Student Code

Student A

```
#upper anterior measurement Linear model
str(PADataNoOutlier)
str(PADataNoOutlierMultMeasure)
names(RPMA2Growth)
linearAnterior <- lm(PADataNoOutlier$Lipid ~ PADataNoOutlier$PSUA)
summary(linearAnterior)
linearAnterior
with(PADataNoOutlier, plot(Lipid ~ PSUA, las = 1, col = ifelse(PADataNoOutlier$`Fork Length`
abline(linearAnterior)

#Exponential function
expAnterior <- lm(PADataNoOutlier$Lipid ~ log(PADataNoOutlier$PSUA))
summary (expAnterior)
expAnterior
with(PADataNoOutlier, plot(Lipid ~ log(PSUA), las = 1,
     col = ifelse(PADataNoOutlier$`Fork Length` <  260, "red", "black")))
abline(expAnterior)
summary(expAnterior)

early <- subset(RPMA2Growth, StockYear < 2006)
mid <- subset(RPMA2Growth, StockYear < 2014 & StockYear > 2003)
RPMA2GrowthSub <- transform(RPMA2Growth, Age = as.integer(Age))
Early <- subset(RPMA2GrowthSub, StockYear < 2004)
Mid <- subset(RPMA2GrowthSub, StockYear < 2018 & StockYear > 2005)
EarlyWeightAge <- ddply(Early, ~Age, summarise, meanWE=mean(Weight, na.rm = T))
EarlyLengthAge <- ddply(Early, ~Age, summarise, meanLE = mean(ForkLength, na.rm = T))
MidLengthAge <- ddply(Mid, ~Age, summarise, meanLM = mean(ForkLength, na.rm = T))
WeightChange <- rep(NA, 9)

library(plyr)
```

```
WeightAge <- ddply(RPMA2GrowthSub, ~Age, summarise, meanW=mean(Weight, na.rm = T))
LengthAge <- ddply(RPMA2GrowthSub, ~Age, summarise, meanL=mean(ForkLength, na.rm = T))
plot(EarlyLengthAge$meanLE ~ EarlyLengthAge$Age, las = 1, ylab = "Fork Length (mm)", xlab = "
lines(EarlyLengthAge$meanLE ~ EarlyLengthAge$Age)
points(MidLengthAge$meanLM ~ MidLengthAge$Age, col = "red")
lines(MidLengthAge$meanLM ~ MidLengthAge$Age, col = "red")
legend(15, 600, legend = c("1998-2003", "2006-2017"), col = c("black", "red"), lty = 1:1, ce

#Tanner's code/help
WeightChange <- rep(NA, 9)
library(plyr)
WeightAge <- ddply(RPMA2GrowthSub, ~Age, summarise, meanW=mean(Weight, na.rm = T))
LengthAge <- ddply(RPMA2GrowthSub, ~Age, summarise, meanL=mean(ForkLength, na.rm = T))
plot(WeightAge$meanW ~ WeightAge$Age)
plot(LengthAge$mean ~ LengthAge$Age)
WeightChange

Weight1 <- mean(RPMA2GrowthSub$Weight[RPMA2GrowthSub$Age == 1], na.rm = TRUE)
Weight1
Length1 <- mean(RPMA2GrowthSub$ForkLength[RPMA2GrowthSub$Age == 1], na.rm  = TRUE)
Weight2 <- mean(RPMA2GrowthSub$Weight[RPMA2GrowthSub$Age == 2], na.rm = TRUE)
Length2 <- mean(RPMA2GrowthSub$ForkLength[RPMA2GrowthSub$Age == 2], na.rm  = TRUE)
Weight3 <- mean(RPMA2GrowthSub$Weight[RPMA2GrowthSub$Age == 3], na.rm = TRUE)
Length3 <- mean(RPMA2GrowthSub$ForkLength[RPMA2GrowthSub$Age == 3], na.rm  = TRUE)
Weight4 <- mean(RPMA2GrowthSub$Weight[RPMA2GrowthSub$Age == 4], na.rm = TRUE)
Length4 <- mean(RPMA2GrowthSub$ForkLength[RPMA2GrowthSub$Age == 4], na.rm  = TRUE)
Weight5 <- mean(RPMA2GrowthSub$Weight[RPMA2GrowthSub$Age == 5], na.rm = TRUE)
Length5 <- mean(RPMA2GrowthSub$ForkLength[RPMA2GrowthSub$Age == 5], na.rm  = TRUE)
Weight6 <- mean(RPMA2GrowthSub$Weight[RPMA2GrowthSub$Age == 6], na.rm = TRUE)
Length6 <- mean(RPMA2GrowthSub$ForkLength[RPMA2GrowthSub$Age == 6], na.rm  = TRUE)
