

Nama: Sharon Zefanya Setiawan

NIM: 2501961022

## Exploratory data analysis and visualization using Python

### Memuat Libraries

In [1]:

```
# Import packages
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
# Set style & figures inline
sns.set()
%matplotlib inline
```

### Memuat Data

In [2]:

```
confirmed_cases_data = 'time_series_covid19_confirmed_global.csv'
death_cases_data = 'time_series_covid19_deaths_global.csv'
recovered_cases_data = 'time_series_covid19_recovered_global.csv'

# Import datasets as pandas dataframes
raw_data_confirmed = pd.read_csv(confirmed_cases_data)
raw_data_deaths = pd.read_csv(death_cases_data)
raw_data_recovered = pd.read_csv(recovered_cases_data)
```

### Kasus terkonfirmasi COVID-19

Langkah selanjutnya dalam proses explorasi dan visualisasi data covid-19 adalah menampilkan beberapa baris teratas, informasi, dan deskriptif statistik dataframe `raw_data_confirmed`

In [3]:

```
raw_data_confirmed.head()
```

Out[3]:

	Province/State	Country/Region	Lat	Long	1/22/20	1/23/20	1/24/20	1/25/20	1/26/20	1/27/20	...	3/29/20	3/30/20	3/31/20	4/1/20
0	NaN	Afghanistan	33.0000	65.0000	0	0	0	0	0	0	...	120	170	174	
1	NaN	Albania	41.1533	20.1683	0	0	0	0	0	0	0	212	223	243	
2	NaN	Algeria	28.0339	1.6596	0	0	0	0	0	0	0	511	584	716	
3	NaN	Andorra	42.5063	1.5218	0	0	0	0	0	0	0	334	370	376	
4	NaN	Angola	-11.2027	17.8739	0	0	0	0	0	0	0	7	7	7	

5 rows × 81 columns

In [4]:

```
raw_data_confirmed.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 263 entries, 0 to 262
Data columns (total 81 columns):
 #   Column           Non-Null Count  Dtype  
 --- 
 0   Province/State  82 non-null     object 
 1   Country/Region  263 non-null    object 
 2   Lat              263 non-null    float64
 3   Long             263 non-null    float64
 4   1/22/20          263 non-null    int64  
 5   1/23/20          263 non-null    int64  
 6   1/24/20          263 non-null    int64  
 7   1/25/20          263 non-null    int64  
 ...
```

```
8 1/26/20      263 non-null    int64
9 1/27/20      263 non-null    int64
10 1/28/20     263 non-null    int64
11 1/29/20     263 non-null    int64
12 1/30/20     263 non-null    int64
13 1/31/20     263 non-null    int64
14 2/1/20      263 non-null    int64
15 2/2/20      263 non-null    int64
16 2/3/20      263 non-null    int64
17 2/4/20      263 non-null    int64
18 2/5/20      263 non-null    int64
19 2/6/20      263 non-null    int64
20 2/7/20      263 non-null    int64
21 2/8/20      263 non-null    int64
22 2/9/20      263 non-null    int64
23 2/10/20     263 non-null    int64
24 2/11/20     263 non-null    int64
25 2/12/20     263 non-null    int64
26 2/13/20     263 non-null    int64
27 2/14/20     263 non-null    int64
28 2/15/20     263 non-null    int64
29 2/16/20     263 non-null    int64
30 2/17/20     263 non-null    int64
31 2/18/20     263 non-null    int64
32 2/19/20     263 non-null    int64
33 2/20/20     263 non-null    int64
34 2/21/20     263 non-null    int64
35 2/22/20     263 non-null    int64
36 2/23/20     263 non-null    int64
37 2/24/20     263 non-null    int64
38 2/25/20     263 non-null    int64
39 2/26/20     263 non-null    int64
40 2/27/20     263 non-null    int64
41 2/28/20     263 non-null    int64
42 2/29/20     263 non-null    int64
43 3/1/20      263 non-null    int64
44 3/2/20      263 non-null    int64
45 3/3/20      263 non-null    int64
46 3/4/20      263 non-null    int64
47 3/5/20      263 non-null    int64
48 3/6/20      263 non-null    int64
49 3/7/20      263 non-null    int64
50 3/8/20      263 non-null    int64
51 3/9/20      263 non-null    int64
52 3/10/20     263 non-null    int64
53 3/11/20     263 non-null    int64
54 3/12/20     263 non-null    int64
55 3/13/20     263 non-null    int64
56 3/14/20     263 non-null    int64
57 3/15/20     263 non-null    int64
58 3/16/20     263 non-null    int64
59 3/17/20     263 non-null    int64
60 3/18/20     263 non-null    int64
61 3/19/20     263 non-null    int64
62 3/20/20     263 non-null    int64
63 3/21/20     263 non-null    int64
64 3/22/20     263 non-null    int64
65 3/23/20     263 non-null    int64
66 3/24/20     263 non-null    int64
67 3/25/20     263 non-null    int64
68 3/26/20     263 non-null    int64
69 3/27/20     263 non-null    int64
70 3/28/20     263 non-null    int64
71 3/29/20     263 non-null    int64
72 3/30/20     263 non-null    int64
73 3/31/20     263 non-null    int64
74 4/1/20      263 non-null    int64
75 4/2/20      263 non-null    int64
76 4/3/20      263 non-null    int64
77 4/4/20      263 non-null    int64
78 4/5/20      263 non-null    int64
79 4/6/20      263 non-null    int64
80 4/7/20      263 non-null    int64
```

dtypes: float64(2), int64(77), object(2)

memory usage: 166.6+ KB

In [5]:

```
raw_data_confirmed.describe()
```

Out[5]:

	Lat	Long	1/22/20	1/23/20	1/24/20	1/25/20	1/26/20	1/27/20	1/28/20	1/29/20	...
count	263.000000	263.000000	263.000000	263.000000	263.000000	263.000000	263.000000	263.000000	263.000000	263.000000	...
mean	21.339244	22.068133	2.110266	2.486692	3.577947	5.452471	8.053232	11.129278	21.209125	23.444867	...
std	24.779585	70.785949	27.434015	27.532888	34.275498	47.702207	66.662110	89.815834	220.427512	221.769901	...
min	-51.796300	-135.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	...
25%	6.938500	-21.031300	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	...
50%	23.634500	20.168300	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	...
75%	41.178850	79.500000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	...
max	71.706900	178.065000	444.000000	444.000000	549.000000	761.000000	1058.000000	1423.000000	3554.000000	3554.000000	...

8 rows × 79 columns

## 1. Jumlah kasus terkonfirmasi COVID-19 berdasarkan negara

Informasi diatas masih sangat general sehingga anda perlu mendapatkan informasi yang lebih spesifik, salah satunya mendapatkan informasi jumlah kasus terkonfirmasi COVID-19 berdasarkan kriteria tertentu. Anda perlu menampilkan dataframe `confirmed_country` yang berisi jumlah kasus terkonfirmasi COVID-19 di setiap negara berdasarkan deret waktu(time series) yang terindeks berdasarkan waktu(date/time) bukan berdasarkan `Country/Region`.

In [6]:

```
# check duplicates
print(raw_data_confirmed.shape)
print(raw_data_confirmed[raw_data_confirmed.duplicated()].shape)
#raw_data_confirmed[raw_data_confirmed.duplicated()]
```

(263, 81)  
(0, 81)

In [7]:

```
cdf = raw_data_confirmed.melt(id_vars=["Province/State", "Country/Region", "Lat", "Long"],
                               var_name="Date",
                               value_name="ConfirmedCases")
cdf["Date"] = pd.to_datetime(cdf["Date"])
cdf["Month"] = cdf["Date"].dt.month_name()
cdf["Yearmonth"] = cdf["Date"].dt.to_period('M')
cdf["Date"] = cdf["Date"].dt.to_period('D')
cdf
```

Out[7]:

	Province/State	Country/Region	Lat	Long	Date	ConfirmedCases	Month	Yearmonth
0	NaN	Afghanistan	33.000000	65.000000	2020-01-22	0	January	2020-01
1	NaN	Albania	41.153300	20.168300	2020-01-22	0	January	2020-01
2	NaN	Algeria	28.033900	1.659600	2020-01-22	0	January	2020-01
3	NaN	Andorra	42.506300	1.521800	2020-01-22	0	January	2020-01
4	NaN	Angola	-11.202700	17.873900	2020-01-22	0	January	2020-01
...	...	...	...	...	...	...	...	...
20246	Falkland Islands (Malvinas)	United Kingdom	-51.796300	-59.523600	2020-04-07	2	April	2020-04
20247	Saint Pierre and Miquelon	France	46.885200	-56.315900	2020-04-07	1	April	2020-04
20248	NaN	South Sudan	6.877000	31.307000	2020-04-07	2	April	2020-04
20249	NaN	Western Sahara	24.215500	-12.885800	2020-04-07	4	April	2020-04
20250	NaN	Sao Tome and Principe	0.18636	6.613081	2020-04-07	4	April	2020-04

20251 rows × 8 columns

In [8]:

```
#data per hari
confirmed_country = pd.crosstab(index=cdf['Date'],
                                 columns=cdf['Country/Region'],
                                 values=cdf['ConfirmedCases'],
                                 aggfunc='sum',
                                 margins=True,
                                 margins_name="TotalCases")
confirmed_country
```

Out[8]:

Country/Region	Afghanistan	Albania	Algeria	Andorra	Angola	Antigua and Barbuda	Argentina	Armenia	Australia	Austria	...	United Kingdom	Uruguay
Date													
2020-01-22	0	0	0	0	0	0	0	0	0	0	0	0	0
2020-01-23	0	0	0	0	0	0	0	0	0	0	0	0	0
2020-01-24	0	0	0	0	0	0	0	0	0	0	0	0	0
2020-01-25	0	0	0	0	0	0	0	0	0	0	0	0	0
2020-01-26	0	0	0	0	0	0	0	0	0	4	0	0	0
...	...	...	...	...	...	...	...	...	...	...	...	...	...
2020-04-04	299	333	1251	466	10	15	1451	770	5550	11781	...	42477	400
2020-04-05	349	361	1320	501	14	15	1451	822	5687	12051	...	48436	400
2020-04-06	367	377	1423	525	16	15	1554	833	5797	12297	...	52279	406
2020-04-07	423	383	1468	545	17	19	1628	853	5895	12639	...	55949	424
TotalCases	3454	4526	13294	6081	129	156	16065	9347	74385	166953	...	467609	5417

78 rows × 185 columns

In [9]:

```
#data per bulan
confirmed_country_m = pd.crosstab(index=cdf['Yearmonth'],
                                    columns=cdf['Country/Region'],
                                    values=cdf['ConfirmedCases'],
                                    aggfunc='sum',
                                    margins=True,
                                    margins_name="TotalCases")
confirmed_country_m
```

Out[9]:

Country/Region	Afghanistan	Albania	Algeria	Andorra	Angola	Antigua and Barbuda	Argentina	Armenia	Australia	Austria	...	United Kingdom	Uruguay
Yearmonth													
2020-01	0	0	0	0	0	0	0	0	38	0	...	2	
2020-02	6	0	5	0	0	0	0	0	431	19	...	242	
2020-03	1219	2232	4823	2787	48	61	6529	4099	35679	84802	...	165497	273
2020-04	2229	2294	8466	3294	81	95	9536	5248	38237	82132	...	301868	268
TotalCases	3454	4526	13294	6081	129	156	16065	9347	74385	166953	...	467609	5417

5 rows × 185 columns

### Hasil Analisis:

Dapat dilihat pada hasil eksplorasi data di atas, beberapa negara sudah mulai mendapat dampak dari COVID-19 pada bulan Januari dan/atau Februari 2020. Diketahui, puncak maraknya COVID-19 berada pada bulan Maret, dengan total 8.894.465 kasus, dihitung dari 185 negara tertentu. Sedangkan, pada bulan April, kasus mulai mengalami penurunan.

Namun, jika dilihat dari segi perkembangan setiap harinya, kasus COVID-19 mengalami peningkatan. Bahkan, meskipun kasus pada bulan April lebih sedikit dibandingkan bulan Maret, masyarakat yang terkena virus COVID-19 ini terus bertambah. Perbedaan ini terjadi akibat kurang lengkapnya data pada bulan Januari dan April.

## 2. Visualisasi kasus terkonfirmasi COVID-19 berdasarkan negara

Anda sudah memiliki sebuah dataframe yang berisi kasus terkonfirmasi COVID-19 yang terindeks berdasarkan waktu. Selanjutnya, visualisasikan data jumlah kasus terkonfirmasi di negara-negara berikut (Prancis, Spanyol, Cina, AS, Italia, dan Australia). Berikan judul, labels, dan spesifikasi (ukuran, warna, ketebalan, dll) yang sesuai, sehingga plot yang dihasilkan rapi, menarik, dan mudah dipahami.

In [10]:

```
#France, Spain, China, US, Italy, Australia
cdf2 = cdf.groupby(['Date', 'Yearmonth', 'Country/Region', 'Month'], as_index=False)[['ConfirmedCases']].sum()
confirmed_map = cdf2[cdf2['Country/Region'] == "France"]
confirmed_map = confirmed_map.append(cdf2[cdf2['Country/Region'] == "Spain"])
confirmed_map = confirmed_map.append(cdf2[cdf2['Country/Region'] == "China"])
confirmed_map = confirmed_map.append(cdf2[cdf2['Country/Region'] == "US"])
confirmed_map = confirmed_map.append(cdf2[cdf2['Country/Region'] == "Italy"])
confirmed_map = confirmed_map.append(cdf2[cdf2['Country/Region'] == "Australia"])
confirmed_map["Date"] = confirmed_map["Date"].astype(str)
confirmed_map["Yearmonth"] = confirmed_map["Yearmonth"].astype(str)
confirmed_map["Month"] = confirmed_map["Month"].astype(str)
confirmed_map
```

Out[10]:

	Date	Yearmonth	Country/Region	Month	ConfirmedCases
61	2020-01-22	2020-01	France	January	0
245	2020-01-23	2020-01	France	January	0
429	2020-01-24	2020-01	France	January	2
613	2020-01-25	2020-01	France	January	3
797	2020-01-26	2020-01	France	January	3
...	...	...	...	...	...
13256	2020-04-03	2020-04	Australia	April	5330
13440	2020-04-04	2020-04	Australia	April	5550
13624	2020-04-05	2020-04	Australia	April	5687
13808	2020-04-06	2020-04	Australia	April	5797
13992	2020-04-07	2020-04	Australia	April	5895

462 rows × 5 columns

In [11]:

```
sns.set(font_scale = 1.3)
sns.barplot(data=confirmed_map,
             x="Month",
             y="ConfirmedCases",
             hue="Country/Region")

plt.title('Total Confirmed Cases of COVID19 on January-April (in Thousands)',
          loc='center', pad=30, fontsize=25, color='darkred')
plt.xlabel('Date', fontsize = 18)
plt.ylabel('Total Cases', fontsize = 18)
plt.grid(color='darkgray', linestyle=':', linewidth=0.5)
plt.ylim(ymin=0, ymax=400000)
labels, locations = plt.yticks()
plt.yticks(labels, (labels/1000).astype(int))
plt.gcf().set_size_inches(15,7)
plt.tight_layout()
plt.show()

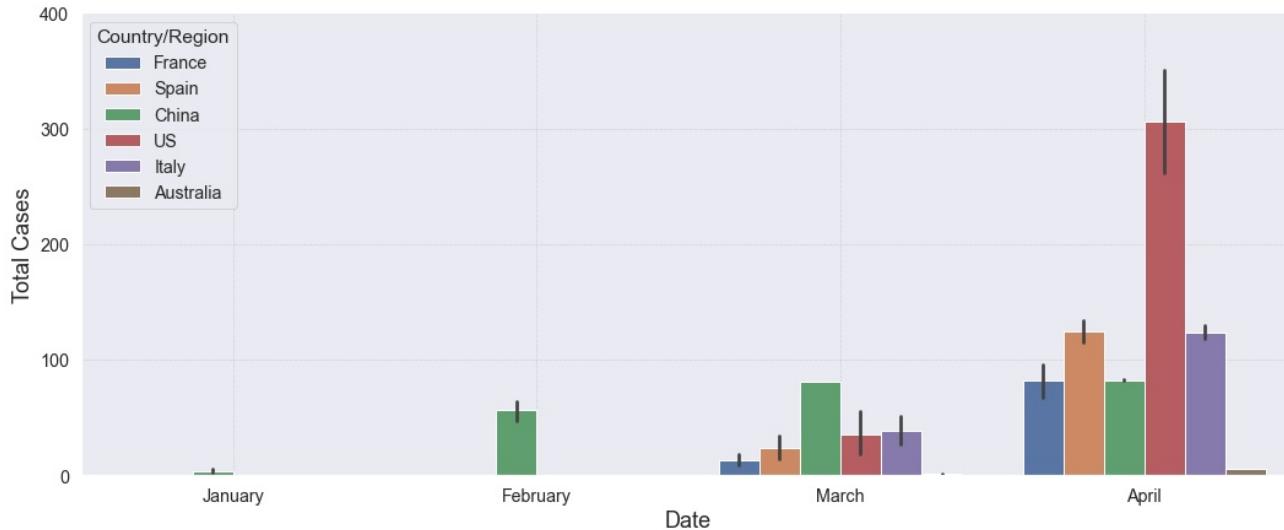
"""

confirmed_country2 = confirmed_country.drop(columns="TotalCases", index="TotalCases")
confirmed_country2.loc[:, ['France', 'Spain', 'China', 'US', 'Italy', 'Australia']].plot()
plt.title('Total Confirmed Cases of COVID19 on January-April (in Millions)',
          loc='center', pad=30, fontsize=25, color='darkred')
plt.xlabel('Date', fontsize = 18)
plt.ylabel('Total Cases', fontsize = 18)
plt.grid(color='darkgray', linestyle=':', linewidth=0.5)
plt.ylim(ymin=0, ymax=410000)
labels, locations = plt.yticks()
plt.yticks(labels, (labels/1000).astype(int))
plt.gcf().set_size_inches(15,7)
plt.tight_layout()
plt.show()

"""


```

Total Confirmed Cases of COVID19 on January-April (in Thousands)



Out[11]:

```
\nconfirmed_country2 = confirmed_country.drop(columns="TotalCases", index="TotalCases")\nconfirmed_country2.loc[:, ['France', 'Spain', 'China', 'US', 'Italy', 'Australia']].plot()\nplt.title('Total Confirmed Cases of COVID19 on January-April (in Millions)',\n          loc='center', pad=30, fontsize=25, color='darkred')\nplt.xlabel('Date', fontsize = 18)\nplt.ylabel('Total Cases', fontsize = 18)\nplt.grid(color='darkgray', linestyle=':', linewidth=0.5)\nplt.ylim(ymin=0, ymax=410000)\nlabels, locations = plt.yticks()\nplt.yticks(labels, (labels/1000).astype(int))\nplt.gcf().set_size_inches(15,7)\nplt.tight_layout()\nplt.show()\n\n'
```

In [12]:

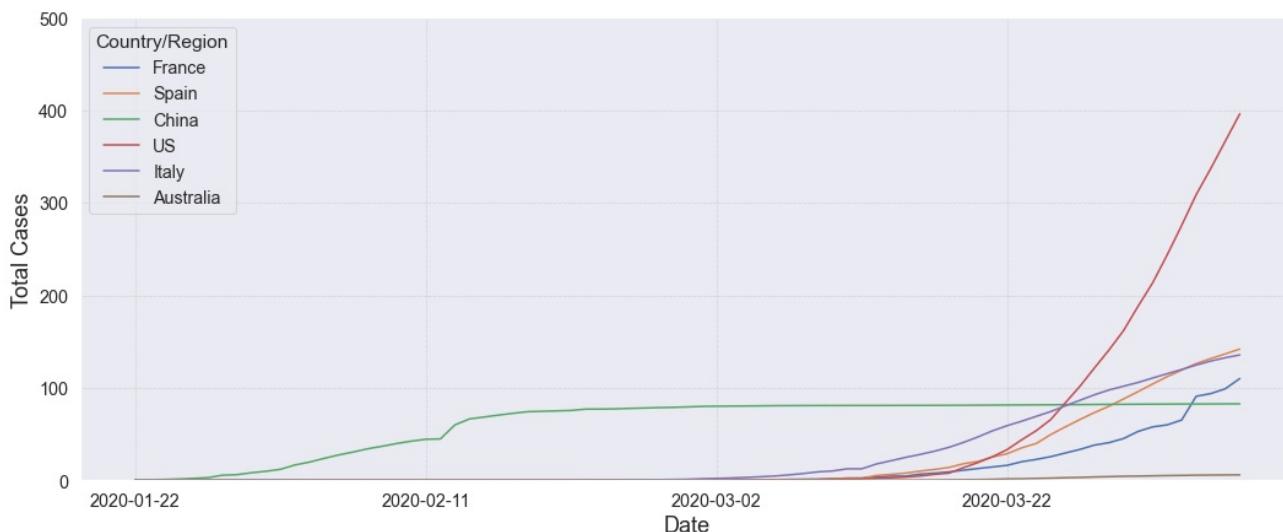
```
sns.set(font_scale = 1.3)
"""
sns.lineplot(data=confirmed_map,
              x="Month",
              y="ConfirmedCases",
              hue="Country/Region")

plt.title('Total Confirmed Cases of COVID19 on January-April (in Millions)',
          loc='center', pad=30, fontsize=25, color='darkred')
plt.xlabel('Date', fontsize = 18)
plt.ylabel('Total Cases', fontsize = 18)
plt.grid(color='darkgray', linestyle=':', linewidth=0.5)
plt.ylim(ymax=4000000)
labels, locations = plt.yticks()
plt.yticks(labels, (labels/1000).astype(int))
plt.gcf().set_size_inches(15,7)
plt.tight_layout()
plt.show()

"""

confirmed_country2 = confirmed_country.drop(columns="TotalCases", index="TotalCases")
confirmed_country2.loc[:, ['France', 'Spain', 'China', 'US', 'Italy', 'Australia']].plot()
plt.title('Total Confirmed Cases of COVID19 on January-April (in Thousands)',
          loc='center', pad=30, fontsize=25, color='darkred')
plt.xlabel('Date', fontsize = 18)
plt.ylabel('Total Cases', fontsize = 18)
plt.grid(color='darkgray', linestyle=':', linewidth=0.5)
plt.ylim(ymax=500000)
labels, locations = plt.yticks()
plt.yticks(labels, (labels/1000).astype(int))
plt.gcf().set_size_inches(15,7)
plt.tight_layout()
plt.show()
```

Total Confirmed Cases of COVID19 on January-April (in Thousands)



#### Hasil Analisis:

- Bar plot menunjukkan perbandingan dari banyaknya kasus COVID-19 di 6 negara (Prancis, Spanyol, Cina, AS, Italia, dan Australia) berdasarkan bulannya. Pada bar plot di atas, diketahui bahwa negara Cina memiliki kasus terbanyak sepanjang bulan Januari hingga Maret. Namun, Cina berhasil mengatasi bertambahnya kasus di bulan April. Di negara lain, kasus mulai berdatangan pada bulan Maret dan terus bertambah seiring berjalananya waktu. Tak diduga, negara AS mengalami peningkatan dengan sangat drastis pada bulan April (kurang lebih 400.000 kasus). Hal ini bahkan melebihi banyaknya kasus yang terjadi di negara Cina ketika sedang berada pada masa puncaknya, yaitu pada bulan Maret dengan total kasus hampir mencapai 100.000 kasus.
- Line Chart berusaha mendeskripsikan pola pertumbuhan COVID-19 pada 6 negara. Pada Line Chart, perhitungan kasus dihitung dengan data harian mulai dari 22 Januari 2020 hingga 7 April 2020 agar bentuk grafik dapat dilihat dengan lebih jelas dan akurat. Grafik ini menjelaskan bahwa negara yang dianggap baik dalam menangani kasus COVID-19 adalah Australia karena jika dibandingkan dengan 5 negara lainnya, Australia memiliki perkembangan konsisten meskipun kasus di beberapa negara kian bertambah. Dapat dikatakan pula bahwa negara Cina lebih baik dalam menangani perkembangan kasus dibandingkan negara AS. Meskipun Cina memiliki kasus terbanyak selama 3 bulan, namun, secara perlahan, mereka berhasil mengurangi perkembangan kasus. Sedangkan, AS terlihat belum siap dalam menangani COVID-19 di negaranya. Hari demi hari, kasus di negara AS semakin meningkat.

## VISUALISASI TAMBAHAN

(menggunakan choropleth map) -> tujuannya adalah untuk mempermudah pembaca dalam memahami jumlah kasus COVID-19 pada 6 negara tertentu setiap harinya. Dengan visualisasi menggunakan choropleth map, jumlah kasus akan ditampilkan dengan hasil akurat (bukan perkiraan). Contoh: ditampilkan dengan jelas, terdapat 59.895 kasus di negara Cina pada tanggal 13 Februari 2020 (ketika dihover ke wilayah peta sesuai negara). Selain itu, terdapat pula pewarnaan pada peta yang menunjukkan tingkat keparahan kasus yang terjadi.

In [13]:

```
"""import pycountry
import pycountry_convert as pc
import plotly_express as px

class country_utils():
    def __init__(self):
        self.d = {}

    def get_dic(self):
        return self.d

    def get_country_details(self,country):
        # Returns country code(alpha_3) and continent
        try:
            country_obj = pycountry.countries.get(name=country)
            if country_obj is None:
                c = pycountry.countries.search_fuzzy(country)
                country_obj = c[0]
            continent_code = pc.country_alpha2_to_continent_code(country_obj.alpha_2)
            continent = pc.convert_continent_code_to_continent_name(continent_code)
            return country_obj.alpha_3, continent
        except:
            if 'France' in country:
                country = 'France'
            elif country == 'Spain':
                country = 'Spain'
            elif country == 'China':
                country = 'China'
            elif country == 'US':
                country = 'US'
            elif country == 'Italy':
                country = 'Italy'
            elif country == 'Australia':
                country = 'Australia'
            else:
                return country, country
            country_obj = pycountry.countries.search_fuzzy(country)
            continent_code = pc.country_alpha2_to_continent_code(country_obj[0].alpha_2)
            continent = pc.convert_continent_code_to_continent_name(continent_code)
            return country_obj[0].alpha_3, continent

    def get_iso3(self, country):
        return self.d[country]['code']

    def get_continent(self, country):
        return self.d[country]['continent']

    def add_values(self, country):
        self.d[country] = {}
        self.d[country]['code'], self.d[country]['continent'] = self.get_country_details(country)

    def fetch_iso3(self, country):
        if country in self.d.keys():
            return self.get_iso3(country)
        else:
            self.add_values(country)
            return self.get_iso3(country)

    def fetch_continent(self, country):
        if country in self.d.keys():
            return self.get_continent(country)
        else:
            self.add_values(country)
            return self.get_continent(country)
"""

```

Out[13]:

```
"\nimport pycountry\nimport pycountry_convert as pc\nimport plotly_express as px\n\nclass country_utilities():\n    def __init__(self):\n        self.d = {} \n\n    def get_dict(self):\n        return self.d\n\n    def get_country_details(self, country):\n        # Returns country code(alpha_3) and continent\n        try:\n            country_obj = pycountry.countries.get(name=country)\n            if country_obj is None:\n                c = pycountry.countries.search_fuzzy(country)\n                country_obj = c[0]\n                continent_code = pc.country_alpha2_to_continent_code(country_obj.alpha_2)\n                continent = pc.convert_continent_code_to_continent_name(continent_code)\n            return country_obj.alpha_3, continent\n        except:\n            if 'France' in country:\n                country = 'France'\n            elif country == 'Spain':\n                country = 'Spain'\n            elif country == 'China':\n                country = 'China'\n            elif country == 'US':\n                country = 'US'\n            elif country == 'Italy':\n                country = 'Italy'\n            elif country == 'Australia':\n                country = 'Australia'\n            else:\n                return country, country\n                country_obj = pycountry.countries.search_fuzzy(country)\n                continent_code = pc.country_alpha2_to_continent_code(country_obj[0].alpha_2)\n                continent = pc.convert_continent_code_to_continent_name(continent_code)\n            return country_obj[0].alpha_3, continent\n\n    def get_iso3(self, country):\n        return self.d[country]['code']\n\n    def get_continent(self, country):\n        return self.d[country]['continent']\n\n    def add_values(self, country):\n        self.d[country] = {}\n        self.d[country]['code'], self.d[country]['continent'] = self.get_country_details(country)\n        self.fetch_iso3(self, country):\n            if country in self.d.keys():\n                return self.get_iso3(country)\n            else:\n                self.add_values(country)\n                return self.get_iso3(country)\n\n    def fetch_continent(self, country):\n        if country in self.d.keys():\n            return self.get_continent(country)\n        else:\n            self.add_values(country)\n            return self.get_continent(country)\n"\n
```

In [14]:

```
cdf_map = confirmed_map.copy()
cdf_map['Date'] = cdf_map['Date'].astype(str)
cdf_map = cdf_map.groupby(['Date', 'Country/Region'], as_index=False)[['ConfirmedCases']].sum()

obj = country_utils()
cdf_map['iso_alpha'] = cdf_map.apply(lambda x: obj.fetch_iso3(x['Country/Region']), axis=1)
cdf_map['log(ConfirmedCases)'] = np.log(cdf_map.ConfirmedCases + 1)

px.choropleth(cdf_map,
              locations="iso_alpha",
              color="log(ConfirmedCases)",
              hover_name="Country/Region",
              hover_data=["ConfirmedCases"] ,
              animation_frame="Date",
              color_continuous_scale=px.colors.sequential.dense,
              title='Total Confirmed Cases Growth of COVID-19 (Logarithmic Scale)')
```

Out[14]:

```
'\nncdf_map = confirmed_map.copy()\nncdf_map[\\"Date\\"] = cdf_map[\\"Date\\"].astype(str)\nncdf_map = cdf_map.groupby([\\"Date\\", \\"Country/Region\\"], as_index=False)[\\"ConfirmedCases\\"].sum()\n\nobj = country_utils()\nncdf_map[\\"iso_alpha\\"] = cdf_map.apply(lambda x: obj.fetch_iso3(x[\\"Country/Region\\"]), axis=1)\nncdf_map[\\"log(ConfirmedCases)\\"] = np.log(cdf_map.ConfirmedCases + 1)\n\npx choropleth(cdf_map, \n    locations="iso_alpha", \n    color="log(ConfirmedCases)", \n    hover_name="Country/Region", \n    hover_data=[\"ConfirmedCases\"], \n    animation_frame="Date", \n    color_continuous_scale=px.colors.sequential.dense, \n    title='Total Confirmed Cases Growth of COVID-19 (Logarithmic Scale)')\n'
```

### 3. Jumlah kasus kematian COVID-19 yang dilaporkan berdasarkan negara

Selain informasi kasus terkonfirmasi, anda juga perlu mendapatkan informasi mengenai kasus kematian COVID-19. Tampilkan beberapa baris teratas/terbawah beserta informasi dari dataframe raw data deaths

In [15]:

```
# Beberapa baris teratas dari dataframe raw_data_death  
raw_data_deaths.head()
```

Out[15]:

Province/State	Country/Region	Lat	Long	1/22/20	1/23/20	1/24/20	1/25/20	1/26/20	1/27/20	...	3/29/20	3/30/20	3/31/20	4/1/2
0	NaN	Afghanistan	33.0000	65.0000	0	0	0	0	0	0	...	4	4	4
1	NaN	Albania	41.1533	20.1683	0	0	0	0	0	0	...	10	11	15
2	NaN	Algeria	28.0339	1.6596	0	0	0	0	0	0	...	31	35	44
3	NaN	Andorra	42.5063	1.5218	0	0	0	0	0	0	...	6	8	12
4	NaN	Angola	-11.2027	17.8739	0	0	0	0	0	0	...	2	2	2

5 rows × 81 columns

In [16]:

```
# Beberapa baris terbawah dari dataframe raw_data_death  
raw_data_deaths.tail()
```

Out[16]:

Province/State	Country/Region	Lat	Long	1/22/20	1/23/20	1/24/20	1/25/20	1/26/20	1/27/20	...	3/29/20	3/30/20	3/31/
258	Falkland Islands (Malvinas)	United Kingdom	-51.79630	-59.523600	0	0	0	0	0	0	...	0	0
259	Saint Pierre and Miquelon	France	46.88520	-56.315900	0	0	0	0	0	0	...	0	0
260	NaN	South Sudan	6.87700	31.307000	0	0	0	0	0	0	...	0	0
261	NaN	Western Sahara	24.21550	-12.885800	0	0	0	0	0	0	...	0	0
262	NaN	Sao Tome and Principe	0.18636	6.613081	0	0	0	0	0	0	...	0	0

5 rows × 81 columns

In [17]:

```
# informasi dataframe raw_data_death  
raw_data_deaths.info()
```

```
<class 'pandas.core.frame.DataFrame'>  
RangeIndex: 263 entries, 0 to 262  
Data columns (total 81 columns):  
 #   Column           Non-Null Count  Dtype     
---  --  
 0   Province/State  82 non-null     object    
 1   Country/Region  263 non-null    object    
 2   Lat              263 non-null    float64  
 3   Long             263 non-null    float64  
 4   1/22/20          263 non-null    int64    
 5   1/23/20          263 non-null    int64    
 6   1/24/20          263 non-null    int64    
 7   1/25/20          263 non-null    int64    
 8   1/26/20          263 non-null    int64    
 9   1/27/20          263 non-null    int64    
 10  1/28/20          263 non-null    int64    
 11  1/29/20          263 non-null    int64    
 12  1/30/20          263 non-null    int64    
 13  1/31/20          263 non-null    int64    
 14  2/1/20           263 non-null    int64    
 15  2/2/20           263 non-null    int64    
 16  2/3/20           263 non-null    int64    
 17  2/4/20           263 non-null    int64    
 18  2/5/20           263 non-null    int64    
 19  2/6/20           263 non-null    int64    
 20  2/7/20           263 non-null    int64    
 21  2/8/20           263 non-null    int64    
 22  2/9/20           263 non-null    int64    
 23  2/10/20          263 non-null    int64    
 24  2/11/20          263 non-null    int64    
 25  2/12/20          263 non-null    int64    
 26  2/13/20          263 non-null    int64    
 27  2/14/20          263 non-null    int64    
 28  2/15/20          263 non-null    int64    
 29  2/16/20          263 non-null    int64
```

```

30 2/17/20          263 non-null    int64
31 2/18/20          263 non-null    int64
32 2/19/20          263 non-null    int64
33 2/20/20          263 non-null    int64
34 2/21/20          263 non-null    int64
35 2/22/20          263 non-null    int64
36 2/23/20          263 non-null    int64
37 2/24/20          263 non-null    int64
38 2/25/20          263 non-null    int64
39 2/26/20          263 non-null    int64
40 2/27/20          263 non-null    int64
41 2/28/20          263 non-null    int64
42 2/29/20          263 non-null    int64
43 3/1/20           263 non-null    int64
44 3/2/20           263 non-null    int64
45 3/3/20           263 non-null    int64
46 3/4/20           263 non-null    int64
47 3/5/20           263 non-null    int64
48 3/6/20           263 non-null    int64
49 3/7/20           263 non-null    int64
50 3/8/20           263 non-null    int64
51 3/9/20           263 non-null    int64
52 3/10/20          263 non-null   int64
53 3/11/20          263 non-null   int64
54 3/12/20          263 non-null   int64
55 3/13/20          263 non-null   int64
56 3/14/20          263 non-null   int64
57 3/15/20          263 non-null   int64
58 3/16/20          263 non-null   int64
59 3/17/20          263 non-null   int64
60 3/18/20          263 non-null   int64
61 3/19/20          263 non-null   int64
62 3/20/20          263 non-null   int64
63 3/21/20          263 non-null   int64
64 3/22/20          263 non-null   int64
65 3/23/20          263 non-null   int64
66 3/24/20          263 non-null   int64
67 3/25/20          263 non-null   int64
68 3/26/20          263 non-null   int64
69 3/27/20          263 non-null   int64
70 3/28/20          263 non-null   int64
71 3/29/20          263 non-null   int64
72 3/30/20          263 non-null   int64
73 3/31/20          263 non-null   int64
74 4/1/20           263 non-null   int64
75 4/2/20           263 non-null   int64
76 4/3/20           263 non-null   int64
77 4/4/20           263 non-null   int64
78 4/5/20           263 non-null   int64
79 4/6/20           263 non-null   int64
80 4/7/20           263 non-null   int64

```

dtypes: float64(2), int64(77), object(2)

memory usage: 166.6+ KB

In [18]:

```
raw_data_deaths.describe()
```

Out[18]:

	Lat	Long	1/22/20	1/23/20	1/24/20	1/25/20	1/26/20	1/27/20	1/28/20	1/29/20	...
<b>count</b>	263.000000	263.000000	263.000000	263.000000	263.000000	263.000000	263.000000	263.000000	263.000000	263.000000	...
<b>mean</b>	21.339244	22.068133	0.064639	0.068441	0.098859	0.159696	0.212928	0.311787	0.498099	0.505703	...
<b>std</b>	24.779585	70.785949	1.048265	1.049842	1.481991	2.467570	3.207860	4.687336	7.707870	7.708363	...
<b>min</b>	-51.796300	-135.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	...
<b>25%</b>	6.938500	-21.031300	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	...
<b>50%</b>	23.634500	20.168300	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	...
<b>75%</b>	41.178850	79.500000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	...
<b>max</b>	71.706900	178.065000	17.000000	17.000000	24.000000	40.000000	52.000000	76.000000	125.000000	125.000000	1

8 rows × 79 columns

### Hasil Analisis:

Dari data raw di atas, diketahui bahwa jumlah negara yang terdapat pada data adalah sebanyak 263 negara. Masing-masing negara memiliki kasus kematian yang berbeda-beda setiap harinya. Sekilas, dapat diketahui jumlah kasus paling sedikit adalah sebanyak 0. Jika dilihat dari 5 data teratas, pada negara-negara tersebut belum terdapat kasus kematian di awal bulan. Namun, seiring berjalanannya waktu, kasus kematian mulai meningkat. Sedangkan, pada 5 data terbawah, negara-negara tersebut masih belum memunculkan tanda-tanda adanya kasus kematian pada awal maupun akhir bulan. Data di atas masih dianggap sebagai data mentah/kasar, sehingga kita tidak dapat menganalisis detail dari keseluruhan data. Oleh karena itu, untuk sementara ini, kesimpulan yang dapat diambil tidak banyak. Untuk analisis lebih lanjut, data dapat dieksplorasi di bagian selanjutnya.

