Molokini Analysis

# Written by Steve Scherrer - July/August 2021

## Background

This notebook documents preliminary analysis of tracking data for fish tagged in Molokini Crater between 2020-05-16 and 2021-05-24.

The purpose of this study is to understand how human impacts affect the fish of Molokini Crater

We are particularly interested in answering the following hypotheses: 1. Is the presence of fish affected by vessel presence

1. Does the proportion of time fish are present within the crater negatively correlated with vessel presence?

Proposed Approach: 1. Begin by calculating the number of each species tagged and basic summary statistics 2. Calculate Metrics - Receiver Use - Pianka’s Niche Overlap - residency 3. Make the following plots - Map - Receiver locations - Map - Average receiver use by Species - Scatterplot - day night plots - Bar Plot - The number of detections per day (individual) - Bar Plot - The number of individuals detected (species) - Line Chart - The proportion of individuals detected n days after tagging (30 day moving average by species) - Bar Plot - Daily vessel traffic - Scatter Plot - vessel traffic vs. proportion of fish detected in crater daily (scatterplot by species) 4. Perform the following statistical Tests - Compare Residency Rates by Species - Compare residency by species, size, and time at liberty - Create a GLM comparing # of individuals in crater regressed against boat traffic and species using AR(1) term on dependent variable on some time scale (daily? 6 hours? depends on resolution of vessel data)

# Workspace Setup

## Establish Directory Heirarchy

project\_directory = '/Users/stephenscherrer/Documents/Programming/Projects/Molokini'  
scripts\_directory = file.path(project\_directory, 'Analysis Scripts')  
data\_directory = file.path(project\_directory, 'Data')  
results\_directory = file.path(project\_directory, 'Results')  
figure\_directory = file.path(results\_directory, 'Figures')

## Source package dependencies and utility functions from ‘Utility Functions.R’ file

## Load Data

* load various datafiles

## Files from VUE   
molo\_df = load\_vemco\_data(file.path(data\_directory, 'VUE\_Export.csv'))  
false\_detections\_df = load\_fdf\_report(file.path(data\_directory, 'FDA.csv'))  
  
## Vessel Traffic data  
vessel\_df = load\_vessel\_data(file.path(data\_directory, "Molokini\_Master\_June\_21.csv"))  
  
## Metadata Files  
tagging\_df = load\_tagging\_data(file.path(data\_directory, 'Molokini\_Fish\_Tagging\_master.xlsx'))  
  
# receiver\_df = load\_receiver\_data(file.path(data\_directory, ))

## Clean Data

* Associate detections with time of day (day, night, dawn, dusk)
* Remove detections from tags not associated with this study
* Remove false detections

## Associate detections with time of day  
molo\_df = get\_time\_of\_day(molo\_df)  
  
## Combine vue df with tagging df - remove irrelevant tags in the process  
molo\_df = inner\_join(x = molo\_df, y = tagging\_df[ ,c('tag\_id', 'species', 'fork\_length', 'tagging\_date' )], by = 'tag\_id')  
  
## Filter false detections  
# molo\_df = filter\_false\_detections(molo\_df)  
  
## Reclass vessel\_df date because R keeps implicitly recasting values...  
vessel\_df$date = as.POSIXct(as.numeric(vessel\_df$date), origin = '1970-01-01', tz = 'HST')  
  
## remove vessel data before start of study  
vessel\_df = vessel\_df[vessel\_df$date >= min(molo\_df$datetime), ]

# Exploratory Data Analysis

## Count of individuals tagged by species

## Get count of individuals tagged by species  
tags\_by\_species = aggregate(tag\_id ~ species, data = tagging\_df, FUN = uniqueN)  
 colnames(tags\_by\_species) = c('species', 'tagged')  
   
## Merge with count of individuals detected by species  
tags\_by\_species = left\_join(tags\_by\_species, aggregate(tag\_id ~ species, data = molo\_df, FUN = uniqueN), by = 'species')  
  
## Replace NA values with 0  
tags\_by\_species[is.na(tags\_by\_species)] = 0  
  
print(tags\_by\_species)

## species tagged tag\_id  
## 1 grey reef shark 2 2  
## 2 kahala 2 0  
## 3 omilu 4 4  
## 4 sandbar shark 1 1  
## 5 uku 1 0  
## 6 ulua 1 1  
## 7 whitetip reef shark 1 1

## Summary Statistics

# Time at liberty  
time\_at\_liberty = calculate\_time\_at\_liberty(molo\_df)  
  
# Days Detected  
days\_detected = calculate\_days\_detected(molo\_df)  
  
# % of days detected  
detection\_stats = merge(x = days\_detected, y = time\_at\_liberty[ ,c('tag\_id', 'days\_at\_liberty')], on.x = 'tag\_id', on.y = 'tag\_id')  
detection\_stats$percent\_days\_detected = round(detection\_stats$unique\_days / detection\_stats$days\_at\_liberty, 4) \* 100  
  
# Merge with tagging data to get fish info  
detection\_stats = merge(x = tagging\_df[ ,c('tagging\_date', 'species', 'tag\_id', 'fork\_length')], y = detection\_stats, on.x = 'tag\_id', on.y = 'tag\_id')  
detection\_stats = detection\_stats[order(detection\_stats$species, detection\_stats$tagging\_date, detection\_stats$tag\_id), ]  
print(detection\_stats)

