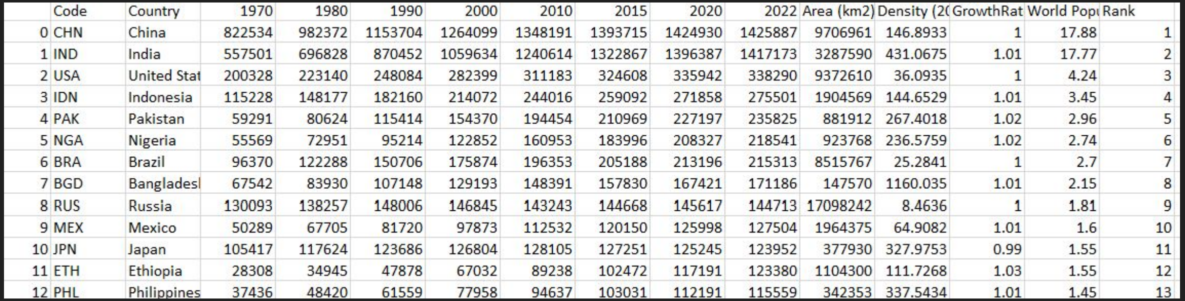
World Population Growth

Caroline Foshee, Eleanor Hayden, Sivangi Raychoudhury, Seth Ritter

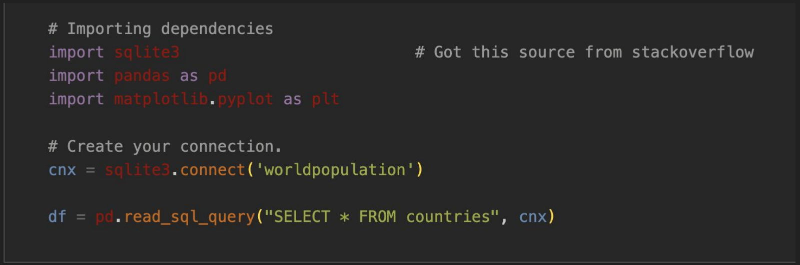
Our Data

- 234 countries in our dataset
  - Includes territories like Gibraltar and Greenland
- 8 years of population data
- Other variables for 2022
  - Area (km2), Density, Growth rate, % of the world population, Rank

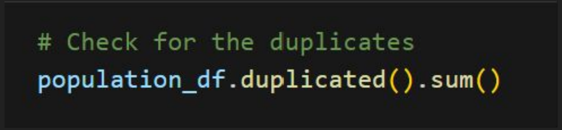


	Code	Country	1970	1980	1990	2000	2010	2015	2020	2022	Area (km2)	Density (2(	GrowthRat	World Popı	Rank
0	CHN	China	822534	982372	1153704	1264099	1348191	1393715	1424930	1425887	9706961	146.8933	1	17.88	1
1	IND	India	557501	696828	870452	1059634	1240614	1322867	1396387	1417173	3287590	431.0675	1.01	17.77	2
2	USA	United Stat	200328	223140	248084	282399	311183	324608	335942	338290	9372610	36.0935	1	4.24	3
3	IDN	Indonesia	115228	148177	182160	214072	244016	259092	271858	275501	1904569	144.6529	1.01	3.45	4
4	PAK	Pakistan	59291	80624	115414	154370	194454	210969	227197	235825	881912	267.4018	1.02	2.96	5
5	NGA	Nigeria	55569	72951	95214	122852	160953	183996	208327	218541	923768	236.5759	1.02	2.74	6
6	BRA	Brazil	96370	122288	150706	175874	196353	205188	213196	215313	8515767	25.2841	1	2.7	7
7	BGD	Bangladesl	67542	83930	107148	129193	148391	157830	167421	171186	147570	1160.035	1.01	2.15	8
8	RUS	Russia	130093	138257	148006	146845	143243	144668	145617	144713	17098242	8.4636	1	1.81	9
9	MEX	Mexico	50289	67705	81720	97873	112532	120150	125998	127504	1964375	64.9082	1.01	1.6	10
10	JPN	Japan	105417	117624	123686	126804	128105	127251	125245	123952	377930	327.9753	0.99	1.55	11
11	ETH	Ethiopia	28308	34945	47878	67032	89238	102472	117191	123380	1104300	111.7268	1.03	1.55	12
12	PHL	Philippines	37436	48420	61559	77958	94637	103031	112191	115559	342353	337.5434	1.01	1.45	13

