How happy are the people in each country?

"Being happy never goes out of style". - Lilly Pulitzer

Team Presentation

Shailza Rattu

Susana Villagrana

Jennifer Rocha

Objective

How happy are the people in each country and

Are there any correlations to be made with suicide rates

and overall life expectancy?

Process

Extract, Transform, Load (ETL)

cleaned up the data to export to MongoDB

Created routes to use data in app.js.....basically creating an API from local server

Used flask to import data from mongoDB

Plotly
Link

Title animation

Link

Used D3 library
Link

Used CSS and Bootstrap
Link

HTML for dash structure

Process in Details

Key Details

Key Partners

- Shailza Rattu
- Susana Villagrana
- Jennifer Rocha

Data

CSV files from Kaggle:

- 2015.csv "Happiness by Country for 2015"
- 2016.csv "Happiness by Country for 2016"
- master.csv "Suicide Rates by year and country from 1985 to 2016"

Data ETL

Extracting:
- Extracted from Kaggle data sets.

Transform:
- These files did not have columns with the year so that was added.
- Concat was used to stack the two dataframes.
- Renamed some columns for ease of use in the database.
- Added country code for use with visualizations

Loading:
- Once the data was cleaned up pandas was used to turn the dataframe into a dictionary. Pymongo helped to load the data into a "happy_db" MongoDB DataBase with collection names of "happy" and "suicide."

Flask

- Used to connect database from MongoDb to feed to our JavaScript for plotting.
- Created route that included both collections within the database.
- Returned the results jsonified for use in JS.
-Removed nulls

HTML/CSS

- Used to display the dashboard of interactive visuals.
- Drop down menu operates all visuals.
- Bootstrap file added to style the page
- Anime script added for to animate the page header.
- Scrolling function added to table.

JavaScript

- Used to display the dashboard of interactive visuals.
- Drop down menu operates all visuals.
- Filters to sort data by year.
- Plotly for plotting multiple traces to different table, bubble, bar, and map plots.
- Anime.js to animate the header.
- optionChanged, aggregation and unpack functions used on the data for plotting.

Tools and Languages

- Jupyter Notebook
- Flask
- MongoDB
- JavaScript
- Plotly
-HTML
- CSS/Bootstrap
- Anime.js to animate the
- GitHub

Additional Analysis Considerations

- Relationship between suicide rates and life expectancy.
- Additional years available to see if there are more trends for the data pieces over a longer period of time.
- Connection between population and either suicide rate, life expectancy, or overall happiness.
- Do the other factors like Freedom, Generosity, or Government Corruption impact any of the rates we have been evaluating?

Key Partners

- Shailza Rattu
- Susana Villagrana
- Jennifer Rocha

Data

CSV files from Kaggl

- 2015.csv "Happines for 2015"
- 2016.csv "Happines for 2016"
- master.csv "Suicide and country from 198

Additional Analysi

- Relationship betwe
- Additional years ava
- Connection betwee
- Do the other factors

