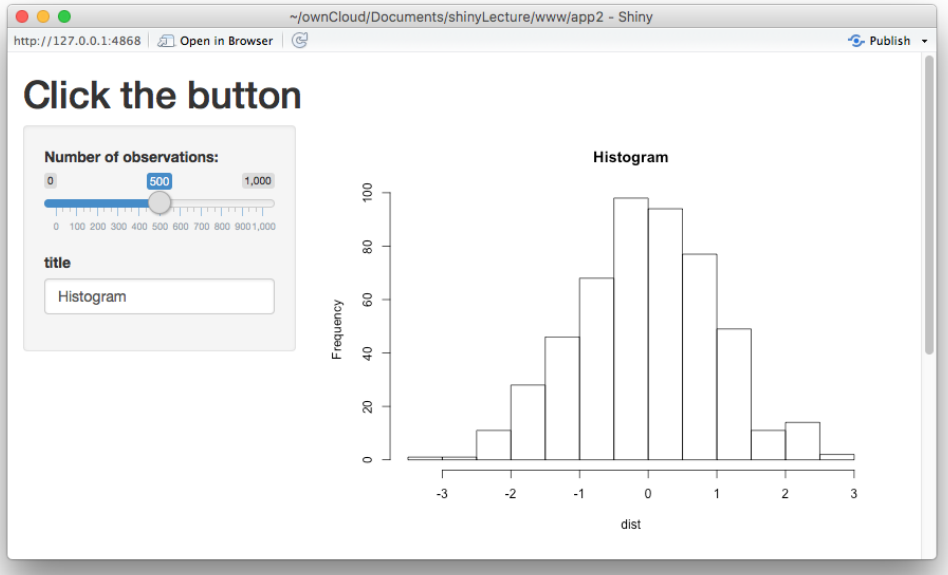
R과 Shiny를 이용한 Web Application의 제작(II)

