

# String Manipulation

Rafael Sanjuan

## Ejercicio 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: R. Romero Zaliz.

```
# La funcion _c_ combina los elementos recibidos en un vector, en este caso le
# paso tres strings con mis nombres y apellidos.
v = c('Rafael', 'Sanjuan', 'Aguilera')

# Uso _substr_ para extraer la primera letra de mi nombre.
primera_letra = substr(v[1], 1, 1)

# Uso la funcion paste para concatenar el resultado, agregando tambien un punto.
nombre_abreviado = paste(primera_letra, ".", v[2], v[3])
nombre_abreviado

## [1] "R . Sanjuan Aguilera"
```

## Ejercicio 2.

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

```
fechas <- c("2005-11-28", "2019-10-15", "2019-07-21", "1996-02-12", "2017-07-24")
for (i in fechas) {
  # Extraigo el mes y lo combierto en numerico
  mes = substr(i, 6, 7)
  mes = as.numeric(mes)

  # Imprimo las fechas con meses impares (el modulo es distinto de cero)
  if (mes %% 2 != 0) {
    print(paste("En", i, "el mes es impar."))
  }
}

## [1] "En 2005-11-28 el mes es impar."
## [1] "En 2019-07-21 el mes es impar."
## [1] "En 2017-07-24 el mes es impar."
```

## Ejercicio 3.

Dado un string con varias palabras (e.g., "Esta 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 <- "Esta es una frase, pero no cualquier frase."
# Usar strsplit divide la frase devolviendo una lista con un solo elemento,
# un vector que contiene todas las palabras. Usando el operador de acceso [[1]]
# devolvemos y asignamos el vector.
```

```

palabras <- strsplit(frase, split=" ")[[1]]
palabras

## [1] "Esta"      "es"        "una"        "frase,"    "pero"      "no"
## [7] "cualquier" "frase."

```

#### Ejercicio 4.

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

```

# Obtengo el conjunto de las palabras que tienen una a o e menos el conjunto
# de palabras que tienen otra vocal diferente.
indices = setdiff(grep("[ae]", palabras), grep("[iou]", palabras))

# Las mostramos
palabras[indices]

## [1] "Esta"      "es"        "frase,"    "frase."

```

#### Ejercicio 5.

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

```

dia <- c("28", "15", "21", "12", "24", "as")
mes <- c("11", "10", "07", "02", "07", "as")
anno <- c("2005", "2019", "2019", "1996", "2017", "as")

# La funcion paste es compatible con vectores, asi que simplemente:
res = paste0(dia, "-", mes, "-", anno)
# Combina los elementos de los vectores uno a uno.

# Combirtiendo en fecha, los incorrectos son devueltos como NA.
res = as.Date(res, "%d-%m-%Y")
# Borramos los NA del vector
res = res[!is.na(res)]
res

## [1] "2005-11-28" "2019-10-15" "2019-07-21" "1996-02-12" "2017-07-24"

```