### Load and clean Data

require(data.table)

require(lubridate)

require(ggplot2)

setwd("C:/Users/stuwe/Desktop/Stat analysis from stat lab")

squirrel <- fread('sphys\_ra528 -ovulation excel data.csv',data.table = F)

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)

names(squirrel)[c(4,6)] <- c('Estradiol','Progesterone')

# Data Exploration and Statistical Analysis

## Log Transformations

# plot(log(squirrel$Progesterone), type='l', ylim=c(4,9), lwd=2, col='green',

# main='Estradiol ~ Progesterone\nAfter Log Transformation', xlab='Day',ylab='Log Concentration')

# lines(log(squirrel$Estradiol), type='l', lwd=2, col='blue')

#

#

# par(mfrow=c(1,2))

# hist(squirrel$Progesterone, main='Progesterone', xlab='Progesterone', 20)

# hist(log(squirrel$Progesterone), main='Log (Progesterone)', xlab='log(Progesterone)', 20)

squirrel$Estradiol <- log(squirrel$Estradiol)

squirrel$Progesterone <- log(squirrel$Progesterone)

### Split year to year

m1 = squirrel[squirrel$year == 2015, -5]

m2 = squirrel[squirrel$year == 2016, -5]

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

## Functions to analyze Data

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

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

## localMaxima : identifies peaks

## in the data

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

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=2000,

cutoff\_progesterone=400){

### Detect events

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 & 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)

### Plot all events

plot(m1$date, m1$Progesterone, type='l', ylim=c(4,9), lwd=2, col='green',

main='Estradiol ~ Progesterone', 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('Estradiol','Progesterone','Ovulation'),

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

}

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

## analyze\_monthly\_trends :

## - compares monthly estradiol

## and progesterone values

## - plots peaks and events

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

analyze\_monthly\_trends <- function(m1, lag=0, cutoff\_estradion=2000,

cutoff\_progesterone=400){

### Detect events

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

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

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

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

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

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

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

}

}

event <- m1$Estradiol \* event

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'))

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

theme\_bw() + theme(

plot.title = element\_text(color="black", size=26, face="bold", hjust = 0.5),

axis.title.x = element\_text(color="black", size=20, face="bold"),

axis.title.y = element\_text(color="black", size=20, face="bold"),

axis.text.x = element\_text(color="black", size=14, face="bold")) +

ggtitle('Monthly Log Concentrations') + xlab('Date') + ylab('Log Concentration')

print(p)

### Look at time between peaks

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

return(time\_between\_peaks)

}

require(knitr)

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

## 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))

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

Mean\_Time = AvgLength,

Median\_Time = median(time\_between\_peaks),

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

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

Std\_Deviation = AvgStd)

return(summary\_stats)

}

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

## 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')

}

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 )

return(KS.res)

}

# Results

## 2015

### Identifying Candidate Ovulation Events

analyze\_yearly\_trends(m1=m1, lag = 0,cutoff\_estradiol = 1500,

cutoff\_progesterone = 200)

analyze\_yearly\_trends(m1=m2, lag = 0,cutoff\_estradiol = 1500,

cutoff\_progesterone = 200)

### Monthly Examination and KS Test

a = analyze\_monthly\_trends(m1=m1, lag = 0,cutoff\_estradiol = 1500,

cutoff\_progesterone = 200)

p1 = test\_between\_peaks\_distribution(a)

## 2016

### Identifying Candidate Ovulation Events

analyze\_yearly\_trends(m1=m1, lag = 0,cutoff\_estradiol = 1500,

cutoff\_progesterone = 200)

analyze\_yearly\_trends(m1=m2, lag = 0,cutoff\_estradiol = 1500,

cutoff\_progesterone = 200)

### Monthly Examination and KS Test

b = analyze\_monthly\_trends(m1=m2, lag = 0)

p2 = test\_between\_peaks\_distribution(b)