Weight7 <- mean(RPMA2GrowthSub$Weight[RPMA2GrowthSub$Age == 7], na.rm = TRUE)
Length7 <- mean(RPMA2GrowthSub$ForkLength[RPMA2GrowthSub$Age == 7], na.rm  = TRUE)
Weight8 <- mean(RPMA2GrowthSub$Weight[RPMA2GrowthSub$Age == 8], na.rm = TRUE)
Length8 <- mean(RPMA2GrowthSub$ForkLength[RPMA2GrowthSub$Age == 8], na.rm  = TRUE)
Weight9 <- mean(RPMA2GrowthSub$Weight[RPMA2GrowthSub$Age == 9], na.rm = TRUE)
Length9 <- mean(RPMA2GrowthSub$ForkLength[RPMA2GrowthSub$Age == 9], na.rm  = TRUE)
x <- data.frame("Age" = 1:9, "Growth" = Weight1, Weight2, Weight3, Weight4, Weight5, Weight6
```

Student B

```
rm(list = ls())
source("./Gas_Functions.R")
```

```
# Load data ####
load("***REDACTED***/gas")
load("***REDACTED***/carboys")

gas <- gas[!(substr(gas$sampleID,3,3) %in% c("b","c")), ]
gas$days <- as.numeric(gas$minutesSinceAmendment/(24*60))

# Calculate molar fraction of N15-N2
RstN <- 0.003678
R <- ((gas$delN2/1000)+1)*RstN
gas$N15_MF <- R/(1+R)


# Calculate concentration of N15-N14 N2 relative to Argon
gas$N15_N2_Ar <- (gas$N15_MF * gas$N2Ar)*(40/28.014)

#mol N15-N2 per mol Ar
# Function to calculate likelihood of parameters given data ####

nmle <- function(P, t, y, N15_NO3_O){
  yhat <- N15_NO3_O * (1-exp(-P[1]*t))
  -sum(dnorm(y,yhat,exp(P[2]), log = T))
}


#### Carboy D ####

# Make vectors for time and N15-N2 observations

timeD <- (subset(gas, gas$carboy == "D"))$days
obsD <- subset(gas, gas$carboy == "D")$N15_N2_Ar
timeD <- timeD[!is.na(obsD)]
obsD <- obsD[!is.na(obsD)]

# Subtract off N15-N2 initially present in sample and set tracer N15-N2 to 0 at t=0
obsD <- obsD - obsD[1]

# Estimate Initial concentration of N15-NO3 relative to Ar

N15_NO3_O_D <- 40*((carboys[carboys$CarboyID == "D",]$EstN15NO3) + (0.7*RstN/(1+RstN)))/(subs

# Estimate fraction of labeled nitrate that gets denitrified
```

```
fracDenD = max(obsD)/N15_NO3_O_D

# Search for best parameters

mle.outD <- nlm(f = nmle, p = c(1,0.01), t = timeD, y = obsD, N15_NO3_O = N15_NO3_O_D*fracDen
ktotEst <- mle.outD$estimate[1]
kuEst <- ktotEst*(1-fracDenD )
kDenEst <- ktotEst*fracDenD

#per day

sigmaEst <- exp(mle.outD$estimate[2])

# Plot model with data

quartz(width = 4.5, height = 4)
par(mar = c(3.5,4,3,1))

predictionTimesD <- seq(0,max(timeD), length.out = 100)
predictionD <- fracDenD*N15_NO3_O_D*(1-exp(-ktotEst*predictionTimesD))

plot(x = predictionTimesD, y = predictionD,
     col = "blue", type = "l",
     xlab = "" ,
     ylab = "",
     ylim = c(0,0.08),
     main = "Mesocosm D",
     las = 1)
points(timeD, obsD, pch = 19)
title(ylab = expression(paste("Tracer "^15, N[2],":Ar")), line = 2.5, font.sub = 2)
title(xlab = "Time (days)", line = 2,font.sub =2)
legend("bottomright", legend = c("Modeled", "Measured"), lty = c("solid", NA), col = c("blue

# Calculate confidence interval

# Make matrix of parameter combinations

numRows <- 1000
kTotMin <- 2
kTotMax <- 60

pMat <- matrix(
```

```
              data = c(seq(kTotMin, kTotMax, length.out = numRows),
              rep(log(sigmaEst), times = numRows)),
              nrow = numRows)

likelihoods <- apply(X = pMat,
                     MARGIN = 1,
                     FUN = nmle,
