Project 3

Visualizing Suicide Rates and Economic Factors

Grace
Shelonta
Brett
Griff

Healthcare

Several of us are involved in the healthcare field.

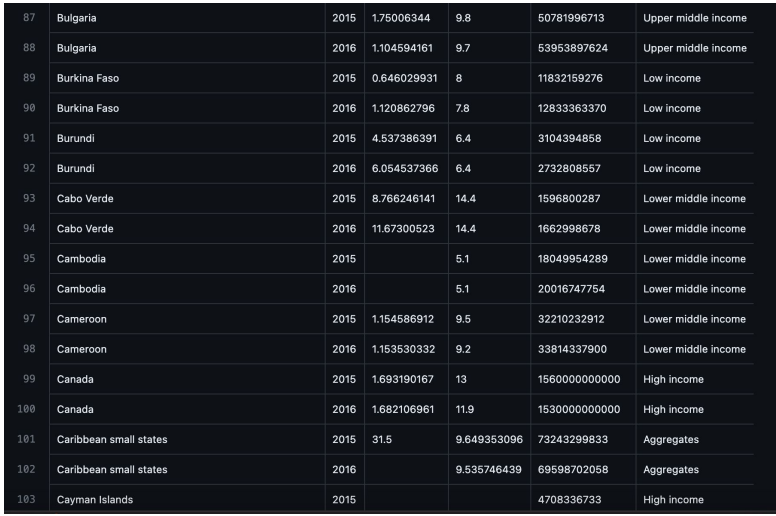
World Health Organization says it is important, among other things, to identify early, assess, manage, and follow up with folks that are affected by suicidal behavior.

To that end, we wanted to build some visualizations to see if we could identify any patterns in the data to guide resource management.

Datasets

Started on Kaggle and let the search results essentially lead us back to the source.

For our purposes, it ended up boiling down to a pair of csv files, one from the World Health Organization and one from World Bank Open Data (this was to add lat/lon to the countries of our initial dataset)

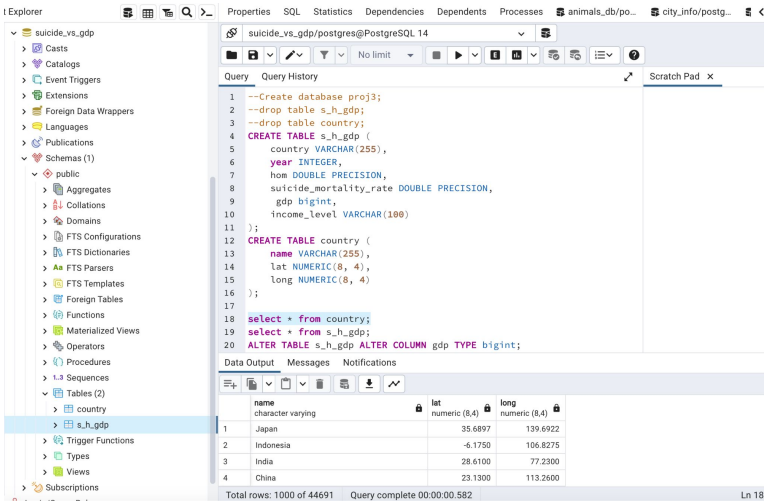


87	Bulgaria	2015	1.75006344	9.8	50781996713	Upper middle income
88	Bulgaria	2016	1.104594161	9.7	53953897624	Upper middle income
89	Burkina Faso	2015	0.646029931	8	11832159276	Low income
90	Burkina Faso	2016	1.120862796	7.8	12833363370	Low income
91	Burundi	2015	4.537386391	6.4	3104394858	Low income
92	Burundi	2016	6.054537366	6.4	2732808557	Low income
93	Cabo Verde	2015	8.766246141	14.4	1596800287	Lower middle income
94	Cabo Verde	2016	11.67300523	14.4	1662998678	Lower middle income
95	Cambodia	2015		5.1	18049954289	Lower middle income
96	Cambodia	2016		5.1	20016747754	Lower middle income
97	Cameroon	2015	1.154586912	9.5	32210232912	Lower middle income
98	Cameroon	2016	1.153530332	9.2	33814337900	Lower middle income
99	Canada	2015	1.693190167	13	1560000000000	High income
100	Canada	2016	1.682106961	11.9	1530000000000	High income
101	Caribbean small states	2015	31.5	9.649353096	73243299833	Aggregates
102	Caribbean small states	2016		9.535746439	69598702058	Aggregates
103	Cayman Islands	2015			4708336733	High income

