

All Training materials are provided "as is" and without warranty and RStudio disclaims any and all express and implied warranties including without limitation the implied warranties of title, fitness for a particular purpose, merchantability and noninfringement.

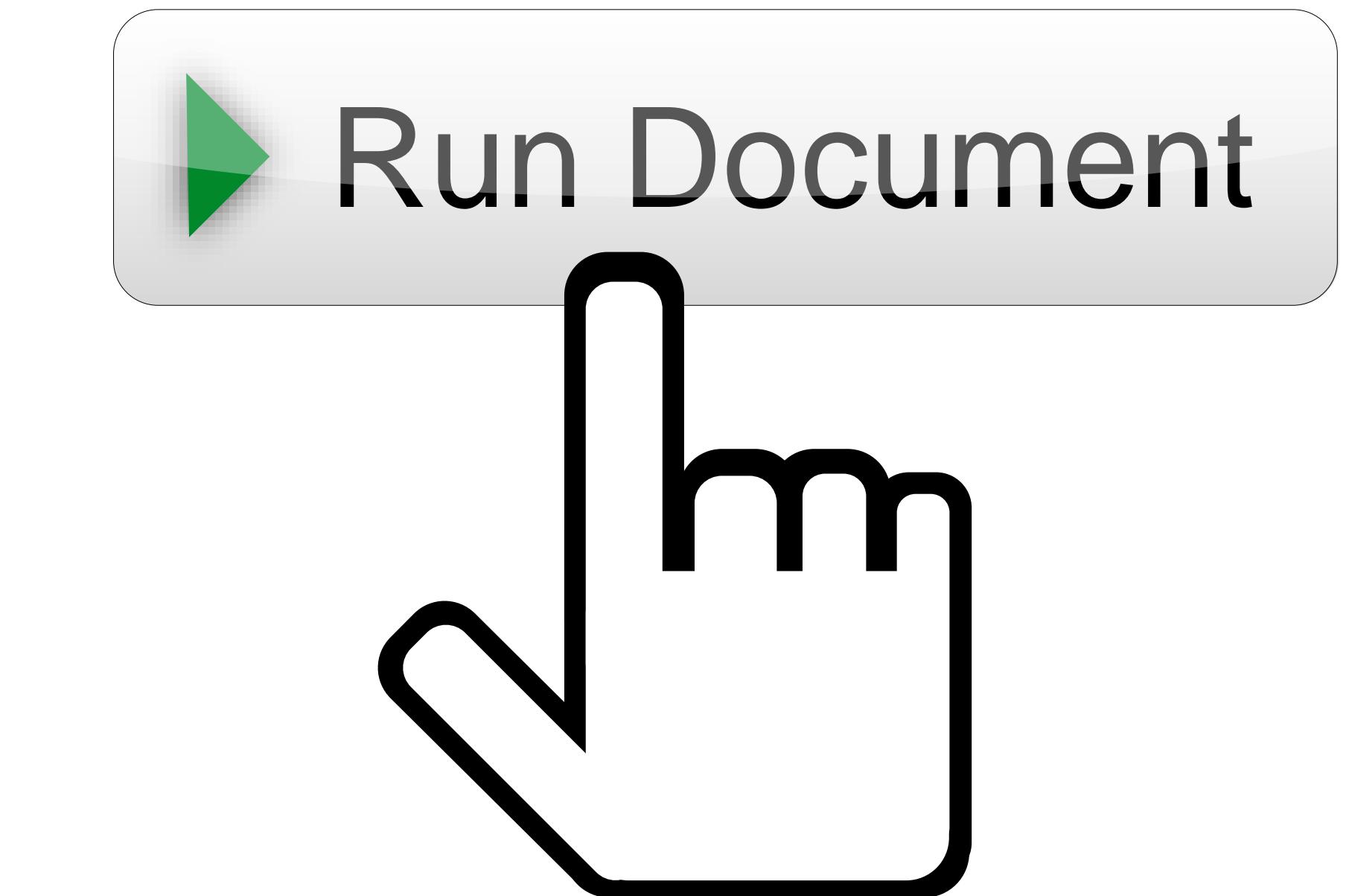
The Training Materials are licensed under the Creative Commons Attribution-Noncommercial 3.0 United States License. To view a copy of this license, visit <http://creativecommons.org/licenses/by-nc/3.0/us/> or send a letter to Creative Commons, 171 Second Street, Suite 300, San Francisco, California, 94105, USA.

Welcome to

Speaking Analytics at MBSW :

An Introduction to Shiny, R Markdown & HTML

Widgets for R with Application in Drug Development



Philip Bowsher
Customer Success
March 2016
[Email: phil@rstudio.com](mailto:phil@rstudio.com)



Schedule

9:00 - 9:15

RStudio

9:15 - 10:00

Intro to Shiny

10:05 - 10:55

Intro to R Markdown

11:00 - 11:30

Intro to HTML Widgets for R

11:40 - 12:30

Shiny Drug Dev. Apps

12:30 - 12:45

Q/A - Chat

12:45 - 1:00

Wrap-up

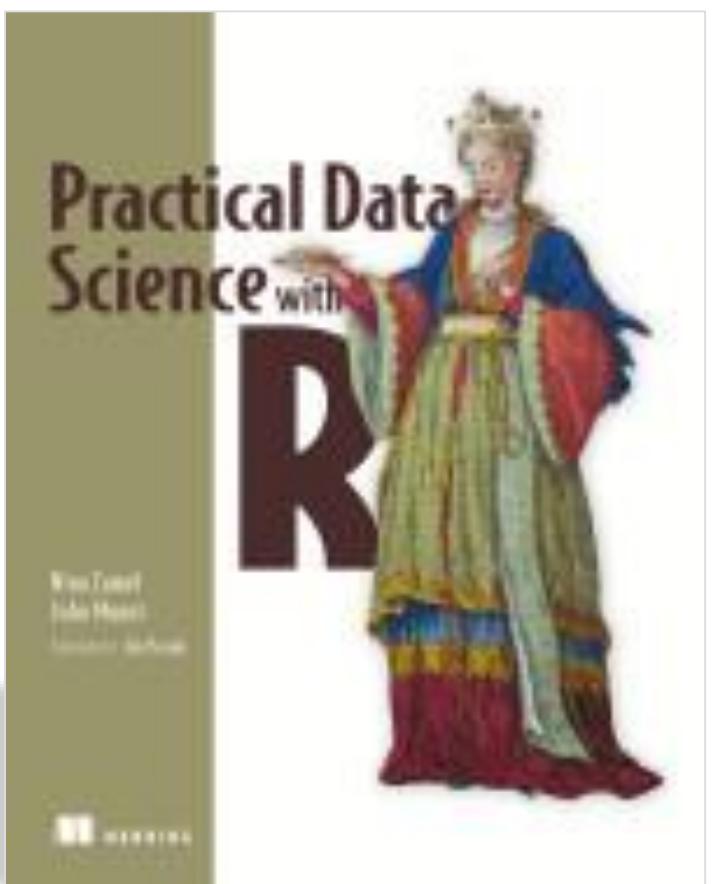
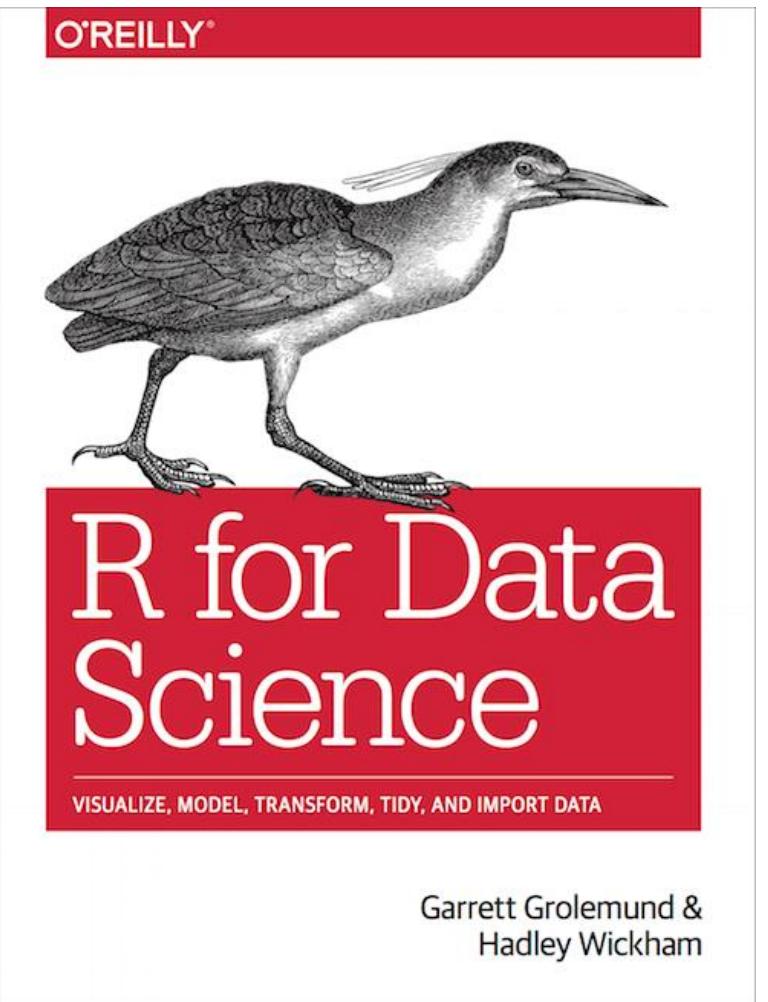
What is Data Science:

“Data science is an exciting discipline that allows you to turn raw data into understanding, insight, and knowledge. The goal of ‘R for Data Science’ is to introduce you to the most important in R tools that you need to do data science.”

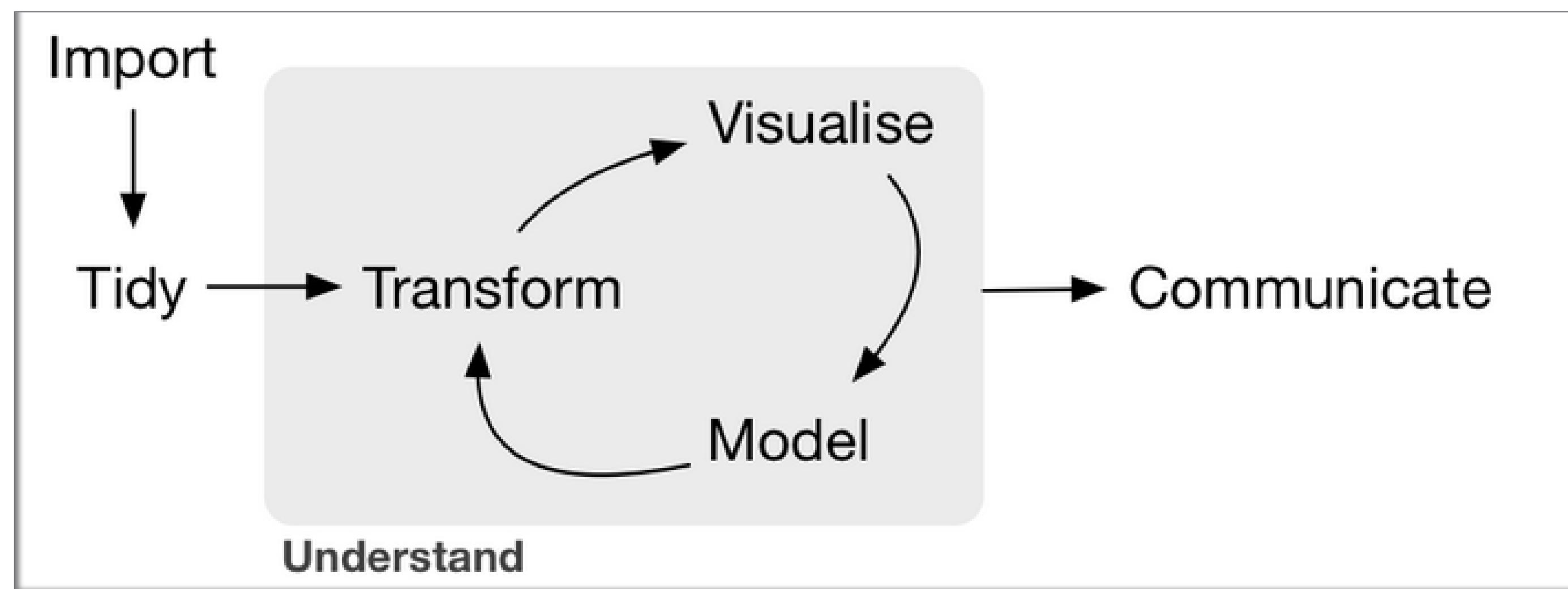
<http://r4ds.had.co.nz/intro.html>

“Data science is a term I use to represent the ownership and management of the entire modeling process: discovering the true business need, collecting data, managing data, building models and deploying models into production.”

<http://www.win-vector.com/blog/2013/04/data-science-machine-learning-and-statistics-what-is-in-a-name/>



Model of the Tools Needed in a Typical Data Science Project :



<http://r4ds.had.co.nz/intro.html>

Developing Data Products

Johns Hopkins University

Part of a 10-course series, the [Data Science Specialization](#)

The Data Scientist's Toolbox

Johns Hopkins University

Part of a 10-course series, the [Data Science Specialization](#)

edX Courses ▾ How It Works ▾ Schools & Partners About ▾ I want to learn about... Sign In Register

 **Data Analysis for Life Sciences 1: Statistics and R**

An introduction to basic statistical concepts and R programming skills necessary for analyzing data in the life sciences.



About this course 11 Reviews 4/5 ★★★★☆

We will learn the basics of statistical inference in order to understand and compute p-values and confidence intervals, all while analyzing data with R. We provide R programming examples in a way that will help make the connection between concepts and implementation. Problem sets requiring R programming will be used to test understanding and ability to implement basic data analyses. We will

Self-Paced **Enroll Now**

I would like to receive email from Harvard University and learn about its other programs.

Length: 4 weeks
Effort: 2-4 hours / week
Price: FREE Add a Verified Certificate for \$50

Reproducible Research

Johns Hopkins University

Part of a 10-course series, the [Data Science Specialization](#)

R Programming

Johns Hopkins University

Part of a 10-course series, the [Data Science Specialization](#)



DataCamp



Reporting with R Markdown



Data Visualization in R with ggvis



Data Manipulation in R with dplyr

What is R?

- Open source software environment for statistical computing and graphics
- Created over 20 years ago by two statistics professors in New Zealand
- Based on S, a similar environment originally developed at AT&T Bell Labs



Why R?

- It's free, open source, and available on every major platform. As a result, if you do your analysis in R, anyone can easily replicate it.
- A massive set of packages for statistical modelling, machine learning, visualization, and importing and manipulating data.
- Cutting edge tools - a powerful and flexible toolkit which allows you to write concise yet descriptive code.
- A fantastic community.
- Powerful tools for communicating your results...Data Products



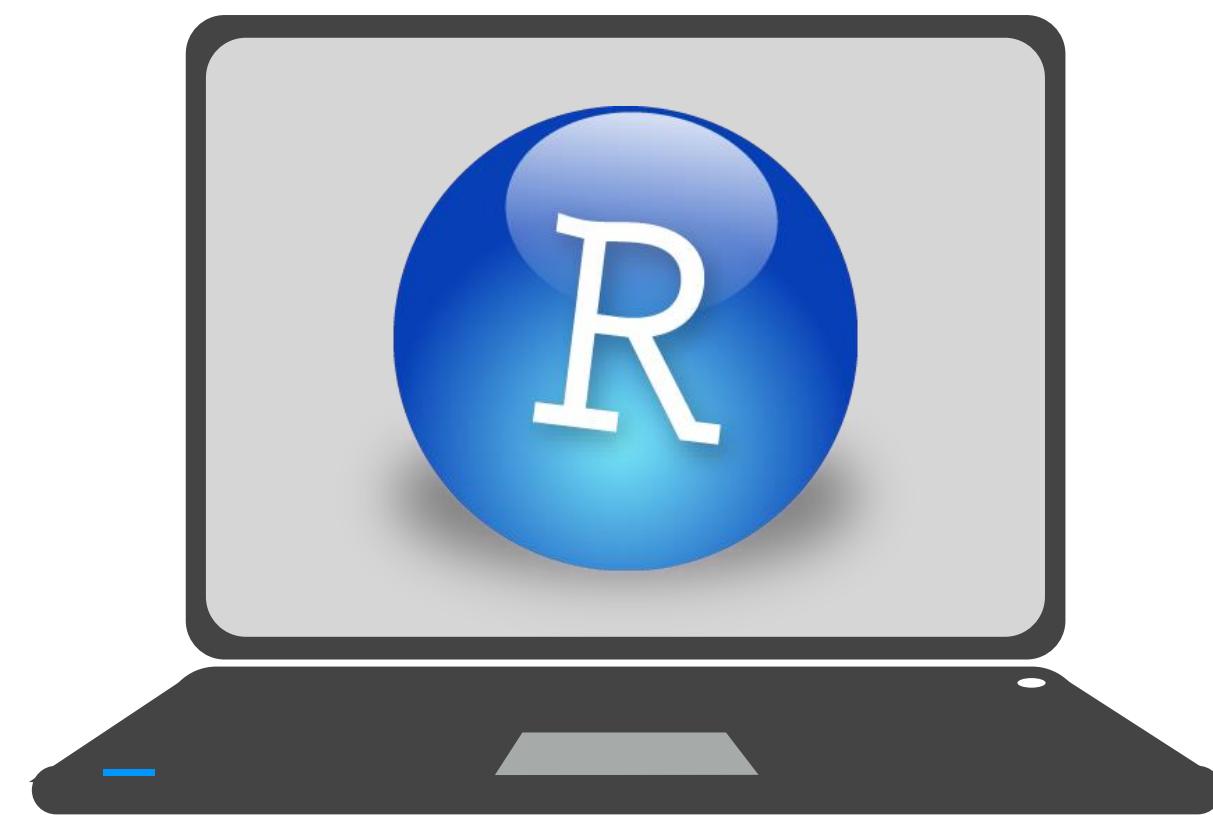
The Mission of Rstudio is...

to provide the most widely used open source and enterprise-ready professional software for the R statistical computing environment.

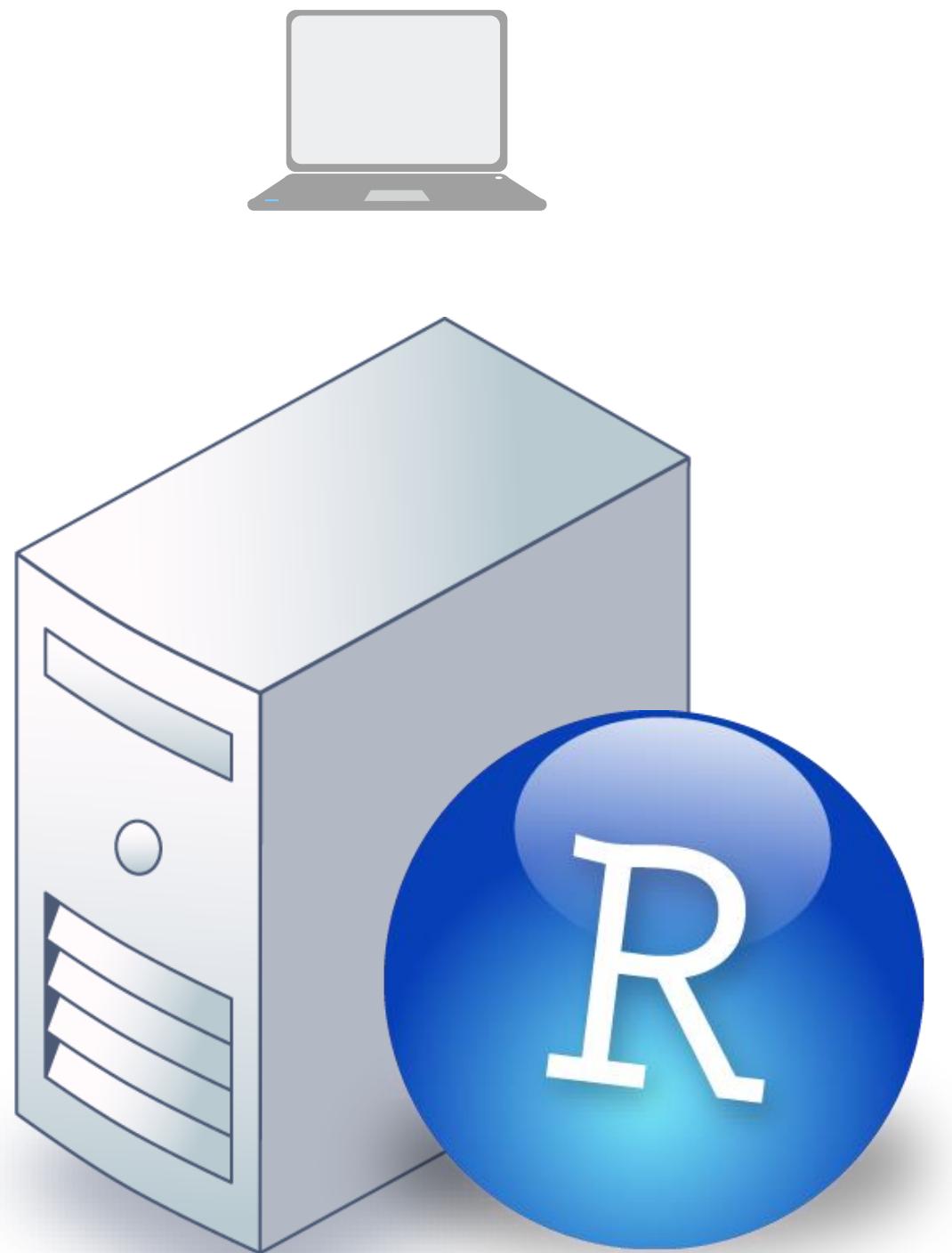


These tools further the cause of equipping everyone, regardless of means, to participate in a global economy that increasingly rewards **data literacy**.

