ESA\_Recovery\_overview\_doc

Annabelle

6/7/2021

##Introduction We are trying to answer the following Qs (which are the same as in the “Overview” doc) 1. What are the conservation initiatives for a species 2. What are the actions specified in the federal register important for the species? 3. Who are the organizations/players involved in doing these conservation initiatives? 4. What are the actions each organization is doing (optional?)

This document has the summary statistics for questions 1 & 2 (added 06/07/2021) and each section is question specific

Notes - Slight issue - missing at least 10 species data from tyler for initial draft of data (10 out of 38 species) … is this data accessible from the drives??

##Set up file structure

##load in different datasets

##Modify datasets

##Modify datasets Tyler Variables needed: # range size –> area c32 # area weighted footprint –> c33 ?have options # total threat count –> c 26 # type of threat –> c27 # employment (all relevant industries) -> 55:59 [avg\_employment ???] # percent public and percent federal c66, c68 # diversity of landowners - shannons H (land use complextity) # taxa (plants v animals // vert v invert) taxa c16

I think *missing* private lands?? (can get with TdataOLD because had variable information there..)

missing species (presumming Tyler dropped them..) # Euphilotes pallescens arenamontana # Nysius wekiuicola

Species Name changes/differences accross datasets Chorizanthe parryi var fernandina -> Chorizanthe parryi var. fernandina Moxostoma sp 2 -> Moxostoma sp.

tdata\_mod <- tdata %>% select(scientific\_name, common\_name, vert\_vs\_invert, species\_group, number\_of\_counties, range\_area, habitat\_modification, overutilization, pollution, species\_species\_interactions, env\_stochasticity, demographic\_stochasticity, total\_threats, public\_mang\_shannons\_diversity, avg\_employment)  
  
#put together with current list of species [doing Proactive dataset - may need to merge again]  
  
t\_pro\_df <- left\_join(proactive2, tdata\_mod, by = c("scientific\_name"))  
t\_pro\_df <- t\_pro\_df %>% select(-common\_name.y)  
t\_pro\_df$common\_name <- t\_pro\_df$common\_name.x  
t\_pro\_df <- t\_pro\_df %>% select(-common\_name.x)  
  
## Need to add values for % public/ private (located in tdata) bc not in new dataset   
## AND get info for species that have been recently removed from tyler's current dataset   
  
percent\_data <- tdataOLD %>% select(scientific\_name, common\_name, percentpublic:percentfederal)  
  
df <- left\_join(t\_pro\_df, percent\_data, by = c("common\_name"))  
df <- df %>% select(-scientific\_name.y)  
df$scientific\_name <- df$scientific\_name.x  
df <- df %>% select(-scientific\_name.x) #changing col name back  
  
## Need to rearrange column order  
  
df <- df[,c(27,23,1:22,24:26)]

## 1. What are the conservation initiatives for a species

#### 1 - Summary stats

* Number of iniatitves per species

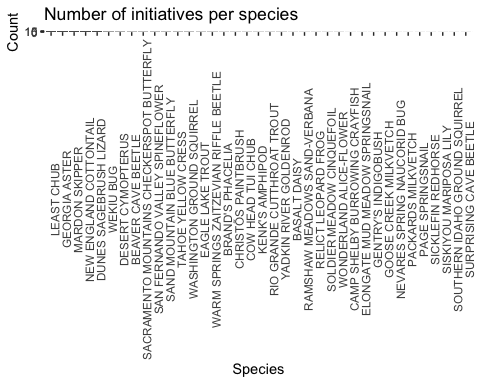
# iniatives per species (mean, median, quartiles)

* Count of efforts, agreements, groups per species

#### - Number of initiatives per species  
  
#number of partners in each of the agreements  
no\_init\_per\_sp <- sub\_initiatves\_partners %>% group\_by(common\_name, RowSpecInfo) %>% distinct(Agreement\_Name) %>% #count how many names are distinct   
summarise(n())

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

## number of initiatives per species   
no\_init\_per\_sp$count <- no\_init\_per\_sp$`n()`   
no\_init\_per\_sp\_2 <- no\_init\_per\_sp %>% group\_by(common\_name) %>%  
summarize(sum = sum(count))  
  
ggplot(no\_init\_per\_sp\_2) + geom\_bar(mapping = aes(x = reorder(common\_name, -sum), y = sum), stat = "identity")+ theme(axis.text.x = element\_text(angle = 90)) + ggtitle("Number of initiatives per species") + scale\_x\_discrete(name ="Species") + scale\_y\_continuous(name ="Count")



##### - # iniatives per species (mean, median, quartiles)  
  
summary(no\_init\_per\_sp\_2)

## common\_name sum   
## Length:37 Min. : 1.000   
## Class :character 1st Qu.: 1.000   
## Mode :character Median : 3.000   
## Mean : 4.081   
## 3rd Qu.: 5.000   
## Max. :15.000

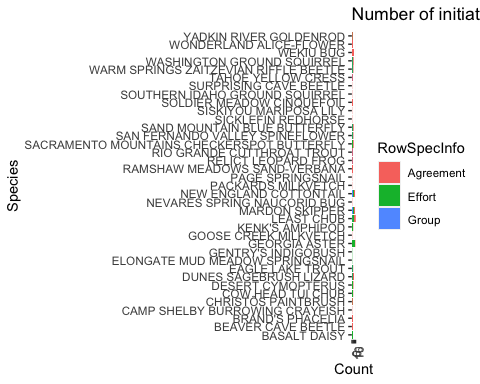
##### - Count of efforts, agreements, groups per species  
  
no\_init\_per\_sp <- sub\_initiatves\_partners %>% group\_by(common\_name, RowSpecInfo) %>% distinct(Agreement\_Name) %>% #count how many names are distinct   
summarise(n())

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

## need to figure out how to add "Landowner" and "document" to other categories to count types of efforts   
no\_init\_per\_sp[which(no\_init\_per\_sp$RowSpecInfo == "Land Owner"),2] <- "Effort"  
no\_init\_per\_sp[which(no\_init\_per\_sp$RowSpecInfo == "Document"),2] <- "Agreement"  
  
no\_init\_per\_sp$count <- no\_init\_per\_sp$`n()`   
  
no\_init\_per\_sp\_3 <- no\_init\_per\_sp %>% group\_by(common\_name, RowSpecInfo) %>%  
summarize(sum = sum(count))

