Ranas\_compare

S. Holden Jones

3/6/24

# COMPARE RICHNESS, ABUNDANCE, DIVERSITY, COMPOSITION ACROSS LANDUSE TYPES

This script will take processed data as input, run statistic tests and produce figures comparing the dependent variables (diversity, abundance, composition) of each landuse type broadly. Other scripts will explore the variation in these response variables w/in specific landuse types.

load packages

load data

define a consistent palet, labels, and theme to be used in figures ## still would be nice to have palet for harvested comparison

# palet for comparisons of all 4 site types  
pal <- brewer.pal(4, "BrBG")  
  
# site labels for 4 site comparison, use in several plots  
site\_labels <- c("Shade \n (n = 12)", "Sun \n (n = 11)",   
 "Abandoned \n (n = 4)", "Forest \n (n = 5)")  
  
# site labels for harvest, non-harvest comparison  
labels\_harvest <- c("Not harvested \n (n = 9)", "Harvested \n (n = 23)")  
  
# set a theme for ggplots  
my\_theme <- function() {  
 theme\_minimal() +   
 theme(legend.position = "none",  
 plot.background = element\_rect("white"),  
 panel.background = element\_rect("white"),  
 panel.grid = element\_line("grey90"),  
 axis.line = element\_line("gray25"),  
 axis.text = element\_text(size = 12, color = "gray25"),  
 axis.title = element\_text(size = 14, color = "gray25"),  
 legend.text = element\_text(size = 12))  
}

# RICHNESS ACROSS LANDUSE TYPES

## 31 total species

new %>%  
 distinct(Final\_ID) %>%  
 count()

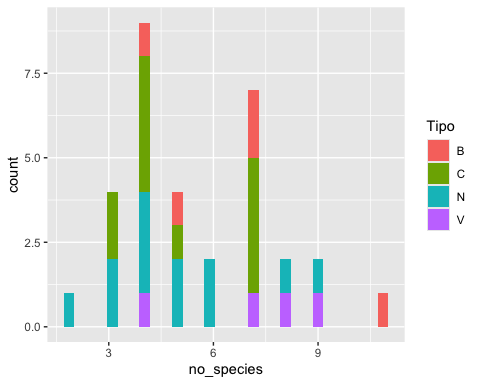
## # A tibble: 1 × 1  
## n  
## <int>  
## 1 31

## before selecting statistical test, need to look at dist. of richness by site

* use Poisson dist. b/c using count data with no upper limit!

(richness\_by\_site\_hist <- ggplot(richness\_by\_site, aes(no\_species, fill = Tipo)) +  
 geom\_histogram())

## `stat\_bin()` using `bins = 30`. Pick better value with `binwidth`.



## run glm for species richness across landuse types, using Poisson dist.

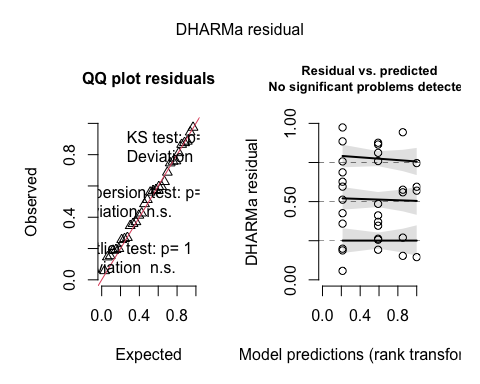
## NO SIG DIFF IN RICHNESS ACROSS LANDUSE TYPES

* bosque is sig diff from nacional w/out elevation as random fact - viejo almost
* bosque and viejo very close to being significant - need final IDs back!
* using poisson dist in glm - DHARMa simulation shows that it works well

richness\_by\_type\_glm <- glm(no\_species ~ Tipo + (1|Elevation),   
 data = richness\_by\_site,   
 family = poisson(link = "log"))  
summary(richness\_by\_type\_glm)

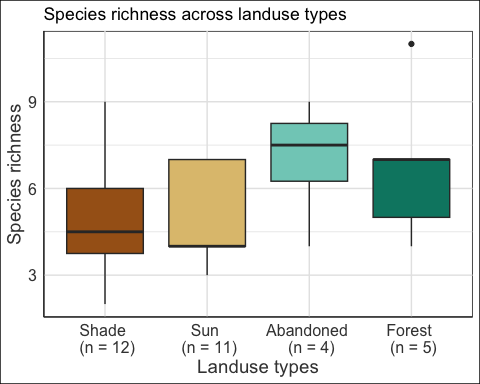
##   
## Call:  
## glm(formula = no\_species ~ Tipo + (1 | Elevation), family = poisson(link = "log"),   
## data = richness\_by\_site)  
##   
## Coefficients: (1 not defined because of singularities)  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) 1.91692 0.17150 11.177 <2e-16 \*\*\*  
## TipoC -0.30748 0.21816 -1.409 0.159   
## TipoN -0.32429 0.21532 -1.506 0.132   
## TipoV 0.02899 0.25520 0.114 0.910   
## 1 | ElevationTRUE NA NA NA NA   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for poisson family taken to be 1)  
##   
## Null deviance: 25.356 on 31 degrees of freedom  
## Residual deviance: 21.137 on 28 degrees of freedom  
## AIC: 141.23  
##   
## Number of Fisher Scoring iterations: 4

# here use DHARMa package to see if we're using the correct model  
# no significant problems detected!  
simulationOutput\_richness\_by\_type <- simulateResiduals(  
 fittedModel = richness\_by\_type\_glm)  
plot(simulationOutput\_richness\_by\_type)



plot species richness across landuse types

# reorder factor levels  
richness\_by\_site$Tipo <- factor(richness\_by\_site$Tipo,  
 c("N", "C", "V", "B"))  
  
# plot species richness across landuse types using palet, my\_theme  
plot\_richness\_by\_site <- ggplot(richness\_by\_site,   
 aes(x = Tipo, y = no\_species, fill = Tipo)) +  
 geom\_boxplot() +  
 scale\_fill\_manual(values = pal) +  
 # order needs to match factor levels above  
 scale\_x\_discrete(labels = c(site\_labels)) +  
 my\_theme() +   
 labs(x = "Landuse types",  
 y = "Species richness",  
 title = "Species richness across landuse types")  
plot\_richness\_by\_site



# RICHNESS ACROSS HARVESTED UNHARVESTED

## run glm for harvest on species richness using Poisson dist

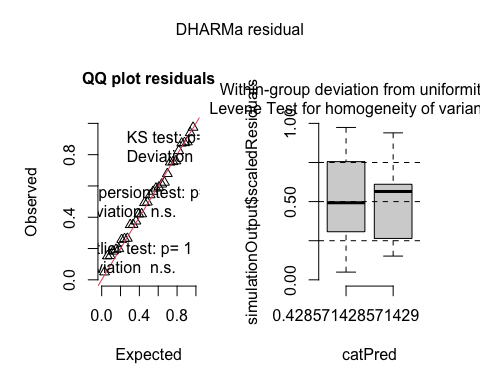
## SIG HIGHER RICHNESS IN NON\_HARVESTED SITES

higher richness in non-harvested sites - using Poisson dist - including elevation as random effect in glm

richness\_by\_harvest\_glm <- glm(no\_species ~ harvested + (1|Elevation),   
 data = richness\_by\_site,   
 family = poisson(link = "log"))  
summary(richness\_by\_harvest\_glm)