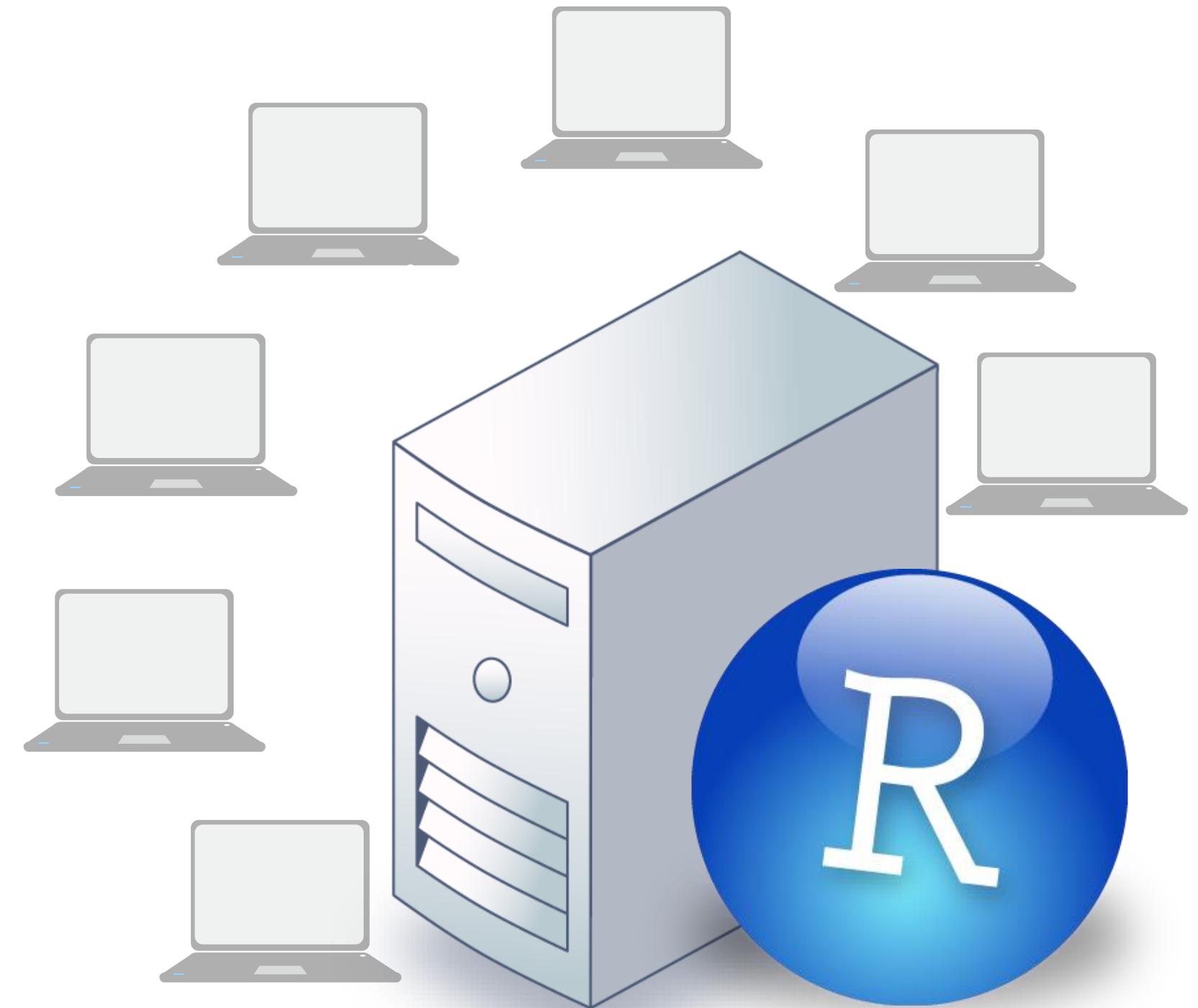




RStudio
Desktop IDE



RStudio Server
Open Source



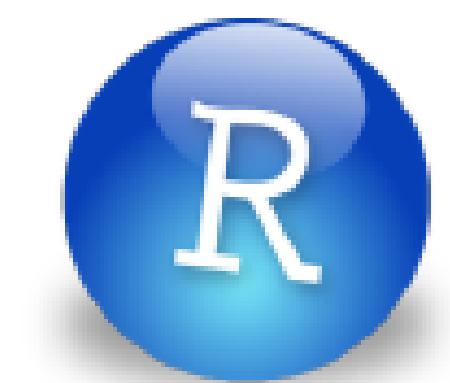
RStudio Server
Pro

<http://www.rstudio.com/wp-content/uploads/2016/01/rstudio-IDE-cheatsheet.pdf>

Class Materials



R
cran.r-project.org



RStudio IDE
www.rstudio.com/download



Code and Exercises
github.com

Set up

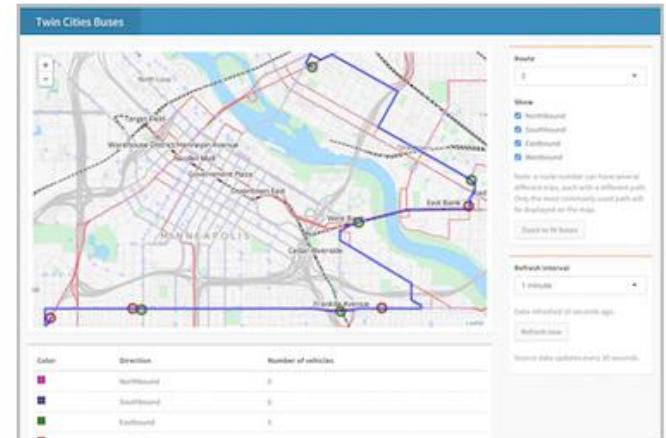
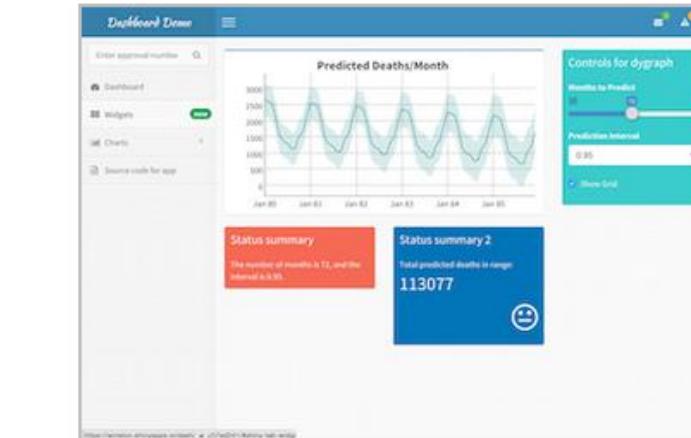
1. Download **R** from cran.r-project.org, follow the links for your operating system
2. Download the **RStudio IDE** from [www.rstudio.com/
download](http://www.rstudio.com/download). Select the version for your operating system.
3. Download the **class materials** from <https://github.com/>
4. Open RStudio and run the command
`install.packages("shiny")` to install/update **shiny**

Introduction to Shiny





Shiny Apps for the Enterprise



[Shiny Dashboard Demo](#)

A dashboard built with Shiny.

[Location tracker](#)

Track locations over time with streaming data.

[Download monitor](#)

Streaming download rates visualized as a bubble chart.

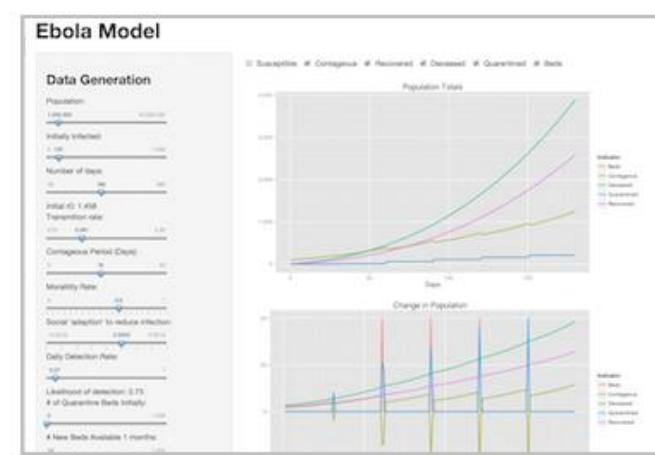
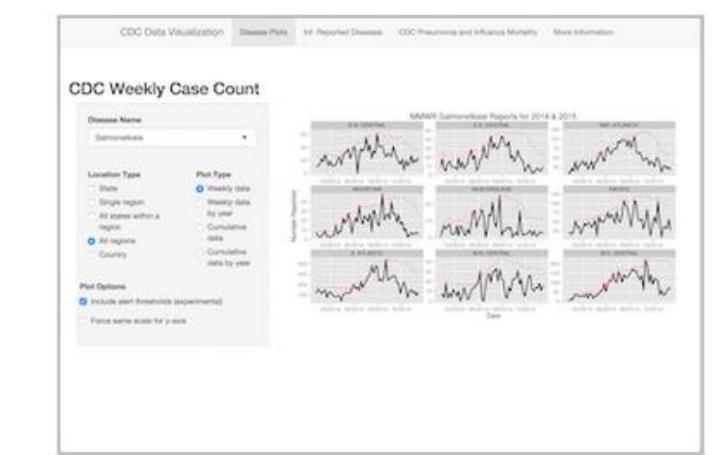
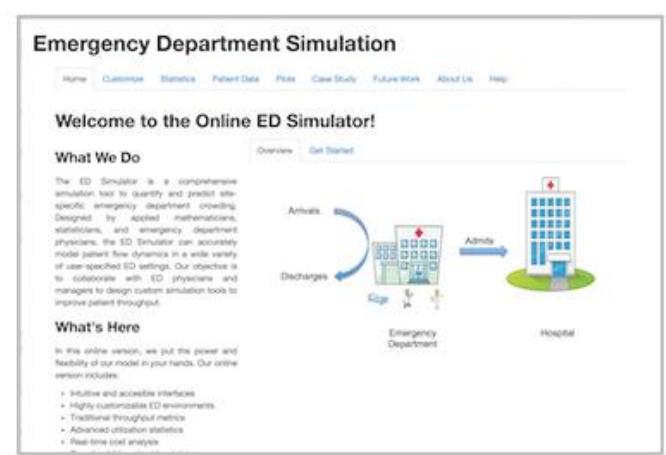
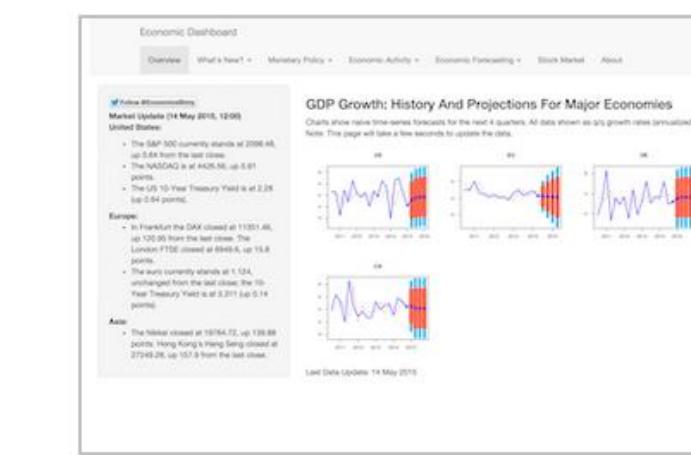
[Supply and Demand](#)

Forecast demand to plan resource allocation.

Shiny Showcase

www.rstudio.com/products/shiny/shiny-user-showcase/

Industry Specific Shiny Apps



[Economic Dashboard](#)

Economic forecasting with macroeconomic indicators.

[ER Optimization](#)

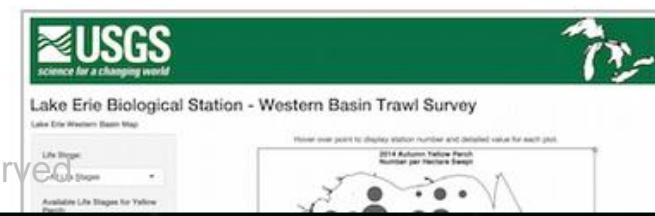
An app that models patient flow.

[CDC Disease Monitor](#)

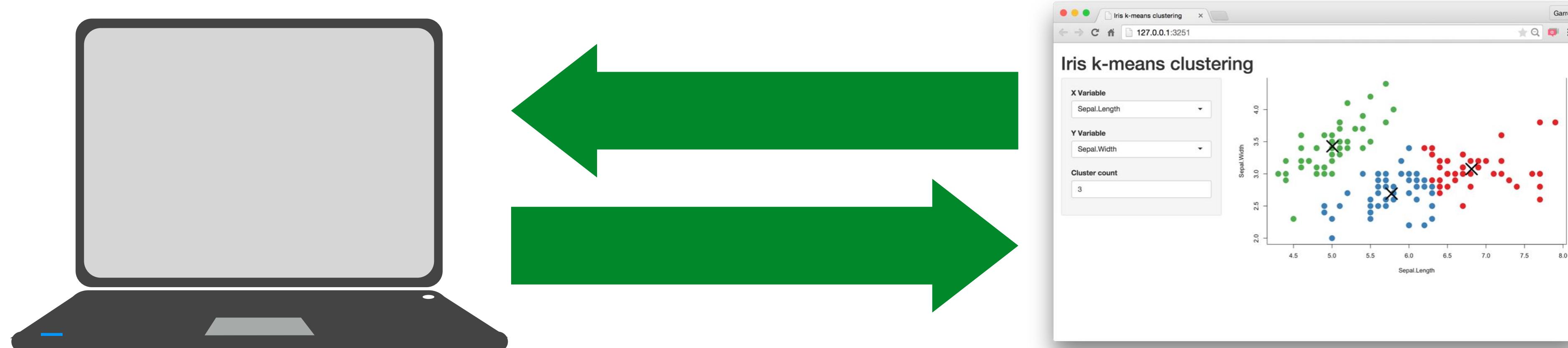
Alert thresholds and automatic weekly updates.

[Ebola Model](#)

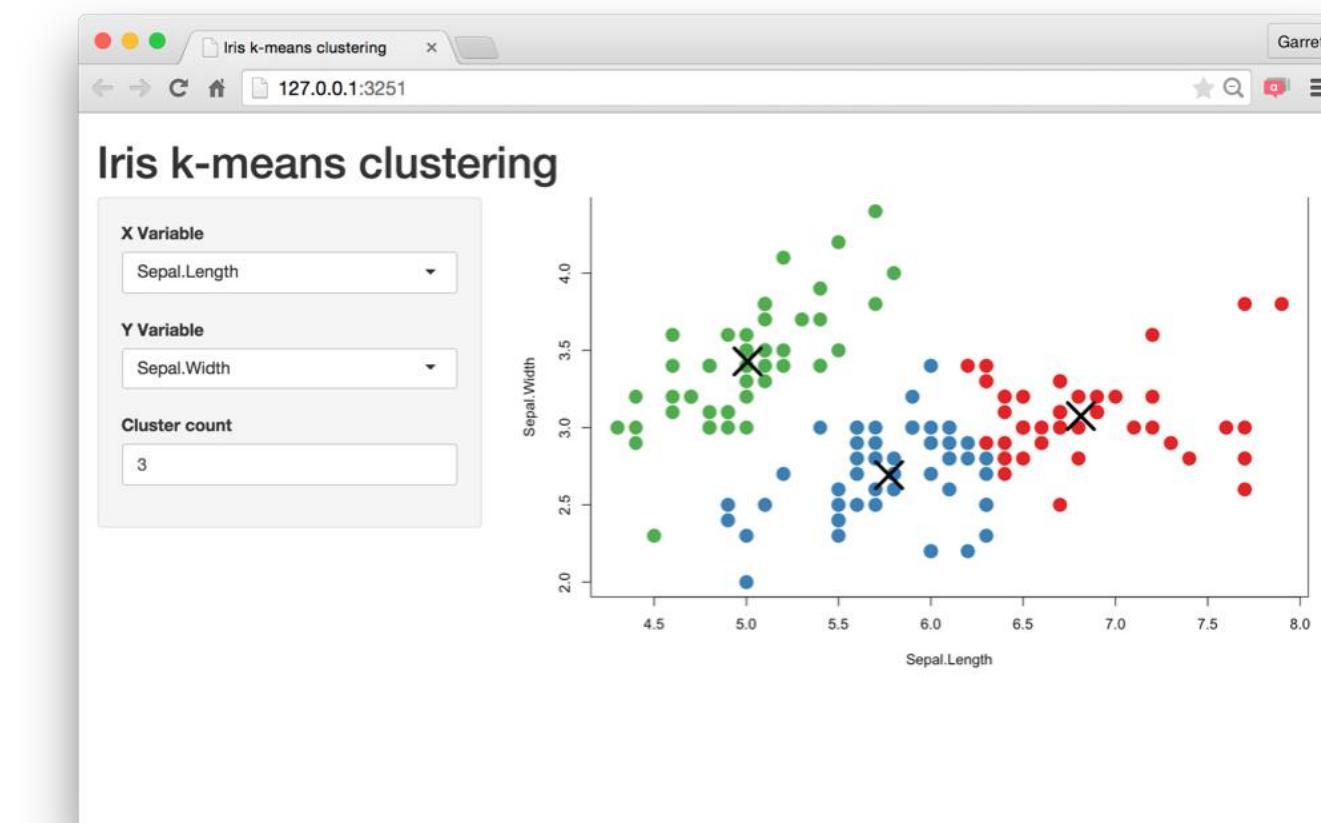
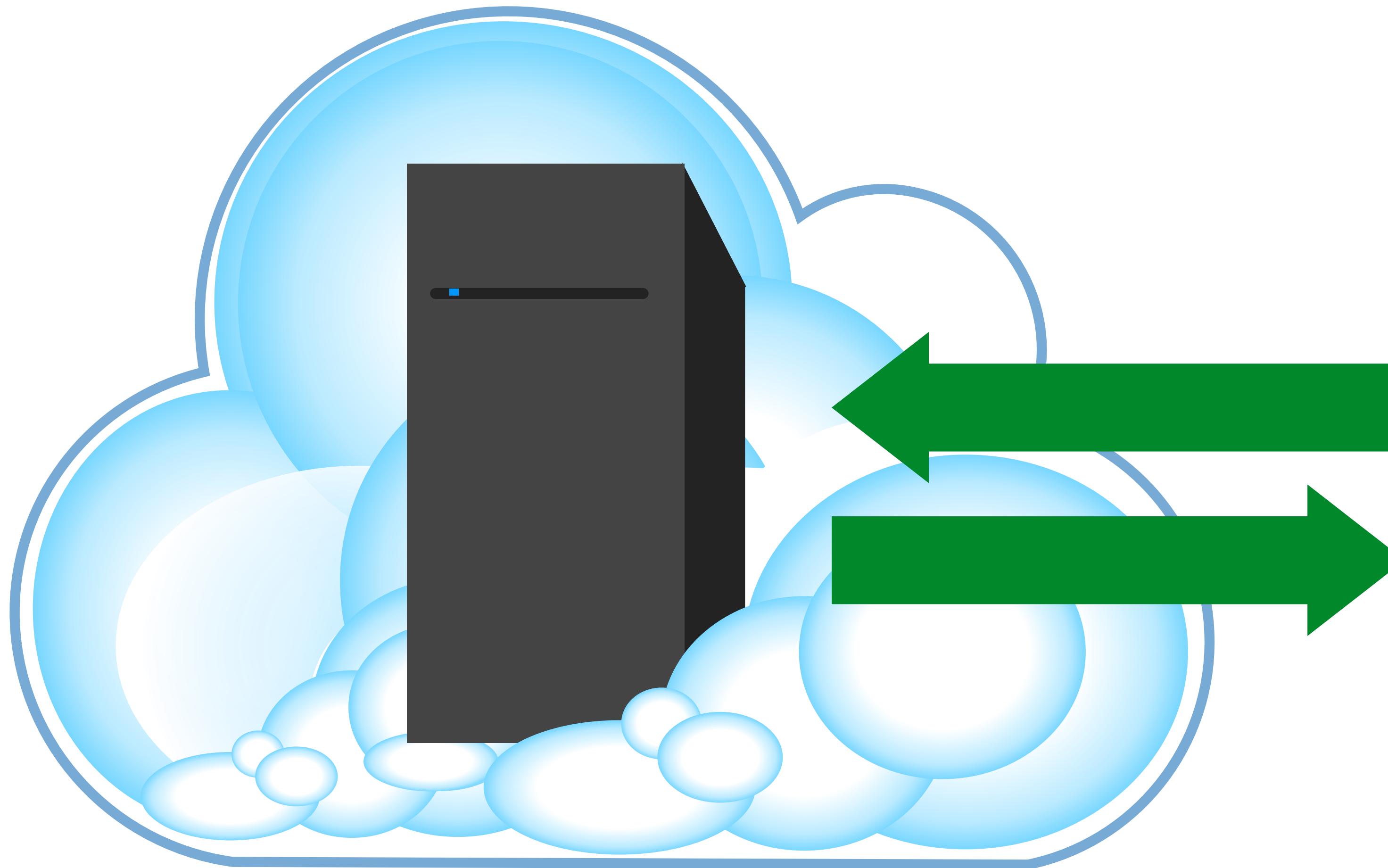
An epidemiological simulation.