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

ggplot(no\_init\_per\_sp\_3, aes(common\_name, sum)) +   
 geom\_bar(aes(fill = RowSpecInfo), position = "stack", stat="identity") + theme(axis.text.x = element\_text(angle = 90)) + ggtitle("Number of initiatives per species") + scale\_x\_discrete(name ="Species") + scale\_y\_continuous(name ="Count") + coord\_flip()



#### 1 - Regression Predictor Modification

* Working group mentioned – logistic regression
* Numbers of efforts or agreements – linear regression or negative binomial or poisson regression (since count data)

NoIniv <- no\_init\_per\_sp\_2   
names(NoIniv) <- c("common\_name", "no\_iniv")  
  
L\_df <- full\_join(df, NoIniv)

## Joining, by = "common\_name"

## taxa (coding for plants)  
plant <- c(0)  
L\_df <- add\_column(L\_df, plant)  
L\_df[which(L\_df$vert\_vs\_invert == "P"),29] <- 1  
  
reg\_prdctr <- L\_df %>% select(scientific\_name, common\_name, no\_iniv, range\_area, percentpublic, plant, public\_mang\_shannons\_diversity, total\_threats)  
  
skim(reg\_prdctr)

Data summary

|  |  |
| --- | --- |
| Name | reg\_prdctr |
| Number of rows | 37 |
| Number of columns | 8 |
| \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |  |
| Column type frequency: |  |
| character | 2 |
| numeric | 6 |
| \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |  |
| Group variables | None |

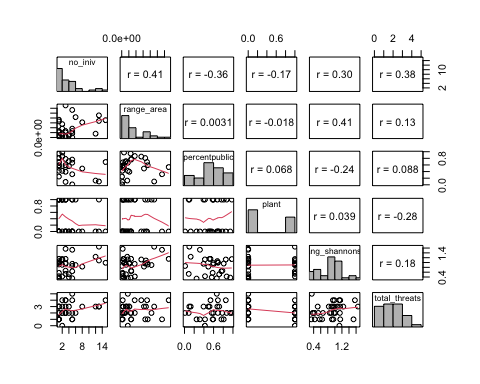
**Variable type: character**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| skim\_variable | n\_missing | complete\_rate | min | max | empty | n\_unique | whitespace |
| scientific\_name | 0 | 1 | 13 | 35 | 0 | 37 | 0 |
| common\_name | 0 | 1 | 9 | 42 | 0 | 37 | 0 |

**Variable type: numeric**

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| skim\_variable | n\_missing | complete\_rate | mean | sd | p0 | p25 | p50 | p75 | p100 | hist |
| no\_iniv | 0 | 1.00 | 4.080000e+00 | 3.750000e+00 | 1.00 | 1.000000e+00 | 3.00000e+00 | 5.000000e+00 | 1.500000e+01 | ▇▃▁▁▁ |
| range\_area | 2 | 0.95 | 3.304669e+10 | 3.478989e+10 | 784468319.00 | 6.509527e+09 | 2.10398e+10 | 5.869356e+10 | 1.317573e+11 | ▇▁▂▁▁ |
| percentpublic | 9 | 0.76 | 5.500000e-01 | 2.600000e-01 | 0.01 | 4.100000e-01 | 5.90000e-01 | 7.300000e-01 | 9.800000e-01 | ▃▂▆▇▅ |
| plant | 0 | 1.00 | 4.100000e-01 | 5.000000e-01 | 0.00 | 0.000000e+00 | 0.00000e+00 | 1.000000e+00 | 1.000000e+00 | ▇▁▁▁▆ |
| public\_mang\_shannons\_diversity | 2 | 0.95 | 9.000000e-01 | 3.500000e-01 | 0.33 | 6.100000e-01 | 9.50000e-01 | 1.130000e+00 | 1.650000e+00 | ▆▂▇▃▂ |
| total\_threats | 2 | 0.95 | 2.370000e+00 | 1.170000e+00 | 0.00 | 1.500000e+00 | 2.00000e+00 | 3.000000e+00 | 5.000000e+00 | ▇▇▇▃▁ |

# Correlation Matrix of Predictors vs Responses ##  
## modified some code from http://handlesman.blogspot.com/2011/03/matrix-plot-with-confidence-intervals.html  
#Not working because of NAs in data - getting error that x is not numeric   
  
  
#set up  
PredictorsOnlyPixel <- reg\_prdctr[,c(3)]  
PredictAndResponsePixel <- reg\_prdctr[,-c(1:2)]  
PredictAndResponseGrid <- reg\_prdctr[,-c(1:2)]  
   
# 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="complete.obs")   
 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[1:6], 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)



### 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(reg\_prdctr[,-c(1:2)], use = "complete.obs") # matrix seems fine

## no\_iniv range\_area percentpublic  
## no\_iniv 1.0000000 0.345049553 -0.356742035  
## range\_area 0.3450496 1.000000000 0.003068972  
## percentpublic -0.3567420 0.003068972 1.000000000  
## plant -0.1801557 -0.071621006 0.068424804  
## public\_mang\_shannons\_diversity 0.3651009 0.523745818 -0.242611010  
## total\_threats 0.4245328 0.201218934 0.088290002  
## plant public\_mang\_shannons\_diversity  
## no\_iniv -0.18015565 0.3651009  
## range\_area -0.07162101 0.5237458  
## percentpublic 0.06842480 -0.2426110  
## plant 1.00000000 0.1455520  
## public\_mang\_shannons\_diversity 0.14555198 1.0000000  
## total\_threats -0.37541923 0.1190053  
## total\_threats  
## no\_iniv 0.4245328  
## range\_area 0.2012189  
## percentpublic 0.0882900  
## plant -0.3754192  
## public\_mang\_shannons\_diversity 0.1190053  
## total\_threats 1.0000000

#no na.rm in this one instead it's "use"

