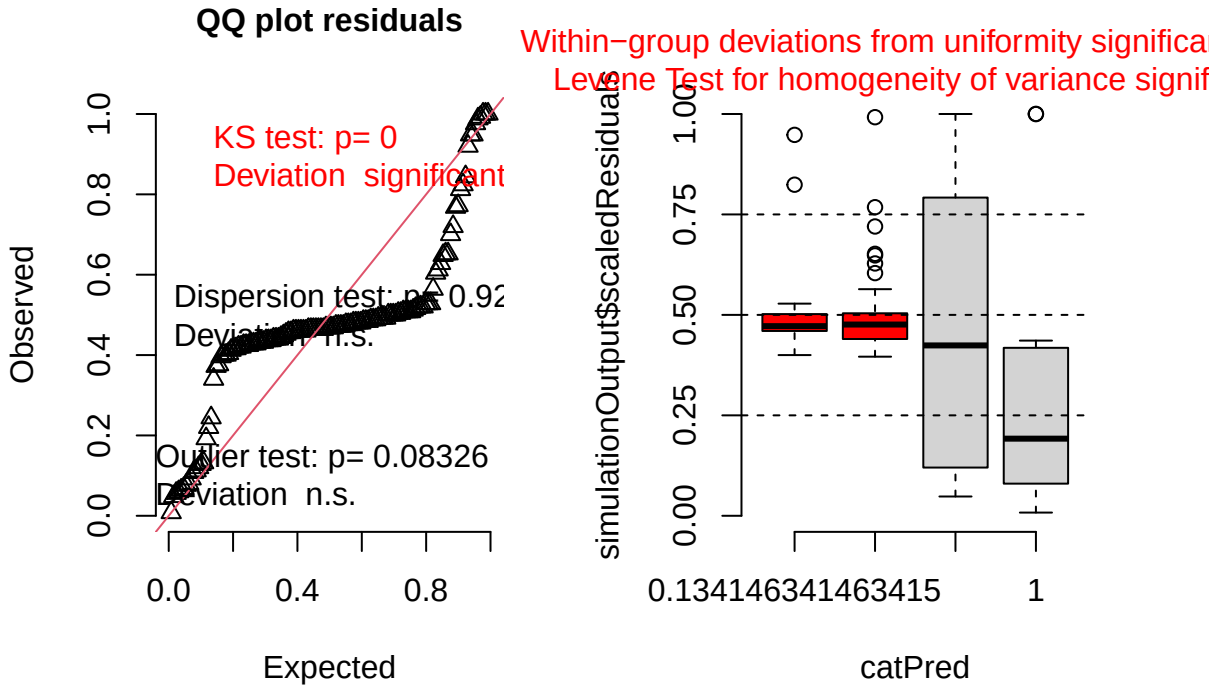
Parasites

First, I want to see if there is a difference in parasite load between Amazons and Sailfins. This will be using the full dataset from Jessica, which includes fish that did not go through behavioral trials. We'll start with both sites, but I may just end up looking at the Weslaco site since that is a more balanced dataset.

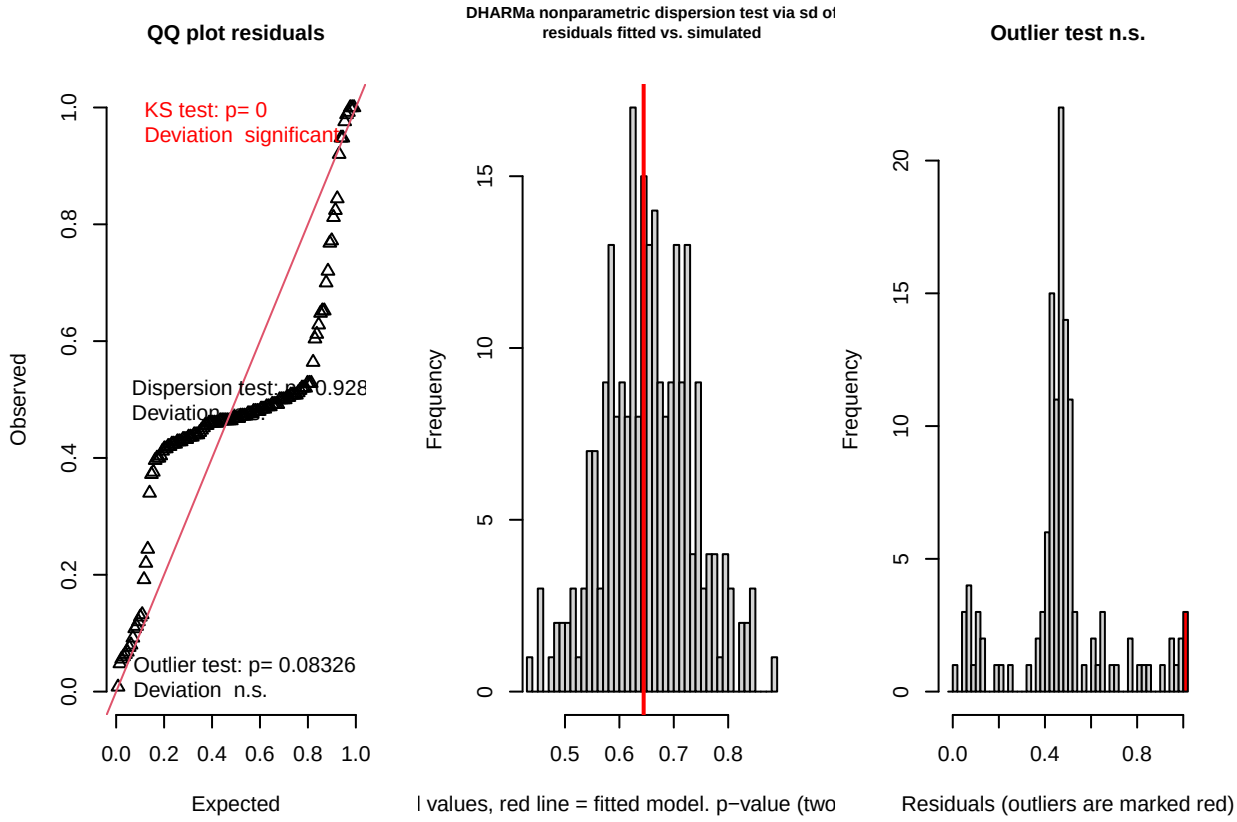
I kind of know already that these are going to be quite zero inflated, but let's start with a full linear model to confirm.

```
mod_full <- lm(totalpara ~ species * site.id,
  data = parasite_data
)

# to run all the tests in DHARMa, you first have to simulate your residuals
sim.mod_full <- DHARMa::simulateResiduals(mod_full)

# then you can plot them
plot(sim.mod_full) # op, both of these look pretty bad
```

DHARMa residual

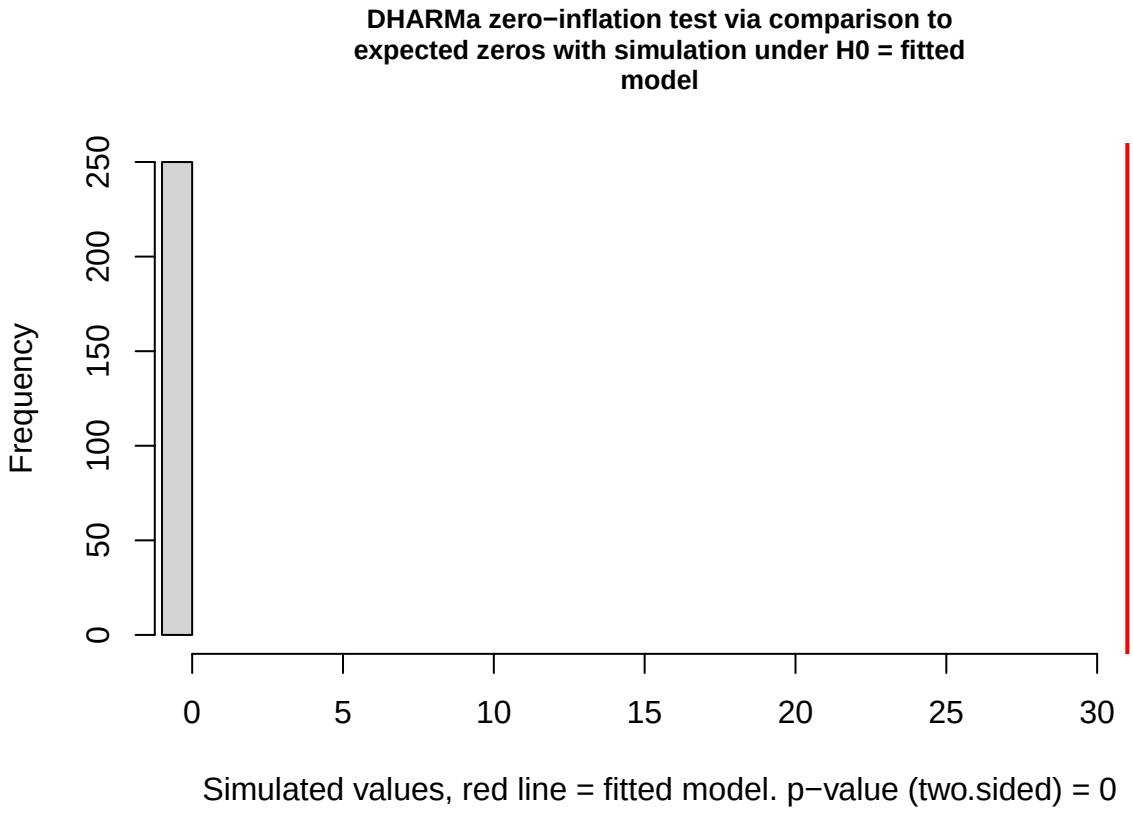



```
# gives you a bunch of tests of dispersion etc
DHARMa::testResiduals(sim.mod_full) # dispersion looks ok, but the QQ plot is bad
```



```
## $uniformity
##
##  Asymptotic one-sample Kolmogorov-Smirnov test
##
## data:  simulationOutput$scaledResiduals
## D = 0.29231, p-value = 6.326e-10
## alternative hypothesis: two-sided
##
##
## $dispersion
##
##  DHARMa nonparametric dispersion test via sd of residuals fitted vs.
##  simulated
##
## data:  simulationOutput
## dispersion = 0.98604, p-value = 0.928
## alternative hypothesis: two.sided
##
##
## $outliers
##
##  DHARMa outlier test based on exact binomial test with approximate
##  expectations
##
## data:  simulationOutput
## outliers at both margin(s) = 3, observations = 128, p-value = 0.08326
## alternative hypothesis: true probability of success is not equal to 0.007968127
## 95 percent confidence interval:
##  0.004859704 0.066966302
## sample estimates:
## frequency of outliers (expected: 0.00796812749003984 )
##                                             0.0234375
```

```
# yes we can see that the data is super zero inflated
DHARMa::testZeroInflation(sim.mod_full) # yep, super zero inflated
```




```
##
##  DHARMa zero-inflation test via comparison to expected zeros with
##  simulation under H0 = fitted model
##
## data:  simulationOutput
## ratioObsSim = Inf, p-value < 2.2e-16
## alternative hypothesis: two.sided
```

Ok, so the data is zero inflated. So we need to fit a different kind of model (poisson or possibly binomial).

```
mod_para_full <- zeroinfl(totalpara ~ species * site.id,
  dist = "negbin",
  lin = "logit",
  data = parasite_data
)

# sim.ouput <- DHARMa::simulateResiduals(mod_para_full)

# DHARMa doesn't like this model type. But it does play well with glmmTMB... I'm going to move forward
```

