World Happiness Report

Project Team 3: Ksenia Dyakova, Patrick Saitta, Max Izotov, Eddie Xu, Mike Dunlap

Summary & Data

Purpose:
1.To visualize the state of global happiness based on publicly-available data.
2. Review the state of happiness in individual countries and show how the new science of happiness explains personal and national variations in happiness

Resources:
Kaggle
https://www.kaggle.com/unsdsn/world-happiness

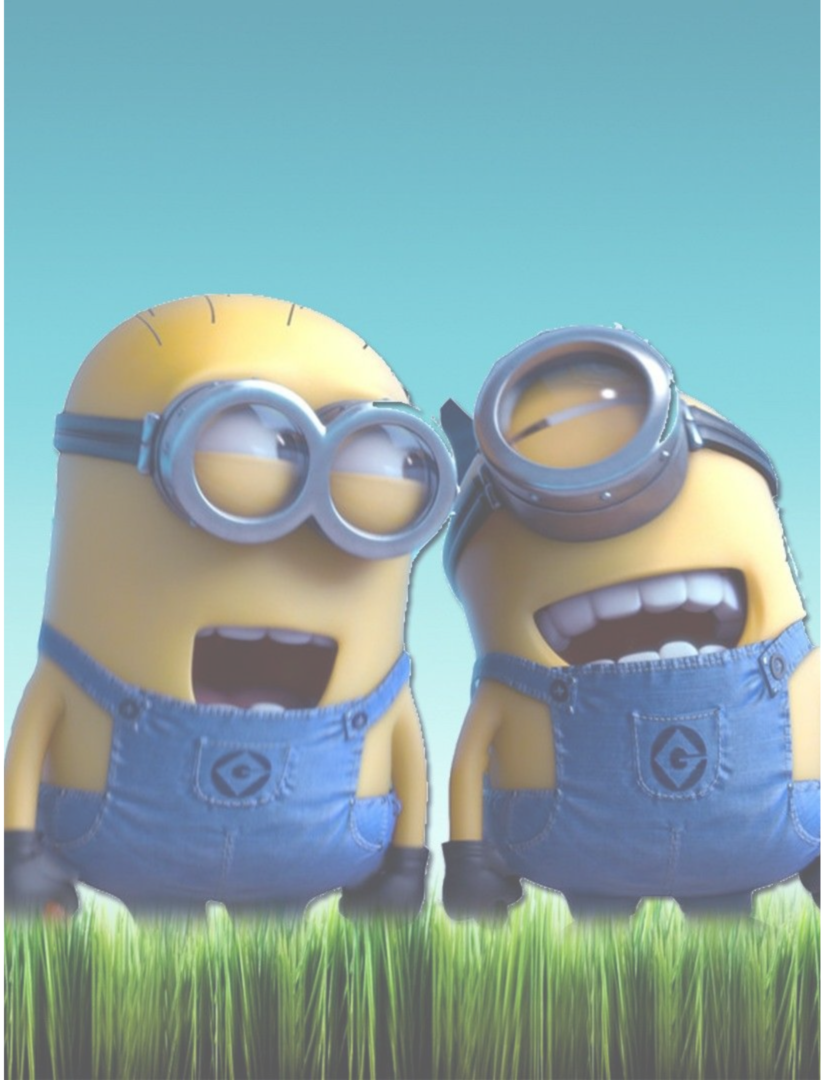
Google Data Search
https://datasetsearch.research.google.com/