Project-3-happiness-b

```
from flask
from flask

app = Flas

mongo = Py

@app.route
def db_pi
    return

@app.route
def db_dat
    db_dat
    ss_dat
    data =
    print(
    return

if __name_
    app.ru
```

```
var ss_filtered = myData.suicide.filter(x => x.year == '2015');

var suicide = [];

var tmp_sNo = 0;
var tmp_pop = 0;
var sPer = 0;
for (var i = 0; i < ss_filtered.length; i++) {

    if (i == (ss_filtered.length-1)) {
        tmp_sNo = tmp_sNo + ss_filtered[i].suicides_no;
        tmp_pop = tmp_pop + ss_filtered[i].population;
        sPer = (tmp_sNo / tmp_pop)* 100000;
        suicide.push({"country": ss_filtered[i].country,
            "year": ss_filtered[i].year,
            "suicideNo": tmp_sNo,
            "population": tmp_pop,
            "perOne": sPer})
        tmp_sNo = 0;
        tmp_pop = 0;}
    else if (ss_filtered[i].country==ss_filtered[i+1].country && ss_filtered[i].year==ss_filtered[i+1].year) {
        tmp_sNo = tmp_sNo + ss_filtered[i].suicides_no;
        tmp_pop = tmp_pop + ss_filtered[i].population;
    } else {
        tmp_sNo = tmp_sNo + ss_filtered[i].suicides_no;
        tmp_pop = tmp_pop + ss_filtered[i].population;
        sPer = (tmp_sNo / tmp_pop) * 100000;
        suicide.push({"country": ss_filtered[i].country,
            "year": ss_filtered[i].year,
            "suicideNo": tmp_sNo,
            "population": tmp_pop,
            "perOne": sPer})
        tmp_sNo = 0;
        tmp_pop = 0;
    }
}
var happy_filtered = myData.happiness.filter(x => x.year == '2015');

var scatterData = [];

var tmp_sNo = 0;
var tmp_pop = 0;
var sPer = 0;
for (var z = 0; z < suicide.length; z++) {
    for (var t = 0; t < happy_filtered.length; t++) {
        if (suicide[z].country==happy_filtered[t].Country && suicide[z].year==happy_filtered[t].year) {
            scatterData.push({"country": suicide[z].country,
                "year": suicide[z].year,
                "suicideNo": suicide[z].suicideNo,
                "population": suicide[z].population,
                "perOne": suicide[z].perOne,
                "happiness_score": happy_filtered[t].happiness_score})
        }
    }
}
////////// Bubble Chart //////////
```

```
ror': False})
ear': False})
x in ss_data]}
```

nguages

book

p

nimate the

Key Details

Key Partners

- Shailza Rattu
- Susana Villagrana
- Jennifer Rocha

Data

CSV files from Kaggle:

- 2015.csv "Happiness by Country for 2015"
- 2016.csv "Happiness by Country for 2016"
- master.csv "Suicide Rates by year and country from 1985 to 2016"

Data ETL

Extracting:
- Extracted from Kaggle data sets.

Transform:
- These files did not have columns with the year so that was added.
- Concat was used to stack the two dataframes.
- Renamed some columns for ease of use in the database.
- Added country code for use with visualizations

Loading:
- Once the data was cleaned up pandas was used to turn the dataframe into a dictionary. Pymongo helped to load the data into a "happy_db" MongoDB DataBase with collection names of "happy" and "suicide."

Flask

- Used to connect database from MongoDb to feed to our JavaScript for plotting.
- Created route that included both collections within the database.
- Returned the results jsonified for use in JS.
-Removed nulls

HTML/CSS

- Used to display the dashboard of interactive visuals.
- Drop down menu operates all visuals.
- Bootstrap file added to style the page
- Anime script added for to animate the page header.
- Scrolling function added to table.

JavaScript

- Used to display the dashboard of interactive visuals.
- Drop down menu operates all visuals.
- Filters to sort data by year.
- Plotly for plotting multiple traces to different table, bubble, bar, and map plots.
- Anime.js to animate the header.
- optionChanged, aggregation and unpack functions used on the data for plotting.

Tools and Languages

- Jupyter Notebook
- Flask
- MongoDB
- JavaScript
- Plotly
-HTML
- CSS/Bootstrap
- Anime.js to animate the
- GitHub

Additional Analysis Considerations

- Relationship between suicide rates and life expectancy.
- Additional years available to see if there are more trends for the data pieces over a longer period of time.
- Connection between population and either suicide rate, life expectancy, or overall happiness.
- Do the other factors like Freedom, Generosity, or Government Corruption impact any of the rates we have been evaluating?

A few questions we had of the data:

- What country ranks the highest for the happiest people?
- Are there any large variations between the years?
- Can you see any correlation between suicide rates and countries happiness score?
- Is there a connection between a country's happiness score and their life expectancy rates?

Maps


CANADA
GREENLAND (DENMARK)
ICELAND
ALASKA (USA)
Hudson Bay
Labrador Sea


Iceland
Reykjavik
Akureyri
Keflavik
Selfoss
Grindavik
Thorlakshofn
Eyrarbakki
Vik
Borgarnes
Grundarfjordhur
Olafsvik
Stykkisholmur
Holmavik
Blonduos
Hvammstangi
Saudharkrokur
Skagastrond
Siglufjordhur
Olafsfjordhur
Dalvik
Husavik
Raufarhofn
Thorshofn
Vopnafjordhur
Seydhisfjordhur
Neskaupstadhur
Faskrudsfjordhur
Djupivogur
Hofn
Bolungavik
Isafjordhur
Thingeyri
Vatneyri
Hvolsvollur
Hella
Kopavogur
Hafnarfjordhur


