




# Dashboards

Mine Çetinkaya-Rundel

@minebocek   
mine-cetinkaya-rundel   
mine@stat.duke.edu 

# flexdashboard vs. shinydashboard

flexdashboard	shinydashboard
R Markdown	Shiny UI code
Super easy	Not quite as easy
Static or dynamic	Dynamic
CSS flexbox layout	Bootstrap grid layout



# shinydashboard



**header**



**sidebar**

**body**



```
library(shiny)
library(shinydashboard)
```

```
ui <- dashboardPage(
  dashboardHeader(),
  dashboardSidebar(),
  dashboardBody()
)
```

```
server <- function(input, output) { }
```

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

### User interface

controls the layout and appearance of dashboard


### Server function


contains instructions needed to build dashboard







 Home

 Subset

**651**

Movies



**6.49**

Avg IMDB score



**7**

Oscar wins



The data come from IMDB and Rotten Tomatoes on a random sample of movies released in the US between 1970 and 2014





Home

Subset

## Select title type

Select a title type using the drop down menu below.

**Title type:**

Feature Film

591

Feature Film



6.39

Avg IMDB score



## Select variables to plot:

**Y-axis:**

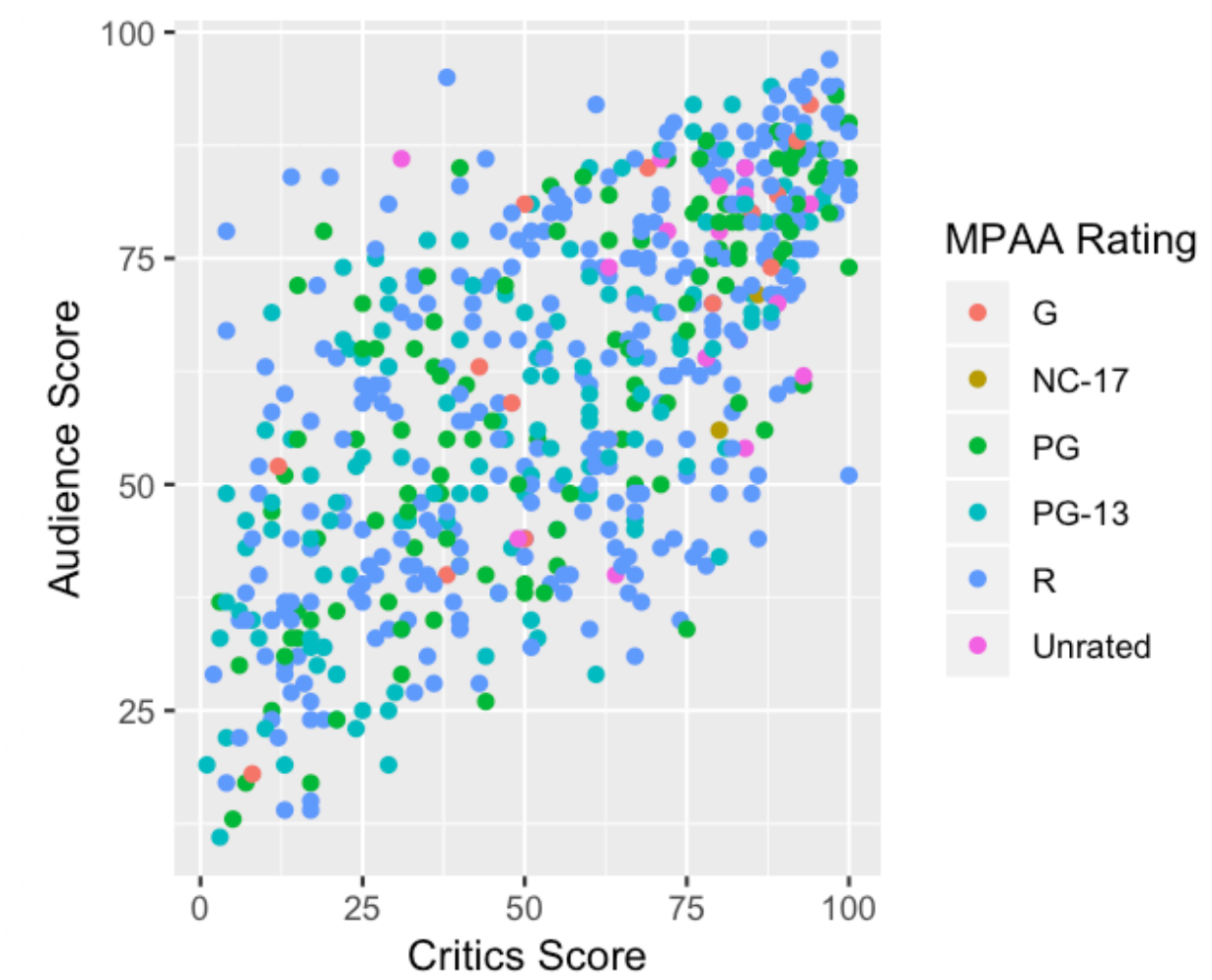
Audience Score

**X-axis:**

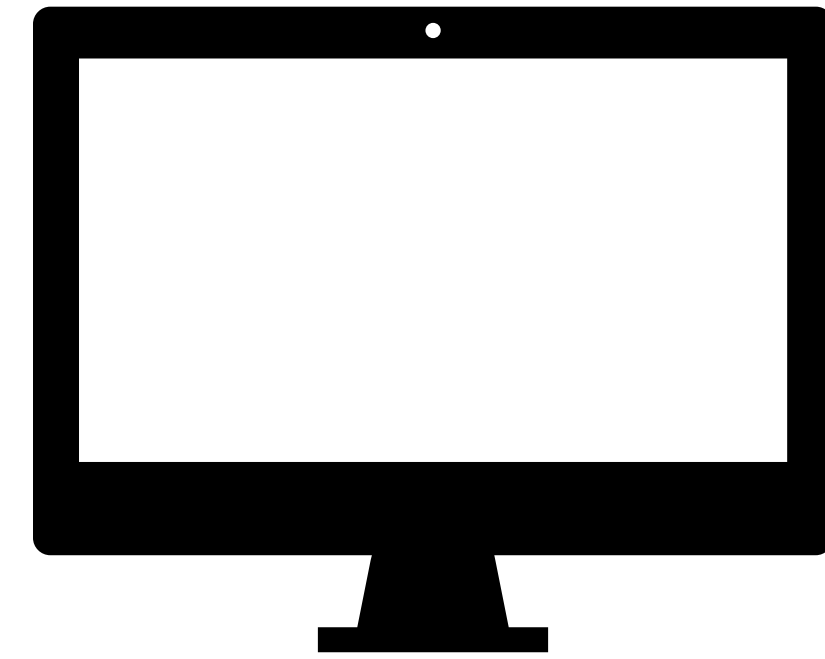
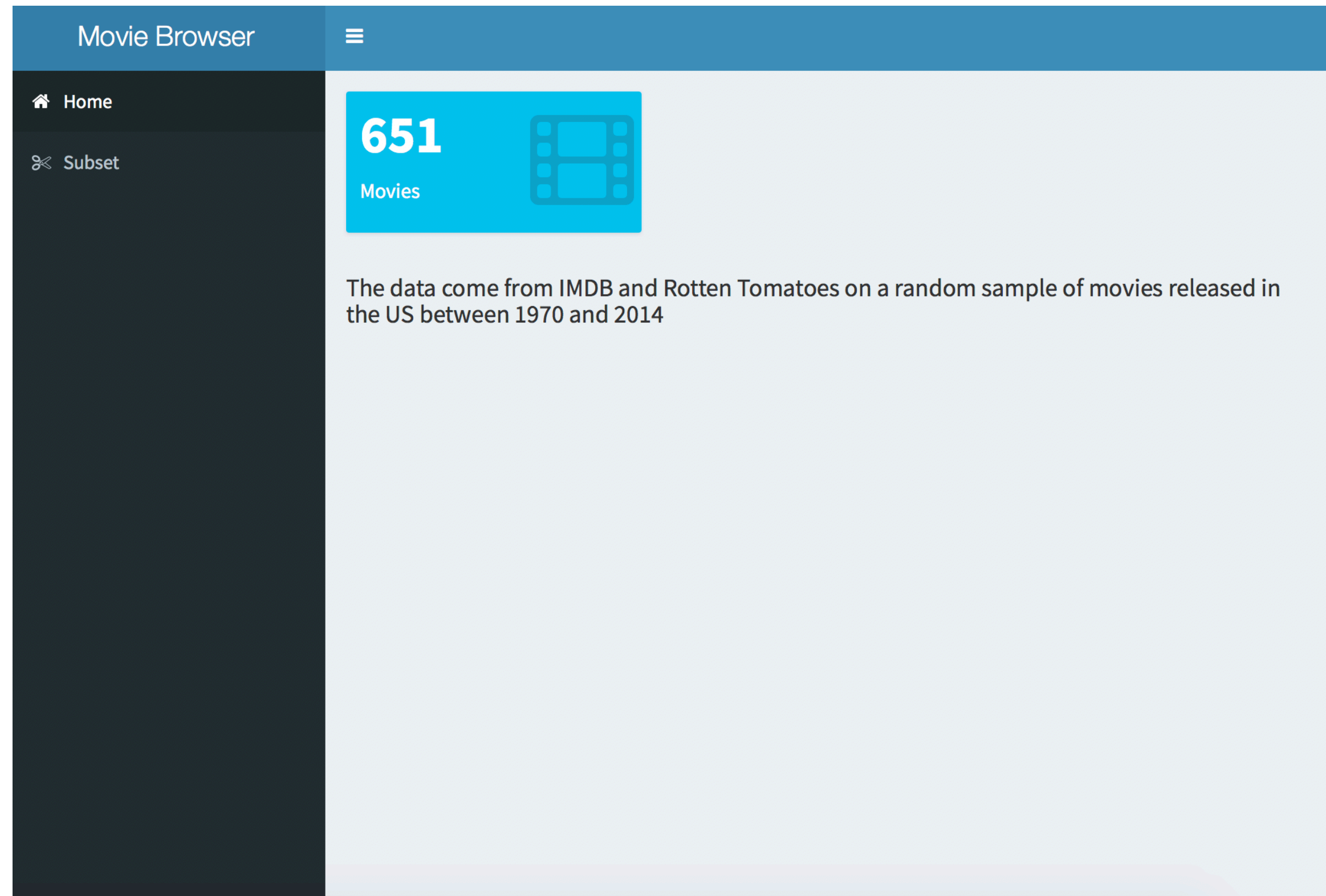
Critics Score

