

Oct 2024

R Shiny Masterclass Series - Introduction

Getting started with Posit Cloud and your first R Shiny app



EPI-interactive

How we run the workshop

- Zoom
- Posit Cloud: code, assignments
 - <https://posit.cloud/spaces/558537/content>
 - Learning platform: slides, exercises, discussion forum
- Agenda

How we run the workshop

- Recording of sessions
- We encourage you to turn on video
- Please mute your mic when you are not speaking
- Ask your question in the Zoom chat – will be addressed either throughout or at the end of each section
- At the end – use chat or raise your hand for questions and comments (don't forget to un-mute)
- Polls and breakout rooms – screen sharing encouraged!

How we run the workshop

Learning platform - Course material, assignments and forum

<https://learn.epi-interactive.com>



Introduction to R Shiny Masterclass, Sept 2024

[Home](#) / [My courses](#) / [R Shiny Masterclass Sept '24](#) / [Introduction to R Shiny Masterclass, Sept 2024](#)

[Course overview](#) 

[Edit](#) ▼

This online masterclass will introduce you to R Shiny programming and will cover R Shiny capabilities, design approaches, coding essentials and how to publish your newly created app. Case studies will be used to provide applied examples of R Shiny apps in action.

Introduce yourself!

- Name
- Organization
- Why are you joining the Masterclass?
- What are you looking forward to the most?
- Fun fact

Agenda

- **Session 1** | 30 September | Getting started with Posit Cloud and your first R Shiny app
- **Session 2** | 01 October | R Shiny core concepts and mobile ready layout
- **Session 3** | 03 October | R Shiny user interface components, reactivity and debugging
- **Session 4** | 07 October | Data sources and data processing in R Shiny
- **Session 5** | 08 October | Interactive charts with Plotly: chart types, customising hover boxes and chart styling
- **Session 6** | 10 October | Maps and spatial visualisation with Leaflet: adding map layers, annotations, pins, filters and legend
- **Session 7** | 14 October | Publishing R Shiny apps, design considerations and case study
- **Session 8** | 15 October | Case study, top 10 tips for data visualisation with R Shiny and wrap-up

Getting Started

What is R?

- A language and environment
- For statistical computing and graphics
- Open source
- Can be extended (easily) via packages (such as Shiny)
- Currently on version 4.4.1
- <https://www.r-project.org/about.html>



What is Shiny?

“A web application framework for R”



- Website: <https://shiny.rstudio.com/>
- Integrates with the RStudio IDE
- Publishing: Shinyapps.io, Shiny Server or Posit Connect
- Other packages: shinydashboard, shinythemes,
<https://rstudio.github.io/shinydashboard/>
<https://rstudio.github.io/shinythemes/>
<https://github.com/Appsilon/shiny.router>
<https://github.com/grabear/awesome-rshiny>

Why Shiny?

- Publishing interactive data visualisations to the web
- Access to R (analytical power/customisation)
- Open source
- Linkage to common frameworks
HTML, CSS (Bootstrap), JavaScript...
- JavaScript libraries: Plot.ly, Leaflet, DT ...
- Well documented
- Vibrant community



Trends


- Making data more accessible
- Interactive data visualisations
- Bridging the gap between analytics, communication, and decision making

Adding context...

- Telling a story
- Enriching data visualisations
- Context sensitive help
- Scenarios, tutorials

Example - epidemix.app





About Contact

Visually explore spatiotemporal trends in disease transmission and improve your understanding of disease modelling.

