W.H.O. SUICIDE ANALYSIS

Kartik Kadian

CONTENTS

Background:

- The World Health Organisation (WHO) estimates that each year approximately one million people die from suicide, which represents a global mortality rate of 16 people per 100,000 or one death every 40 seconds. It is predicted that by 2020 the rate of death will increase to one every 20 seconds.

The WHO further reports that:

- In the last 45 years' suicide rates have increased by 60% worldwide. Suicide is now among the three leading causes of death among those aged 15-44 (male and female). Suicide attempts are up to 20 times more frequent than completed suicides.
- Although suicide rates have traditionally been highest amongst elderly males, rates among young people have been increasing to such an extent that they are now the group at highest risk in a third of all countries.
- Mental health disorders (particularly depression and substance abuse) are associated with more than 90% of all cases of suicide.
- However, suicide results from many complex sociocultural factors and is more likely to occur during periods of socioeconomic, family and individual crisis (e.g. loss of a loved one, unemployment, sexual orientation, difficulties with developing one's identity, disassociation from one's community or other social/belief group, and honour).

The WHO also states that:

- In Europe, particularly Eastern Europe, the highest suicide rates are reported for both men and women.
- The Eastern Mediterranean Region and Central Asia republics have the lowest suicide rates.
- Nearly 30% of all suicides worldwide occur in India and China.
- Suicides globally by age are as follows: 55% are aged between 15 to 44 years and 45% are aged 45 years and over.
- Youth suicide is increasing at the greatest rate.

In the US, the Centre of Disease Control and Prevention reports that:

- Overall, suicide is the eleventh leading cause of death for all US Americans, and is the third leading cause of death for young people 15-24 years.
- Although suicide is a serious problem among the young and adults, death rates continue to be highest among older adults ages 65 years and over.
- Males are four times more likely to die from suicide than are females. However, females are more likely to attempt suicide than are males.

Load the data:

country	year	sex	age	suicides_no	population
Brazil	1979	female	15-24 years	385.0	12448100.0
Netherlands	1979	male	55-74 years	217.0	1040500.0
Netherlands	1979	male	75+ years	93.0	235800.0
Austria	1979	male	75+ years	nan	141900.0
Austria	1979	male	55-74 years	nan	573800.0

Description of Data:

	year	gender	age	suicides	population
count	43776.0	43776.0	43776.0	43776.0	43776.0
mean	1998.5024671052631	0.5	2.5	183.35286458333334	1664090.9937637062
std	10.338711176745791	0.5000057109896576	1.7078446344032152	780.8578982026199	3412201.2300680885
min	1979.0	0.0	0.0	0.0	259.0
25%	1990.0	0.0	1.0	0.0	118498.25
50%	1999.0	0.5	2.5	11.0	517777.5
75%	2007.0	1.0	4.0	83.0	1664090.0
max	2016.0	1.0	5.0	22338.0	43805214.0

Here, we have 43776 rows and 5 columns as: year, gender, age, suicides, population.

In the above table, we can see that the average of suicides is around 183 suicides out of 1664091 populations.

This is the mathematical description of the data. To understand the whole description of the data, we will do some visualizations in python, and for the future prediction we will use some machine learning algorithms.

Checking the null values:

```
# checkinng the null values in the dataset

data.isnull().sum()
```

```
country           0
year              0
gender            0
age               0
suicides       2256
population     5460
dtype: int64
```

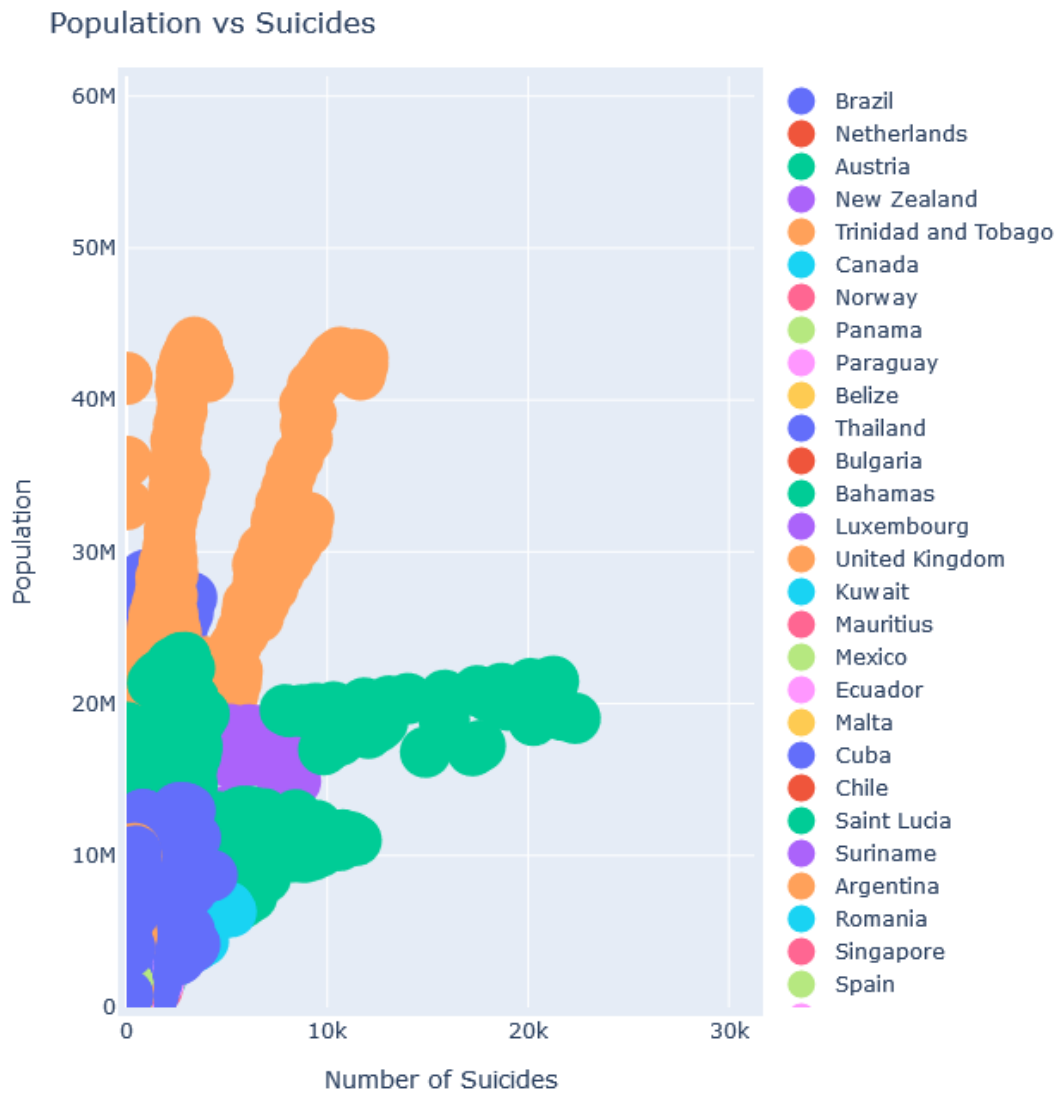
So, we see from the above picture that there are no null values in some columns like: country, year, gender, age which is good thing for our exploratory data analysis. But, as we can see that there are 2256 null values in suicides column and 5460 null values are there in population column which is not a good thing. So, we have to handle with these null values. As suicides can be our target variable and population can be very useful to it, so we cannot drop these columns. So, we will fill these null values with 0.

```
# filling missing values

data['suicides'].fillna(0, inplace = True)
# data['population'].mean()
data['population'].fillna(1664090, inplace = True)
```

