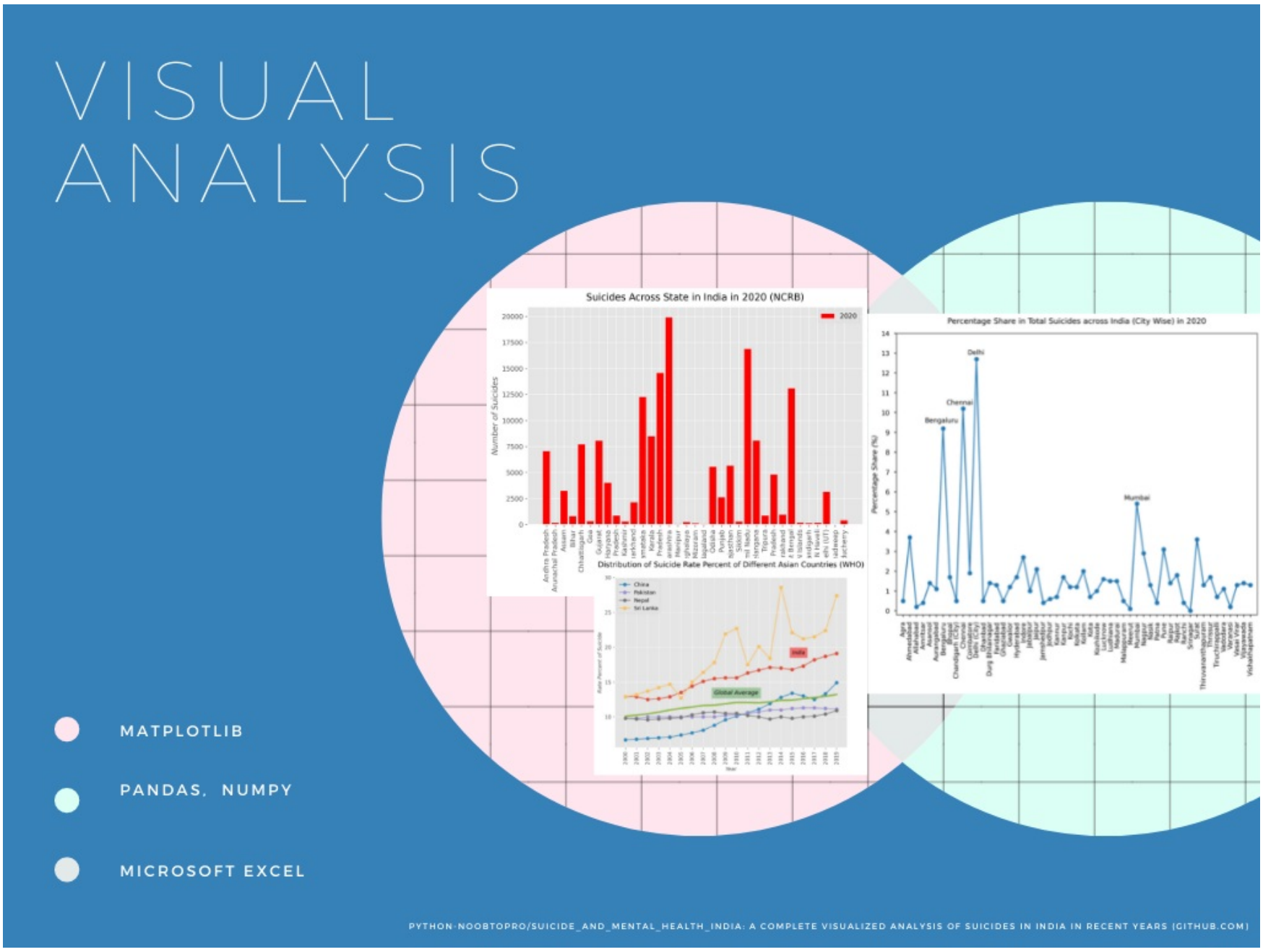
A Visual Analysis of Suicides in pre and post Covid India



India is one of the country with the Highest Suicide rates. The age group of 18 to 30 is majorly affected.

The major cause stands 'Family Problems' as in 2020.

The Covid Outbreak gave a sudden jump of 10% in Suicide Rates across India.

These are some of the many hard facts to digest. India needs to look back at it's roots.

We are the land of Vivekananda. Something needs to be changed, at the earliest.

THE TIMES OF INDIA

NEWS / INDIA NEWS / Suicides Up 10% In Covid-Hit 2020

Suicides up 10% in Covid-hit 2020

Bharti Jain / TNN / Updated: Oct 29, 2021, 02:50 IST

The national suicide prevention strategy in India: context and considerations for urgent action

Lakshmi Vijayakumar, Prabha S Chandra, Munirathinam Suresh Kumar, Soumitra Pathare, Debanjan Banerjee, Tanmoy Goswami, Rakhi Dandona

India has the highest number of suicide deaths in the world, with suicide being the leading cause of death in the 15–39 years age group.[1,2] India's contribution to global suicide deaths has increased from 27·3% in 1990 to 36·5% in 2019 among women and girls, and from 16·7% in 1990 to 20·9% in 2019 among men and boys. On the basis of current trends, India is projected to fall short of the Sustainable Development Goal (SDG) 2030 target of reducing the age-standardised suicide death rate (ASDR; 12·1 per 100 000 population in 2019) by a third.[2] WHO has highlighted suicide as a serious public health concern in India and has called for a comprehensive suicide prevention strategy tailored to India's sociocultural, economic, and health context.[3] At a time when the Government of India is developing a national suicide prevention strategy, the purpose of this Health Policy paper is to identify key challenges,

India Lost More People to Suicide Than to Coronavirus in 2020, Shows NCRB Data

SUICIDE

31 minors died by suicide everyday in India in 2020: Govt

As many as 11,396 children below 18 years died by suicide in 2020 in a significant increase from the previous year.

NEWS / CITY NEWS / AMARAVATI NEWS / Andhra Pradesh: Covid-19 Patient Dies By Suicide At Kuppam Hospital

Andhra Pradesh: Covid-19 patient dies by suicide at Kuppam hospital

Sandeep Raghavan / TNN / Jan 19, 2022, 16:08 IST

IIT-Bombay student jumps to death from hostel

opportunities, and priorities for the national strategy contextualised in the epidemiology, and risk and protective factors, to systematically close the gap towards the SDG target for suicide deaths in India.[2]

building

Narayan Namboodiri / TNN / Updated: Jan 17, 2022, 15:44 IST

All these recent articles and more got me really into using some Data Science skills to get behind the real data and analyze this situation.

In this project I am trying to visualize this grave situation build up in our country.

I will try to analyze the situation before and after Covid Outbreak with as much visualization as I can.

The project starts with Data sources.

My Analysis is majorly based on three data sources:

1. Data by World Health Organization: https://www.who.int/news-room/fact-sheets/detail/suicide

2. Data by National Crime Record Bureau:https://ncrb.gov.in/en/accidental-deaths-suicides-india-adsi

3. Lancet Journals and Research papers:https://www.thelancet.com/journals/lanpsy/article/PIIS2215-0366(21)00152-8/fulltext

The project is intended only for learning Data Visualization and Analysis and no copyright violtaion, whatsoever is intended.

***The Whole Project is done on 'Jupyter Notebook' and is a walkthrough of Data Analysis, Data Cleaning and Data Visualization simultaneously.**

***The Notebook goes in a sequential manner from analyzing the Global Data to the Indian Data and then concluding on some solutions and suggestion to this grave problem.**

The sequence can be understood as follows:

1. WHO Data Analysis and Visualization 2000-2019

1a. Analysis of Suicide rate trend in India

1b. Comparision on India with all other neighbouring countries and Global Avarage

2. NCRB Data Analysis and Visualization 2016-2020

2a. Analysis of Trend in this timeline in India

2b. Zooming in more on Year 2018, 2019, 2020 (State wise)

2c. Specific Analysis of Post Covid India (City wise, Cause wise, Age Group wise)

3. Drawing conclusions and deriving suggestion based on resources and research papers from various active groups dealing with the situation like International Journal of Mental Health Systems

1. WHO Data Analysis and Visualization 2000-2019

Getting WHO Data on Suicide (2000-2019) (All Countries)

https://www.who.int/data/gho/data/themes/mental-health

Data Cleaning

Since the data is not Visualization friendly i.e., not in a format to give x,y directly to plot we need to refine/clean and consolidate the data as per our requirement

This is the main challenge in Data Visualization and it tests patience

In [10]:

```
# Importing Libraries
import pandas as pd
```

1a. WHO India Suicide rate 2000-2019

Starting with oveall suicide rate (per 100,000 population) in India 2000-2019 WHO Data

In [11]:

```
# Importing Data
df = pd.read_csv('./WHO data/who_suiciderate_india_2000-19_overall.csv')
```

In [10]:

```
df.head(3)
```

Out[10]:

	Suicide per 100000 population	Suicide per 100000 population.1	Suicide per 100000 population.2	Suicide per 100000 population.3	Suicide per 100000 population.4	Suicide per 100000 population.5	Suicide per 100000 population.6	Suicide per 100000 population.7	Suicide per 100000 population.8	Suicide per 100000 population.9	Suicide per 100000 population.10	Suicide per 100000 population.11	Suicide per 100000 population.12	Suicide per 100000 population.13	Suicide pe 10000 population.1
0	2019.0	2018.0	2017.0	2016.0	2015.0	2014.0	2013.0	2012.0	2011.0	2010.0	2009.0	2008.0	2007.0	2006.0	200
1	12.9	12.9	12.5	12.6	12.9	13.5	14.4	15.1	15.5	15.6	15.6	16.3	16.7	17.1	1

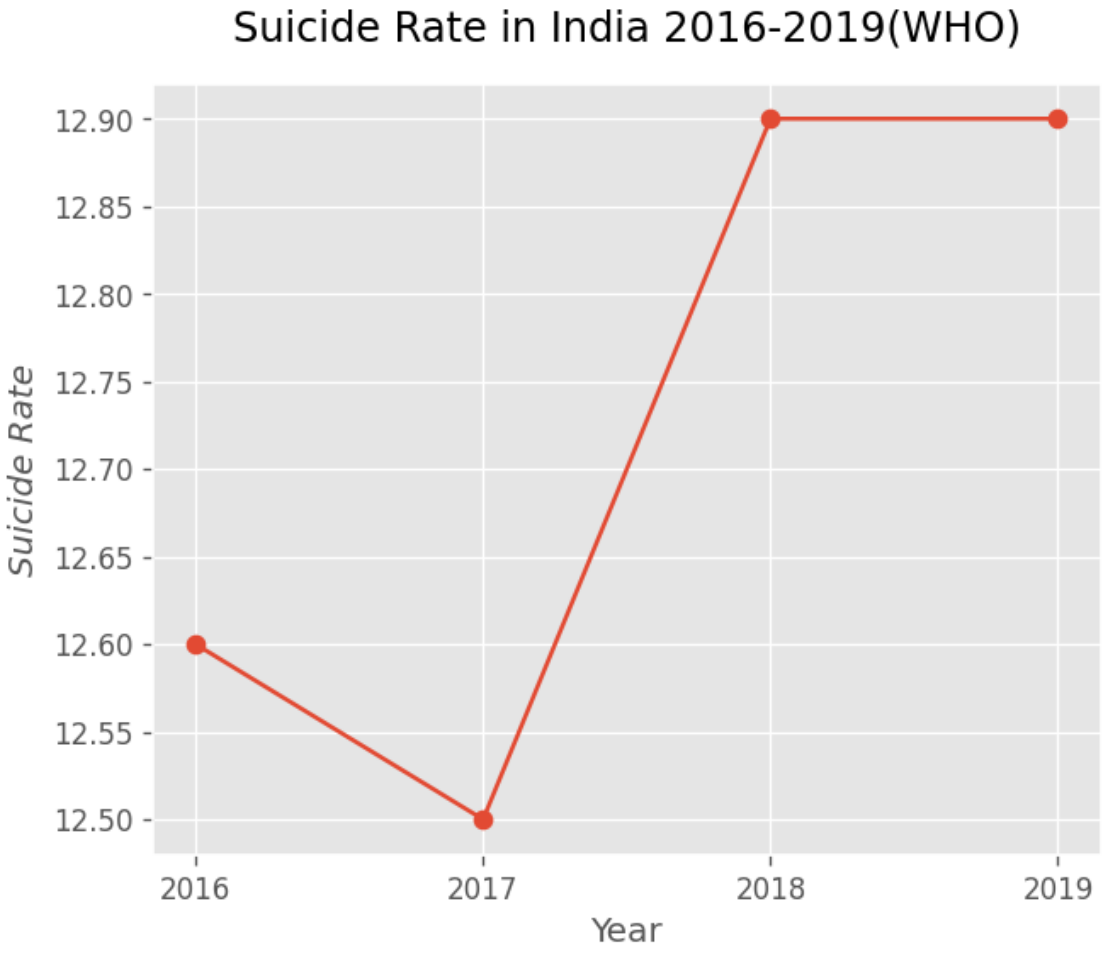
In [11]:

```
# Importing libraries

import matplotlib.pyplot as plt
import numpy as np
```

Plotting a simple data

In [15]:

```
plt.style.use('ggplot')
plt.figure(facecolor='w',figsize=(6,5), dpi=120)
plt.title('Suicide Rate in India 2016-2019(WHO)', pad=15)
plt.plot(df.iloc[0,:4], df.iloc[1,:4], marker='o')
plt.xlabel('Year')
plt.ylabel('Suicide Rate', style='italic')
plt.xticks(np.arange(2016,2020,1))
# plt.savefig('India_WHO.png', bbox_inches = 'tight', facecolor='white', transparent = False, dpi=300)
plt.show()
```



Conclusion:

The Suicide rate in India saw a huge rise after 2017

1b. Comparing the Indian Data with the World Data

In [12]:

```
# importing data
df2 = pd.read_csv('./WHO data/who_suicidedata_world_2000-19.csv')
```

In [17]:

```
cols = df2.columns
```

In [18]:

```
df2.head(5)
```

Out[18]:

	Unnamed: 0	Age-standardized suicide rates (per 100 000 population)	Age-standardized suicide rates (per 100 000 population).1	Age-standardized suicide rates (per 100 000 population).2	Age-standardized suicide rates (per 100 000 population).3	Age-standardized suicide rates (per 100 000 population).4	Age-standardized suicide rates (per 100 000 population).5	Age-standardized suicide rates (per 100 000 population).6	Age-standardized suicide rates (per 100 000 population).7	Age-standardized suicide rates (per 100 000 population).8	...	Age-standardized suicide rates (per 100 000 population).50	Age-standardized suicide rates (per 100 000 population).51	Age-standardized suicide rates (per 100 000 population).52	Age-standardized suicide rates (per 100 000 population).53
0	NaN	2019	2019	2019	2018	2018	2018	2017	2017	2017	...	2003	2002	2002	2002
1	Country	Both sexes	Male	Female	Both sexes	Male	Female	Both sexes	Male	Female	...	Female	Both sexes	Male	Female
2	Afghanistan	6.0 [3.4-9.9]	6.2 [3.5-10.5]	5.7 [3.2-9.2]	5.9 [3.4-9.8]	6.2 [3.5-10.4]	5.6 [3.2-9.1]	6.0 [3.4-9.9]	6.3 [3.7-10.7]	5.5 [3.1-9.0]	...	7.6 [4.5-12.2]	7.9 [4.5-12.8]	8.0 [4.6-13.1]	7.7 [4.5-12.4]
3	Albania	3.7 [2.1-5.7]	5.3 [2.7-8.3]	2.2 [1.4-3.3]	3.9 [2.2-6.0]	5.6 [2.9-8.7]	2.4 [1.5-3.5]	4.1 [2.3-6.3]	5.7 [2.9-9.0]	2.5 [1.6-3.7]	...	3.2 [2.5-4.1]	4.8 [3.2-6.3]	6.6 [4.1-8.7]	3.0 [2.3-3.9]
4	Algeria	2.6 [1.4-4.4]	3.3 [1.8-5.7]	1.9 [1.0-3.1]	2.6 [1.4-4.4]	3.2 [1.8-5.7]	1.9 [1.0-3.2]	2.5 [1.4-4.4]	3.2 [1.7-5.6]	1.9 [1.0-3.1]	...	3.1 [1.6-5.1]	4.4 [2.4-7.6]	5.6 [2.9-9.8]	3.3 [1.8-5.4]

5 rows × 61 columns

Since, the above data contains lot of noise like [ ], str values etc. we need to make a well aligned integer data to get it easily on matplotlib. In the coming section we will be dealing with that.

