thesisresults

Annabelle

April 21, 2020

Set up file structure

### Figure 1: Correlation matrix

#set up  
combooo <- combo  
#change names for graphic   
combooo <- combooo %>% rename("Area" = area\_x)   
combooo <- combooo %>% rename("Threats" = total\_x\_x)  
combooo <- combooo %>% rename("Partners" = total)  
combooo <- combooo %>% rename("Employment" = nsumemploy)  
combooo <- combooo %>% rename("%PublicLand" = percentpublic)  
combooo <- combooo %>% rename("Taxa" = taxa)  
#lapply(combooo, as.numeric)  
PredictorsOnlyPixel <- combooo[,c(5)]  
PredictAndResponsePixel <- combooo[,-c(1:4)]  
PredictAndResponseGrid <- combooo[,-c(1:4)]  
  
  
# put histograms on the diagonal panel   
panel.hist <- function (x,...) # define a function that says what we want to plot in the diagonal  
{  
 usr <- par("usr"); on.exit(par(usr)) # not sure what usr is for?  
 par(usr = c(usr[1:2],0,1.5))  
 h <- hist(x, plot = FALSE)  
 breaks <- h$breaks; nB <- length(breaks) # make the hist   
 y <- h$counts; y <- y/max(y)  
 rect(breaks[-nB], 0, breaks[-1], y, col="grey", ...) # defines what the histogram is going to look like  
}  
  
## put correlations & 95% CIs on the upper panels,  
panel.cor <- function(x, y, digits=2, prefix="", cex.cor, ...)  
{  
 usr <- par("usr"); on.exit(par(usr))  
 par(usr = c(0, 1, 0, 1))  
 r <- cor(x, y,use="complete.obs")  
 txt <- format(c(r, 0.123456789), digits=digits)[1]  
 prefix <- "r = "  
 rc <- cor.test(x,y)  
 rci <- rc$conf.int  
 txt2 <- format(c(rci, 0.123456789), digits=digits)[1]  
 txt3 <- format(c(rci, 0.123456789), digits=digits)[2]  
 prefix2 <- "\nCI = "  
 txt <- paste(prefix, txt, prefix2, txt2, ", ", txt3, sep="")  
 if(missing(cex.cor)) cex.cor <- 0.8/strwidth(txt)  
 text(0.5, 0.5, txt, cex = 1)  
}  
  
## plot a correlation matrix plot that uses the functions specified above to say what to plot where  
 ## this was taken directly from website and still not plotting r values for all   
pairs(PredictAndResponsePixel[1:6], lower.panel=panel.smooth, cex = .8, diag.panel=panel.hist, cex.labels = 1.2, font.labels=2, upper.panel=panel.cor)  
  
  
pairs(PredictAndResponsePixel[1:6],lower.panel = panel.smooth, diag.panel=panel.hist,upper.panel=panel.cor)

### Figure 2: Map of the ranges of all species considered in this dataset and the average center point for all species

Map of all species

Map of all species

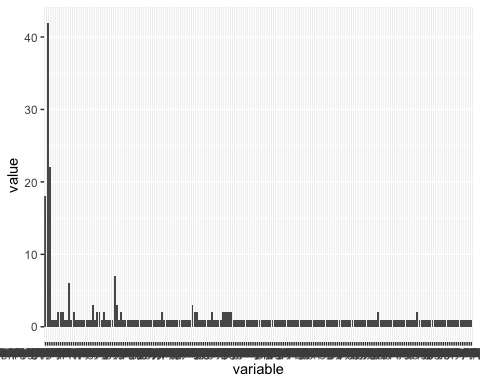
### Figure 3: A subsection of the total dataframe to show the number of species a partner is working on

##### Planning to use the second of two graphs for paper

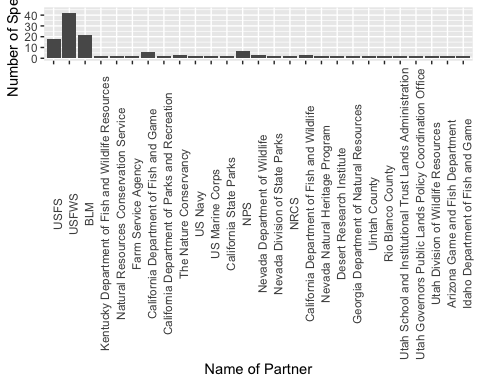
#this is just the diagonal matrix from adj matrix   
graphme <- adjmatrix  
graphme <- graphme %>% melt()

## Using X1 as id variables

graphme <- graphme %>% filter(X1 == variable)  
graphme <- graphme[,-c(1)] #removed duplicate names  
#all partners  
ggplot(data = graphme) + geom\_col(mapping = aes(x=variable, y=value))



g\_one <- graphme %>% filter(value >1)  
ggplot(data = g\_one, mapping = aes(x=variable, y=value)) + geom\_col() + theme(axis.text.x = element\_text(angle = 90)) + scale\_y\_continuous(name ="Number of Species", limits=c(0,45)) + scale\_x\_discrete(name ="Name of Partner")