문건웅

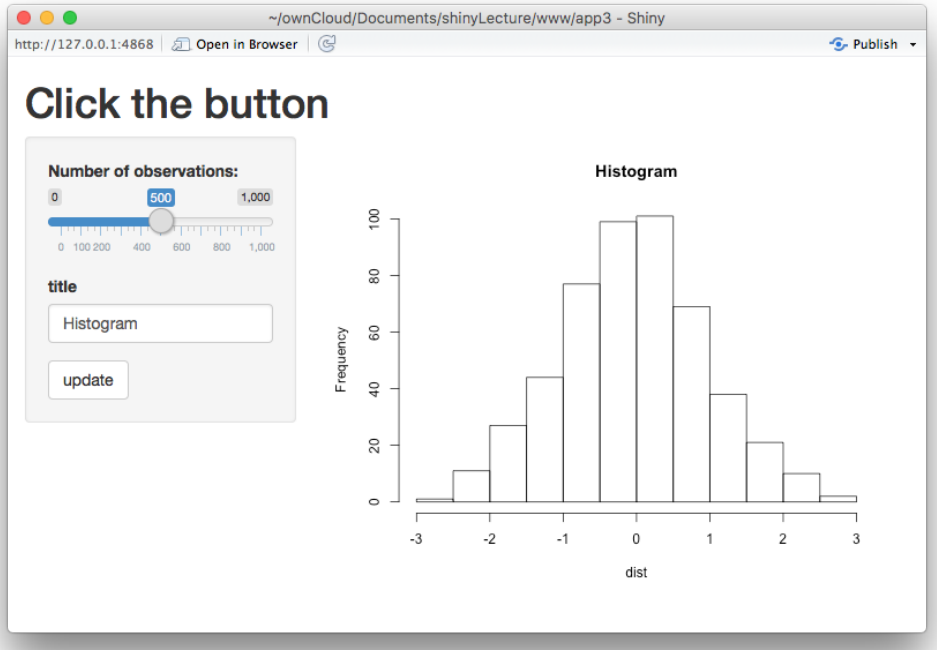
13-Oct-2017

2. Reactivity



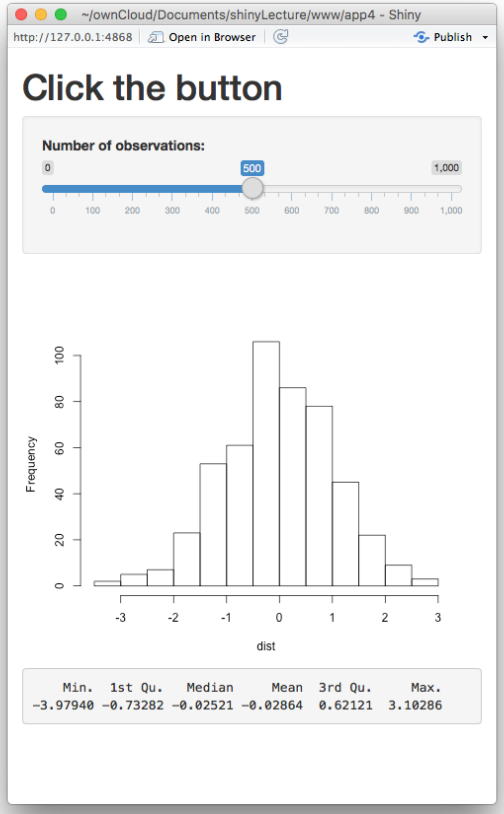

```
shiny::runGitHub('shinyLecture2', 'cardiomoon',subdir='inst/app2')
```

3. Stop reactions with isolate()

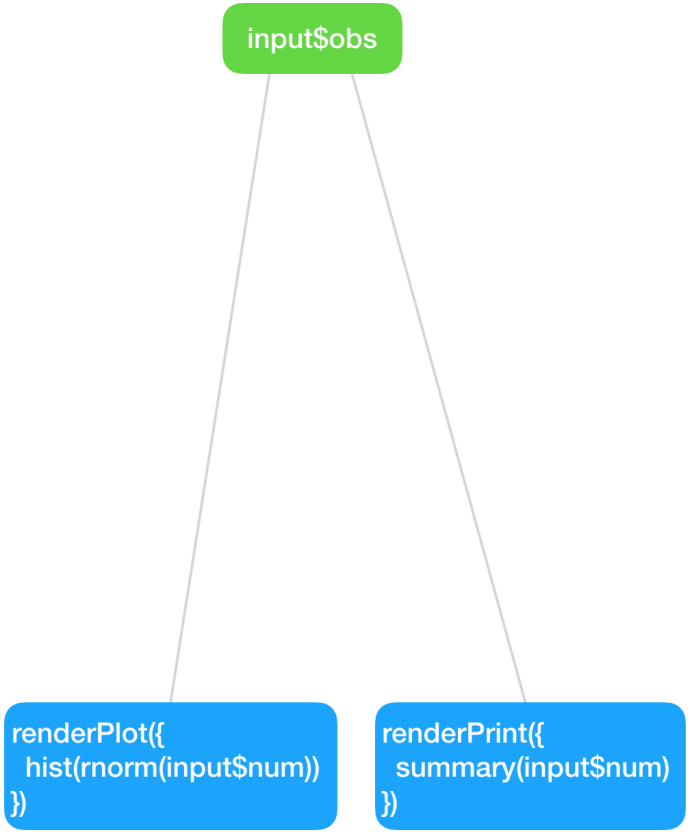


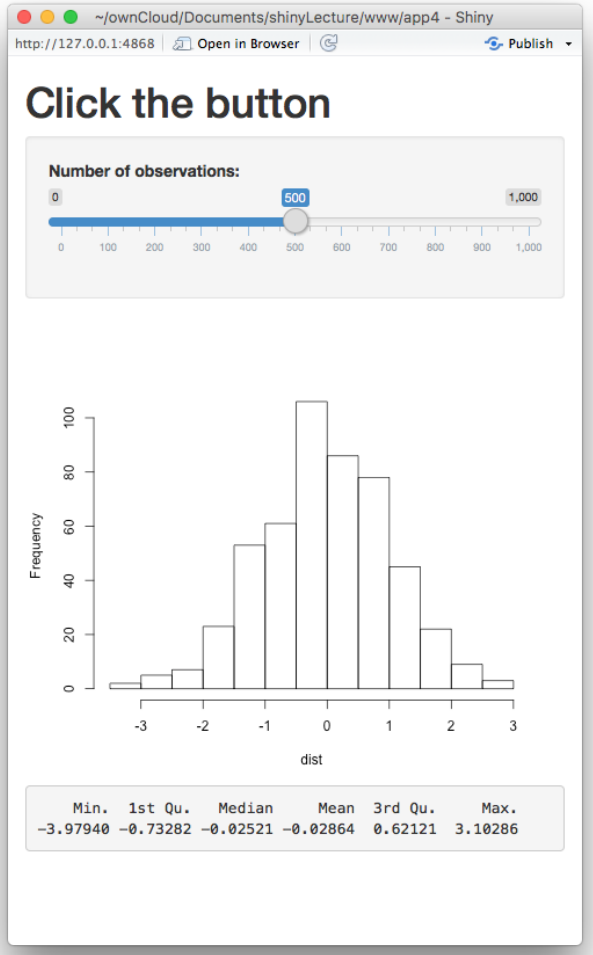

```
shiny::runGitHub('shinyLecture2', 'cardiomoon',subdir='inst/app3')
```

