

# Interactive Graphics in R

Stat 480  
Spring 2015

# Outline

- server.r and ui.r
- widgets
- connecting code and widgets

# shiny Apps

- Very good tutorial at <http://shiny.rstudio.com/>
- `library(shiny)`  
`runExample("01_hello")`

# Your Turn

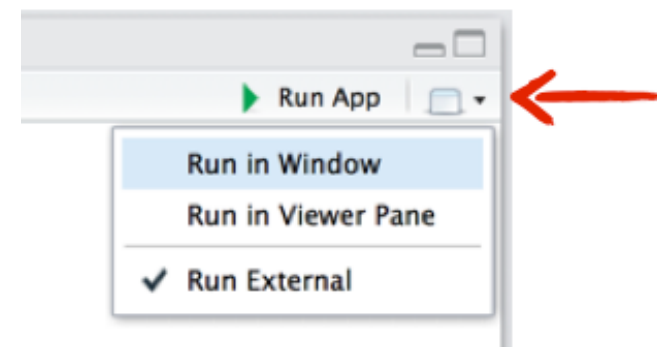
Run a few of the shiny examples (see function `runExample` for the different choices) and play with the different available features

Have a first look at some of the files `server.R` and `ui.R`

Open the first example and save the code into files `server.r` and `ui.r`

# server.r and ui.r

- all of the interactivity is regulated through two files: server.r and ui.r
- both of these files have to go into the same folder
- ... let's do that for the first example ...



# Your Turn

In the first example, make the following changes:

- \* Change the title from “Hello Shiny!” to “Hello World!”.
- \* Set the minimum value of the slider bar to 5.
- \* Change the histogram color from "darkgray" to “forestgreen”.

Advanced: change from a base histogram to a qplot histogram

# ui.r: input and output

```
# ui.R
```

```
shinyUI(fluidPage(  
  titlePanel("title panel"),  
  sidebarLayout(  
    sidebarPanel("sidebar panel"),  
    mainPanel("main panel")  
  )  
))
```

Inside each of the panel commands we can use html text  
see <http://shiny.rstudio.com/tutorial/lesson2/> for more details.

# server.r

In server.r all of the input parameter are evaluated to make the output elements.  
Each output element has to be rendered

```
# server.R
```

```
shinyServer(function(input, output) {  
  
  }  
})
```



# adding widgets

The screenshot shows a web browser window titled 'Basic widgets' with the address bar displaying '127.0.0.1:4733'. The page content is organized into a grid of widget demonstrations:

- Buttons:** Includes a light gray 'Action' button and a blue 'Submit' button.
- Single checkbox:** A checkbox labeled 'Choice A' which is checked.
- Checkbox group:** Three checkboxes labeled 'Choice 1' (checked), 'Choice 2', and 'Choice 3'.
- Date input:** A text field containing the date '2014-01-01'.
- Date range:** Two text fields, both containing '2014-01-24', separated by the word 'to'.
- File input:** A 'Choose File' button followed by the text 'No file chosen'.
- Help text:** A text block stating: 'Note: help text isn't a true widget, but it provides an easy way to add text to accompany other widgets.'
- Numeric input:** A text field containing the number '1' with up and down arrow controls.
- Radio buttons:** Three radio buttons labeled 'Choice 1' (selected), 'Choice 2', and 'Choice 3'.
- Select box:** A dropdown menu currently showing 'Choice 1'.
- Sliders:** Two horizontal sliders. The top slider has a range from 0 to 100 with a blue handle at 50. The bottom slider has a range from 0 to 100 with blue handles at 25 and 75.
- Text input:** A text field with the placeholder text 'Enter text...'.

see <http://shiny.rstudio.com/tutorial/lesson3/>

# input parameters: widgets

actionButton	Action Button
checkboxGroupInput	A group of check boxes
checkboxInput	A single check box
dateInput	A calendar to aid date selection
dateRangeInput	A pair of calendars for selecting a date range
fileInput	A file upload control wizard
helpText	Help text that can be added to an input form
numericInput	A field to enter numbers
radioButtons	A set of radio buttons
selectInput	A box with choices to select from

