**EXERCISE 1**

# R Workshop, March 21-22 2024

# Ty Wagner, USGS

# Exercise 1: Introduction to R

# Description: To gain experience performing basic data manipulations and summaries

# Dataset: Mauch Chunk Lake (Big Bass reg). Nighttime Boat Electrofishing

# Load R packages

library(dplyr) # data manipulation

library(tidyverse) # data manipulation

library(lubridate) # work with dates

library(kableExtra) # make tables

library(ggmap)

# Read in data

# The "././" syntax is backing out 2 directories from our current file

# so we can navigate into the 02\_Data folder where our data are located

dat <- read\_csv("././02\_Data/Mauch\_lake/Mauch\_lake\_surveys\_NBE.csv")

str(dat)

# Data notes:

# Note: in 1986 surveys 35946, 35947 are two seprate entries in the ARDB but have

# exact catch/effort - likely duplicated. Thus only use one (i.e., 35946) and

# not include the other

# Note: in 2000 surveys 35970 and 35971 are likely duplicated in the ARDB,

# thus on retain one for analysis (i.e., 35970) and not inculde the other.

# Clean data

dat\_clean <- dat %>%

select(-c(`Water Name`, `Unique Water Site ID`,

`Water Section ID`, `Group Size Fish Length`,

`Survey Purpose`, `Water Site Comment`,

`Water Site Survey Comment`, Month, Description)) %>% # remove unwanted columns

rename(Water\_site\_survey\_ID = `Water Site Survey ID`,

Lat = `Survey Site Lat DD`,

Long = `Survey Site Lon DD`,

Species = `Fish Species Name`,

Number\_caught = `Number Caught`,

Effort = `Effort Hours`) %>% # rename columns

filter(Water\_site\_survey\_ID != 35947 & Water\_site\_survey\_ID != 35971) %>% # remove duplicate surveys

mutate(Date = mdy\_hm(`Site Date`)) %>% # Convert Site Date to date format

select(-c(`Site Date`)) %>%

# filter(Species == "Largemouth Bass") %>% # Select Largemouth bass

rename\_with(tolower) # make all column names lower case

str(dat\_clean)

head(dat\_clean, 20)

# Calculate total catch for each year, survey, and species

dat\_tot\_catch <- dat\_clean %>%

group\_by(year, water\_site\_survey\_id, species) %>% # group data

mutate(total\_catch = sum(number\_caught)) %>% # sum over catch for each variable in group\_by

ungroup() %>% # ungroup data

distinct(year, water\_site\_survey\_id, species, .keep\_all=TRUE) %>% # retain distinct combos since we don't need size-specific numbers here

select(-c(number\_caught)) # remove number caught, no longer needed

# Species that were not caught in a given survey and year are not recorded as zero catch,

# but we would like to do so. Thus, we have to input missing species names into surveys

# and years where they were not recorded and give them a zero catch values

dat\_tot\_catch2 <- dat\_tot\_catch %>%

select(water\_site\_survey\_id, year, species, total\_catch, effort) %>% # Select columns of interest for summarizing

complete(nesting(water\_site\_survey\_id, year), species, fill = list(total\_catch=0)) %>% # Input missing species for surveys and years

arrange(year, species) # sort by year and species

# Look at first few rows of tot\_catch2

head(dat\_tot\_catch2, 30)

# Lets replace the NA values for imputed species efforts to the actual effort of the survey

dat\_tot\_catch2 <- dat\_tot\_catch2 %>%

group\_by(water\_site\_survey\_id) %>%

mutate(effort = replace\_na(mean(effort, na.rm=T))) %>%

ungroup()

tail(dat\_tot\_catch2, 30)

# Summarize mean catch (across surveys for each year and species)

# sample size, and mean effort.

table1 <- dat\_tot\_catch2 %>%

group\_by(year, species) %>%

summarize(n = n(), total\_catch = sum(total\_catch, na.rm=T),

total\_effort = sum(effort, na.rm=T)) %>%

arrange(year, species)

# Create a table of the catch summary

table1 %>%

kbl(caption = "Catch and effort (hrs) summary table.", digits=2,

col.names = c("Year",

"Species",

"n (surveys)",

"Total catch",

"Total effort (hrs)")) %>%

row\_spec(0,bold=TRUE) %>%

kable\_classic(full\_width = F, html\_font = "Cambria")

