Modelling methods first draft

Survival of 4598 prawns was evaluated over 21 trials. Some prawns lost their band and could not be assigned to a treatment. The percentage of prawns that lost their bands ranged from 0%-13% and decreased as we gained experience banding. These prawns were removed from the data before fitting. Prawns for which carapace length was not measured (usually due to damage) were also not included.

(data paragraph move to end/middle instead?)

To understand the factors that influence post-release survival of *P. platyceros* in commercial fisheries while accounting for variation caused by GENERAL RANDOM EFFECT DESCRIPTION, we used generalized linear mixed-effects models (GLMMs) with binomial error structure to model probability of survival. In our model suite, we included combinations of three fixed effects: time out of water, air temperature, and length, and two-way interaction terms, resulting in 18 candidate models (Table 1). We did not include the three-way interaction term because it is difficult to interpret. All models included a random effect on the intercept to account for variation in survival caused by the trap a prawn was in; there were 123 levels. We expected survival to vary between traps because there were many differences between traps including location, time, and orientation on the ground. Model fitting was done with the lme4 and glmmTMB R packages (Laplace approximation).

To prioritise simplicity and interpretability, models were evaluated with Bayesian Information Criterion (BIC) (Table 1).

RESULTS

Similar BIC’s

Effect size

RE variance

We used logistic regression with a random effect for trap (123 levels) to model probability of survival for (N=4600) prawns. We used exhaustive enumeration and found there were 19 possible models for three predictor variables: treatment, air temperature, and carapace length (table 1). We fit all of these except for the three-way interaction, because of the difficulty in biological interpretation.

Some prawns lost their band and could not be assigned to a treatment. The percentage of prawns that lost their bands ranged from 0%-13% and decreased as we gained experience banding. These prawns were removed from the data before fitting. Prawns for which carapace length was not measured (usually due to damage) were also not included. Model fitting was done with the lme4 and glmmTMB R packages (Laplace approximation).

We expected the trap that a prawn was in for the release stage to affect the probability of survival. The orientation of the trap and location varied between traps, and probably effected survival. We expected variation in survival between traps that we wanted to account for but not capture in the model. The inclusion of the trap random effect accounts for this variation by assuming that for a given value of the linear predictor, the log odds of survival is normally distributed across traps.

More

Decisions made

We used length rather than stage because we had more data

Treatment as continuous

Salinity

Depth

Results

* Describe averaged model

Ideas:

Explain variables?

Decisions made

Not stage

FIGURES

Methods

Results

BCI Table

TTI predicted curves (five temps)

Oust the louse

Discussion

Figure: Unbanded per trial (proportion?)

Figure: Lost per trial (proportion?, per trial/treatment?)

Results

Table: BIC table

Logistic curve with Confidence Intervals (three temperatures)

RE curves (?)

(reflexes?)

Discussion

Unbanded and Lost

Table: Lost example Table

(reflexes?

DISCUSSION?

We banded the prawns in each treatment with a certain colour rostrum band before the experimental stage of each trial. Afterward the treatment a prawn underwent was identified by the colour of the rostrum band.

Prawns were therefore lost from treatments in two ways: they lost their band (unbanded) or were no longer in the trap (truly lost). If either of these methods of loss was biased based on a covariate such as survival (and therefore treatment, temperature, etc.) there would be an effect on the observed proportion that survived, and therefore on the effect size of the covariate.

As expected, the proportion of prawns that became unbanded appeared to be random across treatments.

The number lost from each treatment seemed somewhat biased toward longer treatments (and higher temperatures).

While we expect this effect is present, we believe it has a minor effect on results. Table 1 compares the true vs. observed proportion that survived for a hypothetical trap, under different biases in which prawns were lost.

We expected the trap that a prawn was in for the release stage to have a an effect on the probability of survival. The orientation of a given trap on the ground, the presence of predators (sea bugs) and varying environmental conditions varied between traps, and would have effected survival. For this reason, we included trap as a random effect.

Logistic Regression with a random effect of trial x trap

Why this?

Explain trial and trap and what it captures

Interpretation does it change anything

Why not more?

PER TRIAL ANALYSIS

library("here")

library("ggplot2")

setwd(here("data-clean"))

survival<-read.csv("2023-05-09\_prawn\_combined\_survival\_data.csv")

trial<-read.csv("2023-05-09\_prawn\_combined\_trial\_data.csv")

attach(survival)

trial<-trial[order(trial$trial\_number),]

unique(survival$treatment)

survival[which(survival$treatment=="T"),]

n\_trials<-length(unique(trial$trial\_number))

#set up empty vectors

trial\_number<-vector(mode="numeric", length=n\_trials)

lost\_prawnz<-vector(mode="numeric", length=n\_trials)

total\_treatments<-vector(mode="numeric", length=n\_trials)

unbanded<-vector(mode="numeric", length=n\_trials)

scavenged<-vector(mode="numeric", length=n\_trials)

dead<-vector(mode="numeric", length=n\_trials)

alive<-vector(mode="numeric", length=n\_trials)

max\_surv\_sum<-vector(mode="numeric", length=n\_trials)

min\_surv\_sum<-vector(mode="numeric", length=n\_trials)

pulled<-vector(mode="numeric", length=n\_trials)

remain<-vector(mode="numeric", length=n\_trials)

quarts<-vector(mode="numeric", length=n\_trials)

stage\_0\_per\_trial<-vector(mode="numeric", length=n\_trials)

stage\_1\_per\_trial<-vector(mode="numeric", length=n\_trials)

stage\_2\_per\_trial<-vector(mode="numeric", length=n\_trials)

stage\_3\_per\_trial<-vector(mode="numeric", length=n\_trials)

stage\_NA\_per\_trial<-vector(mode="numeric", length=n\_trials)

lost\_immediate<-vector(mode="numeric", length=n\_trials)

lost\_30<-vector(mode="numeric", length=n\_trials)

lost\_60<-vector(mode="numeric", length=n\_trials)

lost\_90<-vector(mode="numeric", length=n\_trials)

lost\_120<-vector(mode="numeric", length=n\_trials)

file\_names<-vector(mode="character", length=n\_trials)

file\_names\_1<-vector(mode="character", length=n\_trials)

alive\_0<-vector(mode="character", length=n\_trials)

alive\_30<-vector(mode="character", length=n\_trials)

alive\_60<-vector(mode="character", length=n\_trials)

alive\_90<-vector(mode="character", length=n\_trials)

alive\_120<-vector(mode="character", length=n\_trials)

remain\_0<-vector(mode="character", length=n\_trials)

remain\_30<-vector(mode="character", length=n\_trials)

remain\_60<-vector(mode="character", length=n\_trials)

remain\_90<-vector(mode="character", length=n\_trials)

remain\_120<-vector(mode="character", length=n\_trials)

for (i in 1:n\_trials){

#datasets of trial and survival for a trial number

df<-subset(trial, trial\_number==sort(trial$trial\_number)[i])

df1<-subset(survival, trial\_number==sort(trial$trial\_number)[i])

#sum of treatment numbers to ascertain total prawns in each trial

pulled[i]<-sum(c(df$immediate\_release\_number,df$X30min\_number,df$X1h\_number,df$X1h30min\_numner,df$X2h\_number), na.rm = TRUE)

#total prawns left after soak including dead scavenged and alive

#for each trial, as it changes with the for loop

remain[i]<-nrow(df1)

#lost is the difference for each trial

lost\_prawnz[i]<-pulled[i]-remain[i]

#quarts stores the 1st quartile for length of each trial

quarts[i]<-quantile(df1$length,probs = .25,na.rm = TRUE)

#Lost prawns in by treatment, calculated as the number in the treatment at the start minus the

#number of prawns with that colour band in the analysis stage. The difference is therefore truly

#lost or unbanded prawns per treatment, for each trial.

lost\_immediate[i]<-df$immediate\_release\_number-length(which(df1$treatment=="0"))

lost\_30[i]<-df$X30min\_number-length(which(df1$treatment=="30"))

lost\_60[i]<-df$X1h\_number-length(which(df1$treatment=="60"))

lost\_90[i]<-df$X1h30min\_numner-length(which(df1$treatment=="90"))-length(which(df1$treatment=="100"))

lost\_120[i]<-df$X2h\_number-length(which(df1$treatment=="120"))

# these vectors are filled with the maxima and minima of the sums of three

#survival values for each prawn for a trial. The sums should all be 1

#because the three are mutually exclusive. Used for checking data was

#entered correctly

max\_surv\_sum[i]<-max(df1$alive+df1$dead+df1$scavenged, na.rm=TRUE)

min\_surv\_sum[i]<-min(df1$alive+df1$dead+df1$scavenged, na.rm=TRUE)

#number of different treatments for each trial

total\_treatments[i]<-length(unique(df1$treatment))

#unbanded prawns are entered as treatment=NA so the sum of these, per trial

#is the number of unbanded per trial

unbanded[i]<-length(which(is.na(df1$treatment)))

