Occupancy Modeling

Amanda Zellmer

2024-09-13

## Install and open autoOcc package

# install.packages("devtools")  
# devtools::install\_github(  
# "mfidino/autoOcc",  
# build\_vignettes = TRUE  
# )  
  
library(autoOcc)  
#browseVignettes("autoOcc")

## Open other required packages

library(ggplot2)  
library(dplyr)  
library(tidyr)  
library(sf)  
library(terra)

## Open the dataset

# Apply input species name  
species <- params$sp\_name  
# Apply input folder pathway  
filepath <- params$path  
# Apply input data range (all or middle)  
data\_range <- params$range  
# Apply input data type (day or week)  
type <- params$by  
  
# Specifies output filepath, "...Output/<Middle or All> Weeks/By <Day or Week>/Results - <species>/"  
# Change format as intended  
output\_path <- paste0(filepath, "Output/", data\_range, " Weeks/By ", type, "/Results - ", species, "/")  
  
# Read in current file  
glaa <- read.csv(paste0(filepath, "Species Occupancy/OccupancyReport - ", species, " - Seasonal.csv"))

## Summarize the occupancy data by week

* This is particularly important if your species has low detection rates

#### CHANGES  
non\_na\_count <- colSums(!is.na(glaa) & grepl("^Day\_", names(glaa)))  
  
#glaa <- glaa %>%  
# select(-matches("^Day\_(1[0-4]|[1-9])$"), # To drop columns from Day\_1 to Day\_14  
# -matches("^Day\_(5[0-9]|6[0-9]|60)$")) # To drop columns from Day\_48 to Day\_60  
  
  
## Convert Day data into Week Data  
## Convert Day data into Week Data  
glaa <- glaa %>%  
 rowwise() %>%  
 mutate(  
 Week\_1 = ifelse(all(is.na(c\_across(matches("^Day\_[1-7]$")))), NA,  
 ifelse(sum(c\_across(matches("^Day\_[1-7]$")), na.rm = TRUE) > 0, 1, 0)),  
 Week\_2 = ifelse(all(is.na(c\_across(matches("^Day\_(8|9|1[0-4])$")))), NA,  
 ifelse(sum(c\_across(matches("^Day\_(8|9|1[0-4])$")), na.rm = TRUE) > 0, 1, 0)),  
 Week\_3 = ifelse(all(is.na(c\_across(matches("^Day\_(1[5-9]|2[0-1])$")))), NA,  
 ifelse(sum(c\_across(matches("^Day\_(1[5-9]|2[0-1])$")), na.rm = TRUE) > 0, 1, 0)),  
 Week\_4 = ifelse(all(is.na(c\_across(matches("^Day\_(2[2-8])$")))), NA,  
 ifelse(sum(c\_across(matches("^Day\_(2[2-8])$")), na.rm = TRUE) > 0, 1, 0)),  
 Week\_5 = ifelse(all(is.na(c\_across(matches("^Day\_(2[9-9]|3[0-5])$")))), NA,  
 ifelse(sum(c\_across(matches("^Day\_(2[9-9]|3[0-5])$")), na.rm = TRUE) > 0, 1, 0)),  
 Week\_6 = ifelse(all(is.na(c\_across(matches("^Day\_(3[6-9]|4[0-2])$")))), NA,  
 ifelse(sum(c\_across(matches("^Day\_(3[6-9]|4[0-2])$")), na.rm = TRUE) > 0, 1, 0)),  
 Week\_7 = ifelse(all(is.na(c\_across(matches("^Day\_(4[3-9])$")))), NA,  
 ifelse(sum(c\_across(matches("^Day\_(4[3-9])$")), na.rm = TRUE) > 0, 1, 0)),  
 Week\_8 = ifelse(all(is.na(c\_across(matches("^Day\_(5[0-6])$")))), NA,  
 ifelse(sum(c\_across(matches("^Day\_(5[0-6])$")), na.rm = TRUE) > 0, 1, 0))  
 ) %>%  
 ungroup()  
  
# Remove outside days and weeks if only considering middle  
if (data\_range == "Middle") {  
 skip\_days <- c(paste0("Day\_", 1:14), paste0("Day\_", 43:60))  
 skip\_weeks <- c("Week\_1", "Week\_2", "Week\_7", "Week\_8")  
 glaa <- glaa %>%  
 select(-all\_of(skip\_days), -all\_of(skip\_weeks))  
}