# lake\_map <- get\_map(location ='Mauch Chunk Lake', source="google", zoom = 13)

# ggmap(lake\_map) +

# xlab('Longitude')+ylab('Latitude')

**EXERCISE 2**

# R Workshop, March 21-22 2024

# Ty Wagner, USGS

# Exercise 2: Introduction to R - plotting

# Description: To gain experience using ggplot to visualize data

# Dataset: Mauch Chunk Lake (Big Bass reg). Nighttime Boat Electrofishing

# Load R packages

library(dplyr) # data manipulation

library(tidyverse) # data manipulation

library(lubridate) # work with dates

library(kableExtra) # make tables

# New libraries for Ex 2

library(ggplot2) # plotting data and spatial (simple feature; sf) objects

library(sf) # map creation (simple features)

library(spData) # provides access to polygons of US states

# Exercise 1 --------------------------------------------------------------

# Read in data

# The "././" syntax is backing out 2 directories from our current file

# so we can navigate into the 02\_Data folder where our data are located

dat <- read\_csv("././02\_Data/Mauch\_lake/Mauch\_lake\_surveys\_NBE.csv")

str(dat)

# Data notes:

# Note: in 1986 surveys 35946, 35947 are two seprate entries in the ARDB but have

# exact catch/effort - likely duplicated. Thus only use one (i.e., 35946) and

# not include the other

# Note: in 2000 surveys 35970 and 35971 are likely duplicated in the ARDB,

# thus on retain one for analysis (i.e., 35970) and not inculde the other.

# Clean data

dat\_clean <- dat %>%

select(-c(`Water Name`, `Unique Water Site ID`,

`Water Section ID`, `Group Size Fish Length`,

`Survey Purpose`, `Water Site Comment`,

`Water Site Survey Comment`, Month, Description)) %>% # remove unwanted columns

rename(Water\_site\_survey\_ID = `Water Site Survey ID`,

Lat = `Survey Site Lat DD`,

Long = `Survey Site Lon DD`,

Species = `Fish Species Name`,

Number\_caught = `Number Caught`,

Effort = `Effort Hours`) %>% # rename columns

filter(Water\_site\_survey\_ID != 35947 & Water\_site\_survey\_ID != 35971) %>% # remove duplicate surveys

mutate(Date = mdy\_hm(`Site Date`)) %>% # Convert Site Date to date format

select(-c(`Site Date`)) %>%

# filter(Species == "Largemouth Bass") %>% # Select Largemouth bass

rename\_with(tolower) # make all column names lower case

str(dat\_clean)

head(dat\_clean, 20)

# Calculate total catch for each year, survey, and species

dat\_tot\_catch <- dat\_clean %>%

group\_by(year, water\_site\_survey\_id, species) %>% # group data

mutate(total\_catch = sum(number\_caught)) %>% # sum over catch for each variable in group\_by

ungroup() %>% # ungroup data

distinct(year, water\_site\_survey\_id, species, .keep\_all=TRUE) %>% # retain distinct combos since we don't need size-specific numbers here

select(-c(number\_caught)) # remove number caught, no longer needed

# Species that were not caught in a given survey and year are not recorded as zero catch,

# but we would like to do so. Thus, we have to input missing species names into surveys

# and years where they were not recorded and give them a zero catch values

dat\_tot\_catch2 <- dat\_tot\_catch %>%

select(water\_site\_survey\_id, year, species, total\_catch, effort) %>% # Select columns of interest for summarizing

complete(nesting(water\_site\_survey\_id, year), species, fill = list(total\_catch=0)) %>% # Input missing species for surveys and years

arrange(year, species) # sort by year and species

# Look at first few rows of tot\_catch2

head(dat\_tot\_catch2, 30)

# Lets replace the NA values for imputed species efforts to the actual effort of the survey

dat\_tot\_catch2 <- dat\_tot\_catch2 %>%

group\_by(water\_site\_survey\_id) %>%

mutate(effort = replace\_na(mean(effort, na.rm=T))) %>%

ungroup()

tail(dat\_tot\_catch2, 30)

