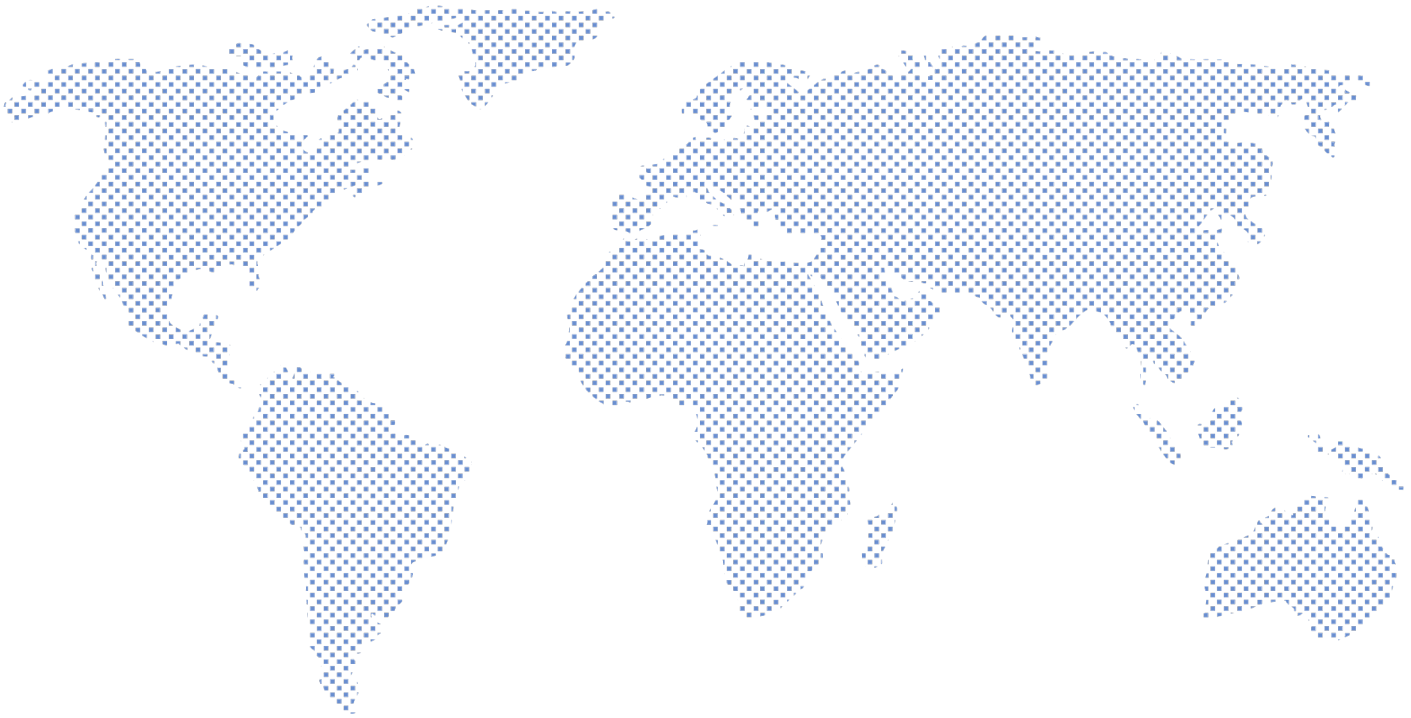
Suicide Rates Overview 1985 to 2016

Compares socio-economic info with suicide rates



Ruozhuo Wang
Sherry Huang

Joyce Jian
Sherry Huang

CONTENT

Overview 1

Data description 1

Data cleaning 2

Methods used 3

Suicide rates and years 4

Suicide rates and age group 6

Suicide rates and gender 8

Suicide rates and country/location 9

Suicide rates and GDP 11

Reference 14

Python codes 15

OVERVIEW

Our project is to uncover trends and patterns in suicide rates in 1985 to 2016. We will examine relationships between suicide rates and age, gender, climate, GDP; trends in suicide rates over the years; and related questions, as the data admits.

DATA DESCRIPTION

This compiled dataset pulled from four other datasets linked by time and place, and was built to find signals correlated to increased suicide rates among different cohorts globally, across the socio-economic spectrum

https://www.kaggle.com/russellyates88/suicide-rates-overview-1985-to-2016.

The datasets includes the following information that we can use to analyze factors associated with suicide rates:

Related factors	Years
	Age Group
	Gender
	Country/Location
	GDP

DATA CLEANING

Country	1985	1986	1987	1988	1989	1990	1991	1992	1993	1994	1995	1996	1997	1998	1999	2000	2001	2002	2003	2004	2005	2006	2007	2008	2009	2010	2011	2012	2013	2014	2015	2016
A	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√															
B			√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√													
C	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√
D								√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√					
E		√	√	√	√	√	√	√	√	√	√	√	√	√	√	√				√	√	√	√									
F		√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√											
G									√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√		
H									√	√		√	√	√				√	√	√	√	√	√	√	√	√	√	√	√			
I		√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√									
J				√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√		
K				√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√
L					√	√	√	√	√	√	√	√	√	√	√	√	√		√	√	√		√									
M		√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√						
N													√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√		

To obtain reliable datasets for further analysis, we performed data cleaning process by doing the following:

- Dropped unnecessary columns in the DataFrame
- Renamed columns to a more recognizable set of labels
- Removed missing and duplicate data
- Set up several bins (groups) for analysis

Analyzing suicide rates VS Years/Age group/Gender/GDP

Since a lot of data are missing in 1985-1989, and 2015-2016 in the raw dataset, we decided to only use data in 1990-2014 from 38 countries for analysis.

Analyzing suicide rate VS Counties/Location

To have the most countries' continuous data for analysis, we decided to use the data in 2010-2014 from 71 countries for analysis.

METHODS USED

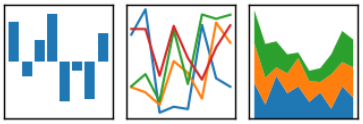
Pandas, Numpy:
Restructured raw data and performed scientific computing for analysis

Matplotlib:
Visualized analysis results and plotted findings using line chart, bar plot, scattered plot and Google Heatmap

API, JSON, Python Requests:
Retrieved coordinates from Google Map using API and imported into the DataFrame

Data Modeling:

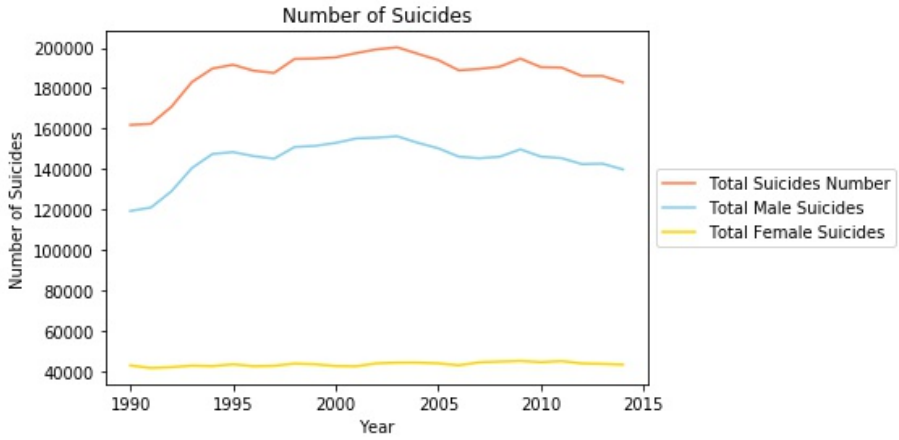
Analyzed output using line of best fit (linear/non-linear regression model) and statistical testing



