AWC_imaging_analysis

Emily Jane Dennis

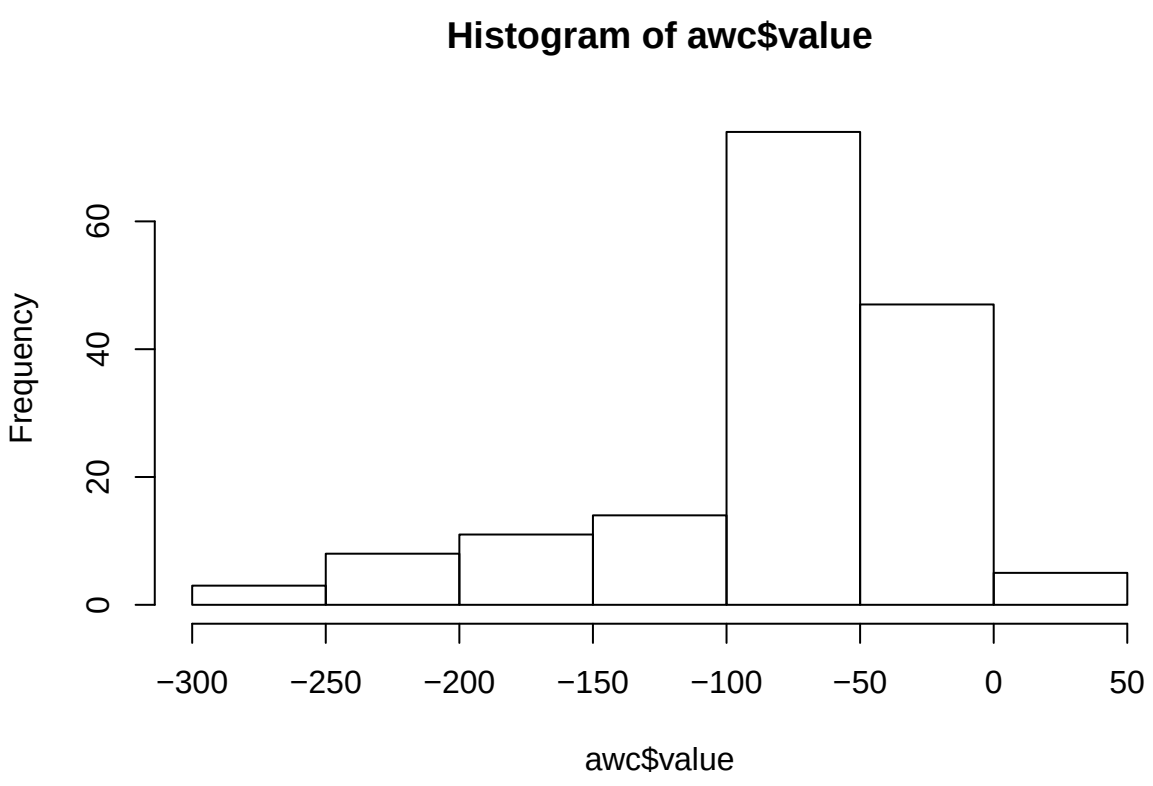
7/27/2017

Data were pre-processed as described in the methods

Raw data are in the AWC_response_mags_rawdata.xlsx file

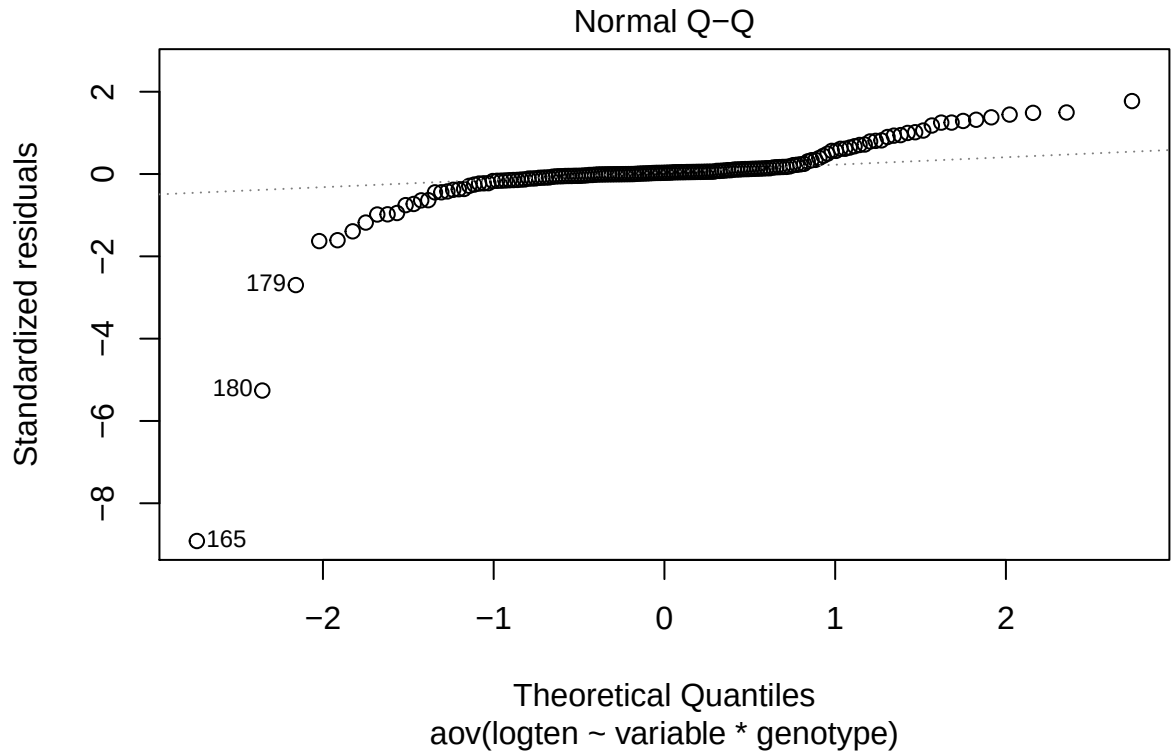
First: tidy these data for analysis

```
fullawcimaging <- read.csv('AWC_response_mags.csv',header=TRUE)
awcformatted <- melt(fullawcimaging,id.vars = c('genotype','animal'),value.var='responsemags')

awcDEETon <- subset(awcformatted,variable=='onset')
awc <- subset(awcformatted,variable!='onset')

hist(awc$value,breaks=10)
```