# Summarize mean catch (across surveys for each year and species)

# sample size, and mean effort.

table1 <- dat\_tot\_catch2 %>%

group\_by(year, species) %>%

summarize(n = n(), total\_catch = sum(total\_catch, na.rm=T),

total\_effort = sum(effort, na.rm=T)) %>%

arrange(year, species)

# Create a table of the catch summary

table1 %>%

kbl(caption = "Catch and effort (hrs) summary table.", digits=2,

col.names = c("Year",

"Species",

"n (surveys)",

"Total catch",

"Total effort (hrs)")) %>%

row\_spec(0,bold=TRUE) %>%

kable\_classic(full\_width = F, html\_font = "Cambria")

# Exercise 2 --------------------------------------------------------------

#----- Create a map showing the spatial location of Mauch Chunk Lake

# We are only plotting the location of a single lake (one x,y coordiate)

# so we can just grab the first row of dat\_tot\_catch for our lat/long data

# Subset first row and all columns of dat\_tot\_catch

map\_dat <- dat\_tot\_catch[1,]

# Convert map\_dat to a spatial object (simple feature)

# and set its coordinate references system (crs)

map\_dat <- st\_as\_sf(map\_dat, coords = c("long", "lat"), crs = 4326)

# Grab state boundaries from the spData package and

# transform the coordinate references system (crs)

# crs = 4326 = WGS84; WGS84 CRS is often used for lat and long positions

us\_states2 <- st\_transform(us\_states, crs = 4326)

# Rename column to "State"

colnames(us\_states2)[2] <- "State"

# Select state(s) of interest

selectStates <- c("Pennsylvania")

# Subset us\_states2 data to grab our selectStates for plotting

# The %in% syntax is matching what is in us\_states2$State with

# what is in selectStates (i.e., Pennsylvania, in this case)

us\_state\_select <- us\_states2[us\_states2$State %in% selectStates, ]

# Create map of PA (us\_state\_select) and our single point

# for Mauch Chunk Lake (map\_dat)

ggplot() +

geom\_sf(data = us\_state\_select, color = "gray30", lwd=1, fill="grey80") +

geom\_sf(data=map\_dat, shape=16, size = 5, colour="red") +

labs(title="Mauch Chunk Lake", y="Latitude", x="Longitude") +

theme\_bw() +

theme(axis.text = element\_text(size = 11),

axis.title = element\_text(size = 12))

# Plot catch-per-unit-effort data over time for select species

# Subset table1 for the species Largemouth Bass, Smallmouth Bass,

# and Walleye. We need to ungroup table1 first, then filter and mutate,

# and use complete to fill in years

dat\_lmb\_plot <- ungroup(table1) %>%

filter(species == "Largemouth Bass") %>%

mutate(cpe = total\_catch/total\_effort) %>%

complete(year = min(year):max(year))

# Plot cpe over time

ggplot(data=dat\_lmb\_plot, aes(x=year, y=cpe)) +

geom\_bar(stat="identity", width=0.5) +

labs(title="", y="CPE (fish/hr)", x="Year") +

theme\_bw() +

scale\_x\_continuous(breaks=seq(1981,2022,5))

# Select data for multiple species

dat\_spp\_plot <- ungroup(table1) %>%

filter(species == "Largemouth Bass" | species == "Smallmouth Bass" |

species == "Walleye") %>%

mutate(cpe = total\_catch/total\_effort) %>%

complete(nesting(species), year = min(year):max(year))

# Plot cpe over time for multiple species

ggplot(data=dat\_spp\_plot, aes(x=year, y=cpe)) +

facet\_wrap(~species, scales = "free\_y") +

geom\_bar(stat="identity", width=0.5) +

labs(title="", y="CPE (fish/hr)", x="Year") +

theme\_bw() +

scale\_x\_continuous(breaks=seq(1981,2022,5)) +

theme(axis.text.x = element\_text(angle = 45, vjust = 0.5, hjust=0.5))

