W3

July 19, 2021

1 Part 1: Visualization Technique

A. Narrative I found the interesting dataset in Kaggle concerning alcohol consumption in the world for adults 15 years and older. I was immediately curious as I pursue a healthy lifestyle. This dataset covers many different regions and countries, and contains many variables that could produce interesting findings.

The Dataset: Alcohol assumption in 2008 downloaded from Kaggle. The original data source came from Gap Minder, a non-profit research entity. The full descriptions of the curated dataset is discussed in detail here: http://makemeanalyst.com/download-and-learn-about-gapminder-dataset/

After having worked on this assignment for a while, I realized that my original dataset is rather limited. It doesn't have interesting layers of categorical data to enhance my visualizations. Hence, I browsed around the web to obtain two additional datasets. These came from the World Health Organization that contains country,region, and income information for the countries of interest. Please note that my demo will attempt to call out this realization and documented the steps chronologically. https://www.who.int/news-room/fact-sheets/detail/alcohol

B. Visualization Techniques:

I will demonstrate some basic charts that were covered in the course, but in this new toolkit (Plotly). Specifically, the charts that I will show are: - Histogram - Bar charts with color coding - Bubble map - Scatter plots (for fun)

C. Discussion:

With this mixed bag of the intended charts above, I will sequentially describe how they work and when they should not be used 1. Histogram - Measures the frequency of a continuous numerical vareiable in the dataset. The bins could be specified - This is not appropriate for categorical data 2. Bar charts with color coding - This is unarguably the most popular visual for anyone. Typtically, one axis is a category variable and the other one shows the numercial data. The data doesn't have to the continuous. I encoded an "extra" feature that shows certain types of grouping as a color callout (see the demo later) 3. Bubble map - I used to work in finance, and bubble chart is used widely to measure sales and their impact across different categories/conditions. And since I have a global dataset, why not combine a bubble and map to create a bubble map? This visual not only effectively describes trends in your data but also offers a "sizing" component (i.e., the bigger the size (coded with a numberical var), the bigger the impact of that data point is with respect to that category. (e.g., Sales rep sales by region and size = total sales made) - It is often not possible to produce this chart if you don't have geographic data for many data points. 4. Scatter plots (extra just for fun) - I would love to understand the relationships between the variables in the dataset.

This chart is often used to understanding the linear correlation between the x and y variables. - x and y must be numberical for this chart to be used.

2 Part 2: Visualization Library

I chose Plotly, an open-source Python library developed by a Canadian firm founded in 2012. The toolkit supports 40 unique plots. It is a more aesthetically pleasing than matplotlib and offers interactive capablity. This comes in handy for my demo as you will see later that it is quite nice to be able to hover your mouse on the data points and glean some data right on the spots! Plotly also highly integrates with Dash, an open-source framework for building analytical applications, with no Javascript required. I will continue to tinker with this library as I progress throughout my journey as a data scientist.

I checked out basic Python documentation via this link:

https://plotly.com/python/

I installed through Conda command line %conda install plotly

[1]:
```
# I will import the crucial standard libraries and modules
import plotly
import plotly.express as px
import pandas as pd
import numpy as np
```

[2]:
```
print(plotly.__version__)
```

```
4.14.3
```

3 Part 3: Demonstration

[3]:
```
# Loading the data, basic manipulations for cleaning and removing unwanted rows
```

