Visualization for Data Science in R

Angela Zoss

Data Matters 2019

https://github.com/amzoss/RVis-DM2019

Objectives/Outline

Day 1: Static visualizations

- Visualization and data science
- Basic ggplot2 syntax
- Basics of geoms and aes
- Manipulating data
- Categorical variables
- Advanced topics: mapping, saving charts out

Day 2: Interactivity

- Simple interactive plots
- Arranging charts into dashboards
- Incorporating Shiny elements into documents, dashboards
- Advanced topics: full Shiny apps

Interactivity

Why make charts interactive?

- Easier for data exploration
  - Drill-down to data subsets of interest
  - Details on demand
  - Customize look-and-feel of chart
- Can be more engaging for users

Visual information seeking mantra

Overview first,
zoom and filter,
then details-on-demand

Shneiderman, B. (1996). The eyes have it: A task by data type taxonomy for information visualization.
In *VL '96 Proceedings of the 1996 IEEE Symposium on Visual Languages.*

Interactivity in R Markdown

- R Markdown gets compiled into HTML
- Some R packages can create interactive elements by converting R output to JavaScript code for the final HTML document
- We will use the **plotly** package to create interactive charts

http://www.htmlwidgets.org/

Other interactive chart packages

- ggiraph for extending ggplot2 with interactive geoms
- rCharts for an R version of Polycharts, NVD3, and MorrisJS
- rBokeh for an R version of Bokeh
- altair for an R version of Altair
- leaflet for interactive maps

Exercise 1: Make yesterday’s charts interactive

Dashboards in R Markdown

“Normal” R Markdown

- R Markdown elements like headings, text

```
# Heading 1
## Heading 2
Regular text
* Bulleted text
```

- Code chunks

```
```{r}

```
```

Markdown for flexdashboards

Page

==========

Column (or Row)

Chart titles

Regular text
* Bulleted text

```{r}

```

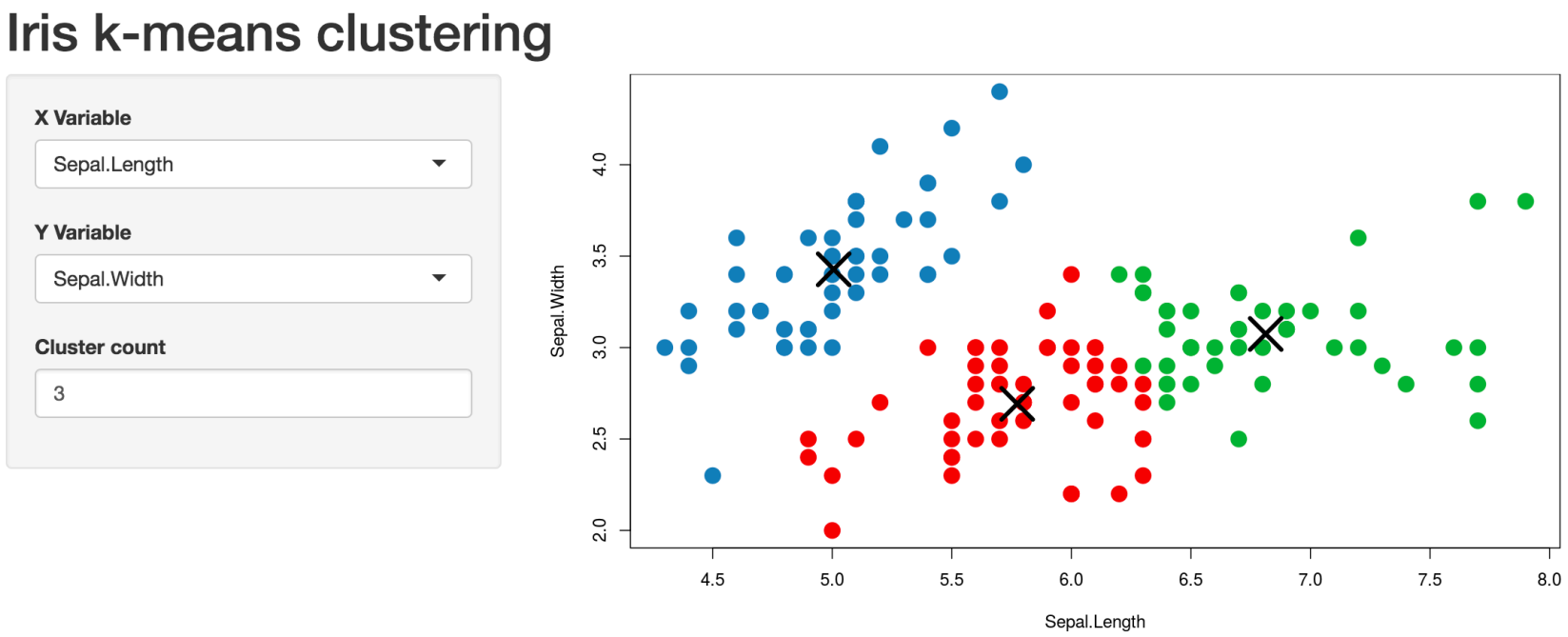
Exercise 2: Star Wars dashboard

Exercise 3: Vis Portfolio

Shiny

What is Shiny?

