

Shiny for Creating Baseball Visualizations

Jim Albert
Emeritus Professor
Bowling Green State University

SLAM Sport Analytics Summer Academy
June 7, 2021

Outline

- Introduce Shiny apps
- Focus on several baseball applications (Pitch Locations and Brushing Batting Averages)
- Provide R code for creating these apps on my Github site
- Show several live Shiny apps
- Suggest Shiny baseball apps that you can do on your own

Communicating with Data

- Want to gain an understanding “how sports organizations analyze, interpret, and communicate in sport”
- In my experience, students can have difficulties in communication of results
- One way to communicate is to develop a web dashboard that illustrates your work.

What is Shiny?

- R package that facilitates the building of interactive web apps from R
- Can host standalone apps on a webpage
- Can enhance the Shiny app with html and JavaScript script

Process of Building a Shiny App

- Get an idea for what you want to produce
- Write a R function with specific inputs and outputs
- Convert this to a Shiny app
- Easy to borrow code from other Shiny apps that are posted

Getting Started with Shiny

- Load the shiny package
- Write a R script `app.r`
- One part of script describes the layout (inputs and outs)
- Second part of script reads the inputs and executes the output
- Third part of script makes the Shiny app

Script app.R

- Function `ui()` describes the user interface (layout of inputs and outputs)
- Function `server()` does the work
- Function `shinyApp()` converts code to a Shiny app

app.R - Basic Shiny Template

```
library(shiny)
```

```
# load in any packages
```

```
# read in data
```

```
ui <- fluidPage(  
  column(4, wellPanel(  
    # space for your inputs  
  )),  
  column(8,  
    # space for your outputs  
  )  
)
```

User
Interface

```
server <- function(input, output, session) {  
  # produces the output  
}
```

Server

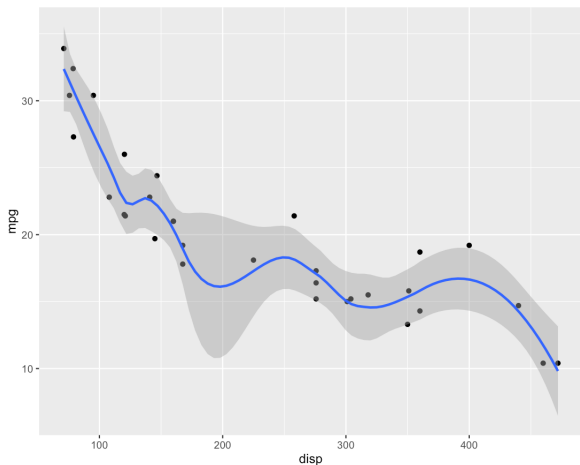
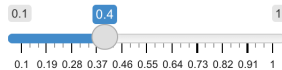
```
shinyApp(ui = ui, server = server)
```


Basic Example: Smoothing a Scatterplot

- Interested in showing a scatterplot and the smoothing curve
- Smoothness of curve depends on argument span
- Want to show how smooth depends on span

Shiny Smoothing Scatterplot

Smoothing parameter:

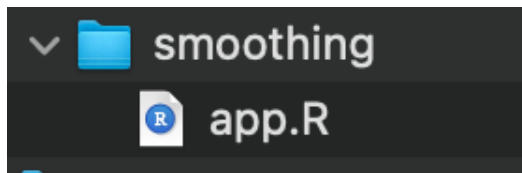


Shiny Smoothing Code

```
library(shiny)
library(ggplot2)
ui <- fluidPage(
  column(4,
    sliderInput("span", "Smoothing parameter:",
               min = 0.1, max = 1, value = 0.4)
  ),
  column(8,
    plotOutput("my_smooth")
  )
)
server <- function(input, output, session) {
  output$my_smooth <- renderPlot(
    ggplot(mtcars, aes(dis, mpg)) +
      geom_point() +
      geom_smooth(method = "loess",
                  formula = "y ~ x",
                  span = input$span)
  )
}
```

