RegressionPredictorExploration

Annabelle

March 7, 2020

## R Markdown

This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. For more details on using R Markdown see <http://rmarkdown.rstudio.com>.

When you click the **Knit** button a document will be generated that includes both content as well as the output of any embedded R code chunks within the document. You can embed an R code chunk like this:

Set up file structure

load in data

## New names:  
## \* `` -> ...26

Combining two datasets and limiting to predictors of interest

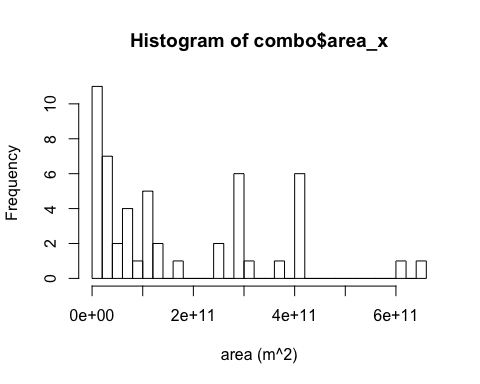
#View(partnercount) looking at data  
partnercount <- partnercount[, c(2,3,11)] #only selected rows with species names and the total counts  
tdata <- tdata[,c(2,5,20,6,21,17,46,53)]  
# proxys being used in analysis and corresponding column number  
 # range size --> area c20   
 # species group --> taxa c6  
 # area weighted footprint --> c21 ?have options  
 # type of threat --> c17  
 # have 4 other issues to address --> may need to come back to this list   
 # employment (all relevant industries) -> 46  
 # income (all relevant industries) -> 53  
#combine two datasets  
#First need to have the names columns be the same (tdata currently has x\_x at end) - actually going to try without  
tdata <- tdata %>% rename(scientific\_name = scientific\_name\_x\_x)  
  
combo <- partnercount %>% full\_join(tdata,  
 by ="scientific\_name")  
#no regression data on 4 of the species that we have partner counts for  
#Removing the data that there are no partner counts for   
combo <- combo[-which(is.na(combo$total)),] #removed all rows that had na's for partners aka the ones with no information  
#removing extra common name colum   
combo <- combo[,-c(4)]

Need some way to standardize number of partners per agreeement (exact number of even weighting?) need to create columns with HCP | CCA | CCAA | Other

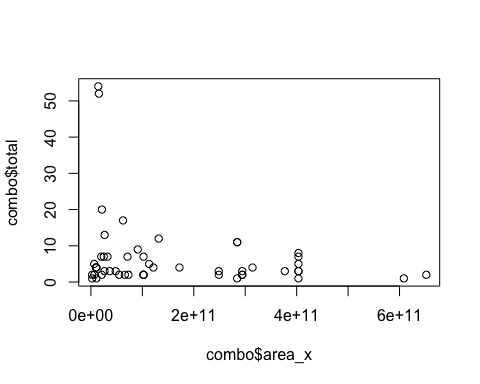
agree <- agree[,c(1,2,8,15)]  
agree <- agree[-which(agree$`type of agreements (source of information)` == "None"),]  
##creating function to check agreement years and then will load in here

Predictor: range size

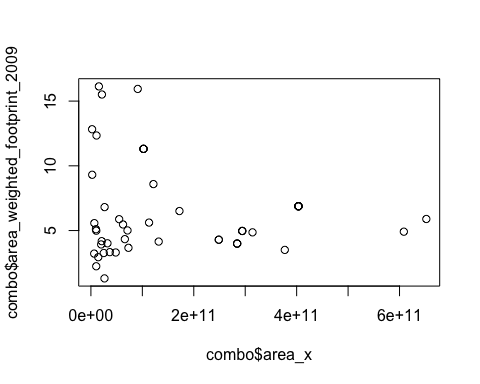
#distribution of area  
hist(combo$area\_x, xlab ="area (m^2)", breaks = 30)



#response variable or total number of partners  
plot(combo$area\_x, combo$total)



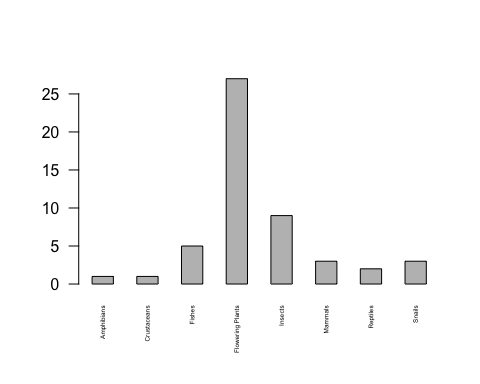
#predictor: taxa  
  
#predictor: human footprint  
plot(combo$area\_x, combo$area\_weighted\_footprint\_2009)



