

Sonido

PRACTICA 5

Pablo García Fernández
Periféricos y Dispositivos de Interfaz Humana



**UNIVERSIDAD
DE GRANADA**

El objetivo de esta práctica es familiarizarse con el lenguaje R y el entorno RStudio a la hora de manipular sonidos.

Una vez instalados tanto R como el entorno RStudio el siguiente paso es crear los sonidos que vamos a utilizar en la práctica. Esto lo haremos con la herramienta *espeak* de Ubuntu. Con el comando *espeak "Pablo" -w nombre.wav* creo una pista de audio en formato Wave que contendrá mi nombre, también hago lo mismo para mis apellidos.

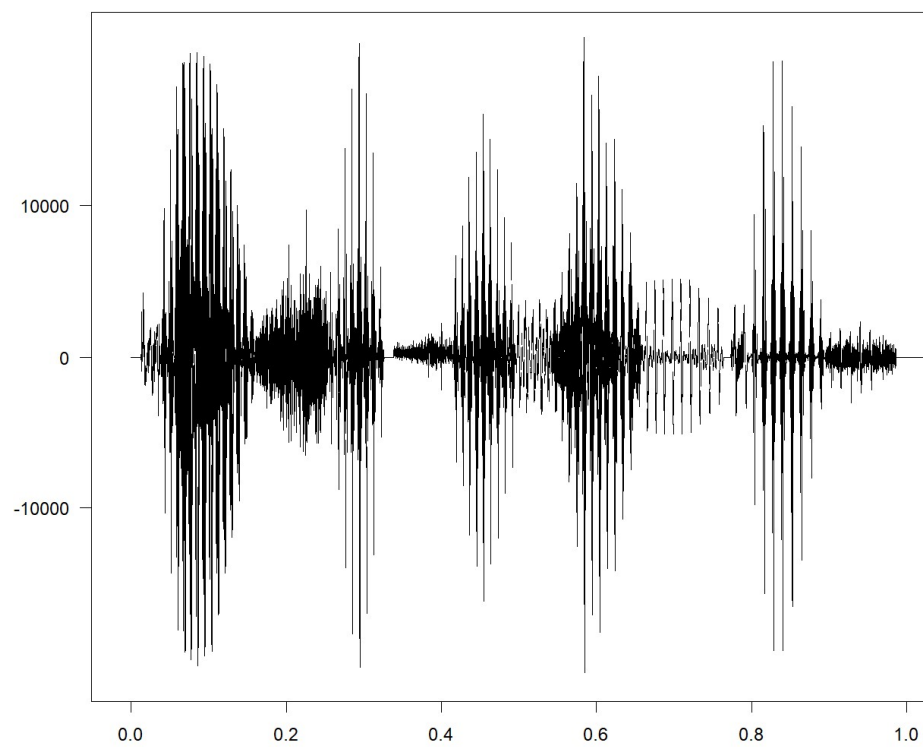
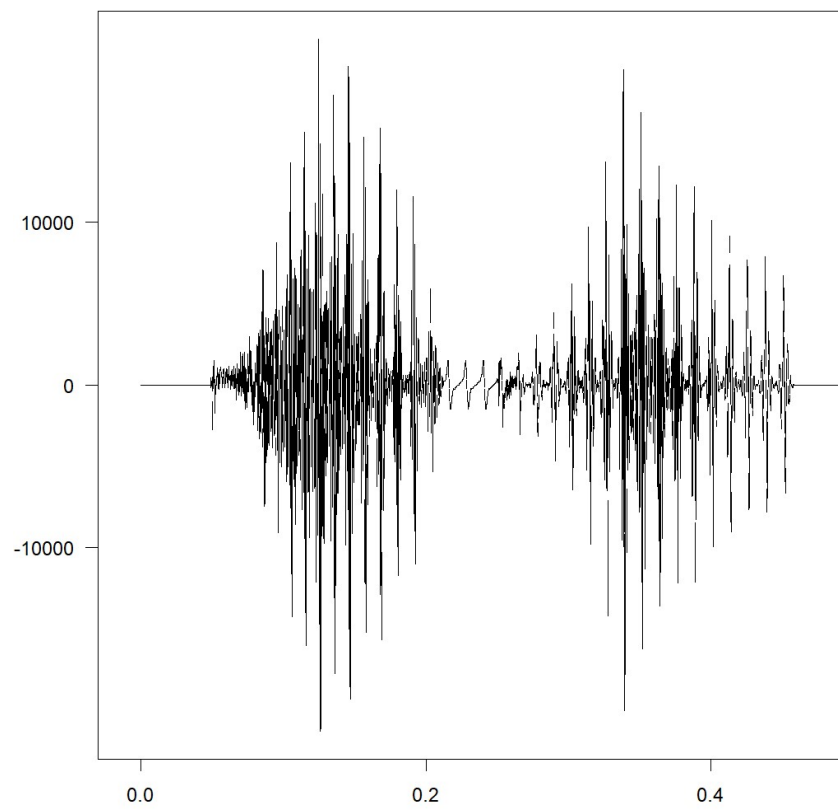
En cuanto al código R es necesario utilizar las librerías *tuneR*, *seewave* y *audio*. También hay que establecer el directorio donde se va a trabajar con *setwd*. Lo siguiente es leer los archivos de audio y mostrar las cabeceras.

```
> # establecer el path concreto en cada caso a la carpeta de trabajo
> setwd("D:\\MisArchivos\\Universidad\\23_24\\PDIH\\P5")
>
> nombre = readWave('nombre.wav')
> apellidos = readWave('apellidos.wav')
>
> str(nombre)
Formal class 'wave' [package "tuneR"] with 6 slots
..@ left      : int [1:16736] 0 0 0 0 0 0 0 0 0 0 ...
..@ 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 [1:28375] 0 0 0 0 0 0 0 0 0 0 ...
..@ right     : num(0)
..@ stereo    : logi FALSE
..@ samp.rate: int 22050
..@ bit       : int 16
..@ pcm       : logi TRUE
> |
```

