

Welcome to the Notebook

Importing modules

```
In [1]: import pandas as pd
import numpy as np
import plotly.express as px
import matplotlib.pyplot as plt
print('modules are imported')

modules are imported
```

Loading the Dataset

```
In [2]: datasetURL = 'https://raw.githubusercontent.com/datasets/covid-19/master/data/countries-aggregated.csv'
df = pd.read_csv(datasetURL)
```

Let's check the dataframe

```
In [3]: df.head()
```

```
Out[3]:
```

	Date	Country	Confirmed	Recovered	Deaths
0	2020-01-22	Afghanistan	0	0	0
1	2020-01-23	Afghanistan	0	0	0
2	2020-01-24	Afghanistan	0	0	0
3	2020-01-25	Afghanistan	0	0	0
4	2020-01-26	Afghanistan	0	0	0

```
In [4]: df.tail()
```

```
Out[4]:
```

	Date	Country	Confirmed	Recovered	Deaths
161563	2022-04-12	Zimbabwe	247094	0	5460
161564	2022-04-13	Zimbabwe	247160	0	5460
161565	2022-04-14	Zimbabwe	247208	0	5462
161566	2022-04-15	Zimbabwe	247237	0	5462
161567	2022-04-16	Zimbabwe	247237	0	5462

Let's check the shape of the dataframe

```
In [5]: df.shape
```

```
Out[5]: (161568, 5)
```

Let's do some preprocessing

```
In [6]: df = df[df.Confirmed > 0]
```

```
In [7]: df.head()
```

```
Out[7]:
```

	Date	Country	Confirmed	Recovered	Deaths
33	2020-02-24	Afghanistan	5	0	0
34	2020-02-25	Afghanistan	5	0	0
35	2020-02-26	Afghanistan	5	0	0
36	2020-02-27	Afghanistan	5	0	0
37	2020-02-28	Afghanistan	5	0	0

Let's see data related to a country for example India

```
In [8]: df[df.Country == 'India']
```

Out[8]:

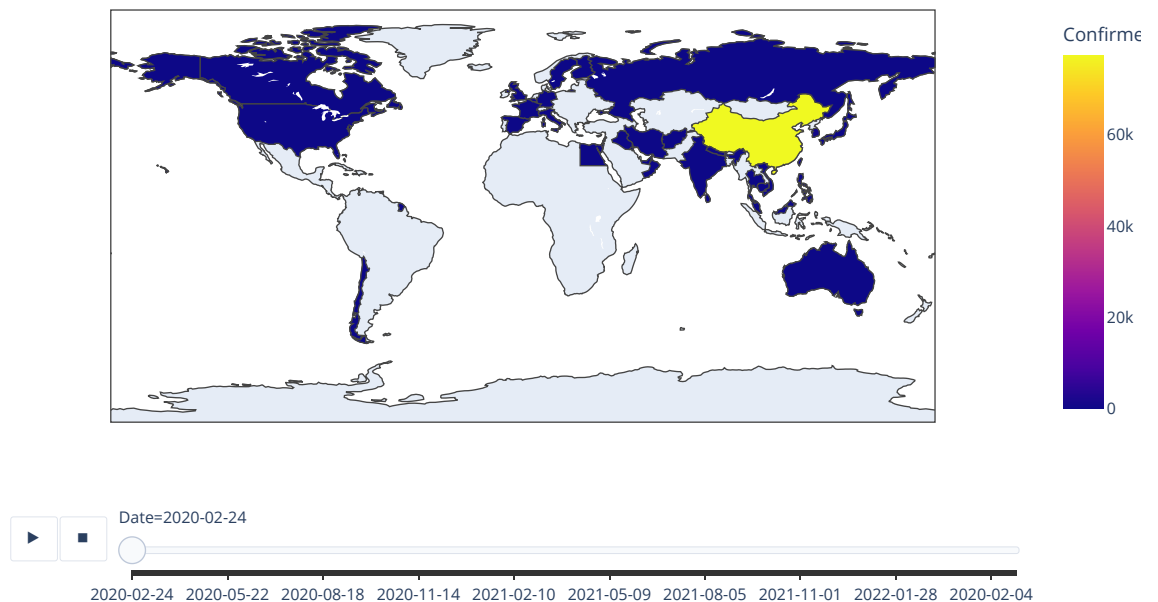
	Date	Country	Confirmed	Recovered	Deaths
65288	2020-01-30	India	1	0	0
65289	2020-01-31	India	1	0	0
65290	2020-02-01	India	1	0	0
65291	2020-02-02	India	2	0	0
65292	2020-02-03	India	3	0	0
...
66091	2022-04-12	India	43038016	0	521736
66092	2022-04-13	India	43039023	0	521737
66093	2022-04-14	India	43039972	0	521743
66094	2022-04-15	India	43040947	0	521747
66095	2022-04-16	India	43042097	0	521751

