Sargasso\_Subisidy\_Code

#libraries

library(knitr)  
library(tidyverse)

## ── Attaching packages ─────────────────────────────────────── tidyverse 1.3.2 ──  
## ✔ ggplot2 3.4.0 ✔ purrr 1.0.0   
## ✔ tibble 3.1.8 ✔ dplyr 1.0.10  
## ✔ tidyr 1.2.1 ✔ stringr 1.5.0   
## ✔ readr 2.1.3 ✔ forcats 0.5.2   
## ── Conflicts ────────────────────────────────────────── tidyverse\_conflicts() ──  
## ✖ dplyr::filter() masks stats::filter()  
## ✖ dplyr::lag() masks stats::lag()

library(tidyr)  
library(ggplot2)  
library(ggfortify)  
library(lme4)

## Loading required package: Matrix  
##   
## Attaching package: 'Matrix'  
##   
## The following objects are masked from 'package:tidyr':  
##   
## expand, pack, unpack

library(car)

## Loading required package: carData  
##   
## Attaching package: 'car'  
##   
## The following object is masked from 'package:dplyr':  
##   
## recode  
##   
## The following object is masked from 'package:purrr':  
##   
## some

library(nlme)

##   
## Attaching package: 'nlme'  
##   
## The following object is masked from 'package:lme4':  
##   
## lmList  
##   
## The following object is masked from 'package:dplyr':  
##   
## collapse

library(ggpubr)  
library(multcompView)  
library(boot)

##   
## Attaching package: 'boot'  
##   
## The following object is masked from 'package:car':  
##   
## logit

library(Hotelling)

## Loading required package: corpcor  
##   
## Attaching package: 'Hotelling'  
##   
## The following object is masked from 'package:dplyr':  
##   
## summarise

library(mvnTest)

## Loading required package: mvtnorm

library(vegan)

## Loading required package: permute  
## Loading required package: lattice  
##   
## Attaching package: 'lattice'  
##   
## The following object is masked from 'package:boot':  
##   
## melanoma  
##   
## This is vegan 2.6-4

library(factoextra)

## Welcome! Want to learn more? See two factoextra-related books at https://goo.gl/ve3WBa

library(FactoMineR)  
library(agricolae)

## Registered S3 methods overwritten by 'klaR':  
## method from   
## predict.rda vegan  
## print.rda vegan  
## plot.rda vegan  
##   
## Attaching package: 'agricolae'  
##   
## The following object is masked from 'package:factoextra':  
##   
## hcut

library(glmm)

## Loading required package: trust  
## Loading required package: parallel  
## Loading required package: doParallel  
## Loading required package: foreach  
##   
## Attaching package: 'foreach'  
##   
## The following objects are masked from 'package:purrr':  
##   
## accumulate, when  
##   
## Loading required package: iterators

library(HH)

## Loading required package: grid  
## Loading required package: latticeExtra  
##   
## Attaching package: 'latticeExtra'  
##   
## The following object is masked from 'package:ggplot2':  
##   
## layer  
##   
## Loading required package: multcomp  
## Loading required package: survival  
##   
## Attaching package: 'survival'  
##   
## The following object is masked from 'package:boot':  
##   
## aml  
##   
## Loading required package: TH.data  
## Loading required package: MASS  
##   
## Attaching package: 'MASS'  
##   
## The following object is masked from 'package:dplyr':  
##   
## select  
##   
##   
## Attaching package: 'TH.data'  
##   
## The following object is masked from 'package:MASS':  
##   
## geyser  
##   
## Loading required package: gridExtra  
##   
## Attaching package: 'gridExtra'  
##   
## The following object is masked from 'package:dplyr':  
##   
## combine  
##   
##   
## Attaching package: 'HH'  
##   
## The following object is masked from 'package:boot':  
##   
## logit  
##   
## The following objects are masked from 'package:car':  
##   
## logit, vif  
##   
## The following object is masked from 'package:purrr':  
##   
## transpose

library(ggpubr)  
library(rstatix)

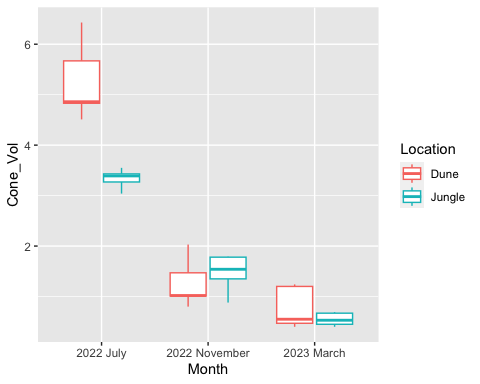
##   
## Attaching package: 'rstatix'  
##   
## The following object is masked from 'package:MASS':  
##   
## select  
##   
## The following object is masked from 'package:stats':  
##   
## filter

library(vegan)

