Fall 2021

Project 2

R Shiny App

Name: Ravleen Kaur

Class ID: 43

School ID: 1113138

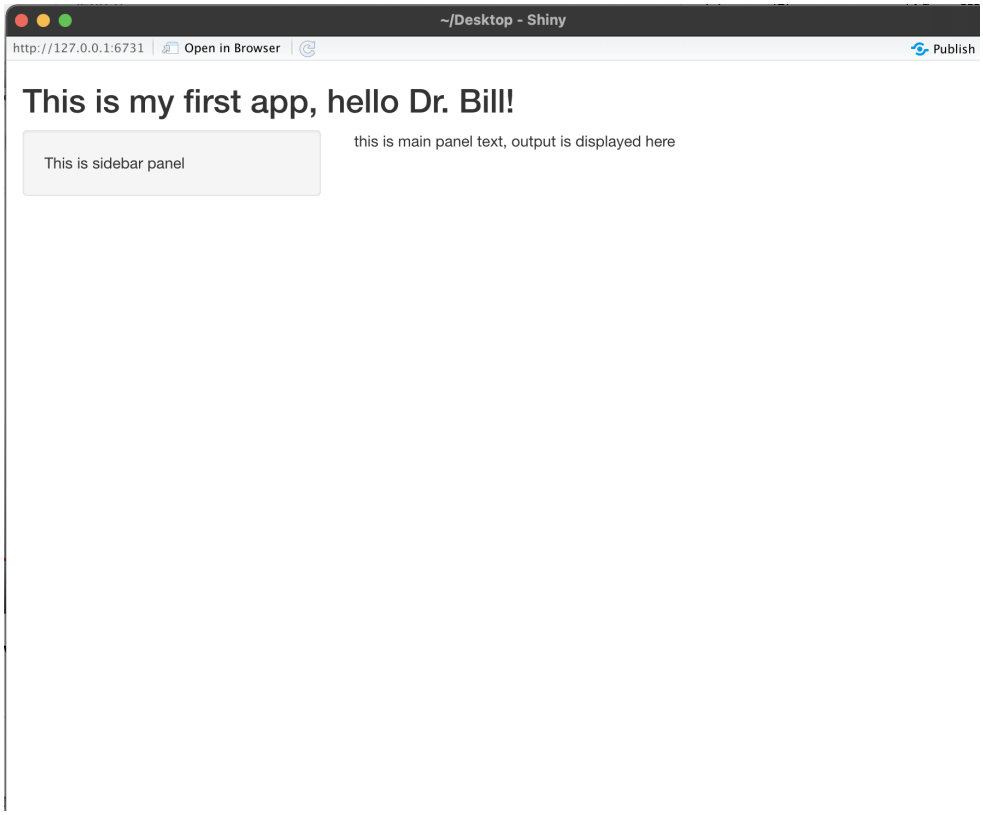
Course: Statistics for Data Science

Course ID: DTSC-620-W01

Downloading Shiny

```
> install.packages("shiny")
--- Please select a CRAN mirror for use in this session ---
also installing the dependencies ‘fs’, ‘rappdirs’, ‘Rcpp’, ‘digest’,
‘sass’, ‘jquerylib’, ‘httpuv’, ‘mime’, ‘jsonlite’, ‘xtable’, ‘fontaw
‘sourcetools’, ‘later’, ‘promises’, ‘crayon’, ‘rlang’, ‘fastmap’, ‘w
‘glue’, ‘bslib’, ‘cachem’, ‘ellipsis’, ‘lifecycle’

trying URL 'https://cran.case.edu/bin/macosx/contrib/4.1/fs_1.5.2.tg
Content type 'application/x-gzip' length 545943 bytes (533 KB)
==================================================
downloaded 533 KB

trying URL 'https://cran.case.edu/bin/macosx/contrib/4.1/rappdirs_0.
Content type 'application/x-gzip' length 45542 bytes (44 KB)
==================================================
downloaded 44 KB
```

App Video #2



server.r | ui.r

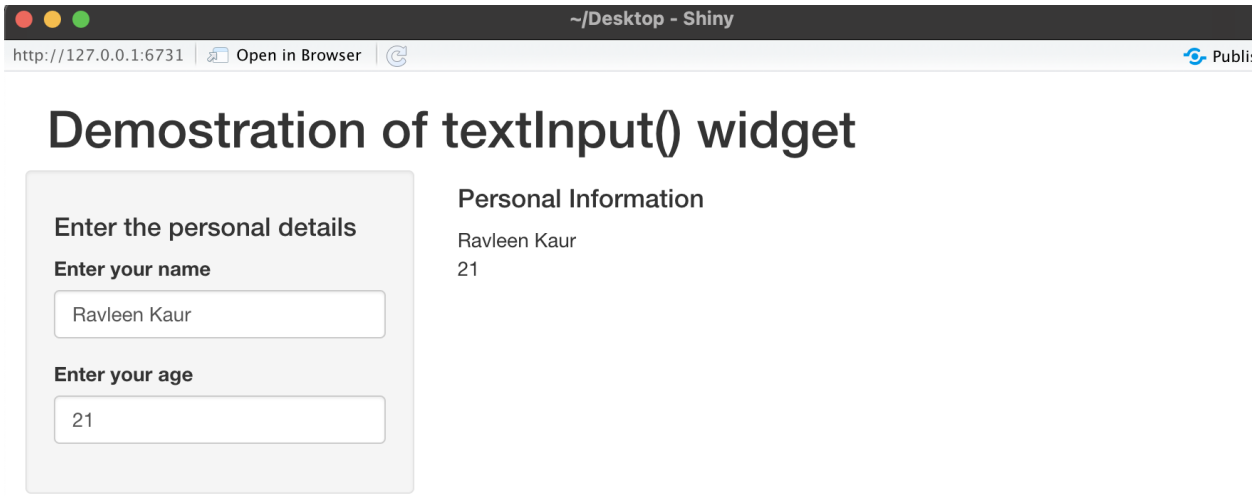
Source on Save

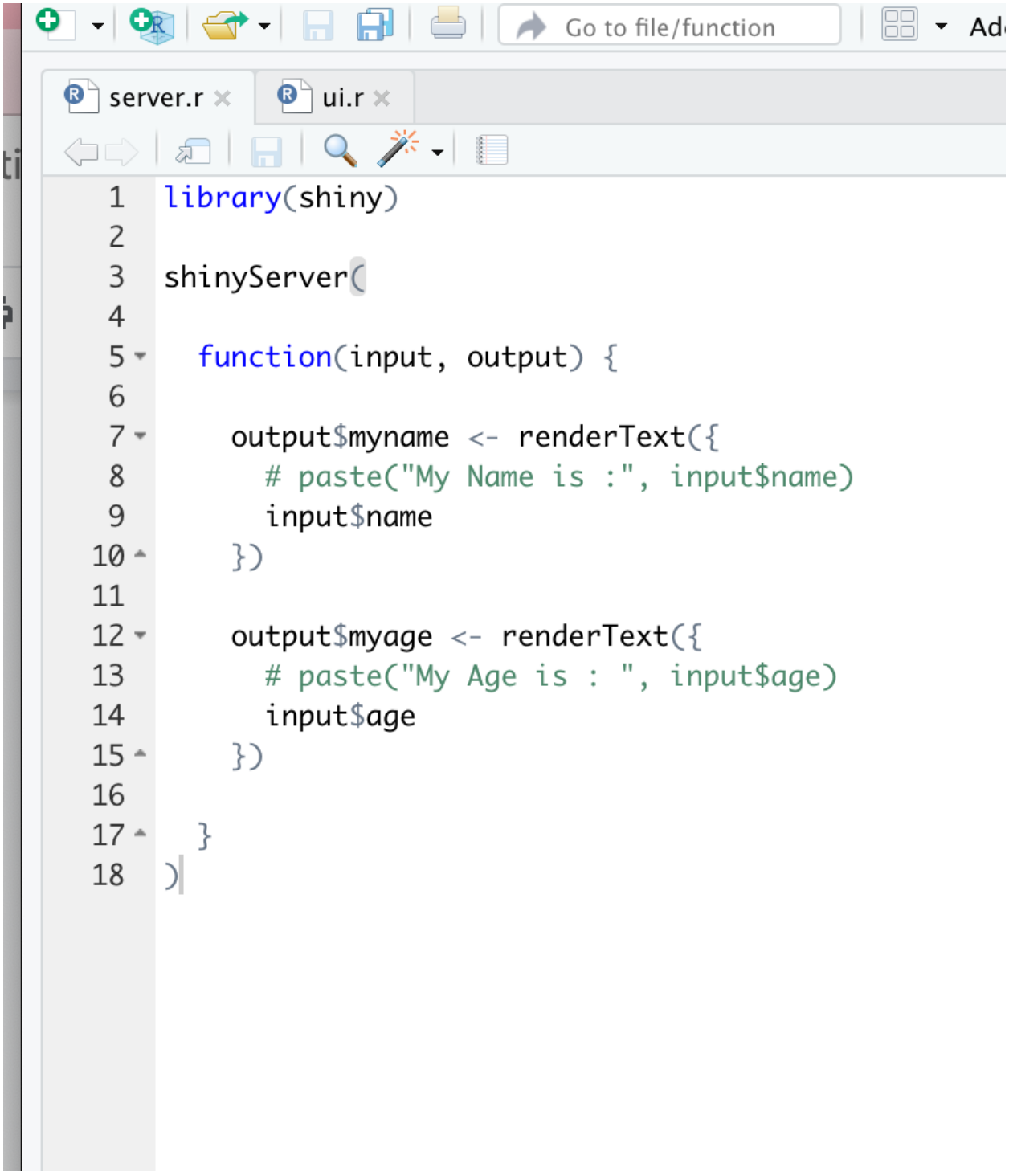
```
1
2  library(shiny)
3
4  shinyUI(fluidPage(
5
6    titlePanel(title= "This is my first app, hello Dr. Bill!"),
7    sidebarLayout(
8      sidebarPanel("This is sidebar panel"),
9      mainPanel("this is main panel text, output is displayed here")
10   )
11
12 )
13
14 )
15
16
```

server.r | ui.r

```
1  library(shiny)
2
3  shinyServer(
4
5    function(input, output) {
6
7    }
8  )
```