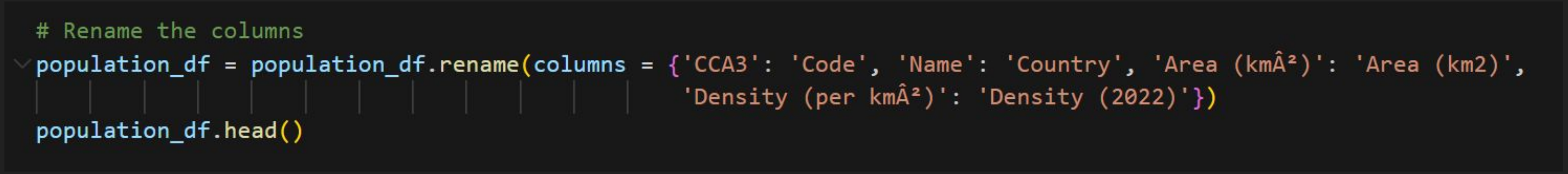
Cleaning the data



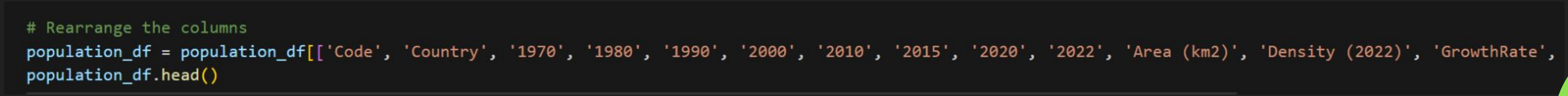
```
# Importing dependencies
import sqlite3                         # Got this source from stackoverflow
import pandas as pd
import matplotlib.pyplot as plt

# Create your connection.
cnx = sqlite3.connect('worldpopulation')

df = pd.read_sql_query("SELECT * FROM countries", cnx)
```



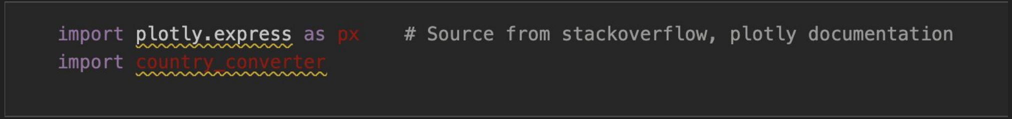
```
# Check for the duplicates
population_df.duplicated().sum()
```



```
# Rename the columns
population_df = population_df.rename(columns = {'CCA3': 'Code', 'Name': 'Country', 'Area (kmÂ²)': 'Area (km2)',
                                                'Density (per kmÂ²)': 'Density (2022)'})
population_df.head()
```

