Creating Interactive Data Visualizations with

GW Libraries Workshop
Dan Kerchner ~ Fall 2023

go.gwu.edu/rshinyworkshop

Shiny is a web application framework for R

Let's look at what Shiny can do

shiny.posit.co/r/gallery

shiny.library.gwu.edu/aciar/StatApps

rstudio.github.io/shinydashboard/examples.html

Why R Shiny?

- You can add interactivity to R data visualizations you've built using *any* R libraries

- Other types of interactivity are also possible with some R libraries when deployed on R Shiny
  - learnr - interactive tutorials
  - flexdashboard - "dashboard" style interactive pages

Goals

Today's Goal

- Create a Shiny app
- Publish a Shiny app in shinyapps.io

Today's Data Set: Old Faithful

Azzalini, A. and Bowman, A.W., 1990. A look at some data on the Old Faithful geyser. Journal of the Royal Statistical Society: Series C (Applied Statistics), 39(3), pp.357-365. doi.org/10.2307/2347385

- Waiting times between eruptions, and duration of eruptions, of the Old Faithful Geyser, Yellowstone National Park, WY

Structure of an R Shiny app

```
library(shiny)
ui <- fluidPage(
  # Comma-separated list of components
  # collecting input from the user
  # using output in rendering
)

server <- function(input, output, session) {
  # server logic
  # using parts of input
  # and setting parts of output
}

# Run the app - ties together ui and server
shinyApp(ui, server)
```

← FORM

shiny.posit.co/r/articles/build/layout-guide

← FUNCTION

mine-cetinkaya-rundel

Connecting inputs and outputs

```
ui <- fluidPage(
  # ...
  sidebarLayout(
    sidebarPanel(
      sliderInput("bins",
                  "Number of bins:",
                  min = 1,
                  max = 50,
                  value = 30)
    ),
    mainPanel(
      plotOutput("distPlot")
    )
  )
)
```

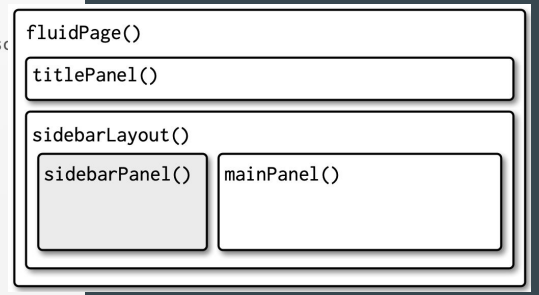
```
server <- function(input, output) {
  output$distPlot <- renderPlot({
    x    <- faithful[, 2]
    bins <- seq(min(x), max(x),
                length.out = input$bins + 1)

    hist(x, breaks = bins, col = 'darkgray',
         border = 'white')
  })
}
```

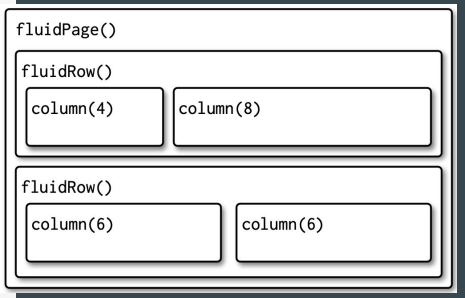
Layouts: Single-page layouts

There are different layouts available:

```
fluidPage(
  titlePanel(
    # app title/desc
  ),
  sidebarLayout(
    sidebarPanel(
      # inputs
    ),
    mainPanel(
      # outputs
    )
  )
)
```



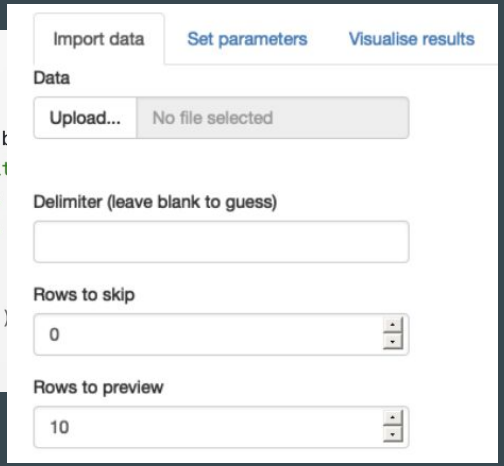

```
fluidPage(
  fluidRow(
    column(4,
      ...
    ),
    column(8,
      ...
    )
  ),
  fluidRow(
    column(6,
      ...
    ),
    column(6,
      ...
    )
  )
)
```