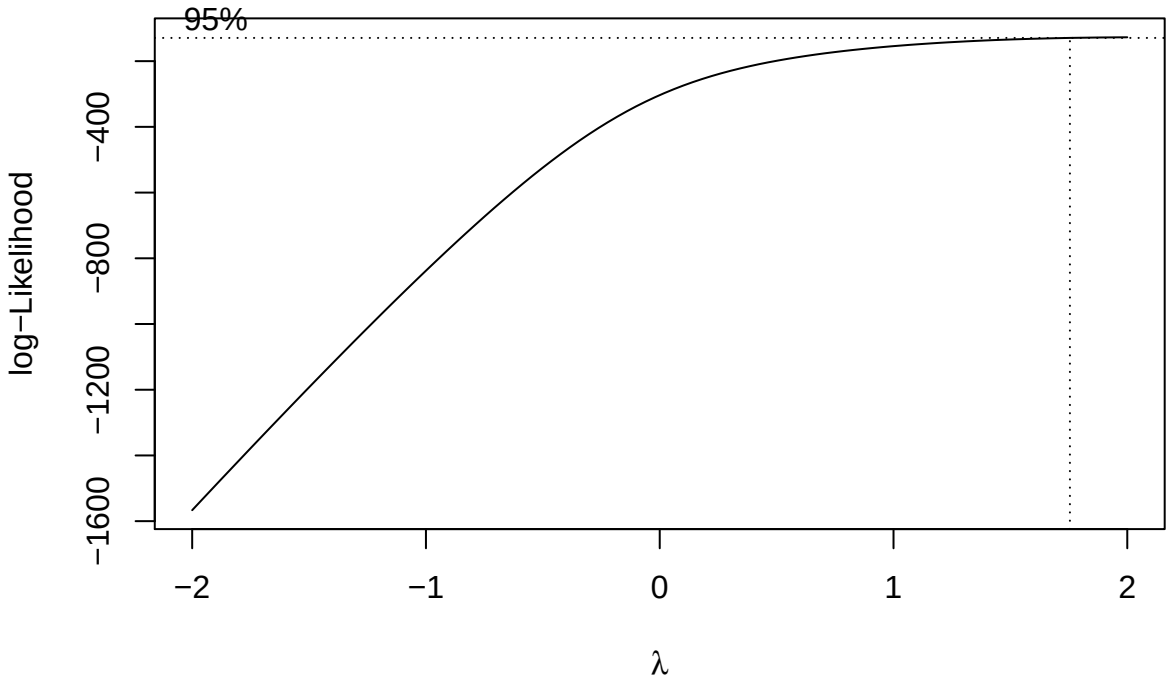



```
#translated to smallest value is 1
translationawc <- 1-(min(awc$value))
awctranslated <- awc
awctranslated$value <- awc$value+translationawc

awctranslated$logten <- log10(awctranslated$value)

#log10 looked worse
```

```
awcstatlog.aov = aov(logten~variable*genotype,data=awctranslated)
plot(awcstatlog.aov,which=2)
```

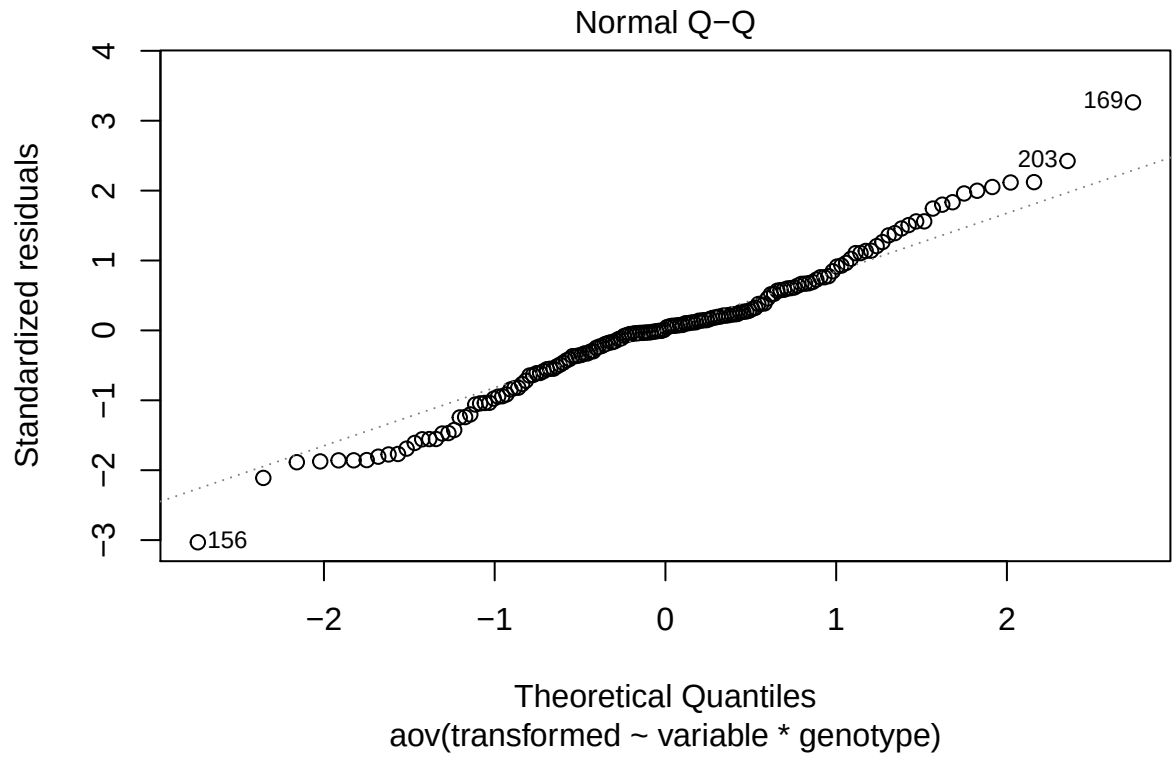



```
#used boxcox to find good lambda
bc <- boxcox(value~variable*genotype,data=awctranslated)
```

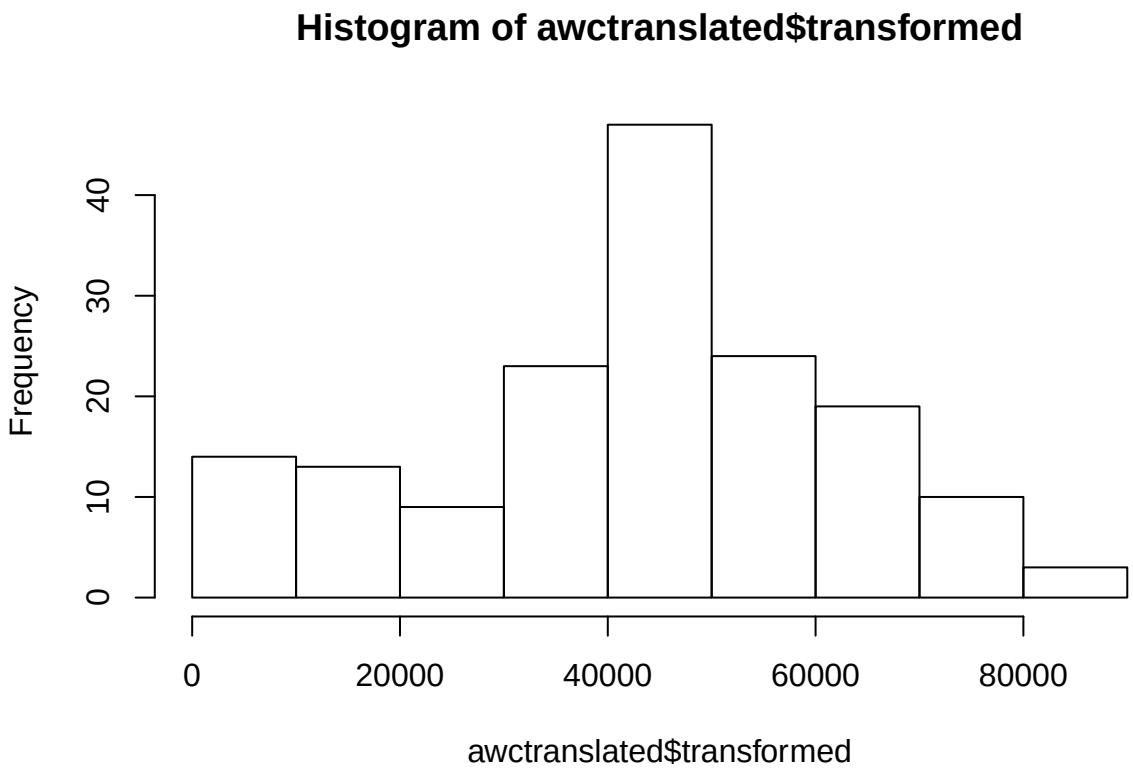



```
lambda <- round(bc$x[which.max(bc$y)],1)
awctranslated$transformed <- (awctranslated$value)^lambda
```

```
awcstat.aov = aov(transformed~variable*genotype,data=awctranslated)
plot(awcstat.aov,which=2)
```




```
hist(awctranslated$transformed,breaks=10)
```



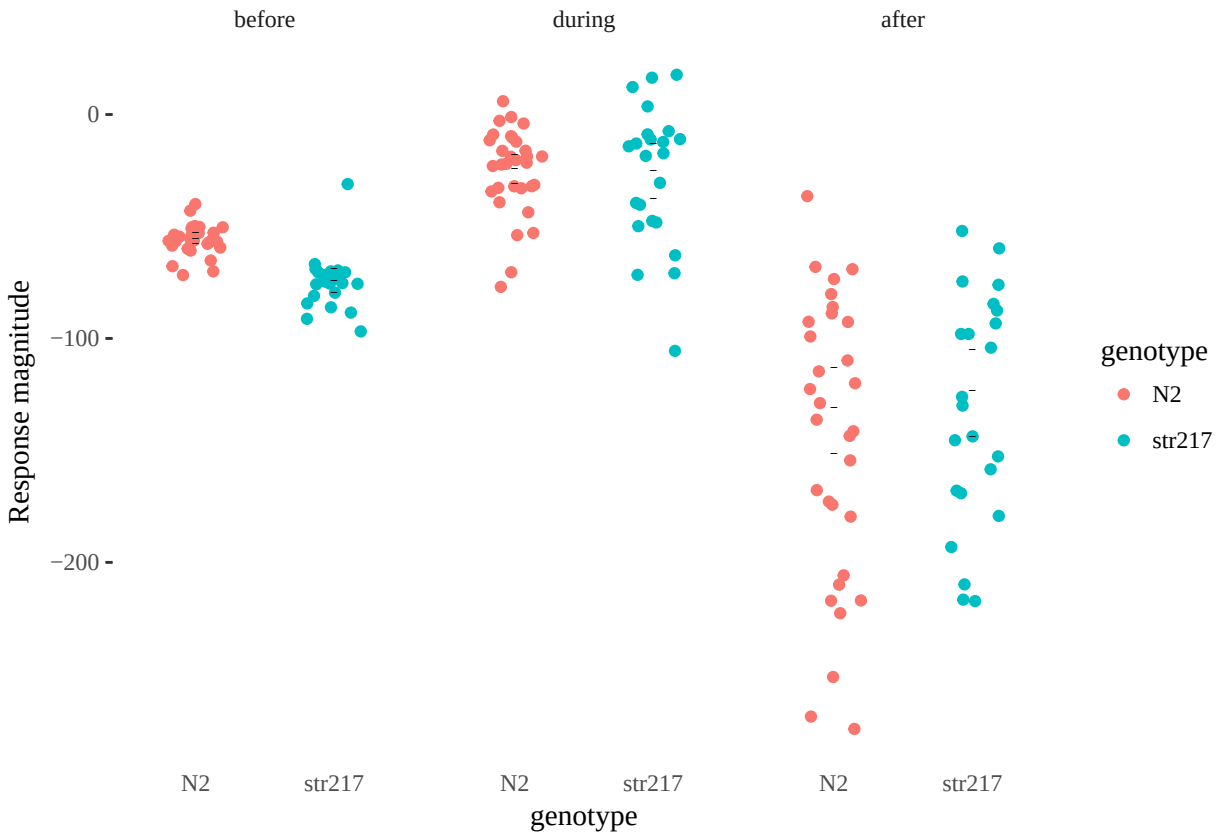
```
tukeyawc <- TukeyHSD(aov(transformed~variable*genotype,data=awctranslated))

tukeyawc.levels <- tukeyawc$`variable:genotype`[,4]
multcompLetters(tukeyawc.levels)['Letters']
```

```
## $Letters
##     during:N2      after:N2 before:str217 during:str217  after:str217
##           "a"           "b"           "c"           "a"           "b"
##     before:N2
##           "c"
```

```
awctransformed <- ddply(awctranslated,.(genotype,variable),summarize,means=mean(transformed),sd=sd(trans

awctransformedback <- awctransformed
awctransformedback$means <- sqrt(awctransformedback$means)-translationawc
awctransformedback$cim <- sqrt(awctransformedback$cim)-translationawc
awctransformedback$cip <- sqrt(awctransformedback$cip)-translationawc
#took the sqrt because lambda==2, therefore to get x from y=x^2 need y^1/2 aka sqrt(y)
```

