

R String Arrays

Fan Wang

2020-08-21

Contents

1	String Arrays	1
1.1	String Replace	1
1.1.1	Search If and Which String Contains	2
1.2	String Split	2
1.3	String Concatenate	3
1.4	String Add Leading Zero	3
1.5	Substring Components	3
1.6	Substring and File Name	3

1 String Arrays

Go to the [RMD](#), [R](#), [PDF](#), or [HTML](#) version of this file. Go back to [fan's REconTools Package](#), [R Code Examples Repository \(bookdown site\)](#), or [Intro Stats with R Repository \(bookdown site\)](#).

1.1 String Replace

- r string wildcard replace between regex
- [R - replace part of a string using wildcards](#)

```
# String replacement
gsub(x = paste0(unique(df.slds.stats.perc$it.inner.counter), ':',
  unique(df.slds.stats.perc$z_n_a_n), collapse = ';'),
  pattern = "\\n",
  replacement = "")
gsub(x = var, pattern = "\\n", replacement = "")
gsub(x = var.input, pattern = "\\.", replacement = "_")
```

String replaces a segment, search by wildcard. Given the string below, delete all text between carriage return and pound sign:

```
st_tex_text <- "\\n% Lat2ex Comments\\n\\newcommand{\\exa}{\\text{from external file: } \\alpha + \\beta}"
st_clean_a1 <- gsub("\\%.*?\\n", "", st_tex_text)
st_clean_a2 <- gsub("L.*?x", "[LATEX]", st_tex_text)
print(paste0('st_tex_text:', st_tex_text))
```

```
## [1] "st_tex_text:\\n% Lat2ex Comments\\n\\newcommand{\\exa}{\\text{from external file: } \\alpha + \\beta}"
print(paste0('st_clean_a1:', st_clean_a1))
```

```
## [1] "st_clean_a1:\\n\\newcommand{\\exa}{\\text{from external file: } \\alpha + \\beta}\\n"
```

```
print(paste0('st_clean_a2:', st_clean_a2))
```

```
## [1] "st_clean_a2:\n% [LATEX] Comments\n\\newcommand{\\exa}{\\text{from external file: } \\alpha + \\
```

String delete after a particular string:

```
st_tex_text <- "\\end{equation}\n}\n% Even more comments from Latex preamble"
st_clean_a1 <- gsub("\\\\n%.*", "", st_tex_text)
print(paste0('st_tex_text:', st_tex_text))
```

```
## [1] "st_tex_text:\\end{equation}\n}\n% Even more comments from Latex preamble"
```

```
print(paste0('st_clean_a1:', st_clean_a1))
```

```
## [1] "st_clean_a1:\\end{equation}\n}"
```

1.1.1 Search If and Which String Contains

- r if string contains
- r if string contains either or grepl
- Use grepl to search either of multiple substrings in a text

Search for a single substring in a single string:

```
st_example_a <- 'C:/Users/fan/R4Econ/amto/tibble/fs_tib_basics.Rmd'
st_example_b <- 'C:/Users/fan/R4Econ/amto/tibble/_main.html'
grepl('_main', st_example_a)
```

```
## [1] FALSE
```

```
grepl('_main', st_example_b)
```

```
## [1] TRUE
```

Search for if one of a set of substring exists in a set of strings. In particular which one of the elements of *ls_spn* contains at least one of the elements of *ls_str_if_contains*. In the example below, only the first path does not contain either the word *aggregate* or *index* in the path. This can be used after all paths have been found recursively in some folder to select only desired paths from the full set of possibilities:

```
ls_spn <- c("C:/Users/fan/R4Econ//panel/basic/fs_genpanel.Rmd",
            "C:/Users/fan/R4Econ//summarize/aggregate/main.Rmd",
            "C:/Users/fan/R4Econ//summarize/index/fs_index_populate.Rmd")
ls_str_if_contains <- c("aggregate", "index")
str_if_contains <- paste(ls_str_if_contains, collapse = "|")
grepl(str_if_contains, ls_spn)
```

```
## [1] FALSE TRUE TRUE
```

1.2 String Split

Given some string, generated for example by cut, get the lower cut starting points, and also the higher end point

```
# Extract 0.216 and 0.500 as lower and upper bounds
st_cut_cate <- '(0.216,0.500]'
# Extract Lower Part
substring(strsplit(st_cut_cate, ",")[[1]][1], 2)
```

```
## [1] "0.216"
```

```
# Extract second part except final bracket Option 1
intToUtf8(rev(utf8ToInt(substring(intToUtf8(rev(utf8ToInt(strsplit(st_cut_cate, ",")[[1]][2])), 2))))

## [1] "0.500"

# Extract second part except final bracket Option 2
gsub(strsplit(st_cut_cate, ",")[[1]][2], pattern = "]", replacement = "")

## [1] "0.500"
```

1.3 String Concatenate

```
# Simple Collapse
vars.group.by <- c('abc', 'efg')
paste0(vars.group.by, collapse='|')

## [1] "abc|efg"
```

1.4 String Add Leading Zero

```
# Add Leading zero for integer values to allow for sorting when
# integers are combined into strings
it_z_n <- 1
it_a_n <- 192
print(sprintf("%02d", it_z_n))

## [1] "01"

print(sprintf("%04d", it_a_n))

## [1] "0192"
```

1.5 Substring Components

Given a string, with certain structure, get components.

- r time string get month and year and day

```
snm_full <- "20100701"
snm_year <- substr(snm_full, 0, 4)
snm_month <- substr(snm_full, 5, 6)
snm_day <- substr(snm_full, 7, 8)
print(paste0('full:', snm_full,
             ', year:', snm_year,
             ', month:', snm_month,
             ', day:', snm_day))

## [1] "full:20100701, year:2010, month:07, day:01"
```

1.6 Substring and File Name

From path, get file name without suffix.

- r string split
- r list last element
- r get file name from path
- r get file path no name

```

st_example <- 'C:/Users/fan/R4Econ/amto/tibble/fs_tib_basics.Rmd'
st_file_wth_suffix_s <- tail(strsplit(st_example, "/")[[1]],n=1)
st_file_wno_suffix_s <- tools::file_path_sans_ext(basename(st_example))
st_fullpath_nosufx_s <- sub('\\.Rmd$', '', st_example)
st_fullpath_noname_s <- dirname(st_example)

print(strsplit(st_example, "/"))

## [[1]]
## [1] "C:"          "Users"       "fan"         "R4Econ"      "amto"
print(paste0('st_file_wth_suffix_s:', st_file_wth_suffix_s))

## [1] "st_file_wth_suffix_s:fs_tib_basics.Rmd"
print(paste0('st_file_wno_suffix_s:', st_file_wno_suffix_s))

## [1] "st_file_wno_suffix_s:fs_tib_basics"
print(paste0('st_fullpath_nosufx_s:', st_fullpath_nosufx_s))

## [1] "st_fullpath_nosufx_s:C:/Users/fan/R4Econ/amto/tibble/fs_tib_basics"
print(paste0('st_fullpath_noname_s:', st_fullpath_noname_s))

## [1] "st_fullpath_noname_s:C:/Users/fan/R4Econ/amto/tibble"

```