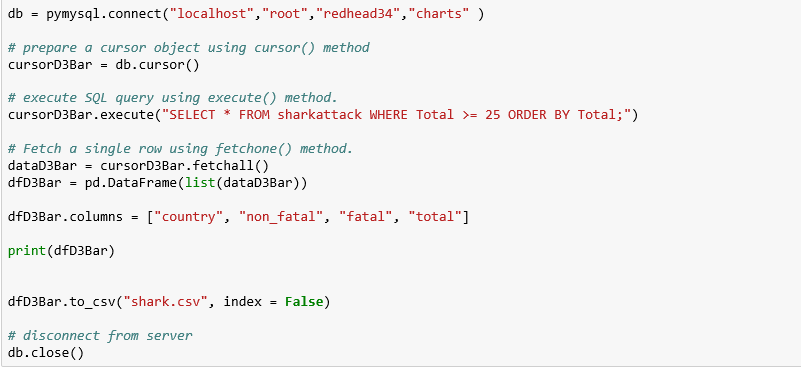
**Documentation, Final Project**

Jackson Crum

**MySQL storage and Python Fetching**

To store the data in a MySQL database, I used MySQL Workbench to create the schema and tables and populate the tables with the data. I created one schema titled “charts” that contained six tables, one for each individual chart dataset. This was completed with the schema navigator and schema inspector.

To fetch the data from the database and get it prepared for visualization, I first attempted to use AngularJS. After struggling with angular, I decided to use Python’s pymysql package to import data from the SQL database using my credentials and SQL queries. For each dataset needed for graphing, I established a connection to the database, prepared a cursor, and used the cursor to execute a SQL query and fetch the queried data. I then manipulated the resulting pandas DataFrames to get them in proper format for graphing in D3.js or Highcharts and exported the dataframe as a csv. **In order to fetch the data from the MySQL database, the Python code must be run prior to running the server in order for the charts to have the necessary data to plot.**



**Node.js Server and index.js**

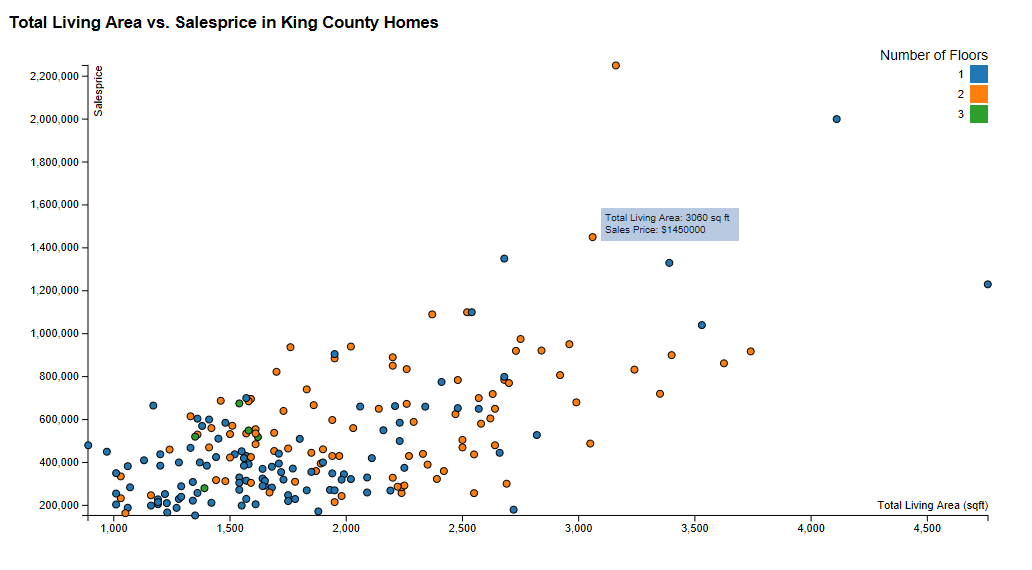
To display the graphs on the web, I used node.js to create the server and index.js to connect the graphs to server. To create a module, I created the package.json file with npm init and downloaded the necessary packages. I used the index.js file to connect the graphs to the server with html hyperlink tag (<a href=>) and create the visuals for the page. The server is connected to port 3429. Only express needed to be installed for this server.

**Chart 1: D3.js, Scatter Chart**

The D3.js scatter plot uses house price sales from King County, Washington (including Seattle) and includes homes sold between May 2014 and May 2015. The data contains and id column, 19 features, and 21613 observations. To see if there was a relationship between total living area and sales price, I queried the MySQL database for those two columns and plotted the data by binding total living area to the x-axis, sales price to the y-axis, and number of floors to color. I created a legend, added axis labels, and stroke around each dot svg to give clear visualization of the data.

