R Shiny Demo

Kevin Mastascusa

What is “Shiny”?

- **Shiny** is an R package that allows you to build interactive web applications directly from R.
- It combines the power of R’s data analysis with the interactivity of web applications, making data visualizations more engaging and dynamic.

Structure of a Shiny App

1. UI (User Interface)

- **Purpose**: Defines the layout and appearance of the app.

Components:

- **fluidPage()**: Creates a flexible layout that adjusts to different screen sizes.
- **titlePanel()**: Adds a title to the app.
- **sidebarLayout()**: Divides the UI into a sidebar and main content area.
- **sidebarPanel()**: Contains input controls like sliders, dropdowns, checkboxes.
- **mainPanel()**: Displays the output, such as plots, tables, or text.

- **Example:**

UI Example

```
ui <- fluidPage(
    titlePanel("My Shiny App"),
    sidebarLayout(
        sidebarPanel(
            selectInput("variable", "Choose a variable:", choices = names(diamonds
        ),
        mainPanel(
            plotOutput("myPlot")
        )
    )
)
```

Structure of a Shiny App

2. Server Logic

Purpose: Contains the instructions that tell Shiny how to respond to user inputs and generate outputs.

Components:

renderPlot(), renderTable(), renderText(): Functions that generate the outputs.

Reactivity: Automatically updates outputs when inputs change.

How it works: The server function processes user inputs, performs calculations, and updates the outputs accordingly.

Example:

Server Logic Example

```
server <- function(input, output) {
    output$myPlot <- renderPlot({
        ggplot(diamonds, aes_string(x = input$variable)) +
            geom_histogram()
    })
}
```

```
shinyApp(ui = ui, server = server)
```

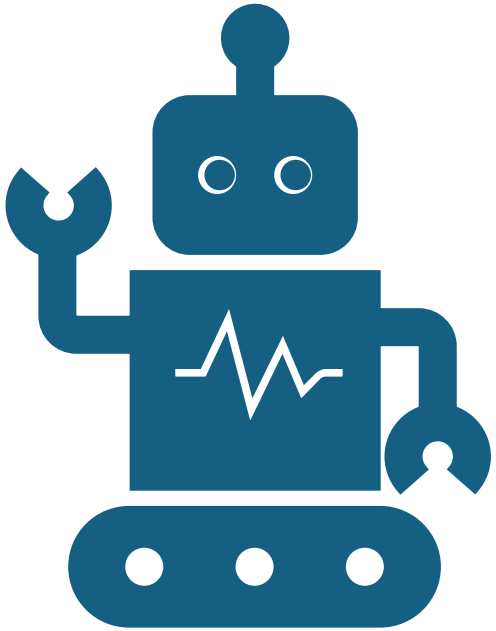
Putting It All Together

3. Running the App

• **Purpose**: Ties the UI and server together into a functional app.

• **shinyApp()**: This function launches the Shiny app by combining the UI and server components.

Example:



Recap and Example

Recap of Shiny Components

- **UI**: Controls the layout and inputs of the app.
- **Server**: Contains the logic and reactivity of the app.
- **Running the App**: Combines UI and server to create the app.

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

ui ← fluidPage(
    titlePanel("Diamonds Visualization"),
    sidebarLayout(
        sidebarPanel(
            selectInput("variable", "Choose a variable:", choices = names(diamonds)),
            selectInput("plotType", "Choose plot type:", choices = c("Histogram", "Boxplot"))
        ),
        mainPanel(
            plotOutput("myPlot")
        )
    )
)

server ← function(input, output) {
    output$myPlot ← renderPlot({
        if (input$plotType == "Histogram") {
            ggplot(diamonds, aes_string(x = input$variable)) + geom_histogram()
        } else {
            ggplot(diamonds, aes_string(x = "cut", y = input$variable)) + geom_boxplot()
        }
    })
}

