Circadian Mood

2022-11-21

### Install missing packages

install.packages('pacman')

## Installing package into '/usr/local/lib/R/site-library'  
## (as 'lib' is unspecified)

pacman::p\_load(lmerTest, tidyverse, lme4, wesanderson, MuMIn)

#Get and clean BodyFeelingExp data

#Get FaceStroopData  
datadir<-"/work/285178/BodyFeelingExp\_data"  
  
#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

## Sigrid’s preprocessing

# getting info about dataset  
# number of participants  
length(unique(dataBFE$id)) # 35 unique entries

## [1] 35

# for loop to figure out how many times each participant participated (suspecting something is off with id's)  
df\_final <- data.frame()  
  
for (i in unique(dataBFE$id)) {  
 name = dataBFE %>%filter(id == i) %>% select(id) %>% slice(1)  
 count = filter(dataBFE, id == i) %>% count()  
   
 df\_temp = data\_frame(name, count)  
 df\_final <- rbind(df\_final, df\_temp)  
}

## Warning: `data\_frame()` was deprecated in tibble 1.1.0.  
## Please use `tibble()` instead.  
## This warning is displayed once every 8 hours.  
## Call `lifecycle::last\_lifecycle\_warnings()` to see where this warning was generated.

# some of the ids are definitely made by mistake

# fixing participants who most likely used several IDs + lower case upper case and space issues  
  
dataBFE <- dataBFE %>%   
 mutate(  
 id = tolower(id),  
 id = gsub(" ", "", id),  
 id = gsub("dig05", "dlg05", id),  
 id = gsub("hej123", "hej12", id),  
 id = gsub("www111", "www11", id),  
 id = gsub("www123", "www11", id),  
 id = gsub("www12", "www11", id)  
 )  
  
# NB! rerun for loop in above chunk to make an updated boxplot below

# number of particants  
length(unique(dataBFE$id)) # 35 unique entries before

## [1] 25

# now we have 25 participants  
  
# mean number of reports   
mean(df\_final$n) # mean 11.31. # mean now; 15.84

## [1] 11.31429

sd(df\_final$n) # sd is 14.71

## [1] 13.90466

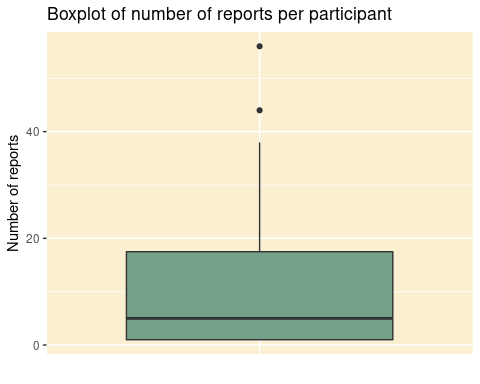
min(df\_final$n) # min 1 # min is still 1 ofc

## [1] 1

max(df\_final$n) # max is 56 # max now 57

## [1] 56

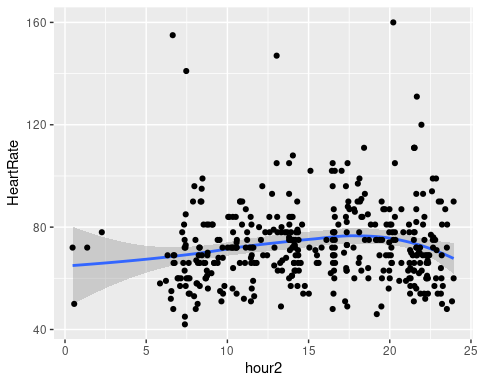
# boxplot   
df\_final %>% mutate(  
 class = "participant") %>%   
 ggplot() +  
 aes(class, n) +   
 geom\_boxplot(fill = wes\_palette("Royal2")[5]) +   
 labs(  
 title = "Boxplot of number of reports per participant",  
 x = "",   
 y = "Number of reports") +   
 theme(  
 panel.background= element\_rect(fill = wes\_palette("Royal1")[3]),   
 axis.text.x = element\_blank(),  
 axis.ticks.x = element\_blank()  
 )



