Shiny

Dayu Tie

2024-11-12

```
library(shiny)
```

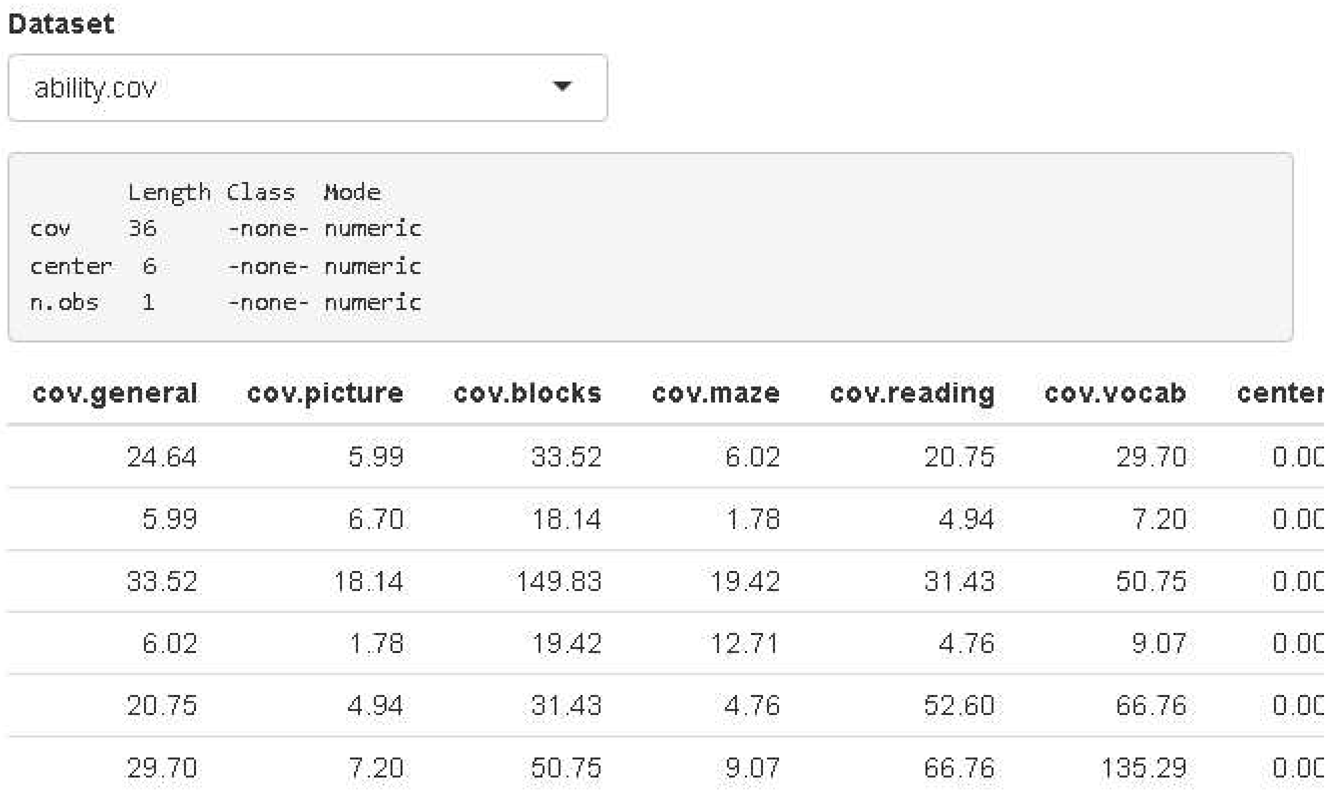
```
## Warning: 程序包'shiny'是用R版本4.4.2 来建造的
```

```
#install.packages('shinydashboard')
```

Hadley_1

```
ui <- fluidPage(
  selectInput("dataset", label = "Dataset", choices = ls("package:datasets")),
  verbatimTextOutput("summary"),
  tableOutput("table")
)



server <- function(input, output, session) {
  output$summary <- renderPrint({
    dataset <- get(input$dataset, "package:datasets")
    summary(dataset)
  })

  output$table <- renderTable({
    dataset <- get(input$dataset, "package:datasets")
    dataset
  })
}

shinyApp(ui, server)
```

```
##
## Listening on http://127.0.0.1:5802
```


