

log-project-aubrie-winnie

2024-02-21

Understanding how spatial variation is linked to diversity maintenance in natural communities is a pillar of plant community ecology. Theoretically, a variable landscape can maintain diversity via niche partitioning: different species can trade off in performing better or worse depending on the conditions of the patch they are growing in, and as a result, more species can sustainably coexist in a community than if it were spatially heterogeneous. In the hyperdiverse system of native annual plants in Western Australia, fallen logs may be one of the greatest contributors to generating spatial variation that could help maintain species diversity. Considerable anecdotal evidence suggests that fallen logs generate spatial variation, or patchiness, in the environment (Figure 1), and that species or assemblages of plants may respond differently depending on if they are near logs or not. Despite such anecdotal evidence, it is yet unknown if and how fallen logs contribute to maintaining species diversity in the native annual plant communities of the Western Australian wheat belt.



Figure 1: Figure 1: image of annual plant halos around logs

The project will address the following questions:

- Q1) Are/how are plant communities in fallen log patches different from patches that are in the open?**
- Q2) Why are plant communities in fallen log patches different from patches in the open?**
- Q3) Are/how are plant species performances affected by proximity to fallen logs?**

Hypotheses

The null hypothesis, H0, is that annual plants in fallen log patches are not different in diversity, abundance, or composition from open patches.

In addition to the null hypothesis, the following constitute four, non-mutually exclusive hypotheses concerning how fallen logs may introduce spatial variation in the environment. I include corresponding predictions

for how plant communities may differ between fallen log patches as compared to open patches. **H1: Log decomposition creates islands of fertility directly around the fallen log.** Prediction 1: Nutrient composition around logs will be higher than in open plots

Prediction 2: Variations in nutrient composition in log vs open environments will correspond to variations in species composition, abundance, and/or richness in these environments.

Prediction 3: All sown plants will perform best in environments where organic logs have been left ‘insitu’. In locations where logs have been removed or replaced with pvc, the legacy of the nutrient island effect will yield higher sown plant performance than when compared to locations where logs have never been. The effect of the nutrient island in locations where logs have been added to open environments should yeild higher plant performance over time. *note: performance is measured in terms of germination rate, survival to fruiting, fecundity, and/or biomass.*

H2: Fallen logs alter the microclimate directly around them by providing shade. Prediction 1: Shade and temperature around logs vs in open plots will be different

Prediction 2: Variation in shade and temperature in log vs open environments will correspond to variation in species composition, abundance, and/or richness in these environments.

Prediction 3: All sown plants will perform best in environments where there are organic or pvc logs, no matter if they have been recently moved or not.

H3: Fallen logs trap dispersing seeds as they are blown along the ground.

Prediction 1: Dispersing seeds accumulate around logs, leading to a denser stand of plants in fallen log patches. Plant abundance in fallen log patches will be higher as compared to open patches. Rare plants will be more common in fallen log patches as compared to open patches

Prediction 2: All sown plants will perform the same in all experimental environments

H4: At least some species perform differently according to variation in log vs. open environments and have short dispersal kernels, causing fitness-density covariance



Figure 2: Photo before germination, after a rain. Notice the seeming wet halo under and around the branch

Experimental Design

In this experiment, 224 plots are arranged in 7 blocks of 32 plots each within the Caron Dam nature reserve. A map can be found here. *note: the location info for 3.02 is probably incorrect as of May 2022, and location info is currently unavailable for plots 6.25 and 7.19*

Each block is approximately 30m X 30m in area. Plots are 1m long and linear, and have a pin tag on either end (see Figure 3). The pin tags have the identity of the plot written on them in the form of “blocknumber.plotnumber”. Plots are 1m or more away from each other.

In each block, plot environments can be one of six types: - A 1m log that is out in the open (open_with_log, 4 plots) - A 1m log that is a part of a tree (insitu_log, 4 plots) - A 1m pvc pipe that is out in the open (open_with_pvc, 4 plots) - A 1m pvc pipe that is a part of a tree (insitu_pvc, 4 plots) - A plot that is out in the open (open, 8 plots) - A gap in a log where a log used to be (gap, 8 plots)

In half of the plots (not including open plots), the addition, exchange, or removal of logs or pvc to the environment was implemented in October 2020, before seed dispersal. In the other half of these plots, these manipulations were implemented after seed dispersal, in March 2021.

Within each 1m long plot, there is a ~20cm long microtransect. The ends of the microtransects are marked by a nail and a washer sunken into the ground. Each microtransect is approximately 21 cm in internal length from inner washer edge to inner washer edge. Microtransects are not sided.

In half of all plots, seeds were sown in March 2021 and February 2022. In these plots, 15 seeds each of *Trachymene ornata* (TROR), *Goodenia rosea* (GORO), and *Trachymene cyanopetala* (TRCY) are sown outside of the microtransects as in the diagram. These plants were selected because they represent plants common to communities next to logs (TROR), out in the open (GORO), or both (TRCY). The plots where seeds were sown are called ‘lambda’ plots as noted in Figure 3. In the dataset, the rows with a “1” in the ‘seeding_trt’ column are the plots that had seeds sown into them.

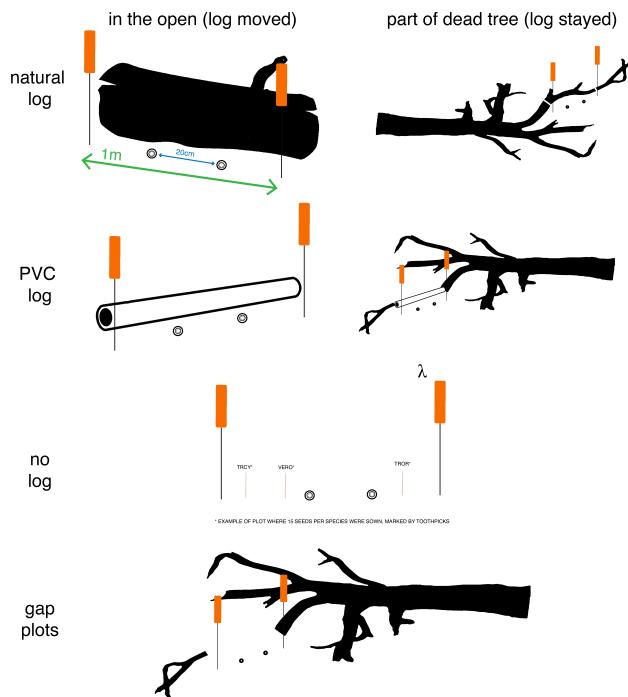


Figure 3: Figure 3: plot schematic

Datasets

The sets of data that we have collected for this arm of the project are the following.

- (1) Community data, before and after the experiment was implemented.

- Every year during peak biomass we surveyed plant communities at every centimeter along each micro-transect. Plant count and identity information is collected at each centimeter.
- These data are available from 2020 (before the experiment began), 2021 (one year into the experiment), and 2022 (two years into the experiment)

(2) Soil nutrient analysis in the open and insitu log plots.

- In 2022, we sampled 14 samples taken from one insitu_open plot and one insitu_log plot from each of the seven blocks. At each plot, three soil cores from random positions were collected by inserting a Hamilton tree planter into a depth of 5 cm. Particularly in insitu_log plots, soil cores were removed within a five-centimeter zone adjacent to the wood debris. Between each extraction, all tools were sterilised with 80% ethanol to prevent cross contamination. Soil cores from each plot were then mixed, cooled and immediately transported to a commercial laboratory (CSBP laboratories). Nutrient analysis was conducted at CSBP laboratories to determine soil pH, organic C, P, K, inorganic nitrogen (NO_3^- and NO_4^+) and basic exchangeable cations (Ca, Mg, Na and K). All basic exchangeable cations were summed together and labelled as “CEC”.

(3) Performance data of TROR, TRCY, and TROR.

- In 2021, the only performance data that were collected after sowing the experiment were the **total number of plants that came up and survived to fruiting for each species in each location, their total biomass, and their per capita biomass**. I calculated per capita biomass by dividing total biomass of the collected focal plants by the number of focal plants observed. This was because of logistical issues due to covid. In this dataset, there were two instances where the number of plants collected was greater than the number of seeds sown. Both instances were *T. cyanopetala*, where nplants = 16 and 18. For these two datapoints I chose to convert the count values to 15, assuming that every individual we planted came up, and that the extra were either naturally occurring seeds or that the number of seeds that we put into the ground was greater than 15 (human error). To calculate total biomass for these two instances, I divided the total biomass by the number of individuals observed, and multiplied by 15.
- In 2022, we went to the field early in the season and counted and thinned the number of germinated seedlings in each location. We therefore have a count for germination, but for the following reasons there are some issues with these data. The first is that we probably surveyed germination a little too early. The seedlings were often super small or hadn't come up yet. Because we couldn't come back later to re-thin the plots, we went ahead with counting and thinning seedlings. The values in the dataset for 2022 (nplants_data_2022.csv) corresponding to this germination survey are “ntrcy_germ”, “ngoro_germ” and “ntror_germ”. The second is that at the end of the growing season, Jake and Winnie came back and found that there was often more than one plant of the focal species where we seeded them, and sometimes there was one or more focal plants that popped up where we had not observed germination earlier in the season. We assume that these plants come from the seeds we planted, and that they came up later than our initial germination survey. The number of plants observed and collected at each location at the end of the season are in the columns “nplants_tror”, “nplants_goro” and “nplants_trcy” in the nplants_data_2022.csv data file. Because ngerm and nplants don't totally capture what came up where we sowed seed, I chose to analyze the total number of observed plants. I calculated this as the number of total plants we observed between the germination and the end of the season, being careful not to double-count the individual that was left after thinning from the first round of germination survey. The performance data I analyze here are **total number of plants that came up for each species in each location, and the per capita biomass of plants collected at the end of the season**.

Methods and analysis

Q1 Are/how are plant communities in fallen log patches different from patches that are in the open?

Overview of results 2020-2022

There are three response variables: count (Poisson), Shannon diversity index (hurdle: binomial and Gaussian), and composition matrix.

1. *Abundance* – the abundance of plants in log patches and open patches that are similar (treatment: $p > 0.05$). The presented model has a structure of “count ~ treatment(log/open) + (1|block) + (1|year)”.
2. *Diversity* – the probability of having a positive Shannon diversity index in the plant community is lower in open patches by 6.04% (treatment: $p = 0.03$). The plant diversity, as measured by the Shannon diversity index, is higher in log patches by 0.09 units (treatment: $p = 0.02$). The presented model has a structure of “diversity ~ treatment(log/open) + (1|block) + (1|year)”.
3. *Composition* – the five-dimensional NMDS ordination has stress values of 9%, which suggests a fairly good fit for the ordinations. The selected two-dimensional NMDS plot can be separated into two groups that overlap - fallen log patches and open patches. Based on the plant species compositions, the open patches consistently have a higher score than fallen log patches along NMDS axes 1 across three years of analysis. This can be interpreted as a turnover in species diversity at the scale of the plot type. [But NMDS axes do not consistently mean the same thing for each year?] Ordinations of plots sampled from the same block are typically close to each other in nondimensional space, reflecting the similarity of plant communities within the same block irrespective of plot type. This can be interpreted as a turnover in species diversity on the scale of the block. According to a partial RDA analysis on the aggregated data, Block and Year accounted for 44.3% of variations in species composition. Plot type (log / open) accounted for about 6% of variations in species composition.

2020

There are three response variables: count (Poisson), Shannon diversity index (hurdle: binomial and Gaussian), and composition matrix.

1. *Abundance* – the abundance of plants in log patches and open patches that are similar (treatment: $p > 0.05$). The presented model has a structure of “count ~ treatment(log/open) + (1|block)”.
2. *Diversity* – the plant diversity and the probability of having a positive Shannon diversity index in the plant community between log and open patches are similar (treatment: $p > 0.05$). The presented model has a structure of “diversity ~ treatment(log/open) + (1|block)”.
3. *Composition* – the four-dimensional NMDS ordination has stress values of 4.8%, which suggests a very good fit for the ordinations. The selected two-dimensional NMDS plot can be separated into two groups that overlap - fallen log patches and open patches. This can be interpreted as a turnover in species diversity at the scale of the plot type. Ordinations of plots sampled from the same block are typically close to each other in nondimensional space, reflecting the similarity of plant communities within the same block irrespective of plot type. This can be interpreted as a turnover in species diversity on the scale of the block. According to a partial RDA analysis, Block accounted for 69% of variation in species composition. Plot type (log / open) accounted for about 8% (2020) of variations in species composition.

2021

There are three response variables: count (Poisson), Shannon diversity index (hurdle: binomial and Gaussian), and composition matrix.

1. *Abundance* – the abundance of plants in log patches and open patches that are similar (treatment: $p > 0.05$). The presented model has a structure of “count ~ treatment(log/open) + (1|block)”.

2. *Diversity* – the plant diversity and the probability of having a positive Shannon diversity index in the plant community between log and open patches are similar (treatment: $p > 0.05$). The presented model has a structure of “diversity ~ treatment(log/open) + (1|block)”.
3. *Composition* – The five-dimensional NMDS ordination has stress values of 4.1%, which suggests a very good fit for the ordinations. The selected two-dimensional NMDS plot can be separated into two groups that overlap - fallen log patches and open patches. This can be interpreted as a turnover in species diversity at the scale of the plot type. Ordinations of plots sampled from the same block are typically close to each other in nondimensional space, reflecting the similarity of plant communities within the same block irrespective of plot type. This can be interpreted as a turnover in species diversity on the scale of the block. According to a partial RDA analysis, Block accounted for 53.7% of variation in species composition. Plot type (log / open) accounted for about 12.1% (2021) of variations in species composition.

2022

There are three response variables: count (Poisson), Shannon diversity index (hurdle: binomial and Gaussian), and composition matrix.

1. *Abundance* – the abundance of plants in log patches and open patches that are similar (treatment: $p > 0.05$). The presented model has a structure of “count ~ treatment(log/open) + (1|block)”.
2. *Diversity* – the probability of having a positive Shannon diversity index in the plant community between log and open patches are similar (treatment: $p > 0.05$). The plant diversity, as measured by the Shannon diversity index, higher in log patches by 0.24 units (treatment: $p = 0.007$). The presented model has a structure of “diversity ~ treatment(log/open) + (1|block)”.
3. *Composition* – The five-dimensional NMDS ordination has stress values of 4%, which suggests a very good fit for the ordinations. The selected two-dimensional NMDS plot can be separated into two groups that overlap - fallen log patches and open patches. This can be interpreted as a turnover in species diversity at the scale of the plot type. Ordinations of plots sampled from the same block are typically close to each other in nondimensional space, reflecting the similarity of plant communities within the same block irrespective of plot type. This can be interpreted as a turnover in species diversity on the scale of the block. According to a partial RDA analysis, Block accounted for 55.7% of variation in species composition. Plot type (log / open) accounted for about 11.4% (2022) of variations in species composition.

Statistical Methods

1. Abundance comparison

Comparison of plant abundance between log patches and open patches involves two types of data: (1) plant counts of all plots in 2020 before the experiment setup differentiated by the initial treatment “log” and “open”; and (2) plant counts from insitu_log and insitu_open plots in 2021 and 2022 respectively. All count data used for abundance comparison includes species with unknown identity. We used a generalized linear mixed-effects model (GLMER) to model the abundance of plant individuals (2020, 2021 and 2022 data) in unaltered fallen log patches and open patches using a Poisson distribution structure with block as random terms. When pulling count data from 2020 to 2022 together, we included time as a random term to avoid pseudo-replication.

2. Diversity comparison

A 30-centimeter linear transect was set up for each plot in 2020. We recorded species identity and individual counts for all species occurring along the transects in 2020, 2021 and 2022 respectively. Comparison of plant diversity between log patches and open patches involves two types of data: (1) plant count for each species of all plots in 2020 before the experiment setup differentiated by the initial treatment “log” and “open”; and

(2) plant count for each species from insitu_log and insitu_open plots in 2021 and 2022 respectively. All data used for diversity comparison excludes species with unknown identity.

We used the Shannon diversity index to calculate the plant species diversity for each transect at the plot level. A considerable number of zeros are generated in the computation of the Shannon diversity index. We first tried modelling diversity data with a linear mixed-effect model (LME) assuming a Gaussian distribution. The resulting model was zero-inflated as testing with DHARMA.

As a solution, we used a two-component hurdle model on the truncated data set where the response diversity index was separated into zero and non-zero values. In the first component, we used a binomial GLMER to model the probability of zeros and non-zeros in the diversity index with blocks as random terms. For the second component, we assumed an approximate Gaussian distribution of the non-zero Shannon index and used a linear mixed-effects regression (LME) to model the response non-zero diversity index with treatment as a fixed effect (log vs open) and block as a random factor. Specifically, for the hurdle model on composite diversity data (2020 to 2022), we also included year as a random factor.

3. Composition

Comparison of plant composition between log patches and open patches involves two types of data: (1) plant count for each species of all plots in 2020 before the experiment setup differentiated by the initial treatment “log” and “open”; and (2) plant count for each species from insitu_log and insitu_open plots in 2021 and 2022 respectively. All data used for diversity comparison excludes species with unknown identity. For transects where no plants were found, an artificial species, “x,” was added to the species composition matrix to represent zero plants.

We used Non-Metric Dimensional Scaling (NMDS) to analyse the differences in plant communities with Bray-Curtis dissimilarity metrics. We retained two dimensions in the NMDS ordination plot for the visualization and interpretation of plant community composition differences between plot types and blocks.

To identify the contribution of each species to the compositional changes amongst different sample plots, we extracted ordination scores for each species along NMDS axes 1 and 2. The species scores represent the weighted average of a species’ abundance score in a sample community along our selected NMDS axes. We also ran a partial Redundancy Analysis (RDA) to test the correlation between species composition (squared-root transformed) and plot type (log vs open patch) with block as a random term.

Analysis

Abundance analysis ** 2020 - 2022 **

```
knitr::opts_chunk$set(echo = TRUE)
# packages

require(vegan)
require(dplyr)
require(tidyr)
require(labdsrv)
require(stringr)
require(ggplot2)
require(ggrepel)
require(lme4)
require(emmeans)
require(ggpubr)
# require(Matrix)
# require(see) #doing this for aubrie because of some weird thing that happened when I installed new ver
```

```

##### data wrangling #####
# This dataset includes unknown species.
comm <- read.csv("20-22_species_composition_data_w_unk.csv", header=T)

# Remove the locations surveyed in 2021 and 2022 that were not surveyed in 2020 - aka, cm=0, cm=21, cm=22
comm<-comm[which(comm$cm_location!=21 & comm$cm_location!=0 & comm$cm_location!=22 & comm$cm_location!=23),]

# make a group name for each row
comm$grp<-apply(comm[c(1,3,5,6,7)], 1, paste, collapse=":") # timepoint, block, transect_name, initial

# Need to make each row a community using matrify.
commsub<-comm[,c(15,10,13)] # group, species_code, and count of each species for each transect. transect
commsub <-as.data.frame(commsub)
commtry<-matrify(commsub) # make it an expanded species matrix
commtry$x[which(is.na(commtry$x))] <- 1 # "x" means that there were no individuals in the transect, but
ncol(commtry) # how many species are we working with in our community matrix

# Store grouping row names as a column, then remove rownames.
commtry$grps<-rownames(commtry)
rownames(commtry)<-NULL
names(commtry)

# Split group info into columns for each variable.
mat<-separate(commtry, 107, c("time","block","transect","init", "treatment"), ":")
names(mat) #check

# Add groupname using time, block, init columns.
mat$grp1<-apply(mat[c(107:110)], 1, paste, collapse=":")
mat$grp2<-apply(mat[c(107:111)], 1, paste, collapse=":")
names(mat) #check

# Another df where the grouping variables are time, block, transect and initial state.
# Each row is a transect in a certain year.
df<-mat[,c(1:106,113)]
# df2 = with treatment in the grouping
df2 = df %>% mutate(across(.cols=1:106,.fns=as.numeric))
rownames(df2)<-NULL # remove rownames

##### Abundance analysis for 2020 (t0) AND in-situ log and in-situ open plots in 2021 (t1) and 2022 (t2)

# Sum observations across initial X transect X time X block X treatment (group variable).
# This gives number of plants in each row observation (transect level).
blocksum2<-rowsum(df2[,c(1:106)], group=df2$grp2)
blocksum2$grps<-rownames(blocksum2)
rownames(blocksum2)<-NULL # remove rownames

# Add in group vars.
nublock2<-separate(blocksum2, 107, c("time","block","transect","init", "treatment"), ":")
nublock2$total<-rowSums(nublock2[,c(1:106)])
nublock2$presence<-ifelse(nublock2$total > 0, 1, 0)

# Subset data where before treatments installed (t0), in-situ log and in-situ open from t1 and t2 are in

```

```

# Hence only absolute log effect and absolute open effect are concerned
dat_t0_insitu<-nublock2[which(nublock2$time=="t0" | nublock2$treatment=="open" | nublock2$treatment==

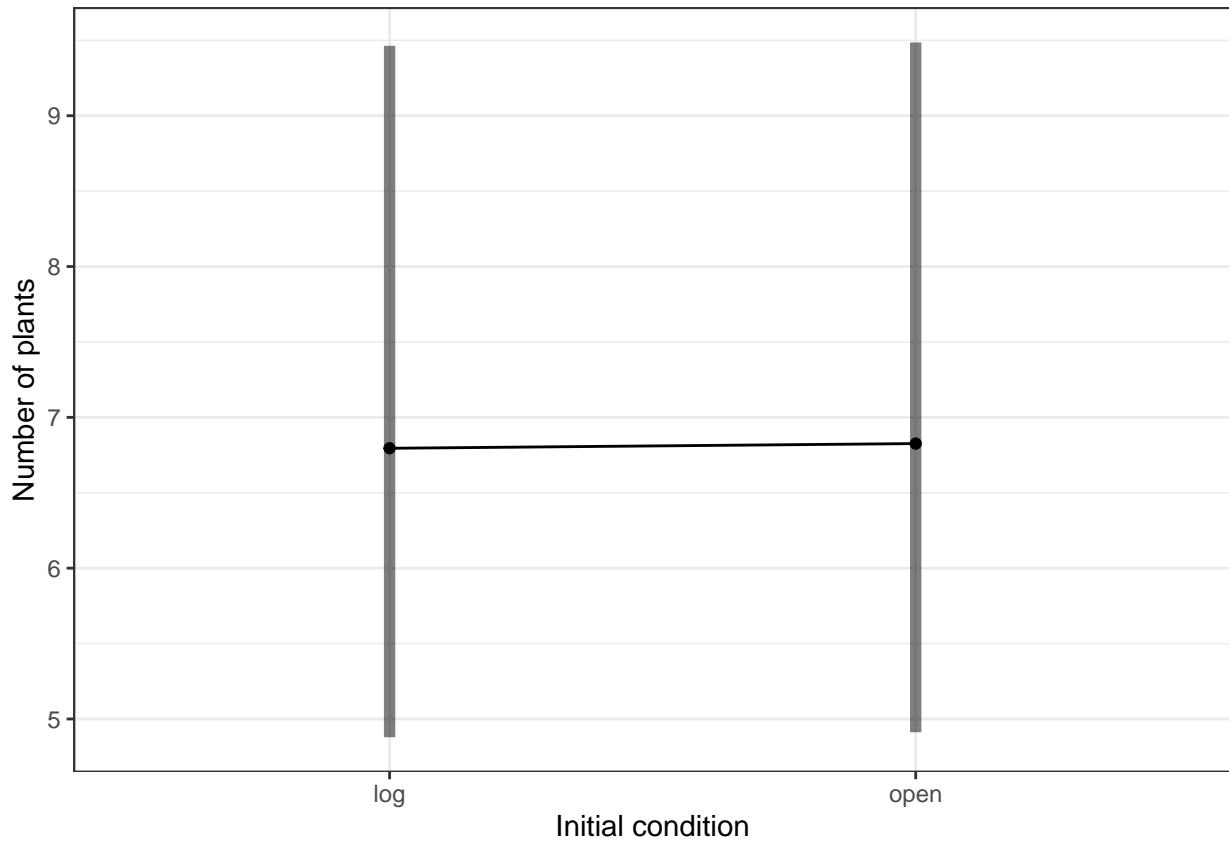
# look at plant abundance in log vs open
# look at range of data - what family should i use?
range(dat_t0_insitu$total)

# since samples from t1 and t2 from are mainly from the same plots as t0. These are not independent replicates
# including year and block as random effects
# (1/year) + (1/block) as we think the effect of init on total is the same (slope) but the intercept different
# but since we have a greater number of sample from t0, data from t0 will contribute more to the variance
# see https://bookdown.org/steve_midway/DAR/random-effects.html#pld-example
abun.mod<- glmer(total ~ init + (1|block) + (1|time), data=dat_t0_insitu, family = 'poisson')
summary(abun.mod)

emmeans(abun.mod, ~init, type='response') # plant abundance in log plots is higher by 0.04 plants - which is not significant

abun.plot <- emmip(abun.mod, ~init, type='response', CI=T)+theme_bw() + labs(x="Initial condition", y="Number of plants")
print(abun.plot)

```



Reason: why we need to include 'time' as random effect?

(1) Samples from t1 and t2 from are from the same plots as t0. These are not independent replicates as they share the same temporal context.
We included year and block as random effects to alleviate the effect of temporal pseudo-replication.

```

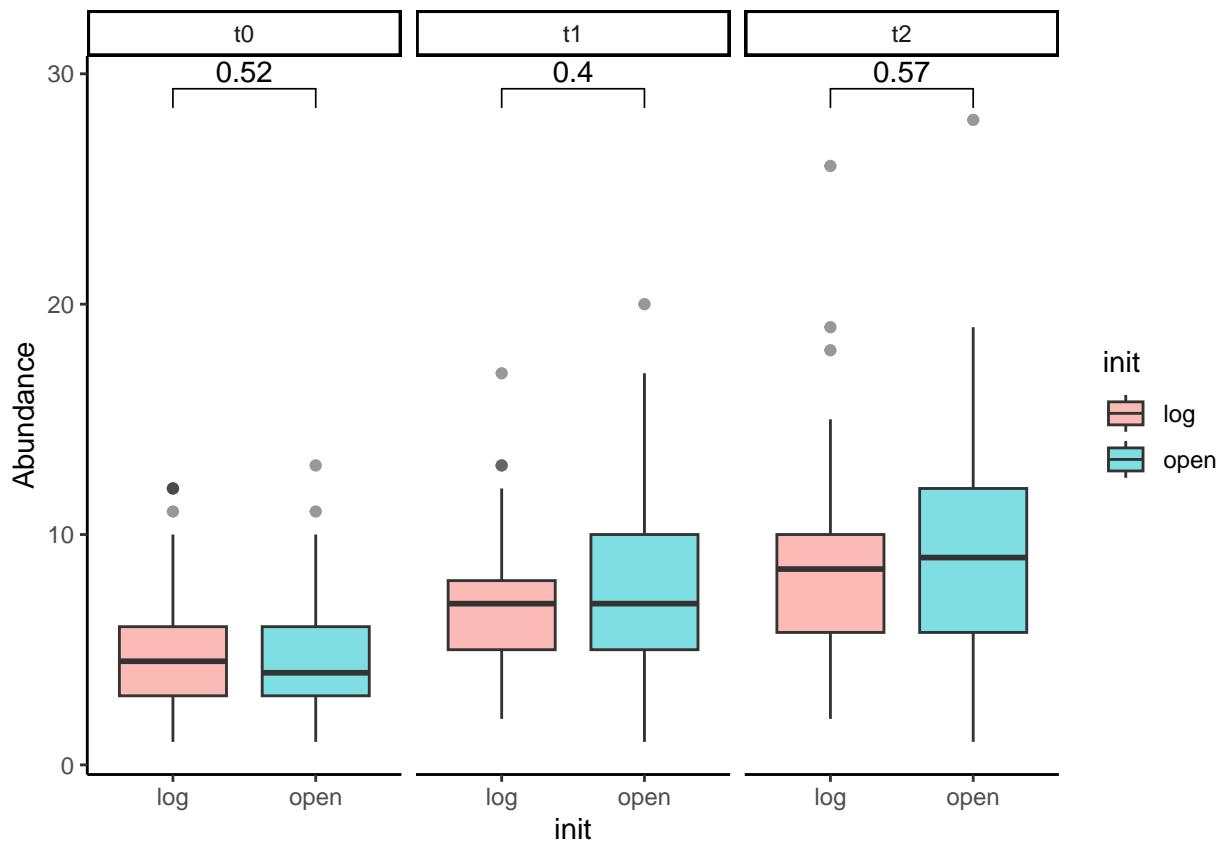
summary(aov(total ~ time, data = dat_t0_insitu)) # plant abundance differs between years

# (2) When we stratify the data by year, significance of the difference between log and open treatment
# if the effect of log-open (slope) is the same across year, year only affecting the intercept.
# since the slopes are similar, we include time as random intercept effect.
summary(glm(total ~ init, data = subset(dat_t0_insitu, time == "t0")), family = poisson)
summary(glm(total ~ init, data = subset(dat_t0_insitu, time == "t1")), family = poisson)
summary(glm(total ~ init, data = subset(dat_t0_insitu, time == "t2")), family = poisson)

abun.plot.year <- ggplot(dat_t0_insitu, aes(x = as.factor(init), y = total, fill = as.factor(init))) +
  geom_boxplot(position = "dodge", alpha = 0.5) +
  labs(x = "init",
       y = "Abundance") +
  scale_fill_discrete(name = 'init') +
  theme_classic() +
  facet_grid(. ~ time) +
  stat_compare_means(comparisons = list(c("log", "open"))), label = "p.format", method="wilcox.test")

print(abun.plot.year)

```



```

require(vegan)
require(dplyr)

```

```

require(tidyr)
require(labdsrv)
require(stringr)
require(ggplot2)
require(ggrepel)
require(lme4)
require(emmeans)
require(lmerTest)
require(performance)
require(ggpubr)
require(DHARMa)
require(patchwork)

# This dataset does not include data where the plant identity is unknown.
comm <- read.csv("20-22_species_composition_data_no_unk.csv", header = T)

# Remove the locations surveyed in 2021 and 2022 that were not surveyed in 2020 - aka, cm=0, cm=21, cm=22
comm<-comm[which(comm$cm_location!=21 & comm$cm_location!=0 & comm$cm_location!=22 & comm$cm_location!=23),]

# make a group name for each row
comm$grp<-apply(comm[c(1,3,5,6,7)], 1, paste, collapse=":") # timepoint, block, transect_name, initial

# Need to make each row a community using matrify
commsub<-comm[,c(15,10,13)]

# group, species_code, and count of each species for each transect. transects are rows.
commsub <-as.data.frame(commsub)
commtry<-labdsrv::matrify(commsub) # make it an expanded species matrix
commtry$x[which(is.na(commtry$x))] <- 1 # "x" means that there were no individuals in the transect, but we still want to keep the column
# ncol(commtry) # how many species are we working with in our community matrix

# Store grouping row names as a column, then remove rownames.
commtry$grps<-rownames(commtry)
rownames(commtry)<-NULL
# names(commtry)

# Split group info into columns for each variable
mat<-separate(commtry, 88, c("time","block","transect","init","treatment"), ":")
# names(mat) #check

# Add groupname using time, block, init columns
mat$grp<-apply(mat[c(88:92)], 1, paste, collapse=":")
# ynames(mat) #check

# Another df where the grouping variables are time, block, transect and initial state.
# Each row is a transect in a certain year.
df<-mat[,c(1:87,93)]
df2 = df %>% mutate(across(.cols=1:87,.fns=as.numeric)) # make everything numeric
rownames(df2)<-NULL # remove rownames

##### Species diversity analysis for 2020 - 2022 data (Shannon diversity on transect level)
numat = mat %>% mutate(across(.cols=1:87,.fns=as.numeric)) # make everything numeric

```

```

# sum species for each group (grouped by init, transect, block, time)
nudat<-numat%>%
  group_by(time, block, transect, init, treatment) %>% summarise(across(where(is.numeric), sum))

# make a data frame
dat<-as.data.frame(nudat) # this df contains transect levels from all years
dat<-dat[which(dat$time=="t0" | dat$treatment=="open" | dat$treatment=="insitu_log"),]

#estimate diversity for each row/group. don't include 'x'

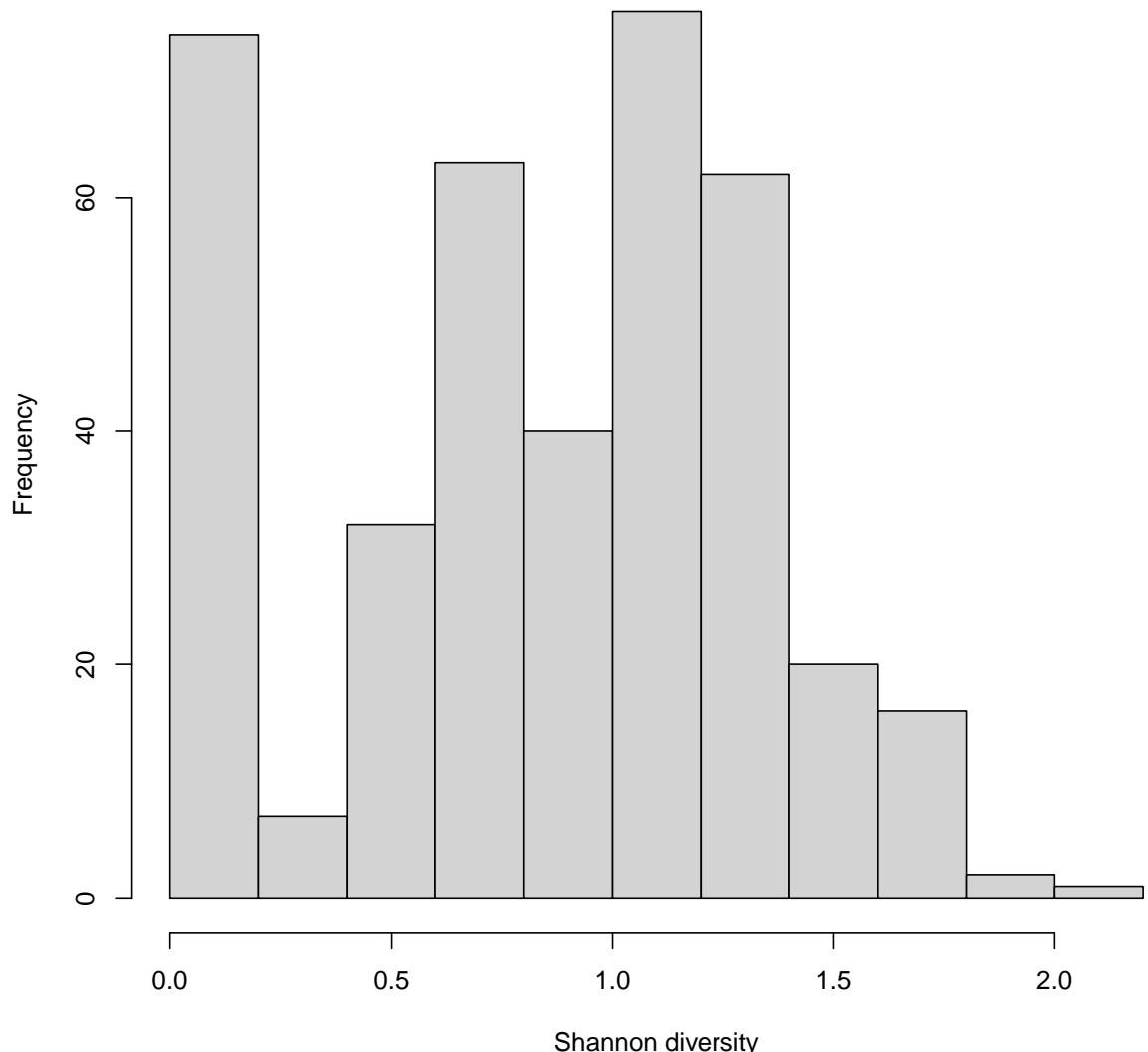
# no groups, just estimate diversity of each row
est<-dat[,c(6:91)]
dat$diversity<-diversity(est, index='shannon')

##### We used a hurdle model since the data is zero-inflated
# separate into zero and non-zero observations
dat$non_zero <- ifelse(dat$diversity > 0, 1, 0)

# what does our data look like?
hist(dat$diversity, xlab = "Shannon diversity", main = "Histogram of Shannon diversity index (all years")

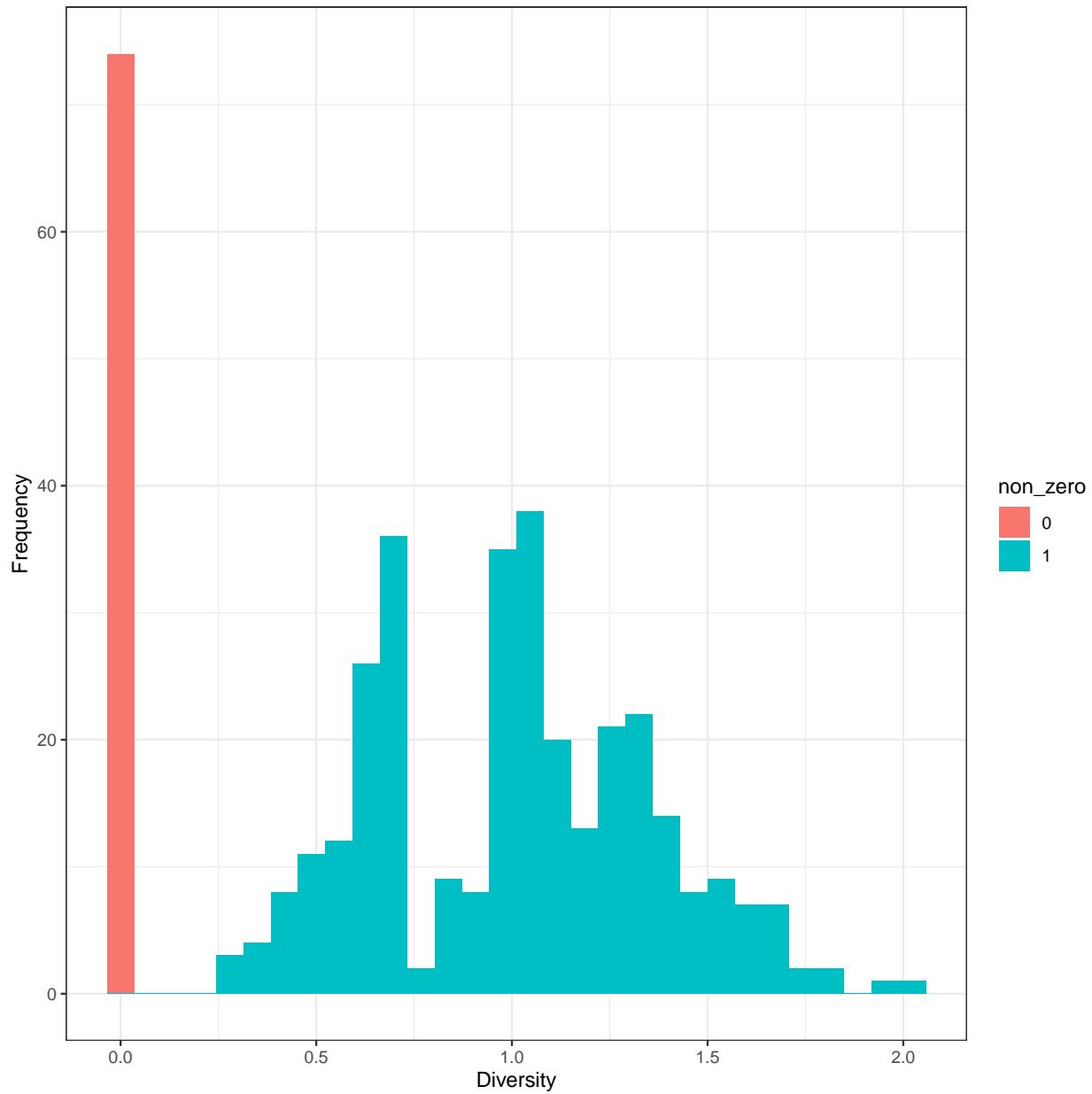
```

Histogram of Shannon diversity index (all years, transect level)



Diversity analysis

```
ggplot(dat, aes(x = diversity, fill = as.factor(non_zero))) +  
  geom_histogram(position = "identity", alpha = 1, bins = 30) +  
  labs(x = "Diversity",  
       y = "Frequency") +  
  scale_fill_discrete(name = 'non_zero') +  
  theme_bw()
```

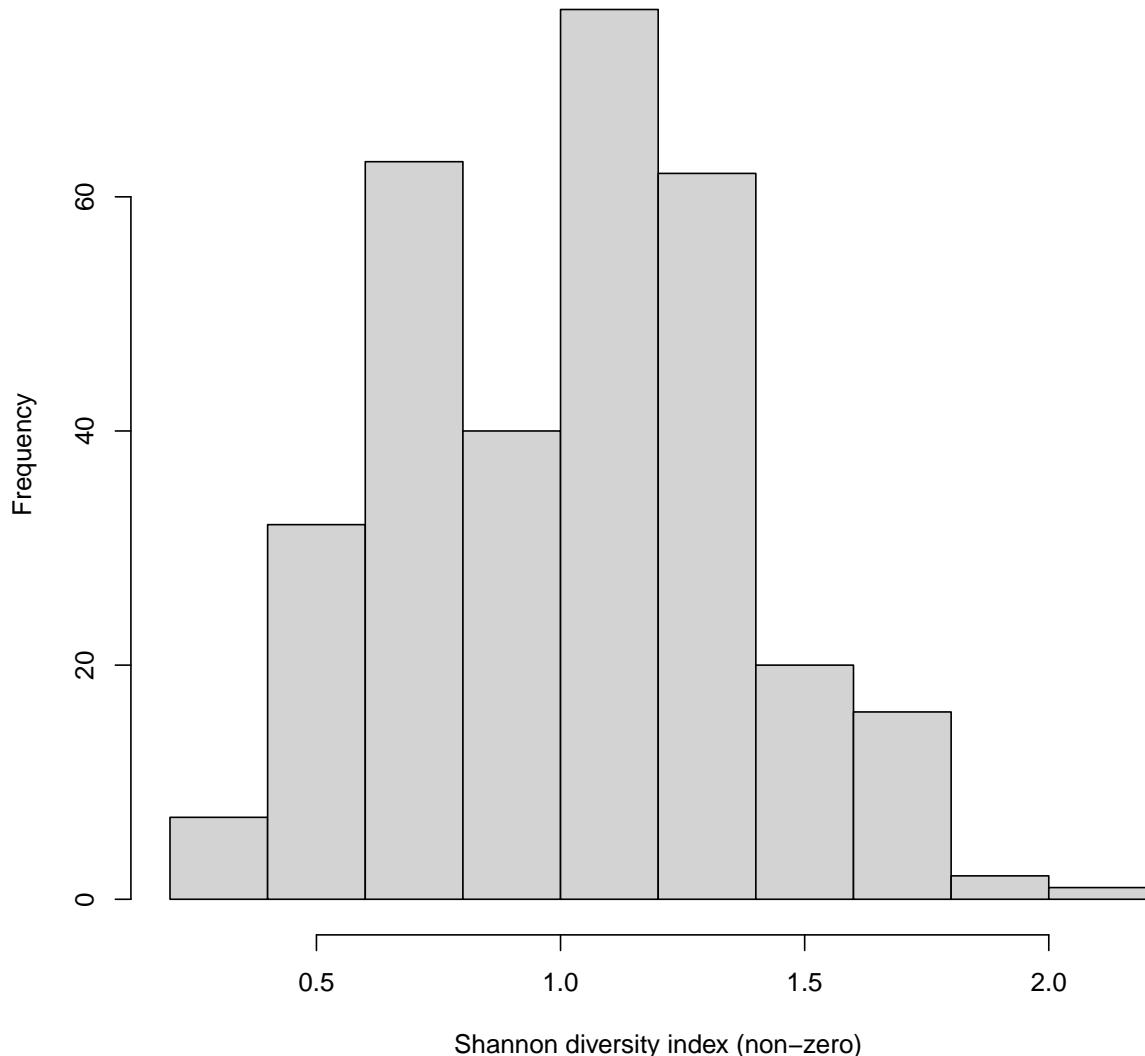


```
shapiro.test((dat$diversity[dat$non_zero == 1])) # non-zero data is not normal
```

```
##
## Shapiro-Wilk normality test
##
## data: (dat$diversity[dat$non_zero == 1])
## W = 0.98023, p-value = 0.0002215
```

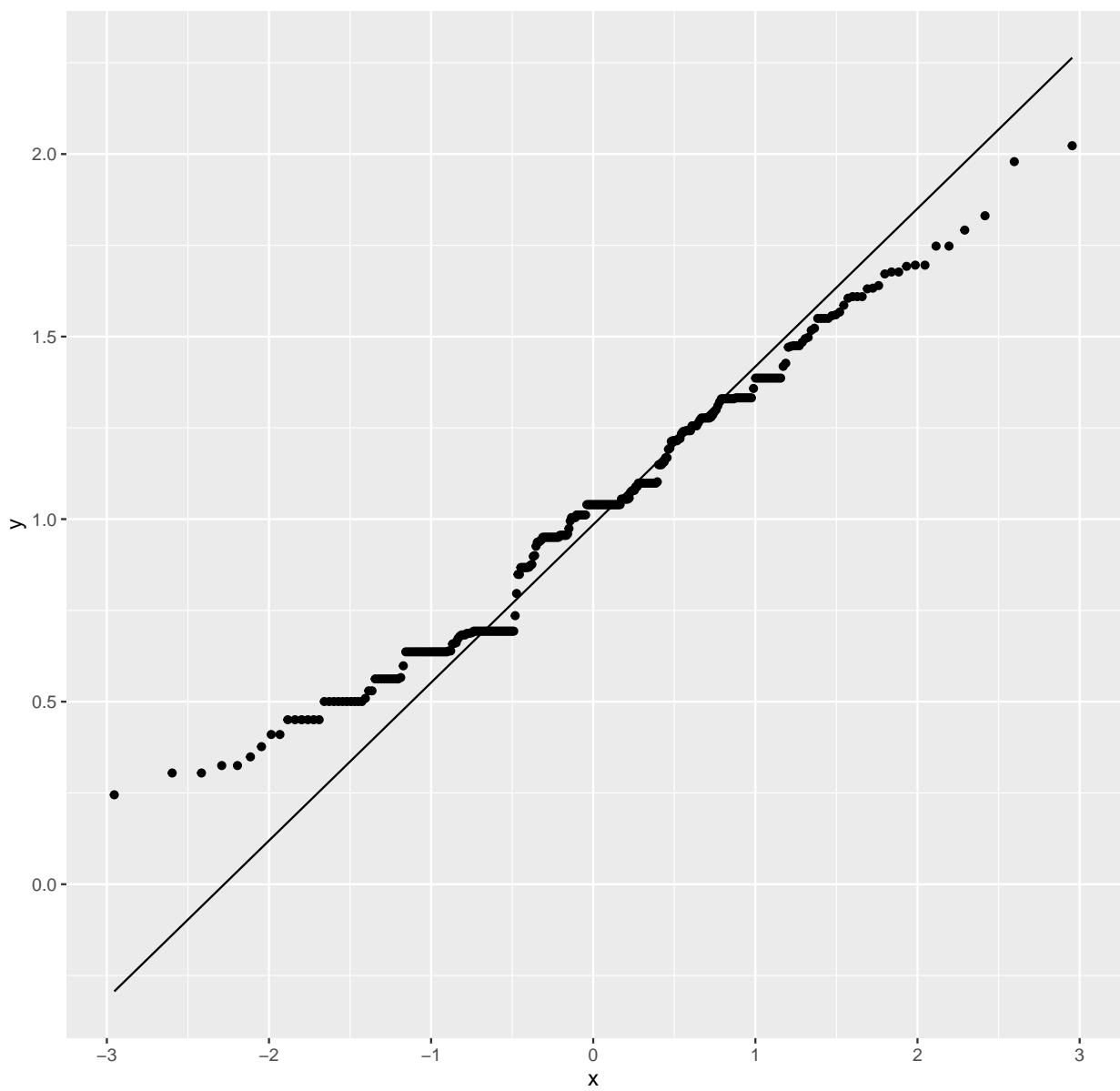
```
# Non-zero data is, however, an approximate Gaussian distribution
hist(dat$diversity[dat$non_zero == 1], xlab = "Shannon diversity index (non-zero)", main = "Histogram of
```

Histogram of Shannon diversity index (non-zero, all years, transect level)



```
# Gaussian distribution produced the best-fitting ggplot
dat_nonzerodat <- dat[dat$non_zero == 1, ]
ggplot(dat_nonzerodat, aes(sample = diversity)) +
  geom_qq(distribution = qnorm) +
  geom_qq_line(distribution = qnorm) +
  ggtitle("normal")
```

normal



```
# Hurdle model part 1. Logistic regression to predict the probability of non-zero.  
# Ppen and log environment does determine the probability of non-zero diversity index  
# 0 Shannon diversity index =/= zero plants  
Hurd.mod.1 <- glmer(non_zero ~ init + (1|block) + (1|time), data = dat, family = binomial)  
summary(Hurd.mod.1) # significant
```

```
## Generalized linear mixed model fit by maximum likelihood (Laplace  
## Approximation) [glmerMod]  
## Family: binomial ( logit )  
## Formula: non_zero ~ init + (1 | block) + (1 | time)  
## Data: dat  
##  
##      AIC      BIC  logLik deviance df.resid
```

```

##      371.1     387.0    -181.6     363.1      389
##
## Scaled residuals:
##      Min       1Q   Median       3Q      Max
## -3.8397   0.2618   0.3840   0.5265   0.8181
##
## Random effects:
## Groups Name        Variance Std.Dev.
## block  (Intercept) 0.1157   0.3401
## time   (Intercept) 0.3970   0.6301
## Number of obs: 393, groups: block, 7; time, 3
##
## Fixed effects:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  2.2440    0.4716   4.758 1.95e-06 ***
## initopen     -0.6040    0.2787  -2.168   0.0302 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##          (Intr)
## initopen -0.420

```

```

# Hurdle model part 1. Linear mixed effect model to predict the mean of the non-zero data.
# We included block and time as random terms.
# Open and log environment is significant in explaining the plant diversity in transect levels.
# Lmer was used because the distribution of non-zero data is approximately normal.
Hurd.mod.2 <- lmer(diversity ~ init + (1|block)+(1|time), data=subset(dat, non_zero == 1))
summary(Hurd.mod.2)

```

```

## Linear mixed model fit by REML. t-tests use Satterthwaite's method [
## lmerModLmerTest]
## Formula: diversity ~ init + (1 | block) + (1 | time)
## Data: subset(dat, non_zero == 1)
##
## REML criterion at convergence: 242.5
##
## Scaled residuals:
##      Min       1Q   Median       3Q      Max
## -2.12842 -0.89782  0.00211   0.68935   2.70325
##
## Random effects:
## Groups Name        Variance Std.Dev.
## block  (Intercept) 0.002110 0.04593
## time   (Intercept) 0.003245 0.05696
## Residual           0.119527 0.34573
## Number of obs: 319, groups: block, 7; time, 3
##
## Fixed effects:
##             Estimate Std. Error      df t value Pr(>|t|)
## (Intercept)  1.07797    0.04800  4.68976 22.46 5.91e-06 ***
## initopen     -0.09274    0.03930 312.27921  -2.36   0.0189 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

```

```

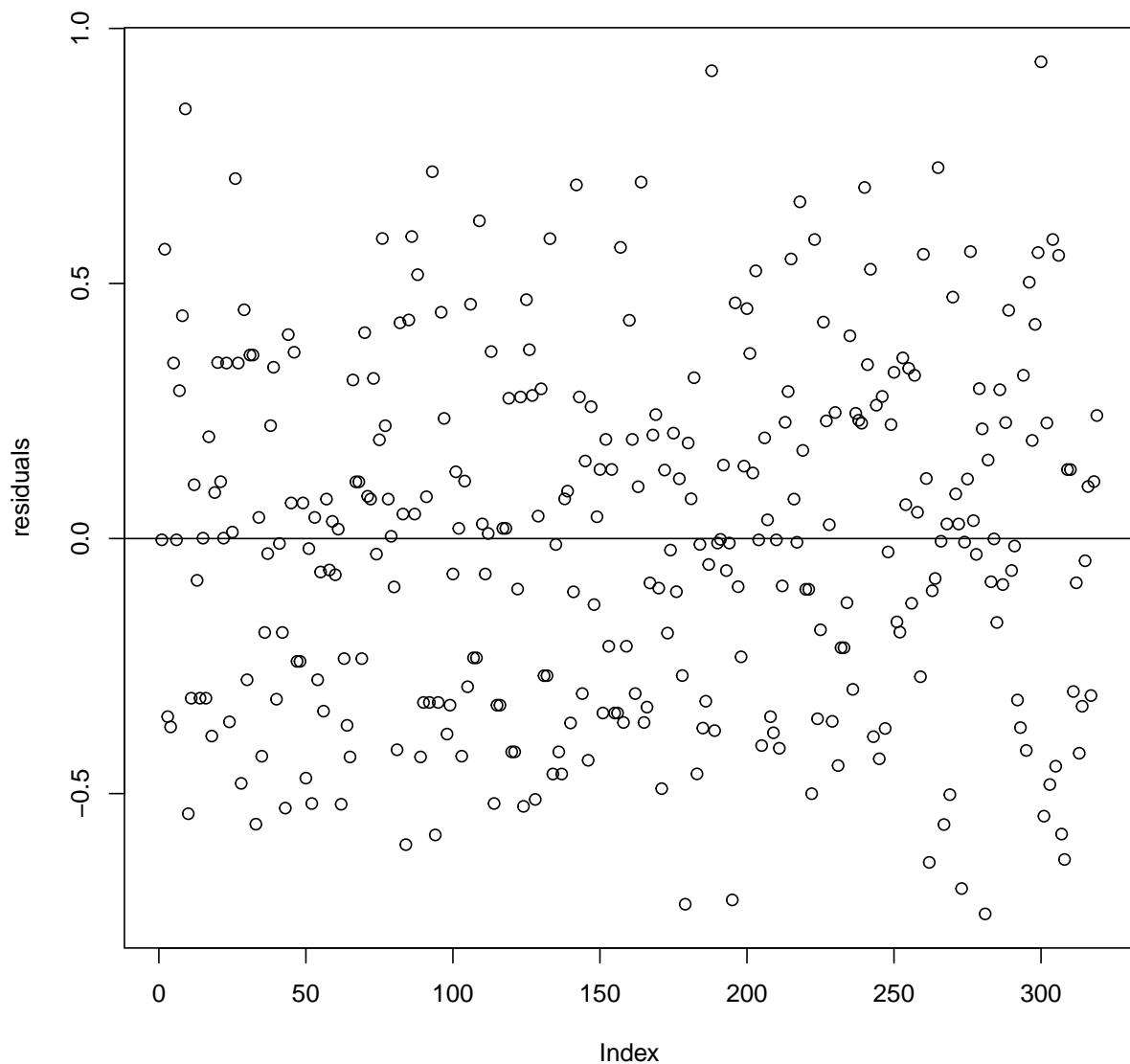
## Correlation of Fixed Effects:
##          (Intr)
## initopen -0.469

anova(Hurd.mod.2) # lmer on non-zero diversity values is significant

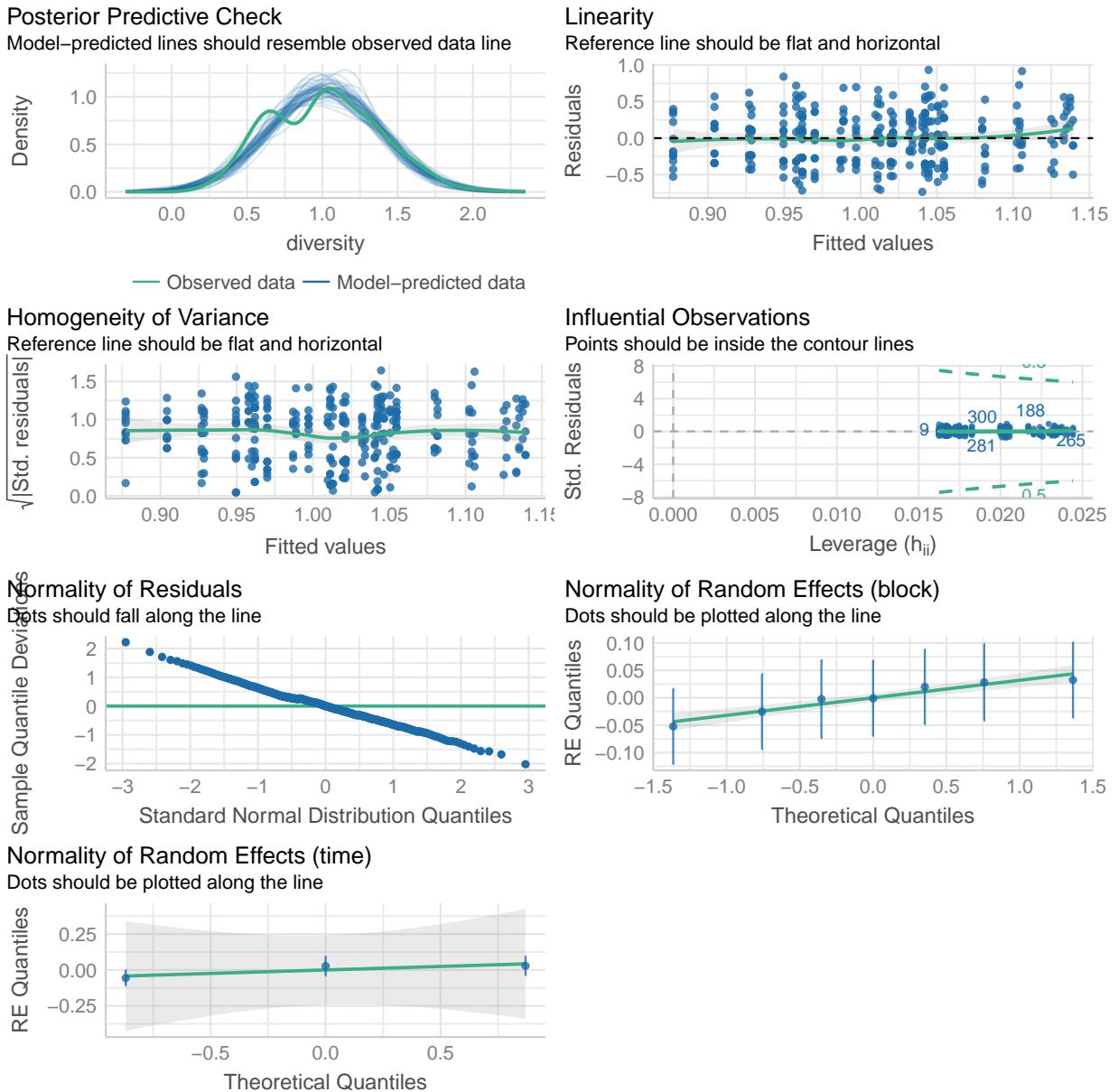
## Type III Analysis of Variance Table with Satterthwaite's method
##           Sum Sq Mean Sq NumDF DenDF F value Pr(>F)
## init 0.66558 0.66558     1 312.28  5.5685 0.0189 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

# Check residual
residuals <- resid(Hurd.mod.2) # Extract residuals
plot(residuals)
abline(0,0)

```



```
check_model(Hurd.mod.2)
```



```
##### Species diversity analysis for 2020 data (Shannon diversity on transect level)
dat$non_zero <- ifelse(dat$diversity > 0, 1, 0)
dat_2020 <- dat[which(dat$time=="t0"),] # just t0

# Hurdle model part 1. Logistic regression to predict the probability of non-zero.
# Ppen and log environment does determine the probability of non-zero diversity index
# 0 Shannon diversity index =/= zero plants
Hurd.mod.1 <- glmer(non_zero ~ init + (1|block), data = dat_2020, family = binomial)
summary(Hurd.mod.1) # significant
```

```
## Generalized linear mixed model fit by maximum likelihood (Laplace
## Approximation) [glmerMod]
## Family: binomial ( logit )
```

```

## Formula: non_zero ~ init + (1 | block)
## Data: dat_2020
##
##      AIC      BIC logLik deviance df.resid
##      259.8    270.0   -126.9     253.8     220
##
## Scaled residuals:
##      Min      1Q Median      3Q      Max
## -2.0667 -1.3966  0.5219  0.6096  0.7160
##
## Random effects:
## Groups Name      Variance Std.Dev.
## block  (Intercept) 0.06444  0.2538
## Number of obs: 223, groups: block, 7
##
## Fixed effects:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  1.2536    0.2499   5.017 5.26e-07 ***
## initopen     -0.3675    0.3093  -1.188    0.235
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##          (Intr)
## initopen -0.678

# Hurdle model part 1. Linear mixed effect model to predict the mean of the non-zero data.
# We included block and time as random terms.
# Open and log environment is significant in explaining the plant diversity in transect levels.
# Lmer was used because the distribution of non-zero data is approximately normal.
Hurd.mod.2 <- lmer(diversity ~ init + (1|block), data=subset(dat_2020, non_zero == 1))
summary(Hurd.mod.2)

```

```

## Linear mixed model fit by REML. t-tests use Satterthwaite's method [
## lmerModLmerTest]
## Formula: diversity ~ init + (1 | block)
## Data: subset(dat_2020, non_zero == 1)
##
## REML criterion at convergence: 121.4
##
## Scaled residuals:
##      Min      1Q Median      3Q      Max
## -1.73628 -0.90818  0.07998  0.73712  2.54204
##
## Random effects:
## Groups Name      Variance Std.Dev.
## block  (Intercept) 0.0000   0.0000
## Residual           0.1168   0.3418
## Number of obs: 165, groups: block, 7
##
## Fixed effects:
##             Estimate Std. Error      df t value Pr(>|t|)
## (Intercept)  1.00355   0.03686 163.00000 27.229 <2e-16 ***
## initopen     -0.08061   0.05326 163.00000 -1.513   0.132

```

```

## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##          (Intr)
## initopen -0.692
## optimizer (nloptwrap) convergence code: 0 (OK)
## boundary (singular) fit: see help('isSingular')

anova(Hurd.mod.2) # lmer on non-zero diversity values is significant

```

```

## Type III Analysis of Variance Table with Satterthwaite's method
##           Sum Sq Mean Sq NumDF DenDF F value Pr(>F)
## init 0.26758 0.26758     1    163  2.2906 0.1321

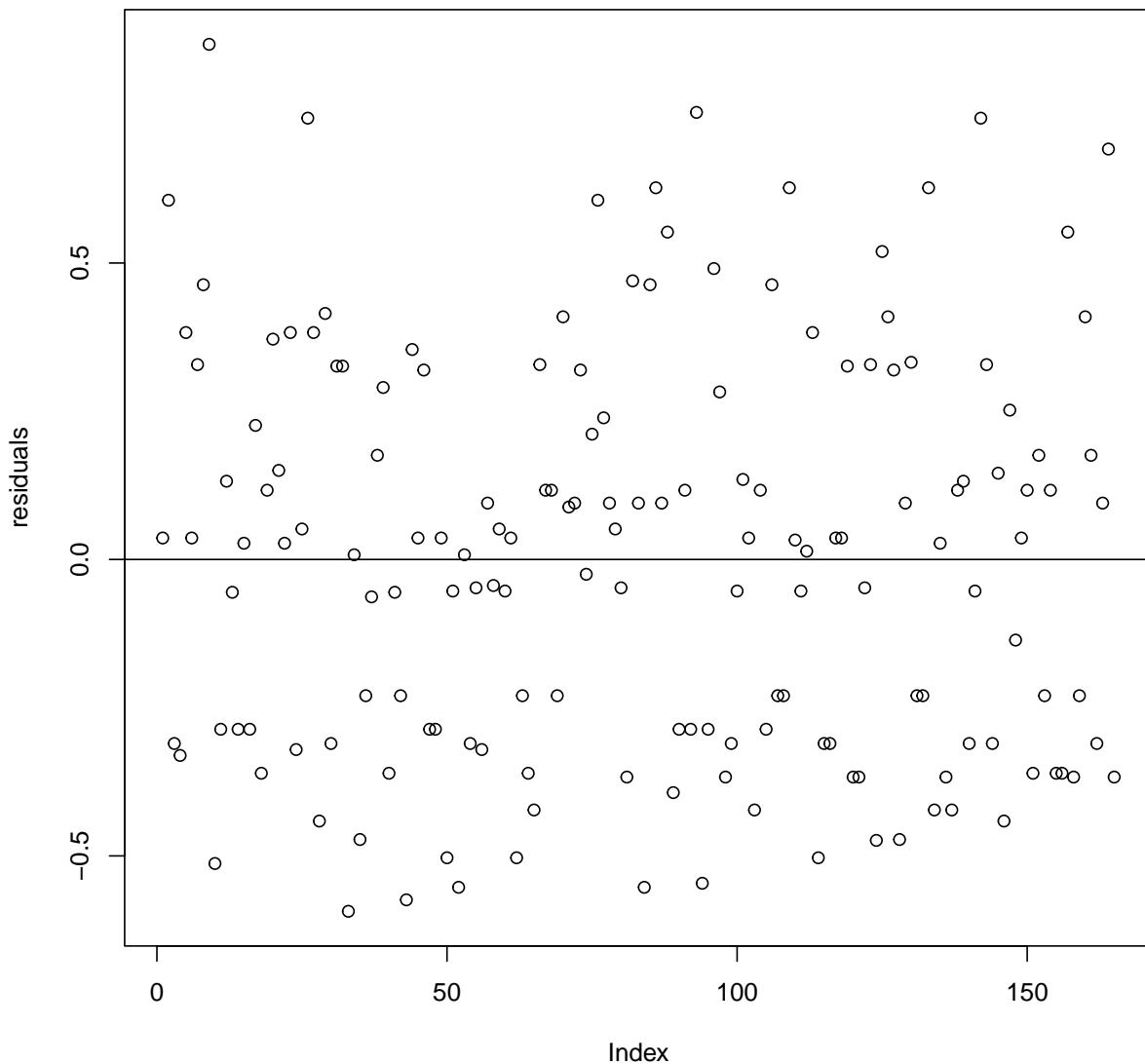
```

```

# Check residual
residuals <- resid(Hurd.mod.2) # Extract residuals
plot(residuals, main="2020")
abline(0,0)

```

2020



```
##### Species diversity analysis for 2021 data (Shannon diversity on transect level)
dat$non_zero <- ifelse(dat$diversity > 0, 1, 0)
dat_2021 <- dat[which(dat$time=="t1"),] # just t1
```

```
# Hurdle model part 1. Logistic regression to predict the probability of non-zero.
# Ppen and log environment does determine the probability of non-zero diversity index
# 0 Shannon diversity index =/= zero plants
Hurd.mod.1 <- glmer(non_zero ~ init + (1|block), data = dat_2021, family = binomial)
summary(Hurd.mod.1) # significant
```

```
## Generalized linear mixed model fit by maximum likelihood (Laplace
## Approximation) [glmerMod]
## Family: binomial ( logit )
```

```

## Formula: non_zero ~ init + (1 | block)
##   Data: dat_2021
##
##      AIC      BIC logLik deviance df.resid
##      57.0     64.3    -25.5     51.0      83
##
## Scaled residuals:
##      Min      1Q Median      3Q      Max
## -5.3852  0.1857  0.3780  0.3780  0.3780
##
## Random effects:
##   Groups Name      Variance Std.Dev.
##   block  (Intercept) 7.176e-14 2.679e-07
## Number of obs: 86, groups: block, 7
##
## Fixed effects:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  3.367     1.017   3.311 0.000931 ***
## initopen     -1.421     1.094  -1.299 0.194025
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##          (Intr)
## initopen -0.929
## optimizer (Nelder_Mead) convergence code: 0 (OK)
## boundary (singular) fit: see help('isSingular')

# Hurdle model part 1. Linear mixed effect model to predict the mean of the non-zero data.
# We included block and time as random terms.
# Open and log environment is significant in explaining the plant diversity in transect levels.
# Lmer was used because the distribution of non-zero data is approximately normal.
Hurd.mod.2 <- lmer(diversity ~ init + (1|block), data=subset(dat_2021, non_zero == 1))
summary(Hurd.mod.2)

## Linear mixed model fit by REML. t-tests use Satterthwaite's method [
## lmerModLmerTest]
## Formula: diversity ~ init + (1 | block)
##   Data: subset(dat_2021, non_zero == 1)
##
## REML criterion at convergence: 58.2
##
## Scaled residuals:
##      Min      1Q Median      3Q      Max
## -2.33234 -0.61602 -0.06749  0.62716  2.87311
##
## Random effects:
##   Groups Name      Variance Std.Dev.
##   block  (Intercept) 0.001897 0.04355
##   Residual           0.112931 0.33605
## Number of obs: 78, groups: block, 7
##
## Fixed effects:
##             Estimate Std. Error      df t value Pr(>|t|)

```

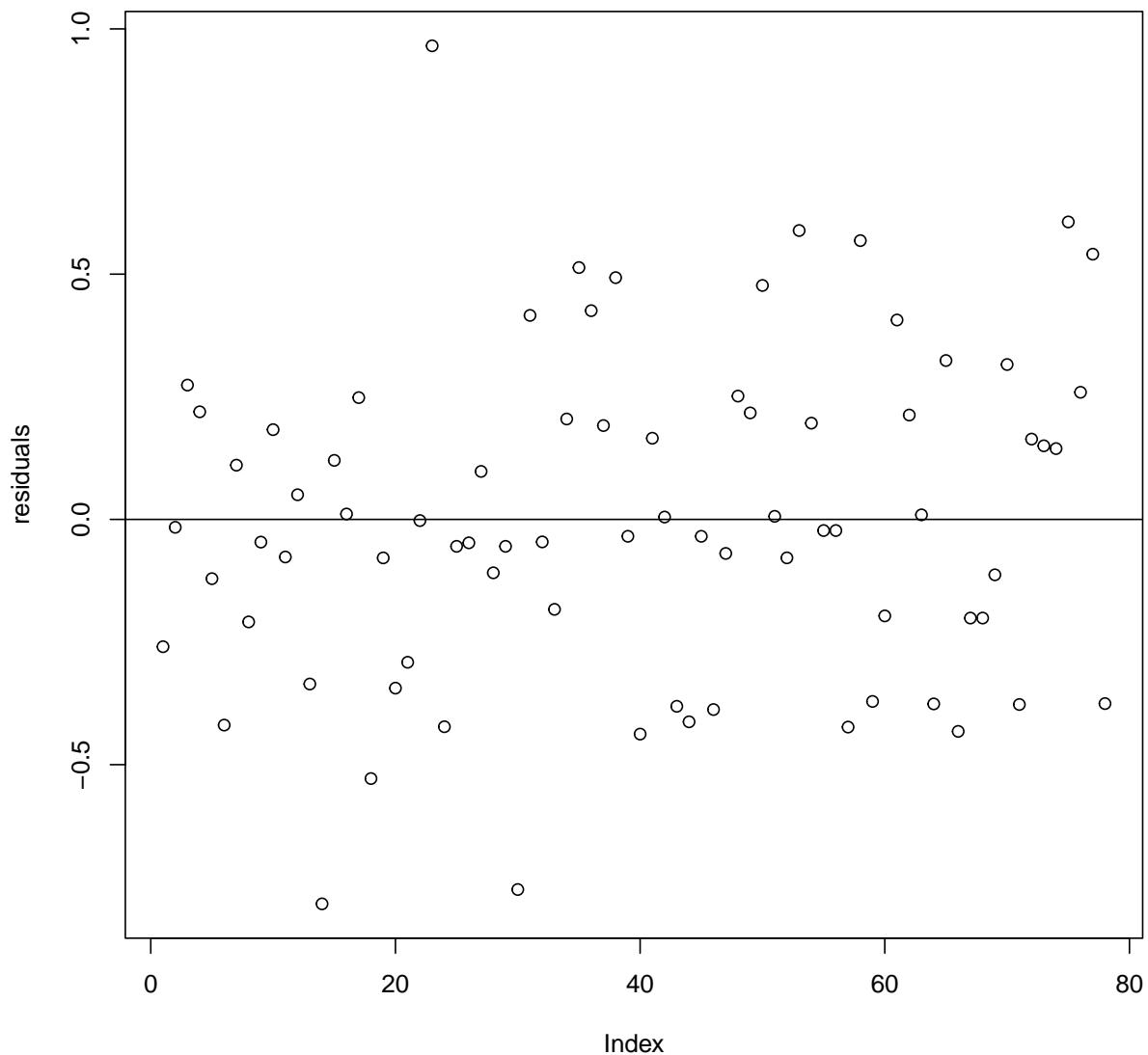
```
## (Intercept) 1.060584 0.064609 26.999316 16.416 1.42e-15 ***
## initopen 0.001865 0.078839 71.502601 0.024 0.981
## ---
## Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Correlation of Fixed Effects:
##          (Intr)
## initopen -0.767
```

```
anova(Hurd.mod.2) # lmer on non-zero diversity values is significant
```

```
## Type III Analysis of Variance Table with Satterthwaite's method
##           Sum Sq   Mean Sq NumDF DenDF F value Pr(>F)
## init 6.3172e-05 6.3172e-05     1 71.503 6e-04 0.9812
```

```
# Check residual
residuals <- resid(Hurd.mod.2) # Extract residuals
plot(residuals, main="2021")
abline(0,0)
```

2021



```
##### Species diversity analysis for 2022 data (Shannon diversity on transect level)
dat$non_zero <- ifelse(dat$diversity > 0, 1, 0)
dat_2022 <- dat[which(dat$time=="t2"),] # just t1
```

```
# Hurdle model part 1. Logistic regression to predict the probability of non-zero.
# Ppen and log environment does determine the probability of non-zero diversity index
# 0 Shannon diversity index =/= zero plants
Hurd.mod.1 <- glmer(non_zero ~ init + (1|block), data = dat_2022, family = binomial)
summary(Hurd.mod.1) # significant
```

```
## Generalized linear mixed model fit by maximum likelihood (Laplace
## Approximation) [glmerMod]
## Family: binomial ( logit )
```

```

## Formula: non_zero ~ init + (1 | block)
## Data: dat_2022
##
##      AIC      BIC logLik deviance df.resid
##      50.0     57.3   -22.0     44.0     81
##
## Scaled residuals:
##      Min      1Q Median      3Q      Max
## -2.8055  0.0000  0.2544  0.3564  0.7136
##
## Random effects:
## Groups Name      Variance Std.Dev.
## block  (Intercept) 1.155    1.075
## Number of obs: 84, groups: block, 7
##
## Fixed effects:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  29.91     915.89   0.033   0.974
## initopen    -27.74     915.89  -0.030   0.976
##
## Correlation of Fixed Effects:
##          (Intr)
## initopen -1.000
## optimizer (Nelder_Mead) convergence code: 0 (OK)
## Model is nearly unidentifiable: large eigenvalue ratio
## - Rescale variables?

# Hurdle model part 1. Linear mixed effect model to predict the mean of the non-zero data.
# We included block and time as random terms.
# Open and log environment is significant in explaining the plant diversity in transect levels.
# Lmer was used because the distribution of non-zero data is approximately normal.
Hurd.mod.2 <- lmer(diversity ~ init + (1|block), data=subset(dat_2022, non_zero == 1))
summary(Hurd.mod.2)

## Linear mixed model fit by REML. t-tests use Satterthwaite's method [ 
## lmerModLmerTest]
## Formula: diversity ~ init + (1 | block)
## Data: subset(dat_2022, non_zero == 1)
##
## REML criterion at convergence: 68
##
## Scaled residuals:
##      Min      1Q Median      3Q      Max
## -1.9499 -0.6218  0.1205  0.6173  2.5134
##
## Random effects:
## Groups Name      Variance Std.Dev.
## block  (Intercept) 0.009191 0.09587
## Residual           0.127140 0.35657
## Number of obs: 76, groups: block, 7
##
## Fixed effects:
##             Estimate Std. Error      df t value Pr(>|t|)
## (Intercept)  1.21013   0.07675 22.14106 15.766 1.6e-13 ***

```

```

## initopen   -0.23615    0.08529 70.27712  -2.769  0.00719 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##           (Intr)
## initopen -0.701

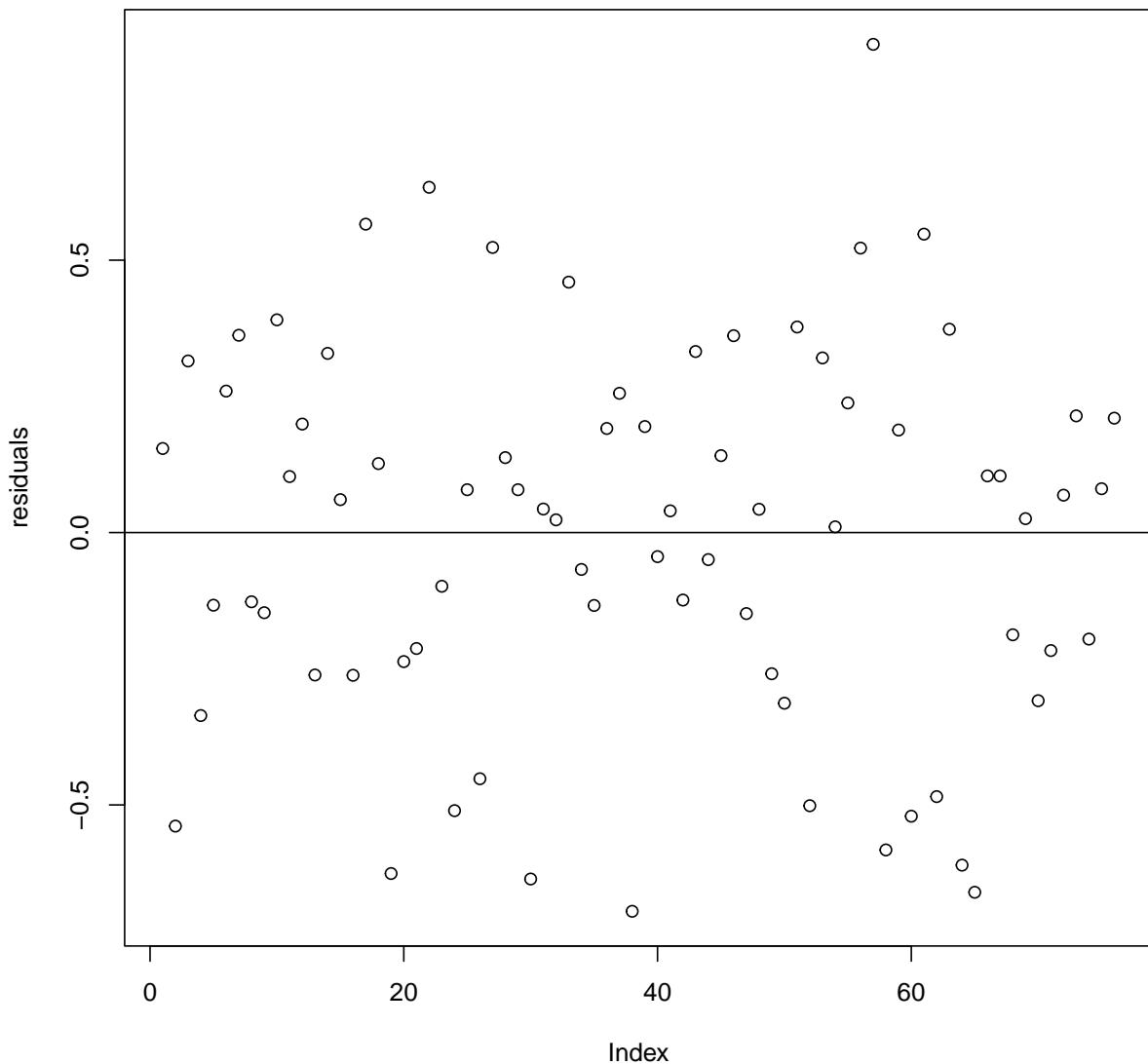
anova(Hurd.mod.2) # lmer on non-zero diversity values is significant

## Type III Analysis of Variance Table with Satterthwaite's method
##          Sum Sq Mean Sq NumDF DenDF F value Pr(>F)
## init 0.97474 0.97474     1 70.277  7.6667 0.007188 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

# Check residual
residuals <- resid(Hurd.mod.2) # Extract residuals
plot(residuals, main="2022")
abline(0,0)

```

2022



Composition analysis Composition dissimilarity * 2020 *

```
# subset data where all t0 communities, insitu log and insitu open communities at t1 and t2 are included
mat2 <- mat[which(mat$time=="t0"),]
mat2$grp<-apply(mat2[c(89,91)], 1, paste, collapse=":") # block, init as grouping
# names(mat2) #check

# another df where the grouping variables are time, block and initial state
# each row is a transect in a certain year.
df<-mat2[,c(1:87, 93)]
df2 = df %>% mutate(across(.cols=1:87,.fns=as.numeric)) # make everything numeric
rownames(df2)<-NULL # remove rownames
```

```

## new with group vars
nublock<-separate(df2, 88, c("block", "init"), ":") # just looking at time, block & initial treatment

# want to sum across transects in same block X init treatment
nublock$sumgrp<-apply(mat2[c(89, 91)], 1, paste, collapse=":")
# head(nublock)

# sum observations across initial X block (group variable)
# this gives number of plants in each transect TYPE for each year in each block. should be 2 types X 3
blocksum<-rowsum(nublock[,c(1:87)], group=nublock$sumgrp)
blocksum$grps<-rownames(blocksum)
rownames(blocksum)<-NULL # remove rownames
# nrow(blocksum) # it is 14 rows as expected

## expand again
blocksum<-separate(blocksum, 88, c("block", "init"), ":")

# at the moment this includes where there were no plants ("x" column in matrix)
assemblies_t0<-blocksum[,c(1:87)]

# group - these are the treatment variables that need to be separately fed into the MDS analysis from
group_init<-blocksum$init
group_block<-blocksum$block

# MDS
ass.rel.t0<-decostand(assemblies_t0, method='hel') #standardize assemblies
ass.rel.t0_NMS <- metaMDS(ass.rel.t0, distance = 'bray', k = 4) # run MDS

```

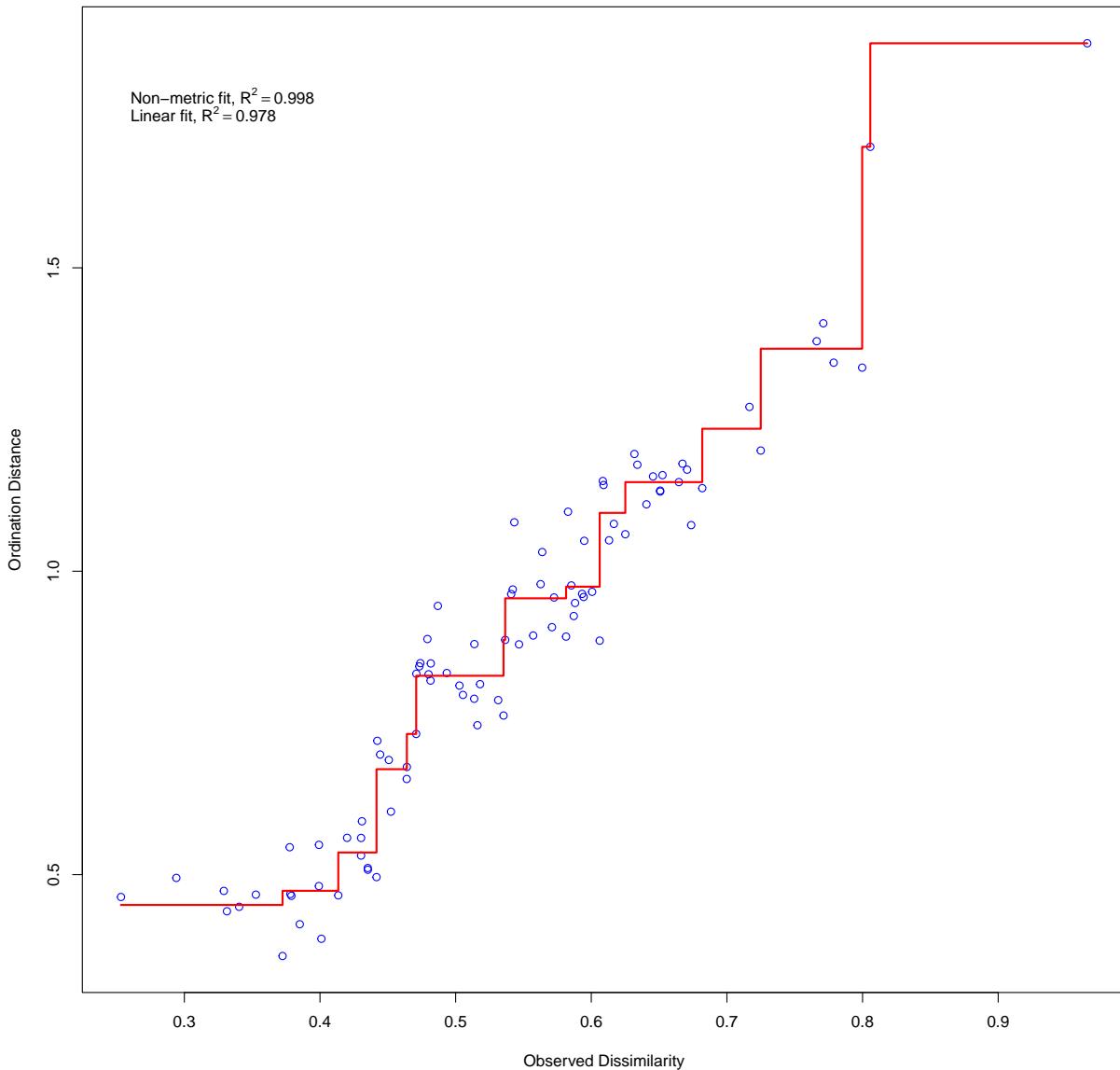
```

## Run 0 stress 0.04866056
## Run 1 stress 0.04798385
## ... New best solution
## ... Procrustes: rmse 0.1313001 max resid 0.3125838
## Run 2 stress 0.05073043
## Run 3 stress 0.04866043
## Run 4 stress 0.04866032
## Run 5 stress 0.04798376
## ... New best solution
## ... Procrustes: rmse 0.0003526648 max resid 0.0009224645
## ... Similar to previous best
## Run 6 stress 0.04866072
## Run 7 stress 0.04866068
## Run 8 stress 0.04798386
## ... Procrustes: rmse 0.0003256098 max resid 0.0008868523
## ... Similar to previous best
## Run 9 stress 0.0523553
## Run 10 stress 0.05073022
## Run 11 stress 0.04798379
## ... Procrustes: rmse 0.0003032552 max resid 0.0006103313
## ... Similar to previous best
## Run 12 stress 0.04798368
## ... New best solution
## ... Procrustes: rmse 0.0001954057 max resid 0.0003932473
## ... Similar to previous best

```

```
## Run 13 stress 0.04798367
## ... New best solution
## ... Procrustes: rmse 4.298873e-05 max resid 0.0001016817
## ... Similar to previous best
## Run 14 stress 0.04798386
## ... Procrustes: rmse 0.0003162744 max resid 0.0005703216
## ... Similar to previous best
## Run 15 stress 0.04866063
## Run 16 stress 0.04866011
## Run 17 stress 0.04866016
## Run 18 stress 0.04866039
## Run 19 stress 0.04798367
## ... New best solution
## ... Procrustes: rmse 6.064216e-05 max resid 0.0001274433
## ... Similar to previous best
## Run 20 stress 0.05073317
## *** Best solution repeated 1 times

stressplot(ass.rel.t0_NMS) # check fit
```



```

# scores
mds_scores_t0<-as.data.frame(vegan::scores(ass.rel.t0_NMS)$sites) # extract scores
mds_scores_t0$site<-rownames(vegan::scores(ass.rel.t0_NMS)$sites) # extract names
mds_scores_t0$treatment<-group_init # grouping factor 1
mds_scores_t0$block<-group_block # grouping factor 2

# explaining factors
init<-as.factor(group_init) # grouping factor 1- convert to factor
block<-as.factor(group_block) # grouping factor 2- convert to factor

#### Redundancy analysis
# can look at significance of model where initial treatment and block explain variation in community
trt_tot_2<-rda(ass.rel.t0~init+block) # run model using standardized data
summary(trt_tot_2)
  
```

```

##
## Call:
## rda(formula = ass.rel.t0 ~ init + block)
##
## Partitioning of variance:
##           Inertia Proportion
## Total      0.42624    1.0000
## Constrained 0.32981    0.7738
## Unconstrained 0.09643    0.2262
##
## Eigenvalues, and their contribution to the variance
##
## Importance of components:
##           RDA1   RDA2   RDA3   RDA4   RDA5   RDA6   RDA7
## Eigenvalue      0.1145  0.0834  0.03956  0.03545  0.02748  0.01667  0.01278
## Proportion Explained 0.2686  0.1957  0.09282  0.08316  0.06446  0.03911  0.02999
## Cumulative Proportion 0.2686  0.4642  0.55704  0.64021  0.70467  0.74378  0.77376
##           PC1     PC2     PC3     PC4     PC5     PC6
## Eigenvalue      0.03207 0.02047 0.01524 0.01394 0.01034 0.004375
## Proportion Explained 0.07523 0.04802 0.03576 0.03271 0.02426 0.010264
## Cumulative Proportion 0.84900 0.89701 0.93277 0.96548 0.98974 1.000000
##
## Accumulated constrained eigenvalues
## Importance of components:
##           RDA1   RDA2   RDA3   RDA4   RDA5   RDA6   RDA7
## Eigenvalue      0.1145  0.0834  0.03956  0.03545  0.02748  0.01667  0.01278
## Proportion Explained 0.3471  0.2529  0.11995  0.10748  0.08331  0.05055  0.03876
## Cumulative Proportion 0.3471  0.6000  0.71991  0.82739  0.91070  0.96124  1.000000
##
## Scaling 2 for species and site scores
## * Species are scaled proportional to eigenvalues
## * Sites are unscaled: weighted dispersion equal on all dimensions
## * General scaling constant of scores: 1.534259
##
##
## Species scores
##
##           RDA1       RDA2       RDA3       RDA4       RDA5       RDA6
## acul -1.641e-01 -3.061e-02  5.709e-03 -6.882e-02  1.731e-01 -1.547e-02
## aicu  9.454e-02 -2.259e-02 -1.224e-02  5.916e-02  6.241e-02  5.770e-02
## arca  1.124e-16 -9.425e-17 -1.501e-17  2.657e-17 -6.239e-18 -5.518e-17
## ardy  1.229e-17 -8.987e-19  1.159e-17 -1.534e-17 -1.923e-18  1.107e-17
## arsp  -2.057e-02  1.214e-03 -1.623e-02 -3.482e-02  1.144e-02  2.195e-02
## auel -1.119e-18  8.250e-19 -1.770e-17 -2.089e-18 -1.530e-17 -1.169e-17
## bldr -4.650e-02 -1.865e-01  1.009e-01 -2.439e-01 -6.057e-02  6.906e-02
## blrd -8.371e-18 -1.889e-17  5.282e-18 -1.012e-17  7.877e-18 -5.634e-18
## brdi  1.057e-17  2.442e-17 -7.953e-19 -9.977e-18 -7.727e-19  1.304e-17
## brdr  9.121e-19  4.301e-19  4.740e-19  1.210e-18 -4.747e-19 -4.240e-19
## brpe  3.002e-17  4.527e-17  1.770e-18  1.447e-17 -1.303e-17  1.438e-17
## brru  9.723e-34 -1.029e-33  2.395e-33  7.087e-35 -3.079e-34 -6.179e-33
## buse -2.968e-02  2.051e-02  3.459e-02 -3.225e-02  4.471e-02 -1.316e-02
## caer  4.476e-02  1.265e-01  5.800e-02  1.528e-02  7.746e-03  2.318e-03

```

```

## cagr -2.673e-02 1.192e-03 9.765e-02 8.019e-02 5.774e-02 -1.108e-02
## cahi -7.468e-02 1.317e-02 -9.131e-02 6.646e-03 -2.484e-02 -5.028e-02
## casp -3.792e-02 5.793e-02 -6.467e-02 -4.605e-02 -7.997e-03 1.873e-03
## cear 0.000e+00 0.000e+00 0.000e+00 0.000e+00 0.000e+00 0.000e+00
## chau 1.462e-01 2.094e-03 -7.128e-02 1.410e-02 1.225e-02 5.751e-02
## chei 0.000e+00 0.000e+00 0.000e+00 0.000e+00 0.000e+00 0.000e+00
## chps -1.385e-01 -5.737e-02 -1.133e-02 -7.725e-02 -2.818e-02 1.728e-02
## crcl 0.000e+00 0.000e+00 0.000e+00 0.000e+00 0.000e+00 0.000e+00
## crco -1.608e-01 5.208e-02 -5.018e-02 -4.208e-02 2.736e-02 -9.587e-02
## cusc 0.000e+00 0.000e+00 0.000e+00 0.000e+00 0.000e+00 0.000e+00
## cusp 0.000e+00 0.000e+00 0.000e+00 0.000e+00 0.000e+00 0.000e+00
## dagl -2.996e-02 -1.069e-03 -7.782e-03 4.675e-02 -2.198e-02 -6.498e-03
## dosp 0.000e+00 0.000e+00 0.000e+00 0.000e+00 0.000e+00 0.000e+00
## ento 0.000e+00 0.000e+00 0.000e+00 0.000e+00 0.000e+00 0.000e+00
## erau 0.000e+00 0.000e+00 0.000e+00 0.000e+00 0.000e+00 0.000e+00
## ercy 0.000e+00 0.000e+00 0.000e+00 0.000e+00 0.000e+00 0.000e+00
## erra -2.630e-02 2.865e-03 5.020e-02 9.019e-03 1.143e-01 -3.362e-02
## ersp -2.457e-03 6.427e-03 2.496e-02 3.462e-03 3.914e-02 -2.745e-02
## gite 2.530e-01 -1.813e-01 -1.258e-01 3.334e-02 5.780e-02 -2.361e-03
## gnre 0.000e+00 0.000e+00 0.000e+00 0.000e+00 0.000e+00 0.000e+00
## gobe -1.398e-02 1.872e-01 -4.743e-02 2.742e-02 8.577e-02 -7.442e-03
## gocy 2.797e-02 1.778e-01 -6.369e-02 -4.962e-02 -2.874e-02 4.811e-02
## gono 0.000e+00 0.000e+00 0.000e+00 0.000e+00 0.000e+00 0.000e+00
## goro -4.960e-02 -7.505e-02 -1.754e-02 5.339e-02 -1.099e-01 -1.844e-02
## gosp 7.031e-02 9.069e-02 -5.139e-02 -3.052e-02 -5.351e-05 8.966e-02
## haod -1.248e-02 6.184e-02 -2.591e-02 -9.287e-02 1.495e-02 2.948e-02
## hygl -2.940e-01 8.742e-02 8.182e-02 -1.113e-01 1.533e-02 1.122e-02
## hypi 0.000e+00 0.000e+00 0.000e+00 0.000e+00 0.000e+00 0.000e+00
## hypo 0.000e+00 0.000e+00 0.000e+00 0.000e+00 0.000e+00 0.000e+00
## jubu 0.000e+00 0.000e+00 0.000e+00 0.000e+00 0.000e+00 0.000e+00
## laro 1.049e-01 1.210e-01 5.046e-02 -1.820e-02 -5.824e-02 -5.001e-03
## ledu 0.000e+00 0.000e+00 0.000e+00 0.000e+00 0.000e+00 0.000e+00
## lele -2.057e-02 1.214e-03 -1.623e-02 -3.482e-02 1.144e-02 2.195e-02
## loef 0.000e+00 0.000e+00 0.000e+00 0.000e+00 0.000e+00 0.000e+00
## misp 3.690e-02 2.197e-01 1.344e-01 4.312e-02 1.060e-01 2.985e-02
## mite 0.000e+00 0.000e+00 0.000e+00 0.000e+00 0.000e+00 0.000e+00
## momo 0.000e+00 0.000e+00 0.000e+00 0.000e+00 0.000e+00 0.000e+00
## mopa 0.000e+00 0.000e+00 0.000e+00 0.000e+00 0.000e+00 0.000e+00
## niro 0.000e+00 0.000e+00 0.000e+00 0.000e+00 0.000e+00 0.000e+00
## omco -6.761e-02 6.264e-03 -8.979e-02 4.478e-02 -3.621e-02 -5.944e-02
## orsp 0.000e+00 0.000e+00 0.000e+00 0.000e+00 0.000e+00 0.000e+00
## pala -2.057e-02 1.214e-03 -1.623e-02 -3.482e-02 1.144e-02 2.195e-02
## peai 8.389e-02 -4.825e-02 -9.446e-02 -6.195e-02 8.932e-02 7.437e-02
## pedu -2.909e-02 1.717e-03 -2.296e-02 -4.924e-02 1.617e-02 3.105e-02
## phsu 0.000e+00 0.000e+00 0.000e+00 0.000e+00 0.000e+00 0.000e+00
## plde 0.000e+00 0.000e+00 0.000e+00 0.000e+00 0.000e+00 0.000e+00
## poar 0.000e+00 0.000e+00 0.000e+00 0.000e+00 0.000e+00 0.000e+00
## poca -3.514e-03 -6.344e-02 -3.487e-03 1.164e-01 1.794e-02 3.093e-02
## pocap 6.856e-02 1.456e-01 -1.503e-02 -7.994e-04 -2.630e-02 7.238e-02
## poce 0.000e+00 0.000e+00 0.000e+00 0.000e+00 0.000e+00 0.000e+00
## pogn 0.000e+00 0.000e+00 0.000e+00 0.000e+00 0.000e+00 0.000e+00
## pole 4.261e-02 -3.143e-01 1.989e-01 5.584e-03 3.903e-02 -2.368e-02
## pomu -3.888e-01 -6.207e-02 6.793e-03 1.601e-01 -7.784e-02 7.456e-02
## pter 0.000e+00 0.000e+00 0.000e+00 0.000e+00 0.000e+00 0.000e+00

```

```

## ptga -2.638e-02 1.156e-01 5.406e-02 -4.216e-03 2.747e-02 1.276e-02
## ptob 0.000e+00 0.000e+00 0.000e+00 0.000e+00 0.000e+00 0.000e+00
## rhla 1.865e-01 -1.907e-01 -5.259e-02 3.145e-02 5.749e-02 2.988e-02
## rhpy -7.913e-02 1.351e-02 -8.944e-02 1.770e-02 5.222e-03 -1.066e-01
## rhsp 0.000e+00 0.000e+00 0.000e+00 0.000e+00 0.000e+00 0.000e+00
## ry 0.000e+00 0.000e+00 0.000e+00 0.000e+00 0.000e+00 0.000e+00
## scna 0.000e+00 0.000e+00 0.000e+00 0.000e+00 0.000e+00 0.000e+00
## sino 0.000e+00 0.000e+00 0.000e+00 0.000e+00 0.000e+00 0.000e+00
## sool 0.000e+00 0.000e+00 0.000e+00 0.000e+00 0.000e+00 0.000e+00
## stfi 2.849e-02 4.794e-02 4.360e-03 2.079e-06 -1.066e-02 7.798e-04
## stpi 0.000e+00 0.000e+00 0.000e+00 0.000e+00 0.000e+00 0.000e+00
## thma 0.000e+00 0.000e+00 0.000e+00 0.000e+00 0.000e+00 0.000e+00
## trcy 2.872e-01 6.164e-02 2.157e-02 -1.214e-01 -9.430e-02 -1.085e-01
## tris 0.000e+00 0.000e+00 0.000e+00 0.000e+00 0.000e+00 0.000e+00
## tror 1.659e-01 1.497e-01 9.863e-02 2.978e-02 -4.833e-02 9.223e-04
## trpi 4.134e-03 -1.302e-02 2.638e-02 -2.102e-02 -2.835e-02 -1.796e-02
## waac 1.327e-01 3.005e-02 1.793e-01 8.160e-02 -6.604e-02 6.145e-03
## wagr 2.804e-02 6.822e-03 2.162e-02 -3.666e-02 6.890e-02 -6.540e-02
## x 4.222e-02 2.810e-02 4.696e-02 1.101e-02 9.372e-02 -2.970e-02
##
##
## Site scores (weighted sums of species scores)
##
##          RDA1     RDA2     RDA3     RDA4     RDA5     RDA6
## row1  0.01134  0.03365  0.5830  0.59882 -0.13199  0.07154
## row2 -0.35273 -0.20722  0.2908  0.68511 -0.24191  0.60925
## row3  0.92160 -0.40472 -0.1107 -0.08527  0.03137 -0.35104
## row4  0.54518 -0.66372 -0.6352  0.27979  0.31059  0.36234
## row5  0.14409 -0.31373  0.5202 -0.58134 -0.85710 -0.21776
## row6 -0.25541 -0.45839  0.3355 -0.32197 -0.24935  0.06507
## row7 -0.07265  0.27890  0.4259  0.29129  0.89699 -0.66481
## row8 -0.31099 -0.24472  0.2812 -0.16299  0.73556  0.20448
## row9 -0.24068  0.09279 -0.5603  0.19753 -0.39454 -1.00721
## row10 -0.67007  0.03489 -0.6692  0.44074 -0.13873  0.05273
## row11 -0.16582  0.16093 -0.0174 -0.44916  0.14658 -0.08480
## row12 -0.33932  0.10302 -0.3393 -0.88552  0.19211  0.31775
## row13  0.40851  0.91226  0.1000  0.05301 -0.04839  0.10831
## row14  0.37695  0.67608 -0.2046 -0.06004 -0.25121  0.53413
##
##
## Site constraints (linear combinations of constraining variables)
##
##          RDA1     RDA2     RDA3     RDA4     RDA5     RDA6
## row1  -0.02692  0.02179  0.57132  6.455e-01 -0.23796  0.03386
## row2  -0.31447 -0.19537  0.30252  6.384e-01 -0.13594  0.64693
## row3   0.87716 -0.42564 -0.23853  1.008e-01  0.11997 -0.30089
## row4   0.58962 -0.64280 -0.50733  9.371e-02  0.22199  0.31219
## row5   0.08811 -0.27748  0.56227 -4.481e-01 -0.60424 -0.38288
## row6  -0.19943 -0.49464  0.29347 -4.552e-01 -0.50222  0.23019
## row7  -0.04805  0.12567  0.48796  6.770e-02  0.76527 -0.53670
## row8  -0.33559 -0.09149  0.21916  6.060e-02  0.86729  0.07638
## row9  -0.31160  0.17242 -0.48036  3.227e-01 -0.31764 -0.78378
## row10 -0.59915 -0.04474 -0.74917  3.156e-01 -0.21562 -0.17070
## row11 -0.10880  0.24056 -0.04395 -6.638e-01  0.11834 -0.19006

```

```

## row12 -0.39634  0.02339 -0.31275 -6.709e-01  0.22036  0.42302
## row13  0.53650  0.90275  0.08210  3.915e-05 -0.20081  0.01469
## row14  0.24896  0.68558 -0.18670 -7.067e-03 -0.09879  0.62776
##
##
## Biplot scores for constraining variables
##
##          RDA1      RDA2      RDA3      RDA4      RDA5      RDA6
## initopen -0.35062 -0.26480 -0.32777 -0.008665  0.1244  0.747567
## block2   0.73017 -0.53188 -0.37129  0.096832  0.1702  0.005627
## block3  -0.05542 -0.38436  0.42599 -0.449672 -0.5508 -0.076008
## block4  -0.19098  0.01701  0.35200  0.063869  0.8127 -0.229152
## block5  -0.45338  0.06356 -0.61207  0.317733 -0.2655 -0.475148
## block6  -0.25146  0.13140 -0.17757 -0.664411  0.1686  0.115967
## block7   0.39101  0.79068 -0.05207 -0.003498 -0.1491  0.319814
##
##
## Centroids for factor constraints
##
##          RDA1      RDA2      RDA3      RDA4      RDA5      RDA6
## initlog   0.14377  0.10858  0.1344  0.003553 -0.05101 -0.306538
## initopen -0.14377 -0.10858 -0.1344 -0.003553  0.05101  0.306538
## block1   -0.17070 -0.08679  0.4369  0.641965 -0.18695  0.340395
## block2    0.73339 -0.53422 -0.3729  0.097259  0.17098  0.005651
## block3  -0.05566 -0.38606  0.4279 -0.451654 -0.55323 -0.076344
## block4  -0.19182  0.01709  0.3536  0.064151  0.81628 -0.230162
## block5  -0.45538  0.06384 -0.6148  0.319134 -0.26663 -0.477242
## block6  -0.25257  0.13197 -0.1784 -0.667340  0.16935  0.116478
## block7   0.39273  0.79417 -0.0523 -0.003514 -0.14980  0.321224

```

```
anova.cca(trt_tot_2, step=1000, by="term") ## test for model significance
```

```

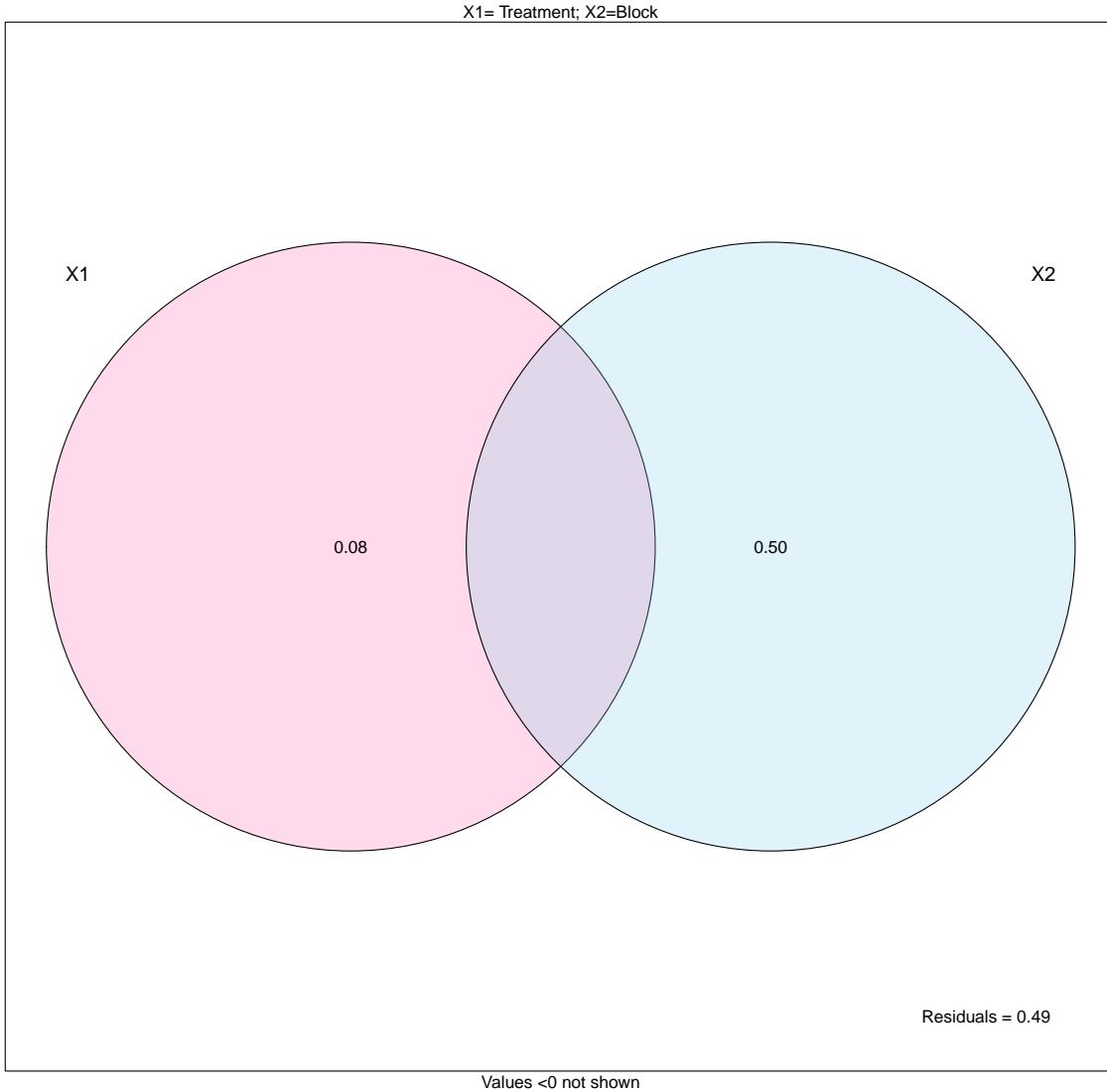
## Permutation test for rda under reduced model
## Terms added sequentially (first to last)
## Permutation: free
## Number of permutations: 999
##
## Model: rda(formula = ass.rel.t0 ~ init + block)
##          Df Variance     F Pr(>F)
## init      1 0.035514 2.2097  0.018 *
## block     6 0.294294 3.0519  0.001 ***
## Residual  6 0.096430
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

```

```

# Using varpart to look at contributions of initial treatment and block
var.mod2<-varpart(ass.rel.t0, init, block) # run model on standardized data
plot(var.mod2, bg=c("hotpink","skyblue"))
mtext("X1= Treatment; X2=Block", side=3)

```



```
# can test for significance of contribution of the fraction of initial treatment
# do this with partial redundancy analysis
trt_Frac<-rda(ass.rel.t0, init, block) # partial rda model
summary(trt_Frac)
```

```
##
## Call:
## rda(X = ass.rel.t0, Y = init, Z = block)
##
## Partitioning of variance:
##                 Inertia Proportion
## Total          0.42624   1.00000
## Conditioned   0.29429   0.69045
```

```

## Constrained 0.03551 0.08332
## Unconstrained 0.09643 0.22624
##
## Eigenvalues, and their contribution to the variance
## after removing the contribution of conditiniong variables
##
## Importance of components:
##          RDA1      PC1      PC2      PC3      PC4      PC5      PC6
## Eigenvalue      0.03551 0.03207 0.02047 0.01524 0.01394 0.01034 0.004375
## Proportion Explained 0.26916 0.24303 0.15511 0.11552 0.10566 0.07837 0.033159
## Cumulative Proportion 0.26916 0.51219 0.66730 0.78282 0.88848 0.96684 1.000000
##
## Accumulated constrained eigenvalues
## Importance of components:
##          RDA1
## Eigenvalue      0.03551
## Proportion Explained 1.000000
## Cumulative Proportion 1.000000
##
## Scaling 2 for species and site scores
## * Species are scaled proportional to eigenvalues
## * Sites are unscaled: weighted dispersion equal on all dimensions
## * General scaling constant of scores: 1.534259
##
##
## Species scores
##
##          RDA1      PC1      PC2      PC3      PC4      PC5
## acul   0.1067309 -9.564e-02 -2.516e-02  6.791e-02 -5.735e-02 -1.985e-02
## aicu   0.0342552  7.579e-02 -2.904e-02 -1.040e-01 -1.656e-02  3.629e-04
## arca   0.0000000 -5.092e-17  4.573e-17 -2.095e-17  2.733e-18  2.352e-18
## ardy   0.0000000 -1.340e-17  2.878e-18  4.804e-18  6.089e-18  3.913e-18
## arsp   0.0212814 -9.296e-03  3.074e-02 -1.243e-02  2.902e-02  2.380e-02
## auel   0.0000000  1.657e-17  1.767e-18 -7.370e-18 -1.235e-17 -4.376e-19
## bldr   0.0627812 -3.489e-02  5.885e-02 -3.487e-02  5.598e-02 -1.193e-01
## blrd   0.0000000  2.296e-18  1.300e-17  5.184e-19  1.453e-17  5.429e-18
## brdi   0.0000000 -1.645e-17  1.224e-17 -1.804e-17 -5.031e-18  1.778e-17
## brdr   0.0000000 -1.535e-18  1.057e-18  1.817e-19  3.219e-18  1.290e-18
## brpe   0.0000000  4.493e-17  7.212e-18 -2.652e-17 -7.924e-17 -1.269e-17
## brru   0.0000000 -3.497e-33 -2.348e-33  3.058e-33  5.468e-33 -1.861e-34
## buse   -0.0399181  8.064e-03 -8.920e-03 -7.638e-03 -1.217e-02  6.087e-02
## caer   -0.0770469  5.739e-03 -4.007e-02  2.232e-02 -3.681e-02  4.977e-02
## cagr   -0.0353354  3.492e-02 -5.082e-02 -7.084e-02  2.218e-02  4.523e-02
## cahi   0.0073560  1.491e-02  4.828e-02 -6.274e-03  3.863e-02  1.972e-02
## casp   0.0071091 -7.320e-02 -4.968e-02  8.494e-03  2.080e-02  8.123e-03
## cear   0.0000000  0.000e+00  0.000e+00  0.000e+00  0.000e+00  0.000e+00
## chau   0.0270050  2.642e-02  5.365e-02 -7.004e-02 -9.242e-02 -2.682e-03
## chei   0.0000000  0.000e+00  0.000e+00  0.000e+00  0.000e+00  0.000e+00
## chps   0.0867979 -8.706e-02 -4.807e-02  5.256e-03 -3.053e-02 -5.813e-02
## crcl   0.0000000  0.000e+00  0.000e+00  0.000e+00  0.000e+00  0.000e+00
## crco   -0.0054567 -2.817e-02 -2.789e-02 -6.578e-02 -6.492e-03  6.700e-02
## cusc   0.0000000  0.000e+00  0.000e+00  0.000e+00  0.000e+00  0.000e+00
## cusp   0.0000000  0.000e+00  0.000e+00  0.000e+00  0.000e+00  0.000e+00
## dagl   -0.0005158 -4.170e-02 -1.197e-02  1.578e-02 -3.486e-02  3.748e-02

```

## dosp	0.0000000	0.000e+00	0.000e+00	0.000e+00	0.000e+00	0.000e+00
## ento	0.0000000	0.000e+00	0.000e+00	0.000e+00	0.000e+00	0.000e+00
## erau	0.0000000	0.000e+00	0.000e+00	0.000e+00	0.000e+00	0.000e+00
## ercy	0.0000000	0.000e+00	0.000e+00	0.000e+00	0.000e+00	0.000e+00
## erra	-0.0017634	1.363e-03	-1.961e-03	-2.641e-03	6.378e-04	2.340e-03
## ersp	-0.0209707	1.621e-02	-2.332e-02	-3.140e-02	7.584e-03	2.783e-02
## gite	-0.0057450	-7.634e-03	5.428e-03	-7.119e-03	-3.080e-03	-1.608e-03
## gnte	0.0000000	0.000e+00	0.000e+00	0.000e+00	0.000e+00	0.000e+00
## gobe	-0.0320039	-3.133e-03	-9.076e-02	5.077e-02	1.243e-01	2.247e-02
## gocy	-0.0012808	6.463e-02	4.761e-02	-1.112e-02	-1.947e-02	-4.368e-03
## gono	0.0000000	0.000e+00	0.000e+00	0.000e+00	0.000e+00	0.000e+00
## goro	0.0215820	2.211e-01	-1.082e-01	-1.141e-02	-4.094e-02	-1.836e-02
## gosp	0.0446889	2.138e-02	1.224e-02	1.185e-02	3.669e-02	2.988e-02
## haod	-0.0050284	-3.182e-02	7.735e-03	1.165e-02	6.116e-02	1.742e-02
## hygl	0.0601486	6.535e-02	7.287e-02	6.640e-02	-5.544e-02	7.196e-02
## hypi	0.0000000	0.000e+00	0.000e+00	0.000e+00	0.000e+00	0.000e+00
## hypo	0.0000000	0.000e+00	0.000e+00	0.000e+00	0.000e+00	0.000e+00
## jubu	0.0000000	0.000e+00	0.000e+00	0.000e+00	0.000e+00	0.000e+00
## laro	-0.1146604	-4.765e-02	-1.202e-02	3.910e-02	-4.021e-02	-1.545e-02
## ledu	0.0000000	0.000e+00	0.000e+00	0.000e+00	0.000e+00	0.000e+00
## lele	0.0212814	-9.296e-03	3.074e-02	-1.243e-02	2.902e-02	2.380e-02
## loef	0.0000000	0.000e+00	0.000e+00	0.000e+00	0.000e+00	0.000e+00
## misp	-0.0859180	4.494e-02	-1.140e-01	-3.971e-02	-2.276e-02	5.166e-03
## mite	0.0000000	0.000e+00	0.000e+00	0.000e+00	0.000e+00	0.000e+00
## momo	0.0000000	0.000e+00	0.000e+00	0.000e+00	0.000e+00	0.000e+00
## mopa	0.0000000	0.000e+00	0.000e+00	0.000e+00	0.000e+00	0.000e+00
## niro	0.0000000	0.000e+00	0.000e+00	0.000e+00	0.000e+00	0.000e+00
## omco	0.0102406	-1.780e-02	-1.290e-02	-4.528e-03	-7.063e-03	3.001e-03
## orsp	0.0000000	0.000e+00	0.000e+00	0.000e+00	0.000e+00	0.000e+00
## pala	0.0212814	-9.296e-03	3.074e-02	-1.243e-02	2.902e-02	2.380e-02
## peai	0.0697049	5.909e-02	-2.821e-03	5.278e-02	-3.971e-02	-2.331e-02
## pedu	0.0300964	-1.315e-02	4.347e-02	-1.758e-02	4.104e-02	3.366e-02
## phsu	0.0000000	0.000e+00	0.000e+00	0.000e+00	0.000e+00	0.000e+00
## plde	0.0000000	0.000e+00	0.000e+00	0.000e+00	0.000e+00	0.000e+00
## poar	0.0000000	0.000e+00	0.000e+00	0.000e+00	0.000e+00	0.000e+00
## poca	0.0311535	7.429e-02	1.806e-02	8.779e-02	2.715e-02	-2.008e-02
## pocap	0.0157620	1.774e-02	1.191e-02	-1.551e-02	-2.774e-02	9.443e-04
## poce	0.0000000	0.000e+00	0.000e+00	0.000e+00	0.000e+00	0.000e+00
## pogn	0.0000000	0.000e+00	0.000e+00	0.000e+00	0.000e+00	0.000e+00
## pole	-0.0067798	6.543e-02	2.974e-02	8.404e-02	-4.950e-03	-2.006e-02
## pomu	0.1806915	-8.052e-02	-1.066e-01	1.495e-02	-1.385e-02	-5.198e-02
## pter	0.0000000	0.000e+00	0.000e+00	0.000e+00	0.000e+00	0.000e+00
## ptga	-0.0265218	3.413e-02	8.198e-05	2.118e-02	4.241e-02	3.353e-02
## ptob	0.0000000	0.000e+00	0.000e+00	0.000e+00	0.000e+00	0.000e+00
## rhla	0.0114587	1.612e-02	-1.907e-02	1.253e-01	-6.133e-02	3.825e-02
## rhpy	-0.0614885	1.909e-01	1.451e-02	3.551e-03	8.057e-02	-4.550e-02
## rhsp	0.0000000	0.000e+00	0.000e+00	0.000e+00	0.000e+00	0.000e+00
## ry	0.0000000	0.000e+00	0.000e+00	0.000e+00	0.000e+00	0.000e+00
## scna	0.0000000	0.000e+00	0.000e+00	0.000e+00	0.000e+00	0.000e+00
## sino	0.0000000	0.000e+00	0.000e+00	0.000e+00	0.000e+00	0.000e+00
## sool	0.0000000	0.000e+00	0.000e+00	0.000e+00	0.000e+00	0.000e+00
## stfi	-0.0217745	-2.451e-02	-1.645e-02	2.143e-02	3.832e-02	-1.304e-03
## stpi	0.0000000	0.000e+00	0.000e+00	0.000e+00	0.000e+00	0.000e+00
## thma	0.0000000	0.000e+00	0.000e+00	0.000e+00	0.000e+00	0.000e+00

```

## trcy -0.2016923 4.678e-02 1.599e-01 -2.560e-02 -4.775e-02 -1.767e-02
## tris 0.0000000 0.000e+00 0.000e+00 0.000e+00 0.000e+00 0.000e+00
## tror -0.1560795 -5.629e-02 -1.247e-02 -3.574e-02 -4.124e-02 -1.417e-02
## trpi -0.0192367 -2.197e-03 9.150e-03 -9.411e-03 2.883e-03 -2.219e-02
## waac -0.1168716 -1.021e-01 4.820e-02 -2.229e-02 7.287e-04 -5.756e-03
## wagr -0.0763637 1.108e-02 -5.682e-02 -5.042e-02 -3.786e-02 2.435e-03
## x    -0.0563652 -6.808e-02 4.036e-02 -1.343e-02 3.574e-02 6.019e-02
##
##
## Site scores (weighted sums of species scores)
##
##          RDA1      PC1      PC2      PC3      PC4      PC5
## sit1   -0.4306 -0.15560  0.2593  0.4984 -0.43518  0.64698
## sit2    0.4306  0.15560 -0.2593 -0.4984  0.43518 -0.64698
## sit3   -0.5088 -0.54489  0.3874 -0.5081 -0.21987 -0.11475
## sit4    0.5088  0.54489 -0.3874  0.5081  0.21987  0.11475
## sit5   -0.3922 -0.04684  0.1950 -0.2006  0.06145 -0.47310
## sit6    0.3922  0.04684 -0.1950  0.2006 -0.06145  0.47310
## sit7   -0.5074  0.31689 -0.4561 -0.6141  0.14830  0.54414
## sit8    0.5074 -0.31689  0.4561  0.6141 -0.14830 -0.54414
## sit9   -0.5125  0.71285  0.5165  0.1813  0.28281 -0.12016
## sit10  0.5125 -0.71285 -0.5165 -0.1813 -0.28281  0.12016
## sit11  -0.2572  0.17912 -0.5923  0.2395 -0.55918 -0.45854
## sit12  0.2572 -0.17912  0.5923 -0.2395  0.55918  0.45854
## sit13  -0.2616 -0.46153 -0.3099  0.4036  0.72167 -0.02456
## sit14  0.2616  0.46153  0.3099 -0.4036 -0.72167  0.02456
##
##
## Site constraints (linear combinations of constraining variables)
##
##          RDA1      PC1      PC2      PC3      PC4      PC5
## con1   -0.41 -0.15560  0.2593  0.4984 -0.43518  0.64698
## con2    0.41  0.15560 -0.2593 -0.4984  0.43518 -0.64698
## con3   -0.41 -0.54489  0.3874 -0.5081 -0.21987 -0.11475
## con4    0.41  0.54489 -0.3874  0.5081  0.21987  0.11475
## con5   -0.41 -0.04684  0.1950 -0.2006  0.06145 -0.47310
## con6    0.41  0.04684 -0.1950  0.2006 -0.06145  0.47310
## con7   -0.41  0.31689 -0.4561 -0.6141  0.14830  0.54414
## con8    0.41 -0.31689  0.4561  0.6141 -0.14830 -0.54414
## con9   -0.41  0.71285  0.5165  0.1813  0.28281 -0.12016
## con10   0.41 -0.71285 -0.5165 -0.1813 -0.28281  0.12016
## con11  -0.41  0.17912 -0.5923  0.2395 -0.55918 -0.45854
## con12   0.41 -0.17912  0.5923 -0.2395  0.55918  0.45854
## con13  -0.41 -0.46153 -0.3099  0.4036  0.72167 -0.02456
## con14   0.41  0.46153  0.3099 -0.4036 -0.72167  0.02456
##
##
## Biplot scores for constraining variables
##
##          RDA1 PC1 PC2 PC3 PC4 PC5
## Yopen     1   0   0   0   0   0