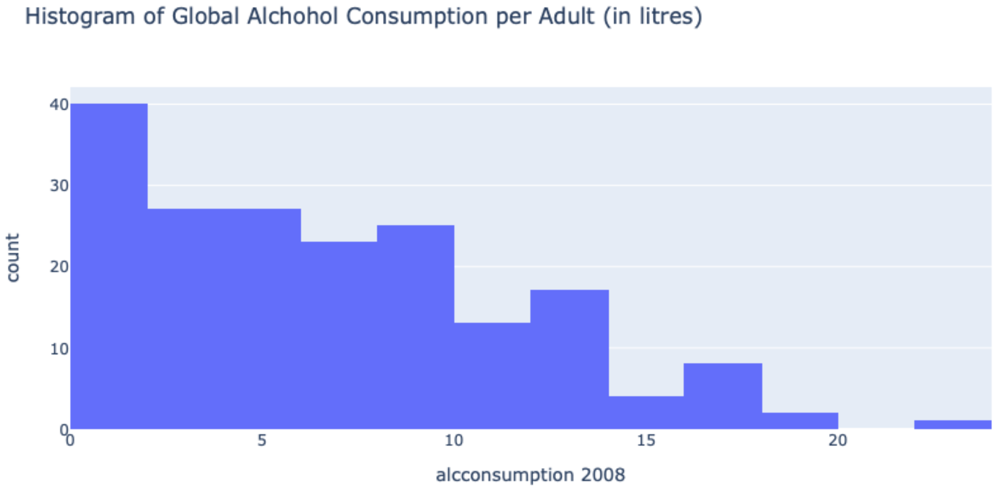
[4]:
```
a_data = pd.read_csv('gapminder_alcohol.csv')
print(a_data.shape)
a_data.set_index('country', inplace=True)
a_data.rename(columns={'alcconsumption':'alcconsumption 2008'}, inplace=True)
a_data.head()
```

```
(213, 6)
```

[4]:
```
             alcconsumption 2008  incomeperperson  suicideper100th  \
country
Afghanistan                 0.03              NaN         6.684385
Albania                     7.29      1914.996551         7.699330
Algeria                     0.69      2231.993335         4.848770
Andorra                    10.17     21943.339900         5.362179
Angola                      5.57      1381.004268        14.554677
```

```
             employrate  urbanrate
country
Afghanistan   55.700001      24.04
Albania       51.400002      46.72
Algeria       50.500000      65.22
Andorra             NaN      88.92
Angola        75.699997      56.70
```

```
[5]: # lets explore a basic histogram of alchohol consumption measured in litres for␣
     ↪adults >= 15 years old to see the
     # frequency of the different buckets measured in litres
     fig = px.histogram(a_data, x="alcconsumption 2008", title='Histogram of Global␣
     ↪Alchohol Consumption per Adult (in litres)')
     fig.show()
```



```
[6]: # The distribution looks right-skewed to me. I'm curious about the basic␣
     ↪summary statistics of the level of alcohol
     # consumptions. Let's perform some basic statistical measures.
     a_data.describe()
```

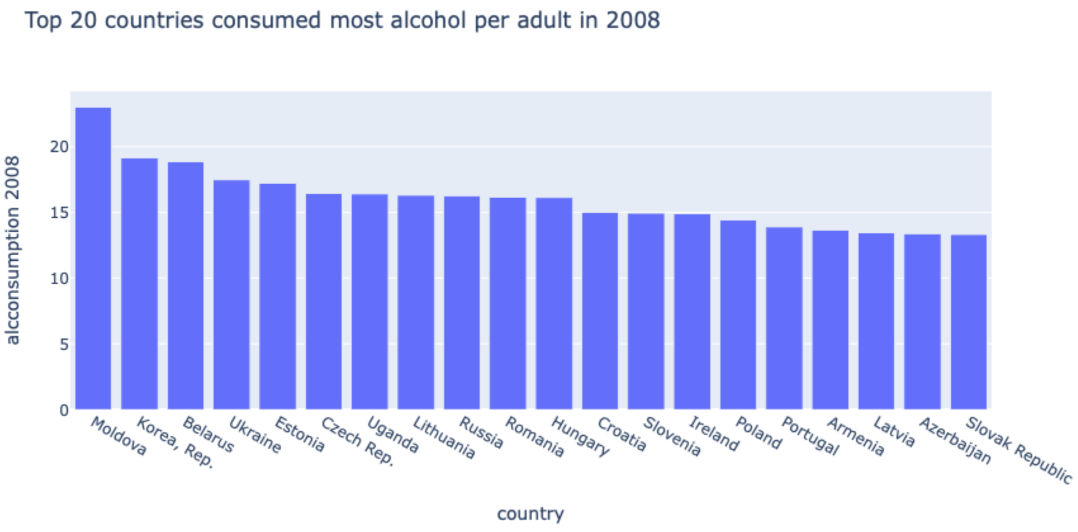
```
[6]:        alcconsumption 2008  incomeperperson  suicideper100th  employrate  \
     count           187.000000       190.000000       191.000000  178.000000
     mean              6.689412      8740.966076         9.640839   58.635955
     std               4.899617     14262.809083         6.300178   10.519454
     min               0.030000       103.775857         0.201449   32.000000
     25%               2.625000       748.245151         4.988449   51.225000
     50%               5.920000      2553.496056         8.262893   58.699999
```

```
75%             9.925000     9379.891166        12.328551   64.975000
max            23.010000   105147.437700        35.752872   83.199997

        urbanrate
count  203.000000
mean    56.769360
std     23.844933
min     10.400000
25%     36.830000
50%     57.940000
75%     74.210000
max    100.000000
```