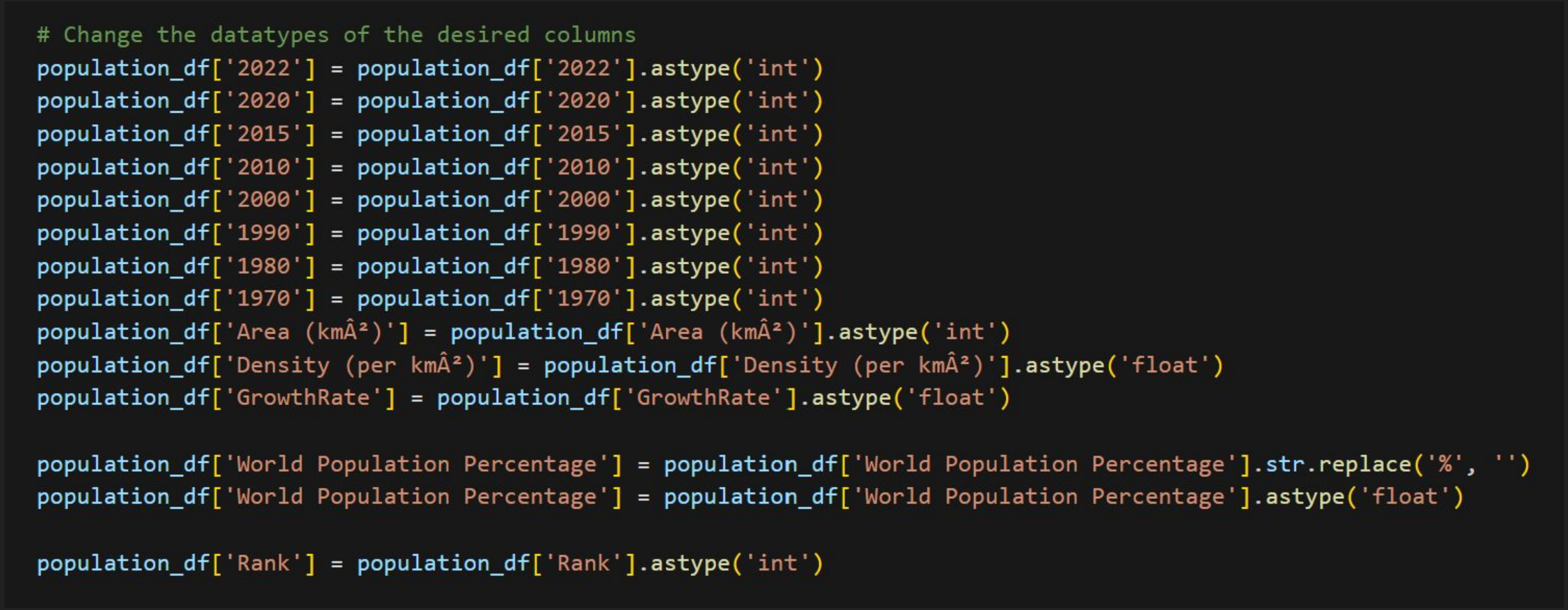


```
# Rearrange the columns
population_df = population_df[['Code', 'Country', '1970', '1980', '1990', '2000', '2010', '2015', '2020', '2022', 'Area (km2)', 'Density (2022)', 'GrowthRate',
population_df.head()
```



```
import plotly.express as px     # Source from stackoverflow, plotly documentation
import country_converter
```

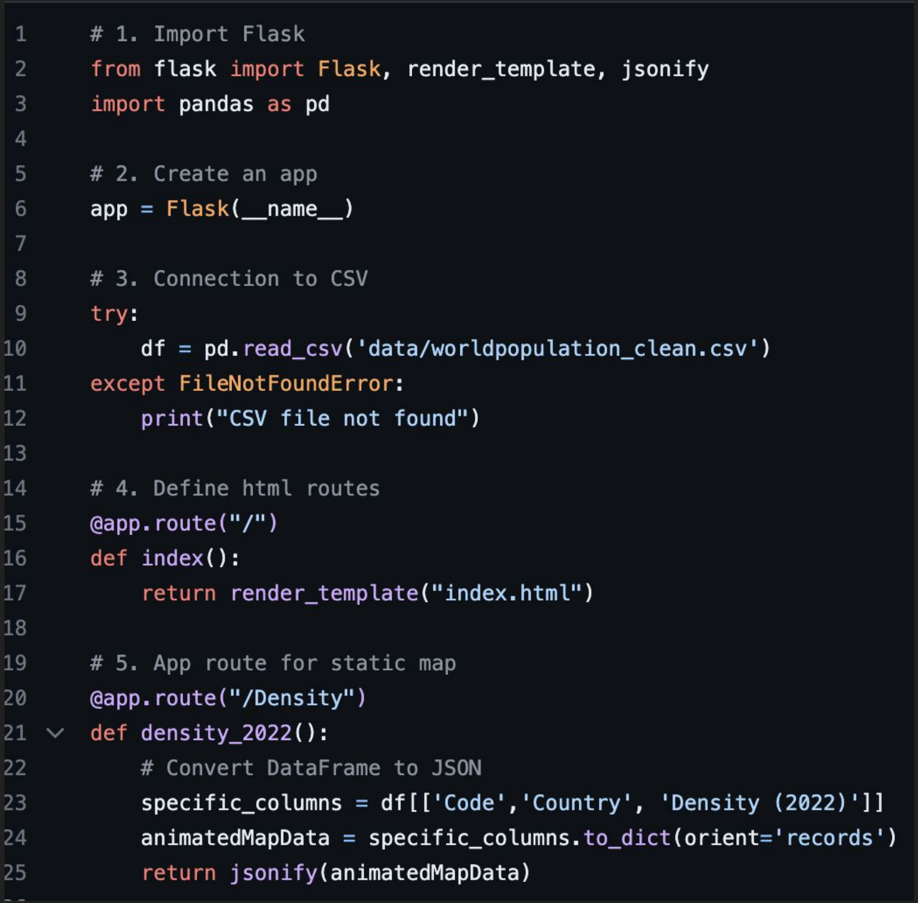
Cleaning the data



```
# Change the datatypes of the desired columns
population_df['2022'] = population_df['2022'].astype('int')
population_df['2020'] = population_df['2020'].astype('int')
population_df['2015'] = population_df['2015'].astype('int')
population_df['2010'] = population_df['2010'].astype('int')
population_df['2000'] = population_df['2000'].astype('int')
population_df['1990'] = population_df['1990'].astype('int')
population_df['1980'] = population_df['1980'].astype('int')
population_df['1970'] = population_df['1970'].astype('int')
population_df['Area (kmÂ²)'] = population_df['Area (kmÂ²)'].astype('int')
population_df['Density (per kmÂ²)'] = population_df['Density (per kmÂ²)'].astype('float')
population_df['GrowthRate'] = population_df['GrowthRate'].astype('float')

population_df['World Population Percentage'] = population_df['World Population Percentage'].str.replace('%', '')
population_df['World Population Percentage'] = population_df['World Population Percentage'].astype('float')

population_df['Rank'] = population_df['Rank'].astype('int')
```

