The Bright Future of Shiny Apps in R

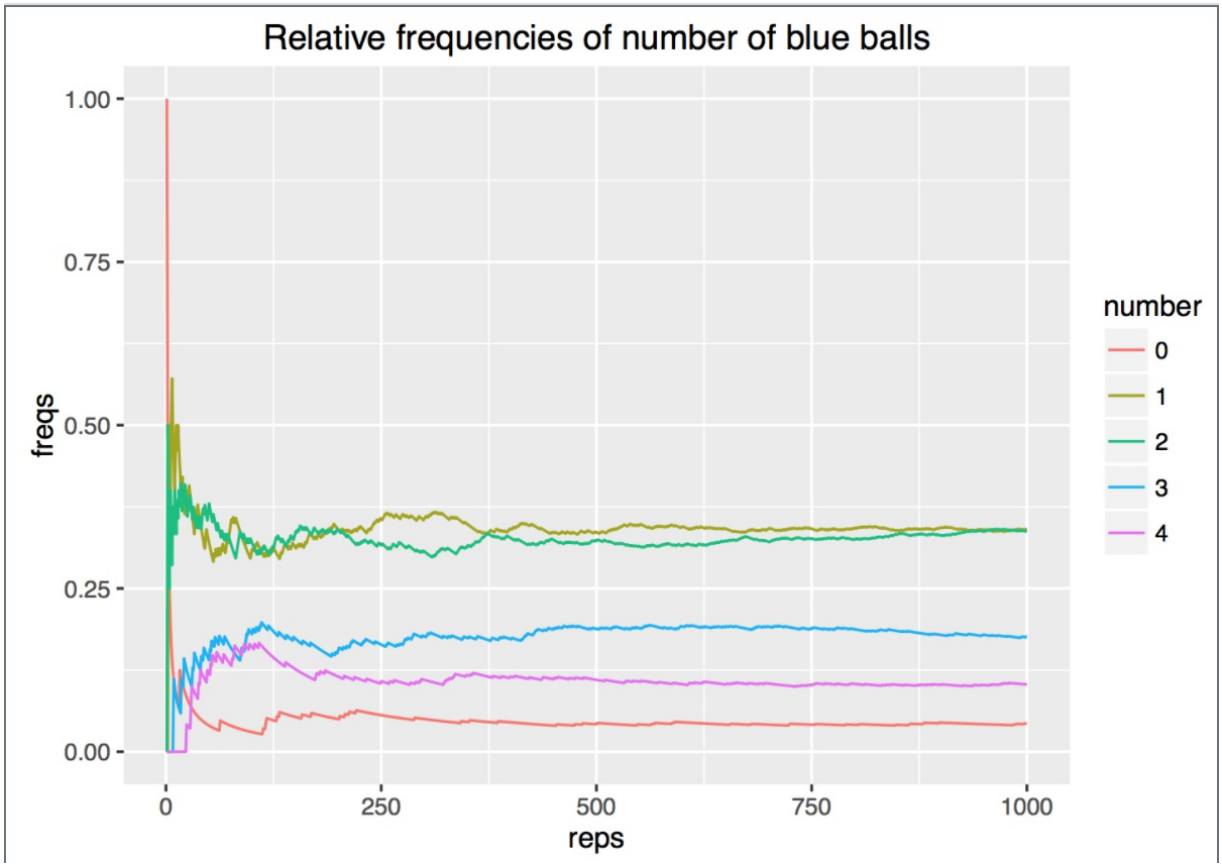
Shiny Logo

Why am I posting about this?

Throughout this course, we've spent a lot of time learning about many different practical tools used in R that can help us analyze data more efficiently and shed light on seemingly meaningless columns and rows of numbers. Given that so much time has been spent in the labs and the homework assignments reviewing and applying concepts to straightforward exercises with NBA data, it was especially refreshing to spend some time learning how to make actual web apps using the Shiny package in R, which will be the main focus of this post. Making presentable and accessible apps in today's generation is so important, and we as students all know this based on how much time we spend on average using the Internet. So as a result, having had the opportunity to learn about the package and figuring out how to apply it to different fields were my main motivators.

So what exactly is Shiny and how do I use it?