Ahora vamos a dibujar la forma de las ondas de cada archivo.

#Mostrar las ondas

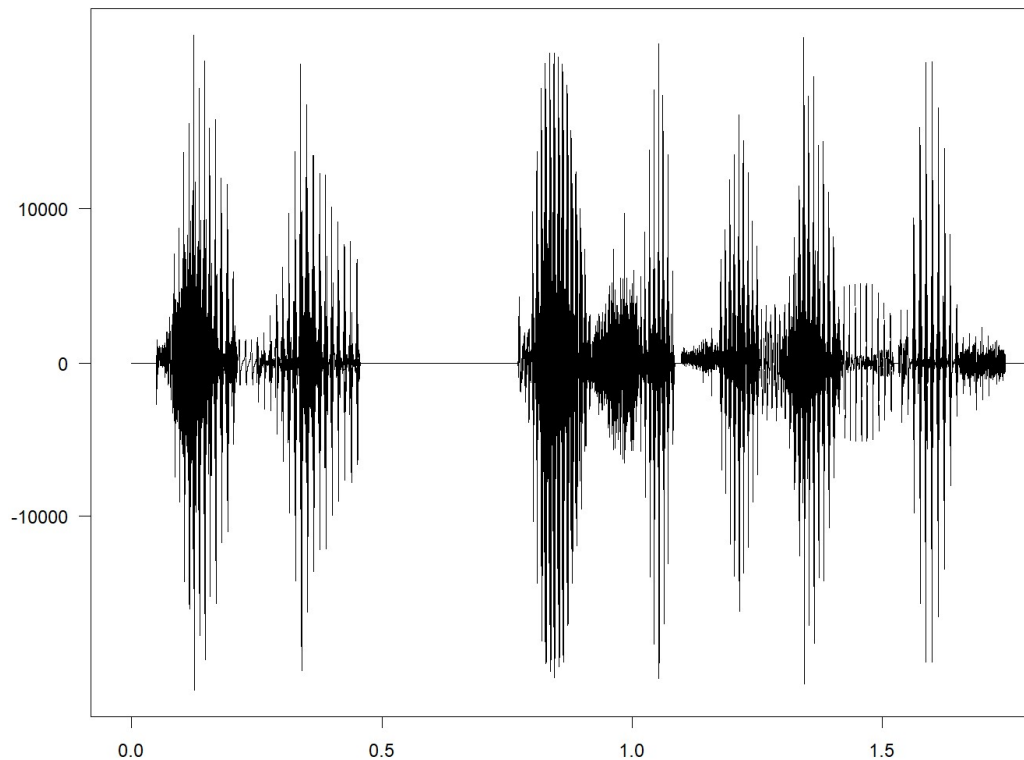
```
plot( extractWave(nombre, from = 1, to = 393984) )|
plot( extractWave(apellidos, from = 1, to = 393984) )
```



Al representar las pistas de audios se pueden observar las sílabas de las palabras.