## Add Season Data

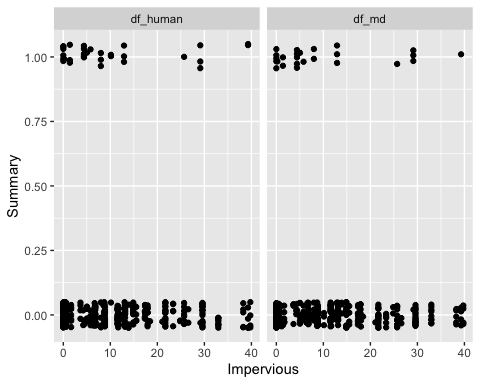
glaaSeasons <- read.csv(paste0(filepath, "Species Occupancy/header.csv"))  
glaaSeasons <- subset(glaaSeasons, Season.Legend. < 23)  
glaaSeasons <- glaaSeasons[,1:3]  
glaaSeasons$Month <- c(rep(c("OC", "JA", "AP", "JU"), times=5), "OC", "JA")  
glaaSeasons$Year <- c("18", rep(c("19", "20", "21", "22", "23"), each = 4), "24")  
glaaSeasons$Season <- paste0(glaaSeasons$Month, glaaSeasons$Year)  
glaaSeasons$SeasonNumber <- 1:nrow(glaaSeasons)  
glaaSeasons$Precipitation <- c("below avg", rep("above avg", times=8), rep("below avg", times=8), rep("above avg", times=5))  
  
glaa <- merge(glaa, glaaSeasons, by = "SeasonNumber", all=T)

## Open Impervious Data

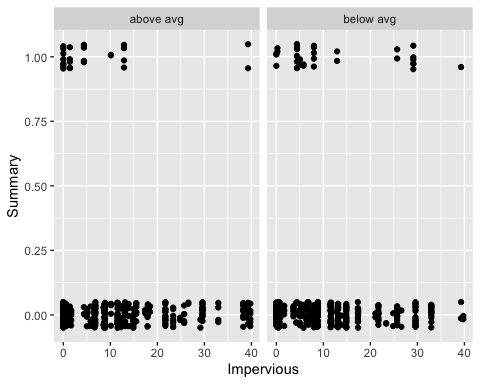
impervious <- rast(paste0(filepath, "impervious\_crop.asc"))  
 impervious[impervious >100] <- NA  
 e <- ext(c(min(glaa$Longitude), max(glaa$Longitude), min(glaa$Latitude), max(glaa$Latitude)))  
 e <- e+0.1  
 impervious <- crop(impervious, e)  
 names(impervious) <- "Layer\_1"  
 #plot(impervious)  
 #points(glaaSP)  
  
  
  
impervious\_df <- as.data.frame(impervious, xy = TRUE)  
  
  
  
## Create buffer around points  
library(sf)  
points\_sf <- st\_as\_sf(glaa[,c("Longitude", "Latitude")], coords = c("Longitude", "Latitude"))  
  
# Create a 400m buffer around each point  
# Conversion factor for latitude 34  
meters\_to\_degrees <- 1 / 110567  
  
# Convert 400 meters to decimal degrees at latitude 34  
distance\_in\_degrees <- 200 \* meters\_to\_degrees  
buffer\_distance <- distance\_in\_degrees # in meters  
#initially tried 0.01 but that is 1106m  
points\_buffer <- st\_buffer(points\_sf, dist = buffer\_distance)  
  
extract.data <- terra::extract(impervious[[1]], data.frame(glaa$Longitude, glaa$Latitude)) #impervious  
glaa$Impervious <- extract.data[,2]  
  
extract.data <- terra::extract(impervious[[1]], points\_buffer) #impervious  
extract.data <- extract.data %>%  
 group\_by(ID) %>%  
 summarise(Impervious = mean(as.numeric(Layer\_1), na.rm = TRUE))  
extract.data2 <- as.data.frame(extract.data)  
glaa$Impervious <- extract.data2[,2]  
  
# Check if directory exists for results of current species; if not, create one  
if(dir.exists(output\_path)==F){dir.create(output\_path)}  
write.csv(glaa, paste0(output\_path, species, "\_final.csv"), row.names=F, quote=F)  
  
  
dev.new()  
jpeg("impervious.jpg")  
plot(impervious)  
points(glaa$Longitude, glaa$Latitude)  
dev.off()