Selanjutnya, tampilkan beberapa baris teratas/terbawah data kasus kematian COVID-19 di setiap negara yang terindeks berdasarkan waktu(date/time) bukan berdasarkan Country/Region .

In [19]:

```
# check duplicates
print(raw_data_deaths.shape)
print(raw_data_deaths[raw_data_deaths.duplicated()].shape)
```

```
(263, 81)
(0, 81)
```

In [20]:

```
ddf = raw_data_deaths.melt(id_vars=["Province/State", "Country/Region", "Lat", "Long"],
                           var_name="Date",
                           value_name="DeathCases")
ddf["Date"] = pd.to_datetime(ddf["Date"])
ddf["Month"] = ddf["Date"].dt.month_name()
ddf["Yearmonth"] = ddf["Date"].dt.to_period('M')
ddf["Date"] = ddf["Date"].dt.to_period('D')
ddf
```

Out[20]:

	Province/State	Country/Region	Lat	Long	Date	DeathCases	Month	Yearmonth
0	NaN	Afghanistan	33.00000	65.000000	2020-01-22	0	January	2020-01
1	NaN	Albania	41.15330	20.168300	2020-01-22	0	January	2020-01
2	NaN	Algeria	28.03390	1.659600	2020-01-22	0	January	2020-01
3	NaN	Andorra	42.50630	1.521800	2020-01-22	0	January	2020-01
4	NaN	Angola	-11.20270	17.873900	2020-01-22	0	January	2020-01
...	...	...	...	...	...	...	...	...
20246	Falkland Islands (Malvinas)	United Kingdom	-51.79630	-59.523600	2020-04-07	0	April	2020-04
20247	Saint Pierre and Miquelon	France	46.88520	-56.315900	2020-04-07	0	April	2020-04
20248	NaN	South Sudan	6.87700	31.307000	2020-04-07	0	April	2020-04
20249	NaN	Western Sahara	24.21550	-12.885800	2020-04-07	0	April	2020-04
20250	NaN	Sao Tome and Principe	0.18636	6.613081	2020-04-07	0	April	2020-04

20251 rows × 8 columns

In [21]:

```
#data per hari
death_country = pd.crosstab(index=ddf['Date'],
                             columns=ddf['Country/Region'],
                             values=ddf['DeathCases'],
                             aggfunc='sum',
                             margins=True,
                             margins_name="TotalCases")
death_country
```

Out[21]:

Country/Region	Afghanistan	Albania	Algeria	Andorra	Angola	Antigua and Barbuda	Argentina	Armenia	Australia	Austria	...	United Kingdom	Uruguay
Date													
2020-01-22	0	0	0	0	0	0	0	0	0	0	0	0	0
2020-01-23	0	0	0	0	0	0	0	0	0	0	0	0	0
2020-01-24	0	0	0	0	0	0	0	0	0	0	0	0	0
2020-01-25	0	0	0	0	0	0	0	0	0	0	0	0	0
2020-01-26	0	0	0	0	0	0	0	0	0	0	0	0	0
...	...	...	...	...	...	...	...	...	...	...	...	...	...
2020-04-04	7	20	130	17	2	0	43	7	30	186	...	4320	5
2020-04-05	7	20	152	18	2	0	44	7	35	204	...	4943	5
2020-04-06	11	21	173	21	2	0	48	8	40	220	...	5385	6
2020-04-07	14	22	193	22	2	1	56	8	45	243	...	6171	7
TotalCases	84	222	1221	162	20	1	452	60	412	1951	...	38835	36

78 rows × 185 columns

In [22]:

```
#data per bulan
death_country_m = pd.crosstab(index=ddf['Yearmonth'],
                               columns=ddf['Country/Region'],
                               values=ddf['DeathCases'],
                               aggfunc='sum',
                               margins=True,
                               margins_name="TotalCases")
death_country_m
```

Out[22]:

Country/Region	Afghanistan	Albania	Algeria	Andorra	Angola	Antigua and Barbuda	Argentina	Armenia	Australia	Austria	...	United Kingdom	Uruguay
Yearmonth													
2020-01	0	0	0	0	0	0	0	0	0	0	0	0	0
2020-02	0	0	0	0	0	0	0	0	0	0	0	0	0
2020-03	29	91	324	39	6	0	158	12	190	626	...	9122	
2020-04	55	131	897	123	14	1	294	48	222	1325	...	29713	
TotalCases	84	222	1221	162	20	1	452	60	412	1951	...	38835	

5 rows × 185 columns

In [23]:

```
#data per hari (teratas)
pd.crosstab(index=ddf['Date'],
            columns=ddf['Country/Region'],
            values=ddf['DeathCases'],
            aggfunc='sum').head()
```

Out[23]:

Country/Region	Afghanistan	Albania	Algeria	Andorra	Angola	Antigua and Barbuda	Argentina	Armenia	Australia	Austria	...	United Arab Emirates	United Kingdom
Date													
2020-01-22	0	0	0	0	0	0	0	0	0	0	0	0	0
2020-01-23	0	0	0	0	0	0	0	0	0	0	0	0	0
2020-01-24	0	0	0	0	0	0	0	0	0	0	0	0	0
2020-01-25	0	0	0	0	0	0	0	0	0	0	0	0	0
2020-01-26	0	0	0	0	0	0	0	0	0	0	0	0	0

5 rows × 184 columns

In [24]:

```
#data per hari (terbawah)
pd.crosstab(index=ddf['Date'],
            columns=ddf['Country/Region'],
            values=ddf['DeathCases'],
            aggfunc='sum').tail()
```

Out[24]:

Country/Region	Afghanistan	Albania	Algeria	Andorra	Angola	Antigua and Barbuda	Argentina	Armenia	Australia	Austria	...	United Arab Emirates	United Kingdom
Date													
2020-04-03	6	17	105	16	2	0	39	7	28	168	...	9	36
2020-04-04	7	20	130	17	2	0	43	7	30	186	...	10	43
2020-04-05	7	20	152	18	2	0	44	7	35	204	...	10	49
2020-04-06	11	21	173	21	2	0	48	8	40	220	...	11	53
2020-04-07	14	22	193	22	2	1	56	8	45	243	...	12	61

5 rows × 184 columns

In [25]:

```
# type your codes to check out the index here
death_country.drop(index="TotalCases").index
```

Out[25]:

```
Index(['2020-01-22', '2020-01-23', '2020-01-24', '2020-01-25', '2020-01-26', '2020-01-27',  
       '2020-01-28', '2020-01-29', '2020-01-30', '2020-01-31', '2020-02-01', '2020-02-02',  
       '2020-02-03', '2020-02-04', '2020-02-05', '2020-02-06', '2020-02-07', '2020-02-08',  
       '2020-02-09', '2020-02-10', '2020-02-11', '2020-02-12', '2020-02-13', '2020-02-14',  
       '2020-02-15', '2020-02-16', '2020-02-17', '2020-02-18', '2020-02-19', '2020-02-20',  
       '2020-02-21', '2020-02-22', '2020-02-23', '2020-02-24', '2020-02-25', '2020-02-26',  
       '2020-02-27', '2020-02-28', '2020-02-29', '2020-03-01', '2020-03-02', '2020-03-03',  
       '2020-03-04', '2020-03-05', '2020-03-06', '2020-03-07', '2020-03-08', '2020-03-09',  
       '2020-03-10', '2020-03-11', '2020-03-12', '2020-03-13', '2020-03-14', '2020-03-15',  
       '2020-03-16', '2020-03-17', '2020-03-18', '2020-03-19', '2020-03-20', '2020-03-21',  
       '2020-03-22', '2020-03-23', '2020-03-24', '2020-03-25', '2020-03-26', '2020-03-27',  
       '2020-03-28', '2020-03-29', '2020-03-30', '2020-03-31', '2020-04-01', '2020-04-02',  
       '2020-04-03', '2020-04-04', '2020-04-05', '2020-04-06', '2020-04-07'],  
      dtype='object', name='Date')
```

### Hasil Analisis:

Dengan melihat 5 data teratas, dapat diketahui bahwa terdapat banyak negara yang masih belum mendapatkan kasus kematian pada bulan Januari. Namun, pada bulan April, keberadaan virus COVID-19 ini mulai berdampak besar hingga menelan banyak korban di berbagai negara. Jika dilihat dari data harian, total kasus kematian di seluruh negara mengalami peningkatan setiap harinya. Karena hari memiliki korelasi yang tinggi dengan bulan, maka dapat disimpulkan pula bahwa kasus kematian juga semakin meningkat setiap bulan. Hal ini dapat dibuktikan dengan memperhatikan data per bulannya, dimana ditunjukkan bahwa setiap bulan terjadi peningkatan kasus kematian. Bahkan pada bulan April pun, kasus kematian belum juga mengalami penurunan dan masih terus meningkat. Contohnya pada negara United Kingdom, dimana kasus kematian terus meningkat hingga mencapai total kasus sebanyak 29.713 kasus hanya dalam waktu seminggu (1 April 2020 - 7 April 2020).

## 4. Penyelarasan kurva pertumbuhan seluruh negara dengan jumlah kasus kematian COVID-19 $\geq 25$

Untuk mendapatkan gambaran perkembangan kasus covid-19 di berbagai negara, diperlukan penyelarasan kurva pertumbuhan setiap negara. Kurva hanya menampilkan informasi yang dimulai dengan hari dimana data kasus kematian COVID-19 setidaknya 25 orang.

In [26]:

```
#type your codes here
death = death_country.drop(index="TotalCases", columns="TotalCases").reset_index().melt(id_vars="Date")
death = death.rename(columns={'value': 'DeathCases'})
conf = confirmed_country.drop(index="TotalCases", columns="TotalCases").reset_index().melt(id_vars="Date")
conf = conf.rename(columns={'value': 'ConfirmedCases'})

conf_death = death.copy()
conf_death['ConfirmedCases'] = conf['ConfirmedCases']
conf_death = conf_death[conf_death['DeathCases'] >= 25]
conf_death['Date'] = conf_death['Date'].astype(str)
conf_death
```

Out[26]:

	Date	Country/Region	DeathCases	ConfirmedCases
218	2020-03-26	Algeria	25	367
219	2020-03-27	Algeria	26	409
220	2020-03-28	Algeria	29	454
221	2020-03-29	Algeria	31	511
222	2020-03-30	Algeria	35	584
...	...	...	...	...
13547	2020-04-03	United Kingdom	3611	38689
13548	2020-04-04	United Kingdom	4320	42477
13549	2020-04-05	United Kingdom	4943	48436
13550	2020-04-06	United Kingdom	5385	52279
13551	2020-04-07	United Kingdom	6171	55949

837 rows  $\times$  4 columns

In [27]:

```
conf_death.sort_values(by="Date")
```

Out[27]:

	Date	Country/Region	DeathCases	ConfirmedCases
2774	2020-01-24	China	26	920
2775	2020-01-25	China	42	1406
2776	2020-01-26	China	56	2075
2777	2020-01-27	China	82	2877
2778	2020-01-28	China	131	5509
...	...	...	...	...
9778	2020-04-07	Norway	89	6086
9701	2020-04-07	North Macedonia	26	599
9316	2020-04-07	Netherlands	2108	19709
692	2020-04-07	Australia	45	5895
13551	2020-04-07	United Kingdom	6171	55949

837 rows × 4 columns

In [28]:

```
death2 = death_country.drop(index="TotalCases").reset_index()
conf2 = confirmed_country.drop(index="TotalCases")

all_ctry = pd.crosstab(index=death2["Date"],
                       columns="DeathCases",
                       values=death2["TotalCases"],
                       aggfunc='sum')
all_ctry["ConfirmedCases"] = conf2["TotalCases"]
all_ctry = all_ctry[all_ctry['DeathCases'] >= 25]
all_ctry = all_ctry.reset_index()
all_ctry['Date'] = all_ctry['Date'].astype(str)
all_ctry
```

Out[28]:

col_0	Date	DeathCases	ConfirmedCases
0	2020-01-24	26	941
1	2020-01-25	42	1434
2	2020-01-26	56	2118
3	2020-01-27	82	2927
4	2020-01-28	131	5578
...	...	...	...
70	2020-04-03	58787	1095917
71	2020-04-04	64606	1197405
72	2020-04-05	69374	1272115
73	2020-04-06	74565	1345101
74	2020-04-07	81865	1426096

75 rows × 3 columns

#### Hasil Analisis:

Berdasarkan data di atas, diketahui bahwa jumlah kasus yang terkonfirmasi terkena COVID-19 lebih banyak dibandingkan kasus kematiannya. Hal ini menunjukkan adanya keseimbangan antara angka kematian dan angka pertumbuhan dari COVID-19 di seluruh negara, baik secara global atau masing-masing negara. Ditunjukkan pula pada tabel, bahwa kasus dengan angka kematian  $\geq 25$  dimulai sejak bulan Januari, dengan China sebagai negara pertama yang gagal menyelamatkan nyawa 26 orang pada tanggal 24 Januari 2020. Dari sanalah, kasus semakin menyebar hingga meningkatkan angka kematian di berbagai negara.

Visualisasikan hasil diatas dan berikan/atur judul, labels, dan spesifikasi (ukuran, warna, ketebalan, dll) yang sesuai sehingga plot/kurva yang dihasilkan rapi, menarik, dan mudah dipahami.

In [29]:

```
#plot time series
# type your codes here
country = conf_death['Country/Region'].unique()

for i in country:
    ctr = conf_death[conf_death['Country/Region'] == i]
    sns.set(font_scale = 1.3)
    ttl_line = "{}'s Growth Curve Alignment with COVID-19 Death Toll of At Least 25 Cases".format(i)
    ttl_bar = "Daily Cases and Death Count of COVID-19 from January - April in {}".format(i)

    #line
    fig, ax = plt.subplots()
    # twin object for two different y-axis on the sample plot
    ax2=ax.twinx()
    # make a plot
    ax.plot(ctr['Date'], ctr['ConfirmedCases'], color="blue", marker="o", label="Confirmed Cases")
    # set x-axis label
    ax.set_xlabel("Date", fontsize=18)
    # set y-axis label
    ax.set_ylabel("Confirmed Cases", color="blue", fontsize=14)

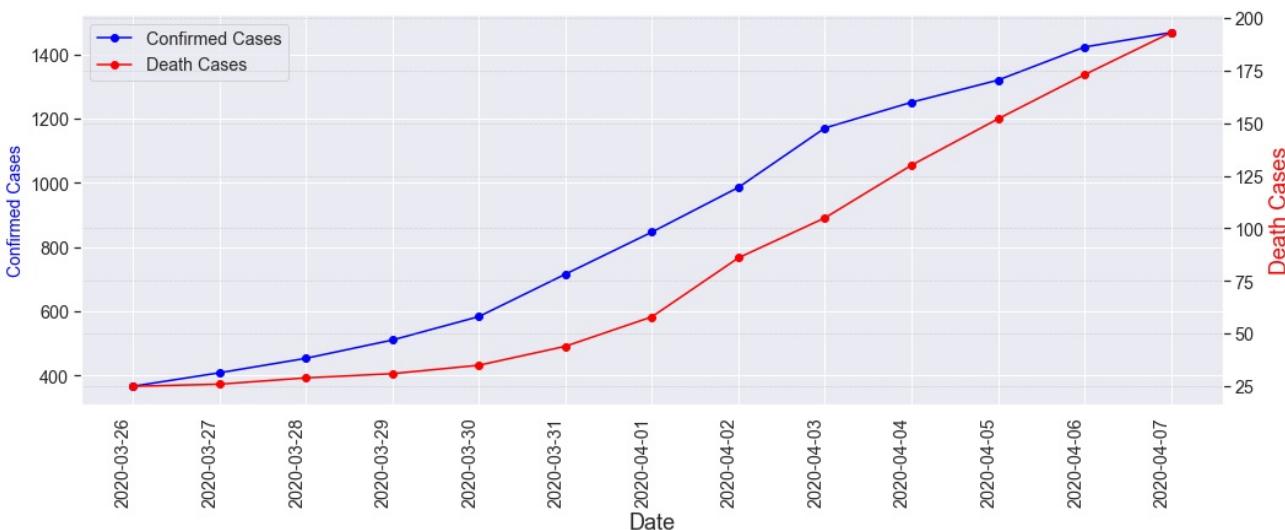
    # make a plot with different y-axis using second axis object
    ax2.plot(ctr['Date'], ctr['DeathCases'], color="red", marker="o", label="Death Cases")
    ax2.set_ylabel("Death Cases", color="red", fontsize=18)

    #get legend
    lines, labels = ax.get_legend_handles_labels()
    lines2, labels2 = ax2.get_legend_handles_labels()
    ax.legend(lines + lines2, labels + labels2, loc=0)

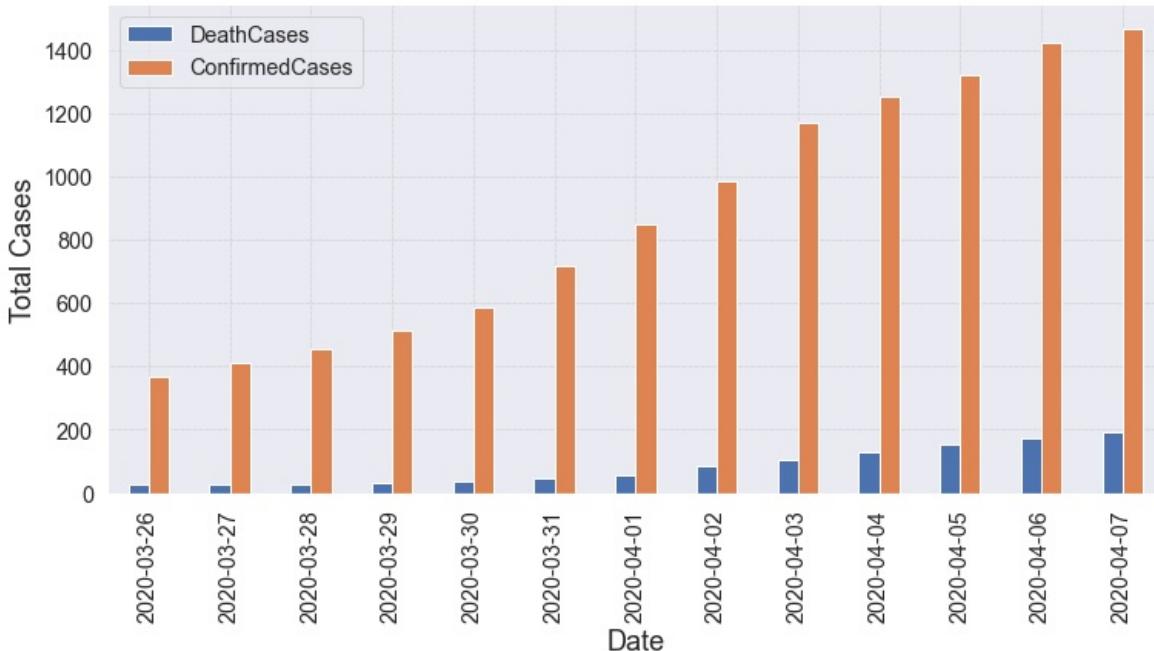
    plt.title(ttl_line, loc='center', pad=30, fontsize=25, color='darkred')
    ax.set_xticklabels(ctr['Date'], rotation=90, ha='right')
    plt.grid(color='darkgray', linestyle=':', linewidth=0.5)
    plt.gcf().set_size_inches(15,7)
    fig.tight_layout()

    #bar
    ctr.plot(x="Date", kind="bar")
    plt.title(ttl_bar, loc='center', pad=30, fontsize=25, color='darkred')
    plt.xlabel('Date', fontsize = 18)
    plt.ylabel('Total Cases', fontsize = 18)
    plt.grid(color='darkgray', linestyle=':', linewidth=0.5)
    plt.ylim(ymin=0)
    plt.gcf().set_size_inches(12,7)
    plt.tight_layout()
    plt.show()
```

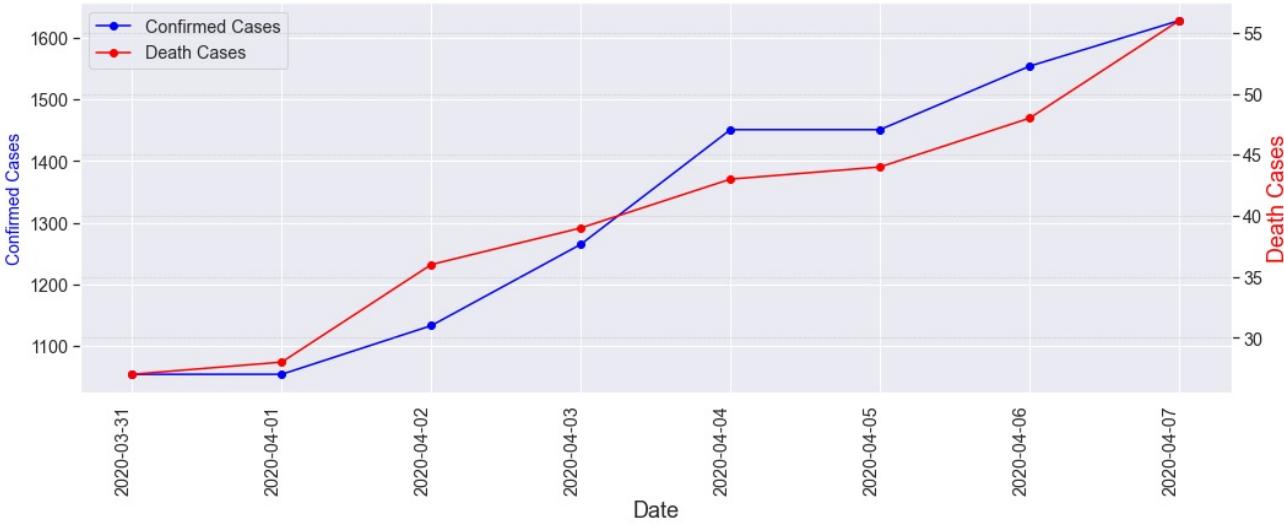
### Algeria's Growth Curve Alignment with COVID-19 Death Toll of At Least 25 Cases



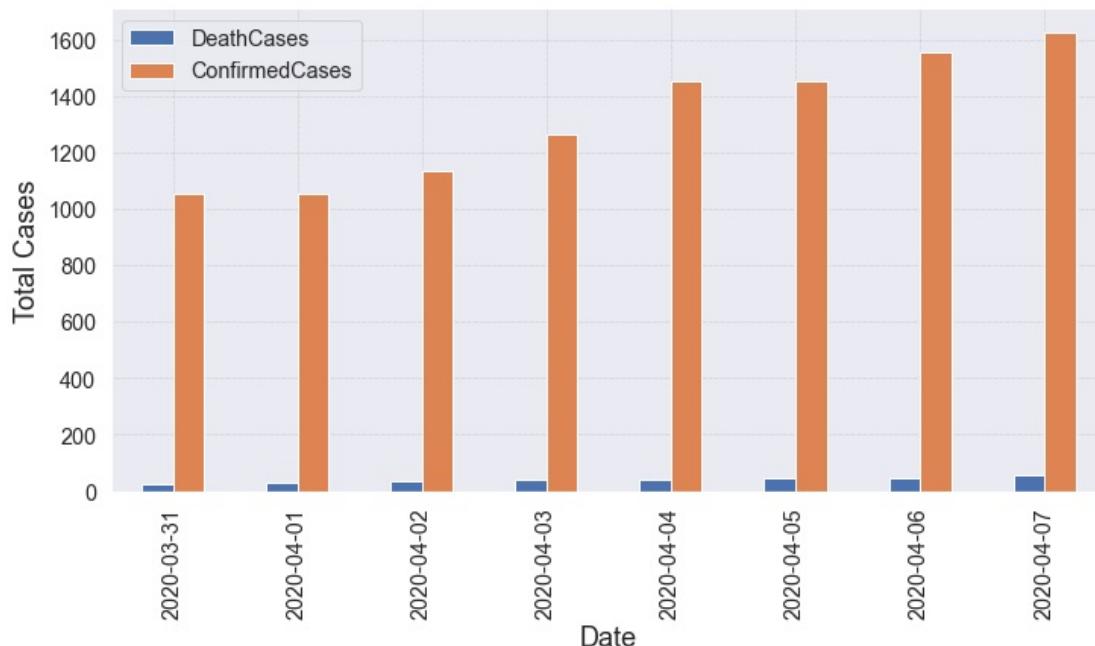
## Daily Cases and Death Count of COVID-19 from January - April in Algeria



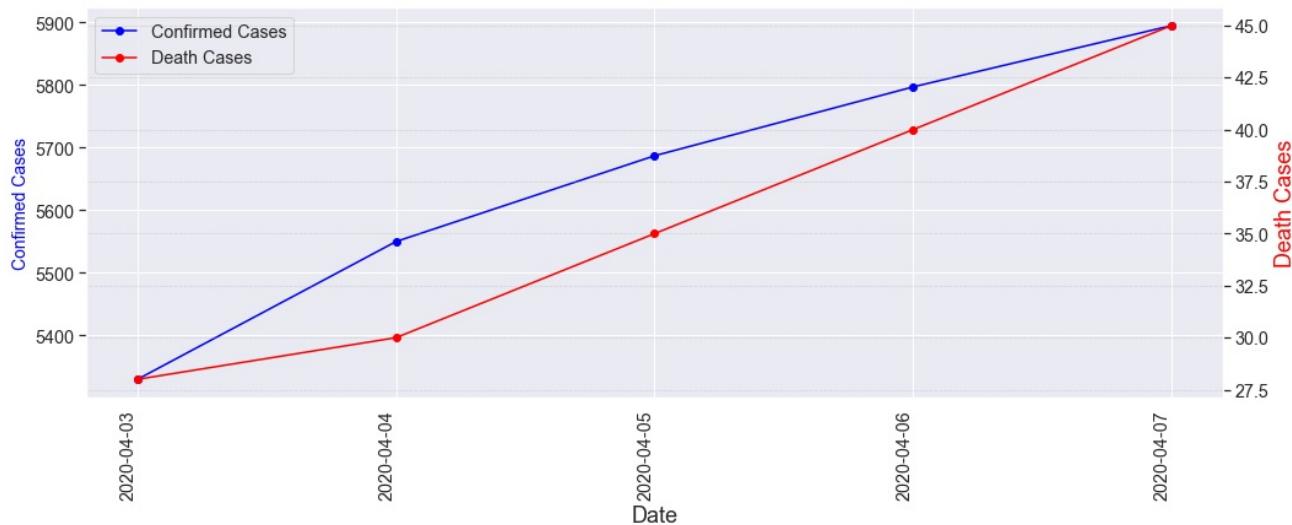
## Argentina's Growth Curve Alignment with COVID-19 Death Toll of At Least 25 Cases



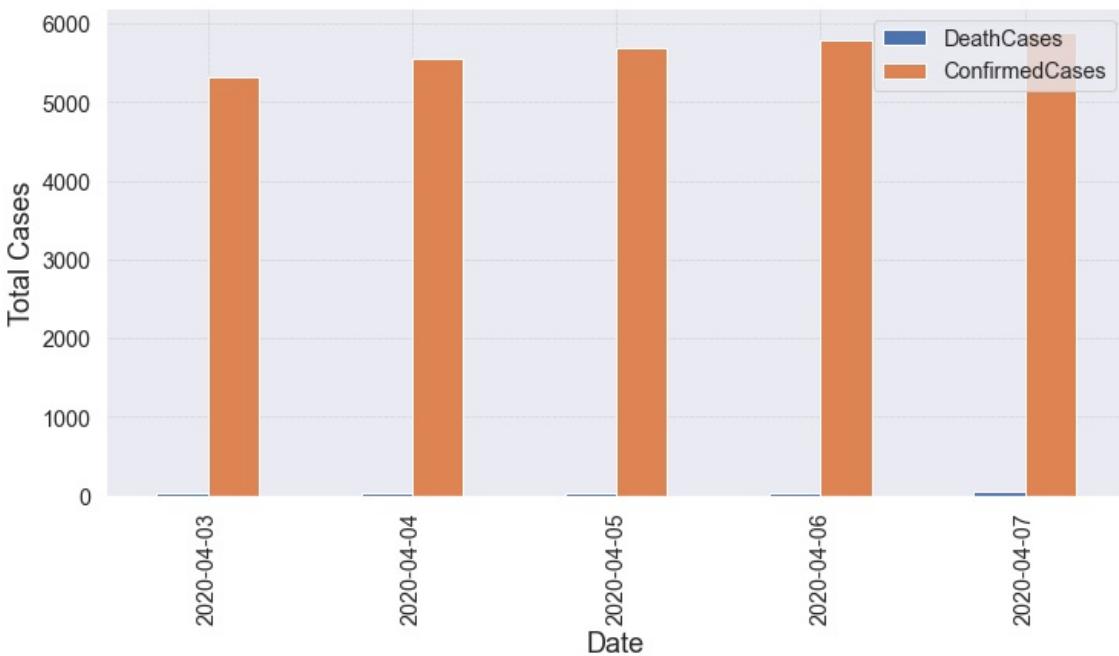
## Daily Cases and Death Count of COVID-19 from January - April in Argentina



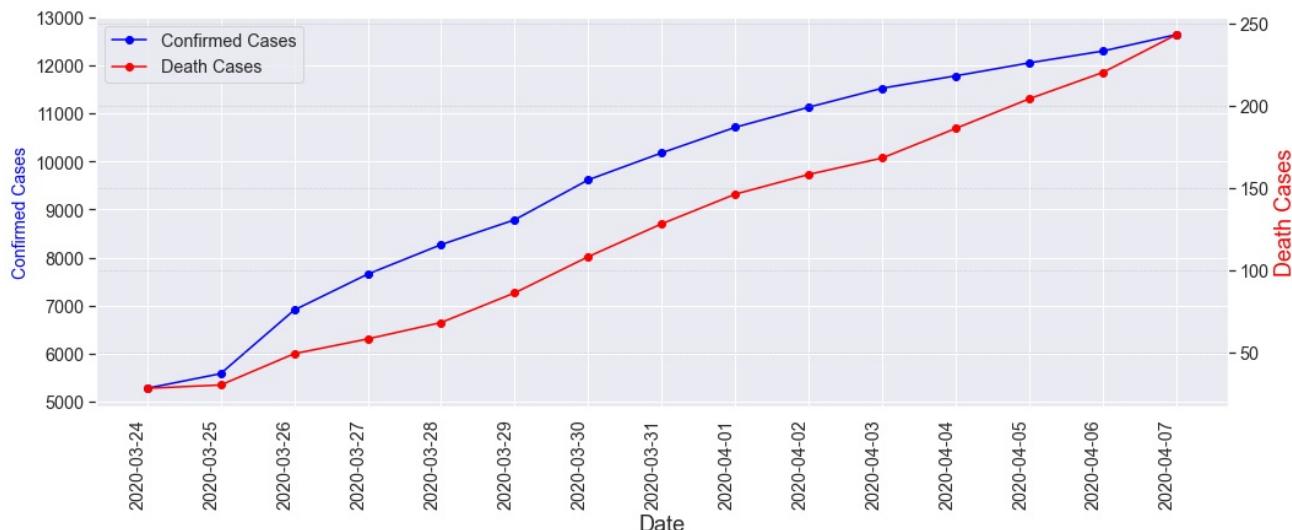
## Australia's Growth Curve Alignment with COVID-19 Death Toll of At Least 25 Cases



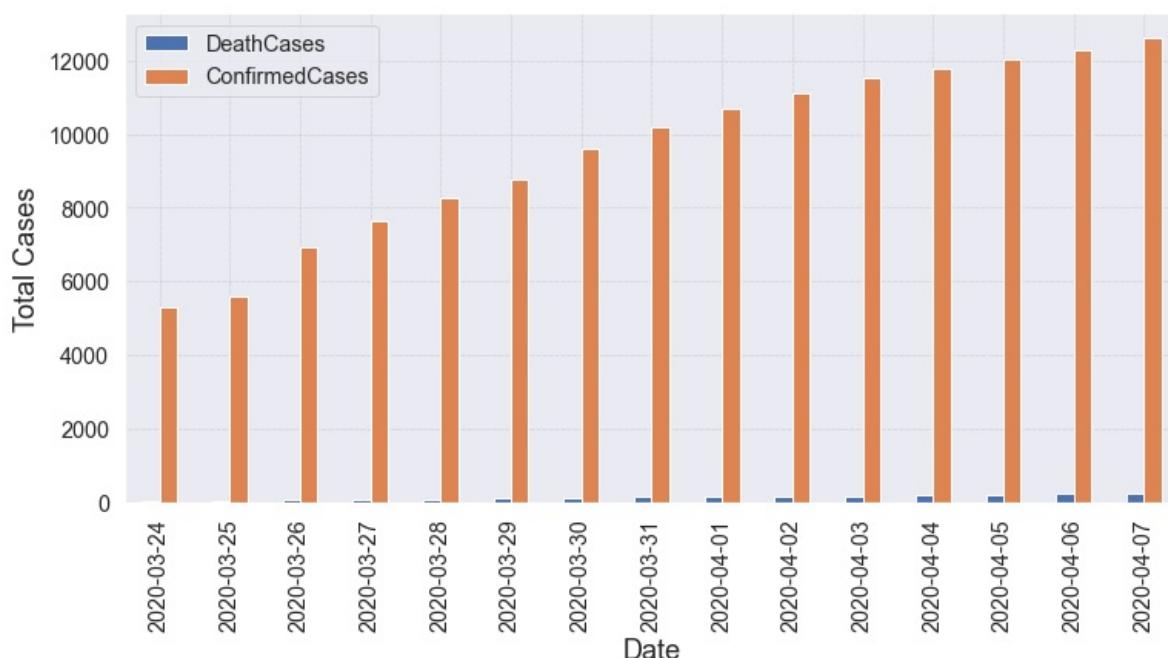
## Daily Cases and Death Count of COVID-19 from January - April in Australia



