Density-dependence seedling mortality in Kadumane

Robert Bagchi and Ashwin Viswanathan

04 June 2024

Contents

set up

```
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.4     v readr     2.1.5
## v forcats   1.0.0     v stringr   1.5.1
## v ggplot2   3.5.1     v tibble    3.2.1
```

```
## v lubridate 1.9.3     v tidyr     1.3.1
## v purrr     1.0.2
## -- Conflicts ---------------------------------------- tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```
library(ggthemes)
library(knitr)

#library(vegan)
#library(grid)
#library(geosphere)
#library(mgcv)
library(glmmTMB)
library(DHARMa)
```

```
## This is DHARMa 0.4.6. For overview type '?DHARMa'. For recent changes, type news(package = 'DHARMa')
```

```
library(broom.mixed)
library(ggeffects)
library(sjPlot)
library(patchwork)
```

```
theme_set(theme_tufte())
```

Data

Load data

```
sp_codes <- read_csv("data/sp_codes.csv")
```

```
## Rows: 5 Columns: 4
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (4): code, genus, species, family
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
site_dat <- read_rds("data/kadumane_site_metadata.rds")
sdls <- read_rds("data/kadumane_seedlings.rds")
plot_dat <- read_rds("data/kadumane_plot_metadata.rds")
```

1. Join in site and plot meta-data.
2. Remove rows for species with no seedlings at start of the census in plot.
3. Select species that

- Are recorded in at least 5 plots.
- Vary in density among plots.
- Are groups of many (unidentified) species with the same code.

```
sdls <-  sdls |> left_join(site_dat) |> left_join(plot_dat) |>
  filter(census.start > 0)
```

```
## Joining with `by = join_by(site)`
## Joining with `by = join_by(site, location, group, plot)`
```

```
## find species with enough individuals and variation in density to allow
## analysis
sp_list <- group_by(sdls, species) |>
  summarise(abund = sum(census.start) ,  ## total abund
            n = sum(census.start > 0), ## number of plots withe species
            sd_dens = sd(census.start[census.start > 0])) |> ##var in density
  filter(n > 1) |> arrange(n)
## only lose 8 species by restricting to 5 or more occurrences (instead of 1)
## Seems reasonable. Also removing species that were unreliably identified.
##
## @Ashwin - what was SC and Artoh?
sp_list <- filter(sp_list, n > 4, !(species %in% c("Palm", "Artoh", "SC")),
                  sd_dens > 0)
sdls <- filter(sdls, species %in% sp_list$species)
dim(sdls) ## 968 columns
```

```
## [1] 968  19
```

Rename columns, add total seedling density, scale and centre data.

```
## shorten names
sdls <- rename(sdls,
               "trt_F" = "treatment.fungicide",
               "trt_I" = "treatment.insecticide",
               "Pr_m" = "proportion.mortality",
               "gr" = "group",
               "loc" = "location")
                ## group causes problems with some helper funcs
## add total density
tot_dens <- sdls |> group_by(site, loc, gr, plot) |>
  summarise(tot_dens = sum(census.start))
```

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

```
## add species mean density
## divide by total number of plots = 21 sites x 4 locs * 3 groups * 5 plots
sp_mean_dens <- sdls |> group_by(species) |>
  summarise(sp_mean_dens = sum(census.start)/(21*4*3*5),
            sp_mean_surv = sum(census.final)/sum(census.start))


sdls <- left_join(sdls, tot_dens, by = c("site", "loc", "gr", "plot")) |>
  left_join(select(sp_mean_dens, - sp_mean_surv))
```

```
## Joining with `by = join_by(species)`
```

```
#scale density by mean, fix couple of NAs and calculate log density
sdls <- mutate(sdls,
               slope.degrees = replace_na(slope.degrees, 5),
               Pr_s = 1 - Pr_m,
               con_dens = census.start,
               con_dens_s = con_dens/sp_mean_dens,
               slope.degrees_s = as.vector(scale(slope.degrees)),
                trt_F = factor(trt_F, labels = c("0", "F")),
                trt_I = factor(trt_I, labels = c("0", "I"))
               )
dim(sdls) ## 968 species x plot combinations
```

```
## [1] 968  25
```

```
sdls |> summarise(n_sdls = sum(census.start),
                  n_survs = sum(census.final),
                  n_species = n_distinct(species)) |> knitr::kable()
```

n_sdls	n_survs	n_species
6208	2680	26

Models

Raw conspecific density model

Fitting a model of mortality as a function of initial conspecific density, initial total density, biocide treatment and fragment area (and their interactions).

Also including slope of plot and random intercepts for plot, nested in location, nested in site. Random slopes and intercepts are included for each species to examine how density dependence varies among species.

Note that including the insecticide x fungicide interaction causes convergence problems and doesn't seem to be important either. Removing for practicality.

Model diagnostics

Take-homes

- Model diagnostics are generally good.
- Diagnostics are slightly better for non-logged version. Logged version has some evidence of quantiles deviating from expectations (removed logged version now).
- No evidence of trends between covariates and residuals.

The log density model has a slightly lower AIC. A little poking around (not shown) suggests that excluding the largest values reverses this order, so that the non-logged version is better.

Looks like the two models lead to roughly similar interpretation.

Inference

The fixed effects

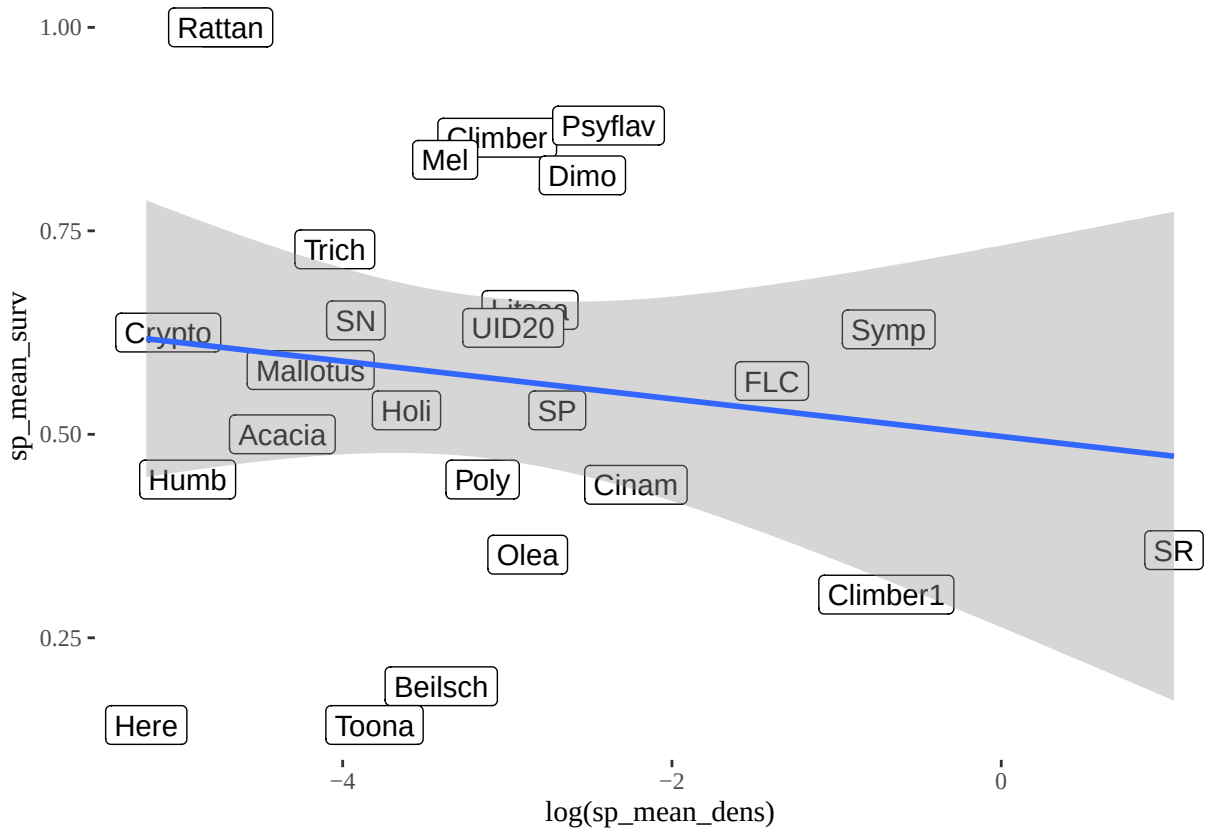
Main conclusions * Suppressing insects and fungi) increases survival. * Conspecific density reduces seedling survival. * The protection by fungicide is less effective at high conspecific density * There is a weak 3-way interaction between conspecific density, fragment size, and fungicide addition, so that the reduction in protection at high densities is more evident in larger fragments.

Scaled conspecific density models

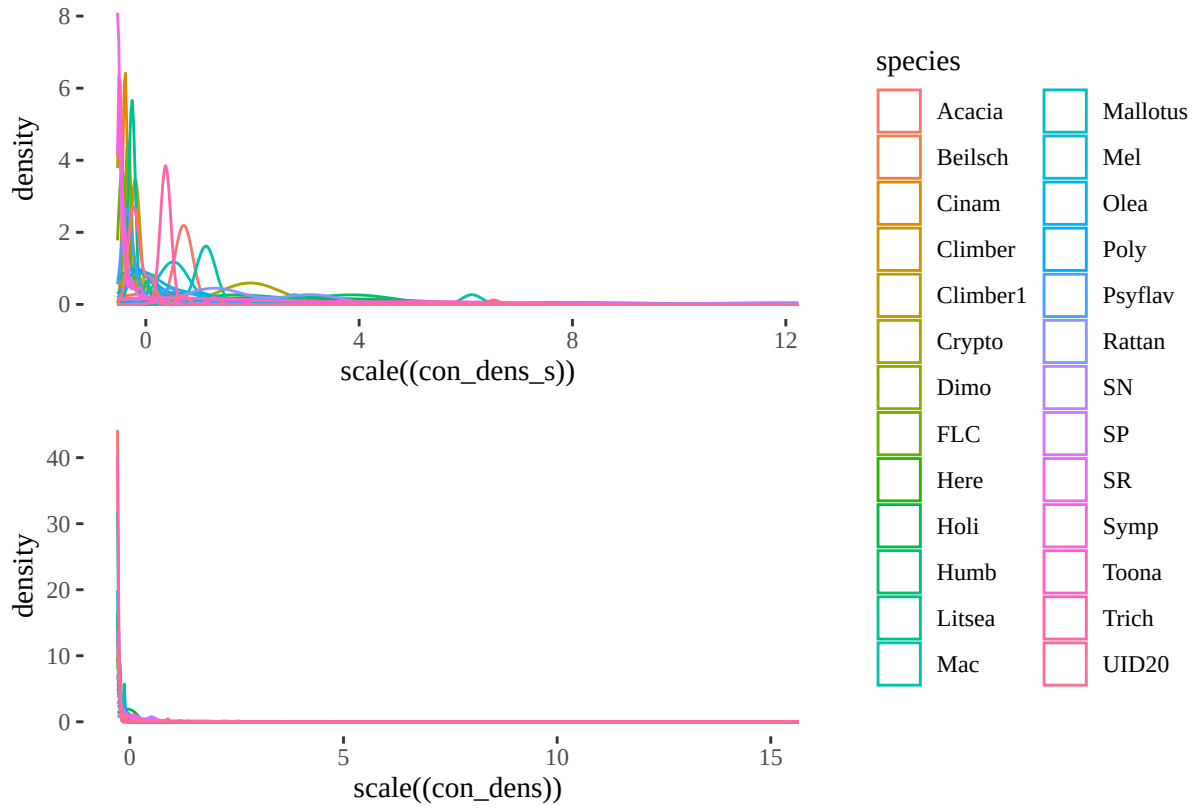
The most abundant species initially will often have lower survival (fecundity/ survival trade-off). This could generate what looks like a density-dependent relationship when looking across species, even without a relationship within species (i.e., Simpson's paradox).

```
ggplot(sp_mean_dens, aes(x = log(sp_mean_dens), y = sp_mean_surv)) +
  geom_label(aes(label= species)) + geom_smooth(method = "lm") + theme_tufte()
```

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




```
## not the clearest pattern, but worth accounting for.
(ggplot(sdls, aes(x = scale((con_dens_s)), colour = species)) + geom_density())/
(ggplot(sdls, aes(x = scale((con_dens)), colour = species)) + geom_density()) + plot_layout(guides = "c
```



To account for this, we can scale conspecific density by dividing by mean density and refitting the models.

```
## models using species-scaled density
##
## Random intercept model
m_cdd_s_ri <- glmmTMB(Pr_s ~ slope.degrees_s +
                        (scale(tot_dens) + scale(con_dens_s)) *
                        (trt_I + trt_F) *
                        scale(log(fragment.size)) +
                        (1|species) +
                        (1|site/loc/gr/plot),
                      weights = census.start, data = sdls,
                      family=binomial)
## Random intercept and slope model for species specific effects
m_cdd_s_ris <- glmmTMB(Pr_s ~ slope.degrees_s +
                        (scale(tot_dens) + scale(con_dens_s)) *
                        (trt_I + trt_F) * scale(log(fragment.size)) *
                        (scale(con_dens_s) + scale(tot_dens)||species) +
                        ## setting cor to 0 to converge
                        (1|site/loc/gr/plot),
                      weights = census.start, data = sdls,
                      family=binomial)
anova(m_cdd_s_ri, m_cdd_s_ris) ## random slope *much* better
```

```
## Data: sdls
## Models:
```

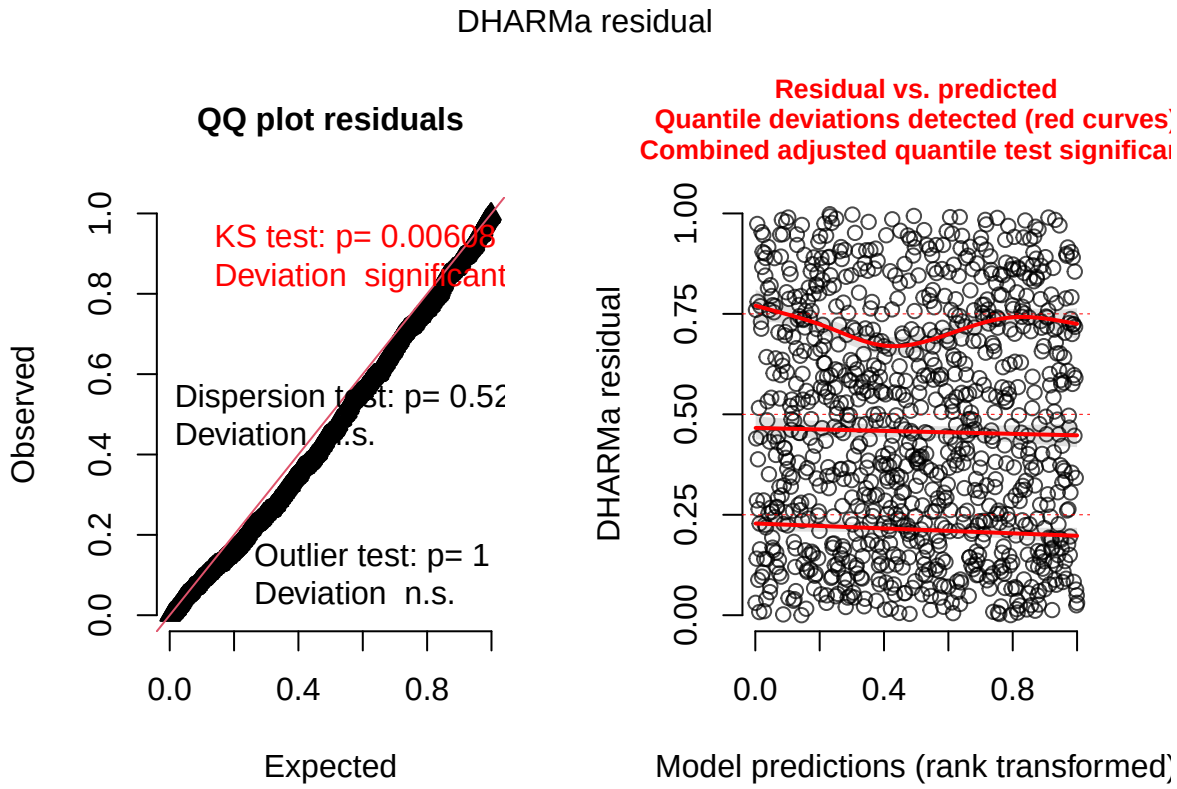
```
## m_cdd_s_ri: Pr_s ~ slope.degrees_s + (scale(tot_dens) + scale(con_dens_s)) * , zi=~0, disp=~1
## m_cdd_s_ri:     (trt_I + trt_F) * scale(log(fragment.size)) + (1 | species) + , zi=~0, disp=~1
## m_cdd_s_ri:     (1 | site/loc/gr/plot), zi=~0, disp=~1
## m_cdd_s_ris: Pr_s ~ slope.degrees_s + (scale(tot_dens) + scale(con_dens_s)) * , zi=~0, disp=~1
## m_cdd_s_ris:     (trt_I + trt_F) * scale(log(fragment.size)) * (scale(con_dens_s) + , zi=~0, disp=~1
## m_cdd_s_ris:     scale(tot_dens) || species) + (1 | site/loc/gr/plot), zi=~0, disp=~1
##             Df    AIC    BIC  logLik deviance  Chisq Chi Df Pr(>Chisq)
## m_cdd_s_ri  24 2197.1 2314.1 -1074.6   2149.1
## m_cdd_s_ris 26 2187.2 2313.9 -1067.6   2135.2 13.954      2  0.0009331 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Models fit without issues as long as we fix the covariance between random slopes and intercepts to 0.

The improvment with the random slopes model suggests we need to look at individual species.

Diagnostics

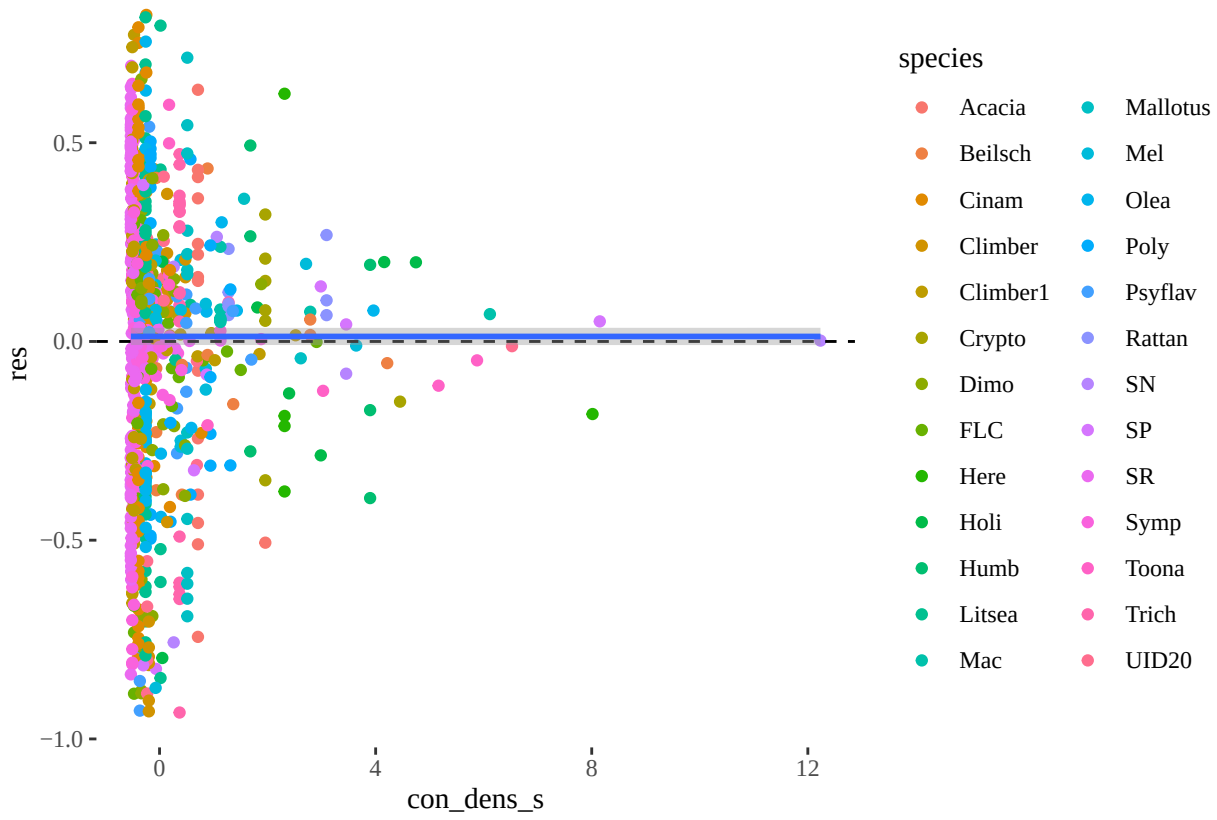
```
res_s <- simulateResiduals(m_cdd_s_ri)
plot(res_s) ## ok - some deviation from ideal residual distribution, but
```