Video 3

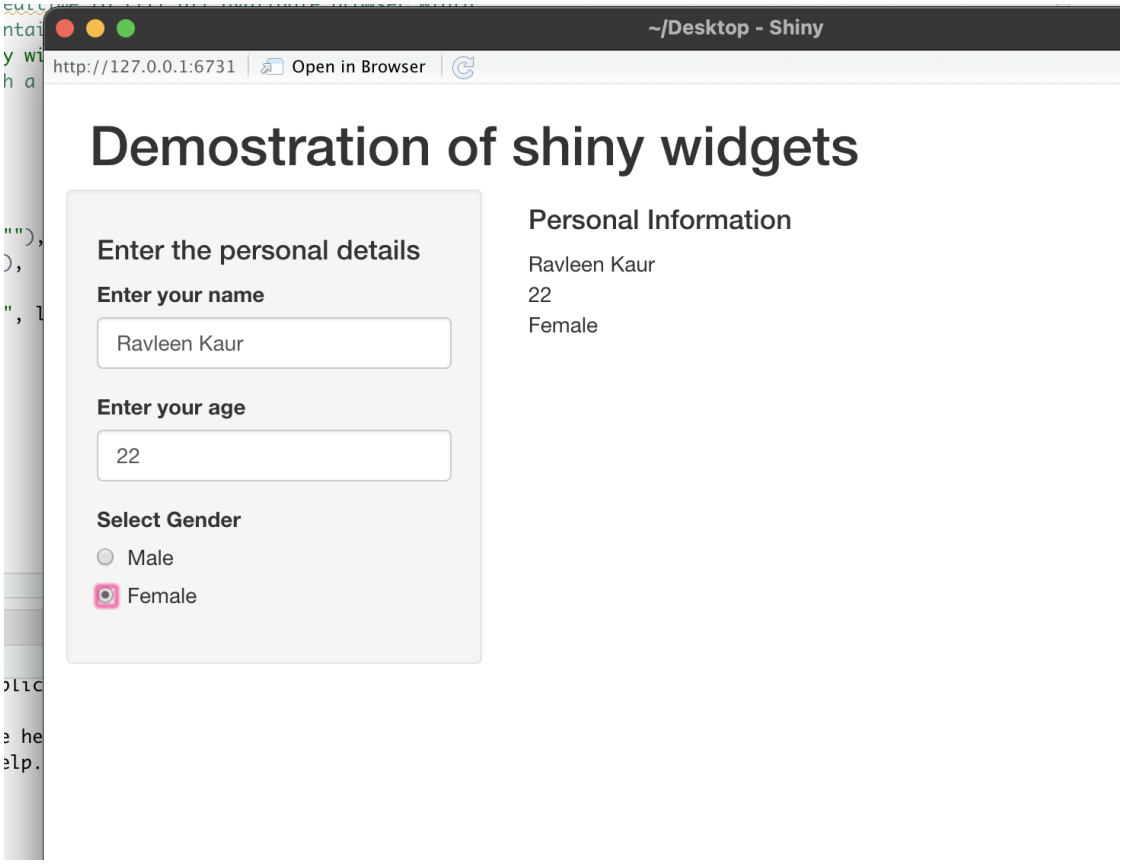

~/Desktop - Shiny
http://127.0.0.1:6731
Open in Browser
Demostration of textInput() widget
Enter the personal details
Enter your name
Ravleen Kaur
Enter your age
21
Personal Information
Ravleen Kaur
21


Go to file/function
server.r
ui.r
library(shiny)
shinyServer(
function(input, output) {
output$myname <- renderText({
# paste("My Name is :", input$name)
input$name
})
output$myage <- renderText({
# paste("My Age is : ", input$age)
input$age
})
}
)

server.r | ui.r

```r
# Load the shiny package
library(shiny)
# Define UI for the shiny application here
shinyUI(fluidPage(
  # fluid pages scale their components in realtime to
  # Header Panel : Create a header panel containing an
  headerPanel(title = "Demostration of textInput() wid
  # SidebarLayout():It creates a layout with a sidebar
  sidebarLayout(
    # Sidebar panel
    sidebarPanel(
      h4("Enter the personal details"),

      textInput("name", "Enter your name", ""),
      textInput("age", "Enter your age", "")

    ),
    # Main Panel
    mainPanel(
      h4("Personal Information"),

      textOutput("myname"),
      textOutput("myage")


```

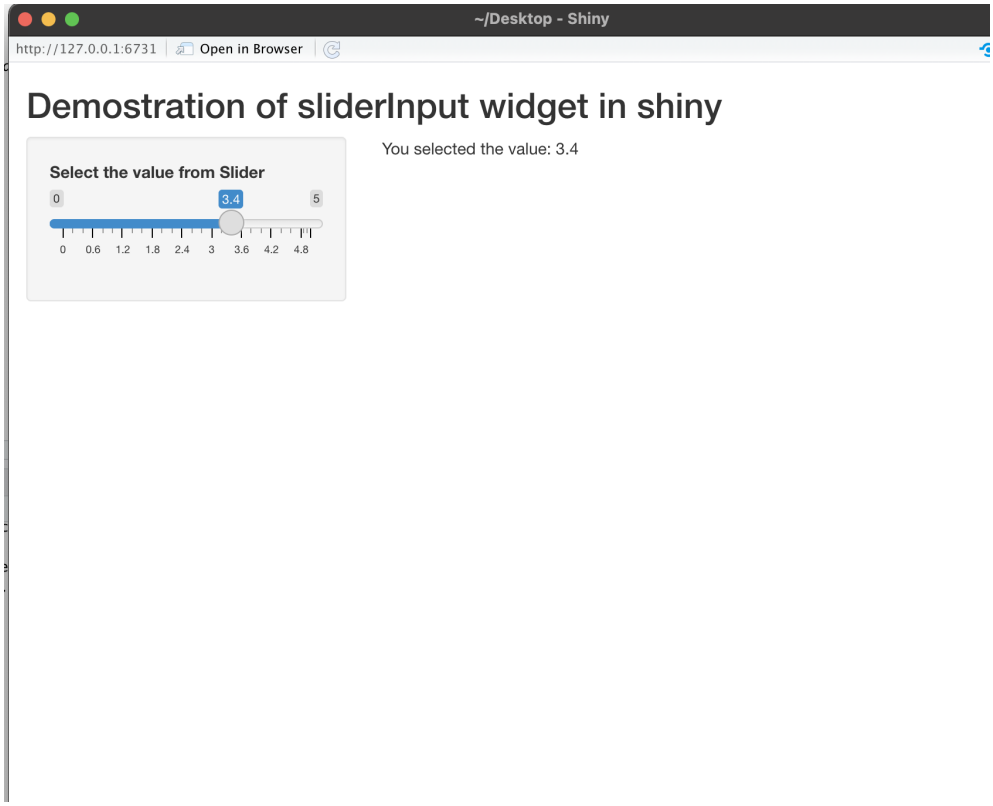
Video 4


~/Desktop - Shiny
http://127.0.0.1:6731
Open in Browser
Demostration of shiny widgets
Enter the personal details
Enter your name
Ravleen Kaur
Enter your age
22
Select Gender
Male
Female
Personal Information
Ravleen Kaur
22
Female

server.r × ui.r ×

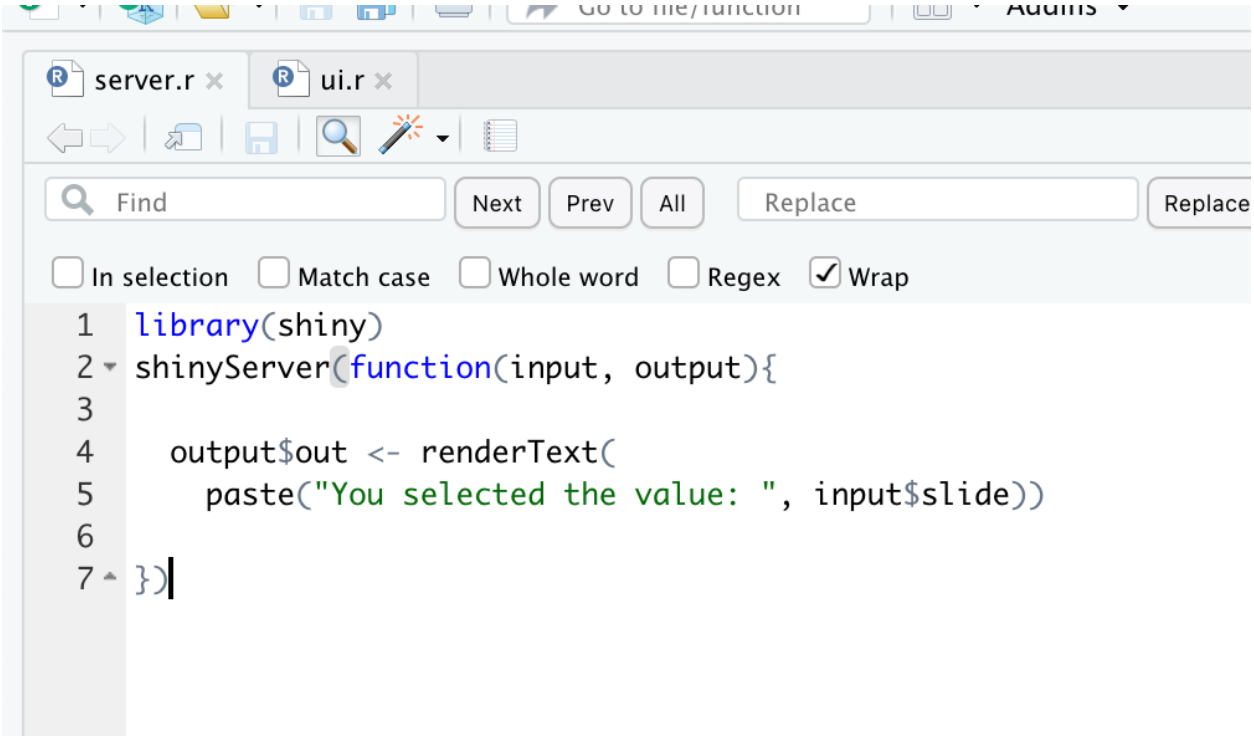
```r
# Load the shiny package
library(shiny)
# Define UI for the shiny application here
shinyUI(fluidPage(
  # fluid pages scale their components in realtime to fill all avail
  # Header Panel : Create a header panel containing an application t
  headerPanel(title = "Demostration of shiny widgets"),
  # SidebarLayout():It creates a layout with a sidebar and main area
  sidebarLayout(
    # Sidebar panel
    sidebarPanel(
      h4("Enter the personal details"),

      textInput("name", "Enter your name", ""),
      textInput("age", "Enter your age", ""),

      radioButtons("gender", "Select Gender", list("Male", "Female")

    ),
    # Main Panel
    mainPanel(
      h4("Personal Information"),

      textOutput("myname"),
      textOutput("myage"),
      textOutput("mygender")
```

Video 5: Slide Input


~/Desktop - Shiny
http://127.0.0.1:6731
Open in Browser
Demostration of sliderInput widget in shiny
Select the value from Slider
0
3.4
5
0
0.6
1.2
1.8
2.4
3
3.6
4.2
4.8
You selected the value: 3.4