#Plant Communities ## Pile metrics

piles <-read.csv("Sargassum\_piles.csv",header=T)  
piles2<-subset(piles, Treatment== "Sargassum")  
  
# Create the scatter plot with jittered points and add a trendline  
ggplot(piles2, aes(x = Month, y = Cone\_Vol, color = Location)) +  
 geom\_boxplot() +  
 geom\_smooth(method = "lm", se = FALSE)

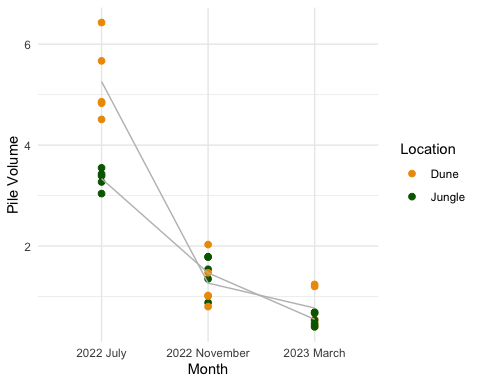
## `geom\_smooth()` using formula = 'y ~ x'



# Calculate the means by month and location  
mean\_data <- piles2 %>%  
 group\_by(Month, Location) %>%  
 summarize(mean\_cone\_vol = mean(Cone\_Vol))

## `summarise()` has grouped output by 'Month'. You can override using the  
## `.groups` argument.

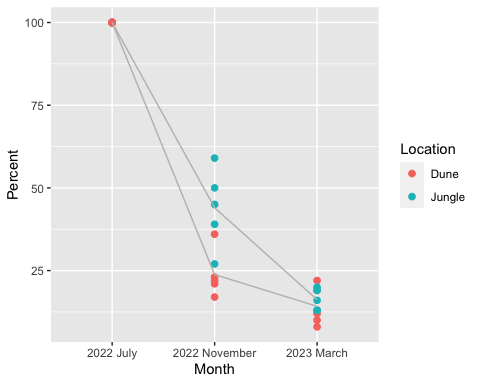
# Create the scatter plot with mean points connected by a line  
scatterplot <- ggplot(data = piles2, aes(x = Month, y = Cone\_Vol, color = Location)) +  
 geom\_point(size = 2, shape = 21, aes(fill = Location)) +  
 geom\_line(data = mean\_data, aes(x = Month, y = mean\_cone\_vol, group = Location), color = "gray")  
  
# Customize the plot appearance  
vol\_x\_time<- scatterplot +   
 labs(x = "Month", y = "Pile Volume") +  
 scale\_color\_manual(values = c("orange2", "darkgreen")) +  
 scale\_fill\_manual(values = c("orange2", "darkgreen")) +  
 theme\_minimal()  
vol\_x\_time



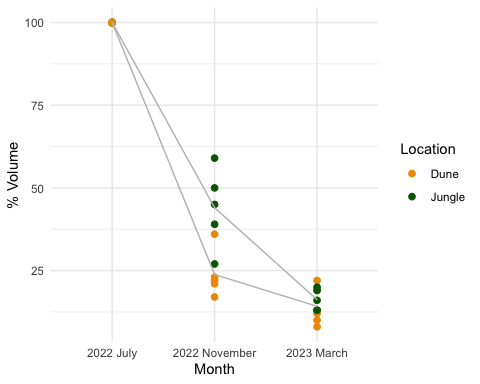
# Calculate the means pf percents by month and location  
percent\_mean\_data <- piles2 %>%  
 group\_by(Month, Location) %>%  
 summarize(mean\_cone\_percent = mean(Percent))

## `summarise()` has grouped output by 'Month'. You can override using the  
## `.groups` argument.

percent\_x\_time <- ggplot(data = piles2, aes(x = Month, y = Percent, color = Location)) +  
 geom\_point(size = 2, shape = 21, aes(fill = Location)) +  
 geom\_line(data = percent\_mean\_data, aes(x = Month, y = mean\_cone\_percent, group = Location), color = "gray")  
  
percent\_x\_time



# Customize the plot appearance  
perc\_x\_time<- percent\_x\_time +   
 labs(x = "Month", y = "% Volume") +  
 scale\_color\_manual(values = c("orange2", "darkgreen")) +  
 scale\_fill\_manual(values = c("orange2", "darkgreen")) +  
 theme\_minimal()  
perc\_x\_time



## Interior survey

#upload and subset data for interior surveys   
#import dataset  
plants<-read.csv("plantcommunity.csv", header = TRUE)  
  
#subset out only the "random" and "interior" since those are comparable (3 quads each pile interior)  
noladder<-subset(plants, !Survey == "Ladder" )  
interiorsurvey<-subset(noladder, !Survey == "HiLo")  
is.factor(interiorsurvey$Treatment)