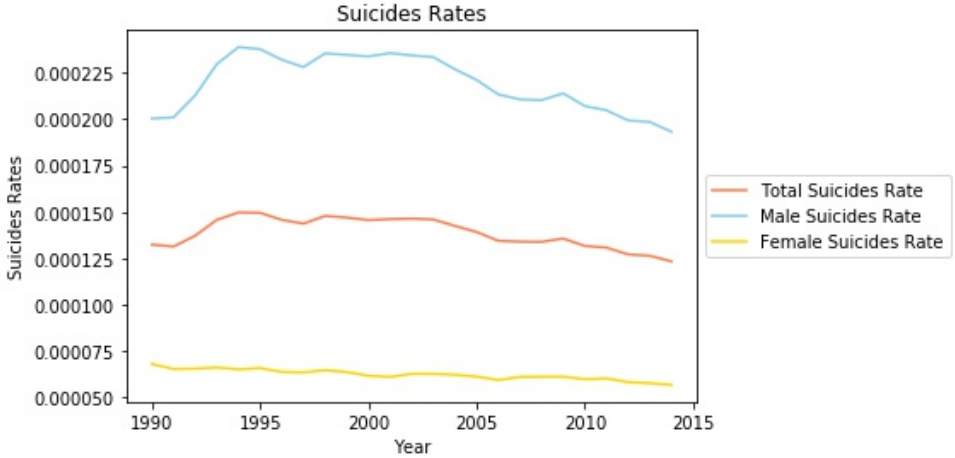
YEARS

Suicide rates and years

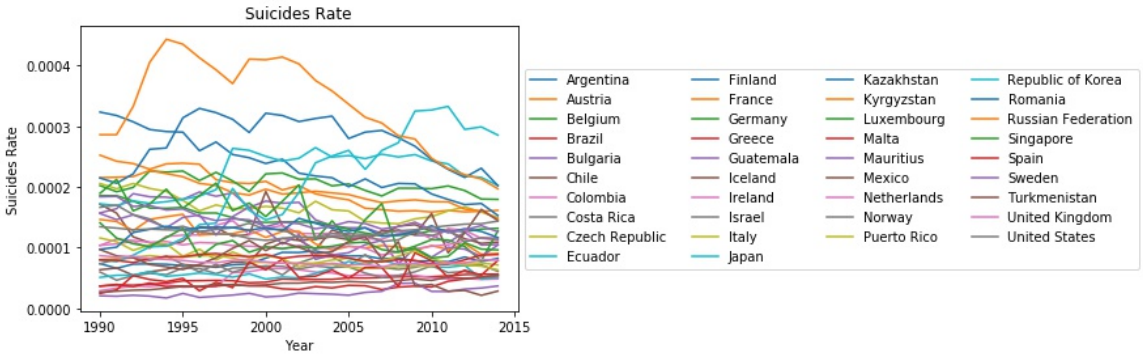
The total number of suicides and number of male suicides fluctuates over the years, but they were all increasing in the long term. However, number of female suicides were almost the same every year.



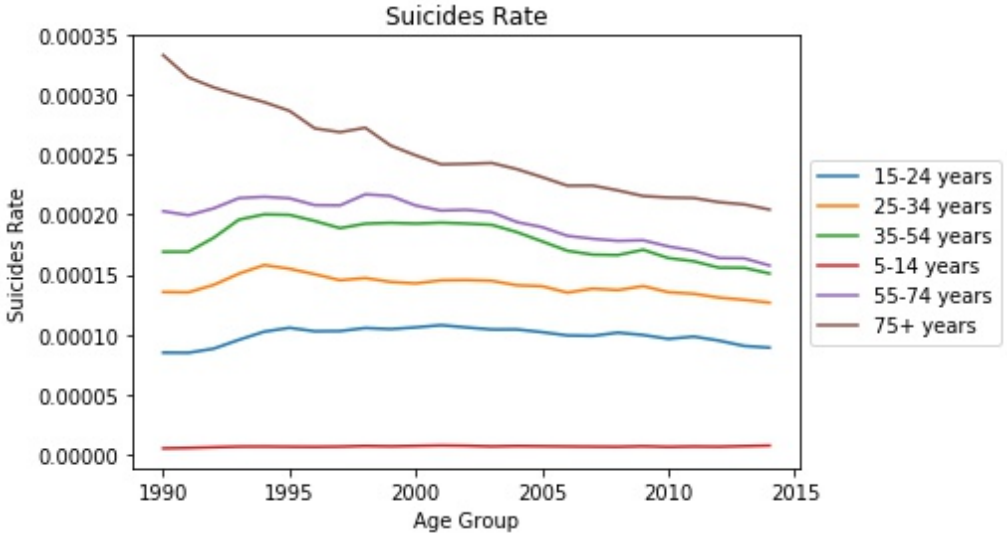
Although the total number of suicides were increasing, the total suicide rates, as well as both gender suicide rates were decreasing from 1990 to 2014. However, the suicide rates of female was relatively consistent over the years. The fluctuation of total suicide rates was mainly caused by suicide rates of male.



YEARS



Among 38 countries in the dataset, there was no clear trend of suicide rates across countries from 1990 to 2014.

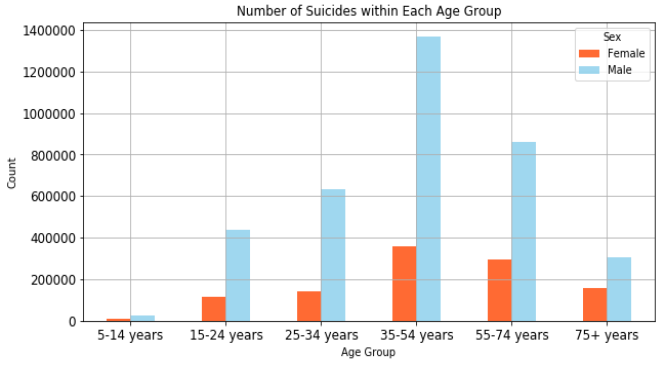


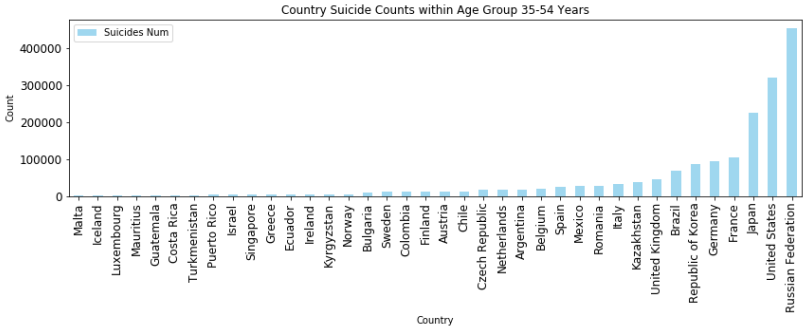
The suicides rate of all age groups were decreasing over 1990-2014. Age group of 75+ years old dropped the most, which might be resulted from improved healthcare and pension system.

AGE GROUP

Suicide rates and age group

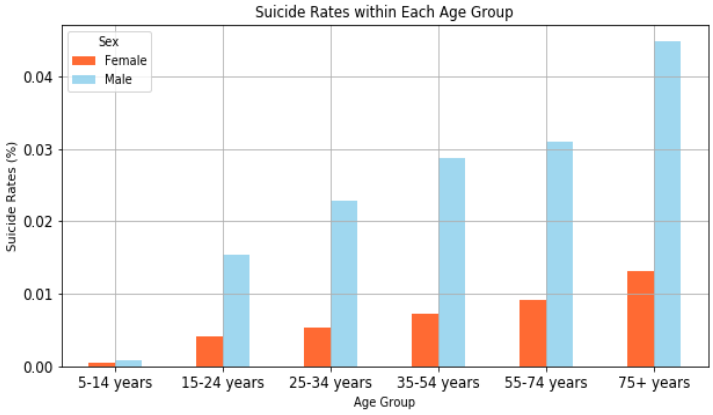
Among all the age groups, female have much less total number of suicides comparing to that of male, and age group of 35-54 years old has the highest number of suicides in both genders. After that, as age increases, the number of suicides decreases. This large number of suicides within middle age group is mainly contributed by the following countries: Japan, United States, and the Russian Federation.

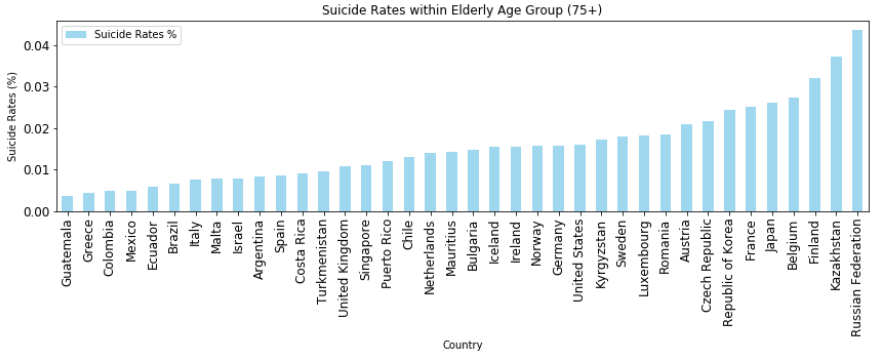




Suicide rates reveal a different trend: the rates are the highest among elderly group (75+ years) for both genders. And the country with the highest elderly suicide rate is still Russia Federation.