```

`RsquareAdj(trt_Frac)$adj.r.squared #explanatory power`

```

## [1] 0.08471055

anova.cca(trt_Frac) ## this tells us if first condition, init, significantly contributes to overall var

## Permutation test for rda under reduced model
## Permutation: free
## Number of permutations: 999
##
## Model: rda(X = ass.rel.t0, Y = init, Z = block)
##          Df Variance      F Pr(>F)
## Model      1 0.035514 2.2097  0.052 .
## Residual   6 0.096430
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

# Extracting species scores and plotting
# Species scores
species.scores<-as.data.frame(vegan::scores(ass.rel.t0_NMS,"species")) ## some species don't have scores
species.scores$species<-rownames(species.scores)

### NMDS 1 and 2
log<-mds_scores_t0[mds_scores_t0$treatment == "log", ][chull(mds_scores_t0[mds_scores_t0$treatment ==
"log", c("NMDS1", "NMDS2")])]

open<-mds_scores_t0[mds_scores_t0$treatment == "open", ][chull(mds_scores_t0[mds_scores_t0$treatment ==
"open", c("NMDS1", "NMDS2")])]

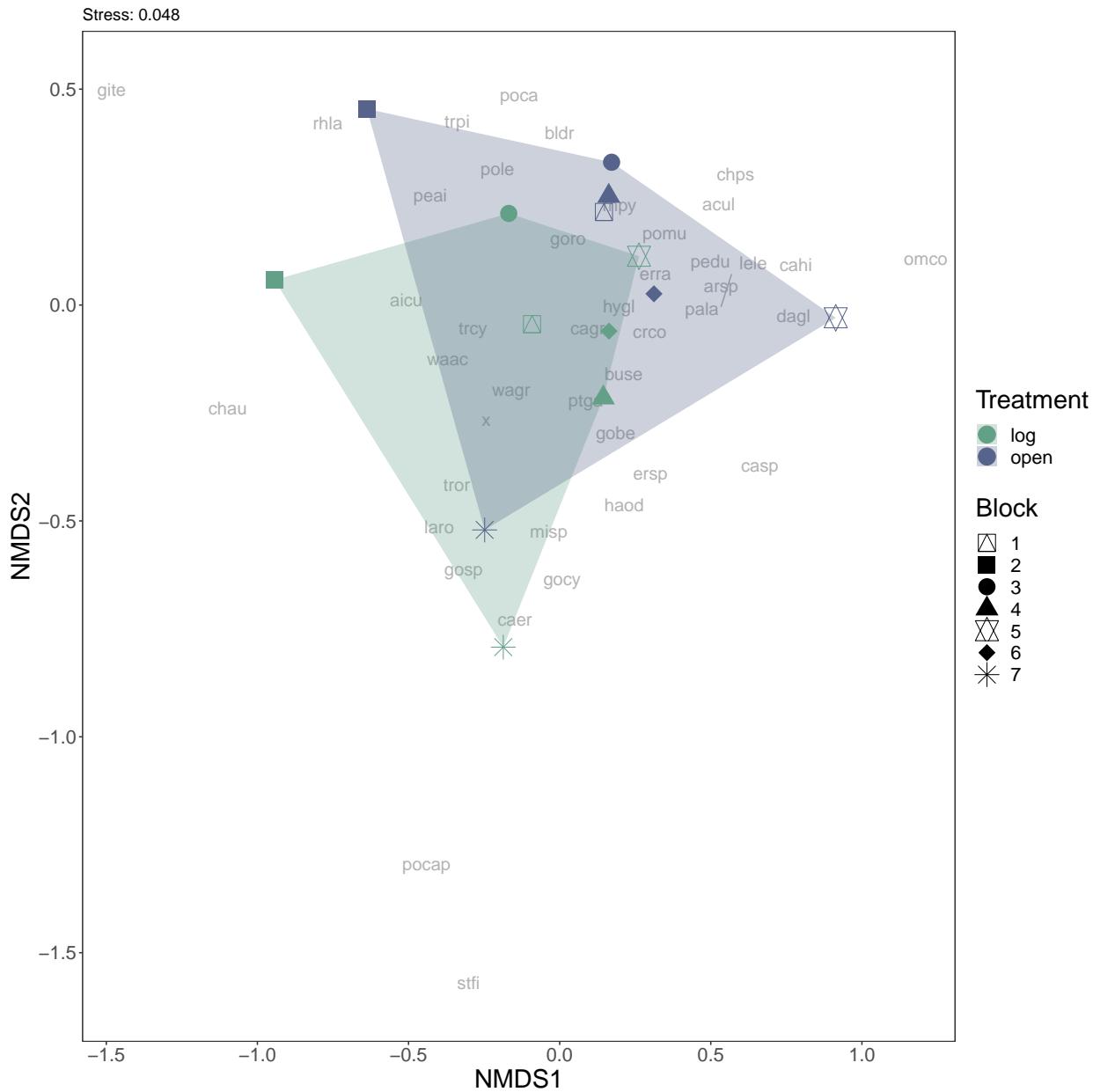
hulldat<-rbind(log,open)

nmds.plot<- ggplot()+
  theme_bw()+
  theme(panel.background = element_blank(),
        panel.grid.major = element_blank(), #remove major-grid labels
        panel.grid.minor = element_blank(), #remove minor-grid labels
        plot.background = element_blank(),
        axis.text = element_text(size = 15),
        axis.title=element_text(size=20),
        legend.title=element_text(size=20),
        legend.text=element_text(size=15))+

  geom_text_repel(data=species.scores, aes(NMDS1, NMDS2, label=species), alpha=0.9, size=5, col='darkgreen')+
  geom_polygon(data=hulldat, aes(NMDS1, NMDS2, fill=treatment, group=treatment), alpha=0.3)+scale_fill_manual(values=c("#63A088", "#56638A"), name="Treatment") +
  geom_point(data=mds_scores_t0, aes(NMDS1, NMDS2, shape=block, col=treatment), size=6)+ scale_shape_manual(values=c("triangle-down", "triangle-up", "square", "diamond", "circle", "cross", "triangle-left", "triangle-right", "square", "diamond", "circle", "cross"))+
  labs(title=paste0("Stress: ", round(ass.rel.t0_NMS$stress,3)))

print(nmds.plot)

```



```
# All other NMDS pairs
treatments <- unique(mds_scores_t0$treatment)
plots <- list()
num_axes <- 4

for (i in 1:num_axes) {
  for (j in (i+1):num_axes) {
    if (j <= num_axes && i != j) {
      log <- mds_scores_t0[mds_scores_t0$treatment == "log", ]
      open <- mds_scores_t0[mds_scores_t0$treatment == "open", ]

      log_hull <- log[chull(log[[paste0("NMDS", i)]], log[[paste0("NMDS", j)]]), ]
      open_hull <- open[chull(open[[paste0("NMDS", i)]], open[[paste0("NMDS", j)]]), ]
```

```

hulldat <- rbind(log_hull, open_hull)

plot_name <- paste0(i, "+", j)

plots[[plot_name]] <- ggplot() +
  theme_bw() +
  theme(panel.background = element_blank(),
        panel.grid.major = element_blank(),
        panel.grid.minor = element_blank(),
        plot.background = element_blank(),
        axis.text = element_text(size = 15),
        axis.title = element_text(size = 15),
        legend.title = element_text(size = 15),
        legend.text = element_text(size = 10)) +
  geom_text_repel(data = species.scores, aes(x = !!sym(paste0("NMDS", i)), y = !!sym(paste0("NMDS", j))), fill = "#63A088", size = 10) +
  geom_polygon(data = hulldat, aes(x = !!sym(paste0("NMDS", i)), y = !!sym(paste0("NMDS", j))), fill = "#56638A", size = 1) +
  geom_point(data = mds_scores_t0, aes(x = !!sym(paste0("NMDS", i)), y = !!sym(paste0("NMDS", j))), size = 10) +
  scale_colour_manual(values = c("#63A088", "#56638A"), name = "Treatment") +
  xlab(paste0("NMDS", i)) +
  ylab(paste0("NMDS", j))
}

}

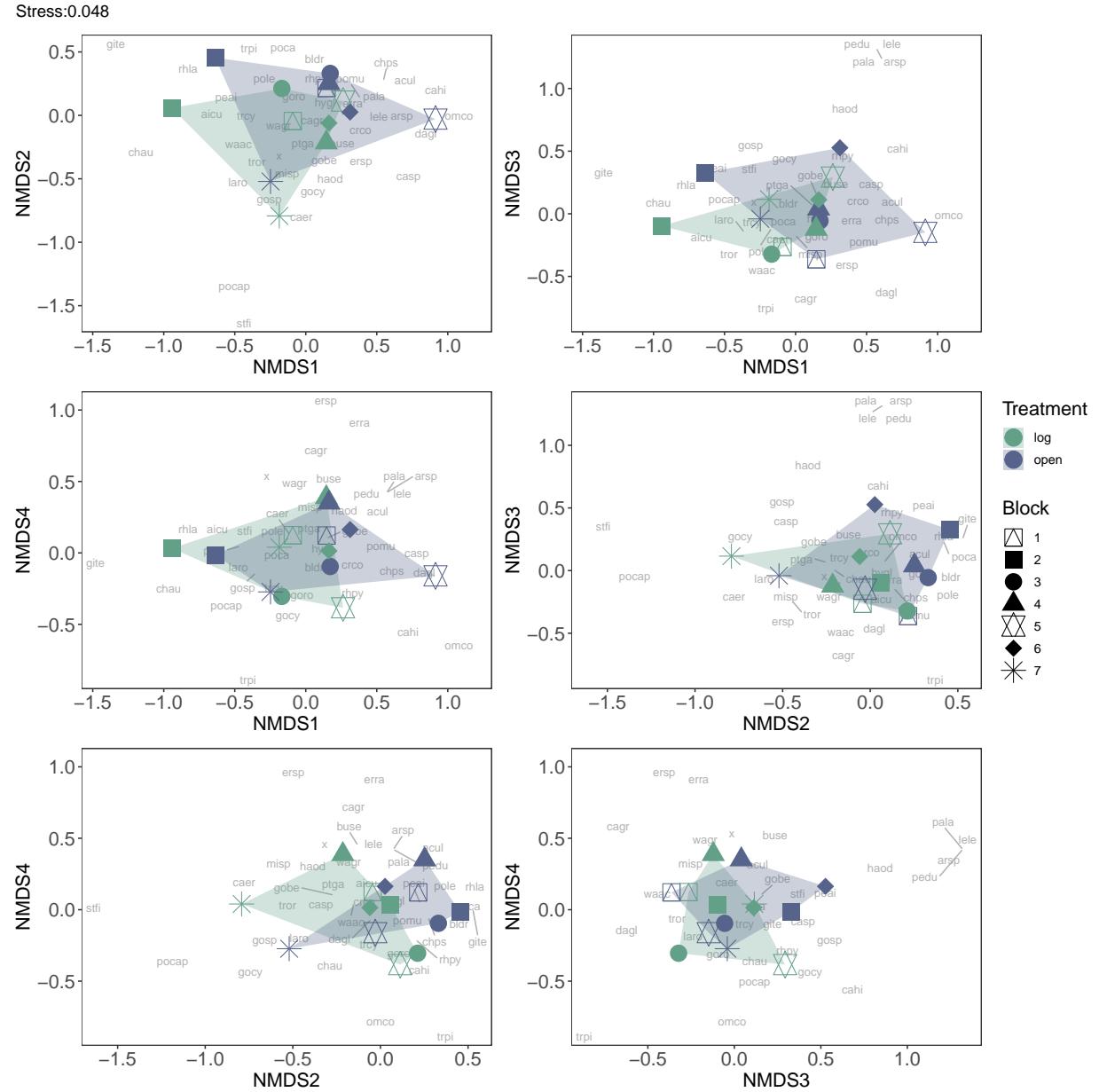
((plots$`1+2` + plots$`1+3`)/(plots$`1+4` + plots$`2+3`)/(plots$`2+4` + plots$`3+4`)) + plot_layout(guides = "none")
}

## Warning: ggrepel: 1 unlabeled data points (too many overlaps). Consider increasing max.overlaps
## ggrepel: 1 unlabeled data points (too many overlaps). Consider increasing max.overlaps

## Warning: ggrepel: 2 unlabeled data points (too many overlaps). Consider
## increasing max.overlaps

## Warning: ggrepel: 9 unlabeled data points (too many overlaps). Consider
## increasing max.overlaps

```



Composition dissimilarity * 2021 *

```
# subset data where all to communities, insitu log and insitu open communities at t1 and t2 are included
mat2 <- mat[which(mat$time=="t1" & mat$treatment=="open" | mat$time=="t1" & mat$treatment=="insitu_log")]
mat2$grp<-apply(mat2[c(89,91)], 1, paste, collapse=":") # block, init as grouping
names(mat2) #check
```

```
## [1] "acul"      "aicu"      "arca"      "ardy"      "arsp"      "auel"
## [7] "bldr"      "blrd"      "brdi"      "brdr"      "brpe"      "brru"
## [13] "buse"      "caer"      "cagr"      "cahi"      "casp"      "cear"
## [19] "chau"      "chei"      "chsps"     "crcl"      "crc"      "cusc"
## [25] "cusp"      "dagl"      "dosp"      "ento"      "erau"      "ercy"
## [31] "erra"      "ersp"      "gite"      "gnte"      "gobe"      "gocy"
## [37] "gono"      "goro"      "gosp"      "haod"      "hygl"      "hyp"  
```

```

## [43] "hypo"      "jubu"       "laro"       "ledu"       "lele"       "loef"
## [49] "misp"        "mite"       "momo"       "mopa"       "niro"       "omco"
## [55] "orsp"        "pala"       "peai"       "pedu"       "phsu"       "plde"
## [61] "poar"        "poca"       "pocap"      "poce"       "pogn"       "pole"
## [67] "pomu"        "pter"       "ptga"       "ptob"       "rhl"       "rhpy"
## [73] "rhsp"        "ry"         "scna"       "sino"       "sool"       "stfi"
## [79] "stpi"        "thma"       "trcy"       "tris"       "tror"       "trpi"
## [85] "waac"        "wagr"       "x"          "time"      "block"      "transect"
## [91] "init"        "treatment" "grp"

# another df where the grouping variables are time, block and initial state
# each row is a transect in a certain year.
df<-mat2[,c(1:87, 93)]
df2 = df %>% mutate(across(.cols=1:87, .fns=as.numeric)) # make everything numeric
rownames(df2)<-NULL # remove rownames

## new with group vars
nublock<-separate(df2, 88, c("block", "init"), ":") # just looking at time, block & initial treatment

# want to sum across transects in same block X init treatment
nublock$sumgrp<-apply(mat2[c(89, 91)], 1, paste, collapse=":")
head(nublock)

```

```

##   acul aicu arca ardy arsp auel bldr blrd brdi brdr brpe brru buse caer cagr
## 1   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
## 2   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
## 3   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   2   2
## 4   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
## 5   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
## 6   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
##   cahi casp clear chau chei chps crcl crco cusc cusp dagl dosp ento erau ercy
## 1   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
## 2   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   1
## 3   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
## 4   0   0   0   0   0   0   0   0   3   0   0   0   0   0   0   0   0   0
## 5   0   0   0   0   0   0   0   0   1   0   0   0   0   0   0   0   0   0
## 6   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
##   erra ersp gite gnte gobe gocy gono goro gosp haad hygl hypi hypo jubu laro
## 1   0   0   0   0   0   0   0   0   1   0   0   0   0   0   0   0   0   0
## 2   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
## 3   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
## 4   0   0   0   0   2   0   0   0   0   0   0   0   0   0   0   0   0   0
## 5   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
## 6   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
##   ledu lele loef misp mite momo mopa niro omco orsp pala peai pedu phsu plde
## 1   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
## 2   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
## 3   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
## 4   0   0   0   1   0   0   0   0   0   1   0   0   0   0   0   0   0   0
## 5   0   0   0   0   0   0   0   0   0   0   0   0   0   0   3   0   0   0
## 6   0   0   0   3   0   0   0   0   0   0   0   0   0   0   0   0   0   0
##   poar poca pocap poce pogn pole pomu pter ptga ptob rhl rhpy rhsp ry scna
## 1   0   0   0   0   0   1   5   0   0   0   0   0   0   0   0   0   0   0
## 2   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0

```

```

## 3   1   0   0   0   1   0   0   0   0   0   0   0   0   0   0
## 4   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
## 5   0   0   0   0   0   0   5   0   0   0   0   0   0   0   0   0
## 6   0   0   0   0   0   0   6   0   0   0   0   0   0   0   0   0
##   sino sool stfi stpi thma trcy tris tror trpi waac wagr x block init sumgrp
## 1   0   0   0   0   0   0   0   0   0   0   0   0   0   0   1   log 1:log
## 2   0   0   0   0   0   0   0   1   0   2   0   0   0   0   1   log 1:log
## 3   0   0   0   0   0   0   0   0   0   0   0   0   0   0   1   log 1:log
## 4   0   0   0   0   0   0   0   0   0   0   0   0   0   0   1   open 1:open
## 5   0   0   0   0   0   0   0   0   0   0   0   0   0   0   1   open 1:open
## 6   0   0   0   0   0   0   0   0   0   0   0   0   0   0   1   log 1:log

# sum observations across initial X block (group variable)
# this gives number of plants in each transect TYPE for each year in each block. should be 2 types X 3
blocksum<-rowsum(nublock[,c(1:87)], group=nublock$sumgrp)
blocksum$grps<-rownames(blocksum)
rownames(blocksum)<-NULL # remove rownames
nrow(blocksum) # it is 14 rows as expected

## [1] 14

## expand again
blocksum<-separate(blocksum, 88, c("block", "init"), ":")

# at the moment this includes where there were no plants ("x" column in matrix)
assemblies_t1<-blocksum[,c(1:87)]

# group - these are the treatment variables that need to be separately fed into the MDS analysis from
group_init<-blocksum$init
group_block<-blocksum$block

# MDS
ass.rel.t1<-decostand(assemblies_t1, method='hel') #standardize assemblies
ass.rel.t1_NMS <- metaMDS(ass.rel.t1, distance = 'bray', k = 5) # run MDS

## Run 0 stress 0.04139077
## Run 1 stress 0.04139111
## ... Procrustes: rmse 0.0003705647 max resid 0.0006272768
## ... Similar to previous best
## Run 2 stress 0.04139099
## ... Procrustes: rmse 0.000288366 max resid 0.0004733431
## ... Similar to previous best
## Run 3 stress 0.04139308
## ... Procrustes: rmse 0.0009120756 max resid 0.001499642
## ... Similar to previous best
## Run 4 stress 0.04139107
## ... Procrustes: rmse 0.0003493975 max resid 0.0005642936
## ... Similar to previous best
## Run 5 stress 0.041391
## ... Procrustes: rmse 0.0002931774 max resid 0.0004831762
## ... Similar to previous best
## Run 6 stress 0.04139102
## ... Procrustes: rmse 0.0002978652 max resid 0.0004903305

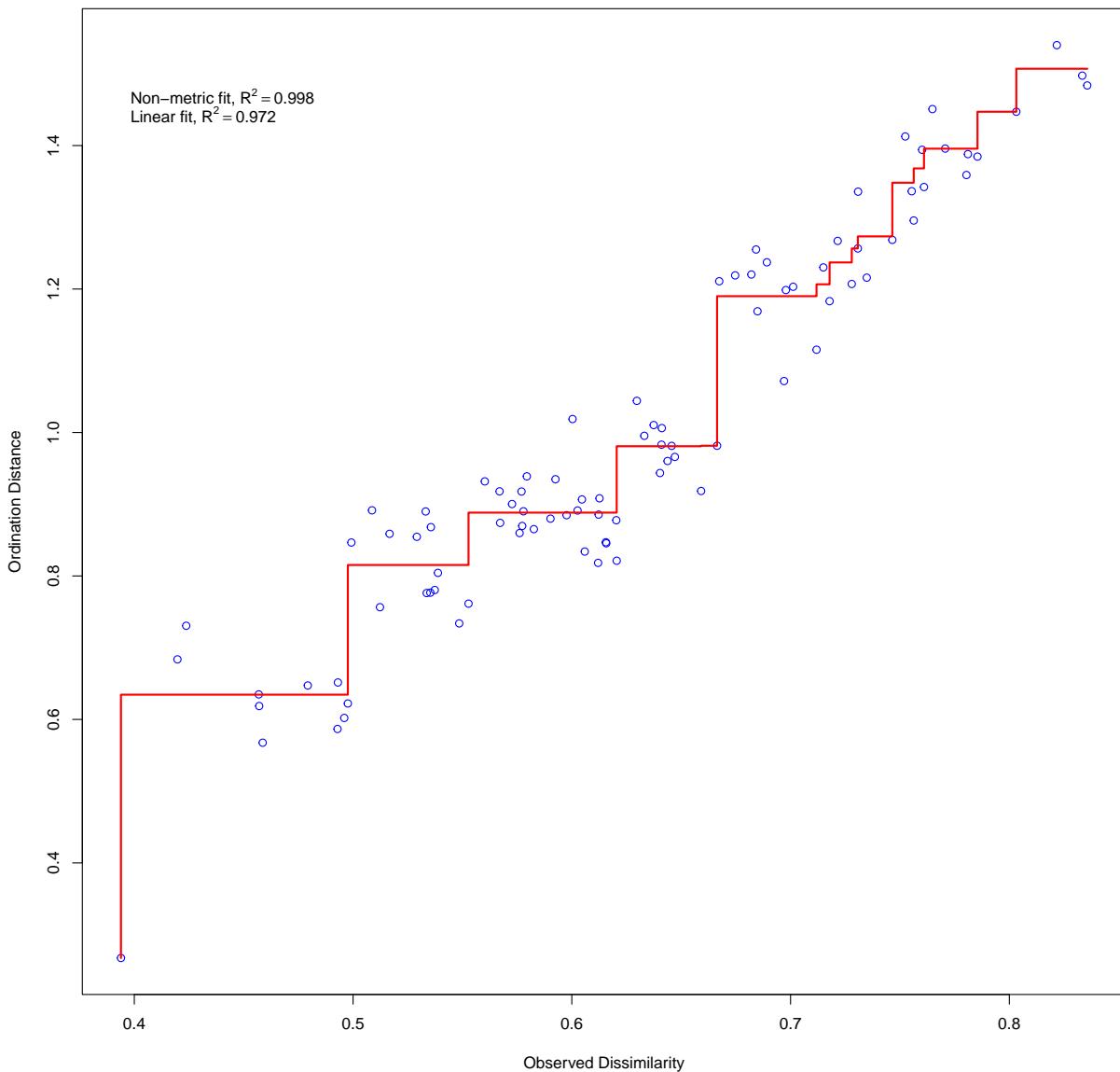
```

```

## ... Similar to previous best
## Run 7 stress 0.04139099
## ... Procrustes: rmse 0.0002837318 max resid 0.0004657474
## ... Similar to previous best
## Run 8 stress 0.04139094
## ... Procrustes: rmse 0.0002448639 max resid 0.0004026579
## ... Similar to previous best
## Run 9 stress 0.04139087
## ... Procrustes: rmse 0.0001315065 max resid 0.0002146393
## ... Similar to previous best
## Run 10 stress 0.04139099
## ... Procrustes: rmse 0.000288938 max resid 0.0004798391
## ... Similar to previous best
## Run 11 stress 0.04139095
## ... Procrustes: rmse 0.0005089322 max resid 0.0008047868
## ... Similar to previous best
## Run 12 stress 0.04139213
## ... Procrustes: rmse 0.0008712126 max resid 0.001430457
## ... Similar to previous best
## Run 13 stress 0.04139108
## ... Procrustes: rmse 0.0006201915 max resid 0.0009644615
## ... Similar to previous best
## Run 14 stress 0.04139852
## ... Procrustes: rmse 0.002266962 max resid 0.003703255
## ... Similar to previous best
## Run 15 stress 0.04139102
## ... Procrustes: rmse 0.000580698 max resid 0.0009089007
## ... Similar to previous best
## Run 16 stress 0.04139086
## ... Procrustes: rmse 0.0004318396 max resid 0.000676505
## ... Similar to previous best
## Run 17 stress 0.04139097
## ... Procrustes: rmse 0.0005376237 max resid 0.000838223
## ... Similar to previous best
## Run 18 stress 0.04139081
## ... Procrustes: rmse 0.0002718303 max resid 0.0003362656
## ... Similar to previous best
## Run 19 stress 0.0413912
## ... Procrustes: rmse 0.0004312002 max resid 0.0007285275
## ... Similar to previous best
## Run 20 stress 0.04139076
## ... New best solution
## ... Procrustes: rmse 1.929166e-05 max resid 3.184763e-05
## ... Similar to previous best
## *** Best solution repeated 1 times

```

```
stressplot(ass.rel.t1_NMS) # check fit
```



```

# scores
mds_scores_t1<-as.data.frame(vegan::scores(ass.rel.t1_NMS)$sites) # extract scores
mds_scores_t1$site<-rownames(vegan::scores(ass.rel.t1_NMS)$sites) # extract names
mds_scores_t1$treatment<-group_init # grouping factor 1
mds_scores_t1$block<-group_block # grouping factor 2

# explaining factors
init<-as.factor(group_init) # grouping factor 1- convert to factor
block<-as.factor(group_block) # grouping factor 2- convert to factor

##### rda model analysis & results #####
# can look at significance of model where initial treatment and block explain variation in community
trt_tot_2<-rda(ass.rel.t1~init+block) # run model using standardized data
summary(trt_tot_2)
  
```

```

##
## Call:
## rda(formula = ass.rel.t1 ~ init + block)
##
## Partitioning of variance:
##           Inertia Proportion
## Total      0.5326    1.0000
## Constrained 0.3506    0.6582
## Unconstrained 0.1821    0.3418
##
## Eigenvalues, and their contribution to the variance
##
## Importance of components:
##           RDA1     RDA2     RDA3     RDA4     RDA5     RDA6     RDA7
## Eigenvalue      0.09831 0.07256 0.0594 0.04249 0.03790 0.02137 0.01854
## Proportion Explained 0.18458 0.13623 0.1115 0.07977 0.07115 0.04012 0.03482
## Cumulative Proportion 0.18458 0.32081 0.4323 0.51209 0.58325 0.62337 0.65818
##           PC1      PC2      PC3      PC4      PC5      PC6
## Eigenvalue      0.05497 0.03785 0.02947 0.02799 0.02046 0.01132
## Proportion Explained 0.10320 0.07107 0.05533 0.05254 0.03842 0.02125
## Cumulative Proportion 0.76139 0.83246 0.88779 0.94033 0.97875 1.00000
##
## Accumulated constrained eigenvalues
## Importance of components:
##           RDA1     RDA2     RDA3     RDA4     RDA5     RDA6     RDA7
## Eigenvalue      0.09831 0.07256 0.0594 0.04249 0.0379 0.02137 0.01854
## Proportion Explained 0.28043 0.20698 0.1694 0.12119 0.1081 0.06096 0.05290
## Cumulative Proportion 0.28043 0.48742 0.6568 0.77804 0.8861 0.94710 1.00000
##
## Scaling 2 for species and site scores
## * Species are scaled proportional to eigenvalues
## * Sites are unscaled: weighted dispersion equal on all dimensions
## * General scaling constant of scores: 1.62215
##
##
## Species scores
##
##           RDA1     RDA2     RDA3     RDA4     RDA5     RDA6
## acul -9.964e-02 4.823e-02 -2.106e-02 1.277e-01 1.875e-02 -7.699e-02
## aicu 8.907e-17 -5.123e-17 8.326e-17 6.638e-17 -2.673e-17 1.400e-17
## arca 2.296e-17 9.720e-17 -3.409e-17 4.292e-17 -4.580e-17 -1.032e-17
## ardy -4.306e-02 1.526e-03 -1.411e-03 -1.678e-02 -4.191e-02 -3.620e-02
## arsp -2.461e-17 4.760e-17 3.101e-17 -6.840e-18 4.536e-17 3.563e-17
## auel 6.616e-17 -4.835e-17 -1.821e-17 -4.209e-18 1.394e-17 2.789e-17
## bldr 4.844e-03 6.118e-03 9.056e-02 -8.132e-02 1.733e-01 9.784e-02
## blrd -4.559e-02 4.352e-02 -9.443e-03 1.226e-02 4.624e-02 8.270e-03
## brdi -3.456e-17 4.651e-17 3.643e-17 1.080e-17 -2.048e-17 7.466e-18
## brdr 4.095e-18 2.731e-18 7.681e-18 7.980e-18 -1.132e-19 7.206e-18
## brpe 3.360e-02 -9.795e-02 1.335e-01 -1.875e-02 3.238e-02 -2.280e-02
## brru -3.432e-34 -5.429e-35 1.158e-33 -1.156e-34 -2.427e-34 -5.492e-34
## buse -2.227e-32 6.989e-33 3.781e-32 -7.980e-33 -1.210e-32 -2.143e-32
## caer -1.023e-01 -6.797e-04 3.047e-02 8.609e-02 -1.012e-01 -7.420e-02

```

```

## cagr  5.897e-02 -1.425e-01  4.363e-02  4.572e-02 -6.406e-02  6.086e-02
## cahi  0.000e+00  0.000e+00  0.000e+00  0.000e+00  0.000e+00  0.000e+00
## casp -3.404e-02  3.541e-04  4.467e-03  5.540e-02 -1.265e-02  1.676e-02
## cear  4.342e-02 -1.362e-02 -7.371e-02  1.556e-02  2.358e-02  4.177e-02
## chau  1.161e-01  1.608e-01 -1.308e-01 -4.803e-02 -5.147e-02 -5.783e-03
## chei  0.000e+00  0.000e+00  0.000e+00  0.000e+00  0.000e+00  0.000e+00
## chps -2.651e-01  1.313e-01 -8.485e-02  7.546e-02  7.404e-02  2.339e-02
## crcl  0.000e+00  0.000e+00  0.000e+00  0.000e+00  0.000e+00  0.000e+00
## crco -3.108e-02  7.252e-02 -7.821e-02  1.545e-02 -6.801e-02  9.685e-02
## cusc  0.000e+00  0.000e+00  0.000e+00  0.000e+00  0.000e+00  0.000e+00
## cusp -9.546e-02 -1.913e-02  3.033e-02 -5.482e-02  1.679e-02 -8.577e-02
## dagl  0.000e+00  0.000e+00  0.000e+00  0.000e+00  0.000e+00  0.000e+00
## dosp  0.000e+00  0.000e+00  0.000e+00  0.000e+00  0.000e+00  0.000e+00
## ento  0.000e+00  0.000e+00  0.000e+00  0.000e+00  0.000e+00  0.000e+00
## erau  0.000e+00  0.000e+00  0.000e+00  0.000e+00  0.000e+00  0.000e+00
## ercy -1.494e-02 -9.609e-02  7.175e-02 -2.555e-03  3.268e-02 -1.492e-02
## erra -2.518e-02  2.121e-02 -1.602e-02  3.898e-02  1.386e-02 -1.231e-02
## ersp  0.000e+00  0.000e+00  0.000e+00  0.000e+00  0.000e+00  0.000e+00
## gite  1.098e-01  6.866e-02  5.880e-02 -6.813e-03 -4.310e-02 -3.238e-02
## gnre  7.822e-02  1.004e-01 -1.078e-01 -3.810e-02 -2.965e-02 -4.746e-03
## gobe  5.311e-02 -9.669e-02 -1.053e-01 -7.205e-03  1.105e-03  5.703e-02
## gocy  0.000e+00  0.000e+00  0.000e+00  0.000e+00  0.000e+00  0.000e+00
## gono  0.000e+00  0.000e+00  0.000e+00  0.000e+00  0.000e+00  0.000e+00
## goro  1.802e-01 -2.829e-02 -2.024e-01 -1.678e-01 -3.322e-02  1.123e-02
## gosp  0.000e+00  0.000e+00  0.000e+00  0.000e+00  0.000e+00  0.000e+00
## haod  6.923e-03 -4.618e-02 -1.119e-02 -3.158e-02  8.584e-03 -4.117e-02
## hygl -2.255e-01  9.594e-02 -1.277e-01 -4.435e-02  1.571e-01 -3.412e-02
## hypi -8.499e-03 -3.399e-02  4.724e-02 -2.940e-03 -2.928e-02 -3.990e-02
## hypo  0.000e+00  0.000e+00  0.000e+00  0.000e+00  0.000e+00  0.000e+00
## jubu  0.000e+00  0.000e+00  0.000e+00  0.000e+00  0.000e+00  0.000e+00
## laro  6.166e-02 -8.661e-02 -2.159e-02 -1.090e-02  3.024e-02 -1.992e-02
## ledu -3.340e-02  3.252e-02  5.320e-02 -6.207e-02  8.923e-02  4.682e-02
## lele -1.116e-02 -8.439e-03 -3.421e-02 -2.560e-02  2.133e-03 -1.973e-02
## loef  0.000e+00  0.000e+00  0.000e+00  0.000e+00  0.000e+00  0.000e+00
## misp  3.447e-02 -1.375e-01 -7.412e-02  2.140e-01 -1.428e-02 -1.892e-02
## mite  0.000e+00  0.000e+00  0.000e+00  0.000e+00  0.000e+00  0.000e+00
## momo  0.000e+00  0.000e+00  0.000e+00  0.000e+00  0.000e+00  0.000e+00
## mopa  0.000e+00  0.000e+00  0.000e+00  0.000e+00  0.000e+00  0.000e+00
## niro -6.420e-03  1.351e-03  6.208e-02 -3.434e-02  6.144e-02  2.506e-02
## omco -2.407e-02  7.323e-04 -1.139e-02  1.863e-03 -3.286e-02  3.819e-02
## orsp  0.000e+00  0.000e+00  0.000e+00  0.000e+00  0.000e+00  0.000e+00
## pala  0.000e+00  0.000e+00  0.000e+00  0.000e+00  0.000e+00  0.000e+00
## peai  2.708e-02  4.611e-02 -6.180e-02  1.195e-01  7.353e-02  5.837e-02
## pedu  0.000e+00  0.000e+00  0.000e+00  0.000e+00  0.000e+00  0.000e+00
## phsu -4.540e-03  9.554e-04  4.390e-02 -2.428e-02  4.344e-02  1.772e-02
## plde  0.000e+00  0.000e+00  0.000e+00  0.000e+00  0.000e+00  0.000e+00
## poar  3.129e-02  5.243e-02  1.267e-02 -3.757e-03 -7.738e-02  3.673e-02
## poca  0.000e+00  0.000e+00  0.000e+00  0.000e+00  0.000e+00  0.000e+00
## pocap 0.000e+00  0.000e+00  0.000e+00  0.000e+00  0.000e+00  0.000e+00
## poce  0.000e+00  0.000e+00  0.000e+00  0.000e+00  0.000e+00  0.000e+00
## pogn 0.000e+00  0.000e+00  0.000e+00  0.000e+00  0.000e+00  0.000e+00
## pole -1.059e-01  1.918e-01  2.540e-01  5.600e-02 -4.763e-02  3.381e-02
## pomu -3.790e-01 -2.547e-01 -2.332e-02 -1.004e-01 -9.950e-02  4.492e-02
## pter  0.000e+00  0.000e+00  0.000e+00  0.000e+00  0.000e+00  0.000e+00

```

```

## ptga  4.191e-03  1.826e-01 -6.538e-02  4.060e-02 -6.820e-02 -9.489e-04
## ptob  0.000e+00  0.000e+00  0.000e+00  0.000e+00  0.000e+00  0.000e+00
## rhla  0.000e+00  0.000e+00  0.000e+00  0.000e+00  0.000e+00  0.000e+00
## rhpy  7.480e-02 -1.733e-01 -5.570e-02  7.898e-03  3.486e-02 -8.805e-02
## rhsp  0.000e+00  0.000e+00  0.000e+00  0.000e+00  0.000e+00  0.000e+00
## ry    0.000e+00  0.000e+00  0.000e+00  0.000e+00  0.000e+00  0.000e+00
## scna  4.567e-02  4.156e-02 -4.560e-02 -1.815e-03 -7.993e-03  1.495e-02
## sino  0.000e+00  0.000e+00  0.000e+00  0.000e+00  0.000e+00  0.000e+00
## sool  6.923e-03 -4.618e-02 -1.119e-02 -3.158e-02  8.584e-03 -4.117e-02
## stfi  0.000e+00  0.000e+00  0.000e+00  0.000e+00  0.000e+00  0.000e+00
## stpi  0.000e+00  0.000e+00  0.000e+00  0.000e+00  0.000e+00  0.000e+00
## thma  0.000e+00  0.000e+00  0.000e+00  0.000e+00  0.000e+00  0.000e+00
## trcy  1.901e-01 -1.943e-02  1.190e-01 -8.643e-03  1.539e-01 -5.244e-02
## tris  -1.016e-02  3.179e-02 -1.165e-01  6.991e-02  1.015e-01 -5.902e-02
## tror  1.207e-01 -1.229e-01 -4.525e-02  1.878e-01  6.974e-02  8.917e-02
## trpi  1.553e-01  9.710e-02  8.316e-02 -9.635e-03 -6.095e-02 -4.579e-02
## waac  7.512e-02 -1.338e-02  6.506e-02  9.716e-04 -2.574e-02  9.506e-02
## wagr -4.651e-02  3.022e-02 -2.281e-02 -7.188e-02  5.822e-02  1.591e-02
## x     -2.407e-02  7.323e-04 -1.139e-02  1.863e-03 -3.286e-02  3.819e-02
##
##
## Site scores (weighted sums of species scores)
##
##          RDA1      RDA2      RDA3      RDA4      RDA5      RDA6
## row1  -0.14539 -0.50546  0.2599  0.216437 -0.91881  0.6085
## row2  -0.46559  0.09913 -0.1111 -0.007202 -0.30677  0.7744
## row3   0.80185  0.50763  0.5725 -0.332829 -0.22459 -0.2324
## row4   0.46163  0.92808 -0.3131  0.097676 -0.47194 -0.1184
## row5   0.14415  0.14924  0.7768 -0.434518  0.85157  0.3859
## row6  -0.63217  0.31773  0.0867 -0.509643  0.52588  0.3249
## row7  -0.26308 -0.04549  0.1058  0.779425  0.30886 -0.1555
## row8  -0.41799  0.46761 -0.1560  0.932183  0.32017 -0.4631
## row9   0.09703 -0.32699  0.8313  0.266940 -0.29875 -0.6554
## row10 -0.64372 -0.07686 -0.2627 -0.475378 -0.49259 -0.2084
## row11 -0.10067 -0.63812 -0.1985 -0.433033  0.20807 -0.9544
## row12 -0.05046 -0.16595 -0.6994 -0.550396 -0.04626 -0.0293
## row13  0.64528 -0.66391 -0.2563  0.413154  0.31358  0.5796
## row14  0.56913 -0.04664 -0.6360  0.037186  0.23158  0.1437
##
##
## Site constraints (linear combinations of constraining variables)
##
##          RDA1      RDA2      RDA3      RDA4      RDA5      RDA6
## row1  -0.13704 -0.420750  0.37318  0.17256 -0.57852  0.6309
## row2  -0.47394  0.014419 -0.22438  0.03668 -0.64707  0.7520
## row3   0.80019  0.500273  0.42845 -0.04964 -0.31399 -0.2359
## row4   0.46329  0.935442 -0.16911 -0.18552 -0.38254 -0.1149
## row5  -0.07556  0.015900  0.73055 -0.40414  0.72300  0.2949
## row6  -0.41246  0.451069  0.13299 -0.54002  0.65445  0.4159
## row7  -0.17209 -0.006526  0.27370  0.92374  0.34879 -0.3698
## row8  -0.50899  0.428644 -0.32386  0.78786  0.28024 -0.2488
## row9  -0.10489 -0.419510  0.58308 -0.03628 -0.36139 -0.4924
## row10 -0.44180  0.015660 -0.01448 -0.17216 -0.42994 -0.3714
## row11  0.09289 -0.619621 -0.15016 -0.42378  0.11518 -0.5524

```

```

## row12 -0.24402 -0.184451 -0.74771 -0.55965  0.04663 -0.4313
## row13  0.77566 -0.572860 -0.14735  0.29311  0.30686  0.3011
## row14  0.43875 -0.137690 -0.74491  0.15723  0.23831  0.4221
##
##
## Biplot scores for constraining variables
##
##          RDA1     RDA2     RDA3     RDA4     RDA5     RDA6
## initopen -0.38855  0.5019  -0.68916 -0.15671 -0.07906  0.1396
## block2   0.59489  0.6760  0.12211 -0.11072 -0.32795 -0.1652
## block3  -0.22977  0.2199  0.40658 -0.44454  0.64855  0.3347
## block4  -0.32067  0.1987 -0.02361  0.80588  0.29617 -0.2912
## block5  -0.25740 -0.1901  0.26772 -0.09814 -0.37259 -0.4067
## block6  -0.07116 -0.3786 -0.42275 -0.46303  0.07619 -0.4632
## block7   0.57178 -0.3346 -0.42011  0.21204  0.25668  0.3405
##
##
## Centroids for factor constraints
##
##          RDA1     RDA2     RDA3     RDA4     RDA5     RDA6
## initlog   0.16845 -0.2176  0.29878  0.06794  0.03428 -0.06053
## initopen -0.16845  0.2176 -0.29878 -0.06794 -0.03428  0.06053
## block1  -0.30549 -0.2032  0.07440  0.10462 -0.61279  0.69144
## block2   0.63174  0.7179  0.12967 -0.11758 -0.34827 -0.17539
## block3  -0.24401  0.2335  0.43177 -0.47208  0.68872  0.35539
## block4  -0.34054  0.2111 -0.02508  0.85580  0.31451 -0.30929
## block5  -0.27335 -0.2019  0.28430 -0.10422 -0.39567 -0.43190
## block6  -0.07556 -0.4020 -0.44893 -0.49171  0.08091 -0.49186
## block7   0.60720 -0.3553 -0.44613  0.22517  0.27258  0.36161

```

```
anova.cca(trt_tot_2, step=1000, by="term") ## test for model significance
```

```

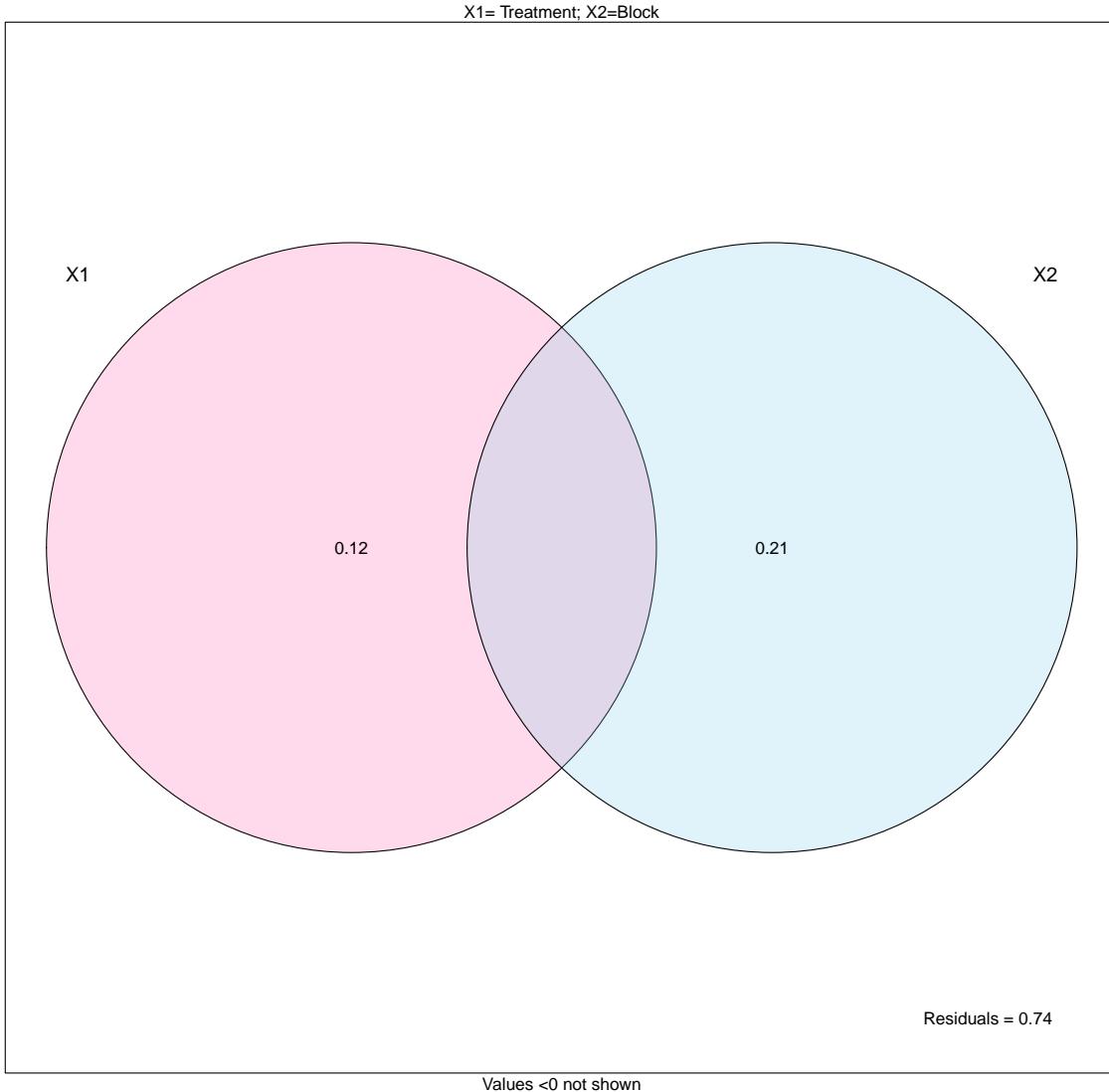
## Permutation test for rda under reduced model
## Terms added sequentially (first to last)
## Permutation: free
## Number of permutations: 999
##
## Model: rda(formula = ass.rel.t1 ~ init + block)
##          Df Variance    F Pr(>F)
## init      1 0.064359 2.121  0.006 **
## block     6 0.286205 1.572  0.002 **
## Residual  6 0.182059
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

```

```

# can model using varpart to look at contributions of initial treatment and block
var.mod2<-varpart(ass.rel.t1, init, block) # run model on standardized data
plot(var.mod2, bg=c("hotpink","skyblue"))
mtext("X1= Treatment; X2=Block", side=3)

```



```
## can test for significance of contribution of the fraction of initial treatment
# do this with partial redundancy analysis
trt_Frac<-rda(ass.rel.t1, init, block) # partial rda model
summary(trt_Frac)
```

```
##
## Call:
## rda(X = ass.rel.t1, Y = init, Z = block)
##
## Partitioning of variance:
##                 Inertia Proportion
## Total          0.53262   1.0000
## Conditioned   0.28621   0.5373
```

```

## Constrained 0.06436      0.1208
## Unconstrained 0.18206     0.3418
##
## Eigenvalues, and their contribution to the variance
## after removing the contribution of conditiniong variables
##
## Importance of components:
##          RDA1      PC1      PC2      PC3      PC4      PC5      PC6
## Eigenvalue      0.06436 0.05497 0.03785 0.02947 0.02799 0.02046 0.01132
## Proportion Explained 0.26118 0.22307 0.15362 0.11959 0.11357 0.08304 0.04593
## Cumulative Proportion 0.26118 0.48425 0.63787 0.75745 0.87103 0.95407 1.00000
##
## Accumulated constrained eigenvalues
## Importance of components:
##          RDA1
## Eigenvalue      0.06436
## Proportion Explained 1.00000
## Cumulative Proportion 1.00000
##
## Scaling 2 for species and site scores
## * Species are scaled proportional to eigenvalues
## * Sites are unscaled: weighted dispersion equal on all dimensions
## * General scaling constant of scores: 1.62215
##
##
## Species scores
##
##          RDA1      PC1      PC2      PC3      PC4      PC5
## acul   0.055571 3.637e-02 4.328e-02 1.783e-02 5.246e-03 5.454e-04
## aicu   0.000000 8.472e-17 -2.496e-17 -1.813e-17 2.580e-17 -2.616e-17
## arca   0.000000 9.597e-18 2.861e-18 1.596e-17 -2.682e-18 -9.970e-19
## ardy   0.042259 5.025e-02 5.860e-02 1.652e-02 4.875e-02 2.875e-02
## arsp   0.000000 1.984e-17 3.886e-17 5.839e-17 -3.819e-17 1.599e-17
## auel   0.000000 9.860e-19 -4.413e-18 8.306e-18 -3.824e-18 -6.153e-18
## bldr   -0.039796 -4.819e-03 3.972e-02 -4.794e-02 -2.337e-02 2.293e-02
## blrd   0.042899 4.726e-03 1.765e-02 2.217e-02 -5.565e-02 -2.605e-02
## brdi   0.000000 -1.170e-17 -1.637e-17 -2.608e-17 -3.440e-18 -2.426e-17
## brdr   0.000000 -2.443e-19 4.135e-18 -1.727e-18 -3.647e-18 3.050e-18
## brpe   -0.128276 -8.636e-02 -6.995e-02 -9.422e-02 -6.120e-02 -1.120e-02
## brru   0.000000 -1.871e-34 5.118e-33 -4.301e-33 -3.629e-33 2.180e-33
## buse   0.000000 5.868e-34 -1.605e-32 1.349e-32 1.138e-32 -6.838e-33
## caer   0.011258 -7.733e-02 7.189e-02 4.870e-03 3.834e-02 -9.362e-02
## cagr   -0.095632 -8.997e-02 -1.221e-02 1.902e-02 -2.621e-02 -1.103e-01
## cahi   0.000000 0.000e+00 0.000e+00 0.000e+00 0.000e+00 0.000e+00
## casp   -0.003109 1.770e-02 -2.479e-02 -2.422e-02 2.065e-02 6.085e-02
## cear   0.042899 2.468e-03 -6.751e-02 5.673e-02 4.787e-02 -2.876e-02
## chau   0.140877 -1.835e-01 -1.649e-02 1.254e-02 8.452e-02 1.746e-02
## chei   0.000000 0.000e+00 0.000e+00 0.000e+00 0.000e+00 0.000e+00
## chps   0.213452 9.333e-02 4.706e-02 -1.968e-02 -1.078e-01 1.722e-02
## crcl   0.000000 0.000e+00 0.000e+00 0.000e+00 0.000e+00 0.000e+00
## crco   0.138108 1.593e-01 -8.706e-02 4.536e-02 -1.939e-02 3.021e-02
## cusc   0.000000 0.000e+00 0.000e+00 0.000e+00 0.000e+00 0.000e+00
## cusp   0.023897 4.130e-03 2.468e-02 9.589e-02 -9.788e-02 9.619e-03
## dagl   0.000000 0.000e+00 0.000e+00 0.000e+00 0.000e+00 0.000e+00

```

## dosp	0.000000	0.000e+00	0.000e+00	0.000e+00	0.000e+00	0.000e+00
## ento	0.000000	0.000e+00	0.000e+00	0.000e+00	0.000e+00	0.000e+00
## erau	0.000000	0.000e+00	0.000e+00	0.000e+00	0.000e+00	0.000e+00
## ercy	-0.129183	-3.188e-02	1.257e-02	8.821e-02	5.864e-02	3.511e-02
## erra	0.021450	-2.405e-03	-1.984e-03	4.725e-03	-3.054e-02	-2.041e-02
## ersp	0.000000	0.000e+00	0.000e+00	0.000e+00	0.000e+00	0.000e+00
## gite	-0.059503	1.330e-01	-2.802e-02	-9.877e-04	-2.136e-02	-4.809e-02
## gnte	0.105088	-1.192e-01	-1.940e-02	3.043e-03	6.651e-02	-1.381e-03
## gobe	0.009616	2.117e-02	6.289e-02	-1.833e-01	-6.378e-02	4.242e-02
## gocy	0.000000	0.000e+00	0.000e+00	0.000e+00	0.000e+00	0.000e+00
## gono	0.000000	0.000e+00	0.000e+00	0.000e+00	0.000e+00	0.000e+00
## goro	0.087805	8.971e-02	6.801e-02	-8.319e-02	1.119e-01	-7.619e-02
## gosp	0.000000	0.000e+00	0.000e+00	0.000e+00	0.000e+00	0.000e+00
## haod	-0.032312	-2.971e-03	-4.026e-03	6.188e-02	-2.109e-02	4.142e-02
## hygl	0.215609	4.905e-02	1.609e-03	6.009e-02	1.333e-02	5.224e-02
## hypi	-0.035126	-4.177e-02	-4.871e-02	-1.374e-02	-4.052e-02	-2.390e-02
## hypo	0.000000	0.000e+00	0.000e+00	0.000e+00	0.000e+00	0.000e+00
## jubu	0.000000	0.000e+00	0.000e+00	0.000e+00	0.000e+00	0.000e+00
## laro	-0.062906	-4.731e-03	4.412e-02	2.142e-02	-5.523e-02	6.194e-02
## ledu	0.004284	1.424e-03	3.922e-03	3.485e-03	-5.014e-03	-1.127e-03
## lele	0.019834	1.824e-03	2.471e-03	-3.798e-02	1.295e-02	-2.543e-02
## loef	0.000000	0.000e+00	0.000e+00	0.000e+00	0.000e+00	0.000e+00
## misp	-0.077168	-1.527e-01	1.772e-02	-2.016e-02	-2.886e-02	-4.694e-02
## mite	0.000000	0.000e+00	0.000e+00	0.000e+00	0.000e+00	0.000e+00
## momo	0.000000	0.000e+00	0.000e+00	0.000e+00	0.000e+00	0.000e+00
## mopa	0.000000	0.000e+00	0.000e+00	0.000e+00	0.000e+00	0.000e+00
## niro	-0.036841	-1.225e-02	-3.373e-02	-2.997e-02	4.312e-02	9.689e-03
## omco	0.022017	1.488e-02	-2.711e-02	-1.869e-02	-1.512e-02	3.695e-02
## orsp	0.000000	0.000e+00	0.000e+00	0.000e+00	0.000e+00	0.000e+00
## pala	0.000000	0.000e+00	0.000e+00	0.000e+00	0.000e+00	0.000e+00
## peai	0.031786	-6.717e-02	-6.703e-02	-9.866e-02	1.006e-01	1.083e-01
## pedu	0.000000	0.000e+00	0.000e+00	0.000e+00	0.000e+00	0.000e+00
## phsu	-0.026050	-8.660e-03	-2.385e-02	-2.119e-02	3.049e-02	6.851e-03
## plde	0.000000	0.000e+00	0.000e+00	0.000e+00	0.000e+00	0.000e+00
## poar	0.006522	-1.086e-01	5.808e-02	2.807e-02	3.614e-02	-2.284e-02
## poca	0.000000	0.000e+00	0.000e+00	0.000e+00	0.000e+00	0.000e+00
## pocap	0.000000	0.000e+00	0.000e+00	0.000e+00	0.000e+00	0.000e+00
## poce	0.000000	0.000e+00	0.000e+00	0.000e+00	0.000e+00	0.000e+00
## pogn	0.000000	0.000e+00	0.000e+00	0.000e+00	0.000e+00	0.000e+00
## pole	-0.037141	-1.016e-01	-9.293e-02	-3.438e-02	-7.013e-02	-1.911e-02
## pomu	0.047780	9.595e-02	1.336e-01	5.875e-02	9.945e-02	8.749e-02
## pter	0.000000	0.000e+00	0.000e+00	0.000e+00	0.000e+00	0.000e+00
## ptga	0.126430	-1.396e-01	1.126e-03	-9.388e-03	-4.387e-02	5.596e-02
## ptob	0.000000	0.000e+00	0.000e+00	0.000e+00	0.000e+00	0.000e+00
## rhla	0.000000	0.000e+00	0.000e+00	0.000e+00	0.000e+00	0.000e+00
## rhpy	-0.080545	-6.671e-02	-9.829e-02	1.253e-01	-1.163e-01	2.039e-02
## rhsp	0.000000	0.000e+00	0.000e+00	0.000e+00	0.000e+00	0.000e+00
## ry	0.000000	0.000e+00	0.000e+00	0.000e+00	0.000e+00	0.000e+00
## scna	0.043870	-4.887e-02	-2.320e-02	2.874e-02	3.198e-02	3.740e-03
## sino	0.000000	0.000e+00	0.000e+00	0.000e+00	0.000e+00	0.000e+00
## sool	-0.032312	-2.971e-03	-4.026e-03	6.188e-02	-2.109e-02	4.142e-02
## stfi	0.000000	0.000e+00	0.000e+00	0.000e+00	0.000e+00	0.000e+00
## stpi	0.000000	0.000e+00	0.000e+00	0.000e+00	0.000e+00	0.000e+00
## thma	0.000000	0.000e+00	0.000e+00	0.000e+00	0.000e+00	0.000e+00

```

## trcy -0.192212 8.297e-02 -2.570e-01 -7.513e-02 4.924e-02 3.987e-03
## tris 0.061108 -4.018e-02 -3.765e-02 9.554e-02 4.144e-02 1.226e-02
## tror -0.092188 -2.616e-02 1.417e-01 -5.750e-02 -5.785e-02 -1.555e-02
## trpi -0.084150 1.881e-01 -3.963e-02 -1.397e-03 -3.021e-02 -6.801e-02
## waac -0.075766 3.898e-02 -3.149e-02 5.714e-02 8.339e-02 -1.179e-01
## wagr 0.056986 1.417e-02 3.648e-02 -7.759e-03 -3.054e-02 -3.520e-02
## x 0.022017 1.488e-02 -2.711e-02 -1.869e-02 -1.512e-02 3.695e-02
##
##
## Site scores (weighted sums of species scores)
##
##          RDA1      PC1      PC2      PC3      PC4      PC5
## sit1 -0.3912 -0.29299  0.53388  0.367996  0.2978 -0.7276
## sit2  0.3912  0.29299 -0.53388 -0.367996 -0.2978  0.7276
## sit3 -0.4999  0.96887 -0.20417 -0.007197 -0.1556 -0.3504
## sit4  0.4999 -0.96887  0.20417  0.007197  0.1556  0.3504
## sit5 -0.5201 -0.14413 -0.39689 -0.352694  0.5075  0.1140
## sit6  0.5201  0.14413  0.39689  0.352694 -0.5075 -0.1140
## sit7 -0.2983  0.04860  0.04011 -0.095492  0.6173  0.4125
## sit8  0.2983 -0.04860 -0.04011  0.095492 -0.6173 -0.4125
## sit9 -0.6722 -0.51555 -0.60119 -0.169521 -0.5001 -0.2950
## sit10 0.6722  0.51555  0.60119  0.169521  0.5001  0.2950
## sit11 -0.3244 -0.03987 -0.05401  0.830224 -0.2830  0.5558
## sit12  0.3244  0.03987  0.05401 -0.830224  0.2830 -0.5558
## sit13 -0.3287 -0.02494  0.68227 -0.573317 -0.4838  0.2906
## sit14  0.3287  0.02494 -0.68227  0.573317  0.4838 -0.2906
##
##
## Site constraints (linear combinations of constraining variables)
##
##          RDA1      PC1      PC2      PC3      PC4      PC5
## con1 -0.4335 -0.29299  0.53388  0.367996  0.2978 -0.7276
## con2  0.4335  0.29299 -0.53388 -0.367996 -0.2978  0.7276
## con3 -0.4335  0.96887 -0.20417 -0.007197 -0.1556 -0.3504
## con4  0.4335 -0.96887  0.20417  0.007197  0.1556  0.3504
## con5 -0.4335 -0.14413 -0.39689 -0.352694  0.5075  0.1140
## con6  0.4335  0.14413  0.39689  0.352694 -0.5075 -0.1140
## con7 -0.4335  0.04860  0.04011 -0.095492  0.6173  0.4125
## con8  0.4335 -0.04860 -0.04011  0.095492 -0.6173 -0.4125
## con9 -0.4335 -0.51555 -0.60119 -0.169521 -0.5001 -0.2950
## con10 0.4335  0.51555  0.60119  0.169521  0.5001  0.2950
## con11 -0.4335 -0.03987 -0.05401  0.830224 -0.2830  0.5558
## con12  0.4335  0.03987  0.05401 -0.830224  0.2830 -0.5558
## con13 -0.4335 -0.02494  0.68227 -0.573317 -0.4838  0.2906
## con14  0.4335  0.02494 -0.68227  0.573317  0.4838 -0.2906
##
##
## Biplot scores for constraining variables
##
##          RDA1 PC1 PC2 PC3 PC4 PC5
## Yopen    1   0   0   0   0   0

```

```
RsquareAdj(trt_Frac)$adj.r.squared #explanatory power
```

```

## [1] 0.1186067

anova.cca(trt_Frac) ## this tells us if first condition, init, significantly contributes to overall var

## Permutation test for rda under reduced model
## Permutation: free
## Number of permutations: 999
##
## Model: rda(X = ass.rel.t1, Y = init, Z = block)
##          Df Variance      F Pr(>F)
## Model      1 0.064359 2.121  0.047 *
## Residual   6 0.182059
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

### extracting species scores and plotting
# species scores
species.scores<-as.data.frame(vegan::scores(ass.rel.t1_NMS,"species")) ## some species don't have scores
species.scores$species<-rownames(species.scores)

### NMDS 1 and 2
log<-mds_scores_t1[mds_scores_t1$treatment == "log", ][chull(mds_scores_t1[mds_scores_t1$treatment ==
"log", c("NMDS1", "NMDS2")])]

open<-mds_scores_t1[mds_scores_t1$treatment == "open", ][chull(mds_scores_t1[mds_scores_t1$treatment ==
"open", c("NMDS1", "NMDS2")])]

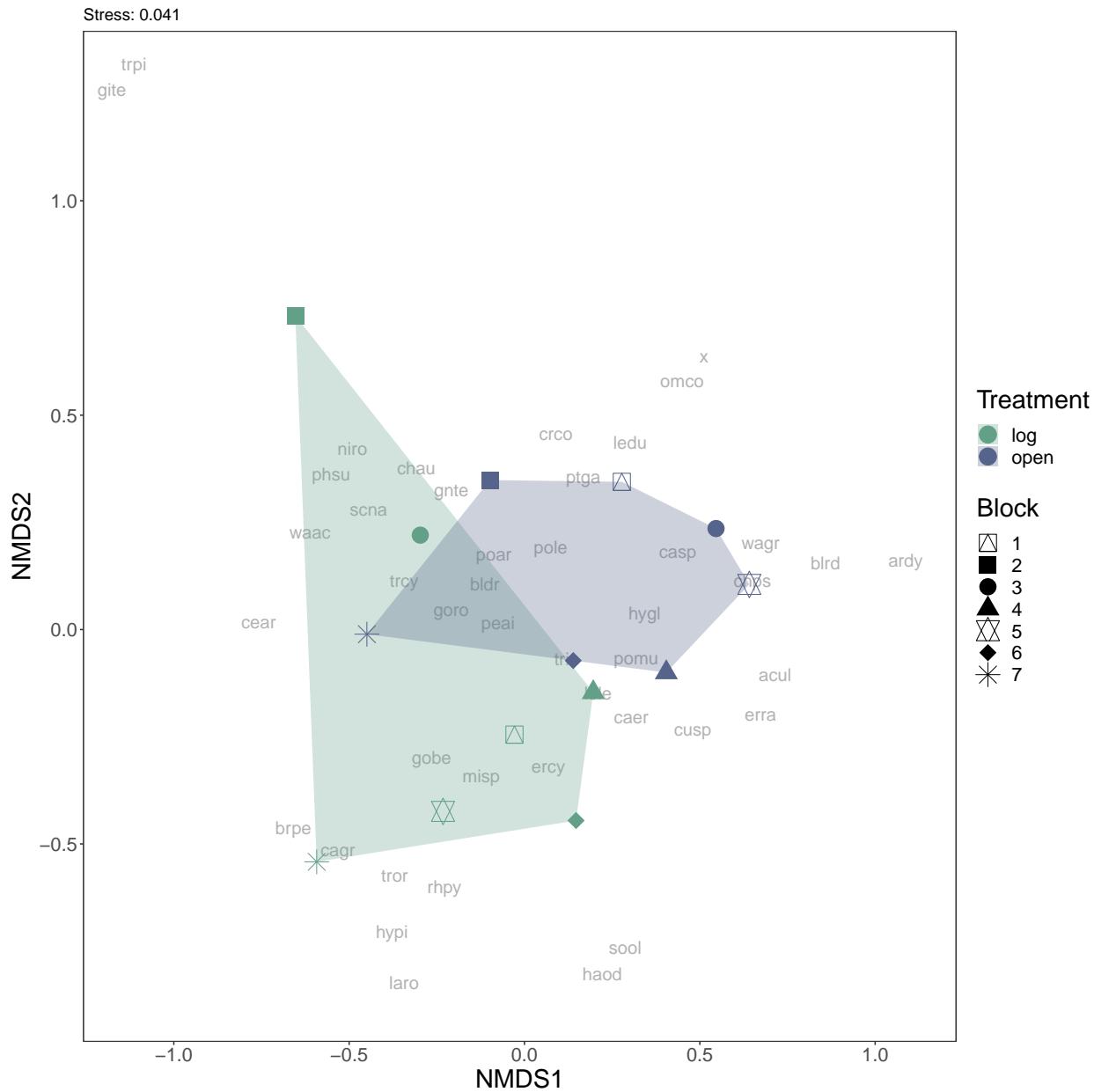
hulldat<-rbind(log,open)

nmds.plot <- ggplot()+
  theme_bw()+
  theme(panel.background = element_blank(),
        panel.grid.major = element_blank(), #remove major-grid labels
        panel.grid.minor = element_blank(), #remove minor-grid labels
        plot.background = element_blank(),
        axis.text = element_text(size = 15),
        axis.title=element_text(size=20),
        legend.title=element_text(size=20),
        legend.text=element_text(size=15))+

  geom_text_repel(data=species.scores, aes(NMDS1, NMDS2, label=species), alpha=0.9, size=5, col='darkgreen')
  geom_polygon(data=hulldat, aes(NMDS1, NMDS2, fill=treatment, group=treatment), alpha=0.3)+scale_fill_manual(values=c("#63A088", "#56638A"), name="Treatment") +
  geom_point(data=mds_scores_t1, aes(NMDS1, NMDS2, shape=block, col=treatment), size=6)+ scale_shape_manual(values=c("triangle-down", "triangle-up"))
  scale_colour_manual(values=c("#63A088", "#56638A"), name="Treatment") +
  labs(title=paste0("Stress: ", round(ass.rel.t1_NMS$stress,3)))

print(nmds.plot)

```



```
# All other NMDS pairs
treatments <- unique(mds_scores_t1$treatment)
plots <- list()
num_axes <- 5

for (i in 1:num_axes) {
  for (j in (i+1):num_axes) {
    if (j <= num_axes && i != j) {
      log <- mds_scores_t1[mds_scores_t1$treatment == "log", ]
      open <- mds_scores_t1[mds_scores_t1$treatment == "open", ]

      log_hull <- log[chull(log[[paste0("NMDS", i)]], log[[paste0("NMDS", j)]]), ]
      open_hull <- open[chull(open[[paste0("NMDS", i)]], open[[paste0("NMDS", j)]]), ]
```

```

hulldat <- rbind(log_hull, open_hull)

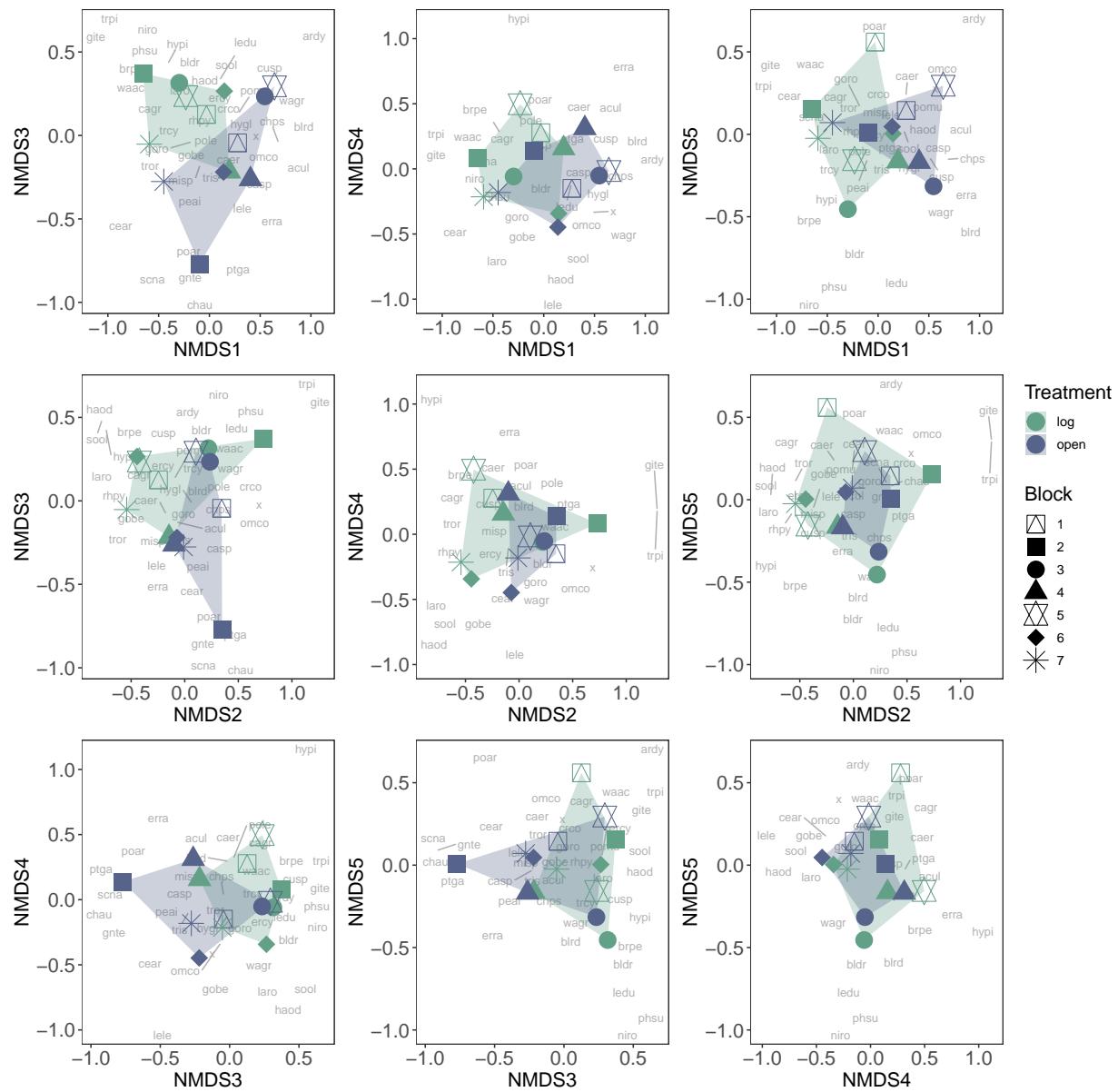
plot_name <- paste0(i, "+", j)

plots[[plot_name]] <- ggplot() +
  theme_bw() +
  theme(panel.background = element_blank(),
        panel.grid.major = element_blank(),
        panel.grid.minor = element_blank(),
        plot.background = element_blank(),
        axis.text = element_text(size = 15),
        axis.title = element_text(size = 15),
        legend.title = element_text(size = 15),
        legend.text = element_text(size = 10)) +
  geom_text_repel(data = species.scores, aes(x = !!sym(paste0("NMDS", i)), y = !!sym(paste0("NMDS",
  geom_polygon(data = hulldat, aes(x = !!sym(paste0("NMDS", i)), y = !!sym(paste0("NMDS", j))), fill = "#63A088"),
  geom_point(data = mds_scores_t1, aes(x = !!sym(paste0("NMDS", i)), y = !!sym(paste0("NMDS", j))),
  scale_colour_manual(values = c("#63A088", "#56638A"), name = "Treatment") +
  xlab(paste0("NMDS", i)) +
  ylab(paste0("NMDS", j))
)
}
}
}

((plots$`1+3` + plots$`1+4` + plots$`1+5`)/(plots$`2+3` + plots$`2+4` + plots$`2+5`)/(plots$`3+4` + plots$`3+5`)

```

Stress:0.041



Composition dissimilarity * 2022 *

```
# subset data where all to communities, insitu log and insitu open communities at t1 and t2 are included
mat2 <- mat[which(mat$time=="t2" & mat$treatment=="open" | mat$time=="t2" & mat$treatment=="insitu_log")]
mat2$grp<-apply(mat2[c(89,91)], 1, paste, collapse=":") # block, init as grouping
names(mat2) #check
```

```
## [1] "acul"      "aicu"      "arca"      "ardy"      "arsp"      "ael" 
## [7] "bldr"      "blrd"      "brdi"      "brdr"      "brpe"      "brru" 
## [13] "buse"      "caer"      "cagr"      "cahi"      "casp"      "cear" 
## [19] "chau"      "chei"      "chps"      "crcl"      "crc"      "cusc" 
## [25] "cusp"      "dagl"      "dosp"      "ento"      "erau"      "ercy" 
## [31] "erra"      "ersp"      "gite"      "gnte"      "gobe"      "gocy" 
## [37] "gono"      "goro"      "gosp"      "haod"      "hygl"      "hyp" 
```

```

## [43] "hypo"      "jubu"       "laro"       "ledu"       "lele"       "loef"
## [49] "misp"        "mite"       "momo"       "mopa"       "niro"       "omco"
## [55] "orsp"        "pala"       "peai"       "pedu"       "phsu"       "plde"
## [61] "poar"        "poca"       "pocap"      "poce"       "pogn"       "pole"
## [67] "pomu"        "pter"       "ptga"       "ptob"       "rhla"       "rhpyp"
## [73] "rhsp"        "ry"         "scna"       "sino"       "sool"       "stfi"
## [79] "stpi"        "thma"       "trcy"       "tris"       "tror"       "trpi"
## [85] "waac"        "wagr"       "x"          "time"      "block"      "transect"
## [91] "init"        "treatment" "grp"

# another df where the grouping variables are time, block and initial state
# each row is a transect in a certain year.
df<-mat2[,c(1:87, 93)]
df2 = df %>% mutate(across(.cols=1:87, .fns=as.numeric)) # make everything numeric
rownames(df2)<-NULL # remove rownames

## new with group vars
nublock<-separate(df2, 88, c("block", "init"), ":") # just looking at time, block & initial treatment

# want to sum across transects in same block X init treatment
nublock$sumgrp<-apply(mat2[c(89, 91)], 1, paste, collapse=":")
head(nublock)

```

```

##   acul aicu arca ardy arsp auel bldr blrd brdi brdr brpe brru buse caer cagr
## 1   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
## 2   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
## 3   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
## 4   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
## 5   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
## 6   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
##   cahi casp clear chau chei chps crcl crco cusc cusp dagl dosp ento erau ercy
## 1   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
## 2   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
## 3   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   3
## 4   0   0   0   0   0   0   0   0   1   0   0   0   0   0   0   0   0   0
## 5   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   1
## 6   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
##   erra ersp gite gnte gobe gocy gono goro gosp haod hygl hypi hypo jubu laro
## 1   0   0   0   0   1   0   0   0   1   0   0   0   0   0   0   0   0   0
## 2   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   1   0
## 3   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   2   0
## 4   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
## 5   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
## 6   0   0   0   0   0   0   0   0   3   0   0   0   1   0   0   0   0   0
##   ledu lele loef misp mite momo mopa niro omco orsp pala peai pedu phsu plde
## 1   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
## 2   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
## 3   0   0   0   0   3   0   0   0   0   0   0   0   0   0   0   0   0   0
## 4   0   0   0   0   1   0   0   0   0   0   0   0   0   0   0   14  0   0
## 5   0   0   0   0   1   0   0   0   0   0   0   0   0   0   0   0   0   0
## 6   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   3   0   0
##   poar poca pocap poce pogn pole pomu pter ptga ptob rhla rhpyp rhsp ry scna
## 1   0   1   0   0   0   1   0   0   0   0   0   0   0   0   0   0   4   0   0
## 2   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0

```

```

## 3 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
## 4 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0
## 5 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
## 6 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0
##   sino sool stfi stpi thma trcy tris tror trpi waac wagr x block init sumgrp
## 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 log 1:log
## 2 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 1 log 1:log
## 3 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 1 open 1:open
## 4 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 open 1:open
## 5 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 1 log 1:log
## 6 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 open 1:open

# sum observations across initial X block (group variable)
# this gives number of plants in each transect TYPE for each year in each block. should be 2 types X 3
blocksum<-rowsum(nublock[,c(1:87)], group=nublock$sumgrp)
blocksum$grps<-rownames(blocksum)
rownames(blocksum)<-NULL # remove rownames
nrow(blocksum) # it is 14 rows as expected

## [1] 14

## expand again
blocksum<-separate(blocksum, 88, c("block", "init"), ":")

# at the moment this includes where there were no plants ("x" column in matrix)
assemblies_t2<-blocksum[,c(1:87)]

# group - these are the treatment variables that need to be separately fed into the MDS analaysis from
group_init<-blocksum$init
group_block<-blocksum$block

# MDS
ass.rel.t2<-decostand(assemblies_t2, method='hel') #standardize assemblies
ass.rel.t2_NMS <- metaMDS(ass.rel.t2, distance = 'bray', k = 4) # run MDS

## Run 0 stress 0.03974779
## Run 1 stress 0.0397477
## ... New best solution
## ... Procrustes: rmse 7.13827e-05 max resid 0.000130094
## ... Similar to previous best
## Run 2 stress 0.03974788
## ... Procrustes: rmse 0.000130337 max resid 0.0002377108
## ... Similar to previous best
## Run 3 stress 0.03974771
## ... Procrustes: rmse 1.145027e-05 max resid 2.688896e-05
## ... Similar to previous best
## Run 4 stress 0.03974787
## ... Procrustes: rmse 0.0005866705 max resid 0.001107309
## ... Similar to previous best
## Run 5 stress 0.03974782
## ... Procrustes: rmse 0.0005155979 max resid 0.001006802
## ... Similar to previous best
## Run 6 stress 0.03974777

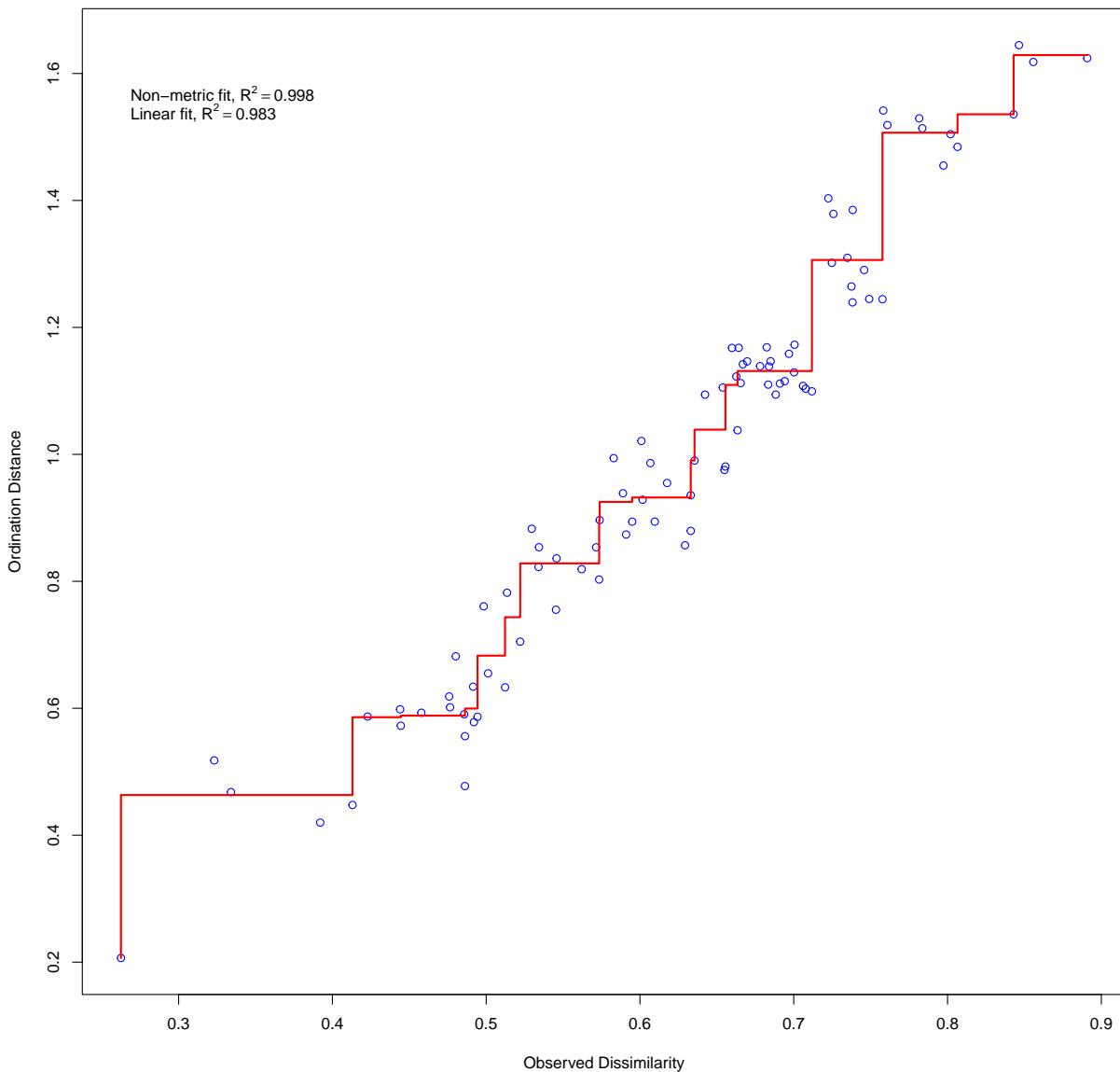
```

```

## ... Procrustes: rmse 5.522057e-05 max resid 0.0001080054
## ... Similar to previous best
## Run 7 stress 0.03974767
## ... New best solution
## ... Procrustes: rmse 3.076272e-05 max resid 5.966055e-05
## ... Similar to previous best
## Run 8 stress 0.03974786
## ... Procrustes: rmse 0.0005537094 max resid 0.001026692
## ... Similar to previous best
## Run 9 stress 0.03974765
## ... New best solution
## ... Procrustes: rmse 3.341022e-05 max resid 6.152772e-05
## ... Similar to previous best
## Run 10 stress 0.03974779
## ... Procrustes: rmse 0.0001217201 max resid 0.000213477
## ... Similar to previous best
## Run 11 stress 0.03974764
## ... New best solution
## ... Procrustes: rmse 5.502505e-05 max resid 0.000100464
## ... Similar to previous best
## Run 12 stress 0.05534903
## Run 13 stress 0.03974759
## ... New best solution
## ... Procrustes: rmse 0.0002430737 max resid 0.0004318672
## ... Similar to previous best
## Run 14 stress 0.0397476
## ... Procrustes: rmse 9.660628e-06 max resid 2.16419e-05
## ... Similar to previous best
## Run 15 stress 0.03974765
## ... Procrustes: rmse 0.0002648231 max resid 0.0005020814
## ... Similar to previous best
## Run 16 stress 0.03974758
## ... New best solution
## ... Procrustes: rmse 0.000133951 max resid 0.0002544132
## ... Similar to previous best
## Run 17 stress 0.03974762
## ... Procrustes: rmse 0.0001727935 max resid 0.0003316766
## ... Similar to previous best
## Run 18 stress 0.03974763
## ... Procrustes: rmse 0.000115191 max resid 0.0002126153
## ... Similar to previous best
## Run 19 stress 0.03974763
## ... Procrustes: rmse 8.531594e-05 max resid 0.000155956
## ... Similar to previous best
## Run 20 stress 0.03974776
## ... Procrustes: rmse 0.000269636 max resid 0.0005165638
## ... Similar to previous best
## *** Best solution repeated 5 times

```

```
stressplot(ass.rel.t2_NMS) # check fit
```



```

# scores
mds_scores_t2<-as.data.frame(vegan::scores(ass.rel.t2_NMS)$sites) # extract scores
mds_scores_t2$site<-rownames(vegan::scores(ass.rel.t2_NMS)$sites) # extract names
mds_scores_t2$treatment<-group_init # grouping factor 1
mds_scores_t2$block<-group_block # grouping factor 2

# explaining factors
init<-as.factor(group_init) # grouping factor 1- convert to factor
block<-as.factor(group_block) # grouping factor 2- convert to factor

##### rda model analysis & results #####
# can look at significance of model where initial treatment and block explain variation in community
trt_tot_2<-rda(ass.rel.t2~init+block) # run model using standardized data
summary(trt_tot_2)

```

```

##
## Call:
## rda(formula = ass.rel.t2 ~ init + block)
##
## Partitioning of variance:
##           Inertia Proportion
## Total      0.5340    1.0000
## Constrained 0.3581    0.6705
## Unconstrained 0.1759    0.3295
##
## Eigenvalues, and their contribution to the variance
##
## Importance of components:
##           RDA1     RDA2     RDA3     RDA4     RDA5     RDA6     RDA7
## Eigenvalue      0.1078  0.07881  0.05379  0.04086  0.03384  0.02612  0.01683
## Proportion Explained 0.2019  0.14758  0.10073  0.07651  0.06338  0.04891  0.03151
## Cumulative Proportion 0.2019  0.34951  0.45024  0.52675  0.59013  0.63904  0.67055
##           PC1     PC2     PC3     PC4     PC5     PC6
## Eigenvalue      0.0547  0.03572  0.03141  0.02309  0.01636  0.01464
## Proportion Explained 0.1024  0.06689  0.05882  0.04325  0.03064  0.02742
## Cumulative Proportion 0.7730  0.83988  0.89870  0.94194  0.97258  1.00000
##
## Accumulated constrained eigenvalues
## Importance of components:
##           RDA1     RDA2     RDA3     RDA4     RDA5     RDA6     RDA7
## Eigenvalue      0.1078  0.07881  0.05379  0.04086  0.03384  0.02612  0.01683
## Proportion Explained 0.3011  0.22009  0.15022  0.11410  0.09452  0.07293  0.04699
## Cumulative Proportion 0.3011  0.52124  0.67146  0.78556  0.88007  0.95301  1.00000
##
## Scaling 2 for species and site scores
## * Species are scaled proportional to eigenvalues
## * Sites are unscaled: weighted dispersion equal on all dimensions
## * General scaling constant of scores: 1.623197
##
##
## Species scores
##
##           RDA1     RDA2     RDA3     RDA4     RDA5     RDA6
## acul   9.052e-02 -6.166e-02 -2.465e-02  1.043e-01  1.350e-01  9.105e-02
## aicu   4.768e-17  4.052e-17 -9.248e-17 -1.056e-17 -1.453e-17 -3.522e-18
## arca  -5.475e-17 -2.441e-18 -4.889e-17  1.376e-17 -1.781e-17  5.141e-17
## ardy  -4.584e-02 -4.434e-02 -1.111e-02  1.969e-02 -2.105e-02 -1.788e-02
## arsp  -1.390e-17  6.109e-19 -1.670e-17  5.467e-17  3.549e-17  3.880e-17
## auel  -1.762e-17  3.587e-17  1.566e-17 -1.261e-17 -4.081e-17 -4.764e-17
## bldr   1.039e-01  4.551e-02 -1.034e-02  5.410e-02  3.353e-02 -5.643e-02
## blrd   1.876e-17 -2.147e-17  1.438e-18  1.943e-17  2.607e-17  1.106e-17
## brdi   1.852e-02  3.590e-02  8.983e-03  2.044e-02  1.514e-03 -4.940e-02
## brdr   6.717e-18 -2.173e-18  1.097e-18  1.249e-18  4.221e-18 -1.502e-17
## brpe   1.354e-17  2.728e-17  4.933e-18  1.483e-17  1.682e-19 -3.402e-17
## brru  -7.711e-33  1.305e-32  2.283e-33  3.454e-33 -6.606e-33  1.260e-32
## buse   2.953e-33 -1.069e-32  2.015e-33 -2.537e-33  5.339e-33 -8.503e-33
## caer   1.272e-01 -6.922e-02  1.166e-01  7.445e-02  2.762e-02  4.229e-03

```

```

## cagr  0.000e+00  0.000e+00  0.000e+00  0.000e+00  0.000e+00  0.000e+00
## cahi  2.197e-02 -8.344e-02  7.037e-02  5.156e-02 -1.853e-02 -2.675e-02
## casp  0.000e+00  0.000e+00  0.000e+00  0.000e+00  0.000e+00  0.000e+00
## cear  0.000e+00  0.000e+00  0.000e+00  0.000e+00  0.000e+00  0.000e+00
## chau  0.000e+00  0.000e+00  0.000e+00  0.000e+00  0.000e+00  0.000e+00
## chei  -6.027e-02  4.108e-02 -1.022e-01  5.789e-02 -9.473e-02  3.468e-02
## chps  1.422e-01  5.056e-02 -9.443e-02 -2.245e-02  2.895e-02 -2.191e-02
## crcl  7.114e-02 -4.012e-02  4.716e-02 -3.862e-02 -7.741e-02  8.715e-03
## crco  9.734e-02 -6.815e-02 -1.138e-01  5.228e-02  9.551e-02 -5.560e-03
## cusc  9.947e-02 -4.263e-02  1.155e-01 -3.845e-02 -1.031e-01 -3.646e-02
## cusp  0.000e+00  0.000e+00  0.000e+00  0.000e+00  0.000e+00  0.000e+00
## dagl  0.000e+00  0.000e+00  0.000e+00  0.000e+00  0.000e+00  0.000e+00
## dosp  -1.315e-01 -1.124e-01 -1.082e-01  5.196e-02 -4.110e-02 -7.973e-02
## ento  0.000e+00  0.000e+00  0.000e+00  0.000e+00  0.000e+00  0.000e+00
## erau  3.620e-02 -3.512e-02  4.765e-02 -2.634e-02 -4.680e-02  5.787e-03
## ercy  4.416e-02 -5.421e-03 -7.098e-02 -1.592e-01  9.063e-02  1.565e-02
## erra  2.116e-02  3.187e-02 -5.398e-02 -1.259e-02  2.466e-02 -1.379e-02
## ersp  0.000e+00  0.000e+00  0.000e+00  0.000e+00  0.000e+00  0.000e+00
## gite  -2.639e-02 -2.973e-02 -6.122e-02  2.699e-02 -3.764e-02 -2.222e-02
## gnre  0.000e+00  0.000e+00  0.000e+00  0.000e+00  0.000e+00  0.000e+00
## gobe  -1.559e-01  1.913e-01  1.457e-01 -2.624e-03 -2.945e-02  7.403e-02
## gocy  0.000e+00  0.000e+00  0.000e+00  0.000e+00  0.000e+00  0.000e+00
## gono  0.000e+00  0.000e+00  0.000e+00  0.000e+00  0.000e+00  0.000e+00
## goro  -8.288e-03  1.152e-01 -1.390e-01  5.549e-02 -1.156e-01  3.310e-02
## gosp  0.000e+00  0.000e+00  0.000e+00  0.000e+00  0.000e+00  0.000e+00
## haod  -9.922e-04  1.618e-01 -1.355e-02 -1.014e-02  4.472e-02 -7.729e-02
## hygl  6.118e-02  9.387e-02  4.443e-02 -1.767e-02 -1.196e-02 -1.030e-01
## hypi  0.000e+00  0.000e+00  0.000e+00  0.000e+00  0.000e+00  0.000e+00
## hypo  -1.494e-01 -1.129e-01  6.094e-02 -1.241e-01 -4.803e-02  2.488e-02
## jubu  0.000e+00  0.000e+00  0.000e+00  0.000e+00  0.000e+00  0.000e+00
## laro  -5.349e-02 -2.490e-02  4.872e-02  4.252e-02 -7.231e-03 -7.957e-02
## ledu  0.000e+00  0.000e+00  0.000e+00  0.000e+00  0.000e+00  0.000e+00
## lele  0.000e+00  0.000e+00  0.000e+00  0.000e+00  0.000e+00  0.000e+00
## loef  -2.155e-02 -2.427e-02 -4.998e-02  2.204e-02 -3.073e-02 -1.814e-02
## misp  0.000e+00  0.000e+00  0.000e+00  0.000e+00  0.000e+00  0.000e+00
## mite  -1.261e-01  1.544e-01  1.114e-01 -9.659e-03  1.890e-01  3.377e-02
## momo  0.000e+00  0.000e+00  0.000e+00  0.000e+00  0.000e+00  0.000e+00
## mopa  0.000e+00  0.000e+00  0.000e+00  0.000e+00  0.000e+00  0.000e+00
## niro  0.000e+00  0.000e+00  0.000e+00  0.000e+00  0.000e+00  0.000e+00
## omco  0.000e+00  0.000e+00  0.000e+00  0.000e+00  0.000e+00  0.000e+00
## orsp  -5.409e-03  1.374e-02  4.231e-02  1.614e-02  9.775e-03 -4.362e-02
## pala  0.000e+00  0.000e+00  0.000e+00  0.000e+00  0.000e+00  0.000e+00
## peai  -1.076e-01  3.483e-01 -1.470e-01 -5.028e-02 -1.466e-02 -3.413e-02
## pedu  0.000e+00  0.000e+00  0.000e+00  0.000e+00  0.000e+00  0.000e+00
## phsu  0.000e+00  0.000e+00  0.000e+00  0.000e+00  0.000e+00  0.000e+00
## plde  0.000e+00  0.000e+00  0.000e+00  0.000e+00  0.000e+00  0.000e+00
## poar  0.000e+00  0.000e+00  0.000e+00  0.000e+00  0.000e+00  0.000e+00
## poca  -7.330e-02  1.228e-02 -1.627e-02 -1.762e-01  1.119e-01 -9.665e-02
## pocap -9.137e-02  1.421e-02 -2.497e-02  4.681e-02 -6.872e-02  3.860e-02
## poce  -5.409e-03  1.374e-02  4.231e-02  1.614e-02  9.775e-03 -4.362e-02
## pogn  0.000e+00  0.000e+00  0.000e+00  0.000e+00  0.000e+00  0.000e+00
## pole  5.618e-02 -1.627e-01 -9.388e-02 -1.303e-01  5.130e-02  8.606e-03
## pomu  4.919e-01  5.095e-02  2.181e-03 -4.628e-02 -5.128e-02 -4.981e-02
## pter  8.015e-02  4.723e-02 -2.287e-02 -4.203e-03 -2.753e-02 -5.322e-02

```

```

## ptga  1.310e-02  2.539e-02  6.352e-03  1.445e-02  1.071e-03 -3.493e-02
## ptob  0.000e+00  0.000e+00  0.000e+00  0.000e+00  0.000e+00  0.000e+00
## rhla  2.267e-02 -6.908e-02  4.303e-02  6.541e-02  8.281e-02  6.017e-02
## rhpy  1.001e-03  5.611e-02  1.010e-01  5.055e-02  2.293e-02 -1.325e-01
## rhsp  -2.148e-01 -1.818e-01 -9.260e-02 -9.215e-02 -2.889e-02 -6.196e-02
## ry    0.000e+00  0.000e+00  0.000e+00  0.000e+00  0.000e+00  0.000e+00
## scna  0.000e+00  0.000e+00  0.000e+00  0.000e+00  0.000e+00  0.000e+00
## sino  3.620e-02 -3.512e-02  4.765e-02 -2.634e-02 -4.680e-02  5.787e-03
## sool  0.000e+00  0.000e+00  0.000e+00  0.000e+00  0.000e+00  0.000e+00
## stfi  -5.409e-03  1.374e-02  4.231e-02  1.614e-02  9.775e-03 -4.362e-02
## stpi  0.000e+00  0.000e+00  0.000e+00  0.000e+00  0.000e+00  0.000e+00
## thma  0.000e+00  0.000e+00  0.000e+00  0.000e+00  0.000e+00  0.000e+00
## trcy  -1.561e-01 -1.095e-01  2.729e-02  1.591e-01  4.816e-02 -1.420e-01
## tris  4.228e-02 -1.673e-02 -2.024e-02  4.110e-02  3.757e-02  3.527e-02
## tror  -5.230e-02  8.443e-02  1.375e-01  8.368e-03 -4.656e-02  3.086e-02
## trpi  0.000e+00  0.000e+00  0.000e+00  0.000e+00  0.000e+00  0.000e+00
## waac -1.822e-02 -3.727e-02  1.688e-01 -1.391e-01 -3.077e-02 -1.163e-02
## wagr  0.000e+00  0.000e+00  0.000e+00  0.000e+00  0.000e+00  0.000e+00
## x     0.000e+00  0.000e+00  0.000e+00  0.000e+00  0.000e+00  0.000e+00
##
##
## Site scores (weighted sums of species scores)
##
##          RDA1      RDA2      RDA3      RDA4      RDA5      RDA6
## row1  -0.529954 -0.50054  0.2468 -1.10353  0.14120  0.29241
## row2   0.027932  0.29494 -0.3213 -0.55623  0.49222 -0.18686
## row3  -0.526506 -0.90064 -0.2668  0.49320 -0.41943 -0.57329
## row4  -0.521497 -0.16611 -0.6799  0.15480 -0.38435  0.01384
## row5   0.523959 -0.45922  0.7665 -0.33398 -1.07139  0.06484
## row6   0.751980 -0.09337 -0.1892 -0.28283 -0.23276  0.07611
## row7  -0.131864  0.32707 -0.1624 -0.14669  0.60202 -0.12836
## row8   0.300694  0.18578 -0.6404 -0.21465  0.18647 -0.22795
## row9  -0.083385 -0.56645  0.3037  0.69285  1.18053  0.55202
## row10  0.816213 -0.18975 -0.2365  0.35601 -0.04331  0.37932
## row11  0.004328  0.36280  0.7414  0.20897  0.15651 -0.81764
## row12  0.187756  0.43299  0.1371  0.38949  0.03776 -0.70925
## row13 -0.590590  0.63290  0.4722  0.06967 -0.07356  0.61728
## row14 -0.229066  0.63960 -0.1712  0.27293 -0.57192  0.64753
##
##
## Site constraints (linear combinations of constraining variables)
##
##          RDA1      RDA2      RDA3      RDA4      RDA5      RDA6
## row1  -0.44158 -0.26053  0.26294 -0.8470  0.39041  0.05381
## row2  -0.06044  0.05493 -0.33743 -0.8128  0.24301  0.05174
## row3  -0.71457 -0.69110 -0.17315  0.3069 -0.32819 -0.27869
## row4  -0.33343 -0.37564 -0.77353  0.3411 -0.47559 -0.28076
## row5   0.44740 -0.43402  0.58885 -0.3255 -0.57837  0.07151
## row6   0.82854 -0.11857 -0.01153 -0.2913 -0.72577  0.06944
## row7  -0.10616  0.09870 -0.10123 -0.1977  0.46794 -0.17712
## row8   0.27499  0.41415 -0.70160 -0.1636  0.32055 -0.17919
## row9   0.17584 -0.53583  0.33379  0.5074  0.64231  0.46671
## row10  0.55699 -0.22037 -0.26659  0.5415  0.49491  0.46464
## row11 -0.09453  0.24017  0.73940  0.2822  0.17083 -0.76241

```

```

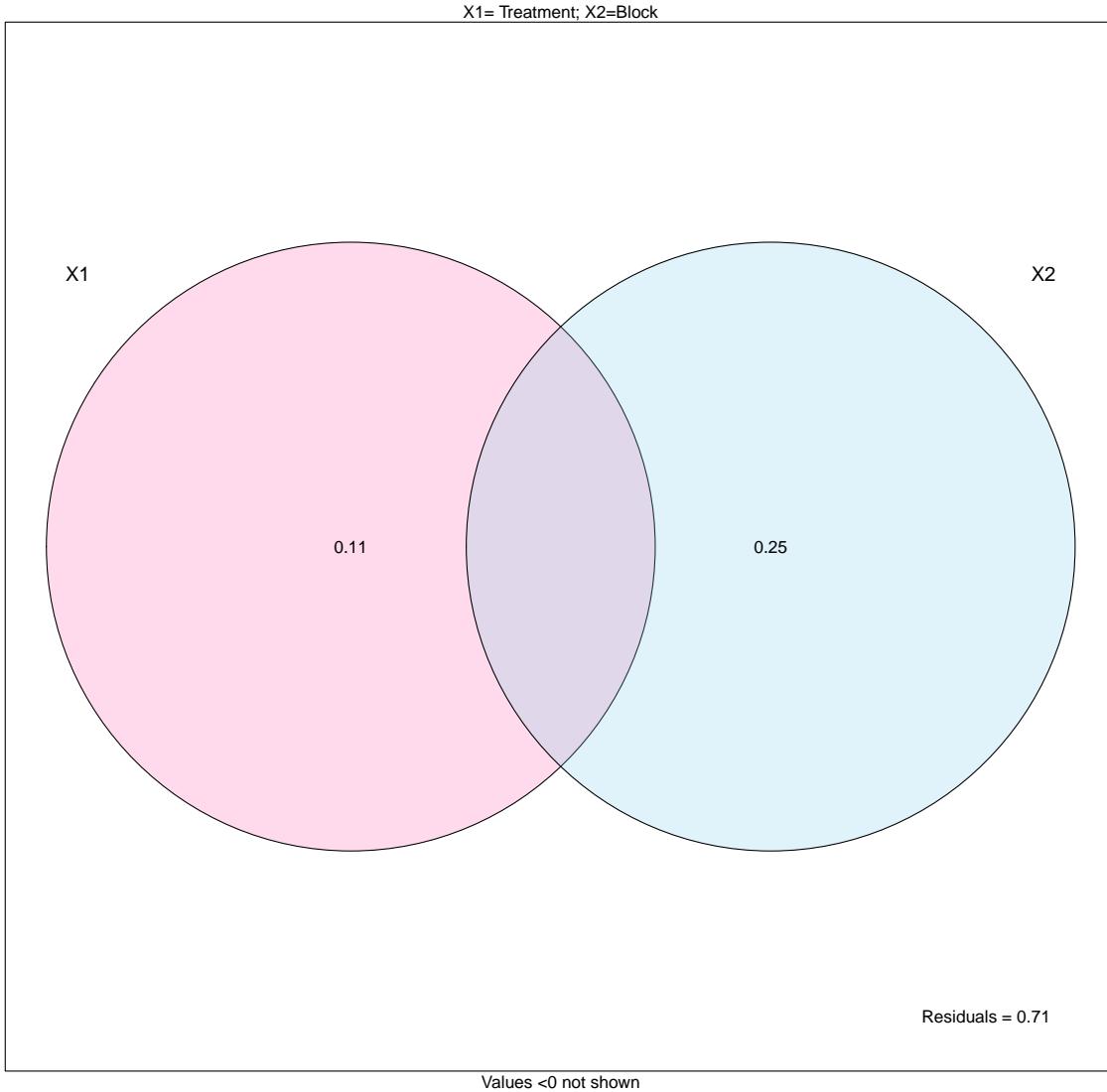
## row12  0.28661  0.55562  0.13902  0.3163  0.02343 -0.76448
## row13 -0.60040  0.47852  0.45072  0.1542 -0.24904  0.63344
## row14 -0.21925  0.79398 -0.14966  0.1884 -0.39644  0.63137
##
##
## Biplot scores for constraining variables
##
##          RDA1     RDA2     RDA3     RDA4     RDA5     RDA6
## initopen  0.43929  0.3636 -0.69197  0.03936 -0.16988 -0.002386
## block2   -0.49312 -0.5019 -0.44544  0.30490 -0.37820 -0.263238
## block3    0.60037 -0.2600  0.27165 -0.29023 -0.61364  0.066324
## block4    0.07944  0.2413 -0.37775 -0.17002  0.37101 -0.167656
## block5    0.34482 -0.3558  0.03162  0.49352  0.53510  0.438225
## block6    0.09038  0.3744  0.41332  0.28159  0.09141 -0.718450
## block7   -0.38567  0.5987  0.14166  0.16120 -0.30372  0.595132
##
##
## Centroids for factor constraints
##
##          RDA1     RDA2     RDA3     RDA4     RDA5     RDA6
## initlog  -0.19057 -0.1577  0.30019 -0.01707  0.07370  0.001035
## initopen  0.19057  0.1577 -0.30019  0.01707 -0.07370 -0.001035
## block1   -0.25101 -0.1028 -0.03724 -0.82988  0.31671  0.052774
## block2   -0.52400 -0.5334 -0.47334  0.32400 -0.40189 -0.279725
## block3    0.63797 -0.2763  0.28866 -0.30841 -0.65207  0.070478
## block4    0.08441  0.2564 -0.40141 -0.18067  0.39425 -0.178156
## block5    0.36641 -0.3781  0.03360  0.52443  0.56861  0.465671
## block6    0.09604  0.3979  0.43921  0.29923  0.09713 -0.763447
## block7   -0.40983  0.6363  0.15053  0.17130 -0.32274  0.632406

anova.cca(trt_tot_2, step=1000, by="term") ## test for model significance

## Permutation test for rda under reduced model
## Terms added sequentially (first to last)
## Permutation: free
## Number of permutations: 999
##
## Model: rda(formula = ass.rel.t2 ~ init + block)
##          Df Variance      F Pr(>F)
## init       1  0.06081 2.0739  0.009 **
## block      6  0.29726 1.6897  0.001 ***
## Residual   6  0.17593
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

# can model using varpart to look at contributions of initial treatment and block
var.mod2<-varpart(ass.rel.t2, init, block) # run model on standardized data
plot(var.mod2, bg=c("hotpink","skyblue"))
mtext("X1= Treatment; X2=Block", side=3)

```



```
## can test for significance of contribution of the fraction of initial treatment
# do this with partial redundancy analysis
trt_Frac<-rda(ass.rel.t2, init, block) # partial rda model
summary(trt_Frac)
```

```
##
## Call:
## rda(X = ass.rel.t2, Y = init, Z = block)
##
## Partitioning of variance:
##                 Inertia Proportion
## Total          0.53400   1.0000
## Conditioned   0.29726   0.5567
```

```

## Constrained 0.06081      0.1139
## Unconstrained 0.17593     0.3295
##
## Eigenvalues, and their contribution to the variance
## after removing the contribution of conditiniong variables
##
## Importance of components:
##          RDA1      PC1      PC2      PC3      PC4      PC5      PC6
## Eigenvalue      0.06081 0.0547 0.03572 0.03141 0.02309 0.01636 0.01464
## Proportion Explained 0.25687 0.2311 0.15089 0.13267 0.09755 0.06910 0.06185
## Cumulative Proportion 0.25687 0.4879 0.63882 0.77149 0.86905 0.93815 1.00000
##
## Accumulated constrained eigenvalues
## Importance of components:
##          RDA1
## Eigenvalue      0.06081
## Proportion Explained 1.00000
## Cumulative Proportion 1.00000
##
## Scaling 2 for species and site scores
## * Species are scaled proportional to eigenvalues
## * Sites are unscaled: weighted dispersion equal on all dimensions
## * General scaling constant of scores: 1.623197
##
##
## Species scores
##
##          RDA1      PC1      PC2      PC3      PC4      PC5
## acul    0.020175 -2.142e-02 -8.157e-03 -2.565e-02 -6.560e-03 -2.267e-02
## aicu    0.000000 -4.876e-17 -4.519e-17 -3.653e-17  6.908e-18 -1.928e-17
## arca    0.000000 -8.942e-18  4.164e-17  3.659e-17  9.181e-18 -7.967e-18
## ardy    -0.027831 -4.756e-02 -3.944e-02  8.675e-03 -1.186e-02  5.108e-03
## arsp    0.000000 -2.014e-17 -2.380e-19 -1.365e-17  1.358e-17  1.535e-17
## auel    0.000000 -4.771e-18 -4.237e-19 -5.032e-18 -3.204e-18 -1.865e-17
## bldr    0.077576 -1.896e-02  1.384e-02  1.648e-02  6.044e-03  1.461e-01
## blrd    0.000000 -8.146e-18  9.663e-18  1.578e-17 -1.302e-17  4.303e-18
## brdi    0.028032  2.184e-03 -2.188e-02 -3.219e-02  1.751e-02  5.284e-02
## brdr    0.000000  5.211e-20 -9.460e-19 -2.257e-18  2.614e-18  9.147e-19
## brpe    0.000000 -7.947e-19  4.558e-18 -2.427e-19  1.126e-17 -2.201e-17
## brru    0.000000  6.061e-35 -3.019e-34  1.792e-34 -1.096e-33  1.716e-33
## buse    0.000000  1.148e-34 -5.719e-34  3.395e-34 -2.077e-33  3.251e-33
## caer    -0.044190  2.898e-02 -2.379e-02 -2.399e-02  2.885e-02 -2.967e-02
## cagr    0.000000  0.000e+00  0.000e+00  0.000e+00  0.000e+00  0.000e+00
## cahi    -0.067194 -7.302e-03 -1.008e-01 -7.009e-02  7.686e-02  2.540e-02
## casp    0.000000  0.000e+00  0.000e+00  0.000e+00  0.000e+00  0.000e+00
## cear    0.000000  0.000e+00  0.000e+00  0.000e+00  0.000e+00  0.000e+00
## chau    0.000000  0.000e+00  0.000e+00  0.000e+00  0.000e+00  0.000e+00
## chei    0.093970  8.115e-02  7.788e-02 -2.052e-02  5.365e-02 -6.052e-02
## chps    0.156906 -6.527e-02 -4.068e-02  4.561e-04 -4.431e-02  2.951e-02
## crcl    -0.016809 -6.644e-03  7.075e-04 -5.708e-03  3.885e-02 -5.185e-03
## crco    0.079849  2.340e-02 -9.206e-02  2.134e-02 -4.315e-02  4.745e-02
## cusc    -0.050462 -2.884e-02 -1.858e-02 -5.884e-02  1.989e-01  2.863e-02
## cusp    0.000000  0.000e+00  0.000e+00  0.000e+00  0.000e+00  0.000e+00
## dagl    0.000000  0.000e+00  0.000e+00  0.000e+00  0.000e+00  0.000e+00

```

```

## dosp -0.053319 -2.094e-01 -1.574e-01 -2.948e-02 -5.637e-02 -3.437e-02
## ento 0.000000 0.000e+00 0.000e+00 0.000e+00 0.000e+00 0.000e+00
## erau -0.035104 -1.388e-02 1.477e-03 -1.192e-02 8.114e-02 -1.083e-02
## ercy 0.046162 -4.595e-02 4.546e-02 -3.595e-03 -6.486e-02 1.494e-02
## erra 0.033378 -3.459e-02 -1.612e-02 -4.611e-02 -1.184e-02 -4.033e-02
## ersp 0.000000 0.000e+00 0.000e+00 0.000e+00 0.000e+00 0.000e+00
## gite 0.034332 5.867e-02 4.865e-02 -1.070e-02 1.463e-02 -6.301e-03
## gnte 0.000000 0.000e+00 0.000e+00 0.000e+00 0.000e+00 0.000e+00
## gobe -0.070327 -2.086e-02 5.792e-02 -7.753e-02 -4.545e-02 7.220e-02
## gocy 0.000000 0.000e+00 0.000e+00 0.000e+00 0.000e+00 0.000e+00
## gono 0.000000 0.000e+00 0.000e+00 0.000e+00 0.000e+00 0.000e+00
## goro 0.192777 -2.313e-02 4.363e-02 2.883e-02 6.048e-02 -2.237e-02
## gosp 0.000000 0.000e+00 0.000e+00 0.000e+00 0.000e+00 0.000e+00
## haod 0.056933 -4.086e-02 -8.614e-02 -4.801e-02 -2.278e-02 -3.305e-03
## hygl 0.032852 -4.006e-02 -3.934e-02 -3.105e-02 9.185e-02 -7.827e-02
## hypi 0.000000 0.000e+00 0.000e+00 0.000e+00 0.000e+00 0.000e+00
## hypo -0.136055 -1.149e-01 -6.122e-02 -1.281e-02 2.658e-02 3.954e-02
## jubu 0.000000 0.000e+00 0.000e+00 0.000e+00 0.000e+00 0.000e+00
## laro -0.062935 -5.030e-02 -1.204e-02 4.898e-02 -3.378e-02 -6.106e-02
## ledu 0.000000 0.000e+00 0.000e+00 0.000e+00 0.000e+00 0.000e+00
## lele 0.000000 0.000e+00 0.000e+00 0.000e+00 0.000e+00 0.000e+00
## loef 0.028032 4.791e-02 3.972e-02 -8.738e-03 1.194e-02 -5.144e-03
## misp 0.000000 0.000e+00 0.000e+00 0.000e+00 0.000e+00 0.000e+00
## mite -0.113449 2.772e-01 -6.553e-02 2.380e-02 -5.406e-02 -9.228e-03
## momo 0.000000 0.000e+00 0.000e+00 0.000e+00 0.000e+00 0.000e+00
## mopa 0.000000 0.000e+00 0.000e+00 0.000e+00 0.000e+00 0.000e+00
## niro 0.000000 0.000e+00 0.000e+00 0.000e+00 0.000e+00 0.000e+00
## omco 0.000000 0.000e+00 0.000e+00 0.000e+00 0.000e+00 0.000e+00
## orsp -0.024822 -1.934e-03 1.938e-02 2.850e-02 -1.550e-02 -4.679e-02
## pala 0.000000 0.000e+00 0.000e+00 0.000e+00 0.000e+00 0.000e+00
## peai 0.142682 1.789e-01 -2.923e-02 1.335e-01 2.529e-02 2.227e-02
## pedu 0.000000 0.000e+00 0.000e+00 0.000e+00 0.000e+00 0.000e+00
## phsu 0.000000 0.000e+00 0.000e+00 0.000e+00 0.000e+00 0.000e+00
## plde 0.000000 0.000e+00 0.000e+00 0.000e+00 0.000e+00 0.000e+00
## poar 0.000000 0.000e+00 0.000e+00 0.000e+00 0.000e+00 0.000e+00
## poca -0.014943 8.686e-02 3.030e-02 8.356e-02 2.960e-02 3.563e-02
## pocap -0.004362 6.952e-02 4.738e-02 -7.135e-03 -1.505e-02 4.273e-02
## poce -0.024822 -1.934e-03 1.938e-02 2.850e-02 -1.550e-02 -4.679e-02
## pogn 0.000000 0.000e+00 0.000e+00 0.000e+00 0.000e+00 0.000e+00
## pole -0.026285 1.014e-01 -6.193e-02 -1.139e-01 -5.895e-02 -6.553e-02
## pomu 0.247365 -1.682e-01 6.404e-02 2.052e-01 -3.013e-02 -6.595e-03
## pter 0.073175 -7.558e-03 -3.227e-02 -5.003e-02 -4.913e-02 3.754e-02
## ptga 0.019822 1.544e-03 -1.547e-02 -2.276e-02 1.238e-02 3.736e-02
## ptob 0.000000 0.000e+00 0.000e+00 0.000e+00 0.000e+00 0.000e+00
## rhla -0.055929 8.999e-02 -7.097e-02 -6.003e-02 -1.719e-02 -3.763e-02
## rhpy -0.035682 -2.780e-03 2.785e-02 4.097e-02 -2.229e-02 -6.726e-02
## rhsp -0.056525 1.354e-02 1.937e-01 -1.522e-01 -3.834e-02 2.451e-02
## ry 0.000000 0.000e+00 0.000e+00 0.000e+00 0.000e+00 0.000e+00
## scna 0.000000 0.000e+00 0.000e+00 0.000e+00 0.000e+00 0.000e+00
## sino -0.035104 -1.388e-02 1.477e-03 -1.192e-02 8.114e-02 -1.083e-02
## sool 0.000000 0.000e+00 0.000e+00 0.000e+00 0.000e+00 0.000e+00
## stfi -0.024822 -1.934e-03 1.938e-02 2.850e-02 -1.550e-02 -4.679e-02
## stpi 0.000000 0.000e+00 0.000e+00 0.000e+00 0.000e+00 0.000e+00
## thma 0.000000 0.000e+00 0.000e+00 0.000e+00 0.000e+00 0.000e+00

```

```

## trcy -0.141493 8.767e-02 -1.282e-01 4.651e-02 -2.340e-02 3.479e-02
## tris 0.032930 -5.298e-02 4.179e-02 3.534e-02 1.012e-02 2.215e-02
## tror -0.121334 1.672e-02 4.009e-02 7.411e-02 5.603e-02 -1.665e-02
## trpi 0.000000 0.000e+00 0.000e+00 0.000e+00 0.000e+00 0.000e+00
## waac -0.130919 -3.799e-02 1.255e-01 -6.015e-02 -5.416e-02 -2.217e-02
## wagr 0.000000 0.000e+00 0.000e+00 0.000e+00 0.000e+00 0.000e+00
## x    0.000000 0.000e+00 0.000e+00 0.000e+00 0.000e+00 0.000e+00
##
##
## Site scores (weighted sums of species scores)
##
##          RDA1      PC1      PC2      PC3      PC4      PC5
## sit1   -0.5054 -0.21824  0.68560 -0.67128 -0.2526  0.14674
## sit2    0.5054  0.21824 -0.68560  0.67128  0.2526 -0.14674
## sit3   -0.3427 -0.74139 -0.61475  0.13523 -0.1848  0.07961
## sit4    0.3427  0.74139  0.61475 -0.13523  0.1848 -0.07961
## sit5   -0.4597 -0.17148  0.01826 -0.14731  1.0027 -0.13382
## sit6    0.4597  0.17148 -0.01826  0.14731 -1.0027  0.13382
## sit7   -0.2952  0.44952  0.20946  0.59934  0.1539  0.52410
## sit8    0.2952 -0.44952 -0.20946 -0.59934 -0.1539 -0.52410
## sit9   -0.7338  0.69798 -0.55047 -0.46560 -0.1334 -0.29185
## sit10   0.7338 -0.69798  0.55047  0.46560  0.1334  0.29185
## sit11   -0.3003 -0.03380  0.33862  0.49814 -0.2710 -0.81776
## sit12   0.3003  0.03380 -0.33862 -0.49814  0.2710  0.81776
## sit13   -0.3996  0.01741 -0.08672  0.05148 -0.3149  0.49297
## sit14   0.3996 -0.01741  0.08672 -0.05148  0.3149 -0.49297
##
##
## Site constraints (linear combinations of constraining variables)
##
##          RDA1      PC1      PC2      PC3      PC4      PC5
## con1   -0.4338 -0.21824  0.68560 -0.67128 -0.2526  0.14674
## con2    0.4338  0.21824 -0.68560  0.67128  0.2526 -0.14674
## con3   -0.4338 -0.74139 -0.61475  0.13523 -0.1848  0.07961
## con4    0.4338  0.74139  0.61475 -0.13523  0.1848 -0.07961
## con5   -0.4338 -0.17148  0.01826 -0.14731  1.0027 -0.13382
## con6    0.4338  0.17148 -0.01826  0.14731 -1.0027  0.13382
## con7   -0.4338  0.44952  0.20946  0.59934  0.1539  0.52410
## con8    0.4338 -0.44952 -0.20946 -0.59934 -0.1539 -0.52410
## con9   -0.4338  0.69798 -0.55047 -0.46560 -0.1334 -0.29185
## con10   0.4338 -0.69798  0.55047  0.46560  0.1334  0.29185
## con11   -0.4338 -0.03380  0.33862  0.49814 -0.2710 -0.81776
## con12   0.4338  0.03380 -0.33862 -0.49814  0.2710  0.81776
## con13   -0.4338  0.01741 -0.08672  0.05148 -0.3149  0.49297
## con14   0.4338 -0.01741  0.08672 -0.05148  0.3149 -0.49297
##
##
## Biplot scores for constraining variables
##
##          RDA1 PC1 PC2 PC3 PC4 PC5
## Yopen     1   0   0   0   0   0

```

`RsquareAdj(trt_Frac)$adj.r.squared #explanatory power`

```

## [1] 0.1095098

anova.cca(trt_Frac) ## this tells us if first condition, init, significantly contributes to overall var

## Permutation test for rda under reduced model
## Permutation: free
## Number of permutations: 999
##
## Model: rda(X = ass.rel.t2, Y = init, Z = block)
##          Df Variance      F Pr(>F)
## Model     1  0.06081 2.0739   0.07 .
## Residual  6  0.17593
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

### extracting species scores and plotting
# species scores
species.scores.t2<-as.data.frame(vegan::scores(ass.rel.t2_NMS,"species")) ## some species don't have scores
species.scores.t2$species<-rownames(species.scores.t2)

### NMDS 1 and 2
log<-mds_scores_t2[mds_scores_t2$treatment == "log", ][chull(mds_scores_t2[mds_scores_t2$treatment ==
"log", c("NMDS1", "NMDS2")])], ]

open<-mds_scores_t2[mds_scores_t2$treatment == "open", ][chull(mds_scores_t2[mds_scores_t2$treatment ==
"open", c("NMDS1", "NMDS2")])], ]

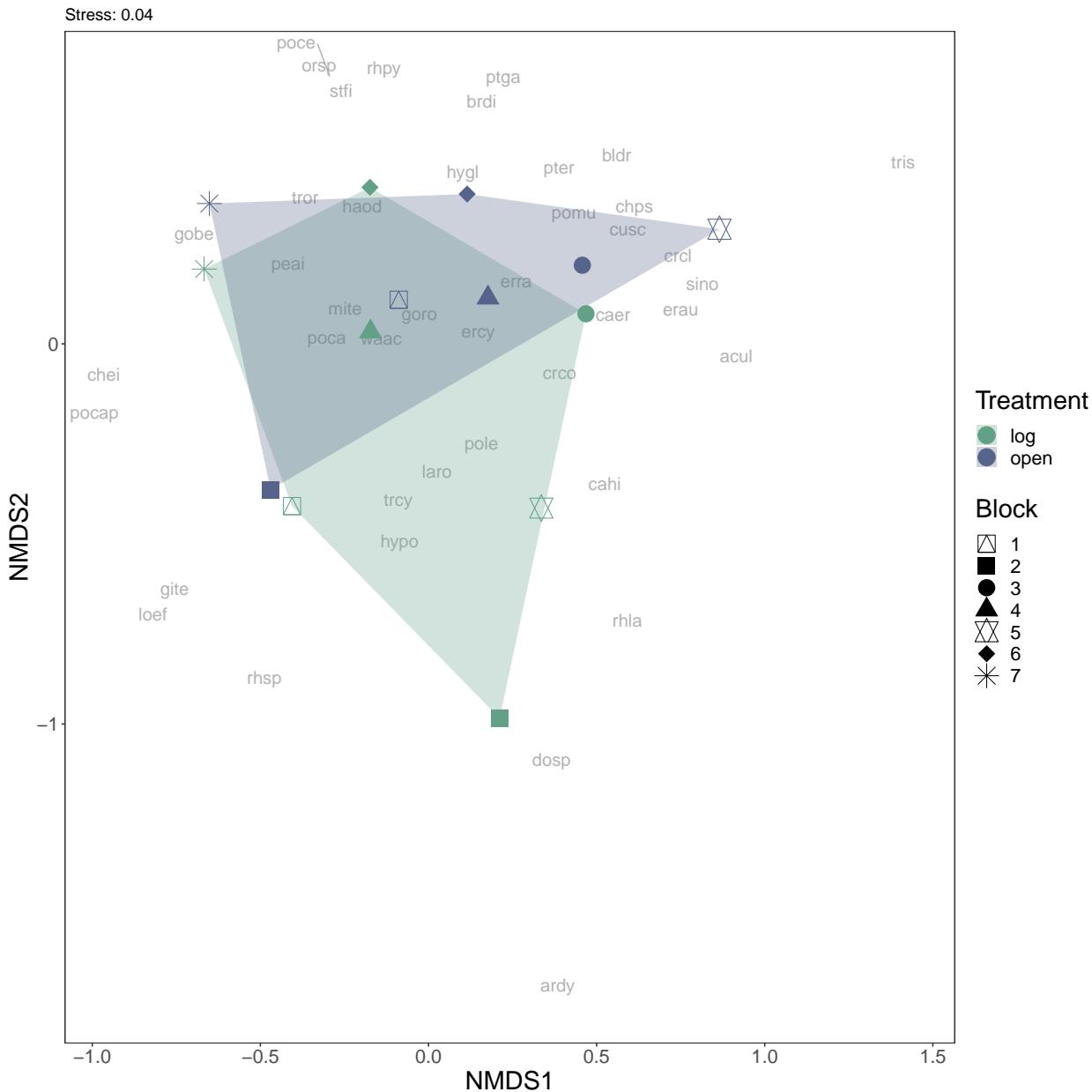
hulldat<-rbind(log,open)

nmds.plot <- ggplot()+
  theme_bw()+
  theme(panel.background = element_blank(),
        panel.grid.major = element_blank(), #remove major-grid labels
        panel.grid.minor = element_blank(), #remove minor-grid labels
        plot.background = element_blank(),
        axis.text = element_text(size = 15),
        axis.title=element_text(size=20),
        legend.title=element_text(size=20),
        legend.text=element_text(size=15))+

  geom_text_repel(data=species.scores.t2, aes(NMDS1, NMDS2, label=species), alpha=0.9, size=5, col='darkblue')+
  geom_polygon(data=hulldat, aes(NMDS1, NMDS2, fill=treatment, group=treatment), alpha=0.3)+scale_fill_manual(values=c("#63A088", "#56638A"), name="Treatment") +
  geom_point(data=mds_scores_t2, aes(NMDS1, NMDS2, shape=block, col=treatment), size=6)+ scale_shape_manual(values=c("triangle-down", "triangle-up", "square", "diamond", "circle", "cross", "triangle-left", "triangle-right", "square", "diamond", "circle", "cross"))+
  scale_colour_manual(values=c("#63A088", "#56638A"), name="Treatment") +
  labs(title=paste0("Stress: ", round(ass.rel.t2_NMS$stress,3))) 

print(nmds.plot)