## [1] FALSE

interiorsurvey$Treatment<-as.factor(interiorsurvey$Treatment)  
is.factor(interiorsurvey$Block)

## [1] FALSE

interiorsurvey$Block<-as.factor(interiorsurvey$Block)  
interiorsurvey$Trip<-as.factor(interiorsurvey$Trip)

###nMDS Plot

#upload and subset data for interior surveys   
#import dataset  
plants<-read.csv("plantcommunity\_NMDS.csv", header = TRUE)  
  
#subset out only the "random" and "interior" since those are comparable (3 quads each pile interior)  
noladder<-subset(plants, !Survey == "Ladder" )  
interiorsurvey<-subset(noladder, !Survey == "HiLo")  
is.factor(interiorsurvey$Treatment)

## [1] FALSE

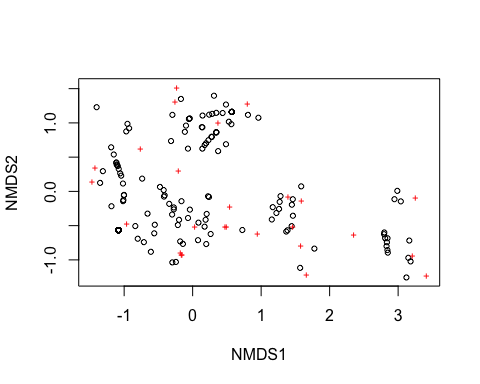
interiorsurvey$Treatment<-as.factor(interiorsurvey$Treatment)  
is.factor(interiorsurvey$Block)

## [1] FALSE

interiorsurvey$Block<-as.factor(interiorsurvey$Block)  
#compare communities differences using NMDS  
#use only the data we are interested in  
##plants first  
nm <- (interiorsurvey[,c(8:60)])  
#make matrix  
matrix = as.matrix(nm)  
nmds\_result = metaMDS(matrix, distance = "bray")

## Square root transformation  
## Wisconsin double standardization  
## Run 0 stress 0.09726287   
## Run 1 stress 0.0996805   
## Run 2 stress 0.08751648   
## ... New best solution  
## ... Procrustes: rmse 0.0160097 max resid 0.1510165   
## Run 3 stress 0.09512999   
## Run 4 stress 0.09523976   
## Run 5 stress 0.08666308   
## ... New best solution  
## ... Procrustes: rmse 0.00926564 max resid 0.07259792   
## Run 6 stress 0.0879051   
## Run 7 stress 0.08737968   
## Run 8 stress 0.08734472   
## Run 9 stress 0.08745407   
## Run 10 stress 0.1019548   
## Run 11 stress 0.08734932   
## Run 12 stress 0.08690114   
## ... Procrustes: rmse 0.005813078 max resid 0.0617269   
## Run 13 stress 0.09359707   
## Run 14 stress 0.0895423   
## Run 15 stress 0.09983789   
## Run 16 stress 0.08647898   
## ... New best solution  
## ... Procrustes: rmse 0.005975936 max resid 0.05491538   
## Run 17 stress 0.08722204   
## Run 18 stress 0.08683709   
## ... Procrustes: rmse 0.0143677 max resid 0.1200767   
## Run 19 stress 0.1063429   
## Run 20 stress 0.08712329   
## \*\*\* Best solution was not repeated -- monoMDS stopping criteria:  
## 1: no. of iterations >= maxit  
## 16: stress ratio > sratmax  
## 3: scale factor of the gradient < sfgrmin

plot(nmds\_result)



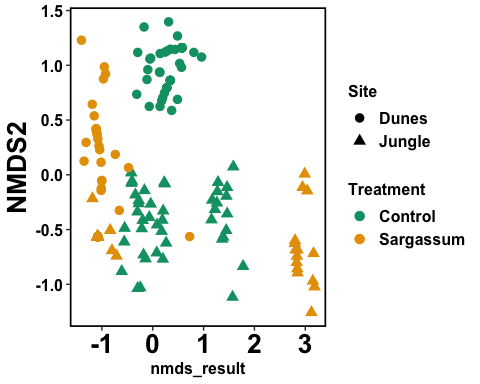
nmds\_result

##   
## Call:  
## metaMDS(comm = matrix, distance = "bray")   
##   
## global Multidimensional Scaling using monoMDS  
##   
## Data: wisconsin(sqrt(matrix))   
## Distance: bray   
##   
## Dimensions: 2   
## Stress: 0.08647898   
## Stress type 1, weak ties  
## Best solution was not repeated after 20 tries  
## The best solution was from try 16 (random start)  
## Scaling: centring, PC rotation, halfchange scaling   
## Species: expanded scores based on 'wisconsin(sqrt(matrix))'

