

Periféricos y Dispositivos de Interfaz Humana

Práctica 5 - Sonido



**UNIVERSIDAD
DE GRANADA**

Luis Miguel López López

La práctica la he realizado usando Rstudio en sistema Windows 11 pero los sonidos los he creado en ubuntu con espeak.

Una vez instalado R y Rstudio, además de las diferentes bibliotecas que se nos indica en el seminario (tuneR y seewave), procedemos a crear los sonidos con:

espeak "Luis Miguel" -w nombre.wav

espeak "López López" -w apellidos.wav

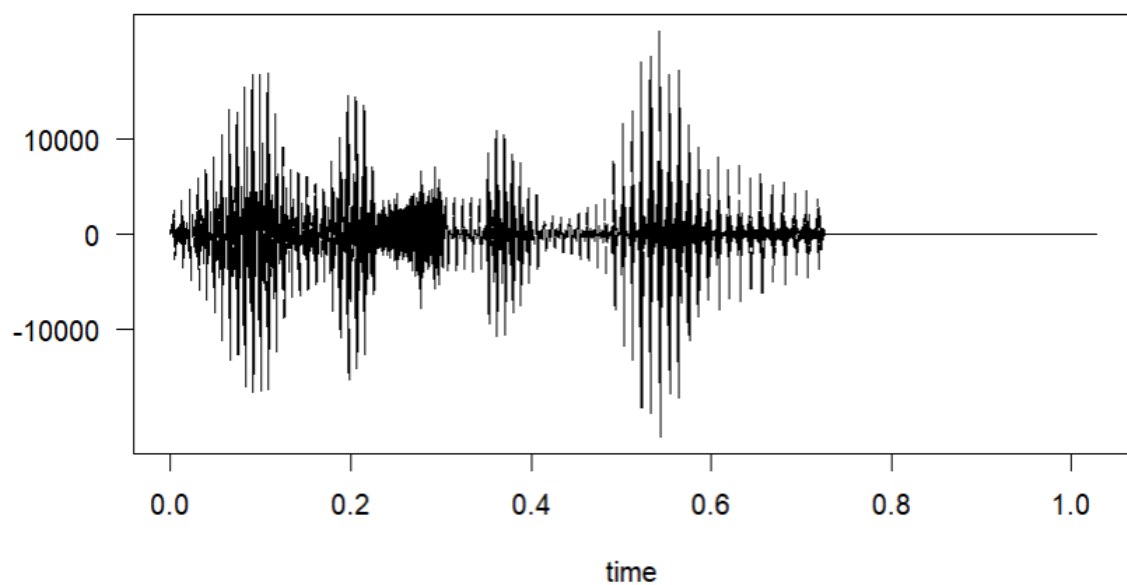
Cargamos librerías y los dos archivos de sonido:

```
1 library(tuneR)
2 library(seewave)
3 library(audio)
4
5 nombre <- readWave('nombre.wav')
6 apellidos <- readWave('apellidos.wav')
7
```

Con las siguientes órdenes obtenemos las muestras y el dibujo de la onda:

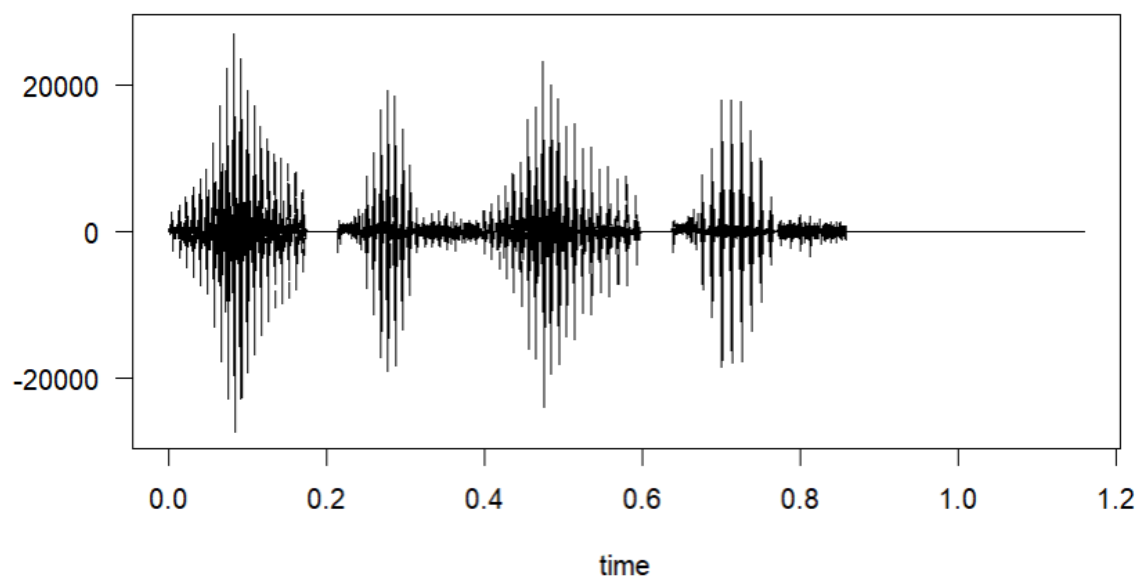
```
length(nombre@left)
plot(extractWave(nombre, from = 1, to = 22644))
```

