Applied Statistics Seminar Series: R Shiny

David Aaby
Senior Biostatistician
Biostatistics Collaboration Center
Department of Preventive Medicine
March 15, 2022
What is R Shiny?

• An R package that makes it easy to turn your analyses into interactive web apps using R

• Needs no knowledge of HTML, CSS, or Javascript
Why use Shiny?
Why use Shiny?

- Exploratory Data Analysis

Why use Shiny?

- Dashboards

https://edgarruiz.shinyapps.io/db-dashboard/
Why use Shiny?

- Teaching statistics

http://shiny.calpoly.sh/MLE_Binomial/
Why use Shiny?

- Apps for your collaborators
Why use Shiny?

- Fun!

https://daattali.com/shiny/lightsout/
Create your first Shiny app
Structure of a Shiny App

- 4 main parts to creating a Shiny app:
  - `library(shiny)`
  - `ui <- fluidPage()`
  - `server <- function(input, output) {}`
  - `shinyApp(ui = ui, server = server)`
Structure of a Shiny App

- `library(shiny)`

This loads the shiny package
Structure of a Shiny App

- `ui <- fluidPage()`

This defines the user interface, which is the HTML webpage that you will create and interact with.
• `server <- function(input, output) {}`

This specifies the behavior of the app.

This is where all of the behind the scenes machinery of the app lives that will get calculated, plotted, analyzed, etc.
Structure of a Shiny App

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

This creates a shinyApp object that can then be run and output as an HTML webpage.
Shiny Resources

• Shiny Cheat Sheet

• Mastering Shiny
  – https://mastering-shiny.org/index.html
GO MOMs
GO MOMs

- Glycemic Observation and Metabolic Outcomes in Mothers and Offspring

- Prospective, observational study involving an anticipated 2150 participants across 9 study sites

- 4 study visits plus additional data collected at delivery and post partum
GO MOMs

• Study visit windows based on gestational age
  – 10 – 14 weeks
  – 16 – 20 weeks
  – 24 – 28
  – 32 – 36 weeks gestation
• RAs across 9 study sites need to quickly calculate gestational age, estimated delivery date, and study visit windows for recruitment and scheduling of participants
• We created a Shiny app for this
GO MOMs Shiny App

• Inputs
  – Participant ID
  – In vitro fertilization
  – LMP date (if available)
  – Gestational age from ultrasound
  – Ultrasound date
  – Today’s Date

• Outputs
  – Gestational age today
  – Estimated delivery date
  – Date ranges for each study visit window
  – Save output as .csv (optional)
GO MOMs - UI
GO MOMs Shiny App - ui

- Title

```
  h1("GO MOMs – Estimated Study Visits",
      style = "color:DarkMagenta")
```

- Typically you would use titlePanel() for the title of your app.
- This allowed for more customization
• Input ID

textInput("id",
    "Participant ID",
    value="A9999")
• In vitro fertilization

radioButtons("in_vitro",
    "In Vitro Fertilization:", 
    c("No", "Yes"))
• Conditional Panels
• We want additional inputs to appear, conditional on the answer of the In Vitro Fertilization radio button

```r
conditionalPanel(
  condition = "input.in_vitro == 'No'",
  .
  .
  .
)
```
Input gestational age from ultrasound in terms of weeks and days

```r
splitLayout(
  numericInput("gestage_wks",
    "Weeks",
    value=1, min=0, max=40),

  numericInput("gestage_days",
    "Days",
    value=0, min=0, max=6))
)
GO MOMs Shiny App - ui

- Input Dates

```r
splitLayout(
  dateInput("ultrasound_dt",
    "Ultrasound Date:",
    value = "2021-01-01",
    format="mm/dd/yyyy"),
  dateInput("todays_dt",
    "Todays Date:",
    value = Sys.Date(),
    format="mm/dd/yyyy")
)
```
• Output text for gestational age based on today’s date

verbatimTextOutput("gest_age_today_text")
• Output table (data frame)

```r
fluidRow(
  column(12, tableOutput('table'))
)
```
• Download button for saving results from the app

downloadButton("downloadData",
   "Download study visit date ranges")
GO MOMs - Server
Takes inputs from ui as arguments into the output function below

```r
output$dateText <- renderText(

  edd = format(edd, "%m/%d/%Y")
  paste("The study defined estimated due date is",
  as.character(edd))

})
```
Output a data frame of study visit windows

df.visits = reactive({
    
    data.frame(ID, Visit, Earliest_Date, Latest_Date, EDD)
})

output$table <- renderTable(df.visits())
Save a data frame of study visit windows as a .csv file

```r
output$downloadData <- downloadHandler(
    filename = function() {
        paste("study_visits_id_",
            input$id, ".csv", sep = "")
    },
    content = function(file) {
        write.csv(df.visits(), file, row.names = FALSE)
    };
)
```
Share your app: One possible approach

• Upload to Gist
  – Gist is a product of Github
  – A way to anonymously post and share files online
• To share your app as a gist:
  – Copy and paste your app.R files to the Gist web page.
  – Copy the URL that GitHub gives the Gist.
  – Once you have made a gist, your users can launch the app with
    
    ```
    shiny::runGist("12345ABCDEF", launch.browser=T)
    ```
  – Where “12345ABCDEF” is the unique number that appears at the end of your Gist address
Thank you!