Dataset

ability.cov

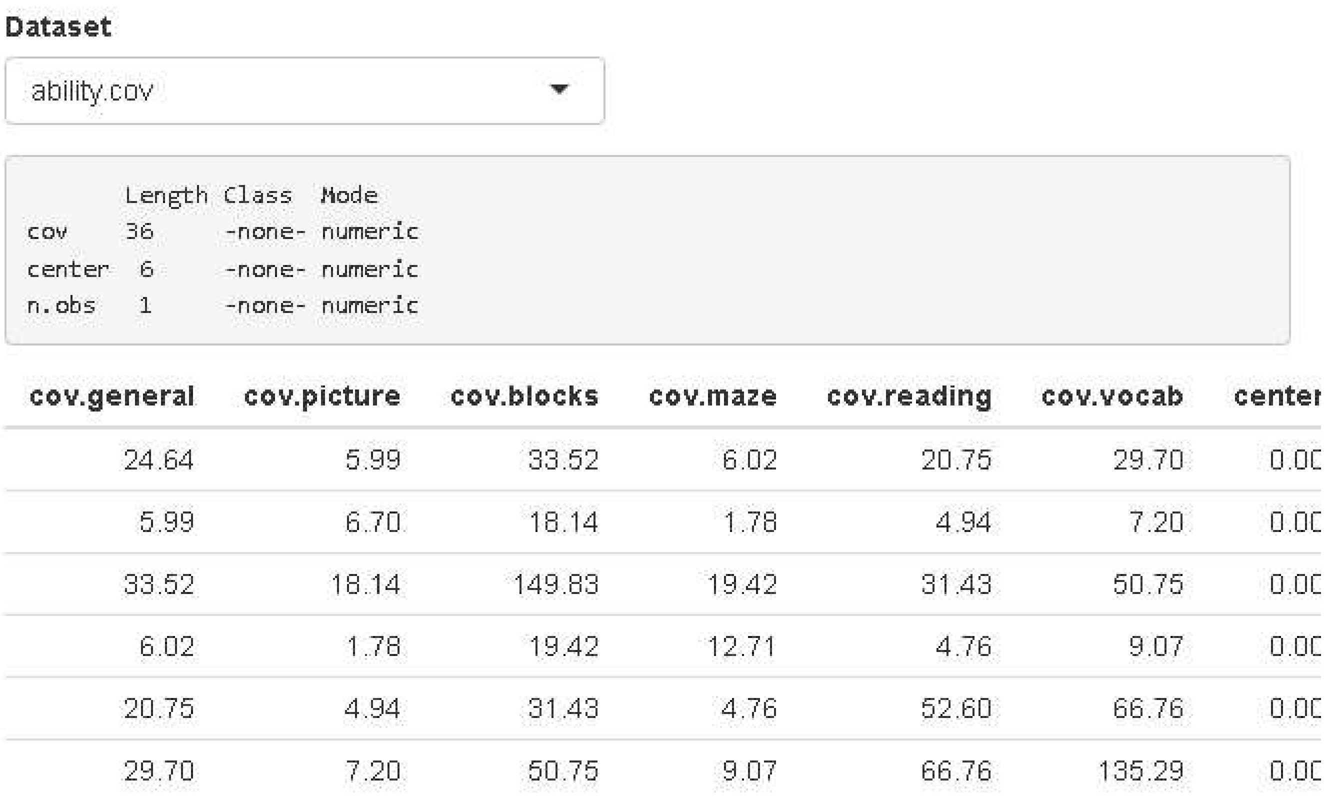
```
       Length Class  Mode
cov    36     -none- numeric
center  6     -none- numeric
n.obs   1     -none- numeric
```

cov.general	cov.picture	cov.blocks	cov.maze	cov.reading	cov.vocab	center
24.64	5.99	33.52	6.02	20.75	29.70	0.0
5.99	6.70	18.14	1.78	4.94	7.20	0.0
33.52	18.14	149.83	19.42	31.43	50.75	0.0
6.02	1.78	19.42	12.71	4.76	9.07	0.0
20.75	4.94	31.43	4.76	52.60	66.76	0.0
29.70	7.20	50.75	9.07	66.76	135.29	0.0

