Texas2022_analysis_working

Kirsten Sheehy

2025-09-04

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

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

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

```
## 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
```

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

```
## 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 with other datasets.

Track down some quirks from data entry (e.g. typos) and get formatting consistent.

```
## '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  "" "P. formosa" "" "P. 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 " ...
```

```
## '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 "","P. formosa",..: 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      : num  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 " ...
```

Length Data

These measurements were done by Nishika and I 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 pixels, 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.

Let's get the column names consistent with the other datasets.

Now, let's manipulate the data so that the lengths are all in their own columns (one row per fish). Remember, the collection information and fish.id are encoded in the 10000 pixel length name (e.g. br-op_06aug_pl-1_f). We can associate the three measurements by the image file name used in QuPath (e.g. IMG_20220807_155019762.jpg)

Now, let's split up the fish.id column into it's various components: site.id, date, fish.id, and sex (e.g. br-op_06aug_pl-1_f becomes Brownsville, 2022-08-06, PL-1, and F).

Let's also get these column names to match the rest of the 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_202208
##  $ 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" ...
```

there are two fish in the length data that need more specific names due to an error in how we initially recorded them in our notebooks. There are two PF-08s, one for Brownsville and one for Weslaco. The Brownsville one eventually became PF-08BR. There is also PF-25B in the boris data. I'm not sure why it was recorded that way (assuming there were two by mistake), but it is in the Weslaco site. I'm changing both of these in the length data to match the Boris data.

ID Data

This is the data from my lab notebook.

We just need to rename the columns to match other datasets.

Boris Data

This tidys up the BORIS data that Nishika collected from our videos. Basically, she recorded when the fish was on the open or sheltered half of the arena, as well as when the "startle stimulus" was applied. The startle was us slapping the water with a pool noodle.

First, let's remove unnecessary ones, create all the columns we need, and rename them to match the other datasets.

Now, let's get the values in each column formatted correctly and fix any typos.

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.

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

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

Quick note: we lose some data when we merge the id.data and boris_data_wide. It seems that we are missing five video.ids in boris_data_merge. This is probably from when we removed the third trial, which we did not do for the id_data.

Now we tidy the merged data.

Now, I need to standardize the trial times. When Nishika and I were observing, we sometimes recorded behaviors for longer than the prescribed 10 minutes. The following code finds the earliest behavior observation (either start.hiding or start.open) and then cuts off any observations 10 minutes after the startle.

```
## Warning: There were 26 warnings in `mutate()`.
## The first warning was:
## i In argument: `earliest.open = min(start.Open, na.rm = TRUE)`.
## i In group 24: `fish.id = "PF-37"`, `trial.id = "2"`.
## Caused by warning in `min()`:
## ! no non-missing arguments to min; returning Inf
## i Run `dplyr::last_dplyr_warnings()` to see the 25 remaining warnings.
```

Now, I am going to replace any end times that go past the prescribed trial end time (10 minutes, or 600 seconds). Nishika and I often accidentally just watched the video longer than we needed to.

```
# now, I need to create an 'end cap' value to replace any 'stop' behaviors that went past the trial end
# 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)
```

Now, I need to id observations that are pre vs. post startle. If they start pre-startle and end post-startle, I'll need to chop up the observation so that there are now two–one pre- and one post-startle.

```
# I need to separate data into before, after, and bridging the cue start time
boris_data_cutoff <- boris_data_cutoff %>%
  group_by(fish.id, trial.id) %>%
  rowwise() %>%
  mutate(
    category = case_when(
      stop.Hiding <= start.Startle ~ "before",
      stop.Open <= start.Startle ~ "before",
      start.Hiding >= start.Startle ~ "after",
      start.Open >= start.Startle ~ "after",
      TRUE ~ "bridge"
    )
  )

# need to compute before/after times to split up the bridging observations
before_startle_data <- boris_data_cutoff %>%
  filter(category == "bridge") %>%
  mutate(stop.Open = start.Startle) %>%
  mutate(stop.Hiding = start.Startle) %>%
  mutate(duration.Hiding = (stop.Hiding - start.Hiding)) %>%
  mutate(duration.Open = (stop.Open - start.Open))

after_startle_data <- boris_data_cutoff %>%
  filter(category == "bridge") %>%
  mutate(start.Open = start.Startle) %>%
  mutate(start.Hiding = start.Startle) %>%
  mutate(duration.Hiding = (stop.Hiding - start.Hiding)) %>%
  mutate(duration.Open = (stop.Open - start.Open))

# now we bring those all back together, but first remove the bridge data
boris_data_cutoff <- boris_data_cutoff %>%
  filter(category != "bridge")

boris_data_cutoff <- rbind(boris_data_cutoff, before_startle_data)
boris_data_cutoff <- rbind(boris_data_cutoff, after_startle_data)
boris_data_cutoff <- as.data.frame(boris_data_cutoff)
```

Creating some summary data

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

All data

Merge all data into a single dataframe.

```
## Warning in left_join(., parasite_data, by = c("site.id", "fish.id", "species")): Detected an unexpec
## i Row 82 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.
```

We did not have fraction of a second precision in our observations, so I'm removing the decimal for time values to create integers (this will also make some anlysis easier).

```r
all_data$total.time.hiding.b4 <- round(all_data$total.time.hiding.b4, 0)
all_data$total.time.hiding.after <- round(all_data$total.time.hiding.after, 0)
all_data$total.time.open.b4 <- round(all_data$total.time.open.b4, 0)
all_data$total.time.open.after <- round(all_data$total.time.open.after, 0)
all_data$total.time.b4 <- round(all_data$total.time.b4, 0)
all_data$total.time.after <- round(all_data$total.time.after, 0)
```

Pivot the data longer for graphing

```r
# pivoting data so I can plot time before/after together
all_data_long <- all_data %>%
  pivot_longer(
    cols = starts_with("prop"),
    names_to = "prop.type",
    values_to = "proportion",
    values_drop_na = TRUE
  )

# set the prop.type order
all_data_long$prop.type <- factor(all_data_long$prop.type, levels = c("prop.open.b4", "prop.open.after"
```

Inspect Data

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

```
##  [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"
```

```
##  [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"
```

```
##   [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
```

```
##   [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
```

```
##  [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"
```

```
##  [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" "WES_trial1_01_20220808.mov"
## [13] "Wes_trial1_01_20220812.mov" "WES_trial1_02_20220808.mov"
## [15] "WES_trial1_02_20220812.mov" "WES_trial1_03_20220808.mov"
## [17] "WES_trial1_03_20220812.mov" "WES_trial1_04_20220812.mov"
## [19] "WES_trial1_05_20220812.mov" "WES_trial2_01_20220808.mov"
## [21] "WES_trial2_01_20220812.mov" "WES_trial2_02_20220808.mov"
## [23] "WES_trial2_02_20220812.mov" "WES_trial2_03_20220808.mov"
## [25] "WES_trial2_03_20220812.mov" "WES_trial2_04_20220812.mov"
## [27] "WES_trial2_05_20220812.mov" "BR2_trial2_02_20220810.mov"
```

```
##  [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"
```

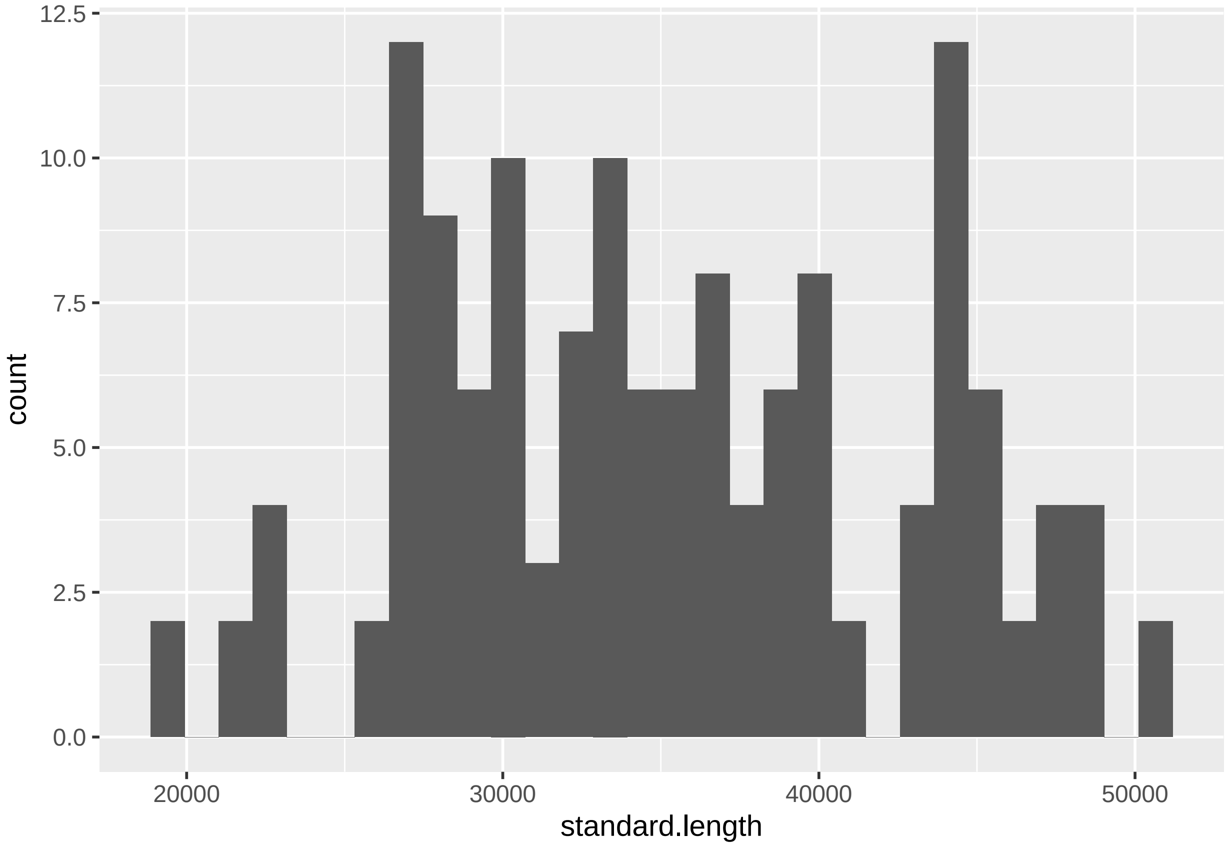
This all makes sense. The number of unique fish IDs from my notebook match up with the boris data. There are more parasite fish IDs because we sent Jessica extra fish that didn't go through trials. Same for fish lengths, more were photographed than went through trials.

We have fewer video IDs in the boris data because we removed trial 3. We removed trial 3 because most fish didn't make it through all three trials.

Now, let's make some histograms.

```r
# length data
length_hist <- all_data %>%
  ggplot(mapping = aes(standard.length)) +
  geom_histogram()
length_hist
```

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

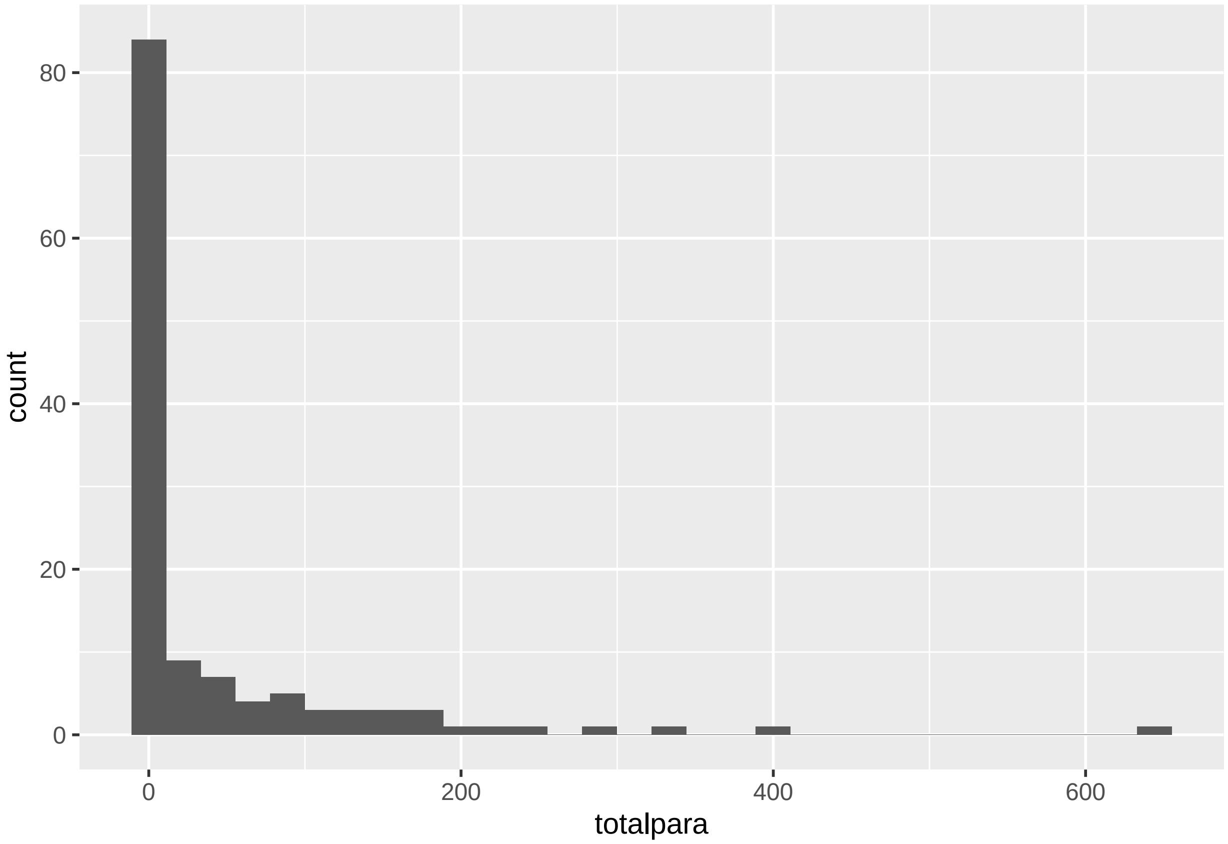


Seems like a fairly normal distribution for length! Two mongo fish were over 5cm!

Let’s take a look at the parasite data.

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

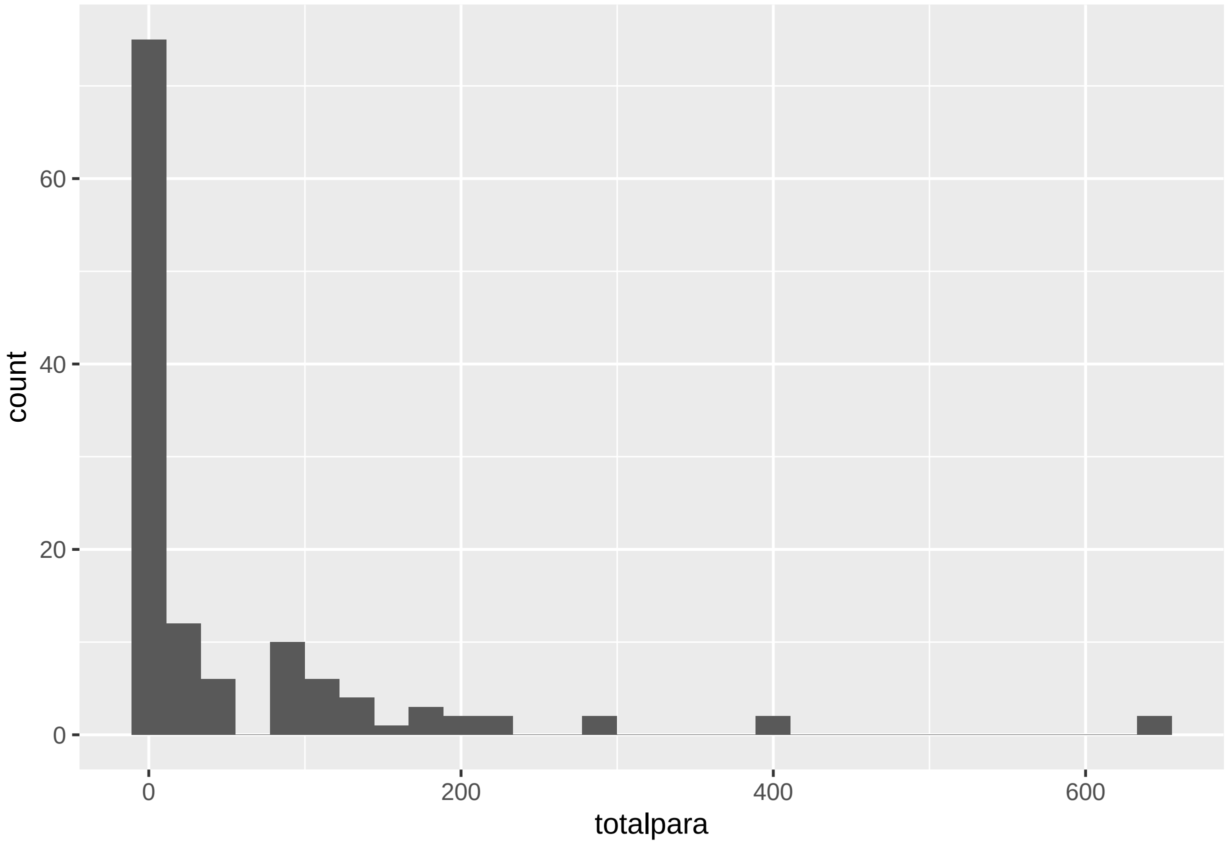
```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



```
# parasite data for fish that went through trials
parasite_beh_hist <- all_data %>%
  ggplot(mapping = aes(totalpara)) +
  geom_histogram()
parasite_beh_hist
```

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

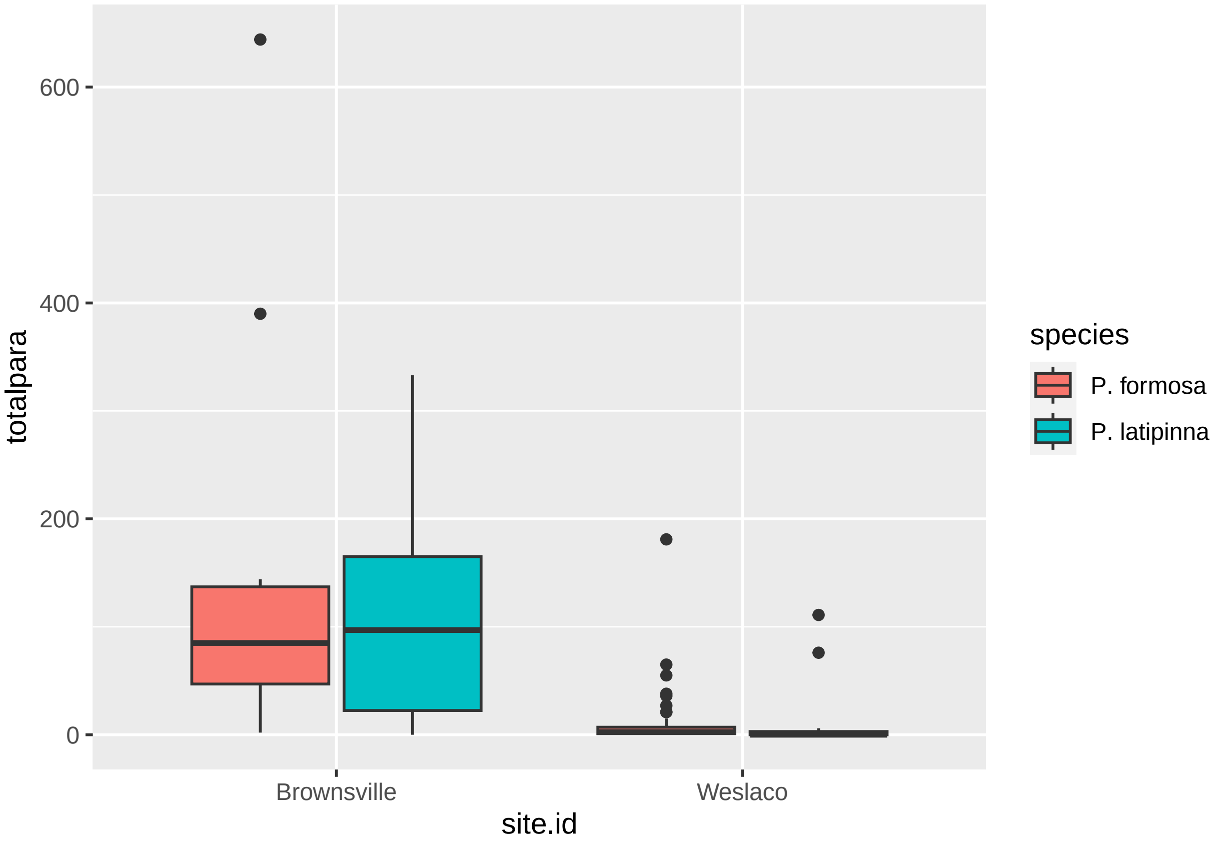
```
## Warning: Removed 14 rows containing non-finite values (`stat_bin()`).
```



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).

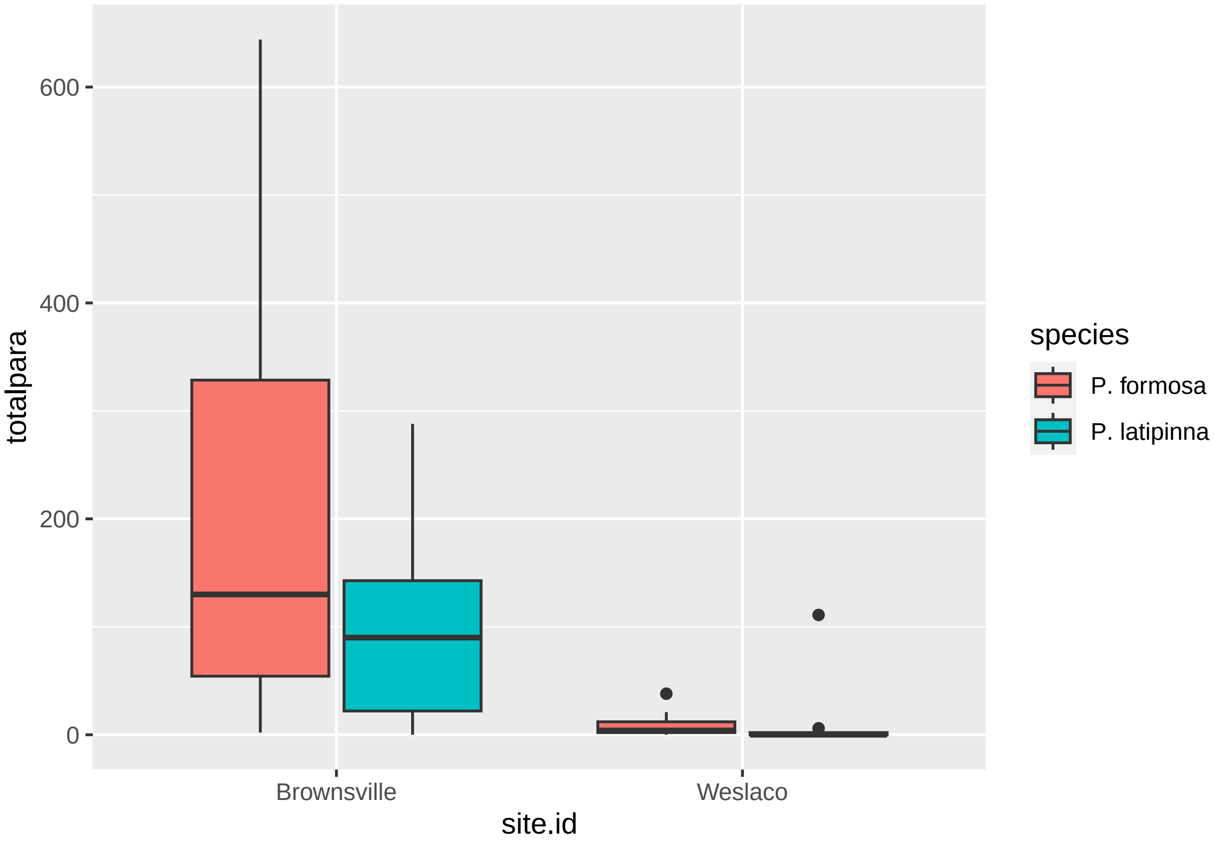
Let's look at the parasites by species and site.

