SHINY WEB APPS

Heike Hofmann

THE

- Create conceptual understanding of shiny apps
- LOTS of practice
- Come out of the course with a shiny app of your own

SETUP

- Materials (pdf, code) are located in GitHub:
  https://github.com/DSPG-ISU/Training
  Clone the directory onto your local machine

- Open an RStudio session
  (on Rivanna, locally or RStudio cloud: https://rstudio.cloud/project/1424983)

OUTLINE

- **Day 1**
  - Getting ready
  - Inputs & Outputs
  - Elements of Reactivity
  - Layouts
  - Some practice
- **Day 2**
  - leaflets in shiny
  - Work on shiny apps for the DSPG projects

CHECKING VERSIONS OF PACKAGES

- Make sure you have package devtools installed:

```
library(devtools)
```

- If the command fails, install the package

```
install.packages("devtools")
```

SESSION INFO

```
library(devtools)
session_info()
```

Whenever you have a strange problem, make a copy of your session info

```
> session_info()
─ Session info ─────────────────────────────────────────────────────────────
 setting  value
 version  R version 4.0.1 (2020-06-06)
 os       macOS Catalina 10.15.5
 system   x86_64, darwin17.0
 ui       RStudio
 language (EN)
 collate  en_US.UTF-8
 ctype    en_US.UTF-8
 tz       America/Chicago
 date     2020-06-27

─ Packages ─────────────────────────────────────────────────────────────────
 package     * version date       lib source
 assertthat    0.2.1   2019-03-21 [2] CRAN (R 4.0.0)
 backports     1.1.7   2020-05-13 [2] CRAN (R 4.0.0)
 callr         3.4.3   2020-03-28 [2] CRAN (R 4.0.0)
 cli           2.0.2   2020-02-28 [2] CRAN (R 4.0.0)
 crayon        1.3.4   2017-09-16 [2] CRAN (R 4.0.0)
 desc          1.2.0   2018-05-01 [2] CRAN (R 4.0.0)
 devtools    * 2.3.0   2020-04-10 [1] CRAN (R 4.0.0)
 digest        0.6.25  2020-02-23 [2] CRAN (R 4.0.0)
 ellipsis      0.3.1   2020-05-15 [2] CRAN (R 4.0.0)
 fansi         0.4.1   2020-01-08 [2] CRAN (R 4.0.0)
 fs            1.4.1   2020-04-04 [1] CRAN (R 4.0.0)
 glue          1.4.1   2020-05-13 [2] CRAN (R 4.0.0)
 magrittr      1.5     2014-11-22 [2] CRAN (R 4.0.0)
 memoise       1.1.0   2017-04-21 [2] CRAN (R 4.0.0)
 pkgbuild      1.0.8   2020-05-07 [2] CRAN (R 4.0.0)
 pkgload       1.1.0   2020-05-29 [2] CRAN (R 4.0.0)
 prettyunits   1.1.1   2020-01-24 [2] CRAN (R 4.0.0)
 processx      3.4.2   2020-02-09 [2] CRAN (R 4.0.0)
 ps            1.3.3   2020-05-08 [2] CRAN (R 4.0.0)
 R6            2.4.1   2019-11-12 [2] CRAN (R 4.0.0)
 remotes       2.1.1   2020-02-15 [1] CRAN (R 4.0.0)
 rlang         0.4.6   2020-05-02 [2] CRAN (R 4.0.0)
 rprojroot     1.3-2   2018-01-03 [2] CRAN (R 4.0.0)
 rstudioapi    0.11    2020-02-07 [2] CRAN (R 4.0.0)
 sessioninfo   1.1.1   2018-11-05 [1] CRAN (R 4.0.0)
 testthat      2.3.2   2020-03-02 [2] CRAN (R 4.0.0)
 usethis     * 1.6.1   2020-04-29 [1] CRAN (R 4.0.0)
 withr         2.2.0   2020-04-20 [2] CRAN (R 4.0.0)

[1] /Users/hofmann/Library/R/4.0/library
[2] /Library/Frameworks/R.framework/Versions/4.0/Resources/library
```

IS SHINY INSTALLED?

- Run the following command in your RStudio Console:
  `library(shiny)`

- Check that you have a new(ish) version of shiny
  1.5.0 is the newest version of shiny

Your Turn: check the version of your shiny package

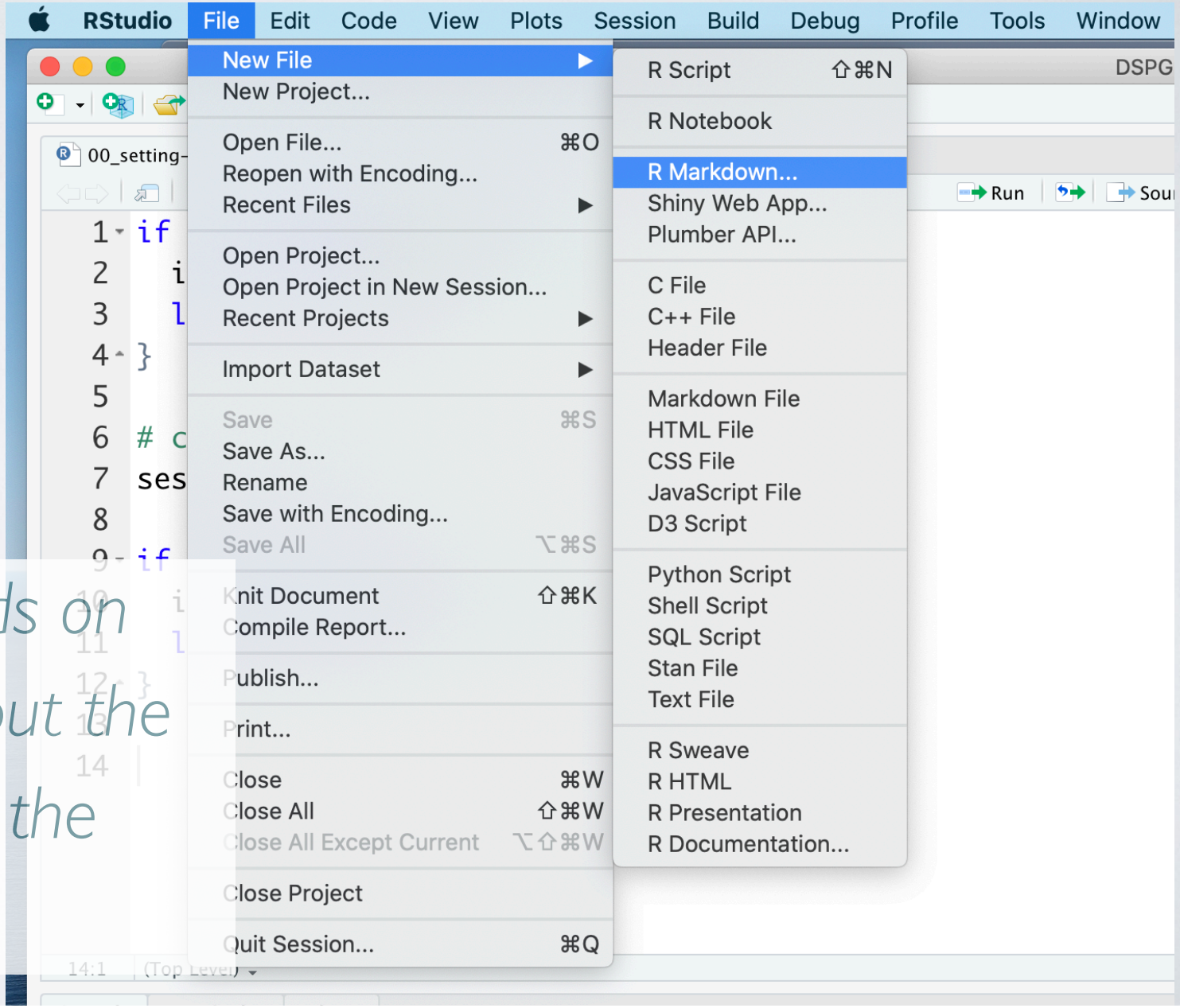
- If either one of the above commands fails/or the answer is no:
  `install.packages("shiny")`

A FIRST APP

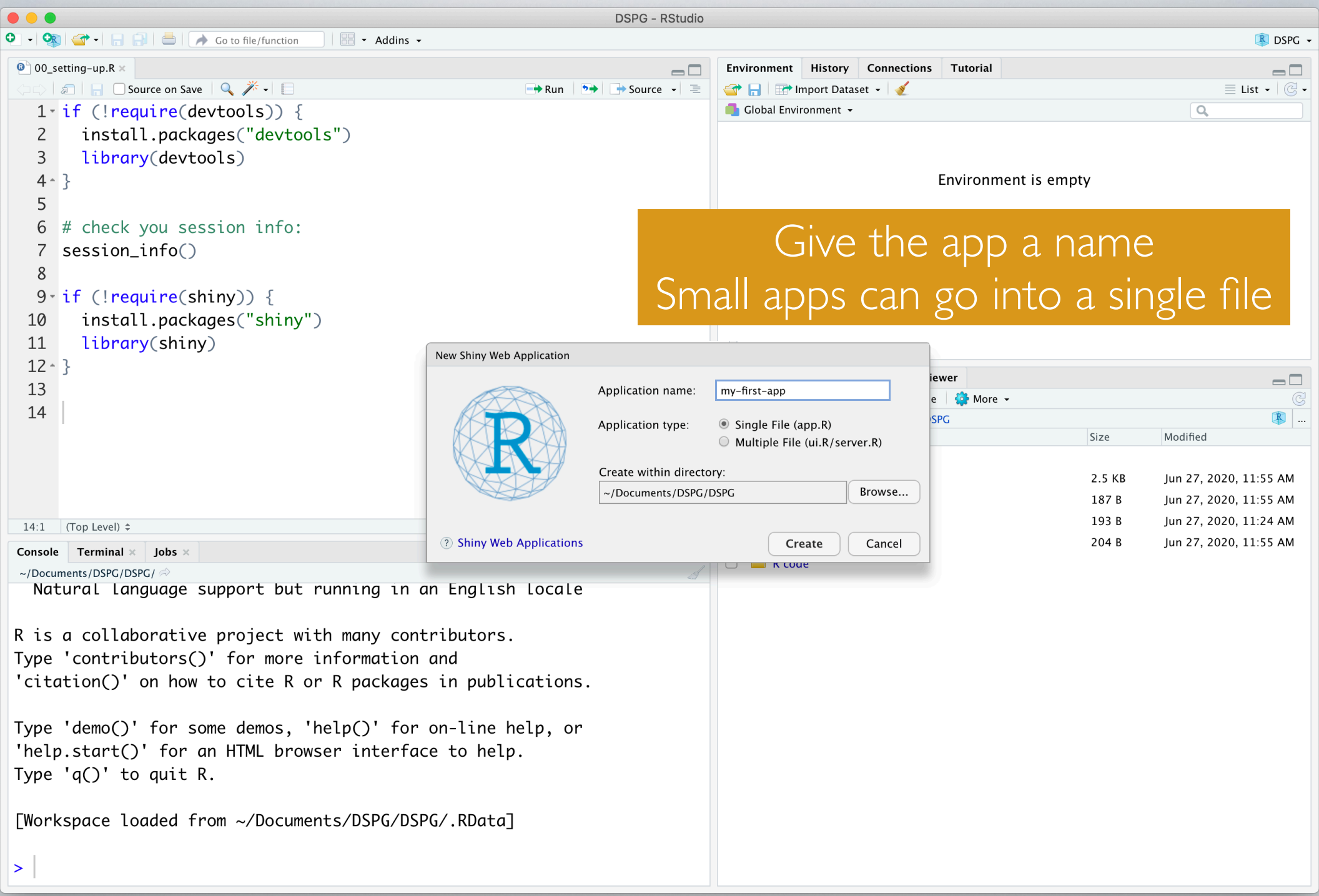
NEXT 5 MINS

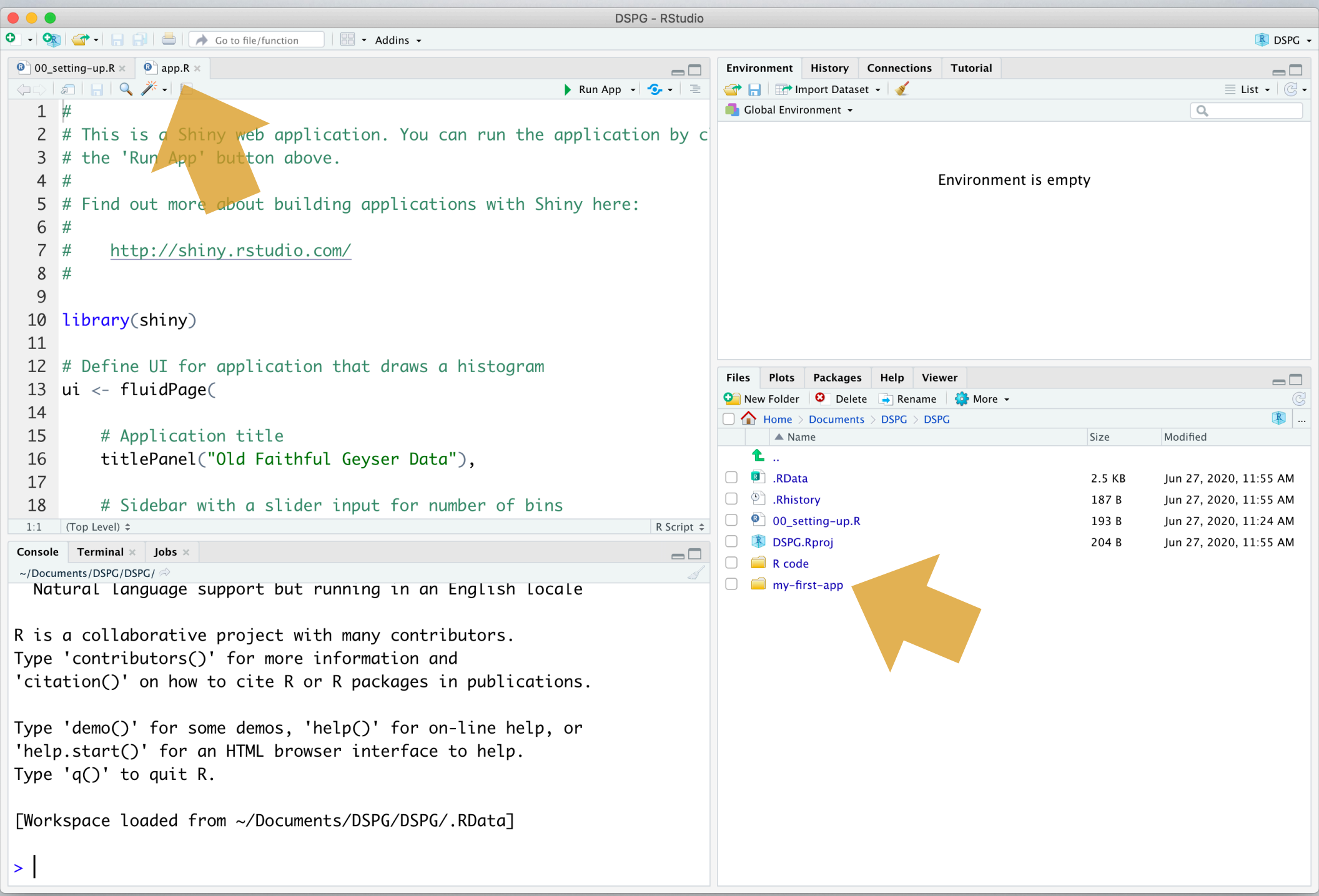
- Create a new shiny app in RStudio
- Run it
- Get out of it