## tag\_id tagging\_date species fork\_length unique\_days  
## 5 47513 2020-05-27 grey reef shark 93.0 360  
## 4 39194 2020-07-01 grey reef shark 101.0 60  
## 1 30711 2020-05-26 omilu 44.0 362  
## 2 30754 2020-05-27 omilu 38.0 362  
## 7 51591 2020-05-27 omilu 49.5 362  
## 6 51590 2020-05-28 omilu 37.0 360  
## 3 30755 2020-07-01 sandbar shark 112.0 25  
## 8 51593 2020-07-01 ulua 94.0 270  
## 9 51594 2020-07-23 whitetip reef shark 67.0 254  
## days\_at\_liberty percent\_days\_detected  
## 5 374 96.26  
## 4 78 76.92  
## 1 363 99.72  
## 2 363 99.72  
## 7 363 99.72  
## 6 361 99.72  
## 3 309 8.09  
## 8 327 82.57  
## 9 288 88.19

# Metric Calculations

## index of receiver use

## sum all spp, sum all individuals (detections of tag at given reciever / all detections of tag)  
  
## Calculate unique detections per tag per receiver station  
detections\_per\_tag\_per\_receiver = aggregate(datetime~tag\_id+receiver+species, data = molo\_df, FUN = uniqueN)  
colnames(detections\_per\_tag\_per\_receiver) = c('tag\_id', 'receiver', 'species', 'detections')  
  
## Calculate receiver use metric for each fish and receiver pair  
detections\_per\_tag\_per\_receiver$receiver\_use = 0  
for (species in detections\_per\_tag\_per\_receiver$species){  
 for (i in 1:nrow(detections\_per\_tag\_per\_receiver)){  
 detections\_per\_tag\_per\_receiver$receiver\_use[i] = detections\_per\_tag\_per\_receiver$detections[i] / sum(detections\_per\_tag\_per\_receiver$detections[detections\_per\_tag\_per\_receiver$tag\_id == detections\_per\_tag\_per\_receiver$tag\_id[i]])  
 }  
}  
  
## Calculate average receiver use metric for each tag - Omit stations with no use as this would bias metric  
indvidual\_receiver\_use = aggregate(receiver\_use~tag\_id+species, data = detections\_per\_tag\_per\_receiver[detections\_per\_tag\_per\_receiver$receiver\_use > 0, ], FUN = mean)  
  
## Add this information to detection\_stats  
detection\_stats = merge(detection\_stats, indvidual\_receiver\_use, on = 'tag\_id')  
  
## Calculate receiver use metric by species  
species\_receiver\_use = aggregate(receiver\_use~species, data = indvidual\_receiver\_use, FUN = mean)  
colnames(species\_receiver\_use) = c('species', 'receiver\_use')  
  
print(species\_receiver\_use)

## species receiver\_use  
## 1 grey reef shark 0.1428571  
## 2 omilu 0.1428571  
## 3 sandbar shark 0.2500000  
## 4 ulua 0.1428571  
## 5 whitetip reef shark 0.1428571

## Calculate Pianka’s Niche Overlap Index - Pianka (1973) The Structure of Lizard Communities

0 = no overlap, 1 = perfect overlap

## Aggregate data averaged by species  
receiver\_use\_aggregated\_by\_species = aggregate(receiver\_use ~ species + receiver , data = detections\_per\_tag\_per\_receiver, FUN = mean)  
 colnames(receiver\_use\_aggregated\_by\_species) = c('species', 'receiver', 'avg\_use\_index')  
   
## Reshape from Long to Wide format  
receiver\_use\_aggregated\_by\_species\_wide = dcast(receiver\_use\_aggregated\_by\_species, species ~ receiver)

## Using avg\_use\_index as value column: use value.var to override.

## Get all species combinations   
species\_combos = data.frame()  
for (i in 1:nrow(receiver\_use\_aggregated\_by\_species\_wide)){  
 if(i != nrow(receiver\_use\_aggregated\_by\_species\_wide)){  
 for (j in (i+1):nrow(receiver\_use\_aggregated\_by\_species\_wide)){  
 species\_combos = rbind(species\_combos, data.frame('species\_1' = receiver\_use\_aggregated\_by\_species\_wide$species[i], 'species\_2' = receiver\_use\_aggregated\_by\_species\_wide$species[j]))  
 }  
 }  
}  
  
## Change any NA values to zero  
receiver\_use\_aggregated\_by\_species\_wide[is.na(receiver\_use\_aggregated\_by\_species\_wide)] = 0  
  
## Calculate Pianka's index for all pairs  
species\_combos$pianka\_index = 0  
for(i in 1:nrow(species\_combos)){  
 species\_combos$pianka\_index[i] = sum(receiver\_use\_aggregated\_by\_species\_wide[receiver\_use\_aggregated\_by\_species\_wide$species == species\_combos$species\_1[i], -1] \*   
 receiver\_use\_aggregated\_by\_species\_wide[receiver\_use\_aggregated\_by\_species\_wide$species == species\_combos$species\_2[i], -1]) /  
 (sqrt(sum(receiver\_use\_aggregated\_by\_species\_wide[receiver\_use\_aggregated\_by\_species\_wide$species == species\_combos$species\_1[i], -1] ^ 2) \*   
 sum(receiver\_use\_aggregated\_by\_species\_wide[receiver\_use\_aggregated\_by\_species\_wide$species == species\_combos$species\_2[i], -1] ^ 2)))  
}  
  
