Circadian Mood

2022-11-21

Install missing packages

```
pacman::p_load(tidyverse, lmerTest)
```

#Get and clean BodyFeelingExp data

```
library(tidyverse)


path <- r"(C:\Users\maria\OneDrive\Dokumenter\Uni\Advanced Cognitive
Neuroscience\homeostasis_and_happiness\BodyFeelingExp_data\BodyFeelingExp_data)"

setwd(path)

#Get FaceStroopData
datadir <- path


#Find files
files<-list.files(datadir,pattern='^BodyFeelingExp_.+?csv',full.names=TRUE)

#Prepare an empty data frame for the data (also removes old version)
dataBFE<-data.frame()

#How many datasets were there
n_datasets_raw<-length(files)
#Prepare a variable to monitor how many datasets we keep
n_datasets<-0
#Prepare a variable to monitor how many points we originally had
n_datapoints_raw<-0

#Loop to go through all files in the list
for(iii in 1:n_datasets_raw){

  #remove old loaded file to not risk importing it multiple times
  if(exists('data_temp')) rm(data_temp)

  #Load data
  data_temp<-read.csv(files[iii])
  if(dim(data_temp)[2]==31){
    data_temp[1,6]<-data_temp[dim(data_temp)[1],6]
    data_temp<-data_temp[1,c(6,8:27)]
```

```
        if(length(colnames(dataBFE))==0){
          dataBFE=data_temp
          rm(data_temp)
          #counter to monitor included datasets
          n_datasets<-n_datasets+1
        }
        #Bind loaded data with actual data
        else {dataBFE<-rbind(dataBFE,data_temp)
          rm(data_temp)
          #counter to monitor included datasets
          n_datasets<-n_datasets+1
        }
    }
}

#A variable to monitor how many points we keep
n_datapoints<-length(dataBFE[,1])
```

Additional preprocessing

```
#Make a variable which has hour and minutes of the day as decimal variable
dataBFE$hour2<-dataBFE$hour+(dataBFE$minute)/60
```

summarizing stats – assess IDs

```
dataBFE %>%
  distinct(id) %>%
  nrow()
```

```
## [1] 35
```

```
dataBFE %>%
  group_by(id) %>%
  dplyr::summarise(count = length((id)))
```

```
## `summarise()` ungrouping output (override with `.groups` argument)
```

```
## # A tibble: 35 x 2
##    id       count
##    <chr>    <int>
##  1 " Woo55"     1
##  2 "Cic22"     20
##  3 "dan72"      6
##  4 "dig05"      1
##  5 "dlg05"     27
##  6 "eeg339"     9
##  7 "ESG44"     22
##  8 "ESG44 "     1
##  9 "hej12"     11
```

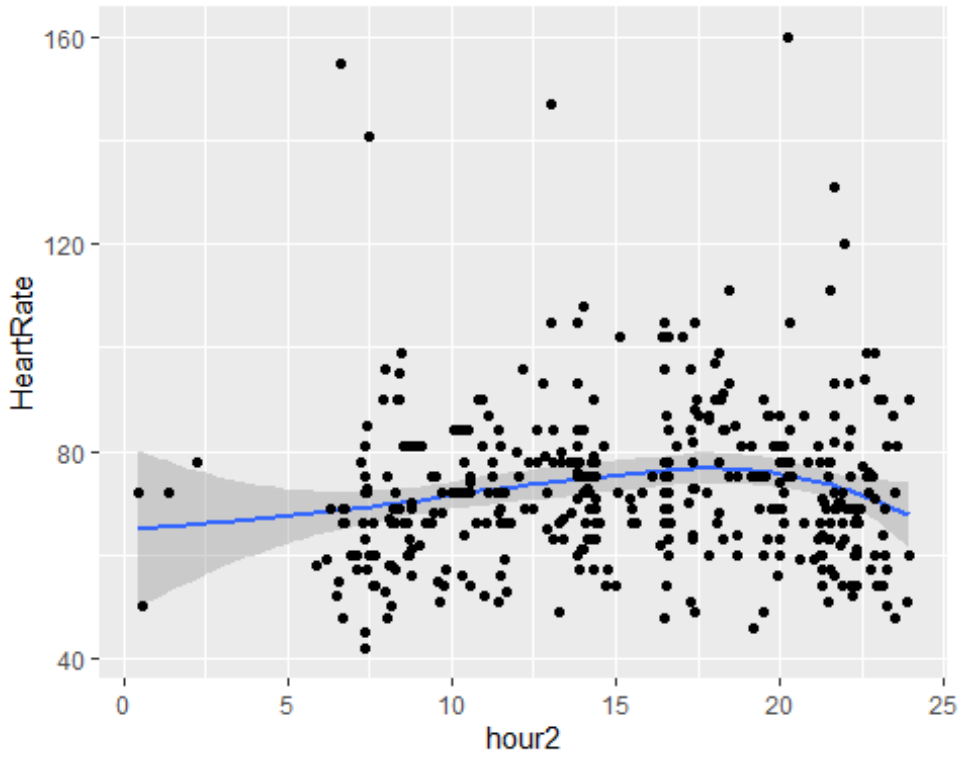
```
## 10 "hej123"     1
## # ... with 25 more rows

# nrow(dataBFE)
```

Take care of duplicate IDs

```
# fix participant IDs
dataBFE <- dataBFE %>%
  mutate(id = ifelse(id == " Woo55", "woo55", id)) %>%
  mutate(id = ifelse(id == "Woo55", "woo55", id)) %>%
  mutate(id = ifelse(id == "ESG44 ", "ESG44", id)) %>%
  mutate(id = ifelse(id == " POP33", "POP33", id)) %>%
  mutate(id = ifelse(id == "POP33 ", "POP33", id)) %>%
  mutate(id = ifelse(id == "hej123", "hej12", id)) %>%
  mutate(id = ifelse(id == "Qwel12", "qwel12", id)) %>%
  mutate(id = ifelse(id == "www111", "www11", id)) %>%
  mutate(id = ifelse(id == "Qwe12", "qwe12", id)) %>%
  mutate(id = ifelse(id == "www123", "www12", id)) %>%
  mutate(id = ifelse(id == "dig05", "dlg05", id))
```

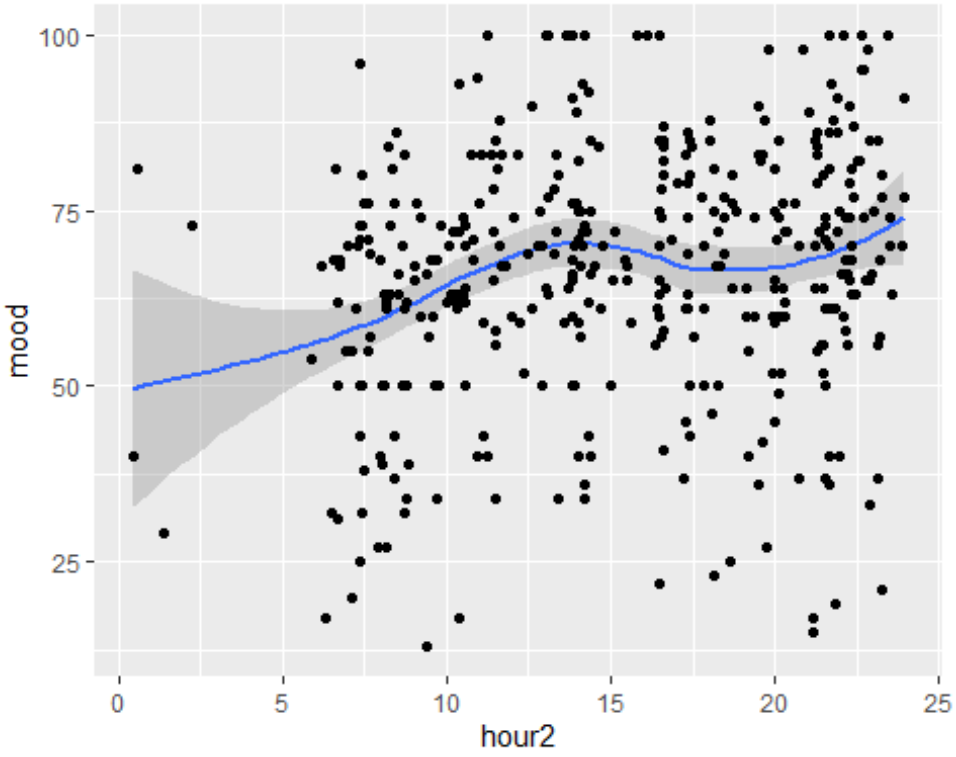
Some plotting

```
library(ggplot2)
ggplot(dataBFE,aes(x=hour2,y=HeartRate))+geom_smooth()+geom_point()

## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```



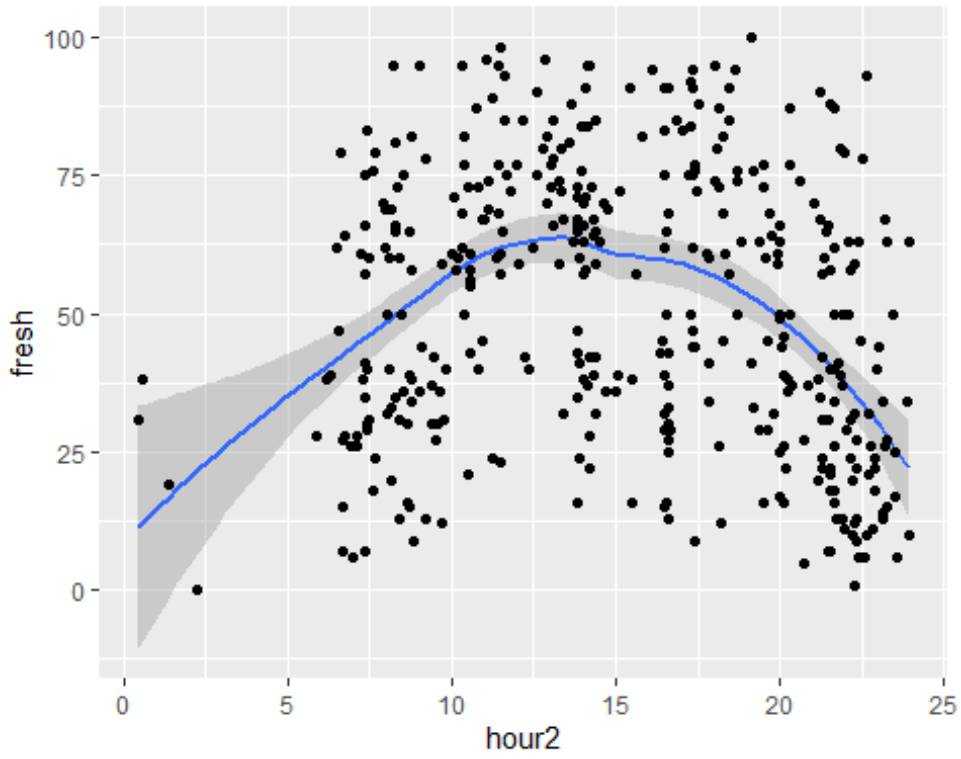

```
ggplot(dataBFE,aes(x=hour2,y=mood))+geom_smooth()+geom_point()
```

```
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```



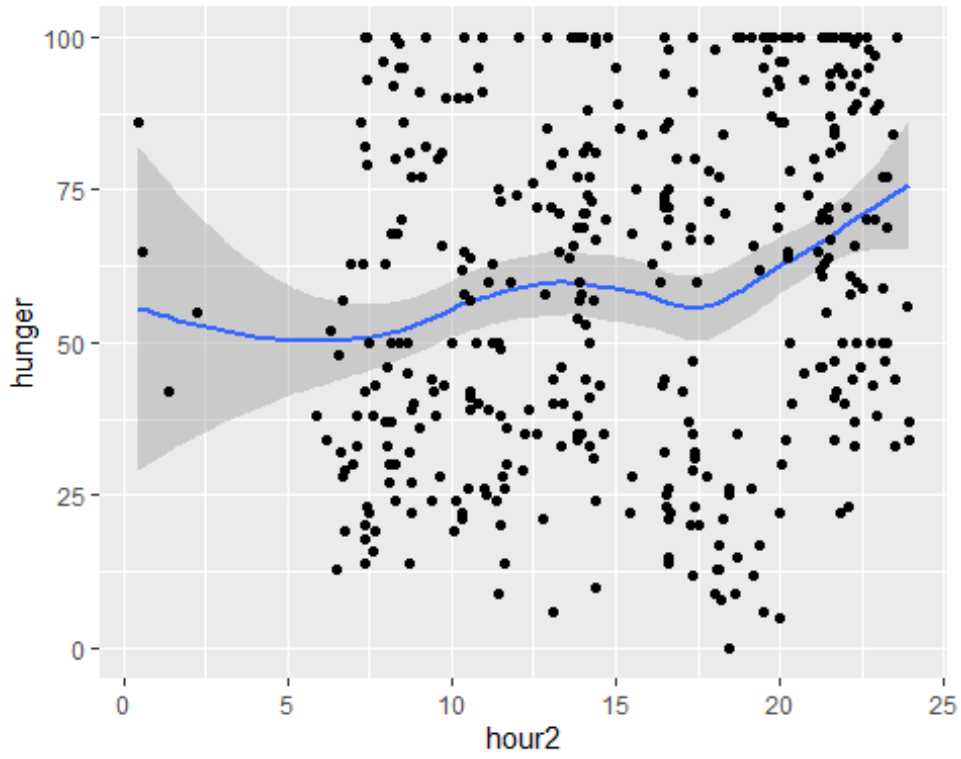

```
ggplot(dataBFE,aes(x=hour2,y=fresh))+geom_smooth()+geom_point()
```

```
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```