```



```
# All other NMDS pairs
treatments <- unique(mds_scores_t2$treatment)
plots <- list()
num_axes <- 4

for (i in 1:num_axes) {
  for (j in (i+1):num_axes) {
    if (j <= num_axes && i != j) {
      log <- mds_scores_t2[mds_scores_t2$treatment == "log", ]
      open <- mds_scores_t2[mds_scores_t2$treatment == "open", ]

      log_hull <- log[chull(log[[paste0("NMDS", i)]], log[[paste0("NMDS", j)]]), ]
      open_hull <- open[chull(open[[paste0("NMDS", i)]], open[[paste0("NMDS", j)]]), ]
```

```

hulldat <- rbind(log_hull, open_hull)

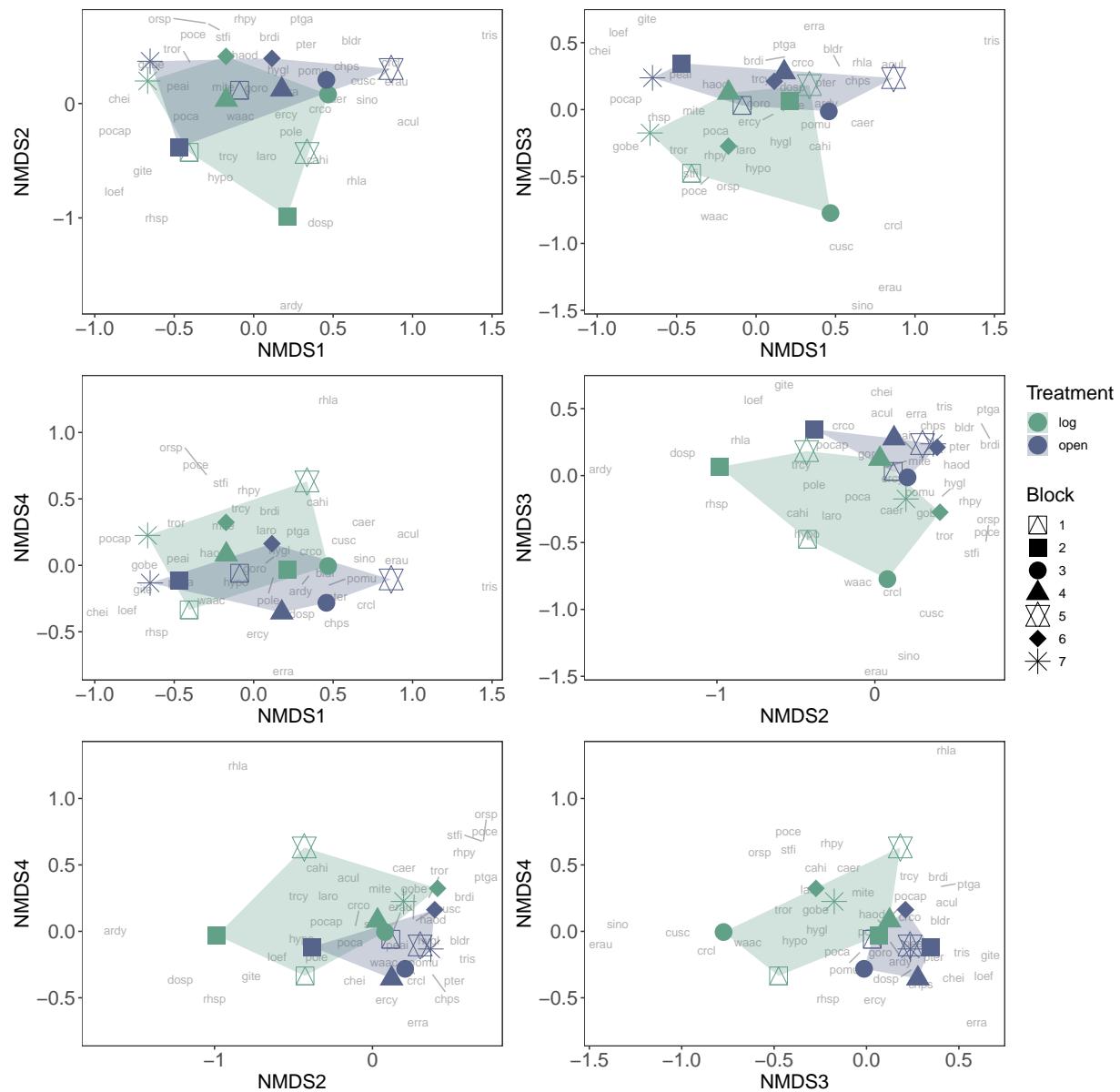
plot_name <- paste0(i, "+", j)

plots[[plot_name]] <- ggplot() +
  theme_bw() +
  theme(panel.background = element_blank(),
        panel.grid.major = element_blank(),
        panel.grid.minor = element_blank(),
        plot.background = element_blank(),
        axis.text = element_text(size = 15),
        axis.title = element_text(size = 15),
        legend.title = element_text(size = 15),
        legend.text = element_text(size = 10)) +
  geom_text_repel(data = species.scores.t2, aes(x = !!sym(paste0("NMDS", i)), y = !!sym(paste0("NMDS", j)),
                                                 label = species), alpha = 0.9, size = 3, col = 'darkgreen')
  geom_polygon(data = hulldat, aes(x = !!sym(paste0("NMDS", i)), y = !!sym(paste0("NMDS", j))), fill = "#63A088")
  scale_fill_manual(values = c("#63A088", "#56638A"), name = "Treatment") +
  geom_point(data = mds_scores_t2, aes(x = !!sym(paste0("NMDS", i)), y = !!sym(paste0("NMDS", j)),
                                         scale_shape_manual(values = c(14, 15, 16, 17, 11, 18, 8), name = 'Block') +
                                         scale_colour_manual(values = c("#63A088", "#56638A"), name = "Treatment") +
                                         xlab(paste0("NMDS", i)) +
                                         ylab(paste0("NMDS", j)))
}
}
}

((plots$`1+2` + plots$`1+3`)/(plots$`1+4` + plots$`2+3`)/(plots$`2+4` + plots$`3+4`)) + plot_layout(guides = "none")

```

Stress:0.04



Composition dissimilarity * 2020 - 2022 *

```
# subset data where all to communities, insitu log and insitu open communities at t1 and t2 are included
mat3 <- mat[which(mat$time=="t0" | mat$treatment=="open" | mat$treatment=="insitu_log") ,]
mat3$grp<-apply(mat3[c(88,89,91)], 1, paste, collapse=":") # block, init as grouping
names(mat3) #check
```

```
## [1] "acul"      "aicu"      "arca"      "ardy"      "arsp"      "auel"
## [7] "bldr"      "blrd"      "brdi"      "brdr"      "brpe"      "brru"
## [13] "buse"      "caer"      "cagr"      "cahi"      "casp"      "cear"
## [19] "chau"      "chei"      "chsps"     "crcl"      "crc"      "cusc"
## [25] "cusp"      "dagl"      "dosp"      "ento"      "erau"      "ercy"
## [31] "erra"      "ersp"      "gite"      "gnte"      "gobe"      "gocy"
## [37] "gono"      "goro"      "gosp"      "haod"      "hygl"      "hypi"
```

```

## [43] "hypo"      "jubu"       "laro"       "ledu"       "lele"       "loef"
## [49] "misp"       "mite"       "momo"       "mopa"       "niro"       "omco"
## [55] "orsp"       "pala"       "peai"       "pedu"       "phsu"       "plde"
## [61] "poar"       "poca"       "pocap"      "poce"       "pogn"       "pole"
## [67] "pomu"       "pter"       "ptga"       "ptob"       "rhla"       "rhpyp"
## [73] "rhsp"       "ry"         "scna"       "sino"       "sool"       "stfi"
## [79] "stpi"       "thma"       "trcy"       "tris"       "tror"       "trpi"
## [85] "waac"       "wagr"       "x"          "time"      "block"      "transect"
## [91] "init"        "treatment" "grp"

# another df where the grouping variables are time, block and initial state
# each row is a transect in a certain year.
df<-mat3[,c(1:87, 93)]
df2 = df %>% mutate(across(.cols=1:87,.fns=as.numeric)) # make everything numeric
rownames(df2)<-NULL # remove rownames

## new with group vars
nublock<-separate(df2, 88, c("time", "block", "init"), ":") # just looking at time, block & initial tre

# want to sum across transects in same block X init treatment
nublock$sumgrp<-apply(mat3[c(88,89, 91)], 1, paste, collapse=":")
head(nublock)

## acul aicu arca ardy arsp auel bldr blrd brdi brdr brpe brru buse caer cagr
## 1   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
## 2   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   1   0
## 3   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   1   0
## 4   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
## 5   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
## 6   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
## cahi casp clear chau chei chps crcl crco cusc cusp dagl dosp ento erau ercy
## 1   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
## 2   0   0   0   0   0   0   0   0   1   0   0   0   0   0   0   0   0   0
## 3   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
## 4   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
## 5   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
## 6   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
## erra ersp gite gnte gobe gocy gono goro gosp haod hygl hypi hypo jubu laro
## 1   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
## 2   0   0   0   0   0   0   0   0   0   0   0   0   0   4   0   0   0   0
## 3   0   0   0   0   0   0   0   0   0   0   0   0   0   1   0   0   0   0
## 4   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
## 5   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   1
## 6   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   1
## ledu lele loef misp mite momo mopa niro omco orsp pala peai pedu phsu plde
## 1   0   0   0   1   0   0   0   0   0   0   0   0   0   0   0   0   0   0
## 2   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
## 3   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
## 4   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
## 5   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
## 6   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
## poar poca pocap poce pogn pole pomu pter ptga ptob rhla rhpyp rhsp ry scna
## 1   0   0   0   0   0   1   0   0   0   0   0   0   0   0   0   0   0   0
## 2   0   1   0   0   0   0   1   2   0   0   0   0   0   0   0   0   0   0

```

```

## 3   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
## 4   0   0   0   0   0   3   2   0   0   0   0   0   0   0   0   0   0
## 5   0   0   0   0   0   0   0   0   1   0   0   0   0   0   0   0   0
## 6   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
##   sino sool stfi stpi thma trcy tris tror trpi waac wagr x time block init
## 1   0   0   0   0   0   0   0   0   0   2   0   0   t0   1   log
## 2   0   0   0   0   0   0   0   0   0   0   0   0   t0   1   log
## 3   0   0   0   0   0   0   0   0   0   0   0   0   t0   1   log
## 4   0   0   0   0   0   0   0   0   0   0   0   0   t0   1   log
## 5   0   0   0   0   0   1   0   0   0   0   1   0   t0   1   log
## 6   0   0   0   0   0   0   0   1   0   2   0   0   t0   1   log
##   sumgrp
## 1 t0:1:log
## 2 t0:1:log
## 3 t0:1:log
## 4 t0:1:log
## 5 t0:1:log
## 6 t0:1:log

# sum observations across initial X block (group variable)
# this gives number of plants in each transect TYPE for each year in each block. should be 2 types X 3 :
blocksum<-rowsum(nublock[,c(1:87)], group=nublock$sumgrp)
blocksum$grps<-rownames(blocksum)
rownames(blocksum)<-NULL # remove rownames
nrow(blocksum) # it is 42 rows as expected

## [1] 42

## expand again
blocksum<-separate(blocksum, 88, c("time", "block", "init"), ":")

# at the moment this includes where there were no plants ("x" column in matrix)
assemblies_t012<-blocksum[,c(1:87)]

# group - these are the treatment variables that need to be separately fed into the MDS analaysis from
group_init<-blocksum$init
group_block<-blocksum$block
group_time<-blocksum$time

# MDS
ass.rel.t012<-decostand(assemblies_t012, method='hel') #standardize assemblies
ass.rel.t012_NMS <- metaMDS(ass.rel.t012, distance = 'bray', k = 5) # run MDS

## Run 0 stress 0.09244926
## Run 1 stress 0.09244993
## ... Procrustes: rmse 0.000381609 max resid 0.001525378
## ... Similar to previous best
## Run 2 stress 0.09244946
## ... Procrustes: rmse 0.000152916 max resid 0.0005665753
## ... Similar to previous best
## Run 3 stress 0.09244991
## ... Procrustes: rmse 0.0003675035 max resid 0.0009098031
## ... Similar to previous best

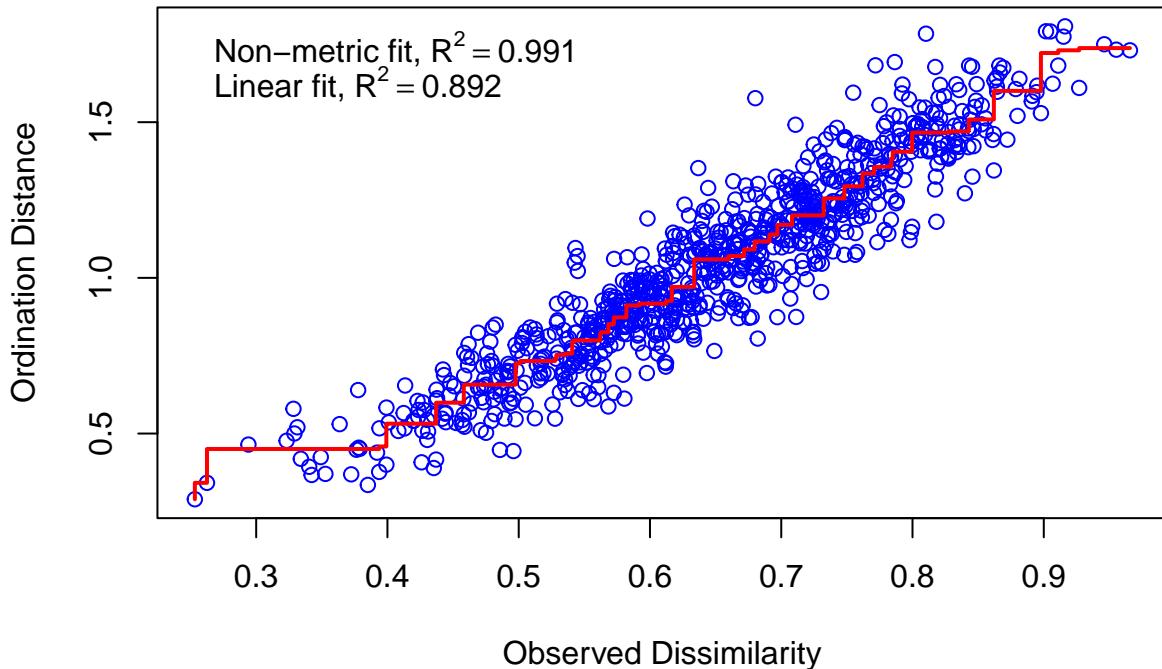
```

```

## Run 4 stress 0.09245811
## ... Procrustes: rmse 0.001948835 max resid 0.007919375
## ... Similar to previous best
## Run 5 stress 0.09245025
## ... Procrustes: rmse 0.001065356 max resid 0.004274963
## ... Similar to previous best
## Run 6 stress 0.09244976
## ... Procrustes: rmse 0.0002936155 max resid 0.001171642
## ... Similar to previous best
## Run 7 stress 0.0926169
## ... Procrustes: rmse 0.00565147 max resid 0.02268149
## Run 8 stress 0.09525969
## Run 9 stress 0.09403585
## Run 10 stress 0.0924651
## ... Procrustes: rmse 0.00314468 max resid 0.01300129
## Run 11 stress 0.09244954
## ... Procrustes: rmse 0.0007655137 max resid 0.003021789
## ... Similar to previous best
## Run 12 stress 0.0924493
## ... Procrustes: rmse 0.0001144719 max resid 0.0003033332
## ... Similar to previous best
## Run 13 stress 0.09244978
## ... Procrustes: rmse 0.0004480928 max resid 0.001512182
## ... Similar to previous best
## Run 14 stress 0.09244939
## ... Procrustes: rmse 0.0001078879 max resid 0.0004184752
## ... Similar to previous best
## Run 15 stress 0.1000227
## Run 16 stress 0.09244982
## ... Procrustes: rmse 0.000903348 max resid 0.003577659
## ... Similar to previous best
## Run 17 stress 0.09517904
## Run 18 stress 0.09245132
## ... Procrustes: rmse 0.0008077709 max resid 0.003286014
## ... Similar to previous best
## Run 19 stress 0.09244964
## ... Procrustes: rmse 0.0002498756 max resid 0.001000342
## ... Similar to previous best
## Run 20 stress 0.09933966
## *** Best solution repeated 13 times

```

```
stressplot(ass.rel.t012_NMS) # check fit
```



```

# scores
mds_scores_t012<-as.data.frame(vegan::scores(ass.rel.t012_NMS)$sites) # extract scores
mds_scores_t012$site<-rownames(vegan::scores(ass.rel.t012_NMS)$sites) # extract names
mds_scores_t012$treatment<-group_init # grouping factor 1
mds_scores_t012$block<-group_block # grouping factor 2
mds_scores_t012$time<-group_time # grouping factor 3

# explaining factors
init<-as.factor(group_init) # grouping factor 1- convert to factor
block<-as.factor(group_block) # grouping factor 2- convert to factor
time<-as.factor(group_time)

##### rda model analysis & results #####
# can look at significance of model where initial treatment and block explain variation in community
trt_tot_2<-rda(ass.rel.t012~init+block+time) # run model using standardized data
summary(trt_tot_2)

##
## Call:
## rda(formula = ass.rel.t012 ~ init + block + time)
##
## Partitioning of variance:
##              Inertia Proportion
## Total          0.5618    1.0000
## Constrained    0.2832    0.5041
## Unconstrained  0.2786    0.4959

```

```

## Eigenvalues, and their contribution to the variance
##
## Importance of components:
##          RDA1     RDA2     RDA3     RDA4     RDA5     RDA6     RDA7
## Eigenvalue      0.08458  0.05964  0.04201  0.02374  0.02255  0.01906  0.01374
## Proportion Explained  0.15056  0.10617  0.07479  0.04226  0.04015  0.03393  0.02446
## Cumulative Proportion 0.15056  0.25673  0.33152  0.37377  0.41392  0.44785  0.47231
##          RDA8     RDA9     PC1      PC2      PC3      PC4      PC5
## Eigenvalue      0.01092  0.006918 0.03970  0.02773  0.02383  0.02008  0.01758
## Proportion Explained 0.01944  0.012315 0.07067  0.04937  0.04241  0.03574  0.03129
## Cumulative Proportion 0.49174  0.504060 0.57473  0.62410  0.66651  0.70225  0.73354
##          PC6      PC7      PC8      PC9      PC10     PC11     PC12
## Eigenvalue      0.01589  0.01486  0.01326  0.01138  0.01024  0.01004  0.008831
## Proportion Explained 0.02828  0.02646  0.02360  0.02025  0.01822  0.01787  0.015720
## Cumulative Proportion 0.76182  0.78828 0.81188  0.83213  0.85035  0.86823  0.883948
##          PC13     PC14     PC15     PC16     PC17     PC18
## Eigenvalue      0.008658 0.007353 0.006713 0.005727 0.005255 0.004236
## Proportion Explained 0.015413 0.013089 0.011951 0.010196 0.009354 0.007542
## Cumulative Proportion 0.899361 0.912451 0.924402 0.934597 0.943951 0.951493
##          PC19     PC20     PC21     PC22     PC23     PC24
## Eigenvalue      0.003939 0.003339 0.003025 0.002930 0.002555 0.002392
## Proportion Explained 0.007013 0.005944 0.005384 0.005215 0.004548 0.004258
## Cumulative Proportion 0.958506 0.964449 0.969833 0.975049 0.979597 0.983855
##          PC25     PC26     PC27     PC28     PC29     PC30
## Eigenvalue      0.00227 0.001666 0.001540 0.001124 0.0009868 0.0006454
## Proportion Explained 0.00404 0.002965 0.002741 0.002001 0.0017567 0.0011489
## Cumulative Proportion 0.98789 0.990860 0.993601 0.995602 0.9973584 0.9985073
##          PC31     PC32
## Eigenvalue      0.0005422 0.0002963
## Proportion Explained 0.0009651 0.0005275
## Cumulative Proportion 0.9994725 1.0000000
##
## Accumulated constrained eigenvalues
## Importance of components:
##          RDA1     RDA2     RDA3     RDA4     RDA5     RDA6     RDA7
## Eigenvalue      0.08458  0.05964  0.04201  0.02374  0.02255  0.01906  0.01374
## Proportion Explained 0.29870  0.21062  0.14837  0.08383  0.07965  0.06732  0.04852
## Cumulative Proportion 0.29870  0.50932  0.65769  0.74152  0.82117  0.88849  0.93701
##          RDA8     RDA9
## Eigenvalue      0.01092  0.006918
## Proportion Explained 0.03856  0.024432
## Cumulative Proportion 0.97557 1.0000000
##
## Scaling 2 for species and site scores
## * Species are scaled proportional to eigenvalues
## * Sites are unscaled: weighted dispersion equal on all dimensions
## * General scaling constant of scores: 2.190697
##
##
## Species scores
##
##          RDA1     RDA2     RDA3     RDA4     RDA5     RDA6
## acul -1.352e-01 5.332e-02 -1.259e-02 5.023e-02 -6.287e-02 -6.341e-02

```

```

## aicu -7.804e-04 -5.802e-02 9.699e-03 6.012e-02 5.474e-03 -4.253e-02
## arca -2.071e-17 -2.635e-18 2.541e-17 3.766e-18 2.527e-17 -5.781e-18
## ardy 5.272e-03 1.040e-02 2.181e-02 -1.130e-02 -2.771e-02 2.509e-02
## arsp -9.643e-03 -1.332e-03 -1.163e-02 1.555e-02 8.335e-04 5.776e-03
## auel -8.884e-18 -2.461e-17 6.210e-18 -1.474e-17 -1.563e-17 -3.929e-18
## bldr -1.408e-01 -3.901e-02 1.983e-02 1.389e-01 9.836e-02 6.296e-02
## blrd -2.264e-02 6.417e-03 8.458e-03 -7.255e-03 -2.197e-02 -9.062e-03
## brdi 3.152e-03 1.948e-02 -1.467e-02 1.369e-02 -1.461e-03 1.076e-02
## brdr 5.608e-18 2.575e-18 1.203e-17 -5.589e-18 5.891e-19 2.690e-18
## brpe -1.602e-02 -3.186e-02 3.804e-03 -8.893e-02 -1.504e-02 5.185e-02
## brru 1.333e-17 -1.433e-17 -2.608e-17 3.433e-18 1.107e-18 -3.085e-18
## buse -2.259e-02 -2.523e-02 -2.112e-02 1.968e-02 2.990e-02 -1.224e-02
## caer -3.385e-02 2.762e-02 -2.170e-02 -8.946e-02 -3.206e-03 3.380e-02
## cagr -2.505e-02 -5.415e-02 -2.573e-02 -1.043e-01 8.204e-03 -3.671e-02
## cahi 1.410e-02 3.252e-02 1.110e-02 2.376e-02 4.086e-02 6.657e-02
## casp -3.535e-02 -2.387e-02 -3.264e-02 6.054e-03 1.350e-03 -5.146e-03
## clear 7.452e-03 -8.262e-03 -2.765e-02 -8.841e-03 -2.988e-02 -5.798e-04
## chau 4.033e-02 -8.487e-02 2.595e-02 6.152e-02 -1.217e-01 1.286e-02
## chei 6.797e-02 3.015e-02 1.177e-03 3.980e-02 -3.057e-02 -1.319e-03
## chps -2.114e-01 1.346e-01 1.206e-02 3.194e-02 -8.714e-02 -4.737e-02
## crcl 5.675e-03 2.313e-02 2.127e-02 2.532e-03 2.900e-02 1.526e-02
## crco -9.751e-02 4.977e-02 2.415e-03 9.312e-02 7.276e-04 -2.347e-02
## cusc 1.839e-02 4.290e-02 1.769e-02 5.705e-03 5.243e-02 3.533e-02
## cusp -6.425e-02 8.335e-03 5.404e-03 -6.236e-02 -5.843e-02 5.023e-02
## dagl -1.434e-02 -4.253e-03 1.104e-03 2.858e-03 1.531e-02 -2.459e-03
## dosp 9.908e-02 2.029e-02 6.549e-02 2.487e-02 -1.511e-02 -2.168e-02
## ento 7.849e-34 -4.227e-33 -1.995e-33 -1.417e-33 -1.322e-33 -2.419e-33
## erau 6.815e-03 1.048e-02 1.447e-02 -3.571e-03 2.410e-02 1.099e-02
## ercy 7.238e-03 8.826e-02 1.439e-02 -9.057e-02 4.129e-02 -4.748e-02
## erra -2.996e-02 1.079e-02 -2.628e-03 1.992e-02 -2.935e-02 -9.026e-02
## ersp -5.503e-03 -9.310e-03 -4.865e-04 1.811e-03 3.647e-03 -1.648e-02
## gite 9.198e-02 -1.160e-01 1.446e-01 8.713e-02 -5.955e-02 1.559e-02
## gnte 9.128e-03 -1.595e-02 6.640e-03 1.396e-03 -8.172e-02 1.637e-02
## gobe 7.151e-02 -8.798e-02 -2.389e-01 4.876e-02 7.212e-02 -3.509e-02
## gocy -5.707e-03 -6.090e-02 -7.598e-02 4.988e-02 2.000e-02 9.361e-03
## gono -3.904e-34 6.158e-34 1.814e-33 -1.112e-33 -6.721e-34 -6.411e-35
## goro 2.038e-01 1.974e-01 -3.493e-02 8.297e-02 -4.879e-02 9.732e-02
## gosp 7.403e-03 -4.710e-02 -4.175e-02 5.262e-02 -1.022e-03 -4.045e-03
## haod 3.898e-02 5.696e-02 -1.143e-01 4.754e-02 1.887e-02 2.985e-03
## hygl -3.834e-01 -1.102e-01 -1.387e-01 1.659e-01 4.343e-02 -4.491e-02
## hypi -9.794e-03 -2.952e-03 3.739e-03 -2.814e-02 -8.109e-03 2.043e-02
## hypo 1.169e-01 5.102e-02 4.109e-02 -2.585e-02 7.716e-02 -1.129e-02
## jubu 0.000e+00 0.000e+00 0.000e+00 0.000e+00 0.000e+00 0.000e+00
## laro 5.560e-02 -1.039e-01 -7.036e-02 3.383e-03 6.148e-02 3.348e-02
## ledu -2.646e-02 -1.599e-03 2.294e-02 -1.829e-02 -1.181e-03 1.968e-02
## lele -1.784e-02 1.768e-03 -2.167e-02 1.617e-02 -9.759e-03 1.495e-02
## loef 2.090e-02 8.347e-03 1.803e-02 1.640e-02 -1.202e-02 1.667e-03
## misp -7.312e-02 -1.752e-01 -1.442e-01 -1.544e-01 -9.022e-02 -1.264e-01
## mite 2.426e-01 2.502e-01 -8.879e-02 -1.518e-02 9.716e-02 -2.975e-02
## momo 0.000e+00 0.000e+00 0.000e+00 0.000e+00 0.000e+00 0.000e+00
## mopa 0.000e+00 0.000e+00 0.000e+00 0.000e+00 0.000e+00 0.000e+00
## niro -1.221e-02 -8.832e-03 1.580e-02 -2.057e-02 7.519e-03 1.444e-02
## omco -3.492e-02 2.675e-03 2.675e-03 -7.803e-04 2.781e-03 1.185e-02
## orsp 9.156e-03 7.486e-03 -1.198e-02 1.317e-03 9.096e-03 1.150e-02

```

```

## pala -9.643e-03 -1.332e-03 -1.163e-02  1.555e-02  8.335e-04  5.776e-03
## peai  1.996e-01  1.408e-01 -9.839e-02  6.276e-02 -1.174e-01 -1.522e-01
## pedu -1.364e-02 -1.883e-03 -1.644e-02  2.199e-02  1.179e-03  8.169e-03
## phsu -8.636e-03 -6.245e-03  1.117e-02 -1.455e-02  5.317e-03  1.021e-02
## plde  0.000e+00  0.000e+00  0.000e+00  0.000e+00  0.000e+00  0.000e+00
## poar  5.214e-03 -1.209e-02  2.999e-02 -2.594e-02 -2.562e-02 -3.414e-03
## poca  4.192e-02  8.124e-02  2.421e-02  2.581e-02  8.822e-02 -8.890e-02
## pocap 8.697e-02 -3.209e-02 -5.302e-02  4.155e-02 -4.701e-03 -9.728e-03
## poce  9.156e-03  7.486e-03 -1.198e-02  1.317e-03  9.096e-03  1.150e-02
## pogn  0.000e+00  0.000e+00  0.000e+00  0.000e+00  0.000e+00  0.000e+00
## pole -1.463e-01 -4.319e-02  3.372e-01 -2.635e-02  5.774e-02 -1.385e-01
## pomu -4.440e-01  2.539e-01 -6.204e-02 -9.485e-02  5.478e-02  6.174e-02
## pter  2.655e-03  5.056e-02 -5.263e-03  2.698e-02 -8.523e-04  1.103e-03
## ptga -5.643e-02 -6.544e-02 -1.842e-02  5.964e-02 -3.666e-02 -5.975e-02
## ptob  0.000e+00  0.000e+00  0.000e+00  0.000e+00  0.000e+00  0.000e+00
## rhla  1.186e-02 -7.171e-02  7.555e-02  8.678e-02  3.446e-02  5.557e-03
## rhpy -4.529e-02 -6.666e-02 -1.228e-01 -3.905e-02 -6.508e-03  9.903e-02
## rhsp  1.334e-01  5.970e-02  8.546e-02  3.620e-03  3.938e-02 -2.220e-02
## ry   0.000e+00  0.000e+00  0.000e+00  0.000e+00  0.000e+00  0.000e+00
## scna  8.647e-03 -9.521e-03  9.665e-04 -1.552e-03 -3.535e-02  2.814e-03
## sino  6.815e-03  1.048e-02  1.447e-02 -3.571e-03  2.410e-02  1.099e-02
## sool -5.069e-03 -7.645e-03 -1.504e-02 -1.305e-02 -3.740e-03  1.751e-02
## stfi  1.665e-02 -9.154e-03 -2.588e-02  3.378e-03  1.522e-02  9.245e-03
## stpi  0.000e+00  0.000e+00  0.000e+00  0.000e+00  0.000e+00  0.000e+00
## thma  0.000e+00  0.000e+00  0.000e+00  0.000e+00  0.000e+00  0.000e+00
## trcy  7.531e-02 -3.294e-01  4.383e-02 -1.270e-02 -1.299e-02  9.935e-02
## tris -3.720e-02  6.677e-03 -2.576e-02 -5.285e-02 -1.163e-01 -9.491e-04
## tror  1.109e-01 -1.595e-01 -1.479e-01 -6.950e-02  7.228e-02 -7.979e-02
## trpi  3.280e-02 -6.171e-02  6.544e-02 -2.251e-02 -2.722e-02  2.228e-02
## waac  4.608e-02 -9.984e-02  2.809e-02 -4.060e-02  1.915e-01 -5.280e-02
## wagr -3.673e-02 -3.151e-02  1.045e-04  1.517e-02  4.158e-04  8.068e-03
## x    -2.098e-02 -5.255e-02 -8.786e-03  2.502e-02  2.387e-02 -5.198e-02
##
##
## Site scores (weighted sums of species scores)
##
##          RDA1      RDA2      RDA3      RDA4      RDA5      RDA6
## row1 -0.257338 -0.313063  0.020105 -0.134844  0.64621 -0.366172
## row2 -0.398714  0.123562 -0.048418 -0.128493  0.42943 -0.105014
## row3  0.317456 -0.769095  0.672801  0.268812  0.11621 -0.014977
## row4  0.272592 -0.226513  0.502008  0.637420 -0.16794 -0.090995
## row5 -0.147937 -0.388979  0.366416  0.303362  0.47504  0.281488
## row6 -0.446754 -0.052862  0.395907  0.521843  0.26280 -0.017165
## row7 -0.210697 -0.395476 -0.215785 -0.030810  0.03097 -0.622618
## row8 -0.494843 -0.082658  0.137590  0.415334  0.13999 -0.704501
## row9 -0.175615 -0.155737 -0.189747  0.332952  0.12227  0.590741
## row10 -0.488579  0.388216 -0.267539  0.204636 -0.02584  0.257674
## row11 -0.292306 -0.240244 -0.390268  0.413529  0.02767 -0.055817
## row12 -0.394403 -0.171885 -0.332983  0.848938  0.21852  0.087984
## row13  0.156068 -0.733320 -0.656816 -0.151352  0.39094 -0.222948
## row14  0.135220 -0.608855 -0.459284  0.283059 -0.19786 -0.021358
## row15 -0.178952  0.116695  0.064179 -1.189249  0.25669 -0.207551
## row16 -0.387875  0.188633  0.008533 -0.008487 -0.16194 -0.304608
## row17  0.381340 -0.448629  0.708800  0.057059 -0.19109  0.488855

```

```

## row18  0.204507 -0.234608  0.324452  0.070379 -1.02230 -0.359593
## row19 -0.065217 -0.291501  0.410172 -0.058580  0.21305  0.257654
## row20 -0.558097  0.151048  0.261748  0.126193 -0.09271  0.179499
## row21 -0.244697 -0.093999 -0.031868 -0.641495 -0.52418 -0.552265
## row22 -0.404294 -0.156219 -0.029492 -0.159984 -0.61672 -0.796503
## row23 -0.115820 -0.355621  0.334501 -1.190103 -0.21593  0.250267
## row24 -0.475941  0.457123 -0.028082 -0.115206 -0.19384  0.422101
## row25 -0.241955  0.001675 -0.373074 -0.579241 -0.15920  0.743215
## row26 -0.110199  0.174188 -0.375245 -0.036516 -0.47646  0.479780
## row27  0.332359 -0.388262 -0.657868 -0.519986 -0.04383  0.074004
## row28  0.169196 -0.284788 -0.290525 -0.175023 -0.67333  0.264396
## row29  0.561163  0.238126  0.347851 -0.239475  0.88229 -0.454358
## row30  0.294810  0.672463  0.013193  0.089305  0.05599 -0.444167
## row31  0.520212 -0.117833  0.505750  0.058177 -0.04588  0.506952
## row32  0.679617  0.223004  0.372613  0.377652 -0.34348 -0.087277
## row33 -0.043987  0.244517  0.149534 -0.326693  0.63019  0.396483
## row34 -0.191807  0.631411  0.213954 -0.289012  0.15988  0.169792
## row35  0.460770  0.404081  0.006818  0.020909  0.06670 -0.448503
## row36  0.009364  0.583820  0.119411  0.239266 -0.27455 -0.575313
## row37  0.331998  0.089081  0.449743  0.038214  0.15789  0.071349
## row38 -0.279893  0.609199 -0.030333 -0.144736 -0.25980  0.576473
## row39  0.261087  0.155062 -0.572465 -0.172904  0.44691  0.319909
## row40  0.152359  0.389691 -0.400201  0.537092  0.07334  0.405173
## row41  0.779430  0.180476 -0.610367  0.130056  0.18986 -0.375367
## row42  0.586373  0.488076 -0.425722  0.318004 -0.30596  0.003282
##
##
## Site constraints (linear combinations of constraining variables)
##
##          RDA1      RDA2      RDA3      RDA4      RDA5      RDA6
## row1   -0.133123 -0.24333  0.050402 -0.17021  0.693128 -0.35389
## row2   -0.335729  0.06734  0.018173  0.17376  0.362387 -0.41678
## row3    0.323982 -0.67668  0.497232  0.34325  0.065938  0.03358
## row4    0.121376 -0.36601  0.465002  0.68721 -0.264802 -0.02931
## row5   -0.314272 -0.36546  0.282450  0.14451  0.616393  0.17105
## row6   -0.516878 -0.05479  0.250220  0.48848  0.285652  0.10816
## row7   -0.219372 -0.37114 -0.019393  0.07220  0.145386 -0.65686
## row8   -0.421978 -0.06047 -0.051622  0.41617 -0.185355 -0.71975
## row9   -0.272614 -0.24236  0.027586 -0.04738  0.272477  0.32119
## row10  -0.475220  0.06831 -0.004644  0.29659 -0.058264  0.25830
## row11  -0.176208 -0.36299 -0.424544  0.26681  0.363482  0.28979
## row12  -0.378814 -0.05231 -0.456774  0.61078  0.032741  0.22690
## row13   0.287802 -0.63885 -0.533935  0.07912  0.234988 -0.08658
## row14   0.085196 -0.32817 -0.566165  0.42309 -0.095753 -0.14947
## row15  -0.080966 -0.06743  0.106991 -0.75626  0.238115 -0.21511
## row16  -0.283571  0.24324  0.074761 -0.41229 -0.092626 -0.27800
## row17   0.376139 -0.50078  0.553821 -0.24280 -0.389075  0.17236
## row18   0.173533 -0.19011  0.521591  0.10117 -0.719815  0.10947
## row19  -0.262114 -0.18956  0.339039 -0.44153  0.161380  0.30983
## row20  -0.464720  0.12111  0.306809 -0.09756 -0.169361  0.24694
## row21  -0.167214 -0.19524  0.037196 -0.51385 -0.309627 -0.51808
## row22  -0.369820  0.11543  0.004967 -0.16988 -0.640368 -0.58097
## row23  -0.220456 -0.06645  0.084175 -0.63342 -0.182536  0.45997
## row24  -0.423062  0.24422  0.051945 -0.28945 -0.513277  0.39708

```

```

## row25 -0.124051 -0.18708 -0.367955 -0.31923 -0.091531 0.42857
## row26 -0.326657 0.12359 -0.400185 0.02474 -0.422272 0.36569
## row27 0.339960 -0.46294 -0.477346 -0.50692 -0.220025 0.05220
## row28 0.137354 -0.15227 -0.509576 -0.16295 -0.550766 -0.01069
## row29 0.334545 0.35796 0.093673 -0.39511 0.619198 -0.27760
## row30 0.131939 0.66863 0.061443 -0.05114 0.288458 -0.34049
## row31 0.791649 -0.07539 0.540503 0.11835 -0.007991 0.10987
## row32 0.589044 0.23528 0.508273 0.46232 -0.338732 0.04698
## row33 0.153396 0.23583 0.325721 -0.08038 0.542463 0.24734
## row34 -0.049210 0.54650 0.293491 0.26359 0.211722 0.18445
## row35 0.248296 0.23015 0.023878 -0.15270 0.071456 -0.58057
## row36 0.045690 0.54082 -0.008352 0.19127 -0.259284 -0.64346
## row37 0.195054 0.35893 0.070856 -0.27227 0.198548 0.39748
## row38 -0.007552 0.66961 0.038627 0.07170 -0.132193 0.33459
## row39 0.291459 0.23831 -0.381274 0.04192 0.289552 0.36609
## row40 0.088853 0.54898 -0.413503 0.38589 -0.041189 0.30320
## row41 0.755470 -0.03755 -0.490664 -0.14577 0.161058 -0.01028
## row42 0.552864 0.27312 -0.522894 0.19820 -0.169683 -0.07317
##
##
## Biplot scores for constraining variables
##
##          RDA1      RDA2      RDA3      RDA4      RDA5      RDA6
## initopen -0.2997  0.45953 -0.047673  0.50878 -0.48922 -0.093023
## block2   0.4782 -0.31676  0.621256  0.29579 -0.33302  0.089163
## block3   -0.2926  0.05910  0.361860  0.05578  0.33177  0.255182
## block4   -0.1780  0.05224 -0.002682 -0.03156 -0.23707 -0.744702
## block5   -0.2423  0.20778  0.054054 -0.17597 -0.08358  0.436512
## block6   -0.1259  0.06209 -0.491993  0.20348  0.02633  0.398597
## block7   0.4345 -0.27107 -0.624106 -0.02320 -0.12886 -0.055956
## timet1   -0.2534 -0.17396  0.048745 -0.66046 -0.58299  0.140340
## timet2   0.6158  0.71588  0.020885  0.09501  0.21417  0.009627
##
##
## Centroids for factor constraints
##
##          RDA1      RDA2      RDA3      RDA4      RDA5      RDA6
## initlog   0.10130 -0.15534  0.016115 -0.17198  0.16537  0.031445
## initopen -0.10130  0.15534 -0.016115  0.17198 -0.16537 -0.031445
## block1   -0.06115  0.17107  0.067574 -0.26854  0.35144 -0.313645
## block2   0.39595 -0.26228  0.514404  0.24492 -0.27575  0.073827
## block3   -0.24230  0.04894  0.299622  0.04619  0.27471  0.211292
## block4   -0.14740  0.04326 -0.002221 -0.02613 -0.19630 -0.616617
## block5   -0.20064  0.17204  0.044757 -0.14571 -0.06921  0.361434
## block6   -0.10424  0.05141 -0.407373  0.16848  0.02180  0.330041
## block7   0.35977 -0.22445 -0.516763 -0.01921 -0.10670 -0.046332
## timet0   -0.17328 -0.25906 -0.033287  0.27031  0.17631 -0.071691
## timet1   -0.12112 -0.08316  0.023302 -0.31573 -0.27870  0.067089
## timet2   0.29439  0.34223  0.009984  0.04542  0.10238  0.004602

anova.cca(trt_tot_2, step=1000, by="term") ## test for model significance

```

```

## Permutation test for rda under reduced model
## Terms added sequentially (first to last)

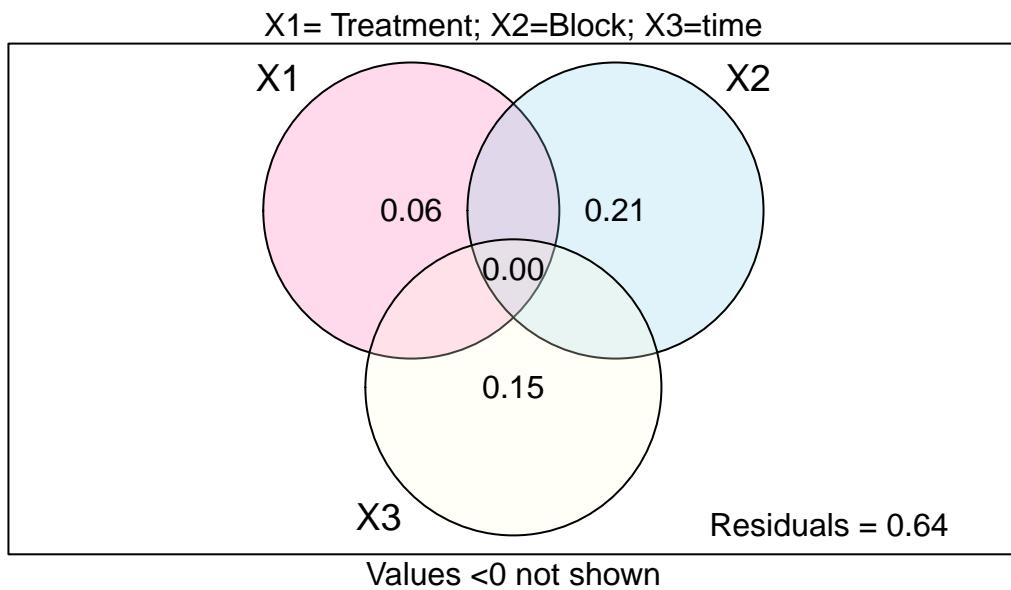
```

```

## Permutation: free
## Number of permutations: 999
##
## Model: rda(formula = ass.rel.t012 ~ init + block + time)
##          Df Variance      F Pr(>F)
## init      1 0.034157 3.9233 0.001 ***
## block     6 0.160594 3.0743 0.001 ***
## time      2 0.088406 5.0773 0.001 ***
## Residual  32 0.278596
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

# can model using varpart to look at contributions of initial treatment and block
var.mod2<-varpart(ass.rel.t012, init, block, time) # run model on standardized data
plot(var.mod2, bg=c("hotpink","skyblue","lightyellow"))
mtext("X1= Treatment; X2=Block; X3=time", side=3)

```



```

## can test for significance of contribution of the fraction of initial treatment
# do this with partial redundancy analysis
trt_Frac<-rda(ass.rel.t012~ init +Condition(block + time)) # partial rda model
summary(trt_Frac)

```

```

##
## Call:
## rda(formula = ass.rel.t012 ~ init + Condition(block + time))

```

```

##
## Partitioning of variance:
##           Inertia Proportion
## Total      0.56175   1.0000
## Conditioned 0.24900   0.4433
## Constrained 0.03416   0.0608
## Unconstrained 0.27860   0.4959
##
## Eigenvalues, and their contribution to the variance
## after removing the contribution of conditioning variables
##
## Importance of components:
##           RDA1    PC1    PC2    PC3    PC4    PC5    PC6
## Eigenvalue      0.03416 0.0397 0.02773 0.02383 0.02008 0.01758 0.01589
## Proportion Explained 0.10921 0.1269 0.08867 0.07618 0.06419 0.05620 0.05080
## Cumulative Proportion 0.10921 0.2362 0.32482 0.40101 0.46520 0.52139 0.57219
##           PC7    PC8    PC9    PC10   PC11   PC12   PC13
## Eigenvalue      0.01486 0.01326 0.01138 0.01024 0.01004 0.008831 0.008658
## Proportion Explained 0.04752 0.04240 0.03637 0.03273 0.03211 0.028235 0.027685
## Cumulative Proportion 0.61971 0.66211 0.69848 0.73121 0.76332 0.791553 0.819237
##           PC14   PC15   PC16   PC17   PC18   PC19
## Eigenvalue      0.007353 0.006713 0.005727 0.005255 0.004236 0.003939
## Proportion Explained 0.023511 0.021466 0.018313 0.016802 0.013546 0.012596
## Cumulative Proportion 0.842748 0.864213 0.882526 0.899328 0.912874 0.925469
##           PC20   PC21   PC22   PC23   PC24   PC25
## Eigenvalue      0.003339 0.003025 0.002930 0.002555 0.002392 0.002270
## Proportion Explained 0.010676 0.009671 0.009367 0.008169 0.007648 0.007257
## Cumulative Proportion 0.936145 0.945816 0.955183 0.963353 0.971001 0.978257
##           PC26   PC27   PC28   PC29   PC30   PC31
## Eigenvalue      0.001666 0.001540 0.001124 0.0009868 0.0006454 0.0005422
## Proportion Explained 0.005326 0.004923 0.003594 0.0031552 0.0020636 0.0017336
## Cumulative Proportion 0.983583 0.988506 0.992100 0.9952553 0.9973189 0.9990524
##           PC32
## Eigenvalue      0.0002963
## Proportion Explained 0.0009476
## Cumulative Proportion 1.0000000
##
## Accumulated constrained eigenvalues
## Importance of components:
##           RDA1
## Eigenvalue      0.03416
## Proportion Explained 1.00000
## Cumulative Proportion 1.00000
##
## Scaling 2 for species and site scores
## * Species are scaled proportional to eigenvalues
## * Sites are unscaled: weighted dispersion equal on all dimensions
## * General scaling constant of scores: 2.190697
##
## 
## Species scores
##
##           RDA1      PC1      PC2      PC3      PC4      PC5
## acul    0.0755447 -1.434e-02 7.884e-02 -1.593e-03 8.467e-02 2.948e-02

```

```

## aicu  0.0138510 -5.744e-03 -2.254e-02 -3.823e-02 -1.389e-02 -2.313e-02
## arca  0.0000000 -9.271e-18 -2.981e-17 -4.432e-18  1.136e-17 -5.090e-17
## ardy  0.0061606 -3.600e-02  2.215e-02  4.187e-02  7.286e-03 -9.784e-05
## arsp  0.0086051  7.583e-03  1.074e-02  5.817e-03  9.230e-03  2.952e-03
## auel  0.0000000 -4.520e-17 -3.721e-17  1.033e-17  1.268e-17  4.689e-17
## bldr  0.0415582  1.024e-01 -5.538e-02  3.641e-02  6.442e-02 -6.221e-03
## blrd  0.0183399  5.422e-03  2.508e-02  2.563e-03  1.299e-02  1.353e-02
## brdi  0.0119918  9.759e-03  2.022e-03 -2.539e-03  2.480e-03 -4.325e-03
## brdr  0.0000000 -7.790e-18 -1.962e-17  2.861e-17 -4.322e-18 -1.126e-17
## brpe  -0.0548396  5.401e-02 -1.678e-02 -2.939e-02 -2.420e-02 -8.146e-03
## brru  0.0000000  2.293e-17 -1.930e-17 -3.465e-18 -2.242e-17 -2.590e-17
## buse  -0.0161408 -3.475e-03  1.721e-02 -2.478e-04 -3.937e-03  9.186e-03
## caer  -0.0452451 -6.696e-02  9.202e-02 -1.616e-01  4.003e-02 -3.100e-02
## cagr  -0.0551717 -3.321e-02 -1.867e-02 -6.880e-02 -7.224e-02 -3.597e-02
## cahi  -0.0257702 -1.225e-02  5.902e-02  2.155e-02 -1.343e-02 -6.753e-03
## casp  0.0015456 -1.514e-02  8.747e-03  2.929e-02  6.102e-02  1.539e-02
## cear  0.0183399  9.032e-03  3.008e-03  7.984e-03 -3.408e-02 -4.754e-02
## chau  0.0711459  2.373e-02  4.398e-02 -1.230e-01 -5.926e-02 -5.960e-02
## chei  0.0401993  7.383e-03 -2.523e-02  6.336e-03 -3.648e-02  4.773e-02
## chps  0.1934722 -2.173e-02  3.508e-02  7.579e-02  1.279e-01  3.187e-02
## crcl  -0.0071908 -4.839e-02  6.939e-03 -3.034e-02 -3.984e-03  1.194e-03
## crco  0.0909945  1.388e-02  3.934e-02  1.367e-01  3.854e-02 -7.024e-02
## cusc  -0.0215871 -7.383e-02  3.359e-02 -5.510e-02 -1.432e-02  1.409e-02
## cusp  0.0102164 -1.830e-03  2.398e-02 -2.390e-03  8.359e-03  1.051e-02
## dagl  -0.0002086 -1.267e-02 -3.465e-03  4.740e-03  5.809e-03  5.524e-03
## dosp  -0.0228091 -7.269e-02  7.834e-02  1.285e-01 -1.337e-02 -7.897e-02
## ento  0.0000000 -3.969e-18 -6.278e-18 -1.112e-17 -2.983e-19  5.476e-18
## erau  -0.0150169 -3.738e-02  1.412e-02 -2.351e-02 -7.513e-03  8.236e-03
## ercy  -0.0354798 -1.267e-02 -4.291e-02  1.901e-02 -9.685e-03 -3.599e-02
## erra  0.0227357 -1.122e-02  2.726e-02  2.923e-03 -1.143e-02  7.106e-03
## ersp  -0.0084795 -7.754e-03  3.923e-03 -1.604e-03 -1.582e-02  6.429e-03
## gite  -0.0130745  3.665e-02 -9.580e-02 -1.441e-02  6.247e-02  9.662e-03
## gnte  0.0449263  6.207e-03  3.387e-02 -5.636e-02 -6.490e-02 -1.134e-02
## gobe  -0.0389147 -1.221e-02  2.324e-03  2.536e-02  1.356e-02  2.056e-01
## gocy  -0.0005179  3.628e-03  4.776e-02  8.507e-03  3.434e-04 -3.315e-02
## gono  0.0000000  5.463e-18  1.998e-18 -1.741e-18 -8.489e-18  6.585e-19
## goro  0.1287318 -6.246e-02 -1.634e-01  6.953e-02 -2.130e-01  3.958e-02
## gosp  0.0180699  4.609e-03  2.551e-02 -1.270e-02  2.529e-02 -1.785e-02
## haod  0.0085083  2.446e-02  2.793e-03  2.387e-02  5.230e-02 -9.922e-03
## hygl  0.1305498  2.905e-03  1.209e-01  8.285e-02 -5.754e-02  2.925e-02
## hypi  -0.0150169  1.725e-02  4.290e-03 -2.482e-02 -1.172e-02 -1.151e-02
## hypo  -0.0582026 -3.197e-02  8.701e-02  6.189e-02 -2.330e-02  3.009e-02
## jubu  0.0000000  0.000e+00  0.000e+00  0.000e+00  0.000e+00  0.000e+00
## laro  -0.1001787 -5.003e-02  2.308e-02  1.384e-02  3.061e-02 -5.441e-02
## ledu  0.0018314  2.583e-02  1.169e-03  1.944e-02  9.125e-03  1.162e-02
## lele  0.0170846  3.174e-03  7.089e-03  5.462e-03  4.878e-04  1.083e-02
## loef  0.0119918  3.267e-03 -7.624e-04  3.625e-03 -1.161e-02  1.227e-02
## misp  -0.0677312 -9.016e-02  5.996e-02 -1.150e-01 -4.657e-02 -5.294e-02
## mite  -0.0485321  2.148e-01 -5.397e-02 -5.310e-02  3.580e-02  5.387e-02
## momo  0.0000000  0.000e+00  0.000e+00  0.000e+00  0.000e+00  0.000e+00
## mopa  0.0000000  0.000e+00  0.000e+00  0.000e+00  0.000e+00  0.000e+00
## niro  -0.0157499  3.081e-02 -8.896e-03  1.270e-02 -3.637e-03 -2.570e-03
## omco  0.0135532 -9.910e-03 -1.428e-02  4.070e-02 -3.485e-04  1.916e-02
## orsp  -0.0106186 -1.238e-03 -5.492e-03 -1.036e-02  5.195e-05 -5.661e-03

```

```

## pala  0.0086051  7.583e-03  1.074e-02  5.817e-03  9.230e-03  2.952e-03
## peai  0.1028116  1.736e-01 -2.005e-01 -3.081e-02  1.369e-02  2.640e-02
## pedu  0.0121694  1.072e-02  1.520e-02  8.226e-03  1.305e-02  4.174e-03
## phsu  -0.0111369  2.179e-02 -6.290e-03  8.980e-03 -2.571e-03 -1.818e-03
## plde  0.0000000  0.000e+00  0.000e+00  0.000e+00  0.000e+00  0.000e+00
## poar  0.0027883 -1.889e-02  1.721e-02 -6.389e-02 -1.656e-02  6.553e-03
## poca  0.0062046  8.808e-02 -6.471e-02 -2.795e-03 -6.767e-02  1.822e-02
## pocap 0.0045075  8.631e-03  3.229e-02 -3.460e-03  2.430e-02  1.976e-02
## poce  -0.0106186 -1.238e-03 -5.492e-03 -1.036e-02  5.195e-05 -5.661e-03
## pogn  0.0000000  0.000e+00  0.000e+00  0.000e+00  0.000e+00  0.000e+00
## pole  -0.0298641  1.578e-01  3.998e-02 -1.008e-01  5.034e-03 -8.070e-03
## pomu  0.1993083 -3.328e-01 -2.100e-01  8.524e-03  8.303e-02 -6.531e-02
## pter  0.0313036 -8.137e-03 -1.022e-02 -6.534e-03  8.000e-03 -2.134e-02
## ptga  0.0518059  6.918e-03  1.242e-01 -6.995e-02 -3.267e-02  1.731e-02
## ptob  0.0000000  0.000e+00  0.000e+00  0.000e+00  0.000e+00  0.000e+00
## rhla  -0.0192925  8.336e-02 -3.393e-02 -1.024e-01  7.481e-02 -1.291e-02
## rhpy  -0.0745613  1.446e-02 -4.187e-02 -1.141e-02 -1.525e-01 -5.915e-02
## rhsp  -0.0241808  1.203e-02  1.080e-01  8.131e-02 -8.713e-02  9.129e-02
## ry    0.0000000  0.000e+00  0.000e+00  0.000e+00  0.000e+00  0.000e+00
## scna  0.0187550  6.243e-03  1.596e-02 -2.118e-02 -3.013e-02 -1.789e-02
## sino  -0.0150169 -3.738e-02  1.412e-02 -2.351e-02 -7.513e-03  8.236e-03
## sool  -0.0138136 -1.380e-02 -7.854e-03  7.919e-03 -7.863e-03 -1.252e-02
## stfi  -0.0194231 -8.594e-03  6.600e-03 -1.169e-02  1.458e-02 -5.568e-03
## stpi  0.0000000  0.000e+00  0.000e+00  0.000e+00  0.000e+00  0.000e+00
## thma  0.0000000  0.000e+00  0.000e+00  0.000e+00  0.000e+00  0.000e+00
## trcy  -0.2242559  2.113e-01  1.067e-02  1.038e-01  8.545e-03 -2.070e-01
## tris  0.0402117 -3.209e-02  1.911e-02 -8.204e-03 -1.746e-02 -5.635e-02
## tror  -0.1544269 -6.830e-02 -3.490e-02 -8.891e-02  5.911e-02  1.022e-02
## trpi  -0.0437537  1.461e-02 -2.404e-02  7.331e-02  2.214e-02  1.813e-03
## waac  -0.1356532 -3.713e-02  4.244e-02 -4.687e-02  6.527e-03 -2.002e-02
## wagr  -0.0065153 -8.588e-03  6.302e-03  2.518e-03  2.094e-03  3.230e-02
## x     -0.0133788 -5.871e-03  2.583e-02 -1.927e-04  4.224e-02  1.243e-03
##
##
## Site scores (weighted sums of species scores)
##
##          RDA1      PC1      PC2      PC3      PC4      PC5
## row1  -0.26449 -0.11731  0.072142 -0.1786127  0.003365 -0.186582
## row2   0.32774 -0.25141 -0.315084 -0.1260442 -0.262245 -0.115106
## row3  -0.46922  0.11517 -0.268870 -0.3406530  0.698008 -0.329868
## row4   0.27897  0.21540 -0.700246 -0.4535292 -0.020427  0.244714
## row5  -0.39084  0.31262 -0.122452  0.2712654 -0.103664 -0.151995
## row6   0.25593  0.23780  0.111247  0.0077685 -0.106470 -0.024464
## row7  -0.28933 -0.30912  0.156394 -0.0639535 -0.630565  0.256295
## row8   0.22871  0.03594  0.165376  0.0061877  0.158766  0.186223
## row9  -0.14623  0.13592 -0.190179  0.4505984 -0.907619  0.085541
## row10  0.63059 -0.41985 -0.221834  0.3855205  0.244325  0.427107
## row11  0.01622 -0.02003  0.033700 -0.0067724  0.106785  0.178492
## row12  0.21309  0.29790  0.422079  0.2285040  0.362588  0.115951
## row13 -0.43323 -0.28242  0.464241 -0.0511444  0.557801  0.003577
## row14  0.04211  0.04939  0.393486 -0.1291351 -0.100648 -0.689887
## row15 -0.20945 -0.53535 -0.191302 -0.4967942  0.149392 -0.088724
## row16  0.48864  0.11795  0.004154  0.4405109  0.555465 -0.021861
## row17 -0.29523  0.06965 -0.199387  0.6302322  0.230438  0.049894

```

```

## row18  0.45621  0.06090  0.509639 -0.8875723 -0.461571  0.207254
## row19 -0.49980  0.66131 -0.190927  0.2725541 -0.078050 -0.055167
## row20  0.45815  0.10541  0.194447  0.2727072  0.304887  0.350394
## row21 -0.15821 -0.13366 -0.241084 -0.0026471  0.350202 -0.173006
## row22  0.08433  0.09447  0.730090 -0.1782412  0.173954  0.148209
## row23 -0.87372  0.38836  0.096577 -0.5587161 -0.263845 -0.259079
## row24  0.54389 -0.30905 -0.033889  0.1578082  0.214942  0.339469
## row25 -0.19692 -0.33779 -0.192198  0.1937864 -0.192412 -0.306358
## row26  0.31799 -0.17579 -0.145749 -0.0141532 -0.348512  0.314036
## row27 -0.30510 -0.17289 -0.395816  0.0233742 -0.006757  0.371098
## row28  0.18924  0.16647  0.055446  0.1471508 -0.628134 -0.876158
## row29 -0.59762  0.26775  0.658661  0.2113891 -0.436792  0.732128
## row30  0.25517  0.51838 -0.228572  0.1495511 -0.009184 -0.319855
## row31 -0.23382 -0.55322  0.680356  0.9493292 -0.119289 -0.517903
## row32  0.26309  0.09208 -0.021492  0.1021932 -0.327159  0.345909
## row33 -0.22411 -0.84145  0.317819 -0.5291453 -0.169109  0.185397
## row34  0.40068 -0.47569 -0.310134 -0.2951499  0.152405 -0.304166
## row35 -0.26668  0.40753 -0.725183  0.0061752 -0.074115 -0.128422
## row36  0.40119 -0.09516 -0.085593  0.2324789  0.021757 -0.289300
## row37 -0.68266  0.77489  0.513757 -0.4592464  0.421879 -0.215783
## row38  0.52814 -0.57026 -0.164432  0.0240354  0.290319 -0.377254
## row39 -0.49625 -0.03940 -0.174828 -0.3297816  0.001654 -0.180206
## row40  0.14586  0.27510  0.056997 -0.0715832  0.069898 -0.121915
## row41 -0.08197  0.20944 -0.101421  0.0087625  0.462695  0.730672
## row42  0.58896  0.03001 -0.415936  0.0009919 -0.284957  0.460698
##
##
## Site constraints (linear combinations of constraining variables)
##
##          RDA1      PC1      PC2      PC3      PC4      PC5
## row1   -0.338 -0.11731  0.072142 -0.1786127  0.003365 -0.186582
## row2    0.338 -0.25141 -0.315084 -0.1260442 -0.262245 -0.115106
## row3   -0.338  0.11517 -0.268870 -0.3406530  0.698008 -0.329868
## row4    0.338  0.21540 -0.700246 -0.4535292 -0.020427  0.244714
## row5   -0.338  0.31262 -0.122452  0.2712654 -0.103664 -0.151995
## row6    0.338  0.23780  0.111247  0.0077685 -0.106470 -0.024464
## row7   -0.338 -0.30912  0.156394 -0.0639535 -0.630565  0.256295
## row8    0.338  0.03594  0.165376  0.0061877  0.158766  0.186223
## row9   -0.338  0.13592 -0.190179  0.4505984 -0.907619  0.085541
## row10   0.338 -0.41985 -0.221834  0.3855205  0.244325  0.427107
## row11   -0.338 -0.02003  0.033700 -0.0067724  0.106785  0.178492
## row12   0.338  0.29790  0.422079  0.2285040  0.362588  0.115951
## row13   -0.338 -0.28242  0.464241 -0.0511444  0.557801  0.003577
## row14   0.338  0.04939  0.393486 -0.1291351 -0.100648 -0.689887
## row15   -0.338 -0.53535 -0.191302 -0.4967942  0.149392 -0.088724
## row16   0.338  0.11795  0.004154  0.4405109  0.555465 -0.021861
## row17   -0.338  0.06965 -0.199387  0.6302322  0.230438  0.049894
## row18   0.338  0.06090  0.509639 -0.8875723 -0.461571  0.207254
## row19   -0.338  0.66131 -0.190927  0.2725541 -0.078050 -0.055167
## row20   0.338  0.10541  0.194447  0.2727072  0.304887  0.350394
## row21   -0.338 -0.13366 -0.241084 -0.0026471  0.350202 -0.173006
## row22   0.338  0.09447  0.730090 -0.1782412  0.173954  0.148209
## row23   -0.338  0.38836  0.096577 -0.5587161 -0.263845 -0.259079
## row24   0.338 -0.30905 -0.033889  0.1578082  0.214942  0.339469

```

```

## row25 -0.338 -0.33779 -0.192198  0.1937864 -0.192412 -0.306358
## row26  0.338 -0.17579 -0.145749 -0.0141532 -0.348512  0.314036
## row27 -0.338 -0.17289 -0.395816  0.0233742 -0.006757  0.371098
## row28  0.338  0.16647  0.055446  0.1471508 -0.628134 -0.876158
## row29 -0.338  0.26775  0.658661  0.2113891 -0.436792  0.732128
## row30  0.338  0.51838 -0.228572  0.1495511 -0.009184 -0.319855
## row31 -0.338 -0.55322  0.680356  0.9493292 -0.119289 -0.517903
## row32  0.338  0.09208 -0.021492  0.1021932 -0.327159  0.345909
## row33 -0.338 -0.84145  0.317819 -0.5291453 -0.169109  0.185397
## row34  0.338 -0.47569 -0.310134 -0.2951499  0.152405 -0.304166
## row35 -0.338  0.40753 -0.725183  0.0061752 -0.074115 -0.128422
## row36  0.338 -0.09516 -0.085593  0.2324789  0.021757 -0.289300
## row37 -0.338  0.77489  0.513757 -0.4592464  0.421879 -0.215783
## row38  0.338 -0.57026 -0.164432  0.0240354  0.290319 -0.377254
## row39 -0.338 -0.03940 -0.174828 -0.3297816  0.001654 -0.180206
## row40  0.338  0.27510  0.056997 -0.0715832  0.069898 -0.121915
## row41 -0.338  0.20944 -0.101421  0.0087625  0.462695  0.730672
## row42  0.338  0.03001 -0.415936  0.0009919 -0.284957  0.460698
##
##
## Biplot scores for constraining variables
##
##          RDA1 PC1 PC2 PC3 PC4 PC5
## initopen    1   0   0   0   0   0
## 
## 
## Centroids for factor constraints
##
##          RDA1 PC1 PC2 PC3 PC4 PC5
## initlog  -0.338   0   0   0   0   0
## initopen  0.338   0   0   0   0   0

```

```
RsquareAdj(trt_Frac)$adj.r.squared #explanatory power
```

```
## [1] 0.05628909
```

```
anova.cca(trt_Frac) ## this tells us if first condition, init, significantly contributes to overall variance
```

```

## Permutation test for rda under reduced model
## Permutation: free
## Number of permutations: 999
##
## Model: rda(formula = ass.rel.t012 ~ init + Condition(block + time))
##          Df Variance      F Pr(>F)
## Model      1 0.034157 3.9233  0.001 ***
## Residual 32 0.278596
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

```

```
### extracting species scores and plotting
```

```
# species scores
```

```
species.scores<-as.data.frame(vegan::scores(ass.rel.t012_NMS,"species")) ## some species don't have scores
```

```

species.scores$species<-rownames(species.scores)

### NMDS 1 and 2
log<-mds_scores_t012[mds_scores_t012$treatment == "log", ][chull(mds_scores_t012[mds_scores_t012$treatment == "log", c("NMDS1", "NMDS2"))], ]

open<-mds_scores_t012[mds_scores_t012$treatment == "open", ][chull(mds_scores_t012[mds_scores_t012$treatment == "open", c("NMDS1", "NMDS2"))], ]

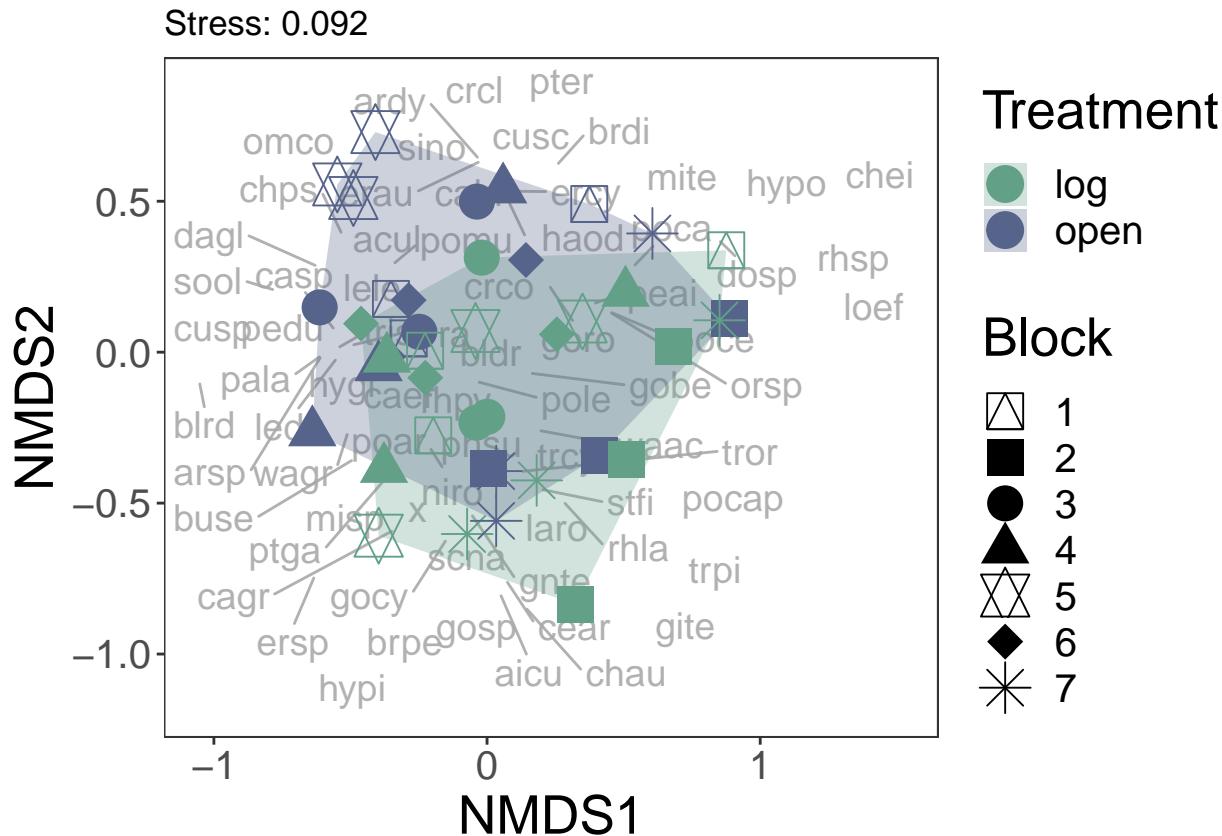
hulldat<-rbind(log,open)

options(ggrepel.max.overlaps = Inf)
nmuds.plot <- ggplot()+
  theme_bw()+
  theme(panel.background = element_blank(),
        panel.grid.major = element_blank(), #remove major-grid labels
        panel.grid.minor = element_blank(), #remove minor-grid labels
        plot.background = element_blank(),
        axis.text = element_text(size = 15),
        axis.title=element_text(size=20),
        legend.title=element_text(size=20),
        legend.text=element_text(size=15))+

  geom_text_repel(data=species.scores, aes(NMDS1, NMDS2, label=species), alpha=0.9, size=5, col='darkgreen')
  geom_polygon(data=hulldat, aes(NMDS1, NMDS2, fill=treatment, group=treatment), alpha=0.3)+scale_fill_manual(values=c("#63A088", "#56638A"), name="Treatment") +
  geom_point(data=mds_scores_t012, aes(NMDS1, NMDS2, shape=block, col=treatment), size=6)+ scale_shape_manual(values=c(1, 2, 3, 4, 5, 6))
  scale_colour_manual(values=c("#63A088", "#56638A"), name="Treatment") +
  labs(title=paste0("Stress: ", round(ass.rel.t012_NMS$stress,3)))

print(nmuds.plot)

```



Interpretation

Q2 Why are plant communities in fallen log patches different from patches in the open?

Overview of results Nutrient comparison Total soil carbon is significantly higher in log patches than open patches (Wilcoxon rank sum test, $p<0.05$)

2020-2022 There are three response variables: count (Poisson), Shannon diversity index (hurdle: binomial and Gaussian), and composition matrix. 1. *Abundance* – the inclusive model includes all nutrient elements as additive fixed factors with block and year as random terms. As suggested by this inclusive model, the abundance of plants is significantly more likely to increase with soil carbon (C: $p=0.004$) level but decrease with plant-available nitrogen (NH_4N : $p<0.001$) and basic exchangeable cations (CEC: $p=0.008$). [This seems to be a bit confounding as it does not make sense biologically. Maybe an artefact of collinearity?] 2. *Diversity* – the probability of having a positive Shannon diversity index in the plant community is significantly higher with higher levels of plant-available nitrogen (NO_3N : $p < 0.05$) and potassium (K: $p < 0.05$) but lower with higher soil phosphorus (P: $p < 0.05$). A similar pattern is observed for the level of diversity. The presented model has a structure of “diversity ~ elements (additive) + (1|block) + (1|year)”. 3. *Composition* – The composition of plant communities is not significantly explained by any of the nutrient elements. See print(en.nutrient).

2020 There are three response variables: count (Poisson), Shannon diversity index (hurdle: binomial and Gaussian), and composition matrix. 1. *Abundance* – the inclusive model includes all nutrient elements as additive fixed factors with block and year as random terms. As suggested by this inclusive model, the abundance of plants is significantly more likely to increase with soil carbon (C: $p=0.04$) level. 2. *Diversity* – the probability of having a positive Shannon diversity index in the plant community is significantly higher

with higher levels of plant-available nitrogen (NO_3N : $p < 0.05$) and potassium (K : $p < 0.05$). A similar pattern is observed for the level of diversity. The presented model has a structure of “diversity ~ elements (additive) + (1|block)”. 3. *Composition* – The composition of plant communities is not significantly explained by any of the nutrient elements. See `print(en.nutrient)`.

2021 There are three response variables: count (Poisson), Shannon diversity index (hurdle: binomial and Gaussian), and composition matrix. 1. *Abundance* – the inclusive model includes all nutrient elements as additive fixed factors with block and year as random terms. As suggested by this inclusive model, the abundance of plants is significantly more likely to increase with plant-available phosphorus (P : $p=0.03$) level but decrease with plant-available nitrogen (NH_4N : $p=0.009$). [Same comment] 2. *Diversity* – the level of plant diversity as represented by the Shannon diversity index and the probability of having a positive Shannon diversity index in the plant community are not significantly explained by any soil nutrient elements. The presented model has a structure of “diversity ~ elements (additive) + (1|block)”. 3. *Composition* – The composition of plant communities is not significantly explained by any of the nutrient elements. See `print(en.nutrient)`.

2022 There are three response variables: count (Poisson), Shannon diversity index (hurdle: binomial and Gaussian), and composition matrix. 1. *Abundance* – the inclusive model includes all nutrient elements as additive fixed factors with block and year as random terms. As suggested by this inclusive model, the abundance of plants is significantly more likely to decrease with plant-available nitrogen (NH_4N : $p<0.001$) and potassium (K : $p=0.03$). [Same comment] 2. *Diversity* – the level of plant diversity as represented by the Shannon diversity index and the probability of having a positive Shannon diversity index in the plant community are not significantly explained by any soil nutrient elements. The presented model has a structure of “diversity ~ elements (additive) + (1|block)”. 3. *Composition* – The composition of plant communities is not significantly explained by any of the nutrient elements. See `print(en.nutrient)`.

Statistical Methods

Analysis

- nutrient analysis (nutrient composition ~ log vs open; abundance ~ nutrient composition , diversity ~ nutrient composition, composition ~ nutrient composition)

```
library(ggplot2)
library(ggpubr)
library(dplyr)
library(lme4)
library(emmeans)
library(pscl)
```

```
## Classes and Methods for R originally developed in the
## Political Science Computational Laboratory
## Department of Political Science
## Stanford University (2002–2015),
## by and under the direction of Simon Jackman.
## hurdle and zeroinfl functions by Achim Zeileis.
```

```
library(glmmTMB)
```

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

```

library(tidyr)
library(DHARMA)
library(ggplot2)
library(AICcmodavg)

##
## Attaching package: 'AICcmodavg'

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

## The following object is masked from 'package:labdsv':
##   importance

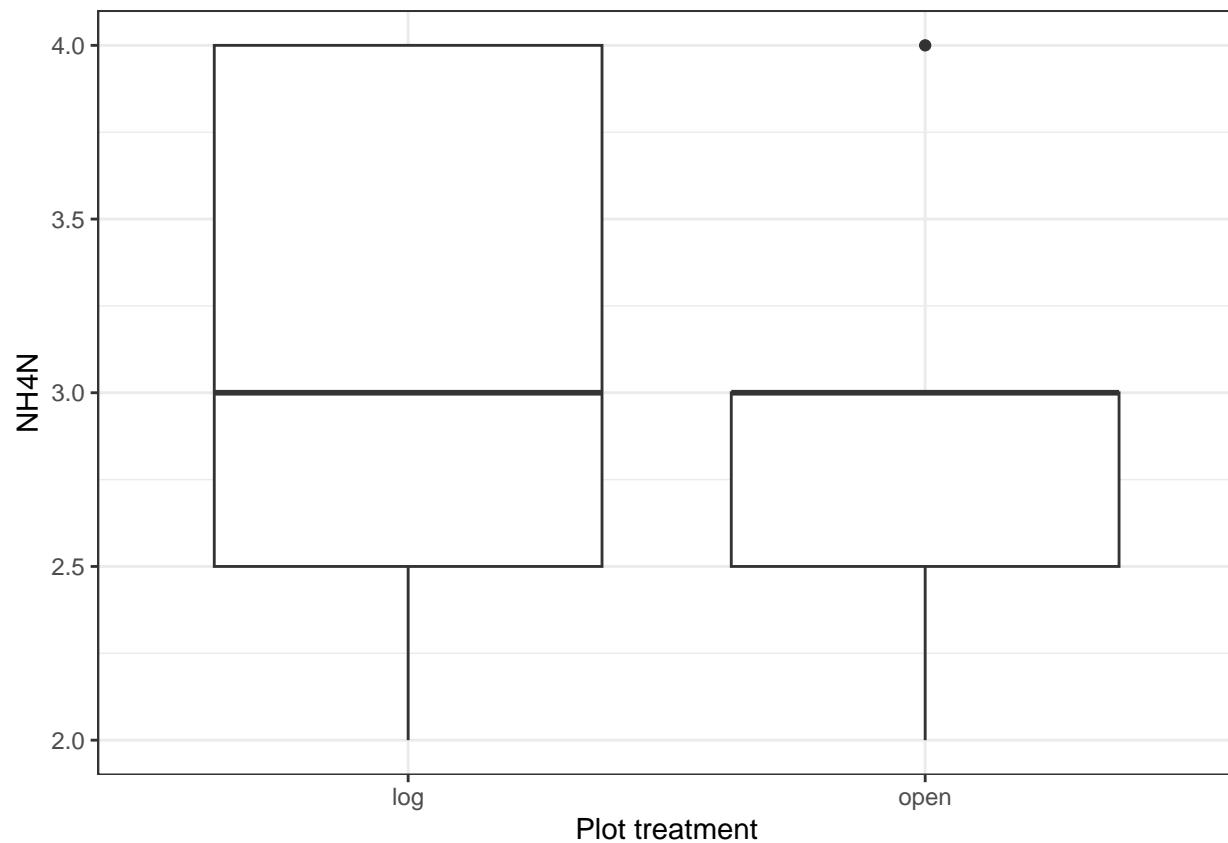
library(ggpubr)

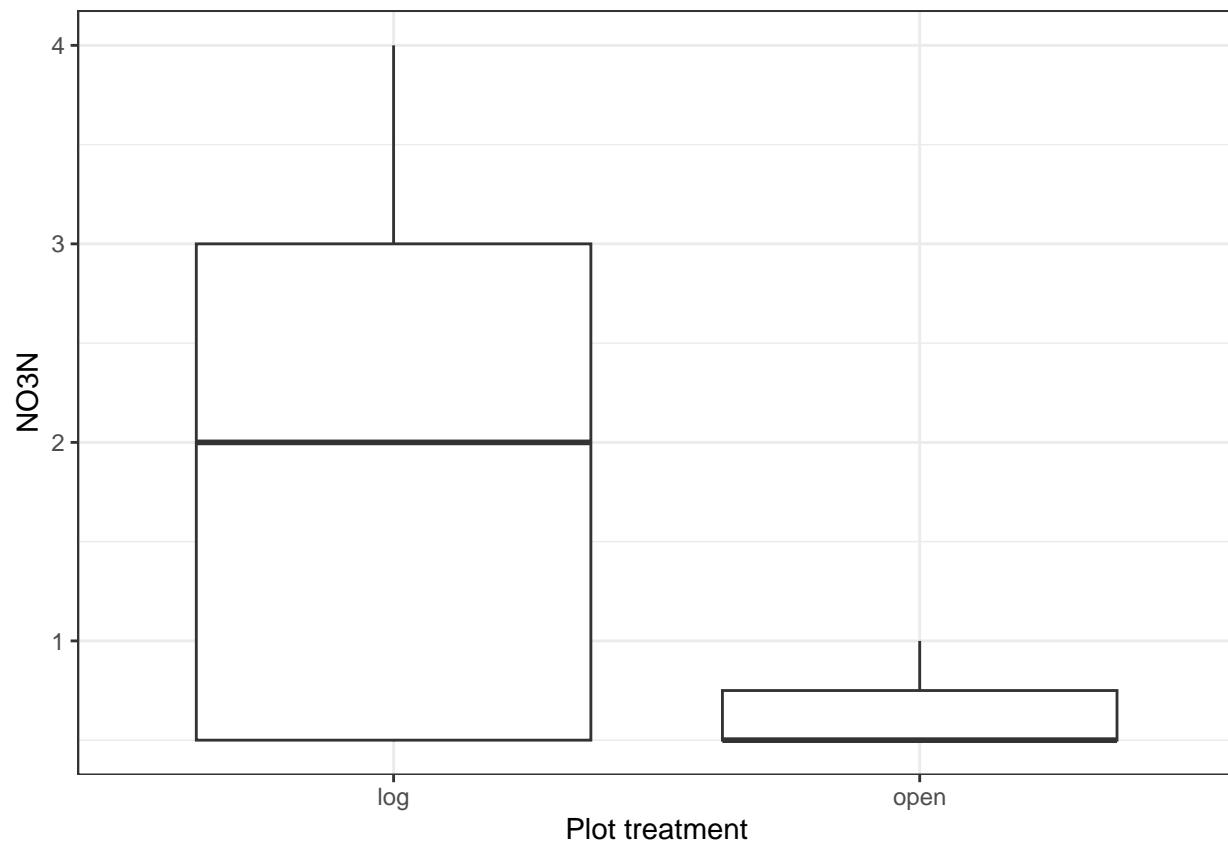
# Set up nutrient analysis data
nutrient <- read.csv("Nutrient.csv", header = T)
nutrient <- separate(nutrient, 2, c("block", "plot"), "_")
nutrient <- nutrient[, 2:17]

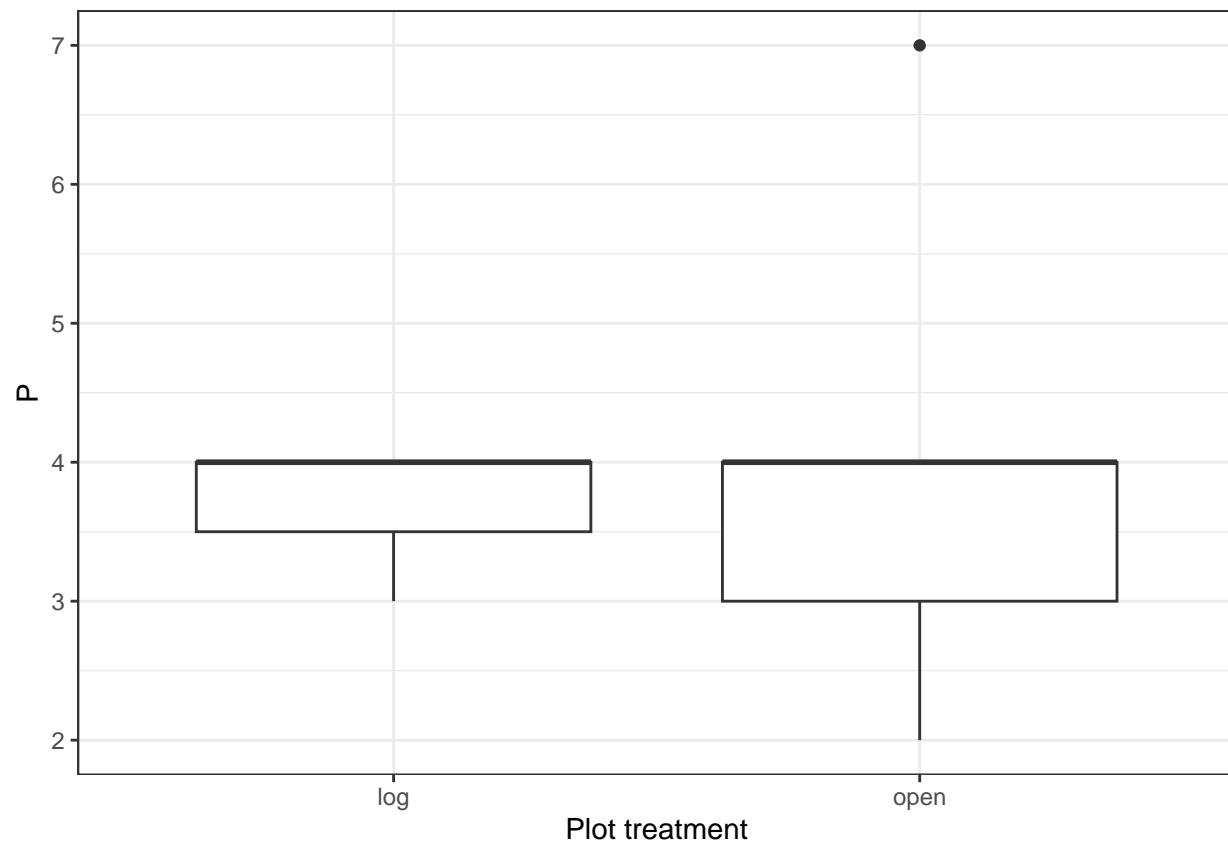
# Visualise data
Nu <- c("NH4N", "NO3N", "P", "K", "C", "conductivity", "pH", "prewash.exch.ca", "prewash.exch.k", "prew"

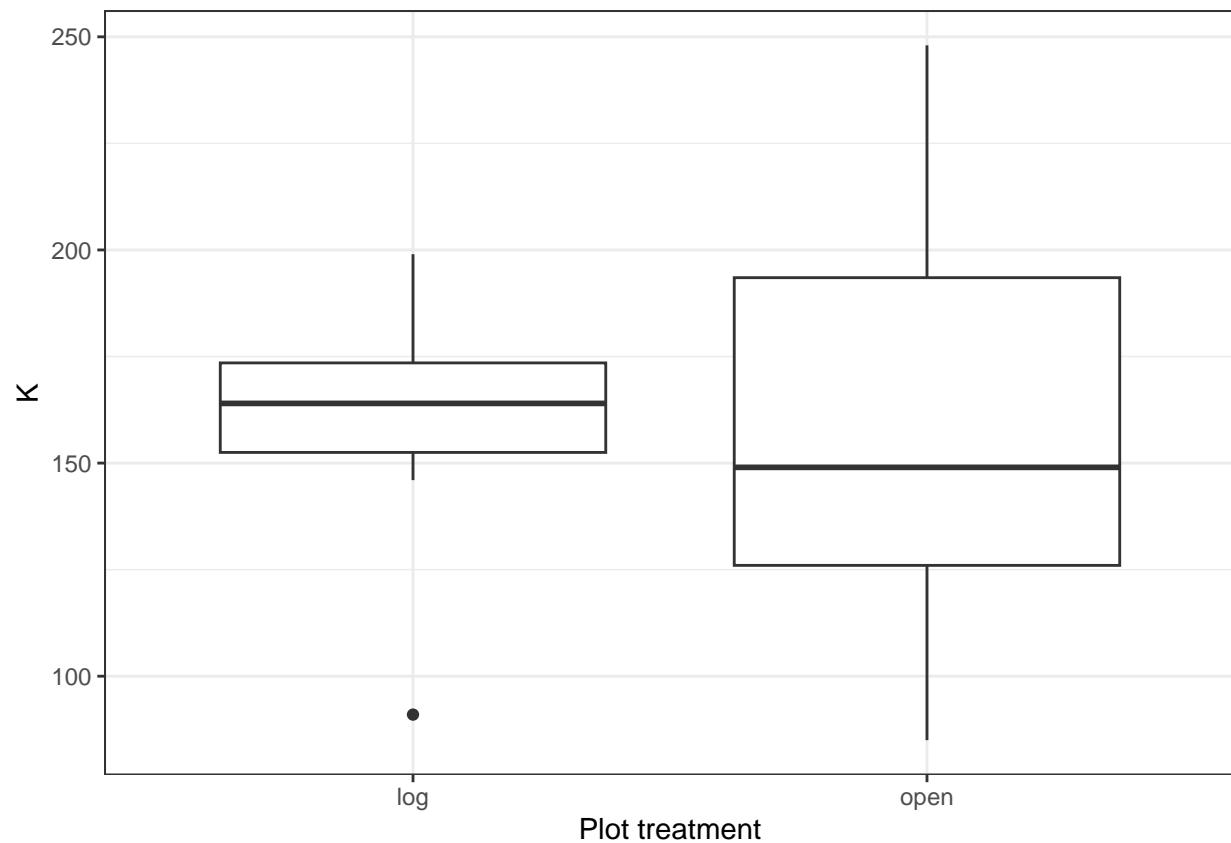
for (element in Nu) {
  print(
    ggplot(nutrient, aes(x = init, y = !!rlang::sym(element))) +
    geom_boxplot() +
    theme_bw() +
    labs(x = "Plot treatment", y = element)
  )
}

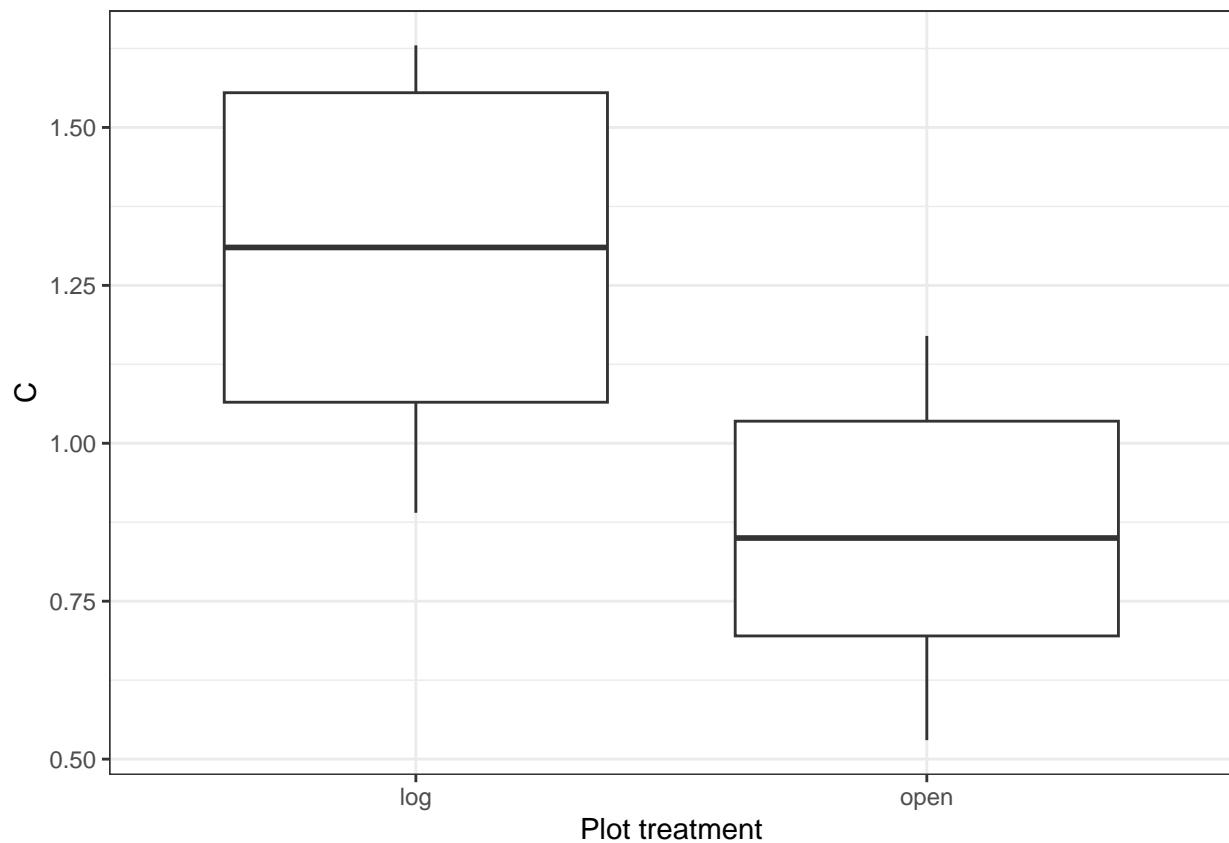
```

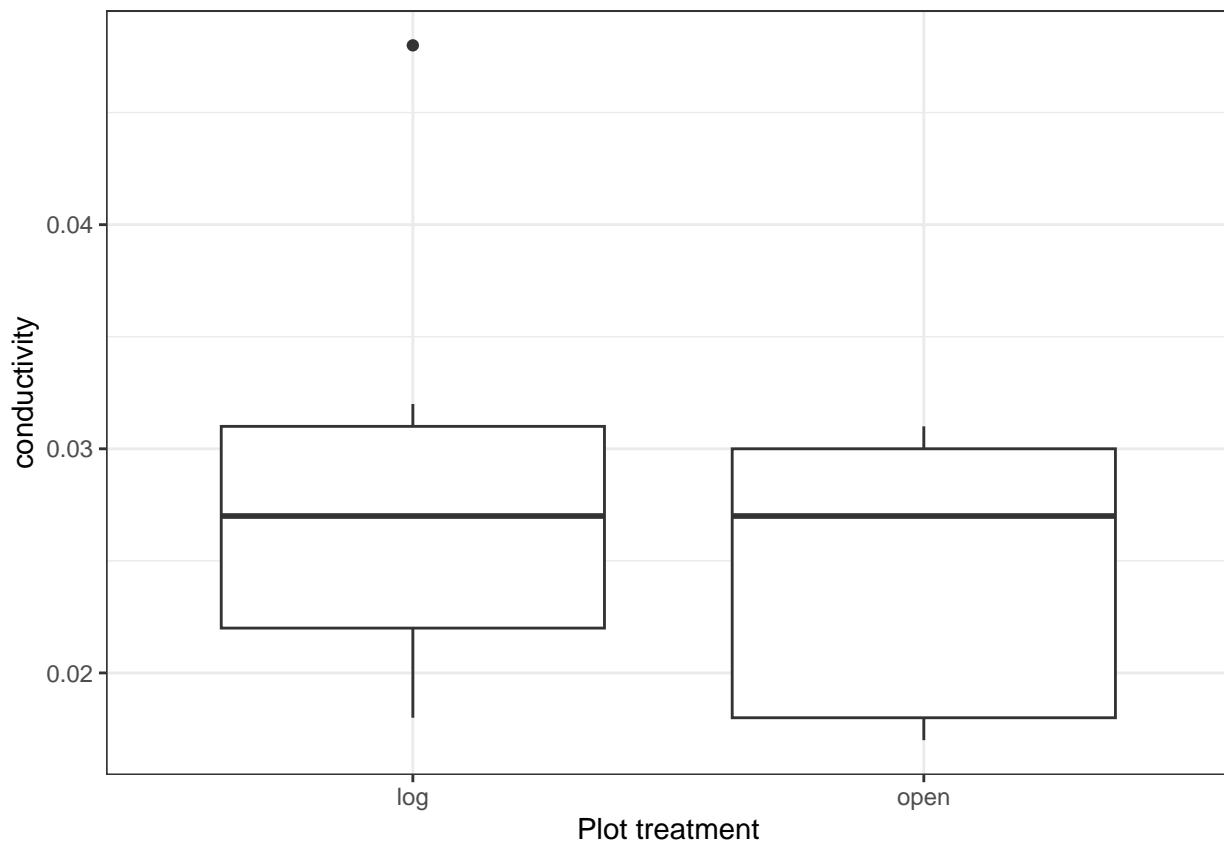


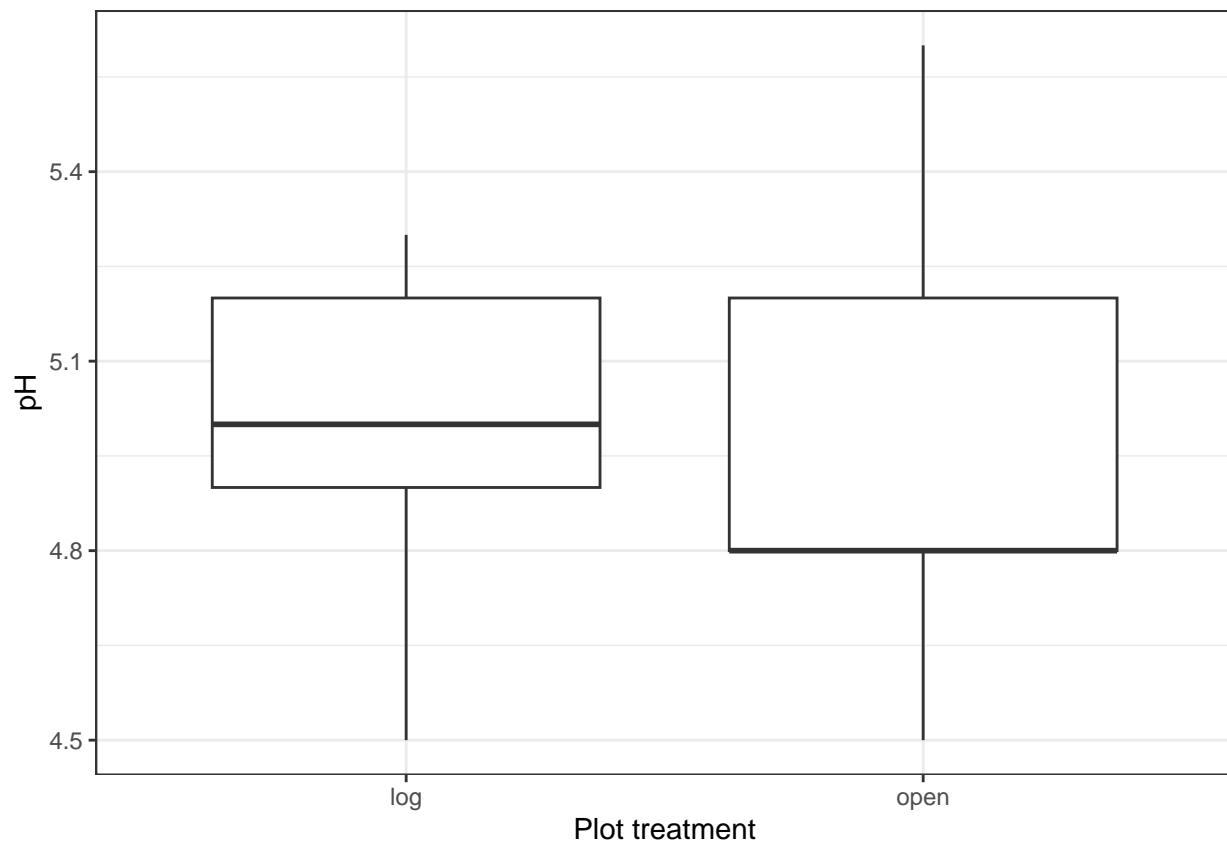


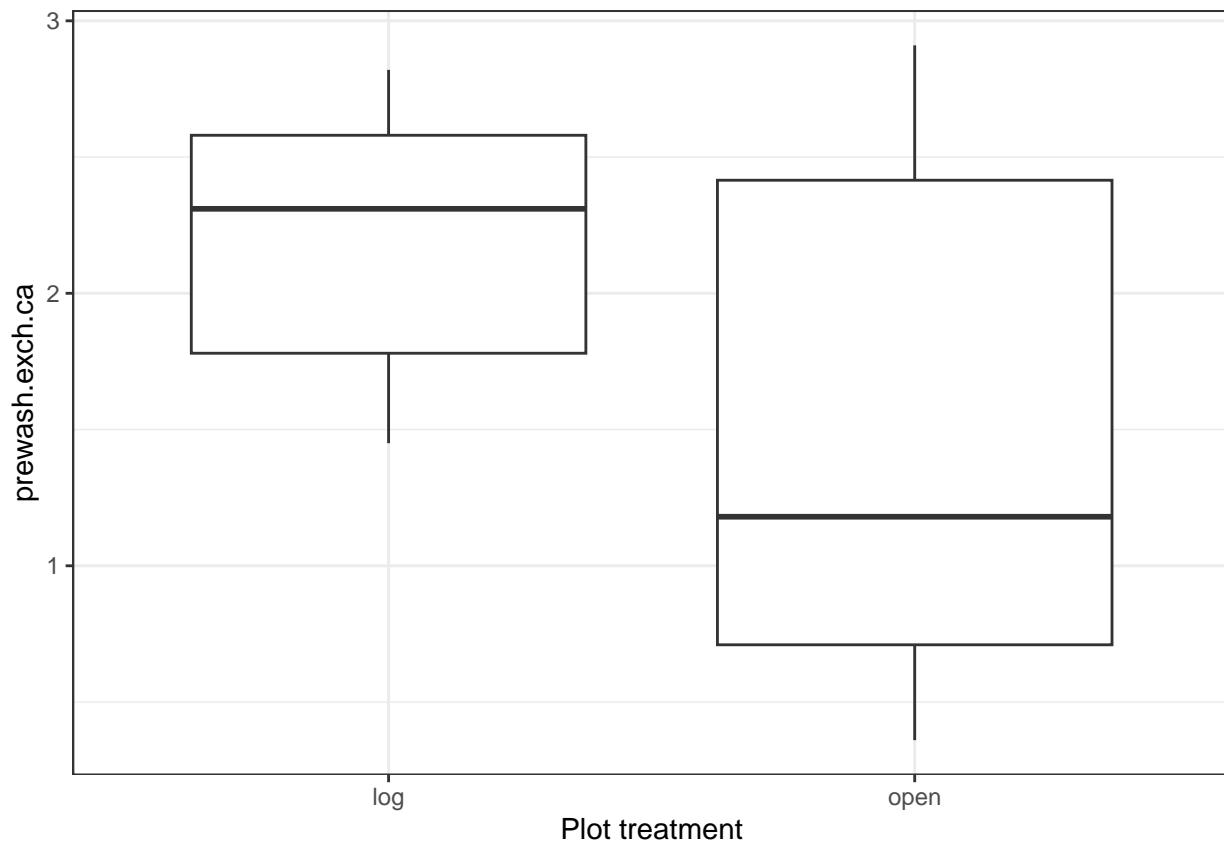


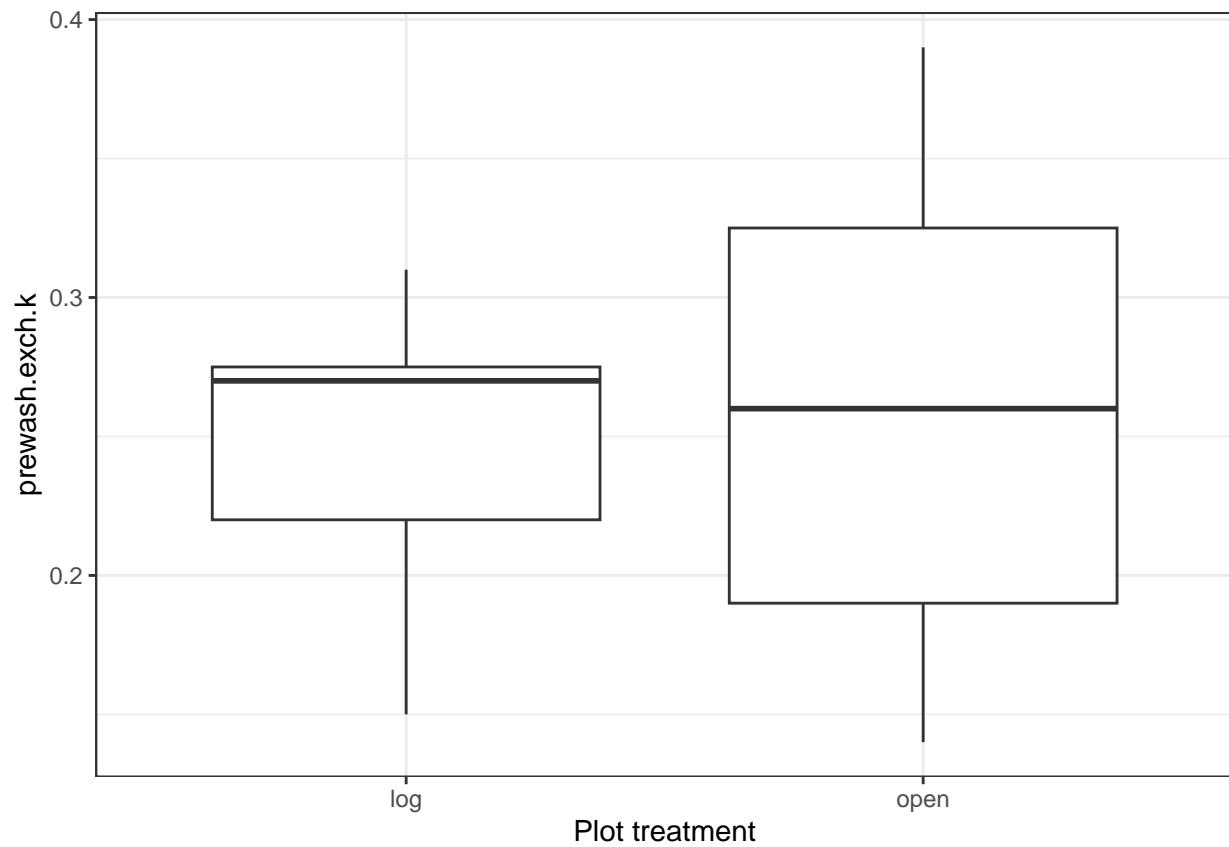


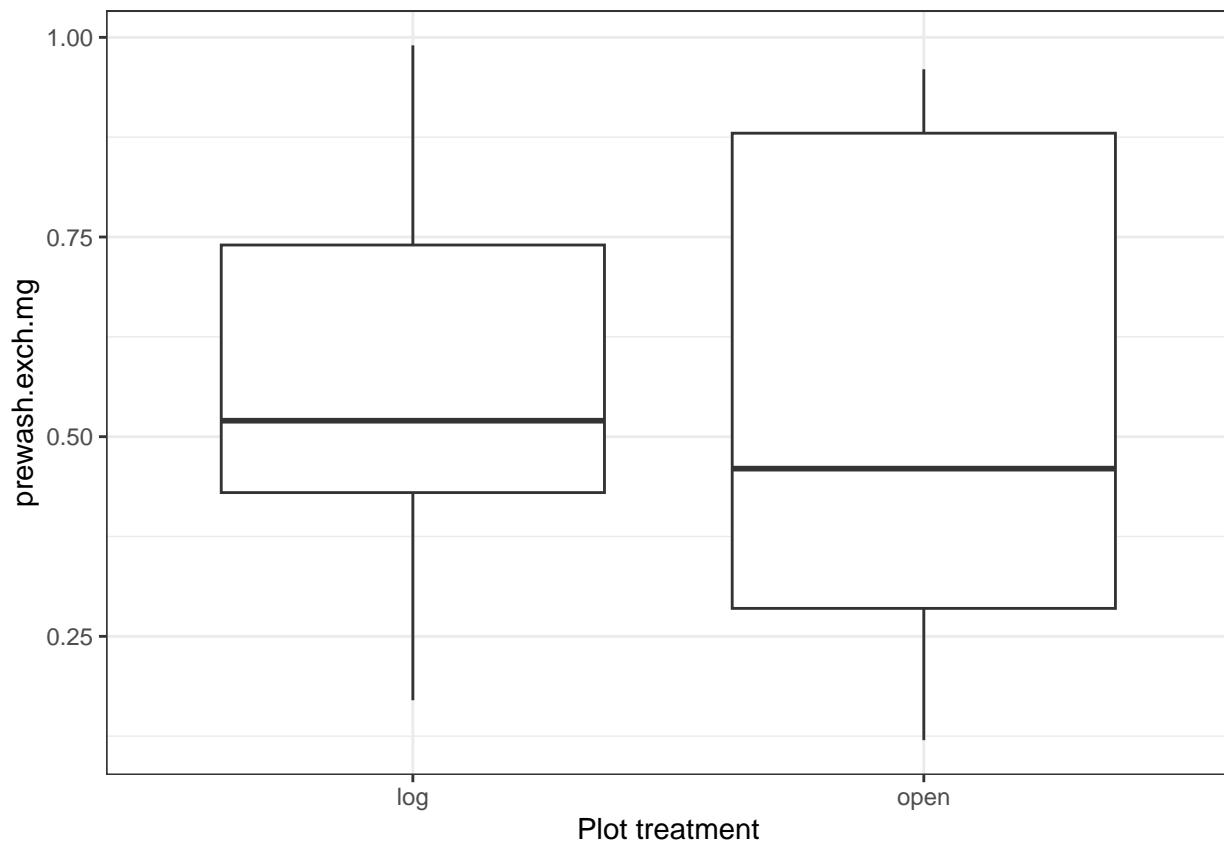


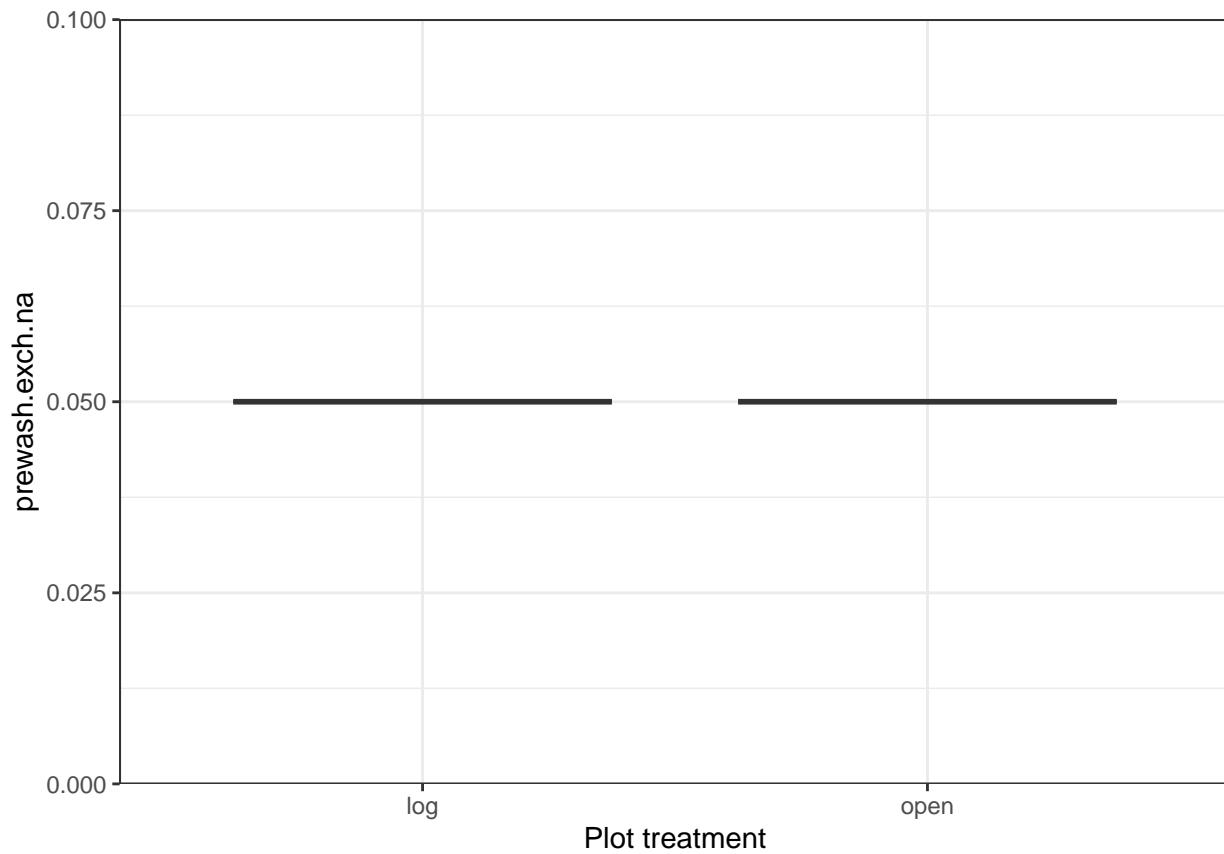


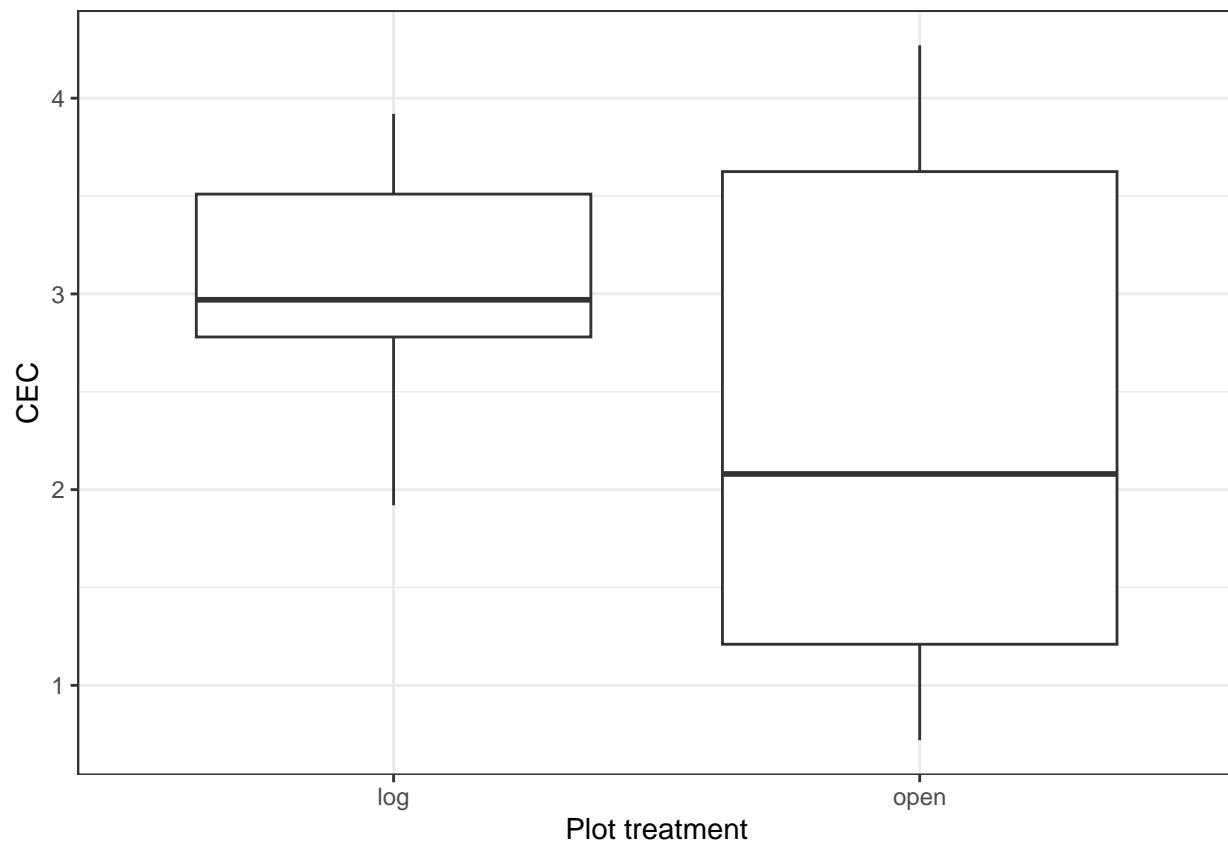




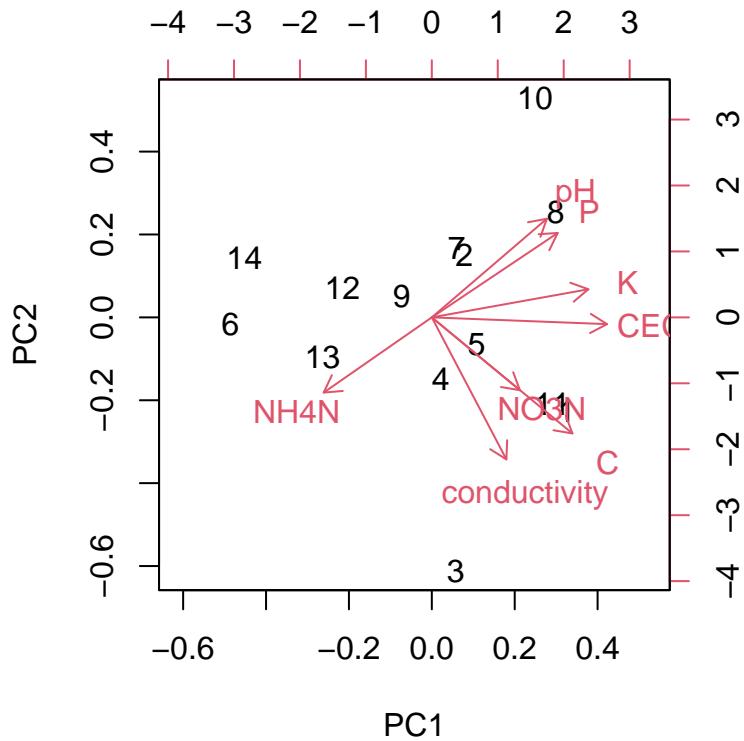








```
nutrient.for.pca <- nutrient[,c(5:11,16)]
nutrient.pca <- prcomp(nutrient.for.pca, center=TRUE, scale=TRUE)
biplot(nutrient.pca)
```



Nutrient composition comparison between log and open

- Organic carbon in soil is significantly higher in fallen log patches.

```

library(ggplot2)
library(ggpubr)
library(dplyr)
library(lme4)
library(emmeans)
library(pscl)
library(glmmTMB)
library(tidyr)
library(DHARMa)
library(ggplot2)
library(AICmodavg)
library(ggpubr)

# Set up nutrient analysis data
nutrient <- read.csv("Nutrient.csv", header = T)
nutrient <- separate(nutrient, 2, c("block", "plot"), "_")
nutrient <- nutrient[, 2:17]

Nu <- c("NH4N", "NO3N", "P", "K", "C", "pH", "CEC")

wilcox_results <- data.frame(element = character(),
                               w = numeric(),

```

```

        p_value = numeric(),
        stringsAsFactors = FALSE)

# Perform Wilcoxon rank sum test for each element between treatments
nutrient$init <- as.factor(nutrient$init)
for (element in Nu) {
  result <- wilcox.test(get(element) ~ init, data = nutrient, exact=TRUE)
  wilcox_results <- rbind(wilcox_results,
                           data.frame(element = element,
                                      w = result$statistic,
                                      p_value = result$p.value,
                                      stringsAsFactors = FALSE,
                                      row.names = NULL))
}
print(wilcox_results) # only organic carbon is significantly different between the two plot types

plots_list <- list()

for (element in c(Nu, "C")) {
  if (element == "C") {
    plot <- ggplot(nutrient, aes(x = init, y = C)) +
      geom_boxplot() +
      theme_bw() +
      labs(x = NULL, y = "C") +
      scale_x_discrete(labels=c('log', 'open')) +
      geom_signif(comparisons = list(c("log", "open")),
                  map_signif_level = TRUE,
                  textsize = 5, vjust = 0.5, y_position = 1.6, tip_length = 0) +
      scale_y_continuous(expand = c(0,0.1))+
      theme(
        plot.background = element_blank(),
        panel.grid.major = element_blank(),
        panel.grid.minor = element_blank(),
        panel.border = element_blank() +
          theme(axis.line = element_line(color = 'black'))
      )
  } else {
    plot <- ggplot(nutrient, aes(x = init, y = !!rlang::sym(element))) +
      geom_boxplot() +
      theme_bw() +
      labs(x = NULL, y = element) +
      scale_x_discrete(labels=c('log', 'open')) +
      theme(
        plot.background = element_blank(),
        panel.grid.major = element_blank(),
        panel.grid.minor = element_blank(),
        panel.border = element_blank() + theme(axis.line = element_line(color = 'black'))
      )
  }
  plots_list[[element]] <- plot
}

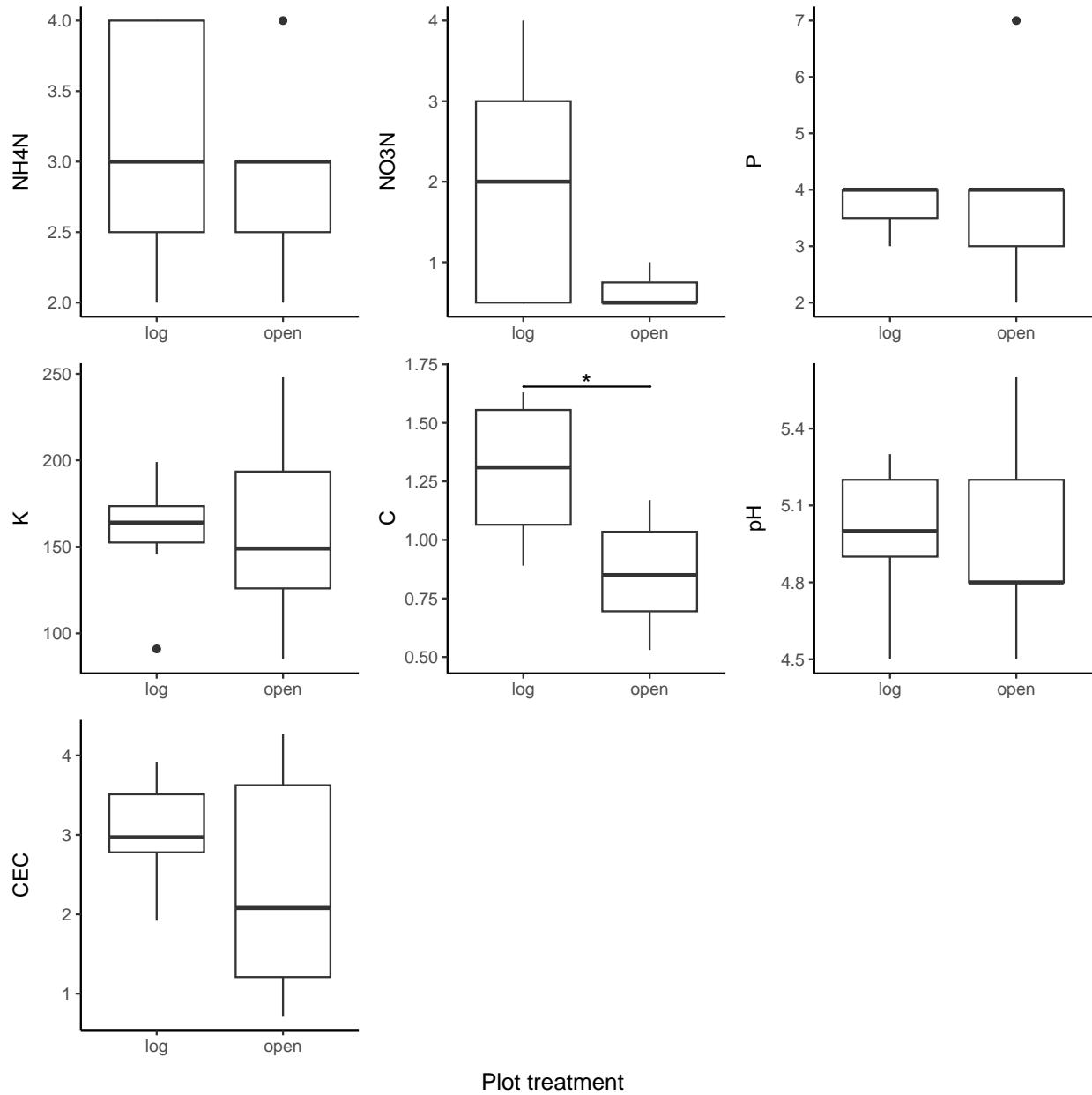

```

```

all_bp <- ggarrange(plotlist=plots_list, ncol=3, nrow=3, common.legend =TRUE, legend="bottom", align =
"center")

all_bp <- annotate_figure(all_bp, bottom = "Plot treatment")
# Show the final plot
print(all_bp)

```



```

##   element     w    p_value
## 1   NH4N 29.5 0.53971328
## 2   NO3N 35.5 0.13627611
## 3     P 26.5 0.82963146
## 4     K 26.0 0.90151515
## 5     C 42.5 0.02518656
## 6   pH 29.5 0.56096176

```

```
## 7      CEC 30.0 0.53496503
```

```
library(ggplot2)
library(ggpubr)
library(dplyr)
library(lme4)
library(emmeans)
library(pscl)
library(glmmTMB)
library(tidyr)
library(DHARMa)
library(ggplot2)
library(AICcmodavg)
library(ggpubr)
require(vegan)
require(labdsrv)
require(stringr)
require(ggrepel)

# Set up nutrient analysis data
nutrient <- read.csv("Nutrient.csv", header = T)
nutrient <- separate(nutrient, 2 , c("block","plot"), "_")
nutrient <- nutrient[,2:17]

# This community data set includes all species (including unidentified species)
comm <- read.csv("20-22_species_composition_data_w_unk.csv", header=T)

# Remove the locations surveyed in 2021 and 2022 that were not surveyed in 2020 - aka, cm=0, cm=21, cm=22
comm<-comm[which(comm$cm_location!=21 & comm$cm_location!=0 & comm$cm_location!=22 & comm$cm_location!=23),]

# make a group name for each row
comm$grp<-apply(comm[c(1,3,5,6,7)], 1, paste, collapse=":")
# timepoint, block, transect_name, initial

# Need to make each row a community using matrify.
commsub<-comm[,c(15,10,13)] # group, species_code, and count of each species for each transect. transect
commsub <-as.data.frame(commsub)
commtry<-matrify(commsub) # make it an expanded species matrix
commtry$x[which(is.na(commtry$x))] <- 1 # "x" means that there were no individuals in the transect, but
# ncol(commtry) # how many species are we working with in our community matrix

# Store grouping row names as a column, then remove rownames.
commtry$grps<-rownames(commtry)
rownames(commtry)<-NULL
# names(commtry)

# Split group info into columns for each variable.
mat<-separate(commtry, 107, c("time","block","transect","init", "treatment"), ":")
# names(mat) #check

# Add groupname using time, block, init columns.
mat$grp1<-apply(mat[c(107:110)], 1, paste, collapse=":")
```

```

mat$grp2<-apply(mat[c(107:111)], 1, paste, collapse=":")
# names(mat) #check

# Another df where the grouping variables are time, block, transect and initial state.
# Each row is a transect in a certain year.
df<-mat[,c(1:106,113)]
# df2 = with treatment in the grouping
df2 = df %>% mutate(across(.cols=1:106,.fns=as.numeric))
rownames(df2)<-NULL # remove rownames

##### Abundance analysis for 2020 (t0) AND in-situ log and in-situ open plots in 2021 (t1) and 2022 (t2)

# Sum observations across initial X transect X time X block X treatment (group variable).
# This gives number of plants in each row observation (transect level).
blocksum2<-rowsum(df2[,c(1:106)], group=df2$grp2)
blocksum2$grps<-rownames(blocksum2)
rownames(blocksum2)<-NULL # remove rownames

# Add in group vars.
nublock2<-separate(blocksum2, 107, c("time","block","transect","init", "treatment"), ":")
nublock2$total<-rowSums(nublock2[,c(1:106)])
nublock2$presence<-ifelse(nublock2$total > 0, 1, 0)

# Subset data where before treatments installed (t0), in-situ log and in-situ open from t1 and t2 are included
# Hence only absolute log effect and absolute open effect are concerned
dat_t0_insitu<-nublock2[which(nublock2$time=="t0" | nublock2$treatment=="open" | nublock2$treatment=="log")]

# look at plant abundance in log vs open
# look at range of data - what family should i use?
range(dat_t0_insitu$total)
nutrient_join <- nutrient[,c(1,3, 5:16)]
dat_t0_insitu <- inner_join(dat_t0_insitu, nutrient_join, by = c("init", "block"))

# since samples from t1 and t2 from are mainly from the same plots as t0. These are not independent replicates
# including year and block as random effects
# (1/year) + (1/block) as we think the effect of init on total is the same (slope) but the intercept different
# but since we have a greater number of sample from t0, data from t0 will contribute more to the variance
# see https://bookdown.org/steve_midway/DAR/random-effects.html#pld-example

# 2020 - 2022 #
Nu <- c("NH4N", "NO3N", "P", "K", "C", "pH", "CEC")

result_t012 <- data.frame(Element = character(), P_Value = numeric(), stringsAsFactors = FALSE)

for (element in Nu) {
  formula <- as.formula(paste("total ~", element, "+ (1|block) + (1|time)", sep = ""))
  
  abun.mod <- glmer(formula, data = dat_t0_insitu, family = poisson)
  
  p.value <- coef(summary(abun.mod))[, "Pr(>|z|)"] [2]
  
  result_t012 <- rbind(result_t012, data.frame(Element = element, P_Value = p.value, stringsAsFactors = TRUE))
  rownames(result_t012) <- NULL
}

