2016 Burnham & Lehman Queen Experirment

P. Alexander Burnham

September 14, 2016

Metadata:

Author: P. Alexander Burnham

Date: 10 August 2016

Data Set: These data were collected in 2016 in New York state by Andre Burnham and the Hamilton College bee research team from two yards in NY owned by one beekeeper who volunteered for the study.

Data Source: 2016 Hamilton Bee Research Project

Funding Source: Hamilton College undergraduate research grants

Data Collection: Weights of colonies were obtained by summing the weights of individual hive bodies and supers for each colony at each yard (add more about protocols for other assays).

Columns: (from left to right) Field ID: Id including number location and treatment, Mass (biomass minus supers in lbs) Time (T1-T4), Origin (California and Local), Yard (1 or 2),Nosema (spores per bee) only two time steps (T1 and T3),Varroa (mites per 300 bees),Brood (frames of brood in colony),NosemaPA (binary data for nosema),VarroaPA (binary data for mites), Rows: Data points for all columns in order from each collection event

Missing values: NA

```
#Preliminaries:

ls()
```

```
## character(0)
```

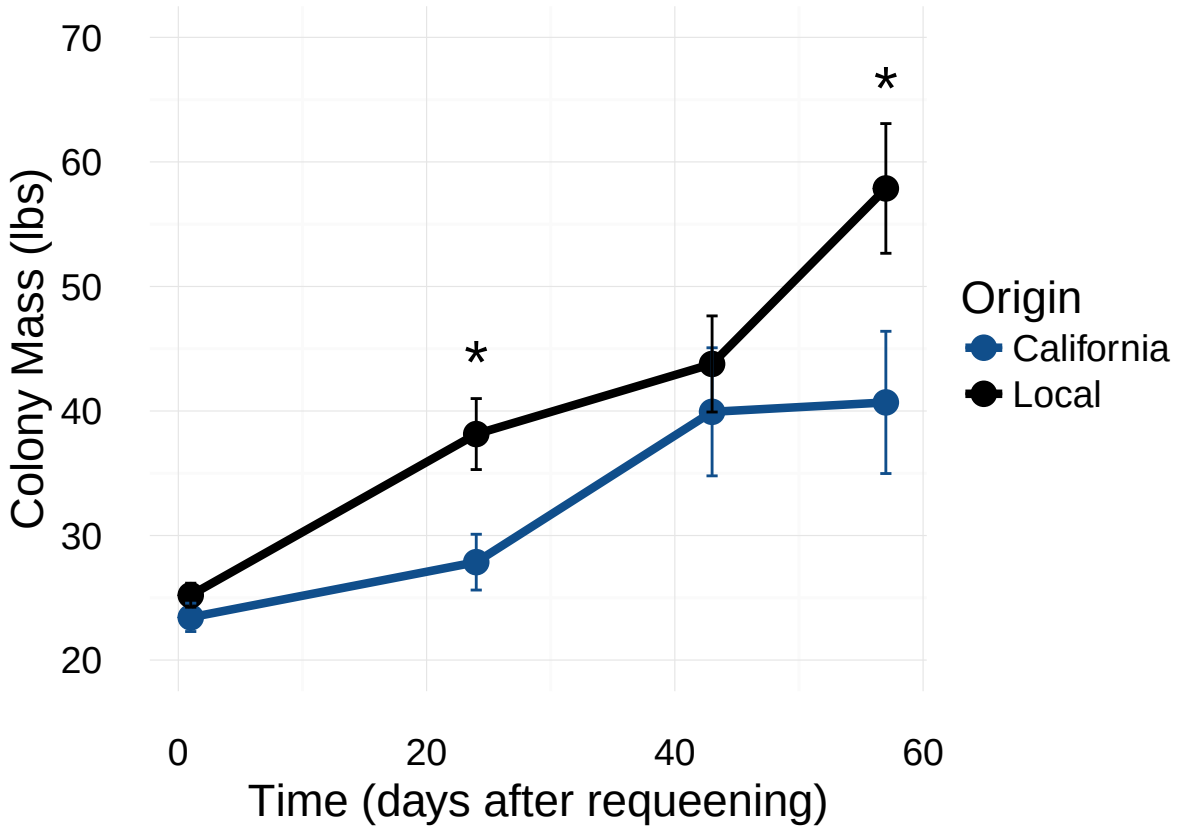
```
rm(list=ls())

# Set Working Directory
setwd("~/Desktop/QueenExperimentBurnham")

# read in data for all three .csv files:
QueensDF <- read.table("2016QueensHam.csv",header=TRUE,sep=",",stringsAsFactors=FALSE)

VirusDF <- read.table("RNAVirus.csv",header=TRUE,sep=",",stringsAsFactors=FALSE)

PollenDF <- read.table("PollenQueens.csv",header=TRUE,sep=",",stringsAsFactors=FALSE)
```

```
# looking at colony mass through time

#subsetting by removing rows that have missing data:
QueensMassDF <- QueensDF[-c(95,98,102,104,105),]

# create summary and sd and se using plyr
library(plyr)
MassSummary <- ddply(QueensMassDF, c("Origin", "Time"), summarise,
```

```
                  n = length(Mass),
                  mean = mean(Mass),
                  sd = sd(Mass),