```
## acceptable.

## look at relationship with covariates
```

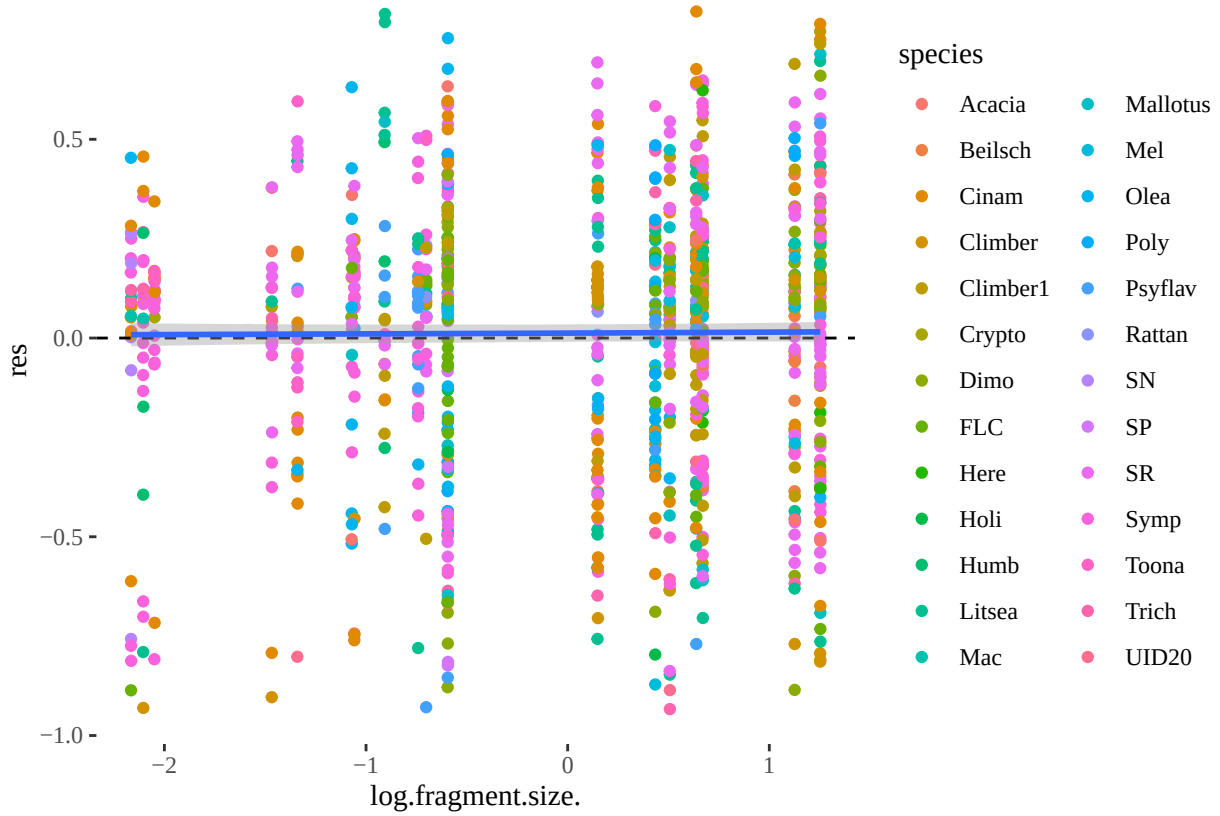
```
diag_dat <- data.frame(m_cdd_s_ri$frame, res = res_s$fittedResiduals)

diag_dat <-   rename_with(diag_dat, ~ str_replace(.x, "scale\\.", "")) |>
  rename_with(~str_replace(.x, "\\.$", ""))

ggplot(diag_dat, aes(x = con_dens_s, y = res)) +
  geom_point(aes(colour = species)) +
  geom_hline(yintercept=0, linetype = "dashed") +
  geom_smooth(method="gam") ## no trend.
```

```
## `geom_smooth()` using formula = 'y ~ s(x, bs = "cs")'
```



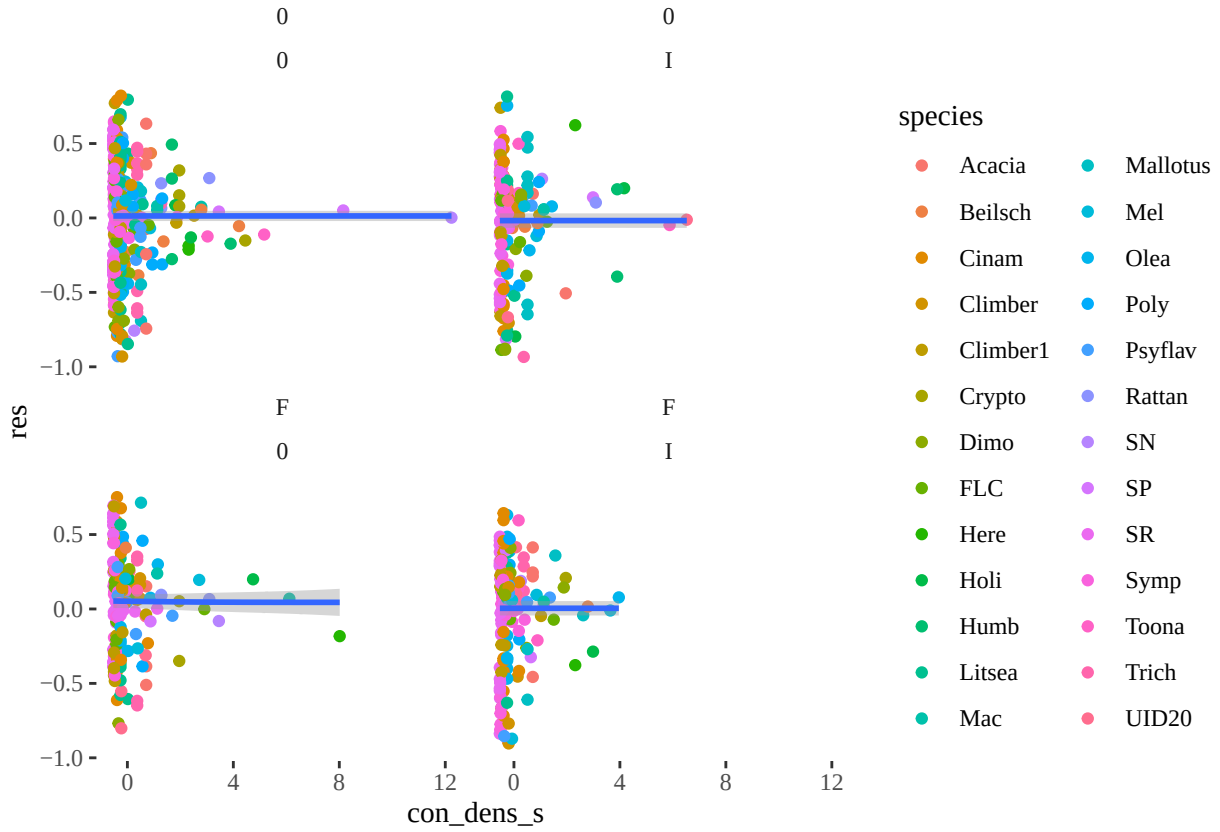

```
ggplot(diag_dat, aes(x = log.fragment.size., y = res)) +
  geom_point(aes(colour = species), position = "jitter") +
  geom_hline(yintercept=0, linetype = "dashed") +
  geom_smooth(method="gam") ## no trend.
```

```
## `geom_smooth()` using formula = 'y ~ s(x, bs = "cs")'
```




```
ggplot(diag_dat, aes(x = con_dens_s, y = res)) +
  facet_wrap(~trt_F + trt_I ) +
  geom_point(aes(colour = species)) +
  geom_smooth(method="gam") ## no trend with treatment
```

```
## `geom_smooth()` using formula = 'y ~ s(x, bs = "cs")'
```



The diagnostics aren't as perfect in the scaled case, but the scaled version does seem to be sensible given the massive differences in density.

Model inference

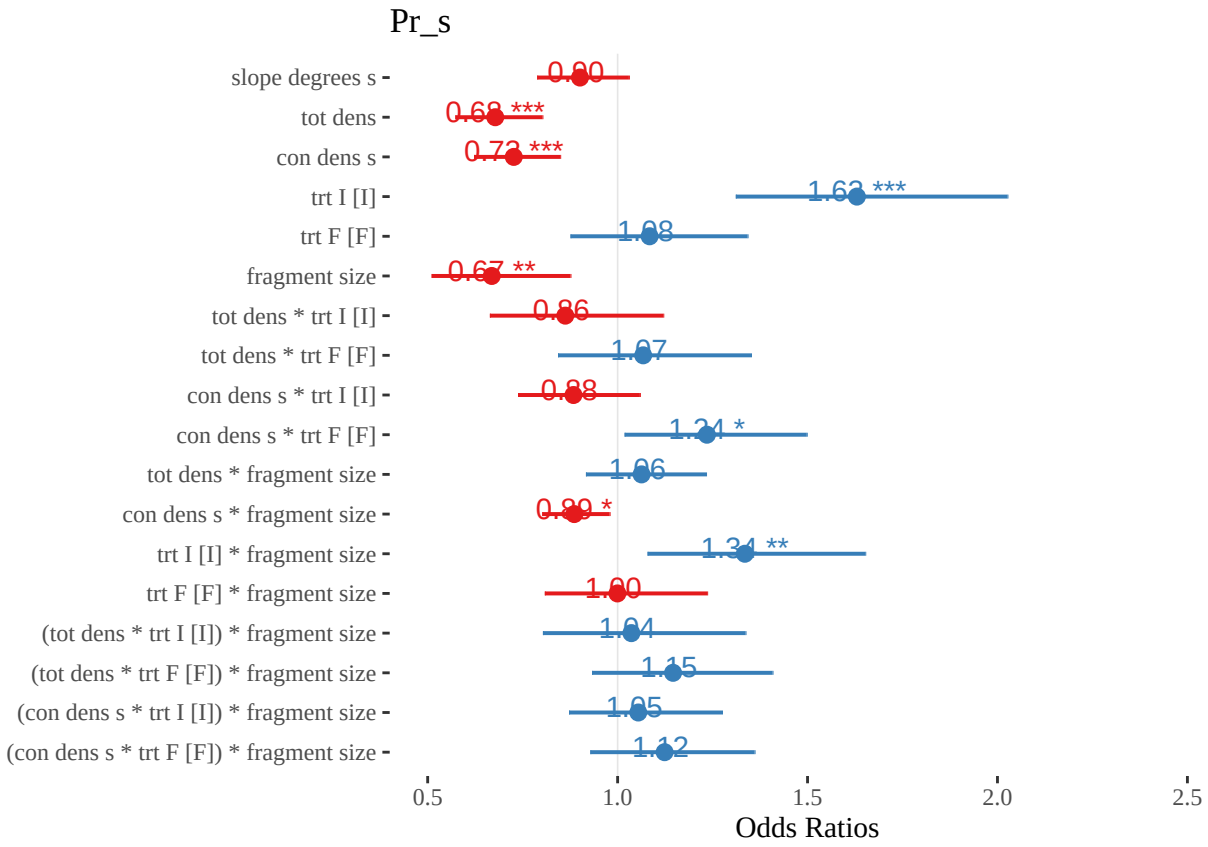
```
summary(m_cdd_s_ri)
```

```
##  Family: binomial  ( logit )
## Formula:
## Pr_s ~ slope.degrees_s + (scale(tot_dens) + scale(con_dens_s)) *
##     (trt_I + trt_F) * scale(log(fragment.size)) + (1 | species) +
##     (1 | site/loc/gr/plot)
## Data: sdls
## Weights: census.start
##
##      AIC      BIC   logLik deviance df.resid
##   2197.1   2314.1  -1074.6   2149.1      944
##
## Random effects:
##
## Conditional model:
##  Groups           Name        Variance  Std.Dev.
##  species          (Intercept) 1.355e+00 1.1642158
##  plot:gr:loc:site (Intercept) 2.340e-01 0.4836923
```

```
##  gr:loc:site      (Intercept) 1.692e-01 0.4113449
##  loc:site         (Intercept) 4.213e-01 0.6490432
##  site             (Intercept) 3.512e-08 0.0001874
## Number of obs: 968, groups:
## species, 26; plot:gr:loc:site, 474; gr:loc:site, 110; loc:site, 37; site, 21
##
## Conditional model:
##                                                    Estimate Std. Error
## (Intercept)                                       0.6755043  0.2920884
## slope.degrees_s                                  -0.1042895  0.0681512
## scale(tot_dens)                                  -0.3887989  0.0858523
## scale(con_dens_s)                                -0.3193334  0.0792852
## trt_II                                            0.4885509  0.1110445
## trt_FF                                            0.0809017  0.1088436
## scale(log(fragment.size))                        -0.4024701  0.1377352
## scale(tot_dens):trt_II                           -0.1481742  0.1335183
## scale(tot_dens):trt_FF                            0.0651851  0.1202666
## scale(con_dens_s):trt_II                         -0.1235055  0.0920978
## scale(con_dens_s):trt_FF                          0.2111937  0.0984225
## scale(tot_dens):scale(log(fragment.size))         0.0611248  0.0755813
## scale(con_dens_s):scale(log(fragment.size))      -0.1211638  0.0509800
## trt_II:scale(log(fragment.size))                  0.2889510  0.1086788
## trt_FF:scale(log(fragment.size))                 -0.0005297  0.1082614
## scale(tot_dens):trt_II:scale(log(fragment.size))  0.0356476  0.1298870
## scale(tot_dens):trt_FF:scale(log(fragment.size))  0.1362907  0.1049890
## scale(con_dens_s):trt_II:scale(log(fragment.size))  0.0529964  0.0968246
## scale(con_dens_s):trt_FF:scale(log(fragment.size))  0.1166648  0.0975779
##                                                   z value Pr(>|z|)
## (Intercept)                                         2.313  0.02074 *
## slope.degrees_s                                    -1.530  0.12595
## scale(tot_dens)                                    -4.529 5.93e-06 ***
## scale(con_dens_s)                                  -4.028 5.63e-05 ***
## trt_II                                              4.400 1.08e-05 ***
## trt_FF                                              0.743  0.45731
## scale(log(fragment.size))                          -2.922  0.00348 **
## scale(tot_dens):trt_II                             -1.110  0.26710
## scale(tot_dens):trt_FF                              0.542  0.58782
## scale(con_dens_s):trt_II                           -1.341  0.17991
## scale(con_dens_s):trt_FF                            2.146  0.03189 *
## scale(tot_dens):scale(log(fragment.size))           0.809  0.41867
## scale(con_dens_s):scale(log(fragment.size))        -2.377  0.01747 *
## trt_II:scale(log(fragment.size))                    2.659  0.00784 **
## trt_FF:scale(log(fragment.size))                   -0.005  0.99610
## scale(tot_dens):trt_II:scale(log(fragment.size))    0.274  0.78374
## scale(tot_dens):trt_FF:scale(log(fragment.size))    1.298  0.19424
## scale(con_dens_s):trt_II:scale(log(fragment.size))  0.547  0.58414
## scale(con_dens_s):trt_FF:scale(log(fragment.size))  1.196  0.23185
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
plot_model(m_cdd_s_ri, show.values=TRUE) + ylim(c(0.5, 2.5))
```