AGE GROUP





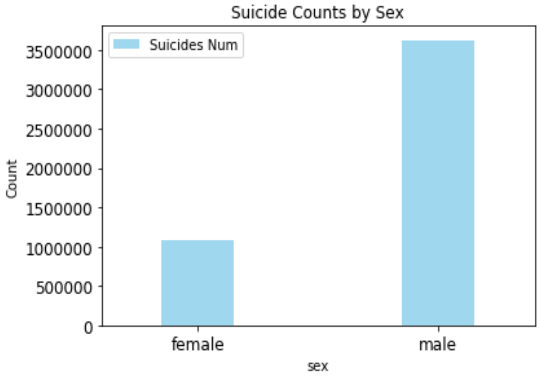
Conclusion

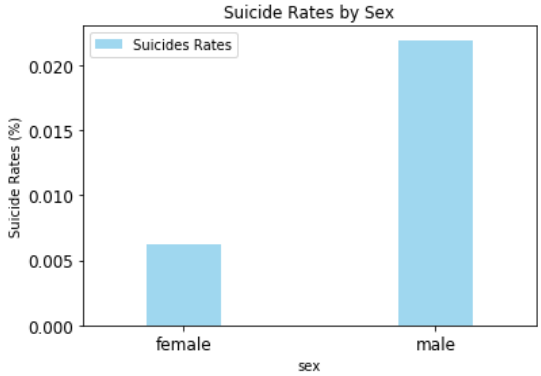
Either from suicide rates or suicide counts perspective, suicide is correlated with age.

GENDER

Suicide rates and gender

Over 1990-2014, the number of suicides for female in all countries is significantly less than that of male (1081921 vs 3625386, *t-test statistics ≈ -18.77, p-value ≈ 1.458e76 ≈ 0)*. This difference is also in suicide rates *(0.006% vs 0.022%, t-test statistics ≈ -51.72, p-value = 0).*





The male to female suicides ratio is highest among 25-34 years old: for each female who committed suicide, there would be 4 male counterparts. Some possible risk factors for high suicides counts within middle age group might be: disrupted marital status, depression, addictive disorder, etc.

sex	female	male	male_to_female_ratio
age			
5-14 years	11519	24501	2
15-24 years	113466	435073	3
25-34 years	143549	631340	4
35-54 years	355754	1368261	3
55-74 years	297580	859581	2
75+ years	160053	306630	1

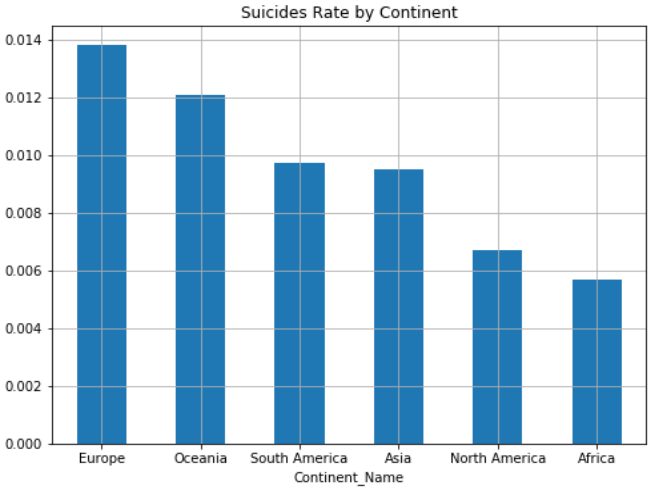
Conclusion

Either from suicide rates or suicide counts perspective, suicide is correlated with gender, with female having significantly less suicides counts comparing to that of male.

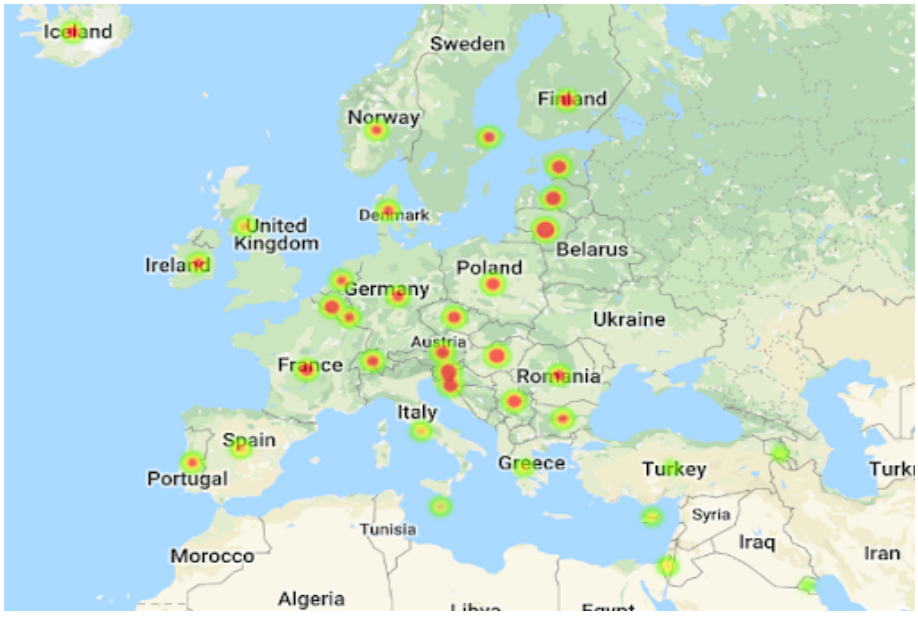
COUNTRY

Suicide rates and country

Continent



Europe suffers the highest rate of suicides (0.0138%) among the six continents, which is more than twice of that of Africa.



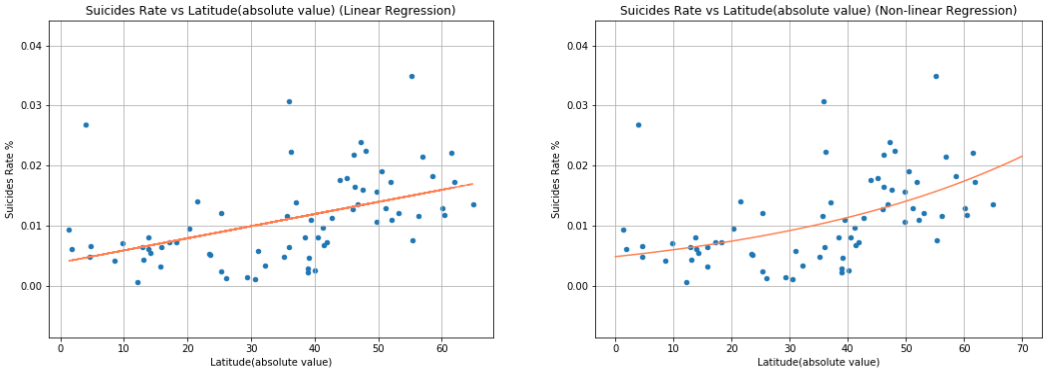
Google Heatmap by Suicide Rate (Europe)

COUNTRY

Latitude (absolute value):

The correlation coefficient between the absolute value of latitude and suicide rate, also known as R-value, is 0.4727, which means they have a moderate positive relationship. In other words, with countries located further away from equator suicide rate sees a remarkable increasing trend.

Comparison using linear and non-linear model



Both linear and non-linear regression analysis have been conducted to describe this trend. Based on the R-value, the exponential model better explains the data. Thus, it can be concluded that the increasing rate also grows with latitude increasing.

