````
---
title: "Model_fitting_3_march2025"
author: "Taylor Azizeh"
date: "2025-03-20"
output:
  pdf_document: default
  html_document: default
---

```{r, message = FALSE}
# Clear the workspace
rm(list = ls(all = TRUE))
gc()

# Load libraries
library(tidyverse) # for general reordering, restructuring
library(ggplot2) # for plotting
library(dplyr) # for general reordering, restructuring
library(MASS)  # for negative binomial
library(lme4)  # for mixed models
library(glmmTMB) # for negative binomial + random effects
library(emmeans) # for pairwise comparison
library(ggeffects) # for generating model predictions
library(knitr) # for making tables
library(performance) # for zero-inflation test
library(AER) # for overdispersion test
library(influence.ME) # for Cook's distance test
library(ggcorrplot) # for correlation plots
```

```{r}
# Read in results table from MATLAB
results <- read_csv('/Users/taylorazizeh/Documents/research/active/emperor
penguins/results/MATLAB/final/prey_capture_results.csv')
#head(results)
#glimpse(results)

# Read in morphometrics table (sex, mass)
morpho <- read_csv('/Users/taylorazizeh/Documents/research/active/emperor
penguins/data/cleaned/metadata/metadata.csv')
#head(morpho)

# Remove the "x" from BirdID (leftover from MATLAB)
results <- results %>%
  mutate(BirdID = sub("^x", "", BirdID))  # Removes "x" at the start
#head(results)

# Add in sex and mass
results <- results %>%
  left_join(morpho %>% dplyr::select(BirdID, Sex, StartMass), by = "BirdID")

head(results)
#colnames(results)
```

First, I suspect that transit dives will have little to no foraging, and for my future analyses, I only want to look at foraging dives. I will do a check to see if transit dive have significantly less foraging and can therefore be excluded from analysis.

```{r}
# Create a simplified dataframe: one row per dive
dive_summary <- results %>%
  group_by(BirdID, DiveNumber, DiveType, DiveDuration) %>%
  summarise(TotalEvents = sum(TotalEvents, na.rm = TRUE)) %>%
````
````

```
  ungroup()

head(dive_summary)
```

First, I fit a Poisson (for count data) generalized linear model to see how dive type affects prey capture rate (with dive duration as an offset term).

```{r}
# Poisson
transit_m1 <- glm(
  TotalEvents ~ DiveType +
    offset(log(DiveDuration)),
  family = poisson,
  data = dive_summary
)
summary(transit_m1)

# Overdispersion check function
overdisp_fun <- function(model) {
  rdf <- df.residual(model)
  rp <- residuals(model, type = "pearson")
  Pearson.chisq <- sum(rp^2)
  ratio <- Pearson.chisq / rdf
  return(ratio)
}

overdisp_fun(transit_m1)
```

The model is very overdispersed (dispersion factor of >8).

```{r}
# Refit with neg binom
transit_m2 <- glm.nb(
  TotalEvents ~ DiveType + offset(log(DiveDuration)),
  data = dive_summary)

summary(transit_m2)
anova(transit_m2, test = "Chisq")

# Overdispersion check function
overdisp_fun <- function(model) {
  rdf <- df.residual(model)
  rp <- residuals(model, type = "pearson")
  Pearson.chisq <- sum(rp^2)
  ratio <- Pearson.chisq / rdf
  return(ratio)
}

overdisp_fun(transit_m2)
```

This fixed the overdispersion (overdispersion ratio is 0.8)
```{r}
# Pearson residuals vs fitted values
plot(fitted(transit_m2), residuals(transit_m2, type = "pearson"),
     xlab = "Fitted values", ylab = "Pearson residuals",
     main = "Residuals vs Fitted")
abline(h = 0, col = "red")

# Q-Q plots
qqnorm(residuals(transit_m2, type = "pearson"))
qqline(residuals(transit_m2, type = "pearson"), col = "red")
```

```
# Residuals
hist(residuals(transit_m2, type = "pearson"),
     main = "Histogram of Pearson Residuals",
     xlab = "Residuals")

# Cooks Distance
library(car)
influencePlot(transit_m2, id.method = "identify", main = "Influence Plot")
```

Cook's distance of 0.06 indicates that values are within typical thresholds (values >1 are a bigger concern)

```{r}
library(emmeans)
emm <- emmeans(transit_m2, ~ DiveType)
summary(emm)

# Back-transform emmeans to the response scale
exp_summary <- summary(emm) %>%
  dplyr::mutate(
    rate = exp(emmean), # Number of events per dive
    lower.CL = exp(asymp.LCL),
    upper.CL = exp(asymp.UCL)
  )
print(exp_summary)

library(ggplot2)
ggplot(exp_summary, aes(x = DiveType, y = rate)) +
  geom_point(size = 3) +
  geom_errorbar(aes(ymin = lower.CL, ymax = upper.CL), width = 0.2) +
  labs(y = "Estimated prey capture rate per dive", x = "Dive Type") +
  theme_minimal()
```

For results:
I used a negative binomial generalized linear model to evaluate whether transit dives exhibited a meaningful level of foraging attempts. The model included dive type as a categorical predictor and dive duration as a log-transformed offset term to account for dive duration. While the effect of dive type on prey cpature rate was statistically significant (x^2 = 1168.3, df = 3, p < 0.001), the estimated number of prey capture attempts per transit dive was low (mean = 0.79 events per dive, 95% CI: 0.68 - 0.91), compared to higher rates in mesopelagic (8.19), benthic (7.49), and epipelagic (4.11) dives. Although prey capture was not entirely absent from transit dives, the estimated rate was nearly an order of magnitude lower than other dive types. Based on this, I excluded transit dives from subsequent analyses focused on testing hypotheses related to foraging dive behavior.

Hypothesis 1: The rate of prey capture attempts will be highest in the bottom phase of the dive, where emperor penguins are believed to be primarily hunting.