```
## Scale for y is already present.
## Adding another scale for y, which will replace the existing scale.
```




```
car::Anova(m_cdd_s_ri)
```

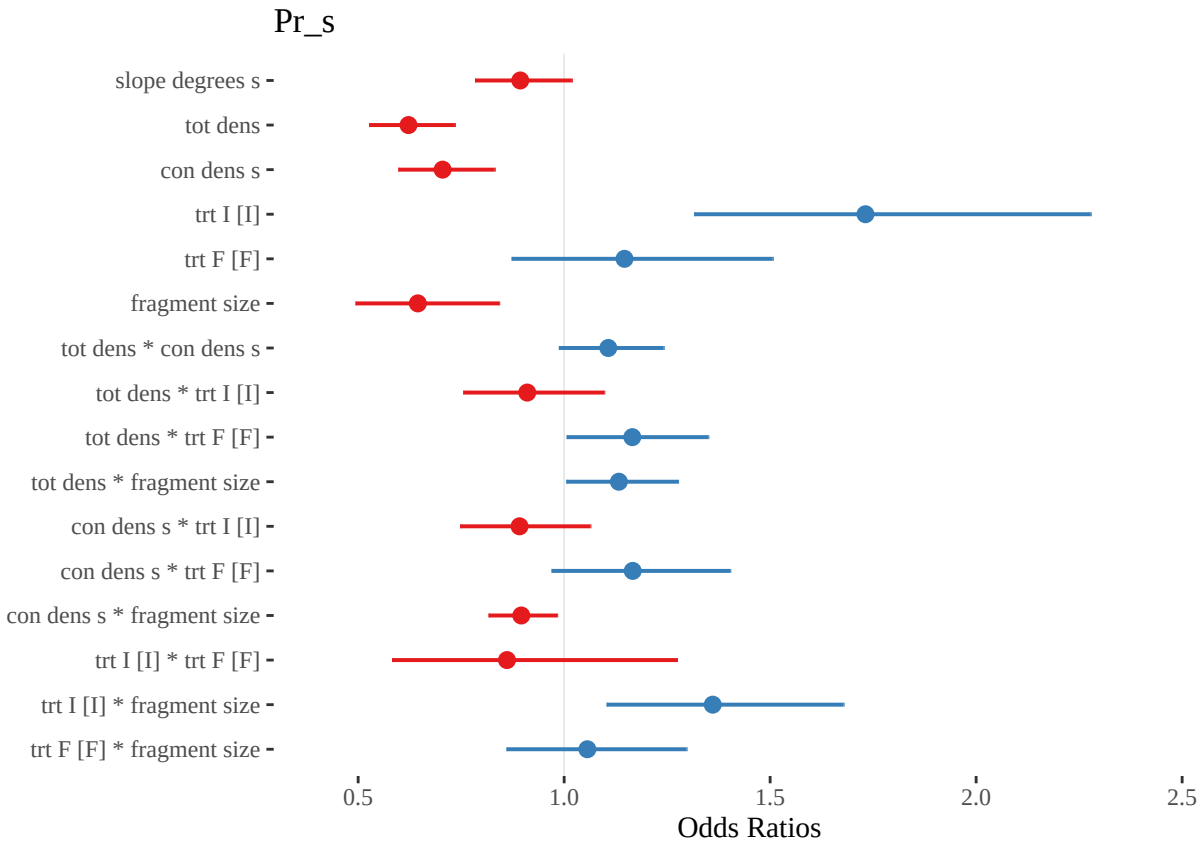
```
## Analysis of Deviance Table (Type II Wald chisquare tests)
##
## Response: Pr_s
##                                                    Chisq Df Pr(>Chisq)
## slope.degrees_s                                   2.3417  1   0.125951
## scale(tot_dens)                                  23.3742  1  1.334e-06 ***
## scale(con_dens_s)                                17.6388  1  2.671e-05 ***
## trt_I                                            17.0703  1  3.602e-05 ***
## trt_F                                             3.8922  1   0.048510 *
## scale(log(fragment.size))                         5.2401  1   0.022072 *
## scale(tot_dens):trt_I                             1.7183  1   0.189916
## scale(tot_dens):trt_F                             6.5759  1   0.010337 *
## scale(con_dens_s):trt_I                           2.4813  1   0.115208
## scale(con_dens_s):trt_F                           3.5771  1   0.058580 .
## scale(tot_dens):scale(log(fragment.size))         3.8857  1   0.048700 *
## scale(con_dens_s):scale(log(fragment.size))       4.4018  1   0.035901 *
## trt_I:scale(log(fragment.size))                   7.4137  1   0.006473 **
## trt_F:scale(log(fragment.size))                   0.0772  1   0.781091
## scale(tot_dens):trt_I:scale(log(fragment.size))   0.0753  1   0.783738
## scale(tot_dens):trt_F:scale(log(fragment.size))   1.6852  1   0.194238
## scale(con_dens_s):trt_I:scale(log(fragment.size)) 0.2996  1   0.584142
## scale(con_dens_s):trt_F:scale(log(fragment.size)) 1.4295  1   0.231850
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
car::Anova(m_cdd_s_ris)
```

```
## Analysis of Deviance Table (Type II Wald chisquare tests)
##
## Response: Pr_s
##                                                 Chisq Df Pr(>Chisq)
## slope.degrees_s                                2.8454  1  0.0916364 .
## scale(tot_dens)                                0.5362  1  0.4640207
## scale(con_dens_s)                             11.0659  1  0.0008793 ***
## trt_I                                         17.7010  1  2.585e-05 ***
## trt_F                                          3.5593  1  0.0592141 .
## scale(log(fragment.size))                      5.5929  1  0.0180337 *
## scale(tot_dens):trt_I                          1.8384  1  0.1751423
## scale(tot_dens):trt_F                          7.6943  1  0.0055396 **
## scale(con_dens_s):trt_I                        0.6202  1  0.4309704
## scale(con_dens_s):trt_F                        2.2386  1  0.1346005
## scale(tot_dens):scale(log(fragment.size))      4.2905  1  0.0383262 *
## scale(con_dens_s):scale(log(fragment.size))    0.2213  1  0.6380294
## trt_I:scale(log(fragment.size))                8.1304  1  0.0043529 **
## trt_F:scale(log(fragment.size))                0.0448  1  0.8324460
## scale(tot_dens):trt_I:scale(log(fragment.size))  0.0813  1  0.7755017
## scale(tot_dens):trt_F:scale(log(fragment.size))  0.9376  1  0.3328944
## scale(con_dens_s):trt_I:scale(log(fragment.size))  0.3644  1  0.5460845
## scale(con_dens_s):trt_F:scale(log(fragment.size))  1.9890  1  0.1584465
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
## Given lack of evidence for 3-way interactions, perhaps try only 2 way?
m_cdd_s_ri_2way <- glmmTMB(Pr_s ~ slope.degrees_s +
                       ((scale(tot_dens) + scale(con_dens_s)) +
                       (trt_I + trt_F) +
                       scale(log(fragment.size)))^2 +
                       (1|species) +
                       (1|site/loc/gr/plot),
                     weights = census.start, data = sdls,
                     family=binomial)
plot_model(m_cdd_s_ri_2way) + ylim(c(0.4, 2.5))
```

```
## Scale for y is already present.
## Adding another scale for y, which will replace the existing scale.
```



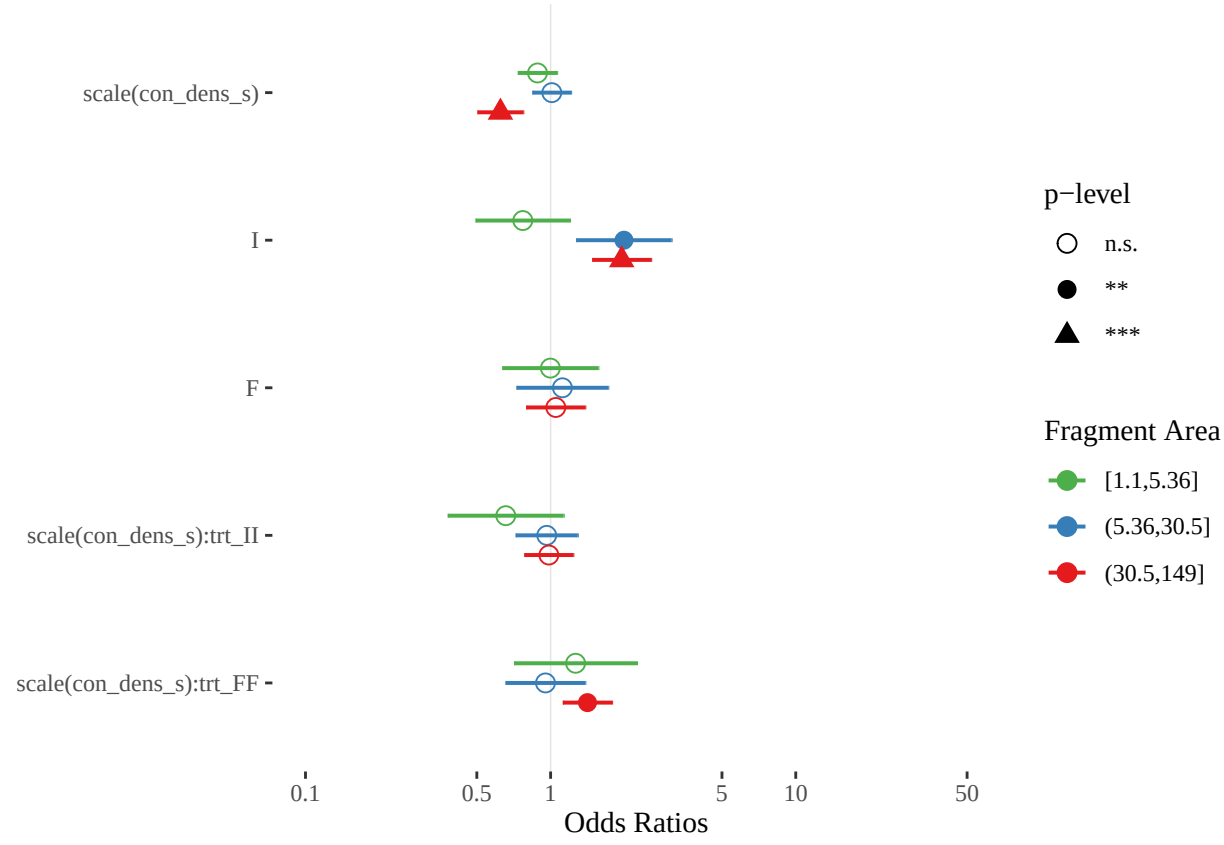
Take-homes:

- Survival is negatively conspecific density dependent
- Fungicide reduces the density dependence, but the evidence is complex. In the full model with a 3 way CDD:frag_area:F interaction, the 2-way cdd:F interaction is significant (P = 0.03). Note that the magnitude of the effect (0.21) is 2/3rds of the overall conspecific density effect (-0.32). Worth remarking on in the results. However, the Type 2 anova suggests that the effect is conditional on the effect of fragment area - excluding the interaction weakens evidence for the 2-way interaction (P = 0.059)
- CDD strengthens significantly with fragment area (P = 0.017).
- The interaction between fungicide, fragment area and CDD is not significant, but its estimated magnitude (0.117) is almost equal (and opposite direction) to the interaction between fragment area and density (-0.121).
- Worth noting that survival also increases with total stem density, with a similar magnitude as with conspecific density.
- Survival also declines with fragment area.
- Insecticide increases survival independent of density, and its effect increases with fragment area.

Models split by categorical fragment size

The effect of fragment

```
# Using 3 categories
sdls <- mutate(sdls,
               frag_sizeclass3 = cut(fragment.size,
                                     quantile(site_dat$fragment.size,
                                              c(0, 0.33, 0.66, 1)),
                                     include.lowest=TRUE))

m_cdd_s_ri_frag <- lapply(split(sdls, f= sdls$frag_sizeclass3),
                          function(d){
                            glmmTMB(Pr_s ~ slope.degrees_s +
                                      (scale(tot_dens) + scale(con_dens_s)) *
                                      (trt_I + trt_F) +
                                      (1|species) +
                                      ## setting cor to 0 to converge
                                      (1|site/loc/gr/plot),
                                    weights = census.start, data = d,
                                    family=binomial)})
term_nms <- names(fixef(m_cdd_s_ri_frag[[1]])$cond)

plot_models(m_cdd_s_ri_frag,
            rm.terms=c("slope.degrees_s",
            term_nms[str_detect(term_nms, "tot")]), ## declutter
            m.labels=names(m_cdd_s_ri_frag), p.shape=TRUE) +
  labs(colour = "Fragment Area")
```



```
map(m_cdd_s_ri_frag, summary)
```

```
## $`[1.1,5.36]`
##  Family: binomial  ( logit )
## Formula:
## Pr_s ~ slope.degrees_s + (scale(tot_dens) + scale(con_dens_s)) *
##     (trt_I + trt_F) + (1 | species) + (1 | site/loc/gr/plot)
## Data: d
## Weights: census.start
##
##      AIC      BIC   logLik deviance df.resid
##    376.7    421.8   -173.4    346.7      134
##
## Random effects:
##
## Conditional model:
##  Groups           Name        Variance  Std.Dev.
##  species          (Intercept) 2.674e-01 5.171e-01
##  plot:gr:loc:site (Intercept) 3.592e-09 5.994e-05
##  gr:loc:site      (Intercept) 9.554e-02 3.091e-01
##  loc:site         (Intercept) 3.052e-01 5.524e-01
##  site             (Intercept) 3.052e-01 5.524e-01
## Number of obs: 149, groups:
## species, 18; plot:gr:loc:site, 87; gr:loc:site, 20; loc:site, 7; site, 7
##
## Conditional model:
##                            Estimate Std. Error z value Pr(>|z|)
## (Intercept)                1.026283   0.417397   2.459 0.013941 *
## slope.degrees_s           -0.147994   0.139908  -1.058 0.290148
## scale(tot_dens)           -0.294738   0.086247  -3.417 0.000632 ***
## scale(con_dens_s)         -0.123143   0.094292  -1.306 0.191562
## trt_II                    -0.261125   0.226313  -1.154 0.248573
## trt_FF                    -0.002225   0.230126  -0.010 0.992284
## scale(tot_dens):trt_II     0.452427   0.181515   2.492 0.012685 *
## scale(tot_dens):trt_FF    -0.121468   0.138459  -0.877 0.380331
## scale(con_dens_s):trt_II -0.421473   0.277932  -1.516 0.129404
## scale(con_dens_s):trt_FF  0.234629   0.294453   0.797 0.425549
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## $`(5.36,30.5]`
##  Family: binomial  ( logit )
## Formula:
## Pr_s ~ slope.degrees_s + (scale(tot_dens) + scale(con_dens_s)) *
##     (trt_I + trt_F) + (1 | species) + (1 | site/loc/gr/plot)
## Data: d
## Weights: census.start
##
##      AIC      BIC   logLik deviance df.resid
##    629.2    683.9   -299.6    599.2      269
##
## Random effects:
##
```

```
## Conditional model:
##  Groups           Name        Variance  Std.Dev.
##  species          (Intercept) 1.497e+00 1.223e+00
##  plot:gr:loc:site (Intercept) 2.744e-01 5.238e-01
##  gr:loc:site      (Intercept) 7.970e-09 8.928e-05
##  loc:site         (Intercept) 1.955e-01 4.421e-01
##  site             (Intercept) 3.622e-01 6.018e-01
## Number of obs: 284, groups:
## species, 21; plot:gr:loc:site, 124; gr:loc:site, 27; loc:site, 9; site, 7
##
## Conditional model:
##                           Estimate Std. Error z value Pr(>|z|)
## (Intercept)                0.50437    0.45066   1.119  0.26307
## slope.degrees_s           -0.21107    0.11866  -1.779  0.07527 .
## scale(tot_dens)           -0.41809    0.13905  -3.007  0.00264 **
## scale(con_dens_s)          0.01107    0.09278   0.119  0.90501
## trt_II                     0.68891    0.22859   3.014  0.00258 **
## trt_FF                     0.11082    0.21994   0.504  0.61435
## scale(tot_dens):trt_II    -0.63928    0.21656  -2.952  0.00316 **
## scale(tot_dens):trt_FF    -0.02168    0.21259  -0.102  0.91877
## scale(con_dens_s):trt_II -0.03646    0.14890  -0.245  0.80659
## scale(con_dens_s):trt_FF -0.04826    0.19110  -0.252  0.80063
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## $`(30.5,149]`
##  Family: binomial  ( logit )
## Formula:
## Pr_s ~ slope.degrees_s + (scale(tot_dens) + scale(con_dens_s)) *
##     (trt_I + trt_F) + (1 | species) + (1 | site/loc/gr/plot)
## Data: d
## Weights: census.start
##
##      AIC      BIC   logLik deviance df.resid
##   1195.2   1259.4   -582.6   1165.2      520
##
## Random effects:
##
## Conditional model:
##  Groups           Name        Variance  Std.Dev.
##  species          (Intercept) 1.408e+00 1.1867797
##  plot:gr:loc:site (Intercept) 1.392e-01 0.3731126
##  gr:loc:site      (Intercept) 3.098e-01 0.5566388
##  loc:site         (Intercept) 3.221e-01 0.5675476
##  site             (Intercept) 1.671e-08 0.0001293
## Number of obs: 535, groups:
## species, 22; plot:gr:loc:site, 263; gr:loc:site, 63; loc:site, 21; site, 7
##
## Conditional model:
##                           Estimate Std. Error z value Pr(>|z|)
## (Intercept)                0.53178    0.33873   1.570  0.11644
## slope.degrees_s           -0.02894    0.09389  -0.308  0.75788
## scale(tot_dens)           -0.32736    0.08167  -4.008 6.12e-05 ***
## scale(con_dens_s)         -0.47153    0.10987  -4.292 1.77e-05 ***
```

