Developing Data Products: Shiny

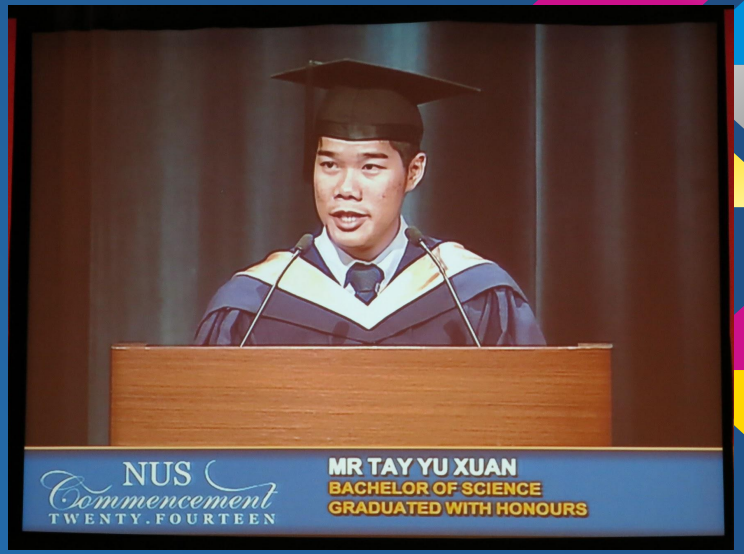
Yu Xuan & Yi Xiang

September 2015

About Me

- Data Scientist at IDA
  - Government Analytics team
  - Work with government agencies to improve public service through Data Science
  - ~ 20 members
  - Sandcrawler @ one-north
- Previously, Actuarial Analyst at Swiss Re
- https://www.linkedin.com/in/yxtay

About Me



- BSc (Hons) in Statistics
  - NUS Class of 2014
- Tools of trade
  - R, Python, Spark, HiveQL
- Interests
  - Infocomm Technology, MOOC
  - Astronomy, Formula 1, Calisthenics

Content (Part 1)

Shiny

1. Overview
2. Examples & Resources
3. Basics & Layout
4. Widgets & Output
5. Reactivity
6. Styles
7. Deployment

JOHNS HOPKINS
BLOOMBERG SCHOOL
of PUBLIC HEALTH

Developing Data Products

Part of the Data Science Specialization »

Learn the basics of creating data products using Shiny, R packages, and interactive graphics. This is the ninth course in the Johns Hopkins Data Science Specialization.

About the Course

A data product is the production output from a statistical analysis. Data products automate complex analysis tasks or use technology to expand the utility of a data informed model, algorithm or inference. This course covers the basics of creating data products using Shiny, R packages, and interactive graphics. The course will focus on the statistical fundamentals of creating a data product that can be used to tell a story about data to a mass audience.

Course Syllabus

Students will learn how communicate using statistics and statistical products. Emphasis will be paid to communicating uncertainty in statistical results. Students will learn how to create simple Shiny web applications and R packages for their data products.

Sessions

July 7, 2014 - August 4, 2014

View course record

Eligible for

Data Science Specialization
Course Certificate

Course at a Glance

Developing Data Products

- Communicating findings of statistical data analysis
- Interactive graphics
- Shiny web application
- R Presentation, slidify
- rCharts, GoogleVis, plotly
- R packages

What is Shiny?

- Web application framework for R
- Project by RStudio
- UI theme based on Bootstrap HTML/CSS framework originally created at Twitter

Tabsets

Distribution type:

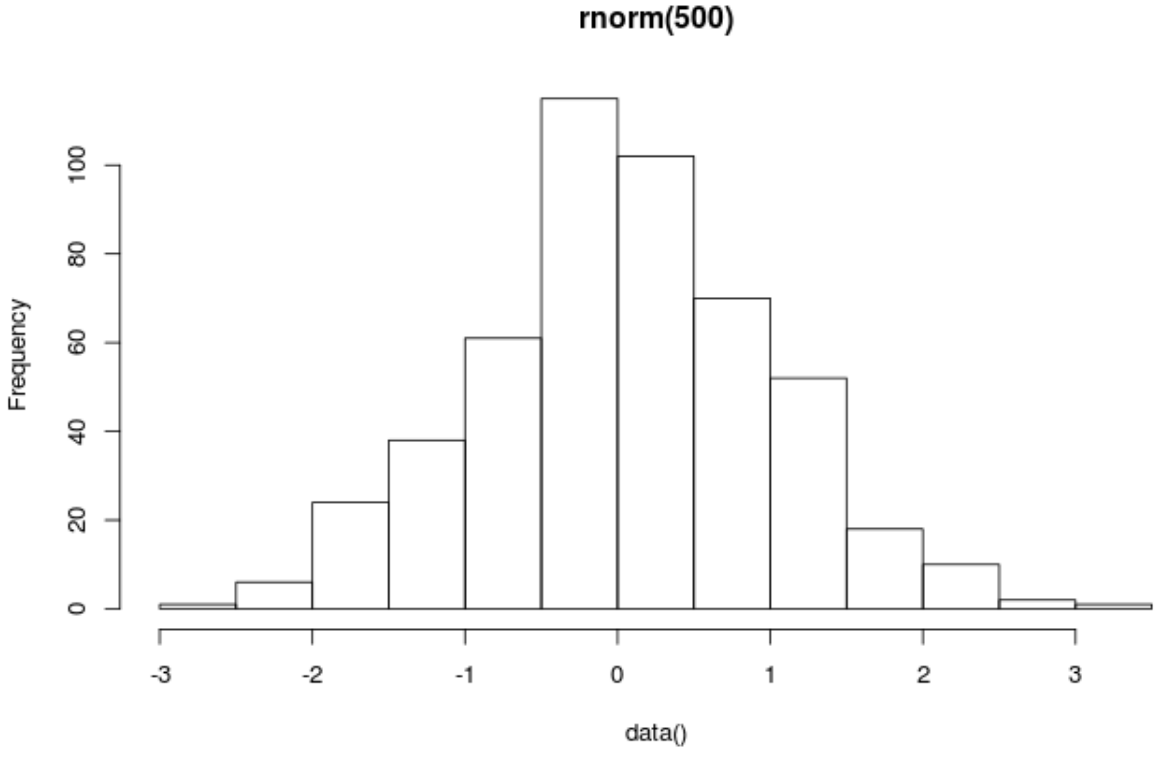
- (•) Normal
- ( ) Uniform
- ( ) Log-normal
- ( ) Exponential

Number of observations:

1 500 1,000

1 101 201 301 401 501 601 701 801 901 1,000

Plot | Summary | Table



Pros

- Easy to build interactive web application straight from R
- No HTML, CSS or Javascript required
- Easy to distribute with shinyapps.io
- Reactive programming model for “live” interactivity
- Variety of pre-built widgets for inputs and outputs
- Well documented examples for reference
- Responsive design: works on PC’s and mobile

Cons

- Commas
- Syntax slightly different from R, can be confusing for unseasoned developers
- Debugging can be difficult
- Not standalone (requires Shiny Server)

Examples

- runExamples(), 01-11
- Shiny gallery: http://shiny.rstudio.com/gallery/
- RStudio Github: https://github.com/rstudio/shiny-examples
  - use: http://gallery.shinyapps.io/example-name
- Shiny User Showcase: http://www.rstudio.com/products/shiny/shiny-user-showcase/
- Show Me Shiny: http://www.showmeshiny.com/

Resources

- Shiny Webinar: http://shiny.rstudio.com/tutorial/video/
- Shiny Tutorial: http://shiny.rstudio.com/tutorial/
- Shiny Basics: http://shiny.rstudio.com/articles/basics.html
- Shiny Articles: http://shiny.rstudio.com/articles/
- Shiny Reference: http://shiny.rstudio.com/reference/shiny/latest/

Resources

- Shiny Cheatsheets
  - http://shiny.rstudio.com/images/shiny-cheatsheet.pdf
  - http://www.rstudio.com/wp-content/uploads/2015/06/shiny-cheatsheet-old.pdf
- Shiny Reference: http://shiny.rstudio.com/reference/shiny/latest/

Shiny Tutorial

- Lesson 1: Shiny app introduction and basics
- Lesson 2: Layout and HTML content
- Lesson 3: Widgets
- Lesson 4: Widget input and Reactive output
- Lesson 5: using R scripts and data
- Lesson 6: reactive expressions
- Lesson 7: deploying Shiny app

Basics

- Tutorial: http://shiny.rstudio.com/tutorial/lesson1/
- 2 R files componenets
- ui.R: shinyUI(<UI elements>)
- server.R: shinyServer(function(input, output, session) {<computations>})
  - input: list with named elements assigned by widgets
  - output: list with named elements assigned by render functions
  - session: session info, used by certain functions