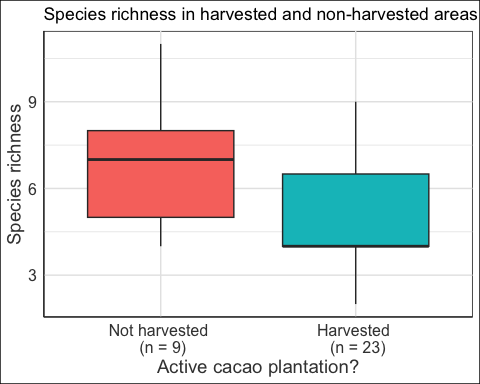
##   
## Call:  
## glm(formula = no\_species ~ harvested + (1 | Elevation), family = poisson(link = "log"),   
## data = richness\_by\_site)  
##   
## Coefficients: (1 not defined because of singularities)  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) 1.9299 0.1270 15.196 <2e-16 \*\*\*  
## harvestedYes -0.3292 0.1578 -2.086 0.037 \*   
## 1 | ElevationTRUE NA NA NA NA   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for poisson family taken to be 1)  
##   
## Null deviance: 25.356 on 31 degrees of freedom  
## Residual deviance: 21.158 on 30 degrees of freedom  
## AIC: 137.26  
##   
## Number of Fisher Scoring iterations: 4

# within group deviation from uniformity n.s.  
# Levene Test for homogeneity of variance n.s.  
simulationOutput\_richness\_by\_harvest <- simulateResiduals(  
 fittedModel = richness\_by\_harvest\_glm)  
plot(simulationOutput\_richness\_by\_harvest)



B+V have sig higher richness than N+C ## aesthetically would want to mix colors from previous plot

plot\_richness\_by\_harvest <- ggplot(richness\_by\_site,   
 aes(x = harvested,   
 y = no\_species, fill = harvested)) +  
 scale\_x\_discrete(labels = c(labels\_harvest)) +  
 geom\_boxplot() +  
 my\_theme() +  
 labs(x = "Active cacao plantation?", y = "Species richness") +  
 ggtitle("Species richness in harvested and non-harvested areas")  
plot\_richness\_by\_harvest



# RAREFIED RICHNESS ACROSS LANDUSE TYPES

need to go through rarefy\_scrap to piece this piece together - use Chao iNEXT package

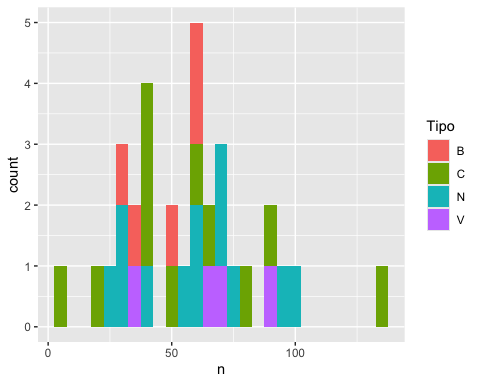
# ABUNDANCE ACROSS LANDUSE TYPES

here just simply number of new individuals - could eventually complicate this to look at pop size estimates?

## before selecting statistical test, need to look at dist. of abundance by site

* use Poisson dist. b/c using count data with no upper limit!

(abundance\_by\_site\_hist <- ggplot(abundance\_by\_site, aes(n, fill = Tipo)) +  
 geom\_histogram(binwidth = 5))



## run glm for abundance across landuse types, using Gaussian dist. Elev rand eff

DONT RUN POISSON HERE! - quantile deviations detected - combined adjusted quantile test significant

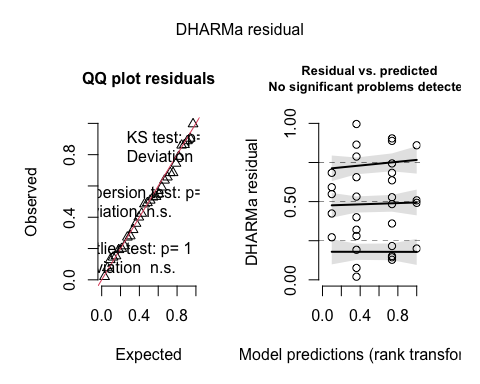
## NO SIG DIFF IN ABUNDANCE ACROSS LANDUSE TYPES

* elevation random effect

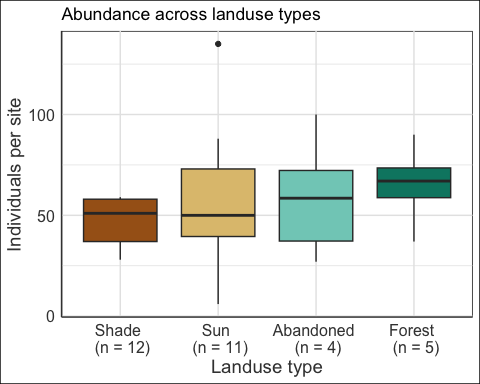
# default family dist is Gaussian  
abundance\_by\_type\_glm <- glm(n ~ Tipo + (1|Elevation), data = abundance\_by\_site)  
summary(abundance\_by\_type\_glm)

##   
## Call:  
## glm(formula = n ~ Tipo + (1 | Elevation), data = abundance\_by\_site)  
##   
## Coefficients: (1 not defined because of singularities)  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 46.60 12.23 3.809 0.0007 \*\*\*  
## TipoC 10.49 14.76 0.711 0.4830   
## TipoN 12.48 14.56 0.857 0.3986   
## TipoV 18.65 18.35 1.016 0.3182   
## 1 | ElevationTRUE NA NA NA NA   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for gaussian family taken to be 748.4206)  
##   
## Null deviance: 21819 on 31 degrees of freedom  
## Residual deviance: 20956 on 28 degrees of freedom  
## AIC: 308.31  
##   
## Number of Fisher Scoring iterations: 2

# no significant problems detected w/Gaussian!  
simulationOutput\_abundance\_by\_type <- simulateResiduals(  
 fittedModel = abundance\_by\_type\_glm)  
plot(simulationOutput\_abundance\_by\_type)



plot\_abundance\_by\_site <- ggplot(abundance\_by\_site, aes(x = Tipo, y = n, fill = Tipo)) +  
 geom\_boxplot() +  
 scale\_fill\_manual(values = pal) +  
 scale\_x\_discrete(labels = c(site\_labels)) +  
 my\_theme() +  
 labs(x = "Landuse type",  
 y = "Individuals per site",  
 title = "Abundance across landuse types")  
plot\_abundance\_by\_site