- ***Linear model****: Suicide Rate = 0.0002017 * Latitude(abs) + 0.003861*

R-square = 0.2235

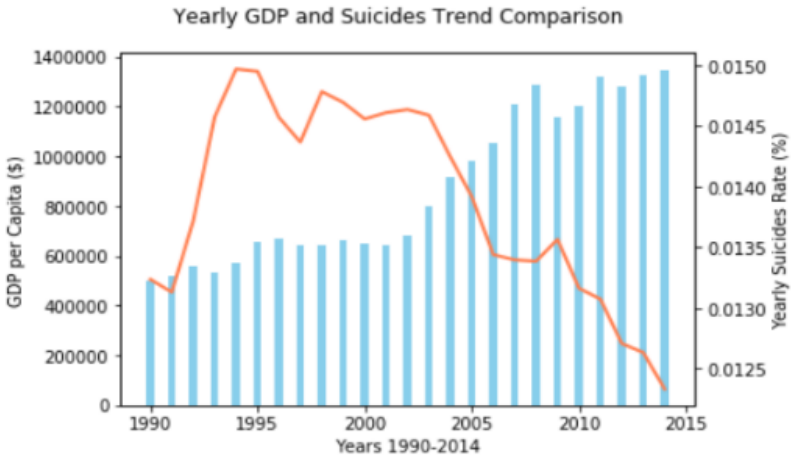
- ***Non-linear model****: Suicide Rate = 0.004839 * exp^(0.02134* Latitude(abs))*

R-square = 0.2492

GDP

Suicide and economic factor (GDP)

Overall trending comparison of suicide and GDP trending of 1990-2014



In general the two series' value trend in an opposite direction over time

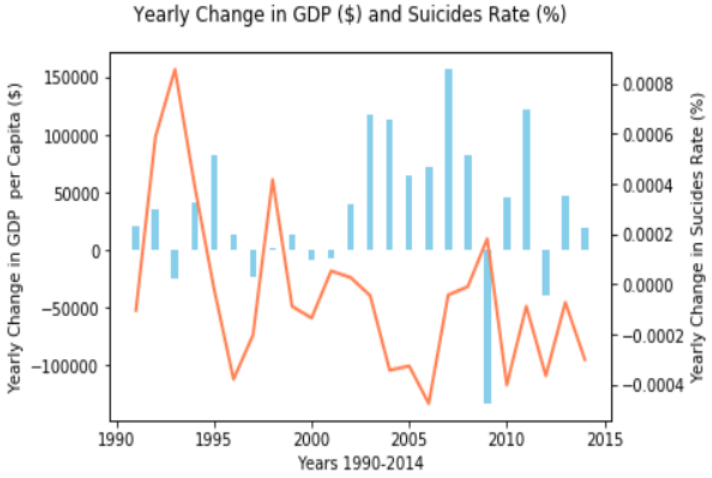
- GDP always increases
- Suicide rate augments first and then shows a continuous downward slope

However yearly data is not consistently matching each other's growth/decrease for the two variables.

Time series difference comparison

To have a better understanding of the evolution of suicide rate and GDP, instead of taking in consideration of each year's value of the two variables directly, we calculate the 1-period-shift difference ($D_1=V_{year2}-V_{year1}$) of the two series and then compare the difference trending of the study year range.

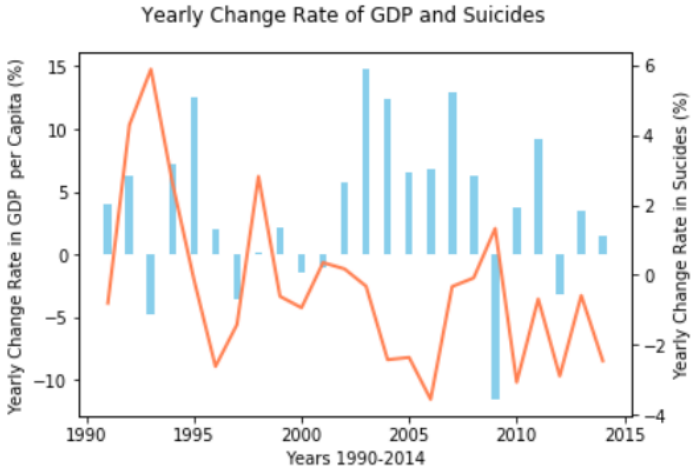
GDP



From the graph representation, the two variables scale very differently and evolution has no matching pattern during the studied period.

Time series difference in % change comparison

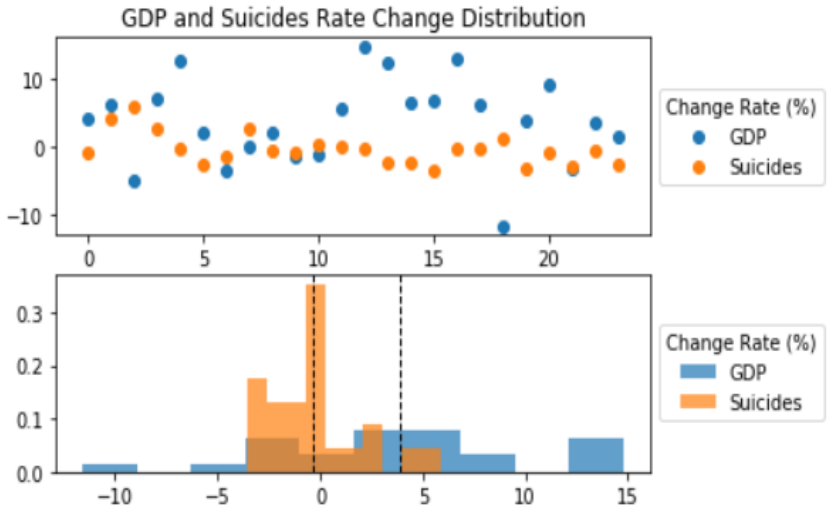
The % change also shows a similar graphic which indicates no significant relationship between the two variables being studied.



GDP

Evolution rate (%) Distribution

We plot the two series of % change rate year over year respectively on a scatterplot and also on a histogram plot.



Both distribution plots tell no similarity, however in order to make sure of the non-existent correlation of suicide rate and GDP figure, we pull out a T-test on the following hypotheses.

Null Hypothesis (H_0): GDP is affecting suicide rates

Alternative Hypothesis (H_1): GDP has no influence on suicide rates

T-test result gives a P value which equals 0.07841685 (>0.05)

Thus statistically saying, the suicides growth is not significantly related to GDP growth to the extent of a confidence level of 95%.

REFERENCE

United Nations Development Program. (2018). Human development index (HDI). http://hdr.undp.org/en/indicators/137506

World Bank. (2018). World development indicators: GDP (current US$) by country:1985 to 2016. http://databank.worldbank.org/data/source/world-development-indicators#

[Szamil]. (2017). *Suicide in the Twenty-First Century* [dataset]. https://www.kaggle.com/szamil/suicide-in-the-twenty-first-century/notebook

World Health Organization. (2018). Suicide prevention. http://www.who.int/mental_health/suicide-prevention/en/

PYTHON CODES

Data cleaning 1990 - 2014

```python
import os
import csv
import pandas as pd
from config import gkey
import gmaps
gmaps.configure(api_key=gkey)
import json
import requests
import numpy as np

file = "Data/master.csv"
suicide_data = pd.read_csv(file, encoding = "ISO-8859-1")
suicide_data.head()

sub_year = suicide_data.loc[(suicide_data["year"]>=1990)&(suicide_data["year"]<=2014),:]
sub_year.head()

country_list = pd.DataFrame(sub_year["country"].value_counts())
country_list = country_list.loc[country_list["country"]== 300,:]
country_list.head()

names = country_list.index

country_chosen = ['Sweden', 'Finland', 'Greece', 'Luxembourg', 'Guatemala', 'Netherlands',
       'Turkmenistan', 'Spain', 'Bulgaria', 'Ecuador', 'Brazil', 'Puerto Rico',
       'Argentina', 'Israel', 'France', 'Germany', 'Russian Federation',
       'Singapore', 'Republic of Korea', 'Iceland', 'Mexico', 'Czech Republic',
       'Romania', 'Chile', 'Mauritius', 'Malta', 'United States', 'Italy',
       'Kazakhstan', 'Belgium', 'Colombia', 'Costa Rica', 'Austria',
       'United Kingdom', 'Ireland', 'Kyrgyzstan', 'Japan', 'Norway']

sub_year = sub_year.set_index("country")
sub_dataset = sub_year.loc[country_chosen,:]
sub_dataset.head()

sub_dataset.to_csv("Data/sub_dataset.csv", encoding = "utf-8", index = True, header = True)
```