                  se = sd / sqrt(n))

print(MassSummary)
```

```
##       Origin Time  n     mean        sd       se
## 1 California   T1 20 23.43000  5.096655 1.139647
## 2 California   T2 16 27.86250  8.960571 2.240143
## 3 California   T3  9 39.93333 15.427573 5.142524
## 4 California   T4  9 40.68889 17.132750 5.710917
## 5      Local   T1 20 25.21000  4.309341 0.963598
## 6      Local   T2 19 38.14737 12.422711 2.849965
## 7      Local   T3 17 43.77647 15.910355 3.858828
## 8      Local   T4 17 57.87059 21.473465 5.208080
```

```
library(ggplot2)

# remove generic time steps and add the correct days for correct time scale
x <- MassSummary[,-2]
times <- c(1,24,43,57,1,24,43,57)
x <- data.frame(times, x)

#Create plot in ggplot
plot <- ggplot(data = x,
               aes(x = times,
                   y = mean,
                   group = Origin,
                   colour = Origin)
               ) + geom_line(size=1.5) + geom_point(size=4) + scale_colour_manual(values = c("dodgerblu

# add a theme and add asterix for significance
plot + scale_fill_brewer(palette = "Paired") + theme_minimal(base_size = 17) + annotate(geom = "text", 
```




```
#Check for normality and log transform
#hist(QueensMassDF$Mass)
#hist(log(QueensMassDF$Mass))

#Subset data into 4 time steps (T1-T4)
QMassT1 <- QueensMassDF[c(1:40),]
QMassT2 <- QueensMassDF[c(41:75),]
QMassT3 <- QueensMassDF[c(76:106),]
QMassT4 <- QueensMassDF[c(107:132),]

# run 4 t-tests comparing california to local colony mass
ComparingQueensMassT1 <- aov(log(Mass)~Origin, data=QMassT1)
ComparingQueensMassT2 <- aov(log(Mass)~Origin, data=QMassT2)
ComparingQueensMassT3 <- aov(log(Mass)~Origin, data=QMassT3)
ComparingQueensMassT4 <- aov(log(Mass)~Origin, data=QMassT4)

# view the summary of the stats:
summary(ComparingQueensMassT1)
```

```
##             Df Sum Sq Mean Sq F value Pr(>F)
## Origin       1 0.0713 0.07132   1.731  0.196
## Residuals   38 1.5656 0.04120
```

```
summary(ComparingQueensMassT2)
```