## Some plotting

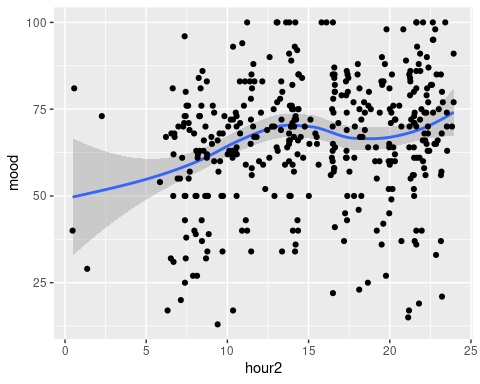
library(ggplot2)  
ggplot(dataBFE,aes(x=hour2,y=HeartRate))+geom\_smooth()+geom\_point()

## `geom\_smooth()` using method = 'loess' and formula 'y ~ x'



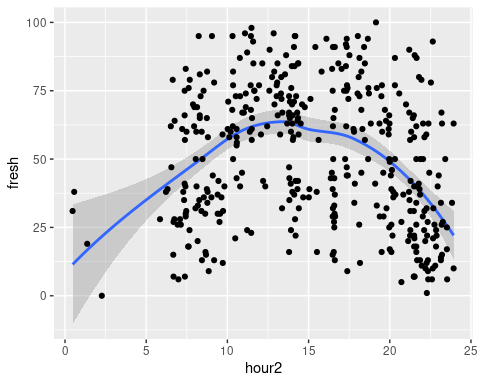
ggplot(dataBFE,aes(x=hour2,y=mood))+geom\_smooth()+geom\_point()

## `geom\_smooth()` using method = 'loess' and formula 'y ~ x'



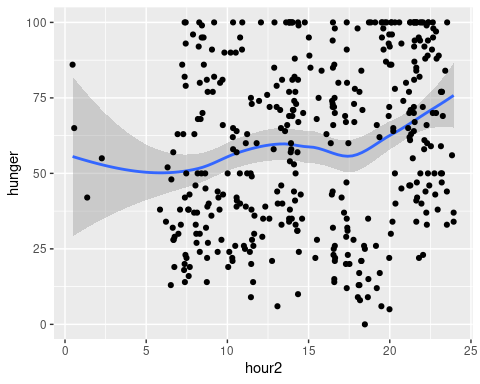
ggplot(dataBFE,aes(x=hour2,y=fresh))+geom\_smooth()+geom\_point()

## `geom\_smooth()` using method = 'loess' and formula 'y ~ x'



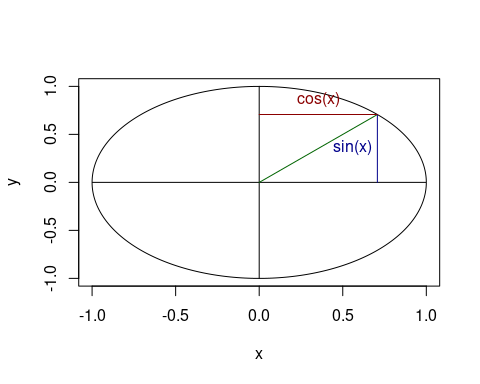
ggplot(dataBFE,aes(x=hour2,y=hunger))+geom\_smooth()+geom\_point()

## `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')

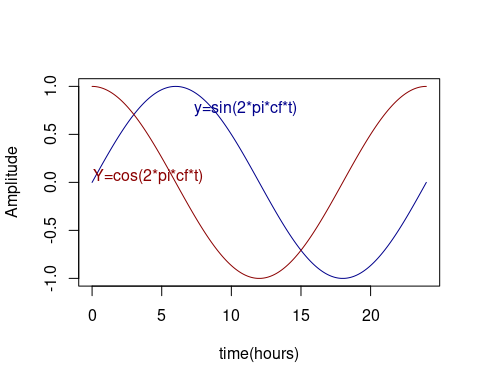


### Going beyond the circle, the sine and cosine functions can describe cycles in time

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

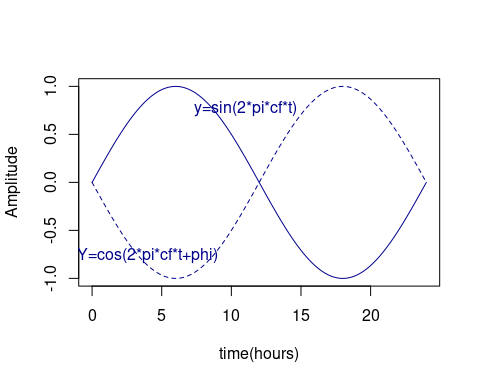
#### Plotting circadian sine and cosine waves

#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.