Let's try that again with glmmTMB

```
mod_full_poisson <- glmmTMB(totalpara ~ species * site.id,
  family = "poisson",
  ziformula = ~1,
  data = parasite_data
)

mod_full_nbin <- glmmTMB(totalpara ~ species * site.id,
  family = "nbinom1",
```

```
  ziformula = ~1,
  data = parasite_data
)
check_overdispersion(mod_full_poisson) # data is over dispersed
```

```
## # Overdispersion test
##
##  dispersion ratio =   5.539
##            p-value = < 0.001
```

```
## Overdispersion detected.
```

```
check_overdispersion(mod_full_nbin) # data is not over dispersed
```

```
## # Overdispersion test
##
##  dispersion ratio = 1.517
##            p-value = 0.136
```

```
## No overdispersion detected.
```

```
lrtest(mod_full_poisson, mod_full_nbin)
```

```
## Likelihood ratio test
##
## Model 1: totalpara ~ species * site.id
## Model 2: totalpara ~ species * site.id
##   #Df   LogLik Df  Chisq Pr(>Chisq)
## 1   5 -3101.13
## 2   6  -477.79  1 5246.7  < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

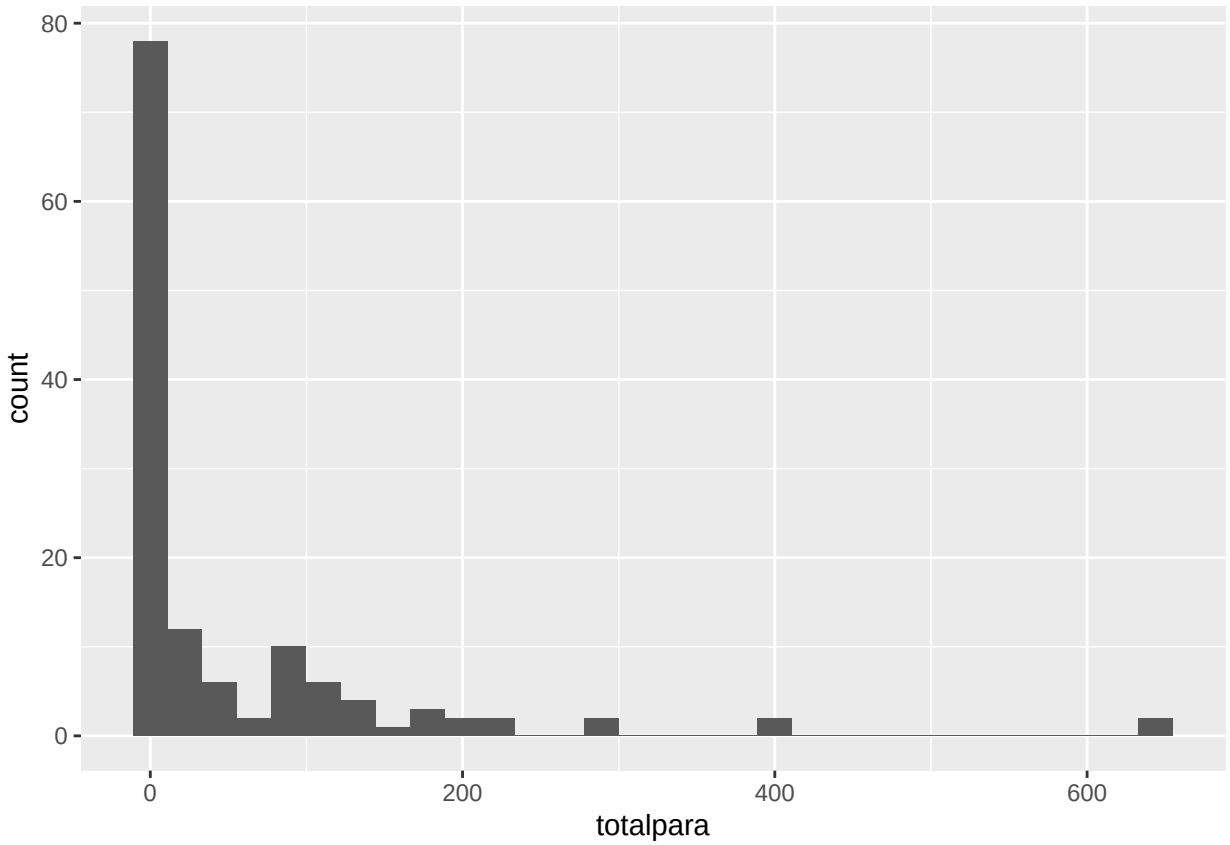
Ok so the negative binomial seems better than the poisson, because the poisson is over dispersed. So what are we to do? I believe that we have lots of true zeros in total.parasites (the folks doing the dissections knew what they were doing and it sounds like it was hard to miss the gill parasites). But maybe we have lots of zeros for non-true reasons (e.g. internal parasites were common, but tough to detect).

We might have overdispersion due to higher than expected spread in the positive values. See the following figure:

```
parasite_hist
```

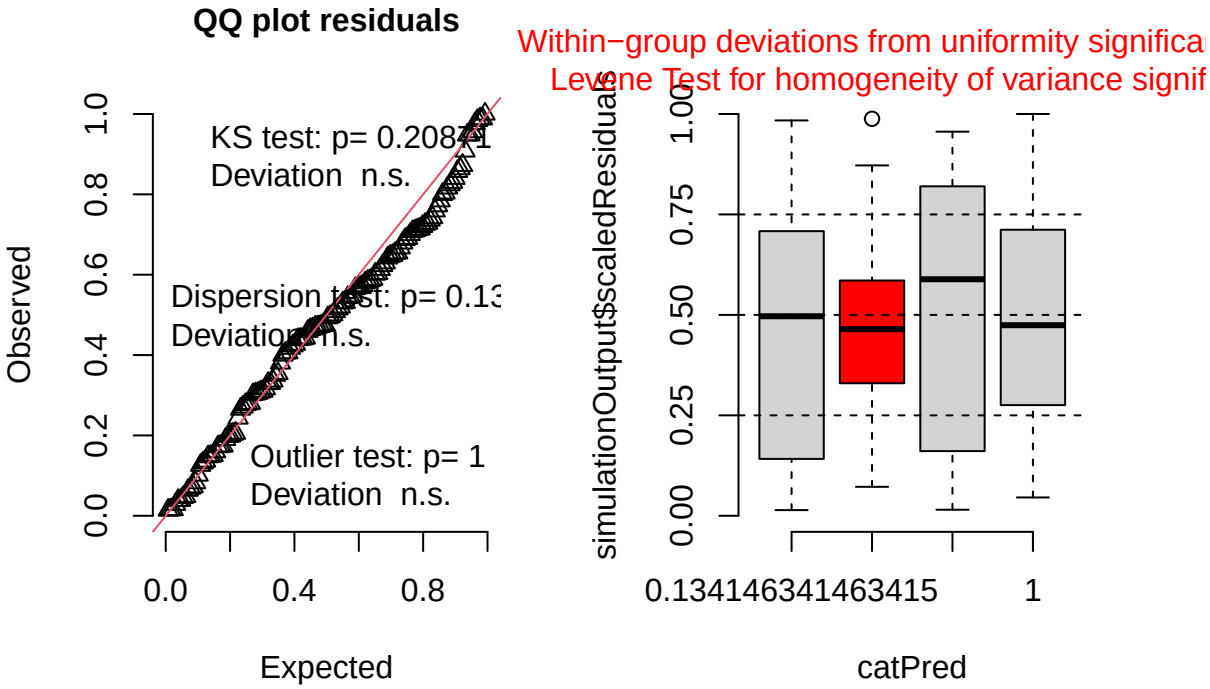
```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

```
## Warning: Removed 12 rows containing non-finite values (`stat_bin()`).
```

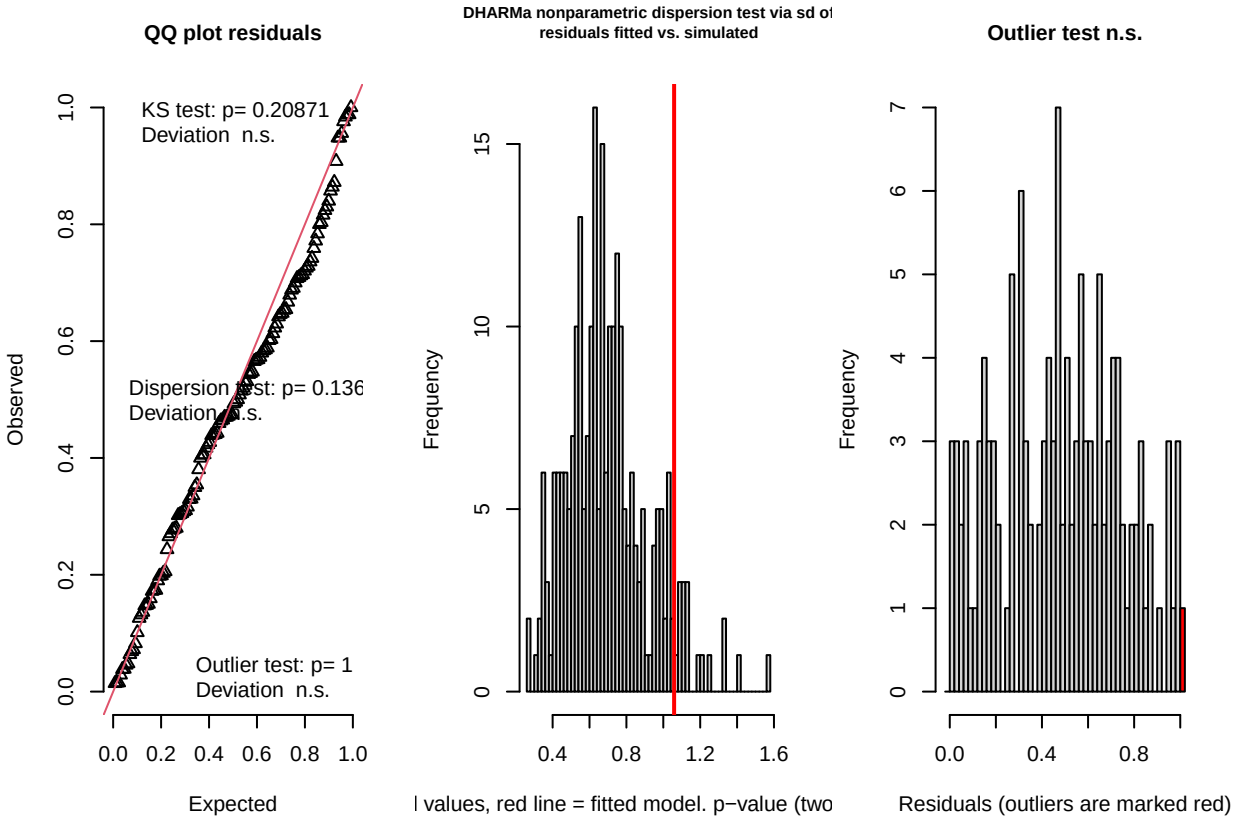


If Kate agrees with that interpretation, then we should probably go with a negative binomial distribution.

```
# to run all the tests in DHARMa, you first have to simulate your residuals
sim.output <- DHARMa::simulateResiduals(mod_full_nbin)

# then you can plot them
plot(sim.output) # these look much better! But the Levene Test for homogeneity of variance was signific
```




```
# gives you a bunch of tests of dispersion etc
DHARMa::testResiduals(sim.output) # QQ looks ok, and outlier test not significant. Dispersion not great
```



```
## $uniformity
##
##  Asymptotic one-sample Kolmogorov-Smirnov test
##
## data:  simulationOutput$scaledResiduals
## D = 0.093933, p-value = 0.2087
## alternative hypothesis: two-sided
##
##
## $dispersion
##
##  DHARMa nonparametric dispersion test via sd of residuals fitted vs.
##  simulated
##
## data:  simulationOutput
## dispersion = 1.5168, p-value = 0.136
## alternative hypothesis: two.sided
##
##
## $outliers
##
##  DHARMa bootstrapped outlier test
##
## data:  simulationOutput
## outliers at both margin(s) = 1, observations = 128, p-value = 1
## alternative hypothesis: two.sided
##  percent confidence interval:
##  0.0000000 0.0234375
## sample estimates:
## outlier frequency (expected: 0.005859375 )
##                                 0.0078125
```

```
summary(mod_full_nbin)
```

```
##  Family: nbinom1  ( log )
## Formula:          totalpara ~ species * site.id
## Zero inflation:             ~1
## Data: parasite_data
##
##      AIC      BIC   logLik deviance df.resid
##    967.6    984.7   -477.8    955.6      122
##
##
## Dispersion parameter for nbinom1 family (): 81.8
##
## Conditional model:
##                                 Estimate Std. Error z value Pr(>|z|)
## (Intercept)                       4.7379     0.2245  21.104  < 2e-16 ***
## specieslatipinna                 -0.0898     0.2703  -0.332    0.740
## site.idWeslaco                   -1.6583     0.2738  -6.056 1.39e-09 ***
## specieslatipinna:site.idWeslaco  -0.6028     0.4014  -1.502    0.133
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Zero-inflation model:
```

```
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -3.5111     0.8071   -4.35 1.36e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Ok so this tells us that there is a difference based on site (which we already kind of knew), but not species (which is interesting! But checks out).

Now, let's do some backwards model selection.