#predictor: threat type

Predictor: taxa

#distribution of taxa  
taxa <- table(combo$species\_group\_x\_x)  
barplot(table(combo$species\_group\_x\_x), las=2, space=1.1,cex.names = .4)



#going to change code to do 1 for flowering plants and 0 for all else  
#create new column and fill it  
taxa <- c(0)  
combo <- add\_column(combo, taxa)  
which(combo$species\_group\_x\_x == "Flowering Plants")

## [1] 1 2 3 5 6 7 9 10 12 13 14 18 23 24 25 26 28 29 33 36 41 44 47  
## [24] 48 53 54 55

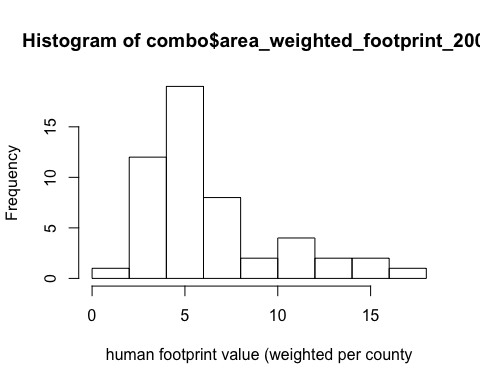
combo[which(combo$species\_group\_x\_x == "Flowering Plants"),10] <- 1  
(table(combo$taxa, combo$total))

##   
## 1 2 3 4 5 6 7 8 9 11 12 13 17 20 52 54  
## 0 1 5 5 4 1 1 4 2 1 0 1 0 1 0 1 1  
## 1 4 8 4 2 2 0 2 0 1 2 0 1 0 1 0 0

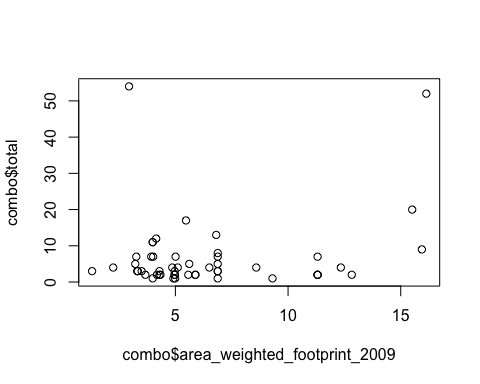
#charasmatic mega funa (how define?)

Predictor: human footprint

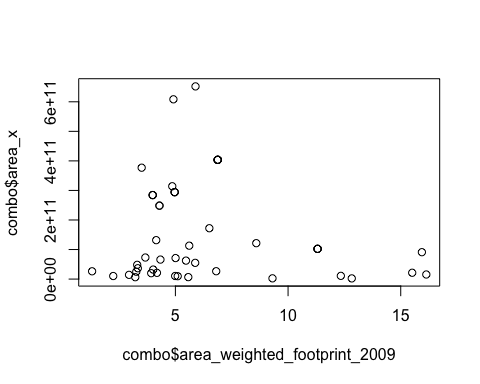
#distribution of human footprint   
hist(combo$area\_weighted\_footprint\_2009, xlab ="human footprint value (weighted per county", breaks = 10)



#response variable or total number of partners  
plot(combo$area\_weighted\_footprint\_2009, combo$total)



#predictor: taxa  
  
#predictor: human footprint  
plot(combo$area\_weighted\_footprint\_2009, combo$area\_x)

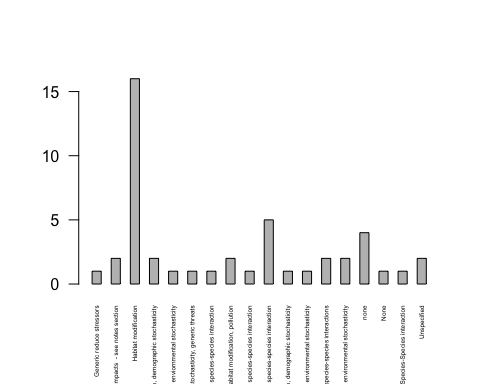


#predictor: threat type

Predictor: number of overlapping ranges (Need to bring in from ArcGIS)

