

Intro to Shiny Practical 1

Note: View this practical sheet online (here

(https://mcguinlu.github.io/slides/data_viz_short_course_shiny/pracs/practical_1.html)) to access the hints and tips.

Install and load packages

```
install.packages(c("shiny", "ggplot2"))
```

```
library(shiny)
library(ggplot2)
```

Warm-up: Run your first app

```
library(shiny)

ui <- fluidPage(
  # Allow user to define two numbers
  numericInput(inputId = "add_1", label = "Number:", value = 0),
  numericInput(inputId = "add_2", label = "Number:", value = 0),
  # Display the output
  textOutput(outputId = "sum")
)
server <- function(input, output) {
  output$sum <- renderText({
    input$add_1 + input$add_2
  })
}
shinyApp(ui = ui, server = server)
```

1. Copy the code above to an empty R file.
2. Make sure you can run the app, both:
 - using CTRL+SHIFT+ENTER keyboard shortcut
 - from the console (note: you'll have to save it first)

Exercise 1: Design a user interface

1. Open the `prac_1.R` file in the `shiny-practicals` folder.
2. Identify the opening and closing brackets of the elements of `sidebarLayout` (e.g. `sidebarLayout()`, `sidePanel()`, `mainPanel()`)
3. Run the app to familiarise yourself with it

4. Make the following changes:

- Change the title of the app to “Demo of a shiny app”
- Add a new `selectInput()` widget to the sidebar, with an `inputId` of “select”, a label of “Choose one:”, and choices of “sex”, “diet”, and “status”
- Add a new `textInput()` widget to the sidebar with an `inputId` of “plottitle” and a label “Plot title:”

► Hint

5. Re-run the app to ensure that the changes have worked

Exercise 2: Create reactive content

1. Open the `prac_2.R` file in the `shiny-practicals` folder.

2. Run the app to familiarise yourself with it

3. Examine the code used to produce the table in the UI/server

- Why is the table not being displayed in the app?
- Fix the code in the UI so that the table is displayed in the `mainPanel`

► Hint

4. Examine the code used to produce the plot in the UI/server

- Update the code so that the colour is set by the value of `input$pointcolour`
- Update the code so that the title of the plot is defined by the value of the input from the `textInput()` widget

► Hint

5. Re-run the app to ensure that the changes have worked