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

```
## Origin       1  0.880  0.8804   8.124 0.00747 **
## Residuals   33  3.576  0.1084
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
summary(ComparingQueensMassT3)
```

```
##             Df Sum Sq Mean Sq F value Pr(>F)
## Origin       1  0.063 0.06296   0.498  0.486
## Residuals   29  3.663 0.12632
```

```
summary(ComparingQueensMassT4)
```

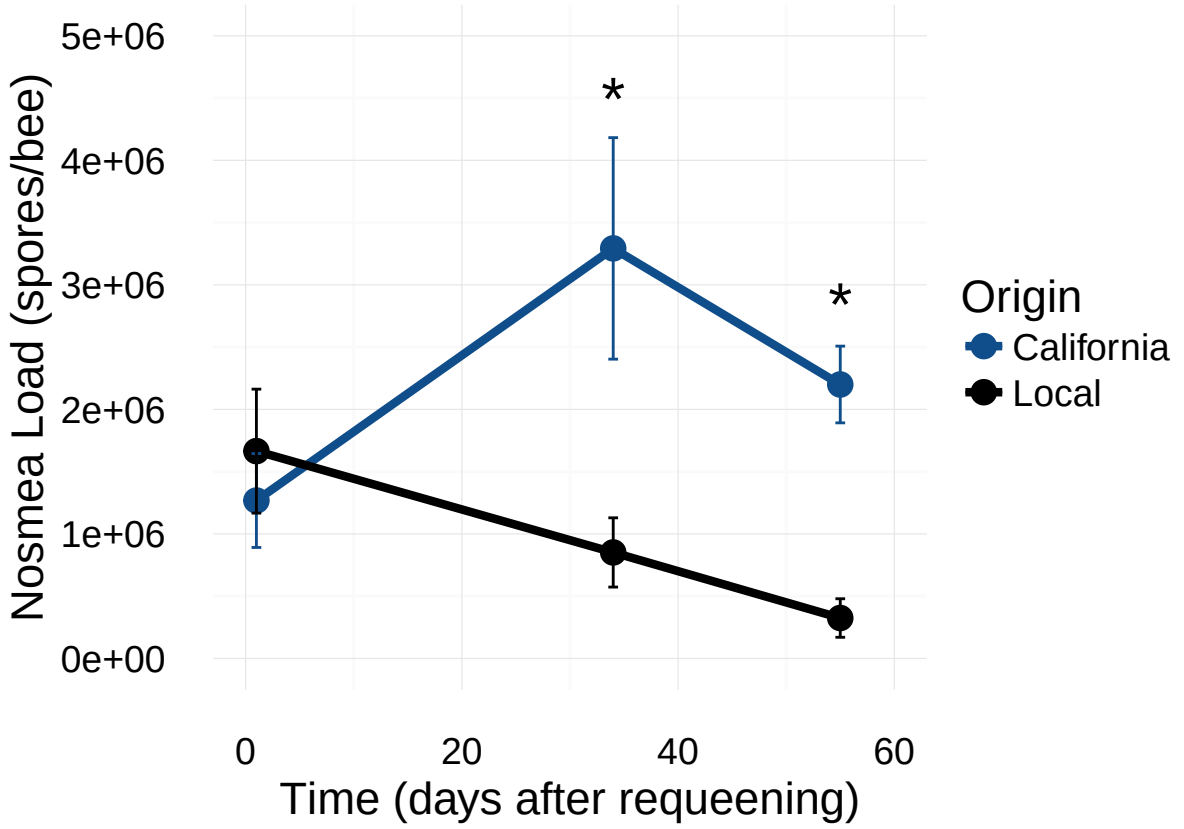
```
##             Df Sum Sq Mean Sq F value Pr(>F)
## Origin       1 0.9836  0.9836   6.831 0.0171 *
## Residuals   19 2.7359  0.1440
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 5 observations deleted due to missingness
```

```
#subsetting by removing rows that have missing data:
QueensNosemaDF <- QueensDF[-c(35,(41:75)),]

times <- c(rep("T1",39),rep("T2",31),rep("T3",26))

QueensNosemaDF <- data.frame(QueensNosemaDF, times)

# create summary and sd and se using plyr
library(plyr)
NosemaSummary <- ddply(QueensNosemaDF, c("Origin", "NosemaDay"), summarise,
                    n = length(Nosema),
                    mean = mean(Nosema),
                    sd = sd(Nosema),
                    se = sd / sqrt(n))

print(NosemaSummary)
```

```
##       Origin NosemaDay  n      mean        sd       se
## 1 California         1 19 1268421.1 1643079.8 376948.4
## 2 California        34 14 3292857.1 3327340.5 889269.2
## 3 California        55  9 2200000.0  923901.4 307967.1
## 4      Local         1 20 1665000.0 2225530.7 497643.8
## 5      Local        34 17  851470.6 1146546.1 278078.3
## 6      Local        55 17  325000.0  637806.3 154690.7
```

```
#Create plot in ggplot
plot <- ggplot(data = NosemaSummary,
               aes(x = NosemaDay,
                   y = mean,
                   group = Origin,
                   colour = Origin)
```

```
) + geom_line(size=1.5) + geom_point(size=4) + scale_colour_manual(values = c("dodgerblue4", "black")) +

# add a theme and add asterix for significance
plot + scale_fill_brewer(palette = "Paired") + theme_minimal(base_size = 17) + annotate(geom = "text", x
```