[7]:
```
# I want to find out the top countries that consume most alcohol in the world
# I will slice a separate df for these top 15 countries
al_df= a_data[['alcconsumption 2008']]
sort_df = al_df.sort_values('alcconsumption 2008', ascending=False)
top20 = sort_df.iloc[:20,:]
top20
```

[7]:
```
                 alcconsumption 2008
country
Moldova                        23.01
Korea, Rep.                    19.15
Belarus                        18.85
Ukraine                        17.47
Estonia                        17.24
Czech Rep.                     16.47
Uganda                         16.40
Lithuania                      16.30
Russia                         16.23
Romania                        16.15
Hungary                        16.12
Croatia                        15.00
Slovenia                       14.94
Ireland                        14.92
Poland                         14.43
Portugal                       13.89
Armenia                        13.66
Latvia                         13.45
Azerbaijan                     13.34
Slovak Republic                13.31
```

[8]:
```
fig = px.bar(top20, x = top20.index, y = 'alcconsumption 2008', title = 'Top 20␣
 ↪countries consumed most alcohol per adult in 2008')
fig.show()
```



[9]:
```
# now i'm going to bring in two other sources of data to provide an additional dimensions--region and incomegroup so that we could enhance
# this bar chart by turning it into more celebrated bar charts

# 1) country code that will bind these dataframes together (df2)
# 2) region and income information for the countries of interest (df3)
# Finally, i will merge the three dataframes together for a holistic view.
```

[10]:
```
df2 = pd.read_csv('country_code.csv')
df2 = df2.drop(columns=['Indicator Name', 'Indicator Code']) # dropping non-essential columns
print(df2.shape)
df2.head()
```

```
(266, 63)
```

[10]:
```
                  Country Name Country Code  1960  1961  1962  1963  1964  \
0                        Aruba          ABW   NaN   NaN   NaN   NaN   NaN
1  Africa Eastern and Southern          AFE   NaN   NaN   NaN   NaN   NaN
2                  Afghanistan          AFG   NaN   NaN   NaN   NaN   NaN
3   Africa Western and Central          AFW   NaN   NaN   NaN   NaN   NaN
4                       Angola          AGO   NaN   NaN   NaN   NaN   NaN

   1965  1966  1967  …  2011  2012  2013  2014      2015  2016  2017  \
0   NaN   NaN   NaN  …   NaN   NaN   NaN   NaN       NaN   NaN   NaN
1   NaN   NaN   NaN  …   NaN   NaN   NaN   NaN  5.200565   NaN   NaN
2   NaN   NaN   NaN  …   NaN   NaN   NaN   NaN  0.210000   NaN   NaN
3   NaN   NaN   NaN  …   NaN   NaN   NaN   NaN  6.869468   NaN   NaN
```

```
4   NaN   NaN   NaN  …   NaN   NaN   NaN   NaN  7.960000   NaN   NaN

        2018  2019  2020
0        NaN   NaN   NaN
1   5.170911   NaN   NaN
2   0.210000   NaN   NaN
3   6.835266   NaN   NaN
4   6.940000   NaN   NaN

[5 rows x 63 columns]
```