PYTHON CODES

Data cleaning 2010 - 2014

```python
# coding: utf-8
import os
import csv
import pandas as pd

#from config import gkey
file = "data/master2.csv"
suicide_data = pd.read_csv(file, encoding = "ISO-8859-1")
suicide_data.head()

group_by_year = suicide_data.groupby(["year"])
country_count = group_by_year["country"].nunique()
country_count

sub_year = suicide_data.loc[(suicide_data["year"]>=2010)&(suicide_data["year"]<=2014),:]
sub_year.head()

country_list = pd.DataFrame(sub_year["country"].value_counts())
country_list = country_list.loc[country_list["country"]== 60,:]
country_list.head()

names = country_list.index

country_chosen = ['Mexico', 'Costa Rica', 'Norway', 'Italy', 'France', 'Argentina',
       'Turkmenistan', 'Cuba', 'Hungary', 'Sweden', 'Australia', 'Croatia',
       'Mauritius', 'Chile', 'Armenia', 'Ireland', 'Grenada', 'Thailand',
       'Ecuador', 'Saint Vincent and Grenadines', 'Cyprus', 'Luxembourg',
       'Seychelles', 'Puerto Rico', 'Belize', 'Bulgaria', 'Brazil',
       'Uzbekistan', 'Serbia', 'Bahrain', 'El Salvador', 'Latvia', 'Belgium',
       'Czech Republic', 'Israel', 'Switzerland', 'Slovenia', 'Austria',
       'Kyrgyzstan', 'United Kingdom', 'Kazakhstan', 'Turkey', 'Qatar',
       'Colombia', 'South Africa', 'Panama', 'Portugal', 'Japan', 'Germany',
       'Republic of Korea', 'Kuwait', 'Lithuania', 'Estonia', 'Iceland',
       'United States', 'Spain', 'Finland', 'Denmark', 'Suriname', 'Poland',
       'Guatemala', 'Georgia', 'Russian Federation', 'Netherlands', 'Romania',
       'Greece', 'Malta', 'Paraguay', 'Nicaragua', 'Saint Lucia', 'Singapore']

sub_year = sub_year.set_index("country")
sub_dataset = sub_year.loc[country_chosen,:]
sub_dataset.head()

sub_dataset.to_csv("Data/sub_dataset_2010-2014.csv", encoding = "utf-8", index = True, header = True)

len(country_chosen)

```

PYTHON CODES

Data analysis

```
## Import Dependencies
import matplotlib.pyplot as plt
import pandas as pd
import numpy as np
import scipy.stats as stats

from config import gkey
import gmaps
gmaps.configure(api_key=gkey)
import json
import requests

from scipy.stats import linregress
from scipy.optimize import curve_fit

# ## Import Cleaned Dataset
data = pd.read_csv("data/sub_dataset.csv")
data.head()

# ## Analyse Year Factor
# Group by Year
groupby_yr=data.groupby("year")
year_sum = groupby_yr["suicides_no"].sum()
year_sum.head()

# Total suicides number and suicides number by Gender Over 1990-2014

male = data.loc[(data["sex"] == "male")]
female = data.loc[(data["sex"] == "female")]

male_sum_yr = male.groupby(['year'])['suicides_no'].sum()
female_sum_yr = female.groupby(['year'])['suicides_no'].sum()

total_population=data.groupby(['year'])['population'].sum()
male_population=male.groupby(['year'])['population'].sum()
female_population=female.groupby(['year'])['population'].sum()

total_rate=year_sum/total_population
male_rate=male_sum_yr/male_population
female_rate=female_sum_yr/female_population

# Plot Number of Suicides by Gender

total_suicides_no = year_sum.plot(kind='line', color="coral",label="Total Suicides Number")
male_sum_yr_plot = male_sum_yr.plot(kind='line', color="skyblue", label="Total Male Suicides")
```

PYTHON CODES

```
female_sum_yr_plot = female_sum_yr.plot(kind='line', color="gold", label='Total Female Suicides')


total_suicides_no.set_xlabel("Year")
total_suicides_no.set_ylabel("Number of Suicides")
plt.title("Number of Suicides")

plt.legend(loc='center left', bbox_to_anchor=(1, 0.5))
plt.savefig("Images/num_suicides.jpg",bbox_inches = "tight")
plt.show()

# Plot Suicide Rates by Gender

total_rate_plot = total_rate.plot(kind='line', color="coral",label="Total Suicides Rate")
male_rate_plot = male_rate.plot(kind='line', color="skyblue", label="Male Suicides Rate")
female_rate_plot = female_rate.plot(kind='line', color="gold", label='Female Suicides Rate')

total_rate_plot.set_xlabel("Year")
total_rate_plot.set_ylabel("Suicides Rates")
plt.title("Suicides Rates")

plt.legend(loc='center left', bbox_to_anchor=(1, 0.5))
plt.savefig("Images/rate_suicides.jpg",bbox_inches = "tight")
plt.show()

# Suicides Rates VS Country Over 1990-2014

country_year=data.groupby(['year','country'])['suicides_no'].sum()
country_population=data.groupby(['year','country'])['population'].sum()
country_rate=country_year/country_population

# Plot Suicide Rates By Country

country_rate_plot= country_rate.unstack().plot(kind='line', figsize = (10,6))

country_rate_plot.set_xlabel("Year")
country_rate_plot.set_ylabel("Suicides Rate")
plt.title("Suicides Rate")

plt.legend(loc='center left', bbox_to_anchor=(1, 0.5),
           ncol=4)
plt.savefig("Images/country_year.jpg",bbox_inches = "tight")
plt.show()

# Specifically look at Suicide Rates in Mexico
```

PYTHON CODES

```
91
92  mexico = data.loc[(data["country"] == "Mexico")]
93  mex_suicides = mexico.groupby(['year'])['suicides_no'].sum()
94  mex_population=mexico.groupby(['year'])['population'].sum()
95
96  mex_rate=mex_suicides/mex_population
97  mex_plot = mex_rate.plot(kind='line', color="gold", label='Mexico Suicides Rate')
98
99  total_suicides_no.set_xlabel("Year")
100 total_suicides_no.set_ylabel("Suicides Rate")
101 plt.title("Mexico Suicides Rate")
102
103 plt.legend(loc='best', bbox_to_anchor=(1, 0.5))
104 plt.savefig("Images/mex_suicides.jpg",bbox_inches='tight')
105 plt.show()
106
107 # Suicide Rates by Age Over 1990-2014
108
109 age_year=data.groupby(['year','age'])['suicides_no'].sum()
110 age_population=data.groupby(['year','age'])['population'].sum()
111 age_rate=age_year/age_population
112
113 # Plot Suicide Rates by Age
114
115 age_rate_plot= age_rate.unstack().plot(kind='line')
116
117 age_rate_plot.set_xlabel("Age Group")
118 age_rate_plot.set_ylabel("Suicides Rate")
119 plt.title("Suicides Rate")
120
121 plt.legend(loc='center left', bbox_to_anchor=(1, 0.5),
122            ncol=1)
123 plt.savefig("Images/age_year.jpg",bbox_inches = "tight")
124 plt.show()
125
126
127 # ## Analyse Age/Gender Factor
128
129 # Group by Age,Gender
130
131 sub_data = data.loc[:,["country","year","sex","age","suicides_no","population","suicides/100k pop"]]
132 groupby_age_sex = sub_data.groupby(["age","sex"])
133 groupby_age_sex.count()
```

PYTHON CODES

```
# Create New DataFrame
suicide_sum = pd.DataFrame({"suicides_sum": groupby_age_sex["suicides_no"].sum()})
# Unstack the DataFrame to create bar chart
suicide_sum = suicide_sum.unstack()
suicide_sum

# Create an index column to sort age group

ind = [2, 3, 4, 1, 5, 6]
suicide_sum["index_col"] = ind
suicide_sum = suicide_sum.sort_values("index_col")['suicides_sum']
suicide_sum

# Plot Number of Suicides by Age Group

suicide_sum.plot(kind="bar", rot = 0, figsize = (10,5),
                 color = ["orangered","skyblue"], fontsize = 12, alpha = 0.8)

plt.legend(["Female","Male"],title="Sex")
plt.title("Number of Suicides within Each Age Group")
plt.xlabel("Age Group")
plt.ylabel("Count")
plt.grid()
plt.savefig("Images/age_group_counts.png", bbox_inches = "tight")
plt.show()

# Suicide rate for Each Age Group

suicide_rate = pd.DataFrame({"suicide_rates_%": groupby_age_sex["suicides_no"].sum()/groupby_age_sex["population"].sum()*100})
suicide_rate = suicide_rate.unstack()

# Sort by Age Group

ind = [2, 3, 4, 1, 5, 6]
suicide_rate["index_col"] = ind
suicide_rate = suicide_rate.sort_values("index_col")["suicide_rates_%"]
suicide_rate

# Plot Suicides Rates by Age Group

suicide_rate.plot(kind="bar", rot = 0, figsize = (10,5),
                  color = ["orangered","skyblue"], fontsize = 12, alpha = 0.8)

plt.legend(["Female","Male"],title="Sex")
plt.title("Suicide Rates within Each Age Group")
```

