

Convert R Markdown File to R, PDF and HTML

Fan Wang

2020-05-16

Contents

1 Rmd to HTML	1
1.1 Search and Find all Files in Repository	1
1.2 Search and Find all Git Modified or New Rmd	2
1.3 Resave an Existing File with Different Name Different Folder	3
1.3.1 Replacment Function Change Markdown Hierarchy and Add to YAML	3
1.4 Search and Render Rmd File and Save HTML, PDF or R	4

1 Rmd to HTML

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 Search and Find all Files in Repository

Search inside directories, for all files in a repository that have a particular suffix and that don't contain skip pattern list string items.

```
# Serch Folder and skip list
spt_roots <- c('C:/Users/fan/R4Econ/amto', 'C:/Users/fan/R4Econ/summarize')
spn_skip <- c('summarize', 'panel', 'support')

# Search and get all Path
ls_sfls <- list.files(path=spt_roots, recursive=T, pattern=".Rmd", full.names=T)

# Skip path if contains words in skip list
if(!missing(spn_skip)) {
  ls_sfls <- ls_sfls[!grepl(paste(spn_skip, collapse = "|"), ls_sfls)]
}

# Loop and print
for (spt_file in ls_sfls) {
  st_fullpath_nosufx <- tail(strsplit(spt_file, "/")[[1]],n=1)
  print(paste0(spt_file, '---', st_fullpath_nosufx))
}

## [1] "C:/Users/fan/R4Econ/amto/array/fs_ary_basics.Rmd---fs_ary_basics.Rmd"
## [1] "C:/Users/fan/R4Econ/amto/array/fs_ary_generate.Rmd---fs_ary_generate.Rmd"
## [1] "C:/Users/fan/R4Econ/amto/array/fs_ary_mesh.Rmd---fs_ary_mesh.Rmd"
## [1] "C:/Users/fan/R4Econ/amto/array/fs_ary_string.Rmd---fs_ary_string.Rmd"
## [1] "C:/Users/fan/R4Econ/amto/array/main.Rmd---main.Rmd"
## [1] "C:/Users/fan/R4Econ/amto/list/fs_lst_basics.Rmd---fs_lst_basics.Rmd"
```

```
## [1] "C:/Users/fan/R4Econ/amto/list/main.Rmd---main.Rmd"
## [1] "C:/Users/fan/R4Econ/amto/main.Rmd---main.Rmd"
## [1] "C:/Users/fan/R4Econ/amto/matrix/fs_mat_generate.Rmd---fs_mat_generate.Rmd"
## [1] "C:/Users/fan/R4Econ/amto/matrix/fs_mat_linear_algebra.Rmd---fs_mat_linear_algebra.Rmd"
## [1] "C:/Users/fan/R4Econ/amto/matrix/main.Rmd---main.Rmd"
## [1] "C:/Users/fan/R4Econ/amto/tibble/fs_tib_basics.Rmd---fs_tib_basics.Rmd"
## [1] "C:/Users/fan/R4Econ/amto/tibble/fs_tib_factors.Rmd---fs_tib_factors.Rmd"
## [1] "C:/Users/fan/R4Econ/amto/tibble/fs_tib_na.Rmd---fs_tib_na.Rmd"
## [1] "C:/Users/fan/R4Econ/amto/tibble/fs_tib_random_draws.Rmd---fs_tib_random_draws.Rmd"
## [1] "C:/Users/fan/R4Econ/amto/tibble/fs_tib_string.Rmd---fs_tib_string.Rmd"
## [1] "C:/Users/fan/R4Econ/amto/tibble/main.Rmd---main.Rmd"
```

