**Supplement 2 – Software Code for R**

By Tom R. Olk, Jens Wollebæk and Espen Lydersen

Tom R. Olk is Ph D. candidate at the University of Southeast-Norway

Jens Wollebæk (Ph D) is senior advisor at the Norwegian Agriculture Agency

Espen Lydersen (Ph D) is professor at the University of Southeast-Norway

*Correspondence to*: Tom R. Olk (Tom.Robin.Olk@usn.no)

rm(list=ls())

setwd("##Directory where data is stored##")

Data <- read.csv2(file = "Data.csv", sep = ";")

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

Plots Accumulated Mortality
#############################################################################

tiff(filename = "AccumulatedMortality.tiff", width = 6000, height = 4000, units = "px", pointsize = 18, res = 600)

op <- par(mfrow = c(1,2), las = 1)

DataFP <- subset(Data, Data$Treatment == 1 & Data$Picking == 1)

plot(NA, ylim = c(0, 1), xlim = range(DataFP$dpf), bty = "c", xaxt = "n", xlab = "", ylab = "")

for(i in unique(DataFP$Location)){

 temp <- subset(DataFP, Location == i)

 points (amf ~ dpf, data = temp, type = "b", pch = as.numeric(temp$dpf))

 points(amf ~ dpf, data = temp, type = "b", pch = 1)

}

axis(1, at = c(35, 42, 49, 56, 63, 70, 77, 84))

mtext("Accumulated Mortality (Fraction)", line = 2.5, adj = 0.5, side = 2, cex = 1, las = 3)

mtext("dpf", line = 2.5, adj = 0.5, side = 1, cex = 1)

DataNFP <- subset(Data, Data$Treatment == 0 & Data$Picking == 1)

plot(NA, ylim = c(0, 1), xlim = range(DataNFP$dpf), bty = "c", xaxt = "n", xlab = "", ylab = "")

for(i in unique(DataNFP$Location)){

 temp <- subset(DataNFP, Location == i)

 points (amf ~ dpf, data = temp, type = "b", pch = as.numeric(temp$dpf))

 points(amf ~ dpf, data = temp, type = "b", pch = 1)

}

axis(1, at = c(35, 42, 49, 56, 63, 70, 77, 84))

mtext("Accumulated Mortality (Fraction)", line = 2.5, adj = 0.5, side = 2, cex = 1, las = 3)

mtext("dpf", line = 2.5, adj = 0.5, side = 1, cex = 1)

dev.off()

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

Plots Accumulated Hatch
#############################################################################

tiff(filename = "AccumulatedHatch.tiff", width = 6000, height = 4000, units = "px", pointsize = 18, res = 600)

op <- par(mfrow = c(1,2), las = 1)

DataFP <- subset(Data, Data$Treatment == 1 & Data$Picking == 1)

plot(NA, ylim = c(0, 0.15), xlim = range(DataFP$dpf), bty = "c", xaxt = "n", xlab = "", ylab = "")

for(i in unique(DataFP$Location)){

 temp <- subset(DataFP, Location == i)

 points (ahf ~ dpf, data = temp, type = "b", pch = as.numeric(temp$dpf))

 points(ahf ~ dpf, data = temp, type = "b", pch = 1)

}

axis(1, at = c(35, 42, 49, 56, 63, 70, 77, 84))

mtext("Accumulated Hatch (Fraction)", line = 2.5, adj = 0.5, side = 2, cex = 1, las = 3)

mtext("dpf", line = 2.5, adj = 0.5, side = 1, cex = 1)

DataNFP <- subset(Data, Data$Treatment == 0 & Data$Picking == 1)

plot(NA, ylim = c(0, 0.15), xlim = range(DataNFP$dpf), bty = "c", xaxt = "n", xlab = "", ylab = "")

for(i in unique(DataNFP$Location)){

 temp <- subset(DataNFP, Location == i)

 points (ahf ~ dpf, data = temp, type = "b", pch = as.numeric(temp$dpf))

 points(ahf ~ dpf, data = temp, type = "b", pch = 1)

}

axis(1, at = c(35, 42, 49, 56, 63, 70, 77, 84))

mtext("Accumulated Hatch (Fraction)", line = 2.5, adj = 0.5, side = 2, cex = 1, las = 3)

mtext("dpf", line = 2.5, adj = 0.5, side = 1, cex = 1)

dev.off()

tiff(filename = "HistogramsHatch.tiff", width = 6000, height = 6000, units = "px", pointsize = 18, res = 600)

op <- par(mfrow = c(2,2), las = 1)

hist(DataP84$ahf[DataP84$Treatment == 1], main = "", xlab = "Eggs hatched (Fraction)", ylab = "Number of plots", xlim = c(0, 0.15), ylim = c(0, 15), breaks = c(0, 0.025, 0.05, 0.075, 0.10, 0.125, 0.15))

