

PDIH - UGR

Práctica 5 - Ángel Gómez Ferrer

Para preparar el entorno solamente he instalado R y RStudio.

Ejercicio 1 :

Leer dos ficheros de sonido (WAV o MP3) de unos pocos segundos de duración cada uno. En el primero debe escucharse el nombre de la persona que realiza la práctica. En el segundo debe escucharse el apellido.

Añadimos las librerías necesarias y las instalamos:

```
1 library(tuneR)
2 library(seewave)
3 library(audio)
4
```

En mi caso utilizo windows y he creado los audios con: <https://speechgen.io>

Tras obtener los audios los añadimos:

```
4
5 # Añadimos los archivos de audio
6
7 Nombre <- readWave('nombre.wav')
8 Apellidos <- readWave('apellidos.wav')
9
10 # calculamos la duración
```

Ejercicio 2 :

Dibujar la forma de onda de ambos sonidos.

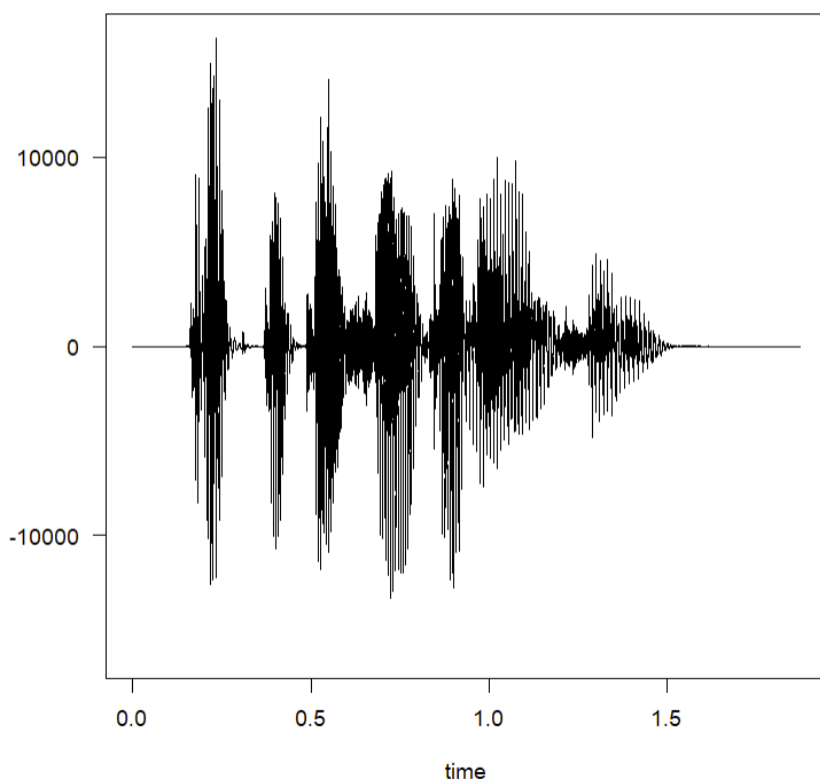
Primero calculamos la longitud exacta de la muestra y utilizando este dato la mostramos:

```

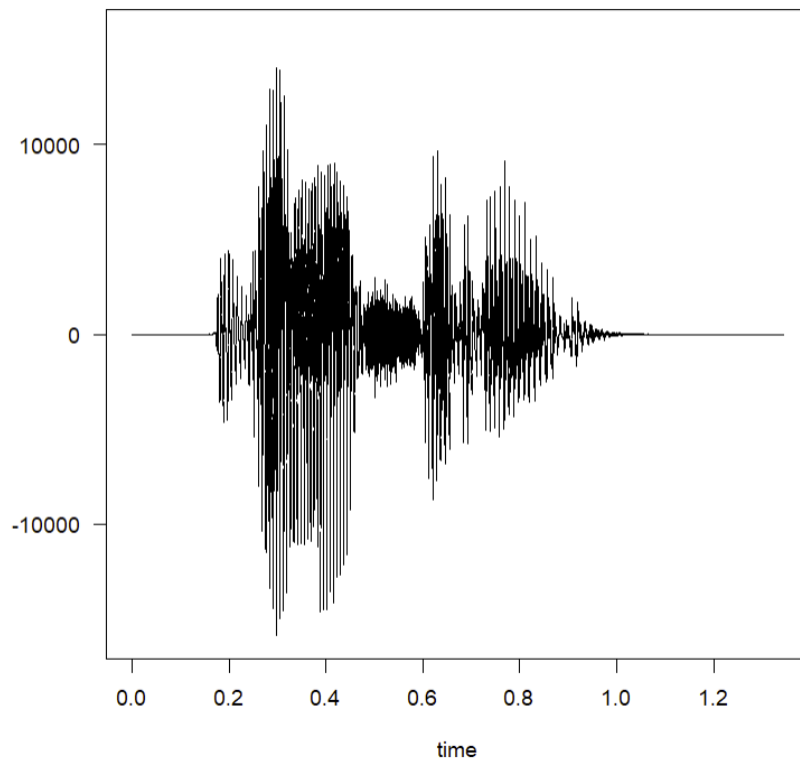
9
10 # Calculamos la duración
11 round(length(Nombre@left) / Nombre@samp.rate, 3)
12
13 # Mostrar forma de onda hasta el final (usamos duración para la última muestra)
14 plot(extractWave(Nombre, from = 1, to = 1873000))
15
16 # Calculamos la duración
17 round(length(Apellidos@left) / Apellidos@samp.rate, 3)
18
19 # Mostrar forma de onda hasta el final
20 plot(extractWave(Apellidos, from = 1, to = 1345000))
21

