Untitled

# Clear R’s brain

rm(list=ls())

setwd(“C:/Users/phipp/Desktop/MyFirstAnalysis”) mydata<-read.csv(“finalcourt.csv”) mydata #Calling all libraries

names(mydata)

length(mydataStage)

library(tidyverse) library(lmerTest)

## Parsed with column specification:

cols(ID = col\_integer(), Pop = col\_character(), Family = col\_integer(), DateBorn = col\_character(), Sex = col\_character(), Length = col\_double(), Stage = col\_character(), Date = col\_character(), Coder = col\_character(), Chase = col\_integer(), ForcedCop = col\_integer(), G\_Swing = col\_integer(), Sigmoid = col\_integer(), Suc\_SigCop = col\_integer(), Ignore = col\_integer(), Lunge = col\_integer(), Nip = col\_integer(), Suc\_ForcCop = col\_integer(), Copulated = col\_character())

data %>% mutate(Pop = as.factor(Pop)) %>% mutate(Stage = as.factor(Stage)) %>% mutate(Family = as.factor(Family)) %>% mutate(ID = as.factor(ID)) -> data

# ggplot to test aggression correlations

library(tidyverse)

court <- read\_csv(“courtForNathan3.csv”)

court %>% mutate(Pop = as.factor(Pop)) %>% mutate(Stage = as.factor(Stage)) %>% mutate(Family = as.factor(Family)) %>% mutate(ID = as.factor(ID)) -> court

court

# Sigmoids by Population and Stage accounting for length

lmm.Sig <- lmer(Sigmoid ~ Pop\*Stage + Length + (1|Family/ID), data=court) lmm.Sig

anova(lmm.Sig)

ranova(lmm.Sig)

# Forced Copulations by Population and Stage accounting for length

lmm.Force <- lmer(ForcedCop ~ Pop\*Stage + Length + (1|Family/ID), data=court)

anova(lmm.Force)

ranova(lmm.Force)

# Gonopodium Swings by Population and Stage accounting for length

lmm.Swing <- lmer(G\_Swing ~ Pop\*Stage + Length + (1|Family/ID), data=court)

anova(lmm.Swing)

ranova(lmm.Swing)

GRAPHING

### Function for getting standard error

## Gives count, mean, standard deviation, standard error of the mean, and confidence interval (default 95%).

## data: a data frame.

## measurevar: the name of a column that contains the variable to be summariezed

## groupvars: a vector containing names of columns that contain grouping variables

## na.rm: a boolean that indicates whether to ignore NA’s

## conf.interval: the percent range of the confidence interval (default is 95%)

summarySE <- function(data=NULL, measurevar, groupvars=NULL, na.rm=FALSE, conf.interval=.95, .drop=TRUE) { library(plyr)

# New version of length which can handle NA’s: if na.rm==T, don’t count them length2 <- function (x, na.rm=FALSE) { if (na.rm) sum(!is.na(x)) else length(x) }

# This does the summary. For each group’s data frame, return a vector with # N, mean, and sd datac <- ddply(data, groupvars, .drop=.drop, .fun = function(xx, col) { c(N = length2(xx[[col]], na.rm=na.rm), mean = mean (xx[[col]], na.rm=na.rm), sd = sd (xx[[col]], na.rm=na.rm) ) }, measurevar )

# Rename the “mean” column  
datac <- rename(datac, c(“mean” = measurevar))

datacsd / sqrt(datac$N) # Calculate standard error of the mean

# Confidence interval multiplier for standard error # Calculate t-statistic for confidence interval: # e.g., if conf.interval is .95, use .975 (above/below), and use df=N-1 ciMult <- qt(conf.interval/2 + .5, datacci <- datac$se \* ciMult

return(datac) }

### Dodge

pd <- position\_dodge(0.5) # move them 0.05 to the left and right

courtPop, levels=c(“CM”,“CQ”,“QC”,“QH”)) courtStage, levels=c(“Before”, “After”))

dSig <- summarySE(court, measurevar=“Sigmoid”, groupvars=c(“Pop”,“Stage”)) dSig

head(dSig)

