# The code below is distributed under the terms of the GNU General Public License, Version 3, June 2007.

# A copy of the license can be found at https://www.R-project.org/Licenses/

# List the packages to be used

library(tidyverse)

library(lme4)

library(car)

library(emmeans)

library(jtools)

library(ggrepel)

library(wesanderson)

library(binom)

# Set the working directory (i.e. folder where .csv data file is located)

setwd(choose.dir())

# Read in the data

Grace\_with\_NIO <- read.csv("milkweed\_monarch.by.grazing\_type\_2023.csv")

head(Grace\_with\_NIO)

# Make data frame without NIO site (without Niobrara Valley Preserve, which is block=6 in the data frame)

Grace <- Grace\_with\_NIO %>% filter(block<=5)

# Add minimum value to milkweed and monarch data that includes zeros to prevent error messages from dividing by zero

Grace$milkweeds\_ha\_plus\_one <- (Grace$milkweeds\_ha + 1)

Grace$monarchs\_ha\_plus\_one <- (Grace$eggs\_plus\_larvae\_ha + 1)

Grace$com\_show\_ha\_plus\_one <- (Grace$com\_show\_ha + 1)

Grace$log10\_com\_show\_ha\_plus\_one <- (log10(Grace$com\_show\_ha\_plus\_one))

Grace$grass\_percent\_plus\_one <- (Grace$grass\_percent + 1)

Grace$log10grass\_percent\_plus\_one <- (log10(Grace$grass\_percent\_plus\_one))

# Convert block, field\_number, and visit from a continuous to a categorical [i.e. factor] variable (R defaults to continuous for numerical variables)

Grace <- within(Grace, {

field\_number <- factor(field\_number)

block <- factor(block)

visit <- factor(visit)

})

# Set the order of groups in tables and plots.

Grace\_with\_NIO$graze\_type <- factor(Grace\_with\_NIO$graze\_type, levels = c("Ungrazed","Bison","Cattle"))

Grace$graze\_type <- factor(Grace$graze\_type, levels = c("Ungrazed","Bison","Cattle"))

# Generalized Linear Mixed-Effects Model of the interactive effects of graze\_type and visit (i.e. time) on milkweeds per hectare

# Note (1 | block / field\_number) is a random factor of field nested within site

# Note the glmerControl() function is allowing additional iterations of the model to run to better estimate parameters

gm1 <- glmer(formula = milkweeds\_ha\_plus\_one ~ graze\_type \* visit + (1 | block / field\_number), data = Grace,

family = Gamma(link = "log"), control=glmerControl(optimizer="bobyqa", optCtrl = list(maxfun = 10000000)))

summary(gm1)

Anova(gm1, test = "Chisq")

# Examine all pairwise contrasts (post-hoc)

pairs(emmeans(gm1, "graze\_type"))

emmeans(gm1, "graze\_type")

# Add output from gm1 to data frames that can be used to create graphs

milkweed\_by\_graze\_type <- as.data.frame(emmeans(gm1, "graze\_type"))

milkweed\_by\_graze\_type$milkweeds\_ha <- (exp(milkweed\_by\_graze\_type$emmean)-1)

# Diagnostics for milkweed gamma models

plot(resid(gm1, type = "deviance")~predict(gm1, type = "response")) #deviance residuals vs. predicted values

hist(resid(gm1)) #histogram of residuals from statistical model

# Generalized Linear Mixed-Effects Model of the interactive effects of graze\_type and visit (i.e. time) on juvenile monarchs per hectare

# Note (1 | block / field\_number) is a random factor of field (i.e. grassland unit) nested within site

# Note the glmerControl() function is allowing additional iterations of the model to run to better estimate parameters

gm2 <- glmer(formula = monarchs\_ha\_plus\_one ~ graze\_type \* visit + (1 | block / field\_number), data = Grace,

family = Gamma(link = "log"), control=glmerControl(optimizer="bobyqa", optCtrl = list(maxfun = 1000000)))

summary(gm2)

Anova(gm2, test = "Chisq")

# Examine all pairwise contrasts (post-hoc)

pairs(emmeans(gm2, "graze\_type"))

emmeans(gm2, "graze\_type")

emmeans(gm2, "visit")

# Add output from gm1 to data frames that can be used to create graphs

monarch\_by\_graze\_type <- as.data.frame(emmeans(gm2, "graze\_type"))

monarch\_by\_graze\_type$monarchs\_ha <- (exp(monarch\_by\_graze\_type$emmean)-1)