## quartz\_off\_screen   
## 2

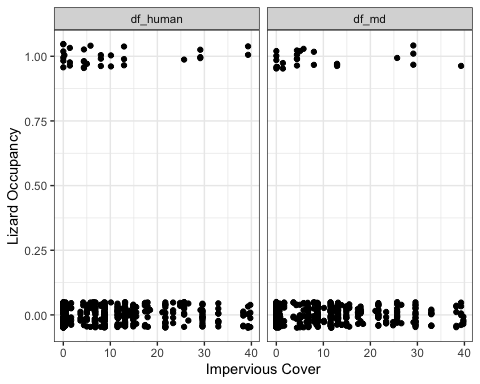
# Order the seasons by date  
seasonOrder <- c("OC18",  
 "JA19", "AP19", "JU19", "OC19",  
 "JA20", "AP20", "JU20", "OC20",  
 "JA21", "AP21", "JU21", "OC21",  
 "JA22", "AP22", "JU22", "OC22",  
 "JA23", "AP23", "JU23", "OC23",  
 "JA24")  
  
## Convert Season into a Factor  
glaa$Season <- factor(glaa$Season, levels=seasonOrder)  
  
## Arrange the data by Season and then Site  
glaa <- arrange(glaa, Season, Site)  
  
## Classify the Seasons by Dry vs Wet Years  
seasonsDro <- c("OC18",   
 "JA21", "AP21", "JU21", "OC21",  
 "JA22", "AP22", "JU22", "OC22")  
   
seasonsWet <- c("JA19", "AP19", "JU19", "OC19",  
 "JA20", "AP20", "JU20", "OC20",  
 "JA23", "AP23", "JU23", "OC23",  
 "JA24")   
   
glaa$Precipitation <- "below avg"  
glaa$Precipitation[glaa$Season %in% seasonsWet] <- "above avg"  
  
## Create a list of unique species  
speciesList <- unique(glaa$Species)  
  
glaa.original <- glaa  
  
####  
# Summarize rows across columns and add a summary column  
glaa2 <- glaa %>%  
 # Drop rows with all NA  
 filter(!rowSums(is.na(select(., starts\_with("Week\_")))) == ncol(select(., starts\_with("Week\_")))) %>%  
 # Create a new column named 'Summary' with 1 if any Week\_ column contains 1, else 0  
 mutate(Summary = ifelse(rowSums(select(., starts\_with("Week\_")) == 1, na.rm = TRUE) > 0, 1, 0))  
  
ggplot()+  
 geom\_jitter(data=glaa2, aes(x=Impervious, y=Summary), height=0.05, width=0)+  
 facet\_wrap(.~Treatment)



# Summarize rows across columns and add a summary column  
glaa3 <- glaa.original %>%  
 # Drop rows with all NA  
 filter(!rowSums(is.na(select(., starts\_with("Day\_")))) == ncol(select(., starts\_with("Day\_")))) %>%  
 # Create a new column named 'Summary' with 1 if any Day\_ column contains 1, else 0  
 mutate(Summary = ifelse(rowSums(select(., starts\_with("Day\_")) == 1, na.rm = TRUE) > 0, 1, 0))  
  
ggplot()+  
 geom\_jitter(data=glaa3, aes(x=Impervious, y=Summary), height=0.05, width=0)+  
 facet\_wrap(.~Precipitation)



ggplot()+  
 geom\_jitter(data=glaa3, aes(x=Impervious, y=Summary), height=0.05, width=0)+  
 facet\_wrap(.~Treatment) +  
 theme\_bw()+  
 xlab("Impervious Cover")+  
 ylab(paste0(species, " Occupancy"))



day\_sums <- glaa.original %>%  
 # Select columns that start with "Day\_"  
 select(starts\_with("Day\_")) %>%  
 # Calculate the sum for each column  
 summarise(across(everything(), ~ sum(., na.rm = TRUE)))

## Run the Occupancy Model

spp <- species  
species\_det\_hist <- glaa %>%  
 dplyr::select(c(Species, Site, Season, Month, starts\_with(paste0(type, "\_"))))  
species\_det\_hist <- as.data.frame(species\_det\_hist)  
species\_covariates <- unique(glaa[,c("Site", "Impervious", "Treatment")])  
  
table(species\_det\_hist$Species)

##   
## Lizard   
## 3408

# function to generate detection history  
# for the opossum data, opossum\_y is  
# a site by primary sampling period by secondary   
# sampling period array.  
species\_y <- autoOcc::format\_y(  
 x = species\_det\_hist,  
 site\_column = "Site",  
 time\_column = "Season",  
 history\_columns = paste0("^", type)  
)