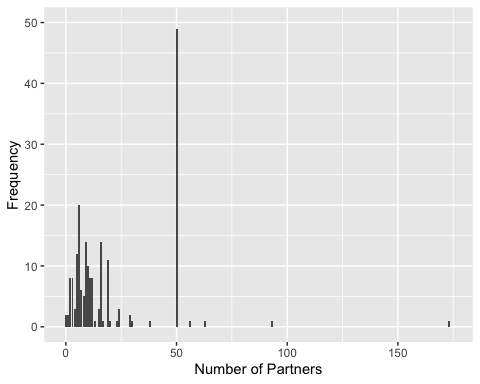
### Number of partners per partner

#### Same question as above about subsections/scale of zoom

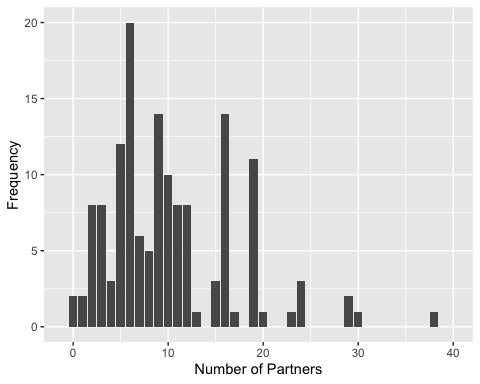
#going to convert all values that are greater than 1 to one so not double counting   
# then subtract diagonal   
#then add   
pdata <- adjmatrix  
newmat <- pdata[1,]  
newmat[,1] <- "empty"   
#add empty row with same no of columns to combine with current matrix   
newMatrix <- rbind(newmat, pdata)  
  
newMatrix <- newMatrix %>% mutate\_if(is.numeric, ~1 \* (. != 0))  
  
diag(newMatrix)=0  
#delete top (empty) row   
newMatrix <- newMatrix[-1,] ##I think this matrix is worth keeping (all numers now 1s and 0s )  
asnum <- newMatrix[,-c(1)]  
names <- newMatrix[,c(1)]  
rssum <- rowSums(newMatrix[,c(-1)])  
newdf <- cbind(names, rssum) ## yay I did it!!   
  
## now going to try and filter  
  
explore <- newdf   
explore <- explore %>% arrange(-rssum)  
(explore[c(1:4),])

## X1 rssum  
## 1 USFWS 173  
## 2 BLM 93  
## 3 USFS 63  
## 4 Natural Resources Conservation Service 56

#still feeds mis-represented   
  
#summary count of the number of partnership for each partner (how many partner have 2 parnters, 3 etc? )   
ggplot(data = explore, mapping = aes(x=rssum)) + geom\_bar() + scale\_x\_continuous(name ="Number of Partners", limits=c(-1,175)) + scale\_y\_continuous(name ="Frequency", limits=c(0,50))



#because data is scewed with giant outlier, have recalibrated to show what majority of the data looks like   
filt <- explore %>% filter(rssum < 50)   
ggplot(data = filt, mapping = aes(x=rssum)) + geom\_bar() + scale\_x\_continuous(name ="Number of Partners", limits=c(-1,40)) + scale\_y\_continuous(name ="Frequency", limits=c(0,20))



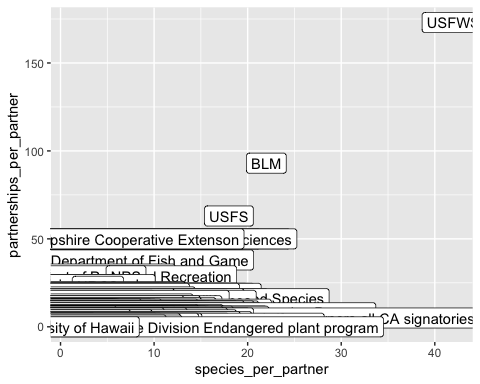
### Figure [not yet referenced in text] Number of partners/partner vs number of species/partner