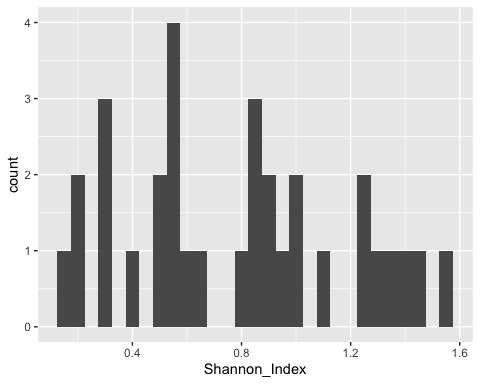


# DIVERSITY ACROSS LANDUSE TYPES

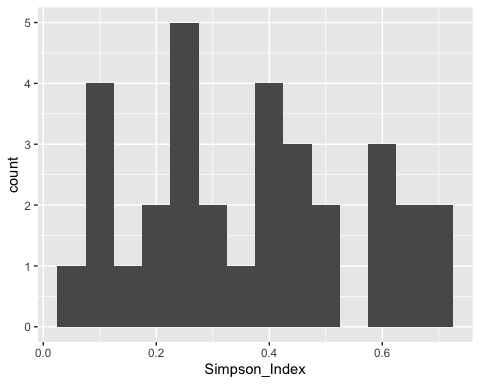
Look at how different diversity metrics compare across landuse types

## look at dist of diversity metrics

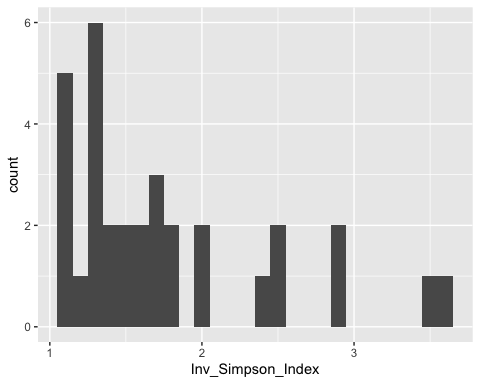
# shannon - clearly not normal dist!   
(shannon\_hist <- ggplot(diversity\_by\_site, aes(Shannon\_Index)) +  
 geom\_histogram(binwidth = .05))



# simpson - clearly not normal dist!   
(simpson\_hist <- ggplot(diversity\_by\_site, aes(Simpson\_Index)) +  
 geom\_histogram(binwidth = .05))



# inv\_simpson - clearly not normal dist!   
(inv\_simpson\_hist <- ggplot(diversity\_by\_site, aes(Inv\_Simpson\_Index)) +  
 geom\_histogram(binwidth = .1))



## use Gaussian dist for Shannon, Simpson, inv\_Gaussian for Inv\_Simpson

Thought Poisson dist would work for these b/c not normal and from count data - woah! sig deviation for Poisson dist. - quantile deviations detected - combined adjusted quantile test sig

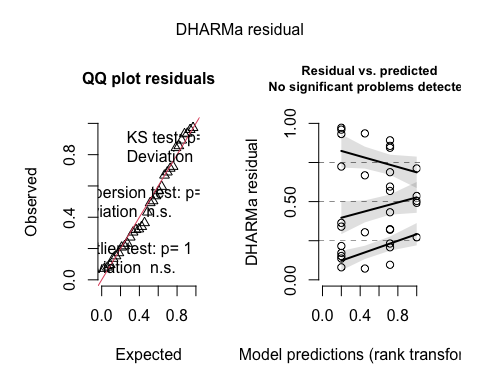
## SIG HIGHER SHANNON DIV IN BOSQUE

significantly higher shannon div in Bosque vs. other sites - viejo is close

# shannon w/ Gaussian dist - all good  
shannon\_by\_type\_glm <- glm(Shannon\_Index ~ Tipo + (1|Elevation),   
 data = diversity\_by\_site)  
summary(shannon\_by\_type\_glm)

##   
## Call:  
## glm(formula = Shannon\_Index ~ Tipo + (1 | Elevation), data = diversity\_by\_site)  
##   
## Coefficients: (1 not defined because of singularities)  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 0.85213 0.18659 4.567 9.06e-05 \*\*\*  
## TipoC -0.14847 0.22504 -0.660 0.515   
## TipoN -0.02145 0.22209 -0.097 0.924   
## TipoV -0.05557 0.27989 -0.199 0.844   
## 1 | ElevationTRUE NA NA NA NA   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for gaussian family taken to be 0.1740863)  
##   
## Null deviance: 4.9953 on 31 degrees of freedom  
## Residual deviance: 4.8744 on 28 degrees of freedom  
## AIC: 40.597  
##   
## Number of Fisher Scoring iterations: 2

# no significant problems detected using Gaussian dist!  
simulationOutput\_shannon\_by\_type <- simulateResiduals(  
 fittedModel = shannon\_by\_type\_glm)  
plot(simulationOutput\_shannon\_by\_type)



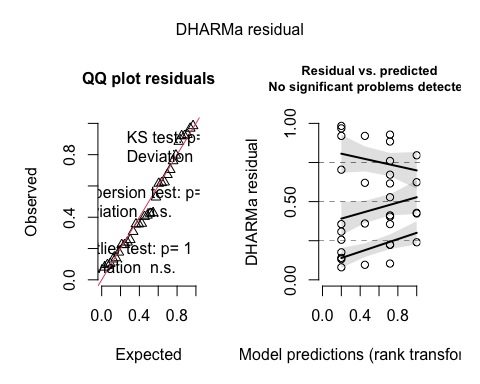
## SIG HIGHER SIMPSON DIV IN BOSQUE

sig higher simpson div in Bosque vs. other sites - viejo not particularly close

# simpson w/ Gaussian dist  
simpson\_by\_type\_glm <- glm(Simpson\_Index ~ Tipo + (1|Elevation),   
 data = diversity\_by\_site)  
summary(simpson\_by\_type\_glm)

##   
## Call:  
## glm(formula = Simpson\_Index ~ Tipo + (1 | Elevation), data = diversity\_by\_site)  
##   
## Coefficients: (1 not defined because of singularities)  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 0.42998 0.09015 4.770 5.2e-05 \*\*\*  
## TipoC -0.10562 0.10872 -0.971 0.340   
## TipoN -0.05759 0.10730 -0.537 0.596   
## TipoV -0.07127 0.13522 -0.527 0.602   
## 1 | ElevationTRUE NA NA NA NA   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for gaussian family taken to be 0.04063234)  
##   
## Null deviance: 1.1777 on 31 degrees of freedom  
## Residual deviance: 1.1377 on 28 degrees of freedom  
## AIC: -5.9631  
##   
## Number of Fisher Scoring iterations: 2

# no significant problems detected using Gaussian dist!  
simulationOutput\_simpson\_by\_type <- simulateResiduals(  
 fittedModel = simpson\_by\_type\_glm)  
plot(simulationOutput\_simpson\_by\_type)



## NO SIG DIFF IN INV\_SIMPSON DIV ACROSS LANDUSE TYPES

no sig difference in inv\_simpson diversity across landtypes - Bosque .18, viejo not close

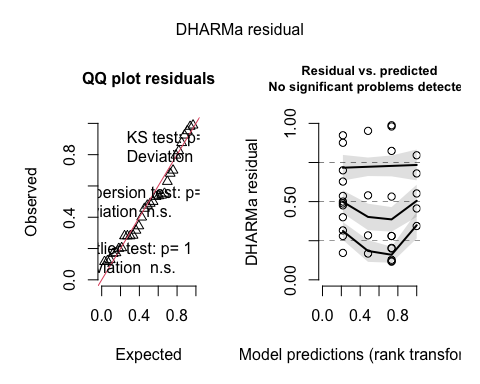
NOTE THAT HERE USING INVERSE GAUSSIAN DIST - Gaussian - quantile deviations detected - comb adjust - quant test sig - Binomial - error - y values must be b/ween 0:1 - Gamma - deviation significant w/ outlier test but better than others - Inv Gaussian - works! need to load additional cran package SuppDists