4. one input, two output




```
shiny::runGitHub('shinyLecture2', 'cardiomoon',subdir='inst/app4')
```


input$obs
renderPlot({
hist(rnorm(input$num))
})
renderPrint({
summary(input$num)
})


~/ownCloud/Documents/shinyLecture/www/app4 - Shiny
http://127.0.0.1:4868
Open in Browser
Publish
Click the button
Number of observations:
0
500
1,000
0
100
200
300
400
500
600
700
800
900
1,000
Frequency
0
20
40
60
80
100
-3
-2
-1
0
1
2
3
dist
Min. 1st Qu. Median Mean 3rd Qu. Max.
-3.97940 -0.73282 -0.02521 -0.02864 0.62121 3.10286

reactive()

- reactive 함수로 반응성 객체를 만든다.

```
data <-reactive({rnorm(input$num)})
```

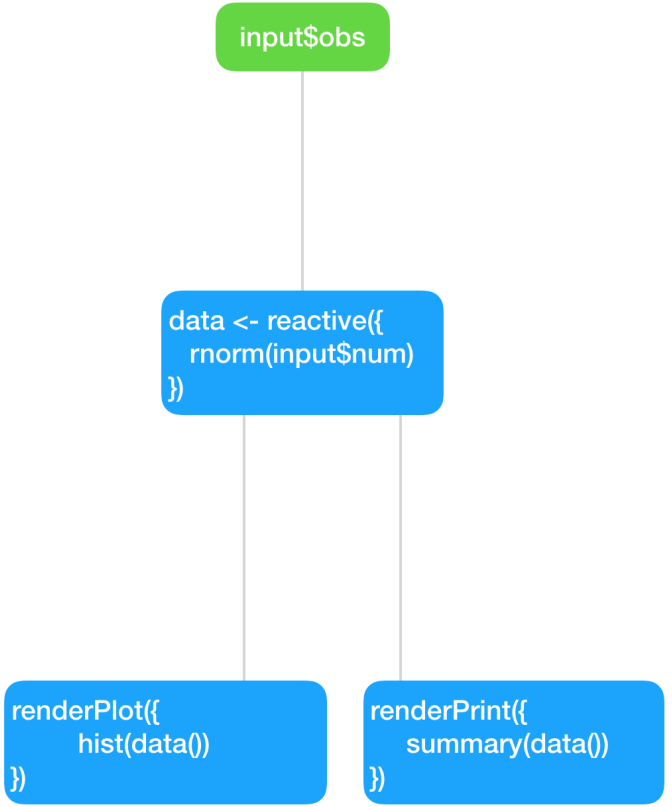
- 이 객체는 reactive value가 변할 때마다 반응한다.

