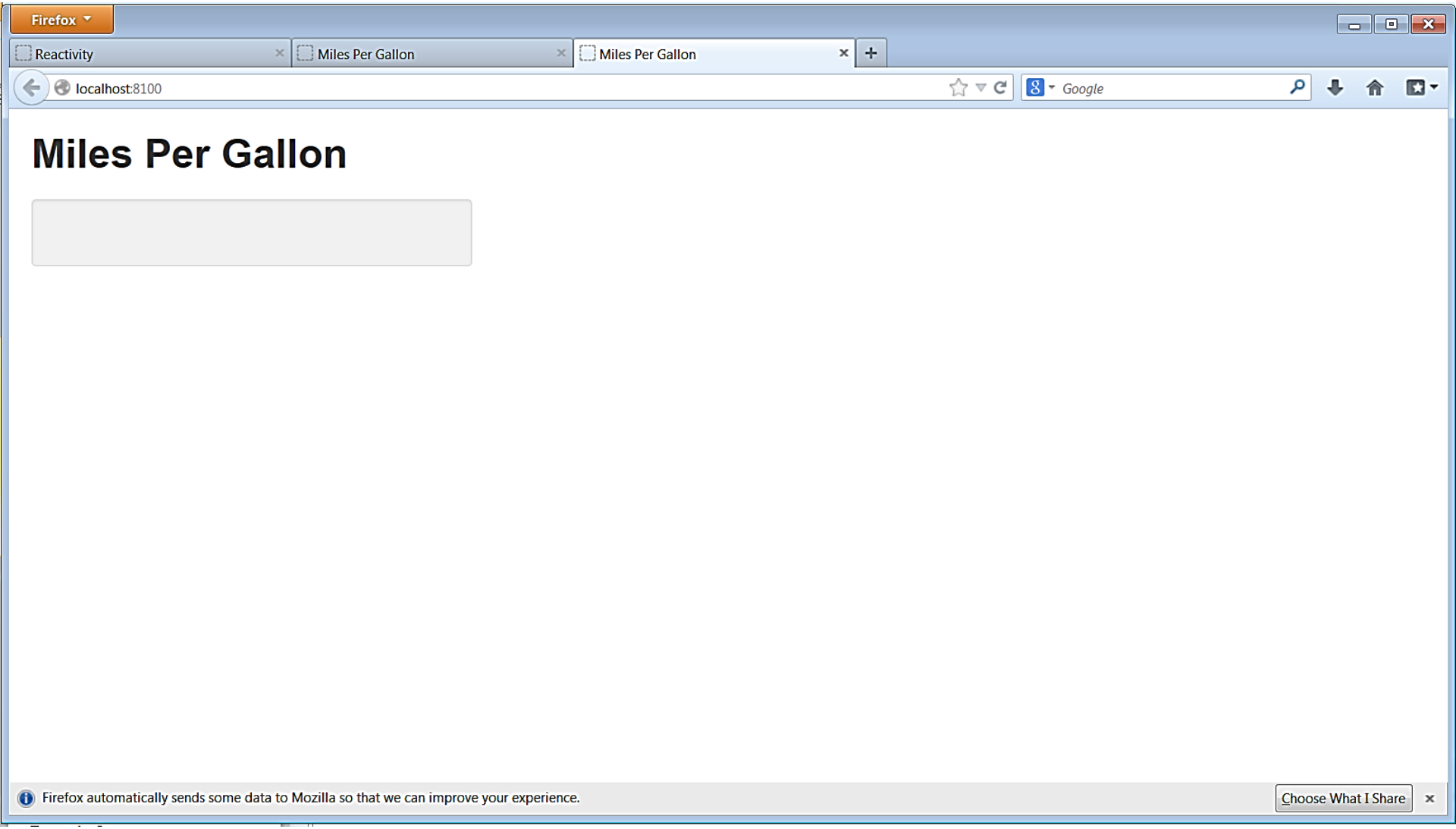
# Building A Shiny Application

# ~/shinyapps Example 001



# ~/shinyapps Example 001 ui.R

**## 001 Live Example ui.R**

**Note that ui.R source file defines the 'look and feel' of the user interface**

```
library(shiny)
```

Just to make sure that the Shiny package Is loaded when the ui.R file is "sourced" or called

```
# Define UI for miles per gallon application
shinyUI(pageWithSidebar(
```