**EXERCISE 3**

# Load R package

library(dplyr) # data management

library(tidyverse)

library(ggplot2) # plot

library(lubridate) # dates

library(stringr) # manipulate character stings

library(sf) # map creation

library(spData) # spatial data

library(car) # logit function

library(qs) # save and read in large files

library(cmdstanr) # Bayesian estimation using stan

library(rstanarm) # Bayesian models using R functions and stan

library(mcmcplots) # as.mcmc function

library(ggrepel)

library(kableExtra) # tables

library(ggmap) # Google API data

library(bayesplot)

# Adult brown trout CPE ---------------------------------------------------

# Read in data

adult\_dat <- read\_csv("././02\_Data/Delaware\_tailwaters/NBE WB 2017 to 2021 TNP ver 121622 djp 122022.csv")

str(adult\_dat)

adult\_dat <- adult\_dat %>%

mutate(SampleDate = mdy(SampleDate),

Site = factor(Site),

Wild = factor(Wild),

Stage = factor(Stage)) %>%

filter(Stage != "Yearling" & Stage != "Young of year") %>%

filter(Wild != "Stocked") %>%

filter(SpeciesCd == 328) %>% # select brown trout

rename\_with(tolower) # make all column names lower case

# Get summary statistics: total catch, effort, discharge, and

# calculate cpe

catch\_dat <-adult\_dat %>%

group\_by(site, year, month, riverside) %>%

summarize(total\_catch = n(), effort\_hrs = mean(timefish),

discharge = mean(discharge), cpe = total\_catch/effort\_hrs) %>%

ungroup()

# Make a table of summary statistics

catch\_dat %>%

kbl(caption = "Adult summary catch by site, month, and year", digits=2,

col.names = c("Site",

"Year",

"Month",

"River side",

"Total catch",

"Total effort (hrs)",

"Discharge",

"CPE")) %>%

kable\_classic(full\_width = F, html\_font = "Cambria")

# Plot data

ggplot(data = catch\_dat, mapping = aes(x=year, y=cpe)) +

facet\_wrap(~site) +

geom\_point(aes(color=factor(month))) +

theme\_bw() +

theme(axis.text = element\_text(size = 13),

axis.title = element\_text(size = 13),

strip.text.x = element\_text(size=13)) +

theme(plot.margin = margin(0, .5, .5, .5, "cm")) +

theme(axis.text.x = element\_text(angle = 90, vjust = 0.5, hjust=1)) +

theme(legend.text=element\_text(size=11)) +

labs(title="", y="CPE (fish/hr)", x="Year", color="Month")

# Standardize discharge

catch\_dat <- catch\_dat %>%

mutate(z\_discharge = as.numeric(scale(discharge)),

year = factor(year))

m1 <- stan\_glmer(formula = total\_catch ~ 1 + z\_discharge + year +

(1 | month) + (1|site) , family = neg\_binomial\_2(link='log'),

offset = log(effort\_hrs),

data = catch\_dat,

iter = 1500, chains = 3)

print(m1, digits=3)

sum1 <- summary(m1, probs = c(0.025, 0.975), digits=2)

# Summary for table

sum1[-c(18,21,22),c(1,3:7)] %>%

kbl(caption = "MCMC posterior summaries", booktabs = TRUE, digits=2,

col.names = c("Parameter",

"Posterior mean",

"SD",

"Lower 95% CI",

"Upper 95% CI",

"Effective sample size",

"Rhat")) %>%

kable\_classic(full\_width = F, html\_font = "Cambria") %>%

kable\_styling(latex\_options=c("striped","scale\_down"))

posterior <- as.array(m1)

mcmc\_trace(posterior, pars = c("(Intercept)", "z\_discharge"))

# Grab mcmc draws for selected parameters

fits <- m1 %>%

as\_tibble() %>%

rename(intercept = `(Intercept)`) %>%

select(-starts\_with("Sigma"),-reciprocal\_dispersion ) # exclude the random effects for no and the dispersion parameter

############ Function to sum to get random intercept and slopes

shift\_draws <- function(draws) {

sweep(draws[, -1], MARGIN = 1, STATS = draws[, 1], FUN = "+")

}

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

# Extract intercept and add to beta year values

ints <- fits %>%

select(intercept, starts\_with("year"))

# Calculate intercepts

year\_effects <- shift\_draws(as.matrix(ints))

year\_effects <- data.frame(ints$intercept, year\_effects)

