

Why Use R and What Kinds of Apps Can I Make?

Dan Holmes AACC 2021, Session 35104

30-Sep-2021

Why use R

R is a fully fledged programming language. Isn't that kind of overkill? In the session we will discuss the motivations for the use of the R-language over and above a GUI tool like Tableau. In a nutshell however, the reasons for R can be summarized as follows:

- Easier than many programming languages.
- Particularly well-suited to tabular data as is created in lab medicine.
- Massive user and contributor base who have coded up solutions to just about every manner of statistical analysis you can imagine.
- High level tools for rapid, visually stunning data visualization out of the box – especially ggplot.
- Tools for report generation in MSWord, MSEXcel, PDF and HTML.
- Tools for rapid creation and deployment of interactive reports.
- An easy pathway to full-fledged, scalable, commercial grade interactive web-apps using Shiny

R Markdown

Markdown is a markup language used for the facile generation of output in HTML, PDF or MS Word output. It uses simple syntax for creating sections, subsections, paragraphs, bulleted and numbered lists and tables. It also permits the incorporation of mathematics adopting the syntax of LaTeX.

RMarkdown can be thought of as a mash-up of R and Markdown, incorporating the syntax of Markdown but permitting the incorporation of R to generate statistical results, figures and tables.

While PDF and MSWord are not apps *per se*, they are reports and I have previously written some articles for previous meetings to explain this process. Here is the content I prepared for the 2019 meeting with Shannon Haymond about writing RMarkdown documents with PDF or MS Word output and here is a pre-print about writing a journal article in a reproducible format using RMarkdown.

If you want a fairly gentle introduction to writing documents in RMarkdown, you can go [here](#).

HTML Widgets

When we speak of apps in this context, we are specifically speaking of web-apps and we can use R to make interactive HTML reports in three (increasingly complex) ways which we will demonstrate. The first way is to embed and interactive widget into your HTML presentation. There are many html widget packages available for R—so many that it would be easy to get overwhelmed. For this reason, we want to highlight a few that are particularly useful.

- plotly - plotly provides an easy set of tools for turning regular R plots into an interactive experience.
- highcharter - a little more complicated than plotly but able to produce more sophisticated renderings.
- DataTables - for making interactive tables
- Leaflet - for interactive maps
- Dygraphs - for presenting timeseries data

Once you have made an HTML page with some interactive graphics, you may want to move to something that looks more like an app. This is where the Flexdashboard package lowers the activation energy.

flexdashboard

The Flexdashboard package allows one to create encapsulated HTML interactive dashboards from an RMarkdown document. The output can be uploaded to the web immediately as .html file and is ready for interaction. Here's a simple example that I made for COVID testing which updates itself hourly. However, without the use of the Shiny package, these dashboards do have limitations. Be not dismayed: many routine applications they are perfectly suited for a simple flexdashbaord.

Here are some excellent flexdashboard examples using the htmlwidgets mentioned in the previous section.

There are number of good introductory tutorials online:

- Rstudio's flexdashboard article
- Yihui Xie's chapter on dashboards from his book.
- Thierry Warren's chapter dashboards

shinydashboard

The shinydashboard package permits the development of web-apps using the Shiny web application framework for R. If you want interactivity like drop-down menus, sliders, radio buttons, live, editable tables, interaction with a database etc, you are going to need to use the Shiny package under the hood. The shinydashboard package simplifies the process to app creation using Shiny. The limitation is that if you want to deploy your app for other to use, you will need a Linux server or you will have to subscribe to commercial services available from RStudio. It's not too hard to deploy Shiny apps on Amazon Web Services. You can host a Shiny app on RStudio's free-use servers provided it contains no sensitive information and does not see heavy web traffic. This app for method comparison was made by my friend Burak Bahar. It's a nice example of a full fledged Shiny app.

Examples

In our session we will build and modify three examples from templates. All code will be hosted at my github repository so you can download and modify them for yourself.