808 rows × 5 columns

Let's see Global spread of Covid19

```
In [9]: fig = px.choropleth(df, locations = 'Country', locationmode = 'country names', color = 'Confirmed', animation_frame='Date',
fig.update_layout(title_text = "Global Spread of COVID-19")
fig.show()
```

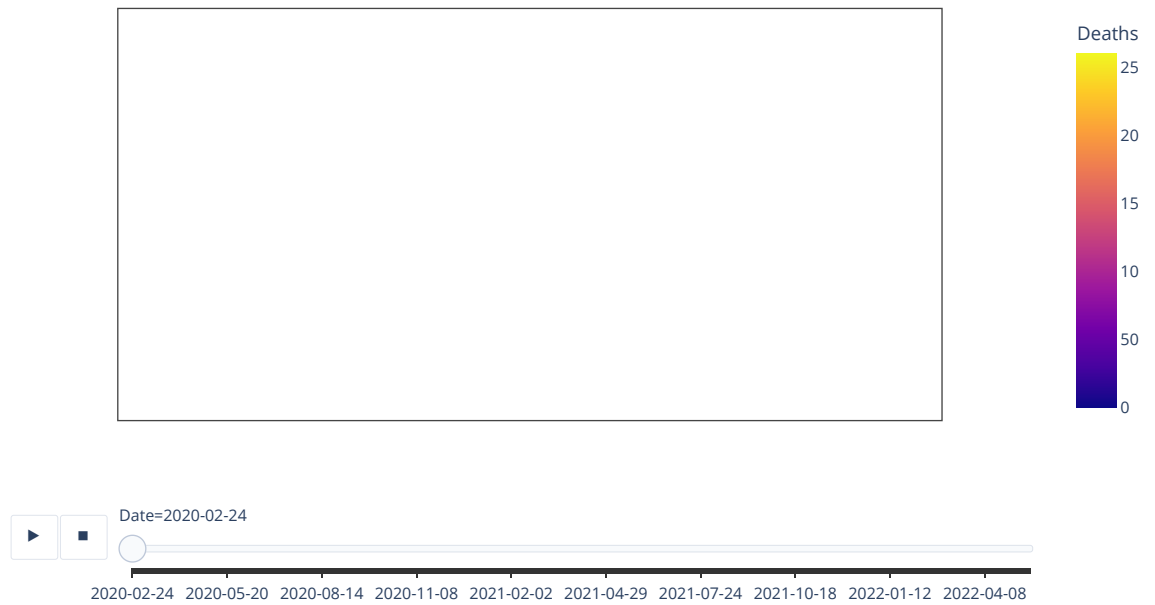
Global Spread of COVID-19



Let's see Global deaths of Covid19

```
In [10]: fig = px.choropleth(df, locations = 'Country', locationmode = 'country names', color = 'Deaths', animation_frame='Date')
fig.update_layout(title_text = "Global Spread of COVID-19")
fig.show()
```

Global Spread of COVID-19



Let's Visualize how intensive the Covid19 Transmission has been in each of the country

let's start with an example:

```
In [11]: df_china = df[df.Country == 'China']
df_china.head()
```

Out[11]:

	Date	Country	Confirmed	Recovered	Deaths
30192	2020-01-22	China	548	28	17
30193	2020-01-23	China	643	30	18
30194	2020-01-24	China	920	36	26
30195	2020-01-25	China	1406	39	42
30196	2020-01-26	China	2075	49	56

let's select the columns that we need

```
In [12]: df_china = df_china[['Date', 'Confirmed']]
```

```
In [13]: df_china.head()
```

Out[13]:

	Date	Confirmed
30192	2020-01-22	548
30193	2020-01-23	643
30194	2020-01-24	920
30195	2020-01-25	1406
30196	2020-01-26	2075

calculating the first derivation of confrimed column

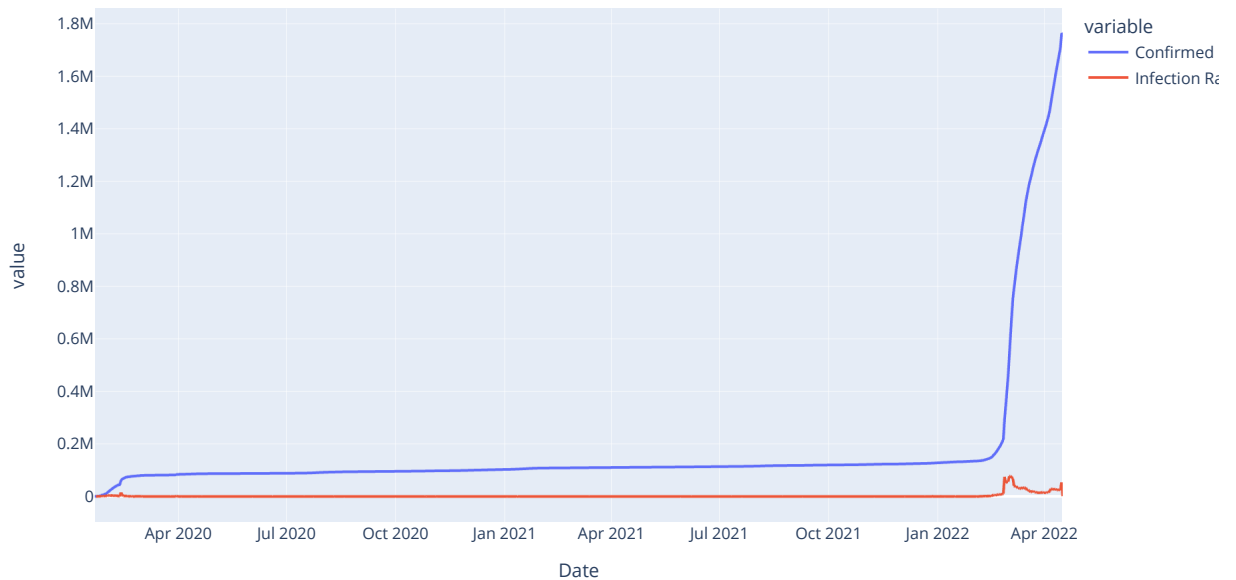
```
In [14]: df_china['Infection Rate'] = df_china['Confirmed'].diff()
```

```
In [15]: df_china.head()
```

```
Out[15]:
```

	Date	Confirmed	Infection Rate
30192	2020-01-22	548	NaN
30193	2020-01-23	643	95.0
30194	2020-01-24	920	277.0
30195	2020-01-25	1406	486.0
30196	2020-01-26	2075	669.0

```
In [16]: px.line(df_china, x = 'Date', y = ['Confirmed', 'Infection Rate'])
```



```
In [17]: df_china['Infection Rate'].max()
```

```
Out[17]: 77402.0
```

Let's Calculate Maximum infection rate for all of the countries

```
In [18]: df.head()
```

```
Out[18]:
```

	Date	Country	Confirmed	Recovered	Deaths
33	2020-02-24	Afghanistan	5	0	0
34	2020-02-25	Afghanistan	5	0	0
35	2020-02-26	Afghanistan	5	0	0
36	2020-02-27	Afghanistan	5	0	0
37	2020-02-28	Afghanistan	5	0	0

```
In [19]: countries = list(df['Country'].unique())
maxInfectionRates = []
for country in countries:
    maxInfection = df[df.Country == country].Confirmed.diff().max()
    maxInfectionRates.append(maxInfection)
```

Let's create a new Dataframe

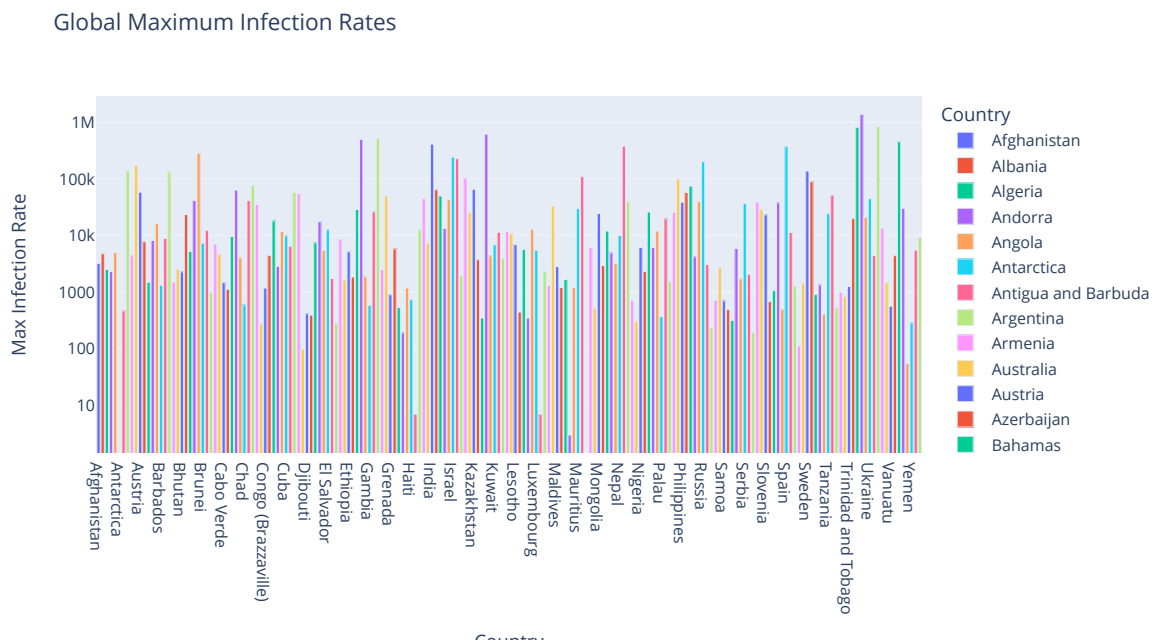
```
In [20]: df_maxInfection = pd.DataFrame()
df_maxInfection['Country'] = countries
df_maxInfection['Max Infection Rate'] = maxInfectionRates
df_maxInfection.head()
```

Out[20]:

	Country	Max Infection Rate
0	Afghanistan	3243.0
1	Albania	4789.0
2	Algeria	2521.0
3	Andorra	2313.0
4	Angola	5035.0

Let's plot the barchart : maximum infection rate of each country

```
In [21]: xInfection, x = 'Country', y = 'Max Infection Rate', color = 'Country', title = 'Global Maximum Infection Rates', log_y=T
```



Let's See how National Lockdowns Impacts Covid19 transmission in India

COVID19 pandemic lockdown in India

On 9 March 2020, the government of India under Prime Minister Narendra Modi imposed a national quarantine, restricting the movement of the population except for necessity, work, and health circumstances, in response to the growing pandemic of COVID-19 in the country. [source](https://en.wikipedia.org/wiki/COVID-19_lockdown_in_India) (https://en.wikipedia.org/wiki/COVID-19_lockdown_in_India).

```
In [22]: india_lockdown_start_date = '2020-08-09'
india_lockdown_a_month_later = '2020-09-09'
```

```
In [23]: df.head()
```

Out[23]:

	Date	Country	Confirmed	Recovered	Deaths
33	2020-02-24	Afghanistan	5	0	0
34	2020-02-25	Afghanistan	5	0	0
35	2020-02-26	Afghanistan	5	0	0
36	2020-02-27	Afghanistan	5	0	0
37	2020-02-28	Afghanistan	5	0	0

let's get data related to India

```
In [24]: df_india = df[df.Country == 'India']
start_date = '2020-01-02'
end_date = '2021-01-03'

df_india = df_india[(df['Date'] >= start_date) & (df['Date'] <= end_date)]

C:\Users\Xploit\AppData\Local\Temp\ipykernel_10048\3307128508.py:5: UserWarning:
Boolean Series key will be reindexed to match DataFrame index.
```

lets check the dataframe

```
In [25]: df_india.head()
```

Out[25]:

	Date	Country	Confirmed	Recovered	Deaths
65288	2020-01-30	India	1	0	0
65289	2020-01-31	India	1	0	0
65290	2020-02-01	India	1	0	0
65291	2020-02-02	India	2	0	0
65292	2020-02-03	India	3	0	0

let's calculate the infection rate in India

```
In [26]: df_india['Infection Rate'] = df_india['Confirmed'].diff()
df_india.head()
```

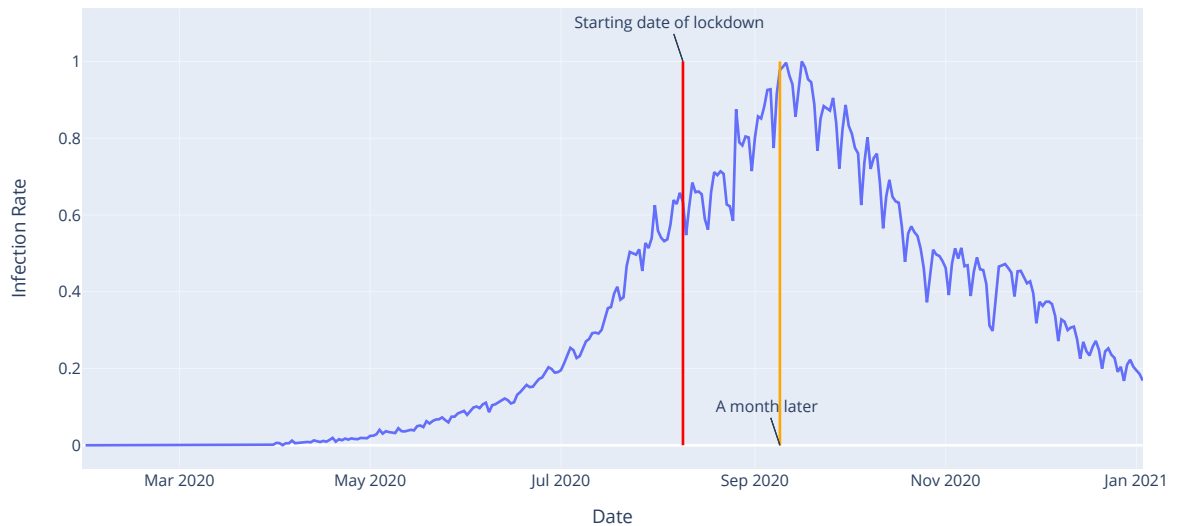
Out[26]:

	Date	Country	Confirmed	Recovered	Deaths	Infection Rate
65288	2020-01-30	India	1	0	0	NaN
65289	2020-01-31	India	1	0	0	0.0
65290	2020-02-01	India	1	0	0	0.0
65291	2020-02-02	India	2	0	0	1.0
65292	2020-02-03	India	3	0	0	1.0

ok! now let's do the visualization

```
In [34]: fig = px.line(df_india, x = 'Date', y = 'Infection Rate', title = 'Before and After Lockdown')
fig.add_shape(
    dict(
        type = "line",
        x0 = india_lockdown_start_date,
        y0 = 0,
        x1 = india_lockdown_start_date,
        y1 = df_india['Infection Rate'].max(),
        line = dict(color = 'red', width = 2)
    )
)
fig.add_shape(
    dict(
        type = "line",
        x0 = india_lockdown_a_month_later,
        y0 = 0,
        x1 = india_lockdown_a_month_later,
        y1 = df_india['Infection Rate'].max(),
        line = dict(color = 'orange', width = 2)
    )
)
fig.add_annotation(
    dict(
        x = india_lockdown_start_date,
        y = df_india['Infection Rate'].max(),
        text = 'Starting date of lockdown'
    )
)
fig.add_annotation(
    dict(
        x = india_lockdown_a_month_later,
        y = 0,
        text = 'A month later'
    )
)
fig.show()
```

Before and After Lockdown



Let's See how National Lockdowns Impacts Covid19 active cases in India

```
In [28]: df_india.head()
```

Out[28]:

	Date	Country	Confirmed	Recovered	Deaths	Infection Rate
65288	2020-01-30	India	1	0	0	NaN
65289	2020-01-31	India	1	0	0	0.0
65290	2020-02-01	India	1	0	0	0.0
65291	2020-02-02	India	2	0	0	1.0
65292	2020-02-03	India	3	0	0	1.0

let's calculate number of active cases day by day

```
In [29]: df_india['Death Rate'] = df_india.Deaths.diff()
```

let's check the dataframe again

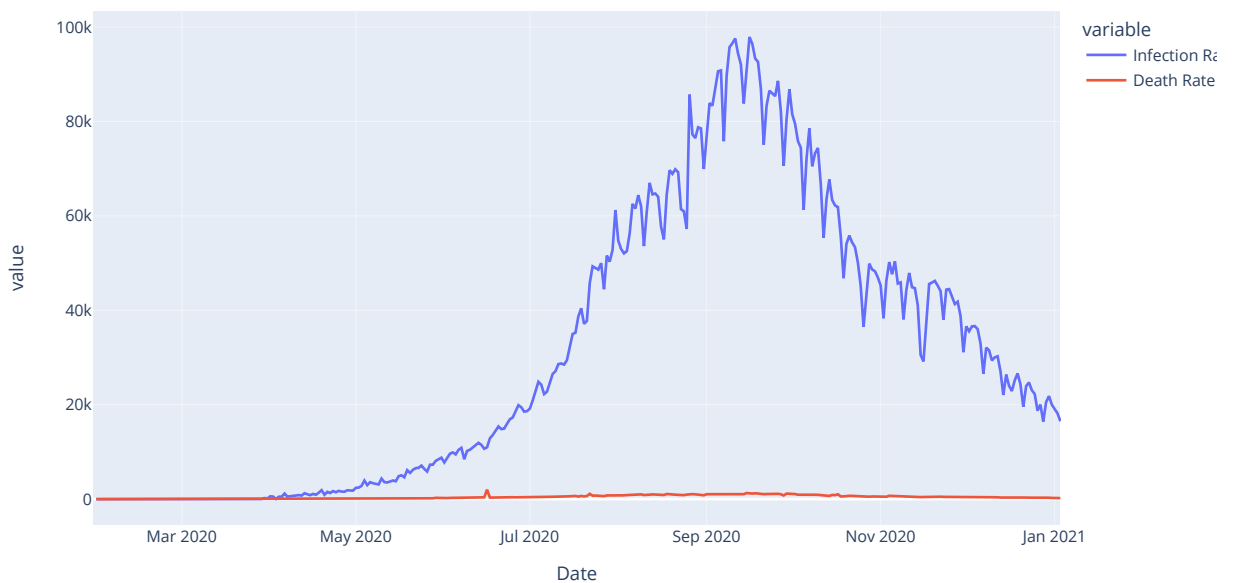
```
In [30]: df_india.head()
```

Out[30]:

	Date	Country	Confirmed	Recovered	Deaths	Infection Rate	Death Rate
65288	2020-01-30	India	1	0	0	NaN	NaN
65289	2020-01-31	India	1	0	0	0.0	0.0
65290	2020-02-01	India	1	0	0	0.0	0.0
65291	2020-02-02	India	2	0	0	1.0	0.0
65292	2020-02-03	India	3	0	0	1.0	0.0

now let's plot a line chart to compare COVID19 national lockdowns impacts on spread of the virus and number of active cases

```
In [31]: fig = px.line(df_india, x = 'Date', y = ['Infection Rate', 'Death Rate'])  
fig.show()
```



let's normalize the columns

```
In [32]: df_india['Infection Rate'] = df_india['Infection Rate']/df_india['Infection Rate'].max()  
df_india['Death Rate'] = df_india['Death Rate']/df_india['Death Rate'].max()
```

let's plot the line chart again


```

In [33]: fig = px.line(df_india, x = 'Date', y = ['Infection Rate', 'Death Rate'])
fig.add_shape(
    dict(
        type = "line",
        x0 = india_lockdown_start_date,
        y0 = 0,
        x1 = india_lockdown_start_date,
        y1 = df_india['Infection Rate'].max(),
        line = dict(color = 'red', width = 2)
    )
)
fig.add_shape(
    dict(
        type = "line",
        x0 = india_lockdown_a_month_later,
        y0 = 0,
        x1 = india_lockdown_a_month_later,
        y1 = df_india['Infection Rate'].max(),
        line = dict(color = 'orange', width = 2)
    )
)
fig.add_annotation(
    dict(
        x = india_lockdown_start_date,
        y = df_india['Infection Rate'].max(),
        text = 'Starting date of lockdown'
    )
)
fig.add_annotation(
    dict(
        x = india_lockdown_a_month_later,
        y = 0,
        text = 'A month later'
    )
)
fig.show()

```