Now I'm going to filter out any non-foraging and dives because we aren't looking at non-foraging dives within this question.

```{r}
# Filter for foraging dives and to remove transit dives
foraging_dives <- results %>%
  filter(TotalEvents >= 1) %>%
  filter(DiveType != "Transit") # Remove transit dives

# Check the outputs
nrow(results)
nrow(foraging_dives)
```

This indicates that 40% of epipelagic, mesopelagic, and benthic penguin dives include at least one foraging attempt. 7% of dives were transit dives.

ZUUR STEP ONE - Explore Your Data (EDA) (pg. 12)

```{r}
# First time only: install and import fonts
#install.packages("showtext")
library(showtext)

# Add Times New Roman support
font_add(family = "Times New Roman", regular = "Times New Roman.ttf")  # Use full path if needed
showtext_auto()

# Now use it in ggplot
ggplot(foraging_dives, aes(x = BirdID, y = TotalEvents)) +
  geom_boxplot(fill = "gray80", color = "black", outlier.shape = 1, outlier.size = 1.5) +
  labs(
    x = "Individual Penguin ID",
    y = "Number of prey capture events per dive",
    title = "Variation in foraging intensity across individuals"
  ) +
  theme_minimal(base_size = 12, base_family = "Times New Roman") +
  theme(
    axis.text.x = element_text(angle = 45, hjust = 1),
    plot.title = element_text(hjust = 0.5, face = "bold")
  )

# Create a histogram plot
hist(foraging_dives$TotalEvents,
         xlab = "Frequency",
         ylab = "Number of prey capture events (per dive)",
         main = "Total prey capture events per dive (all phases)")
```

The histogram looks like it has a high frequency of zeroes, but it's just the scale. It actually starts at 1.

According to Zuur, extreme values within the Cleveland dotplots may be outliers. It looks like potential outliers on the far right of the graph may be \>80 events, and that may be constrained within one or two birds.

```{r}
# Violin plot showing density of prey capture events
ggplot(foraging_dives) +
  geom_violin(aes(x = "Descent", y = ToEvents, fill = "Descent Phase"), alpha = 0.6) +
  geom_violin(aes(x = "Bottom", y = DestEvents, fill = "Bottom Phase"), alpha = 0.6) +
  geom_violin(aes(x = "Ascent", y = FromEvents, fill = "Ascent Phase"), alpha = 0.6) +
  labs(x = "Dive Phase", y = "Prey Capture Events", title = "Density of Prey Capture Events Across Dive Phases") +
  theme_minimal()
```

Looking at the potential outliers, I checked for any rows that have TotalEvents \>60.

```{r}
pot_outliers <- foraging_dives %>%
  filter(foraging_dives$TotalEvents > 60)

head(pot_outliers)
```

I believe that Penguin 19EP_304f's row with 96 prey captures within 51 seconds could be erroneous and biologically implausible, but the other two rows have high number of prey

capture attempts spread across all three phases and longer phase durations. These may likely be valid data.

Remove the one 19EP_304f outlier.

```{r}
foraging_dives <- foraging_dives %>%
  filter(TotalEvents <= 90)

nrow(results)
nrow(foraging_dives)
```

Replot the data without outlier.
```{r}
# Add Times New Roman support
font_add(family = "Times New Roman", regular = "Times New Roman.ttf")  # Use full path if needed
showtext_auto()

# Now use it in ggplot
ggplot(foraging_dives, aes(x = BirdID, y = TotalEvents)) +
  geom_boxplot(fill = "gray80", color = "black", outlier.shape = 1, outlier.size = 1.5) +
  labs(
    x = "Individual Penguin ID",
    y = "Number of prey capture events per dive",
    title = "Variation in foraging intensity across individuals"
  ) +
  theme_minimal(base_size = 12, base_family = "Times New Roman") +
  theme(
    axis.text.x = element_text(angle = 45, hjust = 1),
    plot.title = element_text(hjust = 0.5, face = "bold")
  )

phase_long <- foraging_dives %>%
  dplyr::select(BirdID, DiveNumber, ToEvents, DestEvents, FromEvents) %>%
  pivot_longer(cols = c(ToEvents, DestEvents, FromEvents),
               names_to = "Phase", values_to = "Events") %>%
  dplyr::mutate(Phase = dplyr::recode(Phase,
                        "ToEvents" = "Descent",
                        "DestEvents" = "Bottom",
                        "FromEvents" = "Ascent"))

# Add and activate Times New Roman font (if not already done)
font_add(family = "Times New Roman", regular = "Times New Roman.ttf")  # Use full path if needed
showtext_auto()

# Boxplot by dive phase
ggplot(phase_long, aes(x = Phase, y = Events)) +
  geom_boxplot(fill = "gray80", color = "black", outlier.shape = 1, outlier.size = 1.5) +
  labs(
    x = "Dive Phase",
    y = "Prey capture events per dive",
  ) +
  theme_minimal(base_size = 12, base_family = "Times New Roman") +
  theme(
    axis.text.x = element_text(angle = 0),
    plot.title = element_text(hjust = 0.5, face = "bold")
  )
```

Only the one row was removed, decreasing the sample size from 17494 to 17493.

ZUUR STEP TWO: Check for collinearity (pg. 473)

```{r}
# Check for collinearity across predictors
numeric_vars <- foraging_dives %>%
  dplyr::select(DiveDuration, ToEvents, DestEvents, FromEvents, TotalEvents,
ToPhaseDuration, DestPhaseDuration, FromPhaseDuration, StartMass)

cor <- cor(numeric_vars, use = "everything")
print(cor)

library(corrplot)
corrplot(cor, method = "color", type = "upper",
         tl.col = "black", addCoef.col = "black",
         number.cex = 0.77)

# Pairs plots
library(GGally)
ggpairs(numeric_vars)
```

Collinearity looks okay.

ZUUR STEP THREE: Fit maximal model.