Must call **shinyUI()** function in ui.R Source file which calls all other functions

Defining page as **pageWithSidebar()**

```
  # Application title
  headerPanel("Miles Per Gallon"),
```

Declaring **headerPanel()** shows "Miles Per Gallon"

```
  sidebarPanel(),
```

Declaring will be a **sidebarPanel()** for input, is empty now

```
  mainPanel()
))
```

Where output will display to the right, is empty now

**Three functions headerPanel(), sidebarPanel() and mainPanel define regions of user interface**

# ~/shinyapps Example 001 server.R

**Note that server.R source file defines the logic and functionality of the app**

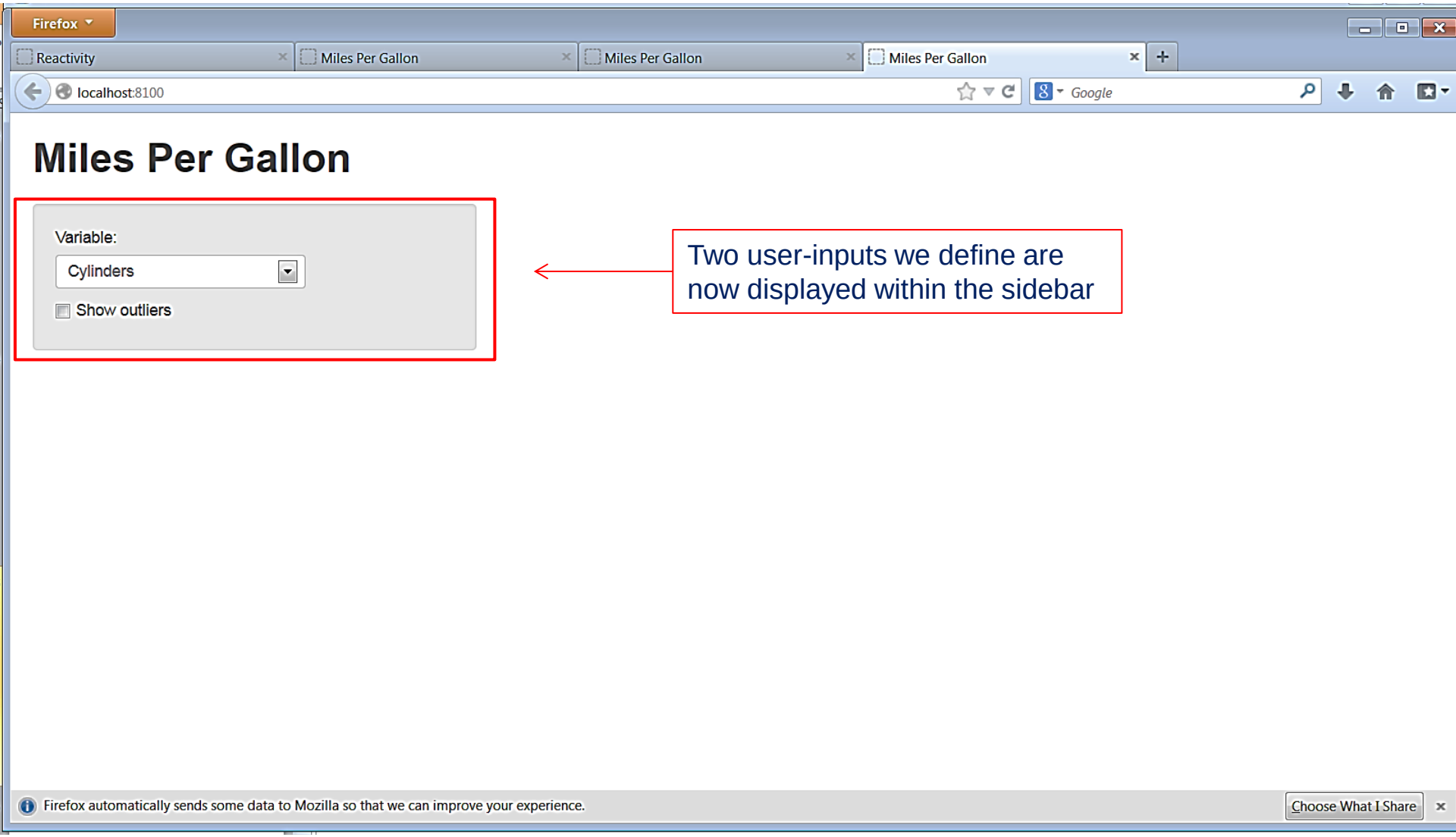
```
## 001 ~/shinyapps Live Example server.R

library(shiny)

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

})
```