```
# parasite data only
sp_parasite_box <- parasite_data %>%
  ggplot(mapping = aes(
    fill = species,
    x = site.id,
    y = totalpara
  )) +
  geom_boxplot()
sp_parasite_box
```



```
# behavior parasite data
# parasite data only
sp_parasite_beh_box <- all_data %>%
  ggplot(mapping = aes(
    fill = species,
    x = site.id,
    y = totalpara
  )) +
  geom_boxplot()
sp_parasite_beh_box
```

```
## Warning: Removed 14 rows containing non-finite values (`stat_boxplot()`).
```



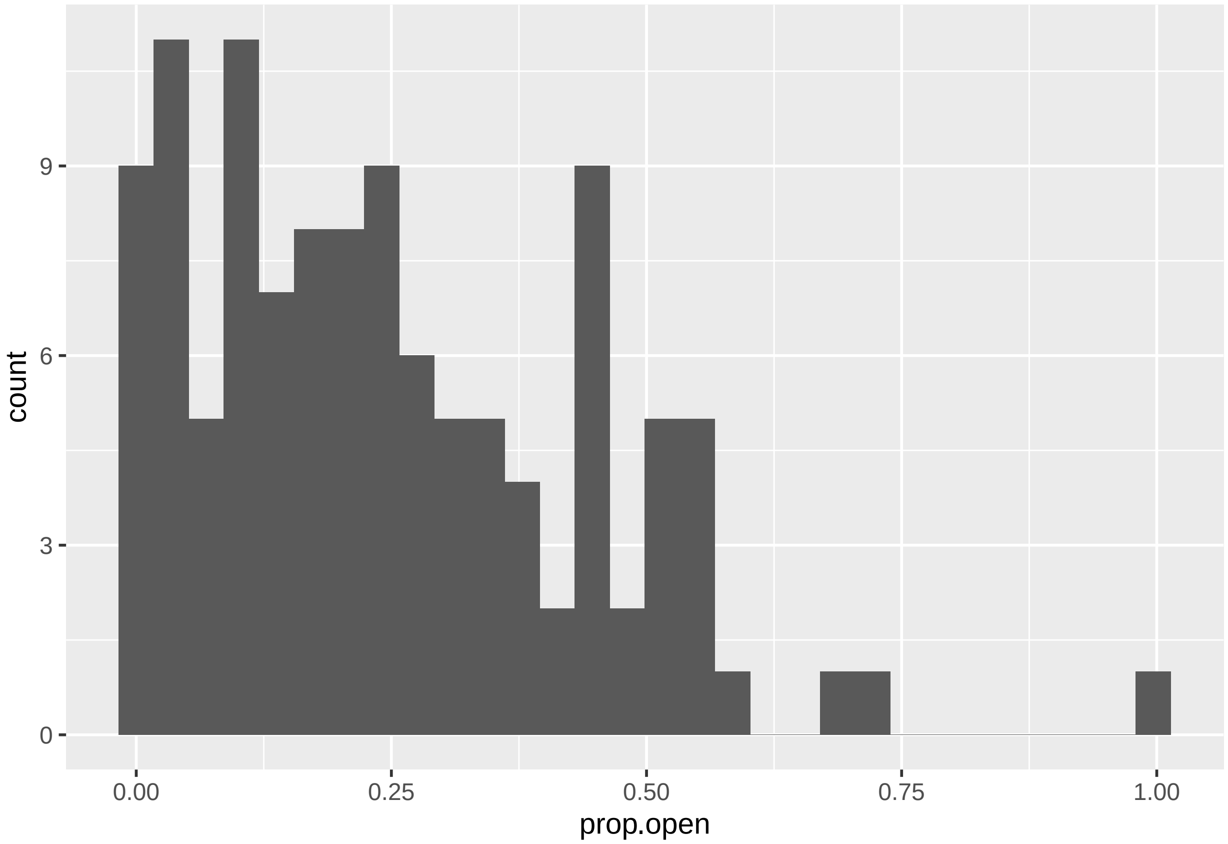
Ok, so there is a clear pattern of more parasites in Brownsville, generally. It also seems like there may be more parasites on Amazons in both sites, but we'll see what the stats say.

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

```
# boris_data, distributions
open_hist <- all_data %>%
  ggplot(mapping = aes(prop.open)) +
  geom_histogram()
open_hist
```

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

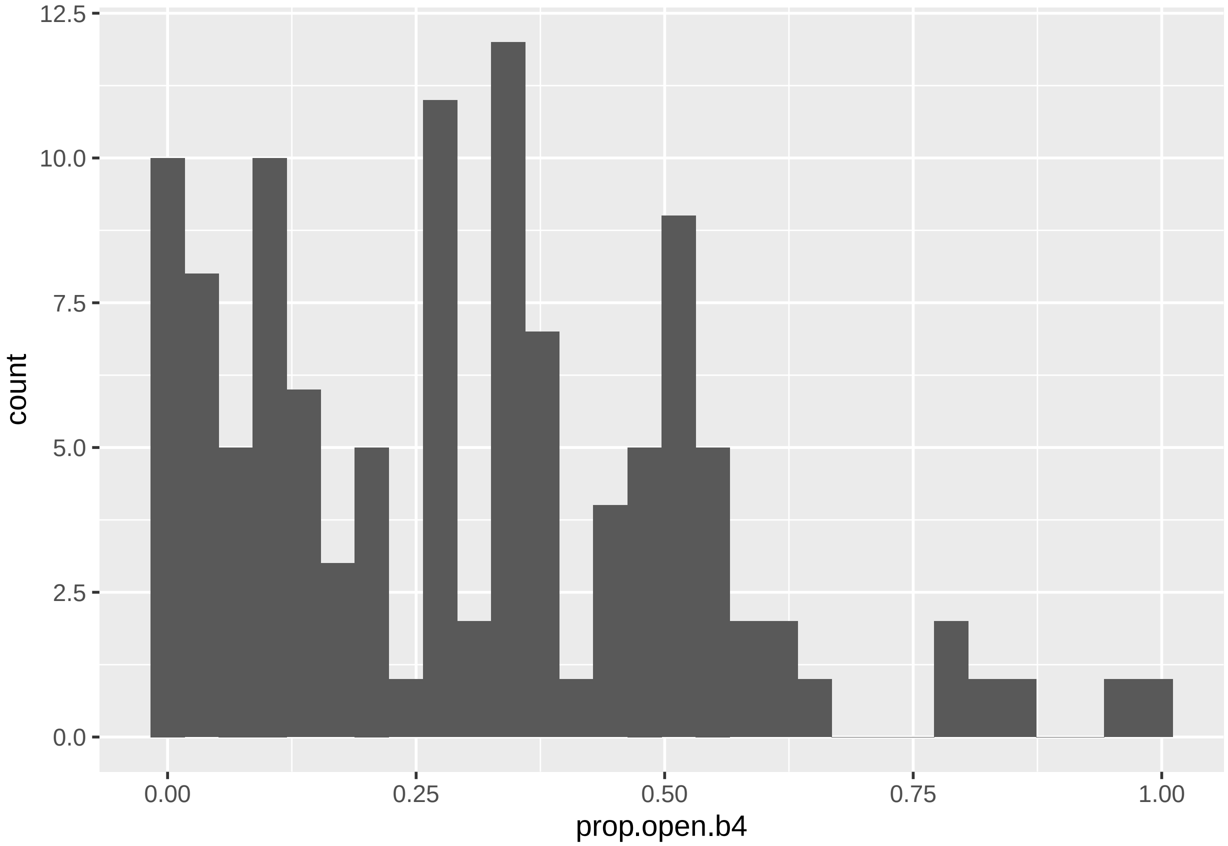
```
## Warning: Removed 26 rows containing non-finite values (`stat_bin()`).
```



```
# before startle
b4.open_hist <- all_data %>%
  ggplot(mapping = aes(prop.open.b4)) +
  geom_histogram()
b4.open_hist
```

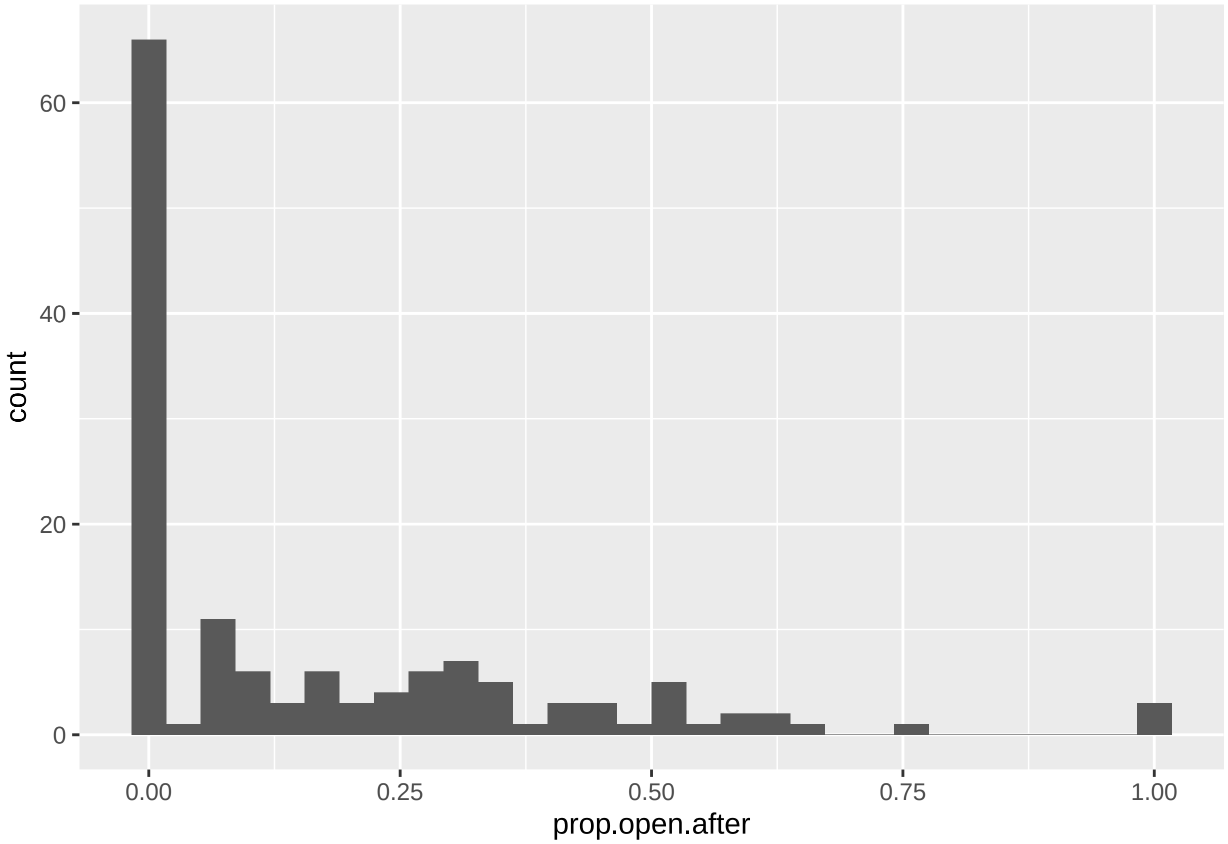
```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

```
## Warning: Removed 26 rows containing non-finite values (`stat_bin()`).
```



```
# after startle
after.open_hist <- all_data %>%
  ggplot(mapping = aes(prop.open.after)) +
  geom_histogram()
after.open_hist
```

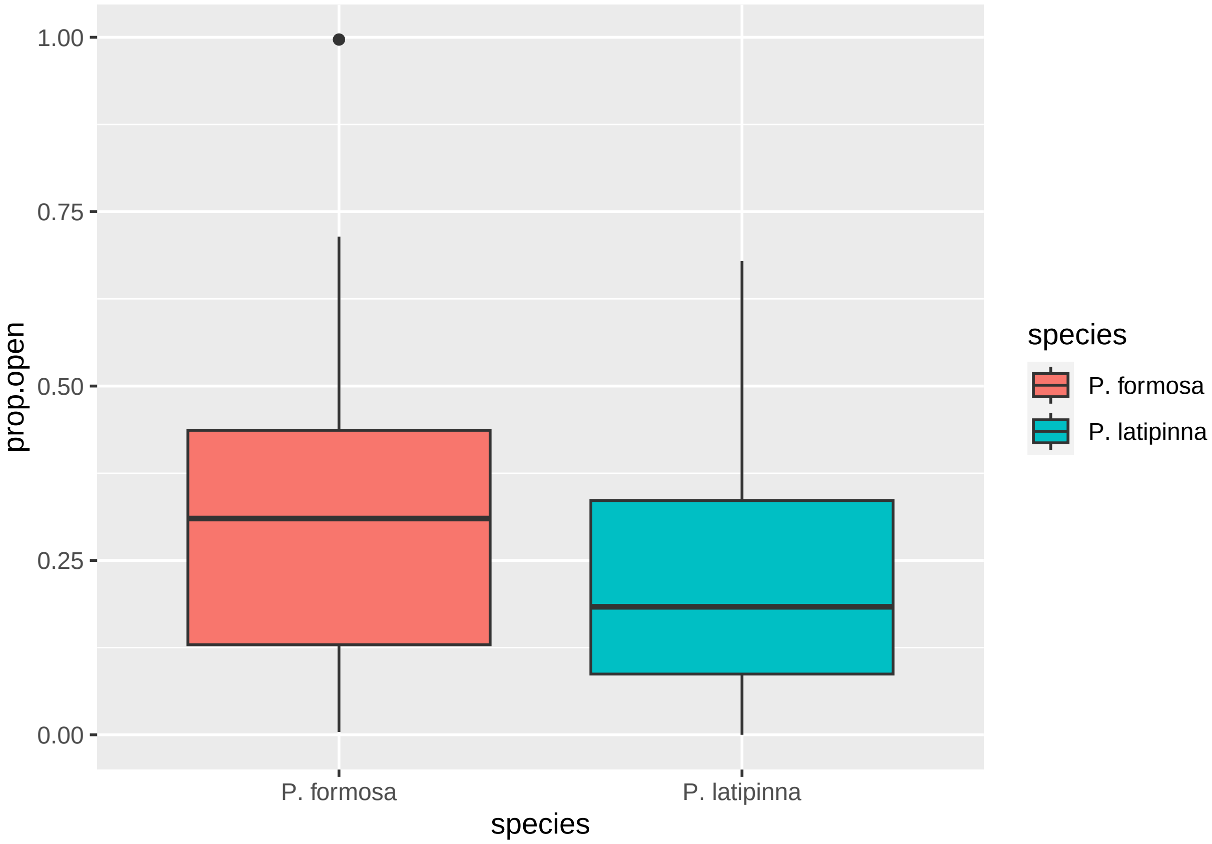
```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



Now, let's take a look at the proportion of time spent in the open by species.

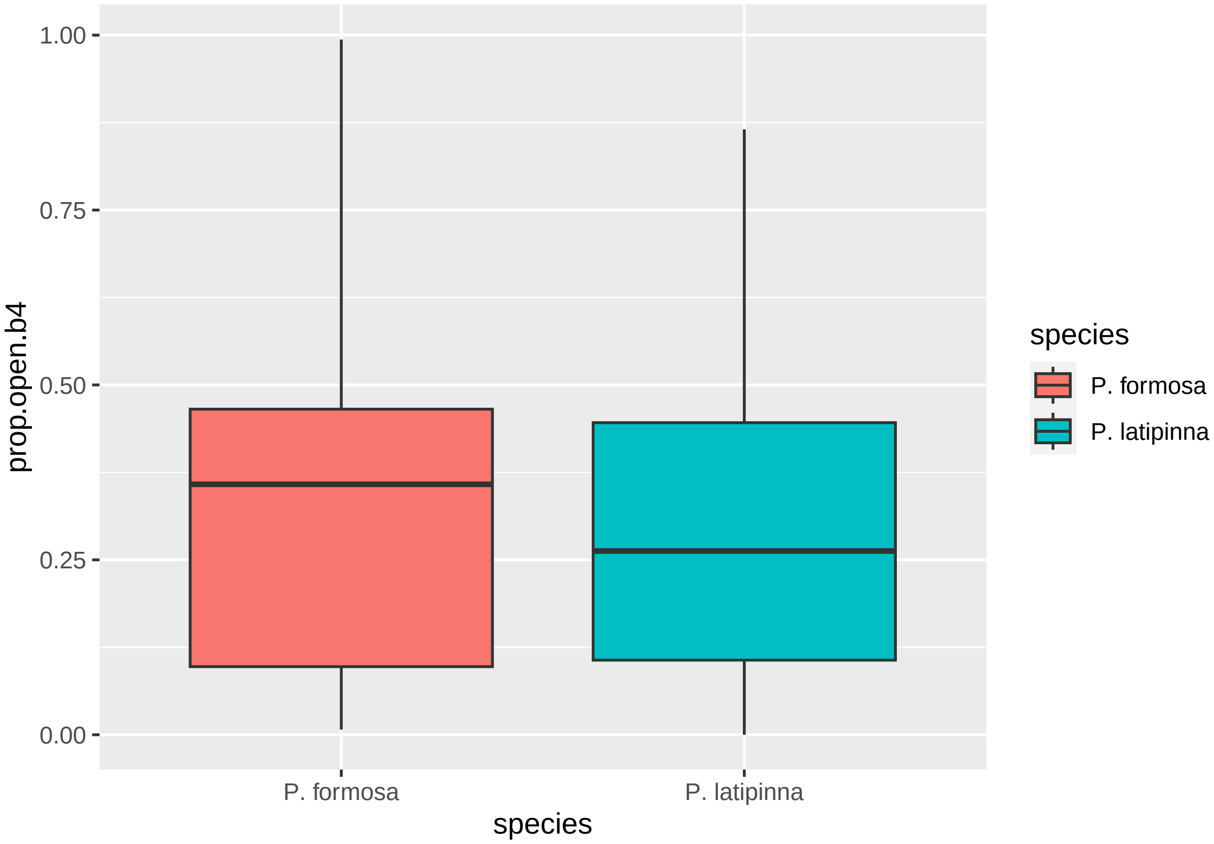
```r
# proportion time in open total
species_prop_total <- all_data %>%
  ggplot(mapping = aes(
    x = species,
    y = prop.open,
    fill = species
  )) +
  geom_boxplot()
species_prop_total
```

```
## Warning: Removed 26 rows containing non-finite values (`stat_boxplot()`).
```

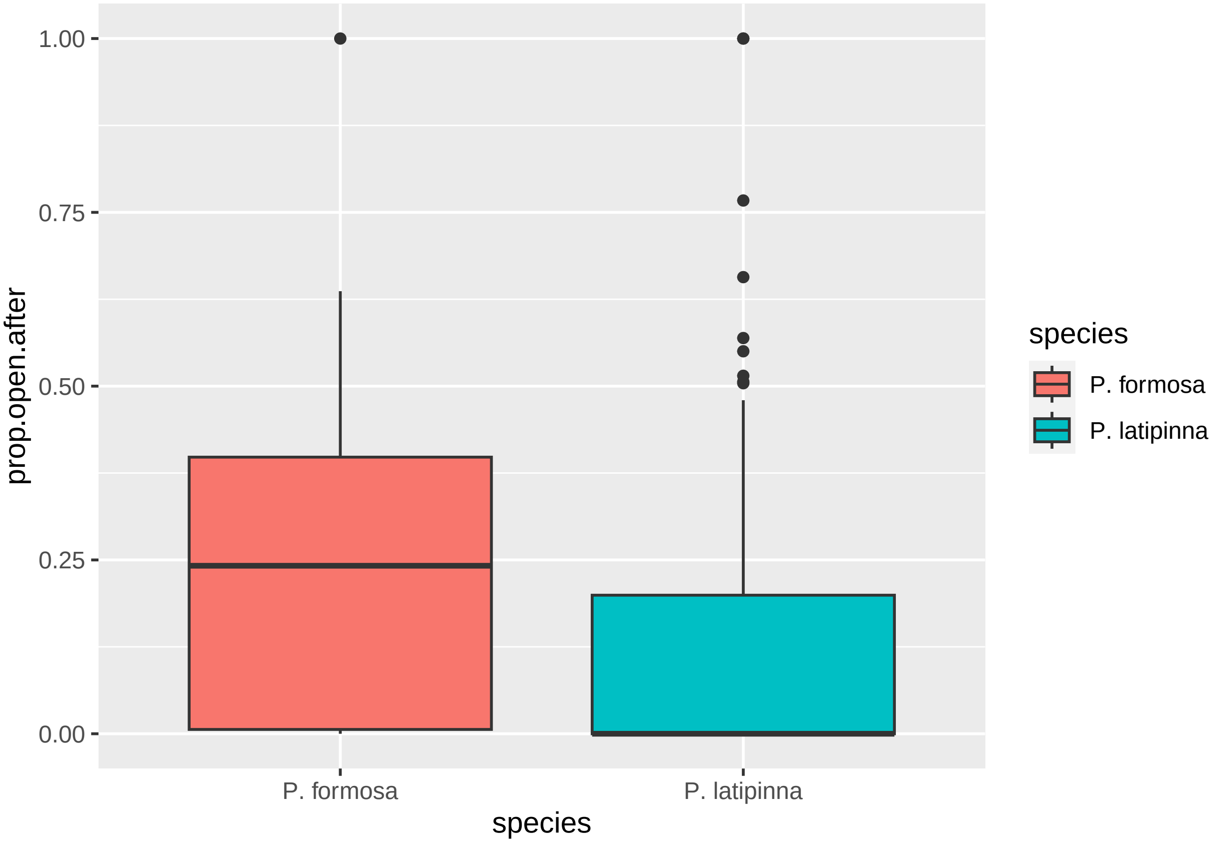


```
# proportion time in open before startle by species
species_prop_open <- all_data %>%
  ggplot(mapping = aes(
    x = species,
    y = prop.open.b4,
    fill = species
  )) +
  geom_boxplot()
species_prop_open
```

```
## Warning: Removed 26 rows containing non-finite values (`stat_boxplot()`).
```

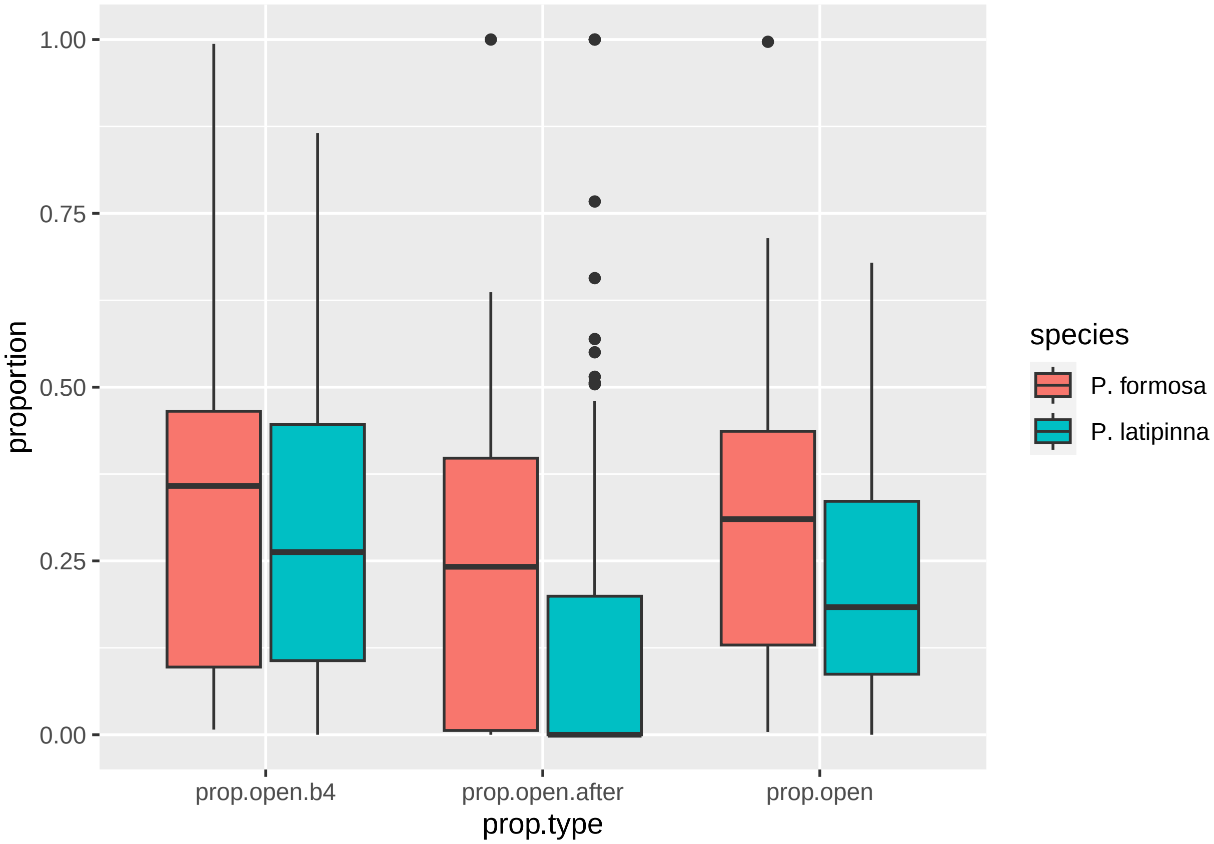


```
# proportion time in open after startle
species_prop_after <- all_data %>%
  ggplot(mapping = aes(
    x = species,
    y = prop.open.after,
    fill = species
  )) +
  geom_boxplot()
species_prop_after
```



Ok, now I want to plot before/after within species. Need to use the all_data_long.