```
## trt_II                    0.66796    0.14118   4.731 2.23e-06 ***
## trt_FF                    0.04856    0.14158   0.343  0.73158
## scale(tot_dens):trt_II   -0.14344    0.10774  -1.331  0.18304
## scale(tot_dens):trt_FF    0.25646    0.08100   3.166  0.00155 **
## scale(con_dens_s):trt_II -0.01647    0.11729  -0.140  0.88830
## scale(con_dens_s):trt_FF  0.34613    0.11810   2.931  0.00338 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
## What about a single model with categorical fragment sizes

m_cdd_s_cat_ri <- glmmTMB(Pr_s ~ slope.degrees_s +
                       (scale(tot_dens) + scale(con_dens_s)) *
                       (trt_I + trt_F) *
                       frag_sizeclass3 +
                       (1|species) +
                       ## setting cor to 0 to converge
                       (1|site/loc/gr/plot),
                     weights = census.start, data = sdls,
                     family=binomial)
summary(m_cdd_s_cat_ri)
```

```
##  Family: binomial  ( logit )
## Formula:
## Pr_s ~ slope.degrees_s + (scale(tot_dens) + scale(con_dens_s)) *
##     (trt_I + trt_F) * frag_sizeclass3 + (1 | species) + (1 |
##     site/loc/gr/plot)
## Data: sdls
## Weights: census.start
##
##      AIC      BIC   logLik deviance df.resid
##   2190.2   2351.1  -1062.1   2124.2      935
##
## Random effects:
##
## Conditional model:
##  Groups           Name        Variance  Std.Dev.
##  species          (Intercept) 1.288e+00 1.1347817
##  plot:gr:loc:site (Intercept) 2.026e-01 0.4501594
##  gr:loc:site      (Intercept) 1.529e-01 0.3909984
##  loc:site         (Intercept) 3.730e-01 0.6107409
##  site             (Intercept) 1.339e-08 0.0001157
## Number of obs: 968, groups:
## species, 26; plot:gr:loc:site, 474; gr:loc:site, 110; loc:site, 37; site, 21
##
## Conditional model:
##                                                 Estimate Std. Error z value
## (Intercept)                                      1.85105    0.41523   4.458
## slope.degrees_s                                 -0.10584    0.06748  -1.568
## scale(tot_dens)                                 -0.48135    0.24136  -1.994
## scale(con_dens_s)                               -0.15125    0.08225  -1.839
## trt_II                                          -0.17653    0.25516  -0.692
## trt_FF                                          -0.12716    0.24995  -0.509
## frag_sizeclass3(5.36,30.5]                      -1.43466    0.42290  -3.392
```

```
## frag_sizeclass3(30.5,149]                              -1.43974    0.37518  -3.838
## scale(tot_dens):trt_II                                  0.37760    0.46687   0.809
## scale(tot_dens):trt_FF                                 -0.12544    0.38609  -0.325
## scale(con_dens_s):trt_II                               -0.33583    0.21172  -1.586
## scale(con_dens_s):trt_FF                                0.21189    0.22324   0.949
## scale(tot_dens):frag_sizeclass3(5.36,30.5]             -0.17833    0.33211  -0.537
## scale(tot_dens):frag_sizeclass3(30.5,149]               0.18246    0.25002   0.730
## scale(con_dens_s):frag_sizeclass3(5.36,30.5]            0.13309    0.12092   1.101
## scale(con_dens_s):frag_sizeclass3(30.5,149]            -0.38047    0.15135  -2.514
## trt_II:frag_sizeclass3(5.36,30.5]                       0.75832    0.32129   2.360
## trt_II:frag_sizeclass3(30.5,149]                        0.86138    0.29402   2.930
## trt_FF:frag_sizeclass3(5.36,30.5]                       0.26949    0.31934   0.844
## trt_FF:frag_sizeclass3(30.5,149]                        0.20441    0.29099   0.702
## scale(tot_dens):trt_II:frag_sizeclass3(5.36,30.5]      -1.51532    0.59619  -2.542
## scale(tot_dens):trt_II:frag_sizeclass3(30.5,149]       -0.48934    0.47641  -1.027
## scale(tot_dens):trt_FF:frag_sizeclass3(5.36,30.5]       0.12640    0.52109   0.243
## scale(tot_dens):trt_FF:frag_sizeclass3(30.5,149]        0.33064    0.39285   0.842
## scale(con_dens_s):trt_II:frag_sizeclass3(5.36,30.5]     0.27666    0.25709   1.076
## scale(con_dens_s):trt_II:frag_sizeclass3(30.5,149]      0.28871    0.25532   1.131
## scale(con_dens_s):trt_FF:frag_sizeclass3(5.36,30.5]    -0.28808    0.28960  -0.995
## scale(con_dens_s):trt_FF:frag_sizeclass3(30.5,149]      0.20920    0.26489   0.790
##                                                        Pr(>|z|)
## (Intercept)                                            8.28e-06 ***
## slope.degrees_s                                        0.116773
## scale(tot_dens)                                        0.046115 *
## scale(con_dens_s)                                      0.065920 .
## trt_II                                                 0.489036
## trt_FF                                                 0.610925
## frag_sizeclass3(5.36,30.5]                             0.000693 ***
## frag_sizeclass3(30.5,149]                              0.000124 ***
## scale(tot_dens):trt_II                                 0.418640
## scale(tot_dens):trt_FF                                 0.745257
## scale(con_dens_s):trt_II                               0.112701
## scale(con_dens_s):trt_FF                               0.342540
## scale(tot_dens):frag_sizeclass3(5.36,30.5]             0.591297
## scale(tot_dens):frag_sizeclass3(30.5,149]              0.465529
## scale(con_dens_s):frag_sizeclass3(5.36,30.5]           0.271081
## scale(con_dens_s):frag_sizeclass3(30.5,149]            0.011942 *
## trt_II:frag_sizeclass3(5.36,30.5]                      0.018264 *
## trt_II:frag_sizeclass3(30.5,149]                       0.003393 **
## trt_FF:frag_sizeclass3(5.36,30.5]                      0.398718
## trt_FF:frag_sizeclass3(30.5,149]                       0.482395
## scale(tot_dens):trt_II:frag_sizeclass3(5.36,30.5]      0.011031 *
## scale(tot_dens):trt_II:frag_sizeclass3(30.5,149]       0.304352
## scale(tot_dens):trt_FF:frag_sizeclass3(5.36,30.5]      0.808345
## scale(tot_dens):trt_FF:frag_sizeclass3(30.5,149]       0.399979
## scale(con_dens_s):trt_II:frag_sizeclass3(5.36,30.5]    0.281872
## scale(con_dens_s):trt_II:frag_sizeclass3(30.5,149]     0.258149
## scale(con_dens_s):trt_FF:frag_sizeclass3(5.36,30.5]    0.319861
## scale(con_dens_s):trt_FF:frag_sizeclass3(30.5,149]     0.429671
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
car::Anova(m_cdd_s_cat_ri)
```

```
## Analysis of Deviance Table (Type II Wald chisquare tests)
##
## Response: Pr_s
##                                          Chisq Df Pr(>Chisq)
## slope.degrees_s                         2.4601  1   0.116773
## scale(tot_dens)                        24.0078  1  9.595e-07 ***
## scale(con_dens_s)                      15.9951  1  6.351e-05 ***
## trt_I                                  17.6831  1  2.609e-05 ***
## trt_F                                   3.7025  1   0.054330 .
## frag_sizeclass3                         8.3175  2   0.015627 *
## scale(tot_dens):trt_I                   2.9478  1   0.085996 .
## scale(tot_dens):trt_F                   7.5625  1   0.005959 **
## scale(con_dens_s):trt_I                 1.3192  1   0.250742
## scale(con_dens_s):trt_F                 5.0734  1   0.024295 *
## scale(tot_dens):frag_sizeclass3         8.3555  2   0.015333 *
## scale(con_dens_s):frag_sizeclass3       8.0497  2   0.017866 *
## trt_I:frag_sizeclass3                  10.0242  2   0.006657 **
## trt_F:frag_sizeclass3                   0.8234  2   0.662525
## scale(tot_dens):trt_I:frag_sizeclass3   8.6274  2   0.013384 *
## scale(tot_dens):trt_F:frag_sizeclass3   1.0077  2   0.604192
## scale(con_dens_s):trt_I:frag_sizeclass3 1.4575  2   0.482505
## scale(con_dens_s):trt_F:frag_sizeclass3 4.5986  2   0.100331
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Take-homes

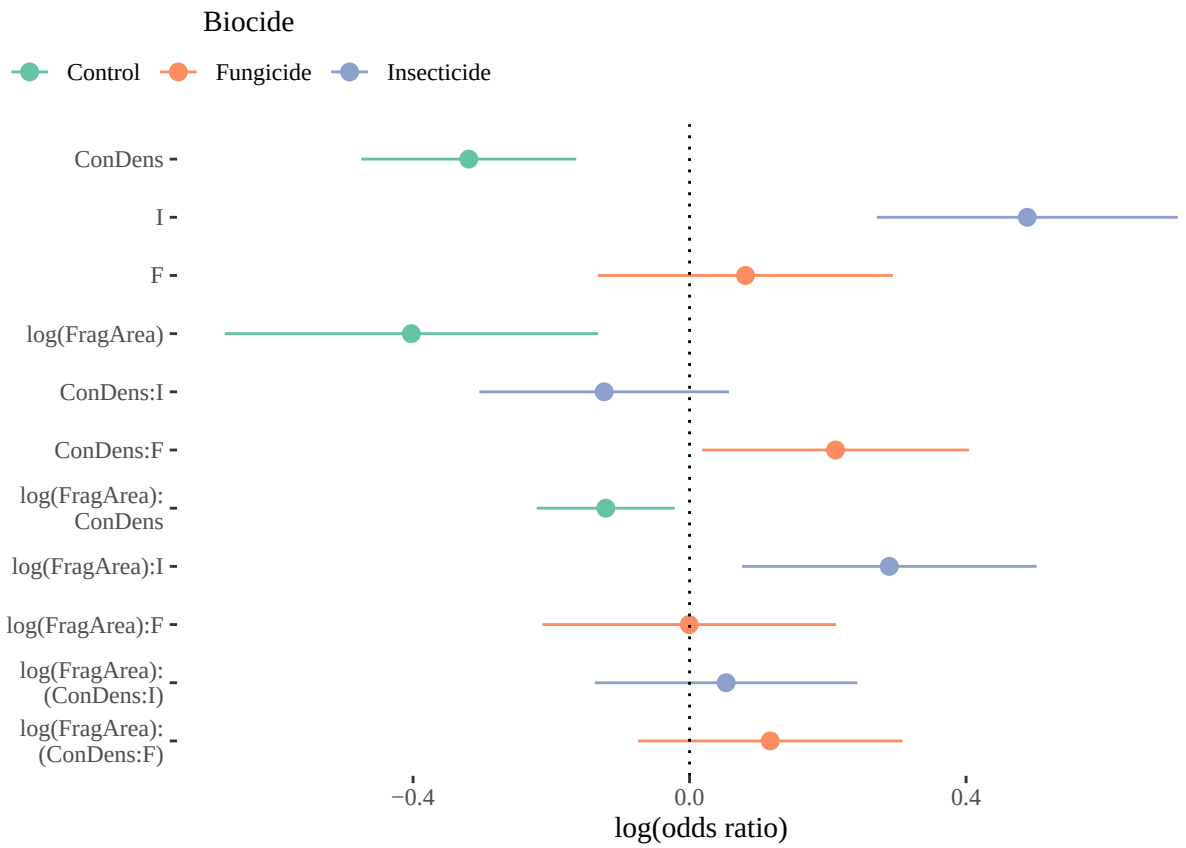
The effect of fragment area is complex. * Density dependence is only present in large fragments. * Fungicide removes the effect in these large fragments.

Although the interaction between fragment size and cdd and fungicide is not significantly different from 0, this is owed more to its uncertainty than its magnitude.

Graphics

```
labs <-  c("ConDens", "I", "F", "log(FragArea)",
             "ConDens:I", "ConDens:F", "log(FragArea):\n ConDens",
             "log(FragArea):I"  , "log(FragArea):F",
             "log(FragArea):\n (ConDens:I)",
             "log(FragArea):\n (ConDens:F)")
tp_cdd_s_ri <-
  tidy(m_cdd_s_ri, conf.int = TRUE) |>
  filter(effect == "fixed",
         !str_detect(term, "tot_dens"),
         !term %in% c("(Intercept)", "slope.degrees_s")) |>
  mutate(labels = factor(labs, levels = rev(labs)),
         Biocide = case_when(
           str_detect(term, "trt_II") ~  "Insecticide",
           str_detect(term, "trt_FF") ~ "Fungicide",
```

```
          .default =  "Control")) |>
  ggplot(aes(y = labels, x = estimate, xmin = conf.low, xmax = conf.high,
             colour = Biocide)) +
  geom_pointrange() +
  geom_vline(xintercept=0, linetype = "dotted") +
  scale_colour_brewer(palette="Set2") +
  labs(y = NULL, x = "log(odds ratio)")  +
  guides(colour = guide_legend(position = "top", direction = "horizontal",
                               title.position = "top", title.hjust = 0.5)) +
  theme(
    legend.margin = margin(0, 0, 0, 0),
    legend.justification.top = "left",
    legend.location = "plot",
    plot.title.position = "plot"
  )
tp_cdd_s_ri
```

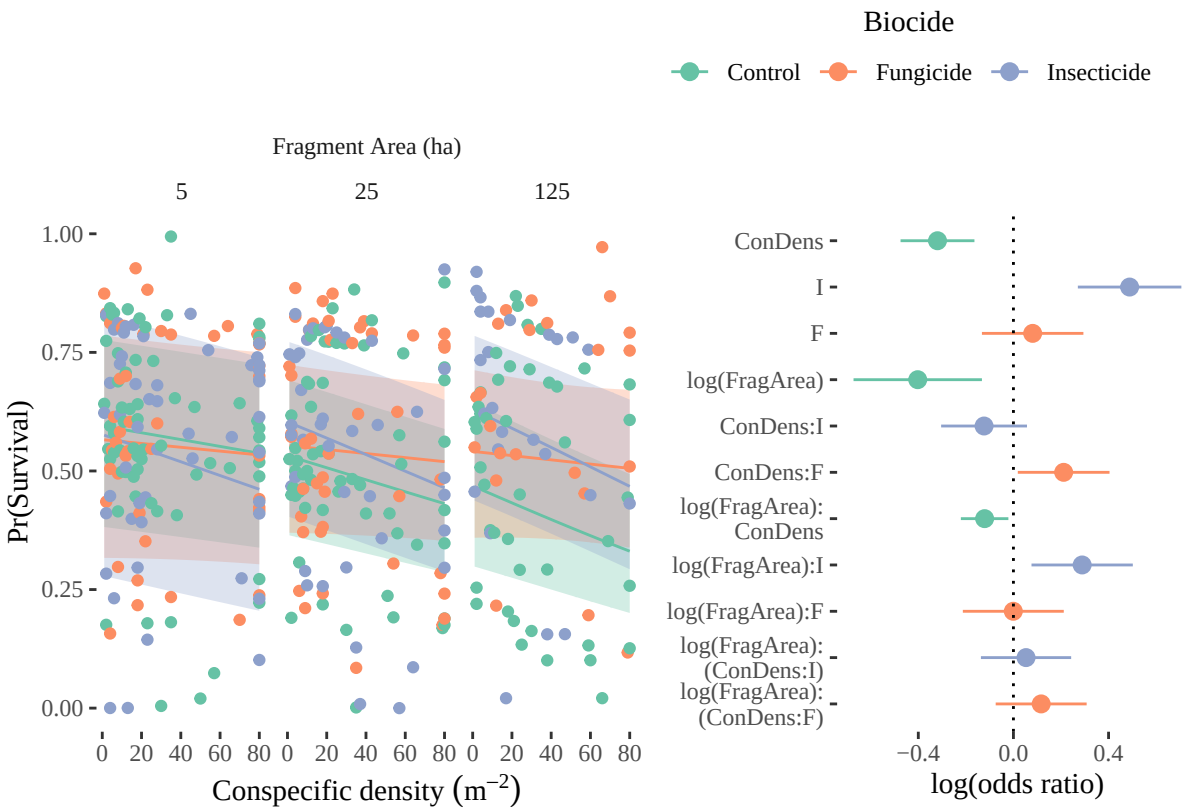



```
p_cdd_s_ri <- predict_response(m_cdd_s_ri,
                               terms=c("con_dens_s[1:80, by = 1]",
                                       "trt_F", "trt_I",
                                       "fragment.size[5, 25, 125]"),
                              margin="marginalmeans")

pr_cdd_s_ri <- residualize_over_grid(p_cdd_s_ri, m_cdd_s_ri) |>
   bind_cols(species = sdls$species) |>
```

```
  mutate(trt = case_when(
    group == "0" & facet == "0" ~ "Control",
    group == "F" & facet == "0" ~ "Fungicide",
    group == "0" & facet == "I" ~ "Insecticide",
    group == "F" & facet == "I" ~ "FI"),
    facet_lab = "Fragment Area (ha)") |>
  rename("con_dens" = "x", "frag_area" = "panel") |>
  group_by(con_dens, trt, frag_area, facet_lab, species) |>
  summarise(predicted = mean(predicted), n = n())
```

```
## `summarise()` has grouped output by 'con_dens', 'trt', 'frag_area',
## 'facet_lab'. You can override using the `.groups` argument.
```

```
p_cdd_s_ri <- as.data.frame(p_cdd_s_ri) |>
  mutate(trt = case_when(
    group == "0" & facet == "0" ~ "Control",
    group == "F" & facet == "0" ~ "Fungicide",
    group == "0" & facet == "I" ~ "Insecticide",
    group == "F" & facet == "I" ~ "FI"),
    facet_lab = "Fragment Area (ha)") |>
  rename("con_dens" = "x", "frag_area" = "panel")

pl_cdd_s <- ggplot(filter(p_cdd_s_ri, trt !="FI"),
       aes(x=con_dens, y = predicted, colour = trt)) +
  ggh4x::facet_nested(~facet_lab + frag_area) +
  geom_ribbon(aes(ymin = conf.low, ymax = conf.high, fill = trt),
              colour = NA, alpha = 0.3) +
  geom_line() +
  geom_point(data = filter(pr_cdd_s_ri, trt != "FI")) +
  scale_colour_brewer(palette="Set2") +
  scale_fill_brewer(palette="Set2", guide = "none") +
  labs(x = expression(Conspecific~density~(m^-2)), y = "Pr(Survival)",
       colour = "Treatment") + theme(legend.position="none")


(pl_cdd_s  | tp_cdd_s_ri )  +
  plot_layout(widths=c(0.6, 0.4))
```