An interactive interface onto an R program



http://shiny.rstudio.com/

How can you use Shiny for visualization?

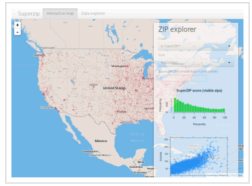
- Use Shiny to control some kind of simulation interactively, then visualize the results
- Use Shiny to change components within the chart (e.g., switch the mappings)
- Use Shiny to filter data to subsets to highlight patterns
- Change type of regression, plot results

Shiny examples

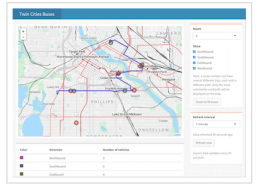
Gallery

Interactive visualizations

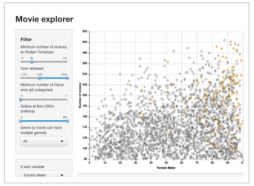
Shiny is designed for fully interactive visualization, using JavaScript libraries like d3, Leaflet, and Google Charts.



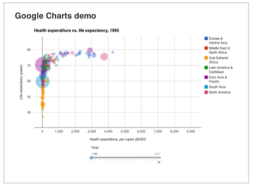
SuperZip example



Bus dashboard



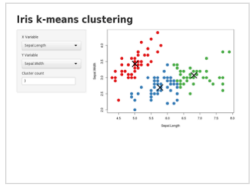
Movie explorer



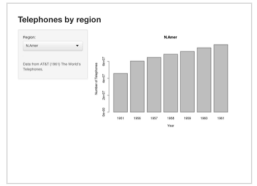
Google Charts

Start simple

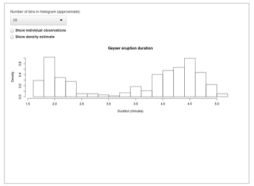
If you're new to Shiny, these simple but complete applications are designed for you to study.



