Data Pipelines: Using a Model-View-Controller Framework in R

Brandon Rose and Natalie Goulett

12/2024

Contents

Chapter 1

What is {RosyREDCap}?

[{RosyREDCap}]

R and REDCap are both widely utilized in medicine, including basic science, clinical research, and clinal trials. Both have independent strengths, but together they can create powerful data pipelines. While several R packages exist for extracting data using the REDCap API, {RosyREDCap} stands out by offering comprehensive extraction of all metadata and data from any REDCap project into a standardized R list object, facilitating effortless downstream use in exports, imports, transformation, and applications. Three core aims of {RosyREDCap} are to

1. Maintain a local version of the database (DB) object by only calling recently updated records using the REDCap log.
2. Allow imports of non-coded versions of the dataset using R or Excel/CSV.
3. Launch a shiny app that allows you to explore all of your REDCap projects.

By leveraging the combined strengths of R and REDCap, users can maintain strong clinical data pipelines, collected and processed appropriately to improve research and patient care. RosyREDCap can be used as a base data object and data quality tool for most REDCap projects to aid in collection, monitoring, transformation, and analysis.

1.1   Installing RosyREDCap

Note: The current version of {RosyREDCap} used when writing this book is 1.0.0.9030, and some of the features presented in this book might not be available

if you are using an older version, or be a little bit different if you have a newer version. Feel free to browse the package NEWS.

The stable release can be found on CRAN and installed with:

```r
install.packages("RosyREDCap")
```

You can install the development version of RosyREDCap from GitHub by using the {remotes} package. Be sure to install {remotes} if you don't have it already.

```r
remotes::install_github("brandonerose/RosyREDCap")
```

Note that the version used at the time of writing this book is 1.0.0.9030. You can check what version you have installed with the following.

```r
packageVersion("RosyREDCap")
#> [1] '1.0.0.9030'
```

The motivation behind {RosyREDCap} is that building a proof-of-concept application is easy, but **things change when the application becomes larger and more complex, and especially when you need to send that app to production**. Until recently there has not been any real framework for building and deploying production-grade {shiny} apps. This is where {RosyREDCap} comes into play: **offering {shiny} developers a toolkit for making a stable, easy-to-maintain, and robust production web application with R**. {RosyREDCap} has been developed to abstract away the most common engineering tasks (for example, module creation, addition and linking of an external CSS or JavaScript file, etc.), so you can focus on what matters: building the application. Once your application is ready to be deployed, {RosyREDCap} guides you through testing and brings tools for deploying to common platforms.

If you have any issues, try downloading the most recent version of R at RStudtio and update all packages in RStudio. See thecodingdocs.com/r/getting-started.

Chapter 2

Intended Uses

RosyREDCap is for all three types of R users below (basic, intermediate, and advanced). For each type there is an *intended* way of using the package, which may evolve over time.

The Basic R User

A basic R user is capable of installing R and RStudio on their computer. They may be very hesitant working through any errors or understanding "how it all works". However, they know how run some lines of code if the process is straight forward.

The Intermediate R User

An intermediate user knows how to create an R project and even has some scripts they have written themselves somewhere on their computer. The haven't gone down many rabbit holes but they know to make some ggplots and are capable of working through most common errors.

The Advanced R User

An advanced R user understands how R packages work. They can write reusable functions and do complex tasks that may include API calls, data transformations, working with SQL and more.