

Universidade Federal de Uberlândia

## Lista do Módulo 5

Levy Gabriel da Silva Galvão

Uberlândia  
2021

# Módulo 5

## Exercício 1

Considerando os dados disponíveis no arquivo “MODULO5.xlsx”, que podem ser visualizados abaixo, responda às questões:

```
library(openxlsx) # biblioteca para abrir arquivo do excel
library(dygraphs)

fname <- 'Dados/MODULO5.xlsx' # nome do arquivo, incluindo o caminho (observe que o caminho d
epende de onde você salvou o arquivo em seu computador)

df_init <- readWorkbook(fname,
                        sheet = 1,
                        detectDates = TRUE) # faz a leitura do arquivo de dados

str(df_init) # visualização das variáveis disponíveis no data frame (df)
```

```
## 'data.frame': 349380 obs. of 4 variables:
## $ time : num 0 0.001 0.002 0.003 0.004 0.005 0.006 0.007 0.008 0.009 ...
## $ chan.1: num 25.4 15.8 15.9 16.1 23.1 ...
## $ chan.2: num 3.31 20.67 37.91 26.08 49.12 ...
## $ pulse : num 0 0 0 0 0 0 0 0 0 0 ...
```

```
df <- data.frame(time=df_init$time, ch1 =df_init$chan.1, ch2 = df_init$chan.2 )
rm(df_init)
#dygraph(df)%>%dyRangeSelector() # comment to avoid RAM consumption peak
```

### Questões:

- a. Qual a relação sinal-ruído do período entre 0 e 14 segundos. Faça a estimativa para os canais ch1 e ch2. Dica: Utilize o valor RMS para o cálculo.

```
rms <- function(signal)
{
  acc<-sum(signal^2)
  acc<-acc/length(signal)
  acc<-sqrt(acc)
  return(acc)
}
idx_t <- which((df$time>=0) & (df$time<=14))
ch1_rms <- rms(df$ch1[idx_t])
ch2_rms <- rms(df$ch2[idx_t])
# considera-se a amplitude do ruído normalizada
snr_ch1 <- 20*log10(ch1_rms)
snr_ch2 <- 20*log10(ch2_rms)
```

```
print(snr_ch1) # snr channel 1
```

```
## [1] 19.21906
```

```
print(snr_ch2) # snr channel 2
```

```
## [1] 21.2486
```

- b. Faça uma comparação entre os trechos de ruído 1 (entre 0 e 14 s) e ruído 2 (entre 18 e 43 s). Faça a estimativa para os canais ch1 e ch2. Dica: Utilize o valor RMS para o cálculo e apresente os resultados em dB.

```
idx_t <- which((df$time>=18) & (df$time<=43))
ch1_rms <- rms(df$ch1[idx_t])
ch2_rms <- rms(df$ch2[idx_t])
# considera-se a amplitude do ruído normalizada
snr_ch1 <- 20*log10(ch1_rms)
snr_ch2 <- 20*log10(ch2_rms)

print(snr_ch1) # snr channel 1
```

```
## [1] 22.89182
```

```
print(snr_ch2) # snr channel 2
```

```
## [1] 25.15818
```

Ao comparar o trecho entre 0 e 14s, em que, visualmente, o ruído predomina, observa-se uma baixa relação sinal-ruído. Enquanto que ao analisar o trecho entre 18 e 43s que possui conteúdo rico em sinal, observa-se uma relação sinal-ruído maior. Assim pode-se concluir que o primeiro trecho é de ruído e o segundo de sinal.

- c. Para cada um dos canais (ch1 e ch2) estimate o valor da tendência linear constante. O cálculo deve ser feito de forma independente para os trechos de ruído 1 e 2, e para o trecho de sinal (de 45 a 334 s).

```
dt <- df$time[2]-df$time[1]

noise_tini <- 0
noise_tend <- 45-dt
signal_tini <- 45
signal_tend <- 334

noise_tidx <- which((df$time>=noise_tini) & (df$time<=noise_tend))
signal_tidx <- which((df$time>=signal_tini) & (df$time<=signal_tend))

# computation of constant linear tendency: noise snippet
clt_noise_ch1 <- mean(df$ch1[noise_tidx])
clt_noise_ch2 <- mean(df$ch2[noise_tidx])
print(clt_noise_ch1) # channel 1
```

```
## [1] -0.04765182
```

```
print(clt_noise_ch2) # channel 2
```

```
## [1] -0.01348574
```

```
# computantion of constant linear tendency: signal snippet
clt_signal_ch1 <- mean(df$ch1[signal_tidx])
clt_signal_ch2 <- mean(df$ch2[signal_tidx])
print(clt_signal_ch1) # channel 1
```

```
## [1] -0.02411887
```

```
print(clt_signal_ch2) # channel 2
```

```
## [1] -0.1327982
```

d. Existe alguma descontinuidade nos sinais dos canais 1 e 2? Justifique a sua resposta por meio do uso de um boxplot. Considere em sua análise o trecho de 0 a 40 s.

```
library(plotly)
```

```
## Carregando pacotes exigidos: ggplot2
```