## Round to 3 digits  
species\_combos$pianka\_index = round(species\_combos$pianka\_index, 3)  
  
print(species\_combos)

## species\_1 species\_2 pianka\_index  
## 1 grey reef shark omilu 0.578  
## 2 grey reef shark sandbar shark 0.873  
## 3 grey reef shark ulua 0.654  
## 4 grey reef shark whitetip reef shark 0.877  
## 5 omilu sandbar shark 0.154  
## 6 omilu ulua 0.274  
## 7 omilu whitetip reef shark 0.788  
## 8 sandbar shark ulua 0.760  
## 9 sandbar shark whitetip reef shark 0.674  
## 10 ulua whitetip reef shark 0.730

## Plots

Study Area

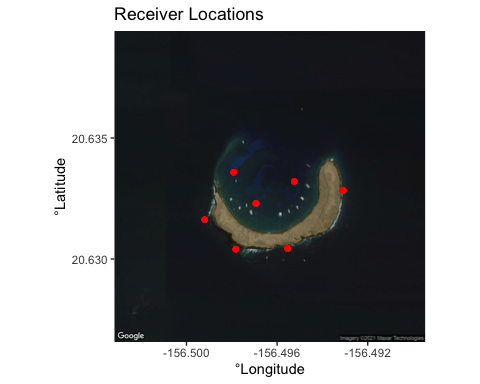
## Plot study area and receivers  
molo\_basemap = get\_map(location = c(lon = -156.496331, lat = 20.633007), zoom = 16, maptype = 'satellite')

## Source : https://maps.googleapis.com/maps/api/staticmap?center=20.633007,-156.496331&zoom=16&size=640x640&scale=2&maptype=satellite&language=en-EN&key=xxx-hggZe5I57UhGHb8

receiver\_map = ggmap(molo\_basemap) + geom\_point(data = molo\_df, mapping = aes(x = lon, y = lat), col = 'red') + labs(x = '°Longitude', y = '°Latitude', title = 'Receiver Locations') + ggsave(filename = 'Receiver Locations Google Map.pdf', path = figure\_directory)

## Saving 5 x 4 in image

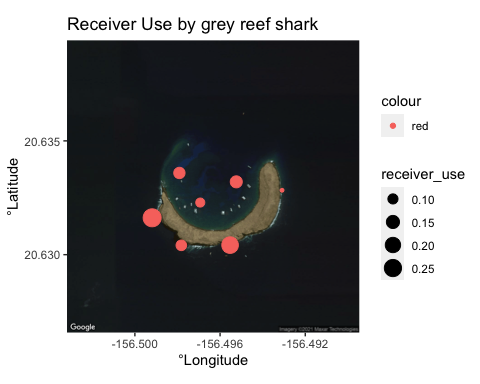
print(receiver\_map)



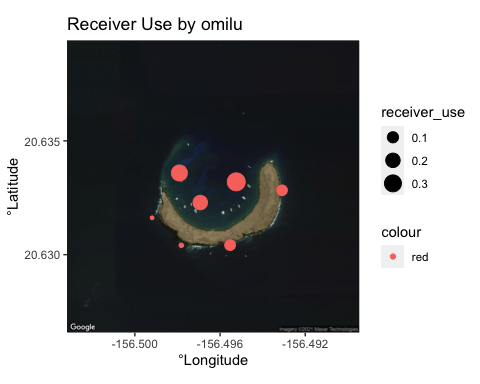
### Species Use Plots

## Get average use of receiver by species   
species\_receiver\_use = aggregate(receiver\_use~species+receiver, data = detections\_per\_tag\_per\_receiver, FUN = mean)  
 colnames(species\_receiver\_use) = c('species', 'receiver' , 'receiver\_use')  
   
## Merge with lat lon positions for each receiver from molo\_df  
receiver\_postions = unique(molo\_df[ ,c('receiver', 'lat', 'lon')])  
species\_receiver\_use = merge(x = species\_receiver\_use, y = receiver\_postions, on = 'receiver', all.x = T, all.y = F)  
  
## Make species plots for receiver use by species  
for(species in unique(species\_receiver\_use$species)){  
 receiver\_use\_by\_spp = ggmap(molo\_basemap) +   
 geom\_point(data = species\_receiver\_use[species\_receiver\_use$species == species, ],   
 mapping = aes(x = lon, y = lat, color = 'red', size = receiver\_use)) +   
 labs(x = '°Longitude', y = '°Latitude', title = paste('Receiver Use by ', species, sep = '')) +  
 ggsave(filename = paste('Receiver Use by ', species, '.pdf', sep = ''), path = figure\_directory)  
 print(receiver\_use\_by\_spp)  
}

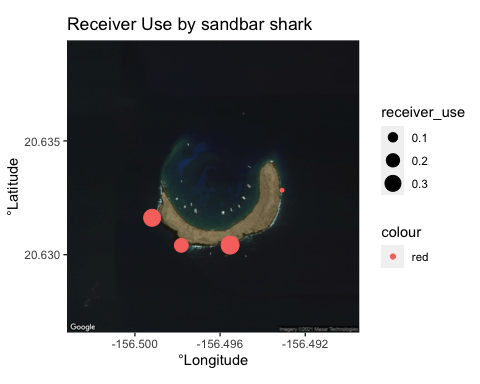
## Saving 5 x 4 in image  
## Saving 5 x 4 in image