```
# proportion time before/after by species
prop_time_total <- all_data_long %>%
  ggplot(mapping = aes(
    x = prop.type,
    y = proportion,
    fill = species
  )) +
  geom_boxplot()
prop_time_total
```

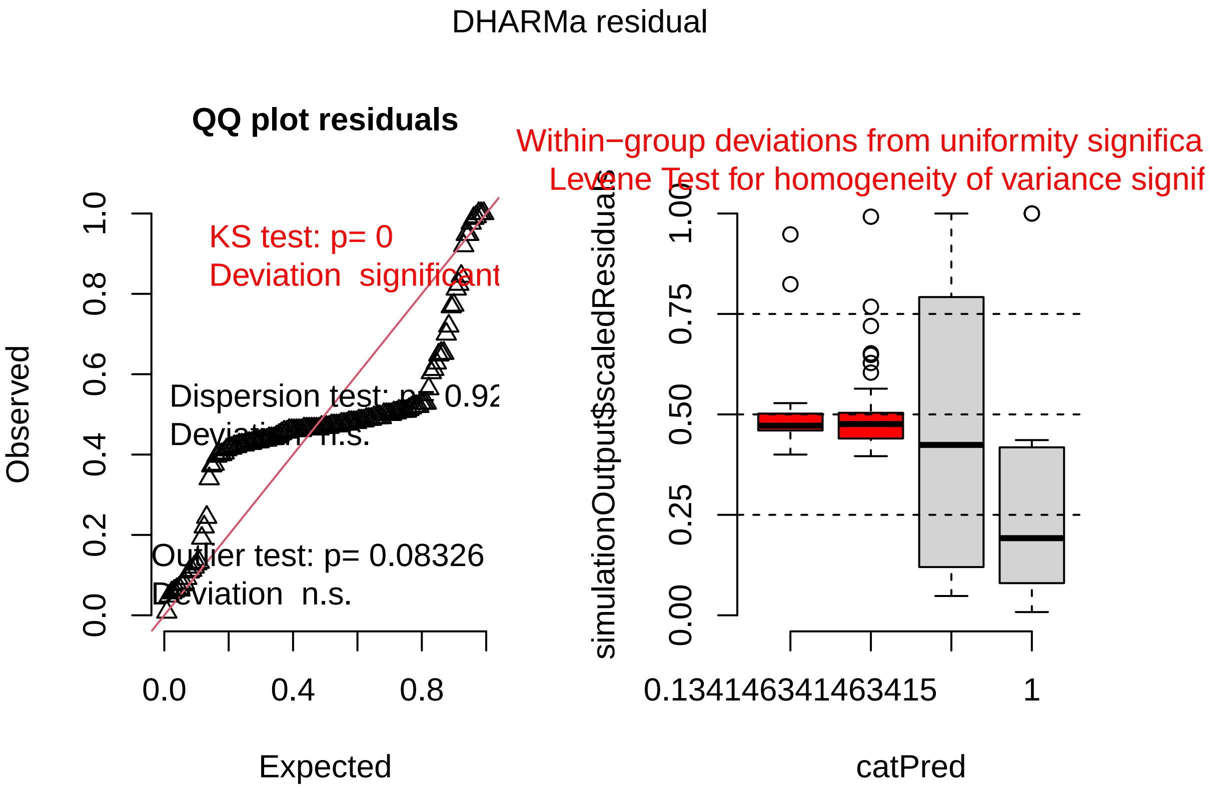


We'll see how the stats pan out, but it looks like Amazons might spend more time in the open than sailfins in general, and especially after the startle! ## Models ### Parasites

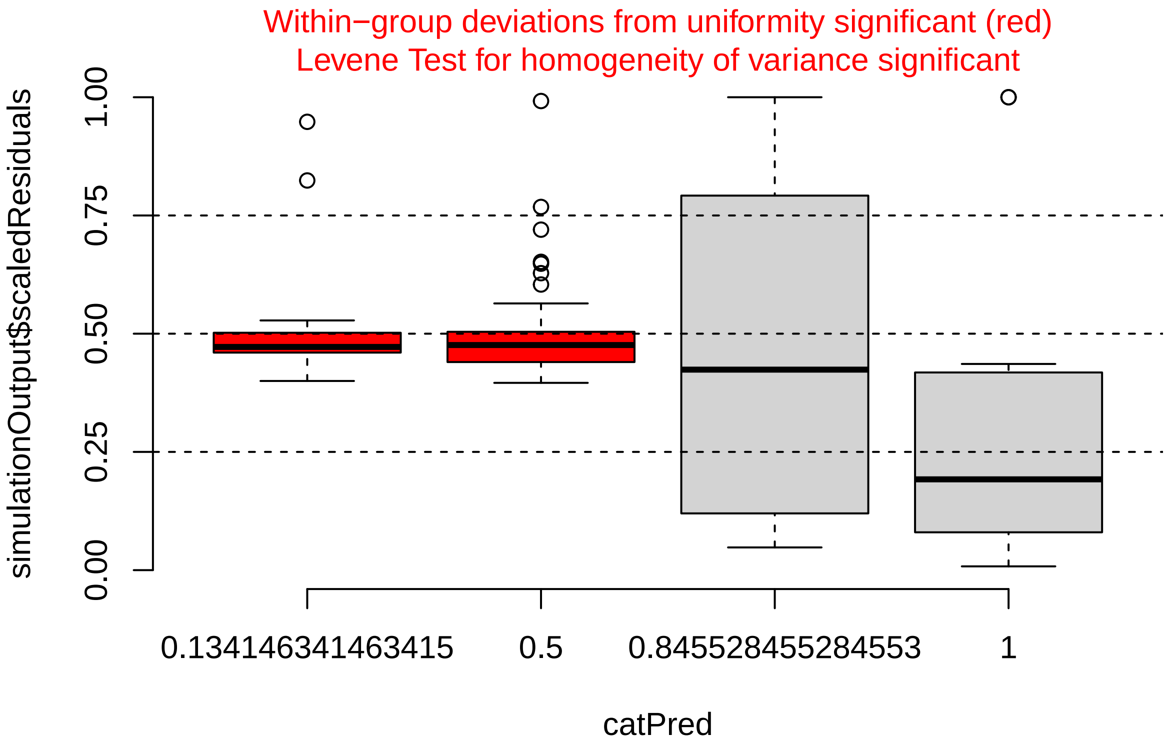
First, I want to see if there is a difference in parasite load between Amazons and Sailfins. This will be using the full dataset from Jessica, which includes fish that did not go through behavioral trials. We'll start with both sites, but I may just end up looking at the Weslaco site since that is a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

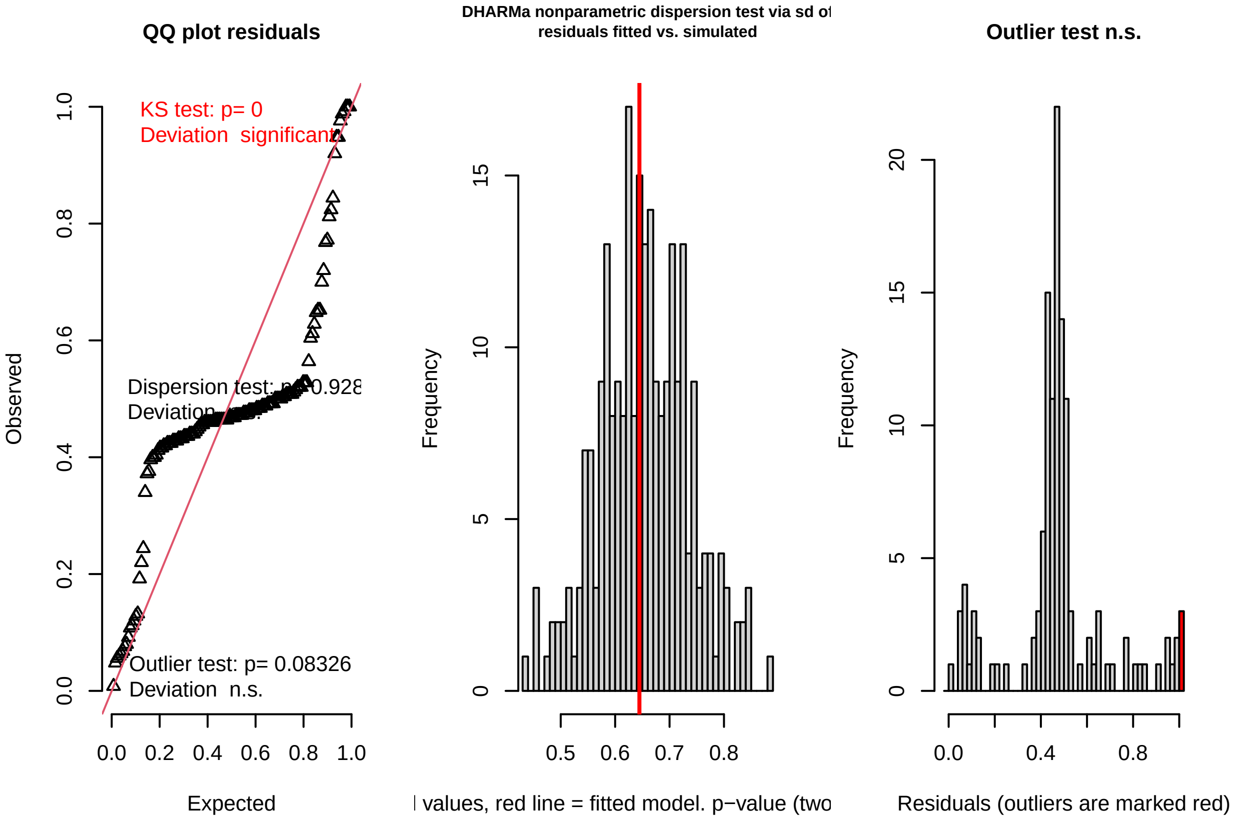
```
mod_full <- lm(totalpara ~ species * site.id,
  data = parasite_data
)

# 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
```



```r
plotResiduals(sim.mod_full)
```



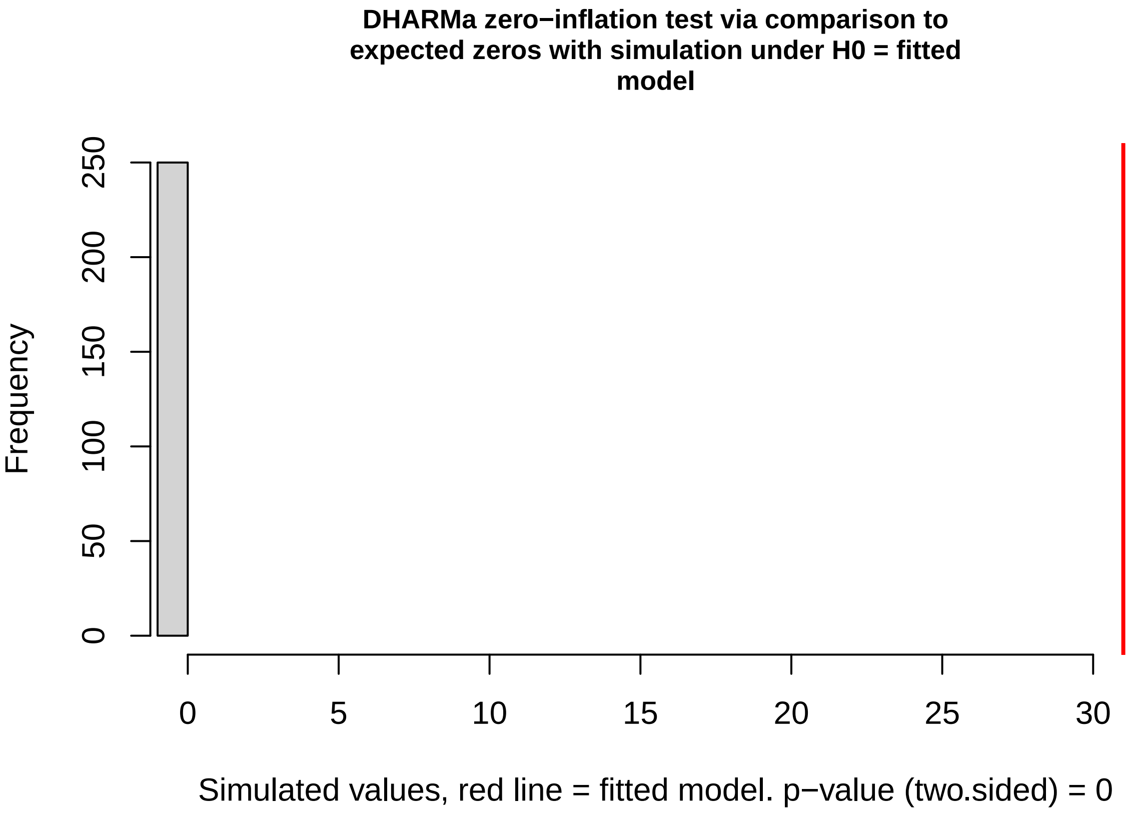
```r
# gives you a bunch of tests of dispersion etc
DHARMa::testResiduals(sim.mod_full) # dispersion looks ok, but the QQ plot is bad
```



```
## $uniformity
##
##  Asymptotic one-sample Kolmogorov-Smirnov test
##
## data:  simulationOutput$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_full_poisson <- glmmTMB(totalpara ~ species * site.id,
  family = "poisson",
  ziformula = ~., # the ~. argument specifies the formula to match the model oarameters (i.e. species*s
  data = parasite_data
)

mod_full_nbin <- glmmTMB(totalpara ~ species * site.id,
  family = "nbinom2", # I chose nbinom2 because nbinom1 fails to converge and it is the "classic parame
  ziformula = ~.,
  data = parasite_data
)
```

```r
#
check_overdispersion(mod_full_poisson) # data is over dispersed
```

```
## # Overdispersion test
##
##  dispersion ratio =  19.355
##           p-value = < 0.001
```

```
## Overdispersion detected.
```

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

```
## # Overdispersion test
##
##  dispersion ratio = 0.530
##           p-value = 0.336
```

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

```r
lrtest(mod_full_poisson, mod_full_nbin) # there is a sig difference between the two
```

```
## Likelihood ratio test
##
## Model 1: totalpara ~ species * site.id
## Model 2: totalpara ~ species * site.id
##   #Df   LogLik Df  Chisq Pr(>Chisq)
## 1   8 -3089.09
## 2   9  -474.74  1 5228.7  < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
# checking AIC
anova(mod_full_poisson, mod_full_nbin) # nbin model is much lower
```

```
## Data: parasite_data
## Models:
## mod_full_poisson: totalpara ~ species * site.id, zi=~., disp=~1
## mod_full_nbin: totalpara ~ species * site.id, zi=~., disp=~1
##                   Df    AIC    BIC   logLik deviance  Chisq Chi Df Pr(>Chisq)
## mod_full_poisson  8 6194.2 6217.0 -3089.09   6178.2
## mod_full_nbin     9  967.5  993.2  -474.74    949.5 5228.7      1  < 2.2e-16
##
## mod_full_poisson
## mod_full_nbin    ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Now, let's do some backwards model selection.

```r
# interaction model
mod_full_nbin <- glmmTMB(totalpara ~ species * site.id,
  family = "nbinom2",
  ziformula = ~.,
  data = parasite_data
)

# combined model
mod_combined_nbin <- glmmTMB(totalpara ~ species + site.id,
  family = "nbinom2",
  ziformula = ~.,
  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 combined
```

```
## Likelihood ratio test
##
## Model 1: totalpara ~ species * site.id
## Model 2: totalpara ~ species + site.id
##   #Df  LogLik Df  Chisq Pr(>Chisq)
## 1   9 -474.74
## 2   7 -474.97 -2 0.4536     0.7971
```

```
# site model
mod_site_nbin <- glmmTMB(totalpara ~ site.id,
  family = "nbinom2",
  ziformula = ~.,
  data = parasite_data
)

# test species effect
lrtest(mod_site_nbin, mod_combined_nbin) # There is a difference between the combined model and the sit
```

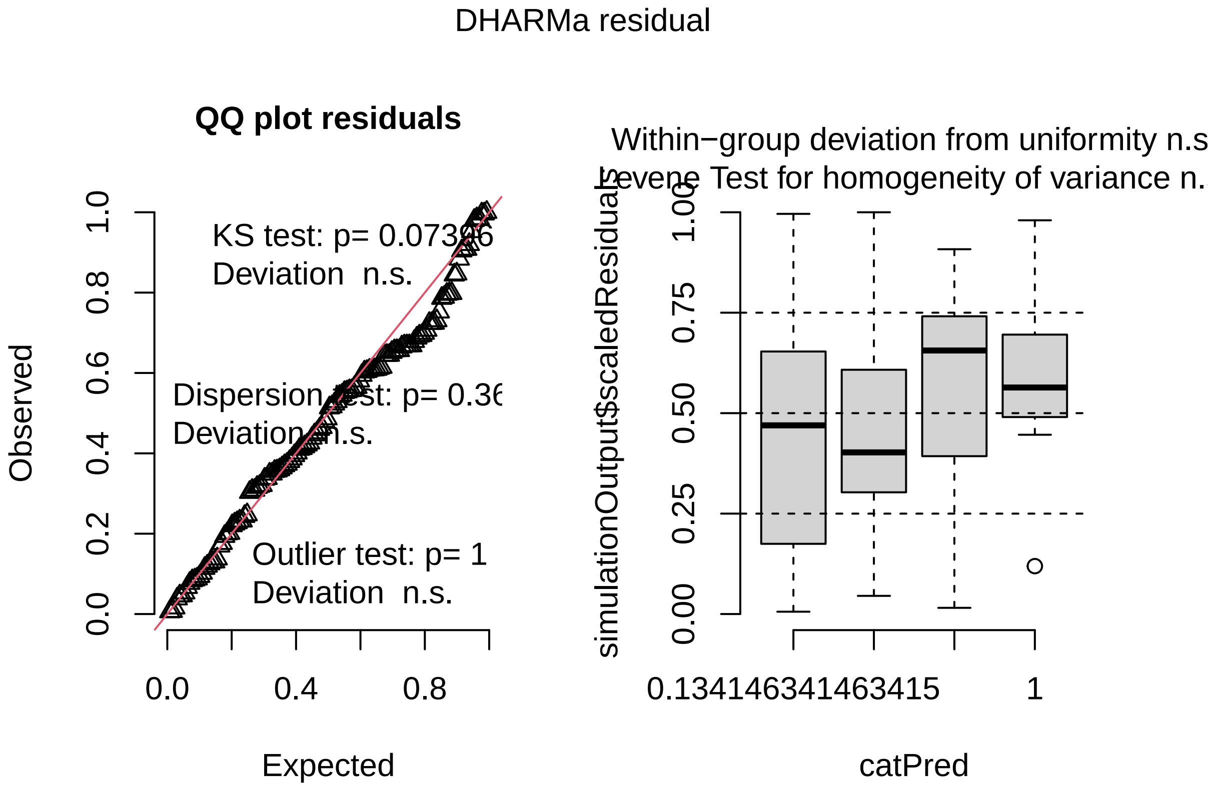
```
## Likelihood ratio test
##
## Model 1: totalpara ~ site.id
## Model 2: totalpara ~ species + site.id
##   #Df  LogLik Df  Chisq Pr(>Chisq)
## 1   5 -479.64
## 2   7 -474.97  2 9.3326   0.009407 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
# species model
mod_species_nbin <- glmmTMB(totalpara ~ species,
  family = "nbinom2",
  ziformula = ~.,
  data = parasite_data
)

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

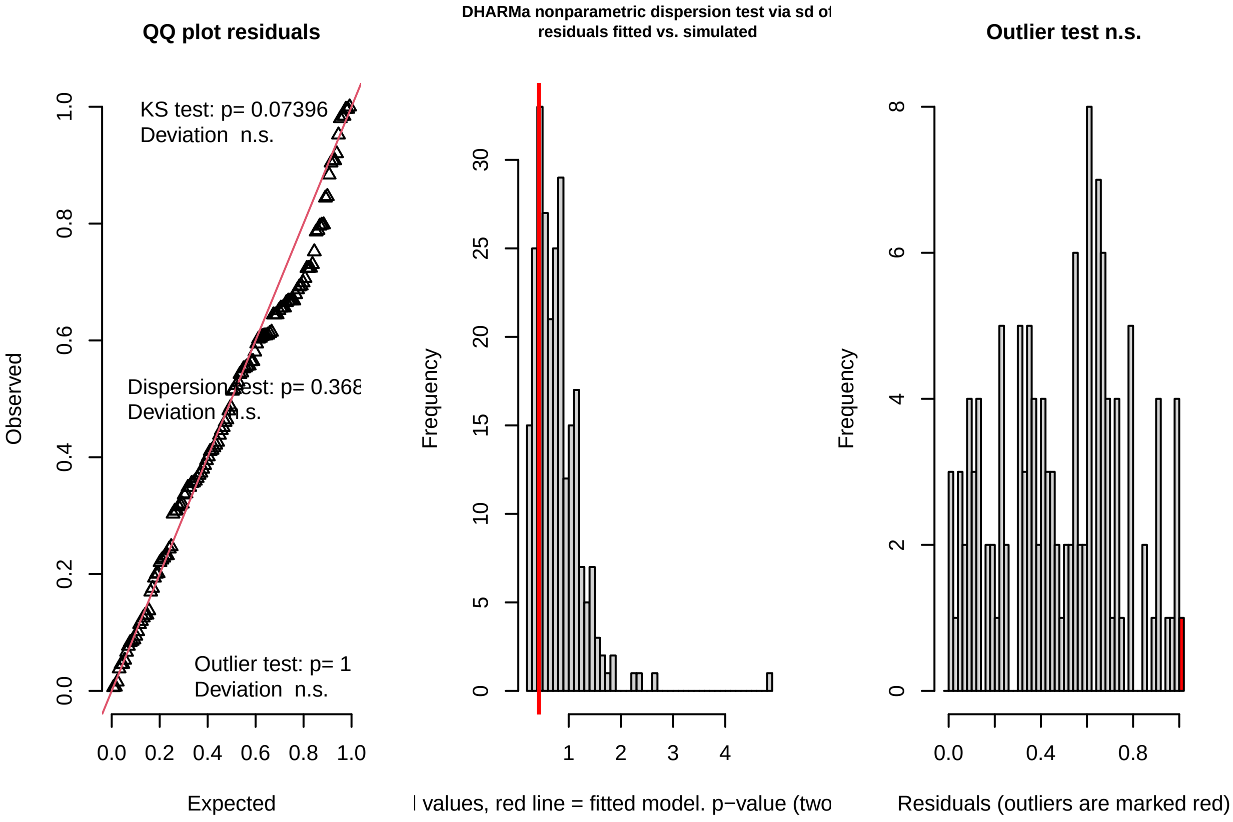
```
## Likelihood ratio test
##
## Model 1: totalpara ~ species
## Model 2: totalpara ~ species + site.id
##   #Df  LogLik Df  Chisq Pr(>Chisq)
## 1   5 -507.20
## 2   7 -474.97  2 64.459  1.007e-14 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Ok let's more forward with the combined model. Let's check those assumptions.

```
# checking assumptions for the combined model
# First we have to simulate your residuals
sim.output <- DHARMa::simulateResiduals(mod_combined_nbin)

# then you can plot them
plot(sim.output) # QQ plot not perfect, but otherwise things look ok
```



```
# gives you a bunch of tests of dispersion etc
DHARMa::testResiduals(sim.output) # QQ not great, but outlier and dispersion ok. Are glmms robust to no
```



```
## $uniformity
##
##  Asymptotic one-sample Kolmogorov-Smirnov test
##
## data:  simulationOutput$scaledResiduals
## D = 0.11349, p-value = 0.07396
## alternative hypothesis: two-sided
##
##
## $dispersion
##
##  DHARMa nonparametric dispersion test via sd of residuals fitted vs.
##  simulated
##
## data:  simulationOutput
## dispersion = 0.54807, p-value = 0.368
## 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: 0.00671875 )
##                                   0.0078125
```

```
# from poking around, it seems like estimating overdispersion is how we evaluate goodness of fit (versu
check_overdispersion(mod_combined_nbin) # no overdispersion detected. dispersion ratio = 1.418, p = 0.2
```

```
## # Overdispersion test
##
##  dispersion ratio = 0.548
##           p-value = 0.368
```

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

Let's take a look at our final model.

```
summary(mod_combined_nbin) # species insignificant (est = -0.06, SE = 0.32, p = 0.85) and site highly s
```

```
##  Family: nbinom2  ( log )
## Formula:          totalpara ~ species + site.id
## Zero inflation:             ~.
## Data: parasite_data
##
##      AIC      BIC   logLik deviance df.resid
##    963.9    983.9   -475.0    949.9      121
##
##
## Dispersion parameter for nbinom2 family (): 0.453
##
## Conditional model:
##                     Estimate Std. Error z value Pr(>|z|)
## (Intercept)          4.84284    0.31717  15.269  < 2e-16 ***
## speciesP. latipinna -0.05957    0.31565  -0.189     0.85
## site.idWeslaco      -2.46169    0.32390  -7.600 2.96e-14 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Zero-inflation model:
##                     Estimate Std. Error z value Pr(>|z|)
## (Intercept)           -37.42   10929.76  -0.003    0.997
## speciesP. latipinna    19.80    9848.17   0.002    0.998
## site.idWeslaco         17.15    4740.59   0.004    0.997
```