```
ggplot(dataBFE,aes(x=hour2,y=hunger))+geom_smooth()+geom_point()
```

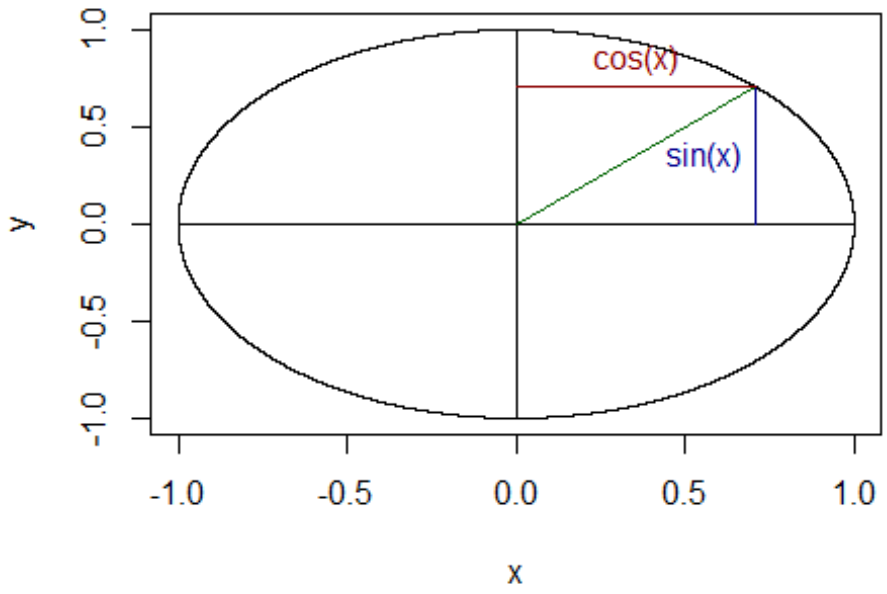
```
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```



High-school trigonometry for circidian analysis

It all starts with a circle and a triangle

```
#Radius
r<-1
# Get x-values from minus 1 to plus 1
x<-seq(-1,1,0.00001)
# A circle with c(0,0) centre can be written with these two equations (following
Pythagoras)
y1<-sqrt(r^2-x^2)
y2<--sqrt(r^2-x^2)
y<-c(y1,y2)
x<-c(x,x)
#Plotting the circle with sine and cosine values
pp=pi/4
plot(x,y,type='l')
lines(x=c(0,0),y=c(-1,1))
lines(x=c(0,cos(pp)),y=c(0,sin(pp)),col='darkgreen')
lines(x=c(cos(pp),cos(pp)),y=c(0,sin(pp)),col='darkblue')
text(x=c(-0.15+cos(pp)),y=c(0.5*sin(pp)),labels='sin(x)',col='darkblue')
lines(x=c(0,cos(pp)),y=c(sin(pp),sin(pp)),col='darkred')
text(x=c(0.5*cos(pp)),y=c(+0.15+sin(pp)),labels='cos(x)',col='darkred')
```



Going beyond the circle, the sine and cosine functions can describe cycles in time

$$y(t) = \beta sin(2\pi f t)$$

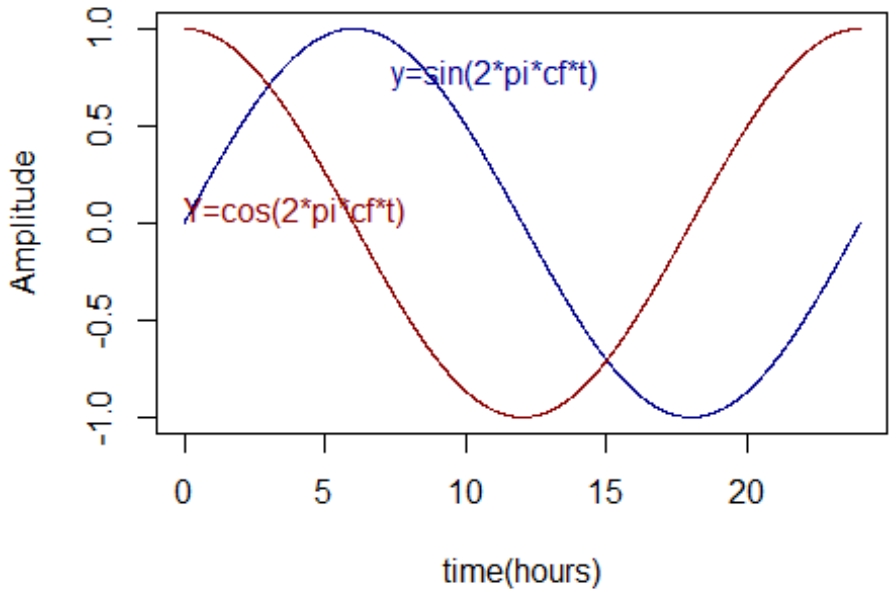
$$y(t) = \beta cos(2\pi f t)$$

Beta is the amplitude (the height of the wave) f is the frequency (the number of cycles per time unit) t is time point

Plotting circadian sine and cosine waves

```
#cycle frequency (in this case per hour) - one cycle per 24 hours
cf=1/24
#sample frequency per hour
fs=100
#Duration in hours
dur=24
#A time vector divided by fs
t = seq(0, dur, 1/fs)
#Make a sine wave (with amplitude =1) for each time point in t
u = sin(2*pi*cf*t)
#Make a cosine wave (with amplitude =1) for each time point in t
u2= cos(2*pi*cf*t)
#Plot the waves
plot(x=t,y=u, type='l',col='darkblue',xlab='time(hours)',ylab='Amplitude')
text(x=1+t[1000],y=-0.2+u[500],labels='y=sin(2*pi*cf*t)',col='darkblue')
```

```
lines(x=t,y=u2, type='l',col='darkred')
text(x=-1+t[500],y=-0.2+u2[500],labels='Y=cos(2*pi*cf*t)',col='darkred')
```



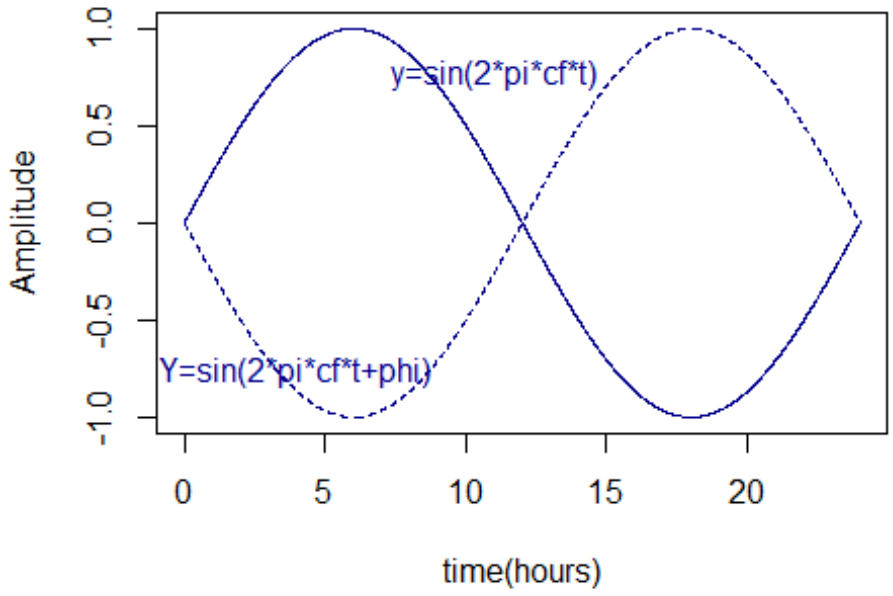
As can be seen, the sine function is a time shifted version of the cosine function and vice versa. The time shift is also called phase. We can add a constant for the phase (phi) to the sine/cosine wave function.

$$y(t) = \beta sin(2\pi ft + \phi)$$

Plotting circadian sine wave with phase shift

```
#cycle frequency (in this case per hour) - one cycle per 24 hours
cf=1/24
#sample frequency per hour
fs=100
#Duration in hours
dur=24
#A time vector divided by fs
t = seq(0, dur, 1/fs)
#a  phase shift of pi/2 radians (half a cycle) This could be any number
phi=pi/2
#Make a sine wave (with amplitude =1)
u = sin(2*pi*cf*t)
#Make a sine wave (with amplitude =1), and phase shift
u2= cos(2*pi*cf*t+phi)
#Plot the waves
plot(x=t,y=u, type='l',col='darkblue',xlab='time(hours)',ylab='Amplitude')
```

```
text(x=1+t[1000],y=-0.2+u[500],labels='y=sin(2*pi*cf*t)',col='darkblue')
lines(x=t,y=u2, type='l',lty='dashed',col='darkblue')
text(x=-1+t[500],y=0.2+u2[500],labels='Y=sin(2*pi*cf*t+phi)',col='darkblue')
```



Using the trigonometric identity ^caption should be sine and cosine

$$sin(A + B) = sin(A)cos(B) - cos(A)sin(B)$$

we can rewrite the sine function (including phase) as

$$y(t) = \beta_1 sin(2\pi ft) + \beta_2 cos(2\pi ft)$$

^so we have a beta estimate for sin and one for cos where

$$\beta_1 = \beta cos(\phi), \beta_2 = -\beta sin(\phi)$$

We can use the rewritten sine function in a linear regression analysis, where we estimate the best fitting B1 and B2. This will yield a composite estimate of the amplitude and the phase of the data.

This will allow us to use sine and cosine waves to model a circadian rhythm, even if we don't know when it peaks. The amplitude will be given by

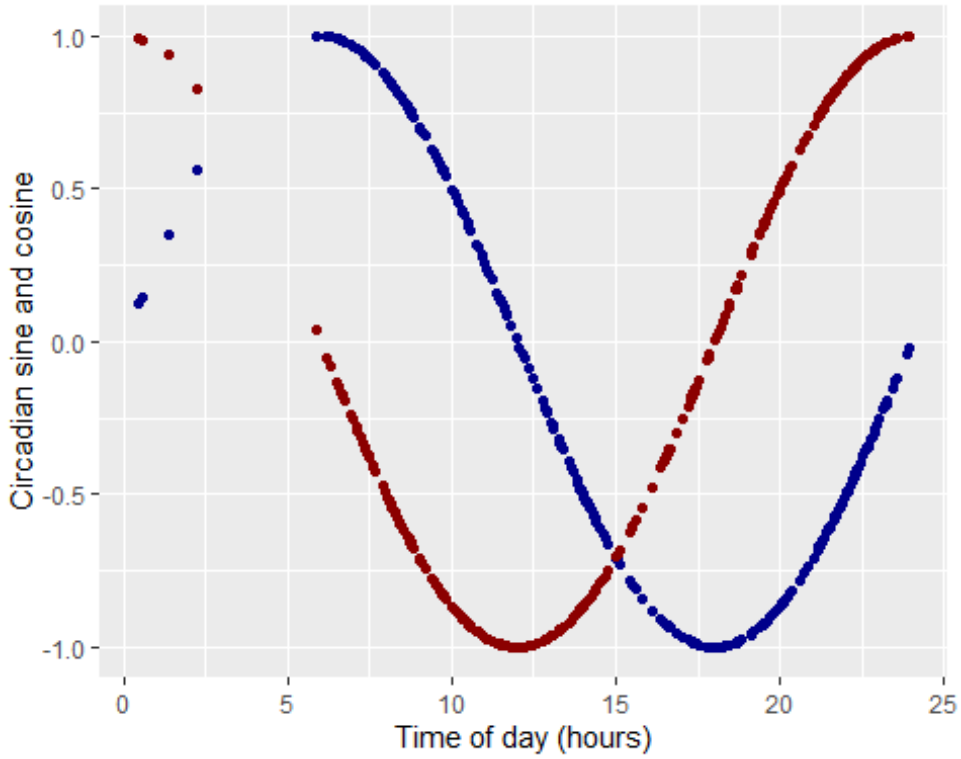
$$\beta = \sqrt{\beta_1^2 + \beta_2^2}$$

The phase (phi) will be given by

$$\phi = atan2(\beta_1, \beta_2)$$

Use the sine and cosine waves to make 24 hour oscillation predictors for the BodyFeelingExp data

```
#cycle frequency (in this case per hour) - one cycle per 24 hours
cf=1/24

#Make sine and cosine waves for each time point present in the data

dataBFE$sinCirc<-sin(2*pi*cf*dataBFE$hour2)
dataBFE$cosCirc<-cos(2*pi*cf*dataBFE$hour2)

# Plot the predictors for each data point in the data
ggplot(dataBFE, aes(x=hour2,y=sinCirc))+
      geom_point(col='darkblue')+
      geom_point(aes(y=cosCirc),col='darkred')+
      ylab('Circadian sine and cosine')+xlab('Time of day (hours)')
```

