



BUILDING WEB APPLICATIONS IN R WITH SHINY

# Interface builder functions

# tags

```
> names(tags)
[1] "a"          "abbr"      "address"   "area"      "article"
[6] "aside"      "audio"     "b"         "base"      "bdi"
[11] "bdo"        "blockquote" "body"      "br"        "button"
[16] "canvas"     "caption"   "cite"      "code"      "col"
[21] "colgroup"   "command"   "data"      "datalist"   "dd"
[26] "del"        "details"   "dfn"       "div"        "dl"
[31] "dt"         "em"        "embed"     "eventsource" "fieldset"
[36] "figcaption" "figure"    "footer"    "form"       "h1"
[41] "h2"         "h3"        "h4"        "h5"        "h6"
[46] "head"       "hr"        "html"
[51] "i"          "input"     "ins"
[56] "kbd"        "legend"    "li"
[61] "link"       "mark"      "map"       "menu"      "meta"
[66] "meter"     "nav"       "noscript"  "object"    "ol"
[71] "optgroup"   "option"    "output"    "p"         "param"
[76] "pre"       "progress"  "q"         "ruby"      "rp"
[81] "rt"        "s"         "samp"      "script"    "section"
[86] "select"    "small"     "source"    "span"      "strong"
[91] "style"     "sub"       "summary"   "sup"       "table"
[96] "tbody"     "td"        "textarea"  "tfoot"     "th"
[101] "thead"    "time"      "title"     "tr"        "track"
[106] "u"        "ul"        "var"       "video"     "wbr"
```



`<i> some text </i>`

# tag → HTML

```
> tags$b("This is my first app")  
<b>This is my first app</b>
```

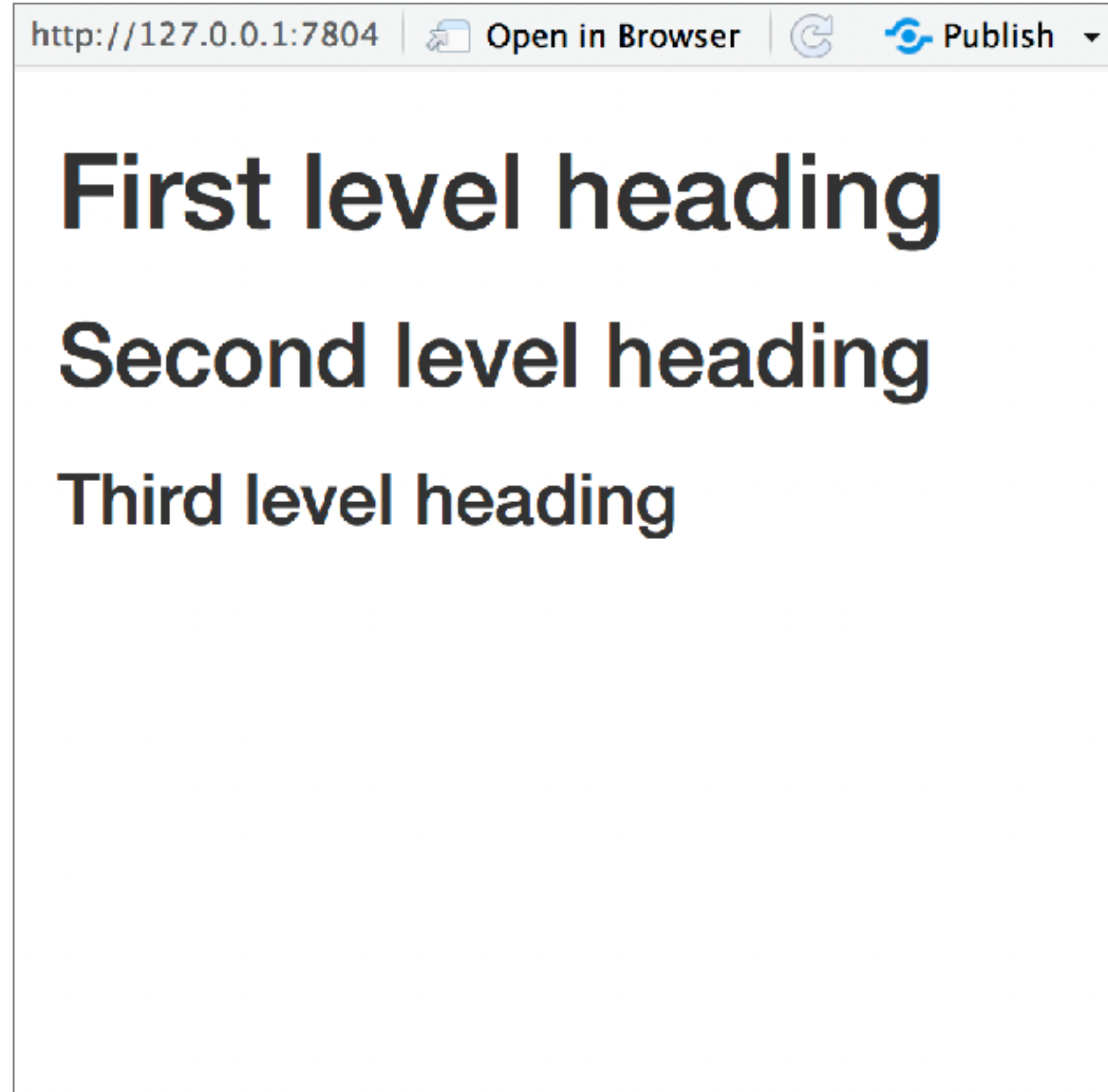
# Header tags

```
library(shiny)

# Define UI with tags
ui <- fluidPage(
  tags$h1("First level heading"),
  tags$h2("Second level heading"),
  tags$h3("Third level heading")
)

# Define server fn that does nothing :)
server <- function(input, output) {}

# Create the app object
shinyApp(ui = ui, server = server)
```



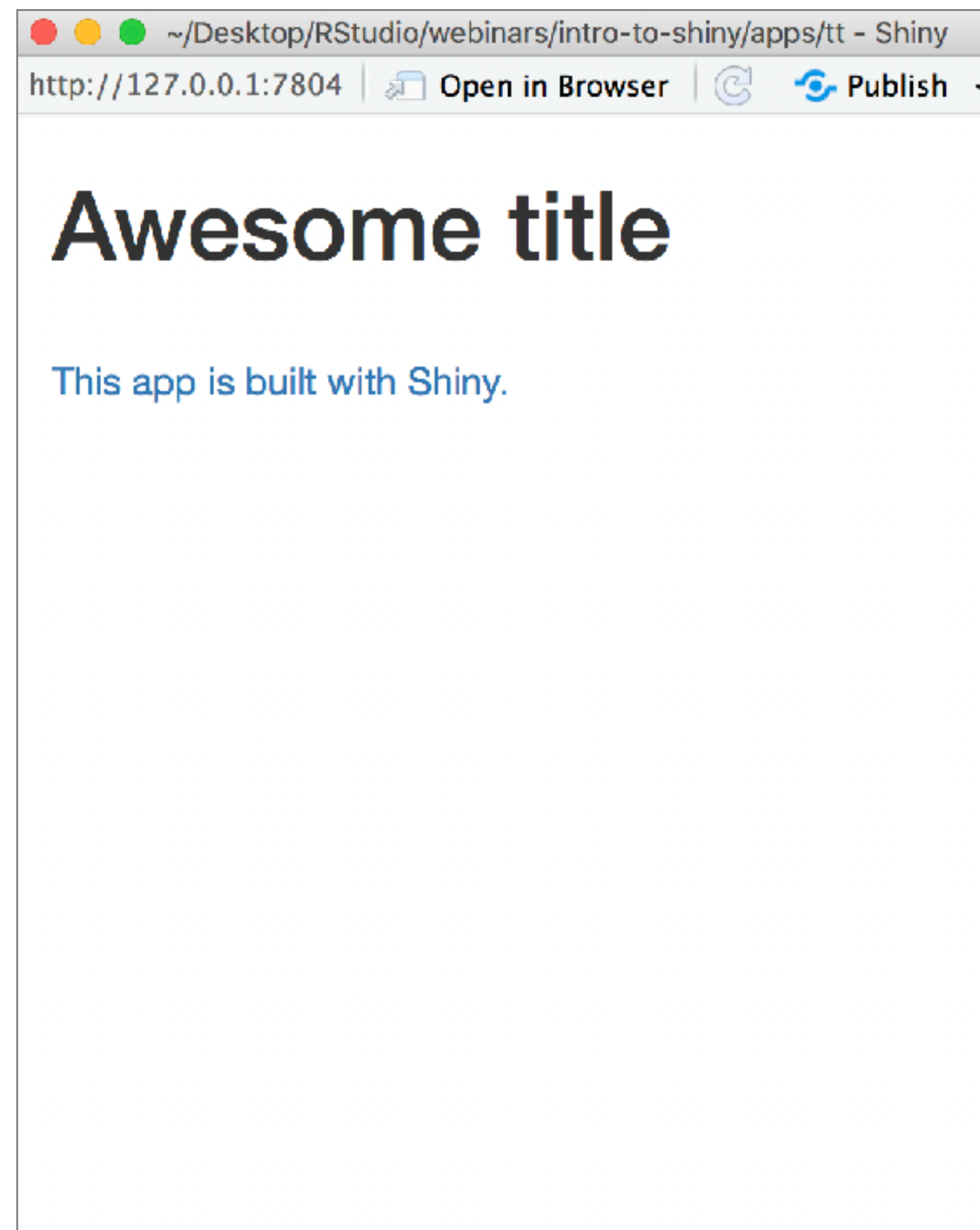
# Linked text

```
library(shiny)

# Define UI with tags
ui <- fluidPage(
  tags$h1("Awesome title"),
  tags$br(), # line break
  tags$a("This app is built with
Shiny.", href = "http://
shiny.rstudio.com/")
)

# Define server fn that does nothing :)
server <- function(input, output) {}

# Create the app object
shinyApp(ui = ui, server = server)
```