[11]:

```
df3 = pd.read_csv('region_incomegroup.csv')
df3 = df3.drop(columns=['SpecialNotes','TableName', 'Unnamed: 5'])
print(df3.shape)
df3.head()
```

```
(265, 3)
```

[11]:

```
  Country Code                     Region          IncomeGroup
0          ABW  Latin America & Caribbean          High income
1          AFE                        NaN                  NaN
2          AFG                 South Asia           Low income
3          AFW                        NaN                  NaN
4          AGO         Sub-Saharan Africa  Lower middle income
```

[12]:

```
# I will merge df2 and df3 together
df_merged = pd.merge(df2, df3,how='outer',left_on='Country Code',␣
 ↪right_on='Country Code')
print(df_merged.shape)
df_merged.head()
```

```
(266, 65)
```

[12]:

```
                  Country Name Country Code  1960  1961  1962  1963  1964  \
0                        Aruba          ABW   NaN   NaN   NaN   NaN   NaN
1  Africa Eastern and Southern          AFE   NaN   NaN   NaN   NaN   NaN
2                  Afghanistan          AFG   NaN   NaN   NaN   NaN   NaN
3   Africa Western and Central          AFW   NaN   NaN   NaN   NaN   NaN
4                       Angola          AGO   NaN   NaN   NaN   NaN   NaN

   1965  1966  1967  …  2013  2014      2015  2016  2017      2018  2019  \
0   NaN   NaN   NaN  …   NaN   NaN       NaN   NaN   NaN       NaN   NaN
1   NaN   NaN   NaN  …   NaN   NaN  5.200565   NaN   NaN  5.170911   NaN
2   NaN   NaN   NaN  …   NaN   NaN  0.210000   NaN   NaN  0.210000   NaN
3   NaN   NaN   NaN  …   NaN   NaN  6.869468   NaN   NaN  6.835266   NaN
4   NaN   NaN   NaN  …   NaN   NaN  7.960000   NaN   NaN  6.940000   NaN
```

```
   2020                     Region          IncomeGroup
0   NaN  Latin America & Caribbean          High income
1   NaN                        NaN                  NaN
2   NaN                 South Asia           Low income
3   NaN                        NaN                  NaN
4   NaN         Sub-Saharan Africa  Lower middle income

[5 rows x 65 columns]
```

[13]:

```
# merging all 3 tables together
combined_df = pd.merge(a_data, df_merged,how='inner', left_index=True,␣
 ↪right_on='Country Name')
combined_df.set_index('Country Name', inplace=True)
print(combined_df.shape)
combined_df.head()
```