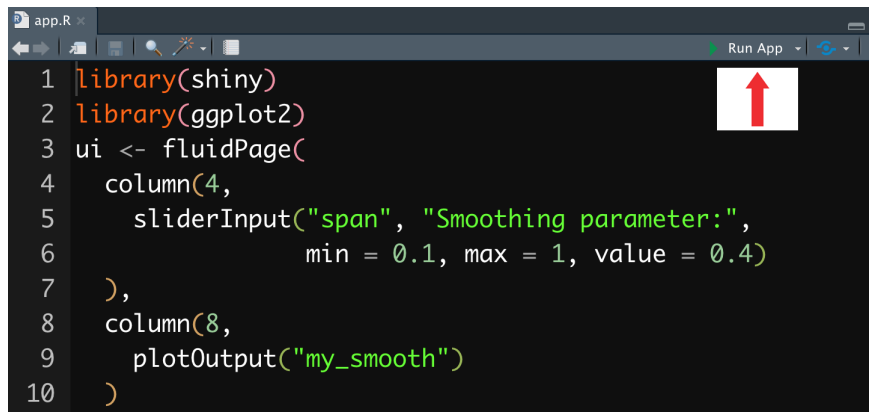
To Run Shiny App 1

Put app.R file in separate folder



To Run Shiny App 2

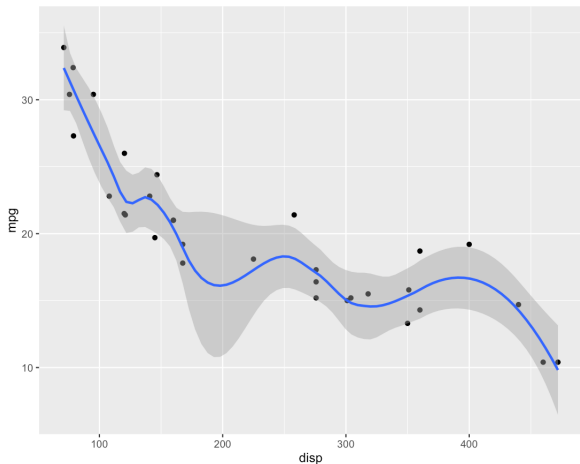
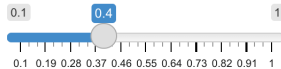
Open file in RStudio and push "Run App" button



```
1 library(shiny)
2 library(ggplot2)
3 ui <- fluidPage(
4   column(4,
5     sliderInput("span", "Smoothing parameter:",
6                 min = 0.1, max = 1, value = 0.4)
7   ),
8   column(8,
9     plotOutput("my_smooth")
10  )
11 )
```

Shiny Output

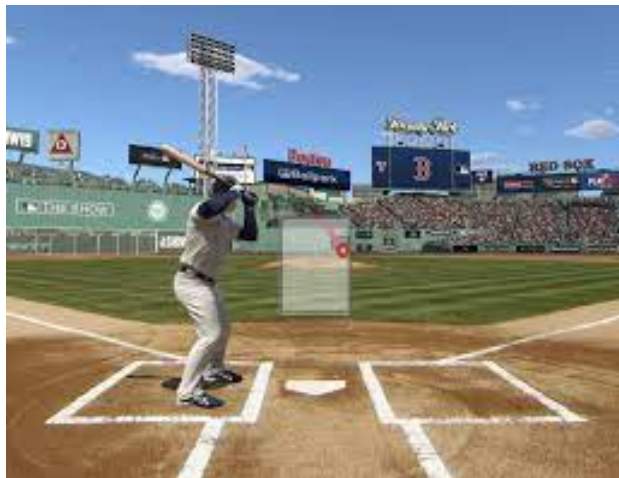
Smoothing parameter:



Two Baseball Applications

- 1 Pitch locations by pitch type and count
- 2 Brushing batting averages over the zone

Watching a MLB Game on Your Device: Pitch Locations



Shiny Pitch Location App

- How does pitch location vary by pitch type and count?
- Want to display pitch locations over zone
- Inputs:
 - specific pitcher
 - specific pitch types
 - specific counts

Shiny Pitch Location App

- Have 2019 Statcast data stored on my Github site
- Information on 732,473 pitches
- Relevant variables
 - pitcher id
 - pitch location (pitch_x, pitch_z)
 - current balls and strikes
 - pitch type

Focus on Aaron Nola (Phillies' Ace)

- Two of his primary pitches are the four-seam fastball and the knuckle curve
- Explore pitch locations of these pitches over 0-2, 1-1, 2-0 counts
- Believe that location depends whether it is pitcher's count or a hitter's count

Write a Function to Perform Task

`construct_graph()`

- input data, pitcher id, pitch types, count
- outputs a ggplot2 display with facets for pitch type and count
- want to turn this function into a Shiny app

Shiny Input Functions

- Use `textInput()` to enter in the pitcher id
- Use `checkboxGroupInput()` to enter in one or more pitch types
- Use `checkboxGroupInput()` to enter in one or more counts

Shiny Output Functions

- `plotOutput()` in user interface is paired with `renderPlot()` in server function
- `tableOutput()` in user interface is paired with `renderTable()` in server function

Shiny Pitch Location App - Aaron Nola

Pitch Locations by Pitch Type & Count

Pitcher MLBAM Id:

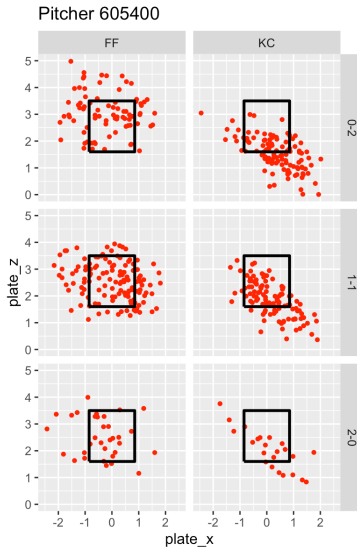
605400

Pitch Type:

- ☐ CH ☐ CU ☐ EP
☐ FC ☒ FF ☐ FO
☐ FS ☐ FT ☒ KC
☐ KN ☐ SI ☐ SL

Count:

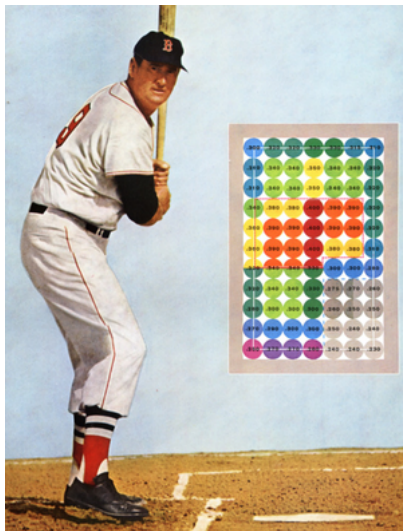
- ☐ 0-0 ☐ 1-0 ☐ 0-1
☒ 2-0 ☒ 1-1 ☒ 0-2
☐ 3-0 ☐ 2-1 ☐ 1-2
☐ 3-1 ☐ 2-2 ☐ 3-2



Inspiration for Brushing BA App

- Famous book: “The Science of Hitting” by John Underwood and Ted Williams
- Includes illustration showing Williams’ batting average over different regions about the zone
- Want to use Shiny to do something similar

Ted Williams AVG by Location



Why is this Useful?

