

Universidade Federal de Uberlândia

Listado Módulo 2

Levy Gabriel da Silva Galvão

Uberlândia
2021

Módulo 2

Exercício 1

Gere sinais sintéticos que simulem a forma de onda do ciclo respiratório, de acordo a faixa etária:

- do nascimento a 6 semanas: 30 a 40 respirações por minuto;
- 6 meses: 25 a 40 respirações por minuto;
- 3 anos: 20 a 30 respirações por minuto;
- 6 anos: 18 a 25 respirações por minuto;
- 10 anos: 17 a 23 respirações por minuto;
- Adultos: 12 a 18 respirações por minuto;
- Idosos (≥ 65 anos): 12 a 28 respirações por minuto;
- Idosos (≥ 80 anos): 10 a 30 respirações por minuto.

```
library(stringr)
genBreathing <- function(duration, sampling_time, verbose_age)
{
  weeks <- 0
  months <- 0
  years <- 0

  # separate verbose age into occurrences
  splitted_age <- unlist(str_extract_all(verbose_age, pattern="\b(\w+=\w+)\b"))
  for(itr in length(splitted_age))
  {
    # extract the name and number of each occurrence
    all <- unlist(str_extract_all(splitted_age[itr], pattern="\b(\w+)\b"))
    word <- all[1]
    number <- strtoi(all[2])
    # age in days is irrelevant
    if(word=='weeks'){weeks <- weeks+number}
    if(word=='months'){months <- months+number}
    if(word=='years'){years <- years+number}
  }
  while(weeks>=5) # 1 month = 5 weeks
  {
    weeks <- weeks-5
    months <- months+1
  }
  while(months>=12) # 1 year = 12 months
  {
    months <- months-12
    years <- years+1
  }

  # define frequency based in age (max of range given)
  if((years==0 && months==1 && weeks<=1) || (years==0 && months==0 && weeks<=6)){freq <- 40}
  # cycles/min
  else if(years==0 && months<=6) {freq <- (25+40)/2} # cycles/min
  else if(years<=3) {freq <- 30} # cycles/min
```

```

else if(years<=6) {freq <- 25} # cycles/min
else if(years<=10) {freq <- 23} # cycles/min
else if(years<65) {freq <- 18} # cycles/min
else if(years<80) {freq <- 28} # cycles/min
else {freq <- 30} # Hz

f <- freq/60 # convert freq to Hz
t <- seq(from=0, to=duration-sampling_time, by=sampling_time)
signal <- sin(2*pi*f*t)

return(signal)
}

```

A função `genBreathing` gera um sinal do ciclo respiratório de acordo com a especificação anterior. Seus parâmetros são a duração do sinal em segundos e o tempo de amostragem. O terceiro parâmetro é uma string que identifica a idade de acordo com alguns exemplos abaixo:

- Idade de 4 semanas implica em `verbose_age = 'weeks=4'`;
- Idade de 2 meses e 3 semanas implica em `verbose_age = 'months=2, weeks=3'`
- Idade de 42 anos implica em `verbose_age = 'years=42'`

Esse parâmetro é acumulativo, então qualquer sequência de idades composta por semanas, meses e anos podem ser inseridas (obs: 5 semanas são consideradas como 1 mês e 12 meses são considerados como 1 ano). Abaixo são inicializados alguns sinais para cada uma das categorias de idade especificadas.

```

duration = 60 # 1 min
sampling_time = 0.001 # 1 ms
t <- seq(from=0, to=duration-sampling_time, by=sampling_time)
array.cat <- vector("list", 8)
array.title <- vector("list", 8)
array.cat[[1]] <- genBreathing(duration, sampling_time, verbose_age='weeks=6')
array.cat[[2]] <- genBreathing(duration, sampling_time, verbose_age='months=6')
array.cat[[3]] <- genBreathing(duration, sampling_time, verbose_age='years=3')
array.cat[[4]] <- genBreathing(duration, sampling_time, verbose_age='years=6')
array.cat[[5]] <- genBreathing(duration, sampling_time, verbose_age='years=10')
array.cat[[6]] <- genBreathing(duration, sampling_time, verbose_age='years=42')
array.cat[[7]] <- genBreathing(duration, sampling_time, verbose_age='years=70')
array.cat[[8]] <- genBreathing(duration, sampling_time, verbose_age='years=90')

array.title[[1]] <- 'Ciclo respiratório: 6 semanas de idade'
array.title[[2]] <- 'Ciclo respiratório: 6 meses de idade'
array.title[[3]] <- 'Ciclo respiratório: 3 anos de idade'
array.title[[4]] <- 'Ciclo respiratório: 6 anos de idade'
array.title[[5]] <- 'Ciclo respiratório: 10 anos de idade'
array.title[[6]] <- 'Ciclo respiratório: 42 anos de idade'
array.title[[7]] <- 'Ciclo respiratório: 70 anos de idade'
array.title[[8]] <- 'Ciclo respiratório: 90 anos de idade'

```

Exercício 2

Utilize a biblioteca **dygraph** para plotar um gráfico para cada um dos sinais simulados na questão 1. Adicione legendas para os eixos x e y de cada gráfico.

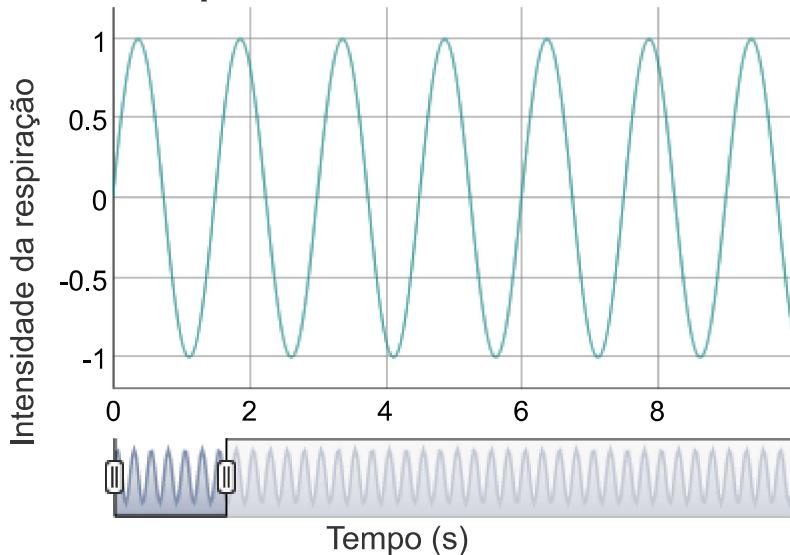
```

library(dygraphs)
library(htmltools)
timeWindow <- c(0,10) # 0 to 10s

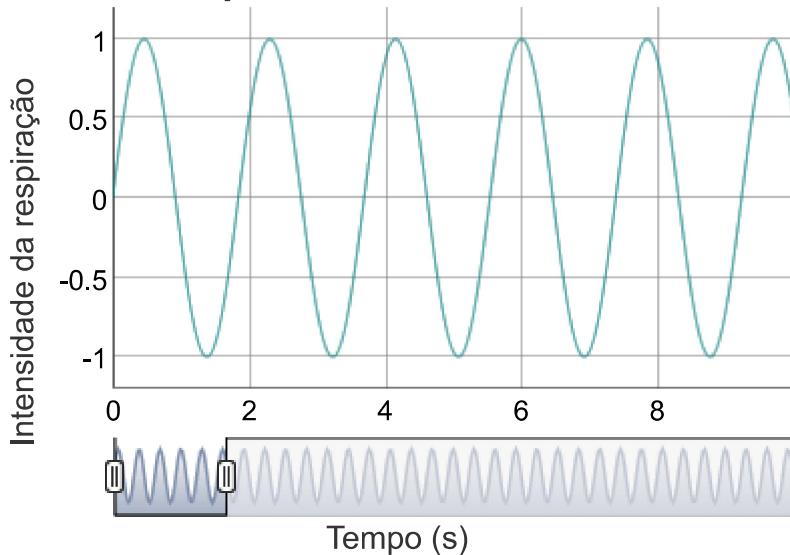
usingDygraph = function(i){
  plotResult <- data.frame(time=t, array.cat[i]) %>%
    dygraph(group='breathing', xlab='Tempo (s)', ylab='Intensidade da respiração', main=arr
y.title[i],
           width=400, height=300) %>%
    dyRangeSelector(dateWindow=timeWindow)
  htmltools::tags$div(plotResult, style = "padding:10px; width: 450px; display:inline-block;")
}
res <- lapply(1:8, usingDygraph)
htmltools::tagList(res)

