R Shiny Training Resources

Jimmy Briggs

2020-06-04

Contents

Chapter 1

Introduction

1.0.1 Contents

- Package HomePage
- R Setup Guide
- R Shiny Training Resources
- List of Helpful Bookdowns

Package {rtraining}: R Training Resources, Guides, Tips, and Knowledge Base. Current version is 0.0.1.

The goal of `rtraining` is to provide useful resources, knowledge, and walkthroughs for new R developers.

The package is split into three main areas:

1. R Setup and Configuration: a thorough walkthrough for setting up and configuring R, RStudio, and various other software in an efficient manner.
2. R Workflows: example workflows showcasing the main types of work done with R, including, but not limited to R Data Analysis Projects, Reporting with RMarkdown, R Shiny Applications, R Package Development.
3. R Tips & Tricks: General tips and tricks learned over time from my experiences with R.

1.1 Installation

You can install the released version of rtraining from CRAN with:

```
install.packages("rtraining")
```

And the development version from GitHub with:

```
# install.packages("devtools")
devtools::install_github("jimbrig/rtraining")
```

The list of dependencies required to install this package is: attachment, chameleon, devtools, knitr, pkgdown, rmarkdown, roxygen2, utils, xfun.

To install the package, you can run the following script

```
# install.packages("remotes")
remotes::install_local(path = "rtraining_0.0.1.tar.gz")
```

In case something went wrong, you may want to install dependencies before using:

```
# Remotes ----
install.packages("remotes")
remotes::install_github('ThinkR-open/chameleon')
# Attachments ----
to_install <- c("xfun")
  for (i in to_install) {
    message(paste("looking for ", i))
    if (!requireNamespace(i)) {
      message(paste("     installing", i))
      install.packages(i)
    }
  }
```

Once you have the package installed you can open the website directly by running:

```
library(rtraining)
rtraining::open_pkgdown()
```

Similarly you can open the package `bookdown` with:

```
rtraining::open_guide()
```

Chapter 2

R Bookdowns List

Chapter 3

R Setup Guide

Author: Jimmy Briggs
Date: June 2, 2020

The purpose of this guide is to provide an in-depth walkthrough of the various tasks, installations, and processes involved with setting up your machine for using R.

3.1 Installing R & RStudio

The first step is to install R and RStudio from their respective websites listed below.

3.2 Download R

Install R from the main CRAN (Comprehensive R Archive Network) website: https://cran.r-project.org/.

- On the page, select "Download R for Windows" > "Base" > "Download R 4.0.0 for Windows"
- Note as of today the latest R version is 4.0.0
- During installation, ensure that you are installing the 64-bit architecture version of R for increased RAM.

3.2.1 Additional Notes on R

- CRAN is composed of a set of mirror servers distributed around the world and is used to distribute R and R packages.
- Don't try and pick a mirror that's close to you: instead use the cloud mirror, https://cloud.r-project.org, which automatically figures it out for you.
- A new major version of R comes out once a year, and there are 2-3 minor releases each year. It's a good idea to update regularly.
- Upgrading can be a bit of a hassle, especially for major versions, which require you to re-install all your packages, but putting it off only makes it worse.

For more details see the section discussing how to efficiently migrate R packages between versions in this guide.

- To ease the process up updating your R version it is helpful to use the `installr` packge via `installr::updateR()` (run this from the native R console not RStudio).

3.3 Download RStudio

Install the free Version of RStudio Desktop for Windows from the RStudio Website here: https://rstudio.com/products/rstudio/download/.

3.3.1 Additional Notes on RStudio

- RStudio is an integrated development environment, or IDE, for R programming.
- RStudio is updated a couple of times a year. When a new version is available, RStudio will let you know.
- It's a good idea to upgrade regularly so you can take advantage of the latest and greatest features.

3.4 Configure RStudio Settings

A range of Project Options and Global Options are available in RStudio from the Tools menu (accessible from the keyboard via Alt+T).

Most of these are self-explanatory but it is worth mentioning a few that can boost your programming efficiency:

- I highly recommend unticking the default "Restore .RData" settings box:

Unticking this default prevents loading previously created R objects. This will make starting R quicker and reduce the chance of getting bugs due to previously created objects. For this reason, I recommend you untick this box.

Alternatively you can simply run this code:

```
require(usethis)
usethis::use_blank_slate(scope = "user")
```

See `code{?usethis::use_blank_slate` for more information.