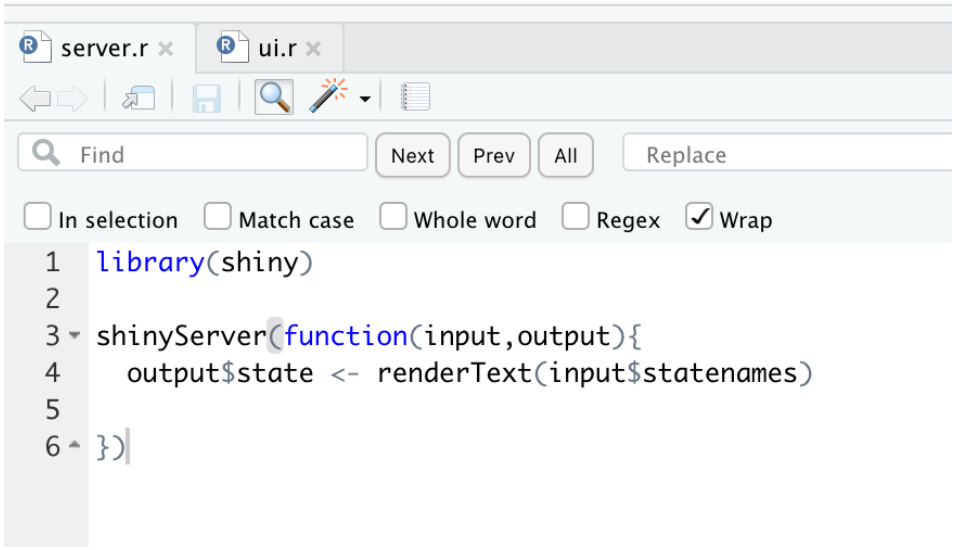
server.r | ui.r

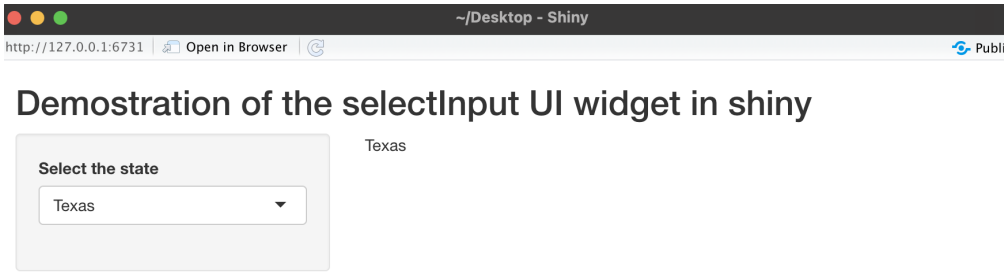
```r
library(shiny)

shinyUI(fluidPage(
  titlePanel("Demostration of sliderInput widget in shiny"),
  sidebarLayout(
    sidebarPanel(
      sliderInput("slide", "Select the value from Slider", min = 0, max=5, value=2, step=0.2)

    ),
    mainPanel(

      textOutput("out")
    )

  )
))
```

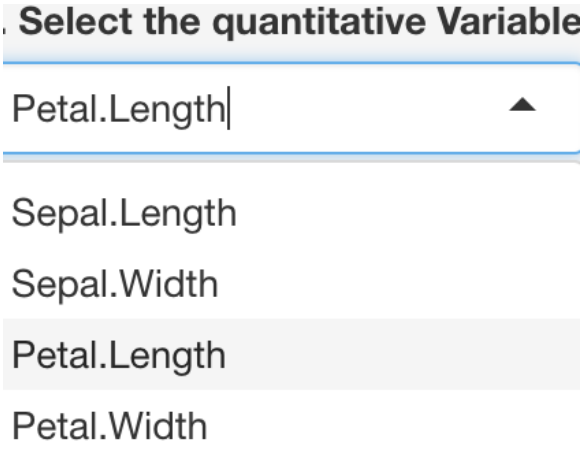



```r
library(shiny)
shinyServer(function(input, output){

  output$out <- renderText(
    paste("You selected the value: ", input$slide))

})
```

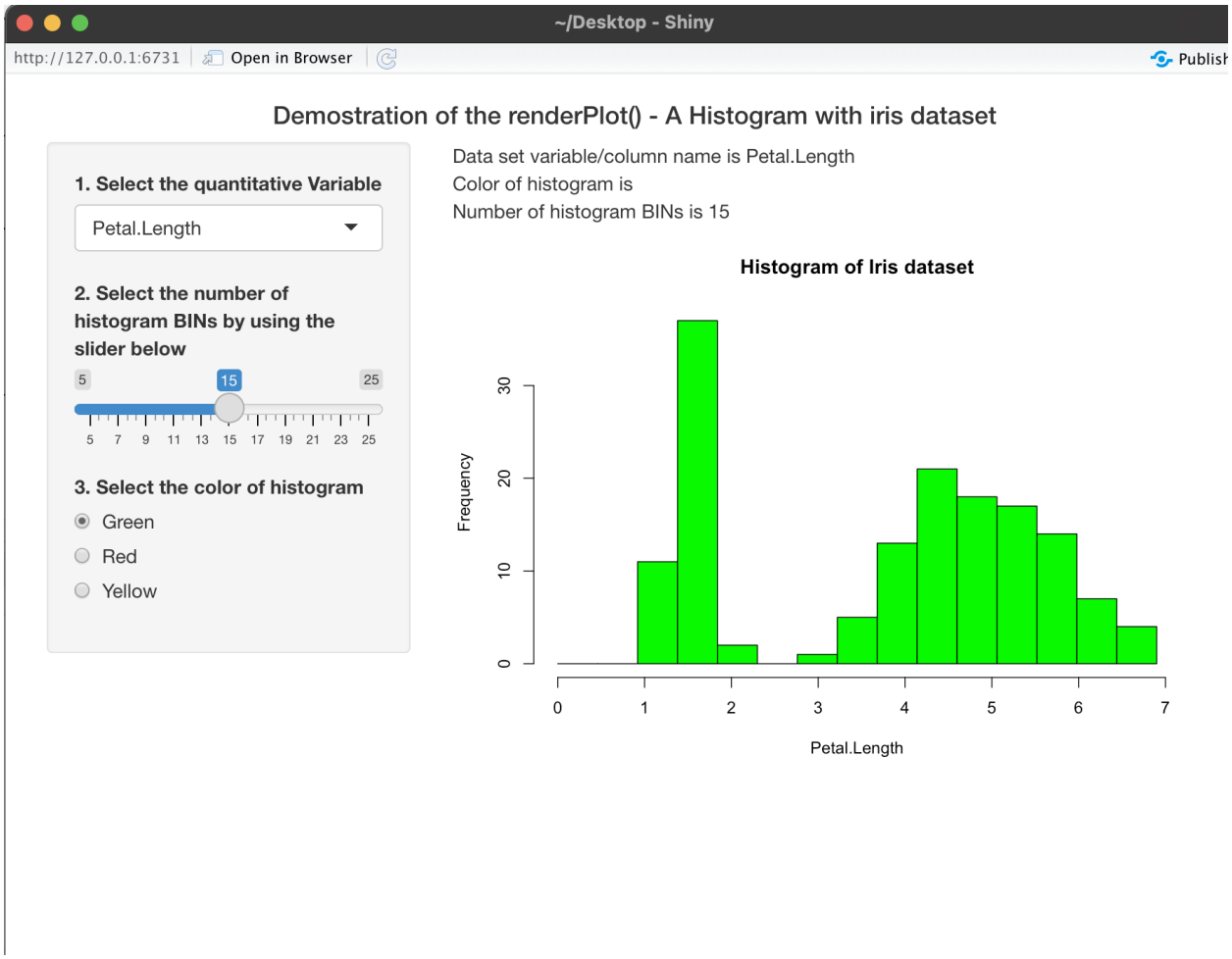
Shiny Video 6

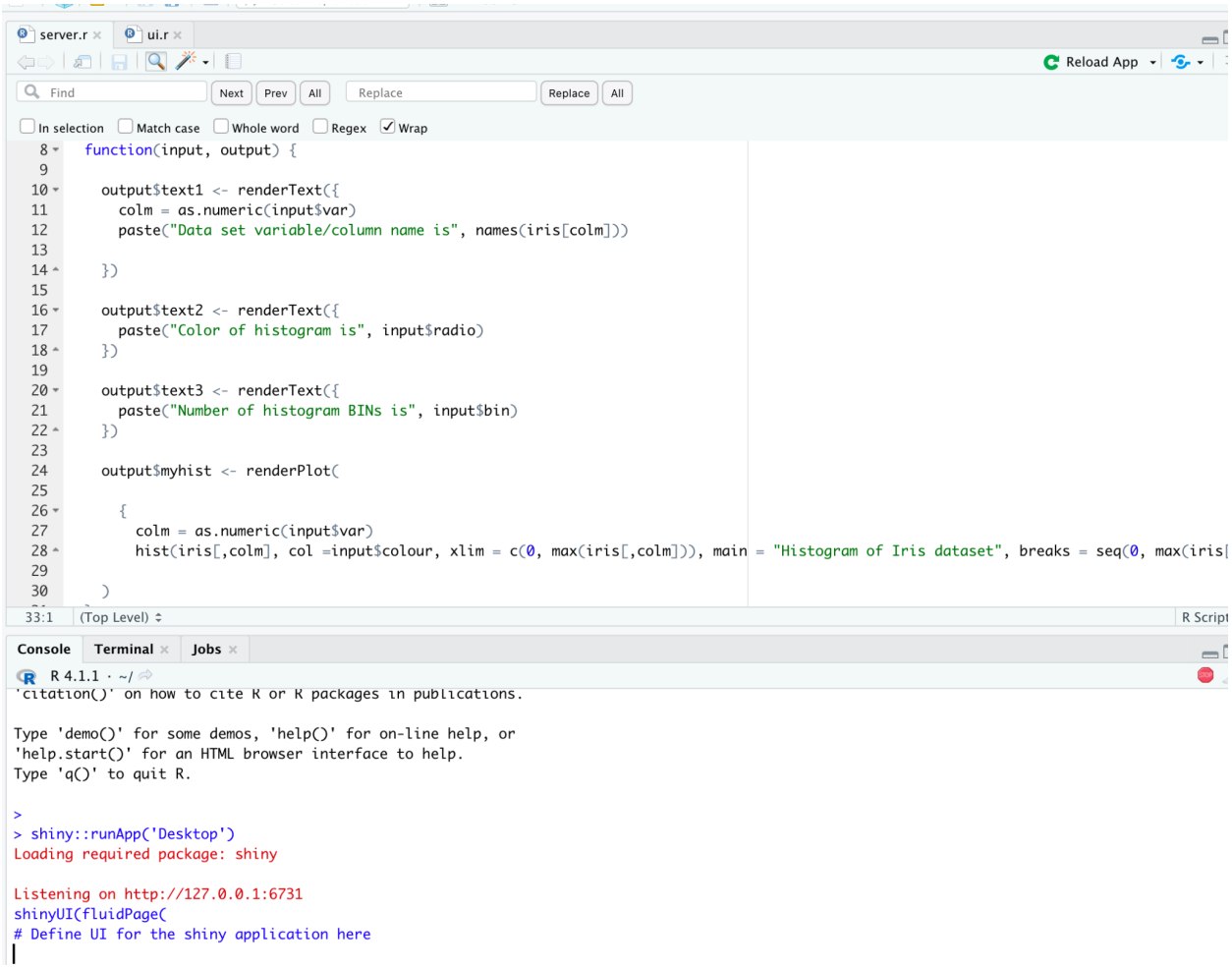






Shiny Video 7




server.r
ui.r
Reload App
Find
Next
Prev
All
Replace
Replace
All
In selection
Match case
Whole word
Regex
Wrap
function(input, output) {
output$text1 <- renderText({
colm = as.numeric(input$var)
paste("Data set variable/column name is", names(iris[colm]))
})
output$text2 <- renderText({
paste("Color of histogram is", input$radio)
})
output$text3 <- renderText({
paste("Number of histogram BINs is", input$bin)
})
output$myhist <- renderPlot(
{
colm = as.numeric(input$var)
hist(iris[,colm], col =input$colour, xlim = c(0, max(iris[,colm])), main = "Histogram of Iris dataset", breaks = seq(0, max(iris[
)
33:1
(Top Level)
R Script
Console
Terminal
Jobs
R 4.1.1 · ~/
'citation()' on how to cite R or R packages in publications.
Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.
>
> shiny::runApp('Desktop')
Loading required package: shiny
Listening on http://127.0.0.1:6731
shinyUI(fluidPage(
Define UI for the shiny application here