# sliderInput

```
sliderInput(inputId, label, min, max, value, step = NULL, round = FALSE,  
  format = NULL, locale = NULL, ticks = TRUE, animate = FALSE,  
  width = NULL, sep = ",", pre = NULL, post = NULL)
```

- e.g.:

```
sliderInput("nbins", "number of bins", min=1,  
max=50, range=10)
```

```
sliderInput("range", "range of X", min=43,  
max=96, range=c(43, 96))
```

# Your Turn

- For a dataset of your choice, set up a folder for a shiny App and start building the interface shown below (input values only)

## Iris k-means clustering

**X Variable**

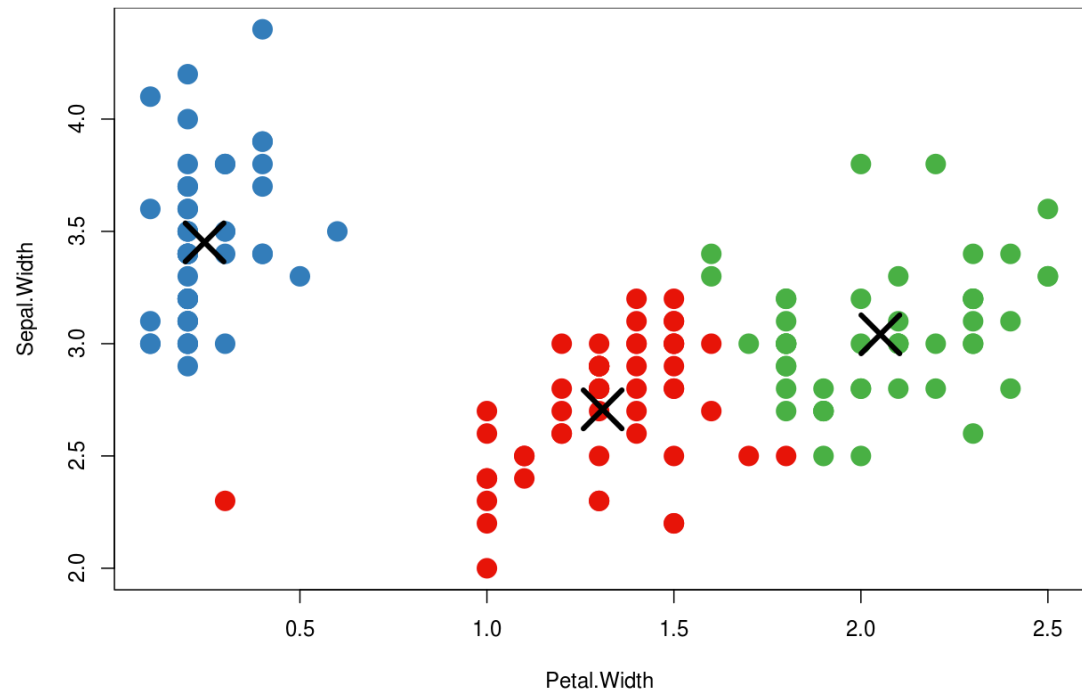
Petal.Width ▼

**Y Variable**

Sepal.Width ▼

**Cluster count**

3



# output parameters (usually in mainPanel)

Output function	creates
htmlOutput	raw HTML
imageOutput	image
plotOutput	plot
tableOutput	table
textOutput	text
uiOutput	raw HTML
verbatimTextOutput	text

# server.r

In server all of the input parameter are evaluated to make the output elements.  
Each output element has to be rendered

```
# server.R
```

```
shinyServer(function(input, output) {
```

```
  output$text1 <- renderText({  
    "You have selected this"  
  })
```

```
})  
)
```