Now that we have identified an appropriate modelling structure (Poisson GLMM to start), we can fit the full model.

```{r}
# Pivot the table
foraging_long <- foraging_dives %>%
  pivot_longer(cols = c(ToEvents, DestEvents, FromEvents),
               names_to = "DivePhase",
               values_to = "PreyCaptureEvents") %>%
  dplyr::mutate(DivePhase = dplyr::recode(DivePhase,
                              "ToEvents" = "Descent",
                              "DestEvents" = "Bottom",
                              "FromEvents" = "Ascent")) %>%
  pivot_longer(cols = c(ToPhaseDuration, DestPhaseDuration, FromPhaseDuration),
               names_to = "DurationPhase",
               values_to = "PhaseDuration") %>%
  dplyr::mutate(DurationPhase = dplyr::recode(DurationPhase,
                                "ToPhaseDuration" = "Descent",
                                "DestPhaseDuration" = "Bottom",
                                "FromPhaseDuration" = "Ascent")) %>%
  dplyr::filter(DivePhase == DurationPhase) %>%  # keep matched phase rows only
  dplyr::mutate(Year = substr(BirdID, 1, 2))  # add Year

foraging_long <- as.data.frame(foraging_long)
head(foraging_long)
```

```{r}
# Fit the full model
model1 <- glmmTMB(
  PreyCaptureEvents ~ DivePhase + DiveType + Sex + StartMass + Year +
    offset(log(PhaseDuration)) +
    (1 | BirdID),
  family = poisson,
  data = foraging_long)

summary(model1)
```

According to Zuur, (pg. 145) you need to check for autocorrelation before simplifying the fixed and random effects.

```{r}
# Check for autocorrelation and overdispersion
acf(residuals(model1, type = "pearson"))

overdisp_fun <- function(model) {
  rdf <- df.residual(model)
  rp <- residuals(model, type = "pearson")
  Pearson.chisq <- sum(rp^2)
  ratio <- Pearson.chisq / rdf
  return(ratio)
}

overdisp_fun(model1)
```

As expected, there is significant temporal autocorrelation and overdispersion. We will bin the data into thirty minute groups to rerun the maximal model.

```{r}
# Convert DiveStartTime to POSIXct and assign 30-min bins
foraging_long <- foraging_long %>%
  mutate(
    DiveStartTime = as.POSIXct(DiveStartTime, format = "%d-%b-%Y %H:%M:%OS", tz = "UTC"),
    HalfHourBin = cut(DiveStartTime, breaks = "30 min")
  )

# Summarize by half-hour bins within each individual, dive type, and phase
bin_summary_foraging <- foraging_long %>%
  mutate(
    DiveStartTime = as.POSIXct(DiveStartTime, format = "%d-%b-%Y %H:%M:%OS", tz = "UTC"),
    HalfHourBin = cut(DiveStartTime, breaks = "30 min")
  ) %>%
  group_by(BirdID, Year, Sex, StartMass, DiveType, DivePhase, HalfHourBin) %>%
  summarise(
    TotalEvents = sum(PreyCaptureEvents, na.rm = TRUE),
    TotalPhaseDuration = sum(PhaseDuration, na.rm = TRUE),
    n_dives = n_distinct(DiveNumber),
    .groups = "drop"
  ) %>%
  rename(Phase = DivePhase) %>%
  arrange(BirdID, HalfHourBin) %>%
  group_by(BirdID) %>%
  mutate(TimeFactor = factor(dense_rank(HalfHourBin))) %>%
  ungroup()

# Preview the result
head(bin_summary_foraging)
```

Refit the model with a different distribution and temporal correlation structure.

```{r}
# Create bins every 30 minutes
bin_summary_foraging <- bin_summary_foraging %>%
  group_by (BirdID) %>%
  arrange (HalfHourBin) %>%
  mutate (TimeFactor = factor (dense_rank (HalfHourBin))) %>%
  ungroup ()

model2 <- glmmTMB(
  TotalEvents ~ Phase + DiveType + Sex + StartMass + Year +
    offset(log(TotalPhaseDuration)) +
    (1 | BirdID) +
    ar1(TimeFactor + 0 | BirdID),
```

```
  family = nbinom2,
  data = bin_summary_foraging
)

summary(model2)
```

```{r}
# Let's check for overdispersion and autocorrelation
# Raw Pearson residuals
acf(residuals(model2, type = "pearson"))

overdisp_fun <- function(model) {
  rdf <- df.residual(model)
  rp <- residuals(model, type = "pearson")
  Pearson.chisq <- sum(rp^2)
  ratio <- Pearson.chisq / rdf
  return(ratio)
}

overdisp_fun(model2)
AIC(model1, model2)
```

This fixed the autocorrelation (acf graph) and overdispersion (0.95). Now we can start simplifying the random effects by removing ID and see if the model improves.

```{r}
# Run the new model
model3 <- glmmTMB(
  TotalEvents ~ Phase + DiveType + Sex + StartMass + Year +
    offset(log(TotalPhaseDuration)) +
    ar1(TimeFactor + 0 | BirdID),
  family = nbinom2,
  data = bin_summary_foraging
)

summary(model3)
anova(model2, model3) # Zuur recommends using Likelihood Ratio Tests instead of AIC
```

The model that included penguin ID has the lower AIC, and therefore fits the data better.