Hadley_2

```
ui <- fluidPage(
  selectInput("dataset", label = "Dataset", choices = ls("package:datasets")),
  verbatimTextOutput("summary"),
  tableOutput("table")
)

server <- function(input, output, session) {
  # Create a reactive expression
  dataset <- reactive({
    get(input$dataset, "package:datasets")
  })

  output$summary <- renderPrint({
    # Use a reactive expression by calling it like a function
    summary(dataset())
  })

  output$table <- renderTable({
    dataset()
  })
}
shinyApp(ui, server)
```

```
##
## Listening on http://127.0.0.1:8738
```



```
       Length Class  Mode
cov    36     -none- numeric
center  6     -none- numeric
n.obs   1     -none- numeric
```

cov.general	cov.picture	cov.blocks	cov.maze	cov.reading	cov.vocab	center
24.64	5.99	33.52	6.02	20.75	29.70	0.00
5.99	6.70	18.14	1.78	4.94	7.20	0.00
33.52	18.14	149.83	19.42	31.43	50.75	0.00
6.02	1.78	19.42	12.71	4.76	9.07	0.00
20.75	4.94	31.43	4.76	52.60	66.76	0.00
29.70	7.20	50.75	9.07	66.76	135.29	0.00

#Hadley_1 demonstrates the basic functionality with duplicated dataset retrieval, #while Hadley_2 showcases the use of reactive programming to optimize the app by eliminating redundancy.

2.3.5

1.Which of and should each of the following render functions be paired with?textOutput()verbatimTextOutput()

A.renderPrint(summary(mtcars))

B.renderText(“Good morning!”)

C.renderPrint(t.test(1:5, 2:6))

D.renderText(str(lm(mpg ~ wt, data = mtcars)))

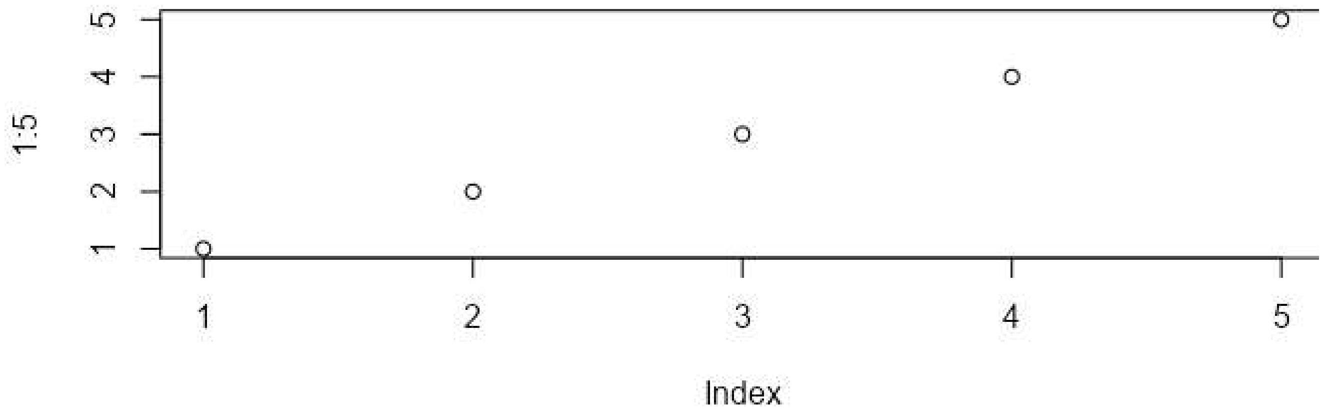
A: verbatimTextOutput() B: textOutput() C: verbatimTextOutput() D: textOutput()

2.

```
#install.packages('shiny')
library('shiny')
library(shiny)

ui <- fluidPage(
  textOutput("plot_description"),
  plotOutput("plot", height = "300px", width = "700px")
)

server <- function(input, output, session) {
  output$plot <- renderPlot(
    plot(1:5), res = 96)

  output$plot_description <- renderText({
    "This scatter plot displays five random points, showing values from 1 to 5 along both axe
s."
  })
}

shinyApp(ui, server)
```

```
##
## Listening on http://127.0.0.1:7103
```

This scatter plot displays five random points, showing values from 1 to 5 along both axes.