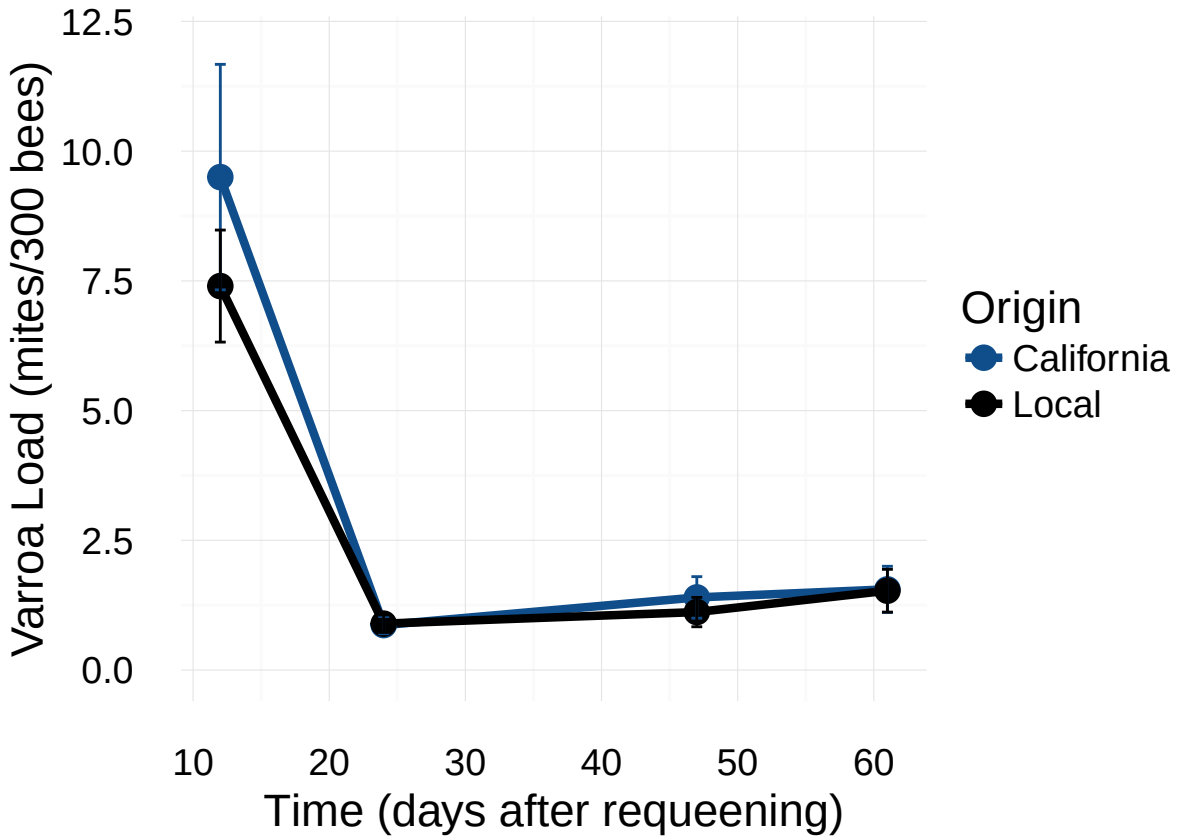



```
# stats comparing treatemetns at each timepoint:

#Check for normality and log transform
#hist(QueensDF$Nosema)
#hist(log(QueensDF$Nosema))

#Subset data into 3 time steps (T1-T4)
NosemaT1 <- QueensDF[c(1:40),]
NosemaT2 <- QueensDF[c(78:106),]
NosemaT3 <- QueensDF[c(107:132),]


# run 3 t-tests comparing california to local colony mass
ComparingQueensNosemaT1 <- aov(Nosema~Origin, data=NosemaT1)
ComparingQueensNosemaT2 <- aov(Nosema~Origin, data=NosemaT2)
ComparingQueensNosemaT3 <- aov(Nosema~Origin, data=NosemaT3)



# checking residuals for T1 for normality
ComparingQueensNosemaT1.res = resid(ComparingQueensNosemaT1)
```

```
NosemaT1 <- NosemaT1[-40,]

#plot(log(NosemaT1$Nosema), ComparingQueensNosemaT1.res)
#abline(0, 0)


#store the summaries in some variables
w <- summary(ComparingQueensNosemaT1) #Not sig
x <- summary(ComparingQueensNosemaT2) #Sig
y <- summary(ComparingQueensNosemaT3) #Sig
```

