

Introduction To Shiny

The R Web Framework



About

Me

- Dai ZJ
- BSc (Maths) & M. Stats
- Risk Analytics
- Cantonese speaking Australian-Chinese



- Credit Scoring & Basel II Specialist
- Software
- Consulting
- Training

Shiny

What?

- R package for building “live” web apps
- Allows you to make web apps entirely* with R
- Great presenting dynamic analysis
- Pedigree RStudio



Why?

- UX - for user experience
- Interactivity

Shine Get!

- `install.packages("shiny")`



A tales of two files

A typical Shiny app consists of two files

ui.R

- User Interface
- Translates to HTML

server.R

- data analysis
- computation

To run a Shiny app

- shiny::runApp(path)
- path is the path that contains ui.R and server.R

First Example

ui.R

```
library(shiny)

shinyUI(pageWithSidebar(
  # Application title
  headerPanel("New Application"),
  # Sidebar with a slider input
  # for number of observations
  sidebarPanel(
    sliderInput("obs",
               "Number of observations:",
               min = 1,
               max = 1000,
               value = 500)
  ),
  # Show a plot of the generated
  # distribution
  mainPanel(
    plotOutput("distPlot")
  )
))
```

server.R

```
library(shiny)

shinyServer(function(input, output) {
  output$distPlot <- renderPlot({
    ...
    # generate and plot an rnorm
    # distribution with the requested
    # number of observations
    dist <- rnorm(input$obs)
    hist(dist)
  })
})
```

First Example - ui.R

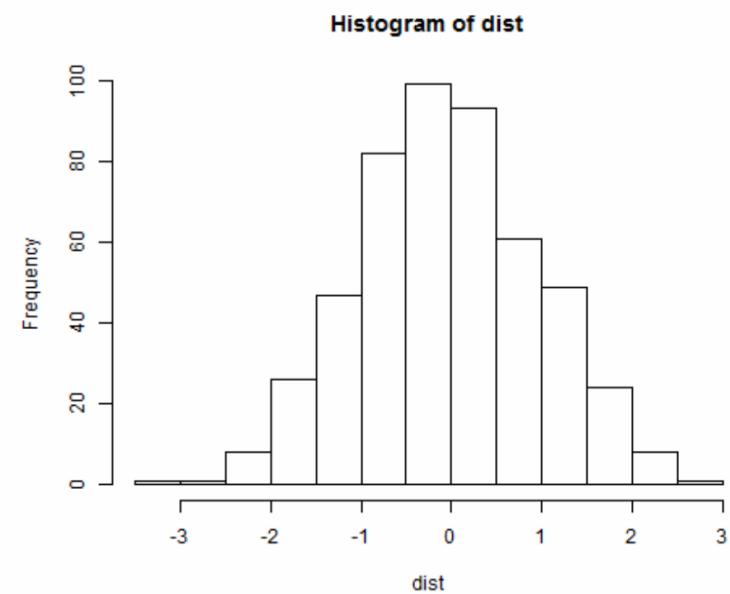
```
library(shiny)

shinyUI(pageWithSidebar(
  # Application title
  headerPanel("New Application"),
  # Sidebar with a slider input
  # for number of observations
  sidebarPanel(
    sliderInput("obs",
               "Number of observations:",
               min = 1,
               max = 1000,
               value = 500)
  ),
  # Show a plot of the generated
  # distribution
  mainPanel(
    plotOutput("distPlot")
  )
))
```

New Application

Number of observations:

1 500 1,000



First Example

ui.R

```
library(shiny)  
  
shinyUI(pageWithSidebar(  
  
  # Application title  
  headerPanel("New Application"),  
  
  # Sidebar with a slider input  
  # for number of observations  
  sidebarPanel(  
    sliderInput("obs",  
      "Number of observations:",  
      min = 1,  
      max = 1000,  
      value = 500)  
  ),  
  
  # Show a plot of the generated  
  # distribution  
  mainPanel(  
    plotOutput("distPlot")  
  )  
)
```

server.R

```
library(shiny)  
  
shinyServer(function(input, output) {  
  
  output$distPlot <- renderPlot({  
    # generate and plot an rnorm  
    # distribution with the requested  
    # number of observations  
    dist <- rnorm(input$obs)  
    hist(dist)  
  })  
})
```

Reactive Programming

What is NOT reactive programming?