Now I can move on to simplify the fixed effects. I'll start with year.

```{r}
# Refit the model with NA sex birds remove
# Need to remove birds with no sex data
bin_summary_foraging_sex <- bin_summary_foraging %>%
  filter(!is.na(Sex))

model3_na_sex <- glmmTMB(
  TotalEvents ~ Phase + DiveType + Sex + StartMass + Year +
    offset(log(TotalPhaseDuration)) +
    (1 | BirdID) +
    ar1(TimeFactor + 0 | BirdID),
  family = nbinom2,
  data = bin_summary_foraging_sex
)

# Remove starting mass
model4 <- update(model3_na_sex, . ~ . - StartMass)

# Remove Sex
model5 <- update(model4, . ~ . - Sex)
```

```
# Remove Sex
model6 <- update(model5, . ~ . - Year)

# Then compare:
anova(model3_na_sex, model4, model5, model6)
```

Removing sex, year, and starting mass created the most parsimonious model. Now, I want to include an interaction term between phase and dive type to see if that improves the model.

```{r}
model6 <- glmmTMB(
  TotalEvents ~ Phase + DiveType +
    offset(log(TotalPhaseDuration)) +
    (1 | BirdID) +
    ar1(TimeFactor + 0 | BirdID),
  family = nbinom2,
  data = bin_summary_foraging
)

# Interaction term
model6_interact <- glmmTMB(
  TotalEvents ~ Phase * DiveType +
    offset(log(TotalPhaseDuration)) +
    ar1(TimeFactor + 0 | BirdID),
  family = nbinom2,
  data = bin_summary_foraging
)

anova(model6, model6_interact)
```

The interaction term really improved the model fit (AIC delta 2000, $p < 0.001$).

```{r}
# Add zero-inflation to the interaction model
model6_interact_zi <- glmmTMB(
  TotalEvents ~ Phase * DiveType +
    offset(log(TotalPhaseDuration)) +
    ar1(TimeFactor + 0 | BirdID),
  ziformula = ~ Phase,
  family = nbinom2,
  data = bin_summary_foraging
)

# Compare with non-zero-inflated version
anova(model6_interact, model6_interact_zi)
```

Adding a zero-inflation term to Phase significantly improves model fit (delta AIC is 478, $p < 0.001$).

```{r}
# Check out model output
summary(model6_interact_zi)

# See classic residuals
# Pearson residuals vs fitted values
plot(fitted(model6_interact_zi), residuals(model6_interact_zi, type = "pearson"),
     xlab = "Fitted values", ylab = "Pearson residuals",
     main = "Residuals vs Fitted")
abline(h = 0, col = "red")

# Q-Q plots
```

```
qqnorm(residuals(model6_interact_zi, type = "pearson"))
qqline(residuals(model6_interact_zi, type = "pearson"), col = "red")

# Residuals
hist(residuals(model6_interact_zi, type = "pearson"),
     main = "Histogram of Pearson Residuals",
     xlab = "Residuals")
```

```{r}
# Libraries
library(ggplot2)
library(emmeans)
library(multcompView)
library(showtext)

# Add and activate Times New Roman font
font_add(family = "Times New Roman", regular = "Times New Roman.ttf")  # Use full path if
needed
showtext_auto()

# Phase duration summaries
phase_means <- bin_summary_foraging %>%
  group_by(Phase) %>%
  summarise(mean_duration = mean(TotalPhaseDuration, na.rm = TRUE))

# Get estimated marginal means grid
emm_grid <- ref_grid(model6_interact_zi,
                     at = list(Phase = unique(bin_summary_foraging$Phase),
                               DiveType = unique(bin_summary_foraging$DiveType)))

# Merge with mean durations
emm_data <- as.data.frame(emm_grid) %>%
  left_join(phase_means, by = "Phase")

# Estimate marginal means with offsets
emm_phase_type <- emmeans(model6_interact_zi,
                          ~ Phase * DiveType,
                          offset = log(emm_data$mean_duration))

# Prepare for plotting
plot_data <- summary(emm_phase_type) %>%
  mutate(
    Events = exp(emmean),
    lower.CL = exp(asymp.LCL),
    upper.CL = exp(asymp.UCL),
    Phase = factor(Phase, levels = c("Descent", "Bottom", "Ascent"))
  )

# Main Plot
ggplot(plot_data, aes(x = Phase, y = Events, fill = DiveType)) +
  geom_col(position = position_dodge(width = 0.8), width = 0.7, color = "black") +
  geom_errorbar(
    aes(ymin = lower.CL, ymax = upper.CL),
    position = position_dodge(width = 0.8), width = 0.2
  ) +
  scale_fill_manual(values = c("azure4", "antiquewhite4", "azure3")) +
  labs(
    x = "Dive Phase",
    y = "Estimated prey capture events\n(per average-duration phase)",
    fill = "Dive Type",
    title = "Predicted prey capture events by phase and dive type"
  ) +
  theme_minimal(base_family = "Times New Roman") +
  theme(
```

```
    text = element_text(size = 12),
    axis.text.x = element_text(size = 11),
    axis.text.y = element_text(size = 11),
    legend.position = "top",
    legend.text = element_text(size = 9),
    panel.grid.major.x = element_blank()
  )
```

```{r}
phase_means <- bin_summary_foraging %>%
  group_by(Phase) %>%
  summarise(mean_duration = mean(TotalPhaseDuration, na.rm = TRUE))

library(dplyr)

# Merge average durations into plot_data
effect_size_table <- plot_data %>%
  left_join(phase_means, by = "Phase") %>%
  dplyr::mutate(
    Events = round(Events, 2),
    lower.CL = round(lower.CL, 2),
    upper.CL = round(upper.CL, 2),
    mean_duration = round(mean_duration, 1),
    Units = paste0("Events per ", mean_duration, " sec phase")
  ) %>%
  dplyr::select(Phase, DiveType, Events, lower.CL, upper.CL, mean_duration, Units) %>%
  arrange(Phase, DiveType)

# View table
print(effect_size_table)
write.csv(effect_size_table, "/Users/taylorazizeh/Documents/research/active/emperor
penguins/results/R/Effect_Size_Table.csv", row.names = FALSE)
```