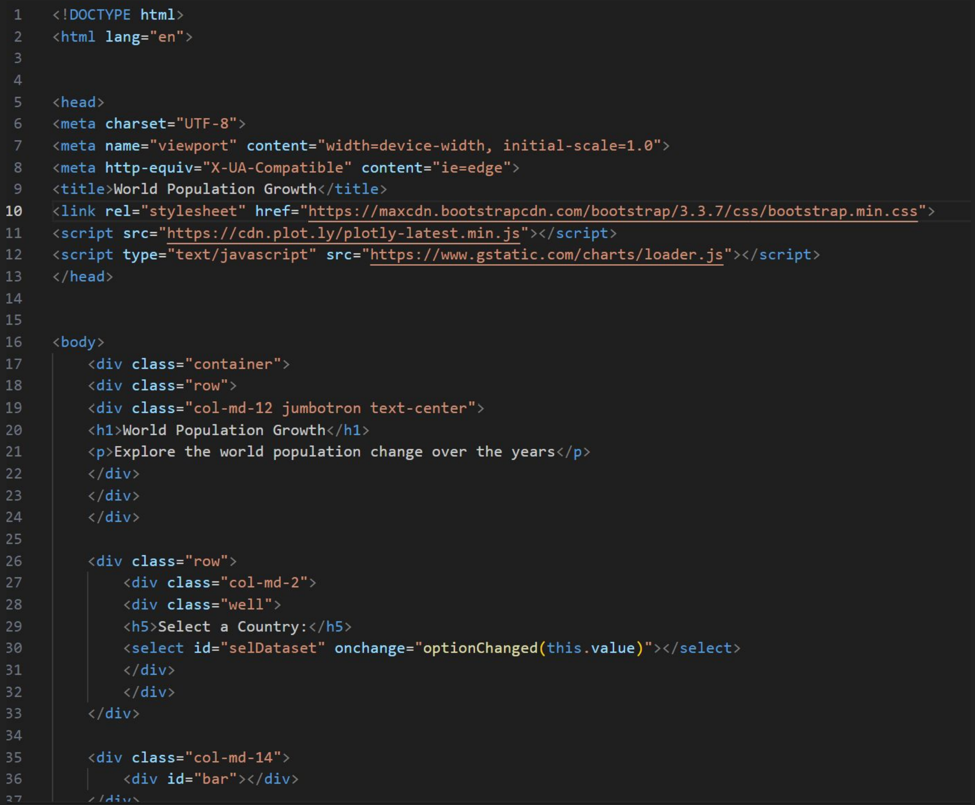
Our API

- Main link: http://127.0.0.1:8000
- Routes:
  - /Density
  - /Population
  - /Country