Every Shiny app is maintained by a computer running R

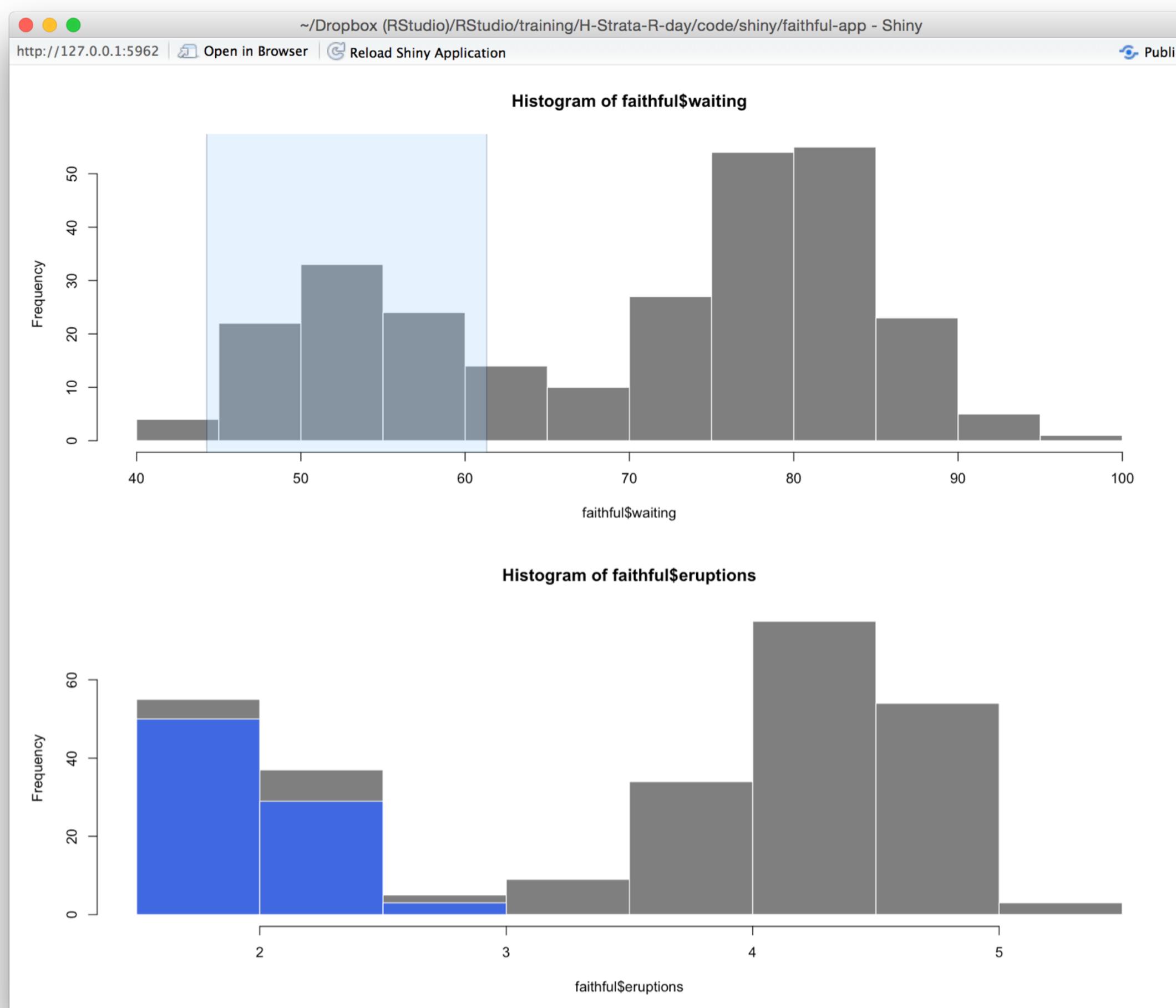


Every Shiny app is maintained by a computer running R

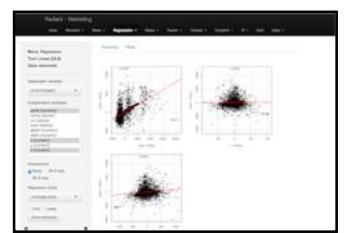


**what can
an app do?**

demos



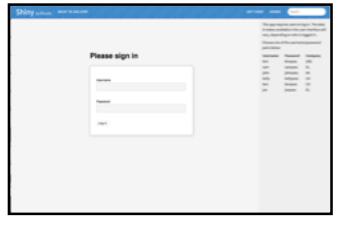
What can a Shiny app do?



Make R analysis accessible to **non-programmers**



Highly **customizable**, highly **shareable** HTML front end



Read and write to **databases**



Monitor **streaming data**



Require and use **authentication**



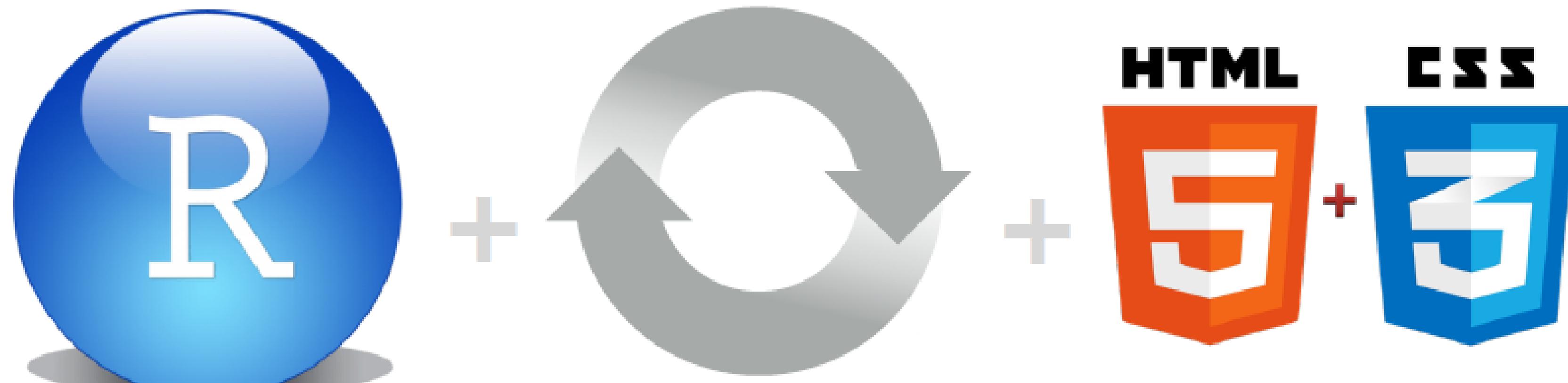
Ideal for Exploratory Data Analysis



Ideal Data Portal / Results Explorer / Simulation API / Dashboard

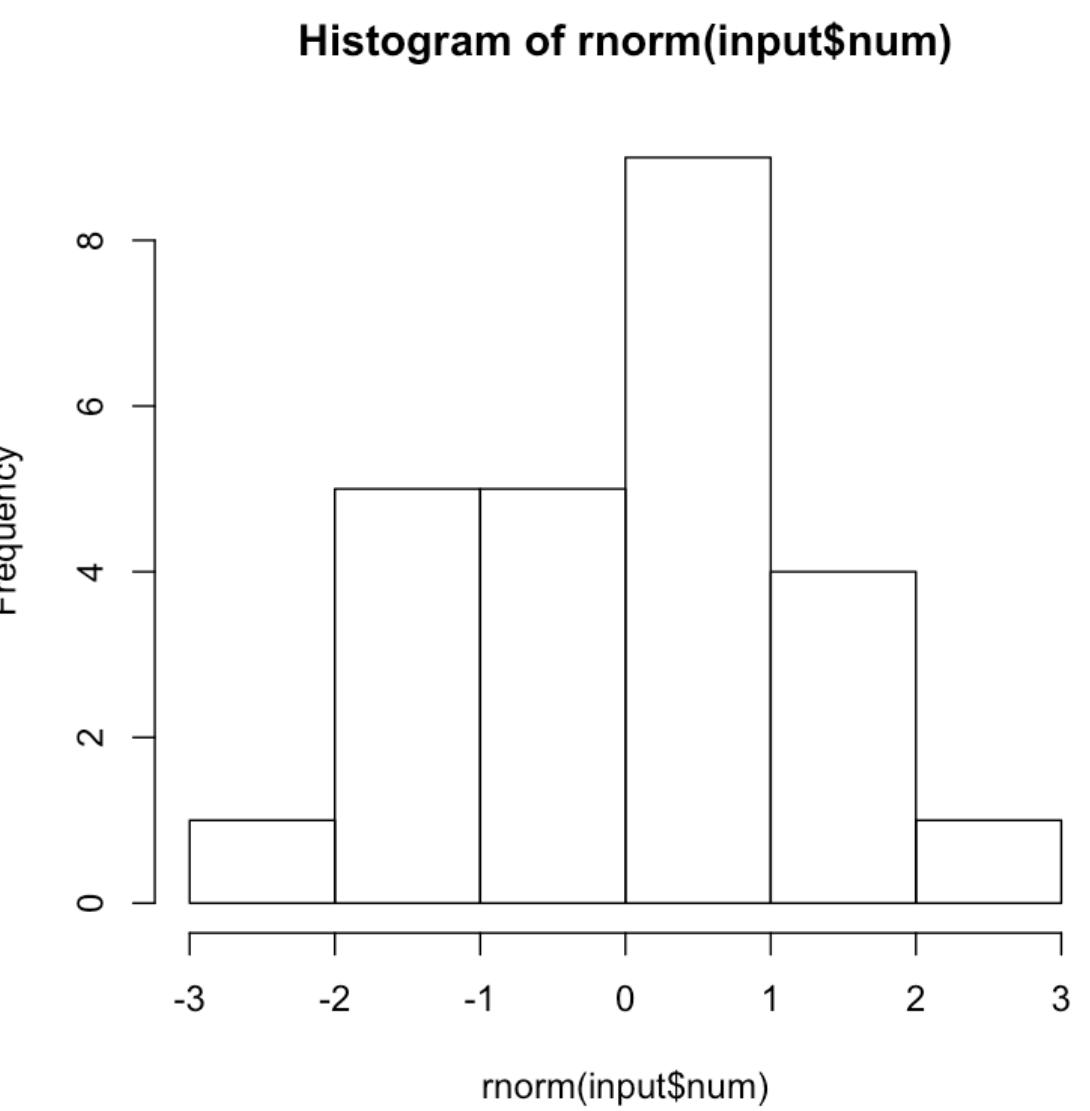
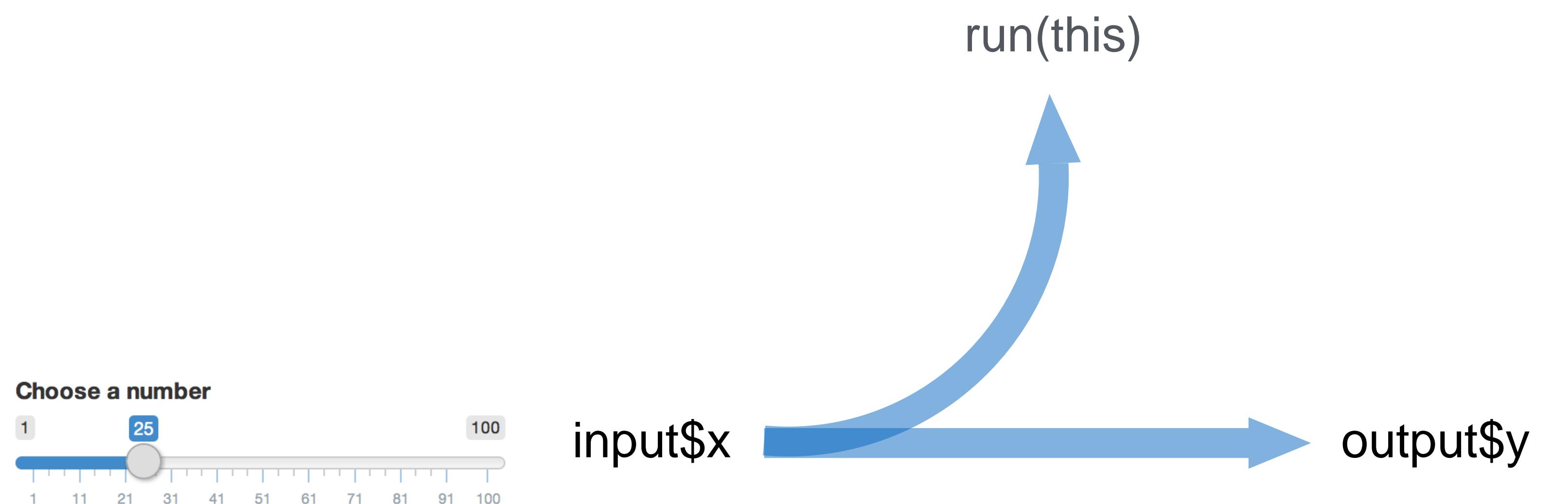
**How to make
an app**

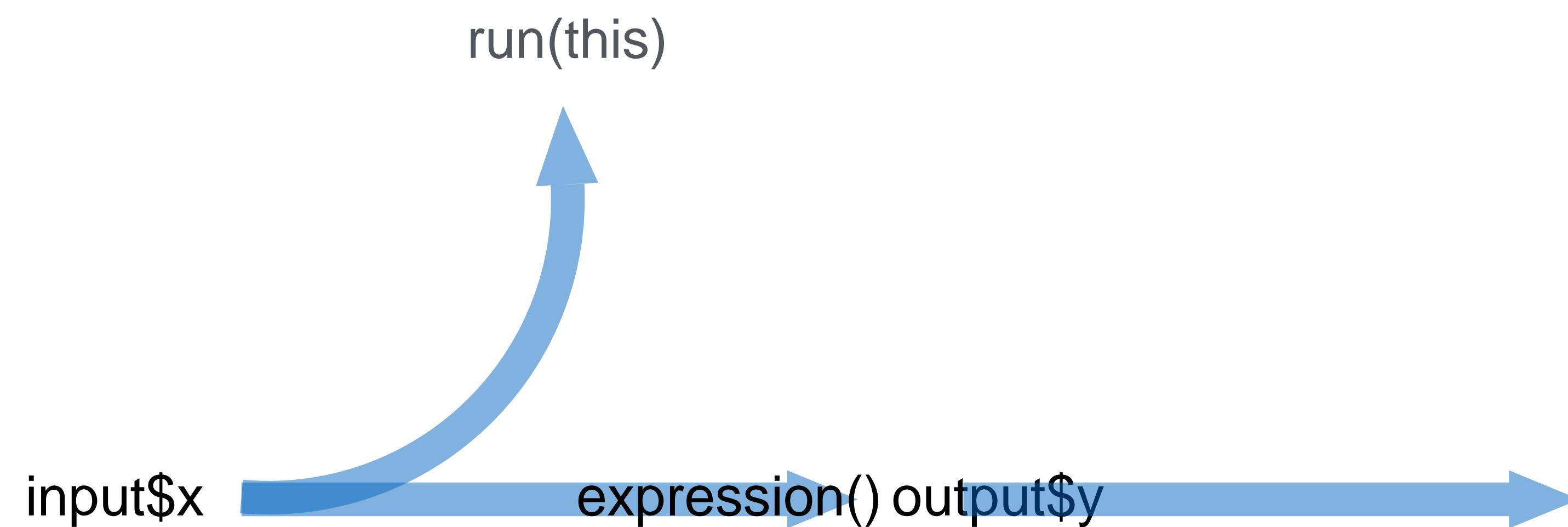
Shiny

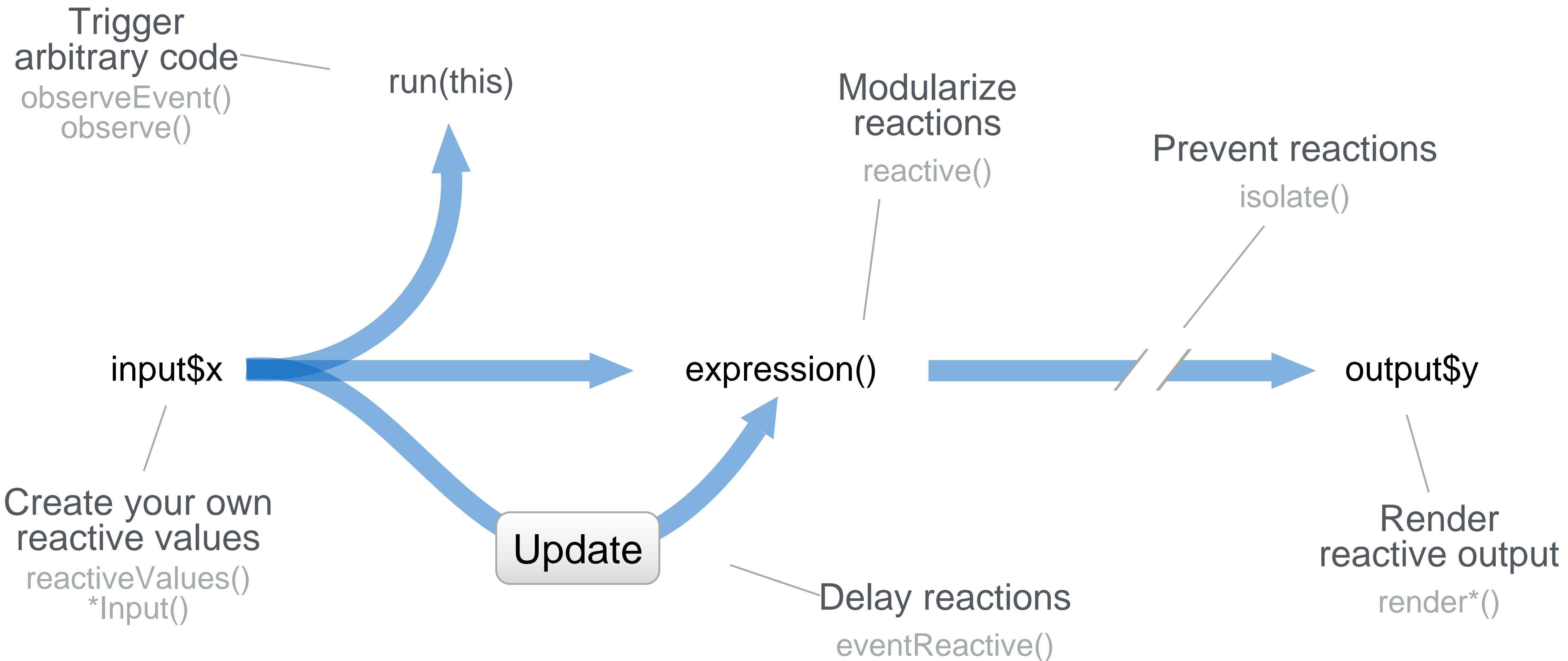


Reactive
Programming

Web based
User Interface

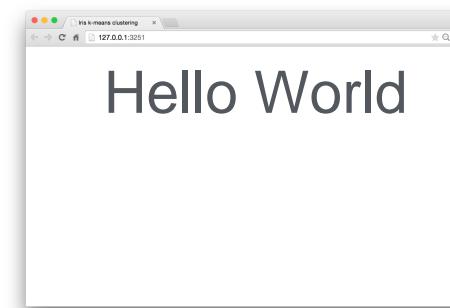




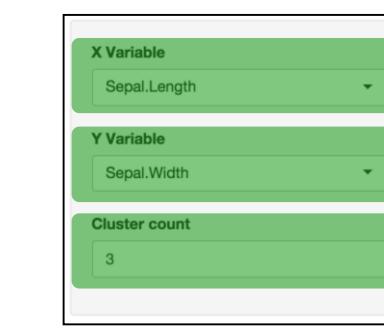


Recap: UI

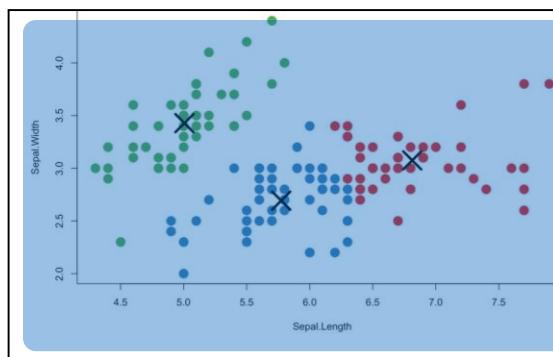
```
library(shiny)
ui <- fluidPage()
server <- function(input, output) {}
shinyApp(ui = ui, server = server)
```



Begin each app with the template



Add elements as arguments to **fluidPage()**



Create reactive inputs with an ***Input()** function

Display reactive results with an ***Output()** function

Recap: Server



`output$hist <-`

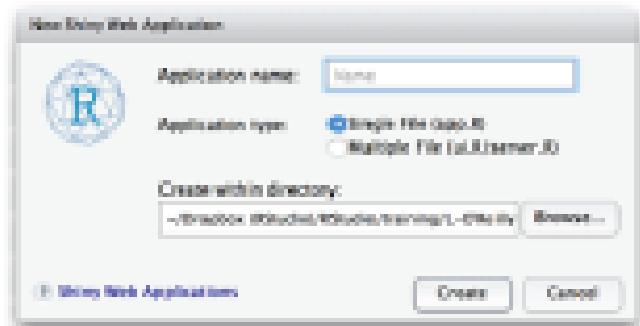
```
renderPlot({  
  hist(rnorm(input$num))  
})
```

`input$num`

Use the `server` function to assemble inputs into outputs. Follow 3 rules:

1. Save the output that you build to `output$`
2. Build the output with a `render*()` function
3. Access input values with `input$`

Recap



Open a new Shiny app with
File ▶ New File ▶ Shiny Web App...