```
# the parasite count data is zero inflated and overdispersed, so I'm going to use a zero-inlfated negat

# I'm going to use backwards elimination

# interaction model
mod_full_nbin <- glmmTMB(totalpara ~ species * site.id,
  family = "nbinom1",
  ziformula = ~1,
  data = parasite_data
)

# combined model
mod_combined_nbin <- glmmTMB(totalpara ~ species + site.id,
  family = "nbinom1",
  ziformula = ~1,
  data = parasite_data
)

# test 2-way with log likelihood ratio test
lrtest(mod_full_nbin, mod_combined_nbin) # no difference, so let's stick with that for now
```

```
## Likelihood ratio test
##
## Model 1: totalpara ~ species * site.id
## Model 2: totalpara ~ species + site.id
##   #Df  LogLik Df  Chisq Pr(>Chisq)
## 1   6 -477.79
## 2   5 -478.96 -1 2.3313     0.1268
```

```
# site model
mod_site_nbin <- glmmTMB(totalpara ~ site.id,
  family = "nbinom1",
  ziformula = ~1,
  data = parasite_data
)

# test species effect
lrtest(mod_site_nbin, mod_combined_nbin) # juuuust over .05, so the site only is a better fit. This mak
```

```
## Likelihood ratio test
##
## Model 1: totalpara ~ site.id
## Model 2: totalpara ~ species + site.id
##   #Df  LogLik Df  Chisq Pr(>Chisq)
## 1   4 -480.85
## 2   5 -478.96  1 3.7882    0.05161 .
```

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
# species model
mod_species_nbin <- glmmTMB(totalpara ~ species,
  family = "nbinom1",
  ziformula = ~1,
  data = parasite_data
)

# test site effect
lrtest(mod_species_nbin, mod_combined_nbin) # the combined model fits the data better, indicating that
```

```
## Likelihood ratio test
##
## Model 1: totalpara ~ species
## Model 2: totalpara ~ species + site.id
##   #Df  LogLik Df  Chisq Pr(>Chisq)
## 1   4 -508.96
## 2   5 -478.96  1 60.005  9.461e-15 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
summary(mod_combined_nbin)
```

```
##  Family: nbinom1  ( log )
## Formula:          totalpara ~ species + site.id
## Zero inflation:             ~1
## Data: parasite_data
##
##      AIC      BIC   logLik deviance df.resid
##    967.9    982.2   -479.0    957.9      123
##
##
## Dispersion parameter for nbinom1 family (): 84.3
##
## Conditional model:
##                  Estimate Std. Error z value Pr(>|z|)
## (Intercept)        4.9106     0.1760  27.898   <2e-16 ***
## specieslatipinna  -0.3733     0.1955  -1.909   0.0562 .
## site.idWeslaco    -1.8986     0.2087  -9.095   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Zero-inflation model:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -3.6138     0.9527  -3.793 0.000149 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Weslaco Now, let's essentially repeat that analysis with just the Weslaco site.

```
## Now, just with the WESLACO site ##

# filter data to just Weslaco
parasite_data_wes <- parasite_data %>%
```

```
  filter(site.id == "Weslaco")

# species model
mod_para_species_wes <- glmmTMB(totalpara ~ species,
  family = "nbinom1",
  ziformula = ~1,
  data = parasite_data
)

summary(mod_para_species_wes) # yep, no significant species differences
```

```
##  Family: nbinom1  ( log )
## Formula:          totalpara ~ species
## Zero inflation:             ~1
## Data: parasite_data
##
##      AIC      BIC   logLik deviance df.resid
##   1025.9   1037.3   -509.0   1017.9      124
##
##
## Dispersion parameter for nbinom1 family ():  176
##
## Conditional model:
##                  Estimate Std. Error z value Pr(>|z|)
## (Intercept)       3.78216    0.20096  18.820   <2e-16 ***
## specieslatipinna -0.06194    0.20102  -0.308    0.758
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Zero-inflation model:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)   -19.48    3777.93  -0.005    0.996
```

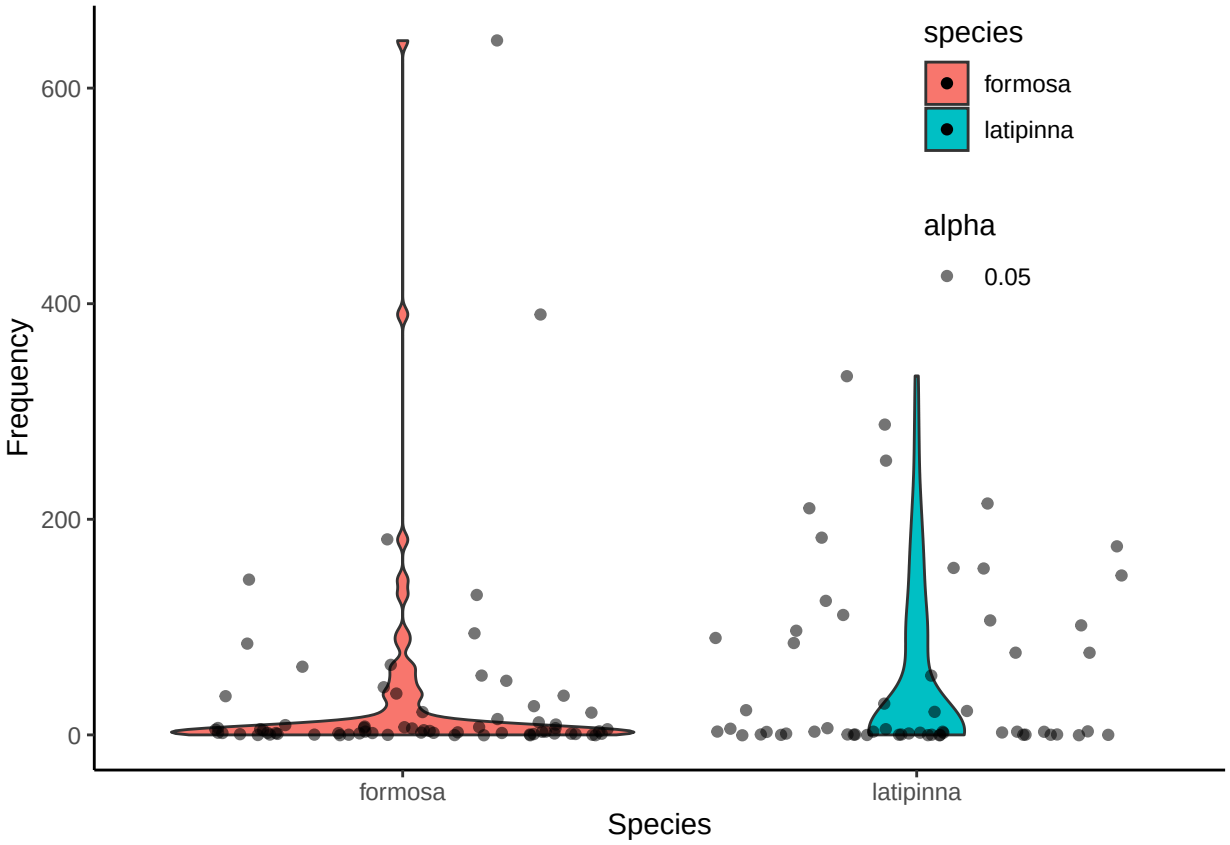
```
# just to be extra sure, I also ran an lrt on the weslaco species model versus a null model set to the
mod_para_null_wes <- glmmTMB(totalpara ~ 1,
  family = "nbinom1",
  ziformula = ~1,
  data = parasite_data
)

lrtest(mod_para_species_wes, mod_para_null_wes) # no significance, suggesting that the species effect i
```

```
## Likelihood ratio test
##
## Model 1: totalpara ~ species
## Model 2: totalpara ~ 1
##   #Df  LogLik Df  Chisq Pr(>Chisq)
## 1   4 -508.96
## 2   3 -509.01 -1 0.0953     0.7575
```

Figures Lets plot our combined parasite model.

```
# diff in total parasites by species
parasites_spp_boxplot <- parasite_data %>%
  ggplot(mapping = aes(
```

```
    x = species,
    y = totalpara,
    fill = species
  )) +
  geom_violin() +
  geom_jitter(aes(
    alpha = 0.05,
    fill = species
  )) +
  xlab("Species") +
  ylab("Frequency") +
  theme_classic() +
  theme(legend.position = c(0.8, 0.8))
parasites_spp_boxplot
```



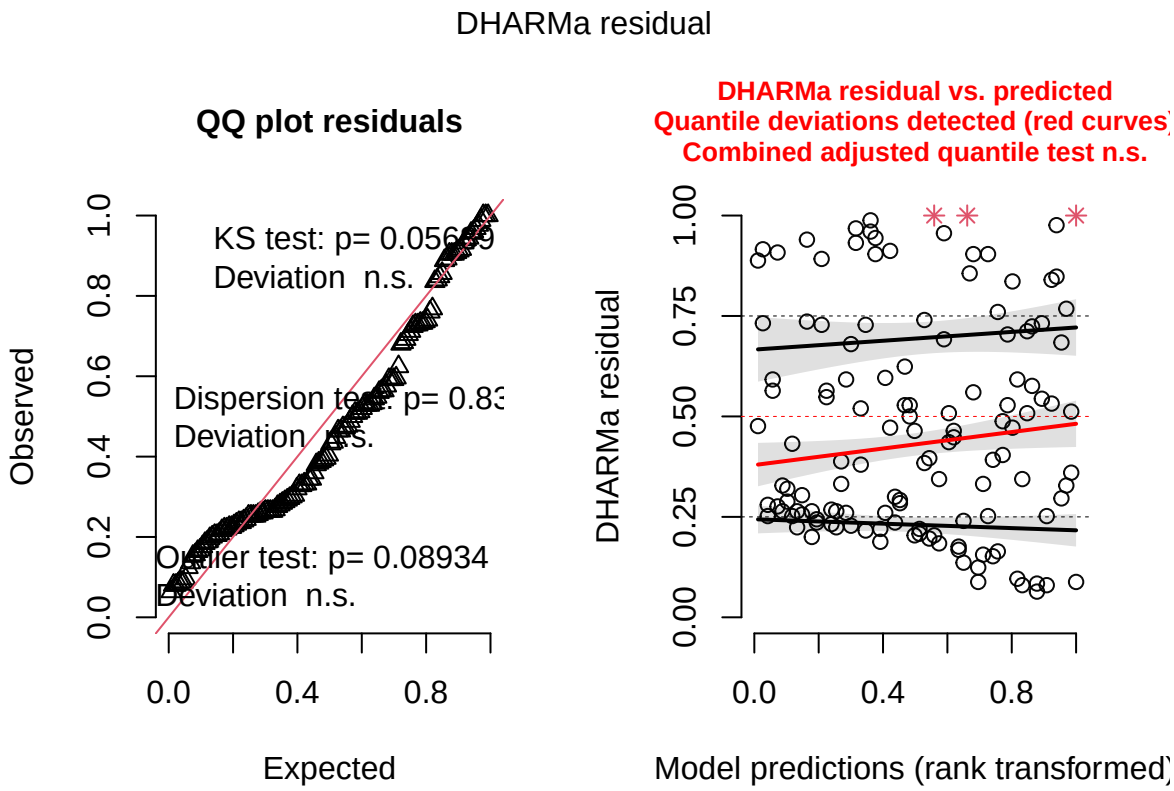
Behavior

Ok, now, we want to determine whether parasite load affects behavior, and whether that relationship differs between species. I'm going to start with all of the data (ignoring site), then repeat with just Weslaco.

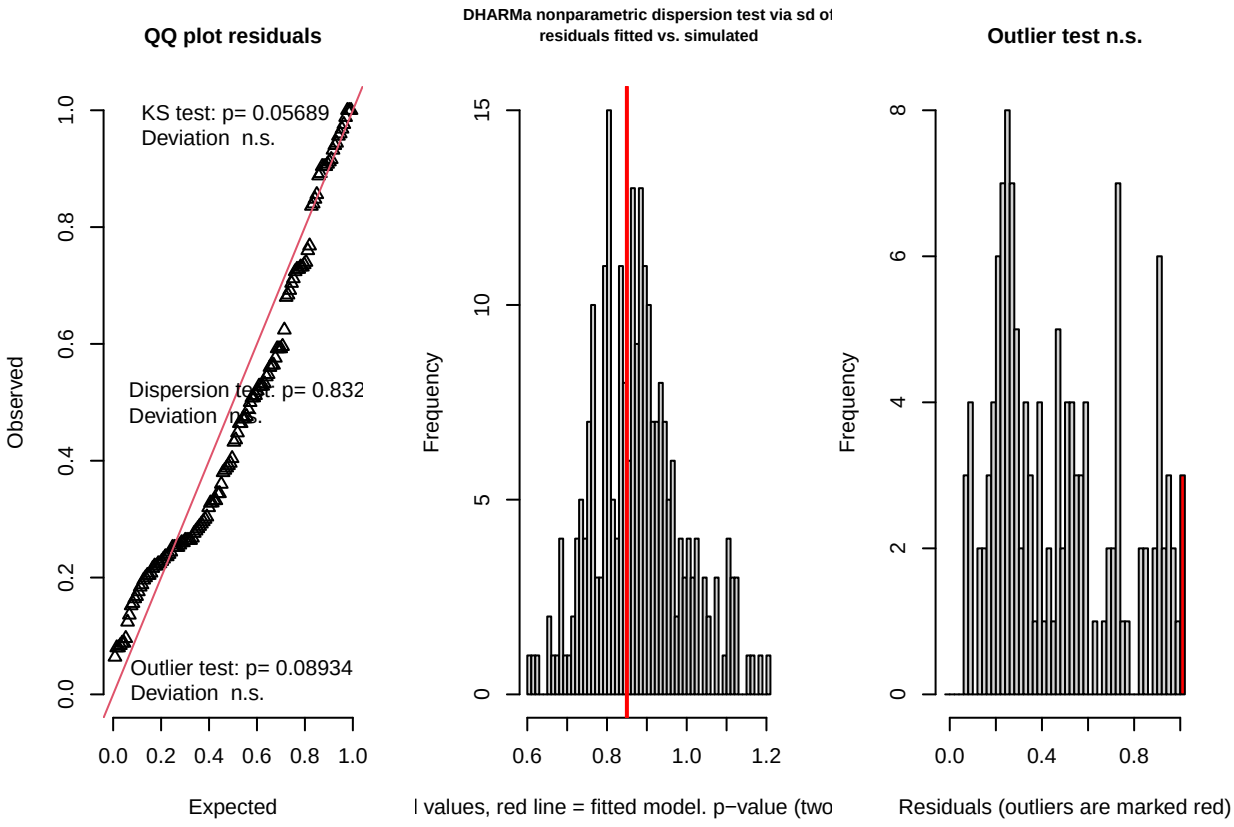
If we've got something interesting going on, it might be worth diving into smaller details, like how many times they switch between open and hiding or how behaviors change after the startle stimulus. But for now, let's just look at the big picture.

All data Does parasite load predict the amount of time spent in the open?

