

# Ejercicios de Strings en R

David Criado Ramón

26/10/2019

1. Crea un vector de strings que contengan 3 datos: tu primer nombre y tus dos apellidos. A partir de éste, crea un nuevo string con la inicial de tu nombre (y un punto) y el apellido completo utilizando las utilidades de R. En mi caso debería quedar: D. Criado Ramón.

```
nombre <- c("David", "Criado", "Ramon")
nombre_abreviado <- paste(substr(nombre[1],1,1), nombre[2], nombre[3])
nombre_abreviado <- paste(nombre_abreviado, '.', sep='')
nombre_abreviado
```

```
## [1] "D Criado Ramon."
```

2. Dado un vector de fechas expresadas como strings (e.g., “2005-11-28”), muestra sólomente aquellas correspondientes a los meses impares.

```
# Creamos un vector con varias fechas.
fechas <- c("1996-10-07", "2019-10-24", "2015-01-13", "2005-11-28")

# Extraemos los números y los pasamos a enteros.
fechas_descompuestas <- strsplit(fechas, split='-')
fechas_descompuestas <- as.integer(unlist(fechas_descompuestas))

# Seleccionamos los meses.
indices_meses <- seq(2, 2 + 3 * (length(fechas) - 1), 3)
meses <- fechas_descompuestas[indices_meses]

# Comprobamos si son impares.
meses_impares_pos <- which(meses%%2==1)

# Mostramos las fechas con meses impares.
fechas_impares <- fechas[meses_impares_pos]
fechas_impares
```

```
## [1] "2015-01-13" "2005-11-28"
```

3. Dado un string con varias palabras (e.g., “Esto es una frase, pero no cualquier frase”) crea un vector con cada una de las palabras del string ([“Esta”, “es”, “una”, “frase”, “pero”, “no”, “cualquier”, “frase”]).

```
frase <- "Esto es una frase, pero no cualquier frase"
# Descomponemos la frase tomando como puntos de separación
# el espacio o la coma seguida de espacio.
frase_descompuesta <- unlist(strsplit(frase, split= " |, "))
frase_descompuesta
```

```
## [1] "Esto"      "es"        "una"       "frase"     "pero"     "no"
## [7] "cualquier" "frase"
```

4. Busca las palabras que sólo usan las vocales “a” y “e” en un vector de strings.

```
palabras <- c("hola", "azul", "naranja", "verde", "lata", "saco", "bola", "bolo", "vela")

# Seleccionamos todas las palabras que contienen a o e.
# quitando aquellas que contienen i, o, u.
palabras_ae <- palabras[grepl("[ae]", palabras)]

palabras_solo_ae <- palabras_ae[!grepl("[iou]", palabras_ae)]
palabras_solo_ae
```

```
## [1] "naranja" "verde"    "lata"     "vela"
```

5. Dados tres vectores día, mes y año crea un vector con las fechas completas. Si la fecha es inválida, ésta se descartará (hint: investiga la función as.Date).

```
dia <- c(31,5,7,11,28,31,2,25,30)
mes <- c(9,1,5,4,6,2,3,4,7)
año <- c(2003,2007,2010,1995,1996,2013,2018,2019,2010)

# Juntamos las fechas en un string.
fechas_str <- paste(dia, mes, año, sep='-')
fechas <- as.Date(fechas_str, format = "%d-%m-%Y")

# Eliminamos fechas inválidas (NA)
fechas <- na.exclude(fechas)
fechas
```

```
## [1] "2007-01-05" "2010-05-07" "1995-04-11" "1996-06-28" "2018-03-02"
## [6] "2019-04-25" "2010-07-30"
```