- Most R code is not reactive

```
a = 1
b = 1
c = a + b
a = 2 #even though I changed the value of a, the value of c remained unchanged
```

Reactive programming is...

- if the value of `c` changed with `a`
- Shiny

reactive()

```
# Show a plot of the generated distribution
mainPanel(
  verbatimTextOutput("summary"),
  br(),
  plotOutput("distPlot")
)
```

ui.R

```
library(shiny)
shinyServer(function(input, output) {
  ...
  random_sample <- reactive({
    rnorm(input$obs)
  })

  median_rm <- reactive({
    median(random_sample())
  })

  mean_rm <- reactive({
    mean(random_sample())
  })

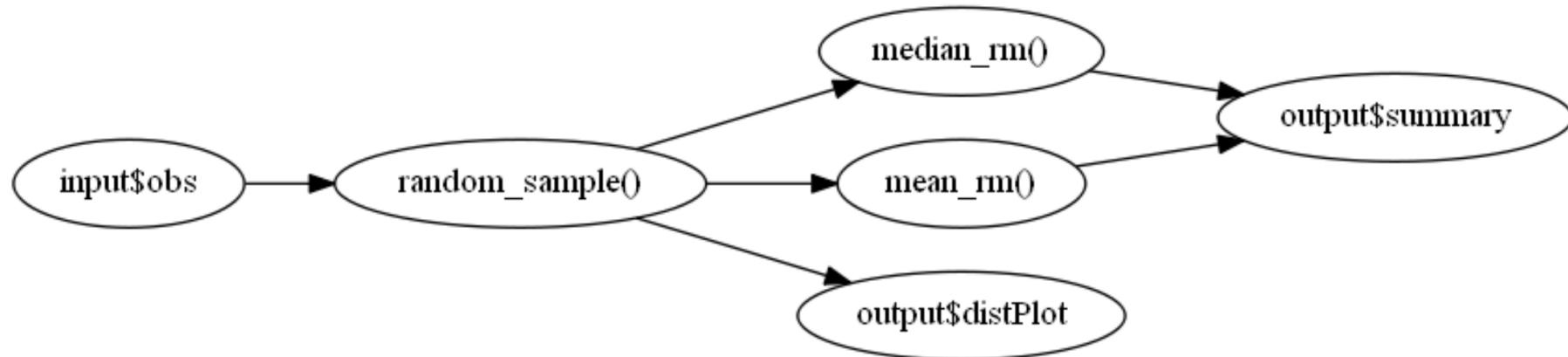
  output$summary <- renderPrint({
    print(paste("The random sample Median:", format(median_rm(), digits=2)))
    print(paste("The random sample Mean:", format(mean_rm(), digits=2)))
  })

  output$distPlot <- renderPlot({
    ...
    # generate and plot an rnorm distribution with the requested
    # number of observations

    #dist <- rnorm(input$obs)
    dist <- random_sample()
    hist(dist)
  })
})
```

server.R

reactive()



actionButton

When you want the output to change based on button press

- e.g. the computation can take too long

Syntax

- `actionButton("id", label)`
- Initially `input$id` will be equal zero
- Every time it is pressed `input$id` increments by 1

actionButton - isolate

server.R

```
output$distPlot <- renderPlot({  
  # generate and plot an rnorm distrib  
  # number of observations  
  
  #dist <- rnorm(input$obs)  
  if(input$actionButton1 ==0) return()  
  isolate({  
    dist <- random_sample()  
    hist(dist)  
  })  
})
```

isolate()

- the output will NOT *react* to components wrapped inside it
- Common Mistake: omitting it