Ok now' lets dig into those effect sizes

```
emm_para <- emmeans::emmeans(mod_combined_nbin, specs = pairwise ~ species:site.id)

emm_para$emmeans
```

```
##  species      site.id     emmean    SE  df asymp.LCL asymp.UCL
##  P. formosa   Brownsville   4.84 0.317 Inf      4.22      5.46
##  P. latipinna Brownsville   4.78 0.265 Inf      4.26      5.30
##  P. formosa   Weslaco       2.38 0.193 Inf      2.00      2.76
##  P. latipinna Weslaco       2.32 0.294 Inf      1.75      2.90
##
## Results are given on the log (not the response) scale.
## Confidence level used: 0.95
```

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 = "nbinom2",
  ziformula = ~.,
  data = parasite_data_wes
)

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

```
##  Family: nbinom2  ( log )
## Formula:          totalpara ~ species
## Zero inflation:             ~.
## Data: parasite_data_wes
##
##      AIC      BIC   logLik deviance df.resid
##    514.5    527.0   -252.3    504.5       85
##
##
## Dispersion parameter for nbinom2 family (): 0.35
##
## Conditional model:
##                      Estimate Std. Error z value Pr(>|z|)
## (Intercept)           2.33981    0.22557  10.373   <2e-16 ***
## speciesP. latipinna  0.01089    0.46685   0.023    0.981
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Zero-inflation model:
##                      Estimate Std. Error z value Pr(>|z|)
## (Intercept)            -20.06    5322.58  -0.004    0.997
## speciesP. latipinna     19.35    5322.58   0.004    0.997
```

```
# 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 = "nbinom2",
  ziformula = ~.,
  data = parasite_data_wes
)

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   5 -252.26
## 2   3 -254.77 -2 5.0265      0.081 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

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

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

# filter data to just Brownsville
parasite_data_br <- parasite_data %>%
  filter(site.id == "Brownsville")

# species model
mod_para_species_br <- glmmTMB(totalpara ~ species,
  family = "nbinom2",
  ziformula = ~.,
  data = parasite_data_br
)

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

```
##  Family: nbinom2  ( log )
## Formula:          totalpara ~ species
## Zero inflation:              ~.
## Data: parasite_data_br
##
##      AIC      BIC   logLik deviance df.resid
##    445.3    453.5   -217.7    435.3       33
##
##
## Dispersion parameter for nbinom2 family (): 0.942
##
## Conditional model:
##                     Estimate Std. Error z value Pr(>|z|)
## (Intercept)           5.0298     0.3116  16.144   <2e-16 ***
## speciesP. latipinna  -0.2677     0.3740  -0.716    0.474
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Zero-inflation model:
##                     Estimate Std. Error z value Pr(>|z|)
## (Intercept)           -19.84    6135.77  -0.003    0.997
## speciesP. latipinna    17.16    6135.77   0.003    0.998
```

```
# just to be extra sure, I also ran an lrt on the brownsville species model versus a null model set to
mod_para_null_br <- glmmTMB(totalpara ~ 1,
  family = "nbinom2",
  ziformula = ~.,
  data = parasite_data_br
)

lrtest(mod_para_species_br, mod_para_null_br) # no significance, suggesting that the species effect is
```

```
## Likelihood ratio test
##
```

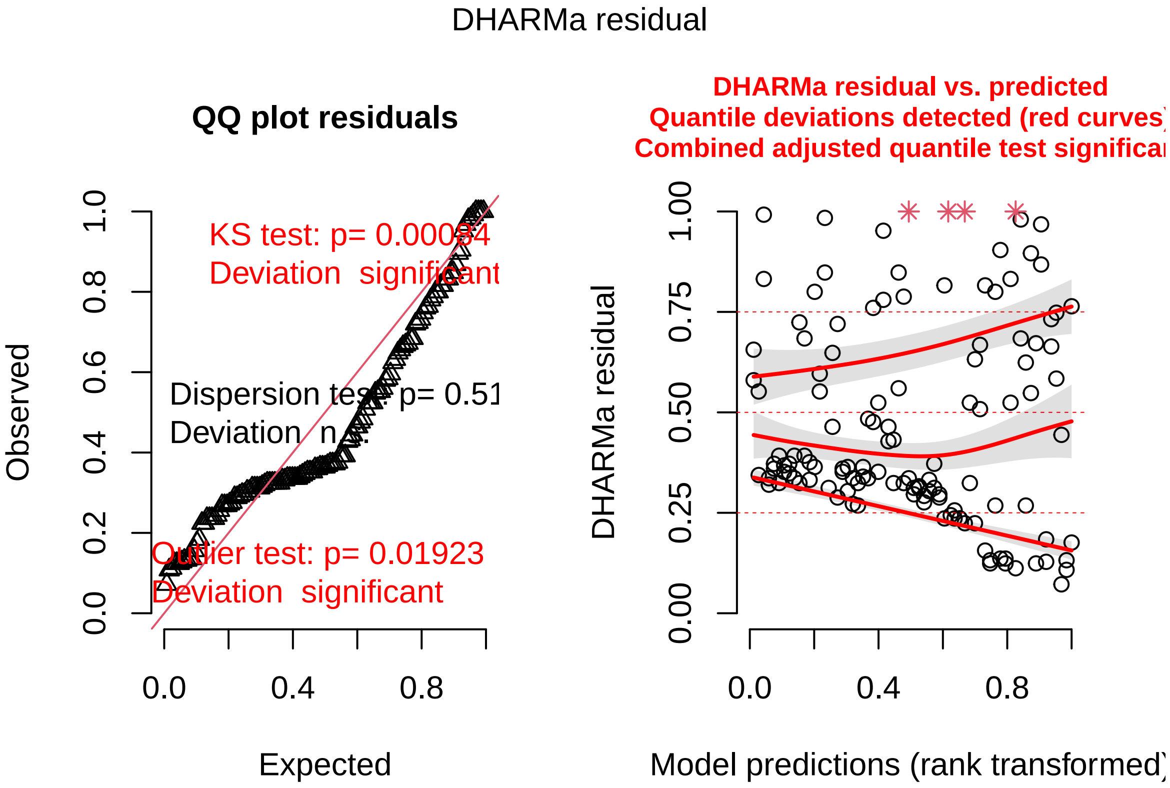
```
## Model 1: totalpara ~ species
## Model 2: totalpara ~ 1
##   #Df  LogLik Df  Chisq Pr(>Chisq)
## 1   5 -217.67
## 2   3 -218.53 -2 1.7064     0.4261
```

Behavior

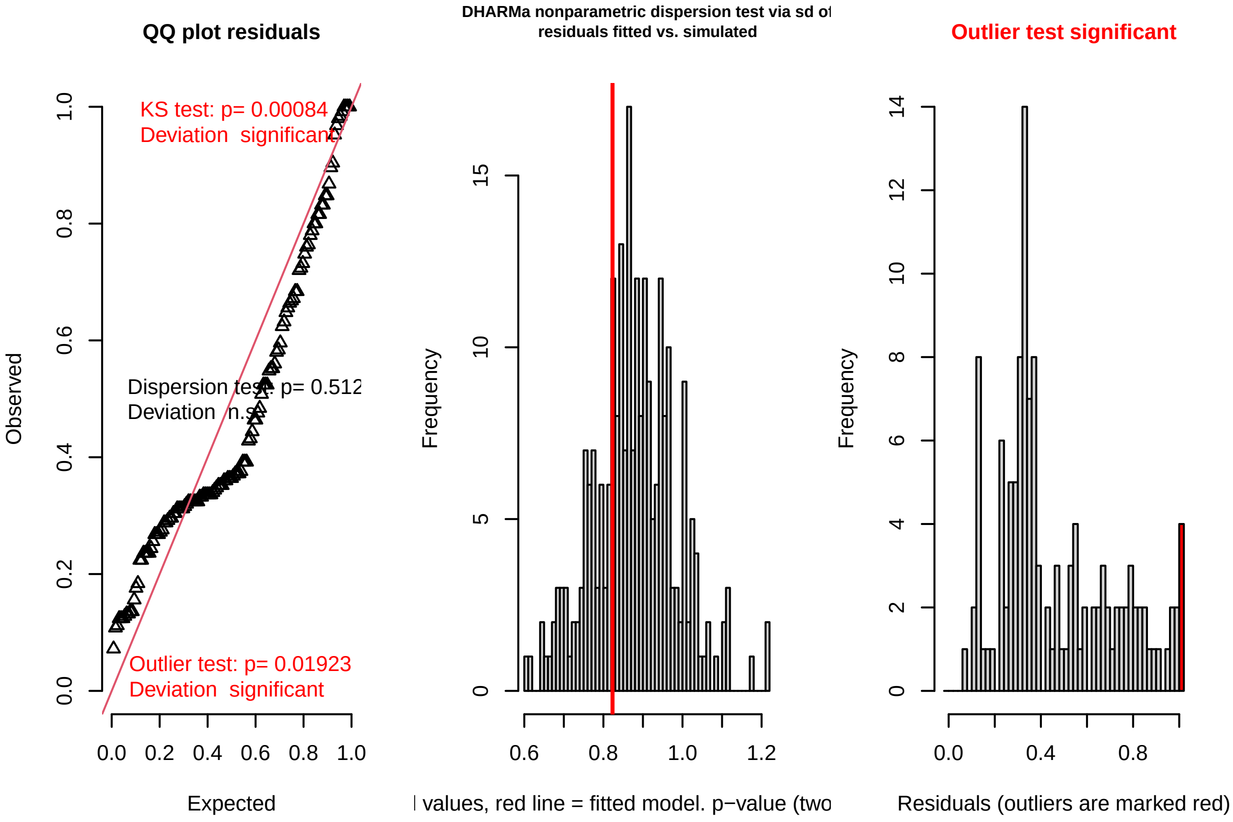
Ok now we'd need to see if behavior overall, before the stimulus, and after the stimulus differs between species and parasite load. We may have to just look at one site, Weslaco since the data is more balanced between species there.

Let's look at the time spent in the open after the startle stimulus, as that is the most relevant behavioral metric (since parasites can affect anti-pred behavior, like swimming in the open)

```
# FULL model, with three-way interaction
mod_beh_full <- lm(total.time.open.after ~ species * site.id * totalpara + standard.length,
                   data = all_data)

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

# then you can plot them
plot(sim.mod_beh) # these both look not great
```



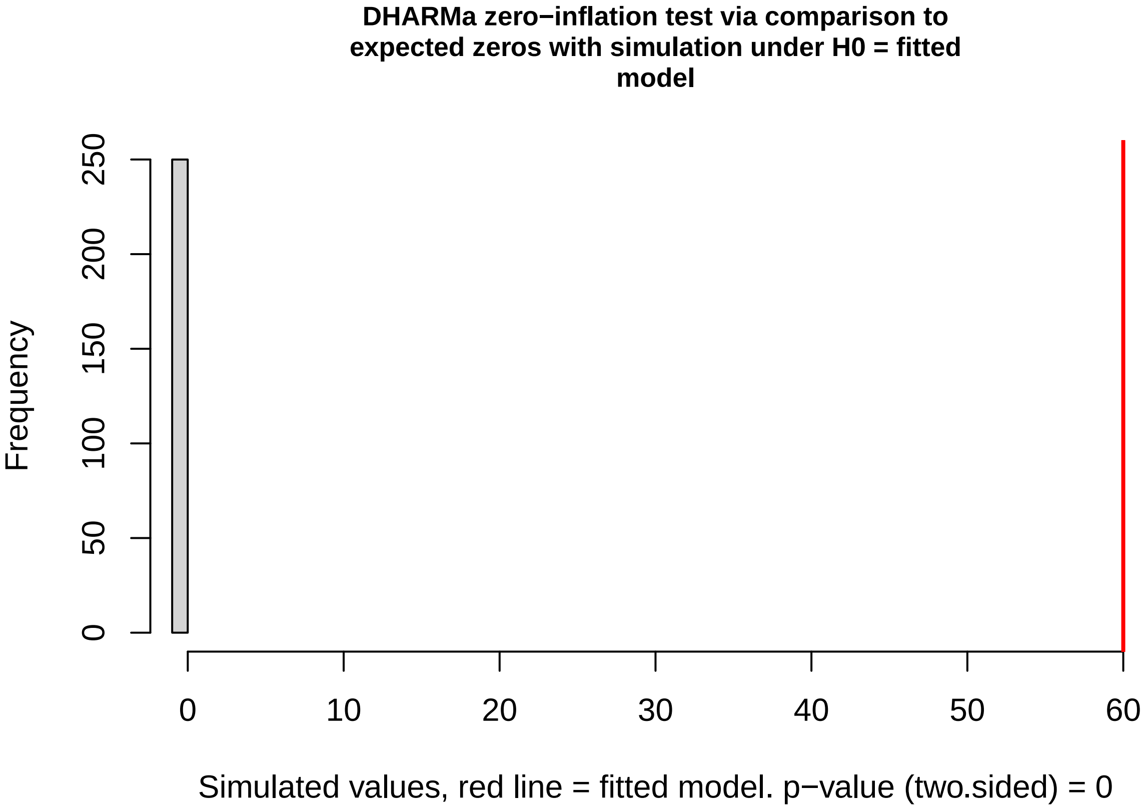
```
# gives you a bunch of tests of dispersion etc
DHARMa::testResiduals(sim.mod_beh) # dispersion looks ok, but the QQ plot is bad and outlier test is si
```



```
## $uniformity
##
##  Asymptotic one-sample Kolmogorov-Smirnov test
##
## data:  simulationOutput$scaledResiduals
## D = 0.17493, p-value = 0.0008424
## alternative hypothesis: two-sided
##
##
## $dispersion
##
##  DHARMa nonparametric dispersion test via sd of residuals fitted vs.
##  simulated
##
## data:  simulationOutput
## dispersion = 0.93602, p-value = 0.512
## alternative hypothesis: two.sided
##
##
## $outliers
##
##  DHARMa outlier test based on exact binomial test with approximate
##  expectations
##
## data:  simulationOutput
## outliers at both margin(s) = 4, observations = 127, p-value = 0.01923
## alternative hypothesis: true probability of success is not equal to 0.007968127
## 95 percent confidence interval:
```

```
##  0.008647004 0.078679127
## sample estimates:
## frequency of outliers (expected: 0.00796812749003984 )
##                                            0.03149606
```

```
# yes we can see that the data is super zero inflated
DHARMa::testZeroInflation(sim.mod_beh) # 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 now let's try to fit some different models.

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

```
## 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
```

```r
mod_beh_nbin <- glmmTMB(total.time.open.after ~ species * site.id * totalpara + standard.length,
  family = "nbinom2",
  ziformula = ~.,
  data = all_data
)

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

```
## # Overdispersion test
##
##  dispersion ratio =   2.067
##           p-value = < 0.001

## Overdispersion detected.
```

```r
check_overdispersion(mod_beh_nbin) # data is not over dispersed
```

```
## # Overdispersion test
##
##  dispersion ratio = 0.795
##           p-value = 0.616

## No overdispersion detected.
```

```r
lrtest(mod_beh_poisson, mod_beh_nbin) # there is a sig difference between the two
```

```
## Likelihood ratio test
##
## Model 1: total.time.open.after ~ species * site.id * totalpara + standard.length
## Model 2: total.time.open.after ~ species * site.id * totalpara + standard.length
##   #Df   LogLik Df  Chisq Pr(>Chisq)
## 1  18 -2990.01
## 2  19  -477.45  1 5025.1  < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
# checking AIC
anova(mod_beh_poisson, mod_beh_nbin) # nbin model is much lower
```

```
## Data: all_data
## Models:
## mod_beh_poisson: total.time.open.after ~ species * site.id * totalpara + standard.length, zi=~., disp
## mod_beh_nbin: total.time.open.after ~ species * site.id * totalpara + standard.length, zi=~., disp=~
##                 Df    AIC    BIC   logLik deviance  Chisq Chi Df Pr(>Chisq)
## mod_beh_poisson 18 6016.0 6067.2 -2990.01   5980.0
## mod_beh_nbin    19  992.9 1046.9  -477.45    954.9 5025.1      1  < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Ok, let's move forward with the nbin model. Time for some backwards model selection.

```r
# interaction model
mod_beh_nbin <- glmmTMB(total.time.open.after ~ species * site.id * totalpara + standard.length,
  family = "nbinom2",
  ziformula = ~.,
  data = all_data
)
```

```
# removing species interaction
mod_spcomb_nbin <- glmmTMB(total.time.open.after ~ species + site.id * totalpara + standard.length,
  family = "nbinom2",
  ziformula = ~.,
  data = all_data
)

# test 2-way with log likelihood ratio test
lrtest(mod_beh_nbin, mod_spcomb_nbin) # no difference, so let's try removing all interactions
```

```
## Likelihood ratio test
##
## Model 1: total.time.open.after ~ species * site.id * totalpara + standard.length
## Model 2: total.time.open.after ~ species + site.id * totalpara + standard.length
##   #Df  LogLik Df Chisq Pr(>Chisq)
## 1  19 -477.45
## 2  13 -482.99 -6 11.08    0.08593 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
# combined model
mod_beh_combined_nbin <- glmmTMB(total.time.open.after ~ species + site.id + totalpara + standard.lengt
  family = "nbinom2",
  ziformula = ~.,
  data = all_data
)

# test species effect
lrtest(mod_beh_nbin, mod_beh_combined_nbin) # No difference, so let's see how the combined does compare
```

```
## Likelihood ratio test
##
## Model 1: total.time.open.after ~ species * site.id * totalpara + standard.length
## Model 2: total.time.open.after ~ species + site.id + totalpara + standard.length
##   #Df  LogLik Df  Chisq Pr(>Chisq)
## 1  19 -477.45
## 2  11 -484.04 -8 13.171     0.1061
```

```
# no length
mod_beh_nolength_nbin <- glmmTMB(total.time.open.after ~ species + site.id + totalpara,
  family = "nbinom2",
  ziformula = ~.,
  data = all_data
)

# test length effect
lrtest(mod_beh_nolength_nbin, mod_beh_combined_nbin) # no sig diff, let's drop length
```

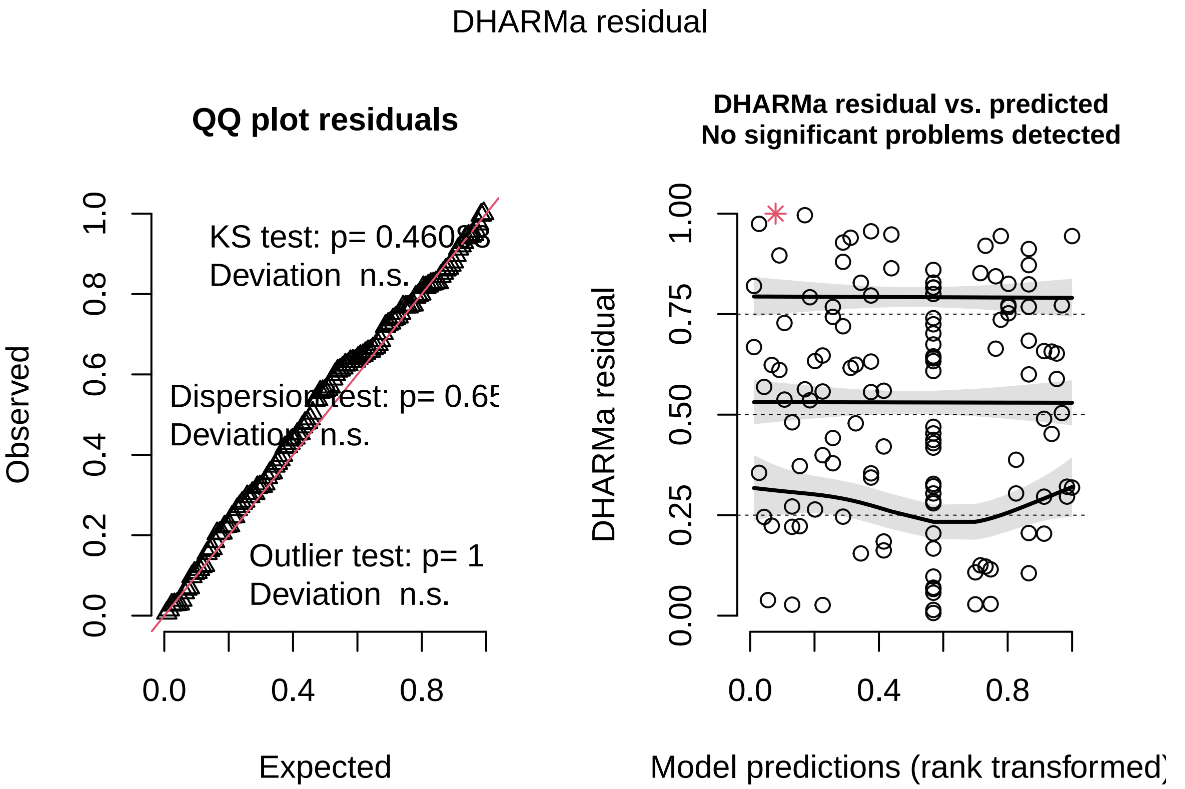
```
## Likelihood ratio test
##
## Model 1: total.time.open.after ~ species + site.id + totalpara
## Model 2: total.time.open.after ~ species + site.id + totalpara + standard.length
##   #Df  LogLik Df  Chisq Pr(>Chisq)
## 1   9 -486.68
## 2  11 -484.04  2 5.2746    0.07155 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
# let's try removing site too (we'd keep it anyways, since we know sites differed in parasite load)
mod_beh_nosite_nbin <- glmmTMB(total.time.open.after ~ species + totalpara,
  family = "nbinom2",
  ziformula = ~.,
  data = all_data
)

# test site.id effect
lrtest(mod_beh_nosite_nbin, mod_beh_nolength_nbin) # there is a sig diff, so let's keep site
```

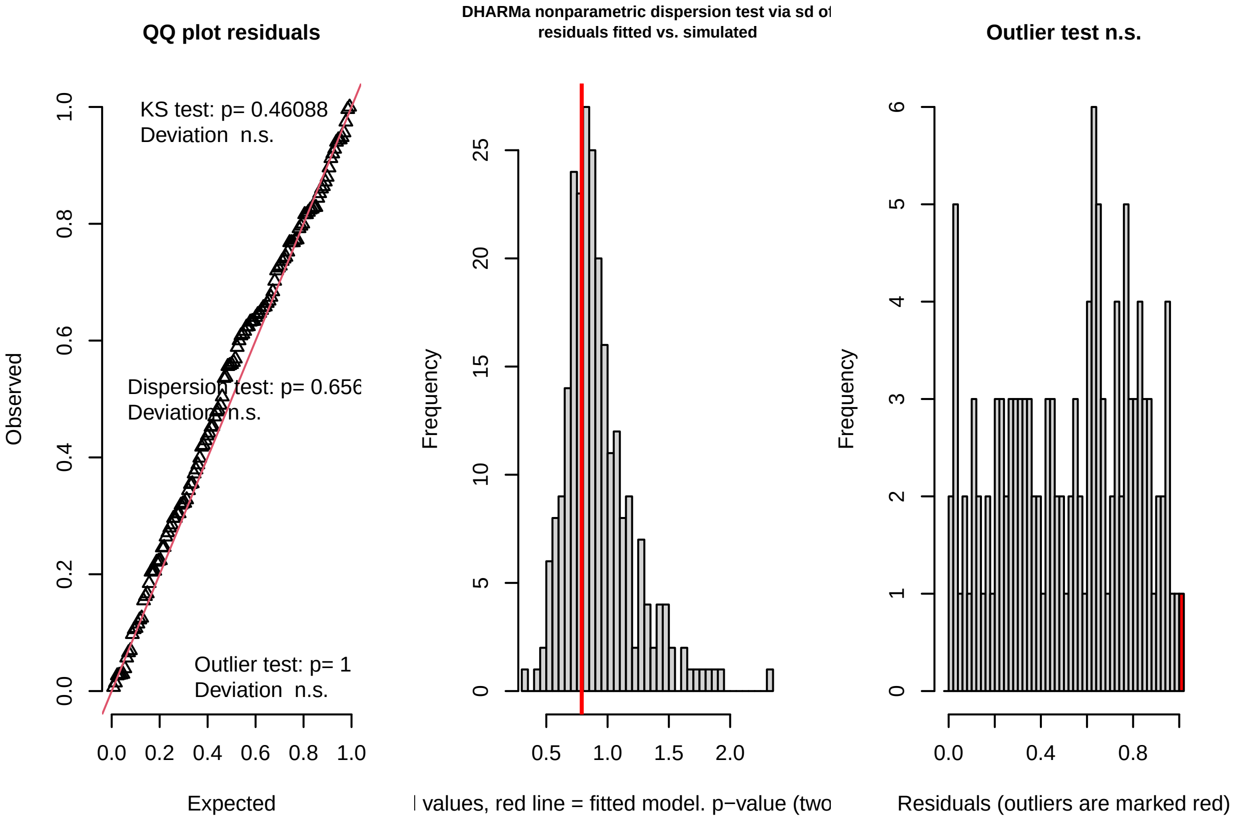
```
## Likelihood ratio test
##
## Model 1: total.time.open.after ~ species + totalpara
## Model 2: total.time.open.after ~ species + site.id + totalpara
##   #Df  LogLik Df  Chisq Pr(>Chisq)
## 1   7 -490.63
## 2   9 -486.68  2 7.9014    0.01924 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Ok so we're moving forward with a model without interactions, including site, species, and total parasites as predictors, but not including length. Let's check those assumptions.