```r
mod_beh_full <- lm(total.time.open ~ totalpara * species + standard.length,
  data = all_data
)

# to run all the tests in DHARMa, you first have to simulate your residuals
sim_mod_beh_full <- DHARMa::simulateResiduals(mod_beh_full)

# then you can plot them
plot(sim_mod_beh_full) # QQ looks fine, but resid vs. predicted not so great? But also not too too bad,
```



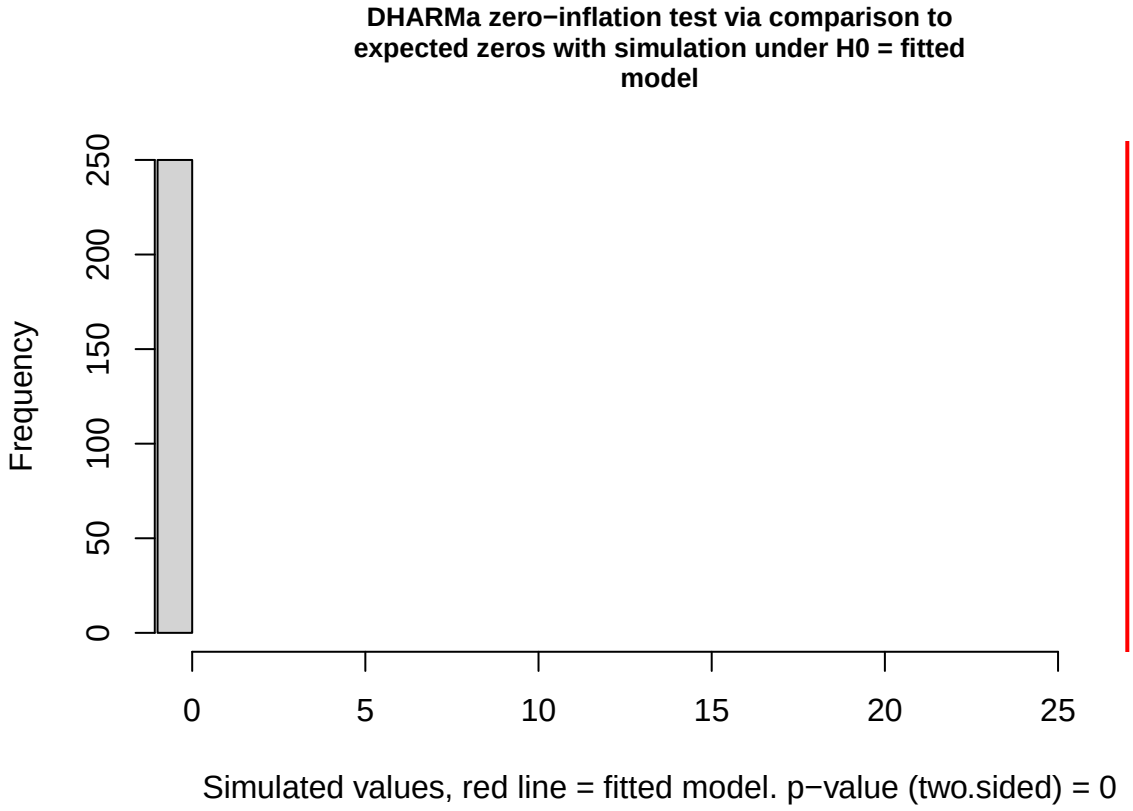

```r
# gives you a bunch of tests of dispersion etc
DHARMa::testResiduals(sim_mod_beh_full) # these all look fine
```




```
## $uniformity
##
##  Asymptotic one-sample Kolmogorov-Smirnov test
##
## data:  simulationOutput$scaledResiduals
## D = 0.11612, p-value = 0.05689
## alternative hypothesis: two-sided
##
##
## $dispersion
##
##  DHARMa nonparametric dispersion test via sd of residuals fitted vs.
##  simulated
##
## data:  simulationOutput
## dispersion = 0.97009, p-value = 0.832
## alternative hypothesis: two.sided
##
##
## $outliers
##
##  DHARMa outlier test based on exact binomial test with approximate
##  expectations
##
## data:  simulationOutput
## outliers at both margin(s) = 3, observations = 132, p-value = 0.08934
## alternative hypothesis: true probability of success is not equal to 0.007968127
## 95 percent confidence interval:
```

```
##  0.004711659 0.064981495
## sample estimates:
## frequency of outliers (expected: 0.00796812749003984 )
##                                              0.02272727
```

```
# just from watching some of the videos and inspecting the data, fish often spend a lot of time hiding
DHARMa::testZeroInflation(sim_mod_beh_full) # dang, zero inflated.
```




```
##
##  DHARMa zero-inflation test via comparison to expected zeros with
##  simulation under H0 = fitted model
##
## data:  simulationOutput
## ratioObsSim = Inf, p-value < 2.2e-16
## alternative hypothesis: two.sided
```

Ok so this data is also zero inflated. In 27/144 trials, fish spend the entire trial hiding. Let's move on with a zero inflated model. I'm going to check a poisson vs. negative binomial the same way I did before, but my gut is telling me this is wrong... These aren't count data, so we probably shouldn't use a negative binomial, but rather a poisson? I could convert the hiding/open times into ratios, which would almost definitely be right for a poisson?

```
# checking poisson vs. negbin
mod_beh_full_nbin <- glmmTMB(total.time.open ~ totalpara * species + standard.length,
  family = "nbinom1",
  ziformula = ~1,
  data = all_data
)
```

```
## Warning in glmmTMB(total.time.open ~ totalpara * species + standard.length, :
```

```
## non-integer counts in a nbinom1 model
```

```
## Warning in (function (start, objective, gradient = NULL, hessian = NULL, :
## NA/NaN function evaluation

## Warning in (function (start, objective, gradient = NULL, hessian = NULL, :
## NA/NaN function evaluation
```

```
# I get warnings here about "non-integer counts".
diagnose(mod_beh_full_nbin)
```

```
##
## predictors with unusually large or small standard deviations (|log10(sd)|>6956.52):
##
## standard.length
##        6956.519
## Predictor variables with very narrow or wide ranges generally give rise
## to parameters with very large or small magnitudes, which can sometimes
## exacerbate numerical instability, and may also be appear (incorrectly)
## to be indicating a poorly defined optimum (i.e., a non-positive
## definite Hessian
##
##
## Unusually large Z-statistics (|x|>5):
##
##    (Intercept) zi~(Intercept)  d~(Intercept)
##      16.233708      -6.277987      34.688211
##
## Large Z-statistics (estimate/std err) suggest a *possible* failure of
## the Wald approximation - often also associated with parameters that are
## at or near the edge of their range (e.g. random-effects standard
## deviations approaching 0).  (Alternately, they may simply represent
## very well-estimated parameters; intercepts of non-centered models may
## fall in this category.) While the Wald p-values and standard errors
## listed in summary() may be unreliable, profile confidence intervals
## (see ?confint.glmmTMB) and likelihood ratio test p-values derived by
## comparing models (e.g. ?drop1) are probably still OK.  (Note that the
## LRT is conservative when the null value is on the boundary, e.g. a
## variance or zero-inflation value of 0 (Self and Liang 1987; Stram and
## Lee 1994; Goldman and Whelan 2000); in simple cases these p-values are
## approximately twice as large as they should be.)
```

```
# As far as I can tell, I should be able to compare with poisson using lrt, but should be cautious abou

mod_beh_full_poisson <- glmmTMB(total.time.open ~ totalpara * species + standard.length,
  family = "poisson",
  ziformula = ~1,
  data = all_data
)
```

```
## Warning in glmmTMB(total.time.open ~ totalpara * species + standard.length, :
## non-integer counts in a poisson model
```

```
## Warning in (function (start, objective, gradient = NULL, hessian = NULL, :
## NA/NaN function evaluation

## Warning in (function (start, objective, gradient = NULL, hessian = NULL, :
```

```
## NA/NaN function evaluation
```

```
# same warnings, and same cautions re: Wald
diagnose(mod_beh_full_poisson)
```

```
##
## predictors with unusually large or small standard deviations (|log10(sd)|>6956.52):
##
## standard.length
##        6956.519
## Predictor variables with very narrow or wide ranges generally give rise
## to parameters with very large or small magnitudes, which can sometimes
## exacerbate numerical instability, and may also be appear (incorrectly)
## to be indicating a poorly defined optimum (i.e., a non-positive
## definite Hessian
##
##
## Unusually large Z-statistics (|x|>5):
##
##         (Intercept)           totalpara           speciesPL     standard.length
##           209.030657          -21.803727          -36.830962           -8.284822
## totalpara:speciesPL      zi~(Intercept)
##            31.984624           -6.294029
##
## Large Z-statistics (estimate/std err) suggest a *possible* failure of
## the Wald approximation - often also associated with parameters that are
## at or near the edge of their range (e.g. random-effects standard
## deviations approaching 0).  (Alternately, they may simply represent
## very well-estimated parameters; intercepts of non-centered models may
## fall in this category.) While the Wald p-values and standard errors
## listed in summary() may be unreliable, profile confidence intervals
## (see ?confint.glmmTMB) and likelihood ratio test p-values derived by
## comparing models (e.g. ?drop1) are probably still OK.  (Note that the
## LRT is conservative when the null value is on the boundary, e.g. a
## variance or zero-inflation value of 0 (Self and Liang 1987; Stram and
## Lee 1994; Goldman and Whelan 2000); in simple cases these p-values are
## approximately twice as large as they should be.)
```

```
check_overdispersion(mod_beh_full_nbin) # data is not over dispersed
```

```
## # Overdispersion test
##
##  dispersion ratio = 0.808
##            p-value = 0.416
```

```
## No overdispersion detected.
```

```
check_overdispersion(mod_beh_full_poisson) # data is over dispersed
```

```
## # Overdispersion test
##
##  dispersion ratio =   3.730
##            p-value = < 0.001
```

```
## Overdispersion detected.
```

```
lrtest(mod_beh_full_poisson, mod_beh_full_nbin)
```

```
## Likelihood ratio test
##
## Model 1: total.time.open ~ totalpara * species + standard.length
## Model 2: total.time.open ~ totalpara * species + standard.length
##   #Df   LogLik Df Chisq Pr(>Chisq)
## 1   6 -10029.6
## 2   7   -768.8  1 18522  < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Looks like the negative binomial distribution is better, since it is not overdispersed. My gut still says this is wrong but I'm going in circles googling things. Let's plow ahead, continuing with analysis like we did for the parasite data to select model. But I am skeptical this is the right model given the warnings.

```
mod_beh_combined_nbin <- glmmTMB(total.time.open ~ totalpara + species + standard.length,
  family = "nbinom1",
  ziformula = ~1,
  data = all_data
)
```

```
## Warning in glmmTMB(total.time.open ~ totalpara + species + standard.length, :
## non-integer counts in a nbinom1 model
```

```
## Warning in (function (start, objective, gradient = NULL, hessian = NULL, :
## NA/NaN function evaluation
```

```
## Warning in (function (start, objective, gradient = NULL, hessian = NULL, :
## NA/NaN function evaluation
```

```
# let's compare these with lrtest
lrtest(mod_beh_full_nbin, mod_beh_combined_nbin) # interesting! There is a sig difference, so the inter
```

```
## Likelihood ratio test
##
## Model 1: total.time.open ~ totalpara * species + standard.length
## Model 2: total.time.open ~ totalpara + species + standard.length
##   #Df  LogLik Df  Chisq Pr(>Chisq)
## 1   7 -768.78
## 2   6 -773.19 -1 8.8225   0.002975 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Ok so the full, interaction model is a better fit. Let's investigate what this model has to tell us.

