Increases in Drug and Opioid Overdose Deaths — United States, 1999–2014

Arun Rajan, Charan Teja Lellaboyena

May 15, 2017

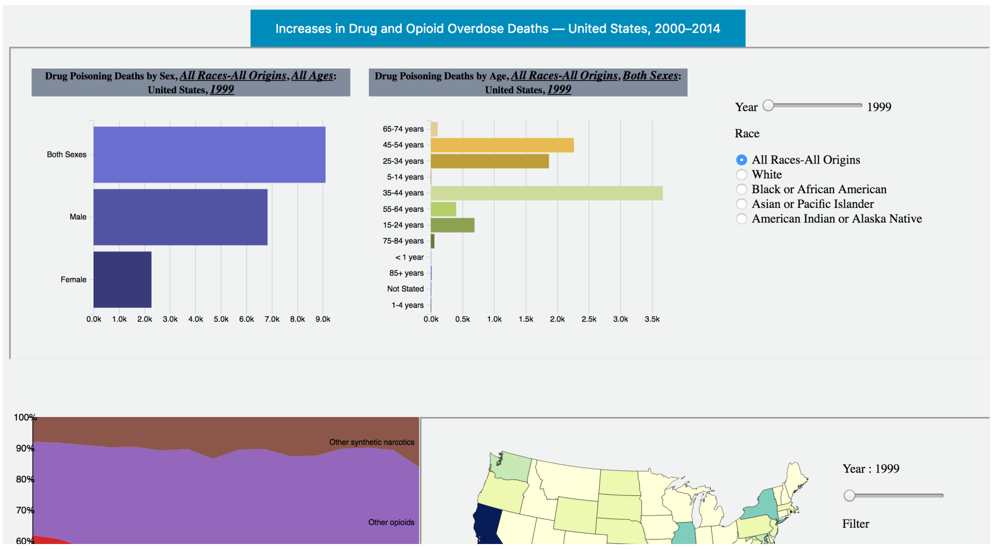
1 Introduction

Opioid addiction and death rates in the U.S. and abroad have reached "epidemic" levels. Every single day, many lives are lost due to careless and unfortunate habits of drugs use. As per a report from 'Center for Disease Control Prevention' the United States is experiencing an epidemic of drug overdose (poisoning) deaths. Since 2000, the rate of deaths from drug overdoses has increased 137%, including a 200% increase in the rate of overdose deaths involving opioids[2] (opioid pain relievers and heroin)[1]. Especially, overdose death linked to prescription opioids is a growing public health crisis. The current epidemic accelerated during a fifteen-year interval between 1999 and 2014. In particular, the mortality rate from drug overdose tripled between 1999 and 2014 (CDC, 2015): by 2014, the number of overdose deaths in the United States exceeded the number of suicides; by 2014, they also exceeded the number of motor vehicle deaths. Prescription opioid analgesics, painkillers derived from opium or synthesized to possess narcotic properties similar to opioids, including oxycodone, hydrocodone and codeine, have driven this overdose epidemic (CDC, 2011; Warner et al., 2009). Prescription opioids may have been obtained legally through a prescription, or diverted through illegal distribution networks.
This document captures the final report made by our team for the final project in CSE-564, Visualization. We discuss the different approaches, ideas and challenges that we met during it completion.