# Rename columns based on year

colnames(year\_effects) <- 2017:2021

head(year\_effects)

# Put year effects on cpe scale

year\_cpe <- exp(year\_effects)

head(year\_cpe)

# Posterior mean for each year

mean\_cpe <- apply(year\_cpe, 2, mean)

# Lower CI

lower <- apply(year\_cpe, 2, quantile, 0.025)

# Upper CI

upper <- apply(year\_cpe, 2, quantile, 0.975)

# Year indicator for plotting

year <- 2017:2021

# Mgt threshold/goal CPE (fish/hr)

mgt\_goal <- 100

# Probability that a given year's CPE is > 100 fish/hr mgt goal

# Write simple function for passing through apply

prob\_fn <- function(x){

mean(x > mgt\_goal)

}

# Probabilities of exceeding mgt goal

mgt\_probs <- apply(year\_cpe, 2, prob\_fn)

plot\_dat <- data.frame(year, mean\_cpe, lower, upper, mgt\_probs)

ggplot(plot\_dat, aes(x=year, y=mean\_cpe)) +

geom\_errorbar(aes(ymin=lower, ymax=upper), width=.1) +

geom\_point(size=2.0) +

geom\_text(aes(label = round(mgt\_probs, digits = 2)),

position=position\_dodge(width=0.9), vjust=-0.75,

hjust = 1.3,size=3.5) +

ylim(0,400) +

ylab('CPE (fish/hr)') +

xlab('Year') +

theme\_bw() +

geom\_hline(yintercept = mgt\_goal, linetype="dashed", color='red') +

theme( axis.title = element\_text(size=12), axis.text = element\_text(size=12) )

#--------------------- Fig non-HM GLM

m\_glm <- stan\_glm(formula = total\_catch ~ 1 + z\_discharge + year,

family = neg\_binomial\_2(link='log'),

offset = log(effort\_hrs),

data = catch\_dat,

iter = 1500, chains = 3)

print(m\_glm, digits=3)

summary(m\_glm, probs = c(0.025, 0.975), digits=2)

# Grab mcmc draws for selected parameters

fits2 <- m\_glm %>%

as\_tibble() %>%

rename(intercept = `(Intercept)`) %>%

select(-reciprocal\_dispersion )

############ Function to sum to get random intercept and slopes

shift\_draws <- function(draws) {

sweep(draws[, -1], MARGIN = 1, STATS = draws[, 1], FUN = "+")

}

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

ints <- fits2 %>%

select(intercept, starts\_with("year"))

# Calculate intercepts

year\_effects <- shift\_draws(as.matrix(ints))

year\_effects <- data.frame(ints$intercept, year\_effects)

# Rename columns based on year

colnames(year\_effects) <- 2017:2021

head(year\_effects)

# Put year effects on cpe scale

year\_cpe <- exp(year\_effects)

head(year\_cpe)

# Posterior mean for each year

mean\_cpe <- apply(year\_cpe, 2, mean)

# Lower CI

lower <- apply(year\_cpe, 2, quantile, 0.025)

# Upper CI

upper <- apply(year\_cpe, 2, quantile, 0.975)

# Year indicator for plotting

year <- 2017:2021

# Mgt goal CPE (fish/hr)

mgt\_goal <- 100

# Probability that a given year's CPE is > 100 fish/hr mgt goal

# Write simple function for passing through apply

prob\_fn <- function(x){

mean(x > mgt\_goal)

}

# Probabilities of exceeding mgt goal

mgt\_probs <- apply(year\_cpe, 2, prob\_fn)

plot\_dat <- data.frame(year, mean\_cpe, lower, upper, mgt\_probs)

ggplot(plot\_dat, aes(x=year, y=mean\_cpe)) +

geom\_errorbar(aes(ymin=lower, ymax=upper), width=.1) +

geom\_point(size=2.0) +

geom\_text(aes(label = round(mgt\_probs, digits = 2)),

position=position\_dodge(width=0.9), vjust=-0.75,

hjust = 1.3,size=3.5) +

ylim(0,400) +

ylab('CPE (fish/hr)') +

xlab('Year') +

theme\_bw() +

labs(title = "Standard GLM") +

geom\_hline(yintercept = mgt\_goal, linetype="dashed", color='red') +

theme( axis.title = element\_text(size=12), axis.text = element\_text(size=12) )