                     t = timeD,
                     y = obsD,
                     N15_NO3_O = fracDenD*(N15_NO3_O_D)
)

mlle <- -min(likelihoods)
mlleIndex <- which.min(likelihoods)
mlleCI <- mlle - 1.96

lowerCIBound <- pMat[1:mlleIndex,1][which.min(abs(mlleCI+likelihoods[1:mlleIndex]))]
upperCIBound <- pMat[mlleIndex:length(likelihoods),1][which.min(abs(mlleCI+likelihoods[mlleIn

CI <- c(lowerCIBound, upperCIBound)
print(CI)

lowerCIBoundkDen <- lowerCIBound*fracDenD
upperCIBoundkDen <- upperCIBound*fracDenD


# Plot likelihoods with confidence intervals

quartz(width = 4.5, height = 4)

plot(x = seq(kTotMin,kTotMax, length.out = numRows),
     y = -likelihoods,
     type = "l",
     xlab = "ktot (per day)",
     ylab = "log(Likelihood)",
     las = 1)

abline(v = lowerCIBound, lty = 2, col = "blue")
abline(v = upperCIBound, lty = 2, col = "blue")


#### Carboy E ####
```

```
# Make vectors for time and N15-N2 observations

timeE <- (subset(gas, gas$carboy == "E"))$days
obsE <- subset(gas, gas$carboy == "E")$N15_N2_Ar
timeE <- timeE[!is.na(obsE)]
obsE <- obsE[!is.na(obsE)]


# Subtract off N15-N2 initially present in sample and set tracer N15-N2 to 0 at t=0

obsE <- obsE - obsE[1]

# Estimate Initial concentration of N15-NO3 relative to Ar

N15_NO3_0_E <- 40*((carboys[carboys$CarboyID == "E",]$EstN15NO3) + (0.7*RstN/(1+RstN)))/(subs

# Estimate fraction of labeled nitrate that gets denitrified

fracDenE = max(obsE)/N15_NO3_0_E

# Search for best parameters

mle.outE <- nlm(f = nmle, p = c(1,0.01), t = timeE, y = obsE, N15_NO3_0 = N15_NO3_0_E*fracDen

ktotEst <- mle.outE$estimate[1]

#per day

kuEst <- ktotEst*(1-fracDenE )
kDenEst <- ktotEst*fracDenE

#per day

sigmaEst <- exp(mle.outE$estimate[2])

# Plot model with data

quartz(width = 4.5, height = 4)
par(mar = c(3.5,4,3,1))

predictionTimesE <- seq(0,max(timeE), length.out = 100)
predictionE <- fracDenE*N15_NO3_0_E*(1-exp(-ktotEst*predictionTimesE))
plot(x = predictionTimesE, y = predictionE,
```

```
     col = "blue", type = "l",
     xlab = "" ,
     ylab = "",
     ylim = c(0,0.08),
     main = "Mesocosm E",
     las = 1)
points(timeE, obsE, pch = 19)
title(ylab = expression(paste("Tracer "^15, N[2],":Ar")), line = 2.5, font.sub = 2)
title(xlab = "Time (days)", line = 2,font.sub =2)
legend("bottomright", legend = c("Modeled", "Measured"), lty = c("solid", NA), col = c("blue

# Calculate confidence interval
# Make matrix of parameter combinations

numRows <- 1000
kTotMin <- 2
kTotMax <- 10

pMat <- matrix(data = c(seq(kTotMin, kTotMax, length.out = numRows),
                        rep(log(sigmaEst), times = numRows)),
               nrow = numRows)

likelihoods <- apply(X = pMat, MARGIN = 1, FUN = nmle,
                     t = timeE, y = obsE,  N15_NO3_0 = fracDenE*(N15_NO3_0_E))

mlle <- -min(likelihoods)
mlleIndex <- which.min(likelihoods)
mlleCI <- mlle - 1.96

lowerCIBound <- pMat[1:mlleIndex,1][which.min(abs(mlleCI+likelihoods[1:mlleIndex]))]

upperCIBound <- pMat[mlleIndex:length(likelihoods),1][which.min(abs(mlleCI+likelihoods[mlleIn

CI <- c(lowerCIBound, upperCIBound)
print(CI)