Predictor: threats data

#cleaning by replacing and with a comma  
combo$threats\_addressed\_by\_conservation\_x\_x[17] <- "Habitat modification, species-species interaction" #added comma  
combo$threats\_addressed\_by\_conservation\_x\_x[18] <- "Habitat modification, pollution, species-species interaction" #added comma  
combo$threats\_addressed\_by\_conservation\_x\_x[26] <- "Habitat modification, species-species interactions, demographic stochasticity, environmental stochasticity" #added comma  
combo$threats\_addressed\_by\_conservation\_x\_x[36] <- "Habitat modification, demographic stochasticity"  
combo$threats\_addressed\_by\_conservation\_x\_x[39] <- "Habitat modification, species-species interaction, demographic stochasticity, environmental stochasticity"  
combo$threats\_addressed\_by\_conservation\_x\_x[44] <-"Habitat modification, species-species interaction" #added comma  
combo$threats\_addressed\_by\_conservation\_x\_x[53] <- "Habitat modification, species-species interactions, demographic stochasticity, environmental stochasticity"  
  
 #distribution of threats  
threats <- table(combo$threats\_addressed\_by\_conservation\_x\_x)  
barplot(threats, las=2, space=1.1,cex.names = .4)

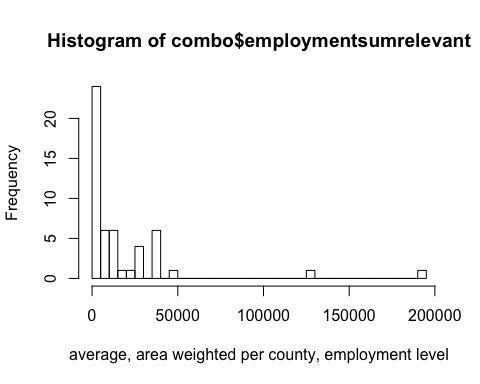


##need to clean difference between none and None? (will check with tyler but as of now merge)  
combo[which(combo$threats\_addressed\_by\_conservation\_x\_x == "none"),7] <- "None"  
#going to change code to do 1 for flowering plants and 0 for all else  
hthreat <- c(0) #create new column and fill it  
combo <- add\_column(combo, hthreat)  
which(combo$threats\_addressed\_by\_conservation\_x\_x == "Habitat modification")

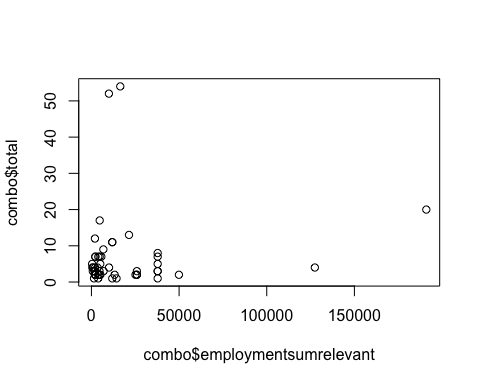
## [1] 2 3 5 8 11 16 22 27 31 32 34 41 45 50 51 52

combo[which(combo$threats\_addressed\_by\_conservation\_x\_x == "Habitat modification"),11] <- 1  
 #this only changes the cells that have habitat modification as the only threat, not ones with multiple threats  
#count the toal number of threats for each species   
noneaszero <- combo[-which(combo$threats\_addressed\_by\_conservation\_x\_x == "None"),]  
noneaszero$totalthreat <- str\_count(noneaszero$threats\_addressed\_by\_conservation\_x\_x, ",") +1 #because partners are seperated by a , need to add 1 to each  
#then rejoin with original dataset  
## remove all datacolumns in noneaszeros besides name and counts  
noneaszero <- noneaszero[,-c(3:11)]  
combo <- combo %>% #combine original and checked into new data frame  
 full\_join(noneaszero,   
 by = c("scientific\_name", "common\_name"))  
#add the "Nones" into totalthreat column   
combo[which(combo$threats\_addressed\_by\_conservation\_x\_x == "None"),12] <- 0  
  
#every column with multiple threats includes habitat as a threat, but because this is in a string it isn't being counted by hthreat column. Going to add count now   
combo[which(combo$totalthreat > 1),11] <- 1