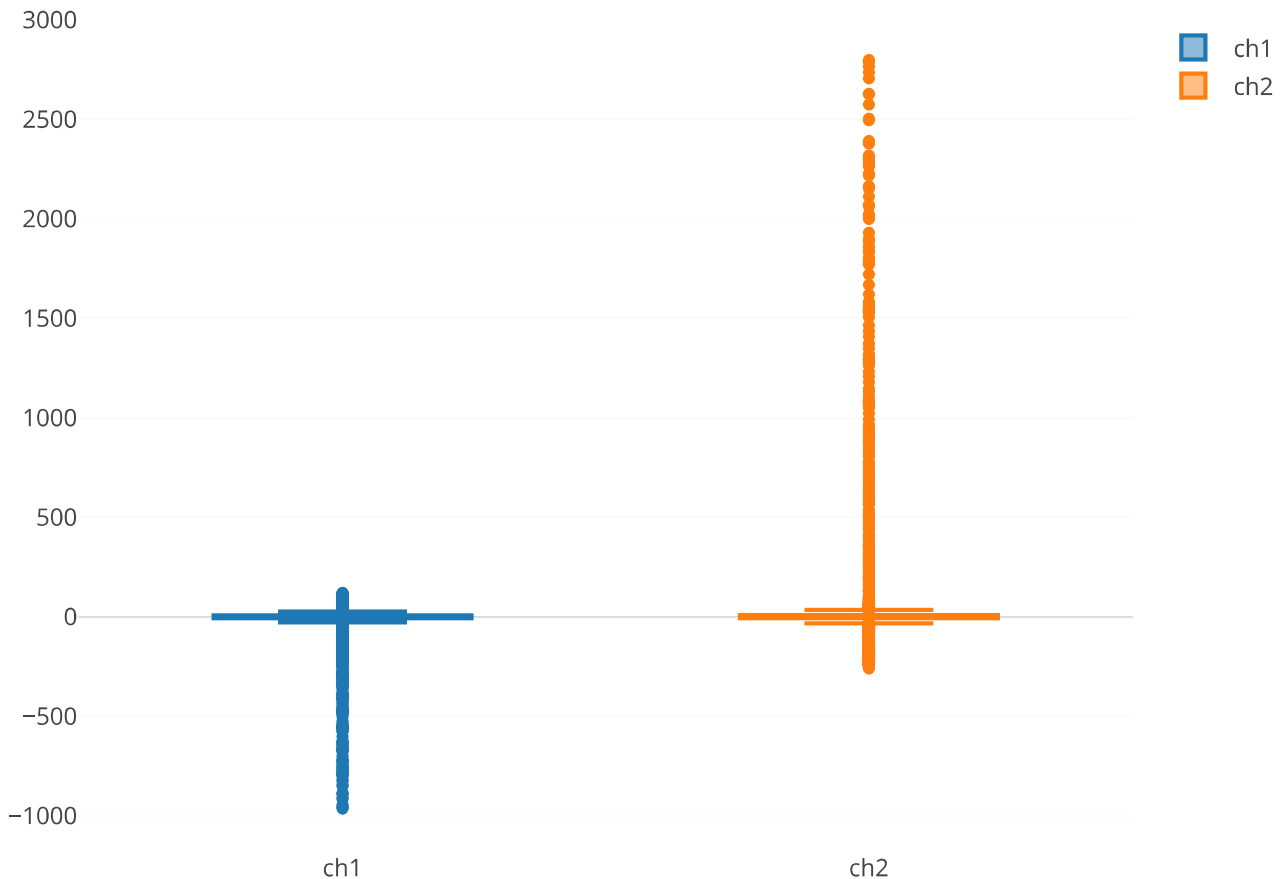
```
##
## Attaching package: 'plotly'
```

```
## The following object is masked from 'package:ggplot2':
##
##     last_plot
```

```
## The following object is masked from 'package:stats':
##
##     filter
```

```
## The following object is masked from 'package:graphics':
##
##     layout
```

```
idx_t <- which((df$time>=0) & (df$time<=40))
p <- plot_ly(y = df$ch1[idx_t], type = "box", name = "ch1") %>%
  add_trace(y = df$ch2[idx_t], name="ch2")
p
```



Por meio do boxplot de saída permite-se observar descontinuidades nos sinais de ambos os canais, majoritariamente devido a “bursting” que ocorreu ao redor do instante de 14 segundos.