Shiny Video 8

server.r | ui.r — Reload App

```
1  library(shiny)
2  # Define UI for application
3  shinyUI(fluidPage(
4    # Header or Title Panel
5    titlePanel(title = h4("Iris Dataset", align="center")),
6    sidebarLayout(
7      # Sidebar panel
8      sidebarPanel(
9        selectInput("var", "1. Select the variable from the iris dataset", choices =c("Sepal.Length" = 1, "Sepal.Width" = 2, "Petal.Length" =
10       br(),
11       sliderInput("bins", "2. Select the number of BINs for histogram", min=5, max = 25, value=15),
12       br(),
13       radioButtons("color", "3. Select the colour of histogram", choices=c("Green", "Red", "Yellow"), selected ="Green")
14
15     ),
16
17     # Main Panel
18     mainPanel(
19       tabsetPanel(type="tab",
20                   tabPanel("Summary",verbatimTextOutput("sum")),
21                   tabPanel("Structure", verbatimTextOutput("str")),
22                   tabPanel("Data", tableOutput("data")),
23                   tabPanel("Plot", plotOutput("myhist"))
24
25       )
26
27   )
```

31:2 (Top Level) — R Script

~/Desktop - Shiny

http://127.0.0.1:6731 | Open in Browser | Publish

Iris Dataset

1. Select the variable from the iris dataset

Sepal.Length

2. Select the number of BINs for histogram

5 15 25

5 7 9 11 13 15 17 19 21 23 25

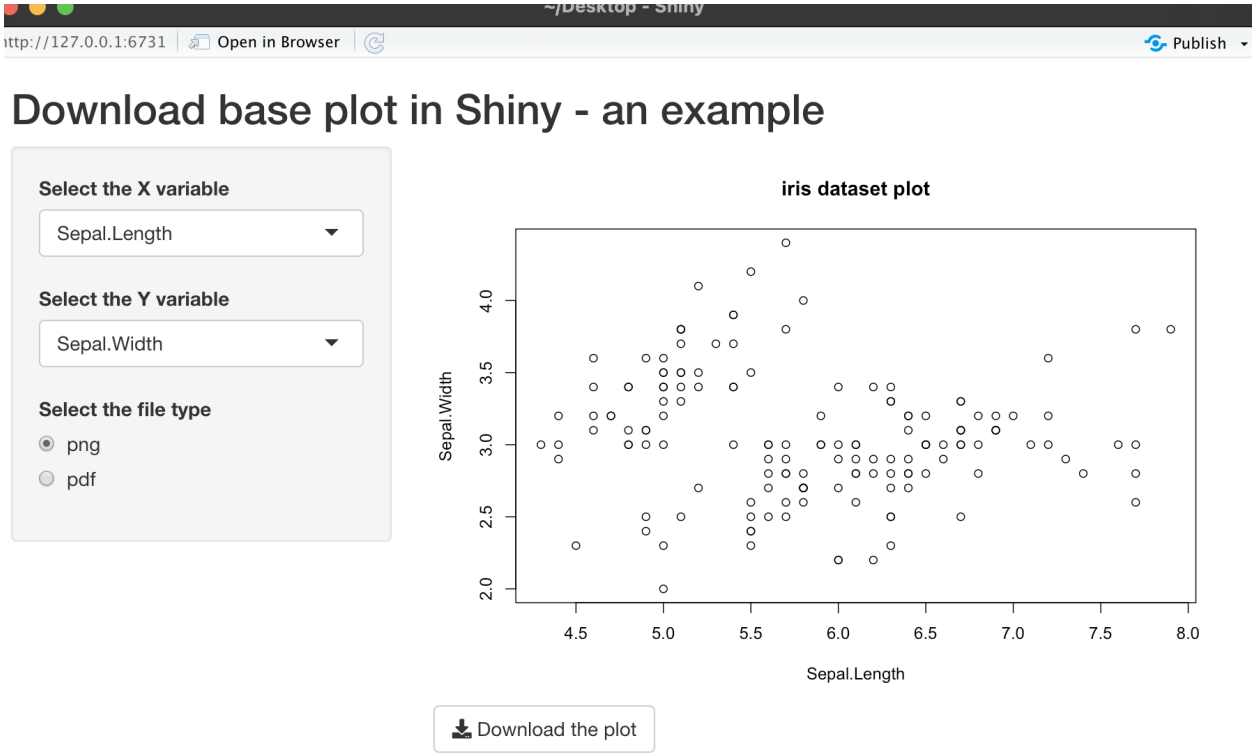
3. Select the colour of histogram

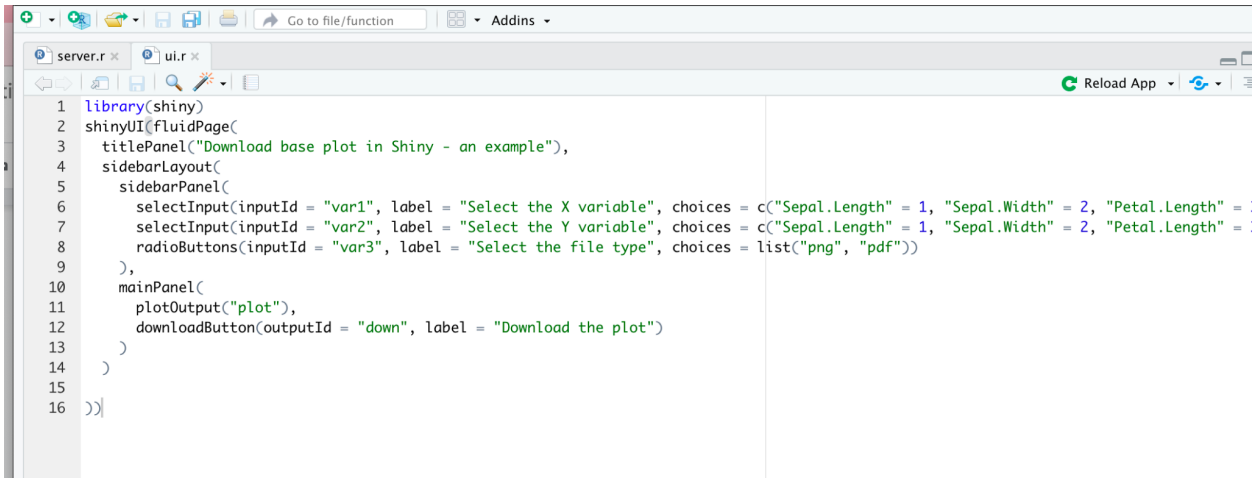
- (•) Green
- ( ) Red
- ( ) Yellow

Summary | Structure | Data | Plot

```
 Sepal.Length    Sepal.Width     Petal.Length    Petal.Width          S
Min.   :4.300   Min.   :2.000   Min.   :1.000   Min.   :0.100   setosa
1st Qu.:5.100   1st Qu.:2.800   1st Qu.:1.600   1st Qu.:0.300   versico
Median :5.800   Median :3.000   Median :4.350   Median :1.300   virgini
Mean   :5.843   Mean   :3.057   Mean   :3.758   Mean   :1.199
3rd Qu.:6.400   3rd Qu.:3.300   3rd Qu.:5.100   3rd Qu.:1.800
Max.   :7.900   Max.   :4.400   Max.   :6.900   Max.   :2.500
```

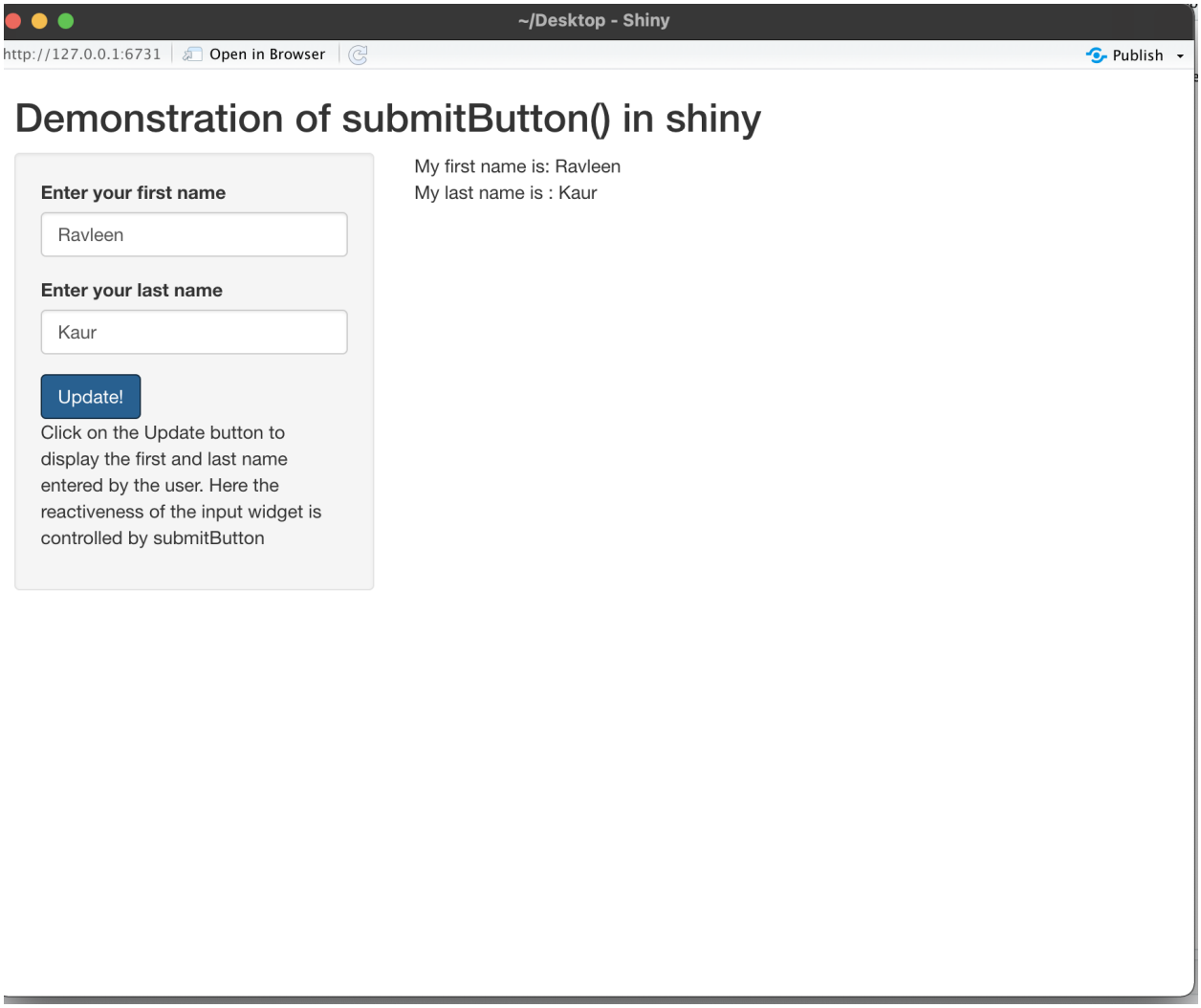
How to download a base plot using the downloadButton() and downloadHandler() functions.