Con round obtenemos el 22644 que usamos en plot, el cual nos muestra la siguiente gráfica:



Con apellidos tenemos un tratamiento similar:

```
length(apellidos@left)
plot(extractwave(apellidos, from = 1, to = 25551))
|
```



Para obtener la información de las cabeceras empleamos str:

```
str(nombre)
str(apellidos)
```

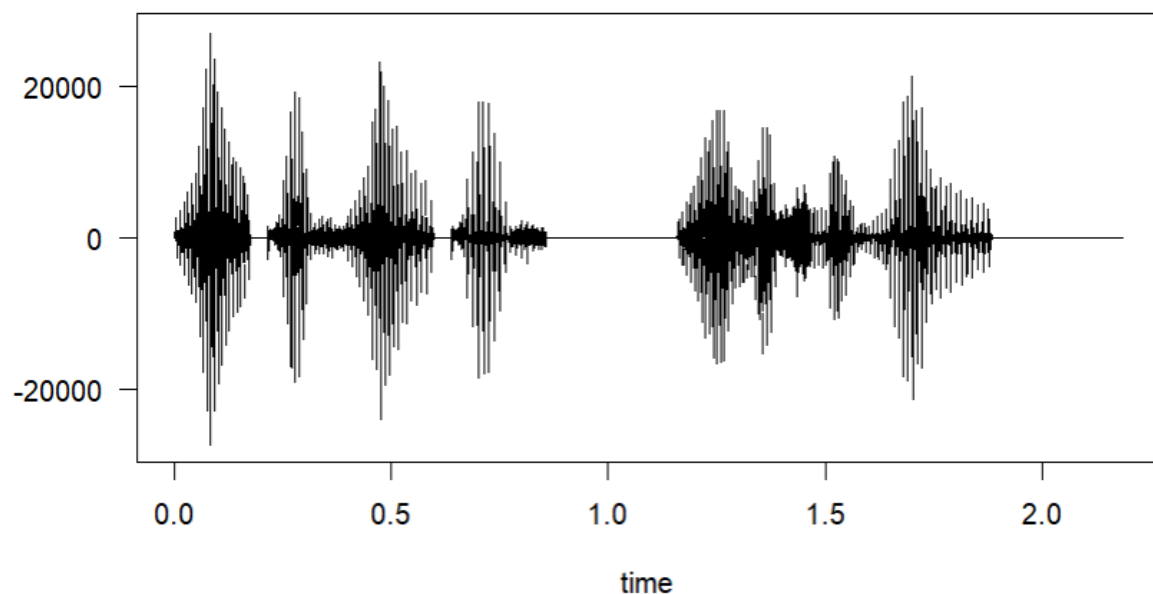
Que nos da como resultado, primero de nombre y luego de apellidos lo siguiente:

```
Formal class 'Wave' [package "tuneR"] with 6 slots
..@ left      : int [1:22644] 30 63 100 141 184 239 303 370 429 472 ...
..@ right     : num(0)
..@ stereo    : logi FALSE
..@ samp.rate: int 22050
..@ bit       : int 16
..@ pcm       : logi TRUE
> str(apellidos)
Formal class 'Wave' [package "tuneR"] with 6 slots
..@ left      : int [1:25551] 26 55 87 124 163 212 272 336 390 428 ...
..@ right     : num(0)
..@ stereo    : logi FALSE
..@ samp.rate: int 22050
..@ bit       : int 16
..@ pcm       : logi TRUE
> |
```

Ahora procedemos a unir las dos partes y a mostrar la onda resultante con lo siguiente:

```
completo <- pastew(nombre, apellidos,output="wave")
length(completo@left)
plot(extractWave(completo, from = 1, to = 48195))
```

y obtenemos como onda:



Aplicamos filtro y guardamos resultado:

```
mezcla <- bwfilter(completo, f=22050, channel=1, n=1, from=10000/2, to=20000/2, bandpass=TRUE,
                  , listen=FALSE, output="Wave")

writeWave(mezcla, file.path("mezcla.wav"))
```

Volvemos a cargar el archivo del nombre completo y procedemos a darle eco y darle la vuelta, finalmente guardaremos también ese sonido:

```
tmp <- readWave('completo.wav')

eco <- echo(tmp, f=22050, amp=c(0.8, 0.4, 0.2), delay=c(1, 2, 3), output="Wave")
eco@left <- 10000*eco@left
alreves <- revw(eco, output="Wave")
writeWave(mezcla, file.path("alreves.wav"))
```

Finalmente todos los archivos de sonido, tanto iniciales como resultantes, se encuentran subidos en github junto a este documento y el script de R.