```

```

}

print(result_t012)

mod.t012 <- glmer(total ~ NH4N + NO3N + P + K + C + pH + CEC + (1|block) + (1|time), data = dat_t0_insitu)

Abundance ~ nutrient composition

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

summary(mod.t012)

# only 2022 #
result_2022 <- data.frame(Element = character(), P_Value = numeric(), stringsAsFactors = FALSE)

for (element in Nu) {
  formula <- as.formula(paste("total ~", element, "+ (1|block)", sep = ""))
  abun.mod <- glmer(formula, data = dat_t0_insitu, family = poisson, subset = (time == "t2"))
  model_summary <- summary(abun.mod)

  p_value <- coef(summary(abun.mod))[, "Pr(>|z|)"][2]

  result_2022 <- rbind(result_2022, data.frame(Element = element, P_Value = p_value, stringsAsFactors = TRUE))
  rownames(result_2022) <- NULL
}

# Print or use the result dataframe
print(result_2022)

mod.t2 <- glmer(total ~ NH4N + NO3N + P + K + C + pH + CEC + (1|block), data = dat_t0_insitu, subset = (time == "t1"))

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

summary(mod.t2)

# only 2021 #
result_2021 <- data.frame(Element = character(), P_Value = numeric(), stringsAsFactors = FALSE)

for (element in Nu) {
  formula <- as.formula(paste("total ~", element, "+ (1|block)", sep = ""))
  abun.mod <- glmer(formula, data = dat_t0_insitu, family = poisson, subset = (time == "t1"))
  model_summary <- summary(abun.mod)

  p_value <- coef(summary(abun.mod))[, "Pr(>|z|)"][2]

  result_2021 <- rbind(result_2021, data.frame(Element = element, P_Value = p_value, stringsAsFactors = TRUE))
  rownames(result_2021) <- NULL
}

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

```

```

# Print or use the result dataframe
print(result_2021)

mod.t1 <- glmer(total ~ NH4N + NO3N + P + K + C + pH + CEC + (1|block), data = dat_t0_in situ, subset = (t

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

summary(mod.t1)

# only 2020 #
result_2020 <- data.frame(Element = character(), P_Value = numeric(), stringsAsFactors = FALSE)

for (element in Nu) {
  formula <- as.formula(paste("total ~", element, "+ (1|block)", sep = ""))
  abun.mod <- glmer(formula, data = dat_t0_in situ, family = poisson, subset = (time == "t1"))
  model_summary <- summary(abun.mod)

  p_value <- coef(summary(abun.mod))[, "Pr(>|z|)"] [2]

  result_2020 <- rbind(result_2020, data.frame(Element = element, P_Value = p_value, stringsAsFactors =
  rownames(result_2020) <- NULL
}

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

# Print or use the result dataframe
print(result_2020)

mod.t0 <- glmer(total ~ NH4N + NO3N + P + K + C + pH + CEC + (1|block), data = dat_t0_in situ, subset = (t

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

summary(mod.t0)

## [1] 1 28
##   Element   P_Value
## 1    NH4N 0.0156207
## 2    NO3N 0.6574029
## 3      P 0.1027048
## 4      K 0.2699504
## 5      C 0.6858746
## 6      pH 0.2386072
## 7     CEC 0.3526452
## Generalized linear mixed model fit by maximum likelihood (Laplace
## Approximation) [glmerMod]
## Family: poisson  ( log )
## Formula: total ~ NH4N + NO3N + P + K + C + pH + CEC + (1 | block) + (1 |
## time)
## Data: dat_t0_in situ
##
##       AIC      BIC      logLik deviance df.resid

```

```

##   2071.8   2111.4  -1025.9   2051.8      381
##
## Scaled residuals:
##    Min     1Q Median     3Q    Max
## -2.7017 -0.8715 -0.2124  0.6574  6.2323
##
## Random effects:
## Groups Name        Variance Std.Dev.
## block  (Intercept) 0.0000  0.0000
## time   (Intercept) 0.0828  0.2877
## Number of obs: 391, groups: block, 7; time, 3
##
## Fixed effects:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) 1.4958369  0.6674769   2.241 0.025024 *
## NH4N       -0.1306295  0.0390351  -3.346 0.000819 ***
## NO3N       -0.0432208  0.0254347  -1.699 0.089266 .
## P          0.0268848  0.0245469   1.095 0.273411
## K          -0.0007813  0.0007929  -0.985 0.324445
## C          0.4456139  0.1532102   2.909 0.003632 **
## pH         0.1647972  0.1371901   1.201 0.229661
## CEC        -0.1513526  0.0572606  -2.643 0.008212 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
## (Intr) NH4N  NO3N   P      K      C      pH
## NH4N -0.045
## NO3N  0.249  0.364
## P      0.314 -0.041  0.104
## K      -0.303  0.563  0.279 -0.444
## C      -0.660 -0.367 -0.536 -0.105 -0.107
## pH     -0.926 -0.209 -0.349 -0.370  0.128  0.717
## CEC    0.789  0.253  0.274  0.159 -0.173 -0.837 -0.841
## optimizer (Nelder_Mead) convergence code: 0 (OK)
## boundary (singular) fit: see help('isSingular')
##
## Element      P_Value
## 1   NH4N 0.092376421
## 2   NO3N 0.171440737
## 3   P   0.034059164
## 4   K   0.494765549
## 5   C   0.358927829
## 6   pH  0.001456178
## 7   CEC 0.017201969
## Generalized linear mixed model fit by maximum likelihood (Laplace
## Approximation) [glmerMod]
## Family: poisson  ( log )
## Formula: total ~ NH4N + NO3N + P + K + C + pH + CEC + (1 | block)
## Data: dat_t0_in situ
## Subset: (time == "t2")
##
##      AIC      BIC  logLik deviance df.resid
## 534.1    556.0   -258.1    516.1      75

```

```

##
## Scaled residuals:
##      Min     1Q Median     3Q    Max
## -2.7616 -1.1027 -0.1047  0.7786  5.5003
##
## Random effects:
##   Groups Name        Variance Std.Dev.
##   block  (Intercept) 0         0
## Number of obs: 84, groups: block, 7
##
## Fixed effects:
##                  Estimate Std. Error z value Pr(>|z|)
## (Intercept) 3.926315  1.088211  3.608 0.000309 ***
## NH4N       -0.320672  0.073899 -4.339 1.43e-05 ***
## NO3N       -0.084596  0.048423 -1.747 0.080635 .
## P          -0.047312  0.046420 -1.019 0.308098
## K          -0.003076  0.001420 -2.166 0.030276 *
## C          0.467473  0.266278  1.756 0.079160 .
## pH         -0.028963  0.230542 -0.126 0.900025
## CEC        -0.135467  0.096268 -1.407 0.159372
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
## (Intr) NH4N  NO3N   P     K     C     pH
## NH4N -0.094
## NO3N  0.157  0.315
## P      0.266 -0.043  0.027
## K      -0.320  0.598  0.293 -0.455
## C      -0.610 -0.343 -0.480 -0.044 -0.123
## pH     -0.945 -0.197 -0.245 -0.322  0.112  0.641
## CEC    0.787  0.253  0.201  0.115 -0.144 -0.823 -0.817
## optimizer (Nelder_Mead) convergence code: 0 (OK)
## boundary (singular) fit: see help('isSingular')
##
## Element P_Value
## 1 NH4N 0.11143644
## 2 NO3N 0.38393083
## 3 P    0.02307903
## 4 K    0.60866099
## 5 C    0.36904584
## 6 pH   0.21321806
## 7 CEC  0.78217717
## Generalized linear mixed model fit by maximum likelihood (Laplace
## Approximation) [glmerMod]
## Family: poisson ( log )
## Formula: total ~ NH4N + NO3N + P + K + C + pH + CEC + (1 | block)
## Data: dat_t0_insitu
## Subset: (time == "t1")
##
##      AIC      BIC logLik deviance df.resid
## 486.7    508.7   -234.3    468.7      77
##
## Scaled residuals:
```

```

##      Min     1Q Median     3Q    Max
## -2.4482 -0.8451 -0.0827  0.6907  4.1828
##
## Random effects:
## Groups Name        Variance Std.Dev.
## block (Intercept) 0         0
## Number of obs: 86, groups: block, 7
##
## Fixed effects:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) 1.126174  1.245205  0.904  0.36578
## NH4N       -0.213162  0.082162 -2.594  0.00947 **
## NO3N       -0.073874  0.053256 -1.387  0.16540
## P          0.095921  0.044745  2.144  0.03205 *
## K          -0.003016  0.001612 -1.871  0.06130 .
## C          0.405362  0.310359  1.306  0.19152
## pH         0.347599  0.264084  1.316  0.18809
## CEC       -0.165413  0.112475 -1.471  0.14138
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
## (Intr) NH4N  NO3N   P     K     C     pH
## NH4N -0.078
## NO3N  0.233  0.307
## P      0.357 -0.145  0.084
## K      -0.328  0.613  0.268 -0.477
## C      -0.635 -0.352 -0.539 -0.026 -0.134
## pH     -0.950 -0.197 -0.316 -0.374  0.118  0.664
## CEC    0.805  0.263  0.276  0.124 -0.142 -0.829 -0.834
## optimizer (Nelder_Mead) convergence code: 0 (OK)
## boundary (singular) fit: see help('isSingular')
##
## Element    P_Value
## 1 NH4N 0.11143644
## 2 NO3N 0.38393083
## 3 P 0.02307903
## 4 K 0.60866099
## 5 C 0.36904584
## 6 pH 0.21321806
## 7 CEC 0.78217717
## Generalized linear mixed model fit by maximum likelihood (Laplace
## Approximation) [glmerMod]
## Family: poisson ( log )
## Formula: total ~ NH4N + NO3N + P + K + C + pH + CEC + (1 | block)
## Data: dat_t0_insitu
## Subset: (time == "t0")
##
##      AIC      BIC logLik deviance df.resid
## 1027.9 1058.5 -505.0   1009.9      212
##
## Scaled residuals:
##      Min     1Q Median     3Q    Max
## -1.9394 -0.8480 -0.0811  0.6001  3.4686

```

```

## 
## Random effects:
## Groups Name           Variance Std.Dev.
## block  (Intercept) 2.721e-17 5.216e-09
## Number of obs: 221, groups: block, 7
##
## Fixed effects:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) -0.172111  1.064069 -0.162  0.8715
## NH4N         0.009015  0.056667  0.159  0.8736
## NO3N        -0.007833  0.037365 -0.210  0.8340
## P            0.034389  0.038947  0.883  0.3773
## K            0.001875  0.001226  1.529  0.1264
## C            0.487943  0.241820  2.018  0.0436 *
## pH           0.234206  0.225980  1.036  0.3000
## CEC          -0.164379  0.093525 -1.758  0.0788 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##   (Intr) NH4N  NO3N   P     K     C    pH
## NH4N  0.003
## NO3N  0.340  0.424
## P     0.336  0.033  0.172
## K     -0.299  0.504  0.265 -0.414
## C     -0.748 -0.385 -0.578 -0.189 -0.075
## pH    -0.963 -0.230 -0.434 -0.392  0.140  0.784
## CEC   0.833  0.246  0.321  0.210 -0.212 -0.848 -0.856
## optimizer (Nelder_Mead) convergence code: 0 (OK)
## boundary (singular) fit: see help('isSingular')

```

Diversity ~ nutrient composition

- P seems to consistently affect the presence of non-zero diversity index.
- C links to a higher Shannon diversity index in plant communities in 2022 only.

```

require(vegan)
require(dplyr)
require(tidyr)
require(labdsrv)
require(stringr)
require(ggplot2)
require(ggrepel)
require(lme4)
require(emmeans)
require(lmerTest)
require(performance)
require(ggpubr)
require(DHARMA)

# Set up nutrient data
nutrient <- read.csv("Nutrient.csv", header = T)
nutrient <- separate(nutrient, 2 , c("block", "plot"), "_")

```

```

nutrient <- nutrient[, 2:17]

# This dataset does not include unknown data.
comm <- read.csv("20-22_species_composition_data_no_unk.csv", header = T)

# Remove the locations surveyed in 2021 and 2022 that were not surveyed in 2020 - aka, cm=0, cm=21, cm
comm<-comm[which(comm$cm_location!=21 & comm$cm_location!=0 & comm$cm_location!=22 & comm$cm_location!=23),]

# make a group name for each row
comm$grp<-apply(comm[c(1,3,5,6,7)], 1, paste, collapse=":") # timepoint, block, transect_name, initial

# Need to make each row a community using matrify
commsub<-comm[,c(15,10,13)]

# group, species_code, and count of each species for each transect. transects are rows.
commsub <-as.data.frame(commsub)
commtry<-labdsv::matrify(commsub) # make it an expanded species matrix
commtry$x[which(is.na(commtry$x))] <- 1 # "x" means that there were no individuals in the transect, but we still want to count them
ncol(commtry) # how many species are we working with in our community matrix

## [1] 87

# Store grouping row names as a column, then remove rownames.
commtry$grps<-rownames(commtry)
rownames(commtry)<-NULL
names(commtry)

## [1] "acul"   "aicu"   "arca"   "ardy"   "arsp"   "auel"   "bldr"   "blrd"   "brdi"
## [10] "brdr"   "brpe"   "brru"   "buse"   "caer"   "cagr"   "cahi"   "casp"   "cear"
## [19] "chau"   "chei"   "chps"   "crcl"   "crc"    "cusc"   "cusp"   "dagl"   "dosp"
## [28] "ento"   "erau"   "ercy"   "erra"   "ersp"   "gite"   "gn"    "gobe"   "gocy"
## [37] "gono"   "goro"   "gosp"   "haod"   "hygl"   "hyp"    "hypo"   "jubu"   "laro"
## [46] "ledu"   "lele"   "loef"   "misp"   "mite"   "momo"   "mopa"   "niro"   "omco"
## [55] "orsp"   "pala"   "peai"   "pedu"   "phsu"   "plde"   "poar"   "poca"   "pocap"
## [64] "poce"   "pogn"   "pole"   "pomu"   "pter"   "ptga"   "ptob"   "rhl"    "rhp"
## [73] "rhsp"   "ry"     "scna"   "sino"   "sool"   "stfi"   "stpi"   "thma"   "trcy"
## [82] "tris"   "tror"   "trpi"   "waac"   "wagr"   "x"      "grps"

# Split group info into columns for each variable
mat<-separate(commtry, 88, c("time", "block", "transect", "init", "treatment"), ":")
names(mat) #check

## [1] "acul"      "aicu"      "arca"      "ardy"      "arsp"      "auel"      "bldr"      "blrd"      "brdi"
## [7] "bldr"      "blrd"      "brdi"      "brdr"      "brpe"      "brru"      "caer"      "capse"      "cear"
## [13] "buse"      "caer"      "cagr"      "cahi"      "casp"      "cear"      "cusc"      "cusp"      "dagl"
## [19] "chau"      "chei"      "chps"      "crcl"      "crc"       "cusc"      "cusp"      "dagl"       "dosp"
## [25] "cusp"      "dagl"      "dosp"      "ento"      "erau"      "ercy"      "gite"      "gn"        "gobe"
## [31] "erra"      "ersp"      "gite"      "gn"        "gobe"      "gocy"      "gn"        "gote"      "gocy"
## [37] "gono"      "goro"      "gosp"      "haod"      "hygl"      "hyp"       "hypo"      "jubu"      "laro"
## [43] "hypo"      "jubu"      "laro"      "ledu"      "lele"      "loef"      "momo"      "mopa"      "niro"
## [49] "misp"      "mite"      "momo"      "momo"      "mopa"      "niro"      "niro"      "omco"      "omco"
## [55] "orsp"      "pala"      "peai"      "pedu"      "phsu"      "plde"      "poar"      "poca"      "pocap"

```

```

## [61] "poar"      "poca"       "pocap"      "poce"       "pogn"       "pole"
## [67] "pomu"       "pter"        "ptga"       "ptob"       "rhla"       "rhpyp"
## [73] "rhsp"       "ry"          "scna"       "sino"       "sool"       "stfi"
## [79] "stpi"       "thma"       "trcy"       "tris"       "tror"       "trpi"
## [85] "waac"       "wagr"        "x"          "time"       "block"      "transect"
## [91] "init"       "treatment"

```

```

# Add groupname using time, block, init columns
mat$grp<-apply(mat[c(88:92)], 1, paste, collapse=":")
names(mat) #check

```

```

## [1] "acul"       "aicu"       "arca"       "ardy"       "arsp"       "ael"
## [7] "bldr"       "blrd"       "brdi"       "brdr"       "brpe"       "brru"
## [13] "buse"       "caer"       "cagr"       "cahi"       "casp"       "cear"
## [19] "chau"       "chei"       "chs"        "crcl"       "crcos"      "cusc"
## [25] "cusp"       "dagl"       "dosp"       "ento"       "erau"       "ercy"
## [31] "erra"       "ersp"       "gite"       "gnre"       "gobe"       "gocy"
## [37] "gono"       "goro"       "gosp"       "haod"       "hygl"       "hypi"
## [43] "hypo"       "jubu"       "laro"       "ledu"       "lele"       "loef"
## [49] "misp"       "mite"       "momo"       "mopa"       "niro"       "omco"
## [55] "orsp"       "pala"       "peai"       "pedu"       "phsu"       "plde"
## [61] "poar"       "poca"       "pocap"      "poce"       "pogn"       "pole"
## [67] "pomu"       "pter"        "ptga"       "ptob"       "rhla"       "rhpyp"
## [73] "rhsp"       "ry"          "scna"       "sino"       "sool"       "stfi"
## [79] "stpi"       "thma"       "trcy"       "tris"       "tror"       "trpi"
## [85] "waac"       "wagr"        "x"          "time"       "block"      "transect"
## [91] "init"       "treatment" "grp"

```

```

# Another df where the grouping variables are time, block, transect and initial state.
# Each row is a transect in a certain year.
df<-mat[,c(1:87,93)]
df2 = df %>% mutate(across(.cols=1:87,.fns=as.numeric)) # make everything numeric
rownames(df2)<-NULL # remove rownames

##### Species diversity analysis for 2020 - 2022 data (Shannon diversity on transect level)
numat = mat %>% mutate(across(.cols=1:87,.fns=as.numeric)) # make everything numeric

# sum species for each group (grouped by init, transect, block, time)
nudat<-numat%%
  group_by(time, block, transect, init, treatment) %>% summarise(across(where(is.numeric), sum))

# make a data frame
dat<-as.data.frame(nudat) # this df contains transect levels from all years
dat<-dat[which(dat$time=="t0" | dat$treatment=="open" | dat$treatment=="insitu_log"),]

#estimate diversity for each row/group. don't include 'x'

# no groups, just estimate diversity of each row
est<-dat[,c(6:91)]
dat$diversity<-diversity(est, index='shannon')
nutrient_join <- nutrient[,c(1,3, 5:16)]
dat <- inner_join(dat, nutrient_join, by = c("init", "block"))

```

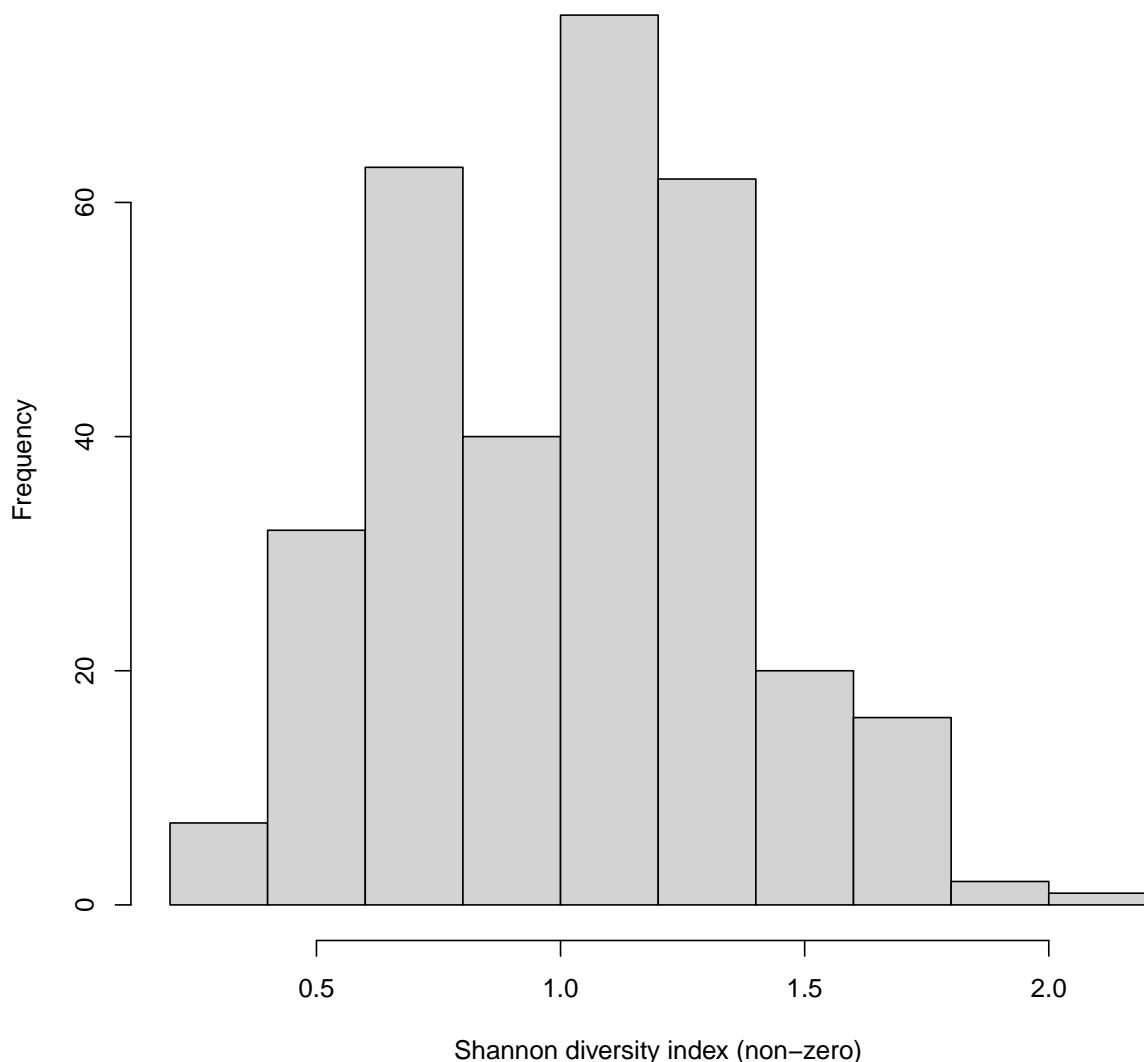
```

##### We used a hurdle model since the data is zero-inflated
# separate into zero and non-zero observations
dat$non_zero <- ifelse(dat$diversity > 0, 1, 0)

# Non-zero data is an approximal Gaussian distribution
hist(dat$diversity[dat$non_zero ==1], xlab = "Shannon diversity index (non-zero)", main = "Histogram of"

```

Histogram of Shannon diversity index (non-zero, all years, transect level)

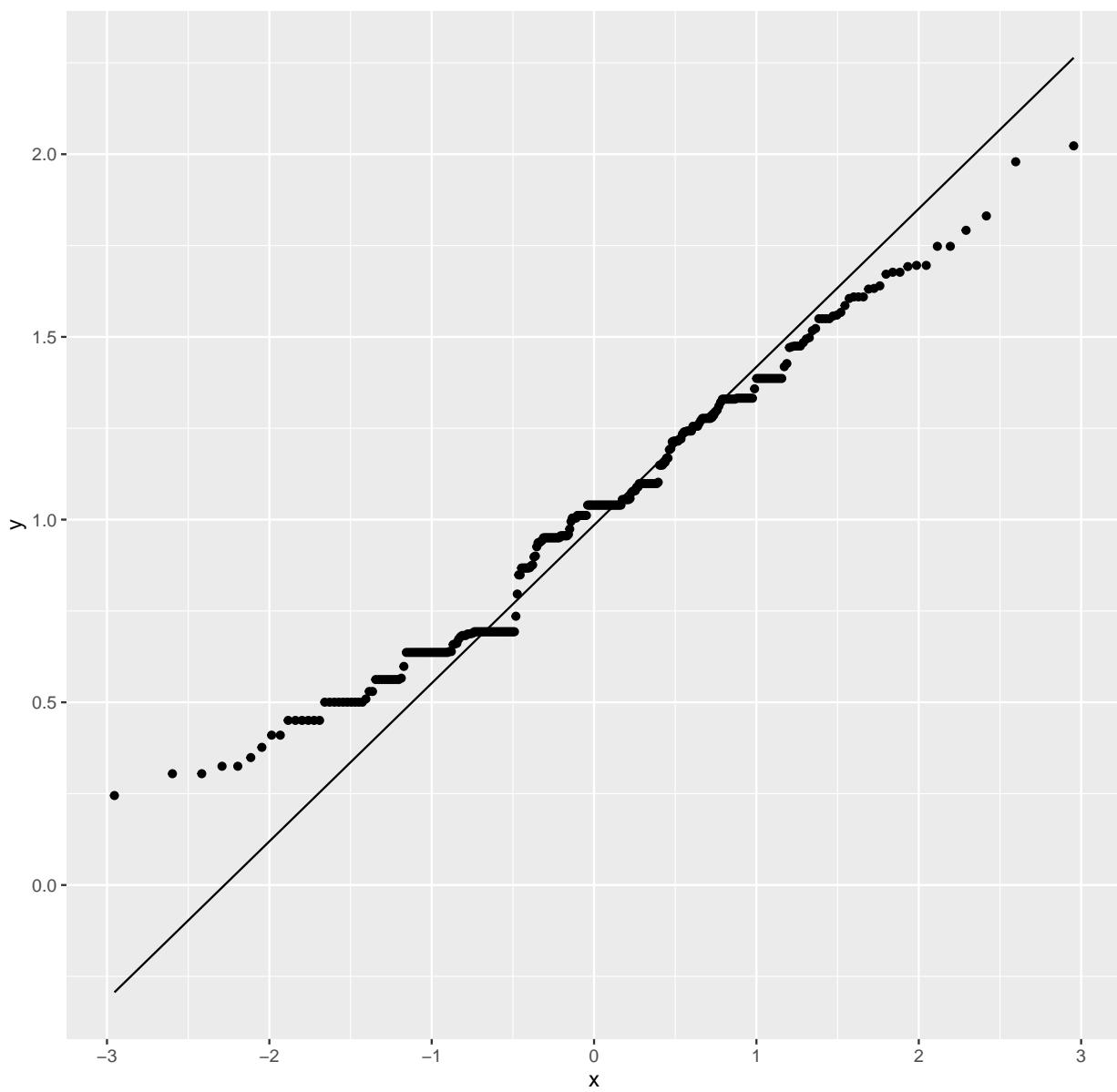


```

# Gaussian distribution produced the best-fitting ggplot
dat_nonzerodat <- dat[dat$non_zero == 1, ]
ggplot(dat_nonzerodat, aes(sample = diversity)) +
  geom_qq(distribution = qnorm) +
  geom_qq_line(distribution = qnorm) +
  ggtitle("normal")

```

normal



```

# Hurdle model part 1. Logistic regression to predict the probability of non-zero.
# Ppen and log environment does determine the probability of non-zero diversity index
# 0 Shannon diversity index =/= zero plants

# 2020-2022 #
# model occurrence of zeros
hurd.mod.1.t012 <- glmer(non_zero ~ NH4N + NO3N + P + K + C + pH + CEC + (1|block) + (1|time), data = da
summary(hurd.mod.1.t012)

## Generalized linear mixed model fit by maximum likelihood (Laplace
## Approximation) [glmerMod]
## Family: binomial ( logit )
## Formula: non_zero ~ NH4N + NO3N + P + K + C + pH + CEC + (1 | block) +

```

```

##      (1 | time)
## Data: dat
##
##      AIC      BIC  logLik deviance df.resid
##    371.4    411.2   -175.7     351.4      383
##
## Scaled residuals:
##      Min    1Q Median    3Q   Max
## -4.1862  0.2389  0.3600  0.5172  1.0318
##
## Random effects:
## Groups Name        Variance Std.Dev.
## block  (Intercept) 0.0000   0.0000
## time   (Intercept) 0.4026   0.6345
## Number of obs: 393, groups: block, 7; time, 3
##
## Fixed effects:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -0.787803  4.333907 -0.182  0.8558
## NH4N         0.222090  0.247133  0.899  0.3688
## NO3N         0.405693  0.185896  2.182  0.0291 *
## P            -0.360533  0.154382 -2.335  0.0195 *
## K            0.012017  0.006022  1.996  0.0460 *
## C            0.545003  0.984345  0.554  0.5798
## pH           0.271936  0.913446  0.298  0.7659
## CEC          -0.329449  0.372631 -0.884  0.3766
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
## (Intr) NH4N  NO3N   P      K      C      pH
## NH4N -0.101
## NO3N  0.253  0.279
## P      0.416 -0.080  0.132
## K      -0.320  0.519  0.241 -0.497
## C      -0.678 -0.283 -0.473 -0.136 -0.081
## pH     -0.954 -0.147 -0.340 -0.445  0.142  0.703
## CEC    0.824  0.162  0.241  0.262 -0.255 -0.816 -0.839
## optimizer (Nelder_Mead) convergence code: 0 (OK)
## boundary (singular) fit: see help('isSingular')

# model diversity
hurd.mod.2.t012 <- lmer(diversity ~ NH4N + NO3N + P + K + C + pH + CEC + (1|block) + (1|time), data = dat)
summary(hurd.mod.1.t012)

## Generalized linear mixed model fit by maximum likelihood (Laplace
## Approximation) [glmerMod]
## Family: binomial ( logit )
## Formula: non_zero ~ NH4N + NO3N + P + K + C + pH + CEC + (1 | block) +
##          (1 | time)
## Data: dat
##
##      AIC      BIC  logLik deviance df.resid
##    371.4    411.2   -175.7     351.4      383

```

```

## Scaled residuals:
##      Min      1Q Median      3Q     Max
## -4.1862  0.2389  0.3600  0.5172  1.0318
##
## Random effects:
##   Groups Name      Variance Std.Dev.
##   block  (Intercept) 0.0000  0.0000
##   time   (Intercept) 0.4026  0.6345
## Number of obs: 393, groups: block, 7; time, 3
##
## Fixed effects:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -0.787803  4.333907 -0.182  0.8558
## NH4N         0.222090  0.247133  0.899  0.3688
## NO3N         0.405693  0.185896  2.182  0.0291 *
## P            -0.360533  0.154382 -2.335  0.0195 *
## K            0.012017  0.006022  1.996  0.0460 *
## C            0.545003  0.984345  0.554  0.5798
## pH           0.271936  0.913446  0.298  0.7659
## CEC          -0.329449  0.372631 -0.884  0.3766
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
## (Intr) NH4N  NO3N   P     K     C     pH
## NH4N -0.101
## NO3N  0.253  0.279
## P      0.416 -0.080  0.132
## K      -0.320  0.519  0.241 -0.497
## C      -0.678 -0.283 -0.473 -0.136 -0.081
## pH     -0.954 -0.147 -0.340 -0.445  0.142  0.703
## CEC    0.824  0.162  0.241  0.262 -0.255 -0.816 -0.839
## optimizer (Nelder_Mead) convergence code: 0 (OK)
## boundary (singular) fit: see help('isSingular')

# 2022 #
# model occurrence of zeros
hurd.mod.1.t2 <- glmer(non_zero ~ NH4N + NO3N + P + K + C + pH + CEC + (1 | block), data = dat, subset = (time == "t2"))
summary(hurd.mod.1.t2)

## Generalized linear mixed model fit by maximum likelihood (Laplace
## Approximation) [glmerMod]
## Family: binomial ( logit )
## Formula: non_zero ~ NH4N + NO3N + P + K + C + pH + CEC + (1 | block)
## Data: dat
## Subset: (time == "t2")
##
##      AIC      BIC logLik deviance df.resid
##      50.1     72.0   -16.1     32.1      75
##
## Scaled residuals:
##      Min      1Q Median      3Q     Max
## -2.64575  0.00000  0.00003  0.37796  1.00000

```

```

## 
## Random effects:
## Groups Name           Variance Std.Dev.
## block (Intercept) 0         0
## Number of obs: 84, groups: block, 7
##
## Fixed effects:
##                   Estimate Std. Error z value Pr(>|z|)
## (Intercept) -3.288e+02 8.028e+04 -0.004  0.997
## NH4N        -2.501e+00 5.957e+03  0.000  1.000
## NO3N        -1.069e+01 5.041e+03 -0.002  0.998
## P           -8.290e+00 2.236e+03 -0.004  0.997
## K            7.256e-02 7.168e+01  0.001  0.999
## C            1.330e+02 3.031e+04  0.004  0.996
## pH           7.031e+01 2.039e+04  0.003  0.997
## CEC          -3.680e+01 8.088e+03 -0.005  0.996
##
## Correlation of Fixed Effects:
## (Intr) NH4N   NO3N    P      K      C      pH
## NH4N  0.332
## NO3N  0.612  0.398
## P     0.929  0.590  0.780
## K     0.310  0.722  0.123  0.492
## C    -0.907 -0.155 -0.647 -0.795  0.090
## pH   -0.952 -0.587 -0.633 -0.978 -0.553  0.780
## CEC  0.937  0.197  0.606  0.820 -0.012 -0.994 -0.823
## optimizer (Nelder_Mead) convergence code: 0 (OK)
## boundary (singular) fit: see help('isSingular')

# model diversity
hurd.mod.2.t2 <- lmer(diversity ~ NH4N + NO3N + P + K + C + pH + CEC + (1 | block), data = dat, subset = (time == "t2"))
summary(hurd.mod.1.t2)

## Generalized linear mixed model fit by maximum likelihood (Laplace
## Approximation) [glmerMod]
## Family: binomial ( logit )
## Formula: non_zero ~ NH4N + NO3N + P + K + C + pH + CEC + (1 | block)
## Data: dat
## Subset: (time == "t2")
##
##       AIC      BIC  logLik deviance df.resid
##      50.1     72.0   -16.1     32.1      75
##
## Scaled residuals:
##       Min     1Q Median     3Q    Max
## -2.64575  0.00000  0.00003  0.37796  1.00000
##
## Random effects:
## Groups Name           Variance Std.Dev.
## block (Intercept) 0         0
## Number of obs: 84, groups: block, 7
##
## Fixed effects:
##                   Estimate Std. Error z value Pr(>|z|)

```

```

## (Intercept) -3.288e+02 8.028e+04 -0.004 0.997
## NH4N -2.501e+00 5.957e+03 0.000 1.000
## NO3N -1.069e+01 5.041e+03 -0.002 0.998
## P -8.290e+00 2.236e+03 -0.004 0.997
## K 7.256e-02 7.168e+01 0.001 0.999
## C 1.330e+02 3.031e+04 0.004 0.996
## pH 7.031e+01 2.039e+04 0.003 0.997
## CEC -3.680e+01 8.088e+03 -0.005 0.996
##
## Correlation of Fixed Effects:
## (Intr) NH4N NO3N P K C pH
## NH4N 0.332
## NO3N 0.612 0.398
## P 0.929 0.590 0.780
## K 0.310 0.722 0.123 0.492
## C -0.907 -0.155 -0.647 -0.795 0.090
## pH -0.952 -0.587 -0.633 -0.978 -0.553 0.780
## CEC 0.937 0.197 0.606 0.820 -0.012 -0.994 -0.823
## optimizer (Nelder_Mead) convergence code: 0 (OK)
## boundary (singular) fit: see help('isSingular')

# 2021 #
# model occurrence of zeros
hurd.mod.1.t1 <- glmer(non_zero ~ NH4N + NO3N + P + K + C + pH + CEC + (1|block), data = dat,subset = (t
summary(hurd.mod.1.t1)

## Generalized linear mixed model fit by maximum likelihood (Laplace
## Approximation) [glmerMod]
## Family: binomial ( logit )
## Formula: non_zero ~ NH4N + NO3N + P + K + C + pH + CEC + (1 | block)
## Data: dat
## Subset: (time == "t1")
##
## AIC BIC logLik deviance df.resid
## 54.1 76.2 -18.1 36.1 77
##
## Scaled residuals:
## Min 1Q Median 3Q Max
## -2.64575 0.00001 0.00006 0.37796 0.77460
##
## Random effects:
## Groups Name Variance Std.Dev.
## block (Intercept) 0 0
## Number of obs: 86, groups: block, 7
##
## Fixed effects:
## Estimate Std. Error z value Pr(>|z|)
## (Intercept) -117.54083 932.39829 -0.126 0.900
## NH4N -4.42809 66.31781 -0.067 0.947
## NO3N -0.53681 683.66058 -0.001 0.999
## P -8.26123 112.65465 -0.073 0.942
## K -0.08177 0.45400 -0.180 0.857
## C 18.75879 467.48021 0.040 0.968
## pH 35.15730 187.68745 0.187 0.851

```

```

## CEC          -1.73617  274.35999 -0.006    0.995
##
## Correlation of Fixed Effects:
##      (Intr) NH4N   NO3N    P     K     C     pH
## NH4N  0.911
## NO3N -0.324  0.041
## P     -0.621 -0.757 -0.008
## K     0.156  0.068  0.037  0.598
## C     -0.871 -0.972 -0.032  0.884  0.164
## pH    -0.881 -0.888 -0.052  0.394 -0.474  0.777
## CEC   0.805  0.919  0.024 -0.950 -0.329 -0.985 -0.660
## optimizer (Nelder_Mead) convergence code: 0 (OK)
## boundary (singular) fit: see help('isSingular')

# model diversity
hurd.mod.2.t1 <- lmer(diversity ~ NH4N + NO3N + P + K + C + pH + CEC + (1|block), data = dat, subset = (time == "t1"))
summary(hurd.mod.2.t1)

## Generalized linear mixed model fit by maximum likelihood (Laplace
## Approximation) [glmerMod]
## Family: binomial ( logit )
## Formula: non_zero ~ NH4N + NO3N + P + K + C + pH + CEC + (1 | block)
## Data: dat
## Subset: (time == "t1")
##
##      AIC      BIC      logLik deviance df.resid
##      54.1     76.2     -18.1      36.1       77
##
## Scaled residuals:
##      Min      1Q      Median      3Q      Max
## -2.64575  0.00001  0.00006  0.37796  0.77460
##
## Random effects:
## Groups Name        Variance Std.Dev.
## block  (Intercept) 0         0
## Number of obs: 86, groups: block, 7
##
## Fixed effects:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -117.54083  932.39829 -0.126   0.900
## NH4N        -4.42809   66.31781 -0.067   0.947
## NO3N        -0.53681  683.66058 -0.001   0.999
## P           -8.26123  112.65465 -0.073   0.942
## K           -0.08177   0.45400 -0.180   0.857
## C           18.75879  467.48021  0.040   0.968
## pH          35.15730  187.68745  0.187   0.851
## CEC         -1.73617  274.35999 -0.006   0.995
##
## Correlation of Fixed Effects:
##      (Intr) NH4N   NO3N    P     K     C     pH
## NH4N  0.911
## NO3N -0.324  0.041
## P     -0.621 -0.757 -0.008
## K     0.156  0.068  0.037  0.598

```

```

## C      -0.871 -0.972 -0.032  0.884  0.164
## pH     -0.881 -0.888 -0.052  0.394 -0.474  0.777
## CEC     0.805  0.919  0.024 -0.950 -0.329 -0.985 -0.660
## optimizer (Nelder_Mead) convergence code: 0 (OK)
## boundary (singular) fit: see help('isSingular')

# 2020 #
# model occurrence of zeros
hurd.mod.1.t0 <- glmer(non_zero ~ NH4N + NO3N + P + K + C + pH + CEC + (1|block), data = dat,subset = (t0 >= 200))
summary(hurd.mod.1.t0)

## Generalized linear mixed model fit by maximum likelihood (Laplace
## Approximation) [glmerMod]
## Family: binomial ( logit )
## Formula: non_zero ~ NH4N + NO3N + P + K + C + pH + CEC + (1 | block)
## Data: dat
## Subset: (time == "t0")
##
##       AIC      BIC  logLik deviance df.resid
##   261.6    292.2   -121.8     243.6     214
##
## Scaled residuals:
##   Min     1Q  Median     3Q     Max
## -2.8569 -1.1897  0.4454  0.6615  0.8405
##
## Random effects:
##   Groups Name        Variance Std.Dev.
##   block  (Intercept) 0         0
## Number of obs: 223, groups: block, 7
##
## Fixed effects:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) 1.150196  4.961672  0.232  0.8167
## NH4N        0.440379  0.273946  1.608  0.1079
## NO3N        0.462437  0.198275  2.332  0.0197 *
## P          -0.083658  0.192287 -0.435  0.6635
## K           0.016228  0.007456  2.176  0.0295 *
## C          -0.304223  1.123534 -0.271  0.7866
## pH          -0.704517  1.037192 -0.679  0.4970
## CEC        -0.131590  0.427371 -0.308  0.7582
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##   (Intr) NH4N  NO3N   P      K      C      pH
## NH4N -0.098
## NO3N  0.273  0.267
## P      0.353 -0.013  0.083
## K      -0.318  0.466  0.237 -0.511
## C      -0.690 -0.243 -0.499 -0.124 -0.060
## pH     -0.956 -0.155 -0.352 -0.396  0.153  0.701
## CEC    0.822  0.143  0.245  0.263 -0.317 -0.794 -0.828
## optimizer (Nelder_Mead) convergence code: 0 (OK)
## boundary (singular) fit: see help('isSingular')

```

```

# model diversity
hurd.mod.2.t0 <- lmer(diversity ~ NH4N + NO3N + P + K + C + pH + CEC + (1|block), data = dat, subset = (time == "t0"))
summary(hurd.mod.1.t0)

## Generalized linear mixed model fit by maximum likelihood (Laplace
## Approximation) [glmerMod]
## Family: binomial ( logit )
## Formula: non_zero ~ NH4N + NO3N + P + K + C + pH + CEC + (1 | block)
## Data: dat
## Subset: (time == "t0")
##
##      AIC      BIC  logLik deviance df.resid
##    261.6    292.2   -121.8     243.6     214
##
## Scaled residuals:
##    Min     1Q Median     3Q    Max
## -2.8569 -1.1897  0.4454  0.6615  0.8405
##
## Random effects:
## Groups Name        Variance Std.Dev.
## block (Intercept) 0         0
## Number of obs: 223, groups: block, 7
##
## Fixed effects:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) 1.150196  4.961672  0.232  0.8167
## NH4N        0.440379  0.273946  1.608  0.1079
## NO3N        0.462437  0.198275  2.332  0.0197 *
## P          -0.083658  0.192287 -0.435  0.6635
## K           0.016228  0.007456  2.176  0.0295 *
## C          -0.304223  1.123534 -0.271  0.7866
## pH          -0.704517  1.037192 -0.679  0.4970
## CEC        -0.131590  0.427371 -0.308  0.7582
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
## (Intr) NH4N  NO3N   P      K      C      pH
## NH4N -0.098
## NO3N  0.273  0.267
## P      0.353 -0.013  0.083
## K      -0.318  0.466  0.237 -0.511
## C      -0.690 -0.243 -0.499 -0.124 -0.060
## pH     -0.956 -0.155 -0.352 -0.396  0.153  0.701
## CEC    0.822  0.143  0.245  0.263 -0.317 -0.794 -0.828
## optimizer (Nelder_Mead) convergence code: 0 (OK)
## boundary (singular) fit: see help('isSingular')

```

Composition ~ nutrient composition 2020 - There is no significant correlation between nutrient elements and plant composition.

```

# Data wrangling
# This data set does not include unknown data.
comm <- read.csv("20-22_species_composition_data_no_unk.csv", header = T)

# Remove the locations surveyed in 2021 and 2022 that were not surveyed in 2020 - aka, cm=0, cm=21, cm=22
comm<-comm[which(comm$cm_location!=21 & comm$cm_location!=0 & comm$cm_location!=22 & comm$cm_location!=20),]

# Make a group name for each row
comm$grp<-apply(comm[c(1,3,5,6,7)], 1, paste, collapse=":") # timepoint, block, transect_name, initial

# Need to make each row a community using matrify.
commsub<-comm[,c(15,10,13)] # group, species_code, and count of each species for each transect. transect
commsub <-as.data.frame(commsub)
commtry<-matrify(commsub) # make it an expanded species matrix

## Warning in matrify(commsub): NAs introduced by coercion

commtry$x[which(is.na(commtry$x))] <- 1 # "x" means that there were no individuals in the transect, but
ncol(commtry) # how many species are we working with in our community matrix

## [1] 87

# Store grouping row names as a column, then remove rownames.
commtry$grps<-rownames(commtry)
rownames(commtry)<-NULL
names(commtry)

## [1] "acul"   "aiku"   "arca"   "ardy"   "arsp"   "ael"    "bldr"   "blrd"   "brdi"
## [10] "brdr"   "brpe"   "brru"   "buse"   "caer"   "cagr"   "cahi"   "casp"   "cear"
## [19] "chau"   "chei"   "chps"   "crcl"   "crc"    "cusc"   "cusp"   "dagl"   "dosp"
## [28] "ento"   "erau"   "ercy"   "erra"   "ersp"   "gite"   "gnate"  "gobe"   "gocy"
## [37] "gono"   "goro"   "gosp"   "haod"   "hygl"   "hyp"    "hypo"   "jubu"   "laro"
## [46] "ledu"   "lele"   "loef"   "misp"   "mite"   "momo"   "mopa"   "niro"   "omco"
## [55] "orsp"   "pala"   "peai"   "pedu"   "phsu"   "plde"   "poar"   "poca"   "pocap"
## [64] "poce"   "pogn"   "pole"   "pomu"   "pter"   "ptga"   "ptob"   "rhl"   "rhpy"
## [73] "rhsp"   "ry"     "scna"   "sino"   "sool"   "stfi"   "stpi"   "thma"   "trcy"
## [82] "tris"   "tror"   "trpi"   "waac"   "wagr"   "x"      "grps"

# Split group info into columns for each variable
mat<-separate(commtry, 88, c("time","block","transect","init","treatment"), ":")
# names(mat) #check

# Set up nutrient analysis data
nutrient <- read.csv("Nutrient.csv", header = T)
nutrient <-separate(nutrient, 2 , c("block","plot"), "_")
nutrient <- nutrient[,2:17]

# subset data where all t0 communities, insitu log and insitu open communities at t1 and t2 are included
mat2 <- mat[which(mat$time=="t0"),]
mat2$grp<-apply(mat2[c(89,91)], 1, paste, collapse=":") # block, init as grouping
# names(mat2) #check

```

```

# another df where the grouping variables are time, block and initial state
# each row is a transect in a certain year.
df<-mat2[,c(1:87, 93)]
df2 = df %>% mutate(across(.cols=1:87,.fns=as.numeric)) # make everything numeric
rownames(df2)<-NULL # remove rownames

## new with group vars
nublock<-separate(df2, 88, c("block", "init"), ":") # just looking at time, block & initial treatment

# want to sum across transects in same block X init treatment
nublock$sumgrp<-apply(mat2[c(89, 91)], 1, paste, collapse=":")
# head(nublock)

# sum observations across initial X block (group variable)
# this gives number of plants in each transect TYPE for each year in each block. should be 2 types X 3 :
blocksum<-rowsum(nublock[,c(1:87)], group=nublock$sumgrp)
blocksum$grps<-rownames(blocksum)
rownames(blocksum)<-NULL # remove rownames
# nrow(blocksum) # it is 14 rows as expected

## expand again
blocksum<-separate(blocksum, 88, c("block", "init"), ":")
nutrient_join <- nutrient[,c(1,3,5:9, 11,16)]
blocksum <- inner_join(blocksum, nutrient_join, by = c("init", "block"))

# at the moment this includes where there were no plants ("x" column in matrix)
assemblies_t0<-blocksum[,c(1:87)]

# group - these are the treatment variables that need to be separately fed into the MDS analaysis from
group_init<-blocksum$init
group_block<-blocksum$block
group_nutrient<-blocksum[,c(90:96)]

# MDS
ass.rel.t0<-decostand(assemblies_t0, method='hel') #standardize assemblies
ass.rel.t0_NMS <- metaMDS(ass.rel.t0, distance = 'bray', k = 4) # run MDS

## Run 0 stress 0.04866056
## Run 1 stress 0.05074883
## Run 2 stress 0.04798366
## ... New best solution
## ... Procrustes: rmse 0.1312748 max resid 0.3125501
## Run 3 stress 0.04798383
## ... Procrustes: rmse 0.0002271243 max resid 0.0004150457
## ... Similar to previous best
## Run 4 stress 0.04866039
## Run 5 stress 0.05073541
## Run 6 stress 0.05073014
## Run 7 stress 0.04798367
## ... Procrustes: rmse 2.784709e-05 max resid 6.19703e-05
## ... Similar to previous best
## Run 8 stress 0.04798371
## ... Procrustes: rmse 0.0001038247 max resid 0.0001806507

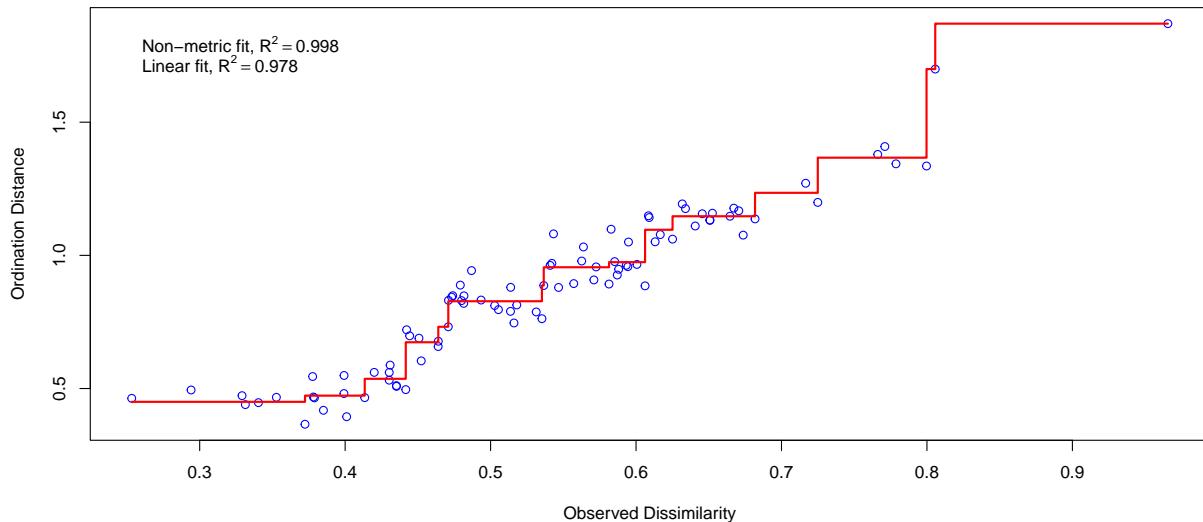
```

```

## ... Similar to previous best
## Run 9 stress 0.04866021
## Run 10 stress 0.04798366
## ... New best solution
## ... Procrustes: rmse 2.299283e-05 max resid 4.479975e-05
## ... Similar to previous best
## Run 11 stress 0.0479837
## ... Procrustes: rmse 0.000118785 max resid 0.0003062292
## ... Similar to previous best
## Run 12 stress 0.04866074
## Run 13 stress 0.04798375
## ... Procrustes: rmse 0.0001708756 max resid 0.0004551857
## ... Similar to previous best
## Run 14 stress 0.05072926
## Run 15 stress 0.04798374
## ... Procrustes: rmse 0.0001337627 max resid 0.0003485695
## ... Similar to previous best
## Run 16 stress 0.05009237
## Run 17 stress 0.04866067
## Run 18 stress 0.04798374
## ... Procrustes: rmse 0.0001657922 max resid 0.0002755445
## ... Similar to previous best
## Run 19 stress 0.05072989
## Run 20 stress 0.04866023
## *** Best solution repeated 5 times

```

```
stressplot(ass.rel.t0_NMS) # check fit
```



```

en.nutrient = envfit(ass.rel.t0_NMS, group_nutrient, permutations = 999, na.rm = TRUE)
# plot(ass.rel.t0_NMS)
# plot(en.nutrient)
print(en.nutrient) ##### it is basically saying most of the nutrient does not affect the composition

```

```

## 
## ***VECTORS
##
##          NMDS1      NMDS2      r2 Pr(>r)
## NH4N -0.83690 -0.54736 0.1046  0.522
## NO3N -0.47216 -0.88151 0.0219  0.901
## P     1.00000 -0.00230 0.2459  0.224
## K     0.36657  0.93039 0.1746  0.340
## C    -0.83028  0.55735 0.1548  0.406
## pH    0.88209  0.47109 0.3441  0.089 .
## CEC   0.11248  0.99365 0.1267  0.486
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## Permutation: free
## Number of permutations: 999

# scores
mds_scores_t0<-as.data.frame(vegan::scores(ass.rel.t0_NMS)$sites) # extract scores
mds_scores_t0$site<-rownames(vegan::scores(ass.rel.t0_NMS)$sites) # extract names
mds_scores_t0$treatment<-group_init # grouping factor 1
mds_scores_t0$block<-group_block # grouping factor 2
en_coord_cont = as.data.frame(vegan::scores(en.nutrient, "vectors")) * ordiArrowMul(en.nutrient)

# explaining factors
init<-as.factor(group_init) # grouping factor 1- convert to factor
block<-as.factor(group_block) # grouping factor 2- convert to factor

##### Redundancy analysis
# can look at significance of model where initial treatment and block explain variation in community
trt_tot_2<-rda(ass.rel.t0~init+block) # run model using standardized data
summary(trt_tot_2)

## 
## Call:
## rda(formula = ass.rel.t0 ~ init + block)
##
## Partitioning of variance:
##           Inertia Proportion
## Total      0.42624    1.0000
## Constrained 0.32981    0.7738
## Unconstrained 0.09643    0.2262
##
## Eigenvalues, and their contribution to the variance
##
## Importance of components:
##          RDA1     RDA2     RDA3     RDA4     RDA5     RDA6     RDA7
## Eigenvalue 0.1145  0.0834  0.03956  0.03545  0.02748  0.01667  0.01278
## Proportion Explained 0.2686  0.1957  0.09282  0.08316  0.06446  0.03911  0.02999
## Cumulative Proportion 0.2686  0.4642  0.55704  0.64021  0.70467  0.74378  0.77376
##          PC1     PC2     PC3     PC4     PC5     PC6
## Eigenvalue 0.03207 0.02047 0.01524 0.01394 0.01034 0.004375
## Proportion Explained 0.07523 0.04802 0.03576 0.03271 0.02426 0.010264
## Cumulative Proportion 0.84900 0.89701 0.93277 0.96548 0.98974 1.000000
##

```

```

## Accumulated constrained eigenvalues
## Importance of components:
##          RDA1     RDA2     RDA3     RDA4     RDA5     RDA6     RDA7
## Eigenvalue      0.1145  0.0834  0.03956  0.03545  0.02748  0.01667  0.01278
## Proportion Explained  0.3471  0.2529  0.11995  0.10748  0.08331  0.05055  0.03876
## Cumulative Proportion 0.3471  0.6000  0.71991  0.82739  0.91070  0.96124  1.00000
##
## Scaling 2 for species and site scores
## * Species are scaled proportional to eigenvalues
## * Sites are unscaled: weighted dispersion equal on all dimensions
## * General scaling constant of scores: 1.534259
##
##
## Species scores
##
##          RDA1     RDA2     RDA3     RDA4     RDA5     RDA6
## acul -1.641e-01 -3.061e-02  5.709e-03 -6.882e-02  1.731e-01 -1.547e-02
## aicu  9.454e-02 -2.259e-02 -1.224e-02  5.916e-02  6.241e-02  5.770e-02
## arca  1.124e-16 -9.425e-17 -1.501e-17  2.657e-17 -6.239e-18 -5.518e-17
## ardy  1.229e-17 -8.987e-19  1.159e-17 -1.534e-17 -1.923e-18  1.107e-17
## arsp  -2.057e-02  1.214e-03 -1.623e-02 -3.482e-02  1.144e-02  2.195e-02
## auel -1.119e-18  8.250e-19 -1.770e-17 -2.089e-18 -1.530e-17 -1.169e-17
## bldr -4.650e-02 -1.865e-01  1.009e-01 -2.439e-01 -6.057e-02  6.906e-02
## blrd -8.371e-18 -1.889e-17  5.282e-18 -1.012e-17  7.877e-18 -5.634e-18
## brdi  1.057e-17  2.442e-17 -7.953e-19 -9.977e-18 -7.727e-19  1.304e-17
## brdr  9.121e-19  4.301e-19  4.740e-19  1.210e-18 -4.747e-19 -4.240e-19
## brpe  3.002e-17  4.527e-17  1.770e-18  1.447e-17 -1.303e-17  1.438e-17
## brru  9.723e-34 -1.029e-33  2.395e-33  7.087e-35 -3.079e-34 -6.179e-33
## buse -2.968e-02  2.051e-02  3.459e-02 -3.225e-02  4.471e-02 -1.316e-02
## caer  4.476e-02  1.265e-01  5.800e-02  1.528e-02  7.746e-03  2.318e-03
## cagr -2.673e-02  1.192e-03  9.765e-02  8.019e-02  5.774e-02 -1.108e-02
## cahi -7.468e-02  1.317e-02 -9.131e-02  6.646e-03 -2.484e-02 -5.028e-02
## casp -3.792e-02  5.793e-02 -6.467e-02 -4.605e-02 -7.997e-03  1.873e-03
## cear  0.000e+00  0.000e+00  0.000e+00  0.000e+00  0.000e+00  0.000e+00
## chau  1.462e-01  2.094e-03 -7.128e-02  1.410e-02  1.225e-02  5.751e-02
## chei  0.000e+00  0.000e+00  0.000e+00  0.000e+00  0.000e+00  0.000e+00
## chps -1.385e-01 -5.737e-02 -1.133e-02 -7.725e-02 -2.818e-02  1.728e-02
## crcl  0.000e+00  0.000e+00  0.000e+00  0.000e+00  0.000e+00  0.000e+00
## crco -1.608e-01  5.208e-02 -5.018e-02 -4.208e-02  2.736e-02 -9.587e-02
## cusc  0.000e+00  0.000e+00  0.000e+00  0.000e+00  0.000e+00  0.000e+00
## cusp  0.000e+00  0.000e+00  0.000e+00  0.000e+00  0.000e+00  0.000e+00
## dagl -2.996e-02 -1.069e-03 -7.782e-03  4.675e-02 -2.198e-02 -6.498e-03
## dosp  0.000e+00  0.000e+00  0.000e+00  0.000e+00  0.000e+00  0.000e+00
## ento  0.000e+00  0.000e+00  0.000e+00  0.000e+00  0.000e+00  0.000e+00
## erau  0.000e+00  0.000e+00  0.000e+00  0.000e+00  0.000e+00  0.000e+00
## ercy  0.000e+00  0.000e+00  0.000e+00  0.000e+00  0.000e+00  0.000e+00
## erra -2.630e-02  2.865e-03  5.020e-02  9.019e-03  1.143e-01 -3.362e-02
## ersp -2.457e-03  6.427e-03  2.496e-02  3.462e-03  3.914e-02 -2.745e-02
## gite  2.530e-01 -1.813e-01 -1.258e-01  3.334e-02  5.780e-02 -2.361e-03
## gnte  0.000e+00  0.000e+00  0.000e+00  0.000e+00  0.000e+00  0.000e+00
## gobe -1.398e-02  1.872e-01 -4.743e-02  2.742e-02  8.577e-02 -7.442e-03
## gocy  2.797e-02  1.778e-01 -6.369e-02 -4.962e-02 -2.874e-02  4.811e-02
## gono  0.000e+00  0.000e+00  0.000e+00  0.000e+00  0.000e+00  0.000e+00
## goro -4.960e-02 -7.505e-02 -1.754e-02  5.339e-02 -1.099e-01 -1.844e-02

```

```

## gosp  7.031e-02 9.069e-02 -5.139e-02 -3.052e-02 -5.351e-05 8.966e-02
## haod -1.248e-02 6.184e-02 -2.591e-02 -9.287e-02 1.495e-02 2.948e-02
## hygl -2.940e-01 8.742e-02 8.182e-02 -1.113e-01 1.533e-02 1.122e-02
## hypi  0.000e+00 0.000e+00 0.000e+00 0.000e+00 0.000e+00 0.000e+00
## hypo  0.000e+00 0.000e+00 0.000e+00 0.000e+00 0.000e+00 0.000e+00
## jubu  0.000e+00 0.000e+00 0.000e+00 0.000e+00 0.000e+00 0.000e+00
## laro  1.049e-01 1.210e-01 5.046e-02 -1.820e-02 -5.824e-02 -5.001e-03
## ledu  0.000e+00 0.000e+00 0.000e+00 0.000e+00 0.000e+00 0.000e+00
## lele  -2.057e-02 1.214e-03 -1.623e-02 -3.482e-02 1.144e-02 2.195e-02
## loef  0.000e+00 0.000e+00 0.000e+00 0.000e+00 0.000e+00 0.000e+00
## misp  3.690e-02 2.197e-01 1.344e-01 4.312e-02 1.060e-01 2.985e-02
## mite  0.000e+00 0.000e+00 0.000e+00 0.000e+00 0.000e+00 0.000e+00
## momo  0.000e+00 0.000e+00 0.000e+00 0.000e+00 0.000e+00 0.000e+00
## mopa  0.000e+00 0.000e+00 0.000e+00 0.000e+00 0.000e+00 0.000e+00
## niro  0.000e+00 0.000e+00 0.000e+00 0.000e+00 0.000e+00 0.000e+00
## omco  -6.761e-02 6.264e-03 -8.979e-02 4.478e-02 -3.621e-02 -5.944e-02
## orsp  0.000e+00 0.000e+00 0.000e+00 0.000e+00 0.000e+00 0.000e+00
## pala -2.057e-02 1.214e-03 -1.623e-02 -3.482e-02 1.144e-02 2.195e-02
## peai  8.389e-02 -4.825e-02 -9.446e-02 -6.195e-02 8.932e-02 7.437e-02
## pedu -2.909e-02 1.717e-03 -2.296e-02 -4.924e-02 1.617e-02 3.105e-02
## phsu  0.000e+00 0.000e+00 0.000e+00 0.000e+00 0.000e+00 0.000e+00
## plde  0.000e+00 0.000e+00 0.000e+00 0.000e+00 0.000e+00 0.000e+00
## poar  0.000e+00 0.000e+00 0.000e+00 0.000e+00 0.000e+00 0.000e+00
## poca -3.514e-03 -6.344e-02 -3.487e-03 1.164e-01 1.794e-02 3.093e-02
## pocap 6.856e-02 1.456e-01 -1.503e-02 -7.994e-04 -2.630e-02 7.238e-02
## poce  0.000e+00 0.000e+00 0.000e+00 0.000e+00 0.000e+00 0.000e+00
## pogn  0.000e+00 0.000e+00 0.000e+00 0.000e+00 0.000e+00 0.000e+00
## pole  4.261e-02 -3.143e-01 1.989e-01 5.584e-03 3.903e-02 -2.368e-02
## pomu -3.888e-01 -6.207e-02 6.793e-03 1.601e-01 -7.784e-02 7.456e-02
## pter  0.000e+00 0.000e+00 0.000e+00 0.000e+00 0.000e+00 0.000e+00
## ptga -2.638e-02 1.156e-01 5.406e-02 -4.216e-03 2.747e-02 1.276e-02
## ptob  0.000e+00 0.000e+00 0.000e+00 0.000e+00 0.000e+00 0.000e+00
## rhla  1.865e-01 -1.907e-01 -5.259e-02 3.145e-02 5.749e-02 2.988e-02
## rhpy -7.913e-02 1.351e-02 -8.944e-02 1.770e-02 5.222e-03 -1.066e-01
## rhsp  0.000e+00 0.000e+00 0.000e+00 0.000e+00 0.000e+00 0.000e+00
## ry   0.000e+00 0.000e+00 0.000e+00 0.000e+00 0.000e+00 0.000e+00
## scna 0.000e+00 0.000e+00 0.000e+00 0.000e+00 0.000e+00 0.000e+00
## sino 0.000e+00 0.000e+00 0.000e+00 0.000e+00 0.000e+00 0.000e+00
## sool 0.000e+00 0.000e+00 0.000e+00 0.000e+00 0.000e+00 0.000e+00
## stfi 2.849e-02 4.794e-02 4.360e-03 2.079e-06 -1.066e-02 7.798e-04
## stpi 0.000e+00 0.000e+00 0.000e+00 0.000e+00 0.000e+00 0.000e+00
## thma 0.000e+00 0.000e+00 0.000e+00 0.000e+00 0.000e+00 0.000e+00
## trcy 2.872e-01 6.164e-02 2.157e-02 -1.214e-01 -9.430e-02 -1.085e-01
## tris 0.000e+00 0.000e+00 0.000e+00 0.000e+00 0.000e+00 0.000e+00
## tror 1.659e-01 1.497e-01 9.863e-02 2.978e-02 -4.833e-02 9.223e-04
## trpi 4.134e-03 -1.302e-02 2.638e-02 -2.102e-02 -2.835e-02 -1.796e-02
## waac 1.327e-01 3.005e-02 1.793e-01 8.160e-02 -6.604e-02 6.145e-03
## wagr 2.804e-02 6.822e-03 2.162e-02 -3.666e-02 6.890e-02 -6.540e-02
## x    4.222e-02 2.810e-02 4.696e-02 1.101e-02 9.372e-02 -2.970e-02
##
##
## Site scores (weighted sums of species scores)
##
##          RDA1      RDA2      RDA3      RDA4      RDA5      RDA6

```

```

## row1  0.01134  0.03365  0.5830  0.59882 -0.13199  0.07154
## row2 -0.35273 -0.20722  0.2908  0.68511 -0.24191  0.60925
## row3  0.92160 -0.40472 -0.1107 -0.08527  0.03137 -0.35104
## row4  0.54518 -0.66372 -0.6352  0.27979  0.31059  0.36234
## row5  0.14409 -0.31373  0.5202 -0.58134 -0.85710 -0.21776
## row6 -0.25541 -0.45839  0.3355 -0.32197 -0.24935  0.06507
## row7 -0.07265  0.27890  0.4259  0.29129  0.89699 -0.66481
## row8 -0.31099 -0.24472  0.2812 -0.16299  0.73556  0.20448
## row9 -0.24068  0.09279 -0.5603  0.19753 -0.39454 -1.00721
## row10 -0.67007  0.03489 -0.6692  0.44074 -0.13873  0.05273
## row11 -0.16582  0.16093 -0.0174 -0.44916  0.14658 -0.08480
## row12 -0.33932  0.10302 -0.3393 -0.88552  0.19211  0.31775
## row13  0.40851  0.91226  0.1000  0.05301 -0.04839  0.10831
## row14  0.37695  0.67608 -0.2046 -0.06004 -0.25121  0.53413
##
##
## Site constraints (linear combinations of constraining variables)
##
##          RDA1      RDA2      RDA3      RDA4      RDA5      RDA6
## row1   -0.02692  0.02179  0.57132  6.455e-01 -0.23796  0.03386
## row2   -0.31447 -0.19537  0.30252  6.384e-01 -0.13594  0.64693
## row3   0.87716 -0.42564 -0.23853  1.008e-01  0.11997 -0.30089
## row4   0.58962 -0.64280 -0.50733  9.371e-02  0.22199  0.31219
## row5   0.08811 -0.27748  0.56227 -4.481e-01 -0.60424 -0.38288
## row6   -0.19943 -0.49464  0.29347 -4.552e-01 -0.50222  0.23019
## row7   -0.04805  0.12567  0.48796  6.770e-02  0.76527 -0.53670
## row8   -0.33559 -0.09149  0.21916  6.060e-02  0.86729  0.07638
## row9   -0.31160  0.17242 -0.48036  3.227e-01 -0.31764 -0.78378
## row10  -0.59915 -0.04474 -0.74917  3.156e-01 -0.21562 -0.17070
## row11  -0.10880  0.24056 -0.04395 -6.638e-01  0.11834 -0.19006
## row12  -0.39634  0.02339 -0.31275 -6.709e-01  0.22036  0.42302
## row13  0.53650  0.90275  0.08210  3.915e-05 -0.20081  0.01469
## row14  0.24896  0.68558 -0.18670 -7.067e-03 -0.09879  0.62776
##
##
## Biplot scores for constraining variables
##
##          RDA1      RDA2      RDA3      RDA4      RDA5      RDA6
## initopen -0.35062 -0.26480 -0.32777 -0.008665  0.1244  0.747567
## block2    0.73017 -0.53188 -0.37129  0.096832  0.1702  0.005627
## block3   -0.05542 -0.38436  0.42599 -0.449672 -0.5508 -0.076008
## block4   -0.19098  0.01701  0.35200  0.063869  0.8127 -0.229152
## block5   -0.45338  0.06356 -0.61207  0.317733 -0.2655 -0.475148
## block6   -0.25146  0.13140 -0.17757 -0.664411  0.1686  0.115967
## block7    0.39101  0.79068 -0.05207 -0.003498 -0.1491  0.319814
##
##
## Centroids for factor constraints
##
##          RDA1      RDA2      RDA3      RDA4      RDA5      RDA6
## initlog   0.14377  0.10858  0.1344  0.003553 -0.05101 -0.306538
## initopen -0.14377 -0.10858 -0.1344 -0.003553  0.05101  0.306538
## block1   -0.17070 -0.08679  0.4369  0.641965 -0.18695  0.340395
## block2    0.73339 -0.53422 -0.3729  0.097259  0.17098  0.005651

```

```

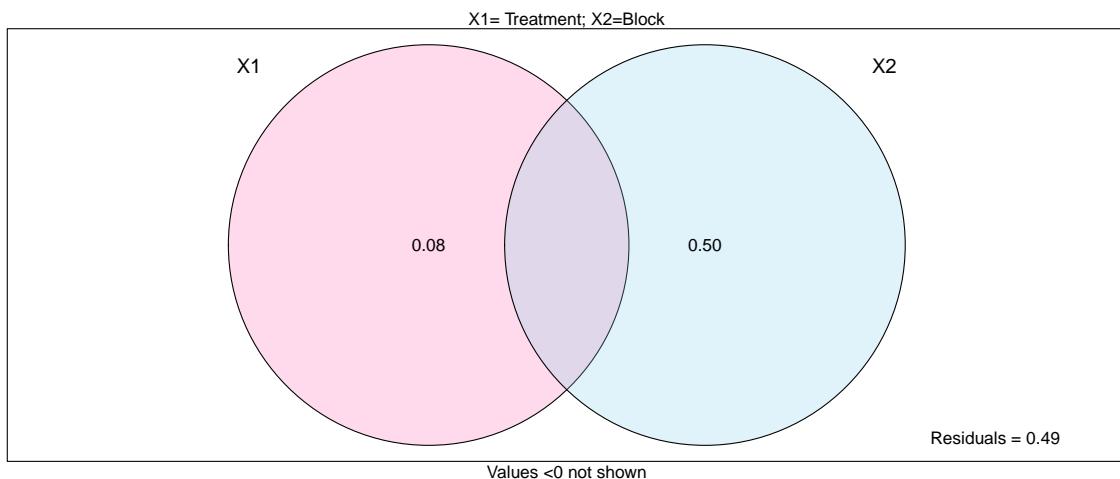
## block3 -0.05566 -0.38606 0.4279 -0.451654 -0.55323 -0.076344
## block4 -0.19182 0.01709 0.3536 0.064151 0.81628 -0.230162
## block5 -0.45538 0.06384 -0.6148 0.319134 -0.26663 -0.477242
## block6 -0.25257 0.13197 -0.1784 -0.667340 0.16935 0.116478
## block7 0.39273 0.79417 -0.0523 -0.003514 -0.14980 0.321224

anova.cca(trt_tot_2, step=1000, by="term") ## test for model significance

## Permutation test for rda under reduced model
## Terms added sequentially (first to last)
## Permutation: free
## Number of permutations: 999
##
## Model: rda(formula = ass.rel.t0 ~ init + block)
##          Df Variance      F Pr(>F)
## init      1 0.035514 2.2097 0.020 *
## block     6 0.294294 3.0519 0.001 ***
## Residual  6 0.096430
## ---
## Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

# Using varpart to look at contributions of initial treatment and block
var.mod2<-varpart(ass.rel.t0, init, block) # run model on standardized data
plot(var.mod2, bg=c("hotpink","skyblue"))
mtext("X1= Treatment; X2=Block", side=3)

```



```

# can test for significance of contribution of the fraction of initial treatment
# do this with partial redundancy analysis
trt_Frac<-rda(ass.rel.t0, init, block) # partial rda model
summary(trt_Frac)

```

##

```

## Call:
## rda(X = ass.rel.t0, Y = init, Z = block)
##
## Partitioning of variance:
##           Inertia Proportion
## Total      0.42624   1.00000
## Conditioned 0.29429   0.69045
## Constrained 0.03551   0.08332
## Unconstrained 0.09643   0.22624
##
## Eigenvalues, and their contribution to the variance
## after removing the contribution of conditioning variables
##
## Importance of components:
##           RDA1      PC1      PC2      PC3      PC4      PC5      PC6
## Eigenvalue      0.03551 0.03207 0.02047 0.01524 0.01394 0.01034 0.004375
## Proportion Explained 0.26916 0.24303 0.15511 0.11552 0.10566 0.07837 0.033159
## Cumulative Proportion 0.26916 0.51219 0.66730 0.78282 0.88848 0.96684 1.000000
##
## Accumulated constrained eigenvalues
## Importance of components:
##           RDA1
## Eigenvalue      0.03551
## Proportion Explained 1.00000
## Cumulative Proportion 1.00000
##
## Scaling 2 for species and site scores
## * Species are scaled proportional to eigenvalues
## * Sites are unscaled: weighted dispersion equal on all dimensions
## * General scaling constant of scores: 1.534259
##
##
## Species scores
##
##           RDA1      PC1      PC2      PC3      PC4      PC5
## acul    0.1067309 -9.564e-02 -2.516e-02  6.791e-02 -5.735e-02 -1.985e-02
## aicu    0.0342552  7.579e-02 -2.904e-02 -1.040e-01 -1.656e-02  3.629e-04
## arca    0.0000000 -5.092e-17  4.573e-17 -2.095e-17  2.733e-18  2.352e-18
## ardy    0.0000000 -1.340e-17  2.878e-18  4.804e-18  6.089e-18  3.913e-18
## arsp    0.0212814 -9.296e-03  3.074e-02 -1.243e-02  2.902e-02  2.380e-02
## auel    0.0000000  1.657e-17  1.767e-18 -7.370e-18 -1.235e-17 -4.376e-19
## bldr    0.0627812 -3.489e-02  5.885e-02 -3.487e-02  5.598e-02 -1.193e-01
## blrd    0.0000000  2.296e-18  1.300e-17  5.184e-19  1.453e-17  5.429e-18
## brdi    0.0000000 -1.645e-17  1.224e-17 -1.804e-17 -5.031e-18  1.778e-17
## brdr    0.0000000 -1.535e-18  1.057e-18  1.817e-19  3.219e-18  1.290e-18
## brpe    0.0000000  4.493e-17  7.212e-18 -2.652e-17 -7.924e-17 -1.269e-17
## brru    0.0000000 -3.497e-33 -2.348e-33  3.058e-33  5.468e-33 -1.861e-34
## buse    -0.0399181  8.064e-03 -8.920e-03 -7.638e-03 -1.217e-02  6.087e-02
## caer    -0.0770469  5.739e-03 -4.007e-02  2.232e-02 -3.681e-02  4.977e-02
## cagr    -0.0353354  3.492e-02 -5.082e-02 -7.084e-02  2.218e-02  4.523e-02
## cahi    0.0073560  1.491e-02  4.828e-02 -6.274e-03  3.863e-02  1.972e-02
## casp    0.0071091 -7.320e-02 -4.968e-02  8.494e-03  2.080e-02  8.123e-03
## clear   0.0000000  0.000e+00  0.000e+00  0.000e+00  0.000e+00  0.000e+00
## chau    0.0270050  2.642e-02  5.365e-02 -7.004e-02 -9.242e-02 -2.682e-03

```

```

## chei  0.0000000 0.000e+00 0.000e+00 0.000e+00 0.000e+00 0.000e+00
## chps 0.0867979 -8.706e-02 -4.807e-02 5.256e-03 -3.053e-02 -5.813e-02
## crcl  0.0000000 0.000e+00 0.000e+00 0.000e+00 0.000e+00 0.000e+00
## crco -0.0054567 -2.817e-02 -2.789e-02 -6.578e-02 -6.492e-03 6.700e-02
## cusc  0.0000000 0.000e+00 0.000e+00 0.000e+00 0.000e+00 0.000e+00
## cusp  0.0000000 0.000e+00 0.000e+00 0.000e+00 0.000e+00 0.000e+00
## dagl -0.0005158 -4.170e-02 -1.197e-02 1.578e-02 -3.486e-02 3.748e-02
## dosp  0.0000000 0.000e+00 0.000e+00 0.000e+00 0.000e+00 0.000e+00
## ento  0.0000000 0.000e+00 0.000e+00 0.000e+00 0.000e+00 0.000e+00
## erau  0.0000000 0.000e+00 0.000e+00 0.000e+00 0.000e+00 0.000e+00
## ercy  0.0000000 0.000e+00 0.000e+00 0.000e+00 0.000e+00 0.000e+00
## erra -0.0017634 1.363e-03 -1.961e-03 -2.641e-03 6.378e-04 2.340e-03
## ersp -0.0209707 1.621e-02 -2.332e-02 -3.140e-02 7.584e-03 2.783e-02
## gite -0.0057450 -7.634e-03 5.428e-03 -7.119e-03 -3.080e-03 -1.608e-03
## gnte  0.0000000 0.000e+00 0.000e+00 0.000e+00 0.000e+00 0.000e+00
## gobe -0.0320039 -3.133e-03 -9.076e-02 5.077e-02 1.243e-01 2.247e-02
## gocy -0.0012808 6.463e-02 4.761e-02 -1.112e-02 -1.947e-02 -4.368e-03
## gono  0.0000000 0.000e+00 0.000e+00 0.000e+00 0.000e+00 0.000e+00
## goro  0.0215820 2.211e-01 -1.082e-01 -1.141e-02 -4.094e-02 -1.836e-02
## gosp  0.0446889 2.138e-02 1.224e-02 1.185e-02 3.669e-02 2.988e-02
## haod -0.0050284 -3.182e-02 7.735e-03 1.165e-02 6.116e-02 1.742e-02
## hygl  0.0601486 6.535e-02 7.287e-02 6.640e-02 -5.544e-02 7.196e-02
## hypi  0.0000000 0.000e+00 0.000e+00 0.000e+00 0.000e+00 0.000e+00
## hypo  0.0000000 0.000e+00 0.000e+00 0.000e+00 0.000e+00 0.000e+00
## jubu  0.0000000 0.000e+00 0.000e+00 0.000e+00 0.000e+00 0.000e+00
## laro -0.1146604 -4.765e-02 -1.202e-02 3.910e-02 -4.021e-02 -1.545e-02
## ledu  0.0000000 0.000e+00 0.000e+00 0.000e+00 0.000e+00 0.000e+00
## lele  0.0212814 -9.296e-03 3.074e-02 -1.243e-02 2.902e-02 2.380e-02
## loef  0.0000000 0.000e+00 0.000e+00 0.000e+00 0.000e+00 0.000e+00
## misp -0.0859180 4.494e-02 -1.140e-01 -3.971e-02 -2.276e-02 5.166e-03
## mite  0.0000000 0.000e+00 0.000e+00 0.000e+00 0.000e+00 0.000e+00
## momo  0.0000000 0.000e+00 0.000e+00 0.000e+00 0.000e+00 0.000e+00
## mopa  0.0000000 0.000e+00 0.000e+00 0.000e+00 0.000e+00 0.000e+00
## niro  0.0000000 0.000e+00 0.000e+00 0.000e+00 0.000e+00 0.000e+00
## omco  0.0102406 -1.780e-02 -1.290e-02 -4.528e-03 -7.063e-03 3.001e-03
## orsp  0.0000000 0.000e+00 0.000e+00 0.000e+00 0.000e+00 0.000e+00
## pala  0.0212814 -9.296e-03 3.074e-02 -1.243e-02 2.902e-02 2.380e-02
## peai  0.0697049 5.909e-02 -2.821e-03 5.278e-02 -3.971e-02 -2.331e-02
## pedu  0.0300964 -1.315e-02 4.347e-02 -1.758e-02 4.104e-02 3.366e-02
## phsu  0.0000000 0.000e+00 0.000e+00 0.000e+00 0.000e+00 0.000e+00
## plde  0.0000000 0.000e+00 0.000e+00 0.000e+00 0.000e+00 0.000e+00
## poar  0.0000000 0.000e+00 0.000e+00 0.000e+00 0.000e+00 0.000e+00
## poca  0.0311535 7.429e-02 1.806e-02 8.779e-02 2.715e-02 -2.008e-02
## pocap 0.0157620 1.774e-02 1.191e-02 -1.551e-02 -2.774e-02 9.443e-04
## poce  0.0000000 0.000e+00 0.000e+00 0.000e+00 0.000e+00 0.000e+00
## pogn  0.0000000 0.000e+00 0.000e+00 0.000e+00 0.000e+00 0.000e+00
## pole -0.0067798 6.543e-02 2.974e-02 8.404e-02 -4.950e-03 -2.006e-02
## pomu  0.1806915 -8.052e-02 -1.066e-01 1.495e-02 -1.385e-02 -5.198e-02
## pter  0.0000000 0.000e+00 0.000e+00 0.000e+00 0.000e+00 0.000e+00
## ptga -0.0265218 3.413e-02 8.198e-05 2.118e-02 4.241e-02 3.353e-02
## ptob  0.0000000 0.000e+00 0.000e+00 0.000e+00 0.000e+00 0.000e+00
## rhla  0.0114587 1.612e-02 -1.907e-02 1.253e-01 -6.133e-02 3.825e-02
## rhpy -0.0614885 1.909e-01 1.451e-02 3.551e-03 8.057e-02 -4.550e-02
## rhsp  0.0000000 0.000e+00 0.000e+00 0.000e+00 0.000e+00 0.000e+00

```

```

## ry      0.0000000 0.000e+00 0.000e+00 0.000e+00 0.000e+00 0.000e+00
## scna    0.0000000 0.000e+00 0.000e+00 0.000e+00 0.000e+00 0.000e+00
## sino    0.0000000 0.000e+00 0.000e+00 0.000e+00 0.000e+00 0.000e+00
## sool    0.0000000 0.000e+00 0.000e+00 0.000e+00 0.000e+00 0.000e+00
## stfi   -0.0217745 -2.451e-02 -1.645e-02 2.143e-02 3.832e-02 -1.304e-03
## stpi    0.0000000 0.000e+00 0.000e+00 0.000e+00 0.000e+00 0.000e+00
## thma    0.0000000 0.000e+00 0.000e+00 0.000e+00 0.000e+00 0.000e+00
## trcy   -0.2016923 4.678e-02 1.599e-01 -2.560e-02 -4.775e-02 -1.767e-02
## tris    0.0000000 0.000e+00 0.000e+00 0.000e+00 0.000e+00 0.000e+00
## tror   -0.1560795 -5.629e-02 -1.247e-02 -3.574e-02 -4.124e-02 -1.417e-02
## trpi   -0.0192367 -2.197e-03 9.150e-03 -9.411e-03 2.883e-03 -2.219e-02
## waac   -0.1168716 -1.021e-01 4.820e-02 -2.229e-02 7.287e-04 -5.756e-03
## wagr   -0.0763637 1.108e-02 -5.682e-02 -5.042e-02 -3.786e-02 2.435e-03
## x      -0.0563652 -6.808e-02 4.036e-02 -1.343e-02 3.574e-02 6.019e-02
##
##
## Site scores (weighted sums of species scores)
##
##          RDA1      PC1      PC2      PC3      PC4      PC5
## sit1   -0.4306 -0.15560  0.2593  0.4984 -0.43518  0.64698
## sit2    0.4306  0.15560 -0.2593 -0.4984  0.43518 -0.64698
## sit3   -0.5088 -0.54489  0.3874 -0.5081 -0.21987 -0.11475
## sit4    0.5088  0.54489 -0.3874  0.5081  0.21987  0.11475
## sit5   -0.3922 -0.04684  0.1950 -0.2006  0.06145 -0.47310
## sit6    0.3922  0.04684 -0.1950  0.2006 -0.06145  0.47310
## sit7   -0.5074  0.31689 -0.4561 -0.6141  0.14830  0.54414
## sit8    0.5074 -0.31689  0.4561  0.6141 -0.14830 -0.54414
## sit9   -0.5125  0.71285  0.5165  0.1813  0.28281 -0.12016
## sit10   0.5125 -0.71285 -0.5165 -0.1813 -0.28281  0.12016
## sit11   -0.2572  0.17912 -0.5923  0.2395 -0.55918 -0.45854
## sit12   0.2572 -0.17912  0.5923 -0.2395  0.55918  0.45854
## sit13   -0.2616 -0.46153 -0.3099  0.4036  0.72167 -0.02456
## sit14   0.2616  0.46153  0.3099 -0.4036 -0.72167  0.02456
##
##
## Site constraints (linear combinations of constraining variables)
##
##          RDA1      PC1      PC2      PC3      PC4      PC5
## con1   -0.41 -0.15560  0.2593  0.4984 -0.43518  0.64698
## con2    0.41  0.15560 -0.2593 -0.4984  0.43518 -0.64698
## con3   -0.41 -0.54489  0.3874 -0.5081 -0.21987 -0.11475
## con4    0.41  0.54489 -0.3874  0.5081  0.21987  0.11475
## con5   -0.41 -0.04684  0.1950 -0.2006  0.06145 -0.47310
## con6    0.41  0.04684 -0.1950  0.2006 -0.06145  0.47310
## con7   -0.41  0.31689 -0.4561 -0.6141  0.14830  0.54414
## con8    0.41 -0.31689  0.4561  0.6141 -0.14830 -0.54414
## con9   -0.41  0.71285  0.5165  0.1813  0.28281 -0.12016
## con10   0.41 -0.71285 -0.5165 -0.1813 -0.28281  0.12016
## con11   -0.41  0.17912 -0.5923  0.2395 -0.55918 -0.45854
## con12   0.41 -0.17912  0.5923 -0.2395  0.55918  0.45854
## con13   -0.41 -0.46153 -0.3099  0.4036  0.72167 -0.02456
## con14   0.41  0.46153  0.3099 -0.4036 -0.72167  0.02456
##
##

```

```

## Biplot scores for constraining variables
##
##          RDA1 PC1 PC2 PC3 PC4 PC5
## Yopen     1   0   0   0   0   0

RsquareAdj(trt_Frac)$adj.r.squared #explanatory power

## [1] 0.08471055

anova.cca(trt_Frac) ## this tells us if first condition, init, significantly contributes to overall var

## Permutation test for rda under reduced model
## Permutation: free
## Number of permutations: 999
##
## Model: rda(X = ass.rel.t0, Y = init, Z = block)
##          Df Variance      F Pr(>F)
## Model     1 0.035514 2.2097  0.06 .
## Residual  6 0.096430
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

# Extracting species scores and plotting
# Species scores
species.scores<-as.data.frame(vegan::scores(ass.rel.t0_NMS,"species")) ## some species don't have scores
species.scores$species<-rownames(species.scores)

### NMDS 1 and 2
log<-mds_scores_t0[mds_scores_t0$treatment == "log", ][chull(mds_scores_t0[mds_scores_t0$treatment ==
"log", c("NMDS1", "NMDS2")])]

open<-mds_scores_t0[mds_scores_t0$treatment == "open", ][chull(mds_scores_t0[mds_scores_t0$treatment ==
"open", c("NMDS1", "NMDS2")])]

hulldat<-rbind(log,open)

nmds.plot.sp <- ggplot()+
  theme_bw()+
  theme(panel.background = element_blank(),
        panel.grid.major = element_blank(), #remove major-grid labels
        panel.grid.minor = element_blank(), #remove minor-grid labels
        plot.background = element_blank(),
        axis.text = element_text(size = 10),
        axis.title=element_text(size=15),
        legend.title=element_text(size=15),
        legend.text=element_text(size=10))+
  geom_text_repel(data=species.scores, aes(NMDS1, NMDS2, label=species), alpha=0.9, size=5, col='darkgreen')+
  geom_polygon(data=hulldat, aes(NMDS1, NMDS2, fill=treatment, group=treatment), alpha=0.3)+#fill the hulls
  scale_fill_manual(values=c("#63A088", "#56638A"), name="Treatment")+
  geom_point(data=mds_scores_t0, aes(NMDS1, NMDS2, shape=block, col=treatment), size=6)+#add points
  scale_shape_manual(values = c(14,15,16,17,11,18,8), name='Block')+#change point shapes
  scale_colour_manual(values=c("#63A088", "#56638A"), name="Treatment")+

```

```

  labs(title=paste0("Stress: ", round(ass.rel.t0_NMS$stress,3)))

nmds.plot.nutrient <- ggplot()+
  theme_bw()+
  theme(panel.background = element_blank(),
        panel.grid.major = element_blank(), #remove major-grid labels
        panel.grid.minor = element_blank(), #remove minor-grid labels
        plot.background = element_blank(),
        axis.text = element_text(size = 10),
        axis.title=element_text(size=15),
        legend.title=element_text(size=15),
        legend.text=element_text(size=10))+

  geom_polygon(data=hulldat, aes(NMDS1, NMDS2, fill=treatment, group=treatment), alpha=0.3)+

  scale_fill_manual(values=c("#63A088", "#56638A"), name="Treatment")+
  geom_point(data=mds_scores_t0, aes(NMDS1, NMDS2, shape=block, col=treatment), size=6)+

  scale_shape_manual(values = c(14,15,16,17,11,18,8), name='Block')+
  scale_colour_manual(values=c("#63A088", "#56638A"), name="Treatment")+
  geom_segment(aes(x = 0, y = 0, xend = NMDS1, yend = NMDS2),
               data = en_coord_cont, size =0.5, alpha = 0.5, colour = "grey30") +
  geom_text(data = en_coord_cont, aes(x = NMDS1, y = NMDS2), colour = "grey30",
            fontface = "bold", label = row.names(en_coord_cont))

```

```

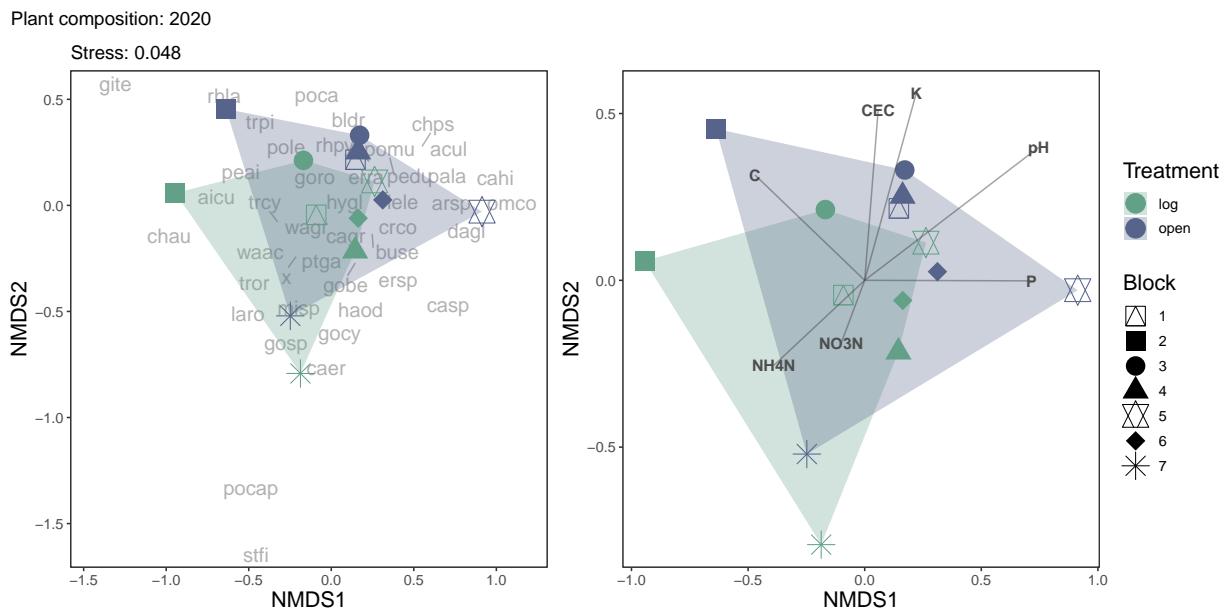
## Warning: Using 'size' aesthetic for lines was deprecated in ggplot2 3.4.0.
## i Please use 'linewidth' instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.

```

```

# put both nmds plots together
(nmds.plot.sp + theme(legend.position = "none")) + nmds.plot.nutrient + plot_layout(guides = "collect")

```



2021 - There is no significant correlation between nutrient elements and plant composition.

```

# Data wrangling
# This data set does not include unidentified species.
comm <- read.csv("20-22_species_composition_data_no_unk.csv", header = T)

# Remove the locations surveyed in 2021 and 2022 that were not surveyed in 2020 - aka, cm=0, cm=21, cm=22
comm<-comm[which(comm$cm_location!=21 & comm$cm_location!=0 & comm$cm_location!=22 & comm$cm_location!=20),]

# Make a group name for each row
comm$grp<-apply(comm[c(1,3,5,6,7)], 1, paste, collapse=":") # timepoint, block, transect_name, initial

# Need to make each row a community using matrify.
commsub<-comm[,c(15,10,13)] # group, species_code, and count of each species for each transect. transect
commsub <-as.data.frame(commsub)
commtry<-matrify(commsub) # make it an expanded species matrix

## Warning in matrify(commsub): NAs introduced by coercion

commtry$x[which(is.na(commtry$x))] <- 1 # "x" means that there were no individuals in the transect, but we still want to work with it
ncol(commtry) # how many species are we working with in our community matrix

## [1] 87

# Store grouping row names as a column, then remove rownames.
commtry$grps<-rownames(commtry)
rownames(commtry)<-NULL
names(commtry)

## [1] "acul"   "aicu"   "arca"   "ardy"   "arsp"   "auel"   "bldr"   "blrd"   "brdi"
## [10] "brdr"   "brpe"   "brru"   "buse"   "caer"   "cagr"   "cahi"   "casp"   "cear"
## [19] "chau"   "chei"   "chps"   "crcl"   "crc"    "cusc"   "cusp"   "dagl"   "dosp"
## [28] "ento"   "erau"   "ercy"   "erra"   "ersp"   "gite"   "gn"    "gobe"   "gocy"
## [37] "gono"   "goro"   "gosp"   "haod"   "hygl"   "hyp"    "hypo"   "jubu"   "laro"
## [46] "ledu"   "lele"   "loef"   "misp"   "mite"   "momo"   "mopa"   "niro"   "omco"
## [55] "or"    "pala"   "peai"   "pedu"   "phsu"   "plde"   "poar"   "poca"   "pocap"
## [64] "poce"   "pogn"   "pole"   "pomu"   "pter"   "ptga"   "ptob"   "rhla"   "rhp"
## [73] "rhsp"   "ry"    "scna"   "sino"   "sool"   "stfi"   "stpi"   "thma"   "trcy"
## [82] "tris"   "tror"   "trpi"   "waac"   "wagr"   "x"      "grps"

# Split group info into columns for each variable
mat<-separate(commtry, 88, c("time","block","transect","init","treatment"), ":")
names(mat) #check

## [1] "acul"      "aicu"      "arca"      "ardy"      "arsp"      "auel"      "bldr"      "blrd"      "brdi"
## [7] "bldr"      "blrd"      "brdi"      "brdr"      "brpe"      "brru"      "caer"      "casp"      "cear"
## [13] "buse"      "caer"      "cagr"      "cahi"      "casp"      "cear"      "chau"      "casc"      "casc"
## [19] "chau"      "chei"      "chps"      "crcl"      "crc"       "cusc"      "cusp"      "dagl"      "dosp"
## [25] "cusp"      "dagl"      "dosp"      "ento"      "erau"      "ercy"      "erra"      "ersp"      "gite"
## [31] "erra"      "ersp"      "gite"      "gn"        "gobe"      "gocy"      "gono"      "goro"      "gosp"
## [37] "gono"      "goro"      "gosp"      "haod"      "hygl"      "hyp"       "hypo"      "jubu"      "laro"
## [43] "hypo"      "jubu"      "laro"      "ledu"      "lele"      "loef"      "misp"      "mite"      "momo"
## [49] "misp"      "mite"      "momo"      "mopa"      "niro"      "omco"

```

```

## [55] "orsp"      "pala"       "peai"       "pedu"       "phsu"       "plde"
## [61] "poar"       "poca"       "pocap"      "poce"       "pogn"       "pole"
## [67] "pomu"       "pter"       "ptga"       "ptob"       "rhla"       "rhpyp"
## [73] "rhsp"       "ry"         "scna"       "sino"       "sool"       "stfi"
## [79] "stpi"       "thma"       "trcy"       "tris"       "tror"       "trpi"
## [85] "waac"       "wagr"       "x"          "time"       "block"      "transect"
## [91] "init"       "treatment"

# Set up nutrient analysis data
nutrient <- read.csv("Nutrient.csv", header = T)
nutrient <- separate(nutrient, 2, c("block", "plot"), "_")
nutrient <- nutrient[, 2:17]

# subset data where all t0 communities, insitu log and insitu open communities at t1 and t2 are included
mat2 <- mat[which(mat$time=="t1" & mat$treatment=="open" | mat$time=="t1" & mat$treatment=="insitu_log")]
mat2$grp<-apply(mat2[c(89, 91)], 1, paste, collapse=":") # block, init as grouping
names(mat2) #check

## [1] "acul"      "aicu"       "arca"       "ardy"       "arsp"       "auel"
## [7] "bldr"      "blrd"       "brdi"       "brdr"       "brpe"       "brru"
## [13] "buse"      "caer"       "cagr"       "cahi"       "casp"       "cear"
## [19] "chau"      "chei"       "chs"        "crcl"       "crc"        "cusc"
## [25] "cusp"      "dagl"       "dosp"       "ento"       "erau"       "ercy"
## [31] "erra"      "ersp"       "gite"       "gnte"       "gobe"       "gocy"
## [37] "gono"      "goro"       "gosp"       "haod"       "hygl"       "hypi"
## [43] "hypo"      "jubu"       "laro"       "ledu"       "lele"       "loef"
## [49] "misp"      "mite"       "momo"       "mopa"       "niro"       "omco"
## [55] "orsp"      "pala"       "peai"       "pedu"       "phsu"       "plde"
## [61] "poar"      "poca"       "pocap"      "poce"       "pogn"       "pole"
## [67] "pomu"      "pter"       "ptga"       "ptob"       "rhla"       "rhpyp"
## [73] "rhsp"      "ry"         "scna"       "sino"       "sool"       "stfi"
## [79] "stpi"      "thma"       "trcy"       "tris"       "tror"       "trpi"
## [85] "waac"      "wagr"       "x"          "time"       "block"      "transect"
## [91] "init"      "treatment" "grp"

# another df where the grouping variables are time, block and initial state
# each row is a transect in a certain year.
df<-mat2[,c(1:87, 93)]
df2 = df %>% mutate(across(.cols=1:87, .fns=as.numeric)) # make everything numeric
rownames(df2)<-NULL # remove rownames

## new with group vars
nublock<-separate(df2, 88, c("block", "init"), ":") # just looking at time, block & initial treatment

# want to sum across transects in same block X init treatment
nublock$sumgrp<-apply(mat2[c(89, 91)], 1, paste, collapse=":")
head(nublock)

## acul aicu arca ardy arsp auel bldr blrd brdi brdr brpe brru buse caer cagr
## 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
## 2 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
## 3 0 0 0 0 0 0 0 0 0 0 0 0 0 0 2 2
## 4 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
```

```

## 5   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
## 6   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
##   cahi casp clear chau chei chps crcl crcs cusc cusp dagl dosp ento erau ercy
## 1   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
## 2   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   1
## 3   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
## 4   0   0   0   0   0   0   0   0   3   0   0   0   0   0   0   0
## 5   0   0   0   0   0   0   0   1   0   0   0   0   0   0   0   0
## 6   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
##   erra ersp gite gnte gobe gocy gono goro gosp haad hygl hypi hypo jubu laro
## 1   0   0   0   0   0   0   0   0   1   0   0   0   0   0   0   0
## 2   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
## 3   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
## 4   0   0   0   0   2   0   0   0   0   0   0   0   0   0   0   0
## 5   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
## 6   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
##   ledu lele loef misp mite momo mopa niro omco orsp pala peai pedu phsu plde
## 1   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
## 2   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
## 3   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
## 4   0   0   0   1   0   0   0   0   0   1   0   0   0   0   0   0
## 5   0   0   0   0   0   0   0   0   0   0   0   0   0   3   0   0
## 6   0   0   0   3   0   0   0   0   0   0   0   0   0   0   0   0
##   poar poca pocap poce pogn pole pomu pter ptga ptob rhla rhpy rhsp ry scna
## 1   0   0   0   0   0   1   5   0   0   0   0   0   0   0   0   0
## 2   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
## 3   1   0   0   0   0   1   0   0   0   0   0   0   0   0   0   0
## 4   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
## 5   0   0   0   0   0   0   5   0   0   0   0   0   0   0   0   0
## 6   0   0   0   0   0   0   6   0   0   0   0   0   0   0   0   0
##   sino sool stfi stpi thma trcy tris tror trpi waac wagr x block init sumgrp
## 1   0   0   0   0   0   0   0   0   0   0   0   0   0   0   1   log 1:log
## 2   0   0   0   0   0   0   0   0   1   0   2   0   0   1   log 1:log
## 3   0   0   0   0   0   0   0   0   0   0   0   0   0   1   log 1:log
## 4   0   0   0   0   0   0   0   0   0   0   0   0   0   1   open 1:open
## 5   0   0   0   0   0   0   0   0   0   0   0   0   0   1   open 1:open
## 6   0   0   0   0   0   0   0   0   0   0   0   0   0   1   log 1:log

```

```

# sum observations across initial X block (group variable)
# this gives number of plants in each transect TYPE for each year in each block. should be 2 types X 3
blocksum<-rowsum(nublock[,c(1:87)], group=nublock$sumgrp)
blocksum$grps<-rownames(blocksum)
rownames(blocksum)<-NULL # remove rownames
nrow(blocksum) # it is 14 rows as expected

```

```
## [1] 14
```

```

## expand again
blocksum<-separate(blocksum, 88, c("block", "init"), ":")
nutrient_join <- nutrient[,c(1,3,5:9, 11,16)]
blocksum <- inner_join(blocksum, nutrient_join, by = c("init", "block"))

# at the moment this includes where there were no plants ("x" column in matrix)

```

```

assemblies_t1<-blocksum[,c(1:87)]  
  

# group - these are the treatment variables that need to be separately fed into the MDS analysis from  

group_init<-blocksum$init  

group_block<-blocksum$block  

group_nutrient<-blocksum[,c(90:96)]  
  

# MDS  

ass.rel.t1<-decostand(assemblies_t1, method='hel') #standardize assemblies  

ass.rel.t1_NMS <- metaMDS(ass.rel.t1, distance = 'bray', k = 4) # run MDS  
  

## Run 0 stress 0.07021466  

## Run 1 stress 0.07021459  

## ... New best solution  

## ... Procrustes: rmse 0.0001921847 max resid 0.0003927085  

## ... Similar to previous best  

## Run 2 stress 0.07697581  

## Run 3 stress 0.07697602  

## Run 4 stress 0.0702147  

## ... Procrustes: rmse 0.0001678941 max resid 0.0002864871  

## ... Similar to previous best  

## Run 5 stress 0.0764755  

## Run 6 stress 0.09908857  

## Run 7 stress 0.07697605  

## Run 8 stress 0.08007202  

## Run 9 stress 0.07021474  

## ... Procrustes: rmse 0.0002692495 max resid 0.0005455116  

## ... Similar to previous best  

## Run 10 stress 0.07021468  

## ... Procrustes: rmse 0.0002630642 max resid 0.0004605901  

## ... Similar to previous best  

## Run 11 stress 0.08043304  

## Run 12 stress 0.07697583  

## Run 13 stress 0.07021465  

## ... Procrustes: rmse 0.0001154851 max resid 0.0001922762  

## ... Similar to previous best  

## Run 14 stress 0.07021457  

## ... New best solution  

## ... Procrustes: rmse 5.93016e-05 max resid 0.0001076682  

## ... Similar to previous best  

## Run 15 stress 0.07647601  

## Run 16 stress 0.07697591  

## Run 17 stress 0.0764761  

## Run 18 stress 0.07021468  

## ... Procrustes: rmse 0.0001688486 max resid 0.0002902134  

## ... Similar to previous best  

## Run 19 stress 0.07021461  

## ... Procrustes: rmse 9.522104e-05 max resid 0.0001725009  

## ... Similar to previous best  

## Run 20 stress 0.07021468  

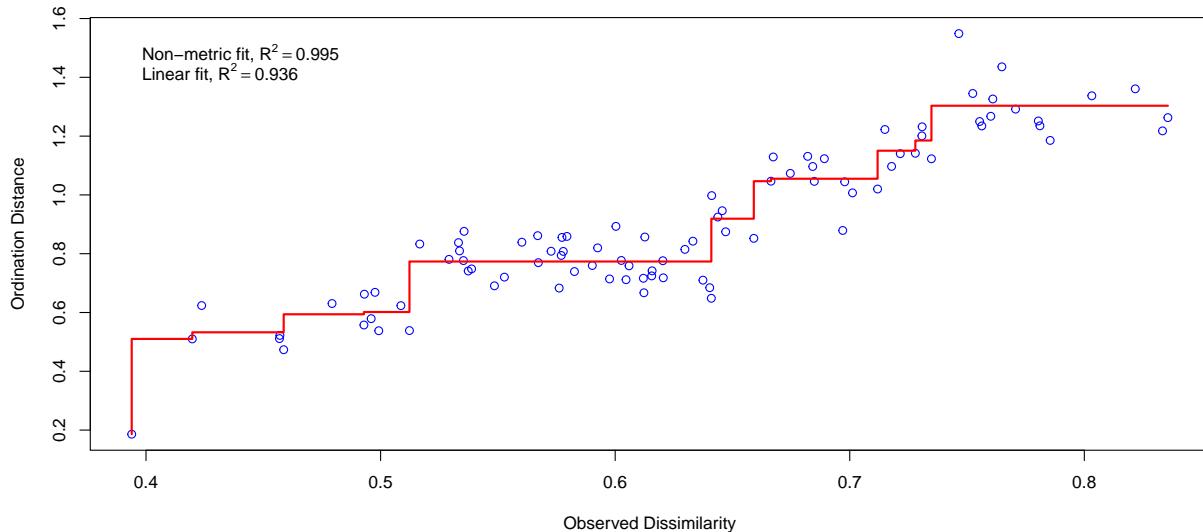
## ... Procrustes: rmse 0.0002052549 max resid 0.0003778684  

## ... Similar to previous best  

## *** Best solution repeated 4 times

```

```
stressplot(ass.rel.t1_NMS) # check fit
```



```
en.nutrient = envfit(ass.rel.t1_NMS, group_nutrient, permutations = 999, na.rm = TRUE)
# plot(ass.rel.t1_NMS)
# plot(en.nutrient)
print(en.nutrient) ##### it is basically saying most of the nutrient does not affect the composition

## 
## ***VECTORS
##
##          NMDS1      NMDS2      r2 Pr(>r)
## NH4N    0.94222 -0.33499 0.2070  0.289
## NO3N    0.05022 -0.99874 0.2152  0.290
## P       -0.86867  0.49539 0.1385  0.427
## K       -0.75127  0.66000 0.1268  0.470
## C       0.84943 -0.52770 0.0169  0.921
## pH     -0.92568 -0.37831 0.3009  0.131
## CEC    -0.88034 -0.47435 0.0770  0.652
## Permutation: free
## Number of permutations: 999

# scores
mds_scores_t1<-as.data.frame(vegan::scores(ass.rel.t1_NMS)$sites) # extract scores
mds_scores_t1$site<-rownames(vegan::scores(ass.rel.t1_NMS)$sites) # extract names
mds_scores_t1$treatment<-group_init # grouping factor 1
mds_scores_t1$block<-group_block # grouping factor 2
en_coord_cont = as.data.frame(vegan::scores(en.nutrient, "vectors")) * ordiArrowMul(en.nutrient)

# explaining factors
init<-as.factor(group_init) # grouping factor 1- convert to factor
block<-as.factor(group_block) # grouping factor 2- convert to factor
```

```
#### rda model analysis & results #####
# can look at significance of model where initial treatment and block explain variation in community
trt_tot_2<-rda(ass.rel.t1~init+block) # run model using standardized data
summary(trt_tot_2)
```

```
##
## Call:
## rda(formula = ass.rel.t1 ~ init + block)
##
## Partitioning of variance:
##           Inertia Proportion
## Total      0.5326    1.0000
## Constrained 0.3506    0.6582
## Unconstrained 0.1821    0.3418
##
## Eigenvalues, and their contribution to the variance
##
## Importance of components:
##          RDA1   RDA2   RDA3   RDA4   RDA5   RDA6   RDA7
## Eigenvalue      0.09831 0.07256 0.0594 0.04249 0.03790 0.02137 0.01854
## Proportion Explained 0.18458 0.13623 0.1115 0.07977 0.07115 0.04012 0.03482
## Cumulative Proportion 0.18458 0.32081 0.4323 0.51209 0.58325 0.62337 0.65818
##          PC1     PC2     PC3     PC4     PC5     PC6
## Eigenvalue      0.05497 0.03785 0.02947 0.02799 0.02046 0.01132
## Proportion Explained 0.10320 0.07107 0.05533 0.05254 0.03842 0.02125
## Cumulative Proportion 0.76139 0.83246 0.88779 0.94033 0.97875 1.00000
##
## Accumulated constrained eigenvalues
## Importance of components:
##          RDA1   RDA2   RDA3   RDA4   RDA5   RDA6   RDA7
## Eigenvalue      0.09831 0.07256 0.0594 0.04249 0.0379 0.02137 0.01854
## Proportion Explained 0.28043 0.20698 0.1694 0.12119 0.1081 0.06096 0.05290
## Cumulative Proportion 0.28043 0.48742 0.6568 0.77804 0.8861 0.94710 1.00000
##
## Scaling 2 for species and site scores
## * Species are scaled proportional to eigenvalues
## * Sites are unscaled: weighted dispersion equal on all dimensions
## * General scaling constant of scores: 1.62215
##
##
## Species scores
##
##          RDA1   RDA2   RDA3   RDA4   RDA5   RDA6
## acul -9.964e-02 4.823e-02 -2.106e-02 1.277e-01 1.875e-02 -7.699e-02
## aicu 8.907e-17 -5.123e-17 8.326e-17 6.638e-17 -2.673e-17 1.400e-17
## arca 2.296e-17 9.720e-17 -3.409e-17 4.292e-17 -4.580e-17 -1.032e-17
## ardy -4.306e-02 1.526e-03 -1.411e-03 -1.678e-02 -4.191e-02 -3.620e-02
## arsp -2.461e-17 4.760e-17 3.101e-17 -6.840e-18 4.536e-17 3.563e-17
## auel 6.616e-17 -4.835e-17 -1.821e-17 -4.209e-18 1.394e-17 2.789e-17
## bldr 4.844e-03 6.118e-03 9.056e-02 -8.132e-02 1.733e-01 9.784e-02
## blrd -4.559e-02 4.352e-02 -9.443e-03 1.226e-02 4.624e-02 8.270e-03
## brdi -3.456e-17 4.651e-17 3.643e-17 1.080e-17 -2.048e-17 7.466e-18
## brdr 4.095e-18 2.731e-18 7.681e-18 7.980e-18 -1.132e-19 7.206e-18
```

```

## brpe  3.360e-02 -9.795e-02  1.335e-01 -1.875e-02  3.238e-02 -2.280e-02
## brru -3.432e-34 -5.429e-35  1.158e-33 -1.156e-34 -2.427e-34 -5.492e-34
## buse -2.227e-32  6.989e-33  3.781e-32 -7.980e-33 -1.210e-32 -2.143e-32
## caer -1.023e-01 -6.797e-04  3.047e-02  8.609e-02 -1.012e-01 -7.420e-02
## cagr  5.897e-02 -1.425e-01  4.363e-02  4.572e-02 -6.406e-02  6.086e-02
## cahi  0.000e+00  0.000e+00  0.000e+00  0.000e+00  0.000e+00  0.000e+00
## casp -3.404e-02  3.541e-04  4.467e-03  5.540e-02 -1.265e-02  1.676e-02
## cear  4.342e-02 -1.362e-02 -7.371e-02  1.556e-02  2.358e-02  4.177e-02
## chau  1.161e-01  1.608e-01 -1.308e-01 -4.803e-02 -5.147e-02 -5.783e-03
## chei  0.000e+00  0.000e+00  0.000e+00  0.000e+00  0.000e+00  0.000e+00
## chps -2.651e-01  1.313e-01 -8.485e-02  7.546e-02  7.404e-02  2.339e-02
## crcl  0.000e+00  0.000e+00  0.000e+00  0.000e+00  0.000e+00  0.000e+00
## crc0 -3.108e-02  7.252e-02 -7.821e-02  1.545e-02 -6.801e-02  9.685e-02
## cusc  0.000e+00  0.000e+00  0.000e+00  0.000e+00  0.000e+00  0.000e+00
## cusp -9.546e-02 -1.913e-02  3.033e-02 -5.482e-02  1.679e-02 -8.577e-02
## dagl  0.000e+00  0.000e+00  0.000e+00  0.000e+00  0.000e+00  0.000e+00
## dosp  0.000e+00  0.000e+00  0.000e+00  0.000e+00  0.000e+00  0.000e+00
## ento  0.000e+00  0.000e+00  0.000e+00  0.000e+00  0.000e+00  0.000e+00
## erau  0.000e+00  0.000e+00  0.000e+00  0.000e+00  0.000e+00  0.000e+00
## ercy -1.494e-02 -9.609e-02  7.175e-02 -2.555e-03  3.268e-02 -1.492e-02
## erra -2.518e-02  2.121e-02 -1.602e-02  3.898e-02  1.386e-02 -1.231e-02
## ersp  0.000e+00  0.000e+00  0.000e+00  0.000e+00  0.000e+00  0.000e+00
## gite  1.098e-01  6.866e-02  5.880e-02 -6.813e-03 -4.310e-02 -3.238e-02
## gntr  7.822e-02  1.004e-01 -1.078e-01 -3.810e-02 -2.965e-02 -4.746e-03
## gobe  5.311e-02 -9.669e-02 -1.053e-01 -7.205e-03  1.105e-03  5.703e-02
## gocy  0.000e+00  0.000e+00  0.000e+00  0.000e+00  0.000e+00  0.000e+00
## gono  0.000e+00  0.000e+00  0.000e+00  0.000e+00  0.000e+00  0.000e+00
## goro  1.802e-01 -2.829e-02 -2.024e-01 -1.678e-01 -3.322e-02  1.123e-02
## gosp  0.000e+00  0.000e+00  0.000e+00  0.000e+00  0.000e+00  0.000e+00
## haod  6.923e-03 -4.618e-02 -1.119e-02 -3.158e-02  8.584e-03 -4.117e-02
## hygl -2.255e-01  9.594e-02 -1.277e-01 -4.435e-02  1.571e-01 -3.412e-02
## hypi -8.499e-03 -3.399e-02  4.724e-02 -2.940e-03 -2.928e-02 -3.990e-02
## hypo  0.000e+00  0.000e+00  0.000e+00  0.000e+00  0.000e+00  0.000e+00
## jubu  0.000e+00  0.000e+00  0.000e+00  0.000e+00  0.000e+00  0.000e+00
## laro  6.166e-02 -8.661e-02 -2.159e-02 -1.090e-02  3.024e-02 -1.992e-02
## ledu -3.340e-02  3.252e-02  5.320e-02 -6.207e-02  8.923e-02  4.682e-02
## lele -1.116e-02 -8.439e-03 -3.421e-02 -2.560e-02  2.133e-03 -1.973e-02
## loef  0.000e+00  0.000e+00  0.000e+00  0.000e+00  0.000e+00  0.000e+00
## misp  3.447e-02 -1.375e-01 -7.412e-02  2.140e-01 -1.428e-02 -1.892e-02
## mite  0.000e+00  0.000e+00  0.000e+00  0.000e+00  0.000e+00  0.000e+00
## momo  0.000e+00  0.000e+00  0.000e+00  0.000e+00  0.000e+00  0.000e+00
## mopa  0.000e+00  0.000e+00  0.000e+00  0.000e+00  0.000e+00  0.000e+00
## niro -6.420e-03  1.351e-03  6.208e-02 -3.434e-02  6.144e-02  2.506e-02
## omco -2.407e-02  7.323e-04 -1.139e-02  1.863e-03 -3.286e-02  3.819e-02
## orsp  0.000e+00  0.000e+00  0.000e+00  0.000e+00  0.000e+00  0.000e+00
## pala  0.000e+00  0.000e+00  0.000e+00  0.000e+00  0.000e+00  0.000e+00
## peai  2.708e-02  4.611e-02 -6.180e-02  1.195e-01  7.353e-02  5.837e-02
## pedu  0.000e+00  0.000e+00  0.000e+00  0.000e+00  0.000e+00  0.000e+00
## phsu -4.540e-03  9.554e-04  4.390e-02 -2.428e-02  4.344e-02  1.772e-02
## plde  0.000e+00  0.000e+00  0.000e+00  0.000e+00  0.000e+00  0.000e+00
## poar  3.129e-02  5.243e-02  1.267e-02 -3.757e-03 -7.738e-02  3.673e-02
## poc0  0.000e+00  0.000e+00  0.000e+00  0.000e+00  0.000e+00  0.000e+00
## pocap 0.000e+00  0.000e+00  0.000e+00  0.000e+00  0.000e+00  0.000e+00
## poce  0.000e+00  0.000e+00  0.000e+00  0.000e+00  0.000e+00  0.000e+00

```

```

## pogn  0.000e+00  0.000e+00  0.000e+00  0.000e+00  0.000e+00  0.000e+00
## pole -1.059e-01  1.918e-01  2.540e-01  5.600e-02 -4.763e-02  3.381e-02
## pomu -3.790e-01 -2.547e-01 -2.332e-02 -1.004e-01 -9.950e-02  4.492e-02
## pter  0.000e+00  0.000e+00  0.000e+00  0.000e+00  0.000e+00  0.000e+00
## ptga  4.191e-03  1.826e-01 -6.538e-02  4.060e-02 -6.820e-02 -9.489e-04
## ptob  0.000e+00  0.000e+00  0.000e+00  0.000e+00  0.000e+00  0.000e+00
## rhla  0.000e+00  0.000e+00  0.000e+00  0.000e+00  0.000e+00  0.000e+00
## rhpyp 7.480e-02 -1.733e-01 -5.570e-02  7.898e-03  3.486e-02 -8.805e-02
## rhsp  0.000e+00  0.000e+00  0.000e+00  0.000e+00  0.000e+00  0.000e+00
## ry    0.000e+00  0.000e+00  0.000e+00  0.000e+00  0.000e+00  0.000e+00
## scna  4.567e-02  4.156e-02 -4.560e-02 -1.815e-03 -7.993e-03  1.495e-02
## sino  0.000e+00  0.000e+00  0.000e+00  0.000e+00  0.000e+00  0.000e+00
## sool  6.923e-03 -4.618e-02 -1.119e-02 -3.158e-02  8.584e-03 -4.117e-02
## stfi  0.000e+00  0.000e+00  0.000e+00  0.000e+00  0.000e+00  0.000e+00
## stpi  0.000e+00  0.000e+00  0.000e+00  0.000e+00  0.000e+00  0.000e+00
## thma  0.000e+00  0.000e+00  0.000e+00  0.000e+00  0.000e+00  0.000e+00
## trcy  1.901e-01 -1.943e-02  1.190e-01 -8.643e-03  1.539e-01 -5.244e-02
## tris -1.016e-02  3.179e-02 -1.165e-01  6.991e-02  1.015e-01 -5.902e-02
## tror  1.207e-01 -1.229e-01 -4.525e-02  1.878e-01  6.974e-02  8.917e-02
## trpi  1.553e-01  9.710e-02  8.316e-02 -9.635e-03 -6.095e-02 -4.579e-02
## waac  7.512e-02 -1.338e-02  6.506e-02  9.716e-04 -2.574e-02  9.506e-02
## wagr -4.651e-02  3.022e-02 -2.281e-02 -7.188e-02  5.822e-02  1.591e-02
## x     -2.407e-02  7.323e-04 -1.139e-02  1.863e-03 -3.286e-02  3.819e-02
##
##
## Site scores (weighted sums of species scores)
##
##          RDA1      RDA2      RDA3      RDA4      RDA5      RDA6
## row1  -0.14539 -0.50546  0.2599  0.216437 -0.91881  0.6085
## row2  -0.46559  0.09913 -0.1111 -0.007202 -0.30677  0.7744
## row3   0.80185  0.50763  0.5725 -0.332829 -0.22459 -0.2324
## row4   0.46163  0.92808 -0.3131  0.097676 -0.47194 -0.1184
## row5   0.14415  0.14924  0.7768 -0.434518  0.85157  0.3859
## row6  -0.63217  0.31773  0.0867 -0.509643  0.52588  0.3249
## row7  -0.26308 -0.04549  0.1058  0.779425  0.30886 -0.1555
## row8  -0.41799  0.46761 -0.1560  0.932183  0.32017 -0.4631
## row9   0.09703 -0.32699  0.8313  0.266940 -0.29875 -0.6554
## row10 -0.64372 -0.07686 -0.2627 -0.475378 -0.49259 -0.2084
## row11 -0.10067 -0.63812 -0.1985 -0.433033  0.20807 -0.9544
## row12 -0.05046 -0.16595 -0.6994 -0.550396 -0.04626 -0.0293
## row13  0.64528 -0.66391 -0.2563  0.413154  0.31358  0.5796
## row14  0.56913 -0.04664 -0.6360  0.037186  0.23158  0.1437
##
##
## Site constraints (linear combinations of constraining variables)
##
##          RDA1      RDA2      RDA3      RDA4      RDA5      RDA6
## row1  -0.13704 -0.420750  0.37318  0.17256 -0.57852  0.6309
## row2  -0.47394  0.014419 -0.22438  0.03668 -0.64707  0.7520
## row3   0.80019  0.500273  0.42845 -0.04964 -0.31399 -0.2359
## row4   0.46329  0.935442 -0.16911 -0.18552 -0.38254 -0.1149
## row5  -0.07556  0.015900  0.73055 -0.40414  0.72300  0.2949
## row6  -0.41246  0.451069  0.13299 -0.54002  0.65445  0.4159
## row7  -0.17209 -0.006526  0.27370  0.92374  0.34879 -0.3698

```

```

## row8 -0.50899 0.428644 -0.32386 0.78786 0.28024 -0.2488
## row9 -0.10489 -0.419510 0.58308 -0.03628 -0.36139 -0.4924
## row10 -0.44180 0.015660 -0.01448 -0.17216 -0.42994 -0.3714
## row11 0.09289 -0.619621 -0.15016 -0.42378 0.11518 -0.5524
## row12 -0.24402 -0.184451 -0.74771 -0.55965 0.04663 -0.4313
## row13 0.77566 -0.572860 -0.14735 0.29311 0.30686 0.3011
## row14 0.43875 -0.137690 -0.74491 0.15723 0.23831 0.4221
##
##
## Biplot scores for constraining variables
##
##          RDA1     RDA2     RDA3     RDA4     RDA5     RDA6
## initopen -0.38855  0.5019 -0.68916 -0.15671 -0.07906  0.1396
## block2    0.59489  0.6760  0.12211 -0.11072 -0.32795 -0.1652
## block3   -0.22977  0.2199  0.40658 -0.44454  0.64855  0.3347
## block4   -0.32067  0.1987 -0.02361  0.80588  0.29617 -0.2912
## block5   -0.25740 -0.1901  0.26772 -0.09814 -0.37259 -0.4067
## block6   -0.07116 -0.3786 -0.42275 -0.46303  0.07619 -0.4632
## block7    0.57178 -0.3346 -0.42011  0.21204  0.25668  0.3405
##
##
## Centroids for factor constraints
##
##          RDA1     RDA2     RDA3     RDA4     RDA5     RDA6
## initlog   0.16845 -0.2176  0.29878  0.06794  0.03428 -0.06053
## initopen  -0.16845  0.2176 -0.29878 -0.06794 -0.03428  0.06053
## block1   -0.30549 -0.2032  0.07440  0.10462 -0.61279  0.69144
## block2    0.63174  0.7179  0.12967 -0.11758 -0.34827 -0.17539
## block3   -0.24401  0.2335  0.43177 -0.47208  0.68872  0.35539
## block4   -0.34054  0.2111 -0.02508  0.85580  0.31451 -0.30929
## block5   -0.27335 -0.2019  0.28430 -0.10422 -0.39567 -0.43190
## block6   -0.07556 -0.4020 -0.44893 -0.49171  0.08091 -0.49186
## block7    0.60720 -0.3553 -0.44613  0.22517  0.27258  0.36161