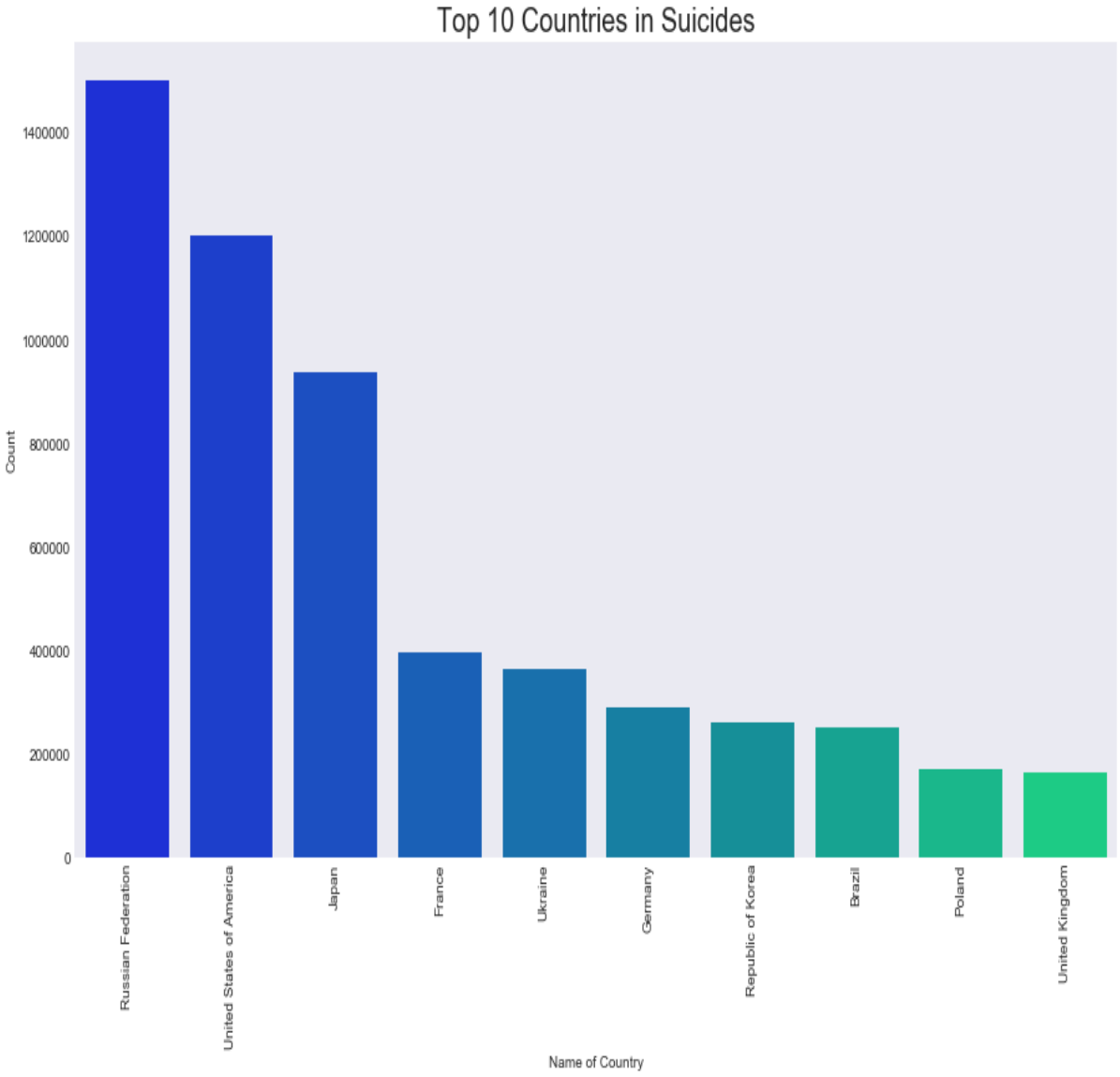
Now, we are done with the null values. So, we can easily do the further Data visualizations.

Bubble Plot:



By looking at the above plot, we can draw a conclusion that the number of suicides is very high in the African and Asian Regions as compared to the American and European Regions.

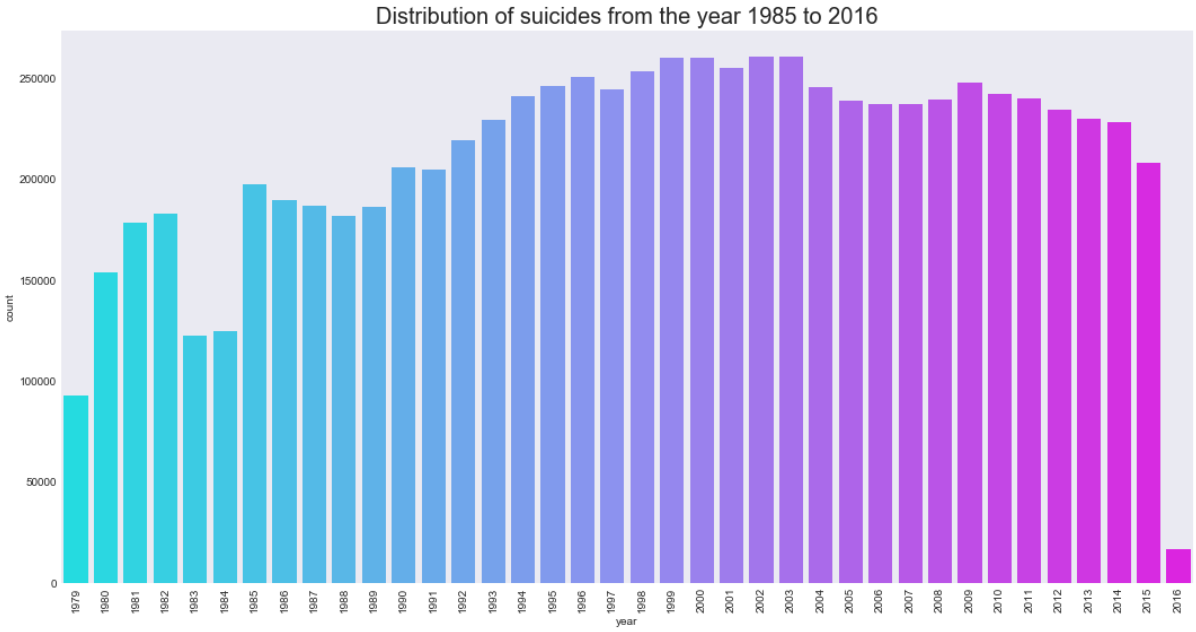
visualising the different countries distribution in the dataset:

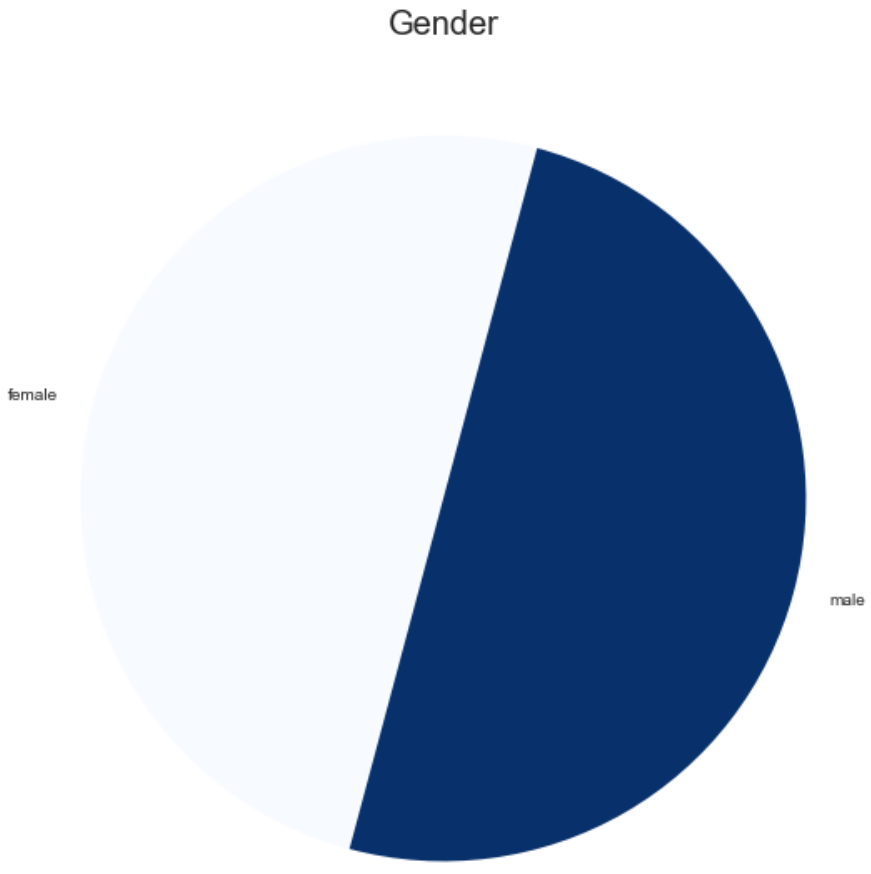


Here, we can look at the Top 10 countries according to the number of suicides in each year. Russia and America which are the most powerful countries in terms of GDP, employment, Growth, Economy, Luxury, rated as one of the best countries to live across the globe are at the top of the list in term of suicides.

Reasons could be unemployment, as living is very costly in these countries, or may be due to drugs/problems in relationships/family related problems etc. could be the reason.

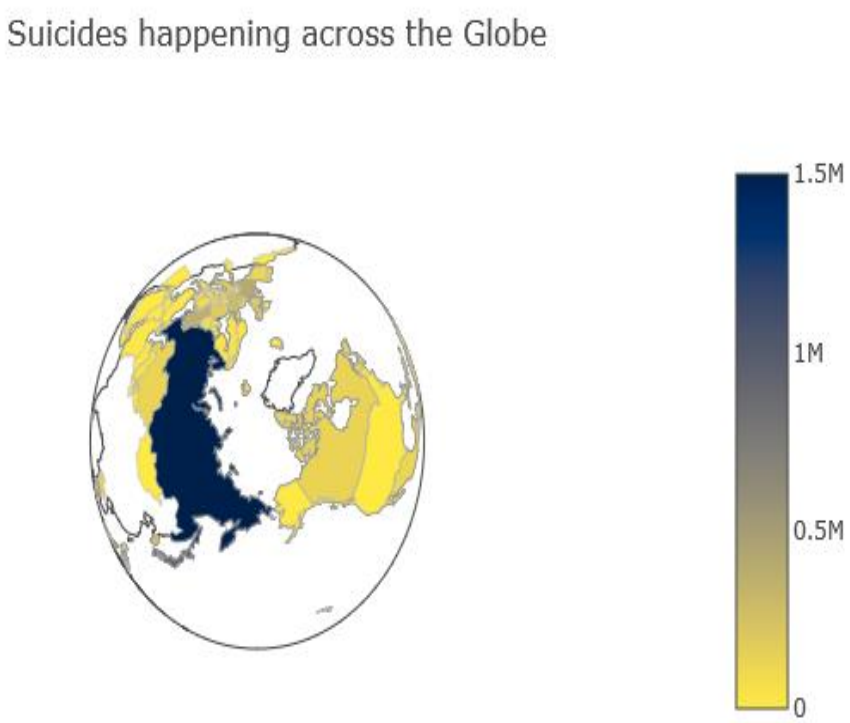
visualising the different year distribution in the dataset:





From the above plots, we can see the distribution of suicides between 1985 and 2016, and the distribution of the suicides among the gender.

Geospatial Analysis for Suicides:



This is the globe plot showing the cases of suicides in the different countries. The area in the dark blue colour shows that in these countries, the suicides are very much, a larger data than the others.