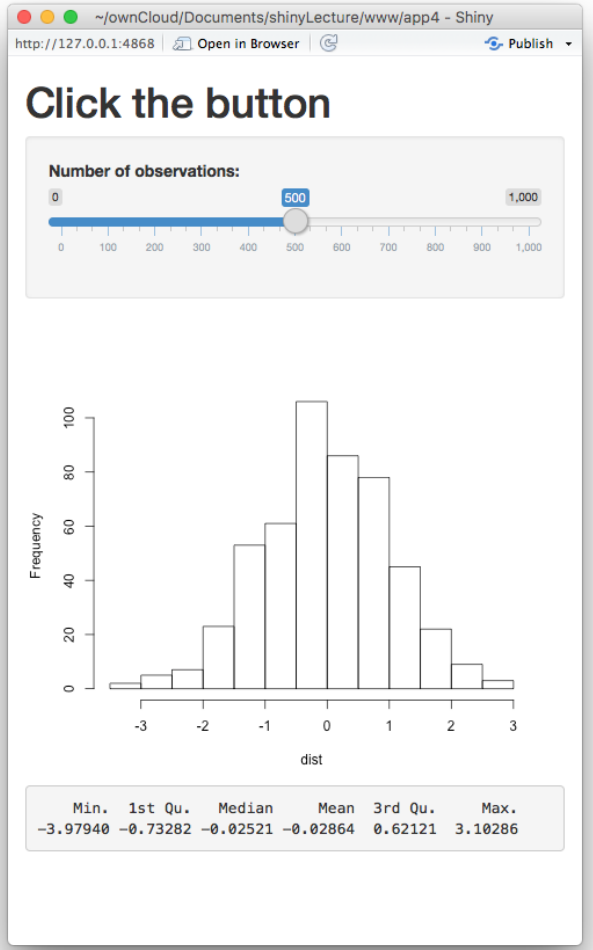
반응성 객체

반응성 객체의 두가지 면에서 특별하다.

```
data()
```

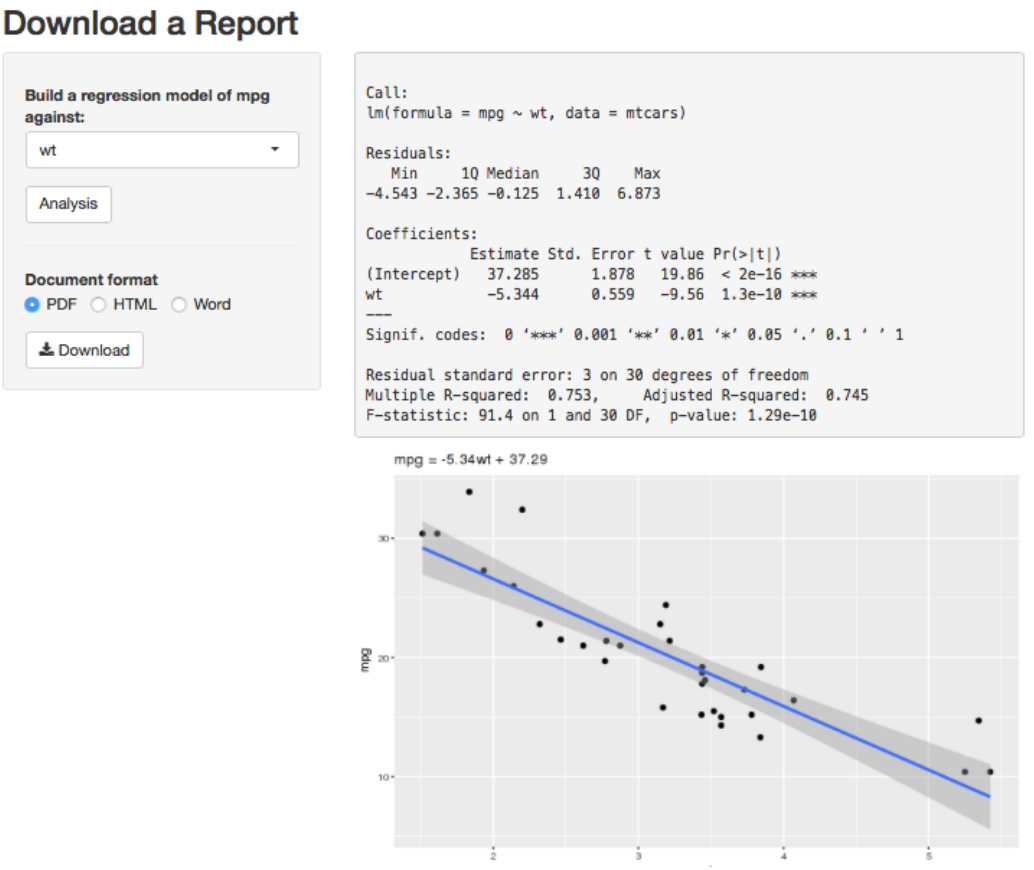
- 반응성 객체를 호출할 때는 함수처럼 호출한다.
- 반응성 객체는 그 값을 임시로 저장한다(cache).
    - 무효화되지 않을 경우 가장 최근의 값을 반환한다.






```
shiny::runGitHub('shinyLecture2', 'cardiomoon',subdir='inst/app5')
```