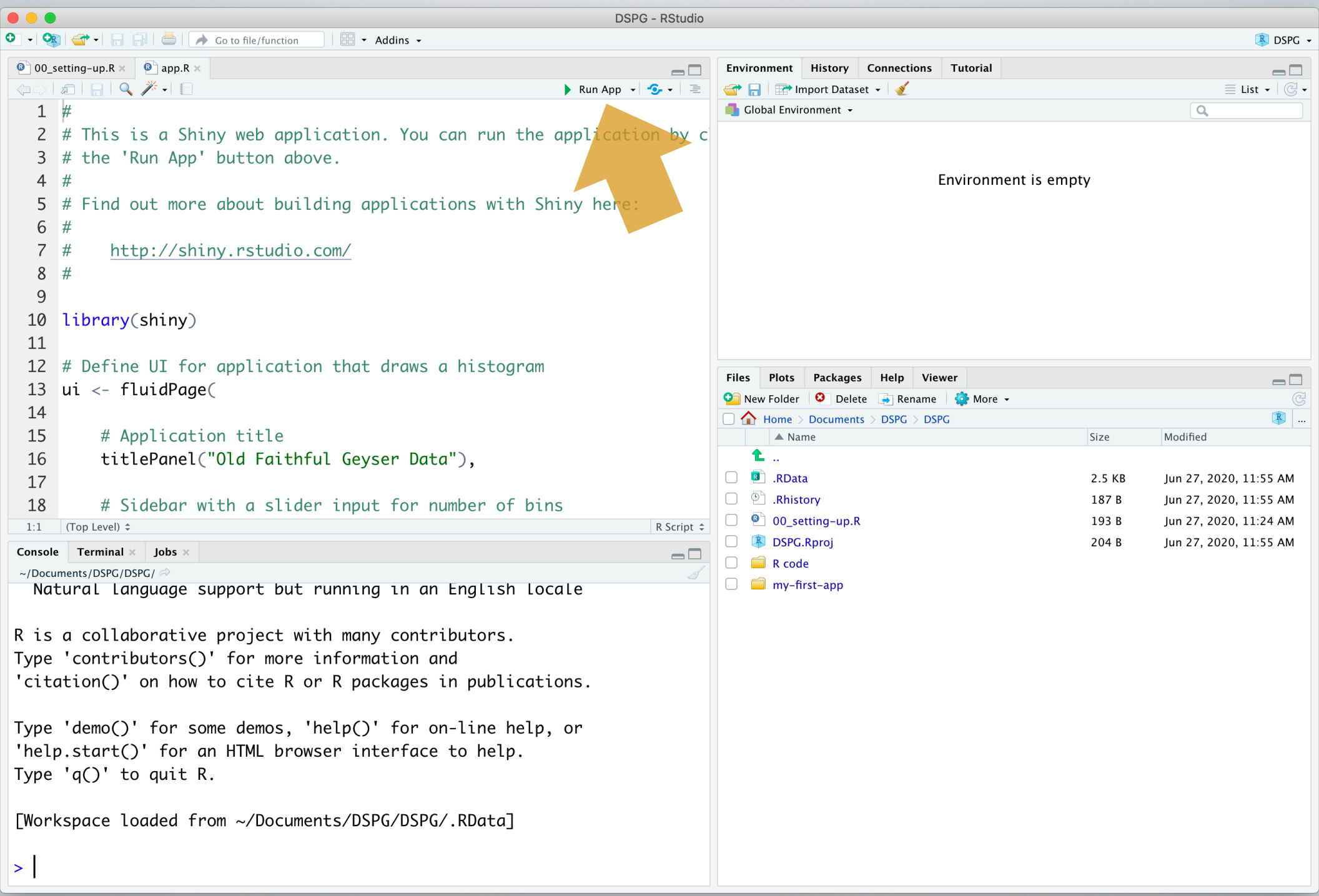
CREATE A NEW SHINY APP



The UI depends on the platform, but the functionality is the same


DSPG - RStudio
Go to file/function
Addins
DSPG
00_setting-up.R
Source on Save
Run
Source
1 if (!require(devtools)) {
2 install.packages("devtools")
3 library(devtools)
4 }
5
6 # check you session info:
7 session_info()
8
9 if (!require(shiny)) {
10 install.packages("shiny")
11 library(shiny)
12 }
13
14
14:1
(Top Level)
Console
Terminal
Jobs
~/Documents/DSPG/DSPG/
Natural language support but running in an English locale
R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.
Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.
[Workspace loaded from ~/Documents/DSPG/DSPG/.RData]
>
Environment
History
Connections
Tutorial
Import Dataset
List
Global Environment
Environment is empty
Give the app a name
Small apps can go into a single file
New Shiny Web Application
Application name:
my-first-app
Application type:
Single File (app.R)
Multiple File (ui.R/server.R)
Create within directory:
~/Documents/DSPG/DSPG
Browse...
Shiny Web Applications
Create
Cancel
More
Size
Modified
2.5 KB
Jun 27, 2020, 11:55 AM
187 B
Jun 27, 2020, 11:55 AM
193 B
Jun 27, 2020, 11:24 AM
204 B
Jun 27, 2020, 11:55 AM


DSPG - RStudio
Go to file/function
Addins
DSPG
00_setting-up.R
app.R
Run App
1 #
2 # This is a Shiny web application. You can run the application by c
3 # the 'Run App' button above.
4 #
5 # Find out more about building applications with Shiny here:
6 #
7 # http://shiny.rstudio.com/
8 #
9
10 library(shiny)
11
12 # Define UI for application that draws a histogram
13 ui <- fluidPage(
14
15 # Application title
16 titlePanel("Old Faithful Geyser Data"),
17
18 # Sidebar with a slider input for number of bins
1:1
(Top Level)
R Script
Console
Terminal
Jobs
~/Documents/DSPG/DSPG/
Natural language support but running in an English locale
R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.
Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.
[Workspace loaded from ~/Documents/DSPG/DSPG/.RData]
>
Environment
History
Connections
Tutorial
Import Dataset
List
Global Environment
Environment is empty
Files
Plots
Packages
Help
Viewer
New Folder
Delete
Rename
More
Home
Documents
DSPG
DSPG
Name
Size
Modified
..
.RData
2.5 KB
Jun 27, 2020, 11:55 AM
.Rhistory
187 B
Jun 27, 2020, 11:55 AM
00_setting-up.R
193 B
Jun 27, 2020, 11:24 AM
DSPG.Rproj
204 B
Jun 27, 2020, 11:55 AM
R code
my-first-app


DSPG - RStudio
Go to file/function
Addins
DSPG
00_setting-up.R
app.R
Run App
1 #
2 # This is a Shiny web application. You can run the application by c
3 # the 'Run App' button above.
4 #
5 # Find out more about building applications with Shiny here:
6 #
7 # http://shiny.rstudio.com/
8 #
9
10 library(shiny)
11
12 # Define UI for application that draws a histogram
13 ui <- fluidPage(
14
15 # Application title
16 titlePanel("Old Faithful Geyser Data"),
17
18 # Sidebar with a slider input for number of bins
1:1
(Top Level)
R Script
Console
Terminal
Jobs
~/Documents/DSPG/DSPG/
Natural language support but running in an English locale
R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.
Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.
[Workspace loaded from ~/Documents/DSPG/DSPG/.RData]
>
Environment
History
Connections
Tutorial
Import Dataset
List
Global Environment
Environment is empty
Files
Plots
Packages
Help
Viewer
New Folder
Delete
Rename
More
Home
Documents
DSPG
DSPG
Name
Size
Modified
..
.RData
2.5 KB
Jun 27, 2020, 11:55 AM
.Rhistory
187 B
Jun 27, 2020, 11:55 AM
00_setting-up.R
193 B
Jun 27, 2020, 11:24 AM
DSPG.Rproj
204 B
Jun 27, 2020, 11:55 AM
R code
my-first-app

~/Documents/DSPG/DSPG/my-first-app - Shiny

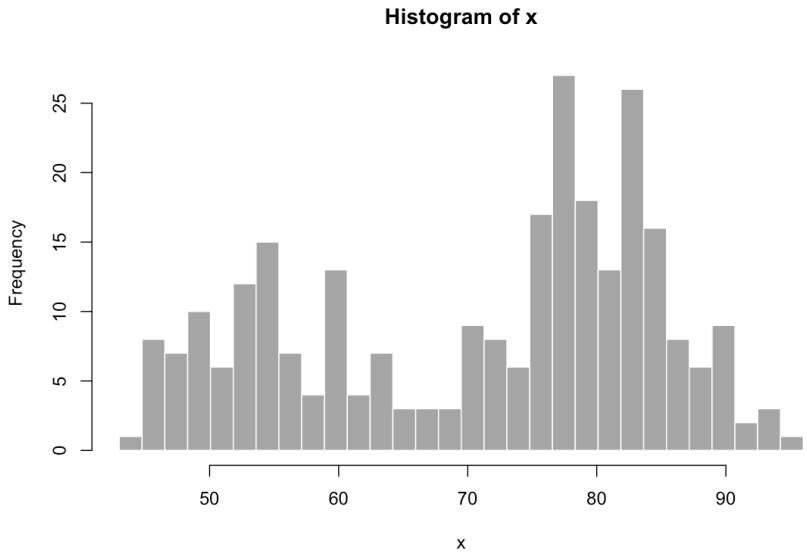
http://127.0.0.1:4127 | Open in Browser | Publish

Old Faithful Geyser Data

Number of bins:

1 30 50

1 6 11 16 21 26 31 36 41 46 50



This is how things should look like

00_setting-up.R | app.R

```
1  #
2  # This is a Shi
3  # the 'Run App'
4  #
5  # Find out more
6  #
7  #    http://shi
8  #
9
10 library(shiny)
11
12 # Define UI for
13 ui <- fluidPage
14
15     # Applicati
16     titlePanel(
17
18     # Sidebar w
```

1:1 (Top Level)

Console | Terminal | Jobs

~/Documents/DSPG/DSPG/

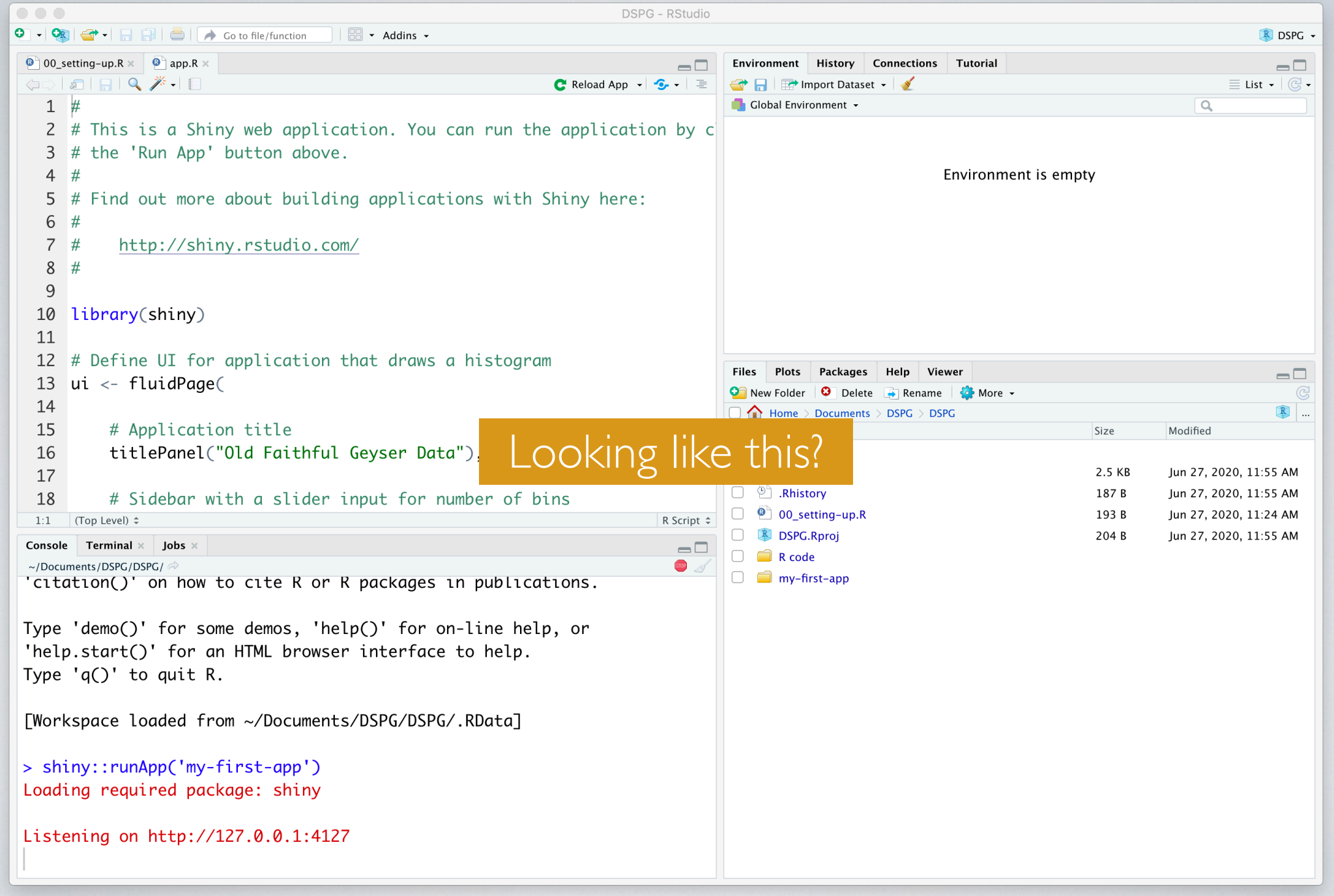
```
'citation()' on how
Type 'demo()' for so
'help.start()' for a
Type 'q()' to quit R
[Workspace loaded fr
> shiny::runApp('my-
Loading required pac
Listening on http://
```

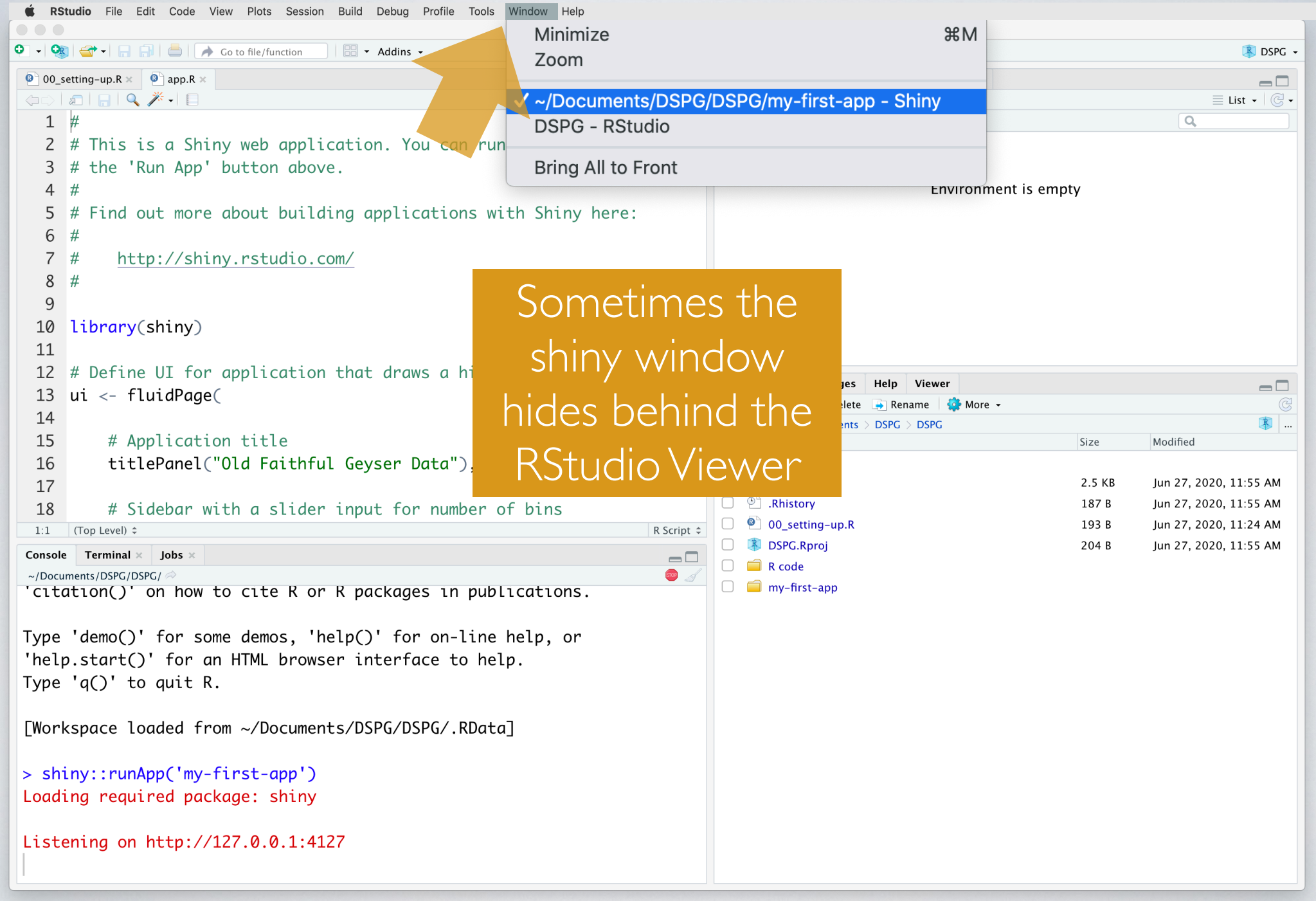
DSPG

List

Modified

KB Jun 27, 2020, 11:55 AM
B Jun 27, 2020, 11:55 AM
B Jun 27, 2020, 11:24 AM
B Jun 27, 2020, 11:55 AM