#the number of scavenged,dead and alive prawns per trial

scavenged[i]<-sum(df1$scavenged,na.rm=TRUE)

dead[i]<-sum(df1$dead,na.rm=TRUE)

alive[i]<-sum(df1$alive,na.rm=TRUE)

remain\_0[i]<-sum(df1$treatment==0, na.rm=TRUE)

alive\_0[i]<-sum(df1[which(df1$treatment==0),]$alive,na.rm=TRUE)

remain\_30[i]<-sum(df1$treatment==30, na.rm=TRUE)

alive\_30[i]<-sum(df1[which(df1$treatment==30),]$alive,na.rm=TRUE)

remain\_60[i]<-sum(df1$treatment==60, na.rm=TRUE)

alive\_60[i]<-sum(df1[which(df1$treatment==60),]$alive,na.rm=TRUE)

remain\_90[i]<-sum(df1$treatment==90, na.rm=TRUE)

alive\_90[i]<-sum(df1[which(df1$treatment==90),]$alive,na.rm=TRUE)

remain\_120[i]<-sum(df1$treatment==120, na.rm=TRUE)

alive\_120[i]<-sum(df1[which(df1$treatment==120),]$alive,na.rm=TRUE)

#vectors of all the prawns lengths is created under the name length\_x

#where x is the trial number

setwd(here("figures"))

violin\_length<-data.frame(length=append(df1[which(df1$stage<=1),]$length,df1[which(df1$stage>1),]$length),sex=append(rep("Juv/Male",length(df1[which(df1$stage<=1),]$length)),rep("Trans/Female",length(df1[which(df1$stage>1),]$length))))

violin\_length$sex<-as.factor(violin\_length$sex)

ggplot(violin\_length, aes(x=sex, y=length)) + geom\_violin()+ylim(15,55)

#ggsave(paste(Sys.Date(), "trial",df$trial\_number, "violin.pdf", sep="\_"))

file\_names\_1[i]<-paste(Sys.Date(), "trial",df$trial\_number, "violin.pdf", sep="\_")

# setwd(here("figures"))

# pdf(paste(Sys.Date(), "trial",df$trial\_number, "hist.pdf", sep="\_"), width=7, height=7, pointsize=12)

#par(mfrow=c(6,2),mar=c(4,4,1,2), oma=c(0,0,4,0))

#hist(df1$length,main=paste("Trial",df$trial\_number,"Total"),xlab="Length")

#if (df$immediate\_release\_number>0){hist(subset(df1, treatment=="0")$length, main=paste("Trial",df$trial\_number,"Immediate"),xlab="Length")}

# if (df$X30min\_number>0){hist(subset(df1, treatment=="30")$length,main=paste("Trial",df$trial\_number,"30 min"),xlab="Length")}

# if (df$X1h\_number>0){hist(subset(df1, treatment=="60")$length,main=paste("Trial",df$trial\_number,"60 min"),xlab="Length")}

# if (df$X1h30min\_numner>0){

# if (nrow(df1[which(df1$treatment=="90"),])>0){hist(subset(df1, treatment=="90")$length,main=paste("Trial",df$trial\_number,"90 min"),xlab="Length")}

# if (nrow(df1[which(df1$treatment=="100"),])>0){hist(subset(df1, treatment=="100")$length,main=paste("Trial",df$trial\_number,"100 min"),xlab="Length")}

# }

#if ((df$X2h\_number>0)|is.na(df$X2h\_number)==FALSE){hist(subset(df1, treatment=="120")$length,main=paste("Trial",df$trial\_number,"120 min"),xlab="Length")}

#if (sum(is.na(df1$treatment))>0){hist(df1[is.na(df1$treatment),]$length,main=paste("Trial",df$trial\_number,"Unbanded"),xlab="Length")}

#plot(NULL, ylim=c(0,1),xlim=c(-5,125), main=paste("Trial",df$trial\_number,"Proportion of Treatment Lost"), xlab="Treatment Time", ylab="Proportion of Loss" )

# points(0,lost\_immediate[i]/(df$immediate\_release\_number))

# points(30,lost\_30[i]/(df$X30min\_number))

# points(60,lost\_60[i]/(df$X1h\_number))

# points(90,lost\_90[i]/(df$X1h30min\_numner))

# points(120,lost\_120[i]/(df$X2h\_number))

# barplot(c(sum(is.na(subset(df1,treatment=="0")$length)),sum(is.na(subset(df1,treatment=="30")$length)), sum(is.na(subset(df1,treatment=="60")$length)),sum(is.na(subset(df1,treatment=="90")$length)),sum(is.na(subset(df1,treatment=="120")$length)), sum(is.na(df1[which(is.na(df1$treatment)),]$length))), names=c("0","30","60","90","120","Unbanded"),xlab="Treatment", ylab="Length NA's", main=paste("Prawns without length data: Trial",df$trial\_number),ylim=c(0,20))

# barplot(c(sum(is.na(subset(df1,treatment=="0")$stage)),sum(is.na(subset(df1,treatment=="30")$stage)), sum(is.na(subset(df1,treatment=="60")$stage)),sum(is.na(subset(df1,treatment=="90")$stage)),sum(is.na(subset(df1,treatment=="120")$stage)), sum(is.na(df1[which(is.na(df1$treatment)),]$stage))), names=c("0","30","60","90","120","Unbanded"),xlab="Treatment", ylab="Stage NA's", main=paste("Prawns without stage data: Trial",df$trial\_number),ylim=c(0,20))

# barplot(c(df$immediate\_release\_number, df$X30min\_number,df$X1h\_number, df$X1h30min\_numner,df$X2h\_number), names=c("0","30","60","90","120"), ylim = c(0,100))

# dev.off()

# file\_names[i]<-paste(Sys.Date(), "trial",df$trial\_number, "hist.pdf", sep="\_")

#The number of prawns of each stage per trial

stage\_0\_per\_trial[i]<-nrow(subset(df1, stage==0))

stage\_1\_per\_trial[i]<-nrow(subset(df1, stage==1))

stage\_2\_per\_trial[i]<-nrow(subset(df1, stage==2))

stage\_3\_per\_trial[i]<-nrow(subset(df1, stage==3))

stage\_NA\_per\_trial[i]<-sum(is.na(df1$stage))

}

library("qpdf")

qpdf::pdf\_combine(input=file\_names,output = paste0(Sys.Date(),"\_combined\_lost\_summary.pdf"))

qpdf::pdf\_combine(input=file\_names\_1,output = paste0(Sys.Date(),"\_combined\_violins.pdf"))

trial$X2h\_number

quantile(survival[which(survival$trial\_number==6),]$length,probs = .25,na.rm = TRUE)

#Dataframe of per trial information

trial\_df<-data.frame(trial$trial\_number, total\_treatments, pulled, remain, lost\_prawnz, unbanded, scavenged, dead, alive, stage\_0\_per\_trial,stage\_1\_per\_trial,stage\_2\_per\_trial,stage\_3\_per\_trial)

#Creates a new column in the summary dataframe showing salinity. If haul temp

#was recorded, salinity takes that value. If not, haul tote temp is the value.

salinity<-trial$exp\_haul\_tote\_sal

salinity[is.na(salinity)]<-exp\_haul\_sal\_0m[is.na(salinity)]

trial\_df$salinity<-salinity

#trial\_df$temperature<-rowSums(cbind(trial$exp\_set\_temp\_0m,trial$exp\_haul\_temp\_0m), na.rm=TRUE)/(rep(2, 21)-(is.na(trial$exp\_haul\_temp\_0m)+is.na(trial$exp\_set\_temp\_0m)))

trial\_df$air\_temp<-trial$exp\_set\_temp\_air

quantile()

# The maxima and minima of the sums of the alive, dead and scavenged per trial

max\_surv\_sum

min\_surv\_sum

test\_df<-subset(survival, trial\_number==16)

which.min(test\_df$dead+test\_df$alive+test\_df$scavenged)

#bar graph unbanded for each trial

setwd(here("figures"))

png(paste(Sys.Date(), "unbanded\_bar.png", sep="\_"), width=480, height=480, units = "px", pointsize=12)

par(mfrow=c(1,1),mar=c(4,4,1,2), oma=c(0,0,4,0))

barplot(unbanded, xlab="Trial", ylab="Unbanded Prawns")

dev.off()

#boxplot of the length distribution for each trial

setwd(here("figures"))

png(paste(Sys.Date(), "length\_trial\_boxplot.png", sep="\_"), width=480, height=480, units = "px", pointsize=12)

par(mfrow=c(1,1),mar=c(4,4,1,2), oma=c(0,0,4,0))

boxplot(lengths\_1, lengths\_2, lengths\_3,lengths\_4, lengths\_5, lengths\_6,lengths\_7, lengths\_8, lengths\_9,lengths\_10, lengths\_11, lengths\_12, lengths\_13,lengths\_14, lengths\_15, lengths\_16,lengths\_17, lengths\_18, lengths\_19,lengths\_20,lengths\_21, names = sort(trial$trial\_number))

dev.off()

#Number of each stage per trial

setwd(here("figures"))

png(paste(Sys.Date(), "stage\_trial\_boxplot.png", sep="\_"), width=480, height=480, units = "px", pointsize=12)

par(mfrow=c(2,2),mar=c(4,4,1,2), oma=c(0,0,4,0))

plot(1:n\_trials, stage\_0\_per\_trial, xlab="Trial", ylab="Stage 0 Prawns")

plot(1:n\_trials, stage\_1\_per\_trial, xlab="Trial", ylab="Stage 1 Prawns")

plot(1:n\_trials, stage\_2\_per\_trial, xlab="Trial", ylab="Stage 2 Prawns")

plot(1:n\_trials, stage\_3\_per\_trial, xlab="Trial", ylab="Stage 3 Prawns")

dev.off()