Ahora vamos a combinar ambas pistas de audio en una sola para que diga el nombre y luego los apellidos.

```
#Combinar audios
mezcla = pastew(apellidos, nombre, output="wave")
plot( extractWave(mezcla, from = 1, to=393984) )
listen(mezcla)
```

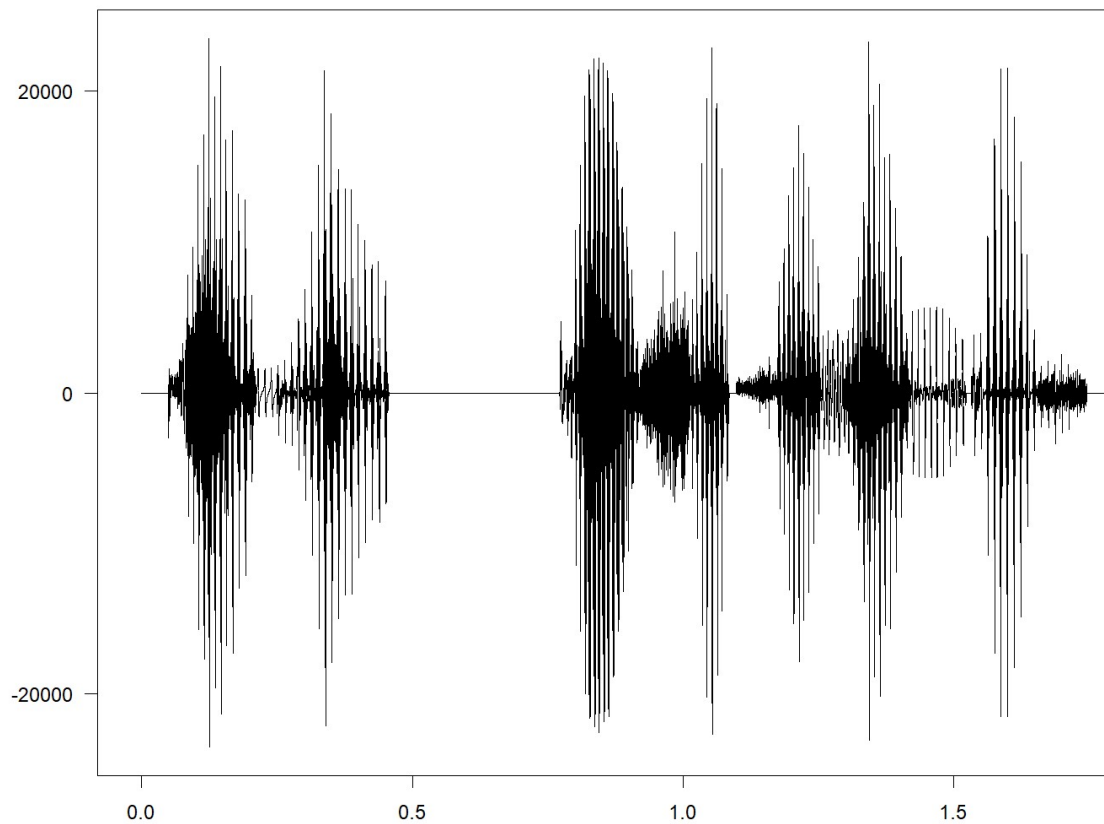


Si observamos los espectrogramas individuales y combinado podemos ver que se ha realizado correctamente.

```
#Filtro de frecuencia entre 10000 HZ y 20000HZ
filtrado = fir(wave = mezcla,
               from = 10000, to = 20000,
               bandpass = FALSE,
               output = "wave")

plot( extractWave(filtrado, from = 1, to = 393984) )
listen(filtrado)
```

La diferencia entre gráficas a penas se nota pero hay ciertos picos entre los valores indicados que desaparecen por lo que el filtro funciona.



Para guardar el sonido en un archivo sin que de un error es necesario normalizarlo de la siguiente forma

```
#Normalizar el sonido para guardarlo
filtrado = normalize(filtrado, unit="16", center = TRUE, level=1)
writewave(filtrado, file.path("mezcla.wav"))
```

Por último escogemos un nuevo sonido, le aplicamos eco y le damos la vuelta.

```
#Nuevo audio para añadirle eco y darle la vuelta
perro = readwave("perro.wav")
eco = echo(perro, amp=c(0.8,0.4,0.2),delay=c(1,2,3),output = "wave")
alreves = revw(eco, output="wave")
listen(alreves)
writewave(alreves, file.path("alreves.wav"))
```