

Práctica 5. Sonido en R

Alberto Socas Mendoza – Periféricos y Dispositivos de Interfaz Humana

En esta práctica trabajaremos la experimentación con el sistema de salida de sonido, cuyos objetivos son:

1. Leer dos ficheros de sonido (WAV y MP3) de unos pocos segundos de duración cada uno.
2. Dibujar la forma de onda de ambos sonidos.
3. Obtener la información de las cabeceras de ambos sonidos
4. Unir ambos sonidos en uno nuevo
5. Dibujar la forma de onda de la señal resultante
6. Pasarle un filtro de frecuencia para eliminar las frecuencias entre 10000Hz y 20000Hz
7. Almacenar la señal obtenida como un fichero WAV denominado “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”

Vamos a empezar:

1. Leer dos ficheros de sonido (WAV y MP3) de unos pocos segundos de duración cada uno.

Tras instalar los paquetes correspondientes, vamos a leer ficheros WAV y MP3: ejecutando lo siguiente:

```
> perro <- readWave('perro.wav')
> perro

Wave Object
  Number of Samples: 159732
  Duration (seconds): 3.62
  Samplingrate (Hertz): 44100
  Channels (Mono/Stereo): Stereo
  PCM (integer format): TRUE
  Bit (8/16/24/32/64): 16
```

Ilustración 1

```
> gato <- readMP3('gato.mp3')
> gato

Wave Object
  Number of Samples: 393984
  Duration (seconds): 8.93
  Samplingrate (Hertz): 44100
  Channels (Mono/Stereo): Stereo
  PCM (integer format): TRUE
  Bit (8/16/24/32/64): 16
```

Ilustración 2

Como podemos observar, es un código muy sencillo. Nos muestra el resultado del audio.

2. Dibujar la forma de onda de ambos sonidos.

Para poder dibujar las ondas de los sonidos debemos ejecutar el extractWave:

```
plot( extractWave(gato, from = 1, to = 393984) )
plot( extractWave(perro, from = 1, to = 393984) )
```

Ilustración 3

Como resultado tendremos ambas ondas, primero vemos la onda de “perro” y después la onda de “gato”:

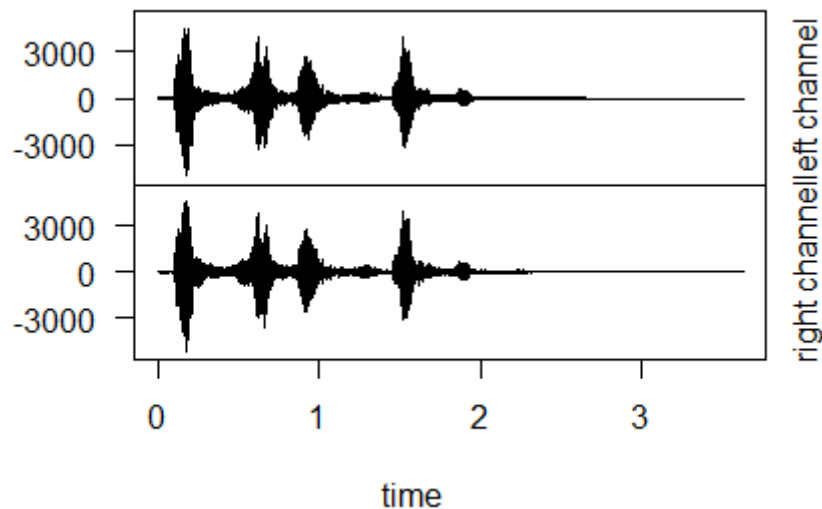


Ilustración 4. Onda Perro

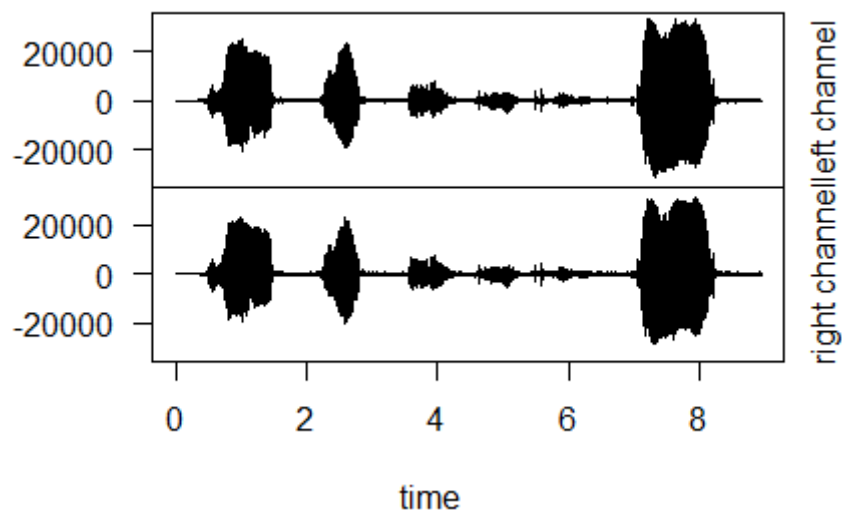


Ilustración 5. Onda Gato

3. Obtener la información de las cabeceras de ambos sonidos

Para obtener las cabeceras de ambos sonidos debemos ejecutar el siguiente código:

```
str(perro)
str(gato)
```

Ilustración 6

Como resultado, R nos mostrará los campos de los archivos de sonido:

```
> str(perro)
Formal class 'Wave' [package "tuneR"] with 6 slots
..@ left      : int [1:159732] 0 0 0 0 0 0 0 0 1 1 ...
..@ right     : int [1:159732] 0 0 0 0 0 0 0 0 1 1 ...
..@ stereo    : logi TRUE
..@ samp.rate: int 44100
..@ bit       : int 16
..@ pcm       : logi TRUE
> str(gato)
Formal class 'Wave' [package "tuneR"] with 6 slots
..@ left      : int [1:393984] 0 0 0 0 0 0 0 0 0 0 ...
..@ right     : int [1:393984] 0 0 0 0 0 0 0 0 0 0 ...
..@ stereo    : logi TRUE
..@ samp.rate: num 44100
..@ bit       : num 16
..@ pcm       : logi TRUE
```

Ilustración 7

4. Unir ambos sonidos en uno nuevo

Primero vamos a cortar el audio del gato con la función **cutw**:

```
a <- cutw(gato, from=6.5, to=8.9, output="Wave")
```

Ilustración 8

Ahora añadiremos la función perro y la nueva de gato, con la función **pastew**:

```
b <- pastew(perro, a, output="Wave")
```

Ilustración 9

5. Dibujar la forma de onda de la señal resultante

Como ya hemos visto anteriormente, usamos la función **plot** para dibujar la onda, cuyo resultado será:

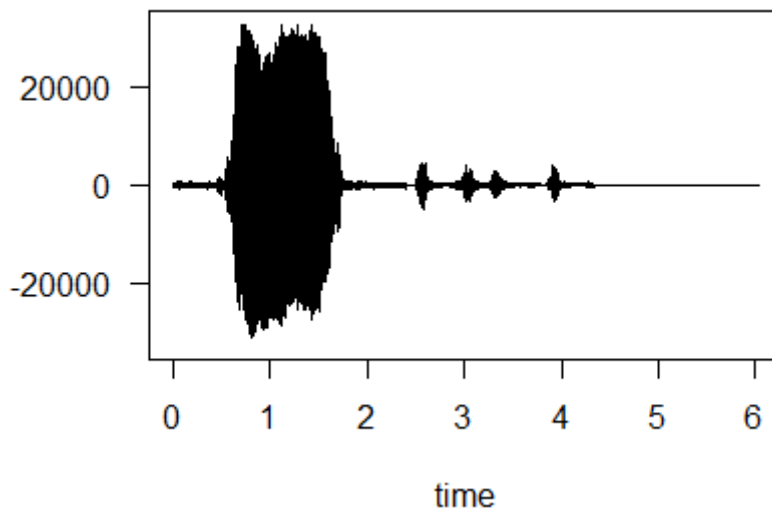


Ilustración 10

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

Para eliminar las frecuencias dentro de un rango debemos utilizar la función **bwfilterr**, como veremos a continuación:

```
filtros <- bwfilter(b, f=f, channel=1, n=1, from=10000, to= 20000, bandpass=TRUE, listen=FALSE, output="Wave")
```

Ilustración 11

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

Para guardar un fichero nuevo, usamos la función `writeWave`, como veremos a continuación:

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

Ilustración 12

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”

Por último vamos a cargar un nuevo archivo de sonido y aplicarle ECO, para ello usaremos la función `echo()`:

```
hola <- readWave('hola.wav')
str(hola)
holaECO <- echo(hola, f=22050, amp=c(0.8,0.4,0.2),delay=c(1,2,3),output="Wave")
```

Ilustración 13

Podemos comprobar su funcionamiento con la función `listen`.

Ahora vamos a darle la vuelta al sonido con la función `revw`:

```
alreves <- revw(hola, output="Wave")
```

Ilustración 14

Y por último guardamos el nuevo sonido con la función ya vista `writeWave`.