# Diagnostics for milkweed gamma models

plot(resid(gm2, type = "deviance")~predict(gm2, type = "response"))#deviance residuals vs. predicted values

hist(resid(gm2))#histogram of residuals from statistical model

# Generalized Linear Mixed-Effects Model of the interactive effects of graze\_type and visit (i.e. time) on milkweeds per hectare +

# the covariate of common\_showy abundance (log10\_com\_show\_ha\_plus\_one) is included to try to remove variation

# because common\_showy milkweeds are the main plant for monarch oviposition in all sites except NIO.

# Note (1 | block / field\_number) is a random factor of field (i.e. grassland unit) nested within site

# Note the glmerControl() function is allowing additional iterations of the model to run to better estimate parameters

gm3 <- glmer(formula = monarchs\_ha\_plus\_one ~ graze\_type \* visit + log10\_com\_show\_ha\_plus\_one + (1 | block / field\_number), data = Grace,

family = Gamma(link = "log"), control=glmerControl(optimizer="bobyqa", optCtrl = list(maxfun = 1000000)))

# Estimate the trendline for log10\_com\_show\_ha\_plus\_one explanatory variable and add values to the data frame under new "pi.hat" variable

summary(gm3)

# The pi.hat predicted trendline uses the summary(gm3) coefficients to solve the equation y = exp(intercept + (slope\*log10\_com\_show\_ha\_plus\_one)) and then subtracts 1 because 1 had been added to the raw data

Grace$pi.hat <- ((exp((1.6336 + (0.6623 \* Grace$log10\_com\_show\_ha\_plus\_one))))-1)

Anova(gm3, test = "Chisq")

# Examine all pairwise contrasts (post-hoc)

pairs(emmeans(gm3, "graze\_type"))

emmeans(gm3, "graze\_type")

# Diagnostics for milkweed gamma models (includes log10\_com\_show\_ha\_plus\_one covariate)

plot(resid(gm3, type = "deviance")~predict(gm3, type = "response"))#deviance residuals vs. predicted values

hist(resid(gm3))#histogram of residuals from statistical model

# Visualization of relationship between log10\_com\_show\_ha\_plus\_one and juvenile monarch abundance (cannot color data points by graze\_type)

basic\_plot <- effect\_plot(gm3, pred = log10\_com\_show\_ha\_plus\_one, interval = TRUE, plot.points = TRUE, jitter = 0.1, colors = "blue")

basic\_plot

# Linear Mixed-Effects Model of the interactive effects of graze\_type and visit (i.e. time) on the percentage of grass clumps grazed

# Unlike the previous statistical models, the log10 transformed response variable data are generally normally distributed, so a generalized model is unnecessary

# Note (1 | block / field\_number) is a random factor of field (i.e. grassland unit) nested within site

gm4 <- lmer(formula = log10grass\_percent\_plus\_one ~ graze\_type \* visit + (1 | block / field\_number), data = Grace)

summary(gm4)

Anova(gm4, test = "F")

# Examine all pairwise contrasts (post-hoc)

pairs(emmeans(gm4, "graze\_type"))

emmeans(gm4, "graze\_type")

# Diagnostics for grass\_percent linear mixed effect model

plot(resid(gm4, type = "deviance")~predict(gm4, type = "response"))#deviance residuals vs. predicted values

hist(resid(gm4))#histogram of residuals from statistical model

####################### Bar + datapoint plots for average milkweeds and monarchs #########

# Create a data frame of mean numbers of milkweeds and monarchs per field across all 3 visits

field\_mean <- Grace\_with\_NIO %>% group\_by(field\_number) %>% dplyr::summarise(site=unique(site), block=mean(block), graze\_type=unique(graze\_type), graze\_num=mean(graze\_num), NIO1=mean(NIO1\_other0), milkweeds\_ha\_field=mean(milkweeds\_ha), monarchs\_ha\_field=mean(eggs\_plus\_larvae\_ha))

# Set the order of groups in tables and plots.

field\_mean$graze\_type <- factor(field\_mean$graze\_type, levels = c("Ungrazed","Bison","Cattle"))