The World Happiness Report
https://worldhappiness.report/ed/2020/

DataMaps
http://datamaps.github.io/

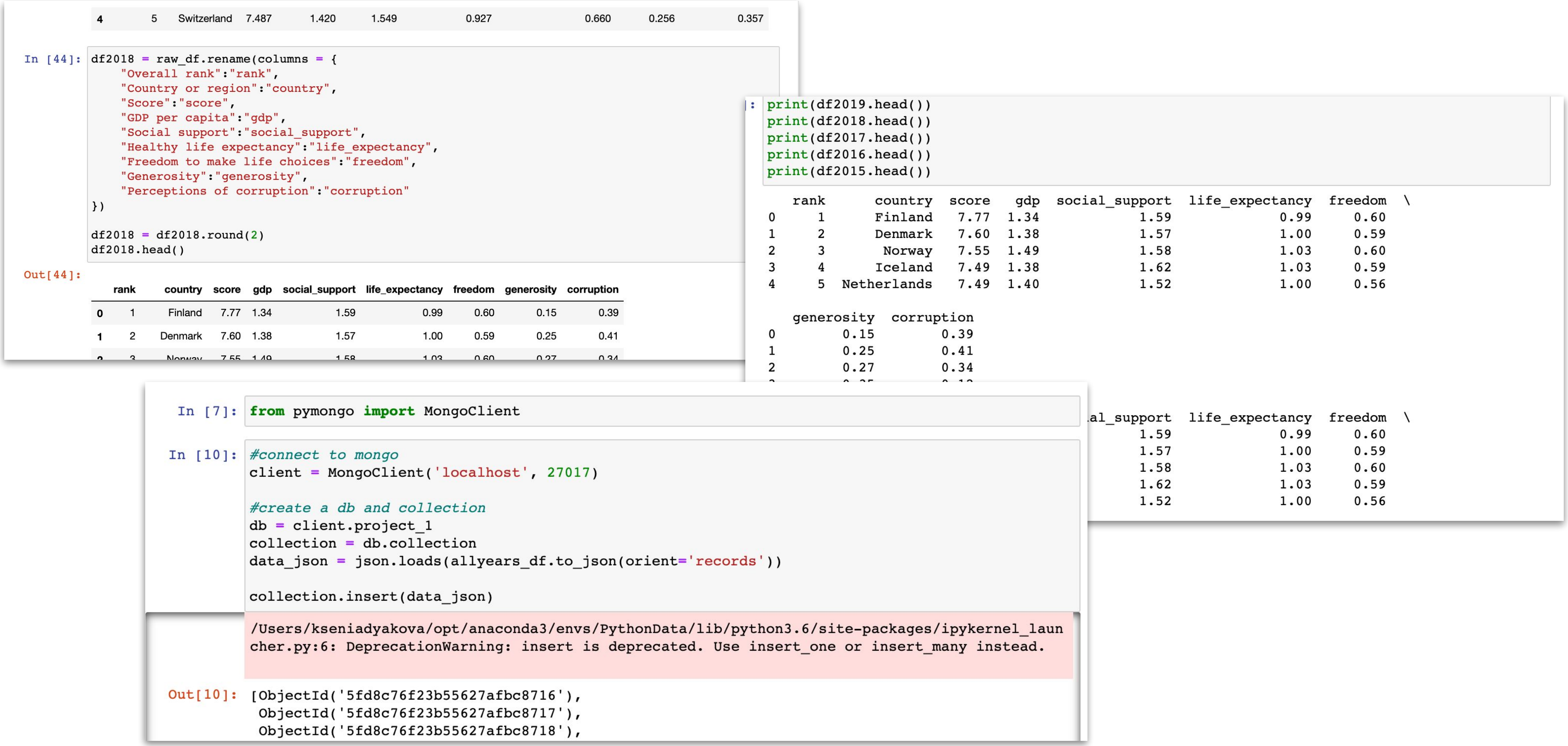
GITHUB PROJECT LINK: https://github.com/eddiexunyc/Project2_Team3

Data Cleanup
&
Exploration



Data Cleanup & Exploration

Cleaned up 5 CSV files and made the columns consistent and ready for export to MongoDB



4	5	Switzerland	7.487	1.420	1.549	0.927	0.660	0.256	0.357

```
In [44]: df2018 = raw_df.rename(columns = {
             "Overall rank":"rank",
             "Country or region":"country",
             "Score":"score",
             "GDP per capita":"gdp",
             "Social support":"social_support",
             "Healthy life expectancy":"life_expectancy",
             "Freedom to make life choices":"freedom",
             "Generosity":"generosity",
             "Perceptions of corruption":"corruption"
         })

         df2018 = df2018.round(2)
         df2018.head()
```

Out[44]:

	rank	country	score	gdp	social_support	life_expectancy	freedom	generosity	corruption
0	1	Finland	7.77	1.34	1.59	0.99	0.60	0.15	0.39
1	2	Denmark	7.60	1.38	1.57	1.00	0.59	0.25	0.41
2	3	Norway	7.55	1.49	1.58	1.03	0.60	0.27	0.34