```

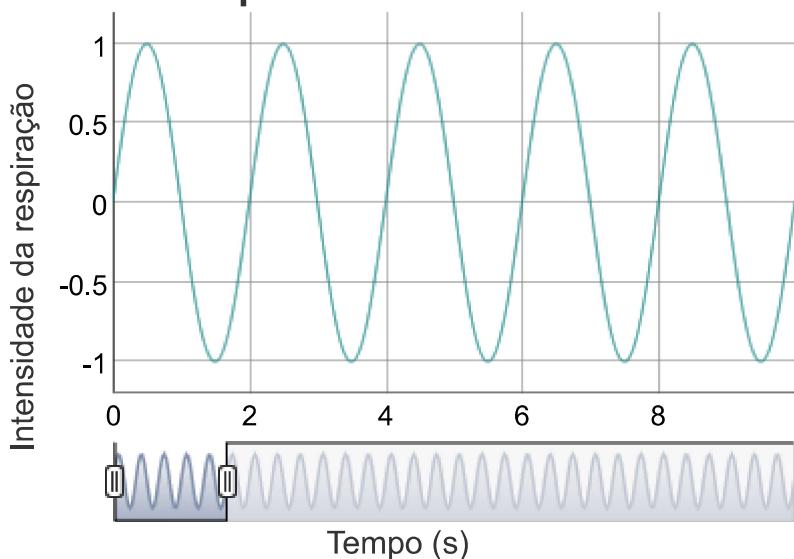
Ciclo respiratório: 6 semanas de idade



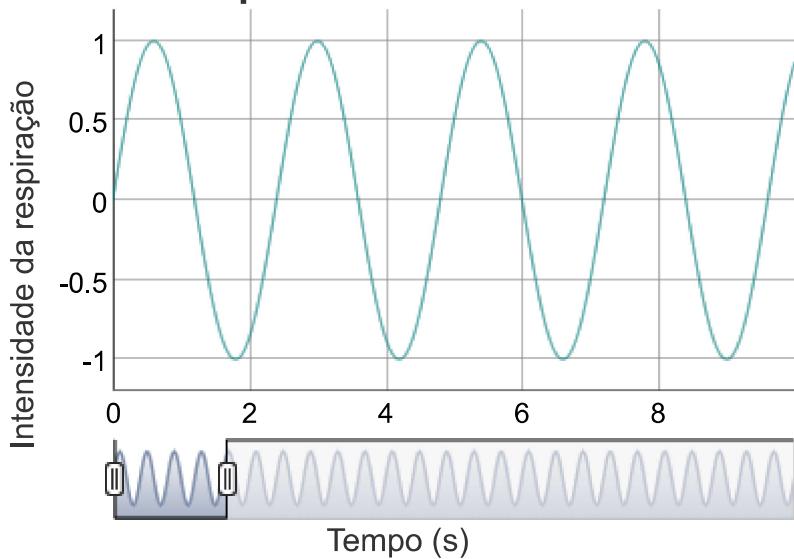
Ciclo respiratório: 6 meses de idade



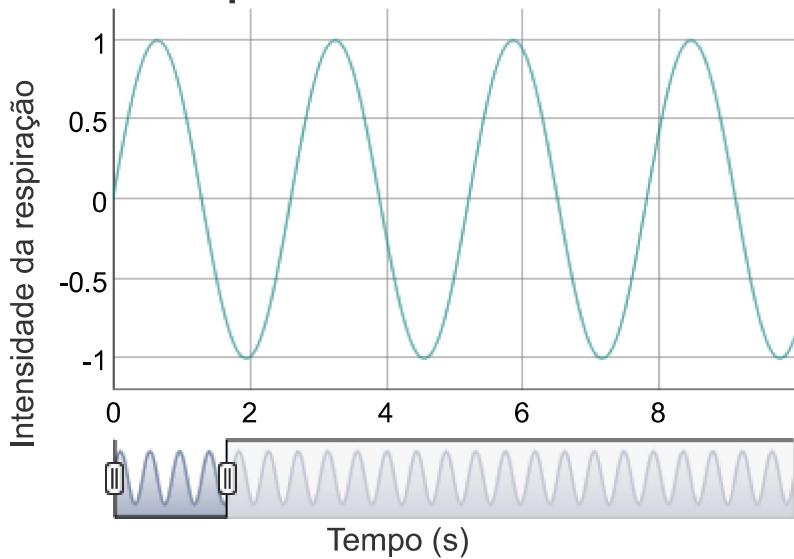
Ciclo respiratório: 3 anos de idade



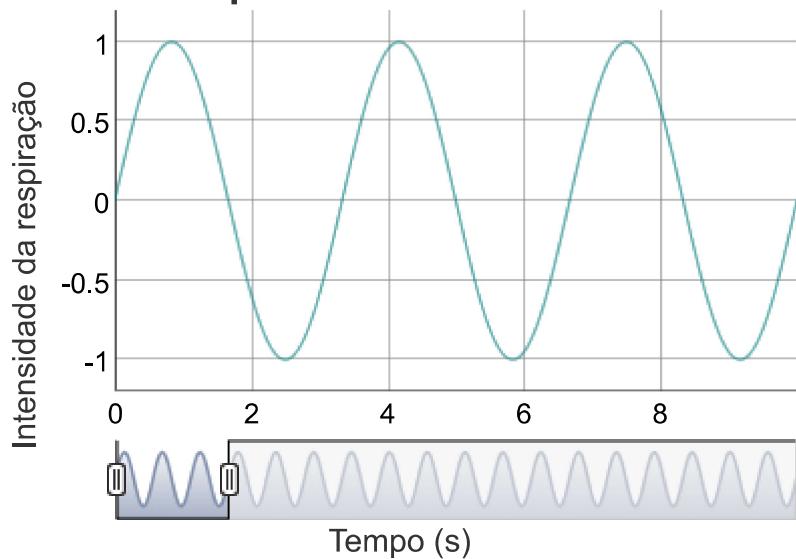
Ciclo respiratório: 6 anos de idade



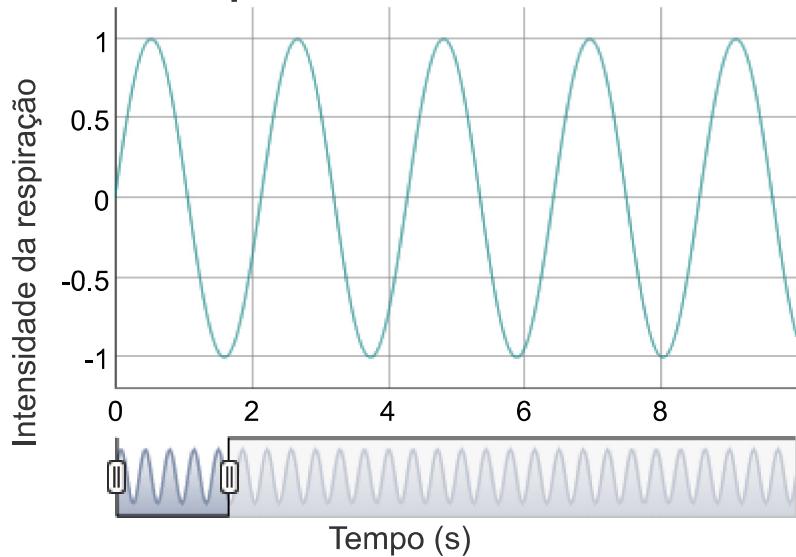
Ciclo respiratório: 10 anos de idade



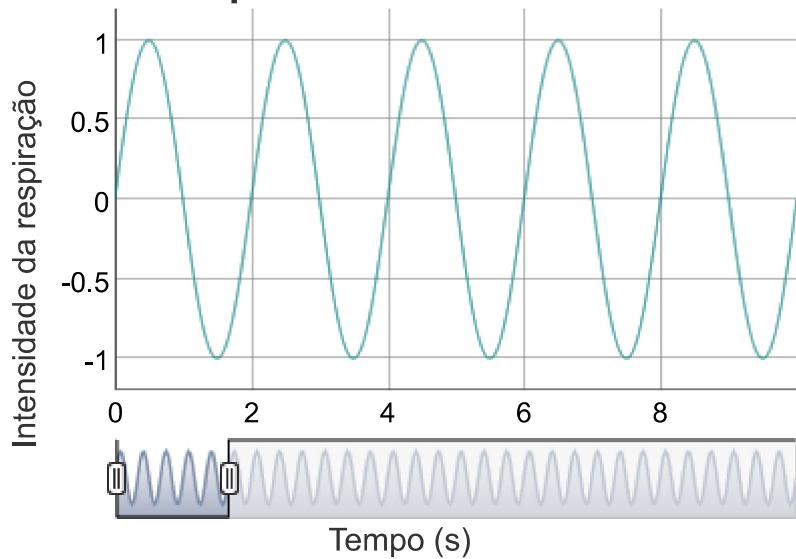
Ciclo respiratório: 42 anos de idade



Ciclo respiratório: 70 anos de idade



Ciclo respiratório: 90 anos de idade

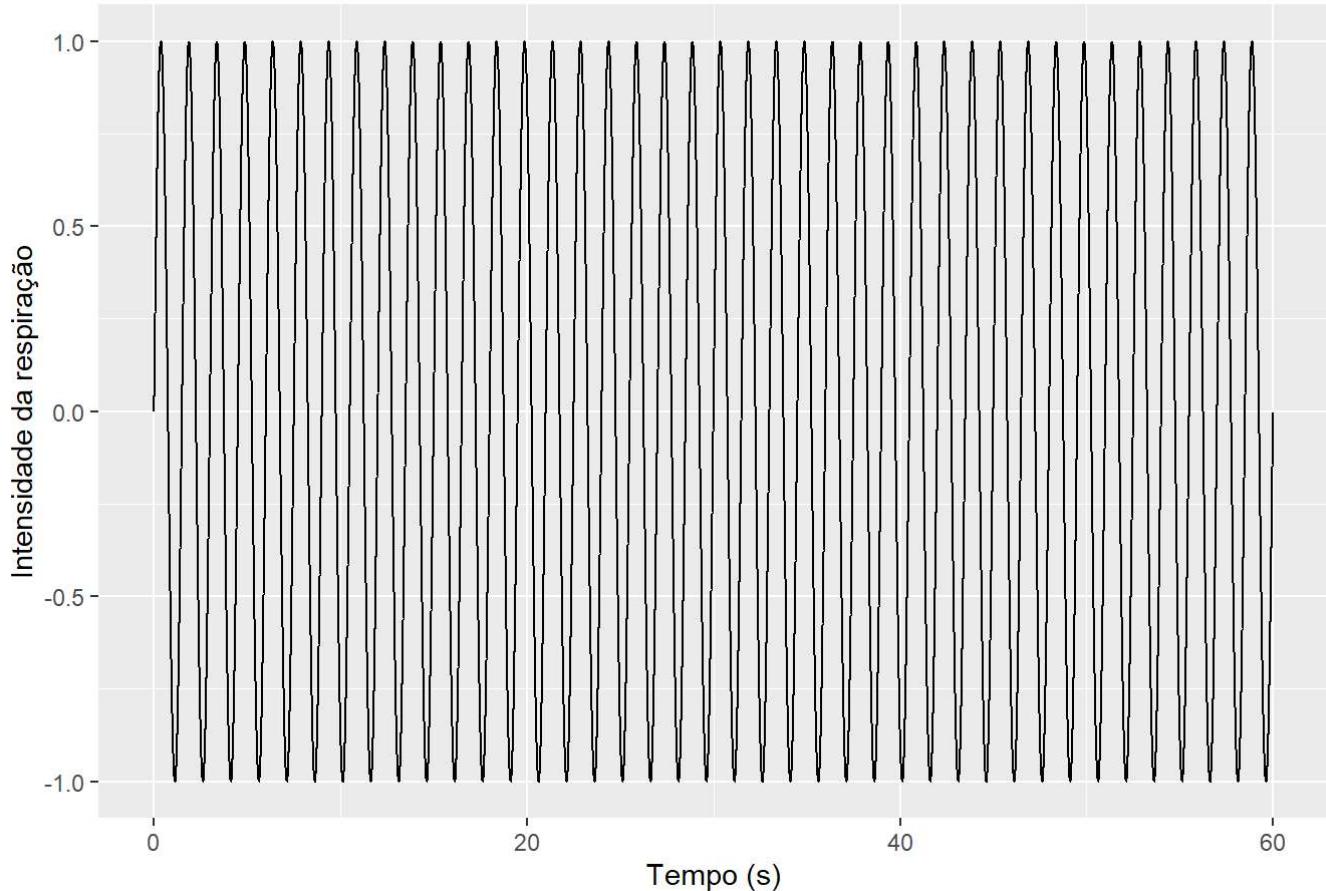


Exercício 3

Utilize a função **ggplot** para plotar um gráfico para cada um dos sinais simulados na questão 1. Adicione legendas para os eixos x e y de cada gráfico.

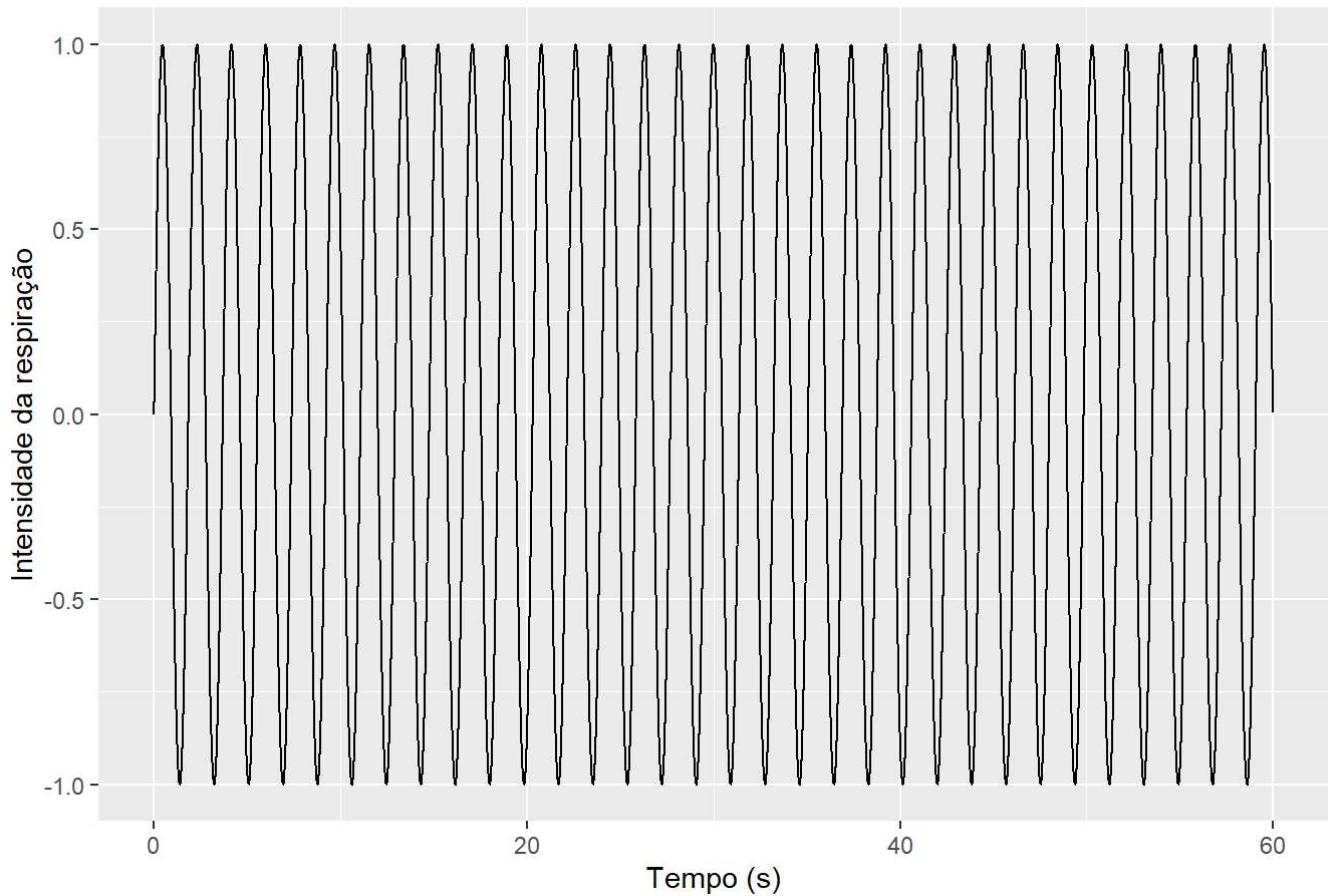
```
library(ggplot2)
library(magrittr)
library(multipanelfigure)
usingGgplot = function(i){
  df <- data.frame(time=t, breath=array.cat[[i]])
  plotResult <- ggplot(data=df, aes(x=time, y=breath)) +
    labs(x='Tempo (s)', y='Intensidade da respiração') +
    ggtitle(array.title[i]) +
    geom_line()
  return(plotResult)
}
usingGgplot(1)
```