hist(DataP84$ahf[DataP84$Treatment == 0], main = "", xlab = "Eggs hatched (Fraction)", ylab = "Number of plots", xlim = c(0, 0.15), ylim = c(0, 15), breaks = c(0, 0.025, 0.05, 0.075, 0.10, 0.125, 0.15))

hist(Data84$ahf[Data84$Treatment == 1], main = "", xlab = "Eggs hatched (Fraction)", ylab = "Number of plots", xlim = c(0, 0.15), ylim = c(0, 15), breaks = c(0, 0.025, 0.05, 0.075, 0.10, 0.125, 0.15))

hist(Data84$ahf[Data84$Treatment == 0], main = "", xlab = "Eggs hatched (Fraction)", ylab = "Number of plots", xlim = c(0, 0.15), ylim = c(0, 15), breaks = c(0, 0.025, 0.05, 0.075, 0.10, 0.125, 0.15))

par(op)

dev.off()

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

Descriptive statistics and t-tests for hatch and mortality
###########################################################################

Data$Treatment <- as.factor(Data$Treatment)

Data$PHS <- Data$Hatched/Data$TotalNo

Data$PDS <- Data$Dead/Data$TotalNo

DataP35 <- subset(Data, Data$Picking == 1 & Data$dpf == 35)

DataP42 <- subset(Data, Data$Picking == 1 & Data$dpf == 42)

DataP49 <- subset(Data, Data$Picking == 1 & Data$dpf == 49)

DataP56 <- subset(Data, Data$Picking == 1 & Data$dpf == 56)

DataP63 <- subset(Data, Data$Picking == 1 & Data$dpf == 63)

DataP70 <- subset(Data, Data$Picking == 1 & Data$dpf == 70)

DataP77 <- subset(Data, Data$Picking == 1 & Data$dpf == 77)

DataP84 <- subset(Data, Data$Picking == 1 & Data$dpf == 84)

Data84 <- subset(Data, Data$Picking == 0 & Data$dpf == 84)

library(psych)

describe(DataP35$amf[DataP35$Treatment == 1])

# vars n mean sd median trimmed mad min max range skew kurtosis se

#X1 1 15 0.57 0.25 0.56 0.57 0.35 0.29 0.95 0.67 0.33 -1.47 0.06

describe(DataP35$amf[DataP35$Treatment == 0])

# vars n mean sd median trimmed mad min max range skew kurtosis se

#X1 1 15 0.6 0.25 0.63 0.6 0.42 0.3 0.97 0.66 0.17 -1.63 0.06

describe(DataP84$amf[DataP84$Treatment == 1])

# vars n mean sd median trimmed mad min max range skew kurtosis se

#X1 1 15 0.95 0.04 0.96 0.96 0.04 0.88 1 0.12 -0.56 -1.13 0.01

describe(DataP84$amf[DataP84$Treatment == 0])

# vars n mean sd median trimmed mad min max range skew kurtosis se

#X1 1 15 0.97 0.03 0.97 0.97 0.03 0.91 1 0.09 -0.46 -1.08 0.01

describe(Data84$amf[Data84$Treatment == 1])

# vars n mean sd median trimmed mad min max range skew kurtosis se

#X1 1 15 0.99 0.01 0.99 0.99 0.01 0.95 1 0.05 -0.97 -0.3 0

describe(Data84$amf[Data84$Treatment == 0])

# vars n mean sd median trimmed mad min max range skew kurtosis se

#X1 1 15 1 0 1 1 0 1 1 0 -0.43 -1.84 0

describe(DataP84$ahf[DataP84$Treatment == 1])

# vars n mean sd median trimmed mad min max range skew kurtosis se

#X1 1 15 0.05 0.04 0.04 0.04 0.04 0 0.12 0.12 0.56 -1.13 0.01

describe(DataP84$ahf[DataP84$Treatment == 0])

# vars n mean sd median trimmed mad min max range skew kurtosis se

#X1 1 15 0.03 0.03 0.03 0.03 0.03 0 0.09 0.09 0.46 -1.08 0.01

describe(Data84$ahf[Data84$Treatment == 1])

# vars n mean sd median trimmed mad min max range skew kurtosis se

#X1 1 15 0.01 0.01 0.01 0.01 0.01 0 0.05 0.05 0.97 -0.3 0

describe(Data84$ahf[Data84$Treatment == 0])

# vars n mean sd median trimmed mad min max range skew kurtosis se

#X1 1 15 0 0 0 0 0 0 0 0 0.43 -1.84 0

DataP70$hp <- DataP70$Hatched/DataP70$TotalNo \* 100

mean(DataP70$hp[DataP70$Treatment == 0])

# [1] 2.232311

mean(DataP70$hp[DataP70$Treatment == 1])

# [1] 3.201297

describe(DataP70$hp[DataP70$Treatment == 0], type=2)

#vars n mean sd median trimmed mad min max range skew kurtosis se

#X1 1 15 2.23 1.98 1.47 2.09 2.17 0 6.31 6.31 0.55 -0.84 0.51

describe(DataP70$hp[DataP70$Treatment == 1], type=2)