##### The first graph is for milkweeds stems per hectare

# Create a dot plot with averages per field, single bars per graze\_type, sites connected by lines, and highlight NIO in red

ggplot() +

geom\_bar(data = milkweed\_by\_graze\_type, aes(x = graze\_type, y = milkweeds\_ha),

stat = "identity", fill = "lightgray", alpha = 0.5, show.legend = FALSE) +

geom\_point(data = subset(field\_mean, NIO1==0), aes(x = graze\_num, y = milkweeds\_ha\_field),

size = 3, color = "black", position = position\_dodge2(width = 0.01), show.legend = FALSE) +

geom\_point(data = subset(field\_mean, NIO1==1), aes(x = graze\_num, y = milkweeds\_ha\_field),

size = 3, color = "red", position = position\_dodge2(width = 0), show.legend = FALSE) +

geom\_text(data = subset(field\_mean, NIO1==0 & graze\_type=='Ungrazed'), aes(x = graze\_num, y = milkweeds\_ha\_field, label = site),

size = 5, color = "black", hjust =1.5, show.legend = FALSE) +

geom\_text(data = subset(field\_mean, NIO1==1 & graze\_type=='Ungrazed'), aes(x = graze\_num, y = milkweeds\_ha\_field, label = site),

size = 5, color = "red", hjust =1.5, vjust = -0.2, show.legend = FALSE) +

geom\_line(data = subset(field\_mean,site=='BRO'), aes(x = graze\_num, y = milkweeds\_ha\_field), size = 1.5, color = "magenta4") +

geom\_line(data = subset(field\_mean,site=='DUN'), aes(x = graze\_num, y = milkweeds\_ha\_field), size = 1.5, color = "blue") +

geom\_line(data = subset(field\_mean,site=='KON'), aes(x = graze\_num, y = milkweeds\_ha\_field), size = 1.5, color = "aquamarine3") +

geom\_line(data = subset(field\_mean,site=='PLA'), aes(x = graze\_num, y = milkweeds\_ha\_field), size = 1.5, color = "sienna3") +

geom\_line(data = subset(field\_mean,site=='TAL'), aes(x = graze\_num, y = milkweeds\_ha\_field), size = 1.5, color = "gold2") +

labs(

x =NULL,

y ="Number of milkweed stems per hectare"

) +

theme(

panel.grid.major = element\_blank(),

panel.grid.minor = element\_blank(),

panel.background = element\_blank(),

strip.text = element\_text(family = "serif", size = 16, face = "bold"),

legend.position = "none",

legend.title = element\_text(family = "serif", size = 14),

legend.text = element\_text(family = "serif", size = 14),

axis.line = element\_line(colour = "black"),

axis.title = element\_text(family = "serif", face = "bold", size = 22),

axis.text = element\_text(family = "serif", face = "bold", size = 18, color = "black")

)

##### The second graph is for monarchs per hectare

# Create a dot plot with averages per field, single bars per graze\_type, sites connected by lines, and highlight NIO in red

# Note that the hash marks ("#") in the plot code can be removed to add the NIO data in red

ggplot() +

geom\_bar(data = monarch\_by\_graze\_type, aes(x = graze\_type, y = monarchs\_ha),

stat = "identity", fill = "lightgray", alpha = 0.5, show.legend = FALSE) +

geom\_point(data = subset(field\_mean, NIO1==0), aes(x = graze\_num, y = monarchs\_ha\_field),

size = 3, color = "black", position = position\_dodge2(width = 0.01), show.legend = FALSE) +

geom\_text\_repel(data = subset(field\_mean, NIO1==0 & graze\_type=='Ungrazed'), aes(x = graze\_num, y = monarchs\_ha\_field, label = site),

size = 5, color = "black", hjust = 1.2, vjust = -0.5, show.legend = FALSE) +

# geom\_point(data = subset(field\_mean, NIO1==1), aes(x = graze\_num, y = monarchs\_ha\_field),

# size = 3, color = "red", position = position\_dodge2(width = 0), show.legend = FALSE) +

# geom\_text\_repel(data = subset(field\_mean, NIO1==1 & graze\_type=='Ungrazed'), aes(x = graze\_num, y = monarchs\_ha\_field, label = site),

# size = 5, color = "red", hjust =1.2, vjust = -0.5, show.legend = FALSE) +

geom\_line(data = subset(field\_mean,site=='BRO'), aes(x = graze\_num, y = monarchs\_ha\_field), size = 1.5, color = "magenta4") +

geom\_line(data = subset(field\_mean,site=='DUN'), aes(x = graze\_num, y = monarchs\_ha\_field), size = 1.5, color = "blue") +

geom\_line(data = subset(field\_mean,site=='KON'), aes(x = graze\_num, y = monarchs\_ha\_field), size = 1.5, color = "aquamarine3") +

geom\_line(data = subset(field\_mean,site=='PLA'), aes(x = graze\_num, y = monarchs\_ha\_field), size = 1.5, color = "sienna3") +

geom\_line(data = subset(field\_mean,site=='TAL'), aes(x = graze\_num, y = monarchs\_ha\_field), size = 1.5, color = "gold2") +

labs(

x =NULL,

y ="Number of juvenile monarchs per hectare"