Basics

- Single-file Shiny App
- app.R: shinyApp(ui, server)
  - Example: http://shiny.rstudio.com/gallery/single-file-shiny-app.html
  - Article: http://shiny.rstudio.com/articles/single-file.html

Basics

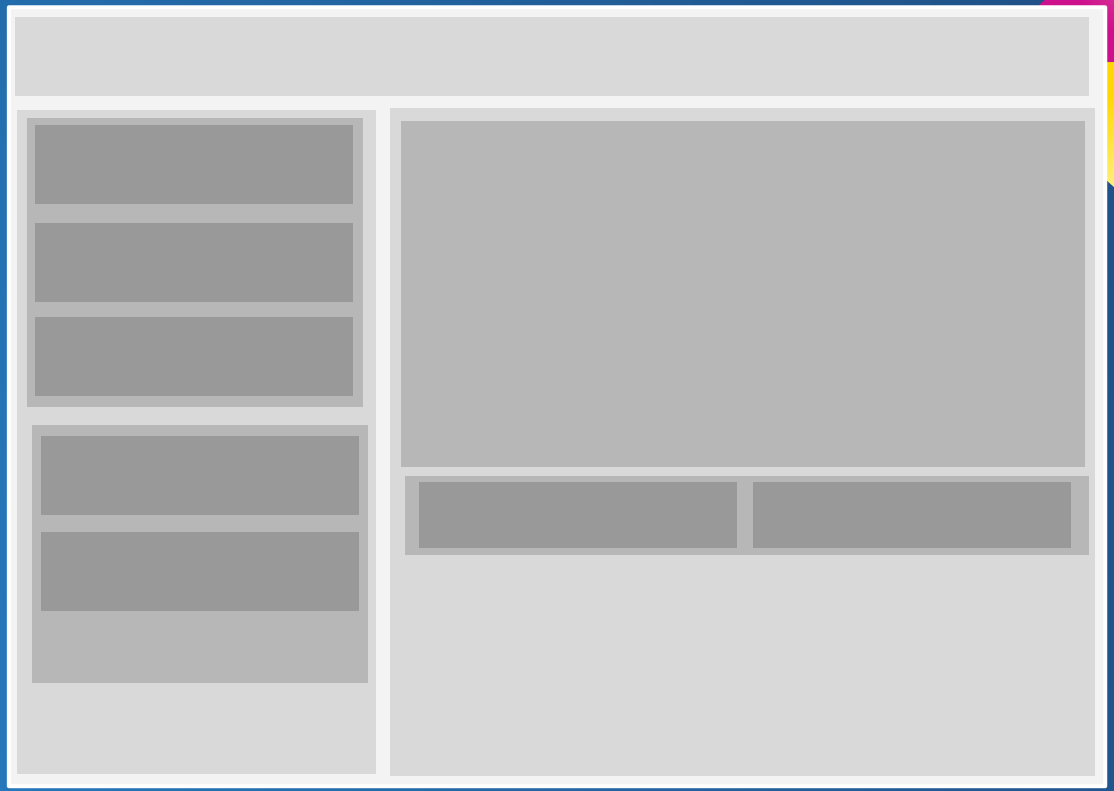
- Running Shiny
  - Run app button in RStudio
  - runApp(appDir) in R
    - set display.mode = “showcase”

Basics

- Article: http://shiny.rstudio.com/articles/basics.html
- Article: http://shiny.rstudio.com/articles/build.html
- Article: http://shiny.rstudio.com/articles/app-formats.html
- Scoping rules
  - Article: http://shiny.rstudio.com/articles/scoping.html

Layout

- Article: http://shiny.rstudio.com/articles/layout-guide.html
- responsive design
- columns and rows
- 12-wide column grid



Layout - Basics

- fluidPage()
  - titlePanel()
  - sidebarLayout()
    - sidebarPanel()
    - mainPanel()
  - splitLayout()
  - column()
  - fluidRow()
  - wellPanel()
- verticalLayout()
- flowLayout()

Layout - More

- tabsetPanel()
  - http://shiny.rstudio.com/gallery/tabsets.html
- navlistPanel()
  - http://shiny.rstudio.com/gallery/navlistpanel-example.html
- navbarPage()
  - http://shiny.rstudio.com/gallery/navbar-example.html

