Stat\_consulting Rcode-4-corrected,

code and errors

**After running lines 330 t0 336- this is the result—which is okay I think**

NumberOfOvulation Avg.Time\_BW\_Ovu Median.Time\_BW\_Ovu Std\_Deviation Pvalue

1 32 11.29 5 14.4 0.21

To determine the strength of the distribution of the data we run a KS test

from fitting an exponential distribution to the observed data

One-sample Kolmogorov-Smirnov test

data: time\_between\_peaks

D = 0.19059, p-value = 0.2101

alternative hypothesis: two-sided

The p-value > 0.05 suggests that the spontaneous ovulation events

are occuring at regular intervals following an exponential distribution

with rate parameter = 0.09

One-sample Kolmogorov-Smirnov test

data: time\_between\_peaks

D = 0.19059, p-value = 0.2101

alternative hypothesis: two-sided

**Warning messages:**

**1: In ks.test(time\_between\_peaks, "pexp", fit1$estimate) :**

**ties should not be present for the Kolmogorov-Smirnov test**

**2: In ks.test(time\_between\_peaks, "pexp", fit1$estimate) :**

**ties should not be present for the Kolmogorov-Smirnov test**

**3: In ks.test(time\_between\_peaks, "pexp", fit1$estimate) :**

**ties should not be present for the Kolmogorov-Smirnov test**

results of looping each female

**analyzing female m**

**NumberOfOvulation Avg.Time\_BW\_Ovu Median.Time\_BW\_Ovu Std\_Deviation Pvalue**

**1 14 9.38 9 6.97 0.72**

**To determine the strength of the distribution of the data we run a KS test**

**from fitting an exponential distribution to the observed data**

**One-sample Kolmogorov-Smirnov test**

**data: time\_between\_peaks**

**D = 0.19194, p-value = 0.7244**

**alternative hypothesis: two-sided**

**The p-value > 0.05 suggests that the spontaneous ovulation events**

**are occuring at regular intervals following an exponential distribution**

**with rate parameter = 0.11**

**One-sample Kolmogorov-Smirnov test**

**data: time\_between\_peaks**

**D = 0.19194, p-value = 0.7244**

**alternative hypothesis: two-sided**

**[1] "insufficient data to run KS Test"**

**NumberOfOvulation Avg.Time\_BW\_Ovu Median.Time\_BW\_Ovu Std\_Deviation Pvalue**

**2015 14 9.38 9 6.97 0.72**

**2016 2 17.00 17 NA NA**

**analyzing female p**

**NumberOfOvulation Avg.Time\_BW\_Ovu Median.Time\_BW\_Ovu Std\_Deviation Pvalue**

**1 9 10 5 8.96 0.78**

**To determine the strength of the distribution of the data we run a KS test**

**from fitting an exponential distribution to the observed data**

**One-sample Kolmogorov-Smirnov test**

**data: time\_between\_peaks**

**D = 0.23153, p-value = 0.7844**

**alternative hypothesis: two-sided**

**The p-value > 0.05 suggests that the spontaneous ovulation events**

**are occuring at regular intervals following an exponential distribution**

**with rate parameter = 0.1**

**One-sample Kolmogorov-Smirnov test**

**data: time\_between\_peaks**

**D = 0.23153, p-value = 0.7844**

**alternative hypothesis: two-sided**

**Error in seq.int(1L, length(y), 2L) : wrong sign in 'by' argument**

**In addition: Warning messages:**

**1: In ks.test(time\_between\_peaks, "pexp", fit1$estimate) :**

**ties should not be present for the Kolmogorov-Smirnov test**

**2: In ks.test(time\_between\_peaks, "pexp", fit1$estimate) :**

**ties should not be present for the Kolmogorov-Smirnov test**

**3: In ks.test(time\_between\_peaks, "pexp", fit1$estimate) :**

**ties should not be present for the Kolmogorov-Smirnov test**

**4: In ks.test(time\_between\_peaks, "pexp", fit1$estimate) :**

**ties should not be present for the Kolmogorov-Smirnov test**

**5: In ks.test(time\_between\_peaks, "pexp", fit1$estimate) :**

**ties should not be present for the Kolmogorov-Smirnov test**

**6: In ks.test(time\_between\_peaks, "pexp", fit1$estimate) :**

**ties should not be present for the Kolmogorov-Smirnov test**

**RStudioGD**

**2**

I am not sure what this text above in the console means, but the year analysis graph does not appearance is not as it should look