Available input widgets

- `textInput()`

- `numericInput()`

- `fileInput()`

 No file chosen

- `sliderInput()` doesn't work in these slides currently

- `actionButton()`

- `dateInput()`

- `checkboxInput()`

- `checkboxGroupInput()`

- A
- B
- C

- `selectInput()`

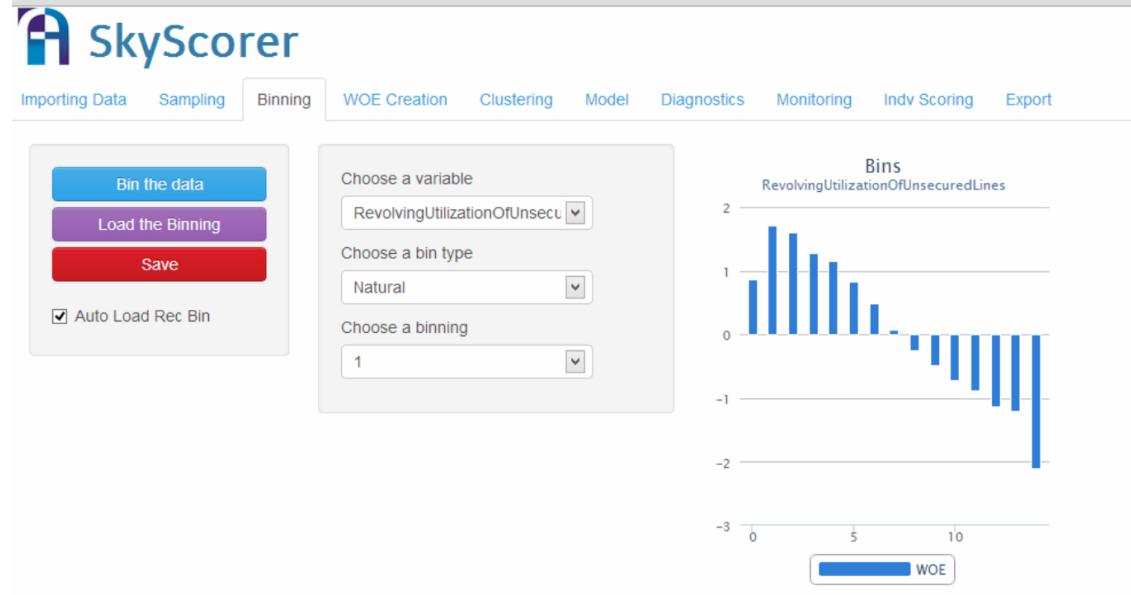
- `radioButtons()`

- A
- B
- C

What next?

- Official Tutorial
 - <http://rstudio.github.io/shiny/tutorial/#hello-shiny>
- Google Shiny Forum (search: *google groups shiny*)
- Cool Packages
 - rCharts (<http://rcharts.io/>)
 - shinyRGL (<http://trestletech.github.io/shinyRGL/>)
- Extend Shiny
 - Future Shiny workshop?
 - Need to know Javascript, HTML/CSS

Demo - SkyScorer



The Binning Table

name	bin_type	generation	bin	b	g	infoVal	sumb	sumg	woe
RevolvingUtilizationOfUnsecuredLines	Natural	1	0.00	322	10637	1.1677	10026	139974	0.86127
RevolvingUtilizationOfUnsecuredLines	Natural	1	0.01	128	9745	1.1677	10026	139974	1.6962
RevolvingUtilizationOfUnsecuredLines	Natural	1	0.05	385	26283	1.1677	10026	139974	1.5872
RevolvingUtilizationOfUnsecuredLines	Natural	1	0.09	260	12923	1.1677	10026	139974	1.2698
RevolvingUtilizationOfUnsecuredLines	Natural	1	0.14	258	11396	1.1677	10026	139974	1.1518
RevolvingUtilizationOfUnsecuredLines	Natural	1	0.19	259	8166	1.1677	10026	139974	0.81463
RevolvingUtilizationOfUnsecuredLines	Natu	Combine Rows		7 788	17818	1.1677	10026	139974	0.48219
RevolvingUtilizationOfUnsecuredLines	Natu	Break Rows		9 531	7975	1.1677	10026	139974	0.07303
RevolvingUtilizationOfUnsecuredLines	Natu	Round to...		9 544	5877	1.1677	10026	139974	-0.25642
RevolvingUtilizationOfUnsecuredLines	Natu	Undo		9 541	4650	1.1677	10026	139974	-0.48507
RevolvingUtilizationOfUnsecuredLines	Natu	Redo		9 565	3848	1.1677	10026	139974	-0.71779
RevolvingUtilizationOfUnsecuredLines	Natu	Console.log selected		5 563	3229	1.1677	10026	139974	-0.88963
RevolvingUtilizationOfUnsecuredLines	Natural	1	0.92	573	2583	1.1677	10026	139974	-1.1305

Finally

GitHub

https://github.com/xiaodaigh/Intro_to_Shiny

Survey

<https://www.surveymonkey.com/s/VKXVDHg>