Suicide Mortality Estimation Using Data Mining Techniques

GROUP 5

PRESENTED BY
Harshitha Mohanraj Radhika
Fnu Maria Poulose,
Divya Nalam,
Sajit Valiya Kizhakke.

INTRODUCTION

Purpose: Suicide is a serious public health problem and identifying risk factors and trends is critical to avoiding suicide fatalities. Gain insights into the risk variables that contribute to suicide mortality and create a prediction model that can be used to identify people at high risk of suicide ideation.

The goal of this project is to use data mining techniques to forecast suicide mortalities and to identify risk behaviors that lead to death.

Dataset: dataset sourced from WHO, World Bank, and UNDP and already published on Kaggle.

Relevance: The findings of this study will assist public health authorities and healthcare providers create effective treatments to lessen the burden of suicide mortality.

Loading the Dataset

```
#Reading the input data
suicide_prevention_df=pd.read_csv(path)
#Check the data
suicide_prevention_df.head(10)
```

	country	year	sex	age	suicides_no	population	suicides/100k pop	country-year	HDI for year	gdp_for_year ($)	gdp_per_capita ($)	generation
0	Albania	1987	male	15-24 years	21	312900	6.71	Albania1987	NaN	2,156,624,900	796	Generation X
1	Albania	1987	male	35-54 years	16	308000	5.19	Albania1987	NaN	2,156,624,900	796	Silent
2	Albania	1987	female	15-24 years	14	289700	4.83	Albania1987	NaN	2,156,624,900	796	Generation X
3	Albania	1987	male	75+ years	1	21800	4.59	Albania1987	NaN	2,156,624,900	796	G.I. Generation
4	Albania	1987	male	25-34 years	9	274300	3.28	Albania1987	NaN	2,156,624,900	796	Boomers
5	Albania	1987	female	75+ years	1	35600	2.81	Albania1987	NaN	2,156,624,900	796	G.I. Generation
6	Albania	1987	female	35-54 years	6	278800	2.15	Albania1987	NaN	2,156,624,900	796	Silent
7	Albania	1987	female	25-34 years	4	257200	1.56	Albania1987	NaN	2,156,624,900	796	Boomers
8	Albania	1987	male	55-74 years	1	137500	0.73	Albania1987	NaN	2,156,624,900	796	G.I. Generation
9	Albania	1987	female	5-14 years	0	311000	0.00	Albania1987	NaN	2,156,624,900	796	Generation X

Familiarizing with data

```
suicide_prevention_df.describe()
```

	year	suicides_no	population	suicides/100k pop	HDI for year	gdp_per_capita ($)
count	27820.000000	27820.000000	2.782000e+04	27820.000000	8364.000000	27820.000000
mean	2001.258375	242.574407	1.844794e+06	12.816097	0.776601	16866.464414
std	8.469055	902.047917	3.911779e+06	18.961511	0.093367	18887.576472
min	1985.000000	0.000000	2.780000e+02	0.000000	0.483000	251.000000
25%	1995.000000	3.000000	9.749850e+04	0.920000	0.713000	3447.000000
50%	2002.000000	25.000000	4.301500e+05	5.990000	0.779000	9372.000000
75%	2008.000000	131.000000	1.486143e+06	16.620000	0.855000	24874.000000
max	2016.000000	22338.000000	4.380521e+07	224.970000	0.944000	126352.000000

```
suicide_prevention_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 27820 entries, 0 to 27819
Data columns (total 12 columns):
 #   Column               Non-Null Count  Dtype
---  ------               --------------  -----
 0   country              27820 non-null  object
 1   year                 27820 non-null  int64
 2   sex                  27820 non-null  object
 3   age                  27820 non-null  object
 4   suicides_no          27820 non-null  int64
 5   population           27820 non-null  int64
 6   suicides/100k pop    27820 non-null  float64
 7   country-year         27820 non-null  object
 8   HDI for year         8364 non-null   float64
 9    gdp_for_year ($)    27820 non-null  object
 10  gdp_per_capita ($)   27820 non-null  int64
 11  generation           27820 non-null  object
dtypes: float64(2), int64(4), object(6)
memory usage: 2.5+ MB
```