Just to make sure that the Shiny package
Is loaded when the server.R file is “sourced” or called

Must call **shinyServer()** function in server.R source file which calls all other functions

**Server function is empty now but we will use it to define relationships between inputs and outputs**

# ~/shinyapps Example 002



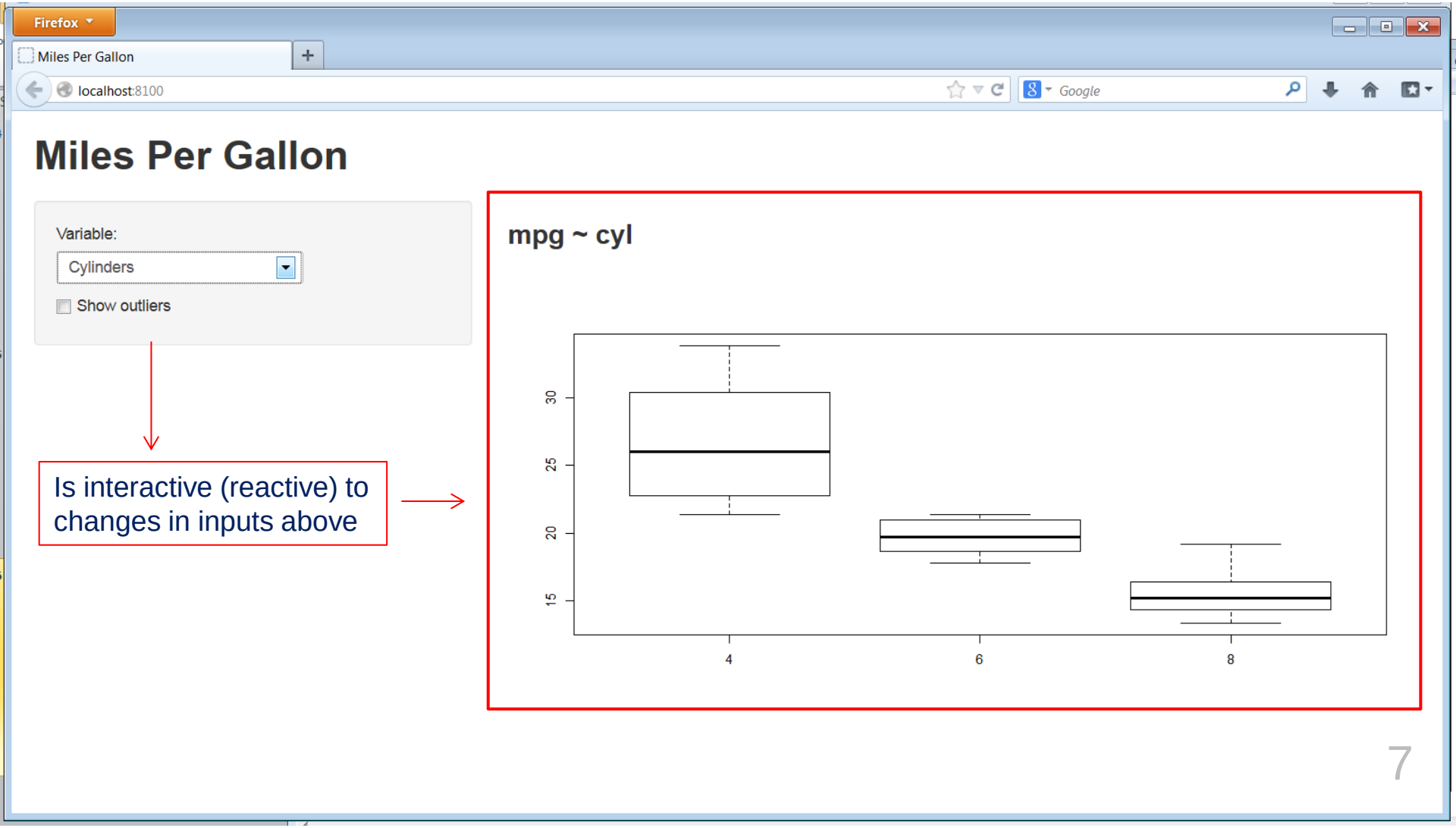
# ~/shinyapps Example 002 ui.R

```
## 002 ~/shinyapps Live Example mtcars ui.R
## Added Two User Inputs Displayed
## Within the Sidebar:

library(shiny)

# Define UI for miles per gallon application
shinyUI(pageWithSidebar(

  # Application title
  headerPanel("Miles Per Gallon"),

  # Sidebar with controls to select the variable
  # to plot against mpg and to specify whether
  # outliers should be included
  sidebarPanel(
    selectInput("variable", "Variable:",
                list("Cylinders" = "cyl",
                     "Transmission" = "am",
                     "Gears" = "gear")),

    checkboxInput("outliers", "Show outliers", FALSE)
  ),

  mainPanel()
))
```