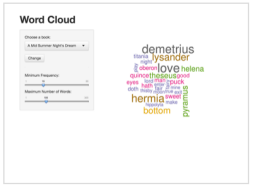
Kmeans example



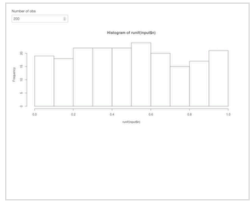
Telephones by region



Faithful



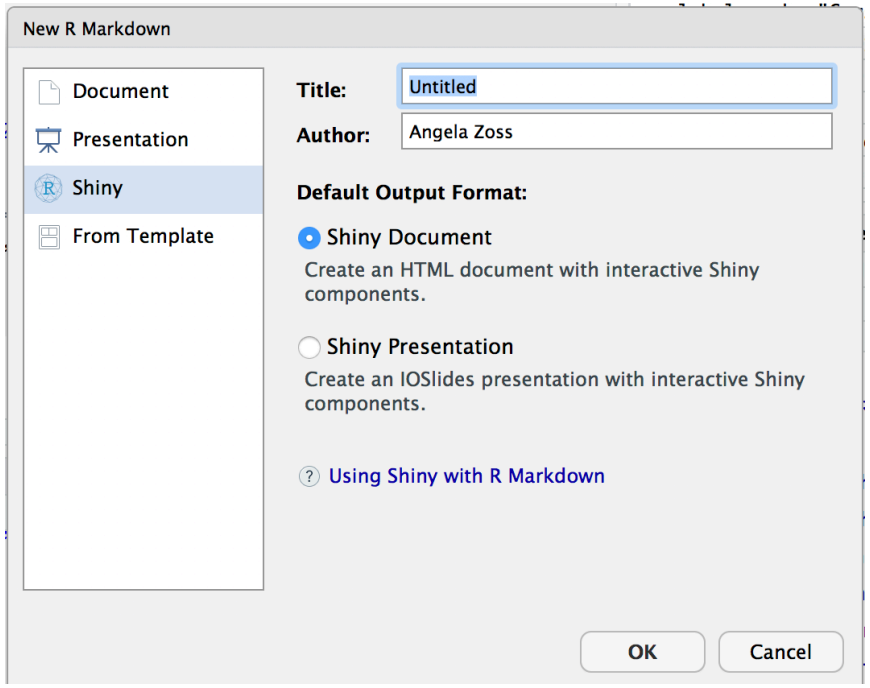
Word cloud



Single-file shiny app

https://shiny.rstudio.com/gallery/

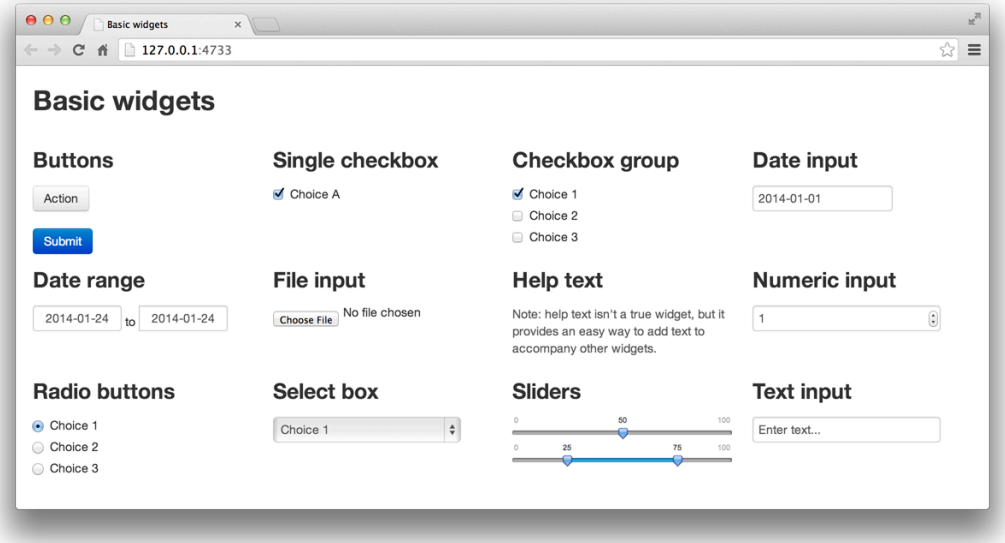
Shiny in R Markdown



Components

Some kind of **input widget**

(e.g., selectInput, sliderInput)



Some kind of **render object**

(e.g., renderPlot, renderTable)

renderPlot wraps around something like a ggplot() plot

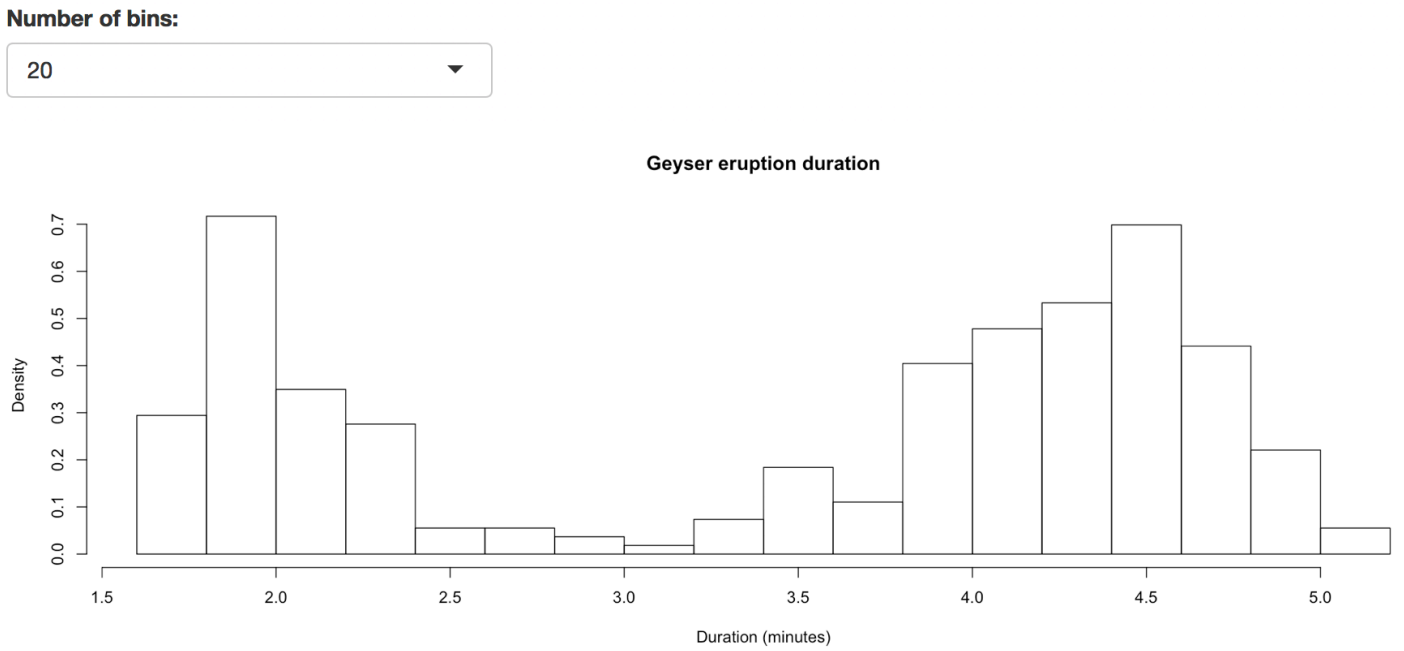
Layout

- In this case, Shiny elements are included to change/control individual charts
- The overall layout of the file is just using normal R Markdown syntax, and Shiny elements get embedded whenever the right code chunk comes up