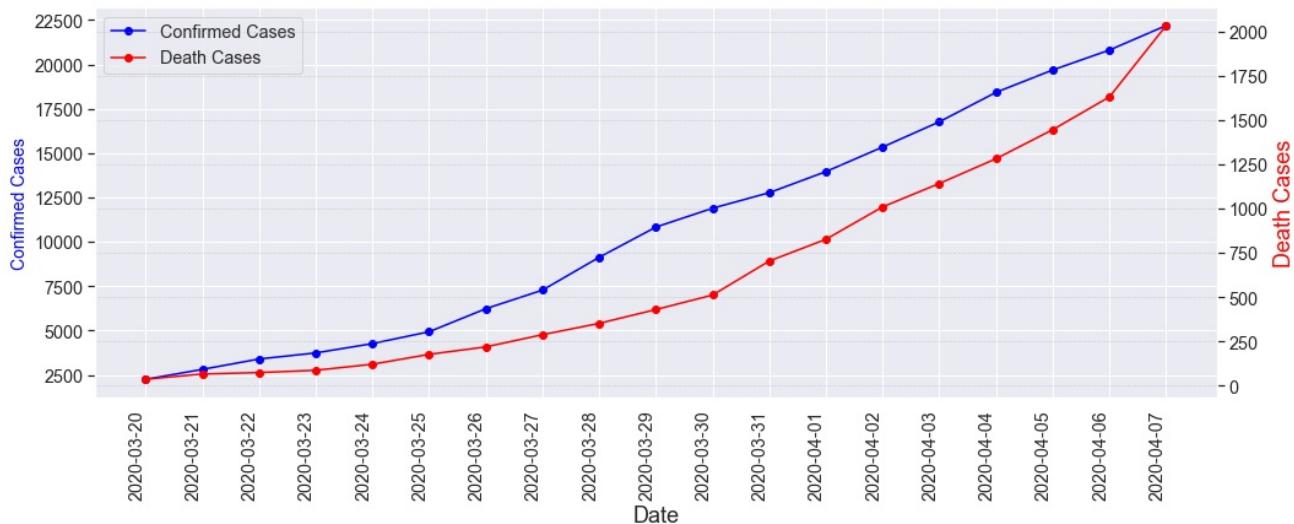
## Austria's Growth Curve Alignment with COVID-19 Death Toll of At Least 25 Cases



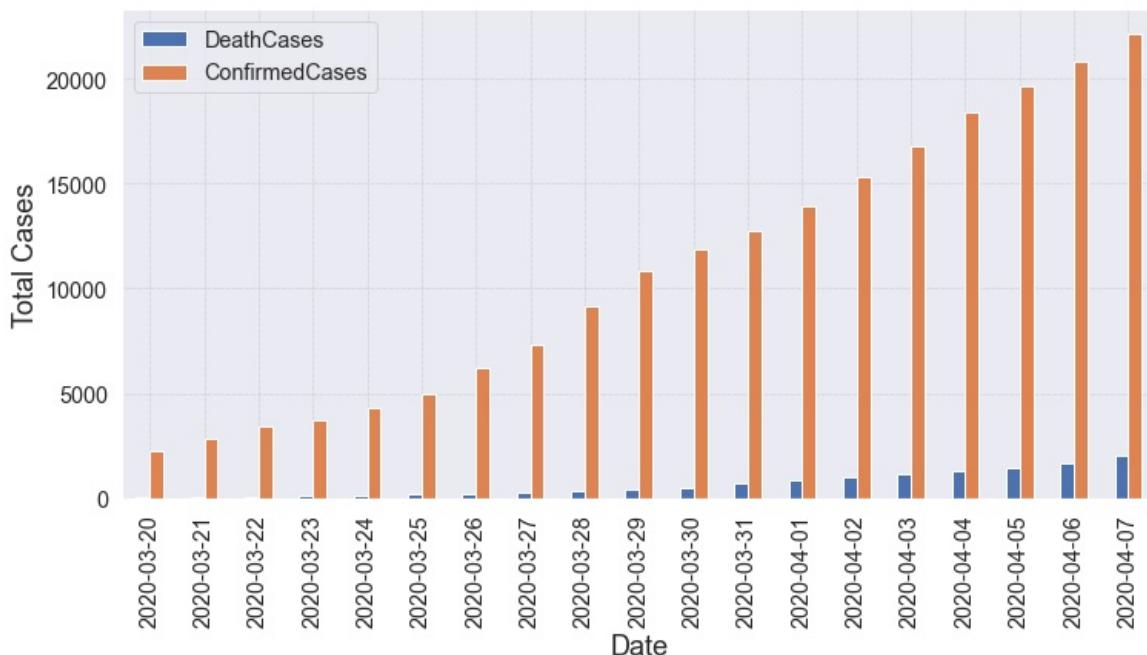
## Daily Cases and Death Count of COVID-19 from January - April in Austria



## Belgium's Growth Curve Alignment with COVID-19 Death Toll of At Least 25 Cases



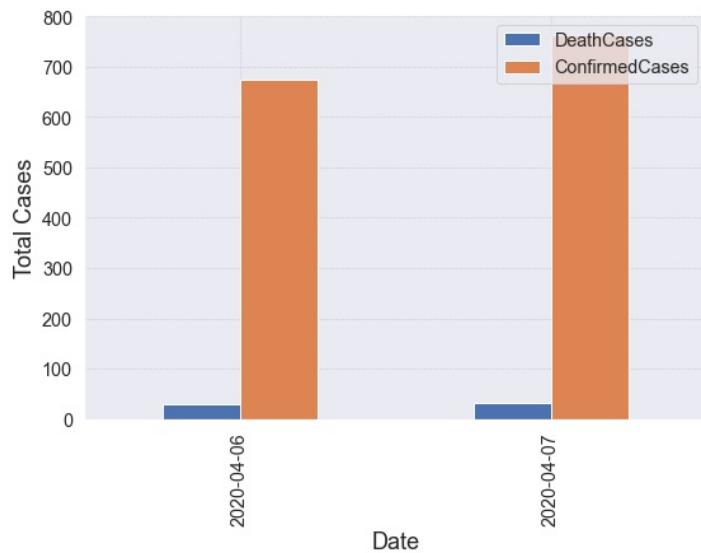
## Daily Cases and Death Count of COVID-19 from January - April in Belgium



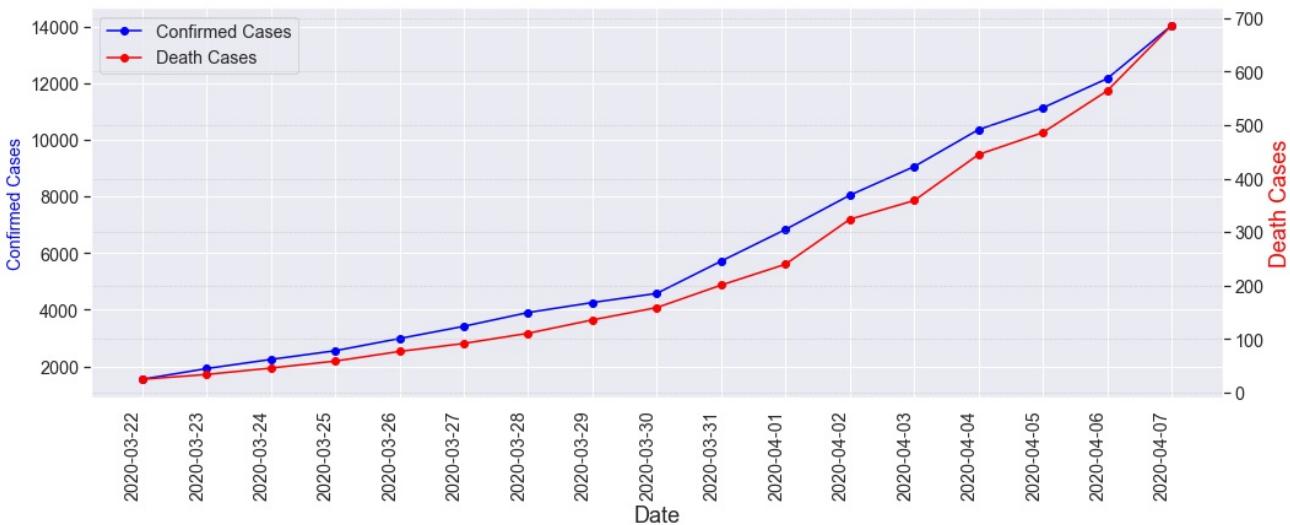
## Bosnia and Herzegovina's Growth Curve Alignment with COVID-19 Death Toll of At Least 25 Cases



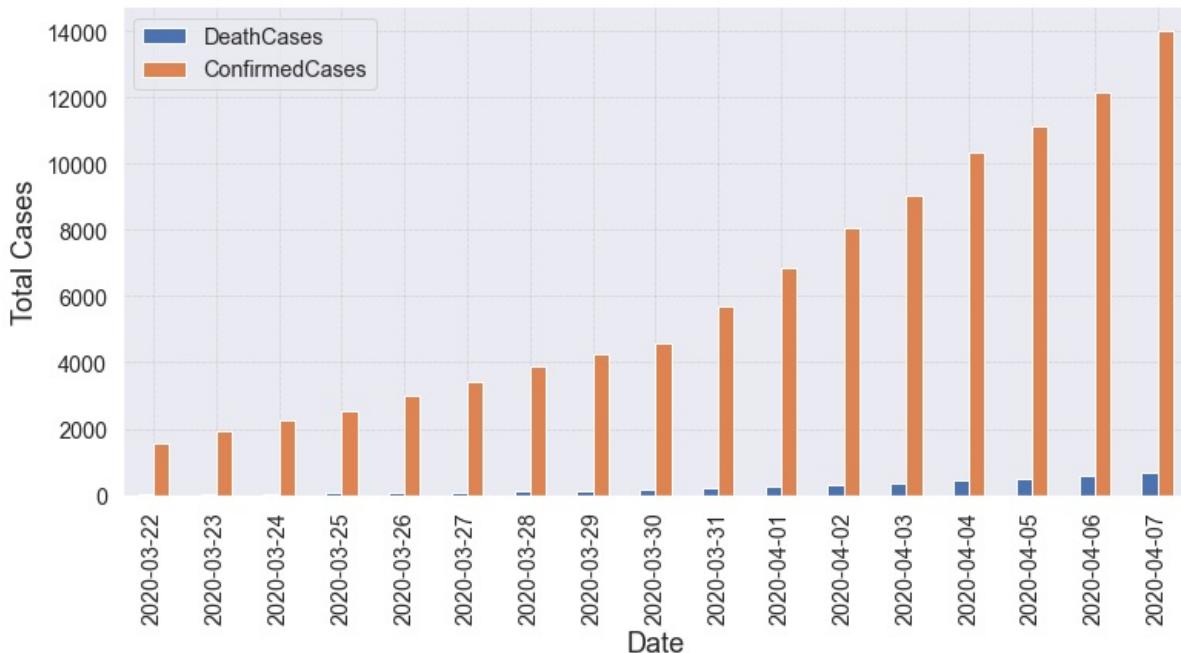
## Daily Cases and Death Count of COVID-19 from January - April in Bosnia and Herzegovina



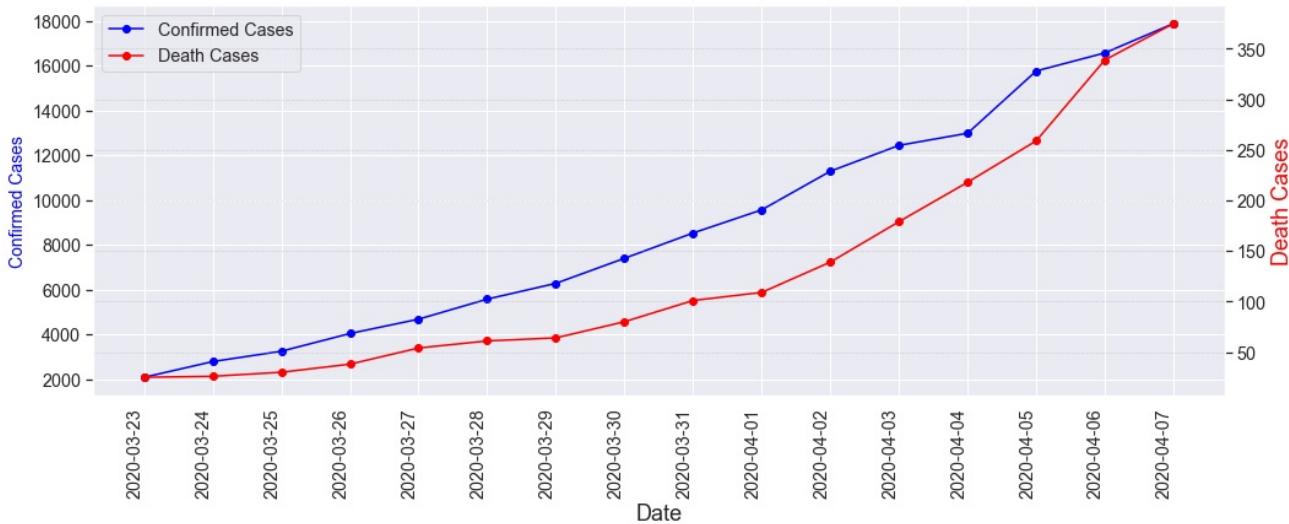
## Brazil's Growth Curve Alignment with COVID-19 Death Toll of At Least 25 Cases



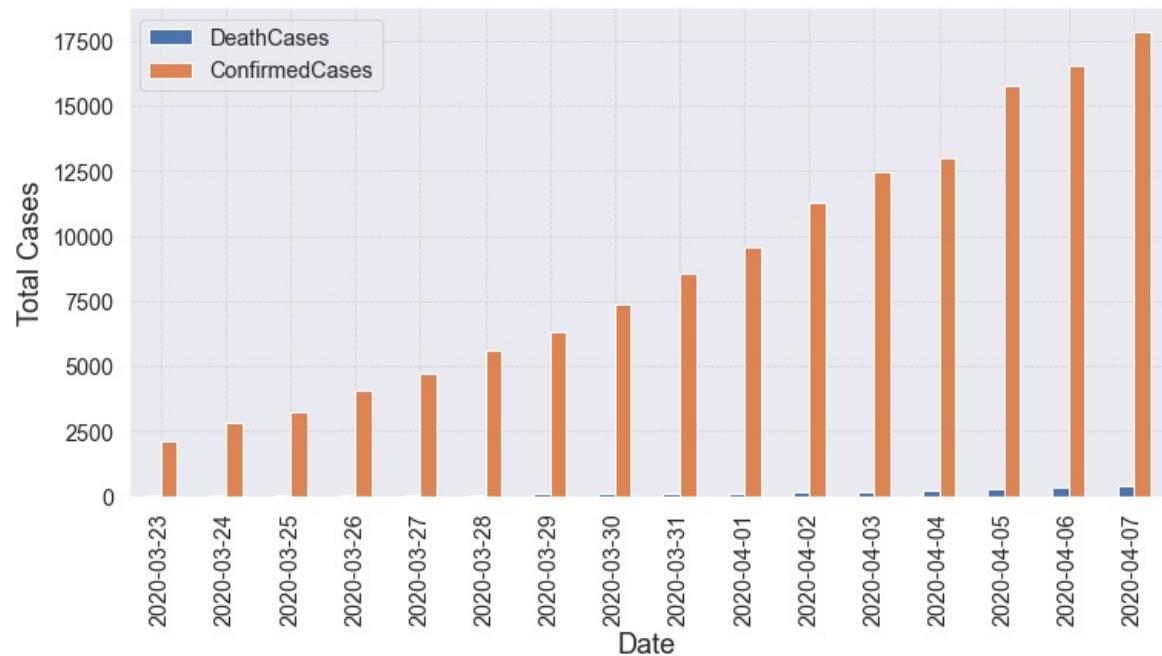
## Daily Cases and Death Count of COVID-19 from January - April in Brazil



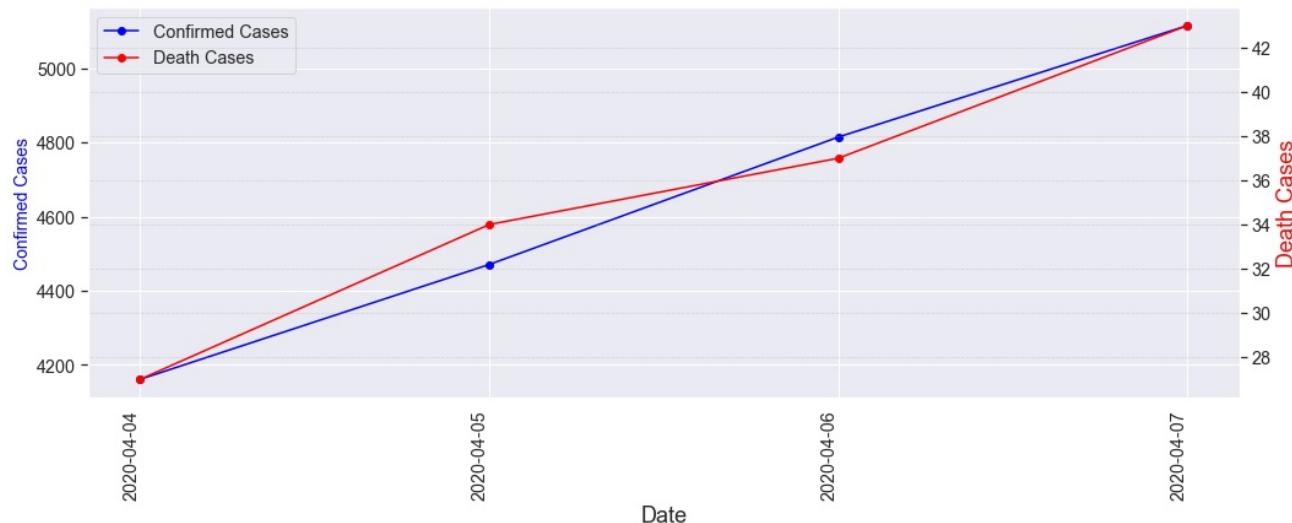
## Canada's Growth Curve Alignment with COVID-19 Death Toll of At Least 25 Cases



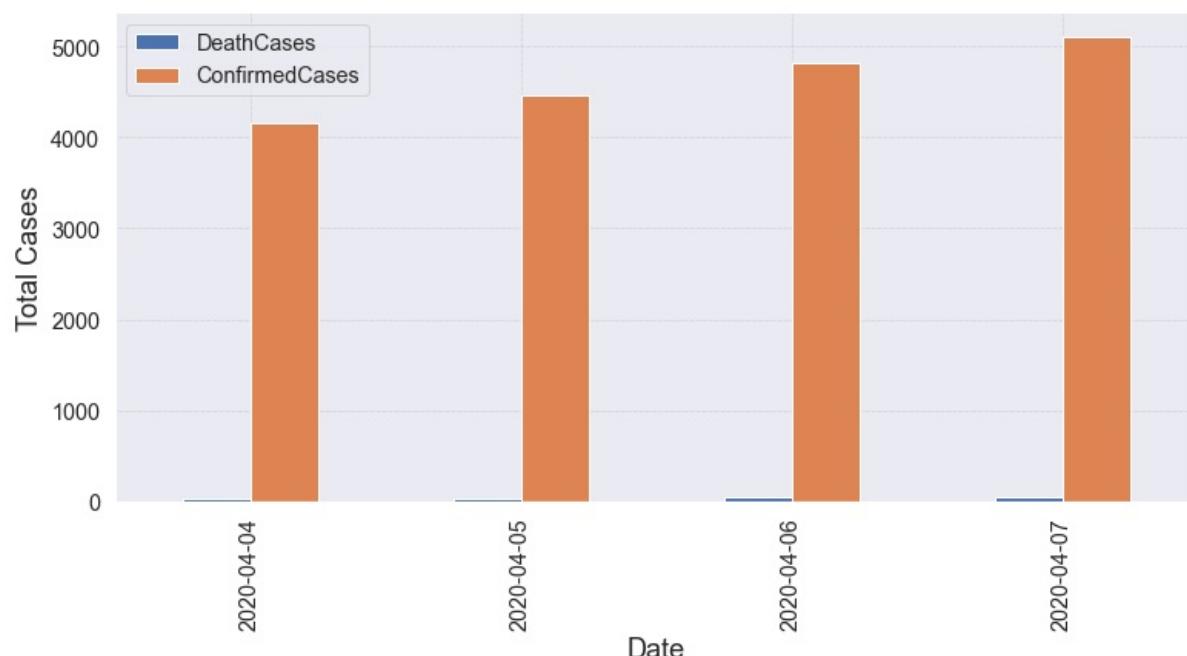
## Daily Cases and Death Count of COVID-19 from January - April in Canada



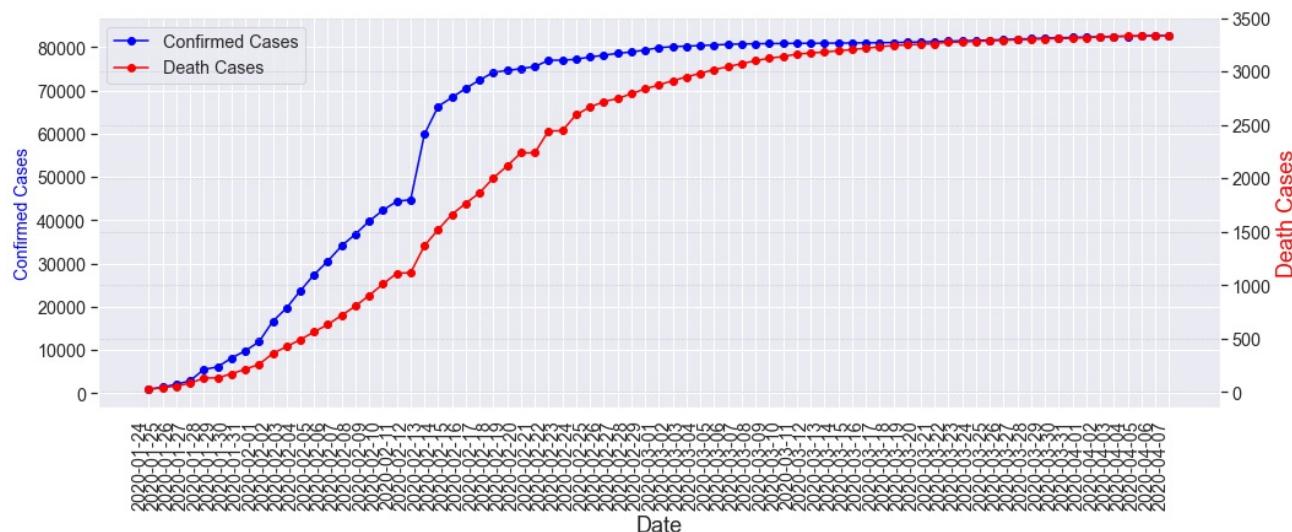
### Chile's Growth Curve Alignment with COVID-19 Death Toll of At Least 25 Cases



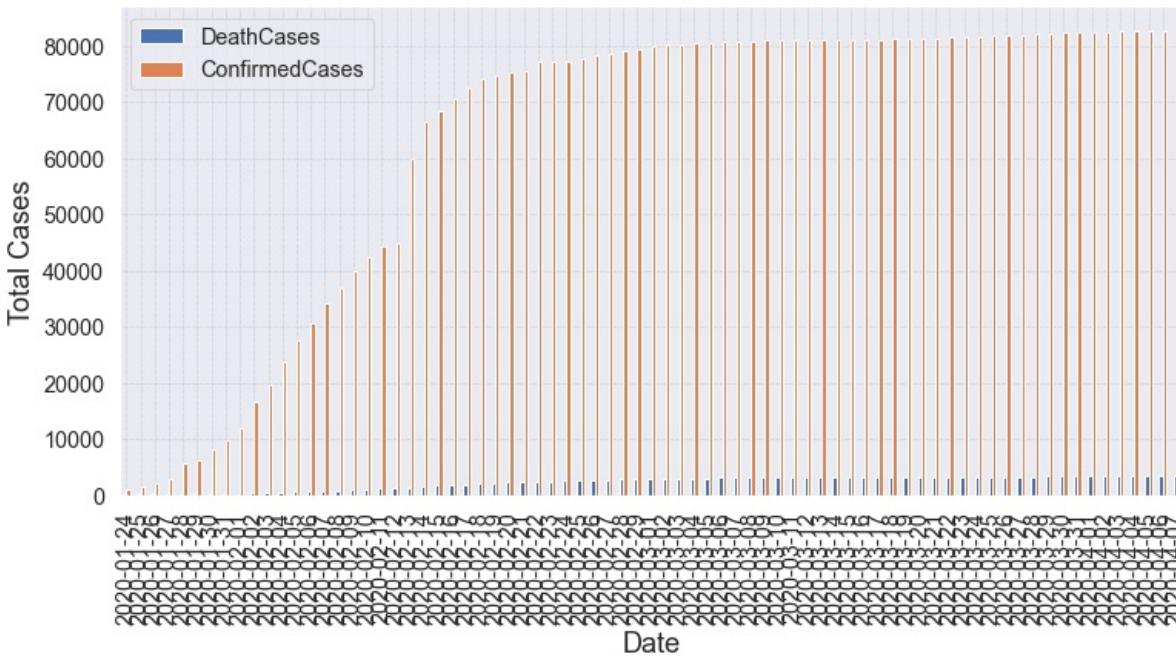
### Daily Cases and Death Count of COVID-19 from January - April in Chile



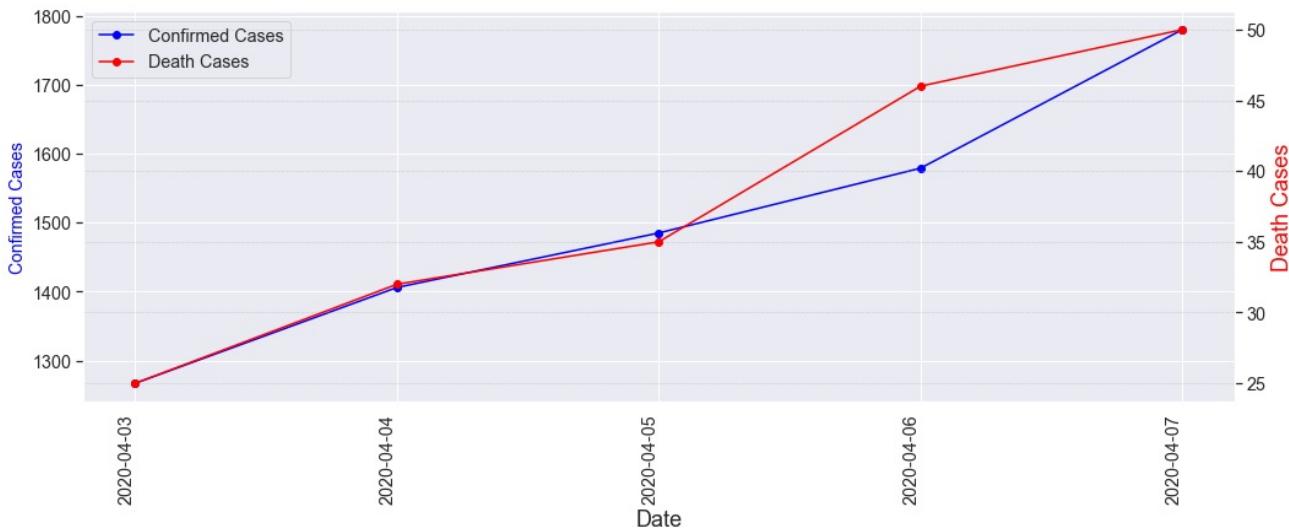
### China's Growth Curve Alignment with COVID-19 Death Toll of At Least 25 Cases



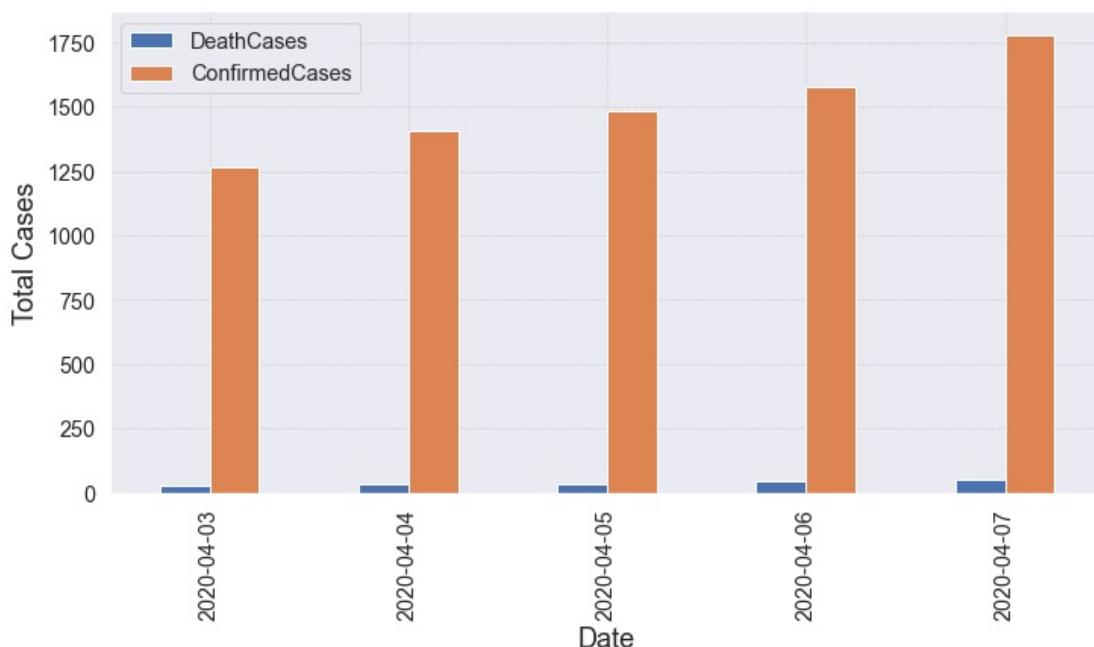
## Daily Cases and Death Count of COVID-19 from January - April in China



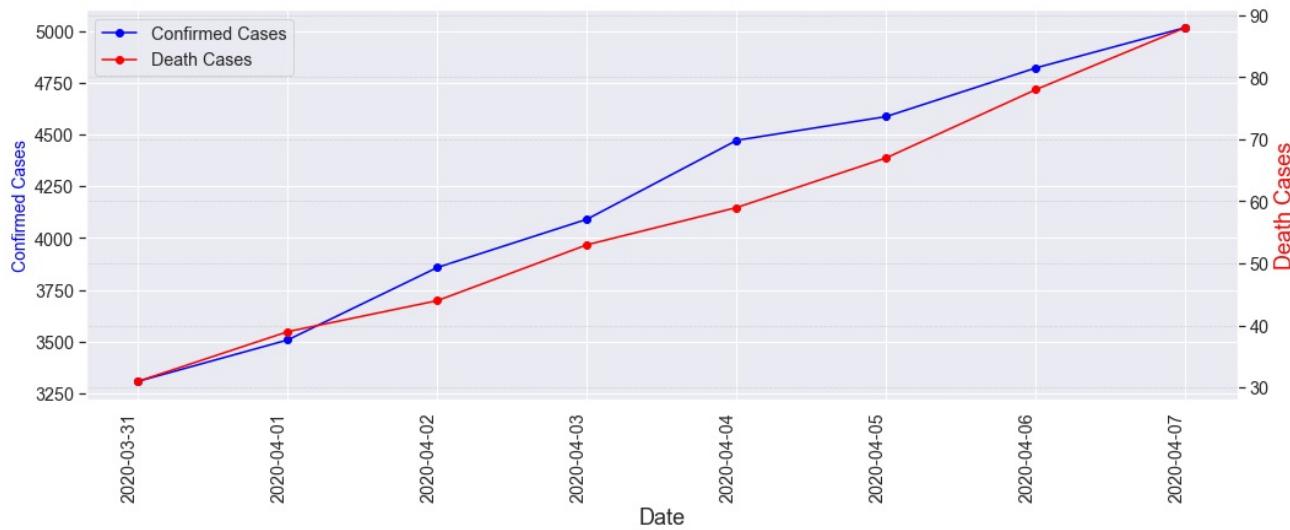
## Colombia's Growth Curve Alignment with COVID-19 Death Toll of At Least 25 Cases



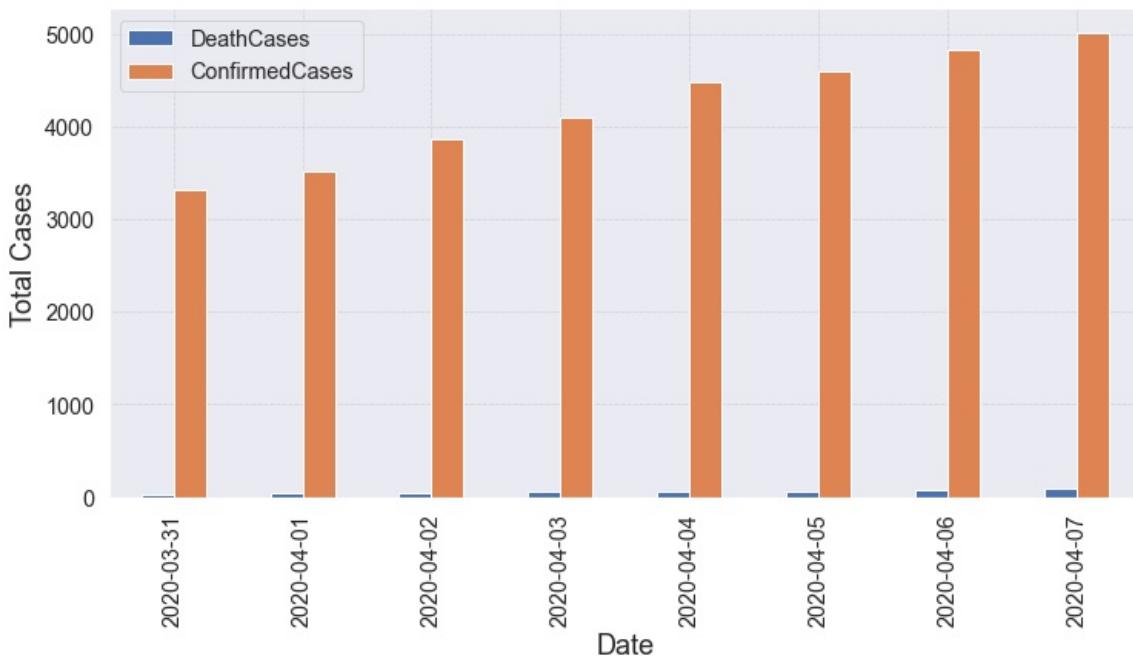
## Daily Cases and Death Count of COVID-19 from January - April in Colombia



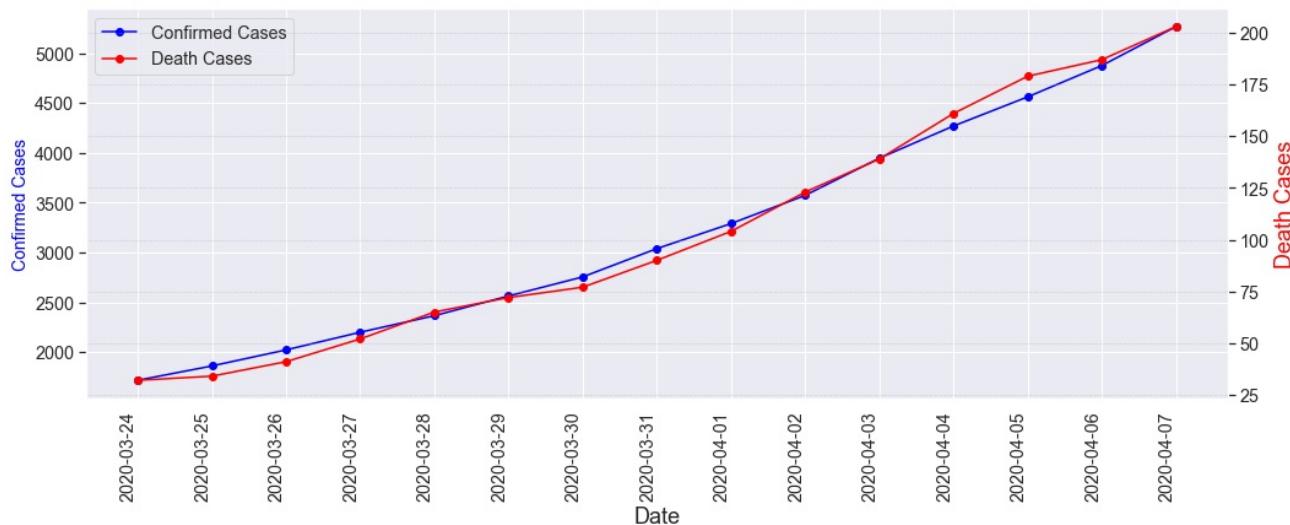
### Czechia's Growth Curve Alignment with COVID-19 Death Toll of At Least 25 Cases



