

Shiny code 04_mpg code

```
library(shiny)
library(datasets)

# Data pre-processing ----
# Tweak the "am" variable to have nicer factor labels -- since this
# doesn't rely on any user inputs, we can do this once at startup
# and then use the value throughout the lifetime of the app
mpgData <- mtcars
mpgData$am <- factor(mpgData$am, labels = c("Automatic", "Manual"))

# Define UI for miles per gallon app ----
ui <- fluidPage(

  # App title ----
  titlePanel("Miles Per Gallon"),

  # Sidebar layout with input and output definitions ----
  sidebarLayout(

    # Sidebar panel for inputs ----
    sidebarPanel(

      # Input: Selector for variable to plot against mpg ----
      selectInput("variable", "Variable:",
                  c("Cylinders" = "cyl",
                    "Transmission" = "am",
                    "Gears" = "gear")),

      # Input: Checkbox for whether outliers should be included ----
      checkboxInput("outliers", "Show outliers", TRUE)

    ),
```

```

# Main panel for displaying outputs ----
mainPanel(

  # Output: Formatted text for caption ----
  h3(textOutput("caption")),

  # Output: Plot of the requested variable against mpg ----
  plotOutput("mpgPlot")

)
)
)

# Define server logic to plot various variables against mpg ----
server <- function(input, output) {

  # Compute the formula text ----
  # This is in a reactive expression since it is shared by the
  # output$caption and output$mpgPlot functions
  formulaText <- reactive({
    paste("mpg ~", input$variable)
  })

  # Return the formula text for printing as a caption ----
  output$caption <- renderText({
    formulaText()
  })

  # Generate a plot of the requested variable against mpg ----
  # and only exclude outliers if requested
  output$mpgPlot <- renderPlot({
    boxplot(as.formula(formulaText()),
            data = mpgData,
            outline = input$outliers,
            col = "#75AADB", pch = 19)
  })

}

# Create Shiny app ----
shinyApp(ui, server)

```