As can be seen by the chart, there is a small positive correlation between total living and sales price, though there is a large amount of spread. There is no obvious correlation between number of floors and sales price. To add to the informative abilities of the chart, I made it interactive using jQuery’s mouseover and mouseout events to display the attributes of each point. The event binds a tooltip to a div element and displays the attribute values when the cursor is scrolled over the dot.

<https://www.kaggle.com/harlfoxem/housesalesprediction/data>



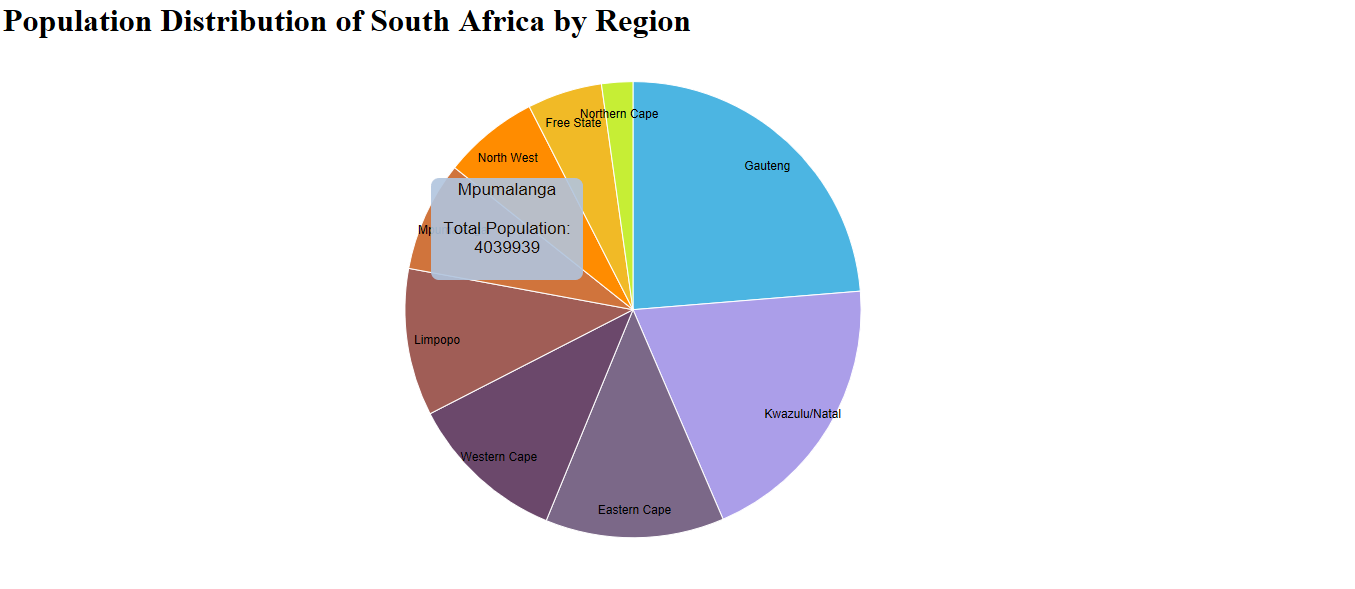
<https://www.kaggle.com/uciml/iris>

**Chart 2: D3.js, Pie Chart**

This pie plot uses D3.js api to plot the population distribution of South Africa by region. The data comes from a large database on South African crime statistics. The data contains three features (population, area, and density) for each of South Africa’s nine provinces. To get the province name and population for each province, I queried the MySQL database for those two columns and bound each population an arc svg. I labelled each arc svg with the province name and made a custom color spectrum that was called to fill the arcs.

As can be seen by the pie chart, almost of half the population is contained in just two of the nine provinces, Gauteng and Kwazulu/Natal, which each contain of 10 million people. Gauteng has approximately the same population as the bottom four provinces, with Northern Cape and Free State each having less than 3 million people. To display the population of each province quickly and informatively, I made it interactive using jQuery’s mouseover and mouseout events to display population and province name of each point. The event binds a tooltip to a div element and displays the attribute values when the cursor is scrolled over the dot.