Launch the app by opening app.R and
clicking **Run App**



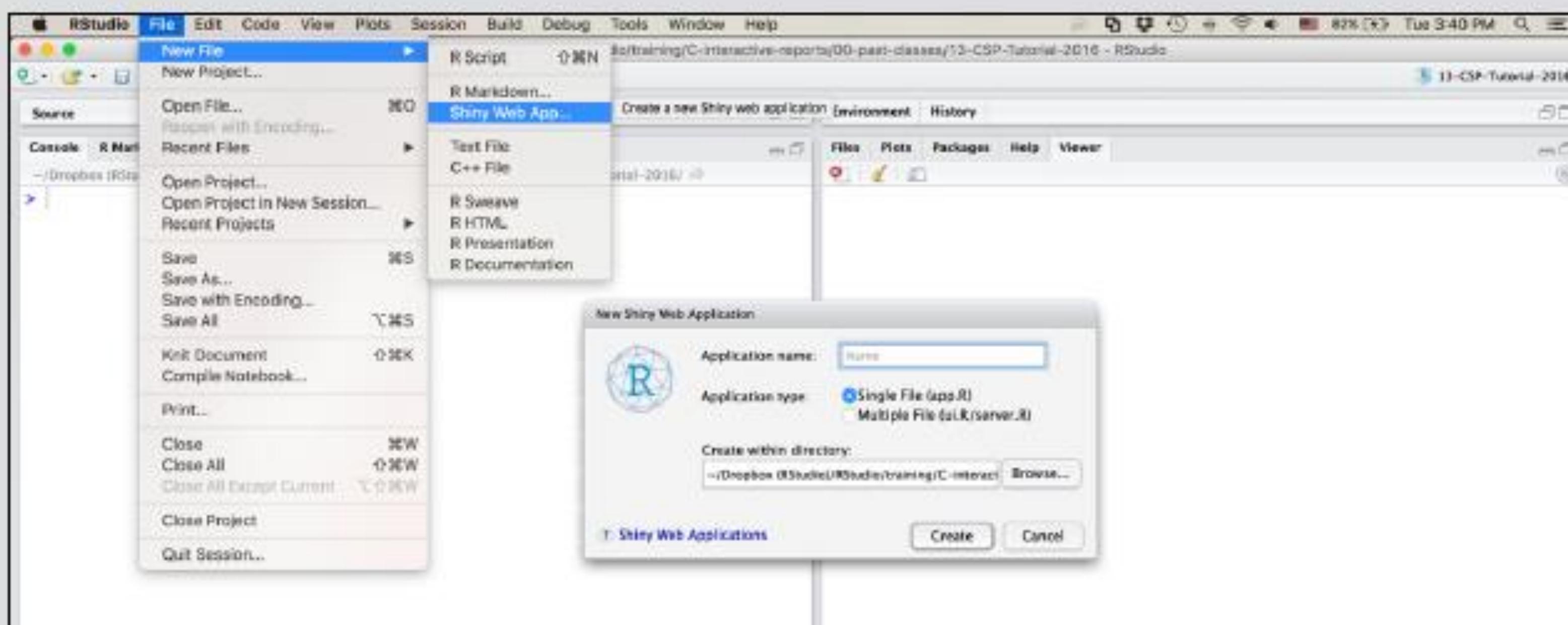
Close app by clicking the stop sign icon



Select view mode in the drop down
menu next to Run App

Warm up

1. Go to File ► New File ► Shiny Web App ► Single File
2. Click Run App at the top of the file to launch your first Shiny App.



03 : 00

runApp

You can launch any app from the command line
with `runApp`

```
runApp("~/Documents/App-1")
```

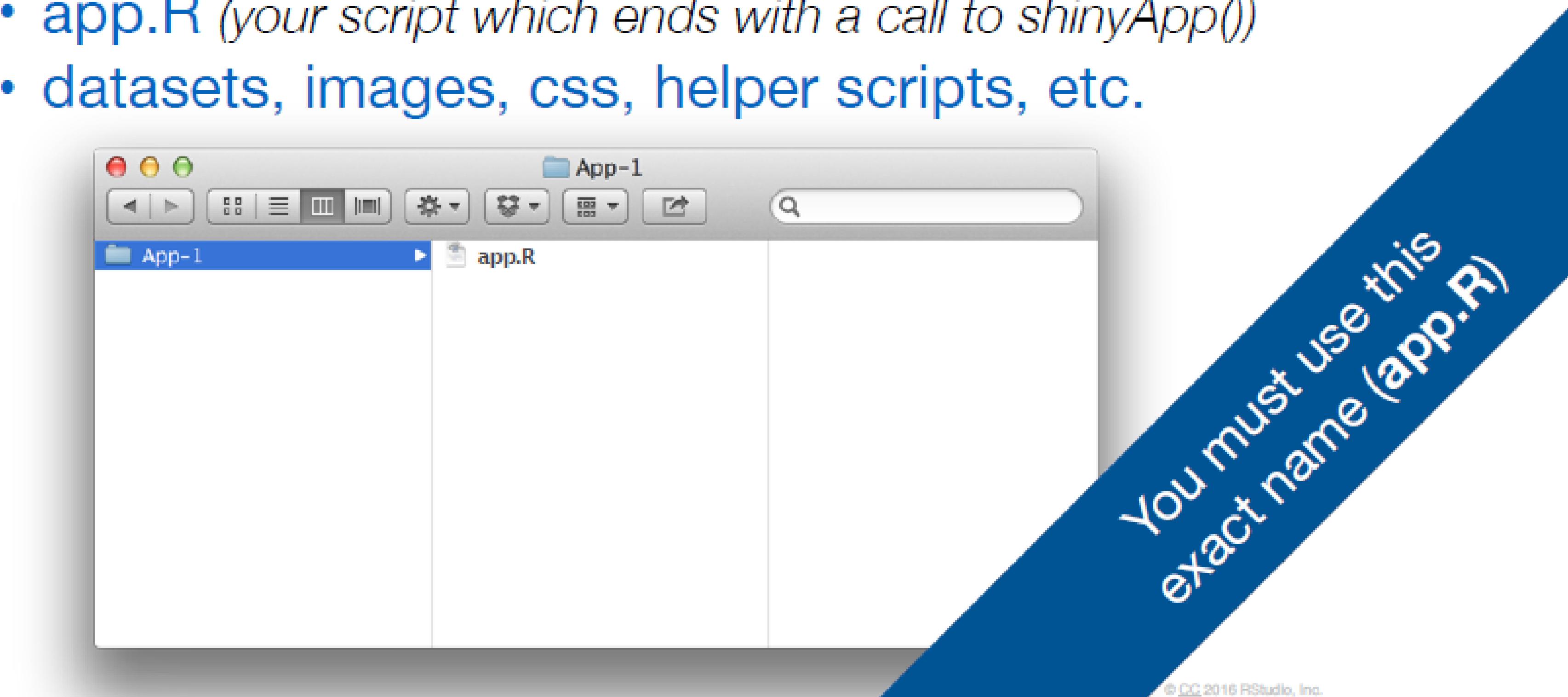
File path to app directory.

R will append the file path to the working directory,
if path does not begin at the home directory

How to save your app

One directory with every file the app needs:

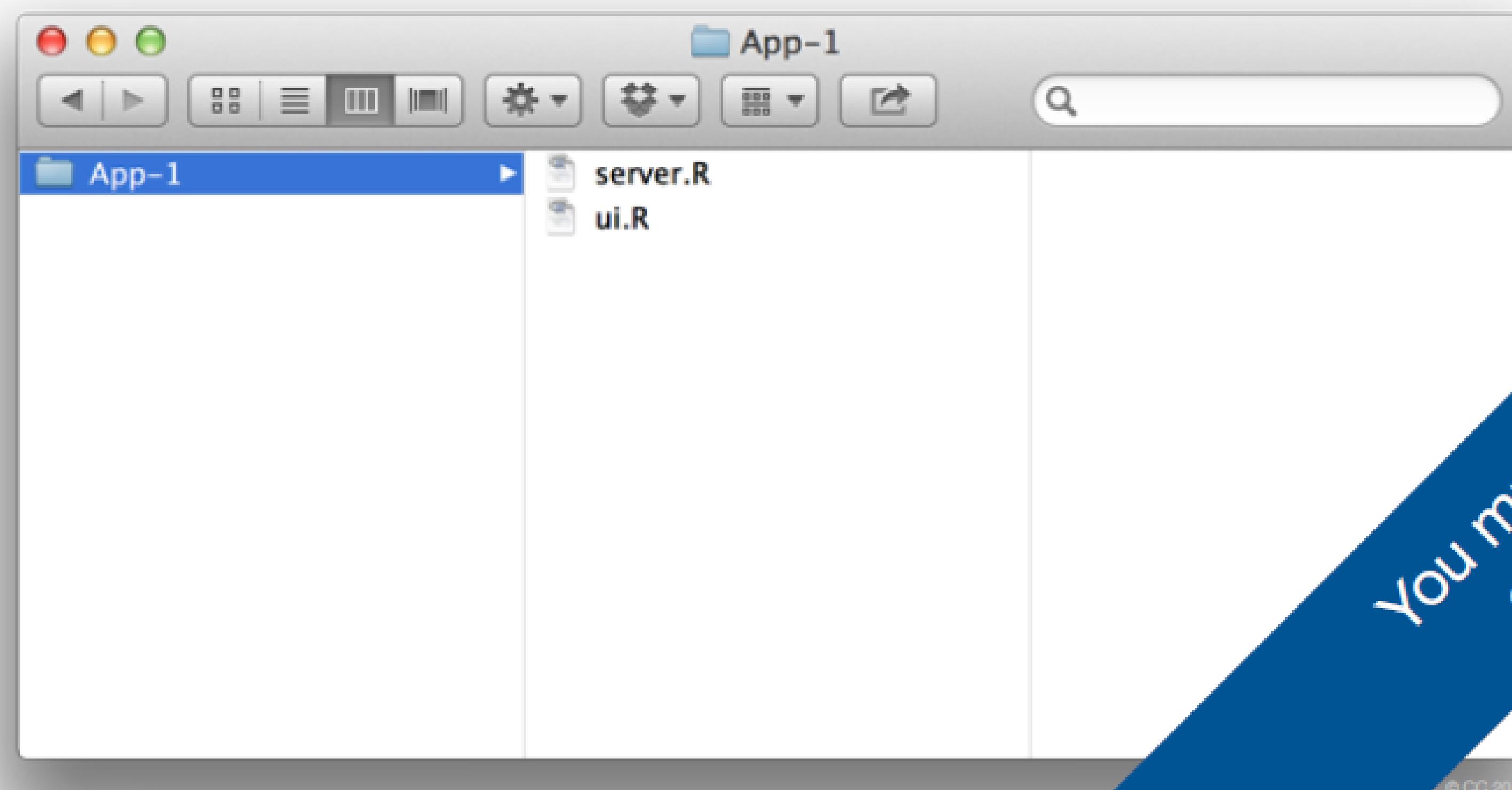
- `app.R` (*your script which ends with a call to shinyApp()*)
- datasets, images, css, helper scripts, etc.



Two file apps

One directory with two files:

- `server.R`
- `ui.R`



You must use these
exact names

© CC 2015 RStudio, Inc.

Two file apps

```
library(shiny)

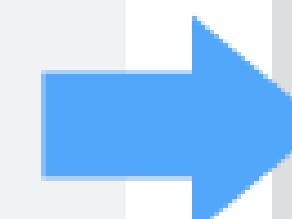
ui <- fluidPage(
  sliderInput(inputId = "num",
    label = "Choose a number",
    value = 25, min = 1, max = 100),
  plotOutput("hist")
)

server <- function(input, output) {
  output$hist <- renderPlot({
    hist(rnorm(input$num))
  })
}

shinyApp(ui = ui, server = server)
```



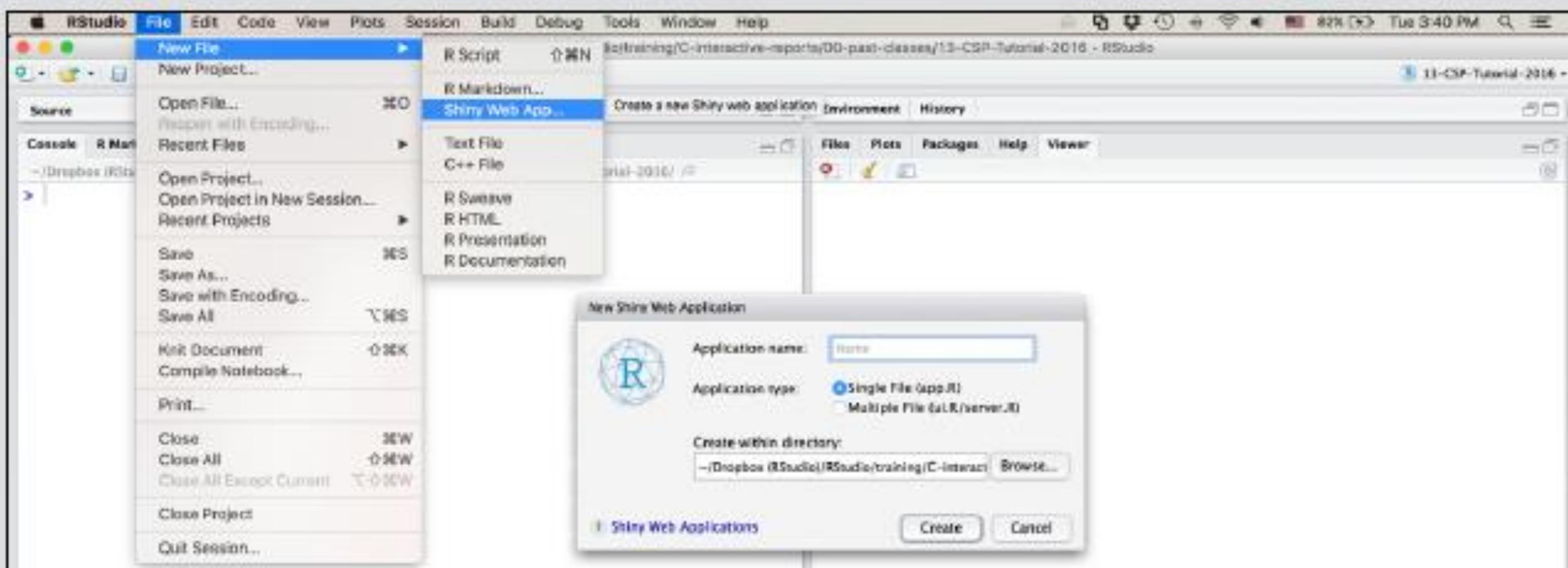
```
# ui.R
library(shiny)
fluidPage(
  sliderInput(inputId = "num",
    label = "Choose a number",
    value = 25, min = 1, max = 100),
  plotOutput("hist")
)
```



```
# server.R
library(shiny)
function(input, output) {
  output$hist <- renderPlot({
    hist(rnorm(input$num))
  })
}
```

Your Turn

1. Go to New > Shiny Web App > Multiple File App > Create
2. Open ui.R and server.R. Edit them to recreate your histogram app.
3. Run the app.



03 : 00

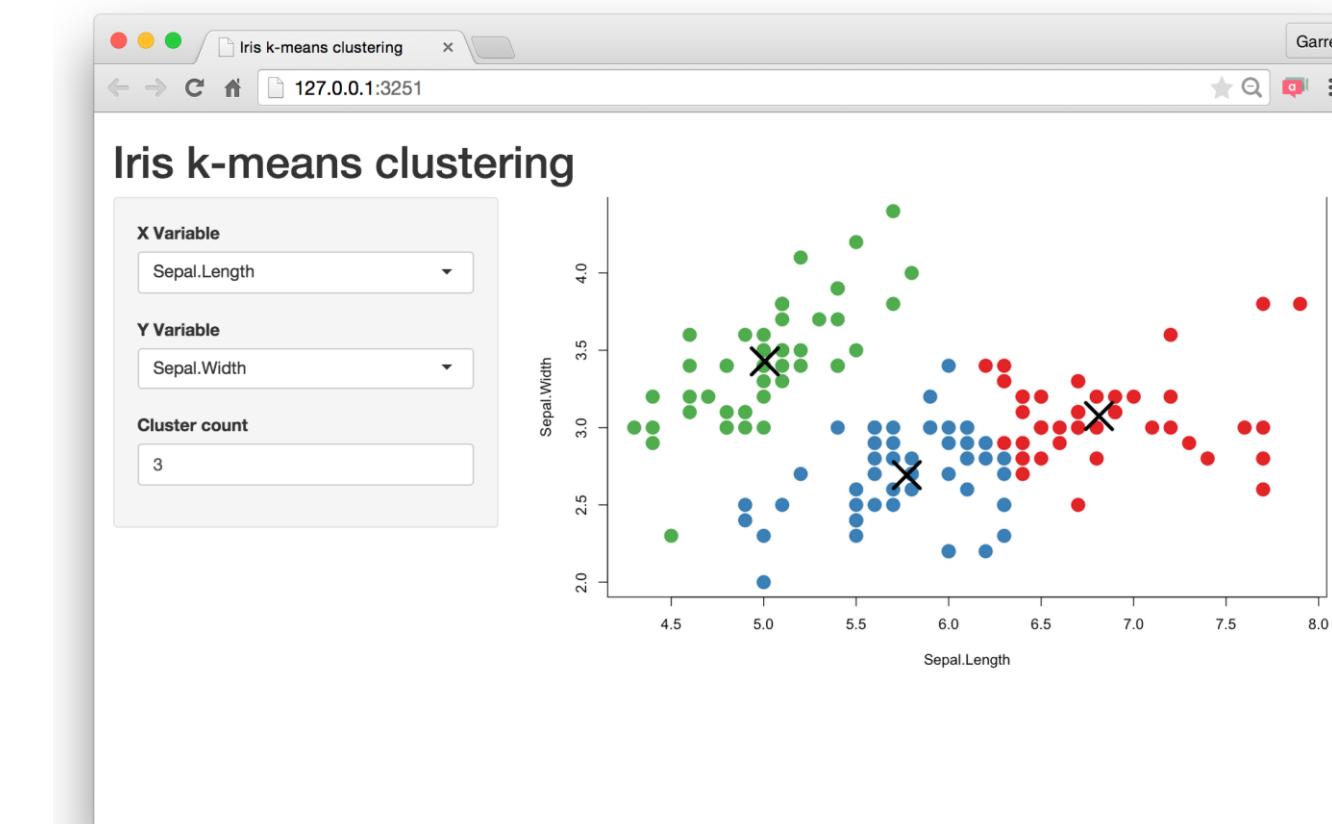
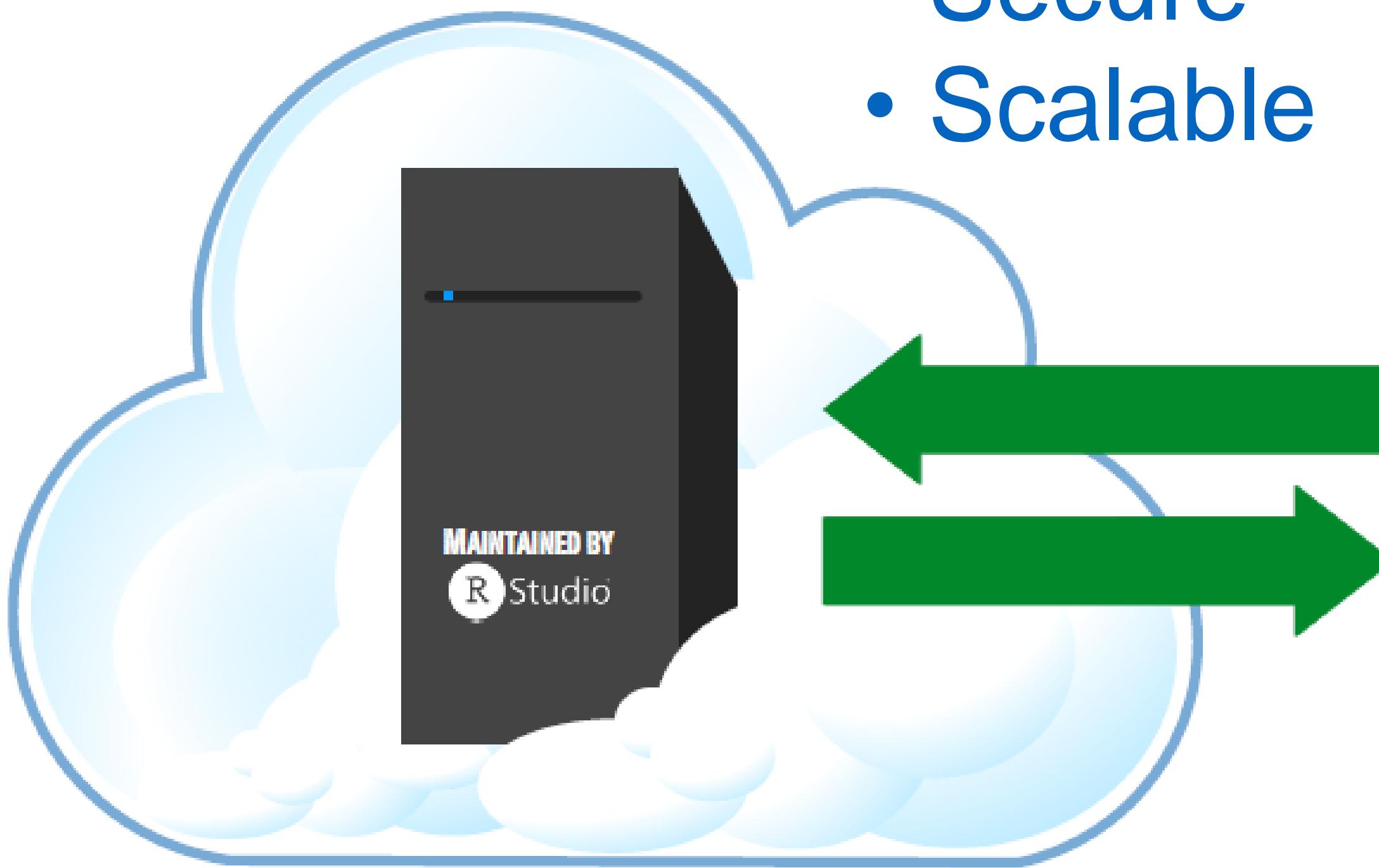
**Share
your app**



