

R String Arrays

Fan Wang

2020-04-11

Contents

String Arrays	1
-------------------------	---

String Arrays

Go to the [RMD](#), [R](#), [PDF](#), or [HTML](#) version of this file. Go back to [fan's REconTools Package](#), [R4Econ Repository \(bookdown site\)](#), or [Intro Stats with R Repository](#).

```
# 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 Replace

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
```

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

String Concatenate

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

```
# 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))
```

String Add Leading Zero

```
## [1] "01"
```

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

```
## [1] "0192"
```

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 <- tail(strsplit(st_example, "/")[[1]],n=1)
st_file_wno_suffix <- sub('\\.Rmd$', '', basename(st_example))
st_fullpath_nosufx <- sub('\\.Rmd$', '', st_example)
st_lastpath_noname <- (dirname(st_example))
st_fullpath_noname <- dirname(st_example)

print(strsplit(st_example, "/"))
```

```
## [[1]]
## [1] "C:" "Users" "fan" "R4Econ" "amto"

print(st_file_wth_suffix)
```

```
## [1] "fs_tib_basics.Rmd"

print(st_file_wno_suffix)
```

```
## [1] "fs_tib_basics"

print(st_fullpath_nosufx)
```

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

```
print(st_lastpath_noname)
```

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

```
print(st_fullpath_noname)
```

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