#lost, dead, scavenged, alive prawns per trial

setwd(here("figures"))

png(paste(Sys.Date(), "lost\_prawns.png", sep="\_"), width=480, height=480, units = "px", pointsize=12)

par(mfrow=c(1,1),mar=c(4,4,1,2), oma=c(0,0,4,0))

barplot(lost\_prawnz, xlab="Trial", ylab="Lost Prawns")

dev.off()

trial\_repeated <- c(rep("1" , 3) , rep("2" , 3) , rep("3" , 3) , rep("4" , 3),rep("5" , 3) , rep("6" , 3) , rep("7" , 3) , rep("8" , 3),rep("9" , 3) , rep("10" , 3) ,rep("13" , 3) , rep("14" , 3) , rep("15" , 3) , rep("16" , 3),rep("17" , 3) , rep("18" , 3) , rep("19" , 3) , rep("20" , 3),rep("21" , 3) , rep("22" , 3) , rep("23" , 3))

condition <- rep(c("Alive" , "Dead" , "Scavenged") , n\_trials)

prawns<-vector(mode="numeric", length=n\_trials\*3)

for (i in 1:n\_trials){

prawns[3\*i-2]<-alive[i]

prawns[3\*i-1]<-dead[i]

prawns[3\*i]<-scavenged[i]

}

rep\_data<- data.frame(trial\_repeated,condition,prawns)

alive

dead

scavenged

setwd(here("figures"))

png(paste(Sys.Date(), "condition\_barplot.png", sep="\_"), width=480, height=480, units = "px", pointsize=12)

par(mfrow=c(1,1),mar=c(4,4,1,2), oma=c(0,0,4,0))

ggplot(rep\_data, aes(fill=condition, y=prawns, x=trial\_repeated)) +

geom\_bar(position="stack", stat="identity")+xlab("Trial")+ylab("Prawns")

dev.off()

#Percent lost, dead, scavenged, alive prawns per trial

#NOTE: dead alive and scavenged are shown as proportions of remaining

#(not including lost) while lost is shown as a proportion of the total

#at the start

trial\_df$percent\_lost<-(100\*lost\_prawnz/pulled)

trial\_df$percent\_dead<-100\*trial\_df$dead/trial\_df$remain

trial\_df$percent\_alive<-100\*trial\_df$alive/trial\_df$remain

trial\_df$percent\_scavenged<-100\*trial\_df$scavenged/trial\_df$remain

dead/remain+alive/remain+scavenged/remain

trial\_df$percent\_dead+trial\_df$percent\_alive+trial\_df$percent\_scavenged

#Plots of Percent survival vs Temperature

p1<-ggplot(data=trial\_df, aes(x=salinity, y=percent\_alive))+geom\_point()

p2<-ggplot(data=trial\_df, aes(x=temperature, y=percent\_alive))+geom\_point()

p1

p2

#Plot of lost per treatment over trial

setwd(here("figures"))

png(paste(Sys.Date(), "lost\_by\_treatment.png", sep="\_"), width=800, height=480, units = "px", pointsize=12)

par(mfrow=c(1,1),mar=c(4,4,1,2), oma=c(0,0,4,0))

plot(sort(trial$trial\_number), lost\_30,col='purple', xlab="Trial", ylab="Lost or Unbanded Prawns")

points(sort(trial$trial\_number),lost\_120, col='red')

points(sort(trial$trial\_number),lost\_60, col='blue')

points(sort(trial$trial\_number),lost\_90, col='green')

points(sort(trial$trial\_number),lost\_immediate)

legend(20, 20, legend=c("0 min", "30 min","60 min", "90 min","120 min"),col=c("black","purple", "blue","green", "red"),pch=1, cex=1)

dev.off()

lost\_prawnz-(lost\_immediate+lost\_30+lost\_60+lost\_90+lost\_120+unbanded)

#SHOULD BE ALL ZEROES

lost\_prawnz+dead+alive+scavenged-pulled

#SHOULD BE ALL 100

trial\_df$percent\_dead+trial\_df$percent\_alive+trial\_df$percent\_scavenged

#

setwd(here("figures"))

png(paste(Sys.Date(), "lost\_vs\_quartile.png", sep="\_"), width=800, height=480, units = "px", pointsize=12)

par(mfrow=c(1,1),mar=c(4,4,1,2), oma=c(0,0,4,0))

plot(quarts,lost\_prawnz, xlab="25th Percentile of Length", ylab="Lost Prawns")

dev.off()

setwd(here("figures"))

png(paste(Sys.Date(), "lost\_by\_treatment\_barplot.png", sep="\_"), width=480, height=480, units = "px", pointsize=12)

par(mfrow=c(1,1),mar=c(4,4,1,2), oma=c(0,0,4,0))

barplot(c(sum(lost\_immediate), sum(lost\_30),sum(lost\_60), sum(lost\_90),sum(lost\_120)), names=c("0","30","60","90","120"), ylab = "Lost Prawns", xlab="Treatment")

dev.off()

#NOTE: for graph below there are ~21x5 points. Adding together 5 from the same trial=1.

#Points are the proportion of prawns lost from the trial that were lost from that trialxtreatment

setwd(here("figures"))

png(paste(Sys.Date(), "lost\_by\_treatment\_and\_trial\_points.png", sep="\_"), width=480, height=480, units = "px", pointsize=12)

par(mfrow=c(1,1),mar=c(4,4,1,2), oma=c(0,0,4,0))

plot(NULL,xlim=c(-5,125),ylim=c(0,1), ylab="Proportion lost or Unbanded", xlab="Treatment Time", main="Proportion of each Treatment Lost")

points(rep(0,21),lost\_immediate/(trial$immediate\_release\_number))

points(rep(30,21),lost\_30/(trial$X30min\_number))

points(rep(60,21),lost\_60/(trial$X1h\_number))

points(rep(90,21),lost\_90/(trial$X1h30min\_numner))

points(rep(120,21),lost\_120/(trial$X2h\_number))

dev.off()

setwd(here("figures"))

png(paste(Sys.Date(), "lost\_percent\_by\_treatment\_barplots.png", sep="\_"), width=480, height=480, units = "px", pointsize=12)

par(mfrow=c(2,1),mar=c(4,4,1,2), oma=c(0,0,4,0))

barplot(c(sum(lost\_immediate)/(sum(lost\_prawnz)+sum(unbanded)), sum(lost\_30)/(sum(lost\_prawnz)+sum(unbanded)),sum(lost\_60)/(sum(lost\_prawnz)+sum(unbanded)), sum(lost\_90)/(sum(lost\_prawnz)+sum(unbanded)),sum(lost\_120)/(sum(lost\_prawnz)+sum(unbanded))),names=c("0","30","60","90","120"), main="Proportion of Lost by Treatment")

barplot(c(sum(lost\_immediate)/sum(trial$immediate\_release\_number), sum(lost\_30)/sum(trial$X30min\_number),sum(lost\_60)/sum(trial$X1h\_number), sum(lost\_90)/sum(trial$X1h30min\_numner),sum(lost\_120)/sum(trial$X2h\_number)),names=c("0","30","60","90","120"), main="Proportion of each Treatment Lost")

dev.off()

#MAKE BIG PDFs

sum(lost\_immediate)/(sum(lost\_prawnz)+sum(unbanded))+ sum(lost\_30)/(sum(lost\_prawnz)+sum(unbanded))+sum(lost\_60)/(sum(lost\_prawnz)+sum(unbanded))+ sum(lost\_90)/(sum(lost\_prawnz)+sum(unbanded))+sum(lost\_120)/(sum(lost\_prawnz)+sum(unbanded))

#Write Trial summary dataframe csv

setwd(here("data-clean"))

write.csv(trial\_df,"2023-05-17\_trial\_summary.csv")

plot(trial\_df$air\_temp,trial\_df$remain, ylim=c(0,350))

points(trial\_df$air\_temp, trial\_df$alive, col="red")

trial\_df$alive/trial\_df$remain

hist()

opar = par(oma = c(2,2,2,2)) # Large right margin for plot

barplot(c(sum(as.integer(remain\_0)),sum(as.integer(remain\_30)),sum(as.integer(remain\_60)),sum(as.integer(remain\_90)),sum(as.integer(remain\_120))),col=rgb(0,0,1,1/4), xlab="Time out of Water (minutes)",names.arg = c("0","30","60","90","120"), beside=TRUE)

barplot(c(sum(as.integer(alive\_0)),sum(as.integer(alive\_30)),sum(as.integer(alive\_60)),sum(as.integer(alive\_90)),sum(as.integer(alive\_120))),col=rgb(0,1,0,1/4),add=T)#, args.legend = list(legend=c("Total Prawns","Surviving Prawns"),col=c("lightblue","purple")))

opar = par(oma = c(0,0,0,0), mar = c(0,0,0,0), new = TRUE)

legend(x="right", legend = c("Total Prawns","Surviving Prawns"), fill=c(rgb(0,0,1,1/4),rgb(0,1,0.3,1/4)))

par(opar)

graphics.off()

barplot\_df<-t(matrix(c(c(sum(as.integer(alive\_0)),sum(as.integer(alive\_30)),sum(as.integer(alive\_60)),sum(as.integer(alive\_90)),sum(as.integer(alive\_120))),c(sum(as.integer(remain\_0))-sum(as.integer(alive\_0)),sum(as.integer(remain\_30))-sum(as.integer(alive\_30)),sum(as.integer(remain\_60))-sum(as.integer(alive\_60)),sum(as.integer(remain\_90))-sum(as.integer(alive\_90)),sum(as.integer(remain\_120))-sum(as.integer(alive\_120)))), nrow=5))

colnames(barplot\_df)<-c("0","30","60","90","120")

barplot(barplot\_df, beside=FALSE, col=c("lightblue", "pink"),legend=c("Surviving Prawns","Total Prawns"),names.arg=c("0","30","60","90","120"),xlab="Time out of Water (minutes)")

barplot\_df2<-matrix(c(barplot\_df[1,]/(barplot\_df[1,]+barplot\_df[2,])),nrow=1)

barplot\_df2<-rbind(barplot\_df2,c(1-c(barplot\_df[1,]/(barplot\_df[1,]+barplot\_df[2,]))))

colnames(barplot\_df2)<-c("0","30","60","90","120")

barplot(barplot\_df2, beside=FALSE, col=c("lightblue", "pink"),legend=c("Surviving Prawns","Total Prawns"),names.arg=c("0","30","60","90","120"),xlab="Time out of Water (minutes)")