Plotting effects of fragment area

The effects are complex and not terribly clear. To interrogate them more, here are some plots

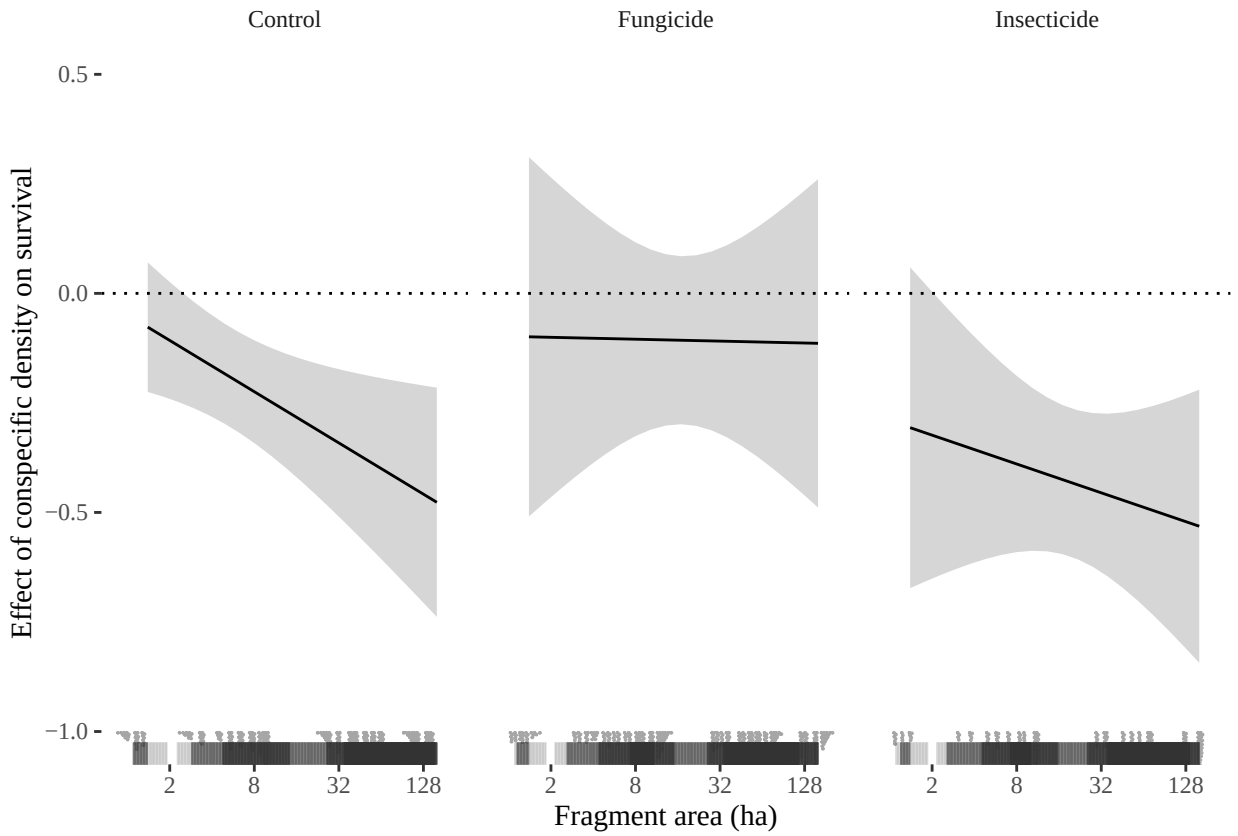
```
## Using the model split by area to
sdls |> group_by(frag_sizeclass3) |> summarize(mean(fragment.size))
```

```
## # A tibble: 3 x 2
##   frag_sizeclass3 `mean(fragment.size)`
##   <fct>                           <dbl>
## 1 [1.1,5.36]                       2.64
## 2 (5.36,30.5]                     12.2
## 3 (30.5,149]                      85.3
```

```
## plot the effect of density over fragment size
int_data <- sdls |>
  mutate(con_dens_ss = scale(con_dens_s),
         tot_dens_s = scale(tot_dens),
         log_fragsize_s = scale(log(fragment.size)))

int_mod <- glmmTMB(Pr_s ~ slope.degrees_s +
                     (tot_dens_s + con_dens_ss) *
                     (trt_I + trt_F) * log_fragsize_s +
                     (1| species) +
```

```
                    (1 | site/loc/gr/plot),
                  weights = census.start, family = binomial,
                  data = int_data)

preddat <- expand.grid(
  trt_F = levels(int_data$trt_F),
  con_dens_ss = 1,
  log_fragsize_s = seq(-2, 1.3, length =20))

xmat <- model.matrix(~ con_dens_ss + con_dens_ss:(trt_F + log_fragsize_s) +
                       con_dens_ss:trt_F:log_fragsize_s,
                     preddat)[ , -1]
preddat$int_hat <- as.vector(xmat %*% fixef(int_mod)$cond[colnames(xmat)])
vmat <- (vcov(int_mod)$cond[colnames(xmat), colnames(xmat)])
preddat$int_se <-  sqrt(diag(xmat %*% vmat %*% t(xmat)))

## Now insecticide
preddat_i <- expand.grid(
  trt_I = levels(int_data$trt_I),
  con_dens_ss = 1,
  log_fragsize_s = seq(-2, 1.3, length =20))

xmat <- model.matrix(~ con_dens_ss + con_dens_ss:(trt_I + log_fragsize_s) +
                       con_dens_ss:trt_I:log_fragsize_s,
                     preddat_i)[ , -1]
preddat_i$int_hat <- as.vector(xmat %*% fixef(int_mod)$cond[colnames(xmat)])
vmat <- (vcov(int_mod)$cond[colnames(xmat), colnames(xmat)])
preddat_i$int_se <-  sqrt(diag(xmat %*% vmat %*% t(xmat)))

preddat <- bind_rows(preddat |> rename("trt" = "trt_F"),
                     filter(preddat_i, trt_I == "I") |> rename("trt" = "trt_I"))

preddat <- preddat |> mutate(.lower = int_hat - 1.96*int_se,
                             .upper = int_hat + 1.96*int_se,
                             frag_area = exp(log_fragsize_s*
                               sd(log(sdls$fragment.size)) +
                               mean(log(sdls$fragment.size))),
                             trt =factor(trt,
                                         labels = c("Control", "Fungicide",
                                                    "Insecticide")))
int_plot <- ggplot(preddat, aes(x = frag_area)) + facet_wrap( ~ trt) +
  geom_ribbon(aes(y = int_hat, ymin = .lower, ymax = .upper), alpha = 0.2, colour = NA) +
  geom_line(aes(y = int_hat)) + geom_hline(yintercept=0, linetype = "dotted") +
  #coord_trans(x = "log") +
  scale_x_continuous(trans = "log", breaks=c(2, 8, 32, 128)) +
  labs(x =  "Fragment area (ha)",
       y = "Effect of conspecific density on survival")


sdls <- mutate(sdls, trt = case_when(
  trt_F == "F" ~ "F",
  trt_I == "I" ~ "I",
  .default = "C"),
```

```
  trt = factor(trt, labels = c("Control", "Fungicide", "Insecticide")))
library(ggdist)

int_plot <- int_plot + ggdist::geom_dots(data = sdls, aes(x = fragment.size), y =  -1,
                              smooth = smooth_unbounded(), layout = "swarm",
                              side = "bottom", binwidth = 0.04,# alpha = 0.7,
                              overflow = "compress")+  ylim(c(-1, 0.5))

int_plot + ggdensity::geom_hdr_rug(data = sdls, aes(x = fragment.size))  +
  theme(legend.position="none")
```




```
## double checking results against interactions package (only works for
## lme4 and 2-way interactions
test_mod_C <-  lme4::glmer(Pr_s ~ slope.degrees_s + trt_I +
                      (tot_dens_s + con_dens_ss) *
                      log_fragsize_s +
                       (1|species) +
                       ## setting cor to 0 to converge
                       (1|site/loc/gr/plot),
                     weights = census.start,
                     data =filter(int_data, trt_F =="0") ,
                     family=binomial)
```

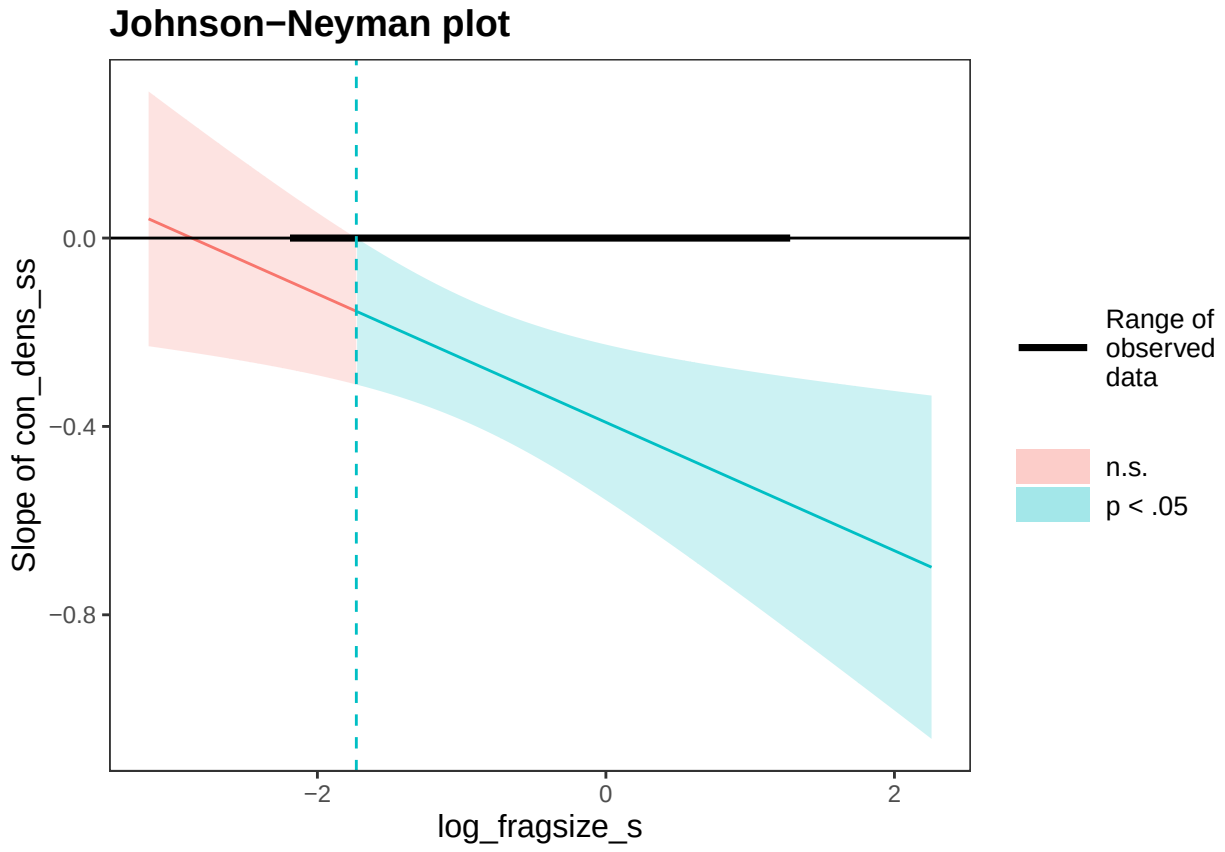
```
## Warning in checkConv(attr(opt, "derivs"), opt$par, ctrl = control$checkConv, :
## Model failed to converge with max|grad| = 0.00859555 (tol = 0.002, component 1)
```

```
test_mod_F <-  lme4::glmer(Pr_s ~ slope.degrees_s + trt_I +
                        (tot_dens_s + con_dens_ss) *
                        log_fragsize_s +
                         (1|species) +
                         ## setting cor to 0 to converge
                         (1|site/loc/gr/plot),
                       weights = census.start,
                       data =filter(int_data, trt_F =="F"),
                       family=binomial)
```

```
## boundary (singular) fit: see help('isSingular')
```

```
interactions::johnson_neyman(test_mod_C, pred = con_dens_ss,
                             modx =   log_fragsize_s)
```

```
## JOHNSON-NEYMAN INTERVAL
##
## When log_fragsize_s is OUTSIDE the interval [-9.98, -1.73], the slope of
## con_dens_ss is p < .05.
##
## Note: The range of observed values of log_fragsize_s is [-2.17, 1.25]
```

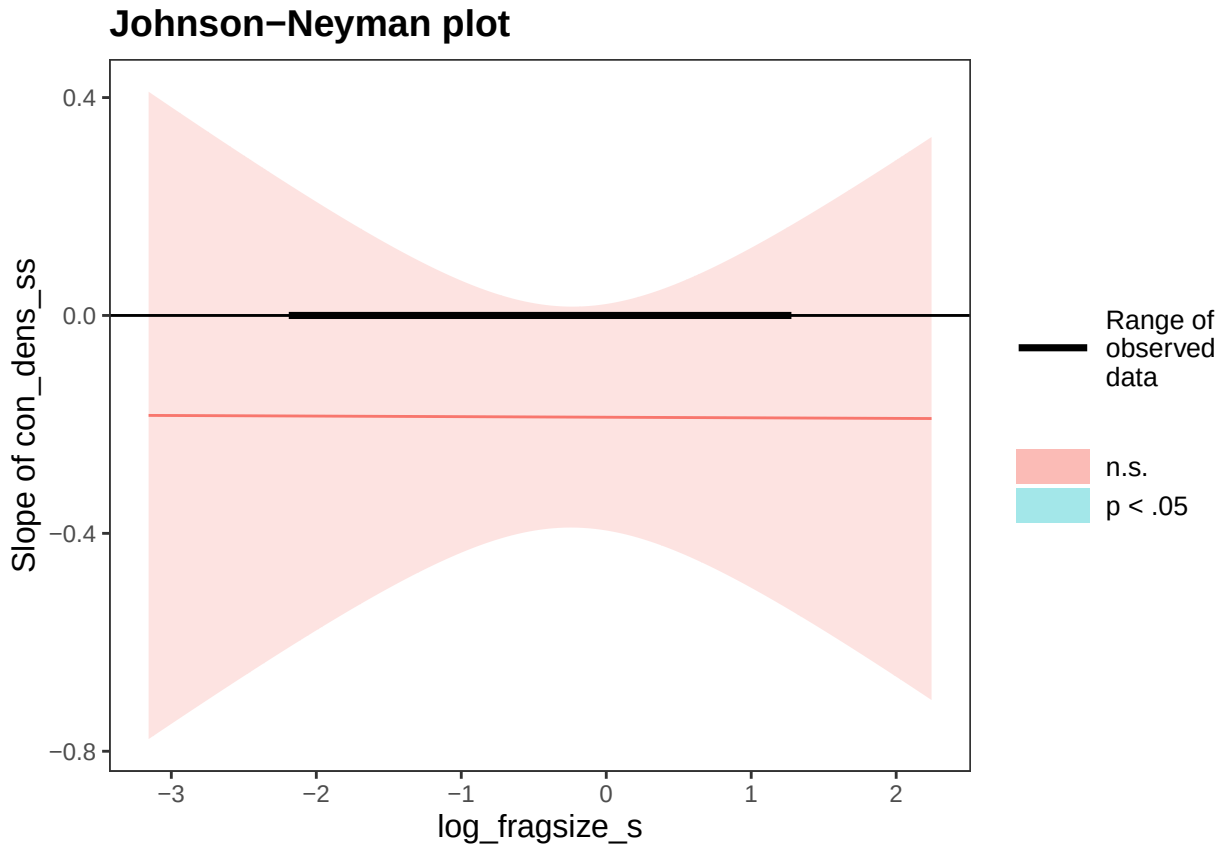


```
exp(-1.73*sd(log(int_data$fragment.size)) + mean(log(int_data$fragment.size)))
```

```
## [1] 2.055449
```

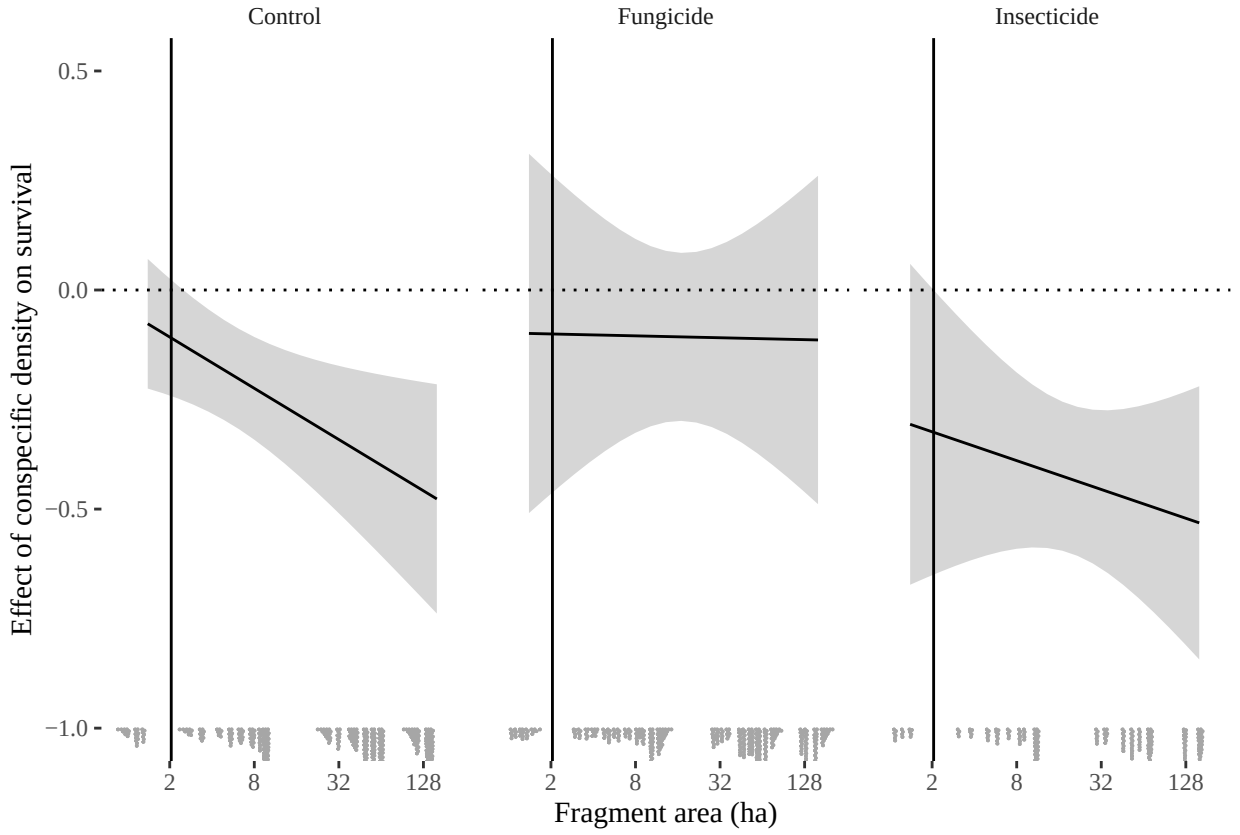
```
interactions::johnson_neyman(test_mod_F, pred = con_dens_ss,
                             modx = log_fragsize_s)
```

```
## JOHNSON-NEYMAN INTERVAL
##
## The Johnson-Neyman interval could not be found. Is the p value for your
## interaction term below the specified alpha?
```




```
## basically the same result
int_plot + ggdist::geom_dots(data = sdls, aes(x = fragment.size), y =  -1,
                             smooth = smooth_unbounded(), layout = "swarm",
                             side = "bottom", binwidth = 0.04,# alpha = 0.7,
                             overflow = "compress")+  ylim(c(-1, 0.5)) +
  geom_vline(xintercept=2.05)
```

```
## Scale for y is already present.
## Adding another scale for y, which will replace the existing scale.
```