jpeg('polyestry stats/yearlyanalysis.jpg',width = 600, height = 800)

years\_plot(days\_lag = days\_lag ,

cutoff\_Estradiol = cutoff\_Estradiol,

cutoff\_Progesterone = cutoff\_Progesterone,

up\_pro = 20,

bot\_pro = 10,

up\_est = 200,

bot\_est = 20)

dev.off()

Current Code

## Remove any existing dataset

## or analysis from your

## workspace

rm(list=ls())

#options(warn=-1)

### Load libraries

require(data.table)

require(lubridate)

require(ggplot2)

### Change your directory

#setwd("C:/Users/stuwe/OneDrive - University of Arizona/Desktop/MGRS stats results/Stat analysis from stat lab")

### Read different squirrel file"

#Start multifemale approach

#readdata

squirrel <- fread("all\_females\_a.csv",data.table = F,)

#squirrel <- fread( "Physiology only stat files/mphys\_ra.csv",

# data.table = F)

#squirrel<- read.csv(file="mphys\_ra.csv")

### manipulate date

squirrel$date <- as.Date(squirrel$date,format = '%m/%d/%Y')

squirrel$year <- year(squirrel$date)

squirrel$months <- month(squirrel$date, label = T)

squirrel$day <- day(squirrel$date)

## Log Transformation

names(squirrel)[c(3,4)] <- c('Progesterone','Estradiol')

squirrel$Estradiol <- log(squirrel$Estradiol)

squirrel$Progesterone <- log(squirrel$Progesterone)

### Split year to year

year2015 = squirrel[squirrel$year == 2015, -5]

year2016 = squirrel[squirrel$year == 2016, -5]

#year2017 = squirrel[squirrel$year == 2017, -6]

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

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

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

##

##

## Functions to analyze Data

##

*##################################*

*## localMaxima : identifies peaks ## in the data—Original Code*

*##################################*

localMaxima <- **function**(x) {

*# Use -Inf instead if x is numeric (non-integer)* y <- **diff**(**c**(**-**.Machine**$**integer.max, x)) **>** 0L **rle**(y)**$**lengths

y <- **cumsum**(**rle**(y)**$**lengths)

y <- y[**seq.int**(1L, **length**(y), 2L)]

**if** (x[[1]] **==** x[[2]]) { y <- y[**-**1]

}

y

}

*##################################*

*## localMaxima : identifies peaks ## in the data—Updated Code*

*##################################*

localMaxima <- function(x) {

# Use -Inf instead if x is numeric (non-integer)

y <- diff(c(-.Machine$integer.max, x)) > 0L

rle(y)$lengths

y <- cumsum(rle(y)$lengths)

y <- y[seq.int(1L, length(y), 2L)]

if (x[[1]] == x[[2]]) {

y <- y[-1]

}

y

}

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

## analyze\_yearly\_trends :

## - compares yearly estradiol

## and progesterone values

## - plots peaks and events

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

analyze\_yearly\_trends <- function(m1, lag=0, cutoff\_Estradiol=390,

cutoff\_Progesterone=5, year='2015',

up\_pro=30,

bot\_pro=10,

up\_est=400,

bot\_est=20){

*### Detect events—Original code*

estra\_max <- **ric**(**nrow**(m1))

estra\_max[**localMaxima**(m1**$**Estradiol)] <- 1

proge\_max <- **numeric**(**nrow**(m1)) proge\_max[**localMaxima**(m1**$**Progesterone)] <- 1

event <- **numeric**(**nrow**(m1)) **for**(i **in** 1**:length**(event)){

**if**(estra\_max[i]**==**1 **&** proge\_max[i**+**lag]**==**1){ event[i] <- 1

} **else if**(estra\_max[i]**==**1 **&** proge\_max[i]**==**1){ event[i] <- 1

}

}

max\_mat <- **data.frame**(estra\_max, proge\_max, event)