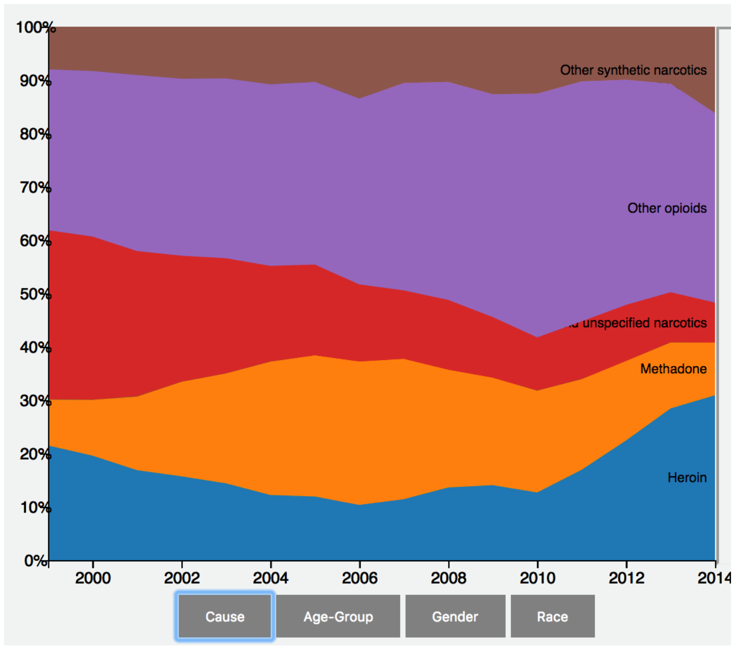
2 Progress

2.1 Insights & Visual Analysis

We brainstormed on the different dimensions that could be shown on the dashboard. These are majorly divided into Geographical and temporal divisions. Further, we have tried to correlate two different data sets, related to retail sales of opioid based analgesic drugs and deaths related to Opioid overdoses.
We started using a Top-down approach, first showing trends over the time period of 15 years. The visualization shown below, that is part of the dashboard,

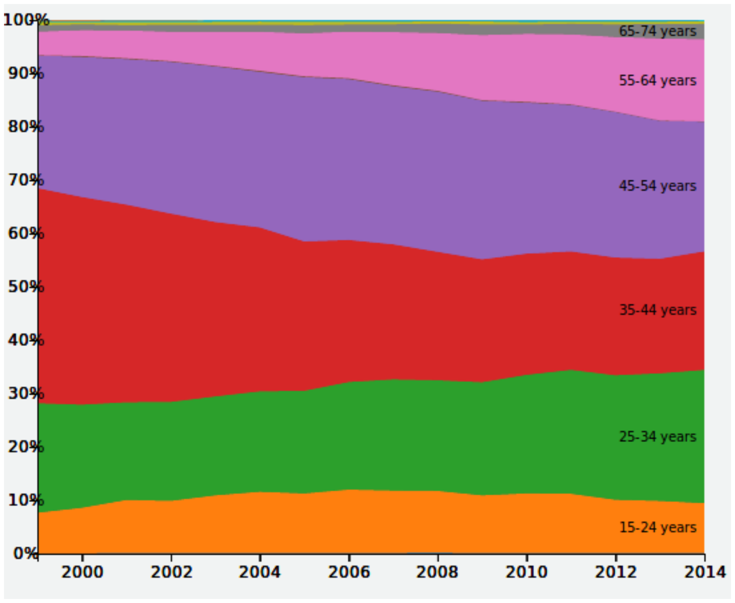

Increases in Drug and Opioid Overdose Deaths — United States, 2000–2014
Drug Poisoning Deaths by Sex, All Races-All Origins, All Ages: United States, 1999
Both Sexes
Male
Female
Drug Poisoning Deaths by Age, All Races-All Origins, Both Sexes: United States, 1999
65-74 years
45-54 years
25-34 years
5-14 years
35-44 years
55-64 years
15-24 years
75-84 years
< 1 year
85+ years
Not Stated
1-4 years
Year 1999
Race
All Races-All Origins
White
Black or African American
Asian or Pacific Islander
American Indian or Alaska Native
Other synthetic narcotics
Other opioids
Year : 1999
Filter

depicts the contribution in number of deaths along Y-axis by different elements like Cause, Age-Group, Gender and Race. One can easily notice that the use of Heroine increased after 2010.