Layouts: Multi-page layouts

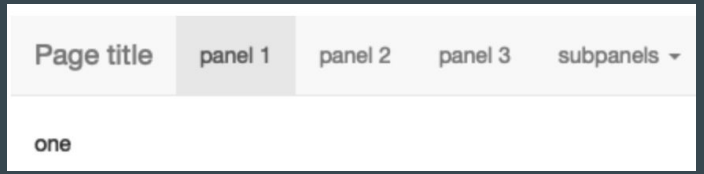
tabsetPanel:

```
ui <- fluidPage(
  tabsetPanel(
    tabPanel("Import data",
      fileInput("file", "Data", b
      textInput("delim", "Delimit
      numericInput("skip", "Rows
      numericInput("rows", "Rows
    ),
    tabPanel("Set parameters"),
    tabPanel("Visualise results")
  )
)
```



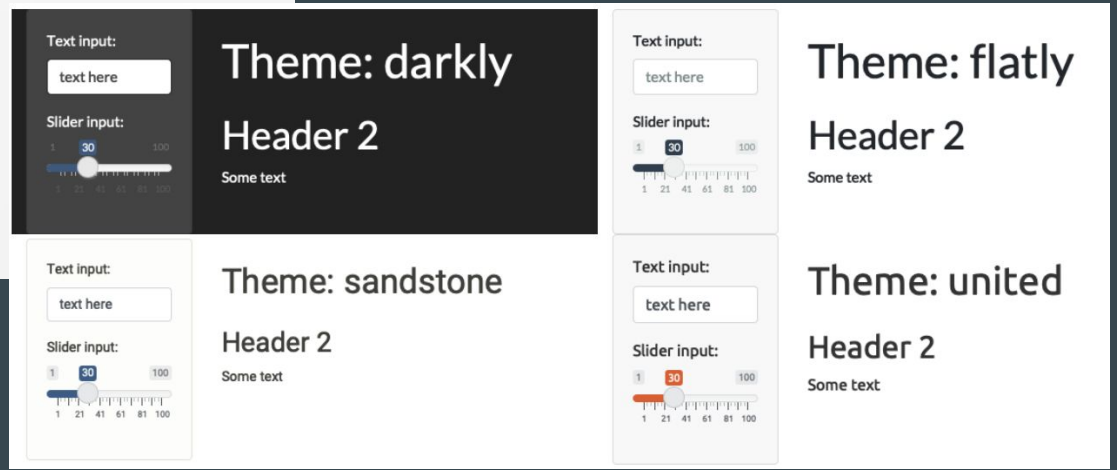
navlistPane

navbarPage:



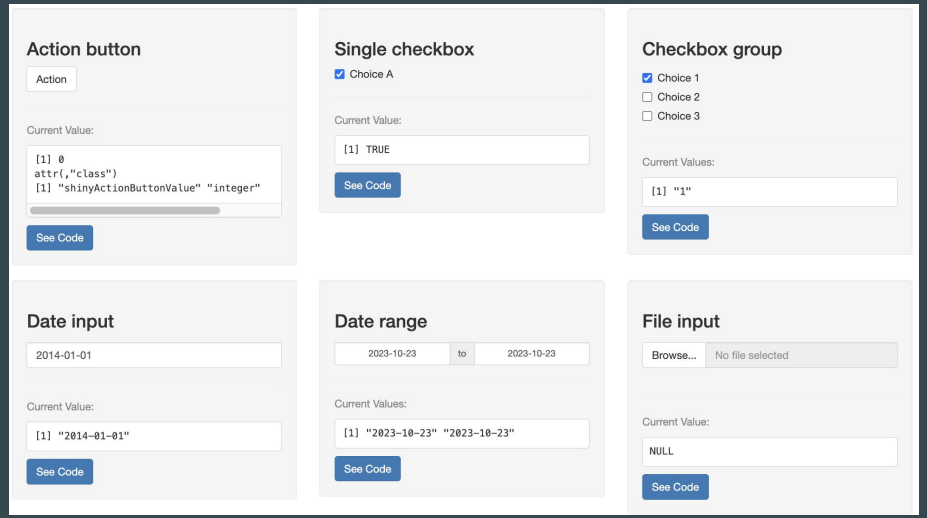
Themes!