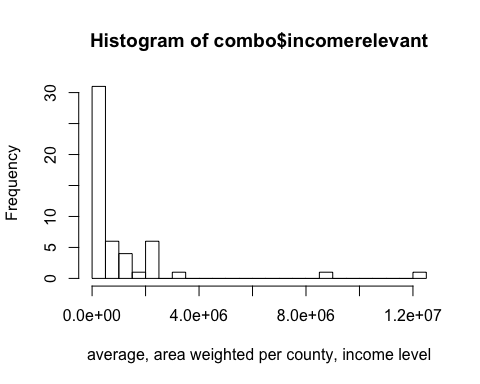
#view distribution of employment  
hist(combo$employmentsumrelevant, xlab ="average, area weighted per county, employment level", breaks = 30)



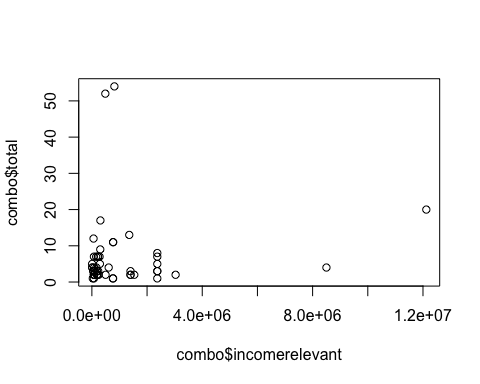
#response variable or total number of partners  
plot(combo$employmentsumrelevant, combo$total)



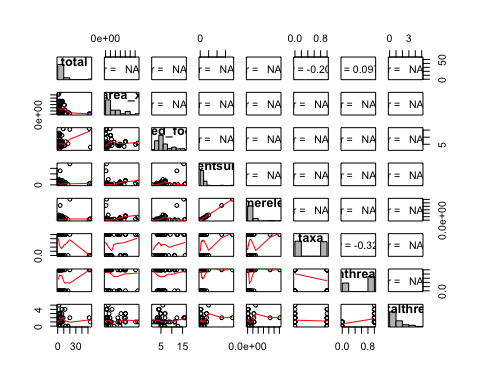
#view distribution of employment  
hist(combo$incomerelevant, xlab ="average, area weighted per county, income level", breaks = 30)



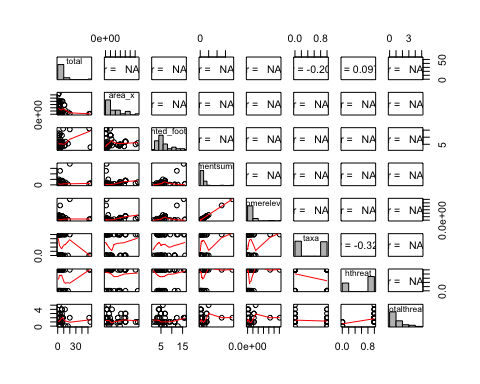
#response variable or total number of partners  
plot(combo$incomerelevant, combo$total)



# Correlation Matrix of Predictors vs Responses #######  
## modified some code from http://handlesman.blogspot.com/2011/03/matrix-plot-with-confidence-intervals.html  
  
#set up  
  
PredictorsOnlyPixel <- combo[,c(3)]  
  
PredictAndResponsePixel <- combo[,-c(1,2, 5,7)]  
  
PredictAndResponseGrid <- combo[,-c(1,2, 5,7)]  
   
# 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 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="everything")   
 txt <- format(c(r, 0.123456789), digits=digits)[1]  
 prefix <- "r = "  
 rc <- cor.test(x,y,method = c("pearson")) ## calculate pearsons rho for upper grid  
 txt <- paste(prefix,txt,sep="")  
 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[1:8], lower.panel=panel.smooth, cex = .8, diag.panel=panel.hist, cex.labels = 1.2, font.labels=2, upper.panel=panel.cor)



##Pixel level  
  
pairs(PredictAndResponsePixel,lower.panel = panel.smooth, diag.panel=panel.hist,upper.panel=panel.cor)



## Grid level  
  
pairs(PredictAndResponseGrid,lower.panel = panel.smooth, diag.panel=panel.hist,upper.panel=panel.cor)  
  
  
### Predictors only --> only have one response so not going to run this one  
##pairs(PredictorsOnlyPixel,lower.panel = panel.smooth, diag.panel=panel.hist,upper.panel=panel.cor)  
  
  
#test correlations to look for covariation between predictors  
#cor(combo$total, combo$taxa) # matrix seems fine