VIFS

vif(lm(no\_iniv ~ range\_area +percentpublic + plant + public\_mang\_shannons\_diversity + total\_threats,data = reg\_prdctr))

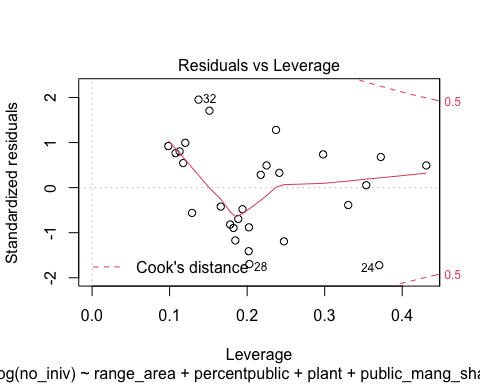
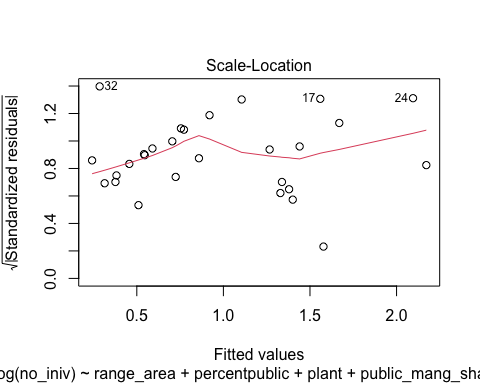
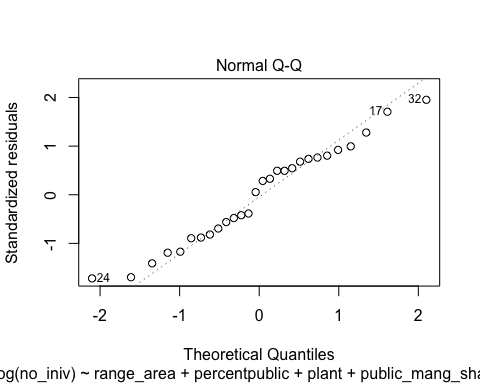
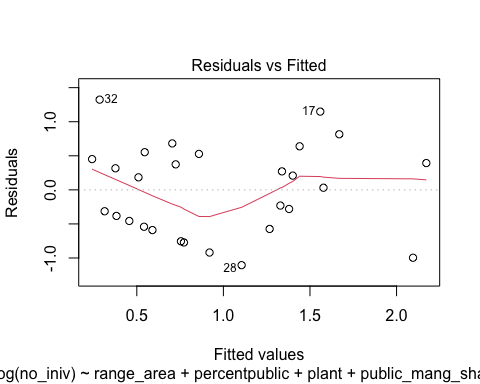
## range\_area percentpublic   
## 1.477608 1.142588   
## plant public\_mang\_shannons\_diversity   
## 1.281775 1.636317   
## total\_threats   
## 1.255091

Q - keep log???

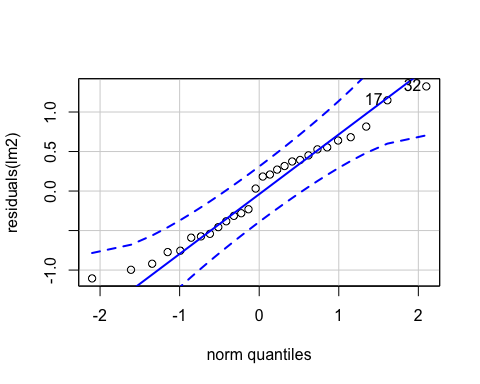
lm2 <- lm(log(no\_iniv) ~ range\_area +percentpublic + plant + public\_mang\_shannons\_diversity + total\_threats,data = reg\_prdctr)  
  
#withou log   
#lm2 <- lm((no\_iniv) ~ range\_area +percentpublic + plant + public\_mang\_shannons\_diversity + total\_threats,data = reg\_prdctr)  
  
summary(lm2)

##   
## Call:  
## lm(formula = log(no\_iniv) ~ range\_area + percentpublic + plant +   
## public\_mang\_shannons\_diversity + total\_threats, data = reg\_prdctr)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -1.1054 -0.5497 0.1075 0.4704 1.3240   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 7.367e-01 5.873e-01 1.254 0.2228   
## range\_area 5.482e-12 4.688e-12 1.169 0.2548   
## percentpublic -1.283e+00 5.682e-01 -2.258 0.0342 \*  
## plant 3.246e-02 3.157e-01 0.103 0.9190   
## public\_mang\_shannons\_diversity 4.829e-02 4.995e-01 0.097 0.9239   
## total\_threats 3.171e-01 1.340e-01 2.366 0.0272 \*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.7302 on 22 degrees of freedom  
## (9 observations deleted due to missingness)  
## Multiple R-squared: 0.4115, Adjusted R-squared: 0.2777   
## F-statistic: 3.076 on 5 and 22 DF, p-value: 0.02965

#kable(table(lm2$coefficients))  
  
par(mfrow = c(1, 1))  
plot(lm2)



qqPlot(residuals(lm2))



## 32 17   
## 24 15

par(mfrow = c(1, 1))  
  
#plot(predicted(lm2), residuals(lm2))  
#hist(residuals(lm2))  
# how add CI equivalent around q-q plot?   
AIC(lm2)

## [1] 69.09574

#check\_overdispersion(lm2)

## 2. What are the actions specified in the federal register important for the species?

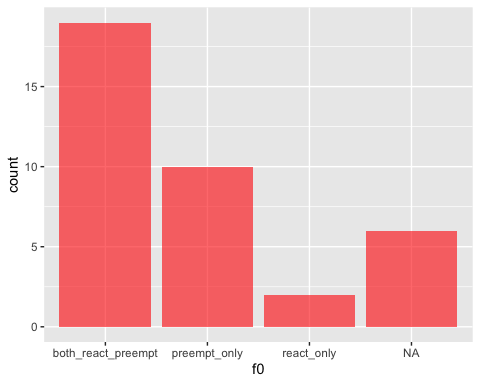
#### 1 - Summary stats

outcomes <- df  
  
outcomes[,c(3:11)] <- sapply(outcomes[ ,c(3:11)], as.numeric)  
  