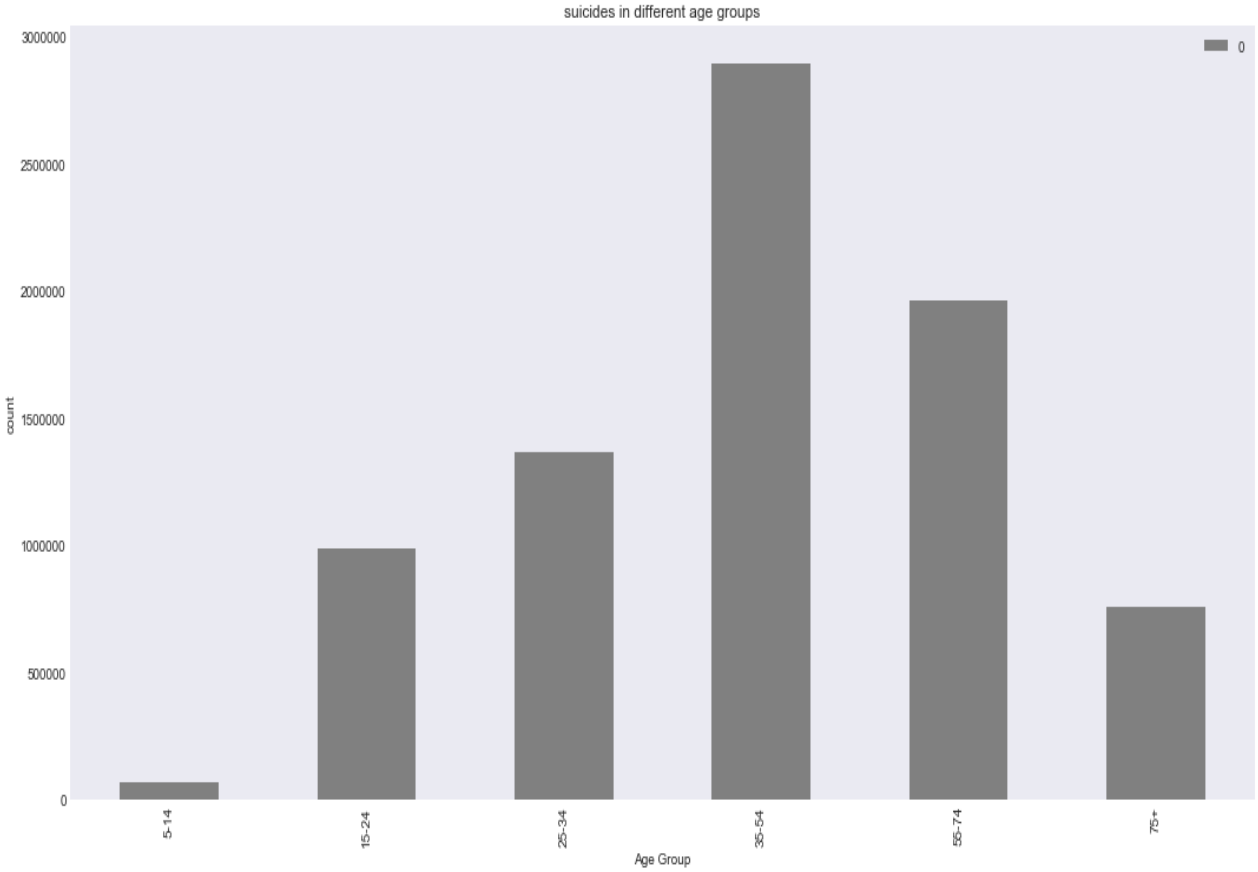
looking at the Suicides in USA.

	country	year	gender	age	suicides	population
42187	United States of America	2001	male	25-34 years	4199	20013572
42106	United States of America	1994	male	55-74 years	4498	18273200
41991	United States of America	1985	female	5-14 years	73	16553000
41923	United States of America	1979	male	25-34 years	4505	17862000
41978	United States of America	1984	female	35-54 years	2267	27106000
42246	United States of America	2006	male	15-24 years	3528	21844954
42059	United States of America	1990	male	75+ years	2653	4585900
41957	United States of America	1982	female	75+ years	367	6914000
42026	United States of America	1988	female	35-54 years	2209	30281000
41925	United States of America	1979	male	5-14 years	104	18075000
42168	United States of America	2000	female	15-24 years	570	19105073
41948	United States of America	1981	male	35-54 years	5462	23879000
42368	United States of America	2016	male	35-54 years	0	41481607
42363	United States of America	2016	female	5-14 years	0	20385205
42088	United States of America	1993	female	55-74 years	1304	21398900
42087	United States of America	1993	female	5-14 years	88	18078200
42255	United States of America	2007	female	5-14 years	53	19714203
41930	United States of America	1980	female	35-54 years	2214	24903000
42183	United States of America	2001	female	5-14 years	65	20032634
42127	United States of America	1998	male	25-34 years	4848	20191300

The above table shows the trend of suicides in U.S.A.

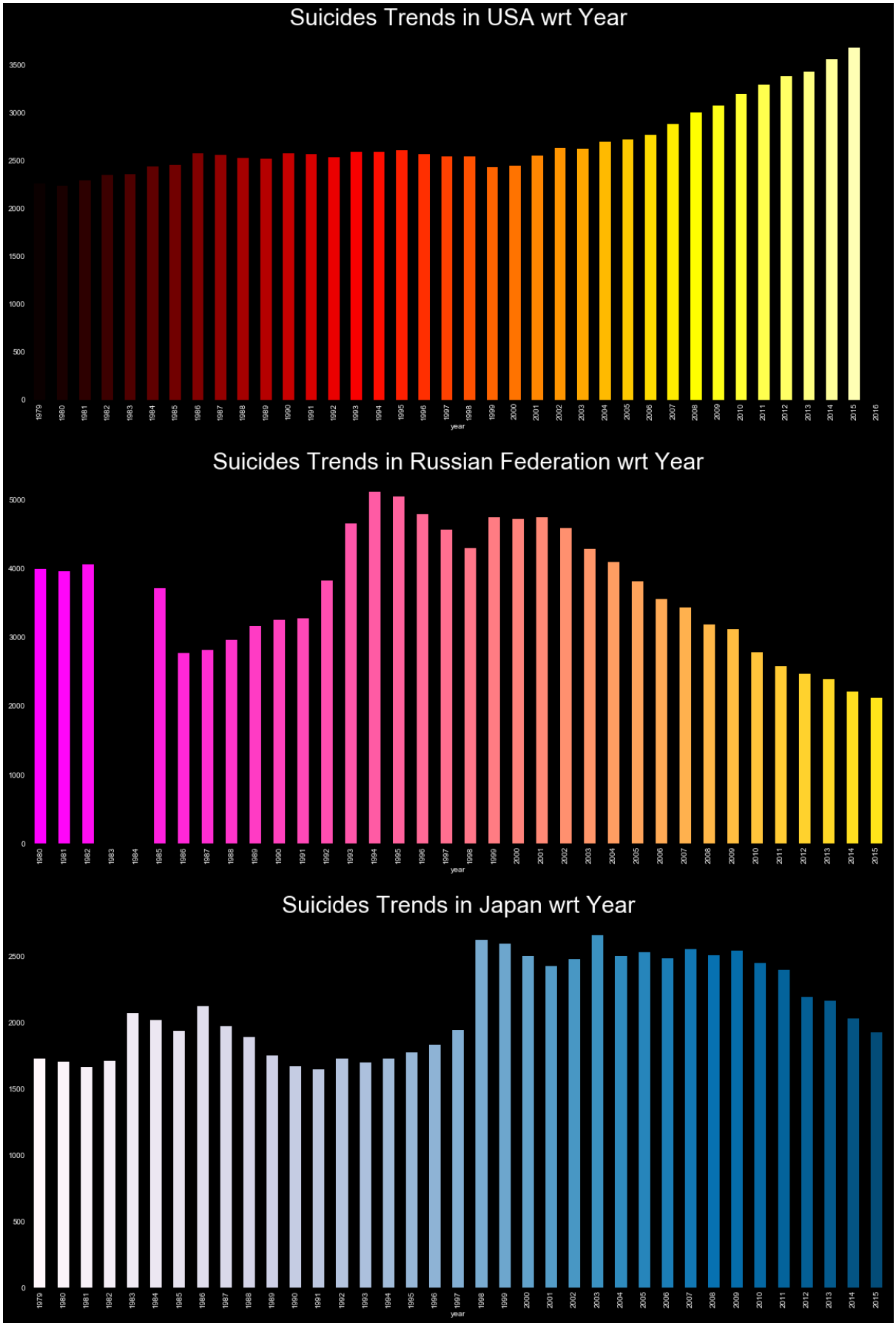
This table contains a sample of 20 random data points from the U.S.A.

