Plot Weight and Blood-pressure

Setups

```
rm(list=ls())
library("readxl")
library(varhandle)
library(lubridate)
```

```
##
## Attaching package: 'lubridate'

## The following objects are masked from 'package:base':
##
##     date, intersect, setdiff, union
```

```
library(MASS)
library(gam)
```

```
## Loading required package: splines

## Loading required package: foreach

## Loaded gam 1.16.1
```

```
library(lomb)
library(extRemes)
```

```
## Loading required package: Lmoments

## Loading required package: distillery

##
## Attaching package: 'extRemes'

## The following objects are masked from 'package:stats':
##
##     qqnorm, qqplot
```

```
library(oce)
```

```
## Loading required package: gsw

## Loading required package: testthat
```

```
library(TideHarmonics)
```

Notes

May have to first convert the xls files to csv with e.g. “ssconvert pressure.xls pressure.csv”

get Weight data

```
weight <- read.csv( "~/WORKSHOP/BP/DATA/weight.csv",header=TRUE,)
weight <- unfactor(weight)
weight$Weight <- as.numeric(gsub("[a-z\\()]","",weight$Weight))
# POSIX
myPOSIX <- as.data.frame(within(weight, { myPOSIX=strptime(paste(Date,Time), "%Y/%m/%d%H:%M:%S") }))
df <- cbind(myPOSIX$myPOSIX,weight)
df <- df[,c("myPOSIX$myPOSIX","Weight")]
```

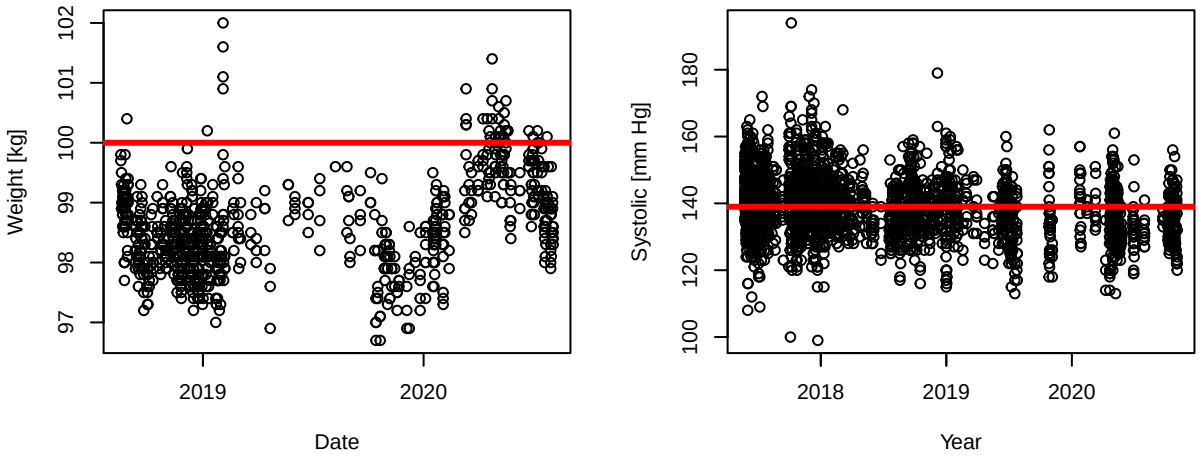
get BP Pressure data

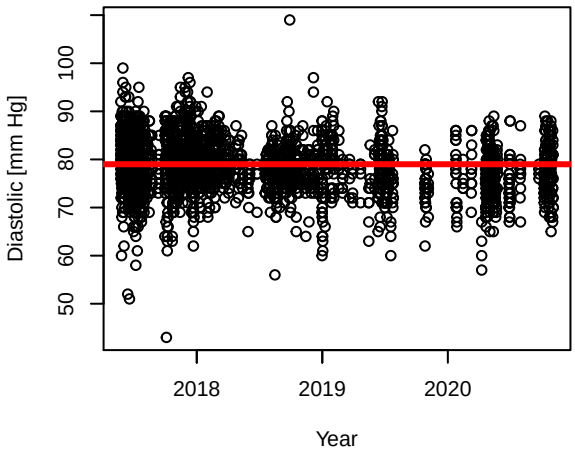
```
pressure <- read.csv( "~/WORKSHOP/BP/DATA/pressure.csv",sep="",header=FALSE)
pressure <- unfactor(pressure)
Date <- pressure[,1]
Time <- pressure[,2]

SYS <- pressure[,3]
DIA <- pressure[,4]
PULSE <- pressure[,5]
# POSIX
datetime <- paste(Date,Time)
myPOSIX <- as.POSIXct(datetime)

df2 <- cbind(myPOSIX,pressure[,c(-1,-2)])
colnames(df2) <- c("myPOSIX","SYS","DIA","Pulse")
```