### ### ### ### ### setting up data mannipulations (unsure if need) by making new columns   
  
### ### ### ### Actions  
outcomes <- outcomes %>% rowwise() %>% mutate("preempt" = sum(preempt\_private, preempt\_public, preempt\_unclear)) %>% #rowSums with mutate (need rowwise)   
 mutate\_at(vars(preempt), ~1 \* (. > 0)) #changed all values back to ones and zeros ## this doesn't work because changes whole df   
  
outcomes <- outcomes %>% rowwise() %>% mutate("react" = sum(respon\_private, respon\_public, respon\_unclear)) %>% #rowSums with mutate (need rowwise)  
 mutate\_at(vars(react), ~1 \* (. > 0)) #changed all values back to ones and zeros   
  
outcomes <- outcomes %>% rowwise() %>% mutate("both\_react\_preempt" = sum(react, preempt)) %>% #rowSums with mutate (need rowwise)  
 mutate\_at(vars(both\_react\_preempt), ~1 \* (. >= 2)) # change all specified mutate functions back to ones and zeros   
  
outcomes <- outcomes %>% rowwise() %>% mutate("react\_only" = react - both\_react\_preempt) #%>% #rowSums with mutate (need rowwise)  
# mutate\_at(vars(react\_only), ~"react\_only" \* (. > 0)) #changed all values back to ones and zeros   
  
outcomes <- outcomes %>% rowwise() %>% mutate("preempt\_only" = preempt - both\_react\_preempt)  
  
#outcomes %>% mutate(vars(both\_react\_preempt:preempt\_only), funs(ifelse(. == 1, deparse(substitute(.)), .)))  
  
## trying to get a single column with preempt, protect, or both as cell values for ggplot ease  
outcomes$actionresults <- apply(outcomes[,30:32], 1, function(x) max(names(which(x >0))))

## Warning in max(names(which(x > 0))): no non-missing arguments, returning NA  
  
## Warning in max(names(which(x > 0))): no non-missing arguments, returning NA  
  
## Warning in max(names(which(x > 0))): no non-missing arguments, returning NA  
  
## Warning in max(names(which(x > 0))): no non-missing arguments, returning NA  
  
## Warning in max(names(which(x > 0))): no non-missing arguments, returning NA  
  
## Warning in max(names(which(x > 0))): no non-missing arguments, returning NA

## Bar chart Preempt: proactive, protective, both  
ggplot(outcomes, aes(f0)) +   
 geom\_histogram(mapping = aes(x = actionresults), stat = "count", fill = "red", alpha = 0.6, position = "dodge")

## Warning: Ignoring unknown parameters: binwidth, bins, pad

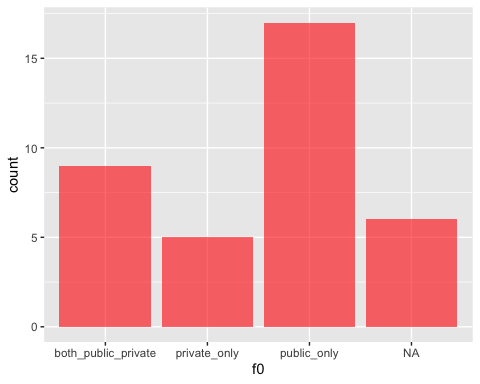


### ### Land  
outcomes <- outcomes %>% rowwise() %>% mutate("public" = sum(preempt\_public, public\_unclear, respon\_public)) %>% #rowSums with mutate (need rowwise)   
 mutate\_at(vars(public), ~1 \* (. > 0)) #changed all values back to ones and zeros ## this doesn't work because changes whole df   
  
outcomes <- outcomes %>% rowwise() %>% mutate("private" = sum(respon\_private, private\_unclear, preempt\_private)) %>% #rowSums with mutate (need rowwise)  
 mutate\_at(vars(private), ~1 \* (. > 0)) #changed all values back to ones and zeros   
  
outcomes <- outcomes %>% rowwise() %>% mutate("both\_public\_private" = sum(public, private)) %>% #rowSums with mutate (need rowwise)  
 mutate\_at(vars(both\_public\_private), ~1 \* (. >= 2)) # change all specified mutate functions back to ones and zeros   
  
outcomes <- outcomes %>% rowwise() %>% mutate("public\_only" = public - both\_public\_private) #%>% #rowSums with mutate (need rowwise)  
# mutate\_at(vars(react\_only), ~"react\_only" \* (. > 0)) #changed all values back to ones and zeros   
  
outcomes <- outcomes %>% rowwise() %>% mutate("private\_only" = private - both\_public\_private)  
  
#outcomes %>% mutate(vars(both\_react\_preempt:preempt\_only), funs(ifelse(. == 1, deparse(substitute(.)), .)))  
  
## trying to get a single column with preempt, protect, or both as cell values for ggplot ease  
outcomes$landresults <- apply(outcomes[,36:38], 1, function(x) max(names(which(x >0))))

## Warning in max(names(which(x > 0))): no non-missing arguments, returning NA  
  
## Warning in max(names(which(x > 0))): no non-missing arguments, returning NA  
  
## Warning in max(names(which(x > 0))): no non-missing arguments, returning NA  
  
## Warning in max(names(which(x > 0))): no non-missing arguments, returning NA  
  
## Warning in max(names(which(x > 0))): no non-missing arguments, returning NA  
  
## Warning in max(names(which(x > 0))): no non-missing arguments, returning NA

## Barchart: Land: public, private, both  
ggplot(outcomes, aes(f0)) +   
 geom\_histogram(mapping = aes(x = landresults), stat = "count", fill = "red", alpha = 0.6, position = "dodge")

## Warning: Ignoring unknown parameters: binwidth, bins, pad



### ### ### ### ### ### Taxa   
  
taxa <- outcomes  
taxa <- taxa %>% select(scientific\_name, common\_name, preempt\_private:species\_group, preempt:landresults)  
  
## Filling in missing data   
taxa[which(is.na(taxa$vert\_vs\_invert)),12] <- c("I")  
taxa[which(is.na(taxa$species\_group)),13] <- "Insects"   
  