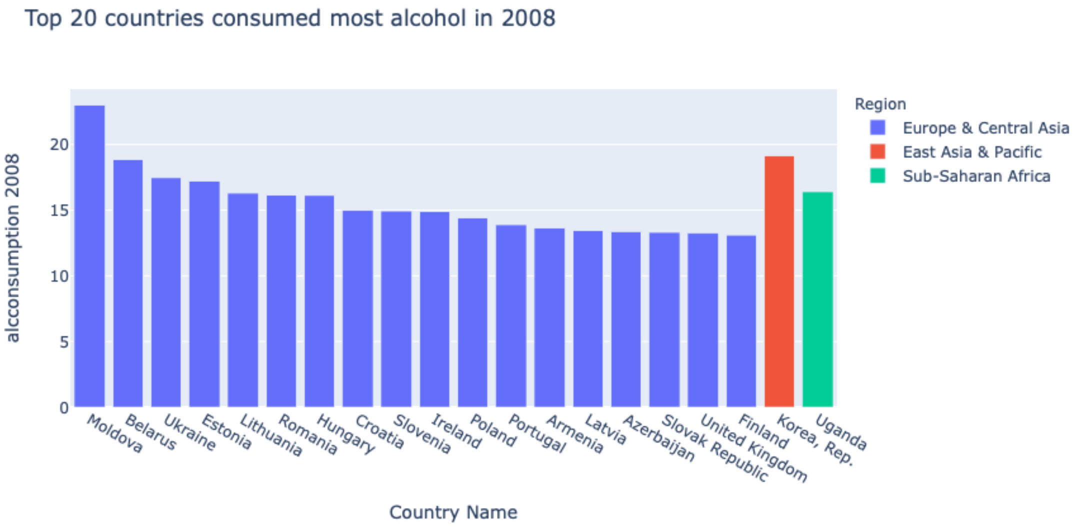
```
(182, 69)
```

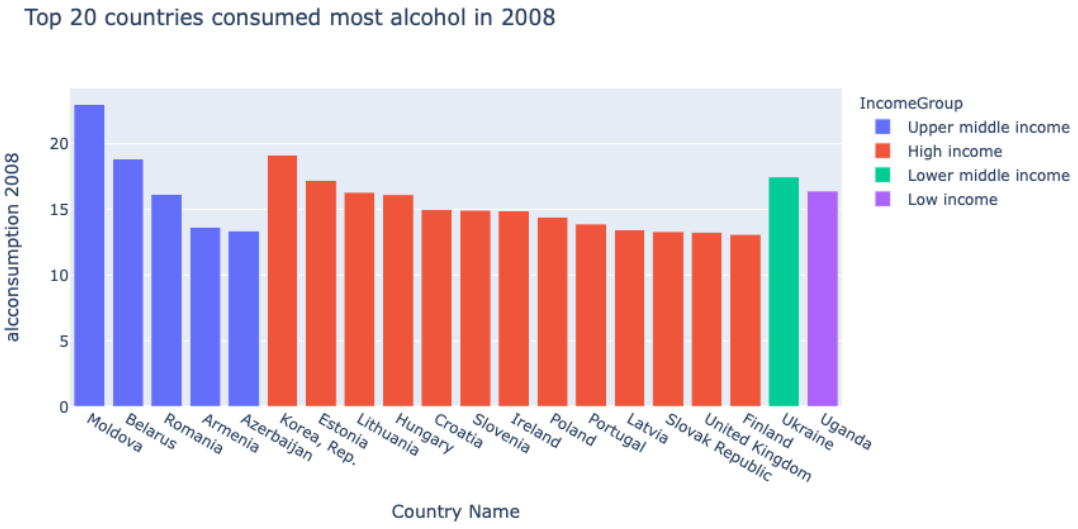
[13]:

```
              alcconsumption 2008  incomeperperson  suicideper100th  \
Country Name
Afghanistan                  0.03              NaN         6.684385
Albania                      7.29      1914.996551         7.699330
Algeria                      0.69      2231.993335         4.848770
Andorra                     10.17     21943.339900         5.362179
Angola                       5.57      1381.004268        14.554677

              employrate  urbanrate Country Code  1960  1961  1962  1963  ...  \
Country Name                                                              ...
Afghanistan    55.700001      24.04          AFG   NaN   NaN   NaN   NaN  ...
Albania        51.400002      46.72          ALB   NaN   NaN   NaN   NaN  ...
Algeria        50.500000      65.22          DZA   NaN   NaN   NaN   NaN  ...
Andorra              NaN      88.92          AND   NaN   NaN   NaN   NaN  ...
Angola         75.699997      56.70          AGO   NaN   NaN   NaN   NaN  ...

              2013  2014   2015  2016  2017   2018  2019  2020  \
Country Name
Afghanistan    NaN   NaN   0.21   NaN   NaN   0.21   NaN   NaN
Albania        NaN   NaN   6.74   NaN   NaN   7.17   NaN   NaN
Algeria        NaN   NaN   0.93   NaN   NaN   0.95   NaN   NaN
Andorra        NaN   NaN  11.01   NaN   NaN  11.02   NaN   NaN
Angola         NaN   NaN   7.96   NaN   NaN   6.94   NaN   NaN

                                   Region          IncomeGroup
Country Name
Afghanistan                    South Asia           Low income
Albania             Europe & Central Asia  Upper middle income
Algeria        Middle East & North Africa  Lower middle income
```