Ciclo respiratório: 6 semanas de idade



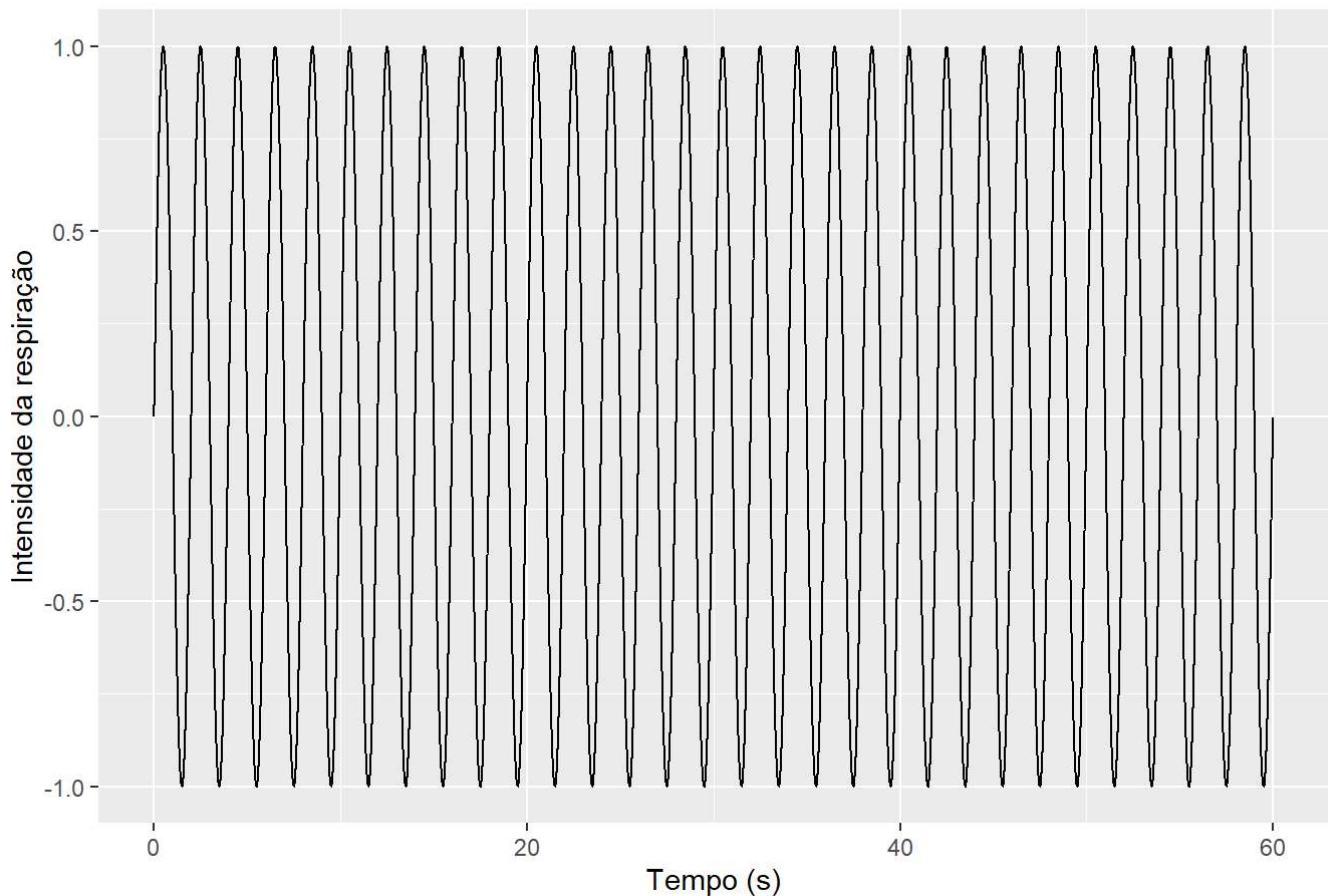
```
usingGgplot(2)
```

Ciclo respiratório: 6 meses de idade



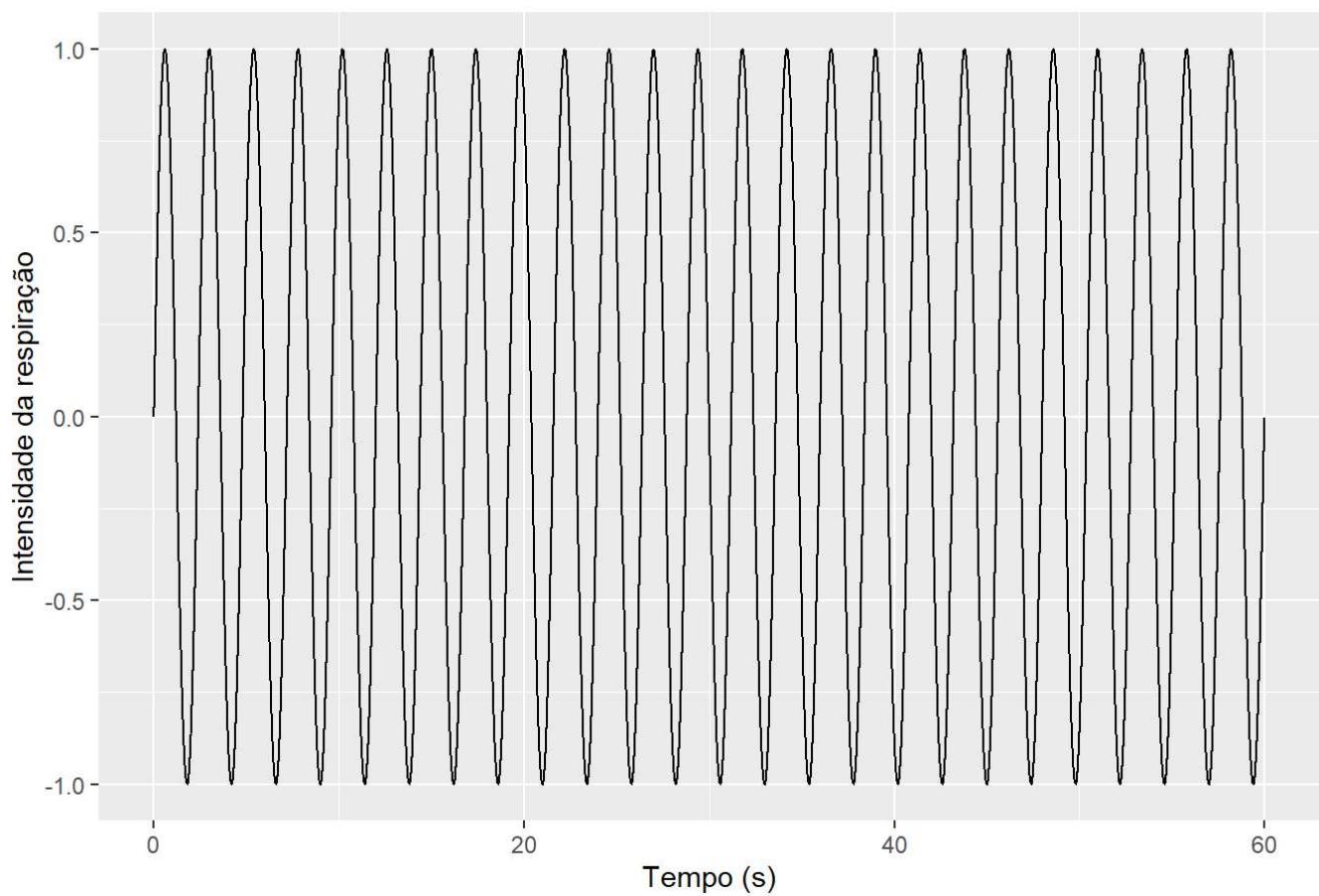
```
usingGgplot(3)
```

Ciclo respiratório: 3 anos de idade



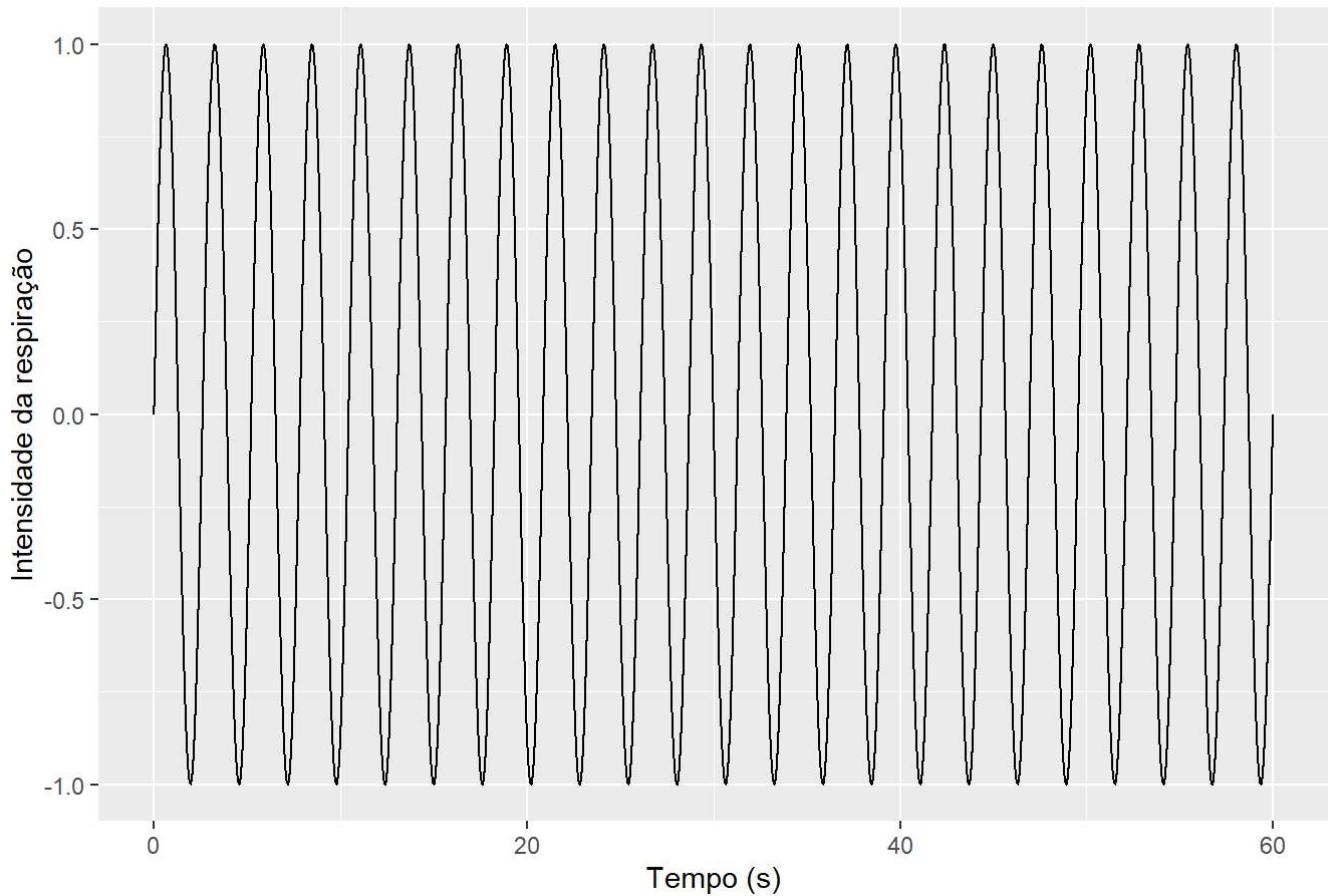
```
usingGgplot(4)
```

Ciclo respiratório: 6 anos de idade



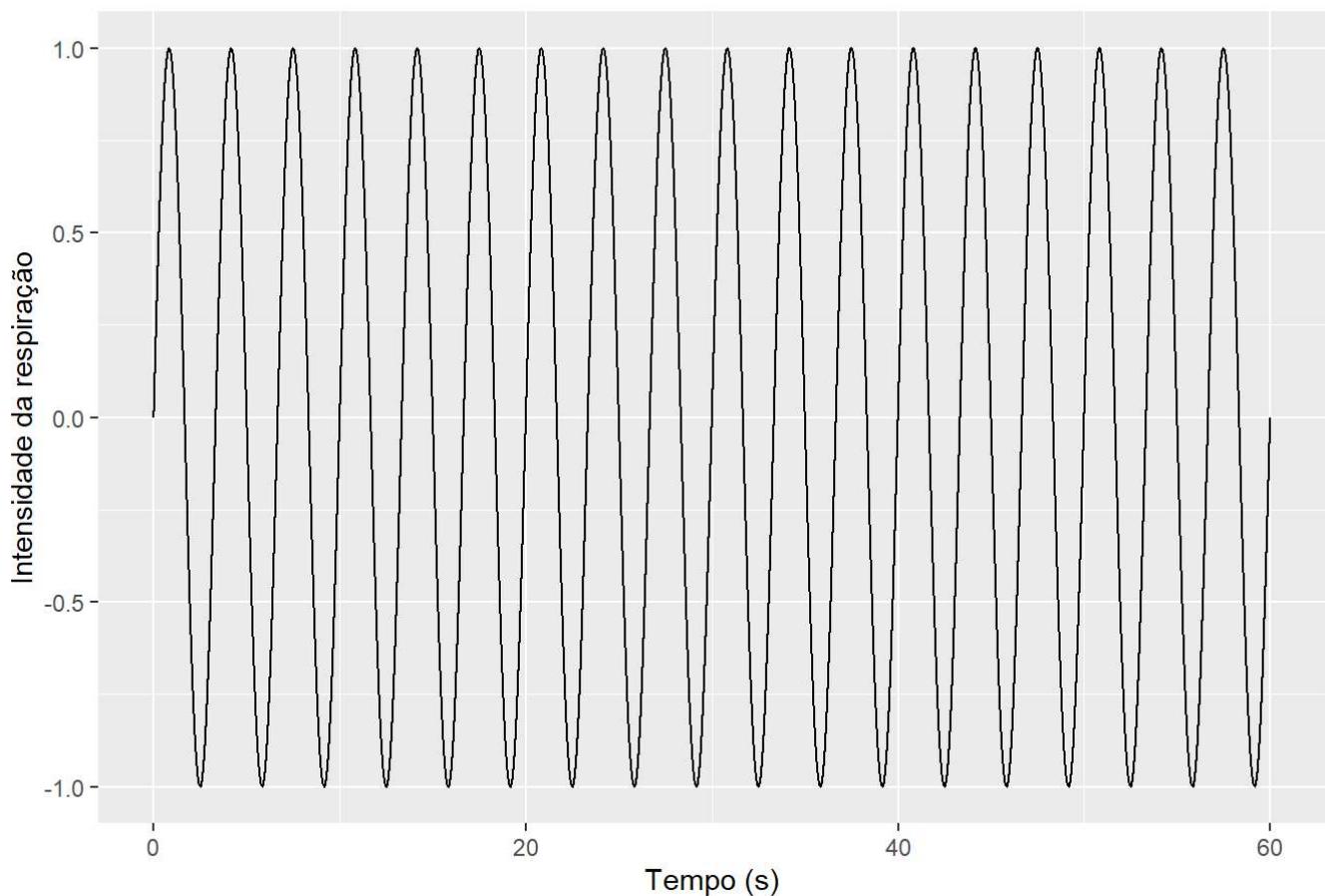
```
usingGgplot(5)
```

Ciclo respiratório: 10 anos de idade



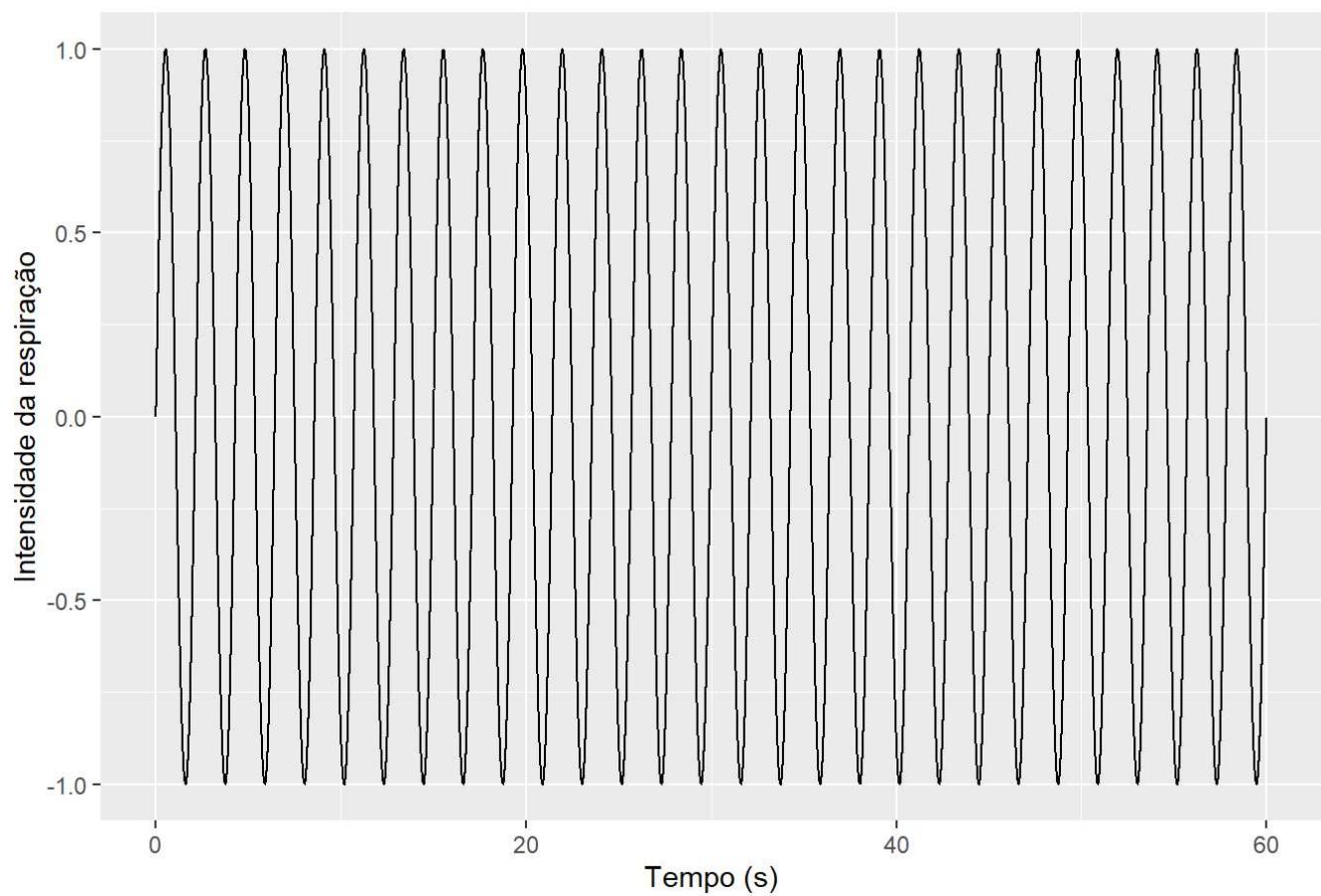
```
usingGgplot(6)
```

Ciclo respiratório: 42 anos de idade



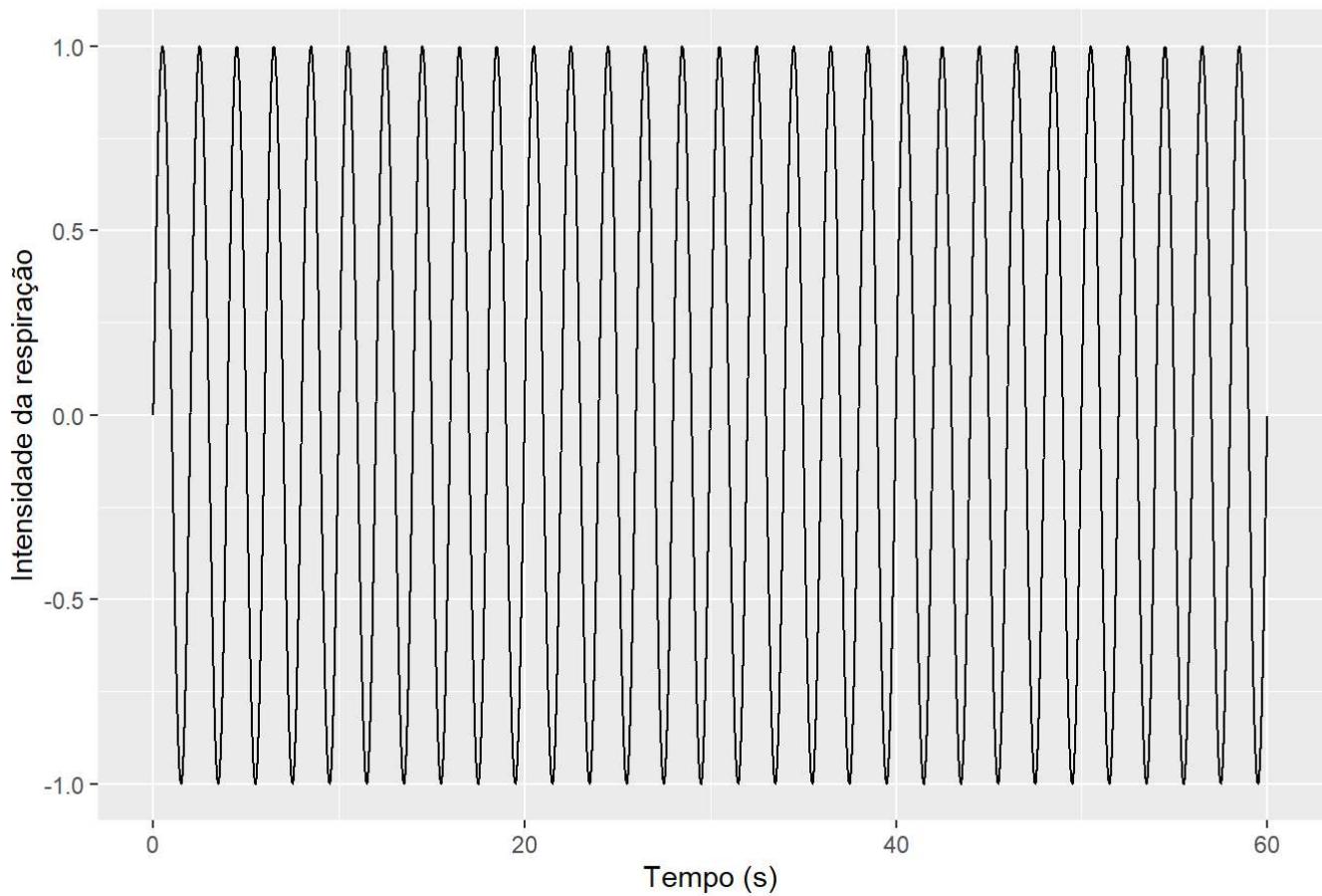
```
usingGgplot(7)
```

Ciclo respiratório: 70 anos de idade



```
usingGgplot(8)
```

Ciclo respiratório: 90 anos de idade

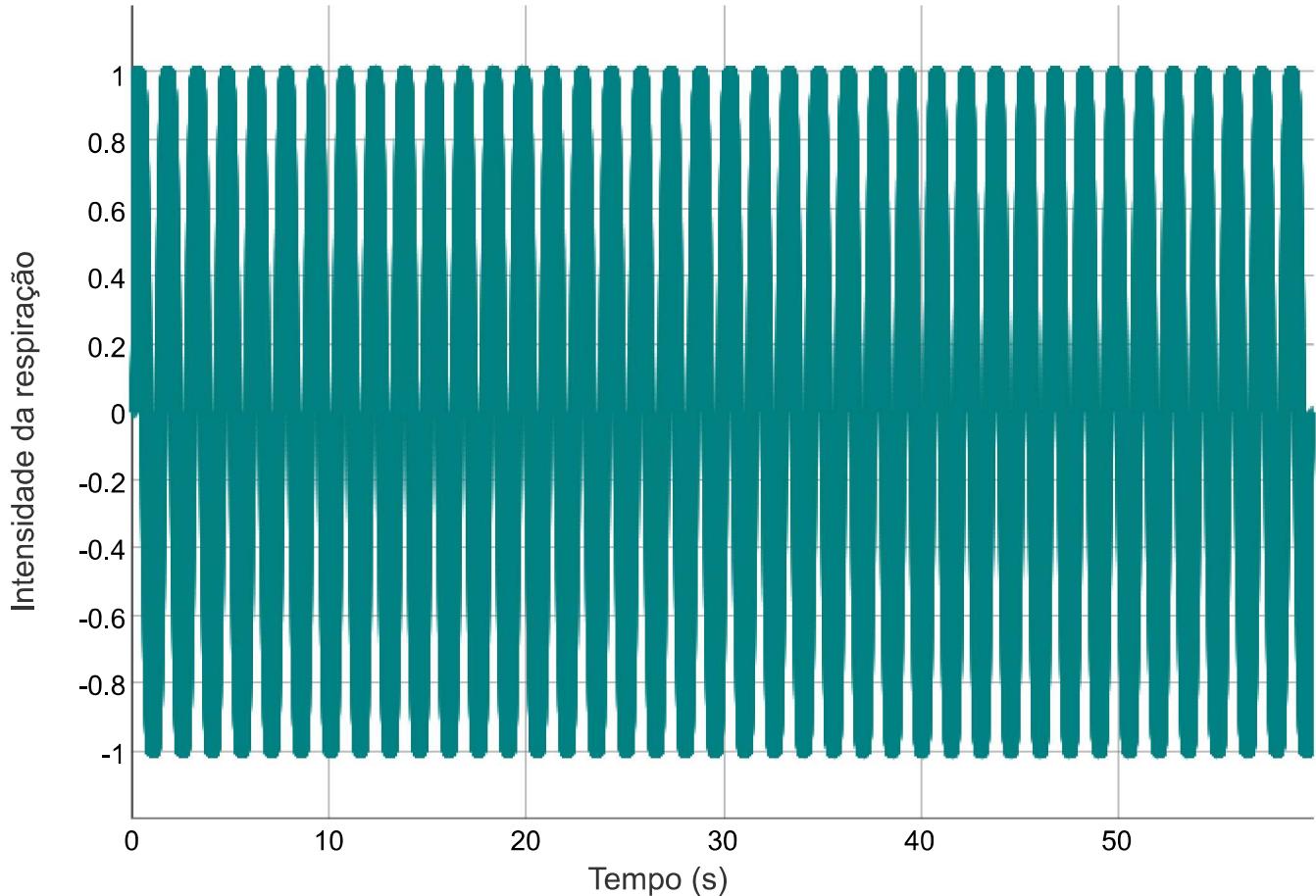


Exercício 4

Escolha um sinal simulado na questão 1, e plote-o utilizando a função **dyStemSeries**. Por que há um intervalo temporal entre cada amostra?

```
data.frame(time=t, array.cat[1]) %>%
  dygraph(xlab='Tempo (s)', ylab='Intensidade da respiração', main=array.title[1]) %>%
  dyOptions(stemPlot=TRUE)
```

Ciclo respiratório: 6 semanas de idade



O intervalo entre as amostras é uma decorrência direta da natureza discreta do sinal gerado. Essa separação é exatamente o tempo entre a tomada de cada amostra do sinal. Ao contrário desse caso, um sinal contínuo não haveria separação, mas este tipo não pode ser representado por um computador, pois necessitaria de infinitas amostras. A quantidade de amostras pode ser aumentada para um intervalo de tempo cada vez menor, o que ocasionaria na qualidade do sinal discreto se aproximando cada vez mais do sinal analógico, porém sempre com erros.

Exercício 5

Explique a estrutura básica de um arquivo EDF []
((<https://www.edfplus.info/specs/edf.html>)).

De um ponto de vista macroscópico, um arquivo EDF é formado por um **cabeçalho** e os **dados gravados**. Os dados dizem respeito a gravações digitalizadas ininterruptas e multicanal. Os detalhes das duas estruturas principais do arquivo EDF serão explicadas abaixo.