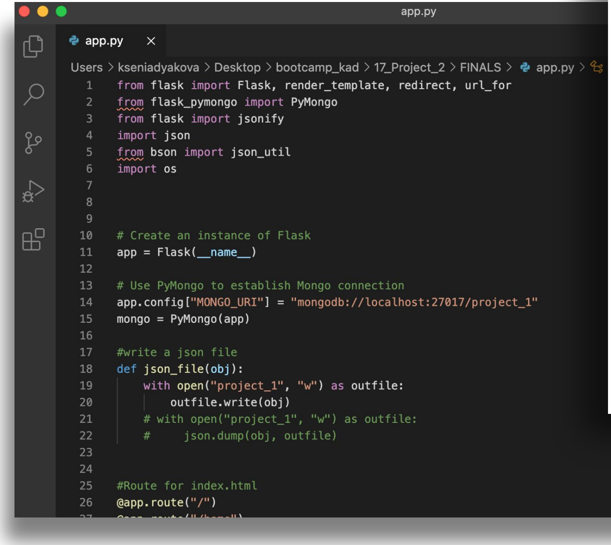
```
]: print(df2019.head())
   print(df2018.head())
   print(df2017.head())
   print(df2016.head())
   print(df2015.head())
```

```
   rank      country  score   gdp  social_support  life_expectancy  freedom  \
0     1      Finland   7.77  1.34            1.59             0.99     0.60
1     2      Denmark   7.60  1.38            1.57             1.00     0.59
2     3       Norway   7.55  1.49            1.58             1.03     0.60
3     4      Iceland   7.49  1.38            1.62             1.03     0.59
4     5  Netherlands   7.49  1.40            1.52             1.00     0.56

   generosity  corruption
0        0.15        0.39
1        0.25        0.41
2        0.27        0.34
```

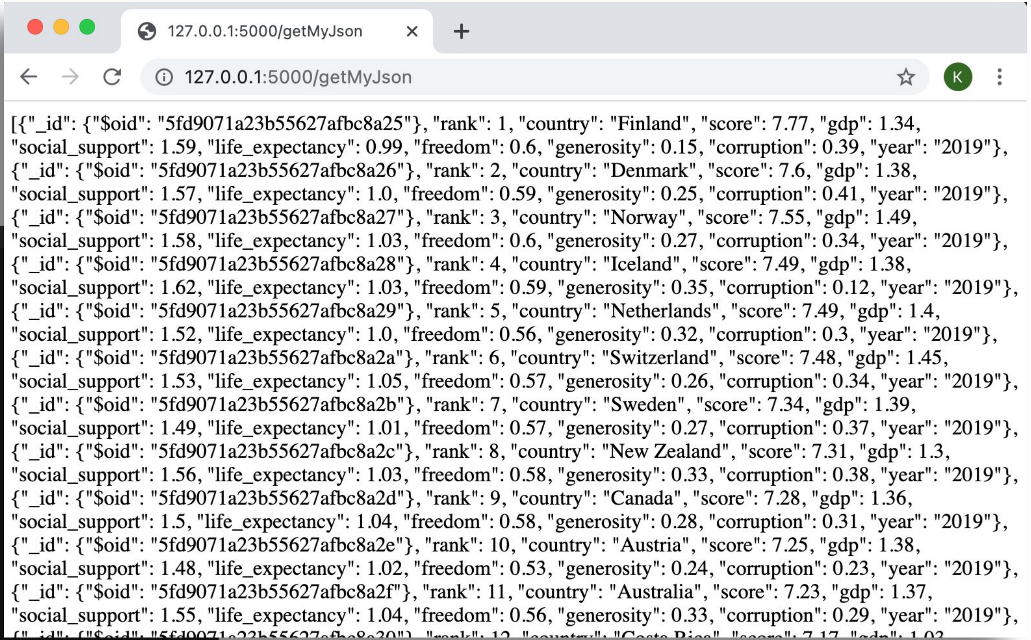
```
al_support  life_expectancy  freedom  \
      1.59             0.99     0.60
      1.57             1.00     0.59
      1.58             1.03     0.60
      1.62             1.03     0.59
      1.52             1.00     0.56
```

```
In [7]: from pymongo import MongoClient

In [10]: #connect to mongo
         client = MongoClient('localhost', 27017)

         #create a db and collection
         db = client.project_1
         collection = db.collection
         data_json = json.loads(allyears_df.to_json(orient='records'))

         collection.insert(data_json)

         /Users/kseniadyakova/opt/anaconda3/envs/PythonData/lib/python3.6/site-packages/ipykernel_laun
         cher.py:6: DeprecationWarning: insert is deprecated. Use insert_one or insert_many instead.

Out[10]: [ObjectId('5fd8c76f23b55627afbc8716'),
          ObjectId('5fd8c76f23b55627afbc8717'),
          ObjectId('5fd8c76f23b55627afbc8718'),
```