Cleaning Data

We need to replace all values containing [ ] with only the integer values before the brackets

Making a column list of all columns except column with country names (only string column)

In [19]:

```
colslist = df2.columns.tolist()

# Removing Country Name column

colslist = colslist[1:]
```

In [1]:

```
# Applying split on values of all columns

# for cols in colslist:
#     print(df2[cols].str.split(' '))
```

Now replace the data with splitted data

In [21]:

```
for cols in colslist:
    df2[cols] = [i[0] for i in df2[cols].str.split(' ')]

# Replaced data
```

In [22]:

```
df2.head()
```

Out[22]:

	Unnamed: 0	Age-standardized suicide rates (per 100 000 population)	Age-standardized suicide rates (per 100 000 population).1	Age-standardized suicide rates (per 100 000 population).2	Age-standardized suicide rates (per 100 000 population).3	Age-standardized suicide rates (per 100 000 population).4	Age-standardized suicide rates (per 100 000 population).5	Age-standardized suicide rates (per 100 000 population).6	Age-standardized suicide rates (per 100 000 population).7	Age-standardized suicide rates (per 100 000 population).8	...	Age-standardized suicide rates (per 100 000 population).50	Age-standardized suicide rates (per 100 000 population).51	Age-standardized suicide rates (per 100 000 population).52	Age-standardized suicide rates (per 100 000 population).53
0	NaN	2019	2019	2019	2018	2018	2018	2017	2017	2017	...	2003	2002	2002	2002
1	Country	Both	Male	Female	Both	Male	Female	Both	Male	Female	...	Female	Both	Male	Female
2	Afghanistan	6.0	6.2	5.7	5.9	6.2	5.6	6.0	6.3	5.5	...	7.6	7.9	8.0	7.7
3	Albania	3.7	5.3	2.2	3.9	5.6	2.4	4.1	5.7	2.5	...	3.2	4.8	6.6	3.0
4	Algeria	2.6	3.3	1.9	2.6	3.2	1.9	2.5	3.2	1.9	...	3.1	4.4	5.6	3.3

5 rows × 61 columns

The data is now much clear to begin analyzation

Saving new csv as a copy csv

In [17]:

```
# df2.to_csv('./WHO Data/who_world_suicide_data_cleaned.csv')
```

After saving this data we cleaned it further with 'Google Sheets' and added a "Global Average" row as well

Loading Data from Google sheets

In [23]:

```
df3 = pd.read_csv('./WHO data/who_world_suicide_data_googlesheets.csv')
```

In [24]:

```
df3.head(3)
```

Out[24]:

	Unnamed: 0	Unnamed: 0.1	Age-standardized suicide rates (per 100 000 population)	Age-standardized suicide rates (per 100 000 population).1	Age-standardized suicide rates (per 100 000 population).2	Age-standardized suicide rates (per 100 000 population).3	Age-standardized suicide rates (per 100 000 population).4	Age-standardized suicide rates (per 100 000 population).5	Age-standardized suicide rates (per 100 000 population).6	Age-standardized suicide rates (per 100 000 population).7	...	Age-standardized suicide rates (per 100 000 population).50	Age-standardized suicide rates (per 100 000 population).51	Age-standardized suicide rates (per 100 000 population).52	Age-standardized suicide rates (per 100 000 population).53	s s po
0	0	NaN	2019	2019	2019	2018	2018	2018	2017	2017	...	2003	2002	2002	2002	
1	1	Country	Both	Male	Female	Both	Male	Female	Both	Male	...	Female	Both	Male	Female	
2	2	Afghanistan	6.00	6.20	5.70	5.90	6.20	5.60	6.00	6.30	...	7.6	7.9	8	7.7	

3 rows × 62 columns

In [25]:

```
df3.drop(columns=['Unnamed: 0'], inplace=True)
```

In [26]:

```
# Avoiding any NaN value
df3 = df3.fillna(' ')
```

In [27]:

```
## Getting 'Global Average' row

globaldf = df3[df3['Unnamed: 0.1'].str.contains('Global')]
globaldf
```

Out[27]:

	Unnamed: 0.1	Age-standardized suicide rates (per 100 000 population)	Age-standardized suicide rates (per 100 000 population).1	Age-standardized suicide rates (per 100 000 population).2	Age-standardized suicide rates (per 100 000 population).3	Age-standardized suicide rates (per 100 000 population).4	Age-standardized suicide rates (per 100 000 population).5	Age-standardized suicide rates (per 100 000 population).6	Age-standardized suicide rates (per 100 000 population).7	Age-standardized suicide rates (per 100 000 population).8	...	Age-standardized suicide rates (per 100 000 population).50	Age-standardized suicide rates (per 100 000 population).51	Age-standardized suicide rates (per 100 000 population).52	Age-standardized suicide rates (per 100 000 population).53
185	Global Average	10.09	16.14	4.54	10.25	16.43	4.58	10.43	16.74	4.66	...	5.86	12.79	20.38	5.94

1 rows × 61 columns

We can keep Global Average seperately if we want

Getting Year row and appending it as a row to global average

In [28]:

```
# Year row
a = df3.iloc[0:1,:]
a
```

Out[28]:

	Unnamed: 0.1	Age-standardized suicide rates (per 100 000 population)	Age-standardized suicide rates (per 100 000 population).1	Age-standardized suicide rates (per 100 000 population).2	Age-standardized suicide rates (per 100 000 population).3	Age-standardized suicide rates (per 100 000 population).4	Age-standardized suicide rates (per 100 000 population).5	Age-standardized suicide rates (per 100 000 population).6	Age-standardized suicide rates (per 100 000 population).7	Age-standardized suicide rates (per 100 000 population).8	...	Age-standardized suicide rates (per 100 000 population).50	Age-standardized suicide rates (per 100 000 population).51	Age-standardized suicide rates (per 100 000 population).52	Age-standardized suicide rates (per 100 000 population).53
0		2019	2019	2019	2018	2018	2018	2017	2017	2017	...	2003	2002	2002	2002

1 rows × 61 columns

In [29]:

```
# Appending Year row on Global Average row and making a new dataframe
a.append(globaldf, sort=False).reset_index(drop = True)
```

Out[29]:

	Unnamed: 0.1	Age-standardized suicide rates (per 100 000 population)	Age-standardized suicide rates (per 100 000 population).1	Age-standardized suicide rates (per 100 000 population).2	Age-standardized suicide rates (per 100 000 population).3	Age-standardized suicide rates (per 100 000 population).4	Age-standardized suicide rates (per 100 000 population).5	Age-standardized suicide rates (per 100 000 population).6	Age-standardized suicide rates (per 100 000 population).7	Age-standardized suicide rates (per 100 000 population).8	...	Age-standardized suicide rates (per 100 000 population).50	Age-standardized suicide rates (per 100 000 population).51	Age-standardized suicide rates (per 100 000 population).52	Age-standardized suicide rates (per 100 000 population).53	p
0		2019	2019	2019	2018	2018	2018	2017	2017	2017	...	2003	2002	2002	2002	
1	Global Average	10.09	16.14	4.54	10.25	16.43	4.58	10.43	16.74	4.66	...	5.86	12.79	20.38	5.94	

2 rows × 61 columns

In [30]:

```
globalavg = a.append(globaldf, sort=False).reset_index(drop = True)
```

We can save this data as a different csv as well

In [29]:

```
# globalavg.to_csv('./WHO data/who_globalavg.csv', index=False)
```

Coming back to original world data set

In [31]:

```
df3.tail(4)
```

Out[31]:

	Unnamed: 0.1	Age-standardized suicide rates (per 100 000 population)	Age-standardized suicide rates (per 100 000 population).1	Age-standardized suicide rates (per 100 000 population).2	Age-standardized suicide rates (per 100 000 population).3	Age-standardized suicide rates (per 100 000 population).4	Age-standardized suicide rates (per 100 000 population).5	Age-standardized suicide rates (per 100 000 population).6	Age-standardized suicide rates (per 100 000 population).7	Age-standardized suicide rates (per 100 000 population).8	...	Age-standardized suicide rates (per 100 000 population).50	Age-standardized suicide rates (per 100 000 population).51	Age-standardized suicide rates (per 100 000 population).52	Age-standardized suicide rates (per 100 000 population).53
182	Yemen	7.10	9.00	5.30	7.10	9.00	5.30	6.90	8.70	5.10	...	6.1	8.2	10.2	6.2
183	Zambia	14.40	25.70	5.30	15.90	28.10	6.00	16.70	29.10	6.60	...	11.4	22.3	34.5	12.6
184	Zimbabwe	23.60	37.80	13.50	23.90	38.60	13.50	25.90	41.30	15.00	...	13.7	20.3	29.8	13.5
185	Global Average	10.09	16.14	4.54	10.25	16.43	4.58	10.43	16.74	4.66	...	5.86	12.79	20.38	5.94

4 rows × 61 columns

In [25]:

```
df3
```

Out[25]:

	Unnamed: 0.1	Age-standardized suicide rates (per 100 000 population)	Age-standardized suicide rates (per 100 000 population).1	Age-standardized suicide rates (per 100 000 population).2	Age-standardized suicide rates (per 100 000 population).3	Age-standardized suicide rates (per 100 000 population).4	Age-standardized suicide rates (per 100 000 population).5	Age-standardized suicide rates (per 100 000 population).6	Age-standardized suicide rates (per 100 000 population).7	Age-standardized suicide rates (per 100 000 population).8	...	Age-standardized suicide rates (per 100 000 population).50	Age-standardized suicide rates (per 100 000 population).51	Age-standardized suicide rates (per 100 000 population).52	Age-standardized suicide rates (per 100 000 population).53
0		2019	2019	2019	2018	2018	2018	2017	2017	2017	...	2003	2002	2002	200
1	Country	Both	Male	Female	Both	Male	Female	Both	Male	Female	...	Female	Both	Male	Femal
2	Afghanistan	6.00	6.20	5.70	5.90	6.20	5.60	6.00	6.30	5.50	...	7.6	7.9	8	7.
3	Albania	3.70	5.30	2.20	3.90	5.60	2.40	4.10	5.70	2.50	...	3.2	4.8	6.6	
4	Algeria	2.60	3.30	1.90	2.60	3.20	1.90	2.50	3.20	1.90	...	3.1	4.4	5.6	3.
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	.
181	Viet Nam	7.20	10.60	4.20	7.40	10.80	4.20	7.50	11.10	4.20	...	4.8	7	9.3	4.
182	Yemen	7.10	9.00	5.30	7.10	9.00	5.30	6.90	8.70	5.10	...	6.1	8.2	10.2	6.
183	Zambia	14.40	25.70	5.30	15.90	28.10	6.00	16.70	29.10	6.60	...	11.4	22.3	34.5	12.
184	Zimbabwe	23.60	37.80	13.50	23.90	38.60	13.50	25.90	41.30	15.00	...	13.7	20.3	29.8	13.
185	Global Average	10.09	16.14	4.54	10.25	16.43	4.58	10.43	16.74	4.66	...	5.86	12.79	20.38	5.9

186 rows × 61 columns

We need to get only overall (Both) values, the data also contains gender distribution (Male/Female), we wwnt to filter that data from the dataframe

From the original data csv we know that the overall rate (male+female) is 1st and every third column thereafter

In [33]:

```
### Selecting all (both) overall columns
df3.iloc[:, 1::3]
```

Out[33]:

	Age-standardized suicide rates (per 100 000 population)	Age-standardized suicide rates (per 100 000 population).3	Age-standardized suicide rates (per 100 000 population).6	Age-standardized suicide rates (per 100 000 population).9	Age-standardized suicide rates (per 100 000 population).12	Age-standardized suicide rates (per 100 000 population).15	Age-standardized suicide rates (per 100 000 population).18	Age-standardized suicide rates (per 100 000 population).21	Age-standardized suicide rates (per 100 000 population).24	Age-standardized suicide rates (per 100 000 population).27	Age-standardized suicide rates (per 100 000 population).30	Age-standardized suicide rates (per 100 000 population).33	Age-standardized suicide rates (per 100 000 population).36	standa suicid (per 1 populati
0	2019	2018	2017	2016	2015	2014	2013	2012	2011	2010	2009	2008	2007	
1	Both	Both	Both	Both	Both	Both	Both	Both	Both	Both	Both	Both	Both	
2	6.00	5.90	6.00	6.00	6.00	6	6.2	6.2	6.4	6.7	6.8	7.2	7.4	
3	3.70	3.90	4.10	4.20	4.20	4.5	4.8	4.8	7.6	7.6	8	8.1	8.1	
4	2.60	2.60	2.50	2.60	2.70	2.8	2.9	2.9	2.9	3	3.2	3.3	3.5	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	
181	7.20	7.40	7.50	7.60	7.60	7.6	7.5	7.5	7.4	7.3	7.3	7.4	7.2	
182	7.10	7.10	6.90	7.00	6.80	7	7.1	7.2	7.2	7.4	7.6	7.8	7.7	
183	14.40	15.90	16.70	16.70	17.30	17.7	17.9	18.5	19.8	19.7	19	19	19.8	
184	23.60	23.90	25.90	28.70	30.70	30.8	31.4	33.1	34.3	35	35.2	31.7	27.2	
185	10.09	10.25	10.43	10.68	10.99	11.23	11.42	11.63	11.68	11.88	12.08	12.07	12.02	

186 rows × 20 columns

In [34]:

```
# Selecting Country name column
df3.iloc[:,0:1]
```

Out[34]:

	Unnamed: 0.1
0	
1	Country

	Unnamed: 0.1
1	Country
2	Afghanistan
3	Albania
4	Algeria
...	...
181	Viet Nam
182	Yemen
183	Zambia
184	Zimbabwe
185	Global Average

186 rows × 1 columns

We need to take every 3rd column starting from index 1 and join it with country name column

In [35]:

```
cols = list(df3.columns.values)
```

In [2]:

```
# cols
```

In [37]:

```
## These are the columns
cols[1::3]
```

Out[37]:

```
['Age-standardized suicide rates (per 100 000 population)',
 'Age-standardized suicide rates (per 100 000 population).3',
 'Age-standardized suicide rates (per 100 000 population).6',
 'Age-standardized suicide rates (per 100 000 population).9',
 'Age-standardized suicide rates (per 100 000 population).12',
 'Age-standardized suicide rates (per 100 000 population).15',
 'Age-standardized suicide rates (per 100 000 population).18',
 'Age-standardized suicide rates (per 100 000 population).21',
 'Age-standardized suicide rates (per 100 000 population).24',
 'Age-standardized suicide rates (per 100 000 population).27',
 'Age-standardized suicide rates (per 100 000 population).30',
 'Age-standardized suicide rates (per 100 000 population).33',
 'Age-standardized suicide rates (per 100 000 population).36',
 'Age-standardized suicide rates (per 100 000 population).39',
 'Age-standardized suicide rates (per 100 000 population).42',
 'Age-standardized suicide rates (per 100 000 population).45',
 'Age-standardized suicide rates (per 100 000 population).48',
 'Age-standardized suicide rates (per 100 000 population).51',
 'Age-standardized suicide rates (per 100 000 population).54',
 'Age-standardized suicide rates (per 100 000 population).57']
```

Concating the both dataframes

In [38]:

```
gencombined = df3[cols[0:1] + cols[1::3]]
```

This is the required dataframe

In [39]:

```
gencombined.head(4)
```

Out[39]:

	Unnamed: 0.1	Age-standardized suicide rates (per 100 000 population)	Age-standardized suicide rates (per 100 000 population).3	Age-standardized suicide rates (per 100 000 population).6	Age-standardized suicide rates (per 100 000 population).9	Age-standardized suicide rates (per 100 000 population).12	Age-standardized suicide rates (per 100 000 population).15	Age-standardized suicide rates (per 100 000 population).18	Age-standardized suicide rates (per 100 000 population).21	Age-standardized suicide rates (per 100 000 population).24	...	Age-standardized suicide rates (per 100 000 population).30	Age-standardized suicide rates (per 100 000 population).33	Age-standardized suicide rates (per 100 000 population).36	Age-standardized suicide (per 10 populatio
0		2019	2018	2017	2016	2015	2014	2013	2012	2011	...	2009	2008	2007	
1	Country	Both	Both	Both	Both	Both	Both	Both	Both	Both	...	Both	Both	Both	
2	Afghanistan	6.00	5.90	6.00	6.00	6.00	6	6.2	6.2	6.4	...	6.8	7.2	7.4	
3	Albania	3.70	3.90	4.10	4.20	4.20	4.5	4.8	4.8	7.6	...	8	8.1	8.1	

4 rows × 21 columns

In [40]:

```
df3.iloc[2,1]
```

Out[40]:

```
'6.00'
```

In [41]:

```
type(df3.iloc[2,1])
```

Out[41]:

```
str
```

There is still a big challenge, i.e, the data is read as str datatype in pandas DataFrame. Why?

This is because if any column value contains any str character the whole column is read as str datatype. This will not allow us to plot the data on Matplotlib. There are many ways to get rid of this problem. Here we are going to take a bit lengthy but most trustworthy way and that is to make a good header and index that defines the data and remove any unwanted rows that contains str. It is shown in the steps below.

In [ ]:

```
## Shortcut to drop headers
## This will be required quite often in the process

# df_dict = dict.fromkeys(df.columns, '')
# df.rename(columns = df_dict)
```

In [42]:

```
# Make a copy of data
data = df3.copy()
```

In [43]:

```
# Dropping Headers

df_dict = dict.fromkeys(data.columns, '')
data.rename(columns = df_dict, inplace=True)
```

In [44]:

```
data.head()
```

Out[44]:

											...										
0		2019	2019	2019	2018	2018	2018	2017	2017	2017	...	2003	2002	2002	2002	2001	2001	2001	2000	2000	2000
1	Country	Both	Male	Female	Both	Male	Female	Both	Male	Female	...	Female	Both	Male	Female	Both	Male	Female	Both	Male	Female
2	Afghanistan	6.00	6.20	5.70	5.90	6.20	5.60	6.00	6.30	5.50	...	7.6	7.9	8	7.7	7.9	7.9	7.9	7.7	7.6	7.8
3	Albania	3.70	5.30	2.20	3.90	5.60	2.40	4.10	5.70	2.50	...	3.2	4.8	6.6	3	4.7	6.5	3	5.2	7.6	2.9
4	Algeria	2.60	3.30	1.90	2.60	3.20	1.90	2.50	3.20	1.90	...	3.1	4.4	5.6	3.3	4.6	5.7	3.4	4.7	5.9	3.5

5 rows × 61 columns

In [45]:

```
# Storing first row (0 index) values as values for new column headers

newheadlist = data.iloc[0:1, :].values
```

In [ ]:

```
# It returns a 2D array, convert it to a list
```

In [4]:

```
# newheadlist.tolist()[0][1:]
```

In [47]:

```
newheadlist = newheadlist.tolist()[0][1:]
```

In [48]:

```
# Adding B for Both, M for Male and F for Female with the Year to make the data more obvious to predict
# Alos, it will remove unwanted rows/columns to make data more consolidated

newheadlist[::3] = [i +'(B)' for i in newheadlist[::3]]
```

In [49]:

```
newheadlist[1::3] = [i +'(M)' for i in newheadlist[1::3]]
newheadlist[2::3] = [i +'(F)' for i in newheadlist[2::3]]
```

In [5]:

```
# newheadlist
```

In [51]:

```
newheadlist.insert(0,'Country')

# Adding one more entry to match the length of dataframe
```

In [3]:

```
# newheadlist
```

Now make this newheadlist as new Header

In [53]:

```
data.columns = newheadlist
```

In [54]:

```
# see the new header (short and obvious)
data.head()
```

Out[54]:

	Country	2019(B)	2019(M)	2019(F)	2018(B)	2018(M)	2018(F)	2017(B)	2017(M)	2017(F)	...	2003(F)	2002(B)	2002(M)	2002(F)	2001(B)	2001(M)	2001(F)	2000(B)	2000(M)	2000(F)
0		2019	2019	2019	2018	2018	2018	2017	2017	2017	...	2003	2002	2002	2002	2001	2001	2001	2000	2000	2000
1	Country	Both	Male	Female	Both	Male	Female	Both	Male	Female	...	Female	Both	Male	Female	Both	Male	Female	Both	Male	Female
2	Afghanistan	6.00	6.20	5.70	5.90	6.20	5.60	6.00	6.30	5.50	...	7.6	7.9	8	7.7	7.9	7.9	7.9	7.7	7.6	7.8
3	Albania	3.70	5.30	2.20	3.90	5.60	2.40	4.10	5.70	2.50	...	3.2	4.8	6.6	3	4.7	6.5	3	5.2	7.6	2.9
4	Algeria	2.60	3.30	1.90	2.60	3.20	1.90	2.50	3.20	1.90	...	3.1	4.4	5.6	3.3	4.6	5.7	3.4	4.7	5.9	3.5

5 rows × 61 columns

In [55]:

```
## Dropping row 0 and 1 (now these rows are not required)
```

```
data.drop(data.index[[0,1]]).reset_index(drop=True)
```

Out[55]:

	Country	2019(B)	2019(M)	2019(F)	2018(B)	2018(M)	2018(F)	2017(B)	2017(M)	2017(F)	...	2003(F)	2002(B)	2002(M)	2002(F)	2001(B)	2001(M)	2001(F)	2000(B)	2000(M)	2000(F)
0	Afghanistan	6.00	6.20	5.70	5.90	6.20	5.60	6.00	6.30	5.50	...	7.6	7.9	8	7.7	7.9	7.9	7.9	7.7	7.6	7.8
1	Albania	3.70	5.30	2.20	3.90	5.60	2.40	4.10	5.70	2.50	...	3.2	4.8	6.6	3	4.7	6.5	3	5.2	7.6	2.9
2	Algeria	2.60	3.30	1.90	2.60	3.20	1.90	2.50	3.20	1.90	...	3.1	4.4	5.6	3.3	4.6	5.7	3.4	4.7	5.9	3.5
3	Angola	12.60	21.70	4.70	12.40	21.30	4.60	12.40	21.00	4.90	...	6.9	17.2	29	6.5	17.5	29.8	6.1	17.6	30	6.2
4	Antigua and Barbuda	0.30	0.00	0.60	0.30	0.00	0.60	0.00	0.00	0.00	...	0	1.3	2.8	0	1.9	4.2	0	2	4.5	0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
179	Viet Nam	7.20	10.60	4.20	7.40	10.80	4.20	7.50	11.10	4.20	...	4.8	7	9.3	4.9	7.1	9.5	5	7.2	9.4	5.2
180	Yemen	7.10	9.00	5.30	7.10	9.00	5.30	6.90	8.70	5.10	...	6.1	8.2	10.2	6.2	8.4	10.4	6.5	8.5	10.5	6.5
181	Zambia	14.40	25.70	5.30	15.90	28.10	6.00	16.70	29.10	6.60	...	11.4	22.3	34.5	12.6	22.7	34.3	13.4	24	35.9	14.5
182	Zimbabwe	23.60	37.80	13.50	23.90	38.60	13.50	25.90	41.30	15.00	...	13.7	20.3	29.8	13.5	19.5	28.6	13	20	28.2	14.2
183	Global Average	10.09	16.14	4.54	10.25	16.43	4.58	10.43	16.74	4.66	...	5.86	12.79	20.38	5.94	12.94	20.62	5.99	13.23	21.04	6.18

