Linda Xia
05-430 PUI
Assignment 8 Writeup: **Wine to the Nines**

Project link: https://lindatxia.github.io/WineToTheNines/

Part 1: Introduction

Wine to the Nines is an interactive website that educates users on the nine most popular types of wines in the world using D3.js data visualizations. This dataset is a real dataset pulled from a Kaggle machine learning dataset from WineEnthusiast, which I cleaned, filtered and parsed using Python libraries. This website is targeted for the amateur wine enthusiast who has an interest in learning more about different kinds of wines, and would like a recommended wine. It has a responsive data visualization that educates users on which types of wines are most popular in which countries around the world; users are also able to see which specific wine varieties are most popular in different countries. This website is engaging because it helps users learn more about the origins of a popular beverage with a data visualization, breaking down complex statistics in an easy-to-view manner. Additionally, it is difficult to find professional ratings for wines online, and this website helps the casual browser find a wine and learn more about its ratings, price, year, and even wine critic reviews.

Part 2: Interactions

- **Full page scrolling that is responsive to screen size**.
    - Either 1) Click on the bubble nav bar on the left, or 2) Scroll down with your trackpad or mouse.
- **D3.js data visualization of wine count by country and type.**
    - Hover over countries to see the tooltip information about how many wines are produced by that country
        - The tooltip appears in a position close to the country
    - Click the stepper tabs to see the country wine counts for a specific wine variety.
        - Darker colors mean a larger amount of wines are produced by this country.
    - Resize the window screen to see the data visualization scale appropriately.
- **Random wine generator that randomly pulls a wine from a large CSV file.**
    - Click on the “Generate” button.
        - There is an animation on first click with the slideDown() function to bring out the box.

Part 3: External Javascript Libraries

1. **D3.js**
    a. **Why**: I wanted to challenge myself and learn how to create responsive, interactive data visualizations, especially with an extremely large dataset. I was inspired by the visualizations that I’ve seen on New York Times.
    b. **How**: I used D3.js to create a responsive visualization of a world map that enumerates how many wines are produced by each country. Using a JSON file of

countries and their geographic coordinates, I was able to create paths mapping to individual countries in the world, and use different data structures to also read in TSV files that showed how many wines were produced by each country. I also modified code from a tooltip D3.js tutorial I found online (cited in source code) to project a tooltip over the map to give specific numeric wine counts for country.

   c. **What it adds**: It allows for visitors of my site to dynamically see where all of the wines in the world come from. Different wines are popularly produced in different countries, and some countries don't produce wine at all, which is really interesting to me. Users are able to learn a large amount of complex information in a short amount of time by looking at this data visualization.

2. **PapaParse.js**
   - **a.** **Why**: I had a large CSV file that needed to be parsed efficiently, especially with different data types (Integers, Strings), since the homegrown ones I had tried to write myself were unable to do so.
   - **b.** **How**: After loading the CSV file through a JQuery ajax call, I took the CSV string that was produced and parsed them using the PapaParse JS library. This outputted an Array of CSV rows. I used a random number generator to give me a random index, so I could find a random row and parse each individual column data into an attribute of a Wine object.
   - c. **What it adds**: This library helps my website process an enormous amount of wine data, and gives the user over 50,000 wines to choose from. Initially, I thought I would have to slim down my data to just several hundred wines with a different file format, but this data parser efficiently reads in CSV files with easy error handling.
3. **[Not Javascript] Numpy and Pandas Python libraries**
   - **a.** **Why**: The original data file had a large amount of missing fields and encoding errors. Using pre-processing scripting tools like Numpy and Pandas helped me efficiently process my data, remove corrupt data, and correct encoding inaccuracies.
   - **b.** **How**: I ran several commands from these Python libraries to clean my data. I eliminated missing rows to avoid having missing data, and also performed feature extraction by parsing through all of the wine titles and pulling out the year the wine was manufactured.
   - **c.** **What it adds**: This is more of a backend process, but cleaning the data ensured that I had no strange encoding errors (for example, the original file had errors like: LÃƒÂ©tÃƒÂ©-Vautrain NV Brut Royal RosÃƒÂ©). Without fixing these encoding errors, presenting this data on my website would have been illegible, decreasing the overall user experience.
4. **FullPage.js**
   - **a.** **Why**: I wanted to have a full-screen scrolling experience, as I became inspired by the scrolling examples that Jason mentioned in lecture. This Javascript library allowed me to create beautiful, scrollable full screens easily.
   - **b.** **How**: I added a call in my main.js file to set up the scrolling, left navigation bar, and speed of the full-page scrolling. Additionally, I modified some of the CSS attributes

of the navigation bar (to hide it) when it reaches a certain small screen size to prevent overlapping div elements.

   c. **What it adds**: This allows users to look at each “chunk” of information independently from one another. It’s similar to having separate html pages for each learning topic, but instead of clicking and reloading, FullPage.js helped me create a seamless, single-page viewing experience for my users.

5. **Bootstrap.js**
   a. **Why**: I used Bootstrap primarily because it was a responsive framework, and I was able to easily design a custom 5-column grid that scales to different sized screens.
   b. **How**: To create a custom 5-column grid, I added @media queries to my css file to create different column percentages for each screen size.
   c. **What it adds**: It looks nice on differently sized screens! Also, as I placed the SVG of my D3.js world map inside a Bootstrap column container, this helped my world map data visualization scale as well.

Part 4: **Iterations from Assignment 7**

My original plan for Assignment 7 was to create a website that taught users more about the specific personality traits of different Myers-Brigg personality types. I decided to shift to a completely different topic after finding an interesting dataset on Kaggle, and I checked with my TA, Kristin, after I made this change. My learning goal was to learn a new library on my own for this assignment, and my high-fi prototypes for the MBTI idea involved lots of vanilla HTML pages, and I could not find an adequate dataset for visualizations I had planned. Therefore, I pivoted by looking for a strong dataset, knowing that I wanted to learn D3.js, and chose to do wine instead.

Part 5: Challenges

My biggest challenge (that took several days to solve) was figuring out how to make my D3.js visualizations responsive to the window screen size, particularly coupled with the fact that I was using a Bootstrap responsive grid. While learning how to use D3.js to draw paths for countries was also tough, another challenge I faced was cleaning the wine dataset I found on Kaggle. It contained over 150,000 different wines, with numerous columns of data left blank/missing, as well as many European special characters that were not correctly encoded. I self-learned some Python libraries to help me filter all these out to narrow down to the 52,091 wines that were used as data for this project.

Thanks for a great semester!