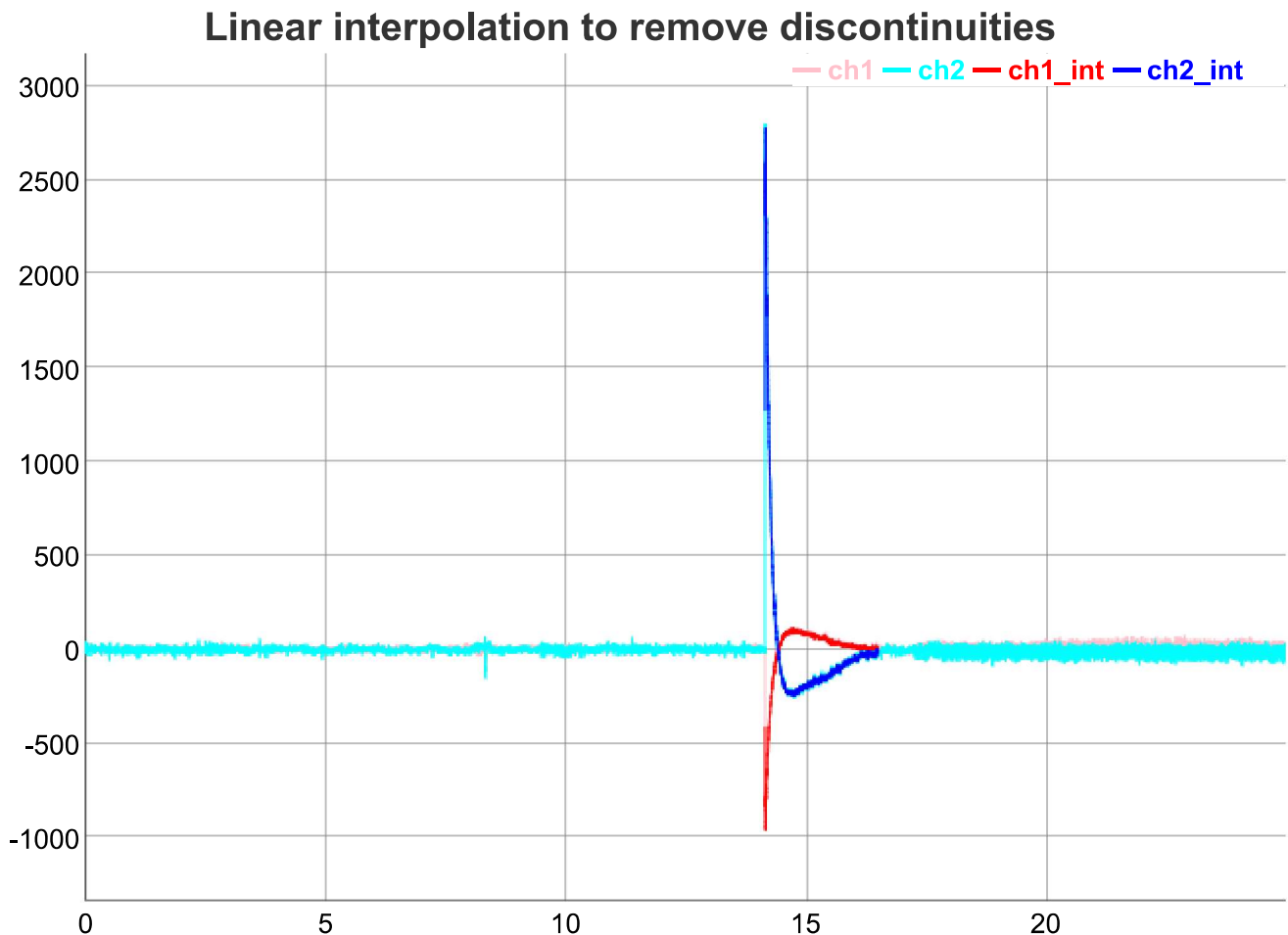
- e. Caso exista alguma descontinuidade no sinal (ver letra d) faça a remoção da mesma por meio do método de interpolação linear. Gere um gráfico que ilustre o trecho do sinal com e sem a descontinuidade.

```
linearInterp <- function(x0,y0,x1,y1,x){ # function linear interpolation to remove discontinuities
  y <- y0 + (y1-y0)*(x-x0)/(x1-x0)
  return(y)
}
dt <- df$time[2]-df$time[1]
tdisc <- which((df$time>=14.157) & (df$time<=16.516)) # range that discontinuities occurred
x0 <- df$time[tdisc-1]
x1 <- df$time[tdisc+1]

# channel 1 efforts
y0_ch1 <- df$ch1[tdisc-1]
y1_ch1 <- df$ch1[tdisc+1]
y_ch1 <- linearInterp(x0,y0_ch1,x1,y1_ch1,df$time[tdisc])
df$ch1_int <- NA
df$ch1_int[tdisc] <- y_ch1

# channel 2 efforts
y0_ch2 <- df$ch2[tdisc-1]
y1_ch2 <- df$ch2[tdisc+1]
y_ch2 <- linearInterp(x0,y0_ch2,x1,y1_ch2,df$time[tdisc])
df$ch2_int <- NA
df$ch2_int[tdisc] <- y_ch2
```

```
# plots
dygraph(df[(df$time<=25),], main = "Linear interpolation to remove discontinuities") %>%
  dySeries(name="ch1", drawPoints=FALSE, color = "pink" ) %>%
  dySeries(name="ch2", drawPoints=FALSE, color = "cyan") %>%
  dySeries(name="ch1_int", drawPoints=FALSE, color = "red") %>%
  dySeries(name="ch2_int", drawPoints=FALSE, color = "blue")
```