**Color by:**

MPAA Rating



# dashboards/moviedash-01.R



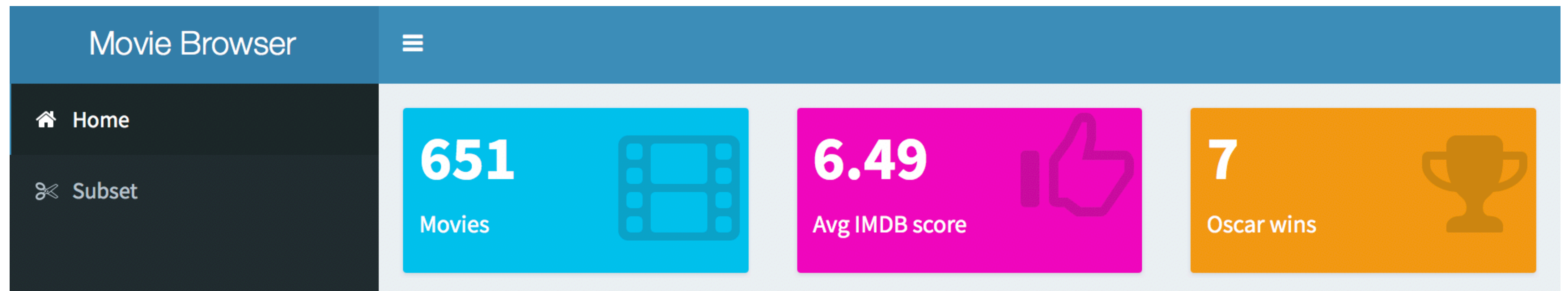
## DEMO





# Your turn

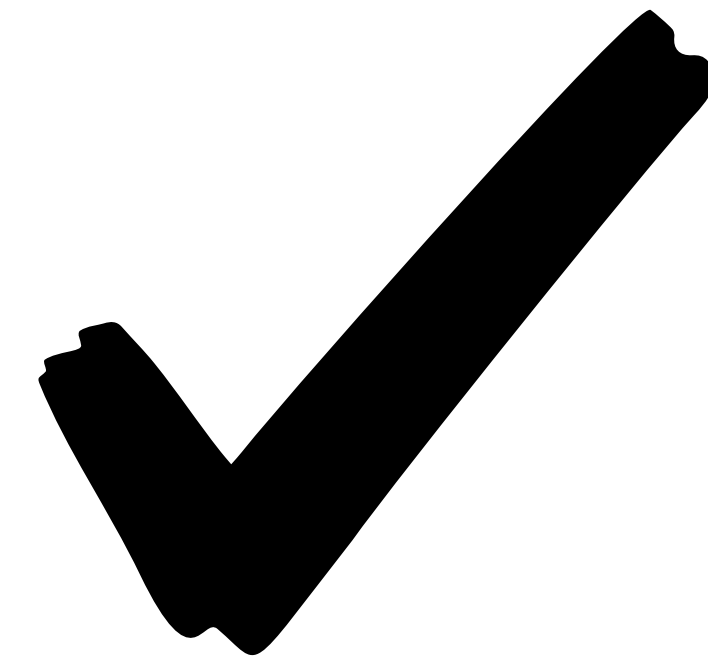
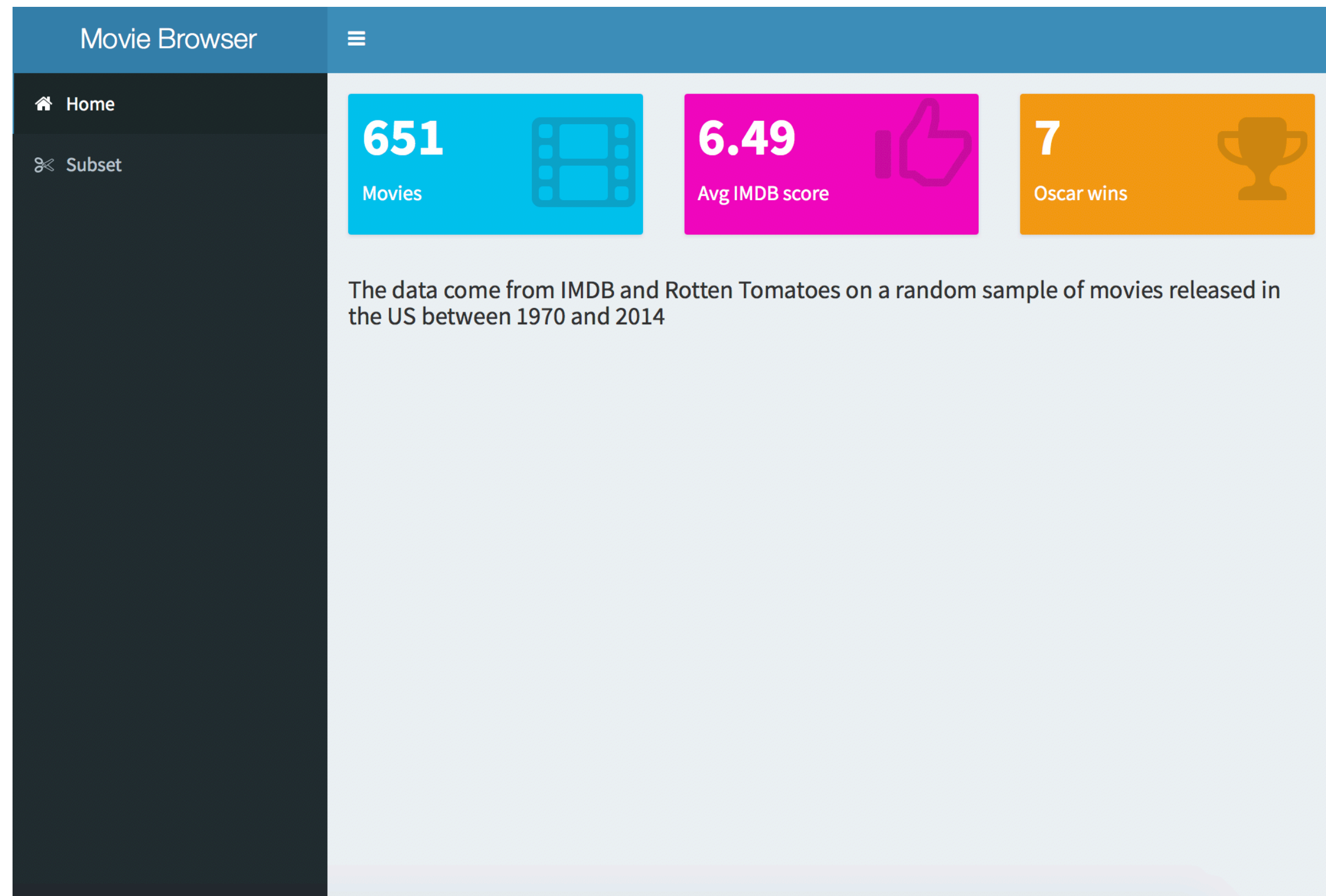
- Start with `dashboards/moviedash-01.R`
- Add two more value boxes: average IMDB rating and number of Oscar wins
- Try to match the colors and icons as well