1.2 Search and Find all Git Modified or New Rmd

Search inside directories, for all files in a git repo folder that are new or have been modified. Ignore possible subset of file based on string search.

```
# Serch Folder and skip list
spt_roots <- c('C:/Users/fan/R4Econ/amto', 'C:/Users/fan/R4Econ/development')
spn_skip <- c('summarize', 'panel', 'support')
ls_sfls <- list.files(path=spt_roots, recursive=T, pattern=".Rmd", full.names=T)
if(!missing(spn_skip)) {
  ls_sfls <- ls_sfls[!grepl(paste(spn_skip, collapse = "|"), ls_sfls)]
}

# Loop and print
for (spt_file in ls_sfls) {
  spg_check_git_status <- paste0('git status -s ', spt_file)
  st_git_status <- toString(system(spg_check_git_status, intern=TRUE))
  bl_modified <- grepl(' M ', st_git_status, fixed=TRUE)
  bl_aneufile <- grepl('?? ', st_git_status, fixed=TRUE)
  bl_nochange <- (st_git_status == "")

  if (bl_modified == 1) {
    print(paste0('MODIFIED: ', spt_file))
  } else if (bl_aneufile == 1) {
    print(paste0('A NEW FL: ', spt_file))
  } else {
    print(paste0('NO CHNGE: ', spt_file))
  }
}
```

```
## [1] "NO CHNGE: C:/Users/fan/R4Econ/amto/array/fs_ary_basics.Rmd"
## [1] "NO CHNGE: C:/Users/fan/R4Econ/amto/array/fs_ary_generate.Rmd"
## [1] "NO CHNGE: C:/Users/fan/R4Econ/amto/array/fs_ary_mesh.Rmd"
## [1] "MODIFIED: C:/Users/fan/R4Econ/amto/array/fs_ary_string.Rmd"
## [1] "NO CHNGE: C:/Users/fan/R4Econ/amto/array/main.Rmd"
## [1] "MODIFIED: C:/Users/fan/R4Econ/amto/list/fs_lst_basics.Rmd"
## [1] "NO CHNGE: C:/Users/fan/R4Econ/amto/list/main.Rmd"
## [1] "NO CHNGE: C:/Users/fan/R4Econ/amto/main.Rmd"
## [1] "NO CHNGE: C:/Users/fan/R4Econ/amto/matrix/fs_mat_generate.Rmd"
## [1] "NO CHNGE: C:/Users/fan/R4Econ/amto/matrix/fs_mat_linear_algebra.Rmd"
## [1] "NO CHNGE: C:/Users/fan/R4Econ/amto/matrix/main.Rmd"
## [1] "NO CHNGE: C:/Users/fan/R4Econ/amto/tibble/fs_tib_basics.Rmd"
## [1] "NO CHNGE: C:/Users/fan/R4Econ/amto/tibble/fs_tib_factors.Rmd"
```

```
## [1] "NO CHNGE: C:/Users/fan/R4Econ/amto/tibble/fs_tib_na.Rmd"
## [1] "NO CHNGE: C:/Users/fan/R4Econ/amto/tibble/fs_tib_random_draws.Rmd"
## [1] "NO CHNGE: C:/Users/fan/R4Econ/amto/tibble/fs_tib_string.Rmd"
## [1] "NO CHNGE: C:/Users/fan/R4Econ/amto/tibble/main.Rmd"
## [1] "MODIFIED: C:/Users/fan/R4Econ/development/inout/_file/rmd/fs_rmd_pdf_html_mod.Rmd"
## [1] "A NEW FL: C:/Users/fan/R4Econ/development/inout/_file/rmd/fs_rmd_pdf_html_mod_mod.Rmd"
## [1] "MODIFIED: C:/Users/fan/R4Econ/development/inout/_file/rmd/fs_text_save_mod.Rmd"
## [1] "NO CHNGE: C:/Users/fan/R4Econ/development/inout/_file/rmd/main_mod.Rmd"
## [1] "MODIFIED: C:/Users/fan/R4Econ/development/inout/fs_rmd_pdf_html.Rmd"
## [1] "A NEW FL: C:/Users/fan/R4Econ/development/inout/fs_rmd_pdf_html_mod.Rmd"
## [1] "NO CHNGE: C:/Users/fan/R4Econ/development/inout/fs_text_save.Rmd"
## [1] "NO CHNGE: C:/Users/fan/R4Econ/development/inout/main.Rmd"
## [1] "NO CHNGE: C:/Users/fan/R4Econ/development/main.Rmd"
```

1.3 Resave an Existing File with Different Name Different Folder

Given an existing Rmd File, Resave it with a different name (add to name suffix), and then save in a different folder:

- old file: `/R4Econ/development/fs_rmd_pdf_html.Rmd`
- new file: `*R4Econ/development/inout/_file/rmd/fs_rmd_pdf_html_mod.Rmd*`

```
# Serch Folder and skip list
spt_roots <- c('C:/Users/fan/R4Econ/development/inout/')
spn_skip <- c('_main', '_file')
ls_sfls <- list.files(path=spt_roots, recursive=T, pattern=".Rmd", full.names=T)
if(!missing(spn_skip)) {
  ls_sfls <- ls_sfls[!grepl(paste(spn_skip, collapse = "|"), ls_sfls)]
}

# Loop and print
for (spt_file in ls_sfls) {
  spt_new <- paste0('_file/rmd/')
  spn_new <- paste0(spt_new, sub('\\.Rmd$', '', basename(spt_file)), '_mod.Rmd')
  print(spt_new)
  print(spn_new)

  fileConn_rd <- file(spt_file, "r")
  st_file_read <- readLines(fileConn_rd)

  fileConn_sr <- file(spn_new)
  writeLines(st_file_read, fileConn_sr)

  close(fileConn_rd)
  close(fileConn_sr)
}
```