Start

Select model type

Deterministic: Homogeneous COMP

Deterministic: Homogeneous COMP

Deterministic: Heterogeneous COMP

Stochastic: Homogeneous COMP

Stochastic: Heterogeneous COMP

Stochastic: Homogeneous IBM

Stochastic: Heterogeneous IBM

Stochastic: Network IBM

edit info

9 different disease models

Select infection states to consider

Current selection: S, I, R

All infected units recover from infection

Removed units are not replaced (closed population selected)

A unit which recovered from infection remains immune until removed

Define host population features

Current selection: Population size = 100

Closed population

Define infection and transmission features

Current selection: Number of infected units at start of simulation = 1

Daily number of effective contacts per unit = 0.4

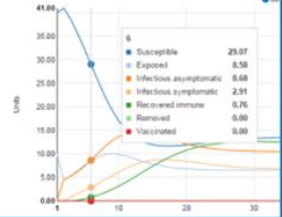
Length of symptomatic infectious period (days) = 10

Population-dependent transmission

Step-by-step parameter selection

Visualisation Data Table

The graph shows the number of units per infection state over time. Roll over



Legend: Susceptible, Exposed, Infectious asymptomatic, Infectious symptomatic, Recovered immune, Removed, Vaccinated

Instantaneous data visualisation

Visualisation Data Table Parameters & Dependencies

Number of units per infection state per timestep. Additionally units removed due to disease [ReD] and vaccination [V]

Download data info

	S	E	Ia	Is	R	ReD	V
0	40.00	10.00	0.00	0.00	0.00	0.00	0.00
0.5	40.50	7.12	2.38	0.00	0.00	0.00	0.00
1	40.05	6.00	3.72	0.23	0.00	0.00	0.00
1.5	39.05	5.78	4.61	0.55	0.02	0.00	0.00
2	37.70	6.02	5.31	0.91	0.07	0.00	0.00

Table view and export options

Posit Cloud & Your First shiny app

Posit Cloud

- Online version of Rstudio Desktop
- Free and paid versions
- Shared workspaces and projects
- Private Collaboration
 - Only one person at a time can edit
- Using assignments means all packages are preinstalled