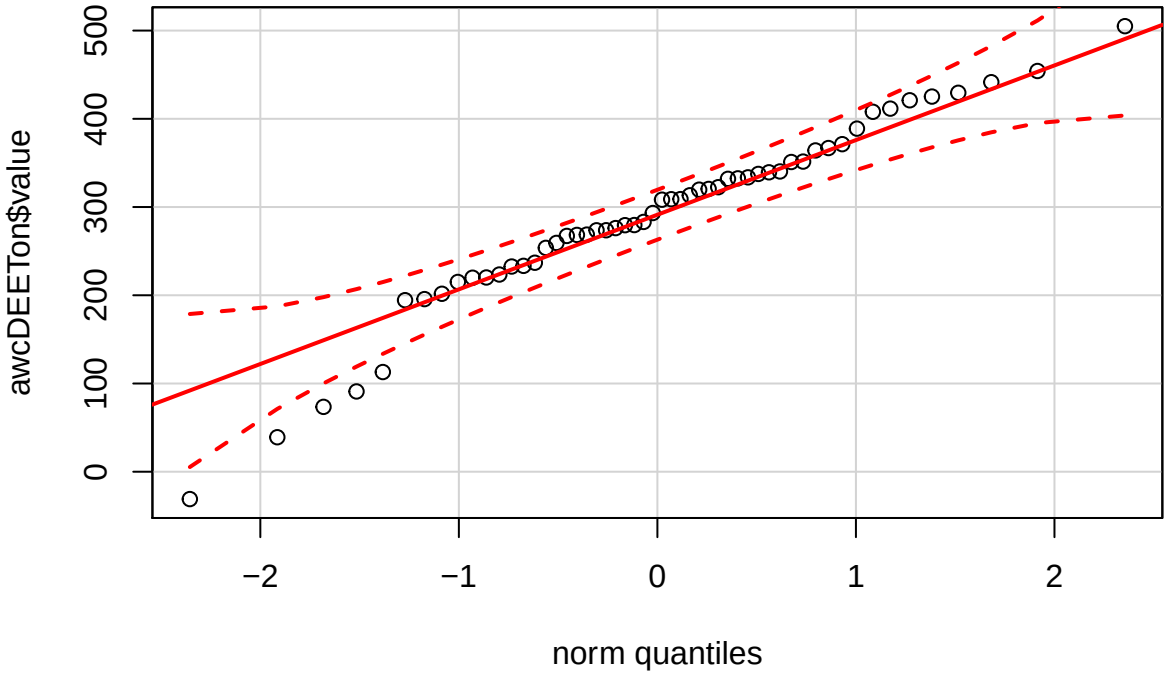
plotting data

```
ggplot(awc,aes(x=genotype,y=value)) +
  geom_jitter(height=0,width=0.2,aes(color=genotype)) +
  facet_grid(~variable) + theme_tufte() +
  ylab("Response magnitude") +
  theme(axis.ticks.x=element_blank()) +
  geom_point(data=awctransformedback,aes(x=genotype,y=(means)),shape='-') +
  geom_point(data=awctransformedback,aes(x=genotype,y=(cip)),shape='-') +
  geom_point(data=awctransformedback,aes(x=genotype,y=(cim)),shape='-')
```