DSPG - RStudio
Go to file/function
Addins
DSPG
00_setting-up.R
app.R
Reload App
1 #
2 # This is a Shiny web application. You can run the application by c
3 # the 'Run App' button above.
4 #
5 # Find out more about building applications with Shiny here:
6 #
7 # http://shiny.rstudio.com/
8 #
9
10 library(shiny)
11
12 # Define UI for application that draws a histogram
13 ui <- fluidPage(
14
15 # Application title
16 titlePanel("Old Faithful Geyser Data");
17
18 # Sidebar with a slider input for number of bins
1:1 (Top Level)
R Script
Looking like this?
Environment
History
Connections
Tutorial
Import Dataset
List
Global Environment
Environment is empty
Files
Plots
Packages
Help
Viewer
New Folder
Delete
Rename
More
Home
Documents
DSPG
DSPG
Size
Modified
2.5 KB
Jun 27, 2020, 11:55 AM
.Rhistory
187 B
Jun 27, 2020, 11:55 AM
00_setting-up.R
193 B
Jun 27, 2020, 11:24 AM
DSPG.Rproj
204 B
Jun 27, 2020, 11:55 AM
R code
my-first-app
Console
Terminal
Jobs
~/Documents/DSPG/DSPG/
'citation()' on how to cite R or R packages in publications.
Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.
[Workspace loaded from ~/Documents/DSPG/DSPG/.RData]
> shiny::runApp('my-first-app')
Loading required package: shiny
Listening on http://127.0.0.1:4127


RStudio File Edit Code View Plots Session Build Debug Profile Tools Window Help
Minimize ⌘M
Zoom
~/Documents/DSPG/DSPG/my-first-app - Shiny
DSPG - RStudio
Bring All to Front
Addins
DSPG
00_setting-up.R
app.R
List
1 #
2 # This is a Shiny web application. You can run
3 # the 'Run App' button above.
4 #
5 # Find out more about building applications with Shiny here:
6 #
7 # http://shiny.rstudio.com/
8 #
9
10 library(shiny)
11
12 # Define UI for application that draws a hi
13 ui <- fluidPage(
14
15 # Application title
16 titlePanel("Old Faithful Geyser Data"),
17
18 # Sidebar with a slider input for number of bins
1:1 (Top Level) R Script
Environment is empty
Help Viewer
Rename More
DSPG DSPG
Size Modified
2.5 KB Jun 27, 2020, 11:55 AM
.Rhistory 187 B Jun 27, 2020, 11:55 AM
00_setting-up.R 193 B Jun 27, 2020, 11:24 AM
DSPG.Rproj 204 B Jun 27, 2020, 11:55 AM
R code
my-first-app
Console Terminal Jobs
~/Documents/DSPG/DSPG/
'citation()' on how to cite R or R packages in publications.
Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.
[Workspace loaded from ~/Documents/DSPG/DSPG/.RData]
> shiny::runApp('my-first-app')
Loading required package: shiny
Listening on http://127.0.0.1:4127
Sometimes the shiny window hides behind the RStudio Viewer

~/Documents/DSPG/DSPG/my-first-app - Shiny

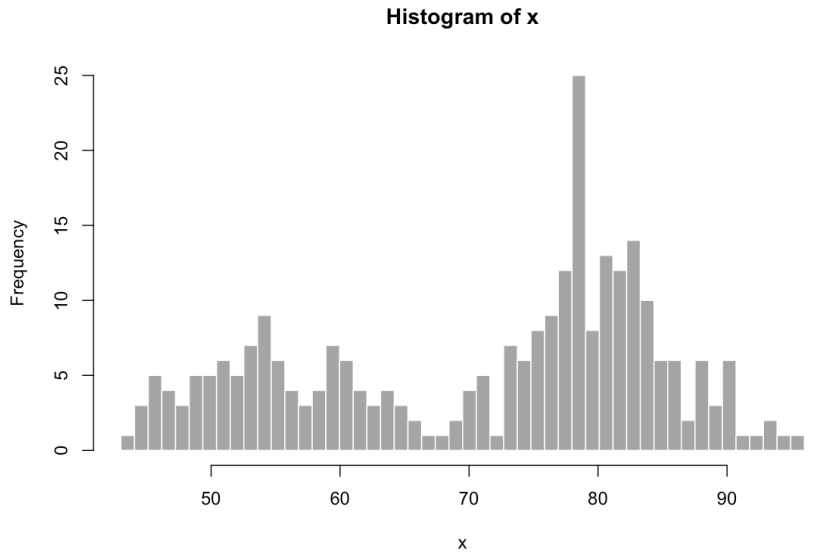
http://127.0.0.1:4127 | Open in Browser | Publish

Old Faithful Geyser Data

Number of bins:

1 50

1 6 11 16 21 26 31 36 41 46 50



00_setting-up.R | app.R

```
1  #
2  # This is a Shiny web application.
3  # the 'Run App' button above.
4  #
5  # Find out more about building app
6  #
7  #    http://shiny.rstudio.com/
8  #
9
10 library(shiny)
11
12 # Define UI for application that d
13 ui <- fluidPage(
14
15    # Application title
16    titlePanel("Old Faithful Geyse
17
18    # Sidebar with a slider input
```

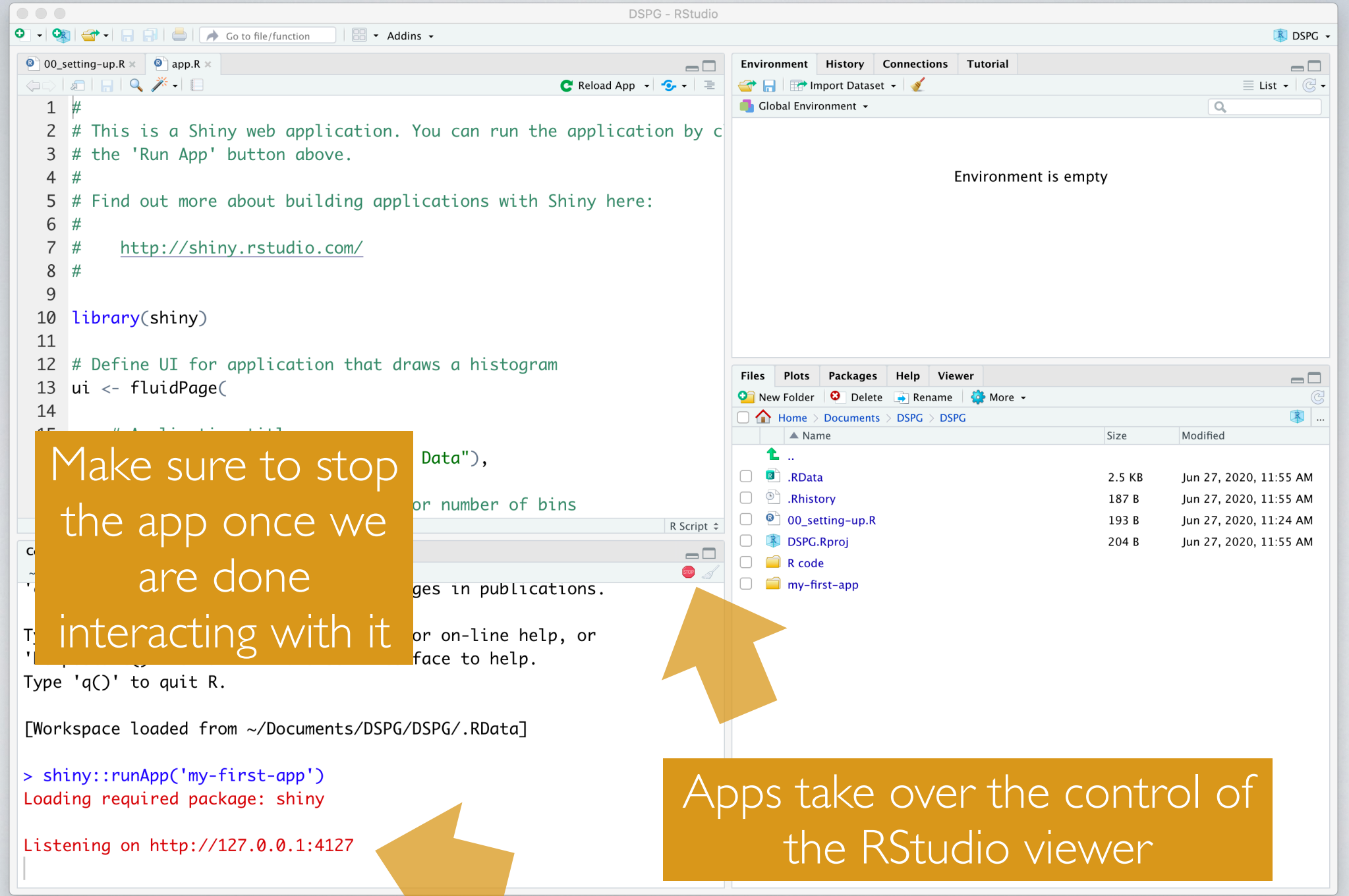
1:1 (Top Level)

Console | Terminal | Jobs

~/Documents/DSPG/DSPG/

```
'citation()' on how to cite R or R pack

Type 'demo()' for some demos, 'help()'
'help.start()' for an HTML browser inte
Type 'q()' to quit R.

Listening on http://127.0.0.1:4127
```

Now interact with the app


DSPG - RStudio
Go to file/function
Addins
DSPG
00_setting-up.R
app.R
Reload App
1 #
2 # This is a Shiny web application. You can run the application by c
3 # the 'Run App' button above.
4 #
5 # Find out more about building applications with Shiny here:
6 #
7 # http://shiny.rstudio.com/
8 #
9
10 library(shiny)
11
12 # Define UI for application that draws a histogram
13 ui <- fluidPage(
14
Data"),
or number of bins
R Script
ges in publications.
or on-line help, or
face to help.
Type 'q()' to quit R.
[Workspace loaded from ~/Documents/DSPG/DSPG/.RData]
> shiny::runApp('my-first-app')
Loading required package: shiny
Listening on http://127.0.0.1:4127
Environment
History
Connections
Tutorial
Import Dataset
List
Global Environment
Environment is empty
Files
Plots
Packages
Help
Viewer
New Folder
Delete
Rename
More
Home
Documents
DSPG
DSPG
Name
Size
Modified
..
.RData
2.5 KB
Jun 27, 2020, 11:55 AM
.Rhistory
187 B
Jun 27, 2020, 11:55 AM
00_setting-up.R
193 B
Jun 27, 2020, 11:24 AM
DSPG.Rproj
204 B
Jun 27, 2020, 11:55 AM
R code
my-first-app
Make sure to stop the app once we are done interacting with it
Apps take over the control of the RStudio viewer

HOW DOES SHINY WORK?

TWO MAIN PARTS

- What we see and interact with:
  - The user interface (UI):
    layout with user input and (plot) output
- What is going on underneath:
  - The server: the glue between input and output

MINIMAL EXAMPLE

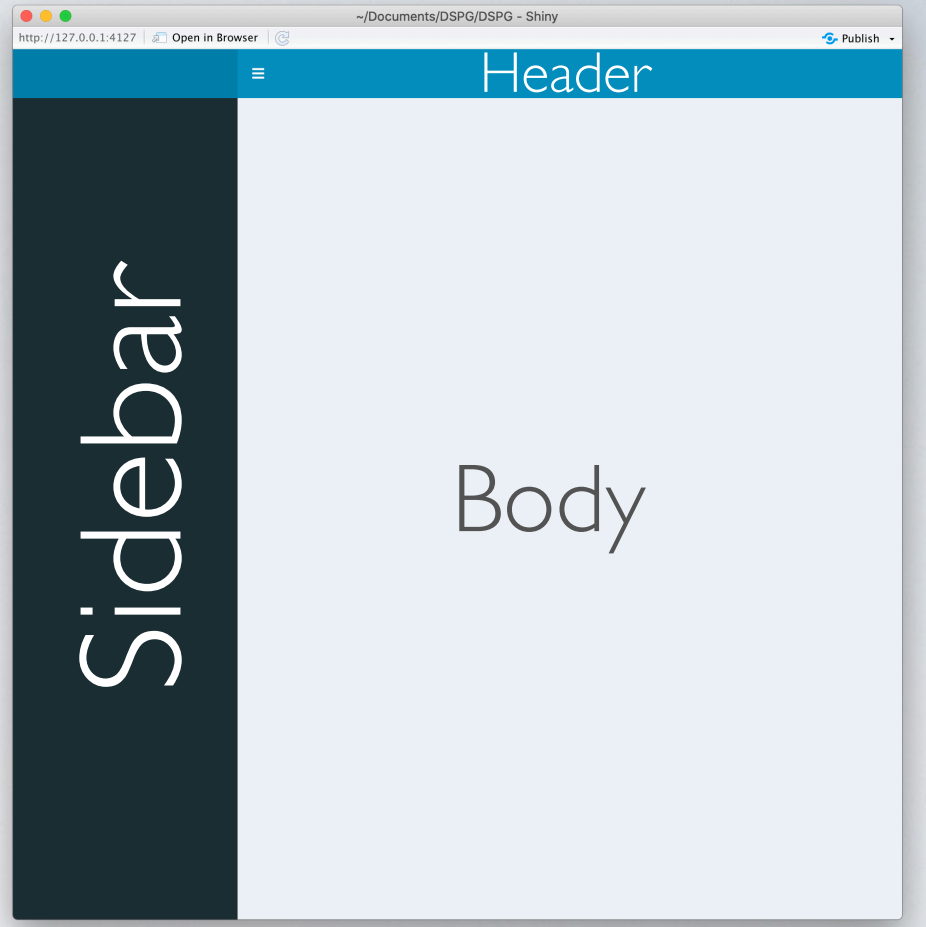
```
library(shiny)

ui <- fluidPage(

)

server <- function(input, output, session) {

}

shinyApp(ui, server)
```