```
# checking assumptions for the combined, no length model
# First we have to simulate your residuals
sim.output <- DHARMa::simulateResiduals(mod_beh_nolength_nbin)

# then you can plot them
plot(sim.output) # these look pretty good!
```



```r
# gives you a bunch of tests of dispersion etc
DHARMa::testResiduals(sim.output) # nice!
```



```
## $uniformity
##
##  Asymptotic one-sample Kolmogorov-Smirnov test
##
## data:  simulationOutput$scaledResiduals
## D = 0.075685, p-value = 0.4609
## alternative hypothesis: two-sided
##
##
## $dispersion
##
##  DHARMa nonparametric dispersion test via sd of residuals fitted vs.
##  simulated
##
## data:  simulationOutput
## dispersion = 0.84943, p-value = 0.656
## alternative hypothesis: two.sided
##
##
## $outliers
##
##  DHARMa bootstrapped outlier test
##
## data:  simulationOutput
## outliers at both margin(s) = 1, observations = 127, p-value = 1
## alternative hypothesis: two.sided
##  percent confidence interval:
##  0.00000000 0.02362205
## sample estimates:
## outlier frequency (expected: 0.00590551181102362 )
##                                          0.007874016
```

```
# from poking around, it seems like estimating overdispersion is how we evaluate goodness of fit (versu
check_overdispersion(mod_beh_nolength_nbin) # no overdispersion detected. dispersion ratio = 0.849, p =
```

```
## # Overdispersion test
##
##  dispersion ratio = 0.849
##           p-value = 0.656
```

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

So, what are the results?

```
summary(mod_beh_nolength_nbin) # species and site are significant predictors of any time spent in the o
```

```
##  Family: nbinom2  ( log )
## Formula:          total.time.open.after ~ species + site.id + totalpara
## Zero inflation:                         ~.
## Data: all_data
##
##      AIC      BIC   logLik deviance df.resid
##    991.4   1016.9   -486.7    973.4      118
##
##
## Dispersion parameter for nbinom2 family (): 1.44
```

```
##
## Conditional model:
##                      Estimate Std. Error z value Pr(>|z|)
## (Intercept)          5.707420   0.290608  19.640   <2e-16 ***
## speciesP. latipinna -0.242366   0.208730  -1.161   0.2456
## site.idWeslaco      -0.491179   0.275555  -1.783   0.0747 .
## totalpara           -0.002049   0.001137  -1.802   0.0715 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Zero-inflation model:
##                      Estimate Std. Error z value Pr(>|z|)
## (Intercept)         -0.389392   0.551167  -0.706  0.47989
## speciesP. latipinna  1.405873   0.438779   3.204  0.00136 **
## site.idWeslaco      -1.008341   0.474234  -2.126  0.03348 *
## totalpara           -0.001915   0.002399  -0.798  0.42471
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Let's do some post-hoc analysis.

```
emmeans(mod_beh_nolength_nbin, "species")
```

```
##  species      emmean    SE  df asymp.LCL asymp.UCL
##  P. formosa     5.35 0.165 Inf      5.03      5.67
##  P. latipinna   5.11 0.145 Inf      4.82      5.39
##
## Results are averaged over the levels of: site.id
## Results are given on the log (not the response) scale.
## Confidence level used: 0.95
```

```
emmeans(mod_beh_nolength_nbin, "site.id")
```

```
##  site.id     emmean    SE  df asymp.LCL asymp.UCL
##  Brownsville   5.47 0.216 Inf      5.05      5.90
##  Weslaco       4.98 0.133 Inf      4.72      5.24
##
## Results are averaged over the levels of: species
## Results are given on the log (not the response) scale.
## Confidence level used: 0.95
```

Note: should I also run a model with the all_data_long to see if fish change the proportion of time they spend in the open before/after the startle.

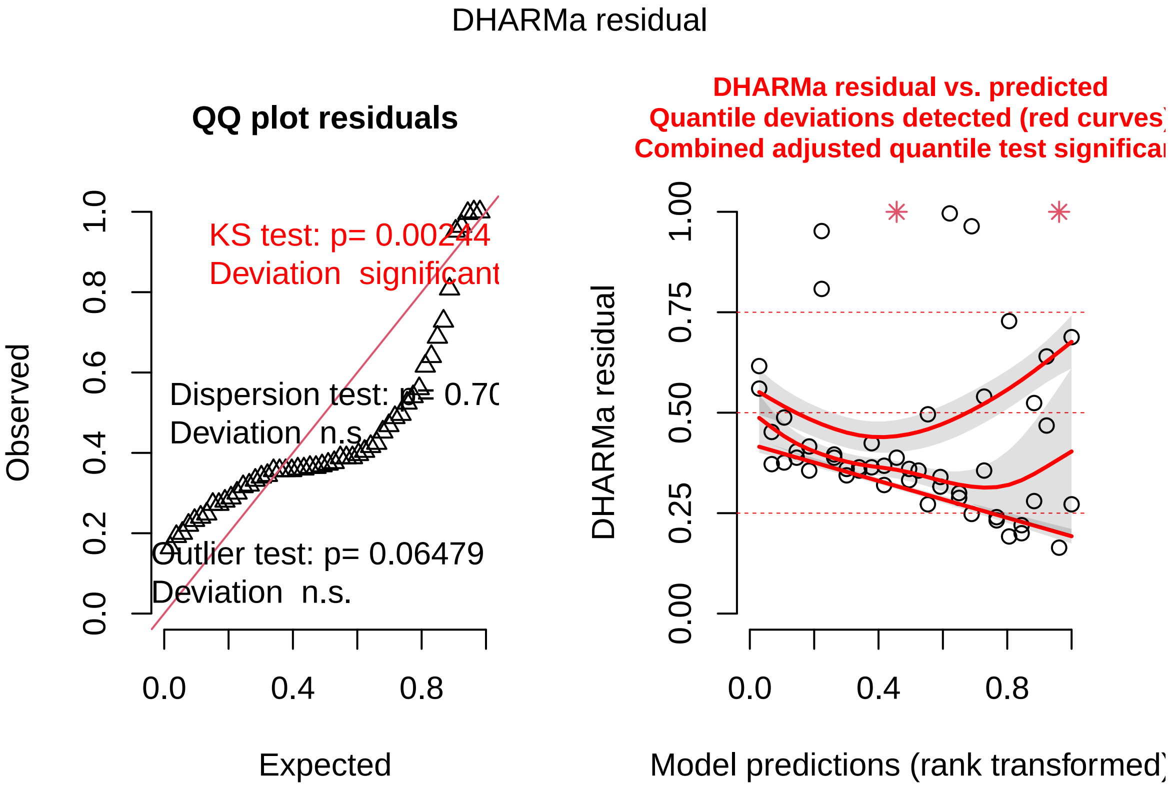
Brownsville Now let's repeat that analysis with just the Brownsville site.

Let's create a Brownsville-only dataset.

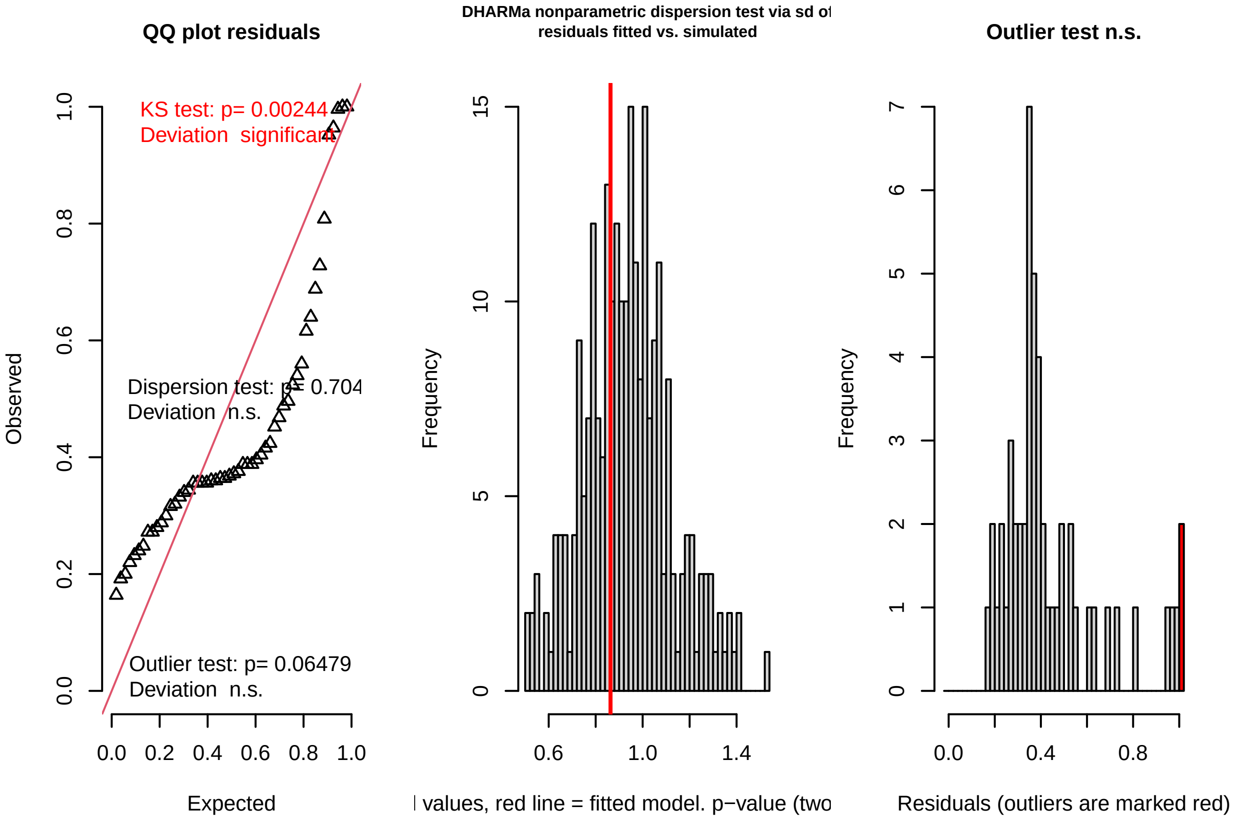
```
br_data <- all_data %>%
  filter(site.id == "Brownsville")
```

```
# FULL model, with three-way interaction
mod_beh_br <- lm(total.time.open.after ~ species * totalpara + standard.length,
                 data = br_data)

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

```
# then you can plot them
plot(sim.mod_beh_br) # these both look not great
```



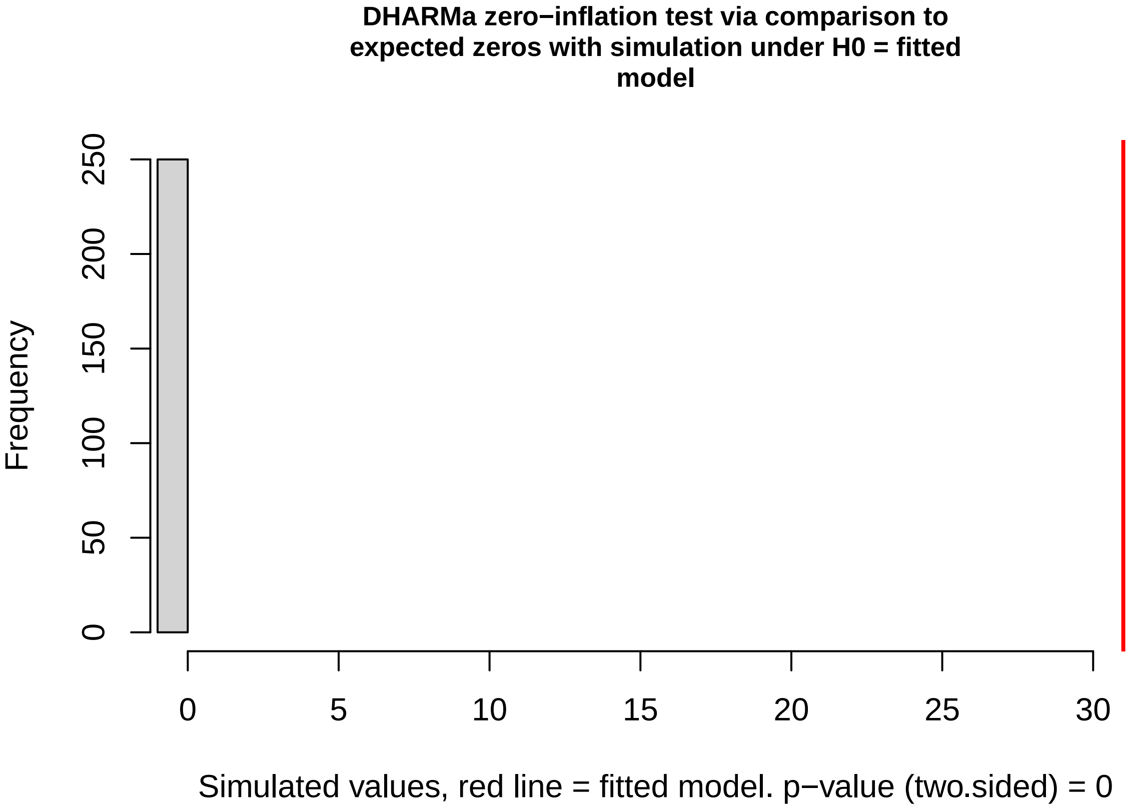
```
# gives you a bunch of tests of dispersion etc
DHARMa::testResiduals(sim.mod_beh_br) # dispersion and outlier look ok, but the QQ plot is bad
```



```
## $uniformity
##
##  Asymptotic one-sample Kolmogorov-Smirnov test
##
## data:  simulationOutput$scaledResiduals
## D = 0.254, p-value = 0.002438
## alternative hypothesis: two-sided
##
##
## $dispersion
##
##  DHARMa nonparametric dispersion test via sd of residuals fitted vs.
##  simulated
##
## data:  simulationOutput
## dispersion = 0.92165, p-value = 0.704
## alternative hypothesis: two.sided
##
##
## $outliers
##
##  DHARMa outlier test based on exact binomial test with approximate
##  expectations
##
## data:  simulationOutput
## outliers at both margin(s) = 2, observations = 52, p-value = 0.06479
## alternative hypothesis: true probability of success is not equal to 0.007968127
## 95 percent confidence interval:
```

```
##  0.004692289 0.132128407
## sample estimates:
## frequency of outliers (expected: 0.00796812749003984 )
##                                               0.03846154
```

```
# yes we can see that the data is super zero inflated
DHARMa::testZeroInflation(sim.mod_beh_br) # 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 now let's try to fit some different models.

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

```
## 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
```

```
mod_beh_br_nbin <- glmmTMB(total.time.open.after ~ species * totalpara + standard.length,
  family = "nbinom2",
  ziformula = ~.,
  data = br_data
)

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

```
## # Overdispersion test
##
##  dispersion ratio =   2.182
##           p-value = < 0.001
```

```
## Overdispersion detected.
```

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

```
## # Overdispersion test
##
##  dispersion ratio = 0.915
##           p-value = 0.872
```

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

```
lrtest(mod_beh_br_poisson, mod_beh_br_nbin) # there is a sig difference between the two
```

```
## Likelihood ratio test
##
## Model 1: total.time.open.after ~ species * totalpara + standard.length
## Model 2: total.time.open.after ~ species * totalpara + standard.length
##   #Df   LogLik Df  Chisq Pr(>Chisq)
## 1  10 -1563.51
## 2  11  -161.23  1 2804.6  < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
# checking AIC
anova(mod_beh_br_poisson, mod_beh_br_nbin) # nbin model is much lower
```

```
## Data: br_data
## Models:
## mod_beh_br_poisson: total.time.open.after ~ species * totalpara + standard.length, zi=~., disp=~1
## mod_beh_br_nbin: total.time.open.after ~ species * totalpara + standard.length, zi=~., disp=~1
##                    Df     AIC    BIC   logLik deviance  Chisq Chi Df Pr(>Chisq)
## mod_beh_br_poisson 10 3147.02 3166.5 -1563.51  3127.02
## mod_beh_br_nbin    11  344.46  365.9  -161.23   322.46 2804.6      1  < 2.2e-16
##
## mod_beh_br_poisson
## mod_beh_br_nbin    ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Ok, let's move forward with the nbin model. Time for some backwards model selection.

```
# interaction model
mod_beh_br_nbin <- glmmTMB(total.time.open.after ~ species * totalpara + standard.length,
  family = "nbinom2",
  ziformula = ~.,
  data = br_data
)

# removing species interaction
mod_spcomb_br_nbin <- glmmTMB(total.time.open.after ~ species + totalpara + standard.length,
  family = "nbinom2",
  ziformula = ~.,
  data = br_data
)

# test 2-way with log likelihood ratio test
lrtest(mod_beh_br_nbin, mod_spcomb_br_nbin) # no difference, so let's start dropping terms
```

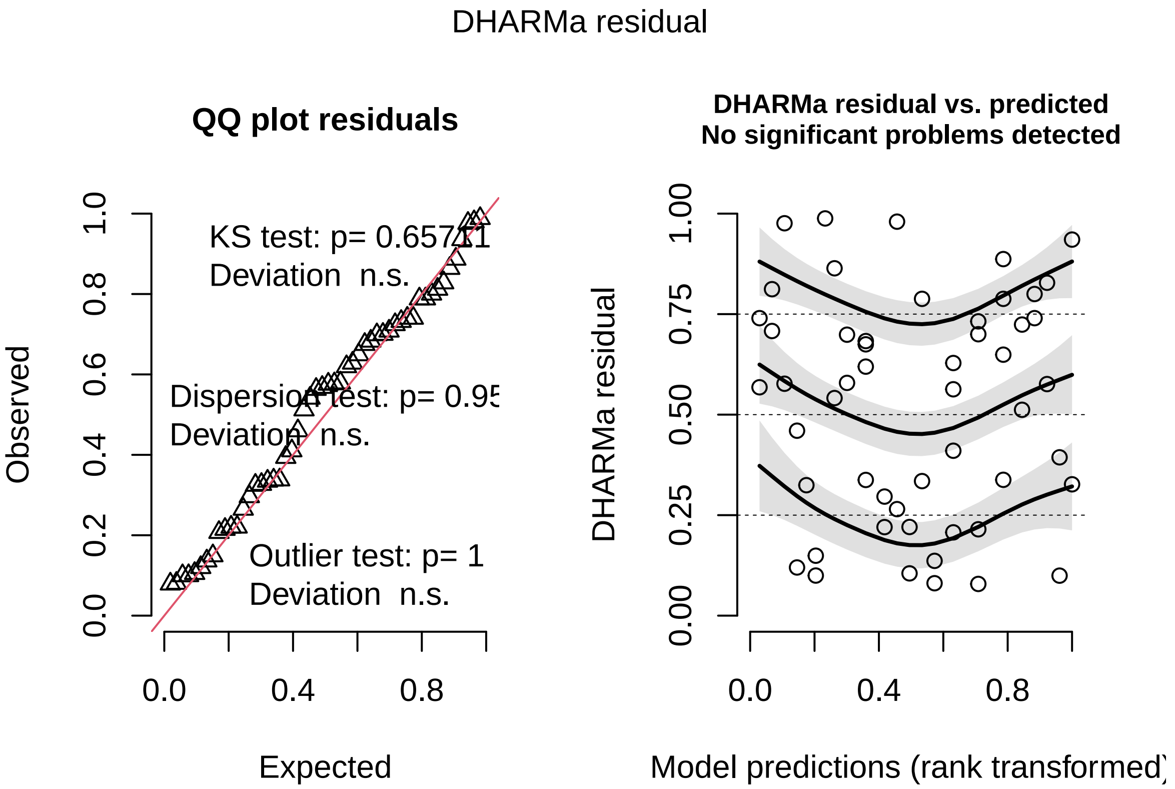
```
## Likelihood ratio test
##
## Model 1: total.time.open.after ~ species * totalpara + standard.length
## Model 2: total.time.open.after ~ species + totalpara + standard.length
##   #Df  LogLik Df  Chisq Pr(>Chisq)
## 1  11 -161.23
## 2   9 -161.94 -2 1.4126     0.4935
```

```
# no length
mod_beh_nolength_br_nbin <- glmmTMB(total.time.open.after ~ species + totalpara,
  family = "nbinom2",
  ziformula = ~.,
  data = br_data
)

# test length effect
lrtest(mod_beh_nolength_br_nbin, mod_spcomb_br_nbin) # no sig diff, let's drop length
```

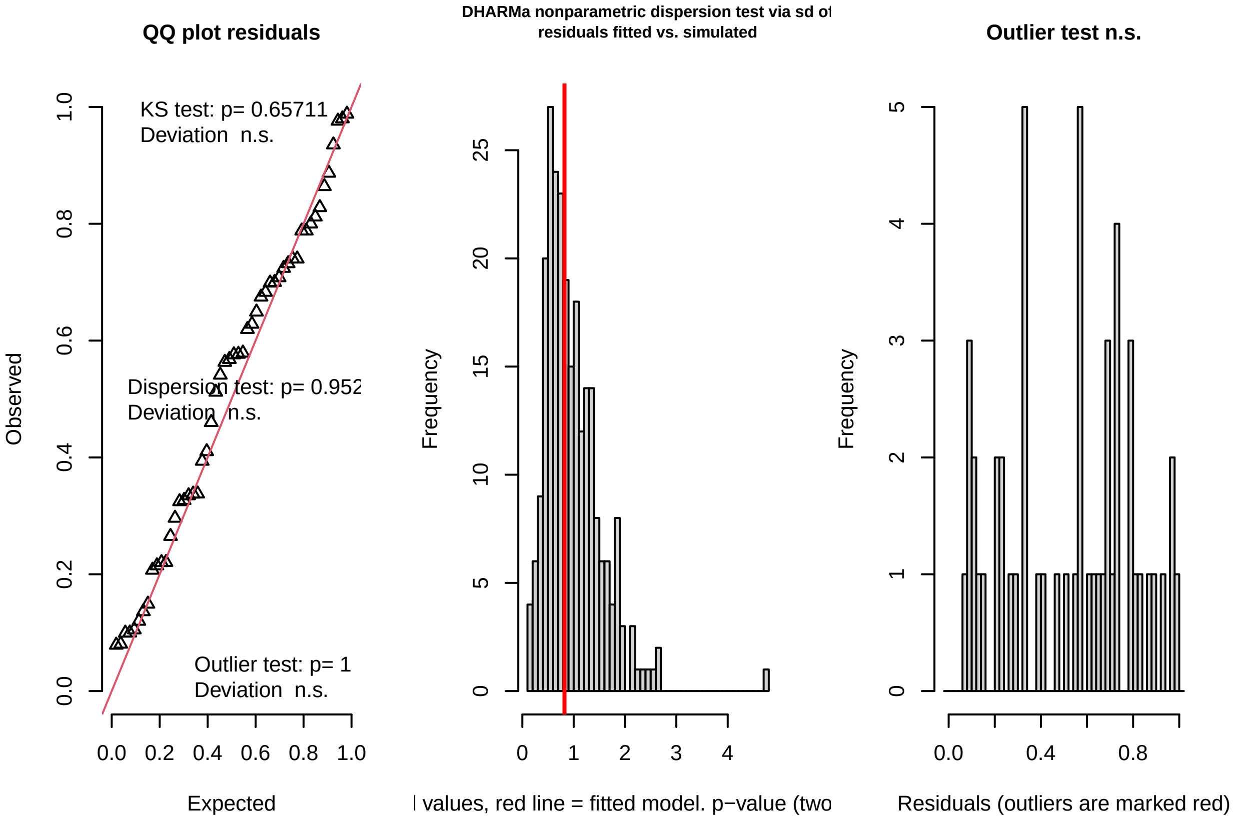
```
## Likelihood ratio test
##
## Model 1: total.time.open.after ~ species + totalpara
## Model 2: total.time.open.after ~ species + totalpara + standard.length
##   #Df  LogLik Df  Chisq Pr(>Chisq)
## 1   7 -163.24
## 2   9 -161.94  2 2.6129     0.2708
```