suicides in different age groups:

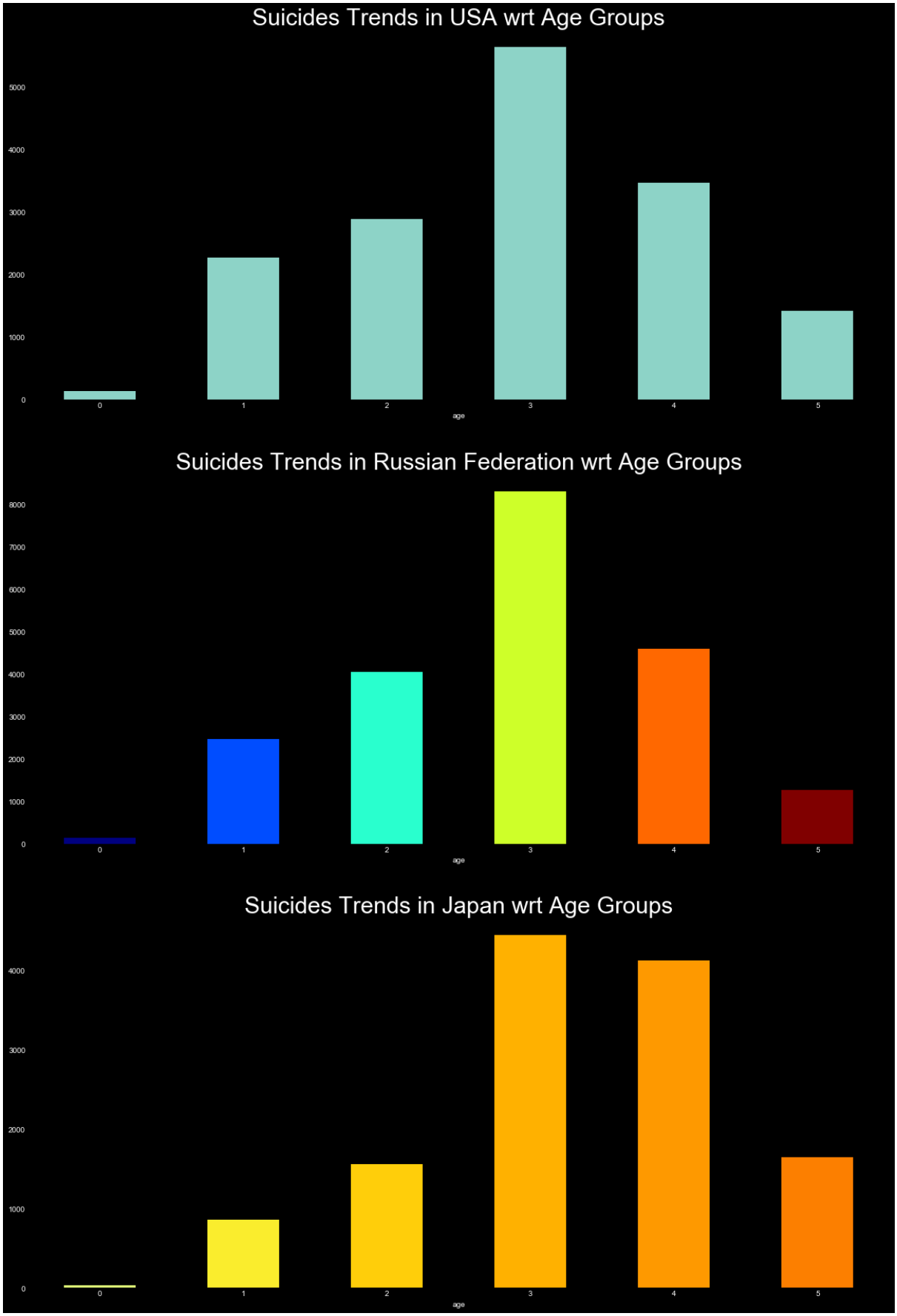


The above figure shows that the maximum suicides are committed by the age group of 35 to 54 years of aged people. And, the minimum suicides are being attempted by the 5 to 14 years of aged people.

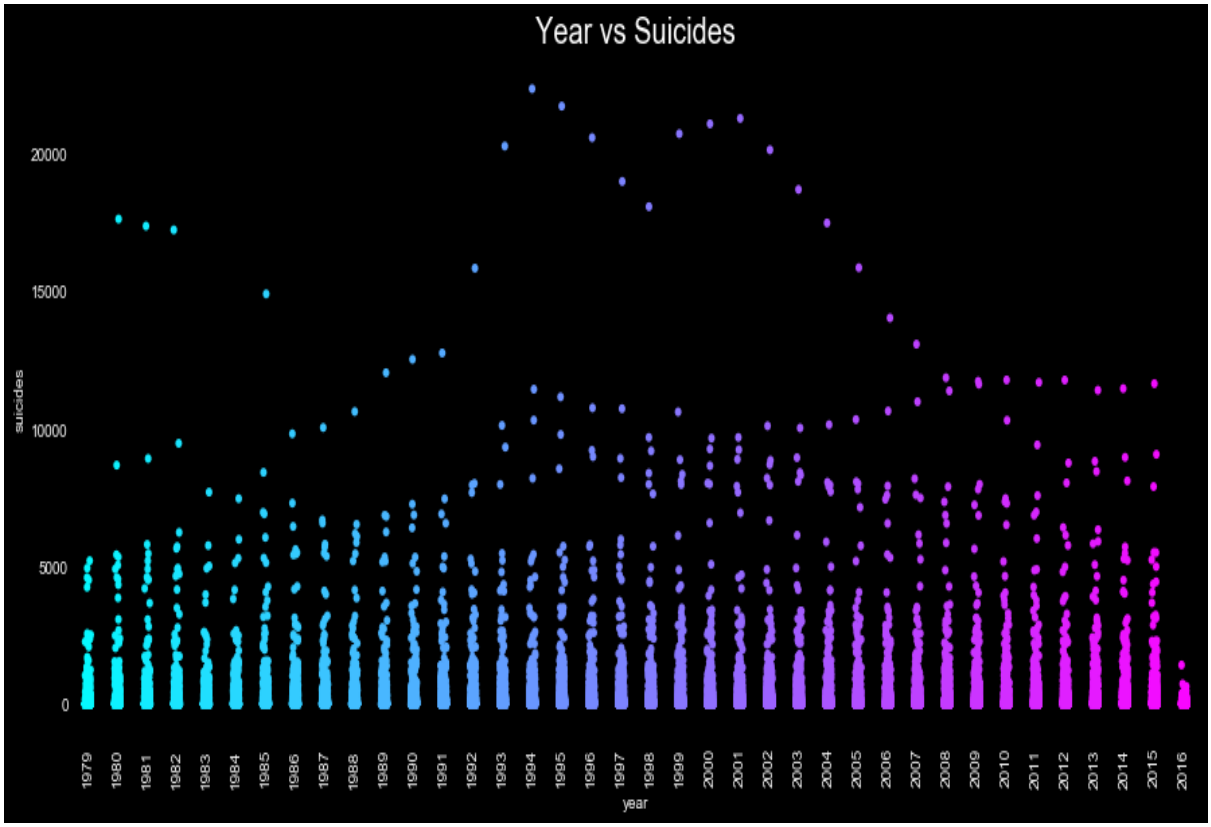
Finding Suicide Trends according to Year:



Finding Suicide Trends according to Age Groups:



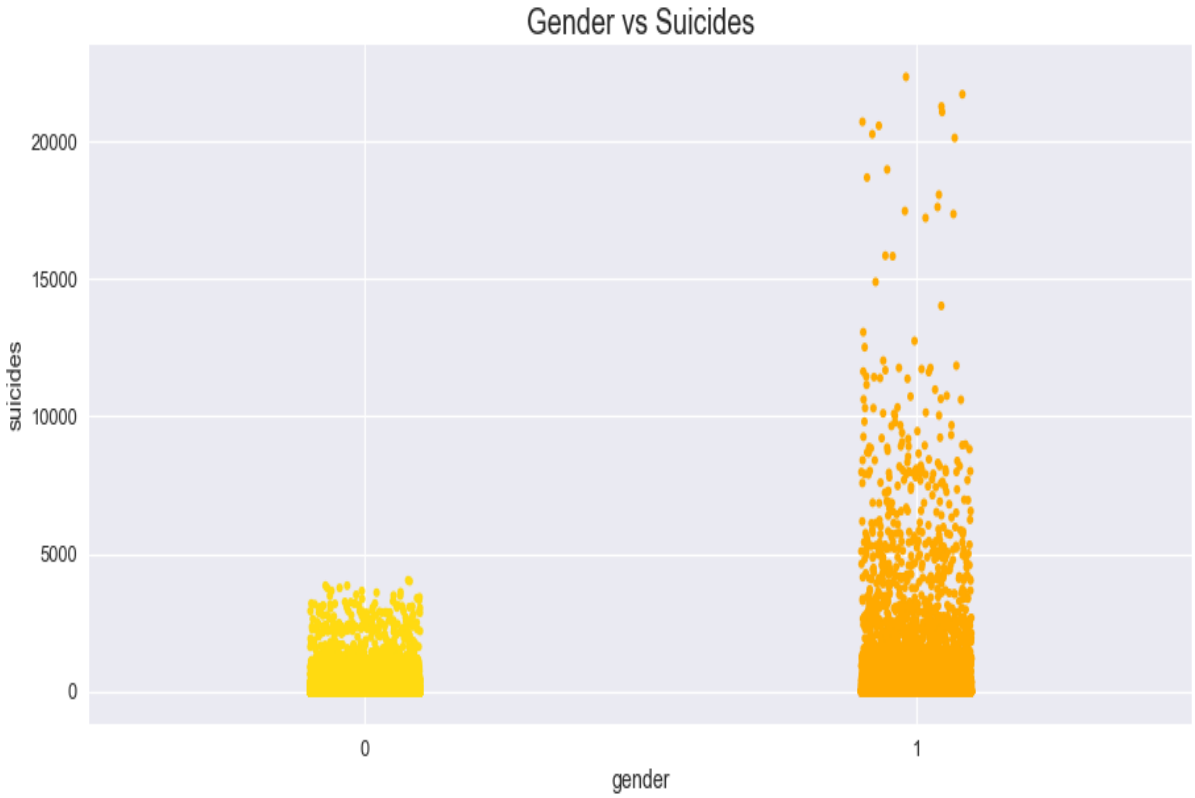
Year vs Suicides:



The above graph shows the suicides with respect to the years.

As we can see from the graph that the maximum suicides were committed in 1995, and the minimum were attempted in 2016.

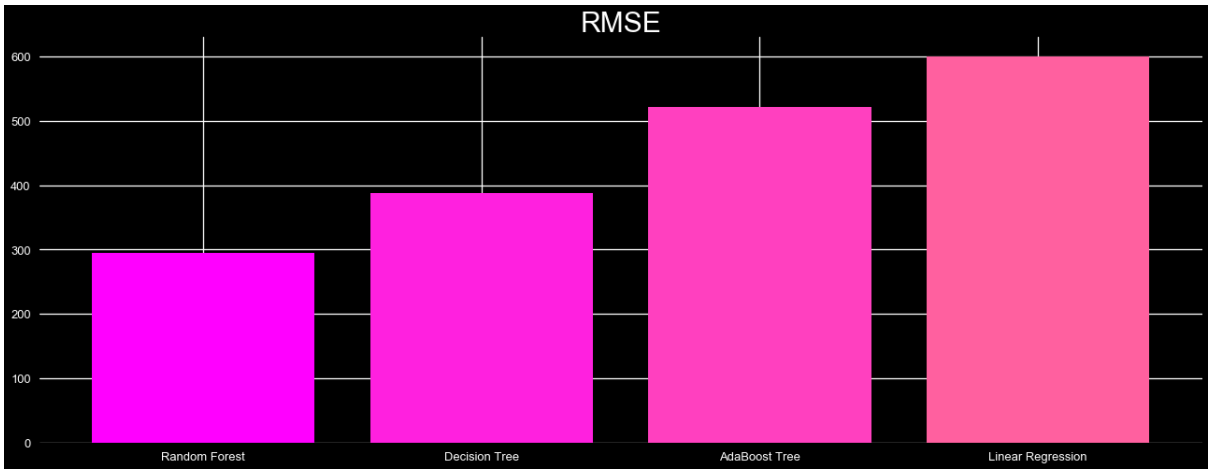
Gender V/S suicides:



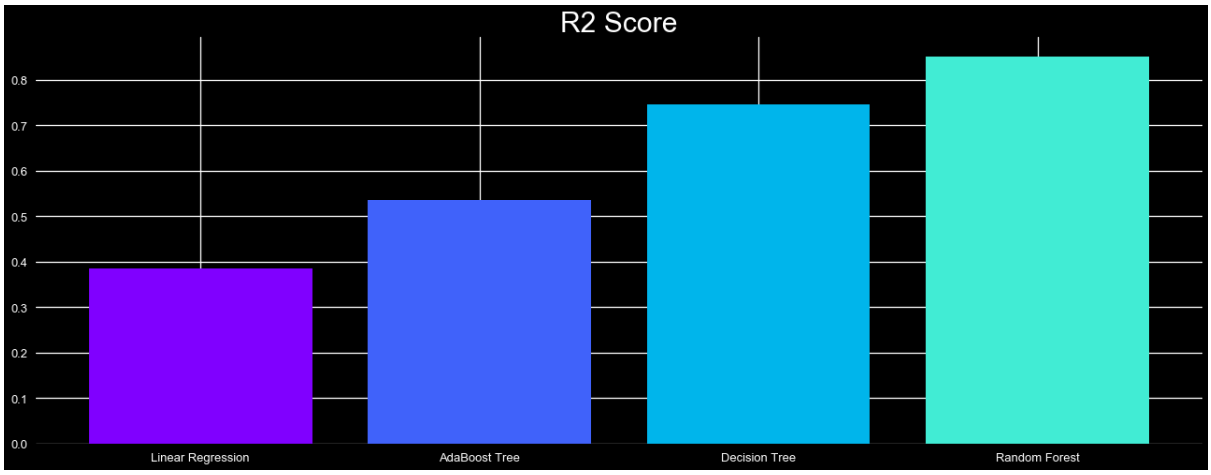
The above figure shows the suicides with respect to the gender. We can see clearly that the females committed more suicides than the males.