5<sub>m</sub> 00<sub>s</sub>



# dashboards/moviedash-02.R

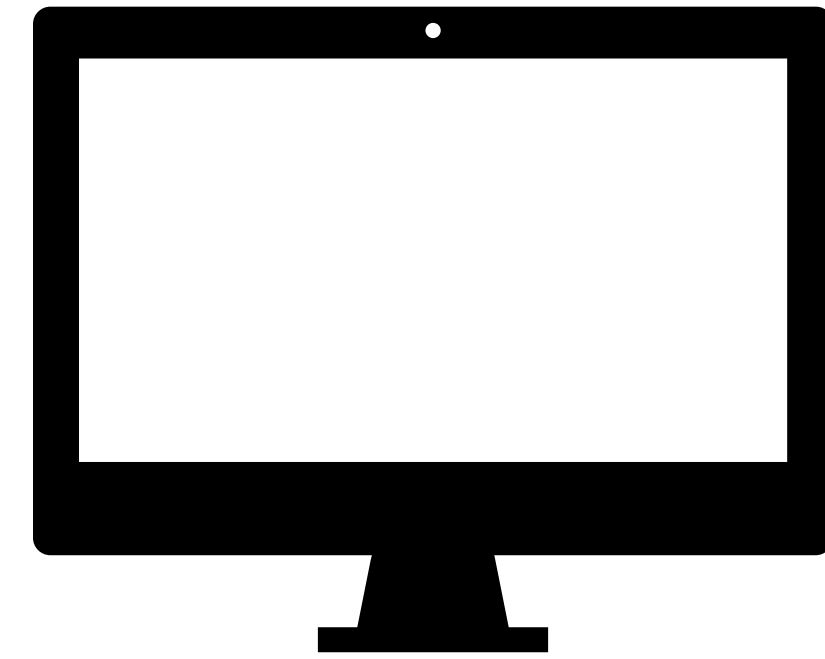
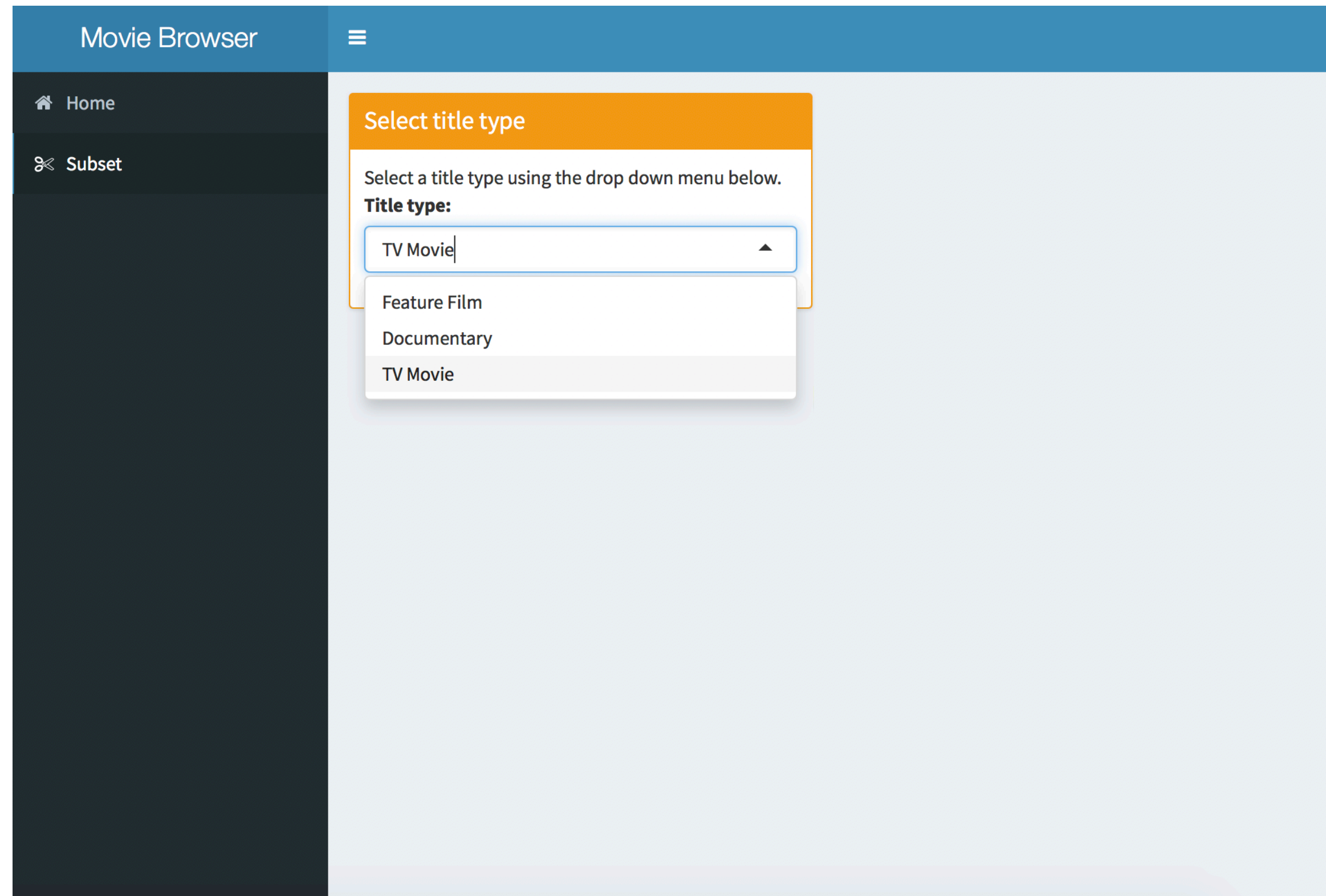