#vars n mean sd median trimmed mad min max range skew kurtosis se

# 1 15 3.2 3.26 2.3 2.98 2.92 0 9.24 9.24 0.87 -0.64 0.84

t.test(PDS ~ Treatment, data = DataP35, paired = TRUE)

# Paired t-test

#data: PDS by Treatment

#t = 1.2942, df = 14, p-value = 0.2165

#alternative hypothesis: true difference in means is not equal to 0

#95 percent confidence interval:

# -0.01911027 0.07726269

#sample estimates:

#mean of the differences

# 0.02907621

t.test(PDS ~ Treatment, data = DataP42, paired = TRUE)

#Paired t-test

#data: PDS by Treatment

#t = 1.4277, df = 14, p-value = 0.1753

#alternative hypothesis: true difference in means is not equal to 0

#95 percent confidence interval:

# -0.006601853 0.032890113

#sample estimates:

#mean of the differences

 0.01314413

t.test(PDS ~ Treatment, data = DataP49, paired = TRUE)

# Paired t-test

#data: PDS by Treatment

#t = 0.68535, df = 14, p-value = 0.5043

#alternative hypothesis: true difference in means is not equal to 0

#95 percent confidence interval:

# -0.01165284 0.02259721

#sample estimates:

#mean of the differences

# 0.005472185

t.test(PDS ~ Treatment, data = DataP56, paired = TRUE)

#Paired t-test

#data: PDS by Treatment

#t = -1.5575, df = 14, p-value = 0.1417

#alternative hypothesis: true difference in means is not equal to 0

#95 percent confidence interval:

# -0.040296825 0.006391833

#sample estimates:

#mean of the differences

# -0.0169525

t.test(PDS ~ Treatment, data = DataP63, paired = TRUE)

#Paired t-test

#data: PDS by Treatment

#t = 0.14963, df = 14, p-value = 0.8832

#alternative hypothesis: true difference in means is not equal to 0

#95 percent confidence interval:

# -0.01374663 0.01580850

#sample estimates:

#mean of the differences

# 0.001030935

t.test(PDS ~ Treatment, data = DataP70, paired = TRUE)

#Paired t-test

#data: PDS by Treatment

#t = -2.238, df = 14, p-value = 0.04199

#alternative hypothesis: true difference in means is not equal to 0

#95 percent confidence interval:

# -0.0351084960 -0.0007470058

#sample estimates:

#mean of the differences

# -0.01792775

t.test(PDS ~ Treatment, data = DataP77, paired = TRUE)

#Paired t-test

#data: PDS by Treatment

#t = -1.2831, df = 14, p-value = 0.2203

#alternative hypothesis: true difference in means is not equal to 0

#95 percent confidence interval:

# -0.007480910 0.001880569

#sample estimates:

#mean of the differences

# -0.00280017

t.test(PDS ~ Treatment, data = DataP84, paired = TRUE)

#Paired t-test

#data: PDS by Treatment

#t = 1.2367, df = 14, p-value = 0.2366

#alternative hypothesis: true difference in means is not equal to 0

#95 percent confidence interval:

# -0.001264036 0.004706677

#sample estimates:

#mean of the differences

# 0.00172132

t.test(PHS ~ Treatment, data = DataP63, paired = TRUE)

# Paired t-test

#data: PHS by Treatment

#t = -0.86542, df = 14, p-value = 0.4014

#alternative hypothesis: true difference in means is not equal to 0

#95 percent confidence interval:

# -0.006099884 0.002592505

#sample estimates:

#mean of the differences

# -0.001753689

t.test(PHS ~ Treatment, data = DataP70, paired = TRUE)

# Paired t-test

#data: PHS by Treatment

#t = -1.7992, df = 14, p-value = 0.09357

#alternative hypothesis: true difference in means is not equal to 0

#95 percent confidence interval:

# -0.021240981 0.001861279

#sample estimates:

#mean of the differences

# -0.009689851

t.test(PHS ~ Treatment, data = DataP77, paired = TRUE)

# Paired t-test

#data: PHS by Treatment

#t = -0.46645, df = 14, p-value = 0.6481

#alternative hypothesis: true difference in means is not equal to 0

#95 percent confidence interval:

# -0.005851946 0.003761268

#sample estimates:

#mean of the differences

# -0.001045339

t.test(PHS ~ Treatment, data = DataP84, paired = TRUE)

#Paired t-test

#data: PHS by Treatment

#t = -1, df = 14, p-value = 0.3343

#alternative hypothesis: true difference in means is not equal to 0

#95 percent confidence interval:

# -0.0008663324 0.0003153682

#sample estimates:

#mean of the differences

# -0.0002754821

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

Two-way ANOVA
#############################################################################

DataP <- read.csv2(file = "SummaryPlot.csv", sep = ";")

DataP$HPercent <- DataP$Hatched/DataP$TotalNo \* 100

DataP$DPercent <- DataP$Dead/DataP$TotalNo \* 100

op <- par(mfrow = c(1,1), las = 1)