The user interface

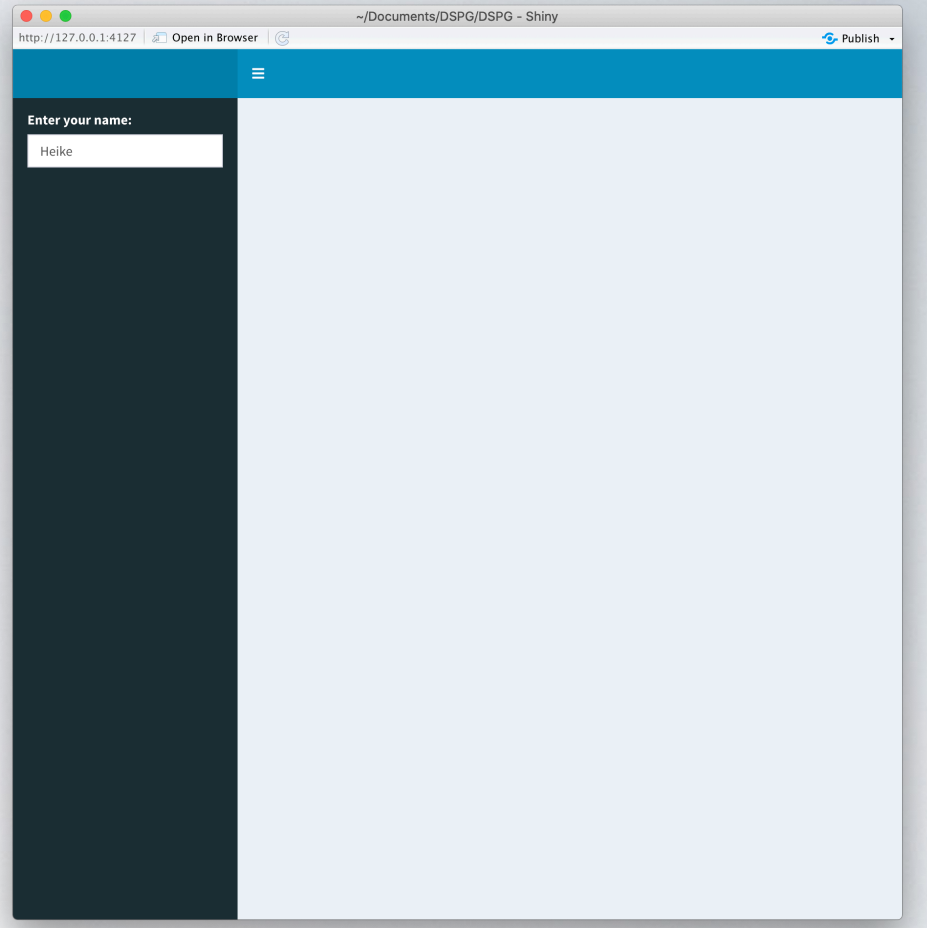
The glue between input and output

Running the app

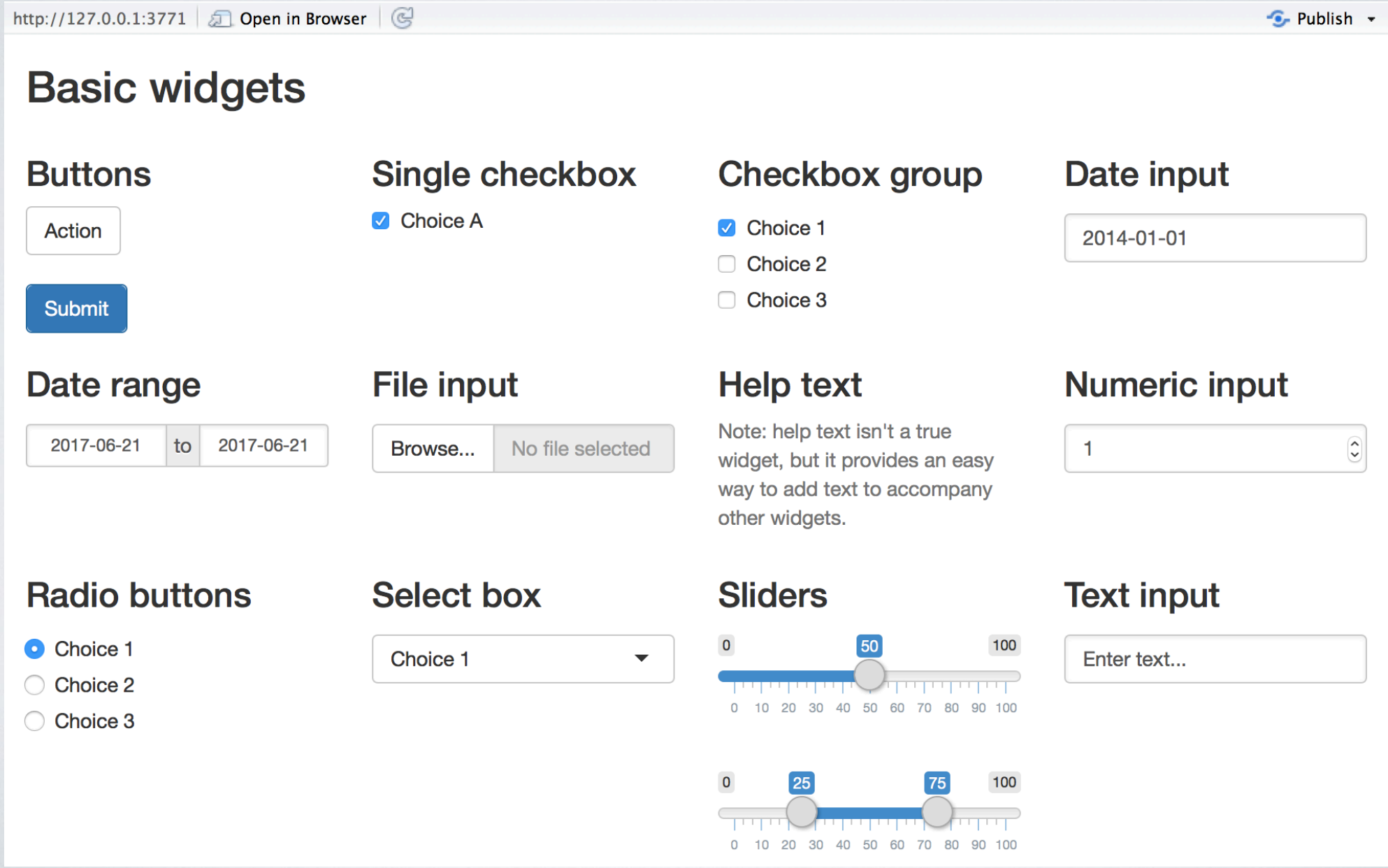
```
library(shiny)
library(shinydashboard)

ui <- dashboardPage(
  dashboardHeader(),
  dashboardSidebar(),
  dashboardBody()
)

server <- function(input, output, session) {

}

shinyApp(ui, server)
```



```
library(shiny)
library(shinydashboard)

sidebar <- dashboardSidebar(
  textInput("name",
            "Enter your name:",
            value = "Heike")
)

ui <- dashboardPage(
  dashboardHeader(),
  sidebar = sidebar,
  dashboardBody()
)

server <- function(input, output, session) {

}

shinyApp(ui, server)

server <- function(input, output, session) {
```



INPUT WIDGETS



INPUT WIDGETS

- Each input widget corresponds to a function of the form

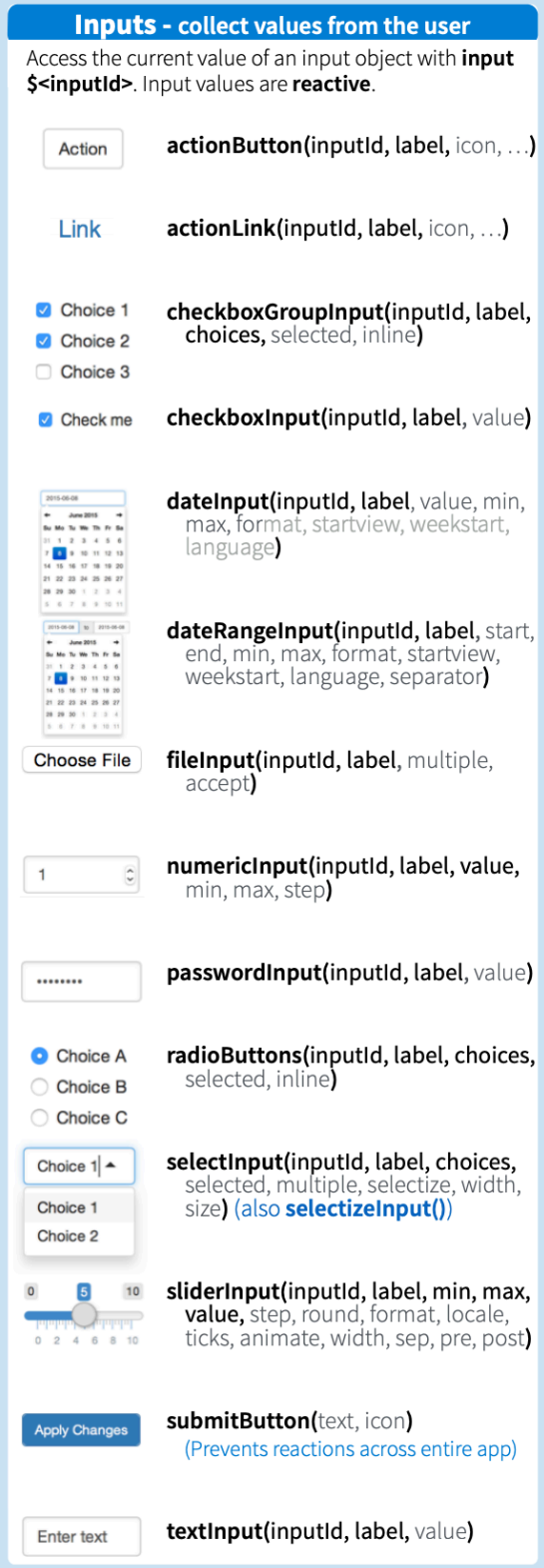
  `xxxInput(inputid, label, value, …)`

- `inputid`: internal name of the widget

  `label`: text shown to user

  `value`: default value at the start

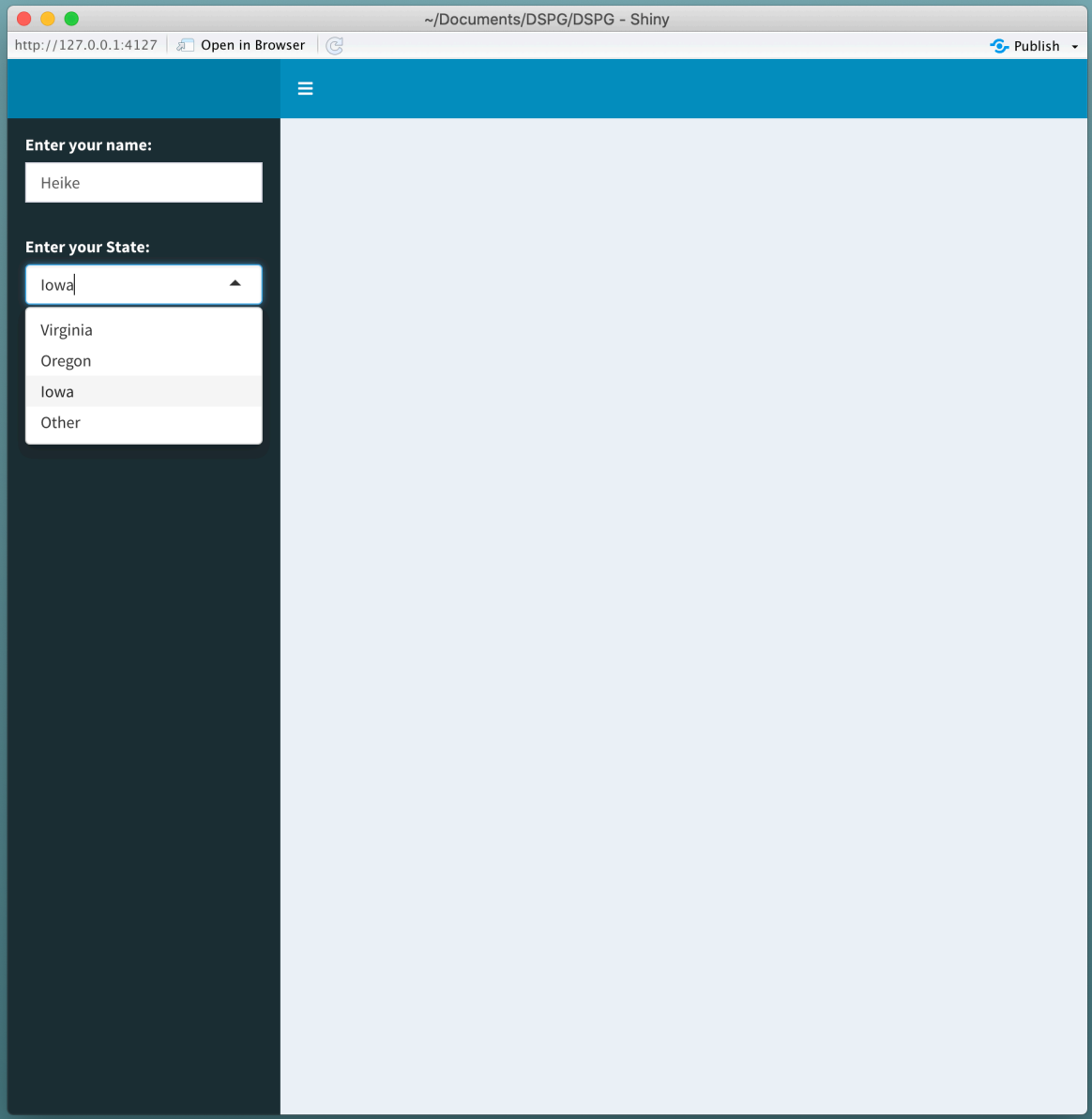
  …: other widget specific parameters



https://shiny.rstudio.com/images/shiny-cheatsheet.pdf

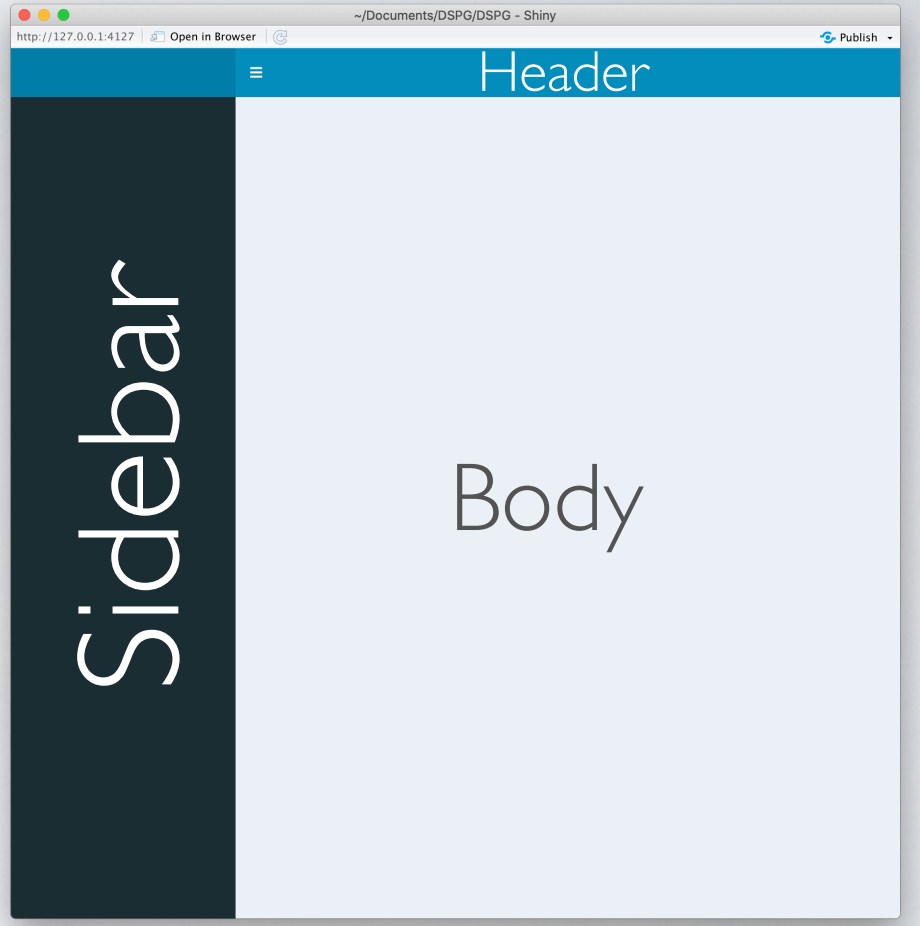
YOUR TURN

- Open the file app.R in folder 02b_minimal
- Modify the function sidebar to include an input as shown on the right
- Advanced: give the app a title
- The answer is in 02c_minimal



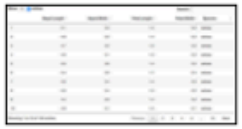
ADDING OUTPUT

- Output typically goes into the body of an app
- Output functions have the form **`xxxOutput(outputid)`**
- Output can be in form of tables, plots, text, …



RENDERING OUTPUT

Outputs - render*() and *Output() functions work together to add R output to the UI



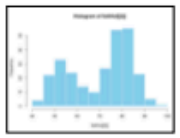
DT::**renderDataTable**(expr, options, callback, escape, env, quoted)

works with

dataTableOutput(outputId, icon, …)

renderImage(expr, env, quoted, deleteFile)

imageOutput(outputId, width, height, click, dblclick, hover, hoverDelay, hoverDelayType, brush, clickId, hoverId, inline)



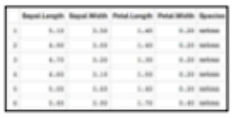
renderPlot(expr, width, height, res, …, env, quoted, func)

plotOutput(outputId, width, height, click, dblclick, hover, hoverDelay, hoverDelayType, brush, clickId, hoverId, inline)

renderPrint(expr, env, quoted, func, width)

verbatimTextOutput(outputId)