# inv\_simpson w/ inverse gaussian dist.  
inv\_simpson\_by\_type\_glm <- glm(Inv\_Simpson\_Index ~ Tipo + (1|Elevation),   
 data = diversity\_by\_site,  
 family = inverse.gaussian(link = "1/mu^2"))  
summary(inv\_simpson\_by\_type\_glm)

##   
## Call:  
## glm(formula = Inv\_Simpson\_Index ~ Tipo + (1 | Elevation), family = inverse.gaussian(link = "1/mu^2"),   
## data = diversity\_by\_site)  
##   
## Coefficients: (1 not defined because of singularities)  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 0.30010 0.11140 2.694 0.0118 \*  
## TipoC 0.01148 0.13556 0.085 0.9331   
## TipoN 0.04273 0.13683 0.312 0.7571   
## TipoV 0.01610 0.17084 0.094 0.9256   
## 1 | ElevationTRUE NA NA NA NA   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for inverse.gaussian family taken to be 0.09435451)  
##   
## Null deviance: 2.2359 on 31 degrees of freedom  
## Residual deviance: 2.2238 on 28 degrees of freedom  
## AIC: 63.92  
##   
## Number of Fisher Scoring iterations: 6

# no significant problems detected using inverse Gaussian dist!  
# all other tested dist threw errors w/ DHARMa  
simulationOutput\_inv\_simpson\_by\_type <- simulateResiduals(  
 fittedModel = inv\_simpson\_by\_type\_glm)  
plot(simulationOutput\_inv\_simpson\_by\_type)

## Warning in newton(lsp = lsp, X = G$X, y = G$y, Eb = G$Eb, UrS = G$UrS, L = G$L,  
## : Fitting terminated with step failure - check results carefully

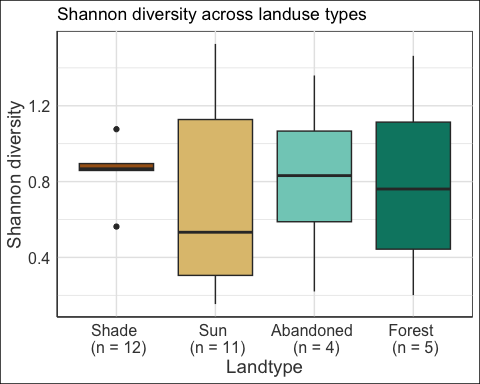


## plot how diversity indices vary across land types

* these are pretty much identical regardless of index used

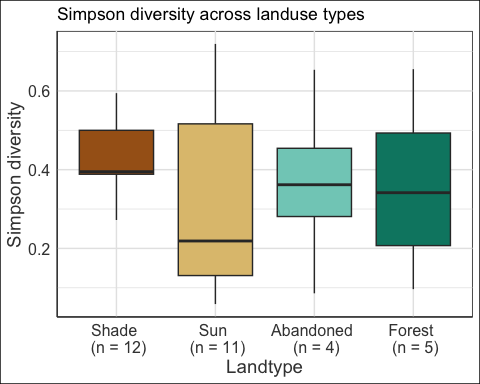
shannon

plot\_shannon <- ggplot(diversity\_by\_site,   
 aes(x = Tipo, y = Shannon\_Index, fill = Tipo)) +  
 geom\_boxplot() +  
 scale\_fill\_manual(values = pal) +  
 scale\_x\_discrete(labels = c(site\_labels)) +  
 my\_theme() +  
 labs(x = "Landtype",  
 y = "Shannon diversity",  
 title = "Shannon diversity across landuse types")  
plot\_shannon



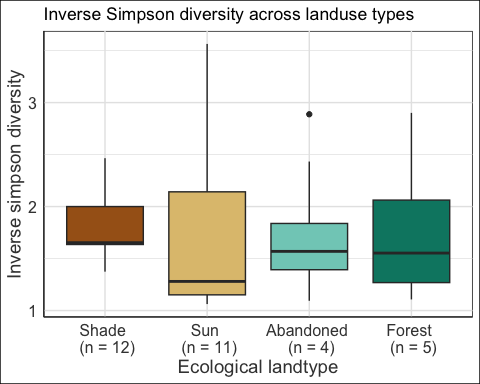
simpson

plot\_simpson <- ggplot(diversity\_by\_site,   
 aes(x = Tipo, y = Simpson\_Index, fill = Tipo)) +  
 geom\_boxplot() +  
 scale\_fill\_manual(values = pal) +  
 scale\_x\_discrete(labels = c(site\_labels)) +  
 my\_theme() +   
 labs(x = "Landtype",  
 y = "Simpson diversity",  
 title = "Simpson diversity across landuse types")  
plot\_simpson



inv simpson

plot\_inv\_simpson <- ggplot(diversity\_by\_site,   
 aes(x = Tipo, y = Inv\_Simpson\_Index, fill = Tipo)) +  
 geom\_boxplot() +  
 scale\_fill\_manual(values = pal) +  
 scale\_x\_discrete(labels = c(site\_labels)) +  
 my\_theme() +  
 labs(x = "Ecological landtype",  
 y = "Inverse simpson diversity",  
 title = "Inverse Simpson diversity across landuse types")  
plot\_inv\_simpson



# DIVERSITY ACROSS HARVEST, UNHARVESTED

do the same thing as above, compare each index using harvested, unharvested column - glms across harvested, unharvested

## use Gaussian dist for Shannon, Simpson, Inv\_Simpson

NOTE THAT HERE THE GAUSSIAN DIST WORKS FINE FOR INV\_SIMPSON

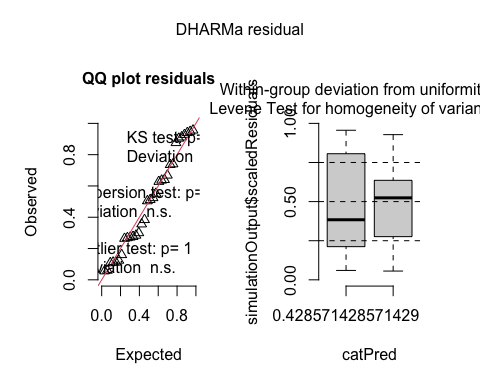
## SIG HIGHER SHANNON DIV IN NON-HARVESTED SITES

check w/ interpretation of this glm summary

# using Gaussian dist.  
shannon\_harvest\_glm <- glm(Shannon\_Index ~ harvested + (1|Elevation),   
 data = diversity\_by\_site)  
summary(shannon\_harvest\_glm)

##   
## Call:  
## glm(formula = Shannon\_Index ~ harvested + (1 | Elevation), data = diversity\_by\_site)  
##   
## Coefficients: (1 not defined because of singularities)  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 0.8274 0.1357 6.096 1.07e-06 \*\*\*  
## harvestedYes -0.0575 0.1601 -0.359 0.722   
## 1 | ElevationTRUE NA NA NA NA   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for gaussian family taken to be 0.1657961)  
##   
## Null deviance: 4.9953 on 31 degrees of freedom  
## Residual deviance: 4.9739 on 30 degrees of freedom  
## AIC: 37.243  
##   
## Number of Fisher Scoring iterations: 2