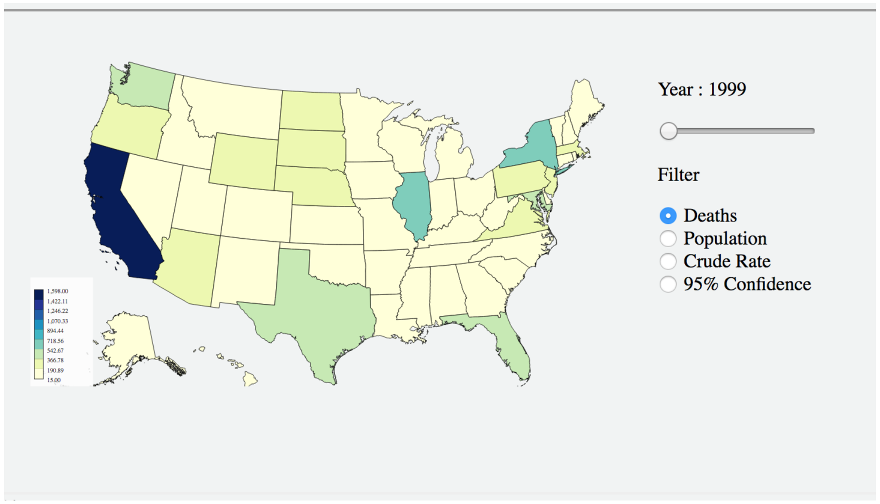
Adults aged 45–54 had the highest rate of drug overdose deaths in 2014.The rates of drug overdose deaths increased from 1999 to 2014 for all age groups. The greatest percentage increase occurred in the drug overdose death rate for adults aged 55–64, from 4.2 per 100,000 in 1999 to 21.8 in 2014, an average increase of 10.5% per year. In 2014, adults aged 45–54 had the highest death rate from drug overdose at 30 deaths per 100,000. In 2014, rates for adults aged 25–34, 35–44, 45–54, and 55–64 were more than twice the rate for younger adults aged 15–24, and more than 3.5 times the rate for adults aged 65 and over.

Clicking on the 'race' tab below the area-time graph, one can notice that 'White' race group has been affected the most by this epidemic.

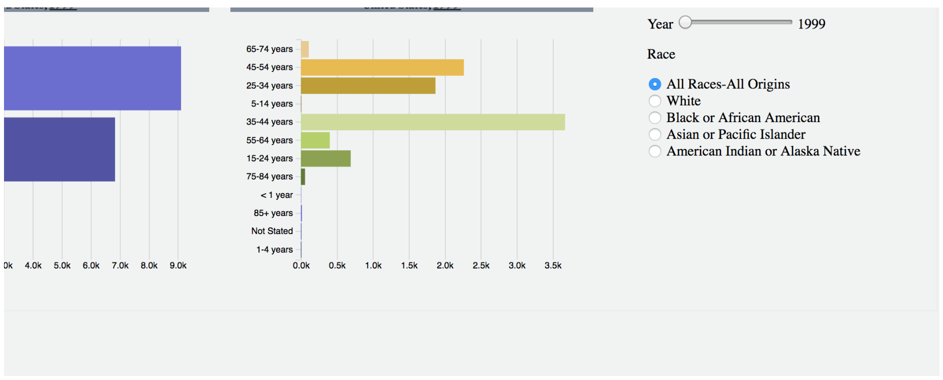


For the Geographical projection, we provided a US map that uses FIPS codes for different states. A time filter for different years is provided that helps seeing the trends in deaths, population, crude rate over different states. Below is the screen-shot. By paying attention to the crude rate as per different geographic regions, it is evident that the west region has constant high use of Heroin related drugs. However, in the states, West Virginia tops the list.

Now once we have seen the top-down trends, we can drill down and see details for different dimensions. A horizontal stacked bar chart, divided into portions,

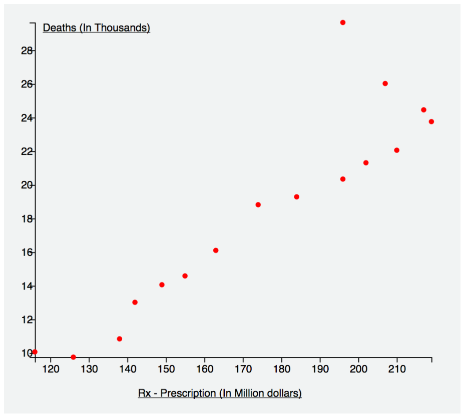


shows the number of deaths in different race groups, different sex over the span of 15 years. This visualization is interactive in nature. Due care has been taken while creating the dash-baord and both the sides in this visualization are linked. By cicking on one, the changes are reflected in the other. Below is the screen-shot.