Shinyapps.io

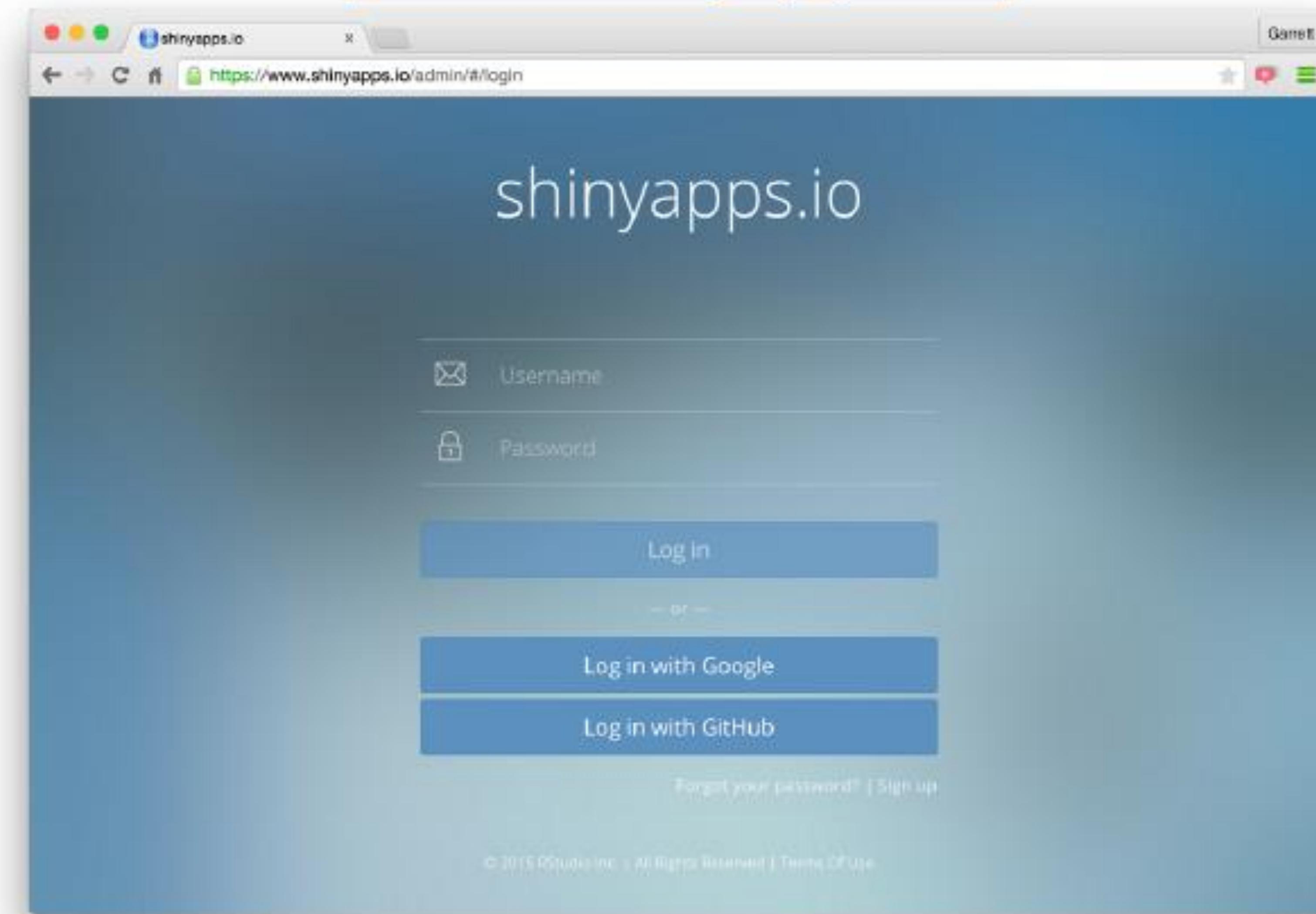
A server maintained by RStudio

- Free tier available
- Easy to use
- Secure
- Scalable



Hassle-free cloud hosting for Shiny

www.shinyapps.io

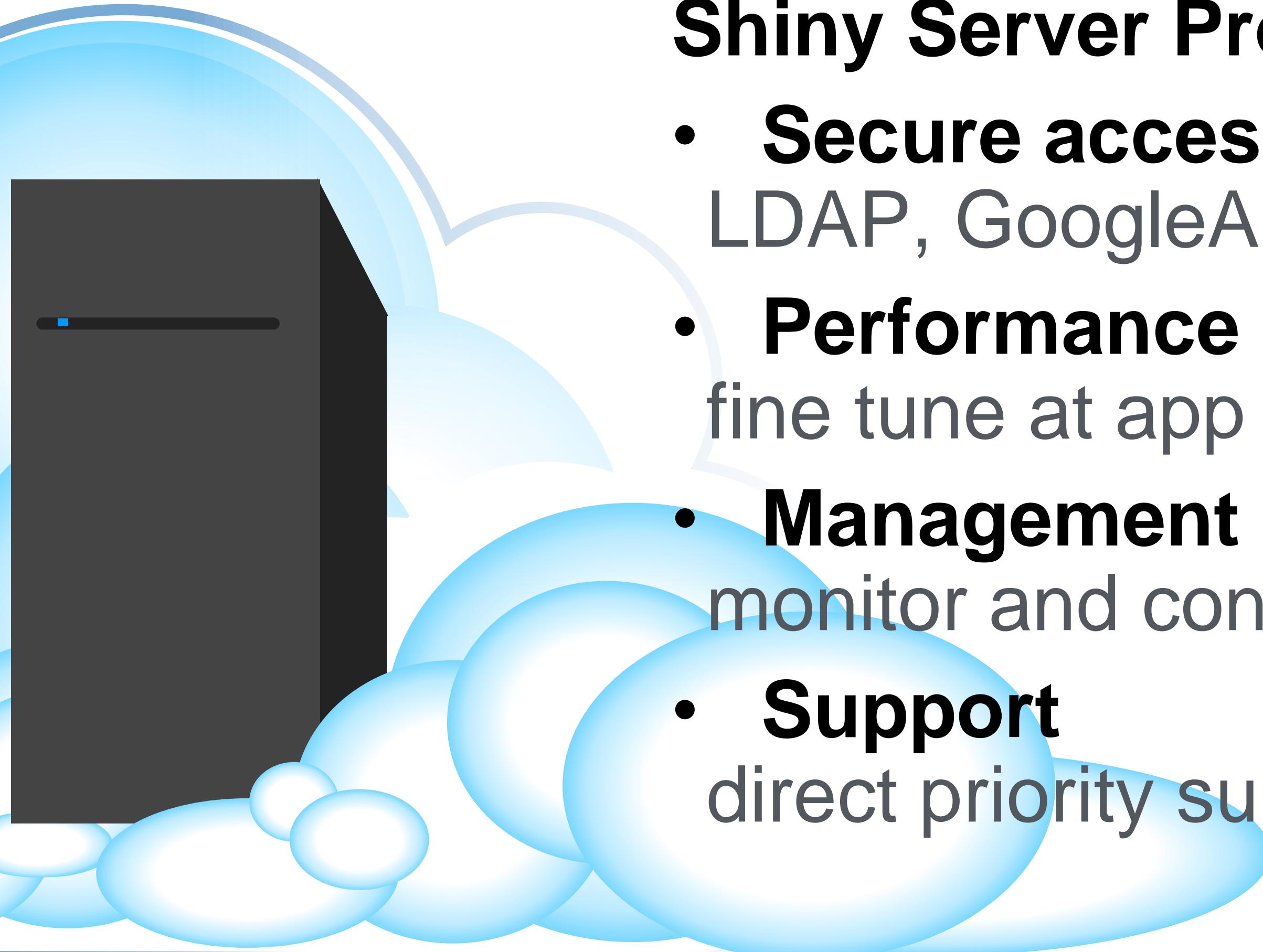


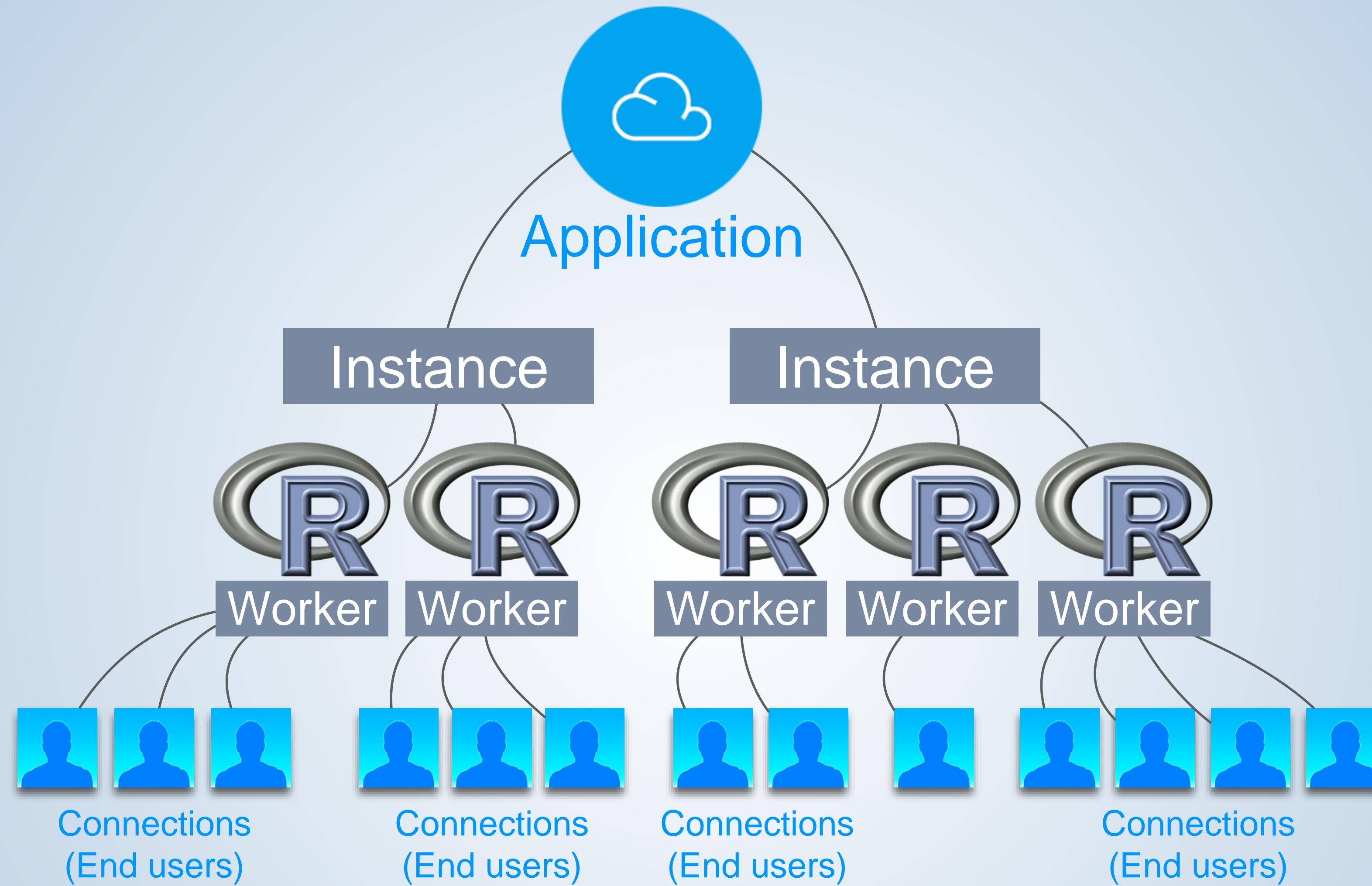
**Build
your own
server**



Shiny Server OS & Shiny Server Pro

www.rstudio.com/products/shiny/shiny-server/

- 
- ## Shiny Server Pro Features:
- **Secure access**
LDAP, GoogleAuth, SSL, and more
 - **Performance**
fine tune at app and server level
 - **Management**
monitor and control resource use
 - **Support**
direct priority support



Teach
yourself
Shiny

Learn more

shiny.rstudio.com/articles

Interactive plots

Create interactive plots with base and ggplot2 graphics

[Interactive plots](#)

[Selecting rows of data](#)

[Interactive plots - advanced](#)

The screenshot shows the Shiny documentation website with a sidebar menu:

- OVERVIEW
- TUTORIAL
- ARTICLES** (selected)
- GALLERY
- REFERENCE
- DEPLOY
- HELP

Articles

The basics

If you've been through the [tutorial](#) and need a refresher, these articles are a good place to start. They describe the lay of the land.

- The basic parts of a Shiny app
- How to build a Shiny app
- How to launch a Shiny app
- How to get help
- The Shiny cheat sheet
- Single-file Shiny apps
- App formats and launching methods
- Persistent data storage in Shiny apps

Extend Shiny

These packages provide advanced features that can enhance your Shiny apps.

- [shinythemes](#) - CSS themes ready to use with Shiny
- [shinydashboard](#) - Shiny powered dashboards
- [htmlwidgets](#) - A framework for embedding JavaScript visualizations into R. Ready to use examples include:
 - [leaflet](#) - Geo-spatial mapping ([article](#))
 - [dygraphs](#) - Time series charting ([article](#))
 - [MetricGraphics](#) - Scatterplots and line charts with D3
 - [networkD3](#) - Graph data visualization with D3
 - [DataTables](#) - Tabular data display ([article](#))
 - [rCharts](#) - 3D scatterplots and globes
 - [d3heatmap](#) - Heatmaps ([article](#))
 - [diagrammeR](#) - Graph and flowchart diagrams ([article](#))

Layouts and UI

These articles explain how to control the layout, user-interface, and general appearance of your Shiny apps.

- [Application layout guide](#)
- [Display modes](#)
- [Tabssets](#)
- [Customize your UI with HTML](#)
- [Build your entire UI with HTML](#)
- [Build a dynamic UI that reacts to user input](#)
- [Shiny HTML Tags Glossary](#)
- [Progress indicators](#)

Deploying apps

These articles describe the different ways to share your Shiny apps with users.

- [Getting started with shinyapps.io](#)
- [Setting up custom domains with shinyapps.io](#)
- [Scaling and Performance Tuning with shinyapps.io](#)
- [Share data across sessions with shinyapps.io](#)
- [Migrating shinyapps.io authentication](#)
- [Introduction to Shiny Server](#)
- [Save your app as a function](#)
- [Sharing apps to run locally](#)

Interactive documents

These articles explain how to add Shiny components to R Markdown reports.

- [Introduction to R Markdown](#)
- [Introduction to interactive documents](#)
- [R Markdown integration in the RStudio IDE](#)
- [The R Markdown Cheat sheet](#)

Widgets

These articles describe Shiny's pre-built widgets and provide ideas on how to use them. (See also [Lesson 3](#) in the tutorial, and the [Widgets](#) section in the [gallery](#).)

- [Using Action Buttons](#)
- [Using sliders](#)
- [Help users download data from your app](#)
- [Using selective input](#)

Reactive programming

These articles describe reactivity from a conceptual level. Understanding reactivity will help you build apps that are more efficient, robust, and correct.

- [Reactivity: An overview](#)
- [Stop reactions with `isolate\(\)`](#)
- [Execution scheduling](#)
- [How to understand reactivity in R](#)

Customizing Shiny

These articles suggest ways to create custom Shiny widgets, layouts and outputs; or to combine Shiny with other web technologies.

- [Style your apps with CSS](#)
- [Build custom input objects](#)
- [Build custom output objects](#)
- [Add Google Analytics to a Shiny app](#)

Shiny Server Pro

Here are some of the unique things you can do when you deploy your apps with Shiny Server Pro.

- [How to create User Privileges](#)
- [Allow different libraries for different apps](#)

Upgrade notes

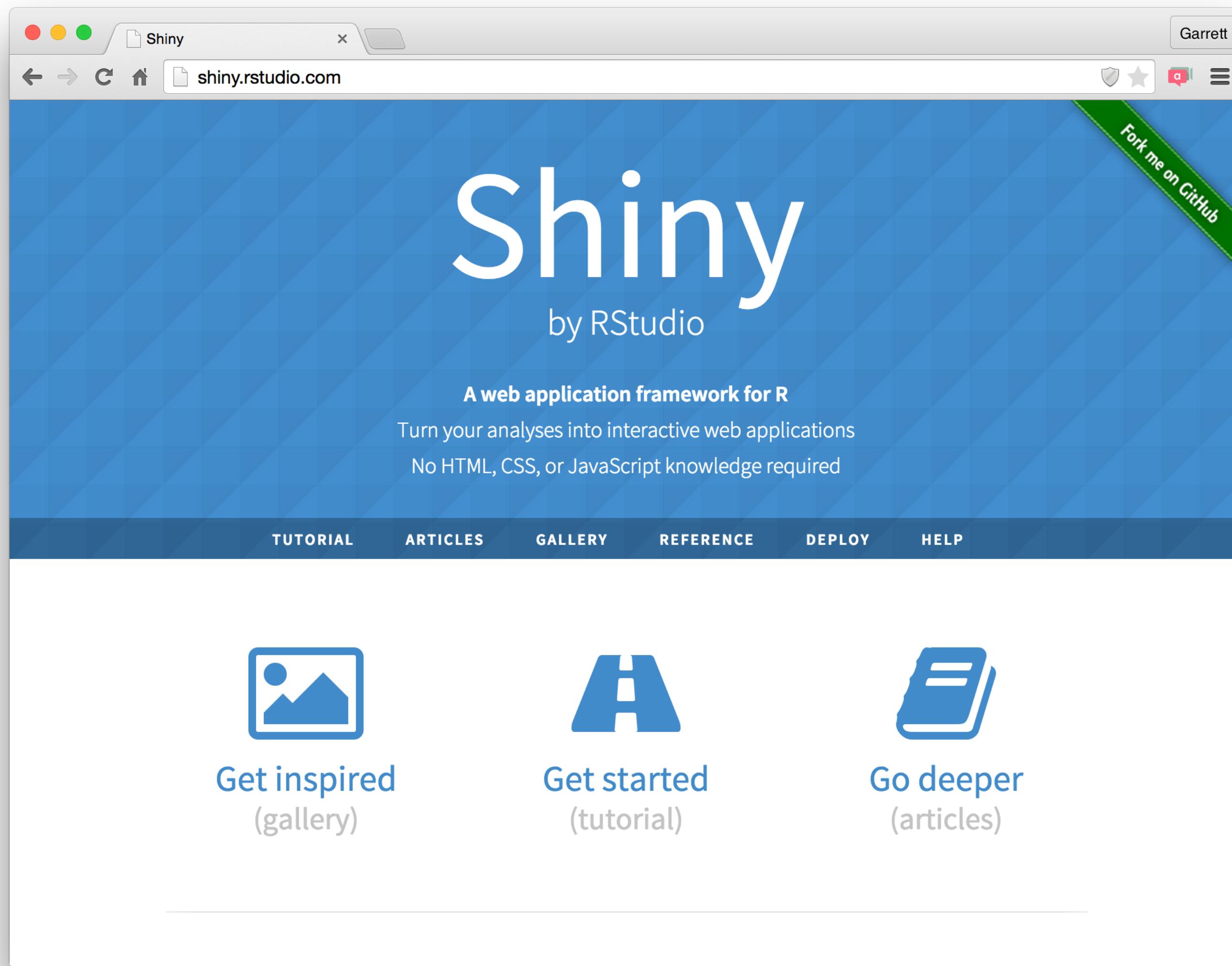
Notes for upgrading to particular versions of Shiny

- [Upgrade notes for Shiny 0.11](#)
- [Upgrade notes for Shiny 0.12](#)

Shiny is an RStudio project. © 2014 RStudio, Inc.

The Shiny Development Center

shiny.rstudio.com



The Shiny Cheat Sheet

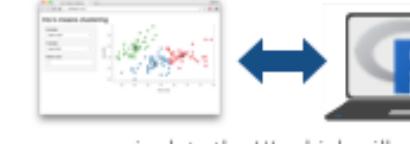
www.rstudio.com/resources/cheatsheets/

Interactive Web Apps with shiny Cheat Sheet

learn more at shiny.rstudio.com

Basics

A **Shiny** app is a web page (**UI**) connected to a computer running a live R session (**Server**)



Users can manipulate the UI, which will cause the server to update the UI's displays (by running R code).

App template

Begin writing a new app with this template. Preview the app by running the code at the R command line.

```
library(shiny)
ui <- fluidPage()
server <- function(input, output) {
  shinyApp(ui = ui, server = server)}
```

- ui** - nested R functions that assemble an HTML user interface for your app
- server** - a function with instructions on how to build and rebuild the R objects displayed in the UI
- shinyApp** - combines **ui** and **server** into a functioning app. Wrap with `runApp()` if calling from a sourced script or inside a function.

Share your app

The easiest way to share your app is to host it on shinyapps.io, a cloud based service from RStudio

- Create a free or professional account at <http://shinyapps.io>
- Click the **Publish** icon in the RStudio IDE (>=0.99) or run: `rsconnect::deployApp("<path to directory>")`

Build or purchase your own Shiny Server

at www.rstudio.com/products/shiny-server/

Building an App

- Complete the template by adding arguments to `fluidPage()` and a body to the `server` function.

Add inputs to the UI with `*Input()` functions

Add outputs with `*Output()` functions

Tell server how to render outputs with R in the server function. To do this:

- Refer to outputs with `output$<id>`
- Refer to inputs with `input$<id>`
- Wrap code in a `render*`() function before saving to output

```
library(shiny)
ui <- fluidPage(
  numericInput(inputId = "n",
    "Sample size", value = 25),
  plotOutput(outputId = "hist"))
server <- function(input, output) {
  output$hist <- renderPlot({
    hist(rnorm(input$n))
  })
}
shinyApp(ui = ui, server = server)
```

Save your template as `app.R`. Alternatively, split your template into two files named `ui.R` and `server.R`.

```
library(shiny)
ui <- fluidPage(
  numericInput(inputId = "n",
    "Sample size", value = 25),
  plotOutput(outputId = "hist"))
server <- function(input, output) {
  output$hist <- renderPlot({
    hist(rnorm(input$n))
  })
}
shinyApp(ui = ui, server = server)
```

`# ui.R` contains everything you would save to `ui`.

`# server.R` ends with the function you would save to `server`.

No need to call `shinyApp()`.

Save each app as a directory that contains an `app.R` file (or a `server.R` file and a `ui.R` file) plus optional extra files.

• `app-name` - The directory name is the name of the app
 • `app.R` - (optional) defines objects available to both `ui.R` and `server.R`
 • `global.R` - (optional) used in showcase mode
 • `DESCRIPTION` - (optional) data, scripts, etc.
 • `README` - (optional) directory of files to share with web browsers (images, CSS, JS, etc.) Must be named "`www`".
 • `<other files>` - (optional)
 • `www` - (optional)

Launch apps with `runApp(<path to directory>)`

Inputs - collect values from the user

Access the current value of an input object with `input $<inputId>`. Input values are `reactive`.

