**H/L Ratio**

# Set working directory

setwd("C:/Users/User/Desktop/File\_Location")

# Load necessary Libraries

library(MuMIn)

library(tweedie)

library(statmod)

# Load data

data <- read.csv("HOFI.csv", header = TRUE)

# Function to fit and evaluate models with different variance powers

evaluate\_tweedie\_models <- function(data) {

results <- data.frame(variance\_power = numeric(), AIC = numeric())

for (p in seq(1.1, 1.9, by = 0.1)) {

model <- tryCatch(

glm(H.L.Ratios. ~ Gradient + Parasitemia + Date + Sex,

data = data,

family = tweedie(var.power = p, link.power = 0)),

error = function(e) NULL

)

if (!is.null(model)) {

aic\_value <- AICtweedie(model)

results <- rbind(results, data.frame(variance\_power = p, AIC = aic\_value))

}

}

return(results)

}

# Run the evaluation

Best\_Var <- evaluate\_tweedie\_models(data)

# Run Tweedie model evaluation and find the best variance power

results <- evaluate\_tweedie\_models(data)

print(results) # Print all AIC values

# Select the best variance power with the lowest AIC

best\_p <- results$variance\_power[which.min(results$AIC)]

print(paste("Best variance power:", best\_p))

# Fit the global model using the best variance power

global\_model <- glm(H.L.Ratios. ~ Gradient + Parasitemia + Date + Sex,

data = data,

family = tweedie(var.power = best\_p, link.power = 0),

na.action = na.fail)

# Perform model selection using dredge

dredge\_results <- dredge(global\_model, rank = AICtweedie)

# Print the dredge results

print(dredge\_results)

# Perform model averaging on the dredge results within 2 AIC units

model\_avg <- model.avg(dredge\_results, subset = delta < 2)

# Print the summary of the model averaging

print(summary(model\_avg))

**Parasitemia**

# Set working directory

setwd("C:/Users/User/Desktop/File\_Location")

# Load necessary Libraries

library(MuMIn)

library(tweedie)

library(statmod)

# Load data

data <- read.csv("HOFI.csv", header = TRUE)

# Function to fit and evaluate models with different variance powers

evaluate\_tweedie\_models <- function(data) {

results <- data.frame(variance\_power = numeric(), AIC = numeric())

for (p in seq(1.1, 1.9, by = 0.1)) {

model <- tryCatch(

glm(Parasitemia ~ Gradient + Date + Sex,

data = data,

family = tweedie(var.power = p, link.power = 0)),

error = function(e) NULL

)

if (!is.null(model)) {

aic\_value <- AICtweedie(model)

results <- rbind(results, data.frame(variance\_power = p, AIC = aic\_value))

}

}

return(results)

}

# Run the evaluation

Best\_Var <- evaluate\_tweedie\_models(data)

# Run Tweedie model evaluation and find the best variance power

results <- evaluate\_tweedie\_models(data)

print(results) # Print all AIC values

# Select the best variance power with the lowest AIC

best\_p <- results$variance\_power[which.min(results$AIC)]

print(paste("Best variance power:", best\_p))

# Fit the global model using the best variance power

global\_model <- glm(Parasitemia ~ Gradient + Date + Sex,

data = data,

family = tweedie(var.power = best\_p, link.power = 0),

na.action = na.fail)

# Perform model selection using dredge

dredge\_results <- dredge(global\_model, rank = AICtweedie)

# Print the dredge results

print(dredge\_results)

# Perform model averaging on the dredge results within 2 AIC units

model\_avg <- model.avg(dredge\_results, subset = delta < 2)

# Print the summary of the model averaging

print(summary(model\_avg))