### ### ### Taxa -- Actions  
 ### all species, plant species, animal species (outcome: preempt)  
  
taxa\_small <- taxa %>% select(scientific\_name, actionresults, vert\_vs\_invert )  
  
## trying to get the following data frame   
#| | both| preempt | protect|  
#|:------|-----|---------|-------:|  
#|all sp | | | |  
#|plant | | | |  
#|animal | | | |  
  
# Doing so by subsetting by animal type and then will recombine   
  
plants <- taxa\_small %>% filter(vert\_vs\_invert == "P") %>% ## Select which species type to subset   
pivot\_wider(names\_from = actionresults, values\_from = vert\_vs\_invert) %>% ## may action types col names  
 as.data.frame() %>% # turn to dataframe  
 replace(!is.na(.), 1) # convert all values to 1s so can col sum below   
plants[,c(1:5)] <- sapply(plants[ ,c(1:5)], as.numeric)  
plants <- colSums(plants[,c(2:5)], na.rm = T)  
plants <- as.data.frame(plants) %>% rownames\_to\_column()  
  
invert <- taxa\_small %>% filter(vert\_vs\_invert == "I") %>% ## Select which species type to subset   
pivot\_wider(names\_from = actionresults, values\_from = vert\_vs\_invert) %>% ## may action types col names  
 as.data.frame() %>% # turn to dataframe  
 replace(!is.na(.), 1) # convert all values to 1s so can col sum below   
invert[,c(1:4)] <- sapply(invert[ ,c(1:4)], as.numeric)  
invert <- colSums(invert[,c(2:4)], na.rm = T)   
invert <- as.data.frame(invert) %>% rownames\_to\_column()  
 #rownames\_to\_column()   
  
vert <- taxa\_small %>% filter(vert\_vs\_invert == "V") %>% ## Select which species type to subset   
pivot\_wider(names\_from = actionresults, values\_from = vert\_vs\_invert) %>% ## may action types col names  
 as.data.frame() %>% # turn to dataframe  
 replace(!is.na(.), 1) # convert all values to 1s so can col sum below   
vert[,c(1:3)] <- sapply(vert[ ,c(1:3)], as.numeric)  
vert <- colSums(vert[,c(2:3)], na.rm = T)   
vert <- as.data.frame(vert) %>% rownames\_to\_column()  
  
species <- full\_join(plants, vert, by = "rowname")  
species <- full\_join(species, invert, by = "rowname")  
species <- species %>% rowwise() %>% mutate("all\_sp" = sum(plants,vert, invert, na.rm = T))  
   
### make bar graph   
  
s <- species  
rownames(s) <- c("NA", "react\_only", "both\_react\_preempt", "preempt\_only" )

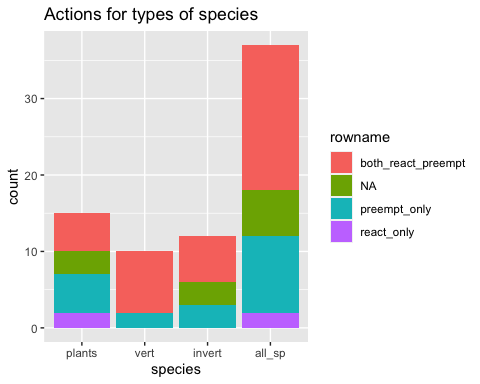
## Warning: Setting row names on a tibble is deprecated.

s <- as.data.frame(s)  
#s <- s[,-c(1)]  
smelt <- s %>% melt()

## Using rowname as id variables

ggplot(smelt) + aes(x=variable, y=value, fill=rowname) +  
 geom\_bar(stat="identity") +   
 scale\_x\_discrete(name ="species") +   
 ggtitle("Actions for types of species") +   
 scale\_y\_continuous(name = "count")

## Warning: Removed 3 rows containing missing values (position\_stack).



### ### ### Taxa -- Land   
 ### all species, plant species, animal species (outcome: land)  
  
taxa\_land <- taxa %>% select(scientific\_name, landresults, vert\_vs\_invert )  
  
plants <- taxa\_land %>% filter(vert\_vs\_invert == "P") %>% ## Select which species type to subset   
pivot\_wider(names\_from = landresults, values\_from = vert\_vs\_invert) %>% ## may action types col names  
 as.data.frame() %>% # turn to dataframe  
 replace(!is.na(.), 1) # convert all values to 1s so can col sum below   
plants[,c(1:5)] <- sapply(plants[ ,c(1:5)], as.numeric)  
plants <- colSums(plants[,c(2:5)], na.rm = T)  
plants <- as.data.frame(plants) %>% rownames\_to\_column()  
  
invert <- taxa\_land %>% filter(vert\_vs\_invert == "I") %>% ## Select which species type to subset   
pivot\_wider(names\_from = landresults, values\_from = vert\_vs\_invert) %>% ## may action types col names  
 as.data.frame() %>% # turn to dataframe  
 replace(!is.na(.), 1) # convert all values to 1s so can col sum below   
invert[,c(1:4)] <- sapply(invert[ ,c(1:4)], as.numeric)  
invert <- colSums(invert[,c(2:4)], na.rm = T)   
invert <- as.data.frame(invert) %>% rownames\_to\_column()  
 #rownames\_to\_column()   
  
vert <- taxa\_land %>% filter(vert\_vs\_invert == "V") %>% ## Select which species type to subset   
pivot\_wider(names\_from = landresults, values\_from = vert\_vs\_invert) %>% ## may action types col names  
 as.data.frame() %>% # turn to dataframe  
 replace(!is.na(.), 1) # convert all values to 1s so can col sum below   
vert[,c(1:5)] <- sapply(vert[ ,c(1:5)], as.numeric)  
vert <- colSums(vert[,c(2:5)], na.rm = T)   
vert <- as.data.frame(vert) %>% rownames\_to\_column()  
  
species\_actions <- full\_join(plants, vert, by = "rowname")  
species\_actions <- full\_join(species\_actions, invert, by = "rowname")  
species\_actions <- species\_actions %>% rowwise() %>% mutate("all\_sp" = sum(plants,vert, invert, na.rm = T))  
   