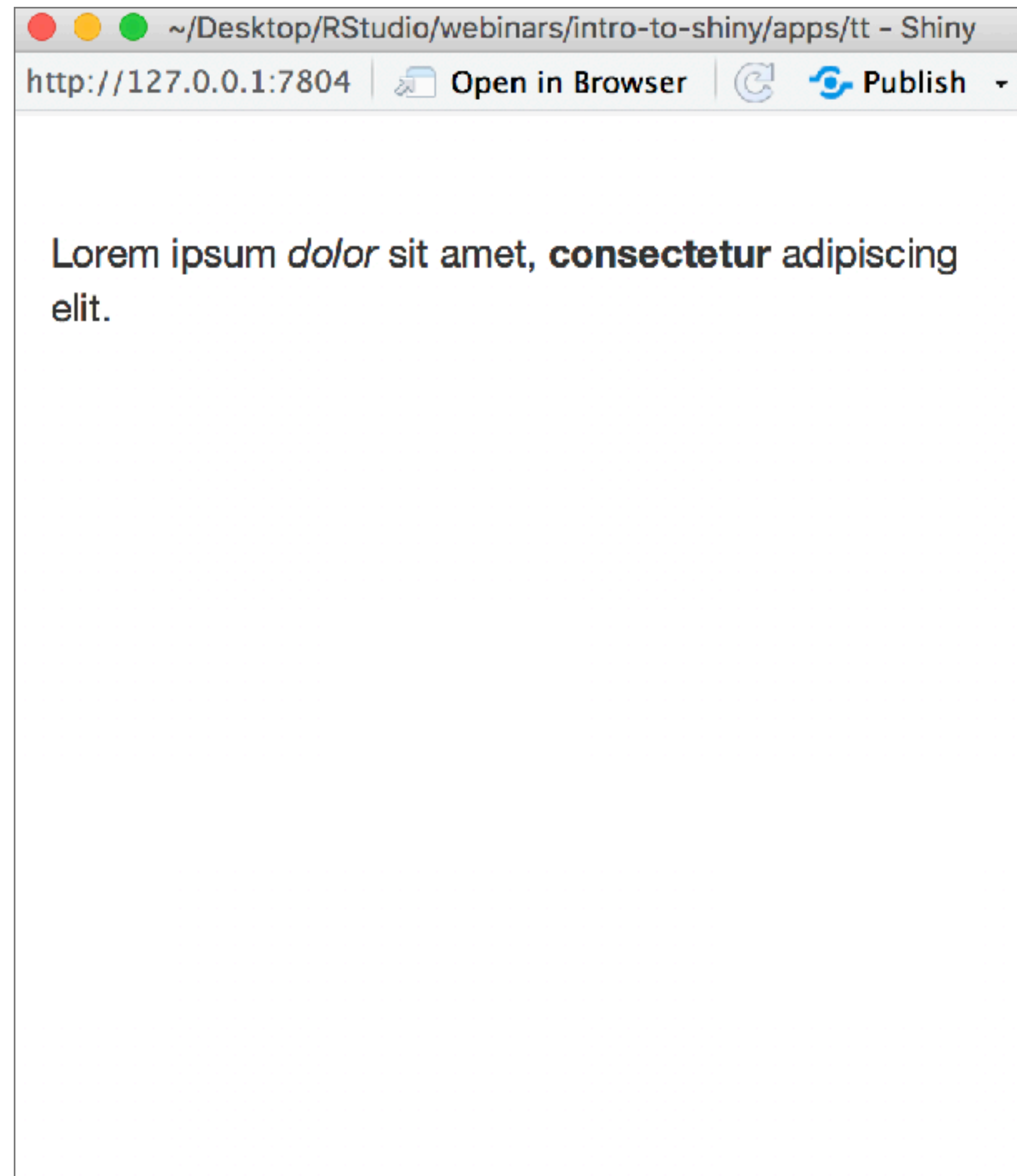
# Nested tags

```
library(shiny)

# Define UI with tags
ui <- fluidPage(
  tags$p("Lorem ipsum",
    tags$em("dolor"), "sit amet,",
    tags$b("consectetur"),
    "adipiscing elit.")
)

# Define server fn that does nothing :)
server <- function(input, output) {}

# Create the app object
shinyApp(ui = ui, server = server)
```



# Common tags

<code>tags\$p(...)</code>	<code>p(...)</code>
<code>tags\$h1(...)</code>	<code>h1(...)</code>
<code>tags\$h2(...)</code>	<code>h2(...)</code>
<code>tags\$h3(...)</code>	<code>h3(...)</code>
<code>tags\$h4(...)</code>	<code>h4(...)</code>
<code>tags\$h5(...)</code>	<code>h5(...)</code>
<code>tags\$h6(...)</code>	<code>h6(...)</code>
<code>tags\$a(...)</code>	<code>a(...)</code>
<code>tags\$br(...)</code>	<code>br(...)</code>
<code>tags\$div(...)</code>	<code>div(...)</code>
<code>tags\$span(...)</code>	<code>span(...)</code>
<code>tags\$pre(...)</code>	<code>pre(...)</code>
<code>tags\$code(...)</code>	<code>code(...)</code>
<code>tags\$img(...)</code>	<code>img(...)</code>
<code>tags\$strong(...)</code>	<code>strong(...)</code>
<code>tags\$em(...)</code>	<code>em(...)</code>
<code>tags\$hr(...)</code>	<code>hr(...)</code>



# Common tags

```
> tags$a("Anchor text")  
<a>Anchor text</a>  
> a("Anchor text")  
<a>Anchor text</a>
```

```
> tags$br()  
<br/>  
> br()  
<br/>
```

```
> tags$code("Monospace text")  
<code>Monospace text</code>  
> code("Monospace text")  
<code>Monospace text</code>
```

```
> tags$h1("First level header")  
<h1>First level header</h1>  
> h1("First level header")  
<h1>First level header</h1>
```