```

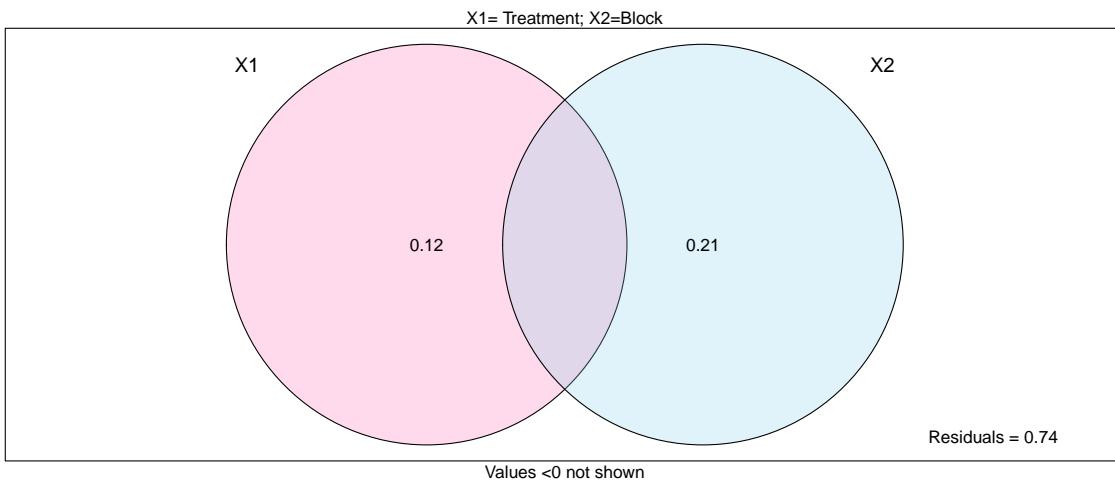
```
anova.cca(trt_tot_2, step=1000, by="term") ## test for model significance
```

```

## Permutation test for rda under reduced model
## Terms added sequentially (first to last)
## Permutation: free
## Number of permutations: 999
##
## Model: rda(formula = ass.rel.t1 ~ init + block)
##          Df Variance      F Pr(>F)
## init      1 0.064359 2.121  0.005 **
## block     6 0.286205 1.572  0.005 **
## Residual  6 0.182059
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

```

```
# can model using varpart to look at contributions of initial treatment and block
var.mod2<-varpart(ass.rel.t1, init, block) # run model on standardized data
plot(var.mod2, bg=c("hotpink","skyblue"))
mtext("X1= Treatment; X2=Block", side=3)
```



```
## can test for significance of contribution of the fraction of initial treatment
# do this with partial redundancy analysis
trt_Frac<-rda(ass.rel.t1, init, block) # partial rda model
summary(trt_Frac)
```

```
##
## Call:
## rda(X = ass.rel.t1, Y = init, Z = block)
##
## Partitioning of variance:
##           Inertia Proportion
## Total      0.53262   1.0000
## Conditioned 0.28621   0.5373
## Constrained 0.06436   0.1208
## Unconstrained 0.18206   0.3418
##
## Eigenvalues, and their contribution to the variance
## after removing the contribution of conditiniong variables
##
## Importance of components:
##           RDA1     PC1     PC2     PC3     PC4     PC5     PC6
## Eigenvalue      0.06436 0.05497 0.03785 0.02947 0.02799 0.02046 0.01132
## Proportion Explained 0.26118 0.22307 0.15362 0.11959 0.11357 0.08304 0.04593
## Cumulative Proportion 0.26118 0.48425 0.63787 0.75745 0.87103 0.95407 1.00000
##
## Accumulated constrained eigenvalues
## Importance of components:
##           RDA1
## Eigenvalue      0.06436
## Proportion Explained 1.00000
## Cumulative Proportion 1.00000
##
## Scaling 2 for species and site scores
```

```

## * Species are scaled proportional to eigenvalues
## * Sites are unscaled: weighted dispersion equal on all dimensions
## * General scaling constant of scores: 1.62215
##
##
## Species scores
##
##          RDA1       PC1       PC2       PC3       PC4       PC5
## acul    0.055571  3.637e-02  4.328e-02  1.783e-02  5.246e-03  5.454e-04
## aicu    0.000000  8.472e-17 -2.496e-17 -1.813e-17  2.580e-17 -2.616e-17
## arca    0.000000  9.597e-18  2.861e-18  1.596e-17 -2.682e-18 -9.970e-19
## ardy    0.042259  5.025e-02  5.860e-02  1.652e-02  4.875e-02  2.875e-02
## arsp    0.000000  1.984e-17  3.886e-17  5.839e-17 -3.819e-17  1.599e-17
## auel    0.000000  9.860e-19 -4.413e-18  8.306e-18 -3.824e-18 -6.153e-18
## bldr   -0.039796 -4.819e-03  3.972e-02 -4.794e-02 -2.337e-02  2.293e-02
## blrd    0.042899  4.726e-03  1.765e-02  2.217e-02 -5.565e-02 -2.605e-02
## brdi    0.000000 -1.170e-17 -1.637e-17 -2.608e-17 -3.440e-18 -2.426e-17
## brdr    0.000000 -2.443e-19  4.135e-18 -1.727e-18 -3.647e-18  3.050e-18
## brpe   -0.128276 -8.636e-02 -6.995e-02 -9.422e-02 -6.120e-02 -1.120e-02
## brru    0.000000 -1.871e-34  5.118e-33 -4.301e-33 -3.629e-33  2.180e-33
## buse    0.000000  5.868e-34 -1.605e-32  1.349e-32  1.138e-32 -6.838e-33
## caer    0.011258 -7.733e-02  7.189e-02  4.870e-03  3.834e-02 -9.362e-02
## cagr   -0.095632 -8.997e-02 -1.221e-02  1.902e-02 -2.621e-02 -1.103e-01
## cahi    0.000000  0.000e+00  0.000e+00  0.000e+00  0.000e+00  0.000e+00
## casp   -0.003109  1.770e-02 -2.479e-02 -2.422e-02  2.065e-02  6.085e-02
## cear    0.042899  2.468e-03 -6.751e-02  5.673e-02  4.787e-02 -2.876e-02
## chau   0.140877 -1.835e-01 -1.649e-02  1.254e-02  8.452e-02  1.746e-02
## chei    0.000000  0.000e+00  0.000e+00  0.000e+00  0.000e+00  0.000e+00
## chps   0.213452  9.333e-02  4.706e-02 -1.968e-02 -1.078e-01  1.722e-02
## crcl    0.000000  0.000e+00  0.000e+00  0.000e+00  0.000e+00  0.000e+00
## crco   0.138108  1.593e-01 -8.706e-02  4.536e-02 -1.939e-02  3.021e-02
## cusc    0.000000  0.000e+00  0.000e+00  0.000e+00  0.000e+00  0.000e+00
## cusp   0.023897  4.130e-03  2.468e-02  9.589e-02 -9.788e-02  9.619e-03
## dagl    0.000000  0.000e+00  0.000e+00  0.000e+00  0.000e+00  0.000e+00
## dosp    0.000000  0.000e+00  0.000e+00  0.000e+00  0.000e+00  0.000e+00
## ento    0.000000  0.000e+00  0.000e+00  0.000e+00  0.000e+00  0.000e+00
## erau    0.000000  0.000e+00  0.000e+00  0.000e+00  0.000e+00  0.000e+00
## ercy   -0.129183 -3.188e-02  1.257e-02  8.821e-02  5.864e-02  3.511e-02
## erra    0.021450 -2.405e-03 -1.984e-03  4.725e-03 -3.054e-02 -2.041e-02
## ersp    0.000000  0.000e+00  0.000e+00  0.000e+00  0.000e+00  0.000e+00
## gite   -0.059503  1.330e-01 -2.802e-02 -9.877e-04 -2.136e-02 -4.809e-02
## gnte   0.105088 -1.192e-01 -1.940e-02  3.043e-03  6.651e-02 -1.381e-03
## gobe   0.009616  2.117e-02  6.289e-02 -1.833e-01 -6.378e-02  4.242e-02
## gocy    0.000000  0.000e+00  0.000e+00  0.000e+00  0.000e+00  0.000e+00
## gono    0.000000  0.000e+00  0.000e+00  0.000e+00  0.000e+00  0.000e+00
## goro   0.087805  8.971e-02  6.801e-02 -8.319e-02  1.119e-01 -7.619e-02
## gosp    0.000000  0.000e+00  0.000e+00  0.000e+00  0.000e+00  0.000e+00
## haod   -0.032312 -2.971e-03 -4.026e-03  6.188e-02 -2.109e-02  4.142e-02
## hygl   0.215609  4.905e-02  1.609e-03  6.009e-02  1.333e-02  5.224e-02
## hypi   -0.035126 -4.177e-02 -4.871e-02 -1.374e-02 -4.052e-02 -2.390e-02
## hypo    0.000000  0.000e+00  0.000e+00  0.000e+00  0.000e+00  0.000e+00
## jubu    0.000000  0.000e+00  0.000e+00  0.000e+00  0.000e+00  0.000e+00
## laro   -0.062906 -4.731e-03  4.412e-02  2.142e-02 -5.523e-02  6.194e-02
## ledu    0.004284  1.424e-03  3.922e-03  3.485e-03 -5.014e-03 -1.127e-03

```

```

## lele  0.019834  1.824e-03  2.471e-03 -3.798e-02  1.295e-02 -2.543e-02
## loef  0.000000  0.000e+00  0.000e+00  0.000e+00  0.000e+00  0.000e+00
## misp -0.077168 -1.527e-01  1.772e-02 -2.016e-02 -2.886e-02 -4.694e-02
## mite  0.000000  0.000e+00  0.000e+00  0.000e+00  0.000e+00  0.000e+00
## momo  0.000000  0.000e+00  0.000e+00  0.000e+00  0.000e+00  0.000e+00
## mopa  0.000000  0.000e+00  0.000e+00  0.000e+00  0.000e+00  0.000e+00
## niro -0.036841 -1.225e-02 -3.373e-02 -2.997e-02  4.312e-02  9.689e-03
## omco  0.022017  1.488e-02 -2.711e-02 -1.869e-02 -1.512e-02  3.695e-02
## orsp  0.000000  0.000e+00  0.000e+00  0.000e+00  0.000e+00  0.000e+00
## pala  0.000000  0.000e+00  0.000e+00  0.000e+00  0.000e+00  0.000e+00
## peai  0.031786 -6.717e-02 -6.703e-02 -9.866e-02  1.006e-01  1.083e-01
## pedu  0.000000  0.000e+00  0.000e+00  0.000e+00  0.000e+00  0.000e+00
## phsu -0.026050 -8.660e-03 -2.385e-02 -2.119e-02  3.049e-02  6.851e-03
## plde  0.000000  0.000e+00  0.000e+00  0.000e+00  0.000e+00  0.000e+00
## poar  0.006522 -1.086e-01  5.808e-02  2.807e-02  3.614e-02 -2.284e-02
## poca  0.000000  0.000e+00  0.000e+00  0.000e+00  0.000e+00  0.000e+00
## pocap 0.000000  0.000e+00  0.000e+00  0.000e+00  0.000e+00  0.000e+00
## poce  0.000000  0.000e+00  0.000e+00  0.000e+00  0.000e+00  0.000e+00
## pogn  0.000000  0.000e+00  0.000e+00  0.000e+00  0.000e+00  0.000e+00
## pole -0.037141 -1.016e-01 -9.293e-02 -3.438e-02 -7.013e-02 -1.911e-02
## pomu  0.047780  9.595e-02  1.336e-01  5.875e-02  9.945e-02  8.749e-02
## pter  0.000000  0.000e+00  0.000e+00  0.000e+00  0.000e+00  0.000e+00
## ptga  0.126430 -1.396e-01  1.126e-03 -9.388e-03 -4.387e-02  5.596e-02
## ptob  0.000000  0.000e+00  0.000e+00  0.000e+00  0.000e+00  0.000e+00
## rhla  0.000000  0.000e+00  0.000e+00  0.000e+00  0.000e+00  0.000e+00
## rhpy -0.080545 -6.671e-02 -9.829e-02  1.253e-01 -1.163e-01  2.039e-02
## rhsp  0.000000  0.000e+00  0.000e+00  0.000e+00  0.000e+00  0.000e+00
## ry   0.000000  0.000e+00  0.000e+00  0.000e+00  0.000e+00  0.000e+00
## scna  0.043870 -4.887e-02 -2.320e-02  2.874e-02  3.198e-02  3.740e-03
## sino  0.000000  0.000e+00  0.000e+00  0.000e+00  0.000e+00  0.000e+00
## sool -0.032312 -2.971e-03 -4.026e-03  6.188e-02 -2.109e-02  4.142e-02
## stfi  0.000000  0.000e+00  0.000e+00  0.000e+00  0.000e+00  0.000e+00
## stpi  0.000000  0.000e+00  0.000e+00  0.000e+00  0.000e+00  0.000e+00
## thma  0.000000  0.000e+00  0.000e+00  0.000e+00  0.000e+00  0.000e+00
## trcy -0.192212  8.297e-02 -2.570e-01 -7.513e-02  4.924e-02  3.987e-03
## tris  0.061108 -4.018e-02 -3.765e-02  9.554e-02  4.144e-02  1.226e-02
## tror -0.092188 -2.616e-02  1.417e-01 -5.750e-02 -5.785e-02 -1.555e-02
## trpi -0.084150  1.881e-01 -3.963e-02 -1.397e-03 -3.021e-02 -6.801e-02
## waac -0.075766  3.898e-02 -3.149e-02  5.714e-02  8.339e-02 -1.179e-01
## wagr  0.056986  1.417e-02  3.648e-02 -7.759e-03 -3.054e-02 -3.520e-02
## x    0.022017  1.488e-02 -2.711e-02 -1.869e-02 -1.512e-02  3.695e-02
##
## Site scores (weighted sums of species scores)
##
##          RDA1      PC1      PC2      PC3      PC4      PC5
## sit1 -0.3912 -0.29299  0.53388  0.367996  0.2978 -0.7276
## sit2  0.3912  0.29299 -0.53388 -0.367996 -0.2978  0.7276
## sit3 -0.4999  0.96887 -0.20417 -0.007197 -0.1556 -0.3504
## sit4  0.4999 -0.96887  0.20417  0.007197  0.1556  0.3504
## sit5 -0.5201 -0.14413 -0.39689 -0.352694  0.5075  0.1140
## sit6  0.5201  0.14413  0.39689  0.352694 -0.5075 -0.1140
## sit7 -0.2983  0.04860  0.04011 -0.095492  0.6173  0.4125
## sit8  0.2983 -0.04860 -0.04011  0.095492 -0.6173 -0.4125

```

```

## sit9 -0.6722 -0.51555 -0.60119 -0.169521 -0.5001 -0.2950
## sit10 0.6722 0.51555 0.60119 0.169521 0.5001 0.2950
## sit11 -0.3244 -0.03987 -0.05401 0.830224 -0.2830 0.5558
## sit12 0.3244 0.03987 0.05401 -0.830224 0.2830 -0.5558
## sit13 -0.3287 -0.02494 0.68227 -0.573317 -0.4838 0.2906
## sit14 0.3287 0.02494 -0.68227 0.573317 0.4838 -0.2906
##
##
## Site constraints (linear combinations of constraining variables)
##
##          RDA1      PC1      PC2      PC3      PC4      PC5
## con1   -0.4335 -0.29299  0.53388  0.367996  0.2978 -0.7276
## con2    0.4335  0.29299 -0.53388 -0.367996 -0.2978  0.7276
## con3   -0.4335  0.96887 -0.20417 -0.007197 -0.1556 -0.3504
## con4    0.4335 -0.96887  0.20417  0.007197  0.1556  0.3504
## con5   -0.4335 -0.14413 -0.39689 -0.352694  0.5075  0.1140
## con6    0.4335  0.14413  0.39689  0.352694 -0.5075 -0.1140
## con7   -0.4335  0.04860  0.04011 -0.095492  0.6173  0.4125
## con8    0.4335 -0.04860 -0.04011  0.095492 -0.6173 -0.4125
## con9   -0.4335 -0.51555 -0.60119 -0.169521 -0.5001 -0.2950
## con10   0.4335  0.51555  0.60119  0.169521  0.5001  0.2950
## con11  -0.4335 -0.03987 -0.05401  0.830224 -0.2830 0.5558
## con12   0.4335  0.03987  0.05401 -0.830224  0.2830 -0.5558
## con13  -0.4335 -0.02494  0.68227 -0.573317 -0.4838 0.2906
## con14   0.4335  0.02494 -0.68227  0.573317 0.4838 -0.2906
##
##
## Biplot scores for constraining variables
##
##          RDA1 PC1 PC2 PC3 PC4 PC5
## Yopen     1   0   0   0   0   0
```

```
RsquareAdj(trt_Frac)$adj.r.squared #explanatory power
```

```
## [1] 0.1186067
```

```
anova.cca(trt_Frac) ## this tells us if first condition, init, significantly contributes to overall var
```

```

## Permutation test for rda under reduced model
## Permutation: free
## Number of permutations: 999
##
## Model: rda(X = ass.rel.t1, Y = init, Z = block)
##          Df Variance      F Pr(>F)
## Model      1 0.064359 2.121  0.063 .
## Residual   6 0.182059
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
### extracting species scores and plotting
```

```
# species scores
```

```
species.scores<-as.data.frame(vegan::scores(ass.rel.t1_NMS,"species")) ## some species don't have scores
```

```

species.scores$species<-rownames(species.scores)

### NMDS 1 and 2
log<-mds_scores_t1[mds_scores_t1$treatment == "log", ][chull(mds_scores_t1[mds_scores_t1$treatment == "log", c("NMDS1", "NMDS2")])]

open<-mds_scores_t1[mds_scores_t1$treatment == "open", ][chull(mds_scores_t1[mds_scores_t1$treatment == "open", c("NMDS1", "NMDS2")])]

hulldat<-rbind(log,open)

nmuds.plot.sp <- ggplot()+
  theme_bw()+
  theme(panel.background = element_blank(),
        panel.grid.major = element_blank(), #remove major-grid labels
        panel.grid.minor = element_blank(), #remove minor-grid labels
        plot.background = element_blank(),
        axis.text = element_text(size = 15),
        axis.title=element_text(size=20),
        legend.title=element_text(size=20),
        legend.text=element_text(size=15))+

  geom_text_repel(data=species.scores, aes(NMDS1, NMDS2, label=species), alpha=0.9, size=5, col='darkgreen')+
  geom_polygon(data=hulldat, aes(NMDS1, NMDS2, fill=treatment, group=treatment), alpha=0.3)+scale_fill_manual(values=c("#63A088", "#56638A"), name="Treatment")+
  geom_point(data=mds_scores_t1, aes(NMDS1, NMDS2, shape=block, col=treatment), size=6)+ scale_shape_manual(values=c(14,15,16,17,11,18,8), name='Block')+scale_colour_manual(values=c("#63A088", "#56638A"), name="Treatment")+
  labs(title=paste0("Stress: ", round(ass.rel.t1_NMS$stress,3)))

nmuds.plot.nutrient <- ggplot()+
  theme_bw()+
  theme(panel.background = element_blank(),
        panel.grid.major = element_blank(), #remove major-grid labels
        panel.grid.minor = element_blank(), #remove minor-grid labels
        plot.background = element_blank(),
        axis.text = element_text(size = 15),
        axis.title=element_text(size=20),
        legend.title=element_text(size=20),
        legend.text=element_text(size=15))+

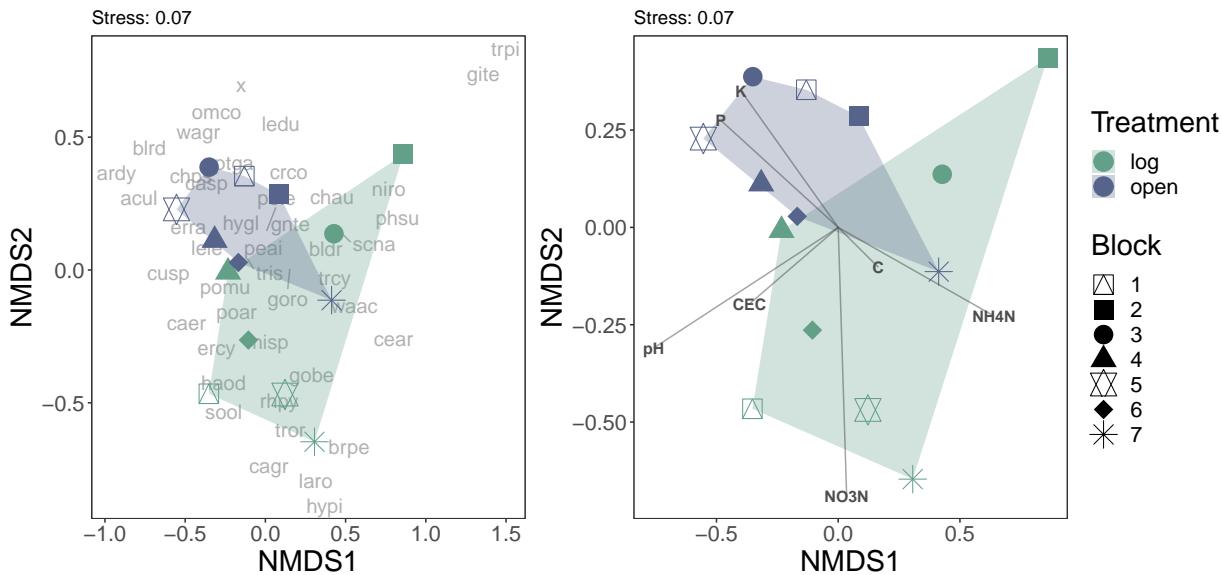
  geom_polygon(data=hulldat, aes(NMDS1, NMDS2, fill=treatment, group=treatment), alpha=0.3)+scale_fill_manual(values=c("#63A088", "#56638A"), name="Treatment")+
  geom_point(data=mds_scores_t1, aes(NMDS1, NMDS2, shape=block, col=treatment), size=6)+scale_shape_manual(values=c(14,15,16,17,11,18,8), name='Block')+scale_colour_manual(values=c("#63A088", "#56638A"), name="Treatment")+
  geom_segment(aes(x = 0, y = 0, xend = NMDS1, yend = NMDS2),
               data = en_coord_cont, size =0.5, alpha = 0.5, colour = "grey30") +
  geom_text(data = en_coord_cont, aes(x = NMDS1, y = NMDS2), colour = "grey30",
            fontface = "bold", label = row.names(en_coord_cont))+

  labs(title=paste0("Stress: ", round(ass.rel.t1_NMS$stress,3)))

(nmuds.plot.sp + theme(legend.position = "none")) + nmuds.plot.nutrient + plot_layout(guides = "collect")

```

Plant composition: 2020



2022 - There is no significant correlation between nutrient elements and plant composition.

```
# Data wrangling
# This data set does not include unknown data.
comm <- read.csv("20-22_species_composition_data_no_unk.csv", header = T)

# Remove the locations surveyed in 2021 and 2022 that were not surveyed in 2020 - aka, cm=0, cm=21, cm=22
comm<-comm[which(comm$cm_location!=21 & comm$cm_location!=0 & comm$cm_location!=22 & comm$cm_location!=20),]

# Make a group name for each row
comm$grp<-apply(comm[c(1,3,5,6,7)], 1, paste, collapse=":") # timepoint, block, transect_name, initial

# Need to make each row a community using matrify.
commsub<-comm[,c(15,10,13)] # group, species_code, and count of each species for each transect. transect
commsub <-as.data.frame(commsub)
commtry<-matrify(commsub) # make it an expanded species matrix
commtry$x[x[which(is.na(commtry$x))]] <- 1 # "x" means that there were no individuals in the transect, but
ncol(commtry) # how many species are we working with in our community matrix
```

```
## [1] 87
```

```
# Store grouping row names as a column, then remove rownames.
commtry$grps<-rownames(commtry)
rownames(commtry)<-NULL
names(commtry)
```

```
## [1] "acul"   "aicu"   "arca"   "ardy"   "arsp"   "auel"   "bldr"   "blrd"   "brdi"
## [10] "brdr"   "brpe"   "brru"   "buse"   "caer"   "cagr"   "cahi"   "casp"   "cear"
## [19] "chau"   "chei"   "chps"   "crcl"   "crc"    "cusc"   "cusp"   "dagl"   "dosp"
## [28] "ento"   "erau"   "ercy"   "erra"   "ersp"   "gite"   "gnte"   "gobe"   "gocy"
## [37] "gono"   "goro"   "gosp"   "haod"   "hygl"   "hyp"    "hypo"   "jubu"   "laro"
## [46] "ledu"   "lele"   "loef"   "misp"   "mite"   "momo"   "mopa"   "niro"   "omco"
## [55] "orsp"   "pala"   "peai"   "pedu"   "phsu"   "plde"   "poar"   "poca"   "pocap"
```

```

## [64] "poce"   "pogn"   "pole"   "pomu"   "pter"   "ptga"   "ptob"   "rhla"   "rhp"
## [73] "rhsp"   "ry"     "scna"   "sino"   "sool"   "stfi"   "stpi"   "thma"   "trcy"
## [82] "tris"   "tror"   "trpi"   "waac"   "wagr"   "x"      "grps"

```

```

# Split group info into columns for each variable
mat<-separate(commtry, 88, c("time","block","transect","init","treatment"), ":")
names(mat) #check

```

```

##  [1] "acul"      "aicu"      "arca"      "ardy"      "arsp"      "auel"
##  [7] "bldr"      "blrd"      "brdi"      "brdr"      "brpe"      "brru"
## [13] "buse"      "caer"      "cagr"      "cahi"      "casp"      "cear"
## [19] "chau"      "chei"      "chs"       "crcl"      "crc"       "cusc"
## [25] "cusp"      "dagl"      "dosp"      "ento"      "erau"      "ercy"
## [31] "erra"      "ersp"      "gite"      "gntr"      "gobe"      "gocy"
## [37] "gono"      "goro"      "gosp"      "haod"      "hygl"      "hypi"
## [43] "hypo"      "jubu"      "laro"      "ledu"      "lele"      "loef"
## [49] "misp"      "mite"      "momo"      "mopa"      "niro"      "omco"
## [55] "orsp"      "pala"      "peai"      "pedu"      "phsu"      "plde"
## [61] "poar"      "poca"      "pocap"     "poce"      "pogn"      "pole"
## [67] "pomu"      "pter"      "ptga"      "ptob"      "rhla"      "rhp"
## [73] "rhsp"      "ry"       "scna"      "sino"      "sool"      "stfi"
## [79] "stpi"      "thma"      "trcy"      "tris"      "tror"      "trpi"
## [85] "waac"      "wagr"      "x"        "time"      "block"      "transect"
## [91] "init"      "treatment"

```

```

# Set up nutrient analysis data
nutrient <- read.csv("Nutrient.csv", header = T)
nutrient <-separate(nutrient,2 , c("block","plot"), "_")
nutrient <- nutrient[,2:17]

```

```

#####
# subset data where all t0 communities, insitu log and insitu open communities at t1 and t2 are included
mat2 <- mat[which(mat$time=="t2" & mat$treatment=="open" | mat$time=="t2" & mat$treatment=="insitu_log")]
mat2$grp<-apply(mat2[c(89,91)], 1, paste, collapse=":") # block, init as grouping
# names(mat2) #check

```

```

# another df where the grouping variables are time, block and initial state
# each row is a transect in a certain year.
df<-mat2[,c(1:87, 93)]

```

```

df2 = df %>% mutate(across(.cols=1:87,.fns=as.numeric)) # make everything numeric
rownames(df2)<-NULL # remove rownames

```

```

## new with group vars
nublock<-separate(df2, 88, c("block", "init"), ":") # just looking at time, block & initial treatment

```

```

# want to sum across transects in same block X init treatment
nublock$sumgrp<-apply(mat2[c(89, 91)], 1, paste, collapse=":")
# head(nublock)

```

```

# sum observations across initial X block (group variable)
# this gives number of plants in each transect TYPE for each year in each block. should be 2 types X 3
blocksum<-rowsum(nublock[,c(1:87)], group=nublock$sumgrp)
blocksum$grps<-rownames(blocksum)

```

```

rownames(blocksum) <-NULL # remove rownames
nrow(blocksum) # it is 14 rows as expected

## [1] 14

## expand again
blocksum<-separate(blocksum, 88, c("block", "init"), ":")
nutrient_join <- nutrient[,c(1,3,5:9, 11,16)]
blocksum <- inner_join(blocksum, nutrient_join, by = c("init", "block"))

# at the moment this includes where there were no plants ("x" column in matrix)
assemblies_t2<-blocksum[,c(1:87)]

# group - these are the treatment variables that need to be separately fed into the MDS analysis from
group_init<-blocksum$init
group_block<-blocksum$block
group_nutrient<-blocksum[,c(90:96)]

# MDS
ass.rel.t2<-decostand(assemblies_t2, method='hell') #standardize assemblies
ass.rel.t2_NMS <- metaMDS(ass.rel.t2, distance = 'bray', k = 4) # run MDS

## Run 0 stress 0.03974779
## Run 1 stress 0.03974784
## ... Procrustes: rmse 2.768082e-05 max resid 5.896309e-05
## ... Similar to previous best
## Run 2 stress 0.03974765
## ... New best solution
## ... Procrustes: rmse 0.0001249323 max resid 0.0002274742
## ... Similar to previous best
## Run 3 stress 0.03974769
## ... Procrustes: rmse 0.0003937509 max resid 0.0007501835
## ... Similar to previous best
## Run 4 stress 0.03974776
## ... Procrustes: rmse 0.0004632314 max resid 0.0008586074
## ... Similar to previous best
## Run 5 stress 0.03974782
## ... Procrustes: rmse 0.000148116 max resid 0.0002691826
## ... Similar to previous best
## Run 6 stress 0.03974758
## ... New best solution
## ... Procrustes: rmse 0.0001952323 max resid 0.0003621718
## ... Similar to previous best
## Run 7 stress 0.03974781
## ... Procrustes: rmse 0.0003357019 max resid 0.0006058614
## ... Similar to previous best
## Run 8 stress 0.03974762
## ... Procrustes: rmse 0.0001068829 max resid 0.0002402203
## ... Similar to previous best
## Run 9 stress 0.03974792
## ... Procrustes: rmse 0.0004003886 max resid 0.0007327718
## ... Similar to previous best

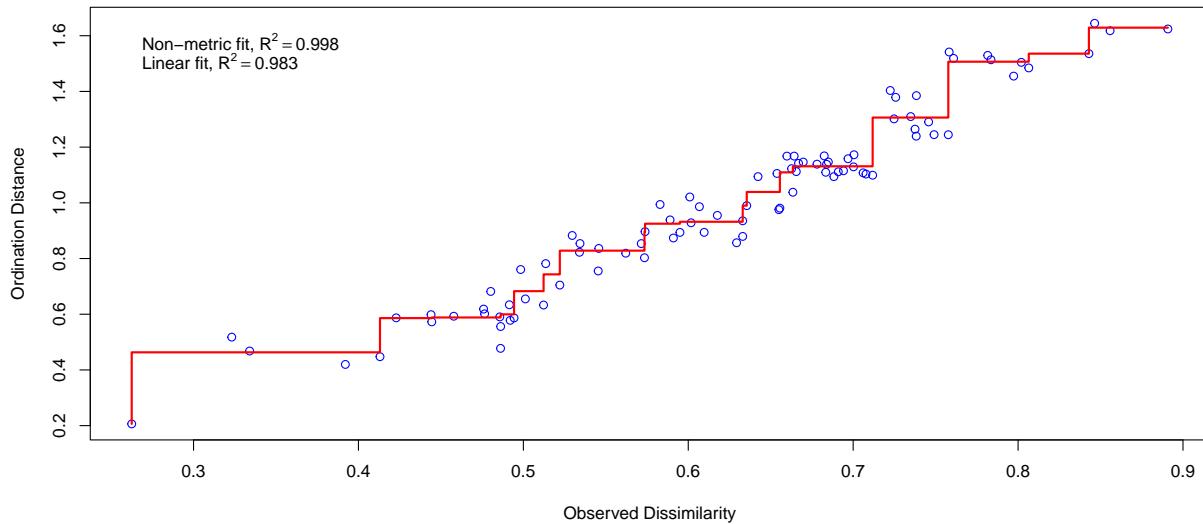
```

```

## Run 10 stress 0.03974775
## ... Procrustes: rmse 0.0002575656 max resid 0.0004779765
## ... Similar to previous best
## Run 11 stress 0.03974763
## ... Procrustes: rmse 0.000149822 max resid 0.0002776498
## ... Similar to previous best
## Run 12 stress 0.03974758
## ... Procrustes: rmse 5.270315e-05 max resid 9.88024e-05
## ... Similar to previous best
## Run 13 stress 0.03974761
## ... Procrustes: rmse 0.000145619 max resid 0.000279871
## ... Similar to previous best
## Run 14 stress 0.05590354
## Run 15 stress 0.03974814
## ... Procrustes: rmse 0.000397458 max resid 0.0007456196
## ... Similar to previous best
## Run 16 stress 0.03974779
## ... Procrustes: rmse 0.0003175446 max resid 0.0005755664
## ... Similar to previous best
## Run 17 stress 0.03974762
## ... Procrustes: rmse 0.0001482995 max resid 0.0002882441
## ... Similar to previous best
## Run 18 stress 0.03974773
## ... Procrustes: rmse 0.0002743997 max resid 0.0004996197
## ... Similar to previous best
## Run 19 stress 0.03974775
## ... Procrustes: rmse 0.0002533129 max resid 0.000510499
## ... Similar to previous best
## Run 20 stress 0.03974795
## ... Procrustes: rmse 0.0003957705 max resid 0.0007221767
## ... Similar to previous best
## *** Best solution repeated 14 times

```

```
stressplot(ass.rel.t2_NMS) # check fit
```



```

en.nutrient = envfit(ass.rel.t2_NMS, group_nutrient, permutations = 999, na.rm = TRUE)
# plot(ass.rel.t2_NMS)
# plot(en.nutrient)
print(en.nutrient) ##### it is basically saying most of the nutrient does not affect the composition

## 
## ***VECTORS
##
##          NMDS1      NMDS2      r2 Pr(>r)
## NH4N    0.26970 -0.96294 0.1829  0.352
## NO3N   -0.86640  0.49935 0.0636  0.706
## P       0.95878  0.28416 0.2378  0.215
## K       0.61448 -0.78893 0.0906  0.607
## C      -0.02114 -0.99978 0.3651  0.067 .
## pH     0.94764  0.31934 0.2138  0.257
## CEC    0.33526 -0.94213 0.1705  0.339
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## Permutation: free
## Number of permutations: 999

# we have created our NMS as ass.rel.t2_NMS
# scores
mds_scores_t2<-as.data.frame(vegan::scores(ass.rel.t2_NMS)$sites) # extract scores
mds_scores_t2$site<-rownames(vegan::scores(ass.rel.t2_NMS)$sites) # extract names
mds_scores_t2$treatment<-group_init # grouping factor 1
mds_scores_t2$block<-group_block # grouping factor 2
mds_scores_t2 <- cbind(mds_scores_t2, group_nutrient)
en_coord_cont = as.data.frame(vegan::scores(en.nutrient, "vectors")) * ordiArrowMul(en.nutrient)

# explaining factors
init<-as.factor(group_init) # grouping factor 1- convert to factor
block<-as.factor(group_block) # grouping factor 2- convert to factor

```

```

# nutri<-as.factor(group_nutrient) # this might be a mistake

##### rda model analysis & results #####
# can look at significance of model where initial treatment and block explain variation in community
trt_tot_2<-rda(ass.rel.t2~init+block) # run model using standardized data
summary(trt_tot_2)

##
## Call:
## rda(formula = ass.rel.t2 ~ init + block)
##
## Partitioning of variance:
##           Inertia Proportion
## Total      0.5340    1.0000
## Constrained 0.3581    0.6705
## Unconstrained 0.1759    0.3295
##
## Eigenvalues, and their contribution to the variance
##
## Importance of components:
##           RDA1     RDA2     RDA3     RDA4     RDA5     RDA6     RDA7
## Eigenvalue   0.1078  0.07881  0.05379  0.04086  0.03384  0.02612  0.01683
## Proportion Explained 0.2019  0.14758  0.10073  0.07651  0.06338  0.04891  0.03151
## Cumulative Proportion 0.2019  0.34951  0.45024  0.52675  0.59013  0.63904  0.67055
##           PC1     PC2     PC3     PC4     PC5     PC6
## Eigenvalue   0.0547  0.03572  0.03141  0.02309  0.01636  0.01464
## Proportion Explained 0.1024  0.06689  0.05882  0.04325  0.03064  0.02742
## Cumulative Proportion 0.7730  0.83988  0.89870  0.94194  0.97258  1.00000
##
## Accumulated constrained eigenvalues
## Importance of components:
##           RDA1     RDA2     RDA3     RDA4     RDA5     RDA6     RDA7
## Eigenvalue   0.1078  0.07881  0.05379  0.04086  0.03384  0.02612  0.01683
## Proportion Explained 0.3011  0.22009  0.15022  0.11410  0.09452  0.07293  0.04699
## Cumulative Proportion 0.3011  0.52124  0.67146  0.78556  0.88007  0.95301  1.00000
##
## Scaling 2 for species and site scores
## * Species are scaled proportional to eigenvalues
## * Sites are unscaled: weighted dispersion equal on all dimensions
## * General scaling constant of scores: 1.623197
##
##
## Species scores
##
##           RDA1     RDA2     RDA3     RDA4     RDA5     RDA6
## acul   9.052e-02 -6.166e-02 -2.465e-02  1.043e-01  1.350e-01  9.105e-02
## aicu   4.768e-17  4.052e-17 -9.248e-17 -1.056e-17 -1.453e-17 -3.522e-18
## arca  -5.475e-17 -2.441e-18 -4.889e-17  1.376e-17 -1.781e-17  5.141e-17
## ardy  -4.584e-02 -4.434e-02 -1.111e-02  1.969e-02 -2.105e-02 -1.788e-02
## arsp  -1.390e-17  6.109e-19 -1.670e-17  5.467e-17  3.549e-17  3.880e-17
## auel  -1.762e-17  3.587e-17  1.566e-17 -1.261e-17 -4.081e-17 -4.764e-17
## bldr   1.039e-01  4.551e-02 -1.034e-02  5.410e-02  3.353e-02 -5.643e-02
## blrd   1.876e-17 -2.147e-17  1.438e-18  1.943e-17  2.607e-17  1.106e-17

```

```

## brdi  1.852e-02  3.590e-02  8.983e-03  2.044e-02  1.514e-03 -4.940e-02
## brdr  6.717e-18 -2.173e-18  1.097e-18  1.249e-18  4.221e-18 -1.502e-17
## brpe  1.354e-17  2.728e-17  4.933e-18  1.483e-17  1.682e-19 -3.402e-17
## brru -7.711e-33  1.305e-32  2.283e-33  3.454e-33 -6.606e-33  1.260e-32
## buse  2.953e-33 -1.069e-32  2.015e-33 -2.537e-33  5.339e-33 -8.503e-33
## caer  1.272e-01 -6.922e-02  1.166e-01  7.445e-02  2.762e-02  4.229e-03
## cagr  0.000e+00  0.000e+00  0.000e+00  0.000e+00  0.000e+00  0.000e+00
## cahi  2.197e-02 -8.344e-02  7.037e-02  5.156e-02 -1.853e-02 -2.675e-02
## casp  0.000e+00  0.000e+00  0.000e+00  0.000e+00  0.000e+00  0.000e+00
## cear  0.000e+00  0.000e+00  0.000e+00  0.000e+00  0.000e+00  0.000e+00
## chau  0.000e+00  0.000e+00  0.000e+00  0.000e+00  0.000e+00  0.000e+00
## chei -6.027e-02  4.108e-02 -1.022e-01  5.789e-02 -9.473e-02  3.468e-02
## chps  1.422e-01  5.056e-02 -9.443e-02 -2.245e-02  2.895e-02 -2.191e-02
## crcl  7.114e-02 -4.012e-02  4.716e-02 -3.862e-02 -7.741e-02  8.715e-03
## crco  9.734e-02 -6.815e-02 -1.138e-01  5.228e-02  9.551e-02 -5.560e-03
## cusc  9.947e-02 -4.263e-02  1.155e-01 -3.845e-02 -1.031e-01 -3.646e-02
## cusp  0.000e+00  0.000e+00  0.000e+00  0.000e+00  0.000e+00  0.000e+00
## dagl  0.000e+00  0.000e+00  0.000e+00  0.000e+00  0.000e+00  0.000e+00
## dosp -1.315e-01 -1.124e-01 -1.082e-01  5.196e-02 -4.110e-02 -7.973e-02
## ento  0.000e+00  0.000e+00  0.000e+00  0.000e+00  0.000e+00  0.000e+00
## erau  3.620e-02 -3.512e-02  4.765e-02 -2.634e-02 -4.680e-02  5.787e-03
## ercy  4.416e-02 -5.421e-03 -7.098e-02 -1.592e-01  9.063e-02  1.565e-02
## erra  2.116e-02  3.187e-02 -5.398e-02 -1.259e-02  2.466e-02 -1.379e-02
## ersp  0.000e+00  0.000e+00  0.000e+00  0.000e+00  0.000e+00  0.000e+00
## gite -2.639e-02 -2.973e-02 -6.122e-02  2.699e-02 -3.764e-02 -2.222e-02
## gntr  0.000e+00  0.000e+00  0.000e+00  0.000e+00  0.000e+00  0.000e+00
## gobe -1.559e-01  1.913e-01  1.457e-01 -2.624e-03 -2.945e-02  7.403e-02
## gocy  0.000e+00  0.000e+00  0.000e+00  0.000e+00  0.000e+00  0.000e+00
## gono  0.000e+00  0.000e+00  0.000e+00  0.000e+00  0.000e+00  0.000e+00
## goro -8.288e-03  1.152e-01 -1.390e-01  5.549e-02 -1.156e-01  3.310e-02
## gosp  0.000e+00  0.000e+00  0.000e+00  0.000e+00  0.000e+00  0.000e+00
## haod -9.922e-04  1.618e-01 -1.355e-02 -1.014e-02  4.472e-02 -7.729e-02
## hygl  6.118e-02  9.387e-02  4.443e-02 -1.767e-02 -1.196e-02 -1.030e-01
## hypi  0.000e+00  0.000e+00  0.000e+00  0.000e+00  0.000e+00  0.000e+00
## hypo -1.494e-01 -1.129e-01  6.094e-02 -1.241e-01 -4.803e-02  2.488e-02
## jubu  0.000e+00  0.000e+00  0.000e+00  0.000e+00  0.000e+00  0.000e+00
## laro -5.349e-02 -2.490e-02  4.872e-02  4.252e-02 -7.231e-03 -7.957e-02
## ledu  0.000e+00  0.000e+00  0.000e+00  0.000e+00  0.000e+00  0.000e+00
## lele  0.000e+00  0.000e+00  0.000e+00  0.000e+00  0.000e+00  0.000e+00
## loef -2.155e-02 -2.427e-02 -4.998e-02  2.204e-02 -3.073e-02 -1.814e-02
## misp  0.000e+00  0.000e+00  0.000e+00  0.000e+00  0.000e+00  0.000e+00
## mite -1.261e-01  1.544e-01  1.114e-01 -9.659e-03  1.890e-01  3.377e-02
## momo  0.000e+00  0.000e+00  0.000e+00  0.000e+00  0.000e+00  0.000e+00
## mopa  0.000e+00  0.000e+00  0.000e+00  0.000e+00  0.000e+00  0.000e+00
## niro  0.000e+00  0.000e+00  0.000e+00  0.000e+00  0.000e+00  0.000e+00
## omco  0.000e+00  0.000e+00  0.000e+00  0.000e+00  0.000e+00  0.000e+00
## orsp -5.409e-03  1.374e-02  4.231e-02  1.614e-02  9.775e-03 -4.362e-02
## pala  0.000e+00  0.000e+00  0.000e+00  0.000e+00  0.000e+00  0.000e+00
## peai -1.076e-01  3.483e-01 -1.470e-01 -5.028e-02 -1.466e-02 -3.413e-02
## pedu  0.000e+00  0.000e+00  0.000e+00  0.000e+00  0.000e+00  0.000e+00
## phsu  0.000e+00  0.000e+00  0.000e+00  0.000e+00  0.000e+00  0.000e+00
## plde  0.000e+00  0.000e+00  0.000e+00  0.000e+00  0.000e+00  0.000e+00
## poar  0.000e+00  0.000e+00  0.000e+00  0.000e+00  0.000e+00  0.000e+00
## poca -7.330e-02  1.228e-02 -1.627e-02 -1.762e-01  1.119e-01 -9.665e-02

```

```

## pocap -9.137e-02 1.421e-02 -2.497e-02 4.681e-02 -6.872e-02 3.860e-02
## poce -5.409e-03 1.374e-02 4.231e-02 1.614e-02 9.775e-03 -4.362e-02
## pogn 0.000e+00 0.000e+00 0.000e+00 0.000e+00 0.000e+00 0.000e+00
## pole 5.618e-02 -1.627e-01 -9.388e-02 -1.303e-01 5.130e-02 8.606e-03
## pomu 4.919e-01 5.095e-02 2.181e-03 -4.628e-02 -5.128e-02 -4.981e-02
## pter 8.015e-02 4.723e-02 -2.287e-02 -4.203e-03 -2.753e-02 -5.322e-02
## ptga 1.310e-02 2.539e-02 6.352e-03 1.445e-02 1.071e-03 -3.493e-02
## ptob 0.000e+00 0.000e+00 0.000e+00 0.000e+00 0.000e+00 0.000e+00
## rhla 2.267e-02 -6.908e-02 4.303e-02 6.541e-02 8.281e-02 6.017e-02
## rhpy 1.001e-03 5.611e-02 1.010e-01 5.055e-02 2.293e-02 -1.325e-01
## rhsp -2.148e-01 -1.818e-01 -9.260e-02 -9.215e-02 -2.889e-02 -6.196e-02
## ry 0.000e+00 0.000e+00 0.000e+00 0.000e+00 0.000e+00 0.000e+00
## scna 0.000e+00 0.000e+00 0.000e+00 0.000e+00 0.000e+00 0.000e+00
## sino 3.620e-02 -3.512e-02 4.765e-02 -2.634e-02 -4.680e-02 5.787e-03
## sool 0.000e+00 0.000e+00 0.000e+00 0.000e+00 0.000e+00 0.000e+00
## stfi -5.409e-03 1.374e-02 4.231e-02 1.614e-02 9.775e-03 -4.362e-02
## stpi 0.000e+00 0.000e+00 0.000e+00 0.000e+00 0.000e+00 0.000e+00
## thma 0.000e+00 0.000e+00 0.000e+00 0.000e+00 0.000e+00 0.000e+00
## trcy -1.561e-01 -1.095e-01 2.729e-02 1.591e-01 4.816e-02 -1.420e-01
## tris 4.228e-02 -1.673e-02 -2.024e-02 4.110e-02 3.757e-02 3.527e-02
## tror -5.230e-02 8.443e-02 1.375e-01 8.368e-03 -4.656e-02 3.086e-02
## trpi 0.000e+00 0.000e+00 0.000e+00 0.000e+00 0.000e+00 0.000e+00
## waac -1.822e-02 -3.727e-02 1.688e-01 -1.391e-01 -3.077e-02 -1.163e-02
## wagr 0.000e+00 0.000e+00 0.000e+00 0.000e+00 0.000e+00 0.000e+00
## x 0.000e+00 0.000e+00 0.000e+00 0.000e+00 0.000e+00 0.000e+00
##
##
## Site scores (weighted sums of species scores)
##
##          RDA1      RDA2      RDA3      RDA4      RDA5      RDA6
## row1 -0.529954 -0.50054  0.2468 -1.10353  0.14120  0.29241
## row2  0.027932  0.29494 -0.3213 -0.55623  0.49222 -0.18686
## row3 -0.526506 -0.90064 -0.2668  0.49320 -0.41943 -0.57329
## row4 -0.521497 -0.16611 -0.6799  0.15480 -0.38435  0.01384
## row5  0.523959 -0.45922  0.7665 -0.33398 -1.07139  0.06484
## row6  0.751980 -0.09337 -0.1892 -0.28283 -0.23276  0.07611
## row7 -0.131864  0.32707 -0.1624 -0.14669  0.60202 -0.12836
## row8  0.300694  0.18578 -0.6404 -0.21465  0.18647 -0.22795
## row9 -0.083385 -0.56645  0.3037  0.69285  1.18053  0.55202
## row10 0.816213 -0.18975 -0.2365  0.35601 -0.04331  0.37932
## row11 0.004328  0.36280  0.7414  0.20897  0.15651 -0.81764
## row12 0.187756  0.43299  0.1371  0.38949  0.03776 -0.70925
## row13 -0.590590  0.63290  0.4722  0.06967 -0.07356  0.61728
## row14 -0.229066  0.63960 -0.1712  0.27293 -0.57192  0.64753
##
##
## Site constraints (linear combinations of constraining variables)
##
##          RDA1      RDA2      RDA3      RDA4      RDA5      RDA6
## row1 -0.44158 -0.26053  0.26294 -0.8470  0.39041  0.05381
## row2 -0.06044  0.05493 -0.33743 -0.8128  0.24301  0.05174
## row3 -0.71457 -0.69110 -0.17315  0.3069 -0.32819 -0.27869
## row4 -0.33343 -0.37564 -0.77353  0.3411 -0.47559 -0.28076
## row5  0.44740 -0.43402  0.58885 -0.3255 -0.57837  0.07151

```

```

## row6  0.82854 -0.11857 -0.01153 -0.2913 -0.72577  0.06944
## row7 -0.10616  0.09870 -0.10123 -0.1977  0.46794 -0.17712
## row8  0.27499  0.41415 -0.70160 -0.1636  0.32055 -0.17919
## row9  0.17584 -0.53583  0.33379  0.5074  0.64231  0.46671
## row10 0.55699 -0.22037 -0.26659  0.5415  0.49491  0.46464
## row11 -0.09453  0.24017  0.73940  0.2822  0.17083 -0.76241
## row12 0.28661  0.55562  0.13902  0.3163  0.02343 -0.76448
## row13 -0.60040  0.47852  0.45072  0.1542 -0.24904  0.63344
## row14 -0.21925  0.79398 -0.14966  0.1884 -0.39644  0.63137
##
##
## Biplot scores for constraining variables
##
##          RDA1     RDA2     RDA3     RDA4     RDA5     RDA6
## initopen  0.43929  0.3636 -0.69197  0.03936 -0.16988 -0.002386
## block2   -0.49312 -0.5019 -0.44544  0.30490 -0.37820 -0.263238
## block3    0.60037 -0.2600  0.27165 -0.29023 -0.61364  0.066324
## block4    0.07944  0.2413 -0.37775 -0.17002  0.37101 -0.167656
## block5    0.34482 -0.3558  0.03162  0.49352  0.53510  0.438225
## block6    0.09038  0.3744  0.41332  0.28159  0.09141 -0.718450
## block7   -0.38567  0.5987  0.14166  0.16120 -0.30372  0.595132
##
##
## Centroids for factor constraints
##
##          RDA1     RDA2     RDA3     RDA4     RDA5     RDA6
## initlog  -0.19057 -0.1577  0.30019 -0.01707  0.07370  0.001035
## initopen  0.19057  0.1577 -0.30019  0.01707 -0.07370 -0.001035
## block1   -0.25101 -0.1028 -0.03724 -0.82988  0.31671  0.052774
## block2   -0.52400 -0.5334 -0.47334  0.32400 -0.40189 -0.279725
## block3    0.63797 -0.2763  0.28866 -0.30841 -0.65207  0.070478
## block4    0.08441  0.2564 -0.40141 -0.18067  0.39425 -0.178156
## block5    0.36641 -0.3781  0.03360  0.52443  0.56861  0.465671
## block6    0.09604  0.3979  0.43921  0.29923  0.09713 -0.763447
## block7   -0.40983  0.6363  0.15053  0.17130 -0.32274  0.632406

```

```
anova.cca(trt_tot_2, step=1000, by="term") ## test for model significance
```

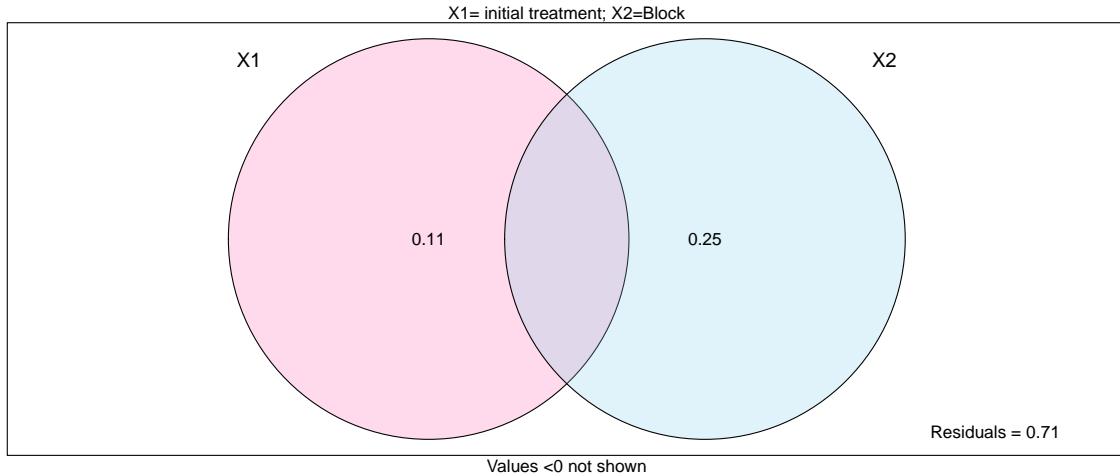
```

## Permutation test for rda under reduced model
## Terms added sequentially (first to last)
## Permutation: free
## Number of permutations: 999
##
## Model: rda(formula = ass.rel.t2 ~ init + block)
##          Df Variance      F Pr(>F)
## init      1  0.06081 2.0739  0.018 *
## block     6  0.29726 1.6897  0.001 ***
## Residual  6  0.17593
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ',' 1

```

```
# can model using varpart to look at contributions of initial treatment and block
var.mod2<-varpart(ass.rel.t2, init, block) # run model on standardized data
```

```
plot(var.mod2, bg=c("hotpink", "skyblue"))
mtext("X1= initial treatment; X2=Block", side=3)
```



```
## can test for significance of contribution of the fraction of initial treatment
# do this with partial redundancy analysis
trt_Frac<-rda(ass.rel.t2, init, block) # partial rda model
summary(trt_Frac)
```

```
##
## Call:
## rda(X = ass.rel.t2, Y = init, Z = block)
##
## Partitioning of variance:
##           Inertia Proportion
## Total      0.53400    1.0000
## Conditioned 0.29726    0.5567
## Constrained 0.06081    0.1139
## Unconstrained 0.17593    0.3295
##
## Eigenvalues, and their contribution to the variance
## after removing the contribution of conditining variables
##
## Importance of components:
##          RDA1     PC1     PC2     PC3     PC4     PC5     PC6
## Eigenvalue   0.06081 0.0547 0.03572 0.03141 0.02309 0.01636 0.01464
## Proportion Explained 0.25687 0.2311 0.15089 0.13267 0.09755 0.06910 0.06185
## Cumulative Proportion 0.25687 0.4879 0.63882 0.77149 0.86905 0.93815 1.00000
##
## Accumulated constrained eigenvalues
## Importance of components:
##          RDA1
## Eigenvalue   0.06081
```

```

## Proportion Explained 1.00000
## Cumulative Proportion 1.00000
##
## Scaling 2 for species and site scores
## * Species are scaled proportional to eigenvalues
## * Sites are unscaled: weighted dispersion equal on all dimensions
## * General scaling constant of scores: 1.623197
##
##
## Species scores
##
##          RDA1      PC1      PC2      PC3      PC4      PC5
## acul    0.020175 -2.142e-02 -8.157e-03 -2.565e-02 -6.560e-03 -2.267e-02
## aicu    0.000000 -4.876e-17 -4.519e-17 -3.653e-17  6.908e-18 -1.928e-17
## arca    0.000000 -8.942e-18  4.164e-17  3.659e-17  9.181e-18 -7.967e-18
## ardy    -0.027831 -4.756e-02 -3.944e-02  8.675e-03 -1.186e-02  5.108e-03
## arsp    0.000000 -2.014e-17 -2.380e-19 -1.365e-17  1.358e-17  1.535e-17
## auel    0.000000 -4.771e-18 -4.237e-19 -5.032e-18 -3.204e-18 -1.865e-17
## bldr    0.077576 -1.896e-02  1.384e-02  1.648e-02  6.044e-03  1.461e-01
## blrd    0.000000 -8.146e-18  9.663e-18  1.578e-17 -1.302e-17  4.303e-18
## brdi    0.028032  2.184e-03 -2.188e-02 -3.219e-02  1.751e-02  5.284e-02
## brdr    0.000000  5.211e-20 -9.460e-19 -2.257e-18  2.614e-18  9.147e-19
## brpe    0.000000 -7.947e-19  4.558e-18 -2.427e-19  1.126e-17 -2.201e-17
## brru    0.000000  6.061e-35 -3.019e-34  1.792e-34 -1.096e-33  1.716e-33
## buse    0.000000  1.148e-34 -5.719e-34  3.395e-34 -2.077e-33  3.251e-33
## caer   -0.044190  2.898e-02 -2.379e-02 -2.399e-02  2.885e-02 -2.967e-02
## cagr    0.000000  0.000e+00  0.000e+00  0.000e+00  0.000e+00  0.000e+00
## cahi   -0.067194 -7.302e-03 -1.008e-01 -7.009e-02  7.686e-02  2.540e-02
## casp    0.000000  0.000e+00  0.000e+00  0.000e+00  0.000e+00  0.000e+00
## cear    0.000000  0.000e+00  0.000e+00  0.000e+00  0.000e+00  0.000e+00
## chau    0.000000  0.000e+00  0.000e+00  0.000e+00  0.000e+00  0.000e+00
## chei    0.093970  8.115e-02  7.788e-02 -2.052e-02  5.365e-02 -6.052e-02
## chps    0.156906 -6.527e-02 -4.068e-02  4.561e-04 -4.431e-02  2.951e-02
## crcl   -0.016809 -6.644e-03  7.075e-04 -5.708e-03  3.885e-02 -5.185e-03
## crco    0.079849  2.340e-02 -9.206e-02  2.134e-02 -4.315e-02  4.745e-02
## cusc   -0.050462 -2.884e-02 -1.858e-02 -5.884e-02  1.989e-01  2.863e-02
## cusp    0.000000  0.000e+00  0.000e+00  0.000e+00  0.000e+00  0.000e+00
## dagl    0.000000  0.000e+00  0.000e+00  0.000e+00  0.000e+00  0.000e+00
## dosp   -0.053319 -2.094e-01 -1.574e-01 -2.948e-02 -5.637e-02 -3.437e-02
## ento    0.000000  0.000e+00  0.000e+00  0.000e+00  0.000e+00  0.000e+00
## erau   -0.035104 -1.388e-02  1.477e-03 -1.192e-02  8.114e-02 -1.083e-02
## ercy    0.046162 -4.595e-02  4.546e-02 -3.595e-03 -6.486e-02  1.494e-02
## erra    0.033378 -3.459e-02 -1.612e-02 -4.611e-02 -1.184e-02 -4.033e-02
## ersp    0.000000  0.000e+00  0.000e+00  0.000e+00  0.000e+00  0.000e+00
## gite    0.034332  5.867e-02  4.865e-02 -1.070e-02  1.463e-02 -6.301e-03
## gnre    0.000000  0.000e+00  0.000e+00  0.000e+00  0.000e+00  0.000e+00
## gobe   -0.070327 -2.086e-02  5.792e-02 -7.753e-02 -4.545e-02  7.220e-02
## gocy    0.000000  0.000e+00  0.000e+00  0.000e+00  0.000e+00  0.000e+00
## gono    0.000000  0.000e+00  0.000e+00  0.000e+00  0.000e+00  0.000e+00
## goro    0.192777 -2.313e-02  4.363e-02  2.883e-02  6.048e-02 -2.237e-02
## gosp    0.000000  0.000e+00  0.000e+00  0.000e+00  0.000e+00  0.000e+00
## haod    0.056933 -4.086e-02 -8.614e-02 -4.801e-02 -2.278e-02 -3.305e-03
## hygl    0.032852 -4.006e-02 -3.934e-02 -3.105e-02  9.185e-02 -7.827e-02
## hypi    0.000000  0.000e+00  0.000e+00  0.000e+00  0.000e+00  0.000e+00

```

```

## hypo -0.136055 -1.149e-01 -6.122e-02 -1.281e-02 2.658e-02 3.954e-02
## jubu 0.000000 0.000e+00 0.000e+00 0.000e+00 0.000e+00 0.000e+00
## laro -0.062935 -5.030e-02 -1.204e-02 4.898e-02 -3.378e-02 -6.106e-02
## ledu 0.000000 0.000e+00 0.000e+00 0.000e+00 0.000e+00 0.000e+00
## lele 0.000000 0.000e+00 0.000e+00 0.000e+00 0.000e+00 0.000e+00
## loef 0.028032 4.791e-02 3.972e-02 -8.738e-03 1.194e-02 -5.144e-03
## misp 0.000000 0.000e+00 0.000e+00 0.000e+00 0.000e+00 0.000e+00
## mite -0.113449 2.772e-01 -6.553e-02 2.380e-02 -5.406e-02 -9.228e-03
## momo 0.000000 0.000e+00 0.000e+00 0.000e+00 0.000e+00 0.000e+00
## mopa 0.000000 0.000e+00 0.000e+00 0.000e+00 0.000e+00 0.000e+00
## niro 0.000000 0.000e+00 0.000e+00 0.000e+00 0.000e+00 0.000e+00
## omco 0.000000 0.000e+00 0.000e+00 0.000e+00 0.000e+00 0.000e+00
## orsp -0.024822 -1.934e-03 1.938e-02 2.850e-02 -1.550e-02 -4.679e-02
## pala 0.000000 0.000e+00 0.000e+00 0.000e+00 0.000e+00 0.000e+00
## peai 0.142682 1.789e-01 -2.923e-02 1.335e-01 2.529e-02 2.227e-02
## pedu 0.000000 0.000e+00 0.000e+00 0.000e+00 0.000e+00 0.000e+00
## phsu 0.000000 0.000e+00 0.000e+00 0.000e+00 0.000e+00 0.000e+00
## plde 0.000000 0.000e+00 0.000e+00 0.000e+00 0.000e+00 0.000e+00
## poar 0.000000 0.000e+00 0.000e+00 0.000e+00 0.000e+00 0.000e+00
## poca -0.014943 8.686e-02 3.030e-02 8.356e-02 2.960e-02 3.563e-02
## pocap -0.004362 6.952e-02 4.738e-02 -7.135e-03 -1.505e-02 4.273e-02
## poce -0.024822 -1.934e-03 1.938e-02 2.850e-02 -1.550e-02 -4.679e-02
## pogn 0.000000 0.000e+00 0.000e+00 0.000e+00 0.000e+00 0.000e+00
## pole -0.026285 1.014e-01 -6.193e-02 -1.139e-01 -5.895e-02 -6.553e-02
## pomu 0.247365 -1.682e-01 6.404e-02 2.052e-01 -3.013e-02 -6.595e-03
## pter 0.073175 -7.558e-03 -3.227e-02 -5.003e-02 -4.913e-02 3.754e-02
## ptga 0.019822 1.544e-03 -1.547e-02 -2.276e-02 1.238e-02 3.736e-02
## ptob 0.000000 0.000e+00 0.000e+00 0.000e+00 0.000e+00 0.000e+00
## rhla -0.055929 8.999e-02 -7.097e-02 -6.003e-02 -1.719e-02 -3.763e-02
## rhpy -0.035682 -2.780e-03 2.785e-02 4.097e-02 -2.229e-02 -6.726e-02
## rhsp -0.056525 1.354e-02 1.937e-01 -1.522e-01 -3.834e-02 2.451e-02
## ry 0.000000 0.000e+00 0.000e+00 0.000e+00 0.000e+00 0.000e+00
## scna 0.000000 0.000e+00 0.000e+00 0.000e+00 0.000e+00 0.000e+00
## sino -0.035104 -1.388e-02 1.477e-03 -1.192e-02 8.114e-02 -1.083e-02
## sool 0.000000 0.000e+00 0.000e+00 0.000e+00 0.000e+00 0.000e+00
## stfi -0.024822 -1.934e-03 1.938e-02 2.850e-02 -1.550e-02 -4.679e-02
## stpi 0.000000 0.000e+00 0.000e+00 0.000e+00 0.000e+00 0.000e+00
## thma 0.000000 0.000e+00 0.000e+00 0.000e+00 0.000e+00 0.000e+00
## trcy -0.141493 8.767e-02 -1.282e-01 4.651e-02 -2.340e-02 3.479e-02
## tris 0.032930 -5.298e-02 4.179e-02 3.534e-02 1.012e-02 2.215e-02
## tror -0.121334 1.672e-02 4.009e-02 7.411e-02 5.603e-02 -1.665e-02
## trpi 0.000000 0.000e+00 0.000e+00 0.000e+00 0.000e+00 0.000e+00
## waac -0.130919 -3.799e-02 1.255e-01 -6.015e-02 -5.416e-02 -2.217e-02
## wagr 0.000000 0.000e+00 0.000e+00 0.000e+00 0.000e+00 0.000e+00
## x 0.000000 0.000e+00 0.000e+00 0.000e+00 0.000e+00 0.000e+00
##
##
## Site scores (weighted sums of species scores)
##
##          RDA1      PC1      PC2      PC3      PC4      PC5
## sit1   -0.5054 -0.21824  0.68560 -0.67128 -0.2526  0.14674
## sit2    0.5054  0.21824 -0.68560  0.67128  0.2526 -0.14674
## sit3   -0.3427 -0.74139 -0.61475  0.13523 -0.1848  0.07961
## sit4    0.3427  0.74139  0.61475 -0.13523  0.1848 -0.07961

```

```

## sit5 -0.4597 -0.17148 0.01826 -0.14731 1.0027 -0.13382
## sit6 0.4597 0.17148 -0.01826 0.14731 -1.0027 0.13382
## sit7 -0.2952 0.44952 0.20946 0.59934 0.1539 0.52410
## sit8 0.2952 -0.44952 -0.20946 -0.59934 -0.1539 -0.52410
## sit9 -0.7338 0.69798 -0.55047 -0.46560 -0.1334 -0.29185
## sit10 0.7338 -0.69798 0.55047 0.46560 0.1334 0.29185
## sit11 -0.3003 -0.03380 0.33862 0.49814 -0.2710 -0.81776
## sit12 0.3003 0.03380 -0.33862 -0.49814 0.2710 0.81776
## sit13 -0.3996 0.01741 -0.08672 0.05148 -0.3149 0.49297
## sit14 0.3996 -0.01741 0.08672 -0.05148 0.3149 -0.49297
##
##
## Site constraints (linear combinations of constraining variables)
##
##          RDA1      PC1      PC2      PC3      PC4      PC5
## con1   -0.4338 -0.21824 0.68560 -0.67128 -0.2526 0.14674
## con2    0.4338 0.21824 -0.68560 0.67128 0.2526 -0.14674
## con3   -0.4338 -0.74139 -0.61475 0.13523 -0.1848 0.07961
## con4    0.4338 0.74139 0.61475 -0.13523 0.1848 -0.07961
## con5   -0.4338 -0.17148 0.01826 -0.14731 1.0027 -0.13382
## con6    0.4338 0.17148 -0.01826 0.14731 -1.0027 0.13382
## con7   -0.4338 0.44952 0.20946 0.59934 0.1539 0.52410
## con8    0.4338 -0.44952 -0.20946 -0.59934 -0.1539 -0.52410
## con9   -0.4338 0.69798 -0.55047 -0.46560 -0.1334 -0.29185
## con10   0.4338 -0.69798 0.55047 0.46560 0.1334 0.29185
## con11   -0.4338 -0.03380 0.33862 0.49814 -0.2710 -0.81776
## con12   0.4338 0.03380 -0.33862 -0.49814 0.2710 0.81776
## con13   -0.4338 0.01741 -0.08672 0.05148 -0.3149 0.49297
## con14   0.4338 -0.01741 0.08672 -0.05148 0.3149 -0.49297
##
##
## Biplot scores for constraining variables
##
##          RDA1 PC1 PC2 PC3 PC4 PC5
## Yopen     1   0   0   0   0   0
```

```
RsquareAdj(trt_Frac)$adj.r.squared #explanatory power
```

```
## [1] 0.1095098
```

```
anova.cca(trt_Frac) ## this tells us if first condition, init, significantly contributes to overall var
```

```

## Permutation test for rda under reduced model
## Permutation: free
## Number of permutations: 999
##
## Model: rda(X = ass.rel.t2, Y = init, Z = block)
##          Df Variance      F Pr(>F)
## Model      1  0.06081 2.0739  0.065 .
## Residual   6  0.17593
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```

### extracting species scores and plotting
# species scores
species.scores.t2 <- as.data.frame(vegan::scores(ass.rel.t2_NMS,"species")) ## some species don't have scores
species.scores.t2$species<-rownames(species.scores.t2)

### NMDS 1 and 2
log<-mds_scores_t2[mds_scores_t2$treatment == "log", ][chull(mds_scores_t2[mds_scores_t2$treatment ==
"log", c("NMDS1", "NMDS2")])], ]

open<-mds_scores_t2[mds_scores_t2$treatment == "open", ][chull(mds_scores_t2[mds_scores_t2$treatment ==
"open", c("NMDS1", "NMDS2")])], ]

hulldat<-rbind(log,open)

nmuds.plot.sp <- ggplot()+
  theme_bw()+
  theme(panel.background = element_blank(),
        panel.grid.major = element_blank(), #remove major-grid labels
        panel.grid.minor = element_blank(), #remove minor-grid labels
        plot.background = element_blank(),
        axis.text = element_text(size = 15),
        axis.title=element_text(size=20),
        legend.title=element_text(size=20),
        legend.text=element_text(size=15))+

  geom_text_repel(data=species.scores.t2, aes(NMDS1, NMDS2, label=species), alpha=0.9, size=5, col='darkblue')
  geom_polygon(data=hulldat, aes(NMDS1, NMDS2, fill=treatment, group=treatment), alpha=0.3)+scale_fill_manual(values=c("#63A088", "#56638A"), name="Treatment")+
  geom_point(data=mds_scores_t2, aes(NMDS1, NMDS2, shape=block, col=treatment), size=6)+ scale_shape_manual(values=c("triangle-down", "triangle-up"))
  scale_colour_manual(values=c("#63A088", "#56638A"), name="Treatment")+
  labs(title=paste0("Stress: ", round(ass.rel.t2_NMS$stress,3)))

nmuds.plot.nutrient <- ggplot()+
  theme_bw()+
  theme(panel.background = element_blank(),
        panel.grid.major = element_blank(), #remove major-grid labels
        panel.grid.minor = element_blank(), #remove minor-grid labels
        plot.background = element_blank(),
        axis.text = element_text(size = 15),
        axis.title=element_text(size=20),
        legend.title=element_text(size=20),
        legend.text=element_text(size=15))+

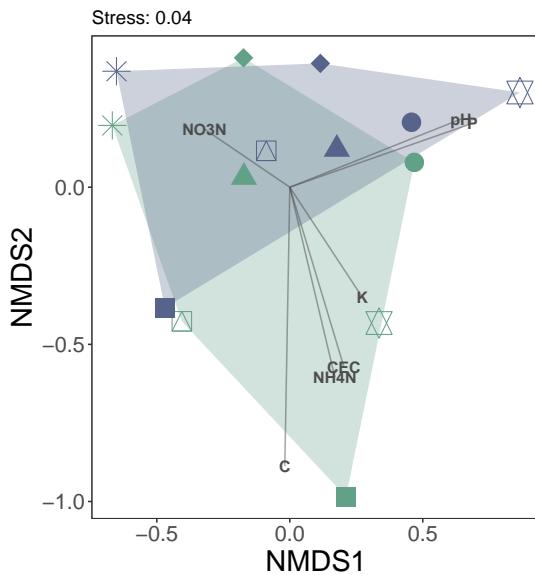
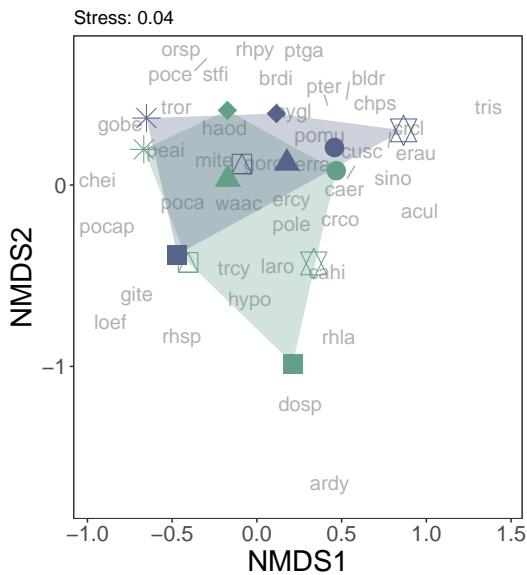
  geom_polygon(data=hulldat, aes(NMDS1, NMDS2, fill=treatment, group=treatment), alpha=0.3)+scale_fill_manual(values=c("#63A088", "#56638A"), name="Treatment")+
  geom_point(data=mds_scores_t2, aes(NMDS1, NMDS2, shape=block, col=treatment), size=6)+ scale_shape_manual(values=c("triangle-down", "triangle-up"))
  scale_colour_manual(values=c("#63A088", "#56638A"), name="Treatment")+
  geom_segment(aes(x = 0, y = 0, xend = NMDS1, yend = NMDS2),
               data = en_coord_cont, size =0.5, alpha = 0.5, colour = "grey30") +
  geom_text(data = en_coord_cont, aes(x = NMDS1, y = NMDS2), colour = "grey30",
            fontface = "bold", label = row.names(en_coord_cont))+

  labs(title=paste0("Stress: ", round(ass.rel.t2_NMS$stress,3)))

(nmuds.plot.sp + theme(legend.position = "none")) + nmuds.plot.nutrient + plot_layout(guides = "collect")

```

Plant composition: 2020



2020-2022 - P and pH seem to be significantly affecting the compositions of log and open communities - I doubt it since envfit function do not take 'year' as a random effect.

```
# Data wrangling
# This data set does not include unknown data.
comm <- read.csv("20-22_species_composition_data_no_unk.csv", header = T)

# Remove the locations surveyed in 2021 and 2022 that were not surveyed in 2020 - aka, cm=0, cm=21, cm=22
comm<-comm[which(comm$cm_location!=21 & comm$cm_location!=0 & comm$cm_location!=22 & comm$cm_location!=20),]

# Make a group name for each row
comm$grp<-apply(comm[c(1,3,5,6,7)], 1, paste, collapse=":") # timepoint, block, transect_name, initial

# Need to make each row a community using matrify.
commsub<-comm[,c(15,10,13)] # group, species_code, and count of each species for each transect. transect
commsub <-as.data.frame(commsub)
commtry<-matrify(commsub) # make it an expanded species matrix
commtry$x[x which(is.na(commtry$x))] <- 1 # "x" means that there were no individuals in the transect, but
ncol(commtry) # how many species are we working with in our community matrix
```

[1] 87

```
# Store grouping row names as a column, then remove rownames.
commtry$grps<-rownames(commtry)
rownames(commtry)<-NULL
names(commtry)
```

```
## [1] "acul"   "aicu"   "arca"   "ardy"   "arsp"   "auel"   "bldr"   "blrd"   "brdi"
## [10] "brdr"   "brpe"   "brru"   "buse"   "caer"   "cagr"   "cahi"   "casp"   "cear"
## [19] "chau"   "chei"   "chs"    "crcl"   "crc"    "cusc"   "cusp"   "dagl"   "dosp"
## [28] "ento"   "erau"   "ercy"   "erra"   "ersp"   "gite"   "gnre"   "gobe"   "gocy"
## [37] "gono"   "goro"   "gosp"   "haod"   "hygl"   "hyp"    "jubu"   "laro"
## [46] "ledu"   "lele"   "loef"   "misp"   "mite"   "momo"   "mopa"   "niro"   "omco"
```

```

## [55] "orssp"  "pala"   "peai"   "pedu"   "phsu"   "plde"   "poar"   "poca"   "pocap"
## [64] "poce"   "pogn"   "pole"   "pomu"   "pter"   "ptga"   "ptob"   "rhla"   "rhpyp"
## [73] "rhsp"   "ry"     "scna"   "sino"   "sool"   "stfi"   "stpi"   "thma"   "trcy"
## [82] "tris"   "tror"   "trpi"   "waac"   "wagr"   "x"      "grps"

```

```

# Split group info into columns for each variable
mat<-separate(commtry, 88, c("time","block","transect","init","treatment"), ":")
names(mat) #check

```

```

## [1] "acul"      "aicu"      "arca"      "ardy"      "arsp"      "auel"
## [7] "bldr"      "blrd"      "brdi"      "brdr"      "brpe"      "brru"
## [13] "buse"      "caer"      "cagr"      "cahi"      "casp"      "cear"
## [19] "chau"      "chei"      "chs"       "crcl"      "crc"       "cusc"
## [25] "cusp"      "dagl"      "dosp"      "ento"      "erau"      "ercy"
## [31] "erra"      "ersp"      "gite"      "gnte"      "gobe"      "gocy"
## [37] "gono"      "goro"      "gosp"      "haod"      "hygl"      "hyp"
## [43] "hypo"      "jubu"      "laro"      "ledu"      "lele"      "loef"
## [49] "misp"      "mite"      "momo"      "mopa"      "niro"      "omco"
## [55] "orssp"      "pala"      "peai"      "pedu"      "phsu"      "plde"
## [61] "poar"      "poca"      "pocap"     "poce"      "pogn"      "pole"
## [67] "pomu"      "pter"      "ptga"      "ptob"      "rhla"      "rhpyp"
## [73] "rhsp"      "ry"        "scna"      "sino"      "sool"      "stfi"
## [79] "stpi"      "thma"      "trcy"      "tris"      "tror"      "trpi"
## [85] "waac"      "wagr"      "x"        "time"      "block"      "transect"
## [91] "init"      "treatment"

```

```

# Set up nutrient analysis data
nutrient <- read.csv("Nutrient.csv", header = T)
nutrient <- separate(nutrient, 2 , c("block","plot"), "_")
nutrient <- nutrient[,2:17]

# subset data where all t0 communities, insitu log and insitu open communities at t1 and t2 are included
mat3 <- mat[which(mat$time=="t0" | mat$treatment=="open" | mat$treatment=="insitu_log") ,]
mat3$grp<-apply(mat3[c(88,89,91)], 1, paste, collapse=":") # block, init as grouping
names(mat3) #check

```

```

## [1] "acul"      "aicu"      "arca"      "ardy"      "arsp"      "auel"
## [7] "bldr"      "blrd"      "brdi"      "brdr"      "brpe"      "brru"
## [13] "buse"      "caer"      "cagr"      "cahi"      "casp"      "cear"
## [19] "chau"      "chei"      "chs"       "crcl"      "crc"       "cusc"
## [25] "cusp"      "dagl"      "dosp"      "ento"      "erau"      "ercy"
## [31] "erra"      "ersp"      "gite"      "gnte"      "gobe"      "gocy"
## [37] "gono"      "goro"      "gosp"      "haod"      "hygl"      "hyp"
## [43] "hypo"      "jubu"      "laro"      "ledu"      "lele"      "loef"
## [49] "misp"      "mite"      "momo"      "mopa"      "niro"      "omco"
## [55] "orssp"      "pala"      "peai"      "pedu"      "phsu"      "plde"
## [61] "poar"      "poca"      "pocap"     "poce"      "pogn"      "pole"
## [67] "pomu"      "pter"      "ptga"      "ptob"      "rhla"      "rhpyp"
## [73] "rhsp"      "ry"        "scna"      "sino"      "sool"      "stfi"
## [79] "stpi"      "thma"      "trcy"      "tris"      "tror"      "trpi"
## [85] "waac"      "wagr"      "x"        "time"      "block"      "transect"
## [91] "init"      "treatment" "grp"

```

```

# another df where the grouping variables are time, block and initial state
# each row is a transect in a certain year.
df<-mat3[,c(1:87, 93)]
df2 = df %>% mutate(across(.cols=1:87, .fns=as.numeric)) # make everything numeric
rownames(df2)<-NULL # remove rownames

## new with group vars
nublock<-separate(df2, 88, c("time", "block", "init"), ":") # just looking at time, block & initial tre

# want to sum across transects in same block X init treatment
nublock$sumgrp<-apply(mat3[c(88,89, 91)], 1, paste, collapse=":")
head(nublock)

```

```

## acul aicu arca ardy arsp auel bldr blrd brdi brdr brpe brru buse caer cagr
## 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
## 2 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1
## 3 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1
## 4 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
## 5 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
## 6 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
## cahi casp clear chau chei chps crcl crc0 cusc cusp dagl dosp ento erau ercy
## 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
## 2 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0
## 3 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
## 4 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
## 5 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
## 6 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
## erra ersp gite gnre gobe gocy gono goro gosp haad hygl hypi hypo jubu laro
## 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
## 2 0 0 0 0 0 0 0 0 0 0 0 4 0 0 0 0 0
## 3 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0
## 4 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
## 5 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1
## 6 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1
## ledu lele loef misp mite momo mopa niro omco orsp pala peai pedu phsu plde
## 1 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0
## 2 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
## 3 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
## 4 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
## 5 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
## 6 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
## poar poca pocap poce pogn pole pomu pter ptga ptob rhla rhpy rhsp ry scna
## 1 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0
## 2 0 1 0 0 0 1 2 0 0 0 0 0 0 0 0 0 0
## 3 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
## 4 0 0 0 0 0 3 2 0 0 0 0 0 0 0 0 0 0
## 5 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0
## 6 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
## sino sool stfi stpi thma trcy tris tror trpi waac wagr x time block init
## 1 0 0 0 0 0 0 0 0 0 0 2 0 0 t0 1 log
## 2 0 0 0 0 0 0 0 0 0 0 0 0 0 t0 1 log
## 3 0 0 0 0 0 0 0 0 0 0 0 0 0 t0 1 log
## 4 0 0 0 0 0 0 0 0 0 0 0 0 0 t0 1 log

```

```

## 5   0   0   0   0   0   1   0   0   0   1   0  0   t0   1   log
## 6   0   0   0   0   0   0   0   1   0   2   0  0   t0   1   log
##       sumgrp
## 1 t0:1:log
## 2 t0:1:log
## 3 t0:1:log
## 4 t0:1:log
## 5 t0:1:log
## 6 t0:1:log

# sum observations across initial X block (group variable)
# this gives number of plants in each transect TYPE for each year in each block. should be 2 types X 3
blocksum<-rowsum(nublock[,c(1:87)], group=nublock$sumgrp)
blocksum$grps<-rownames(blocksum)
rownames(blocksum)<-NULL # remove rownames
nrow(blocksum) # it is 42 rows as expected

## [1] 42

## expand again
blocksum<-separate(blocksum, 88, c("time", "block", "init"), ":")
nutrient_join <- nutrient[,c(1,3,5:9, 11,16)]
blocksum <- inner_join(blocksum, nutrient_join, by = c("init", "block"))

# at the moment this includes where there were no plants ("x" column in matrix)
assemblies_t012<-blocksum[,c(1:87)]

# group - these are the treatment variables that need to be separately fed into the MDS analysis from
group_init<-blocksum$init
group_block<-blocksum$block
group_time<-blocksum$time
group_nutrient<-blocksum[,c(90:96)]

# MDS
ass.rel.t012<-decostand(assemblies_t012, method='hel') #standardize assemblies
ass.rel.t012_NMS <- metaMDS(ass.rel.t012, distance = 'bray', k = 5) # run MDS

## Run 0 stress 0.09244926
## Run 1 stress 0.09501545
## Run 2 stress 0.09244917
## ... New best solution
## ... Procrustes: rmse 0.0003809672 max resid 0.001513277
## ... Similar to previous best
## Run 3 stress 0.09615208
## Run 4 stress 0.09245064
## ... Procrustes: rmse 0.0007757914 max resid 0.003075367
## ... Similar to previous best
## Run 5 stress 0.09245378
## ... Procrustes: rmse 0.001685675 max resid 0.006775918
## ... Similar to previous best
## Run 6 stress 0.09244949
## ... Procrustes: rmse 0.0005095654 max resid 0.002015636
## ... Similar to previous best

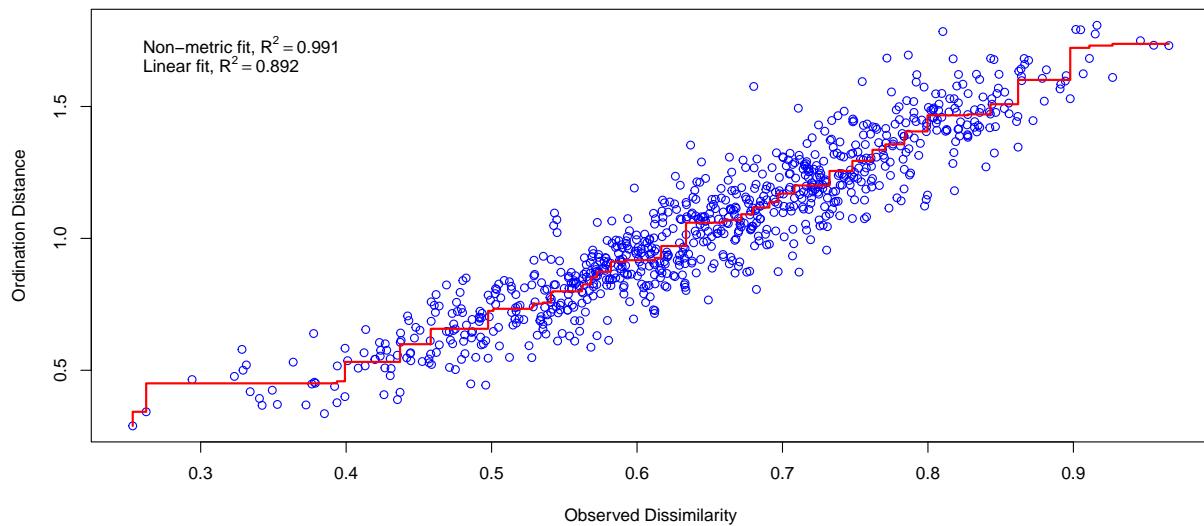
```

```

## Run 7 stress 0.09245036
## ... Procrustes: rmse 0.0009072156 max resid 0.003607938
## ... Similar to previous best
## Run 8 stress 0.09245058
## ... Procrustes: rmse 0.0009503784 max resid 0.003768354
## ... Similar to previous best
## Run 9 stress 0.09647477
## Run 10 stress 0.09244988
## ... Procrustes: rmse 0.0007331996 max resid 0.002911823
## ... Similar to previous best
## Run 11 stress 0.09682312
## Run 12 stress 0.0924632
## ... Procrustes: rmse 0.002813899 max resid 0.0114966
## Run 13 stress 0.0924498
## ... Procrustes: rmse 0.0005180953 max resid 0.002065731
## ... Similar to previous best
## Run 14 stress 0.09508168
## Run 15 stress 0.09245807
## ... Procrustes: rmse 0.001645203 max resid 0.005399029
## ... Similar to previous best
## Run 16 stress 0.09244981
## ... Procrustes: rmse 0.0007071454 max resid 0.00280267
## ... Similar to previous best
## Run 17 stress 0.09244985
## ... Procrustes: rmse 0.0007075729 max resid 0.002799423
## ... Similar to previous best
## Run 18 stress 0.09264001
## ... Procrustes: rmse 0.008771774 max resid 0.03034246
## Run 19 stress 0.09245001
## ... Procrustes: rmse 0.0007877376 max resid 0.003115945
## ... Similar to previous best
## Run 20 stress 0.0924502
## ... Procrustes: rmse 0.0008520355 max resid 0.003392997
## ... Similar to previous best
## *** Best solution repeated 13 times

```

```
stressplot(ass.rel.t012_NMS) # check fit
```



```

en.nutrient = envfit(ass.rel.t012_NMS, group_nutrient, permutations = 999, na.rm = TRUE)
# plot(ass.rel.t012_NMS)
# plot(en.nutrient)
print(en.nutrient) ##### it is basically saying most of the nutrient does not affect the composition

```

```

##
## ***VECTORS
##
##      NMDS1     NMDS2      r2 Pr(>r)
## NH4N  0.69843 -0.71568 0.1190  0.076 .
## NO3N  0.47088 -0.88220 0.0103  0.806
## P    -0.52714  0.84978 0.2411  0.007 **
## K    -0.45721  0.88936 0.0783  0.210
## C    0.63357 -0.77368 0.0728  0.237
## pH   -0.72398  0.68982 0.2344  0.011 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## Permutation: free
## Number of permutations: 999
##
## ***FACTORS:
##
## Centroids:
##      NMDS1     NMDS2
## initlog  0.1007 -0.1303
## initopen -0.1007  0.1303
##
## Goodness of fit:
##      r2 Pr(>r)
## init 0.0902  0.025 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## Permutation: free

```

```

## Number of permutations: 999

# scores
mds_scores_t012<-as.data.frame(vegan::scores(ass.rel.t012_NMS)$sites) # extract scores
mds_scores_t012$site<-rownames(vegan::scores(ass.rel.t012_NMS)$sites) # extract names
mds_scores_t012$treatment<-group_init # grouping factor 1
mds_scores_t012$block<-group_block # grouping factor 2
mds_scores_t012$time<-group_time # grouping factor 3
mds_scores_t012 <- cbind(mds_scores_t012, group_nutrient)
en_coord_cont = as.data.frame(vegan::scores(en.nutrient, "vectors")) * ordiArrowMul(en.nutrient)

# explaining factors
init<-as.factor(group_init) # grouping factor 1- convert to factor
block<-as.factor(group_block) # grouping factor 2- convert to factor
time<-as.factor(group_time)

##### rda model analysis & results #####
# can look at significance of model where initial treatment and block explain variation in community
trt_tot_2<-rda(ass.rel.t012~init+block+time) # run model using standardized data
summary(trt_tot_2)

##
## Call:
## rda(formula = ass.rel.t012 ~ init + block + time)
##
## Partitioning of variance:
##           Inertia Proportion
## Total      0.5618    1.0000
## Constrained 0.2832    0.5041
## Unconstrained 0.2786    0.4959
##
## Eigenvalues, and their contribution to the variance
##
## Importance of components:
##          RDA1     RDA2     RDA3     RDA4     RDA5     RDA6     RDA7
## Eigenvalue      0.08458  0.05964  0.04201  0.02374  0.02255  0.01906  0.01374
## Proportion Explained 0.15056  0.10617  0.07479  0.04226  0.04015  0.03393  0.02446
## Cumulative Proportion 0.15056  0.25673  0.33152  0.37377  0.41392  0.44785  0.47231
##          RDA8     RDA9     PC1      PC2      PC3      PC4      PC5
## Eigenvalue      0.01092  0.006918 0.03970  0.02773  0.02383  0.02008  0.01758
## Proportion Explained 0.01944  0.012315 0.07067  0.04937  0.04241  0.03574  0.03129
## Cumulative Proportion 0.49174  0.504060 0.57473  0.62410  0.66651  0.70225  0.73354
##          PC6      PC7      PC8      PC9      PC10     PC11     PC12
## Eigenvalue      0.01589  0.01486  0.01326  0.01138  0.01024  0.01004  0.008831
## Proportion Explained 0.02828  0.02646  0.02360  0.02025  0.01822  0.01787  0.015720
## Cumulative Proportion 0.76182  0.78828 0.81188  0.83213  0.85035  0.86823  0.883948
##          PC13     PC14     PC15     PC16     PC17     PC18
## Eigenvalue      0.008658 0.007353 0.006713 0.005727 0.005255 0.004236
## Proportion Explained 0.015413 0.013089 0.011951 0.010196 0.009354 0.007542
## Cumulative Proportion 0.899361 0.912451 0.924402 0.934597 0.943951 0.951493
##          PC19     PC20     PC21     PC22     PC23     PC24
## Eigenvalue      0.003939 0.003339 0.003025 0.002930 0.002555 0.002392
## Proportion Explained 0.007013 0.005944 0.005384 0.005215 0.004548 0.004258
## Cumulative Proportion 0.958506 0.964449 0.969833 0.975049 0.979597 0.983855

```

```

##          PC25      PC26      PC27      PC28      PC29      PC30
## Eigenvalue       0.00227 0.001666 0.001540 0.001124 0.0009868 0.0006454
## Proportion Explained 0.00404 0.002965 0.002741 0.002001 0.0017567 0.0011489
## Cumulative Proportion 0.98789 0.990860 0.993601 0.995602 0.9973584 0.9985073
##          PC31      PC32
## Eigenvalue       0.0005422 0.0002963
## Proportion Explained 0.0009651 0.0005275
## Cumulative Proportion 0.9994725 1.0000000
##
## Accumulated constrained eigenvalues
## Importance of components:
##          RDA1      RDA2      RDA3      RDA4      RDA5      RDA6      RDA7
## Eigenvalue       0.08458 0.05964 0.04201 0.02374 0.02255 0.01906 0.01374
## Proportion Explained 0.29870 0.21062 0.14837 0.08383 0.07965 0.06732 0.04852
## Cumulative Proportion 0.29870 0.50932 0.65769 0.74152 0.82117 0.88849 0.93701
##          RDA8      RDA9
## Eigenvalue       0.01092 0.006918
## Proportion Explained 0.03856 0.024432
## Cumulative Proportion 0.97557 1.0000000
##
## Scaling 2 for species and site scores
## * Species are scaled proportional to eigenvalues
## * Sites are unscaled: weighted dispersion equal on all dimensions
## * General scaling constant of scores: 2.190697
##
##
## Species scores
##          RDA1      RDA2      RDA3      RDA4      RDA5      RDA6
## acul  -1.352e-01  5.332e-02 -1.259e-02  5.023e-02 -6.287e-02 -6.341e-02
## aicu  -7.804e-04 -5.802e-02  9.699e-03  6.012e-02  5.474e-03 -4.253e-02
## arca  -2.071e-17 -2.635e-18  2.541e-17  3.766e-18  2.527e-17 -5.781e-18
## ardy   5.272e-03  1.040e-02  2.181e-02 -1.130e-02 -2.771e-02  2.509e-02
## arsp  -9.643e-03 -1.332e-03 -1.163e-02  1.555e-02  8.335e-04  5.776e-03
## auel  -8.884e-18 -2.461e-17  6.210e-18 -1.474e-17 -1.563e-17 -3.929e-18
## bldr  -1.408e-01 -3.901e-02  1.983e-02  1.389e-01  9.836e-02  6.296e-02
## blrd  -2.264e-02  6.417e-03  8.458e-03 -7.255e-03 -2.197e-02 -9.062e-03
## brdi   3.152e-03  1.948e-02 -1.467e-02  1.369e-02 -1.461e-03  1.076e-02
## brdr   5.608e-18  2.575e-18  1.203e-17 -5.589e-18  5.891e-19  2.690e-18
## brpe  -1.602e-02 -3.186e-02  3.804e-03 -8.893e-02 -1.504e-02  5.185e-02
## brru  1.333e-17 -1.433e-17 -2.608e-17  3.433e-18  1.107e-18 -3.085e-18
## buse  -2.259e-02 -2.523e-02 -2.112e-02  1.968e-02  2.990e-02 -1.224e-02
## caer  -3.385e-02  2.762e-02 -2.170e-02 -8.946e-02 -3.206e-03  3.380e-02
## cagr  -2.505e-02 -5.415e-02 -2.573e-02 -1.043e-01  8.204e-03 -3.671e-02
## cahi  1.410e-02  3.252e-02  1.110e-02  2.376e-02  4.086e-02  6.657e-02
## casp  -3.535e-02 -2.387e-02 -3.264e-02  6.054e-03  1.350e-03 -5.146e-03
## clear 7.452e-03 -8.262e-03 -2.765e-02 -8.841e-03 -2.988e-02 -5.798e-04
## chau  4.033e-02 -8.487e-02  2.595e-02  6.152e-02 -1.217e-01  1.286e-02
## chei  6.797e-02  3.015e-02  1.177e-03  3.980e-02 -3.057e-02 -1.319e-03
## chps  -2.114e-01  1.346e-01  1.206e-02  3.194e-02 -8.714e-02 -4.737e-02
## crcl  5.675e-03  2.313e-02  2.127e-02  2.532e-03  2.900e-02  1.526e-02
## crco  -9.751e-02  4.977e-02  2.415e-03  9.312e-02  7.276e-04 -2.347e-02
## cusc  1.839e-02  4.290e-02  1.769e-02  5.705e-03  5.243e-02  3.533e-02
## cusp  -6.425e-02  8.335e-03  5.404e-03 -6.236e-02 -5.843e-02  5.023e-02

```

```

## dag1 -1.434e-02 -4.253e-03 1.104e-03 2.858e-03 1.531e-02 -2.459e-03
## dosp 9.908e-02 2.029e-02 6.549e-02 2.487e-02 -1.511e-02 -2.168e-02
## ento 7.849e-34 -4.227e-33 -1.995e-33 -1.417e-33 -1.322e-33 -2.419e-33
## erau 6.815e-03 1.048e-02 1.447e-02 -3.571e-03 2.410e-02 1.099e-02
## ercy 7.238e-03 8.826e-02 1.439e-02 -9.057e-02 4.129e-02 -4.748e-02
## erra -2.996e-02 1.079e-02 -2.628e-03 1.992e-02 -2.935e-02 -9.026e-02
## ersp -5.503e-03 -9.310e-03 -4.865e-04 1.811e-03 3.647e-03 -1.648e-02
## gite 9.198e-02 -1.160e-01 1.446e-01 8.713e-02 -5.955e-02 1.559e-02
## gntr 9.128e-03 -1.595e-02 6.640e-03 1.396e-03 -8.172e-02 1.637e-02
## gobe 7.151e-02 -8.798e-02 -2.389e-01 4.876e-02 7.212e-02 -3.509e-02
## gocy -5.707e-03 -6.090e-02 -7.598e-02 4.988e-02 2.000e-02 9.361e-03
## gono -3.904e-34 6.158e-34 1.814e-33 -1.112e-33 -6.721e-34 -6.411e-35
## goro 2.038e-01 1.974e-01 -3.493e-02 8.297e-02 -4.879e-02 9.732e-02
## gosp 7.403e-03 -4.710e-02 -4.175e-02 5.262e-02 -1.022e-03 -4.045e-03
## haod 3.898e-02 5.696e-02 -1.143e-01 4.754e-02 1.887e-02 2.985e-03
## hygl -3.834e-01 -1.102e-01 -1.387e-01 1.659e-01 4.343e-02 -4.491e-02
## hypi -9.794e-03 -2.952e-03 3.739e-03 -2.814e-02 -8.109e-03 2.043e-02
## hypo 1.169e-01 5.102e-02 4.109e-02 -2.585e-02 7.716e-02 -1.129e-02
## jubu 0.000e+00 0.000e+00 0.000e+00 0.000e+00 0.000e+00 0.000e+00
## laro 5.560e-02 -1.039e-01 -7.036e-02 3.383e-03 6.148e-02 3.348e-02
## ledu -2.646e-02 -1.599e-03 2.294e-02 -1.829e-02 -1.181e-03 1.968e-02
## lele -1.784e-02 1.768e-03 -2.167e-02 1.617e-02 -9.759e-03 1.495e-02
## loef 2.090e-02 8.347e-03 1.803e-02 1.640e-02 -1.202e-02 1.667e-03
## misp -7.312e-02 -1.752e-01 -1.442e-01 -1.544e-01 -9.022e-02 -1.264e-01
## mite 2.426e-01 2.502e-01 -8.879e-02 -1.518e-02 9.716e-02 -2.975e-02
## momo 0.000e+00 0.000e+00 0.000e+00 0.000e+00 0.000e+00 0.000e+00
## mopa 0.000e+00 0.000e+00 0.000e+00 0.000e+00 0.000e+00 0.000e+00
## niro -1.221e-02 -8.832e-03 1.580e-02 -2.057e-02 7.519e-03 1.444e-02
## omco -3.492e-02 2.675e-03 2.675e-03 -7.803e-04 2.781e-03 1.185e-02
## orsp 9.156e-03 7.486e-03 -1.198e-02 1.317e-03 9.096e-03 1.150e-02
## pala -9.643e-03 -1.332e-03 -1.163e-02 1.555e-02 8.335e-04 5.776e-03
## peai 1.996e-01 1.408e-01 -9.839e-02 6.276e-02 -1.174e-01 -1.522e-01
## pedu -1.364e-02 -1.883e-03 -1.644e-02 2.199e-02 1.179e-03 8.169e-03
## phsu -8.636e-03 -6.245e-03 1.117e-02 -1.455e-02 5.317e-03 1.021e-02
## plde 0.000e+00 0.000e+00 0.000e+00 0.000e+00 0.000e+00 0.000e+00
## poar 5.214e-03 -1.209e-02 2.999e-02 -2.594e-02 -2.562e-02 -3.414e-03
## poca 4.192e-02 8.124e-02 2.421e-02 2.581e-02 8.822e-02 -8.890e-02
## pocap 8.697e-02 -3.209e-02 -5.302e-02 4.155e-02 -4.701e-03 -9.728e-03
## poce 9.156e-03 7.486e-03 -1.198e-02 1.317e-03 9.096e-03 1.150e-02
## pogn 0.000e+00 0.000e+00 0.000e+00 0.000e+00 0.000e+00 0.000e+00
## pole -1.463e-01 -4.319e-02 3.372e-01 -2.635e-02 5.774e-02 -1.385e-01
## pomu -4.440e-01 2.539e-01 -6.204e-02 -9.485e-02 5.478e-02 6.174e-02
## pter 2.655e-03 5.056e-02 -5.263e-03 2.698e-02 -8.523e-04 1.103e-03
## ptga -5.643e-02 -6.544e-02 -1.842e-02 5.964e-02 -3.666e-02 -5.975e-02
## ptob 0.000e+00 0.000e+00 0.000e+00 0.000e+00 0.000e+00 0.000e+00
## rhla 1.186e-02 -7.171e-02 7.555e-02 8.678e-02 3.446e-02 5.557e-03
## rhpy -4.529e-02 -6.666e-02 -1.228e-01 -3.905e-02 -6.508e-03 9.903e-02
## rhsp 1.334e-01 5.970e-02 8.546e-02 3.620e-03 3.938e-02 -2.220e-02
## ry 0.000e+00 0.000e+00 0.000e+00 0.000e+00 0.000e+00 0.000e+00
## scna 8.647e-03 -9.521e-03 9.665e-04 -1.552e-03 -3.535e-02 2.814e-03
## sino 6.815e-03 1.048e-02 1.447e-02 -3.571e-03 2.410e-02 1.099e-02
## sool -5.069e-03 -7.645e-03 -1.504e-02 -1.305e-02 -3.740e-03 1.751e-02
## stfi 1.665e-02 -9.154e-03 -2.588e-02 3.378e-03 1.522e-02 9.245e-03
## stpi 0.000e+00 0.000e+00 0.000e+00 0.000e+00 0.000e+00 0.000e+00

```

```

## thma  0.000e+00  0.000e+00  0.000e+00  0.000e+00  0.000e+00  0.000e+00
## trcy  7.531e-02 -3.294e-01  4.383e-02 -1.270e-02 -1.299e-02  9.935e-02
## tris -3.720e-02  6.677e-03 -2.576e-02 -5.285e-02 -1.163e-01 -9.491e-04
## tror  1.109e-01 -1.595e-01 -1.479e-01 -6.950e-02  7.228e-02 -7.979e-02
## trpi  3.280e-02 -6.171e-02  6.544e-02 -2.251e-02 -2.722e-02  2.228e-02
## waac  4.608e-02 -9.984e-02  2.809e-02 -4.060e-02  1.915e-01 -5.280e-02
## wagr -3.673e-02 -3.151e-02  1.045e-04  1.517e-02  4.158e-04  8.068e-03
## x    -2.098e-02 -5.255e-02 -8.786e-03  2.502e-02  2.387e-02 -5.198e-02
##
##
## Site scores (weighted sums of species scores)
##
##          RDA1      RDA2      RDA3      RDA4      RDA5      RDA6
## row1   -0.257338 -0.313063  0.020105 -0.134844  0.64621 -0.366172
## row2   -0.398714  0.123562 -0.048418 -0.128493  0.42943 -0.105014
## row3   0.317456 -0.769095  0.672801  0.268812  0.11621 -0.014977
## row4   0.272592 -0.226513  0.502008  0.637420 -0.16794 -0.090995
## row5   -0.147937 -0.388979  0.366416  0.303362  0.47504  0.281488
## row6   -0.446754 -0.052862  0.395907  0.521843  0.26280 -0.017165
## row7   -0.210697 -0.395476 -0.215785 -0.030810  0.03097 -0.622618
## row8   -0.494843 -0.082658  0.137590  0.415334  0.13999 -0.704501
## row9   -0.175615 -0.155737 -0.189747  0.332952  0.12227  0.590741
## row10  -0.488579  0.388216 -0.267539  0.204636 -0.02584  0.257674
## row11  -0.292306 -0.240244 -0.390268  0.413529  0.02767 -0.055817
## row12  -0.394403 -0.171885 -0.332983  0.848938  0.21852  0.087984
## row13  0.156068 -0.733320 -0.656816 -0.151352  0.39094 -0.222948
## row14  0.135220 -0.608855 -0.459284  0.283059 -0.19786 -0.021358
## row15  -0.178952  0.116695  0.064179 -1.189249  0.25669 -0.207551
## row16  -0.387875  0.188633  0.008533 -0.008487 -0.16194 -0.304608
## row17  0.381340 -0.448629  0.708800  0.057059 -0.19109  0.488855
## row18  0.204507 -0.234608  0.324452  0.070379 -1.02230 -0.359593
## row19  -0.065217 -0.291501  0.410172 -0.058580  0.21305  0.257654
## row20  -0.558097  0.151048  0.261748  0.126193 -0.09271  0.179499
## row21  -0.244697 -0.093999 -0.031868 -0.641495 -0.52418 -0.552265
## row22  -0.404294 -0.156219 -0.029492 -0.159984 -0.61672 -0.796503
## row23  -0.115820 -0.355621  0.334501 -1.190103 -0.21593  0.250267
## row24  -0.475941  0.457123 -0.028082 -0.115206 -0.19384  0.422101
## row25  -0.241955  0.001675 -0.373074 -0.579241 -0.15920  0.743215
## row26  -0.110199  0.174188 -0.375245 -0.036516 -0.47646  0.479780
## row27  0.332359 -0.388262 -0.657868 -0.519986 -0.04383  0.074004
## row28  0.169196 -0.284788 -0.290525 -0.175023 -0.67333  0.264396
## row29  0.561163  0.238126  0.347851 -0.239475  0.88229 -0.454358
## row30  0.294810  0.672463  0.013193  0.089305  0.05599 -0.444167
## row31  0.520212 -0.117833  0.505750  0.058177 -0.04588  0.506952
## row32  0.679617  0.223004  0.372613  0.377652 -0.34348 -0.087277
## row33  -0.043987  0.244517  0.149534 -0.326693  0.63019  0.396483
## row34  -0.191807  0.631411  0.213954 -0.289012  0.15988  0.169792
## row35  0.460770  0.404081  0.006818  0.020909  0.06670 -0.448503
## row36  0.009364  0.583820  0.119411  0.239266 -0.27455 -0.575313
## row37  0.331998  0.089081  0.449743  0.038214  0.15789  0.071349
## row38  -0.279893  0.609199 -0.030333 -0.144736 -0.25980  0.576473
## row39  0.261087  0.155062 -0.572465 -0.172904  0.44691  0.319909
## row40  0.152359  0.389691 -0.400201  0.537092  0.07334  0.405173
## row41  0.779430  0.180476 -0.610367  0.130056  0.18986 -0.375367

```

```

## row42  0.586373  0.488076 -0.425722  0.318004 -0.30596  0.003282
##
##
## Site constraints (linear combinations of constraining variables)
##
##          RDA1      RDA2      RDA3      RDA4      RDA5      RDA6
## row1   -0.133123 -0.24333  0.050402 -0.17021  0.693128 -0.35389
## row2   -0.335729  0.06734  0.018173  0.17376  0.362387 -0.41678
## row3    0.323982 -0.67668  0.497232  0.34325  0.065938  0.03358
## row4    0.121376 -0.36601  0.465002  0.68721 -0.264802 -0.02931
## row5   -0.314272 -0.36546  0.282450  0.14451  0.616393  0.17105
## row6   -0.516878 -0.05479  0.250220  0.48848  0.285652  0.10816
## row7   -0.219372 -0.37114 -0.019393  0.07220  0.145386 -0.65686
## row8   -0.421978 -0.06047 -0.051622  0.41617 -0.185355 -0.71975
## row9   -0.272614 -0.24236  0.027586 -0.04738  0.272477  0.32119
## row10  -0.475220  0.06831 -0.004644  0.29659 -0.058264  0.25830
## row11  -0.176208 -0.36299 -0.424544  0.26681  0.363482  0.28979
## row12  -0.378814 -0.05231 -0.456774  0.61078  0.032741  0.22690
## row13   0.287802 -0.63885 -0.533935  0.07912  0.234988 -0.08658
## row14   0.085196 -0.32817 -0.566165  0.42309 -0.095753 -0.14947
## row15  -0.080966 -0.06743  0.106991 -0.75626  0.238115 -0.21511
## row16  -0.283571  0.24324  0.074761 -0.41229 -0.092626 -0.27800
## row17   0.376139 -0.50078  0.553821 -0.24280 -0.389075  0.17236
## row18   0.173533 -0.19011  0.521591  0.10117 -0.719815  0.10947
## row19  -0.262114 -0.18956  0.339039 -0.44153  0.161380  0.30983
## row20  -0.464720  0.12111  0.306809 -0.09756 -0.169361  0.24694
## row21  -0.167214 -0.19524  0.037196 -0.51385 -0.309627 -0.51808
## row22  -0.369820  0.11543  0.004967 -0.16988 -0.640368 -0.58097
## row23  -0.220456 -0.06645  0.084175 -0.63342 -0.182536  0.45997
## row24  -0.423062  0.24422  0.051945 -0.28945 -0.513277  0.39708
## row25  -0.124051 -0.18708 -0.367955 -0.31923 -0.091531  0.42857
## row26  -0.326657  0.12359 -0.400185  0.02474 -0.422272  0.36569
## row27   0.339960 -0.46294 -0.477346 -0.50692 -0.220025  0.05220
## row28   0.137354 -0.15227 -0.509576 -0.16295 -0.550766 -0.01069
## row29   0.334545  0.35796  0.093673 -0.39511  0.619198 -0.27760
## row30   0.131939  0.66863  0.061443 -0.05114  0.288458 -0.34049
## row31   0.791649 -0.07539  0.540503  0.11835 -0.007991  0.10987
## row32   0.589044  0.23528  0.508273  0.46232 -0.338732  0.04698
## row33   0.153396  0.23583  0.325721 -0.08038  0.542463  0.24734
## row34  -0.049210  0.54650  0.293491  0.26359  0.211722  0.18445
## row35   0.248296  0.23015  0.023878 -0.15270  0.071456 -0.58057
## row36   0.045690  0.54082 -0.008352  0.19127 -0.259284 -0.64346
## row37   0.195054  0.35893  0.070856 -0.27227  0.198548  0.39748
## row38  -0.007552  0.66961  0.038627  0.07170 -0.132193  0.33459
## row39   0.291459  0.23831 -0.381274  0.04192  0.289552  0.36609
## row40   0.088853  0.54898 -0.413503  0.38589 -0.041189  0.30320
## row41   0.755470 -0.03755 -0.490664 -0.14577  0.161058 -0.01028
## row42   0.552864  0.27312 -0.522894  0.19820 -0.169683 -0.07317
##
##
## Biplot scores for constraining variables
##
##          RDA1      RDA2      RDA3      RDA4      RDA5      RDA6
## initopen -0.2997  0.45953 -0.047673  0.50878 -0.48922 -0.093023

```

```

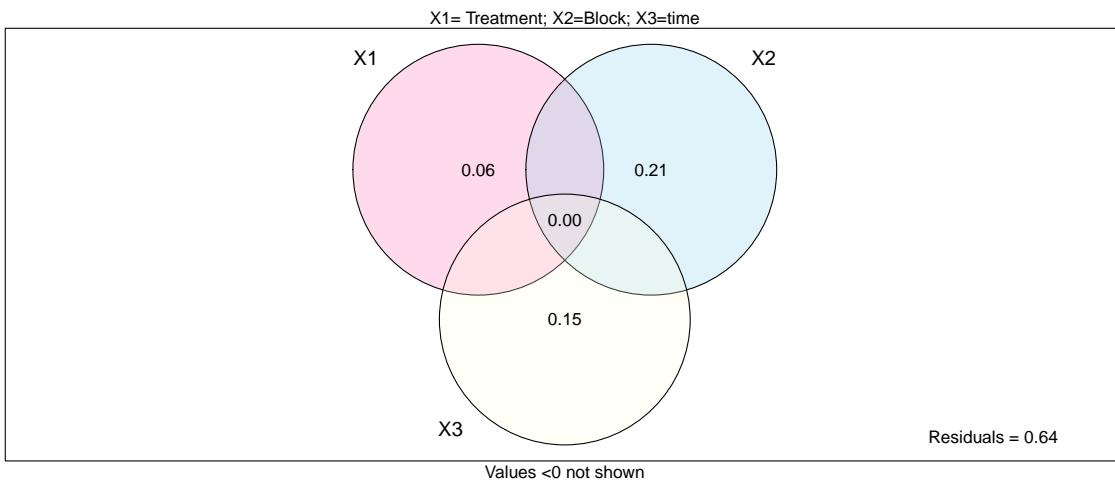
## block2    0.4782 -0.31676  0.621256  0.29579 -0.33302  0.089163
## block3   -0.2926  0.05910  0.361860  0.05578  0.33177  0.255182
## block4   -0.1780  0.05224 -0.002682 -0.03156 -0.23707 -0.744702
## block5   -0.2423  0.20778  0.054054 -0.17597 -0.08358  0.436512
## block6   -0.1259  0.06209 -0.491993  0.20348  0.02633  0.398597
## block7    0.4345 -0.27107 -0.624106 -0.02320 -0.12886 -0.055956
## timet1   -0.2534 -0.17396  0.048745 -0.66046 -0.58299  0.140340
## timet2    0.6158  0.71588  0.020885  0.09501  0.21417  0.009627
##
##
## Centroids for factor constraints
##
##          RDA1     RDA2     RDA3     RDA4     RDA5     RDA6
## initlog  0.10130 -0.15534  0.016115 -0.17198  0.16537  0.031445
## initopen -0.10130  0.15534 -0.016115  0.17198 -0.16537 -0.031445
## block1   -0.06115  0.17107  0.067574 -0.26854  0.35144 -0.313645
## block2    0.39595 -0.26228  0.514404  0.24492 -0.27575  0.073827
## block3   -0.24230  0.04894  0.299622  0.04619  0.27471  0.211292
## block4   -0.14740  0.04326 -0.002221 -0.02613 -0.19630 -0.616617
## block5   -0.20064  0.17204  0.044757 -0.14571 -0.06921  0.361434
## block6   -0.10424  0.05141 -0.407373  0.16848  0.02180  0.330041
## block7    0.35977 -0.22445 -0.516763 -0.01921 -0.10670 -0.046332
## timet0   -0.17328 -0.25906 -0.033287  0.27031  0.17631 -0.071691
## timet1   -0.12112 -0.08316  0.023302 -0.31573 -0.27870  0.067089
## timet2    0.29439  0.34223  0.009984  0.04542  0.10238  0.004602

anova.cca(trt_tot_2, step=1000, by="term") ## test for model significance

## Permutation test for rda under reduced model
## Terms added sequentially (first to last)
## Permutation: free
## Number of permutations: 999
##
## Model: rda(formula = ass.rel.t012 ~ init + block + time)
##           Df Variance      F Pr(>F)
## init        1  0.034157 3.9233  0.001 ***
## block       6  0.160594 3.0743  0.001 ***
## time        2  0.088406 5.0773  0.001 ***
## Residual  32  0.278596
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

# can model using varpart to look at contributions of initial treatment and block
var.mod2<-varpart(ass.rel.t012, init, block, time) # run model on standardized data
plot(var.mod2, bg=c("hotpink","skyblue","lightyellow"))
mtext("X1= Treatment; X2=Block; X3=time", side=3)

```



```
## can test for significance of contribution of the fraction of initial treatment
# do this with partial redundancy analysis
trt_Frac<-rda(ass.rel.t012~ init +Condition(block + time)) # partial rda model
summary(trt_Frac)
```

```
##
## Call:
## rda(formula = ass.rel.t012 ~ init + Condition(block + time))
##
## Partitioning of variance:
##           Inertia Proportion
## Total      0.56175   1.0000
## Conditioned 0.24900   0.4433
## Constrained 0.03416   0.0608
## Unconstrained 0.27860   0.4959
##
## Eigenvalues, and their contribution to the variance
## after removing the contribution of conditiniong variables
##
## Importance of components:
##          RDA1     PC1     PC2     PC3     PC4     PC5     PC6
## Eigenvalue      0.03416 0.0397 0.02773 0.02383 0.02008 0.01758 0.01589
## Proportion Explained 0.10921 0.1269 0.08867 0.07618 0.06419 0.05620 0.05080
## Cumulative Proportion 0.10921 0.2362 0.32482 0.40101 0.46520 0.52139 0.57219
##          PC7     PC8     PC9     PC10    PC11    PC12    PC13
## Eigenvalue      0.01486 0.01326 0.01138 0.01024 0.01004 0.008831 0.008658
## Proportion Explained 0.04752 0.04240 0.03637 0.03273 0.03211 0.028235 0.027685
## Cumulative Proportion 0.61971 0.66211 0.69848 0.73121 0.76332 0.791553 0.819237
##          PC14    PC15    PC16    PC17    PC18    PC19
## Eigenvalue      0.007353 0.006713 0.005727 0.005255 0.004236 0.003939
## Proportion Explained 0.023511 0.021466 0.018313 0.016802 0.013546 0.012596
## Cumulative Proportion 0.842748 0.864213 0.882526 0.899328 0.912874 0.925469
##          PC20    PC21    PC22    PC23    PC24    PC25
```

```

## Eigenvalue      0.003339 0.003025 0.002930 0.002555 0.002392 0.002270
## Proportion Explained 0.010676 0.009671 0.009367 0.008169 0.007648 0.007257
## Cumulative Proportion 0.936145 0.945816 0.955183 0.963353 0.971001 0.978257
##                  PC26     PC27     PC28     PC29     PC30     PC31
## Eigenvalue      0.001666 0.001540 0.001124 0.0009868 0.0006454 0.0005422
## Proportion Explained 0.005326 0.004923 0.003594 0.0031552 0.0020636 0.0017336
## Cumulative Proportion 0.983583 0.988506 0.992100 0.9952553 0.9973189 0.9990524
##                  PC32
## Eigenvalue      0.0002963
## Proportion Explained 0.0009476
## Cumulative Proportion 1.0000000
##
## Accumulated constrained eigenvalues
## Importance of components:
##          RDA1
## Eigenvalue      0.03416
## Proportion Explained 1.00000
## Cumulative Proportion 1.00000
##
## Scaling 2 for species and site scores
## * Species are scaled proportional to eigenvalues
## * Sites are unscaled: weighted dispersion equal on all dimensions
## * General scaling constant of scores: 2.190697
##
##
## Species scores
##
##          RDA1      PC1      PC2      PC3      PC4      PC5
## acul   0.0755447 -1.434e-02  7.884e-02 -1.593e-03  8.467e-02  2.948e-02
## aicu   0.0138510 -5.744e-03 -2.254e-02 -3.823e-02 -1.389e-02 -2.313e-02
## arca   0.0000000 -9.271e-18 -2.981e-17 -4.432e-18  1.136e-17 -5.090e-17
## ardy   0.0061606 -3.600e-02  2.215e-02  4.187e-02  7.286e-03 -9.784e-05
## arsp   0.0086051  7.583e-03  1.074e-02  5.817e-03  9.230e-03  2.952e-03
## auel   0.0000000 -4.520e-17 -3.721e-17  1.033e-17  1.268e-17  4.689e-17
## bldr   0.0415582  1.024e-01 -5.538e-02  3.641e-02  6.442e-02 -6.221e-03
## blrd   0.0183399  5.422e-03  2.508e-02  2.563e-03  1.299e-02  1.353e-02
## brdi   0.0119918  9.759e-03  2.022e-03 -2.539e-03  2.480e-03 -4.325e-03
## brdr   0.0000000 -7.790e-18 -1.962e-17  2.861e-17 -4.322e-18 -1.126e-17
## brpe   -0.0548396  5.401e-02 -1.678e-02 -2.939e-02 -2.420e-02 -8.146e-03
## brru   0.0000000  2.293e-17 -1.930e-17 -3.465e-18 -2.242e-17 -2.590e-17
## buse   -0.0161408 -3.475e-03  1.721e-02 -2.478e-04 -3.937e-03  9.186e-03
## caer   -0.0452451 -6.696e-02  9.202e-02 -1.616e-01  4.003e-02 -3.100e-02
## cagr   -0.0551717 -3.321e-02 -1.867e-02 -6.880e-02 -7.224e-02 -3.597e-02
## cahi   -0.0257702 -1.225e-02  5.902e-02  2.155e-02 -1.343e-02 -6.753e-03
## casp   0.0015456 -1.514e-02  8.747e-03  2.929e-02  6.102e-02  1.539e-02
## cear   0.0183399  9.032e-03  3.008e-03  7.984e-03 -3.408e-02 -4.754e-02
## chau   0.0711459  2.373e-02  4.398e-02 -1.230e-01 -5.926e-02 -5.960e-02
## chei   0.0401993  7.383e-03 -2.523e-02  6.336e-03 -3.648e-02  4.773e-02
## chps   0.1934722 -2.173e-02  3.508e-02  7.579e-02  1.279e-01  3.187e-02
## crcl   -0.0071908 -4.839e-02  6.939e-03 -3.034e-02 -3.984e-03  1.194e-03
## crco   0.0909945  1.388e-02  3.934e-02  1.367e-01  3.854e-02 -7.024e-02
## cusc   -0.0215871 -7.383e-02  3.359e-02 -5.510e-02 -1.432e-02  1.409e-02
## cusp   0.0102164 -1.830e-03  2.398e-02 -2.390e-03  8.359e-03  1.051e-02
## dagl   -0.0002086 -1.267e-02 -3.465e-03  4.740e-03  5.809e-03  5.524e-03

```

```

## dosp -0.0228091 -7.269e-02 7.834e-02 1.285e-01 -1.337e-02 -7.897e-02
## ento 0.0000000 -3.969e-18 -6.278e-18 -1.112e-17 -2.983e-19 5.476e-18
## erau -0.0150169 -3.738e-02 1.412e-02 -2.351e-02 -7.513e-03 8.236e-03
## ercy -0.0354798 -1.267e-02 -4.291e-02 1.901e-02 -9.685e-03 -3.599e-02
## erra 0.0227357 -1.122e-02 2.726e-02 2.923e-03 -1.143e-02 7.106e-03
## ersp -0.0084795 -7.754e-03 3.923e-03 -1.604e-03 -1.582e-02 6.429e-03
## gite -0.0130745 3.665e-02 -9.580e-02 -1.441e-02 6.247e-02 9.662e-03
## gnte 0.0449263 6.207e-03 3.387e-02 -5.636e-02 -6.490e-02 -1.134e-02
## gobe -0.0389147 -1.221e-02 2.324e-03 2.536e-02 1.356e-02 2.056e-01
## gocy -0.0005179 3.628e-03 4.776e-02 8.507e-03 3.434e-04 -3.315e-02
## gono 0.0000000 5.463e-18 1.998e-18 -1.741e-18 -8.489e-18 6.585e-19
## goro 0.1287318 -6.246e-02 -1.634e-01 6.953e-02 -2.130e-01 3.958e-02
## gosp 0.0180699 4.609e-03 2.551e-02 -1.270e-02 2.529e-02 -1.785e-02
## haod 0.0085083 2.446e-02 2.793e-03 2.387e-02 5.230e-02 -9.922e-03
## hygl 0.1305498 2.905e-03 1.209e-01 8.285e-02 -5.754e-02 2.925e-02
## hypi -0.0150169 1.725e-02 4.290e-03 -2.482e-02 -1.172e-02 -1.151e-02
## hypo -0.0582026 -3.197e-02 8.701e-02 6.189e-02 -2.330e-02 3.009e-02
## jubu 0.0000000 0.0000e+00 0.0000e+00 0.0000e+00 0.0000e+00 0.0000e+00
## laro -0.1001787 -5.003e-02 2.308e-02 1.384e-02 3.061e-02 -5.441e-02
## ledu 0.0018314 2.583e-02 1.169e-03 1.944e-02 9.125e-03 1.162e-02
## lele 0.0170846 3.174e-03 7.089e-03 5.462e-03 4.878e-04 1.083e-02
## loef 0.0119918 3.267e-03 -7.624e-04 3.625e-03 -1.161e-02 1.227e-02
## misp -0.0677312 -9.016e-02 5.996e-02 -1.150e-01 -4.657e-02 -5.294e-02
## mite -0.0485321 2.148e-01 -5.397e-02 -5.310e-02 3.580e-02 5.387e-02
## momo 0.0000000 0.0000e+00 0.0000e+00 0.0000e+00 0.0000e+00 0.0000e+00
## mopa 0.0000000 0.0000e+00 0.0000e+00 0.0000e+00 0.0000e+00 0.0000e+00
## niro -0.0157499 3.081e-02 -8.896e-03 1.270e-02 -3.637e-03 -2.570e-03
## omco 0.0135532 -9.910e-03 -1.428e-02 4.070e-02 -3.485e-04 1.916e-02
## orsp -0.0106186 -1.238e-03 -5.492e-03 -1.036e-02 5.195e-05 -5.661e-03
## pala 0.0086051 7.583e-03 1.074e-02 5.817e-03 9.230e-03 2.952e-03
## peai 0.1028116 1.736e-01 -2.005e-01 -3.081e-02 1.369e-02 2.640e-02
## pedu 0.0121694 1.072e-02 1.520e-02 8.226e-03 1.305e-02 4.174e-03
## phsu -0.0111369 2.179e-02 -6.290e-03 8.980e-03 -2.571e-03 -1.818e-03
## plde 0.0000000 0.0000e+00 0.0000e+00 0.0000e+00 0.0000e+00 0.0000e+00
## poar 0.0027883 -1.889e-02 1.721e-02 -6.389e-02 -1.656e-02 6.553e-03
## poca 0.0062046 8.808e-02 -6.471e-02 -2.795e-03 -6.767e-02 1.822e-02
## pocap 0.0045075 8.631e-03 3.229e-02 -3.460e-03 2.430e-02 1.976e-02
## poce -0.0106186 -1.238e-03 -5.492e-03 -1.036e-02 5.195e-05 -5.661e-03
## pogn 0.0000000 0.0000e+00 0.0000e+00 0.0000e+00 0.0000e+00 0.0000e+00
## pole -0.0298641 1.578e-01 3.998e-02 -1.008e-01 5.034e-03 -8.070e-03
## pomu 0.1993083 -3.328e-01 -2.100e-01 8.524e-03 8.303e-02 -6.531e-02
## pter 0.0313036 -8.137e-03 -1.022e-02 -6.534e-03 8.000e-03 -2.134e-02
## ptga 0.0518059 6.918e-03 1.242e-01 -6.995e-02 -3.267e-02 1.731e-02
## ptob 0.0000000 0.0000e+00 0.0000e+00 0.0000e+00 0.0000e+00 0.0000e+00
## rhla -0.0192925 8.336e-02 -3.393e-02 -1.024e-01 7.481e-02 -1.291e-02
## rhpy -0.0745613 1.446e-02 -4.187e-02 -1.141e-02 -1.525e-01 -5.915e-02
## rhsp -0.0241808 1.203e-02 1.080e-01 8.131e-02 -8.713e-02 9.129e-02
## ry 0.0000000 0.0000e+00 0.0000e+00 0.0000e+00 0.0000e+00 0.0000e+00
## scna 0.0187550 6.243e-03 1.596e-02 -2.118e-02 -3.013e-02 -1.789e-02
## sino -0.0150169 -3.738e-02 1.412e-02 -2.351e-02 -7.513e-03 8.236e-03
## sool -0.0138136 -1.380e-02 -7.854e-03 7.919e-03 -7.863e-03 -1.252e-02
## stfi -0.0194231 -8.594e-03 6.600e-03 -1.169e-02 1.458e-02 -5.568e-03
## stpi 0.0000000 0.0000e+00 0.0000e+00 0.0000e+00 0.0000e+00 0.0000e+00
## thma 0.0000000 0.0000e+00 0.0000e+00 0.0000e+00 0.0000e+00 0.0000e+00

```

```

## trcy -0.2242559 2.113e-01 1.067e-02 1.038e-01 8.545e-03 -2.070e-01
## tris 0.0402117 -3.209e-02 1.911e-02 -8.204e-03 -1.746e-02 -5.635e-02
## tror -0.1544269 -6.830e-02 -3.490e-02 -8.891e-02 5.911e-02 1.022e-02
## trpi -0.0437537 1.461e-02 -2.404e-02 7.331e-02 2.214e-02 1.813e-03
## waac -0.1356532 -3.713e-02 4.244e-02 -4.687e-02 6.527e-03 -2.002e-02
## wagr -0.0065153 -8.588e-03 6.302e-03 2.518e-03 2.094e-03 3.230e-02
## x    -0.0133788 -5.871e-03 2.583e-02 -1.927e-04 4.224e-02 1.243e-03
##
##
## Site scores (weighted sums of species scores)
##
##          RDA1      PC1      PC2      PC3      PC4      PC5
## row1   -0.26449 -0.11731  0.072142 -0.1786127  0.003365 -0.186582
## row2    0.32774 -0.25141 -0.315084 -0.1260442 -0.262245 -0.115106
## row3   -0.46922  0.11517 -0.268870 -0.3406530  0.698008 -0.329868
## row4    0.27897  0.21540 -0.700246 -0.4535292 -0.020427  0.244714
## row5   -0.39084  0.31262 -0.122452  0.2712654 -0.103664 -0.151995
## row6    0.25593  0.23780  0.111247  0.0077685 -0.106470 -0.024464
## row7   -0.28933 -0.30912  0.156394 -0.0639535 -0.630565  0.256295
## row8    0.22871  0.03594  0.165376  0.0061877  0.158766  0.186223
## row9   -0.14623  0.13592 -0.190179  0.4505984 -0.907619  0.085541
## row10   0.63059 -0.41985 -0.221834  0.3855205  0.244325  0.427107
## row11   0.01622 -0.02003  0.033700 -0.0067724  0.106785  0.178492
## row12   0.21309  0.29790  0.422079  0.2285040  0.362588  0.115951
## row13   -0.43323 -0.28242  0.464241 -0.0511444  0.557801  0.003577
## row14   0.04211  0.04939  0.393486 -0.1291351 -0.100648 -0.689887
## row15   -0.20945 -0.53535 -0.191302 -0.4967942  0.149392 -0.088724
## row16   0.48864  0.11795  0.004154  0.4405109  0.555465 -0.021861
## row17   -0.29523  0.06965 -0.199387  0.6302322  0.230438  0.049894
## row18   0.45621  0.06090  0.509639 -0.8875723 -0.461571  0.207254
## row19   -0.49980  0.66131 -0.190927  0.2725541 -0.078050 -0.055167
## row20   0.45815  0.10541  0.194447  0.2727072  0.304887  0.350394
## row21   -0.15821 -0.13366 -0.241084 -0.0026471  0.350202 -0.173006
## row22   0.08433  0.09447  0.730090 -0.1782412  0.173954  0.148209
## row23   -0.87372  0.38836  0.096577 -0.5587161 -0.263845 -0.259079
## row24   0.54389 -0.30905 -0.033889  0.1578082  0.214942  0.339469
## row25   -0.19692 -0.33779 -0.192198  0.1937864 -0.192412 -0.306358
## row26   0.31799 -0.17579 -0.145749 -0.0141532 -0.348512  0.314036
## row27   -0.30510 -0.17289 -0.395816  0.0233742 -0.006757  0.371098
## row28   0.18924  0.16647  0.055446  0.1471508 -0.628134 -0.876158
## row29   -0.59762  0.26775  0.658661  0.2113891 -0.436792  0.732128
## row30   0.25517  0.51838 -0.228572  0.1495511 -0.009184 -0.319855
## row31   -0.23382 -0.55322  0.680356  0.9493292 -0.119289 -0.517903
## row32   0.26309  0.09208 -0.021492  0.1021932 -0.327159  0.345909
## row33   -0.22411 -0.84145  0.317819 -0.5291453 -0.169109  0.185397
## row34   0.40068 -0.47569 -0.310134 -0.2951499  0.152405 -0.304166
## row35   -0.26668  0.40753 -0.725183  0.0061752 -0.074115 -0.128422
## row36   0.40119 -0.09516 -0.085593  0.2324789  0.021757 -0.289300
## row37   -0.68266  0.77489  0.513757 -0.4592464  0.421879 -0.215783
## row38   0.52814 -0.57026 -0.164432  0.0240354  0.290319 -0.377254
## row39   -0.49625 -0.03940 -0.174828 -0.3297816  0.001654 -0.180206
## row40   0.14586  0.27510  0.056997 -0.0715832  0.069898 -0.121915
## row41   -0.08197  0.20944 -0.101421  0.0087625  0.462695  0.730672
## row42   0.58896  0.03001 -0.415936  0.0009919 -0.284957  0.460698

```

```

##  

##  

## Site constraints (linear combinations of constraining variables)  

##  

##          RDA1      PC1      PC2      PC3      PC4      PC5  

## row1   -0.338 -0.11731  0.072142 -0.1786127  0.003365 -0.186582  

## row2    0.338 -0.25141 -0.315084 -0.1260442 -0.262245 -0.115106  

## row3   -0.338  0.11517 -0.268870 -0.3406530  0.698008 -0.329868  

## row4    0.338  0.21540 -0.700246 -0.4535292 -0.020427  0.244714  

## row5   -0.338  0.31262 -0.122452  0.2712654 -0.103664 -0.151995  

## row6    0.338  0.23780  0.111247  0.0077685 -0.106470 -0.024464  

## row7   -0.338 -0.30912  0.156394 -0.0639535 -0.630565  0.256295  

## row8    0.338  0.03594  0.165376  0.0061877  0.158766  0.186223  

## row9   -0.338  0.13592 -0.190179  0.4505984 -0.907619  0.085541  

## row10   0.338 -0.41985 -0.221834  0.3855205  0.244325  0.427107  

## row11   -0.338 -0.02003  0.033700 -0.0067724  0.106785  0.178492  

## row12   0.338  0.29790  0.422079  0.2285040  0.362588  0.115951  

## row13   -0.338 -0.28242  0.464241 -0.0511444  0.557801  0.003577  

## row14   0.338  0.04939  0.393486 -0.1291351 -0.100648 -0.689887  

## row15   -0.338 -0.53535 -0.191302 -0.4967942  0.149392 -0.088724  

## row16   0.338  0.11795  0.004154  0.4405109  0.555465 -0.021861  

## row17   -0.338  0.06965 -0.199387  0.6302322  0.230438  0.049894  

## row18   0.338  0.06090  0.509639 -0.8875723 -0.461571  0.207254  

## row19   -0.338  0.66131 -0.190927  0.2725541 -0.078050 -0.055167  

## row20   0.338  0.10541  0.194447  0.2727072  0.304887  0.350394  

## row21   -0.338 -0.13366 -0.241084 -0.0026471  0.350202 -0.173006  

## row22   0.338  0.09447  0.730090 -0.1782412  0.173954  0.148209  

## row23   -0.338  0.38836  0.096577 -0.5587161 -0.263845 -0.259079  

## row24   0.338 -0.30905 -0.033889  0.1578082  0.214942  0.339469  

## row25   -0.338 -0.33779 -0.192198  0.1937864 -0.192412 -0.306358  

## row26   0.338 -0.17579 -0.145749 -0.0141532 -0.348512  0.314036  

## row27   -0.338 -0.17289 -0.395816  0.0233742 -0.006757  0.371098  

## row28   0.338  0.16647  0.055446  0.1471508 -0.628134 -0.876158  

## row29   -0.338  0.26775  0.658661  0.2113891 -0.436792  0.732128  

## row30   0.338  0.51838 -0.228572  0.1495511 -0.009184 -0.319855  

## row31   -0.338 -0.55322  0.680356  0.9493292 -0.119289 -0.517903  

## row32   0.338  0.09208 -0.021492  0.1021932 -0.327159  0.345909  

## row33   -0.338 -0.84145  0.317819 -0.5291453 -0.169109  0.185397  

## row34   0.338 -0.47569 -0.310134 -0.2951499  0.152405 -0.304166  

## row35   -0.338  0.40753 -0.725183  0.0061752 -0.074115 -0.128422  

## row36   0.338 -0.09516 -0.085593  0.2324789  0.021757 -0.289300  

## row37   -0.338  0.77489  0.513757 -0.4592464  0.421879 -0.215783  

## row38   0.338 -0.57026 -0.164432  0.0240354  0.290319 -0.377254  

## row39   -0.338 -0.03940 -0.174828 -0.3297816  0.001654 -0.180206  

## row40   0.338  0.27510  0.056997 -0.0715832  0.069898 -0.121915  

## row41   -0.338  0.20944 -0.101421  0.0087625  0.462695  0.730672  

## row42   0.338  0.03001 -0.415936  0.0009919 -0.284957  0.460698  

##  

##  

## Biplot scores for constraining variables  

##  

##          RDA1 PC1 PC2 PC3 PC4 PC5  

## initopen     1   0   0   0   0   0
##
```

```

##  

## Centroids for factor constraints  

##  

##          RDA1 PC1 PC2 PC3 PC4 PC5  

## initlog -0.338  0   0   0   0   0  

## initopen  0.338  0   0   0   0   0

RsquareAdj(trt_Frac)$adj.r.squared #explanatory power

## [1] 0.05628909

anova.cca(trt_Frac) ## this tells us if first condition, init, significantly contributes to overall var

## Permutation test for rda under reduced model
## Permutation: free
## Number of permutations: 999
##
## Model: rda(formula = ass.rel.t012 ~ init + Condition(block + time))
##          Df Variance      F Pr(>F)
## Model     1 0.034157 3.9233 0.001 ***
## Residual 32 0.278596
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

### extracting species scores and plotting
# species scores
species.scores<-as.data.frame(vegan::scores(ass.rel.t012_NMS,"species")) ## some species don't have scores
species.scores$species<-rownames(species.scores)

### NMDS 1 and 2
log<-mds_scores_t012[mds_scores_t012$treatment == "log", ][chull(mds_scores_t012[mds_scores_t012$treatment == "log", c("NMDS1", "NMDS2")]), ]  

open<-mds_scores_t012[mds_scores_t012$treatment == "open", ][chull(mds_scores_t012[mds_scores_t012$treatment == "open", c("NMDS1", "NMDS2")]), ]  

hulldat<-rbind(log,open)

options(ggrepel.max.overlaps = Inf)

nmds.plot.sp <- ggplot()+
  theme_bw()+
  theme(panel.background = element_blank(),
        panel.grid.major = element_blank(), #remove major-grid labels
        panel.grid.minor = element_blank(), #remove minor-grid labels
        plot.background = element_blank(),
        axis.text = element_text(size = 15),
        axis.title=element_text(size=20),
        legend.title=element_text(size=20),
        legend.text=element_text(size=15))+  

  geom_text_repel(data=species.scores, aes(NMDS1, NMDS2, label=species), alpha=0.9, size=5, col='darkgreen')+
  geom_polygon(data=hulldat, aes(NMDS1, NMDS2, fill=treatment, group=treatment), alpha=0.3)+scale_fill_manual(values=c("NMDS1"="red", "NMDS2"="blue"))

```

```

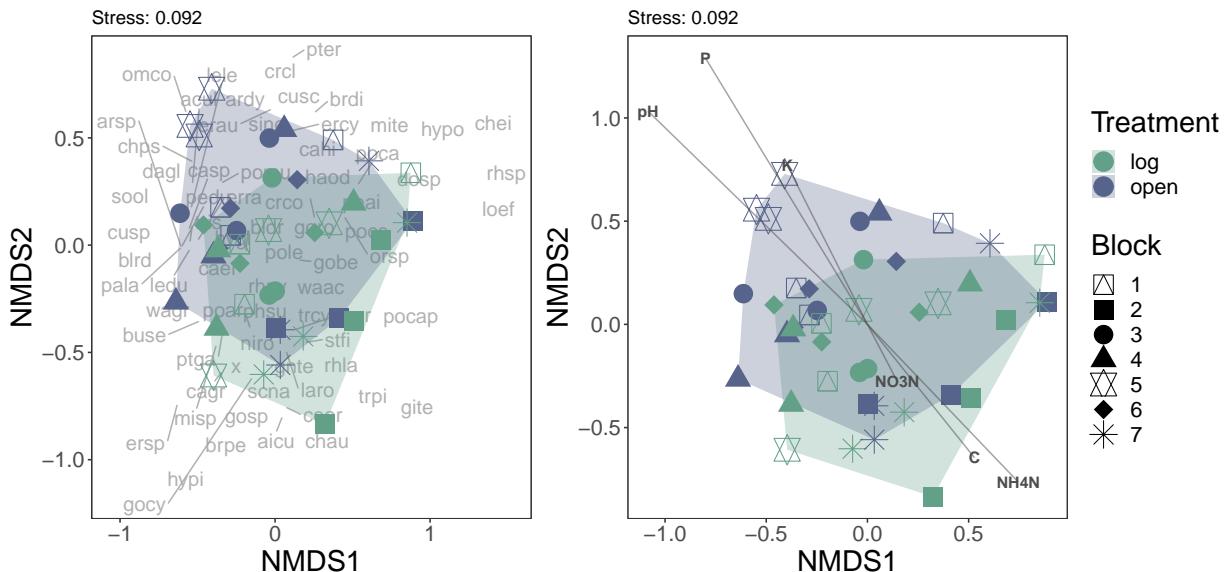
geom_point(data=mds_scores_t012, aes(NMDS1, NMDS2, shape=block, col=treatment), size=6)+ scale_shape_manual(values=c(1,2,3,4,5,6,7), name="Treatment")+
  labs(title=paste0("Stress: ", round(ass.rel.t012_NMS$stress,3)))

nmds.plot.nutrient <- ggplot()+
  theme_bw()+
  theme(panel.background = element_blank(),
        panel.grid.major = element_blank(), #remove major-grid labels
        panel.grid.minor = element_blank(), #remove minor-grid labels
        plot.background = element_blank(),
        axis.text = element_text(size = 15),
        axis.title=element_text(size=20),
        legend.title=element_text(size=20),
        legend.text=element_text(size=15))+
  geom_polygon(data=hulldat, aes(NMDS1, NMDS2, fill=treatment), alpha=0.3)+scale_fill_manual(values=c("log"="#63A088", "open"="#56638A"), name="Treatment")+
  geom_point(data=mds_scores_t012, aes(NMDS1, NMDS2, shape=block, col=treatment), size=6)+ scale_shape_manual(values=c(1,2,3,4,5,6,7), name="Treatment")+
  geom_segment(aes(x = 0, y = 0, xend = NMDS1, yend = NMDS2),
               data = en_coord_cont, size = 0.5, alpha = 0.5, colour = "grey30") +
  geom_text(data = en_coord_cont, aes(x = NMDS1, y = NMDS2), colour = "grey30",
            fontface = "bold", label = row.names(en_coord_cont))+ 
  labs(title=paste0("Stress: ", round(ass.rel.t012_NMS$stress,3)))

(nmds.plot.sp + theme(legend.position = "none")) + nmds.plot.nutrient + plot_layout(guides = "collect")

```

Plant composition: 2020



Interpretation

Q3 Are/how are plant species performances affected by proximity to fallen logs?