graph.Sig <- ggplot(dSig, aes(x=Stage, y=Sigmoid, group=Pop, col=Pop)) + geom\_errorbar(aes(ymin=Sigmoid-se, ymax=Sigmoid+se), position = pd, width=0.1, colour=“black”, size=1) + geom\_line(size=1, position = pd) + geom\_point(position = pd, size=3) + scale\_fill\_brewer(palette=“Set1”) + labs(x=“Exposure Stage”, y=“# Sigmoids +/- SE”) + guides(color=guide\_legend(“Population/Cross”)) + theme\_classic() + theme(axis.title.x = element\_text(face = “bold”, colour = “black”, size = 16, vjust = -1), axis.text.x = element\_text(vjust=0.5, size=12, face = “bold”, colour = “black”), axis.title.y = element\_text(face = “bold”, colour = “black”, size = 16, vjust = 3), axis.text.y = element\_text(vjust = 0.5, size = 12, face = “bold”, colour = “black”), legend.title = element\_text(size = 13), legend.text = element\_text(size = 11, face = “italic”), legend.position=c(0.9, 0.9), legend.background = element\_rect(size=0.5, color=1), plot.margin=unit(c(1,1,1.5,1.2),“cm”))

graph.Sig

dForced <- summarySE(court, measurevar=“ForcedCop”, groupvars=c(“Pop”,“Stage”))

head(dForced) graph.Forced <- ggplot(dForced, aes(x=Stage, y=ForcedCop, group=Pop, col=Pop)) + geom\_errorbar(aes(ymin=ForcedCop-se, ymax=ForcedCop+se), position = pd, width=0.1, colour=“black”, size=1) + geom\_line(size=1, position = pd) + geom\_point(position = pd, size=3) + scale\_fill\_brewer(palette=“Set1”) + labs(x=“Exposure Stage”, y=“# Forced Copulations +/- SE”) + guides(color=guide\_legend(“Population/Cross”)) + theme\_classic() + theme(axis.title.x = element\_text(face = “bold”, colour = “black”, size = 16, vjust = -1), axis.text.x = element\_text(vjust=0.5, size=12, face = “bold”, colour = “black”), axis.title.y = element\_text(face = “bold”, colour = “black”, size = 16, vjust = 3), axis.text.y = element\_text(vjust = 0.5, size = 12, face = “bold”, colour = “black”), legend.title = element\_text(size = 13), legend.text = element\_text(size = 11, face = “italic”), legend.position=c(0.15, 0.8), legend.background = element\_rect(size=0.5, color=1), plot.margin=unit(c(1,1,1.5,1.2),“cm”))

graph.Forced

dSwing <- summarySE(court, measurevar=“G\_Swing”, groupvars=c(“Pop”,“Stage”))

head(dSwing)

graph.Swing <- ggplot(dSwing, aes(x=Stage, y=G\_Swing, group=Pop, col=Pop)) + geom\_errorbar(aes(ymin=G\_Swing-se, ymax=G\_Swing+se), position = pd, width=0.1, colour=“black”, size=1) + geom\_line(size=1, position = pd) + geom\_point(position = pd, size=3) + scale\_fill\_brewer(palette=“Set1”) + labs(x=“Exposure Stage”, y=“# Gonopodium Swings +/- SE”) + guides(color=guide\_legend(“Population/Cross”)) + theme\_classic() + theme(axis.title.x = element\_text(face = “bold”, colour = “black”, size = 16, vjust = -1), axis.text.x = element\_text(vjust=0.5, size=12, face = “bold”, colour = “black”), axis.title.y = element\_text(face = “bold”, colour = “black”, size = 16, vjust = 3), axis.text.y = element\_text(vjust = 0.5, size = 12, face = “bold”, colour = “black”), legend.title = element\_text(size = 13), legend.text = element\_text(size = 11, face = “italic”), legend.position=c(0.15, 0.8), legend.background = element\_rect(size=0.5, color=1), plot.margin=unit(c(1,1,1.5,1.2),“cm”))

graph.Swing

# GG scatterplots

ggplot(court, aes(x = G\_Swing, y = ForcedCop)) + geom\_point() + theme\_classic() court #Add a regression line ggplot(court, aes(x = courtSigmoid)) + geom\_point(color=‘blue’) + theme\_classic() + geom\_smooth(method = “lm”, se = FALSE)

cor(courtSigmoid) #Find moderate positive correlation with correlation coefficient 0.551

court

# Try to run tests of aggressive vs passive behaviors

library(forcats)

levels(

names(court) head(court) #Try to combine variables

mutate(ForcedCop = as.factor(ForcedCop))

court\_temp <- court %>% mutate(Category = fct\_recode(Category, “Aggressive” = “ForcedCop & Lunge & Nip”, “Passive” = “G\_Swing & Chase & Sigmoid”))

menu\_temp <- menu %>% mutate(Category = fct\_recode(Category, “Meats” = “Beef & Pork”, “Meats” = “Chicken & Fish”, “Beverages” = “Coffee & Tea”, “Sweets” = “Desserts”, “Sweets” = “Smoothies & Shakes”))

?dbl ?fct