# no issues with Gaussian dist  
simulationOutput\_shannon\_harvest <- simulateResiduals(  
 fittedModel = shannon\_harvest\_glm)  
plot(simulationOutput\_shannon\_harvest)



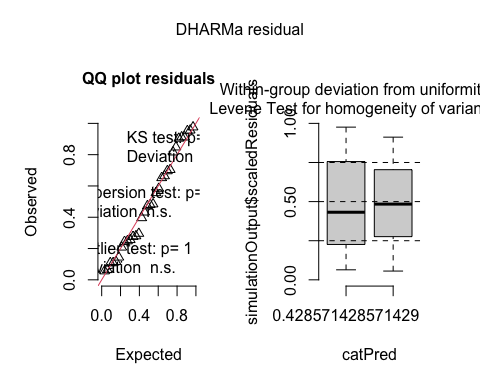
## SIG HIGHER SIMPSON DIV IN NON-HARVESTED SITES

check interpretation of this glm summary

# using Gaussian dist.  
simpson\_harvest\_glm <- glm(Simpson\_Index ~ harvested + (1|Elevation),   
 data = diversity\_by\_site)  
summary(simpson\_harvest\_glm)

##   
## Call:  
## glm(formula = Simpson\_Index ~ harvested + (1 | Elevation), data = diversity\_by\_site)  
##   
## Coefficients: (1 not defined because of singularities)  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 0.39831 0.06561 6.071 1.14e-06 \*\*\*  
## harvestedYes -0.04889 0.07739 -0.632 0.532   
## 1 | ElevationTRUE NA NA NA NA   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for gaussian family taken to be 0.03874104)  
##   
## Null deviance: 1.1777 on 31 degrees of freedom  
## Residual deviance: 1.1622 on 30 degrees of freedom  
## AIC: -9.2806  
##   
## Number of Fisher Scoring iterations: 2

# no issues with Gaussian dist  
simulationOutput\_simpson\_harvest <- simulateResiduals(  
 fittedModel = simpson\_harvest\_glm)  
plot(simulationOutput\_simpson\_harvest)



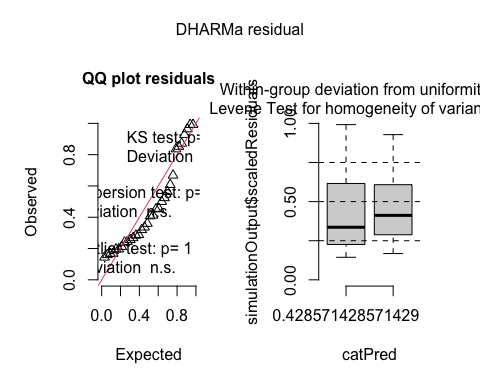
## NO SIG DIFF IN IN\_SIMPSON ACROSS HARVEST, NON-HARVEST

check interpretation of this glm summary

# using Gaussian dist.  
inv\_simpson\_harvest\_glm <- glm(Inv\_Simpson\_Index ~ harvested + (1|Elevation),   
 data = diversity\_by\_site)  
summary(inv\_simpson\_harvest\_glm)

##   
## Call:  
## glm(formula = Inv\_Simpson\_Index ~ harvested + (1 | Elevation),   
## data = diversity\_by\_site)  
##   
## Coefficients: (1 not defined because of singularities)  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 1.80450 0.23399 7.712 1.33e-08 \*\*\*  
## harvestedYes -0.05663 0.27600 -0.205 0.839   
## 1 | ElevationTRUE NA NA NA NA   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for gaussian family taken to be 0.4927601)  
##   
## Null deviance: 14.804 on 31 degrees of freedom  
## Residual deviance: 14.783 on 30 degrees of freedom  
## AIC: 72.099  
##   
## Number of Fisher Scoring iterations: 2

# no issues with Gaussian dist  
simulationOutput\_inv\_simpson\_harvest <- simulateResiduals(  
 fittedModel = inv\_simpson\_harvest\_glm)  
plot(simulationOutput\_inv\_simpson\_harvest)

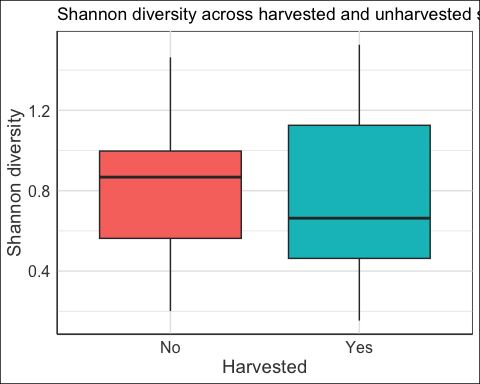


## make figures for each

* aesthetically think about fill colors

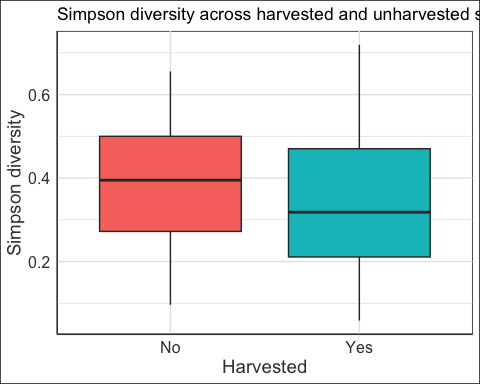
shannon\_harvest

plot\_shannon\_harvest <- ggplot(diversity\_by\_site,   
 aes(x = harvested, y = Shannon\_Index,   
 fill = harvested)) +  
 geom\_boxplot() +  
 my\_theme() +  
 labs(x = "Harvested",  
 y = "Shannon diversity",  
 title = "Shannon diversity across harvested and unharvested sites")  
plot\_shannon\_harvest



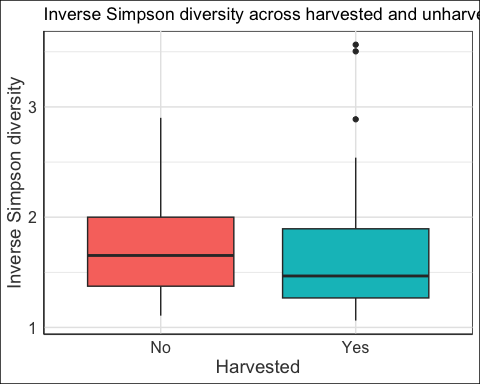
simpson\_harvest

plot\_simpson\_harvest <- ggplot(diversity\_by\_site,   
 aes(x = harvested, y = Simpson\_Index,   
 fill = harvested)) +  
 geom\_boxplot() +  
 my\_theme() +  
 labs(x = "Harvested",  
 y = "Simpson diversity",  
 title = "Simpson diversity across harvested and unharvested sites")  
plot\_simpson\_harvest



inv\_simpson\_harvest

plot\_inv\_simpson\_harvest <- ggplot(diversity\_by\_site,   
 aes(x = harvested, y = Inv\_Simpson\_Index,   
 fill = harvested)) +  
 geom\_boxplot() +  
 my\_theme() +  
 labs(x = "Harvested",  
 y = "Inverse Simpson diversity",  
 title = "Inverse Simpson diversity across harvested and unharvested sites")  
plot\_inv\_simpson\_harvest