Cont..

```
#Listing the features of the dataset
suicide_prevention_df.columns
```

```
Index(['country', 'year', 'sex', 'age', 'suicides_no', 'population',
       'suicides/100k pop', 'country-year', 'HDI for year',
       ' gdp_for_year ($) ', 'gdp_per_capita ($)', 'generation'],
      dtype='object')
```

```
#Shape of dataframe
suicide_prevention_df.shape
```

```
(27820, 12)
```

Cont..

```
#Renaming the columns for easy readability

suicide_prevention_df.columns = ['country', 'year', 'gender', 'age_group', 'suicide_count', 'population', 'suicide_rate', 'country-year', 'HDI for year',
                'gdp_for_year', 'gdp_per_capita', 'generation']
suicide_prevention_df.columns
```

```
Index(['country', 'year', 'gender', 'age_group', 'suicide_count', 'population',
       'suicide_rate', 'country-year', 'HDI for year', 'gdp_for_year',
       'gdp_per_capita', 'generation'],
      dtype='object')
```

```
suicide_prevention_df.age_group.value_counts()
```

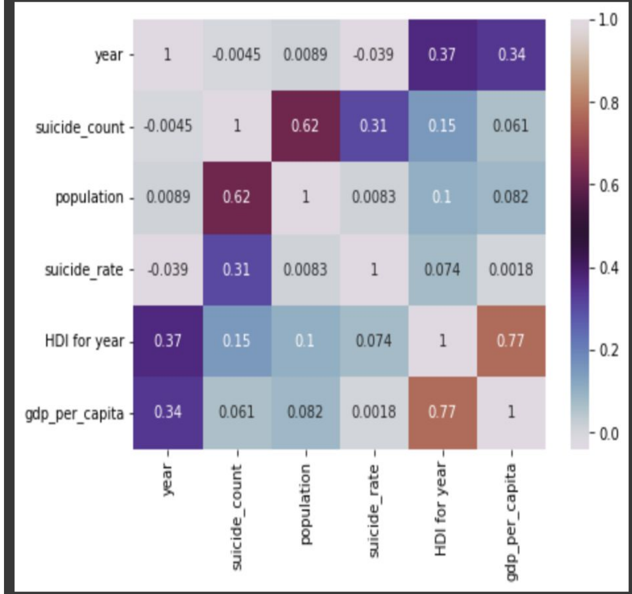
```
15-24 years    4642
35-54 years    4642
75+ years      4642
25-34 years    4642
55-74 years    4642
5-14 years     4610
Name: age_group, dtype: int64
```

```
suicide_prevention_df.generation.value_counts()
```

```
Generation X       6408
Silent             6364
Millenials         5844
Boomers            4990
G.I. Generation    2744
Generation Z       1470
Name: generation, dtype: int64
```

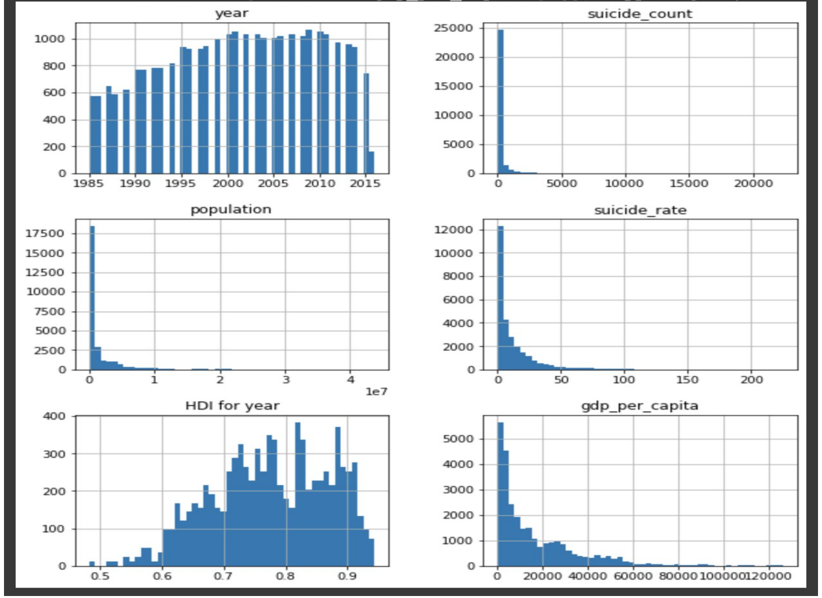
Visualizing the data

```python
# Correlation
plt.figure(figsize=(7,5))
sns.heatmap(suicide_prevention_df.corr(), annot=True, cmap='twilight')
plt.show()
```