Shiny can be defined for us as a tool to turn analytical code chunks and algorithms into a usable web application without needing web-based languages like HTML and CSS. As seen in our ninth lab, we can also add widgets to the app we want to create and make it more elaborate. Although we haven't explored too many examples in class, Shiny apps can be created for many different fields such as economic analysis, location tracking, and A/B testing for marketing purposes. The graph you're seeing below was produced from a Shiny script that calculated the relative frequencies of number of blue balls from a bag in different numbers of trials based on the number of balls in the bag.



picture

So while general R scripts like the one shown below can be used for specific problems, the Shiny app interface allows you to conveniently keep running scripts in a clean and easy-to-use format.

```
## Basic Rankings of Total Salary, Total Points, and Total Efficiency (in order)
teams <- read.csv("../hw03/data/nba2017-teams.csv", stringsAsFactors = FALSE)
library(ggplot2)
```

```
## Warning: package 'ggplot2' was built under R version 3.3.2
```

```
library(dplyr)
```

```
## Warning: package 'dplyr' was built under R version 3.3.2
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

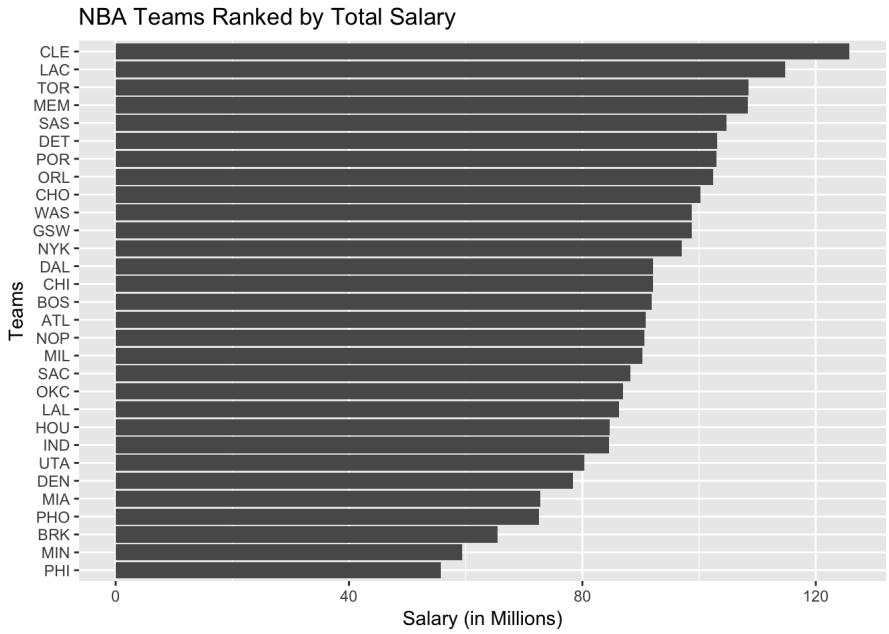
```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
teams <- teams %>% arrange(desc(salary))
```

```
## Warning: package 'bindrcpp' was built under R version 3.3.2
```

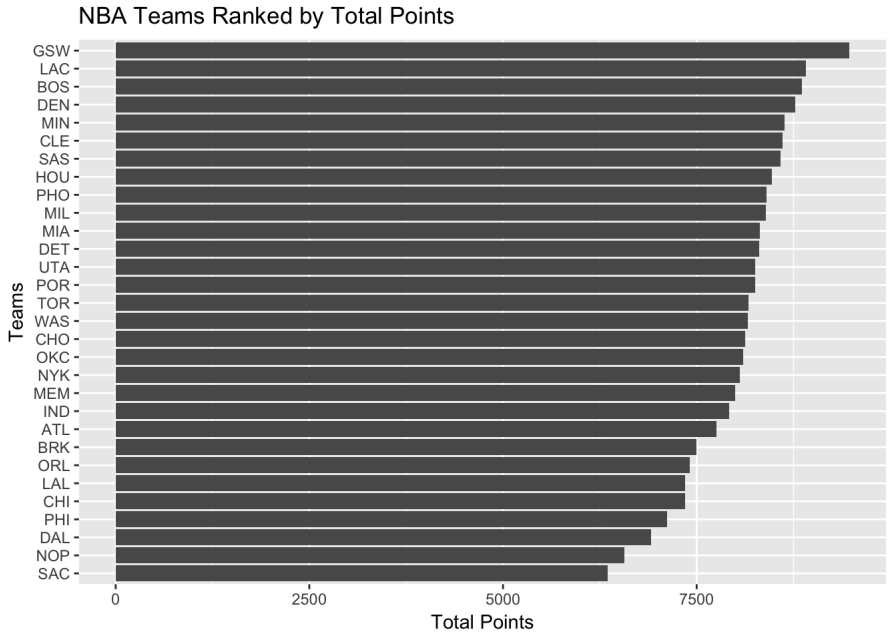
```
ggplot(data = teams, aes(x = reorder(team, c(30:1)), y = (salary/1000000))) + geom_line(y = mean(teams$salary/1000000), color = "red") + geom_bar(stat = "identity") + coord_flip() + labs(x = "Teams", y = "Salary (in Millions)", title = "NBA Teams Ranked by Total Salary")
```