REFLEXES

setwd("/Users/jacobhoutman/Documents/Git Hub/Prawnz-survival-experiment")

library("here")

setwd(here("data-clean"))

reflexes<-read.csv("2023-05-09\_prawn\_combined\_reflex\_data.csv")

survival<-read.csv("2023-05-09\_prawn\_combined\_survival\_data.csv")

trial<-read.csv("2023-05-09\_prawn\_combined\_trial\_data.csv")

#Ordering dataframe

reflexes<-reflexes[order(reflexes$trial\_number,reflexes$trap\_number),]

#New column for sum of reflexes

reflexes$total<-reflexes$abdomen\_turgor+reflexes$abdomen\_retraction+reflexes$leg\_movement+reflexes$leg\_retraction+reflexes$maxilliped\_movement+reflexes$maxilliped\_retraction+reflexes$antenna+reflexes$eye\_turgor+as.integer(reflexes$pleopods)+reflexes$mouth

#Violin plot sum of reflexes for each treatment

setwd(here("figures"))

violin\_reflex<-data.frame(total=c(reflexes[which(reflexes$treatment==0),]$total,reflexes[which(reflexes$treatment==30),]$total,reflexes[which(reflexes$treatment==60),]$total,reflexes[which(reflexes$treatment==90),]$total,reflexes[which(reflexes$treatment==120),]$total),treat=c(rep("0",nrow(reflexes[which(reflexes$treatment==0),])),rep("30",nrow(reflexes[which(reflexes$treatment==30),])),rep("60",nrow(reflexes[which(reflexes$treatment==60),])),rep("90",nrow(reflexes[which(reflexes$treatment==90),])),rep("120",nrow(reflexes[which(reflexes$treatment==120),]))))

violin\_reflex$treat<-as.factor(violin\_reflex$treat)

ggplot(violin\_reflex, aes(x=treat, y=total)) + geom\_violin()

boxplot(reflexes[which(reflexes$treatment==0),]$total,reflexes[which(reflexes$treatment==30),]$total,reflexes[which(reflexes$treatment==60),]$total,reflexes[which(reflexes$treatment==90),]$total,reflexes[which(reflexes$treatment==120),]$total)

graphics.off()

#Plot for each treatment

setwd(here("figures"))

pdf(paste(Sys.Date(), "reflex\_distribution.pdf", sep="\_"), width=7, height=7, pointsize=12)

par(mfrow=c(6,2),mar=c(4,4,1,2), oma=c(0,0,4,0))

for (i in 0:4){

treat\_ref<-reflexes[which(reflexes$treatment==i\*30),]$total

plot(NULL, xlim=c(0,10),ylim=c(0,100), main=paste("Reflex distribution treatment", i\*30))

points(0,sum(treat\_ref==0)\*100/length(treat\_ref==0))

points(1,sum(treat\_ref==1)\*100/length(treat\_ref==1))

points(2,sum(treat\_ref==2)\*100/length(treat\_ref==2))

points(3,sum(treat\_ref==3)\*100/length(treat\_ref==3))

points(4,sum(treat\_ref==4)\*100/length(treat\_ref==4))

points(5,sum(treat\_ref==5)\*100/length(treat\_ref==5))

points(6,sum(treat\_ref==6)\*100/length(treat\_ref==6))

points(7,sum(treat\_ref==7)\*100/length(treat\_ref==7))

points(8,sum(treat\_ref==8)\*100/length(treat\_ref==8))

points(9,sum(treat\_ref==9)\*100/length(treat\_ref==9))

points(10,sum(treat\_ref==10)\*100/length(treat\_ref==10))

}

dev.off()

quantile(x, seq(0.2,0.8,0.2))

plot(NULL, xlim=c(0,120),ylim=c(0,10), main="Reflex quantiles by treatment")

for (i in 0:4){

treat\_ref<-reflexes[which(reflexes$treatment==i\*30),]$total

points(rep(i\*30, 4), quantile(treat\_ref,seq(0.2,0.8,0.2)))

}

quantile(reflexes[which(reflexes$treatment==90),]$total,seq(0.2,0.8,0.2))

table(c(0,2),c(0,2))

ref\_mat<-matrix(data=NA,nrow=10,ncol=21)

trials<-unique(reflexes[order(reflexes$trial\_number),]$trial\_number)

for (i in 1:21){

for(j in 1:10){

ref\_mat[j,i]<-length(which(reflexes[which(reflexes$trial\_number==trials[i]),]$total==j))

}

}

ref\_mat[5:7,]

CONFIDENCE INTERVAL

library(ggplot2)

fit2a<-glm(p.total ~ species, family="poisson", data=FishData)

fit2.simple<-glm(p.total ~ 1, family="poisson", data=FishData)

summary(fit2a) #Yes, effect of species IS significant.

# LRT test for significance:

test.stat <- -2\*logLik(fit2.simple) - (-2\*logLik(fit2a))

p.val <- pchisq(test.stat, df=1, lower.tail=F)

# Avg number of sea lice

b2a<-summary(fit2a)$coefficients

est2a1<-c(b2a[1,1], b2a[1,1]+b2a[2,1])

varc<-b2a[1,2]^2

varp<-b2a[2,2]^2

covcp<-vcov(fit2a)[1,2]

SE.pink<-sqrt(varp+varc+2\*covcp)

SE<-c(b2a[1,2], SE.pink)

est2aL<-est2a1-1.96\*SE

est2aU<-est2a1+1.96\*SE

cbind(est2a1,est2aL,est2aU)

modavgPred(list(fit2a),newdata=data.frame(species=c("chum","pink")))

exp(cbind(est2a1,est2aL,est2aU))

x<-c(1:120)

b2a<-summary(model\_treat1)$coef

spec\_se<-function(x){

return(b2a[1,2]^2+(x^2)\*b2a[2,2]^2+2\*x\*vcov(model\_treat1)[1,2])

}

spec\_se(x)

pred\_0<-predict(model\_treat1, re.form=NA,newdata =data.frame(treatment=x))

plot(x,back\_trans(pred\_0),type = "l")

lines(x,back\_trans(pred\_0+spec\_se(x)),col="red")

lines(x,back\_trans(pred\_0-spec\_se(x)),col="blue")

par(mfrow=c(1,1))

plot(x,pred\_0,type = "l")

lines(x,(pred\_0+spec\_se(x)),col="red")

lines(x,back\_trans(pred\_0-spec\_se(x)),col="blue")

spec\_se<-function(x){

return(sqrt(b2a[1,2]^2+(x^2)\*b2a[2,2]^2+2\*x\*vcov(model\_treat1)[1,2]))

}

predltemp<-predict(model\_tt1,data.frame(treatment=c(1:120),temp=(rep(seq(-2,2),24))), interval='none',re.form=NA)

plot(back\_trans(pred0))

pred0<-predict(model\_tt1,data.frame(treatment=c(1:120),temp=(rep(0,120))),re.form=NA)

b2a<-summary(model\_tt1)$coefficients

spec\_se<-function(newdata){

return(sqrt(b2a[1,2]^2+(newdata[,1]^2)\*b2a[2,2]^2+(newdata[,2]^2)\*b2a[3,2]^2+2\*vcov(model\_tt1)[1,2]+2\*vcov(model\_tt1)[1,3]+2\*vcov(model\_tt1)[2,3]))

}

SE<-spec\_se(data.frame(treatment=c(1:120),temp=(rep(0,120))))

plot(pred0,type="l")

lines(SE+pred0, col="red")

lines(pred0-SE, col="blue")

plot(back\_trans(pred0),type="l")

lines(back\_trans(SE+pred0), col="red")

lines(back\_trans(pred0-SE), col="blue")

spec\_se<-function(newdata){

x1<-newdata[,1]

x2<-newdata[,2]

return(sqrt(b2a[1,2]^2+(newdata[,1]^2)\*b2a[2,2]^2+(newdata[,2]^2)\*b2a[3,2]^2+x1\*2\*vcov(model\_tt1)[1,2]+x2\*2\*vcov(model\_tt1)[1,3]+x1\*x2\*2\*vcov(model\_tt1)[2,3]))