184 rows × 61 columns

In [56]:

```
data = data.drop(data.index[[0,1]]).reset_index(drop=True)
```

In [57]:

```
data.head()
```

Out[57]:

	Country	2019(B)	2019(M)	2019(F)	2018(B)	2018(M)	2018(F)	2017(B)	2017(M)	2017(F)	...	2003(F)	2002(B)	2002(M)	2002(F)	2001(B)	2001(M)	2001(F)	2000(B)	2000(M)	2000(F)
0	Afghanistan	6.00	6.20	5.70	5.90	6.20	5.60	6.00	6.30	5.50	...	7.6	7.9	8	7.7	7.9	7.9	7.9	7.7	7.6	7.8
1	Albania	3.70	5.30	2.20	3.90	5.60	2.40	4.10	5.70	2.50	...	3.2	4.8	6.6	3	4.7	6.5	3	5.2	7.6	2.9
2	Algeria	2.60	3.30	1.90	2.60	3.20	1.90	2.50	3.20	1.90	...	3.1	4.4	5.6	3.3	4.6	5.7	3.4	4.7	5.9	3.5
3	Angola	12.60	21.70	4.70	12.40	21.30	4.60	12.40	21.00	4.90	...	6.9	17.2	29	6.5	17.5	29.8	6.1	17.6	30	6.2
4	Antigua and Barbuda	0.30	0.00	0.60	0.30	0.00	0.60	0.00	0.00	0.00	...	0	1.3	2.8	0	1.9	4.2	0	2	4.5	0

5 rows × 61 columns

It gets easy to visualize the data columnwise in matplotlib rather than row-wise. Therefore we need to Transpose the dataframe

In [58]:

```
# Changing index to head
head = data.Country.values.tolist()
```

In [59]:

```
head[3:7]
```

Out[59]:

```
['Angola', 'Antigua and Barbuda', 'Argentina', 'Armenia']
```

In [60]:

```
# Transposing the dataframe

datat = data.T
```

In [61]:

```
datat
```

Out[61]:

	0	1	2	3	4	5	6	7	8	9	...	174	175	176	177	178	179	180	181	182	18
Country	Afghanistan	Albania	Algeria	Angola	Antigua and Barbuda	Argentina	Armenia	Australia	Austria	Azerbaijan	...	United States of America	Uruguay	Uzbekistan	Vanuatu	Venezuela (Bolivarian Republic of)	Viet Nam	Yemen	Zambia	Zimbabwe	Globa Averag
2019(B)	6.00	3.70	2.60	12.60	0.30	8.10	2.70	11.30	10.40	4.00	...	14.50	18.80	8.30	21.00	2.10	7.20	7.10	14.40	23.60	10.0
2019(M)	6.20	5.30	3.30	21.70	0.00	13.50	4.90	17.00	16.60	6.60	...	22.40	31.10	11.80	33.10	3.70	10.60	9.00	25.70	37.80	16.1
2019(F)	5.70	2.20	1.90	4.70	0.60	3.30	1.00	5.60	4.60	1.50	...	6.80	7.70	4.90	9.00	0.70	4.20	5.30	5.30	13.50	4.5
2018(B)	5.90	3.90	2.60	12.40	0.30	9.00	1.70	11.30	11.20	3.90	...	14.10	18.80	8.70	21.10	2.20	7.40	7.10	15.90	23.90	10.2
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	.
2001(M)	7.9	6.5	5.7	29.8	4.2	17.6	5.3	17.6	23	6.2	...	16.9	21.8	18.9	35.4	11.5	9.5	10.4	34.3	28.6	20.6
2001(F)	7.9	3	3.4	6.1	0	4	1.3	4.9	7.2	1.5	...	4.1	5.9	4.9	9.9	1.8	5	6.5	13.4	13	5.9
2000(B)	7.7	5.2	4.7	17.6	2	9.2	3.3	11.8	15.8	3.4	...	10	14.5	12	23.2	6.4	7.2	8.5	24	20	13.2
2000(M)	7.6	7.6	5.9	30	4.5	16	5.5	18.8	24.9	5.8	...	16.4	25.7	19.6	36	11.3	9.4	10.5	35.9	28.2	21.0
2000(F)	7.8	2.9	3.5	6.2	0	3.4	1.7	5	7.9	1.3	...	4	5.1	4.8	10.1	1.7	5.2	6.5	14.5	14.2	6.1

61 rows × 184 columns

In [62]:

```
# Old index changed to new header
datat.columns = head
```

In [63]:

```
datat
```

Out[63]:

	Afghanistan	Albania	Algeria	Angola	Antigua and Barbuda	Argentina	Armenia	Australia	Austria	Azerbaijan	...	United States of America	Uruguay	Uzbekistan	Vanuatu	Venezuela (Bolivarian Republic of)	Viet Nam	Yemen	Zambia	Zimbabwe	Globa Averag
Country	Afghanistan	Albania	Algeria	Angola	Antigua and Barbuda	Argentina	Armenia	Australia	Austria	Azerbaijan	...	United States of America	Uruguay	Uzbekistan	Vanuatu	Venezuela (Bolivarian Republic of)	Viet Nam	Yemen	Zambia	Zimbabwe	Globa Averag
2019(B)	6.00	3.70	2.60	12.60	0.30	8.10	2.70	11.30	10.40	4.00	...	14.50	18.80	8.30	21.00	2.10	7.20	7.10	14.40	23.60	10.0
2019(M)	6.20	5.30	3.30	21.70	0.00	13.50	4.90	17.00	16.60	6.60	...	22.40	31.10	11.80	33.10	3.70	10.60	9.00	25.70	37.80	16.1
2019(F)	5.70	2.20	1.90	4.70	0.60	3.30	1.00	5.60	4.60	1.50	...	6.80	7.70	4.90	9.00	0.70	4.20	5.30	5.30	13.50	4.5
2018(B)	5.90	3.90	2.60	12.40	0.30	9.00	1.70	11.30	11.20	3.90	...	14.10	18.80	8.70	21.10	2.20	7.40	7.10	15.90	23.90	10.2
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	.
2001(M)	7.9	6.5	5.7	29.8	4.2	17.6	5.3	17.6	23	6.2	...	16.9	21.8	18.9	35.4	11.5	9.5	10.4	34.3	28.6	20.6
2001(F)	7.9	3	3.4	6.1	0	4	1.3	4.9	7.2	1.5	...	4.1	5.9	4.9	9.9	1.8	5	6.5	13.4	13	5.9
2000(B)	7.7	5.2	4.7	17.6	2	9.2	3.3	11.8	15.8	3.4	...	10	14.5	12	23.2	6.4	7.2	8.5	24	20	13.2
2000(M)	7.6	7.6	5.9	30	4.5	16	5.5	18.8	24.9	5.8	...	16.4	25.7	19.6	36	11.3	9.4	10.5	35.9	28.2	21.0
2000(F)	7.8	2.9	3.5	6.2	0	3.4	1.7	5	7.9	1.3	...	4	5.1	4.8	10.1	1.7	5.2	6.5	14.5	14.2	6.1

61 rows × 184 columns

In [64]:

```
datat.drop(datat.index[0], inplace=True)
```

Save this data, it is now friendly for matplotlib

In [71]:

```
# datat.to_csv('./WHO Data/matplot_data_world_who.csv')
```

In [65]:

```
datat = pd.read_csv('.WHO Data/matplot_data_world_who.csv')
```

In [66]:

```
datat.head(5)
```

Out[66]:

	Unnamed: 0	Afghanistan	Albania	Algeria	Angola	Antigua and Barbuda	Argentina	Armenia	Australia	Austria	...	United States of America	Uruguay	Uzbekistan	Vanuatu	Venezuela (Bolivarian Republic of)	Viet Nam	Yemen	Zambia	Zimbabwe	Global Average
0	2019(B)	6.0	3.7	2.6	12.6	0.3	8.1	2.7	11.3	10.4	...	14.5	18.8	8.3	21.0	2.1	7.2	7.1	14.4	23.6	10.09
1	2019(M)	6.2	5.3	3.3	21.7	0.0	13.5	4.9	17.0	16.6	...	22.4	31.1	11.8	33.1	3.7	10.6	9.0	25.7	37.8	16.14
2	2019(F)	5.7	2.2	1.9	4.7	0.6	3.3	1.0	5.6	4.6	...	6.8	7.7	4.9	9.0	0.7	4.2	5.3	5.3	13.5	4.54
3	2018(B)	5.9	3.9	2.6	12.4	0.3	9.0	1.7	11.3	11.2	...	14.1	18.8	8.7	21.1	2.2	7.4	7.1	15.9	23.9	10.25
4	2018(M)	6.2	5.6	3.2	21.3	0.0	14.9	2.9	17.1	18.0	...	21.8	31.5	12.5	33.5	3.9	10.8	9.0	28.1	38.6	16.43

5 rows × 185 columns

Getting a new dataframe of just overall (both) data of countries

Only getting overall data and not gender specific data

In [67]:

```
p = datat.iloc[0::3, :]
p
```

Out[67]:

	Unnamed: 0	Afghanistan	Albania	Algeria	Angola	Antigua and Barbuda	Argentina	Armenia	Australia	Austria	...	United States of America	Uruguay	Uzbekistan	Vanuatu	Venezuela (Bolivarian Republic of)	Viet Nam	Yemen	Zambia	Zimbabwe	Global Average
0	2019(B)	6.0	3.7	2.6	12.6	0.3	8.1	2.7	11.3	10.4	...	14.5	18.8	8.3	21.0	2.1	7.2	7.1	14.4	23.6	10.09
3	2018(B)	5.9	3.9	2.6	12.4	0.3	9.0	1.7	11.3	11.2	...	14.1	18.8	8.7	21.1	2.2	7.4	7.1	15.9	23.9	10.25
6	2017(B)	6.0	4.1	2.5	12.4	0.0	8.7	2.4	11.8	10.8	...	14.4	18.5	8.9	21.1	2.2	7.5	6.9	16.7	25.9	10.43
9	2016(B)	6.0	4.2	2.6	12.9	0.5	8.1	4.4	10.9	11.1	...	13.7	18.7	9.2	21.0	2.3	7.6	7.0	16.7	28.7	10.68
12	2015(B)	6.0	4.2	2.7	13.3	0.4	8.1	4.7	11.8	11.4	...	12.9	16.7	9.2	21.2	2.4	7.6	6.8	17.3	30.7	10.99
15	2014(B)	6.0	4.5	2.8	13.1	0.4	9.2	4.6	11.3	12.0	...	12.4	15.8	9.0	21.1	2.3	7.6	7.0	17.7	30.8	11.23
18	2013(B)	6.2	4.8	2.9	14.2	0.0	8.5	5.3	10.2	11.5	...	12.0	14.2	8.9	21.8	2.6	7.5	7.1	17.9	31.4	11.42
21	2012(B)	6.2	4.8	2.9	14.2	0.0	9.0	6.4	10.5	11.8	...	12.1	16.1	9.0	21.9	3.4	7.5	7.2	18.5	33.1	11.63
24	2011(B)	6.4	7.6	2.9	13.5	0.2	8.4	5.7	10.1	12.3	...	12.0	15.0	8.6	21.8	3.8	7.4	7.2	19.8	34.3	11.68
27	2010(B)	6.7	7.6	3.0	14.2	0.1	8.4	5.9	10.4	12.2	...	11.7	14.3	8.3	22.1	3.6	7.3	7.4	19.7	35.0	11.88
30	2009(B)	6.8	8.0	3.2	13.6	0.0	8.4	4.1	10.0	11.9	...	11.4	13.4	8.2	22.5	3.4	7.3	7.6	19.0	35.2	12.08
33	2008(B)	7.2	8.1	3.3	15.3	0.2	8.6	4.2	10.4	11.7	...	11.4	13.5	8.9	21.7	3.7	7.4	7.8	19.0	31.7	12.07
36	2007(B)	7.4	8.1	3.5	15.0	0.3	8.2	5.0	9.8	12.2	...	11.2	15.8	9.3	22.0	4.0	7.2	7.7	19.8	27.2	12.02
39	2006(B)	7.6	7.8	3.7	17.1	0.4	8.5	5.4	9.5	12.5	...	11.0	13.8	9.6	22.2	4.3	7.0	7.8	21.8	22.0	12.16
42	2005(B)	7.6	7.7	3.8	16.3	1.3	8.5	3.8	10.1	13.3	...	10.7	13.3	10.2	21.4	4.8	6.9	8.0	22.1	22.0	12.38
45	2004(B)	7.8	4.9	4.0	17.2	2.0	8.7	3.6	9.2	13.6	...	10.8	13.9	9.8	21.9	5.1	7.0	8.1	22.5	21.5	12.42
48	2003(B)	7.7	4.9	4.1	17.5	1.4	9.7	2.5	9.6	14.2	...	10.6	14.2	10.4	22.5	5.6	7.0	8.1	21.5	20.2	12.59
51	2002(B)	7.9	4.8	4.4	17.2	1.3	10.3	3.1	10.4	15.1	...	10.6	17.9	11.3	22.8	6.2	7.0	8.2	22.3	20.3	12.79
54	2001(B)	7.9	4.7	4.6	17.5	1.9	10.3	3.1	11.2	14.7	...	10.3	13.2	11.7	22.8	6.5	7.1	8.4	22.7	19.5	12.94
57	2000(B)	7.7	5.2	4.7	17.6	2.0	9.2	3.3	11.8	15.8	...	10.0	14.5	12.0	23.2	6.4	7.2	8.5	24.0	20.0	13.23

20 rows × 185 columns

Unnamed: 0	Afghanistan	Albania	Algeria	Angola	Antigua and	Argentina	Armenia	Australia	Austria	...	United States of	Uruguay	Uzbekistan	Vanuatu	Venezuela (Bolivarian	Viet	Yemen	Zambia	Zimbabwe	Global

In [61]:

```
p.head(5)
```

Out[61]:

	Unnamed: 0	Afghanistan	Albania	Algeria	Angola	Antigua and Barbuda	Argentina	Armenia	Australia	Austria	...	United States of America	Uruguay	Uzbekistan	Vanuatu	Venezuela (Bolivarian Republic of)	Viet Nam	Yemen	Zambia	Zimbabwe	Global Average
0	2019(B)	6.0	3.7	2.6	12.6	0.3	8.1	2.7	11.3	10.4	...	14.5	18.8	8.3	21.0	2.1	7.2	7.1	14.4	23.6	10.09
3	2018(B)	5.9	3.9	2.6	12.4	0.3	9.0	1.7	11.3	11.2	...	14.1	18.8	8.7	21.1	2.2	7.4	7.1	15.9	23.9	10.25
6	2017(B)	6.0	4.1	2.5	12.4	0.0	8.7	2.4	11.8	10.8	...	14.4	18.5	8.9	21.1	2.2	7.5	6.9	16.7	25.9	10.43
9	2016(B)	6.0	4.2	2.6	12.9	0.5	8.1	4.4	10.9	11.1	...	13.7	18.7	9.2	21.0	2.3	7.6	7.0	16.7	28.7	10.68
12	2015(B)	6.0	4.2	2.7	13.3	0.4	8.1	4.7	11.8	11.4	...	12.9	16.7	9.2	21.2	2.4	7.6	6.8	17.3	30.7	10.99

5 rows × 185 columns

In [68]:

```
p.reset_index(inplace=True, drop=True)
```

In [82]:

```
p
```

Out[82]:

	Year	Afghanistan	Albania	Algeria	Angola	Antigua and Barbuda	Argentina	Armenia	Australia	Austria	...	United States of America	Uruguay	Uzbekistan	Vanuatu	Venezuela (Bolivarian Republic of)	Viet Nam	Yemen	Zambia	Zimbabwe	Global Average
19	2000	7.7	5.2	4.7	17.6	2.0	9.2	3.3	11.8	15.8	...	10.0	14.5	12.0	23.2	6.4	7.2	8.5	24.0	20.0	13.23
18	2001	7.9	4.7	4.6	17.5	1.9	10.3	3.1	11.2	14.7	...	10.3	13.2	11.7	22.8	6.5	7.1	8.4	22.7	19.5	12.94
17	2002	7.9	4.8	4.4	17.2	1.3	10.3	3.1	10.4	15.1	...	10.6	17.9	11.3	22.8	6.2	7.0	8.2	22.3	20.3	12.79
16	2003	7.7	4.9	4.1	17.5	1.4	9.7	2.5	9.6	14.2	...	10.6	14.2	10.4	22.5	5.6	7.0	8.1	21.5	20.2	12.59
15	2004	7.8	4.9	4.0	17.2	2.0	8.7	3.6	9.2	13.6	...	10.8	13.9	9.8	21.9	5.1	7.0	8.1	22.5	21.5	12.42
14	2005	7.6	7.7	3.8	16.3	1.3	8.5	3.8	10.1	13.3	...	10.7	13.3	10.2	21.4	4.8	6.9	8.0	22.1	22.0	12.38
13	2006	7.6	7.8	3.7	17.1	0.4	8.5	5.4	9.5	12.5	...	11.0	13.8	9.6	22.2	4.3	7.0	7.8	21.8	22.0	12.16
12	2007	7.4	8.1	3.5	15.0	0.3	8.2	5.0	9.8	12.2	...	11.2	15.8	9.3	22.0	4.0	7.2	7.7	19.8	27.2	12.02
11	2008	7.2	8.1	3.3	15.3	0.2	8.6	4.2	10.4	11.7	...	11.4	13.5	8.9	21.7	3.7	7.4	7.8	19.0	31.7	12.07
10	2009	6.8	8.0	3.2	13.6	0.0	8.4	4.1	10.0	11.9	...	11.4	13.4	8.2	22.5	3.4	7.3	7.6	19.0	35.2	12.08
9	2010	6.7	7.6	3.0	14.2	0.1	8.4	5.9	10.4	12.2	...	11.7	14.3	8.3	22.1	3.6	7.3	7.4	19.7	35.0	11.88
8	2011	6.4	7.6	2.9	13.5	0.2	8.4	5.7	10.1	12.3	...	12.0	15.0	8.6	21.8	3.8	7.4	7.2	19.8	34.3	11.68
7	2012	6.2	4.8	2.9	14.2	0.0	9.0	6.4	10.5	11.8	...	12.1	16.1	9.0	21.9	3.4	7.5	7.2	18.5	33.1	11.63
6	2013	6.2	4.8	2.9	14.2	0.0	8.5	5.3	10.2	11.5	...	12.0	14.2	8.9	21.8	2.6	7.5	7.1	17.9	31.4	11.42
5	2014	6.0	4.5	2.8	13.1	0.4	9.2	4.6	11.3	12.0	...	12.4	15.8	9.0	21.1	2.3	7.6	7.0	17.7	30.8	11.23
4	2015	6.0	4.2	2.7	13.3	0.4	8.1	4.7	11.8	11.4	...	12.9	16.7	9.2	21.2	2.4	7.6	6.8	17.3	30.7	10.99
3	2016	6.0	4.2	2.6	12.9	0.5	8.1	4.4	10.9	11.1	...	13.7	18.7	9.2	21.0	2.3	7.6	7.0	16.7	28.7	10.68
2	2017	6.0	4.1	2.5	12.4	0.0	8.7	2.4	11.8	10.8	...	14.4	18.5	8.9	21.1	2.2	7.5	6.9	16.7	25.9	10.43
1	2018	5.9	3.9	2.6	12.4	0.3	9.0	1.7	11.3	11.2	...	14.1	18.8	8.7	21.1	2.2	7.4	7.1	15.9	23.9	10.25
0	2019	6.0	3.7	2.6	12.6	0.3	8.1	2.7	11.3	10.4	...	14.5	18.8	8.3	21.0	2.1	7.2	7.1	14.4	23.6	10.09

20 rows × 185 columns

In [70]:

```
# Stripping the (B) since it is just overall for the whole dataframe
p['Unnamed: 0'] = p['Unnamed: 0'].str.strip(')')
p['Unnamed: 0'] = p['Unnamed: 0'].str.strip('B')
p['Unnamed: 0'] = p['Unnamed: 0'].str.strip('(')
```

```
C:\Users\rupes\AppData\Local\Temp/ipykernel_756/1515189540.py:2: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  p['Unnamed: 0'] = p['Unnamed: 0'].str.strip(')')
C:\Users\rupes\AppData\Local\Temp/ipykernel_756/1515189540.py:3: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  p['Unnamed: 0'] = p['Unnamed: 0'].str.strip('B')
C:\Users\rupes\AppData\Local\Temp/ipykernel_756/1515189540.py:4: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  p['Unnamed: 0'] = p['Unnamed: 0'].str.strip('(')
```

In [71]:

```
p.tail()
```

Out[71]:

	Unnamed: 0	Afghanistan	Albania	Algeria	Angola	Antigua and Barbuda	Argentina	Armenia	Australia	Austria	...	United States of America	Uruguay	Uzbekistan	Vanuatu	Venezuela (Bolivarian Republic of)	Viet Nam	Yemen	Zambia	Zimbabwe	Global Average
15	2004	7.8	4.9	4.0	17.2	2.0	8.7	3.6	9.2	13.6	...	10.8	13.9	9.8	21.9	5.1	7.0	8.1	22.5	21.5	12.42
16	2003	7.7	4.9	4.1	17.5	1.4	9.7	2.5	9.6	14.2	...	10.6	14.2	10.4	22.5	5.6	7.0	8.1	21.5	20.2	12.59
17	2002	7.9	4.8	4.4	17.2	1.3	10.3	3.1	10.4	15.1	...	10.6	17.9	11.3	22.8	6.2	7.0	8.2	22.3	20.3	12.79

	Unnamed: 0	Afghanistan	Albania	Algeria	Angola	Antigua and Barbuda	Argentina	Armenia	Australia	Austria	Azerbaijan	...	United States of America	Uruguay	Uzbekistan	Vanuatu	Venezuela (Bolivarian Republic of)	Viet Nam	Yemen	Zambia	Zimbabwe	Global Average
18	2001	7.9	4.7	4.6	17.5	1.9	10.3	3.1	11.2	14.7	...	10.3	13.2	11.7	22.8	6.5	7.1	8.4	22.7	19.5	12.94
19	2000	7.7	5.2	4.7	17.6	2.0	9.2	3.3	11.8	15.8	3.4	...	10.0	14.5	12.0	23.2	6.4	7.2	8.5	24.0	20.0	13.23

5 rows × 185 columns

In [72]:

```
p.rename(columns={'Unnamed: 0':'Year'}, inplace=True)
```

```
C:\Users\rupes\AppData\Local\Programs\Python\Python310\lib\site-packages\pandas\core\frame.py:5039: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  return super().rename(
```

In [73]:

```
p.sort_values('Year', ascending=True, inplace=True)
```

```
C:\Users\rupes\AppData\Local\Programs\Python\Python310\lib\site-packages\pandas\util\_decorators.py:311: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  return func(*args, **kwargs)
```

In [74]:

```
p.set_index('Year', drop=True)
```

Out[74]:

Year	Afghanistan	Albania	Algeria	Angola	Antigua and Barbuda	Argentina	Armenia	Australia	Austria	Azerbaijan	...	United States of America	Uruguay	Uzbekistan	Vanuatu	Venezuela (Bolivarian Republic of)	Viet Nam	Yemen	Zambia	Zimbabwe	Global Average
2000	7.7	5.2	4.7	17.6	2.0	9.2	3.3	11.8	15.8	3.4	...	10.0	14.5	12.0	23.2	6.4	7.2	8.5	24.0	20.0	13.23
2001	7.9	4.7	4.6	17.5	1.9	10.3	3.1	11.2	14.7	3.7	...	10.3	13.2	11.7	22.8	6.5	7.1	8.4	22.7	19.5	12.94
2002	7.9	4.8	4.4	17.2	1.3	10.3	3.1	10.4	15.1	3.9	...	10.6	17.9	11.3	22.8	6.2	7.0	8.2	22.3	20.3	12.79
2003	7.7	4.9	4.1	17.5	1.4	9.7	2.5	9.6	14.2	4.0	...	10.6	14.2	10.4	22.5	5.6	7.0	8.1	21.5	20.2	12.59
2004	7.8	4.9	4.0	17.2	2.0	8.7	3.6	9.2	13.6	4.5	...	10.8	13.9	9.8	21.9	5.1	7.0	8.1	22.5	21.5	12.42
2005	7.6	7.7	3.8	16.3	1.3	8.5	3.8	10.1	13.3	4.7	...	10.7	13.3	10.2	21.4	4.8	6.9	8.0	22.1	22.0	12.38
2006	7.6	7.8	3.7	17.1	0.4	8.5	5.4	9.5	12.5	4.9	...	11.0	13.8	9.6	22.2	4.3	7.0	7.8	21.8	22.0	12.16
2007	7.4	8.1	3.5	15.0	0.3	8.2	5.0	9.8	12.2	5.0	...	11.2	15.8	9.3	22.0	4.0	7.2	7.7	19.8	27.2	12.02
2008	7.2	8.1	3.3	15.3	0.2	8.6	4.2	10.4	11.7	4.8	...	11.4	13.5	8.9	21.7	3.7	7.4	7.8	19.0	31.7	12.07
2009	6.8	8.0	3.2	13.6	0.0	8.4	4.1	10.0	11.9	4.6	...	11.4	13.4	8.2	22.5	3.4	7.3	7.6	19.0	35.2	12.08
2010	6.7	7.6	3.0	14.2	0.1	8.4	5.9	10.4	12.2	4.5	...	11.7	14.3	8.3	22.1	3.6	7.3	7.4	19.7	35.0	11.88
2011	6.4	7.6	2.9	13.5	0.2	8.4	5.7	10.1	12.3	4.4	...	12.0	15.0	8.6	21.8	3.8	7.4	7.2	19.8	34.3	11.68
2012	6.2	4.8	2.9	14.2	0.0	9.0	6.4	10.5	11.8	4.3	...	12.1	16.1	9.0	21.9	3.4	7.5	7.2	18.5	33.1	11.63
2013	6.2	4.8	2.9	14.2	0.0	8.5	5.3	10.2	11.5	4.2	...	12.0	14.2	8.9	21.8	2.6	7.5	7.1	17.9	31.4	11.42
2014	6.0	4.5	2.8	13.1	0.4	9.2	4.6	11.3	12.0	4.1	...	12.4	15.8	9.0	21.1	2.3	7.6	7.0	17.7	30.8	11.23
2015	6.0	4.2	2.7	13.3	0.4	8.1	4.7	11.8	11.4	4.0	...	12.9	16.7	9.2	21.2	2.4	7.6	6.8	17.3	30.7	10.99
2016	6.0	4.2	2.6	12.9	0.5	8.1	4.4	10.9	11.1	3.9	...	13.7	18.7	9.2	21.0	2.3	7.6	7.0	16.7	28.7	10.68
2017	6.0	4.1	2.5	12.4	0.0	8.7	2.4	11.8	10.8	3.9	...	14.4	18.5	8.9	21.1	2.2	7.5	6.9	16.7	25.9	10.43
2018	5.9	3.9	2.6	12.4	0.3	9.0	1.7	11.3	11.2	3.9	...	14.1	18.8	8.7	21.1	2.2	7.4	7.1	15.9	23.9	10.25
2019	6.0	3.7	2.6	12.6	0.3	8.1	2.7	11.3	10.4	4.0	...	14.5	18.8	8.3	21.0	2.1	7.2	7.1	14.4	23.6	10.09

20 rows × 184 columns

In [69]:

```
p.India
```

Out[69]:

```
19    19.1
18    18.7
17    18.2
16    17.3
15    16.8
14    17.0
13    17.1
12    16.7
11    16.3
10    15.6
9     15.6
8     15.5
7     15.1
6     14.4
5     13.5
4     12.9
3     12.6
2     12.5
1     12.9
0     12.9
Name: India, dtype: float64
```

In [83]:

```
# p.to_csv('./WHO Data/consolidate.csv')
# pd.read_csv('./WHO Data/consolidate.csv')
```

Plotting

In [75]:

```
p.head()
```

Out[75]:

	Year	Afghanistan	Albania	Algeria	Angola	Antigua and Barbuda	Argentina	Armenia	Australia	Austria	...	United States of America	Uruguay	Uzbekistan	Vanuatu	Venezuela (Bolivarian Republic of)	Viet Nam	Yemen	Zambia	Zimbabwe	Global Average
19	2000	7.7	5.2	4.7	17.6	2.0	9.2	3.3	11.8	15.8	...	10.0	14.5	12.0	23.2	6.4	7.2	8.5	24.0	20.0	13.23
18	2001	7.9	4.7	4.6	17.5	1.9	10.3	3.1	11.2	14.7	...	10.3	13.2	11.7	22.8	6.5	7.1	8.4	22.7	19.5	12.94
17	2002	7.9	4.8	4.4	17.2	1.3	10.3	3.1	10.4	15.1	...	10.6	17.9	11.3	22.8	6.2	7.0	8.2	22.3	20.3	12.79
16	2003	7.7	4.9	4.1	17.5	1.4	9.7	2.5	9.6	14.2	...	10.6	14.2	10.4	22.5	5.6	7.0	8.1	21.5	20.2	12.59
15	2004	7.8	4.9	4.0	17.2	2.0	8.7	3.6	9.2	13.6	...	10.8	13.9	9.8	21.9	5.1	7.0	8.1	22.5	21.5	12.42

5 rows × 185 columns

In [71]:

```
# All the pre available style that can be used for plotting
print(plt.style.available)
```

```
['Solarize_Light2', '_classic_test_patch', '_mpl-gallery', '_mpl-gallery-nogrid', 'bmh', 'classic', 'dark_background', 'fast',
'fivethirtyeight', 'ggplot', 'grayscale', 'seaborn', 'seaborn-bright', 'seaborn-colorblind', 'seaborn-dark', 'seaborn-dark-palette', 'seaborn-
darkgrid', 'seaborn-deep', 'seaborn-muted', 'seaborn-notebook', 'seaborn-paper', 'seaborn-pastel', 'seaborn-poster', 'seaborn-talk', 'seaborn-
ticks', 'seaborn-white', 'seaborn-whitegrid', 'tableau-colorblind10']
```

Start with Analyzing Data of some neighbouring countries and comparing it with India

In [76]:

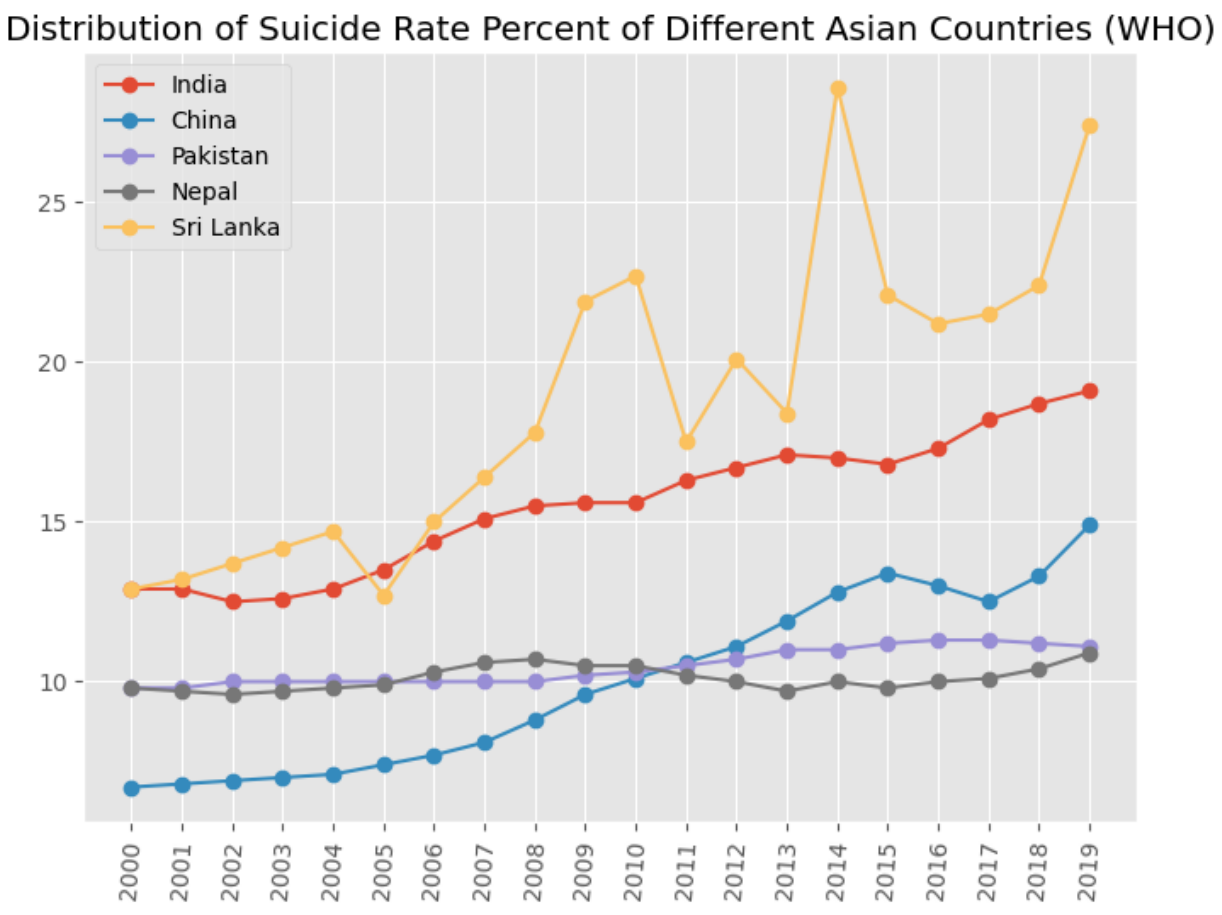
```
## Taking Year in desired sequence
Year = np.arange(2019,1999,-1)
Year
```

Out[76]:

```
array([2019, 2018, 2017, 2016, 2015, 2014, 2013, 2012, 2011, 2010, 2009,
       2008, 2007, 2006, 2005, 2004, 2003, 2002, 2001, 2000])
```

In [77]:

```
plt.style.use('ggplot')
plt.figure(figsize=(8,6), dpi=100)
plt.title('Distribution of Suicide Rate Percent of Different Asian Countries (WHO)')
plt.plot(Year, p.India, marker='o', label='India')
plt.plot(Year, p.China, marker='o', label='China')
plt.plot(Year, p.Pakistan, marker='o', label='Pakistan')
plt.plot(Year, p.Nepal, marker='o', label='Nepal')
plt.plot(Year, p['Sri Lanka'], marker='o', label='Sri Lanka')
plt.xticks(np.arange(2000,2020,1), rotation=86)
plt.legend()
# plt.savefig('asian_countries_who.png', bbox_inches = 'tight', facecolor='white', transparent = False, dpi=300)
plt.show()
```

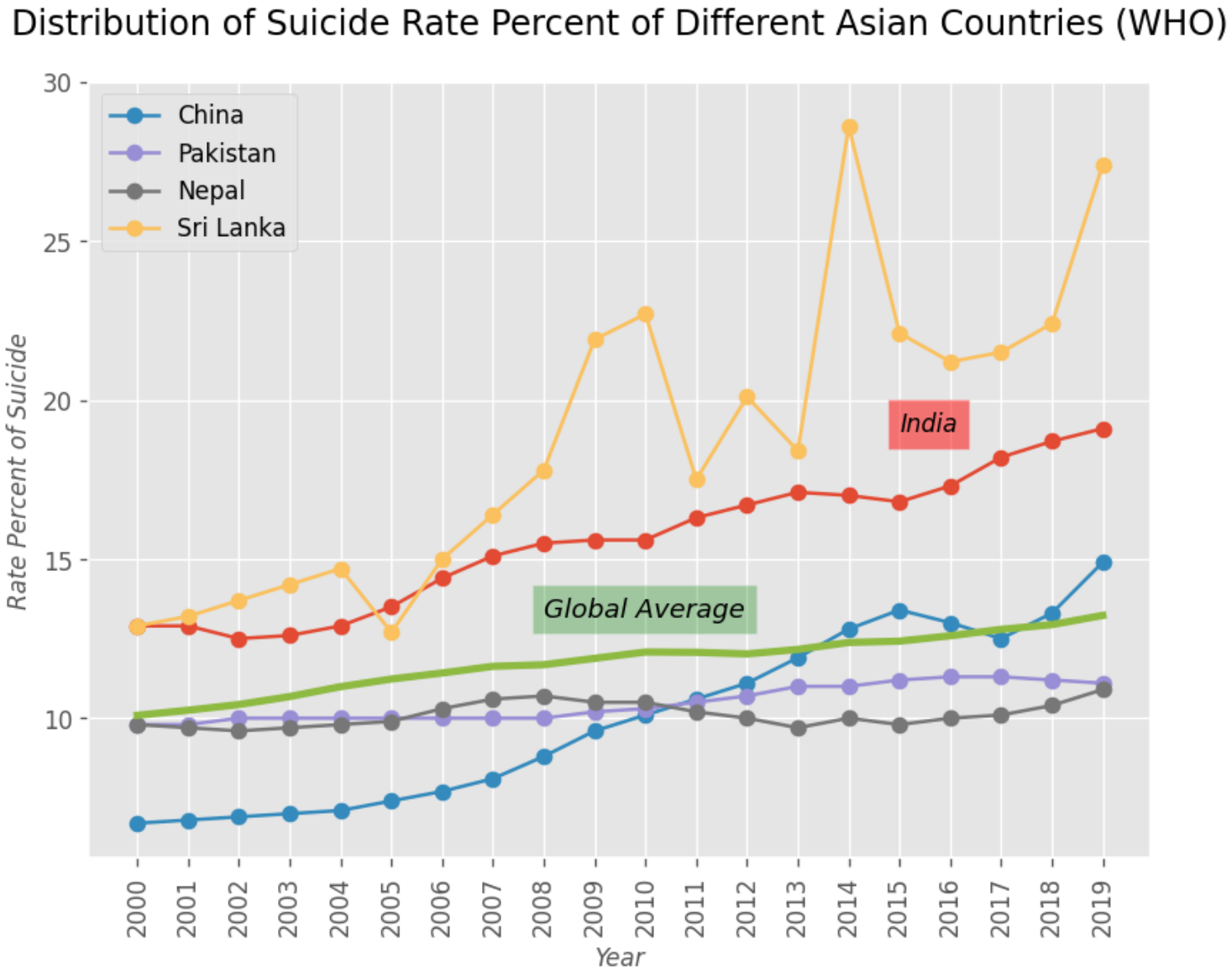


Comparing the Asian countries with the Global Average

In [78]:

```
plt.style.use('ggplot')
plt.figure(figsize=(8,6), dpi=120)
plt.title('Distribution of Suicide Rate Percent of Different Asian Countries (WHO)', pad=20)
plt.plot(Year, p.India, marker='o')
plt.plot(Year, p.China, marker='o', label='China')
plt.plot(Year, p.Pakistan, marker='o', label='Pakistan')
plt.plot(Year, p.Nepal, marker='o', label='Nepal')
plt.plot(Year, p['Sri Lanka'], marker='o', label='Sri Lanka')

plt.plot(Year, p['Global Average'], linewidth=3.1)

### Adding a simple text and bbox
plt.text(2008, 13.2, 'Global Average', style='italic', fontsize=11, bbox={'facecolor': 'green', 'alpha': 0.3, 'pad': 5})
plt.text(2015, 19, 'India', style='italic', fontsize=10, bbox={'facecolor': 'red', 'alpha': 0.5, 'pad': 5})
###
plt.xlabel('Year', fontsize=10, style='italic')
plt.ylabel('Rate Percent of Suicide', fontsize=10, style='italic')
plt.xticks(np.arange(2000,2020,1), rotation=90)
plt.yticks(np.arange(10,35,5))
```

```
plt.legend()
# plt.savefig('asiavsglobal_who.png', bbox_inches = 'tight', facecolor='white', transparent = False, dpi=300)
plt.show()
```



Conclusion:

From the above plot it is more than just clear that India is very poor as compared to the Global Average even behind countries like Pakistan. The only other Asian country performing bad than India is Sri Lanka.

Also, a matter of concern is that the graph is diving nose up in case of India and every projection indicated that the situation will get worse in the coming years.

In [86]:

```
##### 2. <u>NCRB Data Analysis and Visualization 2016-2020</u>
   ###### 2a. Analysis of Trend in this timeline in India
   ###### 2b. Zooming in more on Year 2018, 2019, 2020 (State wise)
   ###### 2c. Specific Analysis of Post Covid India (City wise, Cause wise, Age Group wise)
```

2. NCRB Data Analysis and Visualization 2016-2020

We start with importing the pre-downloaded (csv) data of NCRB ADSI Annual reports for the Year 2018, 2019, 2020 and merging it to make a master dataframe to visualize

In [87]:

```
import pandas as pd
```

In [88]:

```
state18 = pd.read_csv('./2018 ncrb/2018 state.csv')
state19 = pd.read_csv('./2019 ncrb/2019 state.csv')
state20 = pd.read_csv('./2020 ncrb/2020 state.csv')
```

In [90]:

```
# we see that the state name is in caps only in 2019 dataset. Converting it to titlecase before merging
state19['State/UT/City (Col. 2)'] = state19['State/UT/City (Col. 2)'].str.title()
```

In [91]:

```
state19.head(2)
```

Out[91]:

	Unnamed: 0	Category (Col. 1)	State/UT/City (Col. 2)	Number of Suicides (Col.3)	Percentage Share in Total Suicides (Col. 4)	FALSE	Rate of Suicides (Col. 6 = Col.3/Col.5)
0	0	State	Andhra Pradesh	6465	4.6	523.2	12.4
1	1	State	Arunachal Pradesh	112	0.1	15.1	7.4

Merging dataset of 2018, 2019, 2020 of NCRB State wise distribution

In [92]:

```
df = pd.merge(state18, state19, how='left', left_on=['State/UT - (Col.3)'], right_on=['State/UT/City (Col. 2)'])
```

In [93]:

```
state20.head(2)
```

Out[93]:

	Si. No. (Col.1)	Category	State/UT/ (Col.2)	Number of Suicides(Col.3)	Percentage Share in Total Suicides (Col.4)	Projected Mid-Year Population # (in Lakh)\n (in Lakh)\n(Col.5)	Rate of Suicides (Col.3/Col.5)
0	1	State	Andhra Pradesh	7043	4.6	526.0	13.4

	Si. No. (Col.1)	Category	State/UT/ (Col.2)	Number of Suicides(Col.3)	Percentage Share in Total Suicides (Col.4)	Projected Mid-Year Population # (in Lakh)\n (in Lakh)\n(Col.5)	Rate of Suicides (Col.3/Col.5)
1	2	State	Arunachal Pradesh	160	0.1	15.2	10.5

In [94]:

```
dffinal = pd.merge(df, state20, how='left', left_on=['State/UT/City (Col. 2)'], right_on=['State/UT/ (Col.2)'])
```

Dropping extra columns in the merged dataframe

In [96]:

```
dffinal.drop(columns=['Unnamed: 0', 'Category (Col. 1)', 'State/UT/City (Col. 2)', 'Si. No. (Col.1)', 'Category', 'State/UT/ (Col.2)'], inplace=True)
```

In [97]:

```
dffinal.head(3)
```

Out[97]:

	Sl. No. - (Col.1)	Category - (Col.2)	State/UT - (Col.3)	Number of Suicides - (Col.4)	Percentage Share in Total Suicides - (Col.5)	Projected Mid-Year Population # (in Lakh) - (Col.6)	Rate of Suicides - (Col.7 = Col.4/Col.6)	Number of Suicides (Col.3)	Percentage Share in Total Suicides (Col. 4)	FALSE	Rate of Suicides (Col. 6 = Col.3/Col.5)	Number of Suicides(Col.3)	Percentage Share in Total Suicides (Col.4)	Projected Mid-Year Population # (in Lakh)\n (in Lakh)\n(Col.5)	Rate of Suicides (Col.3/Col.5)
0	1	State	Andhra Pradesh	5319	4.0	520.3	10.2	6465.0	4.6	523.2	12.4	7043.0	4.6	526.0	13.4
1	2	State	Arunachal Pradesh	132	0.1	14.9	8.9	112.0	0.1	15.1	7.4	160.0	0.1	15.2	10.5
2	3	State	Assam	2379	1.8	340.4	7.0	2370.0	1.7	344.2	6.9	3243.0	2.1	347.9	9.3

Saving the Dataframe

In [79]:

```
# dffinal.to_csv('finalmergetomanual.csv')
```

Since, we observed that after merging some data is lost therefore we do some manual filtering of data as well in Google Sheets and then import it again in the Notebook.

Reloading the clean merged dataframe

After we did the manual insertion of data and renamed some of columns we are reloading the overall and statewise distribution of data of NCRB to start analyzing the data and visualizing it

In [98]:

```
df1 = pd.read_csv('./2016-20 ncrb/ncrb_suiciderate_india_16-20.csv')
df2 = pd.read_csv('./2016-20 ncrb/finalcleanstate.csv')
```

In [99]:

```
df1.head()
```

Out[99]:

	Sl. No.	Year	Total Number of Suicides	Mid-Year Projected Population*(in Lakh+)	Rate of Suicides (Col.3/Col.4)
0	1	2016	1,31,008	12,739.90	10.3
1	2	2017	1,29,887	13091.6	9.9
2	3	2018	1,34,516	13233.8	10.2
3	4	2019	1,39,123	13376.1	10.4
4	5	2020	1,53,052	13533.9	11.3

Distribution overall (df1)

In [100]:

```
df1
```

Out[100]:

	Sl. No.	Year	Total Number of Suicides	Mid-Year Projected Population*(in Lakh+)	Rate of Suicides (Col.3/Col.4)
0	1	2016	1,31,008	12,739.90	10.3
1	2	2017	1,29,887	13091.6	9.9
2	3	2018	1,34,516	13233.8	10.2
3	4	2019	1,39,123	13376.1	10.4
4	5	2020	1,53,052	13533.9	11.3

Since, the values are in str and also contains commas we have to filter thr data

In [101]:

```
# Replacing commas with no space and converting to int type
df1['Total Number of Suicides'] = df1['Total Number of Suicides'].str.replace(',', '').astype(int)
```

In [102]:

```
df1
```

Out[102]:

	Sl. No.	Year	Total Number of Suicides	Mid-Year Projected Population*(in Lakh+)	Rate of Suicides (Col.3/Col.4)
0	1	2016	131008	12,739.90	10.3
1	2	2017	129887	13091.6	9.9
2	3	2018	134516	13233.8	10.2

	Sl. No.	Year	Total Number of Suicides	Mid-Year Projected Population*(in Lakh+)	Rate of Suicides (Col.3/Col.4)
3	4	2019	139123	13376.1	10.4
4	5	2020	153052	13533.9	11.3

In [103]:

```
df1[' Mid-Year Projected Population*(in Lakh+)'] = df1[' Mid-Year Projected Population*(in Lakh+)'].str.replace(',','').astype(float)
```

In [104]:

```
a = df1['Rate of Suicides (Col.3/Col.4)'][0]
type(a)
```

Out[104]:

```
numpy.float64
```

Now, the data are in desired format

2a. Analysis of Trend in this timeline in India

Plotting Overall Data

In [105]:

```
print(plt.style.available)
```

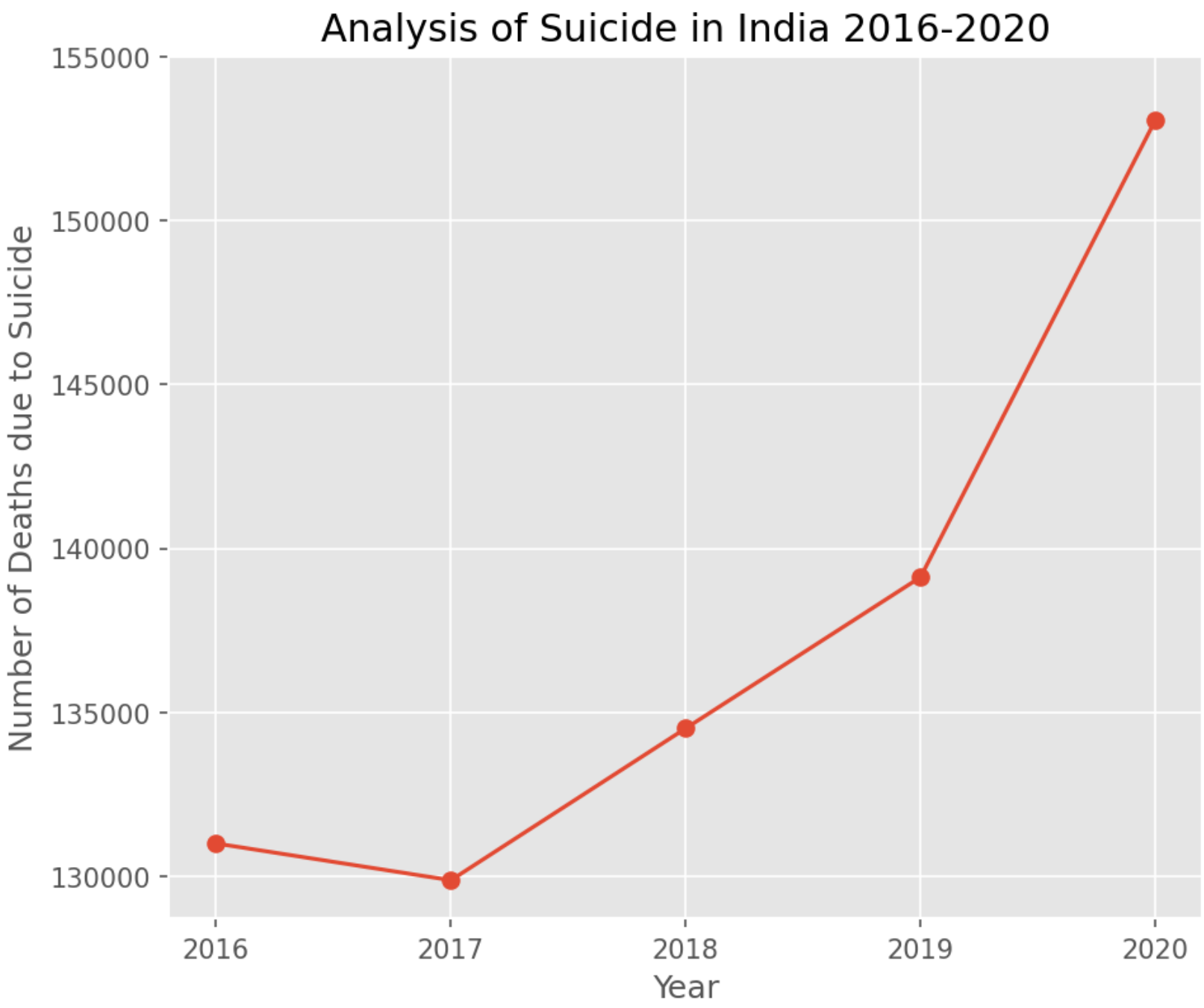
```
['Solarize_Light2', '_classic_test_patch', '_mpl-gallery', '_mpl-gallery-nogrid', 'bmh', 'classic', 'dark_background', 'fast',
'fivethirtyeight', 'ggplot', 'grayscale', 'seaborn', 'seaborn-bright', 'seaborn-colorblind', 'seaborn-dark', 'seaborn-dark-palette', 'seaborn-
darkgrid', 'seaborn-deep', 'seaborn-muted', 'seaborn-notebook', 'seaborn-paper', 'seaborn-pastel', 'seaborn-poster', 'seaborn-talk', 'seaborn-
ticks', 'seaborn-white', 'seaborn-whitegrid', 'tableau-colorblind10']
```

In [15]:

```
import matplotlib.pyplot as plt
import numpy as np
```

In [109]:

```
plt.style.use('ggplot')

plt.figure(figsize=(7,6), dpi=150)

plt.title('Analysis of Suicide in India 2016-2020')
# plt.bar(df1['Year'], df1['Total Number of Suicides'], color='#f24e07')
plt.plot(df1['Year'], df1['Total Number of Suicides'], marker='o')
plt.xticks(np.arange(2016,2021,1))
plt.yticks(np.arange(130000,160000,5000))
plt.xlabel('Year')
plt.ylabel('Number of Deaths due to Suicide')
# plt.savefig('ncrb_india_overall.png', bbox_inches = 'tight', facecolor='white', transparent = False, dpi=300)
plt.show()
```



The NCRB data also aligns with all the other global/national data and indicates that the suicide rate in India is on rise and was at peak in the Covid Era. This is a matter of deep concern with country with a huge young population.

The number of deaths due to Suicide in India in the Year 2020 was as high as 1,53,052

The number of deaths due to Suicide in India in the Year 2020 was as high as 1,53,052

***This rounds to at least 419 deaths in a day**

***That means 17 deaths per Hour in the Year 2020**

Suicide increased by 10% in India in Covid-hit 2020

2b. Zooming in more on Year 2018, 2019, 2020 (State wise)

Now, analyzing the state wise record of NCRB

In [110]:

```
df2.tail()
```

Out[110]:

	Unnamed: 0	Sr. No.	Category	State/UT	Number of Suicides (N)18	Number of Suicides (N)19	Number of Suicides (N)20	Percentage Share in Total Suicides18	Percentage Share in Total Suicides19	Percentage Share in Total Suicides20	Rate of Suicides - (N/P)18	Rate of Suicides - (N/P)19	Rate of Suicides - (N/P)20	Projected Mid-Year Population # (in Lakh)(P)18	Projected Mid-Year Population # (in Lakh)(P)19	Projected Mid-Year Population # (in Lakh)(P)20
34	34	34	Union Territory	Delhi (UT)	2526	2526	3142	1.9	1.8	2.1	12.9	12.7	15.5	195.6	199.4	203.2
35	35	35	Union Territory	Lakshadweep	3	0	2	0.0	0.0	0	4.3	0.0	2.9	0.7	0.7	0.7
36	36	36	Union Territory	Puducherry	500	493	408	0.4	0.4	0.3	33.8	32.5	26.3	14.8	15.2	15.5
37	37	Union Territory	Union Territory	Total (UTs)	3489	3472	4315	2.6	2.5	2.8	14.8	14.4	11.3	236.1	240.8	382.1
38	38	Total (All India)	Total (All India)	Total (All India)	134516	139123	153052	100.0	100.0	100	10.2	10.4	11.3	13233.8	13376.1	13533.9

Ignoring Total rows

In [111]:

```
# Removing total rows

df2 = df2.loc[~df2['State/UT'].str.contains('Total')]
```

In [112]:

```
df2.iloc[:, 0:7]
```

Out[112]:

	Unnamed: 0	Sr. No.	Category	State/UT	Number of Suicides (N)18	Number of Suicides (N)19	Number of Suicides (N)20
0	0	1	State	Andhra Pradesh	5319	6465	7043
1	1	2	State	Arunachal Pradesh	132	112	160
2	2	3	State	Assam	2379	2370	3243
3	3	4	State	Bihar	443	641	809
4	4	5	State	Chhattisgarh	7046	7629	7710
5	5	6	State	Goa	256	259	308
6	6	7	State	Gujarat	7793	7655	8050
7	7	8	State	Haryana	3547	4191	4001
8	8	9	State	Himachal Pradesh	740	584	857
9	9	10	State	Jammu & Kashmir	330	284	287
10	10	11	State	Jharkhand	1317	1646	2145
11	11	12	State	Karnataka	11561	11288	12259
12	12	13	State	Kerala	8237	8556	8500
13	13	14	State	Madhya Pradesh	11775	12457	14578
14	14	15	State	Maharashtra	17972	18916	19909
15	15	16	State	Manipur	52	58	44
16	16	17	State	Meghalaya	189	198	224
17	17	18	State	Mizoram	79	70	108
18	18	19	State	Nagaland	36	41	48
19	19	20	State	Odisha	4592	4582	5546
20	20	21	State	Punjab	1714	2357	2616
21	21	22	State	Rajasthan	4333	4531	5658
22	22	23	State	Sikkim	199	220	285
23	23	24	State	Tamil Nadu	13896	13493	16883
24	24	25	State	Telangana	7845	7675	8058
25	25	26	State	Tripura	720	728	845
26	26	27	State	Uttar Pradesh	4849	5464	4804
27	27	28	State	Uttarakhand	421	516	943
28	28	29	State	West Bengal	13255	12665	13103
30	30	30	Union Territory	A & N Islands	164	181	180
31	31	31	Union Territory	Chandigarh	160	131	128
32	32	32	Union Territory	D & N Haveli	101	95	156
33	33	33	Union Territory	Daman & Diu	35	46	-
34	34	34	Union Territory	Delhi (UT)	2526	2526	3142
35	35	35	Union Territory	Lakshadweep	3	0	2
36	36	36	Union Territory	Puducherry	500	493	408

There is a problem, i.e., Daman and Diu has no data available for 2020 therefore drop it

In [113]:

```
df2.loc[df2['State/UT'].str.contains('Daman & Diu')]
```

Out[113]:

	Unnamed: 0	Sr. No.	Category	State/UT	Number of Suicides (N)18	Number of Suicides (N)19	Number of Suicides (N)20	Percentage Share in Total Suicides18	Percentage Share in Total Suicides19	Percentage Share in Total Suicides20	Rate of Suicides - (N/P)18	Rate of Suicides - (N/P)19	Rate of Suicides - (N/P)20	Projected Mid-Year Population # (in Lakh)(P)18	Projected Mid-Year Population # (in Lakh)(P)19	Projected Mid-Year Population # (in Lakh)(P)20
33	33	33	Union Territory	Daman & Diu	35	46	-	0.0	0.0	-	8.8	10.9	-	4.0	4.2	-

In [114]:

```
df2 = df2.loc[~df2['State/UT'].str.contains('Daman & Diu')]
```

In [115]:

```
df2.head(3)
```

Out[115]:

	Unnamed: 0	Sr. No.	Category	State/UT	Number of Suicides (N)18	Number of Suicides (N)19	Number of Suicides (N)20	Percentage Share in Total Suicides18	Percentage Share in Total Suicides19	Percentage Share in Total Suicides20	Rate of Suicides - (N/P)18	Rate of Suicides - (N/P)19	Rate of Suicides - (N/P)20	Projected Mid-Year Population # (in Lakh)(P)18	Projected Mid-Year Population # (in Lakh)(P)19	Projected Mid-Year Population # (in Lakh)(P)20
0	0	1	State	Andhra Pradesh	5319	6465	7043	4.0	4.6	4.6	10.2	12.4	13.4	520.3	523.2	526
1	1	2	State	Arunachal Pradesh	132	112	160	0.1	0.1	0.1	8.9	7.4	10.5	14.9	15.1	15.2
2	2	3	State	Assam	2379	2370	3243	1.8	1.7	2.1	7.0	6.9	9.3	340.4	344.2	347.9

Plotting Statewise Distribution 2018-20

In [116]:

```
import matplotlib.pyplot as plt
import numpy as np
```

In [117]:

```
df2
```

Out[117]:

	Unnamed: 0	Sr. No.	Category	State/UT	Number of Suicides (N)18	Number of Suicides (N)19	Number of Suicides (N)20	Percentage Share in Total Suicides18	Percentage Share in Total Suicides19	Percentage Share in Total Suicides20	Rate of Suicides - (N/P)18	Rate of Suicides - (N/P)19	Rate of Suicides - (N/P)20	Projected Mid-Year Population # (in Lakh)(P)18	Projected Mid-Year Population # (in Lakh)(P)19	Projected Mid-Year Population # (in Lakh)(P)20
0	0	1	State	Andhra Pradesh	5319	6465	7043	4.0	4.6	4.6	10.2	12.4	13.4	520.3	523.2	526
1	1	2	State	Arunachal Pradesh	132	112	160	0.1	0.1	0.1	8.9	7.4	10.5	14.9	15.1	15.2
2	2	3	State	Assam	2379	2370	3243	1.8	1.7	2.1	7.0	6.9	9.3	340.4	344.2	347.9
3	3	4	State	Bihar	443	641	809	0.3	0.5	0.5	0.4	0.5	0.7	1183.3	1201.1	1219
4	4	5	State	Chhattisgarh	7046	7629	7710	5.2	5.5	5	24.7	26.4	26.4	284.7	288.5	292.4
5	5	6	State	Goa	256	259	308	0.2	0.2	0.2	16.7	16.8	19.9	15.3	15.4	15.5
6	6	7	State	Gujarat	7793	7655	8050	5.8	5.5	5.3	11.6	11.2	11.6	673.2	682.5	691.7
7	7	8	State	Haryana	3547	4191	4001	2.6	3.0	2.6	12.5	14.5	13.7	284.0	288.1	292.1
8	8	9	State	Himachal Pradesh	740	584	857	0.6	0.4	0.6	10.2	8.0	11.6	72.7	73.2	73.6
9	9	10	State	Jammu & Kashmir	330	284	287	0.2	0.2	0.2	2.5	2.1	2.2	134.3	135.3	133.4
10	10	11	State	Jharkhand	1317	1646	2145	1.0	1.2	1.4	3.6	4.4	5.6	370.5	375.8	381.2
11	11	12	State	Karnataka	11561	11288	12259	8.6	8.1	8	17.7	17.1	18.4	654.5	659.7	665
12	12	13	State	Kerala	8237	8556	8500	6.1	6.1	5.6	23.5	24.3	24	350.0	351.9	353.7
13	13	14	State	Madhya Pradesh	11775	12457	14578	8.8	9.0	9.5	14.5	15.1	17.4	814.7	826.1	837.6
14	14	15	State	Maharashtra	17972	18916	19909	13.4	13.6	13	14.8	15.4	16.1	1213.9	1225.3	1236.8
15	15	16	State	Manipur	52	58	44	0.0	0.0	0	1.7	1.9	1.4	30.8	31.1	31.4
16	16	17	State	Meghalaya	189	198	224	0.1	0.1	0.1	16.0	6.1	6.9	11.8	32.3	32.6
17	17	18	State	Mizoram	79	70	108	0.1	0.1	0.1	2.5	5.9	8.9	32.0	12.0	12.1
18	18	19	State	Nagaland	36	41	48	0.0	0.0	0	1.7	1.9	2.2	21.3	21.6	21.8
19	19	20	State	Odisha	4592	4582	5546	3.4	3.3	3.6	10.5	10.5	12.2	435.5	437.3	454.7
20	20	21	State	Punjab	1714	2357	2616	1.3	1.7	1.7	5.8	7.9	8.7	297.0	299.4	301.8
21	21	22	State	Rajasthan	4333	4531	5658	3.2	3.3	3.7	5.7	5.8	7.2	765.9	776.0	786.1
22	22	23	State	Sikkim	199	220	285	0.1	0.2	0.2	30.2	33.1	42.5	6.6	6.7	6.7
23	23	24	State	Tamil Nadu	13896	13493	16883	10.3	9.7	11	18.4	17.8	22.2	754.6	758.1	761.7
24	24	25	State	Telangana	7845	7675	8058	5.8	5.5	5.3	21.2	20.6	21.5	370.3	372.8	375.4
25	25	26	State	Tripura	720	728	845	0.5	0.5	0.6	18.2	18.2	20.9	39.6	40.0	40.4
26	26	27	State	Uttar Pradesh	4849	5464	4804	3.6	3.9	3.1	2.2	2.4	2.1	2230.0	2259.7	2289.3
27	27	28	State	Uttarakhand	421	516	943	0.3	0.4	0.6	3.8	4.6	8.3	110.6	111.8	113.1
28	28	29	State	West Bengal	13255	12665	13103	9.9	9.1	8.6	13.7	13.0	13.4	965.0	971.1	977.2
30	30	30	Union Territory	A & N Islands	164	181	180	0.1	0.1	0.1	41.0	45.5	45	4.0	4.0	4
31	31	31	Union Territory	Chandigarh	160	131	128	0.1	0.1	0.1	13.7	11.1	10.7	11.7	11.8	12
32	32	32	Union	D & N Haveli	101	95	156	0.1	0.1	0.1	19.1	17.1	15	5.3	5.6	10.4

	Unnamed: 0	Sr. No.	Category	State/UT	Number of Suicides (N)18	Number of Suicides (N)19	Number of Suicides (N)20	Percentage Share in Total Suicides18	Percentage Share in Total Suicides19	Percentage Share in Total Suicides20	Rate of Suicides - (N/P)18	Rate of Suicides - (N/P)19	Rate of Suicides - (N/P)20	Projected Mid-Year Population # (in Lakh)(P)18	Projected Mid-Year Population # (in Lakh)(P)19	Projected Mid-Year Population # (in Lakh)(P)20
34	34	34	Union Territory	Dadra & Nagar Haveli	2526	2526	3142	1.4	1.4	1.8	128.9	128.7	156.5	19.6	19.6	20.1
35	35	35	Union Territory	Lakshadweep	3	0	2	0.0	0.0	0	4.3	0.0	2.9	0.7	0.7	0.7
36	36	36	Union Territory	Puducherry	500	493	408	0.4	0.4	0.3	33.8	32.5	26.3	14.8	15.2	15.5

In [118]:

```
# Converting everything to int
df2['Number of Suicides (N)20'] = df2['Number of Suicides (N)20'].astype(int)
```

In [119]:

```
# Checking type
a = df2['Number of Suicides (N)20'][3]
type(a)
```

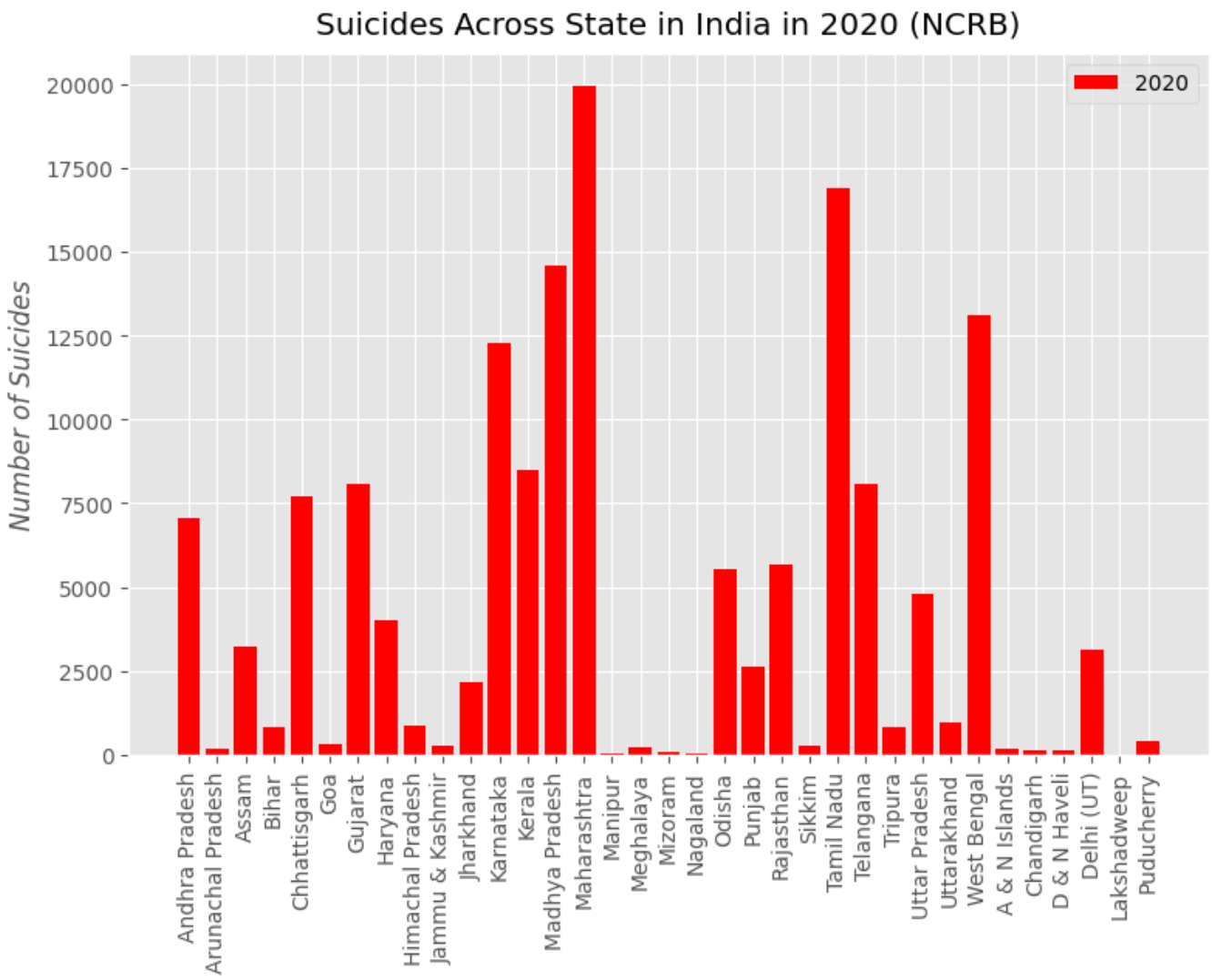
Out[119]:

```
numpy.int32
```

No. of Suicides across States in India 2020 (NCRB)

In [120]:

```
plt.style.use('ggplot')
plt.figure(figsize=(9,6), dpi=100)
plt.title('Suicides Across State in India in 2020 (NCRB)', pad=10)
plt.bar(df2['State/UT'], df2['Number of Suicides (N)20'], color='red', label='2020')
plt.ylabel('Number of Suicides', style='italic')
plt.xticks(rotation = 90)
plt.legend()
# plt.savefig('2020 ncrb india states.png', bbox_inches = 'tight', facecolor='white', transparent = False, dpi=300)
plt.show()
```



Maharashtra, Madhya Pradesh, Tamil Nadu, West Bengal were some of the worst states in terms of Suicide rate in 2020

No. of Suicides committed in Maharashtra that proved fatal was as high as 20,000 in the Year 2020.

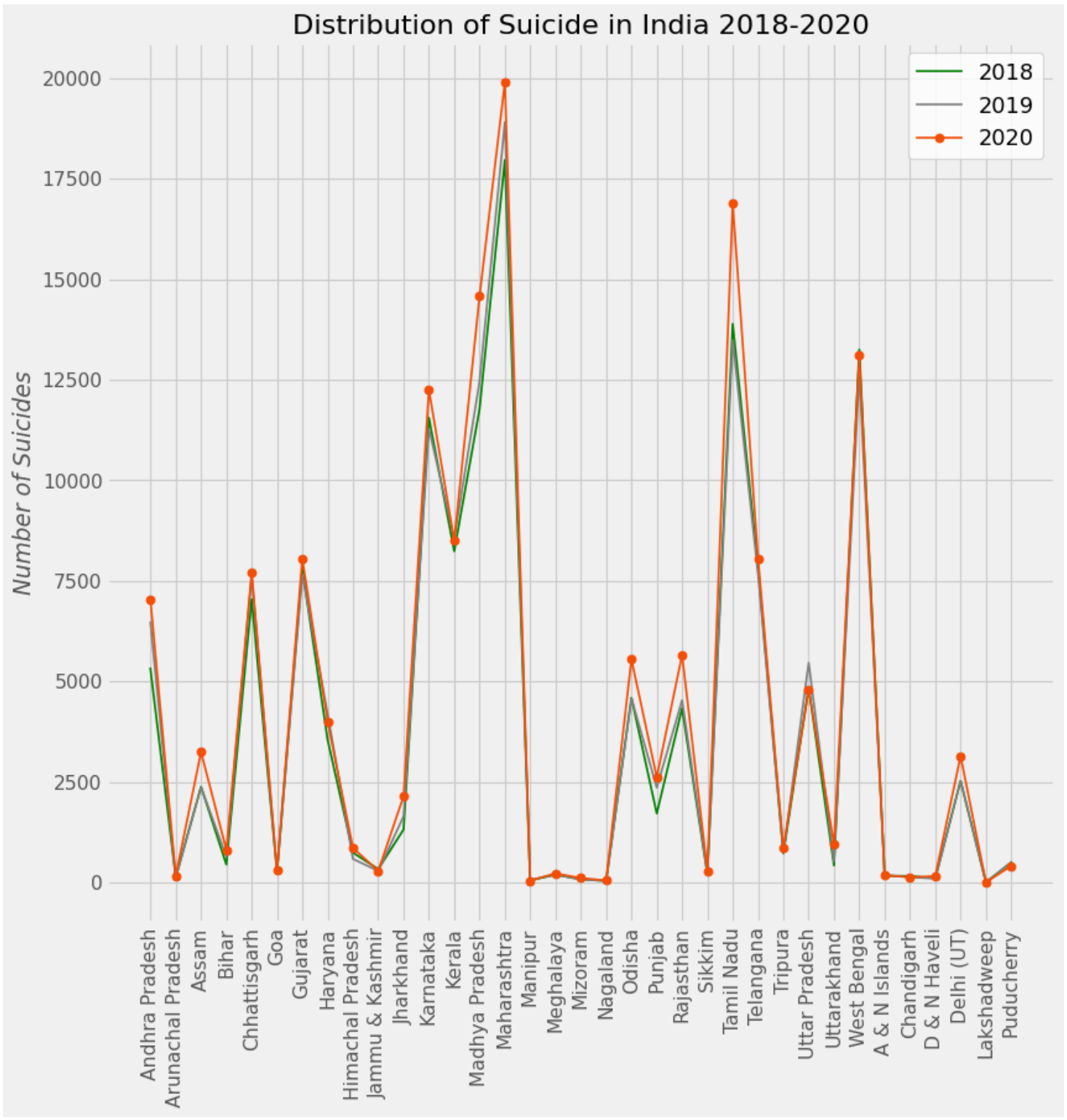
That accounts to at least 54 deaths due to Suicide per day

Plotting the distribution for year 2018, 2019, 2020

In [121]:

```
plt.style.use('fivethirtyeight')
plt.figure(figsize=(11,11), dpi=80)


plt.title('Distribution of Suicide in India 2018-2020')
plt.style.use('default')
plt.plot(df2['State/UT'], df2['Number of Suicides (N)18'], color='green', label='2018')
plt.plot(df2['State/UT'], df2['Number of Suicides (N)19'], color='grey', label='2019')
plt.plot(df2['State/UT'], df2['Number of Suicides (N)20'], color='#f24e07', label='2020', marker='o')


plt.legend(fontsize=16)
##
plt.xticks(rotation = 90)
##
n = np.arange(0,22000,2500)
plt.ylabel('Number of Suicides', style='italic')
```

```
plt.yticks(n)
# plt.savefig('ncrb_india_states_2018-20.png', bbox_inches = 'tight', facecolor='white', transparent = False, dpi=300)

plt.show()
```



Maharashtra, Madhya Pradesh, Tamil Nadu and West Bengal regularly see the greatest number of Suicide Deaths in India and trend says the coming years can see much more.