Both components go in same code chunk

```r
selectInput("n_breaks", label = "Number of bins:",
              choices = c(10, 20, 35, 50), selected = 20)

renderPlot({
  hist(faithful$eruptions, probability = TRUE, breaks = as.numeric(input$n_breaks),
       xlab = "Duration (minutes)", main = "Geyser eruption duration")
})
```



Anatomy of a widget

- **Name** for the widget (internal only)
- **Label** (will be visible)
- Check documentation for other required arguments

Exercise 4: Game of Thrones Markdown

Select Variable for Color:

gender

Change Label Font Size:

1 4 10

Shiny in Dashboards

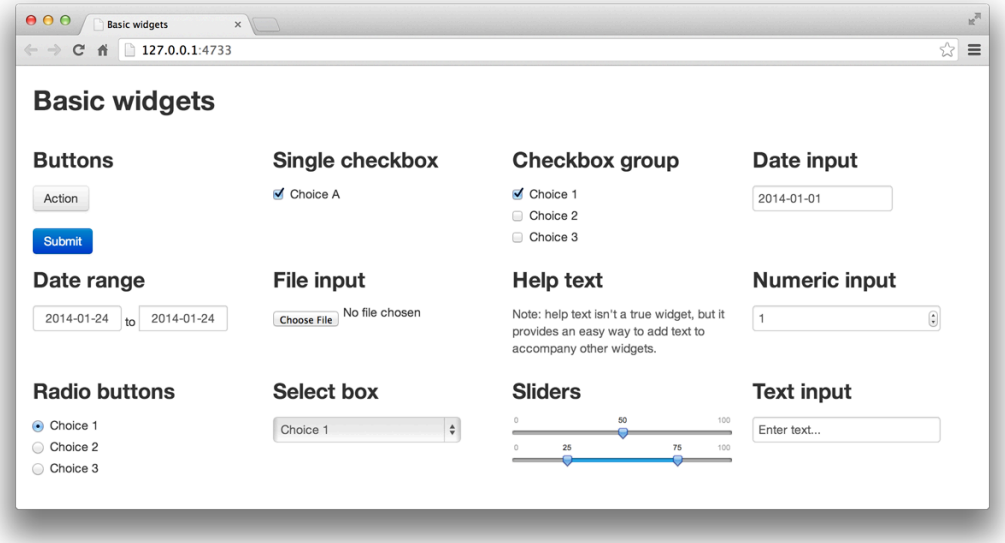
Components (same as R Markdown)

Some kind of **input widget**

(e.g., selectInput, sliderInput)

Some kind of **render object**

(e.g., renderPlot, renderTable)



renderPlot wraps around something like a ggplot() plot

Layout (similar to normal flexdashboards)

Page

==========

Column {.sidebar)

Chart titles

Regular text
* Bulleted text

```{r}

(including Shiny input, render objects)

```

Exercise 5: Interactive Vis Portfolio

R Visualization Portfolio with Shiny Angela Zoss — 8/14/2018 Game of Thrones Star Wars