Apesar dos esforços da interpolação linear, a descontinuidade não é completamente suprimida, pois seu comportamento se perdura como uma exponencial em decaimento, portanto ao tentar interpolar os pontos vizinhos, apesar dos da esquerda apresentarem uma tendência normal para o trecho com ruído, o trecho da direita não possui, portanto prejudicando a interpolação.

- f. Os sinais apresentados nos canais 1 e 2 são de tempo e valores contínuos. Essa afirmação é verdadeira ou falsa? Justifique.

Considerando estes sinais como sinais digitalizados, estes não podem ser considerados de valores ou tempos contínuos. Na verdade os sinais de ambos os canais são denotados como de tempo discreto e de amplitude quantizada, pois estes estão armazenados em hardware de natureza digital.

- g. Desafio: leia o conceito de variável independente e identicamente distribuída (iid) disponível em [link](#). Utilize a função **diffsign.test** disponível na biblioteca **spgs** para verificar se os trechos de sinais compreendidos entre 0 e 8 s, entre 20 e 40 s, e entre 45 e 334 s, são iid. Justifique a razão de um trecho ser ou não iid.

```
library(spgs)

idx_t1 <- which((df$time>=0) & (df$time<=8)) # range 1 - 0~8s
idx_t2 <- which((df$time>=20) & (df$time<=40)) # range 2 - 20~40s
```

```
idx_t3 <- which((df$time>=45) & (df$time<334)) # range 3 - 45~334s
```

```
print(diffsign.test(df$ch1[idx_t1])) # range 1 - ch1
```

```
##  
## Difference-sign test of independence  
##  
## data: df$ch1[idx_t1]  
## D = -0.11617, p-value = 0.9075
```

```
print(diffsign.test(df$ch2[idx_t1])) # range 1 - ch2
```

```
##  
## Difference-sign test of independence  
##  
## data: df$ch2[idx_t1]  
## D = -4.3759, p-value = 1.209e-05
```

```
print(diffsign.test(df$ch1[idx_t2])) # range 2 - ch1
```

```
##  
## Difference-sign test of independence  
##  
## data: df$ch1[idx_t2]  
## D = 11.659, p-value < 2.2e-16
```

```
print(diffsign.test(df$ch2[idx_t2])) # range 2 - ch2
```

```
##  
## Difference-sign test of independence  
##  
## data: df$ch2[idx_t2]  
## D = -13.325, p-value < 2.2e-16
```

```
print(diffsign.test(df$ch1[idx_t3])) # range 3 - ch1
```

```
##  
## Difference-sign test of independence  
##  
## data: df$ch1[idx_t3]  
## D = 50.503, p-value < 2.2e-16
```

```
print(diffsign.test(df$ch2[idx_t3])) # range 3 - ch2
```

```
##  
## Difference-sign test of independence  
##
```

```
## data: df$ch2[idx_t3]
## D = 73.333, p-value < 2.2e-16
```

Para um trecho ser iid, o processo precisa ocorrer independentemente, porém com distribuição de probabilidade semelhante das suas variáveis aleatórias.

No primeiro e segundo trecho, apesar de apresentarem ruído, estes, visualmente, possuem suas diferenças em seus desvios padrões. Já o terceiro trecho é completamente diferente e possui seus valores regidos pelo sinal de análise.

Apesar do segundo trecho apresentar características visuais semelhantes ao primeiro, em termos da análise de seu valor p, observa-se uma independência maior para o primeiro trecho, mostrando assim uma maior aleatoriedade. Como esperado para o terceiro trecho, este possui um baixo valor p, indicando que este trecho segue uma distribuição de probabilidade idêntica.