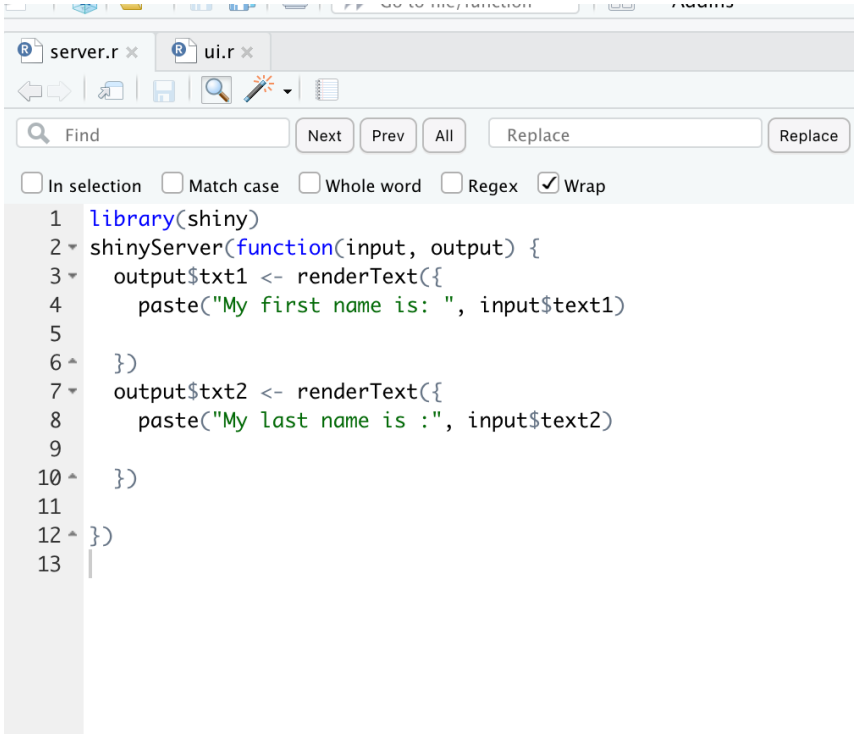




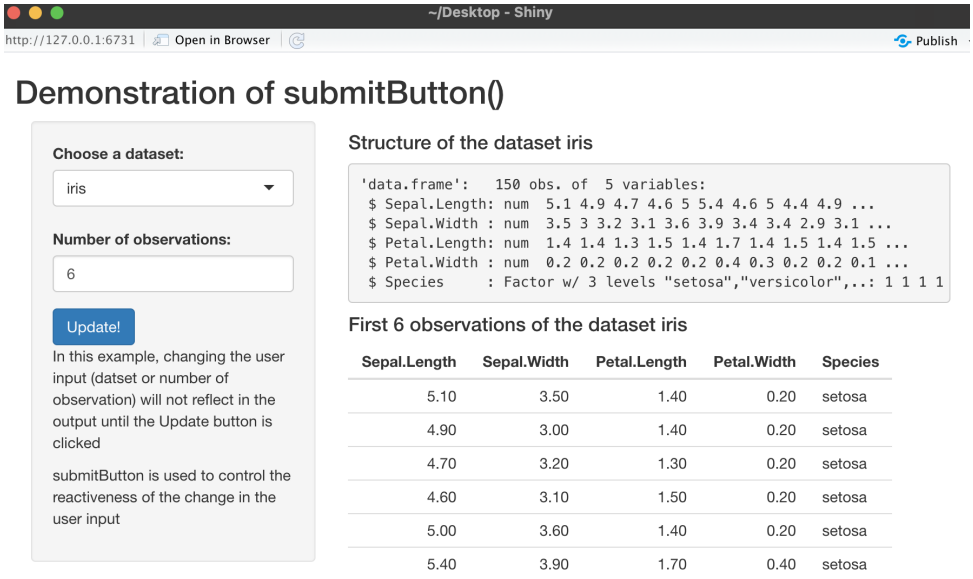
```r
library(shiny)
shinyServer(function(input,output){
  # x contains all the observations of the x variable selected by the user. X is a reactive function
  x <- reactive({
    iris[,as.numeric(input$var1)]
  })
  # x contains all the observations of the y variable selected by the user. Y is a reactive function
  y <- reactive({
    iris[,as.numeric(input$var2)]

  })
  # xl contains the x variable or column name of the iris dataset selected by the user
  xl <- reactive({
    names(iris[as.numeric(input$var1)])
  })
  # yl contains the y variable or column name of the iris dataset selected by the user
  yl <- reactive({
    names(iris[as.numeric(input$var2)])
  })

  # render the plot so could be used to display the plot in the mainPanel
  output$plot <- renderPlot({
    plot(x=x(), y=y(), main = "iris dataset plot", xlab = xl(), ylab = yl())
```

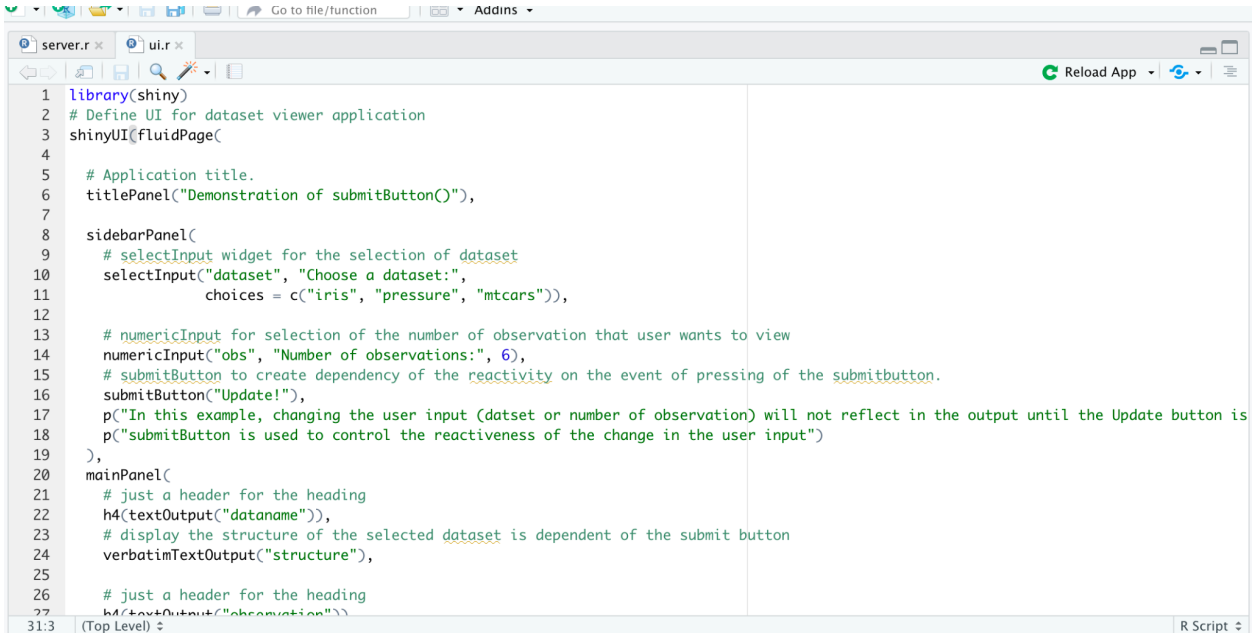
The use of submitButton() in shiny:


~/Desktop - Shiny
http://127.0.0.1:6731
Open in Browser
Publish
Demonstration of submitButton() in shiny
Enter your first name
Ravleen
Enter your last name
Kaur
Update!
Click on the Update button to display the first and last name entered by the user. Here the reactiveness of the input widget is controlled by submitButton
My first name is: Ravleen
My last name is : Kaur