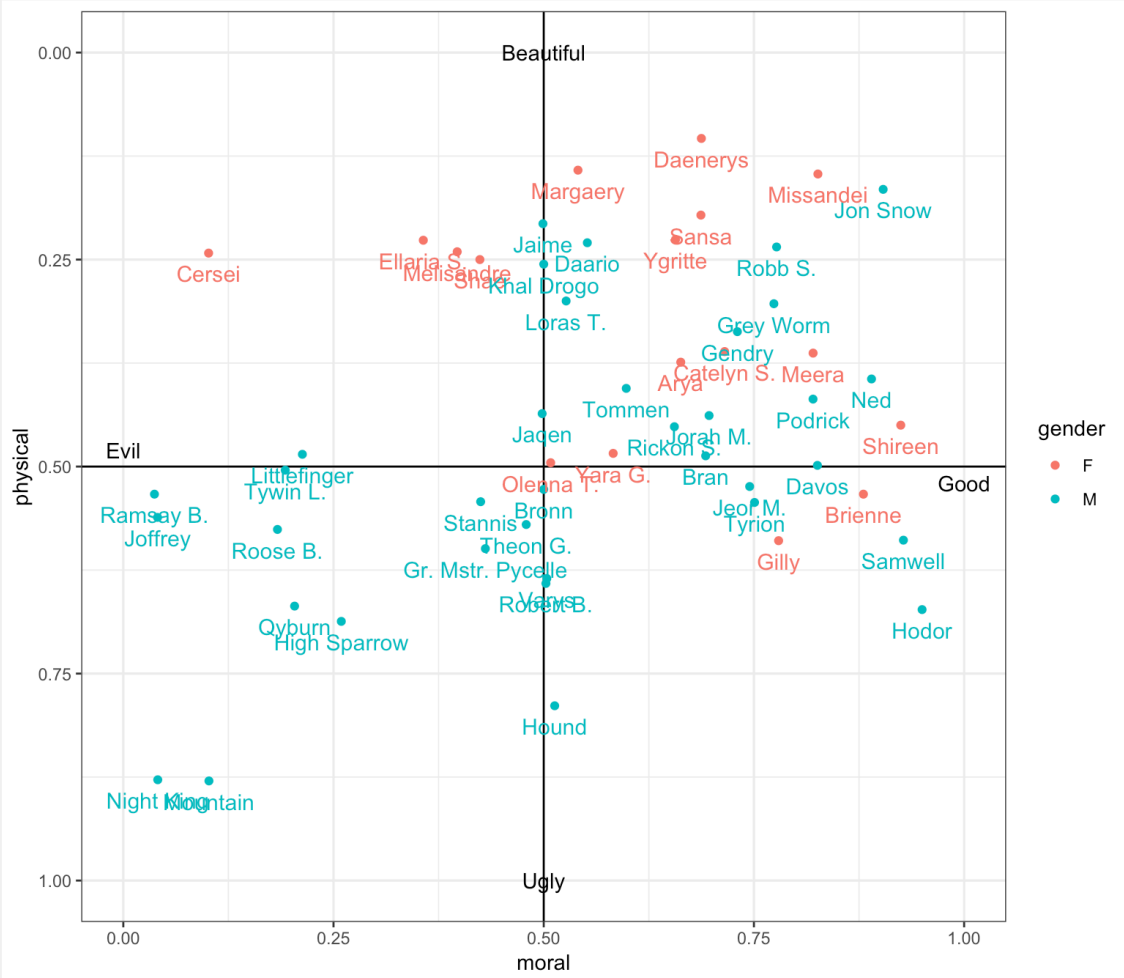
Select Variable for Color:

gender

Change Label Font Size:

1 4 10

1 2 3 4 5 6 7 8 9 10



Shiny Apps

How do you build a Shiny app?

User Interface (UI)

the website people will see and interact with

Server

takes values from the interface, does some calculations, and creates more content for the interface

Step 1: Create the interface

What to put in the UI?

- Layout elements
- Extra text/HTML elements
- Control widgets
- Placeholders for reactive output

Page layout

1. fluidPage
   - titlePanel
   - sidebarLayout
     - sidebarPanel
     - mainPanel
   - fluidRow
     - column
     - wellPanel
   - tabsetPanel
   - navlistPanel
2. fixedPage
   - fixedRow
3. navbarPage
   - tabPanel
   - navbarMenu
     - tabPanel

http://shiny.rstudio.com/articles/layout-guide.html

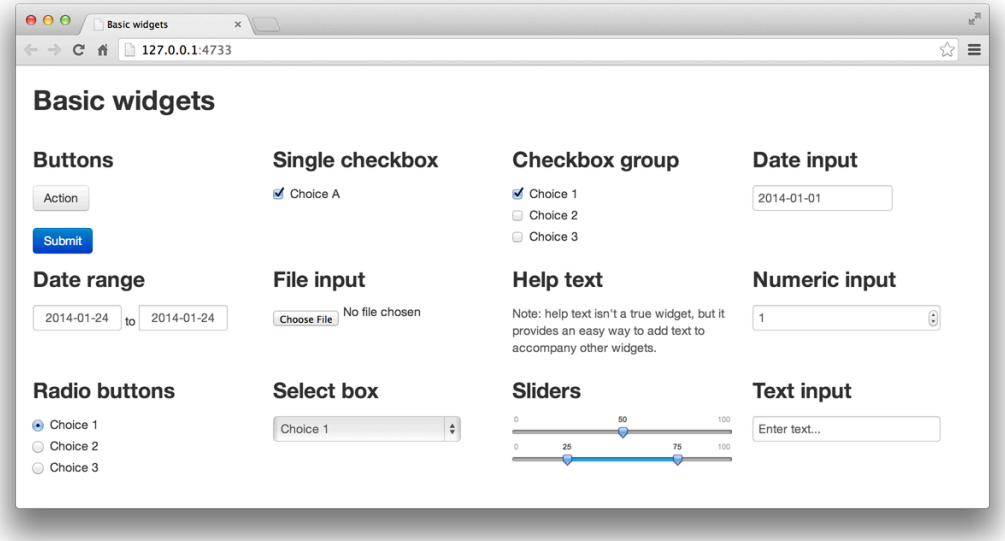
HTML elements

- Shiny has special wrapper functions for this – e.g., h2(), p()