<code>Action</code>	<code>actionButton(inputId, label, icon, ...)</code>
<code>Link</code>	<code>actionLink(inputId, label, icon, ...)</code>
<code>checkbox 1</code>	<code>checkboxGroupInput(inputId, label, choices, selected, inline)</code>
<code>checkbox 2</code>	<code>checkboxInput(inputId, label, value)</code>
<code>checkbox 3</code>	<code>checkboxInput(inputId, label, value)</code>
<code>checkbox me</code>	<code>checkboxInput(inputId, label, value)</code>
<code>dateInput</code>	<code>dateInput(inputId, label, value, min, max, format, startview, weekstart, language)</code>
<code>dateRangeInput</code>	<code>dateRangeInput(inputId, label, start, end, min, max, format, startview, weekstart, language, separator)</code>
<code>Choose File</code>	<code>fileInput(inputId, label, multiple, accept)</code>
<code>numericInput</code>	<code>numericInput(inputId, label, value, min, max, step)</code>
<code>passwordInput</code>	<code>passwordInput(inputId, label, value)</code>
<code>radioButtons</code>	<code>radioButtons(inputId, label, choices, selected, inline)</code>
<code>selectInput</code>	<code>selectInput(inputId, label, choices, selected, multiple, selectize, width, size) (also <code>selectizeInput</code>)</code>
<code>sliderInput</code>	<code>sliderInput(inputId, label, min, max, value, step, round, format, locale, ticks, animate, width, sep, pre, post)</code>
<code>submitButton</code>	<code>submitButton(text, icon) (Prevents reactions across entire app)</code>
<code>textInput</code>	<code>textInput(inputId, label, value)</code>

**Reproducible
Research**

via

R Markdown

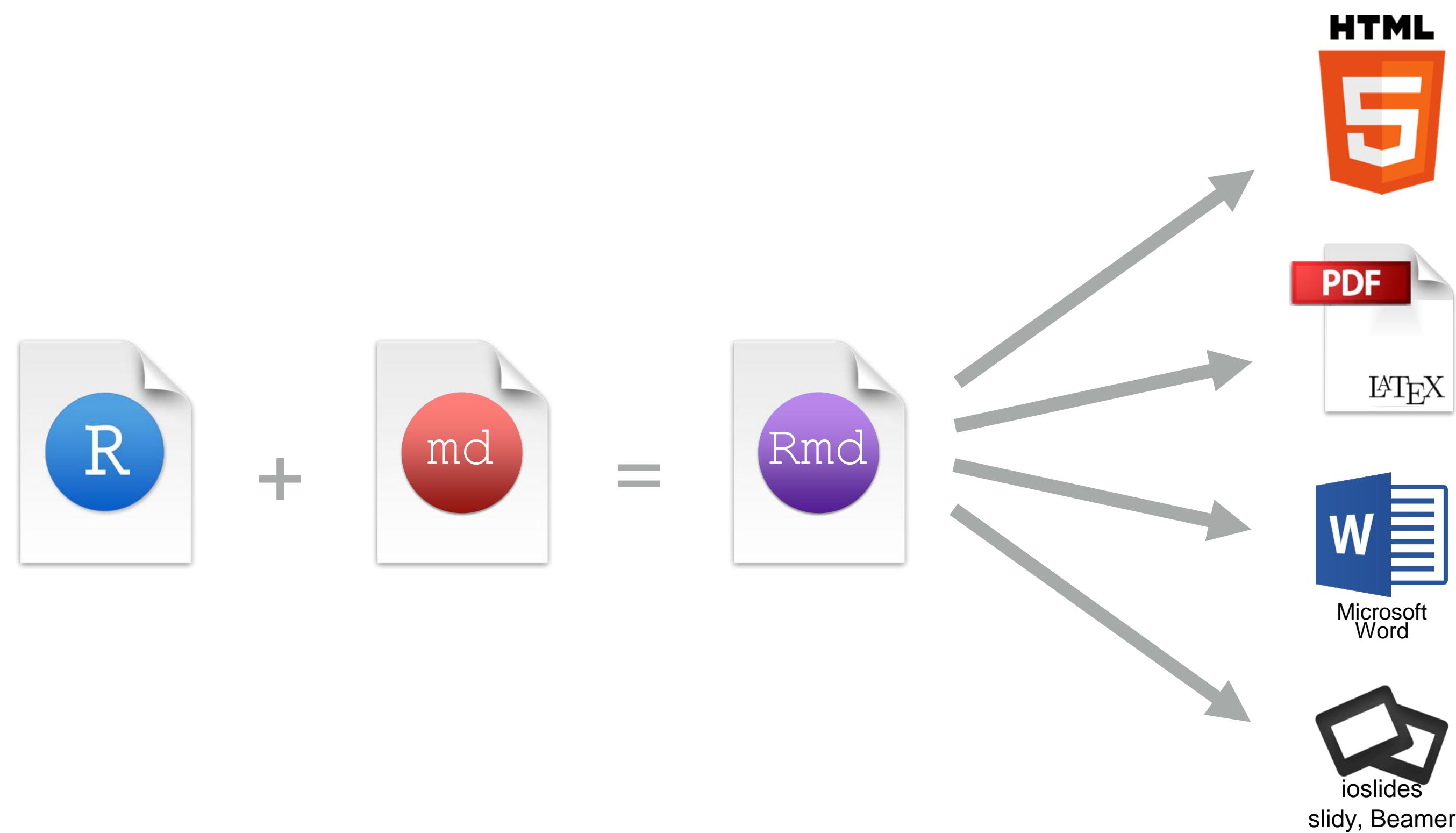


Can we do better?

Ctrl + **C** (Copy)

Ctrl + **V** (Paste)

R Markdown

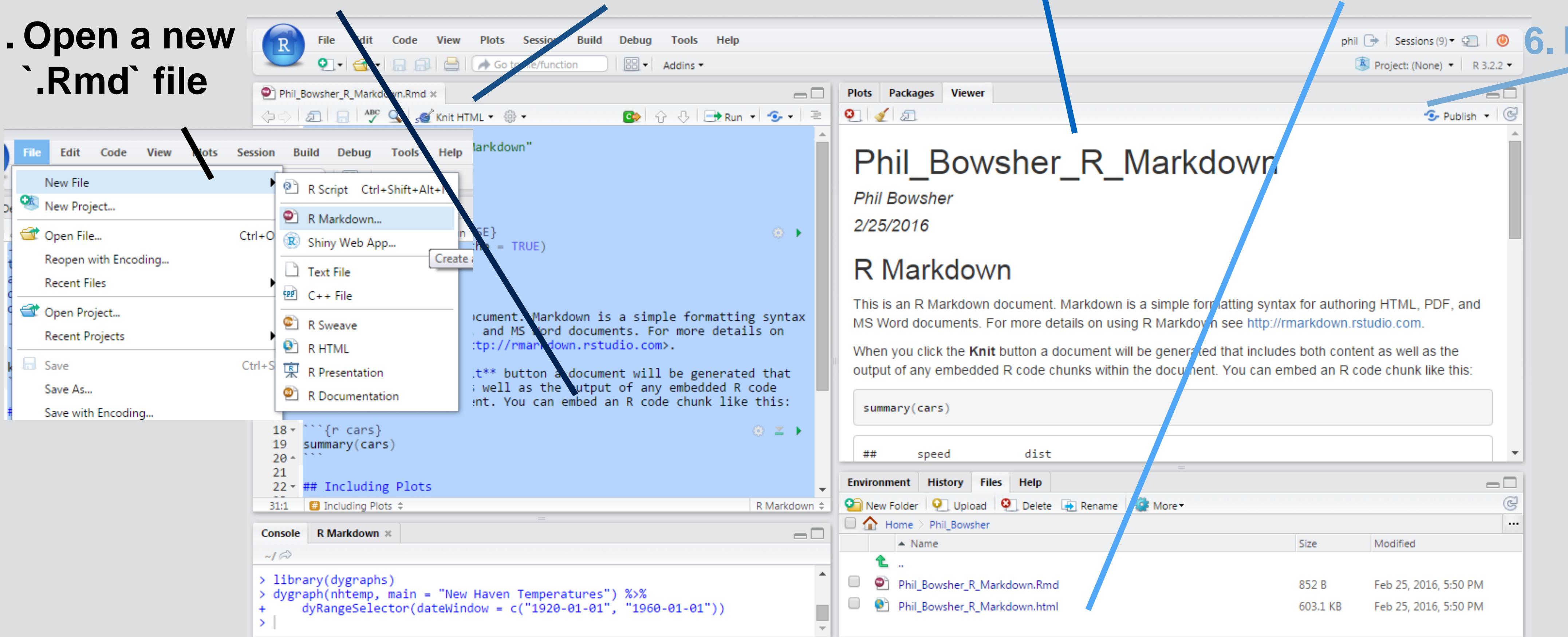


output templates

html_document	→	HTML
pdf_document	→	pdf
word_document	→	Microsoft Word (.docx)
odt_document	→	Open Document Text
rtf_document	→	Rich Text Format
md_document	→	markdown (converts R output to markdown)
ioslides_presentation	→	ioslides (HTML 5 slideshow)
slidy_presentation	→	slidy (HTML 5 slideshow)
beamer_presentation	→	beamer (pdf slideshow)

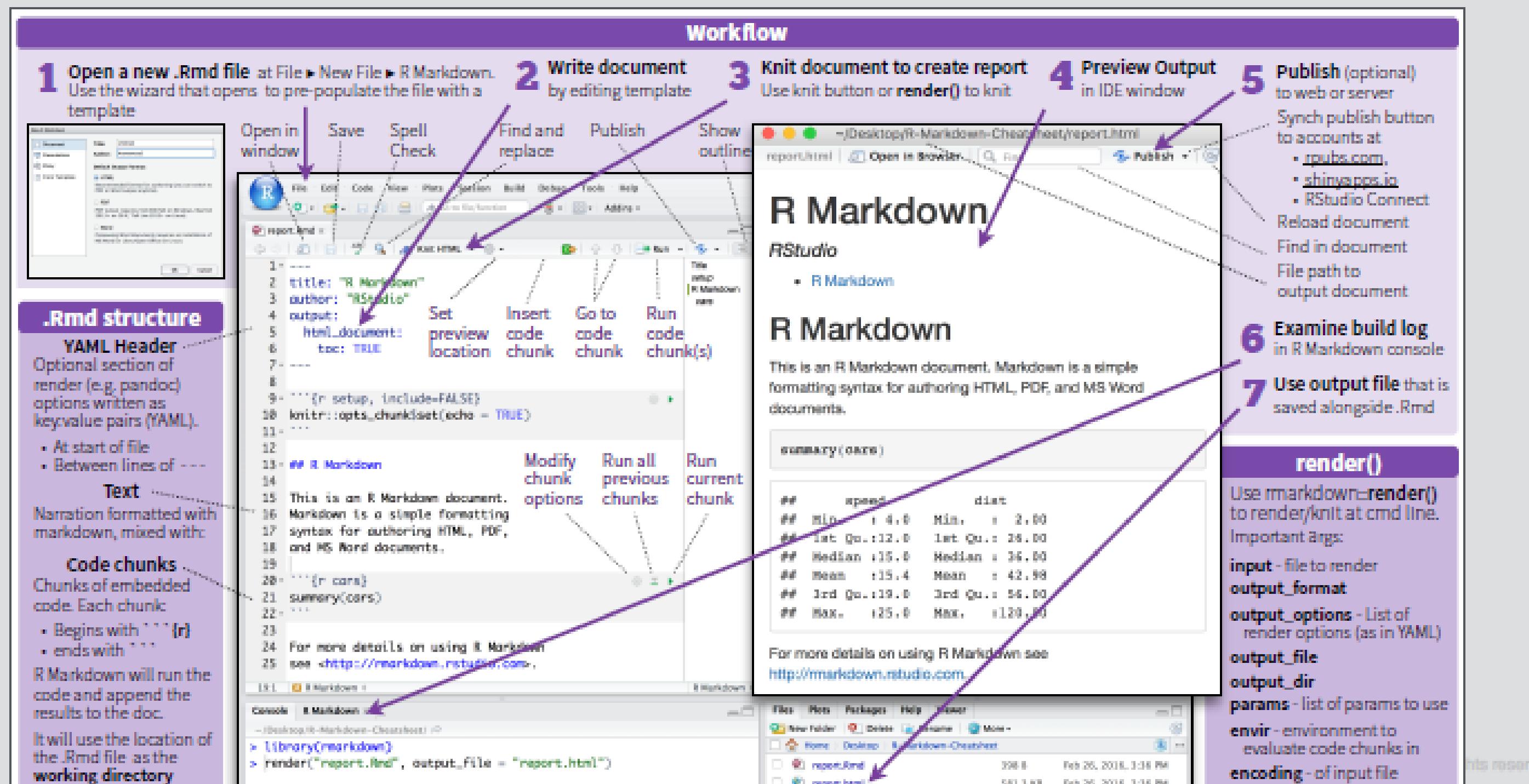
demo

1. Open a new `Rmd` file
2. Edit text
3. Click "Knit HTML"
4. Preview output
5. Access output file
6. Publish



Your turn

Launch RStudio, and open your first R Markdown script.
Make sure you can knit the script into finished HTML output.



rmarkdown::render

Render at the command line with YAML options

```
> render("doc.Rmd")
```

Render at the command line, override output format.

```
> render("doc.Rmd", "html_document")
```

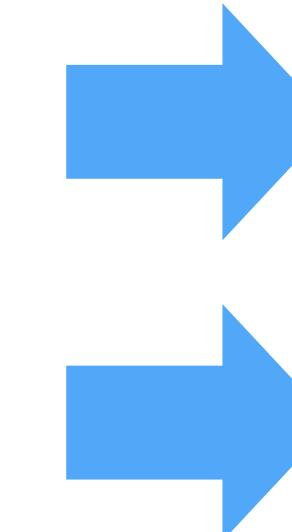
Render at the command line to multiple formats.

```
> render("doc.Rmd", c("html_document", "pdf_document"))
```

Parameters

A list of values that you can call in R code chunks

params list
**elements and
values**



```
...  
title: "Untitled"  
output: html_document  
params:  
  filename: "data.csv"  
  symbol: "GOOG"  
...
```

Access as `params$filename` and `params$symbol`

rmarkdown::render

Render at the command line with YAML options

```
> render("doc.Rmd")
```

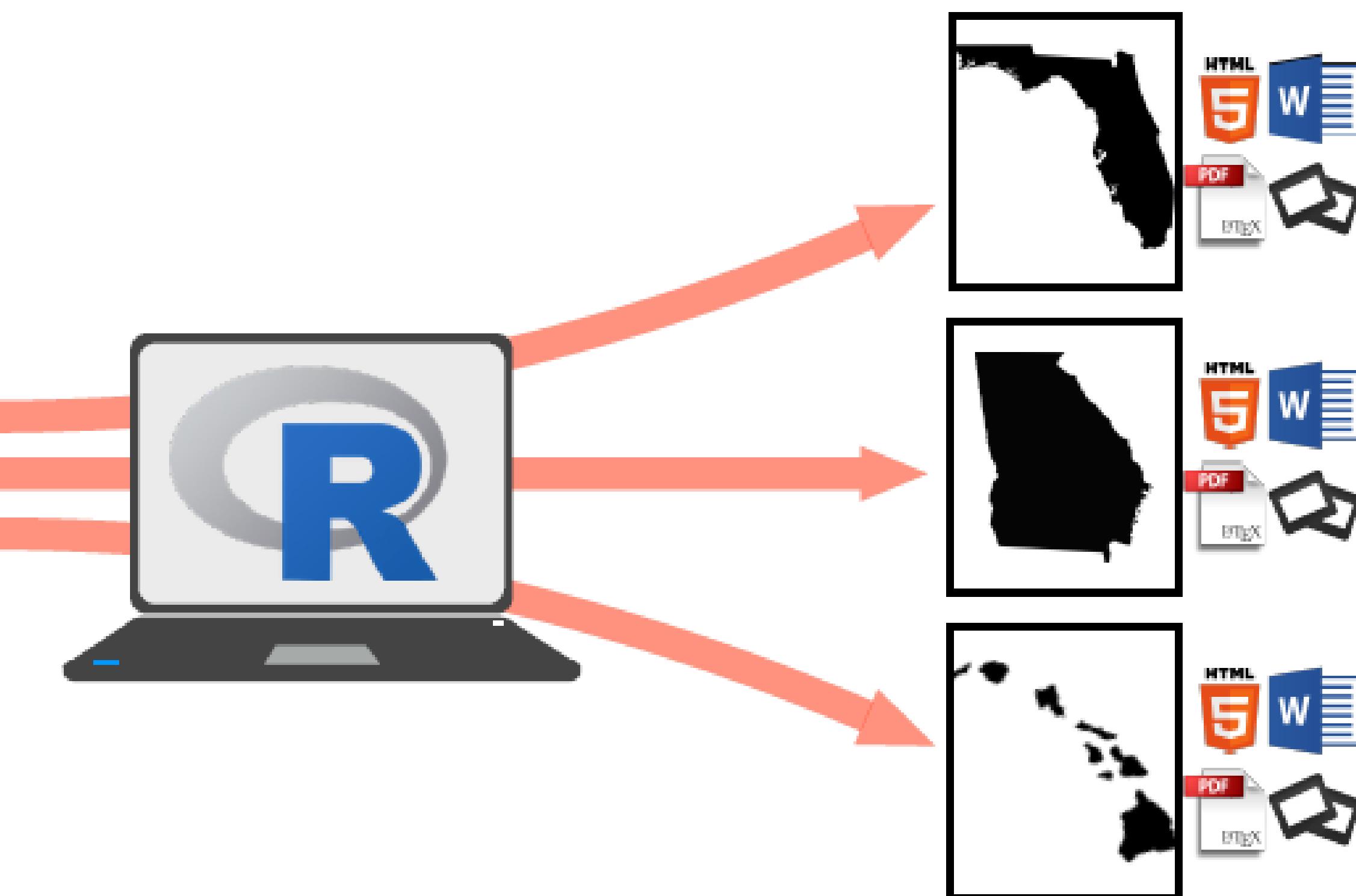
Render at the command line, set parameters.

```
> render("doc.Rmd", params = list(  
  filename = "other_data.csv",  
  symbol = "AAPL")
```

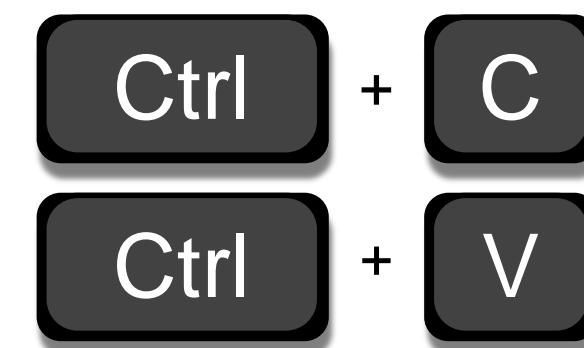
R Markdown

Parameters

- + State = Flo
- + State = Ge
- + State = Ha...



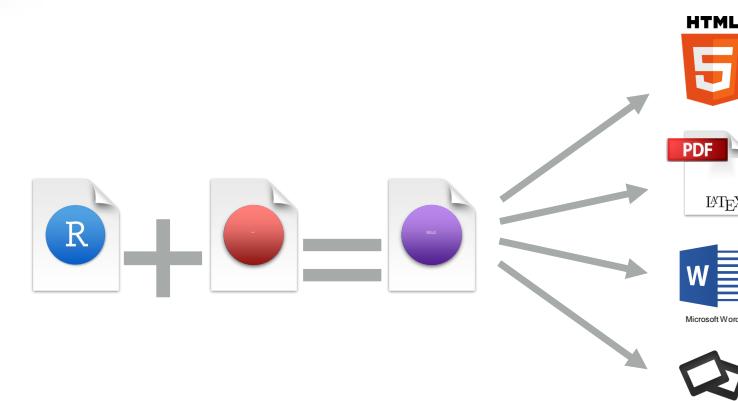
Recap: R Markdown



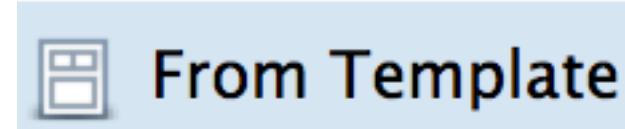
Reproducible



Automatic



Flexible



Reusable

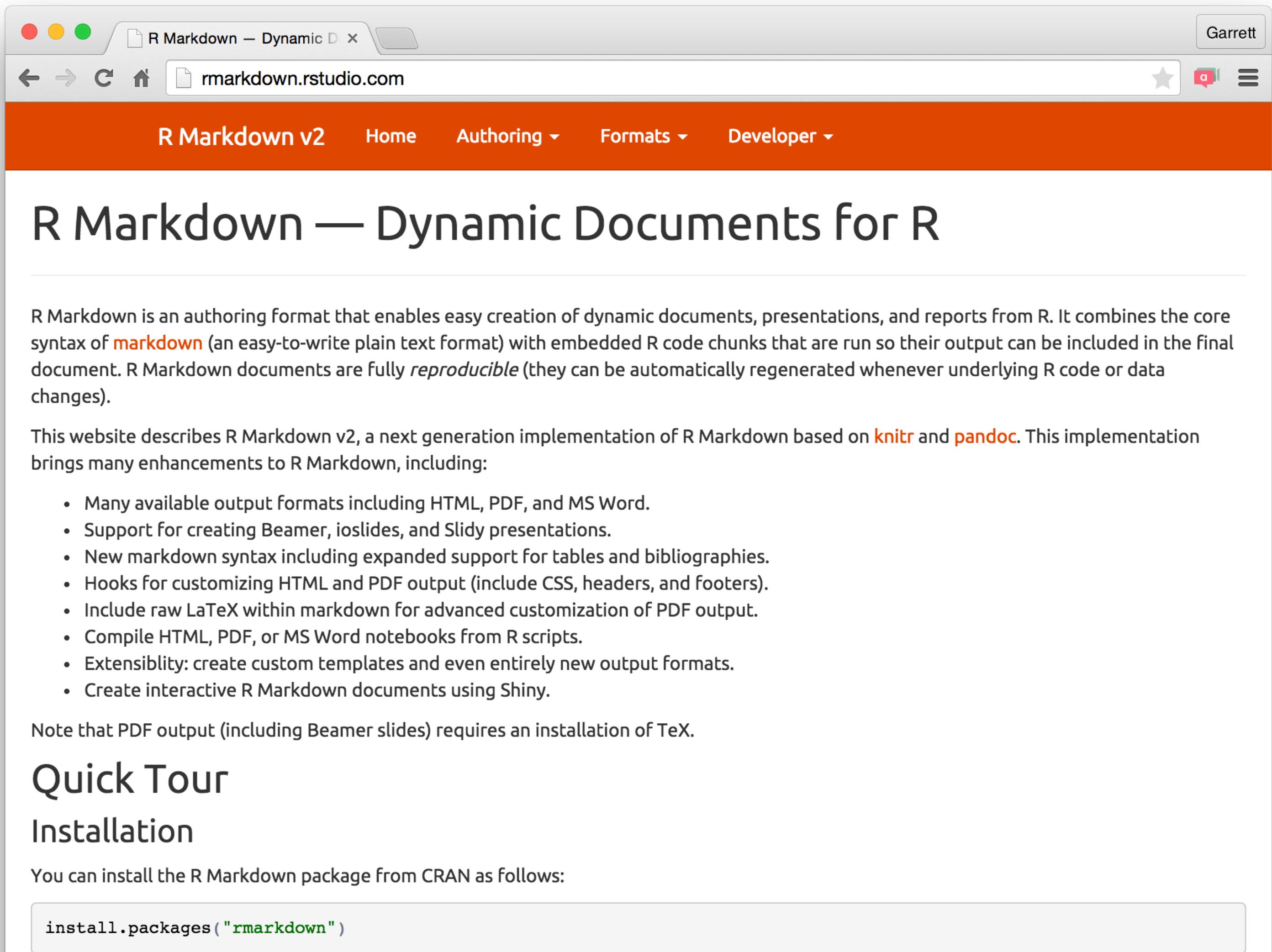
params:

Parameterizable

**Teach
yourself
R Markdown**

The R Markdown Development Center

rmarkdown.rstudio.com



A screenshot of a web browser displaying the R Markdown website (rmarkdown.rstudio.com). The browser window has a title bar showing "R Markdown – Dynamic" and a user profile "Garrett". The address bar shows the URL. The page itself has a header with navigation links for "R Markdown v2", "Home", "Authoring", "Formats", and "Developer". The main content area features a large heading "R Markdown — Dynamic Documents for R". Below it is a paragraph explaining what R Markdown is, followed by another paragraph about the R Markdown v2 implementation. A bulleted list details various features and enhancements. At the bottom, there's a note about TeX requirements and two sections: "Quick Tour" and "Installation". A code snippet for installing the package is shown at the very bottom.