### make bar graph   
  
sactions <- species\_actions  
rownames(sactions) <- c("public\_only", "NA", "both\_public\_private", "private\_only" )

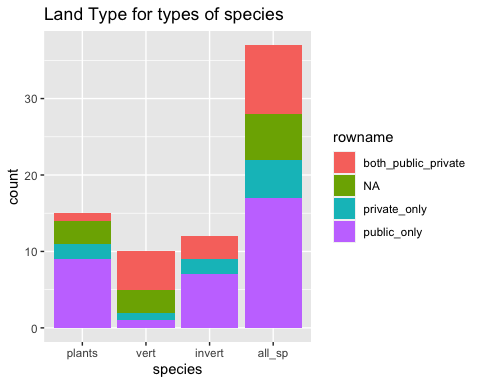
## Warning: Setting row names on a tibble is deprecated.

sactions <- as.data.frame(sactions)  
#s <- s[,-c(1)]  
sactions\_melt <- sactions %>% melt()

## Using rowname as id variables

ggplot(sactions\_melt) + aes(x=variable, y=value, fill=rowname) +  
 geom\_bar(stat="identity") +   
 scale\_x\_discrete(name ="species") +   
 ggtitle("Land Type for types of species") +   
 scale\_y\_continuous(name = "count")

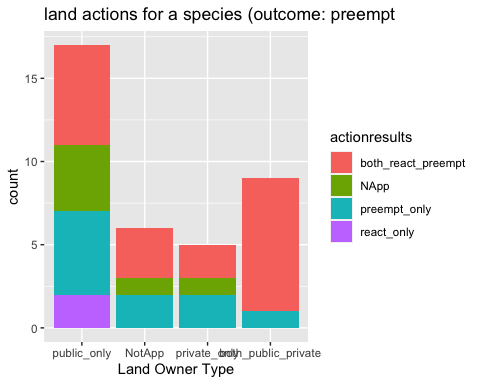
## Warning: Removed 1 rows containing missing values (position\_stack).



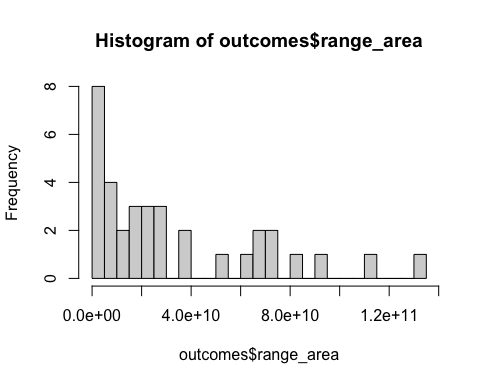
### ### ### ### ### ### ### ### ### ### ### ### ### Land  
  
### ### ### ### ### Land -- Preempt   
 ### public land actions for a species, private land actions for a species (outcome: preempt)  
  
land\_preempt <- outcomes  
land\_preempt <- land\_preempt %>% select(scientific\_name, actionresults, landresults)  
  
land\_preempt$number <- rownames(land\_preempt)  
  
  
land\_preempt\_trial <- land\_preempt %>% pivot\_wider(values\_from = number, names\_from = landresults)  
  
land\_preempt\_trial[,c(3:6)] <- sapply(land\_preempt\_trial[ ,c(3:6)], as.numeric)  
  
land\_preempt\_trial <- land\_preempt\_trial %>% mutate\_if(is.numeric, ~1 \* (. > 0)) #changed all values back to ones and zeros  
  
land\_preempt\_trial <- as.data.table(land\_preempt\_trial)  
names(land\_preempt\_trial) <- c( "scientific\_name" , "actionresults" , "public\_only" , "NotApp" , "private\_only" , "both\_public\_private" )  
  
land\_preempt\_trial[which(is.na(land\_preempt\_trial$actionresults)),2] <- "NApp"   
  
#Want to use equivalent of group by and then colsum (tidyr version didn't work but the datatable method does)  
count <- land\_preempt\_trial[,list(public\_only=sum(public\_only, na.rm = T), NotApp=sum(NotApp, na.rm = T), private\_only = sum(private\_only, na.rm = T), both\_public\_private = sum(both\_public\_private, na.rm = T)), by='actionresults']  
  
## Couldn't get this to work  
#land\_preempt\_trial <- land\_preempt\_trial %>% select(-scientific\_name) %>% group\_by(actionresults) %>%   
#summarize\_at(var(react\_only, NApp, preempt\_only, both\_react\_preempt), fun = colSums)  
# summarise\_if(is.integer, sum, na.rm = T)  
  
act\_land <- count %>% melt()

## Using actionresults as id variables

ggplot(act\_land) + aes(x=variable, y=value, fill=actionresults) +  
 geom\_bar(stat="identity") +   
 scale\_x\_discrete(name ="Land Owner Type") +   
 ggtitle("land actions for a species (outcome: preempt") +   
 scale\_y\_continuous(name = "count")



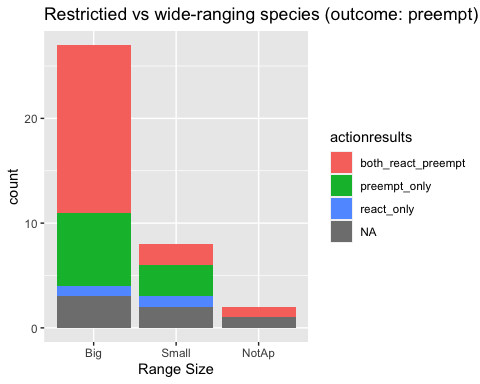
### ### ### ### ### Range size -- Preempt   
 ### restricted versus wide-ranging species? (outcome: preempt)  
## Note - this seems to really depend on what you set the area cutoff to  
  
hist(outcomes$range\_area, breaks = 40)