<https://www.kaggle.com/slwessels/crime-statistics-for-south-africa/data>

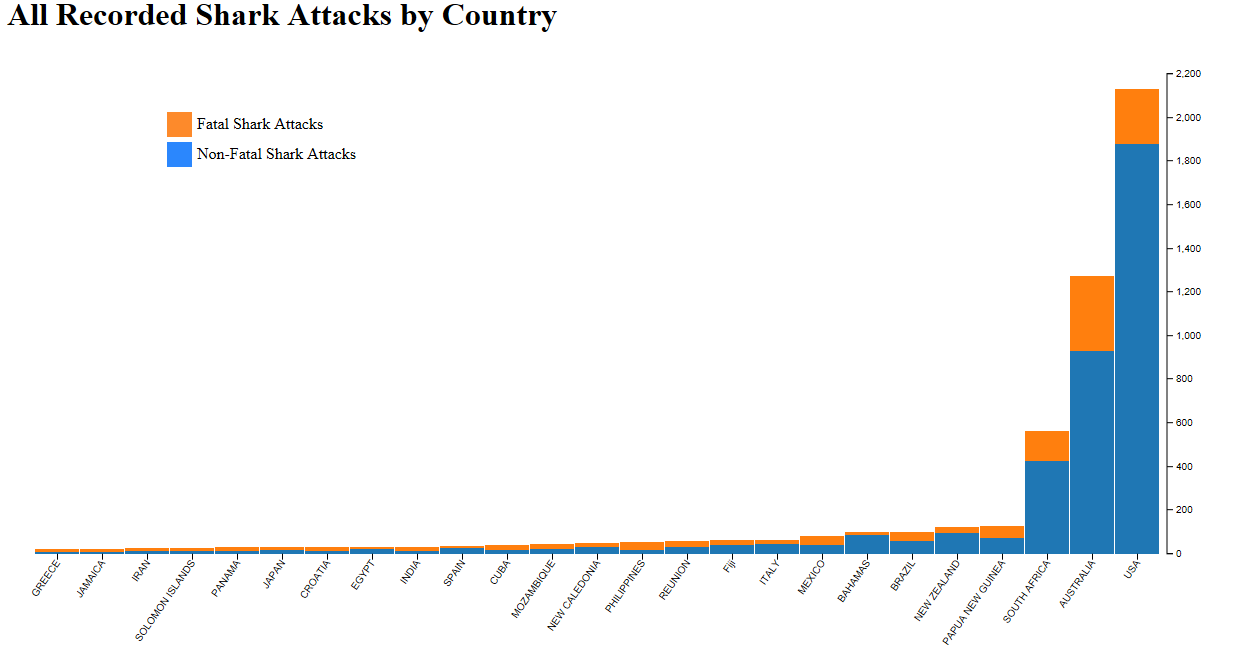


**Chart 3: D3.js, Stacked Bar Chart**

This stacked bar chart uses D3.js api to plot all recorded shark attacks by the country/location of the attack and stacks the bars based on whether the attack was fatal or not. The data comes from a large database on South African crime statistics. The data contains 22 features and 6094 observations. I used an Excel pivot tables to get total attacks per location and split fatal and non-fatal in dummy variables and imported the table into MySQL Workbench. To get the country name, total attacks, fatal, and non-fatal, I queried the MySQL database for those columns and bound each total attack value to a rect svg. I defined the stacking variables, mapped survival on each bar using layout.stack() to the layout variable, mapped the layout variable to a rect svg to create the stacked bar chart, and created a legend to show the proportional of fatal to non-fatal attacks.

The chart shows that the majority of the attacks are shown to have taken place in the United States, Australia, and South Africa. The chart also shows that the majority of shark attacks are non-fatal, although Australia tends to have a higher fatality percentage than other locations. Also, given the fact that there are only 6094 observations, the chart shows that shark attacks are not very common as many believe them to be.

<https://www.kaggle.com/teajay/global-shark-attacks/data>

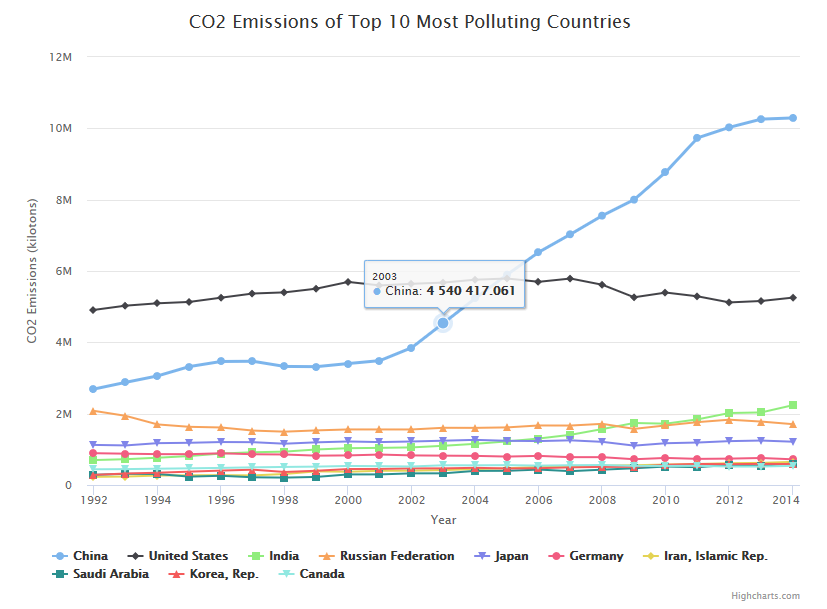


**Chart 4: HighCharts, Line Chart**

For this chart, I am using HighCharts api to create a line graph showing the CO2 emissions in kilotons of the top ten most polluting countries from 1992-2014. The data was collected by the Carbon Dioxide Information Analysis Center and published by the World Bank. The countries used for this chart are China, the United States, India, Russia, Japan, Germany, Iran, Saudi Arabia, South Korea, and Canada. I bound the year to the x-axis and the CO2 Emissions on the y-axis. To get the country name and emission values for year, I queried the MySQL database for all year columns and bound the year to the x-axis and emissions to the y-axis with each line representing a country.