The machine learning models:

We applied some machine learning models like: linear regression, random forest, ada boost, decision tree on the dataset, and predicted some values. Then, we calculated the root mean squared errors and the r2 scores of our models. Let us do the comparison of these:



The above graph shows the comparison of the root mean squared error of all these models, as we can see that the rmse of random forest is minimum and that of linear regression is maximum. Now, let us see the comparison of r2 scores:



Here, the random forest model is showing the highest r2 score among the all four, So, we can say that it is the best model for our prediction.

The entire coding of the above process on out dataset is:

```
get_ipython().system(' pip install plotly')
get_ipython().system(' pip install bubbly')

# for basic operations
import numpy as np
import pandas as pd

# for visualizations
import matplotlib.pyplot as plt
import seaborn as sns

# for interactive visualizations
import plotly.offline as py
from plotly.offline import init_notebook_mode, iplot
init_notebook_mode(connected=True)
import plotly.graph_objs as go
import plotly.offline as offline
offline.init_notebook_mode()
from plotly import tools
import plotly.figure_factory as ff

from bubbly.bubbly import bubbleplot

import plotly.tools as tls
import squarify

from numpy import array
from matplotlib import cm

# for providing path
import os
data=pd.read_csv(r"C:\Users\Akash\Desktop\kartik\python\who_suicide_statistics.csv")


data = data.sort_values(['year'], ascending = True)

print(data.shape)




# let's check the total number of countries' data available for suicidal analysis

print("No. of Countries available for analysis :", data['country'].nunique())




# checking the head of the table
```

```
dat = ff.create_table(data.head())
py.iplot(dat)
```

```
data.info()
```

```
data.describe()
```

```
# let's describe the data

dat = ff.create_table(data.describe(),index=True)
py.iplot(dat)
```

```
# renaming the columns

data.rename({'sex' : 'gender', 'suicides_no' : 'suicides'}, inplace = True, axis = 1)

data.columns
```

```
# checkinng the null values in the dataset

data.isnull().sum()
```

```
# In[ ]:
```

```
# filling missing values

data['suicides'].fillna(0, inplace = True)
# data['population'].mean()
data['population'].fillna(1664090, inplace = True)

# checking if there is any null value left
data.isnull().sum().sum()
```

```
# converting these attributes into integer format
data['suicides'] = data['suicides'].astype(int)
data['population'] = data['population'].astype(int)


# In[ ]:


data.head()


# In[ ]:


import warnings
warnings.filterwarnings('ignore')

figure = bubbleplot(dataset = data, x_column = 'suicides', y_column ='population',
    bubble_column = 'country',  color_column = 'country',
    x_title = "Number of Suicides", y_title = "Population", title = 'Population vs Suicides',
    x_logscale = False, scale_bubble = 3, height = 650)

py.iplot(figure, config={'scrollzoom': True})




# visualising the different countries distribution in the dataset

plt.style.use('seaborn-dark')
plt.rcParams['figure.figsize'] = (15, 9)

color = plt.cm.winter(np.linspace(0, 10, 100))
x = pd.DataFrame(data.groupby(['country'])['suicides'].sum().reset_index())
x.sort_values(by = ['suicides'], ascending = False, inplace = True)

sns.barplot(x['country'].head(10), y = x['suicides'].head(10), data = x, palette = 'winter')
plt.title('Top 10 Countries in Suicides', fontsize = 20)
plt.xlabel('Name of Country')
plt.xticks(rotation = 90)
plt.ylabel('Count')
plt.show()




# visualising the different year distribution in the dataset

plt.style.use('seaborn-dark')
plt.rcParams['figure.figsize'] = (18, 9)
```

```
x = pd.DataFrame(data.groupby(['year'])['suicides'].sum().reset_index())
x.sort_values(by = ['suicides'], ascending = False, inplace = True)

sns.barplot(x['year'], y = x['suicides'], data = x, palette = 'cool')
plt.title('Distribution of suicides from the year 1985 to 2016', fontsize = 20)
plt.xlabel('year')
plt.xticks(rotation = 90)
plt.ylabel('count')
plt.show()



color = plt.cm.Blues(np.linspace(0, 1, 2))
data['gender'].value_counts().plot.pie(colors = color, figsize = (10, 10), startangle = 75)

plt.title('Gender', fontsize = 20)
plt.axis('off')
plt.show()



# visualising the different year distribution in the dataset

plt.style.use('seaborn-dark')
plt.rcParams['figure.figsize'] = (18, 9)

x = pd.DataFrame(data.groupby(['gender'])['suicides'].sum().reset_index())
x.sort_values(by = ['suicides'], ascending = False, inplace = True)

sns.barplot(x['gender'], y = x['suicides'], data = x, palette = 'afmhot')
plt.title('Distribution of suicides wrt Gender', fontsize = 20)
plt.xlabel('gender')
plt.xticks(rotation = 90)
plt.ylabel('count')
plt.show()



suicide = pd.DataFrame(data.groupby(['country','year'])['suicides'].sum().reset_index())

count_max_sui=pd.DataFrame(suicide.groupby('country')['suicides'].sum().reset_index())

count = [ dict(
        type = 'choropleth',
        locations = count_max_sui['country'],
        locationmode='country names',
        z = count_max_sui['suicides'],
        text = count_max_sui['country'],
        colorscale = 'Cividis',
        autocolorscale = False,
```

```
        reversescale = True,
        marker = dict(
            line = dict (
                color = 'rgb(180,180,180)',
                width = 0.5
            ) ),
)]
layout = dict(
    title = 'Suicides happening across the Globe',
    geo = dict(
        showframe = True,
        showcoastlines = True,
        projection = dict(
            type = 'orthographic'
        )
    )
)
fig = dict( data=count, layout=layout )
iplot(fig, validate=False, filename='d3-world-map')

# looking at the Suicides in USA.

data[data['country'] == 'United States of America'].sample(20)


# replacing categorical values in the age column

data['age'] = data['age'].replace('5-14 years', 0)
data['age'] = data['age'].replace('15-24 years', 1)
data['age'] = data['age'].replace('25-34 years', 2)