Ok so we're moving forward with a model without interactions, including species and total parasites as predictors, but not including length. Let's check those assumptions.

```
# checking assumptions for the combined, no length model
# First we have to simulate your residuals
sim.output <- DHARMa::simulateResiduals(mod_beh_nolength_br_nbin)

# then you can plot them
plot(sim.output) # these look pretty good!
```



```r
# gives you a bunch of tests of dispersion etc
DHARMa::testResiduals(sim.output) # nice!
```



```
## $uniformity
##
##  Asymptotic one-sample Kolmogorov-Smirnov test
##
## data:  simulationOutput$scaledResiduals
## D = 0.10154, p-value = 0.6571
## alternative hypothesis: two-sided
##
##
## $dispersion
##
##  DHARMa nonparametric dispersion test via sd of residuals fitted vs.
##  simulated
##
## data:  simulationOutput
## dispersion = 0.83562, p-value = 0.952
## alternative hypothesis: two.sided
##
##
## $outliers
##
##  DHARMa bootstrapped outlier test
##
## data:  simulationOutput
## outliers at both margin(s) = 0, observations = 52, p-value = 1
## alternative hypothesis: two.sided
##  percent confidence interval:
##  0.00000000 0.03846154
## sample estimates:
## outlier frequency (expected: 0.00596153846153846 )
##                                                   0
```

```
# let's check overdispersion
check_overdispersion(mod_beh_nolength_br_nbin) # no overdispersion detected. dispersion ratio = 0.836,
```

```
## # Overdispersion test
##
##  dispersion ratio = 0.836
##           p-value = 0.952
```

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

So, what are the results?

```
summary(mod_beh_nolength_br_nbin) # no sig effects. Total para is marginally significant in the conditi
```

```
##  Family: nbinom2  ( log )
## Formula:          total.time.open.after ~ species + totalpara
## Zero inflation:                         ~.
## Data: br_data
##
##      AIC      BIC   logLik deviance df.resid
##    340.5    354.1   -163.2    326.5       45
##
##
## Dispersion parameter for nbinom2 family (): 0.967
```

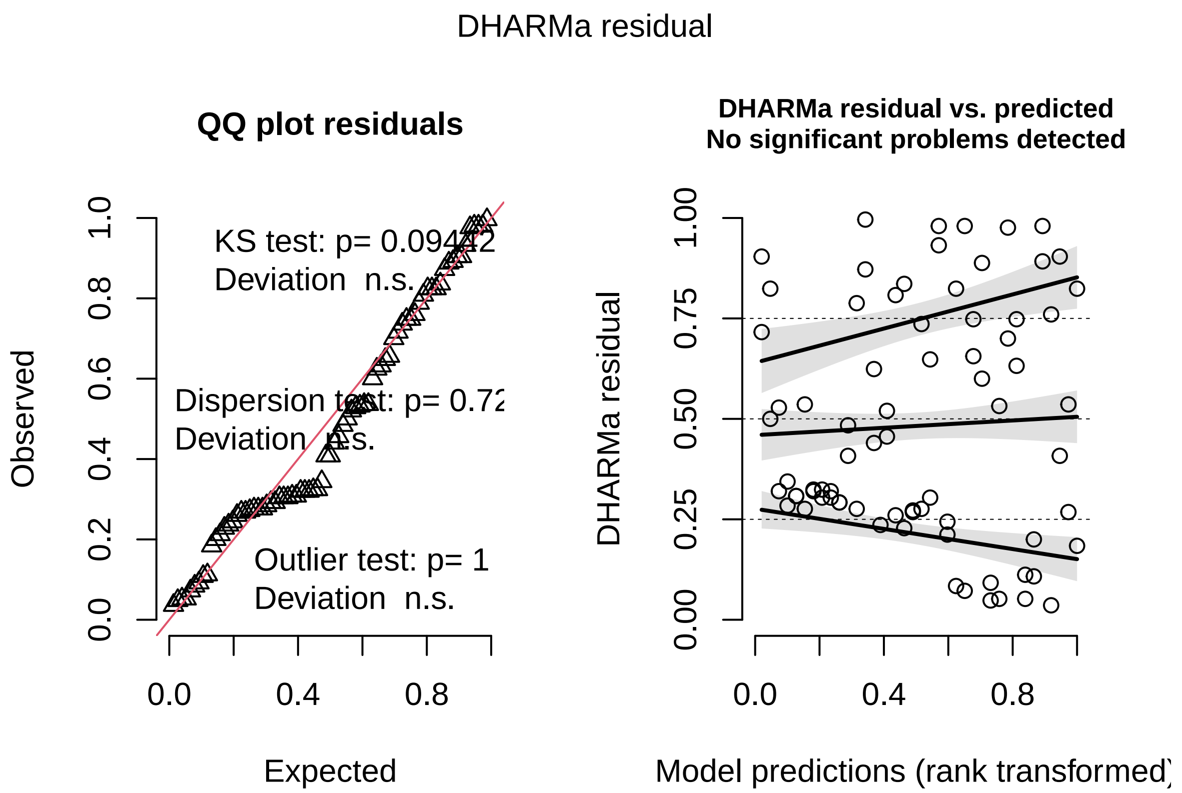
```
##
## Conditional model:
##                      Estimate Std. Error z value Pr(>|z|)
## (Intercept)          5.477100   0.424557  12.901   <2e-16 ***
## speciesP. latipinna  0.152236   0.456954   0.333   0.7390
## totalpara           -0.002176   0.001308  -1.664   0.0962 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Zero-inflation model:
##                      Estimate Std. Error z value Pr(>|z|)
## (Intercept)         -0.257027   0.713155  -0.360    0.719
## speciesP. latipinna  1.199569   0.698595   1.717    0.086 .
## totalpara           -0.001856   0.002440  -0.761    0.447
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Weslaco One more time! With Weslaco.

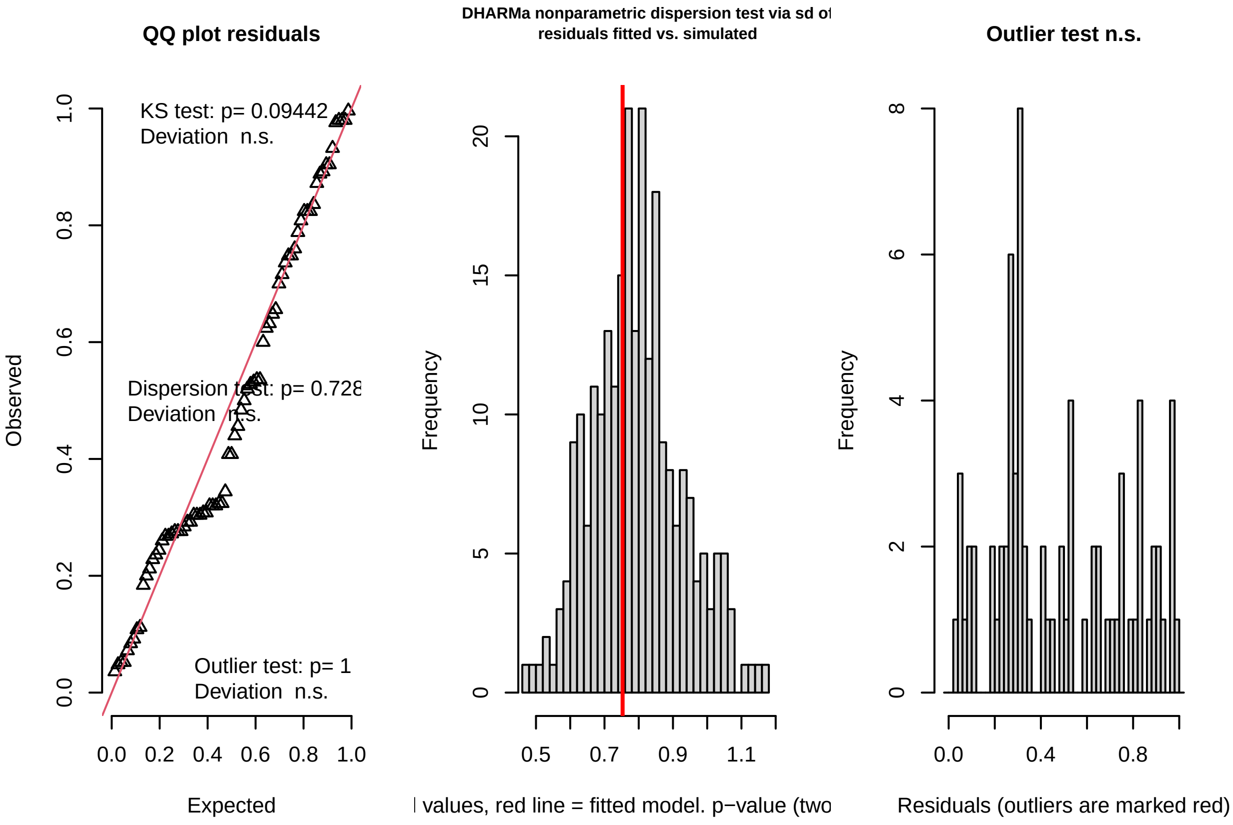
Let's create a Weslaco-only dataset.

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

```
# FULL model, with three-way interaction
mod_beh_wes <- lm(total.time.open.after ~ species * totalpara + standard.length,
                   data = wes_data)

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

# then you can plot them
plot(sim.mod_beh_wes) # all look ok
```

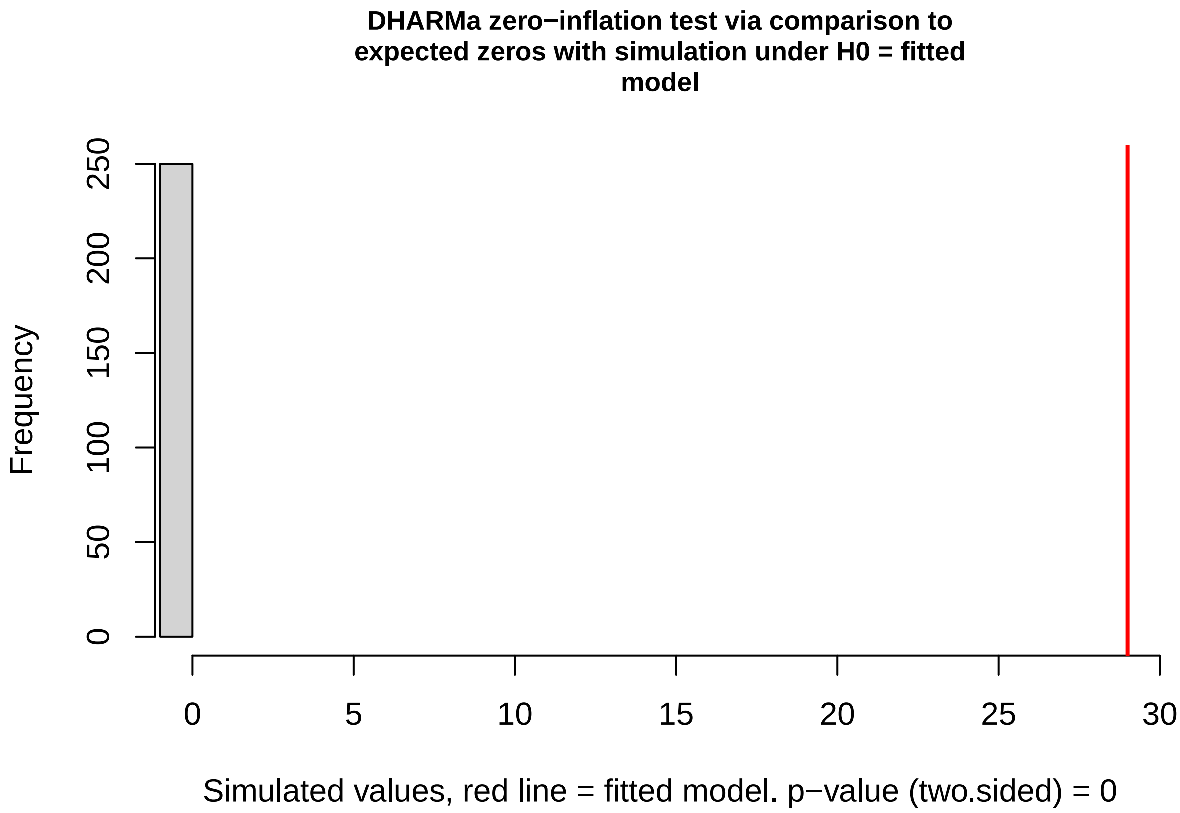


```
# gives you a bunch of tests of dispersion etc
DHARMa::testResiduals(sim.mod_beh_wes) # all ok
```



```
## $uniformity
##
##  Asymptotic one-sample Kolmogorov-Smirnov test
##
## data:  simulationOutput$scaledResiduals
## D = 0.14267, p-value = 0.09442
## alternative hypothesis: two-sided
##
##
## $dispersion
##
##  DHARMa nonparametric dispersion test via sd of residuals fitted vs.
##  simulated
##
## data:  simulationOutput
## dispersion = 0.94599, p-value = 0.728
## 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 = 75, p-value = 1
## alternative hypothesis: true probability of success is not equal to 0.007968127
## 95 percent confidence interval:
##  0.00000000 0.04799506
## sample estimates:
## frequency of outliers (expected: 0.00796812749003984 )
##                                                       0
```

```
# yes we can see that the data is super zero inflated
DHARMa::testZeroInflation(sim.mod_beh_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 now let's try to fit some different models.

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

```
## 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
```

```
mod_beh_wes_nbin <- glmmTMB(total.time.open.after ~ species * totalpara + standard.length,
  family = "nbinom2",
  ziformula = ~.,
  data = wes_data
)

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

```
## # Overdispersion test
##
##  dispersion ratio =   2.047
##           p-value = < 0.001
```

```
## Overdispersion detected.
```

```r
check_overdispersion(mod_beh_wes_nbin) # data is not over dispersed
```

```
## # Overdispersion test
##
##  dispersion ratio = 0.727
##           p-value = 0.408
```

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

```r
lrtest(mod_beh_wes_poisson, mod_beh_wes_nbin) # there is a sig difference between the two
```

```
## Likelihood ratio test
##
## Model 1: total.time.open.after ~ species * totalpara + standard.length
## Model 2: total.time.open.after ~ species * totalpara + standard.length
##   #Df   LogLik Df  Chisq Pr(>Chisq)
## 1  10 -1425.98
## 2  11  -313.75  1 2224.5  < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
# checking AIC
anova(mod_beh_wes_poisson, mod_beh_wes_nbin) # nbin model is much lower
```

```
## Data: wes_data
## Models:
## mod_beh_wes_poisson: total.time.open.after ~ species * totalpara + standard.length, zi=~., disp=~1
## mod_beh_wes_nbin: total.time.open.after ~ species * totalpara + standard.length, zi=~., disp=~1
##                     Df    AIC    BIC   logLik deviance  Chisq Chi Df Pr(>Chisq)
## mod_beh_wes_poisson 10 2872.0 2895.1 -1425.98   2852.0
## mod_beh_wes_nbin    11  649.5  675.0  -313.75    627.5 2224.5      1  < 2.2e-16
##
## mod_beh_wes_poisson
## mod_beh_wes_nbin    ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Ok, let's move forward with the nbin model. Time for some backwards model selection.

```r
# interaction model
mod_beh_wes_nbin <- glmmTMB(total.time.open.after ~ species * totalpara + standard.length,
  family = "nbinom2",
  ziformula = ~.,
  data = wes_data
)

# removing species interaction
mod_spcomb_wes_nbin <- glmmTMB(total.time.open.after ~ species + totalpara + standard.length,
  family = "nbinom2",
  ziformula = ~.,
  data = wes_data
)

# test 2-way with log likelihood ratio test
lrtest(mod_beh_wes_nbin, mod_spcomb_wes_nbin) # sig difference, so let's keep the interaction but try d
```

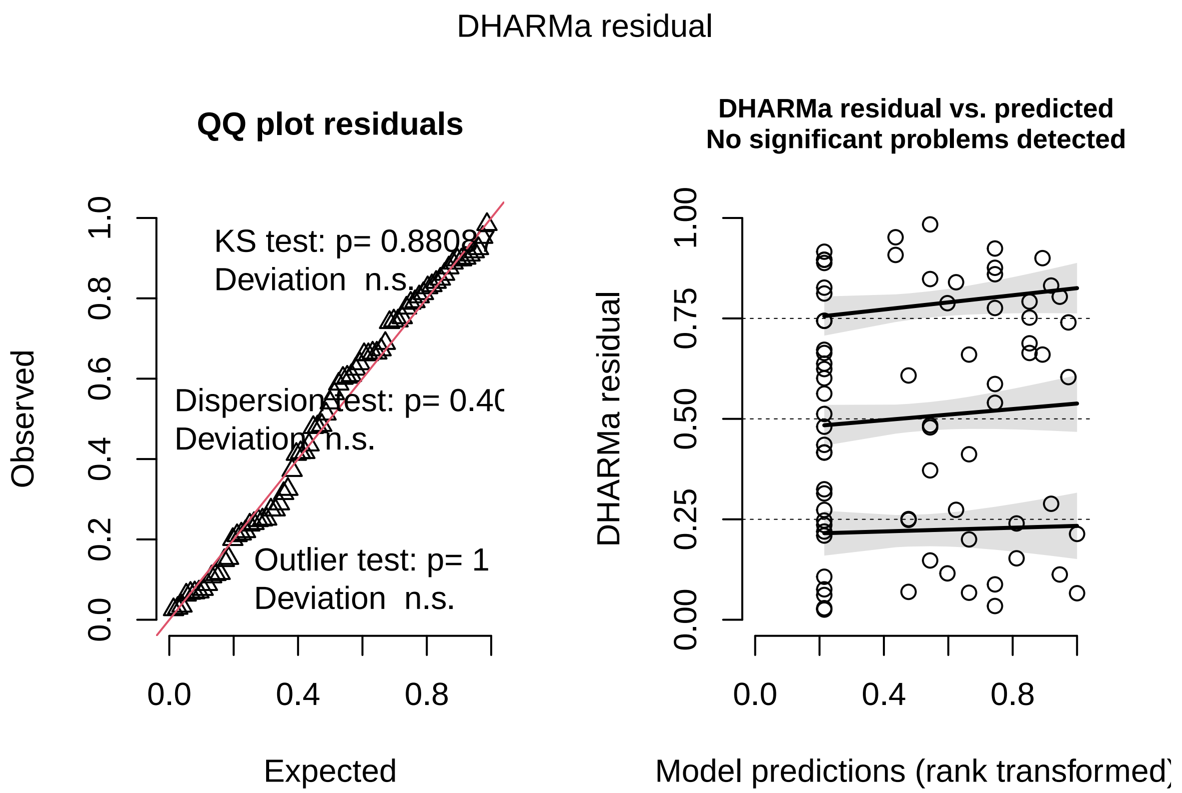
```
## Likelihood ratio test
##
## Model 1: total.time.open.after ~ species * totalpara + standard.length
## Model 2: total.time.open.after ~ species + totalpara + standard.length
##   #Df  LogLik Df Chisq Pr(>Chisq)
## 1  11 -313.75
## 2   9 -317.67 -2 7.838    0.01986 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
# no length
mod_beh_nolength_wes_nbin <- glmmTMB(total.time.open.after ~ species * totalpara,
  family = "nbinom2",
  ziformula = ~.,
  data = wes_data
)

# test length effect
lrtest(mod_beh_nolength_wes_nbin, mod_beh_wes_nbin) # no sig diff, let's drop length
```

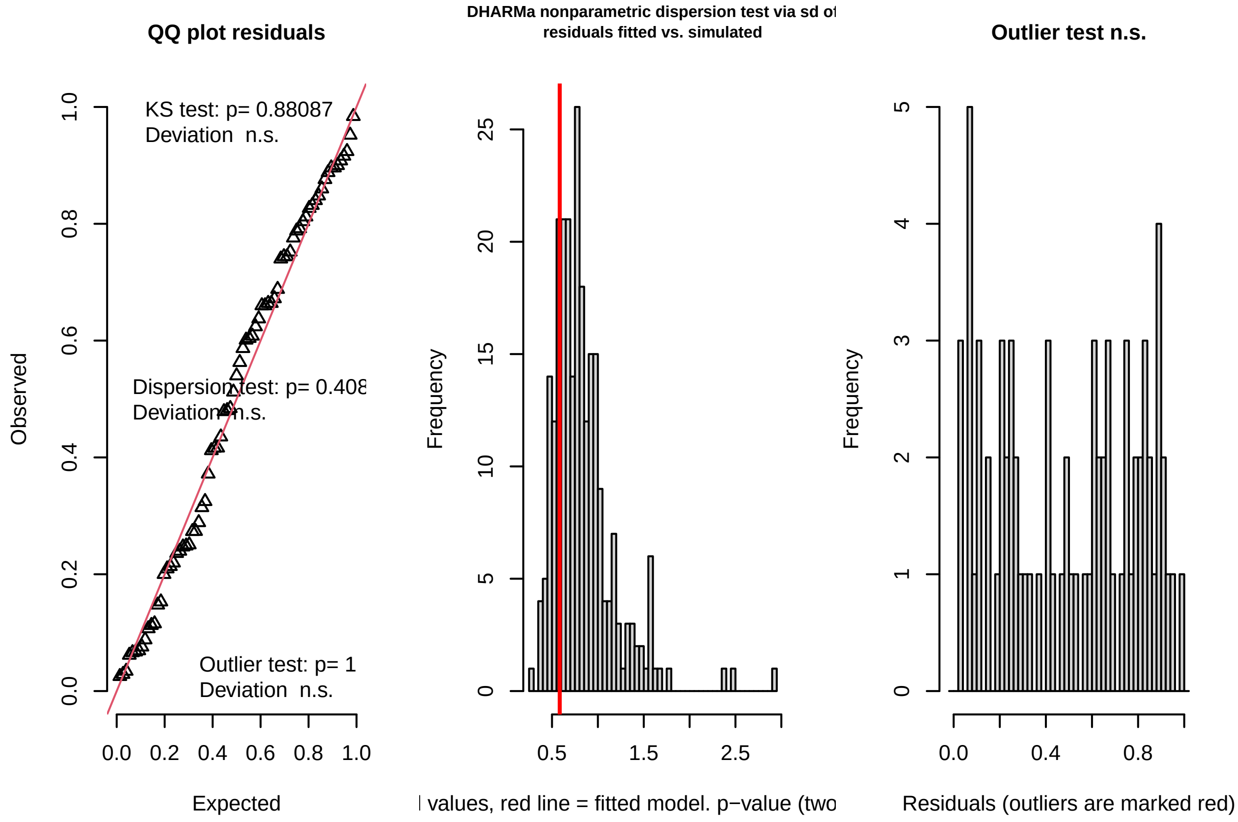
```
## Likelihood ratio test
##
## Model 1: total.time.open.after ~ species * totalpara
## Model 2: total.time.open.after ~ species * totalpara + standard.length
##   #Df  LogLik Df  Chisq Pr(>Chisq)
## 1   9 -314.89
## 2  11 -313.75  2 2.2804     0.3198
```

Ok so we're moving forward with a model without interactions, including species and total parasites as predictors, but not including length. Let's check those assumptions.

```
# checking assumptions for the combined, no length model
# First we have to simulate your residuals
sim.output <- DHARMa::simulateResiduals(mod_beh_nolength_wes_nbin)

# then you can plot them
plot(sim.output) # these look pretty good!
```



```r
# gives you a bunch of tests of dispersion etc
DHARMa::testResiduals(sim.output) # nice!
```



```
## $uniformity
##
##  Asymptotic one-sample Kolmogorov-Smirnov test
##
## data:  simulationOutput$scaledResiduals
## D = 0.067794, p-value = 0.8809
## alternative hypothesis: two-sided
##
##
## $dispersion
##
##  DHARMa nonparametric dispersion test via sd of residuals fitted vs.
##  simulated
##
## data:  simulationOutput
## dispersion = 0.69706, p-value = 0.408
## alternative hypothesis: two.sided
##
##
## $outliers
##
##  DHARMa bootstrapped outlier test
##
## data:  simulationOutput
## outliers at both margin(s) = 0, observations = 75, p-value = 1
## alternative hypothesis: two.sided
##  percent confidence interval:
##  0.00000000 0.02666667
## sample estimates:
## outlier frequency (expected: 0.00666666666666667 )
##                                                 0
```

```r
# let's check overdispersion
check_overdispersion(mod_beh_nolength_wes_nbin) # no overdispersion detected. dispersion ratio = 0.697,
```

```
## # Overdispersion test
##
##  dispersion ratio = 0.697
##           p-value = 0.408
```