Hands-On: shiny 1

HTML Content

- Tutorial: http://shiny.rstudio.com/tutorial/lesson2/
- Article: http://shiny.rstudio.com/articles/html-tags.html
- Article: http://shiny.rstudio.com/articles/tag-glossary.html
- Article: http://shiny.rstudio.com/articles/html-ui.html

Widgets

- Tutorial: http://shiny.rstudio.com/tutorial/lesson3/
- Example: http://shiny.rstudio.com/gallery/widget-gallery.html
- UI elements for interactive input

Widgets

- Buttons
  - actionButton()
  - submitButton()
- Free Text
  - textInput()
- Numeric
  - numericInput()
  - sliderInput()
- Date
  - dateInput()
  - dateRangeInput()
- Choices
  - selectInput()
  - selectizeInput()
  - checkboxInput()
  - checkboxGroupInput()
  - radioButtons()

Widgets

- File
  - fileInput()
  - Example: http://shiny.rstudio.com/gallery/file-upload.html
  - downloadButton()
  - Example: http://shiny.rstudio.com/gallery/file-download.html

Output

- Tutorial: http://shiny.rstudio.com/tutorial/lesson4/
- each output function paired with a render function
- converts R output into HTML
- Text
  - UI: textOutput(), verbatimTextOutput()
  - server: renderText(), renderPrint()
- Table
  - UI: tableOutput(), dataTableOutput()
  - server: renderTable(), renderDataTable(

Output

- Plot
  - UI: plotOutput()
  - server: renderPlot()
- Image
  - UI: imageOutput()
  - server: renderImage()

Hands-On:
shiny 2
shiny 3

Debugging

- Difficult to debug as codes cannot be run line-by-line
- use renderPrint() and verbatimTextOutput()

Reactivity

- Computations run only when required
  - When app loads
  - When inputs change
- Output is cached

Reactive Expressions

- Tutorial: http://shiny.rstudio.com/tutorial/lesson6/
- Cache and reuse intermediate results
- Recalculate only when inputs change
- Reduce expensive computations
- Maintain random elements

Reactive Expressions

- reactive()
- observe()
  - http://shiny.rstudio.com/gallery/update-input-demo.html
- isolate()
  - http://shiny.rstudio.com/gallery/isolate-demo.html
  - Article: http://shiny.rstudio.com/articles/isolation.html

Reactive Expressions

- eventReactive()
  - Example: http://shiny.rstudio.com/gallery/actionbutton-demo.html
- observeEvent()
- reactiveValues()

Reactivity

- Article: http://shiny.rstudio.com/articles/reactivity-overview.html
- Article: http://shiny.rstudio.com/articles/action-buttons.html

Hands-On: shiny 4

Dynamic UI

- UI that changes depending on user input
- Additional interactivity
- Important for complex applications

Dynamic UI

- Article: http://shiny.rstudio.com/articles/dynamic-ui.html
- conditionalPanel()
  - http://shiny.rstudio.com/gallery/conditionalpanel-demo.html
- update*Input()
  - http://shiny.rstudio.com/gallery/update-input-demo.html
- http://shiny.rstudio.com/gallery/dynamic-ui.html
  - renderUI()
  - uiOutput()

Styling Shiny

- Article: http://shiny.rstudio.com/articles/css.html
- Themes
  - fluidPage(theme = bootstrap.css)
  - shinythemes package
    - https://rstudio.github.io/shinythemes/
  - Bootswatch
    - https://bootswatch.com/

Styling Shiny

- Icons
  - icon()
  - Glyphicons
    - http://getbootstrap.com/components/#glyphicons
  - Font Awesome
    - http://fontawesome.io/icons/

Hands-On: shiny 5

Deploying Shiny

- Article: http://shiny.rstudio.com/articles/shinyapps.html
- shinyapps.io: http://www.shinyapps.io/
- Requirements
  - devtools package
    - install.packages("devtools"); library(devtools)
  - shinyapps or rsconnect package
    - install_github("shinyapps"); library(shinyapps)

Deploying Shiny

- Create shinyapp.io account
- Configure shinyapps
  - setAccountInfo(name="<ACCOUNT>", token="<TOKEN>", secret="<SECRET>")
  - accounts()
  - accountInfo(name)
- Deploy app
  - deployApp(appDir, appName)

App Ideas

- Visualisation of data
- Demonstration of concepts
- Explore inbuilt datasets

The End