- Every batter likes to see pitches in specific locations
- Each batter has areas of the zone where he is "hot" or "cold"
- Teams (especially the opposing pitchers) want to know this information

Shiny Inputs and Outputs

- One input – name of player of interest
- First output– the graph of pitch locations
- Second output – a table showing the BA over the brushed region

To Enable Shiny Brushing

In user interface, add brush argument to `plotOutput()` function:

```
plotOutput("plot",  
           brush = brushOpts("plot_brush",  
                             fill = "#0000ff"),  
           width = '455px'),
```

In server (table output) use `brushedPoints()` function:

```
sc1 <- brushedPoints(filter(sc2019_ip,  
                           player_name == correctinput(input$name)),  
                     input$plot_brush)
```

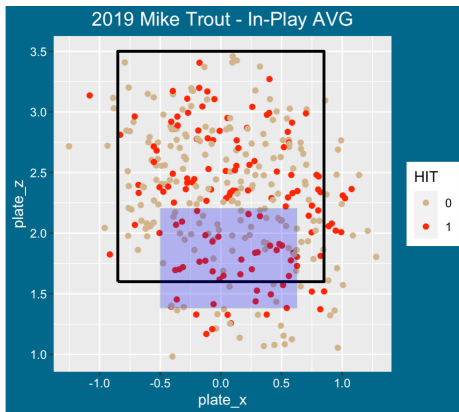
Shiny Brushing App - 2019 Mike Trout

Brushing Batting Average

Enter Batter Name:

Mike Trout

Scatterplot shows pitch locations for balls put into play. Brush over a region of the scatterplot and you will see the in-play batting average over the region.



Brushed Region Stats

Name	BIP	H	BABIP
Mike Trout	77	38	0.494

Demonstrate on RStudio Cloud



- `rstudio.cloud`
- Attractive way to teach data science to students

Sharing Shiny Apps

- Share the Shiny code (the app.R file, data and functions)
- Put the Shiny apps functions in a R package
- Host the Shiny apps on a server such as `Shinyapps.io`

ShinyBaseball Package

- Collect many of my Shiny apps that explore baseball data
- Once installed and loaded, easy to run the apps using functions
- To run the Brushing app, type

`BrushingZone()`

Functions in R Package ShinyBaseball

- BatterFourSeam() - Illustrates Four-Seamer Rates Over Zone for a Batter
- BrushingZone() - Brushing Zone Measurements
- PitchOutcome() - Illustrates Pitch Outcomes
- PredictingBatting Rates() - Illustrates Predicting Batting Rates
- RadialChart() - Radial Chart for a Pitcher
- SprayChart() - Locations of Balls in Play

Put Your Shiny App on the Web

- Can use one's own server or RStudio's hosting service
- Shinyapps.io
- Free and paid options available
- I'll show several of my live Shiny apps

Fancier version of brushing app.

Brushing Zone

Batter Name:

Mike Trout

Measure:

☒ Launch Speed

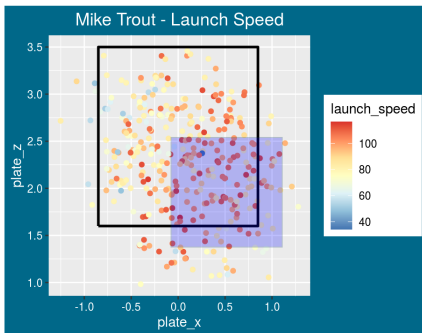
☐ Hit

☐ Home Run

☐ Expected BA

Click for Launch Speed, xBA:

player_name	Launch Speed	xBA
Mike Trout	84.400	0.613



Name	BIP	H	HR	LS	H_Rate	HR_Rate	xBA
Mike Trout	131	63.000	21.000	97.284	0.481	0.160	0.467

Brushed Region Stats

Home Run Story

- MLB has seen a dramatic impact in home run hitting
- Batters have changed approach?
- Does the ball have an impact?
- Weather, ballpark effects?