## SOLUTION





# dashboards/moviedash-03.R



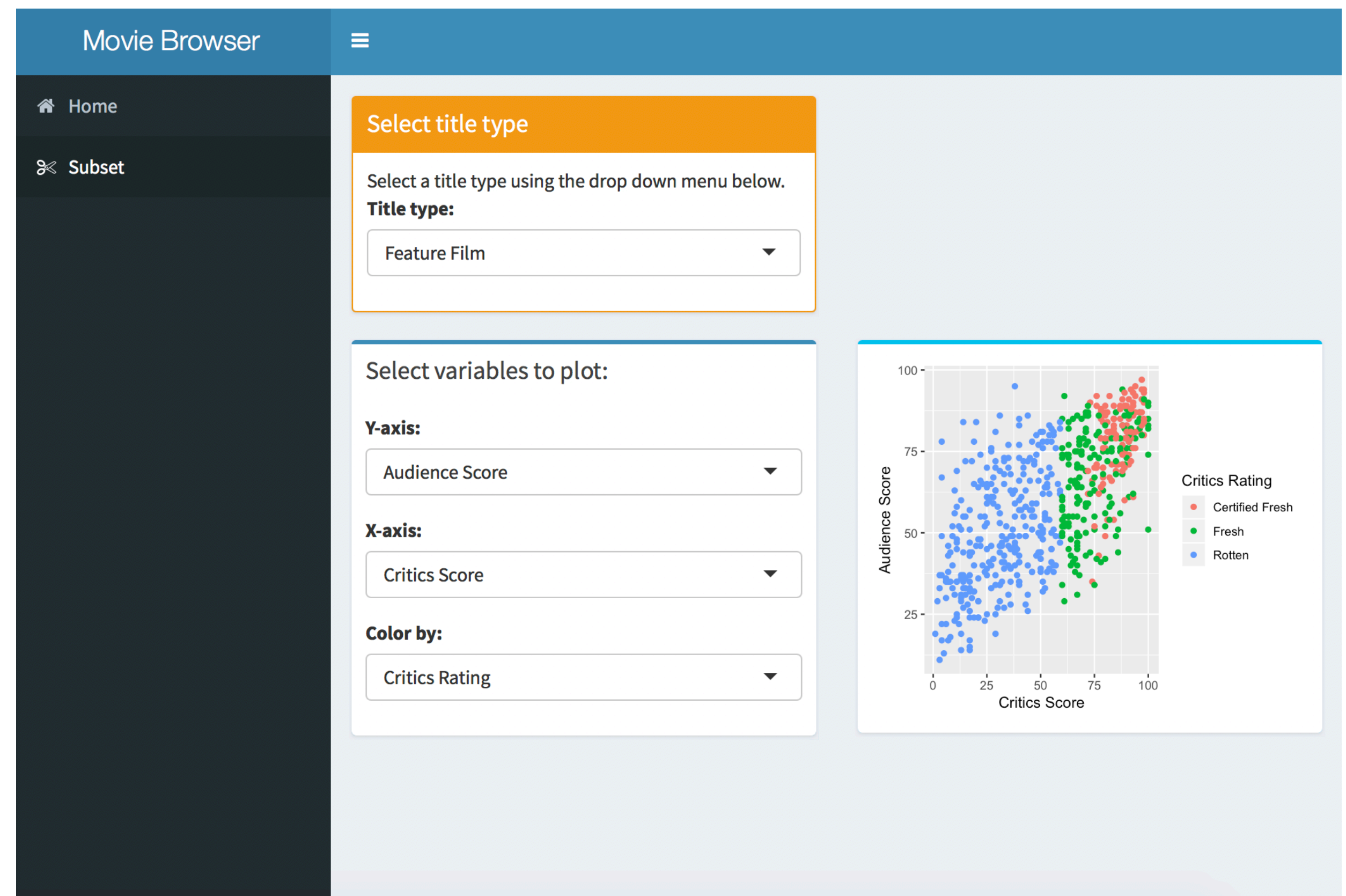
## DEMO





# Your turn

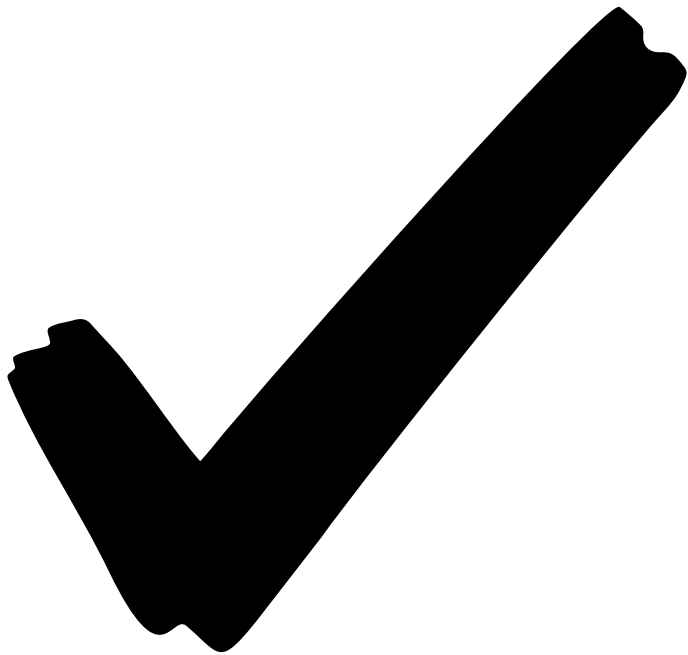