DataP$AHPercent <- asin(sqrt(DataP$HPercent/100))

DataP$Treatment <- as.factor(DataP$Treatment)

DataP$Picking <- as.factor(DataP$Picking)

Ano <- aov(AHPercent ~ 1 + Treatment \* Picking, data = DataP)

summary(Ano)

# Df Sum Sq Mean Sq F value Pr(>F)

#Treatment 1 0.0321 0.03212 4.432 0.0398 \*

#Picking 1 0.1971 0.19706 27.185 2.77e-06 \*\*\*

#Treatment:Picking 1 0.0071 0.00713 0.983 0.3256

#Residuals 56 0.4059 0.00725

#---

#Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1

PostHoc <- TukeyHSD(x = Ano, conf.level = 0.95)

tiff(filename = "TotalHatchPostHoc.tiff", width = 4000, height = 6000, units = "px", pointsize = 18, res = 500)

plot(PostHoc)

dev.off()

PostHoc

#Tukey multiple comparisons of means

# 95% family-wise confidence level

#Fit: aov(formula = AHPercent ~ 1 + Treatment \* Picking, data = DataP)

#$Treatment

 diff lwr upr p adj

# 1-0 0.0462776 0.002240651 0.09031454 0.0397754

#$Picking

 diff lwr upr p adj

# 1-0 0.1146176 0.07058065 0.1586545 2.8e-06

#$`Treatment:Picking`

 diff lwr upr p adj

#1:0-0:0 0.06807703 -0.01424168 0.1503957 0.1386294

#0:1-0:0 0.13641702 0.05409832 0.2187357 0.0002919

#1:1-0:0 0.16089519 0.07857648 0.2432139 0.0000186

#0:1-1:0 0.06834000 -0.01397871 0.1506587 0.1362681

#1:1-1:0 0.09281817 0.01049946 0.1751369 0.0212003

#1:1-0:1 0.02447817 -0.05784054 0.1067969 0.8598787

# Transforming results to fractions

(sin(0.046))^2

# [1] 0.002114508

(sin(0.002))^2

# [1] 3.999995e-06

(sin(0.090))^2

# [1] 0.008078154

(sin(0.115))^2

# [1] 0.0131668

(sin(0.071))^2

# [1] 0.005032535

(sin(0.159))^2

# [1] 0.02506867

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

## ICC

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

#ICC and confidence intervals

library(ICC)

DPForm <- subset(DataPicked, DataPicked$Treatment == 1)

ICCest(x = Location, y = FunIn, data = DPForm, alpha = 0.05, CI.type = "THD")

#$ICC

#[1] 0.02590674

#$LowerCI

#[1] -0.05147685

#$UpperCI

#[1] 0.2083786

#$N

#[1] 15

#$k

#[1] 8

#$varw

#[1] 0.2238095

#$vara

#[1] 0.005952381

DPNOT <- subset(DataPicked, DataPicked$Treatment == 0)

ICCest(x = Location, y = FunIn, data = DPNOT, alpha = 0.05, CI.type = "THD")

#$ICC

#[1] 0.008181246

#$LowerCI

#[1] -0.06175137

#$UpperCI

#[1] 0.1777967

#$N

#[1] 15

#$k

#[1] 8

#$varw

#[1] 0.2345238

#$vara

#[1] 0.001934524

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

Model M1
#############################################################################

MFT <- gamm(FunIn ~ Treatment, random = list(Location = ~ 1), family = binomial(link = "logit"), data = DataPicked, method = "REML")

summary(MFT$gam)

summary(MFT$lme)

##Intraclass correlation

(0.4365363)^2/(0.4365363^2+1^2)

#0.1600619

de2 <- 1 + (16 - 1) \* 0.1600619

de2

# 3.400929

Neff2 <- 15\*16/de2

Neff2

# 70.57

# Plotting the model

op <- par(mfrow = c(2,2), las = 1)

E1 <- resid(MFT$gam, type = "pearson")

F1 <- fitted(MFT$gam)

eta <- predict(MFT$gam, type = "link")

tiff(filename = "MFTResid.tiff", width = 6000, height = 6000, units = "px", pointsize = 18, res = 600)

op <- par(mfrow = c(2,2), las = 1)

plot(E1 ~ F1, xlab = "Fitted Values", ylab = "Pearson Residuals")

abline(h = 0, v = 0, lty = 2)

plot(E1 ~ eta, xlab = "Predicted Values", ylab = "Pearson Residuals")

abline(h = 0, v = 0, lty = 2)

plot(E1 ~ DataPicked$Treatment, xlab = "Treatment", ylab = "Pearson Residuals")

abline(h = 0, lty = 2)

dev.off()

tiff(filename = "MFTResid2.tiff", width = 6000, height = 6000, units = "px", pointsize = 18, res = 500)

op <- par(mfrow = c(1,2), las = 1)

plot(E1 ~ DataPicked$dpf, xlab = "dpf", ylab = "Pearson Residuals")

abline(h = 0, v = 0, lty = 2)

plot(E1 ~ DataPicked$Eleft, xlab = "Number of remaining eggs", ylab = "Pearson Residuals")

abline(h = 0, v = 0, lty = 2)

dev.off()

pdNT <- (75 + 2)/(120+4)

pdT <- (42+2)/(120+4)

