

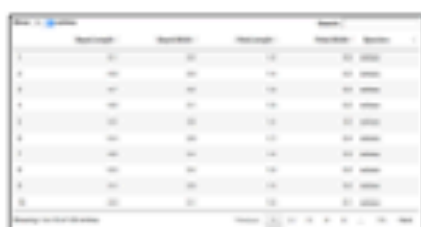


BUILDING WEB APPLICATIONS IN R WITH SHINY

Rendering functions

Outputs – `render*()` and `*Output()` functions work together to add R output to the UI

works
with



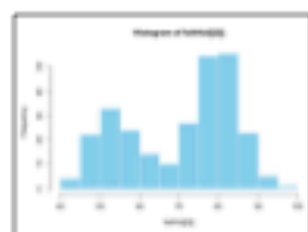
DT::renderDataTable(expr, options,
callback, escape, env, quoted)

dataTableOutput(outputId, icon, ...)



renderImage(expr, env, quoted,
deleteFile)

imageOutput(outputId, width, height,
click, dblclick, hover, hoverDelay, inline,
hoverDelayType, brush, clickId, hoverId)



renderPlot(expr, width, height, res, ...,
env, quoted, func)

plotOutput(outputId, width, height, click,
dblclick, hover, hoverDelay, inline,
hoverDelayType, brush, clickId, hoverId)

```
'data.frame': 3 obs. of 2 variables:
 $ Sepal.Length: num 5.1 4.9 4.7
 $ Sepal.Width : num 3.5 3 3.2
```

renderPrint(expr, env, quoted, func,
width)

verbatimTextOutput(outputId)

	Sepal.Length	Sepal.Width	Petal.Length	Petal.Width	Species
1	5.1	3.5	1.4	0.2	setosa
2	4.9	3.0	1.4	0.2	setosa
3	4.7	3.2	1.3	0.2	setosa
4	5.2	3.7	1.5	0.2	setosa
5	5.0	3.4	1.4	0.2	setosa
6	5.4	3.9	1.7	0.2	setosa

renderTable(expr,..., env, quoted, func)

tableOutput(outputId)

foo

renderText(expr, env, quoted, func)

textOutput(outputId, container, inline)



renderUI(expr, env, quoted, func)

&

uiOutput(outputId, inline, container, ...)

htmlOutput(outputId, inline, container, ...)

Outputs – render*() and *Output() functions work together to add R output to the UI

works
with



DT::renderDataTable(expr, options, callback, escape, env, quoted) **dataTableOutput**(outputId, icon, ...)



renderImage(expr, env, quoted, deleteFile)

imageOutput(outputId, width, height, click, dblclick, hover, hoverDelay, inline, hoverDelayType, brush, clickId, hoverId)



renderPlot(expr, width, height, res, ..., env, quoted, func)

plotOutput(outputId, width, height, click, dblclick, hover, hoverDelay, inline, hoverDelayType, brush, clickId, hoverId)

```
"data-frame" 3 obs. of 2 variables:
  $ Sepal.Length: num  5.1 4.9 4.7
  $ Sepal.Width : num  3.5 3 3.2
```

renderPrint(expr, env, quoted, func, width)

verbatimTextOutput(outputId)

	Sepal.Length	Sepal.Width	Petal.Length	Petal.Width	Species
1	5.10	3.50	1.40	0.20	setosa
2	4.90	3.00	1.40	0.20	setosa
3	4.70	3.20	1.30	0.20	setosa
4	5.00	3.10	1.50	0.20	setosa
5	5.40	3.40	1.40	0.20	setosa
6	5.20	3.20	1.70	0.20	setosa

renderTable(expr,..., env, quoted, func)

tableOutput(outputId)

foo

renderText(expr, env, quoted, func)

textOutput(outputId, container, inline)



renderUI(expr, env, quoted, func)

