

Universidade Federal de Uberlândia

## Lista do Módulo 6

Levy Gabriel da Silva Galvão

Uberlândia  
2021

# Módulo 6

## Exercício 1

Utilize o software Audacity para gravar cinco sinais de voz com o comando “sim” e outros cinco sinais com o comando “não”. Faça a rotulação de cada segmento dos comandos de voz, salvando-os em arquivos distintos. No total, dez arquivos devem ser gerados.

Os arquivos podem ser carregados, como:

```
library(htmltools)
library(dygraphs)
library(tuneR)
library(dygraphs)

audio_path <- "Data/"

# file lists
sim_fname <- list("sim1.wav", "sim2.wav", "sim3.wav", "sim4.wav", "sim5.wav")
nao_fname <- list("nao1.wav", "nao2.wav", "nao3.wav", "nao4.wav", "nao5.wav")

# merge file name with full path
sim_fpath <- as.list(paste(audio_path, sim_fname, sep = ""))
nao_fpath <- as.list(paste(audio_path, nao_fname, sep = ""))

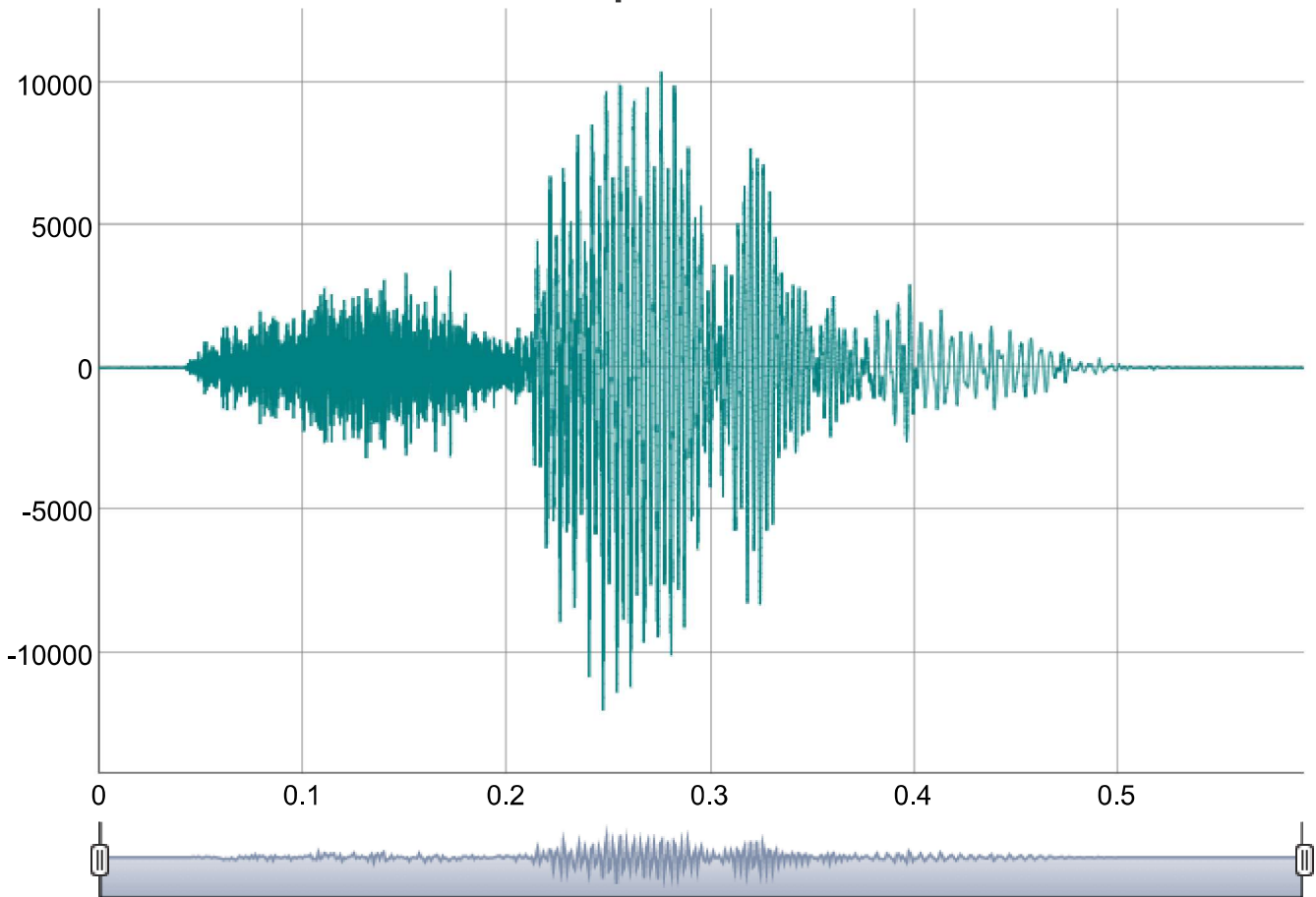
# append names of all files
fnames <- append(sim_fname, nao_fname)
fpaths <- append(sim_fpath, nao_fpath)

data <- list()
n = length(fpaths)
fs <- 44100 # Hz

for(i in c(1:n)){data <- append(data, list(readWave(fpaths[[i]])@left))} # save audio data
names(data) <- fnames

# sample
signal <- data[['sim1.wav']]
m <- length(signal)
dt <- 1/fs
t <- seq(from=0, to=(m-1)*dt, by=dt)
data.frame(time=t, signal=signal) %>%
  dygraph(main='Audio sample - sim1.wav') %>%
  dyRangeSelector()
```