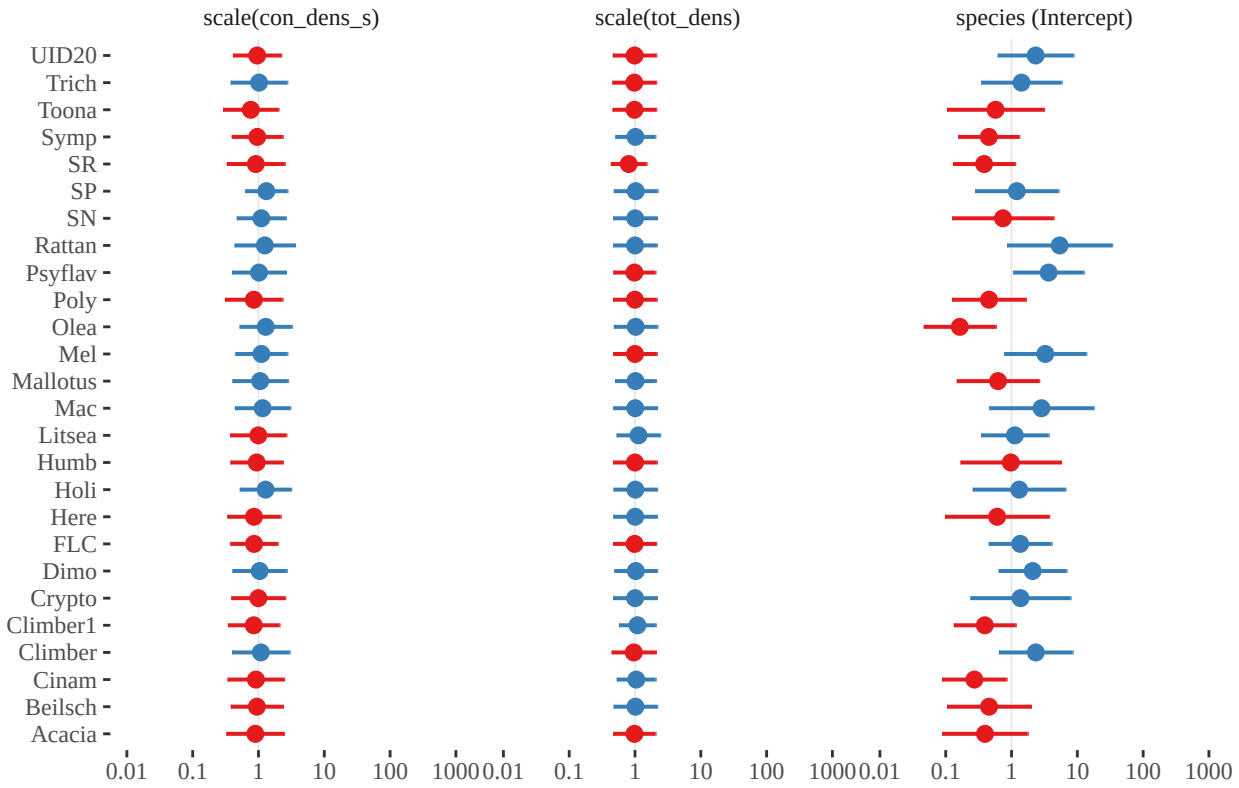


Species specific inferences.

Initial analyses suggested that some species were heavily influencing the results. The random slope should partly account for that.

```
sjPlot::plot_model(m_cdd_s_ris, type = "re", terms = "species", ri.nr = 1)
```

Random effects




```
## very little variation among species in slopes, but perhaps obscured
## by large variation in the intercept
sp_eff <- as.data.frame(ranef(m_cdd_s_ris, condVar = T)$cond$species)

sp_eff <- bind_cols(sp_eff,
                    t(apply(attr(sp_eff, "condVar"), 3, function(x)
                      sqrt(diag(x)))))
```

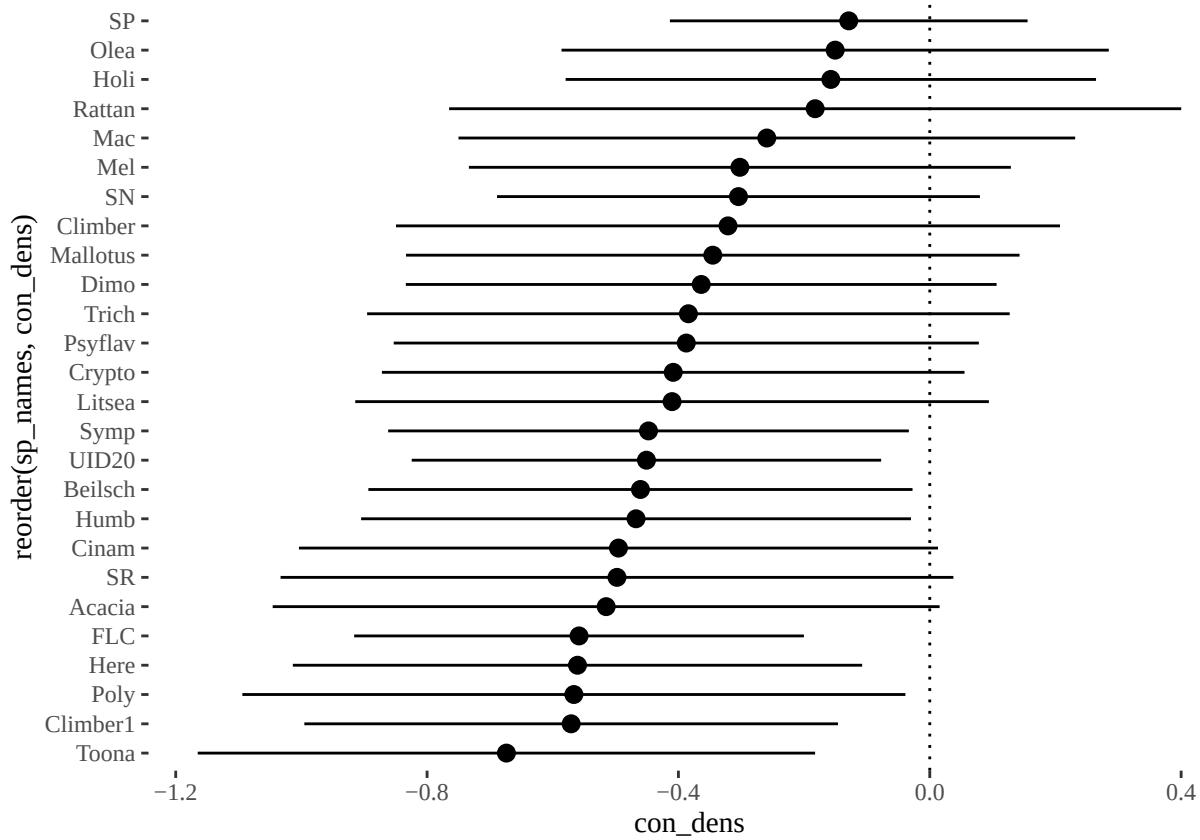
```
## New names:
## * `` -> `...4`
## * `` -> `...5`
## * `` -> `...6`
```

```
names(fixef(m_cdd_s_ris)$cond)
```

```
##  [1] "(Intercept)"
##  [2] "slope.degrees_s"
##  [3] "scale(tot_dens)"
##  [4] "scale(con_dens_s)"
##  [5] "trt_II"
##  [6] "trt_FF"
##  [7] "scale(log(fragment.size))"
##  [8] "scale(tot_dens):trt_II"
##  [9] "scale(tot_dens):trt_FF"
## [10] "scale(con_dens_s):trt_II"
```

```
## [11] "scale(con_dens_s):trt_FF"
## [12] "scale(tot_dens):scale(log(fragment.size))"
## [13] "scale(con_dens_s):scale(log(fragment.size))"
## [14] "trt_II:scale(log(fragment.size))"
## [15] "trt_FF:scale(log(fragment.size))"
## [16] "scale(tot_dens):trt_II:scale(log(fragment.size))"
## [17] "scale(tot_dens):trt_FF:scale(log(fragment.size))"
## [18] "scale(con_dens_s):trt_II:scale(log(fragment.size))"
## [19] "scale(con_dens_s):trt_FF:scale(log(fragment.size))"
```

```
names(sp_eff) <- c("Intercept", "con_dens", "tot_dens",
                   "Intercept_se", "con_dens_se", "tot_dens_se")
sp_eff <- mutate(sp_eff, sp_names = row.names(sp_eff),
                 Intercept = Intercept + fixef(m_cdd_s_ris)$cond[1],
                 con_dens = con_dens + fixef(m_cdd_s_ris)$cond["scale(con_dens_s)"],
                 tot_dens = tot_dens + fixef(m_cdd_s_ris)$cond["scale(tot_dens)"])

sp_eff |> arrange(con_dens) |>
  ggplot(aes(y = reorder(sp_names, con_dens), x = con_dens,
             xmin = con_dens - 2*con_dens_se,
             xmax = con_dens + 2*con_dens_se)) +
    geom_vline(xintercept=0, linetype = "dotted") +
  geom_pointrange()
```

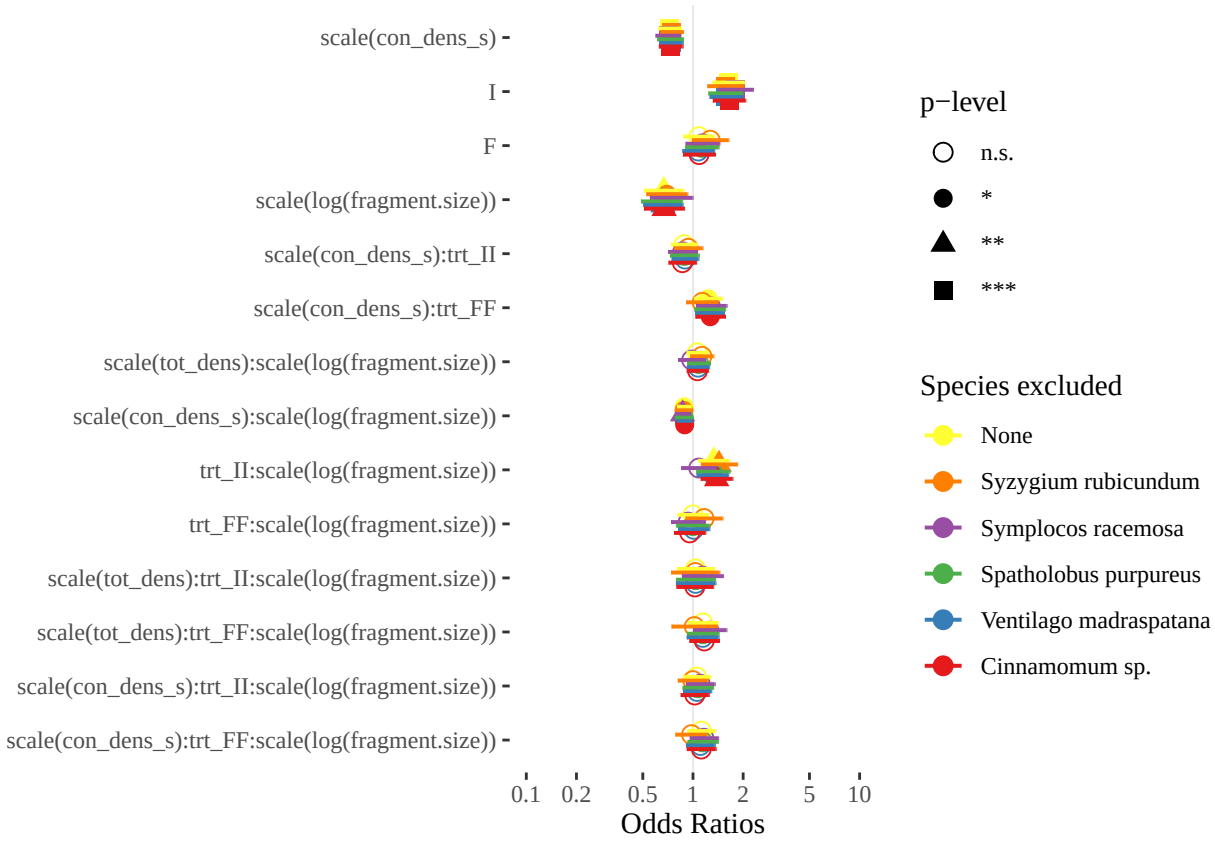


```
## TBH not very illuminating.
```

Little variation in cdd among species.

Perhaps remove most abundant species to check if patterns are robust.

```
sp_common <- sdls |> group_by(species) |> summarise(n = sum(census.start)) |>
   arrange(desc(n)) |> pull(species)

sp_mods <- lapply(c("none", as.character(sp_common[1:5])), function(i) {
  update(m_cdd_s_ri, data = filter(sdls, !species == i))})
sp_codes <- mutate(sp_codes, spbin = paste(genus, species))
term_nms <- names(fixef(m_cdd_s_ri_frag[[1]]))$cond)

plot_models(sp_mods,
            rm.terms = c("slope.degrees_s",
                         term_nms[str_detect(term_nms, "tot")]),
            m.labels = c("None",
                         sp_codes$spbin[match(sp_common[1:5], sp_codes$code)]),
            p.shape=TRUE) + labs(colour = "Species excluded")
```




```
## Only species that makes a difference is S. rubicundum - removing it dampens
## interaction between density and fungicide. Symplocus affects insects a bit.
```