&

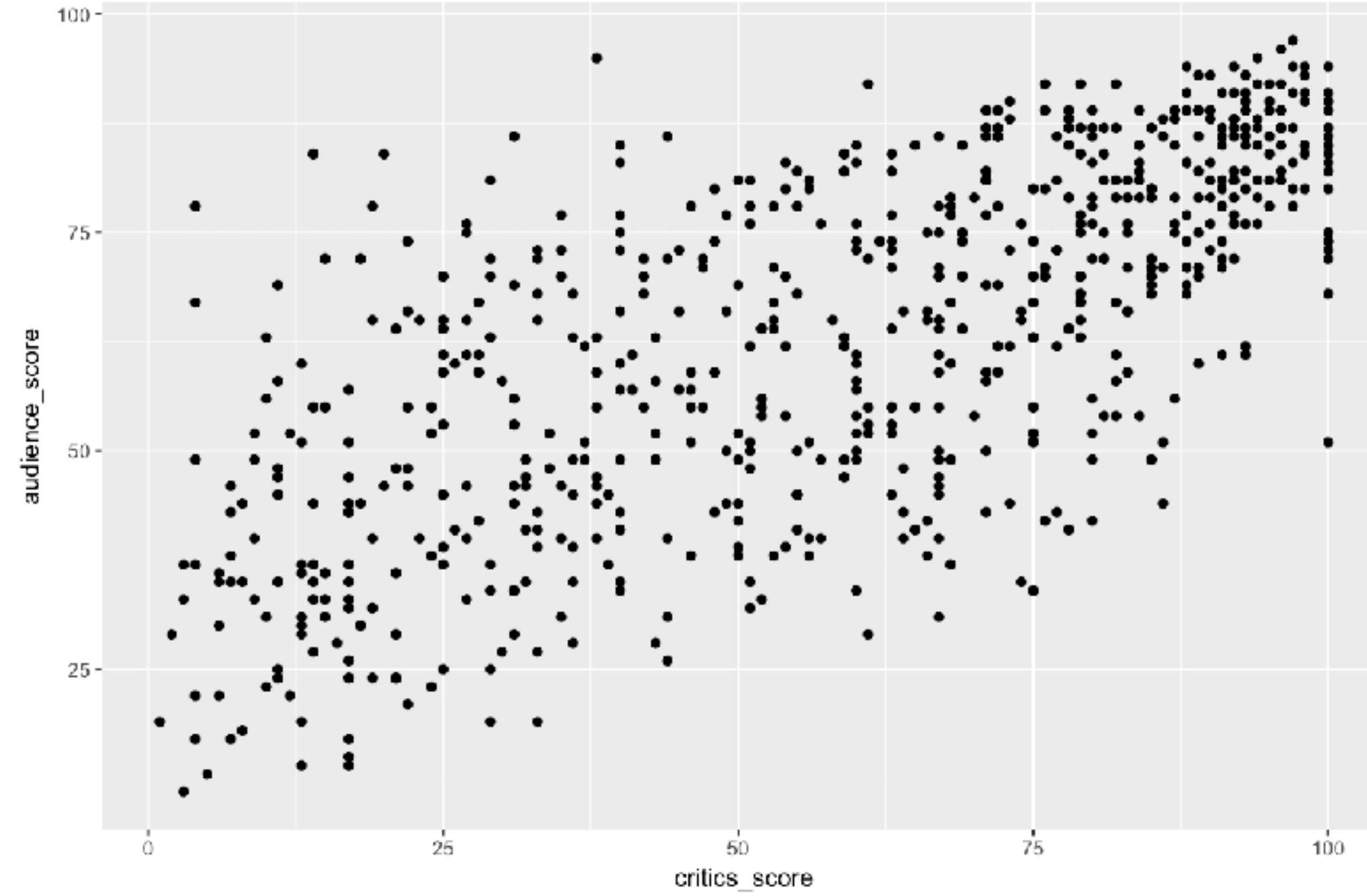
uiOutput(outputId, inline, container, ...)
htmlOutput(outputId, inline, container, ...)