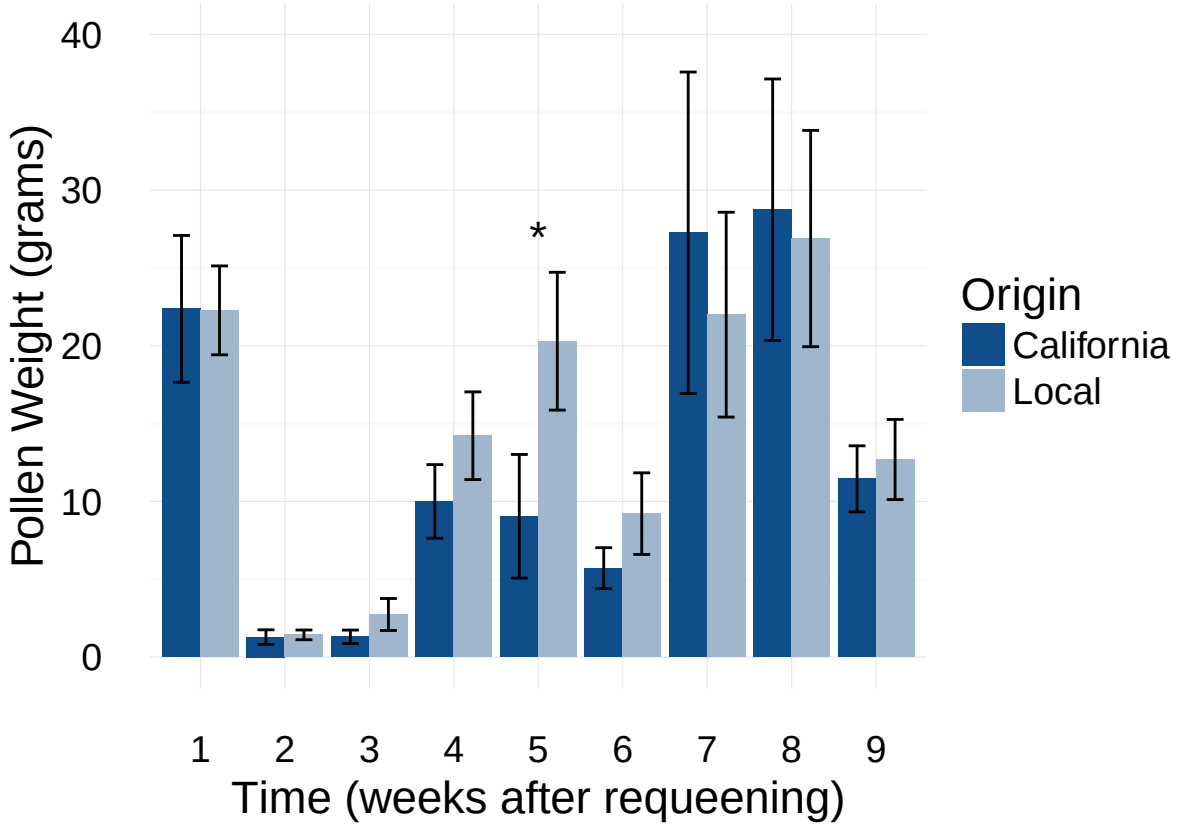
```
# create summary and sd and se using plyr for varroa
# Remove rows containing NAs in column Varroa
VarroaDF <- QueensDF[!is.na(QueensDF$Varroa),]

VarroaSummary <- ddply(VarroaDF, c("Origin", "VarroaDay"), summarise,
                       n = length(Varroa),
                       mean = mean(Varroa),
                       sd = sd(Varroa),
                       se = sd / sqrt(n))
```

```
#Create varroa line graph in ggplot
plot <- ggplot(data = VarroaSummary,
               aes(x = VarroaDay,
                   y = mean,
                   group = Origin,
                   colour = Origin)
) + geom_line(size=1.5) + geom_point(size=4) + scale_colour_manual(values = c("dodgerblue4", "black")) +

# add a theme and add asterix for significance
plot + scale_fill_brewer(palette = "Paired") + theme_minimal(base_size = 17)
```