1.3.1 Replacment Function Change Markdown Hierarchy and Add to YAML

Given an existing Rmd File, Resave it with a different name, and replace (add in) additional yaml contents.

```
spn_file = '_file/rmd/fs_rmd_pdf_html_mod.Rmd'
fileConn_sr <- file(spn_file)
st_file <- readLines(fileConn_sr)
# print(st_file)
```

```

st_search <- "html_document:
  toc: true
  number_sections: true
  toc_float:
    collapsed: false
    smooth_scroll: false
    toc_depth: 3
  toc: true
  number_sections: true
  toc_float:
    collapsed: false
    smooth_scroll: false
    toc_depth: 3"
st_replace <- paste0("html_document:
  toc: true
  number_sections: true
  toc_float:
    collapsed: false
    smooth_scroll: false
    toc_depth: 3
  toc: true
  number_sections: true
  toc_float:
    collapsed: false
    smooth_scroll: false
    toc_depth: 3\n",
  "    toc: true\n",
  "    number_sections: true\n",
  "    toc_float:\n",
  "      collapsed: false\n",
  "      smooth_scroll: false\n",
  "      toc_depth: 3")
st_file_updated <- gsub(x = st_file,
  pattern = st_search,
  replacement = st_replace)

st_search <- "../../"
st_replace <- paste0("../..../..../")
st_file_updated <- gsub(x = st_file_updated,
  pattern = st_search,
  replacement = st_replace)

st_file_updated <- gsub(x = st_file_updated, pattern = '# ', replacement = '# ')
st_file_updated <- gsub(x = st_file_updated, pattern = '## ', replacement = '## ')
st_file_updated <- gsub(x = st_file_updated, pattern = '# ', replacement = '# ')

spn_file = '_file/rmd/fs_rmd_pdf_html_mod.Rmd'
fileConn_sr <- file(spn_file)
st_file <- writeLines(st_file_updated, fileConn_sr)

```

1.4 Search and Render Rmd File and Save HTML, PDF or R

1. Search files satisfying conditions in a folder

2. knit files to HTML (and re-run the contents of the file)
3. Save output to a different folder

```
# Specify Parameters
ar_spt_root = c('C:/Users/fan/R4Econ/amto/array/', 'C:/Users/fan/R4Econ/math/integration')
bl_recursive = TRUE
st_rmd_suffix_pattern = "*.Rmd"
ar_spn_skip <- c('basics', 'integrate', 'main', 'mesh')
ls_bool_convert <- list(bl_pdf=TRUE, bl_html=TRUE, bl_R=TRUE)
spt_out_directory <- 'C:/Users/fan/Downloads/_data'
bl_verbose <- TRUE

# Get Path
ls_sfls <- list.files(path=ar_spt_root,
                      recursive=bl_recursive,
                      pattern=st_rmd_suffix_pattern,
                      full.names=T)

# Exclude Some Files given ar_spn_skip
if(!missing(ar_spn_skip)) {
  ls_sfls <- ls_sfls[!grepl(paste(ar_spn_skip, collapse = "|"), ls_sfls)]
}

# Loop over files
for (spn_file in ls_sfls) {

  # Parse File Name
  spt_file <- dirname(spn_file)
  sna_file <- tools::file_path_sans_ext(basename(spn_file))

  # Output Files
  spn_file_pdf <- paste0(spt_file, sna_file, '.pdf')
  spn_file_html <- paste0(spt_file, sna_file, '.html')
  spn_file_R <- paste0(spt_file, sna_file, '.R')

  # render to PDF
  if (ls_bool_convert$bl_pdf) {
    if (bl_verbose) message(paste0('spn_file_pdf:', spn_file_pdf, ', PDF started'))
    rmarkdown::render(spn_file, output_format='pdf_document',
                      output_dir = spt_out_directory, output_file = sna_file)
    if (bl_verbose) message(paste0('spn_file_pdf:', spn_file_pdf, ', PDF finished'))
    spn_pdf_generated <- paste0(spt_out_directory, '/', spn_file_pdf)
  }

  # render to HTML
  if (ls_bool_convert$bl_html) {
    if (bl_verbose) message(paste0('spn_file_html:', spn_file_html, ', HTML started.'))
    rmarkdown::render(spn_file, output_format='html_document',
                      output_dir = spt_out_directory, output_file = sna_file)
    if (bl_verbose) message(paste0('spn_file_html:', spn_file_html, ', HTML finished.'))
    spn_html_generated <- paste0(spt_out_directory, '/', spn_file_html)
  }

  # purl to R
```

```

if (ls_bool_convert$bl_R) {
  if (bl_verbose) message(paste0('purl_to:', paste0(spn_file_R, ".R")))
  knitr::purl(spn_file, output=paste0(spt_out_directory, '/', sna_file, '.R'), documentation = 1)
  spn_R_generated <- paste0(spt_out_directory, '/', sna_file, '.R')
}

# return(list(ls_spt_pdf_generated=ls_spt_pdf_generated,
#             ls_spt_html_generated=ls_spt_html_generated,
#             ls_spt_R_generated=ls_spt_R_generated))
}

```