```
## geom_path: Each group consists of only one observation. Do you need to
## adjust the group aesthetic?
```



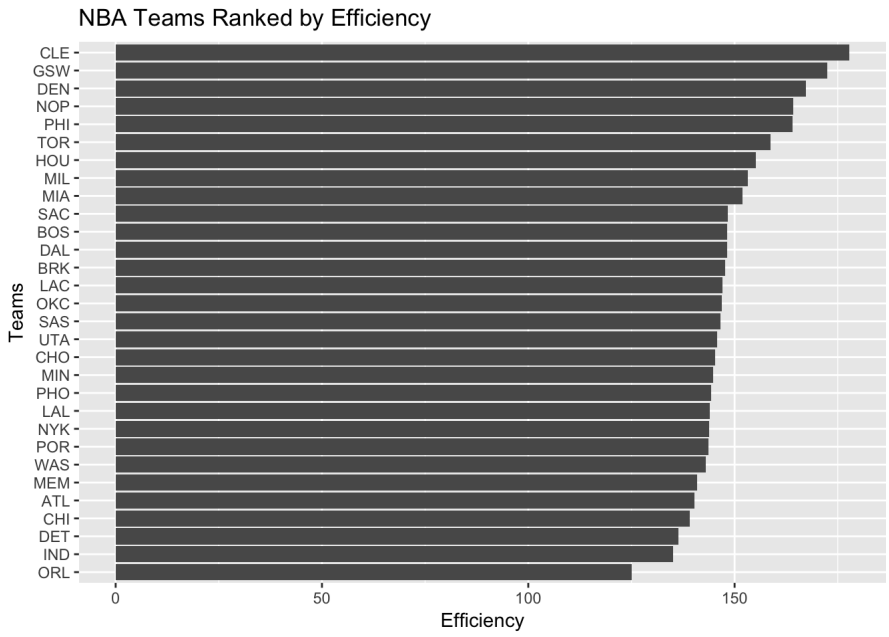

```
## Cleveland leads the total salary rankings with teams such as the Clippers and Raptors right behind them. Golden State ranks fairly low on the chart.

teams <- teams %>% arrange(desc(points))
ggplot(data = teams, aes(x = reorder(team, c(30:1)), y = points)) + geom_bar(stat = "identity") + coord_flip() + labs(x = "Teams", y = "Total Points", title = "NBA Teams Ranked by Total Points")
```




```
## Golden State leads the total points count and is followed by the Clippers and Cleveland in 5th place.