PYTHON CODES

```
plt.xlabel("Age Group")
plt.ylabel("Suicide Rates (%)")
plt.grid()
plt.savefig("Images/age_group_rates.png", bbox_inches = "tight")
plt.show()

# Number of Suicides by Gender

sex_group = sub_data.groupby(["sex"])
sum_by_sex = pd.DataFrame(sex_group["suicides_no"].sum())
sum_by_sex

# Plot Number of Suicides by Gender

plt.figure()

sum_by_sex.plot(kind = "bar", color = "skyblue", width = 0.3, alpha = 0.8, rot = 0, fontsize = 12)
plt.title("Suicide Counts by Sex")
plt.ylabel("Count")
plt.legend(["Suicides Num"],loc="best")
plt.savefig("Images/sex_group_counts.png", bbox_inches = "tight")
plt.show()

# Suicide Rates by Gender

rate_by_sex = pd.DataFrame({"suicide_rates_%": sex_group["suicides_no"].sum()/sex_group["population"].sum()*100})
rate_by_sex

# Plot Suicide Rates by Gender

plt.figure()

rate_by_sex.plot(kind = "bar", color = "skyblue", width = 0.3, alpha = 0.8, rot = 0, fontsize = 12)
plt.title("Suicide Rates by Sex")
plt.ylabel("Suicide Rates (%)")
plt.legend(["Suicides Rates"],loc="best")
plt.savefig("Images/sex_group_rates.png", bbox_inches = "tight")
plt.show()

# Perform two sample T-Test on female and male data

df_female = sub_data.loc[sub_data["sex"] == "female", :]
df_male = sub_data.loc[sub_data["sex"] == "male", :]

```

PYTHON CODES

```
stats.ttest_ind(df_female["suicides_no"], df_male["suicides_no"], equal_var=False)

stats.ttest_ind(df_female["suicides/100k pop"], df_male["suicides/100k pop"], equal_var=False)

# Male to Female Suicides Ratio

suicide_sum["male_to_female_ratio"] = suicide_sum["male"]//suicide_sum["female"]
suicide_sum

# # Location

# get data from 2010 to 2014
file_2010_2014 = "data/sub_dataset_2010-2014.csv"
data_2010_2014 = pd.read_csv(file_2010_2014, encoding = "ISO-8859-1")
data_2010_2014.head()

# calculate total number of suicides and population of each country during this 5 years
rate_2010_2014 = data_2010_2014.groupby(["country"])
rate_2010_2014 = rate_2010_2014.sum()
rate_2010_2014.head()

# add a column and calculate total suicides rate of each country during this 5 years
rate_2010_2014["suicides_rate_%"] = rate_2010_2014["suicides_no"]/rate_2010_2014["population"]*100
rate_2010_2014 = rate_2010_2014[["suicides_no","population","suicides_rate_%"]]

# add 2 columns for lat and lng
rate_2010_2014["lat"]= ""
rate_2010_2014["lng"]= ""
rate_2010_2014.head()

# Loop through the countries and run a lat/long search for each city
# save lat and lng data to rate_2010_2014
for index, row in rate_2010_2014.iterrows():

    country = index
    target_url = ('https://maps.googleapis.com/maps/api/geocode/json?'
                  'address={0}&key={1}').format(country, gkey)

    try:
        response = requests.get(target_url).json()
        lat = response['results'][0]["geometry"]["location"]["lat"]
        lng = response['results'][0]["geometry"]["location"]["lng"]

        rate_2010_2014.loc[index, "lat"] = lat
        rate_2010_2014.loc[index, "lng"] = lng
```

PYTHON CODES

```
    except :
        print(country)

rate_2010_2014.head()

# reset the index
rate_general_2010_2014 = rate_2010_2014.reset_index()
rate_general_2010_2014.head()

# save the df to csv file
rate_general_2010_2014.to_csv("data/dataset_attd_2010_2014.csv", encoding = "utf-8", index = True, header = True)

# get data from dataset_attd_2010_2014.csv
file_att = "data/dataset_attd_2010_2014.csv"
rate_general_2010_2014 = pd.read_csv(file_att, encoding = "ISO-8859-1")
rate_general_2010_2014.head()

# sort data by suicides rate
rate_general_2010_2014_rank = rate_general_2010_2014.sort_values("suicides_rate_%", ascending=False)
rate_general_2010_2014_rank.head()

# save the top 3 to "top3"
rate_general_2010_2014_rank = rate_general_2010_2014_rank.reset_index(drop = True)
rate_general_2010_2014_rank
top3 = rate_general_2010_2014_rank.iloc[0:3,:]
top3

# save the bottom 3 to "bottom3"
bottom3 = rate_general_2010_2014_rank.iloc[68:71,:]
bottom3

# combine the top3 and bottom 3 as top_bottom
top_bottom = pd.concat([top3,bottom3])
top_bottom

# plot the top 3 and bottom 3
plt.figure(figsize=(10,6))
top_bottom_plot = plt.bar(top_bottom['country'], top_bottom["suicides_rate_%"], color='r', alpha=0.5, align="center")

plt.xlabel("Countries")
plt.ylabel("Suicides Rate %")
plt.title("Top 3 and Bottom 3")

top_bottom_plot[3].set_color('b')
top_bottom_plot[4].set_color('b')
```

PYTHON CODES

```
top_bottom_plot[5].set_color('b')

plt.savefig("Images/Location_top_bottom.png")

# Plot Heatmap
locations_71 = rate_general_2010_2014[["lat", "lng"]]
suicides_rate_71 = rate_general_2010_2014["suicides_rate_%"].astype(float)

fig = gmaps.figure()

# Create heat layer
heat_layer = gmaps.heatmap_layer(locations_71, weights=suicides_rate_71 * 1000000,dissipating=False, max_intensity=10000,point_radius=1)

# Add layer
fig.add_layer(heat_layer)

# read data from "Countries of the world"
file_region_71 = "Data/Countries of the world.xls"
region_71 = pd.read_excel(file_region_71, encoding = "ISO-8859-1")
region_71.head()

# remove space in column "country" of "region_71" and "Country" of "rate_general_2010_2014"
region_71['Country'] = region_71["Country"].str.strip()
rate_general_2010_2014['country'] = rate_general_2010_2014["country"].str.strip()

# merge "region_71" and "rate_general_2010_2014"
data_merged_71 = pd.merge(rate_general_2010_2014,region_71,left_on = "country",right_on = "Country", how = "inner")
data_merged_71.head()

# check which country is left after merge
list1 = data_merged_71["country"].tolist()
list2 = rate_general_2010_2014["country"].tolist()

for x in list2:
    if x not in list1:
        print(x)

# read data from "country-and-continent.csv"
file_continent = "Data/country-and-continent.csv"
continent = pd.read_csv(file_continent, encoding = "ISO-8859-1")
continent.head()

# split the "Country_Name" of "continent" and save to new
new= continent["Country_Name"].str.split(",", n = 1, expand = True)
```