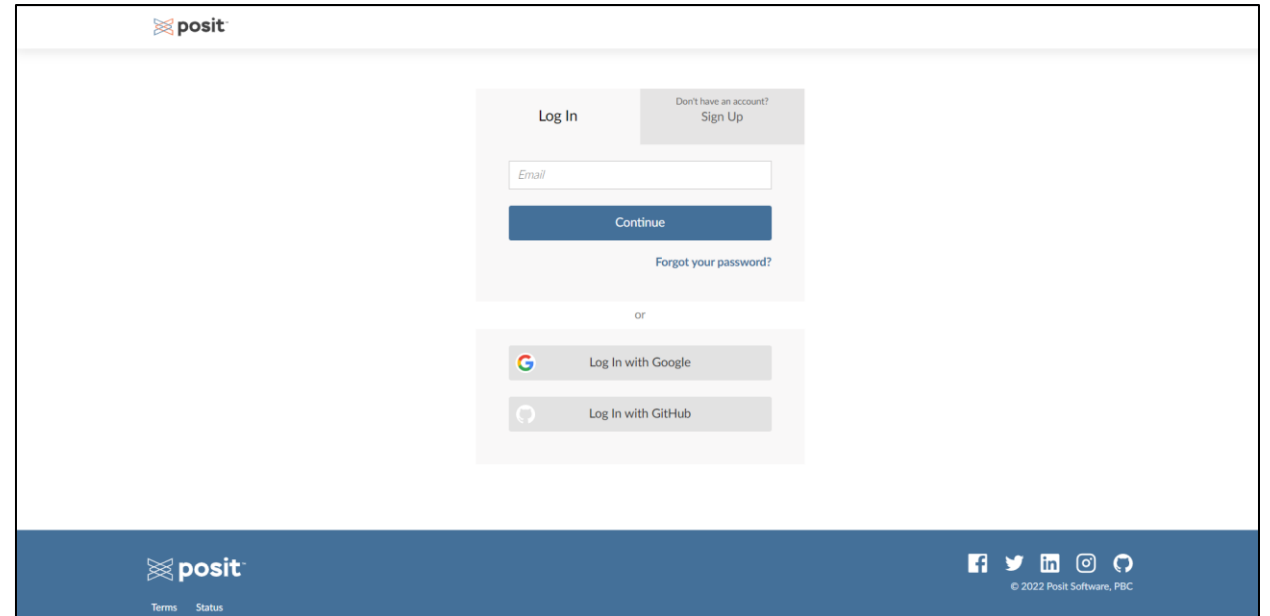
RStudio IDE

- Integrated Development Environment
- Pros:
 - Debugging
 - Integrated with version control
 - Syntax highlighting
 - Auto-complete
 - Code search
 - Built-in Documentation
 - Addins
- RStudio Desktop, Posit Cloud, **Posit Workbench**