6. Download knitr Reports




```
shiny::runGitHub('shinyLecture2', 'cardiomoon',subdir='inst/app13')
```

Report.Rmd File

shinyLecture.Rmd | ShinyGadget.Rmd | test.Rmd | server.R | ui.R | Untitled1 | report.Rmd

Knit | Insert | Run

```
---
title: "Regression Analysis"
output: html_document
---

```{r setup, include=FALSE}
knitr::opts_chunk$set(echo = TRUE,comment=NA)
```

Summary of Regression Model:

```{r model, echo=FALSE}
options(digits = 2)
fit <- eval(parse(text=paste0("lm( mpg ~",input$x,",data = mtcars)")))
b   <- coef(fit)
```

```{r}
summary(fit)
```

The fitting result is $mpg = `r b[2]`r input$x`+`r b[1]`$.
Below is a scatter plot with the regression line.

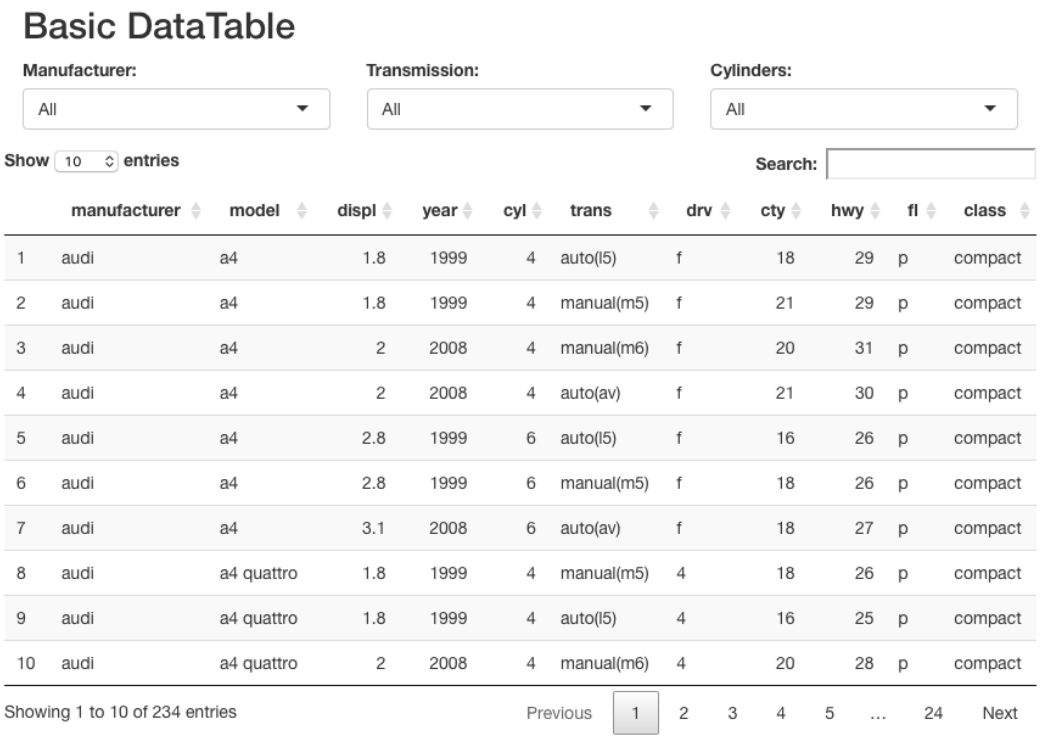
```{r plot, echo=FALSE,fig.height=4}
ggplot(data=mtcars,aes_string(req(input$x),"mpg"))+
          geom_point()+
          geom_smooth(method="lm")+
          ggtitle(regEquation())
```
```