Y-axis:

audience_score ▼

X-axis:

critics_score ▼



Y-axis:

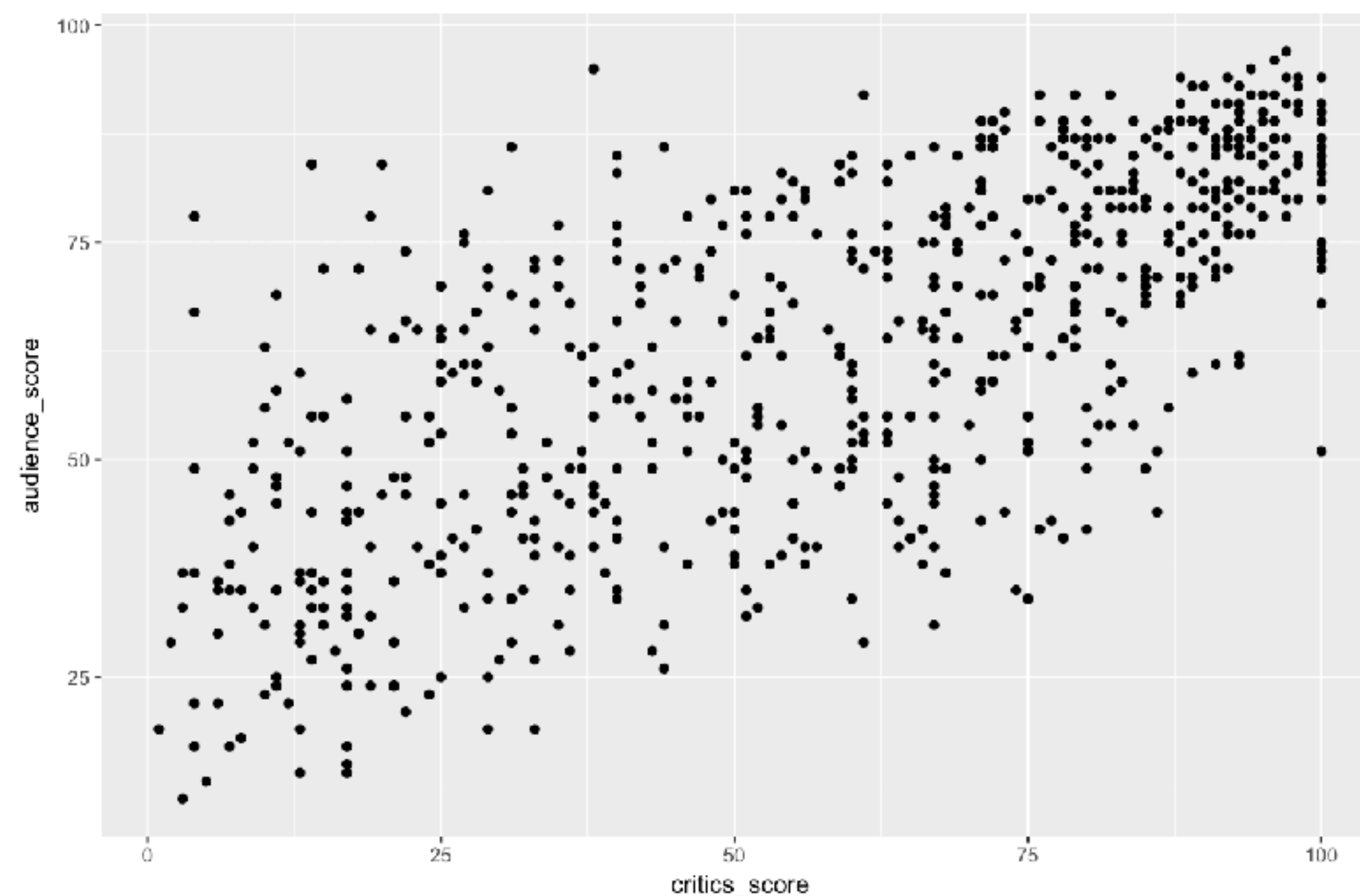
audience_score ▼

X-axis:

critics_score ▼

Select title type:

- ☒ Documentary
- ☒ Feature Film
- ☒ TV Movie



mpaa_rating	Mean	SD	n
G	1.2742	0.8215	19
NC-17	0.7628	0.0888	2
PG	1.4805	1.3242	118
PG-13	1.9962	2.3824	133
R	1.5282	1.7907	329
Unrated	0.9637	0.3054	50

Score ratio (audience / critics' scores) summary statistics by MPAA rating.

renderTable

Add a table beneath the plot displaying summary statistics for a new variable: `score_ratio = audience_score / critics_score`.