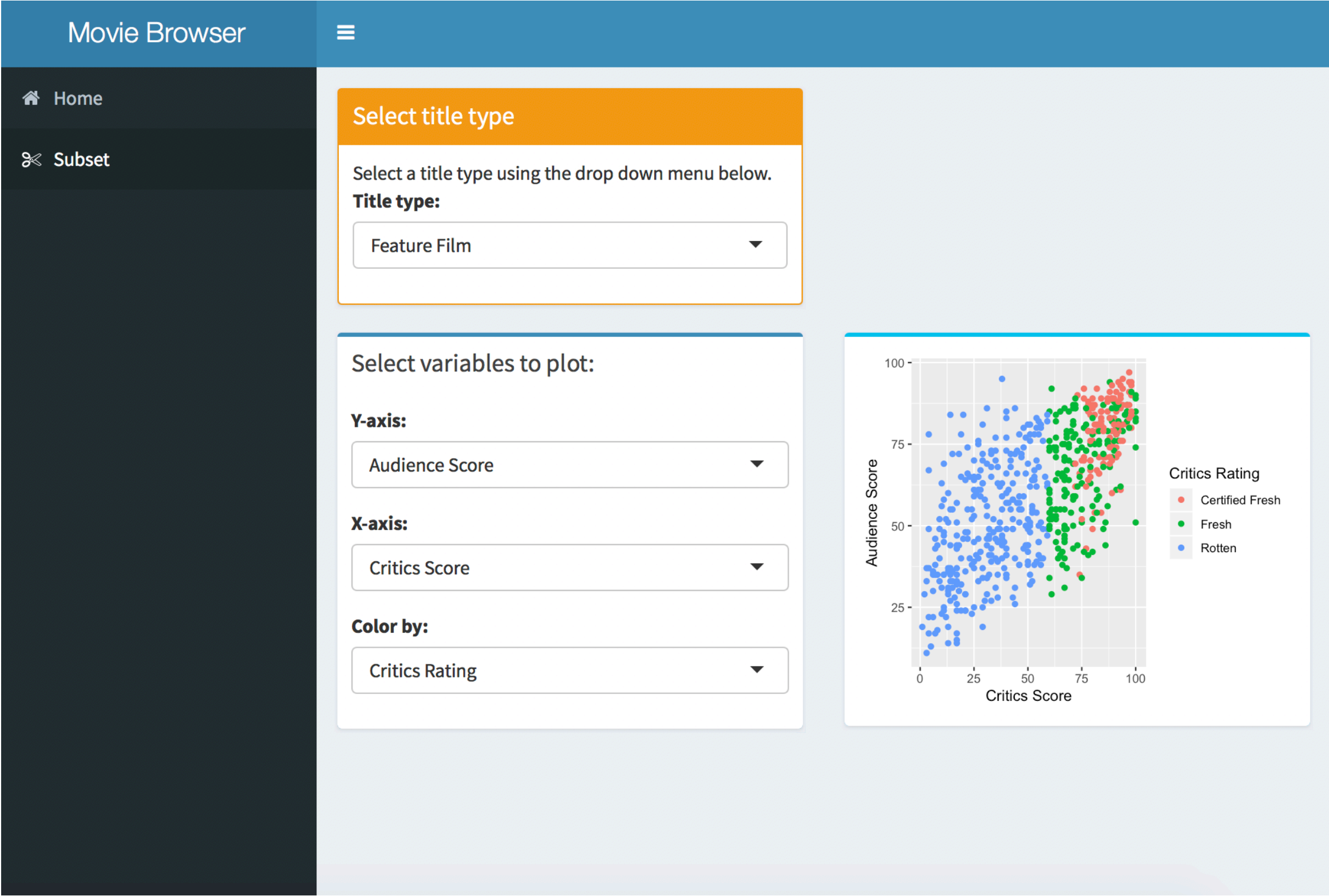
- Start with dashboards/  
moviedash-04.R
- Fill in the blanks to achieve the  
look / functionality on the right
- You can re-use code from  
earlier