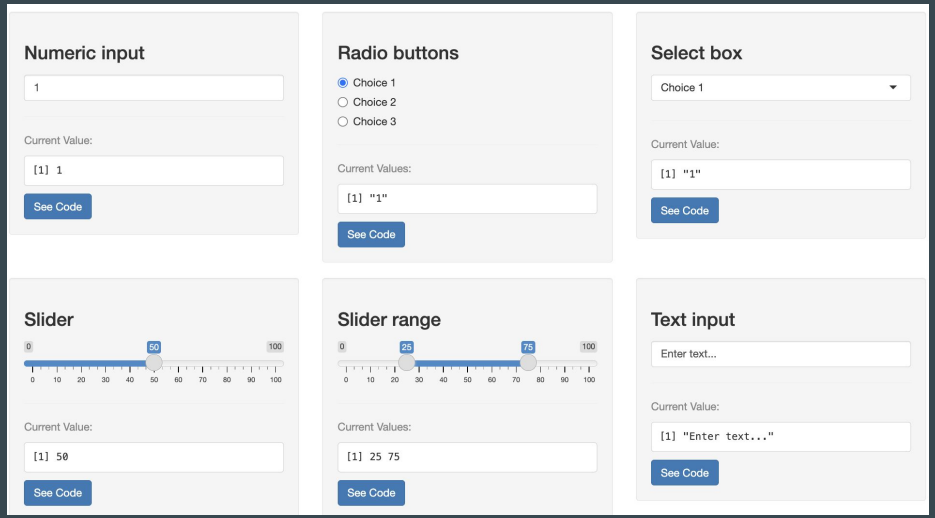
```
ui <- fluidPage(
  theme = bslib::bs_theme(bootswatch = "darkly"),
  sidebarLayout(
    sidebarPanel(
      textInput("txt", "Text input:", "text here"),
      sliderInput("slider", "Slider input:", 1, 100, 30)
    ),
    mainPanel(
      h1(paste0("Theme: darkly")),
      h2("Header 2"),
      p("Some text")
    )
  )
)
```



Input widget types

See shiny.posit.co/r/gallery/widgets/widget-gallery

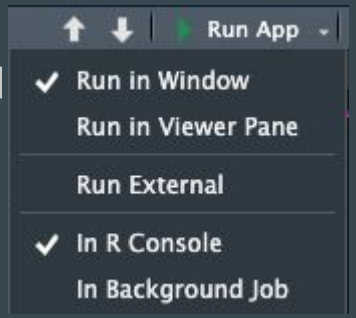




Workflow - Steps to create an RShiny app

(you may need to first: `install.packages("shiny")`)

- Create the "empty" app
- Lay out the app and write the **u**
- Add functionality bit by bit, test incrementally
- Deploy/Distribute

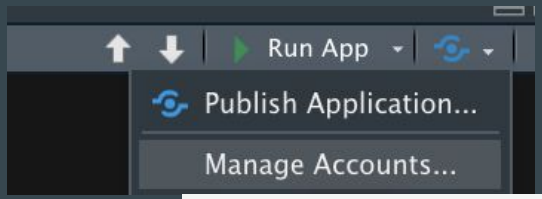


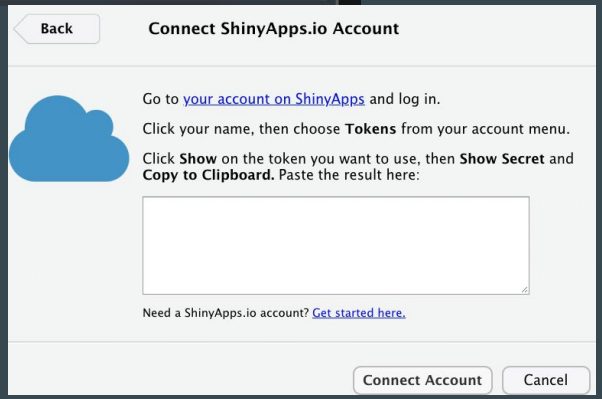
Options for sharing your app with others

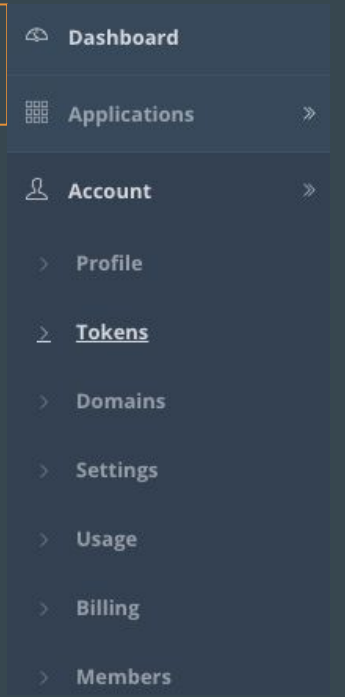
- Share for people to run locally
  shiny.posit.co/r/articles/share/deployment-local
  - Post to a URL as zip file - `runUrl('https://github.com/rstudio/shiny_example/archive/master.zip')`
  - Place on GitHub
    - as repository - `shiny::runGitHub('shiny_example', 'rstudio')`
    - as gist - `shiny::runGist('3239667')`
- Share for people to access on the web
  shiny.posit.co/r/articles/share/deployment-web
  - Publish to shinyapps.io - `https://kerchner.shinyapps.io/rshiny-test/`
  - Publish to another shiny server - `https://shiny.library.gwu.edu/aciar/StatApps/Normal_prob/`