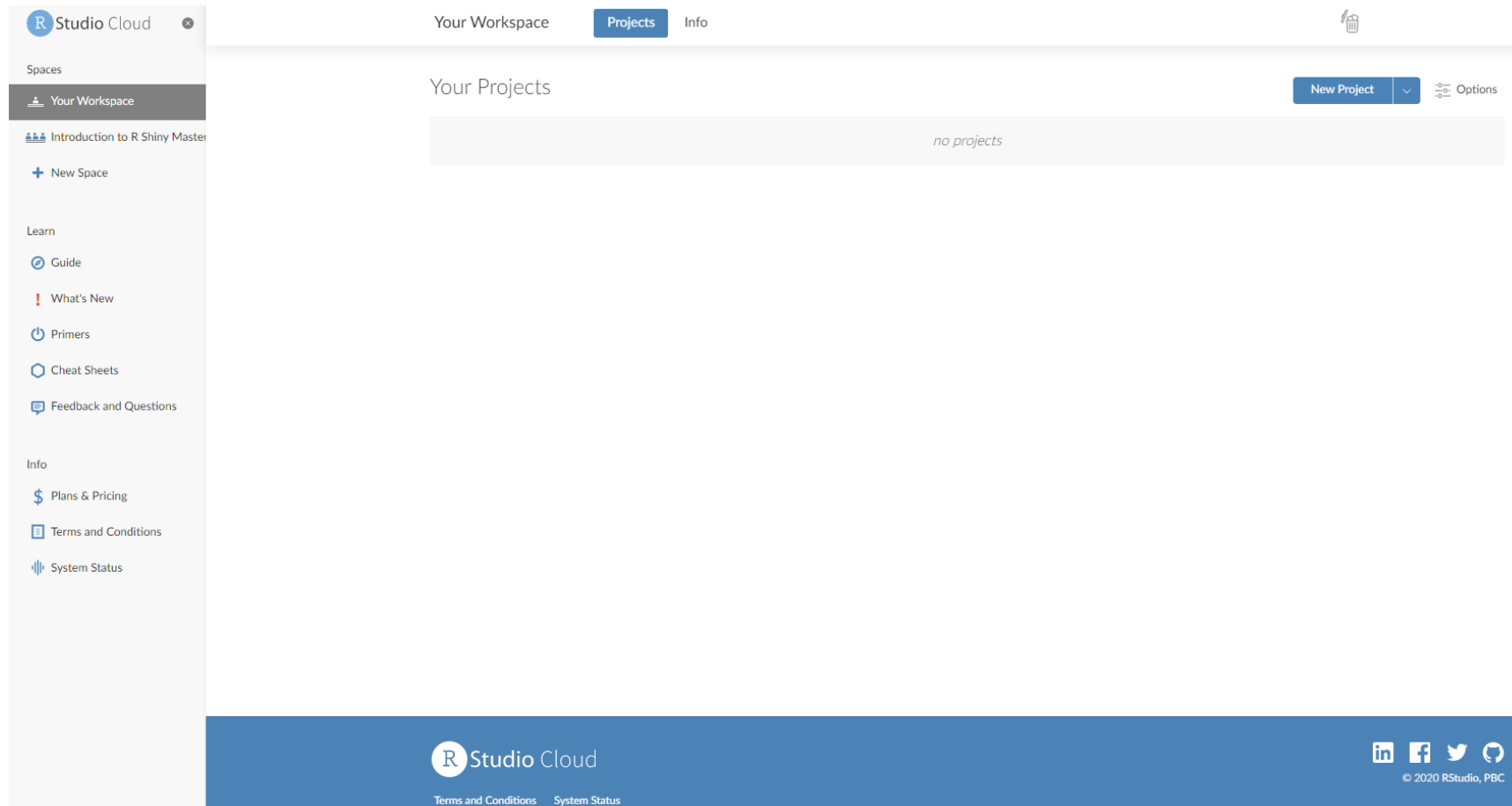
Logging in

- Go to <https://posit.cloud>
- Press “Log in”



The screenshot shows the Posit Cloud login interface. At the top left is the Posit logo. The main content area features a 'Log In' section with a 'Continue' button and a 'Forgot your password?' link. To the right of the 'Log In' section is a 'Sign Up' button. Below the 'Log In' section, there is an 'or' separator and two social login options: 'Log In with Google' and 'Log In with GitHub'. The footer contains the Posit logo, 'Terms' and 'Status' links, social media icons, and copyright information: '© 2022 Posit Software, PBC'.


After logging in



Free account limits

- For custom projects:
 - Create 50 projects total
 - 25 project hours per month
 - Each project is allocated 1GB of RAM and 1 CPU
 - One personal workspace and the option to create one additional shared space
 - The shared space is limited to up to 5 members, and up to 10 projects.
- *Assignments will not cut into your hours or projects*

Checking your stats

 Epi- interactive

YOUR PERSONAL ACCOUNT

PLAN

Cloud Basic


CURRENT USAGE PERIOD


Feb 11, 2024 - Mar 11, 2024


USAGE


Compute hours: 2.5


EPI- INTERACTIVE

 Log Out

 Profile

 Account

 Authentication

 Credentials

Creating your first app

In Posit Cloud:

- (In sidebar) Spaces > “Your Workspace”
- New Project
- File > New File
- You’ll be asked if you want to install Shiny
 - Click yes
- Shiny Web App
 - Name your project
 - Choose “Single file” for now

While we wait, let’s look deeper at the interface...

View Panes

- **Source Editor** - Editing and saving code
- **Console Pane:** Console, Terminal, Jobs
- **Environment Pane:** Environment, History, Connection
- **Files Pane:** Files, Plots, Packages, Help, Viewer

Help

help() / ? – used to search exact name of function

- `help("paste0")`
- `?paste0`

help.search() / ?? – looks through the documentation of currently installed packages in your library

- `help.search("match")`
- `??match`

Keyboard Shortcuts: Posit Cloud

- Indentation: Ctrl+I
- Reformat code: Ctrl+Shift+A
- Comment/uncomment: Ctrl+Shift+C
- Find and replace: Ctrl+Shift+J
- Find in all files: Ctrl+Shift+F

- New file: Ctrl+Shift+Alt+N
- Save all: Ctrl+Alt+S

- Full list: Help > Keyboard Shortcuts Help (Alt+Shift+K)

Creating a simple Shiny app

Anatomy of a Shiny app

There are 3 key actors involved in a Shiny app

1. An end-user who wants to use the Shiny app (*the **User***)
2. A user interface, with which users can interact with data visualisations (*the **UI***)
3. A server function to process user requests in a structured way (*the **Server***)

Anatomy of a Shiny app

- We can think of our Shiny app like a restaurant
- Content in the Shiny app is handled based on user inputs mapping to outputs
- The user interface provides a window for the user to make requests to the server in a managed way
- No direct access to the server



Anatomy of a Shiny app

- Simplest Shiny app definitions will be all in one file: ***app.R***
- We use UI functions from Shiny to create our UI object
 - the look / feel of our application
 - Where certain outputs (plots, tables, maps) will sit on the page
- We write an R function to handle our server logic
 - When the User selects X, how should the plot Y look?
 - Input driven (lazy, more on this later)