**EXERCISE 4**

# R Workshop, March 21-22 2024

# Ty Wagner, USGS

# Exercise 4: Temporal trends in abundance

# Description: To gain experience fitting HLMs

# Dataset: Fantail darter abundance

# Load R package

library(dplyr) # data management

library(tidyverse)

library(ggplot2) # plot

library(lubridate) # dates

library(stringr) # manipulate character stings

library(sf) # map creation

library(spData) # spatial data

library(car) # logit function

library(qs) # save and read in large files

library(cmdstanr) # Bayesian estimation using stan

library(rstanarm) # Bayesian models using R functions and stan

library(mcmcplots) # as.mcmc function

library(ggrepel)

library(kableExtra) # tables

library(ggmap) # Google API data

library(bayesplot)

# Fantail Darter data ---------------------------------------------------

# Read in data

dart\_dat <- read\_csv("././02\_Data/Fischer\_data/fantaildarter.csv")

# Clean data

dart\_dat <- dart\_dat %>%

select(RecordID, Date, County, Basin, N\_Detected, Lat, Long, HUC\_8, HUC\_12) %>%

mutate(Date = mdy(Date),

year = year(Date),

County = factor(County),

Basin = factor(Basin),

HUC\_8 = factor(HUC\_8),

HUC\_12 = factor(HUC\_12),

decade = floor(year/10)\*10) %>% # Assign each date to a decade (not used in analysis)

filter(!is.na(N\_Detected)) %>%

rename\_with(tolower) # make all column names lower case

str(dart\_dat)

# Sample size per HUC 8

huc\_subset <- dart\_dat %>%

group\_by(huc\_8) %>%

summarize(n = n()) %>%

arrange(-n)

huc\_subset$n = cell\_spec(huc\_subset$n, color = ifelse(huc\_subset$n > 50, "red", "darkgray"))

# Create table

huc\_subset %>%

kbl(booktabs = TRUE, digits=1, escape = FALSE,

col.names = c("HUC 8", "Number of observations")) %>%

kable\_paper(full\_width = FALSE)

# Summarize number of observations by HUC\_8

# Select HUC\_8's with at least 50 observations

dart\_subset <- dart\_dat %>%

group\_by(huc\_8) %>%

summarize(n = n()) %>%

filter(n > 50) %>% # retain HUCs with > 50 observations

arrange(-n)

dart\_dat <- dart\_dat %>%

subset(huc\_8 %in% dart\_subset$huc\_8) %>%

mutate(log\_n = log(n\_detected+1)) %>% # log-transform N (add 1 for 0's)

droplevels()

str(dart\_dat)

# Map locations

# Grab state boundaries from spData and

# transform the coordinate references system (crs)

# crs = 4326 = WGS84; WGS84 CRS is often used for lat and long positions

us\_states2 <- st\_transform(us\_states, crs = 4326)

# Rename column

colnames(us\_states2)[2] <- "State"

# Select state(s) of interest

selectStates <- c("Pennsylvania")

# Subset data for plotting

us\_state\_select <- us\_states2[us\_states2$State %in% selectStates, ]

map.dat <- st\_as\_sf(dart\_dat, coords = c("long", "lat"), crs = 4326)

# Create simple map

ggplot() +

geom\_sf(data = us\_state\_select, color = "gray30", lwd=1, fill="grey80") +

geom\_sf(data=map.dat, shape=16, size = 1, aes(colour=log(n\_detected+1))) +

labs(title="Fantail Darter locations", y="Latitude", x="Longitude", color="log(N detected+1)") +

scale\_color\_gradientn(colours = rainbow(10)) +

theme\_bw() +

theme(axis.text = element\_text(size = 11),

axis.title = element\_text(size = 12))

# Create data summary tables

# Summary of the mean number of detects by HUC 8

dart\_tab1 <- dart\_dat %>%

group\_by(huc\_8) %>%

summarize(mean\_abundance = mean(n\_detected))