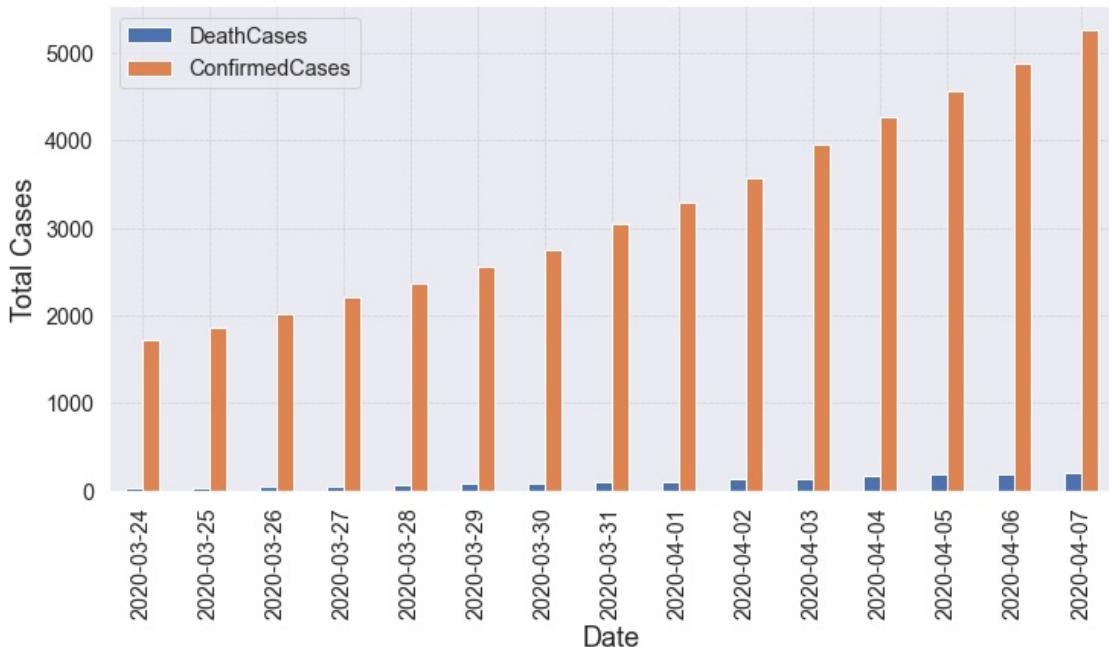
### Daily Cases and Death Count of COVID-19 from January - April in Czechia



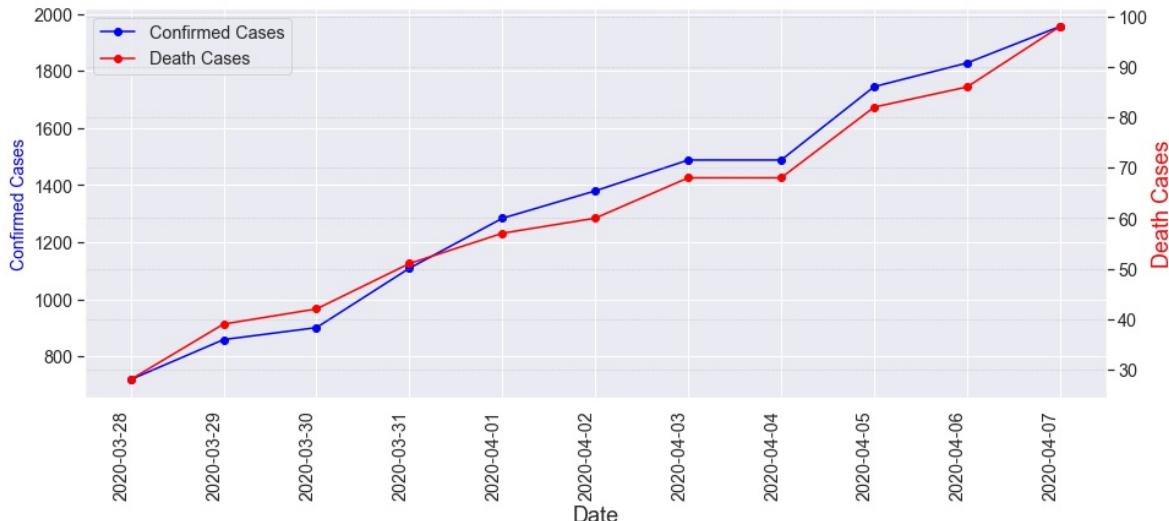
### Denmark's Growth Curve Alignment with COVID-19 Death Toll of At Least 25 Cases



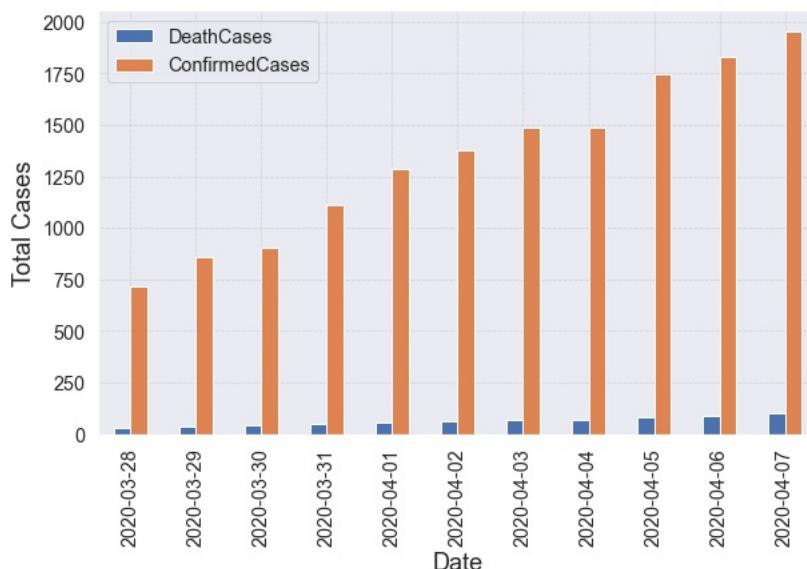
## Daily Cases and Death Count of COVID-19 from January - April in Denmark



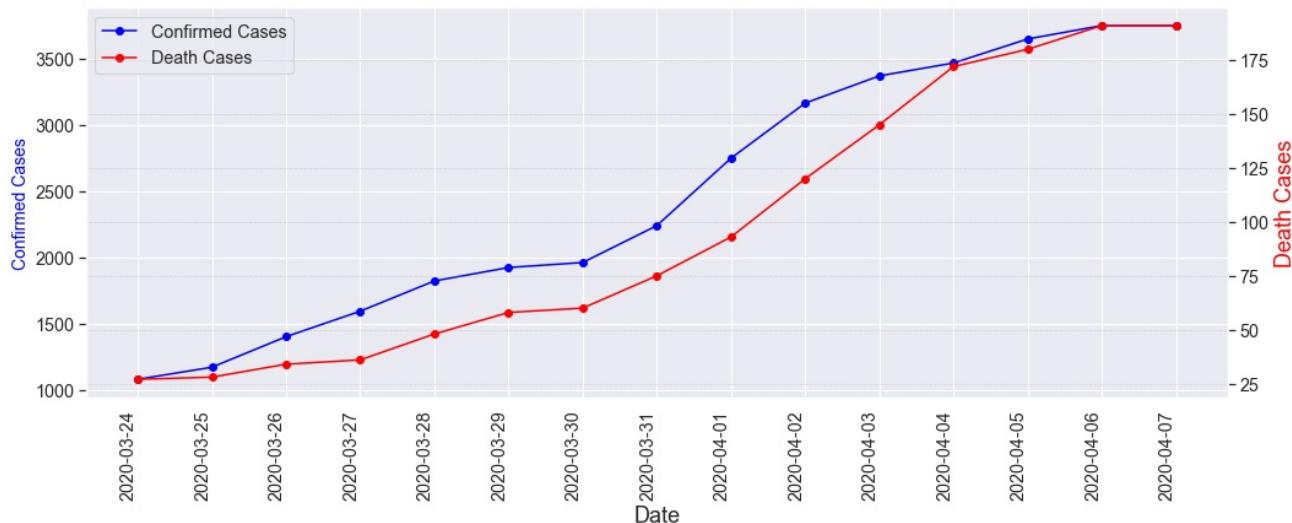
## Dominican Republic's Growth Curve Alignment with COVID-19 Death Toll of At Least 25 Cases



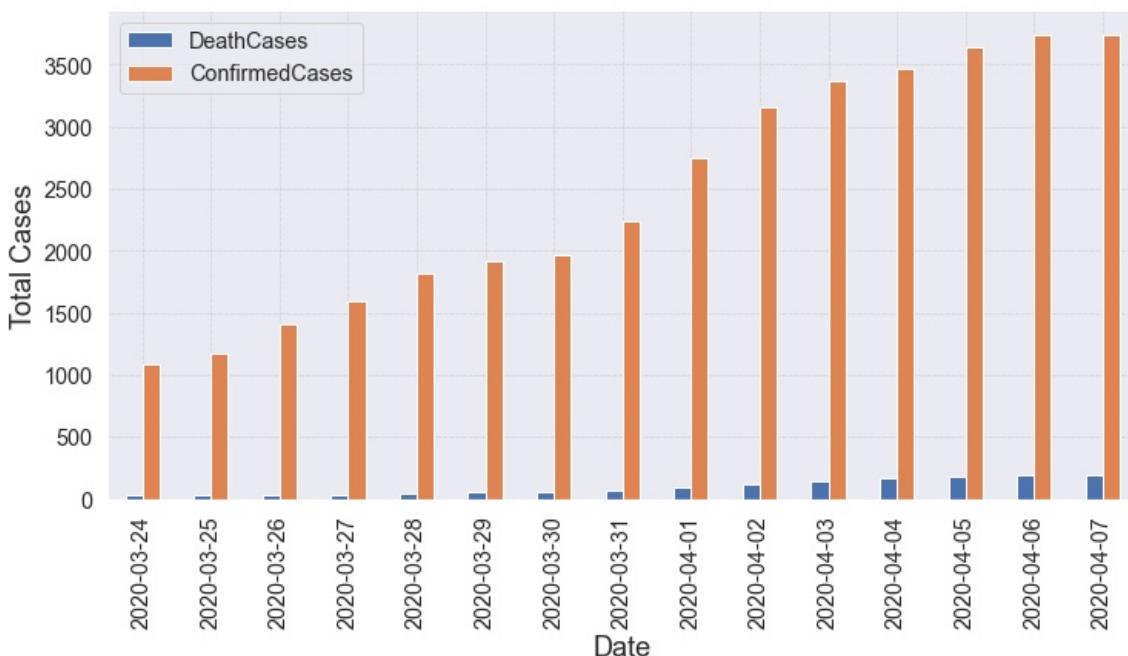
## Daily Cases and Death Count of COVID-19 from January - April in Dominican Republic



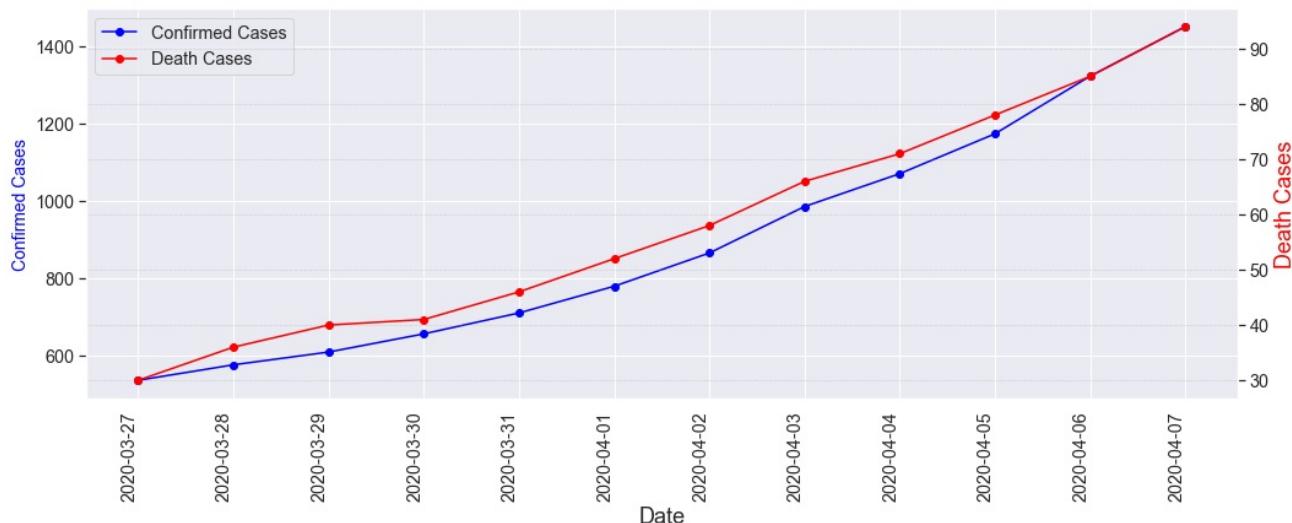
## Ecuador's Growth Curve Alignment with COVID-19 Death Toll of At Least 25 Cases



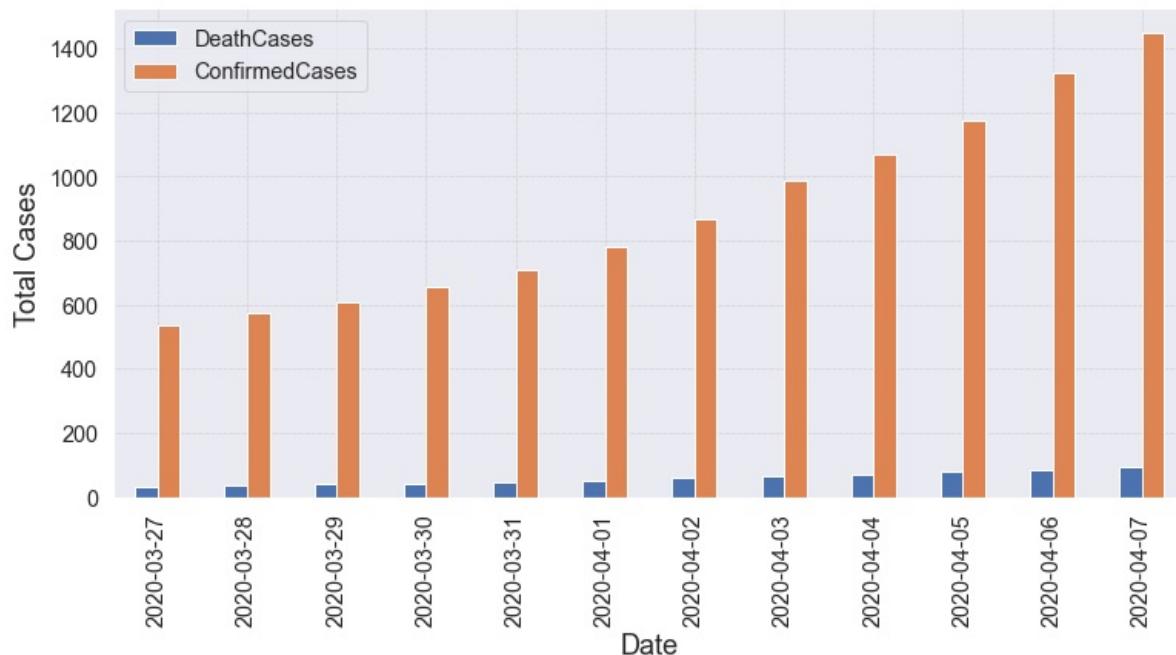
## Daily Cases and Death Count of COVID-19 from January - April in Ecuador



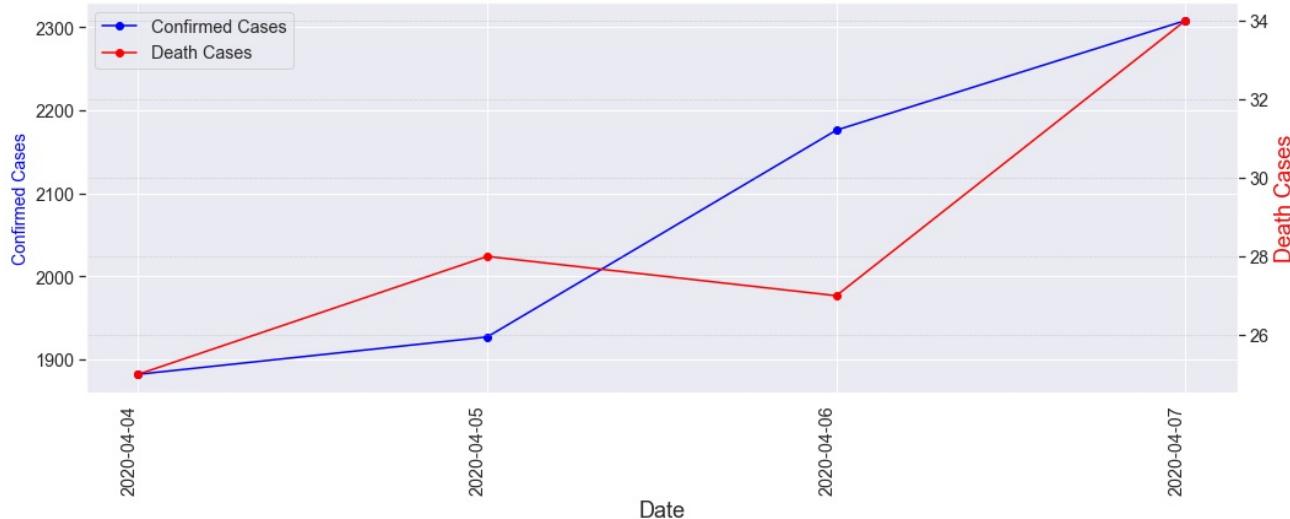
## Egypt's Growth Curve Alignment with COVID-19 Death Toll of At Least 25 Cases



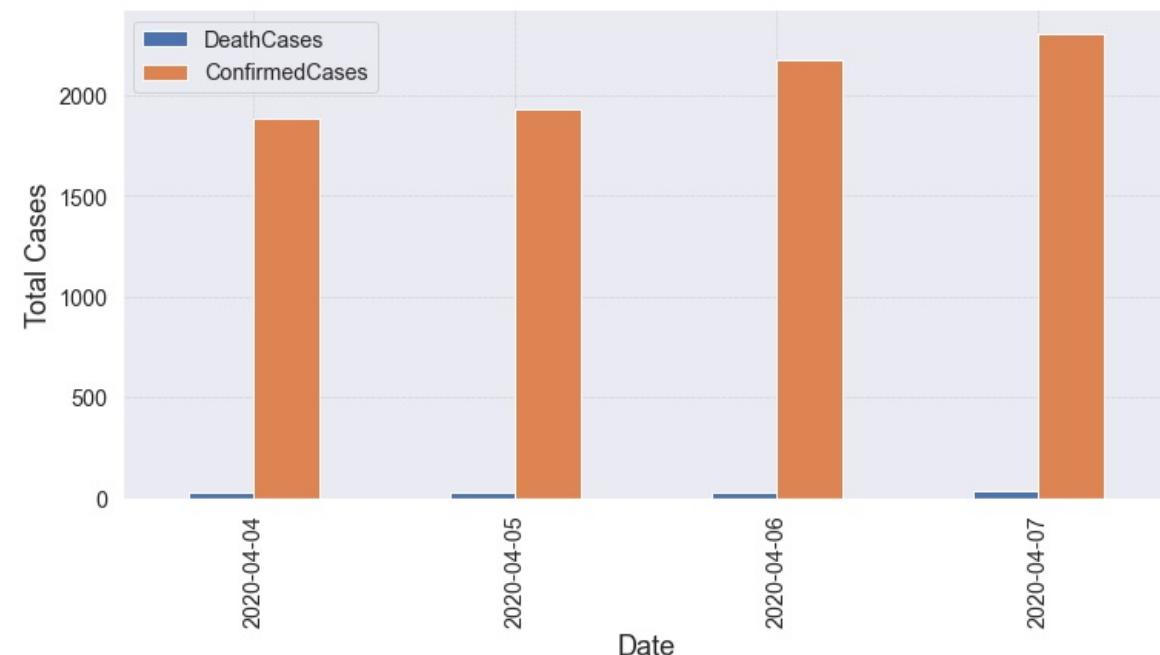
## Daily Cases and Death Count of COVID-19 from January - April in Egypt



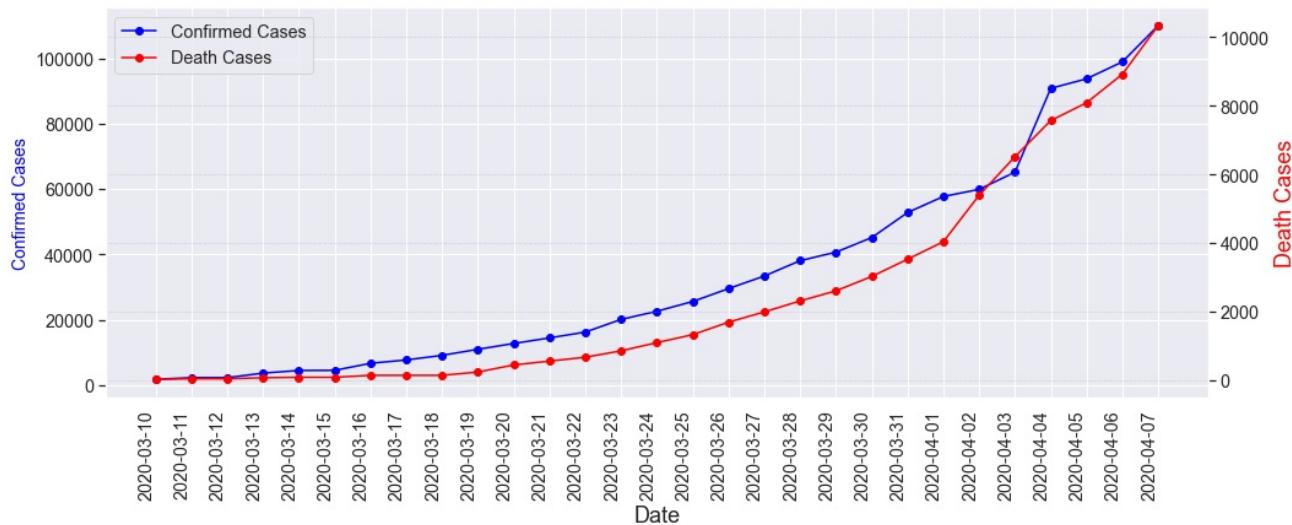
## Finland's Growth Curve Alignment with COVID-19 Death Toll of At Least 25 Cases



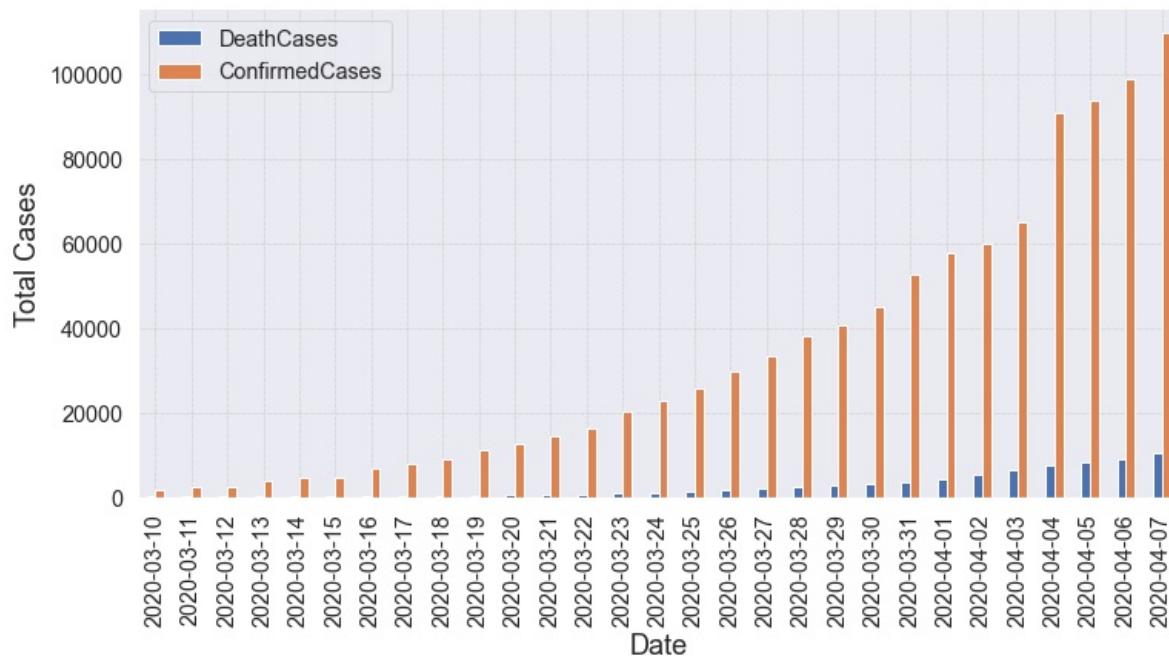
## Daily Cases and Death Count of COVID-19 from January - April in Finland



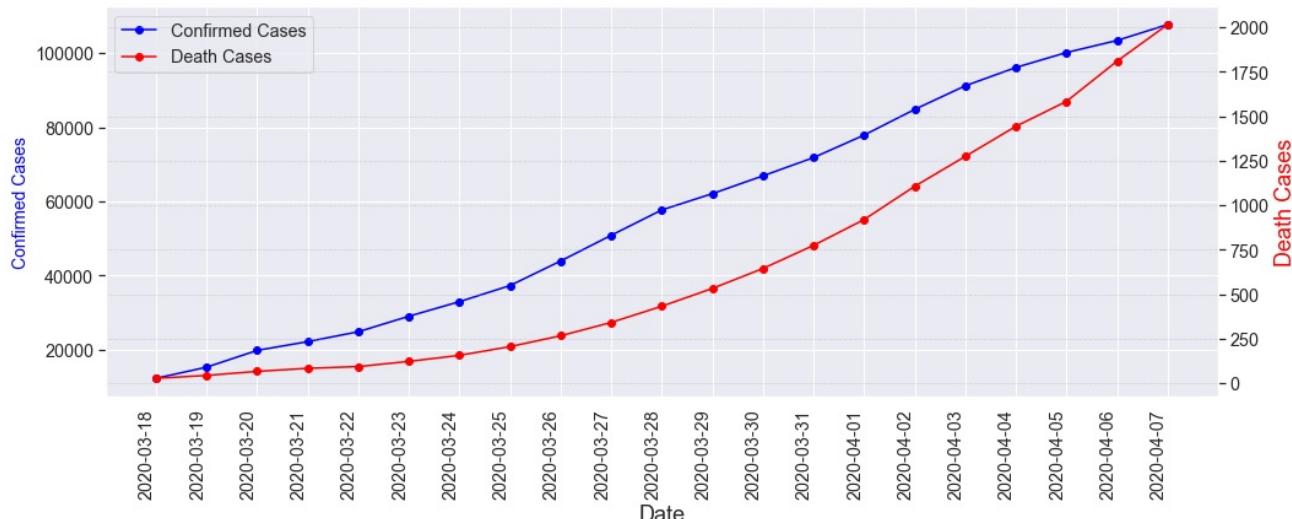
## France's Growth Curve Alignment with COVID-19 Death Toll of At Least 25 Cases



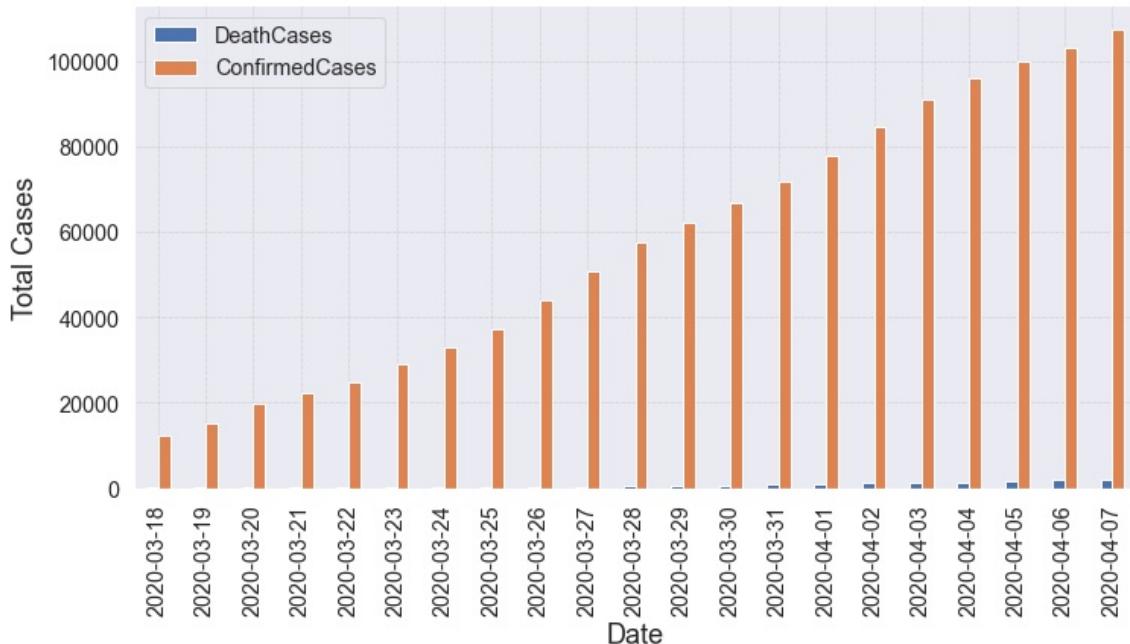
## Daily Cases and Death Count of COVID-19 from January - April in France



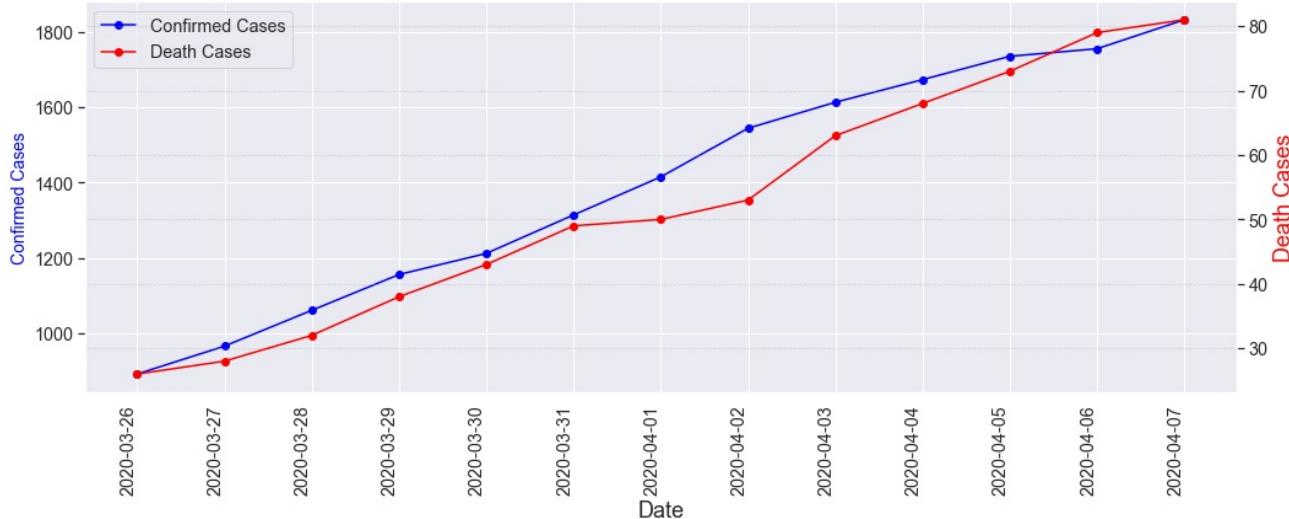
## Germany's Growth Curve Alignment with COVID-19 Death Toll of At Least 25 Cases



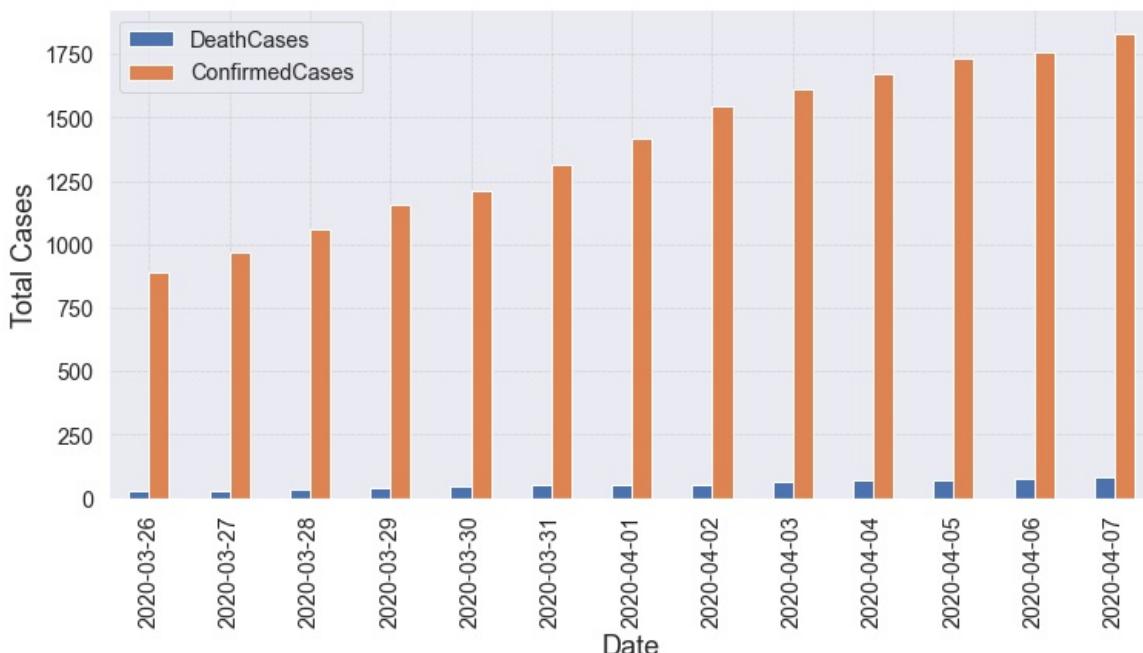
## Daily Cases and Death Count of COVID-19 from January - April in Germany



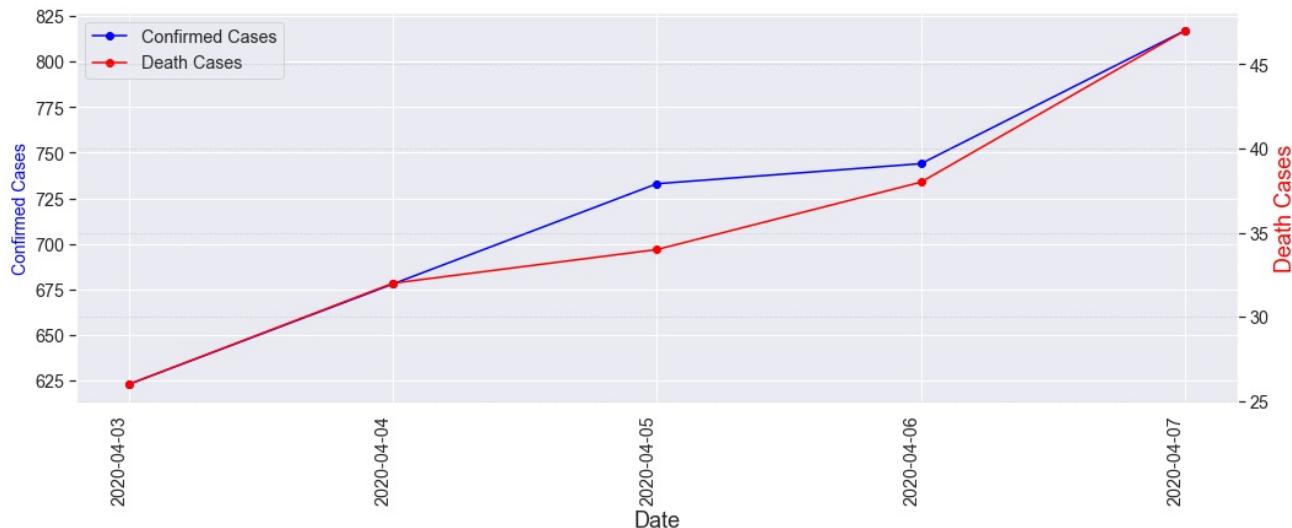
## Greece's Growth Curve Alignment with COVID-19 Death Toll of At Least 25 Cases