Opening the Flask


app.py
app.py
Users > kseniadyakova > Desktop > bootcamp_kad > 17_Project_2 > FINALS > app.py >

```python
from flask import Flask, render_template, redirect, url_for
from flask_pymongo import PyMongo
from flask import jsonify
import json
from bson import json_util
import os



# Create an instance of Flask
app = Flask(__name__)

# Use PyMongo to establish Mongo connection
app.config["MONGO_URI"] = "mongodb://localhost:27017/project_1"
mongo = PyMongo(app)

#write a json file
def json_file(obj):
    with open("project_1", "w") as outfile:
        outfile.write(obj)
    # with open("project_1", "w") as outfile:
    #     json.dump(obj, outfile)


#Route for index.html
@app.route("/")
@app.route("/home")
```

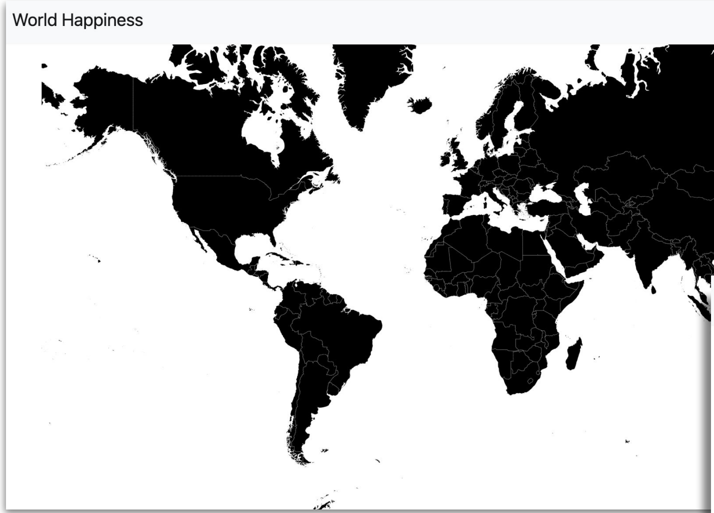


[{"_id": {"$oid": "5fd9071a23b55627afbc8a25"}, "rank": 1, "country": "Finland", "score": 7.77, "gdp": 1.34, "social_support": 1.59, "life_expectancy": 0.99, "freedom": 0.6, "generosity": 0.15, "corruption": 0.39, "year": "2019"}, {"_id": {"$oid": "5fd9071a23b55627afbc8a26"}, "rank": 2, "country": "Denmark", "score": 7.6, "gdp": 1.38, "social_support": 1.57, "life_expectancy": 1.0, "freedom": 0.59, "generosity": 0.25, "corruption": 0.41, "year": "2019"}, {"_id": {"$oid": "5fd9071a23b55627afbc8a27"}, "rank": 3, "country": "Norway", "score": 7.55, "gdp": 1.49, "social_support": 1.58, "life_expectancy": 1.03, "freedom": 0.6, "generosity": 0.27, "corruption": 0.34, "year": "2019"}, {"_id": {"$oid": "5fd9071a23b55627afbc8a28"}, "rank": 4, "country": "Iceland", "score": 7.49, "gdp": 1.38, "social_support": 1.62, "life_expectancy": 1.03, "freedom": 0.59, "generosity": 0.35, "corruption": 0.12, "year": "2019"}, {"_id": {"$oid": "5fd9071a23b55627afbc8a29"}, "rank": 