A simple shiny app to explore Palmer Penguin data using ChatGPT to prototype.

Ronald (Ryy) Glenn Thomas 2025-01-15

Table of contents

1	Introduction		1
	1 2	R code to launch	9

1 Introduction

Shiny apps are a great way to explore data interactively. In this example, we will use ChatGPT to prototype a simple Shiny app to explore the Palmer Penguin data set from the palmerpenguins package.

1.1

chatGPT Prompts:

"I want to use the Palmer Penguin dataset to create a Shiny app for data exploration."

"Update shiny app. Add a dropdown menu to select categorical variables, sex, species or island. Also add a dropdown menu to select continuouse variables Use selected categorical variable as

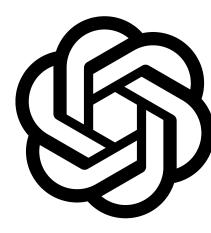


Figure 1: chatGPT

a grouping variable in side-by-side boxplots of selected continuouse variables."

"add a second interactive plot to app.R code to provide scatterplots of 2 selected continuous variables. The two cont. vars are selected from drop down menus."

1.2 R code to launch.

From inside R.

```
shiny::runApp("app.R", launch.browser = TRUE)
```

From shell.

```
R -e "shiny::runApp('app.R', launch.browser=T)"
```

```
library(shiny)
library(palmerpenguins)
library(ggplot2)
# Load the dataset
data <- na.omit(penguins) # Remove rows with NA for simplicity
ui <- fluidPage(</pre>
  titlePanel("Palmer Penguins Explorer"),
  sidebarLayout(
    sidebarPanel(
      selectInput("xvar", "Continuous Variable for Boxplot:", choices = names(data)[3:6]),
      selectInput("groupvar", "Group by (Categorical Variable):",
                  choices = c("species", "sex", "island"), selected = "species"),
      selectInput("scatter_x", "X-axis for Scatterplot:", choices = names(data)[3:6]),
      selectInput("scatter_y", "Y-axis for Scatterplot:", choices = names(data)[3:6]),
      selectInput("groupvar_scatter", "Group by (Scatterplot):",
                  choices = c("species", "sex", "island"), selected = "species")
    ),
    mainPanel(
      plotOutput("boxPlot"),
```

```
plotOutput("scatterPlot"),
      verbatimTextOutput("summary")
    )
 )
)
server <- function(input, output) {</pre>
  # Reactive dataset (no filtering applied anymore)
 filteredData <- reactive({</pre>
    data
 })
  # Boxplot
  output$boxPlot <- renderPlot({</pre>
    req(filteredData()) # Ensure data is available
    ggplot(filteredData(), aes_string(x = input$groupvar, y = input$xvar, fill = input$groupvar
      geom_boxplot(alpha = 0.7) +
      theme_minimal() +
      labs(x = input$groupvar, y = input$xvar, fill = input$groupvar) +
      theme(legend.position = "bottom")
 })
  # Scatterplot with grouping
  output$scatterPlot <- renderPlot({</pre>
    req(filteredData()) # Ensure data is available
    ggplot(filteredData(), aes_string(x = input$scatter_x, y = input$scatter_y, color = input$s
      geom_point(alpha = 0.7, size = 2) +
      theme_minimal() +
      labs(x = input$scatter_x, y = input$scatter_y, color = input$groupvar_scatter) +
      theme(legend.position = "bottom")
 })
  # Summary of the dataset
  output$summary <- renderPrint({</pre>
    req(filteredData()) # Ensure data is available
    summary(filteredData())
 })
}
shinyApp(ui, server)
```