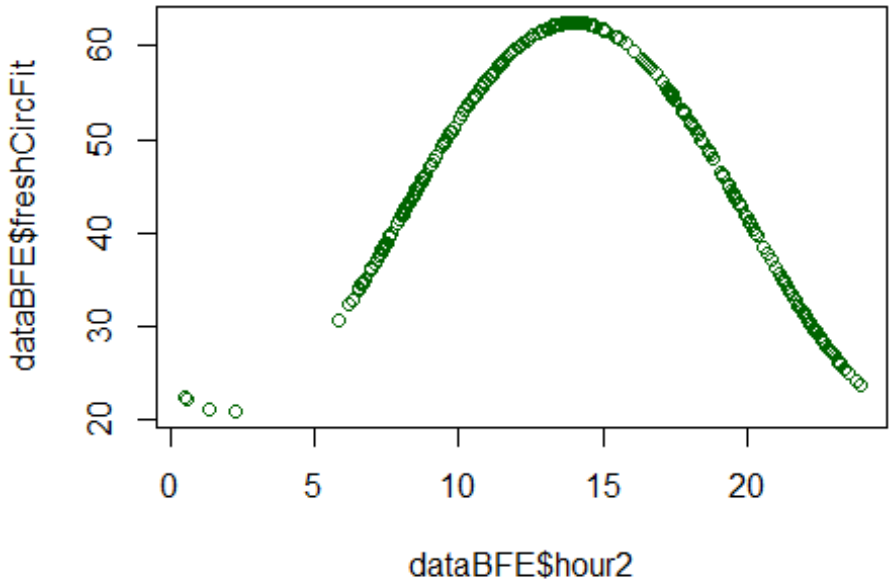


Fitting 24 hour Oscillation model

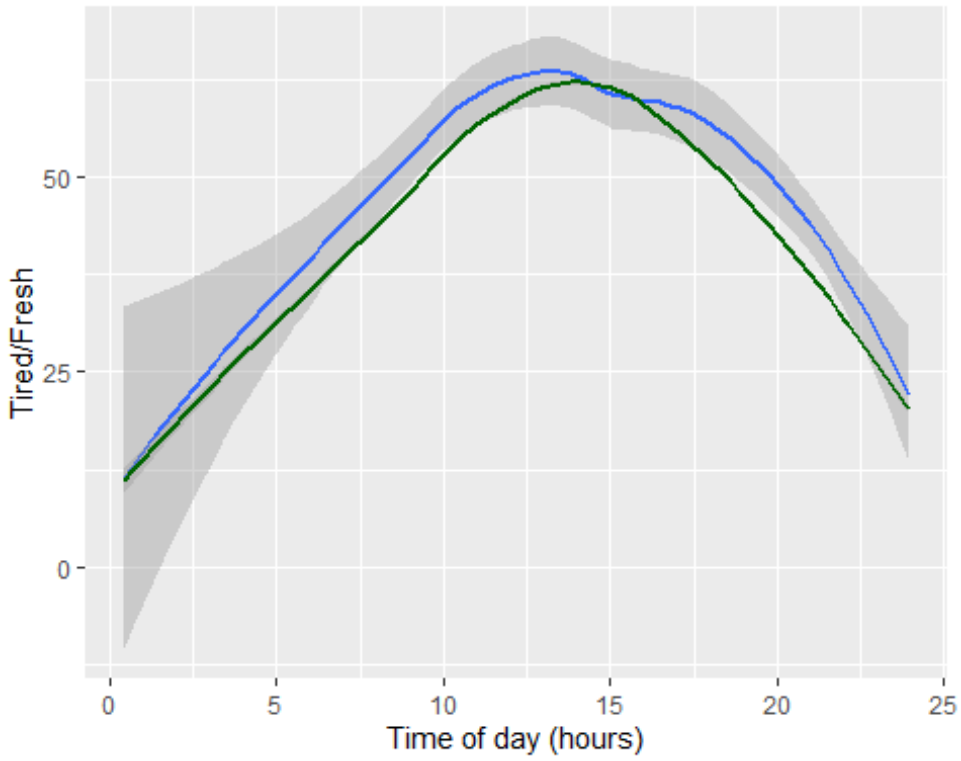
Freshness

```
library(lmerTest)
# Freshness: Simple oscillation model
modelBFEfreshCirc<-lmer(fresh~sinCirc+cosCirc+(1|id),data=dataBFE)
```

```
m_temp<-summary(modelBFEfreshCirc)
m_temp

## Linear mixed model fit by REML. t-tests use Satterthwaite's method [
## lmerModLmerTest]
## Formula: fresh ~ sinCirc + cosCirc + (1 | id)
##    Data: dataBFE
##
## REML criterion at convergence: 3529.8
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -2.4909 -0.7042  0.0631  0.6968  3.6718
##
## Random effects:
##  Groups   Name        Variance Std.Dev.
##  id       (Intercept)  85.86    9.266
##  Residual             415.15   20.375
## Number of obs: 396, groups:  id, 26
##
## Fixed effects:
##             Estimate Std. Error      df t value Pr(>|t|)
## (Intercept)   41.667      2.402  24.562  17.349 2.72e-15 ***
## sinCirc      -10.372      1.698 381.706  -6.107 2.50e-09 ***
## cosCirc      -18.068      1.589 381.510 -11.373  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##         (Intr) sinCrc
## sinCirc 0.234
## cosCirc 0.179  0.346

dataBFE$freshCircFit<-
m_temp$coefficients[1,1]+m_temp$coefficients[2,1]*dataBFE$sinCirc+m_temp$coefficie
nts[3,1]*dataBFE$cosCirc

plot(x=dataBFE$hour2,y=dataBFE$freshCircFit,type='p',col='darkgreen')
```




```
ggplot(dataBFE,aes(x=hour2,y=fresh))+geom_smooth()+geom_smooth(aes(x=hour2,y=fresh
CircFit),col='darkgreen')+labs(x='Time of day (hours)', y='Tired/Fresh')

## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```

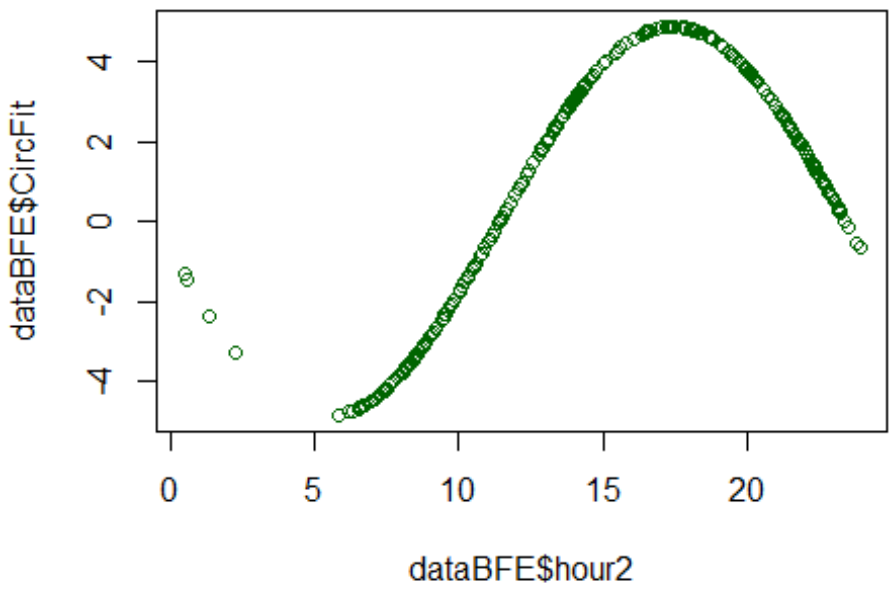


Mood

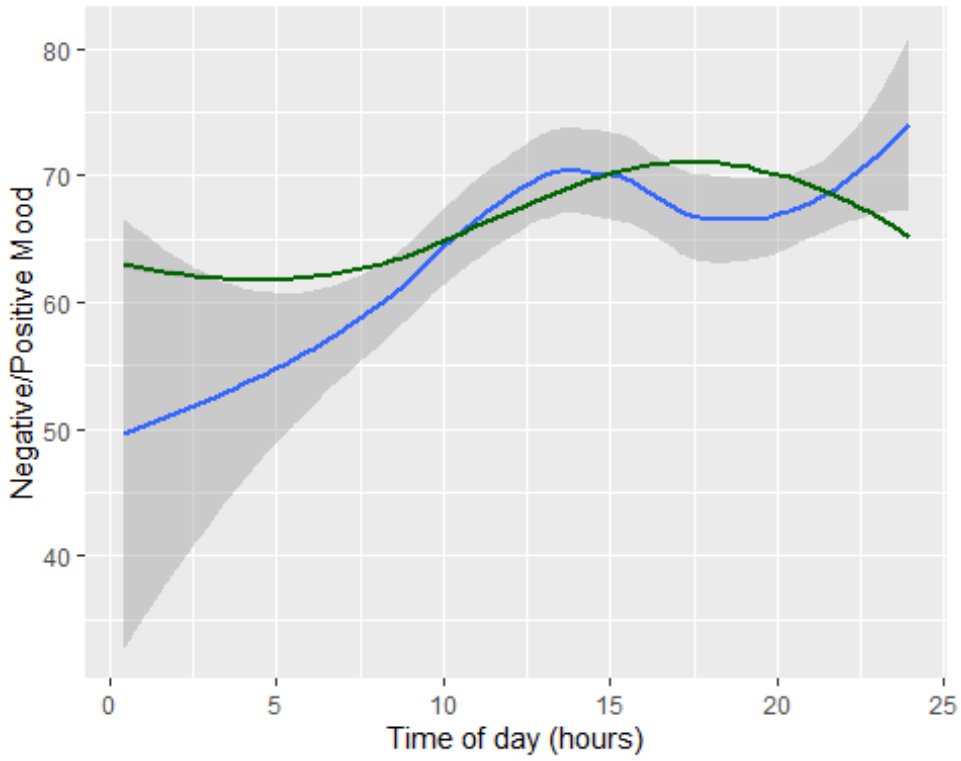
```
library(lmerTest)
# Mood: Simple oscillation model
modelBFEmoodCirc<-lmer(mood~sinCirc+cosCirc+(1|id),data=dataBFE)
m_temp<-summary(modelBFEmoodCirc)
m_temp

## Linear mixed model fit by REML. t-tests use Satterthwaite's method [
## lmerModLmerTest]
## Formula: mood ~ sinCirc + cosCirc + (1 | id)
##    Data: dataBFE
##
## REML criterion at convergence: 3313.3
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -3.3435 -0.5886  0.1779  0.6561  2.7145
##
## Random effects:
##  Groups   Name        Variance Std.Dev.
##  id       (Intercept)  74.81    8.649
##  Residual             235.10   15.333
## Number of obs: 396, groups:  id, 26
##
## Fixed effects:
```

```
##             Estimate Std. Error       df t value Pr(>|t|)
## (Intercept)  65.1079     2.1075  26.9672  30.893  < 2e-16 ***
## sinCirc      -4.8159     1.2807 380.9061  -3.760 0.000196 ***
## cosCirc      -0.7357     1.1979 380.1359  -0.614 0.539470
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##         (Intr) sinCrc
## sinCirc 0.209
## cosCirc 0.157  0.345
```

```
dataBFE$CircFit<-
m_temp$coefficients[2,1]*dataBFE$sinCirc+m_temp$coefficients[3,1]*dataBFE$cosCirc

plot(x=dataBFE$hour2,y=dataBFE$CircFit,type='p',col='darkgreen')
```




```
ggplot(dataBFE,aes(x=hour2,y=mood))+geom_smooth()+geom_smooth(aes(x=hour2,y=CircFi
t+mean(mood)),col='darkgreen')+labs(x='Time of day (hours)', y='Negative/Positive
Mood')

## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```



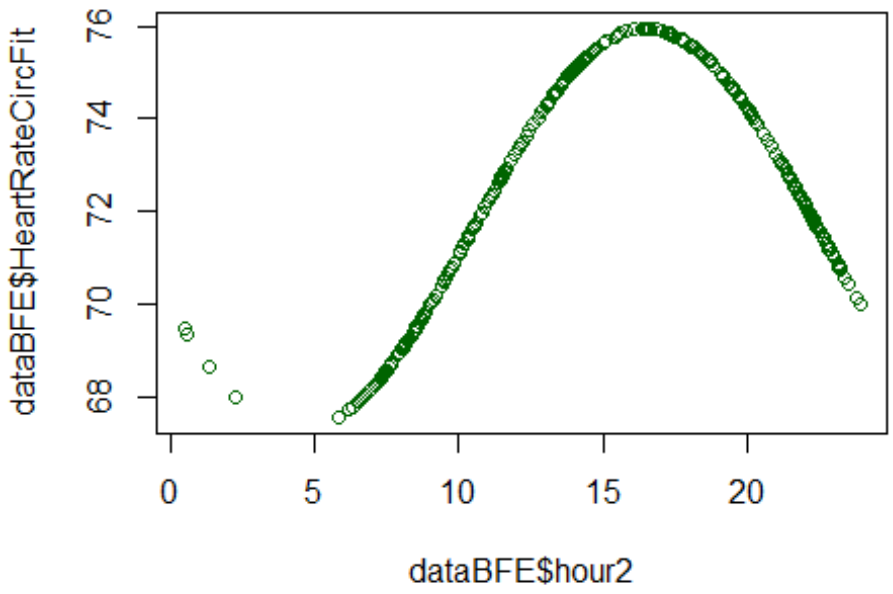
HeartRate

```
# Freshness: Simple oscillation model
modelBFEHeartRateCirc<-lmer(HeartRate~sinCirc+cosCirc+(1|id),data=dataBFE)
m_temp<-summary(modelBFEHeartRateCirc)
m_temp
```