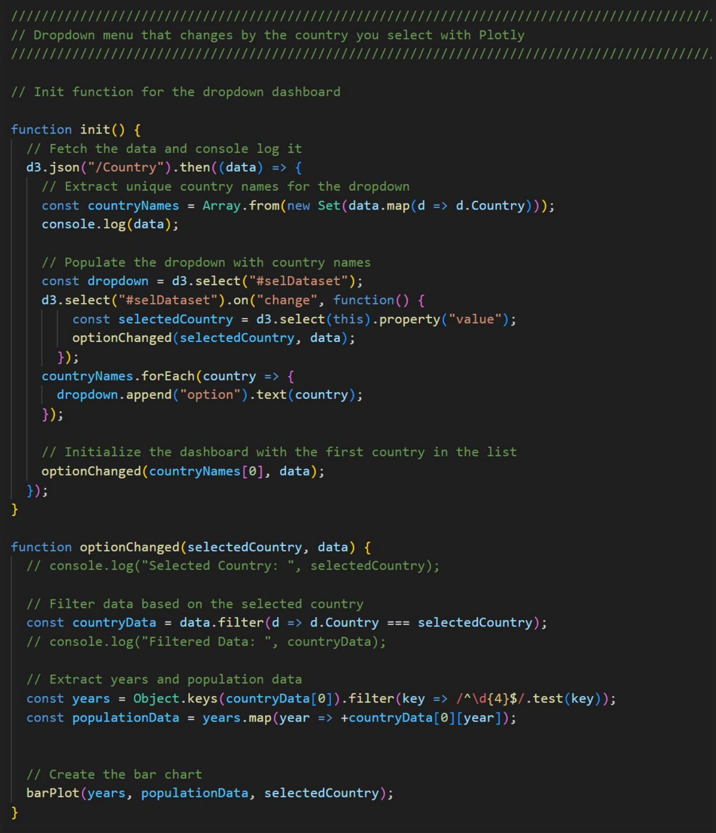


```python
# 1. Import Flask
from flask import Flask, render_template, jsonify
import pandas as pd

# 2. Create an app
app = Flask(__name__)

# 3. Connection to CSV
try:
    df = pd.read_csv('data/worldpopulation_clean.csv')
except FileNotFoundError:
    print("CSV file not found")

# 4. Define html routes
@app.route("/")
def index():
    return render_template("index.html")

# 5. App route for static map
@app.route("/Density")
def density_2022():
    # Convert DataFrame to JSON
    specific_columns = df[['Code','Country', 'Density (2022)']]
    animatedMapData = specific_columns.to_dict(orient='records')
    return jsonify(animatedMapData)
```

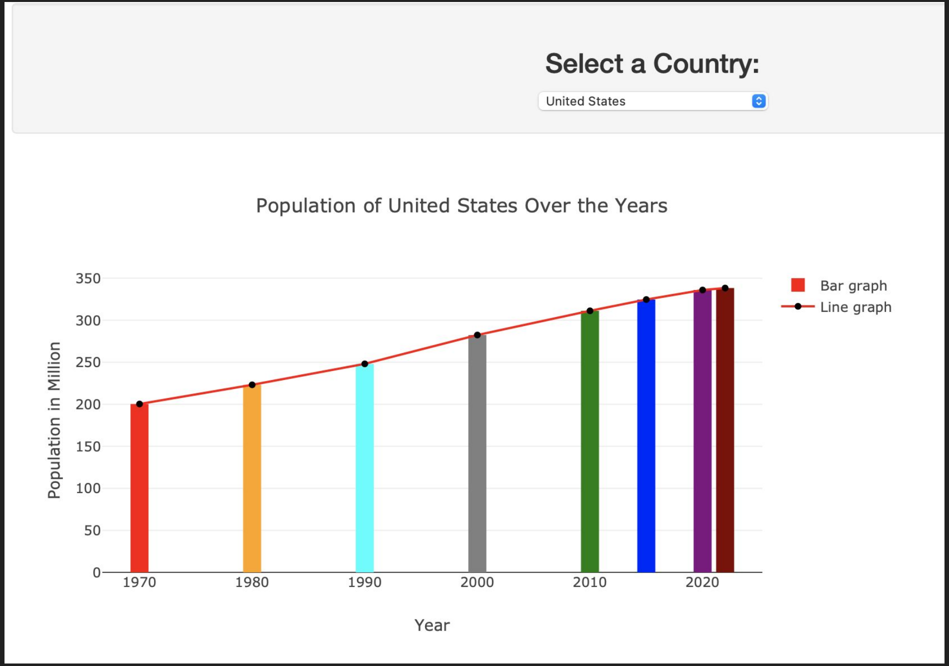
Our HTML



```html
<!DOCTYPE html>
<html lang="en">


<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<meta http-equiv="X-UA-Compatible" content="ie=edge">
<title>World Population Growth</title>
<link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.min.css">
<script src="https://cdn.plot.ly/plotly-latest.min.js"></script>
<script type="text/javascript" src="https://www.gstatic.com/charts/loader.js"></script>
</head>


<body>
    <div class="container">
    <div class="row">
    <div class="col-md-12 jumbotron text-center">
    <h1>World Population Growth</h1>
    <p>Explore the world population change over the years</p>
    </div>
    </div>
    </div>

    <div class="row">
        <div class="col-md-2">
        <div class="well">
        <h5>Select a Country:</h5>
        <select id="selDataset" onchange="optionChanged(this.value)"></select>
        </div>
        </div>
    </div>

    <div class="col-md-14">
        <div id="bar"></div>
    </div>
```