Audio sample - sim1.wav



## Exercício 2

Para cada sinal calcule as estatísticas: média, variância, coeficiente de assimetria e curtose. Organize os resultados em uma tabela (no R) e estime a variância de cada estatística.

- O comando “sim” é estacionário? Justifique.
- O comando “sim” é ergódico? Justifique.
- O comando “não” é estacionário? Justifique.
- O comando “não” é ergódico? Justifique.

```
library(moments)
```

```
## Warning: package 'moments' was built under R version 4.1.1
```

```
stats_ <- function(x)
{
  mean_      <- mean(x)
  variance_  <- var(x)
  skewness_  <- skewness(x)
  kurtosis_  <- kurtosis(x)
  return(list(mean=mean_,
              variance=variance_,
              skewness=skewness_,
```

```

        kurtosis=kurtosis_))
}

data_stats <- data.frame(matrix(ncol = 5, nrow = 10))
colnames(data_stats) <- c('index', 'mean', 'variance', 'skewness', 'kurtosis')
data_stats$index <- as.vector(unlist(fnames))
for(i in c(1:n)){data_stats[i,2:5] <- unlist(stats_(data[[i]]), use.names=FALSE)} # save data
stats

var_row_sim <- data.frame(
  index      = 'var_sim',
  mean       = var(data_stats[1:5,2]),
  variance   = var(data_stats[1:5,3]),
  skewness   = var(data_stats[1:5,4]),
  kurtosis   = var(data_stats[1:5,5]),
  stringsAsFactors = FALSE
)
var_row_nao <- data.frame(
  index      = 'var_nao',
  mean       = var(data_stats[6:10,2]),
  variance   = var(data_stats[6:10,3]),
  skewness   = var(data_stats[6:10,4]),
  kurtosis   = var(data_stats[6:10,5]),
  stringsAsFactors = FALSE
)

data_stats <- rbind(data_stats, var_row_sim)
data_stats <- rbind(data_stats, var_row_nao)
table_ <- as.table(sapply(data_stats,unlist))

# sample
print(table_)

```

```

##   index      mean      variance      skewness
## A sim1.wav -0.0021063112745098  4380828.02339566 -0.0621121303774941
## B sim2.wav -0.069091796875      4262992.11738757  0.101718033142271
## C sim3.wav -0.027047821969697    4175178.1377117   0.00491349537251782
## D sim4.wav  0.00686383928571429   8720886.26709949 -0.184357362037322
## E sim5.wav -0.0201505335365854   4589733.44006843 -0.140982710829439
## F nao1.wav  0.00785405585106383   5624372.22239603 -0.12538285659868
## G nao2.wav -0.03876953125        4230302.85146819  0.00952939208059118
## H nao3.wav  0.0164763621794872    5484047.67434163 -0.110362375330586
## I nao4.wav -0.00668174342105263   4707639.3033735  -0.100740636749412
## J nao5.wav -0.03521728515625     2976327.07115178  0.125007310794438
## K var_sim  0.0008687371619199    3841247891083.69  0.0130800822155535
## L var_nao  0.000621584720506305  1153837940601.3  0.0114025836154641
##   kurtosis
## A 9.15116290441023
## B 3.69267537094301
## C 6.59846673468415
## D 7.04212323790177
## E 8.67960849144599
## F 5.5132324704636
## G 4.32523434766431
## H 5.33373466778619
## I 4.98268994112049
## J 4.93373676063935

```

```
## K 4.636151111942
## L 0.208303636672859
```

Considerando que para um processo estacionário a sua média não varia com o tempo e para um processo ergódico a média entre janelas do sinal é semelhante, com variância da estatística tendendo a zero.

Ao observar a linha da tabela da variância das estatísticas, pode-se observar que a variância da média para ambos os sinais “sim” e “não” é baixa, implicando assim em um processo estacionário para ambos.