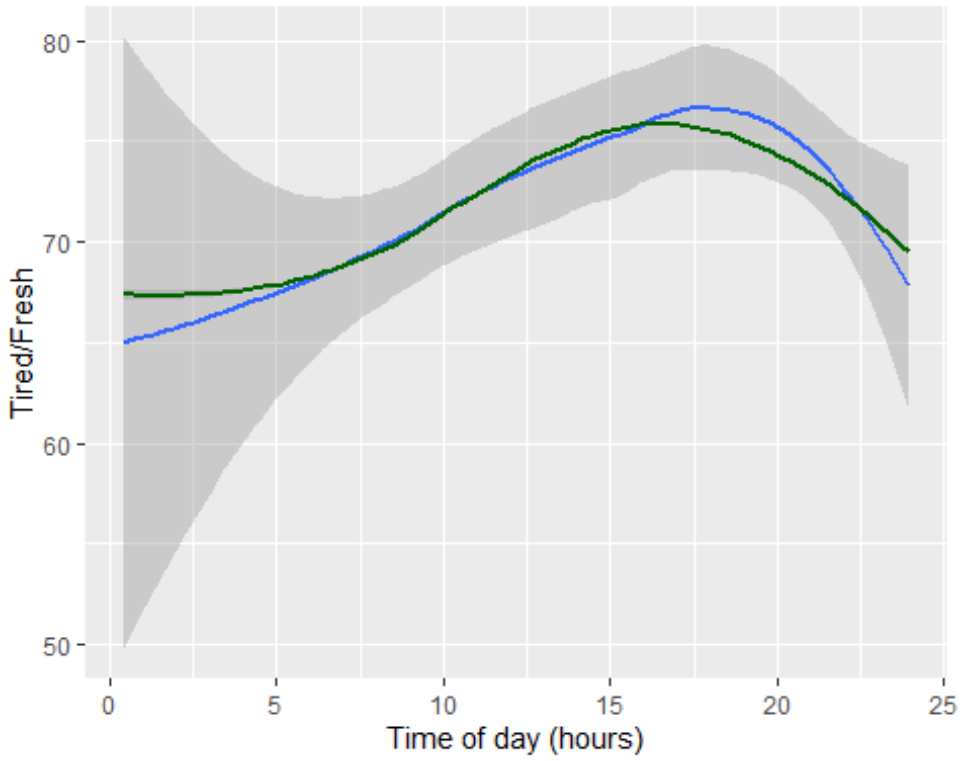
```
## Linear mixed model fit by REML. t-tests use Satterthwaite's method [
## lmerModLmerTest]
## Formula: HeartRate ~ sinCirc + cosCirc + (1 | id)
##    Data: dataBFE
##
## REML criterion at convergence: 3220.5
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -1.8886 -0.5738 -0.1506  0.3482  6.5459
##
## Random effects:
##  Groups   Name        Variance Std.Dev.
##  id       (Intercept)  76.41    8.741
##  Residual             183.45   13.544
## Number of obs: 396, groups:  id, 26
##
## Fixed effects:
##             Estimate Std. Error      df t value Pr(>|t|)
```

```
## (Intercept)   71.606      2.061  22.888  34.735  < 2e-16 ***
## sinCirc       -3.995      1.133 377.383  -3.527 0.000473 ***
## cosCirc       -1.679      1.059 376.051  -1.585 0.113893
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##         (Intr) sinCrc
## sinCirc 0.194
## cosCirc 0.145  0.345
```

```
dataBFE$HeartRateCircFit<-
m_temp$coefficients[1,1]+m_temp$coefficients[2,1]*dataBFE$sinCirc+m_temp$coefficie
nts[3,1]*dataBFE$cosCirc

plot(x=dataBFE$hour2,y=dataBFE$HeartRateCircFit,type='p',col='darkgreen')
```

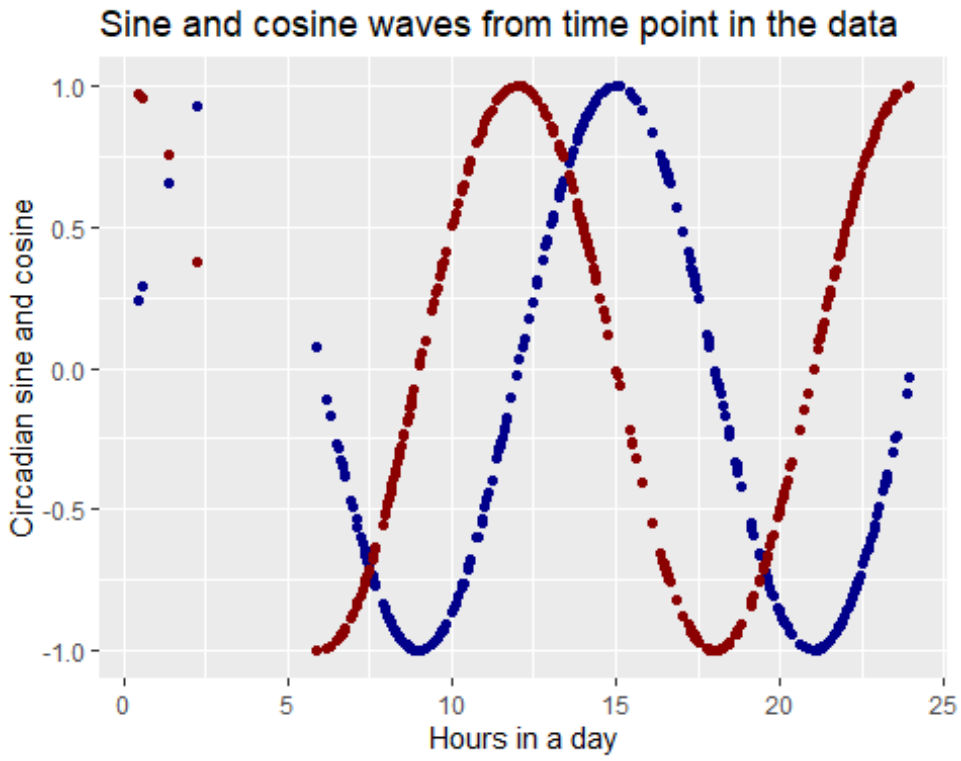



```
ggplot(dataBFE,aes(x=hour2,y=HeartRate))+geom_smooth()+geom_smooth(aes(x=hour2,y=H
eartRateCircFit),col='darkgreen')+labs(x='Time of day (hours)', y='Tired/Fresh')

## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```



Make a 12 hour oscillation model to compare with the 24 hour model

```
#cycle frequency (in this case per hour) - one cycle per 24 hours
cf=1/12

#Make sine and cosine waves for each time point present in the data

dataBFE$sinCirc12<-sin(2*pi*cf*dataBFE$hour2)
dataBFE$cosCirc12<-cos(2*pi*cf*dataBFE$hour2)

# Plot the predictors for each data point in the data
ggplot(dataBFE, aes(x=hour2,y=sinCirc12))+
      geom_point(col='darkblue')+
      geom_point(aes(y=cosCirc12),col='darkred')+
      ylab('sinCirc/cosCirc')+
      ggtitle('Sine and cosine waves from time point in the data')+
      xlab('Hours in a day')+
      ylab('Circadian sine and cosine')
```



Fitting 12 hour Oscillation models

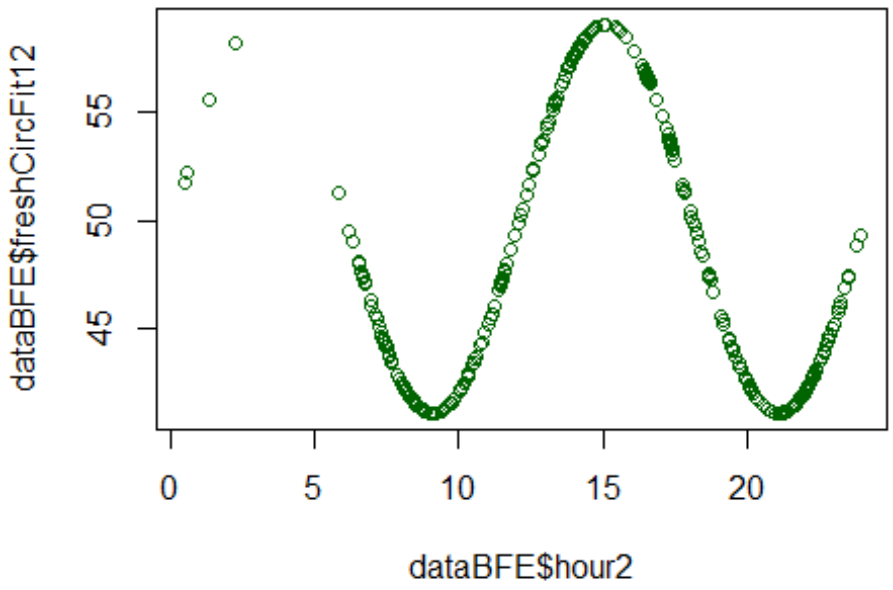
Freshness

```
# Freshness: Simple oscillation model
modelBFEfreshCirc12<-lmer(fresh~sinCirc12+cosCirc12+(1|id),data=dataBFE)
m_temp<-summary(modelBFEfreshCirc12)
m_temp
```

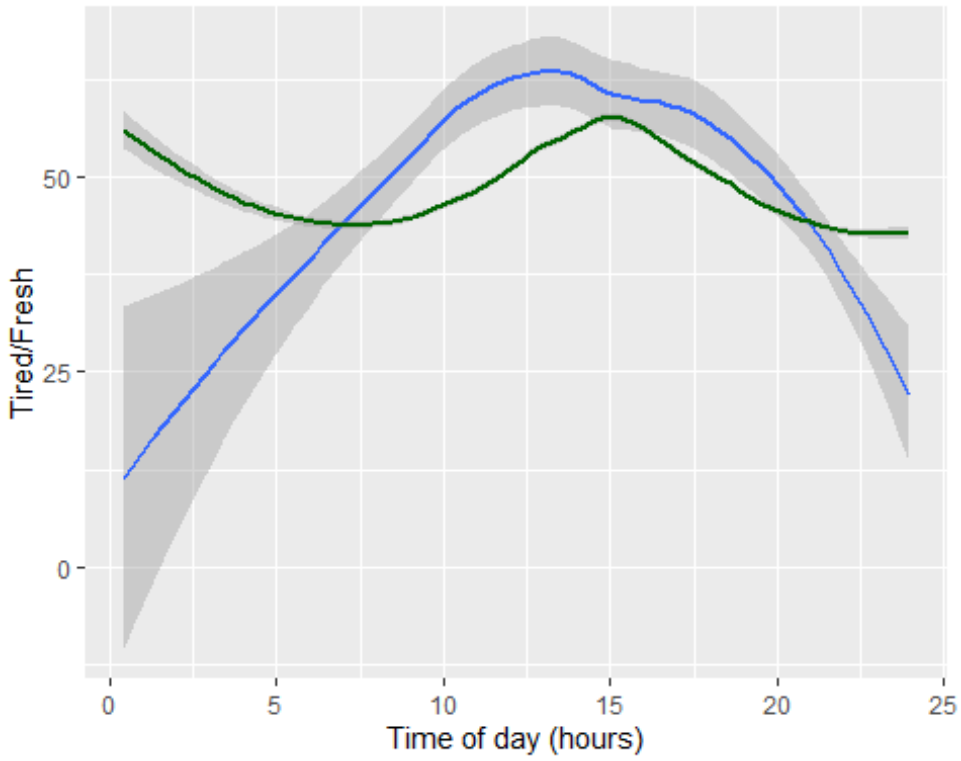
```
## Linear mixed model fit by REML. t-tests use Satterthwaite's method [
## lmerModLmerTest]
## Formula: fresh ~ sinCirc12 + cosCirc12 + (1 | id)
##    Data: dataBFE
##
## REML criterion at convergence: 3616.8
##
## Scaled residuals:
##      Min       1Q   Median       3Q      Max
## -2.22990 -0.78648  0.02404  0.76913  2.63128
##
## Random effects:
##  Groups   Name        Variance Std.Dev.
##  id       (Intercept)  62.69    7.917
##  Residual             528.09   22.980
## Number of obs: 396, groups:  id, 26
```

```
##
## Fixed effects:
##             Estimate Std. Error       df t value Pr(>|t|)
## (Intercept)  50.0578     2.2018  24.0518  22.735  < 2e-16 ***
## sinCirc12     8.9771     1.7016 386.1361   5.276 2.21e-07 ***
## cosCirc12    -0.4856     1.7247 383.0018  -0.282    0.778
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##           (Intr) snCr12
## sinCirc12  0.174
## cosCirc12 -0.015 -0.025
```

```
dataBFE$freshCircFit12<-
m_temp$coefficients[1,1]+m_temp$coefficients[2,1]*dataBFE$sinCirc12+m_temp$coeffic
ients[3,1]*dataBFE$cosCirc12

plot(x=dataBFE$hour2,y=dataBFE$freshCircFit12,type='p',col='darkgreen')
```




```
ggplot(dataBFE,aes(x=hour2,y=fresh))+geom_smooth()+geom_smooth(aes(x=hour2,y=fresh
CircFit12),col='darkgreen')+labs(x='Time of day (hours)', y='Tired/Fresh')

## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```

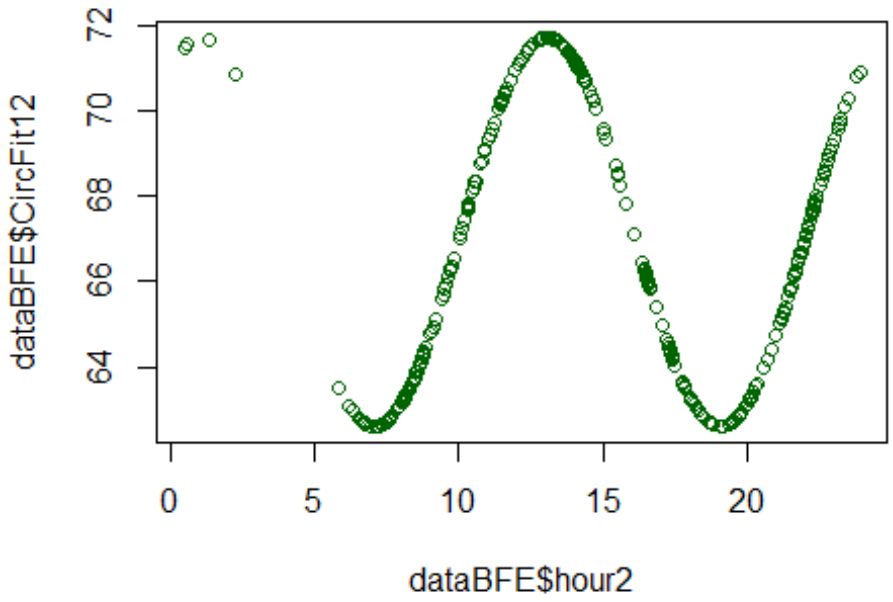


Mood

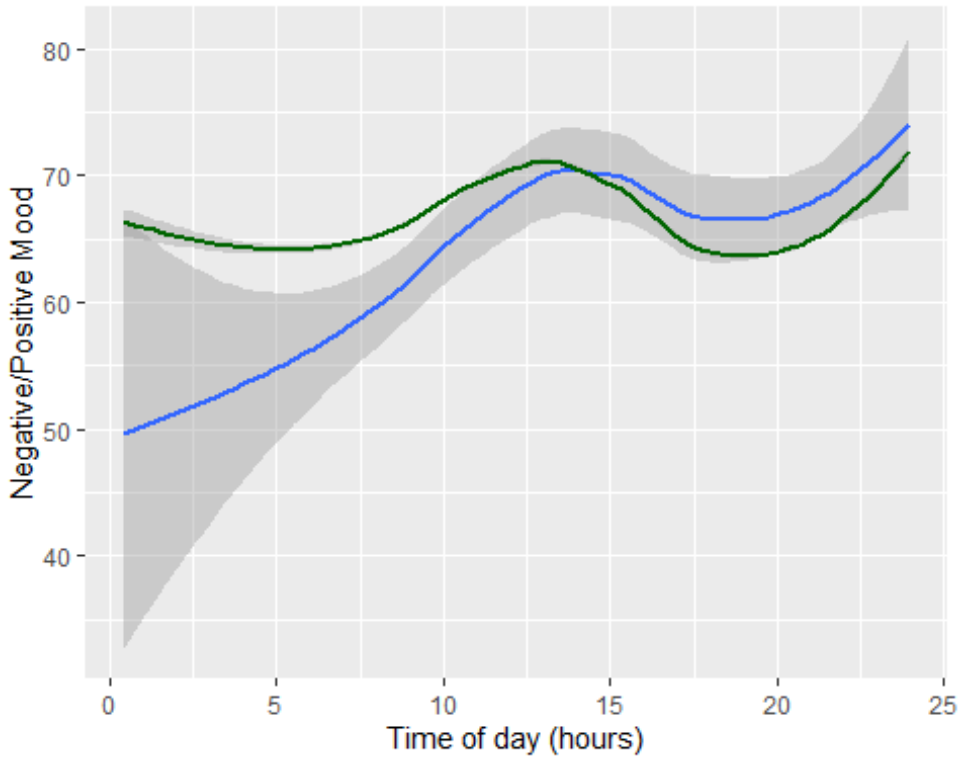
```
library(lmerTest)
# Mood: Simple oscillation model
modelBFEmoodCirc12<-lmer(mood~sinCirc12+cosCirc12+(1|id),data=dataBFE)
m_temp<-summary(modelBFEmoodCirc12)
m_temp

## Linear mixed model fit by REML. t-tests use Satterthwaite's method [
## lmerModLmerTest]
## Formula: mood ~ sinCirc12 + cosCirc12 + (1 | id)
##    Data: dataBFE
##
## REML criterion at convergence: 3312.1
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -3.4374 -0.6323  0.1300  0.6643  2.6313
##
## Random effects:
##  Groups   Name        Variance Std.Dev.
##  id       (Intercept)  80.84    8.991
##  Residual             233.43   15.278
## Number of obs: 396, groups:  id, 26
##
## Fixed effects:
```

```
##             Estimate Std. Error      df t value Pr(>|t|)
## (Intercept)   67.158      2.124  25.357  31.613  < 2e-16 ***
## sinCirc12      2.426      1.139 382.462   2.130 0.033793 *
## cosCirc12      3.846      1.152 379.396   3.339 0.000924 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##           (Intr) snCr12
## sinCirc12  0.103
## cosCirc12  0.005 -0.028
```

```
dataBFE$CircFit12<-
m_temp$coefficients[1,1]+m_temp$coefficients[2,1]*dataBFE$sinCirc12+m_temp$coeffic
ients[3,1]*dataBFE$cosCirc12

plot(x=dataBFE$hour2,y=dataBFE$CircFit12,type='p',col='darkgreen')
```




```
ggplot(dataBFE,aes(x=hour2,y=mood))+geom_smooth()+geom_smooth(aes(x=hour2,y=CircFi
t12),col='darkgreen')+labs(x='Time of day (hours)', y='Negative/Positive Mood')

## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```



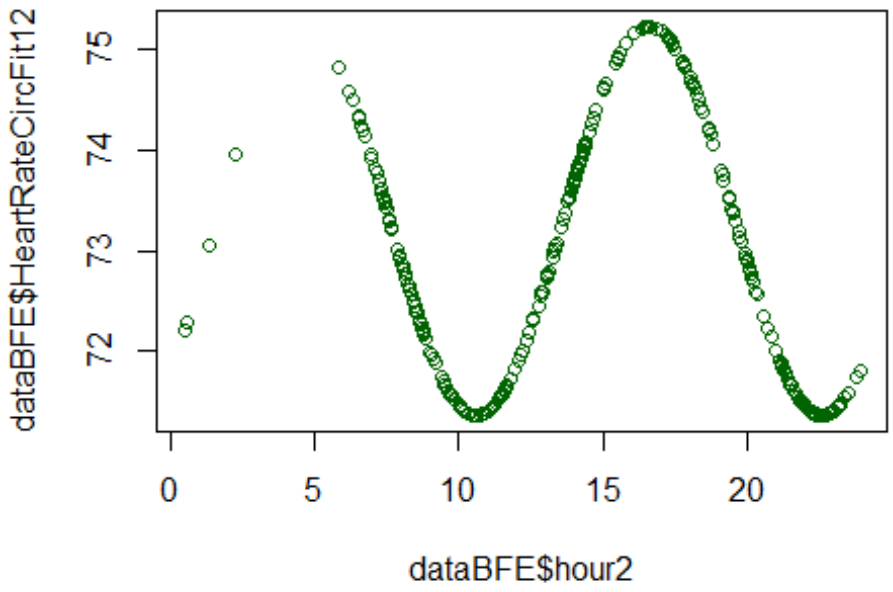
HeartRate

```
# Freshness: Simple oscillation model
modelBFEHeartRateCirc12<-lmer(HeartRate~sinCirc12+cosCirc12+(1|id),data=dataBFE)
m_temp<-summary(modelBFEHeartRateCirc12)
m_temp
```

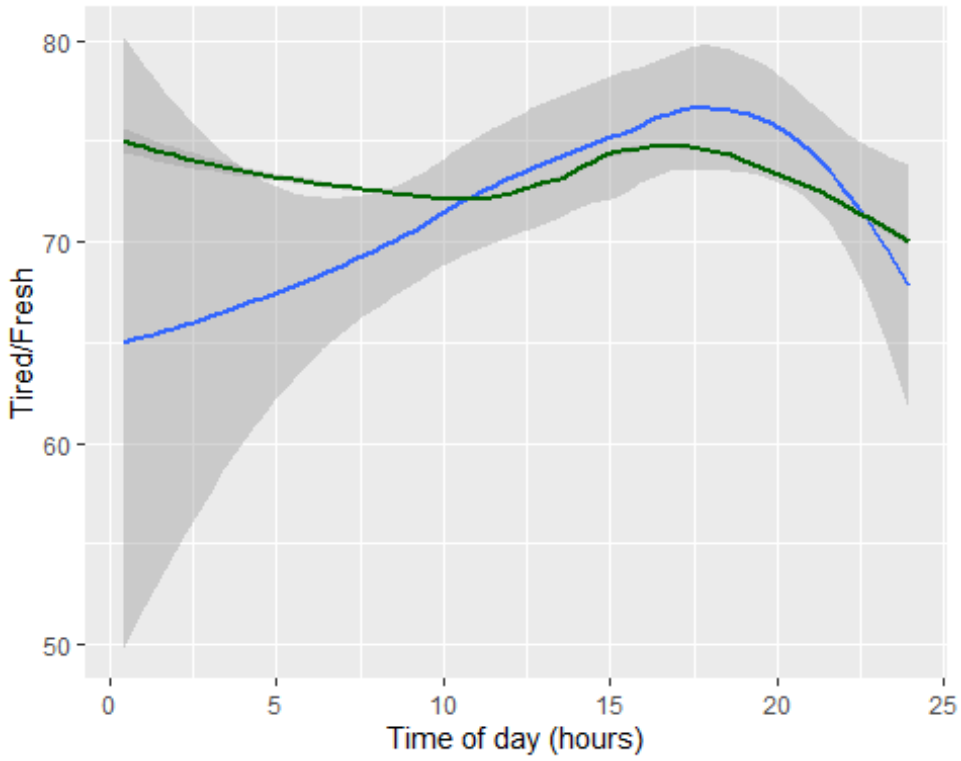
```
## Linear mixed model fit by REML. t-tests use Satterthwaite's method [
## lmerModLmerTest]
## Formula: HeartRate ~ sinCirc12 + cosCirc12 + (1 | id)
##    Data: dataBFE
##
## REML criterion at convergence: 3229.6
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -2.1106 -0.5859 -0.1611  0.3914  6.5602
##
## Random effects:
##  Groups   Name        Variance Std.Dev.
##  id       (Intercept)  77.19    8.786
##  Residual             187.82   13.705
## Number of obs: 396, groups:  id, 26
##
## Fixed effects:
##             Estimate Std. Error      df t value Pr(>|t|)
```

```
## (Intercept)   73.294      2.037  21.859  35.983   <2e-16 ***
## sinCirc12      1.283      1.023 379.884   1.254    0.210
## cosCirc12     -1.434      1.034 376.395  -1.387    0.166
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##             (Intr) snCr12
## sinCirc12  0.093
## cosCirc12  0.007 -0.029
```

```
dataBFE$HeartRateCircFit12<-
m_temp$coefficients[1,1]+m_temp$coefficients[2,1]*dataBFE$sinCirc12+m_temp$coeffic
ients[3,1]*dataBFE$cosCirc12

plot(x=dataBFE$hour2,y=dataBFE$HeartRateCircFit12,type='p',col='darkgreen')
```




```
ggplot(dataBFE,aes(x=hour2,y=HeartRate))+geom_smooth()+geom_smooth(aes(x=hour2,y=H
eartRateCircFit12),col='darkgreen')+labs(x='Time of day (hours)', y='Tired/Fresh')

## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```



Combining 12 and 24 hours

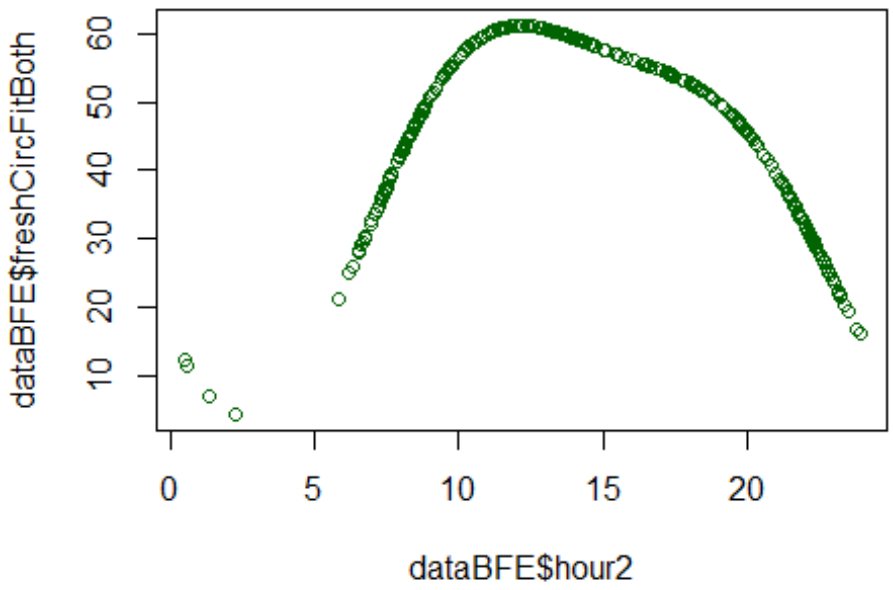
Freshness

```
# Freshness: Simple oscillation model
modelBFEfreshCircBoth<-
lmer(fresh~sinCirc+cosCirc+sinCirc12+cosCirc12+(1|id),data=dataBFE)
m_temp<-summary(modelBFEfreshCircBoth)
m_temp
```

```
## Linear mixed model fit by REML. t-tests use Satterthwaite's method [
## lmerModLmerTest]
## Formula: fresh ~ sinCirc + cosCirc + sinCirc12 + cosCirc12 + (1 | id)
##    Data: dataBFE
##
## REML criterion at convergence: 3513.7
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -2.4633 -0.6868  0.0593  0.6881  3.7693
##
## Random effects:
##  Groups   Name        Variance Std.Dev.
##  id       (Intercept)  88.28    9.396
##  Residual             406.08   20.151
## Number of obs: 396, groups:  id, 26
```

