DASHBOARDING IN SHINY

By: Tonya Filz

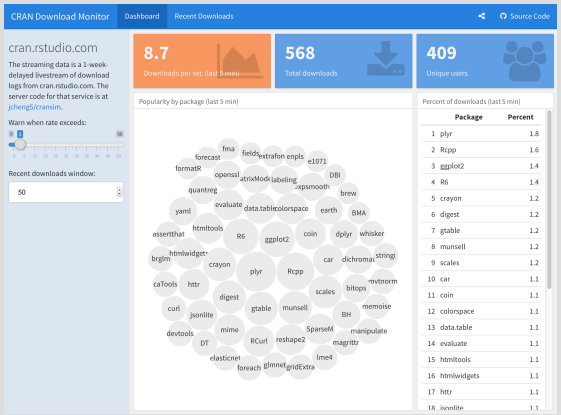
AGENDA

- Possibilities
- Basic Components
  - UI
    - Header
    - Sidebar
    - Body
  - Server
    - Model
- Example
- Resources
- Wrapup

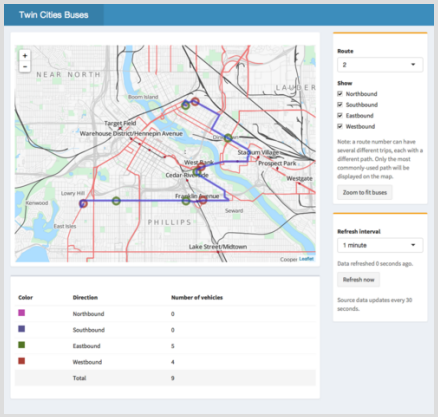
RStudio®

POSSIBILITIES

A DYNAMIC WORD CLOUD DASHBOARD



A DASHBOARD WITH THE LEAFLET PACKAGE



RStudio®

THE BASICS

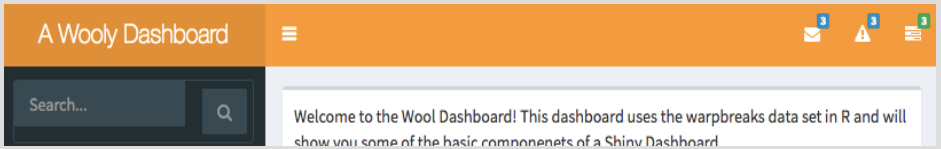
- Same UI/Server format as Shiny applications
- UI calls specific dashboard components
- Server will contain your same R model, plus a few shiny reactive components
- Start with this (or another) shiny dashboard template

```
1  library(shiny)
2  library(shinydashboard)
3
4  ui <- dashboardPage(
5    dashboardHeader(),
6    dashboardSidebar(),
7    dashboardBody()
8  )
9
10 server <- function(input, output) { }
11
12 shinyApp(ui, server)
```

RStudio®

DASHBOARD: HEADER

- Message menu
- Notifications Menu
- Task Menu
- Disable option
  - dashboardHeader(disable = TRUE)

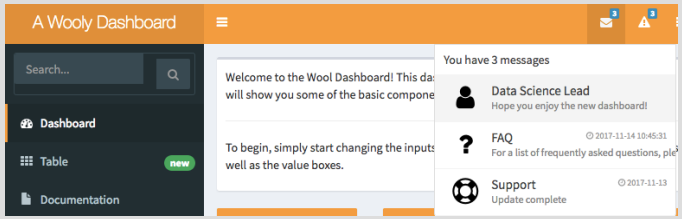


RStudio®

DASHBOARD: HEADER

```
dropdownMenu(type = "messages",
             messageItem(
               from = "Data Science Lead",
               message = "Hope you enjoy the new dashboard!"
             ),

             messageItem(
               from = "FAQ",
               message = "For a list of frequently asked questions, ple
               icon = icon("question"),
               time = Sys
             ),

             messageItem(
               from = "Support",
               message = "Update complete",
               icon = icon("life-ring"),
               time = "2017-11-13"
             )),
```