PYTHON CODES

```
# add the first column of "new" to "CountryName" in "continent"
continent["CountryName"]= new[0]
continent.head()

# Merge "continent" with "data_merged_71"
data_complete = pd.merge(data_merged_71,continent,left_on = "country",right_on = "CountryName", how = "inner")

# list the columns of "data_complete"
data_complete.columns

# remove unuseful columns
data_complete = data_complete[['country','CountryName', 'suicides_no', 'population', 'suicides_rate_%', 'lat', 'lng',
       'Country', 'Region', 'Population', 'Area sq. mi.',
       'Pop. Density per sq. mi.', 'Coastline coast/area ratio',
       'Climate', 'Agriculture', 'Industry',
       'Service', 'Continent_Name','Country_Name']]

data_complete.head(3)

# Delete the country that show twice
data_complete = data_complete.drop(data_complete[(data_complete['country'] == "Armenia") & (data_complete['Continent_Name'] == "Asia")].index
data_complete = data_complete.drop(data_complete[(data_complete['country'] == "Cyprus") & (data_complete['Continent_Name'] == "Asia")].index)
data_complete = data_complete.drop(data_complete[(data_complete['country'] == "Turkey") & (data_complete['Continent_Name'] == "Europe")].inde
data_complete = data_complete.drop(data_complete[(data_complete['country'] == "Kazakhstan") & (data_complete['Continent_Name'] == "Europe")].
data_complete = data_complete.drop(data_complete[(data_complete['country'] == "Russian Federation") & (data_complete['Continent_Name'] == "As
data_complete = data_complete.drop(data_complete[(data_complete['country'] == "Georgia") & (data_complete['Continent_Name'] == "Asia")].index

# Check if each country shows only once
data_complete["country"].value_counts()

# Check which country is left after merge
list1 = data_complete["country"].tolist()
list2 = data_merged_71["country"].tolist()

for x in list2:
    if x not in list1:
        print(x)

# save the df to csv file
data_complete.to_csv("Data/data_complete.csv", encoding = "utf-8", index = True, header = True)

# classify countries by continent
continent_count = pd.DataFrame(data_complete["Continent_Name"].value_counts())
continent_count = continent_count.rename(columns={"Continent_Name":"Continent"})
```

PYTHON CODES

```
# Plot the target dataset in terms of which countries are included
continent_plot = continent_count.plot(kind="bar", figsize=(9,6),alpha=0.7)
plt.xticks(rotation = 0)
plt.title("Number of Countries by Continents")
plt

# Save the figure
plt.savefig("Images/Location_countries_analyzed.png")

# Plot bar chart to show mean suicides rate of each continent
group_by_continent = data_complete.groupby(data_complete["Continent_Name"])
mean_continent = group_by_continent["suicides_rate_%"].mean()
mean_continent.sort_values(inplace=True, ascending=False)
conti_plot = mean_continent.plot(kind="bar", grid=True, figsize=(8,6),
                title="Suicides Rate by Continent")
plt.xticks(rotation = 0)

# Save the figure
plt.savefig("Images/Location_continent.png")

# plot scatter chart show relation between suicides rate and population density
area_plot = data_complete.plot(kind="scatter", x="Pop. Density per sq. mi.", y="suicides_rate_%", grid=True, figsize=(8,6),
                title="Suicides Rate vs Population Density")
plt.ylabel("Suicides Rate %")
plt.xlim(-0.75, 610)

# Calculate correlation coefficient
cc_popden = data_complete["suicides_rate_%"].corr(data_complete["Pop. Density per sq. mi."])
print(f" The R-value is {cc_popden}")

# Save the figure
plt.savefig("Images/Location_PopulationDensity.png")

# plot scatter chart show relation between suicides rate and coast/area ratio
coast_plot = data_complete.plot(kind="scatter", x="Coastline coast/area ratio", y="suicides_rate_%", grid=True, figsize=(8,6),
                title="Suicides Rate vs Coast/Area Ratio")
plt.xlim(-0.75, 12)
plt.ylabel("Suicides Rate %")
plt.xlabel("Coast/Area Ratio")

# Calculate correlation coefficient
cc_coast_area = data_complete["suicides_rate_%"].corr(data_complete["Coastline coast/area ratio"])
print(f" The R-value is {cc_coast_area}")

```

PYTHON CODES

```
449   # Save the figure
450   plt.savefig("Images/Location_CoastAreaRatio.png")
451
452   # plot scatter chart show relation between suicides rate and Industry Percentage
453   agri_plot = data_complete.plot(kind="scatter", x="Industry", y="suicides_rate_%", grid=True, figsize=(8,6),
454               title="Suicides Rate vs Industry Percentage")
455   plt.ylabel("Suicides Rate %")
456   plt.xlabel("Industry Percentage")
457
458   # Calculate correlation coefficient
459   cc_IndPercentage = data_complete["suicides_rate_%"].corr(data_complete["Industry"])
460   print(f" The R-value is {cc_IndPercentage}")
461
462   # Save the figure
463   plt.savefig("Images/Location_Industry.png")
464
465   # Calculate the average suicides rate for each climate type
466   group_by_climate = data_complete.groupby(data_complete["Climate"])
467   mean_climate = group_by_climate["suicides_rate_%"].mean()
468
469   # plot scatter chart show relation between suicides rate and climate
470   ax = data_complete.plot(kind="scatter", x="Climate", y="suicides_rate_%", grid=True, figsize=(8,6),
471               title="Suicides Rate vs Climate")
472
473   # Add a line show mean value
474   mean_climate.plot(kind = "line",ax = ax,color='red',linestyle = "-.")
475   ax.set_ylim(bottom = 0.0,ymax = 0.033)
476   ax.set_xlim(xmin=0.8, xmax=4.2)
477   plt.ylabel("Suicides Rate %")
478
479   # Calculate correlation coefficient
480   cc_climate = data_complete["suicides_rate_%"].corr(data_complete["Climate"])
481   print(f" The R-value is {cc_climate}")
482
483   # Save the figure
484   plt.savefig("Images/Location_Climate.png")
485
486   # convert data of lat to float
487   data_complete["lat"] = data_complete["lat"].astype(float)
488
489   # plot scatter chart show relation between suicides rate and latitude
490   ax = data_complete.plot(kind="scatter", x="lat", y="suicides_rate_%", grid=True, figsize=(8,6),
491               title="Suicides Rate vs Latitude (Linear Regression)")
492
```

PYTHON CODES

```
# Linear regression
(slope, intercept, r_value, p_value, std_err) = linregress(data_complete['lat'], data_complete['suicides_rate_%'])
fit = slope * data_complete['lat'] + intercept
ax.plot(data_complete['lat'],fit,"b--",color = "coral")
plt.ylabel("Suicides Rate %")
plt.xlabel("Latitude")

# get R-value
print(f" The R-value is {r_value}")

# Save the figure
plt.savefig("Images/Location_lat_linear.png")

# Non-Linear regression
ax = data_complete.plot(kind="scatter", x="lat", y="suicides_rate_%", grid=True, figsize=(8,6),
                        title="Suicides Rate vs Latitude (Non-linear Regression)")

def model(z, a, b):
    return a * np.exp(-b * z)

x_axis = data_complete["lat"]
y_axis = data_complete["suicides_rate_%"]

popt, pcov = curve_fit(model, x_axis, y_axis, p0=(5, 0.1))

# prepare some data for a plot
xx = np.linspace(-40, 70)
yy = model(xx, *popt)

ax.plot(xx,yy, '-',color = "coral")

r_value = model(x_axis, *popt) - y_axis
model_y = model(x_axis, *popt)

plt.ylabel("Suicides Rate %")
plt.xlabel("Latitude")

# Save the figure
plt.savefig("Images/Location_lat_expo.png")

print(popt)
r_value.head()
```

PYTHON CODES

```
536  # exponential model analysis
537  # calculate the mean value for "lat" and "suicides_rate_%"
538  data_analysis = pd.DataFrame(x_axis)
539  data_analysis["suicides_rate_%"]=data_complete["suicides_rate_%"]
540  x_mean = data_analysis["lat"].mean()
541  y_mean = data_analysis["suicides_rate_%"].mean()
542
543  # exponential model analysis
544  # calculate r-square
545  data_analysis["model_y"] = model_y
546  data_analysis["r_value"] = r_value
547
548  data_analysis["(y-y_mean)2"] = (data_analysis["suicides_rate_%"]-y_mean)**2
549  ss_tot = data_analysis["(y-y_mean)2"].sum()
550
551  data_analysis["(y_model-y_mean)2"] = (data_analysis["model_y"]-y_mean)**2
552  ss_reg = data_analysis["(y_model-y_mean)2"].sum()
553
554  data_analysis["(r_value)2"] = (data_analysis["r_value"])**2
555  ss_res = data_analysis["(r_value)2"].sum()
556
557  r_square = 1-(ss_res/ss_tot)
558  print(f" The R-square for this model is {r_square}")
559
560  data_analysis.head()
561
562  # linear model analysis
563  data_analysis_1 = pd.DataFrame(x_axis)
564  data_analysis_1["suicides_rate_%"]=data_complete["suicides_rate_%"]
565
566  data_analysis_1["model_y"] = fit
567  data_analysis_1["r_value"] = data_analysis_1["model_y"] - data_analysis_1["suicides_rate_%"]
568
569  data_analysis_1["(y-y_mean)2"] = (data_analysis_1["suicides_rate_%"]-y_mean)**2
570  ss_tot = data_analysis_1["(y-y_mean)2"].sum()
571
572  data_analysis_1["(y_model-y_mean)2"] = (data_analysis_1["model_y"]-y_mean)**2
573  ss_reg = data_analysis_1["(y_model-y_mean)2"].sum()
574
575  data_analysis_1["(r_value)2"] = (data_analysis_1["r_value"])**2
576  ss_res = data_analysis_1["(r_value)2"].sum()
577
578  r_square = 1-(ss_res/ss_tot)
579  print(f" The R-square for this model is {r_square}")
```