## let's do 2 categories of small v large range (split point = 2.0e + 10)  
  
range <- outcomes  
range <- range %>% select(scientific\_name, range\_area, actionresults)  
range <- range %>% mutate(size = range\_area < (.5\*10^10))  
#Want to use equivalent of group by and then colsum (tidyr version didn't work but the datatable method does)  
  
range$number <- 1  
range <- range %>% pivot\_wider(values\_from = number, names\_from = size)  
names(range) <- c("scientific\_name" ,"range\_area" , "actionresults" , "Big", "Small" , "NotAp" )  
range <- as.data.table(range)  
count\_range <- range[,list(Big=sum(Big, na.rm = T), Small=sum(Small, na.rm = T), NotAp = sum(NotAp, na.rm = T)), by='actionresults']  
  
count\_range <- count\_range %>% melt()

## Using actionresults as id variables

ggplot(count\_range) + aes(x=variable, y=value, fill=actionresults) +  
 geom\_bar(stat="identity") +   
 scale\_x\_discrete(name ="Range Size") +   
 ggtitle("Restrictied vs wide-ranging species (outcome: preempt)") +   
 scale\_y\_continuous(name = "count")



### ### ### ### ### Range size -- Preempt   
 ### presence of each threat type (outcome: preempt)  
  
threat\_type <- outcomes  
threat\_type <- threat\_type %>% select(scientific\_name, habitat\_modification:total\_threats, actionresults)  
  
# note - going to do this independently (so for each threat type not by species.. slightly different than other ones)  
  
habitat <- threat\_type %>% filter(habitat\_modification == 1) %>% ## Select which species type to subset   
pivot\_wider(names\_from = actionresults, values\_from = habitat\_modification) %>% ## may action types col names  
 as.data.frame() # turn to dataframe  
habitat[,c(8:11)] <- sapply(habitat[ ,c(8:11)], as.numeric)  
habitat <- colSums(habitat[,c(8:11)], na.rm = T)   
habitat <- as.data.frame(habitat) %>% rownames\_to\_column()  
  
#only one instance for this one  
overut <- threat\_type %>% filter(overutilization == 1) %>% ## Select which species type to subset   
pivot\_wider(names\_from = actionresults, values\_from = overutilization) %>% ## may action types col names  
 as.data.frame() # turn to dataframe  
overut[,c(8)] <- sapply(overut[ ,c(8)], as.numeric)  
overut <- colSums(overut[,c(7:8)], na.rm = T)   
overut <- as.data.frame(overut) %>% rownames\_to\_column()  
  
polu <- threat\_type %>% filter(pollution == 1) %>% ## Select which species type to subset   
pivot\_wider(names\_from = actionresults, values\_from = pollution) %>% ## may action types col names  
 as.data.frame() # turn to dataframe  
polu[,c(7:8)] <- sapply(polu[ ,c(7:8)], as.numeric)  
polu <- colSums(polu[,c(7:8)], na.rm = T)   
polu <- as.data.frame(polu) %>% rownames\_to\_column()  
  
spsp <- threat\_type %>% filter(species\_species\_interactions == 1) %>% ## Select which species type to subset   
pivot\_wider(names\_from = actionresults, values\_from = species\_species\_interactions) %>% ## may action types col names  
 as.data.frame() # turn to dataframe  
spsp[,c(8:11)] <- sapply(spsp[ ,c(8:11)], as.numeric)  
spsp <- colSums(spsp[,c(8:11)], na.rm = T)   
spsp <- as.data.frame(spsp) %>% rownames\_to\_column()  
  
env\_st <- threat\_type %>% filter(env\_stochasticity == 1) %>% ## Select which species type to subset   
pivot\_wider(names\_from = actionresults, values\_from = env\_stochasticity) %>% ## may action types col names  
 as.data.frame() # turn to dataframe  
env\_st[,c(8:10)] <- sapply(env\_st[ ,c(8:10)], as.numeric)  
env\_st <- colSums(env\_st[,c(8:10)], na.rm = T)   
env\_st <- as.data.frame(env\_st) %>% rownames\_to\_column()  
  
demo\_st <- threat\_type %>% filter(demographic\_stochasticity == 1) %>% ## Select which species type to subset   
pivot\_wider(names\_from = actionresults, values\_from = demographic\_stochasticity) %>% ## may action types col names  
 as.data.frame() # turn to dataframe  
demo\_st[,c(8:9)] <- sapply(demo\_st[ ,c(8:9)], as.numeric)  
demo\_st <- colSums(demo\_st[,c(8:9)], na.rm = T)   
demo\_st <- as.data.frame(demo\_st) %>% rownames\_to\_column()  
  
threats\_actions <- full\_join(habitat, overut) %>% full\_join(spsp) %>% full\_join(env\_st) %>% full\_join(demo\_st)

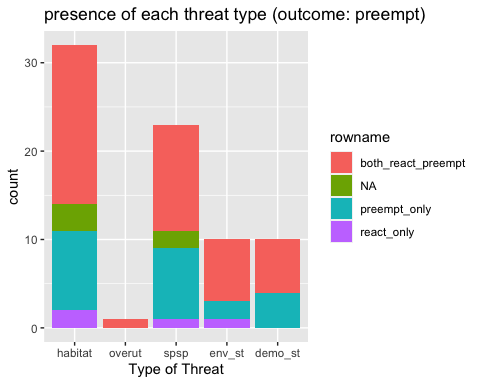
## Joining, by = "rowname"  
## Joining, by = "rowname"  
## Joining, by = "rowname"  
## Joining, by = "rowname"

threats\_actions <- threats\_actions[-which(is.na(threats\_actions$habitat)),]  
  
threats\_actions <- threats\_actions %>% melt()

## Using rowname as id variables

ggplot(threats\_actions) + aes(x=variable, y=value, fill=rowname) +  
 geom\_bar(stat="identity") +   
 scale\_x\_discrete(name ="Type of Threat") +   
 ggtitle("presence of each threat type (outcome: preempt)") +   
 scale\_y\_continuous(name = "count")

## Warning: Removed 6 rows containing missing values (position\_stack).



### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ###

Chi squares

# assumption: 5 or more in at least 80% of the cells, and no cell should have an expected of less than one (3).  
  
  
### ### ### Differences in frequency of proactive and protective measures on public versus private lands (Chi-square)  
  
differences <- outcomes  
differences <- as.data.frame(differences)  
differences[,c(3:10)] <- sapply(differences[ ,c(3:10)], as.numeric)  
count\_diff <- colSums(differences[,c(3:10)], na.rm = T)  
count\_diff <- as.data.frame(count\_diff)  
  