2c. Specific Analysis of Covid-hit India (2020) (City wise, Cause wise, Age Group wise)

Impact on different age groups (2018-2020)

We manually made a data sheet for 2018 age wise distribution by analyzing the ncrb report because no csv was available

Data Source: https://ncrb.gov.in/sites/default/files/ADSI-2018-FULL-REPORT-2018.pdf

In [141]:

```
# Importing handfilled data of 2018 (NCRB)
d18 = pd.read_csv('./2018 ncrb/2018 age.csv')
```

In [142]:

```
d18
```

Out[142]:

	Age Group	Number of Suicides
0	**Below 18 years - Total**	**9431**
1	**18 and Above-Below 30 years - Total**	**46903**
2	**30 and Above-Below 45 years - Total**	**42493**
3	**45 and Above-Below 60 years - Total**	**94982**
4	**60 years & Above - Total**	**10696**
5	**Total - Total**	**204505**

In [143]:

```
# Importing data of 2019 and 2020

d20 = pd.read_csv('./2020 ncrb/2020 cause age.csv')
d19 = pd.read_csv('./2019 ncrb/2019 cause age.csv')
```

In [144]:

```
d19.head(3)
```

Out[144]:

	S.No	Cause	Below 18 years = Male	Below 18 years = Female	Below 18 years = Transgender	Below 18 years = Total	18 and Above-Below 30 years = Male	18 and Above-Below 30 years = Female	18 and Above-Below 30 years = Transgender	18 and Above-Below 30 years = Total	...	45 and Above-Below 60 years = Transgender	45 and Above-Below 60 years = Total	60 years & Above = Male	60 years & Above = Female	60 years & Above = Transgender	60 years & Above = Total	Total = Male	Total = Female	Total = Transgender	Total = Total
0	1	Bankruptcy or Indebtedness	12	6	0	18	909	105	0	1014	...	0	1910	439	49	0	488	5381	527	0	5908
1	2	Marriage Related Issues (Total)	50	147	0	197	1452	2784	0	4236	...	0	480	50	23	0	73	3382	4213	0	7595
2	2.1	Non Settlement of Marriage	24	55	0	79	577	633	0	1210	...	0	91	3	1	0	4	1294	1037	0	2331

3 rows × 26 columns

In [145]:

```
# Since, we are only interested in total distribution
# This is the data we are interested in
d19.iloc[30::, 1::4]
```

Out[145]:

	Cause	Below 18 years - Total	18 and Above-Below 30 years - Total	30 and Above-Below 45 years - Total	45 and Above-Below 60 years - Total	60 years & Above - Total	Total - Total
30	Total	9613	48774	44287	25436	11013	139123

In [146]:

```
d19 = d19.iloc[30::, 1::4]
```

In [147]:

```
# Transposing to make it regular with the data of 2018

d19 = d19.T
```

In [148]:

```
d19
```

Out[148]:

	30
Cause	Total
Below 18 years - Total	9613
18 and Above-Below 30 years - Total	48774
30 and Above-Below 45 years - Total	44287
45 and Above-Below 60 years - Total	25436
60 years & Above - Total	11013
Total - Total	139123

In [149]:

```
# Dropping Cause row

d19.drop(d19.index[[0]], inplace=True)
```

In [150]:

```
d19.reset_index(inplace=True)
```

In [151]:

```
# Dropping Column Header

df_dict = dict.fromkeys(d19.columns, '')
d19.rename(columns = df_dict, inplace=True)
```

In [152]:

```
d19
```

Out[152]:

0	Below 18 years - Total	9613
1	18 and Above-Below 30 years - Total	48774
2	30 and Above-Below 45 years - Total	44287
3	45 and Above-Below 60 years - Total	25436
4	60 years & Above - Total	11013
5	Total - Total	139123

In [153]:

```
# Also it's good to add relevant column names to help us with merge

headers = ['Age Group', 'Number of Suicides']
d19.columns = headers
```

In [154]:

```
d19
```

Out[154]:

	Age Group	Number of Suicides
0	Below 18 years - Total	9613
1	18 and Above-Below 30 years - Total	48774

	Age Group	Number of Suicides
2	30 and Above-Below 45 years - Total	44287
3	45 and Above-Below 60 years - Total	25436
4	60 years & Above - Total	11013
5	Total - Total	139123

In [155]:

```
# Similar operation on 2020

d20 = d20.iloc[30::, 1::4]
```

In [157]:

```
d20 = d20.T
```

In [158]:

```
d20
```

Out[158]:

	30
Cause	Total
Below 18 years - Total	11396
18 and Above-Below 30 years - Total	52718
30 and Above-Below 45 years - Total	47998
45 and Above-Below 60 years - Total	27814
60 years & Above - Total	13126
Total - Total	153052

In [159]:

```
# Dropping Cause row

d20.drop(d20.index[[0]], inplace=True)
```

In [160]:

```
d20.reset_index(inplace=True)
```

In [161]:

```
# Dropping Column Header

df_dict = dict.fromkeys(d20.columns, '')
d20.rename(columns = df_dict, inplace=True)
```

In [162]:

```
d20.columns = headers
```

In [163]:

```
d20
```

Out[163]:

	Age Group	Number of Suicides
0	Below 18 years - Total	11396
1	18 and Above-Below 30 years - Total	52718
2	30 and Above-Below 45 years - Total	47998
3	45 and Above-Below 60 years - Total	27814
4	60 years & Above - Total	13126
5	Total - Total	153052

Merging dataframe made for 2018,19,20 to get a dataframe of age wise distribution of suicides in India

 # The column to merge is 'Age Group'

In [164]:

```
agewise=pd.merge(d18, d19, how='left', left_on = 'Age Group', right_on='Age Group')
```

In [165]:

```
agewise = pd.merge(agewise, d20, how='left', left_on = 'Age Group', right_on='Age Group')
```

In [166]:

```
agewise
```

Out[166]:

	Age Group	Number of Suicides_x	Number of Suicides_y	Number of Suicides
0	Below 18 years - Total	9431	9613	11396
1	18 and Above-Below 30 years - Total	46903	48774	52718
2	30 and Above-Below 45 years - Total	42493	44287	47998
3	45 and Above-Below 60 years - Total	94982	25436	27814
4	60 years & Above - Total	10696	11013	13126
5	Total - Total	204505	139123	153052

In [167]:

```
## We can further rename the Headers as

header = ['Age Group', '2018', '2019', '2020']
agewise.columns = header
```

In [168]:

```
agewise
```

Out[168]:

	Age Group	2018	2019	2020
0	Below 18 years - Total	9431	9613	11396
1	18 and Above-Below 30 years - Total	46903	48774	52718
2	30 and Above-Below 45 years - Total	42493	44287	47998
3	45 and Above-Below 60 years - Total	94982	25436	27814
4	60 years & Above - Total	10696	11013	13126
5	Total - Total	204505	139123	153052

In [169]:

```
a = agewise.copy()
```

Saving the dataframe for future reference

In [318]:

```
# a.to_csv('./2016-20 ncrb/agewise_18_to_20.csv', index=False)
```

In [28]:

```
# a= pd.read_csv('./2016-20 ncrb/agewise_18_to_20.csv')
```

Plotting NCRB Agewise Dataframe 2018-2020

In [29]:

```
# Ignoring the Total row
a['2018'][:-1]
```

Out[29]:

```
0     9431
1    46903
2    42493
3    94982
4    10696
Name: 2018, dtype: int64
```

In [30]:

```
a['Age Group'][:-1]
```

Out[30]:

```
0                  Below 18 years - Total
1    18 and Above-Below 30 years - Total
2    30 and Above-Below 45 years - Total
3    45 and Above-Below 60 years - Total
4                60 years & Above - Total
Name: Age Group, dtype: object
```

In [31]:

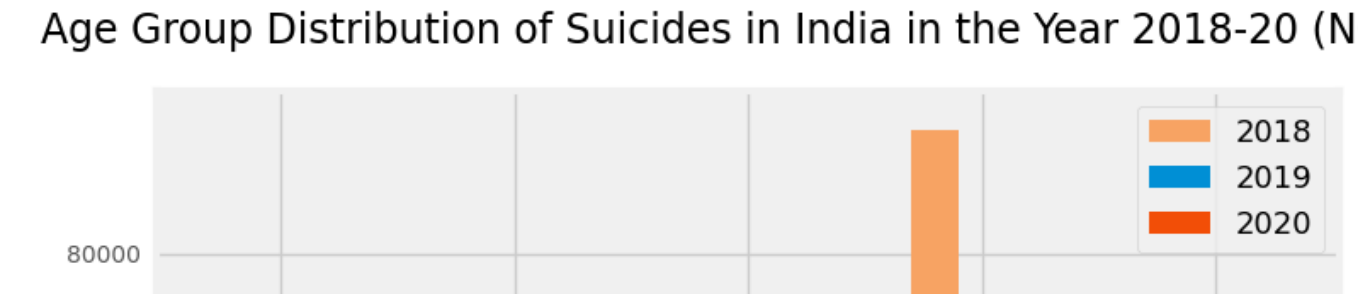
```
a['Age Group'][:-1].tolist()
```

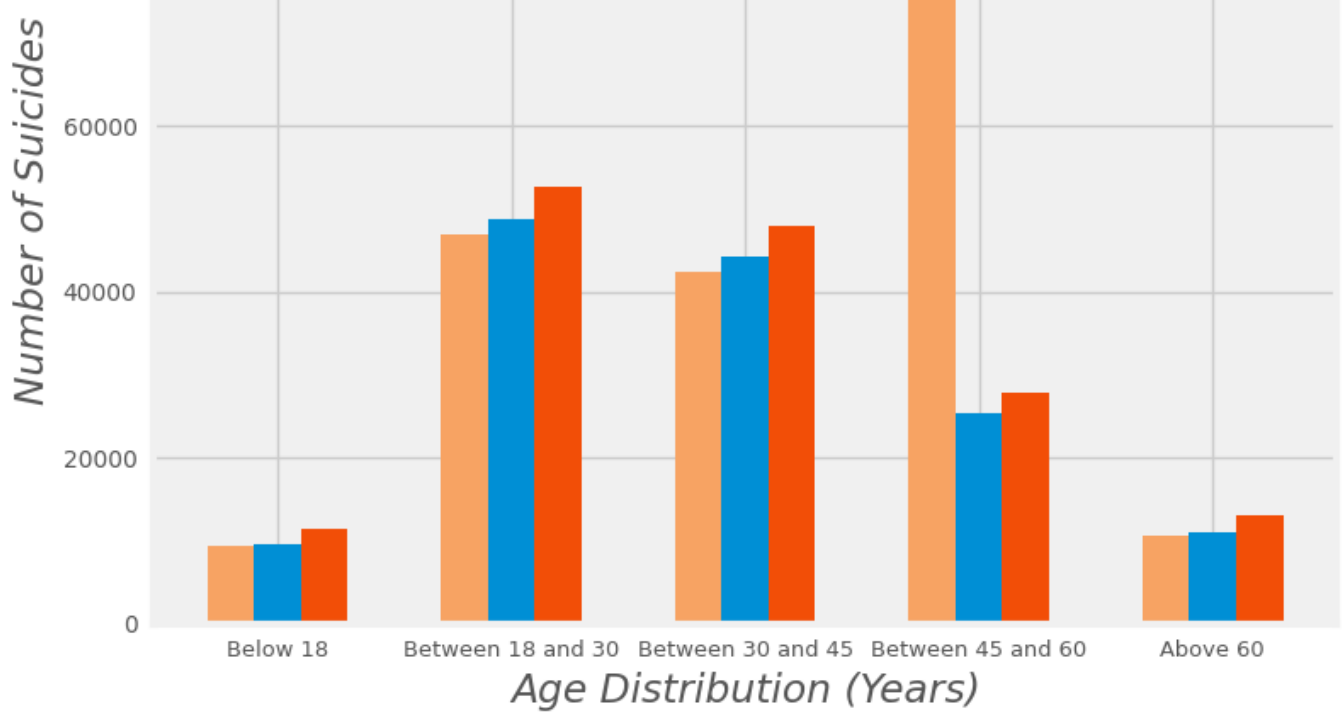
Out[31]:

```
['Below 18 years - Total',
 '18 and Above-Below 30 years - Total',
 '30 and Above-Below 45 years - Total',
 '45 and Above-Below 60 years - Total',
 '60 years & Above - Total']
```

In [32]:

```
plt.style.use('fivethirtyeight')
plt.figure(facecolor='w', figsize=(8,6), dpi=100)
plt.title('Age Group Distribution of Suicides in India in the Year 2018-20 (NCRB)', pad = 20, fontsize=17)
### This is used to divide the x axis in parts
x_axis = np.arange(len(a['Age Group'][:-1]))
###
plt.bar(x_axis - 0.2, a['2018'][:-1], width=0.2, label='2018', color='#f7a363')
plt.bar(x_axis, a['2019'][:-1], width=0.2, label='2019')
plt.bar(x_axis + 0.2, a['2020'][:-1], width=0.2, label='2020', color='#f24e07')
### New way to define xticks
xticks = ['Below 18', 'Between 18 and 30', 'Between 30 and 45', 'Between 45 and 60', 'Above 60']
plt.xticks(x_axis, xticks, fontsize=9.4)
###
plt.yticks(fontsize=10)
plt.xlabel('Age Distribution (Years)', style='italic')
plt.ylabel('Number of Suicides', style='italic')

# Adding arrow
# plt.annotate('', xy=(0.3,15000), xytext=(-0.3,12000), arrowprops=dict(facecolor='red', shrink=0.2))
# plt.savefig('agegroup_india_18-20.png', bbox_inches = 'tight', facecolor='white', transparent = False, dpi=300)

# prop is used to increase the size of legend
plt.legend(prop={'size':13})
plt.show()
```





Two big conclusion can be drawn by above graph

1. The Suicides commiteed by Indian population between age of 45 to 60 was at it's highest in the Year 2018

2. Suicide in the Age Group of 18 to 30 is consistently higher than any other Age Group

3. 11,396 children below 18 years died by Suicide in 2020 i.e., 31 minors death per day

Analyzing the Cause of Death in the Age Group 18-30 Years old in Year 2020

Import 2020 Data of NCRB

In [20]:

```
d20 = pd.read_csv('./2020 ncrb/2020 cause age.csv')
```

In [21]:

```
d20.head(4)
```

Out[21]:

	Si. No	Cause	Below 18 years - Male	Below 18 years - Female	Below 18 years - Transgender	Below 18 years - Total	18 and Above-Below 30 years - Male	18 and Above-Below 30 years - Female	18 and Above-Below 30 years - Transgender	18 and Above-Below 30 years - Total	...	45 and Above-Below 60 years - Transgender	45 and Above-Below 60 years - Total	60 years & Above - Male	60 years & Above - Female	60 years & Above - Transgender	60 years & Above - Total	Total - Male	Total - Female	Total - Transgender	Total - Total
0	1	Bankruptcy or Indebtedness	13	9	0	22	877	135	1	1013	...	0	1589	400	27	0	427	4744	468	1	5213
1	2	Marriage Related Issues (Total)	60	98	0	158	1577	2670	0	4247	...	0	536	42	19	0	61	3484	4152	0	7636
2	2.1	Non Settlement of Marriage	23	44	0	67	716	544	0	1260	...	0	87	13	4	0	17	1372	865	0	2237
3	2.2	Dowry Related Issues	1	3	0	4	140	1337	0	1477	...	0	43	0	0	0	0	269	1749	0	2018

4 rows × 26 columns

We are interested in the Age Group of >18 and <30

In [22]:

```
d20[['Cause', '18 and Above-Below 30 years - Total']]
```

Out[22]:

	Cause	18 and Above-Below 30 years - Total
0	Bankruptcy or Indebtedness	1013
1	Marriage Related Issues (Total)	4247
2	Non Settlement of Marriage	1260
3	Dowry Related Issues	1477
4	Extra Marital Affairs	639
5	Divorce	208
6	Others	663
7	Failure in Examination	860
8	Impotency/Infertility	116
9	Family Problems	17754
10	Illness (Total)	6742
11	AIDS/STD	37
12	Cancer	143
13	Paralysis	78
14	Insanity/ Mental Illness	3955
15	Other Prolonged Illness	2529
16	Death of Dear Person	420

	Cause	18 and Above-Below 30 years - Total
17	Drug Abuse/Alcoholic Addiction	2382
18	Fall in Social Reputation	198
19	Ideological Causes/Hero Worshipping	71
20	Love Affairs	4331
21	Poverty	526
22	Unemployment	1526
23	Property Dispute	373
24	Suspected/ Illicit Relation (Other than Sl. No...	329
25	Illegitimate Pregnancy (Other thanSl. No. 2.3)	11
26	Physical Abuse (Rape, etc.)	23
27	Professional/Career Problem	711
28	Causes Not Known	5975
29	Other Causes	5110
30	Total	52718

In [23]:

```
p = d20[['Cause', '18 and Above-Below 30 years - Total']]
```

In [24]:

```
# We have to select only the overall gist of the above data
p.iloc[[0,1,7,8,9,10,16,17,18,19,20,21,22,23,24,25,26,27,28,29]]
```

Out[24]:

	Cause	18 and Above-Below 30 years - Total
0	Bankruptcy or Indebtedness	1013
1	Marriage Related Issues (Total)	4247
7	Failure in Examination	860
8	Impotency/Infertility	116
9	Family Problems	17754
10	Illness (Total)	6742
16	Death of Dear Person	420
17	Drug Abuse/Alcoholic Addiction	2382
18	Fall in Social Reputation	198
19	Ideological Causes/Hero Worshipping	71
20	Love Affairs	4331
21	Poverty	526
22	Unemployment	1526
23	Property Dispute	373
24	Suspected/ Illicit Relation (Other than Sl. No...	329
25	Illegitimate Pregnancy (Other thanSl. No. 2.3)	11
26	Physical Abuse (Rape, etc.)	23
27	Professional/Career Problem	711
28	Causes Not Known	5975
29	Other Causes	5110

In [25]:

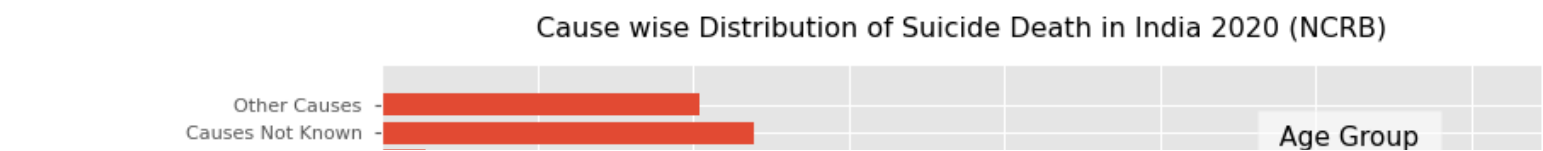
```
data = d = p.iloc[[0,1,7,8,9,10,16,17,18,19,20,21,22,23,24,25,26,27,28,29]]
```

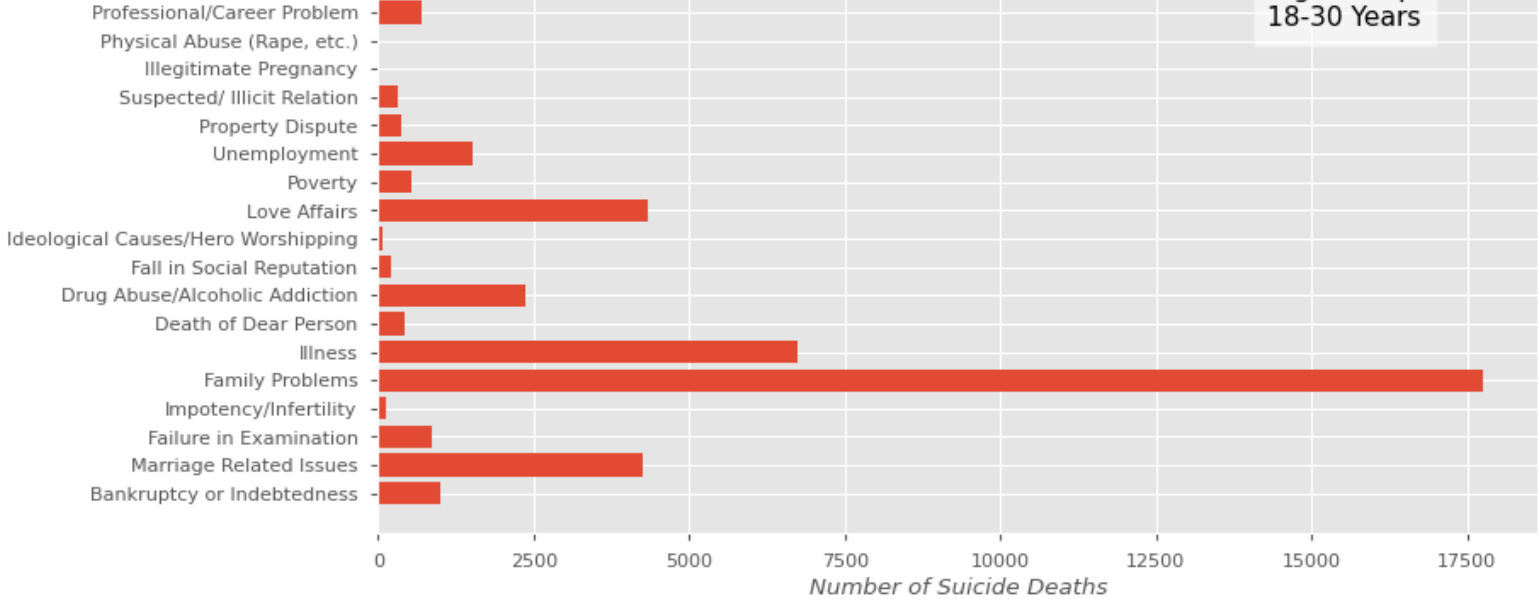
In [573]:

```
# d.Cause.tolist()
```

In [26]:

```
plt.style.use('ggplot')
plt.figure(facecolor='w', figsize=(11,6), dpi=80)
plt.title('Cause wise Distribution of Suicide Death in India 2020 (NCRB)', pad=15)
labels = ['Bankruptcy or Indebtedness',
 'Marriage Related Issues',
 'Failure in Examination',
 'Impotency/Infertility',
 'Family Problems',
 'Illness',
 'Death of Dear Person',
 'Drug Abuse/Alcoholic Addiction',
 'Fall in Social Reputation',
 'Ideological Causes/Hero Worshipping',
 'Love Affairs',
 'Poverty',
 'Unemployment',
 'Property Dispute',
 'Suspected/ Illicit Relation',
 'Illegitimate Pregnancy',
 'Physical Abuse (Rape, etc.)',
 'Professional/Career Problem',
 'Causes Not Known',
 'Other Causes']
plt.barh(labels, d['18 and Above-Below 30 years - Total'])
plt.text(14300, 16.5, ' Age Group\n18-30 Years', fontsize=14, bbox={'facecolor': 'white', 'alpha': 0.6, 'pad': 8})
plt.xlabel('Number of Suicide Deaths', style='italic')
# plt.savefig('causewise_India_2020.png', bbox_inches = 'tight', facecolor='white', transparent = False, dpi=300)

plt.show()
```





Conclusion that can be drawn from the above graph is that 'Family Problems' play a huge role in the cause of Suicides of the this Age Group which is a matter of concern.

Analyzing the City Wise Situation in 2020

In [16]:

```
# Importing Data

city = pd.read_csv('./2020 ncrb/2020 city.csv')
```

In [17]:

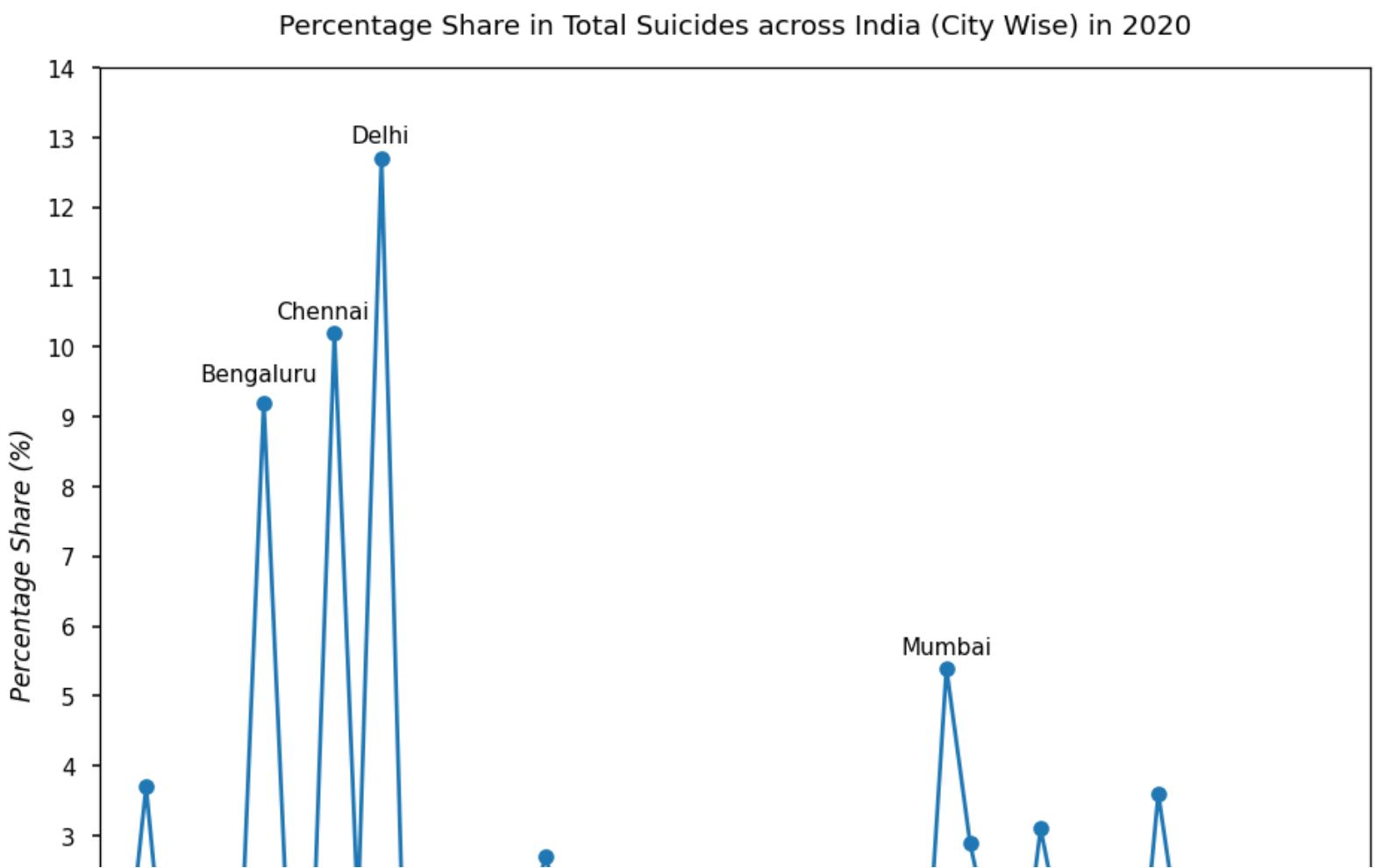
```
city.head(4)
```

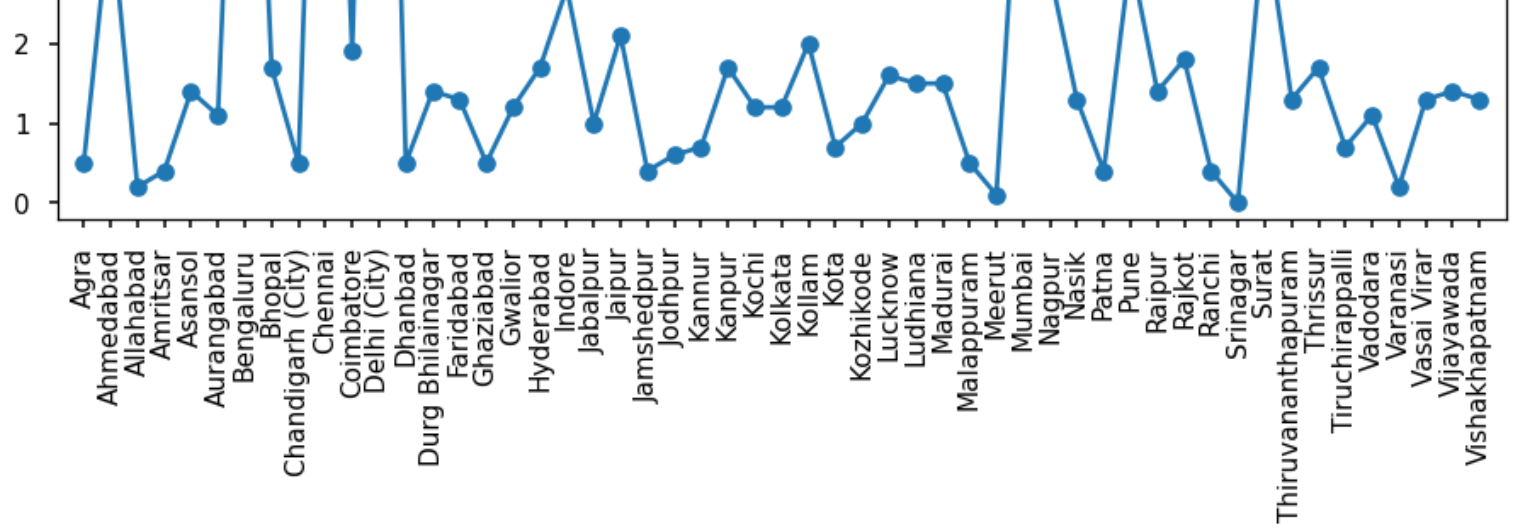
Out[17]:

	Si. No. (Col.1)	Category	City (Col.2)	Number of Suicides(Col.3)	Percentage Share in Total Suicides (Col.4)	Actual Population \n (in Lakh)\n(Col.5)	Rate of Suicides (Col.3/Col.5)
0	**1**	**City**	**Agra**	**115**	**0.5**	**17.5**	**6.6**
1	**2**	**City**	**Ahmedabad**	**871**	**3.7**	**63.5**	**13.7**
2	**3**	**City**	**Allahabad**	**40**	**0.2**	**12.2**	**3.3**
3	**4**	**City**	**Amritsar**	**97**	**0.4**	**11.8**	**8.2**

In [18]:

```
plt.style.use('seaborn-notebook')
plt.figure(facecolor='w', figsize=(10,8), dpi=110)
plt.title('Percentage Share in Total Suicides across India (City Wise) in 2020', pad = 15)
plt.plot(city['City (Col.2)'][:-1], city['Percentage Share in Total Suicides (Col.4)'][:-1],  marker='o')
plt.yticks(np.arange(0,15,1))

# adjusting x and y limits
plt.xlim(left=-1)
plt.xlim(right=53)
plt.ylim(bottom=-0.2)
###

plt.xticks(rotation=90)
plt.ylabel('Percentage Share (%)', style='italic')

### Highest Percentage share
plt.text(3.3, 9.5, 'Bengaluru', fontsize=10)
plt.text(6.5, 10.4, 'Chennai', fontsize=10)
plt.text(9.7, 12.93, 'Delhi', fontsize=10)
plt.text(33.1, 5.6, 'Mumbai', fontsize=10)
# plt.savefig('citywise_india_2020.png', bbox_inches = 'tight', facecolor='white', transparent = False, dpi=300)

plt.show()
```





Worst record was from Delhi, Chennai, Bengaluru and Mumbai in that order (as per the data of NCRB)

Delhi shared 12.7% of total deaths due to suicide in India in 2020 with 3025 deaths in the Year

That is 8 fatal Suicides a day in the Capital

The situation in India is really getiing worse with each passing Year. The Covid gave a huge blow to the already stressed Suicide Rates in major developed cities of India. Keeping in Mind that the data available is as recent as of the Year 2020, we can only imagine what can we see in the reports for the Year 2021, which was even a bad Year in terms of fatalities due to Covid in India.

So a curious question that comes in mind is - What is the source to this? What can be done to make the situation better? What is to be done on the part of Government?

Where from here?

Is there any hope?

Will it get better anytime?

Based on the deep Analysis by some of the esteemed institutes like Lancet I have come of with some really good hope/work in this area

Helplines:

1. MOHFW- GOI has issued a tollfree helpline number for 'Behavioural Health', The Psycho-Social toll-free helpline-08046110007 can be used by anyone needing mental health assistance during the COVID-19 pandemic. A list of videos, advisories and resource materials on coping stress during COVID, yoga and meditation advice, taking care of the mental health of vulnerable groups, etc. have been provided in the MOHFW-GOI web portal (MoHFW | Home, 2020).

Apart from Central Government various state government and other Institutions has also launched Mental Health related support Helplines for anyone who needs help. It is highlighted in P. No. 594 of this article International Journal of Social Psychiatry 2021, Vol. 67(5) 587–600

https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7468668/pdf/10.1177_0020764020950769.pdf

2. SNEHA, a non-governmental organisation that is driven by volunteers and based in the city of Chennai, pioneered the suicide prevention helpline in India as early as 1986. Since then, several suicide prevention and mental health helplines provide support to individuals who need it. Furthermore, these helplines educate gatekeepers (ie, people who are likely to be in contact with people with suicidal behaviour), raise awareness in the public and media, and serve as an entry point for people in need of professional help.

Lists of few of them can be found here:

http://www.healthcollective.in/suicide-prevention-helplines/

Government Specific Initiative:

1. In 2011, the GOI introduced the much needed Mental Health Policy. The vision of the National Mental Health Policy is to promote mental health, prevent mental illness, enable recovery from mental illness, promote destigmatization and desegregation, and ensure socio-economic inclusion of persons affected by mental illness by providing accessible, affordable and quality health and social care to all persons through their life-span within a rights-based frame work.

More of it can be read and understood here:

https://nhm.gov.in/images/pdf/National_Health_Mental_Policy.pdf

Where can we (youths) come in? :

1. Need of more Research Enthusiasts in the field.

A strategic approach to suicide prevention research is needed to understand the mechanisms that protect against suicide and emergence of suicidal ideation in the context of India, gender, and the life course. Intervention studies are among the highest research priority. As many of the successful suicide prevention interventions developed in high-income country settings might not be directly transferable in India,86,87 research is urgently needed to adapt the known interventions and to explore context-specific interventions for India.High-quality timely research is needed to understand the suicide-related consequences of COVID-19 and to understand risk mitigation. These recommendations are relevant to India as well as other countries.Unfortunately, injury research has not received much attention in India.

2. Need to demand more timely, accurate and reliable data.

India has a medically certified cause of death system that is incomplete, covering only 20% of all deaths. Overall improvement in the system for documenting the cause of death would enable more robust data on suicide deaths. Efforts are needed to address the under-reporting and inadequate reporting in the NCRB data on suicide deaths, which has been highlighted previously. To destigmatise suicide and suicide attempts, data should be captured under the national disease surveillance programme within India's health system in addition to the data captured in NCRB, which would facilitate better understanding of burden and risk factors over time. The national disease surveillance programme should collect data on the basis of the recommended injury surveillance guidelines, which allow for systematic capturing of required data on suicides. A comprehensive community surveillance system using third party informants can add value to the existing data from hospitals and the police, and possibly address the under-reporting of the NCRB.

More suggestions and way ahead is beautifully highlighted in the article by 'The Lancet Psychiatry':

https://www.thelancet.com/action/showPdf?pii=S2215-0366%2821%2900152-2

https://www.thelancet.com/action/showPdf?pii=S2215-0366%2821%2900152-8

In the land of Mahatma Gandhi and Vivekananda no issue is bigger than the precious upbringing that this society had. Despite of the fact that this pandemic has brought up a big challeneg to this world and especially to developing countries like ours, I am hopeful that our determination and spirit will keep our vision to a safe, happy and prosperous India undeterred. Sharing some visualization of how to look into the possible pathway depression slips in our brains in this pandemic.

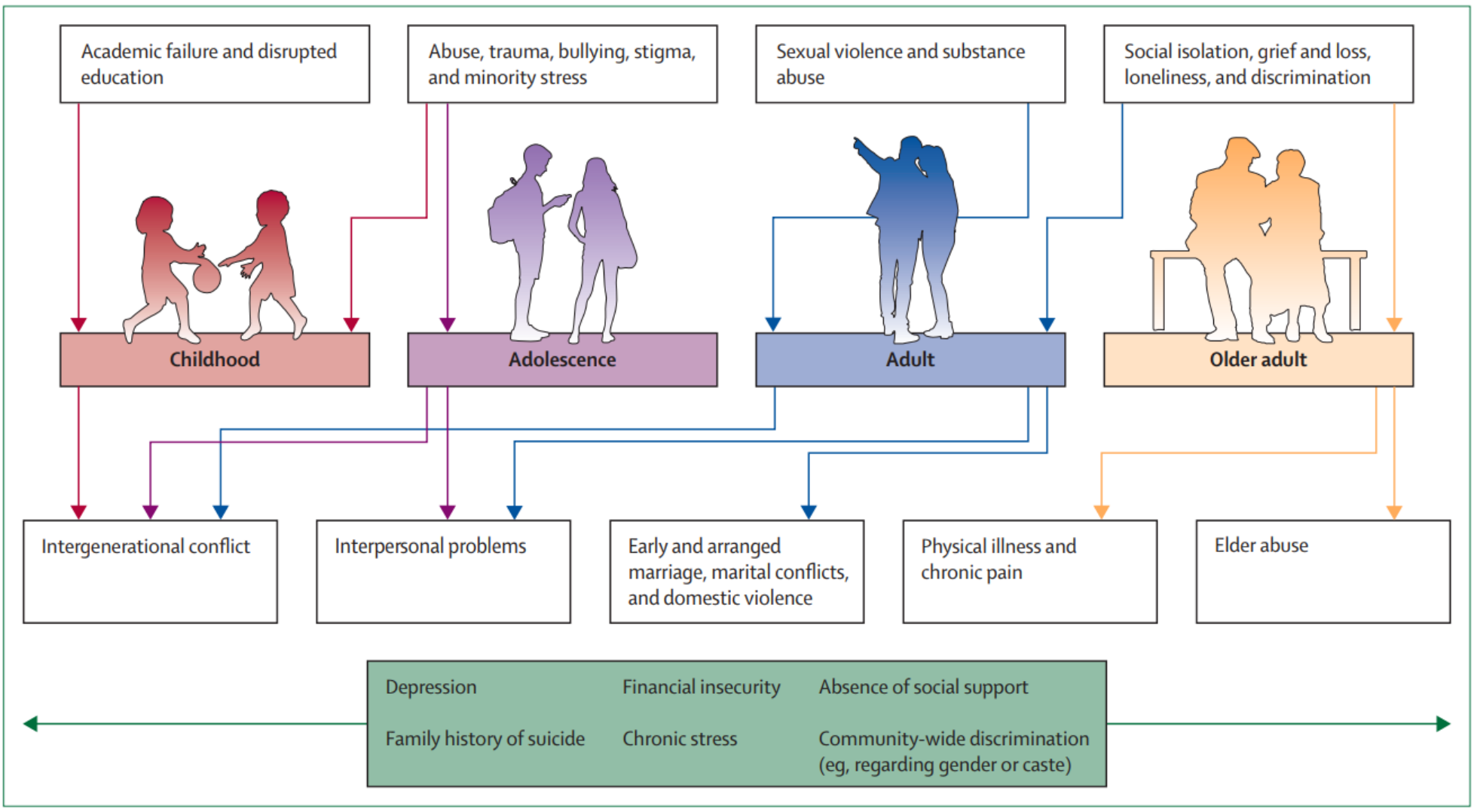


Figure 2: **Evidence-based risk factors for suicide in India across the lifespan**
Various age-group specific risks are mentioned, the risks in the green box are prevalent throughout the lifespan.

Source: ***https://www.thelancet.com/action/showPdf?pii=S2215-0366%2821%2900152-8***

Stay Happy. Thank You.

I would love to connect with you for any suggestions and support:

Mail me at: rupesh19rajan@gmail.com

LinkedIn: ***https://www.linkedin.com/in/rupeshranjan/***

GitHub: ***https://github.com/python-noobtopro***