) +

theme(

panel.grid.major = element\_blank(),

panel.grid.minor = element\_blank(),

panel.background = element\_blank(),

strip.text = element\_text(family = "serif", size = 16, face = "bold"),

legend.position = "none",

legend.title = element\_text(family = "serif", size = 14),

legend.text = element\_text(family = "serif", size = 14),

axis.line = element\_line(colour = "black"),

axis.title = element\_text(family = "serif", face = "bold", size = 22),

axis.text = element\_text(family = "serif", face = "bold", size = 18, color = "black")

)

# Trying to visualize effects of common/showy milkweed abundance on juvenile monarchs in ggplot

# Using ggplot to graph data and predicted line (pi.hat was calculated above)

com\_show\_juveniles\_graph <- ggplot(data = Grace,

aes(x = log10\_com\_show\_ha\_plus\_one, y = jitter(eggs\_plus\_larvae\_ha, 8),

color = factor(graze\_type))) +

geom\_smooth(mapping = aes(y = pi.hat),

color = "black",

linewidth = 1.5) +

geom\_point(show.legend = FALSE, size = 2.5) +

scale\_color\_manual(values = wes\_palette("FantasticFox1")) +

labs( x = expression(bold(paste("log10(", bolditalic("A. speciosa")," and ", bolditalic("A. syriaca "), "stems per hectare + 1)"))),

y = "Number of juvenile monarchs per hectare") +

theme(

panel.grid.major = element\_blank(),

panel.grid.minor = element\_blank(),

panel.background = element\_blank(),

strip.text = element\_text(family = "serif", size = 16, face = "bold"),

legend.position = "none",

legend.title = element\_text(family = "serif", size = 14),

legend.text = element\_text(family = "serif", size = 14),

axis.line = element\_line(colour = "black"),

axis.title = element\_text(family = "serif", face = "bold", size = 22),

axis.text = element\_text(family = "serif", face = "bold", size = 14, color = "black")

)

com\_show\_juveniles\_graph

############ calculating confidence intervals for percentage of milkweed stems containing juvenile monarch per milkweed species in 2023 #########

# For the confidence intervals below, input is the number of stems containing at least one juvenile monarchs, n = the number of total stems sampled

# The Agresti-Coull method is used to calculate confidence intervals

binom.confint(28, n = 508, method = "ac") #A. arenaria

binom.confint(19, n = 363, method = "ac") #A. speciosa / A. syriaca

#binom.confint(0, n = 6, method = "ac") # A. stenophylla (this species is not included in the graph because n is too low)

#binom.confint(0, n = 52, method = "ac") #A. sullivantii (this species is not included in the graph because n is too low)

binom.confint(1, n = 225, method = "ac") #A. tuberosa

binom.confint(0, n = 910, method = "ac") #A. verticillata

binom.confint(1, n = 158, method = "ac") #A. viridiflora

binom.confint(3, n = 459, method = "ac") #A. viridis

# # The following code takes the mean percentage, lower and upper 95% confidence interval for the binom.confint() output for each species and produces Fig. S3

# # The user needs to enter the required data into a dataframe labeled "species\_counts"

# ggplot(species\_counts, aes(x = Species, y = Percentage)) +

# geom\_point(size = 3) +

# geom\_errorbar(aes(ymin = pmax(0, Lower\_95CI), ymax = Upper\_95CI), width = 0.2) +

# labs(

# x = NULL,

# y = "Stems with juvenile monarchs (%)"

# ) +

# scale\_y\_continuous(limits = c(0, 10), labels = scales::label\_number(accuracy = 0.1)) +

# theme\_minimal() +

# theme(

# axis.text.x = element\_text(family = "serif", size = 16, colour = "black", face = "bold.italic", angle = 45, hjust = 1), # X-axis label style

# axis.text.y = element\_text(family = "serif", size = 14, colour = "black"), # Y-axis text style

# axis.title = element\_text(family = "serif", size = 18, face = "bold"),

# axis.line = element\_line(colour = "black"),

# axis.ticks.x = element\_line(colour = "black"), # Adds x-axis tick marks

# axis.ticks.y = element\_line(colour = "black"), # Adds y-axis tick marks

# panel.grid.major = element\_blank(),

# panel.grid.minor = element\_blank(),

# panel.background = element\_blank(),

# strip.text = element\_text(family = "serif", size = 16, face = "bold"),

# legend.position = "none"

# )