1. Calculate the new variable.
2. **ui:** Add an input widget that the user can interact with to check boxes for selected title types.
3. **ui:** Add an output defining where the summary table should appear.
4. **server:** Add a reactive expression that creates the summary table.

renderTable

Add a table beneath the plot displaying summary statistics for a new variable: `score_ratio = audience_score / critics_score`.

1. Calculate the new variable.

```
# Create new variable:  
# ratio of critics and audience scores  
movies <- movies %>%  
  mutate(score_ratio = audience_score / critics_score)
```

renderTable

Add a table beneath the plot displaying summary statistics for a new variable: `score_ratio = audience_score / critics_score`.

2. **ui:** Add an input widget that the user can interact with to check boxes for selected title types.

```
# Subset for title types
checkboxGroupInput(inputId = "selected_title_type",
               label = "Select title type:",
               choices = levels(movies$title_type),
               selected = levels(movies$title_type))
```


renderTable

Add a table beneath the plot displaying summary statistics for a new variable: `score_ratio = audience_score / critics_score`.

3. **ui:** Add an output defining where the summary table should appear.

```
mainPanel(  
  # Show scatterplot  
  plotOutput(outputId = "scatterplot"),  
  # Show data table  
  tableOutput(outputId = "summarytable")  
)
```

renderTable

Add a table beneath the plot displaying summary statistics for a new variable: `score_ratio = audience_score / critics_score`.

4. **server:** Add a reactive expression that creates the summary table.

```
output$summarytable <- renderTable(  
  {movies %>%  
    filter(title_type %in% input$selected_title_type) %>%  
    group_by(mpaa_rating) %>%  
    summarise(Mean = mean(score_ratio), SD = sd(score_ratio), n = n())},  
  striped = TRUE, spacing = "l", align = "lccr", digits = 4, width = "90%",  
  caption = "Score ratio (audience / critics' scores) summary statistics by  
  MPAA rating."  
)
```

mpaa_rating	Mean	SD	n
G	1.27	0.82	19
NC-17	0.76	0.09	2
PG	1.48	1.32	118
PG-13	2.00	2.38	133
R	1.53	1.79	329
Unrated	0.96	0.31	50

mpaa_rating	Mean	SD	n
G	1.2742	0.8215	19
NC-17	0.7628	0.0888	2
PG	1.4805	1.3242	118
PG-13	1.9962	2.3824	133
R	1.5282	1.7907	329
Unrated	0.9637	0.3054	50

Score ratio (audience / critics' scores) summary statistics by MPAA rating.

renderTable

Add a table beneath the plot displaying summary statistics for a new variable: `score_ratio = audience_score / critics_score`.

4. **server:** Add a reactive expression that creates the summary table.

```
output$summarytable <- renderTable(  
  {movies %>%  
    filter(title_type %in% input$selected_title_type) %>%  
    group_by(mpaa_rating) %>%  
    summarise(Mean = mean(score_ratio), SD = sd(score_ratio), n = n())},  
  striped = TRUE, spacing = "l", align = "lccr", digits = 4, width = "90%",  
  caption = "Score ratio (audience / critics' scores) summary statistics by  
  MPAA rating."  
)
```

Recap

- Shiny has a variety of `render*` functions with corresponding `*Output` functions to create and display outputs.
- `render*` functions can take on multiple arguments, the first being the expression for the desired output.
- The expression in the `render*` function should be wrapped in curly braces.



BUILDING WEB APPLICATIONS IN R WITH SHINY

Let's practice!