5, "country": "Netherlands", "score": 7.49, "gdp": 1.4, "social_support": 1.52, "life_expectancy": 1.0, "freedom": 0.56, "generosity": 0.32, "corruption": 0.3, "year": "2019"}, {"_id": {"$oid": "5fd9071a23b55627afbc8a2a"}, "rank": 6, "country": "Switzerland", "score": 7.48, "gdp": 1.45, "social_support": 1.53, "life_expectancy": 1.05, "freedom": 0.57, "generosity": 0.26, "corruption": 0.34, "year": "2019"}, {"_id": {"$oid": "5fd9071a23b55627afbc8a2b"}, "rank": 7, "country": "Sweden", "score": 7.34, "gdp": 1.39, "social_support": 1.49, "life_expectancy": 1.01, "freedom": 0.57, "generosity": 0.27, "corruption": 0.37, "year": "2019"}, {"_id": {"$oid": "5fd9071a23b55627afbc8a2c"}, "rank": 8, "country": "New Zealand", "score": 7.31, "gdp": 1.3, "social_support": 1.56, "life_expectancy": 1.03, "freedom": 0.58, "generosity": 0.33, "corruption": 0.38, "year": "2019"}, {"_id": {"$oid": "5fd9071a23b55627afbc8a2d"}, "rank": 9, "country": "Canada", "score": 7.28, "gdp": 1.36, "social_support": 1.5, "life_expectancy": 1.04, "freedom": 0.58, "generosity": 0.28, "corruption": 0.31, "year": "2019"}, {"_id": {"$oid": "5fd9071a23b55627afbc8a2e"}, "rank": 10, "country": "Austria", "score": 7.25, "gdp": 1.38, "social_support": 1.48, "life_expectancy": 1.02, "freedom": 0.53, "generosity": 0.24, "corruption": 0.23, "year": "2019"}, {"_id": {"$oid": "5fd9071a23b55627afbc8a2f"}, "rank": 11, "country": "Australia", "score": 7.23, "gdp": 1.37, "social_support": 1.55, "life_expectancy": 1.04, "freedom": 0.56, "generosity": 0.33, "corruption": 0.29, "year": "2019"}, {"_id": {"$oid": "5fd9071a23b55627afbc8a30"}, "rank": 12, "country": "Costa Rica", "score": 7.17, "gdp": 1.03

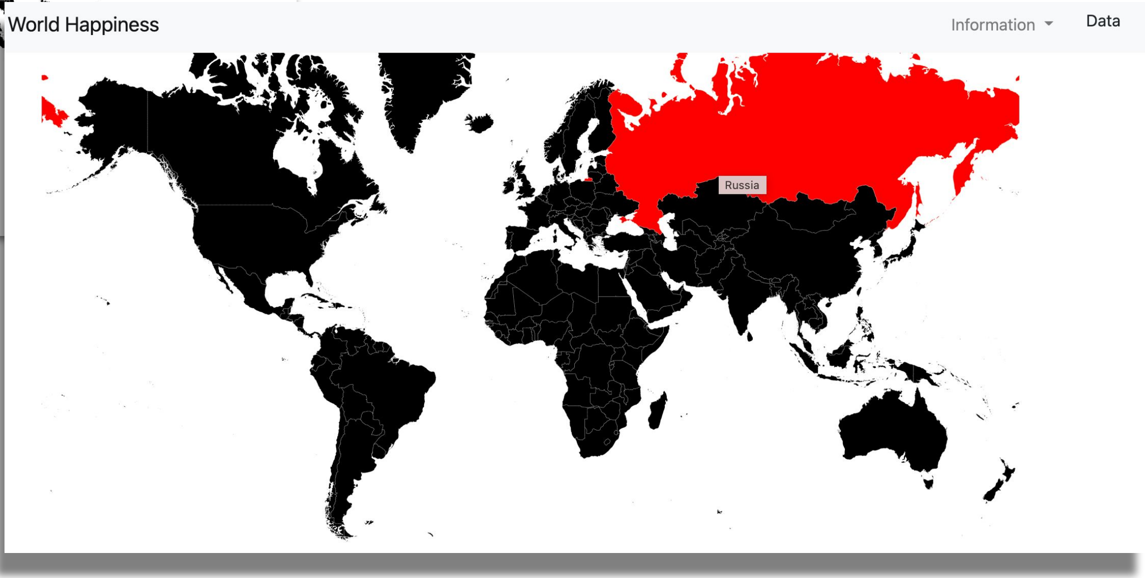
Data Analysis...