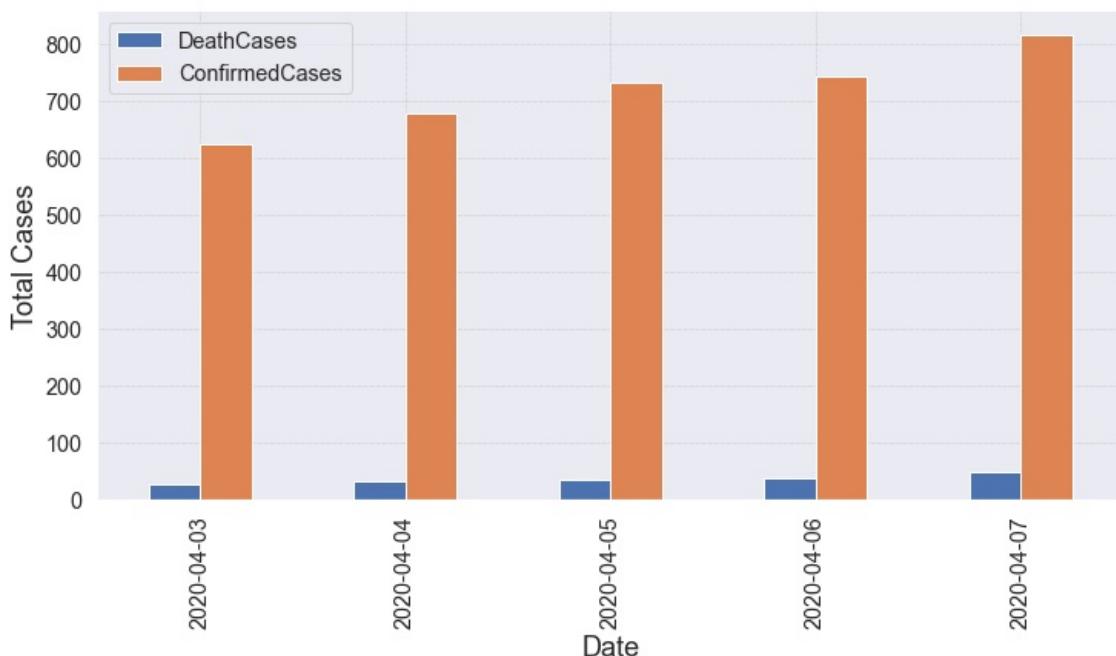
## Daily Cases and Death Count of COVID-19 from January - April in Greece



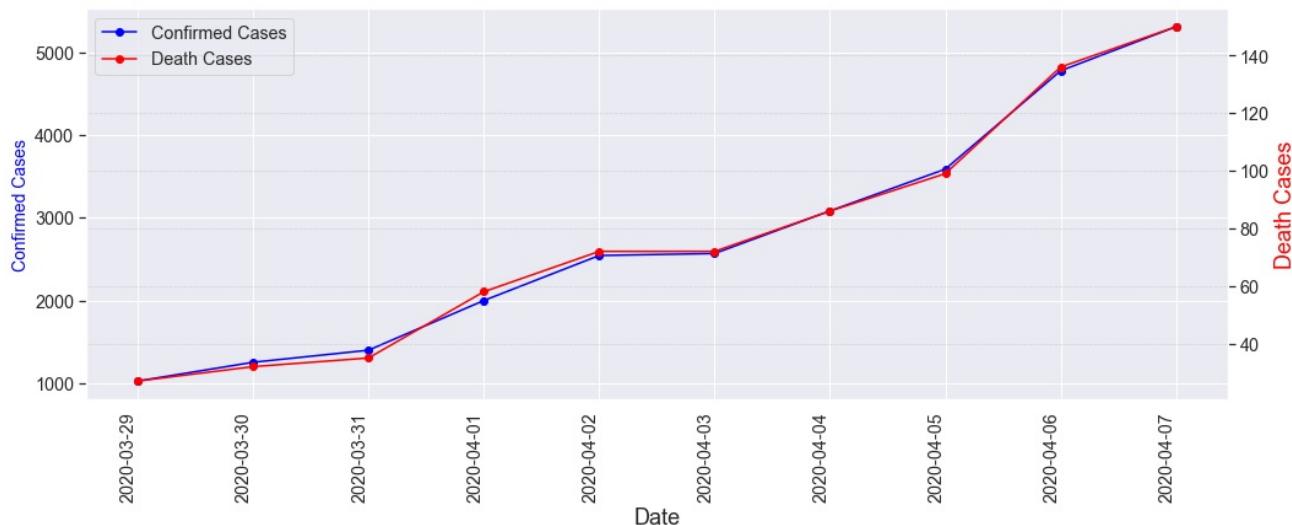
### Hungary's Growth Curve Alignment with COVID-19 Death Toll of At Least 25 Cases



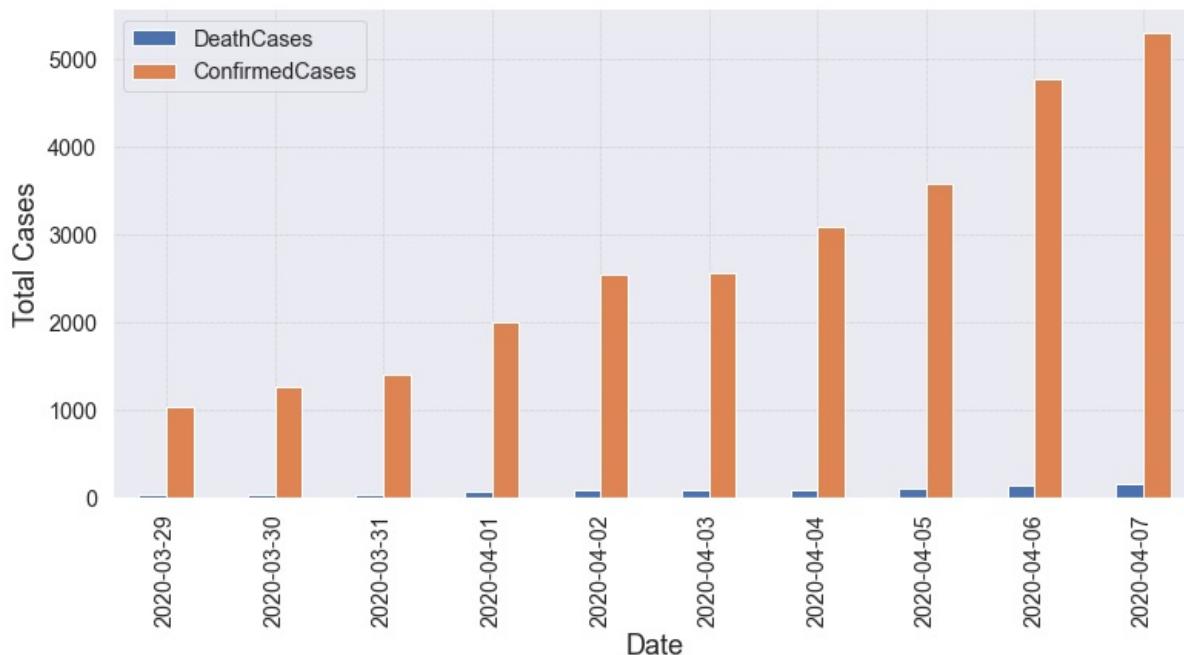
### Daily Cases and Death Count of COVID-19 from January - April in Hungary



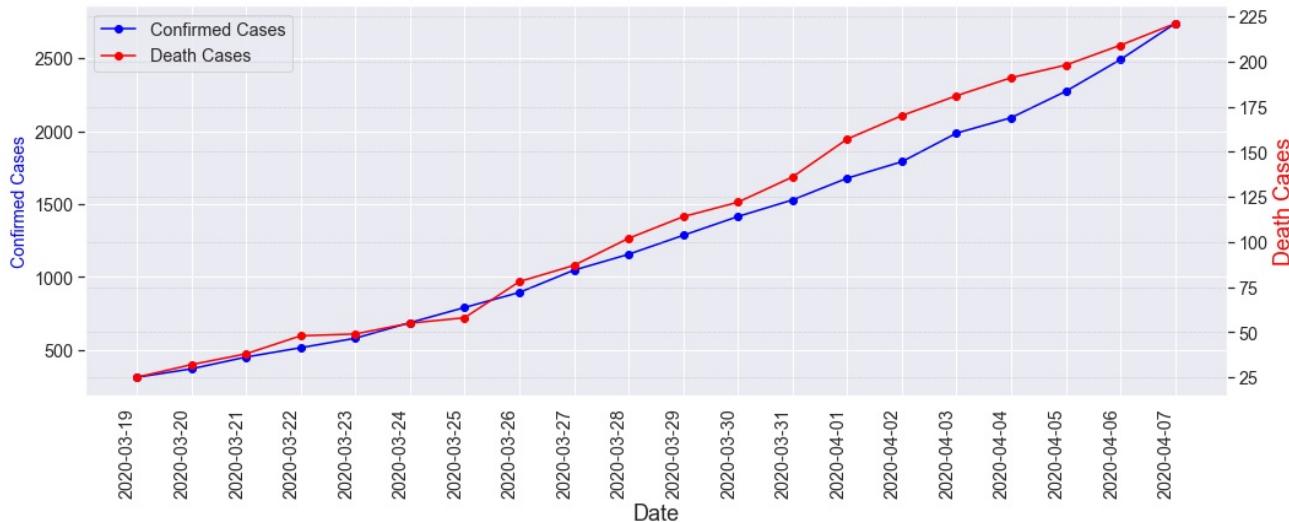
### India's Growth Curve Alignment with COVID-19 Death Toll of At Least 25 Cases



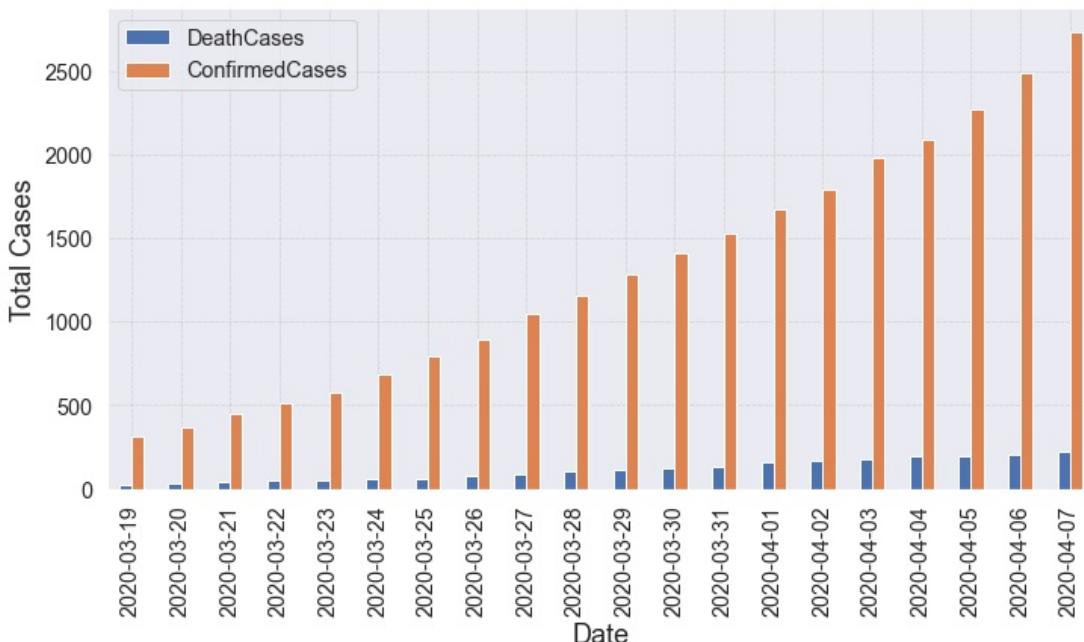
## Daily Cases and Death Count of COVID-19 from January - April in India



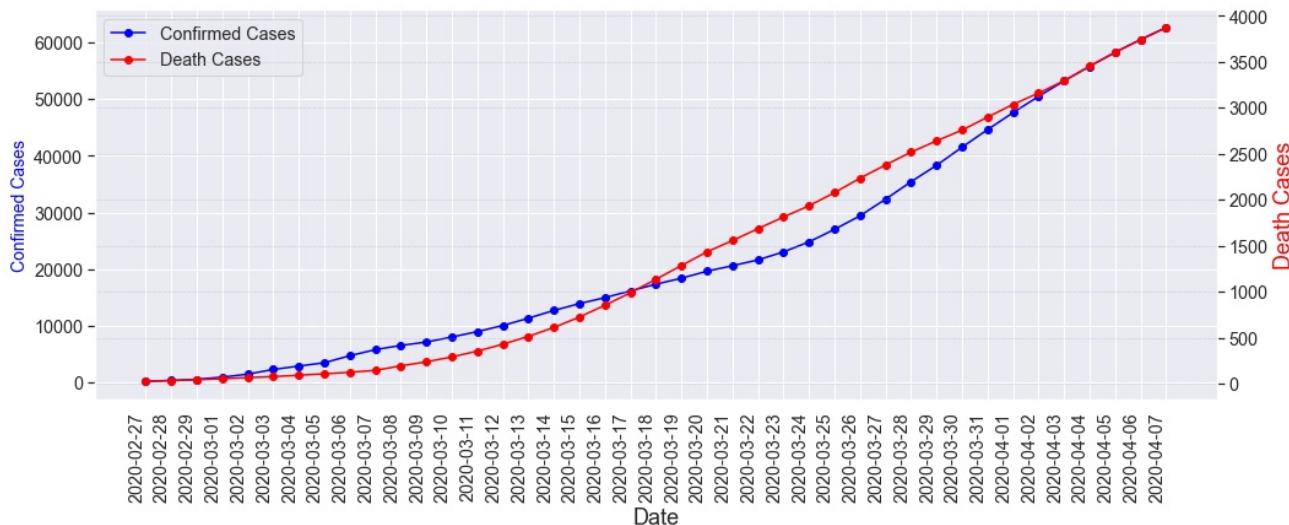
## Indonesia's Growth Curve Alignment with COVID-19 Death Toll of At Least 25 Cases



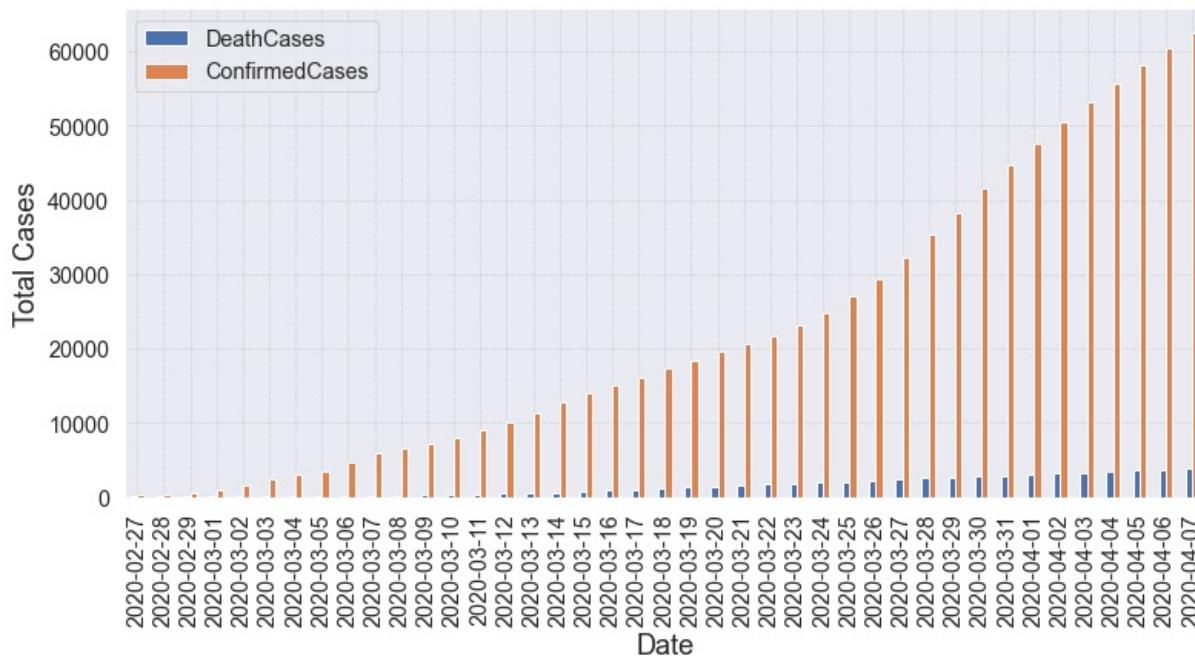
## Daily Cases and Death Count of COVID-19 from January - April in Indonesia



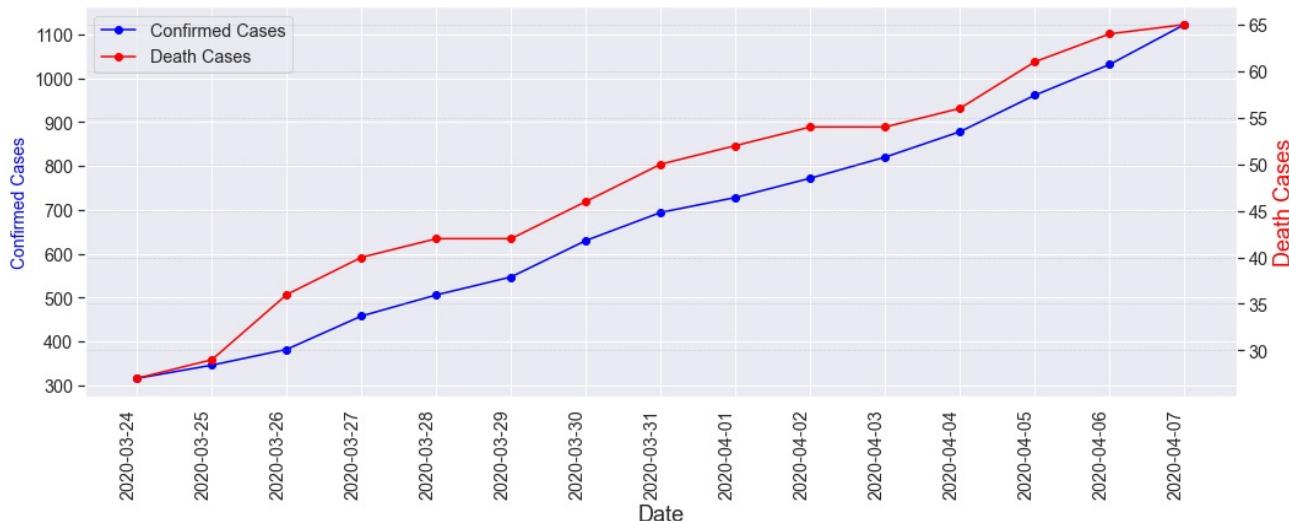
### Iran's Growth Curve Alignment with COVID-19 Death Toll of At Least 25 Cases



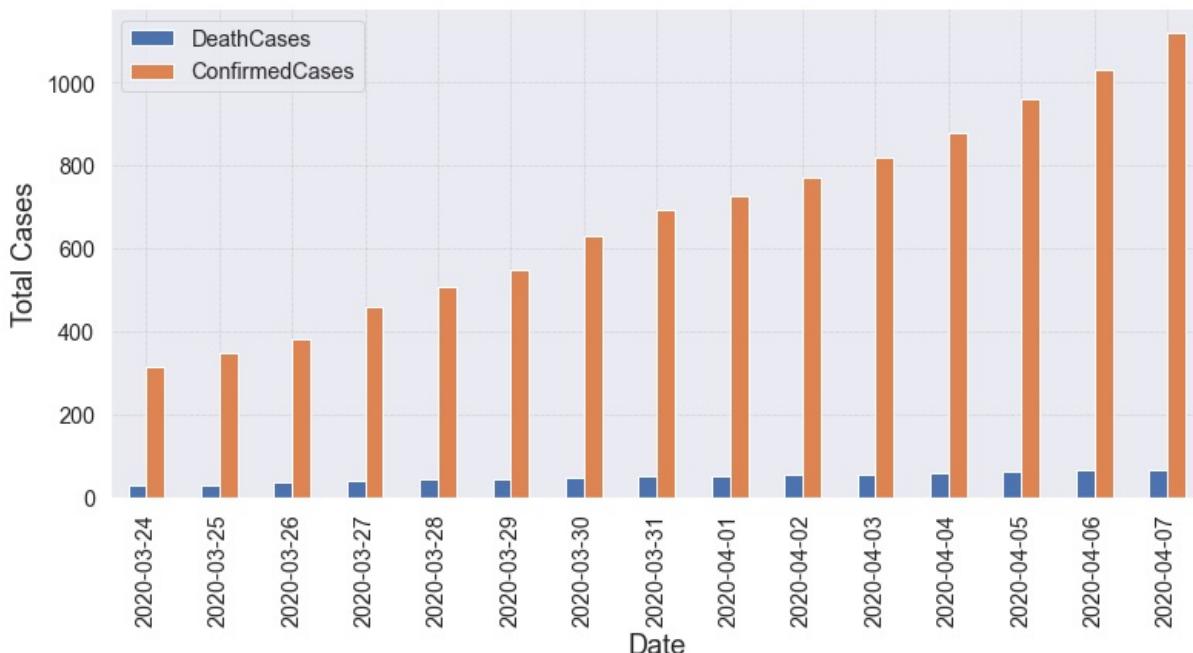
### Daily Cases and Death Count of COVID-19 from January - April in Iran



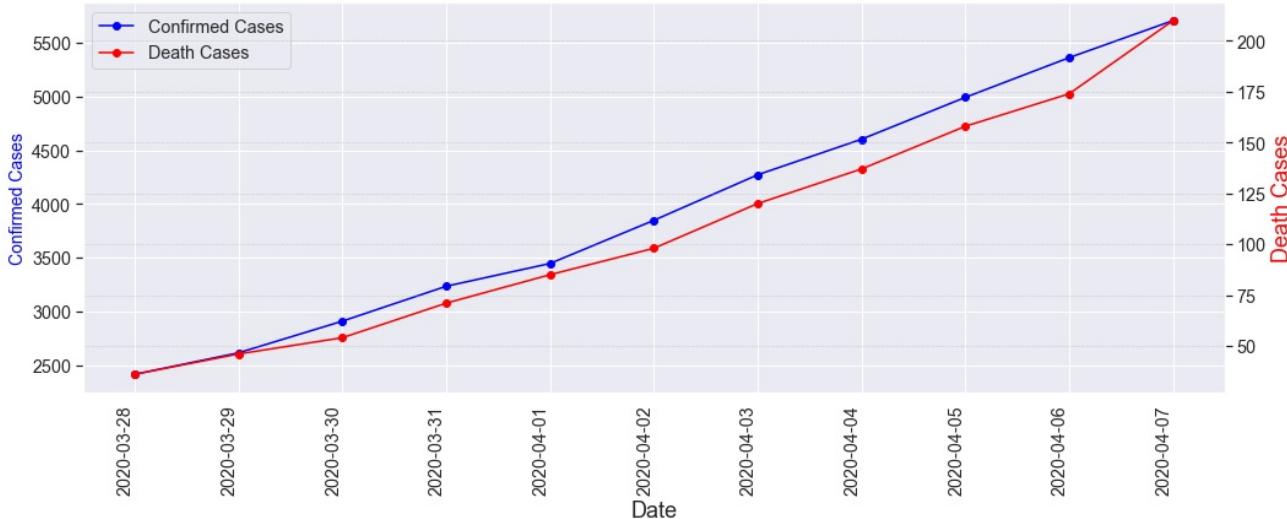
### Iraq's Growth Curve Alignment with COVID-19 Death Toll of At Least 25 Cases



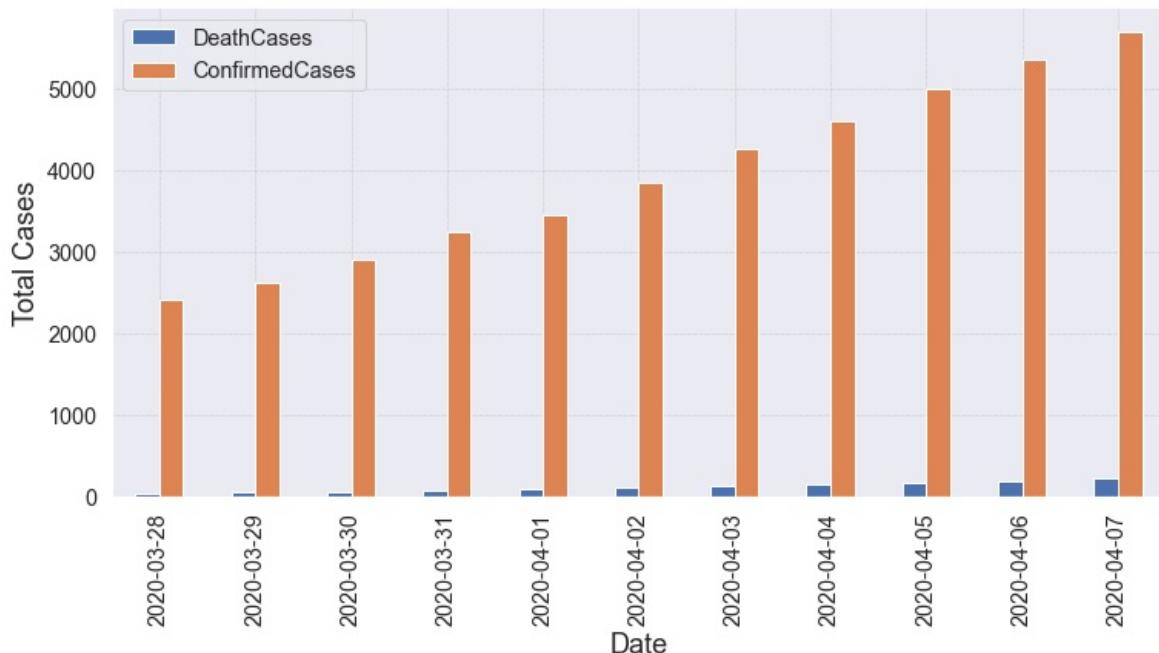
## Daily Cases and Death Count of COVID-19 from January - April in Iraq



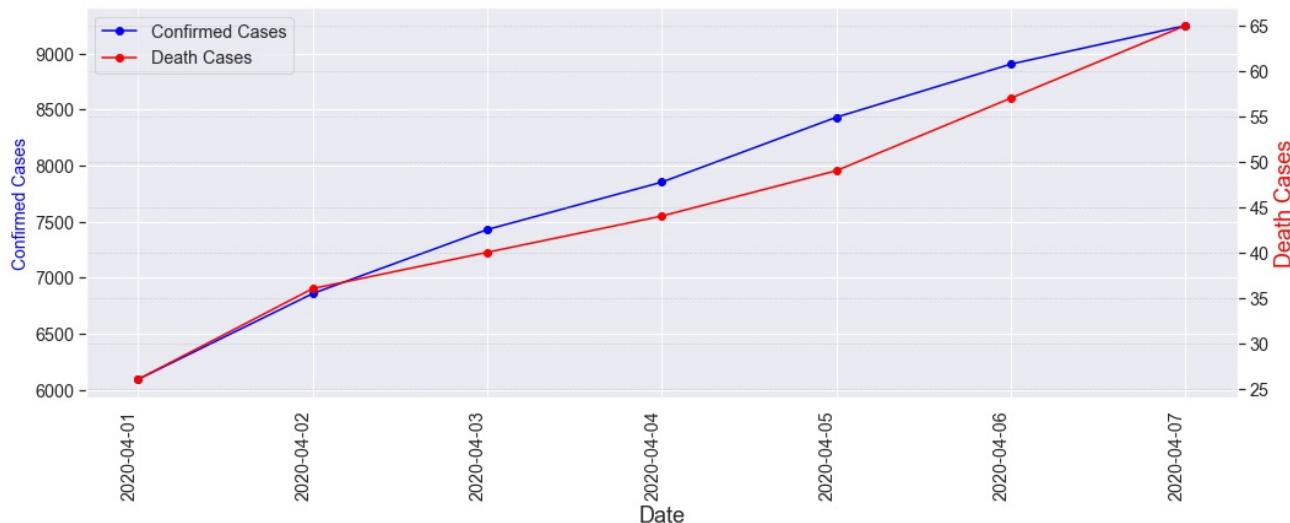
## Ireland's Growth Curve Alignment with COVID-19 Death Toll of At Least 25 Cases



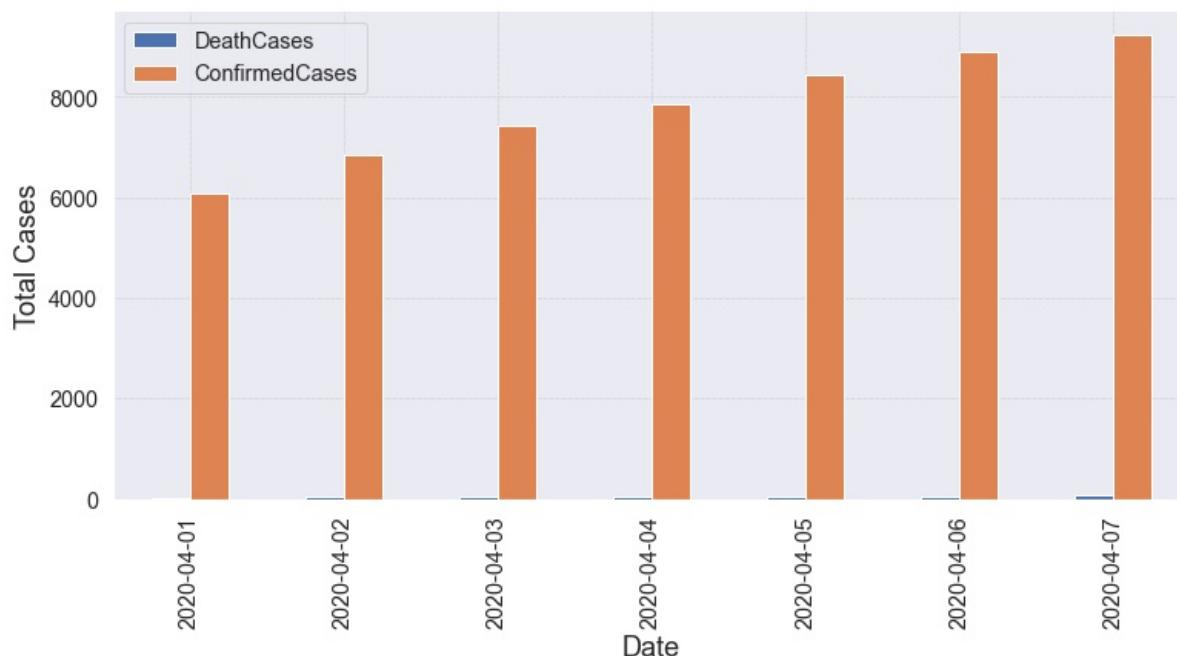
## Daily Cases and Death Count of COVID-19 from January - April in Ireland



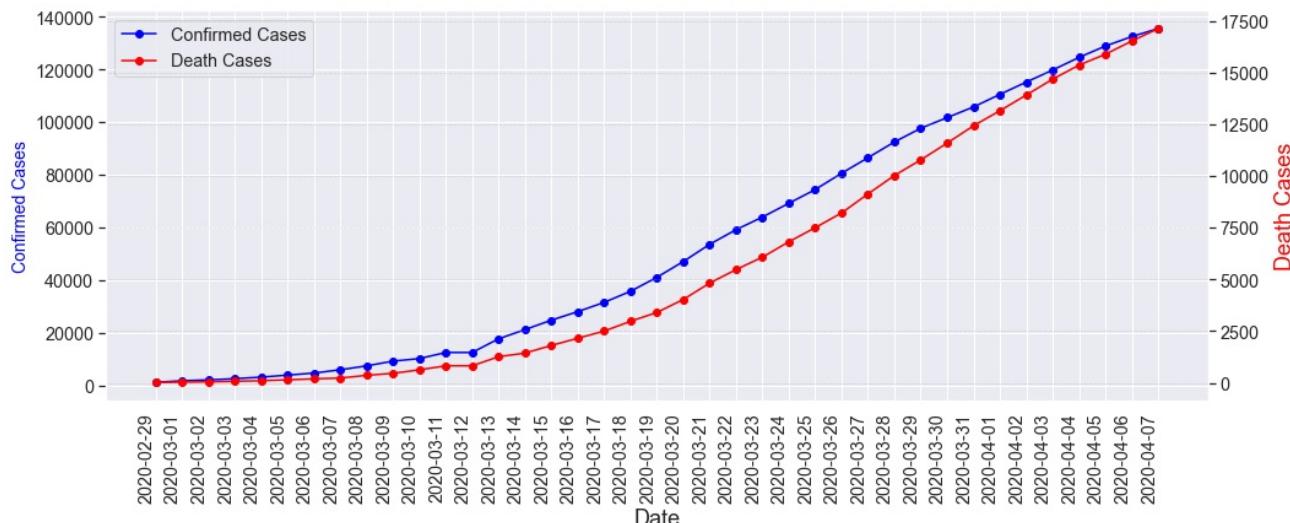
### Israel's Growth Curve Alignment with COVID-19 Death Toll of At Least 25 Cases



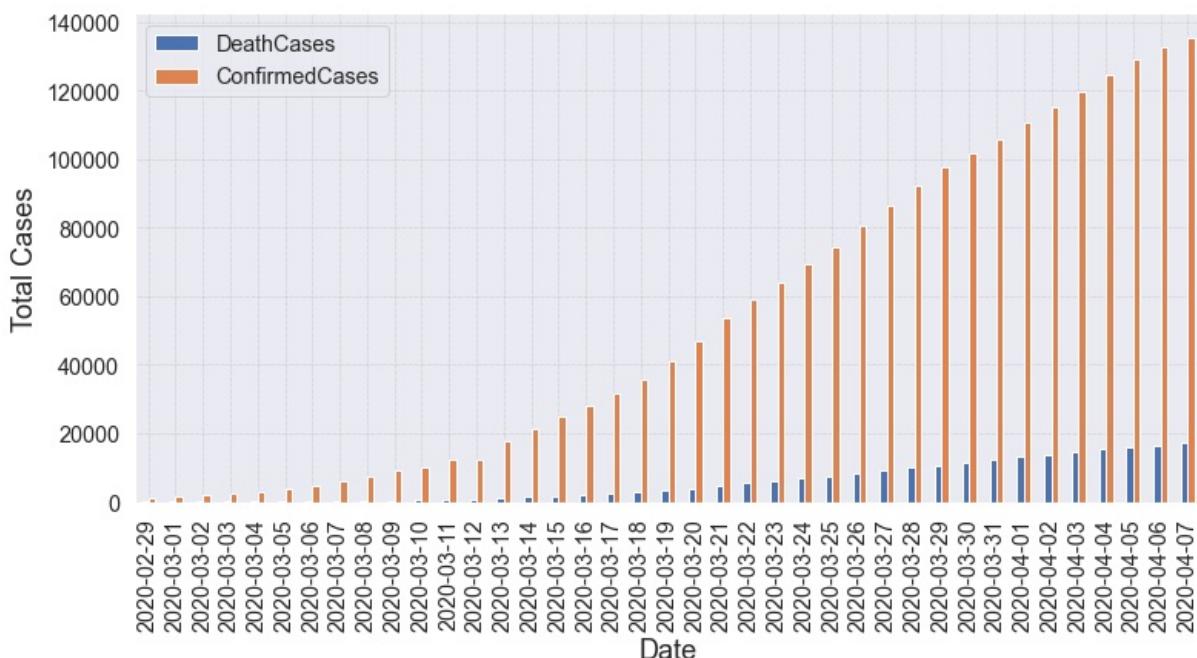
### Daily Cases and Death Count of COVID-19 from January - April in Israel