- Cabeçalho:
 - Especifica características técnicas dos sinais registrados;
 - Tamanho variável;
 - Primeiros 256 bytes identificam:
 - Paciente;
 - Versão do formato EDF usado;
 - Identificação sobre gravação;
 - Informação temporal da gravação;
 - Número de registros;
 - Número de sinais;

- Haverá um sub-cabeçalho de 256 bytes para cada sinal, contendo:
 - Tipo do sinal;
 - Calibração da amplitude;
 - Número de amostras em cada registro (taxa de amostragem pode ser encontrada, uma vez que a duração é conhecida);
- O sub-cabeçalho explicita que pode haver diferente ganhos de amplitude e diferentes frequências de amostragem para cada sinal;
- Dados gravados:
 - Contém registros poligráficos multicanais;
 - Os registros são divididos em épocas consecutivas de mesma duração;
 - Registro da duração do sinal;

Exercício 6

Repita todos os exercícios apresentados neste módulo. Os comandos utilizados devem ser inseridos como resposta a esta questão.

```
library(dygraphs)

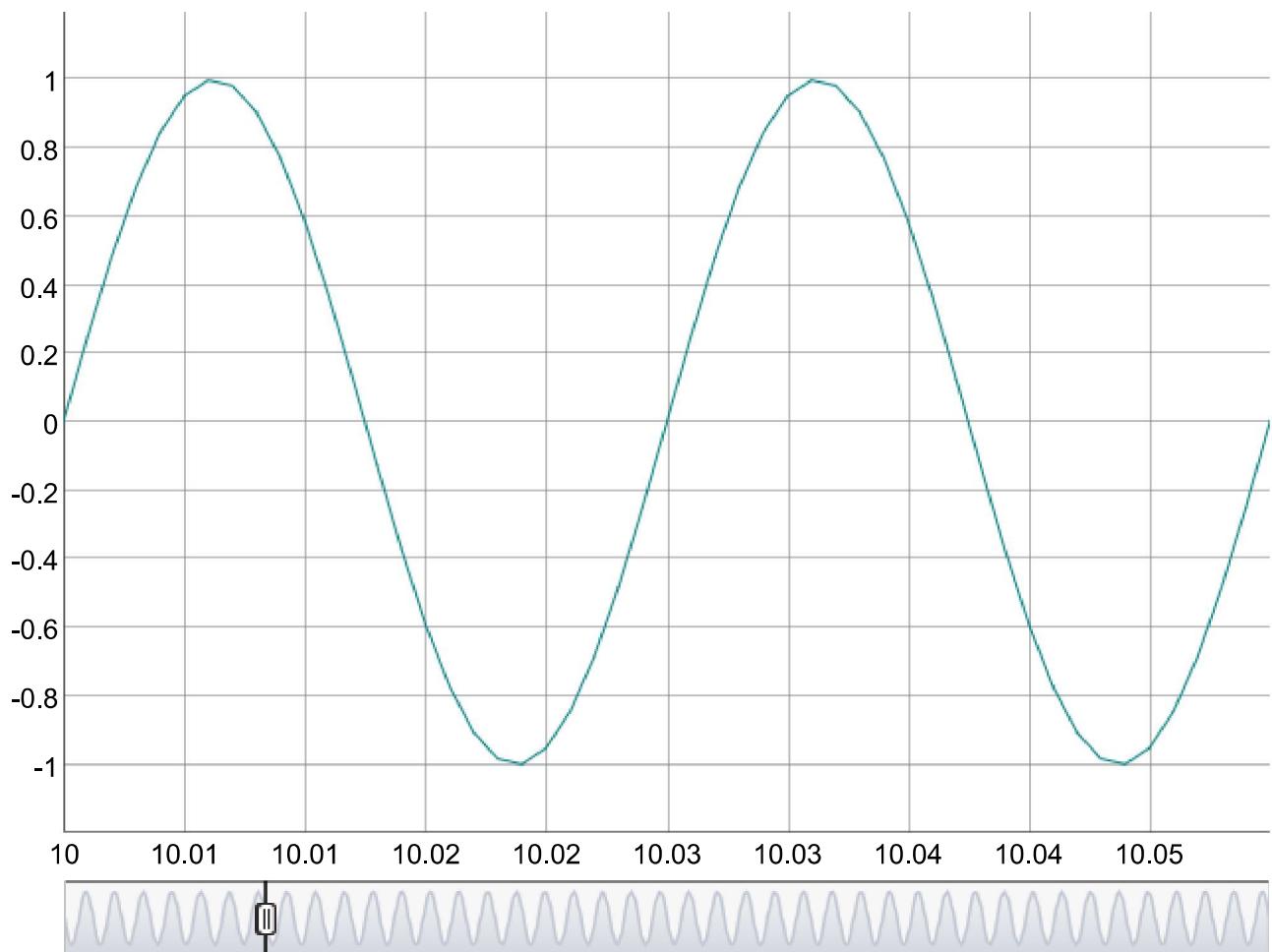
tf <- 60
dt <- 0.001

t <- seq(from = 0, to = tf, by = dt )

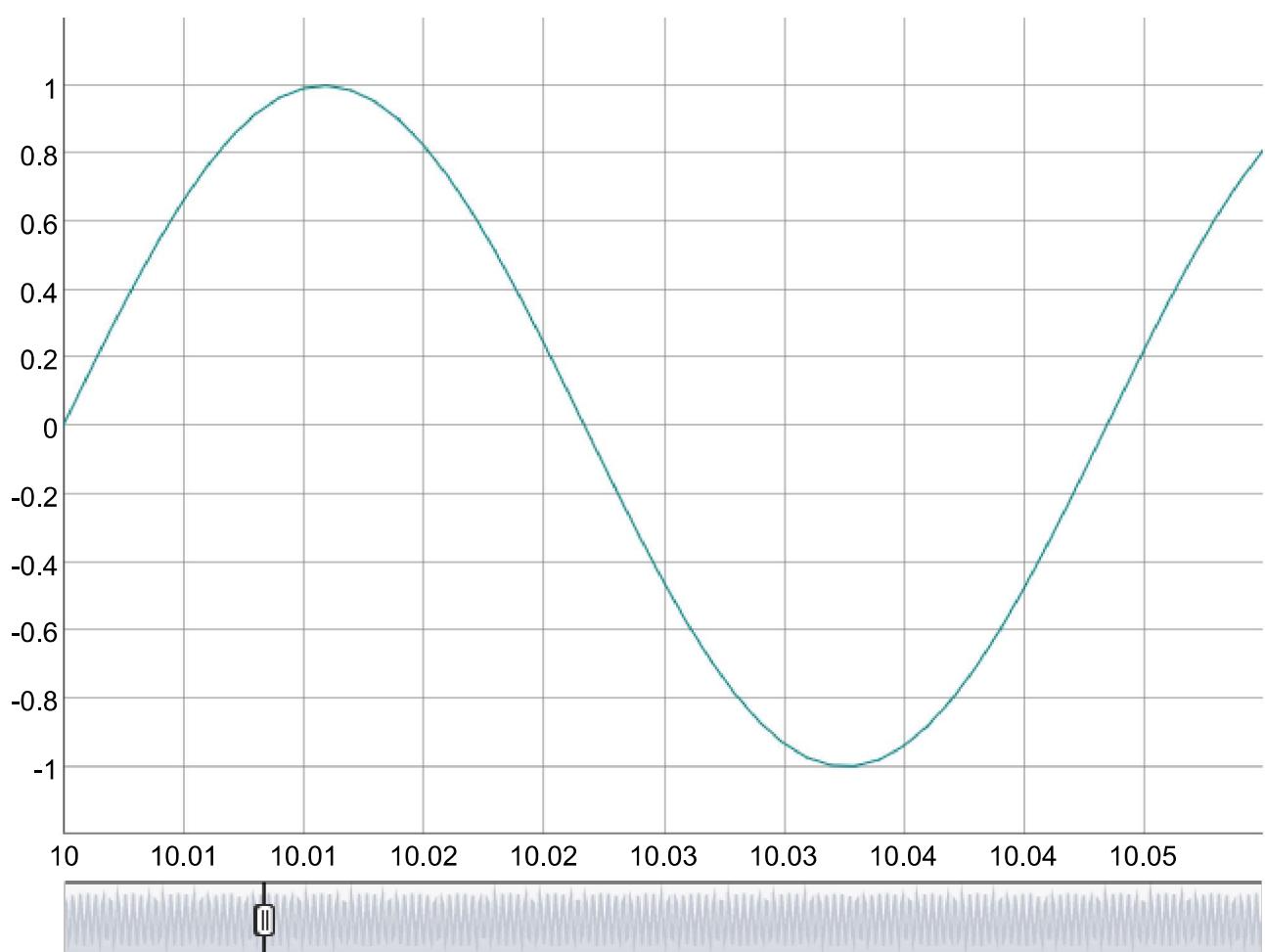
f1 <- 40
y1 <- sin(2*pi*t*f1)

f2 <- 23
y2 <- sin(2*pi*t*f2)

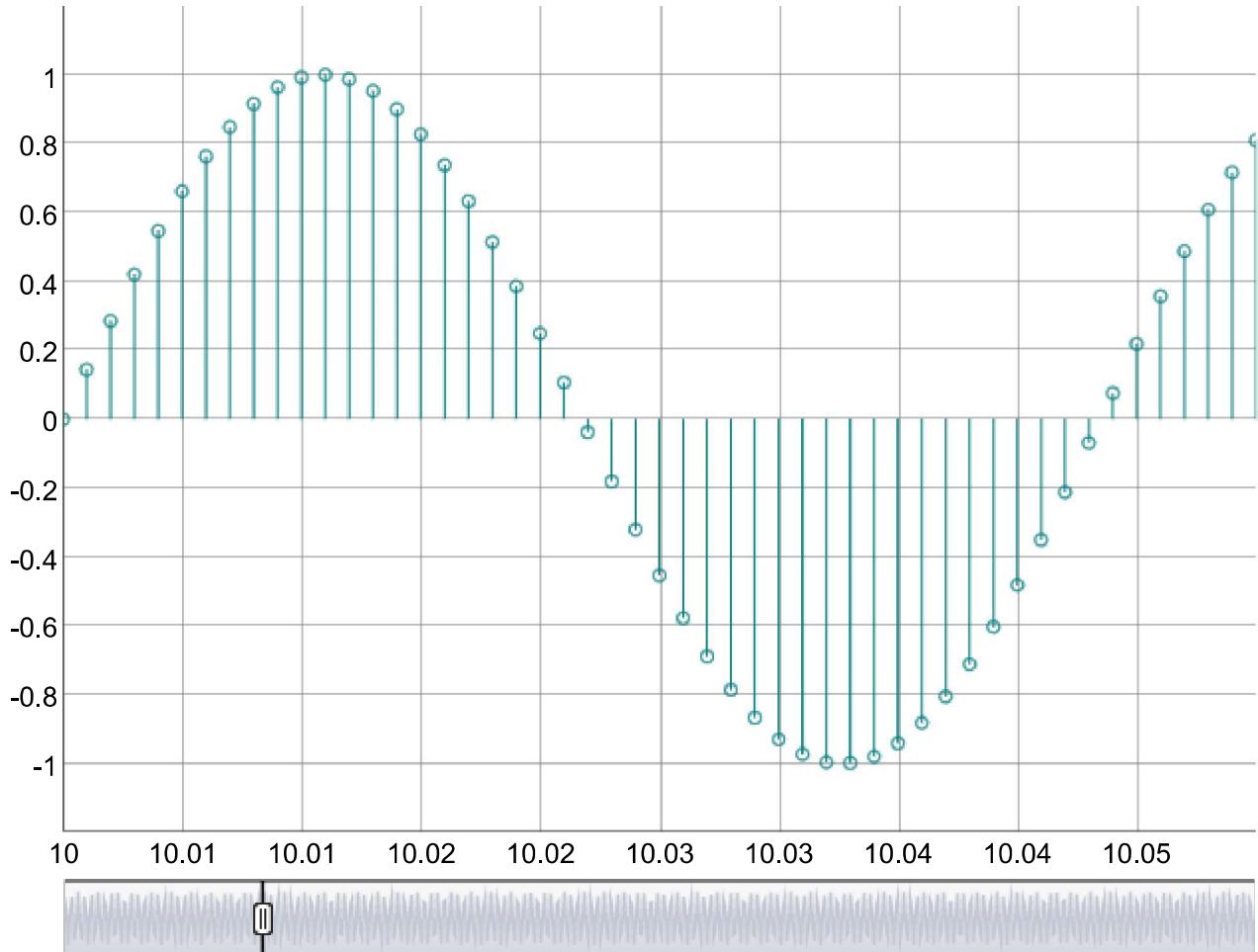
tWindow <- c(10, 11)
data.frame(time=t, y1) %>% dygraph(group = "G1") %>% dyRangeSelector(dateWindow = tWindow)
```



```
data.frame(time=t, y2) %>% dygraph(group = "G1") %>% dyRangeSelector(dateWindow = tWindow)
```



```
tWindow <- c(10, 10.05)
data.frame(time=t, y2) %>%
  dygraph(group = "G1") %>%
  dyRangeSelector(dateWindow = tWindow) %>%
  dyOptions(stemPlot=TRUE)
```

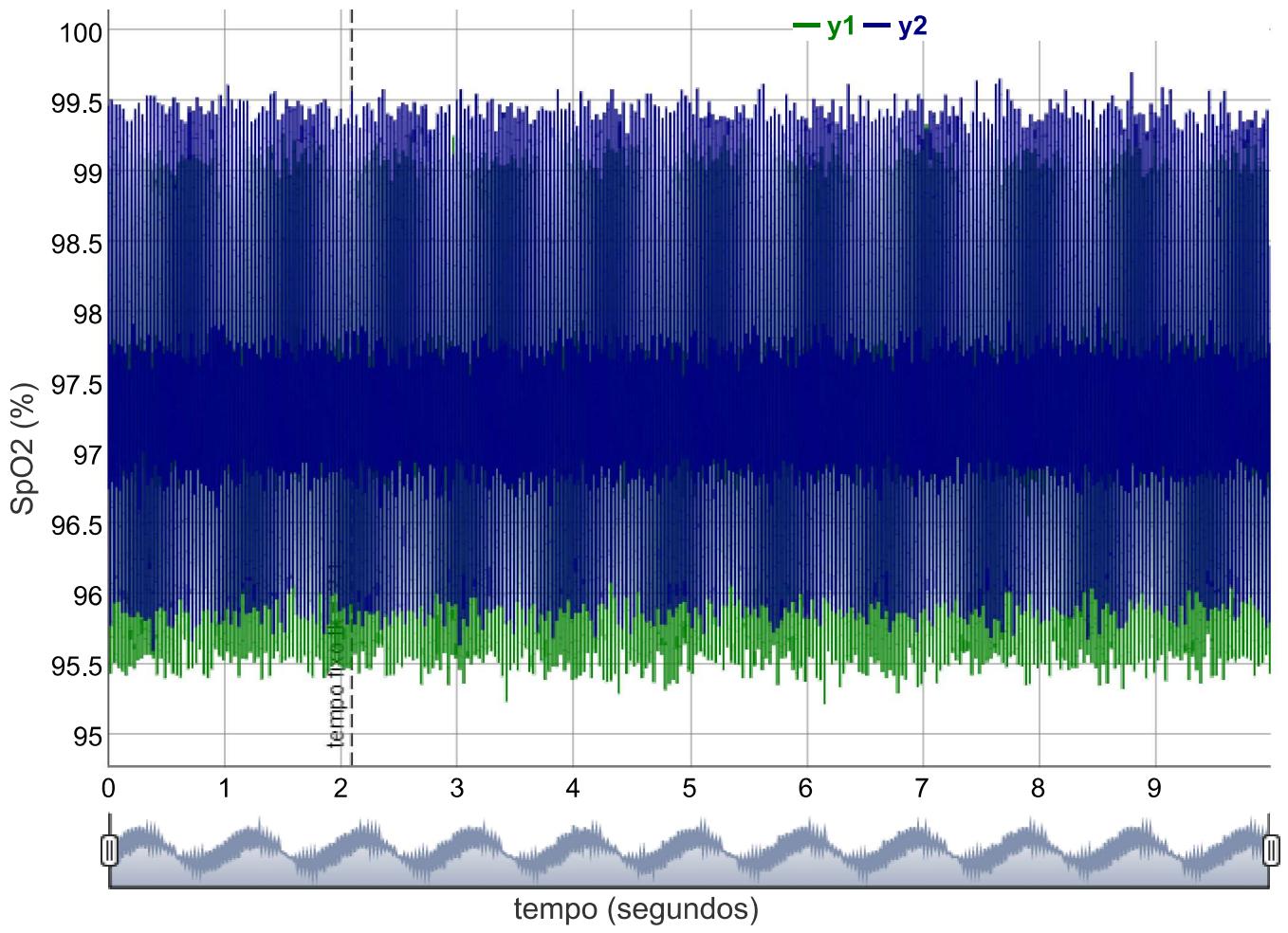


```
set.seed(123)

t <- seq(from = 0, to = 10, by = 0.001)

y1 <- 97 + sin(2*pi*60*t)+sin(2*pi*30*t)+rnorm(length(t), mean = 0.3, sd = 0.1)
y2 <- 97 + sin(2*pi*60*t+(pi/6))+sin(2*pi*30*t+(pi/6))+rnorm(length(t), mean = 0.5, sd = 0.1)

data.frame(t,y1,y2) %>%
  dygraph(xlab = "tempo (segundos)", ylab = "Sp02 (%)") %>%
  dyRangeSelector() %>%
  dyEvent("2.1","tempo fixo tk = 2.1", labelLoc = "bottom")
```