- GIT/SVN project settings allow RStudio to provide a graphical interface to your version control system.
- R version settings allow RStudio to 'point' to different R versions/interpreters, which may be faster for some projects.
- Code editing options can make RStudio adapt to your coding style, for example, by preventing the autocompletion of braces, which some experienced programmers may find annoying. Enabling Vim mode makes RStudio act as a (partial) Vim emulator.
- Diagnostic settings can make RStudio more efficient by adding additional diagnostics or by removing diagnostics if they are slowing down your work. This may be an issue for people using RStudio to analyze large datasets on older low-spec computers.
- Appearance: if you are struggling to see the source code, changing the default font size may make you a more efficient programmer by reducing the time overheads associated with squinting at the screen. Other options in this area relate more to aesthetics. Settings such as font type and background color are also important because feeling comfortable in your programming environment can boost productivity. Go to Tools > Global Options to modify these.

3.5 Installing and Setting Up Additional Software and Utilities

To get the most out of R and RStudio it is helpful to install these additional software resources:

- RTools
- Git
- A Git Client GUI (or just use RStudio)
- Git LFS
- Mercurial

- Tex Distribution
  - Tinytex
  - Miketex
  - TexLive
- Java
- Pandoc
- Node.js
- Hugo
- Inno

Here's the code to install these additional resources:

```
if (!require(pacman)) install.packages("pacman")

pacman::p_load(devtools,
               installr,
               tinytex,
               rstudioapi,
               magrittr,
               dplyr,
               pkgbuild)

# configure RStudio settings ----------------------------------------------

# disable reloading of workspace between sessions
usethis::use_blank_slate(scope = "user")

# review system environment variables:
Sys.getenv()

# configure your R library path for R packages
.libPaths()

# copy packages to new R-version's windows library
libdir_prior <- file.path("<enter prior win-library path here>")
libdir_current <- file.path("<enter current win-library path here>")
installr::copy.packages.between.libraries(
  from = libdir_prior, to = libdir_current
)

# check
.libPaths()[1] == libdir
```

```
# configure dotfiles .Rprofile & .Renvrion -------------------------------

# review dotfiles
usethis::edit_r_environ(scope = "user") # (RTools Path, github PAT, keys, etc.)
usethis::edit_r_profile(scope = "user") # (various options for packages)

# additional software ----------------------------------------
pkgbuild::setup_rtools()

# Rtools
installr::install.rtools()
rstudioapi::restartSession()

# git
installr::install.git()
rstudioapi::restartSession()

# tinytex
tinytex::install_tinytex()
rstudioapi::restartSession()
tinytex::use_tinytex()

# java
installr::install.java()

# pandoc
installr::install.pandoc()

# node.js (only if desired)
installr::install.nodejs()

# github Git Client (only if desired)
installr::install.github()

# inno (only if desired)
installr::install.inno()
```

Note that not all of these are required, however, I recommend at a minimum to install RTools, Git, Pandoc, and a Latex service.

I have provided an R script named R-Setup-Script.R with this guide that uses the `installr` package to help ease the process of installing all these extra resources.

3.6 Advanced Configuration

This section discusses more advanced R related configurations such as:

- Environment Paths
- Detailed System Information
- Dotfiles
- Common pitfalls

For more advanced R developers you may want to further configure your development environement by customizing you R related dotfiles; specifically, your *.Rprofile* and *.Renviron*.

Here is what my minimal setup includes:

Additionally, you can configure keybinding for RStudio addins from RStudio and store them within the .R folder located in your *R_USER* path. To view this path run `Sys.getenv("R_USER")`.

On a windows computer, you may need to adjust where the system chooses to look for various R related items on your system. For example, the path to your RTools bin executable, your HOME path, your library path for packages, and many other windows specific paths. Note the difference between SYSTEM PATHS and USER PATHS.

To add your system RTOOLS PATH to your .Renviron (easier than manually configuring within windows system settings) run the code:

```
cat('PATH = ${RTOOLS40_HOME}\\usr\\bin;${PATH}',
    file = fs::path(Sys.getenv("R_USER"), "/.Renviron"),
    append = TRUE)
```

YOu can also view the allocated memory RAM your machine allows R to use by running `forgot the function...`

Chapter 4

Getting Started with R and Shiny

Below is a collection of resources for getting started in R and Shiny.

They are listed roughly in the order that I would expect a motivated beginner to work through them.

4.0.1 Getting Stated with R

- Download R here: https://cran.r-project.org/
- Download the free version of Rstudio here: https://www.rstudio.com/products/rstudio/download/
- Getting Started with R: https://rladiessydney.org/courses/ryouwithme/
- R4DS: https://r4ds.had.co.nz/
  - note: I recommend skimming through the above book and trying out examples. Skip over sections that you do not understand, and you can come back to them later.

For a more in-depth walkthrough of setting up R see the R Setup Guide vignette

4.0.2 Getting Started with GitHub

- Create a GitHub account: https://github.com/ . You may need to install git first from here: https://git-scm.com/book/en/v2/Getting-Started-Installing-Git
- If you are not familiar with git, this article gives a nice overview: https://jahya.net/blog/git-vs-github/
- Try out example 1 from this tutorial: https://github.com/bcgov/bcgov-data-science-resources/wiki/Tutorial:-Intro-to-Git-&-GitHub-for-the-R-User

4.0.3 A Deeper Dive into R

- Fundamental chapters in Advanced R
  - http://adv-r.had.co.nz/Data-structures.html
  - http://adv-r.had.co.nz/Subsetting.html
  - http://adv-r.had.co.nz/Style.html
- Introduction to the dplyr package for data manipulation: https://cran.r-project.org/web/packages/dplyr/vignettes/dplyr.html

4.0.4 Getting Started with Shiny

What is Shiny from Rstudio: > Shiny is an open source R package that provides an elegant and powerful web framework for building web applications using R. Shiny helps you turn your analyses into interactive web applications without requiring HTML, CSS, or JavaScript knowledge.

- Into to Shiny book: https://laderast.github.io/gradual_shiny/
- Another Intro to Shiny book: https://ourcodingclub.github.io/2017/03/07/shiny.html
- Video on the basics of web development: https://www.youtube.com/watch?v=FXqTHsPaY0A
- Shiny basics: https://shiny.rstudio.com/articles/basics.html
- More Shiny basics: https://deanattali.com/blog/building-shiny-apps-tutorial/ (Dean Attali, the author of this tutorial is one of my clients)
- Articles on Shiny: https://shiny.rstudio.com/articles/

4.0.5 Build your own Shiny app and put it on GitHub!

4.0.6 Full Books on Shiny

- Mastering Shiny: https://github.com/hadley/mastering-shiny
- Engineering Production-Grade Shiny Apps: https://thinkr-open.github.io/building-shiny-apps-workflow/index.html

4.0.7 Want More?

Here's some more resources gathered from a colleagues recent RShiny learning path:

- basic overview of R in two hour video: https://www.youtube.com/watch?v=_V8eKsto3Ug&list=PLWKjhJtqVAblQe2CCWqV4Zy3LY01Z8aF1&index=5&t=6791s
- basic overview of shiny web apps(I found it particularly helpful to follow along with the "https://shiny.rstudio.com/images/shiny-cheatsheet.pdf"): https://vimeo.com/rstudioinc/review/131218530/212d8a5a7a/#t=43m32s
- overview of rstudio IDE and github interactions Part 1 and Part 2 -

  - https://resources.rstudio.com/wistia-rstudio-essentials-2/rstudioessentialsmanagingpart1-2
  - https://resources.rstudio.com/wistia-rstudio-essentials-2/rstudioessentialsmanagingpart2-2

- useful R for data science chapters:
  - workflow:basics link: https://r4ds.had.co.nz/workflow-basics.html
  - workflow:projects link: https://r4ds.had.co.nz/workflow-projects.html
  - tibbles link: https://r4ds.had.co.nz/tibbles.html
  - tidy data link: https://r4ds.had.co.nz/tidy-data.html
  - functions link: https://r4ds.had.co.nz/functions.html
  - vectors link:https://r4ds.had.co.nz/vectors.html
  - iterations with purr link: https://r4ds.had.co.nz/iteration.html
- learn about dplyr package for data manipulation: https://cran.r-project.org/web/packages/dplyr/vignettes/dplyr.html
- learn about shiny dashboards: http://rstudio.github.io/shinydashboard/
- list of cheats sheats and resources found here: https://resources.rstudio.com/
- highcharter API for rich graphics and charts: http://jkunst.com/highcharter/highcharts-api.html
- Article on Databases: https://shiny.rstudio.com/articles/persistent-data-storage.html

Built in tutorials for packages can be accessed through the learnr package. For example if you library learnr and tidyverse, the top right panel of RStudio can display a walk through of different functions in the tidyverse package. Blog post describing package: https://blog.rstudio.com/2020/02/25/rstudio-1-3-integrated-tutorials/