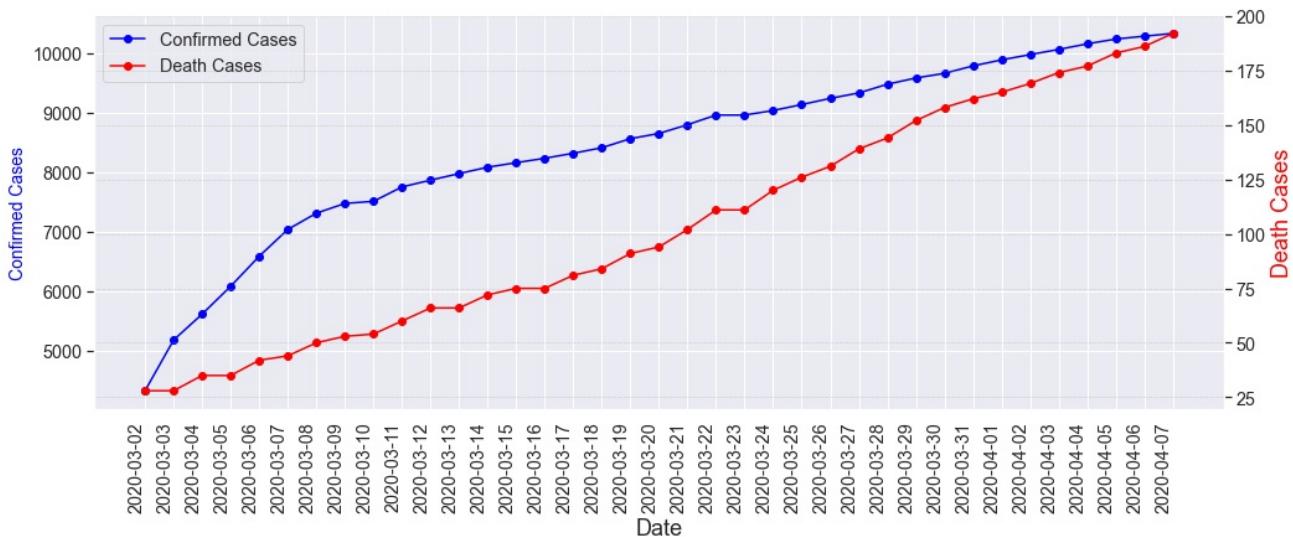
### Italy's Growth Curve Alignment with COVID-19 Death Toll of At Least 25 Cases



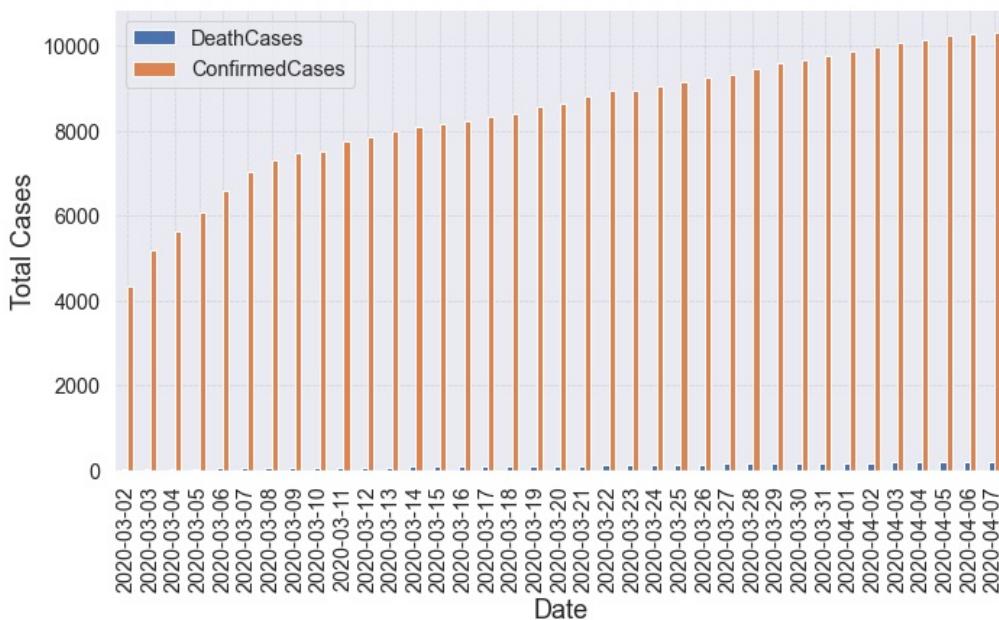
## Daily Cases and Death Count of COVID-19 from January - April in Italy



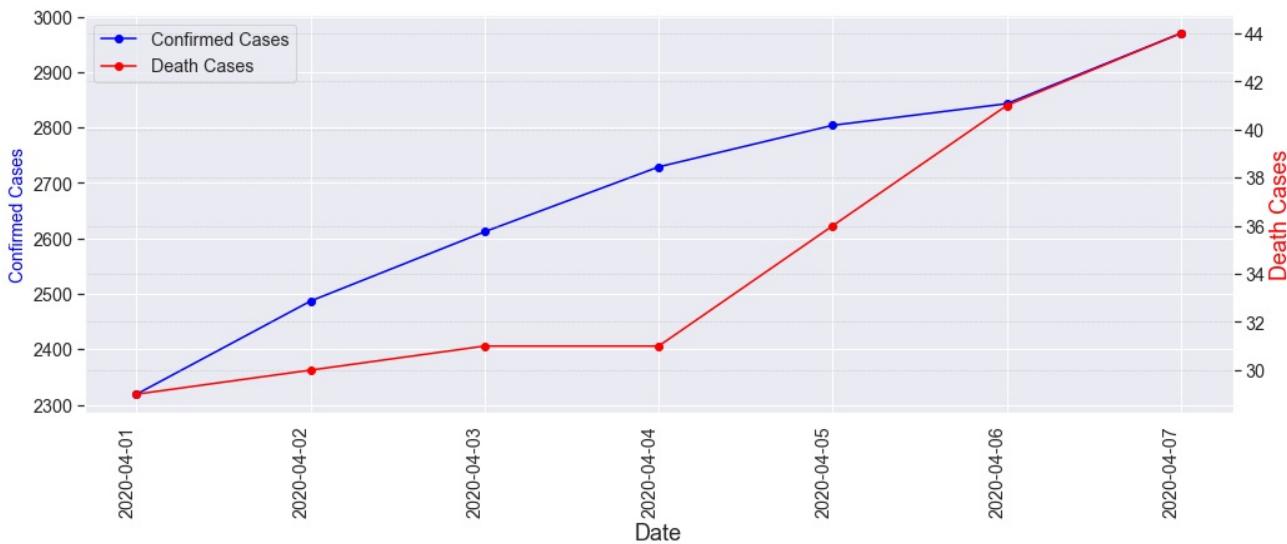
## Korea, South's Growth Curve Alignment with COVID-19 Death Toll of At Least 25 Cases



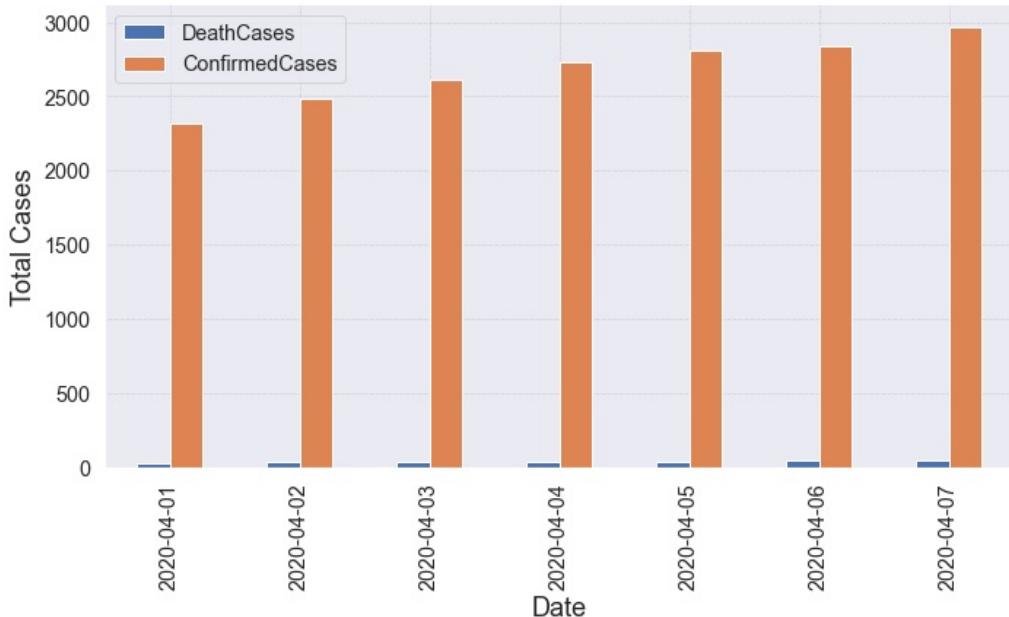
## Daily Cases and Death Count of COVID-19 from January - April in Korea, South



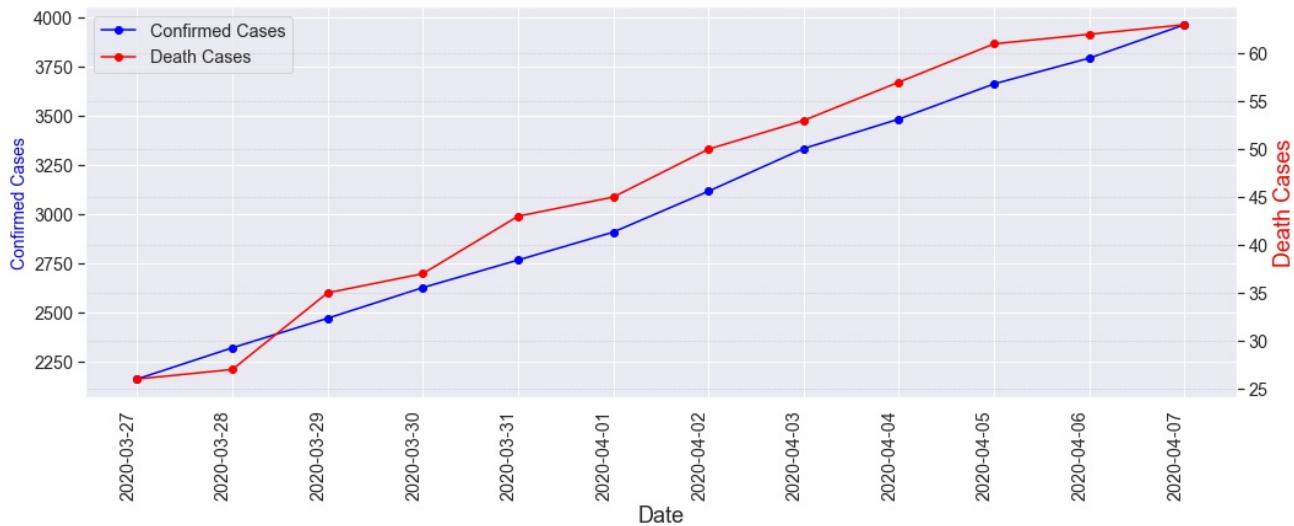
## Luxembourg's Growth Curve Alignment with COVID-19 Death Toll of At Least 25 Cases



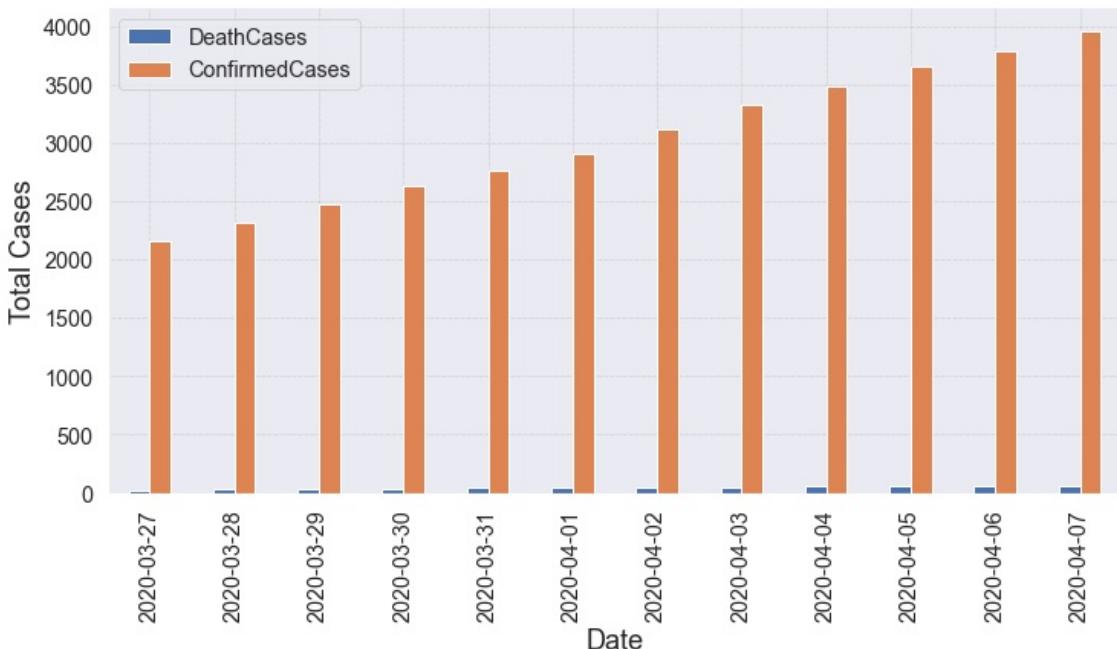
## Daily Cases and Death Count of COVID-19 from January - April in Luxembourg



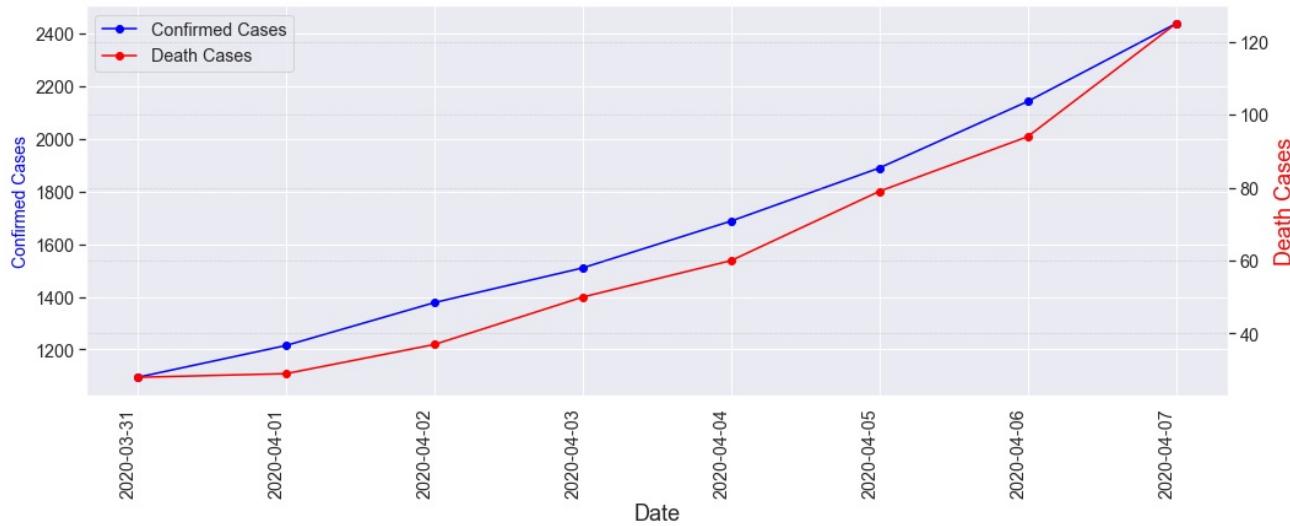
## Malaysia's Growth Curve Alignment with COVID-19 Death Toll of At Least 25 Cases



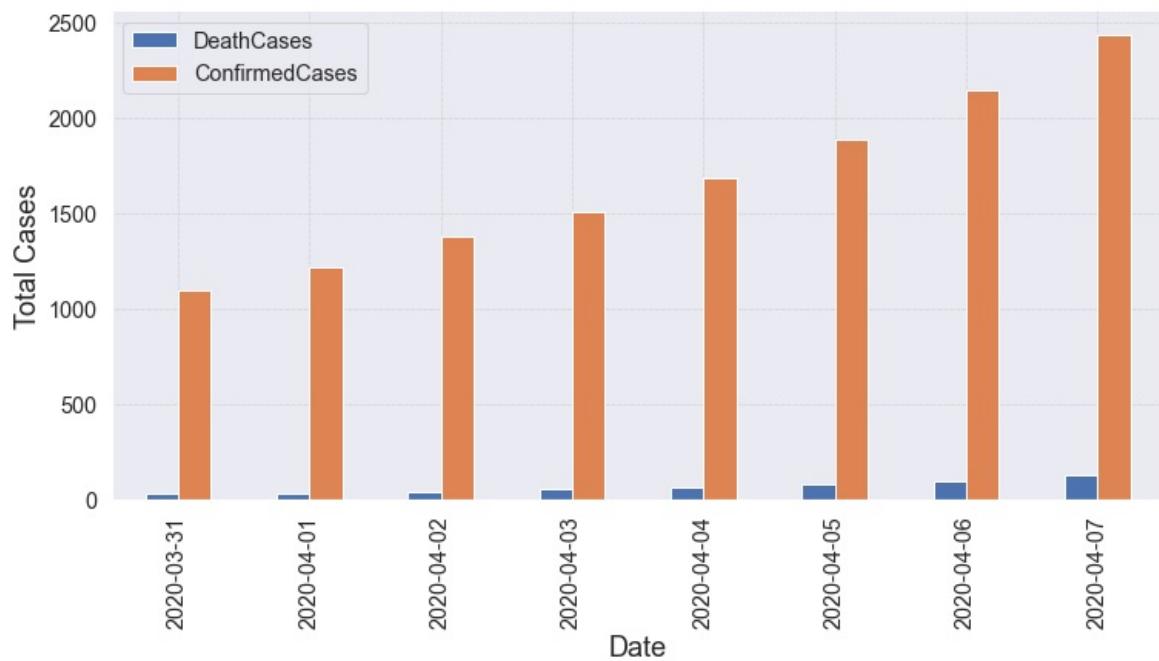
## Daily Cases and Death Count of COVID-19 from January - April in Malaysia



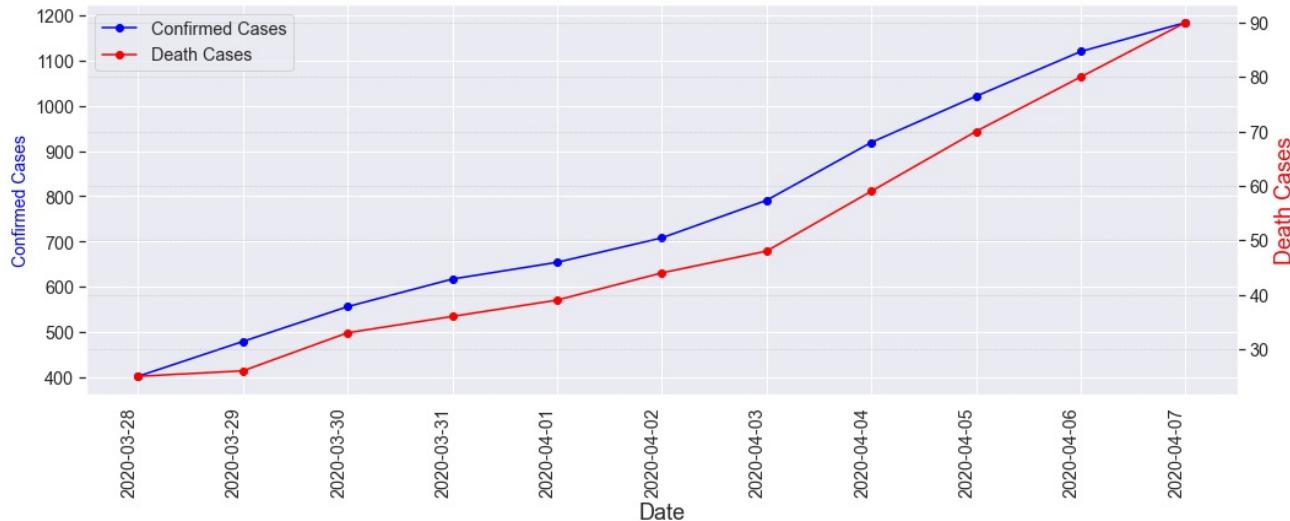
### Mexico's Growth Curve Alignment with COVID-19 Death Toll of At Least 25 Cases



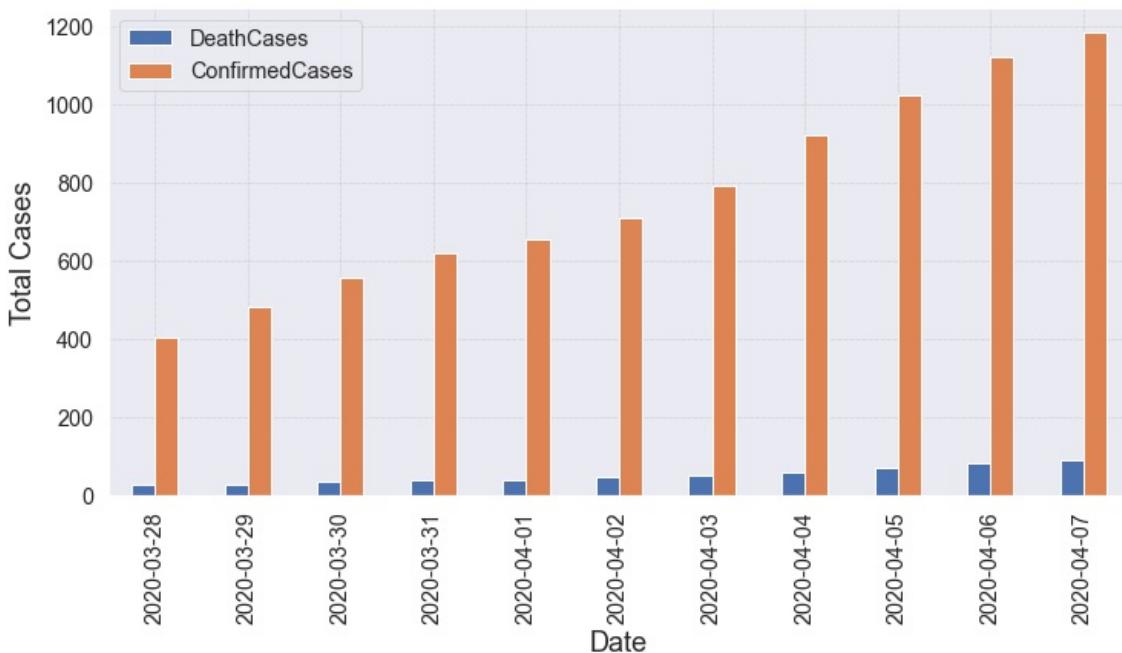
### Daily Cases and Death Count of COVID-19 from January - April in Mexico



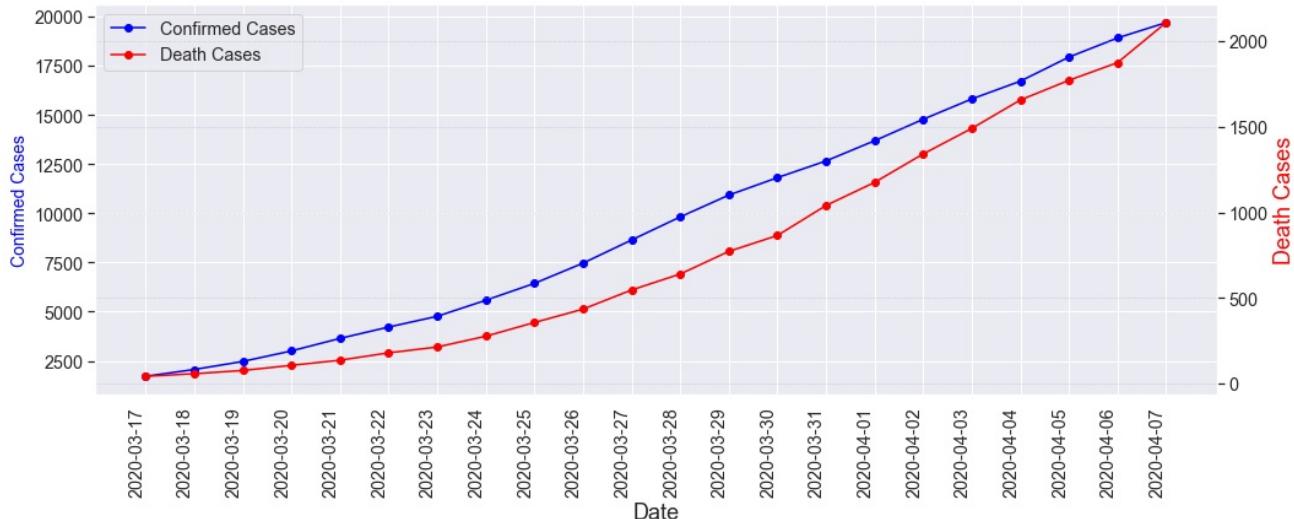
### Morocco's Growth Curve Alignment with COVID-19 Death Toll of At Least 25 Cases



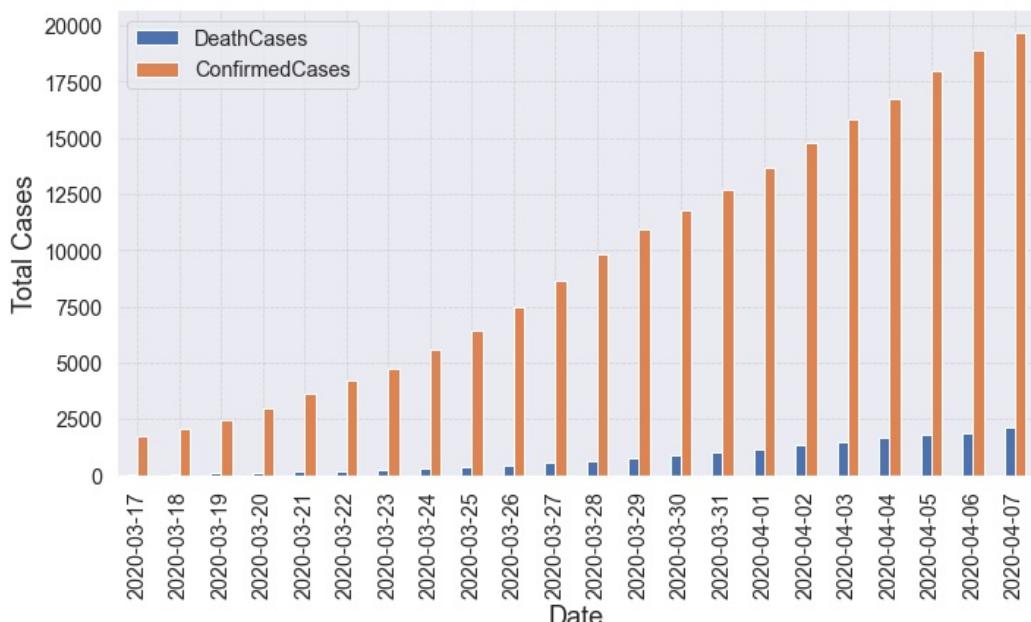
## Daily Cases and Death Count of COVID-19 from January - April in Morocco



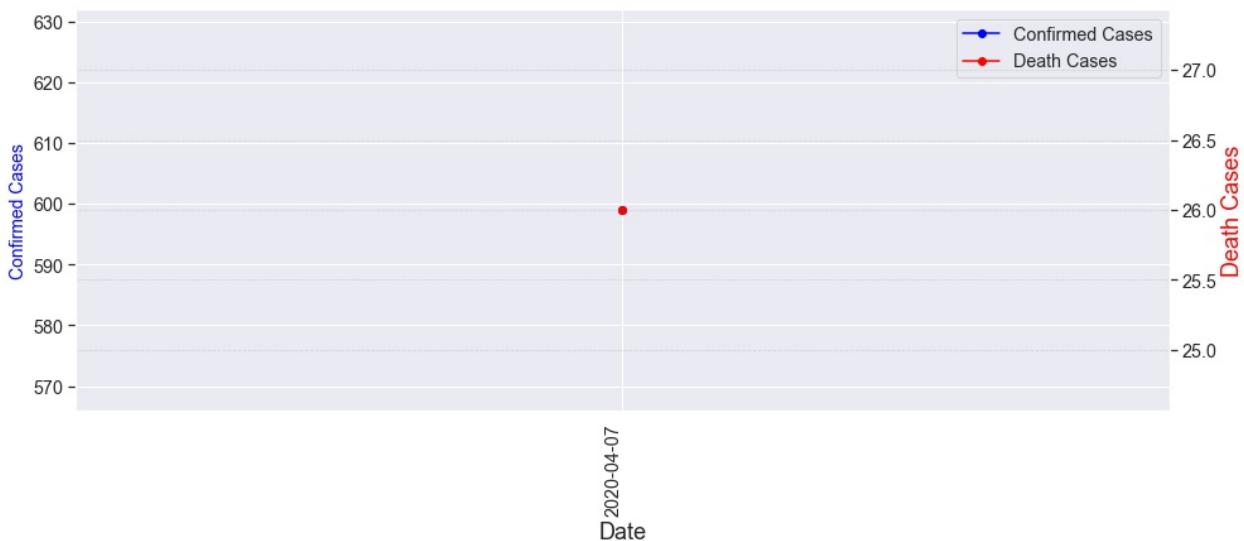
## Netherlands's Growth Curve Alignment with COVID-19 Death Toll of At Least 25 Cases



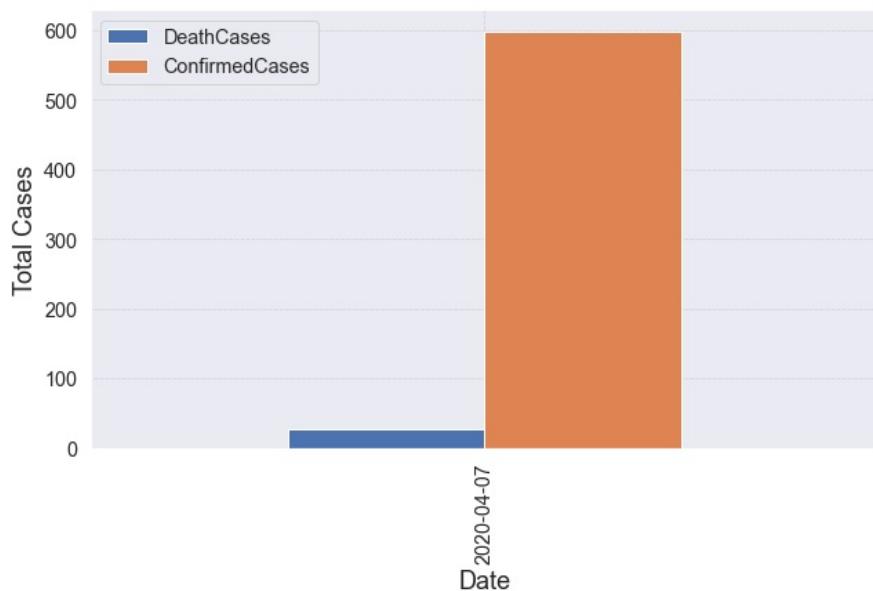
## Daily Cases and Death Count of COVID-19 from January - April in Netherlands



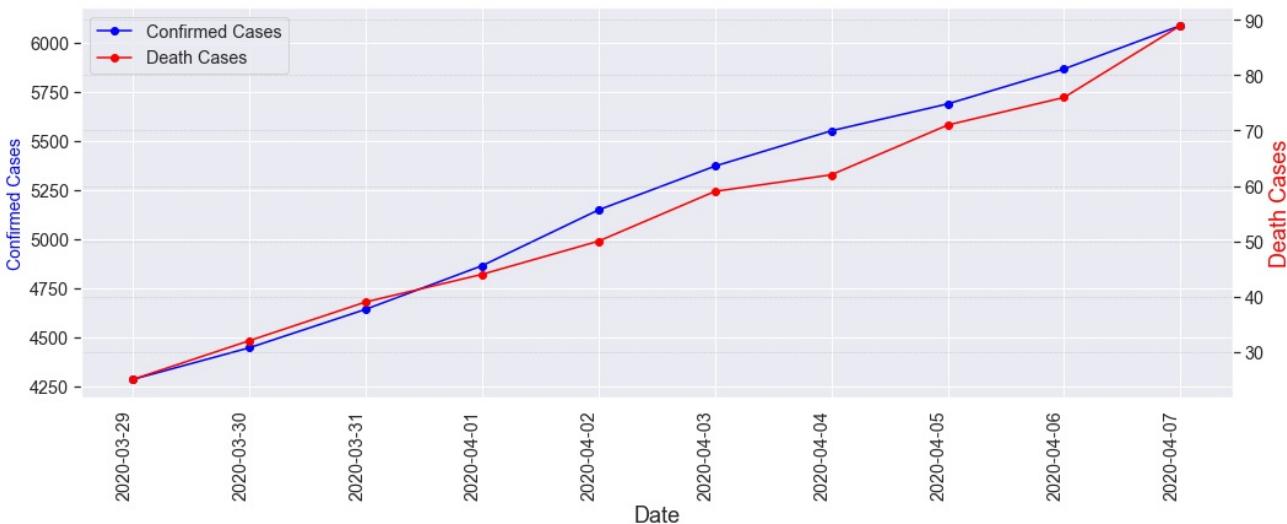
## North Macedonia's Growth Curve Alignment with COVID-19 Death Toll of At Least 25 Cases



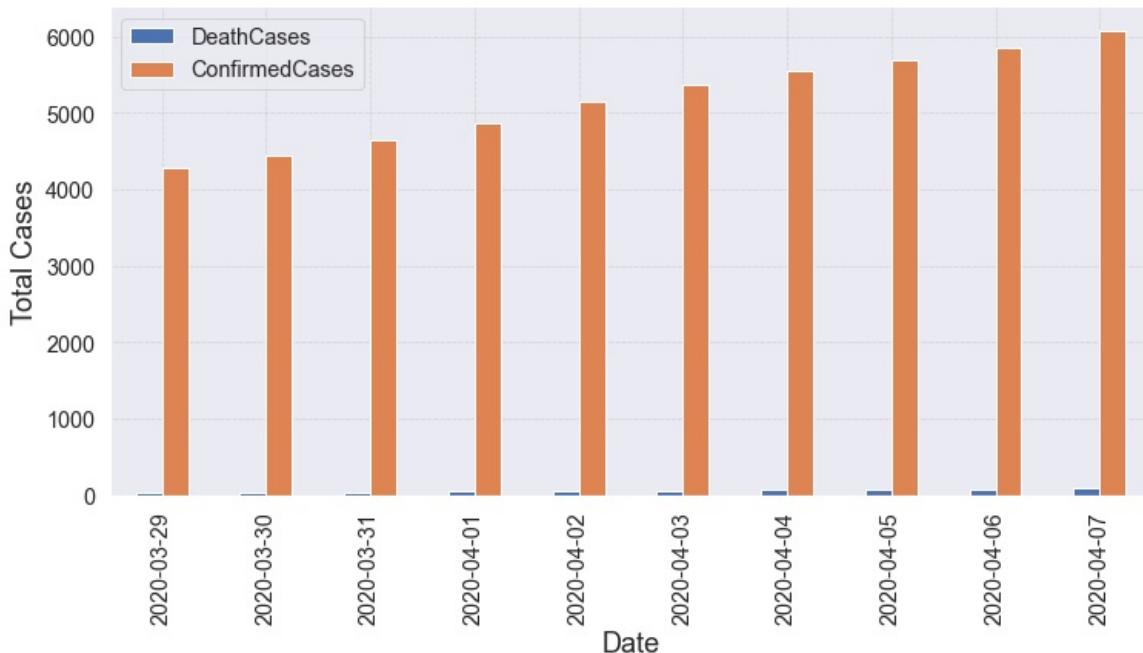
## Daily Cases and Death Count of COVID-19 from January - April in North Macedonia