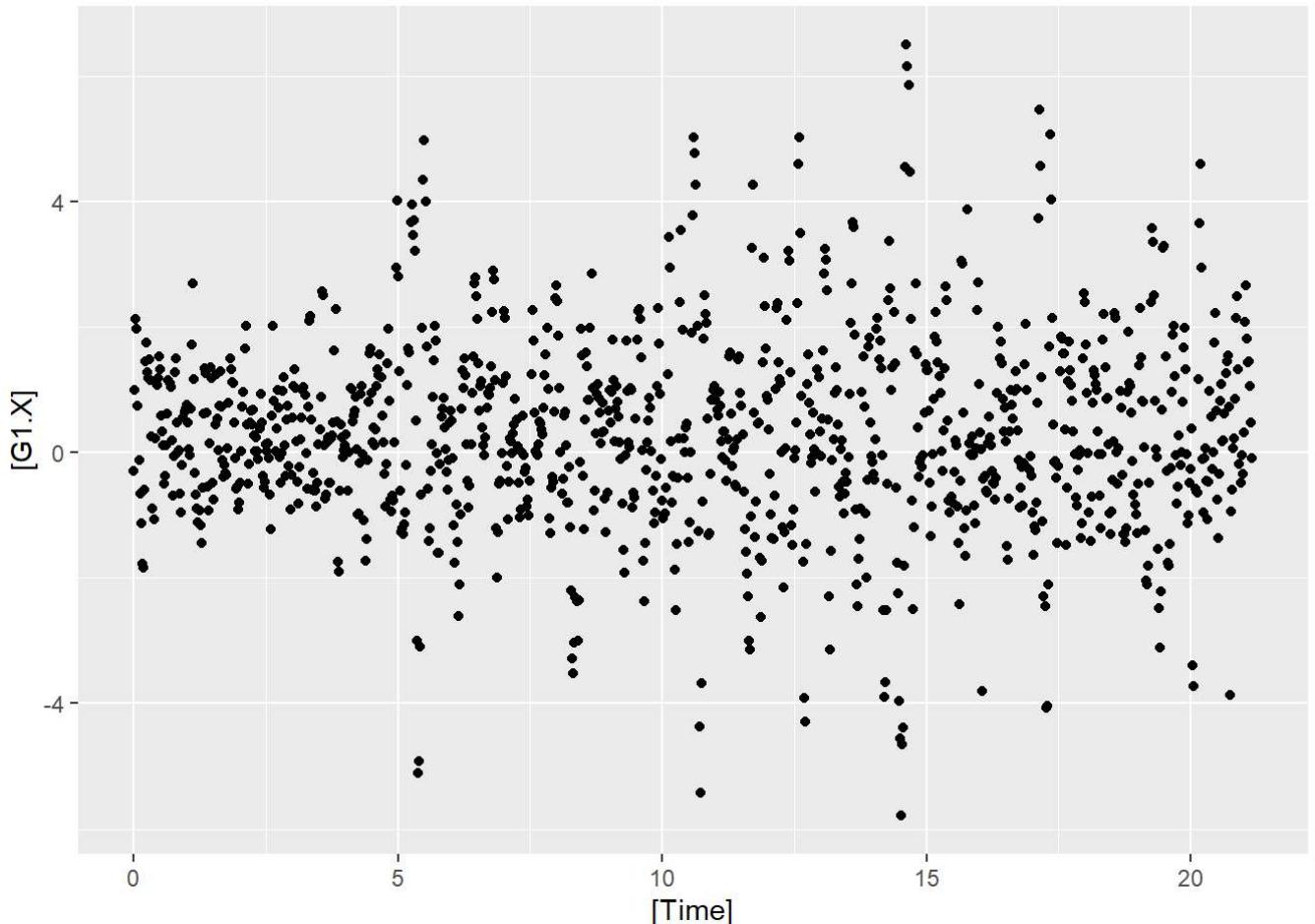


```
library(openxlsx)
library(ggplot2)
library(edfReader)

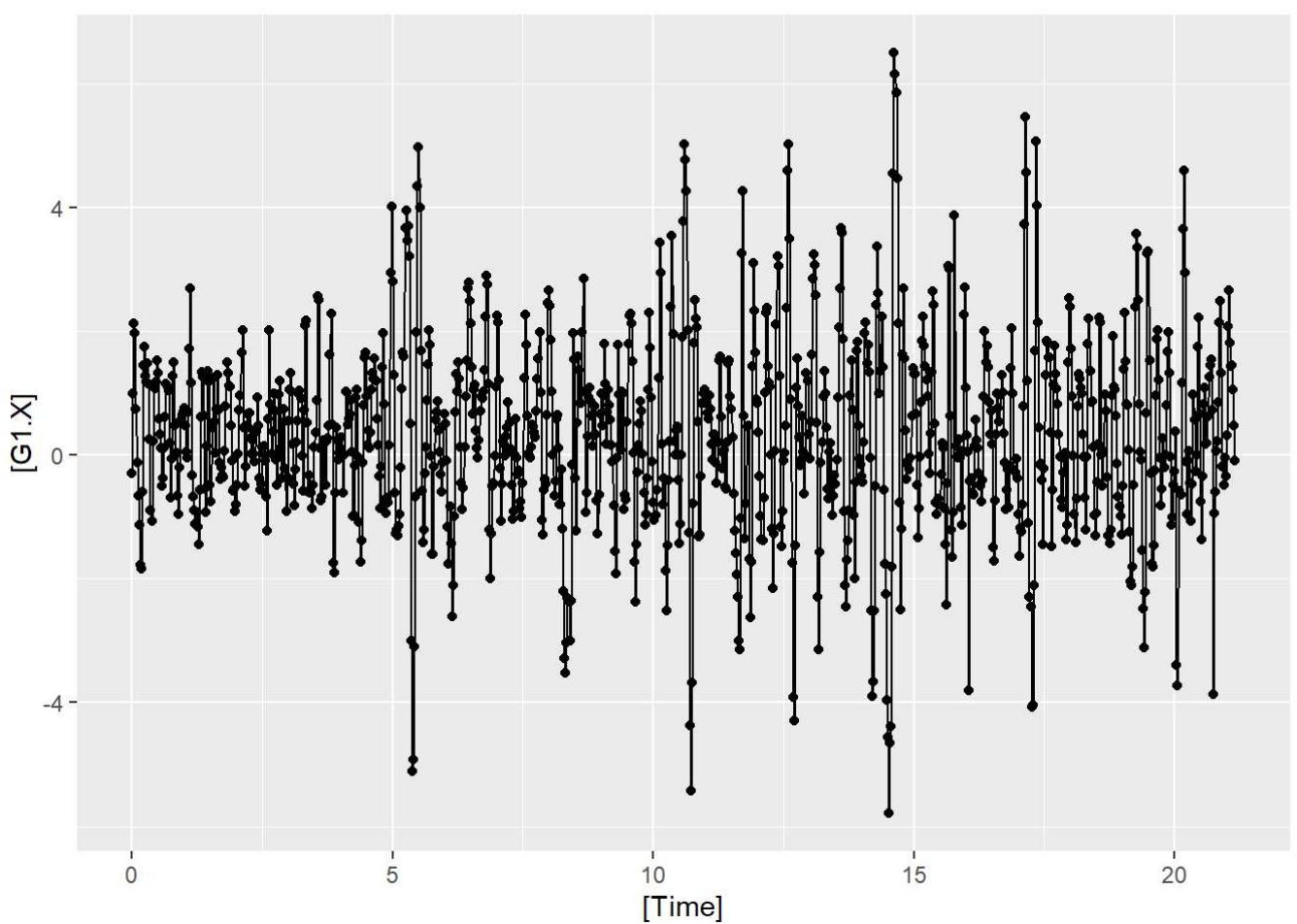
# xlsx

df1 <- read.xlsx("Dados/V16C1RCC92.xlsx", sheet = 1, skipEmptyRows = FALSE)

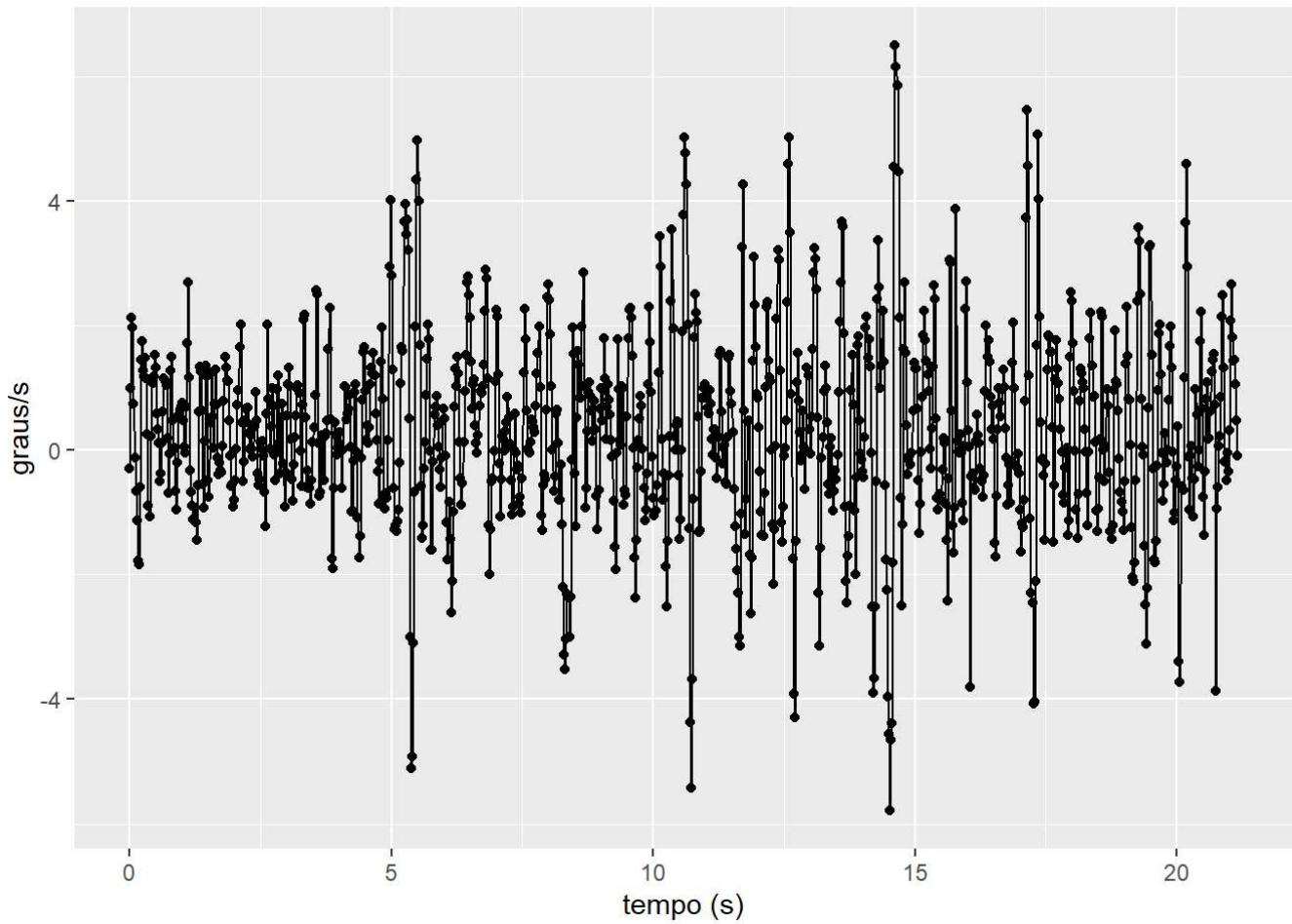
qplot(`[Time]`, `[G1.X]`, data = df1)
```



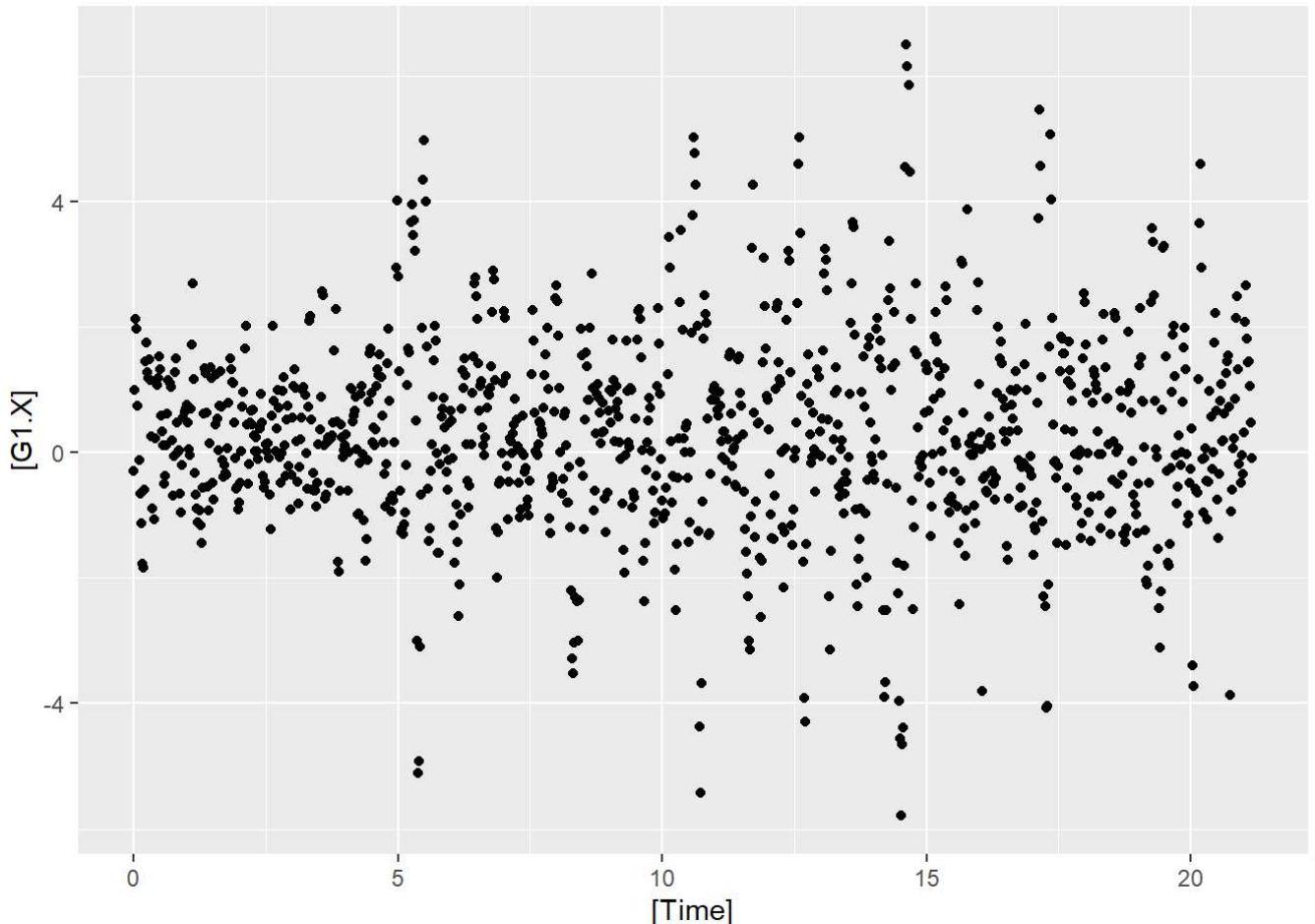
```
qplot(`[Time]`, `[G1.X]`, data = df1, geom=c("point", "line"))
```



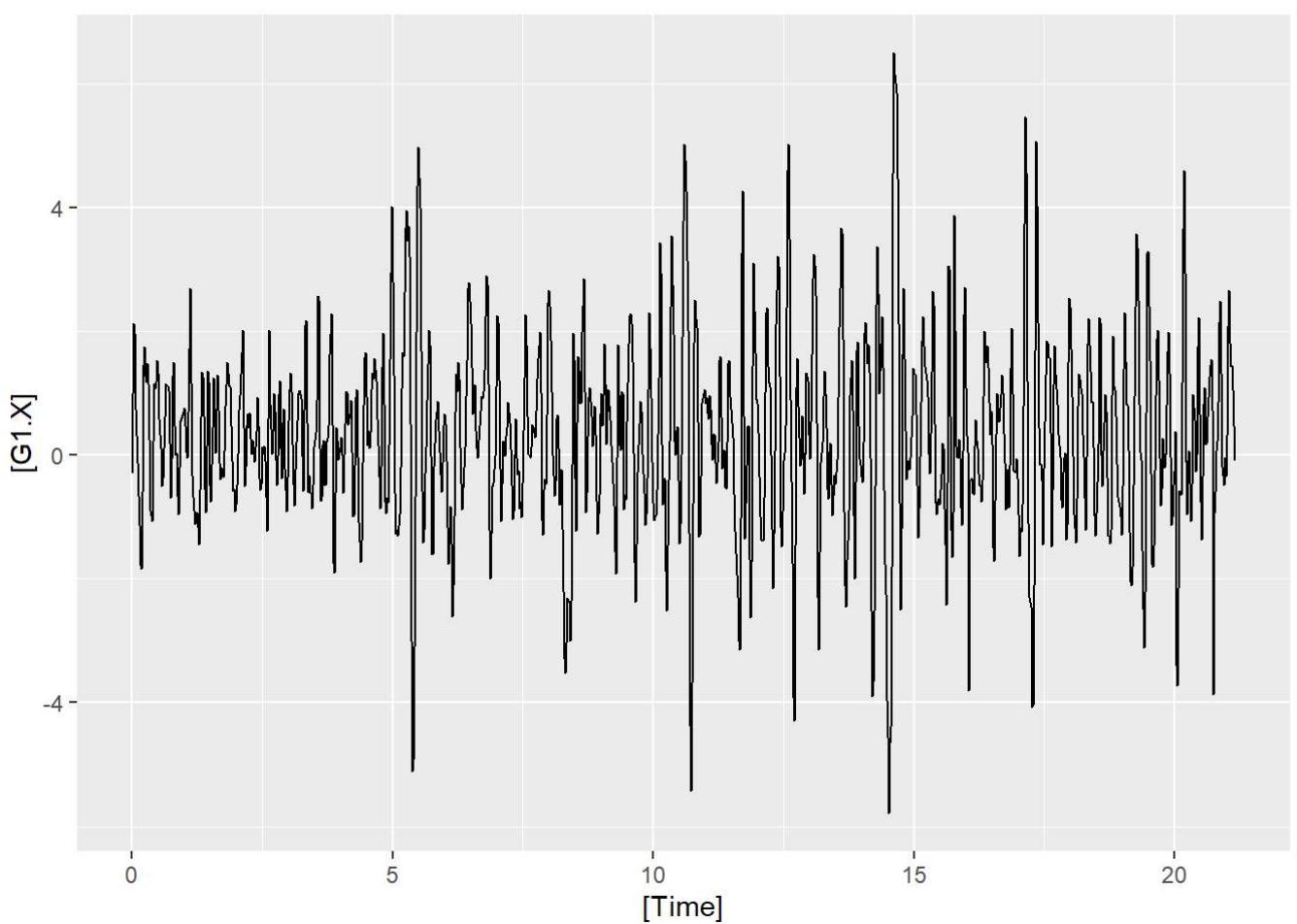
```
p <- qplot(`[Time]`, `[G1.X]`, data = df1, geom=c("point", "line"))
p <- p + labs(x = "tempo (s)", y="graus/s")
print(p)
```