Overview of results 2021

There are three response variables: presence (binomial), count (truncated poisson), and per capita biomass (gaussian).

Presence - The best fit model for presence is a tie between physical barrier and the additive model of physical barrier + nutrient island ($\Delta AIC=0.32$). I present the model analysis of the additive model. - Both TRCY and TROR are significantly more likely to be present where there is a physical barrier as compared to in the open (TRCY: $p=0.03$, TROR: $p=0.01$) - Nutrient island does not significantly explain variation in presence/absence in any species. However, all species trend to having higher probability of presence in log-legacy plots.

Count - The best fit model for count is nutrient island. - Nutrient island does not strongly significantly explain variation in presence/absence in any species. TRCY is marginally significantly affected by nutrient island, where there are more TRCY in plots where there is a ‘log legacy’ (there had been a decomposing log in the plot before the experiment began)

Biomass - The best fit model for per capita biomass is the nutrient island, then physical barrier island, then the additive model. They are all within 2 AICc points of each other ($\Delta AIC=1.55$ and 1.76 , respectively). I present the model analysis of the additive model. - Nutrient island significantly explains variation GORO biomass, where there are larger GORO plants in plots where there is a ‘log legacy’ ($p=0.03$). - Physical barrier does not explain variation in biomass for any species.

2022

There are again three response variables: presence (binomial), count (truncated poisson), and per capita biomass (gaussian).

Presence - The best fit model for presence is a tie between physical barrier and the additive model of physical barrier + nutrient island ($\Delta AIC=0.24$). I present the model analysis of the additive model. - Both TRCY and TROR are significantly more likely to be present where there is a physical barrier as compared to in the open (TRCY: $p=0.007$, TROR: $p=0.03$) - Nutrient island significantly explains the presence/absence of TRCY, where TRCY is more likely to be present in log legacy plots as compared to open legacy plots ($p=0.02$)

Count - The best fit model for count is a tie between nutrient island and physical barrier. I present the model analysis of the nutrient island model. - Nutrient island does not strongly significantly explain variation in presence/absence in any species.

Biomass - The best fit model for per capita biomass is the nutrient island, then the additive model, then the physical barrier model. They are all within 1 AICc points of each other ($\Delta AIC=0.31$ and 0.46 , respectively). I present the model analysis of the additive model. - Nutrient island significantly explains variation in TROR biomass, where log legacy plots have larger TROR as compared to open legacy plots ($p=0.04$) - Physical barrier significantly explains variation in TROR biomass, where plots in the open have larger TROR than plots near a physical barrier (either log or PVC pipe) ($p=0.02$)

Statistical Methods The basic approach is to analyse count and biomass data from 2021 and 2022 sowing experiment. We sowed 15 seeds for each species into 16 plots in each of the seven blocks. There are 6 plot type treatments. The gap and open treatments each have four replicates per block, and the insitu_log, insitu_pvc, open_with_log, and open_with_pvc each have two replicates per block. *note: For each treatment of plot type, there is also a dispersal treatment, but I do not analyse that here (yet).* - For **count data** I use a glmmTMB to run a generalized linear mixed effects model approach to analyse the data. The hurdle model approach I am using is to first code presence/absence as 0 or 1 (1 being any nonzero count value) and run this analysis as a binomial regression. Depending on the model fit and residual dispersion (using DHARMA), I then either run a truncated poisson or truncated negative binomial regression on the count data. Though I did run alternative versions of hurdle models that predict count while accounting for zero inflation, I felt that analyzing the presence/absence, and then analyzing count, may be revealing more of the biology of the system, where presence/absence was distinctly affected by treatments while count was less so. - For **biomass data** I use linear mixed effects model to analyse the per capita biomass data. I first do a $\log(1+n)$ transformation on the per capita biomass, then analyze. I chose to do per capita biomass

because of recalcitrant (!!) residual dispersion in total biomass, even after attempts at log and square root transformations. I chose to do a model selection approach, which I'm trying to move away from generally. However, the models I use to analyze the data represent different hypotheses about *why* plant species might perform differently, and so I chose to do a model comparison and selection approach as a way to not only understand how plant species performance is affected by proximity to fallen logs, but also gain insight as to why this may be the case.

N.B.: Model comparison might not be the approach we want, and we may want to run the analysis differently. I'm open to discussion and change on this point.

After model comparison, I use estimated marginal means (package emmeans) on the best fit model and compute the significance of the difference in estimates.

The models I use test the hypotheses from above: Log decomposition creates islands of fertility directly around the fallen log and fallen logs alter the microclimate directly around them by providing shade.

Analysis

```
### doing vital rate analysis

# packages
require(lme4)
require(emmeans)
require(pscl)
require(glmmTMB)
require(tidyr)
require(DHARMa)
require(ggplot2)
require(AICcmodavg)
require(ggpubr)

##### 2021 data #####
# read csv
dat<-read.csv('nplants_data_2021_git.csv', header=T)
dat1<-dat[which(dat$seeding_trt==1),]
dat1$physical_barrier<-as.factor(dat1$physical_barrier)
dat1$block<-as.factor(dat1$block)

##### treatment response: do analysis for count by species #####
dat2<-(dat1[c(1:4,7,11,15,22)]) # these are block, transect, initial, current_plot_type, ngoro, ntrcy, ntror
head(dat2)
```

2021

```
##   block transect initial current_plot_type ngoro ntrcy ntror physical_barrier
## 1     1       1.01      log           gap    3    0    0        0
## 2     1       1.02      log      insitu_log    0    0    0        1
## 3     1       1.03      log      insitu_pvc    1    3    3        1
## 4     1       1.14      log      insitu_log    0    0    0        1
## 5     1       1.17     open        open    0    0    0        0
## 6     1       1.19     open        open    2    0    0        0
```

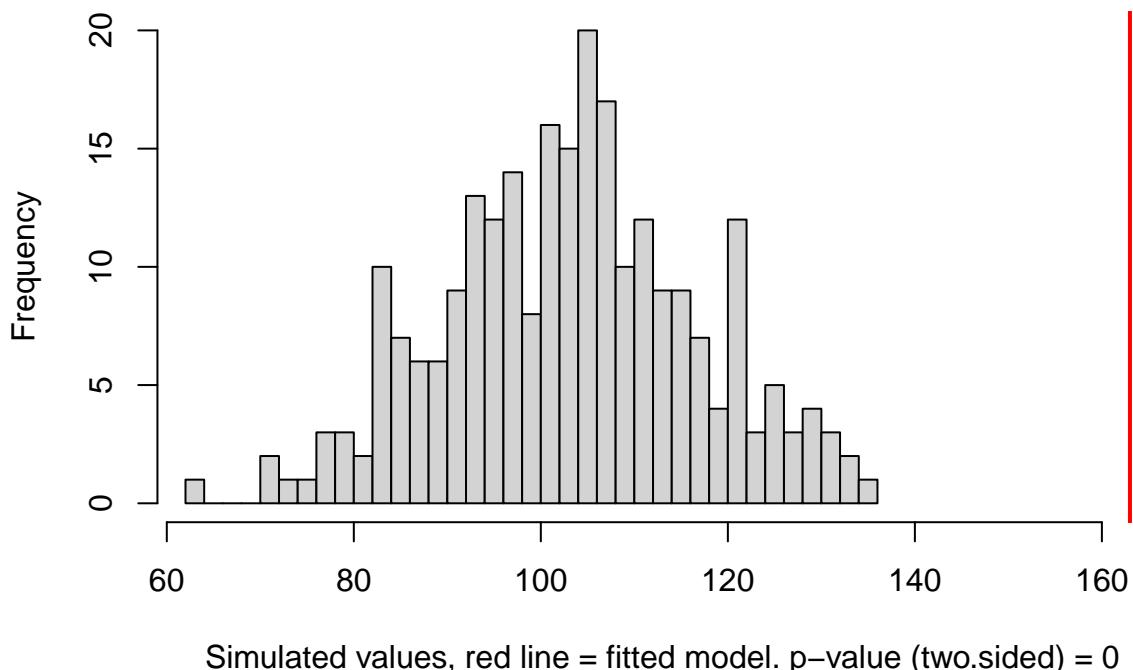
```

countdat<-as.data.frame(dat2 %>% pivot_longer(c(ntror, ngoro, ntrcy)))
countdat$value<-as.numeric(ifelse(countdat$value>15, 15, countdat$value))
countmod<-glmmTMB(value~name*current_plot_type+(1|block), family="poisson", data=countdat)

# test for fit and zero inflation
sim<-simulateResiduals(countmod)
testZeroInflation(sim) # zero-inflated so need something else

```

DHARMA zero-inflation test via comparison to expected zeros with simulation under H0 = fitted model

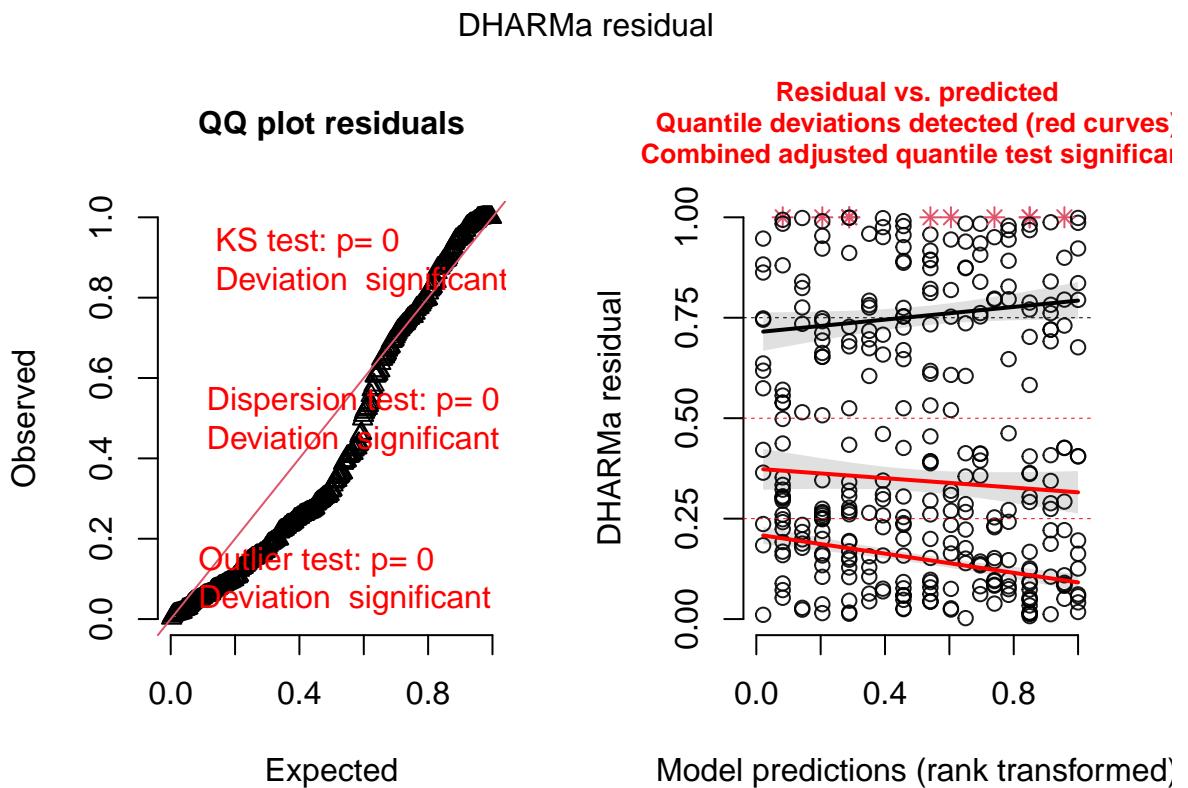


```

## 
## DHARMA zero-inflation test via comparison to expected zeros with
## simulation under H0 = fitted model
## 
## data: simulationOutput
## ratioObsSim = 1.5731, p-value < 2.2e-16
## alternative hypothesis: two.sided

plot(sim)

```



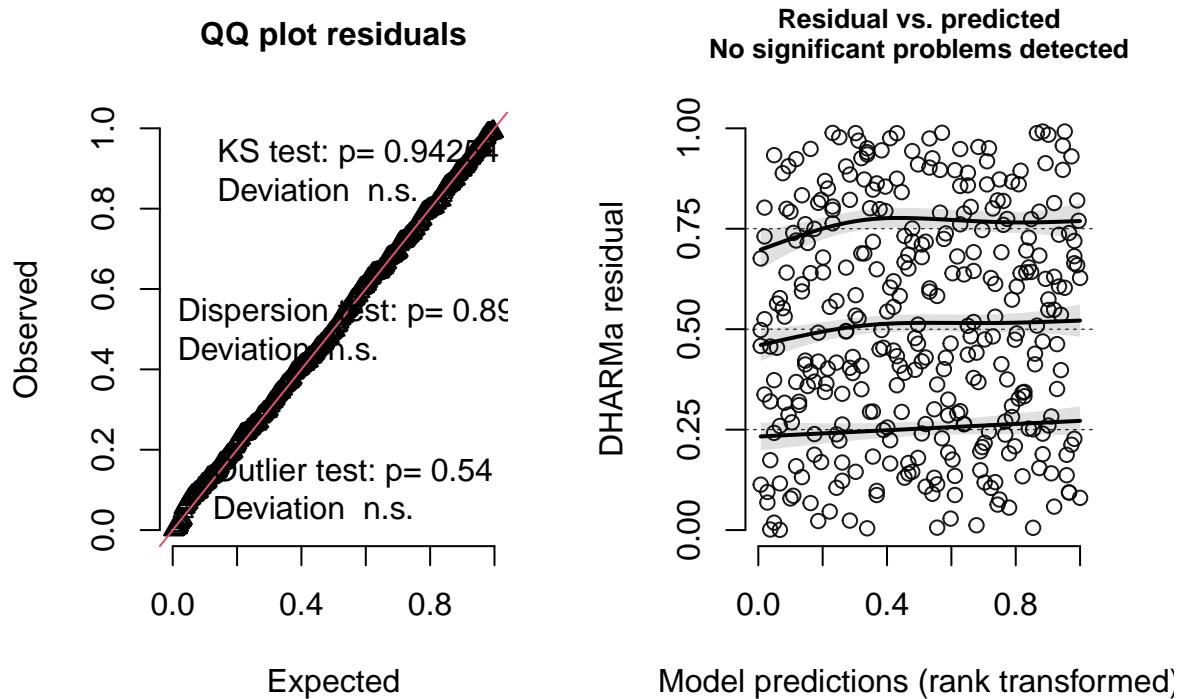
```

## going to do a hurdle model, which assumes a zero is only generated in one way
# https://jsdajournal.springeropen.com/articles/10.1186/s40488-021-00121-4
# #https://stats.stackexchange.com/questions/81457/what-is-the-difference-between-zero-inflated-and-hur

# hurdle model
# using examples as presented here: https://www.biorxiv.org/content/biorxiv/suppl/2017/05/01/132753.DC1/
fit3<-glmmTMB(value~name*current_plot_type+(1|block), ziformula=~., family=nbinom2(), data=countdat)
sim3<-simulateResiduals(fit3)
plot(sim3)

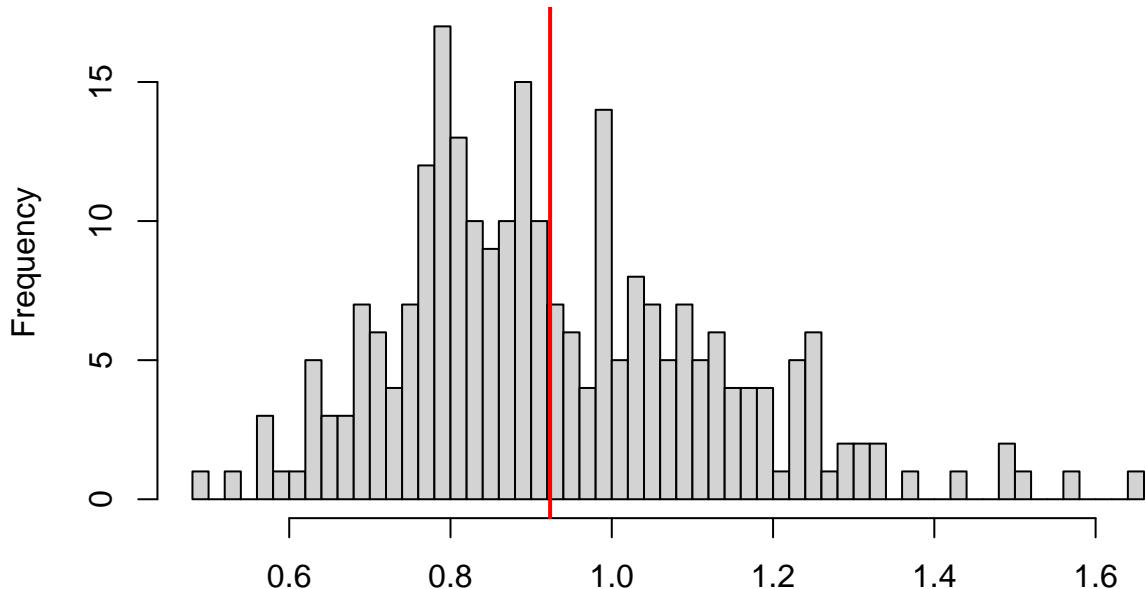
```

DHARMA residual



```
testDispersion(sim3) # looks nice !
```

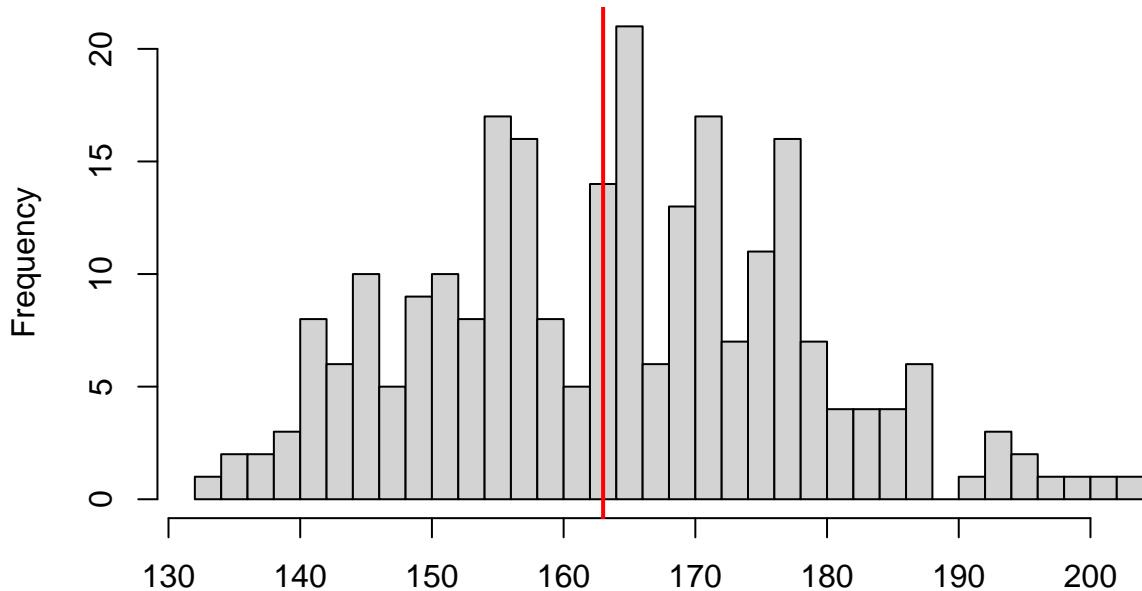
DHARMA nonparametric dispersion test via sd of residuals fitted vs. simulated



Simulated values, red line = fitted model. p-value (two.sided) = 0.896

```
##  
##  DHARMA nonparametric dispersion test via sd of residuals fitted vs.  
##  simulated  
##  
##  data:  simulationOutput  
##  dispersion = 0.99071, p-value = 0.896  
##  alternative hypothesis: two.sided  
  
testZeroInflation(sim3) # looks nice !
```

DHARMA zero-inflation test via comparison to expected zeros with simulation under H0 = fitted model



Simulated values, red line = fitted model. p-value (two.sided) = 0.912

```
## 
## DHARMA zero-inflation test via comparison to expected zeros with
## simulation under H0 = fitted model
## 
## data: simulationOutput
## ratioObsSim = 0.99276, p-value = 0.912
## alternative hypothesis: two.sided

summary(fit3)

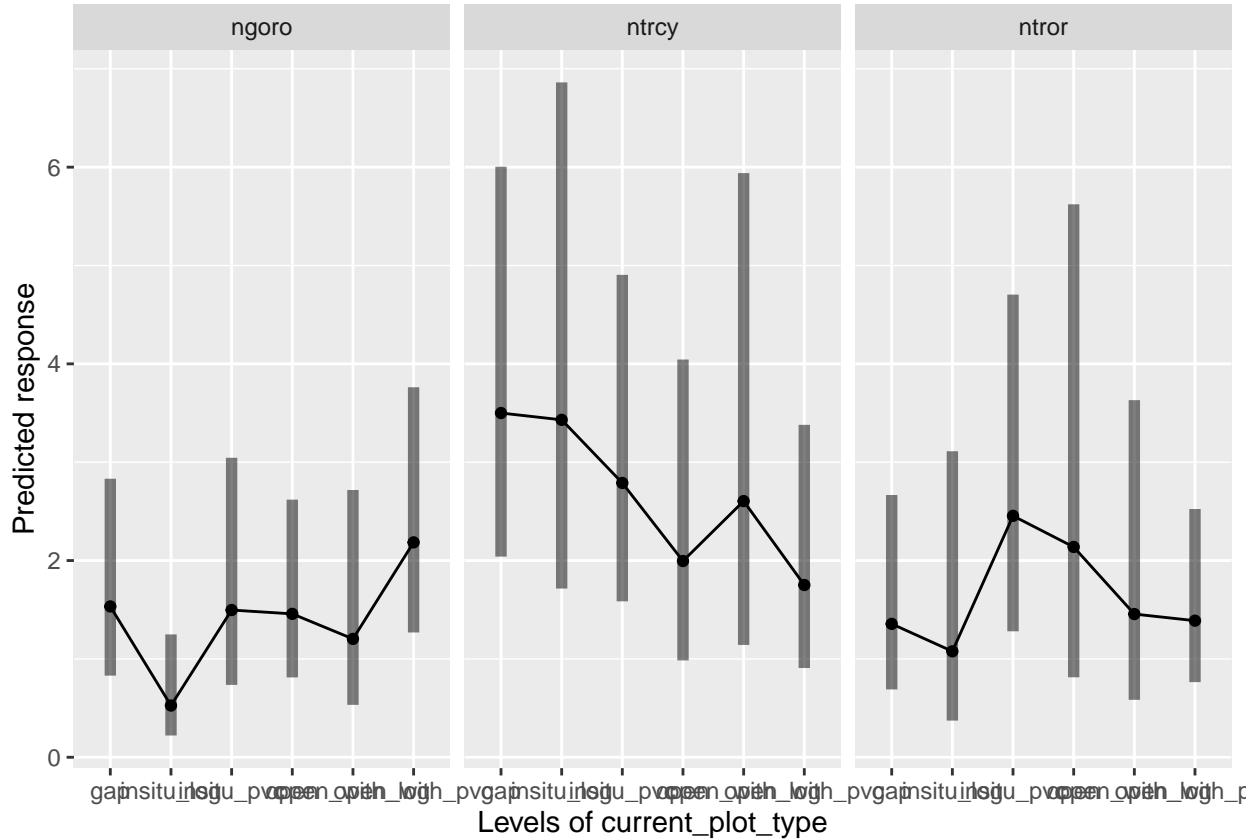
## Family: nbinom2  ( log )
## Formula:           value ~ name * current_plot_type + (1 | block)
## Zero inflation:    ~.
## Data: countdat
## 
##      AIC      BIC   logLik deviance df.resid
##  1101.2  1250.1  -511.6   1023.2      297
## 
## Random effects:
## 
## Conditional model:
## Groups Name        Variance Std.Dev.
## block  (Intercept) 0.03237  0.1799
## Number of obs: 336, groups: block, 7
##
```

```

## Zero-inflation model:
## Groups Name      Variance Std.Dev.
## block (Intercept) 0.9243   0.9614
## Number of obs: 336, groups: block, 7
##
## Dispersion parameter for nbinom2 family (): 1.68
##
## Conditional model:
##                               Estimate Std. Error z value Pr(>|z|)
## (Intercept)                0.42749  0.31299  1.366  0.1720
## namentrcy                  0.82532  0.39283  2.101  0.0356 *
## namentror                 -0.12276  0.44057 -0.279  0.7805
## current_plot_typeinsitu_log -1.07022  0.53312 -2.007  0.0447 *
## current_plot_typeinsitu_pvc -0.02408  0.44823 -0.054  0.9572
## current_plot_typeopen      -0.05040  0.40107 -0.126  0.9000
## current_plot_typeopen_with_log -0.24213  0.50393 -0.480  0.6309
## current_plot_typeopen_with_pvc 0.35404  0.40916  0.865  0.3869
## namentrcy:current_plot_typeinsitu_log 1.05024  0.67754  1.550  0.1211
## namentror:current_plot_typeinsitu_log 0.83983  0.81808  1.027  0.3046
## namentrcy:current_plot_typeinsitu_pvc -0.20308  0.59085 -0.344  0.7311
## namentror:current_plot_typeinsitu_pvc 0.61733  0.63892  0.966  0.3339
## namentrcy:current_plot_typeopen     -0.51151  0.60272 -0.849  0.3961
## namentror:current_plot_typeopen    0.50564  0.70479  0.717  0.4731
## namentrcy:current_plot_typeopen_with_log -0.05338  0.70024 -0.076  0.9392
## namentror:current_plot_typeopen_with_log 0.31312  0.75666  0.414  0.6790
## namentrcy:current_plot_typeopen_with_pvc -1.04638  0.57985 -1.805  0.0711 .
## namentror:current_plot_typeopen_with_pvc -0.33034  0.59667 -0.554  0.5798
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Zero-inflation model:
##                               Estimate Std. Error z value Pr(>|z|)
## (Intercept)                -1.396e+00 1.135e+00 -1.230  0.219
## namentrcy                  9.858e-01 1.131e+00  0.872  0.383
## namentror                 4.034e-01 1.305e+00  0.309  0.757
## current_plot_typeinsitu_log -1.628e+01 5.111e+03 -0.003  0.997
## current_plot_typeinsitu_pvc -2.684e+00 7.986e+00 -0.336  0.737
## current_plot_typeopen      -1.280e+00 2.202e+00 -0.582  0.561
## current_plot_typeopen_with_log -9.202e-01 2.380e+00 -0.387  0.699
## current_plot_typeopen_with_pvc -1.790e+01 4.876e+03 -0.004  0.997
## namentrcy:current_plot_typeinsitu_log 1.551e+01 5.111e+03  0.003  0.998
## namentror:current_plot_typeinsitu_log 1.661e+01 5.111e+03  0.003  0.997
## namentrcy:current_plot_typeinsitu_pvc -5.449e-01 8.249e+00 -0.066  0.947
## namentror:current_plot_typeinsitu_pvc 1.947e+00 8.039e+00  0.242  0.809
## namentrcy:current_plot_typeopen     1.665e+00 2.423e+00  0.687  0.492
## namentror:current_plot_typeopen    3.564e+00 2.581e+00  1.381  0.167
## namentrcy:current_plot_typeopen_with_log 5.747e-01 2.630e+00  0.218  0.827
## namentror:current_plot_typeopen_with_log 8.341e-01 2.920e+00  0.286  0.775
## namentrcy:current_plot_typeopen_with_pvc 1.553e+01 4.876e+03  0.003  0.997
## namentror:current_plot_typeopen_with_pvc -9.760e+00 5.916e+05  0.000  1.000

```

```
emmip(fit3, ~current_plot_type|name, type='response', CI=T)
```



```
# visualize
est<-emmeans(fit3, ~current_plot_type|name, type='response')
pairs(est)
```

```
## name = ngoro:
## contrast                                ratio   SE  df null z.ratio p.value
## gap / insitu_log                          2.916 1.555 Inf   1  2.007  0.3379
## gap / insitu_pvc                         1.024 0.459 Inf   1  0.054  1.0000
## gap / open                               1.052 0.422 Inf   1  0.126  1.0000
## gap / open_with_log                      1.274 0.642 Inf   1  0.480  0.9968
## gap / open_with_pvc                      0.702 0.287 Inf   1 -0.865  0.9547
## insitu_log / insitu_pvc                  0.351 0.198 Inf   1 -1.855  0.4301
## insitu_log / open                          0.361 0.189 Inf   1 -1.943  0.3759
## insitu_log / open_with_log                0.437 0.261 Inf   1 -1.385  0.7365
## insitu_log / open_with_pvc                0.241 0.123 Inf   1 -2.777  0.0611
## insitu_pvc / open                         1.027 0.443 Inf   1  0.061  1.0000
## insitu_pvc / open_with_log                1.244 0.667 Inf   1  0.407  0.9986
## insitu_pvc / open_with_pvc                0.685 0.308 Inf   1 -0.842  0.9596
## open / open_with_log                      1.211 0.588 Inf   1  0.395  0.9988
## open / open_with_pvc                      0.667 0.266 Inf   1 -1.015  0.9131
## open_with_log / open_with_pvc             0.551 0.270 Inf   1 -1.215  0.8297
##
```

```
## name = ntrcy:
```

```

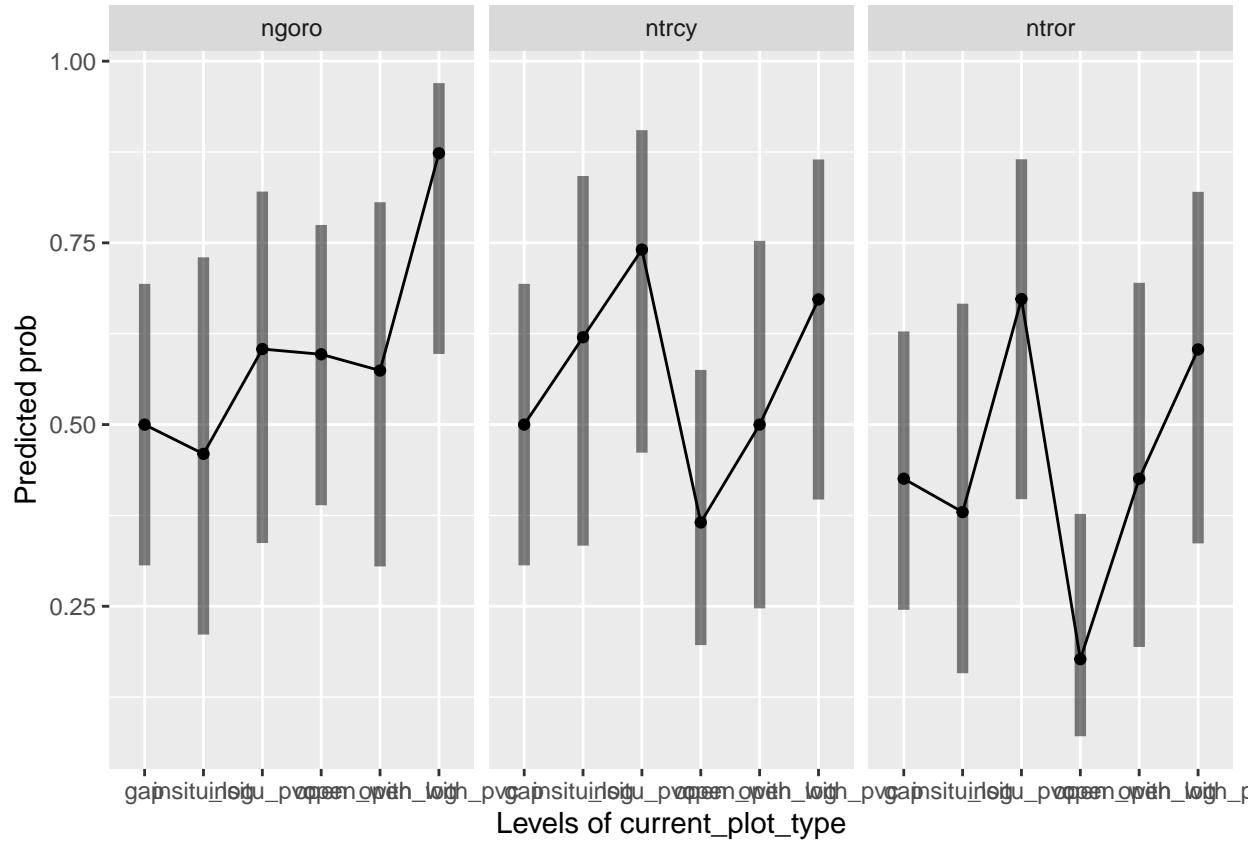
## contrast ratio SE df null z.ratio p.value
## gap / insitu_log 1.020 0.442 Inf 1 0.046 1.0000
## gap / insitu_pvc 1.255 0.483 Inf 1 0.590 0.9917
## gap / open 1.754 0.770 Inf 1 1.280 0.7961
## gap / open_with_log 1.344 0.642 Inf 1 0.619 0.9897
## gap / open_with_pvc 1.998 0.840 Inf 1 1.648 0.5668
## insitu_log / insitu_pvc 1.230 0.545 Inf 1 0.467 0.9972
## insitu_log / open 1.719 0.827 Inf 1 1.127 0.8704
## insitu_log / open_with_log 1.317 0.721 Inf 1 0.503 0.9961
## insitu_log / open_with_pvc 1.959 0.928 Inf 1 1.420 0.7149
## insitu_pvc / open 1.398 0.619 Inf 1 0.756 0.9747
## insitu_pvc / open_with_log 1.071 0.531 Inf 1 0.138 1.0000
## insitu_pvc / open_with_pvc 1.592 0.684 Inf 1 1.083 0.8885
## open / open_with_log 0.766 0.421 Inf 1 -0.485 0.9967
## open / open_with_pvc 1.139 0.543 Inf 1 0.273 0.9998
## open_with_log / open_with_pvc 1.487 0.784 Inf 1 0.752 0.9752
##
## name = ntror:
## contrast ratio SE df null z.ratio p.value
## gap / insitu_log 1.259 0.788 Inf 1 0.368 0.9991
## gap / insitu_pvc 0.553 0.255 Inf 1 -1.283 0.7944
## gap / open 0.634 0.369 Inf 1 -0.783 0.9704
## gap / open_with_log 0.931 0.520 Inf 1 -0.127 1.0000
## gap / open_with_pvc 0.977 0.441 Inf 1 -0.053 1.0000
## insitu_log / insitu_pvc 0.439 0.274 Inf 1 -1.321 0.7735
## insitu_log / open 0.504 0.362 Inf 1 -0.954 0.9323
## insitu_log / open_with_log 0.740 0.520 Inf 1 -0.429 0.9982
## insitu_log / open_with_pvc 0.776 0.477 Inf 1 -0.413 0.9985
## insitu_pvc / open 1.148 0.666 Inf 1 0.238 0.9999
## insitu_pvc / open_with_log 1.686 0.948 Inf 1 0.929 0.9392
## insitu_pvc / open_with_pvc 1.767 0.780 Inf 1 1.291 0.7900
## open / open_with_log 1.469 0.975 Inf 1 0.579 0.9924
## open / open_with_pvc 1.540 0.882 Inf 1 0.753 0.9751
## open_with_log / open_with_pvc 1.048 0.575 Inf 1 0.086 1.0000
##
## P value adjustment: tukey method for comparing a family of 6 estimates
## Tests are performed on the log scale
```

now i'll do a by-hand hurdle model on my own using a truncated negative binomial

zeros and ones

```
countdat$presence<-ifelse(countdat$value==0, 0, 1)
```

```
zerofit<-glmmTMB(presence~name*current_plot_type+(1 | block), family=binomial, data=countdat, REML=FALSE)
emmip(zerofit, ~current_plot_type|name, type='response', CI=T)
```



```
est<-emmeans(zerofit, ~current_plot_type|name, type='response')
pairs(est)
```

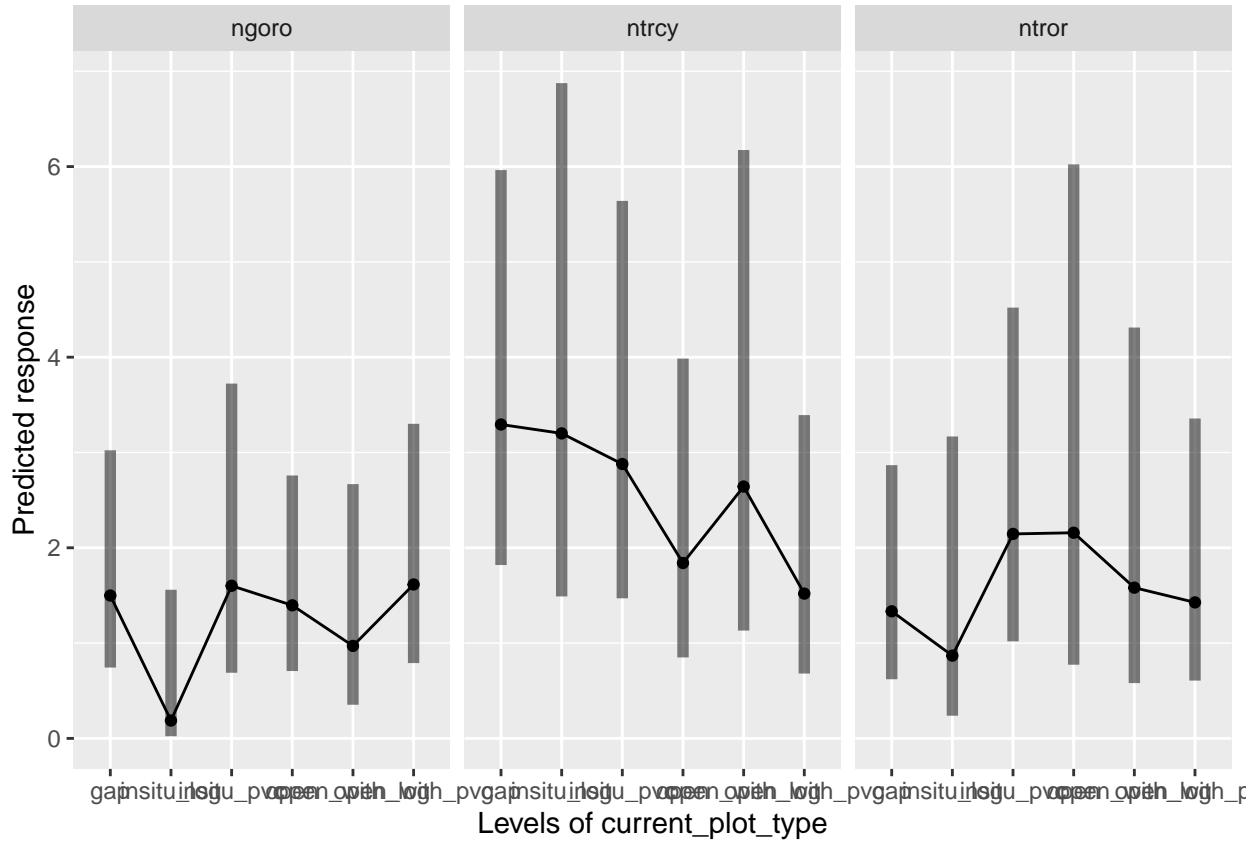
```
## name = ngoro:
## contrast                               odds.ratio    SE   df null z.ratio p.value
## gap / insitu_log                         1.175 0.808 Inf   1  0.235  0.9999
## gap / insitu_pvc                         0.656 0.434 Inf   1 -0.638  0.9882
## gap / open                                0.676 0.376 Inf   1 -0.705  0.9814
## gap / open_with_log                      0.741 0.499 Inf   1 -0.446  0.9978
## gap / open_with_pvc                      0.145 0.125 Inf   1 -2.246  0.2167
## insitu_log / insitu_pvc                  0.558 0.437 Inf   1 -0.745  0.9763
## insitu_log / open                          0.575 0.400 Inf   1 -0.796  0.9684
## insitu_log / open_with_log                0.630 0.500 Inf   1 -0.582  0.9922
## insitu_log / open_with_pvc                0.123 0.118 Inf   1 -2.189  0.2430
## insitu_pvc / open                         1.030 0.690 Inf   1  0.045  1.0000
## insitu_pvc / open_with_log                1.130 0.870 Inf   1  0.159  1.0000
## insitu_pvc / open_with_pvc                0.221 0.207 Inf   1 -1.610  0.5920
## open / open_with_log                      1.097 0.747 Inf   1  0.135  1.0000
## open / open_with_pvc                      0.215 0.186 Inf   1 -1.777  0.4805
## open_with_log / open_with_pvc              0.196 0.185 Inf   1 -1.725  0.5148
##
## name = ntrcy:
## contrast                               odds.ratio    SE   df null z.ratio p.value
## gap / insitu_log                         0.612 0.428 Inf   1 -0.702  0.9818
## gap / insitu_pvc                         0.350 0.248 Inf   1 -1.483  0.6755
```

```

##  gap / open          1.737 0.974 Inf   1   0.984  0.9231
##  gap / open_with_log 1.000 0.668 Inf   1   0.000  1.0000
##  gap / open_with_pvc 0.487 0.331 Inf   1  -1.059  0.8976
##  insitu_log / insitu_pvc 0.571 0.475 Inf   1  -0.673  0.9849
##  insitu_log / open      2.837 2.017 Inf   1   1.467  0.6857
##  insitu_log / open_with_log 1.633 1.304 Inf   1   0.614  0.9900
##  insitu_log / open_with_pvc 0.796 0.642 Inf   1  -0.283  0.9998
##  insitu_pvc / open      4.965 3.577 Inf   1   2.224  0.2265
##  insitu_pvc / open_with_log 2.858 2.306 Inf   1   1.302  0.7842
##  insitu_pvc / open_with_pvc 1.393 1.135 Inf   1   0.407  0.9986
##  open / open_with_log    0.576 0.392 Inf   1  -0.811  0.9656
##  open / open_with_pvc    0.281 0.194 Inf   1  -1.838  0.4411
##  open_with_log / open_with_pvc 0.487 0.381 Inf   1  -0.920  0.9415
##
## name = ntror:
## contrast           odds.ratio   SE  df null z.ratio p.value
##  gap / insitu_log    1.211 0.848 Inf   1   0.273  0.9998
##  gap / insitu_pvc    0.360 0.245 Inf   1  -1.499  0.6651
##  gap / open          3.440 2.188 Inf   1   1.942  0.3763
##  gap / open_with_log 1.000 0.675 Inf   1   0.000  1.0000
##  gap / open_with_pvc 0.487 0.323 Inf   1  -1.083  0.8882
##  insitu_log / insitu_pvc 0.298 0.240 Inf   1  -1.501  0.6637
##  insitu_log / open     2.841 2.185 Inf   1   1.358  0.7523
##  insitu_log / open_with_log 0.826 0.662 Inf   1  -0.238  0.9999
##  insitu_log / open_with_pvc 0.402 0.319 Inf   1  -1.149  0.8608
##  insitu_pvc / open     9.545 7.181 Inf   1   2.999  0.0324
##  insitu_pvc / open_with_log 2.775 2.178 Inf   1   1.301  0.7848
##  insitu_pvc / open_with_pvc 1.351 1.047 Inf   1   0.388  0.9989
##  open / open_with_log   0.291 0.217 Inf   1  -1.656  0.5612
##  open / open_with_pvc   0.142 0.104 Inf   1  -2.652  0.0851
##  open_with_log / open_with_pvc 0.487 0.375 Inf   1  -0.934  0.9377
##
## P value adjustment: tukey method for comparing a family of 6 estimates
## Tests are performed on the log odds ratio scale

#### abundance with a truncated negbinom
countdat$posicounts<-as.numeric(ifelse(countdat$value==0, "NA", countdat$value))
countfit<-glmmTMB(posicounts~name*current_plot_type+(1 | block), family=truncated_nb2(), data=countdat)
emmip(countfit, ~current_plot_type|name, type='response', CI=T)

```



```
est<-emmeans(countfit, ~current_plot_type|name, type='response')
pairs(est)
```

```
## name = ngoro:
## contrast                                ratio    SE  df null z.ratio p.value
## gap / insitu_log                          8.005 8.954 Inf   1  1.860  0.4273
## gap / insitu_pvc                         0.936 0.491 Inf   1 -0.126  1.0000
## gap / open                               1.074 0.491 Inf   1  0.155  1.0000
## gap / open_with_log                      1.545 0.919 Inf   1  0.731  0.9781
## gap / open_with_pvc                      0.928 0.442 Inf   1 -0.157  1.0000
## insitu_log / insitu_pvc                  0.117 0.134 Inf   1 -1.877  0.4162
## insitu_log / open                         0.134 0.149 Inf   1 -1.804  0.4634
## insitu_log / open_with_log                0.193 0.227 Inf   1 -1.399  0.7276
## insitu_log / open_with_pvc               0.116 0.130 Inf   1 -1.923  0.3880
## insitu_pvc / open                        1.147 0.591 Inf   1  0.266  0.9998
## insitu_pvc / open_with_log               1.650 1.056 Inf   1  0.783  0.9705
## insitu_pvc / open_with_pvc              0.992 0.528 Inf   1 -0.016  1.0000
## open / open_with_log                     1.439 0.844 Inf   1  0.620  0.9896
## open / open_with_pvc                    0.864 0.403 Inf   1 -0.313  0.9996
## open_with_log / open_with_pvc           0.601 0.360 Inf   1 -0.849  0.9582
##
## name = ntrcy:
## contrast                                ratio    SE  df null z.ratio p.value
## gap / insitu_log                          1.029 0.484 Inf   1  0.061  1.0000
## gap / insitu_pvc                          1.144 0.497 Inf   1  0.310  0.9996
```

```

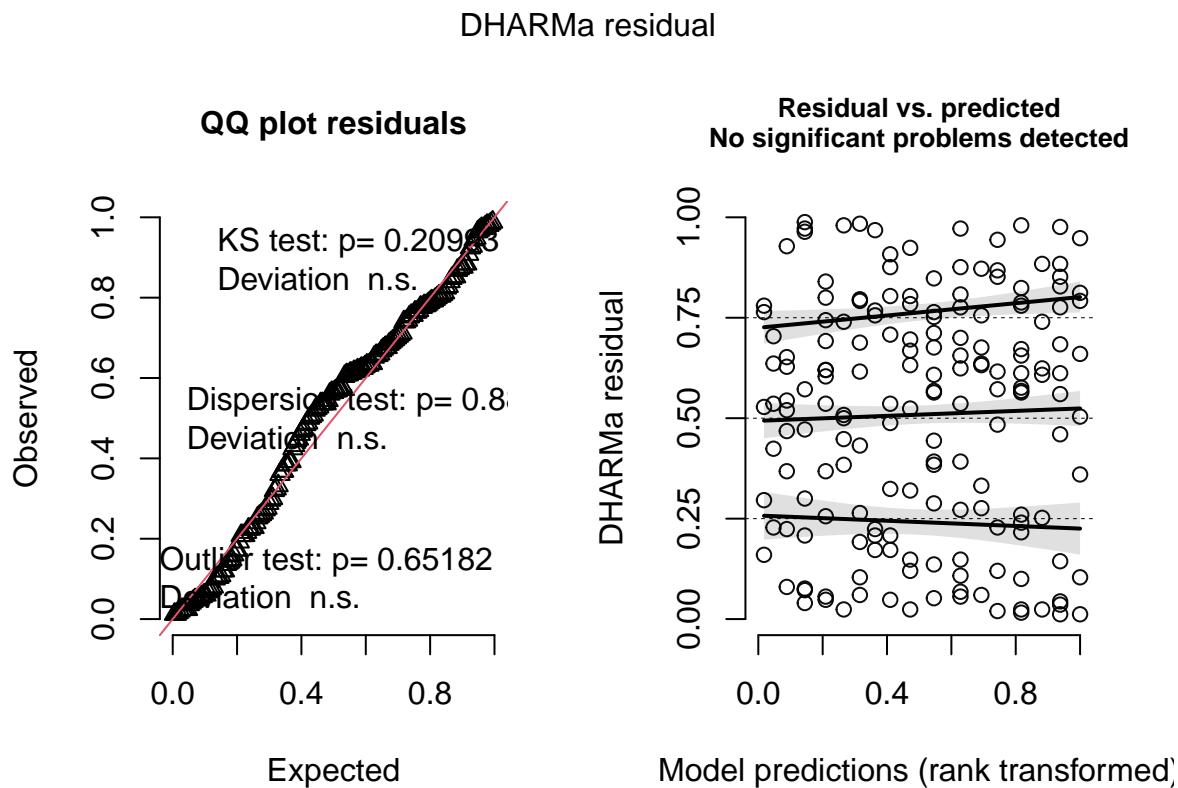
##  gap / open           1.790 0.845 Inf    1   1.233  0.8207
##  gap / open_with_log 1.246 0.628 Inf    1   0.437  0.9980
##  gap / open_with_pvc 2.168 1.045 Inf    1   1.605  0.5953
##  insitu_log / insitu_pvc 1.112 0.549 Inf    1   0.214  0.9999
##  insitu_log / open     1.739 0.913 Inf    1   1.055  0.8991
##  insitu_log / open_with_log 1.211 0.685 Inf    1   0.339  0.9994
##  insitu_log / open_with_pvc 2.107 1.144 Inf    1   1.372  0.7437
##  insitu_pvc / open     1.565 0.768 Inf    1   0.912  0.9436
##  insitu_pvc / open_with_log 1.089 0.579 Inf    1   0.161  1.0000
##  insitu_pvc / open_with_pvc 1.895 0.965 Inf    1   1.255  0.8094
##  open / open_with_log   0.696 0.394 Inf    1   -0.639 0.9881
##  open / open_with_pvc   1.211 0.656 Inf    1   0.354  0.9993
##  open_with_log / open_with_pvc 1.739 0.992 Inf    1   0.970  0.9274
##
## name = ntror:
## contrast                  ratio   SE df null z.ratio p.value
##  gap / insitu_log          1.536 1.136 Inf    1   0.580  0.9923
##  gap / insitu_pvc          0.622 0.320 Inf    1  -0.924  0.9406
##  gap / open                 0.618 0.388 Inf    1  -0.766  0.9731
##  gap / open_with_log       0.843 0.519 Inf    1  -0.277  0.9998
##  gap / open_with_pvc       0.935 0.520 Inf    1  -0.122  1.0000
##  insitu_log / insitu_pvc   0.405 0.299 Inf    1  -1.225  0.8246
##  insitu_log / open          0.402 0.330 Inf    1  -1.110  0.8774
##  insitu_log / open_with_log 0.549 0.446 Inf    1  -0.737  0.9773
##  insitu_log / open_with_pvc 0.608 0.468 Inf    1  -0.646  0.9874
##  insitu_pvc / open          0.994 0.620 Inf    1  -0.009  1.0000
##  insitu_pvc / open_with_log 1.356 0.833 Inf    1   0.497  0.9963
##  insitu_pvc / open_with_pvc 1.503 0.829 Inf    1   0.739  0.9770
##  open / open_with_log       1.364 0.973 Inf    1   0.436  0.9980
##  open / open_with_pvc       1.512 0.999 Inf    1   0.626  0.9891
##  open_with_log / open_with_pvc 1.108 0.717 Inf    1   0.159  1.0000
##
## P value adjustment: tukey method for comparing a family of 6 estimates
## Tests are performed on the log scale
```

```

##### treatment response: do analysis for total biomass by species #####
dat3<-(dat1[c(1:4,10,14,18,22)])
totwtdat<-as.data.frame(dat3 %>% pivot_longer(c(wt_max15_goro, wt_max15_tror, wt_max15_trcy)))
totwtdat$log_wt<-log(totwtdat$value) # log transform the weight data to get it normal looking

# model
totwtmod<-lmer(log_wt~name*current_plot_type+(1|block), data=totwtdat, REML=FALSE)

# test for fit, looks pretty good
sim<-simulateResiduals(totwtmod)
plot(sim)
```



```
# model summary
summary(totwtmod)
```

```
## Linear mixed model fit by maximum likelihood . t-tests use Satterthwaite's
##   method [lmerModLmerTest]
## Formula: log_wt ~ name * current_plot_type + (1 | block)
##   Data: totwtdat
##
##      AIC      BIC  logLik deviance df.resid
##      586.5    649.6   -273.2     546.5     153
##
## Scaled residuals:
##    Min     1Q  Median     3Q    Max
## -2.3315 -0.7238  0.1635  0.6845  2.3280
##
## Random effects:
##   Groups   Name        Variance Std.Dev.
##   block    (Intercept) 0.04225  0.2055
##   Residual             1.34718  1.1607
##   Number of obs: 173, groups: block, 7
##
## Fixed effects:
##   (Intercept)          Estimate Std. Error
##   namewt_max15_trcy -2.143222  0.320185
##   namewt_max15_tror -0.001833  0.439269
##   namewt_max15_tror -0.718937  0.456831
```

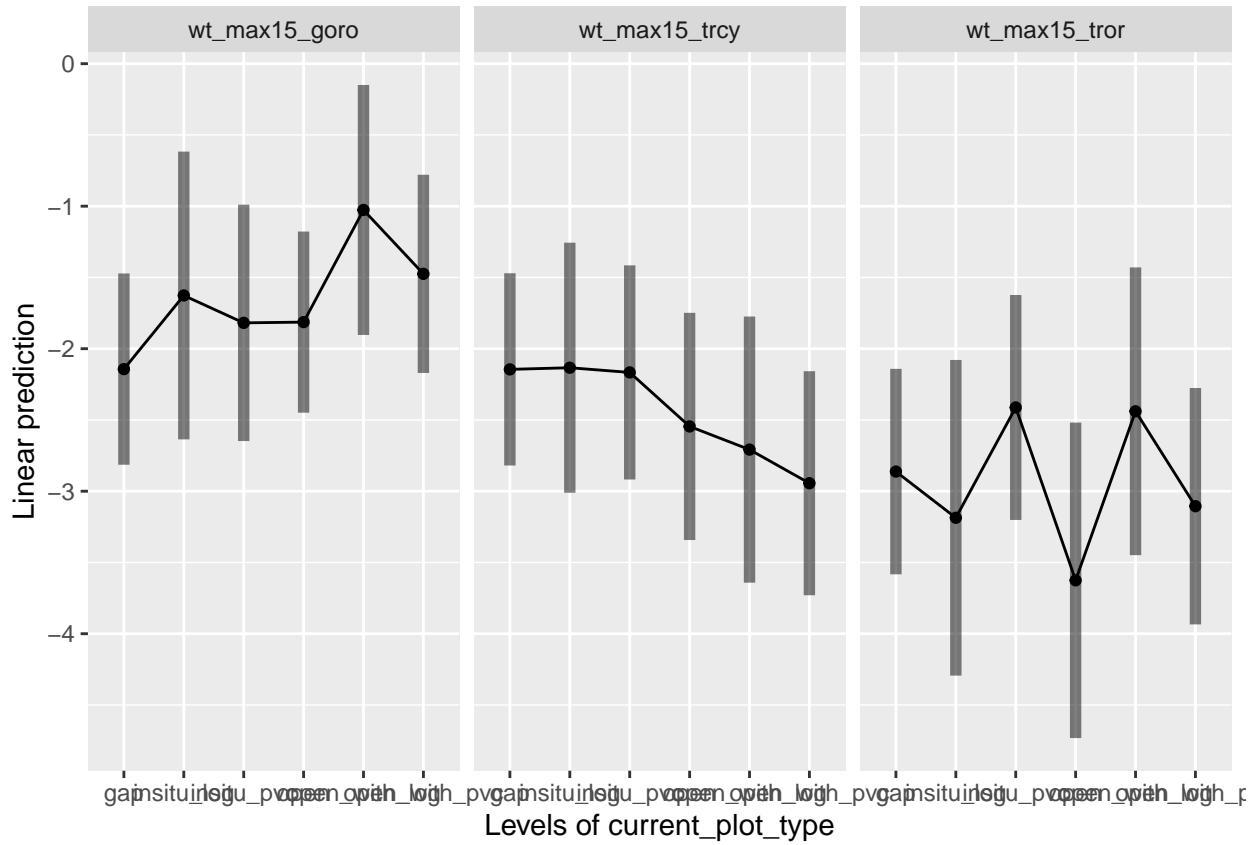
```

## current_plot_typeinsitu_log          0.516663  0.568314
## current_plot_typeinsitu_pvc         0.324586  0.496809
## current_plot_typeopen               0.329858  0.425589
## current_plot_typeopen_with_log     1.116735  0.515346
## current_plot_typeopen_with_pvc     0.668390  0.448068
## namewt_max15_trcy:current_plot_typeinsitu_log -0.505102  0.766320
## namewt_max15_tror:current_plot_typeinsitu_log -0.840945  0.839782
## namewt_max15_trcy:current_plot_typeinsitu_pvc -0.346022  0.683114
## namewt_max15_tror:current_plot_typeinsitu_pvc  0.125206  0.703008
## namewt_max15_trcy:current_plot_typeopen      -0.730280  0.644938
## namewt_max15_tror:current_plot_typeopen      -1.093074  0.751193
## namewt_max15_trcy:current_plot_typeopen_with_log -1.679654  0.745031
## namewt_max15_tror:current_plot_typeopen_with_log -0.693597  0.778032
## namewt_max15_trcy:current_plot_typeopen_with_pvc -1.467203  0.657613
## namewt_max15_tror:current_plot_typeopen_with_pvc -0.911378  0.681196
##
##                                     df t value Pr(>|t|)
## (Intercept)                   140.850563 -6.694 4.77e-10
## namewt_max15_trcy            167.738562 -0.004 0.9967
## namewt_max15_tror            167.112865 -1.574 0.1174
## current_plot_typeinsitu_log 169.260569  0.909 0.3646
## current_plot_typeinsitu_pvc 168.135800  0.653 0.5144
## current_plot_typeopen        168.168196  0.775 0.4394
## current_plot_typeopen_with_log 168.175199  2.167 0.0316
## current_plot_typeopen_with_pvc 168.400310  1.492 0.1376
## namewt_max15_trcy:current_plot_typeinsitu_log 167.901796 -0.659 0.5107
## namewt_max15_tror:current_plot_typeinsitu_log 168.255431 -1.001 0.3181
## namewt_max15_trcy:current_plot_typeinsitu_pvc 168.316088 -0.507 0.6131
## namewt_max15_tror:current_plot_typeinsitu_pvc 167.790062  0.178 0.8589
## namewt_max15_trcy:current_plot_typeopen       170.379238 -1.132 0.2591
## namewt_max15_tror:current_plot_typeopen       168.159556 -1.455 0.1475
## namewt_max15_trcy:current_plot_typeopen_with_log 168.023484 -2.254 0.0255
## namewt_max15_tror:current_plot_typeopen_with_log 169.134810 -0.891 0.3739
## namewt_max15_trcy:current_plot_typeopen_with_pvc 168.137750 -2.231 0.0270
## namewt_max15_tror:current_plot_typeopen_with_pvc 168.495505 -1.338 0.1827
##
## (Intercept)                      ***
## namewt_max15_trcy
## namewt_max15_tror
## current_plot_typeinsitu_log
## current_plot_typeinsitu_pvc
## current_plot_typeopen
## current_plot_typeopen_with_log
## current_plot_typeopen_with_pvc
## namewt_max15_trcy:current_plot_typeinsitu_log
## namewt_max15_tror:current_plot_typeinsitu_log
## namewt_max15_trcy:current_plot_typeinsitu_pvc
## namewt_max15_tror:current_plot_typeinsitu_pvc
## namewt_max15_trcy:current_plot_typeopen
## namewt_max15_tror:current_plot_typeopen
## namewt_max15_trcy:current_plot_typeopen_with_log *
## namewt_max15_tror:current_plot_typeopen_with_log
## namewt_max15_trcy:current_plot_typeopen_with_pvc *
## namewt_max15_tror:current_plot_typeopen_with_pvc
## ---

```

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

```
emmip(totwtmod,~current_plot_type|name, CI=T)
```



```
est<-emmeans(totwtmod,~current_plot_type|name, type='response')  
pairs(est)
```

```
## name = wt_max15_goro:  
## contrast  
## gap - insitu_log  
## gap - insitu_pvc  
## gap - open  
## gap - open_with_log  
## gap - open_with_pvc  
## insitu_log - insitu_pvc  
## insitu_log - open  
## insitu_log - open_with_log  
## insitu_log - open_with_pvc  
## insitu_pvc - open  
## insitu_pvc - open_with_log  
## insitu_pvc - open_with_pvc  
## open - open_with_log  
## open - open_with_pvc  
## open_with_log - open_with_pvc  
##
```

	estimate	SE	df	t.ratio	p.value
gap - insitu_log	-0.51666	0.602	188	-0.859	0.9557
gap - insitu_pvc	-0.32459	0.525	187	-0.618	0.9896
gap - open	-0.32986	0.450	187	-0.733	0.9776
gap - open_with_log	-1.11673	0.545	187	-2.050	0.3183
gap - open_with_pvc	-0.66839	0.474	187	-1.411	0.7205
insitu_log - insitu_pvc	0.19208	0.650	188	0.295	0.9997
insitu_log - open	0.18681	0.589	187	0.317	0.9996
insitu_log - open_with_log	-0.60007	0.663	186	-0.905	0.9447
insitu_log - open_with_pvc	-0.15173	0.608	187	-0.250	0.9999
insitu_pvc - open	-0.00527	0.511	186	-0.010	1.0000
insitu_pvc - open_with_log	-0.79215	0.598	188	-1.324	0.7715
insitu_pvc - open_with_pvc	-0.34380	0.535	189	-0.642	0.9876
open - open_with_log	-0.78688	0.532	186	-1.480	0.6773
open - open_with_pvc	-0.33853	0.462	189	-0.733	0.9776
open_with_log - open_with_pvc	0.44834	0.553	187	0.811	0.9653

```

## name = wt_max15_trcy:
## contrast estimate SE df t.ratio p.value
## gap - insitu_log -0.01156 0.546 187 -0.021 1.0000
## gap - insitu_pvc 0.02144 0.497 188 0.043 1.0000
## gap - open 0.40042 0.512 189 0.782 0.9703
## gap - open_with_log 0.56292 0.568 186 0.990 0.9206
## gap - open_with_pvc 0.79881 0.509 186 1.570 0.6193
## insitu_log - insitu_pvc 0.03300 0.573 188 0.058 1.0000
## insitu_log - open 0.41198 0.586 189 0.703 0.9813
## insitu_log - open_with_log 0.57448 0.639 188 0.899 0.9463
## insitu_log - open_with_pvc 0.81037 0.582 186 1.392 0.7315
## insitu_pvc - open 0.37899 0.540 189 0.701 0.9816
## insitu_pvc - open_with_log 0.54148 0.595 187 0.911 0.9434
## insitu_pvc - open_with_pvc 0.77738 0.536 186 1.449 0.6968
## open - open_with_log 0.16250 0.611 190 0.266 0.9998
## open - open_with_pvc 0.39839 0.552 188 0.722 0.9790
## open_with_log - open_with_pvc 0.23589 0.606 187 0.389 0.9988
##
## name = wt_max15_tror:
## contrast estimate SE df t.ratio p.value
## gap - insitu_log 0.32428 0.656 188 0.494 0.9963
## gap - insitu_pvc -0.44979 0.526 187 -0.855 0.9565
## gap - open 0.76322 0.656 187 1.164 0.8533
## gap - open_with_log -0.42314 0.615 187 -0.688 0.9831
## gap - open_with_pvc 0.24299 0.543 188 0.447 0.9977
## insitu_log - insitu_pvc -0.77407 0.676 189 -1.144 0.8621
## insitu_log - open 0.43893 0.782 189 0.561 0.9934
## insitu_log - open_with_log -0.74742 0.748 188 -1.000 0.9175
## insitu_log - open_with_pvc -0.08129 0.693 191 -0.117 1.0000
## insitu_pvc - open 1.21301 0.676 188 1.794 0.4723
## insitu_pvc - open_with_log 0.02665 0.636 188 0.042 1.0000
## insitu_pvc - open_with_pvc 0.69278 0.570 190 1.216 0.8286
## open - open_with_log -1.18635 0.749 189 -1.583 0.6107
## open - open_with_pvc -0.52023 0.689 189 -0.755 0.9745
## open_with_log - open_with_pvc 0.66613 0.650 188 1.024 0.9093
##
## Degrees-of-freedom method: kenward-roger
## P value adjustment: tukey method for comparing a family of 6 estimates

```

```

##### treatment response: do analysis for per capita biomass by species #####
dat4<-(dat1[c(1:4,9,13,17,22)])
pcwtdat<-as.data.frame(dat4 %>% pivot_longer(c(wt_percapita_goro, wt_percapita_tror, wt_percapita_trcy))
pcwtdat$log_wt<-log(pcwtdat$value) # log transform the weight data to get it normal looking

# model
pcwtmod<-lmer(log_wt~name*current_plot_type+(1|block), data=pcwtdat, REML=FALSE)

# test for fit, looks pretty good
# sim<-simulateResiduals(pcwtmod)
# plot(sim)

# model summary
summary(pcwtmod)

```

```

## Linear mixed model fit by maximum likelihood . t-tests use Satterthwaite's
##   method [lmerModLmerTest]
## Formula: log_wt ~ name * current_plot_type + (1 | block)
##   Data: pcwtdat
##
##          AIC      BIC  logLik deviance df.resid
##      527.8    590.8   -243.9     487.8      153
##
## Scaled residuals:
##      Min      1Q  Median      3Q     Max
## -2.30924 -0.73901 -0.03115  0.63661  2.86707
##
## Random effects:
##   Groups   Name        Variance Std.Dev.
##   block     (Intercept) 0.07573  0.2752
##   Residual           0.93957  0.9693
## Number of obs: 173, groups: block, 7
##
## Fixed effects:
##                                         Estimate Std. Error
## (Intercept)                         -2.8584    0.2797
## namewt_percapita_trcy                -0.2256    0.3671
## namewt_percapita_tror                 -0.5377    0.3816
## current_plot_typeinsitu_log            1.1314    0.4756
## current_plot_typeinsitu_pvc             0.3094    0.4153
## current_plot_typeopen                  0.4074    0.3558
## current_plot_typeopen_with_log          1.2638    0.4309
## current_plot_typeopen_with_pvc          0.6385    0.3747
## namewt_percapita_trcy:current_plot_typeinsitu_log   -1.2080    0.6405
## namewt_percapita_tror:current_plot_typeinsitu_log   -1.3322    0.7021
## namewt_percapita_trcy:current_plot_typeinsitu_pvc   -0.4013    0.5711
## namewt_percapita_tror:current_plot_typeinsitu_pvc   -0.2449    0.5876
## namewt_percapita_trcy:current_plot_typeopen            -0.6393    0.5402
## namewt_percapita_tror:current_plot_typeopen            -1.5347    0.6280
## namewt_percapita_trcy:current_plot_typeopen_with_log   -1.7482    0.6228
## namewt_percapita_tror:current_plot_typeopen_with_log   -0.9779    0.6510
## namewt_percapita_trcy:current_plot_typeopen_with_pvc   -1.1678    0.5497
## namewt_percapita_tror:current_plot_typeopen_with_pvc   -0.9233    0.5696
##
##                                         df t value Pr(>|t|)
## (Intercept)                         101.5741 -10.220 < 2e-16
## namewt_percapita_trcy                167.1779 -0.615  0.53968
## namewt_percapita_tror                166.7751 -1.409  0.16066
## current_plot_typeinsitu_log          168.2573  2.379  0.01847
## current_plot_typeinsitu_pvc          167.4481  0.745  0.45733
## current_plot_typeopen                 167.4462  1.145  0.25377
## current_plot_typeopen_with_log       167.5329  2.933  0.00382
## current_plot_typeopen_with_pvc       167.6177  1.704  0.09017
## namewt_percapita_trcy:current_plot_typeinsitu_log   167.2535 -1.886  0.06102
## namewt_percapita_tror:current_plot_typeinsitu_log   167.5595 -1.897  0.05949
## namewt_percapita_trcy:current_plot_typeinsitu_pvc   167.5144 -0.703  0.48322
## namewt_percapita_tror:current_plot_typeinsitu_pvc   167.2422 -0.417  0.67741
## namewt_percapita_trcy:current_plot_typeopen            168.9857 -1.183  0.23831
## namewt_percapita_tror:current_plot_typeopen            167.4504 -2.444  0.01557
## namewt_percapita_trcy:current_plot_typeopen_with_log 167.4266 -2.807  0.00559

```

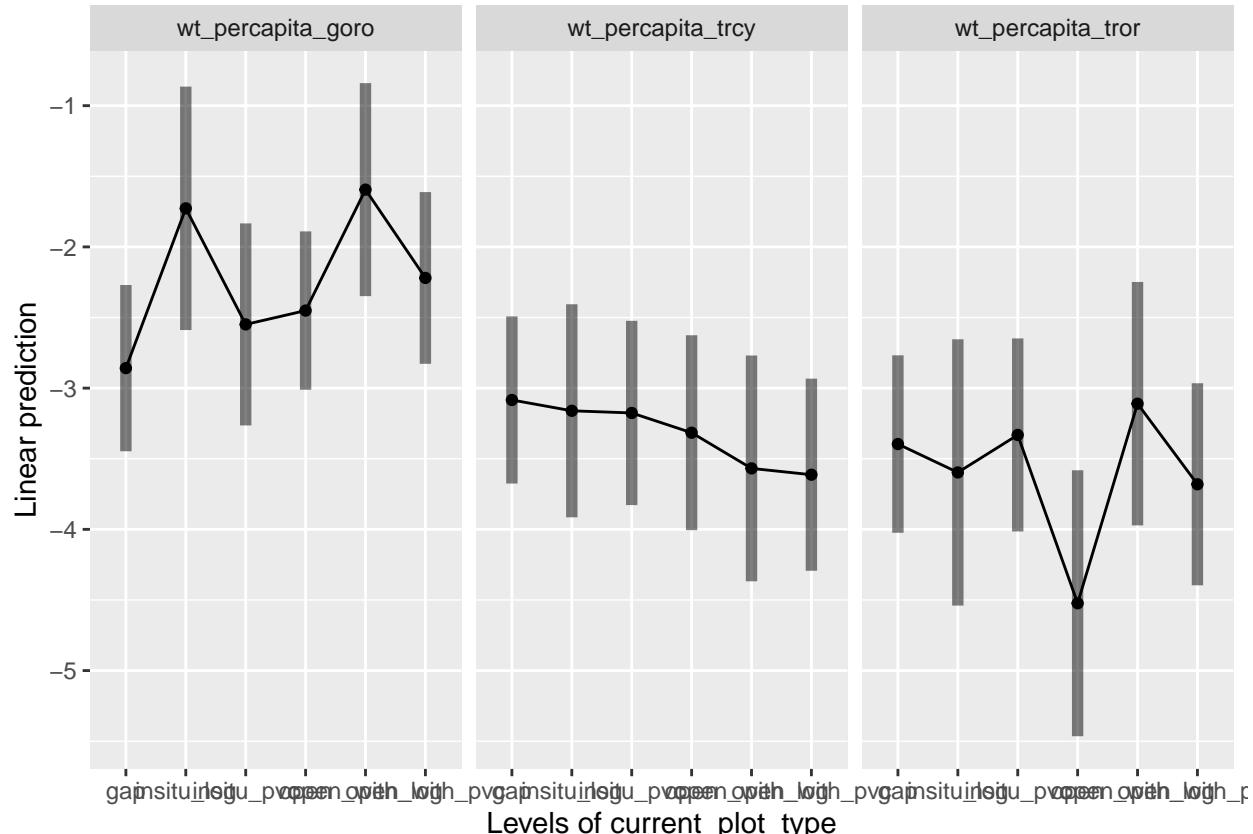
```

## namewt_percapita_tror:current_plot_typeopen_with_log 168.1810 -1.502 0.13493
## namewt_percapita_trcy:current_plot_typeopen_with_pvc 167.4163 -2.124 0.03513
## namewt_percapita_tror:current_plot_typeopen_with_pvc 167.6639 -1.621 0.10694
##
## (Intercept) ***

## namewt_percapita_trcy
## namewt_percapita_tror
## current_plot_typeinsitu_log *
## current_plot_typeinsitu_pvc
## current_plot_typeopen
## current_plot_typeopen_with_log **
## current_plot_typeopen_with_pvc .
## namewt_percapita_trcy:current_plot_typeinsitu_log .
## namewt_percapita_tror:current_plot_typeinsitu_log .
## namewt_percapita_trcy:current_plot_typeinsitu_pvc
## namewt_percapita_tror:current_plot_typeinsitu_pvc
## namewt_percapita_trcy:current_plot_typeopen
## namewt_percapita_tror:current_plot_typeopen *
## namewt_percapita_trcy:current_plot_typeopen_with_log **
## namewt_percapita_tror:current_plot_typeopen_with_log
## namewt_percapita_trcy:current_plot_typeopen_with_pvc *
## namewt_percapita_tror:current_plot_typeopen_with_pvc
## ---
## Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

```

```
emmap(pwctmod, ~current_plot_type | name, CI=T)
```



```
est<-emmeans(totwtmod, ~current_plot_type|name, type='response')
pairs(est)
```

```
## name = wt_max15_goro:
## contrast estimate SE df t.ratio p.value
## gap - insitu_log -0.51666 0.602 188 -0.859 0.9557
## gap - insitu_pvc -0.32459 0.525 187 -0.618 0.9896
## gap - open -0.32986 0.450 187 -0.733 0.9776
## gap - open_with_log -1.11673 0.545 187 -2.050 0.3183
## gap - open_with_pvc -0.66839 0.474 187 -1.411 0.7205
## insitu_log - insitu_pvc 0.19208 0.650 188 0.295 0.9997
## insitu_log - open 0.18681 0.589 187 0.317 0.9996
## insitu_log - open_with_log -0.60007 0.663 186 -0.905 0.9447
## insitu_log - open_with_pvc -0.15173 0.608 187 -0.250 0.9999
## insitu_pvc - open -0.00527 0.511 186 -0.010 1.0000
## insitu_pvc - open_with_log -0.79215 0.598 188 -1.324 0.7715
## insitu_pvc - open_with_pvc -0.34380 0.535 189 -0.642 0.9876
## open - open_with_log -0.78688 0.532 186 -1.480 0.6773
## open - open_with_pvc -0.33853 0.462 189 -0.733 0.9776
## open_with_log - open_with_pvc 0.44834 0.553 187 0.811 0.9653
##
## name = wt_max15_trcy:
## contrast estimate SE df t.ratio p.value
## gap - insitu_log -0.01156 0.546 187 -0.021 1.0000
## gap - insitu_pvc 0.02144 0.497 188 0.043 1.0000
## gap - open 0.40042 0.512 189 0.782 0.9703
## gap - open_with_log 0.56292 0.568 186 0.990 0.9206
## gap - open_with_pvc 0.79881 0.509 186 1.570 0.6193
## insitu_log - insitu_pvc 0.03300 0.573 188 0.058 1.0000
## insitu_log - open 0.41198 0.586 189 0.703 0.9813
## insitu_log - open_with_log 0.57448 0.639 188 0.899 0.9463
## insitu_log - open_with_pvc 0.81037 0.582 186 1.392 0.7315
## insitu_pvc - open 0.37899 0.540 189 0.701 0.9816
## insitu_pvc - open_with_log 0.54148 0.595 187 0.911 0.9434
## insitu_pvc - open_with_pvc 0.77738 0.536 186 1.449 0.6968
## open - open_with_log 0.16250 0.611 190 0.266 0.9998
## open - open_with_pvc 0.39839 0.552 188 0.722 0.9790
## open_with_log - open_with_pvc 0.23589 0.606 187 0.389 0.9988
##
## name = wt_max15_tror:
## contrast estimate SE df t.ratio p.value
## gap - insitu_log 0.32428 0.656 188 0.494 0.9963
## gap - insitu_pvc -0.44979 0.526 187 -0.855 0.9565
## gap - open 0.76322 0.656 187 1.164 0.8533
## gap - open_with_log -0.42314 0.615 187 -0.688 0.9831
## gap - open_with_pvc 0.24299 0.543 188 0.447 0.9977
## insitu_log - insitu_pvc -0.77407 0.676 189 -1.144 0.8621
## insitu_log - open 0.43893 0.782 189 0.561 0.9934
## insitu_log - open_with_log -0.74742 0.748 188 -1.000 0.9175
## insitu_log - open_with_pvc -0.08129 0.693 191 -0.117 1.0000
## insitu_pvc - open 1.21301 0.676 188 1.794 0.4723
## insitu_pvc - open_with_log 0.02665 0.636 188 0.042 1.0000
## insitu_pvc - open_with_pvc 0.69278 0.570 190 1.216 0.8286
```

```

##  open - open_with_log      -1.18635 0.749 189 -1.583 0.6107
##  open - open_with_pvc     -0.52023 0.689 189 -0.755 0.9745
##  open_with_log - open_with_pvc  0.66613 0.650 188  1.024 0.9093
##
## Degrees-of-freedom method: kenward-roger
## P value adjustment: tukey method for comparing a family of 6 estimates

```

```
##### What about the log legacy - initial treatment for 2021 #####

```

```
##### log legacy response: do analysis for count by species #####
fit3_leg<-glmmTMB(value~name*initial+(1 | block), ziformula=~., family=nbinom2(), data=countdat, REML=F)

```

```
## This stuff will just give you the end result counts with the zeros factored in...
summary(fit3_leg)
```

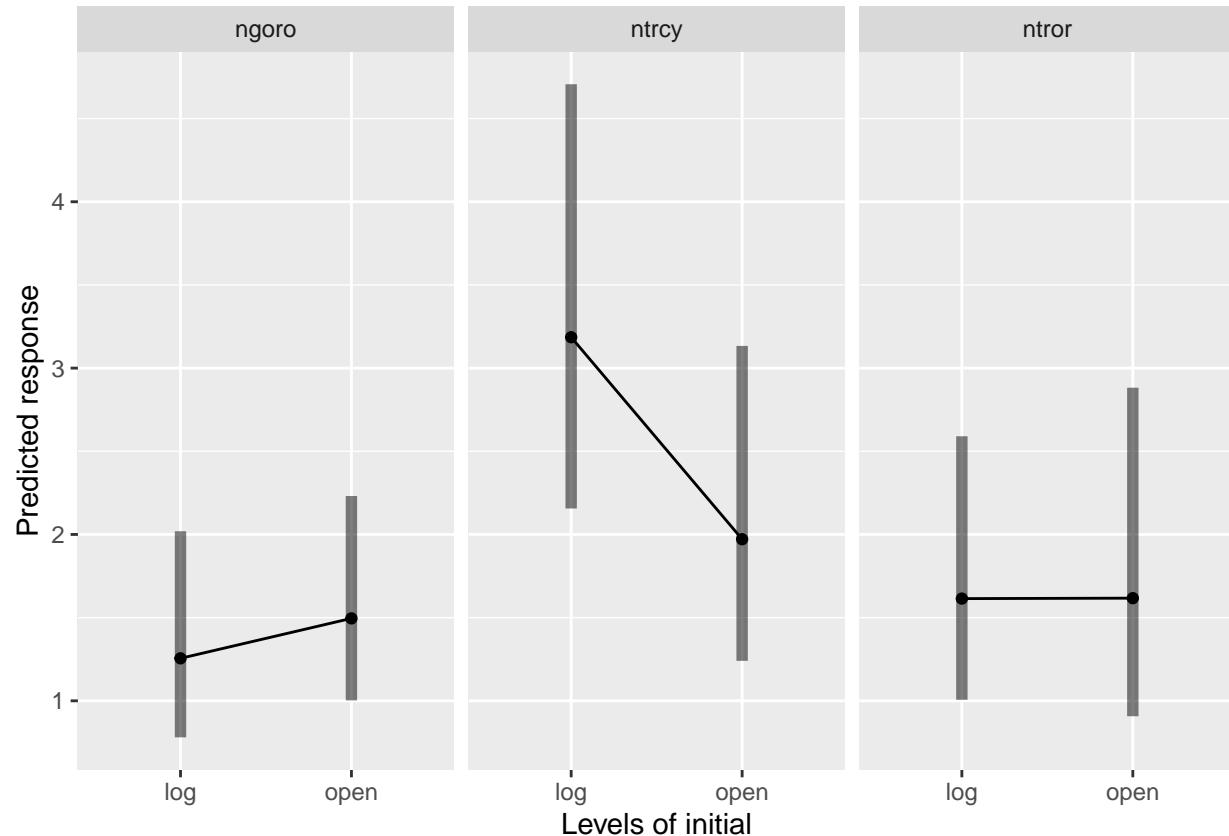
```

## Family: nbinom2  ( log )
## Formula:          value ~ name * initial + (1 | block)
## Zero inflation:   ~.
## Data: countdat
##
##      AIC      BIC  logLik deviance df.resid
##  1085.9  1143.2   -528.0    1055.9      321
##
## Random effects:
## 
## Conditional model:
## Groups Name        Variance Std.Dev.
## block  (Intercept) 0.008194 0.09052
## Number of obs: 336, groups: block, 7
##
## Zero-inflation model:
## Groups Name        Variance Std.Dev.
## block  (Intercept) 0.3856   0.621
## Number of obs: 336, groups: block, 7
##
## Dispersion parameter for nbinom2 family (): 1.34
##
## Conditional model:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  0.2274    0.2425  0.938  0.34834
## namentrcy   0.9312    0.2900  3.211  0.00132 **
## namentror   0.2516    0.3176  0.792  0.42839
## initialopen  0.1751    0.2955  0.592  0.55357
## namentrcy:initialopen -0.6549   0.4113 -1.592  0.11131
## namentror:initialopen -0.1735   0.4673 -0.371  0.71045
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ',' 1
##
## Zero-inflation model:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -2.125     1.490 -1.427  0.154
## namentrcy   1.030     1.413  0.729  0.466
## namentror   1.078     1.456  0.741  0.459
## initialopen -2.851    14.373 -0.198  0.843

```

```
## namentrcy:initialopen    3.113     14.414    0.216    0.829
## namentror:initialopen   3.645     14.425    0.253    0.801
```

```
emmip(fit3_leg, ~initial|name, type='response', CI=T)
```



```
est<-emmeans(fit3_leg, ~initial|name, type='response')
pairs(est)
```

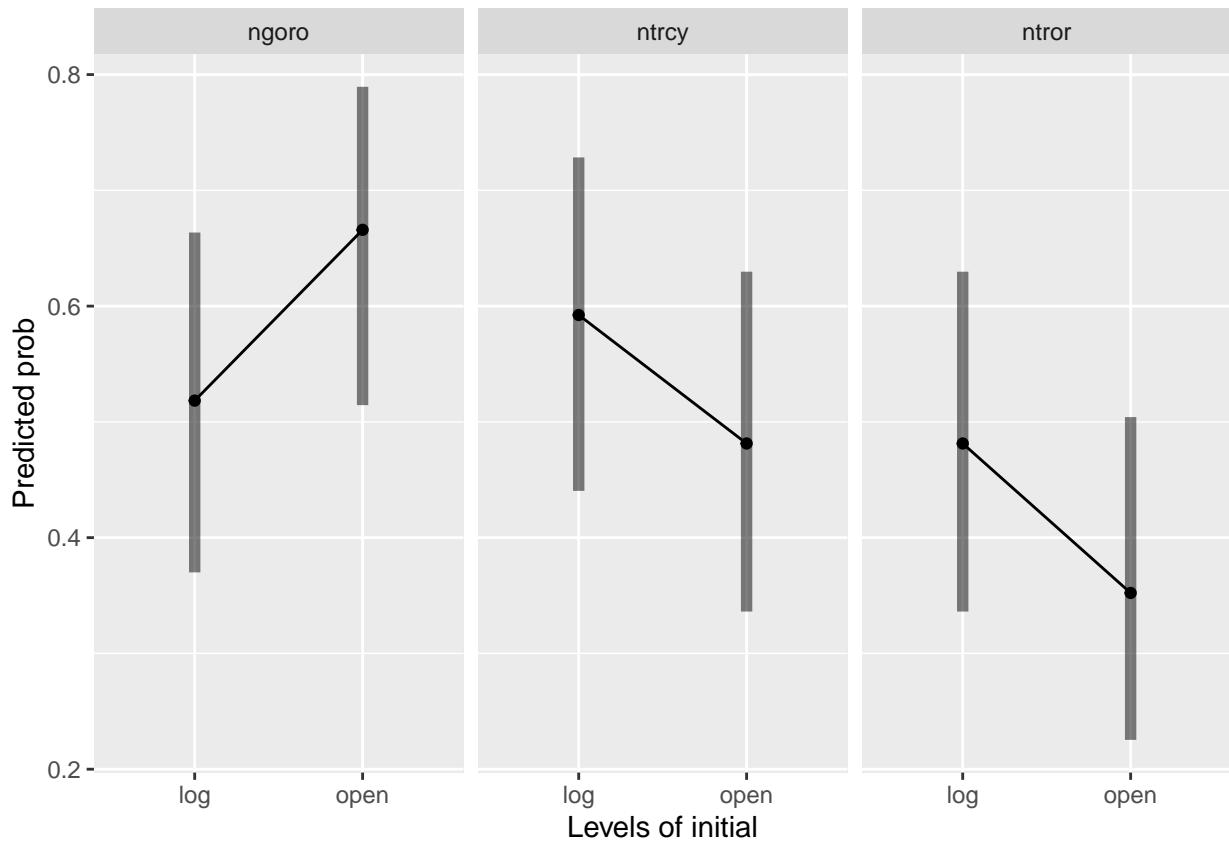
```
## name = ngoro:
## contrast ratio    SE df null z.ratio p.value
## log / open 0.839 0.248 Inf    1  -0.592  0.5536
##
## name = ntrcy:
## contrast ratio    SE df null z.ratio p.value
## log / open 1.616 0.459 Inf    1   1.689  0.0912
##
## name = ntror:
## contrast ratio    SE df null z.ratio p.value
## log / open 0.998 0.356 Inf    1  -0.004  0.9964
##
## Tests are performed on the log scale
```

```
#### split up occurrence and abundance
### zeros and ones
```

```

zeroftit_leg<-glmmTMB(presence~name*initial+(1 | block), family=binomial, data=countdat, REML=FALSE)
emmip(zeroftit_leg, ~initial|name, type='response', CI=T)

```



```

est<-emmeans(zeroftit_leg, ~initial|name, type='response')
pairs(est)

```

```

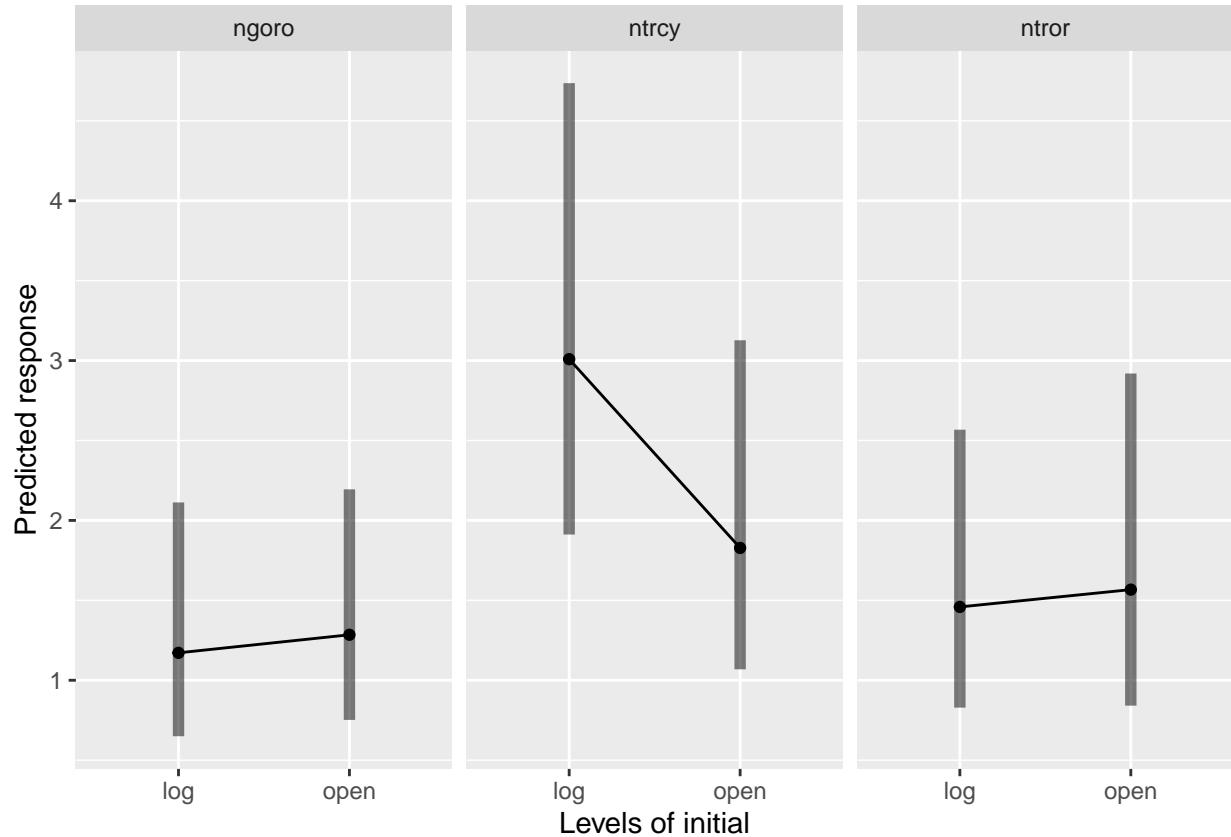
## name = ngoro:
##   contrast   odds.ratio     SE  df null z.ratio p.value
##   log / open      0.54 0.214 Inf    1 -1.557  0.1195
##
## name = ntrcy:
##   contrast   odds.ratio     SE  df null z.ratio p.value
##   log / open      1.57 0.608 Inf    1  1.155  0.2482
##
## name = ntror:
##   contrast   odds.ratio     SE  df null z.ratio p.value
##   log / open      1.71 0.671 Inf    1  1.359  0.1740
##
## Tests are performed on the log odds ratio scale

```

```

### abundance with a truncated negbinom
countfit_leg<-glmmTMB(posicounts~name*initial+(1 | block), family=truncated_nbinom2(), data=countdat, REML=FALSE)
emmip(countfit_leg, ~initial|name, type='response', CI=T)

```



```
est<-emmeans(countfit_leg, ~initial|name, type='response')
pairs(est)
```

```
## name = ngoro:
## contrast ratio SE df null z.ratio p.value
## log / open 0.912 0.304 Inf -0.276 0.7826
##
## name = ntrcy:
## contrast ratio SE df null z.ratio p.value
## log / open 1.646 0.508 Inf 1.615 0.1062
##
## name = ntror:
## contrast ratio SE df null z.ratio p.value
## log / open 0.931 0.350 Inf -0.190 0.8490
##
## Tests are performed on the log scale
```

```
##### log legacy response: do analysis for total weight by species #####
# weights
totwtmod_leg<-lmer(log_wt~name*initial+(1|block), data=totwtdat, REML=FALSE)

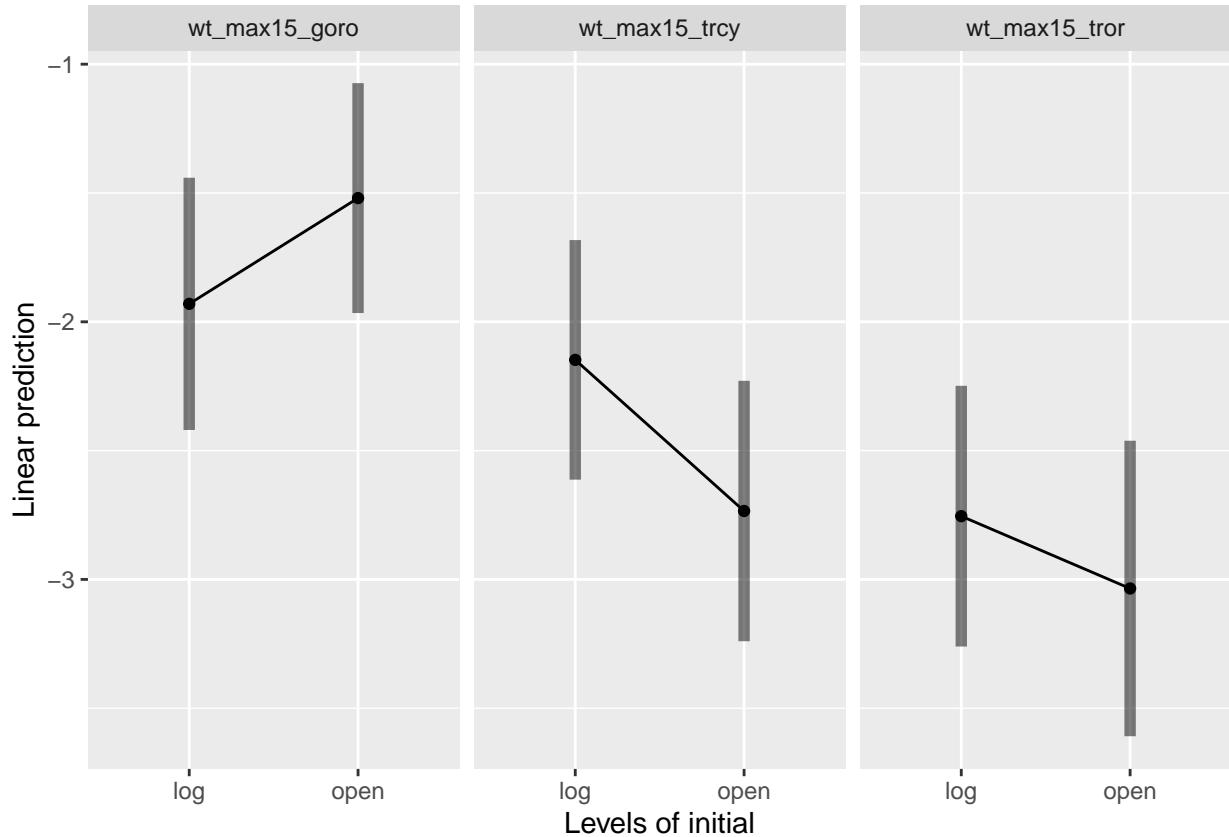
# model summary
summary(totwtmod_leg)
```

```

## Linear mixed model fit by maximum likelihood . t-tests use Satterthwaite's
##   method [lmerModLmerTest]
## Formula: log_wt ~ name * initial + (1 | block)
##   Data: totwtdat
##
##       AIC      BIC  logLik deviance df.resid
##     570.9    596.1   -277.4    554.9      165
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -2.4315 -0.7041  0.1434  0.6437  2.4237
##
## Random effects:
##   Groups   Name        Variance Std.Dev.
##   block    (Intercept) 0.05212  0.2283
##   Residual           1.40964  1.1873
## Number of obs: 173, groups: block, 7
##
## Fixed effects:
##                               Estimate Std. Error      df t value Pr(>|t|)    
## (Intercept)                 -1.9303   0.2375 73.5394 -8.129 7.7e-12 ***
## namewt_max15_trcy          -0.2174   0.3028 168.0694 -0.718  0.4737    
## namewt_max15_tror          -0.8242   0.3184 168.6561 -2.588  0.0105 *  
## initialopen                  0.4106   0.2948 167.5987  1.393  0.1655    
## namewt_max15_trcy:initialopen -0.9973   0.4269 167.9272 -2.336  0.0207 *  
## namewt_max15_tror:initialopen -0.6913   0.4589 168.6129 -1.506  0.1338    
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##                               (Intr) nmwt_mx15_trc nmwt_mx15_trr intlpn nmwt_mx15_trc:
## nmwt_mx15_trc            -0.679
## nmwt_mx15_trr             -0.645  0.508
## initialopen                -0.696  0.546      0.518
## nmwt_mx15_trc:            0.479 -0.707      -0.357      -0.691
## nmwt_mx15_trr:            0.446 -0.352      -0.693      -0.641  0.443

emmap(totwtdat, ~initial|name, CI=T)

```



```
est<-emmeans(totwtmod_leg, ~initial|name, type='response')
pairs(est)
```

```
## name = wt_max15_goro:
## contrast estimate SE df t.ratio p.value
## log - open -0.411 0.300 172 -1.370 0.1723
##
## name = wt_max15_trcy:
## contrast estimate SE df t.ratio p.value
## log - open 0.587 0.314 172 1.870 0.0632
##
## name = wt_max15_tror:
## contrast estimate SE df t.ratio p.value
## log - open 0.281 0.359 175 0.782 0.4355
##
## Degrees-of-freedom method: kenward-roger
```

```
##### log legacy response: do analysis for per capita weight by species #####
# model
pcwtmod_leg<-lmer(log_wt~name*initial+(1|block), data=pcwtdat, REML=FALSE)

# test for fit, looks pretty good
# sim<-simulateResiduals(pcwtmod_leg)
# plot(sim)
```

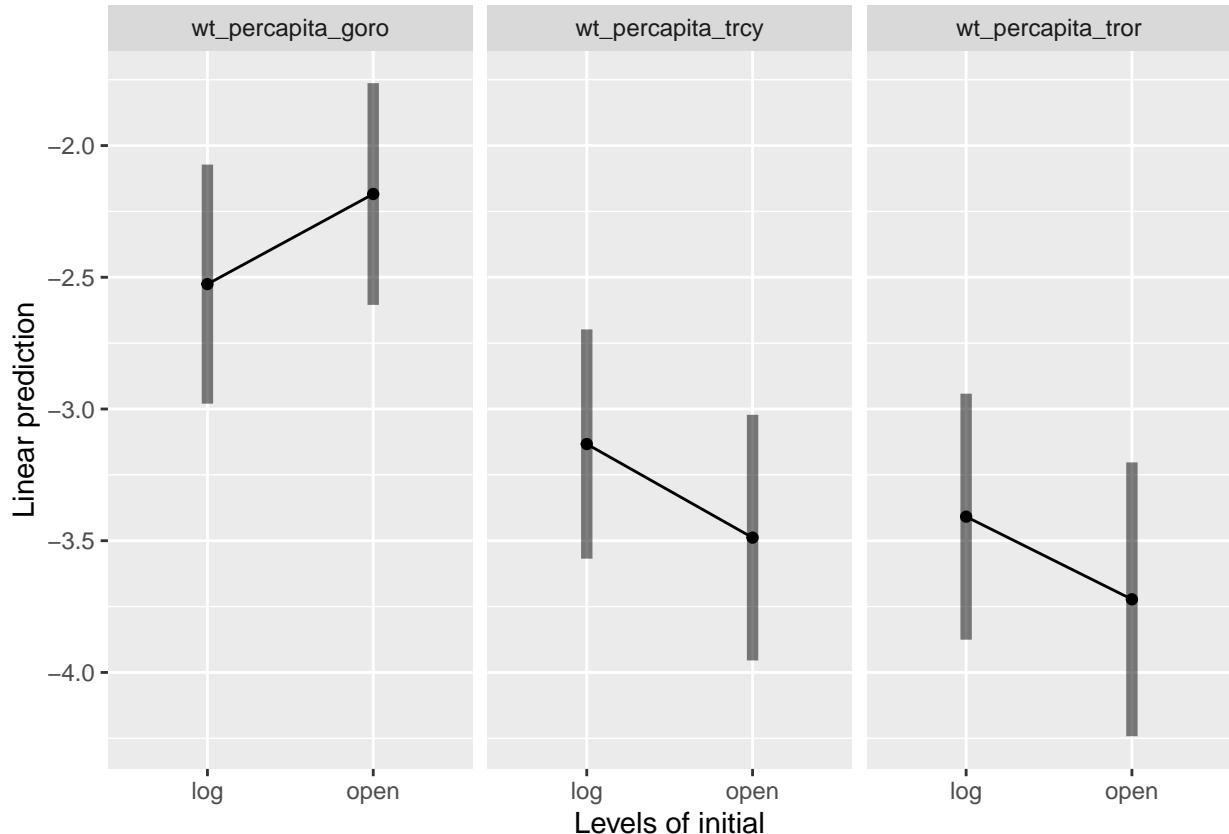
```

# model summary
summary(pcwtmod_leg)

## Linear mixed model fit by maximum likelihood . t-tests use Satterthwaite's
## method [lmerModLmerTest]
## Formula: log_wt ~ name * initial + (1 | block)
## Data: pcwtdat
##
##      AIC      BIC  logLik deviance df.resid
##      519.5    544.7   -251.7     503.5      165
## 
## Scaled residuals:
##      Min      1Q  Median      3Q     Max
## -2.03880 -0.72942 -0.05825  0.70998  2.68818
## 
## Random effects:
##   Groups   Name        Variance Std.Dev.
##   block    (Intercept) 0.07959  0.2821
##   Residual           1.03008  1.0149
## Number of obs: 173, groups: block, 7
## 
## Fixed effects:
##                               Estimate Std. Error      df t value Pr(>|t|)    
## (Intercept)                 -2.5260   0.2173 48.0659 -11.624 1.44e-15  
## namewt_percapita_trcy      -0.6070   0.2590 167.4836 -2.344  0.02026  
## namewt_percapita_tror      -0.8828   0.2725 167.9442 -3.240  0.00144  
## initialopen                  0.3420   0.2521 167.1927  1.357  0.17664  
## namewt_percapita_trcy:initialopen -0.6973   0.3651 167.4033 -1.910  0.05787  
## namewt_percapita_tror:initialopen -0.6557   0.3927 167.8602 -1.670  0.09683  
## 
## (Intercept)                 ***  
## namewt_percapita_trcy       *   
## namewt_percapita_tror       **  
## initialopen                  .  
## namewt_percapita_trcy:initialopen .
## namewt_percapita_tror:initialopen .
## --- 
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Correlation of Fixed Effects:
##          (Intr) nmwt_prcpt_trc nmwt_prcpt_trr intlpn nmwt_prcpt_trc:
## nmwt_prcpt_trc  -0.635
## nmwt_prcpt_trr  -0.602  0.509
## initialopen      -0.650  0.546      0.517
## nmwt_prcpt_trc:  0.447 -0.706      -0.356      -0.691
## nmwt_prcpt_trr:  0.416 -0.352      -0.693      -0.640  0.443

emmap(pcwtmod_leg,-initial|name,CI=T)

```



```
est<-emmeans(pcwtmod_leg, ~initial|name, type='response')
pairs(est)
```

```
## name = wt_percapita_goro:
## contrast estimate SE df t.ratio p.value
## log - open -0.342 0.256 172 -1.335 0.1835
##
## name = wt_percapita_trcy:
## contrast estimate SE df t.ratio p.value
## log - open 0.355 0.268 172 1.324 0.1871
##
## name = wt_percapita_tror:
## contrast estimate SE df t.ratio p.value
## log - open 0.314 0.307 174 1.021 0.3086
##
## Degrees-of-freedom method: kenward-roger
```

```
##### What about the physical barrier treatment for 2021 #####

```

```
##### physical barrier response: do analysis for count by species #####
fit3_phys<-glmmTMB(value~name*physical_barrier+(1 | block), ziformula=~., family=nbinom2(), data=countda
```

This stuff will just give you the end result counts with the zeros factored in...

