HW 8 Assignment Write Up

Daniel Le Compte

Website (Github Pages): https://lecomptedaniel.github.io/DecisiveAction/
Github Repo: https://github.com/lecomptedaniel/DecisiveAction

Part 1

This is a storytelling and interactive data visualization website that makes the point and argument that early and rapid responses by governments to COVID saves lives. I show data that is updated weekly from a COVID database in the EU, as well as other major sources, such as the US CDC and John Hopkins University. This data will include key COVID data, such as cases, testing, hospitalization, deaths, and recoveries, as well as governmental actions taken to mitigate and contain COVID spread (usually contained within or adjacent to the COVID numbers themselves).

Most COVID dashboards and websites accurately detail the number of cases per day, the spikes, the dips, the changes, and even the biases in how COVID has a far greater impact on BIPOC and the poor. Unfortunately, not many COVID dashboards or sites make the case or show the relationship between government actions and the results within COVID cases and deaths. This website shows this relationship clearly and explains why it is relevant through data visualizations. This website is designed to be a conversation starter among regular adults, more likely aimed towards young adults who are open to reading and learning about other perspectives.

Part 2

- Animated data visualization - a user clicks on a play button and a predefined animation of a data viz will play.
- Full interactive data visualization - a user, on the last page, can fully interact with the data visualization through Plot.ly's interactive menu. This means that users can select a piece of the data to zoom in on, hover over data instances to view more information with labels, among other interactive features of the Plot.ly plot.

Part 3

Tool #1:

1. Plot.ly (https://plotly.com/)

2. A smooth, quick, and interactive way to visualize data, export to a file, and host within a website through simple code. Integrates well with other data wrangling and visualization languages Python and R.
3. Using Plot.ly to convert my ggplot plots (see tool below) into HTML Plot.ly plots, that are then transferred into <iframe> elements for easy integration into my code.
4. Plot.ly transforms my standard ggplot plots and visualizations into interactive plots that add another layer or dimension to the visualizations. In addition, the process for moving visualizations to be hosted on within an HTML site for Plot.ly is very simple compared to ggplot.

Tool #2:

1. R and several R packages (ggplot2, tidyverse, lubridate, htmlwidgets, widgetframe, utils)
2. Using R and R packages to wrangle and visualize the data. ggplot2 affords the ability to create beautiful visualizations easily, and R, in general, allows for great flexibility in scraping data and working with large datasets.
3. I used R and the above R packages to create a script that, when run, imports data from the various sources for COVID data, wrangles them into a usable format, visualizes, and then using Plot.ly, exports to an HTML file and gives me <i frame> code for importing into my HTML files.
4. This adds substantial data visualization capabilities to my site, far more flexible than other formats such as Excel, and produces a greater end-result that is easier to understand and read than other formats as well.

Part 4

The largest change I made to my website from my HW7 mockups was to switch to a desktop focus with mobile and other sizes being prioritized lower. By and large, data visualizations are typically created for desktop consumption, where there is room to show further details and labels. That said, my website is responsive and works across multiple devices.

Part 5

The largest challenge I encountered is in R code and getting the R script to work correctly. I have run into many errors that I am still figuring out, and most likely will work one even after submission.