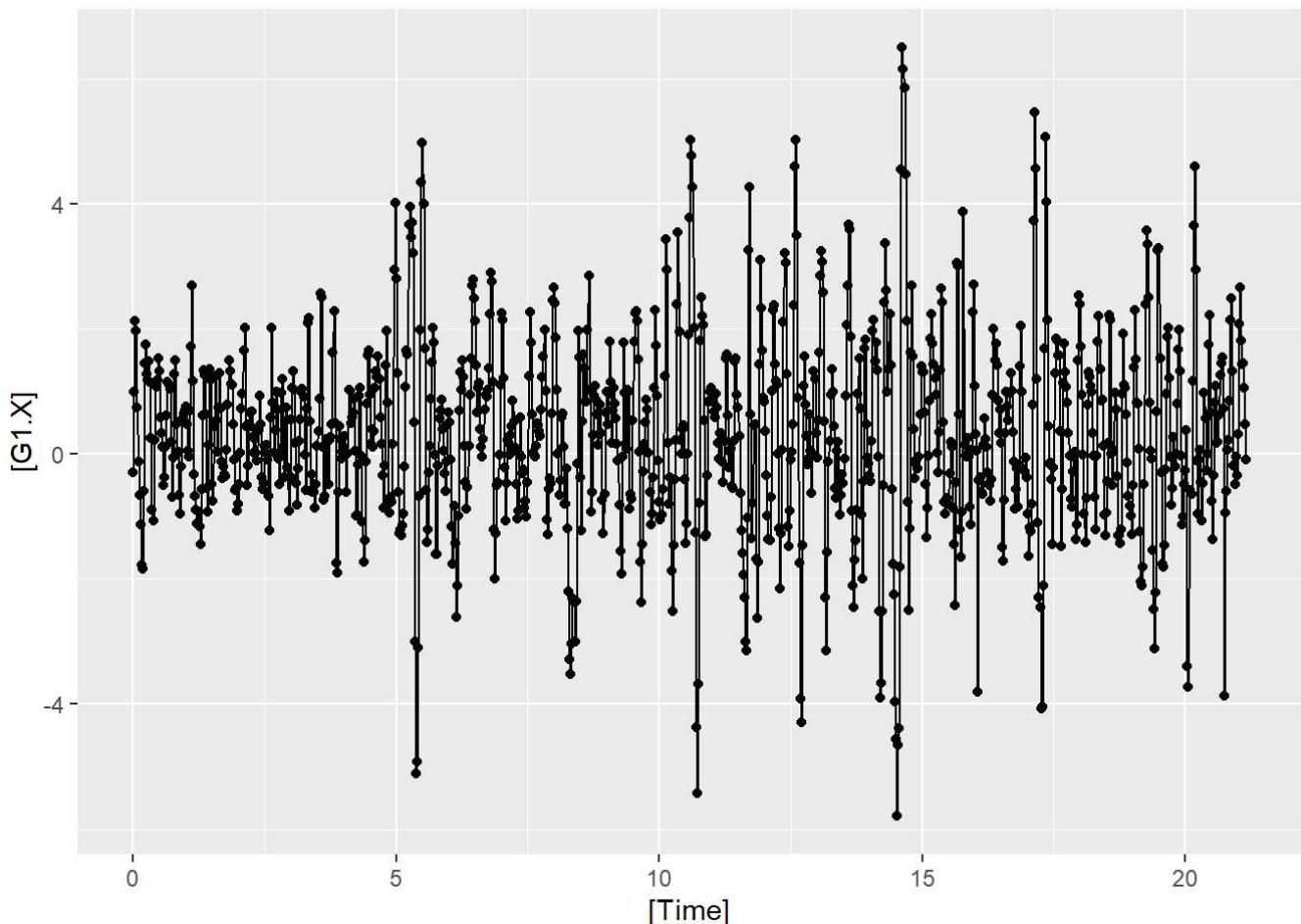
```
ggplot(data=df1, aes(x=`[Time]`, y=`[G1.X]`)) + geom_point()
```



```
ggplot(data=df1, aes(x=[Time], y=[G1.X])) + geom_line()
```



```
ggplot(data=df1, aes(x=`[Time]`, y=`[G1.X]`)) + geom_point() + geom_line()
```

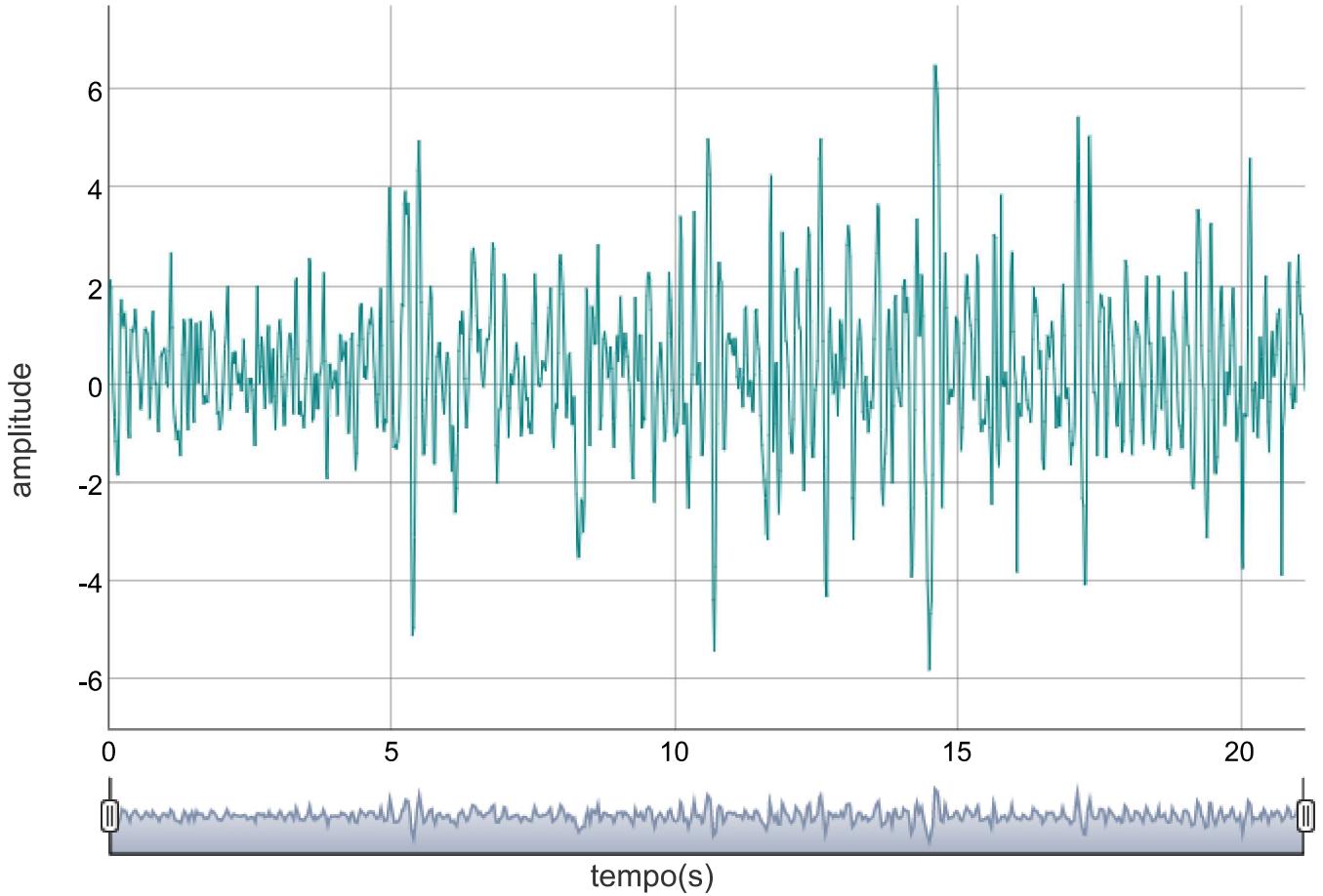


```
# txt
df2 <- read.table("Dados/V16C1RCC92.txt", header = TRUE, sep = "\t", dec = ".", skip = 1, quote = "", comment.char = "")

df2.1 <- data.frame(time=df2$X.Time., df2$X.G1.X.)

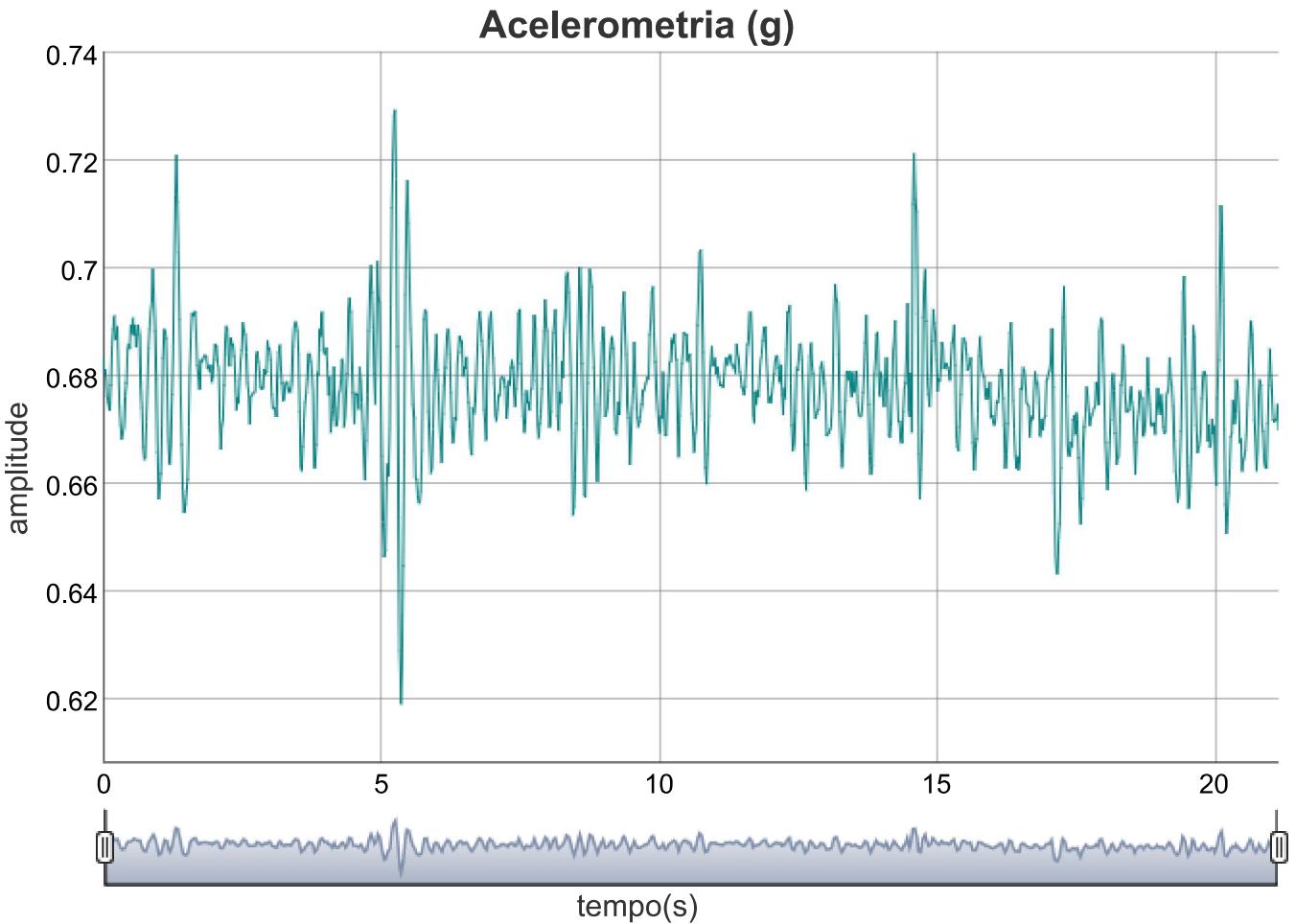
dygraph(df2.1, main = "Velocidade angular (dps - degree per second)") %>%
  dyRangeSelector() %>%
  dyAxis("x", label = "tempo(s)") %>%
  dyAxis("y", label = "amplitude")
```

Velocidade angular (dps - degree per second)



```
# CSV
df3 <- read.csv("Dados/V16C1RCC92.csv", header = TRUE, sep = ";")
df3.1 <- data.frame(time=df3[[1]], amp=df3$X.A1.X.)

dygraph(df3.1,main = "Acelerometria (g)") %>% dyRangeSelector() %>%
  dyAxis("x", label = "tempo(s"))%>%
  dyAxis("y", label = "amplitude")
```

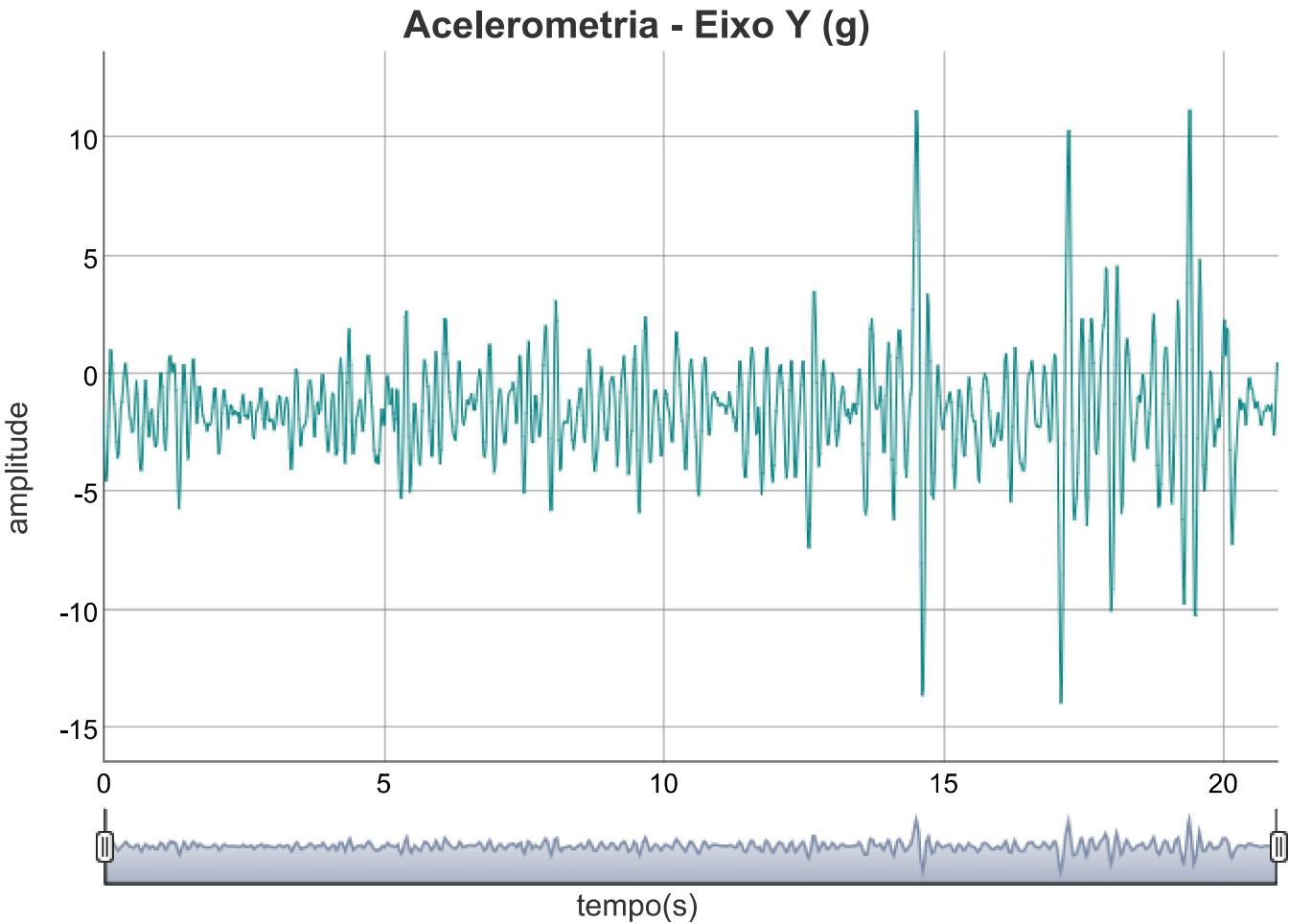


```
# edf
CHdr <- readEdfHeader("Dados/V16C1RCC92.edf")
Signals <- readEdfSignals(CHdr)

tt = seq(from=0, to=1049, by=1) * 1/Signals$`Ch: 0-:G1.X`$sRate

df4 <- data.frame(time= tt, amp=Signals$`Ch: 1-:G1.Y`$signal)

dygraph(df4,main = "Acelerometria - Eixo Y (g)") %>% dyRangeSelector() %>%
  dyAxis("x", label = "tempo(s"))%>%
  dyAxis("y", label = "amplitude")
```



Exercício 7

A base de dados disponível na plataforma Moodle foi coletada utilizando-se o dispositivo TREMSEN (Precise Tremor Sensing), que é um dispositivo para a coleta de dados iniciais (i.e., movimentos).

A coleta de dados foi executada considerando os seguinte protocolo:

- Um acelerômetro triaxial foi posicionado no dorso da mão, sobre o osso Capitato.
- O eixo X do acelerômetro foi alinhado paralelamente à terceira falange distal.
- Os dados foram coletados enquanto o participante realizava os seguintes movimentos: – 5 flexões do punho; – 5 extensões do punho; – 5 aduções do punho; – 5 abduções do punho.
- Os dados coletados foram salvos no formato EDF e TXT, e nomeados de acordo com o tipo de movimento realizado (Adução, Flexão e Rotação).