# COMPOSITION ACROSS LANDUSE TYPES

Variety of ways to look at this, can’t do all of them! - NMDS - more for visualizing differences - start here - Bray-Curtiss Dissimilarity Index - quantifies dissimilarity - 0:1 (same:diff) - this one preferred b/c incorporates species abundances - Jaccard Similarity Index - doesn’t incorporate species abundances - Analysis of similarities for statistics

## NEED TO GET THIS CODE REVIEWED - NOT SURE IF CORRECT

## create new\_mat matrix

new\_mat - three column format to use in labdsv package - reorder rows to allow NMDS to draw by site type - creating here b/c was running into row names issue importing from processing script

three\_column <- new %>%  
 group\_by(Sitio) %>%  
 count(`Final\_ID`)  
  
# needs to be converted to data.frame!  
three\_column <- data.frame(three\_column)  
  
new\_mat <- matrify(three\_column)  
  
# Define the desired order of rows: Shade, Sun, Abandoned, Forest  
new\_order <- c(1,3,5,8,19,20,22,25,27,29,30,32, # shade (n=12)  
 2,4,7,16,18,21,23,24,26,28,31, # sun (n=11)  
 9,13,14,17, # abandoned (n=4)  
 6,10,11,12,15 # forest (n=5)  
 )  
  
# Reorder rows  
new\_mat <- new\_mat[new\_order, ]

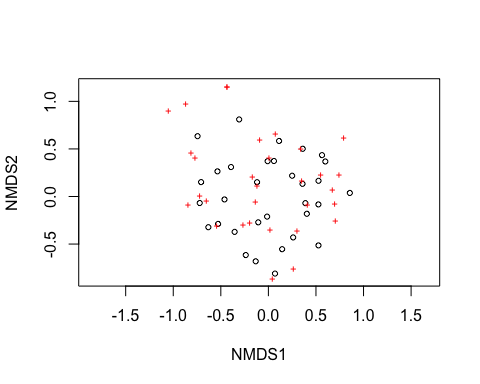
## create nmds plot

* draw boundaries of landuse types

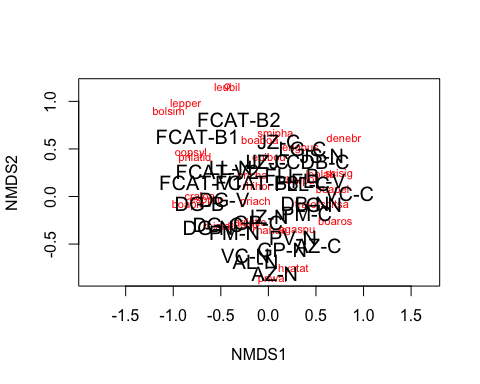
# assign treatment levels per landuse type on reordered new\_mat  
landuse\_treatments=c(rep("Shade",12),  
 rep("Sun",11),   
 rep("Abandoned",4),   
 rep("Forest",5))  
  
# use community by species matrix - stress = 0.26 - okay representation  
new\_mat\_NMDS = metaMDS(new\_mat, k = 2)

## Square root transformation  
## Wisconsin double standardization  
## Run 0 stress 0.2572911   
## Run 1 stress 0.2882819   
## Run 2 stress 0.2643339   
## Run 3 stress 0.2561223   
## ... New best solution  
## ... Procrustes: rmse 0.1052352 max resid 0.2453518   
## Run 4 stress 0.2568899   
## Run 5 stress 0.2635971   
## Run 6 stress 0.2607069   
## Run 7 stress 0.2587171   
## Run 8 stress 0.2607069   
## Run 9 stress 0.2572911   
## Run 10 stress 0.2992853   
## Run 11 stress 0.258717   
## Run 12 stress 0.272841   
## Run 13 stress 0.277447   
## Run 14 stress 0.2637561   
## Run 15 stress 0.2973355   
## Run 16 stress 0.2561223   
## ... Procrustes: rmse 0.0001129971 max resid 0.0004274693   
## ... Similar to previous best  
## Run 17 stress 0.257291   
## Run 18 stress 0.2763257   
## Run 19 stress 0.2554318   
## ... New best solution  
## ... Procrustes: rmse 0.03482019 max resid 0.1429769   
## Run 20 stress 0.2806868   
## \*\*\* Best solution was not repeated -- monoMDS stopping criteria:  
## 20: stress ratio > sratmax

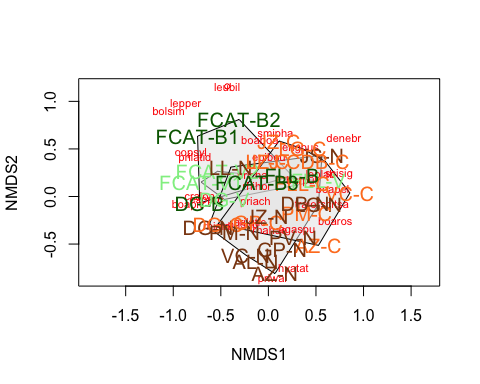
# sites are open circles, species are red crosses  
plot(new\_mat\_NMDS)



# help visualize by labeling specific sites and species  
  
# ordination plot function especially for congested plots  
ordiplot(new\_mat\_NMDS,type="n")  
# this function adds text or points to ordination plots  
orditorp(new\_mat\_NMDS,display="species",col="red",air=0.01)  
  
orditorp(new\_mat\_NMDS,display="sites",cex=1.25,air=0.01)



ordiplot(new\_mat\_NMDS,type="n")  
  
ordihull(new\_mat\_NMDS,groups=landuse\_treatments,draw="polygon",  
 col="grey90",label=F)  
  
orditorp(new\_mat\_NMDS,display="species",col="red",air=0.01)  
  
# here colors need to align with landuse\_treatments from above!  
# of course need to use chocolate!  
orditorp(new\_mat\_NMDS,display="sites",col=c(rep("chocolate4",12),  
 rep("chocolate1",11),  
 rep("lightgreen",4),  
 rep("darkgreen",5)),   
 air=0.01,cex=1.25)



# INCORPORATE iNEXT PACKAGE

not sure how to interpret this data

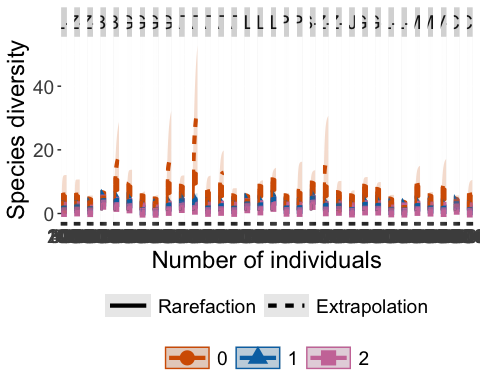
## look at rarefaction curves of all sites at once

all sites - pretty crazy

# just forest sites:  
  
# okay, so it seems like it thinks that species are sites and sites are species  
# can we rotate the dataframe?  
new\_mat\_test <- new\_mat %>%   
 rotate\_df()  
  
  
out <- iNEXT(new\_mat\_test, q=c(0, 1, 2), datatype="abundance", endpoint=500)