NTL <- pdNT - 1.96\*sqrt((pdNT\*(1-pdNT))/(120+4))

NTU <- pdNT + 1.96\*sqrt((pdNT\*(1-pdNT))/(120+4))

TL <- pdT - 1.96\*sqrt((pdT\*(1-pdT))/(120+4))

TU <- pdT + 1.96\*sqrt((pdT\*(1-pdT))/(120+4))

NTL

NTU

TL

TU

tiff(filename = "FungusMFT.tiff", width = 6000, height = 6000, units = "px", pointsize = 18, res = 600)

op <- par(mfrow = c(1,1), las = 1)

DataT1 <- data.frame(Eleft = seq(1, 120, length = 120), Treatment = 1)

DataT0 <- data.frame(Eleft = seq(1, 120, length = 120), Treatment = 0)

P0 <- predict(MFT$gam, newdata = DataT0, type = "response")

P1 <- predict(MFT$gam, newdata = DataT1, type = "response")

plot(DataPicked$FunIn[DataPicked$Treatment == 0] ~ DataT0$Eleft, col = "red", xlab = "Number of observation", ylab = "Probability of fungal infection")

points(DataPicked$FunIn[DataPicked$Treatment == 1] ~ DataT1$Eleft, col = "blue")

lines(DataT0$Eleft, P0, col = "red", lty = 1)

lines(DataT1$Eleft, P1, col = "blue", lty = 1)

abline(h = NTL, lty = 2, col = "red")

abline(h = NTU, lty = 2, col = "red")

abline(h = TL, lty = 2, col = "blue")

abline(h = TU, lty = 2, col = "blue")

legend(70, 0.2, c("Not treated with formalin", "Treated with formalin"), lty = 1, col = c("red", "blue"))

par(op)

dev.off()

## Intercept, which is the probability to contract fungal infections within one week for untreated plots:

# 0.5199

exp(0.5199+0)/(1+exp(0.5199+0))

# 0.6271244

## Estimating for treated eggs

exp(0.5199-1.1510974)/(1+exp(0.5199-1.1510974))

# 0.3472391

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

 Model M2
###########################################################################

DataPicked <- subset(Data, Data$Picking == 1)

library(mgcv)

MFfull <- gamm(FunIn ~ Treatment + Eleft, random = list(Location = ~ 1), family = binomial(link = "logit"), data = DataPicked, method = "REML")

summary(MFfull)

summary(MFfull$gam)

summary(MFfull$lme)

op <- par(mfrow = c(2,2), las = 1)

E1 <- resid(MFfull$gam, type = "pearson")

F1 <- fitted(MFfull$gam)

eta <- predict(MFfull$gam, type = "link")

tiff(filename = "FPResid.tiff", width = 6000, height = 6000, units = "px", pointsize = 18, res = 600)

op <- par(mfrow = c(2,2), las = 1)

plot(E1 ~ F1, xlab = "Fitted Values", ylab = "Pearson Residuals")

abline(h = 0, v = 0, lty = 2)

plot(E1 ~ eta, xlab = "Predicted Values", ylab = "Pearson Residuals")

abline(h = 0, v = 0, lty = 2)

plot(E1 ~ DataPicked$Eleft, xlab = "Eggs left", ylab = "Pearson Residuals")

abline(h = 0, v = 0, lty = 2)

plot(E1 ~ DataPicked$Treatment, xlab = "Treatment", ylab = "Pearson Residuals")

abline(h = 0, lty = 2)

dev.off()

tiff(filename = "Fungus.tiff", width = 6000, height = 6000, units = "px", pointsize = 18, res = 600)

op <- par(mfrow = c(1,1), las = 1)

min(DataPicked$Eleft)

max(DataPicked$Eleft)

# 0 , 202

DataT1 <- data.frame(Eleft = seq(0, 202, length = 200), Treatment = 1)

DataT0 <- data.frame(Eleft = seq(0, 202, length = 200), Treatment = 0)

P0 <- predict(MFfull$gam, newdata = DataT0, type = "response")

P1 <- predict(MFfull$gam, newdata = DataT1, type = "response")

plot(DataPicked$FunIn[DataPicked$Treatment == 0] ~ DataPicked$Eleft[DataPicked$Treatment == 0], col = "red", xlab = "Eggs left", ylab = "Probability of fungal infection")

points(DataPicked$FunIn[DataPicked$Treatment == 1] ~ DataPicked$Eleft[DataPicked$Treatment == 1], col = "blue")

lines(DataT0$Eleft, P0, col = "red", lty = 1)

lines(DataT1$Eleft, P1, col = "blue", lty = 1)

legend(120, 0.2, c("Not treated with formalin", "Treated with formalin"), lty = 1, col = c("red", "blue"))

par(op)

dev.off()

summary(MFfull$lme)