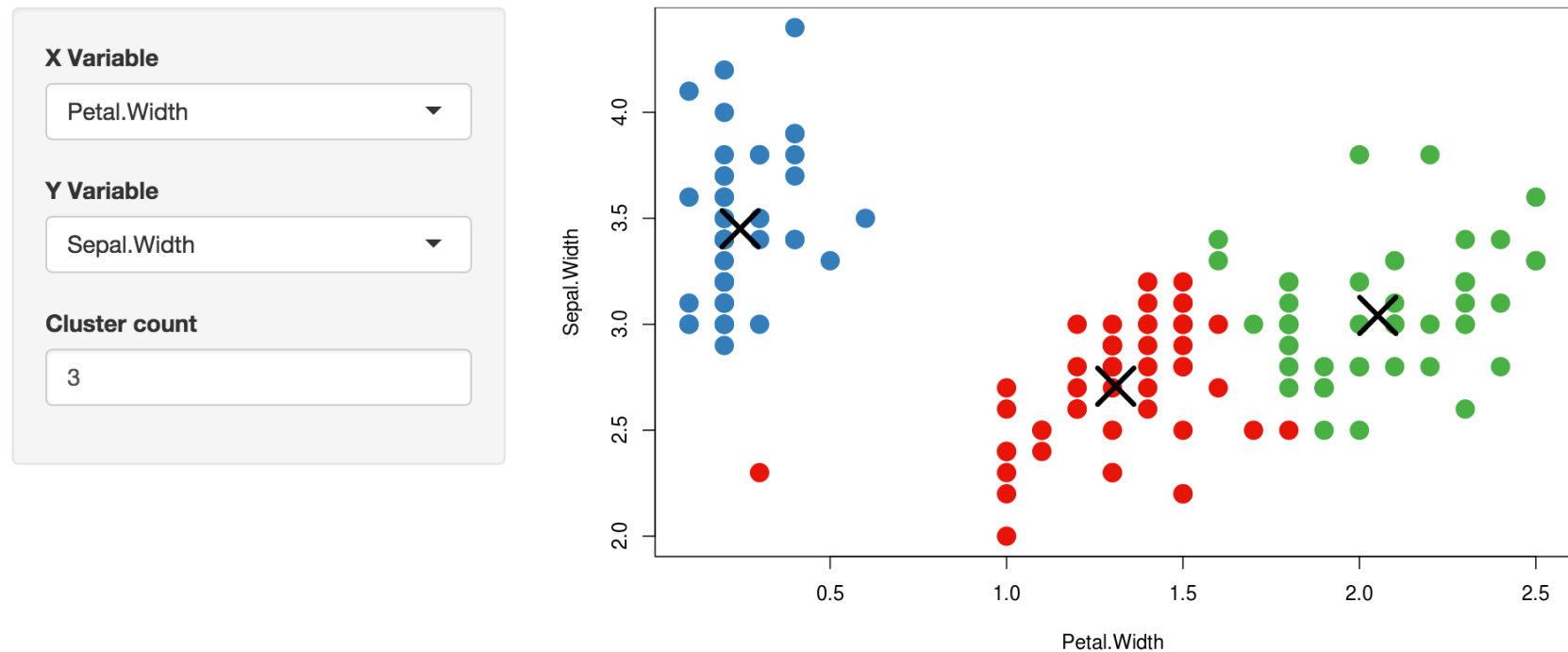
# render functions

function	object created
renderImage	images (saved as a link to a source file)
renderPlot	plots
renderPrint	any printed output
renderTable	data frame, matrix, other table like structures
renderText	character strings
renderUI	a Shiny tag object or HTML

# Your Turn

- add the output elements to ui.r and work on the rendering in server.r

## Iris k-means clustering





# Congratulations

- you just built your first shiny app!
- here are more examples:  
<http://shiny.rstudio.com/gallery/>