# HTML

```
> HTML("Hello world, <br/> and then a line break.")  
Hello world, <br/> and then a line break.
```



BUILDING WEB APPLICATIONS IN R WITH SHINY

# Let's practice!



BUILDING WEB APPLICATIONS IN R WITH SHINY

# Layout panels

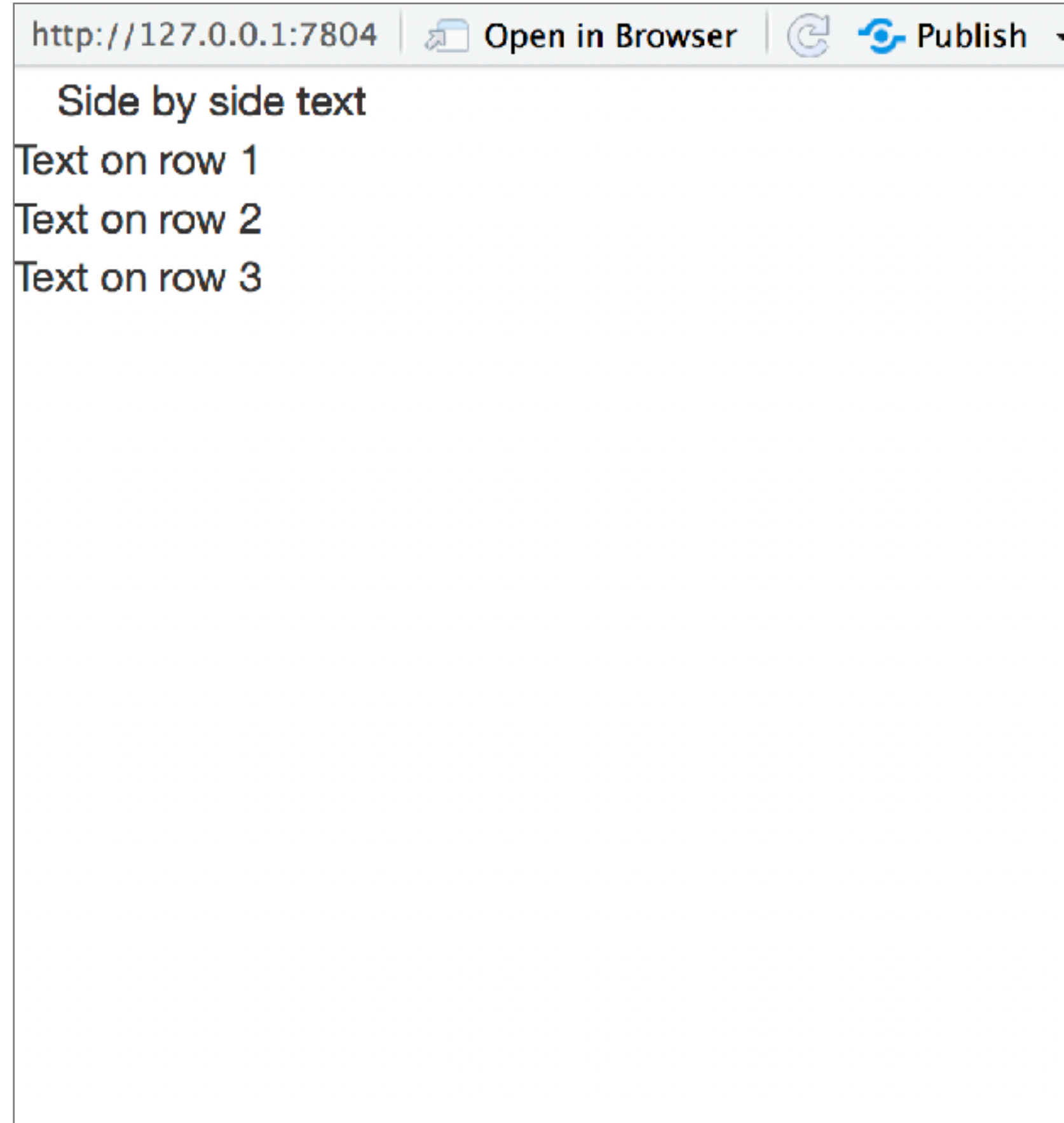
# fluidrow()

```
library(shiny)

# Define UI with fluid rows
ui <- fluidPage(
  "Side", "by", "side", "text",
  fluidRow("Text on row 1"),
  fluidRow("Text on row 2"),
  fluidRow("Text on row 3")
)

# Define server fn that does nothing :)
server <- function(input, output) {}

# Create the app object
shinyApp(ui = ui, server = server)
```



# column()

```
library(shiny)

# Define UI with fluid rows and columns
ui <- fluidPage(
  fluidRow(
    column("R1, C1, width = 3", width = 3),
    column("R1, C2, width = 9", width = 9)
  ),
  fluidRow(
    column("R2, C1, width = 4", width = 4),
    column("R2, C2, width = 4", width = 4),
    column("R2, C3, width = 4", width = 4)
  )
)

# Define server fn that does nothing :)
server <- function(input, output) {}

# Create the app object
shinyApp(ui = ui, server = server)
```

http://127.0.0.1:7804			Open in Browser		Publish
R1, C1, width = 3		R1, C2, width = 9			
R2, C1, width = 4		R2, C2, width = 4		R2, C3, width = 4	

# column()

http://127.0.0.1:7804 | [Open in Browser](#) | [Publish](#)

R1, C1, width = 3	R1, C2, width = 9	
R2, C1, width = 4	R2, C2, width = 4	R2, C3, width = 4

$$3 + 9 = 12$$

$$4 + 4 + 4 = 12$$

http://127.0.0.1:7804 | [Open in Browser](#) | [Publish](#)

R1, C1, width = 3	R1, C2, width = 9	
R2, C1, width = 4	R2, C2, width = 4	R2, C3, width = 4

# Panels

- Use panels to group multiple elements into a single element that has its own properties.
- Especially important and useful for complex apps with a large number of inputs and outputs such that it might not be clear to the user where to get started.



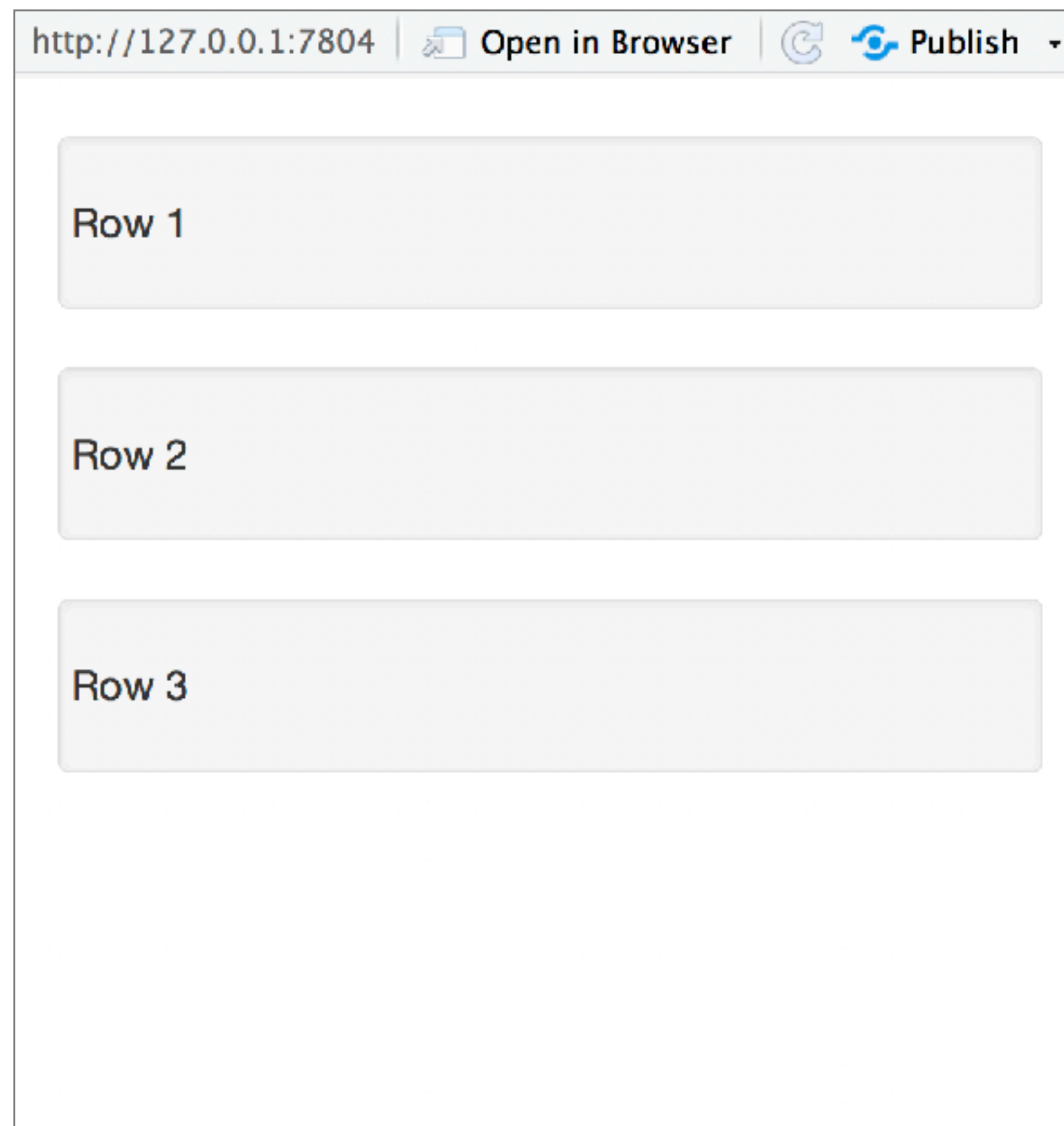
# wellPanel()

```
library(shiny)

# Define UI with wellPanels
ui <- fluidPage(
  wellPanel( fluidRow("Row 1") ),
  wellPanel( fluidRow("Row 2") ),
  wellPanel( fluidRow("Row 3") )
)

# Define server fn that does nothing :)
server <- function(input, output) {}

# Create the app object
shinyApp(ui = ui, server = server)
```