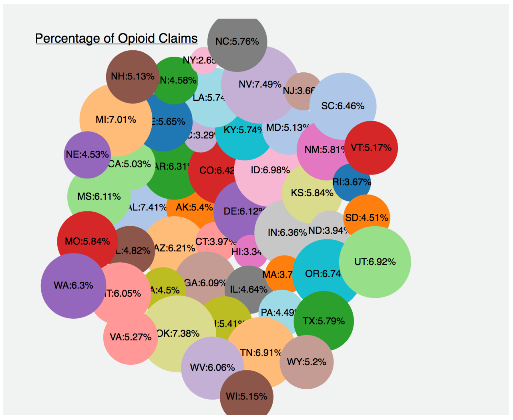


2.1.1 Correlation

We found a positive a correlation between the number of prescription sold and deaths. It strengthens the argument that the people use drugs given by the doctors, first to ease the pain, but with time, they become addicted to the Opioid content present. Eventually, the get addicted to Heroin as the tolerance builds up. It is shown by a rather simple scatter plot shown below

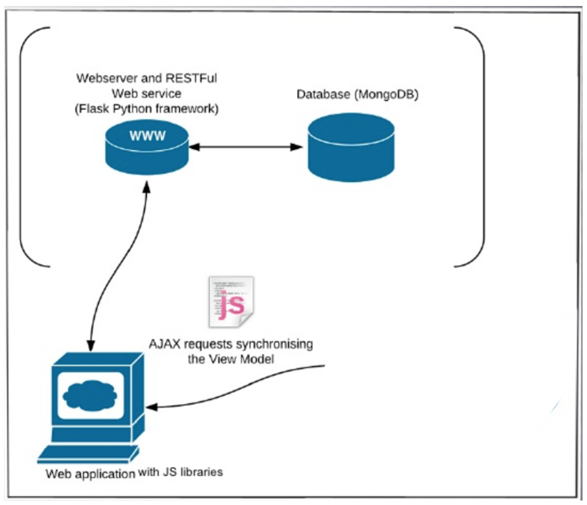


Finally, we also projected the number of Opioid based claims on the total number of claims filled in different states. We showed it using a bubble chart, where the radius explains the ratio proportionally. The bubble chart is shown below:



2.2 Framework Setup

The framework for the project as detailed in the primary proposal report uses Python-Flask as web server. For the database requirements, we set up a non-relational database, MongoDB. However, we are also using csv files in some part of the projects. To support the communication between Python code and MongoDB[4], we are using Pymongo. On the client side, we are using D3 for visualization. For asynchronous service calls from client to server (Flask) are made using AJAX[6] http requests with JSON responses to accommodate the REST requests(service calls).[8]

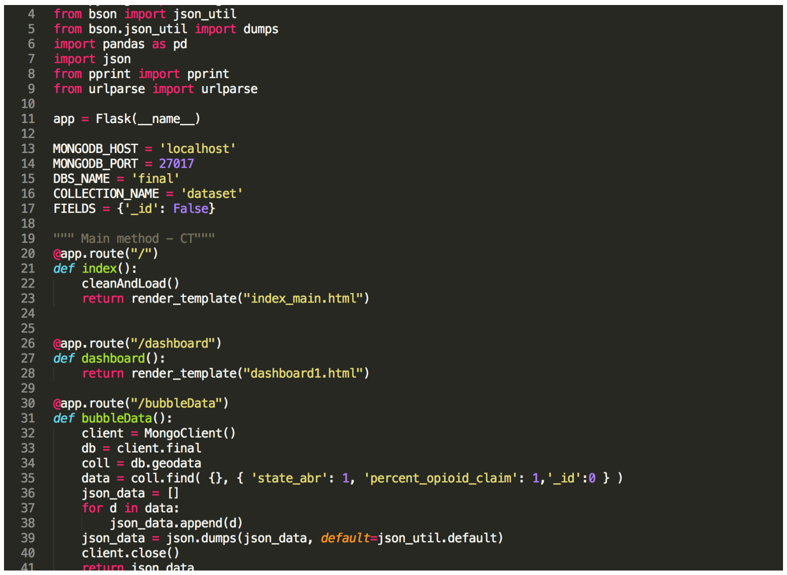


2.3 Data Cleaning

The originally gathered data [3] had many columns that contained null or invalid data. It was necessary to clean and fill the spaces with appropriate heuristics before loading the data in the mongoDB. For example, the value in the 'crude rate' column was filled by taking the average of 95% lower and upper crude rates.