# Summary of mean number of detects by HUC and year

dart\_tab2 <- dart\_dat %>%

group\_by(huc\_8, year) %>%

summarize(mean\_logN = mean(log\_n))

# Create a list that has the number detected by year for each HUC 8

dart\_list <- split(dart\_tab2$mean\_logN, dart\_tab2$huc\_8)

# Add a new column to dart\_tab1 that will contain our time trend of detections

# in the table

dart\_tab1 <- data.frame(dart\_tab1, "Trend" = " ")

# Create table

dart\_tab1 %>%

kbl(booktabs = TRUE, digits=1,

col.names = c("HUC 8", "Mean abundance", "Temporal trend (log[N])")) %>%

kable\_paper(full\_width = FALSE) %>%

column\_spec(3, image = spec\_plot(dart\_list, same\_lim = TRUE, minmax = list(),

col = "black",width = 500,

height = 80))

str(dart\_dat)

# Plot data

ggplot(data = dart\_dat, mapping = aes(x=year, y=log\_n) )+

facet\_wrap(~huc\_8) +

geom\_point(alpha=0.3) +

theme\_bw() +

theme(axis.text = element\_text(size = 11),

axis.title = element\_text(size = 13),

strip.text.x = element\_text(size=13)) +

theme(plot.margin = margin(0, .5, .5, .5, "cm")) +

theme(axis.text.x = element\_text(angle = 90, vjust = 0.5, hjust=0.5)) +

scale\_x\_continuous(breaks=seq(1934,2013,10)) +

labs(title="", y="log(N)", x="Year")

# Standardize year predictor

dart\_dat <- dart\_dat %>%

mutate(z\_year = as.numeric(scale(year)))

######## ---------------------------------

# Fit varying intercept and slope model

m1 <- stan\_glmer(formula = log\_n ~ 1 + z\_year + (1 + z\_year | huc\_8),

family = gaussian,

data = dart\_dat,

iter = 1500, chains = 3)

print(m1, digits=3)

# Prepare MCMC summary stats for a table

sum1 <- summary(m1, probs = c(0.025, 0.975), digits=2)

sum1 <- data.frame(sum1[-c(27, 28),c(1,3:5)] )

sum1$parameter <- rownames(sum1)

rownames(sum1) <- NULL

# Summary for table

sum1[-c(27, 28),c(5,1:4)] %>%

kbl(caption = "MCMC posterior summaries", booktabs = TRUE, digits=2,

col.names = c("Parameter",

"Posterior mean",

"SD",

"Lower 95% CI",

"Upper 95% CI")) %>%

kable\_classic(full\_width = F, html\_font = "Cambria") %>%

kable\_styling(latex\_options=c("striped","scale\_down"))

# Put m1 output as an array for use in mcmc\_trace() function

posterior <- as.array(m1)

mcmc\_trace(posterior, pars = c("(Intercept)", "z\_year"))

# Grab mcmc draws for selected parameters

fits <- m1 %>%

as\_tibble() %>%

rename(intercept = `(Intercept)`) %>%

select(-starts\_with("Sigma") ) # exclude the estimated variances

############ Function to sum to get random intercept and slopes

shift\_draws <- function(draws) {

sweep(draws[, -1], MARGIN = 1, STATS = draws[, 1], FUN = "+")

}

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

# Extract intercept and random huc\_8 intercepts and add them together

ints <- fits %>%

select(intercept, starts\_with("b[(Intercept)"))

# Calculate huc\_8 specific intercepts

huc\_ints <- shift\_draws(as.matrix(ints))

# Extract slope and random huc\_8 slopes and add them together

slopes <- fits %>%

select(z\_year, starts\_with("b[z\_year"))

# Calculate huc\_8 specific intercepts

huc\_slopes <- shift\_draws(as.matrix(slopes))

# Create predicted regression line and uncertainty for each HUC\_8

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

# ### BEGIN plotting code

# Create numeric HUC indicator for plotting

dart\_dat <- dart\_dat %>%

mutate(huc\_8\_id = as.numeric(huc\_8)) %>%

arrange(huc\_8\_id)

# Number of hucs

J <- max(dart\_dat$huc\_8\_id)