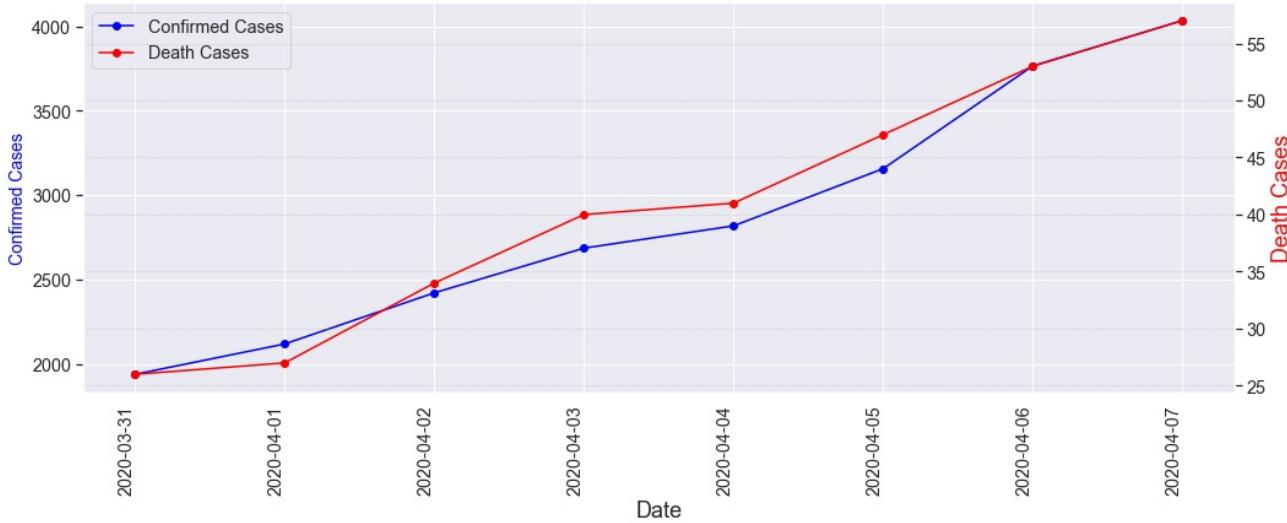
## Norway's Growth Curve Alignment with COVID-19 Death Toll of At Least 25 Cases



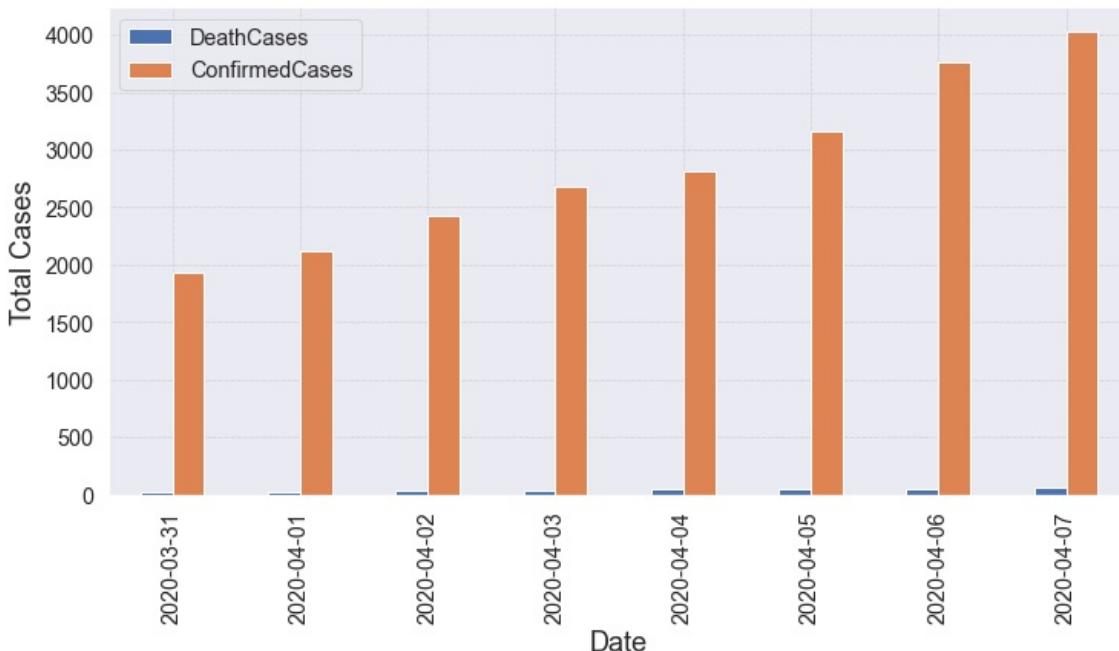
## Daily Cases and Death Count of COVID-19 from January - April in Norway



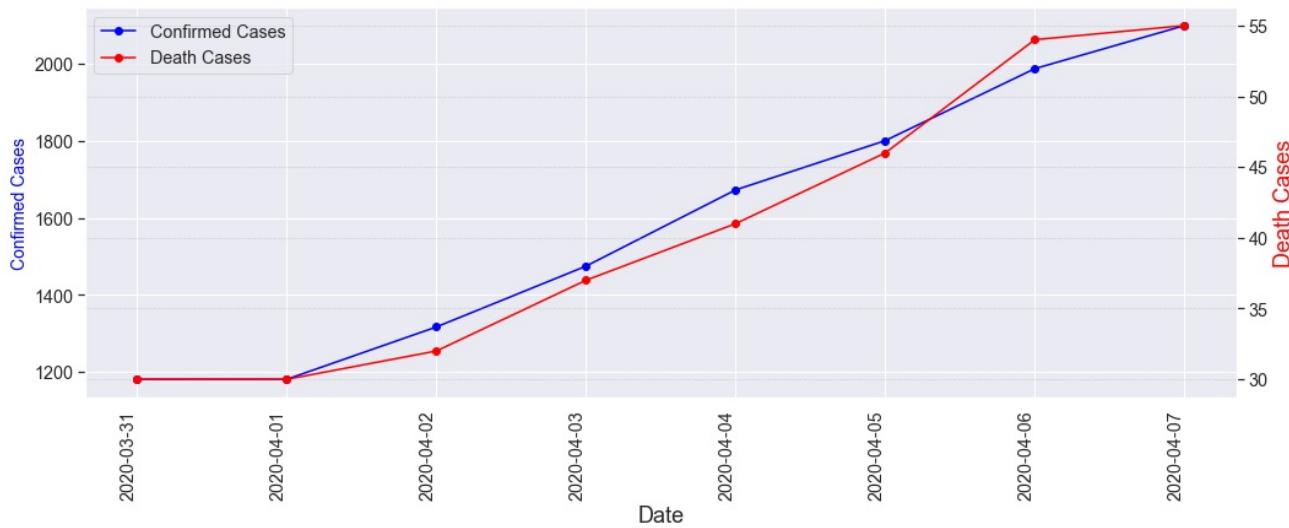
## Pakistan's Growth Curve Alignment with COVID-19 Death Toll of At Least 25 Cases



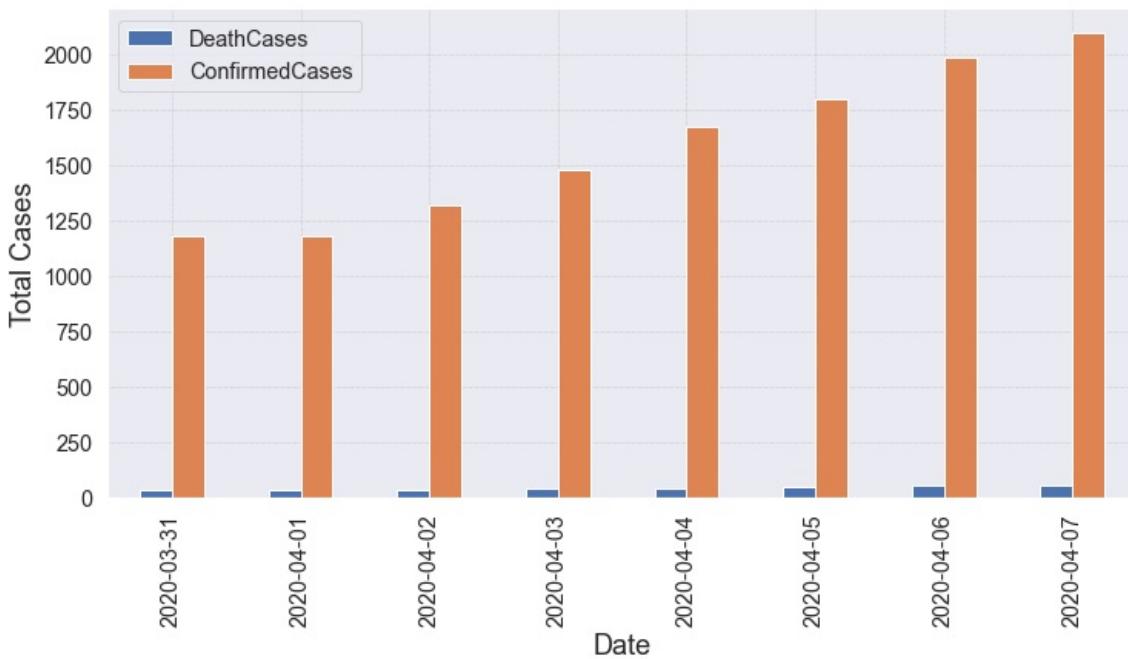
## Daily Cases and Death Count of COVID-19 from January - April in Pakistan



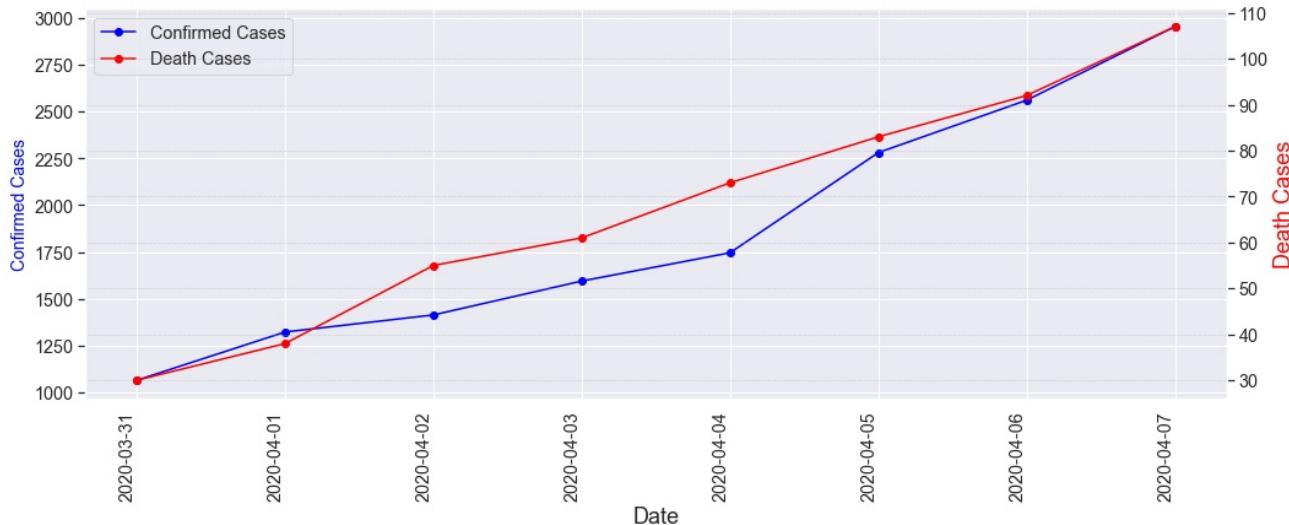
## Panama's Growth Curve Alignment with COVID-19 Death Toll of At Least 25 Cases



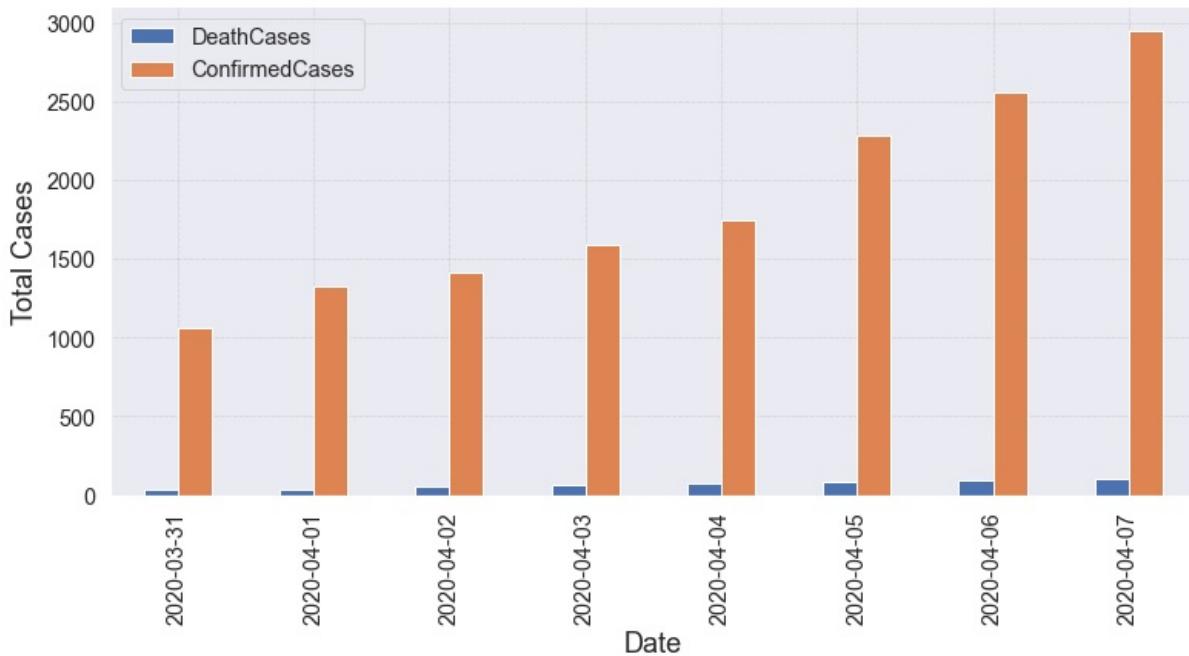
## Daily Cases and Death Count of COVID-19 from January - April in Panama



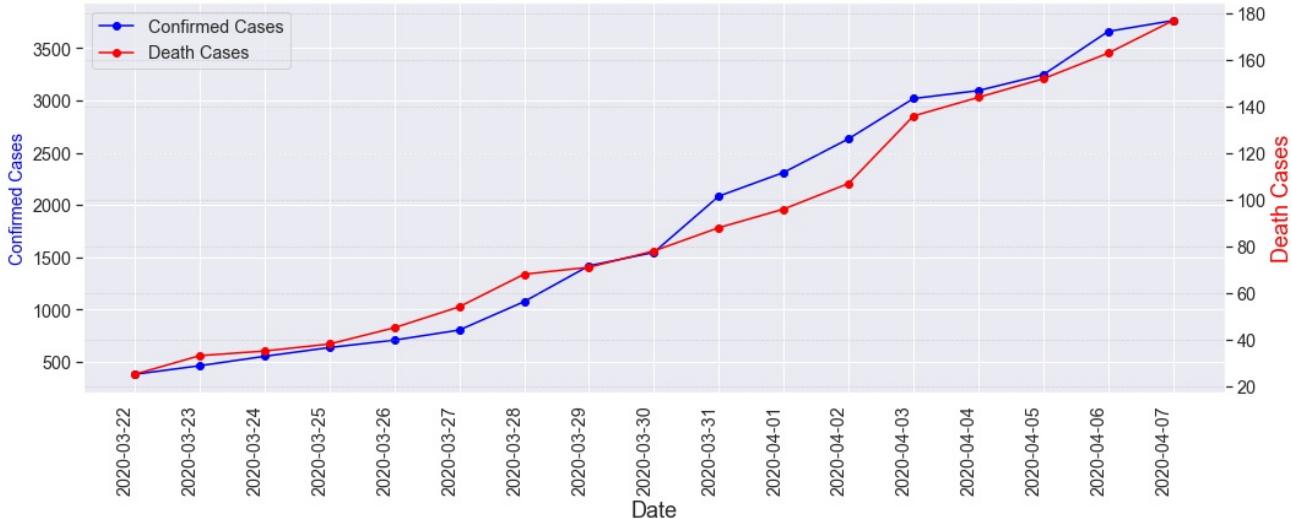
## Peru's Growth Curve Alignment with COVID-19 Death Toll of At Least 25 Cases



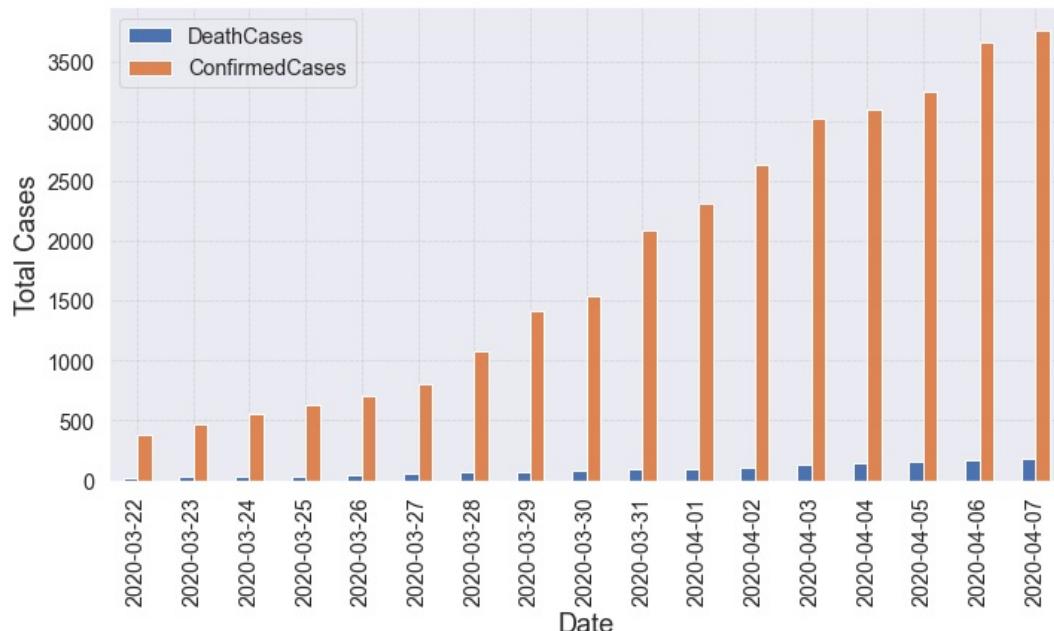
## Daily Cases and Death Count of COVID-19 from January - April in Peru



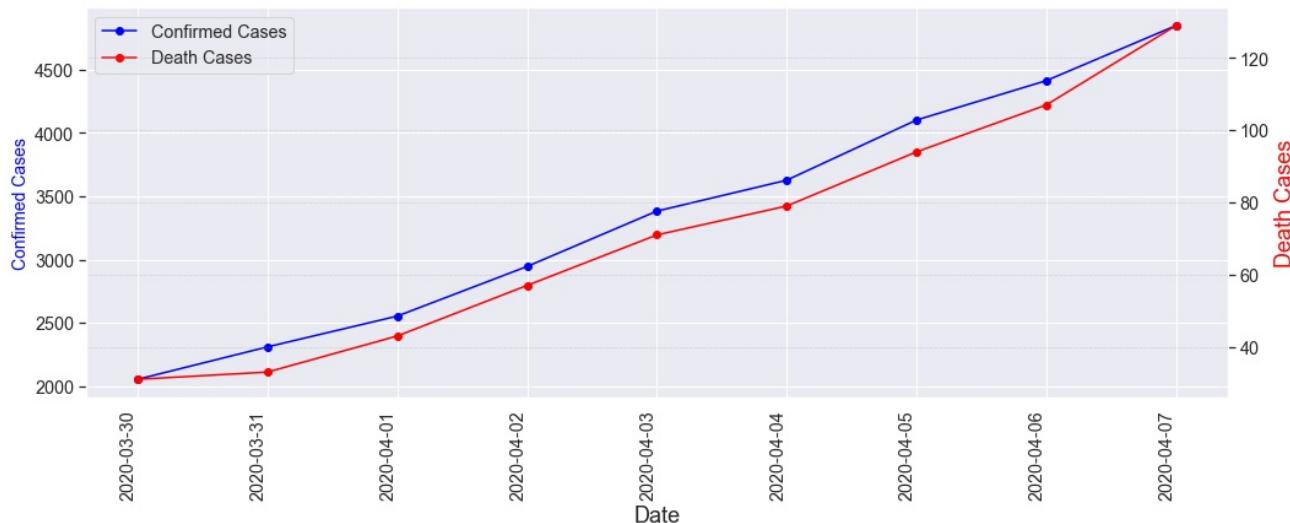
## Philippines's Growth Curve Alignment with COVID-19 Death Toll of At Least 25 Cases



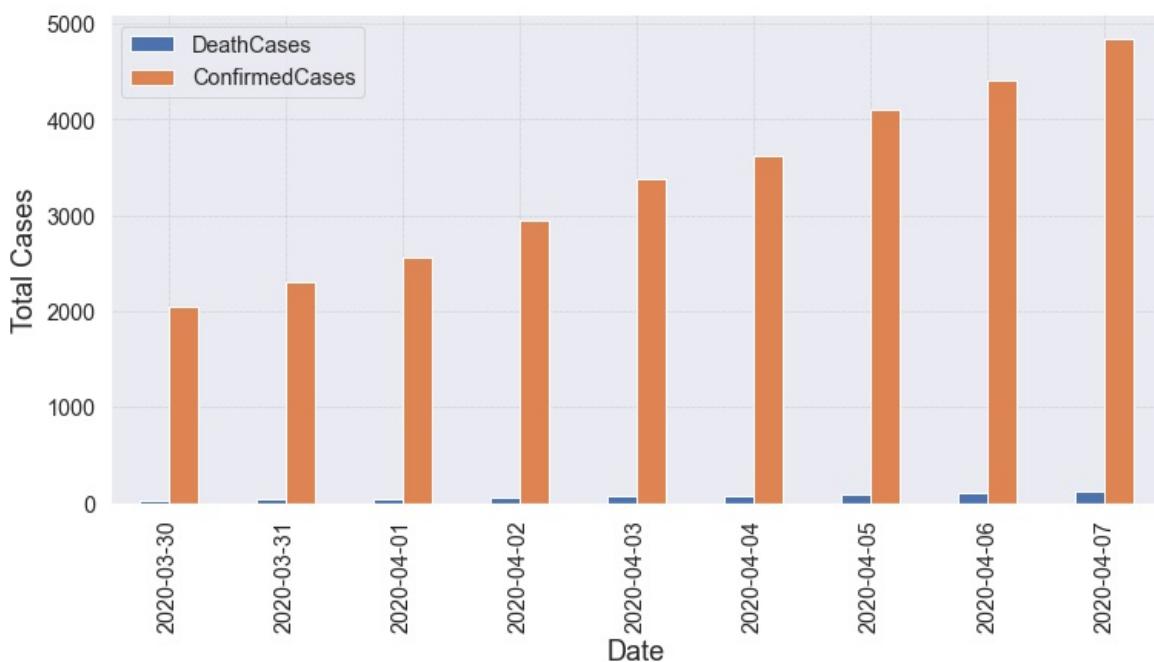
## Daily Cases and Death Count of COVID-19 from January - April in Philippines



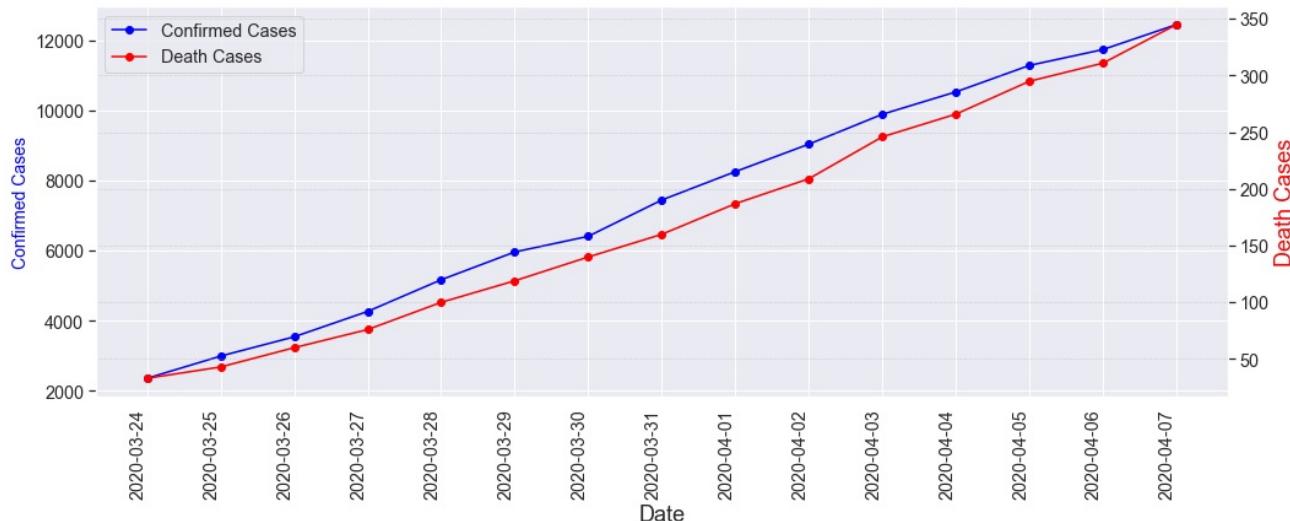
## Poland's Growth Curve Alignment with COVID-19 Death Toll of At Least 25 Cases



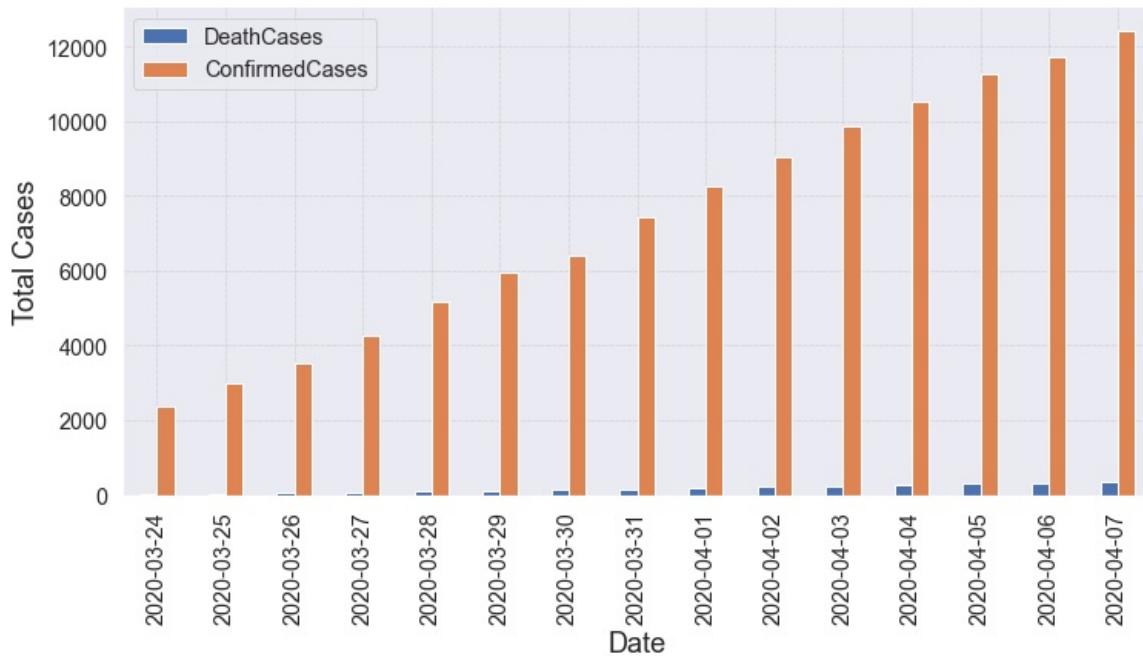
## Daily Cases and Death Count of COVID-19 from January - April in Poland



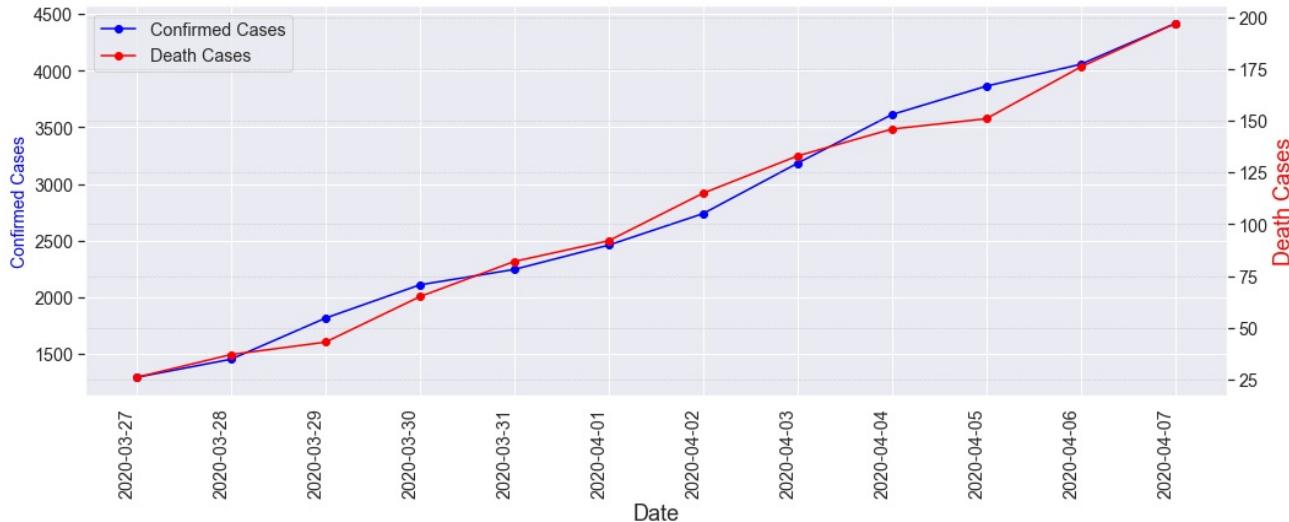
## Portugal's Growth Curve Alignment with COVID-19 Death Toll of At Least 25 Cases



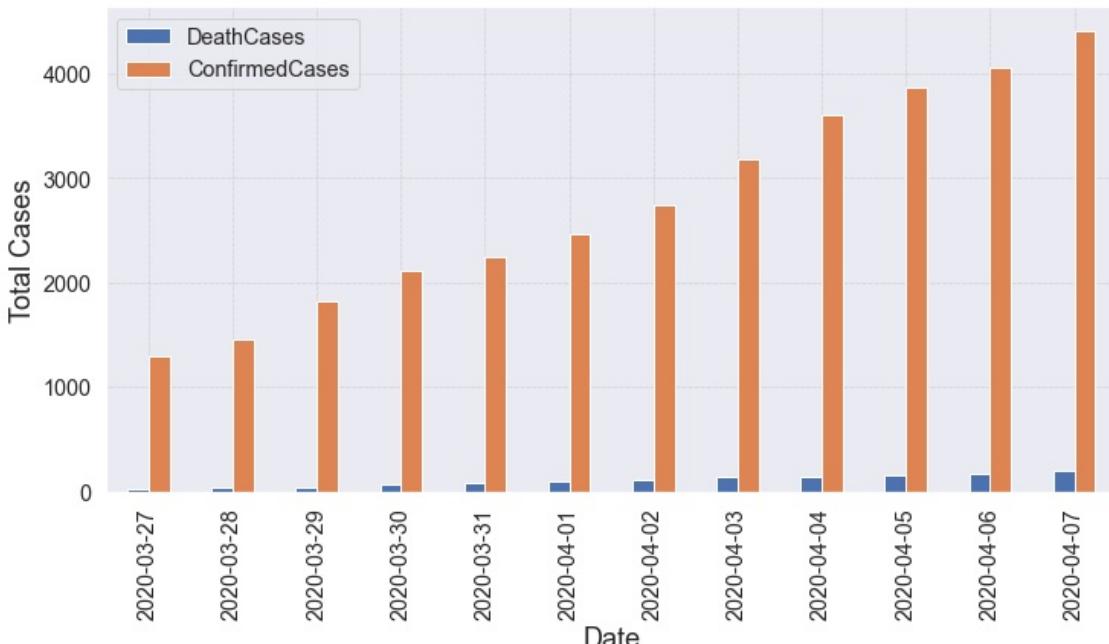
## Daily Cases and Death Count of COVID-19 from January - April in Portugal



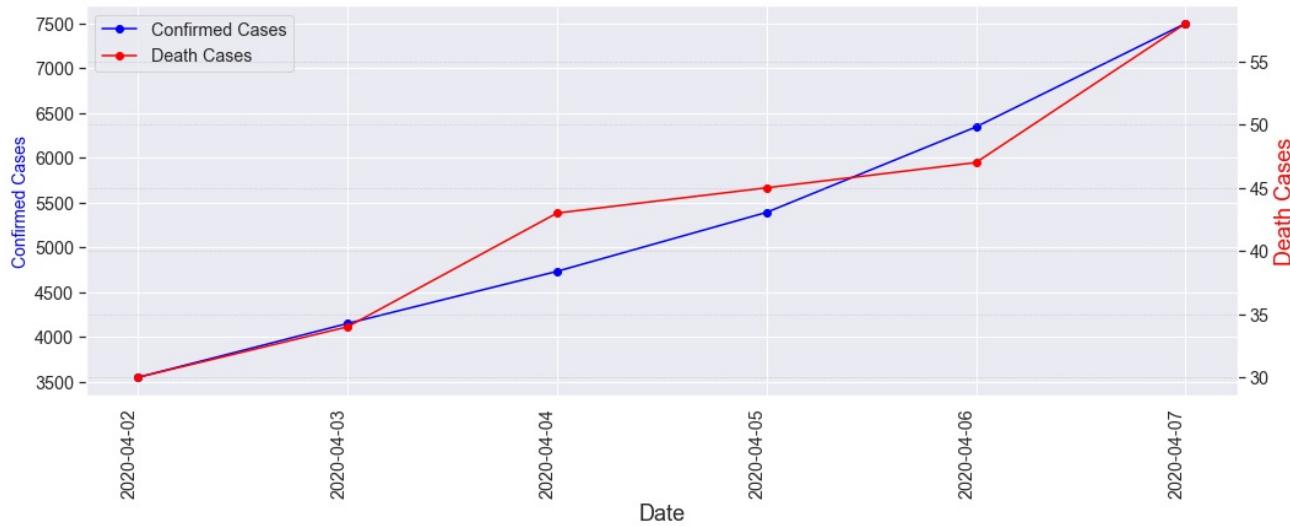
## Romania's Growth Curve Alignment with COVID-19 Death Toll of At Least 25 Cases



