

# Intro to Shiny - Practical 2

Luke A McGuinness - 15 November 2019

---

```
#####
# LOAD EXTERNAL PACKAGES, SCRIPTS AND DATA #
#####
library(shiny)
library(ggplot2)
library(BristolVis)
#####
# USER INTERFACE #
#####
ui <- fluidPage(
  titlePanel(title = "Demo of a shiny app"),

  # Define sidebar layout
  sidebarLayout(
    sidebarPanel(
      # Define number of observations to plot in the figure
      sliderInput(inputId = "numberofrowsplot",
                  label = "Number of rows to plot in figure",
                  value = 100,
                  min = 50,
                  max = 150),

      # Define number of observations to show in the table
      numericInput(inputId = "numberofrowstable",
                  label = "Number of rows to show in table",
                  value = 10,
                  min = 5,
                  max = 20,
                  step = 5),

      # Define the variable that is used to colour the points
      selectInput(inputId = "pointcolour",
                  label = "Variable to fill by:",
                  choices = c("sex", "diet", "status")),

      # Define the plot's title
      textInput(inputId = "titletext",
                label = "Plot title:")
    ),

    mainPanel(
      plotOutput("barPlot")
    )
  )
)
#####
# SERVER #
#####
server <- function(input, output) {
```

```

# Create plot object
output$barPlot <- renderPlot({

  # Restrict the bmi dataset to the number of rows defined by the slider
  bmi2_plot <- head(x = bmi2,
                    n = input$numberofrowsplot)

  # Create plot using the restricted dataset
  ggplot(data = bmi2_plot, aes_string(color = "diet")) +
    geom_point(aes(x = age, y= bmi)) +
    labs(title = "Plot title")

})

# Create table object
output$table <- renderTable({
  head(x = bmi2,
        n = input$numberofrowstable)
})

}
#####
# CALL TO shinyApp FUNCTION #
#####
shinyApp(ui = ui, server = server)

```