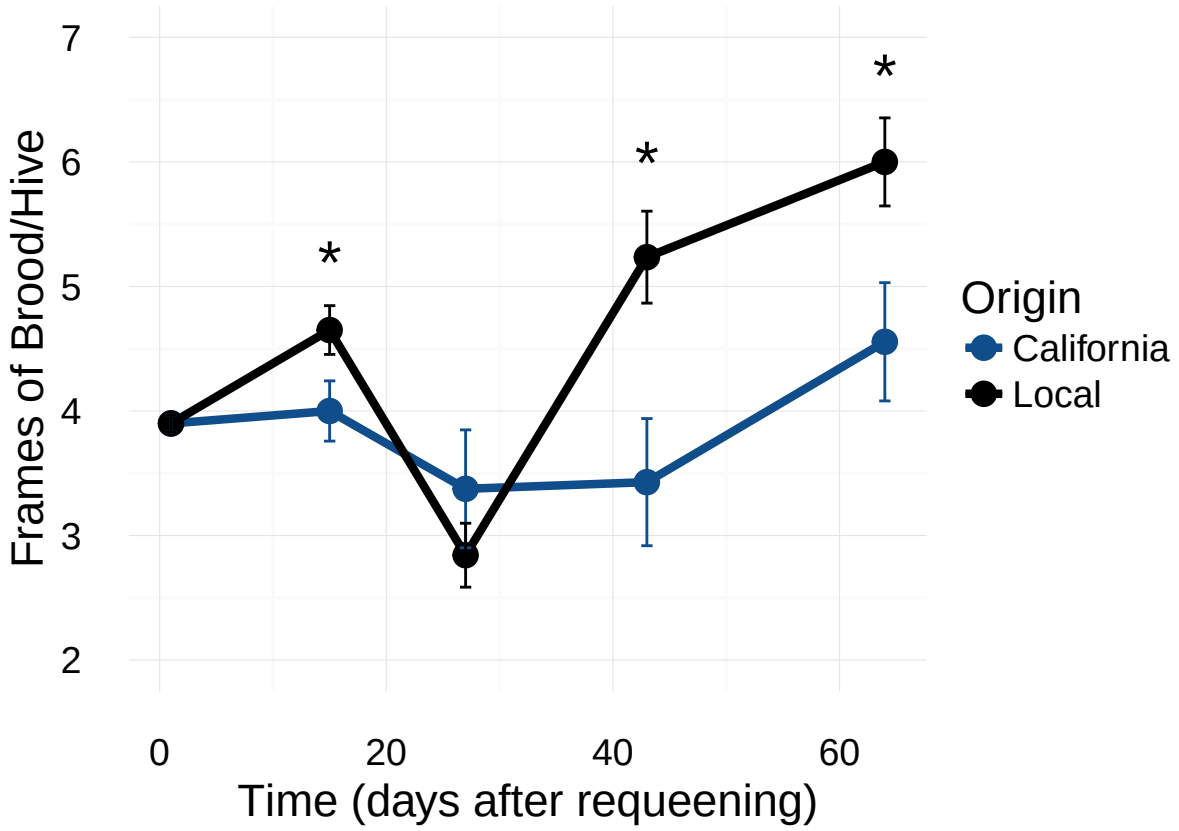

```
# pollen work making a bargraph
# read in data
PollenDF <- read.table("PollenQueens.csv",header=TRUE,sep=",",stringsAsFactors=FALSE)

PollenDF2 <- PollenDF[!is.na(PollenDF$Pollen),]

PollenDF2$WeekChar <- as.character(PollenDF2$Week)

PollenSummary <- ddply(PollenDF2, c("Origin", "WeekChar"), summarise,
                       n = length(Pollen),
                       mean = mean(Pollen),
                       sd = sd(Pollen),
                       se = sd / sqrt(n))
```

```
colors <- c("dodgerblue4", "slategray3")

plot2 <- ggplot(PollenSummary, aes(x=WeekChar, y=mean, fill=Origin)) +
  geom_bar(stat="identity",
           position=position_dodge()) +
  geom_errorbar(aes(ymin=mean-se, ymax=mean+se),
                width=.4,
                position=position_dodge(.9)) + labs(x="Time (weeks after requeening)",
                                                    y = "Pollen Weight (grams)")

plot2 + theme_minimal(base_size = 17) + coord_cartesian(ylim = c(0, 40)) + scale_fill_manual(values=col
```




```
# Brood line graph

BroodDF <- PollenDF[!is.na(PollenDF$Brood),]

BroodSummary <- ddply(BroodDF, c("Origin", "BroodDays"), summarise,
                      n = length(Brood),
                      mean = mean(Brood),
                      sd = sd(Brood),
                      se = sd / sqrt(n))
```

```
plot <- ggplot(data = BroodSummary,
               aes(x = BroodDays,
                   y = mean,
                   group = Origin,
                   colour = Origin)
) + geom_line(size=1.5) + geom_point(size=4) + scale_colour_manual(values = c("dodgerblue4", "black")) +