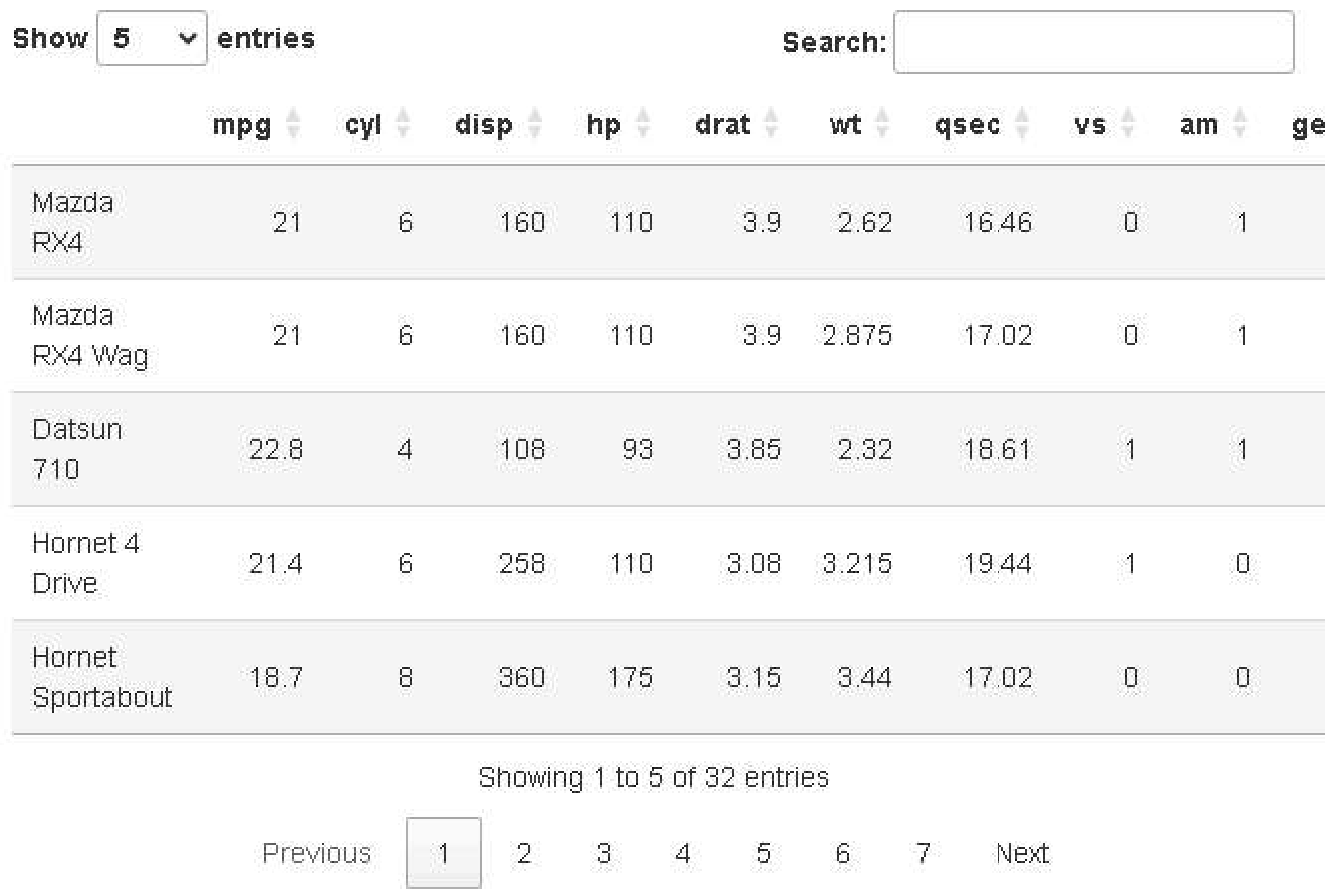
3.

```
ui <- fluidPage(
  dataTableOutput("table")
)
```

```
## `shiny::dataTableOutput()` is deprecated as of shiny 1.8.1.
## Please use `DT::DTOutput()` instead.
## Since you have a suitable version of DT (>= v0.32.1), shiny::dataTableOutput() will automati
cally use DT::DTOutput() under-the-hood.
## If this happens to break your app, set `options(shiny.legacy.datatable = TRUE)` to get the l
egacy datatable implementation (or `FALSE` to squelch this message).
## See <https://rstudio.github.io/DT/shiny.html> for more information.
```

```
server <- function(input, output, session) {
  output$table <- DT::renderDT(mtcars, options = list(pageLength = 5))
}

shinyApp(ui, server)
```

```
##
## Listening on http://127.0.0.1:4790
```



