Productivity Hacks with your .Rprofile

SatRday Johannesburg

Cesaire Tobias | cesaire@fractalva.com

Fractal Value Advisors

07 March 2020

Overview

1. What is a .Rprofile?
2. Where does it live?
3. What do you do with it?
4. Conclusion
5. Useful links for further research

What is a .Rprofile?

- A script that is executed at startup
- Allows you to customise your setup
- You may have multiple .Rprofiles
    - Different profiles for different projects
    - Multiple profile scripts in a single .Rprofile.d directory

Where does your .Rprofile live?

At startup R searches for your .Rprofile in the following locations:

1. R_HOME, the directory in which R is installed. Find using: `R.home()`
2. HOME, your home directory. Find using: `path.expand("~")`
3. Your current working directory. Find using: `getwd()`

- Only one profile is loaded per session.
    - .Rprofile file in your current wd > HOME > R_HOME
- If you don't have a .Rprofile, you can create one with `file.create()` or simply use a text editor like Notepad++

What I use my .Rprofile for

- To store a hodge-podge collection of functions that haven't made their way to a package.
- These functions are run interactively and are intended to streamline any repetitive tasks that arise in my workflow.
- These functions are most often written in base R however there are a few exceptions.
- Do not put things in your .Rprofile that limit the reproducibility or portability of your code.
    - e.g. reading in data or loading packages.

Examples

Dataframes

```r
# ht==headtail. show the first and last 10, by default, rows of a table
ht <- function(x, n = 10) rbind(head(x, n), tail(x, n))

# lr==leftright. show the first and last 3, by default columns of a table
lr <- function(x, n = 3) {
    maxCol <- max(col(x))
    lastCols <- maxCol
    for (i in 1:n) {
        lastCols <- append(lastCols, maxCol - i)
    }
    cbind(x[, c(1:n)], x[, lastCols[n:1]])
}

# Return rows that contain NAs
na_rows <- function(df) {
    contains_na <- apply(df, 1, function(rowX) {
        any(is.na(rowX))
    })
    rv <- df[contains_na, ]
    return(rv)
}
```

Utility

```r
# run all .R files in a folder
runR <- function(folderName = "./R", verbose = TRUE, showWarnings = TRUE) {

    files <- list.files(folderName, full.names = TRUE)

    # get only R files
    files <- files[grepl("\\.[rR]$", files)]

    if (!length(files) && showWarnings)
        warning("No R files in ", folderName)

    for (f in files) {
        if (verbose)
            cat("sourcing: ", f, "\n")
        try(source(f, local = FALSE, echo = FALSE), silent != verbose)
    }

}
```

Package Development

```r
# switch between a script and its test file
oof <- function() {
    file <- devtools:::find_active_file()
    is_source_file <- basename(dirname(file)) == "R"
    has_r_ext <- grepl("\\.[rR]$", file)
    if (any(!has_r_ext)) {
        stop("file(s): ", paste0("'", file[!has_r_ext], "'", collapse = ", "),
            " are not R files", call. = FALSE)
    }
    base_file <- basename(file)
    if (is_source_file) {
        new_file_name <- glue::glue("test-{base_file}")
        new_path <- usethis::proj_path("tests", "testthat", new_file_name)
    } else {
        new_file_name <- sub("^test-?", "", base_file)
        new_path <- usethis::proj_path("R", new_file_name)
    }

    usethis::edit_file(new_path)
}
```

Conclusion

- Your .Rprofile is a great tool to help mitigate repetitive, mundane processes in your workflow.
- As a rule of thumb try not to include anything that won't be run interactively.

Bonus

Snippets

Code snippets are text macros that are used for quickly inserting common snippets of code You can also run R code in your snippet. Use ‘r expr anywhere in your snippet Paste Microsoft paths with forward slashes instead of backslashes

```
snippet pastePath
"`r gsub("\\\\", "/", readClipboard())`"
```

Useful Links

- What is an Rprofile
- Efficient R Programming
- How to Pimp Your .Rprofile
- Understanding R's Startup
- Friendly R Startup Configuration
- What They Forgot To Teach You About R - CH.7
- 4 Ways to be more Efficient with Snippets