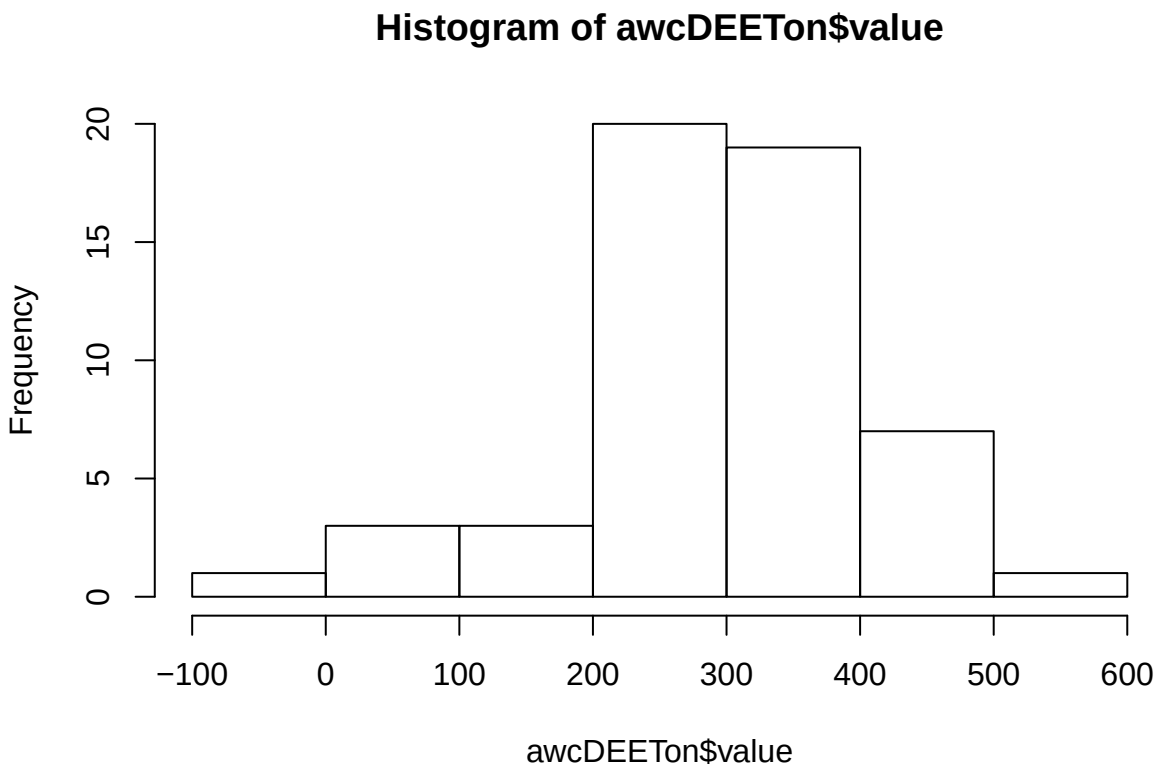


the distributions are skewed:

```
qqPlot(awcDEETon$value)
```



```
hist(awcDEETon$value)
```



so I used a non-parametric test, because these data are close but not sufficiently large

```
wilcox.test(value~genotype,data=awcDEETon)
```