```

Forma de onda de Nombre:



Forma de onda de Apellidos:



Ejercicio 3 :

Obtener la información de las cabeceras de ambos sonidos.

```
22 # Cabecera audio Nombre  
23 Nombre  
24  
25 # Cabecera audio Apellidos  
26 Apellidos  
27
```

```

> # Cabecera audio Nombre
> Nombre

Wave Object
  Number of Samples: 89903
  Duration (seconds): 1.87
  Samplingrate (Hertz): 48000
  Channels (Mono/Stereo): Mono
  PCM (integer format): TRUE
  Bit (8/16/24/32/64): 16

```

```

>
> # Cabecera audio Apellidos
> Apellidos

```

```

Wave Object
  Number of Samples: 64559
  Duration (seconds): 1.34
  Samplingrate (Hertz): 48000
  Channels (Mono/Stereo): Mono
  PCM (integer format): TRUE
  Bit (8/16/24/32/64): 16

```

Ejercicio 4 :

Unir ambos sonidos en uno nuevo.

Para unirlos usaremos la función `pasteW`:

```

28 # Unimos nombre y apellidos
29
30 NombreApellidos <-| pasteW(Apellidos, Nombre , output="Wave")
31

```

Ejercicio 5 :

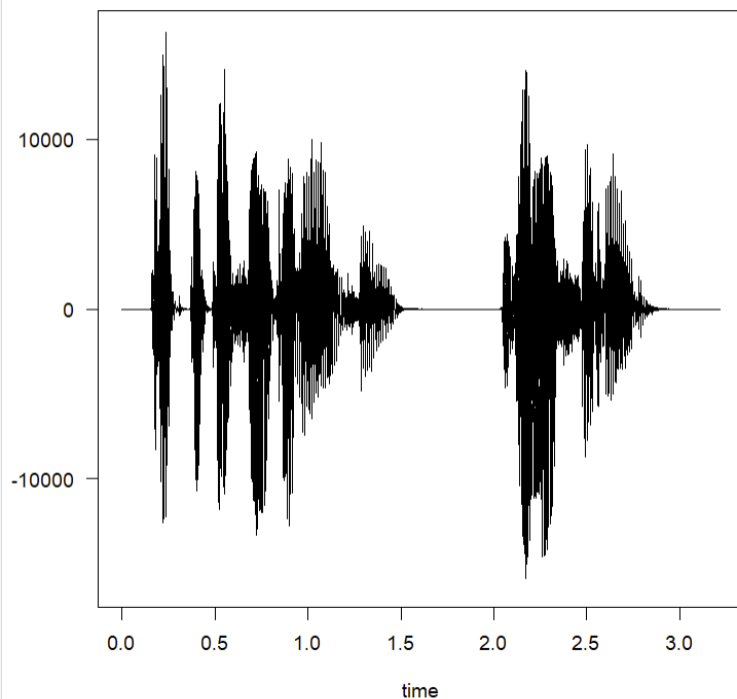
Dibujar la forma de onda de la señal resultante.

```

32 # Forma de onda resultante
33
34 round(length(NombreApellidos@left) / NombreApellidos@samp.rate, 3)
35 plot(extractWave(NombreApellidos, from = 1, to = 3218000))
36

```

Forma de onda resultante:



Ejercicio 6 :

Pasarle un filtro de frecuencia para eliminar las frecuencias entre 10000Hz y 20000Hz

Utilizaremos la función `bwfilter`:

```
37 # Eliminamos las frecuencias entre 10000Hz y 20000Hz
38 NombreApellidosFiltrado <- bwfilter(NombreApellidos, 48000, channel = 1, n = 1, from = 10000,
39                                     to = 20000, bandpass = TRUE, listen = FALSE, output = "Wave")
40
```

Ejercicio 7 :

Almacenar la señal obtenida como un fichero WAV denominado “mezcla.wav”.

Guardamos utilizando `writeWave`:

```
44 # Guardamos el resultado como mezcla.wav
45 writeWave(NombreApellidosFiltrado, file.path("mezcla.wav"))
46
```

Ejercicio 8 :

Cargar un nuevo archivo de sonido, aplicarle eco y a continuación darle la vuelta al sonido. Almacenar la señal obtenida como un fichero WAV denominado “alreves.wav”.

Añadimos el nuevo archivo en mi caso he usado el que antes generamos (nombreApellidos y lo guarde con la función previamente mencionada writeWave)

Para este ejercicio primero usaremos la función readWave para añadir el audio, tras esto le aplicaremos el eco con la función echo, lo pondremos al revés con revw y lo guardaremos con writeWave:

```
50 # Añadimo eco:
51
52 nuevoAudioEco <- echo(nuevoAudio, f=22050, amp=c(0.8,0.4,0.2), delay=c(1,2,3),
53                       output="Wave")
54 nuevoAudioEco@left <- 10000 * nuevoAudioEco@left
55
56
57 # Al reves:
58 nuevoAudioEcoAlreves <- revw(nuevoAudioEco, output="Wave")
59
60 # Visualizamos la forma de onda
61 #plot(extractWave(nuevoAudioAlreves, from = 1, to = 3218000))
62
63 # Guardamos
64 writeWave(nuevoAudioEcoAlreves, file.path("alreves.wav"))
65
```

Nota:

Todos los resultados audios resultantes y utilizados del ejercicio están en el repositorio.