In server()



renderTable(expr, …, env, quoted, func)

tableOutput(outputId)

Usually in dashboardBody()

foo

renderText(expr, env, quoted, func)

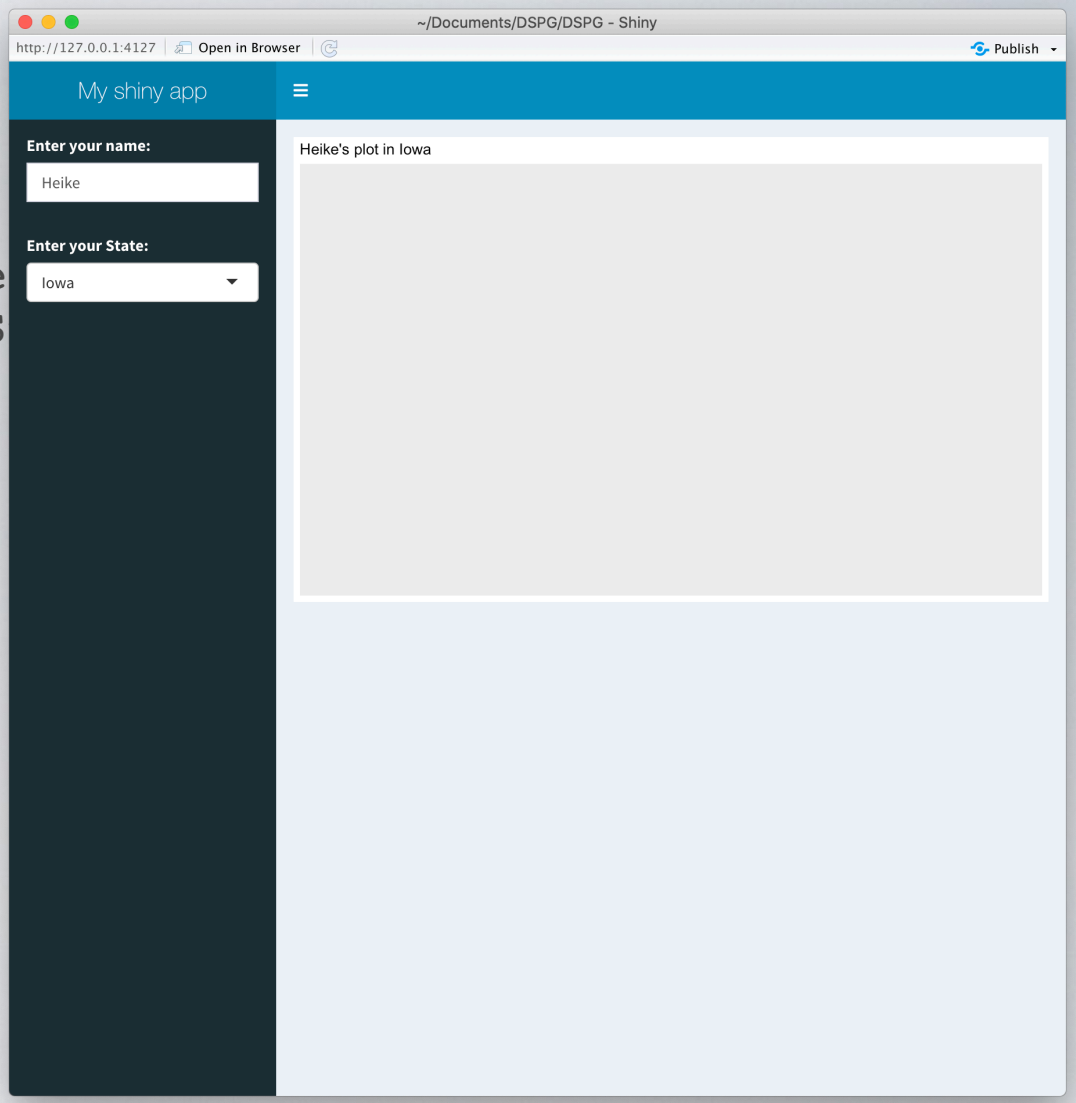
textOutput(outputId, container, inline)

renderUI(expr, env, quoted, func)

uiOutput(outputId, inline, container, …)

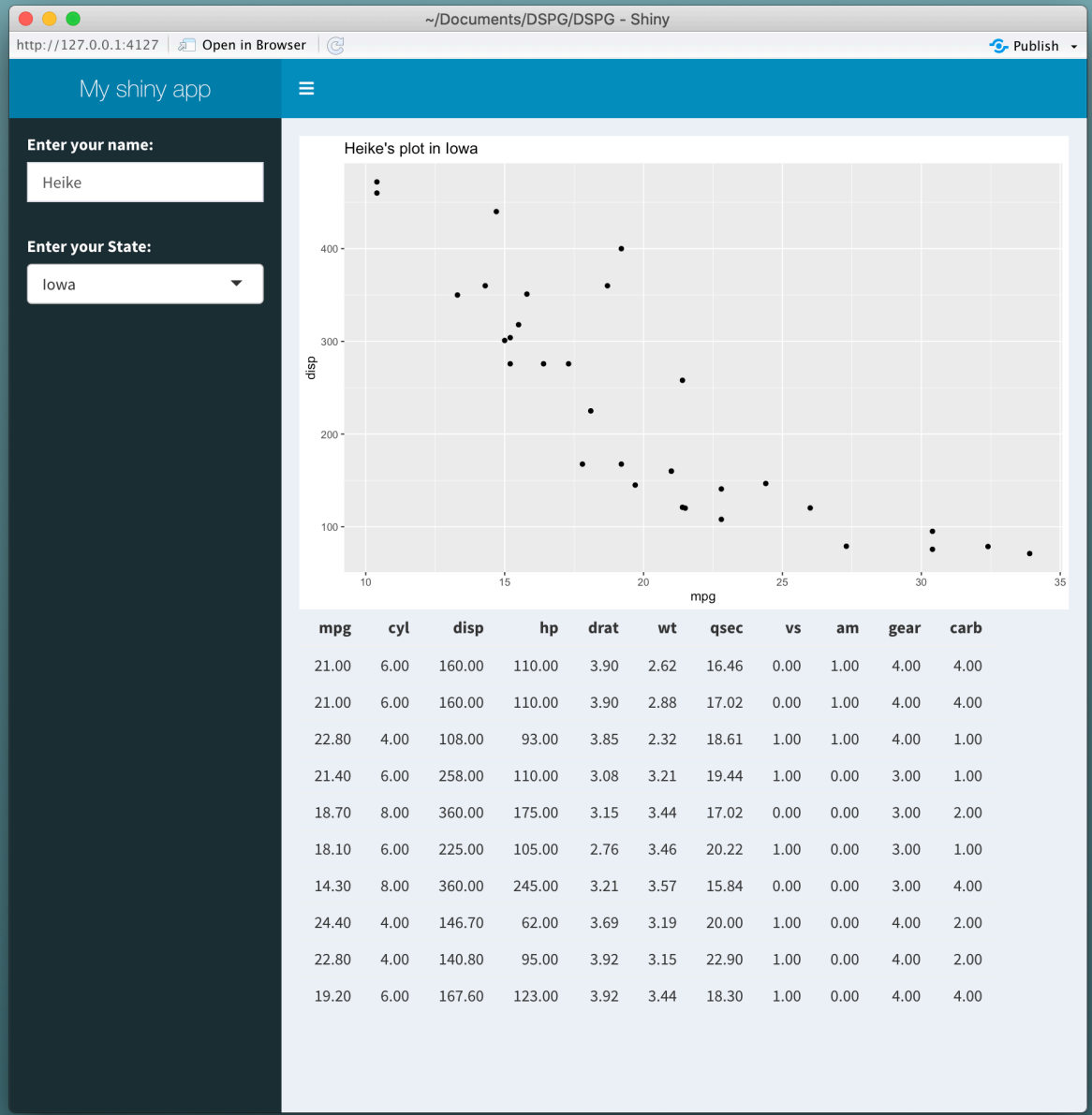
& **htmlOutput**(outputId, inline, container, …)

```
library(shiny)
library(shinydashboard)
library(ggplot2)

sidebar <- dashboardSidebar(
  textInput("name", "Enter your name
  selectInput("state", "Enter your S
              choices=c("Virginia",
              selected = "Iowa")
)

body <- dashboardBody(
  plotOutput("myplot")
)

ui <- dashboardPage(
  dashboardHeader(title = "My shiny
  sidebar = sidebar,
  body = body
)

server <- function(input, output, session) {
  output$myplot <- renderPlot({
    ggplot() +
      ggtitle(sprintf("%s's plot in %s", input$name, input$state))
  })
}

shinyApp(ui, server)
```



YOUR TURN

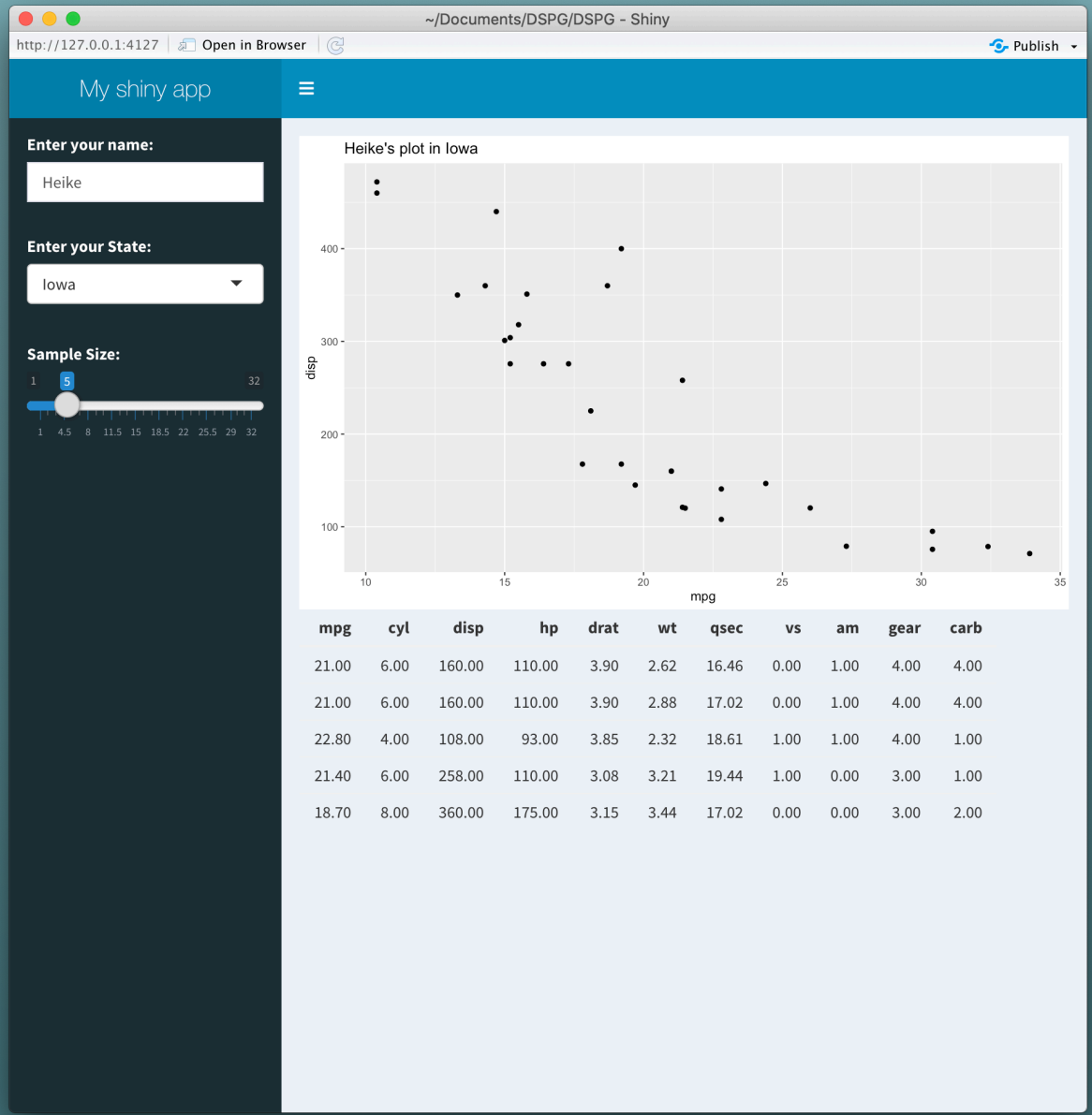
- Open the file app.R in folder 02d_minimal
- Modify the plot to show data points from the mtcars data
- Add a table below to show the first ten lines of the data
- The answer is in 02e_minimal



mpg	cyl	disp	hp	drat	wt	qsec	vs	am	gear	carb
21.00	6.00	160.00	110.00	3.90	2.62	16.46	0.00	1.00	4.00	4.00
21.00	6.00	160.00	110.00	3.90	2.88	17.02	0.00	1.00	4.00	4.00
22.80	4.00	108.00	93.00	3.85	2.32	18.61	1.00	1.00	4.00	1.00
21.40	6.00	258.00	110.00	3.08	3.21	19.44	1.00	0.00	3.00	1.00
18.70	8.00	360.00	175.00	3.15	3.44	17.02	0.00	0.00	3.00	2.00
18.10	6.00	225.00	105.00	2.76	3.46	20.22	1.00	0.00	3.00	1.00
14.30	8.00	360.00	245.00	3.21	3.57	15.84	0.00	0.00	3.00	4.00
24.40	4.00	146.70	62.00	3.69	3.19	20.00	1.00	0.00	4.00	2.00
22.80	4.00	140.80	95.00	3.92	3.15	22.90	1.00	0.00	4.00	2.00
19.20	6.00	167.60	123.00	3.92	3.44	18.30	1.00	0.00	4.00	4.00

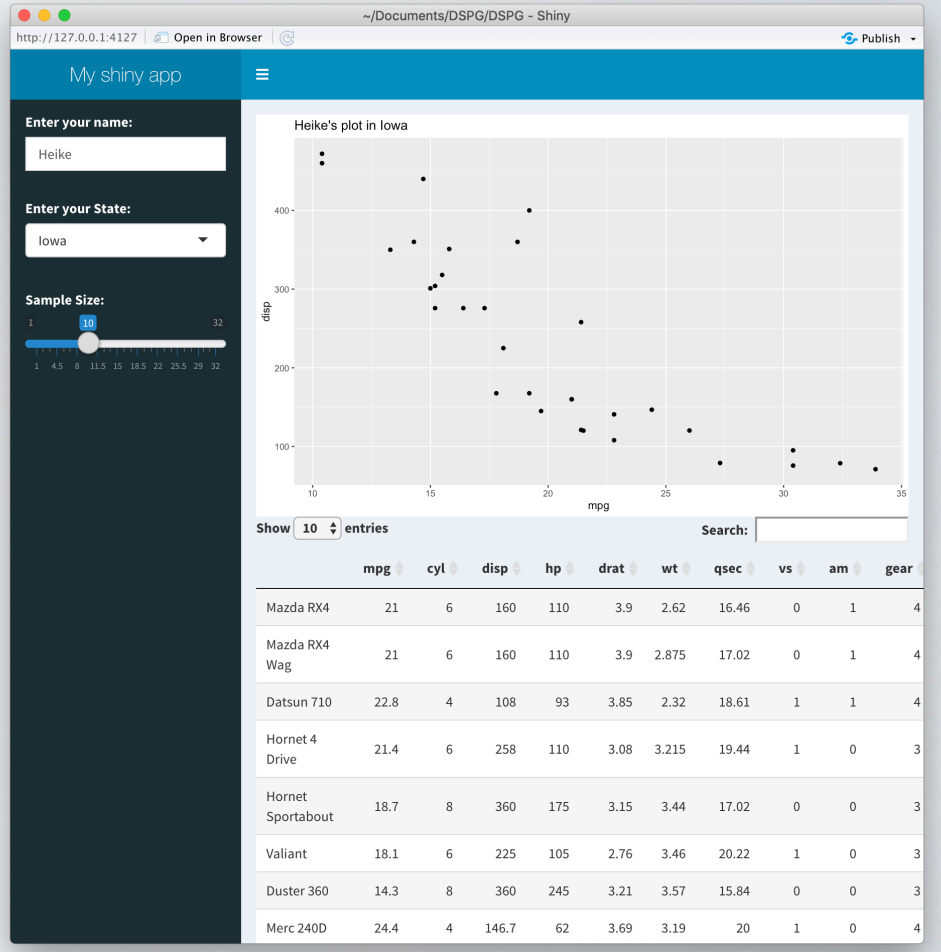
YOUR TURN

- Open the file app.R in folder 02e_minimal
- Include a slider with values between 1 and 32
- Make the number of data rows shown in the table dependent on the slider
- The answer is in 02f_minimal