Highcharts uses high levels of interactivity and visualization to produce very aesthetically pleasing and informative charts. As the chart shows, the United States had over double the emissions as all other nations except China. While the US still emits a significant amount of pollutants, China surpassed the US in 2005 in emissions and now has double the total American emissions. With the exception of China, all countries have relatively steady emissions amounts from 1992-2014.

<https://data.worldbank.org/indicator/EN.ATM.CO2E.KT>

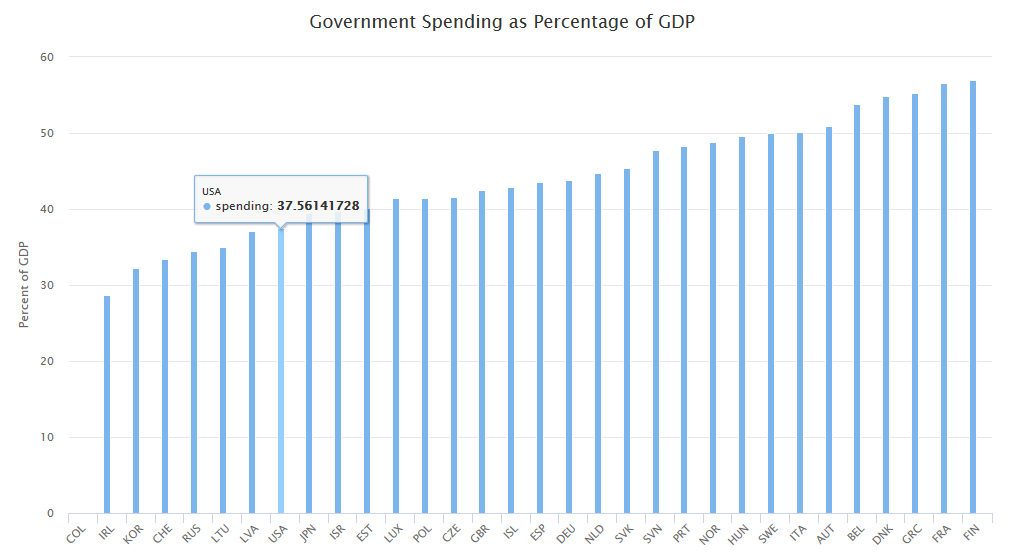


**Chart 5: HighCharts, Bar Chart**

For this chart, I am using HighCharts api to develop a bar graph showing general government spending as a percent of GDP. General government spending consists of spending on central, state and local governments, and social security funds. This indicator is measured in terms of thousand USD per capita and as a percentage of GDP. This measure provides an indication of the size of each country’s government. The data consists of the general government spending of 32 nations. The data was collected and published by the Organization for Economic Co-operation and Development (OECD).

To get the country name and government spending, I queried the MySQL database for those columns and bound the year to x-axis and spending to the y-axis with each bar representing a country and bar height representing spending percentage. European nations spend significantly more on government than any other nations, with Finland being the highest spender at 56.96% of GDP spent on government. Colombia, on the other hand, spends only 0.03% of GDP on government.

<https://data.oecd.org/gga/general-government-spending.htm>



**Chart 6: HighCharts, Area Chart**

For this chart, I am using HighCharts api to develop an area graph that shows new education commitments from the International Development Association of World Bank for each year split by region. The original data consists of 7 features and 780 observations. I used to Excel pivot table to manipulate the dataset so that regions were the columns, years were the rows, and total new commitments (in $USD millions) for each year and region was the value. There are seven regions and the data goes from 1990 to 2015, 16 years total. To get the country name, year, and government spending, I queried the MySQL database for those columns and bound the year to x-axis, commitments to the y-axis, and country to each area plot. I rearranged the columns using Python to order the regions so that the largest area would be in the back and make a more aesthetically pleasing chart.

The chart shows that South Asia and Africa get the most commitments from the IDA, with South Asia getting just over $1.4 billion in 2010 and Africa getting $800 million in 2009. This chart utilizes the Highcharts api to make a very informative and beautiful graph by adjusting opacity and color so each area can be seen and provides interactive events to display information at each plotted point.

<https://data.worldbank.org/data-catalog/64f3-59mr>