## manually entering into table   
  
diff\_tb <- data.frame('preempt' = c(13,18,19), 'respon' = c(1,12,13), 'unc' = c(7,13,0))  
rownames(diff\_tb) <- c("private", "public", "unc")  
  
kable(diff\_tb)

|  |  |  |  |
| --- | --- | --- | --- |
|  | preempt | respon | unc |
| private | 13 | 1 | 7 |
| public | 18 | 12 | 13 |
| unc | 19 | 13 | 0 |

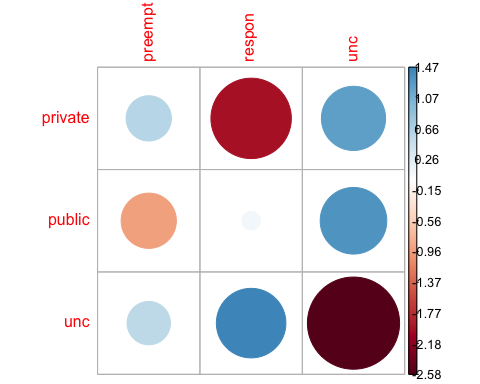
chsq <- chisq.test(diff\_tb)

## Warning in stats::chisq.test(x, y, ...): Chi-squared approximation may be  
## incorrect

chsq$residuals

## preempt respon unc  
## private 0.6236414 -1.9655341 1.254990  
## public -0.9288757 0.1037821 1.350352  
## unc 0.5715476 1.4719601 -2.581989

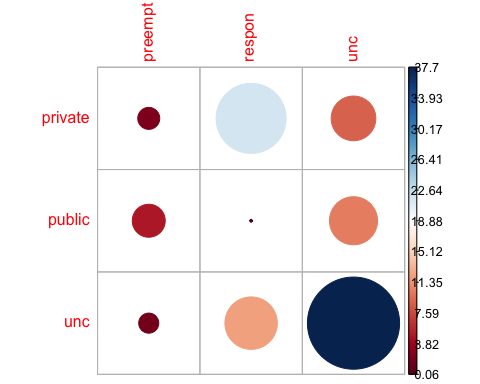
corrplot(chsq$residuals, is.cor = FALSE)



contrib <- 100\*chsq$residuals^2/chsq$statistic  
round(contrib, 3)

## preempt respon unc  
## private 2.199 21.846 8.906  
## public 4.879 0.061 10.311  
## unc 1.847 12.252 37.698

corrplot(contrib, is.cor = FALSE)



### ### ### Difference in protective versus proactive for plants versus animals  
  
act\_taxa <- s  
act\_taxa <- act\_taxa %>% select(-rowname)  
  
#add vert and invert to see if can get values up for chi square  
act\_taxa <- act\_taxa %>% mutate(allvert = vert + invert)  
act\_taxa <- act\_taxa %>% select(-vert, -invert)  
  
## \*\*\* issue here - values are still too low   
  
### ### ### Difference in public versus private lands actions for plant versus animal  
  
land\_taxa <- sactions  
land\_taxa <- land\_taxa %>% select(-rowname)  
rname <- rownames(land\_taxa)  
  
#add vert and invert to see if can get values up for chi square  
land\_taxa <- land\_taxa %>% mutate(allvert = vert + invert)  
land\_taxa <- land\_taxa %>% select(-vert, -invert)  
rownames(land\_taxa) <- rname  
  
## \*\*\* issue here - values are still too low

## 3. Who are the organizations/players involved in conserving the conservation of the species (eg in an agreement of effort)?

#### 3 - Data wrangling

partners <- sub\_initiatves\_partners  
  
# remove rows where there is an NA for partner  
##partners <- partners[-which(is.na(partners$RowSpecInfo)),] ## this might be redudent and cause issues later but for now needed (delete later if run into NA issues)  
  
#need to remove multi partner strings  
partners <- partners[-which(partners$Type\_of\_Org == "multi"),]  
  
#remove black "NA" partners  
partners <- partners[-which(is.na(partners$Partners)),]  
  
## count number of partners for each species  
## make two versions   
 # one normal  
 # the other where remove text after - (eg BLM-Suprise Feild office would be BLM)  
  
normal\_partner\_count <- partners %>% group\_by(scientific\_name) %>%   
 distinct(Partners) %>% summarise(n())   
#nothing removed or changed   
  
#remove offices  
sub\_partners <- partners  
sub\_partners$Partners <- gsub("(.\*)-.\*", "\\1", sub\_partners$Partners) ## Altered text from https://stackoverflow.com/questions/25307899/r-remove-anything-after-comma-from-columnn   
  
mod\_partner\_count <- sub\_partners %>% group\_by(scientific\_name) %>%   
 distinct(Partners) %>% summarise(n())   
  
  
# note - getting a bug when use duplicate so next time try match\_df from plyr   
   
  
# also want to do a check of spelling to see if sort partner names alphabetically, do any different verions show up (might not even be user error - some entities had name changes that were flagged in FR)  
# check complete - ran:   
 #mod\_partner\_count <- sub\_partners %>% group\_by(scientific\_name) %>% distinct(Partners)  
# and looks good   
  
### ### So need to add all remaining partners, re run and then decide on bins

o Summary stats… histograms/bar charts… (perhaps stacked bar charts where color code stacks for low, medium, high confidence in having collected most of partners)?

o To account for the uncertainty in number of partners, we put this information into bins and then “then use an ordered probit to look at how well variables predict the rough estimate of numbers of partners.”

o We use logit regression models to predict participation by different “types” of partners

test relationship between number of iniatives and number of partners

test <- full\_join(normal\_partner\_count, names)

## Joining, by = "scientific\_name"

test2 <- test %>% full\_join(no\_init\_per\_sp\_2, by = "common\_name")  
names(test2) <- c("scientific\_name", "part" , "common\_name" , "initi")  
  
plot(test2$initi ~ test2$part)