# add a theme and add asterix for significance
plot + scale_fill_brewer(palette = "Paired") + theme_minimal(base_size = 17) + annotate(geom = "text", x
```




```
# split data frame and do t-tests between treatments and each time point
x<-split(BroodDF, BroodDF$BroodDays)

t.test(x$`15`$Brood~x$`15`$Origin)
```

```
##
##  Welch Two Sample t-test
##
## data:  x$`15`$Brood by x$`15`$Origin
## t = -2.0895, df = 35.052, p-value = 0.04399
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -1.28148687 -0.01851313
## sample estimates:
## mean in group California      mean in group Local
##                     4.00                     4.65
```

```
t.test(x$`27`$Brood~x$`27`$Origin)
```

```
##
##  Welch Two Sample t-test
##
## data:  x$`27`$Brood by x$`27`$Origin
## t = 0.98986, df = 23.432, p-value = 0.3324
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
```

```
##  -0.579641  1.645430
## sample estimates:
## mean in group California      mean in group Local
##                 3.375000                 2.842105
```

```
t.test(x$`43`$Brood~x$`43`$Origin)
```

```
##
##  Welch Two Sample t-test
##
## data:  x$`43`$Brood by x$`43`$Origin
## t = -2.868, df = 24.668, p-value = 0.008336
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -3.1050229 -0.5084225
## sample estimates:
## mean in group California      mean in group Local
##                 3.428571                 5.235294
```

```
t.test(x$`64`$Brood~x$`64`$Origin)
```

```
##
##  Welch Two Sample t-test
##
## data:  x$`64`$Brood by x$`64`$Origin
## t = -2.4405, df = 16.76, p-value = 0.02609
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -2.6945432 -0.1943457
## sample estimates:
## mean in group California      mean in group Local
##                 4.555556                 6.000000
```