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

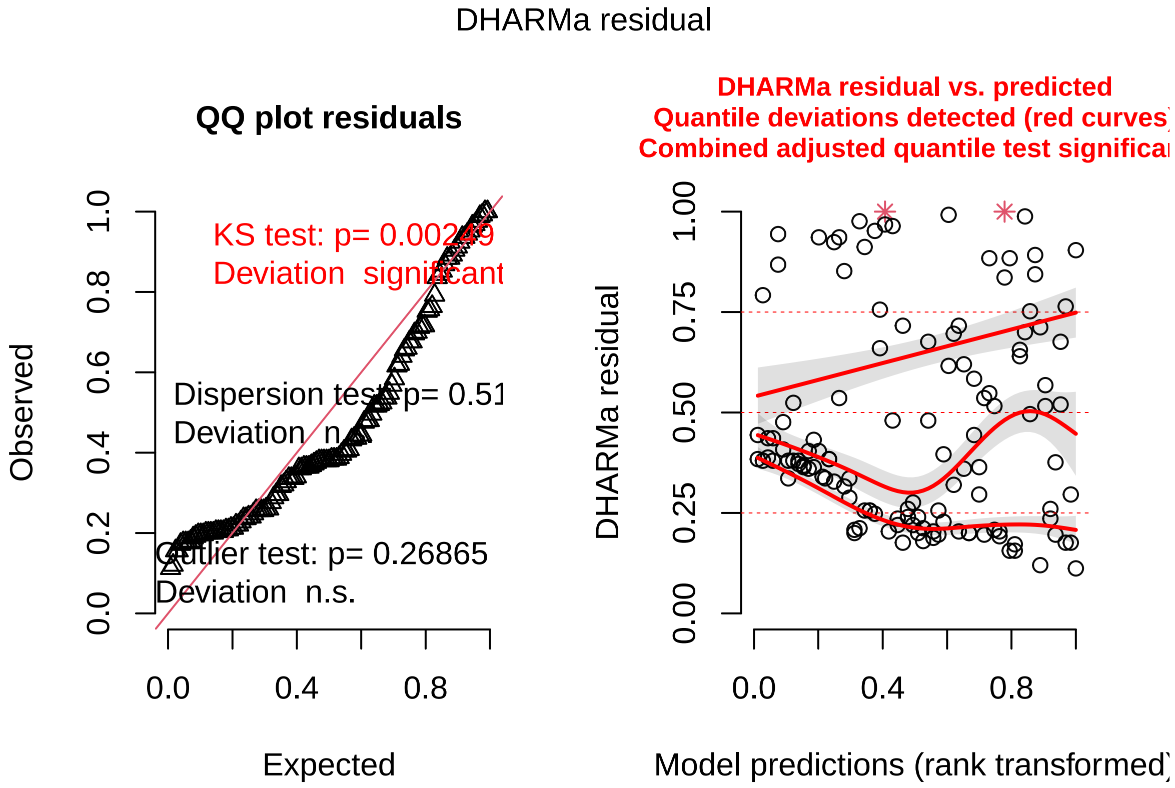
So, what are the results?

```r
summary(mod_beh_nolength_wes_nbin) # no sig effects in conditional model. Species and total para are si
```

```
##  Family: nbinom2  ( log )
## Formula:          total.time.open.after ~ species * totalpara
## Zero inflation:                         ~.
## Data: wes_data
##
##      AIC      BIC   logLik deviance df.resid
##    647.8    668.6   -314.9    629.8       66
##
##
## Dispersion parameter for nbinom2 family (): 2.06
```

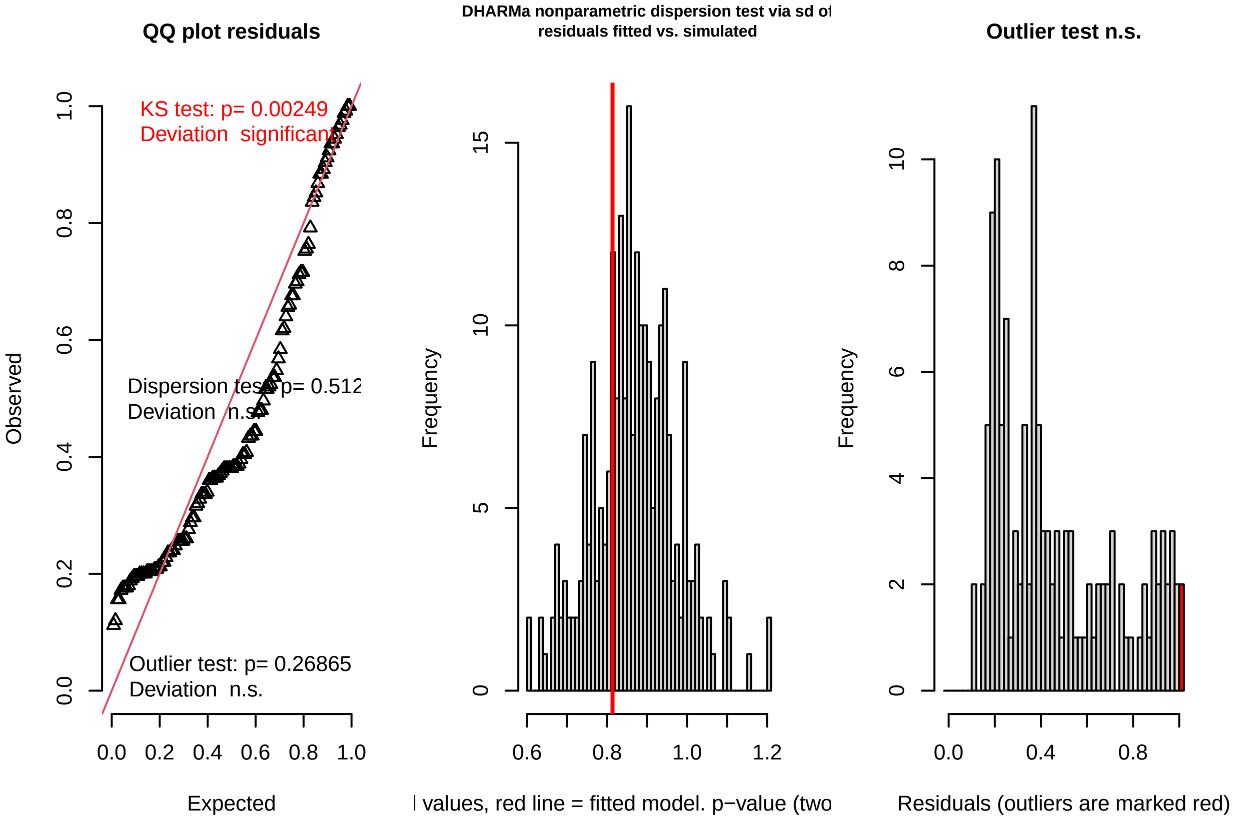
```
##
## Conditional model:
##                              Estimate Std. Error z value Pr(>|z|)
## (Intercept)                   5.10842    0.19158  26.665   <2e-16 ***
## speciesP. latipinna          -0.29385    0.24637  -1.193    0.233
## totalpara                     0.02526    0.02019   1.251    0.211
## speciesP. latipinna:totalpara -0.02206    0.02074  -1.064    0.287
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Zero-inflation model:
##                              Estimate Std. Error z value Pr(>|z|)
## (Intercept)                  -2.57084    0.80982  -3.175  0.00150 **
## speciesP. latipinna           2.81461    0.88073   3.196  0.00139 **
## totalpara                     0.09612    0.04814   1.997  0.04587 *
## speciesP. latipinna:totalpara -0.22040    0.20113  -1.096  0.27318
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
mod_switch_full <- lm(total.behavior.switches.after ~ species * site.id * totalpara + standard.length,
                      data = all_data)

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

# then you can plot them
plot(sim.mod_switch) # these both look not great
```



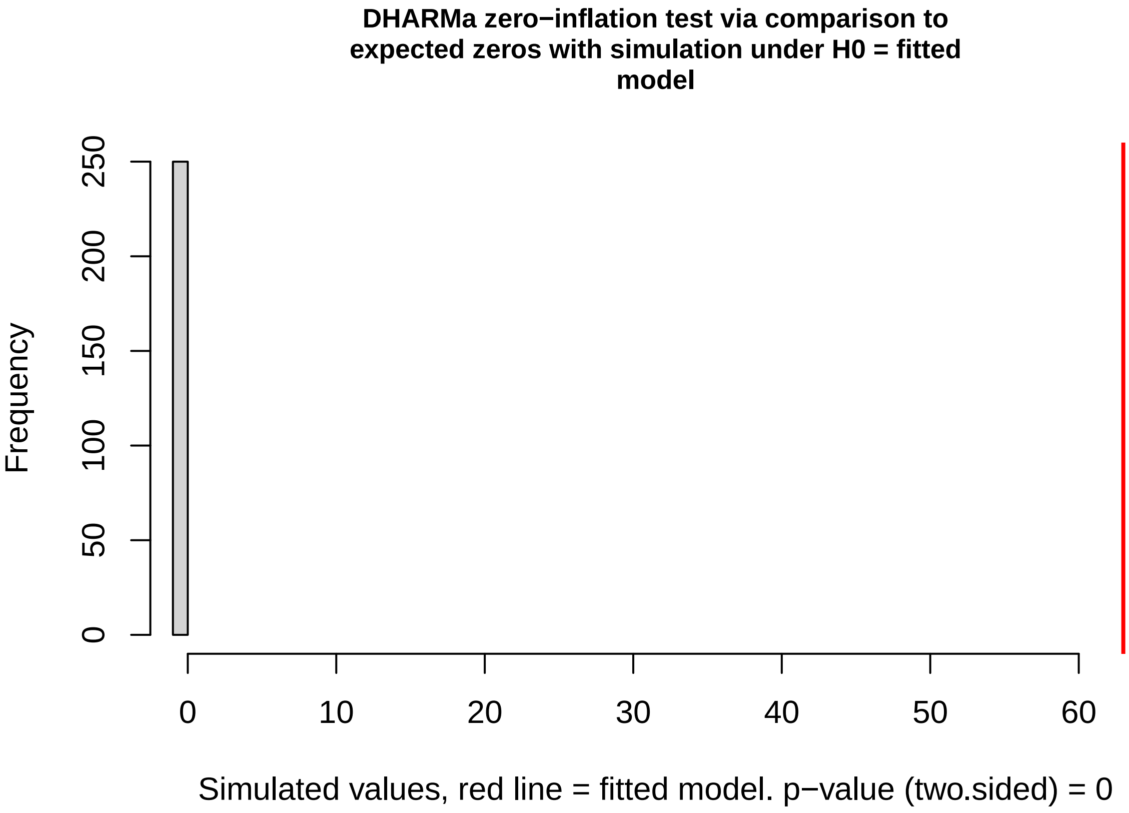
Switching

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



```
## $uniformity
##
##  Asymptotic one-sample Kolmogorov-Smirnov test
##
## data:  simulationOutput$scaledResiduals
## D = 0.1623, p-value = 0.002485
## alternative hypothesis: two-sided
##
##
## $dispersion
##
##  DHARMa nonparametric dispersion test via sd of residuals fitted vs.
##  simulated
##
## data:  simulationOutput
## dispersion = 0.93602, p-value = 0.512
## alternative hypothesis: two.sided
##
##
## $outliers
##
##  DHARMa outlier test based on exact binomial test with approximate
##  expectations
##
## data:  simulationOutput
## outliers at both margin(s) = 2, observations = 127, p-value = 0.2687
## alternative hypothesis: true probability of success is not equal to 0.007968127
## 95 percent confidence interval:
##  0.001912882 0.055729087
## sample estimates:
## frequency of outliers (expected: 0.00796812749003984 )
##                                            0.01574803
```

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

```
mod_switch_poisson <- glmmTMB(total.behavior.switches.after ~ species * site.id * totalpara + standard.
  family = "poisson",
  ziformula = ~.,
  data = all_data
)
```

```
## 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
```

```r
mod_switch_nbin <- glmmTMB(total.behavior.switches.after ~ species * site.id * totalpara + standard.leng
  family = "nbinom2",
  ziformula = ~.,
  data = all_data
)

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

```
## # Overdispersion test
##
##  dispersion ratio =   1.588
##           p-value = < 0.001
```

```
## Overdispersion detected.
```

```r
check_overdispersion(mod_switch_nbin) # data is not over dispersed
```

```
## # Overdispersion test
##
##  dispersion ratio = 1.072
##           p-value =  0.68
```

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

```r
lrtest(mod_switch_poisson, mod_switch_nbin) # there is a sig difference between the two
```

```
## Likelihood ratio test
##
## Model 1: total.behavior.switches.after ~ species * site.id * totalpara +
##     standard.length
## Model 2: total.behavior.switches.after ~ species * site.id * totalpara +
##     standard.length
##   #Df  LogLik Df  Chisq Pr(>Chisq)
## 1  18 -304.97
## 2  19 -266.93  1 76.088  < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
# checking AIC
anova(mod_switch_poisson, mod_switch_nbin) # nbin model is much lower
```

```
## Data: all_data
## Models:
## mod_switch_poisson: total.behavior.switches.after ~ species * site.id * totalpara + , zi=~., disp=~1
## mod_switch_poisson:     standard.length, zi=~., disp=~1
## mod_switch_nbin: total.behavior.switches.after ~ species * site.id * totalpara + , zi=~., disp=~1
## mod_switch_nbin:     standard.length, zi=~., disp=~1
##                    Df    AIC    BIC  logLik deviance  Chisq Chi Df Pr(>Chisq)
## mod_switch_poisson 18 645.95 697.14 -304.97   609.95
## mod_switch_nbin    19 571.86 625.90 -266.93   533.86 76.088      1  < 2.2e-16
##
## mod_switch_poisson
## mod_switch_nbin    ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
# removing species interaction
mod_spcomb_switch_nbin <- glmmTMB(total.behavior.switches.after ~ species + site.id * totalpara + standa
  family = "nbinom2",
  ziformula = ~.,
  data = all_data
)

# test 2-way with log likelihood ratio test
lrtest(mod_switch_nbin, mod_spcomb_switch_nbin) # no difference, so let's try removing all interactions
```

```
## Likelihood ratio test
##
## Model 1: total.behavior.switches.after ~ species * site.id * totalpara +
##     standard.length
## Model 2: total.behavior.switches.after ~ species + site.id * totalpara +
##     standard.length
##   #Df  LogLik Df  Chisq Pr(>Chisq)
## 1  19 -266.93
## 2  13 -271.05 -6 8.2409      0.221
```

```
# combined model
mod_switch_combined_nbin <- glmmTMB(total.behavior.switches.after ~ species + site.id + totalpara + stan
  family = "nbinom2",
  ziformula = ~.,
  data = all_data
)

# test species effect
lrtest(mod_switch_nbin, mod_switch_combined_nbin) # No difference, so let's see how the combined does c
```

```
## Likelihood ratio test
##
## Model 1: total.behavior.switches.after ~ species * site.id * totalpara +
##     standard.length
## Model 2: total.behavior.switches.after ~ species + site.id + totalpara +
##     standard.length
##   #Df  LogLik Df  Chisq Pr(>Chisq)
## 1  19 -266.93
## 2  11 -271.84 -8 9.8246     0.2776
```

```
# no length
mod_switch_nolength_nbin <- glmmTMB(total.behavior.switches.after ~ species + site.id + totalpara,
  family = "nbinom2",
  ziformula = ~.,
  data = all_data
)

# test length effect
lrtest(mod_switch_nolength_nbin, mod_switch_combined_nbin) # sig diff, let's stop here and use the one 
```

```
## Likelihood ratio test
##
## Model 1: total.behavior.switches.after ~ species + site.id + totalpara
## Model 2: total.behavior.switches.after ~ species + site.id + totalpara +
##     standard.length
##   #Df  LogLik Df Chisq Pr(>Chisq)
## 1   9 -274.75
## 2  11 -271.84  2 5.818    0.05453 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
summary(mod_switch_nolength_nbin)
```

```
##  Family: nbinom2  ( log )
## Formula:          total.behavior.switches.after ~ species + site.id + totalpara
## Zero inflation:                                 ~.
## Data: all_data
##
##      AIC      BIC   logLik deviance df.resid
##    567.5    593.1   -274.7    549.5      118
##
##
## Dispersion parameter for nbinom2 family (): 3.59
##
## Conditional model:
##                        Estimate Std. Error z value Pr(>|z|)
## (Intercept)          2.1576430  0.2229058   9.680   <2e-16 ***
## speciesP. latipinna  0.0415143  0.1568096   0.265    0.791
## site.idWeslaco       0.1803015  0.2149002   0.839    0.401
## totalpara           -0.0002905  0.0008147  -0.357    0.721
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Zero-inflation model:
##                       Estimate Std. Error z value Pr(>|z|)
## (Intercept)         -0.016382   0.561912  -0.029  0.97674
## speciesP. latipinna  1.381958   0.444306   3.110  0.00187 **
## site.idWeslaco      -1.376098   0.495233  -2.779  0.00546 **
## totalpara           -0.002702   0.002527  -1.069  0.28494
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

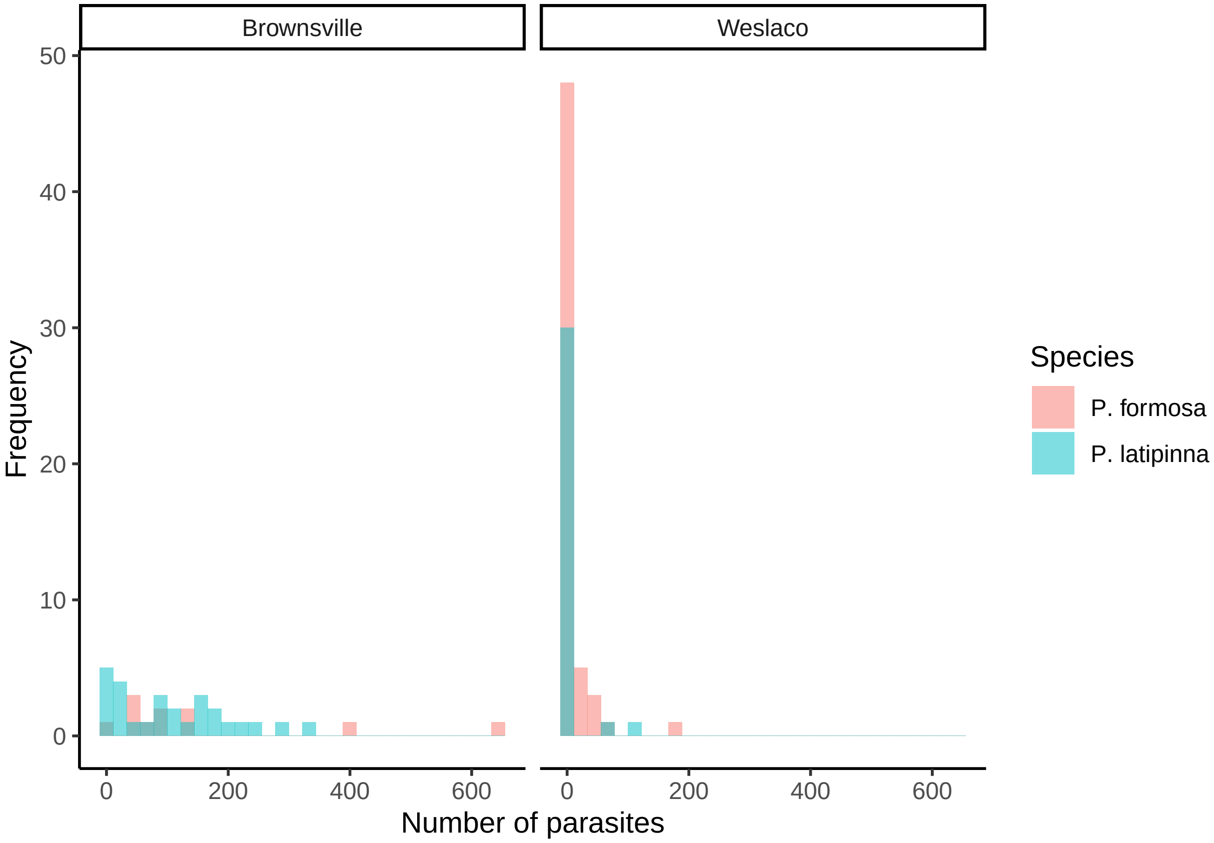
Figures

Parasites

Let’s create a figure that shows the difference in parasite loads between sites and species.

```r
para_site_hist <- parasite_data %>%
  ggplot(mapping = aes(totalpara, fill = species)) +
  geom_histogram(position = "identity", alpha = 0.5) +
  facet_wrap(vars(site.id)) +
  labs(x = "Number of parasites",
       y = "Frequency") +
  scale_fill_discrete(name = "Species") +
  theme_classic()
para_site_hist
```

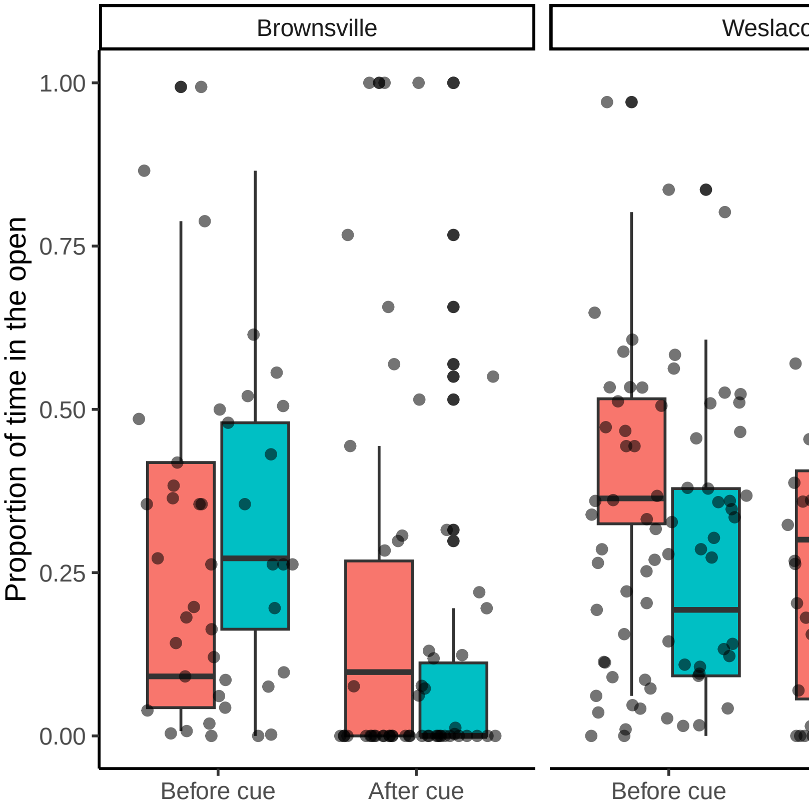
```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



Behavior

Let's look at the difference in average behavior before/after by site.

```r
beh_site_boxplot <- all_data_long %>%
  filter(prop.type != "prop.open") %>%
  ggplot(mapping = aes(
    x = prop.type,
    y = proportion,
    fill = species)) +
  geom_boxplot() +
  geom_jitter(aes(alpha = 0.25)) +
  facet_wrap(vars(site.id)) +
  labs(x = "",
       y = "Proportion of time in the open") +
  scale_fill_discrete(name = "Species") +
  scale_x_discrete(labels = c(
                     "prop.open.b4" = "Before cue",
                     "prop.open.after" = "After cue"
                   )) +
  scale_alpha(guide = "none") +
  theme_classic()
beh_site_boxplot
```

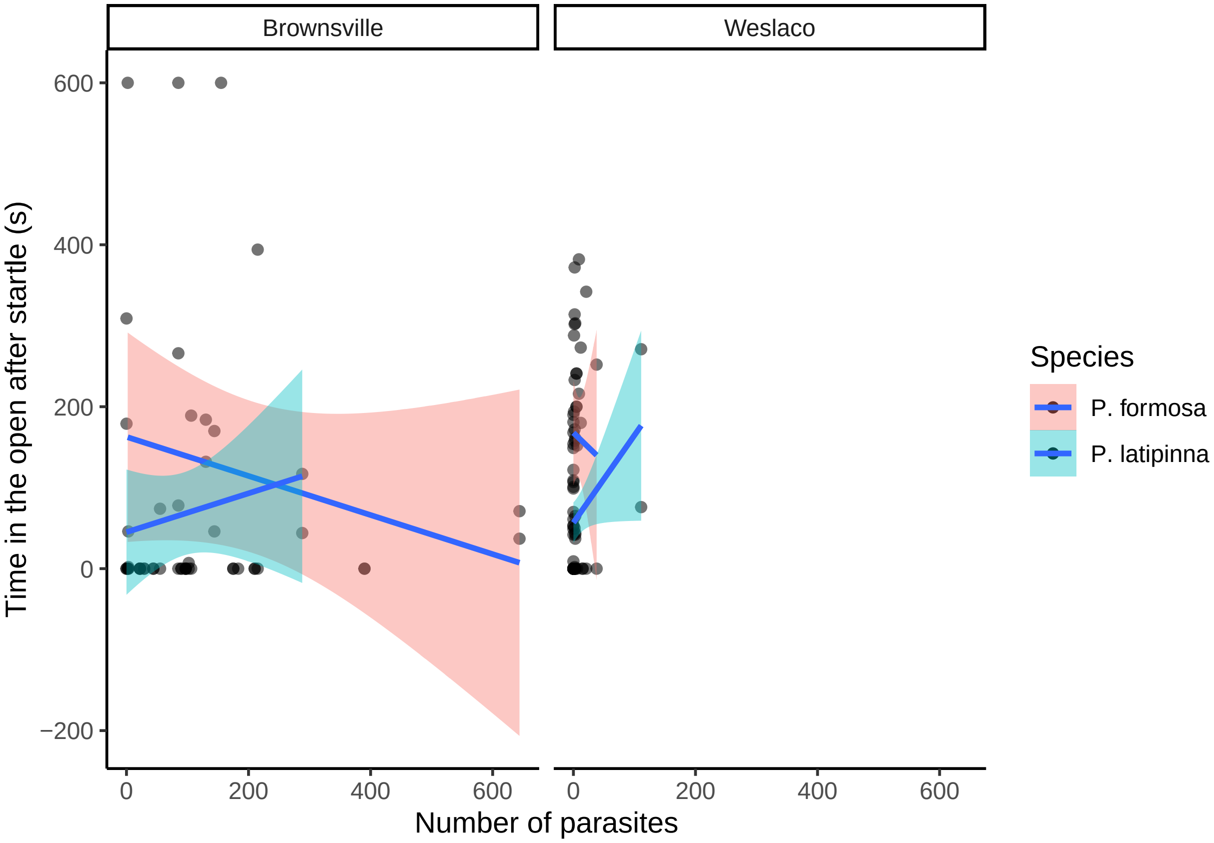


NOTE: need to update figure lables and legend.

Now, let's look at the relationship between parasite and species by site.