# Panels

`absolutePanel(...)`

`fixedPanel(...)`

`conditionalPanel(...)`

`headerPanel(...)`

`mainPanel(...)`

`navlistPanel(...)`

`sidebarPanel(...)`

`tabPanel(...)`

`tabsetPanel(...)`

`titlePanel(...)`

`inputPanel(...)`

`wellPanel(...)`

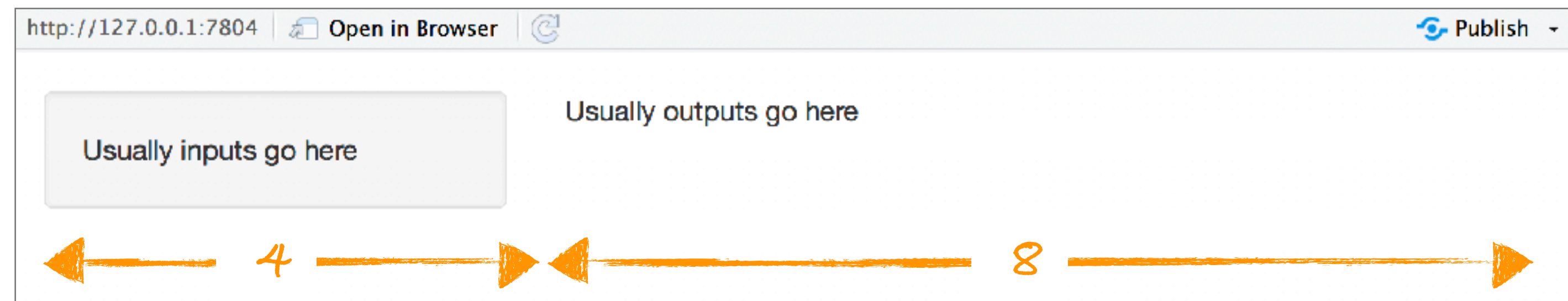
# sidebarPanel() and mainPanel()

```
library(shiny)

# Define UI with default width sidebar
ui <- fluidPage(
  sidebarLayout(
    sidebarPanel("Usually inputs go here"),
    mainPanel("Usually outputs go here")
  )
)

# Define server fn that does nothing :)
server <- function(input, output) {}

# Create the app object
shinyApp(ui = ui, server = server)
```



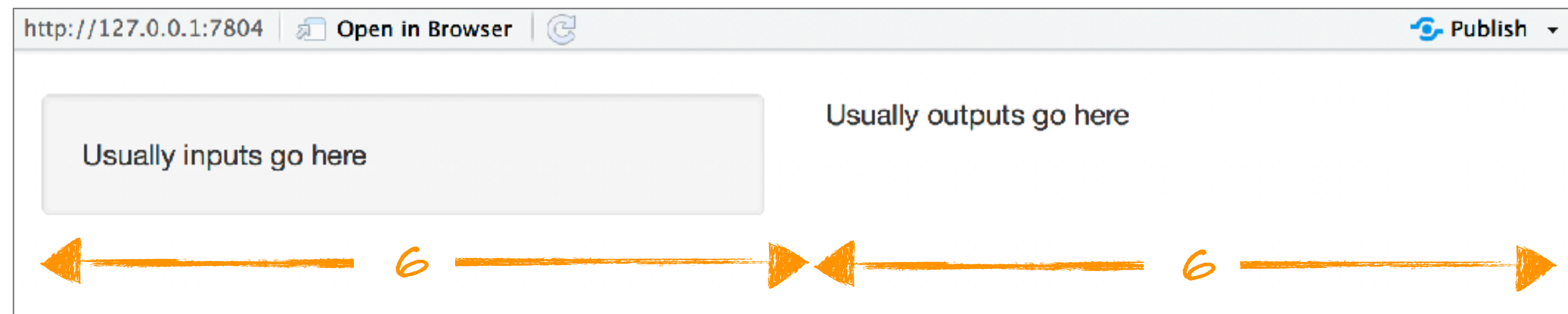
# sidebarPanel() and mainPanel()

```
library(shiny)

# Define UI with custom width sidebar
ui <- fluidPage(
  sidebarLayout(
    sidebarPanel("Usually inputs go here",
                 width = 6),
    mainPanel("Usually outputs go here",
              width = 6)
  )
)

# Define server fn that does nothing :)
server <- function(input, output) {}

# Create the app object
shinyApp(ui = ui, server = server)
```



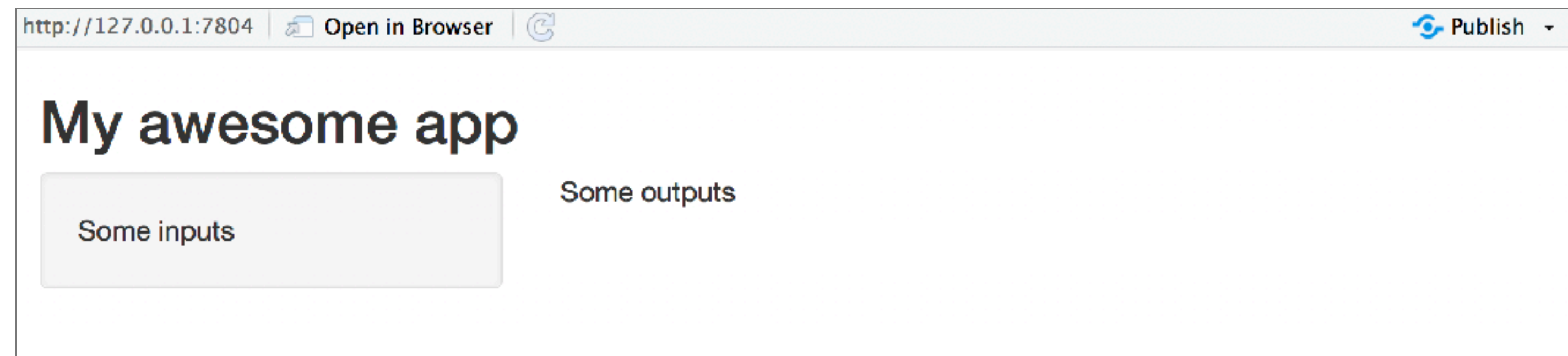
# titlePanel()

```
library(shiny)

# Define UI with title panel
ui <- fluidPage(
  titlePanel("My awesome app"),
  sidebarLayout(
    sidebarPanel("Some inputs"),
    mainPanel("Some outputs")
  )
)

# Define server fn that does nothing :)
server <- function(input, output) {}

# Create the app object
shinyApp(ui = ui, server = server)
```