Sidebar with controls to select variable to plot against mpg

Specify whether outliers should be included; default is “no”

# ~/shinyapps Example 003



# ~/shinyapps Example 003 ui.R

```
## 003 ~/shinyapps Live Example mtcars ui.R
library(shiny)

# Define UI for miles per gallon application
shinyUI(pageWithSidebar(

  # Application title
  headerPanel("Miles Per Gallon"),

  # Sidebar with controls to select the variable to plot against mpg
  # and to specify whether outliers should be included
  sidebarPanel(
    selectInput("variable", "Variable:",
                list("Cylinders" = "cyl",
                     "Transmission" = "am",
                     "Gears" = "gear")),

    checkboxInput("outliers", "Show outliers", FALSE)
  ),

  # Show the caption and plot of the requested variable against mpg
  mainPanel(
    h3(textOutput("caption")),

    plotOutput("mpgPlot")
  )
))
```

User interface now displays “caption” and plot of “mpgPlot”

# ~/shinyapps Example 003 server.R

```
## 003 ~/shinyapps Live Example mtcars ui.R
# server.R - Full server script with inline comments
# that explain the implementation techniques in
# greater detail (Example 003):

library(shiny)
library(datasets)

# We tweak the "am" field 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 application
mpgData <- mtcars
mpgData$am <- factor(mpgData$am, labels = c("Automatic", "Manual"))
```

# ~/shinyapps Example 003 server.R

```
## 003 ~/shinyapps Live Example mtcars ui.R (continued from previous slide)
# Define server logic required to plot various variables against mpg
shinyServer(function(input, output) {

  # Compute the formula text in a reactive expression since it is
  # shared by the output$caption and output$mpgPlot expressions
  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
  # include outliers if requested
  output$mpgPlot <- renderPlot({
    boxplot(as.formula(formulaText()),
            data = mpgData,
            outline = input$outliers)
  })
})
```