Considerando as informações dadas, você deve:

- Abrir o arquivo TXT no Excel e salvar uma versão deste arquivo no formato xls.
- Salvar o arquivo xls gerado no formato csv.
- Abrir o arquivo coletado em xls no R e gerar um gráfico utilizando a função qplot. Incluir legendas dos eixos x - tempo (s) e y - amplitude (g).
- Abrir o arquivo coletado em csv no R e gerar um gráfico utilizando a função qplot. Incluir as legendas dos eixos x - tempo (s) e y - amplitude (g).
- Abrir o arquivo coletado em edf no R e gerar um gráfico utilizando a função qplot. Incluir as legendas dos eixos x - tempo (s) e y - amplitude (g).

- Incluir todos os resultados e códigos resultantes da execução desta sequência de passos na resposta desta questão.

Os passos de conversão .txt para .xls e de .xls para .csv foram feitos em softwares externos.

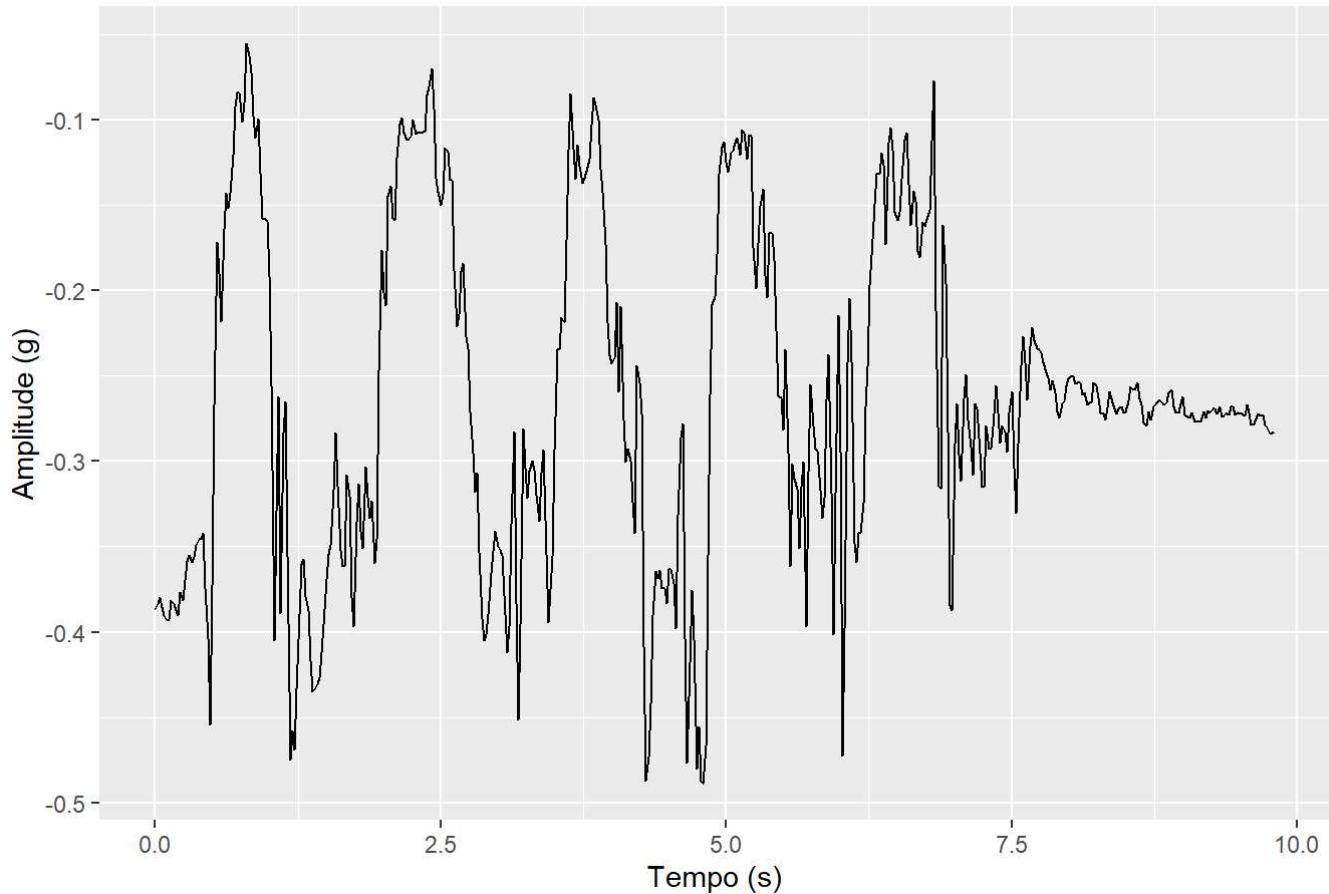
Portanto, em seguida, no trecho abaixo os arquivos .xlsx são lidos para que os gráficos sejam gerados:

```
library(openxlsx)
library(ggplot2)

# Loading
df_add_xlsx <- read.xlsx("Dados/Aduccao.xlsx", skipEmptyRows = FALSE, startRow = 2)
df_flx_xlsx <- read.xlsx("Dados/Flexao.xlsx", skipEmptyRows = FALSE, startRow = 2)
df_rot_xlsx <- read.xlsx("Dados/Rotacao.xlsx", skipEmptyRows = FALSE, startRow = 2)

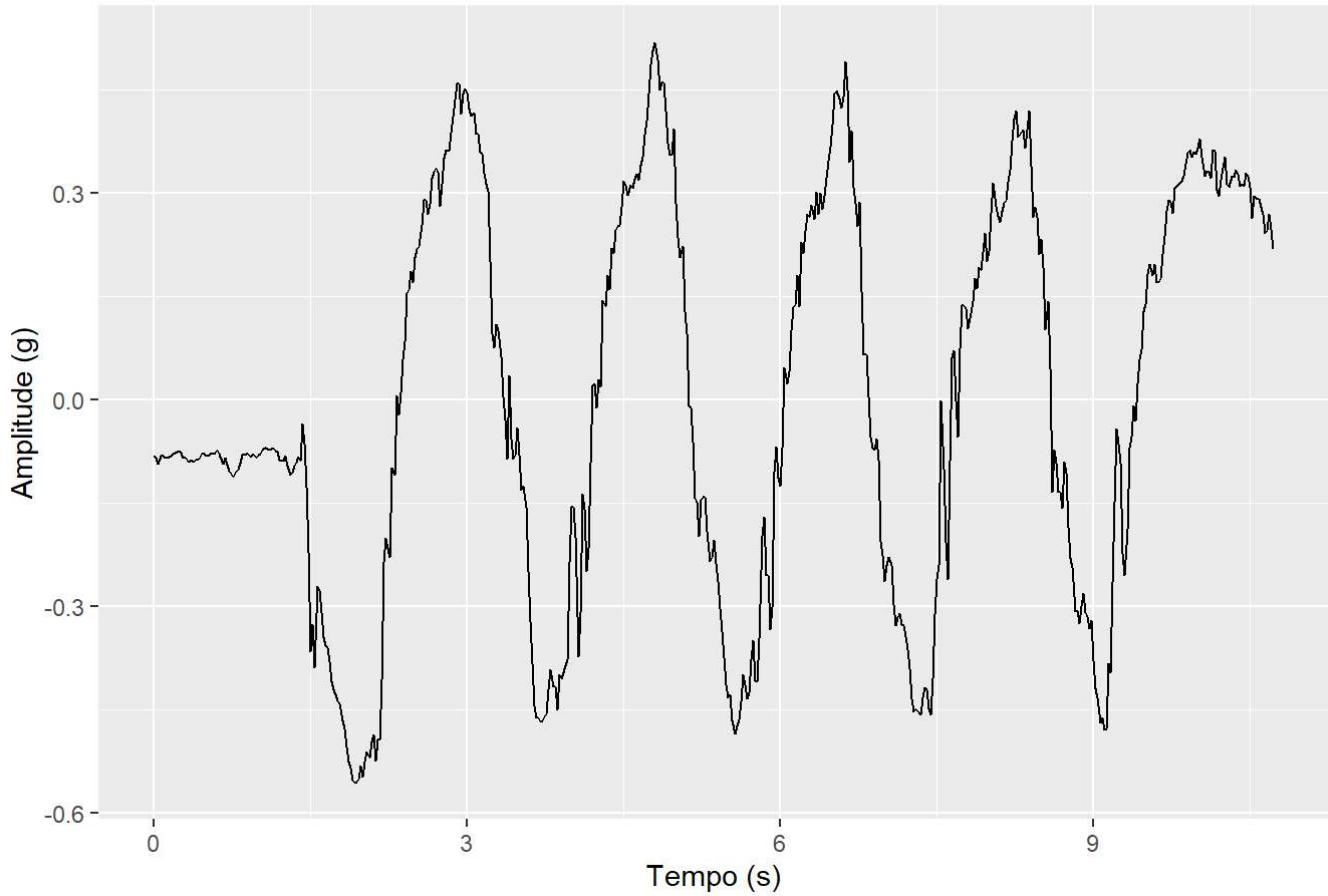
# pLots
qplot(df_add_xlsx$`[Time]`, df_add_xlsx$`[A1.X]`, geom=c("line")) +
  labs(x='Tempo (s)', y='Amplitude (g)',
       title='Acelerômetro 1 de aducção eixo-x')
```

Acelerômetro 1 de aducção eixo-x



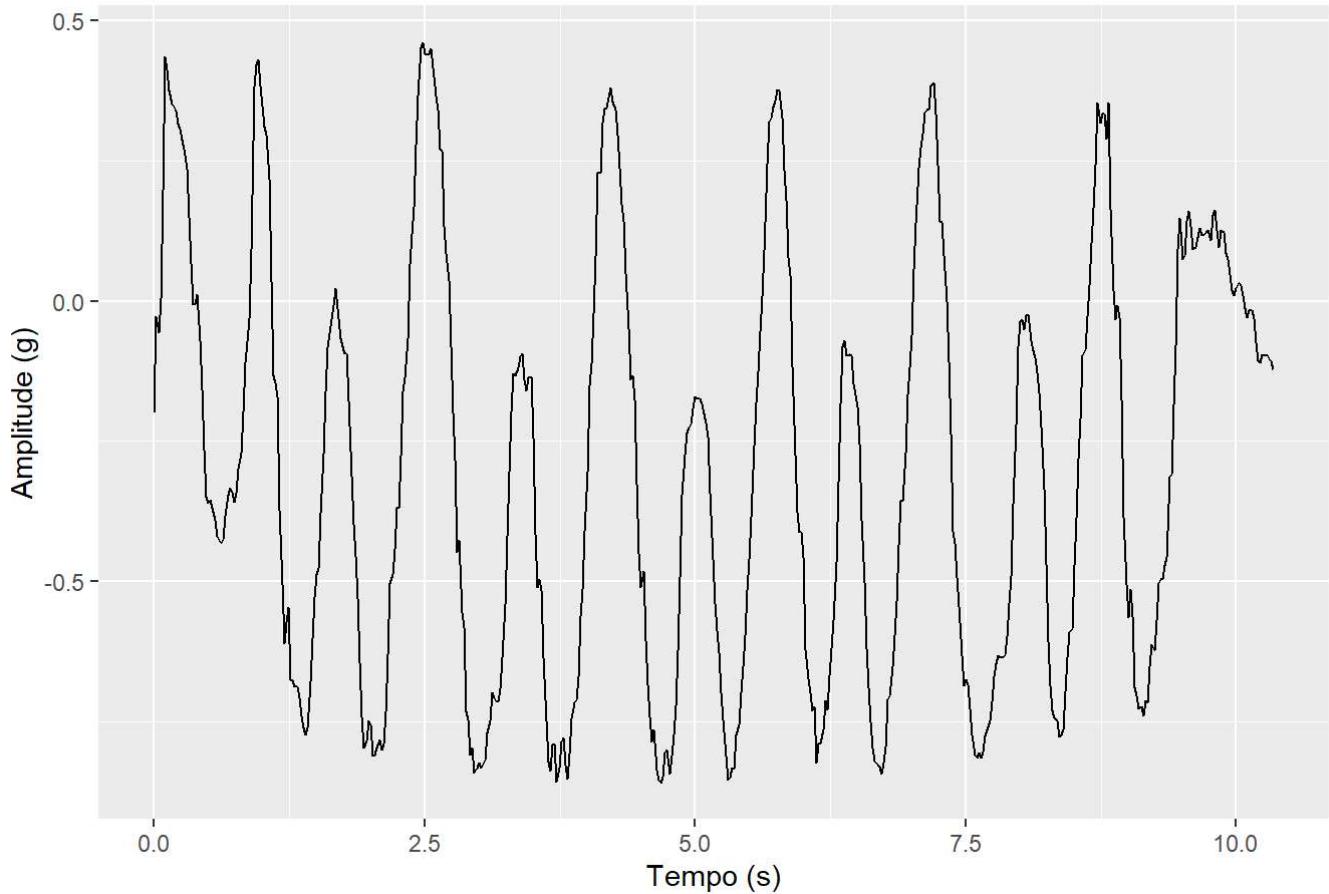
```
qplot(df_flx_xlsx$`[Time]`, df_flx_xlsx$`[A1.Y]`, geom=c("line")) +
  labs(x='Tempo (s)', y='Amplitude (g)',
       title='Acelerômetro 1 de flexão eixo-y')
```

Acelerômetro 1 de flexão eixo-y



```
qplot(df_rot_xlsx$`[Time]`, df_rot_xlsx$`[A1.Z]`, geom=c("line")) +  
  labs(x='Tempo (s)', y='Amplitude (g)',  
    title='Acelerômetro 1 de rotação eixo-z')
```