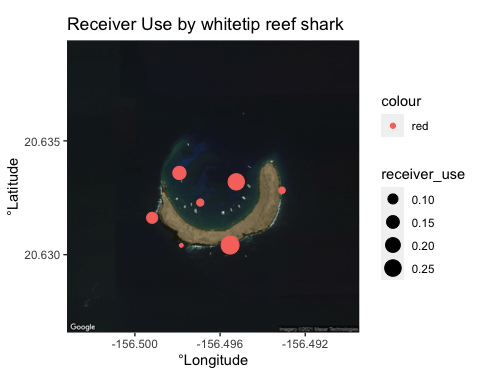
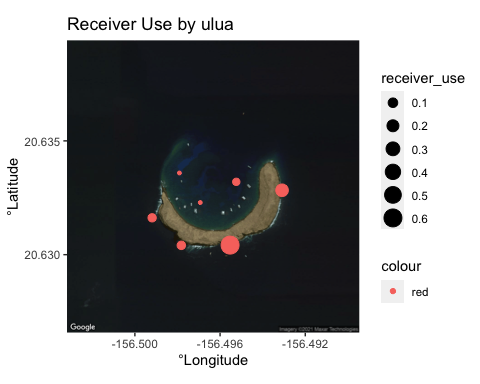
## Saving 5 x 4 in image



## Saving 5 x 4 in image



## Saving 5 x 4 in image



### Day Night Plots

### Day Night Plots  
## For all fish  
pdf(file = file.path(figure\_directory, 'Day Night Plot - All Fish.pdf'))  
plot\_day\_night(molo\_df, plot\_title = 'All Fish')  
dev.off()

## quartz\_off\_screen   
## 2

## By Species  
for (spp in unique(molo\_df$species)){  
 pdf(file = file.path(figure\_directory, paste('Day Night Plot - Species ', spp, '.pdf', sep = '')))  
 plot\_day\_night(molo\_df[molo\_df$tag\_id == molo\_df$tag\_id[molo\_df$species == spp], ], plot\_title = spp)  
 dev.off()  
}

## Warning in molo\_df$tag\_id == molo\_df$tag\_id[molo\_df$species == spp]: longer  
## object length is not a multiple of shorter object length  
  
## Warning in molo\_df$tag\_id == molo\_df$tag\_id[molo\_df$species == spp]: longer  
## object length is not a multiple of shorter object length  
  
## Warning in molo\_df$tag\_id == molo\_df$tag\_id[molo\_df$species == spp]: longer  
## object length is not a multiple of shorter object length  
  
## Warning in molo\_df$tag\_id == molo\_df$tag\_id[molo\_df$species == spp]: longer  
## object length is not a multiple of shorter object length  
  
## Warning in molo\_df$tag\_id == molo\_df$tag\_id[molo\_df$species == spp]: longer  
## object length is not a multiple of shorter object length

## By Individual  
for (tag\_id in unique(molo\_df$tag\_id)){  
 pdf(file = file.path(figure\_directory, paste('Day Night Plot - Tag ID ', tag\_id, '.pdf', sep = '')))  
 plot\_day\_night(molo\_df[molo\_df$tag\_id == tag\_id, ], plot\_title = paste(tagging\_df$species[tagging\_df$tag\_id == tag\_id], '- Tag', as.character(tag\_id), sep = ' '))  
 dev.off()  
}

### Barplot of detections by date

### Bar plot of detections in crater by date   
detections\_per\_day\_df = count\_detections\_per\_date(molo\_df)  
  
## Barplot of detections by individual  
for(i in 1:nrow(detections\_per\_day\_df)){  
 ## Convert from wide to long format  
 indv\_data = melt(detections\_per\_day\_df[i, ])  
 colnames(indv\_data) = c('date', 'detections')  
   
 ## Make and save plot  
 ggplot(data = indv\_data, mapping = aes(x = date, y = detections)) +  
 geom\_bar(stat = "identity") +   
 labs(title = paste('Tag ', rownames(detections\_per\_day\_df)[i], sep = ' '), x = 'Date', y = 'Detections') +   
 ggsave(filename = paste('Daily Detection Barplot -', rownames(detections\_per\_day\_df)[i], '.pdf'), path = figure\_directory)  
}

## No id variables; using all as measure variables

## Saving 5 x 4 in image

## No id variables; using all as measure variables

## Saving 5 x 4 in image

## No id variables; using all as measure variables

## Saving 5 x 4 in image

## No id variables; using all as measure variables

## Saving 5 x 4 in image

## No id variables; using all as measure variables

## Saving 5 x 4 in image

## No id variables; using all as measure variables

## Saving 5 x 4 in image

## No id variables; using all as measure variables

## Saving 5 x 4 in image

## No id variables; using all as measure variables

## Saving 5 x 4 in image

## No id variables; using all as measure variables

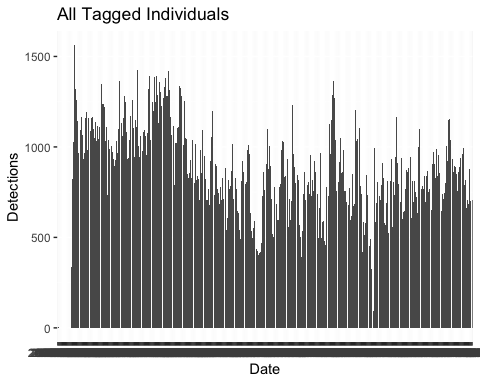
## Saving 5 x 4 in image

## Detections by species  
detections\_per\_day\_spp\_stg = detections\_per\_day\_df  
detections\_per\_day\_spp\_stg$tag\_id = rownames(detections\_per\_day\_spp\_stg)  
detections\_per\_day\_spp\_stg = left\_join(x = detections\_per\_day\_spp\_stg, tagging\_df[ ,c('tag\_id', 'species')], by = 'tag\_id')  
  