https://github.com/cardiomoon/shinyLecture2/tree/master/inst/app13

PDF 다운로드를 위해서는

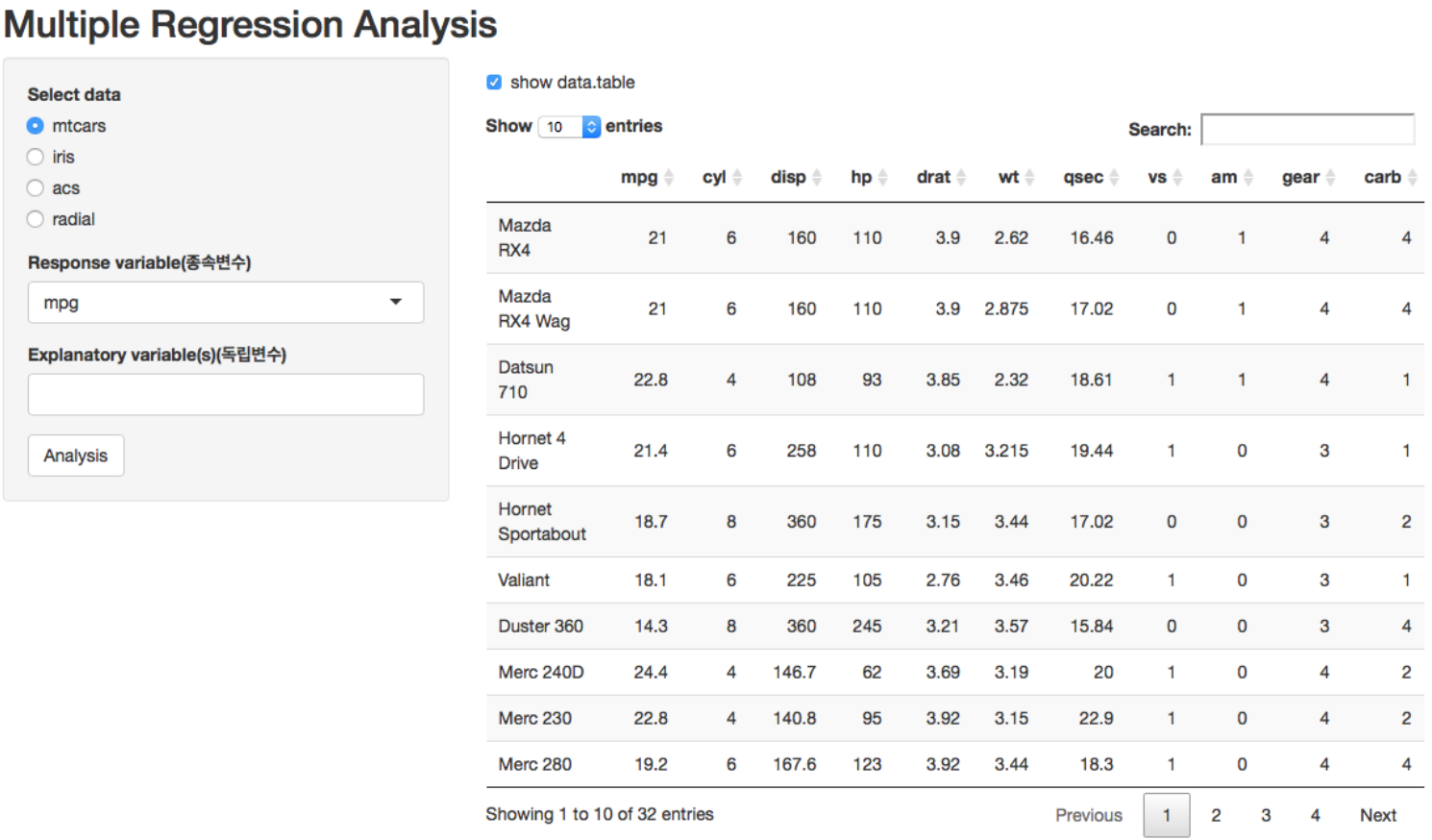
- 자신의 컴퓨터에 LaTex이 설치되어 있어야 한다. (http://ktug.or.kr)
- 또는 LaTex가 설치된 shiny server에서 shiny app을 실행하여야 한다.