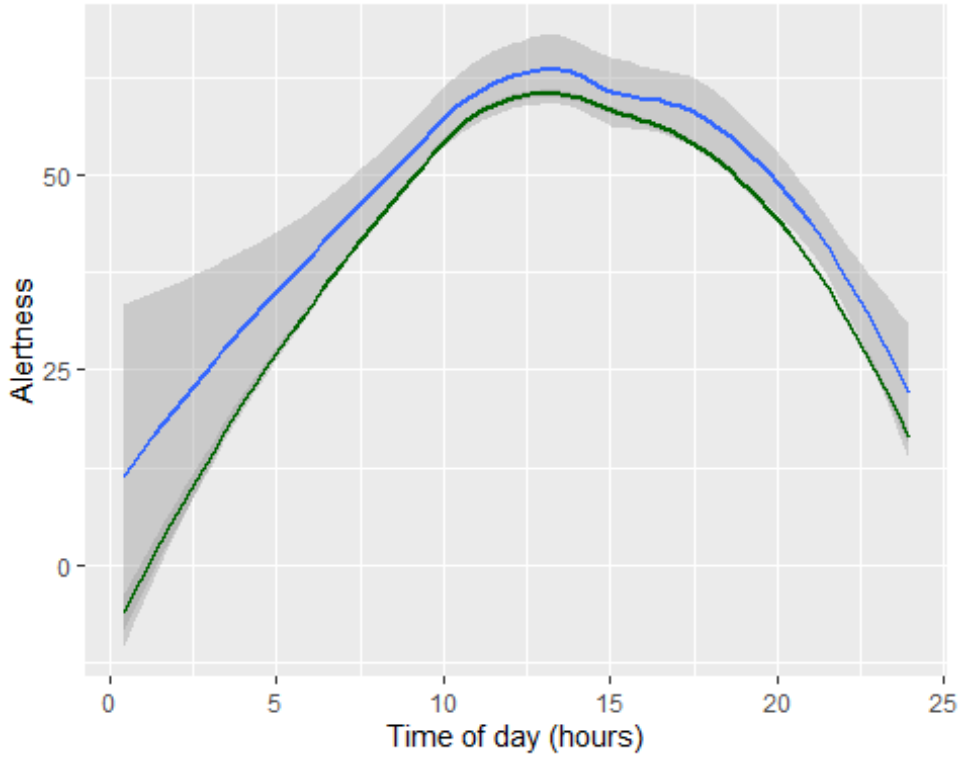
```
##
## Fixed effects:
##             Estimate Std. Error       df t value Pr(>|t|)
## (Intercept)  38.0672     2.6638  37.6164  14.291  < 2e-16 ***
## sinCirc     -14.9801     2.2352 379.9714  -6.702 7.45e-11 ***
## cosCirc     -22.7696     2.1593 381.0939 -10.545  < 2e-16 ***
## sinCirc12    -7.0612     2.2323 383.2458  -3.163  0.00168 **
## cosCirc12     0.3007     1.5352 378.5242   0.196  0.84479
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##           (Intr) sinCrc cosCrc snCr12
## sinCirc    0.434
## cosCirc    0.404  0.636
## sinCirc12  0.420  0.654  0.686
## cosCirc12 -0.057 -0.147 -0.050 -0.094
```

```
dataBFE$freshCircFitBoth<-
m_temp$coefficients[1,1]+m_temp$coefficients[2,1]*dataBFE$sinCirc+m_temp$coefficie
nts[3,1]*dataBFE$cosCirc+m_temp$coefficients[4,1]*dataBFE$sinCirc12+m_temp$coeffic
ients[5,1]*dataBFE$cosCirc12

plot(x=dataBFE$hour2,y=dataBFE$freshCircFitBoth,type='p',col='darkgreen')
```



```
ggplot(dataBFE,aes(x=hour2,y=fresh))+
  geom_smooth()+
  geom_smooth(aes(x=hour2,y=freshCircFitBoth),col='darkgreen')+
  labs(x='Time of day (hours)', y='Alertness')
```

```
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```



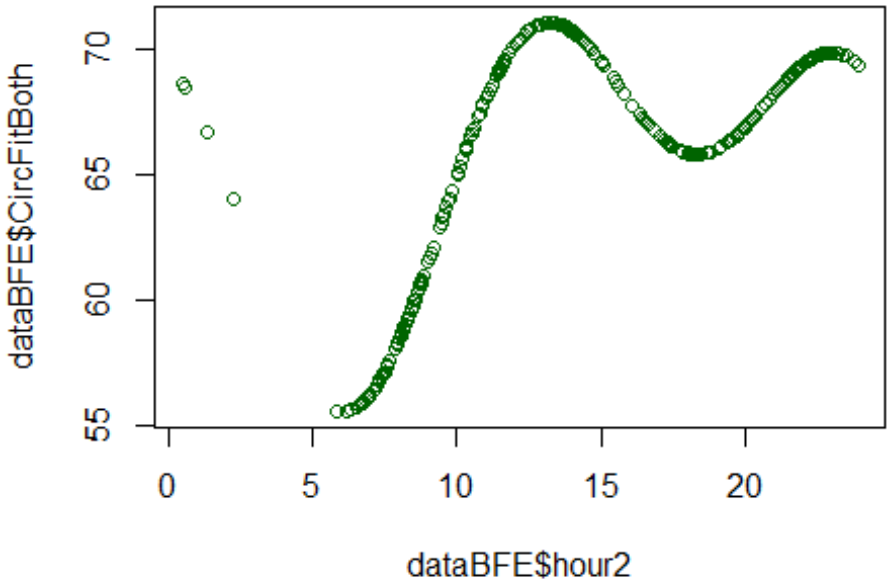
Mood

```
library(lmerTest)
# Mood: Simple oscillation model
modelBFEmoodCircBoth<-
lmer(mood~sinCirc+cosCirc+sinCirc12+cosCirc12+(1|id),data=dataBFE)
m_temp<-summary(modelBFEmoodCircBoth)
m_temp
```

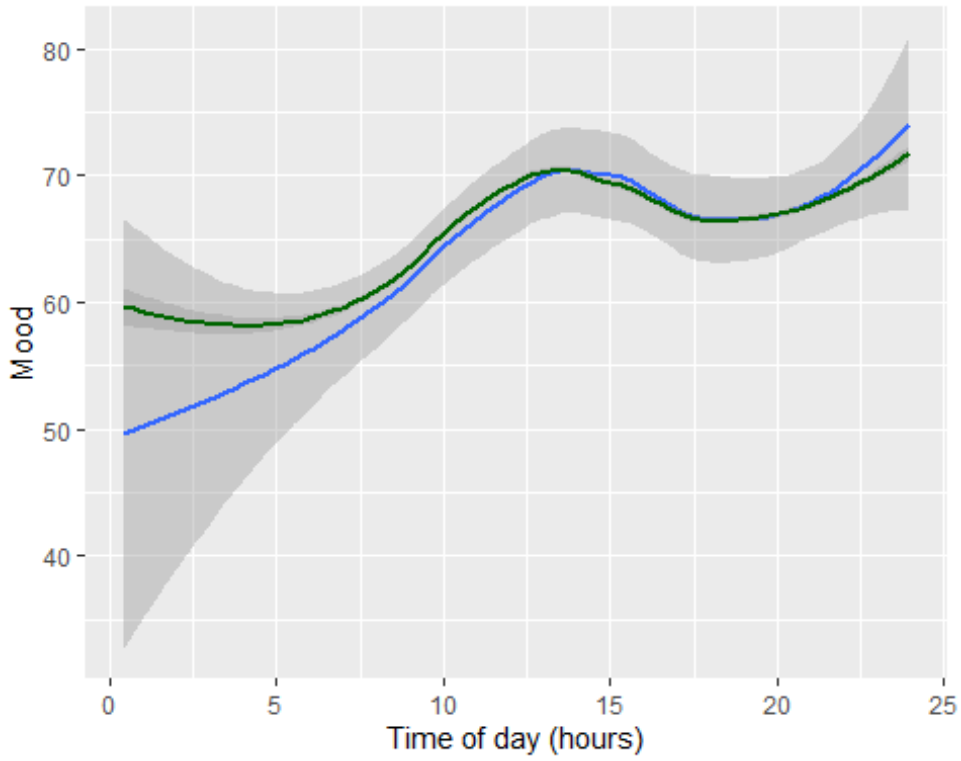
```
## Linear mixed model fit by REML. t-tests use Satterthwaite's method [
## lmerModLmerTest]
## Formula: mood ~ sinCirc + cosCirc + sinCirc12 + cosCirc12 + (1 | id)
##    Data: dataBFE
##
## REML criterion at convergence: 3292.8
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
```

```
## -3.3779 -0.5934  0.1227  0.6882  2.5560
##
## Random effects:
##  Groups   Name        Variance Std.Dev.
##  id       (Intercept)  76.26    8.732
##  Residual             226.65   15.055
## Number of obs: 396, groups:  id, 26
##
## Fixed effects:
##             Estimate Std. Error       df t value Pr(>|t|)
## (Intercept)  65.1934     2.2698  36.1882  28.722  < 2e-16 ***
## sinCirc      -5.1397     1.6733 377.7559  -3.072 0.002283 **
## cosCirc      -0.4126     1.6170 378.5828  -0.255 0.798711
## sinCirc12     0.3546     1.6729 381.0348   0.212 0.832234
## cosCirc12     4.5011     1.1490 377.5299   3.917 0.000106 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##           (Intr) sinCrc cosCrc snCr12
## sinCirc    0.384
## cosCirc    0.355  0.636
## sinCirc12  0.365  0.654  0.686
## cosCirc12 -0.045 -0.147 -0.051 -0.096
```

```
dataBFE$CircFitBoth<-
m_temp$coefficients[1,1]+m_temp$coefficients[2,1]*dataBFE$sinCirc+m_temp$coefficie
nts[3,1]*dataBFE$cosCirc+m_temp$coefficients[4,1]*dataBFE$sinCirc12+m_temp$coeffic
ients[5,1]*dataBFE$cosCirc12

plot(x=dataBFE$hour2,y=dataBFE$CircFitBoth,type='p',col='darkgreen')
```




```
ggplot(dataBFE,aes(x=hour2,y=mood))+geom_smooth()+geom_smooth(aes(x=hour2,y=CircFi
tBoth),col='darkgreen')+labs(x='Time of day (hours)', y='Mood')

## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```



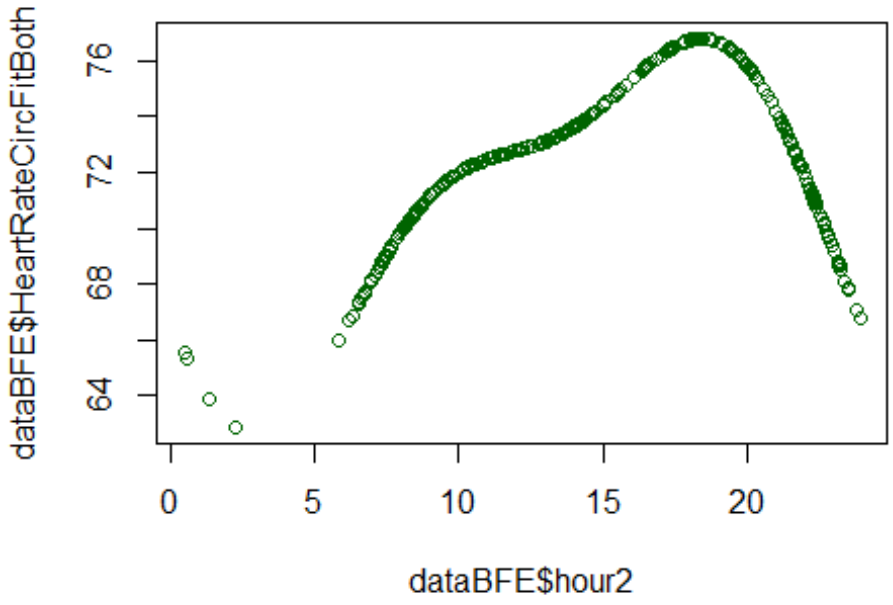
HeartRate

```
# Freshness: Simple oscillation model
modelBFEHeartRateCircBoth<-
lmer(HeartRate~+sinCirc+cosCirc+sinCirc12+cosCirc12+(1|id),data=dataBFE)
m_temp<-summary(modelBFEHeartRateCircBoth)
m_temp
```

```
## Linear mixed model fit by REML. t-tests use Satterthwaite's method [
## lmerModLmerTest]
## Formula: HeartRate ~ +sinCirc + cosCirc + sinCirc12 + cosCirc12 + (1 |      id)
##    Data: dataBFE
##
## REML criterion at convergence: 3213
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -1.8723 -0.5640 -0.1956  0.3550  6.4625
##
## Random effects:
##  Groups   Name        Variance Std.Dev.
##  id       (Intercept)  79.06    8.892
##  Residual             182.70   13.517
## Number of obs: 396, groups:  id, 26
##
## Fixed effects:
##             Estimate Std. Error      df t value Pr(>|t|)
```

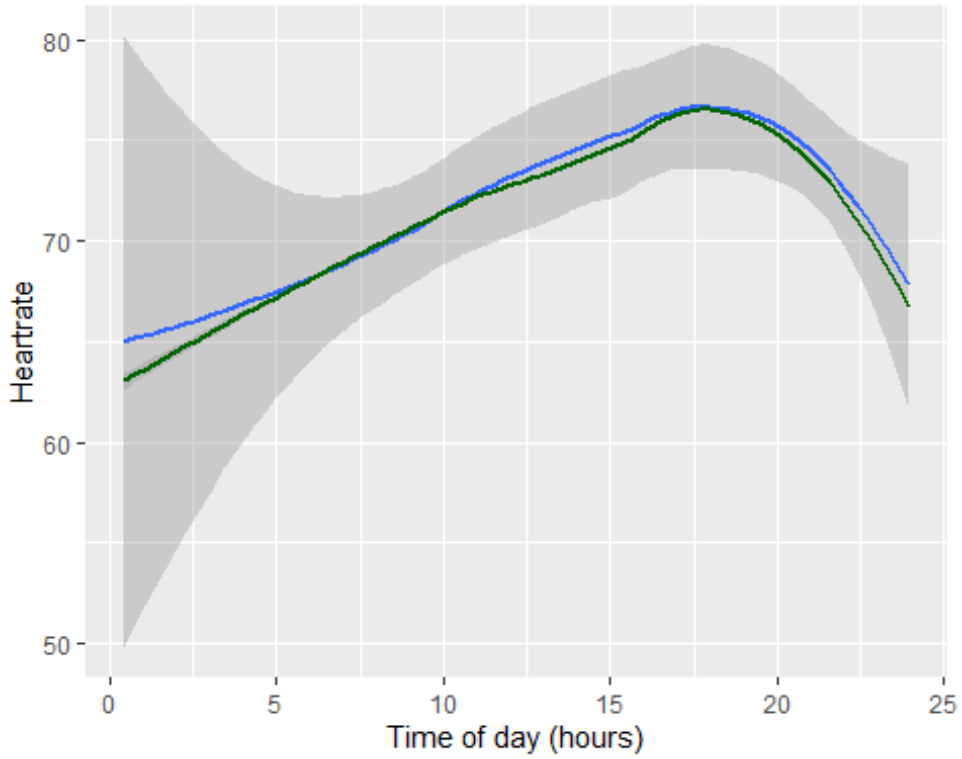
```
## (Intercept)   70.601      2.215  29.567  31.870  < 2e-16 ***
## sinCirc       -5.219      1.504 373.627  -3.471  0.00058 ***
## cosCirc       -3.075      1.453 374.427  -2.116  0.03503 *
## sinCirc12     -2.072      1.504 377.462  -1.377  0.16919
## cosCirc12     -0.903      1.033 374.044  -0.875  0.38239
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##           (Intr) sinCrc cosCrc snCr12
## sinCirc    0.355
## cosCirc    0.327  0.636
## sinCirc12  0.334  0.654  0.686
## cosCirc12 -0.038 -0.147 -0.052 -0.097
```

```
dataBFE$HeartRateCircFitBoth<-
m_temp$coefficients[1,1]+m_temp$coefficients[2,1]*dataBFE$sinCirc+m_temp$coefficie
nts[3,1]*dataBFE$cosCirc++m_temp$coefficients[4,1]*dataBFE$sinCirc12+m_temp$coeffi
cients[5,1]*dataBFE$cosCirc12

plot(x=dataBFE$hour2,y=dataBFE$HeartRateCircFitBoth,type='p',col='darkgreen')
```




```
ggplot(dataBFE,aes(x=hour2,y=HeartRate))+geom_smooth()+geom_smooth(aes(x=hour2,y=H
eartRateCircFitBoth),col='darkgreen')+labs(x='Time of day (hours)', y='Heartrate')
```

```
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```



Model comparisons

Is the 24 hour or the 12 hour model the best?

```
#Comparing models for freshness
anova(modelBFEfreshCirc12,  modelBFEfreshCircBoth)

## refitting model(s) with ML (instead of REML)

## Data: dataBFE
## Models:
## modelBFEfreshCirc12: fresh ~ sinCirc12 + cosCirc12 + (1 | id)
## modelBFEfreshCircBoth: fresh ~ sinCirc + cosCirc + sinCirc12 + cosCirc12 + (1 |
id)
##                        npar    AIC    BIC  logLik deviance  Chisq Df Pr(>Chisq)
## modelBFEfreshCirc12       5 3636.0 3655.9 -1813.0   3626.0
## modelBFEfreshCircBoth     7 3542.8 3570.7 -1764.4   3528.8 97.121  2  < 2.2e-16
##
## modelBFEfreshCirc12
## modelBFEfreshCircBoth ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
anova( modelBFEfreshCirc, modelBFEfreshCircBoth)
```

```
## refitting model(s) with ML (instead of REML)
```

```
## Data: dataBFE
## Models:
## modelBFEfreshCirc: fresh ~ sinCirc + cosCirc + (1 | id)
## modelBFEfreshCircBoth: fresh ~ sinCirc + cosCirc + sinCirc12 + cosCirc12 + (1 |
id)
##                        npar    AIC    BIC  logLik deviance  Chisq Df Pr(>Chisq)
## modelBFEfreshCirc         5 3548.8 3568.7 -1769.4   3538.8
## modelBFEfreshCircBoth     7 3542.8 3570.7 -1764.4   3528.8 9.9679  2   0.006847
##
## modelBFEfreshCirc
## modelBFEfreshCircBoth **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
#Comparing models for mood
anova(modelBFEmoodCirc12, modelBFEmoodCircBoth)
```

```
## refitting model(s) with ML (instead of REML)
```

```
## Data: dataBFE
## Models:
## modelBFEmoodCirc12: mood ~ sinCirc12 + cosCirc12 + (1 | id)
## modelBFEmoodCircBoth: mood ~ sinCirc + cosCirc + sinCirc12 + cosCirc12 + (1 |
id)
##                       npar    AIC    BIC  logLik deviance  Chisq Df Pr(>Chisq)
## modelBFEmoodCirc12       5 3329.6 3349.5 -1659.8   3319.6
## modelBFEmoodCircBoth     7 3319.4 3347.3 -1652.7   3305.4 14.222  2  0.0008161
##
## modelBFEmoodCirc12
## modelBFEmoodCircBoth ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
anova(modelBFEmoodCirc, modelBFEmoodCircBoth)
```

```
## refitting model(s) with ML (instead of REML)
```

```
## Data: dataBFE
## Models:
## modelBFEmoodCirc: mood ~ sinCirc + cosCirc + (1 | id)
## modelBFEmoodCircBoth: mood ~ sinCirc + cosCirc + sinCirc12 + cosCirc12 + (1 |
id)
##                       npar    AIC    BIC  logLik deviance  Chisq Df Pr(>Chisq)
## modelBFEmoodCirc         5 3330.9 3350.8 -1660.5   3320.9
## modelBFEmoodCircBoth     7 3319.4 3347.3 -1652.7   3305.4 15.522  2  0.0004261
##
```

```
## modelBFEmoodCirc
## modelBFEmoodCircBoth ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
#Comparing models for heartrate
anova(modelBFEHeartRateCirc12,modelBFEHeartRateCircBoth)
```

```
## refitting model(s) with ML (instead of REML)

## Data: dataBFE
## Models:
## modelBFEHeartRateCirc12: HeartRate ~ sinCirc12 + cosCirc12 + (1 | id)
## modelBFEHeartRateCircBoth: HeartRate ~ +sinCirc + cosCirc + sinCirc12 +
cosCirc12 + (1 |
## modelBFEHeartRateCircBoth:     id)
##                            npar    AIC    BIC  logLik deviance  Chisq Df
## modelBFEHeartRateCirc12       5 3246.7 3266.6 -1618.3   3236.7
## modelBFEHeartRateCircBoth     7 3238.7 3266.6 -1612.3   3224.7 11.973  2
##                            Pr(>Chisq)
## modelBFEHeartRateCirc12
## modelBFEHeartRateCircBoth    0.002513 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

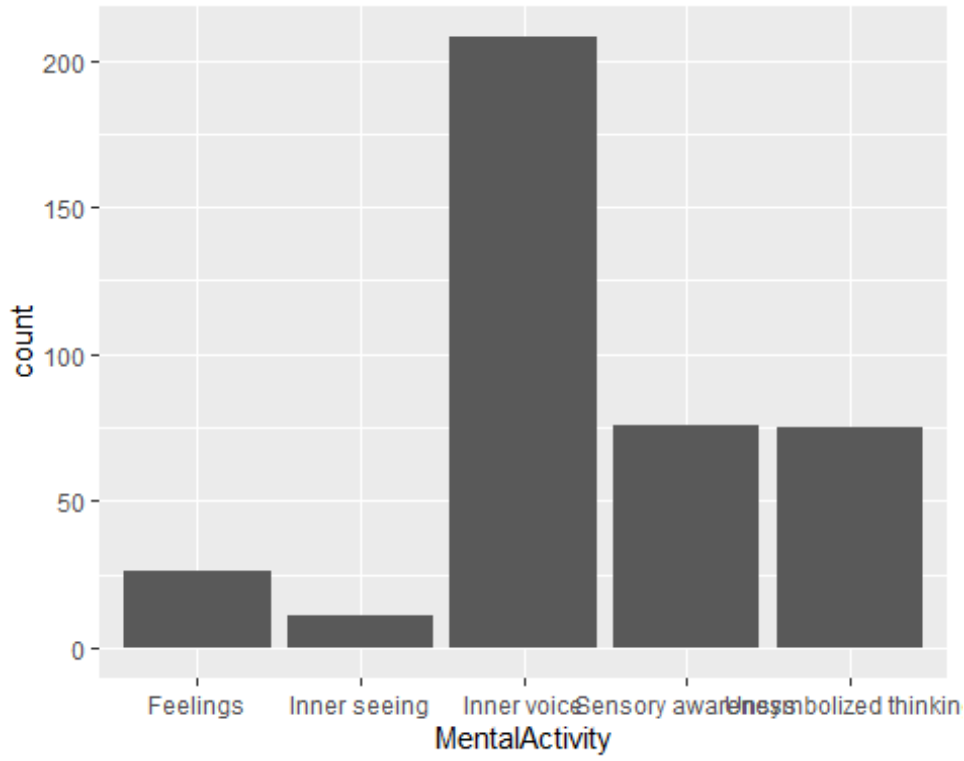
```
anova(modelBFEHeartRateCirc,modelBFEHeartRateCircBoth)
```

```
## refitting model(s) with ML (instead of REML)

## Data: dataBFE
## Models:
## modelBFEHeartRateCirc: HeartRate ~ sinCirc + cosCirc + (1 | id)
## modelBFEHeartRateCircBoth: HeartRate ~ +sinCirc + cosCirc + sinCirc12 +
cosCirc12 + (1 |
## modelBFEHeartRateCircBoth:     id)
##                            npar    AIC    BIC  logLik deviance  Chisq Df
## modelBFEHeartRateCirc         5 3237.6 3257.5 -1613.8   3227.6
## modelBFEHeartRateCircBoth     7 3238.7 3266.6 -1612.3   3224.7 2.9091  2
##                            Pr(>Chisq)
## modelBFEHeartRateCirc
## modelBFEHeartRateCircBoth     0.2335
```

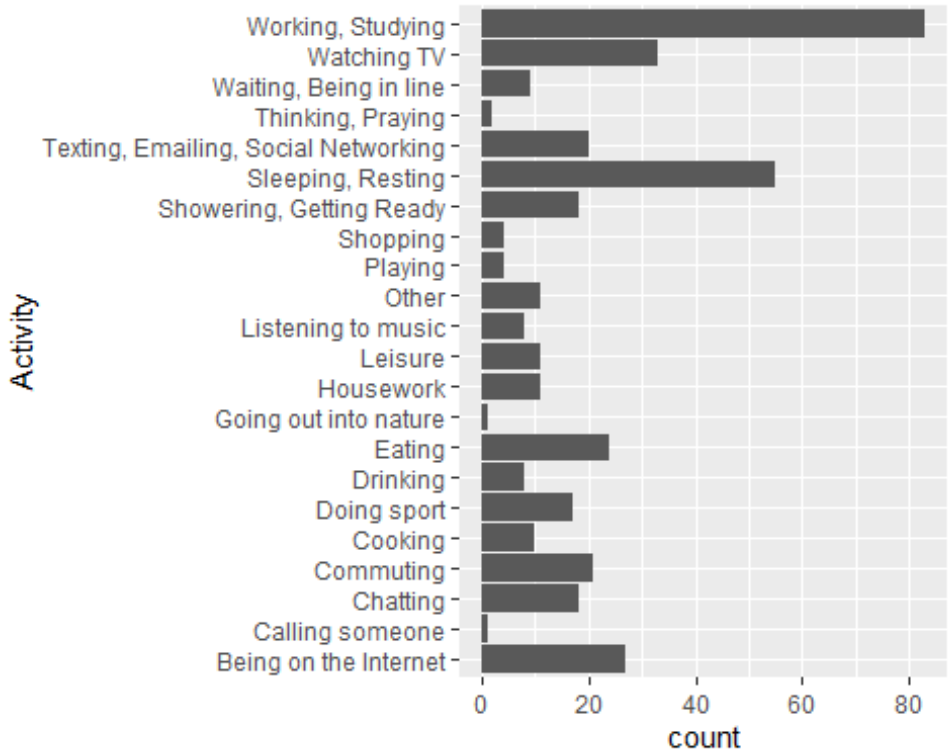
```
#effect(modelBFEHeartRateCirc)
```

Activity and Mental activity

```
library(ggplot2)
ggplot(dataBFE,aes(x=MentalActivity))+geom_bar()
```




```
ggplot(dataBFE,aes(x=Activity))+geom_bar()+coord_flip()
```



```r
#Mood as function of Mental activity
modelBFEmoodMentalAct<-lmer(mood~MentalActivity + (1|id),data=dataBFE)
anova(modelBFEmoodMentalAct)
```