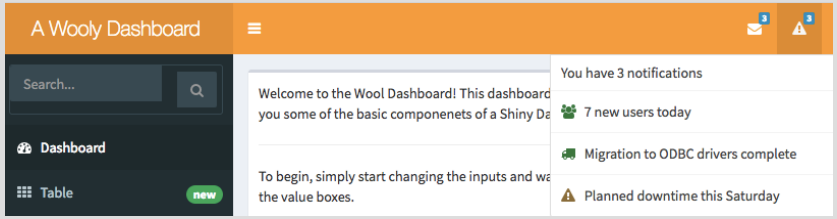
```
messageItem(from, message, icon = shiny::icon("user"), time
href = NULL)
```



RStudio®

DASHBOARD HEADER

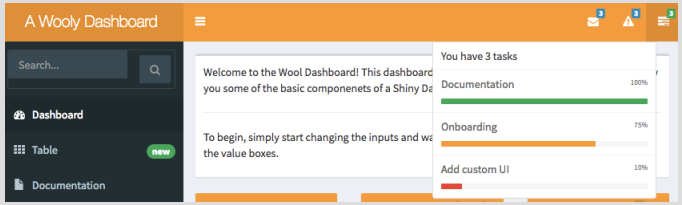
```
dropdownMenu(type = "notifications",

             notificationItem(
               text = "7 new users today",
               icon("users")
             ),

             notificationItem(
               text = "Migration to ODBC drivers complet
               icon ("truck"),
               status = "success"
             ),

             notificationItem(
               text = "Planned downtime this Saturday",
               icon("exclamation-triangle"),
               status = "warning"
             )
```



RStudio®

DASHBOARD HEADER

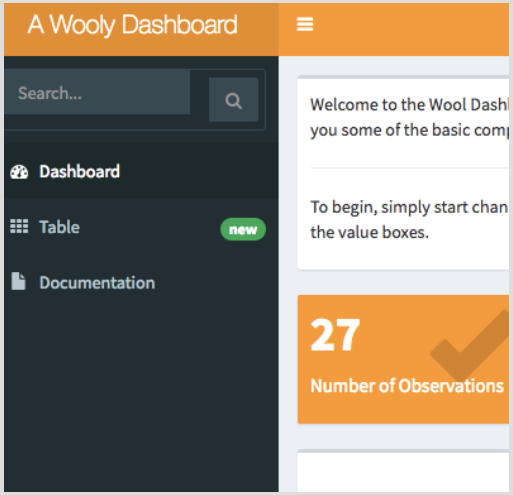
```
dropdownMenu(type = "tasks", badgeStatus = "success",
             taskItem(value = 100, color = "green",
                      "Documentation"
             ),

             taskItem(value = 75, color = "yellow",
                      "Onboarding"
             ),
             taskItem(value = 10, color = "red",
                      "Add custom UI"
             )
```



RStudio®

DASHBOARD: SIDEBAR

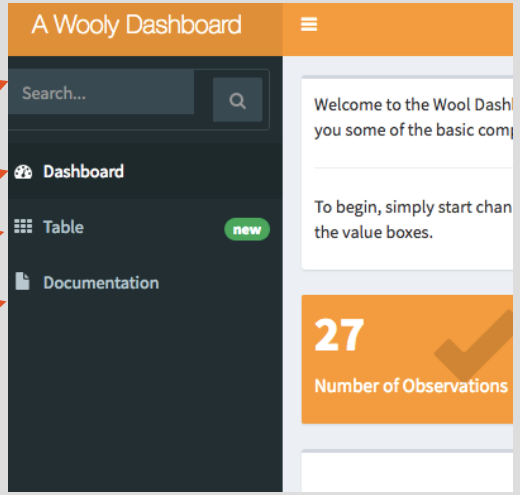
- Can contain:
  - Inputs
  - Dashboards
- Fully customizable appearance
- Possibilities are endless!