7. Basic DataTable



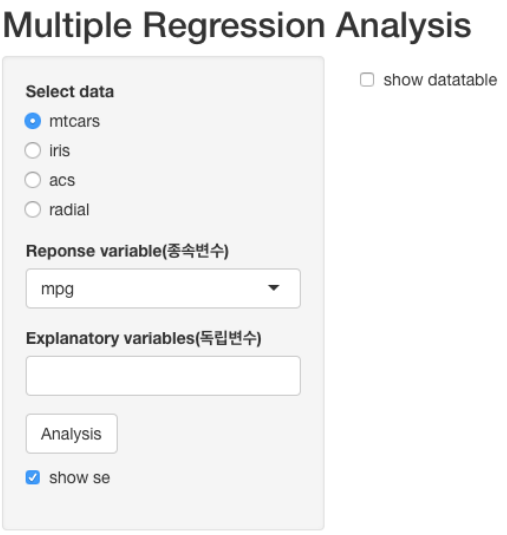

```
shiny::runGitHub('shinyLecture2', 'cardiomoon',subdir='inst/app14')
```

8. Advanced App - Multiple Reactive Output




```
shiny::runGitHub('shinyLecture2', 'cardiomoon',subdir='inst/app15')
```

9. Advanced App - Interactive Plot




```
shiny::runGitHub('shinyLecture2', 'cardiomoon',subdir='inst/app19')
```

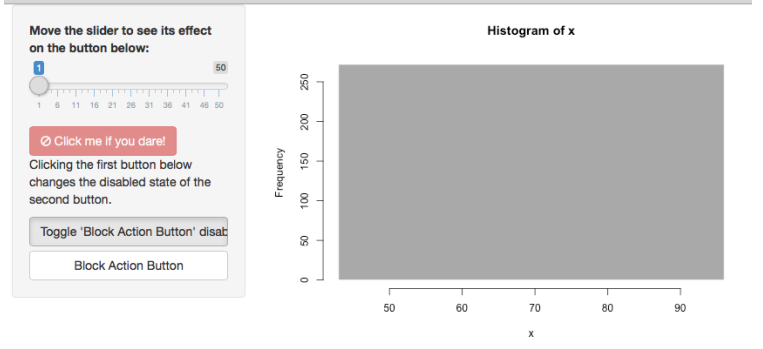
Add-on packages for shiny

1. shinyBS

homepage: https://ebailey78.github.io/shinyBS/index.html

Twitter Bootstrap Components for Shiny. Add additional functionality and interactivity to your Shiny applications.

bsButton, bsToolip, bsPopover

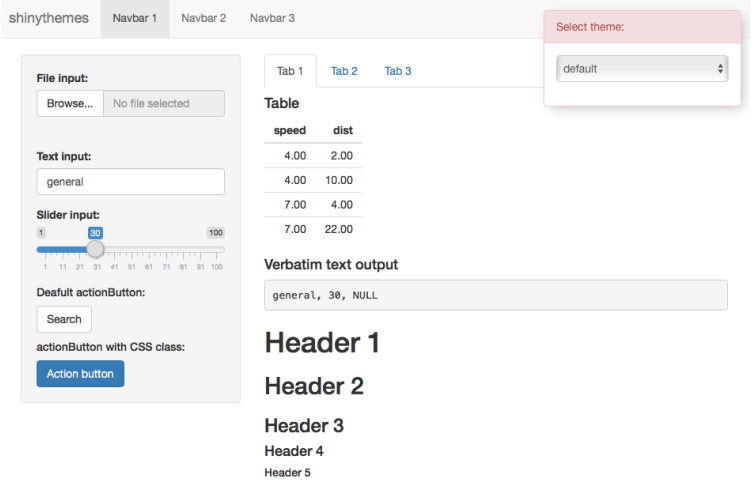



```
shiny::runGitHub('shinyLecture2', 'cardiomoon',subdir='inst/app16')
```