#### 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=cos(2\*pi\*cf\*t+phi)',col='darkblue')

 Using the trigonometric identity

we can rewrite the sine function (including phase) as

where

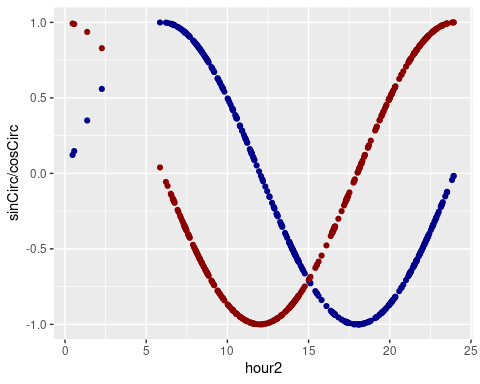
We can use the rewritten sine function in a linear regression analysis, where we estimate the best fitting B1 and B2. This will yield a composite estimate of the amplitude and the phase of the data.

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

The phase (phi) will be given by

#### 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('sinCirc/cosCirc')



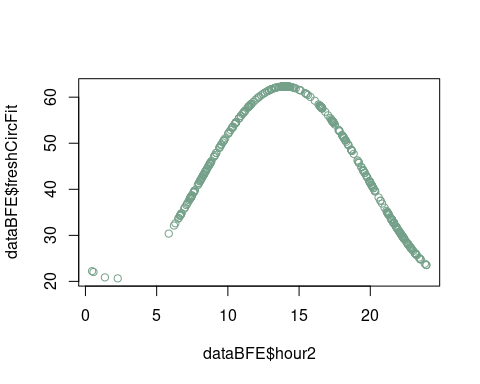
## Fitting 24 hour Oscillation model

#### Freshness

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

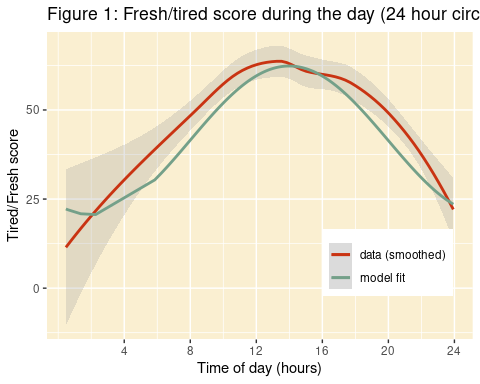
## Linear mixed model fit by REML. t-tests use Satterthwaite's method [  
## lmerModLmerTest]  
## Formula: fresh ~ sinCirc + cosCirc + (1 | id)  
## Data: dataBFE  
##   
## REML criterion at convergence: 3529.2  
##   
## Scaled residuals:   
## Min 1Q Median 3Q Max   
## -2.4949 -0.6996 0.0657 0.6999 3.6838   
##   
## Random effects:  
## Groups Name Variance Std.Dev.  
## id (Intercept) 89.52 9.462   
## Residual 414.33 20.355   
## Number of obs: 396, groups: id, 25  
##   
## Fixed effects:  
## Estimate Std. Error df t value Pr(>|t|)   
## (Intercept) 41.464 2.469 22.766 16.80 2.56e-14 \*\*\*  
## sinCirc -10.383 1.696 381.725 -6.12 2.32e-09 \*\*\*  
## cosCirc -18.102 1.585 380.438 -11.42 < 2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Correlation of Fixed Effects:  
## (Intr) sinCrc  
## sinCirc 0.229   
## cosCirc 0.176 0.345

dataBFE$freshCircFit<-m\_temp$coefficients[1,1]+m\_temp$coefficients[2,1]\*dataBFE$sinCirc+m\_temp$coefficients[3,1]\*dataBFE$cosCirc  
  
plot(x=dataBFE$hour2,y=dataBFE$freshCircFit,type='p',col=wes\_palette("Royal2")[5])