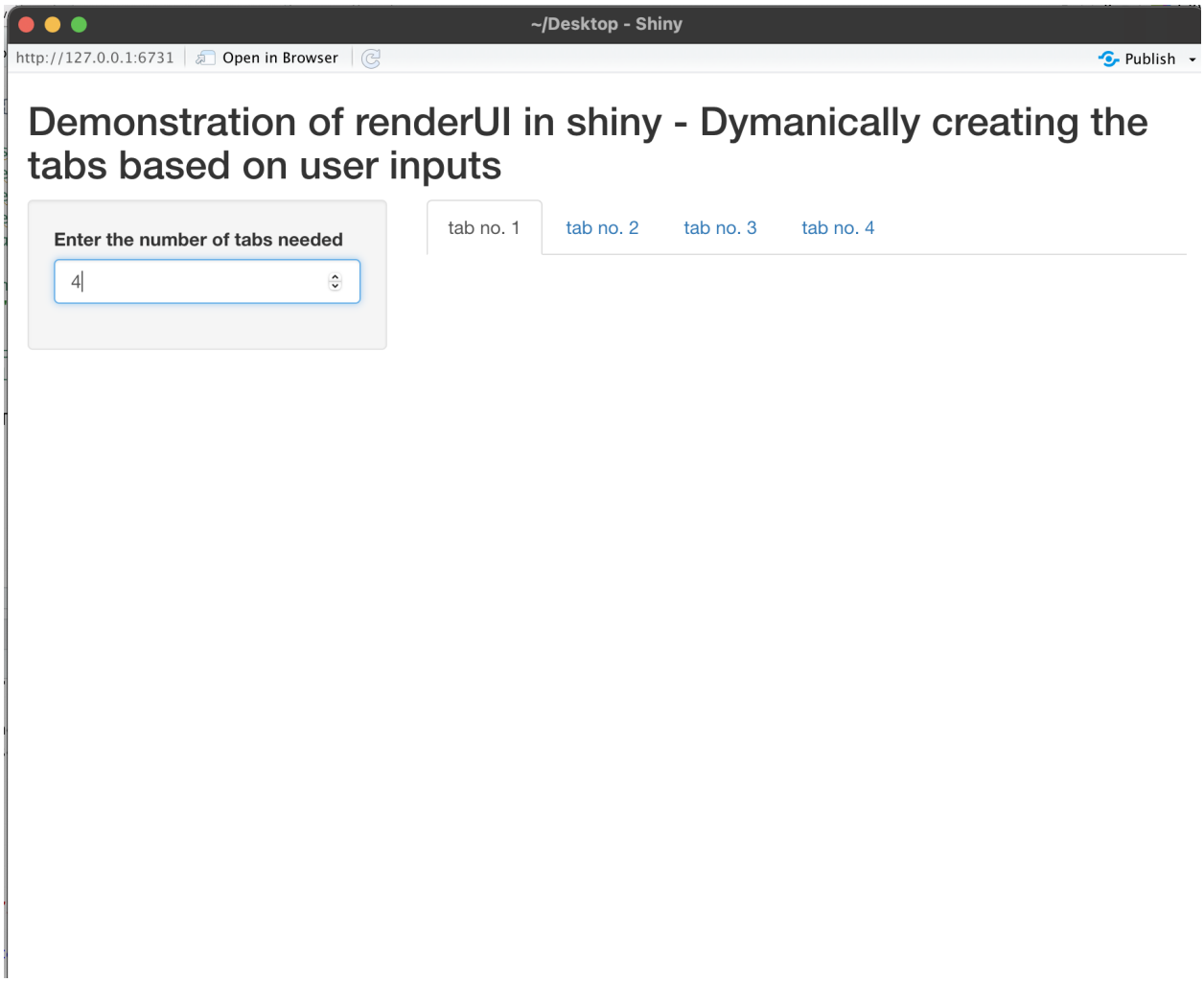


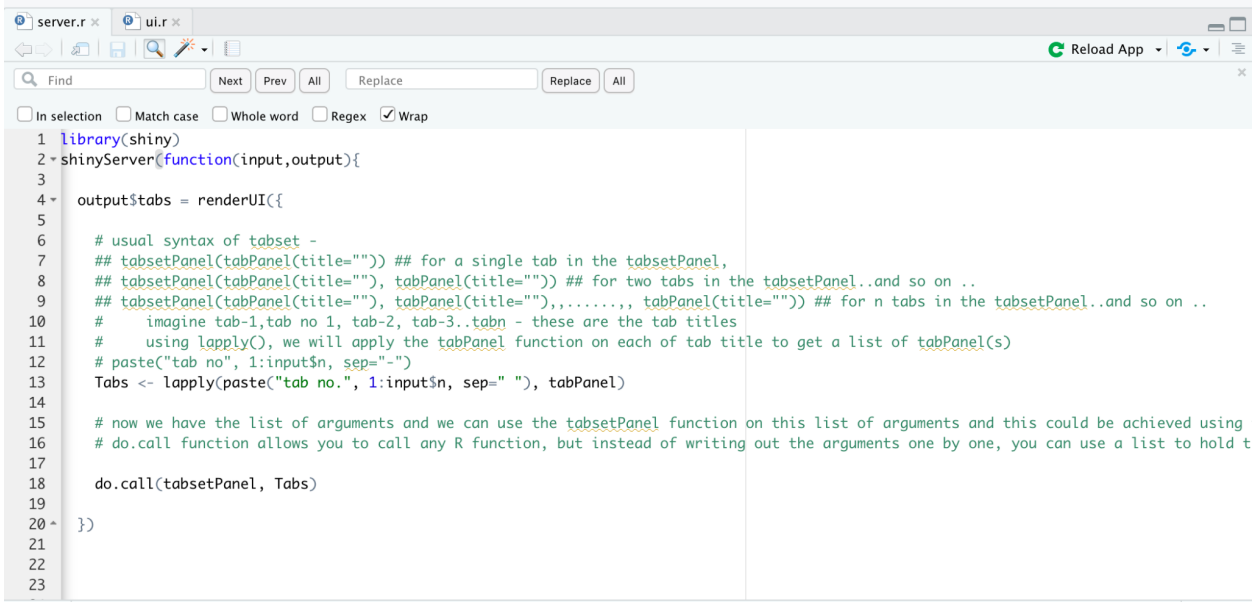
Another use of submitButton() in shiny.





How to use renderUI() and uiOutput() functions to create the tabs dynamically based on the number of tabs selected by the user


~/Desktop - Shiny
http://127.0.0.1:6731
Open in Browser
Publish
Demonstration of renderUI in shiny - Dymanically creating the tabs based on user inputs
Enter the number of tabs needed
4
tab no. 1
tab no. 2
tab no. 3
tab no. 4


server.r
ui.r
Reload App
Find
Next
Prev
All
Replace
Replace
All
In selection
Match case
Whole word
Regex
Wrap
library(shiny)
shinyServer(function(input,output){
output$tabs = renderUI({
# usual syntax of tabset -
tabsetPanel(tabPanel(title="")) ## for a single tab in the tabsetPanel,
tabsetPanel(tabPanel(title=""), tabPanel(title="")) ## for two tabs in the tabsetPanel..and so on ..
tabsetPanel(tabPanel(title=""), tabPanel(title=""),,.....,, tabPanel(title="")) ## for n tabs in the tabsetPanel..and so on ..
# imagine tab-1,tab no 1, tab-2, tab-3..tabn - these are the tab titles
# using lapply(), we will apply the tabPanel function on each of tab title to get a list of tabPanel(s)
# paste("tab no", 1:input$n, sep="-")
Tabs <- lapply(paste("tab no.", 1:input$n, sep=" "), tabPanel)
# now we have the list of arguments and we can use the tabsetPanel function on this list of arguments and this could be achieved using
# do.call function allows you to call any R function, but instead of writing out the arguments one by one, you can use a list to hold t
do.call(tabsetPanel, Tabs)
})

How to use the fileInput() function in RShiny to add a file upload option to a shiny app

```
Find  Next  Prev  All  Replace  Replace  All
In selection  Match case  Whole word  Regex  Wrap
```

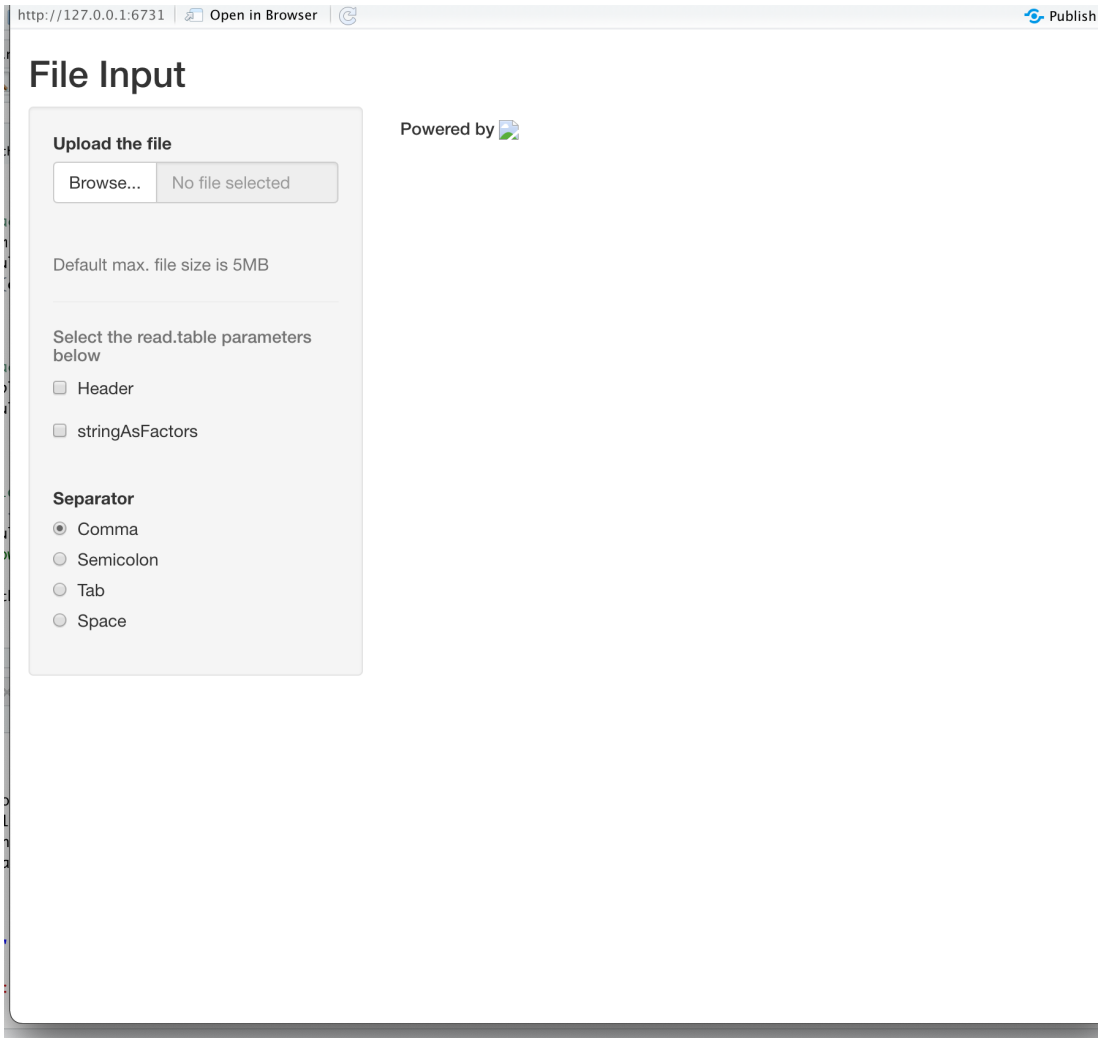
```r
library(shiny)
# use the below options code if you wish to increase the file input limit, in this example file input limit is increased from 5MB to 9
# options(shiny.maxRequestSize = 9*1024^2)

shinyServer(function(input,output){

  # This reactive function will take the inputs from UI.R and use them for read.table() to read the data from the file. It returns the
  # file$datapath -> gives the path of the file
  data <- reactive({
    file1 <- input$file
    if(is.null(file1)){return()}
    read.table(file=file1$datapath, sep=input$sep, header = input$header, stringsAsFactors = input$stringAsFactors)

  })

  # this reactive output contains the summary of the dataset and display the summary in table format
  output$filedf <- renderTable({
    if(is.null(data())){return ()}
    input$file
  })

  # this reactive output contains the summary of the dataset and display the summary in table format
  output$sum <- renderTable({
```

```
42:3  (Top Level)  R
```

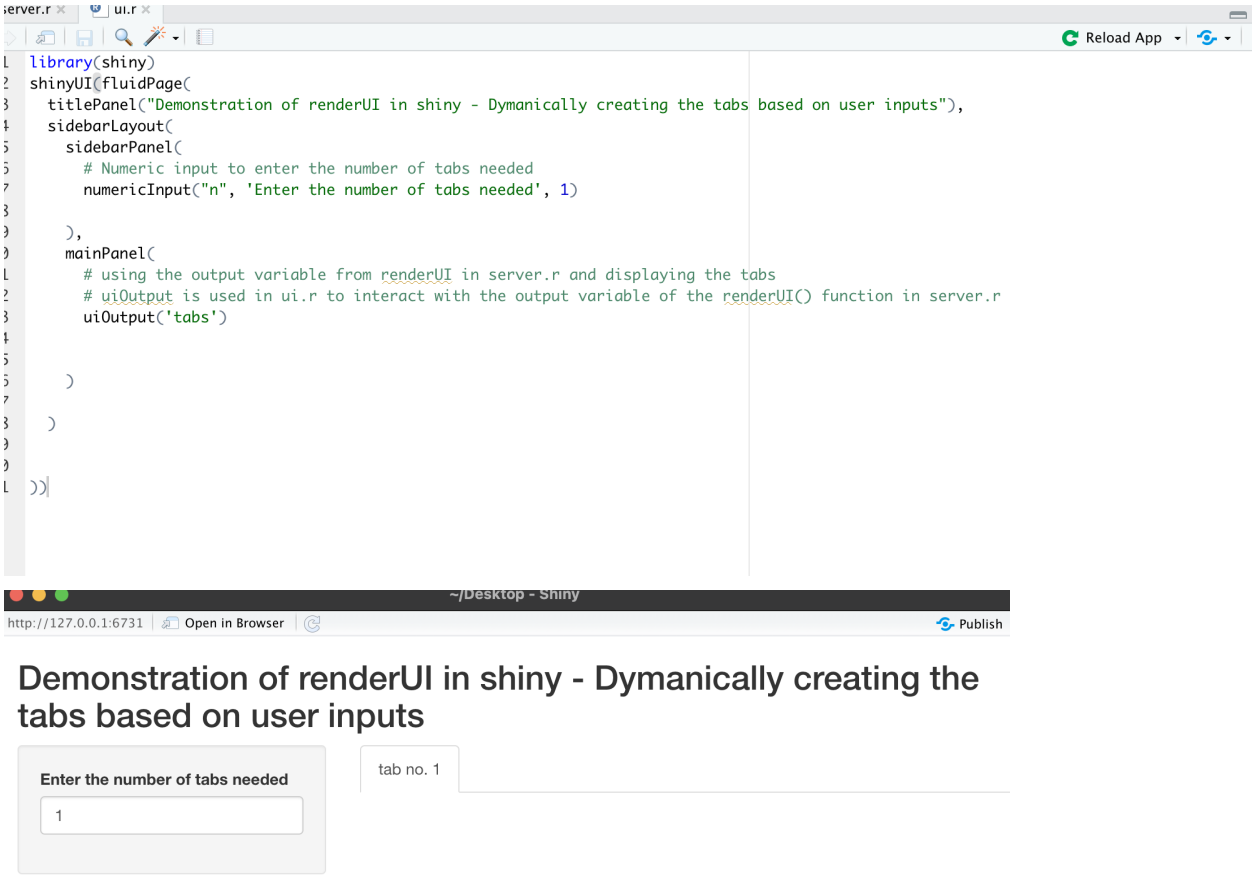
```
server.r  ui.r  Reload App
```

```r
library(shiny)
shinyUI(fluidPage(
  titlePanel("File Input"),
  sidebarLayout(
    sidebarPanel(
      fileInput("file","Upload the file"), # fileinput() function is used to get the file upload contorl option
      helpText("Default max. file size is 5MB"),
      tags$hr(),
      h5(helpText("Select the read.table parameters below")),
      checkboxInput(inputId = 'header', label = 'Header', value = FALSE),
      checkboxInput(inputId = "stringAsFactors", "stringAsFactors", FALSE),
      br(),
      radioButtons(inputId = 'sep', label = 'Separator', choices = c(Comma=',',Semicolon=';',Tab='\t', Space=''), selected = ',')
    ),
    mainPanel(
      uiOutput("tb")

      # use below code if you want the tabset programming in the main panel. If so, then tabset will appear when the app loads for the fi
      #       tabsetPanel(tabPanel("Summary", verbatimTextOutput("sum")),
      #                   tabPanel("Data", tableOutput("table")))
    )

  )
))
```