RStudio®

DASHBOARD: SIDEBAR

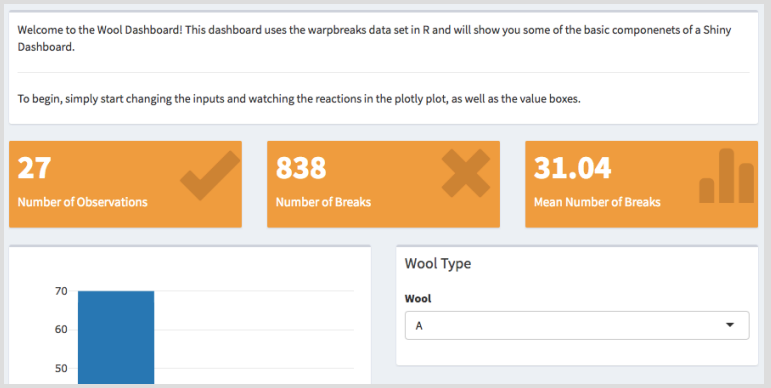
```
dashboardSidebar(
  sidebarMenu(

    #Add in search bar
    sidebarSearchForm(textId = "searchText", buttonId = "searchButton",
                      label = "Search..."),

    #Add in 'Dashboard' tab
    menuItem("Dashboard", tabName = "dashboard", icon = icon("dashboard")),

    #Add in 'Table' tab
    menuItem("Table", icon = icon("th"), tabName = "table",
             badgeLabel = "new", badgeColor = "green"),

    #Add in 'Documentation tab' with link to Shiny Dashboard site
    menuItem("Documentation", icon = icon("file", lib= "glyphicon"),
             href = "https://rstudio.github.io/shinydashboard/index.html")
  )
),
```



RStudio®

DASHBOARD: BODY

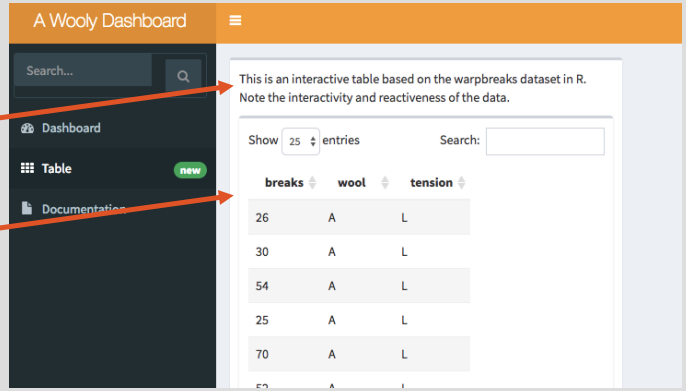
```
tabItem(tabName = "dashboard",
        fluidRow(
          box(
            tags$div(class = "header", checked = NA,
                     tags$p("Welcome to the Wool Dashboard!
                            This dashboard uses the warpbreaks data set
                            tags$hr(),
                            "To begin, simply start changing the inputs a
          ),

          valueBox(27, "Number of Observations", icon = icon("ok", lib =

          valueBoxOutput("breakbox"),

          valueBoxOutput("meanbreakbox"),

          box(plotlyOutput("plot1")),

          box(
            title = "Wool Type",
            selectInput("wool",
                        "Wool",
                        choices = levels(warpbreaks$wool))
          )
        )
        ),
```



RStudio®

DASHBOARD: BODY

```
#Create body for 'Table' tab
tabItem(tabName = "table",
        fluidRow(
          box(
            tags$div(class = "header", checked = NA,
                     tags$p("This is an interactive table based on
          ),

          box(
            dataTableOutput("table1"), width = 16
          )
          )
        )
      )
```