DENMARK
North Jutlandic Island
Laso
Mors
Kattegat
Samso
Sjaelland
Funen
Ree
Langeland
Lolland
Falster
Ale
Aero
Copenhagen
GERMANY


400km
Norway
Norwegian Sea
Baltic Sea
North


FRANCE
Basel
Zürich
Biel
Lucerne
LIECHTENSTEIN
Bern
SWITZERLAND
Lausanne
Geneva
Lugano

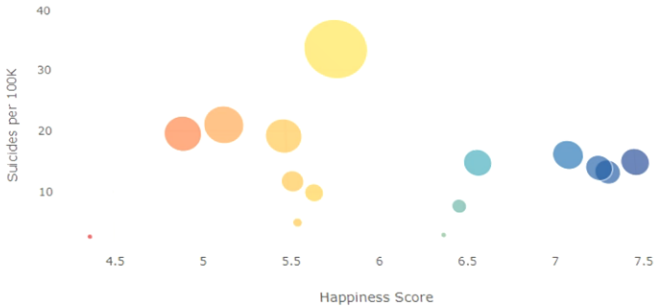
Let's Visit our Dashboard

Use the drop down button below to view charts by year.

2016 Chart

World Happiness Map

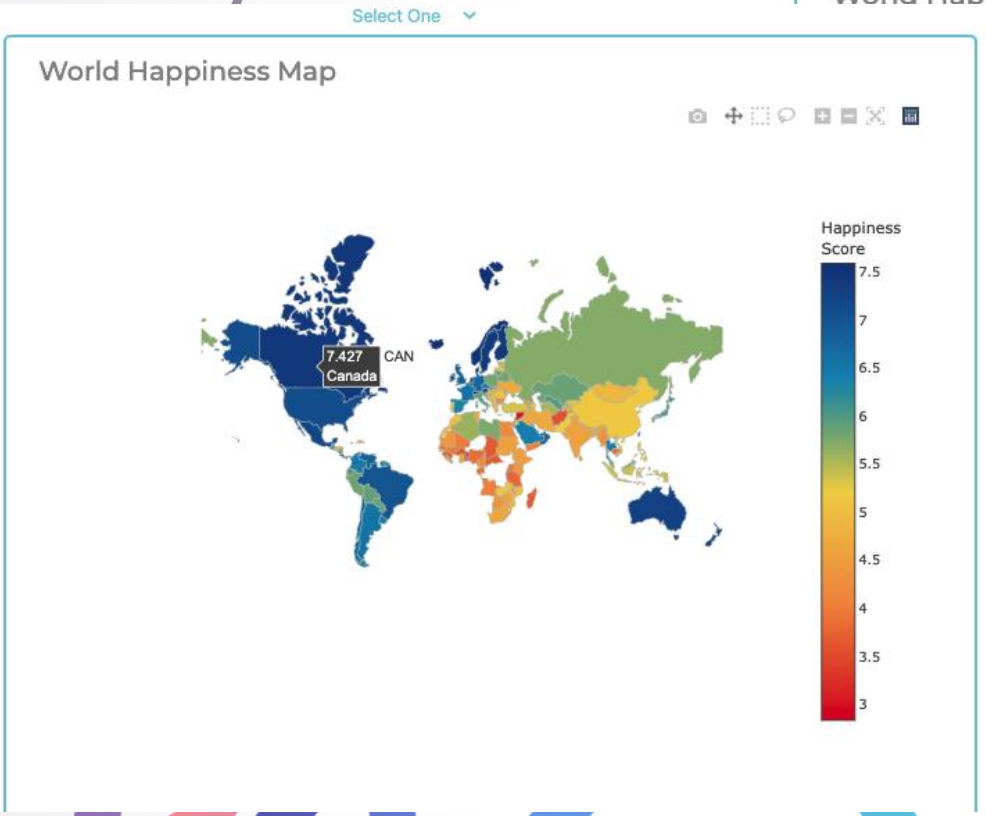
Happiness Score vs Suicide Rates by Country