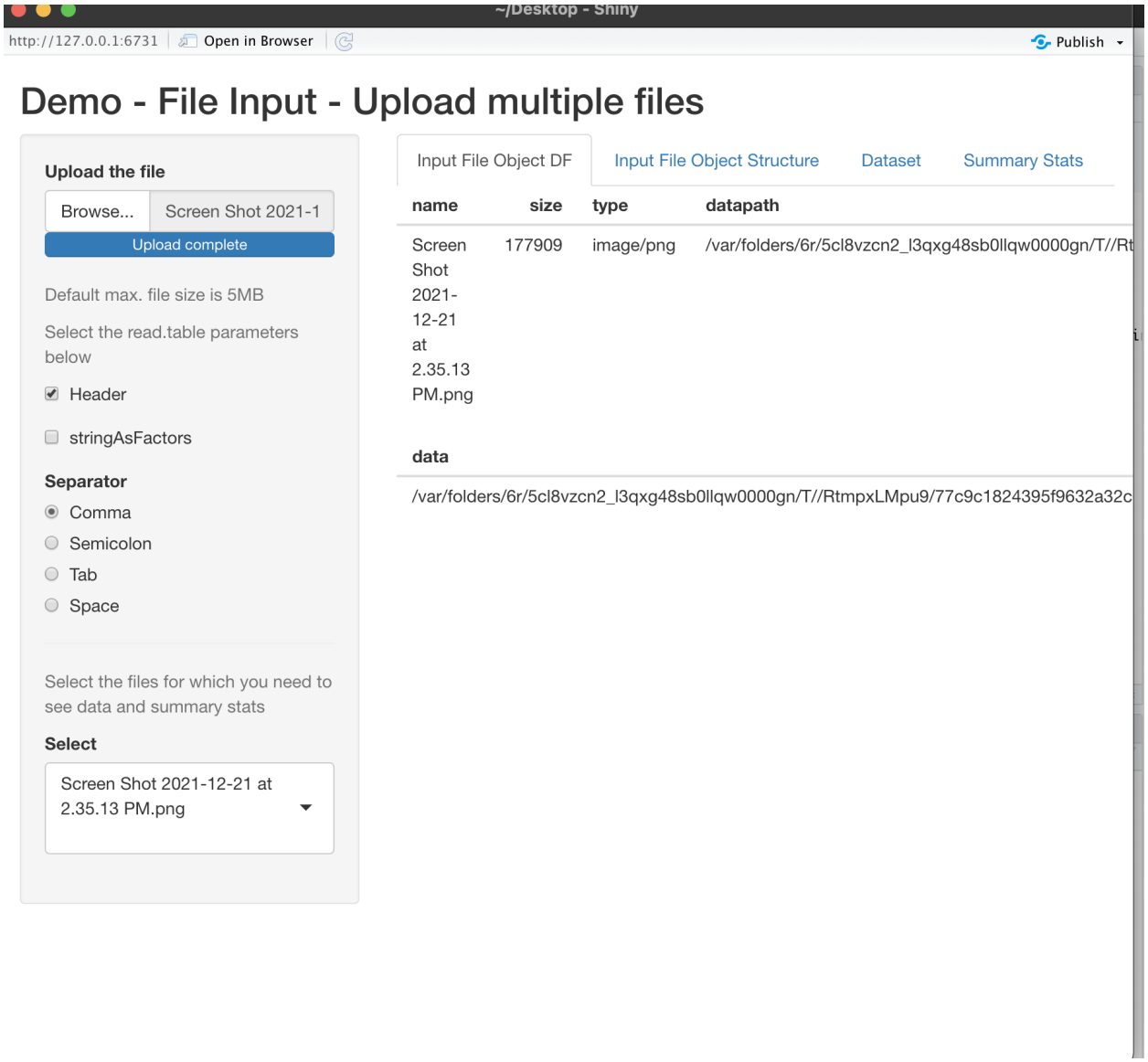
```
24:3  (Top Level)  R Script
```


http://127.0.0.1:6731
Open in Browser
Publish
File Input
Upload the file
Browse...
No file selected
Default max. file size is 5MB
Select the read.table parameters below
Header
stringAsFactors
Separator
Comma
Semicolon
Tab
Space
Powered by

how to use renderUI() and uiOutput() functions to create the tabs dynamically based on the number of tabs selected by the user.



How to upload multiple files using fileinput() in R Shiny (1)


~/Desktop - Shiny
http://127.0.0.1:6731
Open in Browser
Publish
Demo - File Input - Upload multiple files
Upload the file
Browse...
Screen Shot 2021-1
Upload complete
Default max. file size is 5MB
Select the read.table parameters below
Header
stringAsFactors
Separator
Comma
Semicolon
Tab
Space
Select the files for which you need to see data and summary stats
Select
Screen Shot 2021-12-21 at 2.35.13 PM.png
Input File Object DF
Input File Object Structure
Dataset
Summary Stats
name
size
type
datapath
Screen Shot 2021-12-21 at 2.35.13 PM.png
177909
image/png
/var/folders/6r/5cl8vzcn2_l3qxg48sb0llqw0000gn/T//Rt
data
/var/folders/6r/5cl8vzcn2_l3qxg48sb0llqw0000gn/T//RtmpxLMpu9/77c9c1824395f9632a32c

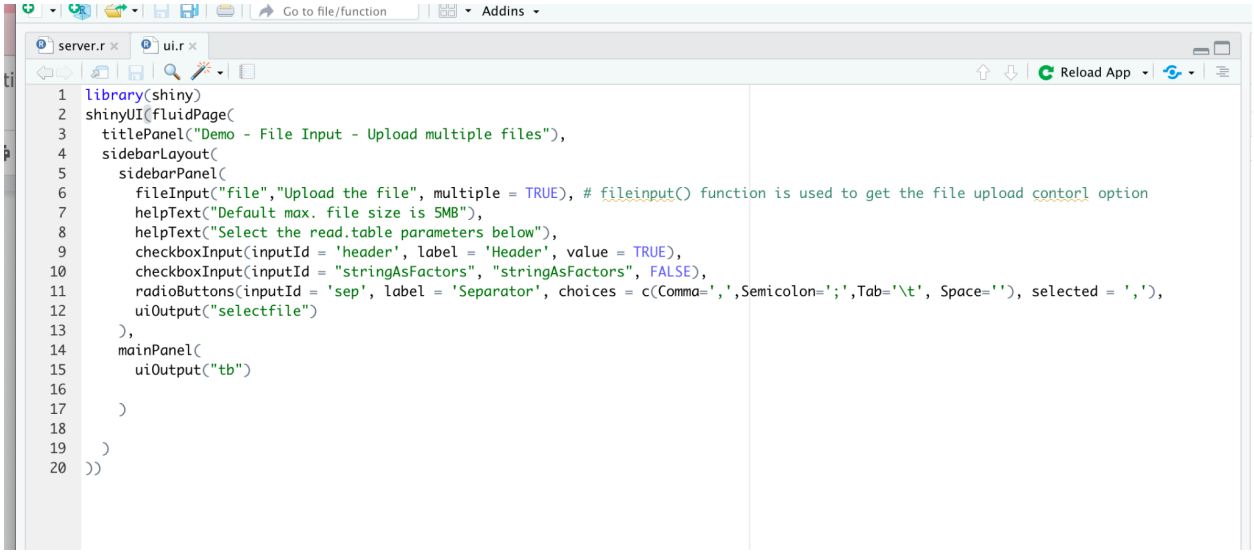
In selection | Match case | Whole word | Regex | Wrap

```r
library(shiny)
# use the below options code if you wish to increase the file input limit, in this example file input limit is increased from 5MB to 9MB
# options(shiny.maxRequestSize = 9*1024^2)

shinyServer(function(input,output) {

  ## input$file is a data frame and contains the details around the name, size and temp location of the files uploaded
  # this reactive output display the content of the input$file dataframe
  output$filedf <- renderTable({
    if(is.null(input$file)){return ()}
    input$file # the file input data frame object that contains the file attributes
  })

  # Extract the file path for file
  output$filedf2 <- renderTable({
    if(is.null(input$file)){return ()}
    input$file$datapath # the file input data frame object that contains the file attributes
  })

  ## Below code to display the structure of the input file object
  output$fileob <- renderPrint({
    if(is.null(input$file)){return ()}
    str(input$file)
  })

  ## Side bar select input widget coming through renderUI()
  # Following code displays the select input widget with the list of file loaded by the user
  output$selectfile <- renderUI({
    if(is.null(input$file)) {return()}
    list(hr(),
         helpText("Select the files for which you need to see data and summary stats"),
         selectInput("Select", "Select", choices=input$file$name)
    )

  })
```

66:3 (Top Level) R Script

Input File Object DF | Input File Object Structure | Dataset | Summary Stats

```
'data.frame':   1 obs. of  4 variables:
 $ name    : chr "Screen Shot 2021-12-21 at 2.35.13 PM.png"
 $ size    : int 177909
 $ type    : chr "image/png"
 $ datapath: chr "/var/folders/6r/5cl8vzcn2_l3qxg48sb0llqw0000gn/T//Rtmp
```