## Intraclass correlation

(0.429176)^2/(0.429176^2+1^2)

# 0.1555424

#design effect:

de <- 1 + (16 - 1) \* 0.1555424

de

# 3.33

Neff <- 15 \* 16/de

Neff

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

Model M3
##############################################################################

MDI <- gamm(FunIn ~ Treatment + Dead, random = list(Location = ~ 1), family = binomial(link = "logit"), data = DataPicked, method = "REML")

summary(MDI$gam)

#Family: binomial

#Link function: logit

#Formula:

# FunIn ~ Treatment + Dead

#Parametric coefficients:

# Estimate Std. Error t value Pr(>|t|)

#(Intercept) 0.391361 0.234175 1.671 0.0960 .

#Treatment -1.171124 0.275609 -4.249 3.08e-05 \*\*\*

# Dead 0.004119 0.002391 1.723 0.0863 .

#---

# Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1

#R-sq.(adj) = 0.0782

#Scale est. = 1 n = 240

summary(MDI$lme)

#Linear mixed-effects model fit by maximum likelihood

#Data: data

#AIC BIC logLik

#1047.815 1061.738 -519.9076

#Random effects:

# Formula: ~1 | Location

#(Intercept) Residual

#StdDev: 0.4371495 1

#Variance function:

# Structure: fixed weights

#Formula: ~invwt

#Fixed effects: list(fixed)

#Value Std.Error DF t-value p-value

#X(Intercept) 0.3913613 0.23417541 223 1.671231 0.0961

#XTreatment -1.1711244 0.27560854 223 -4.249231 0.0000

#XDead 0.0041191 0.00239133 223 1.722505 0.0864

#Correlation:

# X(Int) XTrtmn

#XTreatment -0.555

#XDead -0.295 -0.071

#Standardized Within-Group Residuals:

# Min Q1 Med Q3 Max

#-1.4563768 -0.8085888 -0.5136913 0.8118378 1.9425968

#Number of Observations: 240

#Number of Groups: 15

##Intraclass correlation

(0.4371495)^2/(0.4371495^2+1^2)

#0.1604397

de2 <- 1 + (16 - 1) \* 0.1604397

de2

# 3.406595

Neff2 <- 15\*16/de2

Neff2

# 70.45157

# No pronounced reduction in effective sample size, Without the insignificant fixed effect, this model is more certain

# Plotting the model

op <- par(mfrow = c(2,2), las = 1)

E1 <- resid(MDI$gam, type = "pearson")

F1 <- fitted(MDI$gam)

eta <- predict(MDI$gam, type = "link")

tiff(filename = "MDIResid.tiff", width = 6000, height = 6000, units = "px", pointsize = 18, res = 600)

op <- par(mfrow = c(2,2), las = 1)

plot(E1 ~ F1, xlab = "Fitted Values", ylab = "Pearson Residuals")

abline(h = 0, v = 0, lty = 2)

plot(E1 ~ eta, xlab = "Predicted Values", ylab = "Pearson Residuals")

abline(h = 0, v = 0, lty = 2)

plot(E1 ~ DataPicked$Treatment, xlab = "Treatment", ylab = "Pearson Residuals")

abline(h = 0, lty = 2)

plot(E1 ~ DataPicked$Dead, xlab = "Number of dead eggs", ylab = "Pearson Residuals")

abline(h = 0, lty = 2)

dev.off()

tiff(filename = "MDIResid2.tiff", width = 6000, height = 6000, units = "px", pointsize = 18, res = 600)

op <- par(mfrow = c(1,2), las = 1)

plot(E1 ~ DataPicked$dpf, xlab = "dpf", ylab = "Pearson Residuals")

abline(h = 0, v = 0, lty = 2)

plot(E1 ~ DataPicked$Eleft, xlab = "Number of remaining eggs", ylab = "Pearson Residuals")

abline(h = 0, v = 0, lty = 2)

dev.off()

pdNT <- (75 + 2)/(120+4)

pdT <- (42+2)/(120+4)

NTL <- pdNT - 1.96\*sqrt((pdNT\*(1-pdNT))/(120+4))

NTU <- pdNT + 1.96\*sqrt((pdNT\*(1-pdNT))/(120+4))

TL <- pdT - 1.96\*sqrt((pdT\*(1-pdT))/(120+4))

TU <- pdT + 1.96\*sqrt((pdT\*(1-pdT))/(120+4))

NTL

NTU

TL

TU

tiff(filename = "FungusMDI.tiff", width = 6000, height = 6000, units = "px", pointsize = 18, res = 600)

op <- par(mfrow = c(1,1), las = 1)

DataT1 <- data.frame(Dead = seq(0, 320, length = 320), Treatment = 1)

DataT0 <- data.frame(Dead = seq(0, 320, length = 320), Treatment = 0)

P0 <- predict(MDI$gam, newdata = DataT0, type = "response")