}

SE<-spec\_se(data.frame(treatment=c(1:120),temp=(rep(0,120))))

plot(pred0,type="l")

lines(1.96\*SE+pred0, col="red")

lines(-1.96\*SE+pred0, col="blue")

plot(back\_trans(pred0),type="l")

lines(back\_trans(1.96\*SE+pred0), col="red")

lines(back\_trans(-1.96\*SE+pred0), col="blue")

library(AICcmodavg)

CI<-modavgPred(list(model\_tt1), newdata=data.frame(temp=rep(mu\_temp,120),treatment=c(1:120)))

CI1<-modavgPred(list(model\_tt1), newdata=data.frame(temp=rep(max\_temp,120),treatment=c(1:120)))

CI2<-modavgPred(list(model\_tt1), newdata=data.frame(temp=rep(min\_temp,120),treatment=c(1:120)))

plot(CI$mod.avg.pred,type="l")

lines(CI$upper.CL, col="red")

lines(CI$lower.CL, col="blue")

plot(CI1$mod.avg.pred,type="l")

lines(CI1$upper.CL, col="red")

lines(CI1$lower.CL, col="blue")

plot(CI2$mod.avg.pred,type="l")

lines(CI2$upper.CL, col="red")

lines(CI2$lower.CL, col="blue")

trial$exp\_set\_temp\_air

min\_temp<-min(trial$exp\_set\_temp\_air)

mu\_temp<-mean(trial$exp\_set\_temp\_air)

max\_temp<-max(trial$exp\_set\_temp\_air)

library(AICcmodavg)

CI<-modavgPred(list(model\_tti1), newdata=data.frame(temp=rep(mu\_temp,120),treatment=c(1:120)))

CI1<-modavgPred(list(model\_tti1), newdata=data.frame(temp=rep(max\_temp,120),treatment=c(1:120)))

CI2<-modavgPred(list(model\_tti1), newdata=data.frame(temp=rep(min\_temp,120),treatment=c(1:120)))

CI3<-modavgPred(list(model\_tti1), newdata=data.frame(temp=rep(mu\_temp,120),treatment=c(1:120)), type="link")

CI4<-modavgPred(list(model\_tti1), newdata=data.frame(temp=rep(max\_temp,120),treatment=c(1:120)),type="link")

CI5<-modavgPred(list(model\_tti1), newdata=data.frame(temp=rep(min\_temp,120),treatment=c(1:120)),type="link")

#FAKE

model\_df\_2$fake\_cov<-c(1,2,rep(3,nrow(model\_df\_2)-5),4,5,6)

model\_fake<-glmer(alive~fake\_cov+(1|trial\_trap),data=model\_df\_2,family=binomial)

CI6<-modavgPred(list(model\_fake), newdata=data.frame(fake\_cov=c(1:7)))

CI7<-modavgPred(list(model\_fake), newdata=data.frame(fake\_cov=c(1:7)),type="link")

plot(CI7$mod.avg.pred,type="l")

lines(CI7$lower.CL)

lines(CI7$upper.CL)

library(ggplot2)

colors<-c("Min temp"="black", 'Mean temp'="purple", "Max temp"="hotpink")

plot\_df1<-data.frame(x=1:120,min=CI5$mod.avg.pred,mid=CI3$mod.avg.pred,max=CI4$mod.avg.pred)

plot\_df<-data.frame(x=1:120,min=CI2$mod.avg.pred,mid=CI$mod.avg.pred,max=CI1$mod.avg.pred)

#plot2<-ggplot(plot\_df, aes(x=x))+xlim(c(0,120))+ylim(c(0,1))+geom\_line(aes(y=min, color="Min temp"), size = 1)+geom\_ribbon(aes(ymin=CI2$lower.CL, ymax=CI2$upper.CL), alpha=0.1, fill = "black", color = "black", linetype = "dotted")+geom\_line(aes(y=mid, color="Mean temp"), size = 1) +geom\_line(aes(y=max, color="Max temp"), size = 1) +geom\_ribbon(aes(ymin=CI1$lower.CL, ymax=CI1$upper.CL), alpha=0.1, fill = "hotpink", color = "black", linetype = "dotted")+geom\_ribbon(aes(ymin=CI$lower.CL, ymax=CI$upper.CL), alpha=0.1, fill = "pink", color = "black", linetype = "dotted")+labs(x="Treatment",y="Ps", color="Legend")+scale\_color\_manual(values = colors)

#plot1<-ggplot(plot\_df1, aes(x=x))+xlim(c(0,120))+ylim(c(-5,3))+geom\_line(aes(y=min, color="Min temp"), size = 1)+geom\_ribbon(aes(ymin=CI5$lower.CL, ymax=CI5$upper.CL), alpha=0.1, fill = "black", color = "black", linetype = "dotted")+geom\_line(aes(y=mid, colour="Mean temp"), size = 1) +geom\_line(aes(y=max, colour="Max temp"), size = 1) +geom\_ribbon(aes(ymin=CI4$lower.CL, ymax=CI4$upper.CL), alpha=0.1, fill = "hotpink", color = "black", linetype = "dotted")+geom\_ribbon(aes(ymin=CI3$lower.CL, ymax=CI3$upper.CL), alpha=0.1, fill = "pink", color = "black", linetype = "dotted")+labs(x="Treatment",y="Logit Ps", color="Legend")+scale\_color\_manual(values = colors)

library(ggplot2)

x0<-vector(length=21)

x30<-vector(length=21)

x60<-vector(length=21)

x90<-vector(length=21)

x120<-vector(length=21)

point\_df<-trial

point\_df$quart<-rep(0,21)

point\_df[which(point\_df$exp\_set\_temp\_air<=q25),]$quart<-"black"

point\_df[which(point\_df$exp\_set\_temp\_air>q25),]$quart<-"purple"

point\_df[which(point\_df$exp\_set\_temp\_air>=q75),]$quart<-"hotpink"

for (i in c(1:10)){

trial\_df<-subset(model\_df\_2,model\_df\_2$trial\_number==i)

x0[i]<-sum(trial\_df[which(trial\_df$treatment==0),]$alive)/nrow(trial\_df[which(trial\_df$treatment==0),])

x30[i]<-sum(trial\_df[which(trial\_df$treatment==30),]$alive)/nrow(trial\_df[which(trial\_df$treatment==30),])

x60[i]<-sum(trial\_df[which(trial\_df$treatment==60),]$alive)/nrow(trial\_df[which(trial\_df$treatment==60),])

x90[i]<-sum(trial\_df[which(trial\_df$treatment==90),]$alive)/nrow(trial\_df[which(trial\_df$treatment==90),])

x120[i]<-sum(trial\_df[which(trial\_df$treatment==120),]$alive)/nrow(trial\_df[which(trial\_df$treatment==120),])

}

for (i in c(13:23)){

trial\_df<-subset(model\_df\_2,model\_df\_2$trial\_number==i)

x0[i-2]<-sum(trial\_df[which(trial\_df$treatment==0),]$alive)/nrow(trial\_df[which(trial\_df$treatment==0),])

x30[i-2]<-sum(trial\_df[which(trial\_df$treatment==30),]$alive)/nrow(trial\_df[which(trial\_df$treatment==30),])

x60[i-2]<-sum(trial\_df[which(trial\_df$treatment==60),]$alive)/nrow(trial\_df[which(trial\_df$treatment==60),])

x90[i-2]<-sum(trial\_df[which(trial\_df$treatment==90),]$alive)/nrow(trial\_df[which(trial\_df$treatment==90),])

x120[i-2]<-sum(trial\_df[which(trial\_df$treatment==120),]$alive)/nrow(trial\_df[which(trial\_df$treatment==120),])

}

point\_df$x0<-x0

point\_df$x30<-x30

point\_df$x60<-x60

point\_df$x90<-x90

point\_df$x120<-x120

ggplot(point\_df, aes(x=treatment, y=x0))

sort(trial$exp\_set\_temp\_air)

q25<-quantile(model\_df\_2$temp,0.04)

q75<-quantile(model\_df\_2$temp,0.96)

sum(is.na(model\_df\_mean$alive))

model\_df\_cold<-subset(model\_df\_2, model\_df\_2$temp<=q25)

model\_df\_mean\_1<-subset(model\_df\_2, q25<model\_df\_2$temp)

model\_df\_mean<-subset(model\_df\_mean\_1, q75>model\_df\_mean\_1$temp)

model\_df\_hot<- subset(model\_df\_2, model\_df\_2$temp>=q75)

sort(unique(model\_df\_2$temp))

unique(model\_df\_cold$trial\_number)

nrow(model\_df\_cold)+nrow(model\_df\_mean)+nrow(model\_df\_hot)

ci\_df\_cold<-data.frame(treatment=c(0,30,60,90,120),proportion=c(sum(model\_df\_cold[which(model\_df\_cold$treatment==0),]$alive)/length(model\_df\_cold[which(model\_df\_cold$treatment==0),]$alive),sum(model\_df\_cold[which(model\_df\_cold$treatment==30),]$alive)/length(model\_df\_cold[which(model\_df\_cold$treatment==30),]$alive),sum(model\_df\_cold[which(model\_df\_cold$treatment==60),]$alive)/length(model\_df\_cold[which(model\_df\_cold$treatment==60),]$alive),sum(model\_df\_cold[which(model\_df\_cold$treatment==90),]$alive)/length(model\_df\_cold[which(model\_df\_cold$treatment==90),]$alive),sum(model\_df\_cold[which(model\_df\_cold$treatment==120),]$alive)/length(model\_df\_cold[which(model\_df\_cold$treatment==120),]$alive)))