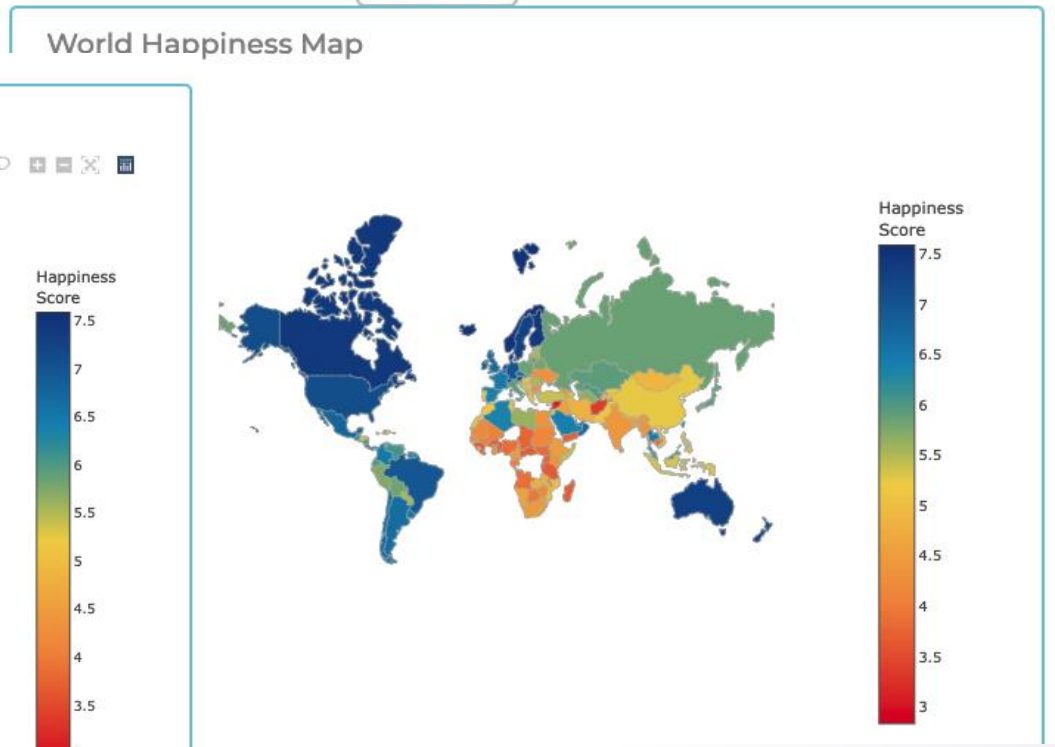
Tour of countries on map

Map

Select One



2016 Chart



Countries that were happy stayed happy

Country	Happiness Score	Happiness Rank
Switzerland	7.587	1
Iceland	7.561	2
Denmark	7.527	3
Norway	7.522	4
Canada	7.427	5
Finland	7.406	6
Netherlands	7.378	7
Sweden	7.364	8
New Zealand	7.286	9
Australia	7.284	10
Israel	7.278	11
Costa Rica	7.226	12
Austria	7.2	13
Mexico	7.187	14
United States	7.119	15
Brazil	6.983	16
Luxembourg	6.946	17

Denmark	7.526	1
Switzerland	7.509	2
Iceland	7.501	3
Norway	7.498	4
Finland	7.413	5
Canada	7.404	6
Netherlands	7.339	7
New Zealand	7.334	8
Australia	7.313	9
Sweden	7.291	10
Israel	7.267	11
Austria	7.119	12
United States	7.104	13
Costa Rica	7.087	14
Puerto Rico	7.039	15
Germany	6.994	16
Brazil	6.952	17
Belgium	6.929	18

JavaScript code that went into building these:

Sort the Data

```
for (var i = 0; i < myData.happiness.length; i++) {
    if (myData.happiness[i].year == '2015') {

        var country = myData.happiness[i].Country;
        var code = myData.happiness[i].code;
        var rank = myData.happiness[i].happiness_rank;
        var score = myData.happiness[i].happiness_score;

        rows.push({
            "country": country,
            "score": score,
            "rank": rank,
            "code": code
        });
    }

};
```