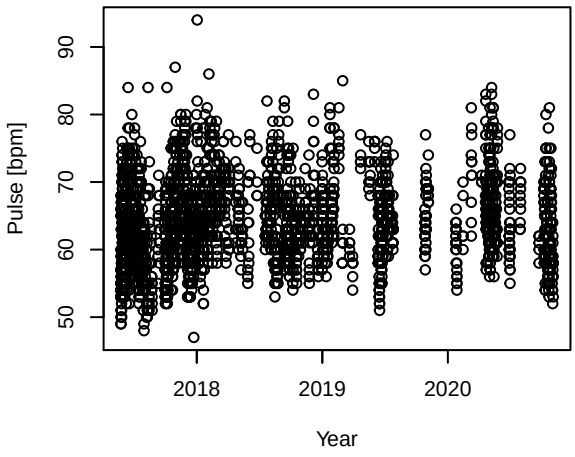
Plots

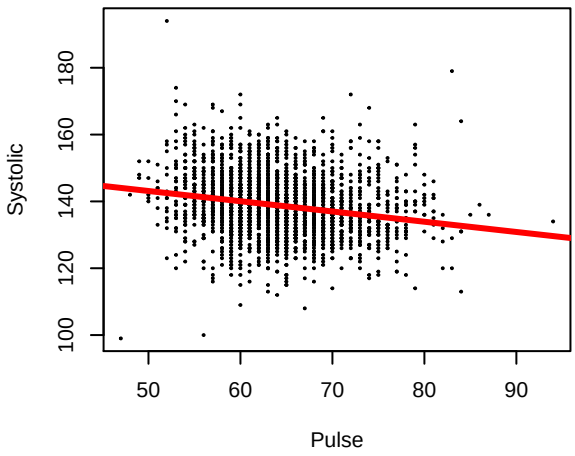
```
par(mfrow=c(3,2))
plot(df,xlab="Date",ylab="Weight [kg]")
abline(h=100,col=2,lwd=3)
plot(df2$myPOSIX,df2$SYS,xlab="Year",ylab="Systolic [mm Hg]")
abline(h=median(df2$SYS),col=2,lwd=3)
plot(df2$myPOSIX,df2$DIA,xlab="Year",ylab="Diastolic [mm Hg]")
abline(h=median(df2$DIA),col=2,lwd=3)
plot(df2$myPOSIX,df2$Pulse,xlab="Year",ylab="Pulse [bpm]")
```

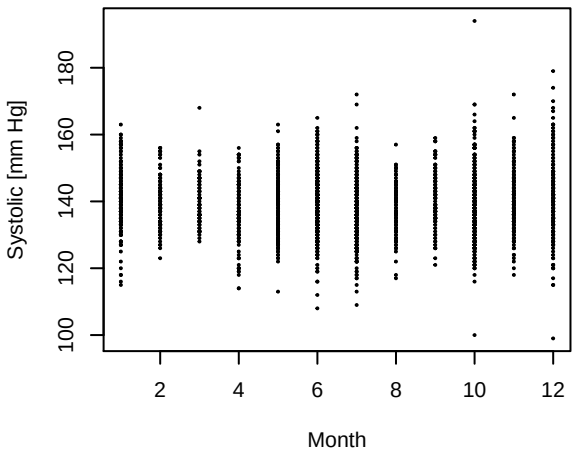
```
#
plot(df2$Pulse,df2$SYS,xlab="Pulse",ylab="Systolic",pch=19,cex=0.2)
rlmfit <- rlm(df2$SYS ~ df2$Pulse)
abline(rlmfit,col=2,lwd=3)
plot(month(df2$myPOSIX),df2$SYS,xlab="Month",ylab="Systolic [mm Hg]",pch=19,cex=0.2)
```



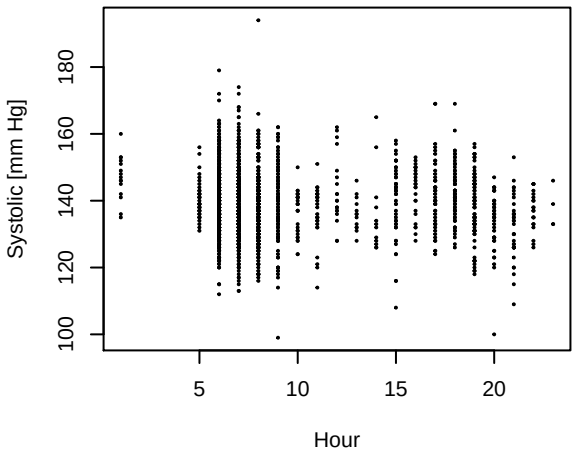









```
#
plot(hour(df2$myPOSIX),df2$SYS,xlab="Hour",ylab="Systolic [mm Hg]",pch=19,cex=0.2)
```



Model

```
rlm1 <- rlm(df2$SYS ~ df2$DIA+df2$Pulse )

summary(rlm1)
```

```
##
## Call: rlm(formula = df2$SYS ~ df2$DIA + df2$Pulse)
## Residuals:
##      Min       1Q   Median       3Q      Max
## -34.7959  -4.2432  -0.1576   4.2037  49.3191
##
## Coefficients:
##             Value    Std. Error t value
## (Intercept)  76.3898   2.3648    32.3031
## df2$DIA       1.0254   0.0238    43.0153
## df2$Pulse    -0.2840   0.0218   -13.0021
##
## Residual standard error: 6.248 on 2511 degrees of freedom
```

```
rlm2 <- rlm(df2$SYS ~ df2$DIA+df2$Pulse+hour(df2$myPOSIX) )

summary(rlm2)
```

```
##
## Call: rlm(formula = df2$SYS ~ df2$DIA + df2$Pulse + hour(df2$myPOSIX))
## Residuals:
##      Min       1Q   Median       3Q      Max
## -34.7353  -4.2450  -0.1379   4.2071  49.3616
##
## Coefficients:
##                    Value    Std. Error t value
## (Intercept)         75.8190   2.4734    30.6539
## df2$DIA              1.0280   0.0241    42.6939
## df2$Pulse           -0.2816   0.0220   -12.7718
```

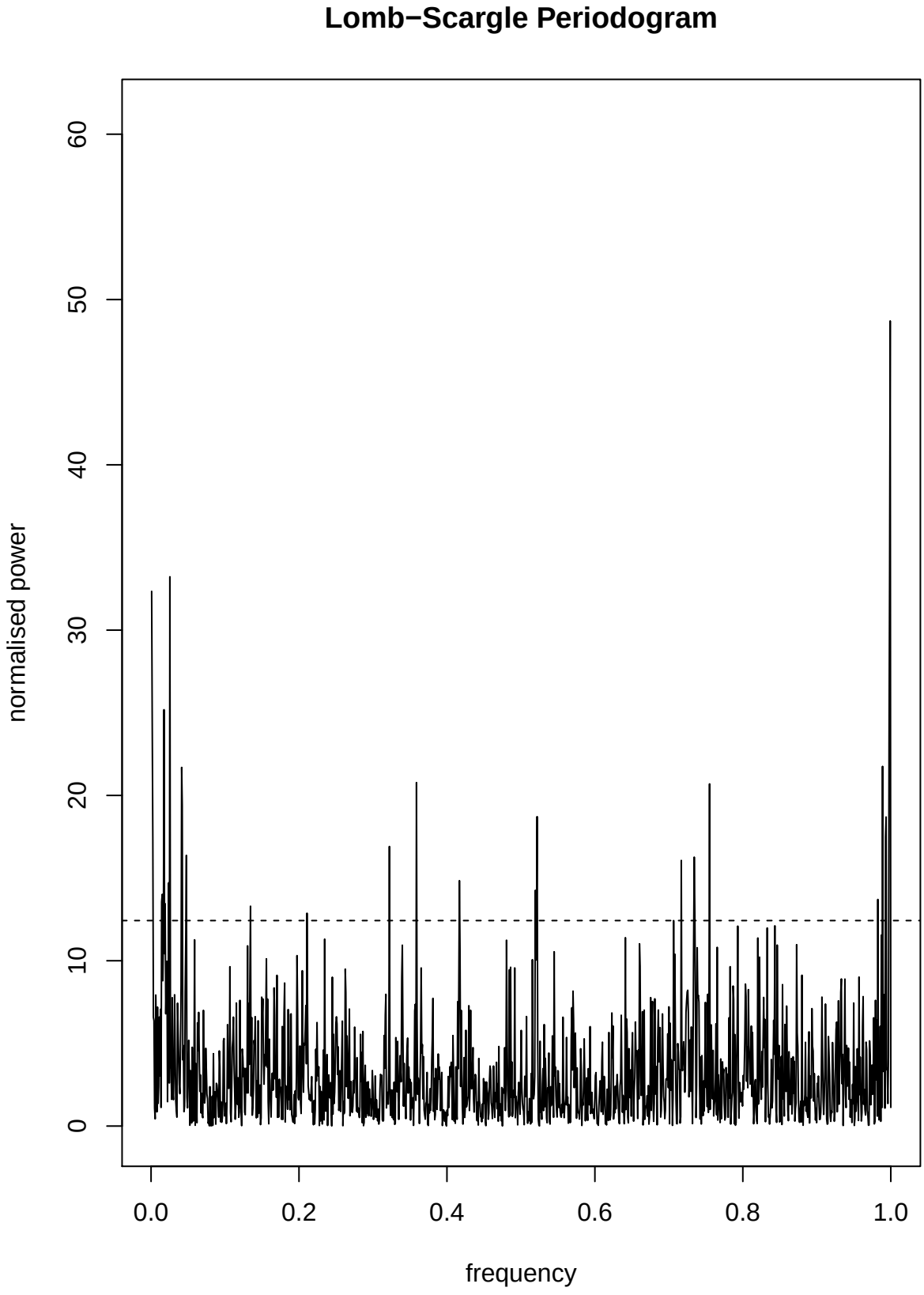
```
## hour(df2$myPOSIX)   0.0246   0.0313     0.7871
##
## Residual standard error: 6.253 on 2510 degrees of freedom
```

```
#
gam1 <- gam(df2$SYS ~ df2$DIA+df2$Pulse+hour(df2$myPOSIX) )
summary(gam1)
```