```
summary(mod_beh_full_nbin) # ok, looks like there might be something here.
```

```
##  Family: nbinom1  ( log )
## Formula:          total.time.open ~ totalpara * species + standard.length
## Zero inflation:                     ~1
## Data: all_data
##
##      AIC      BIC   logLik deviance df.resid
##   1551.6   1571.7   -768.8   1537.6      125
##
##
```

```
## Dispersion parameter for nbinom1 family ():  240
##
## Conditional model:
##                       Estimate Std. Error z value Pr(>|z|)
## (Intercept)          5.996e+00  3.694e-01  16.234  < 2e-16 ***
## totalpara           -2.322e-03  1.161e-03  -2.000 0.045536 *
## speciesPL           -5.668e-01  1.663e-01  -3.409 0.000652 ***
## standard.length      1.727e-06  1.011e-05   0.171 0.864397
## totalpara:speciesPL  4.757e-03  1.600e-03   2.973 0.002951 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Zero-inflation model:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)   -1.375      0.219  -6.278 3.43e-10 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Ok so our full conditional model returns significant effects for total number of parasites, species, and an interaction between parasite load and species.

The zero-inflation model only returns significance for the intercept.

Let's get into some post-hoc testing with emmeans.

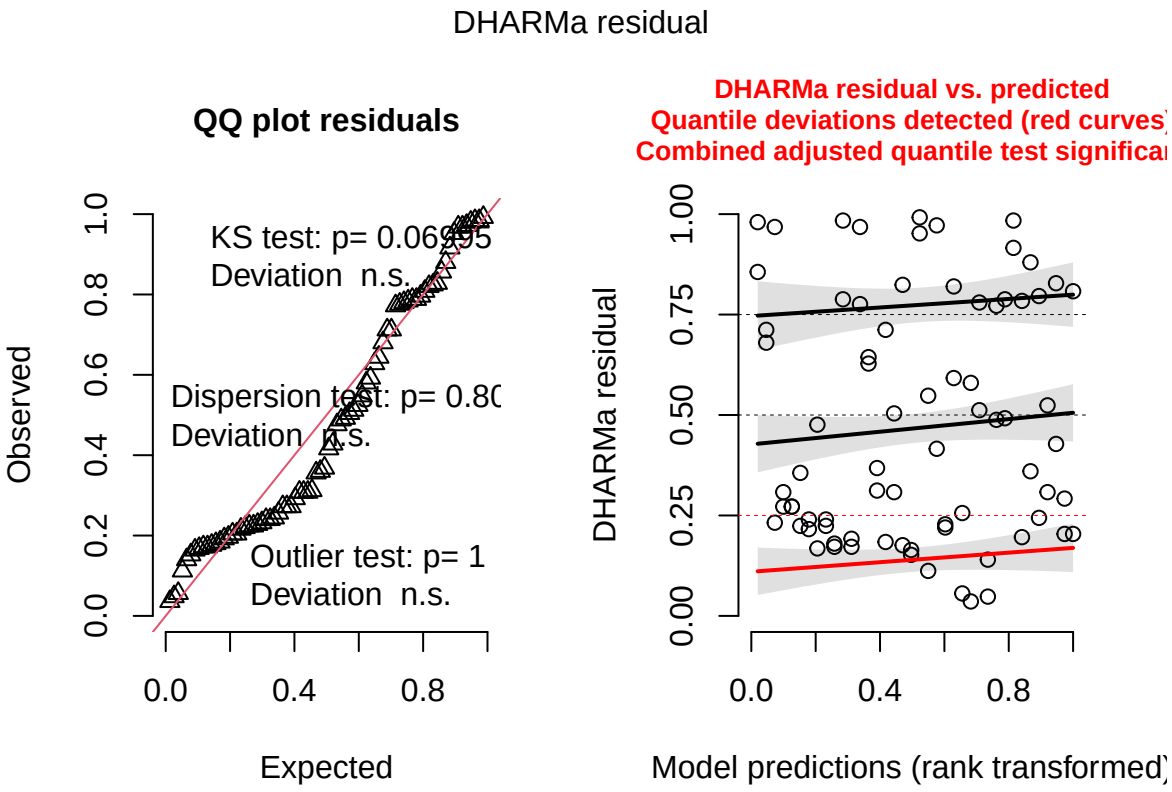
```
# post-hoc with emmeans

# species by parasite
emm_sp <- emmeans::emmeans(mod_beh_full_nbin,
  specs = pairwise ~ species | totalpara,
  type = "response"
)
emm_sp
```

```
## $emmeans
## totalpara = 53.5:
##  species response   SE  df asymp.LCL asymp.UCL
##  PF           377 43.6 Inf       300       473
##  PL           276 30.3 Inf       222       342
##
## Confidence level used: 0.95
## Intervals are back-transformed from the log scale
##
## $contrasts
## totalpara = 53.5151515151515:
##  contrast ratio  SE  df null z.ratio p.value
##  PF / PL   1.37 0.2 Inf    1   2.136  0.0326
##
## Tests are performed on the log scale
```

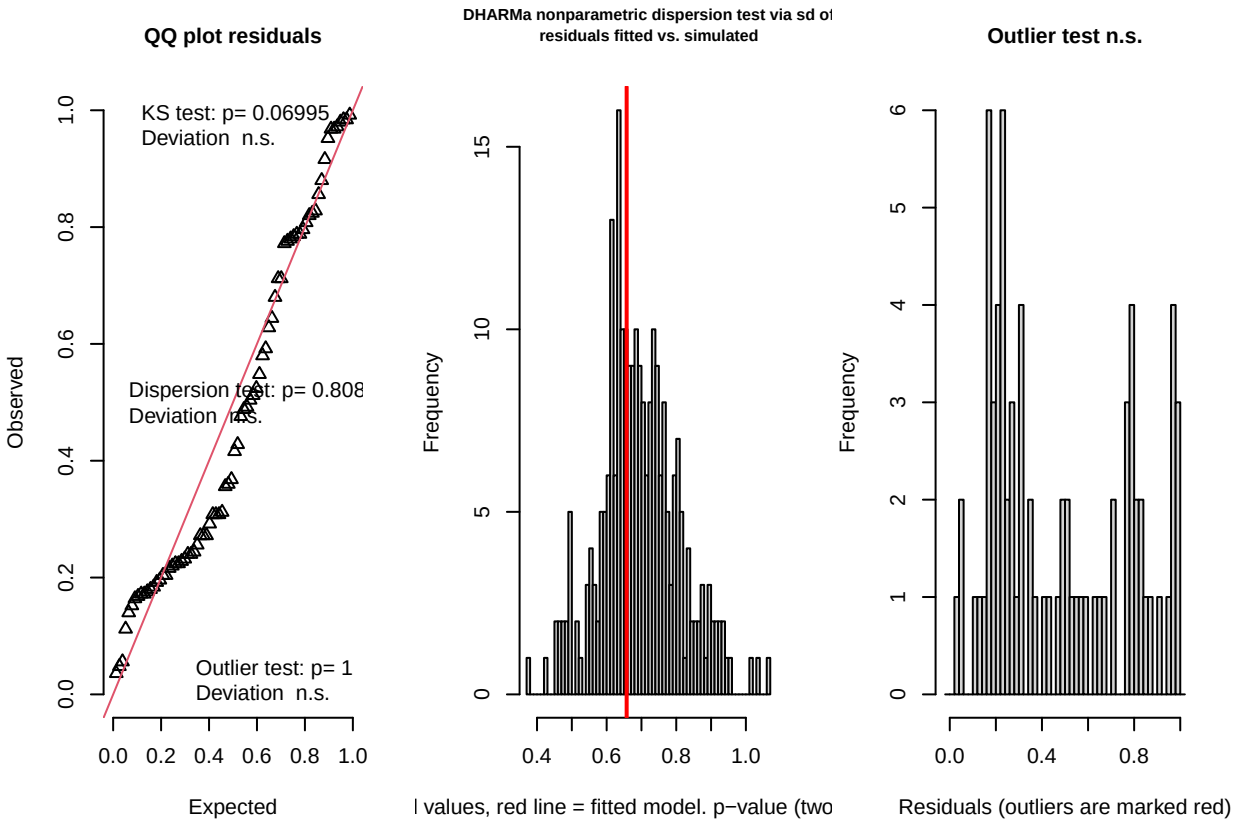
Weslaco Does parasite load predict the amount of time spent in the open at just the Weslaco site?

```
# just Weslaco data
all_data_wes <- all_data %>%
  filter(site.id == "Weslaco")

mod_beh_full_wes <- lm(total.time.open ~ totalpara * species + standard.length,
```

```
  data = all_data_wes
)

# to run all the tests in DHARMa, you first have to simulate your residuals
sim_mod_beh_full_wes <- DHARMa::simulateResiduals(mod_beh_full_wes)

# then you can plot them
plot(sim_mod_beh_full_wes) # QQ looks fine, but resid vs. predicted not so great? The combined adjusted
```



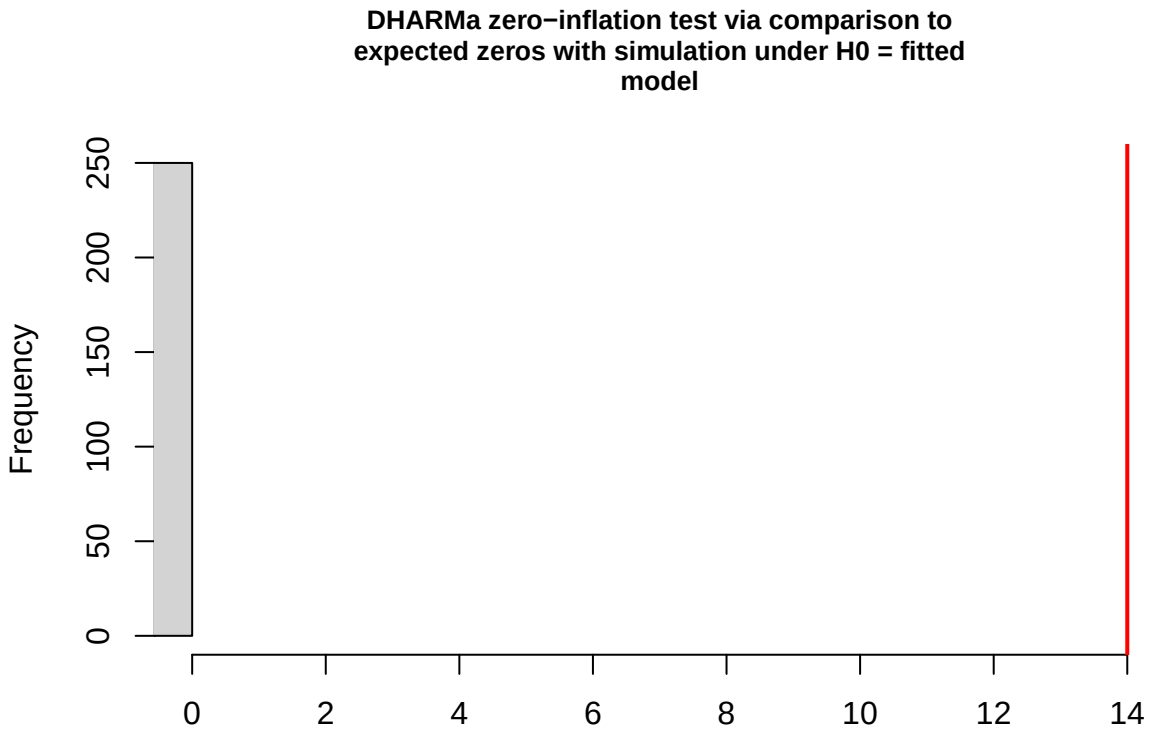

```
# gives you a bunch of tests of dispersion etc
DHARMa::testResiduals(sim_mod_beh_full_wes) # these all look fine
```




```
## $uniformity
##
##  Asymptotic one-sample Kolmogorov-Smirnov test
##
## data:  simulationOutput$scaledResiduals
## D = 0.14853, p-value = 0.06995
## alternative hypothesis: two-sided
##
##
## $dispersion
##
##  DHARMa nonparametric dispersion test via sd of residuals fitted vs.
##  simulated
##
## data:  simulationOutput
## dispersion = 0.94518, p-value = 0.808
## alternative hypothesis: two.sided
##
##
## $outliers
##
##  DHARMa outlier test based on exact binomial test with approximate
##  expectations
##
## data:  simulationOutput
## outliers at both margin(s) = 0, observations = 76, p-value = 1
## alternative hypothesis: true probability of success is not equal to 0.007968127
## 95 percent confidence interval:
```

```
##  0.00000000 0.04737875
## sample estimates:
## frequency of outliers (expected: 0.00796812749003984 )
##                                                      0
```

```
# since we know from previous stuff that the data is zero inflated, let's test that just to be sure
DHARMa::testZeroInflation(sim_mod_beh_full_wes) # yep, zero inflated.
```




```
##
##  DHARMa zero-inflation test via comparison to expected zeros with
##  simulation under H0 = fitted model
##
## data:  simulationOutput
## ratioObsSim = Inf, p-value < 2.2e-16
## alternative hypothesis: two.sided
```