Show 5 entries Search:

	mpg	cyl	disp	hp	drat	wt	qsec	vs	am	ge
Mazda RX4	21	6	160	110	3.9	2.62	16.46	0	1	
Mazda RX4 Wag	21	6	160	110	3.9	2.875	17.02	0	1	
Datsun 710	22.8	4	108	93	3.85	2.32	18.61	1	1	
Hornet 4 Drive	21.4	6	258	110	3.08	3.215	19.44	1	0	
Hornet Sportabout	18.7	8	360	175	3.15	3.44	17.02	0	0	

Showing 1 to 5 of 32 entries

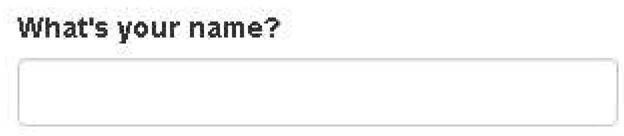
Previous 1 2 3 4 5 6 7 Next

3.3.6 1.

```
ui <- fluidPage(
  textInput("name", "What's your name?"),
  textOutput("greeting")
)
```

```
server1 <- function(input, output, session) {
  output$greeting <- renderText({
    paste0("Hello ", input$name)
  })
}
server2 <- function(input, output, session) {
  output$greeting <- renderText({
    paste0("Hello ", input$name)
  })
}
server3 <- function(input, output, session) {
  output$greeting <- renderText({
    paste0("Hello ", input$name)
  })
}
shinyApp(ui, server)
```

```
##
## Listening on http://127.0.0.1:3632
```



What's your name?

2. reactive graph1

input$ainput$b | | V V reactive(c) (c <- input$a + input$b) | V input$dreactive(e)(e < -c() + input$d) | | V V output$f (renderText(e()))

reactive graph2

input$x1input$x2 input$x3|||VVVreactive(x)(x < -input$x1 + input$x2 + input$x3)

input$y1input$y2 | | V V reactive(y) (y <- input$y1 + input$y2) | V output$z (renderText(x() / y()))

reactive graph3 input$ainput$b input$cinput$d | | | | V V V V reactive(a) reactive(b) reactive(c) reactive(d) (a <- input$a * 10)(b < -a() + input$b) (c <- b() / input$c)(d < -c()^input$d)

3. This code will fail because of a naming conflict. In R, range is the name of a base R function, so defining a reactive variable called range will lead to unexpected behavior or errors. It’s best to rename this reactive expression to avoid overriding the base function.

```
var <- reactive(df[[input$var]])
var_range <- reactive(range(var(), na.rm = TRUE))
```