### Detect events -updated code

estra\_max <- numeric(nrow(m1))

estra\_max[localMaxima(m1$Estradiol)] <- 1

estra\_max[which(m1$Estradiol < log(cutoff\_Estradiol))] <- 0

proge\_max <- numeric(nrow(m1))

proge\_max[localMaxima(m1$Progesterone)] <- 1

proge\_max[which(m1$Progesterone < log(cutoff\_Progesterone))] <- 0

event <- numeric(nrow(m1))

for(i in 1:length(event)){

if( estra\_max[i]==1 & sum(proge\_max[i:(i+lag)])>0 &

day(m1$date[i+lag]) - day(m1$date[i]) <= lag){

#event[i:(i+lag)] <- 1

event[i] <- 1

}

}

max\_mat <- data.frame(estra\_max, proge\_max, event)

Reply from Jeff

The detection of events (in analyze\_yearly\_trends) seems to identify not only the maxima, and the updated version also detect values for estradiol & progesterone that are below some threshold:

estra\_max[which(m1$Estradiol < log(cutoff\_Estradiol))] <- 0 ...

proge\_max[which(m1$Progesterone < log(cutoff\_Progesterone))] <- 0

The identification of actual events is subtly different between the original and updated versions. The original version categorizes events as those rows where the estradiol was max and progesterone was max in the same row OR in the row + lag (i.e. if lag = 2, a row is categorized as an event if progesterone was max in the same row as max estradiol or two rows after the max estradiol row). Note this would mean that if progesterone peaks one row after max estradiol, it would not be categorized as an event. The updated code is more flexible, where an event is identified when progesterone peaks at the same time as estradiol, or within a time period defined by lag. In this updated code, if lag = 2, an event would occur if estradiol and progesterone are max in the same row or progesterone is max one or two rows after estradiol is max.

In regards to translating plots to ggplot, those are some pretty complex plots in the analyze\_yearly\_trends function. The way I would go about this would be to start with the based plot in analyze\_yearly\_trends, get that working, then add points & lines to make it match the previous plots.

For example, in the analyze\_yearly\_trends, the plot starts out with:

plot(m1$date, m1$Progesterone, type='l', ylim=c(4,8), lwd=2, col='green',

main=paste('Progesterone ~ Estradiol',year,sep=':'), xlab='Day',ylab='Log Concentration')

A very rough translation to ggplot would be:

ggplot(data = m1, mapping = aes(x = date, y = Progesterone)) +

ylim(c(4, 8)) +

geom\_line(lwd = 2, color = "green") +

ggtitle(label = paste("Progesterone ~ Estradiol", year, sep = ":")) +

xlab(label = "Day") +

ylab(label = "Log Concentration")

In regards to sharing code, GitHub is probably the easiest way - just point me to the file on GitHub that you are working on.

Cheers,

Jeff

More insight from Jeff

Yeah, it looks like the problem lies somewhere in that function. When running these functions, I use the cat function to print out values to see if things are working the way I think they are. You can try adding the highlighted lines below to your code and see what prints out.

for(i in 1:length(event)){

date\_search = min(i+lag, length(event))

cat("Current row:", i, "\n")

cat("date\_search:", date\_search, "\n")

cat("lag:", lag, "\n")

cat("First day:", day(m1$date[i]), "\n")

cat("Last day:", day(m1$date[date\_search]), "\n"))

cat("Estra max:", estra\_max[i], "\n")

cat("Progesterone sum:", sum(proge\_max[i:(i+lag)]), "\n")

if(estra\_max[i]==1 & sum(proge\_max[i:(i+lag)])>0 &

day(m1$date[date\_search]) - day(m1$date[i]) <= lag){

When you run the code, there should be some value printed out for each of them. e.g. you should see something like:

Current row: 2

date\_search: 4

lag: 2

### Plot all events

plot(m1$date, m1$Progesterone, type='l', ylim=c(4,8), lwd=2, col='green',

main=paste('Progesterone ~ Estradiol',year,sep=':'), xlab='Day',ylab='Log Concentration')

lines(m1$date, m1$Estradiol, type='l', lwd=2, col='blue')

event <- m1$Estradiol \* max\_mat$event

event2 <- m1$Progesterone \* max\_mat$event

points(m1$date, event, col='red', lwd=5)

points(m1$date, event2, col='red',lwd=5)

segments(x0 =m1$date , y0 = event2, x1 = m1$date, y1=event, lwd=2, lty=2, col='red' )

legend('topright', c('Progesterone','Estradiol','Ovulation'),

col=c('blue','green','red'), lwd=2, pch=c(1,1,1), cex=0.8)

abline(h=log(up\_est), lty=2, lwd=2)

abline(h=log(bot\_est), lty=2, lwd=2)

abline(h=log(up\_pro), lty=2, col='brown', lwd=2)

abline(h=log(bot\_pro), lty=2, col='brown', lwd=2)

legend('topleft', c(paste('Estr', c(up\_est,bot\_est),sep='=')), lty=2, col='black', lwd=2)

legend('bottomleft',c(paste('Proge', c(up\_pro,bot\_pro),sep='=')), lty=2, col='brown', lwd=2)

}

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

## analyze\_monthly\_trends :

## - compares monthly estradiol

## and progesterone values

## - plots peaks and events

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

analyze\_monthly\_trends <- function(m1, lag=0, cutoff\_Estradiol=150,

cutoff\_Progesterone=20){

### Detect events -a third line of code was added to detect progesterone for two days

estra\_max <- numeric(nrow(m1))

estra\_max[localMaxima(m1$Estradiol)] <- 1

estra\_max[which(m1$Estradiol < log(cutoff\_Estradiol))] <- 0

proge\_max <- numeric(nrow(m1))

proge\_max[localMaxima(m1$Progesterone)] <- 1

proge\_max[which(m1$Progesterone < log(cutoff\_Progesterone))] <- 0

event <- numeric(nrow(m1))

#cat("starting vaule of event",event,"\n") debugging code

df = data.frame(estra\_max,

proge\_max,

m1$date)

for(i in 1:length(event)){

date\_search = min(i+lag, length(event))

if( estra\_max[i]==1 & sum(proge\_max[i:(i+lag)])>0 &

day(m1$date[date\_search]) - day(m1$date[i]) <= lag){

event[i] <- 1

#event[i:(i+lag)] <- 1

}

}

df$event <- event

event <- m1$Estradiol \* event

# cat("event after multiplication",event,"\n") debugging code

max\_mat <- data.frame(estra\_max, proge\_max, event)

### Do monthly panel plots

m1$event <- event

#newM1 = melt(m1,id.vars = c('id','day','date','months','year'))

newM1 <- tidyr::pivot\_longer(data=m1,cols=-c('ID','day','date','months','year'),

names\_to="variable",values\_to ="value")

p1 = ggplot(newM1, aes(x = day, value, col=variable)) + geom\_line() + facet\_grid(months~.) +

theme\_bw() + theme(

plot.title = element\_text(color="black", size=24, face="bold", hjust = 0.5),

axis.title.x = element\_text(color="black", size=18, face="bold"),

axis.title.y = element\_text(color="black", size=18, face="bold"),

axis.text.x = element\_text(color="black", size=12, face="bold")) +

ggtitle('Ovulation') + xlab('date') + ylab('Log Concentration')

### Look at time between peaks

time\_between\_peaks <- diff(which(event > 0 ))

# cat("about to look at event...\n") debugging code

if(sum(event>0,na.rm=TRUE) < 5){

print('insufficient data to run KS Test')

} else{

print(summary\_peaks(time\_between\_peaks))

print(test\_between\_peaks\_distribution(time\_between\_peaks))

fit1 <- fitdistr(time\_between\_peaks, "exponential")

