

# Intro to Shiny Practical 2

Note: View this practical sheet online (here [https://mcguinlu.github.io/slides/data\\_viz\\_short\\_course\\_shiny/pracs/practical\\_2.html](https://mcguinlu.github.io/slides/data_viz_short_course_shiny/pracs/practical_2.html)) to access the hints and tips.

## Exercise 3: Customising the UI

1. Open the `prac_3.R` file in the `shiny-practicals` folder.
2. Run the app to see what it does
3. Add some text to the end of the sidebar to describe the dataset used

► Hint

4. Add some text to the `mainPanel()` e.g. between the plot and the table
5. Re-run your app to ensure that it works as expected

---

## Exercise 4: The `reactive({})` function

In the previous session, we described an example where the data was being subset within the function that created the `barPlot` output object:

```
server <- function(input, output) {  
  
  output$barPlot <- renderPlot({  
    data <- head(med, n = input$slider)  
  
    ggplot(data = data, aes_string(fill = input$fill)) +  
    geom_histogram(aes(x = status), stat = "count")  
  })  
}
```

We converted this to the following code, using the `reactive({})` function, to prevent the data from being re-subset every time the user changed the value of `input$fill`:

```
server <- function(input, output) {  
  data <- reactive({  
    head(med, n = input$slider)  
  })  
  
  output$barPlot <- renderPlot({  
    ggplot(data = data(), aes_string(fill = input$fill)) +  
    geom_histogram(aes(x = status), stat = "count")  
  })  
}
```

Can you apply the same principle to the app in `prac_3.R`?

1. In the the `prac_3.R` , create a reactive expression so that the data does not have to be subset every time the title of the plot is changed.
- 

## Exercise 5: Build an app from scratch

1. Open a new Shiny app file in RStudio (File -> New File -> Shiny Web App. . .)
2. Use one of the layouts we discussed as part of this workshop.

► Hint

3. Load `shiny` and `ggplot2` packages

4. Load the `med` data set

```
med <- read.csv("http://bit.ly/bris-data-viz-med")
```

5. Add some text to the app to describe the dataset used

► Hint

6. Add at least one widget to the `sidebarPanel` (but feel free to add as many as you like!)

7. Add some reactive content (e.g. a plot or a table) to the `mainPanel` that makes use of the input from your widget

8. Run the app to ensure that it works as expected.
- 

## Exercise 5: Publish your app [Optional]

1. Ensure that your app.R file is contained within its own folder

2. Deploy your app online

- Go to shinyapps.io (<https://www.shinyapps.io/>)
- Follow the instructions there to publish your app