## Loop through species  
for (spp in unique(detections\_per\_day\_spp\_stg$species)){  
   
 ## Subset individual df by species  
 spp\_subset\_df = detections\_per\_day\_spp\_stg[detections\_per\_day\_spp\_stg$species == spp, -which(colnames(detections\_per\_day\_spp\_stg) %in% c('tag\_id', 'species'))]  
   
 ## Convert to long format  
 detections\_per\_spp = melt(colSums(spp\_subset\_df), value.name = 'detections')  
 detections\_per\_spp$date = rownames(detections\_per\_spp)  
   
 ## Make and save plot  
 ggplot(data = detections\_per\_spp, mapping = aes(x = date, y = detections)) +  
 geom\_bar(stat = "identity") +   
 labs(title = spp, x = 'Date', y = 'Detections') +   
 ggsave(filename = paste('Daily Detection Barplot -', spp, '.pdf'), path = figure\_directory)  
}

## Saving 5 x 4 in image  
## Saving 5 x 4 in image  
## Saving 5 x 4 in image  
## Saving 5 x 4 in image  
## Saving 5 x 4 in image

## Barplot of all detections  
all\_detections = colSums(detections\_per\_day\_df)  
## Convert to long format  
all\_detections\_long = melt(all\_detections, value.name = 'detections')  
all\_detections\_long$date = rownames(all\_detections\_long)  
   
## Make and save plot  
ggplot(data = all\_detections\_long, mapping = aes(x = date, y = detections)) +  
geom\_bar(stat = "identity") +   
labs(title = 'All Tagged Individuals', x = 'Date', y = 'Detections') +   
ggsave(filename = paste('Daily Detection Barplot - all tags.pdf'), path = figure\_directory)

## Saving 5 x 4 in image



### Bar plot # of Fish (standardized percent of fish tagged to date) by date and Spp

THIS NEEDS WORK!!!

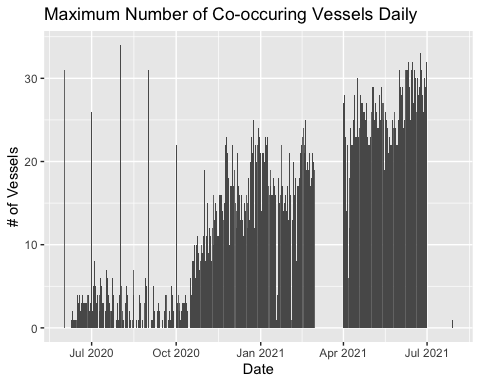
### Bar plot vessel traffic by date

In the future, might also consider max vessels present at a given time

## Calculate Daily Vessel Stats  
vessels\_per\_day = aggregate(vessel\_name ~ date, data = vessel\_df, FUN = uniqueN)  
colnames(vessels\_per\_day) = c('date', 'daily\_vessels')  
  
# Make plot for total   
total\_vessels\_plot = ggplot(data = vessels\_per\_day, mapping = aes(x = date, y = daily\_vessels)) +   
 geom\_bar(stat = 'identity') +   
 labs(title = 'Maximum Number of Co-occuring Vessels Daily', x = 'Date', y = '# of Vessels') +  
 ggsave(filename = paste('Total Vessels Daily.pdf ', species, '.pdf', sep = ''), path = figure\_directory)

## Saving 5 x 4 in image

print(total\_vessels\_plot)

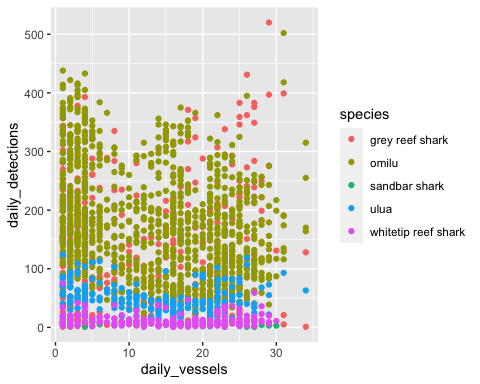


### Scatter plot x axis boat traffic, y axis detections per individual color by spp add error bars for daily detections

## combine vessel and detections per individual  
detections\_per\_day\_per\_tag = aggregate(datetime ~ date + tag\_id + species, data = molo\_df, FUN = uniqueN)  
colnames(detections\_per\_day\_per\_tag) = c('date', 'tag\_id', 'species', 'daily\_detections')  
detection\_vessel\_counts = left\_join(detections\_per\_day\_per\_tag, vessels\_per\_day, by = 'date')  
  
ggplot(data = detection\_vessel\_counts, mapping = aes(x = daily\_vessels, y = daily\_detections, color = species)) + geom\_point() + ggsave(filename = file.path(figure\_directory, 'detections vs vessels.pdf'))

