

# **PRACTICA 5**

## **Experimentación con el sistema de salida de sonido**

**Pablo Fernández Tello**

En esta práctica se nos pide aprender a identificar y representar gráficamente la forma de onda de señales de sonido, conocer la estructura de un fichero típico de sonido (ficheros WAV) y entender y operar con los parámetros principales de una señal de sonido.  
Para la realización de esta práctica he instalado Rstudio en Windows y el lenguaje R.

**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.**

Cargo las librerías previamente instaladas:

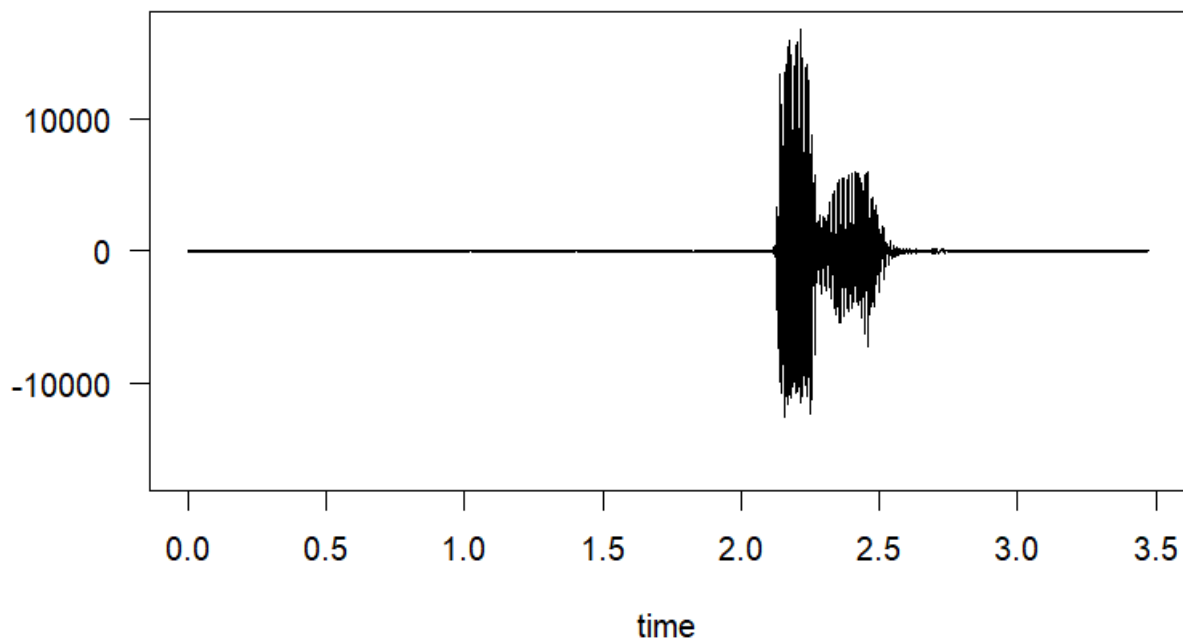
```
library(tuneR)  
library(seewave)  
library(audio)
```

Leo los ficheros de sonido (wav) y los guardo en un objeto cada uno:

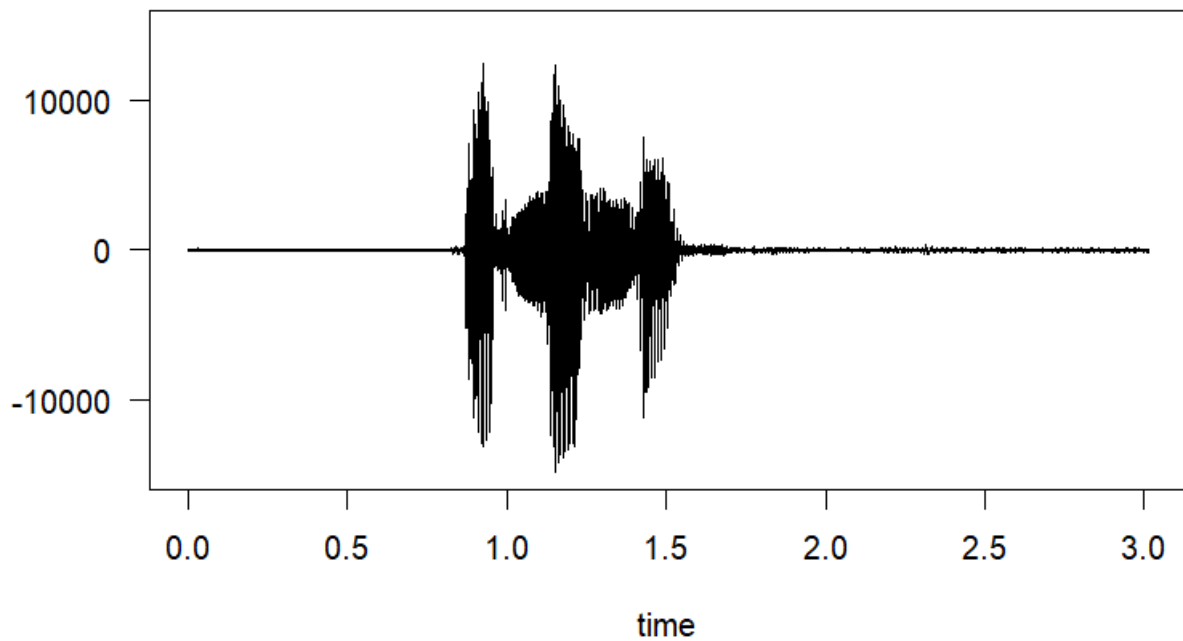
```
Nombre <- readWave('nombre.wav')  
Apellido <- readWave('apellido.wav')
```