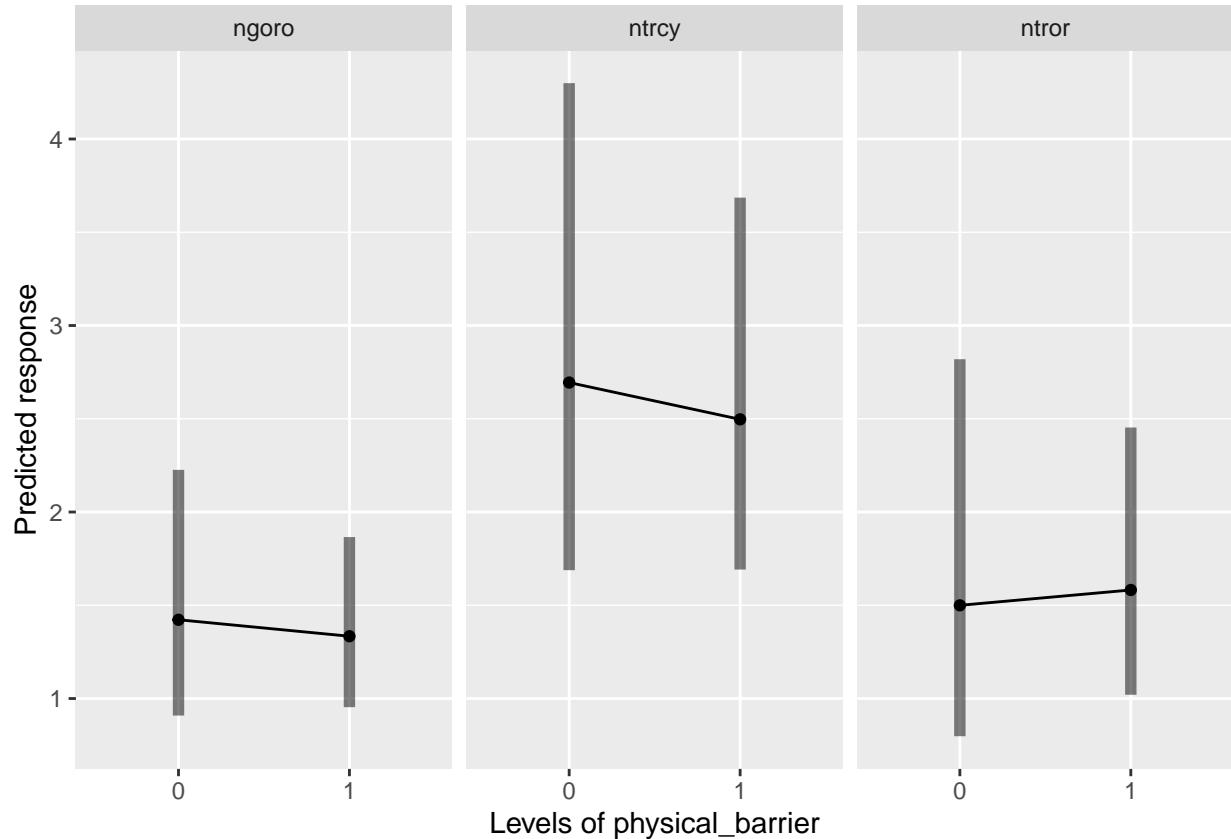
```
summary(fit3_phys)
```

```

## Family: nbinom2 ( log )
## Formula:           value ~ name * physical_barrier + (1 | block)
## Zero inflation:    ~.
## Data: countdat
##
##      AIC      BIC  logLik deviance df.resid
##  1081.4   1138.6   -525.7    1051.4      321
##
## Random effects:
##
## Conditional model:
## Groups Name        Variance Std.Dev.
## block  (Intercept) 0.01393  0.118
## Number of obs: 336, groups: block, 7
##
## Zero-inflation model:
## Groups Name        Variance Std.Dev.
## block  (Intercept) 0.6455   0.8034
## Number of obs: 336, groups: block, 7
##
## Dispersion parameter for nbinom2 family (): 1.25
##
## Conditional model:
##                               Estimate Std. Error z value Pr(>|z|)
## (Intercept)                  0.35230   0.22855  1.542   0.1232
## namentrcy                   0.63881   0.31276  2.042   0.0411 *
## namentrор                   0.05297   0.36846  0.144   0.8857
## physical_barrier1          -0.06418   0.27850 -0.230   0.8177
## namentrcy:physical_barrier1 -0.01190   0.40264 -0.030   0.9764
## namentrор:physical_barrier1  0.11757   0.45074  0.261   0.7942
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Zero-inflation model:
##                               Estimate Std. Error z value Pr(>|z|)
## (Intercept)                 -2.222     1.377  -1.613   0.1067
## namentrcy                   1.911     1.330   1.436   0.1509
## namentrор                   2.226     1.331   1.673   0.0944 .
## physical_barrier1          -15.234    4791.566 -0.003   0.9975
## namentrcy:physical_barrier1 13.581    4791.566  0.003   0.9977
## namentrор:physical_barrier1 13.477    4791.566  0.003   0.9978
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

emmap(fit3_phys, ~physical_barrier|name, type='response', CI=T)

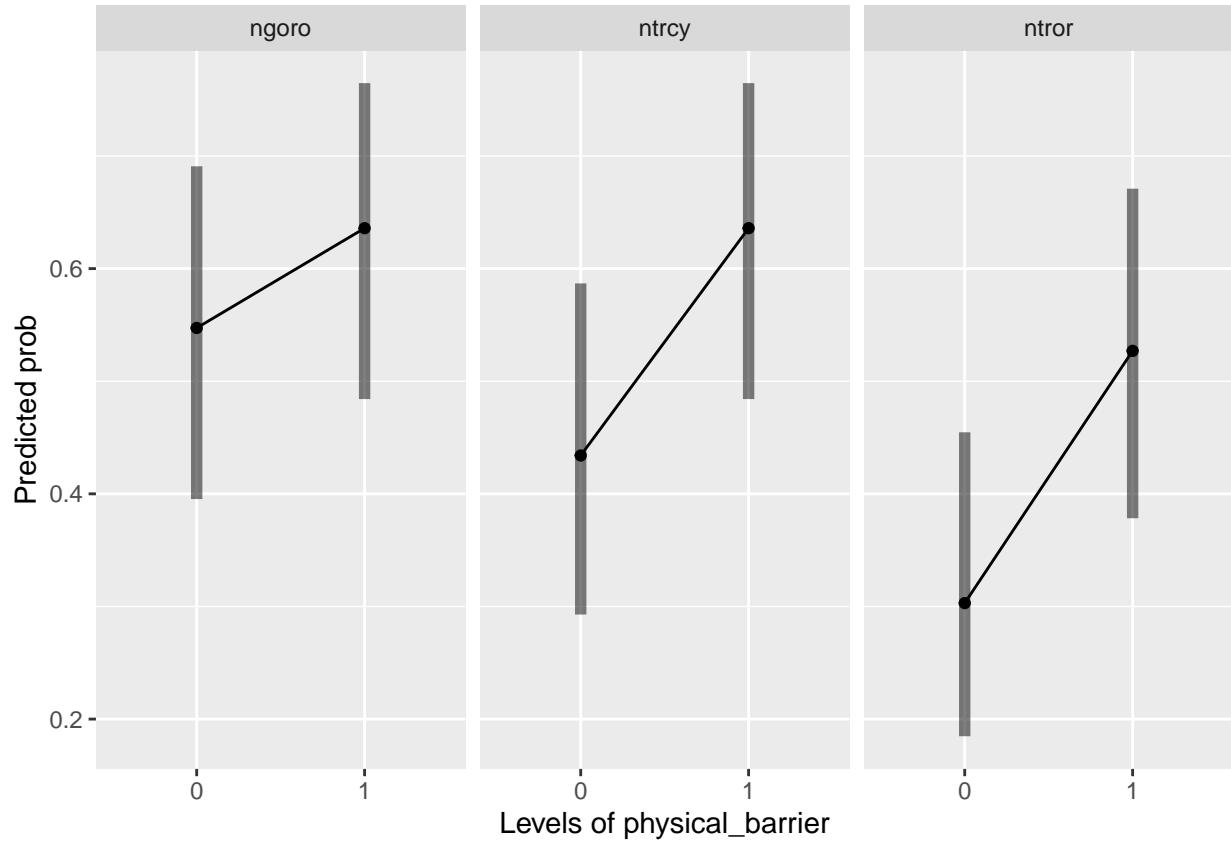
```



```
est<-emmeans(fit3_phys,~physical_barrier|name, type='response')
pairs(est)
```

```
## name = ngoro:
## contrast                                ratio     SE  df null z.ratio p.value
## physical_barrier0 / physical_barrier1 1.066 0.297 Inf   1  0.230  0.8177
##
## name = ntrcy:
## contrast                                ratio     SE  df null z.ratio p.value
## physical_barrier0 / physical_barrier1 1.079 0.311 Inf   1  0.264  0.7916
##
## name = ntror:
## contrast                                ratio     SE  df null z.ratio p.value
## physical_barrier0 / physical_barrier1 0.948 0.346 Inf   1 -0.146  0.8837
##
## Tests are performed on the log scale

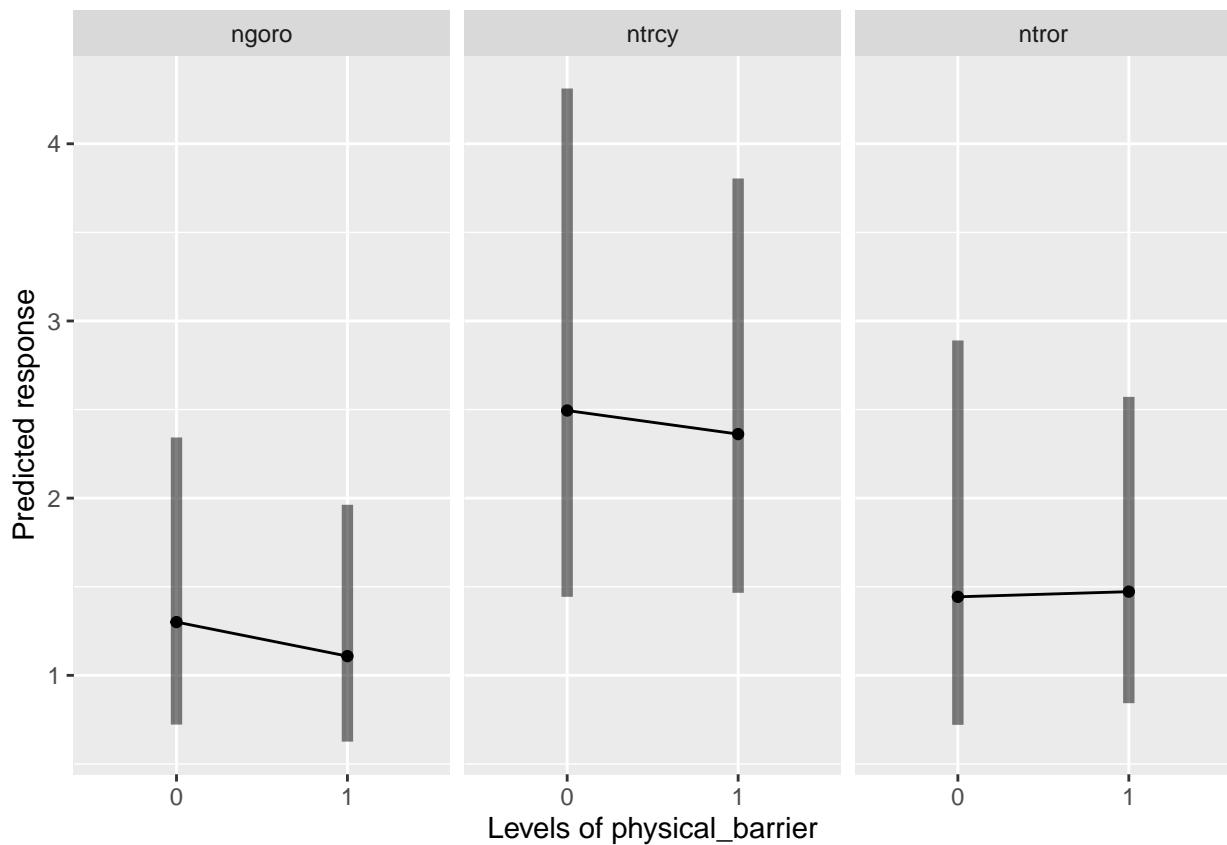
#### split up occurrence and abundance
#### zeros and ones
zerofit_phys<-glmmTMB(presence~name*physical_barrier+(1 | block), family=binomial, data=countdat, REML=F)
emmip(zerofit_phys,~physical_barrier|name, type='response',CI=T)
```



```
est<-emmeans(zeroefit_phys, ~physical_barrier|name, type='response')
pairs(est)
```

```
## name = ngoro:
##   contrast                               odds.ratio    SE  df null z.ratio
##   physical_barrier0 / physical_barrier1      0.692 0.272 Inf   1 -0.936
##   p.value
##   0.3491
##
## name = ntrcy:
##   contrast                               odds.ratio    SE  df null z.ratio
##   physical_barrier0 / physical_barrier1      0.439 0.173 Inf   1 -2.089
##   p.value
##   0.0367
##
## name = ntror:
##   contrast                               odds.ratio    SE  df null z.ratio
##   physical_barrier0 / physical_barrier1      0.390 0.157 Inf   1 -2.342
##   p.value
##   0.0192
##
## Tests are performed on the log odds ratio scale
```

```
### abundance with a truncated negbinom
countfit_phys<-glmmTMB(poiscounts~name*physical_barrier+(1 | block), family=truncated_nbinom2(), data=co
emmip(countfit_phys,~physical_barrier|name, type='response',CI=T)
```



```

est<-emmeans(countfit_phys, ~physical_barrier|name, type='response')
pairs(est)

## name = ngoro:
##   contrast                                ratio    SE  df null z.ratio p.value
##   physical_barrier0 / physical_barrier1  1.17 0.395 Inf   1  0.475  0.6349
## 
## name = ntrcy:
##   contrast                                ratio    SE  df null z.ratio p.value
##   physical_barrier0 / physical_barrier1  1.06 0.333 Inf   1  0.174  0.8618
## 
## name = ntror:
##   contrast                                ratio    SE  df null z.ratio p.value
##   physical_barrier0 / physical_barrier1  0.98 0.388 Inf   1 -0.050  0.9597
## 
## Tests are performed on the log scale

##### physical barrier response: do analysis for total weight by species #####
# weights
totwtmod_phys<-lmer(log_wt~name*physical_barrier+(1|block), data=totwtdat, REML=FALSE)

# model summary
summary(totwtmod_phys)

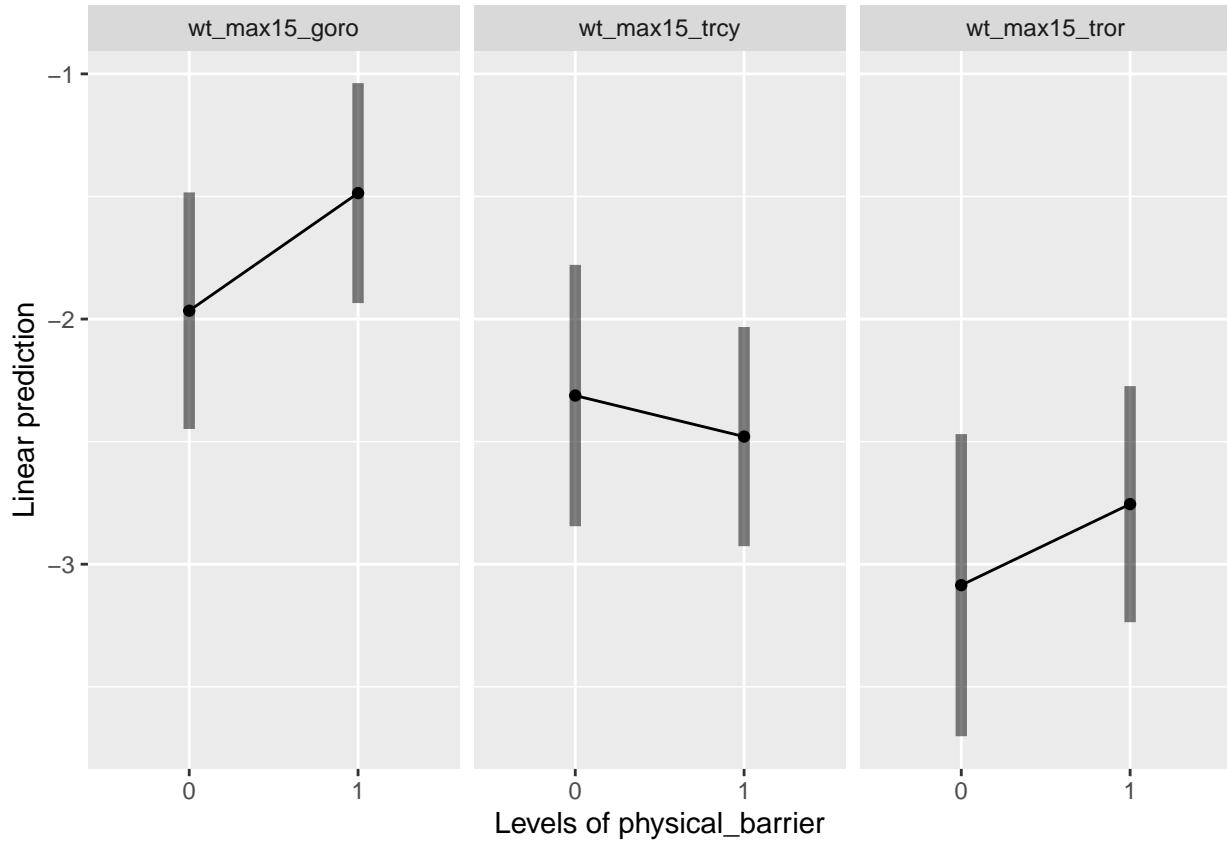
```

```

## Linear mixed model fit by maximum likelihood . t-tests use Satterthwaite's
##   method [lmerModLmerTest]
## Formula: log_wt ~ name * physical_barrier + (1 | block)
##   Data: totwtdat
##
##      AIC      BIC  logLik deviance df.resid
##      573.2    598.5   -278.6     557.2      165
##
## Scaled residuals:
##    Min     1Q Median     3Q    Max
## -2.5416 -0.7613  0.1349  0.6948  2.0654
##
## Random effects:
##   Groups   Name        Variance Std.Dev.
##   block    (Intercept) 0.04565  0.2137
##   Residual           1.43320  1.1972
## Number of obs: 173, groups: block, 7
##
## Fixed effects:
##                               Estimate Std. Error      df t value
## (Intercept)                 -1.9660   0.2339 74.7835 -8.405
## namewt_max15_trcy          -0.3461   0.3295 170.1738 -1.050
## namewt_max15_tror          -1.1195   0.3640 168.1650 -3.076
## physical_barrier1          0.4796   0.2966 168.3691  1.617
## namewt_max15_trcy:physical_barrier1 -0.6470   0.4334 168.3829 -1.493
## namewt_max15_tror:physical_barrier1 -0.1491   0.4693 167.9327 -0.318
##                               Pr(>|t|)
## (Intercept)                 2.06e-12 ***
## namewt_max15_trcy          0.29502
## namewt_max15_tror          0.00245 **
## physical_barrier1          0.10769
## namewt_max15_trcy:physical_barrier1 0.13741
## namewt_max15_tror:physical_barrier1 0.75110
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##                               (Intr) nmwt_mx15_trc nmwt_mx15_trr phys_1 nmwt_mx15_trc:_1
## nmwt_mx15_trc            -0.624
## nmwt_mx15_trr            -0.562  0.403
## physcl_brr1              -0.692  0.489      0.443
## nmwt_mx15_trc:_1         0.473 -0.758      -0.305     -0.683
## nmwt_mx15_trr:_1         0.434 -0.310      -0.774     -0.631  0.432

emmap(totwtdmod_phys, ~physical_barrier | name, CI=T)

```



```
est<-emmeans(totwtmod_phys, ~physical_barrier|name, type='response')
pairs(est)
```

```
## name = wt_max15_goro:
## contrast                                estimate    SE  df t.ratio p.value
## physical_barrier0 - physical_barrier1   -0.480 0.302 173 -1.589  0.1138
##
## name = wt_max15_trcy:
## contrast                                estimate    SE  df t.ratio p.value
## physical_barrier0 - physical_barrier1   0.167 0.323 174  0.519  0.6048
##
## name = wt_max15_tror:
## contrast                                estimate    SE  df t.ratio p.value
## physical_barrier0 - physical_barrier1   -0.331 0.371 173 -0.892  0.3736
##
## Degrees-of-freedom method: kenward-roger
```

physical barrier response: do analysis for per capita weight by species

```
# model
pcwtmod_phys<-lmer(log_wt~name*physical_barrier+(1|block), data=pcwtdat, REML=FALSE)

# test for fit, looks pretty good
# sim<-simulateResiduals(pcwtmod_leg)
# plot(sim)
```

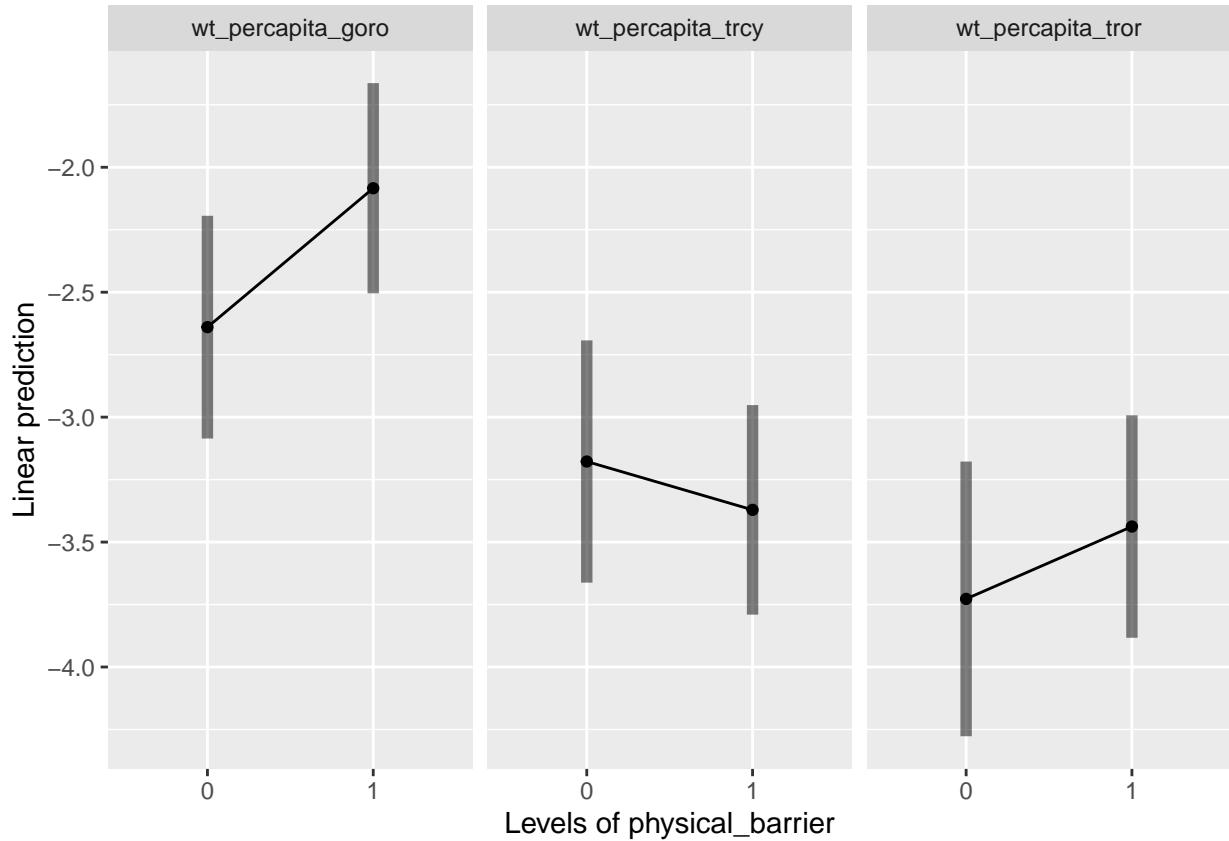
```

# model summary
summary(pcwtmod_phys)

## Linear mixed model fit by maximum likelihood . t-tests use Satterthwaite's
## method [lmerModLmerTest]
## Formula: log_wt ~ name * physical_barrier + (1 | block)
## Data: pcwtdat
##
##      AIC      BIC  logLik deviance df.resid
##      517.9    543.2   -251.0     501.9      165
## 
## Scaled residuals:
##      Min      1Q  Median      3Q     Max
## -2.4634 -0.6793 -0.0181  0.6950  2.6000
## 
## Random effects:
##   Groups   Name        Variance Std.Dev.
##   block    (Intercept) 0.07667  0.2769
##   Residual           1.02169  1.0108
## Number of obs: 173, groups: block, 7
## 
## Fixed effects:
##                               Estimate Std. Error      df t value
## (Intercept)                 -2.6399   0.2132 46.7683 -12.384
## namewt_percapita_trcy      -0.5377   0.2789 168.9358 -1.928
## namewt_percapita_tror      -1.0875   0.3076 167.5235 -3.535
## physical_barrier1          0.5559   0.2506 167.6282  2.218
## namewt_percapita_trcy:physical_barrier1 -0.7494   0.3663 167.6441 -2.046
## namewt_percapita_tror:physical_barrier1 -0.2664   0.3965 167.3819 -0.672
## 
## Pr(>|t|)                  2.27e-16 ***
## (Intercept)                 0.055547 .
## namewt_percapita_trcy      0.000527 ***
## namewt_percapita_tror      0.027889 *
## namewt_percapita_trcy:physical_barrier1 0.042344 *
## namewt_percapita_tror:physical_barrier1 0.502654
## --- 
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Correlation of Fixed Effects:
## (Intr) nmwt_prcpt_trc nmwt_prcpt_trr phys_1 nmwt_prcpt_trc:_1
## nmwt_prcpt_trc      -0.578
## nmwt_prcpt_trr      -0.520  0.404
## physcl_brr1         -0.642  0.488      0.443
## nmwt_prcpt_trc:_1   0.439 -0.758      -0.306      -0.682
## nmwt_prcpt_trr:_1   0.402 -0.309      -0.774      -0.631  0.432

emmap(pcwtmod_phys, ~physical_barrier | name, CI=T)

```

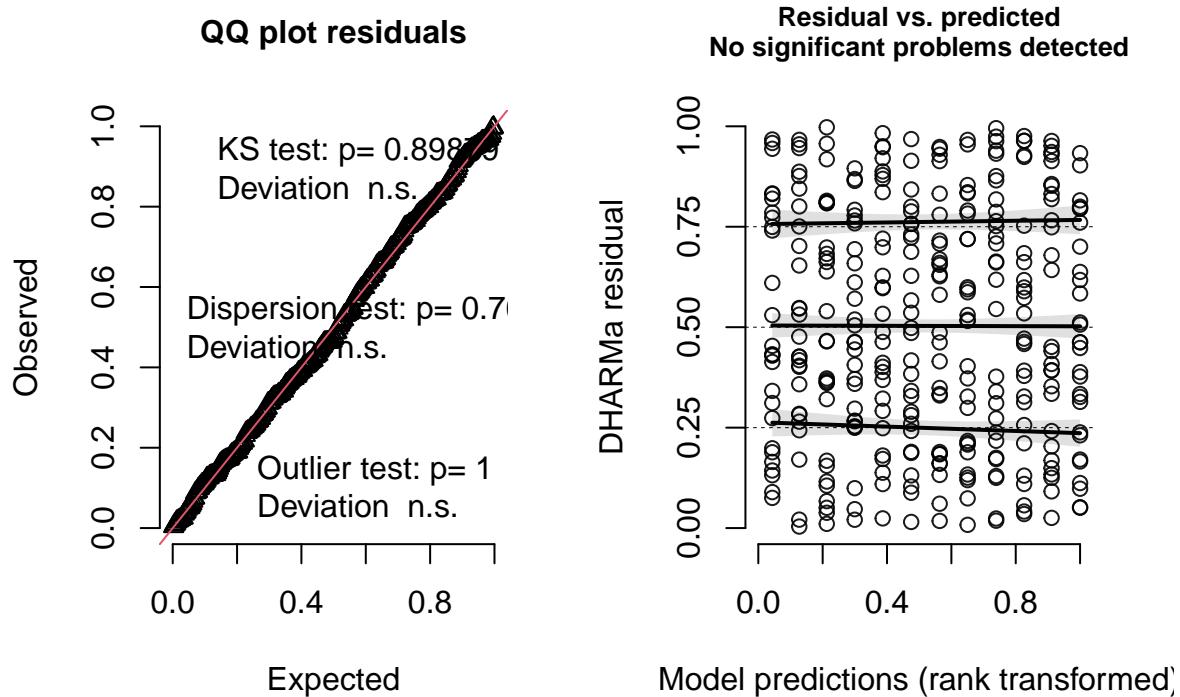


```
est<-emmeans(pcwtmod_phys, ~physical_barrier|name, type='response')
pairs(est)
```

```
## name = wt_percapita_goro:
## contrast                                estimate    SE  df t.ratio p.value
## physical_barrier0 - physical_barrier1   -0.556 0.255 172 -2.182  0.0305
##
## name = wt_percapita_trcy:
## contrast                                estimate    SE  df t.ratio p.value
## physical_barrier0 - physical_barrier1   0.193 0.273 173  0.710  0.4789
##
## name = wt_percapita_tror:
## contrast                                estimate    SE  df t.ratio p.value
## physical_barrier0 - physical_barrier1   -0.290 0.313 172 -0.926  0.3560
##
## Degrees-of-freedom method: kenward-roger
```

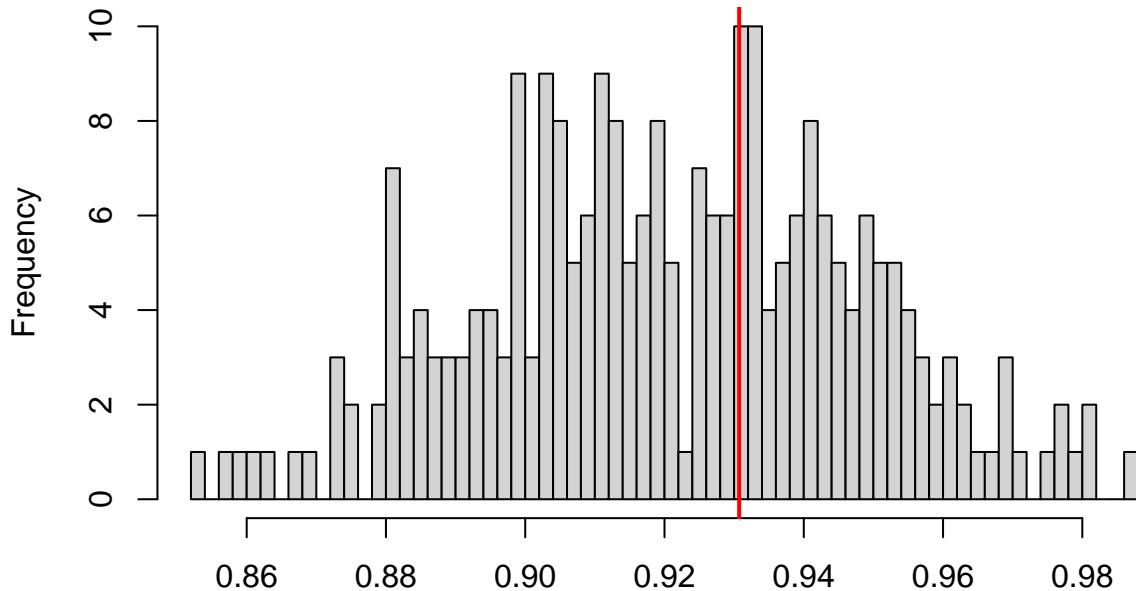
```
#### What about both physical and legacy? #####
### zeros and ones
zerofit_intxn<-glmmTMB(presence~name*physical_barrier*initial+(1 | block), family=binomial, data=countdat)
sim<-simulateResiduals(zerofit_intxn)
plot(sim)
```

DHARMA residual



```
testDispersion(sim)
```

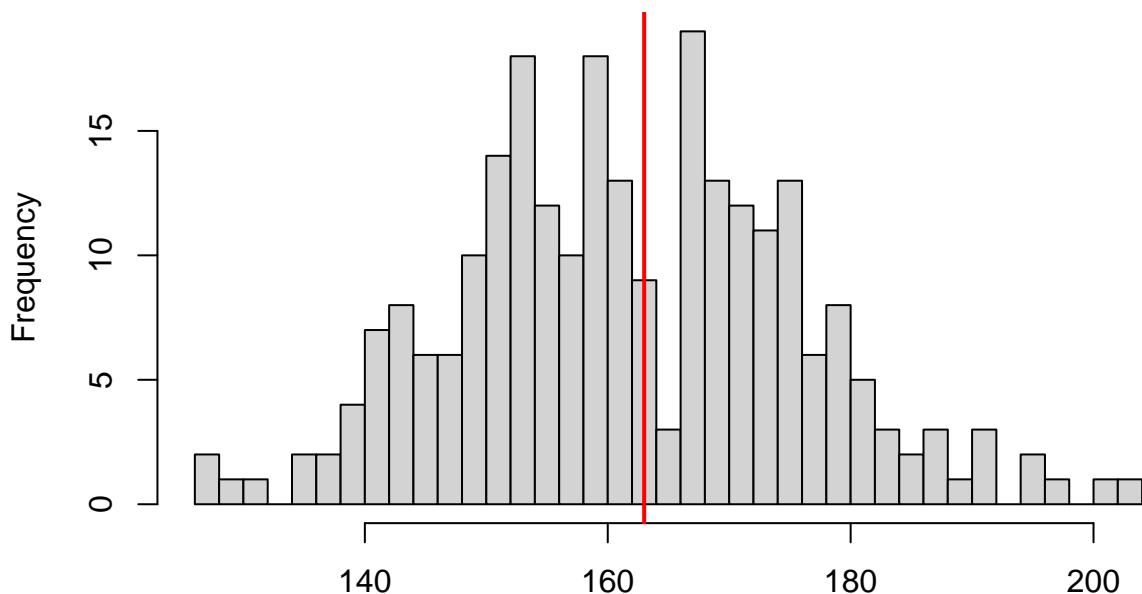
DHARMA nonparametric dispersion test via sd of residuals fitted vs. simulated



Simulated values, red line = fitted model. p-value (two.sided) = 0.76

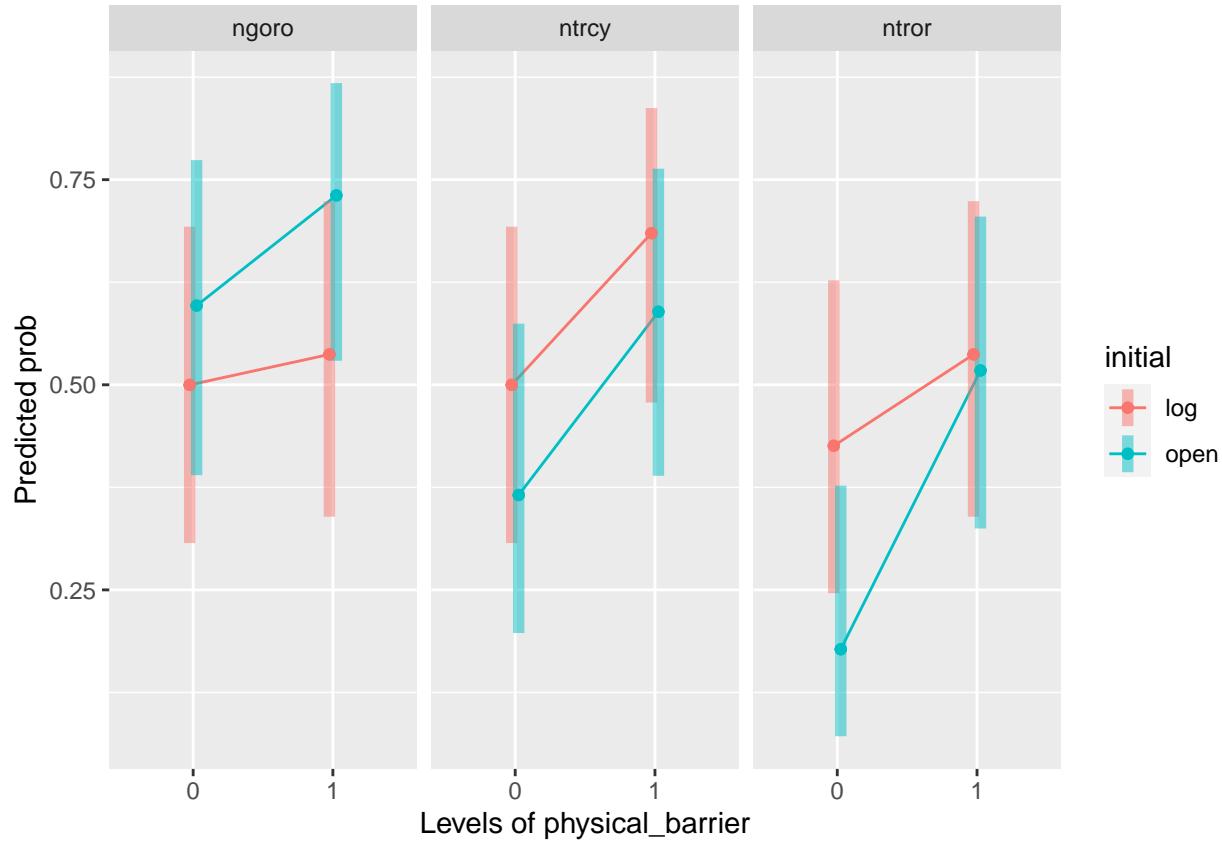
```
##  
## DHARMA nonparametric dispersion test via sd of residuals fitted vs.  
## simulated  
##  
## data: simulationOutput  
## dispersion = 1.0104, p-value = 0.76  
## alternative hypothesis: two.sided  
  
testZeroInflation(sim)
```

DHARMA zero-inflation test via comparison to expected zeros with simulation under H0 = fitted model



Simulated values, red line = fitted model. p-value (two.sided) = 0.928

```
##  
## DHARMA zero-inflation test via comparison to expected zeros with  
## simulation under H0 = fitted model  
##  
## data: simulationOutput  
## ratioObsSim = 1.0049, p-value = 0.928  
## alternative hypothesis: two.sided  
  
emmap(zeroFit_intxn, initial~physical_barrier|name, type='response', CI=T)
```



```
est<-emmeans(zerofit_intxn, ~physical_barrier|name|initial, type='response')
pairs(est)
```

```
## name = ngoro, initial = log:
## contrast                                odds.ratio      SE   df null z.ratio
## physical_barrier0 / physical_barrier1     0.862  0.470 Inf   1 -0.273
## p.value
## 0.7851
##
## name = ntrcy, initial = log:
## contrast                                odds.ratio      SE   df null z.ratio
## physical_barrier0 / physical_barrier1     0.460  0.260 Inf   1 -1.375
## p.value
## 0.1690
##
## name = ntror, initial = log:
## contrast                                odds.ratio      SE   df null z.ratio
## physical_barrier0 / physical_barrier1     0.639  0.351 Inf   1 -0.817
## p.value
## 0.4142
##
## name = ngoro, initial = open:
## contrast                                odds.ratio      SE   df null z.ratio
## physical_barrier0 / physical_barrier1     0.544  0.317 Inf   1 -1.045
## p.value
```

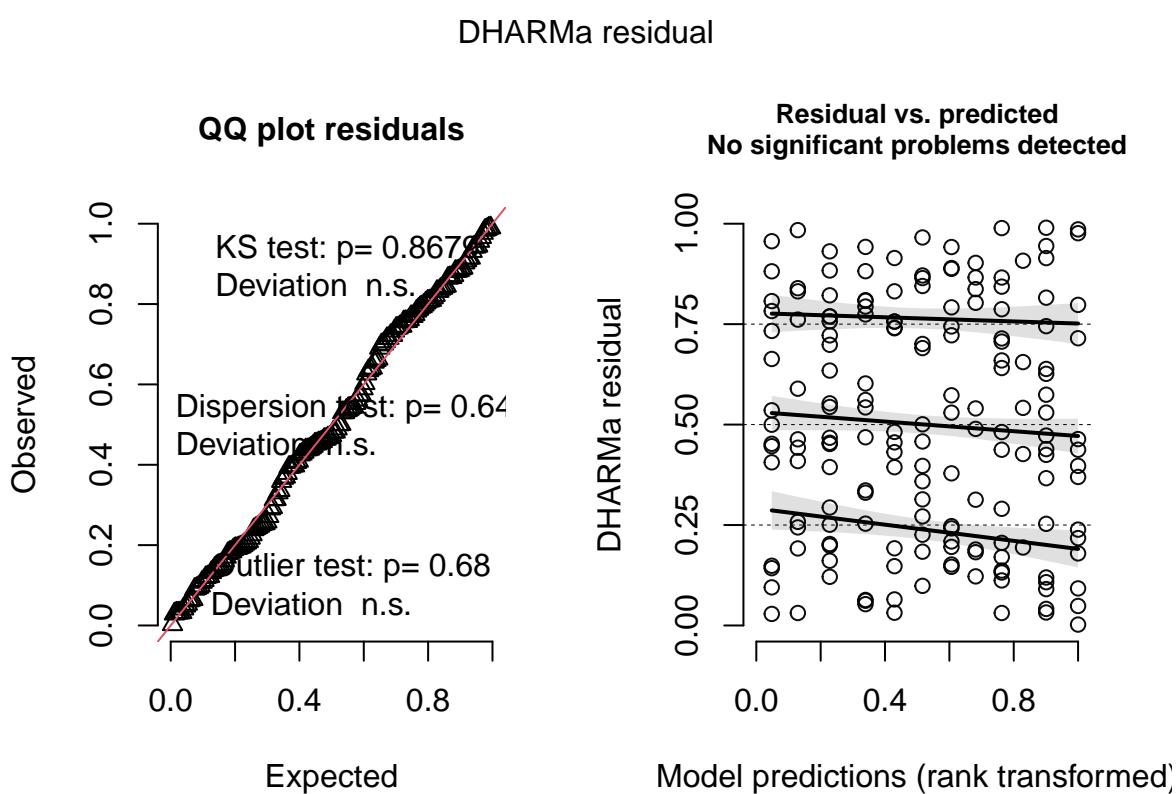
```

##    0.2958
##
## name = ntrcy, initial = open:
## contrast                               odds.ratio     SE   df null z.ratio
## physical_barrier0 / physical_barrier1      0.402 0.225 Inf   1 -1.628
## p.value
## 0.1036
##
## name = ntror, initial = open:
## contrast                               odds.ratio     SE   df null z.ratio
## physical_barrier0 / physical_barrier1      0.201 0.127 Inf   1 -2.544
## p.value
## 0.0110
##
## Tests are performed on the log odds ratio scale

### abundance with a truncated negbinom
countfit_intxn<-glmmTMB(posicounts~name*physical_barrier*initial+(1 | block), family=truncated_nbinom2()

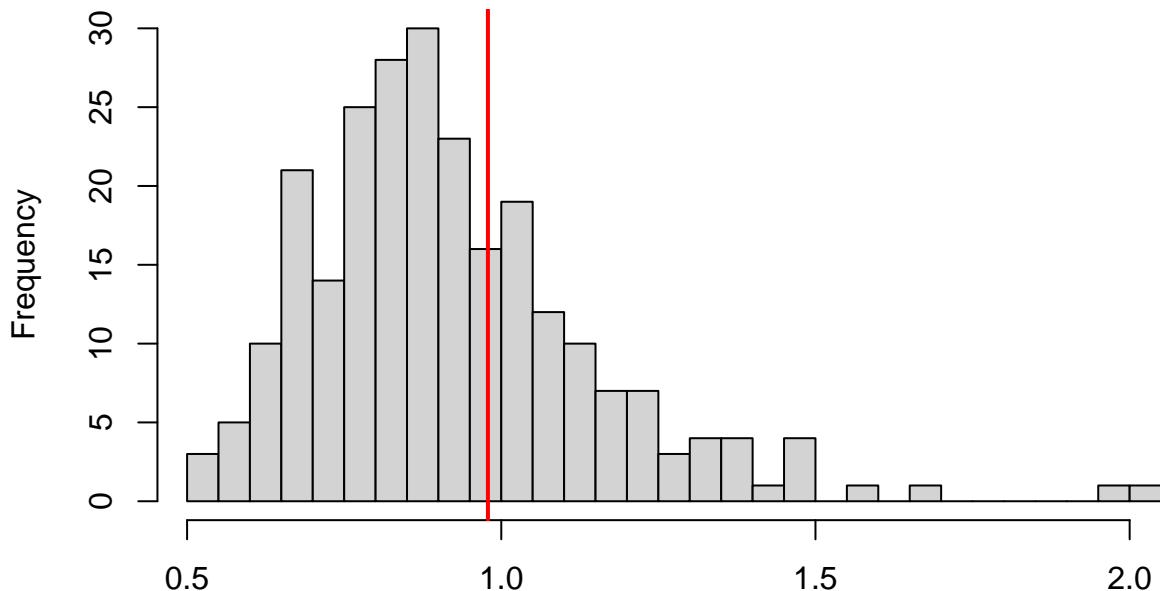
sim<-simulateResiduals(countfit_intxn)
plot(sim)

```



```
testDispersion(sim)
```

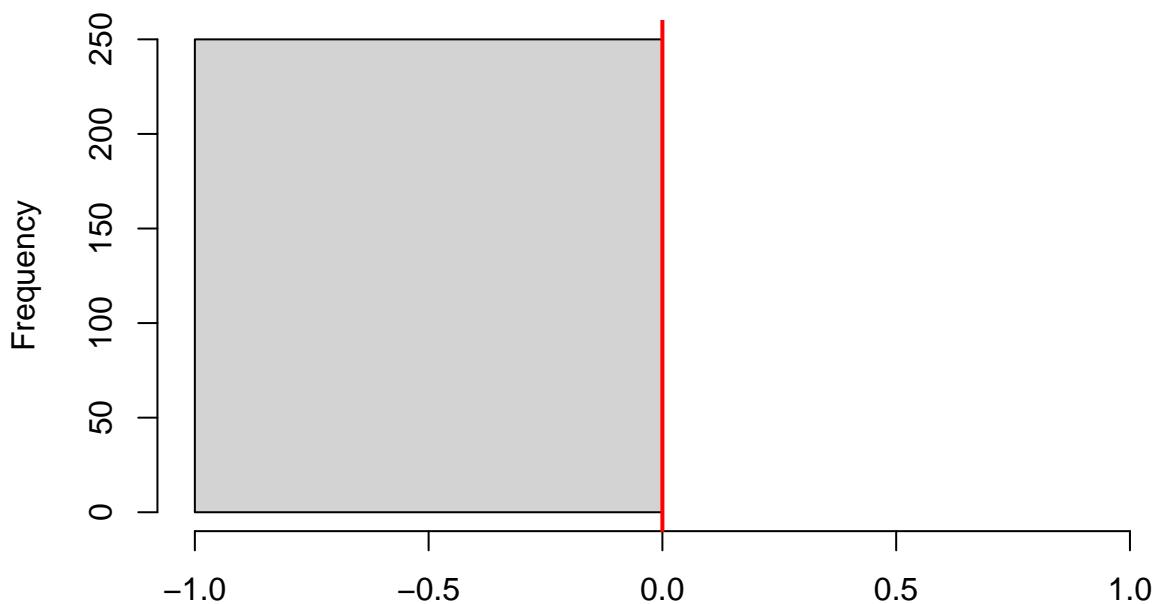
DHARMA nonparametric dispersion test via sd of residuals fitted vs. simulated



Simulated values, red line = fitted model. p-value (two.sided) = 0.648

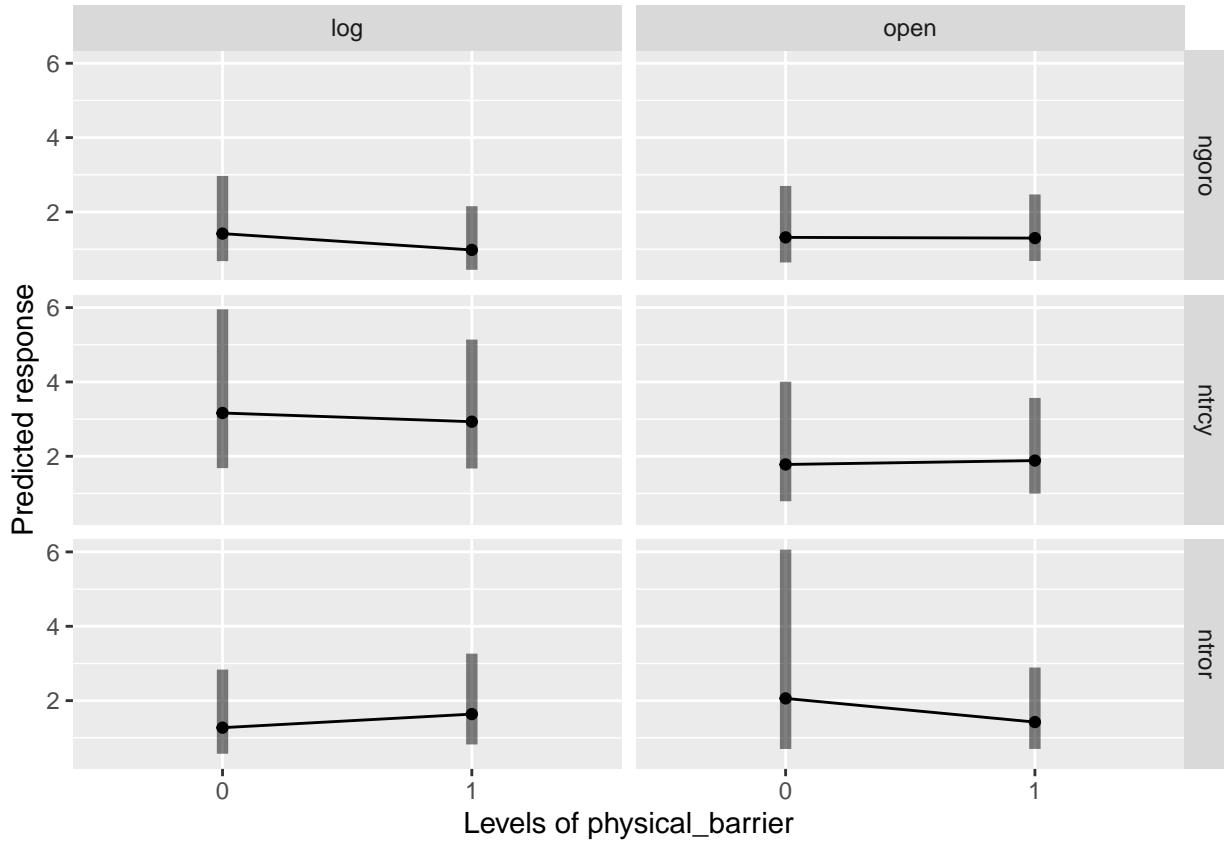
```
##  
##  DHARMA nonparametric dispersion test via sd of residuals fitted vs.  
##  simulated  
##  
##  data:  simulationOutput  
##  dispersion = 1.0637, p-value = 0.648  
##  alternative hypothesis: two.sided  
  
testZeroInflation(sim)
```

DHARMA zero-inflation test via comparison to expected zeros with simulation under H0 = fitted model



Simulated values, red line = fitted model. p-value (two.sided) = 1

```
##  
## DHARMA zero-inflation test via comparison to expected zeros with  
## simulation under H0 = fitted model  
##  
## data: simulationOutput  
## ratioObsSim = NaN, p-value = 1  
## alternative hypothesis: two.sided  
  
emmap(countfit_intxn, ~physical_barrier | name | initial, type='response', CI=T)
```



```
est<-emmeans(countfit_intxn, ~physical_barrier|name|initial, type='response')
pairs(est)
```

```
## name = ngoro, initial = log:
## contrast                                ratio      SE  df null z.ratio p.value
## physical_barrier0 / physical_barrier1 1.448 0.727 Inf   1  0.738  0.4603
##
## name = ntrcy, initial = log:
## contrast                                ratio      SE  df null z.ratio p.value
## physical_barrier0 / physical_barrier1 1.080 0.432 Inf   1  0.192  0.8479
##
## name = ntror, initial = log:
## contrast                                ratio      SE  df null z.ratio p.value
## physical_barrier0 / physical_barrier1 0.778 0.387 Inf   1 -0.506  0.6132
##
## name = ngoro, initial = open:
## contrast                                ratio      SE  df null z.ratio p.value
## physical_barrier0 / physical_barrier1 1.015 0.444 Inf   1  0.033  0.9735
##
## name = ntrcy, initial = open:
## contrast                                ratio      SE  df null z.ratio p.value
## physical_barrier0 / physical_barrier1 0.944 0.467 Inf   1 -0.116  0.9080
##
## name = ntror, initial = open:
## contrast                                ratio      SE  df null z.ratio p.value
```

```

##  physical_barrier0 / physical_barrier1 1.447 0.912 Inf      1    0.586  0.5580
##
## Tests are performed on the log scale

### additive example
zerofit_add<-glmmTMB(presence~name*physical_barrier+name*initial+(1 | block), family=binomial, data=counts)
countfit_add<-glmmTMB(posicounts~name*physical_barrier+name*initial+(1 | block), family=truncated_nbino)

### per capita biomass

# model - intxn
pcwtmod_intxn<-lmer(log_wt~name*physical_barrier*initial+(1|block), data=pcwtdat, REML=FALSE)

# model - no intxn
pcwtmod_add<-lmer(log_wt~name*physical_barrier+name*initial+(1|block), data=pcwtdat, REML=FALSE)

#### 2021 model comparison for counts #####
## zeros
zero_candmods<-list("Plot type"=zerofit,
                      "Physical barrier"=zerofit_phys,
                      "Nutrient island"=zerofit_leg,
                      "Physical Barrier + Nutrient Island"=zerofit_add,
                      "Physical Barrier x Nutrient Island"=zerofit_intxn)
aictab(zero_candmods)

##
## Model selection based on AICc:
##
##          K   AICc Delta_AICc AICcWt Cum.Wt      LL
## Physical barrier        7  457.58      0.00   0.49   0.49 -221.62
## Physical Barrier + Nutrient Island 10  457.90      0.32   0.42   0.91 -218.61
## Physical Barrier x Nutrient Island 13  462.04      4.46   0.05   0.96 -217.45
## Nutrient island         7  462.89      5.31   0.03   1.00 -224.27
## Plot type               19 466.87      9.30   0.00   1.00 -213.23

pairs(emmeans(zerofit_add, ~physical_barrier|name|initial))

## name = ngoro, initial = log:
## contrast                         estimate     SE  df z.ratio p.value
## physical_barrier0 - physical_barrier1 -0.365 0.397 Inf -0.918  0.3584
##
## name = ntrcy, initial = log:
## contrast                         estimate     SE  df z.ratio p.value
## physical_barrier0 - physical_barrier1 -0.844 0.398 Inf -2.122  0.0338
##
## name = ntror, initial = log:
## contrast                         estimate     SE  df z.ratio p.value
## physical_barrier0 - physical_barrier1 -0.971 0.407 Inf -2.385  0.0171
##
## name = ngoro, initial = open:
## contrast                         estimate     SE  df z.ratio p.value

```

```

##  physical_barrier0 - physical_barrier1   -0.365 0.397 Inf  -0.918  0.3584
##
## name = ntrcy, initial = open:
## contrast                           estimate    SE  df z.ratio p.value
## physical_barrier0 - physical_barrier1   -0.844 0.398 Inf  -2.122  0.0338
##
## name = ntror, initial = open:
## contrast                           estimate    SE  df z.ratio p.value
## physical_barrier0 - physical_barrier1   -0.971 0.407 Inf  -2.385  0.0171
##
## Results are given on the log odds ratio (not the response) scale.

# best fit model is a tie between physical barrier and additive model (deltaAIC=0.32)
# i will show results from additive model.

# counts
count_candmods<-list("Plot type"=countfit,
                      "Physical barrier"=countfit_phys,
                      "Nutrient island"=countfit_leg,
                      "Physical Barrier + Nutrient Island"=countfit_add,
                      "Physical Barrier x Nutrient Island"=countfit_intxn)
aictab(count_candmods)

##
## Model selection based on AICc:
##
##                               K   AICc Delta_AICc AICcWt Cum.Wt      LL
## Nutrient island            8  621.48     0.00  0.75  0.75 -302.30
## Physical barrier           8  623.89     2.41  0.22  0.97 -303.51
## Physical Barrier + Nutrient Island 11 627.98     6.50  0.03  1.00 -302.17
## Physical Barrier x Nutrient Island 14 634.07    12.59  0.00  1.00 -301.71
## Plot type                  20 640.85    19.37  0.00  1.00 -297.66

pairs(emmeans(countfit_leg, ~initial|name))

## name = ngoro:
## contrast   estimate    SE  df z.ratio p.value
## log - open -0.0920 0.333 Inf  -0.276  0.7826
##
## name = ntrcy:
## contrast   estimate    SE  df z.ratio p.value
## log - open  0.4984 0.309 Inf   1.615  0.1062
##
## name = ntror:
## contrast   estimate    SE  df z.ratio p.value
## log - open -0.0716 0.376 Inf  -0.190  0.8490
##
## Results are given on the log (not the response) scale.

# best fit model is nutrient island (next best deltaAIC=2.41 and it's physical barrier.)
# I will show the results for nutrient island.

```

```

##### 2021 model comparison for per capita biomass #####
pcwt_candmods<-list("Plot type"=pcwtmod,
                      "Physical barrier"=pcwtmod_phys,
                      "Nutrient island"=pcwtmod_leg,
                      "Physical Barrier + Nutrient Island" = pcwtmod_add,
                      "Physical Barrier x Nutrient Island"=pcwtmod_intxn)
aictab(pcwt_candmods)

##
## Model selection based on AICc:
##
##                                     K   AICc Delta_AICc AICcWt Cum.Wt      LL
## Physical barrier                  8 518.81      0.00  0.52  0.52 -250.97
## Nutrient island                  8 520.36      1.55  0.24  0.75 -251.74
## Physical Barrier + Nutrient Island 11 520.57      1.76  0.21  0.97 -248.47
## Physical Barrier x Nutrient Island 14 524.43      5.62  0.03  1.00 -246.88
## Plot type                         20 533.29     14.48  0.00  1.00 -243.88

pairs(emmeans(pcwtmod_phys, ~physical_barrier|name))

## name = wt_percapita_goro:
## contrast                           estimate    SE  df t.ratio p.value
## physical_barrier0 - physical_barrier1 -0.556 0.255 172 -2.182  0.0305
##
## name = wt_percapita_trcy:
## contrast                           estimate    SE  df t.ratio p.value
## physical_barrier0 - physical_barrier1  0.193 0.273 173  0.710  0.4789
##
## name = wt_percapita_tror:
## contrast                           estimate    SE  df t.ratio p.value
## physical_barrier0 - physical_barrier1 -0.290 0.313 172 -0.926  0.3560
##
## Degrees-of-freedom method: kenward-roger

# best fit model is physical barrier by a hair, then nutrient island, then additive model. They are all
# deltaAIC physical barrier - nutrient island = 1.55
# deltaAIC physical - nutrient island = 1.76
# I will show the results for additive model.

#### figures ####
# best fit models - zeros
# colors
mimiscols<-c("#D66972","#108780")
# zerofit_add<-glmmTMB(presence~name*physical_barrier+name*initial+(1 / block), family=binomial, data=c

zfit_est<-as.data.frame(emmeans(zerofit_add,~initial|physical_barrier|name, type='response'))
zfit_est$name<-c(rep("ngoro", 4), rep("ntrcy",4), rep("ntror",4))

pl1<-ggplot(zfit_est,aes(physical_barrier, prob,group=initial),)+ 
  scale_color_manual(values=mimiscols)+ 
  geom_point(aes(col=initial), size=2, position=position_dodge(width=0.5))+
```

```

geom_linerange(aes(ymin=asymp.LCL, ymax=asymp.UCL, col=initial), position=position_dodge(width=0.5))+
geom_line(aes(col=initial), position=position_dodge(width=0.5))+ 
facet_wrap(vars(name))+ 
theme_bw()+
theme(strip.text.x = element_text(size=0),
      strip.background = element_blank(),
      axis.text=element_text(size=15),
      axis.title=element_text(size=20),
      legend.text=element_text(size=15),
      legend.title=element_text(size=15),
      legend.position="top")+
xlab("Physical Barrier")+
ylab("Occurrence")+
geom_jitter(data=countdat,
            aes(x=physical_barrier, y=presence, color=initial),
            height=0.1,
            alpha=0.5)+ 
labs(color = "Initial Plot Type")

# best fit model - counts
# countfit_leg<-glmmTMB(posicounts~name*initial+(1 / block), family=truncated_nbinom2(), data=countdat)

cfit_est<-as.data.frame(emmeans(countfit_leg, ~initial|name, type='response'))
cfit_est$name<-c(rep("ngoro", 2), rep("ntrcy", 2), rep("ntror", 2))

pl2<-ggplot(cfit_est,aes(initial, response, group=1))+ 
  geom_jitter(data=countdat,
              aes(x=initial, y=posicounts),
              width=0.1,
              alpha=0.4,
              color="gray")+
  geom_point(size=2, color="black")+
  geom_linerange(aes(ymin=asymp.LCL, ymax=asymp.UCL),color="black")+
  geom_line(color="black")+
  facet_wrap(vars(name))+ 
  theme_bw()+
  theme(strip.text.x = element_text(size=0),
        strip.background = element_blank(),
        axis.text=element_text(size=15),
        axis.title=element_text(size=20),
        legend.text=element_text(size=15),
        legend.title=element_text(size=15))+ 
  xlab("Initial Plot Type")+
  ylab("Abundance")+
  labs(color = "Initial Plot \n Type")

# best fit model - pcbiomass
# pcwtmod_phys<-lmer(log_wt~name*physical_barrier+(1/block), data=pcwtdat, REML=FALSE)

pcbfit_est<-as.data.frame(emmeans(pcwtmod_add, ~physical_barrier|name|initial, type='response')) 

pl3<-ggplot(pcbfit_est,aes(physical_barrier, emmean, group=initial))+ 

```

```

scale_color_manual(values=mimiscols)+  

geom_point(aes(col=initial), size=2, position=position_dodge(width=0.5))+  

geom_linerange(aes(ymin=lower.CL, ymax=upper.CL, col=initial), position=position_dodge(width=0.5))+  

geom_line(aes(col=initial), position=position_dodge(width=0.5))+  

facet_wrap(vars(name))+  

theme_bw()  

theme(strip.text.x = element_text(size=0),  

      strip.background = element_blank(),  

      axis.text=element_text(size=15),  

      axis.title=element_text(size=20),  

      legend.text=element_text(size=15),  

      legend.title=element_text(size=15),  

      legend.position="top")  

xlab("Physical Barrier")  

ylab("log(Biomass)")  

geom_jitter(data=pcwtdat,  

            aes(x=physical_barrier, y=log_wt, color=initial),  

            height=0.1,  

            alpha=0.5)+  

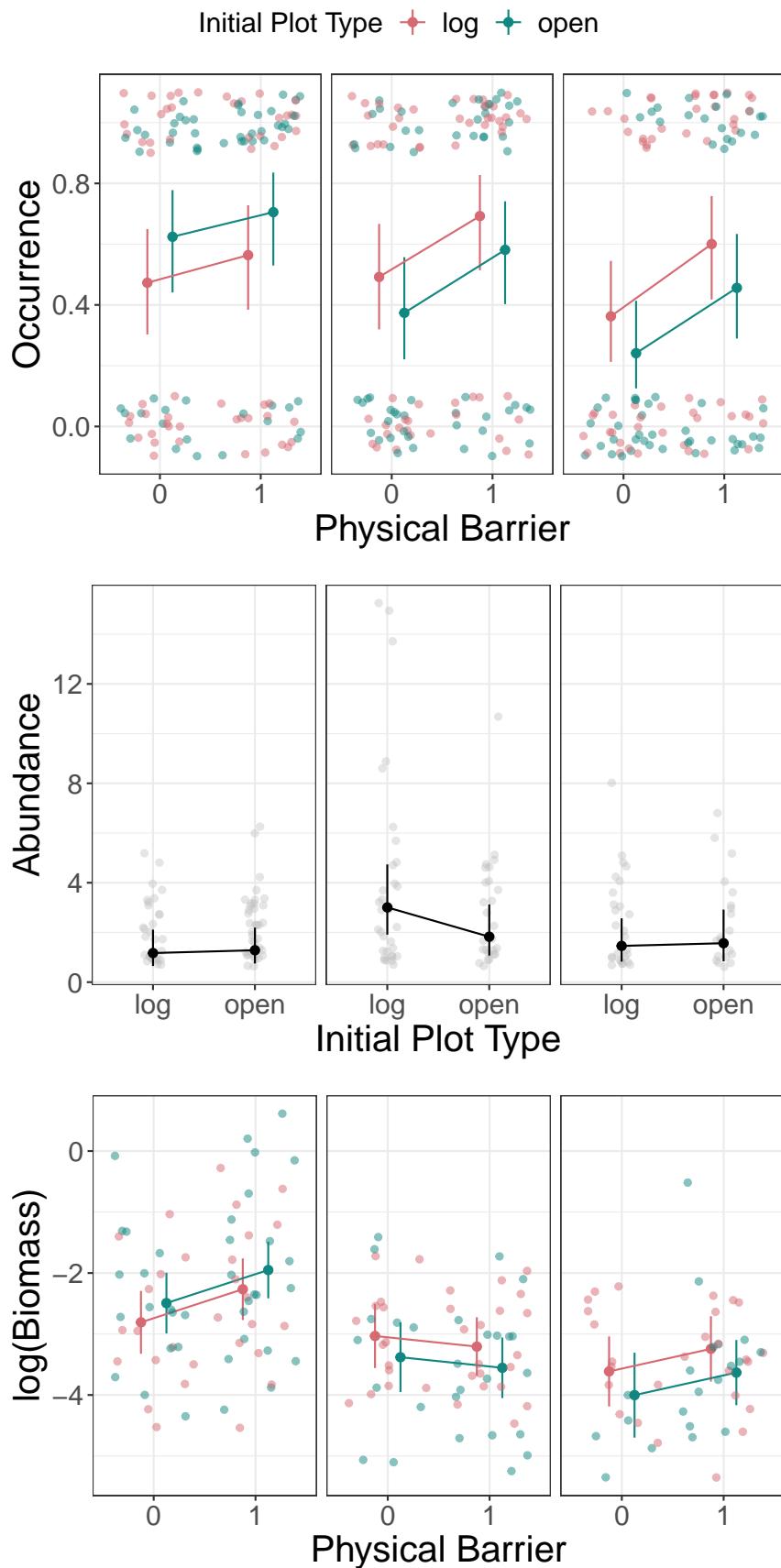
labs(color = "Initial Plot Type")  

# pl3  

### altogether now

```

```
ggarrange(pl1, pl2, pl3, ncol=1, common.legend = T)
```



```

# packages
require(lme4)
require(emmeans)
require(pscl)
require(glmmTMB)
require(tidyr)
require(DHARMa)
require(ggplot2)
require(AICcmodavg)
require(ggpubr)

##### 2022 data #####
# read csv
dat<-read.csv('nplants_data_2022.csv', header=T)
dat1<-dat[which(dat$seedling_trt==1),]
dat1$physical_barrier<-as.factor(dat1$physical_barrier)
dat1$block<-as.factor(dat1$block)
names(dat1)

```

2022

```

## [1] "block"                  "transect"                "initial"
## [4] "current_plot_type"      "dispersal_treatment" "seedling_trt"
## [7] "ntrcy_germ"             "ngoro_germ"              "ntror_germ"
## [10] "ntrcy_plants"           "ngoro_plants"            "ntror_plants"
## [13] "ntrcy_biomass"          "ngoro_biomass"           "ntror_biomass"
## [16] "ntrcy_seeds"            "ngoro_seeds"             "ntror_seeds"
## [19] "physical_barrier"        "wt_percapita_trcy"     "wt_max15_trcy"
## [22] "wt_percapita_goro"      "wt_max15_goro"           "wt_percapita_tror"
## [25] "wt_max15_tror"          "ntrcy_tot"               "ngoro_tot"
## [28] "ntror_tot"               "notes"                   "to.do"
## [31] "tree.fallen"             "tree.cleared"

# subset
dat2<-dat1[,c(1,2:4,26:28,19)] # these are block, transect, initial, current_plot_type, ngoro_plants, n
head(dat2)

##   block transect initial current_plot_type ntrcy_tot ngoro_tot ntror_tot
## 1    1      log    1.01       gap        1       1       4
## 2    1      log    1.4       gap        2       2       1
## 3    1      log    1.6       gap        2       2       1
## 4    1      log    1.8       gap        1       0       3
## 5    1      log    1.02  insitu_log       8       0       0
## 6    1      log    1.14  insitu_log       2       1       0
##   physical_barrier
## 1                      0
## 2                      0
## 3                      0
## 4                      0
## 5                      1
## 6                      1

```

```

## because of the way weeding worked, where we thinned only once and probably too early in the season,
# I am choosing to use the total number of a species that popped up in the zone of planting. this requi
# did not survive. This value is called 'tot' (e.g. ngoro_tot).
# It's impossible to get the same dataset as the one from 2021 because we planted the plants and only c
# to get the total in the "tot" columns, I wrote a formula in excel, written below
# total = if((t0+t2=0),0,if(t0=0,t2,if(t2=0,t0, if(t0+t2=1,1,(t0-1+t2)))))
# in this expression t0 is the germ value (e.g. ngoro_germ) and t2 is the plants value (e.g. ngoro_plan
# I can run the analysis with germ, plants, or tot. I'm choosing to do the analysis on tot, but we can

# pivot
dat22<-as.data.frame(dat2 %>% pivot_longer(c(ntrcy_tot, ngoro_tot, ntror_tot)))
range(dat22$value)

## [1] 0 25

dat22$value>15 # one sample is larger than 15, it is a tror.

## [1] FALSE FALSE
## [13] FALSE FALSE
## [25] FALSE FALSE
## [37] FALSE FALSE
## [49] FALSE FALSE
## [61] FALSE FALSE
## [73] FALSE FALSE
## [85] FALSE FALSE
## [97] FALSE FALSE
## [109] FALSE FALSE
## [121] FALSE FALSE
## [133] FALSE FALSE
## [145] FALSE FALSE
## [157] FALSE FALSE
## [169] FALSE FALSE
## [181] FALSE FALSE
## [193] FALSE FALSE
## [205] FALSE FALSE
## [217] FALSE FALSE
## [229] FALSE FALSE
## [241] FALSE FALSE
## [253] FALSE FALSE
## [265] FALSE FALSE
## [277] FALSE FALSE
## [289] FALSE FALSE
## [301] FALSE FALSE
## [313] FALSE FALSE
## [325] FALSE FALSE

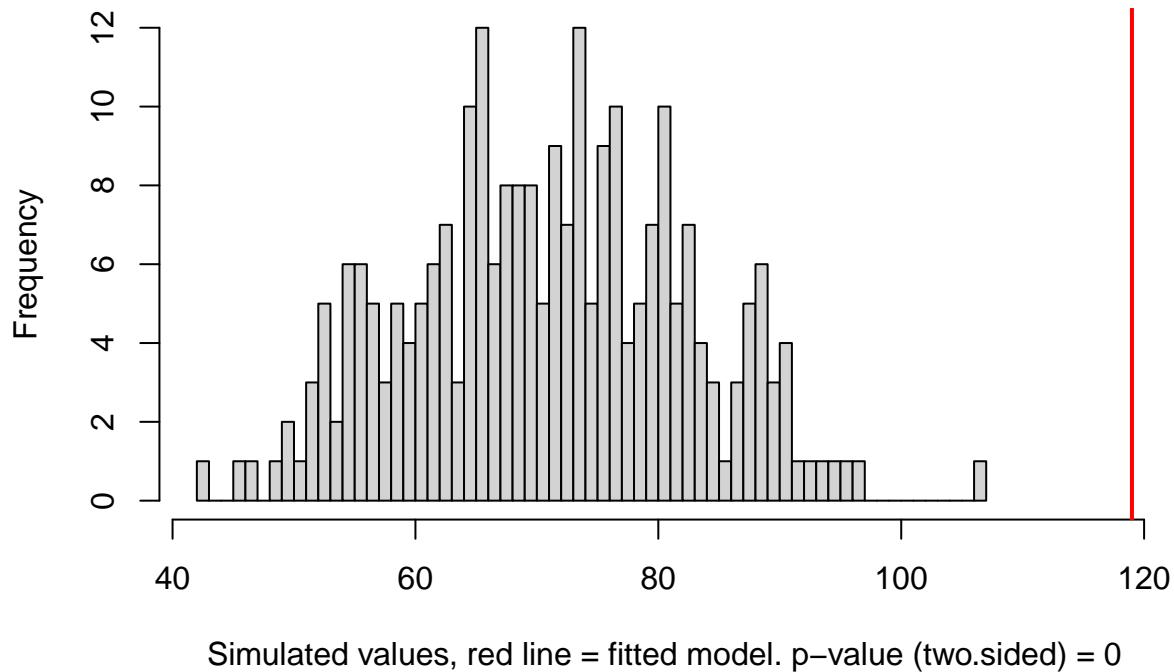
# max out at 15
dat22$value<-as.numeric(ifelse(dat22$value>15, 15, dat22$value))

#####
##### TREATMENT RESPONSE PRELIM ANALYSIS #####
#####
```

```
##### treatment response: do analysis for count by species #####
# I am going to do the analysis as in 2021 now, with final counts and biomass.
countmod<-glmmTMB(value~name*current_plot_type+(1|block), family="poisson", data=dat22)

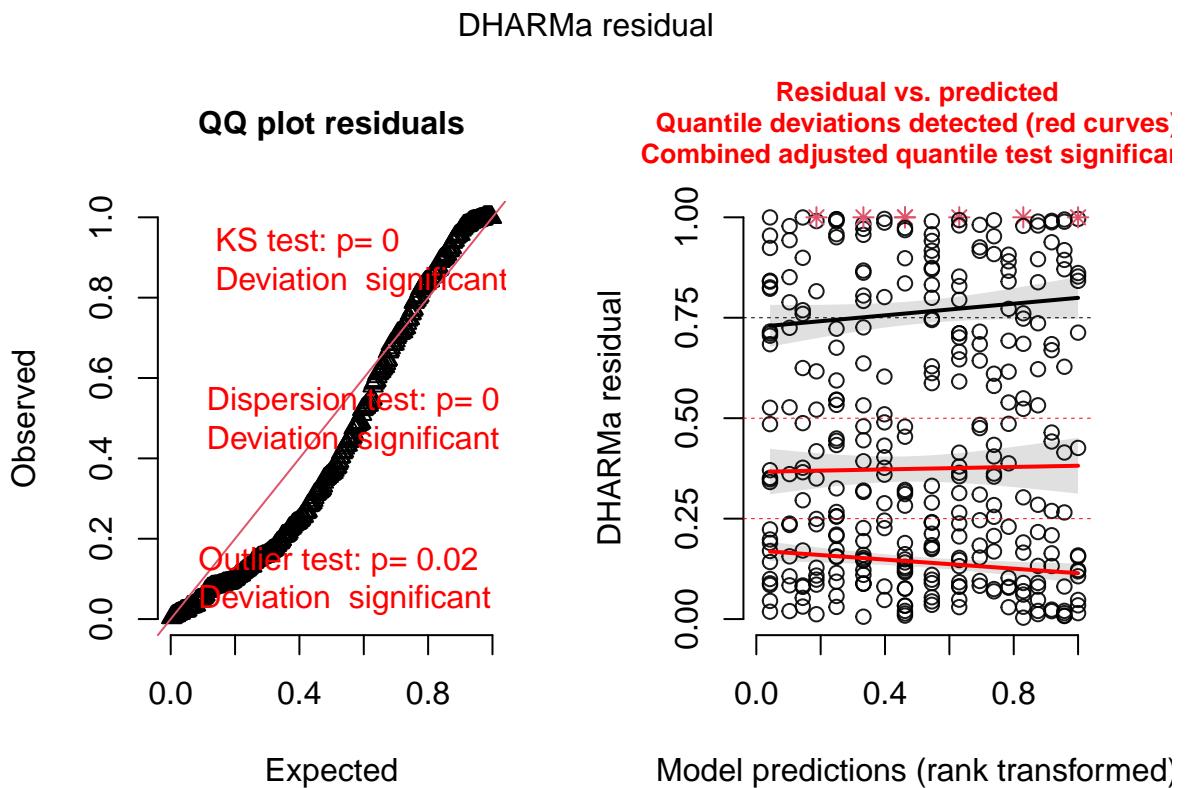
# test for fit and zero inflation
sim<-simulateResiduals(countmod)
testZeroInflation(sim) # zero-inflated so need something else!
```

DHARMA zero-inflation test via comparison to expected zeros with simulation under H0 = fitted model



```
##
## DHARMA zero-inflation test via comparison to expected zeros with
## simulation under H0 = fitted model
##
## data: simulationOutput
## ratioObsSim = 1.6664, p-value < 2.2e-16
## alternative hypothesis: two.sided

plot(sim)
```

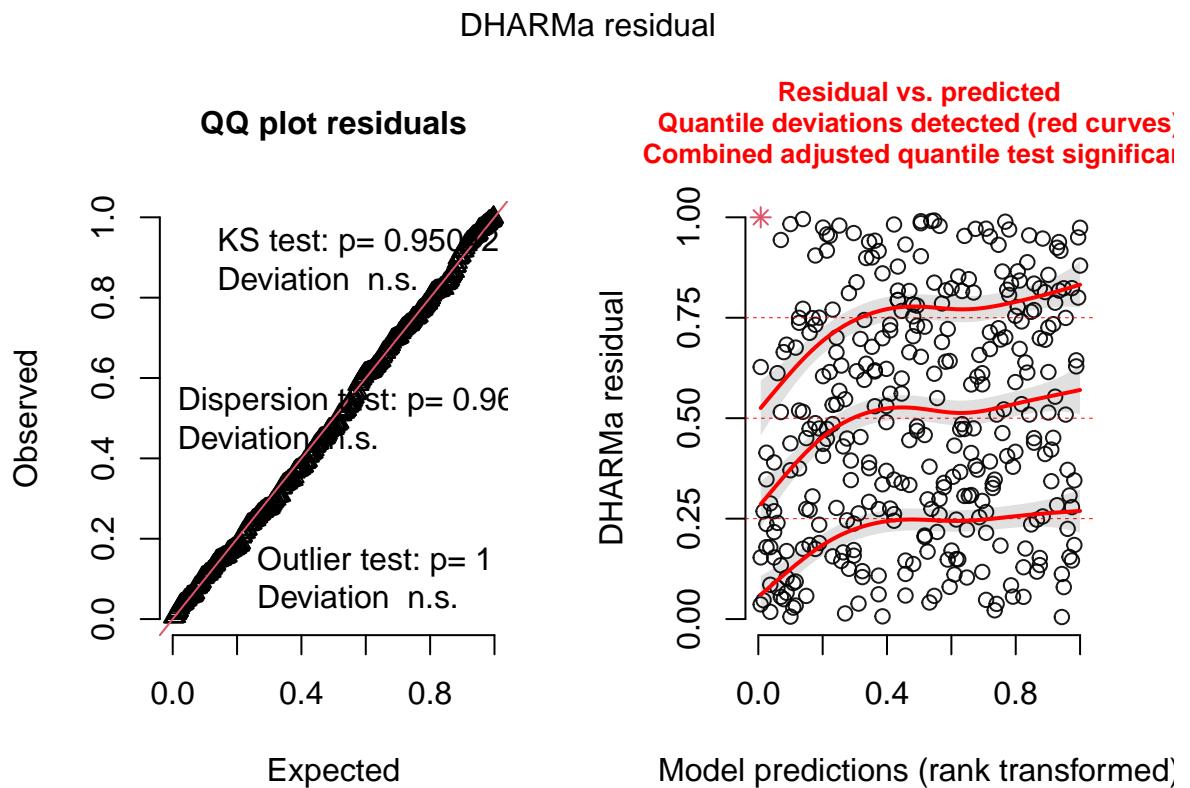


```

## going to do a hurdle model, which assumes a zero is only generated in one way
# https://jsdajournal.springeropen.com/articles/10.1186/s40488-021-00121-4
# #https://stats.stackexchange.com/questions/81457/what-is-the-difference-between-zero-inflated-and-hur

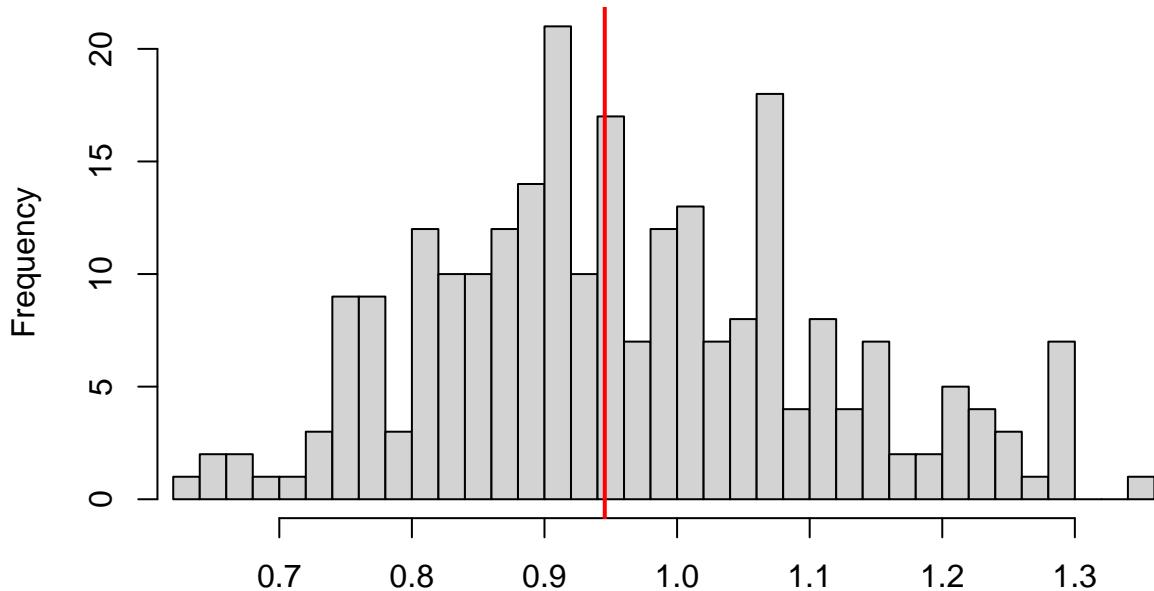
# hurdle model
# using examples as presented here: https://www.biorxiv.org/content/biorxiv/suppl/2017/05/01/132753.DC1/
fit3<-glmmTMB(value~name+current_plot_type+(1|block), ziformula=~., family=nbinom2(), data=dat22) # mod
sim3<-simulateResiduals(fit3)
plot(sim3)

```



```
testDispersion(sim3) # looks ok !
```

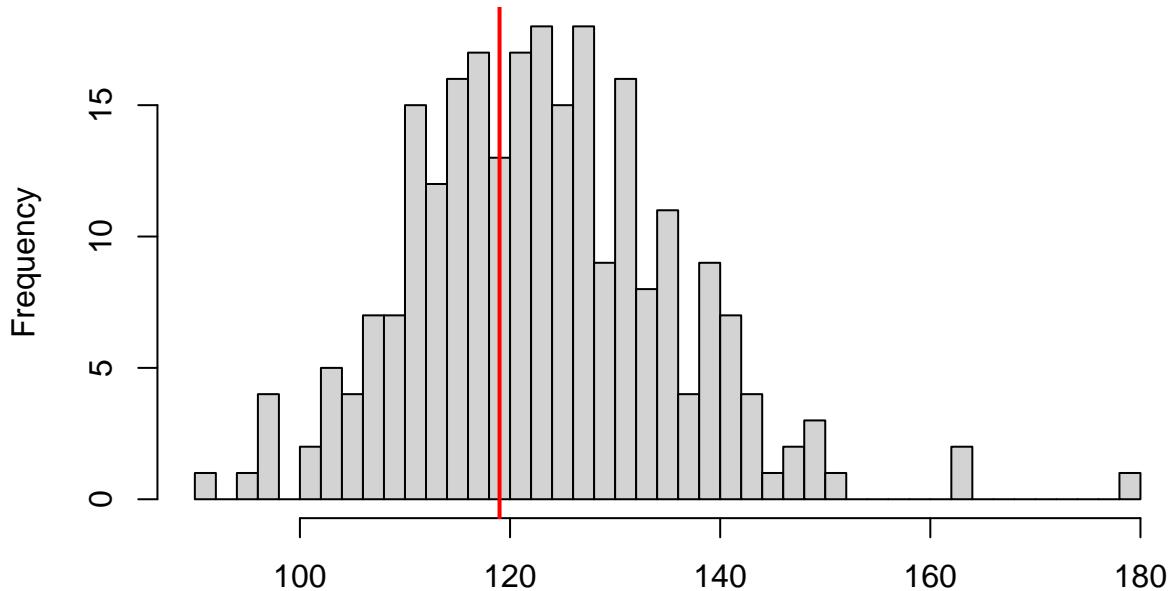
DHARMA nonparametric dispersion test via sd of residuals fitted vs. simulated



Simulated values, red line = fitted model. p-value (two.sided) = 0.968

```
##  
## DHARMA nonparametric dispersion test via sd of residuals fitted vs.  
## simulated  
##  
## data: simulationOutput  
## dispersion = 0.98275, p-value = 0.968  
## alternative hypothesis: two.sided  
  
testZeroInflation(sim3) # looks ok !
```

DHARMA zero-inflation test via comparison to expected zeros with simulation under H0 = fitted model



Simulated values, red line = fitted model. p-value (two.sided) = 0.792

```
## 
## DHARMA zero-inflation test via comparison to expected zeros with
## simulation under H0 = fitted model
## 
## data: simulationOutput
## ratioObsSim = 0.96353, p-value = 0.792
## alternative hypothesis: two.sided

summary(fit3)

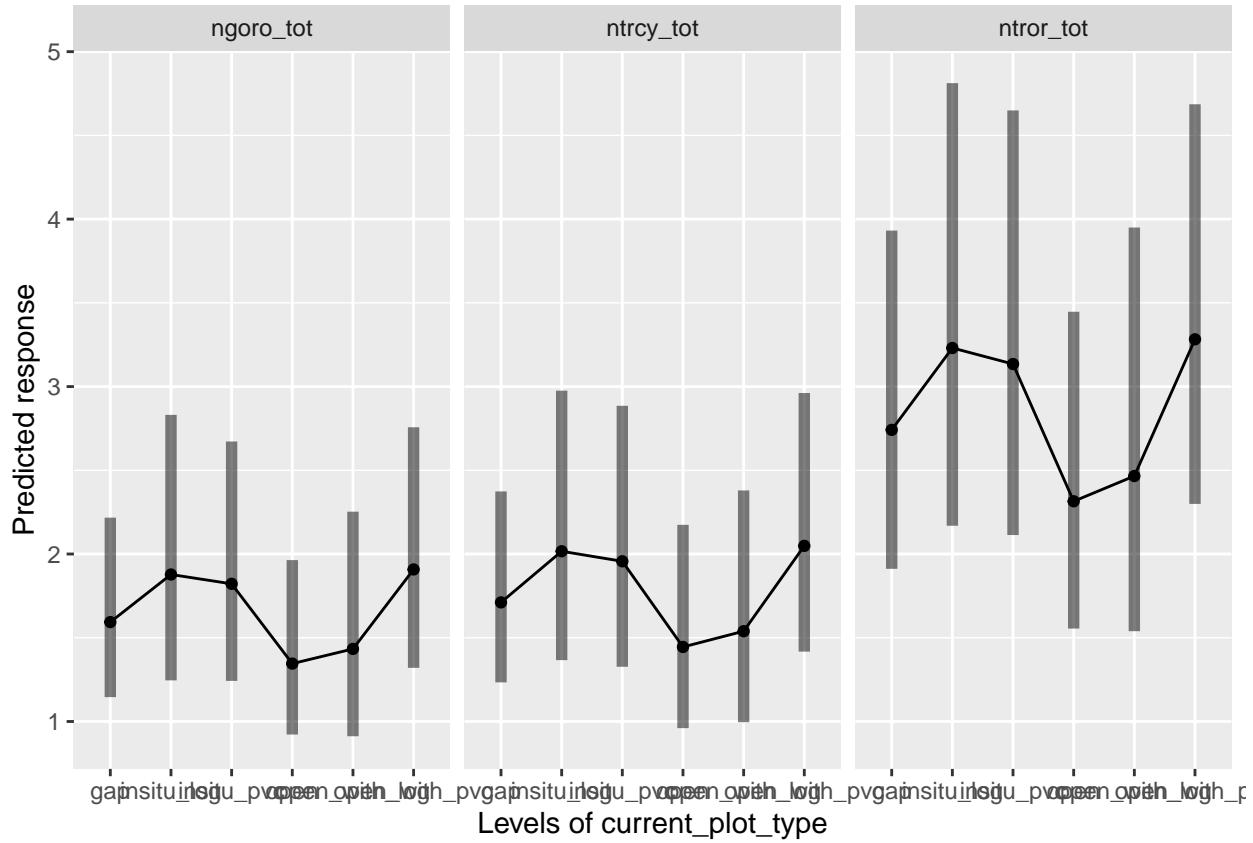
## Family: nbinom2  ( log )
## Formula:           value ~ name + current_plot_type + (1 | block)
## Zero inflation:    ~.
## Data: dat22
## 
##      AIC      BIC   logLik deviance df.resid
##  1229.7  1302.3 -595.9   1191.7      317
## 
## Random effects:
## 
## Conditional model:
## Groups Name        Variance Std.Dev.
## block  (Intercept) 0.007064 0.08405
## Number of obs: 336, groups: block, 7
##
```

```

## Zero-inflation model:
## Groups Name      Variance Std.Dev.
## block (Intercept) 2.217    1.489
## Number of obs: 336, groups: block, 7
##
## Dispersion parameter for nbinom2 family (): 1.53
##
## Conditional model:
##                               Estimate Std. Error z value Pr(>|z|)
## (Intercept)                0.46596   0.16852   2.765  0.00569 **
## namentrcy_tot              0.07124   0.16537   0.431  0.66664
## namentror_tot              0.54253   0.16727   3.243  0.00118 **
## current_plot_typeinsitu_log 0.16415   0.22352   0.734  0.46272
## current_plot_typeinsitu_pvc 0.13402   0.21840   0.614  0.53944
## current_plot_typeopen       -0.16911   0.21437  -0.789  0.43019
## current_plot_typeopen_with_log -0.10599   0.24127  -0.439  0.66044
## current_plot_typeopen_with_pvc 0.18007   0.20993   0.858  0.39102
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Zero-inflation model:
##                               Estimate Std. Error z value Pr(>|z|)
## (Intercept)                -5.2167   3.3577  -1.554  0.120
## namentrcy_tot               1.6850   2.9719   0.567  0.571
## namentror_tot               3.7206   3.0319   1.227  0.220
## current_plot_typeinsitu_log -2.0663   2.4030  -0.860  0.390
## current_plot_typeinsitu_pvc -0.7034   1.2255  -0.574  0.566
## current_plot_typeopen        1.2724   0.9896   1.286  0.199
## current_plot_typeopen_with_log 0.5711   1.0626   0.537  0.591
## current_plot_typeopen_with_pvc -17.1386 3844.9771 -0.004  0.996

emmap(fit3, ~current_plot_type | name, type='response', CI=T)

```



```
# visualize
est<-emmeans(fit3, ~current_plot_type|name, type='response')
pairs(est)
```

```
## name = ngoro_tot:
## contrast                                ratio    SE  df null z.ratio p.value
## gap / insitu_log                         0.849 0.190 Inf   1 -0.734  0.9777
## gap / insitu_pvc                        0.875 0.191 Inf   1 -0.614  0.9901
## gap / open                             1.184 0.254 Inf   1  0.789  0.9695
## gap / open_with_log                     1.112 0.268 Inf   1  0.439  0.9979
## gap / open_with_pvc                    0.835 0.175 Inf   1 -0.858  0.9563
## insitu_log / insitu_pvc                1.031 0.252 Inf   1  0.123  1.0000
## insitu_log / open                      1.396 0.341 Inf   1  1.362  0.7499
## insitu_log / open_with_log             1.310 0.353 Inf   1  1.003  0.9172
## insitu_log / open_with_pvc            0.984 0.231 Inf   1 -0.068  1.0000
## insitu_pvc / open                     1.354 0.323 Inf   1  1.271  0.8009
## insitu_pvc / open_with_log            1.271 0.336 Inf   1  0.907  0.9448
## insitu_pvc / open_with_pvc           0.955 0.220 Inf   1 -0.200  1.0000
## open / open_with_log                  0.939 0.244 Inf   1 -0.243  0.9999
## open / open_with_pvc                 0.705 0.164 Inf   1 -1.499  0.6650
## open_with_log / open_with_pvc        0.751 0.195 Inf   1 -1.102  0.8809
##
## name = ntrcy_tot:
## contrast                                ratio    SE  df null z.ratio p.value
## gap / insitu_log                         0.849 0.190 Inf   1 -0.734  0.9777
## gap / insitu_pvc                        0.875 0.191 Inf   1 -0.614  0.9901
```

```

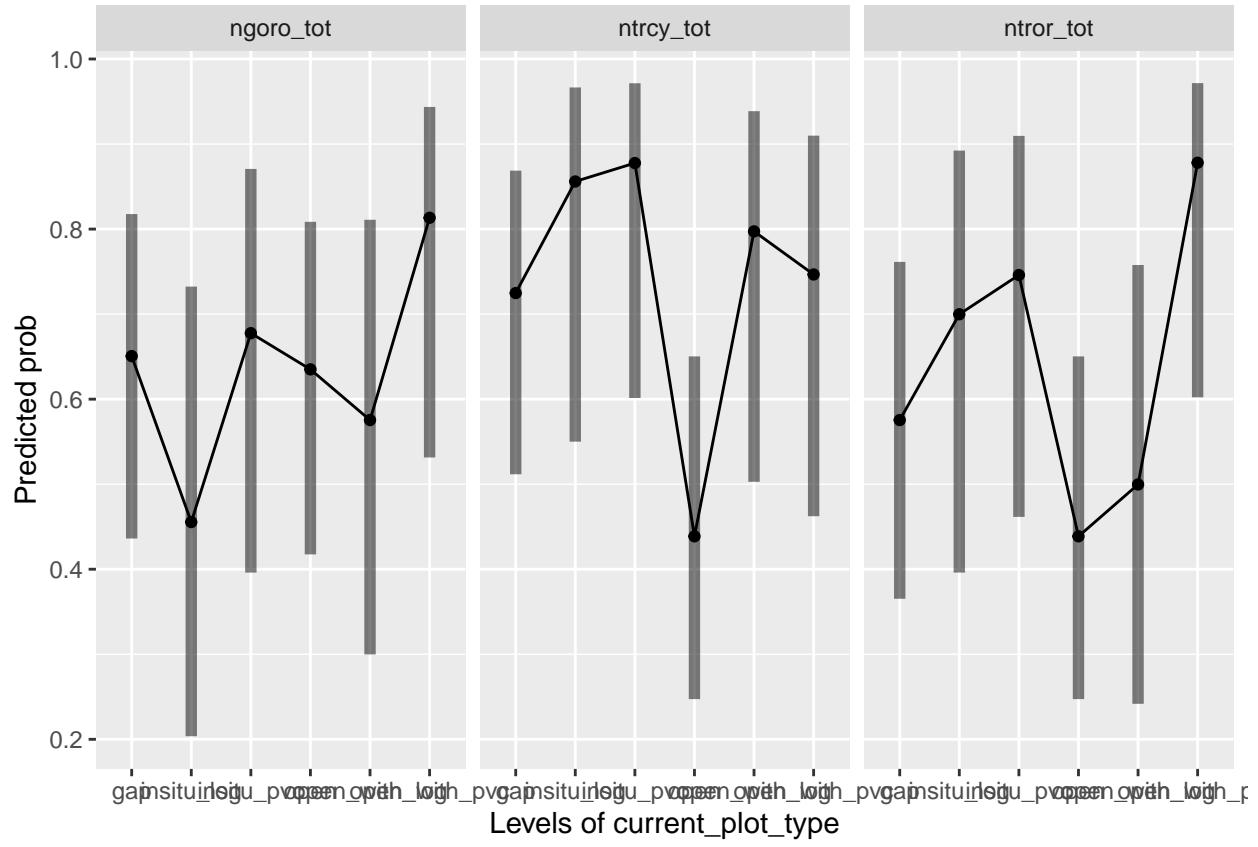
##  gap / open          1.184 0.254 Inf   1  0.789  0.9695
##  gap / open_with_log 1.112 0.268 Inf   1  0.439  0.9979
##  gap / open_with_pvc 0.835 0.175 Inf   1 -0.858  0.9563
##  insitu_log / insitu_pvc 1.031 0.252 Inf   1  0.123  1.0000
##  insitu_log / open    1.396 0.341 Inf   1  1.362  0.7499
##  insitu_log / open_with_log 1.310 0.353 Inf   1  1.003  0.9172
##  insitu_log / open_with_pvc 0.984 0.231 Inf   1 -0.068  1.0000
##  insitu_pvc / open    1.354 0.323 Inf   1  1.271  0.8009
##  insitu_pvc / open_with_log 1.271 0.336 Inf   1  0.907  0.9448
##  insitu_pvc / open_with_pvc 0.955 0.220 Inf   1 -0.200  1.0000
##  open / open_with_log 0.939 0.244 Inf   1 -0.243  0.9999
##  open / open_with_pvc 0.705 0.164 Inf   1 -1.499  0.6650
##  open_with_log / open_with_pvc 0.751 0.195 Inf   1 -1.102  0.8809
##
## name = ntror_tot:
## contrast           ratio   SE  df null z.ratio p.value
##  gap / insitu_log  0.849 0.190 Inf   1 -0.734  0.9777
##  gap / insitu_pvc  0.875 0.191 Inf   1 -0.614  0.9901
##  gap / open         1.184 0.254 Inf   1  0.789  0.9695
##  gap / open_with_log 1.112 0.268 Inf   1  0.439  0.9979
##  gap / open_with_pvc 0.835 0.175 Inf   1 -0.858  0.9563
##  insitu_log / insitu_pvc 1.031 0.252 Inf   1  0.123  1.0000
##  insitu_log / open    1.396 0.341 Inf   1  1.362  0.7499
##  insitu_log / open_with_log 1.310 0.353 Inf   1  1.003  0.9172
##  insitu_log / open_with_pvc 0.984 0.231 Inf   1 -0.068  1.0000
##  insitu_pvc / open    1.354 0.323 Inf   1  1.271  0.8009
##  insitu_pvc / open_with_log 1.271 0.336 Inf   1  0.907  0.9448
##  insitu_pvc / open_with_pvc 0.955 0.220 Inf   1 -0.200  1.0000
##  open / open_with_log 0.939 0.244 Inf   1 -0.243  0.9999
##  open / open_with_pvc 0.705 0.164 Inf   1 -1.499  0.6650
##  open_with_log / open_with_pvc 0.751 0.195 Inf   1 -1.102  0.8809
##
## P value adjustment: tukey method for comparing a family of 6 estimates
## Tests are performed on the log scale
```

```

##### now i'll do a by-hand hurdle model on my own using a truncated negative binomial

### zeros and ones

dat22$presence<-ifelse(dat22$value==0, 0, 1)
zerofit<-glmmTMB(presence~name*current_plot_type+(1 | block), family=binomial, data=dat22, REML=FALSE)
emmp(emzerofit, ~current_plot_type|name, type='response', CI=T)
```



```
est<-emmeans(zerofit, ~current_plot_type|name, type='response')
pairs(est)
```

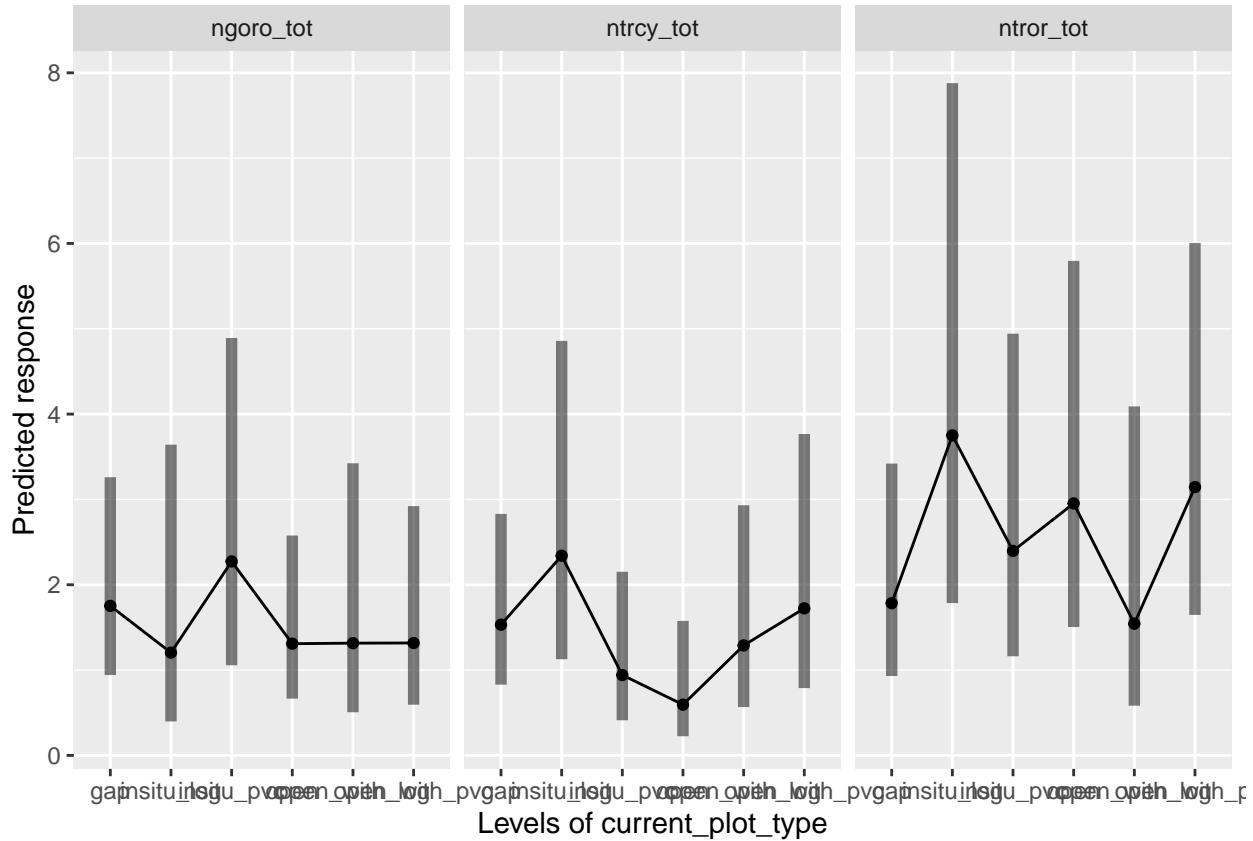
```
## name = ngoro_tot:
## contrast                               odds.ratio     SE   df null z.ratio p.value
## gap / insitu_log                      2.226 1.5662 Inf   1  1.138  0.8656
## gap / insitu_pvc                      0.886 0.6144 Inf   1 -0.174  1.0000
## gap / open                            1.070 0.6176 Inf   1  0.118  1.0000
## gap / open_with_log                   1.374 0.9460 Inf   1  0.462  0.9974
## gap / open_with_pvc                  0.427 0.3307 Inf   1 -1.099  0.8821
## insitu_log / insitu_pvc              0.398 0.3203 Inf   1 -1.145  0.8626
## insitu_log / open                     0.481 0.3394 Inf   1 -1.037  0.9054
## insitu_log / open_with_log           0.617 0.4935 Inf   1 -0.604  0.9908
## insitu_log / open_with_pvc          0.192 0.1680 Inf   1 -1.886  0.4107
## insitu_pvc / open                   1.208 0.8410 Inf   1  0.271  0.9998
## insitu_pvc / open_with_log          1.551 1.2269 Inf   1  0.555  0.9938
## insitu_pvc / open_with_pvc          0.482 0.4178 Inf   1 -0.842  0.9597
## open / open_with_log                1.284 0.8874 Inf   1  0.362  0.9992
## open / open_with_pvc               0.399 0.3101 Inf   1 -1.182  0.8456
## open_with_log / open_with_pvc      0.311 0.2683 Inf   1 -1.354  0.7546
##
## name = ntrcy_tot:
## contrast                               odds.ratio     SE   df null z.ratio p.value
## gap / insitu_log                      0.443 0.3956 Inf   1 -0.912  0.9437
## gap / insitu_pvc                      0.367 0.3239 Inf   1 -1.136  0.8666
```

```

##  gap / open          3.370 1.9788 Inf    1   2.069  0.3039
##  gap / open_with_log 0.669 0.5301 Inf    1  -0.507  0.9959
##  gap / open_with_pvc 0.894 0.6576 Inf    1  -0.153  1.0000
##  insitu_log / insitu_pvc 0.829 0.9107 Inf    1  -0.171  1.0000
##  insitu_log / open      7.606 6.6977 Inf    1   2.304  0.1923
##  insitu_log / open_with_log 1.511 1.5526 Inf    1   0.402  0.9987
##  insitu_log / open_with_pvc 2.017 1.9871 Inf    1   0.712  0.9805
##  insitu_pvc / open      9.178 7.9857 Inf    1   2.548  0.1105
##  insitu_pvc / open_with_log 1.823 1.8568 Inf    1   0.590  0.9917
##  insitu_pvc / open_with_pvc 2.434 2.3749 Inf    1   0.912  0.9436
##  open / open_with_log   0.199 0.1545 Inf    1  -2.079  0.2984
##  open / open_with_pvc   0.265 0.1911 Inf    1  -1.842  0.4385
##  open_with_log / open_with_pvc 1.335 1.1941 Inf    1   0.323  0.9995
##
## name = ntror_tot:
## contrast           odds.ratio   SE  df null z.ratio p.value
##  gap / insitu_log   0.581 0.4259 Inf    1  -0.740  0.9769
##  gap / insitu_pvc   0.461 0.3304 Inf    1  -1.080  0.8895
##  gap / open          1.734 0.9721 Inf    1   0.982  0.9239
##  gap / open_with_log 1.357 0.9183 Inf    1   0.451  0.9977
##  gap / open_with_pvc 0.188 0.1631 Inf    1  -1.927  0.3852
##  insitu_log / insitu_pvc 0.794 0.6831 Inf    1  -0.268  0.9998
##  insitu_log / open     2.983 2.1972 Inf    1   1.484  0.6746
##  insitu_log / open_with_log 2.335 1.9333 Inf    1   1.024  0.9101
##  insitu_log / open_with_pvc 0.324 0.3201 Inf    1  -1.141  0.8644
##  insitu_pvc / open     3.757 2.7057 Inf    1   1.838  0.4409
##  insitu_pvc / open_with_log 2.940 2.3920 Inf    1   1.326  0.7707
##  insitu_pvc / open_with_pvc 0.408 0.3980 Inf    1  -0.919  0.9418
##  open / open_with_log   0.783 0.5322 Inf    1  -0.360  0.9992
##  open / open_with_pvc   0.109 0.0945 Inf    1  -2.551  0.1097
##  open_with_log / open_with_pvc 0.139 0.1316 Inf    1  -2.082  0.2967
##
## P value adjustment: tukey method for comparing a family of 6 estimates
## Tests are performed on the log odds ratio scale
```

```
# no significant differences
```

```
### abundance with a truncated negbinom
dat22$posicounts<-as.numeric(ifelse(dat22$value==0, "NA", dat22$value))
countfit<-glmmTMB(posicounts~name*current_plot_type+(1 | block), family=truncated_nbinom2(), data=dat22)
emmip(countfit, ~current_plot_type|name, type='response', CI=T)
```



```
est<-emmeans(countfit, ~current_plot_type|name, type='response')
pairs(est)
```

```
## name = ngoro_tot:
## contrast                                ratio    SE  df null z.ratio p.value
## gap / insitu_log                         1.455 0.901 Inf   1  0.606  0.9906
## gap / insitu_pvc                        0.771 0.364 Inf   1 -0.551  0.9940
## gap / open                             1.339 0.575 Inf   1  0.679  0.9843
## gap / open_with_log                     1.332 0.734 Inf   1  0.520  0.9954
## gap / open_with_pvc                     1.330 0.641 Inf   1  0.592  0.9916
## insitu_log / insitu_pvc                0.530 0.351 Inf   1 -0.960  0.9305
## insitu_log / open                       0.920 0.582 Inf   1 -0.132  1.0000
## insitu_log / open_with_log              0.915 0.658 Inf   1 -0.123  1.0000
## insitu_log / open_with_pvc              0.914 0.610 Inf   1 -0.135  1.0000
## insitu_pvc / open                      1.737 0.853 Inf   1  1.124  0.8718
## insitu_pvc / open_with_log             1.728 1.032 Inf   1  0.916  0.9425
## insitu_pvc / open_with_pvc             1.726 0.919 Inf   1  1.024  0.9100
## open / open_with_log                   0.995 0.565 Inf   1 -0.009  1.0000
## open / open_with_pvc                  0.994 0.497 Inf   1 -0.013  1.0000
## open_with_log / open_with_pvc          0.999 0.601 Inf   1 -0.002  1.0000
##
## name = ntrcy_tot:
## contrast                                ratio    SE  df null z.ratio p.value
## gap / insitu_log                         0.655 0.298 Inf   1 -0.932  0.9383
## gap / insitu_pvc                        1.627 0.794 Inf   1  0.998  0.9188
```

```

##  gap / open           2.575 1.419 Inf    1   1.717  0.5204
##  gap / open_with_log 1.189 0.579 Inf    1   0.355  0.9993
##  gap / open_with_pvc 0.889 0.420 Inf    1  -0.249  0.9999
##  insitu_log / insitu_pvc 2.485 1.322 Inf    1   1.712  0.5238
##  insitu_log / open     3.934 2.325 Inf    1   2.318  0.1868
##  insitu_log / open_with_log 1.815 0.966 Inf    1   1.121  0.8730
##  insitu_log / open_with_pvc 1.358 0.702 Inf    1   0.592  0.9916
##  insitu_pvc / open     1.583 0.972 Inf    1   0.748  0.9758
##  insitu_pvc / open_with_log 0.730 0.409 Inf    1  -0.561  0.9935
##  insitu_pvc / open_with_pvc 0.546 0.299 Inf    1  -1.103  0.8804
##  open / open_with_log   0.461 0.284 Inf    1  -1.256  0.8087
##  open / open_with_pvc   0.345 0.209 Inf    1  -1.755  0.4953
##  open_with_log / open_with_pvc 0.748 0.412 Inf    1  -0.528  0.9951
##
## name = ntror_tot:
## contrast                  ratio   SE df null z.ratio p.value
## gap / insitu_log          0.476 0.228 Inf    1  -1.552  0.6302
## gap / insitu_pvc          0.745 0.347 Inf    1  -0.633  0.9886
## gap / open                 0.604 0.272 Inf    1  -1.121  0.8730
## gap / open_with_log       1.155 0.656 Inf    1   0.254  0.9999
## gap / open_with_pvc       0.567 0.248 Inf    1  -1.296  0.7875
## insitu_log / insitu_pvc   1.565 0.792 Inf    1   0.885  0.9502
## insitu_log / open          1.270 0.625 Inf    1   0.486  0.9967
## insitu_log / open_with_log 2.429 1.470 Inf    1   1.466  0.6858
## insitu_log / open_with_pvc 1.193 0.574 Inf    1   0.366  0.9991
## insitu_pvc / open          0.811 0.390 Inf    1  -0.435  0.9980
## insitu_pvc / open_with_log 1.552 0.920 Inf    1   0.741  0.9768
## insitu_pvc / open_with_pvc 0.762 0.357 Inf    1  -0.580  0.9923
## open / open_with_log       1.913 1.108 Inf    1   1.120  0.8735
## open / open_with_pvc       0.939 0.423 Inf    1  -0.139  1.0000
## open_with_log / open_with_pvc 0.491 0.278 Inf    1  -1.256  0.8091
##
## P value adjustment: tukey method for comparing a family of 6 estimates
## Tests are performed on the log scale

```

```
# no significant differences in abundance
```

```
##### treatment response: do analysis for total biomass by species #####
```

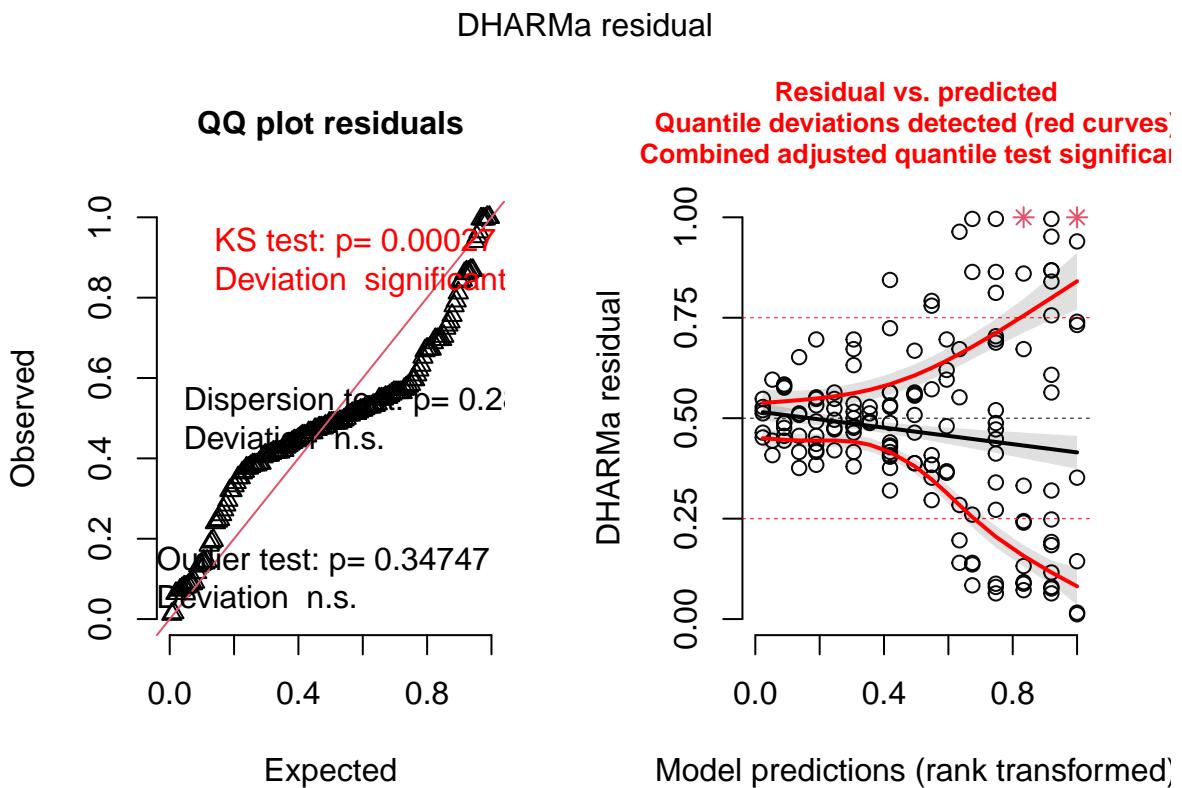
```

dat3<-(dat1[c(1:4,19,21,23,25)])
totwtdat<-as.data.frame(dat3 %>% pivot_longer(c(wt_max15_goro, wt_max15_tror, wt_max15_trcy)))
totwtdat$log_wt<-log1p(totwtdat$value) # log transformation does not help the insane amount of heterosc

# model
totwtmod<-lm(log_wt~name*current_plot_type, data=totwtdat) # fit is singular when including a random ef

# test for fit
sim<-simulateResiduals(totwtmod) # insanely heteroscedastic, doesn't get better with sqrt transform (al
plot(sim)

```



```

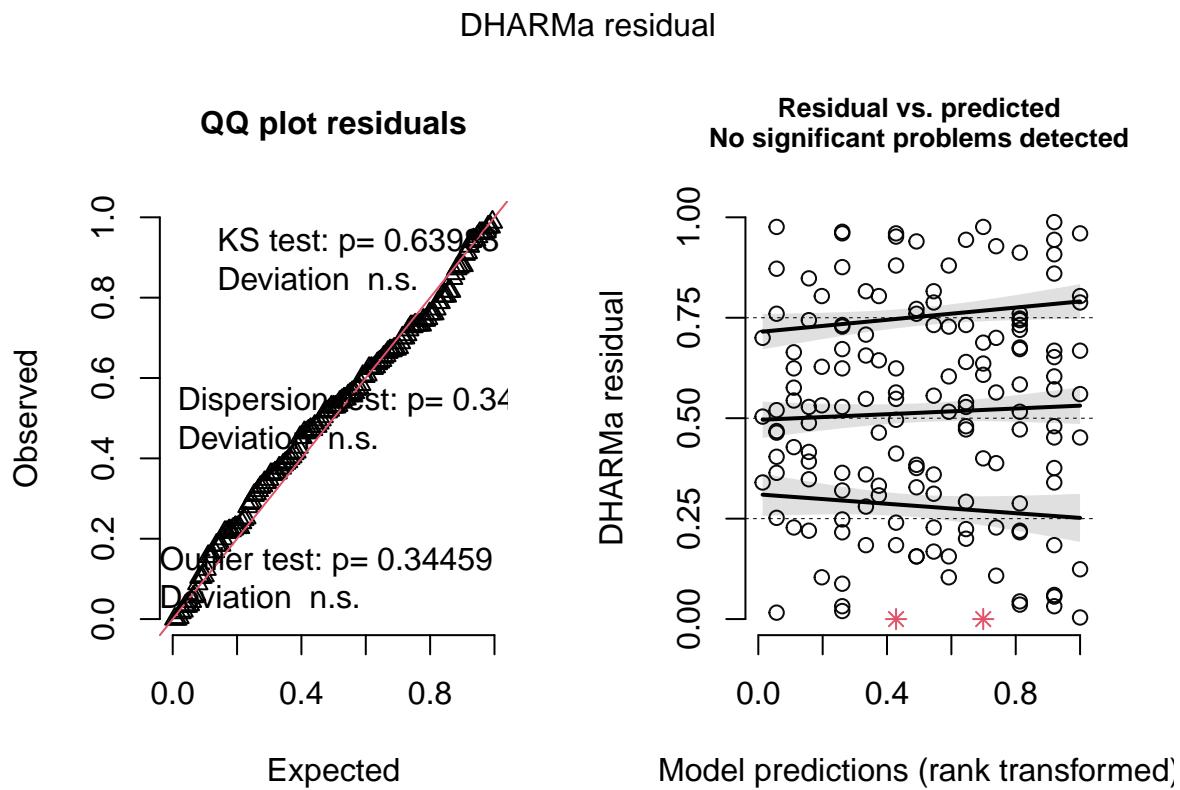
# model summary - don't trust this the residual dispersion is fucked, commenting out.
# summary(totwtmod)
# emmip(totwtmod,~current_plot_type/name, CI=T)
#
# est<-emmeans(totwtmod,~current_plot_type/name, type='response')
# pairs(est)

##### treatment response: do analysis for per capita biomass by species #####
# per capita might help with heteroscedasticity
dat4<-(dat1[c(1:4,19,20,22,24)])
pcwtdat<-as.data.frame(dat4 %>% pivot_longer(c(wt_percapita_goro, wt_percapita_tror, wt_percapita_trcy))
pcwtdat$log_wt<-log(pcwtdat$value) # log transform the weight data to get it normal looking - it is het

# model
pcwtmod<-lm(log_wt~name*current_plot_type, data=pcwtdat) # fit is singular with random effect.

# test for fit, looks pretty good
sim<-simulateResiduals(pcwtmod)
plot(sim) ## much better!