4.8 1.

```
              ┌────────────────────────┐
              │ input$code     │
              └──────────┬─────────────┘
                   │
           ┌─────────────▼────────────────┐
           │ selected()          │
           │ (filters injuries   │
           │  based on prod_code)│
           └─────┬───────────────┬────────┘
             │        │
    ┌─────────────────▼────┐  ┌───────▼──────────────────┐
    │ output$diag    │  │ output$body_part│
    │ (table count   │  │ (table count    │
    │  by diag)      │  │  by body_part)  │
    └──────────────────────┘  └──────────────────────────┘
              │
              │
            ┌────▼───────────────────────┐
            │ output$location    │
            │ (table count       │
            │  by location)      │
            └────▲───────────────────────┘
              │
         ┌────────────┴───────────────────────┐
         │ summary()              │
         │ (count by age, sex,    │
         │  then joined with      │
         │  population)           │
         └──────────────┬─────────────────────┘
               │
        ┌──────────────▼────────────────┐
        │ output$age_sex          │
        │ (line plot by age,      │
        │  n per age-sex group)   │
        └─────────────────────────────────────┘



              ┌────────────────────────┐
              │ input$code     │
              └──────────┬─────────────┘
                   │
           ┌─────────────▼────────────────┐
           │ selected()          │
           │ (filters injuries   │
           │  based on prod_code)│
           └─────┬───────────────┬────────┘
             │        │
    ┌─────────────────▼────┐  ┌───────▼──────────────────┐
    │ output$diag    │  │ output$body_part│
    │ (table count   │  │ (table count    │
    │  by diag)      │  │  by body_part)  │
    └──────────────────────┘  └──────────────────────────┘
              │
```

```
                  |
              ┌───▼────────────────────┐
              | output$location   |
              | (table count      |
              |  by location)     |
              └───▲────────────────────┘
                  |
          ┌──────────┴─────────────────────────┐
          | summary()              |
          | (count by age, sex,    |
          |  then joined with      |
          |  population)           |
          └──────────────┬─────────────────────────┘
                    |
      ┌────────────────────▼─────────────────────────────────────────┐
      | input$y                                     |
      | (user choice: rate vs count)                |
      └────────────────────┬─────────────────────────────────────────┘
                    |
         ┌──────────────────▼──────────────────┐
         | output$age_sex          |
         | (conditional plot by    |
         |  age: rate or count     |
         |  based on input$y)      |
         └────────────────────────────────────────┘



                ┌────────────────────────┐
                | input$code     |
                └──────────┬─────────────┘
                       |
             ┌─────────────────▼─────────────────┐
             | selected()             |
             | (filters injuries      |
             |  based on prod_code)   |
             └─────┬───────────────────┬───────────┘
                   |           |
     ┌──────────────────────▼─────┐   ┌────────▼──────────────────────┐
     | output$diag      |  | output$body_part |
     | (table count     |  | (table count     |
     |  by diag)        |  |  by body_part)   |
     └────────────────────────────┘   └───────────────────────────────┘
                  |
                  |
              ┌───▼────────────────────┐
              | output$location   |
              | (table count      |
              |  by location)     |
              └───▲────────────────────┘
                  |
          ┌──────────┴─────────────────────────┐
          | summary()              |
          | (count by age, sex,    |
          |  then joined with      |
     │   population)          │
     └────────────┬─────────────────┘
                  │
    ┌─────────────▼─────────────────┐
    │  output$age_sex        │
    │  (conditional plot by  │
    │   age: rate or count)  │
    └─────────────┬─────────────────┘
                  │
    ┌─────────────▼─────────────────┐
    │  input$story           │
    └─────────────┬─────────────────┘
                  │
   ┌──────────────▼──────────────────┐
   │  narrative_sample()     │
   │  (gets new narrative    │
   │   on button click)      │
   └──────────────┬──────────────────┘
                  │
     ┌───────────▼───────────────────┐
     │  output$narrative       │
     │  (displays current      │
     │   narrative text)       │
     └───────────────────────────────┘
```

2. If you flip fct_infreq() and fct_lump(), the code will lump all values first, then order by frequency. This would lead to a less accurate table where less common factors may end up lumped with more common ones, affecting the interpretability and accuracy of the summarized table.

3.

#column(4, sliderInput("num_rows", "Number of rows:", min = 1, max = 10, value = 5))

#output$diag <- renderTable(count_top(selected(), diag, n = input$num_rows), width = "100%") #output$body_part <- renderTable(count_top(selected(), body_part, n = input$num_rows), width = "100%") #output$location <- renderTable(count_top(selected(), location, n = input$num_rows), width = "100%")

4.

```
fluidRow(
  column(1, actionButton("prev_story", "Previous")),
  column(1, actionButton("next_story", "Next")),
  column(10, textOutput("narrative"))
)
```

Previous

Next

#narrative_index <- reactiveVal(1)

#observeEvent(input$next_story, { # current <- narrative_index() # narrative_index(min(current + 1, nrow(selected()))) #})

#observeEvent(input$prev_story, { # current <- narrative_index() # narrative_index(max(current - 1, 1)) #})

#output$narrative <- renderText({ # selected() %>% pull(narrative) %>% .[narrative_index()] #})