Porém ao avaliar a ergodicidade, nota-se que a variância das estatísticas de variância, coeficiente de assimetria e curtose não tende a zero. Isso mostra que um conceito amplo de ergodicidade não pode ser atribuído a esses sinais. Mas uma vez que a variância da média tende a zero, ainda pode-se dizer que o processo dos sinais de “sim” e “não” podem ser considerados ergódicos para a média.

## Exercício 3

Utilize o valor máximo da correlação cruzada para realizar a comparação entre comandos de voz. Anote estes valores em uma tabela, conforme padrão abaixo. Utilize um lag de 10.000 amostras no cálculo.

- Baseado nos resultados apresentados na tabela, como você poderia desenvolver um sistema de reconhecimento automático de voz, que realiza a distinção entre os comandos “sim” e “não”?

```
cctable <- matrix(nrow=10,ncol=10)
names <- c('SIM-1','SIM-2','SIM-3','SIM-4','SIM-5','NÃO-1','NÃO-2','NÃO-3','NÃO-4','NÃO-5')
colnames(cctable) <- names
rownames(cctable) <- names

print(cctable) # initialize table
```

##	SIM-1	SIM-2	SIM-3	SIM-4	SIM-5	NÃO-1	NÃO-2	NÃO-3	NÃO-4	NÃO-5
## SIM-1	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
## SIM-2	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
## SIM-3	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
## SIM-4	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
## SIM-5	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
## NÃO-1	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
## NÃO-2	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
## NÃO-3	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
## NÃO-4	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
## NÃO-5	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA

```
for(i in c(1:n))
{
  for(j in c(1:n))
  {
    cctable[i,j] <- ccf(data[[i]], data[[j]], lag.max=10000, pl=FALSE)$acf %>% max()
  }
}

print(cctable) # max cross-correlation table
```

##	SIM-1	SIM-2	SIM-3	SIM-4	SIM-5	NÃO-1	NÃO-2
## SIM-1	1.00000000	0.6021928	0.8187084	0.5288100	0.6863245	0.6135171	0.6035822
## SIM-2	0.60219279	1.00000000	0.8294615	0.5796261	0.7259943	0.6602159	0.7008873
## SIM-3	0.81870843	0.8294615	1.00000000	0.7048958	0.8501412	0.6565547	0.6231247
## SIM-4	0.52881003	0.5796261	0.7048958	1.00000000	0.7468441	0.6967719	0.6405836
## SIM-5	0.68632452	0.7259943	0.8501412	0.7468441	1.00000000	0.6679746	0.6184094
## NÃO-1	0.61351707	0.6602159	0.6565547	0.6967719	0.6679746	1.00000000	0.6454052
## NÃO-2	0.60358223	0.7008873	0.6231247	0.6405836	0.6184094	0.6454052	1.00000000
## NÃO-3	0.67252251	0.6775701	0.6773707	0.7012880	0.6522093	0.8193042	0.6875380
## NÃO-4	0.62581498	0.5780334	0.6515413	0.6725198	0.6551921	0.7396599	0.6676438
## NÃO-5	0.09965576	0.1264772	0.1419436	0.1164425	0.1221640	0.2358013	0.2164641
##	NÃO-3	NÃO-4	NÃO-5				
## SIM-1	0.6725225	0.6258150	0.09965576				
## SIM-2	0.6775701	0.5780334	0.12647720				
## SIM-3	0.6773707	0.6515413	0.14194355				
## SIM-4	0.7012880	0.6725198	0.11644254				
## SIM-5	0.6522093	0.6551921	0.12216399				
## NÃO-1	0.8193042	0.7396599	0.23580131				
## NÃO-2	0.6875380	0.6676438	0.21646410				
## NÃO-3	1.00000000	0.7881993	0.20337231				
## NÃO-4	0.7881993	1.00000000	0.17830106				
## NÃO-5	0.2033723	0.1783011	1.00000000				

Para desenvolver um sistema de reconhecimento automático de voz, como para distinguir entre “sim” e “não” pronunciado, inicialmente seriam identificados padrões inerentes a cada tipo de comando. Após isso, esse vetor de padrões seria correlacionado com cada sinal a ser reconhecido por meio de uma correlação cruzada e, aquele padrão que tiver maior correlação com o sinal de interesse, implicará que o sinal de interesse será um “sim” ou “não”.