Our first Shiny app

app.R

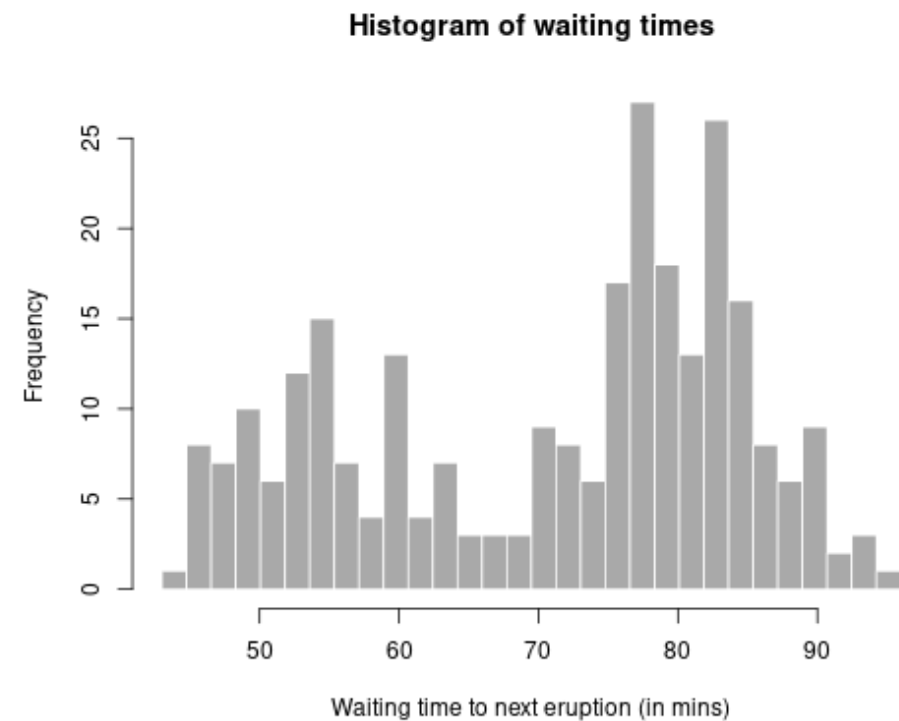
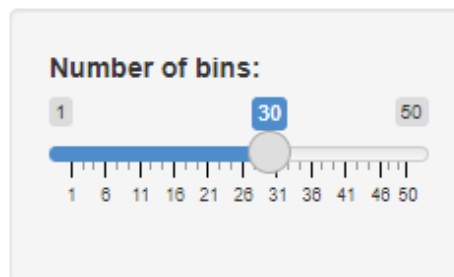
```
library(shiny)

# Define UI for application
ui <- fluidPage(
  ...
)

# Define server logic
server <- function(input, output) {
  ...
}

# Run the application
shinyApp(ui = ui, server = server)
```

Old Faithful Geyser Data



Adjusting to multi-file format

app.R

```
library(shiny)

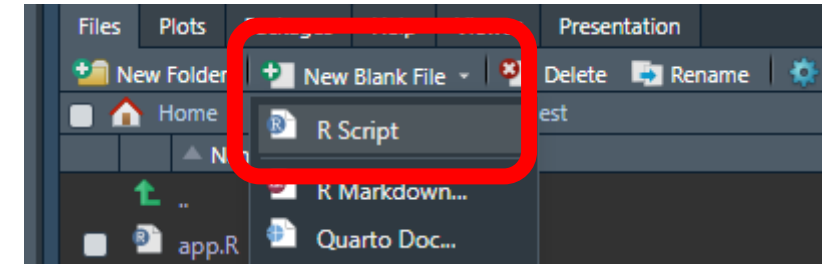
# Define UI for application that draws a histogram
ui <- fluidPage(
  ...
)
```

```
# Define server logic required to draw a histogram
server <- function(input, output) {
  ...
}
```

```
# Run the application
shinyApp(ui = ui, server = server)
```