P1 <- predict(MDI$gam, newdata = DataT1, type = "response")

plot(DataPicked$FunIn[DataPicked$Treatment == 1] ~ DataPicked$Dead[DataPicked$Treatment == 1], col = "blue", xlab = "Number of dead eggs", ylab = "Probability of fungal infection")

points(DataPicked$FunIn[DataPicked$Treatment == 0] ~ DataPicked$Dead[DataPicked$Treatment == 0], col = "red")

lines(DataT0$Dead, P0, col = "red", lty = 1)

lines(DataT1$Dead, P1, col = "blue", lty = 1)

legend(200, 0.2, c("Not treated with formalin", "Treated with formalin"), lty = 1, col = c("red", "blue"))

par(op)

dev.off()

0.391361

-1.171124

0.004119

## Intercept, which is the probability to contract fungal infections within one week for untreated plots:

exp(0.391361+0)/(1+exp(0.391361+0))

# 0.5966103

## Estimating for treated eggs

exp(0.391361-1.171124)/(1+exp(0.391361-1.171124))

# 0.314371

# Chances at 0, 50, 100, 150, 200, 250, 300 dead eggs

exp(0.391361-1.171124+0.004119\*50)/(1+exp(0.391361-1.171124+0.004119\*50))

# [1] 0.3603575

exp(0.391361-1.171124+0.004119\*100)/(1+exp(0.391361-1.171124+0.004119\*100))

# [1] 0.4090575

exp(0.391361-1.171124+0.004119\*150)/(1+exp(0.391361-1.171124+0.004119\*150))

# [1] 0.4596099

exp(0.391361-1.171124+0.004119\*200)/(1+exp(0.391361-1.171124+0.004119\*200))

# [1] 0.5110075

exp(0.391361-1.171124+0.004119\*250)/(1+exp(0.391361-1.171124+0.004119\*250))

# [1] 0.5621733

exp(0.391361-1.171124+0.004119\*300)/(1+exp(0.391361-1.171124+0.004119\*300))

# [1] 0.6120499

#Not treated

exp(0.391361+0.004119\*50)/(1+exp(0.391361+0.004119\*50))

# [1] 0.6450409

exp(0.391361+0.004119\*100)/(1+exp(0.391361+0.004119\*100))

# [1] 0.6906716

exp(0.391361+0.004119\*150)/(1+exp(0.391361+0.004119\*150))

# [1] 0.7328657

exp(0.391361+0.004119\*200)/(1+exp(0.391361+0.004119\*200))

# [1] 0.7712109

exp(0.391361+0.004119\*250)/(1+exp(0.391361+0.004119\*250))

# [1] 0.8055125

exp(0.391361+0.004119\*300)/(1+exp(0.391361+0.004119\*300))

# [1] 0.8357666

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

 Model M4
#############################################################################

MFDPF <- gamm(FunIn ~ Treatment + s(dpf, k = 4), random = list(Location = ~ 1), family = binomial(link = "logit"), data = DataPicked, method = "REML")

summary(MFDPF$gam)

summary(MFDPF$lme)

##Intraclass correlation

(0.6410314)^2/(0.6410314^2+1^2)

#0.2912432

de2 <- 1 + (16 - 1) \* 0.2912432

de2

# 5.368648

Neff2 <- 15\*16/de2

Neff2

# 44.70399

# Not a vialble sample size, model over fitted

# Plotting the model

op <- par(mfrow = c(2,2), las = 1)

E1 <- resid(MFDPF$gam, type = "pearson")

F1 <- fitted(MFDPF$gam)

eta <- predict(MFDPF$gam, type = "link")

tiff(filename = "MFDPFResid.tiff", width = 6000, height = 6000, units = "px", pointsize = 18, res = 600)

op <- par(mfrow = c(2,2), las = 1)

plot(E1 ~ F1, xlab = "Fitted Values", ylab = "Pearson Residuals")

abline(h = 0, v = 0, lty = 2)

plot(E1 ~ eta, xlab = "Predicted Values", ylab = "Pearson Residuals")

abline(h = 0, v = 0, lty = 2)

plot(E1 ~ DataPicked$Treatment, xlab = "Treatment", ylab = "Pearson Residuals")

abline(h = 0, lty = 2)

plot(E1 ~ DataPicked$dpf, xlab = "dpf", ylab = "Pearson Residuals")

abline(h = 0, lty = 2)

dev.off()

pdNT <- (75 + 2)/(120+4)

pdT <- (42+2)/(120+4)

F035 <- subset(DataPicked, DataPicked$dpf == 35 & DataPicked$Treatment == 0 & DataPicked$FunIn == 1)

F042 <- subset(DataPicked, DataPicked$dpf == 42 & DataPicked$Treatment == 0 & DataPicked$FunIn == 1)

F049 <- subset(DataPicked, DataPicked$dpf == 49 & DataPicked$Treatment == 0 & DataPicked$FunIn == 1)

F056 <- subset(DataPicked, DataPicked$dpf == 56 & DataPicked$Treatment == 0 & DataPicked$FunIn == 1)