FANCIER OUTPUTS

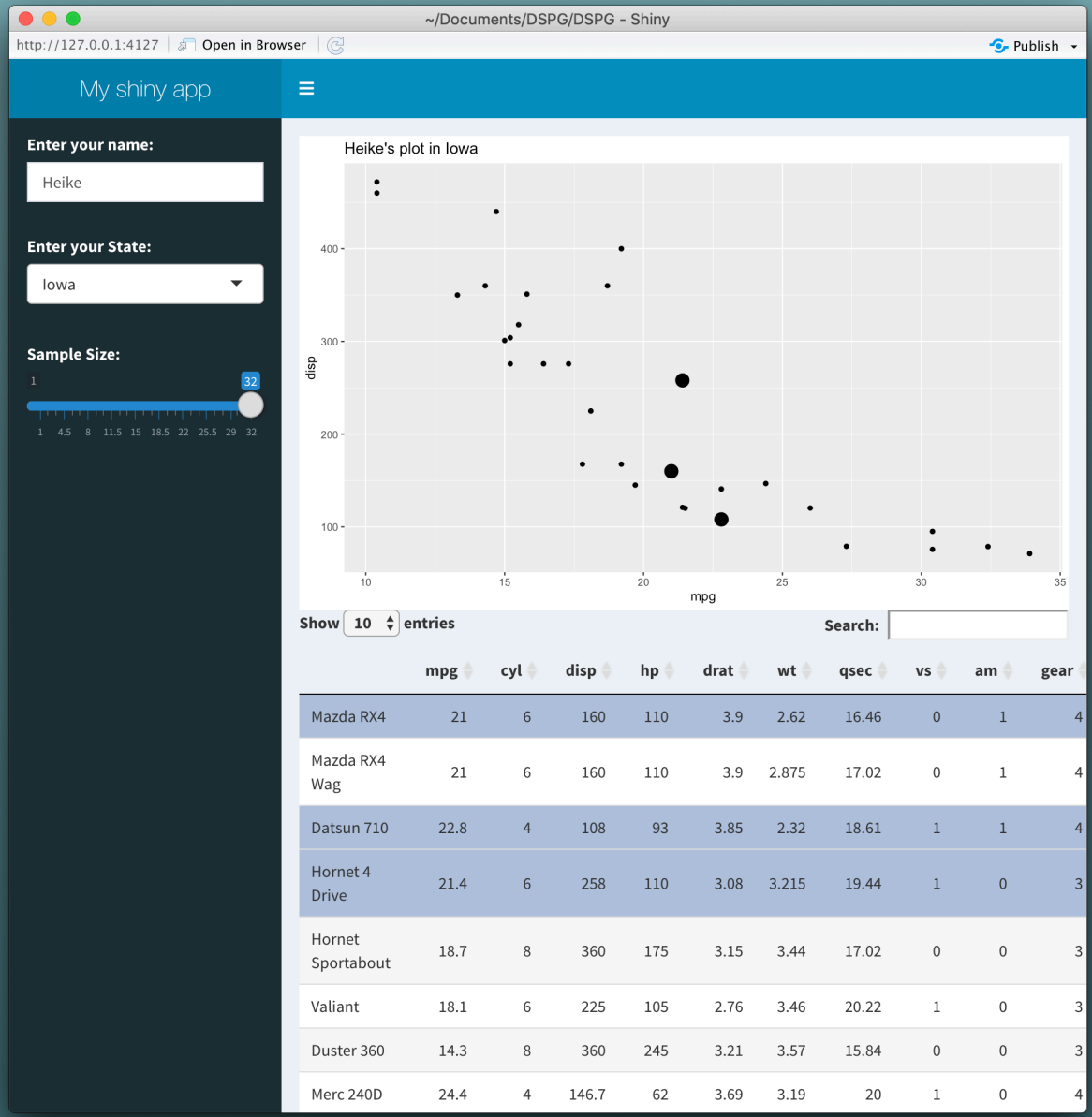
- The package DT provides functions to render data tables
- DTOutput and renderDT help defining and rendering the table
- See 02g_minimal.R



https://rstudio.github.io/DT/

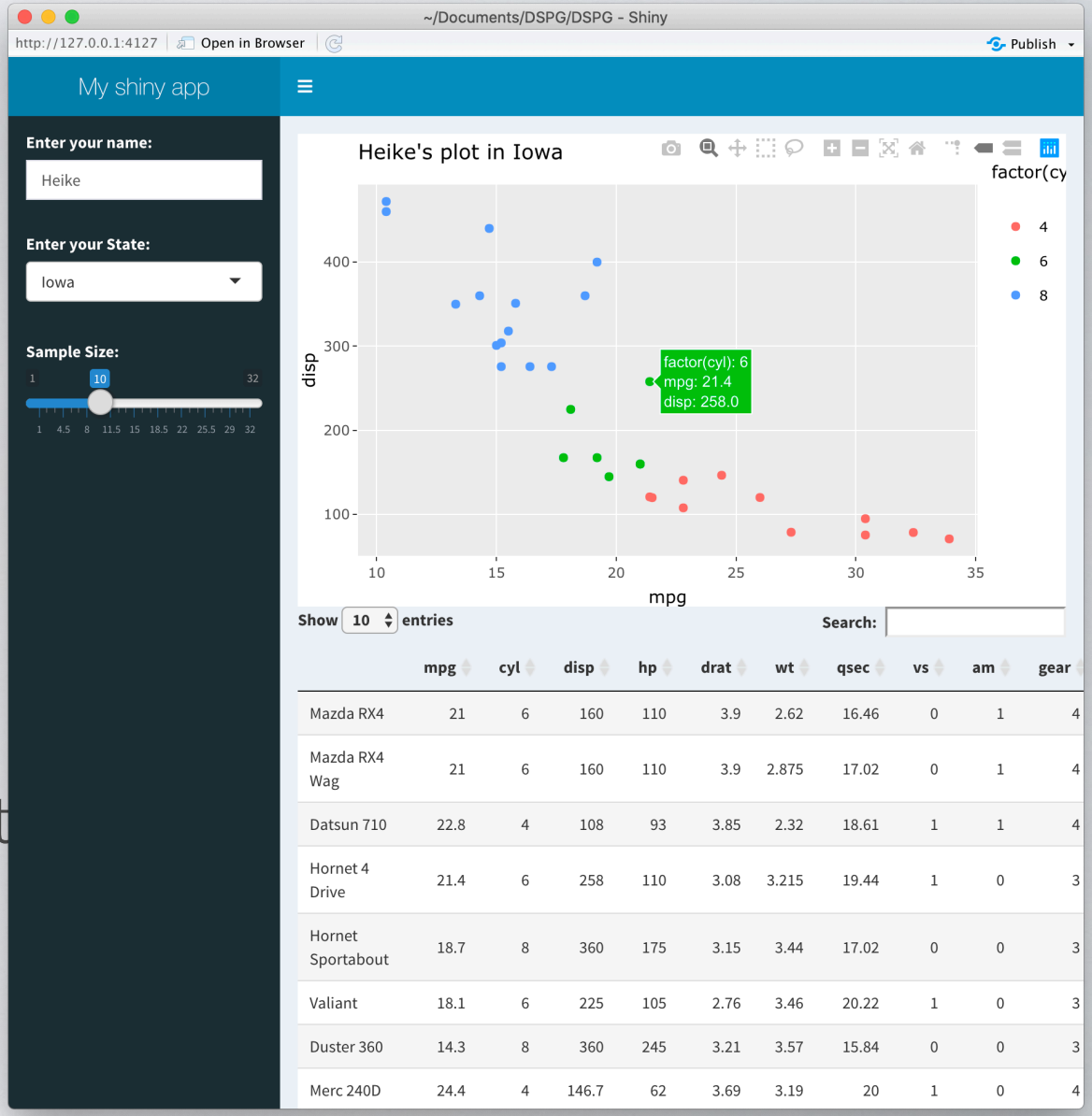
YOUR TURN

- Open the file app.R in folder 02h_minimal
- Make points bigger in the plot that are selected in the table
- input$tableId_rows_selected contains all of the ids of the selected rows
- The answer is in 02i_minimal

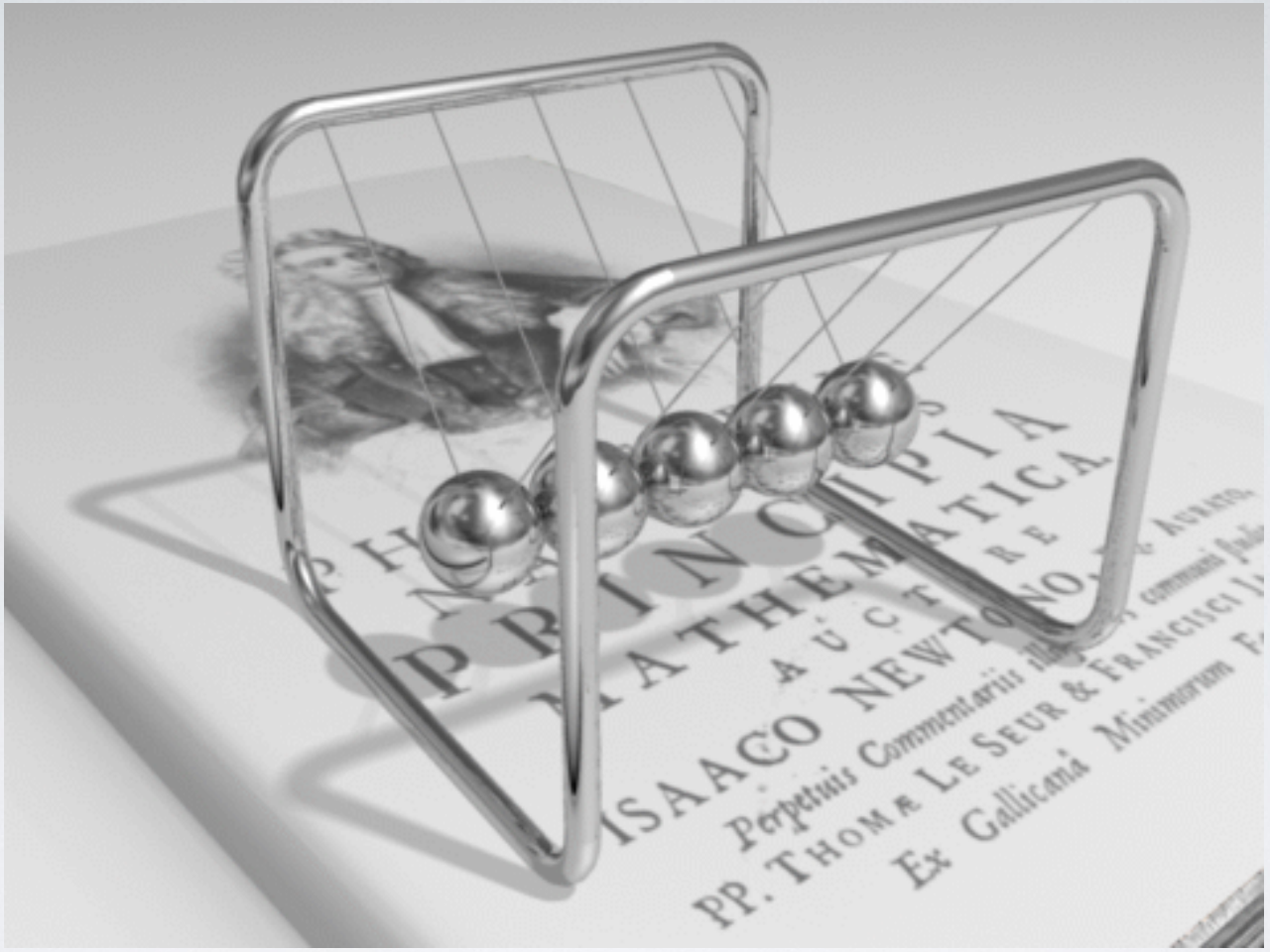


FANCIER OUTPUTS

- The package plotly implements interactive plots try plotly(gg) of a ggplot2 plot
- plotlyOutput and renderPlotly define and render the interactive plot
- See 02j_minimal.R
  https://plotly-r.com/



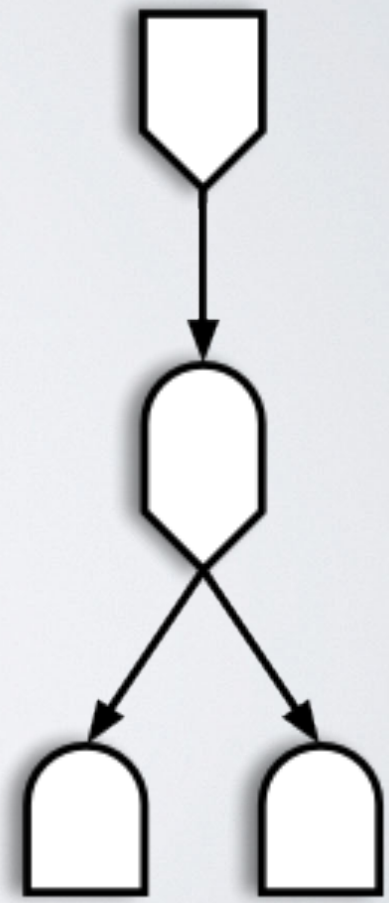
QUESTIONS?


PRINCIPIA
MATHEMATICA
AUCTORE
ISAACO NEWTONO
Perpetuis Commentariis
PP. THOMÆ LE SEUR & FRANCISCI
Ex Gallicanâ

REACTIVITY

ELEMENTS OF REACTIVITY

- Sources
  - Any input widget is a source
- Conductors
  - Use input and are being used further along
- Observers
  - Any output is an observer



TWO CONDUCTORS

- Reactive expressions and reactive events are two types of conductors
- Reactive expressions are the archetypical conductor: envelope functionality used in multiple places of an app, run evaluations only once and store current values.
- Reactive events are only triggered by specific events (such as a click on an action button)

REACTIVE EXPRESSIONS

```
rval <- reactive({
…
})
```

Called like a function as:

```
rval()
```

- reactive expressions are executed **lazily**, and their values are **cached**

- **Lazy:** evaluated only on demand, typically requested by a reactive endpoint.

- **Cached:** (re-)evaluated only when the value of a dependency changed.