## Warning in matrix(apply(Abun.Mat, 2, function(x) invChat.Ind(x, q, goalSC)$qD),  
## : data length [1500] is not a sub-multiple or multiple of the number of rows  
## [27]

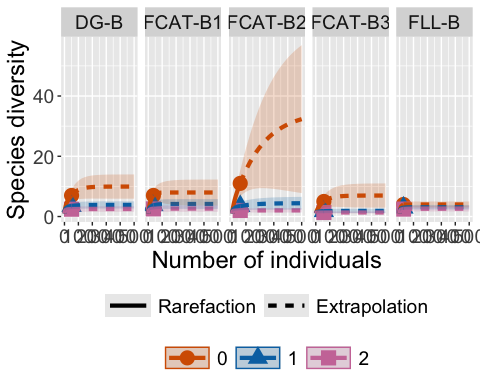
# Sample-size-based R/E curves, separating plots by "Assemblage"  
ggiNEXT(out, type=1, facet.var="Assemblage")



## unsightly! too many sites at once - go one site type at a time

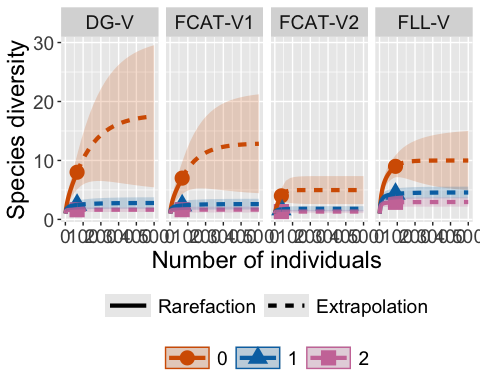
try subsetting new\_mat into only forest sites for example - may be easier to visualize

# just forest sites:  
forest\_mat <- new\_mat[28:32,]  
  
# okay, so it seems like it thinks that species are sites and sites are species  
# can we rotate the dataframe?  
forest\_mat <- forest\_mat %>%   
 rotate\_df()  
  
  
out <- iNEXT(forest\_mat, q=c(0, 1, 2), datatype="abundance", endpoint=500)  
# Sample-size-based R/E curves, separating plots by "Assemblage"  
ggiNEXT(out, type=1, facet.var="Assemblage")



how about just abandoned sites?

# just forest sites:  
abandoned\_mat <- new\_mat[24:27,]  
  
# okay, so it seems like it thinks that species are sites and sites are species  
# can we rotate the dataframe?  
abandoned\_mat <- abandoned\_mat %>%   
 rotate\_df()  
  
  
out <- iNEXT(abandoned\_mat, q=c(0, 1, 2), datatype="abundance", endpoint=500)  
# Sample-size-based R/E curves, separating plots by "Assemblage"  
ggiNEXT(out, type=1, facet.var="Assemblage")

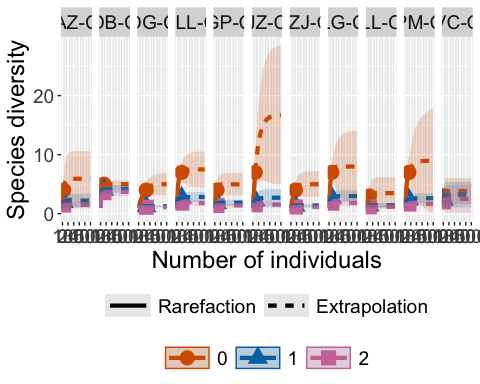


sun sites

# just forest sites:  
sun\_mat <- new\_mat[13:23,]  
  
# okay, so it seems like it thinks that species are sites and sites are species  
# can we rotate the dataframe?  
sun\_mat <- sun\_mat %>%   
 rotate\_df()  
  
  
out <- iNEXT(sun\_mat, q=c(0, 1, 2), datatype="abundance", endpoint=500)

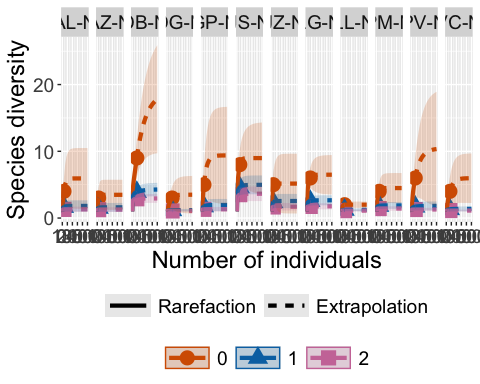
## Warning in matrix(apply(Abun.Mat, 2, function(x) invChat.Ind(x, q, goalSC)$qD),  
## : data length [1500] is not a sub-multiple or multiple of the number of rows  
## [27]

# Sample-size-based R/E curves, separating plots by "Assemblage"  
ggiNEXT(out, type=1, facet.var="Assemblage")



shade sites

# just forest sites:  
shade\_mat <- new\_mat[1:12,]  
  
# okay, so it seems like it thinks that species are sites and sites are species  
# can we rotate the dataframe?  
shade\_mat <- shade\_mat %>%   
 rotate\_df()  
  
out <- iNEXT(shade\_mat, q=c(0, 1, 2), datatype="abundance", endpoint=500)  
# Sample-size-based R/E curves, separating plots by "Assemblage"  
ggiNEXT(out, type=1, facet.var="Assemblage")



## BELOW HERE NOT WORKING

# ugh wtf - not working - take a break!

# WARNING MESSAGES: WARNING THIS SITE HAS ONLY ONE SPECIES

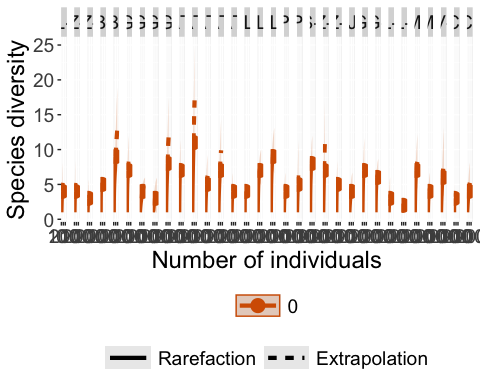
## try again here - what is output of bird data?

# data(spider)  
# out <- iNEXT(spider, q=c(0, 1, 2), datatype="abundance", endpoint=500)  
# # Sample-size-based R/E curves, separating plots by "Assemblage"  
# ggiNEXT(out, type=1, facet.var="Assemblage")  
#   
#   
# test <- as.matrix(new\_mat)  
#   
#   
#   
# out <- iNEXT(test, q=c(0, 1, 2), datatype="abundance", endpoint=500)  
# # Sample-size-based R/E curves, separating plots by "Assemblage"  
# ggiNEXT(out, type=1, facet.var="Assemblage")  
#   
#   
# data(new\_mat)

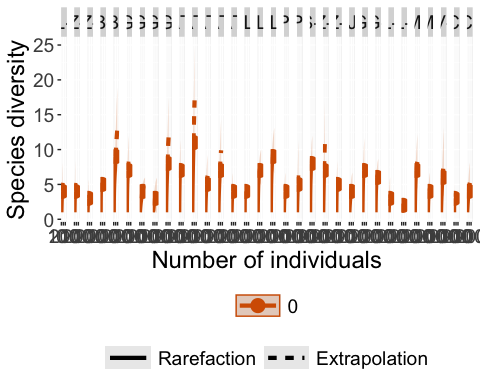
# maybe have to transpose (swap matrix?)  
  