## Exercício 4

Considerando a avaliação de três grupos neurônios, com as características abaixo:

- E1: média de disparo = 53 ms; desvio padrão = 50 ms
- E2: média de disparo = 100 ms; desvio padrão = 35 ms
- E3: média de disparo = 53 ms; desvio padrão = 15 ms
- Dica: o intervalo entre disparos é um processo pontual com distribuição gaussina (Utilize a função `rnorm` do R para gerar os tempos de disparo)

```
fs <- 10000 # Hz
dt <- 1/fs # s
t_end <- 10 # s
t <- seq(0, t_end, by=dt)
n <- length(t)

E1.firing <- rep(0,n)
E1.mean <- 53e-3
E1.std <- 50e-3
rand_ <- rnorm(100, mean=E1.mean, sd=E1.std)
E1.range1 <- round(rand_[cumsum(rand_)<1],4)
rand_ <- rnorm(100, mean=E1.mean, sd=E1.std)
E1.range2 <- round(rand_[cumsum(rand_)<2],4)
```

```

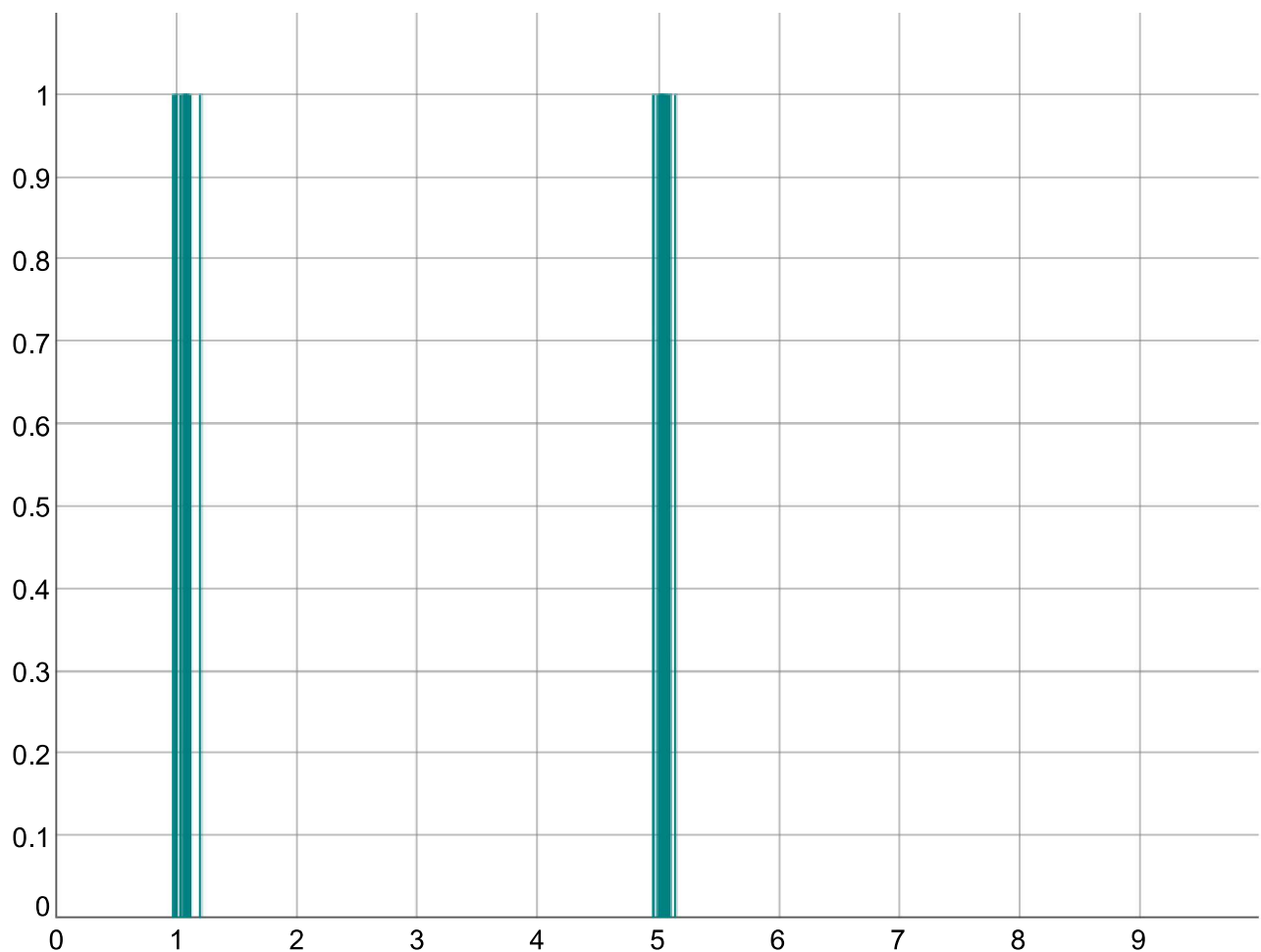