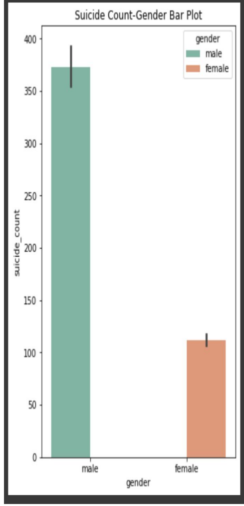

```python
suicide_prevention_df.hist(bins = 50,figsize = (10,11))
```

```
array([[<Axes: title={'center': 'year'}>,
        <Axes: title={'center': 'suicide_count'}>],
       [<Axes: title={'center': 'population'}>,
        <Axes: title={'center': 'suicide_rate'}>],
       [<Axes: title={'center': 'HDI for year'}>,
        <Axes: title={'center': 'gdp_per_capita'}>]], dtype=object)
```

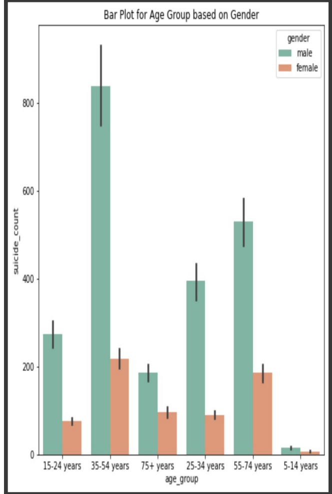


Cont..

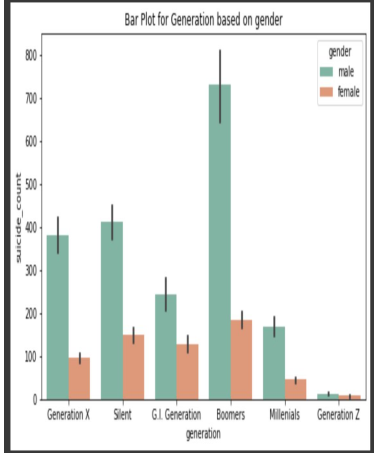
```
#Bar plot to show the gender and suicide count
plt.figure(figsize=(5,8))
sns.barplot(x="gender", y="suicide_count", hue="gender", data=suicide_prevention_df, palette="Set2")
plt.title('Bar plot for gender and suicide count')
plt.show()
```




```
#Bar Plot for age group based on gender
plt.figure(figsize=(8,8))
sns.barplot(x = "age_group", y = "suicide_count", hue = "gender", data =suicide_prevention_df,palette="Set2")
plt.title("Bar Plot for Age Group based on Gender")
plt.show()
```



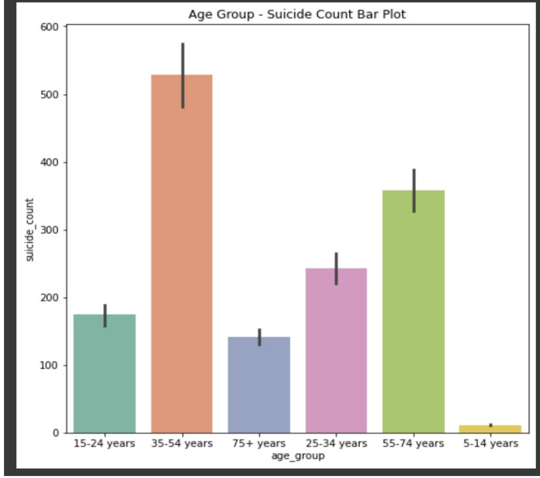

```
#Bar Plot for Generation based on gender
plt.figure(figsize=(9,5))
sns.barplot(x = "generation", y = "suicide_count", hue = "gender", data =suicide_prevention_df,palette="Set2" )
plt.title('Bar Plot for Generation based on gender')
plt.show()
```