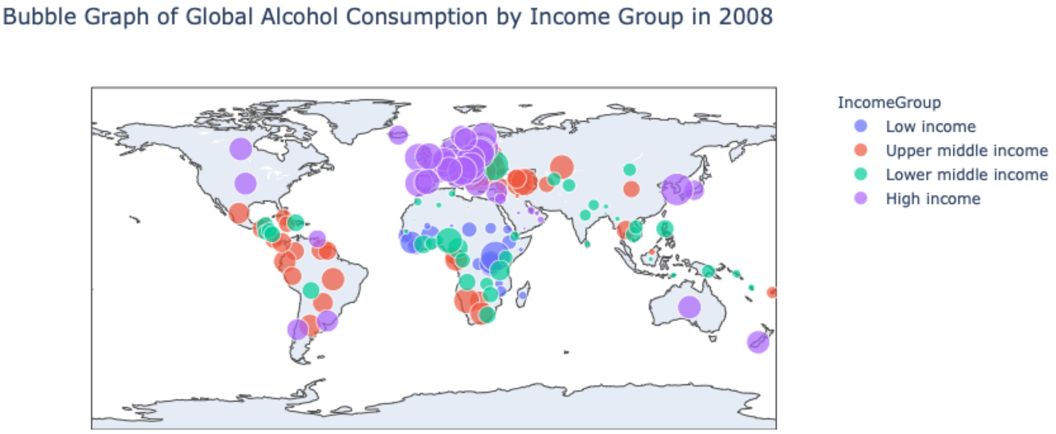
```
Andorra              Europe & Central Asia          High income
Angola                  Sub-Saharan Africa  Lower middle income

[5 rows x 69 columns]
```

```
[14]: # now i want to revisit my bar chart above and add another level of dimension␣
       ↪to it.
      sort_df2 = combined_df.sort_values('alcconsumption 2008', ascending=False)
      top20 = sort_df2.iloc[:20,:]

      #let's get plotting on this df and add 'region code' and 'income group',␣
       ↪respectively, to encode the colors

      fig = px.bar(top20, x = top20.index, y = 'alcconsumption 2008', color='Region',␣
       ↪title = 'Top 20 countries consumed most alcohol in 2008')
      fig.show()
      fig = px.bar(top20, x = top20.index, y = 'alcconsumption 2008',␣
       ↪color='IncomeGroup', title = 'Top 20 countries consumed most alcohol in␣
       ↪2008')
      fig.show()
```