For results:
To determine the effect of dive phase and type on prey capture attempts, I fit a GLMM with a negative binomial distribution and an AR(1) correlation structure to account for repeated measures within individual birds. The model included an interaction term between dive phase and type, an offset for phase duration, and a zero-inflation term conditional on dive phase to account for excess zeroes. There was a strong effect of dive phase on prey capture attempts, with significantly more attempts occuring during the bottom phase compared to the ascent and descent phases. The descent phase had significantly fewer attempts than the ascent (B = −0.76 +− 0.08 SE, z = −9.14, p < 0.001), with the bottom phase have the highest number of prey capture attempts (B = 2.21 +− 0.06 SE, z = 34.92, p < 0.001). There was also a significant interaction between phase and dive type. The number of prey capture attempts during the bottom phase was substantially lower in epipelagic dives compared to benthic dives (epipelagic x bottom phase: B = −1.51 +− 0.07 SE, z = −22.97, p < 0.001), while mesopelagic dives showed similar bottom-phase rates to benthic dives (B = −0.09 +− 0.07 SE, p = 0.22). During the descent phase, prey capture attempts in mesopelagic dives were significantly lower than in benthic dives (mesopelagic x descent phase: B = −0.29 +− 0.09 SE, p = 0.002), while epipelagic descent rates did not differ significantly did not differ significantly from benthic. Adding a zero-inflation term for dive phase supported the data. The likelihood of zero prey capture events was significantly higher during descent phases (B = 3.74 +− 0.89 SE, z = 4.19, p < 0.001), but not during bottom phases (p = 0.12).

Hypothesis 2: Pelagic dives will have a higher rate of prey capture attempts, compared to benthic dives, because benthic prey are predicted to be of a higher caloric content.

ZUUR STEP ONE - Explore Your Data (EDA) (pg. 12)

```{r}
```

```
# Filter out transit dives and only include foraging dives

# Create a simplified dataframe: one row per dive
dive_summary <- results %>%
  filter(DiveType != "Transit") %>%  # Remove transit dives
  filter(TotalEvents >= 1) %>%
  group_by(BirdID, DiveNumber, DiveType, DiveDuration, Sex, StartMass) %>%
  summarise(TotalEvents = sum(TotalEvents, na.rm = TRUE), .groups = "drop") %>%
  mutate(Year = substr(BirdID, 1, 2))  # Create Year column

# View
head(dive_summary)
nrow(dive_summary)
```

```{r}
# Create a box plot
boxplot(dive_summary$TotalEvents ~ factor(dive_summary$DiveType),
         xlab = "ID",
         ylab = "Number of prey capture events (per dive)",
         main = "Total prey capture events per dive",
         data = dive_summary)

# Check prey capture events by dive type
ggplot(dive_summary, aes(x = DiveType, y = TotalEvents)) +
  geom_boxplot(width = 0.1) +
  labs(title = "Distribution of Prey Capture Events by Dive Type",
       y = "Total Prey Capture Events",
       x = "Dive Type") +
  theme_minimal()

# Boxplot of dive duration by type
ggplot(dive_summary, aes(x = DiveType, y = DiveDuration)) +
  geom_boxplot(fill = "lightgreen", alpha = 0.6) +
  labs(title = "Dive Duration by Dive Type",
       y = "Dive Duration (s)",
       x = "Dive Type") +
  theme_minimal()

# Scatter plot
ggplot(dive_summary, aes(x = DiveDuration, y = TotalEvents, color = DiveType)) +
  geom_point(alpha = 0.5) +
  geom_smooth(method = "lm", se = FALSE) +
  labs(title = "Prey Capture Events vs Dive Duration",
       x = "Dive Duration (s)",
       y = "Total Prey Capture Events") +
  theme_minimal()
```

```{r}
# Check for collinearity
numeric_vars <- dive_summary %>%
  dplyr::select(DiveDuration, StartMass, TotalEvents)

cor <- cor(numeric_vars, use = "everything")
print(cor)

library(corrplot)
corrplot(cor, method = "color", type = "upper",
         tl.col = "black", addCoef.col = "black",
         number.cex = 0.77)

# Pairs plots
library(GGally)
ggpairs(numeric_vars)
```

```

Fit the maximal model.

```{r}
# Fit maximal model
model7 <- glmmTMB(
  TotalEvents ~ DiveType + Sex + StartMass + Year +
    offset(log(DiveDuration)) +
    (1 | BirdID),
  family = poisson,
  data = dive_summary)

# Model summary
summary(model7)
```

Check for overdispersion and autocorrelation.

```{r}
acf(residuals(model7, type = "pearson"))

overdisp_fun <- function(model) {
  rdf <- df.residual(model)
  rp <- residuals(model, type = "pearson")
  Pearson.chisq <- sum(rp^2)
  ratio <- Pearson.chisq / rdf
  return(ratio)
}
overdisp_fun(model7)
```

There is significant overdispersion and autocorrelation, so we will switch to a negative binomial distribution and add a correlation structure.

```{r}
# Filter, summarize, and prep
dive_summary_bins <- results %>%
  filter(DiveType != "Transit") %>%           # Remove transit dives
  filter(TotalEvents >= 1) %>%                # Keep only foraging dives
  group_by(BirdID, DiveNumber, DiveType, DiveDuration, Sex, StartMass) %>%
  summarise(TotalEvents = sum(TotalEvents, na.rm = TRUE), .groups = "drop") %>%
  mutate(Year = substr(BirdID, 1, 2))         # Add Year column

# Create DiveStartTime as POSIXct
dive_summary_bins <- dive_summary_bins %>%
  left_join(results %>%
              dplyr::select(BirdID, DiveNumber, DiveStartTime) %>% distinct(),
            by = c("BirdID", "DiveNumber")) %>%
  mutate(DiveStartTime = as.POSIXct(DiveStartTime, format = "%d-%b-%Y %H:%M:%OS", tz =
"UTC"))

# Create HalfHourBin and TimeFactor
dive_summary_bins <- dive_summary_bins %>%
  mutate(
    HalfHourBin = cut(DiveStartTime, breaks = "30 min"),
    HalfHourBinID = as.integer(as.factor(HalfHourBin))
  ) %>%
  group_by(BirdID) %>%
  arrange(HalfHourBin) %>%
  mutate(TimeFactor = factor(dense_rank(HalfHourBin))) %>%
  ungroup()

head(dive_summary_bins)
```
```

```{r}
# Fit maximal model
model8 <- glmmTMB(
  TotalEvents ~ DiveType + Sex + StartMass + Year +
    offset(log(DiveDuration)) +
    (1 | BirdID) +
    ar1(TimeFactor + 0 | BirdID),
  family = nbinom2,
  data = dive_summary_bins
)

summary(model8)
acf(residuals(model8, type = "pearson"))

# Check for overdispersion
overdisp_fun(model8)
```