teams <- teams%>% arrange(desc(efficiency))
ggplot(data = teams, aes(x = reorder(team, c(30:1)), y = efficiency)) + geom_bar(stat = "identity") + coord_flip()
+ labs(x = "Teams", y = "Efficiency", title = "NBA Teams Ranked by Efficiency")
```

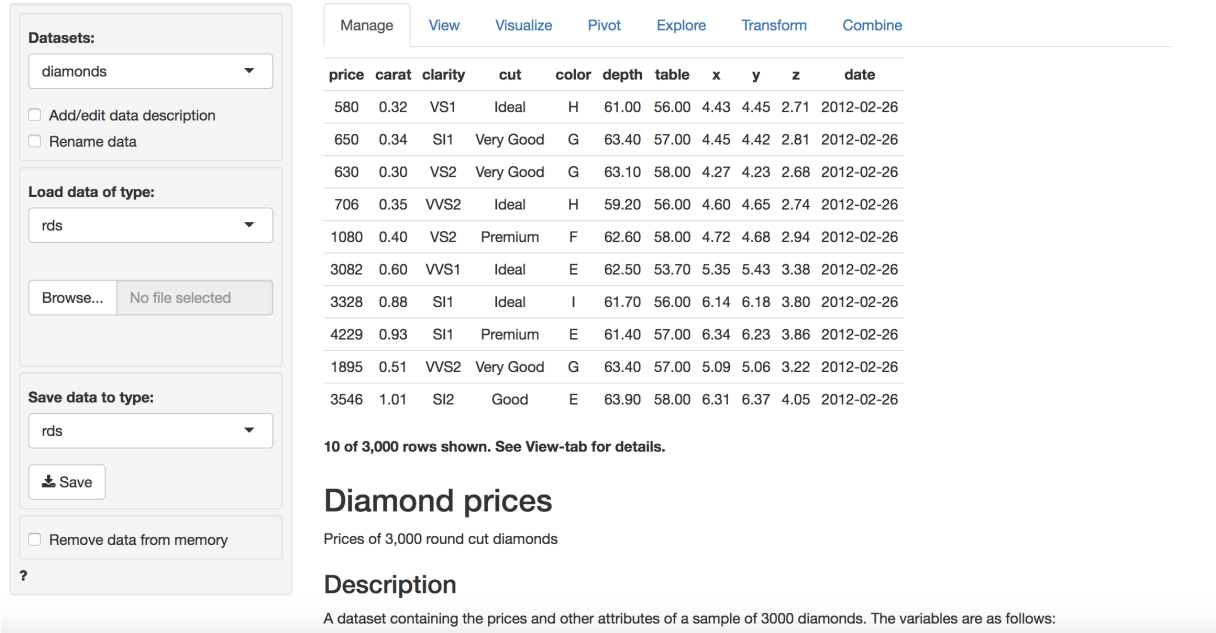



