

Interactive Dashboards

HES 505 Fall 2024: Session 28

Carolyn Koehn

Objectives

- List the necessary elements of an interactive dashboard
- Outline the structure of code needed to build a **flexdashboard**
- Build a simple interactive dashboard with spatial data

What is an interactive dashboard?

- Reactive to user inputs
- Examples

What do we need?

1. Proper YAML header
2. **global** code chunk to load libraries and data
3. **Shiny** inputs and outputs
4. Render results

YAML header

- Output is now a flexdashboard (instead of html)
- **runtime:: shiny** allows R Shiny to handle interactivity

```
1 ---  
2 title: "Climate, social, and environmental justice markers for the Pacific  
3 output: flexdashboard::flex_dashboard  
4 runtime: shiny  
5 ---
```

global code

Code that only needs to run once

```
1  ```{r global}
2  # include: false
3
4  library(shiny)
5  library(sf)
6  library(tidyverse)
7  library(tmap)
8  tmap_mode("view")
9
10 cejst <- st_read("/opt/data/data/assignment01/cejst_nw.shp")
11
12 # get column codes and meanings
13 col_choices <- read_csv("/opt/data/data/assignment04/columns.csv") %>%
14   # make nicer column names for a display table
15   rename("Code" = "shapefile_column", "Description" = "column_name") %>%
16   # keep only "percentile" type columns
17   filter(str_detect(Code, "PFS"))
18 ````
```

Inputs

- Live in a **sidebar**

```
1 Column { .sidebar}  
2 -----
```

Types of Inputs

R Function	Input Type
<code>selectInput</code>	A box with choices to select from
<code>sliderInput</code>	A slider bar
<code>radioButtons</code>	A set of radio buttons
<code>textInput</code>	A field to enter text
<code>numericInput</code>	A field to enter numbers
<code>checkboxInput</code>	A single check box
<code>dateInput</code>	A calendar to aid date selection
<code>dateRangeInput</code>	A pair of calendars for selecting a date range
<code>fileInput</code>	A file upload control wizard

Adding Inputs

```
1 # Box with choices: which cejst column to map
2 selectInput("column_select", label = "Justice Marker:",
3             choices = col_choices$Code, selected = "DF_PFS")
4
5 # Two sliders to select the maximum and minimum values to map
6 sliderInput("min_threshold_adjust", label = "Minimum value:",
7             min = 0, max = 1, value = 0.5, step = 0.05)
8 sliderInput("max_threshold_adjust", label = "Maximum value:",
9             min = 0, max = 1, value = 1, step = 0.05)
```

Adding Outputs

Create a new column with title

```
1 Column  
2 -----  
3  
4 ### Climate, Social, and Environmental Justice
```

Types of Outputs

R Function	Output Type
renderPlot	R graphics output
renderPrint	R printed output
renderTable	Data frame, matrix, other table like structures
renderText	Character vectors

Adding Reactive Output

- Specify reactive elements with `input$NameOfInput`
- In this example, we use reactive filtering to only map **cejst** tracts that meet user criteria

```
1 # renderTmap is a tmap special case of renderPlot
2 renderTmap({
3   # put reactively filtered data in tm_shape
4   tm_shape(subset(cejst[, input$column_select], # subset data to user's col
5                 # use the subset in the filtering steps, selecting the co
6                 cejst[, input$column_select][[1]] <= input$max_threshold_
7                 cejst[, input$column_select][[1]] >= input$min_threshold_
8   # add the polygons filled by the user's selected column
9   tm_polygons(col = input$column_select)
10 })
```

Add text/explanation to sidebar

- Add plain text before or after code chunks
- I've also added a nice table with the column name meanings for reference

```
1 knitr:::kable(col_choices[,1:2])
```

Publishing

- On shinyapps.io (free; [tutorial here](#))
- On GitHub Pages (limited free pages; [tutorial here](#))
- Contact Research Computing for options