# Get range of year predictor for each huc

z\_year\_range <- list()

for(i in 1:J){

z\_year\_range[[i]] <- seq(min(dart\_dat$z\_year[dart\_dat$huc\_8\_id==i]), max(dart\_dat$z\_year[dart\_dat$huc\_8\_id==i]), length.out = 30)

}

# Container for predicted values for each group (i.e., each HUC)

linPredGroup <- array(NA, c(dim(huc\_ints)[1],length(z\_year\_range[[1]]),J))

dim(linPredGroup)

# Put each groups MCMC draws for all parameters in its own list

group.ints <- list()

group.slopes <- list()

for(m in 1:J){

group.ints[[m]] <- huc\_ints[,m]

group.slopes[[m]] <- huc\_slopes[,m]

}

for(p in 1:J){ # loop over groups (J)

for(t in 1:length(z\_year\_range[[1]])){

linPredGroup[ ,t,p] <- group.ints[[p]] + group.slopes[[p]] \* z\_year\_range[[p]][t]

}

}

dim(linPredGroup)

# Create containers

# Store posterior means

meanProbGroup <- array(NA, c(length(z\_year\_range[[1]]),J) )

# Store CIs

upperCI.Group <- array(NA, c(length(z\_year\_range[[1]]),J) )

lowerCI.Group <- array(NA, c(length(z\_year\_range[[1]]),J) )

for(i in 1:J){

# Means

meanProbGroup[,i] <- apply(linPredGroup[,,i], 2, mean )

# 95% CIs for fitted values

upperCI.Group[,i] <- apply(linPredGroup[,,i], 2, quantile, probs=c(0.975) )

lowerCI.Group[,i] <- apply(linPredGroup[,,i], 2, quantile, probs=c(0.025) )

}

#######################GGPLOT####################

# New data frame for plotting (not really necessary)

toplot <- dart\_dat

# If you use this scale it cuts out a few points from the graph

Ymin <- 0

Ymax <- max(dart\_dat$log\_n)

# Make a dataframe for lines

# x-values in z\_year\_range

for(i in 1:J){

temp.data=data.frame("huc\_8\_id"=i,"z\_year"=z\_year\_range[[i]], log\_n=meanProbGroup[,i])

if(i==1) line.plot=temp.data else line.plot=rbind(line.plot, temp.data)

temp.CIs=data.frame("huc\_8\_id"=i,"z\_year"=z\_year\_range[[i]], "lower"=lowerCI.Group[,i],"upper"=upperCI.Group[,i])

if(i==1) ci.line.plot=temp.CIs else ci.line.plot=rbind(ci.line.plot,temp.CIs)

}

# New lables for facets

site.labs <- as.character(unique(dart\_dat$huc\_8) )

names(site.labs) <- c("1", "2", "3", "4", "5", "6", "7", "8", "9", "10")

# Function for passing through apply to get

# posterior probs of a decline

post\_prob <- function(x){

mean(x < 0)

}

# Posterior prob of negative trend

slope\_probs <- apply(huc\_slopes, 2, post\_prob)

slope\_probs <- as.vector(slope\_probs)

# Create data frame required for adding probs in a facet for ggplot

slope\_probs\_plot <- data.frame(huc\_8\_id = 1:10, probs = slope\_probs)

# Plot

ggplot() +

geom\_point(data=dart\_dat, aes(z\_year, log\_n),colour="blue", size=.5) +

facet\_wrap(~huc\_8\_id, labeller = labeller(huc\_8\_id = site.labs)) +

theme\_bw() +

scale\_y\_continuous(limits = c(Ymin, Ymax)) +

geom\_line(data=line.plot, aes(z\_year, log\_n), lwd=1) +

geom\_ribbon(data=ci.line.plot, aes(x=z\_year, ymax=upper, ymin=lower), fill="grey", alpha=.75) +

theme(panel.grid = element\_blank(),strip.background = element\_blank()) +

xlab("Year (standardized)")+

ylab(expression(paste(log[e],'(N)' ))) +

theme(axis.text = element\_text(size = 13),

axis.title = element\_text(size = 13),

strip.text.x = element\_text(size=13)) +

geom\_text(aes(x=-3, y=5, label = round(probs, digits = 2)), data=slope\_probs\_plot) # add post probs