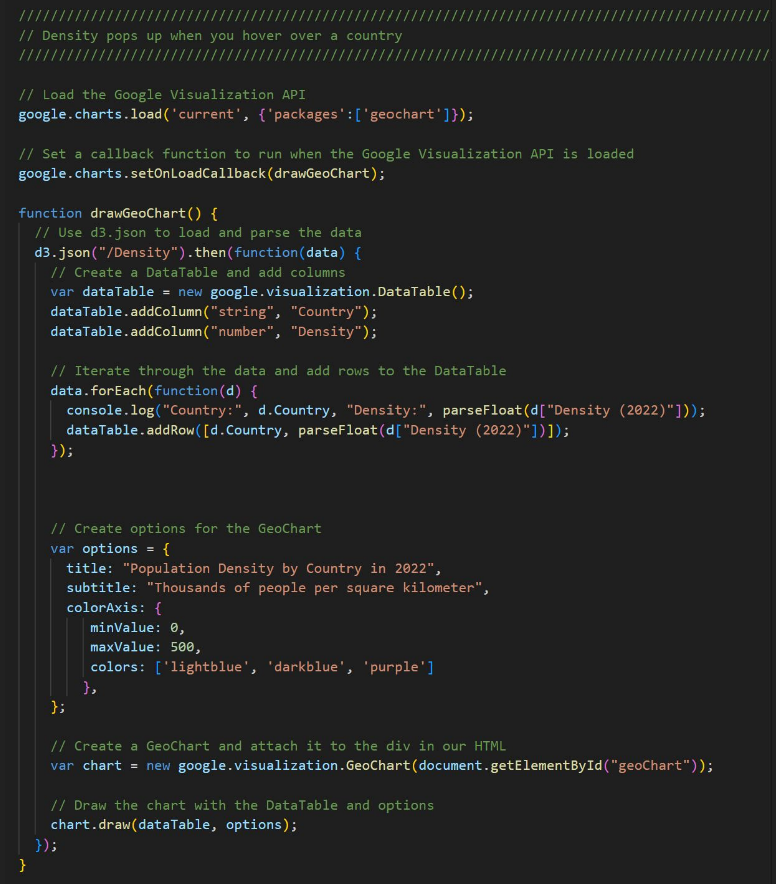


```
///////////////////////////////////////////////////////////////////////////////////////////
// Dropdown menu that changes by the country you select with Plotly
///////////////////////////////////////////////////////////////////////////////////////////

// Init function for the dropdown dashboard

function init() {
  // Fetch the data and console log it
  d3.json("/Country").then((data) => {
    // Extract unique country names for the dropdown
    const countryNames = Array.from(new Set(data.map(d => d.Country)));
    console.log(data);

    // Populate the dropdown with country names
    const dropdown = d3.select("#selDataset");
    d3.select("#selDataset").on("change", function() {
        const selectedCountry = d3.select(this).property("value");
        optionChanged(selectedCountry, data);
      });
    countryNames.forEach(country => {
      dropdown.append("option").text(country);
    });

    // Initialize the dashboard with the first country in the list
    optionChanged(countryNames[0], data);
  });
}

function optionChanged(selectedCountry, data) {
  // console.log("Selected Country: ", selectedCountry);

  // Filter data based on the selected country
  const countryData = data.filter(d => d.Country === selectedCountry);
  // console.log("Filtered Data: ", countryData);

  // Extract years and population data
  const years = Object.keys(countryData[0]).filter(key => /^\d{4}$/.test(key));
  const populationData = years.map(year => +countryData[0][year]);


  // Create the bar chart
  barPlot(years, populationData, selectedCountry);
}
```