WWW.LEON-GAO.COM ALL RIGHTS RESERVED

World Map



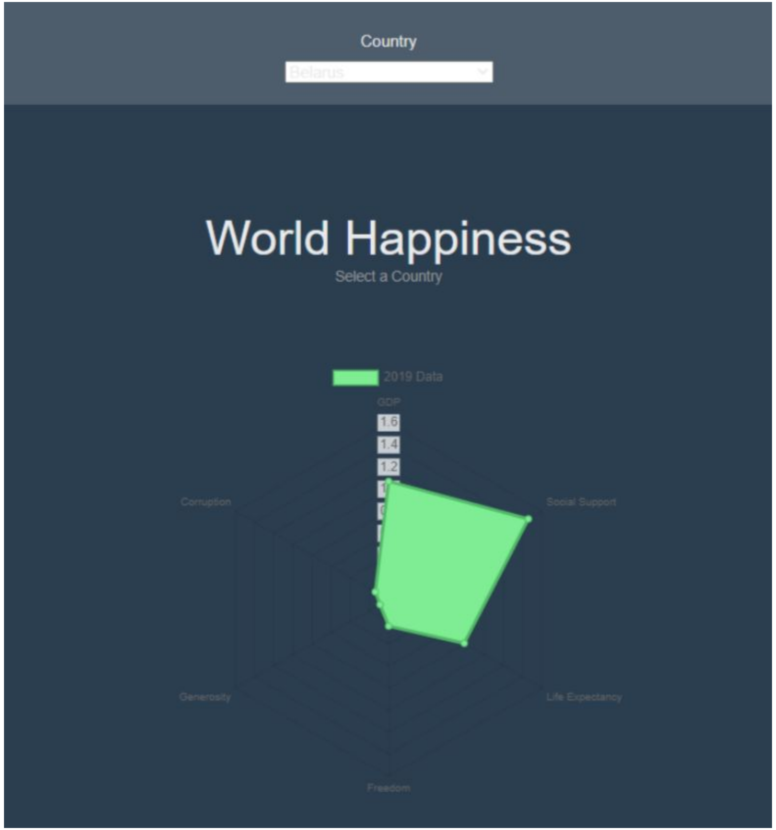
- ➔ Using World Atlas TopoJson file from NPM repository, that provides a convenient redistribution of Natural Earth’s vector data, we were able to put together a map of the world and identify all the countries that were within our World Happiness dataset.



- ➔ By using SVG elements and some CSS “tricks”, we were able to create a “highlight” effect when hovering over the countries and display their names.

Radar Chart

- Interactive data is based on Year 2019
- 255+ Countries to choose from
- 6 Data points:
  - GDP
  - Social Support
  - Life Expectancy
  - Freedom
  - Generosity
  - Corruption

