

Manipulación de cadenas con stringr : : GUÍA RÁPIDA



El paquete **stringr** proporciona un conjunto de herramientas internamente consistentes para trabajar con cadenas de caracteres, por ej. secuencias de caracteres delimitados por comillas.

Detectar Coincidencias



str_detect(cadena, **patrón**) Detecta el patrón coincidente en una cadena.
`str_detect(fruta, "a")`



str_which(cadena, **patrón**) Encuentra los índices de las cadenas que contienen un patrón coincidente.
`str_which(fruta, "a")`



str_count(cadena, **patrón**) Cuenta el número de coincidencias en una cadena.
`str_count(fruta, "a")`



str_locate(cadena, **patrón**) Localiza las posiciones en las que el patrón coincide con la cadena. También **str_locate_all**.
`str_locate(fruta, "a")`

Subconjunto de cadenas



str_sub(cadena, start = 1L, end = -1L) Extrae subcadenas de un vector de caracteres.
`str_sub(fruta, 1, 3); str_sub(fruta, -2)`



str_subset(cadena, **patrón**) Devuelve sólo las cadenas que contienen un patrón coincidente.
`str_subset(fruta, "b")`



str_extract(cadena, **patrón**) Devuelve el primer patrón encontrado que coincide en cada cadena, como un vector. También **str_extract_all** para devolver cada patrón coincidente. `str_extract(fruta, "[aeiou]")`



str_match(cadena, **patrón**) Devuelve el primer patrón encontrado que coincide en cada cadena, como una matriz, con una columna para cada una () agrupado por patrón. También **str_match_all**.
`str_match(sentences, "(el|la) ([^]+)")`

Gestionar Longitudes



str_length(cadena) Los largos de las cadenas (por ej. número de puntos de código, suele ser igual al número de caracteres). `str_length(fruta)`



str_pad(cadena, largo, side = c("left", "right", "both"), pad = " ") Extiende cadenas a un largo constante. `str_pad(fruta, 17)`



str_trunc(cadena, ancho, side = c("right", "left", "center"), ellipsis = "...") Trunca el ancho de una cadena, eliminando el contenido sobrante. `str_trunc(fruta, 3)`



str_trim(cadena, side = c("both", "left", "right")) Elimina los espacios en blanco desde el inicio y/o al final de una cadena. `str_trim(fruta)`

Transformar Cadenas



str_sub() <- valor. Reemplaza subcadenas identificadas con `str_sub()` y se asignan al resultado.
`str_sub(fruta, 1, 3) <- "str"`



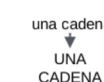
str_replace(cadena, **patrón**, reemplazo) Reemplaza el primer patrón coincidente en cada cadena. `str_replace(fruta, "a", "-")`



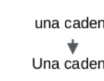
str_replace_all(cadena, **patrón**, replacement) Reemplaza todos los patrones coincidentes en cada cadena.
`str_replace_all(fruta, "a", "-")`



str_to_lower(cadena, locale = "en")¹ Convierte cadenas a minúscula.
`str_to_lower(oraciones)`



str_to_upper(string, locale = "en")¹ Convierte cadenas a mayúsculas.
`str_to_upper(oraciones)`



str_to_title(string, locale = "en")¹ Convierte cadenas a título. `str_to_title(oraciones)`

Juntar y Separar



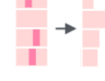
str_c(..., sep = "", collapse = NULL) Une múltiples cadenas en una. `str_c(letters, LETTERS)`



str_c(..., sep = "", collapse = NULL) Colapsa un vector de cadenas en una sola cadena.
`str_c(letters, collapse = "")`



str_dup(cadena, veces) Repite cadenas varias veces. `str_dup(fruta, times = 2)`



str_split_fixed(cadena, **patrón**, n) Divide un vector de cadenas en una matriz de subcadenas (dividiendo en las ocurrencias de cada patrón de coincidencia). También **str_split** para devolver una lista de subcadenas.
`str_split_fixed(fruta, " ", n=2)`



str_glue(..., .sep = "", .envir = parent.frame()) Crea una cadena a partir de cadenas y {expresiones} para evaluar. `str_glue("Pi is {pi}")`



str_glue_data(x, ..., .sep = "", .envir = parent.frame(), .na = "NA") Usa un conjunto de datos, lista, o entorno para crear una cadena a partir de cadenas y {expresiones} para evaluar. `str_glue_data(mtcars, "{rownames(mtcars)} has {hp} hp")`

Ordenar Cadenas



str_order(x, decreasing = FALSE, na.last = TRUE, locale = "en", numeric = FALSE, ...) ¹ Devuelve el vector de índices que ordena un vector de caracteres. `x[str_order(x)]`



str_sort(x, decreasing = FALSE, na.last = TRUE, locale = "en", numeric = FALSE, ...) ¹ Ordena un vector de caracteres. `str_sort(x)`

Funciones auxiliares

manzana
banana
pera

str_conv(cadena, encoding) Sobrescribe el tipo de codificación de una cadena.
`str_conv(fruta, "ISO-8859-1")`

manzana
banana
pera

str_view(cadena, **patrón**, match = NA) Vista en HTML de la primera coincidencia de una expresión regular en cada cadena.
`str_view(fruta, "[aeiou]")`

str_view_all(cadena, **patrón**, match = NA) Vista en HTML de todas las coincidencias de la expresión regular. `str_view_all(fruta, "[aeiou]")`

str_wrap(cadena, width = 80, indent = 0, exdent = 0) Envuelve cadenas en párrafos formateados adecuadamente.
`str_wrap(sentences, 20)`

¹ Ver bit.ly/ISO639-1 para una lista completa de locales.