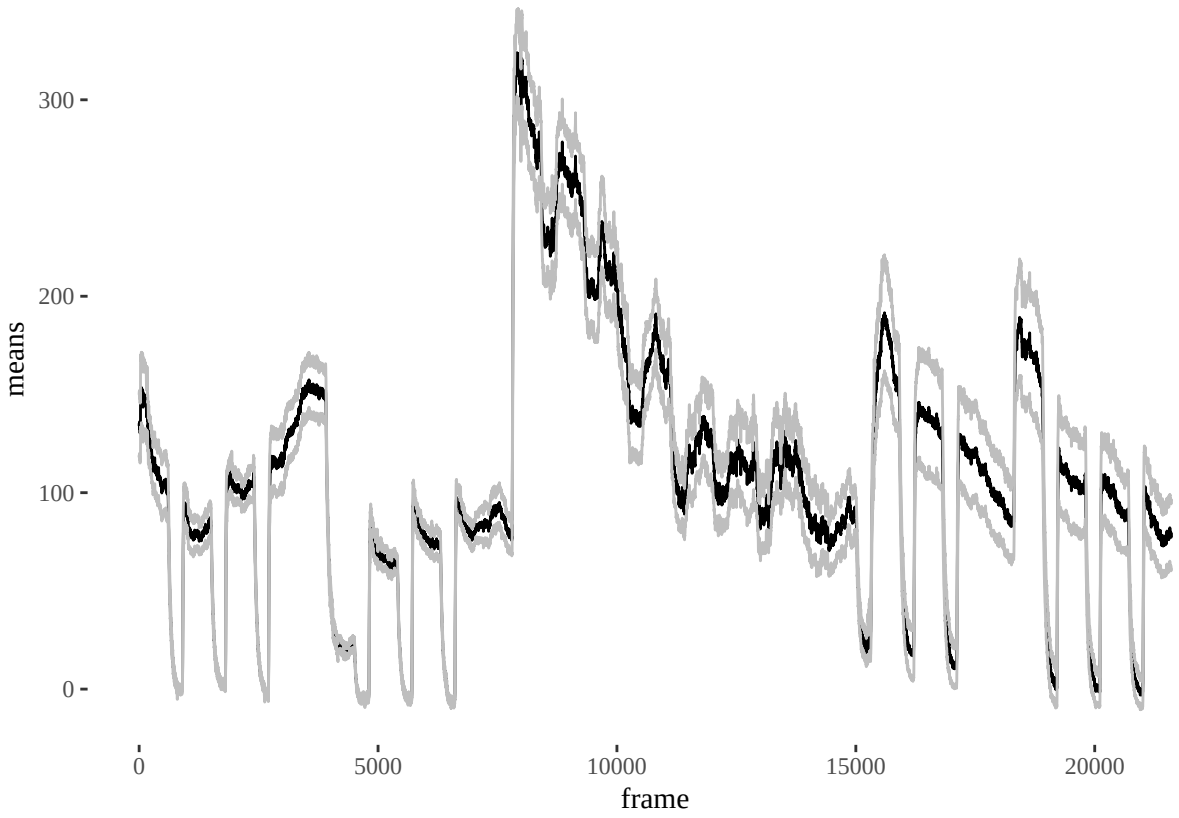
```
##
##  Wilcoxon rank sum test
##
## data:  value by genotype
## W = 389, p-value = 0.5786
## alternative hypothesis: true location shift is not equal to 0
```

and therefore plotted medians and confidence intervals

```
n2 <- subset(awcDEETon,genotype=='N2')$value
str217 <- subset(awcDEETon,genotype=='str217')$value

citoadd <- ddply(awcDEETon,.(genotype),summarize,median=median(value))

cin2 <- boxplot.stats(n2, do.conf=T)$conf
cistr217 <-  boxplot.stats(str217, do.conf=T)$conf

citoadd$cimax <- c(max(cin2),max(cistr217))
citoadd$cimin <- c(min(cin2),(min(cistr217)))

ggplot(awcDEETon,aes(x=genotype,y=value)) +
  geom_jitter(width=0.2,height=0,aes(color=genotype)) +
  theme_tufte()+
  geom_point(data=citoadd,aes(x=genotype,y=cimax),shape='-') +
  geom_point(data=citoadd,aes(x=genotype,y=cimin),shape='-')+
```

```
geom_point(data=citoadd,aes(x=genotype,y=median),shape='+')
```


```
mutplot <- read.csv('str217_forplot.csv',header=TRUE)
n2plot <- read.csv('N2_forplot.csv',header=TRUE)
strmelt <- melt(mutplot,id.vars =c('frame'))
strply <- ddply(strmelt,~frame,summarise, means=mean(value),N=length(value),sd=sd(value),se=sd/(sqrt(N)
n2melt <- melt(n2plot,id.vars =c('frame'))
n2ply <- ddply(n2melt,~frame,summarise, means=mean(value),N=length(value),sd=sd(value),se=sd/(sqrt(N)))
ggplot(n2ply,aes(x=frame,y=means)) + geom_line(color='black') + geom_line(aes(x=frame,y=means+se),color=
```




```
ggplot(strply,aes(x=frame,y=means)) + geom_line(color='red3') + geom_line(aes(x=frame,y=means+se),color=
```