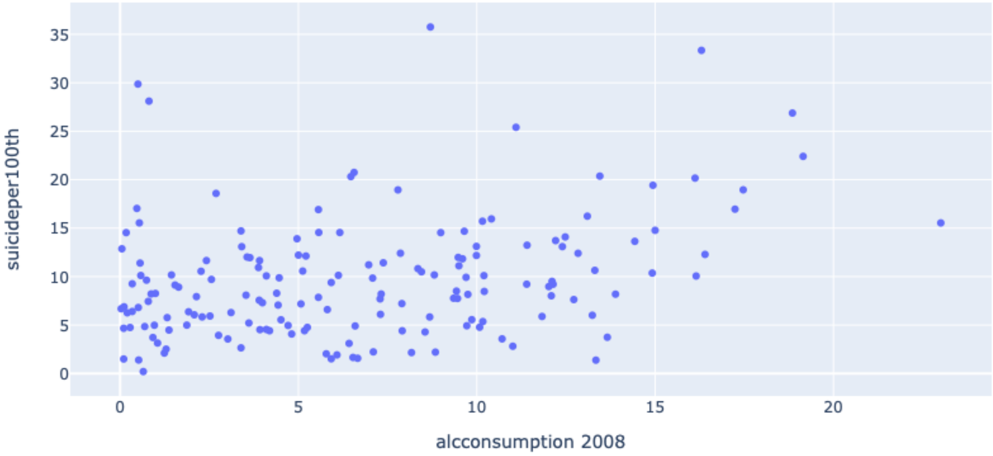


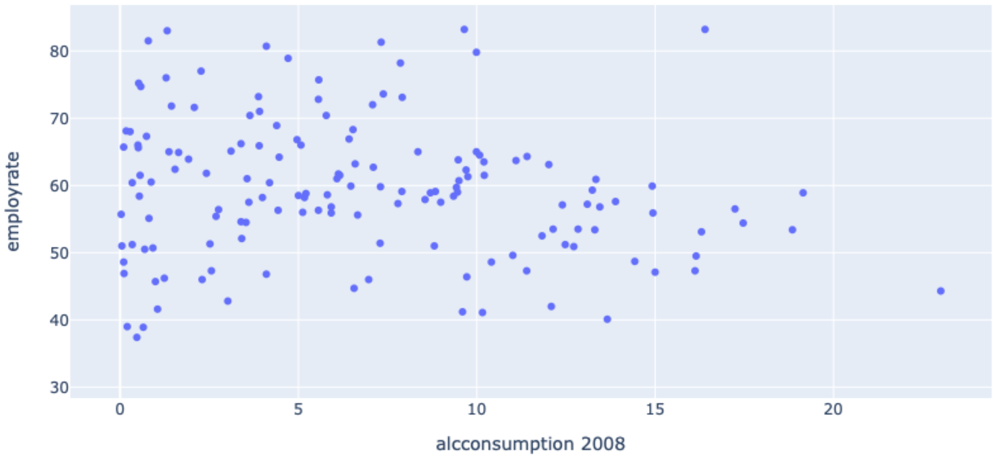
```
[15]: # Wow, this is so much better!
      # Finally, I will create a bubble map that show the total avg alcohol␣
       ↪consumption per capita. The only thing I could
      # have add is to code 'year'as the animation frame but unfortunately our yearly␣
       ↪data is sporadic
      combined_df['2008'] = combined_df['alcconsumption 2008'].fillna(0)
      fig = px.scatter_geo(combined_df, locations='Country Code', color='IncomeGroup',
                           hover_name=combined_df.index, size='2008',
                          title='Bubble Graph of Global Alcohol Consumption by Income␣
       ↪Group in 2008')
      fig.show()
```



[16]:

```
# Bonus just for fun
# When I have more time to work on this, perhaps I want to look at something␣
 ↪more predictive than just simply
# summarizing data
# Moving on from descriptive statistics, I want to explore potential␣
 ↪relationships between these variables in the dataset
# Let's plot a few scatter plots for differnt pair of x and y axes

fig = px.scatter(combined_df, x='alcconsumption 2008', y='suicideper100th')
fig.show()
fig = px.scatter(combined_df, x='alcconsumption 2008', y='employrate')
fig.show()
```





[17]:
```
# it looks like there might exist a positive correlation between alcohol consumption and suicidal rate per 100,000th
# On the other hand, the relationship between alcohol consumption and employment rate is not quiet clear
```