```



```
# model summary
summary(pcwtmod)
```

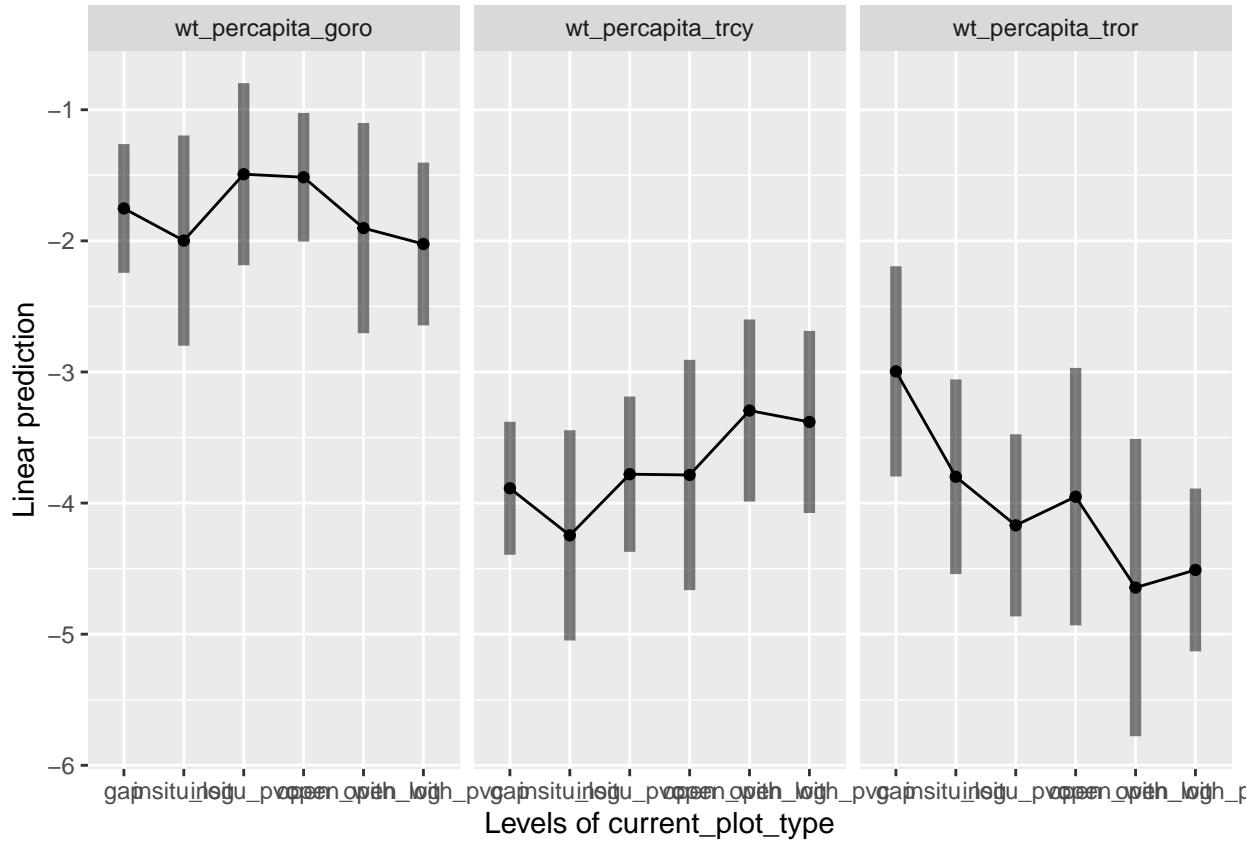
```
##
## Call:
## lm(formula = log_wt ~ name * current_plot_type, data = pcwtdat)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.78438 -0.55882  0.03892  0.53364  2.07151
##
## Coefficients:
## (Intercept)                               Estimate Std. Error
## namewt_per capita_trcy                  -1.7528   0.2482
## namewt_per capita_tror                  -2.1342   0.3569
## current_plot_typeinsitu_log             -1.2428   0.4754
## current_plot_typeinsitu_pvc              -0.2453   0.4754
## current_plot_typeopen                  0.2613   0.4300
## current_plot_typeopen_with_log          0.2378   0.3511
## current_plot_typeopen_with_pvc          -0.1498   0.4754
## namewt_per capita_trcy:current_plot_typeinsitu_log -0.2712   0.4003
## namewt_per capita_trcy:current_plot_typeinsitu_log -0.1139   0.6753
## namewt_per capita_tror:current_plot_typeinsitu_log -0.5584   0.7288
## namewt_per capita_trcy:current_plot_typeinsitu_pvc -0.1540   0.5833
## namewt_per capita_tror:current_plot_typeinsitu_pvc -1.4357   0.6874
## namewt_per capita_trcy:current_plot_typeopen     -0.1365   0.6214
```

```

## namewt_per capita_tror:current_plot_typeopen      -1.1933  0.7308
## namewt_per capita_trcy:current_plot_typeopen_with_log  0.7427  0.6442
## namewt_per capita_tror:current_plot_typeopen_with_log -1.4992  0.8479
## namewt_per capita_trcy:current_plot_typeopen_with_pvc  0.7768  0.5909
## namewt_per capita_tror:current_plot_typeopen_with_pvc -1.2434  0.6505
##
## t value Pr(>|t|)
## (Intercept)          -7.061 7.92e-11 ***
## namewt_per capita_trcy -5.980 1.88e-08 ***
## namewt_per capita_tror -2.614  0.00995 **
## current_plot_typeinsitu_log -0.516  0.60664
## current_plot_typeinsitu_pvc   0.608  0.54437
## current_plot_typeopen        0.677  0.49929
## current_plot_typeopen_with_log -0.315  0.75309
## current_plot_typeopen_with_pvc -0.677  0.49926
## namewt_per capita_trcy:current_plot_typeinsitu_log -0.169  0.86632
## namewt_per capita_tror:current_plot_typeinsitu_log -0.766  0.44488
## namewt_per capita_trcy:current_plot_typeinsitu_pvc -0.264  0.79211
## namewt_per capita_tror:current_plot_typeinsitu_pvc -2.089  0.03862 *
## namewt_per capita_trcy:current_plot_typeopen       -0.220  0.82646
## namewt_per capita_tror:current_plot_typeopen       -1.633  0.10484
## namewt_per capita_trcy:current_plot_typeopen_with_log  1.153  0.25096
## namewt_per capita_tror:current_plot_typeopen_with_log -1.768  0.07930 .
## namewt_per capita_trcy:current_plot_typeopen_with_pvc  1.315  0.19088
## namewt_per capita_tror:current_plot_typeopen_with_pvc -1.911  0.05808 .
##
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.993 on 135 degrees of freedom
##   (183 observations deleted due to missingness)
## Multiple R-squared:  0.5801, Adjusted R-squared:  0.5272
## F-statistic: 10.97 on 17 and 135 DF,  p-value: < 2.2e-16

```

```
emmap(pcwtmod, ~current_plot_type | name, CI=T)
```



```
est<-emmeans(totwtmod, ~current_plot_type|name, type='response')
pairs(est)
```

## name = wt_max15_goro:	estimate	SE	df	t.ratio	p.value
## contrast	0.06531	0.0967	136	0.675	0.9844
## gap - insitu_log	-0.17108	0.0875	136	-1.955	0.3737
## gap - insitu_pvc	-0.05976	0.0714	136	-0.836	0.9602
## gap - open	0.03726	0.0967	136	0.385	0.9989
## gap - open_with_log	-0.00687	0.0815	136	-0.084	1.0000
## insitu_log - insitu_pvc	-0.23640	0.1091	136	-2.166	0.2606
## insitu_log - open	-0.12508	0.0967	136	-1.293	0.7885
## insitu_log - open_with_log	-0.02805	0.1167	136	-0.240	0.9999
## insitu_log - open_with_pvc	-0.07218	0.1044	136	-0.692	0.9826
## insitu_pvc - open	0.11132	0.0875	136	1.272	0.7996
## insitu_pvc - open_with_log	0.20835	0.1091	136	1.909	0.4012
## insitu_pvc - open_with_pvc	0.16422	0.0959	136	1.713	0.5256
## open - open_with_log	0.09703	0.0967	136	1.003	0.9163
## open - open_with_pvc	0.05290	0.0815	136	0.649	0.9869
## open_with_log - open_with_pvc	-0.04413	0.1044	136	-0.423	0.9982
##					
## name = wt_max15_trcy:	estimate	SE	df	t.ratio	p.value
## contrast	0.04197	0.0976	136	0.430	0.9981
## gap - insitu_log	0.01608	0.0802	136	0.200	1.0000

```

## gap - open          0.03084 0.0976 136  0.316  0.9996
## gap - open_with_log -0.01903 0.0885 136 -0.215  0.9999
## gap - open_with_pvc -0.04116 0.0885 136 -0.465  0.9972
## insitu_log - insitu_pvc -0.02589 0.1026 136 -0.252  0.9999
## insitu_log - open   -0.01113 0.1167 136 -0.095  1.0000
## insitu_log - open_with_log -0.06100 0.1091 136 -0.559  0.9934
## insitu_log - open_with_pvc -0.08313 0.1091 136 -0.762  0.9734
## insitu_pvc - open   0.01476 0.1026 136  0.144  1.0000
## insitu_pvc - open_with_log -0.03511 0.0939 136 -0.374  0.9990
## insitu_pvc - open_with_pvc -0.05724 0.0939 136 -0.610  0.9902
## open - open_with_log -0.04987 0.1091 136 -0.457  0.9975
## open - open_with_pvc -0.07200 0.1091 136 -0.660  0.9859
## open_with_log - open_with_pvc -0.02213 0.1010 136 -0.219  0.9999
##
## name = wt_max15_tror:
## contrast           estimate    SE  df t.ratio p.value
## gap - insitu_log   0.06360 0.1124 136  0.566  0.9930
## gap - insitu_pvc   0.07855 0.1091 136  0.720  0.9793
## gap - open          0.05806 0.1304 136  0.445  0.9978
## gap - open_with_log 0.08039 0.1429 136  0.563  0.9932
## gap - open_with_pvc 0.06465 0.1044 136  0.620  0.9894
## insitu_log - insitu_pvc 0.01495 0.1046 136  0.143  1.0000
## insitu_log - open   -0.00554 0.1267 136 -0.044  1.0000
## insitu_log - open_with_log 0.01679 0.1395 136  0.120  1.0000
## insitu_log - open_with_pvc 0.00105 0.0996 136  0.011  1.0000
## insitu_pvc - open   -0.02049 0.1238 136 -0.166  1.0000
## insitu_pvc - open_with_log 0.00184 0.1368 136  0.013  1.0000
## insitu_pvc - open_with_pvc -0.01390 0.0959 136 -0.145  1.0000
## open - open_with_log 0.02233 0.1543 136  0.145  1.0000
## open - open_with_pvc 0.00659 0.1196 136  0.055  1.0000
## open_with_log - open_with_pvc -0.01574 0.1330 136 -0.118  1.0000
##
## P value adjustment: tukey method for comparing a family of 6 estimates

est2<-emmeans(totwtmod, ~name, type='response')
pairs(est)

## name = wt_max15_goro:
## contrast           estimate    SE  df t.ratio p.value
## gap - insitu_log   0.06531 0.0967 136  0.675  0.9844
## gap - insitu_pvc   -0.17108 0.0875 136 -1.955  0.3737
## gap - open          -0.05976 0.0714 136 -0.836  0.9602
## gap - open_with_log 0.03726 0.0967 136  0.385  0.9989
## gap - open_with_pvc -0.00687 0.0815 136 -0.084  1.0000
## insitu_log - insitu_pvc -0.23640 0.1091 136 -2.166  0.2606
## insitu_log - open   -0.12508 0.0967 136 -1.293  0.7885
## insitu_log - open_with_log -0.02805 0.1167 136 -0.240  0.9999
## insitu_log - open_with_pvc -0.07218 0.1044 136 -0.692  0.9826
## insitu_pvc - open   0.11132 0.0875 136  1.272  0.7996
## insitu_pvc - open_with_log 0.20835 0.1091 136  1.909  0.4012
## insitu_pvc - open_with_pvc 0.16422 0.0959 136  1.713  0.5256
## open - open_with_log 0.09703 0.0967 136  1.003  0.9163
## open - open_with_pvc 0.05290 0.0815 136  0.649  0.9869
## open_with_log - open_with_pvc -0.04413 0.1044 136 -0.423  0.9982

```

```

##  

## name = wt_max15_trcy:  

## contrast estimate SE df t.ratio p.value  

## gap - insitu_log 0.04197 0.0976 136 0.430 0.9981  

## gap - insitu_pvc 0.01608 0.0802 136 0.200 1.0000  

## gap - open 0.03084 0.0976 136 0.316 0.9996  

## gap - open_with_log -0.01903 0.0885 136 -0.215 0.9999  

## gap - open_with_pvc -0.04116 0.0885 136 -0.465 0.9972  

## insitu_log - insitu_pvc -0.02589 0.1026 136 -0.252 0.9999  

## insitu_log - open -0.01113 0.1167 136 -0.095 1.0000  

## insitu_log - open_with_log -0.06100 0.1091 136 -0.559 0.9934  

## insitu_log - open_with_pvc -0.08313 0.1091 136 -0.762 0.9734  

## insitu_pvc - open 0.01476 0.1026 136 0.144 1.0000  

## insitu_pvc - open_with_log -0.03511 0.0939 136 -0.374 0.9990  

## insitu_pvc - open_with_pvc -0.05724 0.0939 136 -0.610 0.9902  

## open - open_with_log -0.04987 0.1091 136 -0.457 0.9975  

## open - open_with_pvc -0.07200 0.1091 136 -0.660 0.9859  

## open_with_log - open_with_pvc -0.02213 0.1010 136 -0.219 0.9999  

##  

## name = wt_max15_tror:  

## contrast estimate SE df t.ratio p.value  

## gap - insitu_log 0.06360 0.1124 136 0.566 0.9930  

## gap - insitu_pvc 0.07855 0.1091 136 0.720 0.9793  

## gap - open 0.05806 0.1304 136 0.445 0.9978  

## gap - open_with_log 0.08039 0.1429 136 0.563 0.9932  

## gap - open_with_pvc 0.06465 0.1044 136 0.620 0.9894  

## insitu_log - insitu_pvc 0.01495 0.1046 136 0.143 1.0000  

## insitu_log - open -0.00554 0.1267 136 -0.044 1.0000  

## insitu_log - open_with_log 0.01679 0.1395 136 0.120 1.0000  

## insitu_log - open_with_pvc 0.00105 0.0996 136 0.011 1.0000  

## insitu_pvc - open -0.02049 0.1238 136 -0.166 1.0000  

## insitu_pvc - open_with_log 0.00184 0.1368 136 0.013 1.0000  

## insitu_pvc - open_with_pvc -0.01390 0.0959 136 -0.145 1.0000  

## open - open_with_log 0.02233 0.1543 136 0.145 1.0000  

## open - open_with_pvc 0.00659 0.1196 136 0.055 1.0000  

## open_with_log - open_with_pvc -0.01574 0.1330 136 -0.118 1.0000  

##  

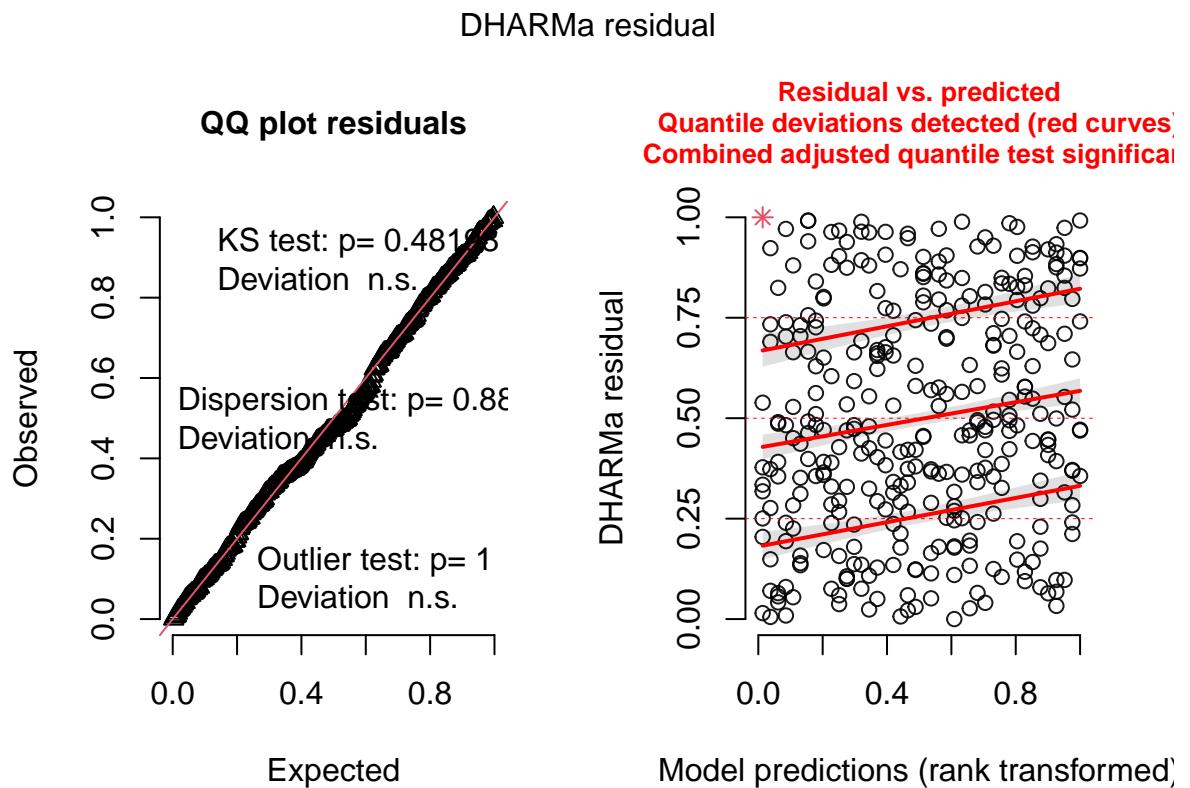
## P value adjustment: tukey method for comparing a family of 6 estimates

# not any differences in per capita biomass among treatments within species, but goro is larger than the others

##### PHYSICAL BARRIER ANALYSIS #####
##### What about the physical barrier - initial treatment for 2022 #####
##### This is a model where do zero inflation and count together (occurrence and abundance)

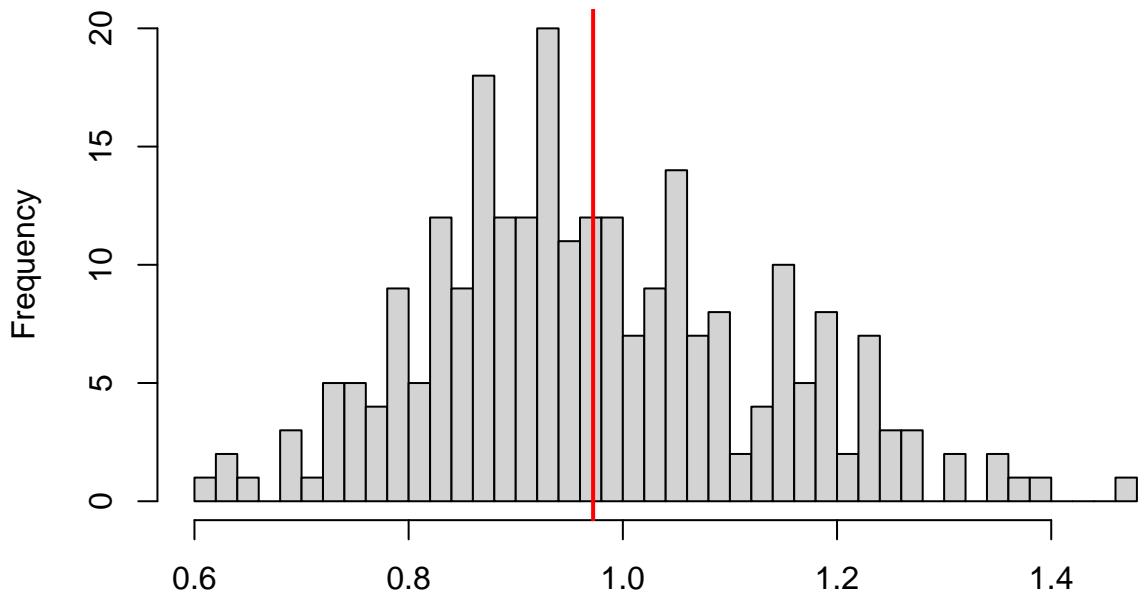
fit_phys<-glmmTMB(value~name*physical_barrier+(1 | block), ziformula=~., family=nbinom2(), data=dat22, link=log)
sim<-simulateResiduals(fit_phys)
plot(sim)

```



```
testDispersion(sim)
```

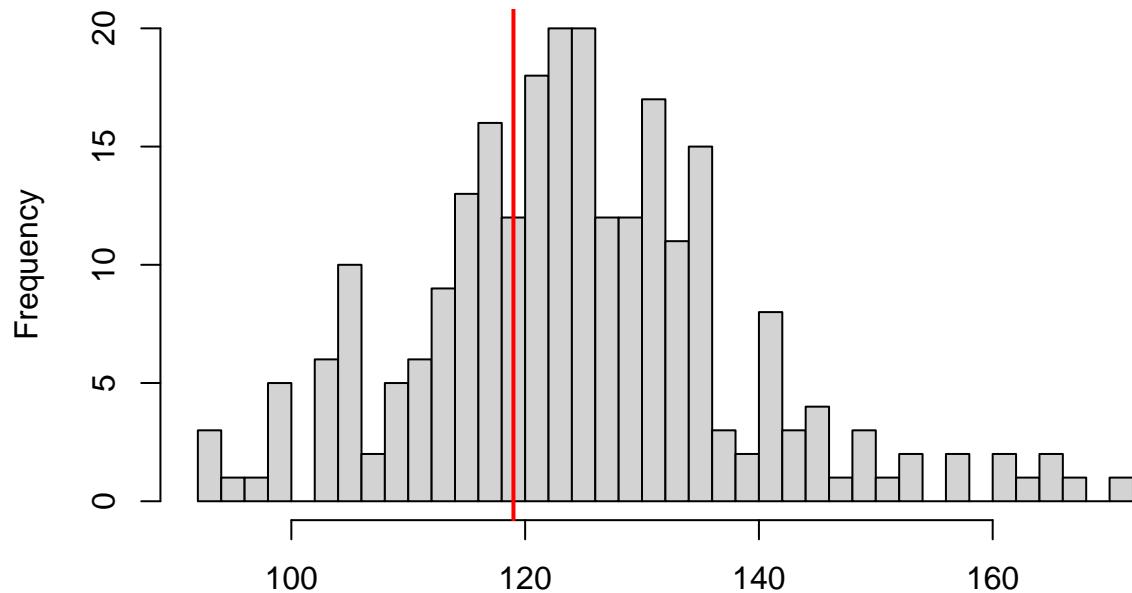
DHARMA nonparametric dispersion test via sd of residuals fitted vs. simulated



Simulated values, red line = fitted model. p-value (two.sided) = 0.888

```
##  
##  DHARMA nonparametric dispersion test via sd of residuals fitted vs.  
##  simulated  
##  
##  data:  simulationOutput  
##  dispersion = 0.99867, p-value = 0.888  
##  alternative hypothesis: two.sided  
  
testZeroInflation(sim)
```

DHARMA zero-inflation test via comparison to expected zeros with simulation under H0 = fitted model



Simulated values, red line = fitted model. p-value (two.sided) = 0.664

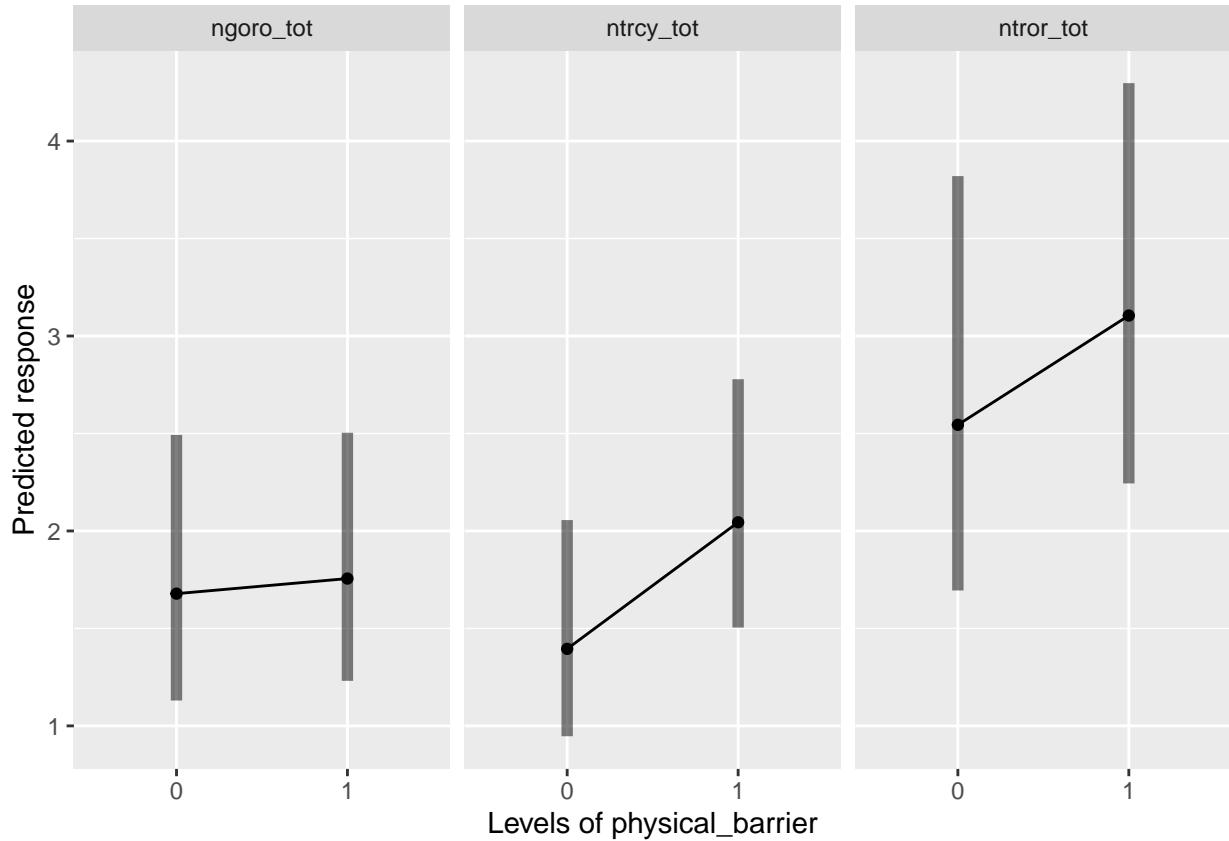
```
##  
## DHARMA zero-inflation test via comparison to expected zeros with  
## simulation under H0 = fitted model  
##  
## data: simulationOutput  
## ratioObsSim = 0.95112, p-value = 0.664  
## alternative hypothesis: two.sided  
  
summary(fit_phys)  
  
## Family: nbinom2  ( log )  
## Formula:           value ~ name * physical_barrier + (1 | block)  
## Zero inflation:    ~.  
## Data: dat22  
##  
##      AIC      BIC  logLik deviance df.resid  
##  1229.9   1287.2   -600.0    1199.9      321  
##  
## Random effects:  
##  
## Conditional model:  
## Groups Name        Variance Std.Dev.  
## block  (Intercept) 0.00587  0.07662  
## Number of obs: 336, groups: block, 7  
##
```

```

## Zero-inflation model:
## Groups Name      Variance Std.Dev.
## block  (Intercept) 1.278     1.131
## Number of obs: 336, groups: block, 7
##
## Dispersion parameter for nbinom2 family (): 1.59
##
## Conditional model:
##                               Estimate Std. Error z value Pr(>|z|)
## (Intercept)                  0.51782   0.20178  2.566   0.0103 *
## namentrcy_tot                -0.18481   0.27397 -0.675   0.4999
## namentror_tot                 0.41609   0.26653  1.561   0.1185
## physical_barrier1            0.04502   0.25651  0.176   0.8607
## namentrcy_tot:physical_barrier1 0.33708   0.34860  0.967   0.3336
## namentror_tot:physical_barrier1 0.15410   0.35400  0.435   0.6633
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Zero-inflation model:
##                               Estimate Std. Error z value Pr(>|z|)
## (Intercept)                 -3.4280   2.8635 -1.197   0.231
## namentrcy_tot                 0.8232   2.8146  0.292   0.770
## namentror_tot                 2.6663   2.6805  0.995   0.320
## physical_barrier1              0.7603   2.6214  0.290   0.772
## namentrcy_tot:physical_barrier1 -2.9115   4.4282 -0.657   0.511
## namentror_tot:physical_barrier1 -2.1069   2.7842 -0.757   0.449

emmap(fit_phys, ~physical_barrier1|name, type='response', CI=T)

```



```
est<-emmeans(fit_phys, ~physical_barrier|name, type='response')
pairs(est)
```

```
## name = ngoro_tot:
##   contrast           ratio      SE  df null z.ratio p.value
##   physical_barrier0 / physical_barrier1 0.956 0.245 Inf     1 -0.176  0.8607
##
## name = ntrcy_tot:
##   contrast           ratio      SE  df null z.ratio p.value
##   physical_barrier0 / physical_barrier1 0.682 0.168 Inf     1 -1.556  0.1198
##
## name = ntror_tot:
##   contrast           ratio      SE  df null z.ratio p.value
##   physical_barrier0 / physical_barrier1 0.819 0.205 Inf     1 -0.795  0.4264
##
## Tests are performed on the log scale
```

```
# no significant differences in total plants (occurrence and abundance)
```

```
# split up occurrence and abundance
```

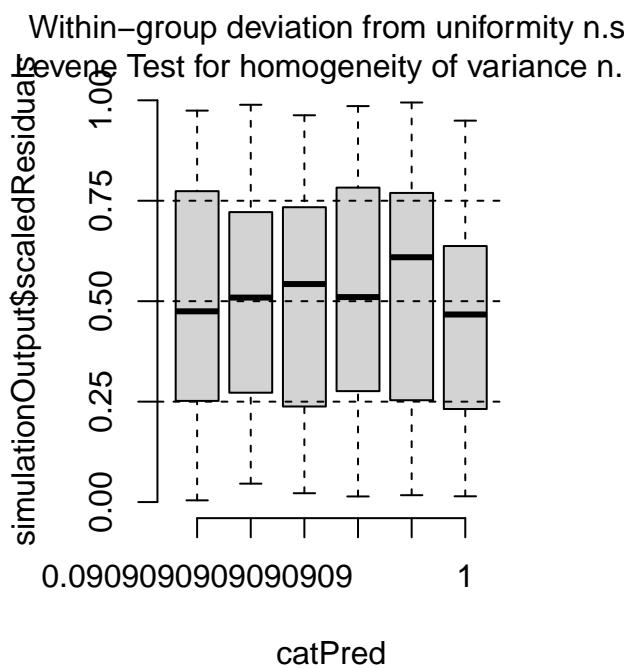
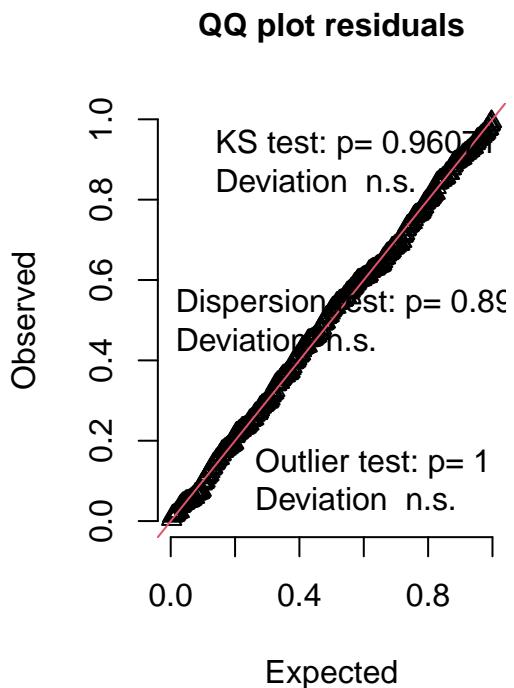
```
#### zeros and ones
```

```
zerofit_phys<-glmmTMB(presence~name*physical_barrier+(1 | block), family=binomial, data=dat22, REML=F)
```

```
sim<-simulateResiduals(zerofit_phys)
```

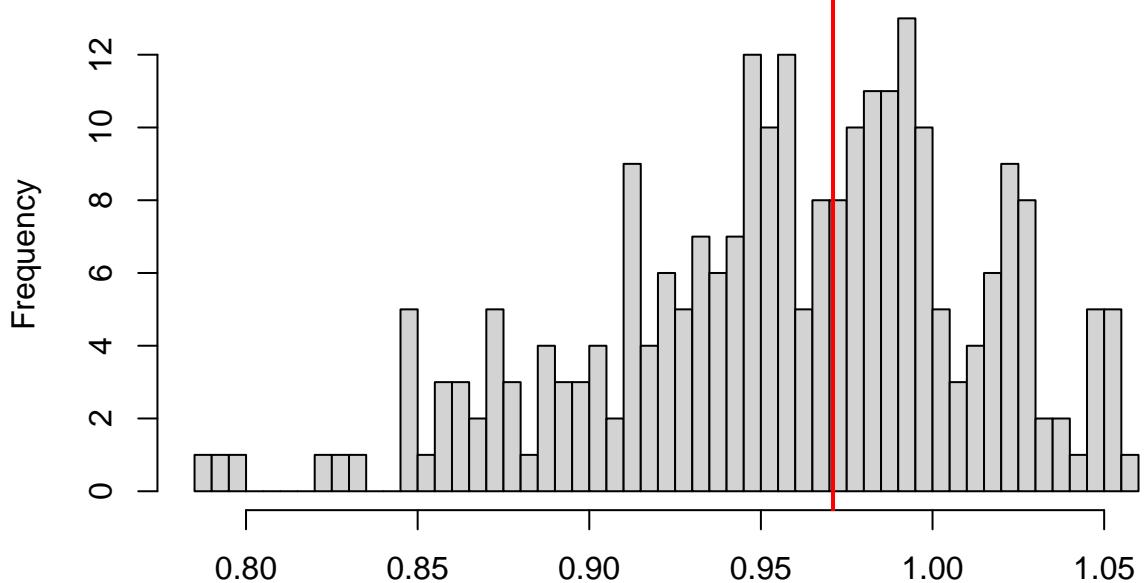
```
plot(sim)
```

DHARMA residual



```
testDispersion(sim) # looks good
```

DHARMA nonparametric dispersion test via sd of residuals fitted vs. simulated

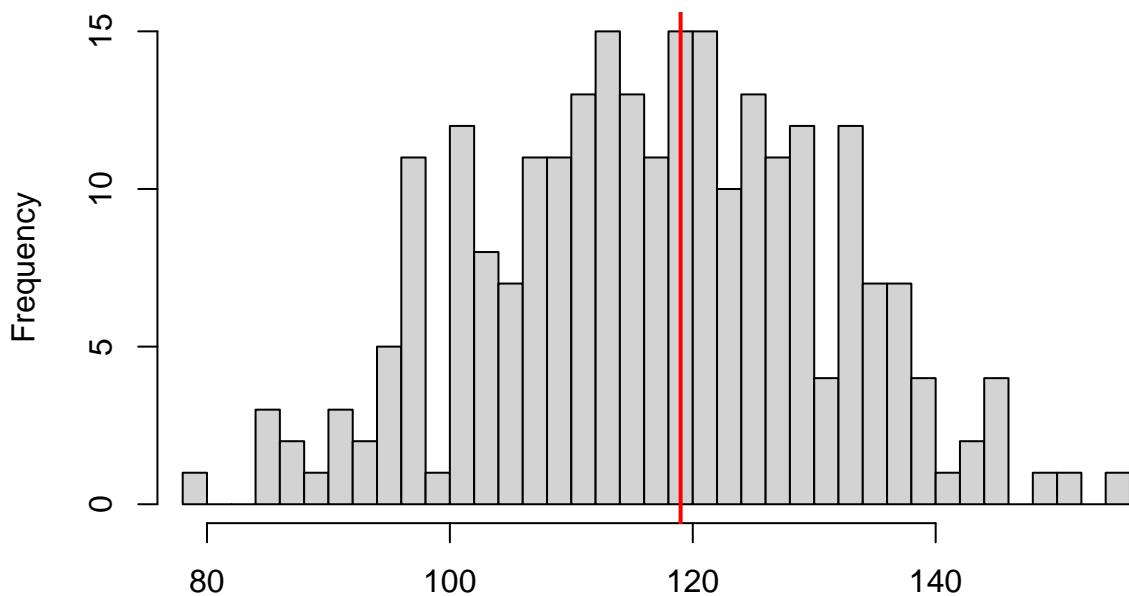


Simulated values, red line = fitted model. p-value (two.sided) = 0.896

```
##  
## DHARMA nonparametric dispersion test via sd of residuals fitted vs.  
## simulated  
##  
## data: simulationOutput  
## dispersion = 1.0146, p-value = 0.896  
## alternative hypothesis: two.sided
```

```
testZeroInflation(sim) # looks good
```

DHARMA zero-inflation test via comparison to expected zeros with simulation under H0 = fitted model



Simulated values, red line = fitted model. p-value (two.sided) = 0.96

```
##
## DHARMA zero-inflation test via comparison to expected zeros with
## simulation under H0 = fitted model
##
## data: simulationOutput
## ratioObsSim = 1.0141, p-value = 0.96
## alternative hypothesis: two.sided

summary(zeroFit_phys)

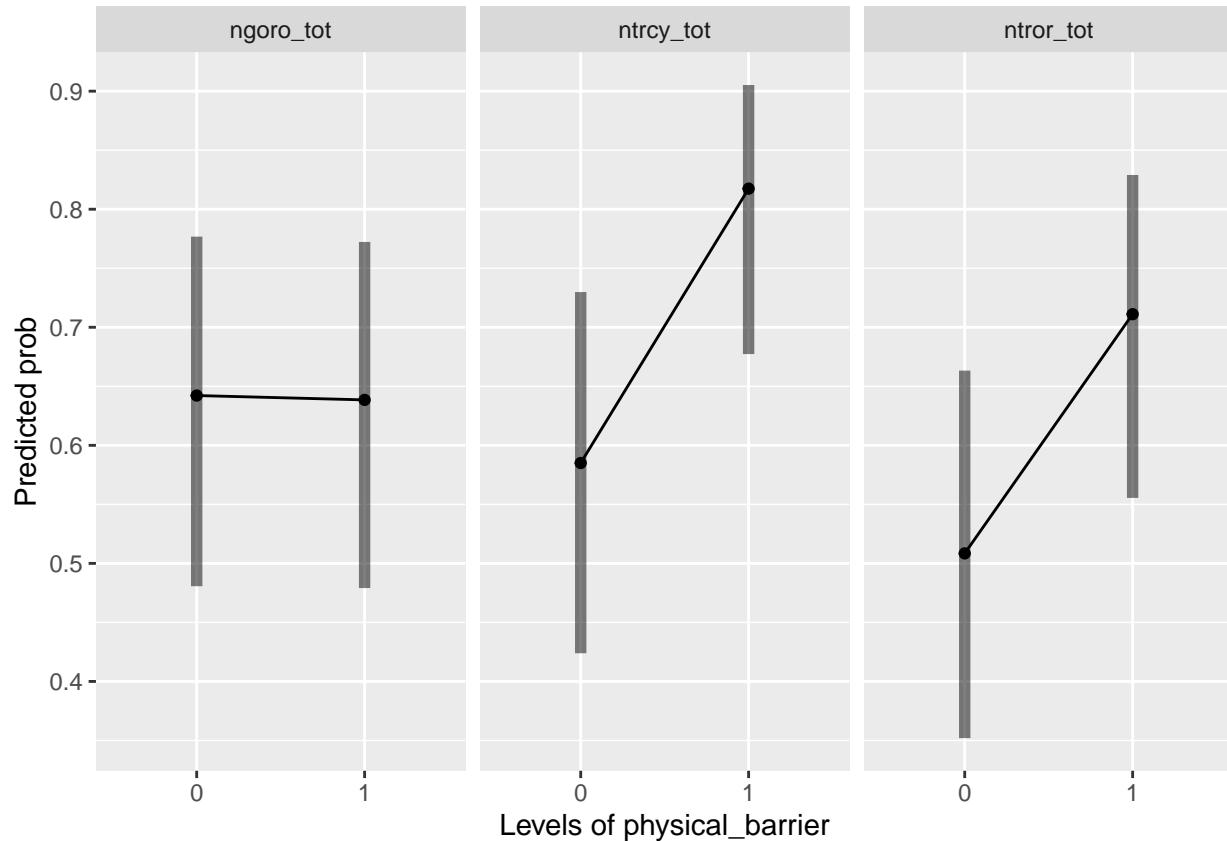
## Family: binomial ( logit )
## Formula: presence ~ name * physical_barrier + (1 | block)
## Data: dat22
##
##      AIC      BIC   logLik deviance df.resid
##      430.3    457.0   -208.1     416.3      329
##
## Random effects:
##
## Conditional model:
## Groups Name        Variance Std.Dev.
## block  (Intercept) 0.2193   0.4683
## Number of obs: 336, groups: block, 7
##
## Conditional model:
```

```

##                               Estimate Std. Error z value Pr(>|z|)
## (Intercept)                  0.58477   0.33785   1.731   0.0835 .
## namentrcy_tot                -0.24130   0.40162  -0.601   0.5480
## namentror_tot                -0.55089   0.39934  -1.380   0.1677
## physical_barrier1             -0.01595   0.40226  -0.040   0.9684
## namentrcy_tot:physical_barrier1 1.17142   0.59819   1.958   0.0502 .
## namentror_tot:physical_barrier1  0.88272   0.57151   1.544   0.1225
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

```

```
emmip(zerofit_phys, ~physical_barrier | name, type='response', CI=T)
```



```

est<-emmeans(zerofit_phys, ~physical_barrier | name, type='response')
pairs(est)

```

```

## name = ngoro_tot:
## contrast                                odds.ratio     SE   df null z.ratio
## physical_barrier0 / physical_barrier1      1.016 0.409 Inf   1   0.040
## p.value
## 0.9684
##
## name = ntency_tot:
## contrast                                odds.ratio     SE   df null z.ratio
## physical_barrier0 / physical_barrier1      0.315 0.140 Inf   1  -2.608
## p.value

```

```

##    0.0091
##
## name = ntror_tot:
## contrast                               odds.ratio      SE   df null z.ratio
## physical_barrier0 / physical_barrier1     0.420 0.171 Inf    1 -2.134
## p.value
## 0.0329
##
## Tests are performed on the log odds ratio scale

```

```

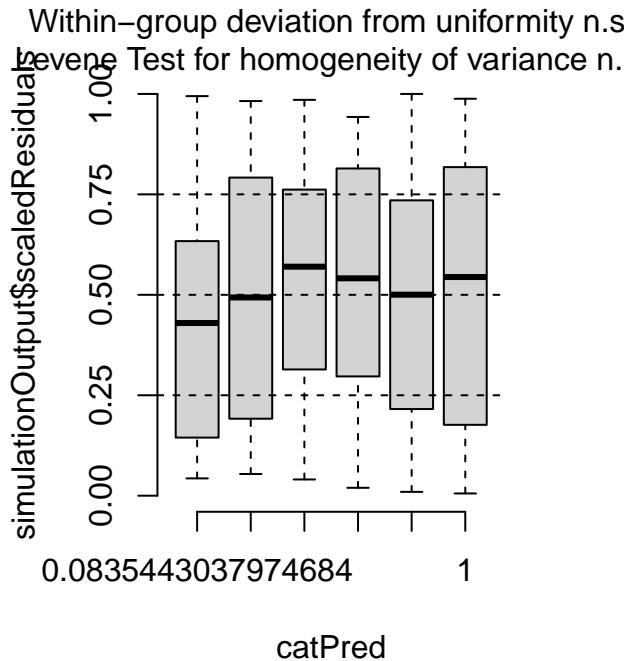
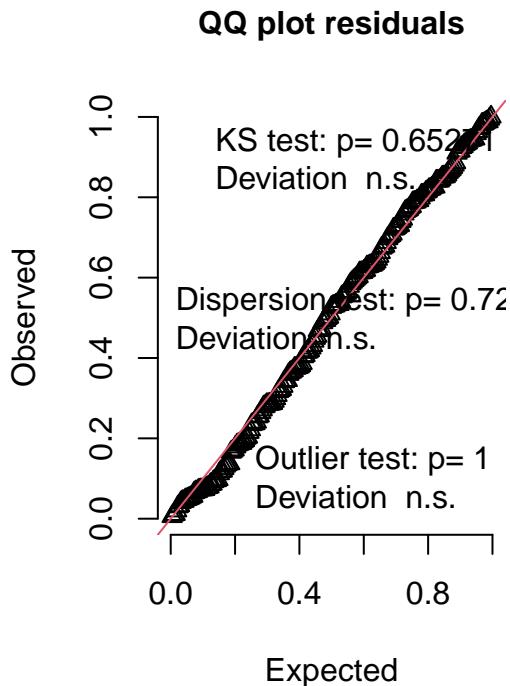
# physical barrier occurrence results
# physical barrier level does not significantly explain variation in goro
# trcy and tror probability of occurrence are higher in places where there is a physical barrier (p=0.0329)

### abundance with a truncated negbinom
countfit_phys<-glmmTMB(posicounts~name*physical_barrier+(1 | block), family=truncated_nbinom2(), data=d)

sim<-simulateResiduals(countfit_phys)
plot(sim)

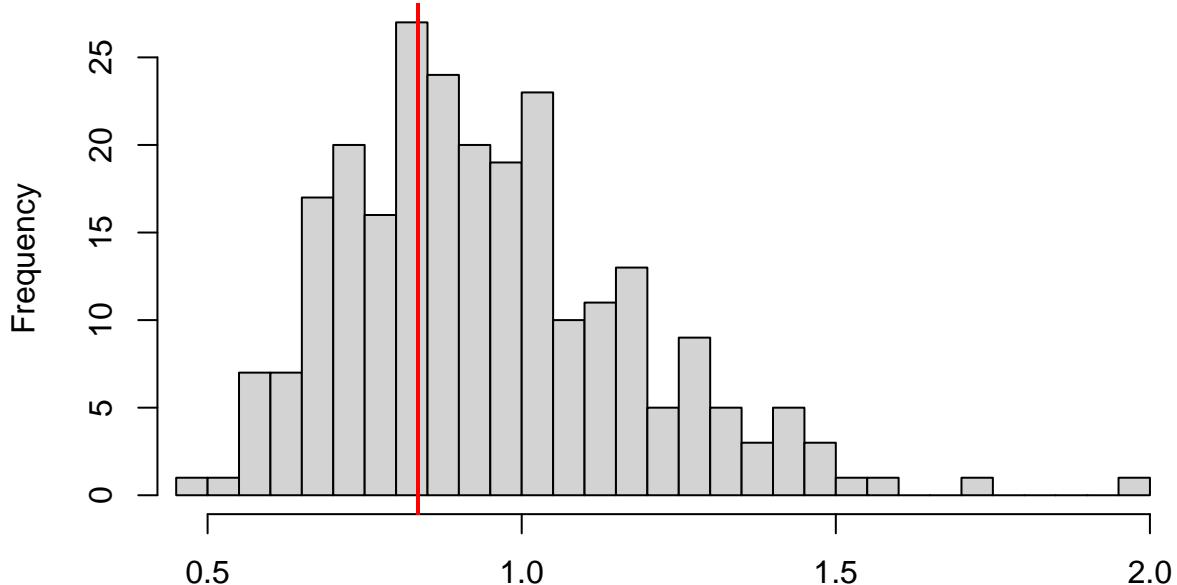
```

DHARMA residual



```
testDispersion(sim)
```

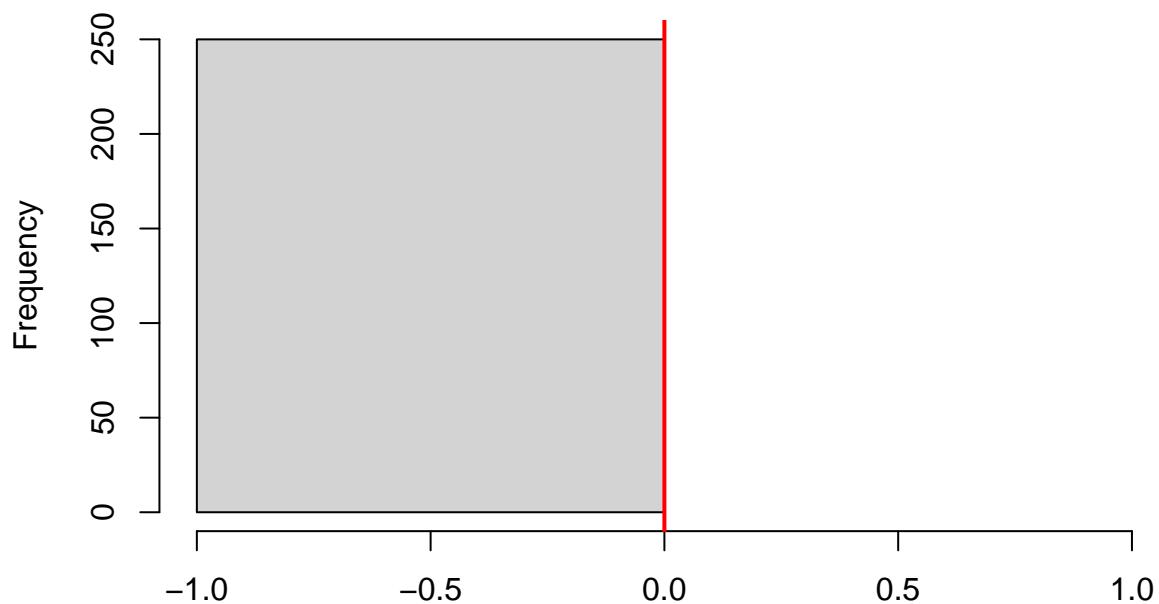
DHARMA nonparametric dispersion test via sd of residuals fitted vs. simulated



Simulated values, red line = fitted model. p-value (two.sided) = 0.728

```
##  
##  DHARMA nonparametric dispersion test via sd of residuals fitted vs.  
##  simulated  
##  
##  data:  simulationOutput  
##  dispersion = 0.88162, p-value = 0.728  
##  alternative hypothesis: two.sided  
  
testZeroInflation(sim)
```

DHARMA zero-inflation test via comparison to expected zeros with simulation under H0 = fitted model



Simulated values, red line = fitted model. p-value (two.sided) = 1

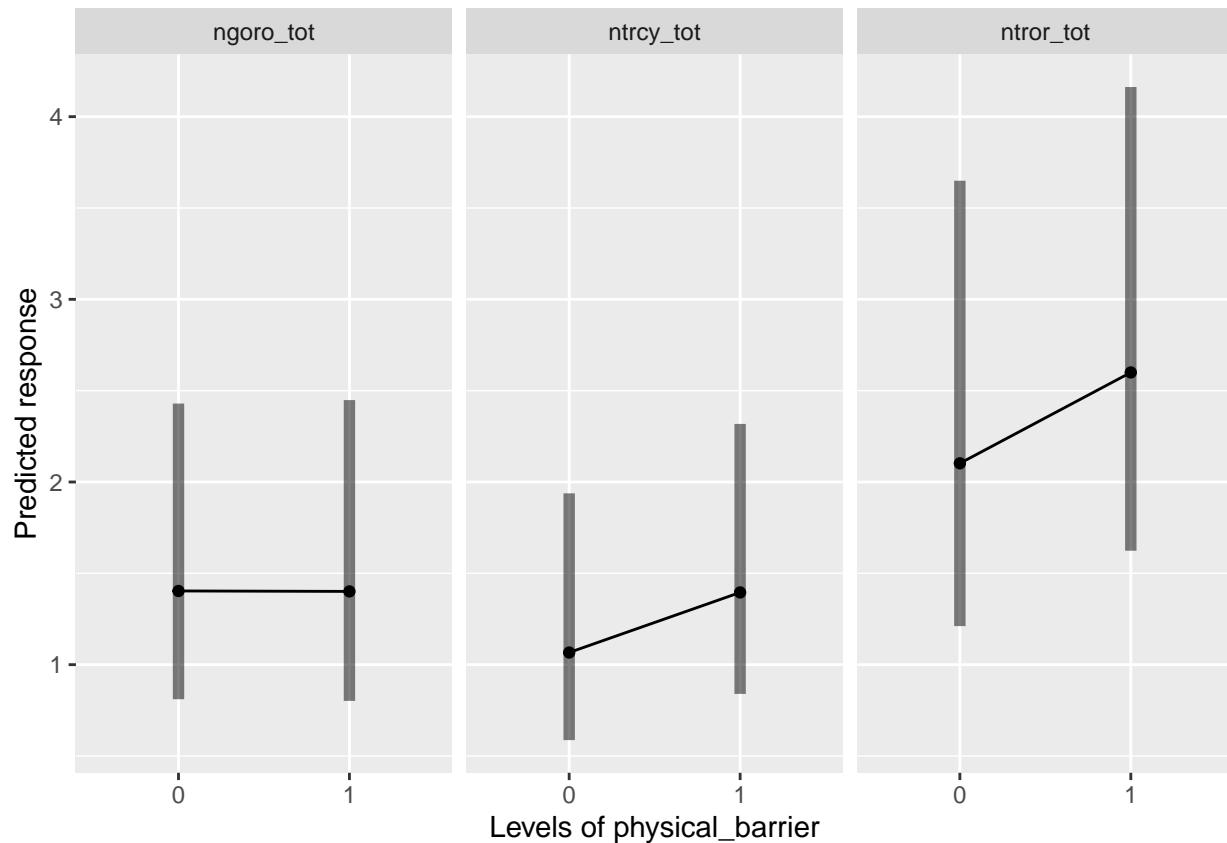
```
##  
## DHARMA zero-inflation test via comparison to expected zeros with  
## simulation under H0 = fitted model  
##  
## data: simulationOutput  
## ratioObsSim = NaN, p-value = 1  
## alternative hypothesis: two.sided  
  
summary(countfit_phys)  
  
## Family: truncated_nbinom2 ( log )  
## Formula: posicounts ~ name * physical_barrier + (1 | block)  
## Data: dat22  
##  
##      AIC      BIC  logLik deviance df.resid  
##    792.2    819.2   -388.1     776.2      209  
##  
## Random effects:  
##  
## Conditional model:  
## Groups Name        Variance Std.Dev.  
## block  (Intercept) 0.003716 0.06096  
## Number of obs: 217, groups: block, 7  
##  
## Dispersion parameter for truncated_nbinom2 family (): 0.826
```

```

## Conditional model:
## (Intercept)          Estimate Std. Error z value Pr(>|z|)
## namentrcy_tot       0.338919  0.279962  1.211   0.226
## namentror_tot       -0.274546  0.339942 -0.808   0.419
## physical_barrier1    0.404083  0.331575  1.219   0.223
## namentrcy_tot:physical_barrier1  -0.001728  0.326981 -0.005   0.996
## namentrcy_tot:physical_barrier1     0.270464  0.454932  0.595   0.552
## namentror_tot:physical_barrier1    0.214126  0.452559  0.473   0.636

```

```
emmip(countfit_phys, ~physical_barrier | name, type='response', CI=T)
```



```

est<-emmeans(countfit_phys, ~physical_barrier | name, type='response')
pairs(est)

```

```

## name = ngoro_tot:
## contrast                  ratio      SE  df null z.ratio p.value
## physical_barrier0 / physical_barrier1 1.002 0.328 Inf     1  0.005  0.9958
##
## name = ntrcy_tot:
## contrast                  ratio      SE  df null z.ratio p.value
## physical_barrier0 / physical_barrier1 0.764 0.247 Inf     1 -0.833  0.4048
##
## name = ntror_tot:
## contrast                  ratio      SE  df null z.ratio p.value

```

```

##  physical_barrier0 / physical_barrier1 0.809 0.251 Inf      1  -0.684  0.4937
##
## Tests are performed on the log scale

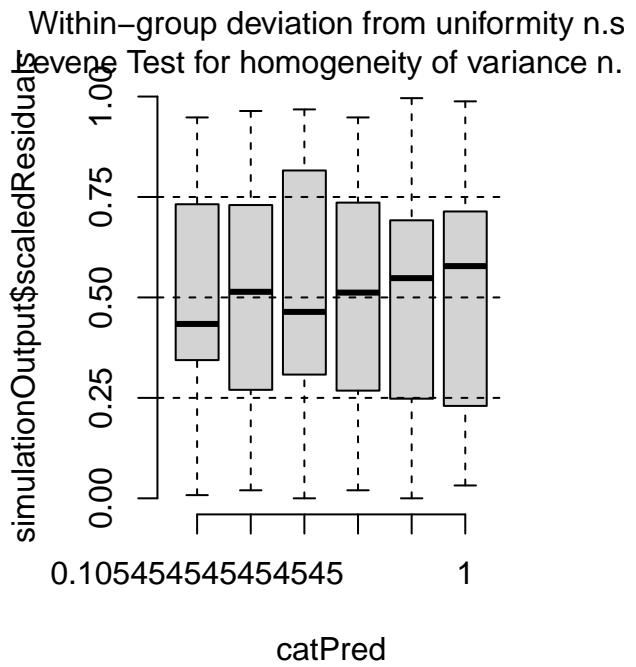
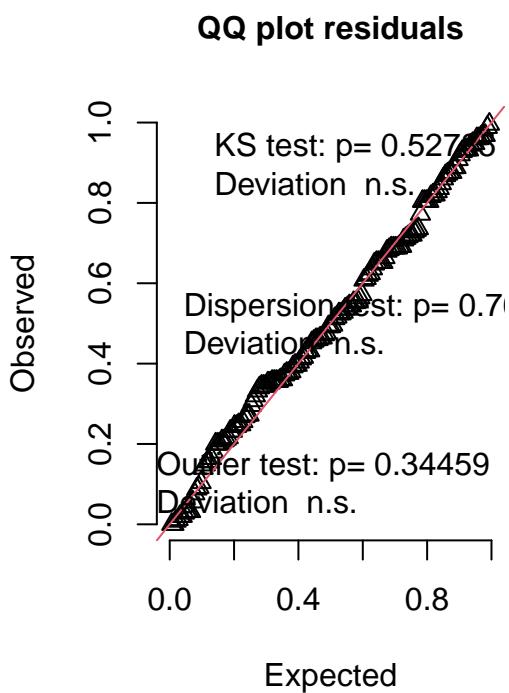
# abundance does not differ between physical barrier treatments for any of the plant species

# model
# pcwtmod_phys<-lmer(log_wt~name*physical_barrier+(1/block), data=pcwtdat, REML=FALSE) # singular
pcwtmod_phys<-lm(log_wt~name*physical_barrier, data=pcwtdat)

# test for fit, looks pretty good
sim<-simulateResiduals(pcwtmod_phys)
plot(sim)

```

DHARMA residual



```

# model summary
summary(pcwtmod_phys)

```

```

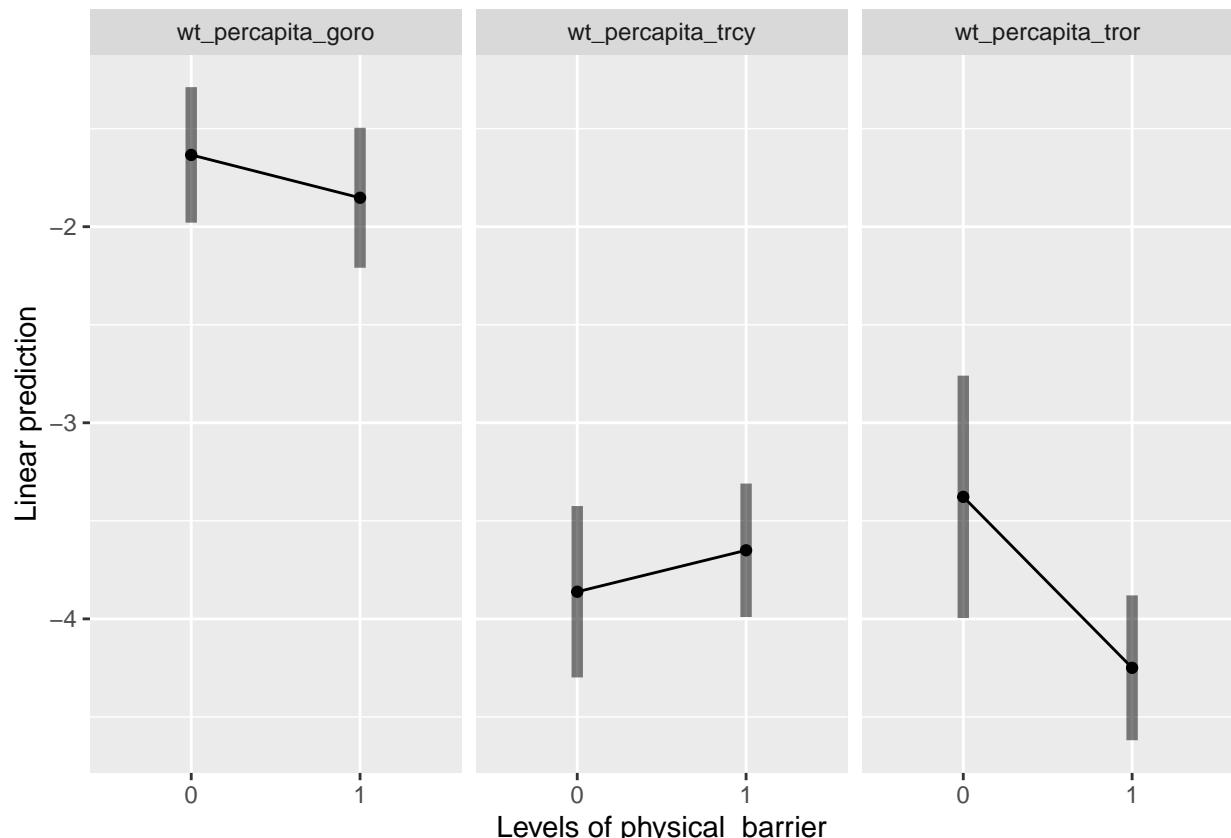
##
## Call:
## lm(formula = log_wt ~ name * physical_barrier, data = pcwtdat)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.684   -0.4937
## 
```

```

## -2.92995 -0.58816 -0.01781  0.55317  2.19042
##
## Coefficients:
##                               Estimate Std. Error t value Pr(>|t|)
## (Intercept)                 -1.6339   0.1748 -9.345 < 2e-16
## namewt_per capita_trcy     -2.2278   0.2819 -7.902 5.93e-13
## namewt_per capita_tror     -1.7439   0.3583 -4.867 2.89e-06
## physical_barrier1          -0.2187   0.2514 -0.870  0.386
## namewt_per capita_trcy:physical_barrier1  0.4301   0.3765  1.142  0.255
## namewt_per capita_tror:physical_barrier1 -0.6532   0.4426 -1.476  0.142
##
## (Intercept)                  ***
## namewt_per capita_trcy      ***
## namewt_per capita_tror      ***
## physical_barrier1
## namewt_per capita_trcy:physical_barrier1
## namewt_per capita_tror:physical_barrier1
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.9891 on 147 degrees of freedom
##   (183 observations deleted due to missingness)
## Multiple R-squared:  0.5463, Adjusted R-squared:  0.5309
## F-statistic: 35.41 on 5 and 147 DF,  p-value: < 2.2e-16

```

```
emmip(pwtmod_phys, ~physical_barrier | name, CI=T)
```



```

est<-emmeans(pcwtmod_phys, ~physical_barrier|name, type='response')
pairs(est)

## name = wt_percapita_goro:
## contrast                                estimate    SE  df t.ratio p.value
## physical_barrier0 - physical_barrier1    0.219 0.251 147   0.870  0.3858
##
## name = wt_percapita_trcy:
## contrast                                estimate    SE  df t.ratio p.value
## physical_barrier0 - physical_barrier1   -0.211 0.280 147  -0.754  0.4518
##
## name = wt_percapita_tror:
## contrast                                estimate    SE  df t.ratio p.value
## physical_barrier0 - physical_barrier1    0.872 0.364 147   2.393  0.0180

##### LEGACY ANALYSIS #####
##### log legacy response: do analysis for count by species #####
# fit3_leg<-glmmTMB(value~name*initial+(1 | block), ziformula=~., family=nbinom2(), data=dat22, REML=FALSE)
fit3_leg<-glmmTMB(value~name*initial+(1 | block), ziformula=~., family=poisson(), data=dat22, REML=FALSE)

## This stuff will just give you the end result counts with the zeros factored in...
summary(fit3_leg)

## Family: poisson  ( log )
## Formula:         value ~ name * initial + (1 | block)
## Zero inflation: ~.
## Data: dat22
##
##      AIC      BIC  logLik deviance df.resid
##  1308.2  1361.7  -640.1   1280.2     322
##
## Random effects:
##
## Conditional model:
## Groups Name        Variance Std.Dev.
## block  (Intercept) 0.01342  0.1158
## Number of obs: 336, groups: block, 7
##
## Zero-inflation model:
## Groups Name        Variance Std.Dev.
## block  (Intercept) 0.3289   0.5735
## Number of obs: 336, groups: block, 7
##
## Conditional model:
##                               Estimate Std. Error z value Pr(>|z|)
## (Intercept)                0.93257   0.12586  7.410 1.27e-13 ***
## namentrcy_tot             -0.11740   0.16141 -0.727  0.467
## namentror_tot              0.24085   0.15311  1.573  0.116
## initialopen               -0.26979   0.17503 -1.541  0.123
## namentrcy_tot:initialopen  0.01648   0.25354  0.065  0.948
## namentror_tot:initialopen  0.35102   0.22307  1.574  0.116

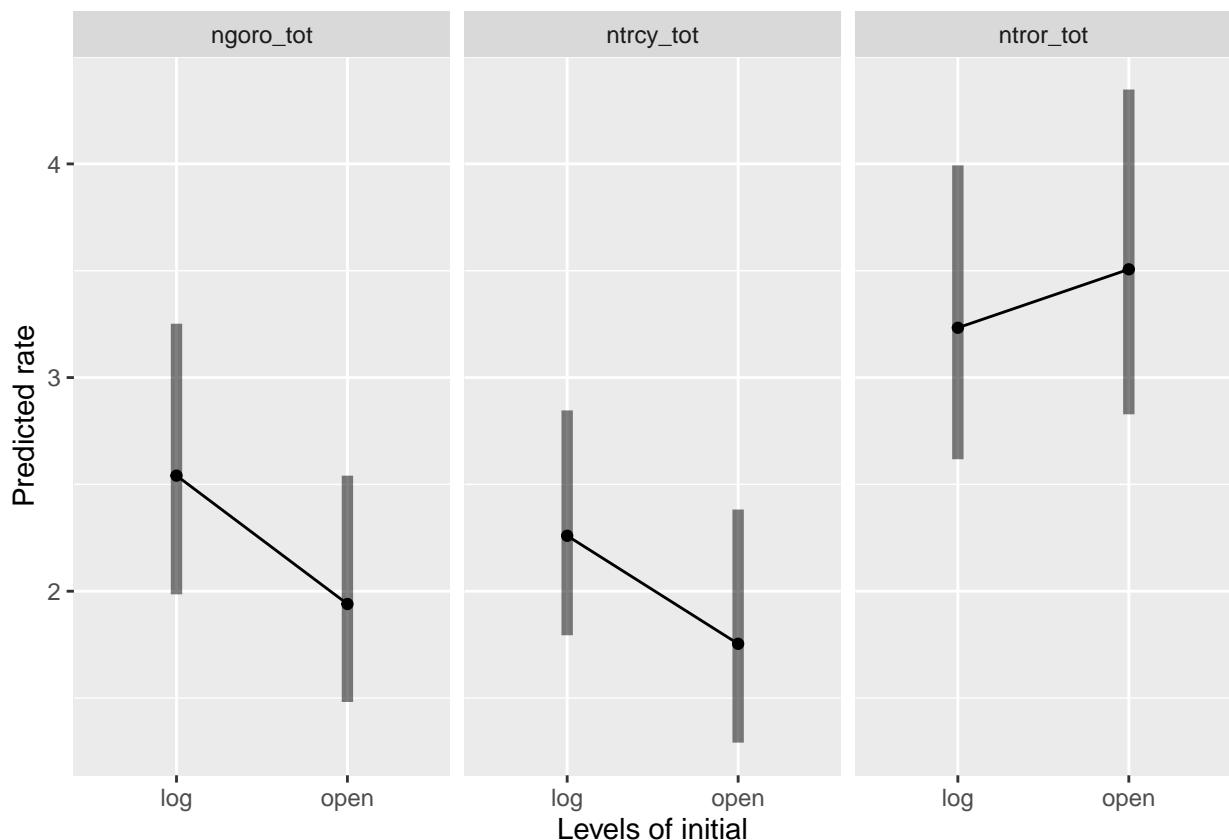
```

```

## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Zero-inflation model:
##                               Estimate Std. Error z value Pr(>|z|)
## (Intercept)                -0.71756   0.40605 -1.767   0.0772 .
## namentrcy_tot              -1.32361   0.68892 -1.921   0.0547 .
## namentror_tot              -0.03813   0.46329 -0.082   0.9344
## initialopen                -0.55825   0.56794 -0.983   0.3256
## namentrcy_tot:initialopen  1.49156   0.94728  1.575   0.1154
## namentror_tot:initialopen  0.91272   0.71464  1.277   0.2015
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

emmap(fit3_leg, ~initial|name, type='response', CI=T)

```



```

est<-emmeans(fit3_leg, ~initial|name, type='response')
pairs(est)

```

```

## name = ngoro_tot:
## contrast ratio    SE df null z.ratio p.value
## log / open 1.310 0.229 Inf     1    1.541  0.1232
##
## name = ntrcy_tot:
## contrast ratio    SE df null z.ratio p.value

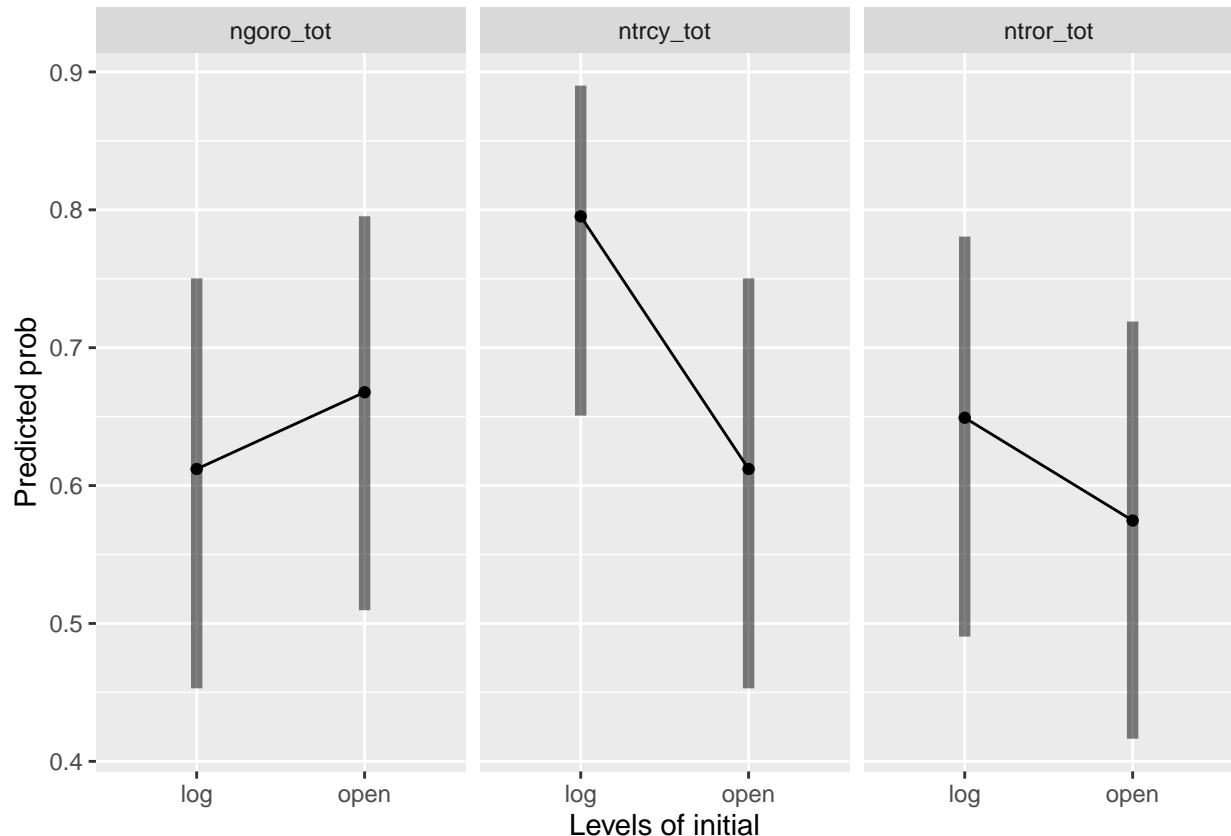
```

```

##  log / open 1.288 0.237 Inf     1   1.380  0.1676
##
## name = ntror_tot:
## contrast   ratio    SE  df null z.ratio p.value
##  log / open 0.922 0.127 Inf     1  -0.589  0.5561
##
## Tests are performed on the log scale

##### split up occurrence and abundance
##### zeros and ones
zerofit_leg<-glmmTMB(presence~name*initial+(1 | block), family=binomial, data=dat22, REML=FALSE)
emmip(zerofit_leg, ~initial|name, type='response', CI=T)

```



```

est<-emmeans(zerofit_leg, ~initial|name, type='response')
pairs(est)

```

```

## name = ngoro_tot:
## contrast   odds.ratio    SE  df null z.ratio p.value
##  log / open      0.785 0.316 Inf     1  -0.602  0.5473
##
## name = ntrcy_tot:
## contrast   odds.ratio    SE  df null z.ratio p.value
##  log / open      2.462 1.070 Inf     1   2.074  0.0381
##
## name = ntror_tot:

```

```

## contrast odds.ratio    SE   df null z.ratio p.value
## log / open      1.370 0.545 Inf     1   0.791  0.4287
##
## Tests are performed on the log odds ratio scale

```

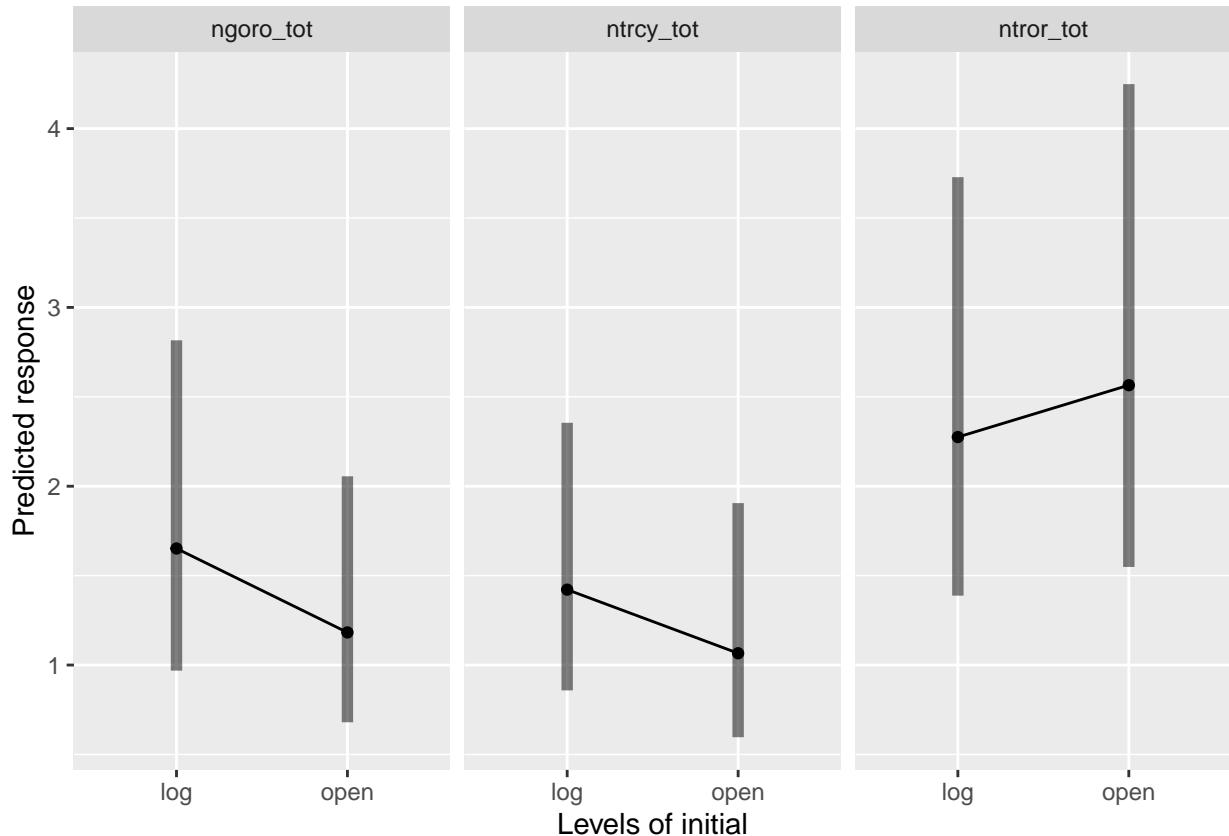
probability of occurrence is lower for trcy in places where initial treatment is open (higher where trcy is log)

abundance with a truncated negbinom

```

countfit_leg<-glmmTMB(posicounts~name*initial+(1 | block), family=truncated_nbinom2(), data=dat22, REML=T)
emmip(countfit_leg, ~initial|name, type='response', CI=T)

```



```

est<-emmeans(countfit_leg, ~initial|name, type='response')
pairs(est)

```

```

## name = ngoro_tot:
## contrast ratio    SE   df null z.ratio p.value
## log / open 1.397 0.449 Inf     1   1.041  0.2979
##
## name = ntrcy_tot:
## contrast ratio    SE   df null z.ratio p.value
## log / open 1.334 0.423 Inf     1   0.908  0.3641
##
## name = ntror_tot:
## contrast ratio    SE   df null z.ratio p.value
## log / open 0.887 0.268 Inf     1   -0.397  0.6917

```

```

##  

## Tests are performed on the log scale  
  

# model for per capita weight  

# pcwtmod_leg<-lmer(log_wt~name*initial+(1/block), data=pcwtdat, REML=FALSE) #singular  

pcwtmod_leg<-lm(log_wt~name*initial, data=pcwtdat)  
  

# test for fit, looks pretty good  

# sim<-simulateResiduals(pcwtmod_leg)  

# plot(sim)  
  

# model summary  

summary(pcwtmod_leg)  
  

##  

## Call:  

## lm(formula = log_wt ~ name * initial, data = pcwtdat)  

##  

## Residuals:  

##      Min      1Q  Median      3Q     Max  

## -3.0503 -0.5868 -0.0689  0.5899  2.3033  

##  

## Coefficients:  

##                               Estimate Std. Error t value Pr(>|t|)  

## (Intercept)             -1.73219   0.18030 -9.607 < 2e-16 ***  

## namewt_per capita_trcy -2.18533   0.25097 -8.708 5.88e-15 ***  

## namewt_per capita_tror -1.97868   0.28098 -7.042 6.75e-11 ***  

## initialopen            -0.01455   0.25097 -0.058  0.9539  

## namewt_per capita_trcy:initialopen  0.48763   0.37404  1.304  0.1944  

## namewt_per capita_tror:initialopen -0.67691   0.40841 -1.657  0.0996 .  

## ---  

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

##  

## Residual standard error: 0.9876 on 147 degrees of freedom  

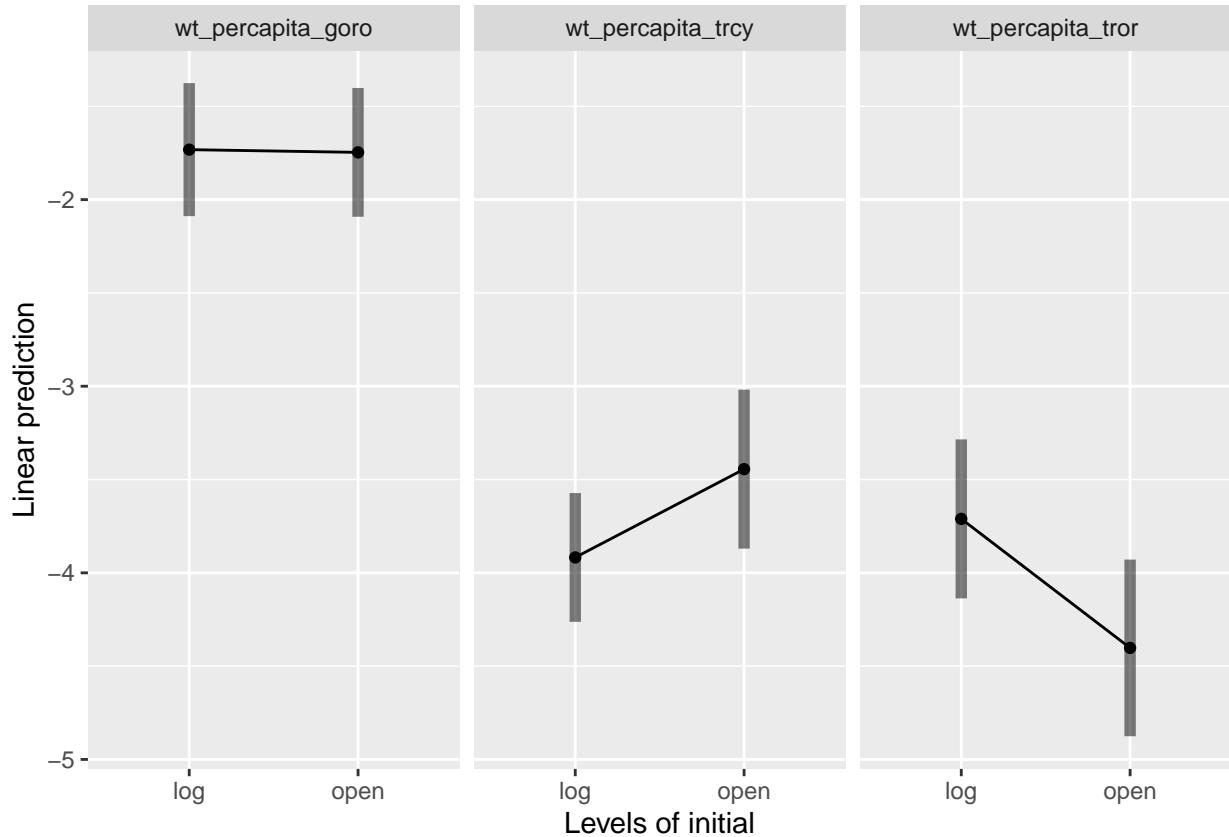
##   (183 observations deleted due to missingness)  

## Multiple R-squared:  0.5477, Adjusted R-squared:  0.5323  

## F-statistic:  35.6 on 5 and 147 DF,  p-value: < 2.2e-16  
  

emmap(pcwtmod_leg,~initial|name,CI=T)

```



```
est<-emmeans(pcwtmod_leg, ~initial|name, type='response')
pairs(est)
```

```
## name = wt_per capita_goro:
## contrast estimate SE df t.ratio p.value
## log - open 0.0145 0.251 147 0.058 0.9539
##
## name = wt_per capita_trcy:
## contrast estimate SE df t.ratio p.value
## log - open -0.4731 0.277 147 -1.706 0.0902
##
## name = wt_per capita_tror:
## contrast estimate SE df t.ratio p.value
## log - open 0.6915 0.322 147 2.146 0.0335
```

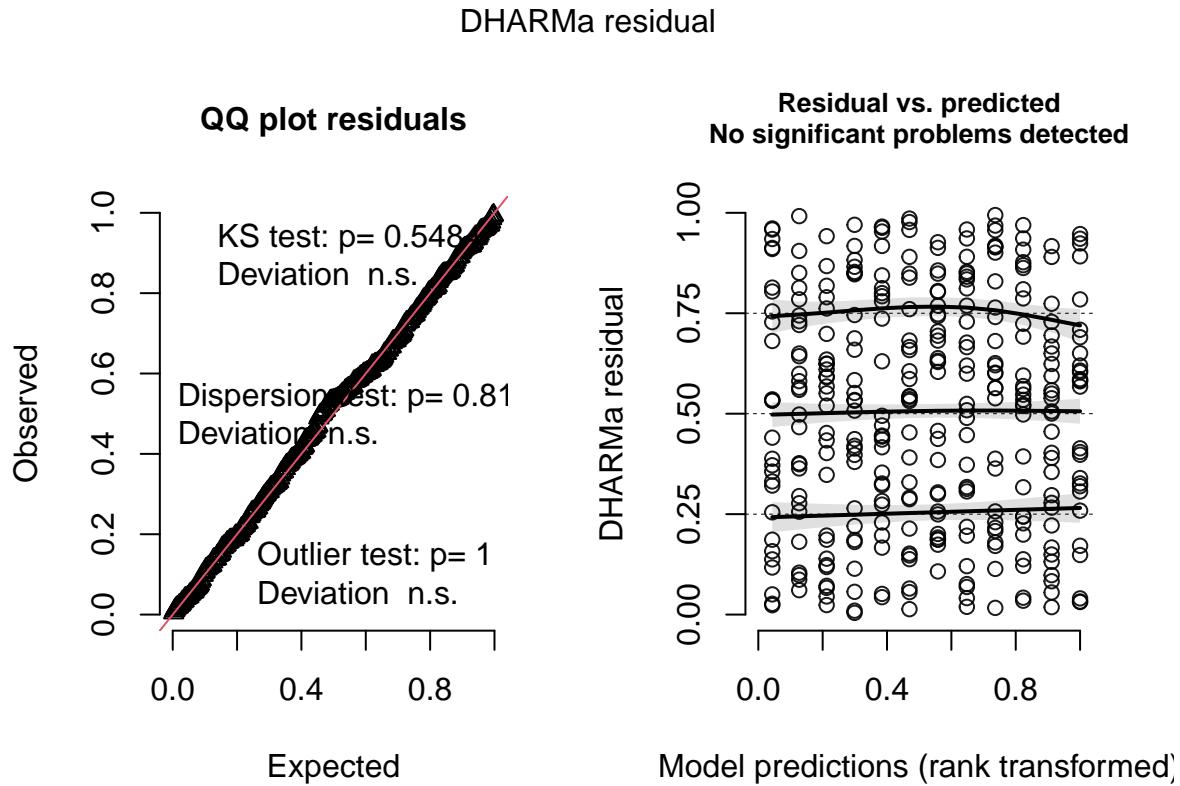
biomass per capita in tror is lower in open legacy environments ($p=0.03$); biomass per capita in trcy

PHYSICAL BARRIER X LEGACY ANALYSIS #####
What about both physical and legacy?

interaction model
zeros and ones

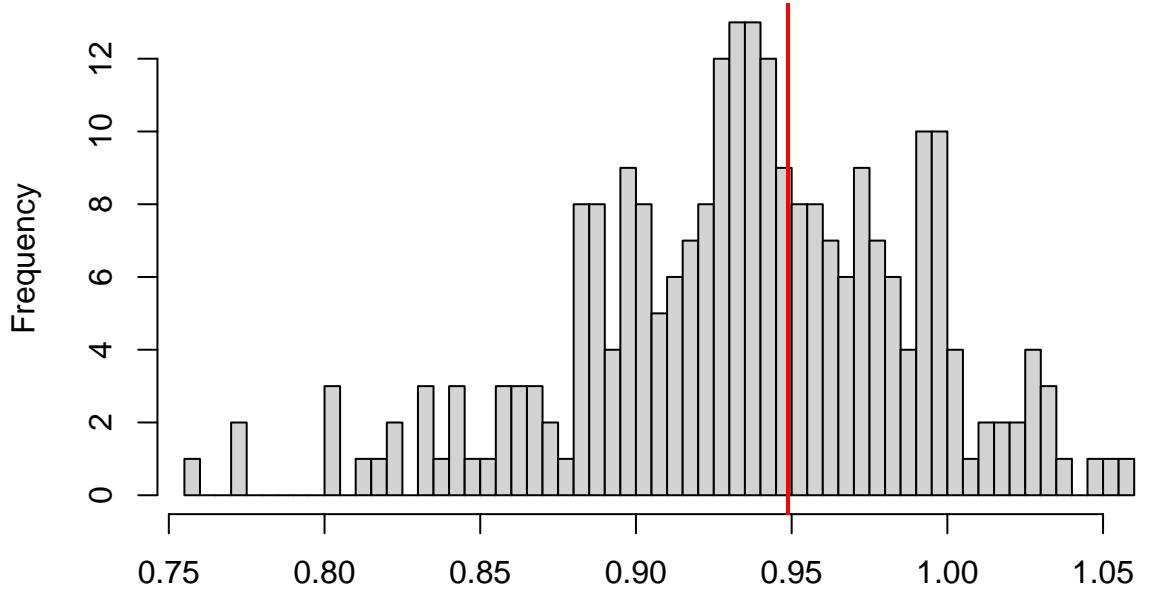
```
zerofit_intxn<-glmmTMB(presence~name*physical_barrier*initial+(1 | block), family=binomial, data=dat22,
```

```
sim<-simulateResiduals(zeroFit_intxn)
plot(sim)
```



```
testDispersion(sim)
```

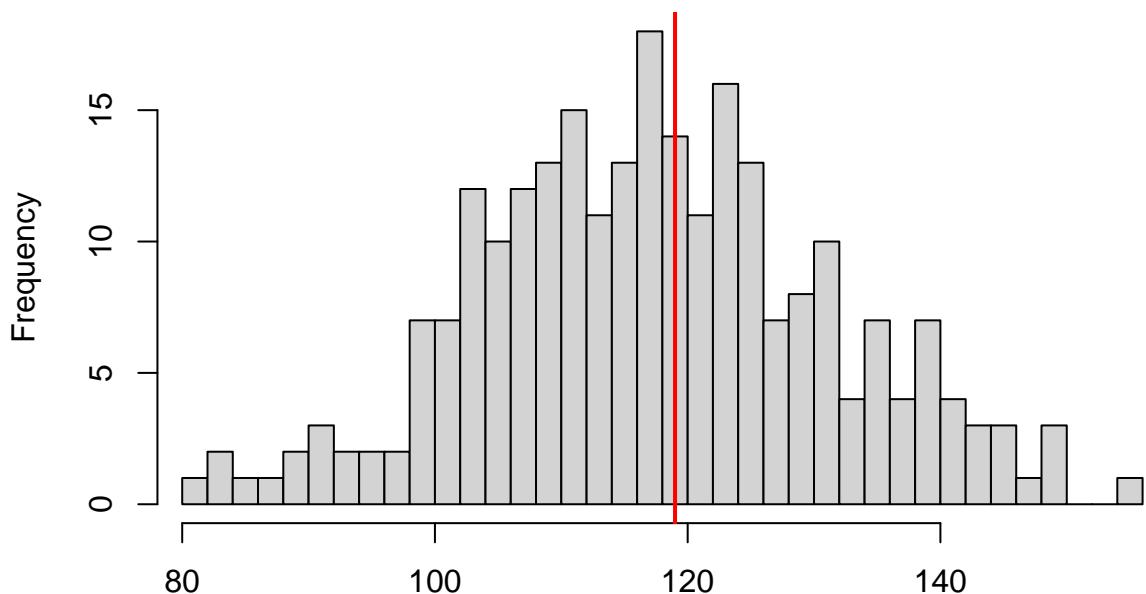
DHARMA nonparametric dispersion test via sd of residuals fitted vs. simulated



Simulated values, red line = fitted model. p-value (two.sided) = 0.816

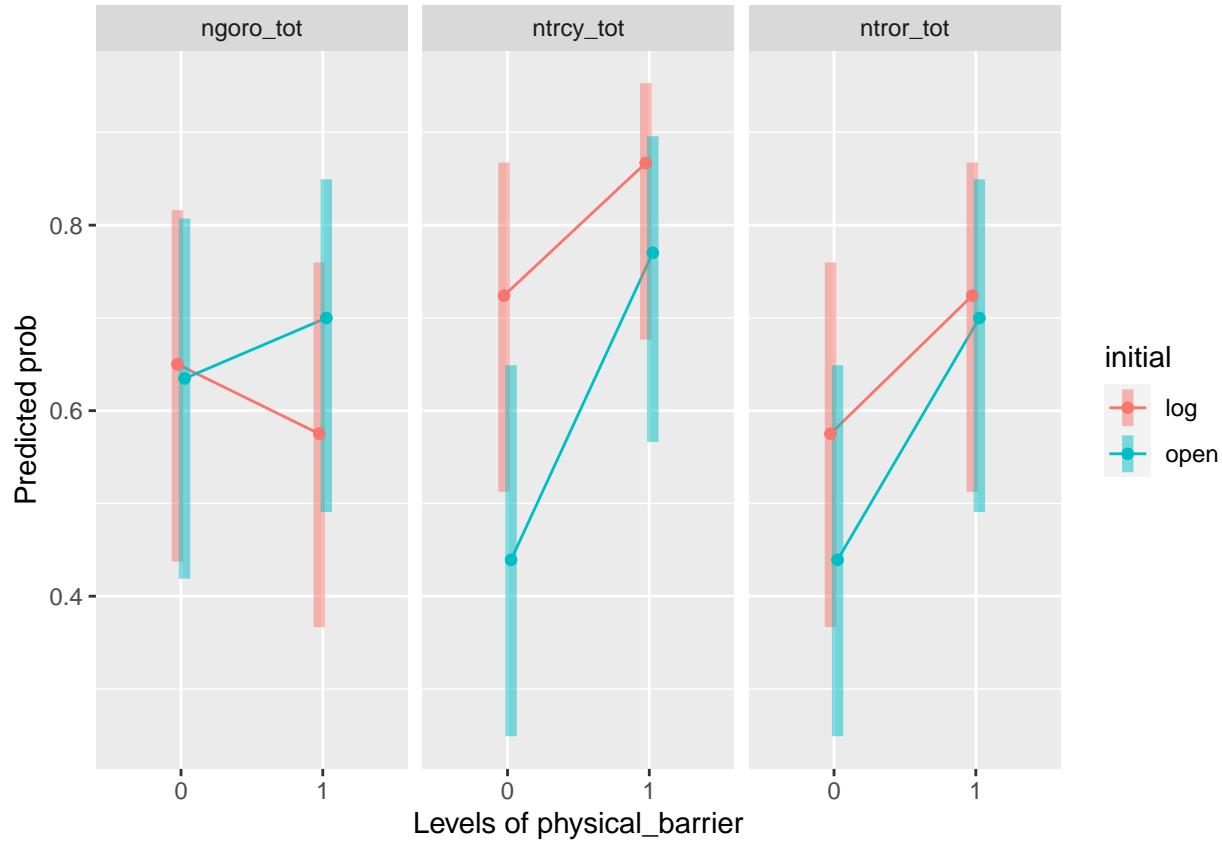
```
##  
## DHARMA nonparametric dispersion test via sd of residuals fitted vs.  
## simulated  
##  
## data: simulationOutput  
## dispersion = 1.0148, p-value = 0.816  
## alternative hypothesis: two.sided  
  
testZeroInflation(sim)
```

DHARMA zero-inflation test via comparison to expected zeros with simulation under H0 = fitted model



Simulated values, red line = fitted model. p-value (two.sided) = 0.928

```
##  
## DHARMA zero-inflation test via comparison to expected zeros with  
## simulation under H0 = fitted model  
##  
## data: simulationOutput  
## ratioObsSim = 1.0104, p-value = 0.928  
## alternative hypothesis: two.sided  
  
emmap(zeroFit_intxn, initial ~ physical_barrier | name, type = 'response', CI = T)
```



```
est<-emmeans(zerofit_intxn, ~physical_barrier|name|initial, type='response')
pairs(est)
```

```
## name = ngoro_tot, initial = log:
## contrast                                odds.ratio      SE   df null z.ratio
## physical_barrier0 / physical_barrier1     1.372 0.774 Inf   1  0.561
## p.value
## 0.5745
##
## name = ntrcy_tot, initial = log:
## contrast                                odds.ratio      SE   df null z.ratio
## physical_barrier0 / physical_barrier1     0.403 0.280 Inf   1 -1.307
## p.value
## 0.1911
##
## name = ntror_tot, initial = log:
## contrast                                odds.ratio      SE   df null z.ratio
## physical_barrier0 / physical_barrier1     0.516 0.300 Inf   1 -1.139
## p.value
## 0.2549
##
## name = ngoro_tot, initial = open:
## contrast                                odds.ratio      SE   df null z.ratio
## physical_barrier0 / physical_barrier1     0.745 0.433 Inf   1 -0.507
## p.value
```

```

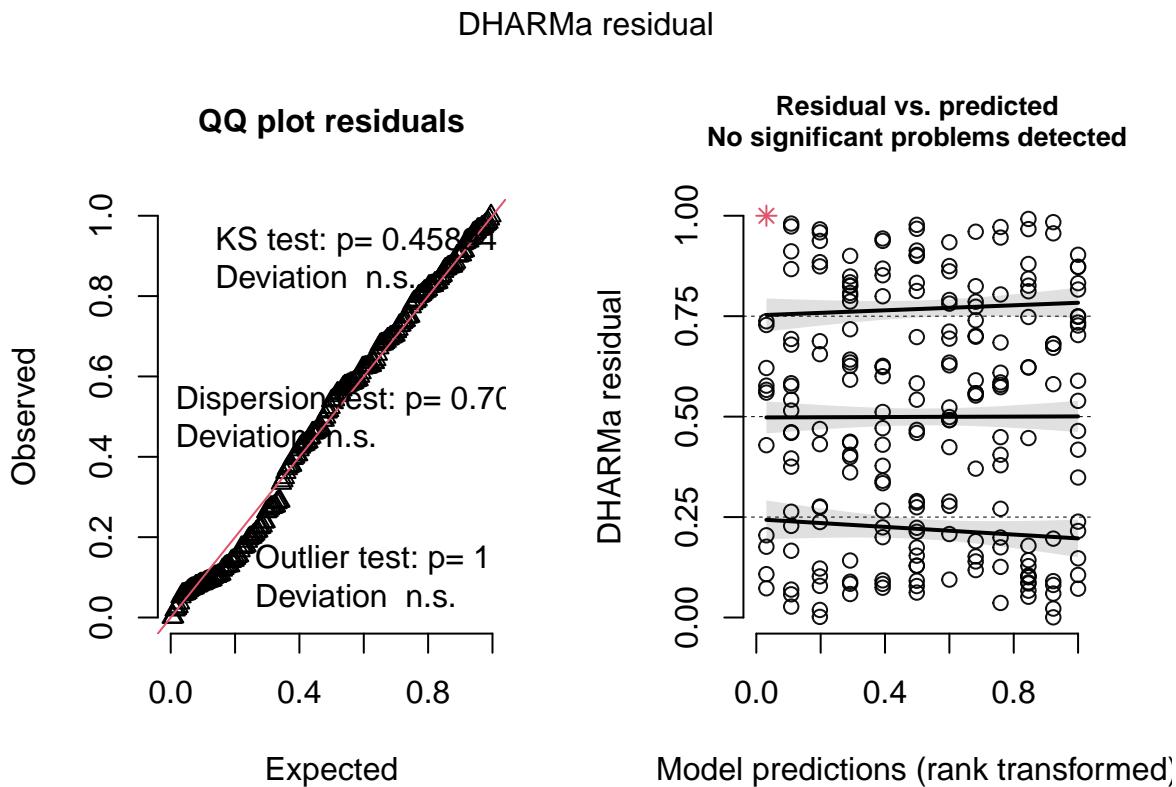
##    0.6121
##
## name = ntrcy_tot, initial = open:
## contrast                               odds.ratio      SE   df null z.ratio
## physical_barrier0 / physical_barrier1     0.234 0.140 Inf   1 -2.433
## p.value
## 0.0150
##
## name = ntror_tot, initial = open:
## contrast                               odds.ratio      SE   df null z.ratio
## physical_barrier0 / physical_barrier1     0.336 0.193 Inf   1 -1.902
## p.value
## 0.0571
##
## Tests are performed on the log odds ratio scale

# goro presence/absence not explained by physical barrier or initial treatment
# when the initial treatment is "open", trcy has higher probability of occurring when there is a physical barrier
# if there is a legacy of a log, then there is no significant difference between physical barrier treatments

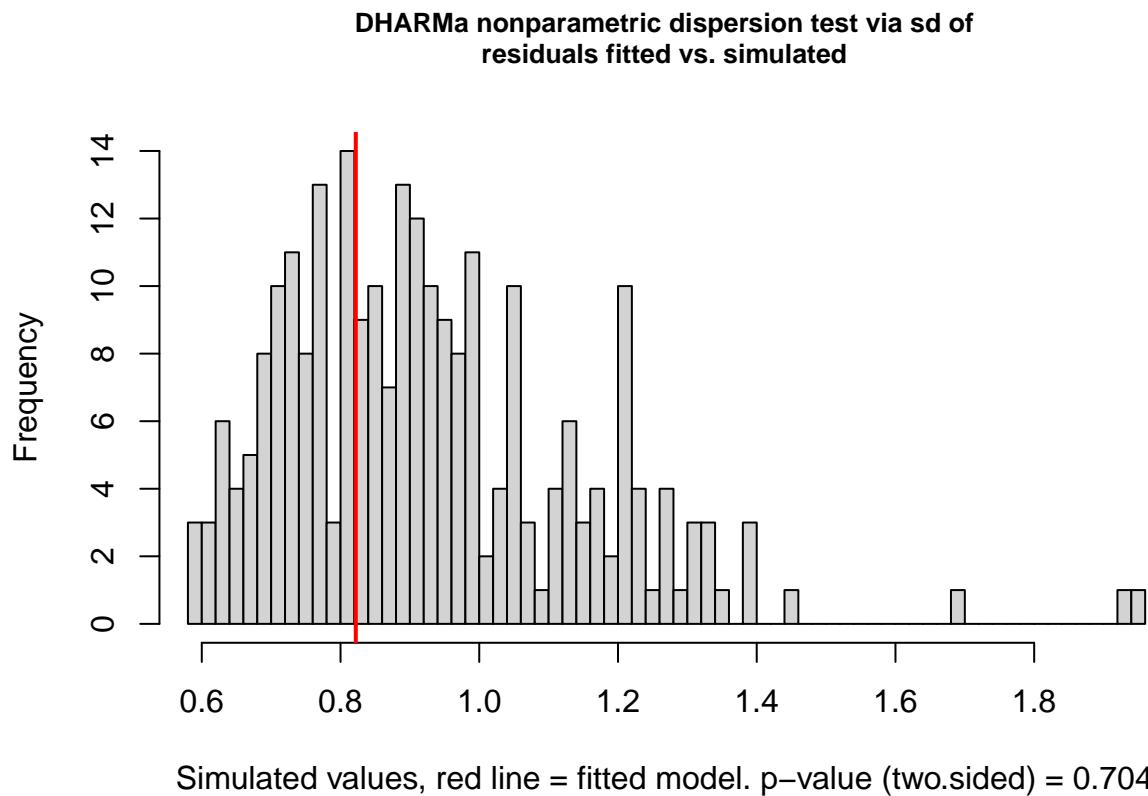
### abundance with a truncated negbinom
countfit_intxn<-glmmTMB(posicounts~name*physical_barrier*initial+(1 | block), family=truncated_nbino(2))

sim<-simulateResiduals(countfit_intxn)
plot(sim)

```



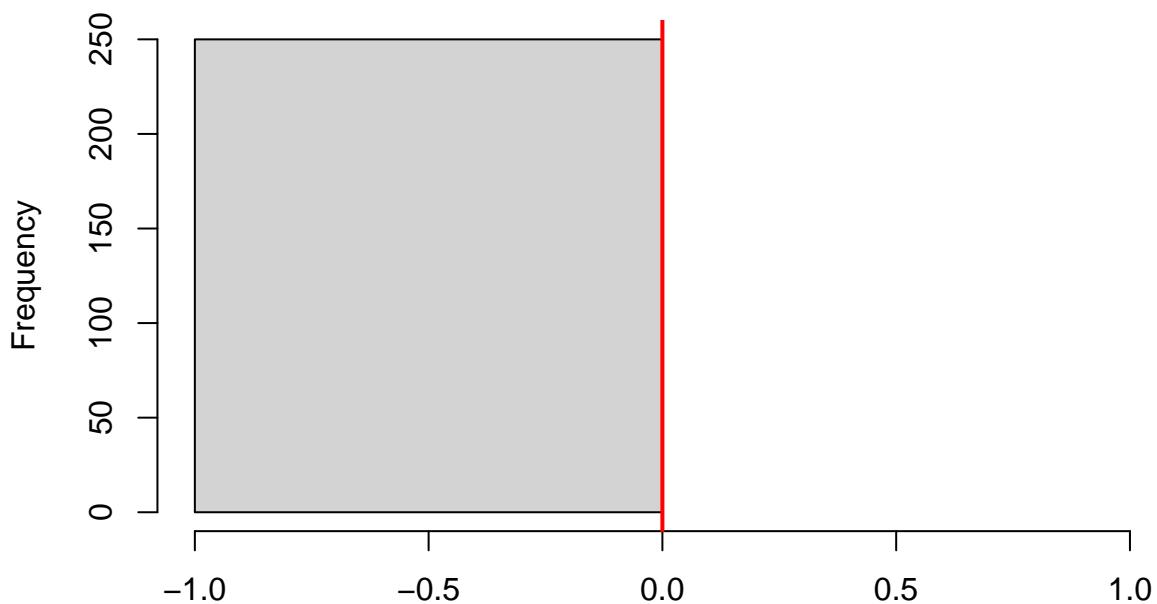
```
testDispersion(sim)
```



```
##  
## DHARMA nonparametric dispersion test via sd of residuals fitted vs.  
## simulated  
##  
## data: simulationOutput  
## dispersion = 0.88343, p-value = 0.704  
## alternative hypothesis: two.sided
```

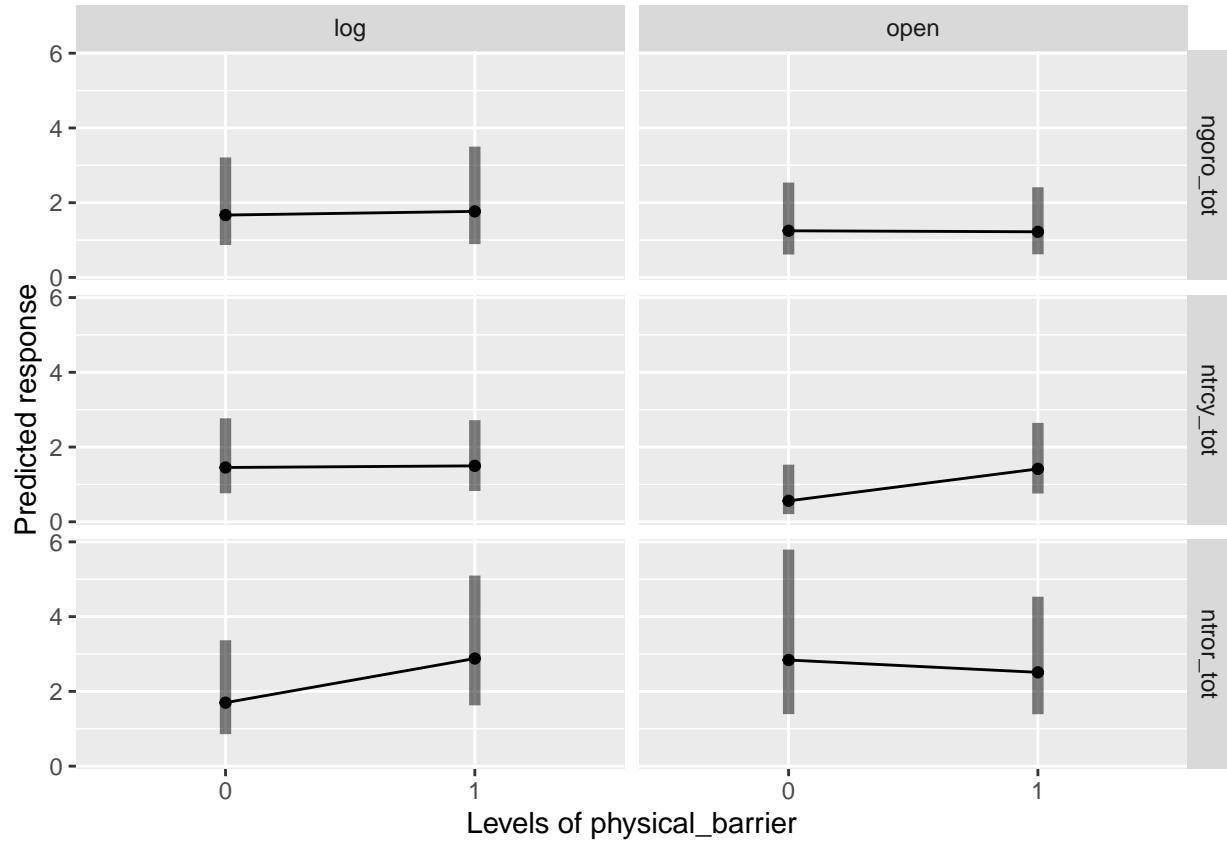
```
testZeroInflation(sim)
```

DHARMA zero-inflation test via comparison to expected zeros with simulation under H0 = fitted model



Simulated values, red line = fitted model. p-value (two.sided) = 1

```
##  
## DHARMA zero-inflation test via comparison to expected zeros with  
## simulation under H0 = fitted model  
##  
## data: simulationOutput  
## ratioObsSim = NaN, p-value = 1  
## alternative hypothesis: two.sided  
  
emmap(countfit_intxn, ~physical_barrier | name | initial, type='response', CI=T)
```



```
est<-emmeans(countfit_intxn, ~physical_barrier|name|initial, type='response')
pairs(est)
```

```
## name = ngoro_tot, initial = log:
## contrast                                ratio      SE  df null z.ratio p.value
## physical_barrier0 / physical_barrier1 0.944  0.415 Inf   1 -0.130  0.8965
##
## name = ntrcy_tot, initial = log:
## contrast                                ratio      SE  df null z.ratio p.value
## physical_barrier0 / physical_barrier1 0.971  0.387 Inf   1 -0.075  0.9405
##
## name = ntror_tot, initial = log:
## contrast                                ratio      SE  df null z.ratio p.value
## physical_barrier0 / physical_barrier1 0.590  0.246 Inf   1 -1.268  0.2047
##
## name = ngoro_tot, initial = open:
## contrast                                ratio      SE  df null z.ratio p.value
## physical_barrier0 / physical_barrier1 1.022  0.465 Inf   1  0.047  0.9624
##
## name = ntrcy_tot, initial = open:
## contrast                                ratio      SE  df null z.ratio p.value
## physical_barrier0 / physical_barrier1 0.396  0.222 Inf   1 -1.655  0.0979
##
## name = ntror_tot, initial = open:
## contrast                                ratio      SE  df null z.ratio p.value
```

```

##  physical_barrier0 / physical_barrier1 1.131 0.496 Inf      1   0.281  0.7786
##
## Tests are performed on the log scale

# abundance not explained by physical barrier or initial treatment for all three species

######_ additive model
zerofit_add<-glmmTMB(presence~name*physical_barrier+name*initial+(1 | block), family=binomial, data=dat22)
summary(zerofit_add)

## Family: binomial ( logit )
## Formula:
## presence ~ name * physical_barrier + name * initial + (1 | block)
## Data: dat22
##
##      AIC      BIC  logLik deviance df.resid
##     430.2    468.3   -205.1     410.2      326
##
## Random effects:
##
## Conditional model:
## Groups Name            Variance Std.Dev.
## block  (Intercept) 0.2301   0.4797
## Number of obs: 336, groups: block, 7
##
## Conditional model:
##                               Estimate Std. Error z value Pr(>|z|)
## (Intercept)                  0.46829   0.39088   1.198   0.2309
## namentrcy_tot                 0.38079   0.50971   0.747   0.4550
## namentror_tot                -0.26455   0.48701  -0.543   0.5870
## physical_barrier1             -0.02049   0.40340  -0.051   0.9595
## initialopen                   0.24376   0.40345   0.604   0.5457
## namentrcy_tot:physical_barrier1 1.24985   0.60881   2.053   0.0401 *
## namentror_tot:physical_barrier1  0.90138   0.57391   1.571   0.1163
## namentrcy_tot:initialopen      -1.23613   0.60711  -2.036   0.0417 *
## namentror_tot:initialopen      -0.58979   0.57314  -1.029   0.3035
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

countfit_add<-glmmTMB(posicounts~name*physical_barrier+name*initial+(1 | block), family=truncated_nbinom2, data=dat22)
summary(countfit_add)

## Family: truncated_nbinom2 ( log )
## Formula:           posicounts ~ name * physical_barrier + name * initial + (1 |
##                      block)
## Data: dat22
##
##      AIC      BIC  logLik deviance df.resid
##     795.7    832.9   -386.8     773.7      206
##
## Random effects:
##
## Conditional model:

```

```

## Groups Name      Variance Std.Dev.
## block  (Intercept) 0.001068 0.03268
## Number of obs: 217, groups: block, 7
##
## Dispersion parameter for truncated_nbinom2 family (): 0.86
##
## Conditional model:
##                               Estimate Std. Error z value Pr(>|z|)
## (Intercept)                0.50851   0.30858  1.648   0.0994 .
## namentrcy_tot             -0.32686   0.37921 -0.862   0.3887
## namentror_tot              0.18583   0.39190  0.474   0.6354
## physical_barrier1         0.01149   0.32558  0.035   0.9718
## initialopen                -0.33362   0.31936 -1.045   0.2962
## namentrcy_tot:physical_barrier1 0.32698   0.45821  0.714   0.4755
## namentror_tot:physical_barrier1 0.20696   0.44973  0.460   0.6454
## namentrcy_tot:initialopen     -0.02167   0.45416 -0.048   0.9619
## namentror_tot:initialopen     0.46119   0.44224  1.043   0.2970
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

### per capita biomass

# model - intxn
# pcwtmod_intxn<-lmer(log_wt~name*physical_barrier*initial+(1/block), data=pcwtdat, REML=FALSE) # singular
pcwtmod_intxn<-lm(log_wt~name*physical_barrier*initial, data=pcwtdat)

# model - no intxn
# pcwtmod_add<-lmer(log_wt~name*physical_barrier+name*initial+(1/block), data=pcwtdat, REML=FALSE) # singular
pcwtmod_add<-lm(log_wt~name*physical_barrier+name*initial, data=pcwtdat)
summary(pcwtmod_add)

##
## Call:
## lm(formula = log_wt ~ name * physical_barrier + name * initial,
##      data = pcwtdat)
##
## Residuals:
##      Min        1Q        Median        3Q       Max
## -2.93383 -0.49154 -0.02478  0.54954  2.19405
##
## Coefficients:
##                               Estimate Std. Error t value Pr(>|t|)
## (Intercept)                -1.630260   0.212724 -7.664 2.44e-12
## namewt_percapita_trcy      -2.343605   0.312963 -7.488 6.41e-12
## namewt_percapita_tror      -1.487665   0.396231 -3.755 0.000251
## physical_barrier1           -0.218414   0.248409 -0.879 0.380728
## initialopen                 -0.007268   0.248409 -0.029 0.976699
## namewt_percapita_trcy:physical_barrier1 0.324482   0.377806  0.859 0.391846
## namewt_percapita_tror:physical_barrier1 -0.611710   0.437783 -1.397 0.164475
## namewt_percapita_trcy:initialopen        0.455888   0.375886  1.213 0.227179
## namewt_percapita_tror:initialopen        -0.642331   0.404507 -1.588 0.114494
##
## (Intercept) ***
## namewt_percapita_trcy ***

```

```

## namewt_percapita_tror ***  

## physical_barrier1  

## initialopen  

## namewt_percapita_trcy:physical_barrier1  

## namewt_percapita_tror:physical_barrier1  

## namewt_percapita_trcy:initialopen  

## namewt_percapita_tror:initialopen  

## ---  

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

##  

## Residual standard error: 0.9769 on 144 degrees of freedom  

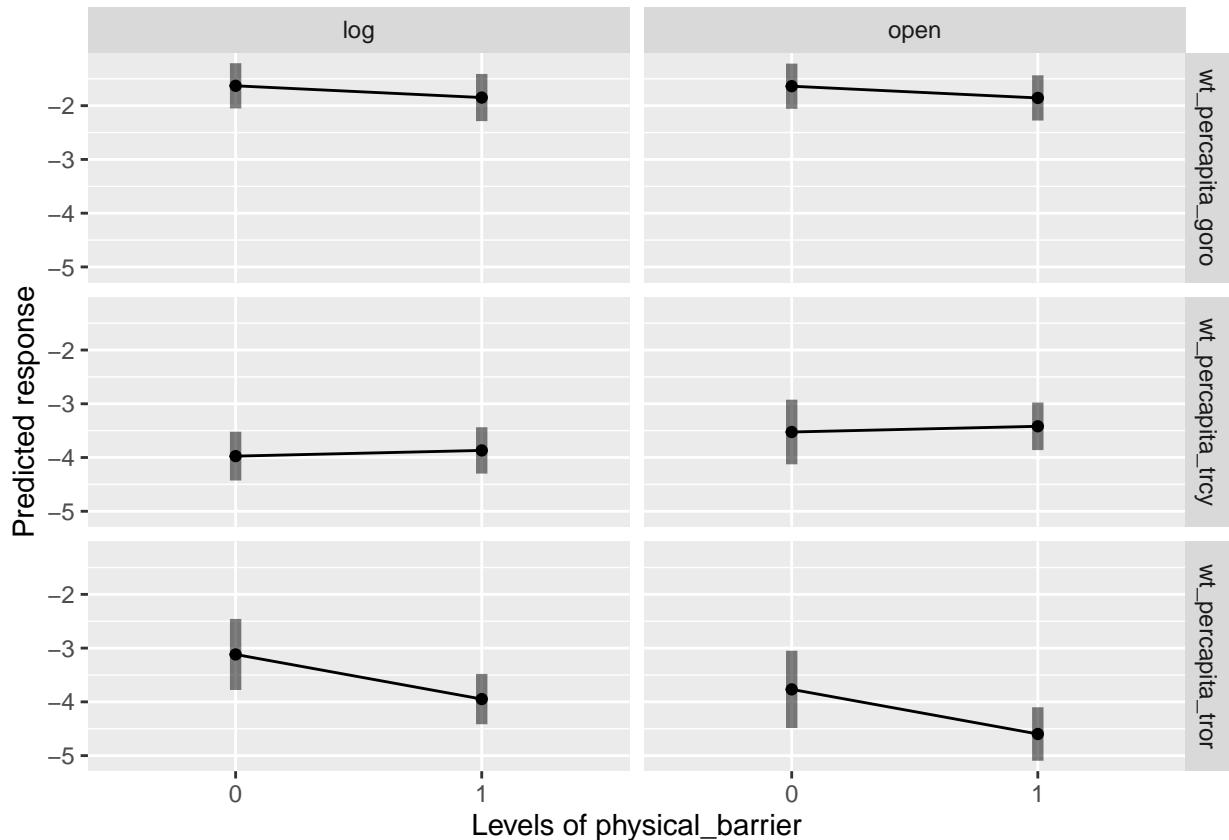
##   (183 observations deleted due to missingness)  

## Multiple R-squared: 0.5664, Adjusted R-squared: 0.5423  

## F-statistic: 23.51 on 8 and 144 DF, p-value: < 2.2e-16

```

```
emmip(pcwtmod_add, ~physical_barrier | name | initial, type='response', CI=T)
```



```
est<-emmeans(pcwtmod_add, ~initial | name, type='response')  
pairs(est)
```

```

## name = wt_percapita_goro:  

## contrast estimate SE df t.ratio p.value  

## log - open 0.00727 0.248 144 0.029 0.9767  

##  

## name = wt_percapita_trcy:

```

```

## contrast estimate SE df t.ratio p.value
## log - open -0.44862 0.282 144 -1.590 0.1140
##
## name = wt_percapita_tror:
## contrast estimate SE df t.ratio p.value
## log - open 0.64960 0.319 144 2.035 0.0437
##
## Results are averaged over the levels of: physical_barrier

##### 2022 candidate model comparison #####
## zeros
zero_candmods<-list("Plot type"=zerofit,
                      "Physical barrier"=zerofit_phys,
                      "Nutrient island"=zerofit_leg,
                      "Physical Barrier + Nutrient Island"=zerofit_add,
                      "Physical Barrier x Nutrient Island"=zerofit_intxn)
aictab(zero_candmods)

##
## Model selection based on AICc:
##
##                               K   AICc Delta_AICc AICcWt Cum.Wt      LL
## Physical barrier           7 430.60      0.00  0.50  0.50 -208.13
## Physical Barrier + Nutrient Island 10 430.84      0.24  0.44  0.94 -205.08
## Physical Barrier x Nutrient Island 13 436.10      5.49  0.03  0.98 -204.48
## Nutrient island            7 436.99      6.39  0.02  1.00 -211.33
## Plot type                  19 440.88     10.28  0.00  1.00 -200.24

# best fit model is a tie between physical barrier and physical barrier + nutrient island
# i will show the results for physical barrier + nutrient island.

# counts
count_candmods<-list("Plot type"=countfit,
                      "Physical barrier"=countfit_phys,
                      "Nutrient island"=countfit_leg,
                      "Physical Barrier + Nutrient Island"=countfit_add,
                      "Physical Barrier x Nutrient Island"=countfit_intxn)
aictab(count_candmods)

##
## Model selection based on AICc:
##
##                               K   AICc Delta_AICc AICcWt Cum.Wt      LL
## Nutrient island             8 791.95      0.00  0.58  0.58 -387.63
## Physical barrier            8 792.85      0.89  0.37  0.95 -388.08
## Physical Barrier + Nutrient Island 11 796.98      5.02  0.05  0.99 -386.84
## Physical Barrier x Nutrient Island 14 800.90      8.94  0.01  1.00 -385.41
## Plot type                  20 808.82     16.86  0.00  1.00 -382.26

# best fit model is nutrient island, tied with physical barrier.
# i will show the results for nutrient island.

```

```

pcwt_candmods<-list("Plot type"=pcwtmod,
                      "Physical barrier"=pcwtmod_phys,
                      "Nutrient island"=pcwtmod_leg,
                      "Physical Barrier + Nutrient Island" = pcwtmod_add,
                      "Physical Barrier x Nutrient Island"=pcwtmod_intxn)
aictab(pcwt_candmods)

##  

## Model selection based on AICc:  

##  

##  

##          K   AICc Delta_AICc AICcWt Cum.Wt      LL  

## Nutrient island        7  439.02      0.00  0.37  0.37 -212.12  

## Physical Barrier + Nutrient Island 10  439.33      0.31  0.31  0.68 -208.89  

## Physical barrier        7  439.48      0.46  0.29  0.97 -212.35  

## Physical Barrier x Nutrient Island 13  444.20      5.19  0.03  1.00 -207.79  

## Plot type               19  456.61     17.59  0.00  1.00 -206.45

# best fit model is nutrient island, tied with physical barrier + nutrient island, and then physical barrier x nutrient island. I will show the results for the additive model.

# tror is bigger when the legacy of the log and the log is still present

### figures ###
# best fit models - zeros
# colors
mimiscols<-c("#D66972","#108780")
zerofit_add<-glmmTMB(presence~name*physical_barrier+name*initial+(1 | block), family=binomial, data=data2)

zfit_est<-as.data.frame(emmeans(zerofit_add, ~initial|physical_barrier|name, type='response'))

pl4<-ggplot(zfit_est,aes(physical_barrier,prob,group=initial),)+  

  scale_color_manual(values=mimiscols)+  

  geom_point(aes(col=initial), size=2, position=position_dodge(width=0.5))+  

  geom_linerange(aes(ymin=asymp.LCL, ymax=asymp.UCL, col=initial), position=position_dodge(width=0.5))+  

  geom_line(aes(col=initial), position=position_dodge(width=0.5))+  

  facet_wrap(vars(name))+  

  theme_bw()+
  theme(strip.text.x = element_text(size=0),
        strip.background = element_blank(),
        axis.text=element_text(size=15),
        axis.title=element_text(size=20),
        legend.text=element_text(size=15),
        legend.title=element_text(size=15),
        legend.position="top")+
  xlab("Physical Barrier")+
  ylab("Occurrence")+
  geom_jitter(data=dat22,
              aes(x=physical_barrier, y=presence, color=initial),
              height=0.1,
              alpha=0.5)+  

  labs(color = "Initial Plot Type")
# pl4
summary(zerofit_add)

```

```

## Family: binomial ( logit )
## Formula:
## presence ~ name * physical_barrier + name * initial + (1 | block)
## Data: dat22
##
##      AIC      BIC  logLik deviance df.resid
##     430.2    468.3   -205.1     410.2      326
##
## Random effects:
##
## Conditional model:
## Groups Name            Variance Std.Dev.
## block  (Intercept) 0.2301   0.4797
## Number of obs: 336, groups: block, 7
##
## Conditional model:
##                               Estimate Std. Error z value Pr(>|z|)
## (Intercept)                 0.46829   0.39088  1.198  0.2309
## namentrcy_tot                0.38079   0.50971  0.747  0.4550
## namentror_tot               -0.26455   0.48701 -0.543  0.5870
## physical_barrier1            -0.02049   0.40340 -0.051  0.9595
## initialopen                  0.24376   0.40345  0.604  0.5457
## namentrcy_tot:physical_barrier1 1.24985   0.60881  2.053  0.0401 *
## namentror_tot:physical_barrier1  0.90138   0.57391  1.571  0.1163
## namentrcy_tot:initialopen     -1.23613   0.60711 -2.036  0.0417 *
## namentror_tot:initialopen     -0.58979   0.57314 -1.029  0.3035
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

pairs(emmeans(zeroefit_add, ~physical_barrier|name|initial))

## name = ngoro_tot, initial = log:
## contrast                           estimate   SE df z.ratio p.value
## physical_barrier0 - physical_barrier1  0.0205 0.403 Inf  0.051  0.9595
##
## name = ntrcy_tot, initial = log:
## contrast                           estimate   SE df z.ratio p.value
## physical_barrier0 - physical_barrier1 -1.2294 0.456 Inf -2.694  0.0071
##
## name = ntror_tot, initial = log:
## contrast                           estimate   SE df z.ratio p.value
## physical_barrier0 - physical_barrier1 -0.8809 0.408 Inf -2.156  0.0310
##
## name = ngoro_tot, initial = open:
## contrast                           estimate   SE df z.ratio p.value
## physical_barrier0 - physical_barrier1  0.0205 0.403 Inf  0.051  0.9595
##
## name = ntrcy_tot, initial = open:
## contrast                           estimate   SE df z.ratio p.value
## physical_barrier0 - physical_barrier1 -1.2294 0.456 Inf -2.694  0.0071
##
## name = ntror_tot, initial = open:
## contrast                           estimate   SE df z.ratio p.value
## physical_barrier0 - physical_barrier1 -0.8809 0.408 Inf -2.156  0.0310

```

```

##  

## Results are given on the log odds ratio (not the response) scale.  

# tror and trcy do better where there is a physical barrier as compared to when there is not (tror p=0.03)  

pairs(emmeans(zerofit_add, ~initial|name|physical_barrier))  

## name = ngoro_tot, physical_barrier = 0:  

## contrast estimate SE df z.ratio p.value  

## log - open -0.244 0.403 Inf -0.604 0.5457  

##  

## name = ntrcy_tot, physical_barrier = 0:  

## contrast estimate SE df z.ratio p.value  

## log - open 0.992 0.453 Inf 2.189 0.0286  

##  

## name = ntror_tot, physical_barrier = 0:  

## contrast estimate SE df z.ratio p.value  

## log - open 0.346 0.407 Inf 0.850 0.3951  

##  

## name = ngoro_tot, physical_barrier = 1:  

## contrast estimate SE df z.ratio p.value  

## log - open -0.244 0.403 Inf -0.604 0.5457  

##  

## name = ntrcy_tot, physical_barrier = 1:  

## contrast estimate SE df z.ratio p.value  

## log - open 0.992 0.453 Inf 2.189 0.0286  

##  

## name = ntror_tot, physical_barrier = 1:  

## contrast estimate SE df z.ratio p.value  

## log - open 0.346 0.407 Inf 0.850 0.3951  

##  

## Results are given on the log odds ratio (not the response) scale.  

# trcy does better with log initial as compared to open initial (p=0.03)  

countfit_leg<-glmmTMB(posicounts~name*initial+(1 | block), family=truncated_nbino2(), data=dat22, REML=F)  

cfit_est<-as.data.frame(emmeans(countfit_leg, ~initial|name, type='response'))  

pl1<-ggplot(cfit_est,aes(initial, response, group=1))+  

  geom_jitter(data=dat22,  

              aes(x=initial, y=posicounts),  

              width=0.1,  

              alpha=0.4,  

              color="gray")  

  geom_point(size=2, color="black")  

  geom_linerange(aes(ymin=asymp.LCL, ymax=asymp.UCL),color="black")  

  geom_line(color="black")  

  facet_wrap(vars(name))  

  theme_bw()  

  theme(strip.text.x = element_text(size=0),  

        strip.background = element_blank(),  

        axis.text=element_text(size=15),

```

```

    axis.title=element_text(size=20),
    legend.text=element_text(size=15),
    legend.title=element_text(size=15))+  

  xlab("Initial Plot Type")+
  ylab("Abundance")+
  labs(color = "Initial Plot \n Type")
pairs(emmeans(countfit_leg, ~initial|name))

```

```

## name = ngoro_tot:  

##   contrast   estimate     SE  df z.ratio p.value  

##   log - open    0.334  0.321 Inf   1.041  0.2979  

##  

## name = ntency_tot:  

##   contrast   estimate     SE  df z.ratio p.value  

##   log - open    0.288  0.317 Inf   0.908  0.3641  

##  

## name = ntror_tot:  

##   contrast   estimate     SE  df z.ratio p.value  

##   log - open   -0.120  0.303 Inf  -0.397  0.6917  

##  

## Results are given on the log (not the response) scale.

```

pl5

```

pcbfit_est<-as.data.frame(emmeans(pcwtmod_add, ~physical_barrier|name|initial, type='response'))  

pl6<-ggplot(pcbfit_est,aes(physical_barrier, emmean, group=initial))+  

  scale_color_manual(values=mimiscols)+  

  geom_point(aes(col=initial), size=2, position=position_dodge(width=0.5))+  

  geom_linerange(aes(ymin=lower.CL, ymax=upper.CL, col=initial), position=position_dodge(width=0.5))+  

  geom_line(aes(col=initial), position=position_dodge(width=0.5))+  

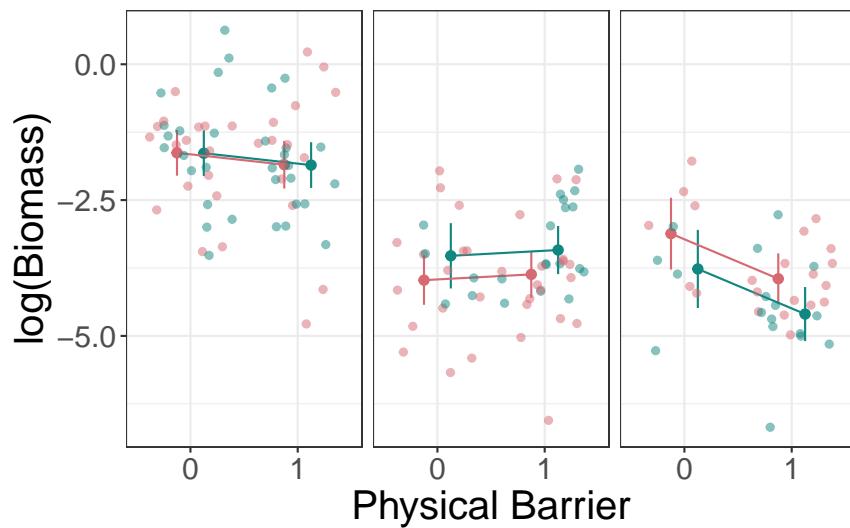
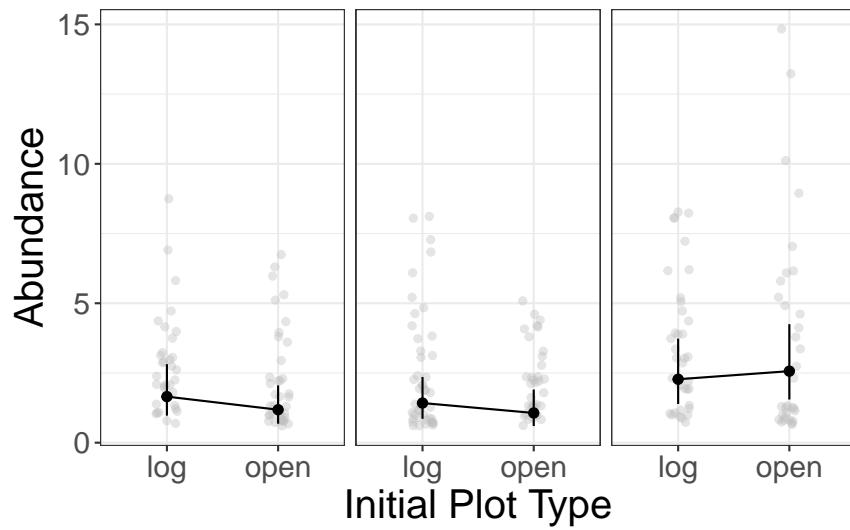
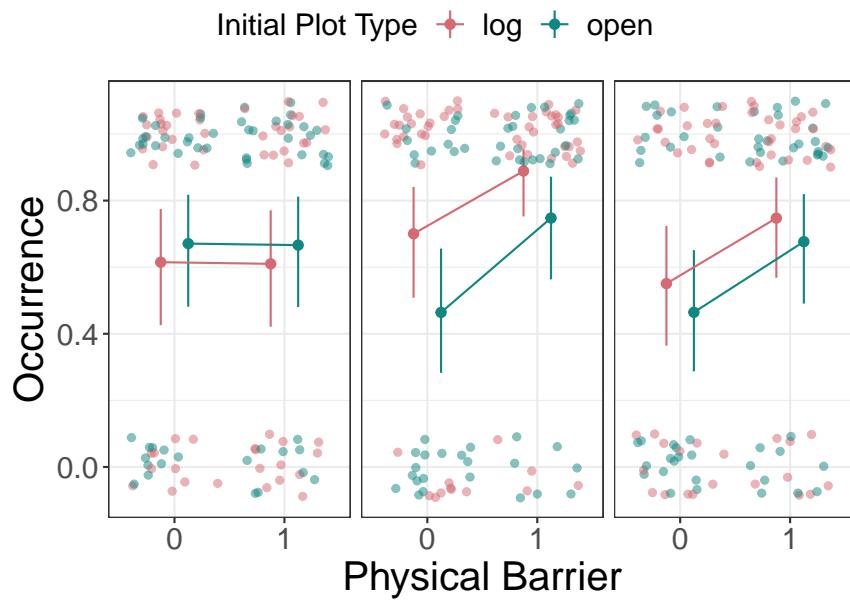
  facet_wrap(vars(name))+  

  theme_bw()+
  theme(strip.text.x = element_text(size=0),
        strip.background = element_blank(),
        axis.text=element_text(size=15),
        axis.title=element_text(size=20),
        legend.text=element_text(size=15),
        legend.title=element_text(size=15),
        legend.position="top")+
  xlab("Physical Barrier")+
  ylab("log(Biomass)")+
  geom_jitter(data=pcwtdata,
              aes(x=physical_barrier, y=log_wt, color=initial),
              height=0.1,
              alpha=0.5)+  

  labs(color = "Initial Plot Type")
#pl6

```

```
ggarrange(pl4, pl5, pl6, ncol=1, common.legend = T)
```



Interpretation