Demo rowbind data from multiple uploaded files

multiple files

Upload the file

Browse... Screen Shot 2021-1

Upload complete

Default max. file size is 5MB

Select the read.table parameters below

- [x] Header
- [ ] stringAsFactors

Separator

- (•) Comma
- ( ) Semicolon
- ( ) Tab
- ( ) Space

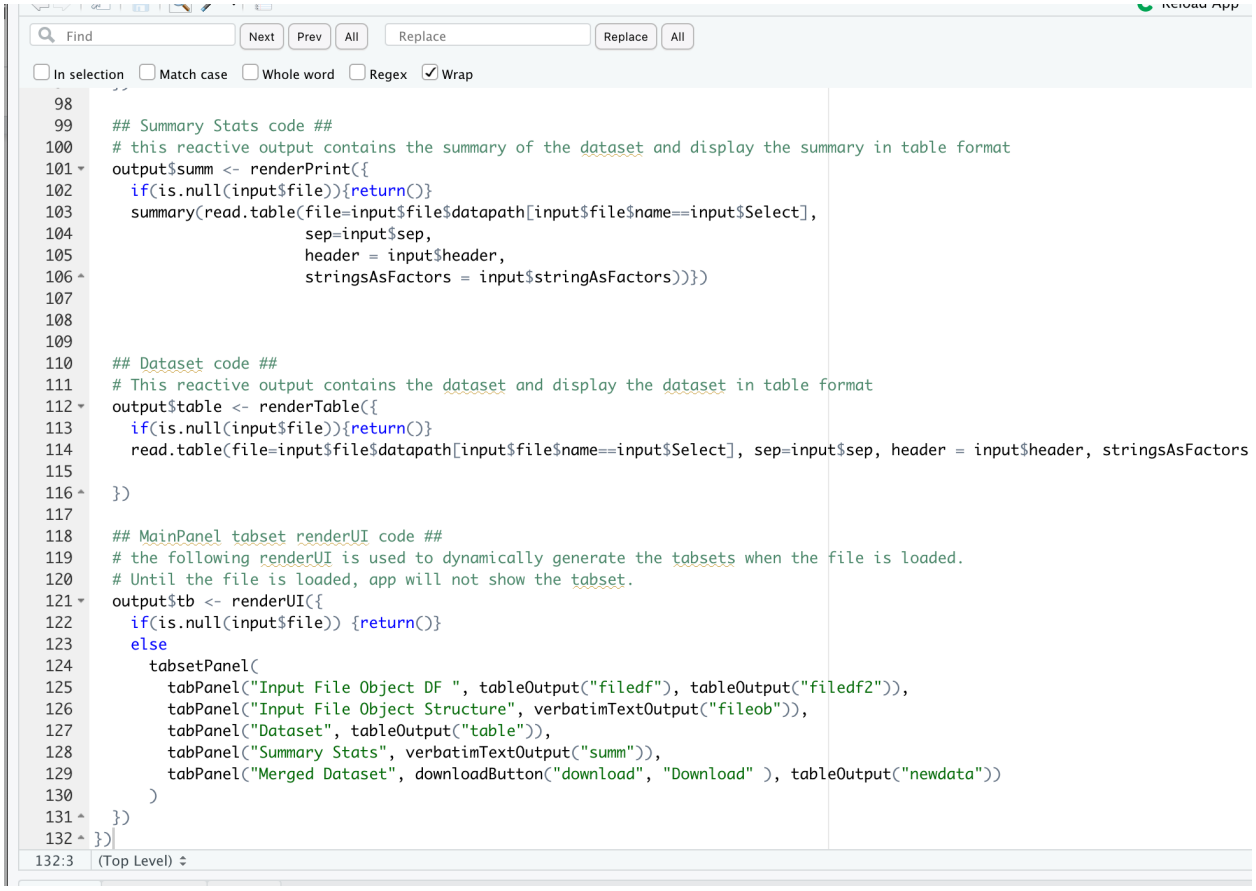
Select the files for which you need to see data and summary stats

Select

Screen Shot 2021-12-21 at 2.31.24 PM.png

Input File Object DF | Input File Object Structure | Dataset | Summary Stats | Merged Dataset

```
'data.frame':   1 obs. of  4 variables:
 $ name    : chr "Screen Shot 2021-12-21 at 2.31.24 PM.png"
 $ size    : int 159328
 $ type    : chr "image/png"
 $ datapath: chr "/var/folders/6r/5cl8vzcn2_l3qxg48sb0llqw0000gn/T//Rtm
```



```
Find  Next  Prev  All  Replace  Replace  All
In selection  Match case  Whole word  Regex  Wrap

## Summary Stats code ##
# this reactive output contains the summary of the dataset and display the summary in table format
output$summ <- renderPrint({
  if(is.null(input$file)){return()}
  summary(read.table(file=input$file$datapath[input$file$name==input$Select],
                     sep=input$sep,
                     header = input$header,
                     stringsAsFactors = input$stringAsFactors))})



## Dataset code ##
# This reactive output contains the dataset and display the dataset in table format
output$table <- renderTable({
  if(is.null(input$file)){return()}
  read.table(file=input$file$datapath[input$file$name==input$Select], sep=input$sep, header = input$header, stringsAsFactors

})

## MainPanel tabset renderUI code ##
# the following renderUI is used to dynamically generate the tabsets when the file is loaded.
# Until the file is loaded, app will not show the tabset.
output$tb <- renderUI({
  if(is.null(input$file)) {return()}
  else
    tabsetPanel(
      tabPanel("Input File Object DF ", tableOutput("filedf"), tableOutput("filedf2")),
      tabPanel("Input File Object Structure", verbatimTextOutput("fileob")),
      tabPanel("Dataset", tableOutput("table")),
      tabPanel("Summary Stats", verbatimTextOutput("summ")),
      tabPanel("Merged Dataset", downloadButton("download", "Download" ), tableOutput("newdata"))
    )
})
})
132:3  (Top Level)
```

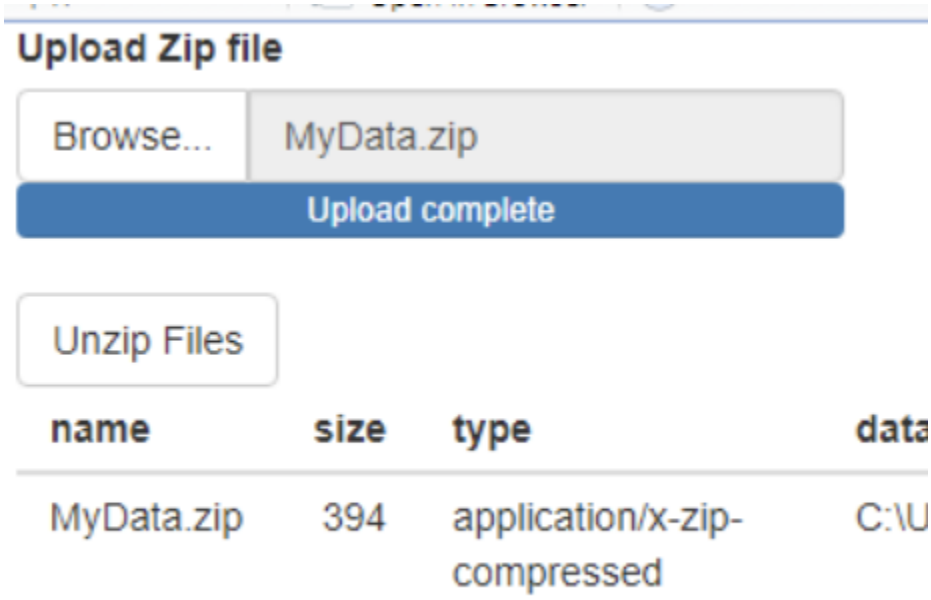
Input File Object DF | Input File Object Structure | Dataset | Summary Stats

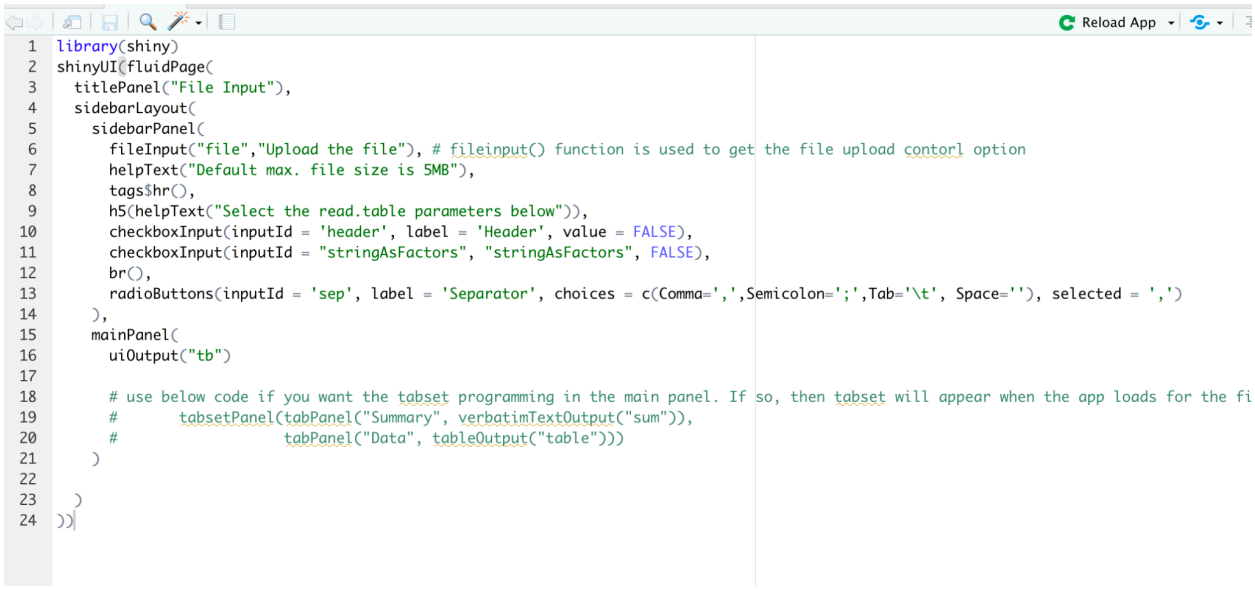
Merged Dataset

name	size	type	datapath
Screen Shot 2021-12-21 at 2.31.24 PM.png	159328	image/png	/var/folders/6r/5cl8vzcn2_l3qxg48sb0llqw0000gn/T//RtmpxLMpu9/25dfcf186d69253f8cb07146/0

data
/var/folders/6r/5cl8vzcn2_l3qxg48sb0llqw0000gn/T//RtmpxLMpu9/25dfcf186d69253f8cb07146/0.png

upload a zip file and unzip it to local machine





```
library(shiny)
shinyUI(fluidPage(
  titlePanel("File Input"),
  sidebarLayout(
    sidebarPanel(
      fileInput("file","Upload the file"), # fileinput() function is used to get the file upload contorl option
      helpText("Default max. file size is 5MB"),
      tags$hr(),
      h5(helpText("Select the read.table parameters below")),
      checkboxInput(inputId = 'header', label = 'Header', value = FALSE),
      checkboxInput(inputId = "stringAsFactors", "stringAsFactors", FALSE),
      br(),
      radioButtons(inputId = 'sep', label = 'Separator', choices = c(Comma=',',Semicolon=';',Tab='\t', Space=''), selected = ',')
    ),
    mainPanel(
      uiOutput("tb")

      # use below code if you want the tabset programming in the main panel. If so, then tabset will appear when the app loads for the fi
      #       tabsetPanel(tabPanel("Summary", verbatimTextOutput("sum")),
      #                   tabPanel("Data", tableOutput("table")))
    )

  )
))
```

File Input

Upload the file

Browse... No file selected

Default max. file size is 5MB

Select the read.table parameters below

- [ ] Header
- [ ] stringAsFactors

Separator

- (•) Comma
- ( ) Semicolon
- ( ) Tab
- ( ) Space

Powered by