test <- t(new\_mat)  
out <- iNEXT(test, q = 0, datatype = "abundance")

## Warning in matrix(apply(Abun.Mat, 2, function(x) invChat.Ind(x, q, goalSC)$qD),  
## : data length [500] is not a sub-multiple or multiple of the number of rows [9]

ggiNEXT(out, type=1, facet.var="Assemblage")



# str(new\_mat)  
# out <- iNEXT(new\_mat, q = 0, datatype = "abundance")  
  
  
ggiNEXT(out, type=1, facet.var="Assemblage")



sample\_matrix <- matrix(1:9, nrow = 3, ncol = 3, byrow = TRUE)  
  
# Original matrix  
print("Original matrix:")

## [1] "Original matrix:"

print(sample\_matrix)

## [,1] [,2] [,3]  
## [1,] 1 2 3  
## [2,] 4 5 6  
## [3,] 7 8 9

# Transpose the matrix using dplyr  
transposed\_matrix <- t(sample\_matrix)  
  
# Transposed matrix  
print("Transposed matrix:")

## [1] "Transposed matrix:"

print(transposed\_matrix)

## [,1] [,2] [,3]  
## [1,] 1 4 7  
## [2,] 2 5 8  
## [3,] 3 6 9

# tutorial from Francisco - need to import matrix as .txt file

# move this to process eventually

<https://www.youtube.com/watch?v=4yUXG28QPLM&t=28s>

# # # Define the file path where you want to save the matrix  
# # file\_path <- "matrix\_data.txt"  
# #   
# # # Export the matrix to a text file  
# # write.table(matrix\_data, file = file\_path, sep = "\t", row.names = FALSE, col.names = FALSE)  
# #   
#   
# # USING THIS BELOW:  
#   
#   
# # this will be consistent for each output in this script  
# data\_dir <- file.path(".", "data/processed\_data")  
#   
# # output  
# filename <- "test\_table\_iNEXT.txt"  
# filepath <- file.path(data\_dir, filename)  
# write.table(test, file = filepath)  
# ```  
#   
#   
# ## my attempt using Francisco's code  
#   
# ```{r}  
# data.ants <- read.table("data/processed\_data/test\_table\_iNEXT.txt",   
# header = TRUE, sep = "\t")  
#   
#   
#   
#   
# ```  
#   
#   
#   
#   
#   
#   
#   
# ## code from francisco - not working!  
#   
# ```{r}  
# library(iNEXT)  
# library(ggplot2)  
# library(devtools)  
# library(httr)  
#   
# #https://www.youtube.com/watch?v=24PtDhIYg8g tutorial   
#   
# #Step 1 load the data  
#   
# iNEXT\_matrix = read.table("test\_table\_iNEXT.txt", head = T)  
#   
# head(data.ants)  
#   
# #You can upload more matrix to do one   
#   
# #Convert the data frame into a list matrix for inext   
#   
# divh <- as.matrix(apply(data.ants[,-1],2, as.integer))  
#   
# #Name as row names  
#   
# row.names(divh) <- data.ants[,1]  
#   
# #List the matrix or matrices   
#   
# divant = list(Hymenoptera=divh)  
#   
# str(divant)  
#   
# edit = edit(divant) #to look if all are good   
#   
# #Alpha diversity   
#   
# out.inc = iNEXT(divant, q=0, datatype = "abundance")   
# out.inc  
#   
# #SC sampling completeness   
#   
# #For ants: 84.62% (n of 65, ci= 78-90) duplication of the sample effort mean into 93.42% of   
# #the SC   
#   
# #Figures   
#   
# #FigDivAlpha = ggiNEXT(out.inc, type = 1, color.var <- "Order.q", z <- fortify(out.inc),   
# # z$col <- z$shape <- factor(z$Order.q)) Revisar   
# par(mfrow=c(2,3))  
#   
# #figures species diversity   
# fig1 <- plot(out.inc, col= "Blue")  
# fig1  
# #figures sample coverage   
# fig2 <-plot(out.inc, type = 2, col= "Blue")  
# fig2  
# #figures species diversity against sample coverage   
# fig3 <-plot(out.inc, type = 3, col= "Blue")  
# fig3  
#

# ERRORS SHOWING UP BELOW HERE

## calculate Bray-Curtis Index

* maybe move this to ranas\_process actually!

# # Here calculating distance first, then creating NMDS off this  
# dist <- vegdist(new\_mat, method="bray")  
#   
# # set seed so consistent across iterations of nmds  
# set.seed(19960710)  
#   
# nmds <- metaMDS(dist)  
# # very low stress!  
#   
# scores(nmds) %>%  
# as\_tibble(rownames = "Group") %>% # needs to be converted to tibble for ggplot2  
# ggplot(aes(x = NMDS1, y = NMDS2)) +  
# geom\_point()

## can add rarefaction into vegdist!

# # need to set sanmple here, what should it be? randomly picked 50  
# dist <- avgdist(new\_mat, dmethod = "bray", sample = 50)  
# # this is a rarefied dist - trying to account for uneven sampling effort  
# # these metrics are very sensitive to uneven sampling effort  
#   
# nmds <- metaMDS(dist)  
#   
# scores(nmds) %>%  
# as\_tibble(rownames = "Group") %>%  
# ggplot(aes(x = NMDS1, y = NMDS2)) +  
# geom\_point()

## WHY ARENT ROW NAMES BEING USED? HEATMAP UNINTELLIGBLE

# Step 1: Calculate Bray-Curtis distance  
# bray\_matrix <- vegdist(new\_mat, method="bray")  
#   
# # Step 2: Convert distance matrix to percentage similarity  
# # Convert distances to similarities (1 - distance)  
# similarity\_matrix <- 1 - as.matrix(bray\_matrix)  
#   
# # Step 3: Plot the percentage similarity matrix  
# # Example heatmap plot  
# heatmap(similarity\_matrix,   
# main="Percentage Similarity Heatmap",  
# xlab="Samples", ylab="Samples",  
# labRow=NA, labCol=NA,  
# col=rev(heat.colors(10))) # Reversed heat colors for better visualization

## adjacent, random piece from Riffomonas project

# # ordination of bray\_matrix using cmdscale  
#   
# cmdscale(bray\_matrix) %>% plot()  
#   
# pcoa <- cmdscale(bray\_matrix)  
# colnames(pcoa) <- c("pcoa1", "pcoa2")  
# pcoa %>% head()

## adjacent, jaccard similarity index

# # create a sample data frame with mixed data types   
# df <- data.frame(   
# numeric\_var = c(1,2,3,4),   
# categorical\_var = c("A", "B", "A", "C"),   
# binary\_var = c(0,1,0,1)   
# )   
#   
# # convert categorical variable to binary   
# df$categorical\_var <- ifelse(df$categorical\_var == "A", 1, 0)   
#   
# # convert binary variable to binary   
# df$binary\_var <- as.numeric(df$binary\_var)   
#   
# # calculate the Jaccard distance   
# # matrix using the dist function   
# jaccard\_distance\_matrix <- dist(df,   
# method = "binary")   
#   
# #Print the distance matrix   
# print(jaccard\_distance\_matrix)