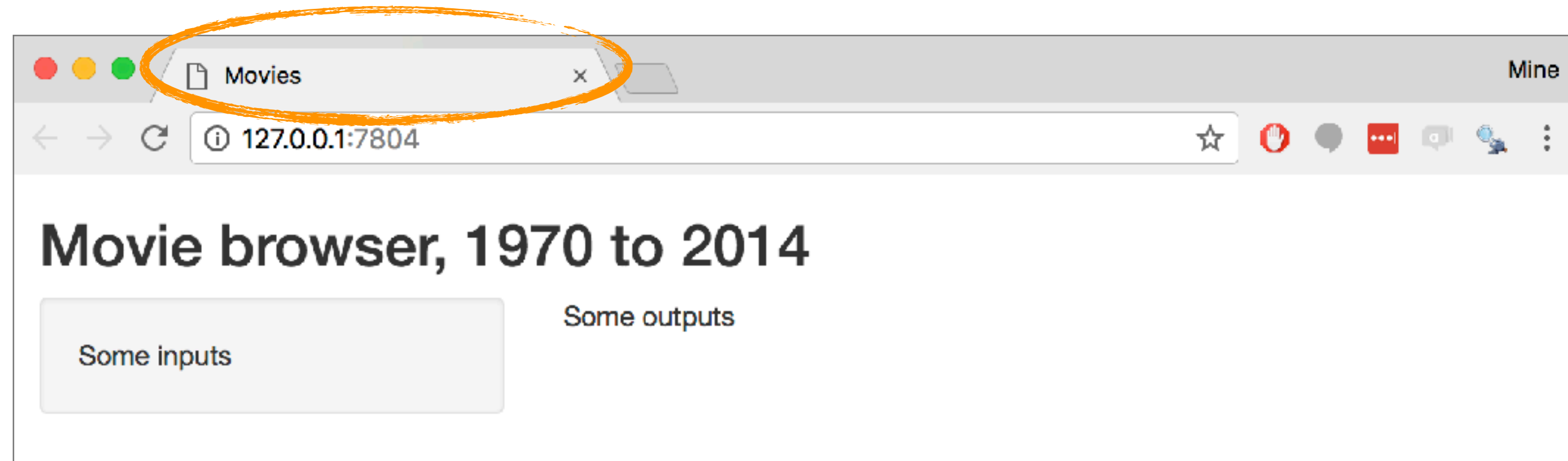
# titlePanel() with windowTitle

```
library(shiny)

# Define UI with title panel
ui <- fluidPage(
  titlePanel("Movie browser, 1970 to 2014",
    windowTitle = "Movies"),
  sidebarLayout(
    sidebarPanel("Some inputs"),
    mainPanel("Some outputs")
  )
)



# Define server fn that does nothing :)
server <- function(input, output) {}

# Create the app object
shinyApp(ui = ui, server = server)
```





# conditionalPanel()

http://127.0.0.1:7804 |  Open in Browser |  Publish ▾

## Random number generator

**Number of digits:**

any ▾




Generate new random number

Your random number is

**21454**



# conditionalPanel()

http://127.0.0.1:7804 |  Open in Browser |  |  Publish ▾

## Random number generator

**Number of digits:**

any ▲

any

selected

Your random number is

**9770637928**

# conditionalPanel()

http://127.0.0.1:7804 [Open in Browser](#) [Publish](#)

## Random number generator

**Number of digits:**

selected ▼

**How many digits?**

1 4 10

1 2 3 4 5 6 7 8 9 10

Generate new random number

Your random number is

**3239**



BUILDING WEB APPLICATIONS IN R WITH SHINY

# Let's practice!



BUILDING WEB APPLICATIONS IN R WITH SHINY

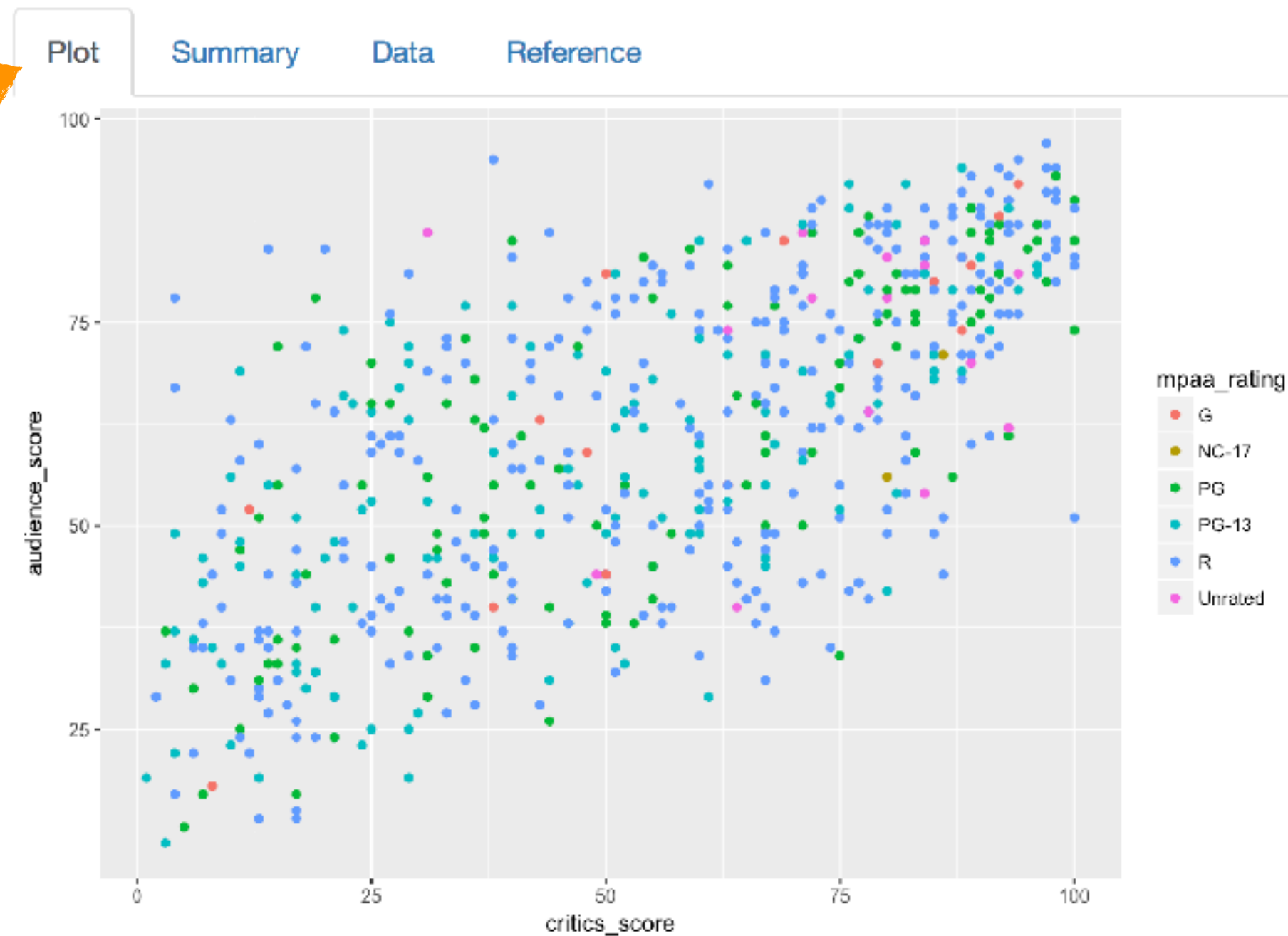
# **Tabs and tabsets**

# tabsetPanel() and tabPanel()

```
mainPanel(  
  tabsetPanel(type = "tabs",  
    tabPanel("Plot", plotOutput("plot")),  
    tabPanel("Summary", tableOutput("summary")),  
    tabPanel("Data", DT::dataTableOutput("data")),  
    tabPanel("Reference",  
      tags$p("There data were obtained from",  
        tags$a("IMDB", href = "http://www.imdb.com/"), "and",  
        tags$a("Rotten Tomatoes", href = "https://www.rottentomatoes.com/"),  
        "."),  
      tags$p("The data represent", nrow(movies), "randomly sampled movies  
released between 1972 to 2014 in the United States.")  
    )  
  )  
)
```

# tabPanel()

```
mainPanel(  
  tabsetPanel(type = "tabs",  
    tabPanel("Plot", plotOutput("plot")),  
    tabPanel("Summary", tableOutput("summary")),  
    tabPanel("Data", DT::dataTableOutput("data")),  
    tabPanel("Reference",  
      tags$p("There data were obtained from",  
        tags$a("IMDB", href = "http://www.imdb.com/"), "and",  
        tags$a("Rotten Tomatoes", href = "https://  
www.rottentomatoes.com/"), "."),  
      tags$p("The data represent", nrow(movies), "randomly  
sampled movies released between 1972 to 2014 in the United  
States.")  
    )  
  )  
)
```





# tabPanel()

```
mainPanel(  
  tabsetPanel(type = "tabs",  
    tabPanel("Plot", plotOutput("plot")),  
    tabPanel("Summary", tableOutput("summary")),  
    tabPanel("Data", DT::dataTableOutput("data")),  
    tabPanel("Reference",  
      tags$p("There data were obtained from",  
        tags$a("IMDB", href = "http://www.imdb.com/"), "and",  
        tags$a("Rotten Tomatoes", href = "https://  
www.rottentomatoes.com/"), "."),  
      tags$p("The data represent", nrow(movies), "randomly  
sampled movies released between 1972 to 2014 in the United  
States.")  
    )  
  )  
)
```

