BMI 706 Homework 3

Learning Goals

- Learn how to link interactive plots using Plotly and the `shiny` R library

For the problem in this dataset, you will use R to implement interactive Plotly visualizations that support basic brushing and linking and can be manipulated using external controls such as checkboxes, sliders, and dropdown lists.

Please obtain the following from the Resources section on Piazza:

1. The data set to be used for this problem is the same as for Homework 1:
   Assignment 1 Data - flunet2010_11countries.csv
2. Code to wrangle the data as well as code for a partial solution of Problem 1.1:
   Assignment 3 Sample Code
3. A demo and sample solution that illustrates the interactive plots:
   Assignment 3 Demo - Interactive Visualizations
   (https://gehlenborglab.shinyapps.io/HW_3_demo/)

Please review the demo before starting to work on these problems. The questions below are referring to the demo. Please also review the documentation on R shiny apps here: https://shiny.rstudio.com/tutorial/. There's a lot of information there but unifying pattern is that an R shiny app consists of a server and a UI and both have to be instantiated separately. The UI collects parameters from the user and passes them to the server. The server, in turn, takes those parameters and generates a plot.

Problem 1 (15 points)

In the first problem, we will examine how to create a simple set of controls that can be used to determine which subset of the data is plotted. In the last part of the problem, we will examine how to create interaction between two different plots without the use of the Shiny controls.

The questions below ask you to create the visualizations shown in the demo. Your solutions can either implement them exactly as shown or contain variations that meet the requirements set forth in the questions. This will consists of three tasks:

1.1 (5 points)

Implement a dropdown list to support selection of the country whose data is displayed in the plot. This dropdown should list the countries available in the data and then show a line plot of flu cases per week for the selected one.

For this section, look into using the *selectInput* function in Shiny.

1.2 (5 points)

Implement a slider that lets you select the range of weeks to display in the output plot. The slider should go from the minimum number of weeks (1) to the maximum (52). Sliding either end should change the range of the *x* axis in the plot.

Consider using the *sliderInput* function to accomplish this task.

1.3 (5 points)

If you have completed section 1.2, you will notice that the slider-plot communication is one-directional. It changes the range of the plot, but when we zoom into a range in the plot, the slider doesn't change. For this problem, we'd like you to try to implement a bidirectional control which listens to events in the plot and adjusts the slider in the UI.

One way to accomplish this is to *listen* to the events emitted by the Plotly component and use the *updateSliderInput* function to change the value of the slider. The event that plotly emits when changing the visible region is *plotly_relayout*.

Problem 2 (25 points)

In this problem will look into how to link two plots and combine them with a controller that modifies both. For this example, we will create a correlation matrix and then use that to create a correlation network where highly correlated nodes (countries) are linked.

2.1 (10 points)

Selecting which countries to link in our node-link diagram requires filtering for pairs of countries which have a correlation coefficient greater than some threshold. There is no concrete heuristic for deciding what that coefficient should be. This is a perfect example for using visualization to explore a range of values and see how changing them affects the output.

For the first problem, we would like you to add a controller which lets you select this threshold and updates the heatmap and node link diagrams directly.

2.2 (15 points)

One of the challenges of displaying linked plots is providing necessary for viewers to see the linkage. In our example, we have a heatmap and a node-link diagram. We would like to show how the cells in the heatmap are related to the links in below. For that we will implement an

interaction that highlights links in the node-link diagram when somebody clicks on a cell in the heatmap.

What to submit?

1. An R Markdown file with all source code that you used to create the plots.
2. A knitted HTML file including the interactive plots.
3. *Optional:* Short video or GIF that shows the interactive features of your plots.

Administrative

When is it due?

- Thursday, April 20th, 11:59 pm

How do I hand in my solutions?

- Email required materials to bmi706.2018@gmail.com.

What if I have questions?

- Please use Piazza (https://piazza.com/harvard/spring2018/bmi706/home) to ask questions about this assignment. Please avoid sending messages for clarification directly to the instructors.