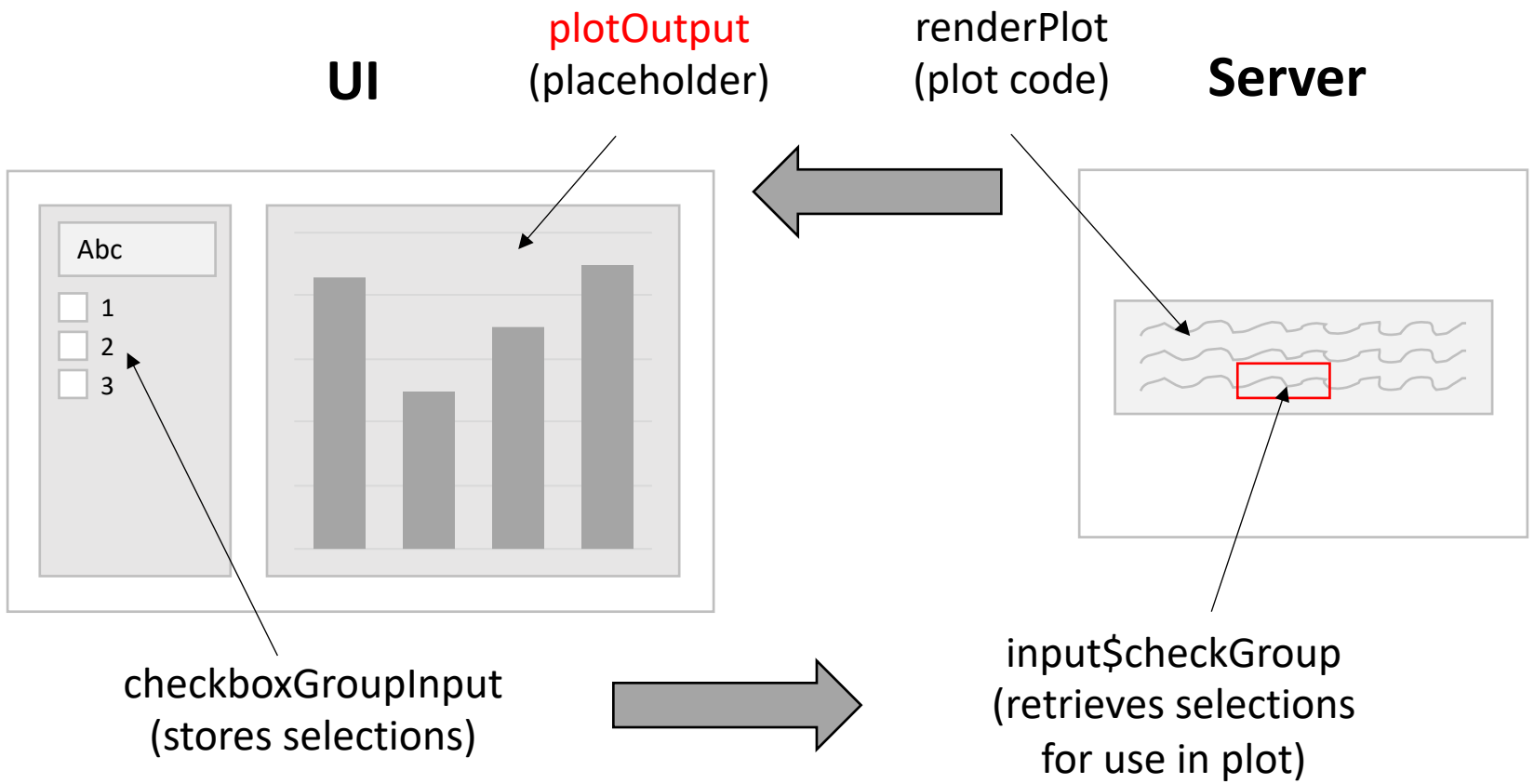
http://shiny.rstudio.com/tutorial/written-tutorial/lesson2/

Control widgets

- Button
- Checkboxes
- Date, date range input
- File input
- Numeric input
- Radio buttons
- Drop-down (select) box
- Slider bar
- Text input
- Text



http://shiny.rstudio.com/tutorial/written-tutorial/lesson3/
http://shiny.rstudio.com/gallery/widget-gallery.html