ui.R

server.R



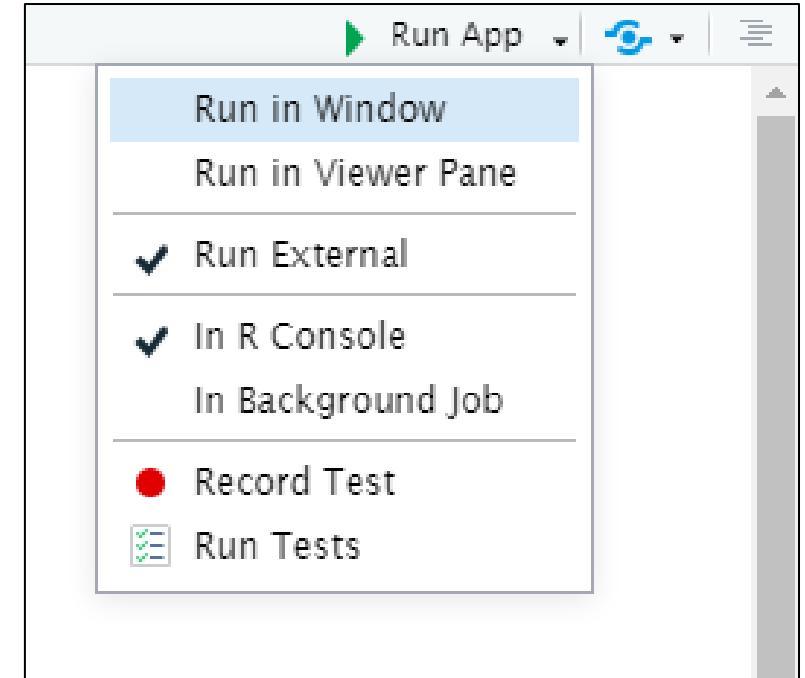
Our first app

Desired code structure:

- Interface: `ui.R`
 - Must return a single UI object
 - Pre-defined interface components (widgets), e.g. `sliderInput`
 - Translated into HTML and JS
- Server side: `server.R`
 - Must return the server function
 - R functions
 - Always run as R code
- Let's change our app over to this now

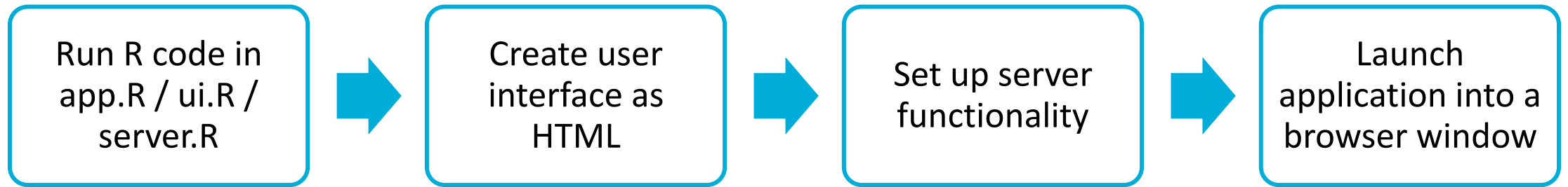
Run in browser

- By default, RStudio runs Shiny apps in an RStudio window
- Select 'Run External' to make your Shiny application run in your default browser