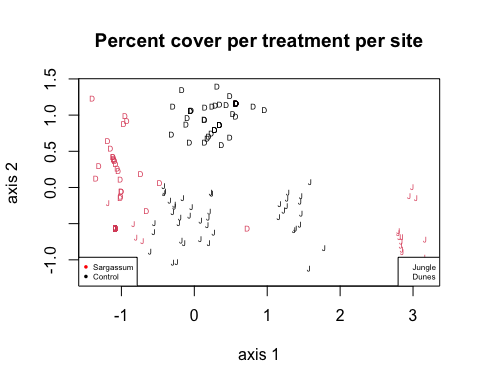
# Create a new data frame with NMDS coordinates, grouping, and additional descriptor information  
nmds\_data <- data.frame(nmds\_result$points, Site = interiorsurvey$Site, Treatment = interiorsurvey$Treatment)  
  
  
plot1 <- ggplot(nmds\_data, aes(x = MDS1, y = MDS2)) +   
 geom\_point(size = 3, aes( shape = Site, colour = Treatment))+   
 theme(axis.text.y = element\_text(colour = "black", size = 12, face = "bold"),   
 axis.text.x = element\_text(colour = "black", face = "bold", size = 20),   
 legend.text = element\_text(size = 12, face ="bold", colour ="black"),   
 legend.position = "right", axis.title.y = element\_text(face = "bold", size = 20),   
 axis.title.x = element\_text(face = "bold", size = 12, colour = "black"),   
 legend.title = element\_text(size = 12, colour = "black", face = "bold"),   
 panel.background = element\_blank(), panel.border = element\_rect(colour = "black", fill = NA, size = 1.2),  
 legend.key=element\_blank()) +   
 labs(x = "nmds\_result", colour = "Treatment", y = "NMDS2", shape = "Site") +   
 scale\_colour\_manual(values = c("#009E73", "#E69F00"))

## Warning: The `size` argument of `element\_rect()` is deprecated as of ggplot2 3.4.0.  
## ℹ Please use the `linewidth` argument instead.

plot1



#plot2  
co=c("red","black")   
shape=c(18,17,16)   
plot(nmds\_result$points, col=interiorsurvey$Treatment, pch = interiorsurvey$Site, cex = 0.5, main = "Percent cover per treatment per site", xlab = "axis 1", ylab = "axis 2")   
 Site <- c("Jungle","Dunes")   
 legend('bottomright', Site, pch = c(), cex = 0.5, bty = "y" )   
 Treatment <- c("Sargassum", "Control")   
 legend('bottomleft', Treatment, col=c("red","black"), pch = c(16), cex = 0.5, bty = "y" )



###diversity index

survey <- (interiorsurvey[,c(1,3,4,5,6,8:60)])  
survey$Block<-as.factor(survey$Block)  
survey$Quadrat<-as.factor(survey$Quadrat)  
survey$Trip<-as.factor(survey$Trip)

###dune index

survey <- (interiorsurvey[,c(1,3,4,5,6,8:60)])  
survey$Block<-as.factor(survey$Block)  
survey$Quadrat<-as.factor(survey$Quadrat)  
survey$Trip<-as.factor(survey$Trip)  
  
Dunes\_survey\_data<-subset(survey, Site == "Dunes")  
#merge litter and dead since litter was supposed to be categorized as dead  
  
Dunes\_survey\_data$Dead\_Lit <- paste(Dunes\_survey\_data$Dead + Dunes\_survey\_data$Litter)  
  
# Remove the original columns if needed  
Dunes\_survey\_data <- Dunes\_survey\_data[, !(names(Dunes\_survey\_data) %in% c("Dead", "Litter"))]  
  
  
# Get the column names from the 6th column onwards  
Dunes\_species\_columns <- names(Dunes\_survey\_data)[6:ncol(Dunes\_survey\_data)]  
  
# Filter out columns with only zeros  
Dunes\_non\_zero\_columns <- Dunes\_species\_columns[!apply(Dunes\_survey\_data[Dunes\_species\_columns], 2, function(x) all(x == 0))]  
  
# Create a new data frame with the non-zero columns  
Dunes\_filtered\_df <- Dunes\_survey\_data[c('Trip', 'Site', 'Block', 'Treatment', 'Quadrat', Dunes\_non\_zero\_columns)]  
view(Dunes\_filtered\_df)  
  
  
# Convert the columns to numeric if they are not already  
Dunes\_filtered\_df[, Dunes\_non\_zero\_columns] <- lapply(Dunes\_filtered\_df[, Dunes\_non\_zero\_columns], as.numeric)  
  
# Convert numbers to percent cover for each row independently  
for (row in 1:nrow(Dunes\_filtered\_df)) {  
 row\_sum <- sum(Dunes\_filtered\_df[row, Dunes\_non\_zero\_columns])  
 if (row\_sum != 0) {  
 Dunes\_filtered\_df[row, Dunes\_non\_zero\_columns] <- (Dunes\_filtered\_df[row, Dunes\_non\_zero\_columns] / row\_sum) \* 100  
 }  
}  
  