## Saving 5 x 4 in image

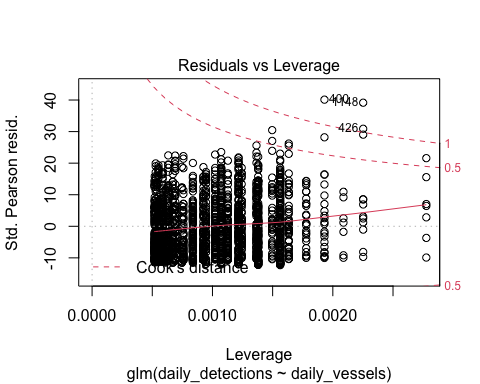
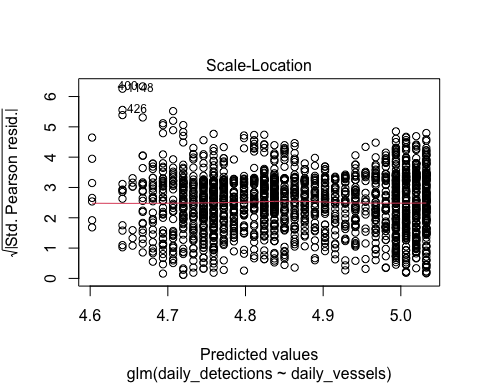
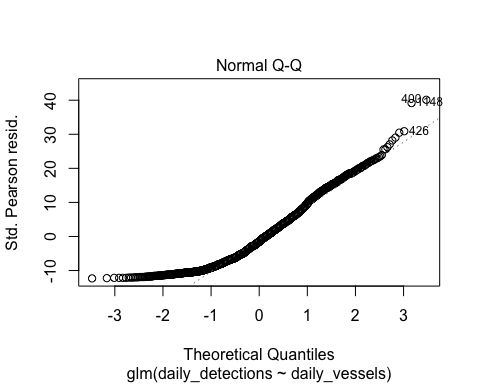
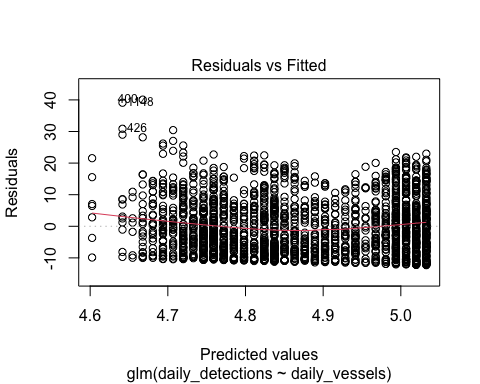
## Warning: Removed 475 rows containing missing values (geom\_point).  
  
## Warning: Removed 475 rows containing missing values (geom\_point).

 # Model the relationship between detections and vessels (this is ts data so this is just quick and dirty)

vessel\_detection\_glm = glm(daily\_detections ~ daily\_vessels, data = detection\_vessel\_counts, family = poisson)  
print(summary(vessel\_detection\_glm))

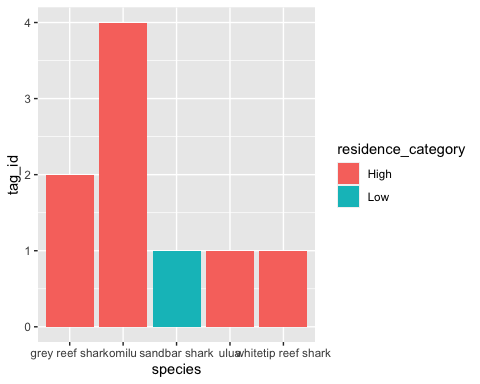
##   
## Call:  
## glm(formula = daily\_detections ~ daily\_vessels, family = poisson,   
## data = detection\_vessel\_counts)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -17.166 -8.347 -1.554 5.236 28.679   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) 5.0458586 0.0033706 1497.03 <2e-16 \*\*\*  
## daily\_vessels -0.0130406 0.0002241 -58.18 <2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for poisson family taken to be 1)  
##   
## Null deviance: 158720 on 1939 degrees of freedom  
## Residual deviance: 155305 on 1938 degrees of freedom  
## (475 observations deleted due to missingness)  
## AIC: 167410  
##   
## Number of Fisher Scoring iterations: 5

plot(vessel\_detection\_glm)

 Conclusion: There does appear to be a negative trend where increased vessel traffic results in fewer fish. See coefficients. Need to dig deeper if trying to publish these results though, model should be better

# Residency and dispersal

## Calculate residency  
detection\_stats$residence\_metric = detection\_stats$unique\_days / detection\_stats$days\_at\_liberty  
  
## Assign residence category: low = < 33%, medium = 33 - 66, high = >= 66 (Tinhan et al. 2014) -  
detection\_stats$residence\_category = 'Low'  
for (i in 1:nrow(detection\_stats)){  
 if (detection\_stats$residence\_metric[i] >= (1/3)) {  
 detection\_stats$residence\_category[i] = 'Medium'  
 }  
 if (detection\_stats$residence\_metric[i] >= (2/3)) {  
 detection\_stats$residence\_category[i] = 'High'  
 }  
}  
  
## Create grouped barplot of residency by species  
residence\_counts\_by\_species = aggregate(tag\_id ~ species + residence\_category, data = detection\_stats, FUN = length)  
  
ggplot(data = residence\_counts\_by\_species, mapping = aes(x=species, y=tag\_id, fill=residence\_category)) +  
 geom\_bar(stat="identity", position = "dodge")