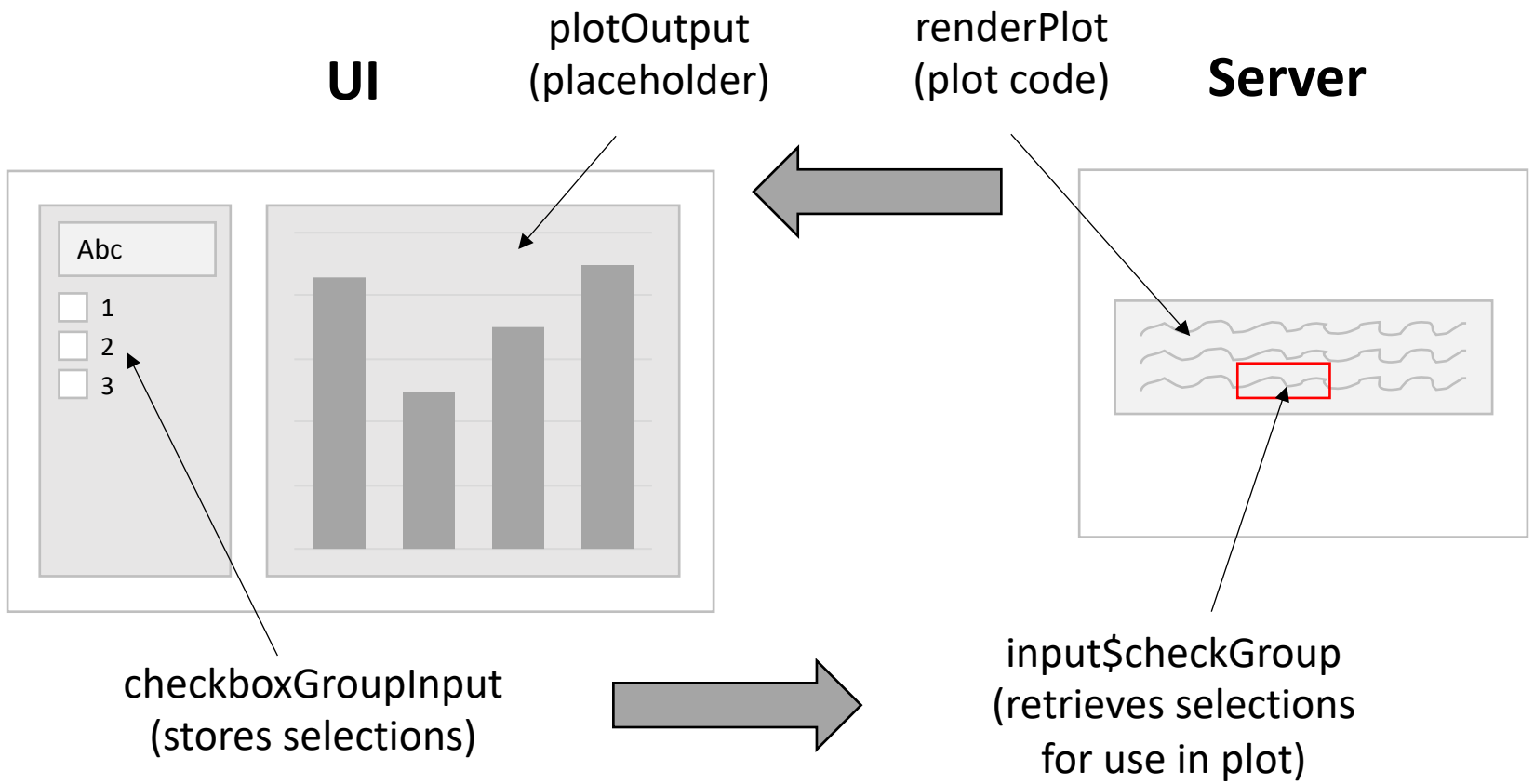

UI
plotOutput
(placeholder)
renderPlot
(plot code)
Server
Abc
1
2
3
checkboxGroupInput
(stores selections)
input$checkGroup
(retrieves selections
for use in plot)

Reactive output objects

UI	Server
htmlOutput	renderUI
imageOutput	renderImage
plotOutput	renderPlot
tableOutput	renderTable
textOutput	renderText
uiOutput	renderUI
verbatimTextOutput	renderPrint

http://shiny.rstudio.com/tutorial/written-tutorial/lesson4/

Step 2: Set up server to create dynamic objects


UI
plotOutput
(placeholder)
renderPlot
(plot code)
Server
Abc
1
2
3
checkboxGroupInput
(stores selections)
input$checkGroup
(retrieves selections
for use in plot)

What to put in the server

- R code
- Render objects with same names and types as the ones listed in UI
- Input objects with the same names as the control widgets