Prep data for plotting

```
function unpack(rows, key) {
    return rows.map(function (row) { return row[key]; });
}

var data = [{
    type: 'choropleth',
    locations: unpack(rows, 'code'),
    z: unpack(rows, 'score'),
    text: unpack(rows, 'country'),
    colorscale: 'Portland',
    autocolorscale: false,
    reversescale: true,
    marker: {
        line: {
            color: 'rgb(180,180,180)',
            width: 0.5
        }
    },
    tick0: 0,
    zmin: 0,
    dtick: 1000,
    colorbar: {
        autotic: false,
        tickprefix: '',
        title: 'Happiness<br>Score'
    }
}];
```

Plot data to map and table

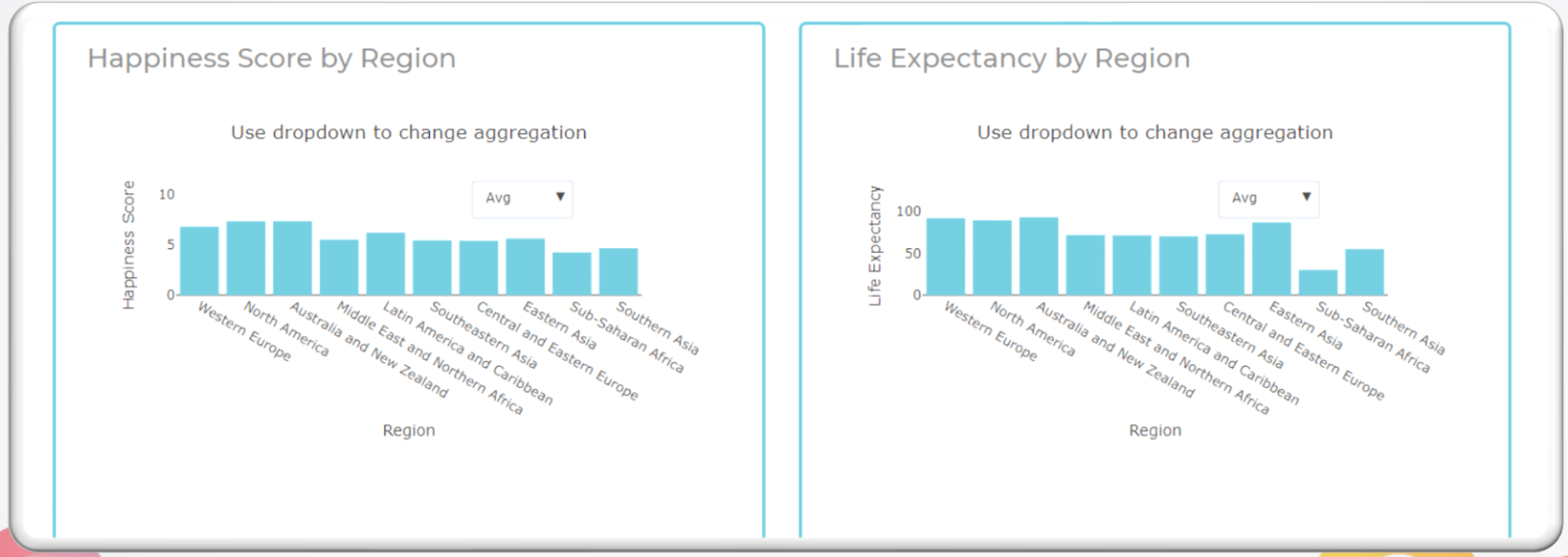
```
var layout = {
    geo: {
        showframe: false,
        showcoastlines: false,
        projection: {
            type: 'mercator'
        }
    },
    width: 800,
    height: 650
};
Plotly.newPlot("myDiv", data, layout, {showLink: false}, {responsive: true});


rows.forEach((row_data) => {
    // Create tr for each row of the table
    const row = tbody.append("tr");

    // Create multiple td cells for each row
    Object.values(row_data).forEach((value) => {
        if (value != '2015') {
            let cell = row.append("td");
            cell.text(value);
        }
    });
});
```