Acelerômetro 1 de rotação eixo-z

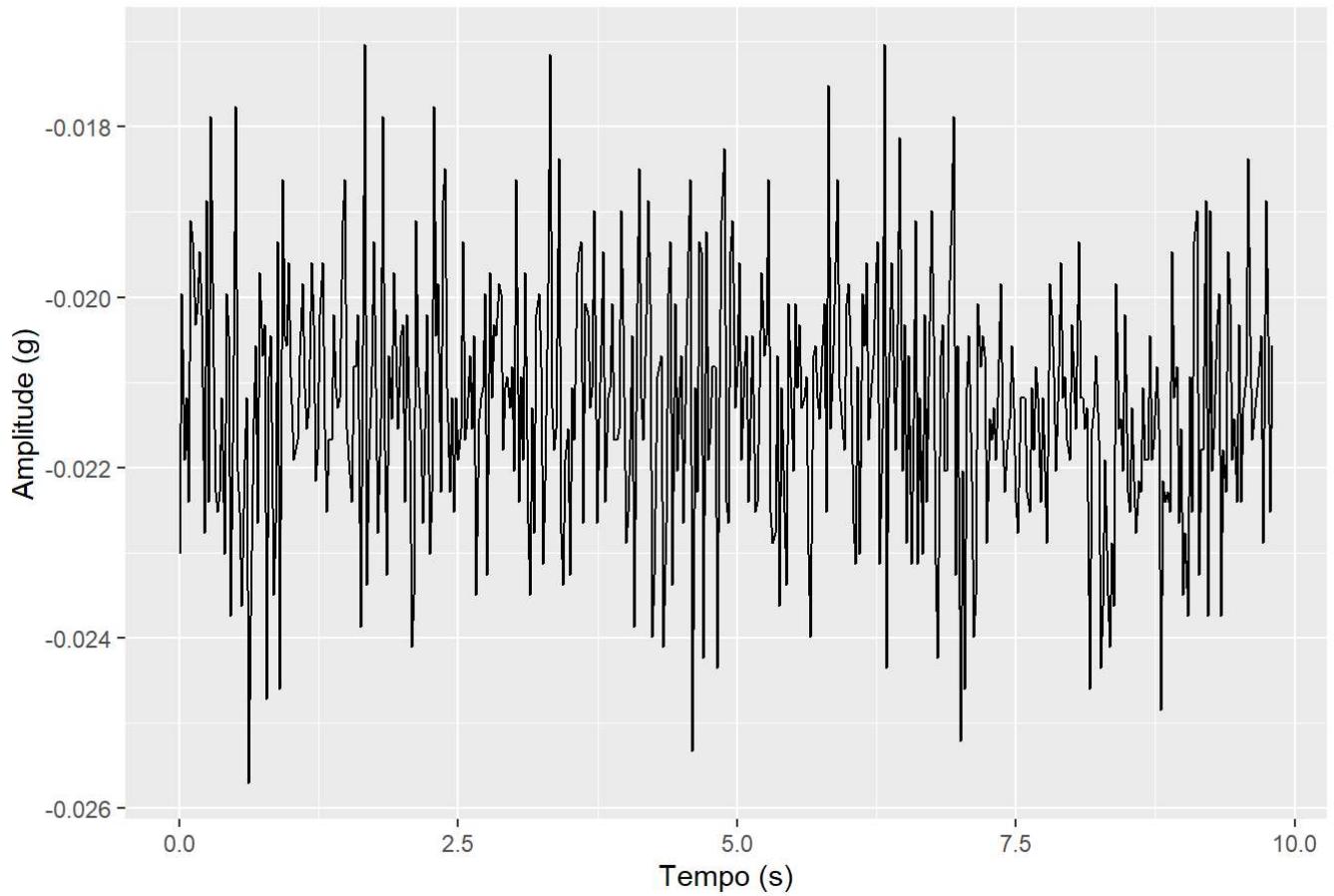


O passo seguinte é carregar os arquivos .csv:

```
# Loading
df_add_csv <- read.csv("Dados/Aduccao.csv", skip=1, header = TRUE, sep = ",",
check.names = FALSE)
df_flx_csv <- read.csv("Dados/Flexao.csv", skip=1, header = TRUE, sep = ",",
check.names = FALSE)
df_rot_csv <- read.csv("Dados/Rotacao.csv", skip=1, header = TRUE, sep = ",",
check.names = FALSE)

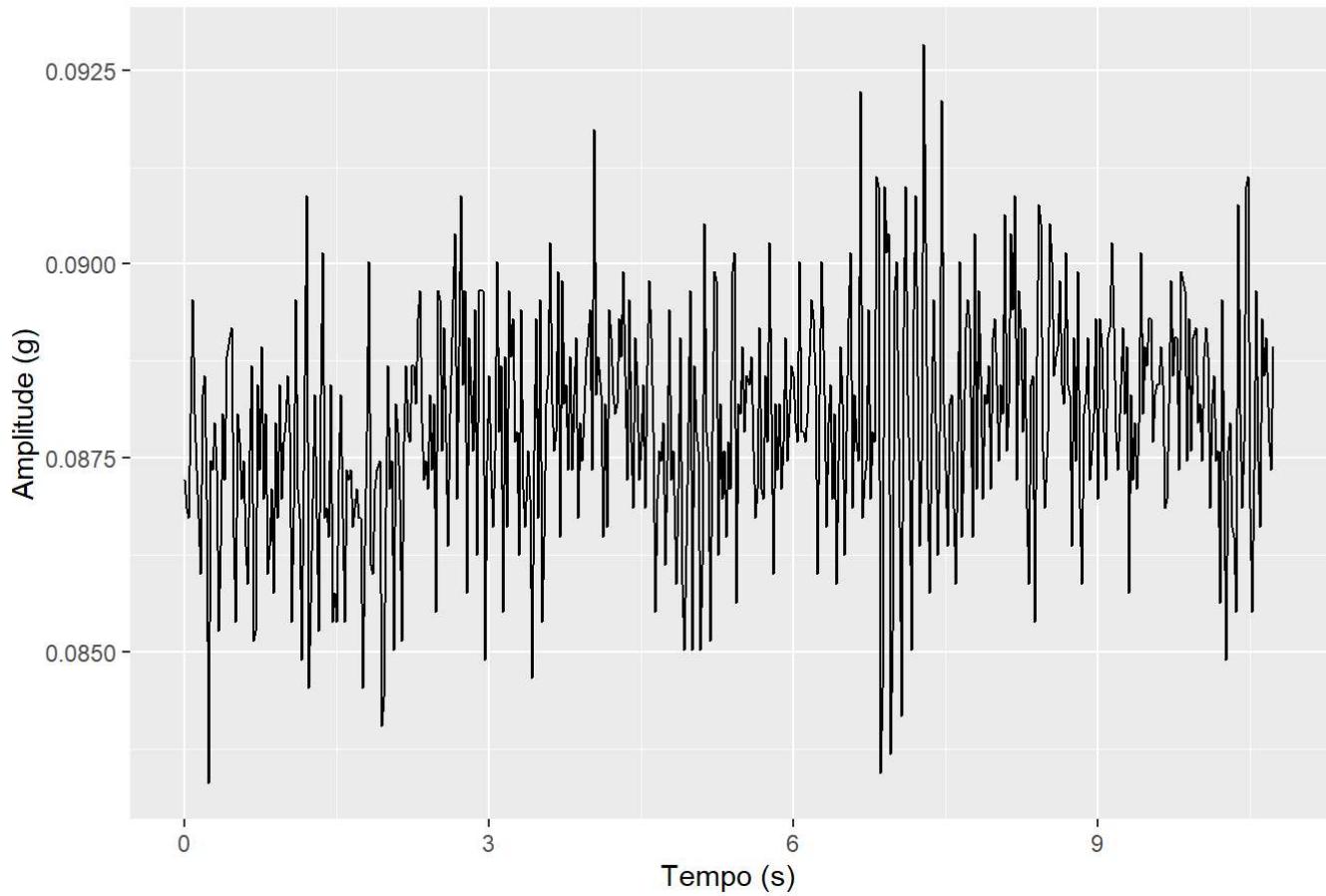
# plots
qplot(df_add_csv$`[Time]`, df_add_csv$`[A2.X]`, geom=c("line")) +
  labs(x='Tempo (s)', y='Amplitude (g)',
       title='Acelerômetro 2 de aducção eixo-x')
```

Acelerômetro 2 de aducção eixo-x



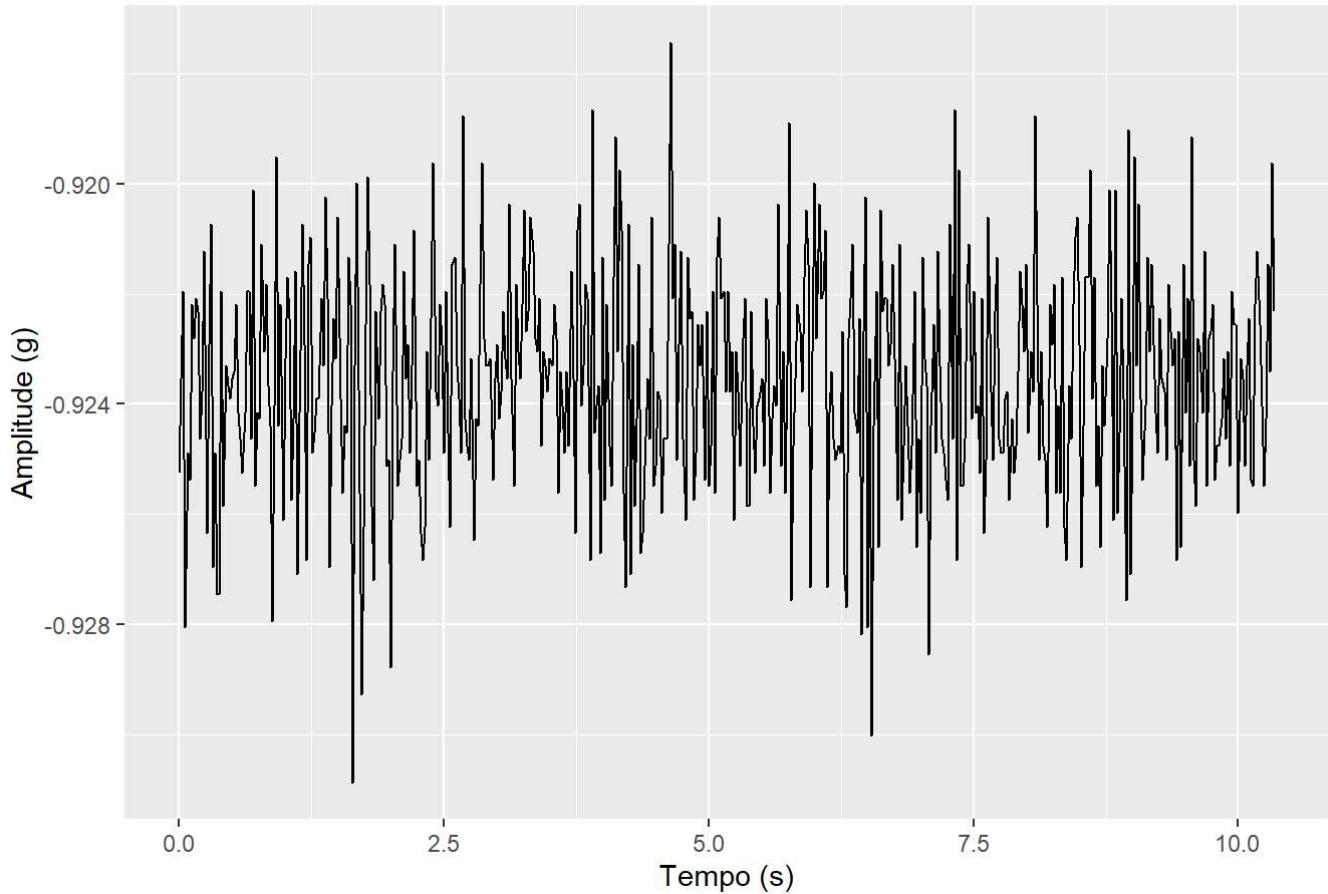
```
qplot(df_flx_csv$`[Time]`, df_flx_csv$`[A2.Y]`, geom=c("line")) +  
  labs(x='Tempo (s)', y='Amplitude (g)',  
    title='Acelerômetro 2 de flexão eixo-y')
```

Acelerômetro 2 de flexão eixo-y



```
qplot(df_rot_csv$`[Time]`, df_rot_csv$`[A2.Z]`, geom=c("line")) +  
  labs(x='Tempo (s)', y='Amplitude (g)',  
    title='Acelerômetro 2 de rotação eixo-z')
```

Acelerômetro 2 de rotação eixo-z



Por fim, a última rodada de leituras e plots são utilizando os arquivos .edf lidos abaixo:

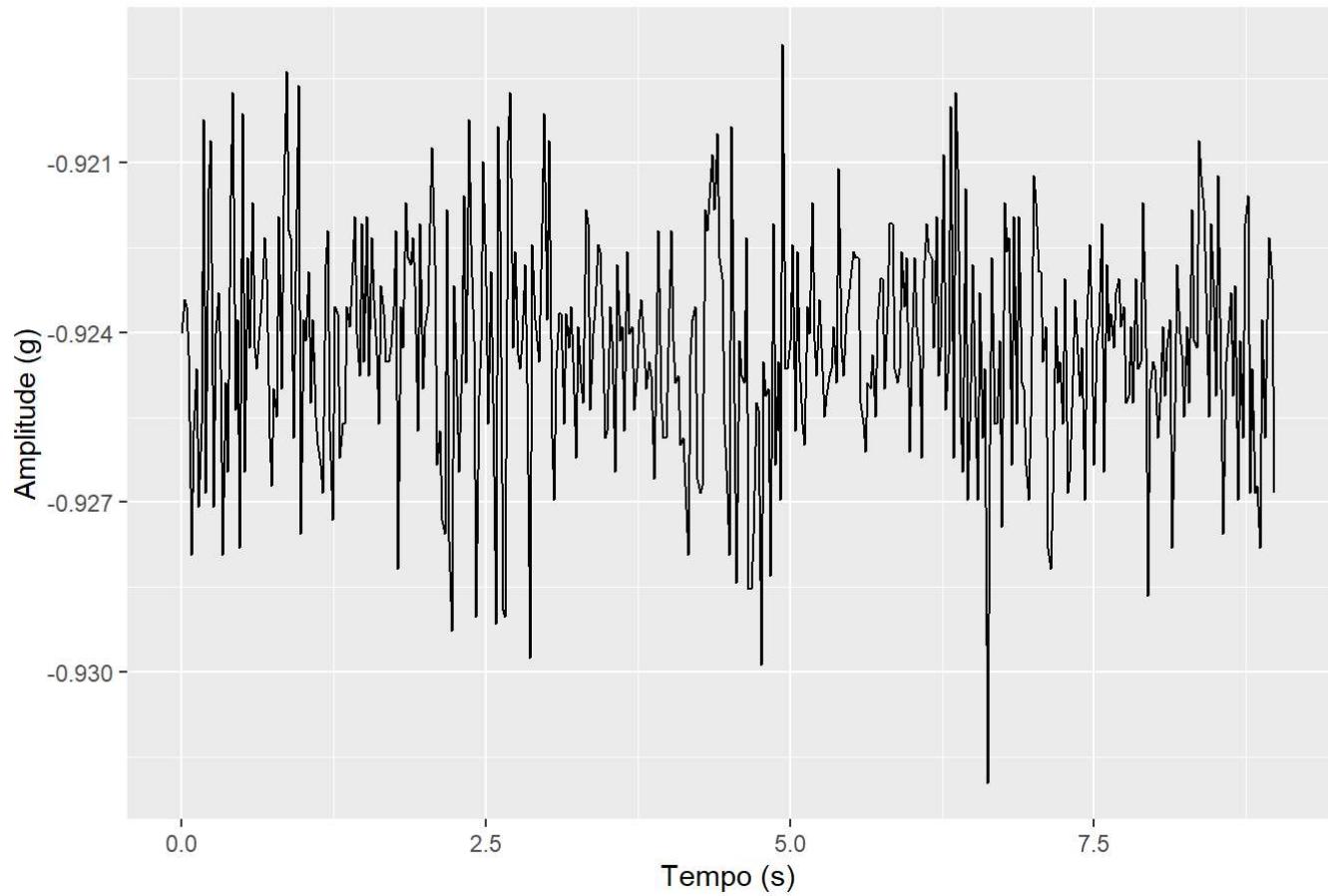
```
library(edfReader)

# Loading
header <- readEdfHeader("Dados/Aduccao.edf")
df_add_edf <- readEdfSignals(header)
header <- readEdfHeader("Dados/Flexao.edf")
df_flx_edf <- readEdfSignals(header)
header <- readEdfHeader("Dados/Rotacao.edf")
df_rot_edf <- readEdfSignals(header)

t_add = seq(from=0, to=df_add_edf$`Ch: 17-:A2.Z`$sRate*df_add_edf$`Ch: 17-:A2.Z`$recordedPeriod-1, by=1) * 1/df_add_edf$`Ch: 17-:A2.Z`$sRate
t_flx = seq(from=0, to=df_flx_edf$`Ch: 16-:A2.Y`$sRate*df_flx_edf$`Ch: 16-:A2.Y`$recordedPeriod-1, by=1) * 1/df_flx_edf$`Ch: 16-:A2.Y`$sRate
t_rot = seq(from=0, to=df_rot_edf$`Ch: 15-:A2.X`$sRate*df_rot_edf$`Ch: 15-:A2.X`$recordedPeriod-1, by=1) * 1/df_rot_edf$`Ch: 15-:A2.X`$sRate

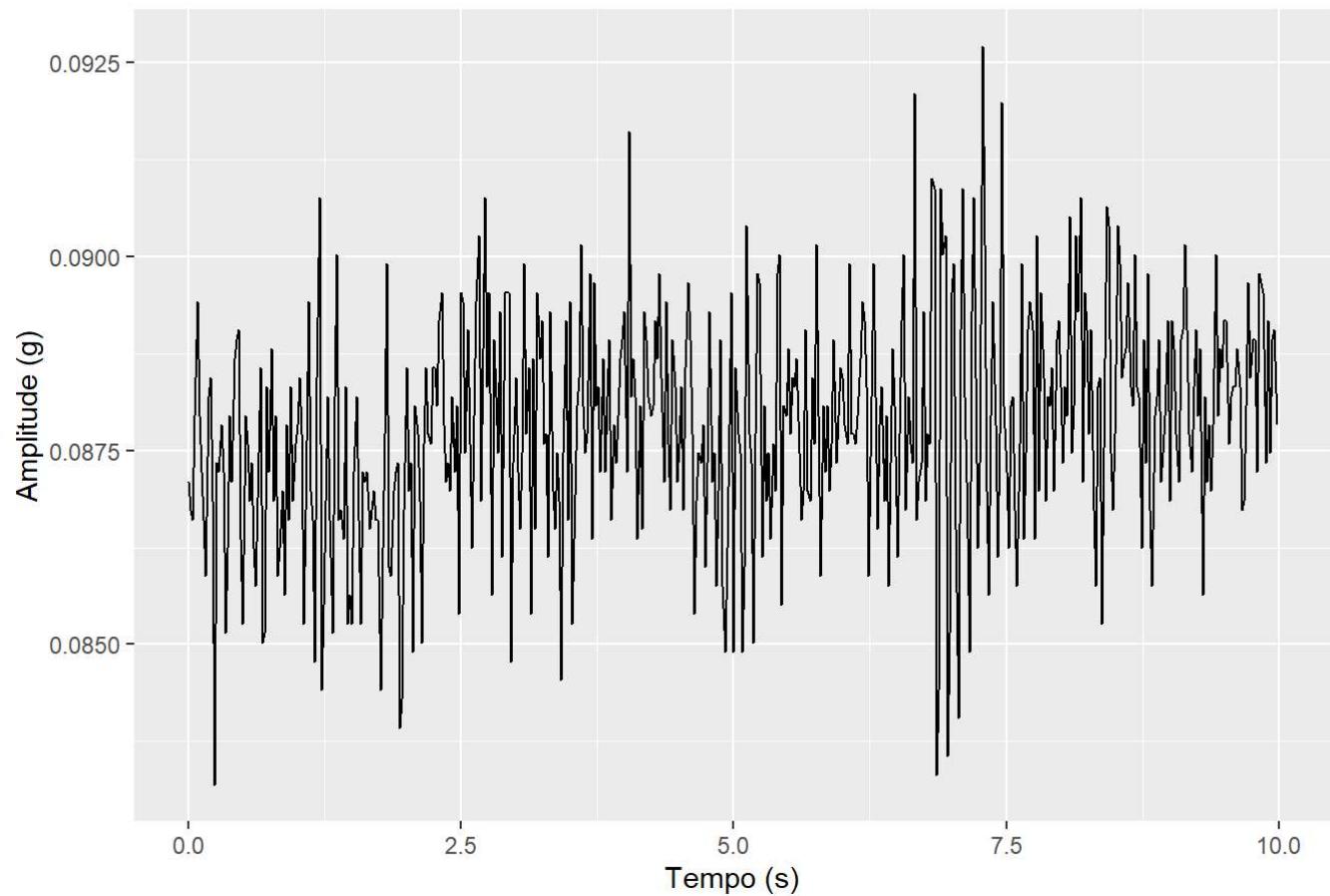
# plot
qplot(t_add, df_add_edf$`Ch: 17-:A2.Z`$signal, geom=c("line")) +
  labs(x='Tempo (s)', y='Amplitude (g)',
       title='Acelerômetro 2 de aducção eixo-z, canal 17')
```

Acelerômetro 2 de aducção eixo-z, canal 17



```
qplot(t_flx, df_flx_edf$`Ch: 16-A2.Y`$signal, geom=c("line")) +  
  labs(x='Tempo (s)', y='Amplitude (g)',  
    title='Acelerômetro 2 de flexão eixo-y, canal 16')
```

Acelerômetro 2 de flexão eixo-y, canal 16



```
qplot(t_rot, df_rot_edf$`Ch: 15-A2.X`$signal, geom=c("line")) +  
  labs(x='Tempo (s)', y='Amplitude (g)',  
    title='Acelerômetro 2 de rotação eixo-x, canal 15')
```

Acelerômetro 2 de rotação eixo-x, canal 15