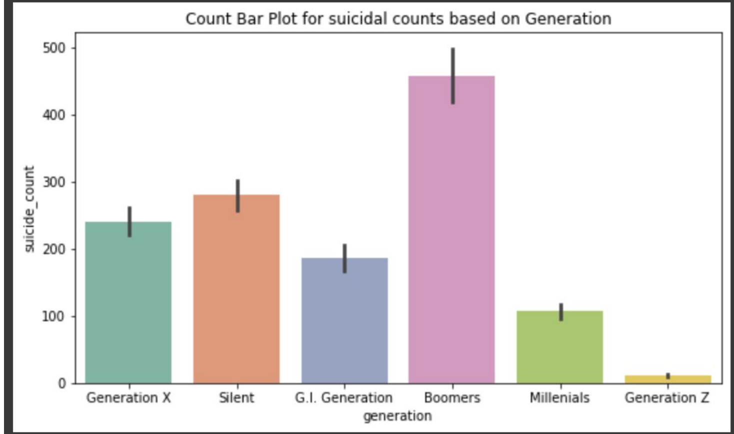
INFERENCE: From the above plots we can see that the count of male commit suicide considerably more than women based on whatever generation, age-group

Cont..

```
# Count bar plot for Age Group and Suicide
plt.figure(figsize=(8,8))
sns.barplot(x='age_group', y='suicide_count',data =suicide_prevention_df,palette="Set2")
plt.title('Age Group - Suicide Count Bar Plot')
plt.show()
```




```
# Count Bar Plot
plt.figure(figsize=(9,5))
sns.barplot(x='generation', y='suicide_count',data =suicide_prevention_df,palette="Set2")
plt.title('Count Bar Plot for suicidal counts based on Generation')
plt.show()
```

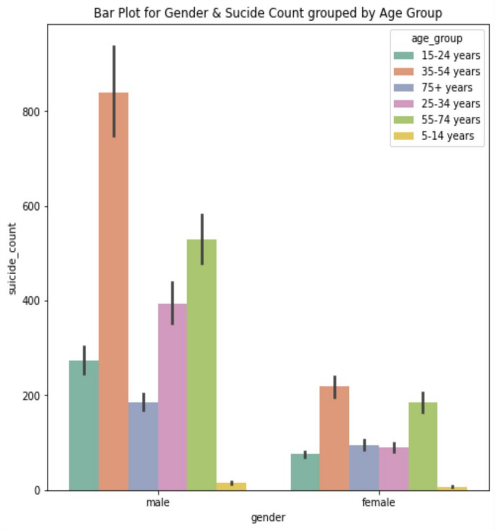


INFERENCE:

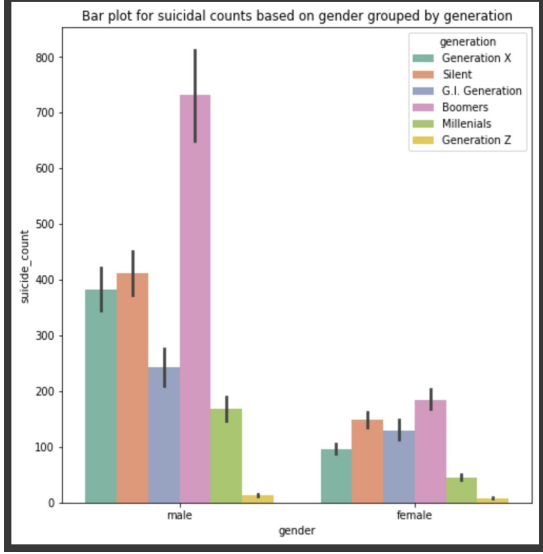
- From 1st graph, Suicide cases are more in the age group of 35-54 years followed by 55- 74 years.
- From the 2nd graph, The next boxplot shows that the suicide cases are more in the boomers, silent and X generations.