10<sub>m</sub> 00<sub>s</sub>



dashboards/moviedash-05.R

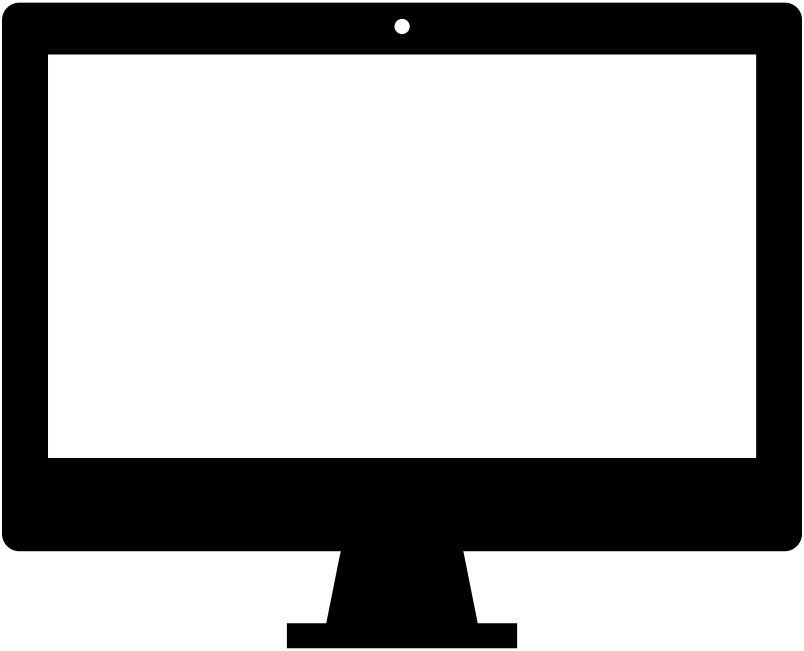
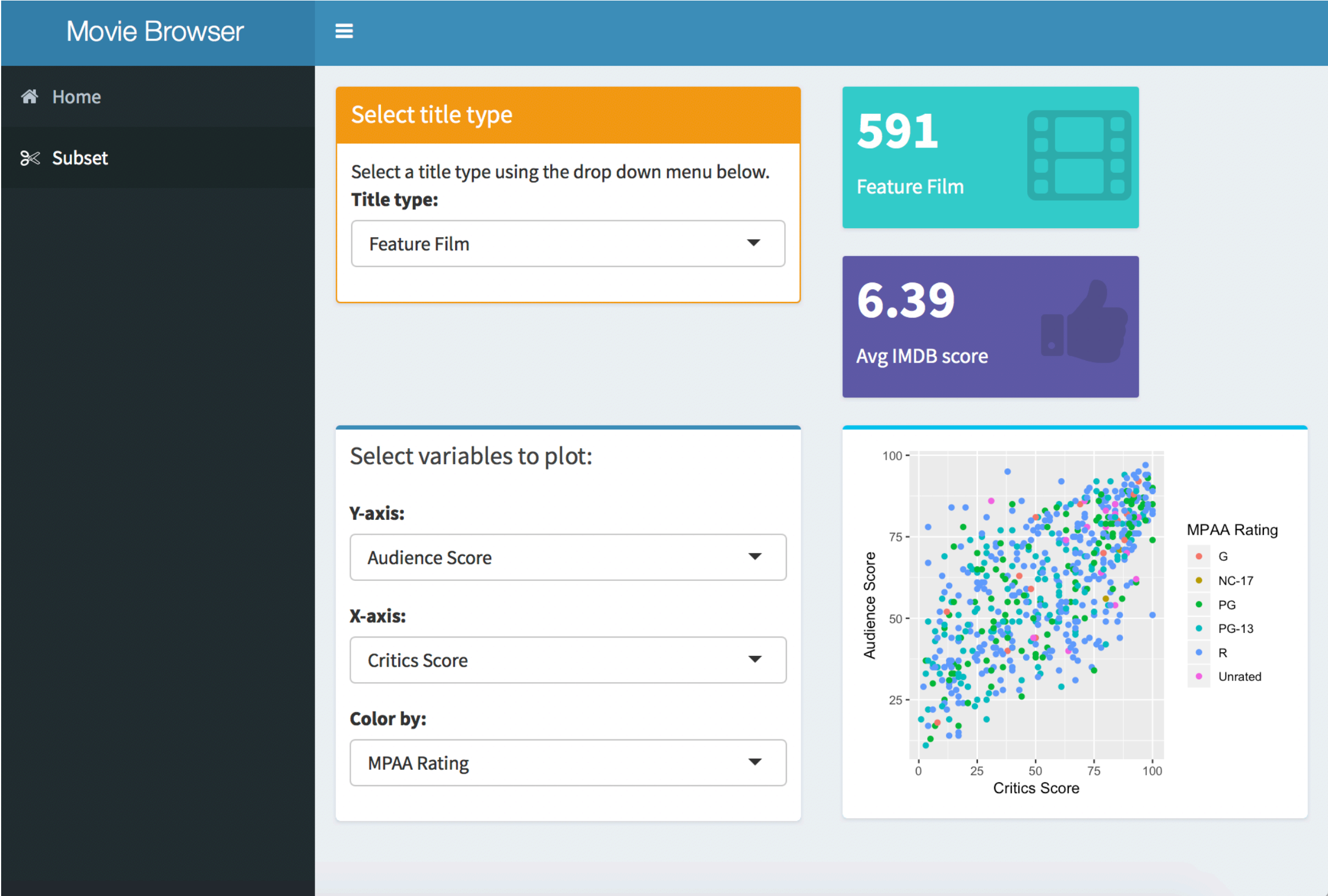


