

Práctica 5. Experimentación con el sistema de salida de sonido

Realizado por Rubén Martín

Al igual que con el seminario, siguiendo el guión del seminario y centrándome en los comandos dados, he tenido suficiente material para completar esta práctica.

- Comprobación del funcionamiento del script

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.

Código:

```
nombre <- readWave('C:/Users/gilru/OneDrive/Escritorio/Uni/4/2 Cuatri/PDIH/P5/nombre.wav')
nombre
apellido <- readWave('C:/Users/gilru/OneDrive/Escritorio/Uni/4/2 Cuatri/PDIH/P5/apellido.wav')
apellido
```

Salida:

```
> nombre <- readWave('C:/Users/gilru/OneDrive/Escritorio/Uni/4/2 Cuatri/PDIH/P5/nombre.wav')
> nombre

wave object
  Number of Samples: 23685
  Duration (seconds): 0.49
  Samplingrate (Hertz): 48000
  Channels (Mono/Stereo): Mono
  PCM (integer format): TRUE
  Bit (8/16/24/32/64): 16

> apellido <- readWave('C:/Users/gilru/OneDrive/Escritorio/Uni/4/2 Cuatri/PDIH/P5/apellido.wav')
> apellido

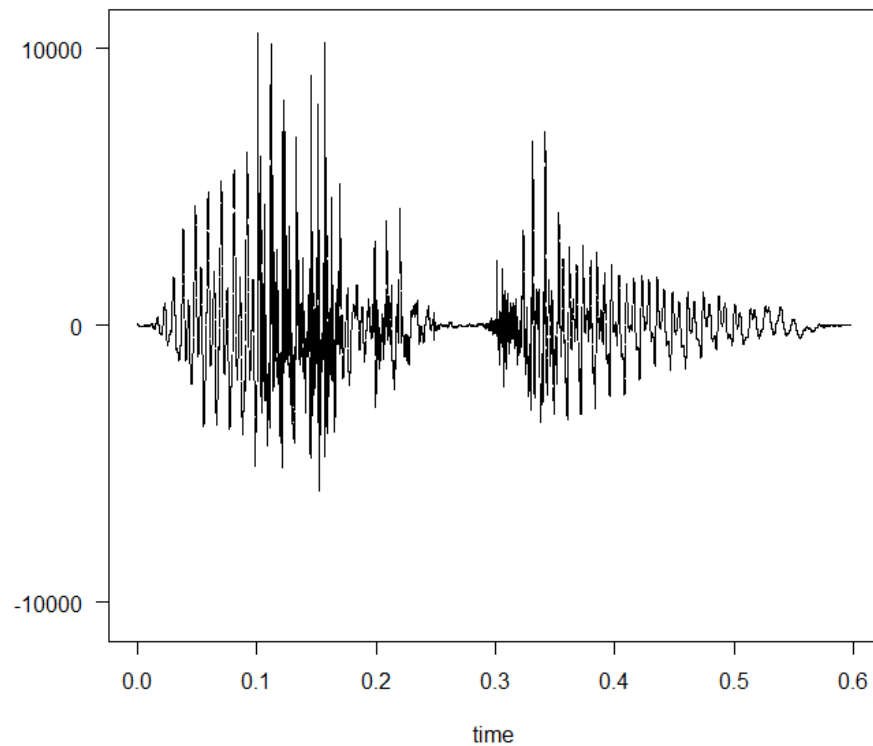
wave object
  Number of Samples: 28700
  Duration (seconds): 0.6
  Samplingrate (Hertz): 48000
  Channels (Mono/Stereo): Mono
  PCM (integer format): TRUE
  Bit (8/16/24/32/64): 16
```

2. Dibujar la forma de onda de ambos sonidos

Código:

```
plot(extractWave(nombre, from=1, to = 23685))
plot(extractWave(apellido, from=1, to = 28700))
```

Salida:



En este caso, no entiendo porque se representan ambas ondas juntas, pero el script realiza una lectura correcta.

3. Obtener la información de las cabeceras de ambos sonidos.

Código:

```
str(nombre)
str(apellido)
```

Salida:

```
> proc(extractwave(apellido, from = 2, to = 28700))
> str(nombre)
Formal class 'wave' [package "tuner"] with 6 slots
..@ left      : int [1:23685] 0 -1 -1 -2 -1 -1 -2 -2 -2 -2 ...
..@ right     : num(0)
..@ stereo    : logi FALSE
..@ samp.rate: int 48000
..@ bit       : int 16
..@ pcm       : logi TRUE
> str(apellido)
Formal class 'wave' [package "tuner"] with 6 slots
..@ left      : int [1:28700] 32 47 49 43 37 37 42 46 47 45 ...
..@ right     : num(0)
..@ stereo    : logi FALSE
..@ samp.rate: int 48000
..@ bit       : int 16
..@ pcm       : logi TRUE
```