```
## Cleveland and Golden State lead the efficiency rankings and both teams lead one of the other rankings charts es
tablished so far.
```

Shiny apps are generally composed of a user interface function, a server function, a ggplot function for graphical displays, and a call to the shiny library to be able to create the interface. This makes assembling the app much easier and can be compared to fitting pieces of a puzzle together.

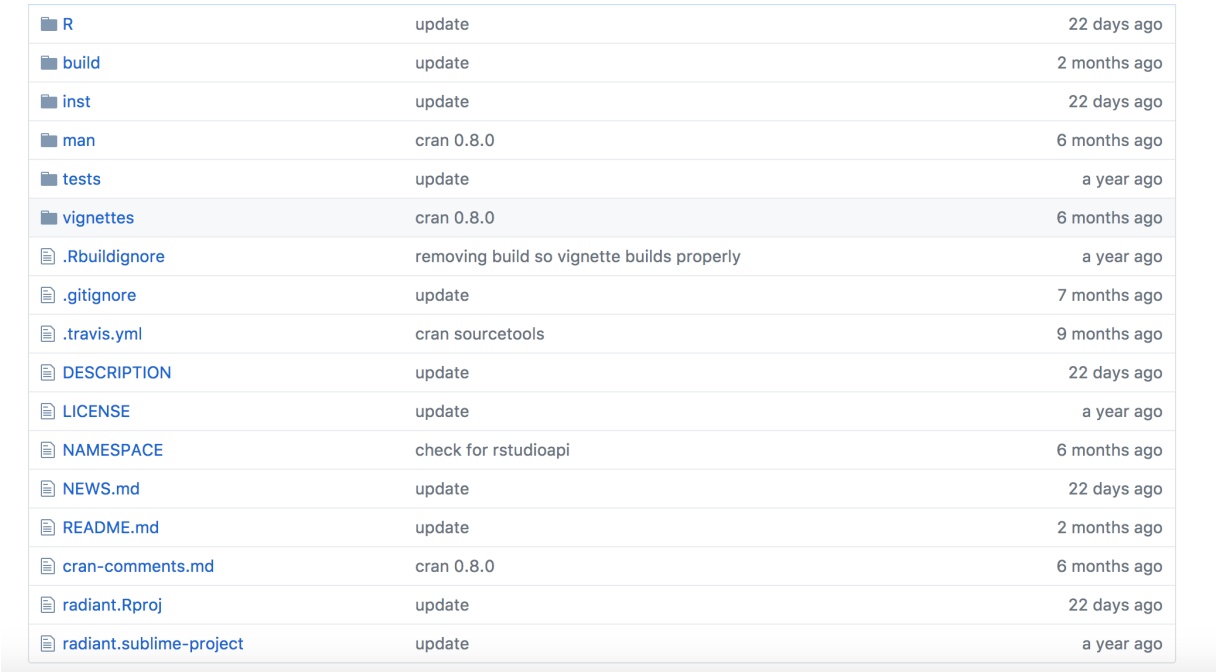
Radiant Example

Since we've gone over how apps are assembled and how customizable they can be, let's look at a specific application related specifically to business analytics called Radiant. Radiant takes in a specific data set and maps out the relationships between mutliple variables and displays important variable values related to the dataset.



radiant

As you can see at the top of the image, this app has many different functions and categorizes them through different R files in the build folder. This allows for clear organization and accesibility based on your specific purpose for using the app.



file structure

This is particularly useful because now we could take ANY given data set with x amount of variables, plug it into the application, and get all of these different summaries using the functions available to us through the app.

What Else Could Shiny Be Used For?

Given that Shiny allows users to get past the difficulties of implementing HTML/CSS web dev scripts to their R code which was surely already difficult to write, this is a good way for non-tech-savvy people who are interested in creating better-looking web apps that still contain a lot of functionality. As I mentioned previously, some other applications could be to provide more clarity to economic supply-and-demand models and comparing machine learning algorithms.

MULTIDIMENSIONAL SCALING

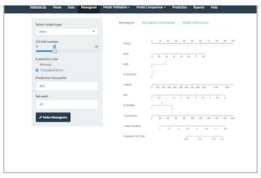
Similarity analysis tool.

INTERACTIVE REPORTING

Interactive report with Shiny and R Markdown

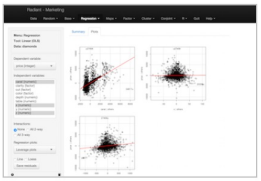
GEOMETRY OF CLASSIFIERS

Comparison of machine learning algorithms.



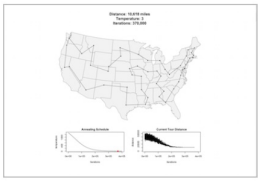
NOMOGRAM GENERATOR

Tune analysis then export as pdf, HTML, or Word file.



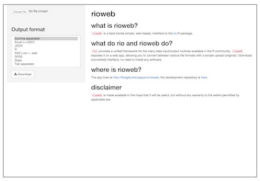
RADIANT

Extensive app for teaching business analytics. (documentation) (code)



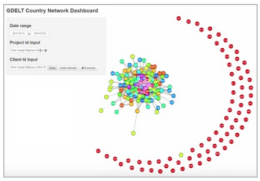
TRAVELING SALESMAN

Optimization fun.



FILE CONVERTER

Upload a data file, then download in various formats.



GDELT & BIGQUERY

Dashboard to GDELT database with quarter-billion records

Different Apps

Because Shiny makes our jobs as users who want to create the app easier, it could allow a much wider audience to start learning about higher-level programming and become more educated about programming.

So What?

Given that so many people in the US today are still nowhere close to proficient with technology and unaware of how relevant this knowledge is to their futures, it is important to find creative ways to increase accessibility. Since R isn't the most difficult language to learn, the Shiny package helps to solve this problem by providing an easy way for the average user to put together a fully functional and useful application with different widgets and designs.

References

- Shiny RStudio Tutorial: https://shiny.rstudio.com/tutorial/
- About the Radiant Business Analytics Tool: https://vnijs.shinyapps.io/radiant/?SSUID=6d5900265a
- Data Visualization in R: http://opensourcesoftware.casact.org/draftblogs:6
- Relative Frequencies Graph from Lab: https://github.com/ucb-stat133/stat133-fall-2017/blob/master/labs/lab09-random-simulations/lab09-random-simulation.md
- Why Use Shiny?: https://www.lynda.com/RStudio-tutorials/Why-use-Shiny/452087/490039-4.html
- Intro to Shiny Apps: https://www.youtube.com/watch?v=Ido56dwDTg8
- Radiant Source Code: https://github.com/radiant-rstats/radiant/blob/master/R/radiant.R