Ok so no surprise, Weslaco data is also zero inflated.

```
mod_beh_full_wes_nbin <- glmmTMB(total.time.open ~ totalpara * species + standard.length,
  family = "nbinom1",
  ziformula = ~1,
  data = all_data_wes
)
```

```
## Warning in glmmTMB(total.time.open ~ totalpara * species + standard.length, :
## non-integer counts in a nbinom1 model
```

```
## Warning in (function (start, objective, gradient = NULL, hessian = NULL, :
## NA/NaN function evaluation
```

```
## Warning in (function (start, objective, gradient = NULL, hessian = NULL, :
```

```
## NA/NaN function evaluation
```

```
# same warnings as before, plowing ahead for now

mod_beh_combined_wes_nbin <- glmmTMB(total.time.open ~ totalpara + species + standard.length,
  family = "nbinom1",
  ziformula = ~1,
  data = all_data_wes
)
```

```
## Warning in glmmTMB(total.time.open ~ totalpara + species + standard.length, :
## non-integer counts in a nbinom1 model
```

```
## Warning in (function (start, objective, gradient = NULL, hessian = NULL, :
## NA/NaN function evaluation

## Warning in (function (start, objective, gradient = NULL, hessian = NULL, :
## NA/NaN function evaluation
```

```
# let's compare these with lrtest
lrtest(mod_beh_full_wes_nbin, mod_beh_combined_wes_nbin) # ok, no difference.
```

```
## Likelihood ratio test
##
## Model 1: total.time.open ~ totalpara * species + standard.length
## Model 2: total.time.open ~ totalpara + species + standard.length
##   #Df  LogLik Df  Chisq Pr(>Chisq)
## 1   7 -444.62
## 2   6 -444.67 -1 0.1102     0.7399
```

No difference between the combined model and the interaction model. Let's move on with backwards model selection with the combined model.

```
mod_beh_sp_wes_nbin <- glmmTMB(total.time.open ~ species + standard.length,
  family = "nbinom1",
  ziformula = ~1,
  data = all_data_wes
)
```

```
## Warning in glmmTMB(total.time.open ~ species + standard.length, family =
## "nbinom1", : non-integer counts in a nbinom1 model
```

```
## Warning in (function (start, objective, gradient = NULL, hessian = NULL, :
## NA/NaN function evaluation

## Warning in (function (start, objective, gradient = NULL, hessian = NULL, :
## NA/NaN function evaluation
```

```
# same non-integer warnings
# lrtest(mod_beh_combined_wes_nbin, mod_beh_sp_wes_nbin) # ??? no chisq or pvalues??

mod_beh_para_wes_nbin <- glmmTMB(total.time.open ~ totalpara + standard.length,
  family = "nbinom1",
  ziformula = ~1,
  data = all_data_wes
)
```

```
## Warning in glmmTMB(total.time.open ~ totalpara + standard.length, family =
## "nbinom1", : non-integer counts in a nbinom1 model
```

```
## Warning in (function (start, objective, gradient = NULL, hessian = NULL, :
## NA/NaN function evaluation

## Warning in (function (start, objective, gradient = NULL, hessian = NULL, :
## NA/NaN function evaluation

## Warning in finalizeTMB(TMBStruc, obj, fit, h, data.tmb.old): Model convergence
## problem; non-positive-definite Hessian matrix. See vignette('troubleshooting')
```

```
# hmm model convergence problem here, not sure what to do next. Trying to avoid rabbit holes, but will

# I'm going to just move forward with the combined model for now
summary(mod_beh_combined_wes_nbin)
```

```
##  Family: nbinom1  ( log )
## Formula:          total.time.open ~ totalpara + species + standard.length
## Zero inflation:                     ~1
## Data: all_data_wes
##
##      AIC      BIC   logLik deviance df.resid
##    901.3    915.3   -444.7    889.3       70
##
##
## Dispersion parameter for nbinom1 family ():  163
##
## Conditional model:
##                   Estimate Std. Error z value Pr(>|z|)
## (Intercept)      6.407e+00  5.137e-01  12.472  < 2e-16 ***
## totalpara        8.328e-03  3.386e-03   2.460   0.0139 *
## speciesPL       -6.600e-01  1.694e-01  -3.896 9.76e-05 ***
## standard.length -1.159e-05  1.376e-05  -0.842   0.3998
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Zero-inflation model:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -1.4978     0.2985  -5.017 5.24e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Ok so our full conditional model returns significant effects for total number of parasites, species, and an interaction between parasite load and species.

The zero-inflation model only returns significance for the intercept.

Let's get into some post-hoc testing with emmeans.

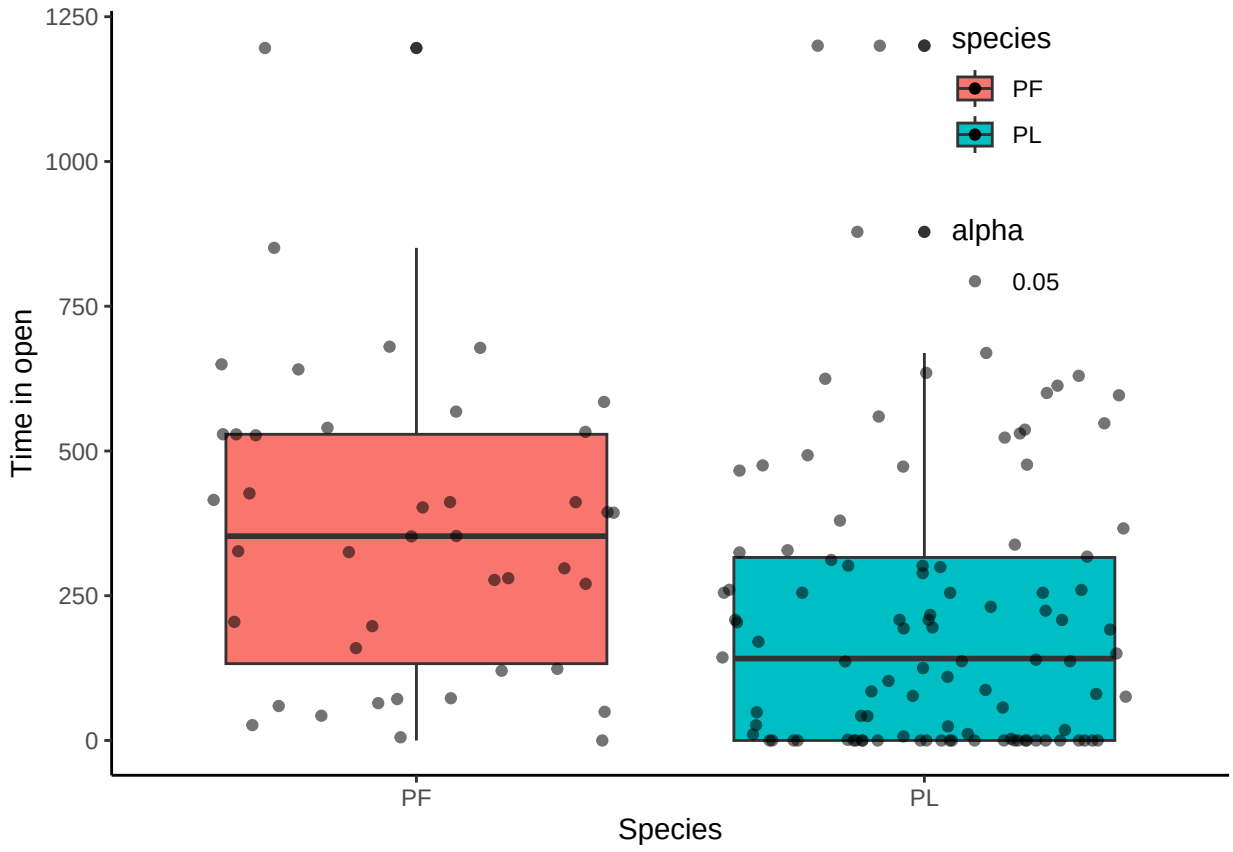
```
# post-hoc with emmeans

# main effect of species
emm_sp_wes <- emmeans::emmeans(mod_beh_combined_wes_nbin,
                           specs = pairwise ~ species | totalpara,
                           type = "response"
                           )
emm_sp_wes
```

```
## $emmeans
## totalpara = 6.5:
```

```
##  species response   SE  df asymp.LCL asymp.UCL
##  PF           420 47.3 Inf       337       524
##  PL           217 30.8 Inf       165       287
##
## Confidence level used: 0.95
## Intervals are back-transformed from the log scale
##
## $contrasts
## totalpara = 6.5:
##  contrast ratio    SE  df null z.ratio p.value
##  PF / PL   1.93 0.328 Inf    1   3.896  0.0001
##
## Tests are performed on the log scale
```

Figures

```
# diff in time spent in open by species
beh_spp_boxplot <- all_data %>%
  ggplot(mapping = aes(
    x = species,
    y = total.time.open,
    fill = species
  )) +
  geom_boxplot() +
  geom_jitter(aes(
    alpha = 0.05,
    fill = species
  )) +
  xlab("Species") +
  ylab("Time in open") +
  theme_classic() +
  theme(legend.position = c(0.8, 0.8))
beh_spp_boxplot
```



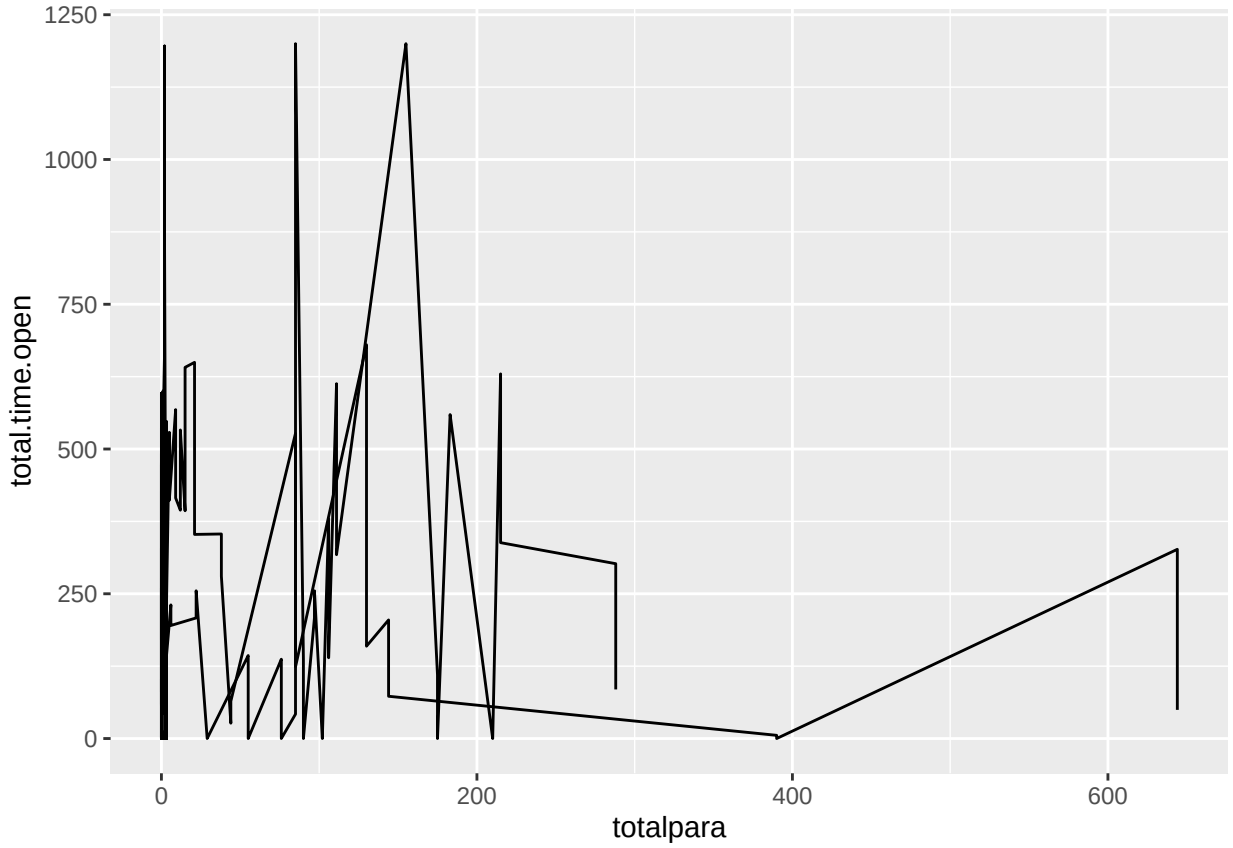
All data