what about single species models?

```r
## base model
## remove species random effect
names(sp_common) <- sp_common
single_sp_mods <- map(sp_common[1:5], function(i) {
  update(m_cdd_s_ri, data = filter(sdls, species == i))})
names(single_sp_mods) <- sp_codes$spbin[match(names(single_sp_mods), sp_codes$code)]
single_sp_mods$All <- m_cdd_s_ris

map(single_sp_mods, summary)
```

```
## $`Syzygium rubicundum`
##  Family: binomial  ( logit )
## Formula:
## Pr_s ~ slope.degrees_s + (scale(tot_dens) + scale(con_dens_s)) *
##     (trt_I + trt_F) * scale(log(fragment.size)) + (1 | species) +
##     (1 | site/loc/gr/plot)
## Data: filter(sdls, species == i)
## Weights: census.start
##
##      AIC      BIC   logLik deviance df.resid
##    665.3    740.8   -308.6    617.3      148
##
## Random effects:
##
## Conditional model:
##  Groups           Name        Variance  Std.Dev.
##  species          (Intercept) 2.446e-09 4.946e-05
##  plot:gr:loc:site (Intercept) 2.332e-01 4.830e-01
##  gr:loc:site      (Intercept) 2.258e-01 4.752e-01
##  loc:site         (Intercept) 3.933e-01 6.272e-01
##  site             (Intercept) 1.395e-01 3.735e-01
## Number of obs: 172, groups:
## species, 1; plot:gr:loc:site, 172; gr:loc:site, 61; loc:site, 30; site, 16
##
## Conditional model:
##                                                  Estimate Std. Error z value
## (Intercept)                                      -0.08341    0.23663  -0.352
## slope.degrees_s                                  -0.26105    0.10750  -2.428
## scale(tot_dens)                                  -0.97014    1.11892  -0.867
## scale(con_dens_s)                                 0.26289    1.11257   0.236
## trt_II                                            0.59067    0.19786   2.985
## trt_FF                                           -0.14284    0.17851  -0.800
## scale(log(fragment.size))                        -0.30395    0.23268  -1.306
## scale(tot_dens):trt_II                            1.10639    1.40006   0.790
## scale(tot_dens):trt_FF                            1.31054    1.61439   0.812
## scale(con_dens_s):trt_II                         -1.09087    1.37685  -0.792
## scale(con_dens_s):trt_FF                         -1.17973    1.53541  -0.768
## scale(tot_dens):scale(log(fragment.size))         2.18684    1.33337   1.640
## scale(con_dens_s):scale(log(fragment.size))      -2.10714    1.32263  -1.593
## trt_II:scale(log(fragment.size))                  0.13646    0.19755   0.691
## trt_FF:scale(log(fragment.size))                 -0.24242    0.20876  -1.161
## scale(tot_dens):trt_II:scale(log(fragment.size)) -3.10582    2.10539  -1.475
## scale(tot_dens):trt_FF:scale(log(fragment.size)) -1.30978    2.73360  -0.479
```

```
## scale(con_dens_s):trt_II:scale(log(fragment.size))  2.96720    2.00343   1.481
## scale(con_dens_s):trt_FF:scale(log(fragment.size))  1.69460    2.59809   0.652
##                                                    Pr(>|z|)
## (Intercept)                                         0.72447
## slope.degrees_s                                     0.01516 *
## scale(tot_dens)                                     0.38593
## scale(con_dens_s)                                   0.81321
## trt_II                                              0.00283 **
## trt_FF                                              0.42361
## scale(log(fragment.size))                           0.19145
## scale(tot_dens):trt_II                              0.42939
## scale(tot_dens):trt_FF                              0.41692
## scale(con_dens_s):trt_II                            0.42819
## scale(con_dens_s):trt_FF                            0.44228
## scale(tot_dens):scale(log(fragment.size))           0.10099
## scale(con_dens_s):scale(log(fragment.size))         0.11113
## trt_II:scale(log(fragment.size))                    0.48973
## trt_FF:scale(log(fragment.size))                    0.24554
## scale(tot_dens):trt_II:scale(log(fragment.size))    0.14017
## scale(tot_dens):trt_FF:scale(log(fragment.size))    0.63184
## scale(con_dens_s):trt_II:scale(log(fragment.size))  0.13859
## scale(con_dens_s):trt_FF:scale(log(fragment.size))  0.51424
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## $`Symplocos racemosa`
##  Family: binomial  ( logit )
## Formula:
## Pr_s ~ slope.degrees_s + (scale(tot_dens) + scale(con_dens_s)) *
##     (trt_I + trt_F) * scale(log(fragment.size)) + (1 | species) +
##     (1 | site/loc/gr/plot)
## Data: filter(sdls, species == i)
## Weights: census.start
##
##      AIC      BIC   logLik deviance df.resid
##    328.6    395.9   -140.3    280.6       98
##
## Random effects:
##
## Conditional model:
##  Groups           Name        Variance  Std.Dev.
##  species          (Intercept) 2.301e-10 1.517e-05
##  plot:gr:loc:site (Intercept) 4.766e-02 2.183e-01
##  gr:loc:site      (Intercept) 2.541e-01 5.041e-01
##  loc:site         (Intercept) 2.110e-01 4.593e-01
##  site             (Intercept) 7.718e-09 8.785e-05
## Number of obs: 122, groups:
## species, 1; plot:gr:loc:site, 122; gr:loc:site, 53; loc:site, 24; site, 17
##
## Conditional model:
##                                                    Estimate Std. Error z value
## (Intercept)                                         0.56597    0.26352   2.148
## slope.degrees_s                                     0.04268    0.15992   0.267
## scale(tot_dens)                                     0.03218    0.43931   0.073
```

```
## scale(con_dens_s)                                     -0.50090    0.43672  -1.147
## trt_II                                                -0.19685    0.41647  -0.473
## trt_FF                                                 0.34812    0.53743   0.648
## scale(log(fragment.size))                             -0.82686    0.29724  -2.782
## scale(tot_dens):trt_II                                -2.34392    1.34650  -1.741
## scale(tot_dens):trt_FF                                 1.92338    1.65705   1.161
## scale(con_dens_s):trt_II                               0.96019    1.11657   0.860
## scale(con_dens_s):trt_FF                              -1.32933    1.11749  -1.190
## scale(tot_dens):scale(log(fragment.size))             -0.06029    0.43948  -0.137
## scale(con_dens_s):scale(log(fragment.size))           -0.21580    0.38066  -0.567
## trt_II:scale(log(fragment.size))                       1.07587    0.51913   2.072
## trt_FF:scale(log(fragment.size))                      -0.07420    0.62343  -0.119
## scale(tot_dens):trt_II:scale(log(fragment.size))       1.45064    0.99840   1.453
## scale(tot_dens):trt_FF:scale(log(fragment.size))       0.24135    1.66497   0.145
## scale(con_dens_s):trt_II:scale(log(fragment.size)) -0.38161    0.96850  -0.394
## scale(con_dens_s):trt_FF:scale(log(fragment.size)) -1.53899    1.19084  -1.292
##                                                       Pr(>|z|)
## (Intercept)                                           0.03173 *
## slope.degrees_s                                       0.78957
## scale(tot_dens)                                       0.94161
## scale(con_dens_s)                                     0.25139
## trt_II                                                0.63646
## trt_FF                                                0.51715
## scale(log(fragment.size))                             0.00541 **
## scale(tot_dens):trt_II                                0.08173 .
## scale(tot_dens):trt_FF                                0.24576
## scale(con_dens_s):trt_II                              0.38982
## scale(con_dens_s):trt_FF                              0.23422
## scale(tot_dens):scale(log(fragment.size))             0.89089
## scale(con_dens_s):scale(log(fragment.size))           0.57078
## trt_II:scale(log(fragment.size))                      0.03822 *
## trt_FF:scale(log(fragment.size))                      0.90526
## scale(tot_dens):trt_II:scale(log(fragment.size))      0.14624
## scale(tot_dens):trt_FF:scale(log(fragment.size))      0.88474
## scale(con_dens_s):trt_II:scale(log(fragment.size))    0.69356
## scale(con_dens_s):trt_FF:scale(log(fragment.size))    0.19623
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## $`Spatholobus purpureus`
##  Family: binomial  ( logit )
## Formula:
## Pr_s ~ slope.degrees_s + (scale(tot_dens) + scale(con_dens_s)) *
##     (trt_I + trt_F) * scale(log(fragment.size)) + (1 | species) +
##     (1 | site/loc/gr/plot)
## Data: filter(sdls, species == i)
## Weights: census.start
##
##      AIC      BIC   logLik deviance df.resid
##    223.9    280.2    -88.0    175.9       53
##
## Random effects:
##
## Conditional model:
```

```
##  Groups           Name        Variance  Std.Dev.
##  species          (Intercept) 6.017e-15 7.757e-08
##  plot:gr:loc:site (Intercept) 8.422e-10 2.902e-05
##  gr:loc:site      (Intercept) 5.644e-13 7.513e-07
##  loc:site         (Intercept) 3.947e-12 1.987e-06
##  site             (Intercept) 2.264e-12 1.505e-06
## Number of obs: 77, groups:
## species, 1; plot:gr:loc:site, 77; gr:loc:site, 30; loc:site, 15; site, 10
##
## Conditional model:
##                                                  Estimate Std. Error z value
## (Intercept)                                      -0.69938    0.26370  -2.652
## slope.degrees_s                                  -0.18592    0.16791  -1.107
## scale(tot_dens)                                  -0.35819    0.34294  -1.044
## scale(con_dens_s)                                 0.30691    0.38895   0.789
## trt_II                                            0.46813    0.89568   0.523
## trt_FF                                            0.03359    0.35750   0.094
## scale(log(fragment.size))                         0.06661    0.37245   0.179
## scale(tot_dens):trt_II                           -0.70777    0.41933  -1.688
## scale(tot_dens):trt_FF                            0.42425    0.42882   0.989
## scale(con_dens_s):trt_II                         -0.41483    1.81542  -0.228
## scale(con_dens_s):trt_FF                         -0.92405    0.54114  -1.708
## scale(tot_dens):scale(log(fragment.size))         0.37376    0.60126   0.622
## scale(con_dens_s):scale(log(fragment.size))      -1.30508    1.05454  -1.238
## trt_II:scale(log(fragment.size))                  1.06328    1.95823   0.543
## trt_FF:scale(log(fragment.size))                  0.59247    0.55174   1.074
## scale(tot_dens):trt_II:scale(log(fragment.size))  0.37598    0.71378   0.527
## scale(tot_dens):trt_FF:scale(log(fragment.size)) -0.48323    0.76957  -0.628
## scale(con_dens_s):trt_II:scale(log(fragment.size))  1.42882    5.04251   0.283
## scale(con_dens_s):trt_FF:scale(log(fragment.size))  2.10801    1.40836   1.497
##                                                  Pr(>|z|)
## (Intercept)                                        0.0080 **
## slope.degrees_s                                    0.2682
## scale(tot_dens)                                    0.2963
## scale(con_dens_s)                                  0.4301
## trt_II                                             0.6012
## trt_FF                                             0.9251
## scale(log(fragment.size))                          0.8581
## scale(tot_dens):trt_II                             0.0914 .
## scale(tot_dens):trt_FF                             0.3225
## scale(con_dens_s):trt_II                           0.8193
## scale(con_dens_s):trt_FF                           0.0877 .
## scale(tot_dens):scale(log(fragment.size))          0.5342
## scale(con_dens_s):scale(log(fragment.size))        0.2159
## trt_II:scale(log(fragment.size))                   0.5871
## trt_FF:scale(log(fragment.size))                   0.2829
## scale(tot_dens):trt_II:scale(log(fragment.size))   0.5984
## scale(tot_dens):trt_FF:scale(log(fragment.size))   0.5301
## scale(con_dens_s):trt_II:scale(log(fragment.size))  0.7769
## scale(con_dens_s):trt_FF:scale(log(fragment.size))  0.1345
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## $`Ventilago madraspatana`
```

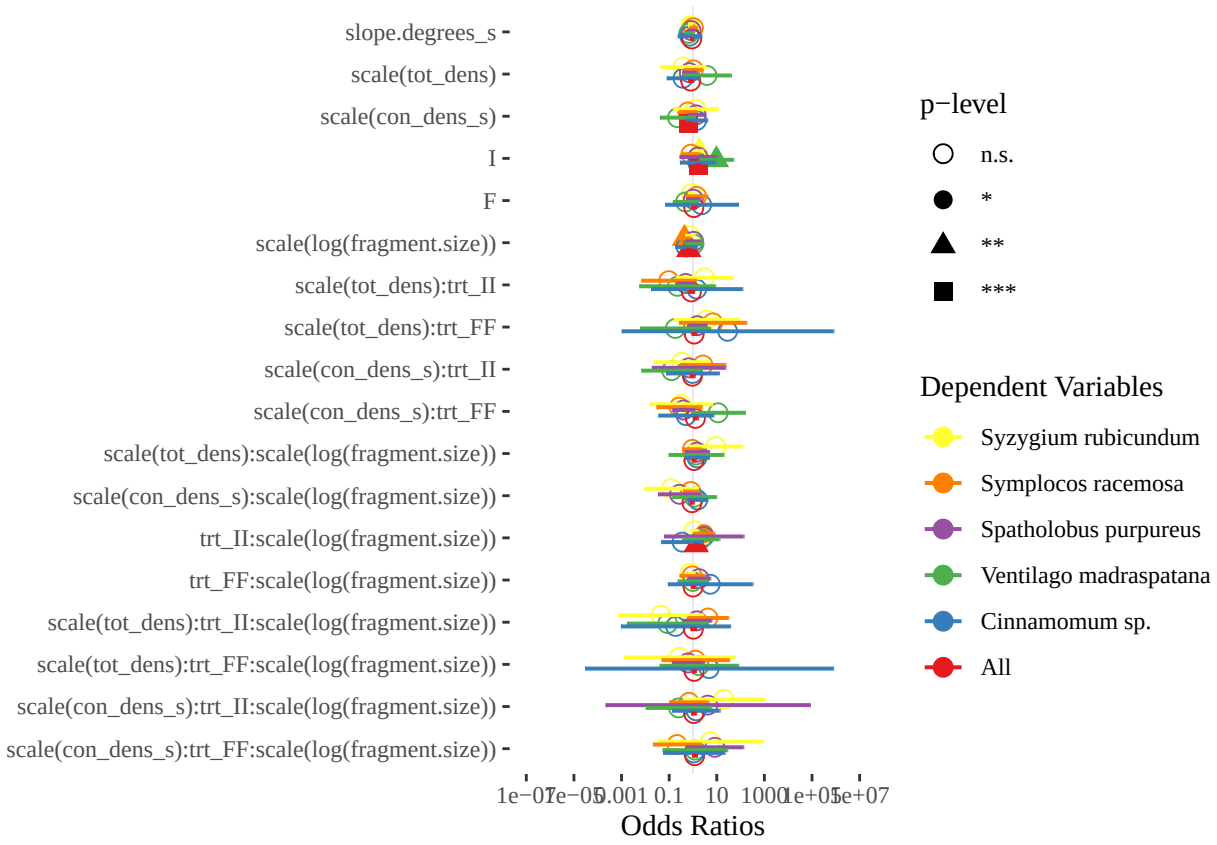
```
##  Family: binomial  ( logit )
## Formula:
## Pr_s ~ slope.degrees_s + (scale(tot_dens) + scale(con_dens_s)) *
##     (trt_I + trt_F) * scale(log(fragment.size)) + (1 | species) +
##     (1 | site/loc/gr/plot)
## Data: filter(sdls, species == i)
## Weights: census.start
##
##      AIC      BIC   logLik deviance df.resid
##    143.4    195.3    -47.7     95.4       40
##
## Random effects:
##
## Conditional model:
##  Groups           Name        Variance  Std.Dev.
##  species          (Intercept) 3.615e-14 1.901e-07
##  plot:gr:loc:site (Intercept) 2.882e-10 1.698e-05
##  gr:loc:site      (Intercept) 1.192e-14 1.092e-07
##  loc:site         (Intercept) 1.293e-14 1.137e-07
##  site             (Intercept) 1.187e-13 3.445e-07
## Number of obs: 64, groups:
## species, 1; plot:gr:loc:site, 64; gr:loc:site, 27; loc:site, 16; site, 10
##
## Conditional model:
##                                                  Estimate Std. Error z value
## (Intercept)                                       1.17286    0.41678   2.814
## slope.degrees_s                                  -0.40301    0.24721  -1.630
## scale(tot_dens)                                   1.35978    1.17608   1.156
## scale(con_dens_s)                                -1.50109    0.86550  -1.734
## trt_II                                            2.26471    0.83208   2.722
## trt_FF                                           -0.72767    0.61846  -1.177
## scale(log(fragment.size))                         0.08774    0.45545   0.193
## scale(tot_dens):trt_II                           -1.51287    1.87200  -0.808
## scale(tot_dens):trt_FF                           -1.71057    1.72319  -0.993
## scale(con_dens_s):trt_II                         -2.06232    1.49050  -1.384
## scale(con_dens_s):trt_FF                          2.43123    1.32627   1.833
## scale(tot_dens):scale(log(fragment.size))         0.31136    1.35724   0.229
## scale(con_dens_s):scale(log(fragment.size))       0.12988    1.08290   0.120
## trt_II:scale(log(fragment.size))                  0.77671    0.91299   0.851
## trt_FF:scale(log(fragment.size))                 -0.01336    0.74706  -0.018
## scale(tot_dens):trt_II:scale(log(fragment.size)) -2.45905    1.99499  -1.233
## scale(tot_dens):trt_FF:scale(log(fragment.size))  0.58715    1.94177   0.302
## scale(con_dens_s):trt_II:scale(log(fragment.size)) -1.39202    1.61694  -0.861
## scale(con_dens_s):trt_FF:scale(log(fragment.size))  0.19768    1.59299   0.124
##                                                  Pr(>|z|)
## (Intercept)                                       0.00489 **
## slope.degrees_s                                   0.10306
## scale(tot_dens)                                   0.24760
## scale(con_dens_s)                                 0.08285 .
## trt_II                                            0.00649 **
## trt_FF                                            0.23936
## scale(log(fragment.size))                         0.84725
## scale(tot_dens):trt_II                            0.41900
## scale(tot_dens):trt_FF                            0.32087
```

```
## scale(con_dens_s):trt_II                              0.16647
## scale(con_dens_s):trt_FF                              0.06678 .
## scale(tot_dens):scale(log(fragment.size))             0.81856
## scale(con_dens_s):scale(log(fragment.size))           0.90453
## trt_II:scale(log(fragment.size))                      0.39492
## trt_FF:scale(log(fragment.size))                      0.98573
## scale(tot_dens):trt_II:scale(log(fragment.size))      0.21772
## scale(tot_dens):trt_FF:scale(log(fragment.size))      0.76236
## scale(con_dens_s):trt_II:scale(log(fragment.size))    0.38930
## scale(con_dens_s):trt_FF:scale(log(fragment.size))    0.90124
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## $`Cinnamomum sp.`
##  Family: binomial  ( logit )
## Formula:
## Pr_s ~ slope.degrees_s + (scale(tot_dens) + scale(con_dens_s)) *
##     (trt_I + trt_F) * scale(log(fragment.size)) + (1 | species) +
##     (1 | site/loc/gr/plot)
## Data: filter(sdls, species == i)
## Weights: census.start
##
##      AIC      BIC   logLik deviance df.resid
##    162.9    221.8    -57.5    114.9       62
##
## Random effects:
##
## Conditional model:
##  Groups           Name        Variance  Std.Dev.
##  species          (Intercept) 6.917e-10 2.630e-05
##  plot:gr:loc:site (Intercept) 7.773e-11 8.816e-06
##  gr:loc:site      (Intercept) 2.035e+00 1.427e+00
##  loc:site         (Intercept) 3.525e-08 1.878e-04
##  site             (Intercept) 9.322e-01 9.655e-01
## Number of obs: 86, groups:
## species, 1; plot:gr:loc:site, 86; gr:loc:site, 42; loc:site, 23; site, 17
##
## Conditional model:
##                                                    Estimate Std. Error z value
## (Intercept)                                        -0.40851    0.58085  -0.703
## slope.degrees_s                                    -0.32176    0.58540  -0.550
## scale(tot_dens)                                    -0.97114    0.79709  -1.218
## scale(con_dens_s)                                   0.35924    0.52315   0.687
## trt_II                                              0.45861    0.87887   0.522
## trt_FF                                              0.84066    1.79322   0.469
## scale(log(fragment.size))                          -0.65747    0.54850  -1.199
## scale(tot_dens):trt_II                              0.35824    2.25319   0.159
## scale(tot_dens):trt_FF                              3.34970    5.22263   0.641
## scale(con_dens_s):trt_II                           -0.03752    1.29818  -0.029
## scale(con_dens_s):trt_FF                           -0.67189    1.35854  -0.495
## scale(tot_dens):scale(log(fragment.size))           0.38663    0.59626   0.648
## scale(con_dens_s):scale(log(fragment.size))         0.46242    0.47832   0.967
## trt_II:scale(log(fragment.size))                   -1.04864    1.03031  -1.018
## trt_FF:scale(log(fragment.size))                    1.67013    2.08919   0.799
```