PYTHON CODES

```
# # Economy Analysis (GDP)

# Read source data again for GDP
file = "data/sub_dataset.csv"
datagdp =pd.read_csv(file)
datagdp.head()

# sorting by country, year and sex
datagdp.sort_values(["country","year","sex"],inplace = True)

# # dropping ALL duplicte rows
datagdp.drop_duplicates(subset =['country',' gdp_for_year ($) ','gdp_per_capita ($)'],keep = 'first', inplace = True)

# Sum up suicide_no per year
grouped_per_year = datagdp.groupby('year')

# Sum up GDP per year
gdp = grouped_per_year['gdp_per_capita ($)'].sum()

# x axis is year
yeardata = datagdp['year'].unique()

# Suicices Group by Year  (from earlier analysis)
groupby_yr=data.groupby("year")
year_sum = groupby_yr["suicides_no"].sum()
year_sum.head()

total_rate=year_sum/total_population

# Suicides rate (%)
suicides = total_rate*100

# Plot GDP and Suicide Rate yearly

width = .35 # width of a bar

m1_t = pd.DataFrame({
 'gdp' : gdp,
 'suicides' : suicides})

m1_t = m1_t.reset_index()


fig, ax = plt.subplots()
```

PYTHON CODES

```
fig.suptitle('Yearly GDP and Suicides Trend Comparison')
ax1 = plt.gca()
ax2 = ax1.twinx()  # instantiate a second axes that shares the same x-axis

ax1.bar(x =m1_t['year'], height = m1_t['gdp'], width = width, color ='skyblue')

ax2.plot(m1_t['year'], m1_t['suicides'] ,color ='Coral', lw=2, ls='-')

ax.set(xlabel='Years 1990-2014', ylabel='GDP per Capita ($)')
ax2.set(ylabel='Yearly Suicides Rate (%)')

plt.savefig('Images/Yearly GDP and Suicides Trend Comparison.png',bbox_inches='tight')

# Yearly Change analyis of GDP and Suicides Rate

#Find First discrete difference of  series Suicides and GDP
# Shift 1 period to calculate difference ---> yearly difference (GDP growth/decrease)
s_diff = suicides.diff()
g_diff = gdp.diff()


m2_t = pd.DataFrame({
 'gdp_diff' : g_diff,
 'suicides_diff' : s_diff})

m2_t = m2_t.reset_index()
m2_t.head()

fig, ax1 = plt.subplots()
fig.suptitle('Yearly Change in GDP ($) and Suicides Rate (%) ')

ax = plt.gca()
ax2 = ax1.twinx()  # instantiate a second axes that shares the same x-axis

ax1.bar(x =m2_t['year'], height = m2_t['gdp_diff'], width = width, color ='skyblue')
ax2.plot(m2_t['year'], m2_t['suicides_diff'], color ='Coral', lw=2, ls='-')

ax.set(xlabel='Years 1990-2014', ylabel='Yearly Change in GDP  per Capita ($)')
ax2.set(ylabel='Yearly Change in Sucides Rate (%)')

plt.savefig('Images/Yearly Change in GDP ($) and Suicides Rate (%).png',bbox_inches='tight')
plt.show()

```

PYTHON CODES

```
671   # Yearly Change Rate analyis of GDP and Suicides Rate
672
673   # Ratio calculation
674   ggrowth = g_diff*100/gdp
675   sgrowth = s_diff*100/suicides
676
677
678   m3_t = pd.DataFrame({
679    'gdp_growth' : g_diff*100/gdp,
680    'suicides_growth' : s_diff*100/suicides})
681
682   m3_t = m3_t.reset_index()
683   m3_t.head()
684
685   fig, ax1 = plt.subplots()
686   fig.suptitle('Yearly % Change of GDP and Suicides ')
687
688   ax = plt.gca()
689   ax2 = ax1.twinx()  # instantiate a second axes that shares the same x-axis
690
691   ax1.bar(x =m3_t['year'], height = m3_t['gdp_growth'], width = width, color ='skyblue')
692   ax2.plot(m3_t['year'], m3_t['suicides_growth'], color ='Coral', lw=2, ls='-')
693
694   ax.set(xlabel='Years 1990-2014', ylabel='Yearly % Change in GDP  per Capita (%)')
695   ax2.set(ylabel='Yearly % Change in Sucides (%)')
696
697   plt.savefig('Images/Yearly Change Rate of GDP and Suicides.png',bbox_inches='tight')
698
699   plt.show()
700
701   # Check m3_t table
702   m3_t.head()
703
704   # Finding: first value is invalid since no prior year for first year 1990
705
706   # drop first value which is invalid  (missing value)
707   sgrowth= sgrowth.drop(index=[1990])
708   ggrowth = ggrowth.drop(index=[1990])
709
710   # Find  statistics for suicides
711   v1 = np.var(sgrowth)
712
713   std1 = np.std(sgrowth)
714
```

PYTHON CODES

```
mean1 = np.mean(sgrowth)
print('Statistics for Suicides change rate data:')
print('-----------------------------------------------------')
print(f'Variance: {v1}   Mean: {mean1} and ')
print(f'Standard Deviation: {std1}')

# Find statistics for  GDP
v2 = np.var(ggrowth)

std2 = np.std(ggrowth)

mean2 = np.mean(ggrowth)
print('Statistics for GDP change rate data:')
print('-----------------------------------------------------')
print(f'Variance: {v2}   Mean: {mean2} and')
print(f'Standard Deviation: {std2}')

# Coefficient of Variation = Standard deviation / mean
coef1 = std1/ mean1

coef2  = std2 /mean2
print(f'Coefficient of Suicides change rate: {coef1}   and Coefficient of GDP change rate: {coef2} ')


# ### Coefficient difference is evident and important
# * Opposite sign  (resulting from negative mean of Suicides Rate change rate)
# * Different absolute value : -7.131614 and 1.60759356
#
# ```However when coefficient is a negative value , its representation and comparison power with
# another positive mean population is limited, thus it is elimited from final presentation```

# # Null hypothesis:
# Suicides evolution is not significantly related to GDP evolution.
#
# # Hypothesis:
# Suicides evolution is significantly influenced by GDP evolution.

fig, ax = plt.subplots()

# Scatter Plot of Data
plt.subplot(2,1,1)
plt.scatter(range(len(ggrowth)), ggrowth, label="GDP")
plt.scatter(range(len(sgrowth)), sgrowth, label="Suicides")

```

PYTHON CODES

```
758   # Create a legend
759   plt.legend(title="Change Rate (%)", fancybox=True, loc='center left', bbox_to_anchor=(1, 0.5))
760   plt.title('GDP and Suicides Rate Change Distribution')
761
762   # Histogram Plot of Data
763   plt.subplot(2,1,2)
764   plt.hist(ggrowth, 10, density=True, alpha=0.7, label="GDP")
765   plt.hist(sgrowth, 10, density=True, alpha=0.7, label="Suicides")
766   plt.axvline(ggrowth.mean(), color='k', linestyle='dashed', linewidth=1)
767   plt.axvline(sgrowth.mean(), color='k', linestyle='dashed', linewidth=1)
768
769   # Create a legend
770   plt.legend(title="Change Rate (%)", fancybox=True,loc='center left', bbox_to_anchor=(1, 0.5))
771   #---> define legend box location figure , same for subplot1
772
773   plt.savefig('Images/GDP and Suicides Rate Change Distribution.png',bbox_inches='tight')
```