2. Shiny themes

Bootstrap themes

homepage: https://rstudio.github.io/shinythemes/

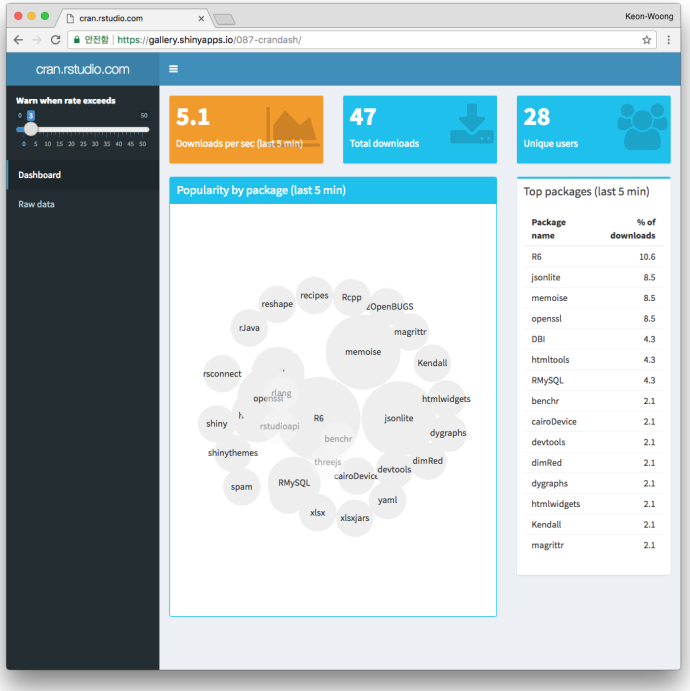


https://gallery.shinyapps.io/117-shinythemes/

3.shinydashboard

shinydashboard makes it easy to use Shiny to create dashboards like these:

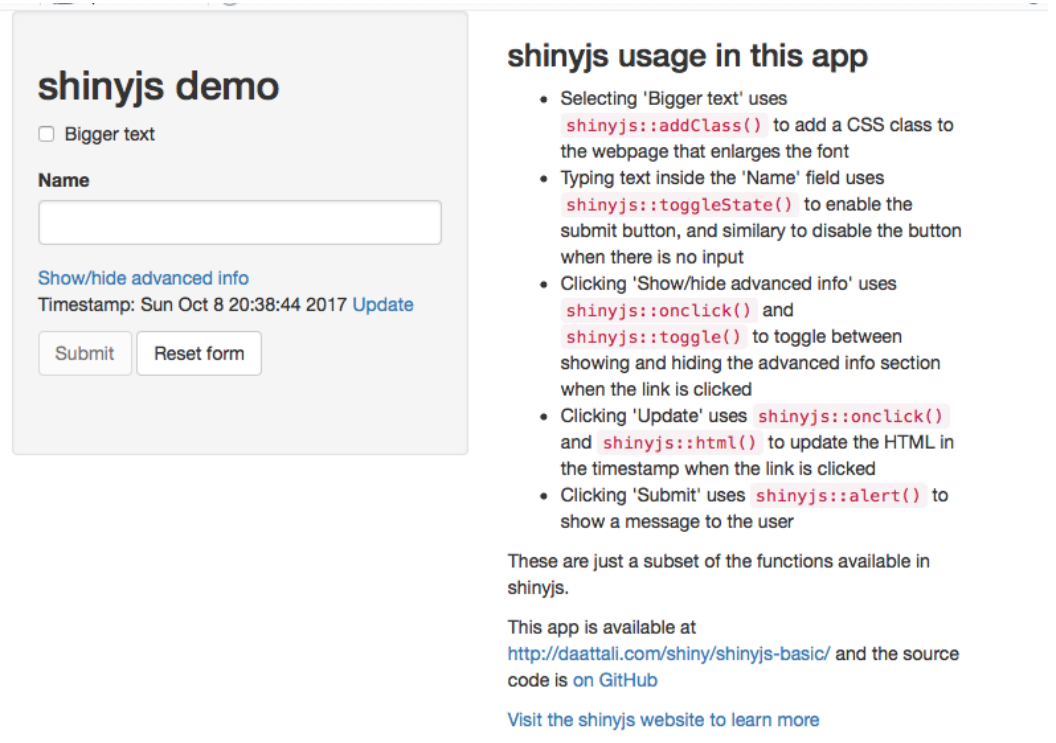
homepage: https://rstudio.github.io/shinydashboard/index.html



https://gallery.shinyapps.io/087-crandash/

4.shinyjs

homepage: https://github.com/daattali/shinyjs



http://daattali.com/shiny/shinyjs-basic/