REACTIVE EVENTS

```
rval <- eventReactive(actionbutton,{
…
})
```

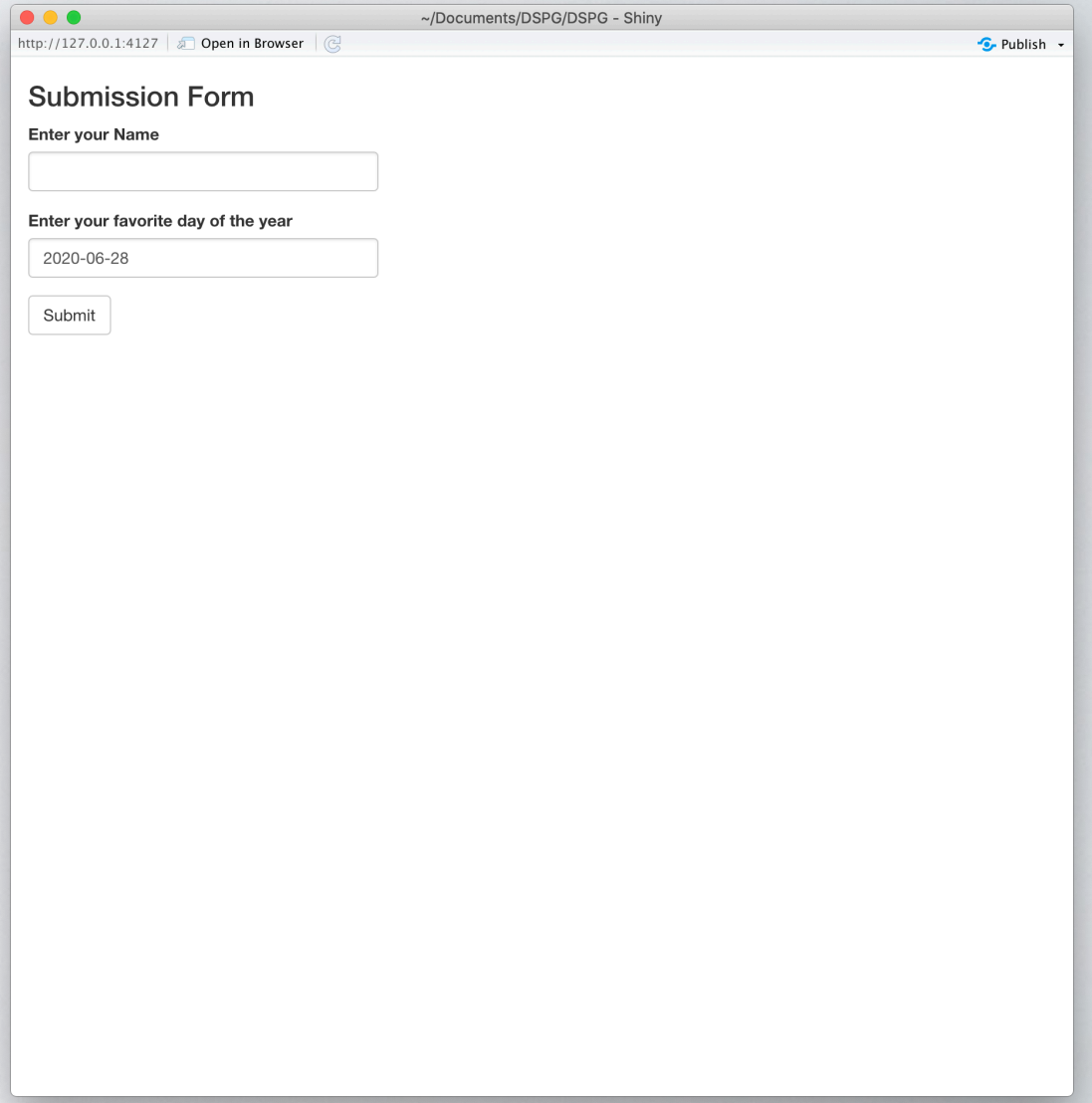
Called like a function as:

```
rval()
```

- reactive events are executed even more **lazily**: only on demand, typically requested by an action button

EXAMPLE: SUBMISSION FORM

- In RStudio, open file 03_submission.R
- Run the app (a couple of times)
- Turn on showcase mode



03_SUBMISSION.R

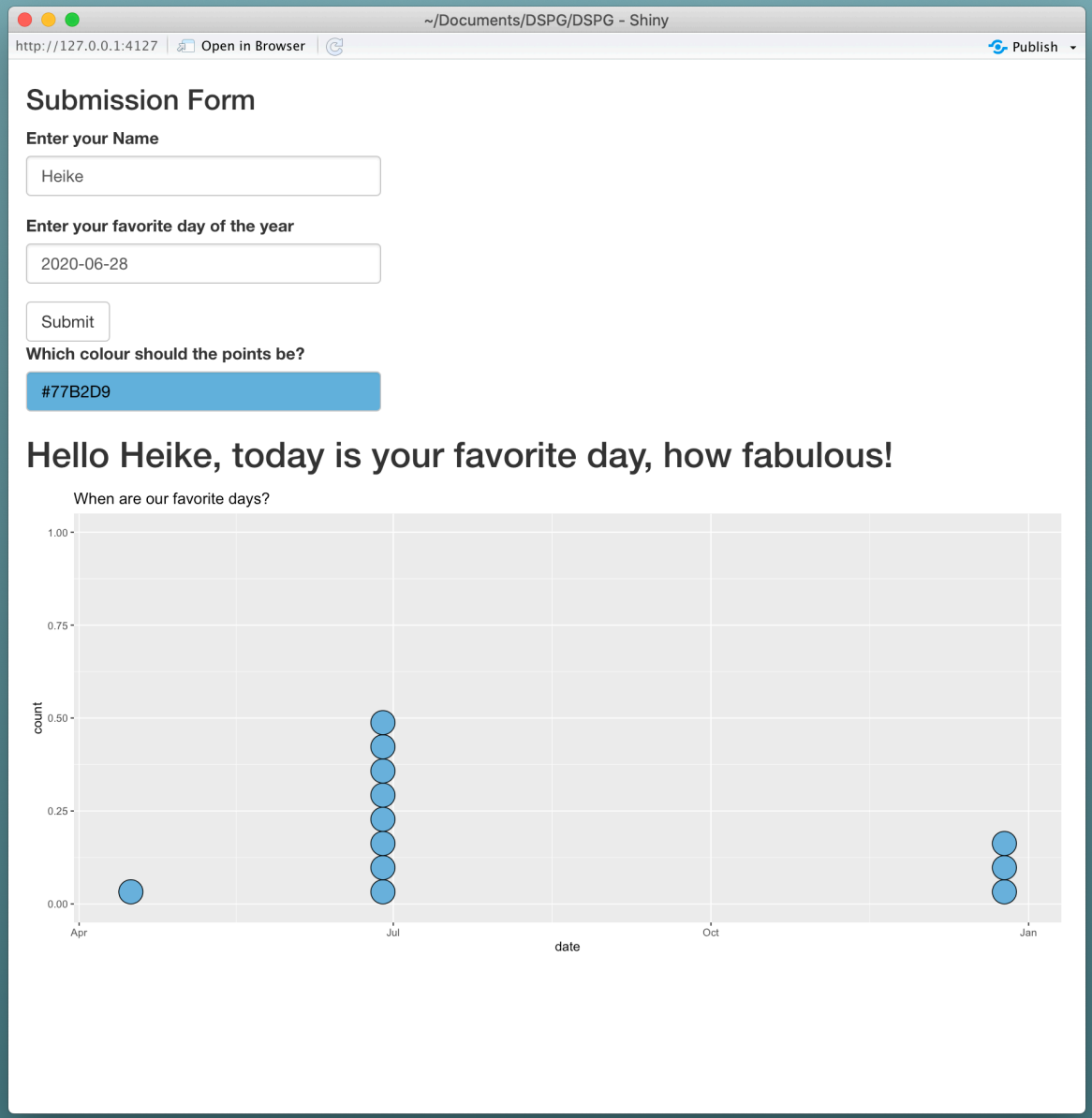
```
library(shiny)
library(lubridate) # install.packages("lubridate")
library(tidyverse)

ui <- fluidPage(
  h3("Submission Form"),
  textInput("name", "Enter your Name"),
  dateInput("date", "Enter your favorite day of the year", value=today()),
  actionButton("submit", "Submit"),
  h2(textOutput("greeting")),
  plotOutput("histogram")
)

server <- function(input, output, session) {
  rval_favs <- eventReactive(input$submit, {
    if (!file.exists("favorite-days.csv")) {
      favs <- data.frame(name=input$name, date = input$date, stringsAsFactors = FALSE)
    } else {
      favs <- read.csv("favorite-days.csv", stringsAsFactors = FALSE)
      favs <- rbind(favs, data.frame(name=input$name,
                                     date = as.character(input$date),
                                     stringsAsFactors = FALSE))
    }
    favs <- favs %>% mutate(date = lubridate::ymd(date))
    write.csv(favs, "favorite-days.csv", row.names=FALSE)

    favs
  })
```

03_SUBMISSION.R

```
library(shiny)
library(lubridate) # install.packages("lubridate")
library(tidyverse)

ui <- fluidPage(
  h3("Submission Form"),
  textInput("name", "Enter your Name"),
  dateInput("date", "Enter your favorite day of the year", value=today()),
  actionButton("submit", "Submit"),
  h2(textOutput("greeting")),
  plotOutput("histogram")
)

server <- function(input, output, session) {
  output$greeting <- eventReactive(input$submit, {
    daysdiff <- today() - input$date
    if (daysdiff > 1) dateout <- paste0("it has been ", daysdiff, " days since your favorite day."
    else if (daysdiff == 1) dateout <- paste0("yesterday was your favorite day, I hope you enjoyed
it!")
    else if (daysdiff == 0) dateout <- paste0("today is your favorite day, how fabulous!")
    else if (daysdiff == -1) dateout <- paste0("tomorrow is your favorite day, how exciting!")
    else dateout <- paste0("it is ", -daysdiff, " days until your favorite day.")


    paste0("Hello ", input$name, ", ", dateout)
  })
```

YOUR TURN

- Open the file 03_submission.R
- The package colourpicker implements a color wheel as input widget
- Allow users to change the color of the dots in the dot plot
- What other interactive elements can you think of adding?
- The answer is in 03b_submission.R



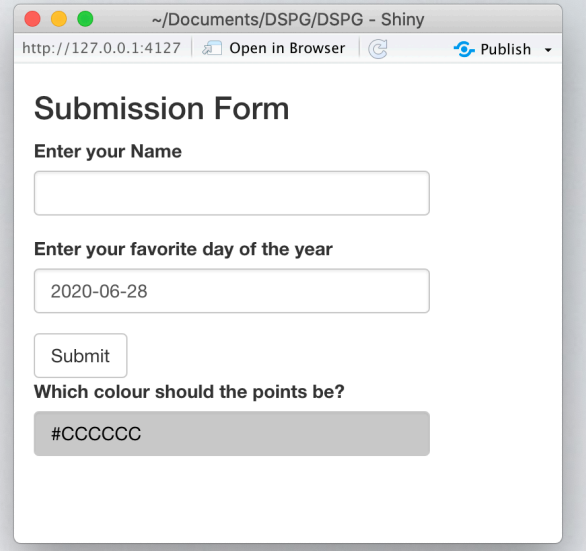
CONDITIONAL PANELS

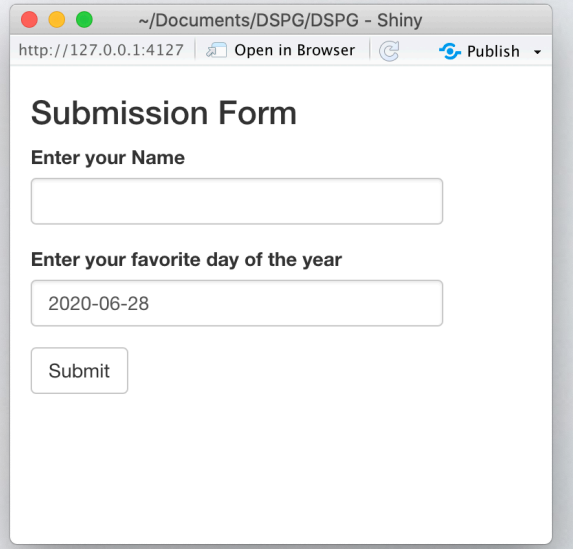
- Showing a color picker before needed … might be confusing
  users of the app

- `conditionalPanel(condition, …)`
  allows us to encapsulate elements of the user interface and only show when 'condition' is fulfilled

- Here, a condition of
  `condition = "input.submit > 0"`
  is true when the submit button was pressed at least once

- This is implemented in 03c_submission.R

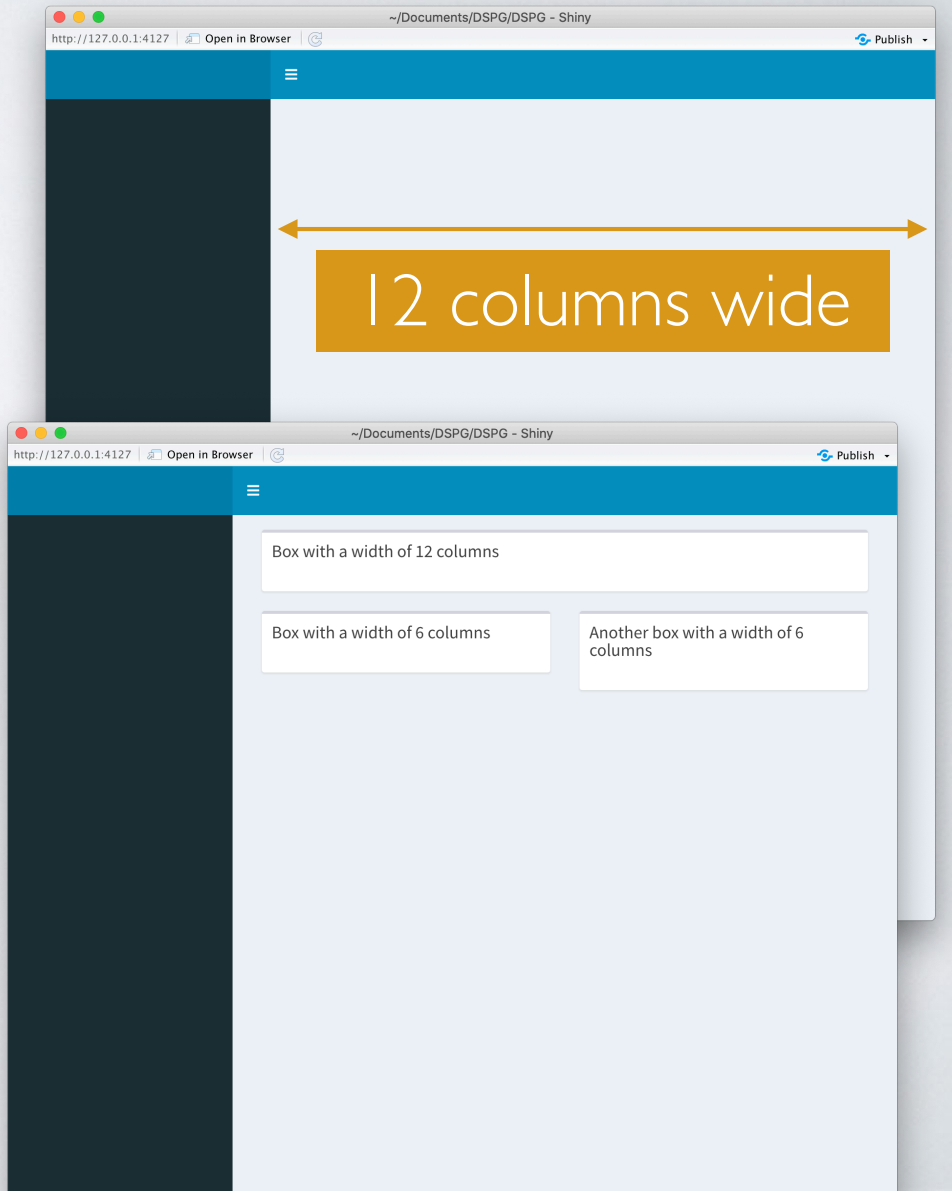




LAYOUT OF DASHBOARDS

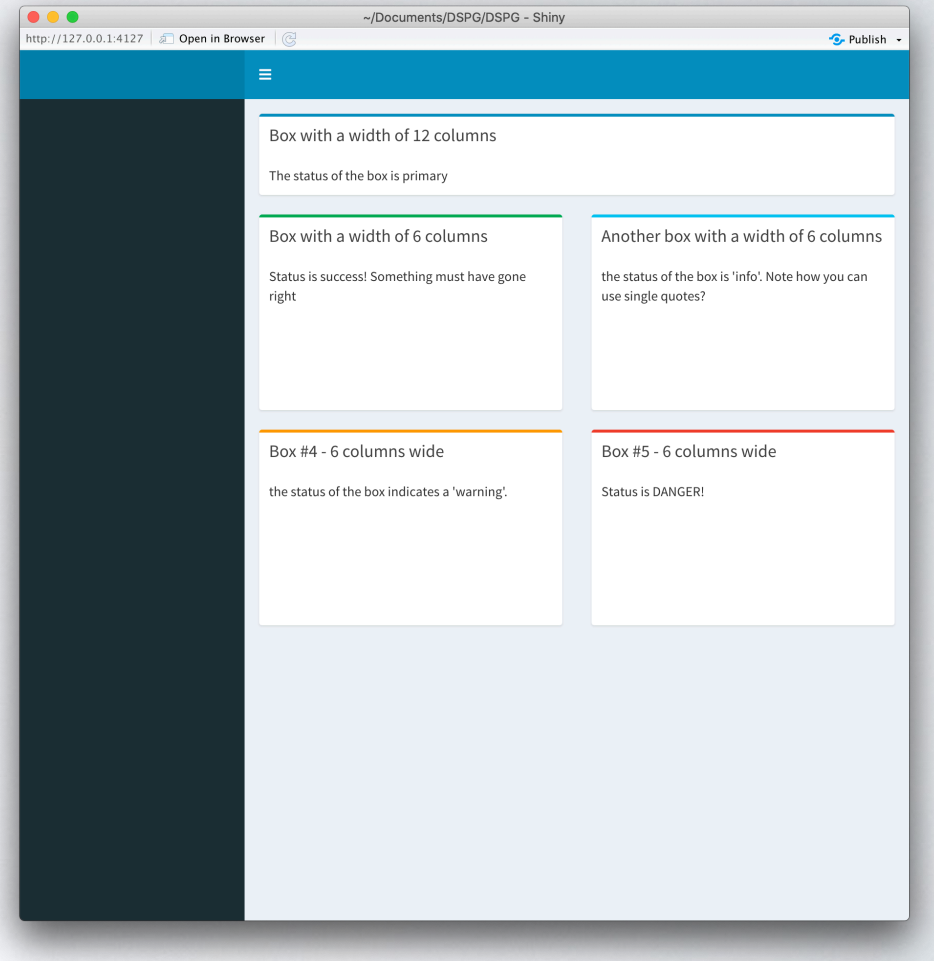
- The body can be laid out in a grid - either row based or column based
- Structure is introduced by boxes:

```
box(..., title = NULL,
width = 6, height = NULL)
```