UI:

```
sliderInput("slider1")

textOutput("text1")
```

Server:

```
output$text1  <-  renderText({
                    input$slider1
                  })
```

http://shiny.rstudio.com/tutorial/lesson4/

Step 3: Test

Running the app

Set options in RStudio:

- Window
- Viewer
- External

Exercise 6: Portfolio as full Shiny App

R Visualization Portfolio with Shiny | Game of Thrones | Star Wars

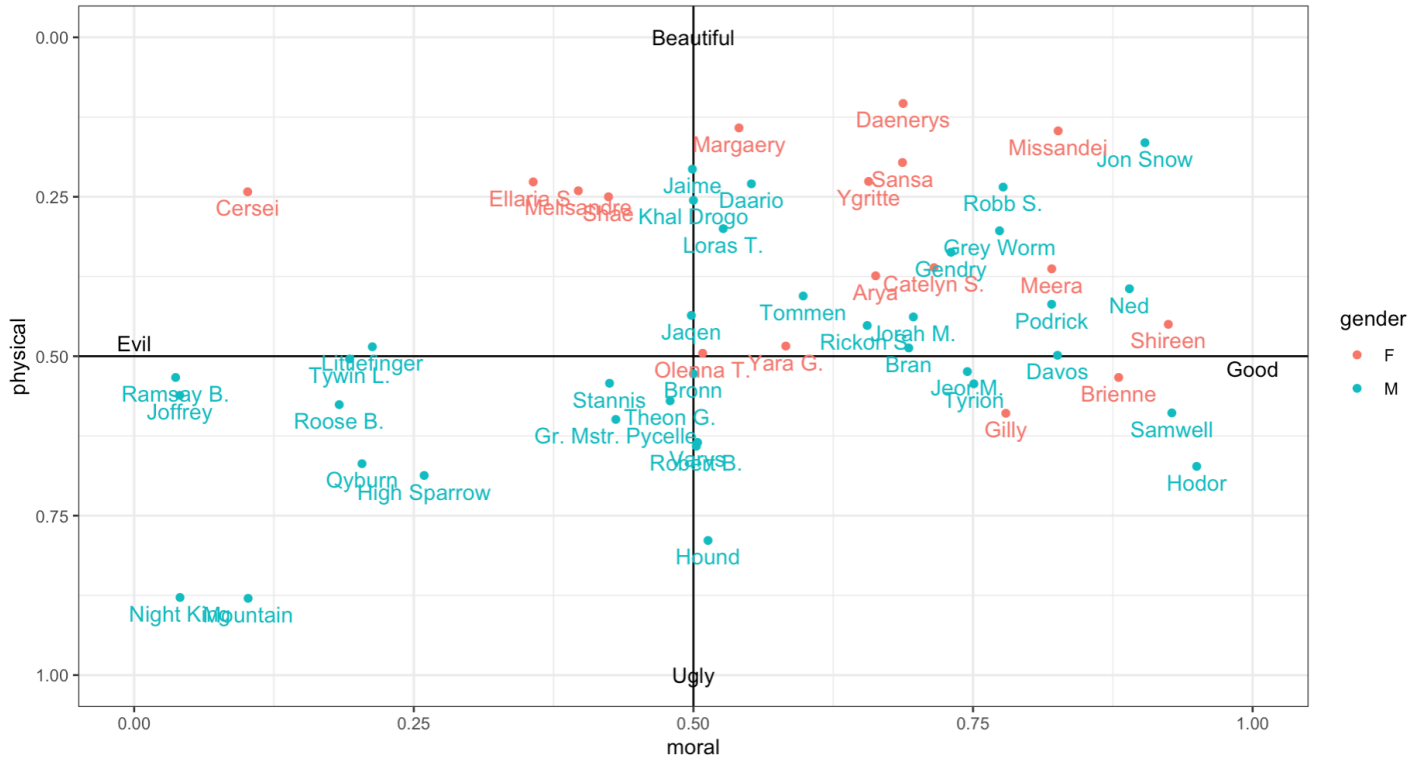
Select Variable for Color:

gender

Change Label Font Size:

1 4 10

1 2 3 4 5 6 7 8 9 10



Sharing an app

- Shiny Apps
  http://www.shinyapps.io/
- Shiny Server (free – host on your own server)
  https://github.com/rstudio/shiny-server/blob/master/README.md
- Shiny Server Pro (fee)
  https://www.rstudio.com/products/shiny/shiny-server/

Shiny resources

- Shiny Gallery
- Shiny Tutorial
- Shiny Articles
- Shiny function reference
- Shinyapps.io
- RStudio::conf 2019 workshop: Introduction to Shiny and R Markdown
- Shiny in Production (slides), Shiny in Production (book)
- Interactive web-based data visualization with R, plotly, and shiny
- Accessing and responding to plotly events in shiny

Thanks for your time this week!

angela.zoss@duke.edu