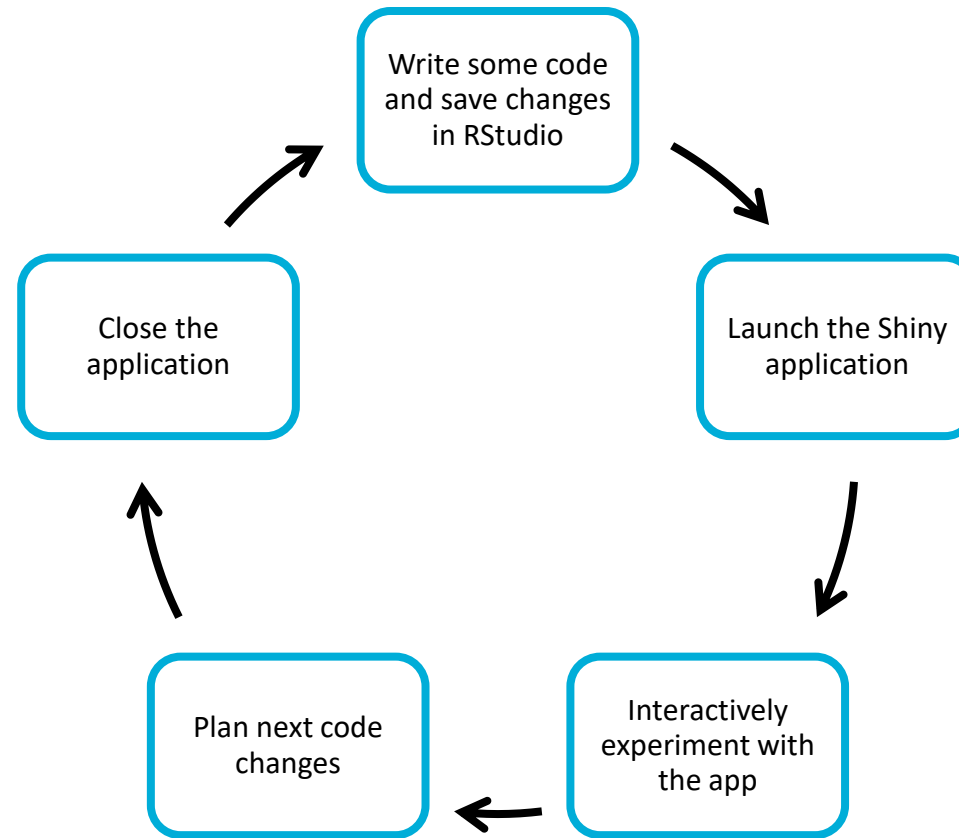
Running a Shiny App

- When running a Shiny app, things happen in a set order:



- While running, the R session is 'busy'
- Variables made in app are 'short-lived'

Basic Shiny Application workflow



Exercise

In ui.R:

- Change title of the page
- Change the title and width of the slider input
- Move the sidebar to the right of the page

In server.R

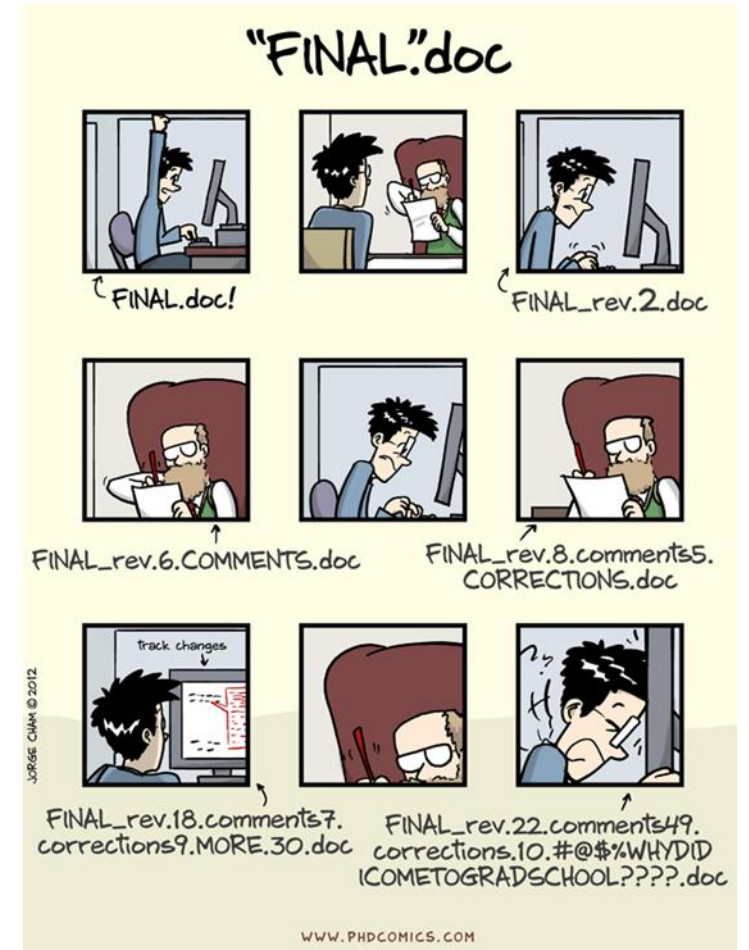
- Change the bar colour, the title, and axes labels of the histogram