##   
##   
## TEMPORAL ORDERING  
## -----------------  
##   
## Primary sampling period column is a factor, factor levels to order temporally.  
## Ordering: OC18, JA19, AP19, JU19, OC19, JA20, AP20, JU20, OC20, JA21, AP21, JU21, OC21, JA22, AP22, JU22, OC22, JA23, AP23, JU23, OC23, JA24  
##   
## DETECTION HISTORIES  
## -------------------  
##   
## 4 detection history columns found.  
## Column names: Week\_3, Week\_4, Week\_5, Week\_6

# scale the covariates for analysis  
oc\_scaled <- as.data.frame(  
 lapply(  
 species\_covariates,  
 function(x){  
 if(is.numeric(x)){  
 scale(x)  
 }else{  
 x  
 }  
 }  
 )  
)  
# dropping site column from oc\_scaled  
oc\_scaled <- oc\_scaled[,-1]  
#oc\_scaled <- data.frame(Impervious = oc\_scaled) #only need if only one variable in oc\_scaled  
  
# season\_frame <- list(  
# Precipitation = matrix(  
# species\_det\_hist$Precipitation,  
# nrow = length(unique(species\_det\_hist$Site)),  
# ncol = length(unique(species\_det\_hist$Season))  
# ),  
# Impervious = oc\_scaled$Impervious  
# )  
#   
# ##turn into a factor rather than character  
# unique\_seasons <- as.vector(  
# unique(  
# species\_det\_hist$Precipitation  
# )  
# )  
#   
# season\_frame$Precipitation <- as.data.frame(season\_frame$Precipitation)  
#   
# # make each column into a factor with the appropriate levels  
# season\_frame$Precipitation <- as.data.frame(  
# lapply(  
# season\_frame$Precipitation,  
# function(x) factor(x, levels = unique\_seasons)  
# )  
# )  
#   
  
  
# This temporal covariate is categorical and varies by primary sampling period,  
# as such, the first column of this matrix is full of "JA19", the second is  
# full of "AP19", etc. etc.  
# head(season\_frame$Precipitation)  
  
m1 <- auto\_occ(  
 ~1~1,  
 y = species\_y,  
 det\_covs = oc\_scaled,  
 occ\_covs = oc\_scaled  
)

## Warning in auto\_occ(~1 ~ 1, y = species\_y, det\_covs = oc\_scaled, occ\_covs = oc\_scaled):   
## Some sites have no data.  
## Removed sites at rows: 19, 20, 75, 76, 77, 78, 103, 104, 131, 132

## Warning:   
## occ\_covs: 'Treatment' column is a character object. Converted it to a factor.

## Warning:   
## det\_covs: 'Treatment' column is a character object. Converted it to a factor.

m2 <- auto\_occ(  
 ~Impervious~Impervious,  
 y = species\_y,  
 det\_covs = oc\_scaled,  
 occ\_covs = oc\_scaled  
)

## Warning in auto\_occ(~Impervious ~ Impervious, y = species\_y, det\_covs = oc\_scaled, :   
## Some sites have no data.  
## Removed sites at rows: 19, 20, 75, 76, 77, 78, 103, 104, 131, 132

## Warning:   
## occ\_covs: 'Treatment' column is a character object. Converted it to a factor.

## Warning:   
## det\_covs: 'Treatment' column is a character object. Converted it to a factor.

m3 <- auto\_occ(  
 ~Treatment~Treatment,  
 y = species\_y,  
 det\_covs = oc\_scaled,  
 occ\_covs = oc\_scaled  
)

## Warning in auto\_occ(~Treatment ~ Treatment, y = species\_y, det\_covs = oc\_scaled, :   
## Some sites have no data.  
## Removed sites at rows: 19, 20, 75, 76, 77, 78, 103, 104, 131, 132

## Warning:   
## occ\_covs: 'Treatment' column is a character object. Converted it to a factor.

## Warning:   
## det\_covs: 'Treatment' column is a character object. Converted it to a factor.

m4 <- auto\_occ(  
 ~Impervious + Treatment ~ Impervious + Treatment,  
 y = species\_y,  
 det\_covs = oc\_scaled,  
 occ\_covs = oc\_scaled  
)

## Warning in auto\_occ(~Impervious + Treatment ~ Impervious + Treatment, y = species\_y, :   
## Some sites have no data.  
## Removed sites at rows: 19, 20, 75, 76, 77, 78, 103, 104, 131, 132

## Warning:   
## occ\_covs: 'Treatment' column is a character object. Converted it to a factor.

## Warning:   
## det\_covs: 'Treatment' column is a character object. Converted it to a factor.

m5 <- auto\_occ(  
 ~Impervious \* Treatment ~ Impervious \* Treatment,  
 y = species\_y,  
 det\_covs = oc\_scaled,  
 occ\_covs = oc\_scaled  
)

## Warning in auto\_occ(~Impervious \* Treatment ~ Impervious \* Treatment, y = species\_y, :   
## Some sites have no data.  
## Removed sites at rows: 19, 20, 75, 76, 77, 78, 103, 104, 131, 132

## Warning:   
## occ\_covs: 'Treatment' column is a character object. Converted it to a factor.

## Warning:   
## det\_covs: 'Treatment' column is a character object. Converted it to a factor.

# compare models  
my\_aic\_results <- compare\_models(  
 list(m1, m2, m3, m4, m5),  
 digits = 2  
)  
  
write.csv(my\_aic\_results, paste0(output\_path, spp, "\_AIC.csv"), row.names=F, quote=F)  
  
cat(capture.output(summary(m1)), file=paste0(output\_path, spp, ".txt"), sep="\n", append=FALSE)  
cat(capture.output(summary(m2)), file=paste0(output\_path, spp, ".txt"), sep="\n", append=TRUE)  
cat(capture.output(summary(m3)), file=paste0(output\_path, spp, ".txt"), sep="\n", append=TRUE)  
cat(capture.output(summary(m4)), file=paste0(output\_path, spp, ".txt"), sep="\n", append=TRUE)  
cat(capture.output(summary(m5)), file=paste0(output\_path, spp, ".txt"), sep="\n", append=TRUE)  
  
  
# first make the prediction data.frame with a realistic  
# range based on the actual data and not the scaled data.  
# The range(oc$Impervious) is about 0 to 81, so choose 0  
# to 80. We do this so that we have nice numbers for plotting.  
# Likewise, we scaled all of the other data, so we leave Income  
# at it's mean (i.e., 0) for predictions.  
imperv\_seq <- seq(0,ceiling(max(glaa2$Impervious)),1)  
season\_levels <- unique(oc\_scaled$Treatment)  
imperv\_real <- data.frame(  
 Impervious = rep(  
 imperv\_seq,  
 length(season\_levels)  
 ),  
 Treatment = factor(  
 rep(season\_levels,  
 each = length(imperv\_seq)  
 ),  
 levels = season\_levels  
 )  
)  
  
  
imperv\_scaled <- imperv\_real  
# Error here when species is "Mountain lion", or "Domestic Horse", or "Western Toad" (determinant is 0? It's a smaller set of data)  
imperv\_scaled$Impervious <- (  
 imperv\_scaled$Impervious - mean(species\_covariates$Impervious)  
) / sd(species\_covariates$Impervious)  
  
for(i in c("m1", "m2", "m3", "m4", "m5")){ # Broke on m2 for "Snake", "sigma must be a symmetric matrix"  
 print(i)  
 set.seed(13)  
 # the model prediction across a gradient of Impervious cover  
 opo\_imperv <- predict(  
 object = get(i), #m3, #get(my\_aic\_results[1,1]),  
 type = "psi",  
 newdata = imperv\_scaled  
 )  
   
 # add on the covariate data  
 opo\_imperv <- data.frame(  
 opo\_imperv,  
 imperv\_real  
 )  
   
   
 # plot it out  
   
 ggplot()+  
 geom\_line(data=opo\_imperv, aes(x=Impervious, y=estimate, color=Treatment))+  
 geom\_ribbon(data=opo\_imperv,   
 aes(x=Impervious, ymin=lower,ymax=upper, fill=Treatment), alpha=0.25)+  
 scale\_color\_manual(values=c("red", "dodgerblue"))+  
 scale\_fill\_manual(values=c("red", "dodgerblue"))+  
 geom\_point(data=glaa2, aes(x=Impervious, y=Summary, shape=Treatment),  
 alpha=0.25)+  
 facet\_grid(.~Treatment)+  
 theme\_bw()+  
 ylab("Occupancy")+  
 xlab("Impervious Cover (%)")+  
 ggtitle(i)+  
 ylim(0,1)  
   
 ggsave(paste0(output\_path, i, ".jpg"), units="in", height=5, width=6, dpi=600)  
 rm(opo\_imperv)  
}

## [1] "m1"

## Warning in data.frame(opo\_imperv, imperv\_real): row names were found from a short variable and have been discarded

## [1] "m2"

## [1] "m3"

## [1] "m4"

## [1] "m5"

#my\_aic\_results <- my\_aic\_results  
#}  
  
#autoocc.out <- lapply(speciesList, autoocc.fun)