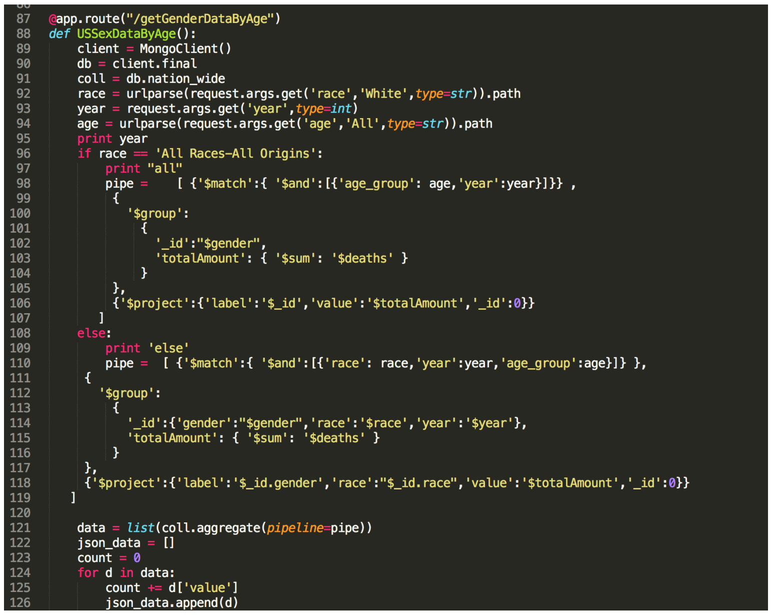
2.4 Implementation

Our code implementation is in JavaScript, Python and JQuery. Most of the pre-processing part has been done in python. It includes data cleaning and restructuring such as a pivoting. We would like to mention that we used Geo-Map[7], a d3 extension library for geographical projections. Python-Flask framework was used to setup the endpoints for REST calls. This section includes the code snippets for different part of the projects.
Code Snippet for the database connection



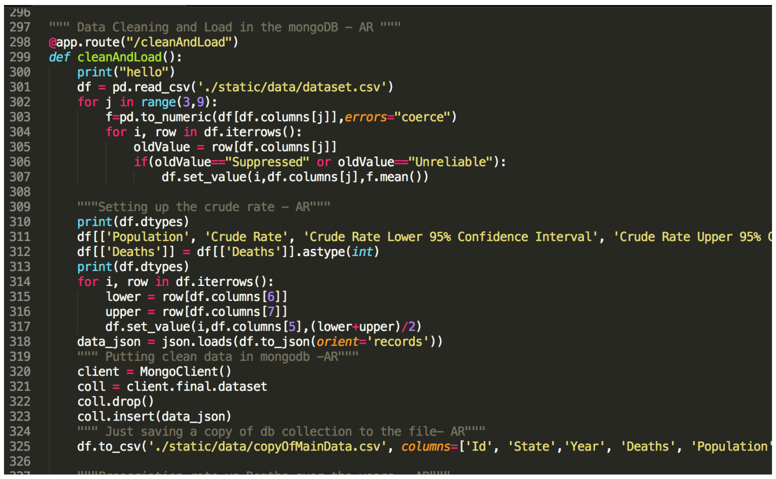
```python
from bson import json_util
from bson.json_util import dumps
import pandas as pd
import json
from pprint import pprint
from urlparse import urlparse

app = Flask(__name__)

MONGODB_HOST = 'localhost'
MONGODB_PORT = 27017
DBS_NAME = 'final'
COLLECTION_NAME = 'dataset'
FIELDS = {'_id': False}

""" Main method - CT"""
@app.route("/")
def index():
    cleanAndLoad()
    return render_template("index_main.html")


@app.route("/dashboard")
def dashboard():
    return render_template("dashboard1.html")

@app.route("/bubbleData")
def bubbleData():
    client = MongoClient()
    db = client.final
    coll = db.geodata
    data = coll.find( {}, { 'state_abr': 1, 'percent_opioid_claim': 1,'_id':0 } )
    json_data = []
    for d in data:
        json_data.append(d)
    json_data = json.dumps(json_data, default=json_util.default)
    client.close()
    return json_data
```