Questions to answer:

- In sliderInput, how do we create a double-ended range slider? Does this work with our current plot?
- What happens if we remove mainPanel from the code, why does this happen?

Version Control

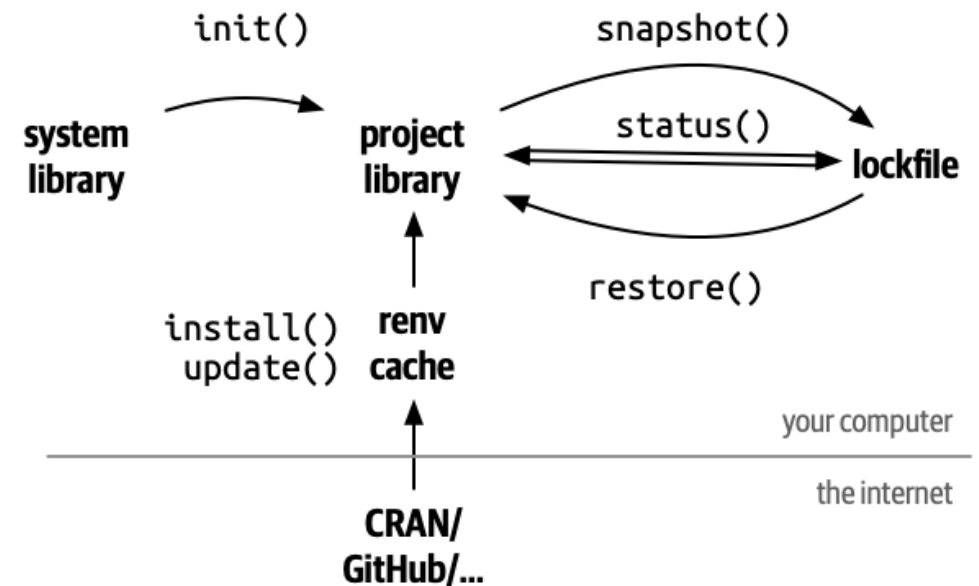
- System for maintaining versions when developing software
- Store code in remote repositories
- Collaborative working
- Meaningfully named updates for later reference
- Roll back changes if required
- Especially useful for software:
 - Lots of moving parts / different files that are edited
 - Long term projects with multiple collaborators at once



phdcomics.com

The renv package

- Used for recording and reproducing the projects environment
- Loads libraries and repositories with versions
- **Setup:** `renv::init()`
- **Save:** `renv::snapshot()`
- **Load:** `renv::restore()`
- <https://rstudio.github.io/renv/articles/renv.html>



Next time

- Shiny core concepts
- Bootstrap and mobile-ready page layouts

Challenge:

- Create a new repository on [GitHub](#)
- Create a new project in RStudio Cloud from this git repository
- Copy over the files from your first shiny app to this new project
- Set up renv in the project
- Make some changes to your app and push those changes to the git repository
- Share the link to your repository on the **Session 1 forum**