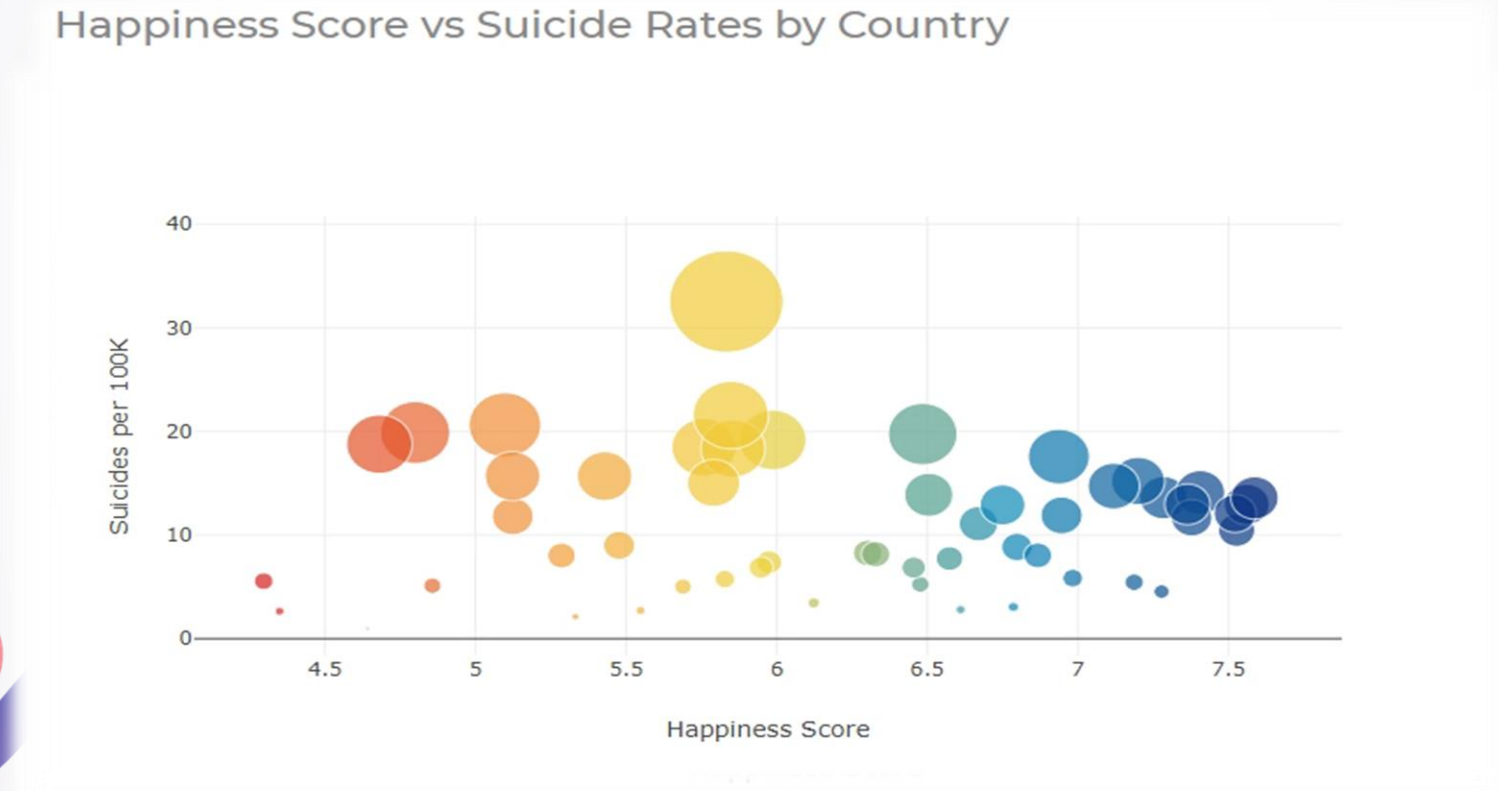
life expectancy & happiness score

Based on average there seem to be a relation



Suicides VS Happiness Score

Country happiness score vs it's number of suicides



Next Steps

In addition to the analysis mentioned before, our wishlist is as below:

- Alignment fine tuning of certain elements throughout the page, including the colorbar on the world map, the card holders for the visuals, and the buttons on the bar graphs,
- Additional styling available,
- Additional pages to dig deeper into the visualizations,
- Reset the table data upon reload of the button.

Thanks!

Any questions?