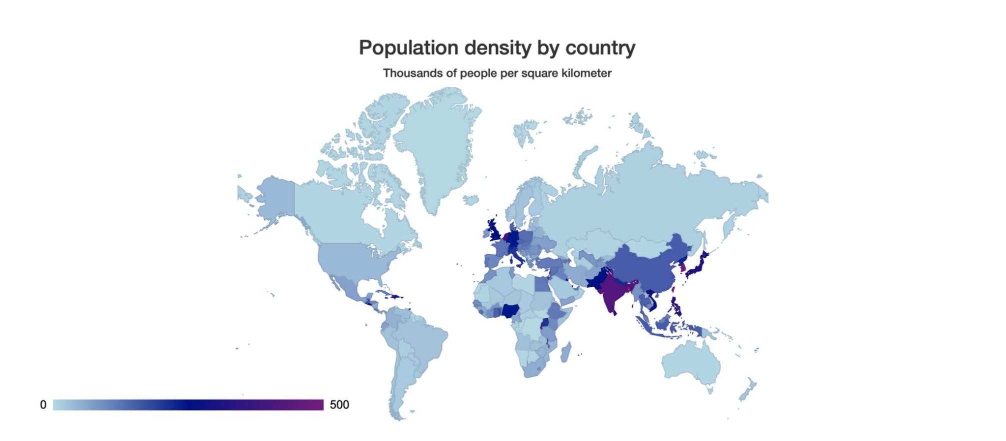


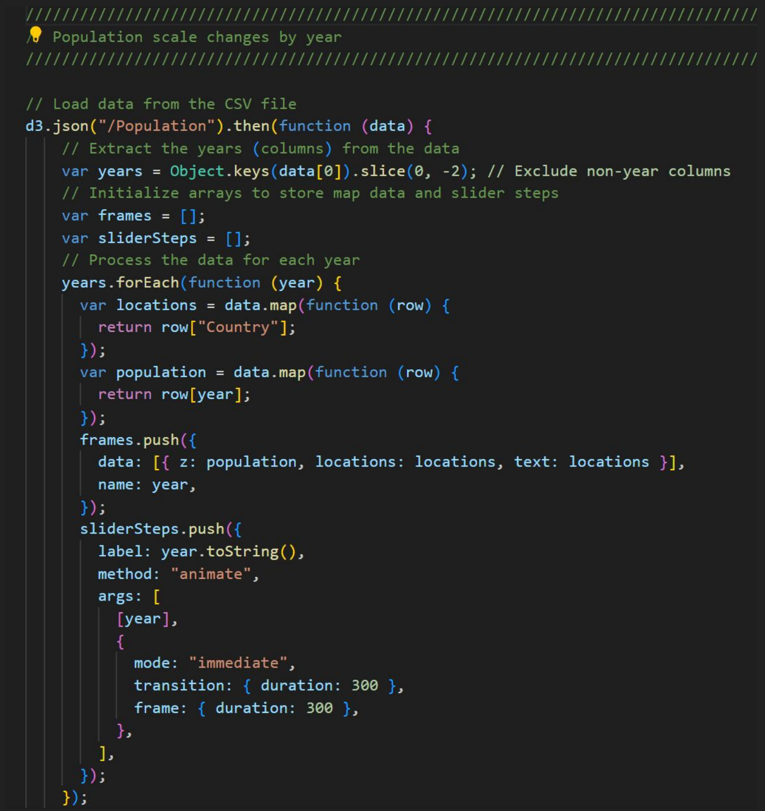
Select a Country:

United States

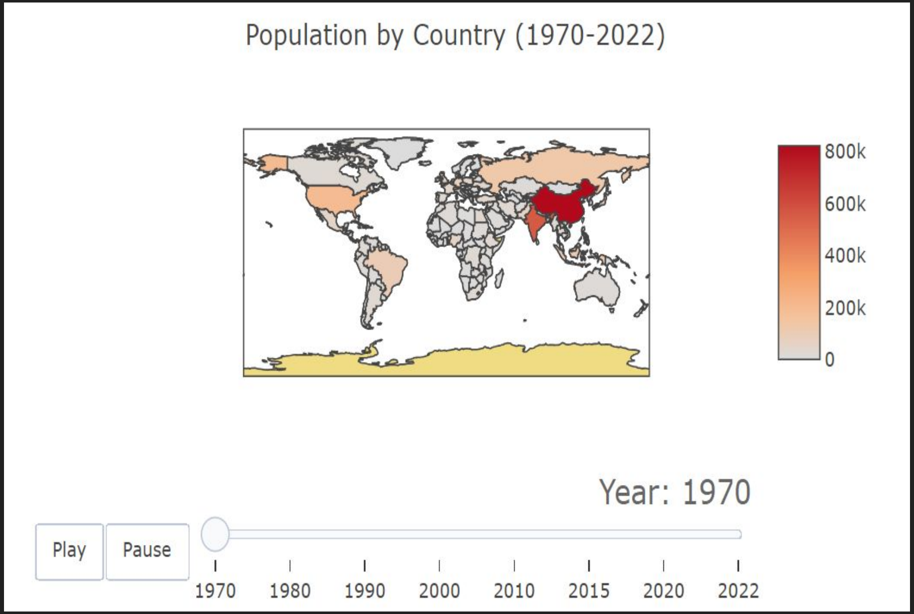


```
//////////////////////////////////////////////////////////////////////////////////////////////
// Density pops up when you hover over a country
//////////////////////////////////////////////////////////////////////////////////////////////

// Load the Google Visualization API
google.charts.load('current', {'packages':['geochart']});

// Set a callback function to run when the Google Visualization API is loaded
google.charts.setOnLoadCallback(drawGeoChart);

function drawGeoChart() {
  // Use d3.json to load and parse the data
  d3.json("/Density").then(function(data) {
    // Create a DataTable and add columns
    var dataTable = new google.visualization.DataTable();
    dataTable.addColumn("string", "Country");
    dataTable.addColumn("number", "Density");

    // Iterate through the data and add rows to the DataTable
    data.forEach(function(d) {
      console.log("Country:", d.Country, "Density:", parseFloat(d["Density (2022)"]));
      dataTable.addRow([d.Country, parseFloat(d["Density (2022)"])]);
    });



    // Create options for the GeoChart
    var options = {
      title: "Population Density by Country in 2022",
      subtitle: "Thousands of people per square kilometer",
      colorAxis: {
         minValue: 0,
         maxValue: 500,
         colors: ['lightblue', 'darkblue', 'purple']
        },
    };

    // Create a GeoChart and attach it to the div in our HTML
    var chart = new google.visualization.GeoChart(document.getElementById("geoChart"));

    // Draw the chart with the DataTable and options
    chart.draw(dataTable, options);
  });
}
```


Population density by country
Thousands of people per square kilometer
0
500



```
/////////////////////////////////////////////////////////////////////////////////
/💡 Population scale changes by year
/////////////////////////////////////////////////////////////////////////////////

// Load data from the CSV file
d3.json("/Population").then(function (data) {
    // Extract the years (columns) from the data
    var years = Object.keys(data[0]).slice(0, -2); // Exclude non-year columns
    // Initialize arrays to store map data and slider steps
    var frames = [];
    var sliderSteps = [];
    // Process the data for each year
    years.forEach(function (year) {
      var locations = data.map(function (row) {
        return row["Country"];
      });
      var population = data.map(function (row) {
        return row[year];
      });
      frames.push({
        data: [{ z: population, locations: locations, text: locations }],
        name: year,
      });
      sliderSteps.push({
        label: year.toString(),
        method: "animate",
        args: [
          [year],
          {
            mode: "immediate",
            transition: { duration: 300 },
            frame: { duration: 300 },
          },
        ],
      });
    });
```


Population by Country (1970-2022)
800k
600k
400k
200k
0
Year: 1970
Play
Pause
1970
1980
1990
2000
2010
2015
2020
2022