#

# true\_Exp <- rexp(10000, fit1$estimate)

#

# plot(ecdf(time\_between\_peaks), xlim=range(c(time\_between\_peaks, true\_Exp)), col="dodgerblue", main='KS Test Visualization',

# ylab=paste("Wait-time b/w Events CDF"), xlab= "")

# plot(ecdf(true\_Exp), add=TRUE, lty="dashed", col="purple", ylab="", xlab="")

# legend("right", legend=c("2016 observed", "Exponential Dist'n"), col=c("dodgerblue", "purple"),

# lty="dashed", lwd=2 )

#

# p2 <- recordPlot()

}

print(p1)

return(list(summary\_peaks(time\_between\_peaks),

time\_between\_peaks,

df))

}

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

## summary\_peaks :

## - provides tabular summary of

## peaks

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

summary\_peaks <- function(time\_between\_peaks){

numPeaks <- length(time\_between\_peaks)+1

AvgLength<- mean(time\_between\_peaks)

AvgStd <- sqrt(var(time\_between\_peaks))

if(numPeaks < 5){

Pvalue <- NA\_real\_

} else{

require(MASS)

fit1 <- fitdistr(time\_between\_peaks, "exponential")

KS.res <- ks.test(time\_between\_peaks, "pexp", fit1$estimate)

Pvalue = KS.res$p.value

}

summary\_stats <- data.frame(NumberOfOvulation = numPeaks,

Avg.Time\_BW\_Ovu = AvgLength,

Median.Time\_BW\_Ovu = median(time\_between\_peaks),

# Max\_Time\_Between\_Peaks = max(time\_between\_peaks),

# Min\_Time\_Between\_Peaks = min(time\_between\_peaks),

Std\_Deviation = AvgStd,

Pvalue=Pvalue)

return(round(summary\_stats,2))

}

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

## test\_between\_peaks\_distribution :

## - conducts KS test

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

test\_between\_peaks\_distribution <- function(time\_between\_peaks){

cat('\nTo determine the strength of the distribution of the data we run a KS test

from fitting an exponential distribution to the observed data\n')

require(MASS)

fit1 <- fitdistr(time\_between\_peaks, "exponential")

KS.res <- ks.test(time\_between\_peaks, "pexp", fit1$estimate)

print(KS.res)

if(KS.res$p.value > 0.05){

cat(paste('The p-value > 0.05 suggests that the spontaneous ovulation events

are occuring at regular intervals following an exponential distribution

with rate parameter =', round(fit1$estimate,2)))

} else {

cat('The p-value < 0.05 suggests that the spontaneous ovulation events are not

occuring at regular intervals, and thus do not follow an exponential distribution

of equal-peak intervals')

}

return(KS.res)

}

### Yearly summary

years\_plot <- function(days\_lag, cutoff\_Estradiol, cutoff\_Progesterone,

up\_pro = 30,

bot\_pro = 10,

up\_est = 400,

bot\_est = 20){

par(mfrow=c(2,1), mar=c(5,5,5,5))

analyze\_yearly\_trends(m1=year2015, lag = days\_lag,

cutoff\_Estradiol = cutoff\_Estradiol,

cutoff\_Progesterone = cutoff\_Progesterone,

year='2015',

up\_pro = up\_pro,

bot\_pro = bot\_pro,

up\_est = up\_est,

bot\_est = bot\_est)

analyze\_yearly\_trends(m1=year2016, lag = days\_lag,

cutoff\_Estradiol = cutoff\_Estradiol,

cutoff\_Progesterone = cutoff\_Progesterone,

year='2016',

up\_pro = up\_pro,

bot\_pro = bot\_pro,

up\_est = up\_est,

bot\_est = bot\_est)