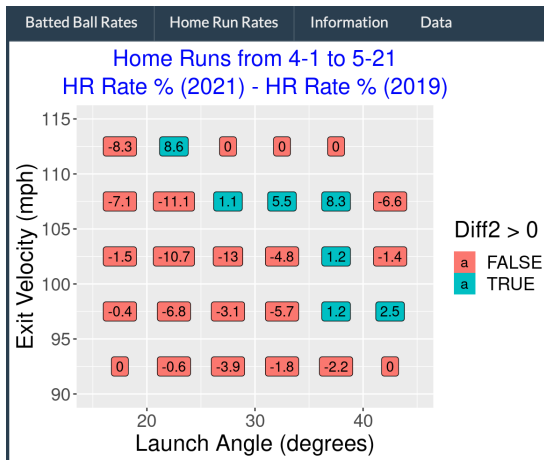
Home Run Story

- To hit a home run, need to hit the ball hard (high exit velocity)
- Also need a suitable launch angle (around 30 degrees)
- Given exit velocity and launch angle measurements, what is the chance of home run
- This HR probability has changed over seasons

Home Run App

- Select a time interval and two seasons to compare
- Categorize balls in play by launch angle and exit velocity
- Compare home run rates in two seasons
- Takeaway: the 2021 ball is different from the 2019 ball

Compares home run rates for two seasons.



How to Learn Shiny?

- Look at the Shiny tutorial material at <https://shiny.rstudio.com/>
- Look at Shiny demos
- Material for this talk can be found at <https://github.com/bayesball/LearnShiny>
- Start with a simple template and add features incrementally

Your Turn

- I have posted several Shiny apps on my Github site
- Try them out
- I will suggest some modifications for you to try

github.com/bayesball/LearnShiny



bayesball Create README.md



brushing

add files



pitch_type_count

add files



smoothing

add files



.DS_Store

add files



README.md

Create README.md

The Pitch Type Count App



main ▾

[LearnShiny](#) / [pitch_type_count](#) / Seminar Version /



bayesball add files

..



.DS_Store

add files



.Rhistory

add files



app.R

add files



construct_plot.R

add files


Dataset Variables


```
pitcher = col_double(),  
batter = col_double(),  
pitch_type = col_character(),  
plate_x = col_double(),  
plate_z = col_double(),  
p_throws = col_character(),  
stand = col_character(),  
type = col_character(),  
events = col_character(),  
description = col_character(),  
estimated_ba_using_speedangle = col_double(),  
Count = col_character()
```

Things to Try


- Focus on a specific pitcher
- Focus on how a pitcher throws to Right and Left handed batters (variable stand)
- Show pitch locations of a specific pitch type against R and L
- Use facets to show pitch locations by pitch type and batter side

The Brushing App

 main ▾ [LearnShiny](#) / [brushing](#) / Seminar Version /

 **bayesball** add files

..

 app.R add

Dataset Variables

```
player_name = col_character(),  
plate_x = col_double(),  
plate_z = col_double(),  
H = col_double(),  
HR = col_double(),  
launch_speed = col_double(),  
launch_angle = col_double(),  
estimated_ba = col_double(),  
hc_x = col_double(),  
hc_y = col_double(),  
stand = col_character(),  
pitcher = col_double(),  
game_pk = col_double(),  
game_date = col_date(format = "")
```

Things to Try

- Focus on a specific hitter
- Instead of brushing hits (variable H) ...
- Display and brush home runs
- Display expected batting average and brush these values

To Get Further Ideas

- Visit the Baseball Savant website (home of Statcast data)
- Look at the various visualizations
- Try to replicate or improve what they show on this site
- I wrote a Shiny app to produce a "Radial Chart"

To Get Further Ideas

- I contribute to the blog “Exploring Baseball Data with R”
<https://baseballwithr.wordpress.com/>
- All of my R code is available on Github
<https://github.com/bayesball>

Questions?



Ask me here or email me at
albert@bgsu.edu