Cont..

```
# Bar plot to show the Sucidal Counts based on Gender, grouped by Age Group
plt.figure(figsize=(8,8))
sns.barplot(x="gender", y="suicide_count", hue="age_group",data =suicide_prevention_df,palette="Set2")
plt.title('Bar Plot for Gender & Sucide Count grouped by Age Group')
plt.show()
```



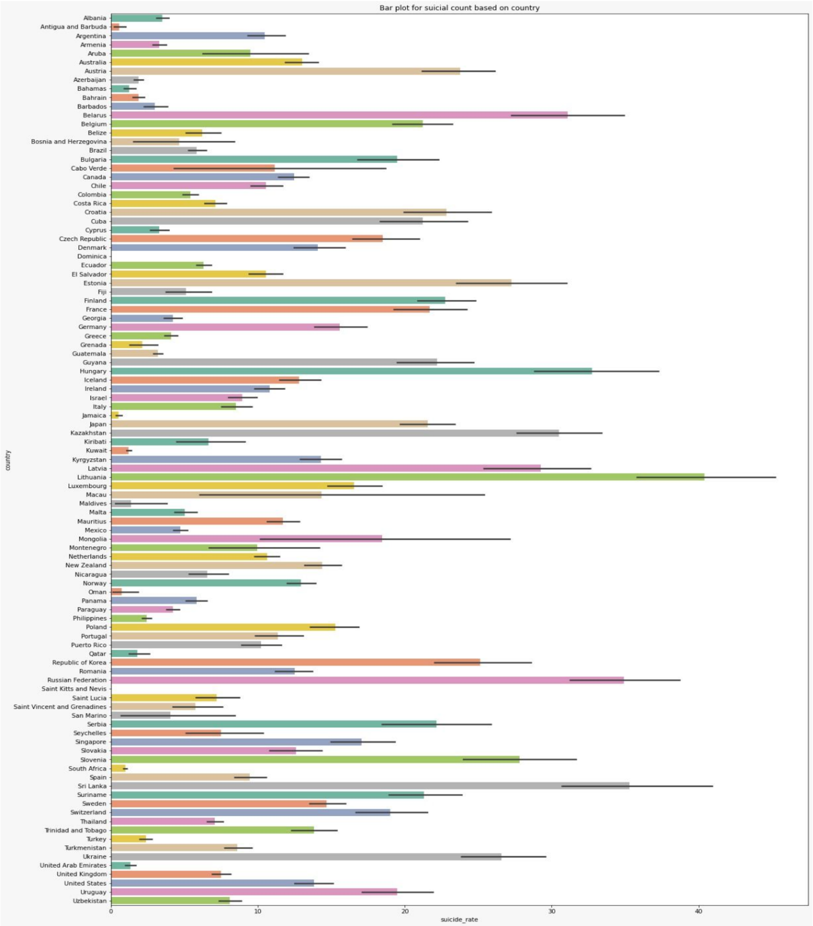

```
#Bar plot for suicidal counts based on gender grouped by generation
plt.figure(figsize=(8,8))
sns.barplot(x="gender", y="suicide_count", hue="generation",data =suicide_prevention_df,palette="Set2")
plt.title('Bar plot for suicidal counts based on gender grouped by generation')
plt.show()
```



INFERENCE:

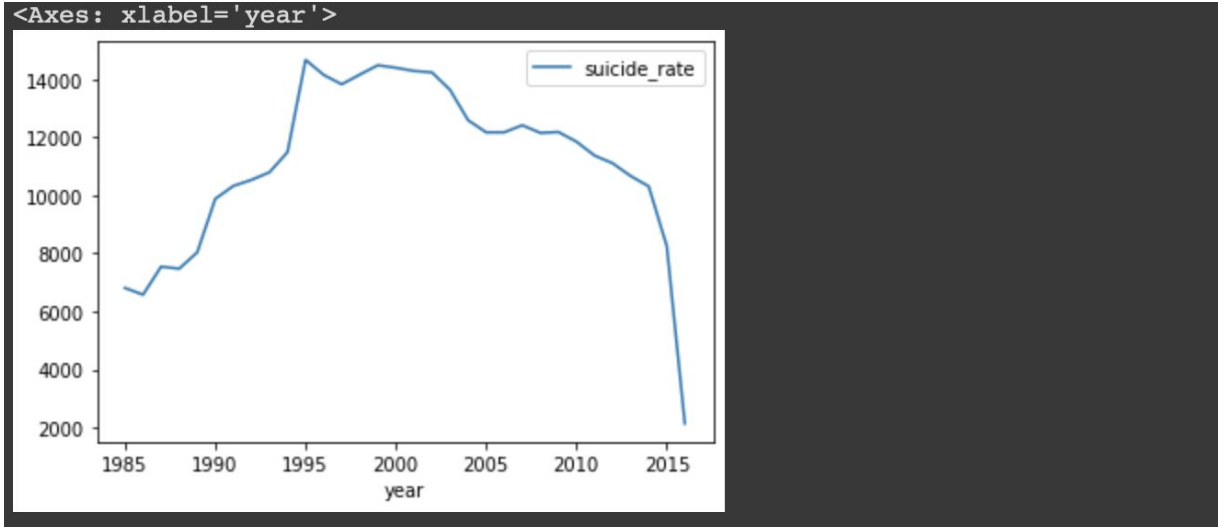
- From the first graph, We can infer that 35-54 years age group is more prone to suicides irrespective of the gender followed by 55-74 years age group and is also obvious that males tend to commit suicide more than female.
- In the second graph, the Bloomers generation had more suicide cases followed by Silent generation irrespective of the gender and even when considered generation, males are more prone to commit suicide.