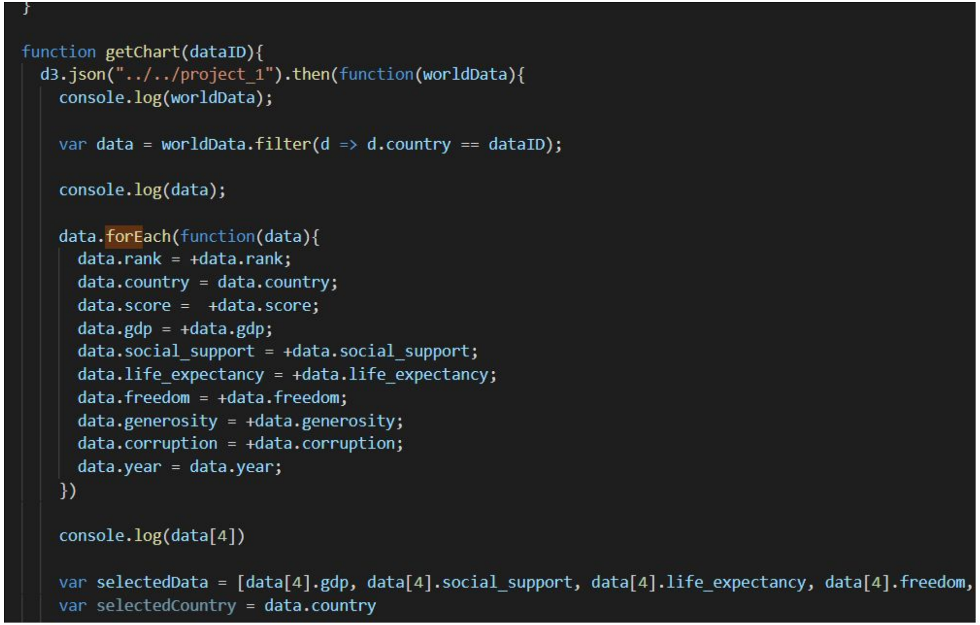


Country: Belarus

World Happiness

Select a Country

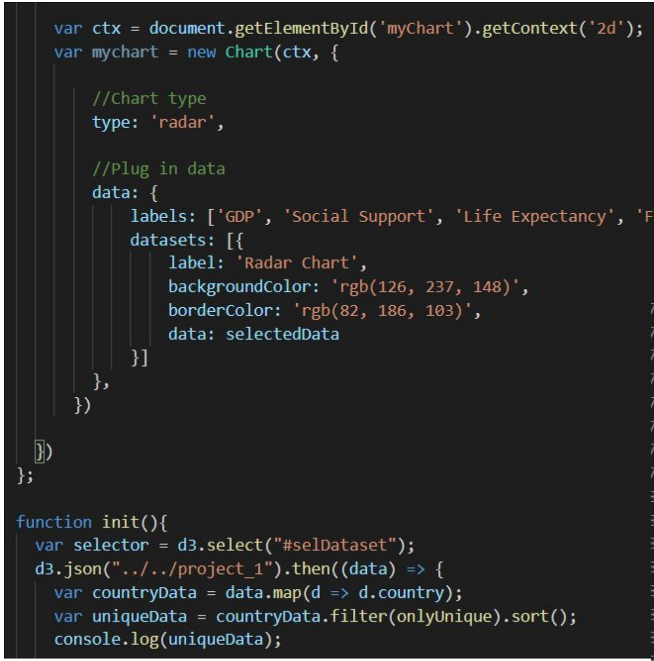
Radar Chart



```
}

function getChart(dataID){
  d3.json("../../project_1").then(function(worldData){
    console.log(worldData);

    var data = worldData.filter(d => d.country == dataID);

    console.log(data);

    data.forEach(function(data){
      data.rank = +data.rank;
      data.country = data.country;
      data.score =  +data.score;
      data.gdp = +data.gdp;
      data.social_support = +data.social_support;
      data.life_expectancy = +data.life_expectancy;
      data.freedom = +data.freedom;
      data.generosity = +data.generosity;
      data.corruption = +data.corruption;
      data.year = data.year;
    })

    console.log(data[4])

    var selectedData = [data[4].gdp, data[4].social_support, data[4].life_expectancy, data[4].freedom,
    var selectedCountry = data.country
```

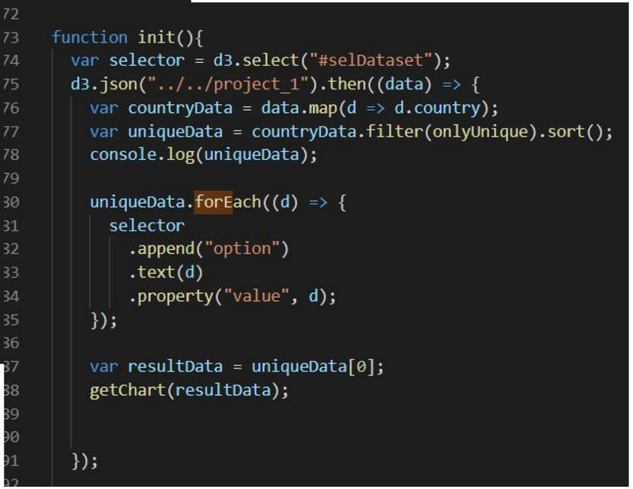
getChart gets called, which stores various data for the country selected, including the six sub-scores we mapped out on the radar chart