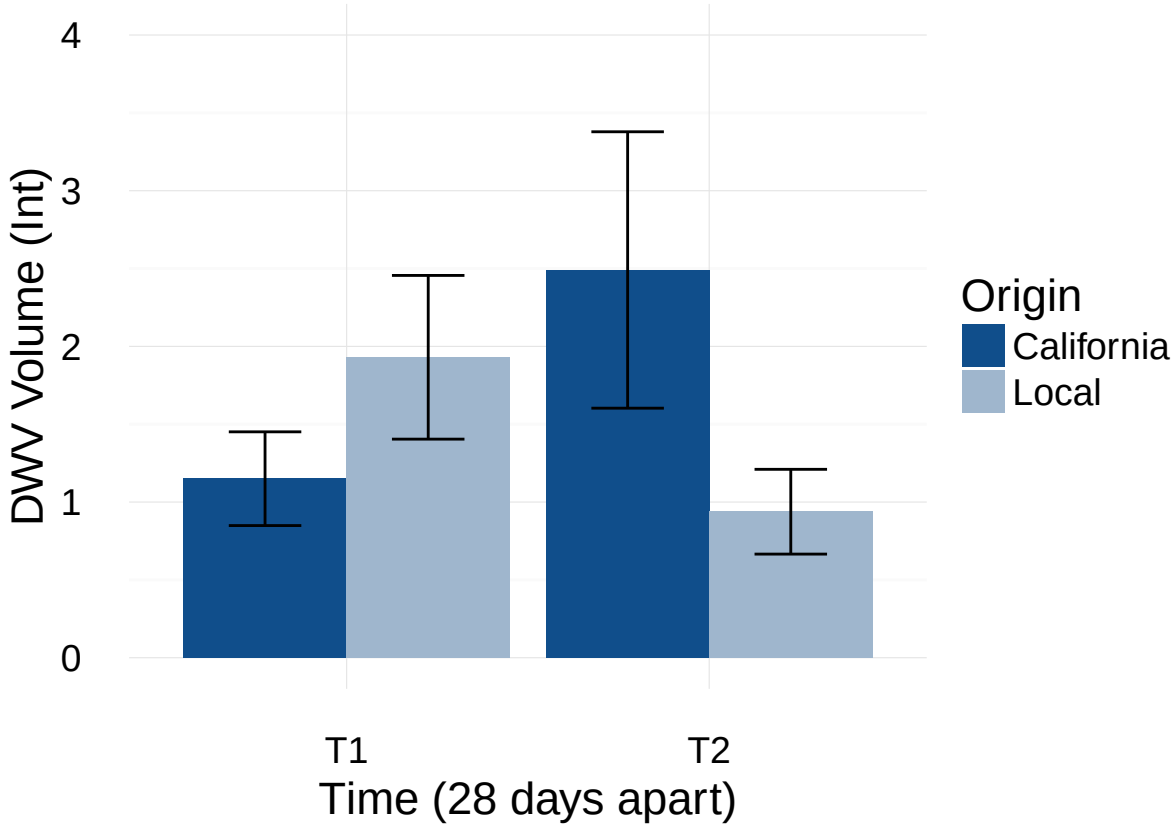
```
#Virus stats:

# read in data frame:
VirusDF <- read.table("RNAVirus.csv",header=TRUE,sep=",",stringsAsFactors=FALSE)

VirusDF <- VirusDF[!is.na(VirusDF$Intensity),]

Virus1 <- split(VirusDF,VirusDF$Virus)
DWV <- Virus1$DWV
IAPV <- Virus1$IAPV

x <- split(DWV, DWV$Time)

t.test(x$T2$Intensity~x$T2$Origin, alternative="greater")
```

```
##
##  Welch Two Sample t-test
##
## data:  x$T2$Intensity by x$T2$Origin
```

```
## t = 1.6722, df = 14.284, p-value = 0.05812
## alternative hypothesis: true difference in means is greater than 0
## 95 percent confidence interval:
##  -0.08045176         Inf
## sample estimates:
## mean in group California      mean in group Local
##                 2.490769                 0.938000
```

```
VirusSummary <- ddply(DWV, c("Origin", "Time"), summarise,
                      n = length(Intensity),
                      mean = mean(Intensity),
                      sd = sd(Intensity),
                      se = sd / sqrt(n))
```

```
#Plot Virus in ggplot

colors <- c("dodgerblue4", "slategray3")

plot1 <- ggplot(VirusSummary, aes(x=Time, y=mean, fill=Origin)) +
  geom_bar(stat="identity",
           position=position_dodge()) +
  geom_errorbar(aes(ymin=mean-se, ymax=mean+se),
                width=.4,
                position=position_dodge(.9)) + labs(x="Time (28 days apart)",
                                                    y = "DWV Volume (Int)")

plot1 + theme_minimal(base_size = 17) + coord_cartesian(ylim = c(0, 4)) + scale_fill_manual(values=colo
```



```r
#IAPV
VirusDF <- VirusDF[!is.na(VirusDF$Intensity),]

Virus1 <- split(VirusDF,VirusDF$Virus)
IAPV <- Virus1$IAPV

z <- split(IAPV, IAPV$Time)

t.test(z$T2$Intensity~z$T2$Origin)
```

```
##
##  Welch Two Sample t-test
##
## data:  z$T2$Intensity by z$T2$Origin
## t = 2.5582, df = 22.041, p-value = 0.01791
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  0.253737 2.425494
## sample estimates:
## mean in group California      mean in group Local
##                 2.324615                 0.985000
```

```r
t.test(z$T1$Intensity~z$T1$Origin)
```

```
##
##  Welch Two Sample t-test
##
## data:  z$T1$Intensity by z$T1$Origin
## t = 2.6418, df = 34.262, p-value = 0.01234
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  0.2464288 1.8874659
## sample estimates:
## mean in group California      mean in group Local
##                 1.708947                 0.642000
```

```r
VirusSummary1 <- ddply(IAPV, c("Origin", "Time"), summarise,
                       n = length(Intensity),
                       mean = mean(Intensity),
                       sd = sd(Intensity),
                       se = sd / sqrt(n))
```

```r
#Plot IAPV in ggplot


plot2 <- ggplot(VirusSummary1, aes(x=Time, y=mean, fill=Origin)) +
  geom_bar(stat="identity",
           position=position_dodge()) +
  geom_errorbar(aes(ymin=mean-se, ymax=mean+se),
                width=.4,
                position=position_dodge(.9)) + labs(x="Time (28 days apart",
                                                    y = "IAPV Volume (Int)")
```

```
plot2 + theme_minimal(base_size = 17) + coord_cartesian(ylim = c(0, 3)) + scale_fill_manual(values=colo
```