lowerCIBoundkDen <- lowerCIBound*fracDenE
upperCIBoundkDen <- upperCIBound*fracDenE

# Plot likelihoods with confidence intervals

quartz(width = 4.5, height = 4)
```

```
plot(x = seq(kTotMin,kTotMax, length.out = numRows),
     y = -likelihoods,
     type = "l",
     xlab = "ktot (per day)",
     ylab = "log(Likelihood)",
     las = 1)
abline(v = lowerCIBound, lty = 2, col = "blue")
abline(v = upperCIBound, lty = 2, col = "blue")


#### Carboy F ####

# Make vectors for time and N15-N2 observations

timeF <- (subset(gas, gas$carboy == "F"))$days
obsF <- subset(gas, gas$carboy == "F")$N15_N2_Ar
timeF <- timeF[!is.na(obsF)]
obsF <- obsF[!is.na(obsF)]

# Subtract off N15-N2 initially present in sample and set tracer N15-N2 to 0 at t=0

obsF <- obsF - obsF[1]

# Estimate Initial concentration of N15-NO3 relative to Ar

N15_NO3_O_F <- 40*((carboys[carboys$CarboyID == "F",]$EstN15NO3) + (0.7*RstN/(1+RstN)))/(subs

# Estimate fraction of labeled nitrate that gets denitrified

fracDenF = max(obsF)/N15_NO3_O_F

# Search for best parameters

mle.outF <- nlm(f = nmle, p = c(1,0.01), t = timeF, y = obsF, N15_NO3_O = N15_NO3_O_F*fracDen
ktotEst <- mle.outF$estimate[1]

#per day

kuEst <- ktotEst*(1-fracDenF )
kDenEst <- ktotEst*fracDenF

#per day
```

```
sigmaEst <- exp(mle.outE$estimate[2])

# Plot model with data

quartz(width = 4.5, height = 4)
par(mar = c(3.5,4,3,1))

predictionTimesF <- seq(0,max(timeF), length.out = 100)
predictionF <- fracDenF*N15_NO3_O_F*(1-exp(-ktotEst*predictionTimesF))

plot(x = predictionTimesF, y = predictionF,
     col = "blue", type = "l",
     xlab = "" ,
     ylab = "",
     ylim = c(0,0.08),
     main = "Mesocosm F",
     las = 1)
points(timeF, obsF, pch = 19)
title(ylab = expression(paste("Tracer "^15, N[2],":Ar")), line = 2.5, font.sub = 2)
title(xlab = "Time (days)", line = 2,font.sub =2)
legend("bottomright", legend = c("Modeled", "Measured"), lty = c("solid", NA), col = c("blue'

# Calculate confidence interval

# Make matrix of parameter combinations

numRows <- 1000
kTotMin <- 2
kTotMax <- 5

pMat <- matrix(data = c(seq(kTotMin, kTotMax, length.out = numRows),
                        rep(log(sigmaEst), times = numRows)),
               nrow = numRows)

likelihoods <- apply(X = pMat, MARGIN = 1, FUN = nmle,
                     t = timeF, y = obsF,  N15_NO3_O = fracDenF*(N15_NO3_O_F))

mlle <- -min(likelihoods)
mlleIndex <- which.min(likelihoods)
mlleCI <- mlle - 1.96

lowerCIBound <- pMat[1:mlleIndex,1][which.min(abs(mlleCI+likelihoods[1:mlleIndex]))]
upperCIBound <- pMat[mlleIndex:length(likelihoods),1][which.min(abs(mlleCI+likelihoods[mlleIr
```

```
CI <- c(lowerCIBound, upperCIBound)
print(CI)

lowerCIBoundkDen <- lowerCIBound*fracDenF
upperCIBoundkDen <- upperCIBound*fracDenF

# Plot likelihoods with confidence intervals

quartz(width = 4.5, height = 4)

plot(x = seq(kTotMin,kTotMax, length.out = numRows),
     y = -likelihoods,
     type = "l",
     xlab = "ktot (per day)",
     ylab = "log(Likelihood)",
     las = 1)
abline(v = lowerCIBound, lty = 2, col = "blue")
abline(v = upperCIBound, lty = 2, col = "blue")
```