ci\_df\_mean<-data.frame(treatment=c(0,30,60,90,100,120),proportion=c(sum(model\_df\_mean[which(model\_df\_mean$treatment==0),]$alive)/length(model\_df\_mean[which(model\_df\_mean$treatment==0),]$alive),sum(model\_df\_mean[which(model\_df\_mean$treatment==30),]$alive)/length(model\_df\_mean[which(model\_df\_mean$treatment==30),]$alive),sum(model\_df\_mean[which(model\_df\_mean$treatment==60),]$alive)/length(model\_df\_mean[which(model\_df\_mean$treatment==60),]$alive),sum(model\_df\_mean[which(model\_df\_mean$treatment==90),]$alive)/length(model\_df\_mean[which(model\_df\_mean$treatment==90),]$alive),sum(model\_df\_mean[which(model\_df\_mean$treatment==100),]$alive)/length(model\_df\_mean[which(model\_df\_mean$treatment==100),]$alive),sum(model\_df\_mean[which(model\_df\_mean$treatment==120),]$alive)/length(model\_df\_mean[which(model\_df\_mean$treatment==120),]$alive)))

ci\_df\_hot<-data.frame(treatment=c(0,30,60,90,120),proportion=c(sum(model\_df\_hot[which(model\_df\_hot$treatment==0),]$alive)/length(model\_df\_hot[which(model\_df\_hot$treatment==0),]$alive),sum(model\_df\_hot[which(model\_df\_hot$treatment==30),]$alive)/length(model\_df\_hot[which(model\_df\_hot$treatment==30),]$alive),sum(model\_df\_hot[which(model\_df\_hot$treatment==60),]$alive)/length(model\_df\_hot[which(model\_df\_hot$treatment==60),]$alive),sum(model\_df\_hot[which(model\_df\_hot$treatment==90),]$alive)/length(model\_df\_hot[which(model\_df\_hot$treatment==90),]$alive),sum(model\_df\_hot[which(model\_df\_hot$treatment==120),]$alive)/length(model\_df\_hot[which(model\_df\_hot$treatment==120),]$alive)))

plot2<-ggplot(plot\_df, aes(x=x))+geom\_line(aes(y=min, color="Min temp"), size = 1)+

geom\_ribbon(aes(ymin=CI2$lower.CL, ymax=CI2$upper.CL), alpha=0.5, fill = "black", color = "black", linetype = "dotted")+

geom\_line(aes(y=mid, color="Mean temp"), size = 1) +geom\_line(aes(y=max, color="Max temp"), size = 1) +

geom\_ribbon(aes(ymin=CI1$lower.CL, ymax=CI1$upper.CL), alpha=0.5, fill = "hotpink", color = "hotpink", linetype = "dotted")+

geom\_ribbon(aes(ymin=CI$lower.CL, ymax=CI$upper.CL), alpha=0.5, fill = "purple", color = "purple", linetype = "dotted")+

labs(x="Treatment",y="Ps", color="Legend")+scale\_color\_manual(values = colors)+

theme(panel.grid.major.y = element\_line(color="grey"),panel.background = element\_rect(fill = "white", colour = "grey50"))+

geom\_jitter(data=model\_df\_cold,aes(x=treatment,y=alive), size=0.2, height = 0.03,width=5,color='black', alpha=0.3)+

geom\_jitter(data=model\_df\_mean,aes(x=treatment,y=alive), size=0.2, height = 0.03,width=5,color='purple', alpha=0.2)+

geom\_jitter(data=model\_df\_hot,aes(x=treatment,y=alive), size=0.2, height = 0.03,width=5,color='hotpink', alpha=0.2)

plot3<-ggplot(plot\_df, aes(x=x))+geom\_line(aes(y=min, color="Min temp"), size = 1)+

geom\_ribbon(aes(ymin=CI2$lower.CL, ymax=CI2$upper.CL), alpha=0.5, fill = "black", color = "black", linetype = "dotted")+

geom\_line(aes(y=mid, color="Mean temp"), size = 1) +geom\_line(aes(y=max, color="Max temp"), size = 1) +

geom\_ribbon(aes(ymin=CI1$lower.CL, ymax=CI1$upper.CL), alpha=0.5, fill = "hotpink", color = "hotpink", linetype = "dotted")+

geom\_ribbon(aes(ymin=CI$lower.CL, ymax=CI$upper.CL), alpha=0.5, fill = "purple", color = "purple", linetype = "dotted")+

labs(x="Treatment",y="Ps", color="Legend")+scale\_color\_manual(values = colors)+

theme(panel.grid.major.y = element\_line(color="grey"),panel.background = element\_rect(fill = "white", colour = "grey50"))+

geom\_point(data=ci\_df\_cold, aes(x=treatment,y=proportion),color="black")+geom\_point(data=ci\_df\_mean, aes(x=treatment,y=proportion),color="purple")+

geom\_point(data=ci\_df\_hot, aes(x=treatment,y=proportion),color="hotpink")

median(trial$exp\_set\_temp\_air)

plot1<-ggplot(plot\_df1, aes(x=x))+xlim(c(5,120))+ylim(c(-7,3))+geom\_line(aes(y=min, color="Min temp"), size = 1)+

geom\_ribbon(aes(ymin=CI5$lower.CL, ymax=CI5$upper.CL), alpha=0.3, fill = "black", color = "black", linetype = "dotted")+

geom\_line(aes(y=mid, colour="Mean temp"), size = 1) +geom\_line(aes(y=max, colour="Max temp"), size = 1) +

geom\_ribbon(aes(ymin=CI4$lower.CL, ymax=CI4$upper.CL), alpha=0.3, fill = "hotpink", color = "black", linetype = "dotted")+

geom\_ribbon(aes(ymin=CI3$lower.CL, ymax=CI3$upper.CL), alpha=0.3, fill = "purple", color = "black", linetype = "dotted")+

labs(x="Treatment",y="Logit Ps", color="Legend")+scale\_color\_manual(values = colors)+

theme(panel.grid = element\_line(color="grey"), panel.border=element\_rect(fill=NA),panel.background = element\_rect(fill = "white", colour = "grey50"))+

geom\_jitter(data=model\_df\_2,aes(x=treatment,y=trans(alive)), size=0.2, height = 0.03,width=3)

point\_df$exp\_set\_temp\_air

plot4<-ggplot(plot\_df, aes(x=x))+geom\_line(aes(y=min, color="Min temp"), size = 1)+

geom\_ribbon(aes(ymin=CI2$lower.CL, ymax=CI2$upper.CL), alpha=0.5, fill = "black", color = "black", linetype = "dotted")+

geom\_line(aes(y=mid, color="Mean temp"), size = 1) +geom\_line(aes(y=max, color="Max temp"), size = 1) +

geom\_ribbon(aes(ymin=CI1$lower.CL, ymax=CI1$upper.CL), alpha=0.5, fill = "hotpink", color = "hotpink", linetype = "dotted")+

geom\_ribbon(aes(ymin=CI$lower.CL, ymax=CI$upper.CL), alpha=0.5, fill = "purple", color = "purple", linetype = "dotted")+

labs(x="Treatment",y="Ps", color="Legend")+scale\_color\_manual(values = colors)+

theme(panel.grid.major.y = element\_line(color="grey"),panel.background = element\_rect(fill = "white", colour = "grey50"))

ggplot(point\_df, aes(x=0:120, y=x0))+geom\_point(point\_df,aes(x=rep(0,21),colour = cut(x0, c(-Inf, q25, q75, Inf))),

size = 5) +scale\_color\_manual(name = "x0", values = c("(-Inf,12.8]" = "black",

"(12.8,17.8]" = "purple",

"(17.8, Inf]" = "hotpink"),

labels = c("<= 12.8", "12.8 < qsec <= 17.8", "> 17.8"))

library(here)

setwd(here("figures"))

png(paste(Sys.Date(), "tti\_survival\_curve\_jitter.png", sep="\_"), width=480, height=480, units = "px", pointsize=12)

par(mfrow=c(2,1),mar=c(4,4,1,2), oma=c(0,0,4,0))

plot2

dev.off()

setwd(here("figures"))

png(paste(Sys.Date(), "tti\_survival\_curve\_average.png", sep="\_"), width=480, height=480, units = "px", pointsize=12)

par(mfrow=c(2,1),mar=c(4,4,1,2), oma=c(0,0,4,0))

plot3

dev.off()

setwd(here("figures"))

png(paste(Sys.Date(), "tti\_survival\_lines.png", sep="\_"), width=480, height=480, units = "px", pointsize=12)

par(mfrow=c(2,1),mar=c(4,4,1,2), oma=c(0,0,4,0))

plot1

dev.off()

library(boot)

b\_par<-bootMer(x=model\_tti1,FUN=fixef,nsim=200)

boot.ci(b\_par,type="perc",index=1)

boot(model\_tti1, R=5000)

par(mfrow = c(2,2))

plot(model\_tti1)

ggplot(data.frame(eta=predict(model\_tti1,type="link"),pearson=residuals(model\_tti1,type="pearson")),

aes(x=eta,y=pearson)) +

geom\_point() +

theme\_bw()

ggplot(data.frame(x1=model\_df\_2$treatment,pearson=residuals(model\_tti1,type="pearson")),

aes(x=x1,y=pearson)) +

geom\_point() +

theme\_bw()