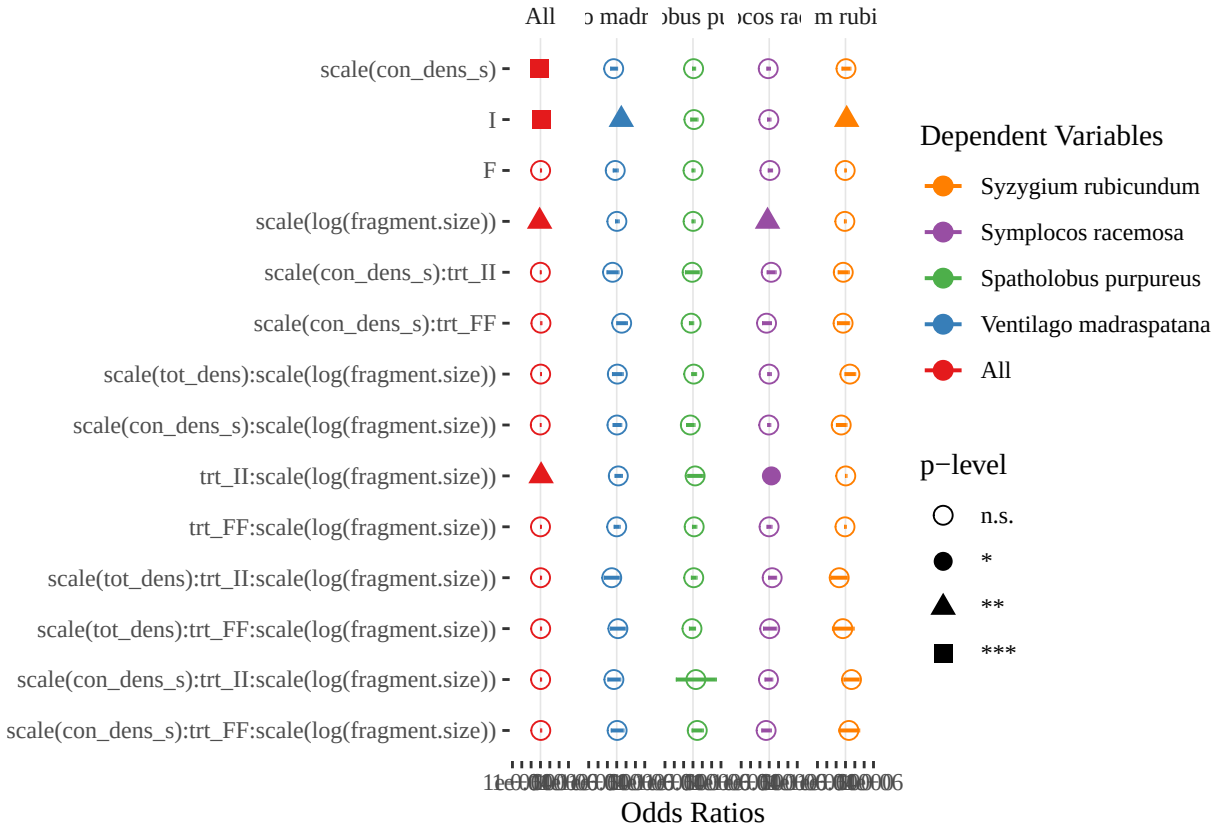
```
## scale(tot_dens):trt_II:scale(log(fragment.size))   -1.68922    2.69246  -0.627
## scale(tot_dens):trt_FF:scale(log(fragment.size))    1.56474    6.11823   0.256
## scale(con_dens_s):trt_II:scale(log(fragment.size))  0.29282    1.16878   0.250
## scale(con_dens_s):trt_FF:scale(log(fragment.size))  0.08156    1.51954   0.054
##                                                    Pr(>|z|)
## (Intercept)                                           0.482
## slope.degrees_s                                       0.583
## scale(tot_dens)                                       0.223
## scale(con_dens_s)                                     0.492
## trt_II                                                0.602
## trt_FF                                                0.639
## scale(log(fragment.size))                             0.231
## scale(tot_dens):trt_II                                0.874
## scale(tot_dens):trt_FF                                0.521
## scale(con_dens_s):trt_II                              0.977
## scale(con_dens_s):trt_FF                              0.621
## scale(tot_dens):scale(log(fragment.size))             0.517
## scale(con_dens_s):scale(log(fragment.size))           0.334
## trt_II:scale(log(fragment.size))                      0.309
## trt_FF:scale(log(fragment.size))                      0.424
## scale(tot_dens):trt_II:scale(log(fragment.size))      0.530
## scale(tot_dens):trt_FF:scale(log(fragment.size))      0.798
## scale(con_dens_s):trt_II:scale(log(fragment.size))    0.802
## scale(con_dens_s):trt_FF:scale(log(fragment.size))    0.957
##
## $All
##  Family: binomial  ( logit )
## Formula:
## Pr_s ~ slope.degrees_s + (scale(tot_dens) + scale(con_dens_s)) *
##     (trt_I + trt_F) * scale(log(fragment.size)) * (scale(con_dens_s) +
##     scale(tot_dens) || species) + (1 | site/loc/gr/plot)
## Data: sdls
## Weights: census.start
##
##      AIC      BIC   logLik deviance df.resid
##   2187.2   2313.9  -1067.6   2135.2      942
##
## Random effects:
##
## Conditional model:
##  Groups           Name              Variance  Std.Dev.  Corr
##  species          (Intercept)       1.135e+00 1.0653405
##                   scale(con_dens_s) 6.608e-02 0.2570611 0.00
##                   scale(tot_dens)   2.325e-02 0.1524655 0.00 0.00
##  plot:gr:loc:site (Intercept)       2.424e-01 0.4923436
##  gr:loc:site      (Intercept)       1.804e-01 0.4247101
##  loc:site         (Intercept)       3.926e-01 0.6266100
##  site             (Intercept)       2.861e-08 0.0001691
## Number of obs: 968, groups:
## species, 26; plot:gr:loc:site, 474; gr:loc:site, 110; loc:site, 37; site, 21
##
## Conditional model:
##                                                  Estimate Std. Error z value
## (Intercept)                                       0.69057    0.27999   2.466
```

```
## slope.degrees_s                                    -0.11658    0.06911  -1.687
## scale(tot_dens)                                    -0.21421    0.12004  -1.785
## scale(con_dens_s)                                  -0.40432    0.11910  -3.395
## trt_II                                              0.49115    0.11285   4.352
## trt_FF                                              0.08216    0.11073   0.742
## scale(log(fragment.size))                          -0.41120    0.13689  -3.004
## scale(tot_dens):trt_II                             -0.15738    0.13792  -1.141
## scale(tot_dens):trt_FF                              0.11644    0.12482   0.933
## scale(con_dens_s):trt_II                           -0.05721    0.11426  -0.501
## scale(con_dens_s):trt_FF                            0.23160    0.12231   1.894
## scale(tot_dens):scale(log(fragment.size))           0.09539    0.08419   1.133
## scale(con_dens_s):scale(log(fragment.size))        -0.09961    0.09165  -1.087
## trt_II:scale(log(fragment.size))                    0.31420    0.11084   2.835
## trt_FF:scale(log(fragment.size))                    0.00739    0.10952   0.067
## scale(tot_dens):trt_II:scale(log(fragment.size))    0.03831    0.13434   0.285
## scale(tot_dens):trt_FF:scale(log(fragment.size))    0.10500    0.10844   0.968
## scale(con_dens_s):trt_II:scale(log(fragment.size))  0.07438    0.12322   0.604
## scale(con_dens_s):trt_FF:scale(log(fragment.size))  0.16136    0.11441   1.410
##                                                    Pr(>|z|)
## (Intercept)                                        0.013646 *
## slope.degrees_s                                    0.091636 .
## scale(tot_dens)                                    0.074342 .
## scale(con_dens_s)                                  0.000686 ***
## trt_II                                             1.35e-05 ***
## trt_FF                                             0.458117
## scale(log(fragment.size))                          0.002666 **
## scale(tot_dens):trt_II                             0.253811
## scale(tot_dens):trt_FF                             0.350924
## scale(con_dens_s):trt_II                           0.616592
## scale(con_dens_s):trt_FF                           0.058277 .
## scale(tot_dens):scale(log(fragment.size))          0.257195
## scale(con_dens_s):scale(log(fragment.size))        0.277076
## trt_II:scale(log(fragment.size))                   0.004586 **
## trt_FF:scale(log(fragment.size))                   0.946205
## scale(tot_dens):trt_II:scale(log(fragment.size))   0.775502
## scale(tot_dens):trt_FF:scale(log(fragment.size))   0.332894
## scale(con_dens_s):trt_II:scale(log(fragment.size)) 0.546084
## scale(con_dens_s):trt_FF:scale(log(fragment.size)) 0.158447
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

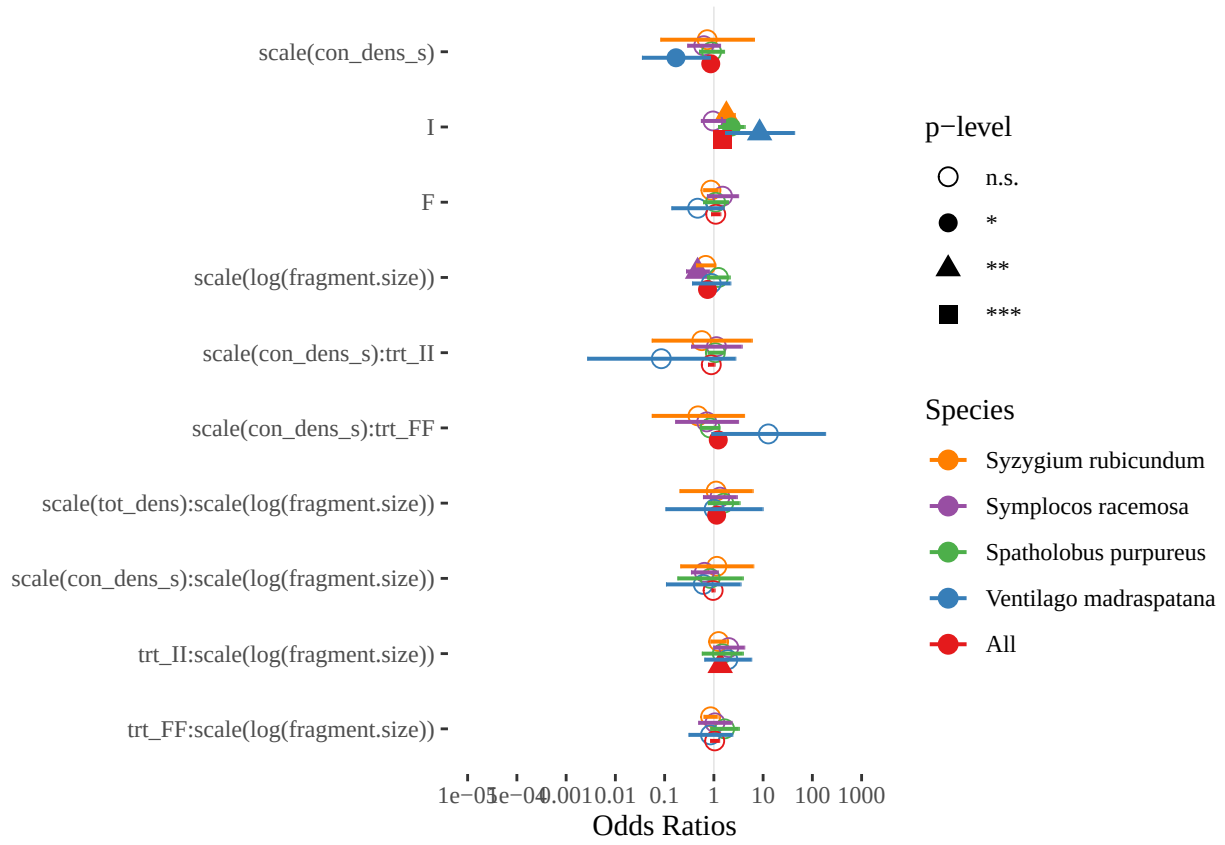
```
plot_models(single_sp_mods, m.labels=names(single_sp_mods), p.shape=TRUE)
```




```
## Cinnamonam distorts scale, so dropping it
plot_models(single_sp_mods[c(1:4, 6)],
            rm.terms = c("slope.degrees_s",
                         term_nms[str_detect(term_nms, "tot")]),
            m.labels=names(single_sp_mods)[c(1:4, 6)],
            p.shape = TRUE, grid=TRUE)
```




```
## perhaps too much complexity. Dropping 3-way interactions
##

m_cdd_s_ri_2way <- glmmTMB(Pr_s ~ slope.degrees_s +
                        (scale(tot_dens) + scale(con_dens_s)) *
                        (trt_I + trt_F) +
                        scale(log(fragment.size)) *
                          (scale(tot_dens) + scale(con_dens_s) + trt_I + trt_F) +
                        (1|site/loc/gr/plot),
                      weights = census.start, data = sdls,
                      family=binomial)

single_sp_mods <- map(sp_common[1:4], function(i) {
  update(m_cdd_s_ri_2way, data = filter(sdls, species == i))})
names(single_sp_mods) <- sp_codes$spbin[match(names(single_sp_mods), sp_codes$code)]
single_sp_mods$All <- m_cdd_s_ri_2way


plot_models(single_sp_mods,
            rm.terms = c("slope.degrees_s",
                         term_nms[str_detect(term_nms, "tot")]),
            m.labels=names(single_sp_mods),
            p.shape = TRUE, grid=FALSE)  + labs(colour = "Species")
```



To be honest, probably not enough data for individual species. Dropping the 3-way interactions helps fit the models, but only notable effects are density dependence in Ventilago and effects of fragment size on symplocus.

Session Information

```
sessionInfo()
```

```
## R version 4.4.0 (2024-04-24 ucrt)
## Platform: x86_64-w64-mingw32/x64
## Running under: Windows 10 x64 (build 17763)
##
## Matrix products: default
##
##
## locale:
## [1] LC_COLLATE=English_United States.1252
## [2] LC_CTYPE=English_United States.1252
## [3] LC_MONETARY=English_United States.1252
## [4] LC_NUMERIC=C
## [5] LC_TIME=English_United States.1252
##
## time zone: America/New_York
```

```
## tzcode source: internal
##
## attached base packages:
## [1] stats     graphics  grDevices utils     datasets  methods   base
##
## other attached packages:
##  [1] ggdist_3.3.2        patchwork_1.2.0     sjPlot_2.8.16
##  [4] ggeffects_1.6.0.2   broom.mixed_0.2.9.5 DHARMa_0.4.6
##  [7] glmmTMB_1.1.9       knitr_1.47          ggthemes_5.1.0
## [10] lubridate_1.9.3     forcats_1.0.0       stringr_1.5.1
## [13] dplyr_1.1.4         purrr_1.0.2         readr_2.1.5
## [16] tidyr_1.3.1         tibble_3.2.1        ggplot2_3.5.1
## [19] tidyverse_2.0.0
##
## loaded via a namespace (and not attached):
##   [1] Rdpack_2.6           rlang_1.1.3          magrittr_2.0.3
##   [4] furrr_0.3.1          compiler_4.4.0       mgcv_1.9-1
##   [7] vctrs_0.6.5          pkgconfig_2.0.3      crayon_1.5.2
##  [10] fastmap_1.2.0        backports_1.5.0      labeling_0.4.3
##  [13] pander_0.6.5         effectsize_0.8.8     utf8_1.2.4
##  [16] promises_1.3.0       rmarkdown_2.27       tzdb_0.4.0
##  [19] haven_2.5.4          nloptr_2.0.3         bit_4.0.5
##  [22] xfun_0.44            highr_0.11           later_1.3.2
##  [25] sjmisc_2.8.10        broom_1.0.6          parallel_4.4.0
##  [28] R6_2.5.1             gap.datasets_0.0.6   stringi_1.8.4
##  [31] qgam_1.3.4           RColorBrewer_1.1-3   parallelly_1.37.1
##  [34] car_3.1-2            boot_1.3-30          numDeriv_2016.8-1.1
##  [37] estimability_1.5.1   Rcpp_1.0.12          iterators_1.0.14
##  [40] parameters_0.21.7    httpuv_1.6.15        Matrix_1.7-0
##  [43] splines_4.4.0        timechange_0.3.0     tidyselect_1.2.1
##  [46] rstudioapi_0.16.0    abind_1.4-5          yaml_2.3.8
##  [49] doParallel_1.0.17    TMB_1.9.11           codetools_0.2-20
##  [52] sjlabelled_1.2.0     listenv_0.9.1        lattice_0.22-6
##  [55] plyr_1.8.9           shiny_1.8.1.1        withr_3.0.0
##  [58] bayestestR_0.13.2    coda_0.19-4.1        evaluate_0.23
##  [61] future_1.33.2        jtools_2.2.2         pillar_1.9.0
##  [64] gap_1.5-3            carData_3.0-5        foreach_1.5.2
##  [67] insight_0.19.11      distributional_0.4.0 generics_0.1.3
##  [70] vroom_1.6.5          hms_1.1.3            munsell_0.5.1
##  [73] scales_1.3.0         minqa_1.2.7          globals_0.16.3
##  [76] xtable_1.8-4         glue_1.7.0           emmeans_1.10.2
##  [79] tools_4.4.0          lme4_1.1-35.3        mvtnorm_1.2-5
##  [82] grid_4.4.0           rbibutils_2.2.16     datawizard_0.10.0
##  [85] colorspace_2.1-0     nlme_3.1-164         performance_0.11.0
##  [88] beeswarm_0.4.0       cli_3.6.2            fansi_1.0.6
##  [91] ggdensity_1.0.0      interactions_1.1.5   sjstats_0.19.0
##  [94] ggh4x_0.2.8          gtable_0.3.5         digest_0.6.35
##  [97] farver_2.1.2         htmltools_0.5.8.1    lifecycle_1.0.4
## [100] mime_0.12            bit64_4.0.5          MASS_7.3-60.2
```