shinyApp(ui = ui, server = server)
```

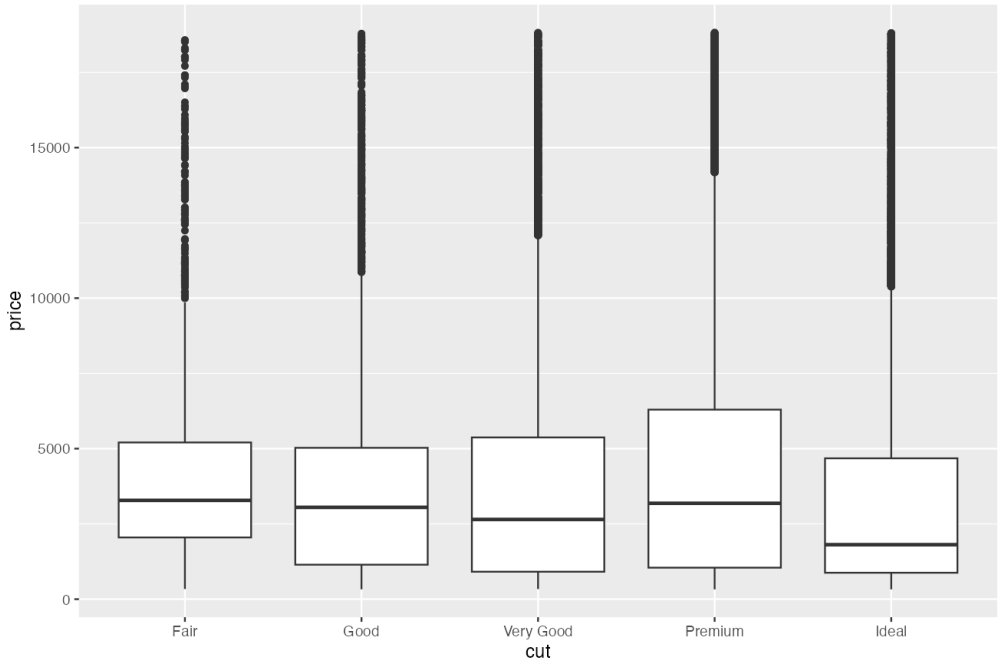
Diamonds Visualization

Choose a variable:

price

Choose plot type:

Boxplot

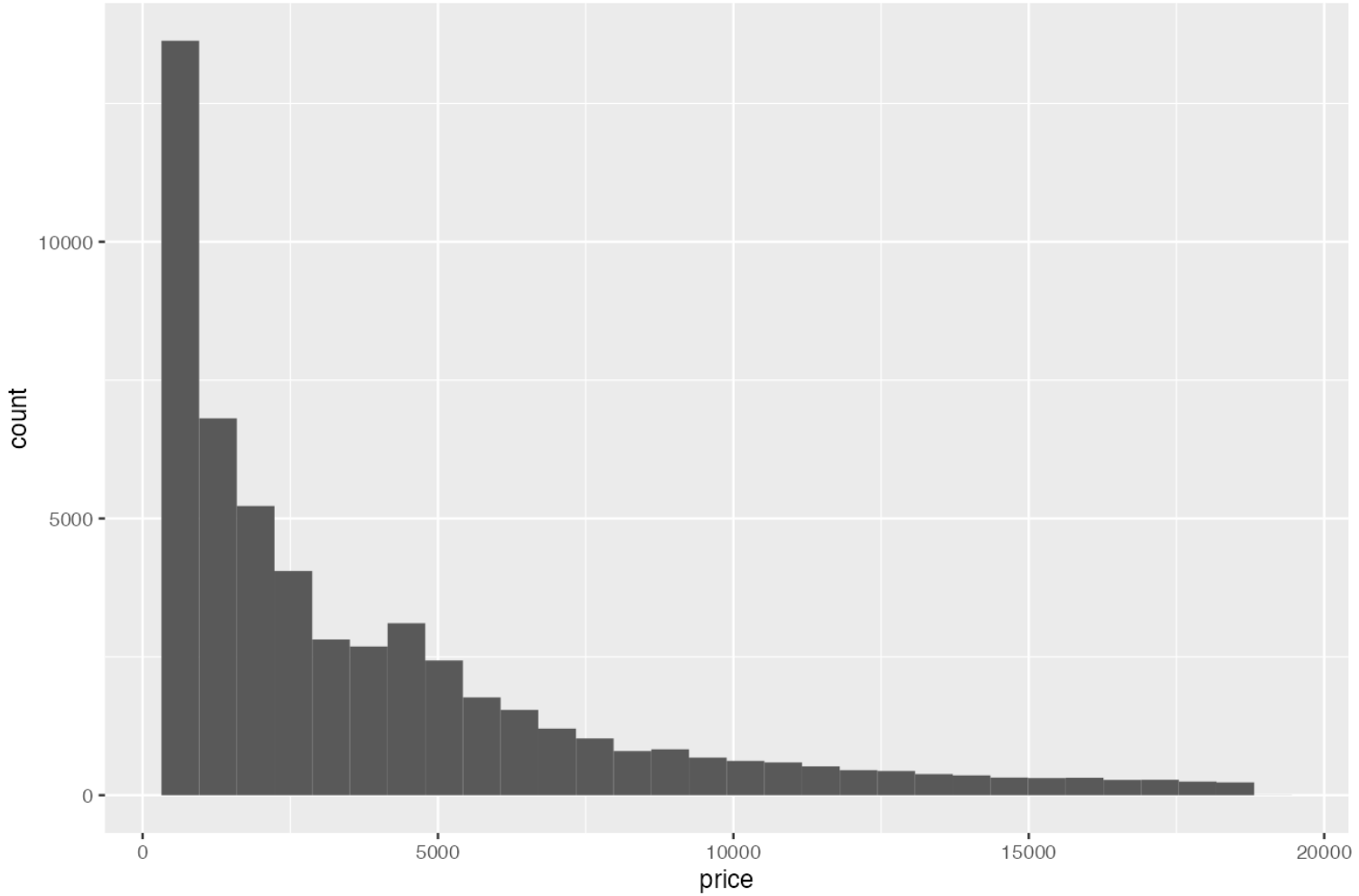


Choose a variable:

price

Choose plot type:

Histogram



http://127.0.0.1:5883 | Open in Browser | Publish

Diamonds Dataset Visualization

X-axis Variable:

carat

Y-axis Variable:

price

Plot Type:

Histogram

☐ Logarithmic Scale: