#Multinomial HGAM analysis

#load the libraries

library(mgcv)

library(boot)

library(dplyr)

library(reshape2)

library(lattice)

library(ggplot2)

library(mgcv)

library(plyr)

library(tidyr)

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

# Data Cleaning

df\_dep <- read.csv("C:\\Users\\kaity\\Downloads\\transfer\_8208108\_files\_7e0af806\\AD\_All\_depth.csv", )

colnames(df\_dep)

df\_dep <- df\_dep %>% dplyr::select(-"X")

head(df\_dep)

str(df\_dep)

df\_dep$DE\_DATE <- lubridate::parse\_date\_time(df\_dep$DE\_DATE, orders = c('%Y-%m-%d %H:%M:%S', '%Y-%m-%d'))

df\_dep$MLD <- -df\_dep$MLD

df\_dep$Decadef <- as.factor(df\_dep$Decade)

df\_dep$Reff <- as.factor(df\_dep$Ref)

df\_dep$fFOD <- as.factor(df\_dep$FOD)

df\_dep$fPPD <- as.factor(df\_dep$PPD)

df\_dep$julian <- as.Date(df\_dep$DE\_DATE, format = '%Y-%m-%d %H:%M:%S')

df\_dep$julian <- format(df\_dep$julian, "%j")

df\_dep$Breed\_areaf <- as.factor(df\_dep$Breed\_area)

df\_dep$km\_dist\_ice <- df\_dep$dist\_ice/1000

df\_dep <- df\_dep %>% filter(nbdive < 29 & nbdive > 1)

df\_dep <- df\_dep %>% dplyr::arrange(Ref, DE\_DATE) %>%

group\_by(Ref) %>%

dplyr::mutate(AR.start = if\_else(

row\_number() == 1,

TRUE,

FALSE

)) %>%

ungroup()

df\_dep = df\_dep[!is.na(df\_dep$Bin1\_dep),]

df\_GULF <- df\_dep[df\_dep$Breed\_area == "GULF",]

dep\_long <- reshape2::melt(df\_dep,

id = c("Ref", "DE\_DATE", "nbdive", "Decade",

"Month", "Location", "Breed\_area", "Bathy",

"SIC", "MLD", "doy", "Lat", "Lon", "x", "y",

"FOD", "dist\_ice", "PPD", "xyT", "Sex",

"DayFromStart", "WeekFromStart", "Decadef",

"Reff", "fFOD", "fPPD","julian", "Breed\_areaf",

"km\_dist\_ice", "AR.start"), variable.name = "BIN")

#now multiply the columns by the value in value

dep\_expand <- dep\_long[rep(row.names(dep\_long), dep\_long$value), 1:31]

dep\_expand$BIN\_num <- ifelse(dep\_expand$BIN == "Bin1\_dep", 0,

ifelse(dep\_expand$BIN == "Bin2\_dep", 1,

ifelse(dep\_expand$BIN == "X100.300m", 2, 3)))

dep\_expand <- dep\_expand |> dplyr::arrange(Ref, DE\_DATE)

dep\_expand$fMonth <- as.factor(dep\_expand$Month)

dep\_expand$fSex <- as.factor(dep\_expand$Sex)

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

# Pull out a subset of the data to make the model

dataSub = subset(dep\_expand,

Breed\_area== 'GULF' &

fFOD == '2' &

km\_dist\_ice <10000)

# build the model

mod = gam( data= dataSub,

list(

BIN\_num

~s(km\_dist\_ice, by = Decadef, bs = "cr") + s(Reff, bs='re'),

~s(km\_dist\_ice, by = Decadef, bs = "cr") + s(Reff, bs='re'),

~s(km\_dist\_ice, by = Decadef, bs = "cr") + s(Reff, bs='re')),

family=multinom(K=3))

# Plot on the response scale

library(boot)

plot(mod, all.terms =TRUE, trans = inv.logit)

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

# Points for plotting

library(dplyr)

# Divide the data into distance bins by decade,

dataSub <- dataSub %>%

mutate(distance\_bin = cut(km\_dist\_ice,

breaks = seq(min(km\_dist\_ice, na.rm = TRUE),

max(km\_dist\_ice+50, na.rm = TRUE),

by = 50))) %>%

mutate(distance\_bin = coalesce(distance\_bin, as.factor(km\_dist\_ice)))%>%

filter(!is.na(Ref) & !is.na(km\_dist\_ice))

# Calculate the total number of dives in each distance\_bin

df\_total <- dataSub %>%

dplyr::group\_by(distance\_bin, Decade, FOD, Ref, BIN\_num) %>%

dplyr::summarise(total\_dives = n(), .groups = 'drop')

# Calculate the total number of dives per individual (Ref)

df\_ref\_total <- dataSub %>%

dplyr::group\_by(Ref, FOD, Decade) %>%

dplyr::summarise(ref\_total\_dives = n(), .groups = 'drop')

# Merge the two dataframes to include ref\_total\_dives in df\_total

df\_total <- df\_total %>%

left\_join(df\_ref\_total, by = "Ref")

# The proporiton of each animal's dives that were in each disance bin

df\_total$prop = df\_total$total\_dives/df\_total$ref\_total\_dives

# numeric distancces

df\_total$Dist= as.numeric(df\_total$distance\_bin)\*50

ggplot(df\_total)+

facet\_wrap(~Decade.x+BIN\_num)+

geom\_point(aes(x= Dist, y= prop, color = Ref))+

scale\_color\_discrete(guide="none")+

ggtitle('FOD 2')

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

# Grid space for plotting, we are going to ignore the random effects

# predict(b2, newdata = Rail, exclude = "s(Rail)")

newData = expand.grid(Decadef = unique(dataSub$Decadef),

km\_dist\_ice= seq(0,1000, by =50),

BIN\_num= unique(dataSub$BIN\_num),

Reff = as.factor(0))

# GAM predictions, retursn a list with the median and standard error

aa = predict(mod, type = "response",

newdata = newData,

exclude = "s(Reff)",

se.fit = TRUE)

library(nnet)

nnetMod <- multinom(factor(BIN\_num) ~ km\_dist\_ice + am \* vs,

data = dataSub, trace = FALSE)

mod = gam( data= dataSub,

list(

BIN\_num

~s(km\_dist\_ice, by = Decadef, bs = "cr") + s(Reff, bs='re'),

~s(km\_dist\_ice, by = Decadef, bs = "cr") + s(Reff, bs='re'),

~s(km\_dist\_ice, by = Decadef, bs = "cr") + s(Reff, bs='re')),

family=multinom(K=3))

plot\_predictions(

mod,

type = "probs",

condition = "Decadef") +

facet\_wrap(~Decadef)

newData$Preds = predict(mod,

newdata = newData,

exclude = "s(Reff)", )