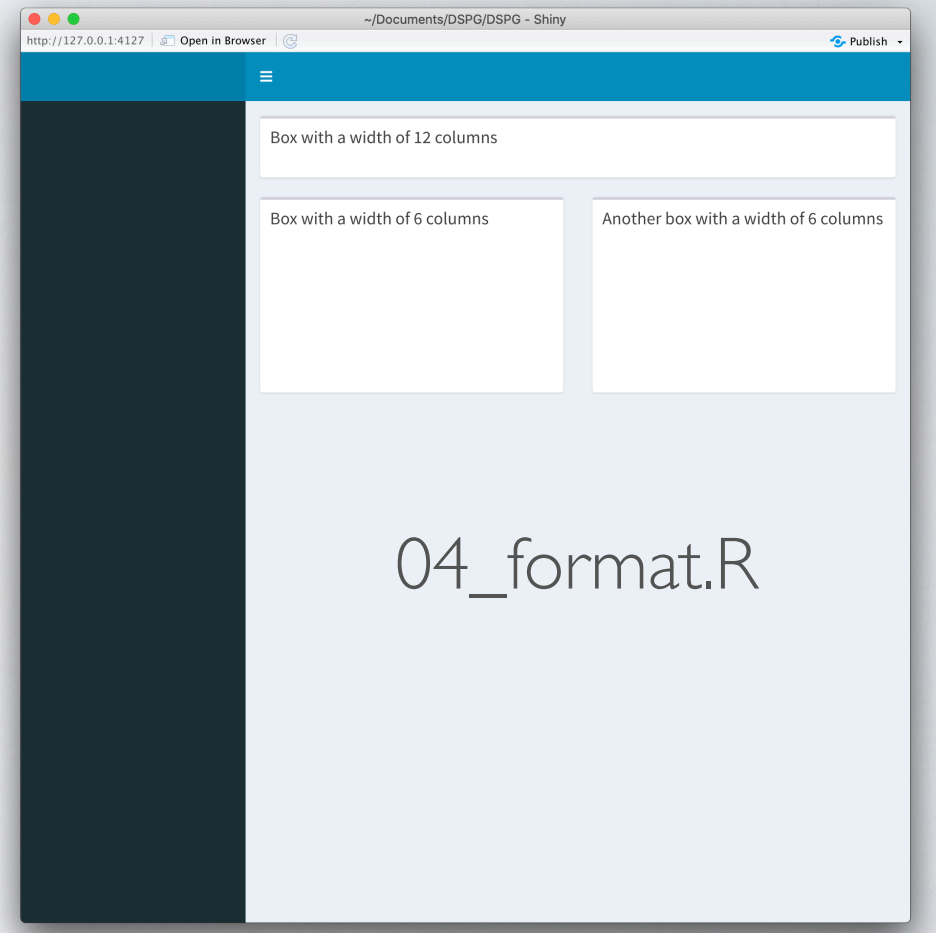
BOXES

- Boxes help with structuring output
- Boxes also have a `status` parameter
- Status is shown as a colored bar along the top of a box
- `?validStatuses` are `primary, success, info, warning, danger`



ROW BASED LAYOUT

- Body is wrapped in a fluidRow function
- Tops of boxes are aligned
- Bottom of the boxes can be aligned by setting the height (in pixel)

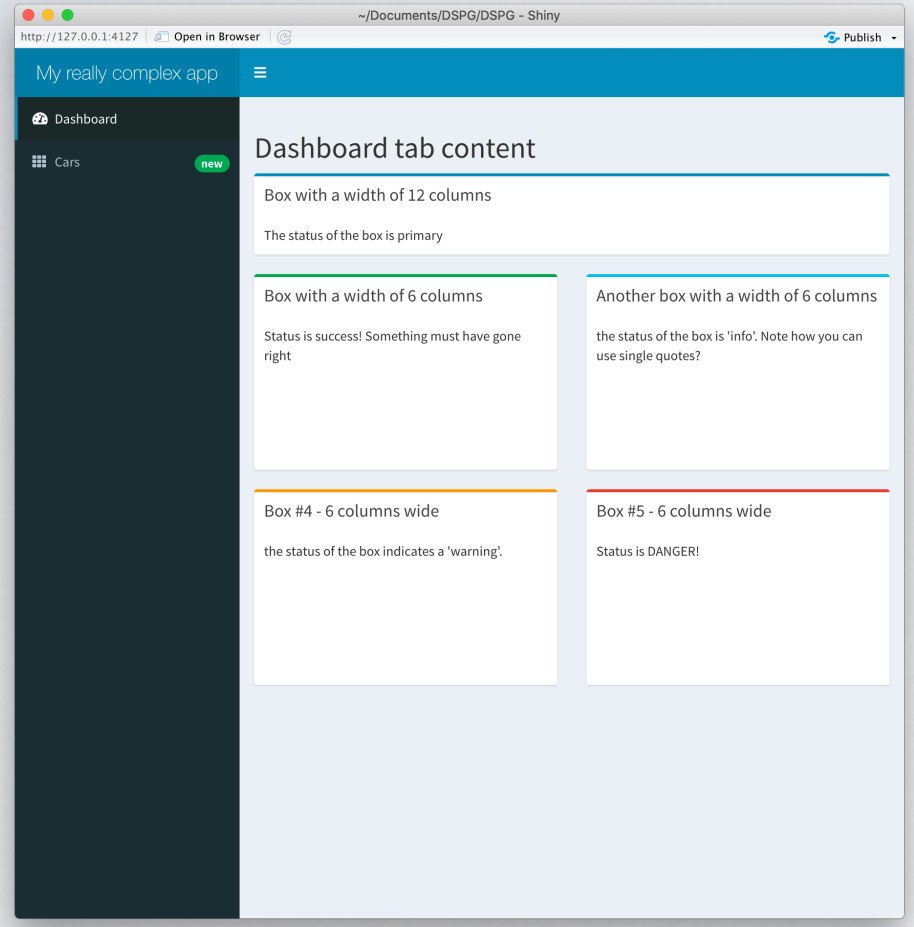


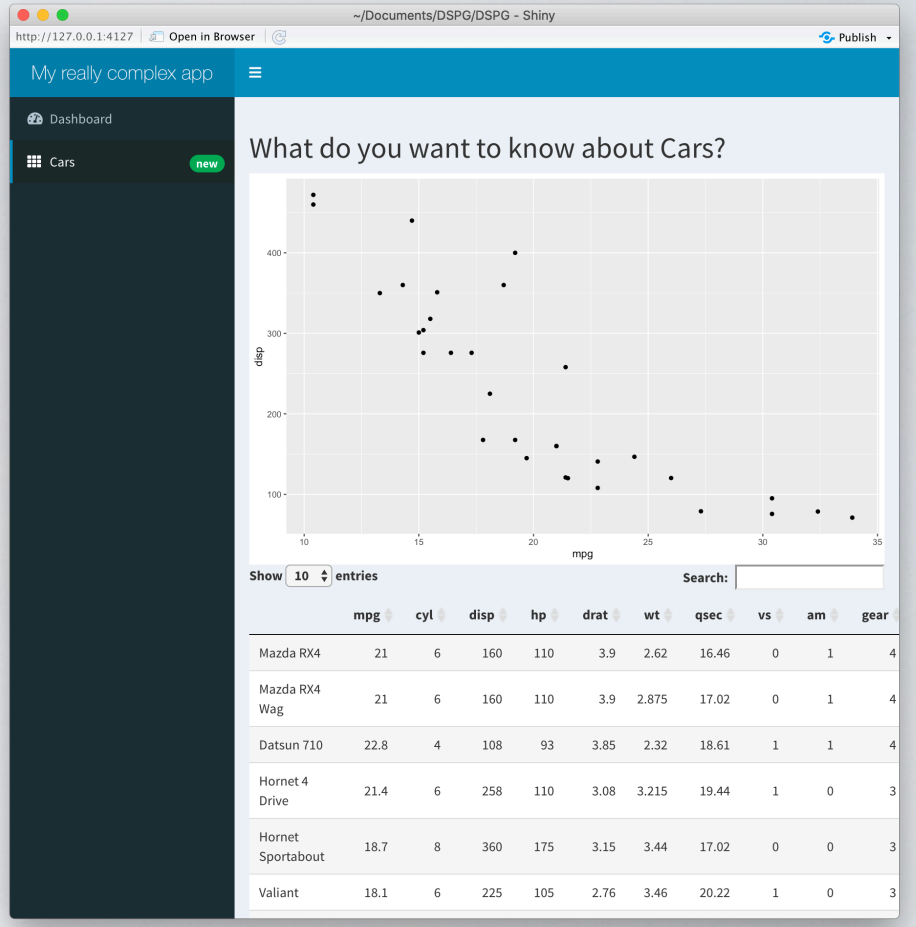

```
body <- dashboardBody(
  fluidRow(
  box(title = "Box with a width of 12 columns", width = 12),
  box(title = "Box with a width of 6 columns", width = 6, height = 200),
  box(title = "Another box with a width of 6 columns", width = 6, height = 200)
  )
)
```

OTHER LAYOUTS

- In **column** based layouts, the body is wrapped in a fluidRow function
  - Height of boxes are aligned,
    each column has to define a width, boxes are aligned in width.
- In mixed layouts fluidRow and column can be used sequentially

TABS IN DASHBOARDS



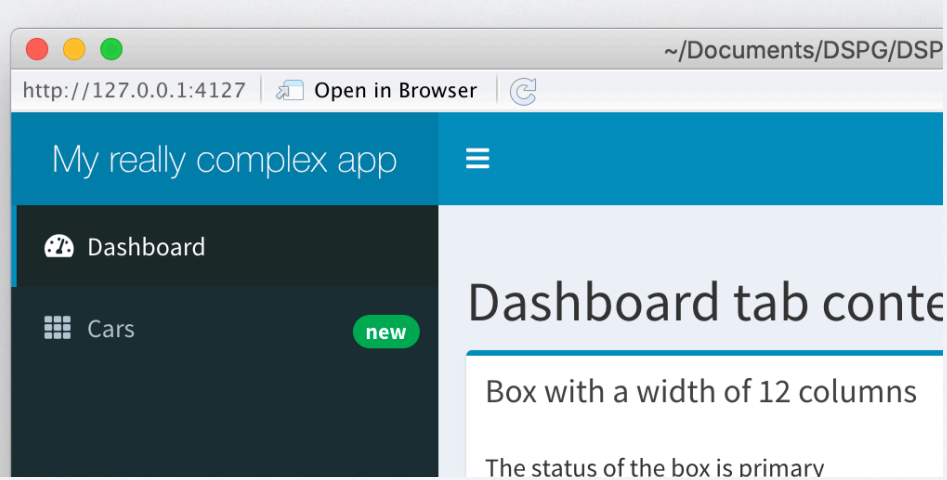


- The sidebar menu can be used to introduce tabs for quick navigation

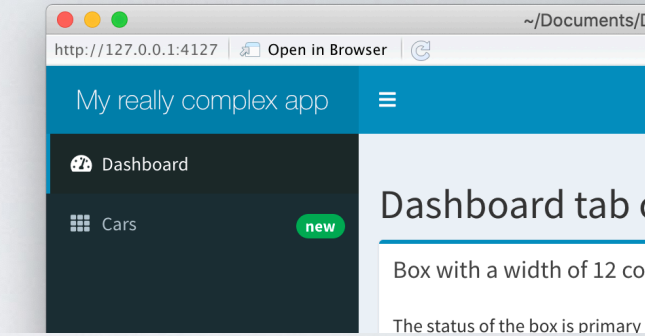
05_tabsets.R

TABS IN DASHBOARDS

```
sidebar <- dashboardSidebar(
  sidebarMenu(
    menuItem("Dashboard", tabName = "dashboard",
             icon = icon("dashboard")),
    menuItem("Cars", icon = icon("th"), tabName = "cars",
             badgeLabel = "new", badgeColor = "green")
  )
)
```

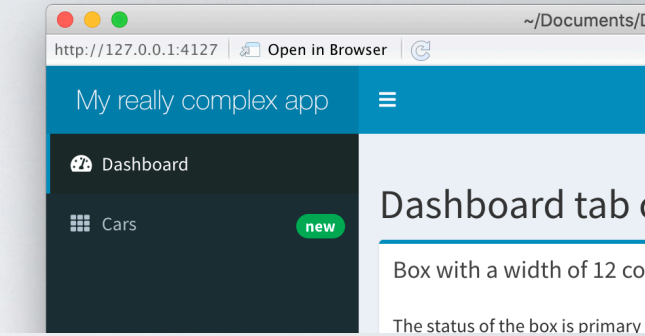


TABS IN DASHBOARDS




```
sidebar <- dashboardSidebar(
  sidebarMenu(
    menuItem("Dashboard", tabName = "dashboard",
             icon = icon("dashboard")),
    menuItem("Cars", icon = icon("th"), tabName = "cars",
             badgeLabel = "new", badgeColor = "green")
  )
)
        body <- dashboardBody(
          tabItems(
            tabItem(tabName = "dashboard",
                    h2("Dashboard tab content"),
                    fluidRow(
                      box(title = "Box with a width of 12 columns", width = 12,
                          status = "primary", "The status of the box is primary"),
                      …
                      box(title = "Box #5 - 6 columns wide",
                          status = "danger", "Status is DANGER!",
                          width = 6, height = 200)
                    )
            ),

            tabItem(tabName = "cars",
                    h2("What do you want to know about Cars?"),
                    plotOutput("myplot"),
                    DTOutput("mytable")
            )
```

TABS IN DASHBOARDS




```
sidebar <- dashboardSidebar(
  sidebarMenu(
    menuItem("Dashboard", tabName = "dashboard",
             icon = icon("dashboard")),
    menuItem("Cars", icon = icon("th"), tabName = "cars",
             badgeLabel = "new", badgeColor = "green")
  )
)
        body <- dashboardBody(
          tabItems(
            tabItem(tabName = "dashboard",
                    h2("Dashboard tab content"),
                    fluidRow(
                      box(title = "Box with a width of 12 columns", width = 12,
                          status = "primary", "The status of the box is primary"),
                      …
                      box(title = "Box #5 - 6 columns wide",
                          status = "danger", "Status is DANGER!",
                          width = 6, height = 200)
                    )
            ),

            tabItem(tabName = "cars",
                    h2("What do you want to know about Cars?"),
                    plotOutput("myplot"),
                    DTOutput("mytable")
            )
```

YOUR TURN

YOUR TURN

- Option 1
  - Longer form your turn: explained later
- Option 2
  - Prepping for tomorrow: get together with your team members and make a (hand-drawn) sketch of what you would like your project dashboard to look like.

YOUR TURN - WHAT?

- Build a shiny dashboard with at least three tabs
- Minimal requirements:
  - Incorporate at least
    one (interactive?) plot,
    three user widgets
  - One of the tabs should include the population numbers published by the census bureau (co_est2019_alldata.csv)
  - Be creative!

YOUR TURN - HOW & WHEN?

- We will use the break rooms to give teams of about 5 a private room
- Haley, Mattie, and Heike will check in periodically and help solve problems
- Work on the app until 1:15.
- From 1:15 - 2:00 each team will present their app for a few minutes (3-5mins) to all of us