```
beh_para_site_lm <- all_data %>%
  ggplot(mapping = aes(
    x = totalpara,
    y = total.time.open.after,
    fill = species)) +
  geom_point(aes(alpha = 0.15)) +
  geom_smooth(method = 'lm') +
  facet_wrap(vars(site.id)) +
  labs(x = "Number of parasites",
       y = "Time in the open after startle (s)") +
  scale_fill_discrete(name = "Species") +
  scale_alpha(guide = "none") +
  theme_classic()
beh_para_site_lm
```

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

## Warning: Removed 14 rows containing non-finite values (`stat_smooth()`).

## Warning: Removed 14 rows containing missing values (`geom_point()`).
```

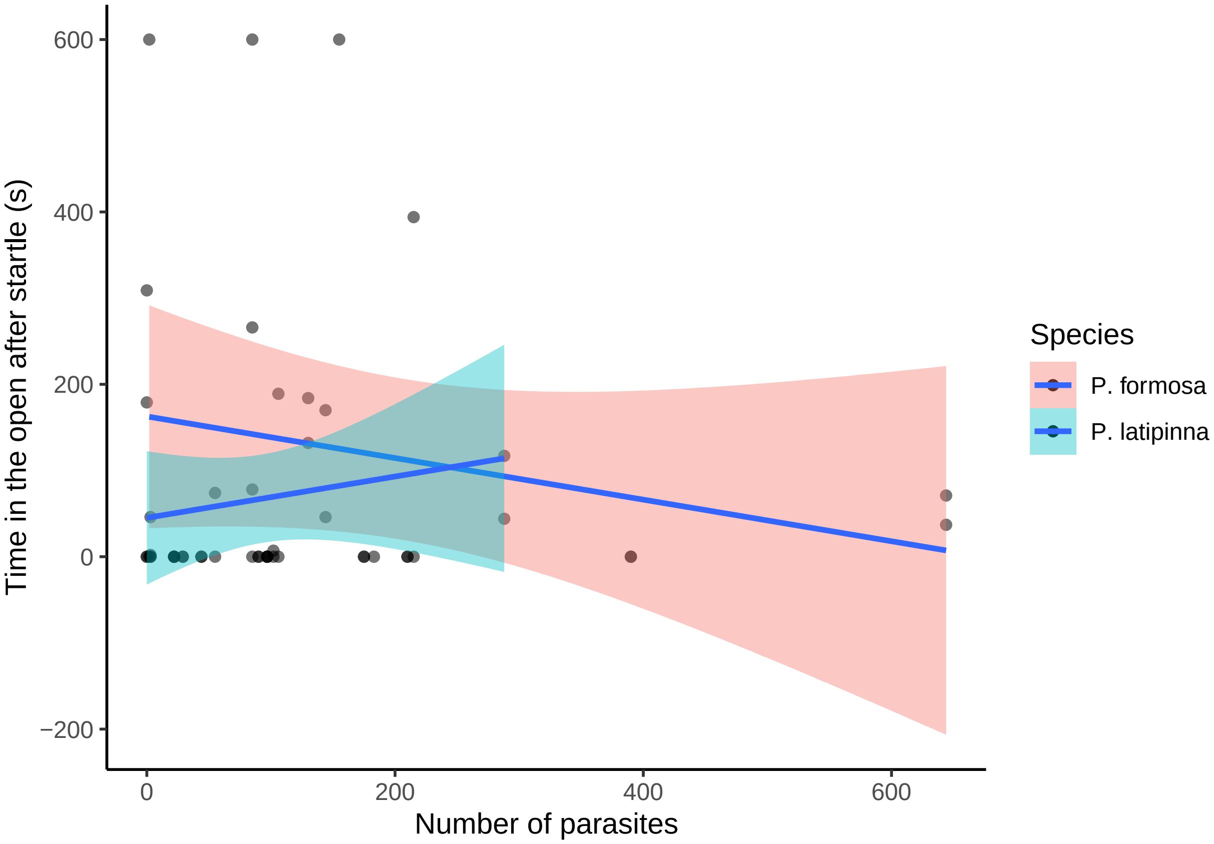


```
beh_para_site_br <- all_data %>%
  filter(site.id == "Brownsville") %>%
  ggplot(mapping = aes(
    x = totalpara,
    y = total.time.open.after,
    fill = species)) +
  geom_point(aes(alpha = 0.25)) +
  geom_smooth(method = 'lm') +
  labs(x = "Number of parasites",
       y = "Time in the open after startle (s)") +
  scale_fill_discrete(name = "Species") +
  scale_alpha(guide = "none") +
  theme_classic()
beh_para_site_br
```

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

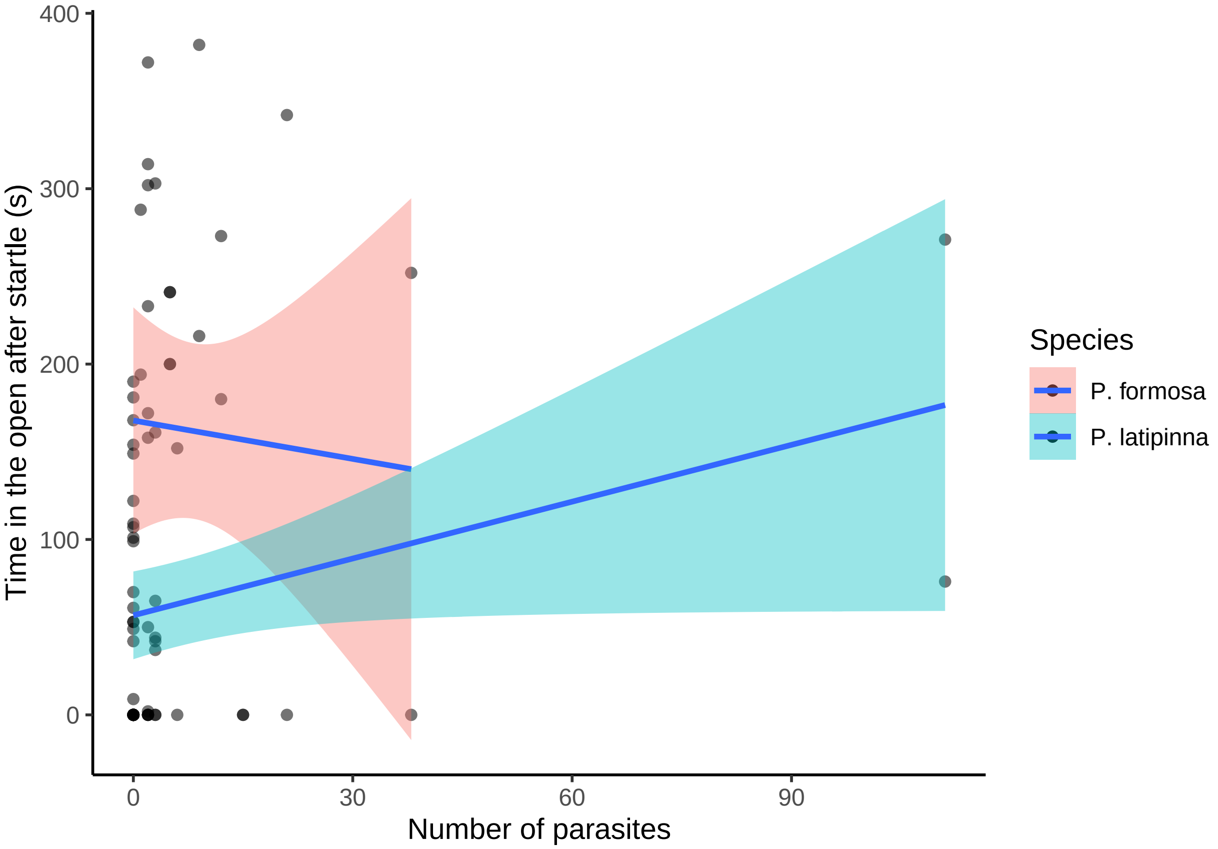
```
## Warning: Removed 4 rows containing non-finite values (`stat_smooth()`).
```

```
## Warning: Removed 4 rows containing missing values (`geom_point()`).
```



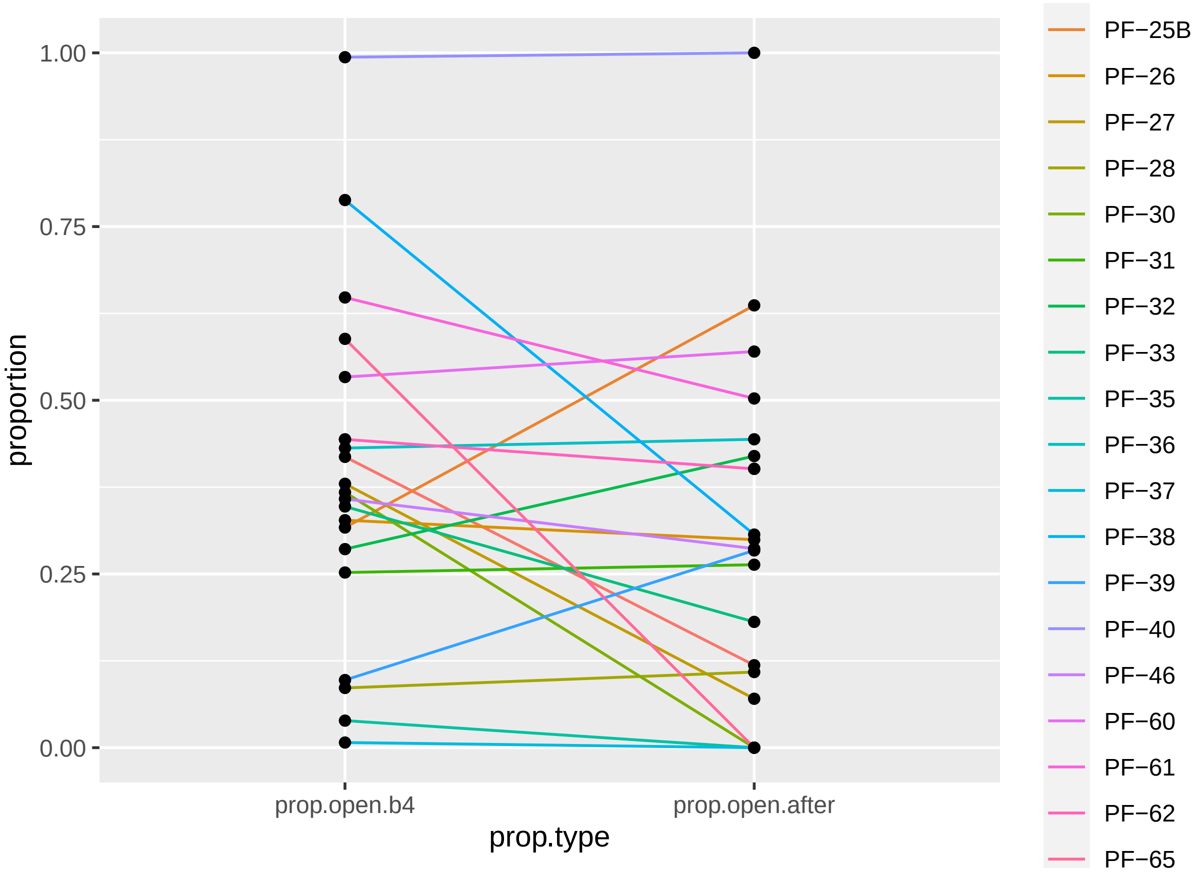
```
beh_para_site_wes <- all_data %>%
  filter(site.id == "Weslaco") %>%
  ggplot(mapping = aes(
    x = totalpara,
    y = total.time.open.after,
    fill = species)) +
  geom_point(aes(alpha = 0.25)) +
  geom_smooth(method = 'lm') +
  labs(x = "Number of parasites",
       y = "Time in the open after startle (s)") +
  scale_fill_discrete(name = "Species") +
  scale_alpha(guide = "none") +
  theme_classic()
beh_para_site_wes
```

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

## Warning: Removed 10 rows containing non-finite values (`stat_smooth()`).

## Warning: Removed 10 rows containing missing values (`geom_point()`).
```

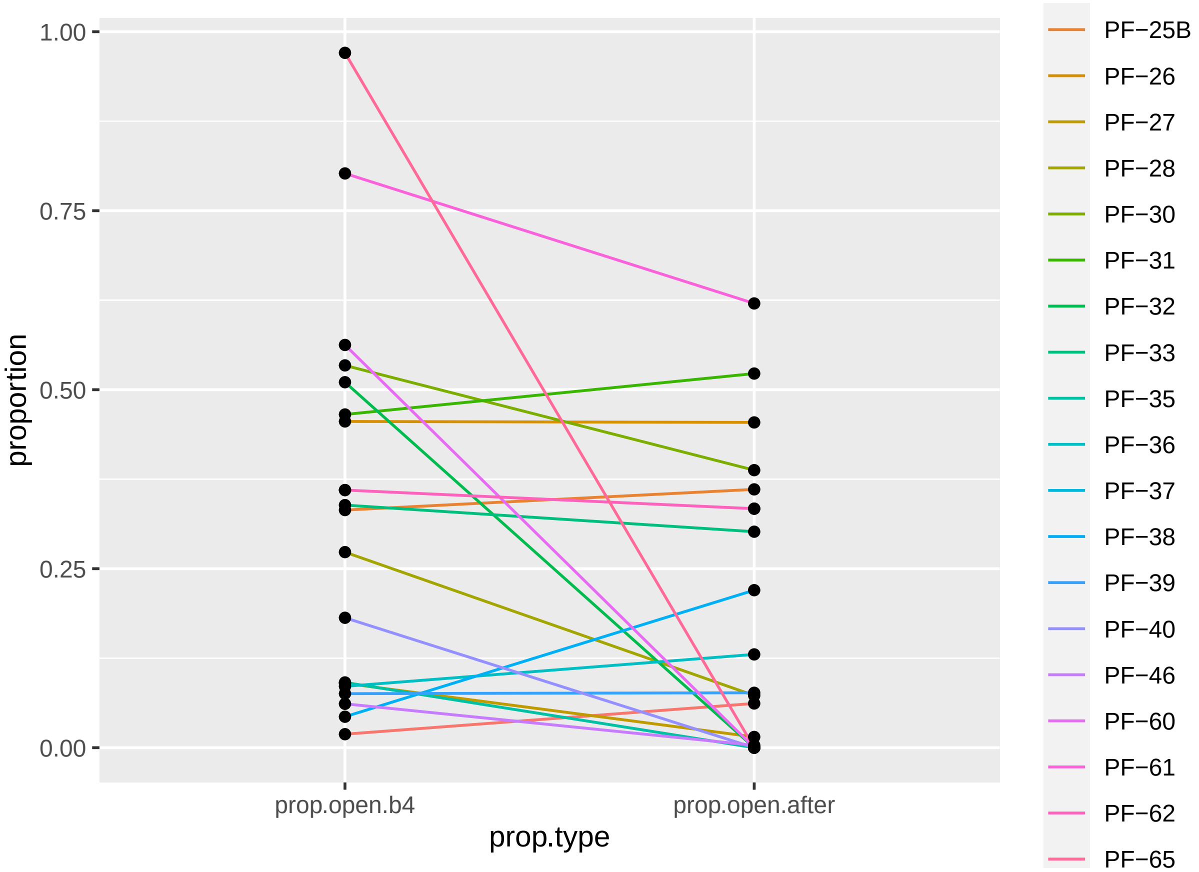


Let's also take a peek at the changes on an individual basis.

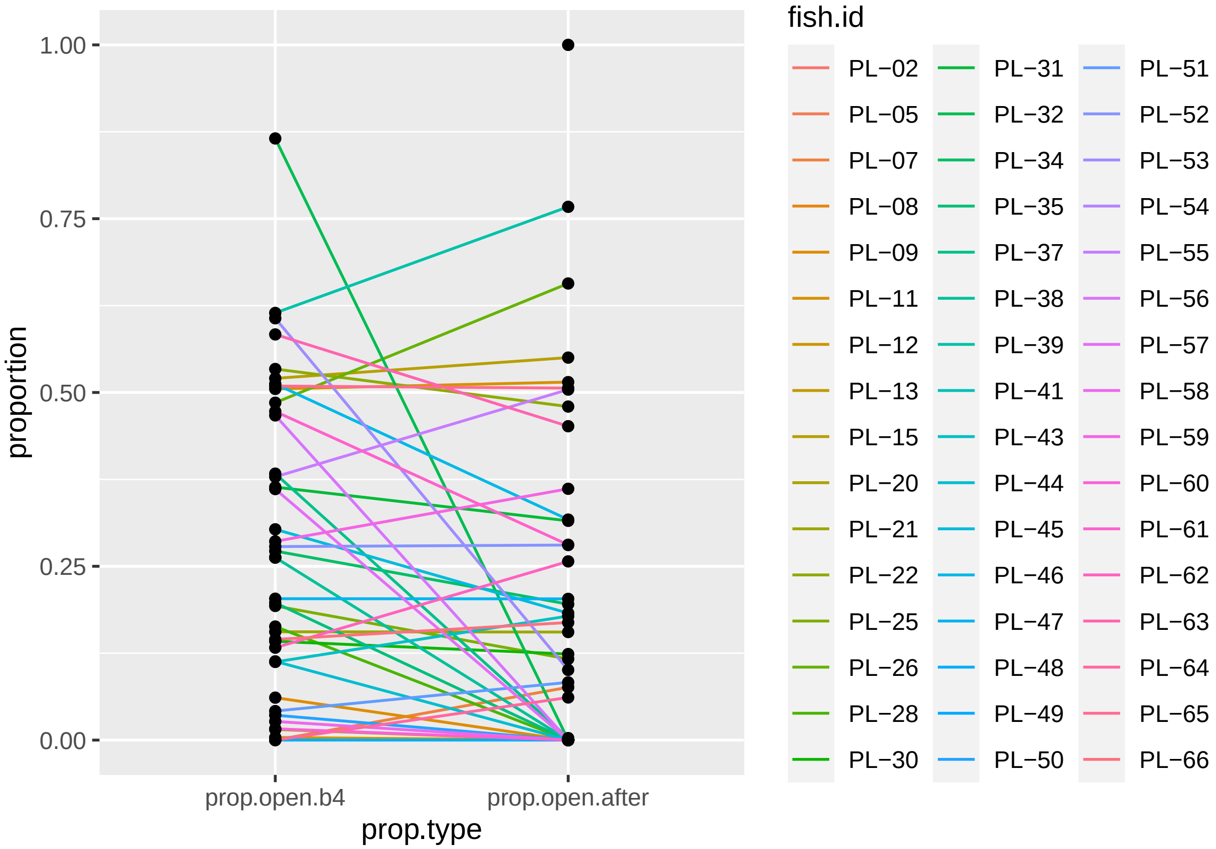
```
beh_indiv_pf1 <- all_data_long %>%
  filter(trial.id == "1") %>%
  filter(species == "P. formosa") %>%
  filter(prop.type == "prop.open.b4" | prop.type == "prop.open.after") %>%
  ggplot(mapping = aes(
    x = prop.type,
    y = proportion,
    group = fish.id)) +
  geom_line(aes(color = fish.id)) +
  geom_point()
beh_indiv_pf1
```



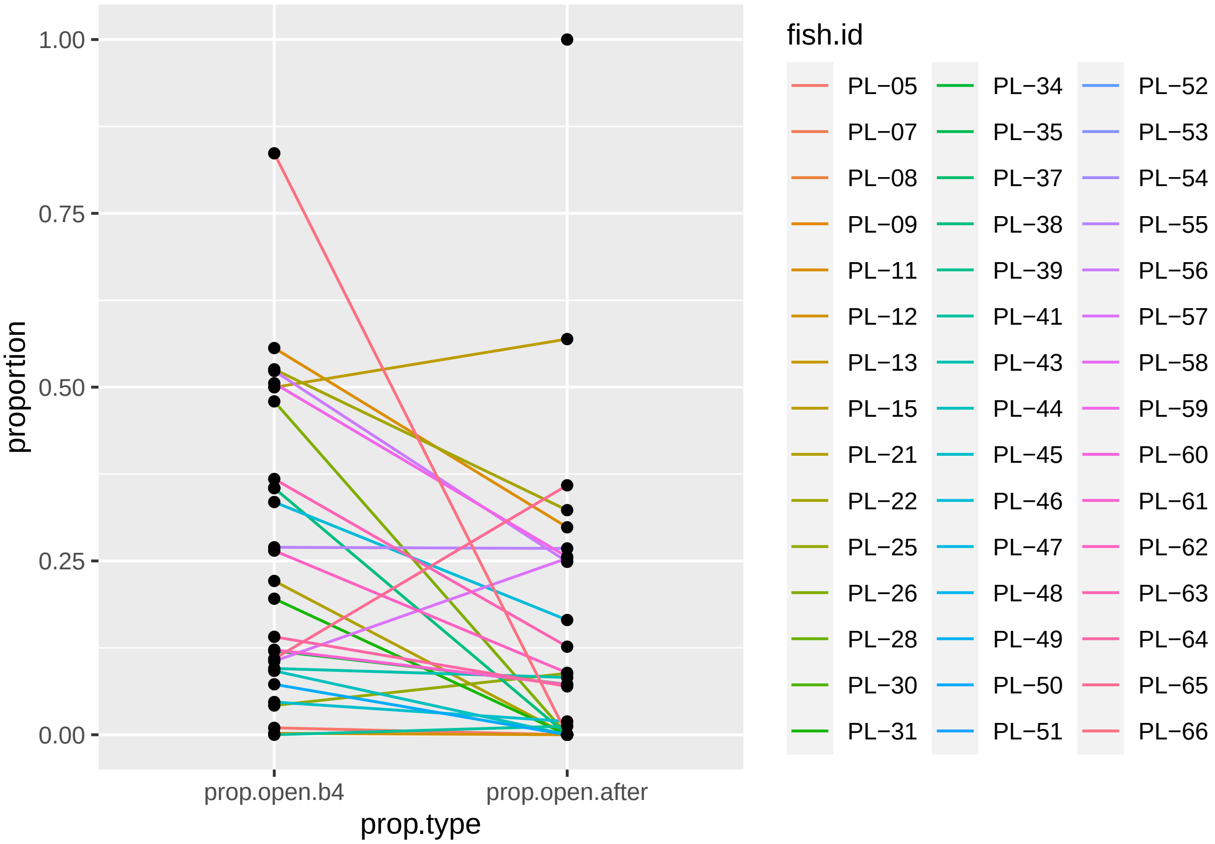
```
beh_indiv_pf2 <- all_data_long %>%
  filter(trial.id == "2") %>%
  filter(species == "P. formosa") %>%
  filter(prop.type == "prop.open.b4" | prop.type == "prop.open.after") %>%
  ggplot(mapping = aes(
    x = prop.type,
    y = proportion,
    group = fish.id)) +
  geom_line(aes(color = fish.id)) +
  geom_point()
beh_indiv_pf2
```



```
beh_indiv_pl1 <- all_data_long %>%
  filter(trial.id == "1") %>%
  filter(species == "P. latipinna") %>%
  filter(prop.type == "prop.open.b4" | prop.type == "prop.open.after") %>%
  ggplot(mapping = aes(
    x = prop.type,
    y = proportion,
    group = fish.id)) +
  geom_line(aes(color = fish.id)) +
  geom_point()
beh_indiv_pl1
```



```
beh_indiv_pl2 <- all_data_long %>%
  filter(trial.id == "2") %>%
  filter(species == "P. latipinna") %>%
  filter(prop.type == "prop.open.b4" | prop.type == "prop.open.after") %>%
  ggplot(mapping = aes(
    x = prop.type,
    y = proportion,
    group = fish.id)) +
  geom_line(aes(color = fish.id)) +
  geom_point()
beh_indiv_pl2
```


1.00
0.75
0.50
0.25
0.00
proportion
prop.open.b4
prop.open.after
prop.type
fish.id
PL−05
PL−07
PL−08
PL−09
PL−11
PL−12
PL−13
PL−15
PL−21
PL−22
PL−25
PL−26
PL−28
PL−30
PL−31
PL−34
PL−35
PL−37
PL−38
PL−39
PL−41
PL−43
PL−44
PL−45
PL−46
PL−47
PL−48
PL−49
PL−50
PL−51
PL−52
PL−53
PL−54
PL−55
PL−56
PL−57
PL−58
PL−59
PL−60
PL−61
PL−62
PL−63
PL−64
PL−65
PL−66