ggplot(data.frame(x2=model\_df\_2$temp,pearson=residuals(model\_tti1,type="pearson")),

aes(x=x2,y=pearson)) +

geom\_point() +

theme\_bw()

means <- aggregate(model\_df\_2[,c("treatment","temp")],by=list(model\_df\_2$trial\_trap),FUN=mean)

lmcoefs <- summary(lm(alive ~ treatment + temp + trial\_trap, data=model\_df\_2))$coefficients[,"Estimate"]

means$effects <- c(0,0,lmcoefs[substr(names(lmcoefs),1,3) == "tri"])

means$effects <- means$effects - mean(means$effects)

cor(means[,c("treatment","temp","effects")])

ggplot(means, aes(x=treatment,y=effects)) +

geom\_point() +

theme\_bw()

ggplot(means, aes(x=temp,y=effects)) +

geom\_point() +

theme\_bw()

means$treatment

qqnorm(residuals(model\_length2))

min\_temp<-min(model\_df\_2$temp)

nrow(newdata1)

newdata1<-expand.grid(temp=c(min\_temp,mean(model\_df\_2$temp),max\_temp),treatment=c(0,15,30,45,60,75,90,105,120), trial\_trap=levels(model\_df\_2$trial\_trap))

newdata1$logit\_ps <- predict(model\_tti1, newdata = newdata1)

newdata1$ps<-back\_trans(newdata1$logit\_ps)

newdata1[order(newdata1$treatment,newdata1$temp),]

plot(newdata1$treatment,newdata1$ps)

length(levels(newdata1$trial\_trap))

graphics.off()

setwd(here("figures"))

png(paste(Sys.Date(), "min\_temp\_re.png", sep="\_"), width=480, height=480, units = "px", pointsize=12)

par(mfrow=c(1,1),mar=c(4,4,1,2), oma=c(0,0,4,0))

ggplot(newdata1[which(newdata1$temp==min\_temp),],aes(x=treatment, y=ps, col=trial\_trap))+geom\_line()#+theme(legend.position = "none")

dev.off()

setwd(here("figures"))

png(paste(Sys.Date(), "mean\_temp\_re.png", sep="\_"), width=480, height=480, units = "px", pointsize=12)

par(mfrow=c(2,2),mar=c(4,4,1,2), oma=c(0,0,4,0))

ggplot(newdata1[which(newdata1$temp==mean(model\_df\_2$temp)),],aes(x=treatment, y=ps, col=trial\_trap))+geom\_line()+theme(legend.position = "none")

dev.off()

setwd(here("figures"))

png(paste(Sys.Date(), "max\_temp\_re.png", sep="\_"), width=480, height=480, units = "px", pointsize=12)

par(mfrow=c(2,2),mar=c(4,4,1,2), oma=c(0,0,4,0))

ggplot(newdata1[which(newdata1$temp==max\_temp),],aes(x=treatment, y=ps, col=trial\_trap))+geom\_line()+theme(legend.position = "none")

dev.off()

CI<-modavgPred(list(model\_tti1), newdata=data.frame(temp=rep(10,120),treatment=c(1:120)))

CI1<-modavgPred(list(model\_tti1), newdata=data.frame(temp=rep(14,120),treatment=c(1:120)))

CI2<-modavgPred(list(model\_tti1), newdata=data.frame(temp=rep(18,120),treatment=c(1:120)))

CI3<-modavgPred(list(model\_tti1), newdata=data.frame(temp=rep(22,120),treatment=c(1:120)))

CI4<-modavgPred(list(model\_tti1), newdata=data.frame(temp=rep(26,120),treatment=c(1:120)))

CI5<-modavgPred(list(model\_tti1), newdata=data.frame(temp=rep(30,120),treatment=c(1:120)))

colors<-c("10"="black","14"="blue" ,'18'="purple", "22"="hotpink","26"="red","30"="green")

plot\_df<-data.frame(x=1:120,min=CI$mod.avg.pred,mid1=CI1$mod.avg.pred,mid2=CI2$mod.avg.pred,max3=CI3$mod.avg.pred,max4=CI4$mod.avg.pred,max5=CI5$mod.avg.pred)

plot5<-ggplot(plot\_df, aes(x=x))+geom\_line(aes(y=min, color="10"), size = 1)+ geom\_line(aes(y=mid1, color="14"), size = 1) +

geom\_line(aes(y=mid2, color="18"), size = 1) +geom\_line(aes(y=max3, color="22"), size = 1) +geom\_line(aes(y=max4, color="26"), size = 1)+geom\_line(aes(y=max5, color="30"), size = 1)+

geom\_ribbon(aes(ymin=CI$lower.CL, ymax=CI$upper.CL), alpha=0.5, fill = "black", color = "black", linetype = "dotted")+

geom\_ribbon(aes(ymin=CI1$lower.CL, ymax=CI1$upper.CL), alpha=0.5, fill = "blue", color = "blue", linetype = "dotted")+

geom\_ribbon(aes(ymin=CI2$lower.CL, ymax=CI2$upper.CL), alpha=0.5, fill = "purple", color = "purple", linetype = "dotted")+

geom\_ribbon(aes(ymin=CI3$lower.CL, ymax=CI3$upper.CL), alpha=0.5, fill = "hotpink", color = "hotpink", linetype = "dotted")+

geom\_ribbon(aes(ymin=CI4$lower.CL, ymax=CI4$upper.CL), alpha=0.5, fill = "red", color = "red", linetype = "dotted")+

geom\_ribbon(aes(ymin=CI5$lower.CL, ymax=CI5$upper.CL), alpha=0.5, fill = "green", color = "green", linetype = "dotted")+

labs(x="Treatment",y="Ps", color="Legend")+scale\_color\_manual(values = colors)+

theme(panel.grid.major.y = element\_line(color="grey"),panel.background = element\_rect(fill = "white", colour = "grey50"))

plot(model\_tti1,residuals(.) ~log(fitted(.)))

ggplot(fortify(model\_tt1),

aes(x=.fitted,y=sqrt(abs(.scresid))))+geom\_point()+

geom\_smooth(colour="red",alpha=0.3)

dd <- lattice::dotplot(ranef(model\_tti1,condVar=TRUE))

do.call(gridExtra::grid.arrange,c(dd,list(nrow=1)))

par(mfrow=c(1,1))

ci\_best<-confint(profile(model\_tti2))

plot(model\_tti1,ylim=c(-3,3),type=c("p","smooth"))

install.packages('report')

report::report(model\_tti2)

lme4::qqmath(ranef(model\_tti1,condVar=TRUE))

CI4

library(here)

setwd(here("data-clean"))

install.packages("nem")

#Read in dataframe

reflexes<-read.csv("2023-05-09\_prawn\_combined\_reflex\_data.csv")

survival<-read.csv("2023-05-09\_prawn\_combined\_survival\_data.csv")

trial<-read.csv("2023-05-09\_prawn\_combined\_trial\_data.csv")

#Order trial dataframe

trial<-trial[order(trial$trial\_number),]

temp<-trial$exp\_set\_temp\_air

#

model\_df<-survival[(is.na(survival$treatment)==FALSE),]

model\_df$treatment<-as.integer(model\_df$treatment)

model\_df<-model\_df[order(model\_df$trial\_number,model\_df$prawn\_id),]

model\_df$trial\_trap<-paste(model\_df$trial\_number,"-",model\_df$trap\_number)

model\_df\_1<-model\_df[c(-564,-1255),]

model\_df\_1$temp<-rep(0, nrow(model\_df\_1))

n\_trials<-nrow(trial)

for(i in 1:n\_trials){

if(i<11){

model\_df\_1[which(model\_df\_1$trial\_number==i),]$temp<-rep(temp[i],length(which(model\_df\_1$trial\_number==i)))

}

if(i>=11){

model\_df\_1[which(model\_df\_1$trial\_number==i+2),]$temp<-rep(temp[i],length(which(model\_df\_1$trial\_number==i+2)))

}

}

unique(model\_df\_1$temp)

temp

model\_df\_1$trial\_trap<-as.factor(model\_df\_1$trial\_trap)

model\_df\_1$length<-round(model\_df\_1$length/0.5)\*0.5

mean(model\_df\_1$length, na.rm=TRUE)

install.packages("glmmTMB")

install.packages("lme4")

library("glmmTMB")

library("lme4")

model\_df\_1

model\_df\_2<-model\_df\_1[is.na(model\_df\_1$length)==FALSE,]

#Intercept only model

model\_null1<-lme4::glmer(alive~(1|trial\_trap),data=model\_df\_2,family=binomial)

model\_null2<-glmmTMB::glmmTMB(alive~(1|trial\_trap),data=model\_df\_2,family=binomial)

#treatment model

model\_treat1<-glmer(alive~treatment+(1|trial\_trap),data=model\_df\_2,family=binomial)

model\_treat2<-glmmTMB(alive~treatment+(1|trial\_trap),data=model\_df\_2,family=binomial)

#temp model

model\_temp1<-glmer(alive~temp+(1|trial\_trap),data=model\_df\_2,family=binomial)

model\_temp2<-glmmTMB(alive~temp+(1|trial\_trap),data=model\_df\_2,family=binomial)

#length model

model\_length1<-glmer(alive~length+(1|trial\_trap),data=model\_df\_2,family=binomial)

model\_length2<-glmmTMB(alive~length+(1|trial\_trap),data=model\_df\_2,family=binomial)