Radar Chart



```
    var ctx = document.getElementById('myChart').getContext('2d');
    var mychart = new Chart(ctx, {

        //Chart type
        type: 'radar',

        //Plug in data
        data: {
            labels: ['GDP', 'Social Support', 'Life Expectancy', 'Freedom', 'Generosit
            datasets: [{
                label: 'Radar Chart',
                backgroundColor: 'rgb(126, 237, 148)',
                borderColor: 'rgb(82, 186, 103)',
                data: selectedData
            }]
        },
      })

  })
};

function init(){
  var selector = d3.select("#selDataset");
  d3.json("../../project_1").then((data) => {
    var countryData = data.map(d => d.country);
    var uniqueData = countryData.filter(onlyUnique).sort();
    console.log(uniqueData);
```

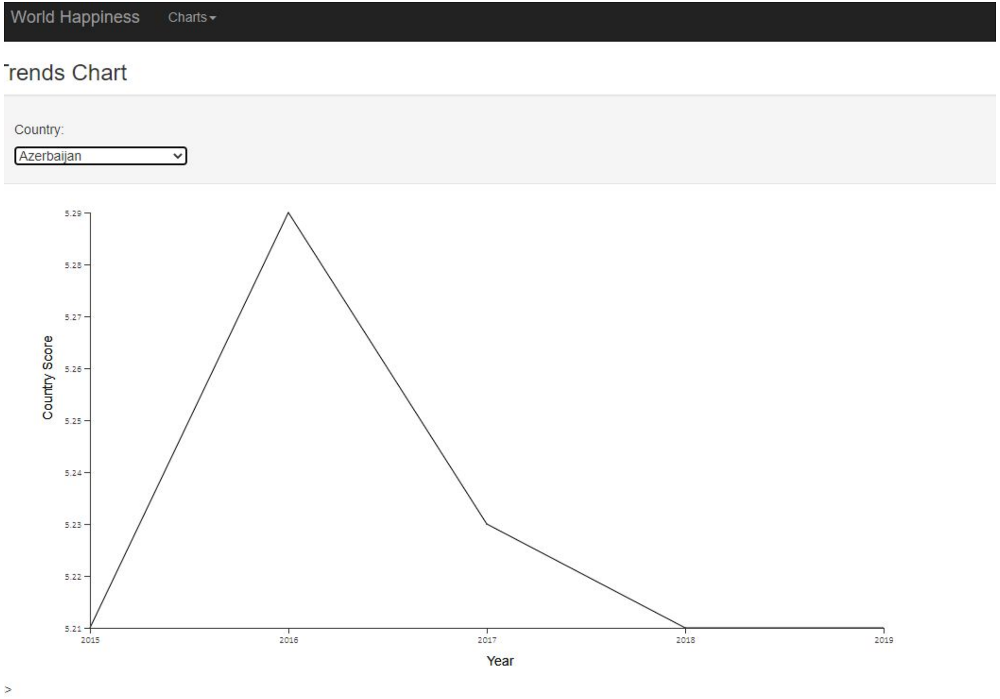
The sub-score data is saved in selectedData, which is then referenced in 'data', allowing a radar chart with that country's data to be contructed when the getChart function is called



```
function init(){
  var selector = d3.select("#selDataset");
  d3.json("../../project_1").then((data) => {
    var countryData = data.map(d => d.country);
    var uniqueData = countryData.filter(onlyUnique).sort();
    console.log(uniqueData);

    uniqueData.forEach((d) => {
      selector
        .append("option")
        .text(d)
        .property("value", d);
    });

    var resultData = uniqueData[0];
    getChart(resultData);


  });
```

Line Chart

- The data is based on countries’ score and how are they perform from 2015 to 2019.
- Using the scroll down menu, we can see each country and their score.



Trends Chart

Country: Azerbaijan

Next Steps

- Next steps to improve the site:
  - Update the world map with pop-ups showing scores and color-shading by score
  - Add dropdown year options to the radar chart
  - Improve line chart functionality (e.g. add multiple countries)
  - Create a data table for the country data
  - Enhance the UI
  - Deploy to Heroku