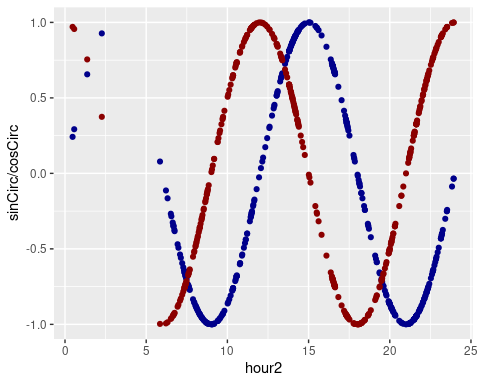
# testing colours for plotting  
#wes\_palette("Royal2")  
#wes\_palette("Royal1") # [2] er rød  
#wes\_palette("Royal2") # [5] er grøn  
  
# plot 24 hour rhythm  
dataBFE %>%   
 ggplot() +  
 aes(hour2, fresh) +  
 geom\_smooth(  
 aes(col = "data (smoothed)"), alpha = 0.25) +  
 geom\_line(aes(  
 hour2, freshCircFit, col = "model fit"), size = 1) +  
 labs(  
 title = "Figure 1: Fresh/tired score during the day (24 hour circadian)",  
 x='Time of day (hours)',   
 y='Tired/Fresh score') +   
 scale\_x\_continuous(breaks=c(4,8,12,16,20,24)) + # changing the x-axis values   
 scale\_colour\_manual("",  
 values = c("data (smoothed)" = wes\_palette("Royal1")[2],  
 "model fit" = wes\_palette("Royal2")[5]  
 )) +  
 theme(  
 panel.background= element\_rect(fill = wes\_palette("Royal1")[3]),   
 legend.title = element\_blank(),   
 legend.position = c(.80, .25)  
 )

## `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')



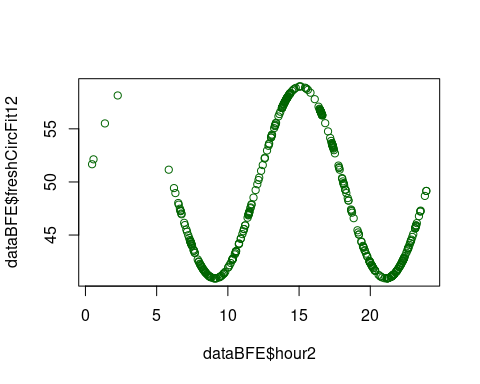
## Fitting 12 hour Oscillation models

#### Freshness

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

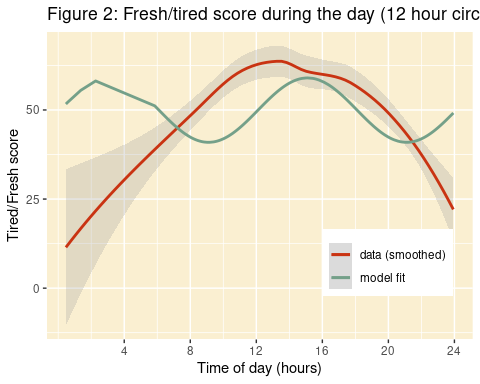
## Linear mixed model fit by REML. t-tests use Satterthwaite's method [  
## lmerModLmerTest]  
## Formula: fresh ~ sinCirc12 + cosCirc12 + (1 | id)  
## Data: dataBFE  
##   
## REML criterion at convergence: 3616.6  
##   
## Scaled residuals:   
## Min 1Q Median 3Q Max   
## -2.23217 -0.78346 0.02882 0.76421 2.63982   
##   
## Random effects:  
## Groups Name Variance Std.Dev.  
## id (Intercept) 64.59 8.037   
## Residual 527.64 22.970   
## Number of obs: 396, groups: id, 25  
##   
## Fixed effects:  
## Estimate Std. Error df t value Pr(>|t|)   
## (Intercept) 49.9564 2.2439 22.2551 22.263 < 2e-16 \*\*\*  
## sinCirc12 9.0226 1.6990 385.1641 5.311 1.85e-07 \*\*\*  
## cosCirc12 -0.4889 1.7243 382.8468 -0.284 0.777   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Correlation of Fixed Effects:  
## (Intr) snCr12  
## sinCirc12 0.167   
## cosCirc12 -0.017 -0.023

dataBFE$freshCircFit12<-m\_temp$coefficients[1,1]+m\_temp$coefficients[2,1]\*dataBFE$sinCirc12+m\_temp$coefficients[3,1]\*dataBFE$cosCirc12  
  
plot(x=dataBFE$hour2,y=dataBFE$freshCircFit12,type='p',col='darkgreen')