Publishing your app to shinyapps.io

- Create account
- Create a token
- Copy the token (***with secret***) into RStudio "Connect Account"
- Publish your application
- View at: *youraccount.shinyapps.io/yourapp*



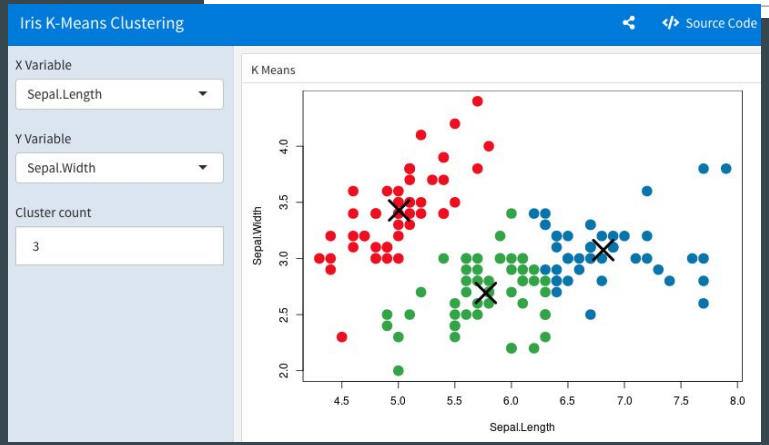




Dashboards with RShiny

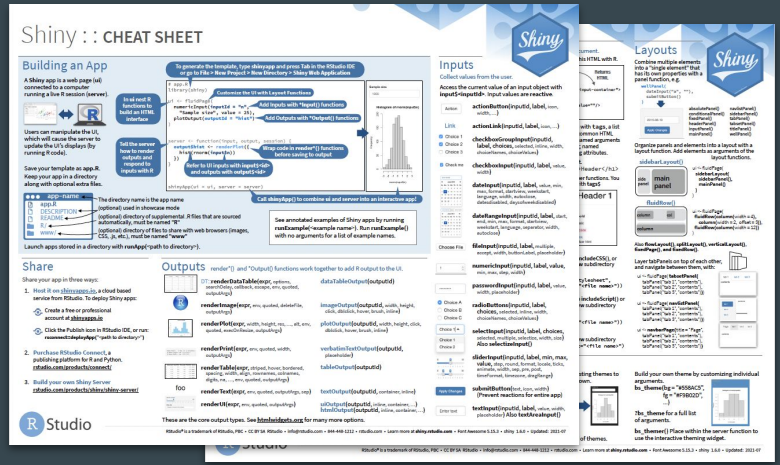
- Two options (see shiny.posit.co/r/articles/build/dashboards)
  - shinydashboard
    rstudio.github.io/shinydashboard
  - flexdashboard
    pkgs.rstudio.com/flexdashboard
    This is an RMarkdown document that "knits" to HTML

flexdashboard	shinydashboard
R Markdown	Shiny UI code
Super easy	Not quite as easy
Static or dynamic	Dynamic
CSS flexbox layout	Bootstrap grid layout



Shiny Resources

- Reference: shiny.posit.co/reference/shiny/latest
- Tutorial : shiny.posit.co/tutorial
- Online book: mastering-shiny.org
- Gallery : shiny.posit.co/r/gallery
- Cheat sheet : shiny.posit.co/articles/cheatsheet.html
- LinkedIn Learning
- Library Resources



Interactive Data Visualization - in R, *beyond* Shiny

- Libraries
  - plotly, d3heatmap, heatmaply, leaflet
- Resources
  - htmlwidgets.org - htmlwidgets for R
  - r-graph-gallery.com/interactive-charts.html
  - posit.co/resources/cheatsheets
  - ggplot2-book.org
  - rkabacoff.github.io/datavis
  - worldbank.github.io/r-econ-visual-library

Interactive Data Visualization - Not in R

- Tableau
- Power BI
- Python libraries
- ...

Statistics+R help @ GW

R-Statistics Appointments:
academiccommons.gwu.edu/data-consulting

Also...

Appointments with me:
calendly.com/kerchner

Coding consultations (Python, R, git, etc.):
calendly.com/gwul-coding

Thanks!

Dan Kerchner kerchner@gwu.edu