Plot

Summary

Data

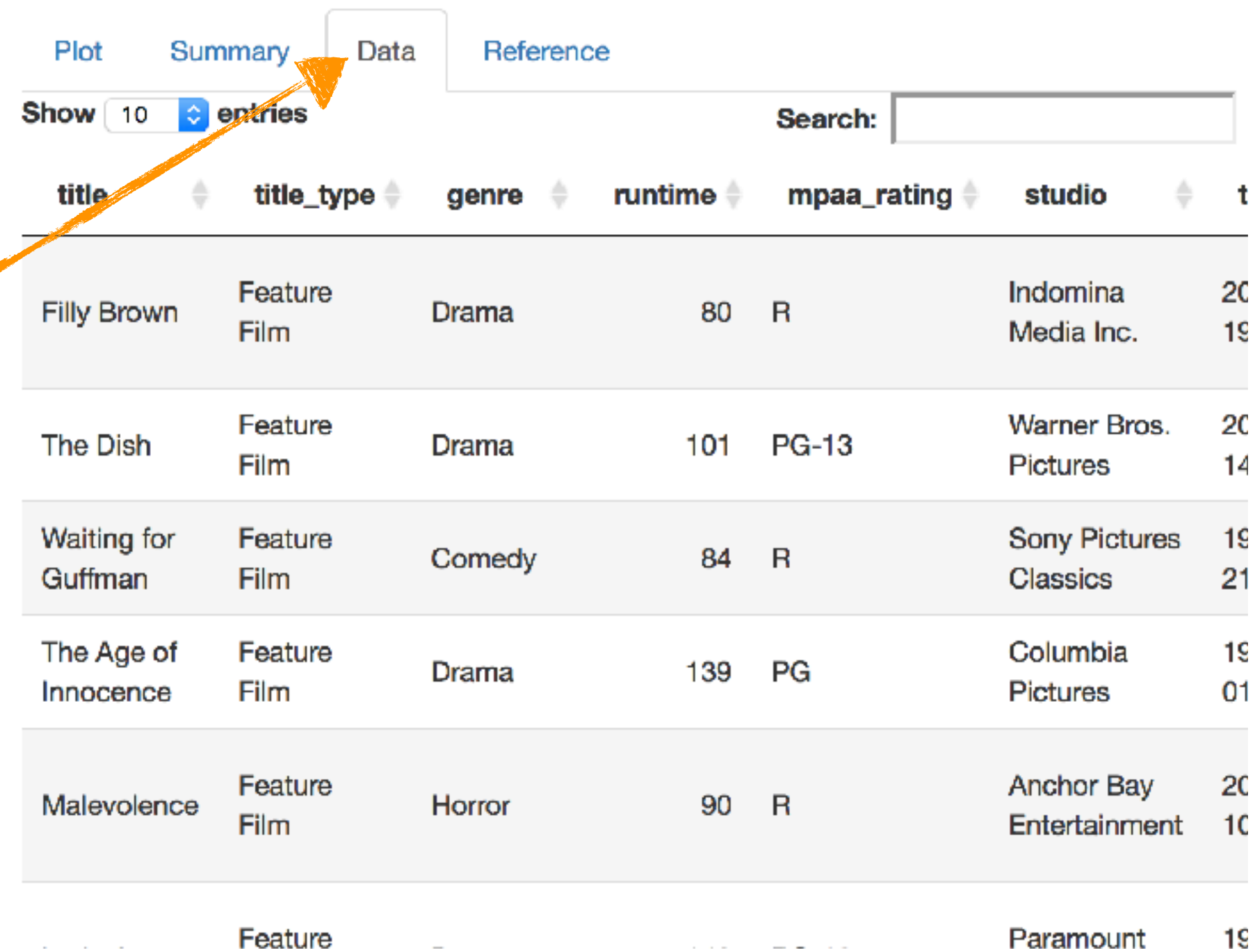
Reference

mpaa_rating	mean_as	sd_as	mean_cs	sd_cs	n	cor
G	66.625	20.656	62.250	27.939	16	0.836
NC-17	63.500	10.607	83.000	4.243	2	1.000
PG	60.418	20.110	54.491	28.503	110	0.733
PG-13	56.015	19.002	46.085	26.518	130	0.662
R	61.454	19.986	56.877	27.463	317	0.648
Unrated	70.812	14.725	74.938	16.631	16	0.105



# tabPanel()

```
mainPanel(  
  tabsetPanel(type = "tabs",  
    tabPanel("Plot", plotOutput("plot")),  
    tabPanel("Summary", tableOutput("summary")),  
    tabPanel("Data", DT::dataTableOutput("data")),  
    tabPanel("Reference",  
      tags$p("There data were obtained from",  
        tags$a("IMDB", href = "http://www.imdb.com/"), "and",  
        tags$a("Rotten Tomatoes", href = "https://  
www.rottentomatoes.com/"), "."),  
      tags$p("The data represent", nrow(movies), "randomly  
sampled movies released between 1972 to 2014 in the United  
States.")  
    )  
  )  
)
```



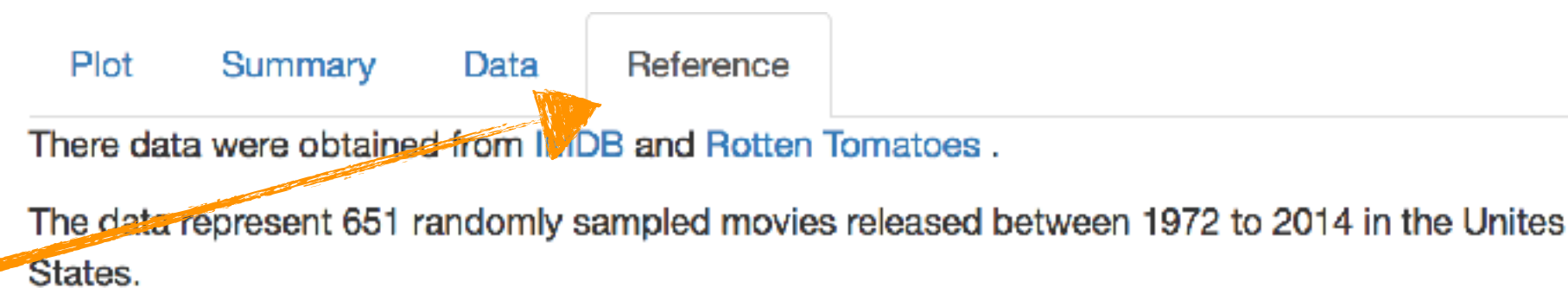
Plot Summary **Data** Reference

Show 10 entries Search:

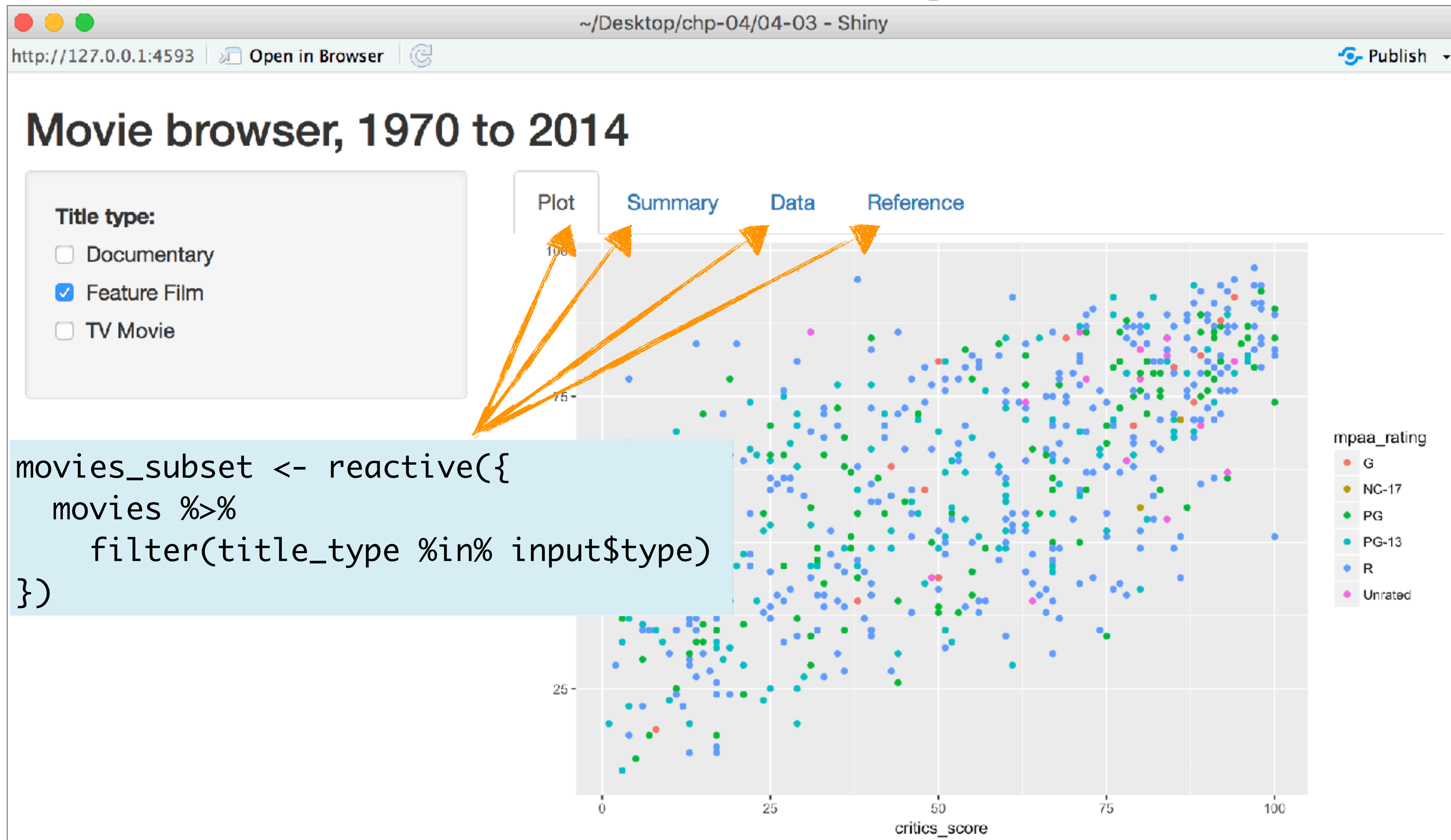
title	title_type	genre	runtime	mpaa_rating	studio	t
Filly Brown	Feature Film	Drama	80	R	Indomina Media Inc.	2019
The Dish	Feature Film	Drama	101	PG-13	Warner Bros. Pictures	2014
Waiting for Guffman	Feature Film	Comedy	84	R	Sony Pictures Classics	1999
The Age of Innocence	Feature Film	Drama	139	PG	Columbia Pictures	1993
Malevolence	Feature Film	Horror	90	R	Anchor Bay Entertainment	2007
...	Feature	-	...	-	Paramount	19...

# tabPanel()

```
mainPanel(  
  tabsetPanel(type = "tabs",  
    tabPanel("Plot", plotOutput("plot")),  
    tabPanel("Summary", tableOutput("summary")),  
    tabPanel("Data", DT::dataTableOutput("data")),  
    tabPanel("Reference",  
      tags$p("There data were obtained from",  
        tags$a("IMDB", href = "http://www.imdb.com/"), "and",  
        tags$a("Rotten Tomatoes", href = "https://  
www.rottentomatoes.com/"), "."),  
      tags$p("The data represent", nrow(movies), "randomly  
sampled movies released between 1972 to 2014 in the United  
States.")  
    )  
  )  
)
```

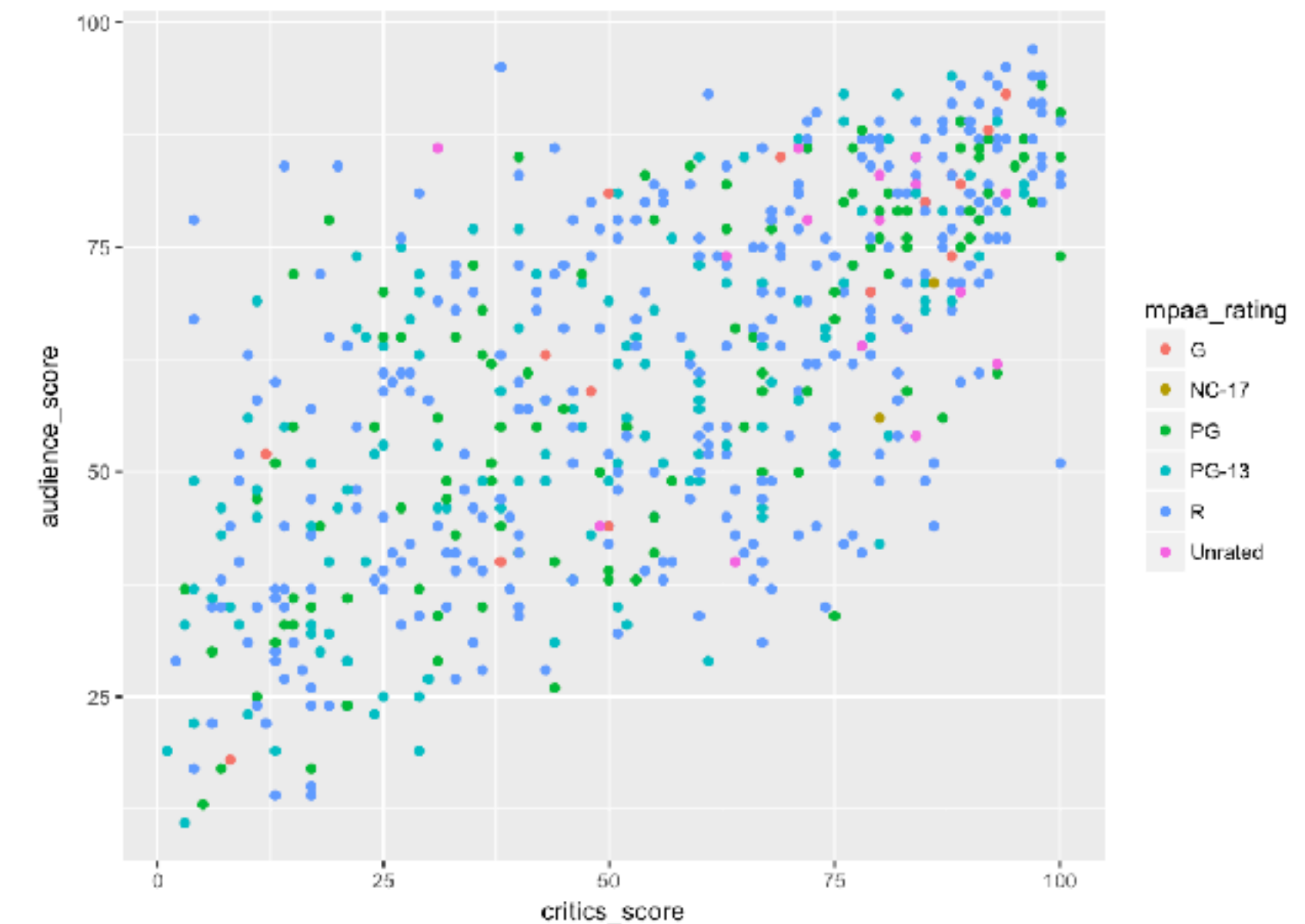


# Tabs and reactivity



# navlistPanel()

```
mainPanel(  
  navlistPanel(tabPanel("Plot", plotOutput("plot")),  
               tabPanel("Summary", tableOutput("summary")),  
               tabPanel("Data", DT::dataTableOutput("data")),  
               tabPanel("Reference",  
                 tags$p("There data were obtained from",  
                   tags$a("IMDB", href = "http://www.imdb.com/"), "and",  
                   tags$a("Rotten Tomatoes", href = "https://  
www.rottentomatoes.com/"), "."),  
                 tags$p("The data represent", nrow(movies), "randomly  
sampled movies released between 1972 to 2014 in the United  
States.")  
               )  
)
```