# plot 12 hour rhythm  
dataBFE %>%   
 ggplot() +  
 aes(hour2, fresh) +  
 geom\_smooth(  
 aes(col = "data (smoothed)"), alpha = 0.25) +  
 geom\_line(aes(  
 hour2, freshCircFit12, col = "model fit"), size = 1) +  
 labs(  
 title = "Figure 2: Fresh/tired score during the day (12 hour circadian)",  
 x='Time of day (hours)',   
 y='Tired/Fresh score') +   
 scale\_x\_continuous(breaks=c(4,8,12,16,20,24)) + # changing the x-axis values   
 scale\_colour\_manual("",  
 values = c("data (smoothed)" = wes\_palette("Royal1")[2],  
 "model fit" = wes\_palette("Royal2")[5]  
 )) +  
 theme(  
 panel.background= element\_rect(fill = wes\_palette("Royal1")[3]),   
 legend.title = element\_blank(),   
 legend.position = c(.80, .25)  
 )

## `geom\_smooth()` using method = 'loess' and formula 'y ~ x'



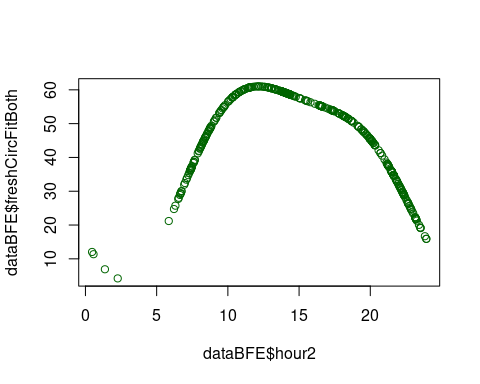
## Combining 12 and 24 hours

#### Freshness

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

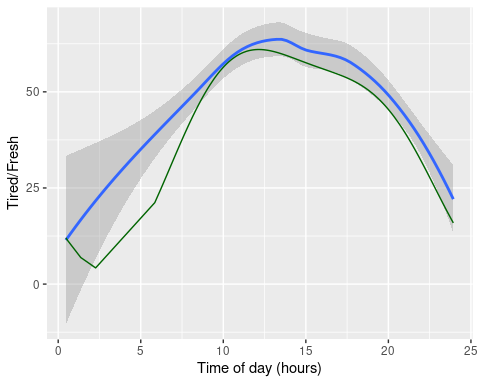
## Linear mixed model fit by REML. t-tests use Satterthwaite's method [  
## lmerModLmerTest]  
## Formula: fresh ~ sinCirc + cosCirc + sinCirc12 + cosCirc12 + (1 | id)  
## Data: dataBFE  
##   
## REML criterion at convergence: 3513.2  
##   
## Scaled residuals:   
## Min 1Q Median 3Q Max   
## -2.4678 -0.6970 0.0584 0.6880 3.7803   
##   
## Random effects:  
## Groups Name Variance Std.Dev.  
## id (Intercept) 91.85 9.584   
## Residual 405.38 20.134   
## Number of obs: 396, groups: id, 25  
##   
## Fixed effects:  
## Estimate Std. Error df t value Pr(>|t|)   
## (Intercept) 37.8986 2.7212 34.2210 13.927 1.16e-15 \*\*\*  
## sinCirc -14.9788 2.2338 380.0559 -6.706 7.28e-11 \*\*\*  
## cosCirc -22.7959 2.1576 381.0855 -10.566 < 2e-16 \*\*\*  
## sinCirc12 -7.0362 2.2299 383.0482 -3.155 0.00173 \*\*   
## cosCirc12 0.2997 1.5339 378.4250 0.195 0.84520   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Correlation of Fixed Effects:  
## (Intr) sinCrc cosCrc snCr12  
## sinCirc 0.424   
## cosCirc 0.396 0.637   
## sinCirc12 0.408 0.654 0.687   
## cosCirc12 -0.058 -0.146 -0.051 -0.093

dataBFE$freshCircFitBoth<-m\_temp$coefficients[1,1]+m\_temp$coefficients[2,1]\*dataBFE$sinCirc+m\_temp$coefficients[3,1]\*dataBFE$cosCirc+m\_temp$coefficients[4,1]\*dataBFE$sinCirc12+m\_temp$coefficients[5,1]\*dataBFE$cosCirc12  
  
plot(x=dataBFE$hour2,y=dataBFE$freshCircFitBoth,type='p',col='darkgreen')