F063 <- subset(DataPicked, DataPicked$dpf == 63 & DataPicked$Treatment == 0 & DataPicked$FunIn == 1)

F070 <- subset(DataPicked, DataPicked$dpf == 70 & DataPicked$Treatment == 0 & DataPicked$FunIn == 1)

F077 <- subset(DataPicked, DataPicked$dpf == 77 & DataPicked$Treatment == 0 & DataPicked$FunIn == 1)

F084 <- subset(DataPicked, DataPicked$dpf == 84 & DataPicked$Treatment == 0 & DataPicked$FunIn == 1)

F135 <- subset(DataPicked, DataPicked$dpf == 35 & DataPicked$Treatment == 1 & DataPicked$FunIn == 1)

F142 <- subset(DataPicked, DataPicked$dpf == 42 & DataPicked$Treatment == 1 & DataPicked$FunIn == 1)

F149 <- subset(DataPicked, DataPicked$dpf == 49 & DataPicked$Treatment == 1 & DataPicked$FunIn == 1)

F156 <- subset(DataPicked, DataPicked$dpf == 56 & DataPicked$Treatment == 1 & DataPicked$FunIn == 1)

F163 <- subset(DataPicked, DataPicked$dpf == 63 & DataPicked$Treatment == 1 & DataPicked$FunIn == 1)

F170 <- subset(DataPicked, DataPicked$dpf == 70 & DataPicked$Treatment == 1 & DataPicked$FunIn == 1)

F177 <- subset(DataPicked, DataPicked$dpf == 77 & DataPicked$Treatment == 1 & DataPicked$FunIn == 1)

F184 <- subset(DataPicked, DataPicked$dpf == 84 & DataPicked$Treatment == 1 & DataPicked$FunIn == 1)

dpf <- c(35, 42, 49, 56, 63, 70, 77, 84)

T0 <- c(14, 12, 9, 11, 13, 13, 2, 1)

T1 <- c(3, 0, 3, 8, 7, 13, 7, 1)

CI <- data.frame(dpf, T0, T1)

CI$PS0 <- (CI$T0 + 2)/(15 + 4)

CI$PS1 <- (CI$T1 + 2)/(15 + 4)

CI$L0 <- CI$PS0 - 1.96\*sqrt((CI$PS0\*(1-CI$PS0))/(15 + 4))

CI$H0 <- CI$PS0 + 1.96\*sqrt((CI$PS0\*(1-CI$PS0))/(15 + 4))

CI$L1 <- CI$PS1 - 1.96\*sqrt((CI$PS1\*(1-CI$PS1))/(15 + 4))

CI$H1 <- CI$PS1 + 1.96\*sqrt((CI$PS1\*(1-CI$PS1))/(15 + 4))

CI[8, 6] = 0

CI[1, 7] = 1

CI[2, 8] = 0

CI[8, 8] = 0

tiff(filename = "FungusMFDPF.tiff", width = 6000, height = 6000, units = "px", pointsize = 18, res = 600)

op <- par(mfrow = c(1,1), las = 1)

DataT1 <- data.frame(dpf = seq(35, 84, length = 50), Treatment = 1)

DataT0 <- data.frame(dpf = seq(35, 84, length = 50), Treatment = 0)

P0 <- predict(MFDPF$gam, newdata = DataT0, type = "response")

P1 <- predict(MFDPF$gam, newdata = DataT1, type = "response")

plot(DataPicked$FunIn[DataPicked$Treatment == 0] ~ jitter(DataPicked$dpf[DataPicked$Treatment == 0], factor = 1), col = "red", xlab = "dpf", ylab = "Probability of fungal infection")

points(DataPicked$FunIn[DataPicked$Treatment == 1] ~ jitter(DataPicked$dpf[DataPicked$Treatment == 1], factor = 1), col = "blue")

lines(DataT0$dpf, P0, col = "red", lty = 1)

lines(DataT1$dpf, P1, col = "blue", lty = 1)

lines(CI$dpf, CI$L0, col = "red", lty = 2)

lines(CI$dpf, CI$H0, col = "red", lty = 2)

lines(CI$dpf, CI$L1, col = "blue", lty = 2)

lines(CI$dpf, CI$H1, col = "blue", lty = 2)

xlimit <- seq(35, 84, length = 8)

polygon(c(rev(xlimit), xlimit), c(rev(CI$L0), CI$H0), col = rgb(1, 0, 0, 0.2), border = NA)

polygon(c(rev(xlimit), xlimit), c(rev(CI$L1), CI$H1), col = rgb(0, 0, 1, 0.2), border = NA)

legend(55, 0.2, c("Not treated with formalin", "Treated with formalin"), lty = 1, col = c("red", "blue"))

par(op)

dev.off()

## Intercept, which is the probability to contract fungal infections within one week for untreated plots:

exp(0.5545+0)/(1+exp(0.5545+0))

# 0.635179

## Estimating for treated eggs

exp(0.5545-1.4791)/(1+exp(0.5545-1.4791))

# 0.2840215