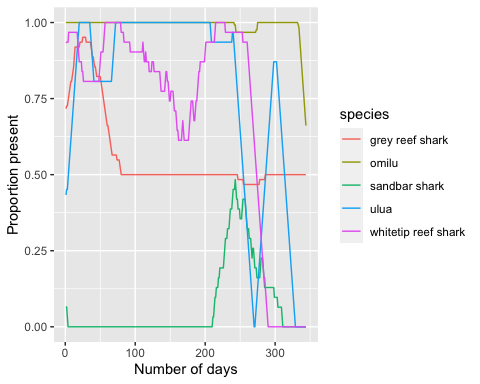
Takeaways - All 4 omilus were highly resident as were grey reef sharks. No other species have replicates so…?

## Calculate 30 day moving average of residency, then plot against days since tagging

## Get total days in the study  
total\_days\_in\_study = as.numeric(diff.Date(c(min(molo\_df$date), max(molo\_df$date))))  
  
## Create a dataframe where rows are tag id and columns are study date  
present\_after\_n\_days\_df = data.frame()  
  
## Determine if a tag was detected on a receiver n days after tagging  
for (i in 1:uniqueN(molo\_df$tag\_id)){  
 ## Subset data for individual tags  
 indv\_data = molo\_df[molo\_df$tag\_id == unique(molo\_df$tag\_id)[i], ]  
 ## Determine if a fish was present n days after tagging  
 difftimes = rep(0, len = total\_days\_in\_study)  
 # determine difference in days between each unique day a tag was detected and the tag's earliest detection, flip the corresponding value in difftimes array to 1  
 detected\_dates = unique(indv\_data$date)  
 for (j in 1:length(detected\_dates)){  
 difftimes[as.numeric(diff.Date(c(min(indv\_data$date), detected\_dates[j]))) + 1] = 1  
 }  
 df\_row = c(unique(molo\_df$tag\_id)[i], difftimes)  
 present\_after\_n\_days\_df = rbind(present\_after\_n\_days\_df, df\_row)  
}  
colnames(present\_after\_n\_days\_df) = c('tag\_id', as.character(1:total\_days\_in\_study))  
  
## Convert from wide format to long format  
present\_after\_n\_days\_df\_long\_df = melt(present\_after\_n\_days\_df, id.vars = 'tag\_id', measure.vars = colnames(present\_after\_n\_days\_df)[2:ncol(present\_after\_n\_days\_df)], variable.name = 'day', value.name = 'detected')  
  
## Merge with species data  
present\_after\_n\_days\_df\_long\_df = left\_join(x = present\_after\_n\_days\_df\_long\_df, y = tagging\_df[ ,c('tag\_id', 'species')], by = 'tag\_id')  
# Recast to numeric because of the join function  
present\_after\_n\_days\_df\_long\_df$detected = as.numeric(present\_after\_n\_days\_df\_long\_df$detected)  
  
## Calculate number of each species present n days after tagging  
species\_presence\_after\_tagging = aggregate(detected ~ species + day, data = present\_after\_n\_days\_df\_long\_df, FUN = sum)  
 colnames(species\_presence\_after\_tagging) = c('species', 'day', 'n\_detected')  
  
## Count unique tags by species  
individuals\_per\_species = aggregate(tag\_id ~ species, data = present\_after\_n\_days\_df\_long\_df, FUN = uniqueN)  
colnames(individuals\_per\_species) = c('species', 'n\_tagged')  
  
## Standardize species level daily counts by number of tags belonging to that species  
species\_presence\_after\_tagging = left\_join(x = species\_presence\_after\_tagging, y = individuals\_per\_species, by = 'species')  
species\_presence\_after\_tagging$percent\_individuals\_detected = species\_presence\_after\_tagging$n\_detected / species\_presence\_after\_tagging$n\_tagged  
  
## Convert day from factor to numeric  
species\_presence\_after\_tagging$day = as.numeric(levels(species\_presence\_after\_tagging$day)[species\_presence\_after\_tagging$day])  
  
## remove any NA days  
species\_presence\_after\_tagging = species\_presence\_after\_tagging[!is.na(species\_presence\_after\_tagging$day), ]  
  
## Calculate 30 day moving average  
spp\_presence\_30\_day\_avg = data.frame()  
for (species in unique(species\_presence\_after\_tagging$species)){  
 spp\_presence\_after\_tagging = species\_presence\_after\_tagging[species\_presence\_after\_tagging$species == species, ]  
 moving\_average\_30 = c()  
 for (i in 30:max(spp\_presence\_after\_tagging$day)){  
 moving\_average\_30 = c(moving\_average\_30, mean(spp\_presence\_after\_tagging$percent\_individuals\_detected[spp\_presence\_after\_tagging$day >= i-30 & spp\_presence\_after\_tagging$day <= i]))  
 }  
 df\_row = c(species, moving\_average\_30)  
 spp\_presence\_30\_day\_avg = rbind(spp\_presence\_30\_day\_avg, df\_row)  
}  
colnames(spp\_presence\_30\_day\_avg) = c('species', as.character(1:(ncol(spp\_presence\_30\_day\_avg)-1)))  
  