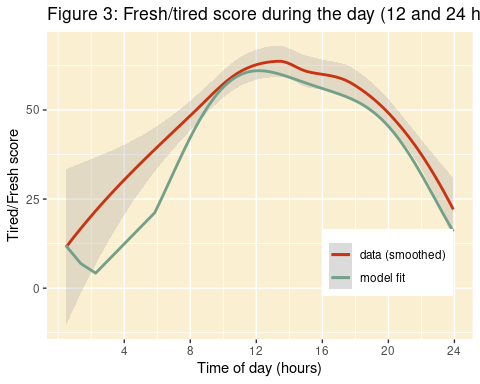
ggplot(dataBFE,aes(x=hour2,y=fresh))+geom\_smooth()+geom\_line(aes(x=hour2,y=freshCircFitBoth),col='darkgreen')+labs(x='Time of day (hours)', y='Tired/Fresh')

## `geom\_smooth()` using method = 'loess' and formula 'y ~ x'



# plot both 24h and 12h oscillation  
dataBFE %>%   
 ggplot() +  
 aes(hour2, fresh) +  
 geom\_smooth(  
 aes(col = "data (smoothed)"), alpha = 0.25) +  
 geom\_line(aes(  
 hour2, freshCircFitBoth, col = "model fit"), size = 1) +  
 labs(  
 title = "Figure 3: Fresh/tired score during the day (12 and 24 hour circadian)",  
 x='Time of day (hours)',   
 y='Tired/Fresh score') +   
 scale\_x\_continuous(breaks=c(4,8,12,16,20,24)) + # changing the x-axis values   
 scale\_colour\_manual("",  
 values = c("data (smoothed)" = wes\_palette("Royal1")[2],  
 "model fit" = wes\_palette("Royal2")[5]  
 )) +  
 theme(  
 panel.background= element\_rect(fill = wes\_palette("Royal1")[3]),   
 legend.title = element\_blank(),   
 legend.position = c(.80, .25)  
 )

## `geom\_smooth()` using method = 'loess' and formula 'y ~ x'



# calculate R-squared  
library('MuMIn')  
  
r.squaredGLMM(modelBFEfreshCircBoth)

## Warning: 'r.squaredGLMM' now calculates a revised statistic. See the help page.

## R2m R2c  
## [1,] 0.2428835 0.3827357

# marginal Rsq = 0.2429  
# conditional Rsq = 0.3827

## 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 3635.8 3655.7 -1812.9 3625.8   
## modelBFEfreshCircBoth 7 3542.4 3570.2 -1764.2 3528.4 97.44 2 < 2.2e-16  
##   
## modelBFEfreshCirc12   
## modelBFEfreshCircBoth \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

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 3548.3 3568.2 -1769.2 3538.3   
## modelBFEfreshCircBoth 7 3542.4 3570.2 -1764.2 3528.4 9.9206 2 0.007011  
##   
## modelBFEfreshCirc   
## modelBFEfreshCircBoth \*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

# Getting more info about participants  
df\_final <- data.frame()  
  
for (i in unique(dataBFE$id)) {  
 name = dataBFE %>%filter(id == i) %>% select(id, Gender, age, Handedness) %>% slice(1)  
 count = filter(dataBFE, id == i) %>% count()  
   
 df\_temp = data\_frame(name, count)  
 df\_final <- rbind(df\_final, df\_temp)  
}  
  
# gender  
df\_final %>% filter(  
 Gender == "female"  
 ) %>% count() # 17 girls

## # A tibble: 1 × 1  
## n  
## <int>  
## 1 17

df\_final %>% filter(  
 Gender == "male"  
 ) %>% count() # 8 boys

## # A tibble: 1 × 1  
## n  
## <int>  
## 1 8

# age  
mean(df\_final$age) # mean age 24.44

## [1] 24.44

sd(df\_final$age) # sd age 5.38

## [1] 5.378042

min(df\_final$age) # 21

## [1] 21

max(df\_final$age) # 49

## [1] 49

# figuring out the start and end data of the experiment by manipulating the Time column  
  
#dataBFE$Time # commented out because the output is very long  
  
dataBFE <- dataBFE %>%   
 mutate(  
 new\_date = as.Date(Time, format = "%a %b %d %Y")  
 )  
# start data Sep 8  
# end data Nov 21   
# = 2 months and 13 days