E1.firing[t %in% (1+E1.range1)] = 1
E1.firing[t %in% (5+E1.range2)] = 1

E2.firing <- rep(0,n)
E2.mean <- 100e-3
E2.std <- 35e-3
rand_ <- rnorm(100, mean=E2.mean, sd=E2.std)
E2.range1 <- round(rand_[cumsum(rand_)<2],4)
rand_ <- rnorm(100, mean=E2.mean, sd=E2.std)
E2.range2 <- round(rand_[cumsum(rand_)<2],4)
E2.firing[t %in% (2+E2.range1)] = 1
E2.firing[t %in% (7+E2.range2)] = 1

E3.firing <- rep(0,n)
E3.mean <- 53e-3
E3.std <- 15e-3
rand_ <- rnorm(100, mean=E3.mean, sd=E3.std)
E3.range1 <- round(rand_[cumsum(rand_)<2],4)
rand_ <- rnorm(100, mean=E3.mean, sd=E3.std)
E3.range2 <- round(rand_[cumsum(rand_)<2],4)
E3.firing[t %in% (1+E3.range1)] = 1
E3.firing[t %in% (9+E3.range2)] = 1

# plots
data.frame(time=t, data=E1.firing) %>% dygraph

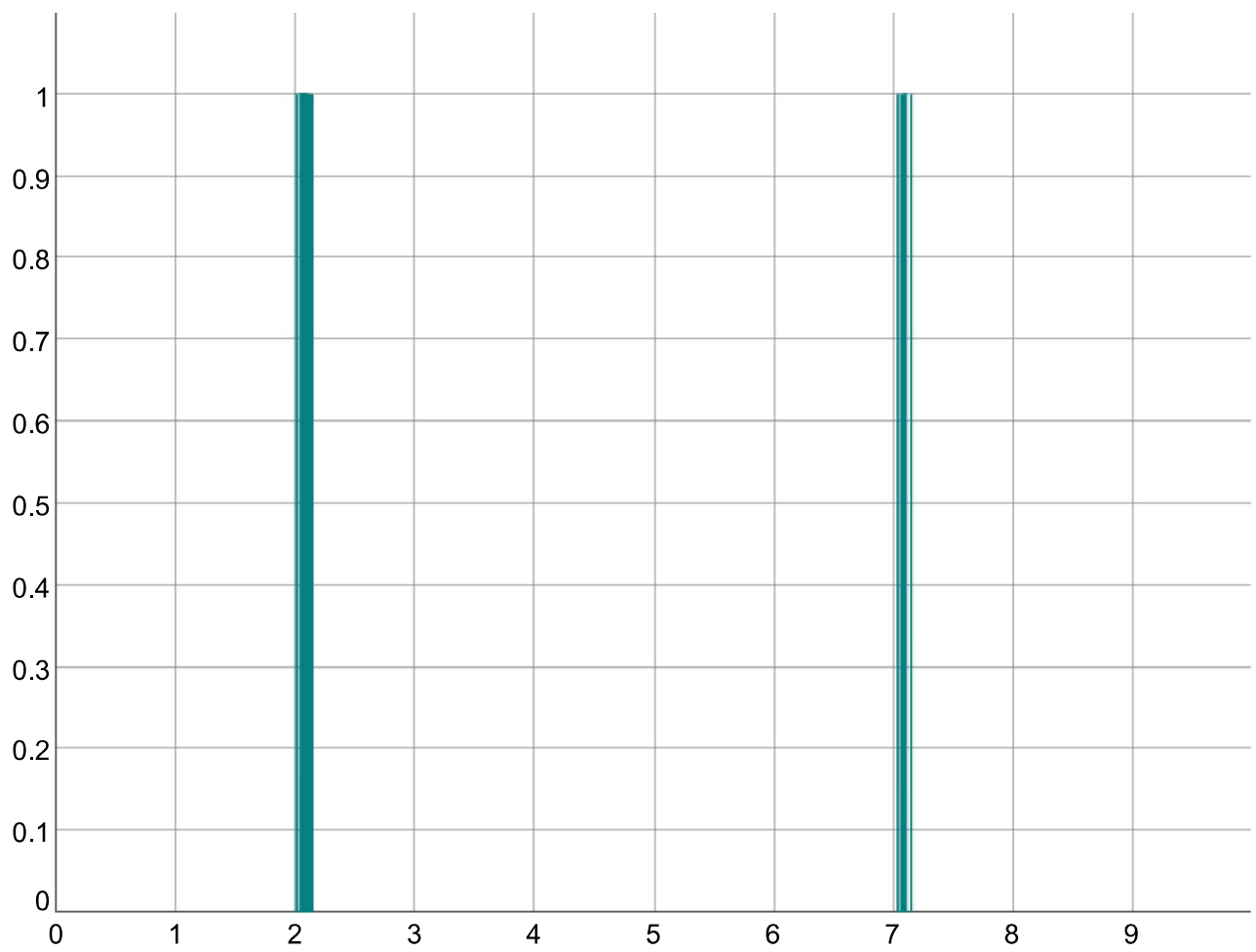
```