The overdispersion and autocorrelation are much better now. We can now move forward to simplifying the random effects by trying to remove BirdID.

```{r}
# Fit maximal model
model9 <- update(model8, . ~ . - (1 | BirdID))

# Model summary
summary(model9)
anova(model8, model9)
```

Removing ID did not improve so we will reintroduce it. Now we can start simplifying the fixed effects.

```{r}
# Refit the model with no sex birds removed
dive_summary_na_sex <- dive_summary_bins %>%
  filter(!is.na(Sex))

model9_no_sex <- glmmTMB(
  TotalEvents ~ DiveType + Sex + StartMass + Year +
    offset(log(DiveDuration)) +
    (1 | BirdID) +
    ar1(TimeFactor + 0 | BirdID),
  family = nbinom2,
  data = dive_summary_na_sex)

model10 <- update(model9_no_sex, . ~ . - Sex)

# Model summary
anova(model9_no_sex, model10)
```
Removing sex improved the model.
```{r}
model11 <- glmmTMB(
  TotalEvents ~ DiveType + StartMass + Year +
    offset(log(DiveDuration)) +
    (1 | BirdID) +
    ar1(TimeFactor + 0 | BirdID),
  family = nbinom2,
  data = dive_summary_bins)

model12 <- update(model11, . ~ . - StartMass)
model13 <- update(model12, . ~ . - Year)
```

```
anova(model11, model12, model13)
```

The best model includes Year, but not sex or starting mass. I'm going to check model fit with a zero inflation factor.

```{r}
model12 <- glmmTMB(
  TotalEvents ~ DiveType + Year +
    offset(log(DiveDuration)) +
    (1 | BirdID) +
    ar1(TimeFactor + 0 | BirdID),
  family = nbinom2,
  data = dive_summary_bins)

model14 <- glmmTMB(
  TotalEvents ~ DiveType + Year +
    offset(log(DiveDuration)) +
    (1 | BirdID) +
    ar1(TimeFactor + 0 | BirdID),
  family = nbinom2,
  ziformula = ~ DiveType,
  data = dive_summary_bins)

anova(model12, model14)
```

The zero-inflation model did not fit the data better.

```{r}
# See classic residuals
# Pearson residuals vs fitted values
plot(fitted(model12), residuals(model12, type = "pearson"),
     xlab = "Fitted values", ylab = "Pearson residuals",
     main = "Residuals vs Fitted")
abline(h = 0, col = "red")

# Q-Q plots
qqnorm(residuals(model12, type = "pearson"))
qqline(residuals(model12, type = "pearson"), col = "red")

# Residuals
hist(residuals(model12, type = "pearson"),
     main = "Histogram of Pearson Residuals",
     xlab = "Residuals")
```

```{r}
summary(model12)
```

```{r}
library(ggplot2)

df <- data.frame(
  Term = c("Epipelagic vs Benthic", "Mesopelagic vs Benthic", "Year 2022 vs 2019"),
  Estimate = c(0.01053, 0.01176, 0.19997),
  SE = c(0.03658, 0.04016, 0.08282)
)

df$RR <- exp(df$Estimate)
df$CI_low <- exp(df$Estimate - 1.96 * df$SE)
df$CI_high <- exp(df$Estimate + 1.96 * df$SE)

ggplot(df, aes(x = RR, y = Term)) +
  geom_point(size = 3) +
```

```
  geom_errorbarh(aes(xmin = CI_low, xmax = CI_high), height = 0.2) +
  geom_vline(xintercept = 1, linetype = "dashed", color = "gray50") +
  scale_x_log10() +
  labs(
    x = "Rate Ratio (log scale)",
    y = "",
    title = "Effect of Dive Type and Year on Prey Capture Attempts",
    subtitle = "Rate ratios with 95% Wald confidence intervals"
  ) +
  theme_minimal(base_family = "Times New Roman") +
  theme(text = element_text(size = 12))

library(dplyr)
library(broom)

# Grouped summary table with mean and CI for TotalEvents and DiveDuration
descriptive_table <- dive_summary_bins %>%
  group_by(DiveType) %>%
  summarise(
    Mean_Events = mean(TotalEvents, na.rm = TRUE),
    SD_Events = sd(TotalEvents, na.rm = TRUE),
    N = n(),
    SE_Events = SD_Events / sqrt(N),
    CI_Low_Events = Mean_Events - 1.96 * SE_Events,
    CI_High_Events = Mean_Events + 1.96 * SE_Events,

    Mean_Duration = mean(DiveDuration, na.rm = TRUE),
    SD_Duration = sd(DiveDuration, na.rm = TRUE),
    SE_Duration = SD_Duration / sqrt(N),
    CI_Low_Duration = Mean_Duration - 1.96 * SE_Duration,
    CI_High_Duration = Mean_Duration + 1.96 * SE_Duration
  )


```


Final model interpretation:
There wasn't an effect of dive type on prey capture rate once duration is accounted for.
There is moderate variation between individuals and strong autocorrelation (p = 0.79)
which validates the need for the correlation structure.