Database

Used SQL

Imported relevant CSVs



Flask

```python
from flask import Flask, jsonify, render_template
from flask_cors import CORS
import psycopg2
from dotenv import load_dotenv
import os

load_dotenv()  # take environment variables from .env.

API_KEY = os.getenv("apiKey")

app = Flask(__name__)
CORS(app)

DB_HOST = os.getenv("DB_HOST")
DB_NAME = os.getenv("DB_NAME")
DB_USER = os.getenv("DB_USER")
DB_PASSWORD = os.getenv("DB_PASSWORD")


def connect_db():
    conn = psycopg2.connect(host=DB_HOST, database=DB_NAME, user=DB
    return conn


@app.route('/')
def welcome():
    return (
        f"Thank you for visiting our API on world suicide data! Ple
```

```python
@app.route('/data')
def data():
    conn = connect_db()
    cur = conn.cursor()
    query = "SELECT s.country, s.year, s.hom, s.suicide_mortality_r
    cur.execute(query)
    geojson_data = {
        "type": "FeatureCollection",
        "features": []
    }
    for row in cur.fetchall():

        country, year, hom, s_m_r, gdp, income, lat, long = row

        feature = {
            "type": "Feature",
            "properties": {
                "name": country,
                "Year": year,
                "homicide_rate": hom,
                "suicide_rate": s_m_r,
                "GDP": gdp,
                "income_level": income
            },
            "geometry": {
                "type": "Point",
                "coordinates": [long, lat]
```

```python
            },
            "geometry": {
                "type": "Point",
                "coordinates": [long, lat]
            }

        }
        geojson_data["features"].append(feature)

    cur.close()
    conn.close()
    return jsonify(geojson_data)


if __name__ == '__main__':
    app.run(debug=True,port=5000)

```

Combined dataset


```
{
  "features": [
    {
      "geometry": {
        "coordinates": [
          "139.6922",
          "35.6897"
        ],
        "type": "Point"
      },
      "properties": {
        "GDP": 5000000000000,
        "Year": 2016,
        "homicide_rate": 0.283336524,
        "income_level": "High income",
        "name": "Japan",
        "suicide_rate": 17.5
      },
      "type": "Feature"
    },
    {
      "geometry": {
        "coordinates": [
          "139.6922",
          "35.6897"
        ],
        "type": "Point"
      },
      "properties": {
        "GDP": 4440000000000,
        "Year": 2015,
        "homicide_rate": 0.28362668,
        "income_level": "High income",
        "name": "Japan",
        "suicide_rate": 19.1
      },
      "type": "Feature"
    },
    {
```

Visualizations

Now that we had a foundation to work off of, we started trying to build different visualizations.

Does a country with high GDP have lower rates of suicide? Higher? We tried designing several kinds of visualizations to explore that question.

Heatmap to see at a glance what countries/areas have the highest and lowest rates

Cluster map using ESRI mapping

Country color varying by rates (dropped)

Plot/bar graph to map suicide rates vs homicide rates and also gdp.

Basket Weaving

Once we had 3 working visualizations, it was time to bring them together.

All the relevant js files were put in templates, and routes were added to app.py

```
42
43  @app.route('/ShelontaClusterMap')
44  def index1():
45      return render_template('shelonta_cluster.html', API_KEY=API_KEY
46
47  @app.route('/BarGraph')
48  def index2():
49      return render_template('BrettBarGraphs.html')
50
51  @app.route('/HeatMap')
52  def index3():
53      return render_template('heatmap.html')
54
55  @app.route('/data')
56  def data():
57      conn = connect_db()
58      cur = conn.cursor()
59      query = "SELECT s.country, s.year, s.hom, s.suicide_mortality_r
60      cur.execute(query)
61      geojson_data = {
62          "type": "FeatureCollection",
63          "features": []
64      }
65      for row in cur.fetchall():
66
```

Finally on the index page we added a directory for the end user accessing these charts/maps.

```
24
25  @app.route('/')
26  def welcome():
27      return (
28          f"Thank you for visiting our API on world suicide data! Ple
29          f"<br/>"
30          f"<br/>"
31          f"Available Routes:<br/>"
32          f"<br/>"
33          f"<br/>"
34          f"Cluster Map comparing average suicide rates and GDP: <a h
35          f"<br/>"
36          f"Bar Chart comparing average suicide and homicide rate bas
37          f"<br/>"
38          f"Heatmap showing the suicide rate per country: <a href='/H
39          f"<br/>"
40          f"General data: <a href='/data'>data</a><br/>"
41      )
42
43  @app.route('/ShelontaClusterMap')
44  def index1():
45      return render_template('shelonta_cluster.html', API_KEY=API_KEY
46
```

Limitations/Challenges

Dataset is truncated.

Even with truncated data, some countries did not provide data so the dataset is extra incomplete.

Our dataset does not perfectly gel with our intentions with the maps.

Dataset could use more detailed info for country boundaries

Check out our page!

And thanks for watching!