Cont..



Above bar plot shows that the highest suicide rate country is Lithuania followed by Sri Lanka.

Cont..



The above graph shows the observations that, the suicide rate had grown rapidly from year 1990 & the rate of suicide has drastically reduced in year 2016. The dataset was collected during early 2016. So all the suicide cases of 2016 are not recorded in the dataset

Observations

- We found that HDI for year column has missing values. Since it's an irrelevant column, we are planning to remove the same.
- Based on generation and age group features, Male commit suicide more than women.
- Age feature has 6 unique age groups and Generation feature has 6 types of generations.
- Sex column is categorical so it is encoded in the later steps

Data Preprocessing

```
suicide_prevention_df.nunique()

country              101
year                  32
gender                 2
age_group              6
suicide_count       2084
population         25564
suicide_rate        5298
country-year        2321
HDI for year         305
gdp_for_year        2321
gdp_per_capita      2233
generation             6
dtype: int64
```

```
#checking the data for null or missing values
suicide_prevention_df.isnull().sum()

country               0
year                  0
gender                0
age_group             0
suicide_count         0
population            0
suicide_rate          0
country-year          0
HDI for year      19456
gdp_for_year          0
gdp_per_capita        0
generation            0
dtype: int64
```

There are no null values in any columns other than HDI for year column. There are 19456 null values and they are more than 70% of the values. So its clear we can't use this column as it can impact the performance of the model. We are removing HDI for year column.

Cont..

```
#dropping the HDI for year column
suicide_prevention_df = suicide_prevention_df.drop(['HDI for year'], axis = 1)
suicide_prevention_df.shape
```

```
(27820, 11)
```

```
suicide_prevention_df = suicide_prevention_df.drop(['country-year'], axis = 1)
suicide_prevention_df.shape
```

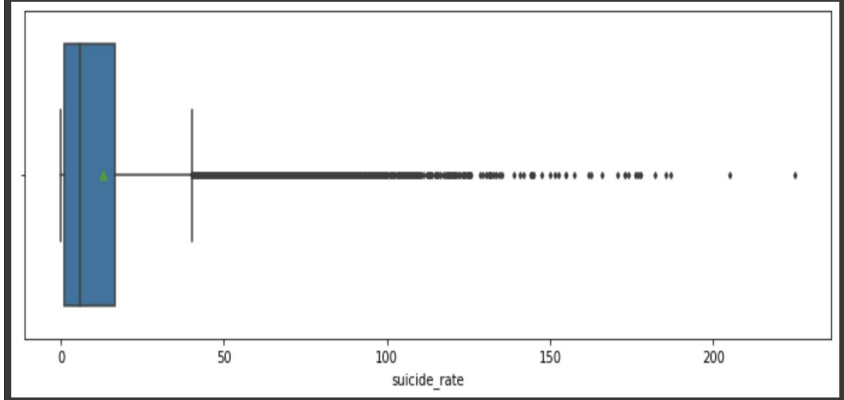
```
(27820, 10)
```

INFERENCE:

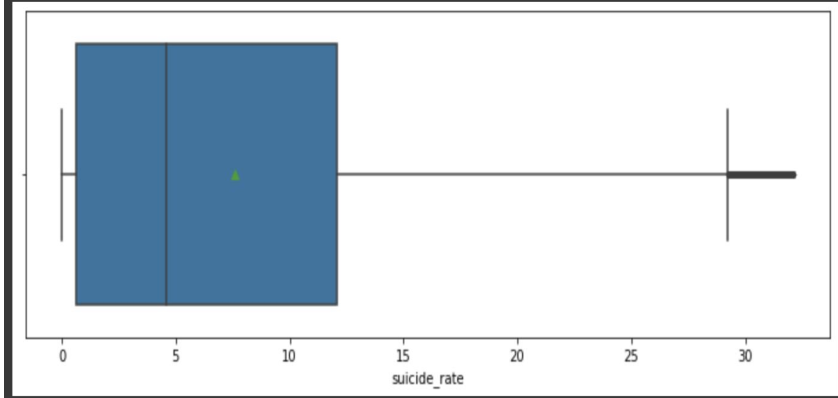
- The country_year column is a combination of the country and year column so the column is dropped
- Now we have 10 features which also includes the target variable

Boxplot Before and After removing outliers

```
fig, ax = plt.subplots(figsize=(12,4))
Add code cell
⌘/Ctrl+M B
ot(x = 'suicide_rate', data = suicide_prevention_df , orient = 'h', width = 0.8,
                 fliersize = 3, showmeans=True, ax = ax)
plt.show()
```




```
fig, ax = plt.subplots(figsize=(12,4))
sns.boxplot(x = 'suicide_rate', data = suicide_prevention_df , orient = 'h', width = 0.8,
                 fliersize = 3, showmeans=True, ax = ax)
plt.show()
```



The outliers are removed

Converting non-numeric columns to numerical

Using sklearn libraries, LabelEncoder we are converting the non-numerical labeled columns like country, year, gender, age_group and generation to numerical labels.

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 24880 entries, 0 to 24879
Data columns (total 10 columns):
 #   Column           Non-Null Count  Dtype
---  ------           --------------  -----
 0   country          24880 non-null  int64
 1   year             24880 non-null  int64
 2   gender           24880 non-null  int64
 3   age_group        24880 non-null  int64
 4   suicide_count    24880 non-null  int64
 5   population       24880 non-null  int64
 6   suicide_rate     24880 non-null  float64
 7   gdp_for_year     24880 non-null  object
 8   gdp_per_capita   24880 non-null  int64
 9   generation       24880 non-null  int64
dtypes: float64(1), int64(8), object(1)
memory usage: 1.9+ MB
```

Cont..

```
# Converting the column 'gdp_for_year' to float from object
data_copy['gdp_for_year'] = data_copy['gdp_for_year'].str.replace(',','').astype(float)
```

INFERENCE:

1. Converting gdp_for_year column to float.

2. Standardization of a dataset is a important sometimes they might behave badly. So we are standardizing few columns using RobustScalar.

Splitting the Dataset

```
#Assigning feature variables and target columns to X & y
X = data_copy['suicide_rate']
Y = data_copy.drop('suicide_rate',axis=1)
X.shape, Y.shape
```

```
((25774,), (25774, 9))
```

Splitting the dataset into train and test sets: 80-20 split

```
# Splitting the dataset into train and test sets: 80-20 split
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, Y, test_size = 0.2, random_state = 42)
X_train.shape,X_test.shape,y_train.shape,y_test.shape
```

```
((20619,), (5155,), (20619, 9), (5155, 9))
```

Completed Steps

1. Loading the data
2. Familiarizing with data
3. Visualizing the data
4. Data Preprocessing & EDA
5. Splitting the data

Models

- We plan to build multiple machine learning models from these features-label pairs, which comprise our training set. Our goal is to make accurate predictions for new, and never-before-seen data.
- There are two major types of supervised machine learning problems, called classification and regression. Our data set comes under regression problem, as the prediction of suicide rate is a continuous number, or a floating-point number in programming terms. The supervised machine learning models (regression) we intend to cover are:
  1) Decision Tree
  2) Random Forest
  3) Gradient Boosting
  4) Support Vector Regression
- The evaluation metrics we will be using are accuracy and rmse.