Cadenas de texto

Ya hemos comentado en documentos anteriores que R permite manejar cadenas de texto utilizando indistintamente comillas simples o dobles. Podemos crearlas con un texto predefinido en el programa, o pedirlas al usuario a través de la instrucción `readLines`:

```
texto <- "Hola"
texto2 <- 'Buenas'
texto3 <- readLines("stdin", n=1)
```

A continuación veremos algunas de las funciones principales incorporadas en R para manejo de cadenas de texto, aunque ya adelantamos que, comparado con otros lenguajes como Python o Java, estas funciones son más reducidas. Sin embargo, gracias a algunas librerías de tratamiento de datos que veremos en algún documento posterior, estas funcionalidades pueden ampliarse.

1. Algunas operaciones básicas

Podemos hacer algunas operaciones básicas sobre las cadenas de texto, tales como concatenar con `paste`. En este caso, el parámetro *sep* indica el carácter de separación entre los elementos que concatenamos.

```
resultado <- paste("uno", "dos", "tres", sep=" ")
print(resultado)
# uno dos tres
```

Podemos convertir también cualquier dato simple a cadena usando la instrucción `as.character`:

```
edad <- 43
texto <- paste("Tengo", as.character(edad), "años", sep= " ")
```

Para saber la longitud de una cadena (número de caracteres) usamos la instrucción `nchar`:

```
texto <- "Hola, buenas"
longitud <- nchar(texto)
```

A diferencia de otros lenguajes, en R no existe un modo sencillo de acceder a un carácter suelto dentro de la cadena. Lo que deberemos hacer es obtener una subcadena, como veremos más adelante en este documento.

2. Otras operaciones sobre cadenas

Además de estas operaciones básicas, también podemos utilizar algunas opciones que también ofrecen otros lenguajes. Por ejemplo, tenemos la instrucción `strsplit` que devuelve una lista con los elementos que ha podido **extraer de una cadena**, a partir de un delimitador especificado. Sin embargo, a diferencia de cómo actúan funciones similares en otros lenguajes, en este caso debemos añadir un paso adicional más, con la función `unlist`, para poder recorrer los elementos extraídos de la cadena original:

```
texto <- "uno dos tres"
partes <- unlist(strsplit(texto, split=" "))
for (parte in partes)
{
    print(parte)
}
```

El paso inverso, es decir, **unir** partes de un texto con un separador, se puede hacer a través de la instrucción `paste` vista antes para concatenar, añadiendo en el parámetro *collapse* el carácter separador:

```
texto <- "uno dos tres"
partes <- unlist(strsplit(texto, split=" "))
texto2 <- paste(partes, collapse=',')
print(texto2)
# uno,dos,tres
```

Las instrucciones `tolower` y `toupper` convierten todo el texto en **minúsculas / mayúsculas**, respectivamente, devolviendo una cadena con el resultado:

```
texto <- "Hola, buenas"
textoMayus <- toupper(texto) # HOLA, BUENAS
```

La instrucción `grepl` permite **buscar** si un texto está contenido dentro de otro.

```
texto <- "Hola, buenas"
if(grepl("la", texto))
{
    # true
}
```

Por otra parte, la instrucción `gregexpr` devuelve todas las posiciones en que se encuentra una subcadena dentro de otra. Notar que las posiciones empiezan a contarse desde 1, a diferencia de otros lenguajes.

```
texto <- "Erase una vez un ratón y un duende"
posiciones <- unlist(gregexpr("un", texto))
print(posiciones)
# 7 15 27
```

Si queremos **extraer una subcadena** a partir de una cadena principal, usamos la instrucción `substring` indicando el texto con el que trabajar, la posición de inicio donde queremos empezar a contar y la posición de fin, ambos inclusive y contando desde 1:

```
texto <- "Hola, buenas tardes"
subcadena <- substring(texto, 7, 12)  # buenas
```

A la hora de **comparar** cadenas para ver cuál es mayor o menor alfabéticamente, podemos emplear los operadores de comparación.

```
texto1 <- "Uno"
texto2 <- "dos"

if(texto1 > texto2) {
    print("Es mayor el 1")
} else {
    print("Es mayor el 2") # Muestra esto
}
```

También nos puede resultar útil **limpiar** una cadena: eliminar espacios innecesarios al inicio o al final. Para ello podemos emplear la instrucción `trimws`:

```
texto1 <- "  Uno   "
texto2 <- trimws(texto1)
```

Ejercicio 1:

Crea un programa llamado `sumaSecuencia.r` que le pida al usuario una secuencia de números separados por espacios y calcule la suma total de esos números.

AYUDA: vídeo con solución del ejercicio.