**2. Dibujar la forma de onda de ambos sonidos.**

```
round(length(Nombre@left) / Nombre@samp.rate, 3)  
total_samples_Nombre <- length(Nombre@left)  
plot(extractWave(Nombre, from = 1, to = total_samples_Nombre))
```



```
round(length(Apellido@left) / Apellido@samp.rate, 3)
total_samples_Apellido <- length(Apellido@left)
plot(extractWave(Apellido, from = 1, to = total_samples_Apellido))
```



**length(Nombre@left) / Nombre@samp.rate** calcula la duración total del archivo de audio en segundos dividiendo el número total de muestras por la tasa de muestreo y **round**, redondea ese resultado a 3 decimales.

**extractWave** es una función del paquete seewave que extrae una porción de una señal de audio. En nuestro caso desde 1 al número total de muestras calculado con la función **length** y **plot** dibuja la forma de onda de la señal de audio extraída.

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

Simplemente escribiendo el nombre de los objetos:

Nombre

```
> Nombre
```

```
wave Object
  Number of Samples: 166728
  Duration (seconds): 3.47
  Samplingrate (Hertz): 48000
  Channels (Mono/Stereo): Mono
  PCM (integer format): TRUE
  Bit (8/16/24/32/64): 16
```

Apellido

```
> Apellido
```

```
Wave Object
```

```
Number of Samples: 144648  
Duration (seconds): 3.01  
Samplingrate (Hertz): 48000  
Channels (Mono/Stereo): Mono  
PCM (integer format): TRUE  
Bit (8/16/24/32/64): 16
```

```
... 1
```

#### 4. . Unir ambos sonidos en uno nuevo.

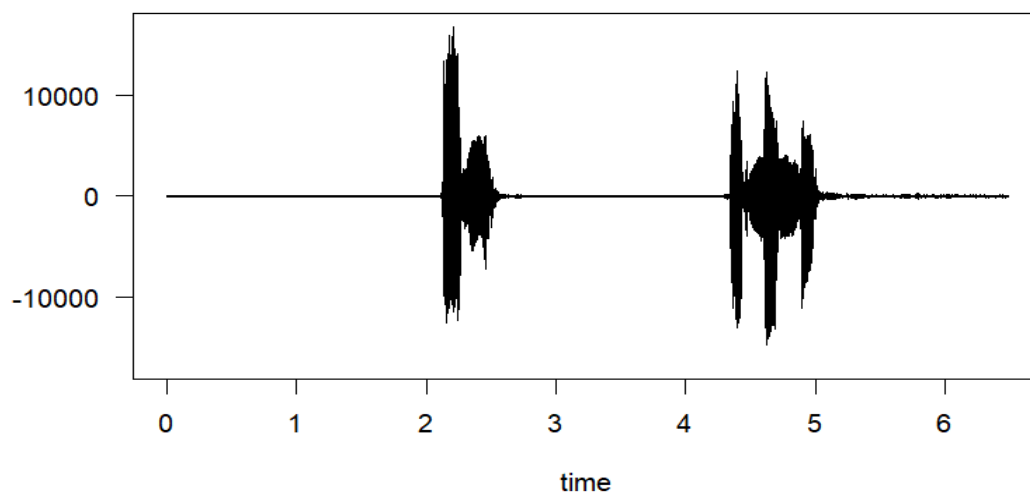
```
NombreApellido <- pastew(Apellido, Nombre , output="Wave")
```

La función **pastew** se utiliza para concatenar dos objetos de clase wave en secuencia, creando un nuevo objeto wave (NombreApellido) que contiene las señales de audio de ambos objetos uno después del otro.

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

Hacemos lo mismo que para dibujar las dos ondas anteriores:

```
round(length(NombreApellido@left) / NombreApellido@samp.rate, 3)  
total_samples_NombreApellido <- length(NombreApellido@left)  
plot(extractWave(NombreApellido, from = 1, to = total_samples_NombreApellido))
```



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

```
NombreApellidoFiltrado <- bwfilter(NombreApellido, 48000, channel = 1, n = 1, from = 10000, to = 20000, bandpass = TRUE, listen = FALSE, output = "Wave")
```

**bwfilter** es una función del paquete seewave en R que se utiliza para aplicar un filtro de banda a una señal de audio.

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

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

## **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”.**

Cargamos un nuevo archivo de sonido llamado edad:

```
Edad <- readWave('edad.wav')
```

Le aplicamos eco con la función echo:

```
EdadEco <- echo(Edad,f=22050,amp=c(0.8,0.4,0.2),delay=c(1,2,3), output="Wave")
```

```
EdadEco@left <- 10000 * EdadEco@left
```

Le damos la vuelta al sonido con la función revw:

```
EdadEcoAlreves <- revw(EdadEco, output="Wave")
```

Almacenamos la señal obtenida como un fichero WAV denominado "alreves.wav":

```
writeWave(EdadEcoAlreves, file.path("alreves.wav"))
```