Results figures
```{r}
ggplot(dive_summary_bins, aes(x = DiveType, y = TotalEvents, fill = DiveType)) +
  geom_boxplot(alpha = 0.6, outlier.shape = NA) +
  geom_jitter(width = 0.2, alpha = 0.2, size = 1) +
  scale_fill_brewer(palette = "Set2") +
  labs(
    title = "Raw Prey Capture Events by Dive Type",
    y = "Prey Capture Events",
    x = "Dive Type"
  ) +
  theme_minimal(base_size = 14) +
  theme(legend.position = "none")

ggplot(dive_summary_bins, aes(x = DiveType, y = TotalEvents, fill = DiveType)) +
  geom_boxplot(alpha = 0.6, outlier.shape = NA) +
  geom_jitter(width = 0.2, alpha = 0.2, size = 0.8) +
  facet_wrap(~ BirdID, scales = "free_y") +
  scale_fill_brewer(palette = "Dark2") +
  labs(
    title = "Prey Capture Rate by Dive Type Across Individuals",
    x = "Dive Type",
    y = "Prey Capture Rate (events per second)"
```

```
  ) +
  theme_bw(base_size = 11) +
  theme(
    legend.position = "none",
    strip.background = element_rect(fill = "grey90")
  )

```

Hypothesis 3: Energy expenditure (activity) will increase with prey capture rate, depending on dive type.

Zuur STEP ONE: EDA

```{r}
# Filter out transit dives, keep foraging dives, and remove any NAs
odba_dives<- results %>%
  filter(DiveType != "Transit") %>%
  filter(!is.na(ODBA_Total), !is.na(TotalEvents)) %>%
  filter(TotalEvents >= 1)

summary(odba_dives)

# Scatterplot
ggplot(odba_dives, aes(x = TotalEvents / DiveDuration, y = ODBA_Total, color = DiveType))
+
  geom_point(alpha = 0.5) +
  geom_smooth(method = "lm", se = FALSE) +
  theme_minimal() +
  labs(x = "Prey Capture Rate (events/sec)", y = "ODBA Total")
```

Let's check the outliers.

```{r}
# Make sure variables are formatted correctly
Q1 <- quantile(results$ODBA_Total, 0.25, na.rm = TRUE)
Q3 <- quantile(results$ODBA_Total, 0.75, na.rm = TRUE)
IQR_value <- Q3 - Q1
upper_bound <- Q3 + 3 * IQR_value  # Conservative

# Filter out outliers
results_clean <- results %>%
  filter(ODBA_Total <= upper_bound)

# Check how many rows were removed
nrow(results) - nrow(results_clean)

# Replot
ggplot(results_clean, aes(x = TotalEvents, y = ODBA_Total, color = DiveType)) +
  geom_point(alpha = 0.4) +
  theme_minimal()
```

```{r}
# Plot a histogram of the data
hist(results_clean$ODBA_Total, breaks = 50)

#Check for collinearity
library(car)
vif(lm(ODBA_Total ~ TotalEvents * DiveType + StartMass + Sex, data = results_clean))
```

```{r}
# Reformat table
```

```
results_clean <- results_clean %>%
  filter(DiveType != "Transit") %>%
  filter(TotalEvents >= 1) %>%
  mutate(
    Year = substr(BirdID, 1, 2),
    Sex = factor(Sex),
    DiveType = factor(DiveType)
  )

head(results_clean)
unique(results_clean$BirdID)
```

Fit the maximal model.

Normally, I would use lmer() to fit the model, but I know that I am going to have to fit a correlation strucure, so I am going to start with glmmTMB.

```{r}
model13 <- glmmTMB(
  ODBA_Total ~ TotalEvents * DiveType + StartMass + Sex + Year +
    offset(log(DiveDuration)) +
    (1 | BirdID),
  family = gaussian(),
  data = results_clean)

summary(model13)
```

Now I am going to remove the random effect for ID and compare.

```{r}
model14 <- glmmTMB(
  ODBA_Total ~ TotalEvents * DiveType + StartMass + Sex + Year +
    offset(log(DiveDuration)),
  data = results_clean,
  family = gaussian()
)

summary(model14)
```

This model converged correctly but we know that there is likely variability among individuals. So, I'm going to simplify the fixed effects and refit. First, I'm going to check the autocorrelation.

```{r}
acf(residuals(model14, type = "pearson"))
```

As expected, there is temporal autocorrelation. So we can set a 30-minute bin correlation structure. Then we will simplify effects.

```{r}
# Let's bin the data first
# Filter and clean
results_binned <- results_clean %>%
  filter(DiveType != "Transit") %>%  # Remove transit dives
  filter(TotalEvents >= 1)           # Keep only foraging dives

# Create DiveStartTime as POSIXct
results_binned <- results_binned %>%
  mutate(DiveStartTime = as.POSIXct(DiveStartTime, format = "%d-%b-%Y %H:%M:%OS", tz =
"UTC"))
```

```r
# Create HalfHourBin and TimeFactor
results_binned <- results_binned %>%
  mutate(
    HalfHourBin = cut(DiveStartTime, breaks = "30 min"),
    HalfHourBinID = as.integer(as.factor(HalfHourBin))
  ) %>%
  group_by(BirdID) %>%
  arrange(HalfHourBin) %>%
  mutate(TimeFactor = factor(dense_rank(HalfHourBin))) %>%
  ungroup()

# Quick check
head(results_binned)
```

We can fit the correlation structure and recheck.

```{r}
model15 <- glmmTMB(
  ODBA_Total ~ TotalEvents * DiveType + StartMass + Sex + Year +
    offset(log(DiveDuration)) +
    ar1(TimeFactor + 0 | BirdID),
  data = results_binned,
  family = gaussian()
)

acf(residuals(model15, type = "pearson"))
```

That fixed the autocorrelation. Now we simplify fixed effects.

```{r}
# Refit the model with no sex birds removed
results_clean_no_sex <- results_binned %>%
  filter(!is.na(Sex))

model15_no_sex <- glmmTMB(
  ODBA_Total ~ TotalEvents * DiveType + StartMass + Sex + Year +
    offset(log(DiveDuration)) +
    ar1(TimeFactor + 0 | BirdID),
  data = results_clean_no_sex,
  family = gaussian()
)

model16 <- glmmTMB(
  ODBA_Total ~ TotalEvents * DiveType + StartMass + Year +
    offset(log(DiveDuration)) +
    ar1(TimeFactor + 0 | BirdID),
  data = results_clean_no_sex,
  family = gaussian()
)

anova(model15_no_sex, model16)
```