#analyze\_yearly\_trends(m1=year2017, lag = days\_lag,

# cutoff\_Estradiol = cutoff\_Estradiol,

## year='2017',

# up\_pro = up\_pro,

# bot\_pro = bot\_pro,

# up\_est = up\_est,

# bot\_est = bot\_est)

}

##

##

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

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

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

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

##

## Analyze Results

##

### Set custom parameters

cutoff\_Estradiol = 100

cutoff\_Progesterone = 20

days\_lag = 0

## Analyze squirrel data using

## Monthly Panels + KS Test

#jpeg('C:/Users/stuwe/Desktop/Stat analysis from stat lab/monthy\_ovulation2015.jpeg')

time\_between\_peaks2015 <- analyze\_monthly\_trends(m1 = year2015, lag = days\_lag,

cutoff\_Estradiol = cutoff\_Estradiol,

cutoff\_Progesterone = cutoff\_Progesterone)

#dev.off()

#jpeg('C:/Users/stuwe/Desktop/Stat analysis from stat lab/monthy\_ovulation2016.jpeg')

time\_between\_peaks2016 <- analyze\_monthly\_trends(m1= year2016, lag=days\_lag,

cutoff\_Estradiol = cutoff\_Estradiol,

cutoff\_Progesterone = cutoff\_Progesterone)

#time\_between\_peaks2017 <- analyze\_monthly\_trends(m1=year2017, lag=days\_lag,

# cutoff\_Estradiol = cutoff\_Estradiol,

#cutoff\_Progesterone = cutoff\_Progesterone)

#dev.off()

peakSummary = data.frame(rbind(time\_between\_peaks2015[[1]],

time\_between\_peaks2016[[1]]), row.names = c('2015','2016'))

print(peakSummary)

write.csv(peakSummary, file ='polyestry stats/ovulation\_summary.csv')

## Analyze squirrel data using

## Yearly analysis

jpeg('polyestry stats/yearlyanalysis.jpg',width = 600, height = 800)

years\_plot(days\_lag = days\_lag ,

cutoff\_Estradiol = cutoff\_Estradiol,

cutoff\_Progesterone = cutoff\_Progesterone,

up\_pro = 20,

bot\_pro = 10,

up\_est = 200,

bot\_est = 20)

dev.off()

#Start multifemale approach

#readdata

#loop over each female

female\_ID <- unique(squirrel$ID)

for (one\_ID in female\_ID) {

#do all the stuff here

cat("analyzing female",one\_ID,"\n")

#subsetdata for this female

one\_female <- squirrel[squirrel$ID == one\_ID,-5]

#subset 2015 and 2016 data

year\_2015 <- one\_female[one\_female$year == 2015,]

year\_2016 <- one\_female[one\_female$year == 2016,]

# year\_2016 <- one\_female[one\_female$year == 2017,]

#analyze yearly trends

time\_between\_peaks2015 <- analyze\_monthly\_trends(m1 = year\_2015, lag = days\_lag,

cutoff\_Estradiol = cutoff\_Estradiol,

cutoff\_Progesterone = cutoff\_Progesterone)

time\_between\_peaks2016 <- analyze\_monthly\_trends(m1=year\_2016, lag=days\_lag,

cutoff\_Estradiol = cutoff\_Estradiol,

cutoff\_Progesterone = cutoff\_Progesterone)

# time\_between\_peaks2017 <- analyze\_monthly\_trends(m1=year\_2017, lag=days\_lag,

# cutoff\_Estradiol = cutoff\_Estradiol,

# cutoff\_Progesterone = cutoff\_Progesterone)

peakSummary = data.frame(rbind(time\_between\_peaks2015[[1]],

time\_between\_peaks2016[[1]]), row.names = c('2015','2016'))

print(peakSummary)

#save output files

ovulation\_file<-paste0("polyestry stats/ovulation\_summary.csv",one\_ID,".csv")

write.csv(peakSummary, file= ovulation\_file)

## Analyze squirrel data using

## Yearly analysis

comparison\_file <- paste0("polyestry stats/monthlysummary",one\_ID,".jpeg")

jpeg(comparison\_file ,width = 600, height= 800)

years\_plot(days\_lag = days\_lag ,

cutoff\_Estradiol = cutoff\_Estradiol,

cutoff\_Progesterone = cutoff\_Progesterone,

up\_pro = 50,

bot\_pro =10,

up\_est = 200,

bot\_est = 20)

dev.off()

}