#### different options here

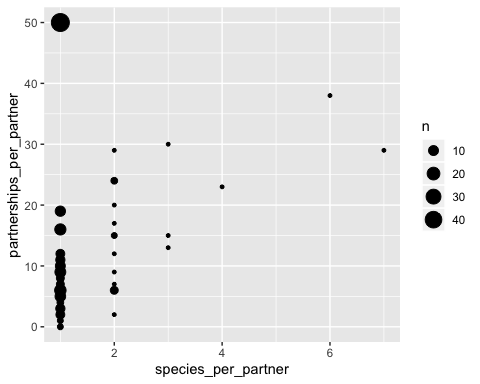
specesperpartner <- colSums(PartnerData[,-c(1:3)])  
speperpart <- t(specesperpartner)  
sppart <- t(speperpart) #trying to get partner names as row names  
number\_of\_species\_per\_partner <- sppart[-c(199),] #gets werid when I delete the bottom row  
number\_of\_species\_per\_partner <- t(number\_of\_species\_per\_partner)  
number\_of\_species\_per\_partner <- t(number\_of\_species\_per\_partner) #this works  
#making rownames a column so can join later   
number\_of\_species\_per\_partner <- as.data.frame(number\_of\_species\_per\_partner)  
number\_of\_species\_per\_partner <- number\_of\_species\_per\_partner %>% rownames\_to\_column(var = "X1")  
number\_of\_species\_per\_partner <- number\_of\_species\_per\_partner %>% rename("species\_per\_partner" = V1)   
  
explore <- explore %>% rename("partnerships\_per\_partner" = rssum)  
  
## the joins aren't working - just getting partner names then replicated row columns  
and <- explore %>% left\_join(number\_of\_species\_per\_partner, by = NULL)

## Joining, by = "X1"

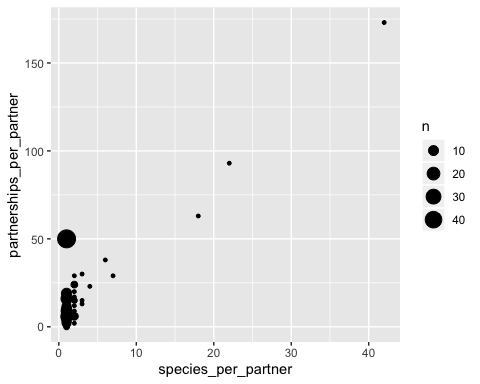
#need to fix NRCS  
and <- and[-which(is.na(and$species\_per\_partner)),]  
  
ggplot(data = and, mapping = aes(x=partnerships\_per\_partner, y= species\_per\_partner)) + geom\_point() +geom\_label(mapping = aes(x=partnerships\_per\_partner, y=species\_per\_partner, label = X1)) +coord\_flip()



and2 <- and %>% filter(X1 != "BLM") %>% filter(X1 != "USFWS") %>% filter(X1 != "USFS")   
ggplot(data = and2, mapping = aes(x=partnerships\_per\_partner, y= species\_per\_partner)) + geom\_count() +coord\_flip()



ggplot(data = and, mapping = aes(x=partnerships\_per\_partner, y= species\_per\_partner)) + geom\_count() +coord\_flip()

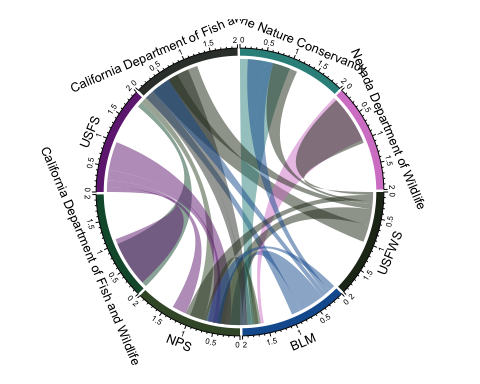


### Network analysis / chord diagram

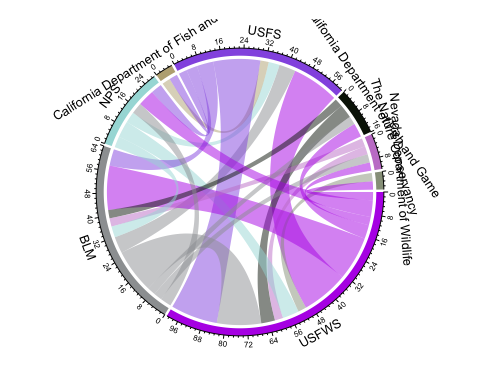
dfadj <- melt(adjmatrix) #changes matrix to dataframe

## Using X1 as id variables

dfadj <- dfadj %>% filter(value > 1) #remove all partner less than 1  
  
#see if can remove the count of how many species a partner is working on (loop going back to self)  
#to do this can remove row where first two columns have same character  
noextra <- dfadj %>% filter(X1 != variable)  
#chordDiagram(noextra)  
  
two <- noextra %>% filter(value > 2) #remove all partners less than 2   
chordDiagram(two, scale = TRUE)



## so their still seems to be some redundancy where the is a line going from parter a to partner b and another line (samve value) going from b back to a   
### scale = TRUE seems to fix this?   
chordDiagram(two, scale = FALSE)



### with scale - "value represents the fraction of the interaction going to one other secto"  
  
#labels - need to pre-allocate empty track and costomize later (going to be a pain for each so just do later?)