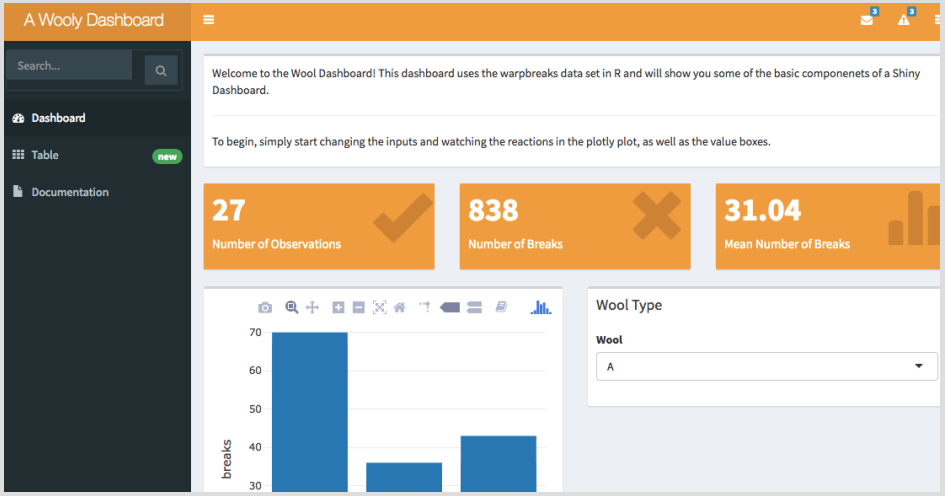


RStudio®

MODEL

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

  #Reactive output for 'Dashboard'
  output$plot1 <- renderPlotly(
    plot_ly(subset(warpbreaks, wool==input$wool), x= ~tension, y= ~breaks)
  )

  woolsum <- reactive(subset(warpbreaks, wool == input$wool))
  woolsubsum <- reactive(sum(woolsum()$breaks))

  output$breakbox <- renderValueBox(
    valueBox(
      woolsubsum(), "Number of Breaks", icon = icon("remove", lib = "glyphicon"),
      color = "yellow"
      )
    )

  output$meanbreakbox <- renderValueBox(
    valueBox(
      round((woolsubsum()/27), digits = 2), "Mean Number of Breaks", icon = icon("stats", lib= "glyphicon"),
      color = "yellow"
    )
  )


  #Interactive table for 'Table'
  output$table1 <- renderDataTable(warpbreaks)

  }
```

AN EXAMPLE



RStudio®

RESOURCES TO NOTE

- Shiny
  - Gallery: https://shiny.rstudio.com/gallery/
  - Cheatsheet: http://shiny.rstudio.com/images/shiny-cheatsheet.pdf
  - Learning Shiny
    - Shiny references: https://shiny.rstudio.com/articles/
    - Shiny webinar: https://www.rstudio.com/resources/webinars/
- Shinydashboard: https://rstudio.github.io/shinydashboard/
  - Basic tutorial: https://rstudio.github.io/shinydashboard/get_started.html
  - Advanced tutorial: https://rstudio.github.io/shinydashboard/structure.html
  - CSS tags: https://shiny.rstudio.com/articles/tag-glossary.html
  - Glyphicons: https://getbootstrap.com/docs/3.3/components/
- Advanced Shiny
  - Bookmarkeable State: https://shiny.rstudio.com/articles/bookmarking-state.html
  - Custom Appearance (HTML required): https://shiny.rstudio.com/articles/html-tags.html
  - Custom Appearance (CSS required): https://shiny.rstudio.com/articles/css.html
  - Shiny Gadgets: https://shiny.rstudio.com/articles/gadgets.html.
  - Linked Brushing: https://rstudio.github.io/crosstalk/shiny.html
  - Modularizing shiny code: https://shiny.rstudio.com/articles/modules.html
  - Profiling and Performance: https://shiny.rstudio.com/articles/profiling.html
  - Shiny Datatables: https://shiny.rstudio.com/articles/datatables.html

(MORE) RESOURCES TO NOTE

- Deployment
  - Connect: https://www.rstudio.com/products/connect/
  - .Shiny apps.io: http://www.shinyapps.io/
  - Shiny Server (Open Source or Professional): https://www.rstudio.com/products/shiny/shiny-server/
- Support
  - RStudio Community: https://community.rstudio.com/
  - Stackoverflow: https://stackoverflow.com/

ANY QUESTIONS?