data['age'] = data['age'].replace('35-54 years', 3)
data['age'] = data['age'].replace('55-74 years', 4)
data['age'] = data['age'].replace('75+ years', 5)

#data['age'].value_counts()

# suicides in different age groups

x1 = data[data['age'] == 0]['suicides'].sum()
x2 = data[data['age'] == 1]['suicides'].sum()
x3 = data[data['age'] == 2]['suicides'].sum()
x4 = data[data['age'] == 3]['suicides'].sum()
x5 = data[data['age'] == 4]['suicides'].sum()
x6 = data[data['age'] == 5]['suicides'].sum()

x = pd.DataFrame([x1, x2, x3, x4, x5, x6])
x.index = ['5-14', '15-24', '25-34', '35-54', '55-74', '75+']
x.plot(kind = 'bar', color = 'grey')

plt.title('suicides in different age groups')
plt.xlabel('Age Group')
```

```
plt.ylabel('count')
plt.show()
df = data.groupby(['country', 'year'])['suicides'].mean()
df = pd.DataFrame(df)

# looking at the suicides trends for any 3 countries
plt.rcParams['figure.figsize'] = (20, 30)
plt.style.use('dark_background')

plt.subplot(3, 1, 1)
color = plt.cm.hot(np.linspace(0, 1, 40))
df['suicides']['United States of America'].plot.bar(color = color)
plt.title('Suicides Trends in USA wrt Year', fontsize = 30)

plt.subplot(3, 1, 2)
color = plt.cm.spring(np.linspace(0, 1, 40))
df['suicides']['Russian Federation'].plot.bar(color = color)
plt.title('Suicides Trends in Russian Federation wrt Year', fontsize = 30)

plt.subplot(3, 1, 3)
color = plt.cm.PuBu(np.linspace(0, 1, 40))
df['suicides']['Japan'].plot.bar(color = color)
plt.title('Suicides Trends in Japan wrt Year', fontsize = 30)

plt.show()

df2 = data.groupby(['country', 'age'])['suicides'].mean()
df2 = pd.DataFrame(df2)

# looking at the suicides trends for any 3 countries
plt.rcParams['figure.figsize'] = (20, 30)

plt.subplot(3, 1, 1)
df2['suicides']['United States of America'].plot.bar()
plt.title('Suicides Trends in USA wrt Age Groups', fontsize = 30)
plt.xticks(rotation = 0)

plt.subplot(3, 1, 2)
color = plt.cm.jet(np.linspace(0, 1, 6))
df2['suicides']['Russian Federation'].plot.bar(color = color)
plt.title('Suicides Trends in Russian Federation wrt Age Groups', fontsize = 30)
plt.xticks(rotation = 0)

plt.subplot(3, 1, 3)
color = plt.cm.Wistia(np.linspace(0, 1, 6))
df2['suicides']['Japan'].plot.bar(color = color)
plt.title('Suicides Trends in Japan wrt Age Groups', fontsize = 30)
plt.xticks(rotation = 0)

plt.show()
```

```
plt.rcParams['figure.figsize'] = (15, 7)
plt.style.use('dark_background')

sns.stripplot(data['year'], data['suicides'], palette = 'cool')
plt.title('Year vs Suicides', fontsize = 20)
plt.xticks(rotation = 90)
plt.show()

# age-group vs suicides

plt.rcParams['figure.figsize'] = (15, 7)


sns.stripplot(data['gender'], data['suicides'], palette = 'Wistia')
plt.title('Gender vs Suicides', fontsize = 20)
plt.grid()
plt.show()
# label encoding for gender

from sklearn.preprocessing import LabelEncoder

# creating an encoder
le = LabelEncoder()
data['gender'] = le.fit_transform(data['gender'])

data['gender'].value_counts()
# deleting unnecassary column

data = data.drop(['country'], axis = 1)

data.columns
#splitting the data into dependent and independent variables

x = data.drop(['suicides'], axis = 1)
y = data['suicides']

print(x.shape)
print(y.shape)
# splitting the dataset into training and testing sets

from sklearn.model_selection import train_test_split

x_train, x_test, y_train, y_test = train_test_split(x, y, test_size = 0.25, random_state = 45)

print(x_train.shape)
print(y_train.shape)
print(x_test.shape)
print(y_test.shape)
import warnings
warnings.filterwarnings('ignore')
```

```
# importing the min max scaler
from sklearn.preprocessing import MinMaxScaler

# creating a scaler
mm = MinMaxScaler()

# scaling the independent variables
x_train = mm.fit_transform(x_train)
x_test = mm.transform(x_test)
from sklearn.linear_model import LinearRegression
from sklearn.metrics import r2_score

# creating the model
model = LinearRegression()

# feeding the training data into the model
model.fit(x_train, y_train)

# predicting the test set results
y_pred = model.predict(x_test)

# calculating the mean squared error
mse = np.mean((y_test - y_pred)**2)
print("MSE :", mse)
rmse = np.sqrt(mse)
print("RMSE :", rmse)
r2 = r2_score(y_test, y_pred)
print("r2_score :", r2)
from sklearn.ensemble import RandomForestRegressor
model = RandomForestRegressor()
model.fit(x_train, y_train)
y_pred = model.predict(x_test)
mse = np.mean((y_test - y_pred)**2)
print("MSE :", mse)
rmse = np.sqrt(mse)
print("RMSE :", rmse)
r2 = r2_score(y_test, y_pred)
print("r2_score :", r2)
from sklearn.tree import DecisionTreeRegressor
model = DecisionTreeRegressor()
model.fit(x_train, y_train)
y_pred = model.predict(x_test)
mse = np.mean((y_test - y_pred)**2)
print("MSE :", mse)
rmse = np.sqrt(mse)
print("RMSE :", rmse)
r2 = r2_score(y_test, y_pred)
print("r2_score :", r2)
from sklearn.ensemble import AdaBoostRegressor
model = AdaBoostRegressor()
model.fit(x_train, y_train)
```

```
y_pred = model.predict(x_test)
mse = np.mean((y_test - y_pred)**2)
print("MSE :", mse)
rmse = np.sqrt(mse)
print("RMSE :", rmse)
r2 = r2_score(y_test, y_pred)
print("r2_score :", r2)
r2_score = np.array([0.385, 0.851, 0.745, 0.535])
labels = np.array(['Linear Regression', 'Random Forest', 'Decision Tree', 'AdaBoost Tree'])
indices = np.argsort(r2_score)
color = plt.cm.rainbow(np.linspace(0, 1, 9))
plt.style.use('seaborn-talk')
plt.rcParams['figure.figsize'] = (18, 7)
plt.bar(range(len(indices)), r2_score[indices], color = color)
plt.xticks(range(len(indices)), labels[indices])
plt.title('R2 Score', fontsize = 30)
plt.grid()
plt.tight_layout()
plt.show()
rmse = np.array([600, 295, 388, 521])
labels = np.array(['Linear Regression', 'Random Forest', 'Decision Tree', 'AdaBoost Tree'])
indices = np.argsort(rmse)
color = plt.cm.spring(np.linspace(0, 1, 9))
plt.style.use('seaborn-talk')
plt.rcParams['figure.figsize'] = (18, 7)
plt.bar(range(len(indices)), rmse[indices], color = color)
plt.xticks(range(len(indices)), labels[indices])
plt.title('RMSE', fontsize = 30)
plt.grid()
plt.tight_layout()
plt.show()
```