```
beh_spp_para <- all_data %>%
  ggplot(mapping = aes(
    x = totalpara,
    y = total.time.open,
    fill = species
  )) +
  geom_line(aes(fill = species))
```

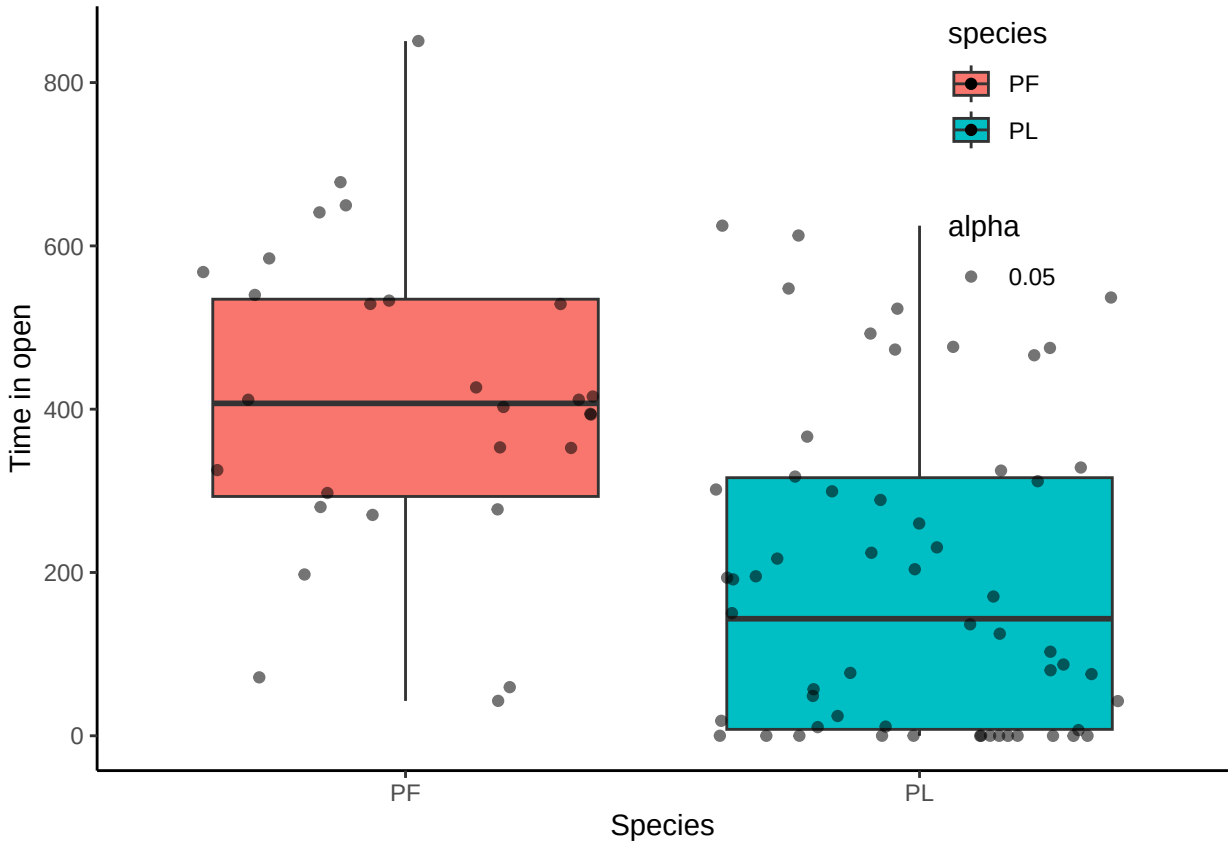
```
## Warning in geom_line(aes(fill = species)): Ignoring unknown aesthetics: fill
```

```
beh_spp_para
```

```
## Warning: Removed 12 rows containing missing values (`geom_line()`).
```




```
# diff in time spent in open by species
beh_spp_wes_boxplot <- all_data_wes %>%
  ggplot(mapping = aes(
    x = species,
    y = total.time.open,
    fill = species
  )) +
  geom_boxplot() +
  geom_jitter(aes(
    alpha = 0.05,
    fill = species
  )) +
  xlab("Species") +
  ylab("Time in open") +
  theme_classic() +
  theme(legend.position = c(0.8, 0.8))
beh_spp_wes_boxplot
```



Weslaco data

More Figs & Tables

Parasite data total counts of each species by site

```
parasite_data_wes %>%
  group_by(species) %>%
  tally()
```

```
## # A tibble: 2 x 2
##   species       n
##   <fct>     <int>
## 1 formosa      58
## 2 latipinna    32
```

```
parasite_data_br <- parasite_data %>%
  filter(site.id == "Brownsville")

parasite_data_br %>%
  group_by(species) %>%
  tally()
```

```
## # A tibble: 2 x 2
##   species       n
##   <fct>     <int>
## 1 formosa      11
## 2 latipinna    27
```

Behavior data total counts of each species by site

```r
all_data_wes <- all_data %>%
  filter(site.id == "Weslaco")

all_data_wes %>%
  group_by(species) %>%
  tally()
```

```
## # A tibble: 2 x 2
##   species     n
##   <chr>   <int>
## 1 PF         28
## 2 PL         58
```

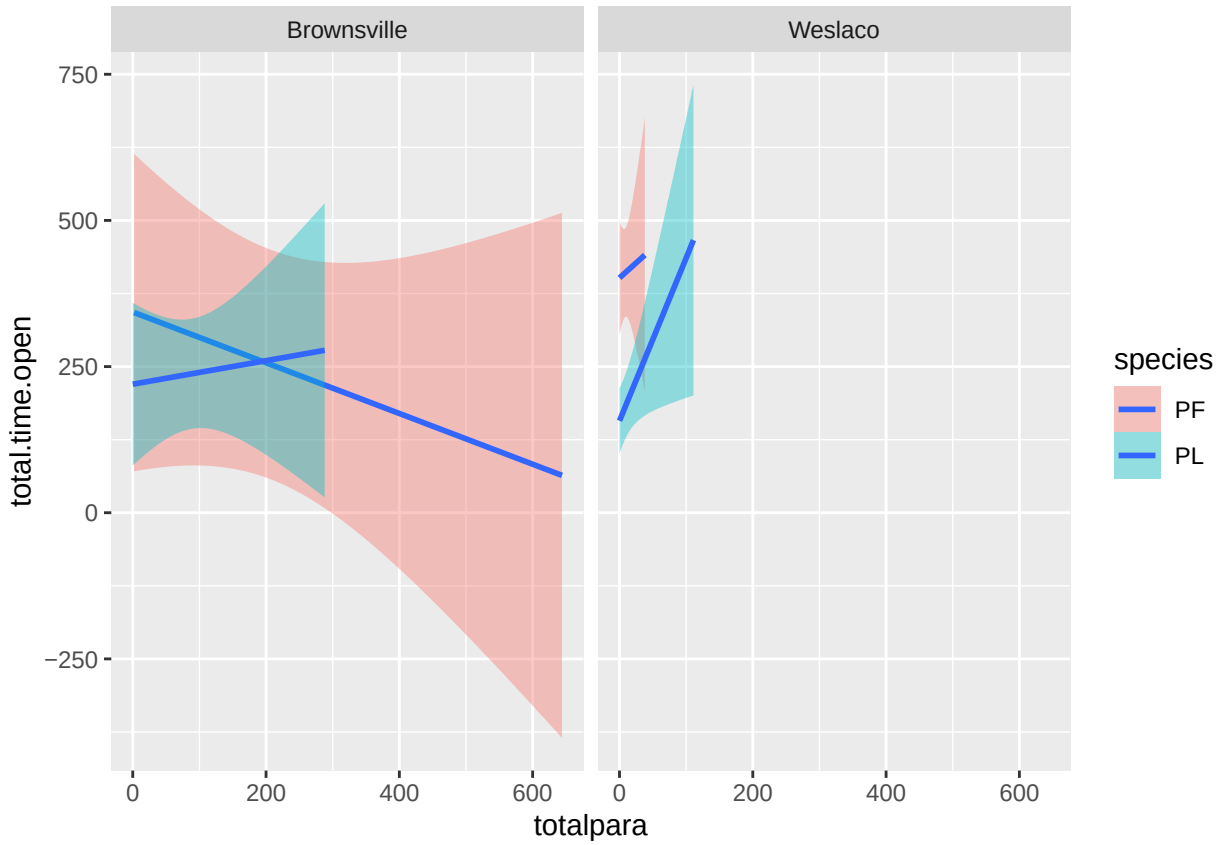
```r
all_data_br <- all_data %>%
  filter(site.id == "Brownsville")

all_data_br %>%
  group_by(species) %>%
  tally()
```

```
## # A tibble: 2 x 2
##   species     n
##   <chr>   <int>
## 1 PF         14
## 2 PL         44
```

```r
# This plots number of parasites vs time spent in the open. Each line represents a species.
# I've also split the plot into the two sites: Brownsville and Weslaco.
pairwise_species_plot <- all_data %>%
  ggplot(mapping = aes(
    y = total.time.open,
    x = totalpara,
    fill = species
  )) +
  geom_smooth(method = "lm") +
  facet_wrap(vars(site.id))
pairwise_species_plot
```

```
## `geom_smooth()` using formula = 'y ~ x'
```

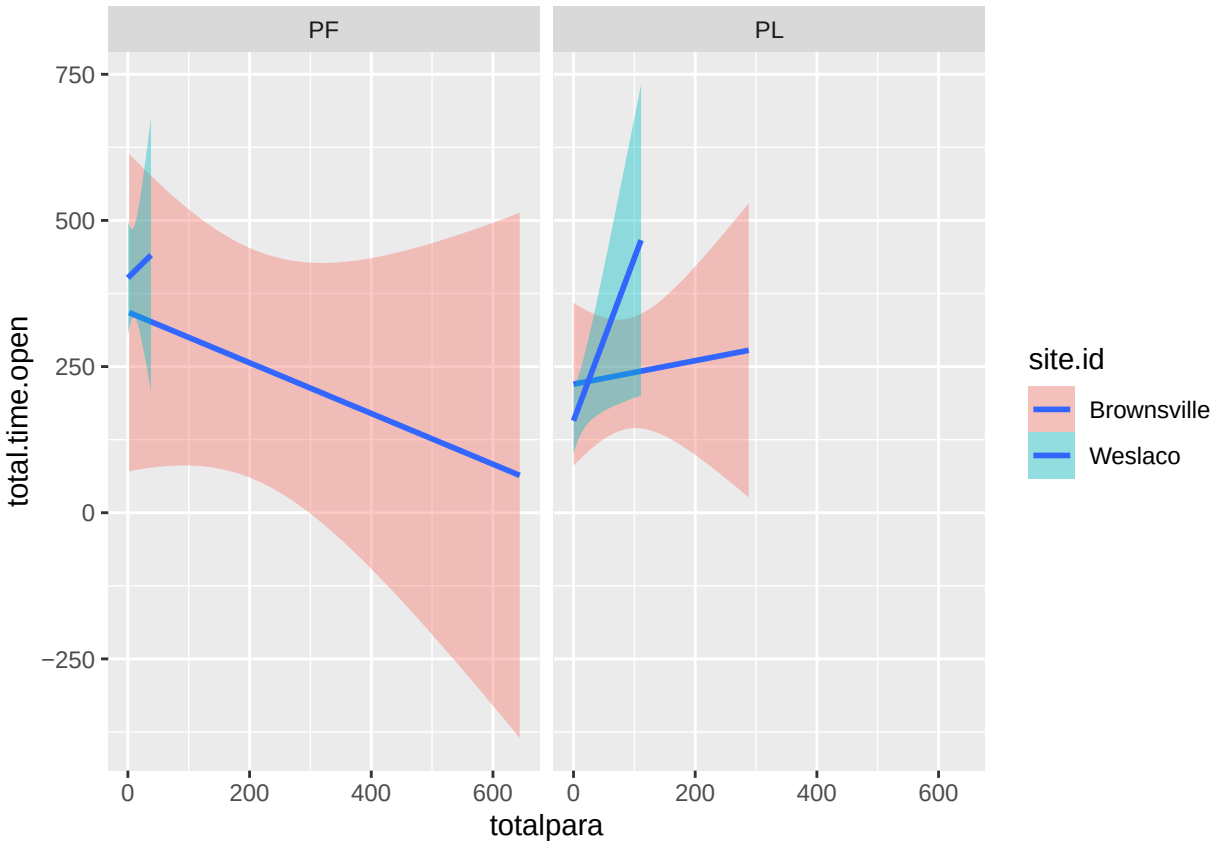
```
## Warning: Removed 12 rows containing non-finite values (`stat_smooth()`).
```