Code Snippet for illustrating how to access MongoDb and use Python Flask



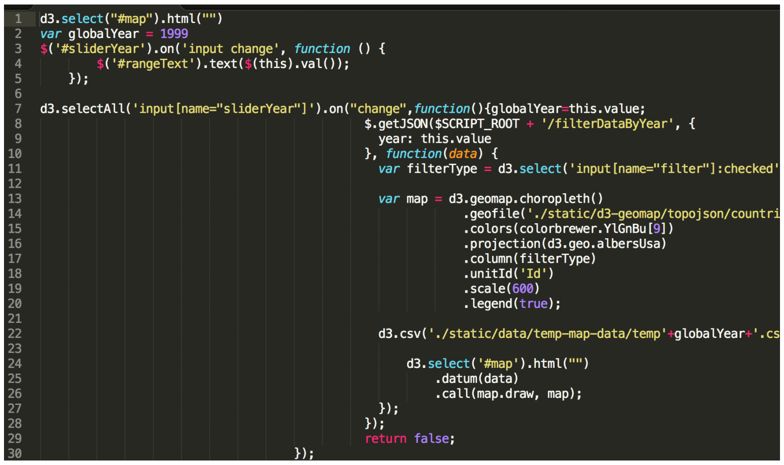
```python
@app.route("/getGenderDataByAge")
def USSexDataByAge():
    client = MongoClient()
    db = client.final
    coll = db.nation_wide
    race = urlparse(request.args.get('race','White',type=str)).path
    year = request.args.get('year',type=int)
    age = urlparse(request.args.get('age','All',type=str)).path
    print year
    if race == 'All Races-All Origins':
        print "all"
        pipe =    [ {'$match':{ '$and':[{'age_group': age,'year':year}]}} ,
         {
           '$group':
             {
              '_id':"$gender",
              'totalAmount': { '$sum': '$deaths' }
             }
         },
         {'$project':{'label':'$_id','value':'$totalAmount','_id':0}}
       ]
    else:
        print 'else'
        pipe =  [ {'$match':{ '$and':[{'race': race,'year':year,'age_group':age}]} },
     {
       '$group':
         {
           '_id':{'gender':"$gender",'race':'$race','year':'$year'},
           'totalAmount': { '$sum': '$deaths' }
         }
     },
     {'$project':{'label':'$_id.gender','race':"$_id.race",'value':'$totalAmount','_id':0}}
   ]

    data = list(coll.aggregate(pipeline=pipe))
    json_data = []
    count = 0
    for d in data:
        count += d['value']
        json_data.append(d)
```

Code Snippet for illustrating how to clean data using Python Pandas and dump json into MongoDb



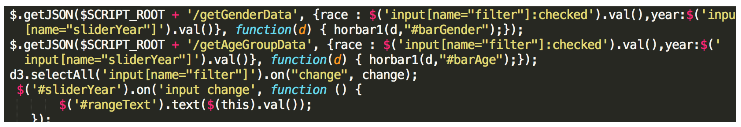
```python
""" Data Cleaning and Load in the mongoDB - AR """
@app.route("/cleanAndLoad")
def cleanAndLoad():
    print("hello")
    df = pd.read_csv('./static/data/dataset.csv')
    for j in range(3,9):
        f=pd.to_numeric(df[df.columns[j]],errors="coerce")
        for i, row in df.iterrows():
            oldValue = row[df.columns[j]]
            if(oldValue=="Suppressed" or oldValue=="Unreliable"):
                df.set_value(i,df.columns[j],f.mean())

    """Setting up the crude rate - AR"""
    print(df.dtypes)
    df[['Population', 'Crude Rate', 'Crude Rate Lower 95% Confidence Interval', 'Crude Rate Upper 95% C
    df[['Deaths']] = df[['Deaths']].astype(int)
    print(df.dtypes)
    for i, row in df.iterrows():
        lower = row[df.columns[6]]
        upper = row[df.columns[7]]
        df.set_value(i,df.columns[5],(lower+upper)/2)
    data_json = json.loads(df.to_json(orient='records'))
    """ Putting clean data in mongodb -AR"""
    client = MongoClient()
    coll = client.final.dataset
    coll.drop()
    coll.insert(data_json)
    """ Just saving a copy of db collection to the file- AR"""
    df.to_csv('./static/data/copyOfMainData.csv', columns=['Id', 'State','Year', 'Deaths', 'Population'