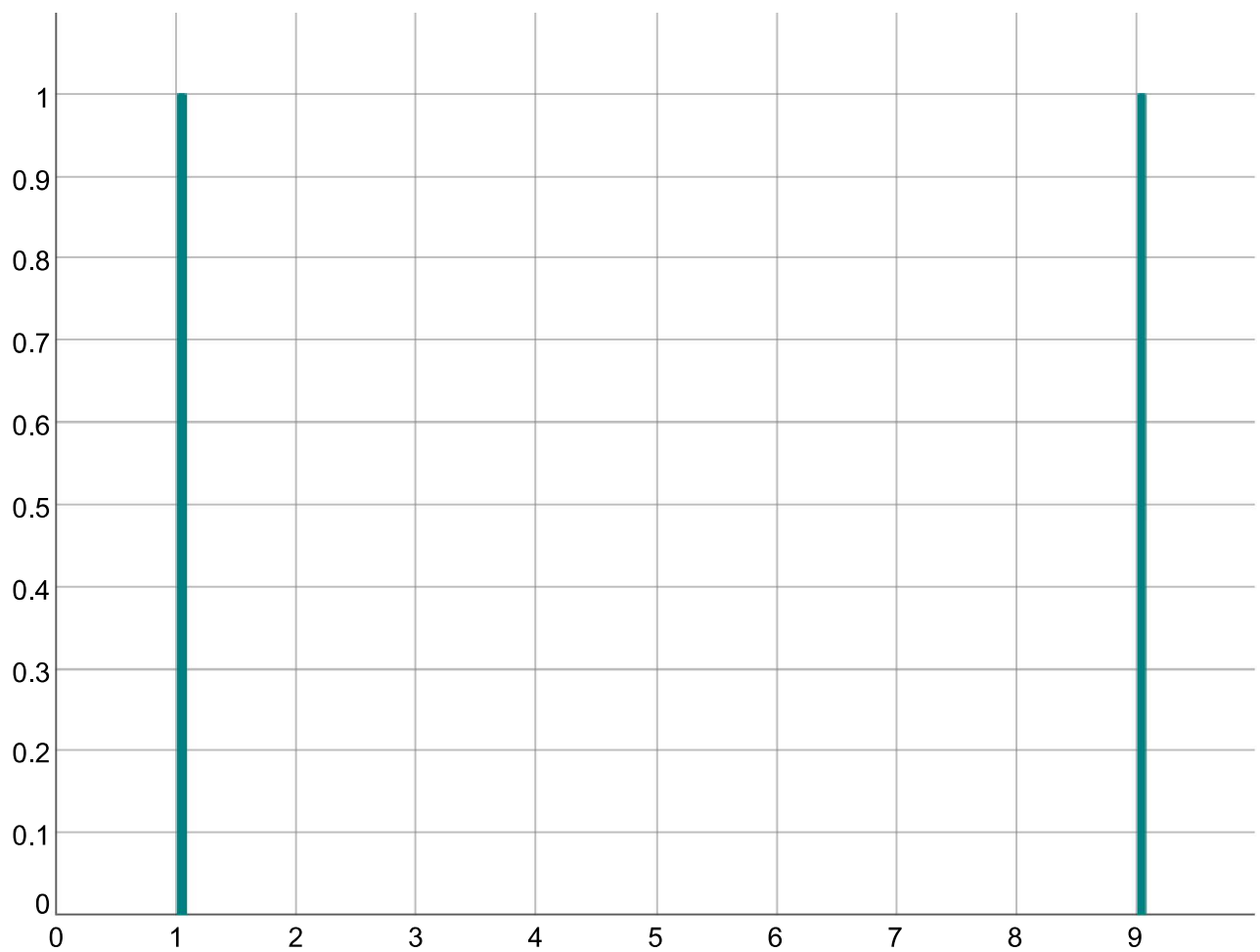
```

data.frame(time=t, data=E2.firing) %>% dygraph