```
pairwise_site_plot <- all_data %>%
  ggplot(mapping = aes(
    y = total.time.open,
    x = totalpara,
    fill = site.id
  )) +
  geom_smooth(method = "lm") +
  facet_wrap(vars(species))
pairwise_site_plot
```

```
## `geom_smooth()` using formula = 'y ~ x'
```

```
## Warning: Removed 12 rows containing non-finite values (`stat_smooth()`).
```




```
# note from Kate: add raw data points
```

SCRAPS

I'm hashing out all of this so that the pdf knits.

KATE NOTES

parasites bw sp We'll probably want to use DHARMa to make sure our residuals look good before jumping into more complicated models. So we'll start with a 'simple' model and then check and see which of our assumptions are violated

And no surprise, as we expected, yes we can see our data is super zero-inflated

```
# para.lm <- lm(totalpara ~ species * site.id, data = parasite_data)
#
# plot(para.lm) # even just using the standard plots you can see variance heterogeneity and major non-n
#
#
# # to run all the tests in DHARMa, you first have to simulate your residuals
# sim.output <- DHARMa::simulateResiduals(para.lm)
#
# # then you can plot them
# plot(sim.output)
#
# # gives you a bunch of tests of dispersion etc
# DHARMa::testResiduals(sim.output)
```

```
#
# # yes we can clearly see that data super zero inflated
# DHARMa::testZeroInflation(sim.output)
```

Let's run the same checks on the zero-inflated models to see if they look better.

Ha, well DHARMa apparently doesn't like this model type.... So I think we just move forward assuming that the zero-inflated model is appropriate

```
# mod_para_interaction <- zeroinfl(totalpara ~ species * site.id,
#   dist = "negbin",
#   lin = "logit",
#   data = parasite_data
# )
#
# sim.ouput <- DHARMa::simulateResiduals(mod_para_interaction) # doesn't work!
# summary(mod_para_interaction)
```

So let's explore this summary. You can see two sets of parameter estimates: "count model" - do our predictors affect the number of parasites we see when they are >0 "zero-inflation" - do our predictors affect the presences of parasites

Looking at this summary there is a significant effect of site on the count of parasites (which is reassuring since we do know that the sites differ in parasite loads!)

(I have no idea what the 'log(theta) terms means... my googling suggests it's some measure of dispersion in our model but I don't feel like I have a good hand on how to interpret/if we need to interpret it)

And yeah no difference in species when only looking at weslaco. Also when only looking at brownsville too.

```
# # combined model
# mod_para_combined <- zeroinfl(totalpara ~ species + site.id,
#   dist = "negbin",
#   lin = "logit",
#   data = parasite_data
# )
#
# summary(mod_para_combined)
#
#
# parasite_data_wes <- parasite_data %>%
#   filter(site.id == "Weslaco")
#
# parasite_data_br <- parasite_data %>%
#   filter(site.id == "Brownsville")
#
#
# # species model
# mod_para_species_wes <- zeroinfl(totalpara ~ species,
#   dist = "negbin",
#   lin = "logit",
#   data = parasite_data_wes
# )
#
# summary(mod_para_species_wes)
# hist(parasite_data_wes$totalpara)
#
```

```
# # species model
# mod_para_species_br <- zeroinfl(totalpara ~ species,
#   dist = "negbin",
#   lin = "logit",
#   data = parasite_data_br
# )
#
# summary(mod_para_species_br)
# hist(parasite_data_br$totalpara)
```

histograms let's make some nice histograms

```
# # basic
# ggplot(parasite_data, aes(x = totalpara, colour = species)) +
#   geom_histogram(alpha = 0.5, position = "identity")
#
# # prettify
# ggplot(parasite_data, aes(x = totalpara, fill = species, colour = species)) +
#   geom_histogram(alpha = 0.3, position = "identity", binwidth = 10) +
#   scale_fill_manual(values = c("#0066FF", "#FF0033")) +
#   scale_colour_manual(values = c("#0066FF", "#FF0033")) +
#   xlab("Total parasite count") +
#   ylab("Frequency") +
#   theme_classic() +
#   theme(legend.position = c(0.8, 0.8))
```

Zero inflated models are hungry so maybe we just create new column that bins parasites into 0/1

```
# parasite_data <- parasite_data %>%
#   mutate(para_yes = ifelse(totalpara > 0, 1, 0))
#
# mod_para_log <- glm(para_yes ~ species * site.id, family = "binomial", data = parasite_data)
#
# sim.output <- DHARMa::simulateResiduals(mod_para_log)
# plot(sim.output)
#
# summary(mod_para_log)
```

Yeahhh.... absolutely no difference in parasite presence/absence. I think we're safe with saying that these two species are the same in parasite loads/presences etc. We can present our zero-inflated model and our logistic regression and the histogram graph as support of this.

CLOSE NOTES

Parasites x behavior

I want to see if the time spent hiding is predicted by parasites, species, trial number, fish.ID or their interaction.

```
# # full model
# mod_full <- lmer(duration.Hiding ~ total.parasites * species * trial + (1 | fish.ID),
#   data = total_hide_open
# )
# summary(mod_full)
#
```

```
# ## evaluating assumptions
# ggResidpanel::resid_panel(mod_full)
```

Looking at our residual panel, most assumptions look ok! Residuals vs predicted might be a bit trumet-y? Q-Q looks nice and linear. Index plot is an even scatter. Residual histogram looks pretty normal.

```
# # decompose to two way
# mod_2way <- lmer(duration.Hiding ~ total.parasites:species + total.parasites:trial + species:trial +
#   data = total_hide_open
# )
# summary(mod_2way)
# ## evaluating assumptions
# ggResidpanel::resid_panel(mod_2way)
```

```
# anova(mod_full, mod_2way)
```

Ok now I'm going to make a plot to disentangle the pairwise interaction we've got going on. ## Old Summary data code Now let's create some columns for summary data (e.g. total time hiding)

```
# # time hiding per trial
# total_hiding <- boris_data_cutoff %>%
#   aggregate(
#     duration.Hiding ~ fish.id + trial.id,
#     sum
#   )
#
# total_hiding <- total_hiding %>%
#   mutate(duplicate.fish.id = fish.id)
#
# total_hiding <- total_hiding %>%
#   separate_wider_delim(duplicate.fish.id,
#     delim = "-",
#     names = c(
#       "species",
#       "junk.num"
#     )
#   )
#
# total_hiding <- total_hiding %>%
#   dplyr::select(-junk.num)
#
# # time in open per trial
# total_open <- boris_data_cutoff %>%
#   aggregate(
#     duration.Open ~ fish.id + trial.id,
#     sum
#   )
#
# total_open <- total_open %>%
#   mutate(duplicate.fish.id = fish.id)
#
# total_open <- total_open %>%
#   separate_wider_delim(duplicate.fish.id,
#     delim = "-",
#     names = c(
```

```
#       "species",
#       "junk.num"
#     )
#   )
#
# total_open <- total_open %>%
#   dplyr::select(-junk.num)
```

Now, I want to merge total hiding and open per fish per trial.

```
# # I'm going to create a column with both fish ID and trial to create a unique
# # row for each fish/trial combination. I'll then use this to join the hiding
# # and open datasets
# total_open <- total_open %>%
#   unite(fish.ID.trial, c(fish.ID, trial.ID))
#
# total_hiding <- total_hiding %>%
#   unite(fish.ID.trial, c(fish.ID, trial.ID))
#
# # merge
# total_hide_open <- total_hiding %>%
#   left_join(total_open, by = "fish.ID.trial")
#
# # get rid of duplicate columns
# total_hide_open <- total_hide_open %>%
#   dplyr::select(-species.y)
#
# # separate fish ID and trial
# total_hide_open <- total_hide_open %>%
#   separate_wider_delim(fish.ID.trial,
#     delim = "_",
#     names = c(
#       "fish.ID",
#       "trial"
#     )
#   )
#
# # and for my own sanity, renaming the species column
# total_hide_open <- total_hide_open %>%
#   rename(species = species.x)
#
# # time hiding and open by trial
# total_hide_open <- total_hide_open %>%
#   mutate(site.ID = boris_data_cutoff$site.ID[match(fish.ID, boris_data_cutoff$fish.ID)])
#
# # hiding and open by trial, joined with parasite and size data
# total_hide_open <- total_hide_open %>%
#   left_join(parasite_data, by = "fish.ID", relationship = "many-to-many")
#
# total_hide_open <- total_hide_open %>%
#   left_join(length_data, by = "fish.ID", relationship = "many-to-many")
#
# total_hide_open <- total_hide_open %>%
#   dplyr::select(
```

```
#     -site.ID.y,
#     -species.y,
#     -site.ID,
#   )
#
# # rename columns
# total_hide_open <- total_hide_open %>% rename(
#   species = species.x,
#   site.ID = site.ID.x,
#   total.parasites = totalpara
# )
```

Extra plots

There were lots of other plots I made examining site by site or trial by trial patterns as well. They are copied below, but have not been reviewed/error checked.

```
# # change in hiding over trials by species
# trial_hiding <- total_hide_open %>%
#   ggplot(mapping = aes(
#     x = trial.ID,
#     y = duration_Hiding,
#     fill = species,
#     dodge = species
#   )) +
#   geom_boxplot()
#
# # change in hiding over trials by species
# trial_open <- total_hide_open %>%
#   ggplot(mapping = aes(
#     x = trial.ID,
#     y = duration_Open,
#     fill = species,
#     dodge = species
#   )) +
#   geom_boxplot()
#
# # diff in open between sites
# site_open <- total_hide_open %>%
#   ggplot(mapping = aes(
#     x = site.ID.x,
#     y = duration_Open
#   )) +
#   geom_boxplot()
#
# # diff in hiding between sites
# site_hiding <- total_hide_open %>%
#   ggplot(mapping = aes(
#     x = site.ID.x,
#     y = duration_Hiding
#   )) +
#   geom_boxplot()
# # diff in opn between sites by species
# site_spp_open <- total_hide_open %>%
```

```
#   ggplot(mapping = aes(
#     x = site.ID.x,
#     y = duration_Open,
#     fill = species.x,
#     dodge = species.x
#   )) +
#   geom_boxplot()
#
# # diff in hiding between sites by species
# site_spp_hiding <- total_hide_open %>%
#   ggplot(mapping = aes(
#     x = site.ID.x,
#     y = duration_Hiding,
#     fill = species.x,
#     dodge = species.x
#   )) +
#   geom_boxplot()
#
# # diff in total parasites by species
# parasites_spp <- total_hide_open %>%
#   ggplot(mapping = aes(
#     x = species.x,
#     y = totalpara,
#     fill = species.x
#   )) +
#   geom_boxplot()
#
# # diff in total parasites by site
# parasites_site <- total_hide_open %>%
#   ggplot(mapping = aes(
#     x = site.ID.x,
#     y = totalpara
#   )) +
#   geom_boxplot()
#
# # diff in total parasites by site and spp
# parasites_site_spp <- total_hide_open %>%
#   ggplot(mapping = aes(
#     x = site.ID.x,
#     y = totalpara,
#     fill = species.x,
#     dodge = species.x
#   )) +
#   geom_boxplot()
#
# # variation in parasites with open beh
# parasites_open <- total_hide_open %>%
#   ggplot(mapping = aes(
#     x = totalpara,
#     y = duration_Open,
#     color = species.x
#   )) +
#   geom_point()
```

```
#
# # variation in parasites with hiding beh
# parasites_hiding <- total_hide_open %>%
#   ggplot(mapping = aes(
#     x = totalpara,
#     y = duration_Hiding,
#     fill = species.x
#   )) +
#   geom_smooth()
#
# # avg length at each site by spp.
# length_site <- total_hide_open %>%
#   ggplot(mapping = aes(
#     x = site.ID.x,
#     y = total_length..mm.,
#     fill = species.x,
#     dodge = species.x
#   )) +
#   geom_boxplot()
```