**Using Shiny and Leaflet to Draw Maps**

**Installing Shiny Server**

Currently, the Shiny server runs on an installation of Ubuntu Server 16.04.2 LTS. When the server is live, any Shiny app running on the server can be accessed by pointing the URL to the root folder of the Shiny project.

* Installation: <https://www.rstudio.com/products/shiny/download-server/>
* Documentation: <http://docs.rstudio.com/shiny-server/>

**Basic structure**

A Shiny app is composed of two components: the server and the user interface, stored in server.R and ui.R respectively.

Their roles function just as any frontend/backend setup would. The server opens the connection with the user, listens for input, processes the data, and renders the graphics. The user interface defines the layout of the visible components of the app.

* Shiny documentation: <https://shiny.rstudio.com/reference/shiny/latest/>

**The user interface (ui.R)**

Shiny wraps all the UI inside a fluidPage function, so that elements are responsive to resizing the window. The functions that Shiny provides for visual layout are intuitive, and allow different elements such as sidebars, radio buttons, and file upload buttons to be implemented. Shiny also provides equivalent functions for HTML5 tags.

* The UI is based on Twitter Bootstrap. Further customization can be made with additional CSS:  
  <https://shiny.rstudio.com/articles/css.html>
* For more on UI:

<https://shiny.rstudio.com/tutorial/lesson2/>

**The server (server.R)**

Everything else is done inside the server.

*Shiny Input/Output Model*

From “How does Shiny work?” by Ian Pylvainen (emphasis added):

At its base level, Shiny is an R package that brings R to the web.

Shiny is based on a **reactive** programming model, similar to a spreadsheet. Spreadsheet cells can contain literal values, or formulas that are evaluated based on other cells. Whenever the value of the other cells change, the value of the formula is automatically updated.

(<https://support.rstudio.com/hc/en-us/articles/218294767-How-does-Shiny-work-)>

Users begin a **session** when they connect to the Shiny app, and end when they disconnect. During that time, Shiny continuously listens for input, and updates its output accordingly.

*Opening the Shapefile*

Use the readOGR() function from the rgdal library for converting from Shapefiles to SpatialPolygonDataFrames, as readShapePoly() is deprecated.

* How to open a shapefile in R:   
  <https://gis.stackexchange.com/questions/19064/how-to-open-a-shapefile-in-r>

**Leaflet**

Maps are best drawn on the web using Javascript. Out of the many Javascript libraries that do this (D3, MapBox, OpenLayers, etc.), Leaflet seemed to have the most well-supported API for R. Plot.ly was also a consideration, but Leaflet allows for finer grain control and room for customization.

Leaflet includes most basic interactive web map features, such as zooming, panning, hovering, and displaying tooltips.

* Leaflet homepage: <http://leafletjs.com/>
* Leaflet for R: <https://rstudio.github.io/leaflet/>

**Resources**

* Choropleths with Leaflet and R: <https://rstudio.github.io/leaflet/choropleths.html>
* Highlight Polygon on click/hover: <https://www.r-bloggers.com/r-shiny-leaflet-using-observers/>