4. Unir ambos sonidos en uno nuevo.

Código:

```
mezcla <- pastew(apellido, nombre, output="wave")
mezcla
```

Salida:

```
> mezcla <- pastew(apellido, nombre, output="wave")
> mezcla

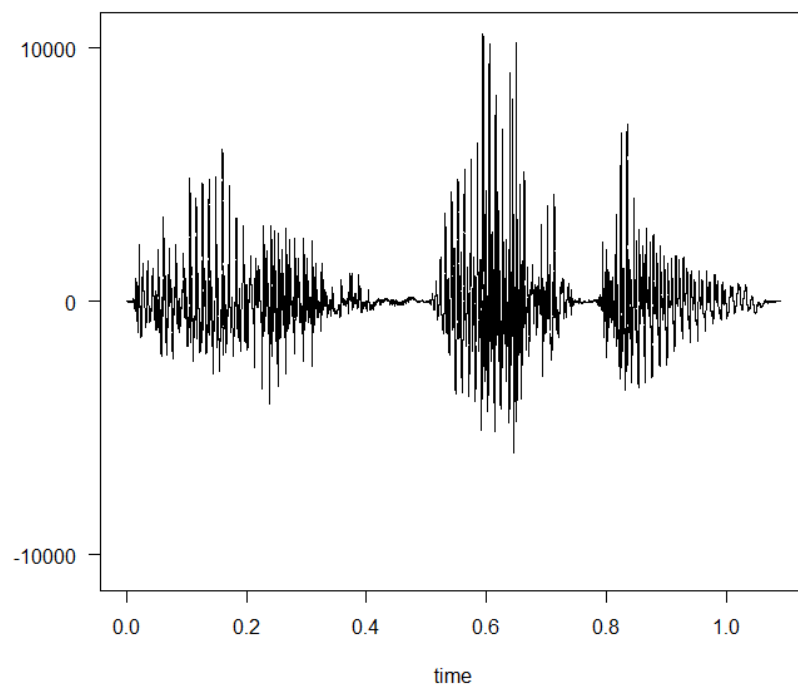
wave object
  Number of Samples:    52385
  Duration (seconds):   1.09
  Samplingrate (Hertz): 48000
  Channels (Mono/Stereo): Mono
  PCM (integer format): TRUE
  Bit (8/16/24/32/64): 16
```

5. Dibujar la forma de onda de la señal resultante.

Código:

```
plot(extractwave(mezcla,from=1,to = 52385))
```

Salida:



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

Código:

```
filtro <- bwfilter(mezcla,f=f, channel=1, n=1, from=10000, to=20000, bandpass=TRUE, listen = FALSE, output = "wave")
```

Salida:

```
> filtro <- bwfilter(mezcla,f=f, channel=1, n=1, from=10000, to=20000, bandpass=TRUE, listen = FALSE, output = "wave")
```

7. Almacenar la señal obtenida como un fichero WAV denominado "mezcla.wav".

Código:

```
writewave(filtro, file.path("C:/Users/gilru/OneDrive/Escritorio/Uni/4/2 Cuatri/PDIH/P5/mezcla.wav"))
```

Salida:

```
> writewave(filtro, file.path("C:/Users/gilru/OneDrive/Escritorio/Uni/4/2 Cuatri/PDIH/P5/mezcla.wav"))
Warning message:
In writewave(filtro, file.path("C:/Users/gilru/OneDrive/Escritorio/Uni/4/2 Cuatri/PDIH/P5/mezcla.wav")) :
  channels' data will be rounded to integers for writing the wave file
```

Aquí me sale un warning y cuando abro el archivo, se escucha un poco bajo pero se guarda bien el archivo y con las frecuencias eliminadas.

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".

Código:

```
oveja <- readWave('C:/Users/gilru/OneDrive/Escritorio/Uni/4/2 Cuatri/PDIH/P5/oveja.wav')
oveja

g <- oveja@samp.rate
ovejaECO <- echo(oveja,g,amp=c(0.8,0.4,0.2),delay=c(1,2,3), output="wave")

ovejaECO@left <- 10000 * ovejaECO@left
alreves <- revw(ovejaECO, output="wave")
listen(alreves)

writewave(alreves, file.path("C:/Users/gilru/OneDrive/Escritorio/Uni/4/2 Cuatri/PDIH/P5/alreves.wav"))
```

Salida:

```
> oveja <- readwave('C:/Users/gilru/OneDrive/Escritorio/Uni/4/2 Cuatri/PDIH/P5/oveja.wav')
> oveja

Wave object
  Number of Samples: 19764
  Duration (seconds): 2.47
  Samplingrate (Hertz): 8000
  Channels (Mono/Stereo): Mono
  PCM (integer format): TRUE
  Bit (8/16/24/32/64): 16

>
> g <- oveja@samp.rate
>
> ovejaECO <- echo(oveja,g,amp=c(0.8,0.4,0.2),delay=c(1,2,3), output="Wave")
>
> ovejaECO@left <- 10000 * ovejaECO@left
> alreves <- revw(ovejaECO, output="Wave")
> listen(alreves)
>
> writewave(alreves, file.path("C:/Users/gilru/OneDrive/Escritorio/Uni/4/2 Cuatri/PDIH/P5/alreves.wav"))
warning message:
In writewave(alreves, file.path("C:/Users/gilru/OneDrive/Escritorio/Uni/4/2 Cuatri/PDIH/P5/alreves.wav")) :
  channels' data will be rounded to integers for writing the wave file
> |
```