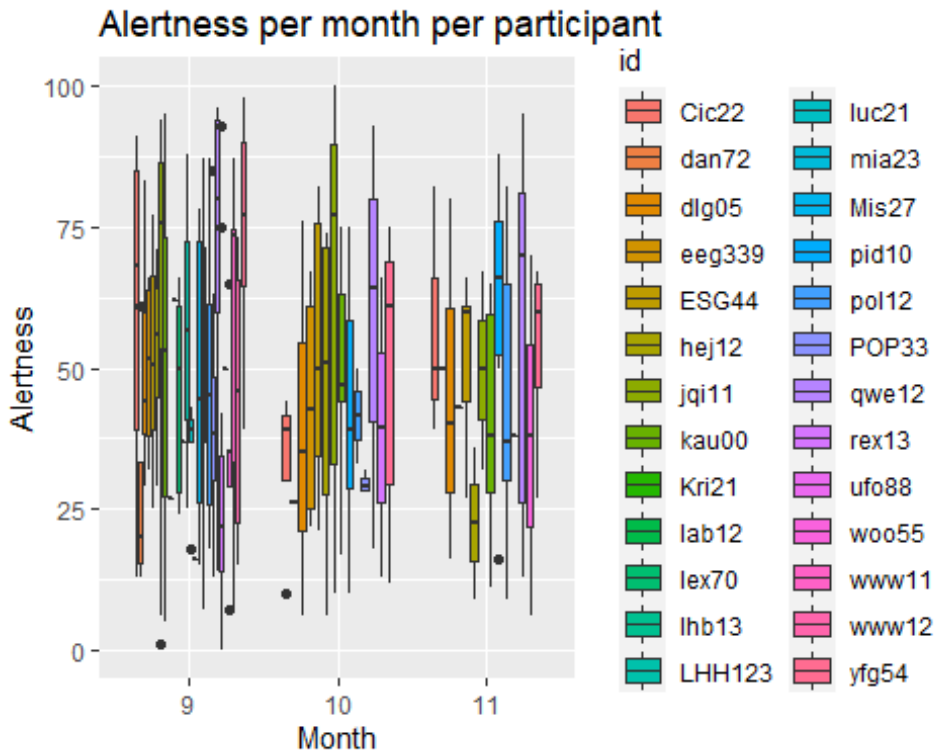
```
## Type III Analysis of Variance Table with Satterthwaite's method
##                Sum Sq Mean Sq NumDF DenDF F value Pr(>F)
## MentalActivity 2571.2  642.79     4 387.7  2.7043 0.0302 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
#Mood as function of activity
modelBFEmoodAct<-lmer(mood~Activity + (1|id),data=dataBFE)
anova(modelBFEmoodAct)
```

```
## Type III Analysis of Variance Table with Satterthwaite's method
##          Sum Sq Mean Sq NumDF DenDF F value  Pr(>F)
## Activity 8962.1  426.76    21 362.3  1.8355 0.01444 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

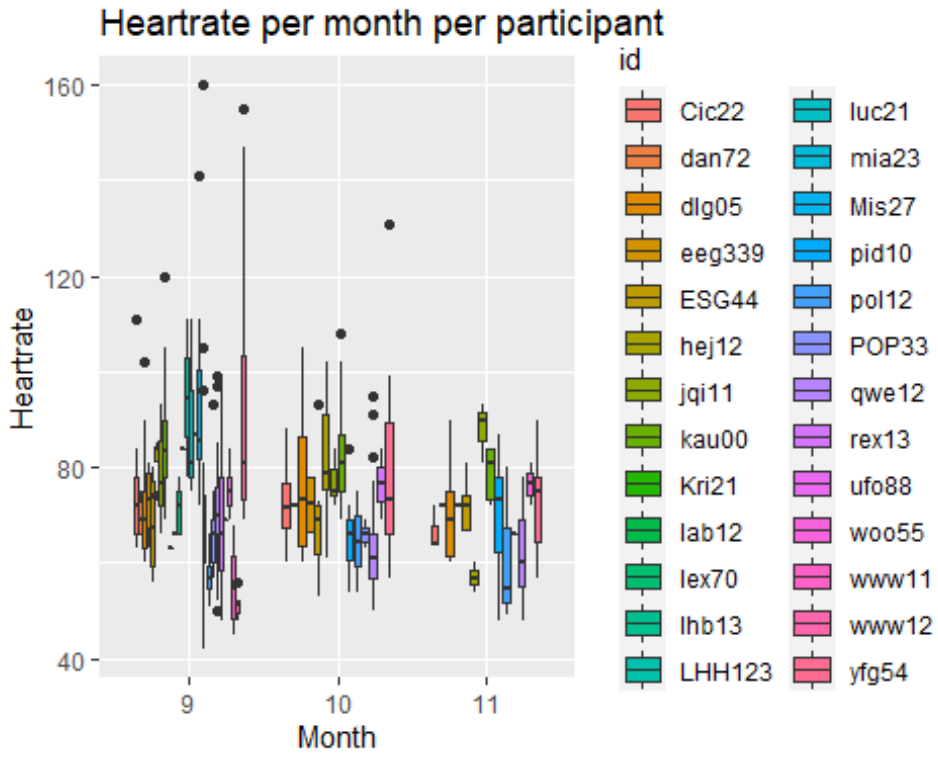
alertness

```r
# alertness boxplot
ggplot(dataBFE, aes(as.factor(month), fresh, fill = id)) +
  geom_boxplot() + labs(x = "Month", y = "Alertness")+
  ggtitle("Alertness per month per participant")
```

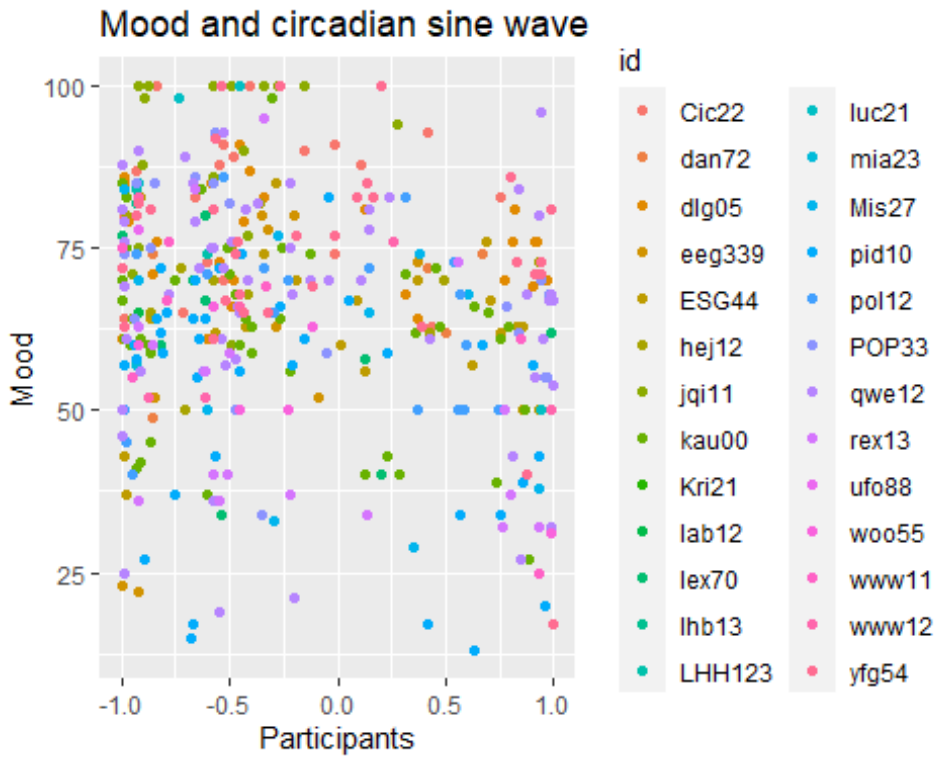


heartrate

```
# heartrate boxplot
ggplot(dataBFE, aes(as.factor(month), HeartRate, fill = id)) +
  geom_boxplot() + labs(x = "Month", y = "Heartrate")+
  ggtitle("Heartrate per month per participant")
```



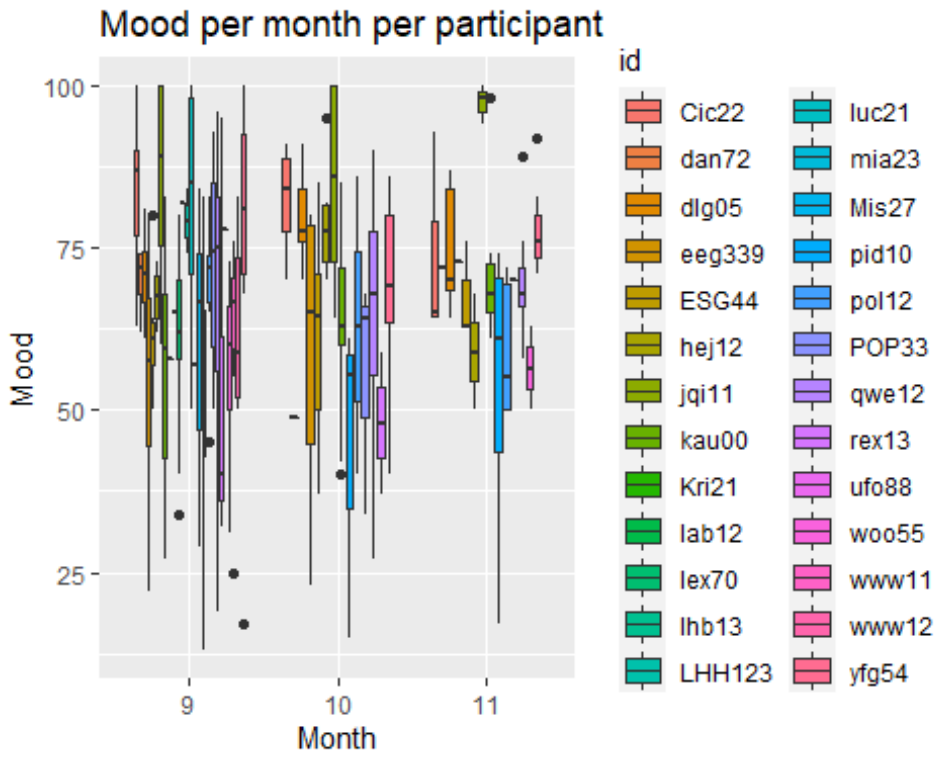

```
# mood scatterplot circadian sine
ggplot(dataBFE, aes(sinCirc, mood, colour = id)) +
  # geom_smooth(method = "loess") +
  geom_point() + labs(x = "Participants", y = "Mood") +
  ggtitle("Mood and circadian sine wave")
```



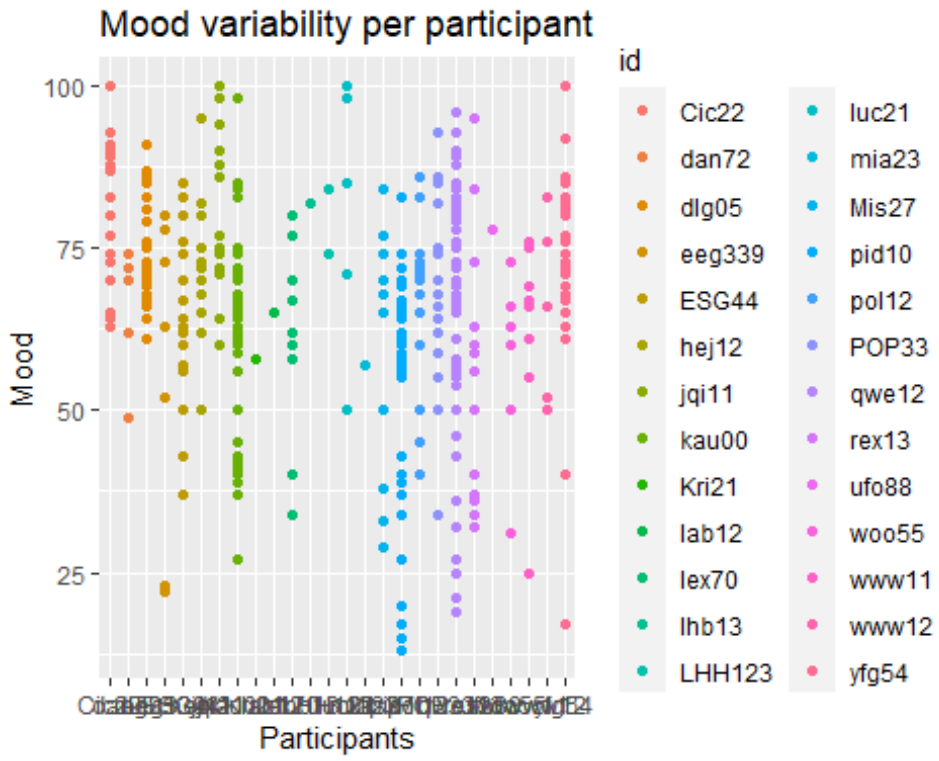

```r
# mood by activity
ggplot(dataBFE, aes(mood, Activity, colour = Activity)) +
  # geom_point()+
  geom_boxplot()+
  ggtitle("Activity and mood")+xlab('Mood')
```

Mood

```
# mood boxplot
ggplot(dataBFE, aes(as.factor(month), mood, fill = id)) +
  geom_boxplot() + labs(x = "Month", y = "Mood") +
  ggtitle("Mood per month per participant")
```

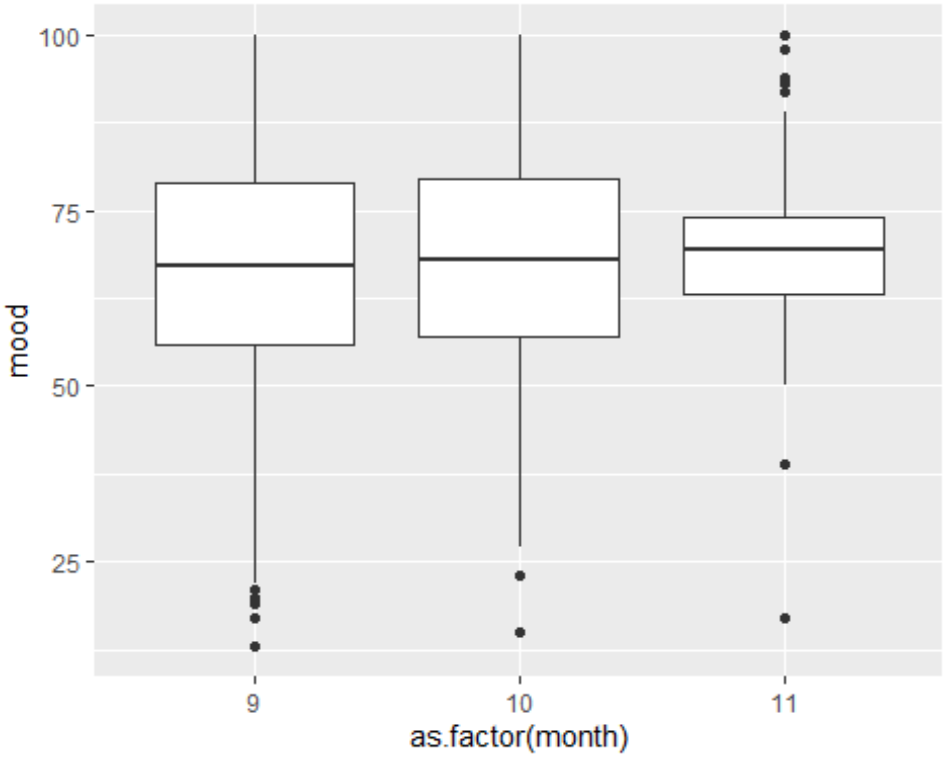



```r
# mood scatterplot
ggplot(dataBFE, aes(as.factor(id), mood, colour = id)) +
  geom_point() + labs(x = "Participants", y = "Mood") +
  ggtitle("Mood variability per participant")
```



Mood by month

```
library(ggplot2)
ggplot(dataBFE,aes(y=mood, x=as.factor(month)))+geom_boxplot()
```


100
75
50
25
mood
9
10
11
as.factor(month)