# Calculate Bray-Curtis dissimilarity matrix  
Dunes\_bray\_curtis\_dist <- vegdist(Dunes\_filtered\_df[, Dunes\_non\_zero\_columns], method = "bray")  
  
# Calculate beta diversity (average dissimilarity) using Bray-Curtis  
Dunes\_beta\_diversity <- mean(Dunes\_bray\_curtis\_dist)  
  
# View the calculated beta diversity  
Dunes\_beta\_diversity

## [1] 0.6073517

# Assuming your dissimilarity matrix is named 'bray\_curtis\_dist' and you have the corresponding factors in 'Trip', and 'Treatment'  
  
# Perform PERMANOVA to compare beta diversity  
Dunes\_permanova\_result <- adonis2(Dunes\_bray\_curtis\_dist ~ Treatment/Trip, data = Dunes\_filtered\_df)  
  
  
# View the PERMANOVA results  
Dunes\_permanova\_result

## Permutation test for adonis under reduced model  
## Terms added sequentially (first to last)  
## Permutation: free  
## Number of permutations: 999  
##   
## adonis2(formula = Dunes\_bray\_curtis\_dist ~ Treatment/Trip, data = Dunes\_filtered\_df)  
## Df SumOfSqs R2 F Pr(>F)   
## Treatment 1 14.5086 0.61435 259.09 0.001 \*\*\*  
## Treatment:Trip 4 4.4036 0.18647 19.66 0.001 \*\*\*  
## Residual 84 4.7039 0.19918   
## Total 89 23.6161 1.00000   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

###jungles index

survey <- (interiorsurvey[,c(1,3,4,5,6,8:60)])  
survey$Block<-as.factor(survey$Block)  
survey$Quadrat<-as.factor(survey$Quadrat)  
survey$Trip<-as.factor(survey$Trip)  
Jungle\_survey\_data <- subset(survey, Site == "Jungle")  
  
#merge litter and dead since litter was supposed ot be categorized as dead  
Jungle\_survey\_data$Dead\_Lit <- paste(Jungle\_survey\_data$Dead + Jungle\_survey\_data$Litter)  
# Remove the original columns if needed  
Jungle\_survey\_data <- Jungle\_survey\_data[, !(names(Jungle\_survey\_data) %in% c("Dead", "Litter"))]  
# Get the column names from the 6th column onwards  
Jungle\_species\_columns <- names(Jungle\_survey\_data)[6:ncol(Jungle\_survey\_data)]  
view(Jungle\_species\_columns)  
# Filter out columns with only zeros  
Jungle\_non\_zero\_columns <- Jungle\_species\_columns[!apply(Jungle\_survey\_data[Jungle\_species\_columns], 2, function(x) all(x == 0))]  
# Create a new data frame with the non-zero columns  
Jungle\_filtered\_df <- Jungle\_survey\_data[c('Trip', 'Site', 'Block', 'Treatment', 'Quadrat', Jungle\_non\_zero\_columns)]  
view(Jungle\_filtered\_df)  
# Convert the columns to numeric if they are not already  
Jungle\_filtered\_df[, Jungle\_non\_zero\_columns] <- lapply(Jungle\_filtered\_df[, Jungle\_non\_zero\_columns], as.numeric)  
# Convert numbers to percent cover for each row independently  
for (row in 1:nrow(Jungle\_filtered\_df)) {  
 row\_sum <- sum(Jungle\_filtered\_df[row, Jungle\_non\_zero\_columns])  
 if (row\_sum != 0) {  
 Jungle\_filtered\_df[row, Jungle\_non\_zero\_columns] <- (Jungle\_filtered\_df[row, Jungle\_non\_zero\_columns] / row\_sum) \* 100  
 }  
}  
# Calculate Bray-Curtis dissimilarity matrix  
Jungle\_bray\_curtis\_dist <- vegdist(Jungle\_filtered\_df[, Jungle\_non\_zero\_columns], method = "bray")  
# Calculate beta diversity (average dissimilarity) using Bray-Curtis  
Jungle\_beta\_diversity <- mean(Jungle\_bray\_curtis\_dist)  
# View the calculated beta diversity  
Jungle\_beta\_diversity

## [1] 0.3547753

# Assuming your dissimilarity matrix is named 'bray\_curtis\_dist' and you have the corresponding factors in 'Trip', and 'Treatment'  
# Perform PERMANOVA to compare beta diversity  
Jungle\_permanova\_result <- adonis2(Jungle\_bray\_curtis\_dist ~ Treatment/Trip, data = Jungle\_filtered\_df)  
# View the PERMANOVA results  
Jungle\_permanova\_result