```

Code Snippet for illustrating Geo Map and time slider



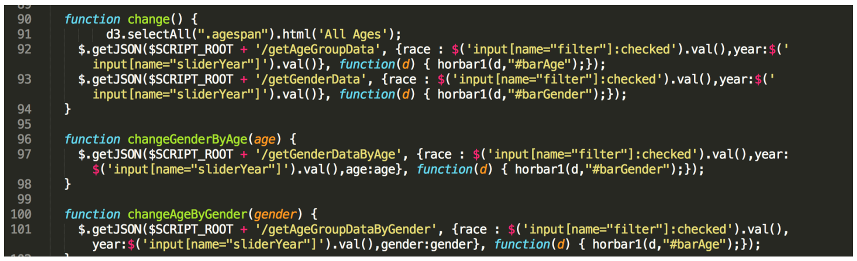
```javascript
d3.select("#map").html("")
var globalYear = 1999
$('#sliderYear').on('input change', function () {
        $('#rangeText').text($(this).val());
    });

d3.selectAll('input[name="sliderYear"]').on("change",function(){globalYear=this.value;
                                                $.getJSON($SCRIPT_ROOT + '/filterDataByYear', {
                                                  year: this.value
                                                }, function(data) {
                                                  var filterType = d3.select('input[name="filter"]:checked'

                                                  var map = d3.geomap.choropleth()
                                                              .geofile('./static/d3-geomap/topojson/countri
                                                              .colors(colorbrewer.YlGnBu[9])
                                                              .projection(d3.geo.albersUsa)
                                                              .column(filterType)
                                                              .unitId('Id')
                                                              .scale(600)
                                                              .legend(true);

                                                  d3.csv('./static/data/temp-map-data/temp'+globalYear+'.cs

                                                      d3.select('#map').html("")
                                                          .datum(data)
                                                          .call(map.draw, map);
                                                  });
                                                });
                                                return false;
                                    });
```

Code Snippet for illustrating how to use JQuery AJAX calls



```javascript
$.getJSON($SCRIPT_ROOT + '/getGenderData', {race : $('input[name="filter"]:checked').val(),year:$('inpu
  [name="sliderYear"]').val()}, function(d) { horbar1(d,"#barGender");});
$.getJSON($SCRIPT_ROOT + '/getAgeGroupData', {race : $('input[name="filter"]:checked').val(),year:$('
  input[name="sliderYear"]').val()}, function(d) { horbar1(d,"#barAge");});
d3.selectAll('input[name="filter"]').on("change", change);
 $('#sliderYear').on('input change', function () {
        $('#rangeText').text($(this).val());
   });
```

Code Snippet for illustrating how to link two visualizations



```
function change() {
        d3.selectAll(".agespan").html('All Ages');
  $.getJSON($SCRIPT_ROOT + '/getAgeGroupData', {race : $('input[name="filter"]:checked').val(),year:$('
    input[name="sliderYear"]').val()}, function(d) { horbar1(d,"#barAge");});
  $.getJSON($SCRIPT_ROOT + '/getGenderData', {race : $('input[name="filter"]:checked').val(),year:$('
    input[name="sliderYear"]').val()}, function(d) { horbar1(d,"#barGender");});
}

function changeGenderByAge(age) {
  $.getJSON($SCRIPT_ROOT + '/getGenderDataByAge', {race : $('input[name="filter"]:checked').val(),year:
    $('input[name="sliderYear"]').val(),age:age}, function(d) { horbar1(d,"#barGender");});
}

function changeAgeByGender(gender) {
  $.getJSON($SCRIPT_ROOT + '/getAgeGroupDataByGender', {race : $('input[name="filter"]:checked').val(),
    year:$('input[name="sliderYear"]').val(),gender:gender}, function(d) { horbar1(d,"#barAge");});
```

3 References

1 [https://www.cdc.gov/mmwr/preview/mmwrhtml/mm6450a3.htm]
2 [https://en.wikipedia.org/wiki/Opioid]
3 [https://data.world/health/opioid-overdose-deaths]
4 [https://en.wikipedia.org/wiki/MongoDB]
5 [https://en.wikipedia.org/wiki/Flask$_{(}web_{f}ramework)$]
6$[https://cdnjs.cloudflare.com/ajax/libs/jquery/3.0.0-beta1/jquery.min.js]$
7$[https://d3-geomap.github.io/]$
8$[https://image.slidesharecdn.com/finalpresentation-120501061120-phpapp01/95/listentoitlater-final-year-project-presentation-13-728.jpg?cb=1365141016]$