

SUPER DATA ANALYSIS

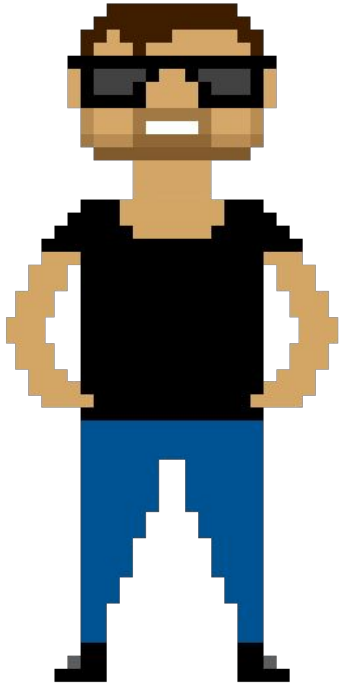
WITH R SHINY

1 Player Game

2 Player Game



Patrick S. Medina



- From: Los Angeles
- Status: Ph.D. Candidate
- Year: Fourth
- Research: Topological Data Analysis
- Member: RW Doerge Research Group
- Contact: patrick.medina1@gmail.com

Notable Shiny Projects

- Genentech: Phase 2 drug trial planning tool
- Weake's Lab: Studying temporal gene expression.

What is Shiny?

How does it work?

Components of a Shiny App

- Getting Started: `library("shiny")`
- User Interface: The user-interface (`ui.R`) script controls the appearance of the app.
- Server: The server (`server.R`) script contains the R functions that runs the app.

User Interface

- General Format:

```
> shinyUI(fluidPage(  
  > ))
```

User Interface

- The interface is built off of Bootstrap's 12-column fluid grid system.

Server Side

- General Format:

```
> shinyServer(function(input, output) {  
  > })
```
- **input** variables are reactive variables received from the user-interface.
- **output** variables are reactive variables that are the results of the different R functions. They may be in the form of text, plots, lists, arrays, tables, etc.
- Different R packages may be included in the server.R file and used in computation.

Reactive Programming

- Simple Anecdote: $a = b + c$
 - Static programming: Downstream changes to **b** or **c** have no effect on **a**.
 - Dynamic programming: Downstream changes to **b** or **c** immediately change **a**.
- The **input** variables from, and the **output** variables to, ui.R are reactive variables.
- When computation is intensive, or redundant, intermediate reactive variables called **reactive expressions**, may be used.
- Calling reactive variables outside of a reactive context will result in an error.

Distribution

- Standalone apps launched from R Studio.
- HTML Apps on the Web
 - Free hosting on Shinyapps.io (Up to 5 active apps for free)
 - Setup a Shiny Server
- Embed in R Markdown Documents
- R Studio Addins (NEW!)

Extending Shiny

Dynamic UI

- The User Interface is dynamic and can change based off of user preferences or results.

HTML Customization

- The user interface may be designed in HTML.
- Style the app using CSS (Cascading Style Sheets).
- JavaScript events.

Extending Shiny

Useful Javascript Widgets

- Interactive Tables with [DataTables](#)
- GeoSpatial Mapping with [Leaflet](#)
- Interactive Scatterplots and line charts with [Metricgraphics](#)
- Interactive network graphs with [networkD3](#)

Advanced Customization

- Create your own widgets with [htmlwidgets for R](#)

Useful Links

Shiny Tutorials

- <http://shiny.rstudio.com/>

Shiny Examples

- [R Studio Examples](#)
- [Shiny User Gallery](#)

Shiny Distribution

- [Shiny Server](#)
- [Shinyapps.io](#)