R Markdown is an authoring format that enables easy creation of dynamic documents, presentations, and reports from R. It combines the core syntax of [markdown](#) (an easy-to-write plain text format) with embedded R code chunks that are run so their output can be included in the final document. R Markdown documents are fully *reproducible* (they can be automatically regenerated whenever underlying R code or data changes).

This website describes R Markdown v2, a next generation implementation of R Markdown based on [knitr](#) and [pandoc](#). This implementation brings many enhancements to R Markdown, including:

- Many available output formats including HTML, PDF, and MS Word.
- Support for creating Beamer, ioslides, and Slidy presentations.
- New markdown syntax including expanded support for tables and bibliographies.
- Hooks for customizing HTML and PDF output (include CSS, headers, and footers).
- Include raw LaTeX within markdown for advanced customization of PDF output.
- Compile HTML, PDF, or MS Word notebooks from R scripts.
- Extensibility: create custom templates and even entirely new output formats.
- Create interactive R Markdown documents using Shiny.

Note that PDF output (including Beamer slides) requires an installation of TeX.

Quick Tour

Installation

You can install the R Markdown package from CRAN as follows:

```
install.packages("rmarkdown")
```

The R Markdown Reference Guide

www.rstudio.com/resources/cheatsheets/

The screenshot shows the R Markdown Reference Guide website. It features a large 'R' logo on the left, followed by the title 'R Markdown Reference Guide'. Below the title are links to 'Learn more about R Markdown' at rmarkdown.rstudio.com and 'Learn more about Interactive Docs' at shiny.rstudio.com/articles. To the right is a 'Contents' sidebar with links to 'Markdown Syntax', 'Knitr chunk options' (which is bolded), and 'Pandoc options'. The main content area is divided into two columns: 'Syntax' and 'Becomes'. Under 'Syntax', there are examples for code chunks with three back ticks, inline code with single back ticks, and code chunks with chunk options. Each example shows the R code on the left and its resulting output on the right. At the bottom, there's a table titled 'Chunk options' with rows for 'Code evaluation' and 'Results'.

R Markdown Reference Guide

Learn more about R Markdown at rmarkdown.rstudio.com
Learn more about Interactive Docs at shiny.rstudio.com/articles

Syntax **Becomes**

Make a code chunk with three back ticks followed by an r in braces. End the chunk with three back ticks:

```
```{r}  
paste("Hello", "World!")
```
```

Place code inline with a single back ticks. The first back tick must be followed by an R, like this `r paste("Hello", "World!")`.

Add chunk options within braces. For example, `echo=FALSE` will prevent source code from being displayed:

```
```{r eval=TRUE, echo=FALSE}  
paste("Hello", "World!")
```
```

Make a code chunk with three back ticks followed by an r in braces. End the chunk with three back ticks:

```
paste("Hello", "World!")
```

[1] "Hello World!"

Place code inline with a single back ticks. The first back tick must be followed by an R, like this Hello World!.

Add chunk options within braces. For example, `echo=FALSE` will prevent source code from being displayed:

```
## [1] "Hello World!"
```

Learn more about chunk options at <http://yihui.name/knitr/options>

| Chunk options | | |
|------------------------|---------------|---|
| option | default value | description |
| Code evaluation | | |
| <code>child</code> | NULL | A character vector of filenames. Knitr will knit the files and place them into the main document. |
| <code>code</code> | NULL | Set to R code. Knitr will replace the code in the chunk with the code in the code option. |
| <code>engine</code> | 'R' | Knitr will evaluate the chunk in the named language, e.g. <code>engine = 'python'</code> . Run <code>names(knitr::knit_engines\$get())</code> to see supported languages. |
| <code>eval</code> | TRUE | If FALSE, knitr will not run the code in the code chunk. |
| <code>include</code> | TRUE | If FALSE, knitr will run the chunk but not include the chunk in the final document. |
| <code>purl</code> | TRUE | If FALSE, knitr will not include the chunk when running <code>purl()</code> to extract the source code. |
| Results | | |
| <code>collapse</code> | FALSE | If TRUE, knitr will collapse all the source and output blocks created by the chunk into a single block. |
| <code>echo</code> | TRUE | If FALSE, knitr will not display the code in the code chunk above it's results in the final document. |
| | | If ' <code>hide</code> ' knitr will not display the code's results in the final document. If ' <code>hold</code> ', knitr will delay displaying all output |

The R Markdown Cheat Sheet

www.rstudio.com/resources/cheatsheets/

R Markdown Cheat Sheet
learn more at rmarkdown.rstudio.com
rmarkdown 0.2.50 Updated: 8/14

R Studio

1. Workflow R Markdown is a format for writing reproducible, dynamic reports with R. Use it to embed R code and results into slideshows, pdfs, html documents, Word files and more. To make a report:

- Open** - Open a file that uses the .Rmd extension.
- Write** - Write content with the easy to use R Markdown syntax
- Embed** - Embed R code that creates output to include in the report
- Render** - Replace R code with its output and transform the report into a slideshow, pdf, html or ms Word file.

2. Open File Start by saving a text file with the extension .Rmd, or open an RStudio Rmd template

- In the menu bar, click **File > New File > R Markdown...**
- A window will open. Select the class of output you would like to make with your .Rmd file
- Select the specific type of output to make with the radio buttons (you can change this later)
- Click OK

3. Markdown Next, write your report in plain text. Use markdown syntax to describe how to format text in the final report.

| syntax | becomes |
|--|---|
| Plain text
End a line with two spaces to start a new paragraph.
<code>*italics*</code> and <code>_italics_</code>
<code>**bold**</code> and <code>__bold__</code>
<code>superscript²</code>
<code>--strikethrough--</code>
<code>[link](www.rstudio.com)</code> | Plain text
End a line with two spaces to start a new paragraph.
<i>italics</i> and <i>italics</i>
bold and bold
^{superscript²}
strikethrough
link |
| # Header 1 | Header 1 |
| ## Header 2 | Header 2 |
| ### Header 3 | Header 3 |
| #### Header 4 | Header 4 |
| ##### Header 5 | Header 5 |
| ###### Header 6 | Header 6 |
| endash: -- | endash: – |
| emdash: --- | emdash: — |
| ellipsis: ... | ellipsis: ... |
| inline equation: \$A = \pi * r^2\$ | inline equation: $A = \pi * r^2$ |
| image: | image: |
| horizontal rule (or slide break): | horizontal rule (or slide break): |
| *** | *** |
| > block quote | > block quote |
| * unordered list | * unordered list |
| * item 2 | * item 2 |
| + sub-item 1 | + sub-item 1 |
| + sub-item 2 | + sub-item 2 |
| 1. ordered list | 1. ordered list |
| 2. item 2 | 2. item 2 |
| + sub-item 1 | + sub-item 1 |
| + sub-item 2 | + sub-item 2 |
| Table Header Second Header | Table Header Second Header |
| ----- | ----- |
| Table Cell Cell 2 | Table Cell Cell 2 |
| Cell 3 Cell 4 | Cell 3 Cell 4 |

4. Choose Output Write a YAML header that explains what type of document to build from your R Markdown file.

YAML
A YAML header is a set of key: value pairs at the start of your file. Begin and end the header with a line of three dashes (- - -)

```
title: "Untitled"
author: "Anonymous"
output: html_document
```

The RStudio template writes the YAML header for you

The output value determines which type of file R will build from your .Rmd file (in Step 6)

- `output: html_document` html file (web page)
- `output: pdf_document` pdf document
- `output: word_document` Microsoft Word .docx
- `output: beamer_presentation` beamer slideshow (pdf)
- `output: ioslides_presentation` ioslides slideshow (html)

RStudio® is a trademark of RStudio, Inc. • All rights reserved • info@rstudio.com • 844-448-1212 • rstudio.com

htmlwidgets

&

R Graphics

htmlwidgets for R:

- R bindings to JavaScript libraries
- Used to create interactive visualizations
- A line or two of R code is all it takes to produce an example

Use htmlwidgets in:

- RStudio viewer pane
- R Markdown files
- Shiny Apps

www.htmlwidgets.org

htmlwidgets

The screenshot shows the homepage of www.htmlwidgets.org. At the top, there's a navigation bar with links for Home, Showcase, Develop, and GitHub. Below the navigation, a main heading reads "htmlwidgets for R". To the right, there's a section titled "Bring the best of JavaScript data visualization to R" with sub-points about using JavaScript visualization libraries at the R console, embedding widgets in R Markdown documents and Shiny web applications, and developing new widgets using a framework that bridges R and JavaScript.

On the left side of the main content area, there's a heading "Widgets in action" followed by four small images illustrating different types of interactive visualizations: a map with colored circles, a line chart, a network graph, and a 3D surface plot.

At the bottom, a purple button says "See the showcase »".

At the very bottom of the page, a footer bar contains the text "javascript:void(0)" and the copyright notice "© 2015 RStudio, Inc. All rights reserved. CC-BY-SA".

htmlwidgets gallery

<http://gallery.htmlwidgets.org/>

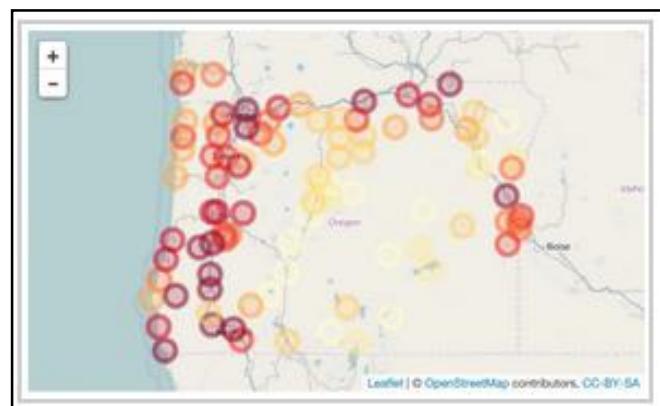
The screenshot shows the main interface of the htmlwidgets gallery. At the top, there are filters for Sort (Github stars), Text Filter, Author Filter, Tag Filter, and CRAN Only. Below the filters, it says "Showing 55 of 55".

- DiagrammeR** (Star 366)
With DiagrammeR, you can easily create graph diagrams using R.
■ author: rich-iannone
■ tags: visualization, diagram
■ is libraries: d3.viz.mermaid
- leaflet** (Star 141)
Leaflet is an open-source JavaScript library for interactive maps. This R package makes it easy to create Leaflet maps from R.
■ author: rstudio
■ tags: visualization, maps
- networkD3** (Star 125)
A port of Christopher Gandrud's d3Network package to the htmlwidgets framework.
■ author: christophergandrud
■ tags: visualization, networks
■ is libraries: d3

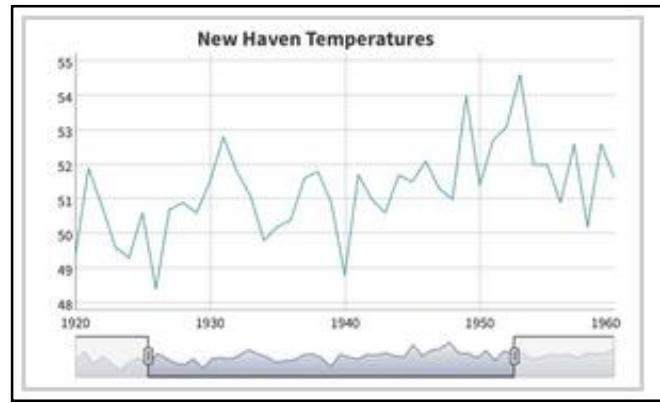


Packages htmlwidgets

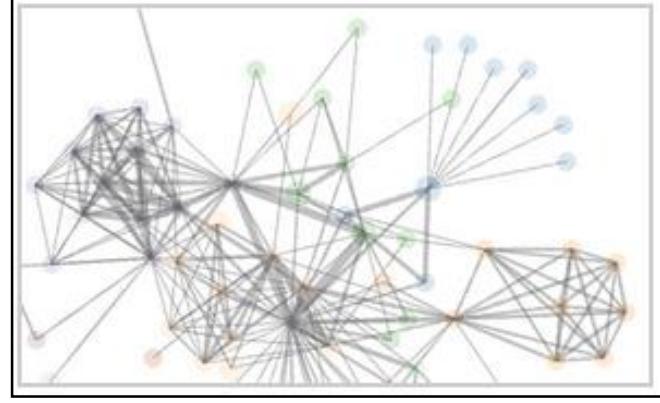
Ready-to-use htmlwidgets:



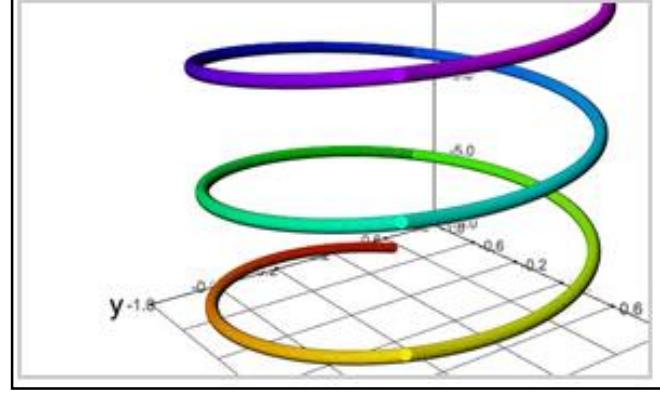
Leaflet - Interactive maps
rstudio.github.io/leaflet/



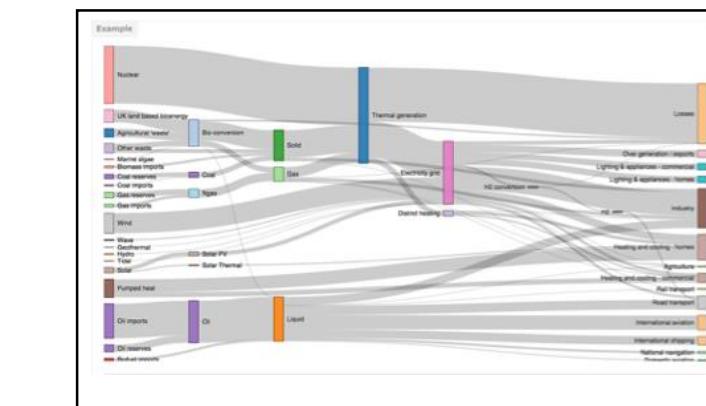
dygraphs - Time series
rstudio.github.io/dygraphs



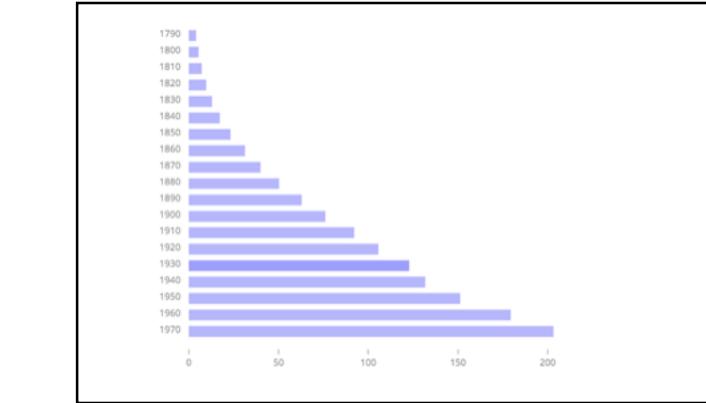
networkD3 - network graphs
christophergandrud.github.io/networkD3/



threejs - 3D charts
github.com/bwlewis/rthreejs



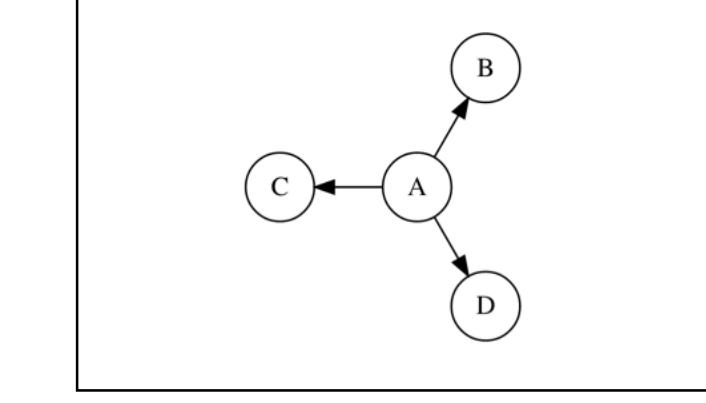
rCharts - Various charts
[rCharts.io](https://rcharts.io)



MetricsGraphics - d3 charts
hrbrmstr.github.io/metricsgraphics/

| Show 10 entries | | | | Search: |
|-------------------------------|-------------|--------------|-------------|--------------------------------|
| Sepal.Length | Sepal.Width | Petal.Length | Petal.Width | Species |
| 5.1 | 3.5 | 1.4 | 0.2 | setosa |
| 4.9 | 3 | 1.4 | 0.2 | setosa |
| 4.7 | 3.2 | 1.3 | 0.2 | setosa |
| 4.6 | 3.1 | 1.5 | 0.2 | setosa |
| 5 | 3.6 | 1.4 | 0.2 | setosa |
| Showing 1 to 5 of 150 entries | | | | Previous 1 2 3 4 5 ... 30 Next |