SOLUTION





# dashboards/moviedash-06.R



DEMO





# flexdashboard



# Your turn

- `library(flexdashboard)`
- File → New file → R Markdown → From Template
- Create three plots that go in each of the panes using built-in R datasets or any data we have used in the workshop (or your own data)



5<sub>m</sub> 00<sub>s</sub>



# Your turn

- Open `flexdashboards/flexdash-01.Rmd`
- How is it different than Shiny apps and dashboards we have been building so far, how is it similar?
- Make a change to the layout of the dashboard, see <http://rmarkdown.rstudio.com/flexdashboard/using.html#layout> for help
- Change the theme of the dashboard, see <http://rmarkdown.rstudio.com/flexdashboard/using.html#appearance> for help



5<sub>m</sub> 00<sub>s</sub>





# Shiny documents

- Add `runtime: shiny` to header.
- Add `inputs` in code chunks.
- Add `renderXyz` functions in code chunks.
  - No need for `output$x <-` assignment, or for `xyzOutput` functions.



# Your turn

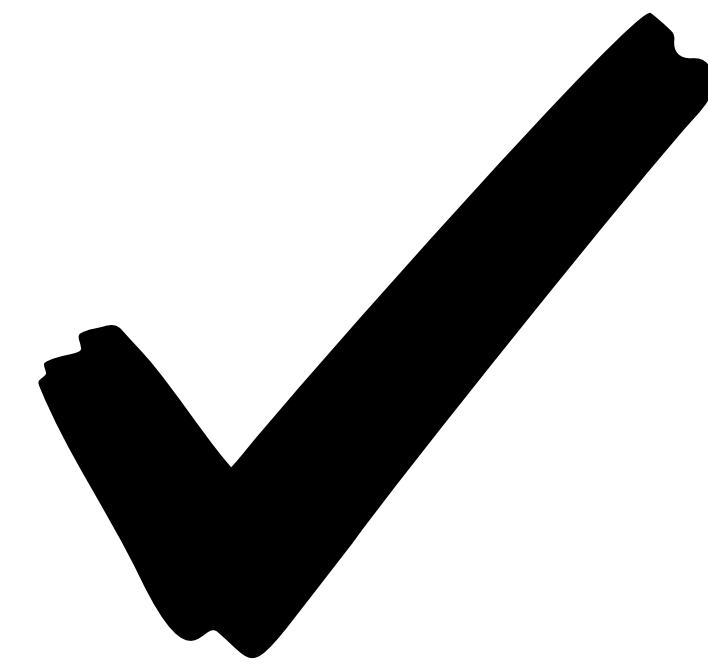
- Continue working on `flexdashboards/flexdash-01.Rmd`
- Add another UI widget, a `radioButton`, that allows the user to select whether the plot used to visualize the distribution of weight should be histogram or a violin plot



3<sub>m</sub> 00<sub>s</sub>



# Solution to the previous exercise



`flexdashboards/flexdash-02.Rmd`

## SOLUTION





# Your turn

- Recreate the app dashboards/moviedash-06 (or as much of it as you can) in the flexdashboard.