```



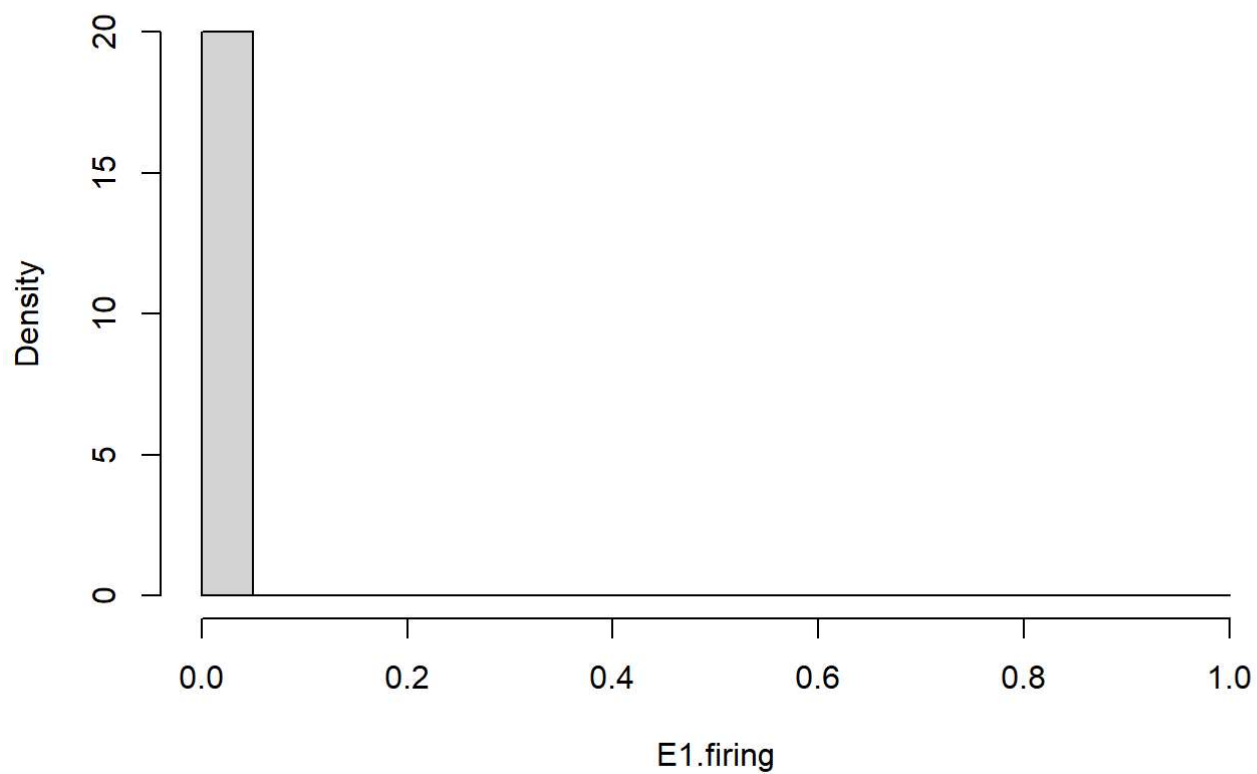
```
data.frame(time=t, data=E3.firing) %>% dygraph
```





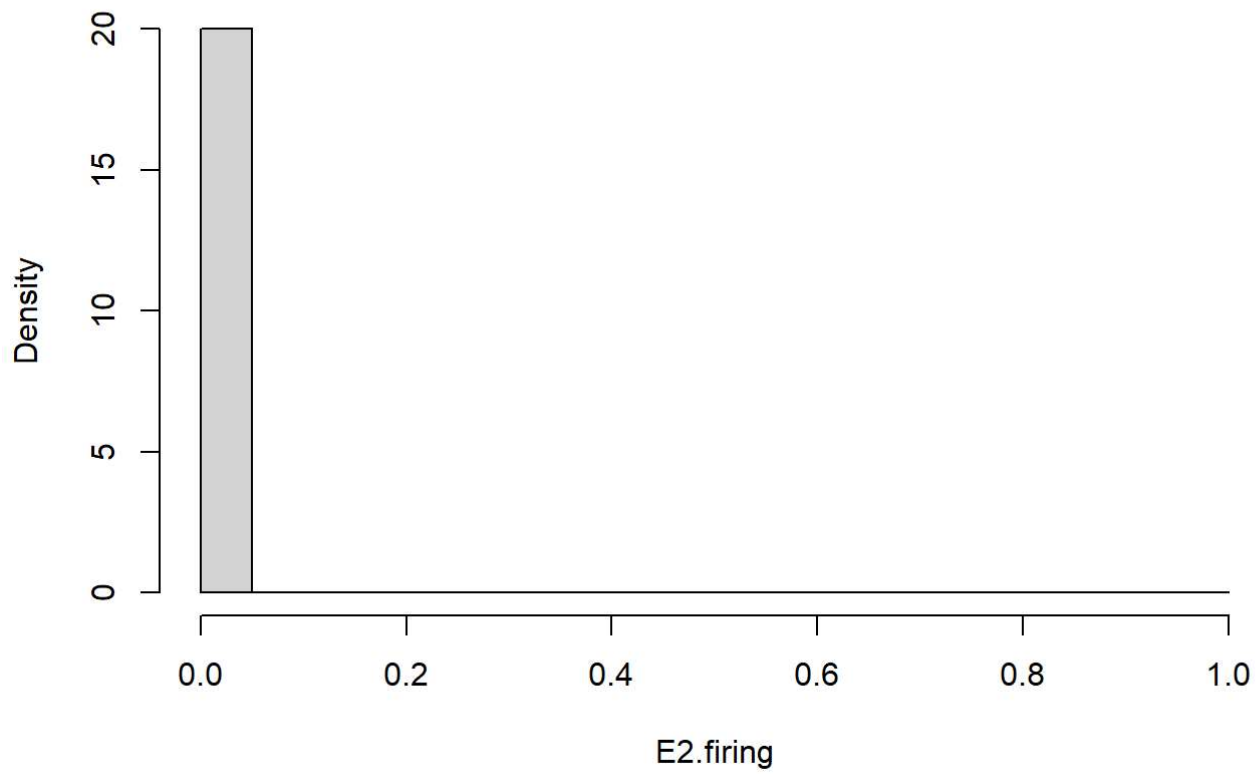
```
# histograms  
E1.hist <- hist(E1.firing, probability=TRUE)
```