#treatment and temp model

model\_tt1<-glmer(alive~treatment+temp+(1|trial\_trap),data=model\_df\_2,family=binomial)

model\_tt2<-glmmTMB(alive~treatment+temp+(1|trial\_trap),data=model\_df\_2,family=binomial)

#temp and length model

model\_templ1<-glmer(alive~length+temp+(1|trial\_trap),data=model\_df\_2,family=binomial)

model\_templ2<-glmmTMB(alive~length+temp+(1|trial\_trap),data=model\_df\_2,family=binomial)

#treatment and length model

model\_treatl1<-glmer(alive~treatment+length+(1|trial\_trap),data=model\_df\_2,family=binomial)

model\_treatl2<-glmmTMB(alive~treatment+length+(1|trial\_trap),data=model\_df\_2,family=binomial)

#treatment and temp and interaction model

model\_tti1<-glmer(alive~treatment+temp+treatment\*temp+(1|trial\_trap),data=model\_df\_2,family=binomial)

model\_tti2<-glmmTMB(alive~treatment+temp+treatment\*temp+(1|trial\_trap),data=model\_df\_2,family=binomial)

#treatment and temp and length model

model\_ttl1<-glmer(alive~treatment+temp+length+(1|trial\_trap),data=model\_df\_2,family=binomial)

model\_ttl2<-glmmTMB(alive~treatment+temp+length+(1|trial\_trap),data=model\_df\_2,family=binomial)

#treatment and temp and length and length interactions model

model\_big1<-glmer(alive~treatment+temp+length+temp\*length+treatment\*length+(1|trial\_trap),data=model\_df\_2,family=binomial)

model\_big2<-glmmTMB(alive~treatment+temp+length+temp\*length+treatment\*length+(1|trial\_trap),data=model\_df\_2,family=binomial)

#ANALYSIS AND PLOTTING

back\_trans<-function(x){

return(exp(x)/(1+exp(x)))

}

trans<-function(x){

return(log(x/(1-x)))

}

back\_trans\_1<-function(x){

return(exp(0.3-0.14\*x)/(1+exp(0.3-0.14\*x)))

}

par(mfrow=c(2,2))

plot(model\_tti1)

pred0<-predict(model\_treatl1,newdata=data.frame(treatment=model.matrix(model\_treatl1)[,2],length=model.matrix(model\_treatl1)[,3]), re.form=NA)

plot(model.matrix(model\_treatl1)[,],back\_trans(pred0))

pframe<-model.frame(model\_tti1)

model.matrix(formula(model\_tti1,fixed.only=TRUE)[-2],pframe)

pframe <- data.frame(trial\_trap=factor(levels(model\_df\_2$trial\_trap), levels=levels(model\_df\_2$trial\_trap)))

cpred1 <- predict(model\_length1,re.form=NA,newdata=pframe,type="response")

easyPredCI <- function(model,newdata=NULL,alpha=0.05) {

## baseline prediction, on the linear predictor (logit) scale:

pred0 <- predict(model,re.form=NA,newdata=newdata)

## fixed-effects model matrix for new data

X <- model.matrix(formula(model,fixed.only=TRUE)[-2],newdata)

beta <- fixef(model) ## fixed-effects coefficients

V <- vcov(model) ## variance-covariance matrix of beta

pred.se <- sqrt(diag(X %\*% V %\*% t(X))) ## std errors of predictions

## inverse-link function

linkinv <- family(model)$linkinv

## construct 95% Normal CIs on the link scale and

## transform back to the response (probability) scale:

crit <- -qnorm(alpha/2)

linkinv(cbind(conf.low=pred0-crit\*pred.se,

conf.high=pred0+crit\*pred.se))

}

cpred1.CI <- easyPredCI(cmod\_lme4\_L,pframe)

mfrow(2,2)

plot(model\_tti1, type=c(1,2,3,4))

g\_pframe <- cbind(expand.grid(year=2004:2006,prev=0:80),Area=1)

g\_pred <- predict(gmod\_lme4\_L,newdata=g\_pframe,re.form=NA,

type="response")

g\_predCI <- easyPredCI(gmod\_lme4\_L,newdata=g\_pframe)

set.seed(101)

g\_bb <- bootMer(gmod\_lme4\_L,FUN=function(x),predict(x,re.form=NA,newdata=g\_pframe,type="response"),nsim=400)

p1 <- plot(model\_tti1,id=0.05,idLabels=~.obs)

p2 <- plot(model\_tti1,ylim=c(-1.5,1),type=c("p","smooth"))

predict(model\_length1,re.form=NA,newdata=model.frame(model\_length1))

easyPredCI <- function(model,newdata=NULL,alpha=0.05) {

## baseline prediction, on the linear predictor (logit) scale:

pred0 <- predict(model,re.form=NA,newdata=newdata)

## fixed-effects model matrix for new data

X <- model.matrix(formula(model,fixed.only=TRUE)[-2],model.frame(model))

beta <- fixef(model) ## fixed-effects coefficients

V <- vcov(model) ## variance-covariance matrix of beta

pred.se <- sqrt(diag(X %\*% V %\*% t(X))) ## std errors of predictions

## inverse-link function

linkinv <- family(model)$linkinv

## construct 95% Normal CIs on the link scale and

## transform back to the response (probability) scale:

crit <- -qnorm(alpha/2)

linkinv(cbind(conf.low=pred0-crit\*pred.se,conf.high=pred0+crit\*pred.se))

}

plot(sort(model\_df\_2$length),back\_trans(summary(model\_length1)$coef[1]+summary(model\_length1)$coef[2]\*sort(model\_df\_2$length)),ylim=c(0,1))

lines(model\_df\_2$length,easyPredCI(model\_length1)[,1])

lines(model\_df\_2$length,easyPredCI(model\_length1)[,2])

plot(sort(model\_df\_2$treatment),back\_trans(summary(model\_treat1)$coef[1]+summary(model\_treat1)$coef[2]\*sort(model\_df\_2$treatment)),ylim=c(0,1))

points(model\_df\_2$treatment,easyPredCI(model\_treat1)[,1], col="red")

points(model\_df\_2$treatment,easyPredCI(model\_treat1)[,2], col="blue")

plot(sort(model\_df\_2$treatment),back\_trans(summary(model\_tti1)$coef[1]+summary(model\_tti1)$coef[2]\*sort(model\_df\_2$treatment)),ylim=c(0,1))

points(model\_df\_2$treatment,easyPredCI(model\_treat1)[,1], col="red")

points(model\_df\_2$treatment,easyPredCI(model\_treat1)[,2], col="blue")

plot(model\_df\_2$treatment, model\_df\_2$alive)

points(sort(model\_df\_2$length),back\_trans(summary(model\_length1)$coef[1]+summary(model\_length1)$coef[2]\*sort(model\_df\_2$length)))

surv\_sub<-subset(sdata\_total,sdata\_total$trial\_number==12)

sum(surv\_sub$alive, na.rm=TRUE)/nrow(surv\_sub)

BIC.names

fixef(model\_tt2)

install.packages("MuMIn") # needed for BIC weights

# Get BIC values for each mixed-effects model

BIC.values <- BIC(model\_null2, model\_treat2, model\_temp2, model\_length2, model\_tt2,model\_templ2,model\_treatl2,model\_tti2, model\_ttl2,model\_big2)

# Shorthand names for models (in the same order as in BIC.values

BIC.names <- c("Null", "Treatment", "Temperature", "Length","Treatment + Temperature","Temperature + Length","Treatment + Length", "Treatment + Temperature + Treatment x Temperature","Treatment + Temperature + Length","Treatment + Temperature + Length + Treatment x Length + Temperature x Length")

# Order the BIC and model name vectors by ascending order of the AIC vector elements

BIC.order <- order(BIC.values$BIC)

BIC.names <- BIC.names[BIC.order]

BIC.values <- BIC.values$BIC[BIC.order]

# Calculate BIC weights

BIC.weights <- MuMIn::Weights(BIC.values)

# Calculate BIC cumulative weights

BIC.cumul.weights <- numeric(length(BIC.weights))

BIC.cumul.weights[1] <- BIC.weights[1]

for (i in 2:length(BIC.weights)) {

BIC.cumul.weights[i] <- BIC.cumul.weights[i-1] + BIC.weights[i]

}

# Calculate maximum log likelihood values and re-order

BIC.logLik <- c(logLik(model\_null2), logLik(model\_treat2), logLik(model\_temp2), logLik(model\_length2), logLik(model\_tt2),logLik(model\_templ2),logLik(model\_treatl2),logLik(model\_tti2),logLik(model\_ttl2),logLik(model\_big2))

BIC.logLik <- BIC.logLik[BIC.order]

# Create delta BIC function and calculate for each model relative to top model

deltaFcn = function(listBIC) {

deltas = -(min(listBIC) - listBIC)

}

BIC.deltas = deltaFcn(BIC.values)

# Create BIC table

BIC.table <- data.frame(

model.name = BIC.names, BIC=BIC.values,

deltaBIC = round(BIC.deltas,2),

weight = round(BIC.weights,3),

cumul.weight = round(BIC.cumul.weights,3),

logLik = round(BIC.logLik,2))

formula(model\_big1)

# Save BIC table

setwd(here("figures"))

write.csv(BIC.table, paste(Sys.Date(),"Prawn\_Survival\_BIC\_table.csv"))