DT - Data tables
rstudio.github.io/DT/



Diagrammer - Diagrams
rich-iannone.github.io/DiagrammeR/

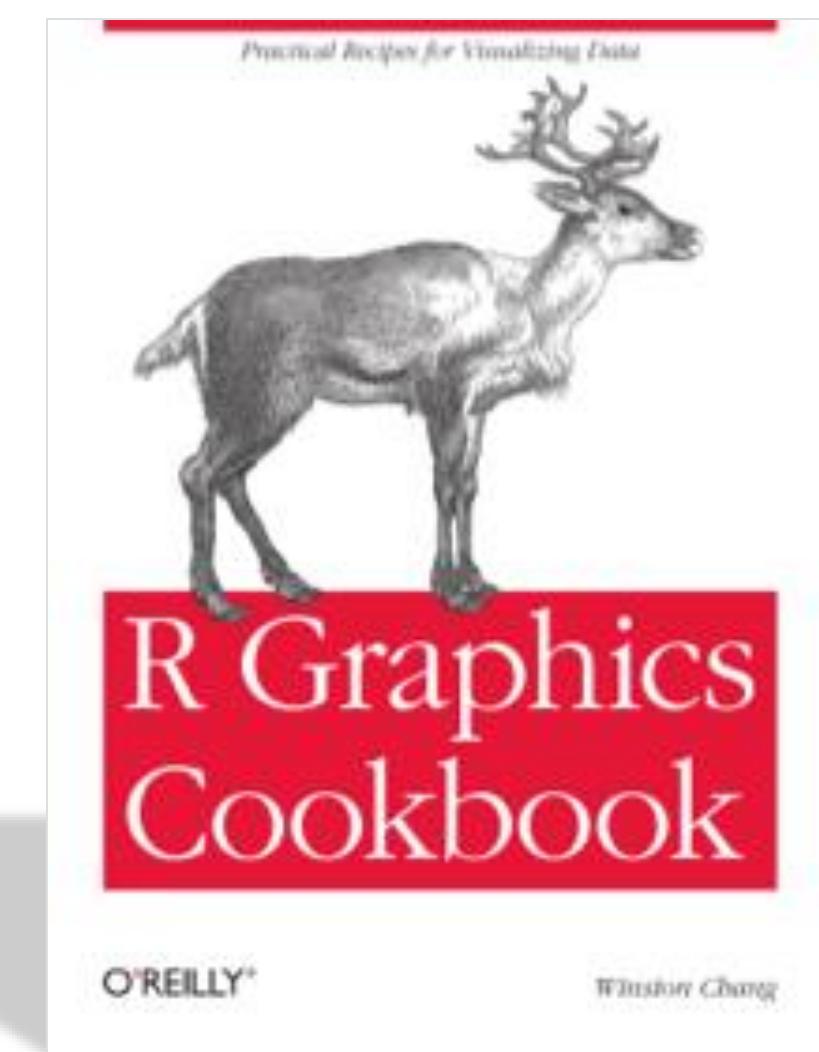
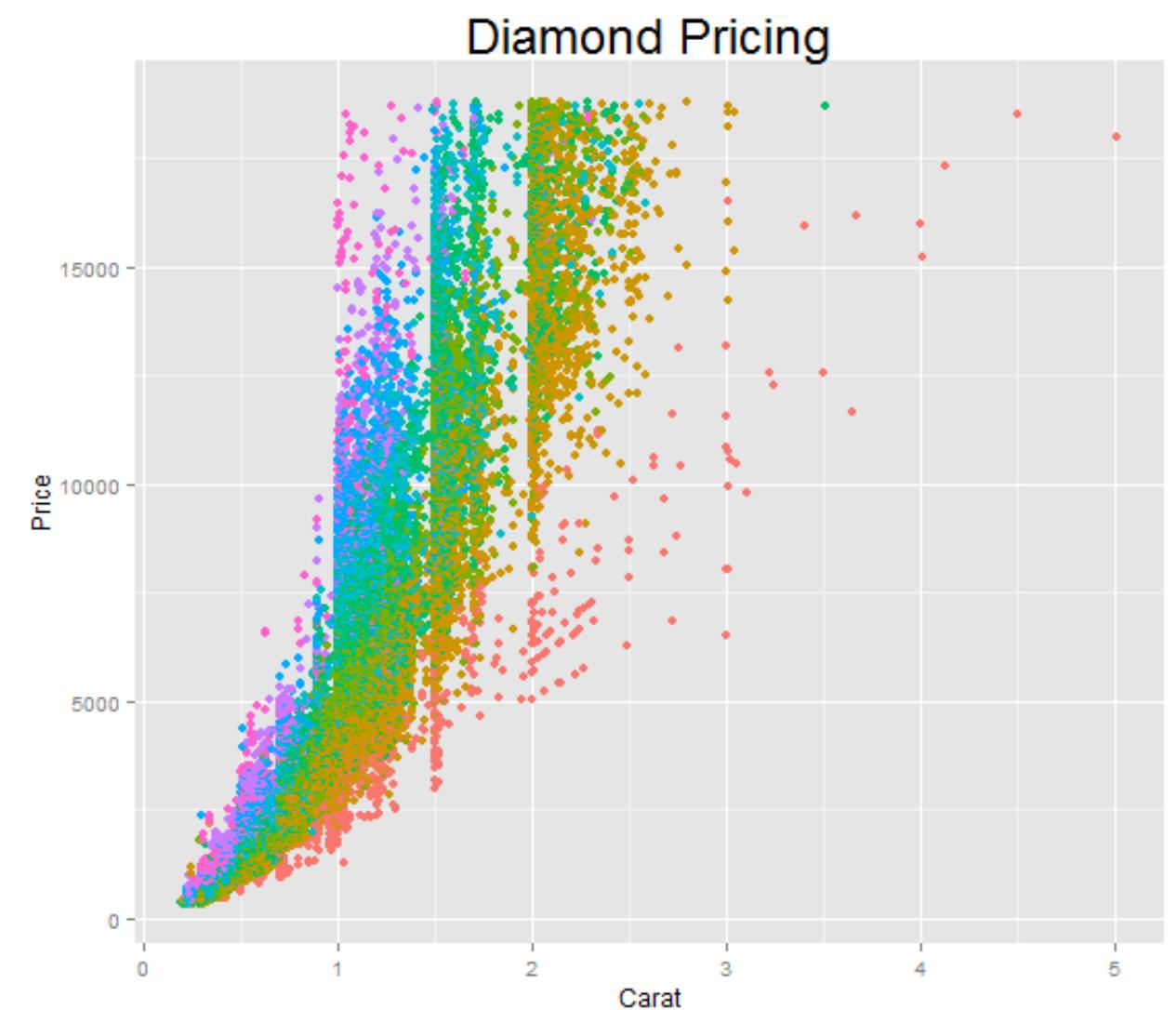
R Graphics:

Four Main Graphical Systems in R:

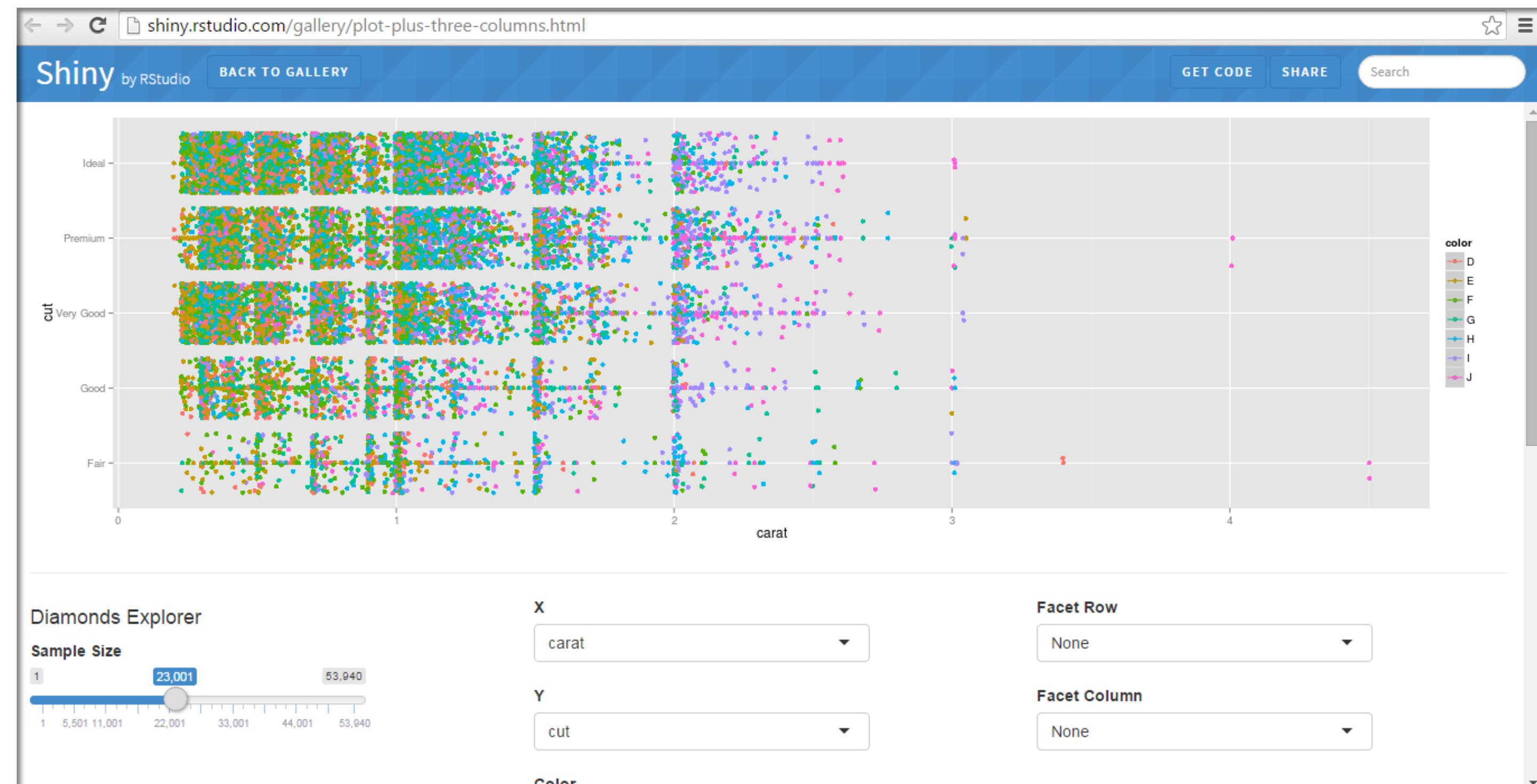
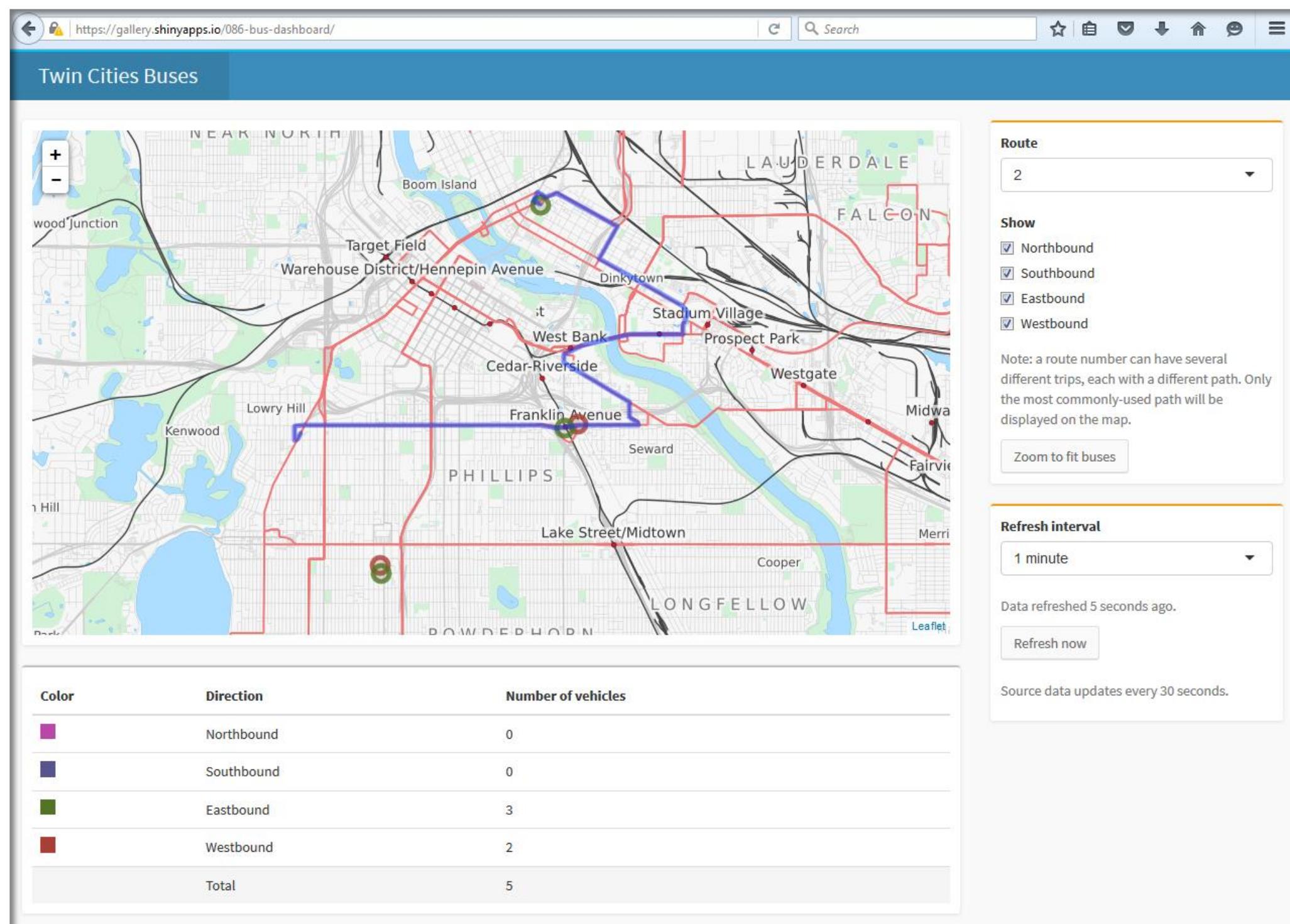
- R's Base Graphics
- Grid Graphics System
- The lattice Package
- The ggplot2 Package – Created by Hadley Wickham

Why ggplot2?

- Consistent underlying - Grammar of Graphics (Wilkinson, 2005)
- Very flexible
- Mature and complete graphics system
- Many users, active mailing list
- <http://www.cookbook-r.com> & <http://tutorials.iq.harvard.edu/R/Rgraphics/Rgraphics.html>



demos



Application in Drug Development

Appendix

WHAT'S NEW? RStudio IDE (0.99)

New packages

Data import:

- readr
- haven
- rvest
- purr

Data tidying:

- tidyverse

Web graphics:

- htmlwidgets
- crosstalk
- tufte
- profvis

General Enhancements

- dplyr
- RMySQL
- RSQLite
- ggvis
- httr
- packrat
- testthat
- ggplot2

Shiny Enhancements

- Shiny Gadgets
- Shiny Dashboard
- Shiny Modules
- Templates
- miniUI
- Flexdashboard

R Markdown Resources

File > New File > R Markdown...

<http://rmarkdown.rstudio.com/>

<http://www.rstudio.com/resources/cheatsheets/>

http://rmarkdown.rstudio.com/developer_document_templates.html

http://rmarkdown.rstudio.com/developer_parameterized_reports.html

htmlwidgets Resources

htmlwidgets.org

http://www.htmlwidgets.org/showcase_leaflet.html

Shiny Resources

shiny.rstudio.com/articles

shiny.rstudio.com/tutorial

<http://www.rstudio.com/resources/cheatsheets/>

Shiny Demos

<http://www.rstudio.com/products/shiny/shiny-user-showcase/>

<http://shiny.rstudio.com/gallery/>

<http://webpopix.org:8080/dashboard/absorption/>

<http://shiny.rstudio.com/gallery/widget-gallery.html>

<http://rstudio.github.io/shinydashboard/>

<https://gallery.shinyapps.io/marketing/>

<https://gallery.shinyapps.io/LDAelite/>

<http://shiny.rstudio.com/gallery/superzip-example.html>

<https://gallery.shinyapps.io/EDsimulation/>

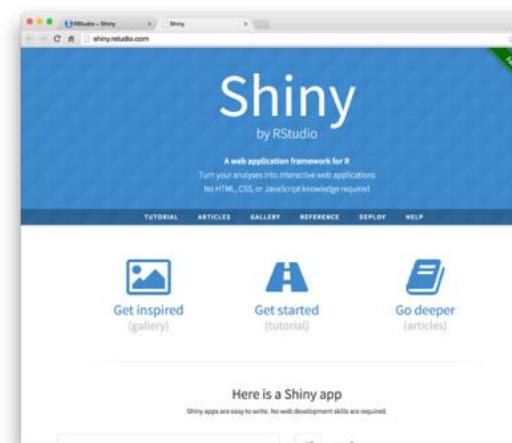
<http://shiny.rstudio.com/articles/#interactive-plots>

<http://shiny.rstudio.com/gallery/authentication-and-database.html>

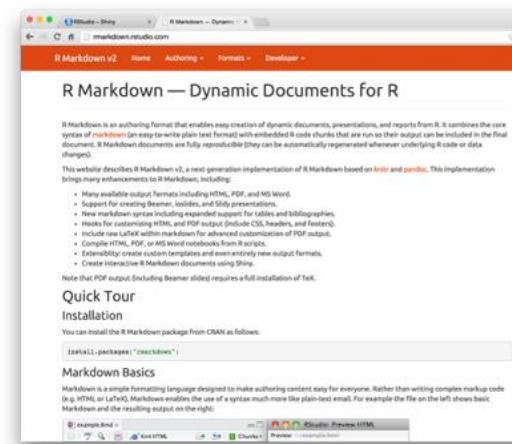
<https://gallery.shinyapps.io/ga-effect/>

<http://gallery.shinyapps.io/087-crandash>

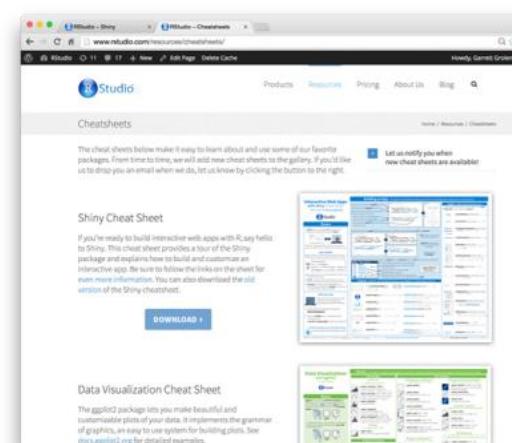
Useful websites



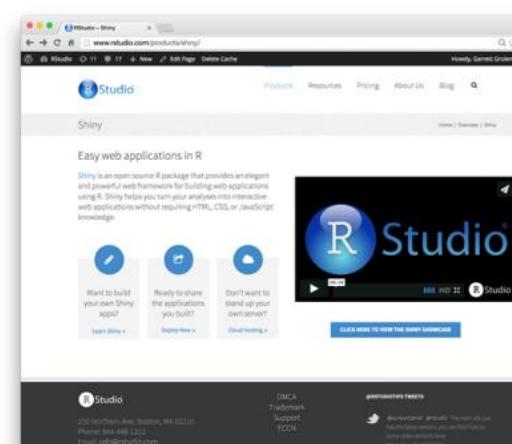
The Shiny development center:
shiny.rstudio.com



The R Markdown development center:
rmarkdown.rstudio.com



Shiny and R Markdown cheat sheets:
www.rstudio.com/resources/cheatsheets



RStudio products:
www.rstudio.com/products/shiny

OTHER GOODIES

Latest Webinars

Getting your data into R

You can't use R for data analysis unless you can get your data into R. Getting your data into R can be a major hassle, so in the last few months Hadley Wickham has been working hard to make it easier.

In this webinar Hadley will discuss the places you most often find data (databases, excel, text files, other statistical packages, web apis, and web pages) and the packages (DBI, xml2, jsonlite, haven, readr, exel) that make it easy to get your data into R.

[GitHub Webinar Repository](#)

Materials for this specific webinar

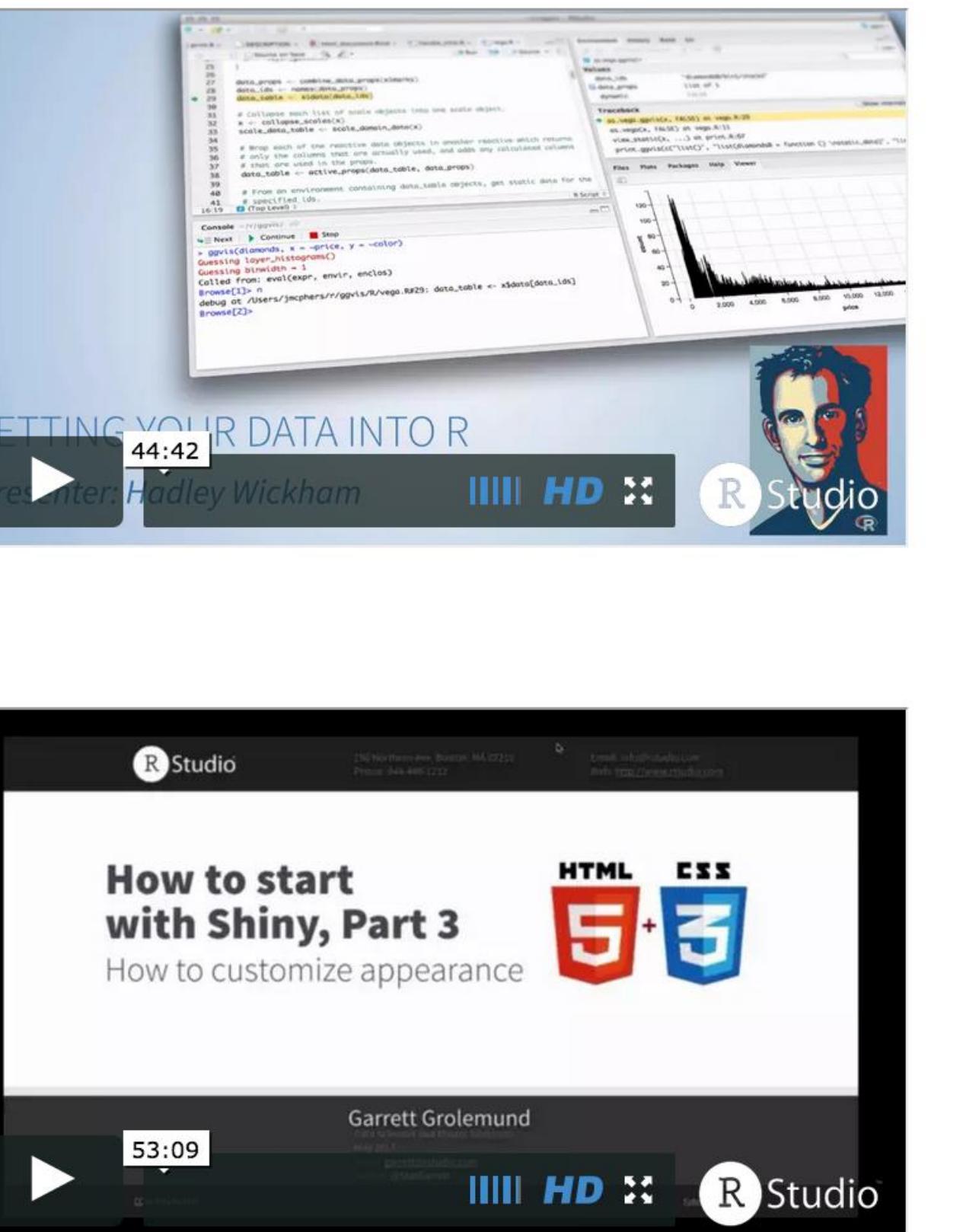
How to start with Shiny – Part 3

In this talk (Part 3 of 3), Garrett Grolemund will show you how to customize the appearance of your app. You will learn how to arrange the components of your app into an attractive layout, as well as how to change the appearance of text, images, and other HTML elements in your app.

[GitHub Webinar Repository](#)

Materials for this specific webinar

Alternate Link: bit.ly/shiny-quickstart-3



The collage includes the following sections:

- Data Visualization with ggplot2 Cheat Sheet**: Shows examples for Geoms (One Variable, Two Variables) and Geom Functions.
- Package Development with devtools Cheat Sheet**: Details for DESCRIPTION file, Package Structure, and devtools:use_package().
- Data Wrangling with dplyr and tidyverse Cheat Sheet**: Explains Tidy Data principles and Syntax (dplyr, tibble, dft).
- R Markdown Cheat Sheet**: Workflow (Open, Write, Embed, Render), syntax (R, HTML, CSS, JS), and output formats (HTML, PDF, Word).
- Shiny Cheat Sheet**: Structure (app-name, server.R, ui.R, README, www), server.R (shinyServer), and reactivity (reactive, observe, isolate).