Including sex fits the data better, now we can check Year.

```{r}
model17 <- glmmTMB(
  ODBA_Total ~ TotalEvents * DiveType + StartMass + Sex +
    offset(log(DiveDuration)) +
    ar1(TimeFactor + 0 | BirdID),
  data = results_binned,
  family = gaussian()
)
```

```
anova(model15, model17)
```

Including year also fit the data better. Let's try excluding mass.

```{r}
model18 <- glmmTMB(
  ODBA_Total ~ TotalEvents * DiveType + Year + Sex +
    offset(log(DiveDuration)) +
    ar1(TimeFactor + 0 | BirdID),
  data = results_binned,
  family = gaussian()
)

anova(model15, model18)
```

Removing start mass also made the model worse, so we will keep it. I am going to try to reintroduce individual ID. I'm going to scale the predictors which will help hopefully.

```{r}
results_binned <- results_binned %>%
  mutate(
    StartMass_z = scale(StartMass),
    TotalEvents_z = scale(TotalEvents)
  )

model_ar1_only <- glmmTMB(
  ODBA_Total ~ TotalEvents_z * DiveType + StartMass_z + Sex + Year +
    offset(log(DiveDuration)) +
    ar1(TimeFactor + 0 | BirdID),  # Only AR1, no random intercept
  family = gaussian(),
  data = results_binned
)

anova(model15, model_ar1_only)
```

This model converged successfully. I used the AR(1) structure on the individual dive sequences (30 minute bin), which accounted for repeated measures within individuals.

```{r}
summary(model_ar1_only)
# Pearson residuals vs fitted values
plot(fitted(model_ar1_only), residuals(model_ar1_only, type = "pearson"),
     xlab = "Fitted values", ylab = "Pearson residuals",
     main = "Residuals vs Fitted")
abline(h = 0, col = "red")

# Q-Q plots
qqnorm(residuals(model_ar1_only, type = "pearson"))
qqline(residuals(model_ar1_only, type = "pearson"), col = "red")

# Residuals
hist(residuals(model_ar1_only, type = "pearson"),
     main = "Histogram of Pearson Residuals",
     xlab = "Residuals")
```

```{r}
library(dplyr)
library(emmeans)
library(ggplot2)
```

```r
# Step 1: Calculate average dive durations by dive type
avg_durations <- results_binned %>%
  group_by(DiveType) %>%
  summarise(avg_dur = mean(DiveDuration, na.rm = TRUE))

# Step 2: Get marginal means (ODBA per second)
emm <- emmeans(model_ar1_only, ~ DiveType, offset = TRUE, type = "response",
               condition = c(DiveDuration = 1))  # Offset set to 1 second

emm_df <- as.data.frame(emm)

# Step 3: Merge durations and calculate total ODBA per dive
emm_df <- left_join(emm_df, avg_durations, by = "DiveType") %>%
  mutate(ODBA_total = emmean * avg_dur,
         lower.CL_total = lower.CL * avg_dur,
         upper.CL_total = upper.CL * avg_dur)

# Step 4: Plot estimated total ODBA per dive
library(scales)

ggplot(emm_df, aes(x = DiveType, y = ODBA_total)) +
  geom_point(size = 3) +
  geom_errorbar(aes(ymin = lower.CL_total, ymax = upper.CL_total), width = 0.2) +
  scale_y_continuous(labels = label_scientific(digits = 2)) +  # scientific notation
  labs(
    title = "Estimated total ODBA per dive by dive type",
    x = "Dive Type",
    y = "Estimated total ODBA per dive"
  ) +
  theme_minimal(base_size = 14)
```

Summary info

```r
library(dplyr)

summary_by_bird <- results %>%
  group_by(BirdID) %>%
  summarise(
    Sex = unique(Sex),
    StartMass = unique(StartMass),
    n_dives = n(),
    mean_dive_duration = mean(DiveDuration, na.rm = TRUE),
    sd_dive_duration = sd(DiveDuration, na.rm = TRUE),
    min_dive_duration = min(DiveDuration, na.rm = TRUE),
    max_dive_duration = max(DiveDuration, na.rm = TRUE),
    mean_total_events = mean(TotalEvents, na.rm = TRUE),
    sd_total_events = sd(TotalEvents, na.rm = TRUE),
    dives_with_events = sum(TotalEvents > 0, na.rm = TRUE),
    prop_dives_with_events = mean(TotalEvents > 0, na.rm = TRUE),
    epipelagic = sum(DiveType == "Epipelagic", na.rm = TRUE),
    mesopelagic = sum(DiveType == "Mesopelagic", na.rm = TRUE),
    benthic = sum(DiveType == "Benthic", na.rm = TRUE)
  ) %>%
  mutate(
    dive_duration_summary = sprintf("%.1f ± %.1f", mean_dive_duration, sd_dive_duration),
    event_summary = sprintf("%.1f ± %.1f", mean_total_events, sd_total_events),
    range_duration = sprintf("%.1f – %.1f", min_dive_duration, max_dive_duration),
    prop_epipelagic = round(epipelagic / n_dives, 2),
    prop_mesopelagic = round(mesopelagic / n_dives, 2),
    prop_benthic = round(benthic / n_dives, 2)
  ) %>%
  dplyr::select(BirdID, Sex, StartMass, n_dives,
```

```
          dive_duration_summary, range_duration,
          event_summary, dives_with_events, prop_dives_with_events,
          epipelagic, mesopelagic, benthic,
          prop_epipelagic, prop_mesopelagic, prop_benthic)

# View the result
print(summary_by_bird, n = Inf)

write.csv(summary_by_bird, "/Users/taylorazizeh/Documents/research/active/emperor
penguins/results/R/summary_by_bird.csv", row.names = FALSE)
```