## Convert from wide format to long format  
spp\_presence\_30\_day\_avg\_long\_df = melt(spp\_presence\_30\_day\_avg, id.vars = 'species', measure.vars = colnames(spp\_presence\_30\_day\_avg)[2:ncol(spp\_presence\_30\_day\_avg)], variable.name = 'day', value.name = 'percent\_individuals\_detected')  
  
# Convert percent\_individuals\_detected and date  
spp\_presence\_30\_day\_avg\_long\_df$percent\_individuals\_detected = as.numeric(spp\_presence\_30\_day\_avg\_long\_df$percent\_individuals\_detected)  
  
spp\_presence\_30\_day\_avg\_long\_df$day = as.numeric(levels(spp\_presence\_30\_day\_avg\_long\_df$day)[spp\_presence\_30\_day\_avg\_long\_df$day])  
  
  
## Generate line plot  
present\_after\_tagging\_plot = ggplot(spp\_presence\_30\_day\_avg\_long\_df, mapping = aes(x = day, y = percent\_individuals\_detected, color = species)) +   
 geom\_line() +   
 labs(x = 'Number of days', y = 'Proportion present') +  
 ggsave(filename = 'Proportion of tags present after tagging.pdf', path = figure\_directory)

## Saving 5 x 4 in image

print(present\_after\_tagging\_plot)



# Statistical analysis

Calculate mean residency by spp (irregardless of time), then ANOVA by spp Use Tukey’s HSD to determine significance

## ANOVA model for residency metric by species  
residence\_by\_species\_anova = aov(residence\_metric ~ species, data=detection\_stats)  
summary(residence\_by\_species\_anova)

## Df Sum Sq Mean Sq F value Pr(>F)   
## species 4 0.6781 0.16954 36.28 0.00212 \*\*  
## Residuals 4 0.0187 0.00467   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

## Tukey's Honestly Significant Differences between species  
TukeyHSD(residence\_by\_species\_anova)

## Tukey multiple comparisons of means  
## 95% family-wise confidence level  
##   
## Fit: aov(formula = residence\_metric ~ species, data = detection\_stats)  
##   
## $species  
## diff lwr upr p adj  
## omilu-grey reef shark 0.13134256 -0.1318229 0.3945080 0.3282773  
## sandbar shark-grey reef shark -0.78499266 -1.1571648 -0.4128205 0.0034142  
## ulua-grey reef shark -0.04021073 -0.4123829 0.3319614 0.9852738  
## whitetip reef shark-grey reef shark 0.01604564 -0.3561265 0.3882178 0.9995567  
## sandbar shark-omilu -0.91633521 -1.2560803 -0.5765901 0.0013197  
## ulua-omilu -0.17155329 -0.5112984 0.1681918 0.3205721  
## whitetip reef shark-omilu -0.11529692 -0.4550421 0.2244482 0.6064486  
## ulua-sandbar shark 0.74478192 0.3150345 1.1745293 0.0071653  
## whitetip reef shark-sandbar shark 0.80103830 0.3712909 1.2307857 0.0054544  
## whitetip reef shark-ulua 0.05625637 -0.3734910 0.4860037 0.9710009

GLM comparing residency time by spp independent var (time at liberty) dependent (residency index)

## Fit binomial GLM to average residency metric data (proportional between 0-1)  
species\_glm = glm(residence\_metric ~ species \* days\_at\_liberty, data = detection\_stats, family = binomial(logit))

## Warning in eval(family$initialize): non-integer #successes in a binomial glm!

summary(species\_glm)

##   
## Call:  
## glm(formula = residence\_metric ~ species \* days\_at\_liberty, family = binomial(logit),   
## data = detection\_stats)  
##   
## Deviance Residuals:   
## 1 2 3 4 5 6   
## 9.551e-09 9.551e-09 0.000e+00 -5.845e-09 0.000e+00 0.000e+00   
## 7 8 9   
## 9.551e-09 0.000e+00 0.000e+00   
##   
## Coefficients: (3 not defined because of singularities)  
## Estimate Std. Error z value  
## (Intercept) 6.656e-01 3.305e+00 0.201  
## speciesomilu 4.221e+00 3.984e+03 0.001  
## speciessandbar shark -5.228e+00 5.534e+00 -0.945  
## speciesulua -1.367e+00 5.170e+00 -0.264  
## specieswhitetip reef shark -6.425e-01 4.904e+00 -0.131  
## days\_at\_liberty 6.902e-03 1.952e-02 0.354  
## speciesomilu:days\_at\_liberty -4.132e-03 1.099e+01 0.000  
## speciessandbar shark:days\_at\_liberty NA NA NA  
## speciesulua:days\_at\_liberty NA NA NA  
## specieswhitetip reef shark:days\_at\_liberty NA NA NA  
## Pr(>|z|)  
## (Intercept) 0.840  
## speciesomilu 0.999  
## speciessandbar shark 0.345  
## speciesulua 0.791  
## specieswhitetip reef shark 0.896  
## days\_at\_liberty 0.724  
## speciesomilu:days\_at\_liberty 1.000  
## speciessandbar shark:days\_at\_liberty NA  
## speciesulua:days\_at\_liberty NA  
## specieswhitetip reef shark:days\_at\_liberty NA  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 4.3149e+00 on 8 degrees of freedom  
## Residual deviance: 1.5038e-16 on 2 degrees of freedom  
## AIC: 15.426  
##   
## Number of Fisher Scoring iterations: 8

No Significant differences found

## GLM comparing time in crater to vessel traffic - Still to come