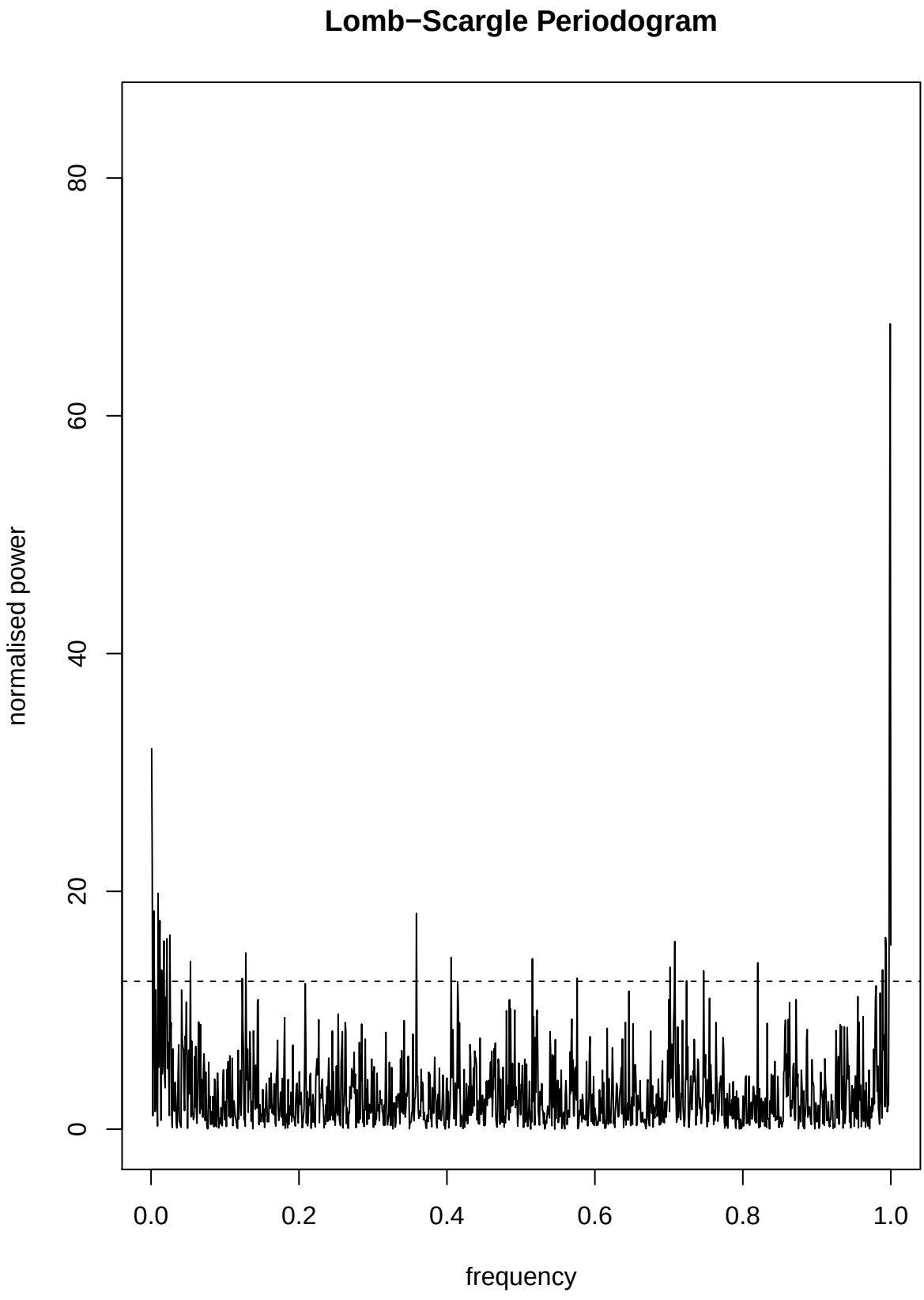
```
##
## Call: gam(formula = df2$SYS ~ df2$DIA + df2$Pulse + hour(df2$myPOSIX))
## Deviance Residuals:
##      Min       1Q   Median       3Q      Max
## -34.6335  -4.4626  -0.2861   4.0610  49.4327
##
## (Dispersion Parameter for gaussian family taken to be 47.4752)
##
##     Null Deviance: 210677.9 on 2513 degrees of freedom
## Residual Deviance: 119162.6 on 2510 degrees of freedom
## AIC: 16844.98
##
## Number of Local Scoring Iterations: 2
##
## Anova for Parametric Effects
##                     Df Sum Sq Mean Sq   F value Pr(>F)
## df2$DIA              1  84847   84847 1787.1871 <2e-16 ***
## df2$Pulse            1   6642    6642  139.9096 <2e-16 ***
## hour(df2$myPOSIX)    1     26      26    0.5479 0.4592
## Residuals         2510 119163      47
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Spectra

```
JD <- julianDay(df2$myPOSIX, year = year(df2$myPOSIX), month = month(df2$myPOSIX), day = day(df2$myPOSI
lsp(df2$SYS,times=JD)
```


Lomb−Scargle Periodogram
normalised power
0
10
20
30
40
50
60
0.0
0.2
0.4
0.6
0.8
1.0
frequency

```
lsp(df2$DIA,times=JD)
```



```
lsp(df2$Pulse,times=JD)
```