**Histogram of E1.firing**



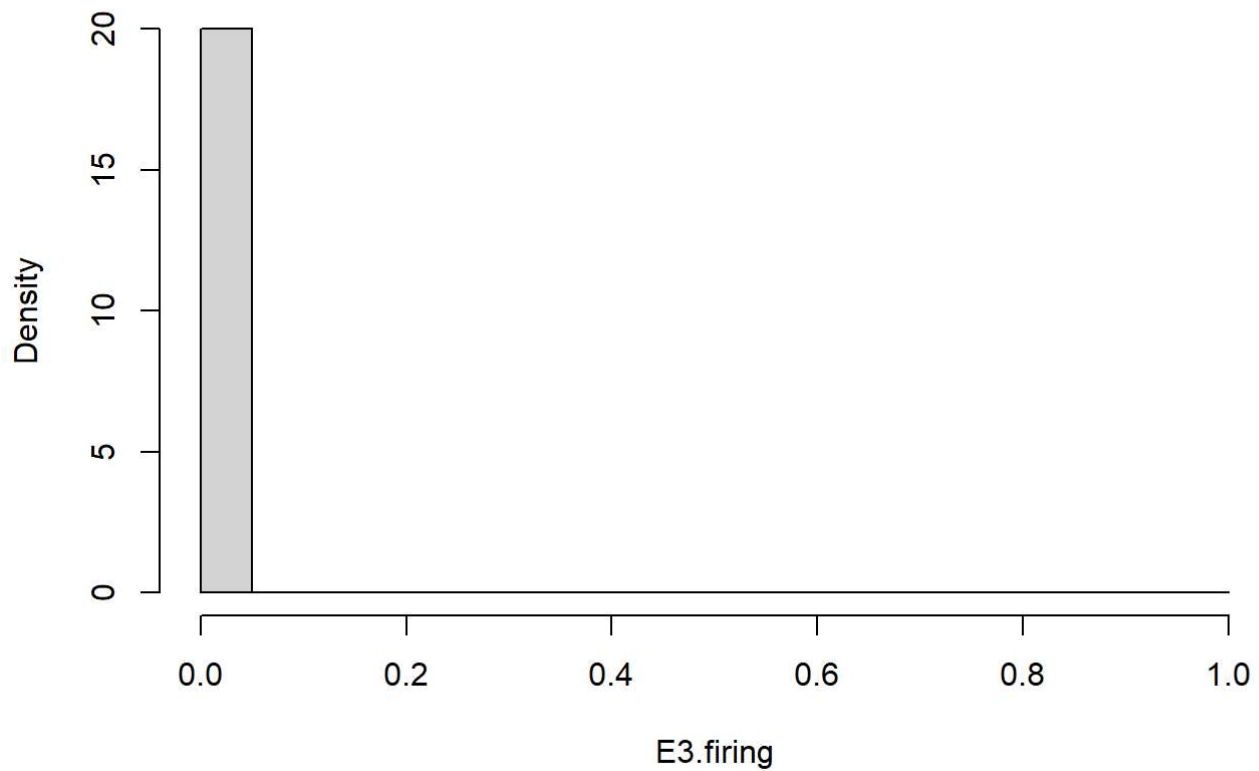
```
E2.hist <- hist(E2.firing, probability=TRUE)
```

### Histogram of E2.firing



```
E3.hist <- hist(E3.firing, probability=TRUE)
```

### Histogram of E3.firing



Devido à baixa densidade de disparos, acaba que a visualização do histograma é tendenciosa para o estado de repouso, porém pode-se observar os valores armazenados no histograma para retirar as conclusões de que ocorreu o disparo:

```
# histogram counts  
print(E1.hist$counts)
```

```
## [1] 99972    0    0    0    0    0    0    0    0    0    0    0  
## [13]    0    0    0    0    0    0    0    29
```

```
print(E2.hist$counts)
```

```
## [1] 99975    0    0    0    0    0    0    0    0    0    0    0  
## [13]    0    0    0    0    0    0    0    26
```

```
print(E3.hist$counts)
```

```
## [1] 99953    0    0    0    0    0    0    0    0    0    0    0  
## [13]    0    0    0    0    0    0    0    48
```