# Tabs

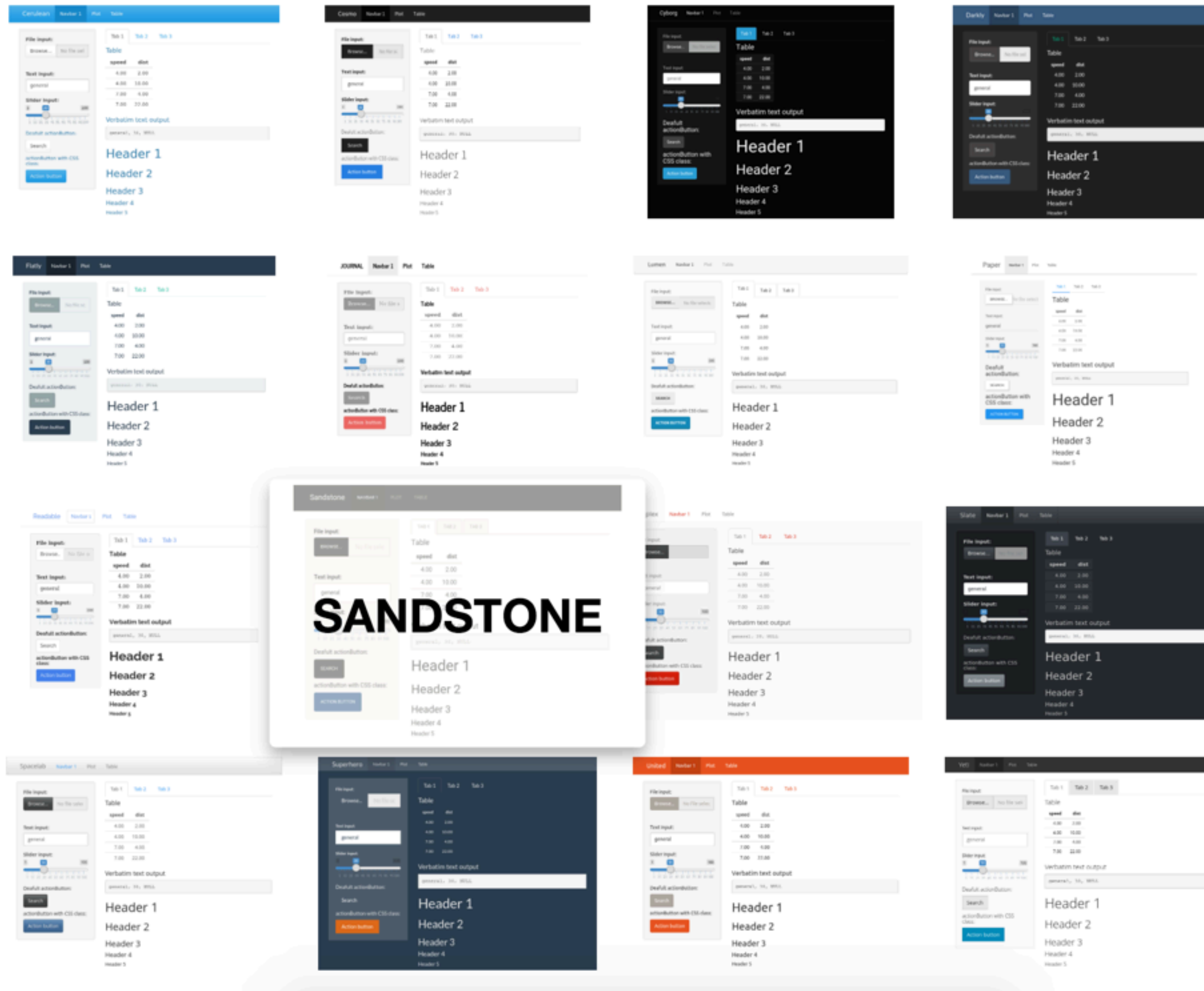
[Plot](#)[Summary](#)[Data](#)[Reference](#)

There data were obtained from [IMDB](#) and [Rotten Tomatoes](#) .

The data represent 651 randomly sampled movies released between 1972 to 2014 in the Unites States.



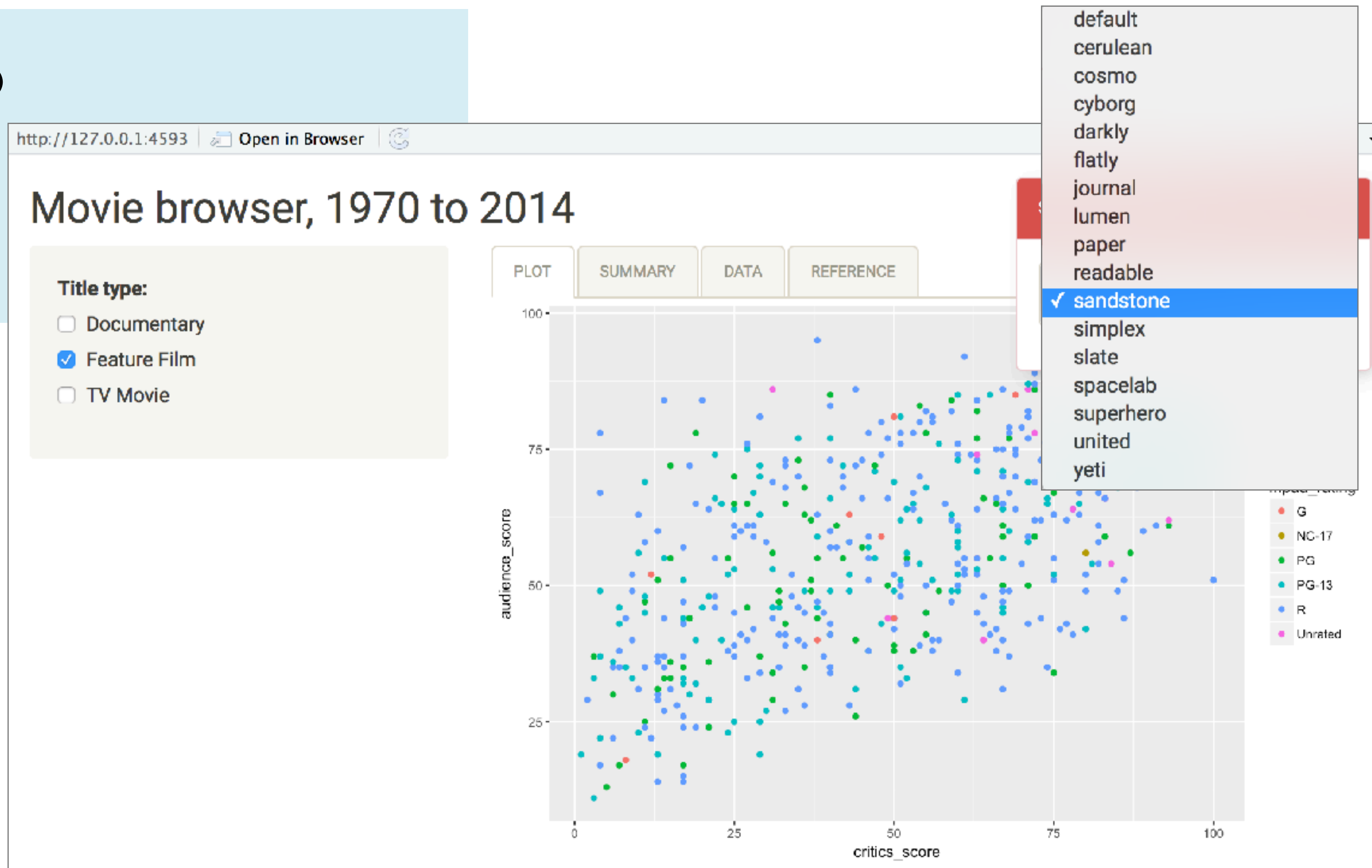
# shinythemes



# shinythemes

```
library(shiny)
library(shinythemes)
```

```
ui <- fluidPage(
  themeSelector(),
  ...
)
```







BUILDING WEB APPLICATIONS IN R WITH SHINY

# Let's practice!



BUILDING WEB APPLICATIONS IN R WITH SHINY

# Congratulations!

# What did we learn?

- Design a Shiny app from scratch
- Essentials of reactive programming
- Customizing your app's UI
- Reactivity best practices

# shinyapps.io

- Server maintained by RStudio
- Easy to use, secure, and scalable
- Comes with built in metrics
- Free tier available!

# Shiny Server

- Free and open source
- Runs on premises, moving your computation closer to your data

# Pro options

- RStudio Server Pro
- RStudio Connect
- Find out more at [shiny.rstudio.com/deploy](https://shiny.rstudio.com/deploy)



BUILDING WEB APPLICATIONS IN R WITH SHINY

# Congratulations!