## Permutation test for adonis under reduced model  
## Terms added sequentially (first to last)  
## Permutation: free  
## Number of permutations: 999  
##   
## adonis2(formula = Jungle\_bray\_curtis\_dist ~ Treatment/Trip, data = Jungle\_filtered\_df)  
## Df SumOfSqs R2 F Pr(>F)   
## Treatment 1 1.5236 0.17362 43.911 0.001 \*\*\*  
## Treatment:Trip 4 4.3372 0.49425 31.250 0.001 \*\*\*  
## Residual 84 2.9146 0.33213   
## Total 89 8.7753 1.00000   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

#all significant but we should focus on seeing if there is a treatment effect within each site rather than by between

# Plant Community Structure

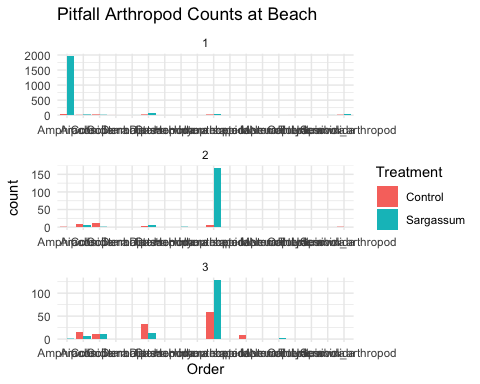
#import dataset  
plants<-read.csv("plantcommunity.csv", header = TRUE)  
  
#subset out only the "random" and "interior" since those are comparable (3 quads each pile interior)  
noladder<-subset(plants, !Survey == "Ladder" )  
nohilo<-subset(noladder, !Survey == "HiLo")  
  
#subset out the sites  
duneinterior<-subset(nohilo, Site == "Dunes")  
jungleinterioir<-subset(nohilo, Site == "Jungle")

#Arthropods ##pitfalls

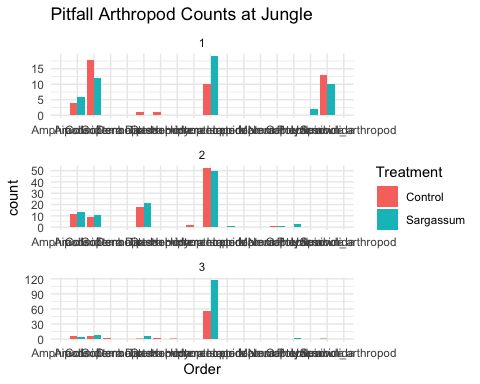
Pitfalls <-read.csv("Pitfalls.csv",header=T)  
#subset out the X treatments since it was just to try out  
pitfalls<-subset(Pitfalls, !Placement == "X" )  
  
  
# Convert columns 9 to 26 into rows while keeping columns 1 to 8 as factors  
pitfalls\_data\_long <- gather(pitfalls, key = "Order", value = "Count", 9:26, -c(1:8))  
is.factor(pitfalls\_data\_long$Trip)

## [1] FALSE

pitfalls\_data\_long$Trip<-as.factor(pitfalls\_data\_long$Trip)  
  
#subset dunes   
pitfall\_dunes<- subset(pitfalls\_data\_long, Site == "Dune")  
  
ggplot(pitfall\_dunes) +  
 aes(x = Order, fill = Treatment, weight = Count) +  
 geom\_bar(position = "dodge") +  
 scale\_fill\_hue(direction = 1) +  
 theme\_minimal() +  
 facet\_wrap(vars(Trip), scales = "free", nrow = 4L) +  
 labs(title = "Pitfall Arthropod Counts at Beach")



#subset jungle  
pitfall\_jungle<- subset(pitfalls\_data\_long, Site == "Jungle")  
  
ggplot(pitfall\_jungle) +  
 aes(x = Order, fill = Treatment, weight = Count) +  
 geom\_bar(position = "dodge") +  
 scale\_fill\_hue(direction = 1) +  
 theme\_minimal() +  
 facet\_wrap(vars(Trip), scales = "free", nrow = 4L) +  
 labs(title = "Pitfall Arthropod Counts at Jungle")

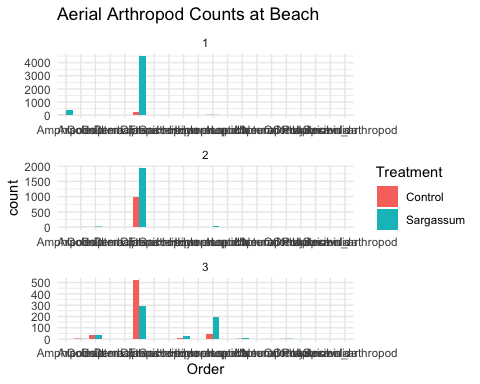


##stickies ###data Exploration

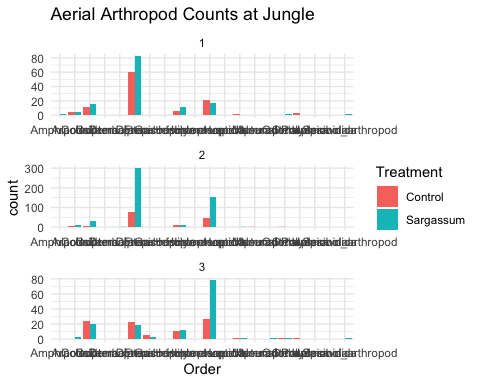
stickies1 <-read.csv("stickies.csv",header=T)  
#remove total column  
  
stickies <- subset(stickies1, select = -Total)  
  
  
# Convert columns 9 to 26 into rows while keeping columns 1 to 8 as factors  
stickies\_data\_long <- gather(stickies, key = "Order", value = "Count", 9:28, -c(1:8))  
is.factor(stickies\_data\_long$Trip)

## [1] FALSE

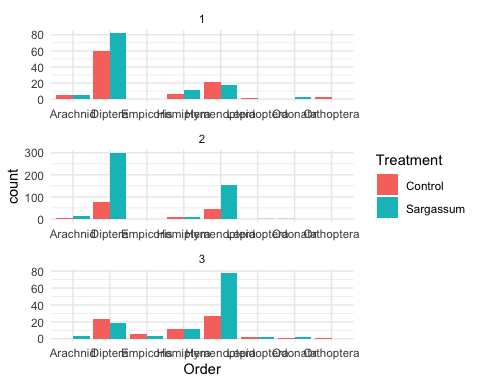
stickies\_data\_long$Trip<-as.factor(stickies\_data\_long$Trip)  
  
#subset dunes   
stickies\_dunes<- subset(stickies\_data\_long, Site == "Dune")  
  
ggplot(stickies\_dunes) +  
 aes(x = Order, fill = Treatment, weight = Count) +  
 geom\_bar(position = "dodge") +  
 scale\_fill\_hue(direction = 1) +  
 theme\_minimal() +  
 facet\_wrap(vars(Trip), scales = "free", nrow = 4L) +  
 labs(title = "Aerial Arthropod Counts at Beach")



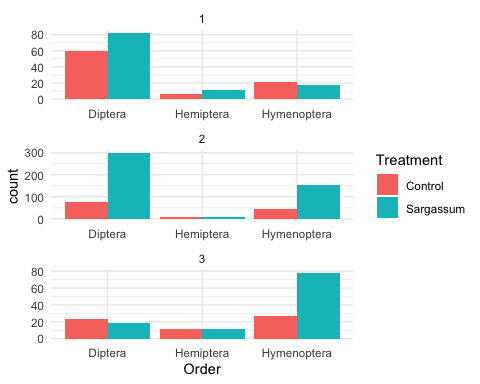
#subset jungle  
stickies\_jungle<- subset(stickies\_data\_long, Site == "Jungle")  
  
ggplot(stickies\_jungle) +  
 aes(x = Order, fill = Treatment, weight = Count) +  
 geom\_bar(position = "dodge") +  
 scale\_fill\_hue(direction = 1) +  
 theme\_minimal() +  
 facet\_wrap(vars(Trip), scales = "free", nrow = 4L) +  
 labs(title = "Aerial Arthropod Counts at Jungle")



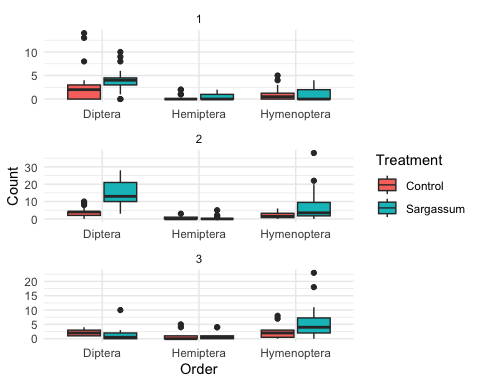
#explore Jungle Data  
#total counts bar graph  
  
stickies\_jungle %>%  
 filter(Order %in% c("Orthoptera", "Hemiptera", "Arachnid", "Diptera", "Hymenoptera",   
"Lepidoptera", "Empicoris", "Odonata")) %>%  
 ggplot() +  
 aes(x = Order, fill = Treatment, weight = Count) +  
 geom\_bar(position = "dodge") +  
 scale\_fill\_hue(direction = 1) +  
 theme\_minimal() +  
 facet\_wrap(vars(Trip), scales = "free", nrow = 4L)



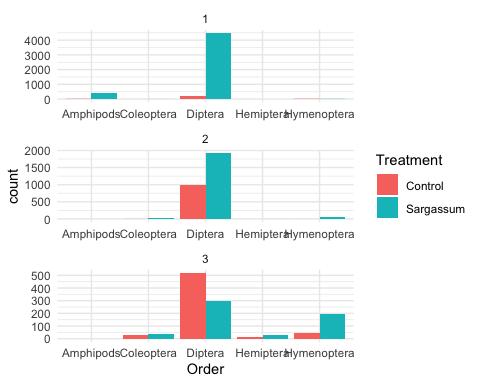
#the three main contributors to variability   
stickies\_jungle %>%  
 filter(Order %in% c("Hemiptera", "Diptera", "Hymenoptera")) %>%  
 ggplot() +  
 aes(x = Order, fill = Treatment, weight = Count) +  
 geom\_bar(position = "dodge") +  
 scale\_fill\_hue(direction = 1) +  
 theme\_minimal() +  
 facet\_wrap(vars(Trip), scales = "free", nrow = 4L)



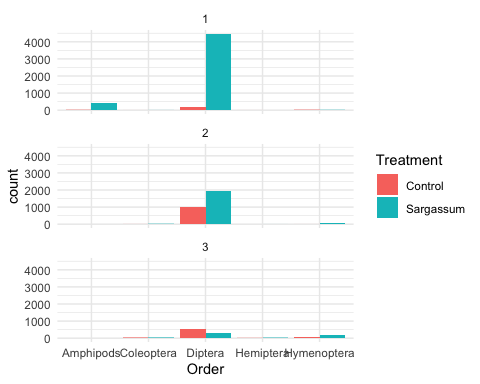
#now as a box plot   
  
stickies\_jungle %>%  
 filter(Order %in% c("Hemiptera", "Diptera", "Hymenoptera")) %>%  
 ggplot() +  
 aes(x = Order, y = Count, fill = Treatment) +  
 geom\_boxplot() +  
 scale\_fill\_hue(direction = 1) +  
 theme\_minimal() +  
 facet\_wrap(vars(Trip), scales = "free", nrow = 4L)



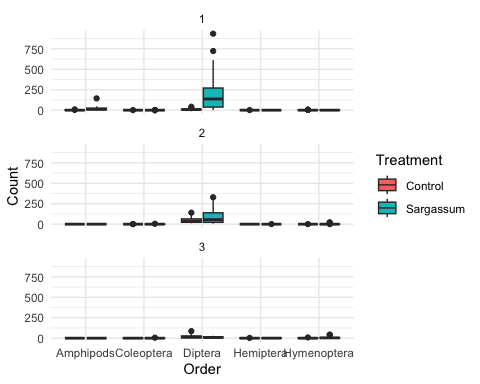
#explore Dune Data  
#total counts bar graph with free scales after removing arthropods that are crawlers, don’t add much to the graphs, or pollinators.  
stickies\_dunes %>%  
 filter(Order %in% c("Amphipods", "Coleoptera", "Hemiptera", "Diptera", "Hymenoptera"  
)) %>%  
 ggplot() +  
 aes(x = Order, fill = Treatment, weight = Count) +  
 geom\_bar(position = "dodge") +  
 scale\_fill\_hue(direction = 1) +  
 theme\_minimal() +  
 facet\_wrap(vars(Trip), scales = "free", nrow = 4L)



# the pattern looks more pronounced and maybe reads easiest with fixed scales  
  
stickies\_dunes %>%  
 filter(Order %in% c("Amphipods", "Coleoptera", "Hemiptera", "Diptera", "Hymenoptera"  
)) %>%  
 ggplot() +  
 aes(x = Order, fill = Treatment, weight = Count) +  
 geom\_bar(position = "dodge") +  
 scale\_fill\_hue(direction = 1) +  
 theme\_minimal() +  
 facet\_wrap(vars(Trip), nrow = 4L)



#the fly pattern is so interesting! No doubt of a subsidy effect there.   
  
#now as a boxplot  
stickies\_dunes %>%  
 filter(Order %in% c("Amphipods", "Coleoptera", "Hemiptera", "Diptera", "Hymenoptera"  
)) %>%  
 ggplot() +  
 aes(x = Order, y = Count, fill = Treatment) +  
 geom\_boxplot() +  
 scale\_fill\_hue(direction = 1) +  
 theme\_minimal() +  
 facet\_wrap(vars(Trip), nrow = 4L)



#pods only… as expected, we see a surge on initial day and nothing much after  
  
stickies\_dunes %>%  
 filter(Order %in% "Amphipods") %>%  
 ggplot() +  
 aes(x = Order, y = Count, fill = Treatment) +  
 geom\_boxplot() +  
 scale\_fill\_hue(direction = 1) +  
 theme\_minimal() +  
 facet\_wrap(vars(Trip), scales = "free", nrow = 4L)