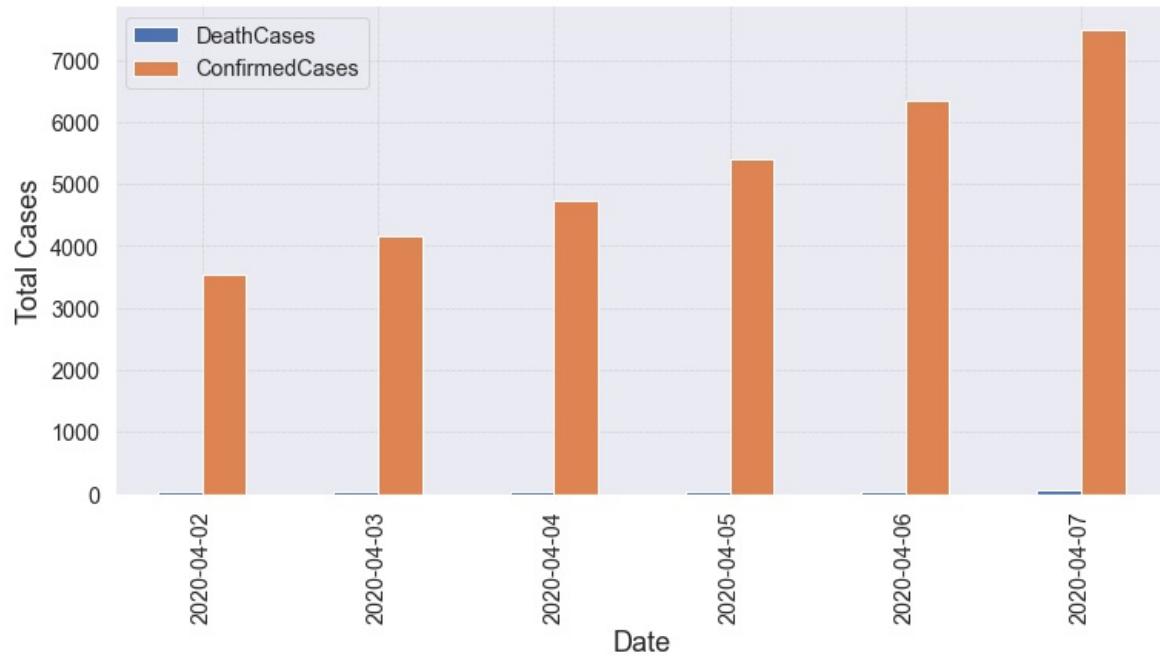
## Daily Cases and Death Count of COVID-19 from January - April in Romania



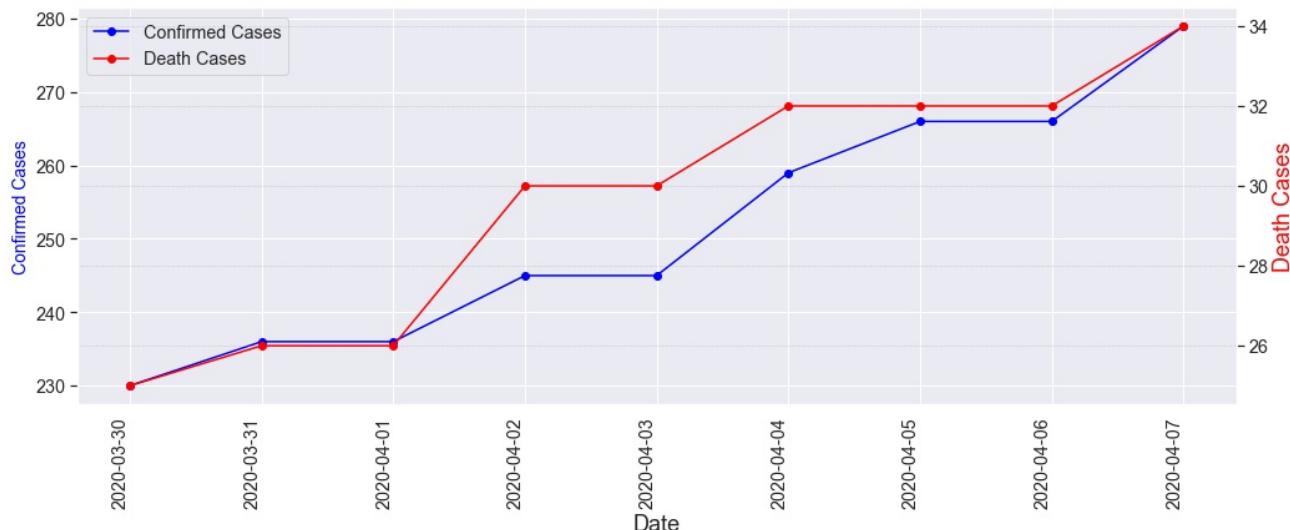
## Russia's Growth Curve Alignment with COVID-19 Death Toll of At Least 25 Cases



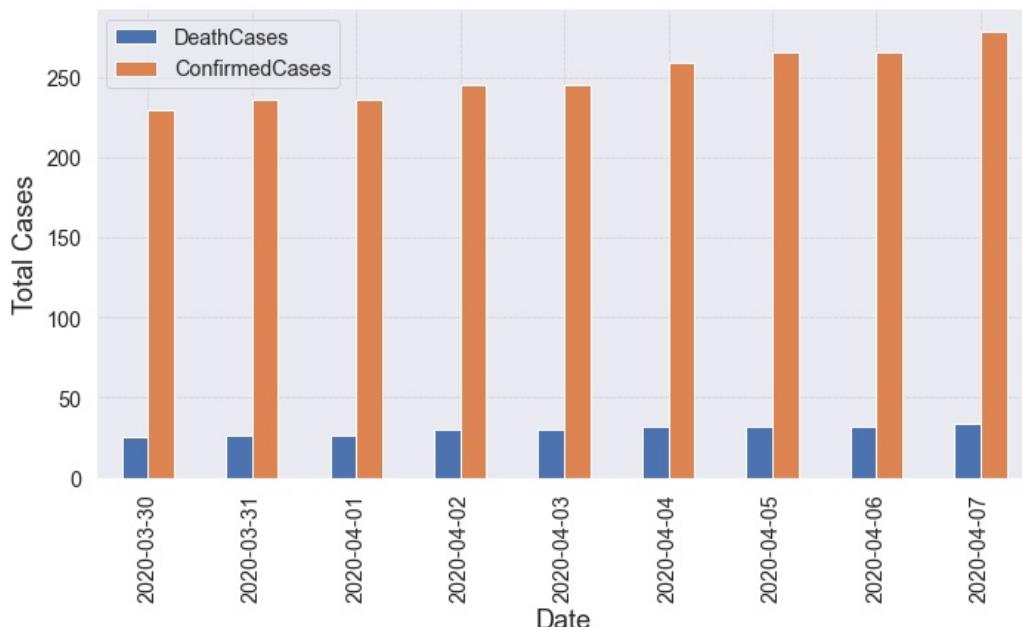
## Daily Cases and Death Count of COVID-19 from January - April in Russia



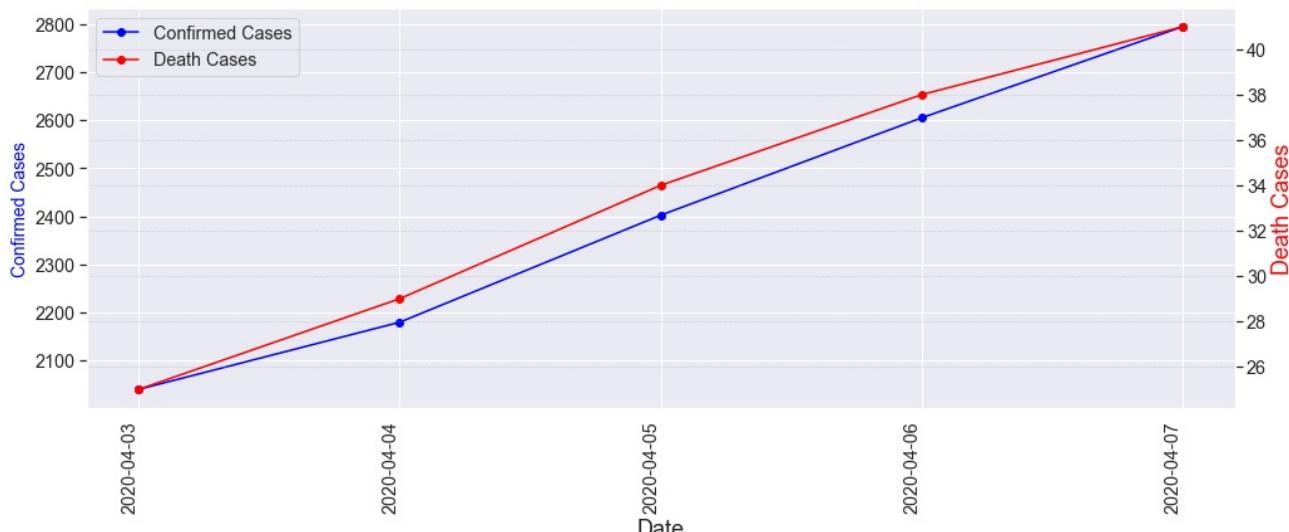
## San Marino's Growth Curve Alignment with COVID-19 Death Toll of At Least 25 Cases



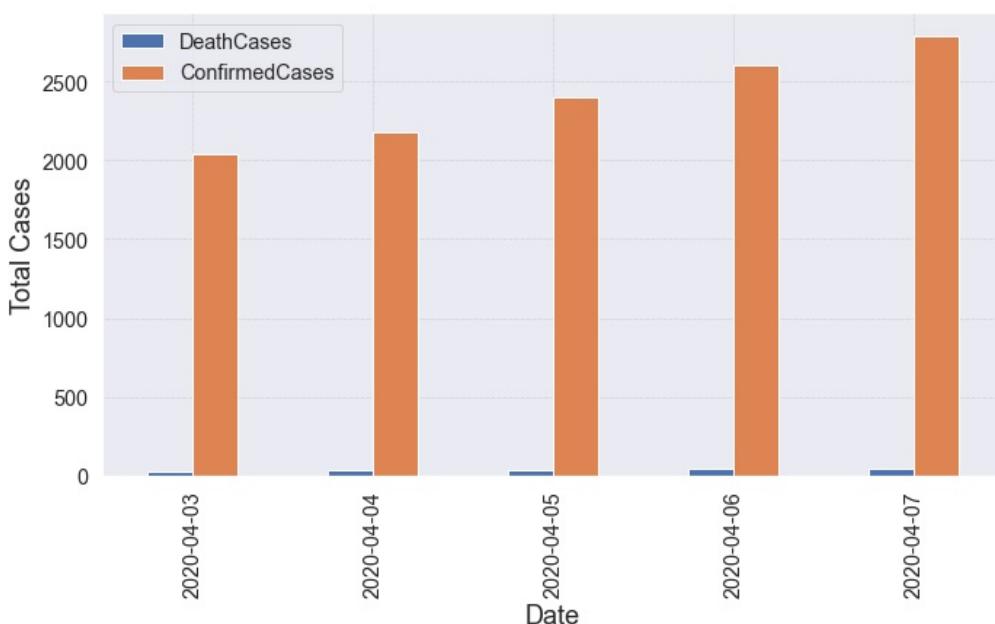
## Daily Cases and Death Count of COVID-19 from January - April in San Marino



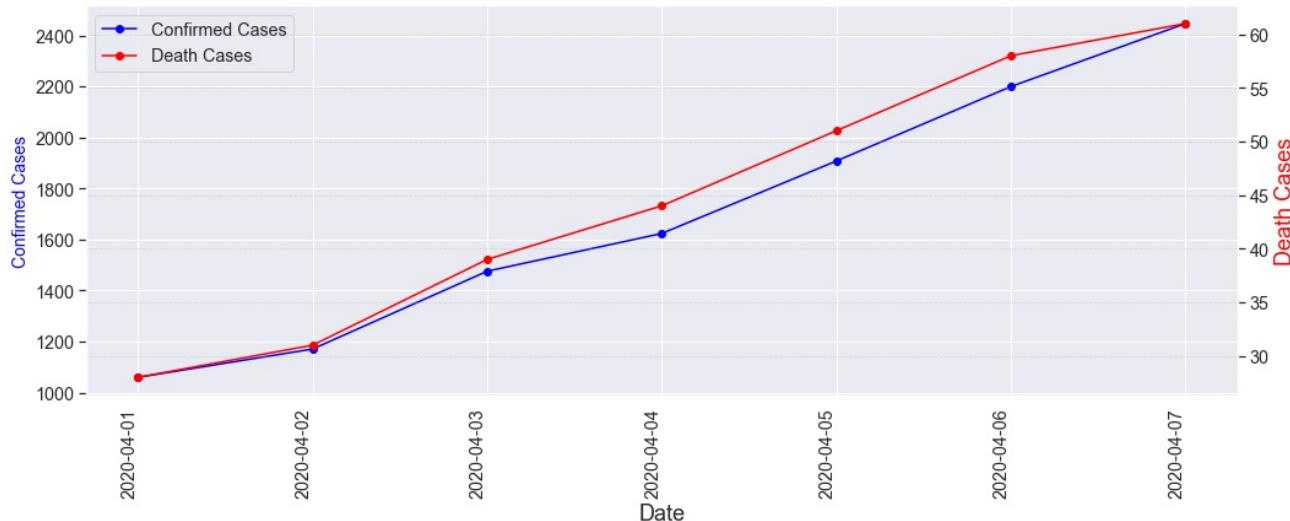
## Saudi Arabia's Growth Curve Alignment with COVID-19 Death Toll of At Least 25 Cases



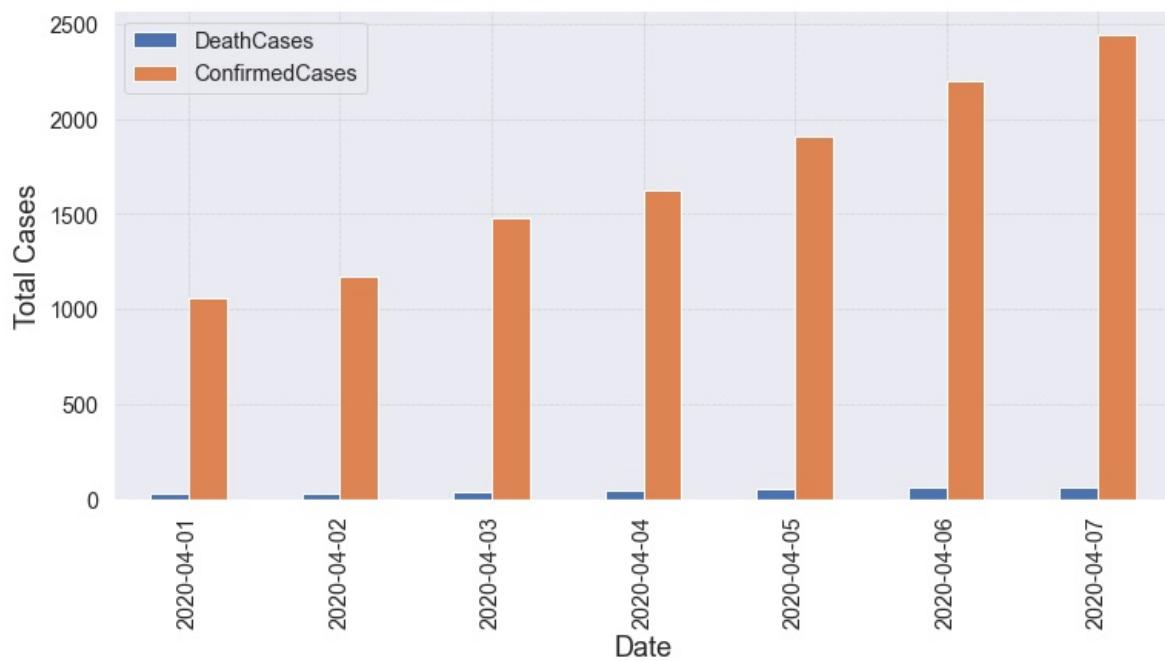
## Daily Cases and Death Count of COVID-19 from January - April in Saudi Arabia



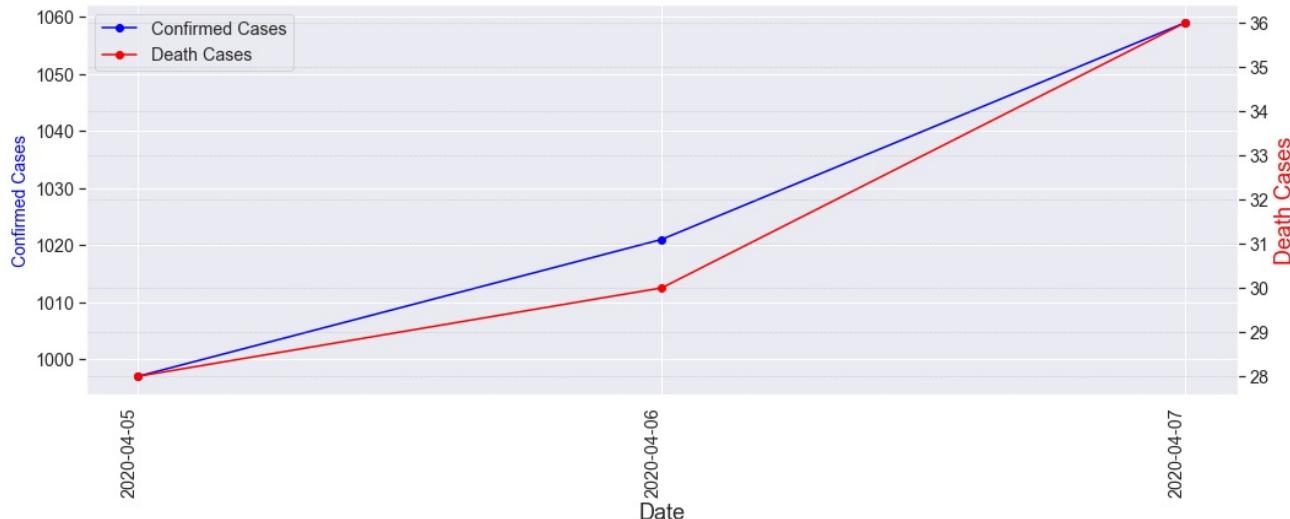
### Serbia's Growth Curve Alignment with COVID-19 Death Toll of At Least 25 Cases



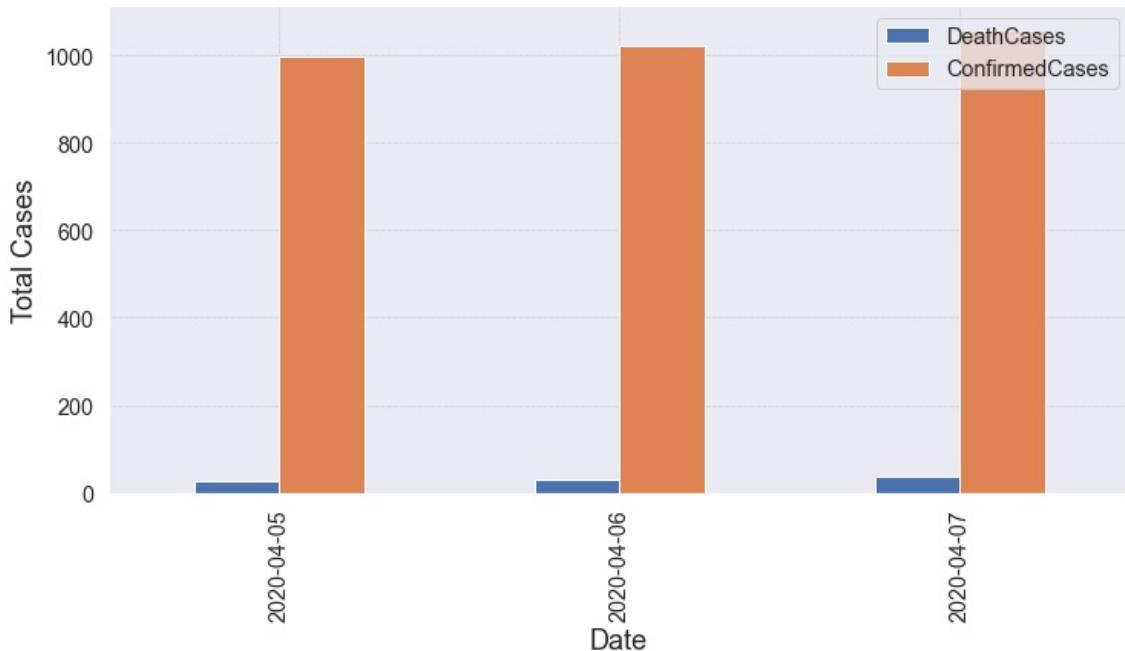
### Daily Cases and Death Count of COVID-19 from January - April in Serbia



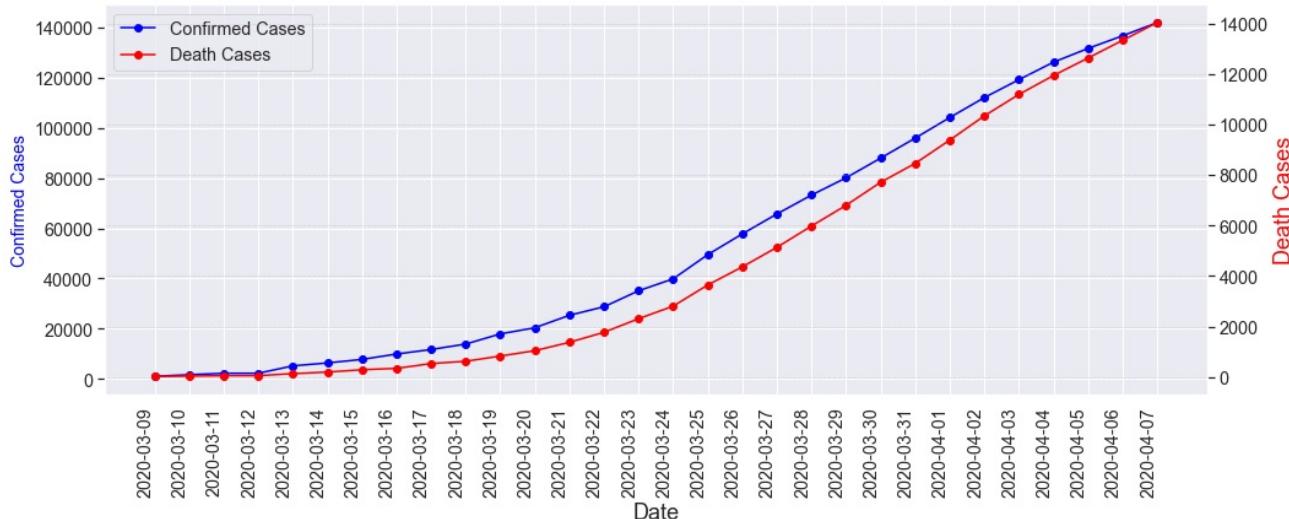
### Slovenia's Growth Curve Alignment with COVID-19 Death Toll of At Least 25 Cases



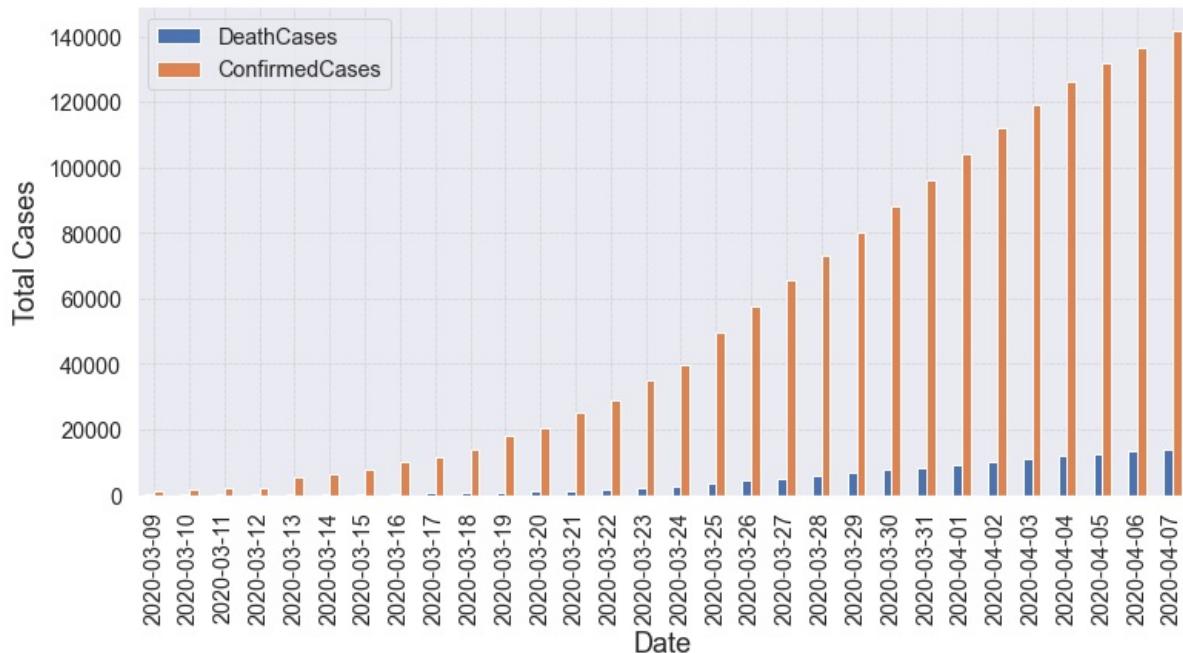
## Daily Cases and Death Count of COVID-19 from January - April in Slovenia



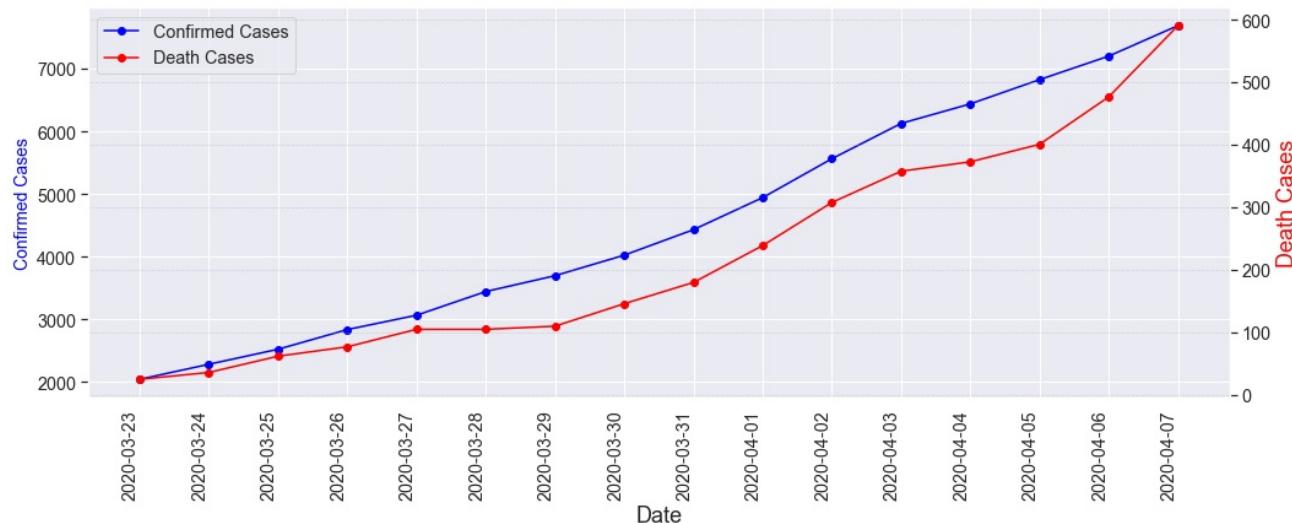
## Spain's Growth Curve Alignment with COVID-19 Death Toll of At Least 25 Cases



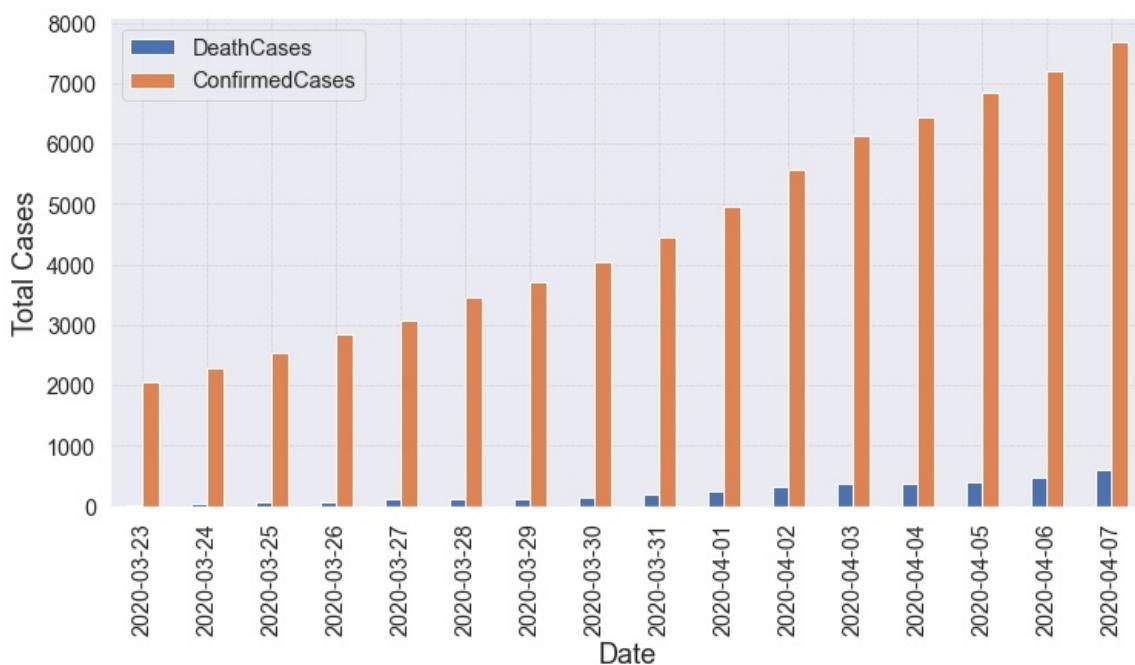
## Daily Cases and Death Count of COVID-19 from January - April in Spain



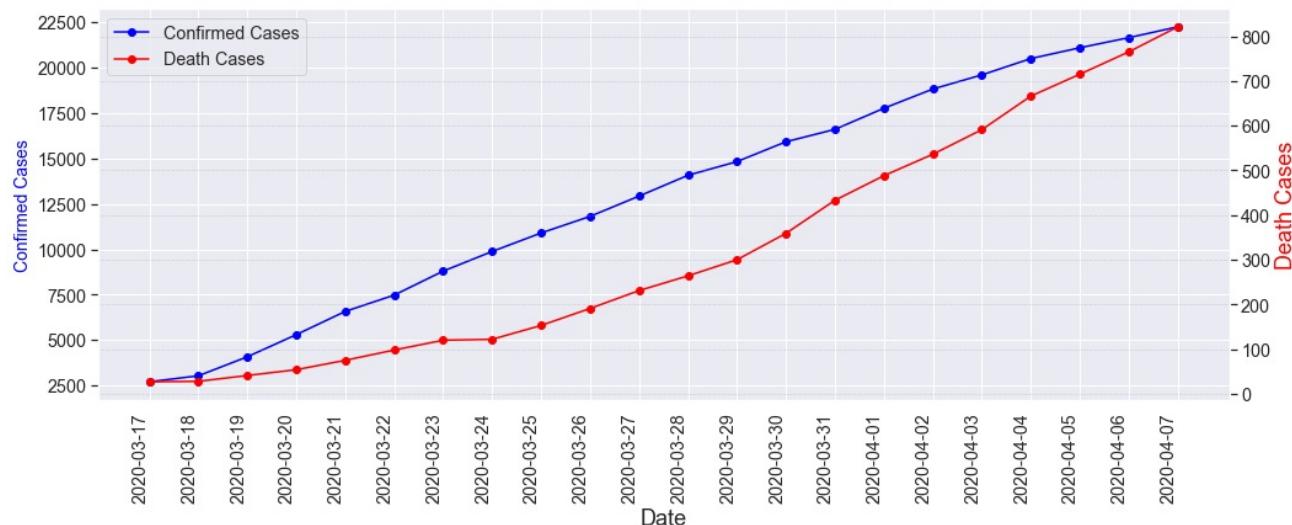
## Sweden's Growth Curve Alignment with COVID-19 Death Toll of At Least 25 Cases



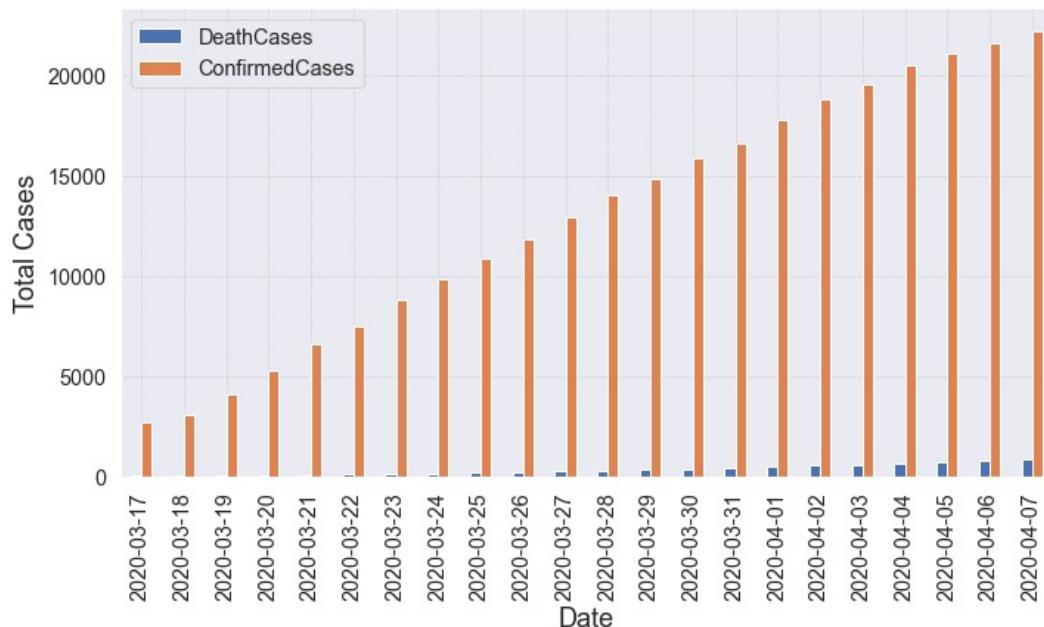
## Daily Cases and Death Count of COVID-19 from January - April in Sweden



## Switzerland's Growth Curve Alignment with COVID-19 Death Toll of At Least 25 Cases



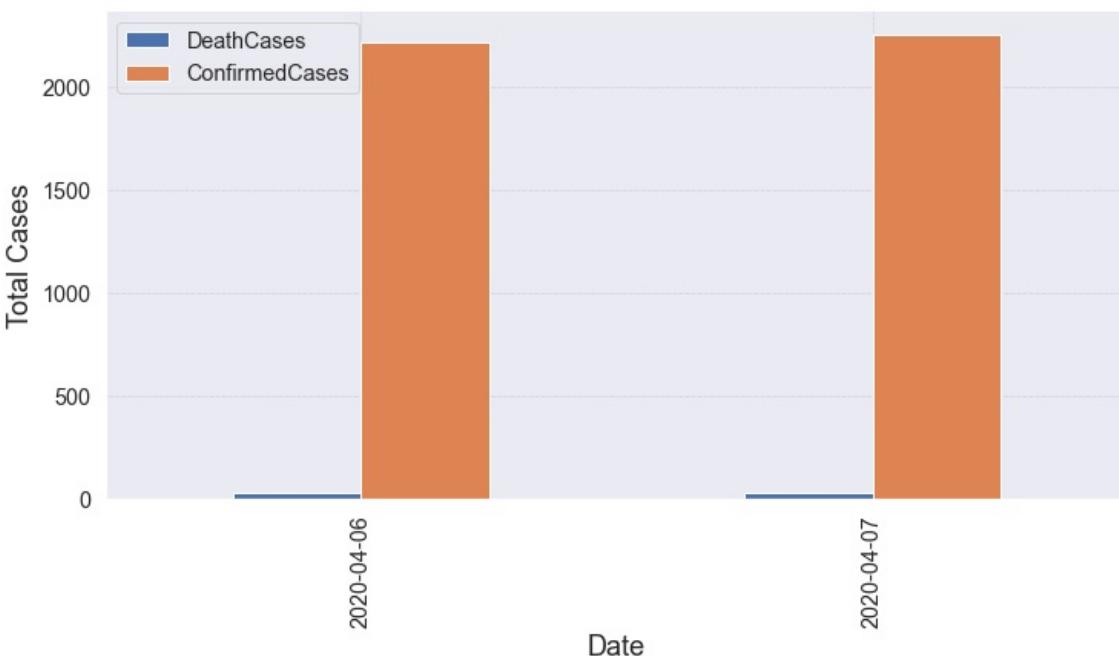
## Daily Cases and Death Count of COVID-19 from January - April in Switzerland



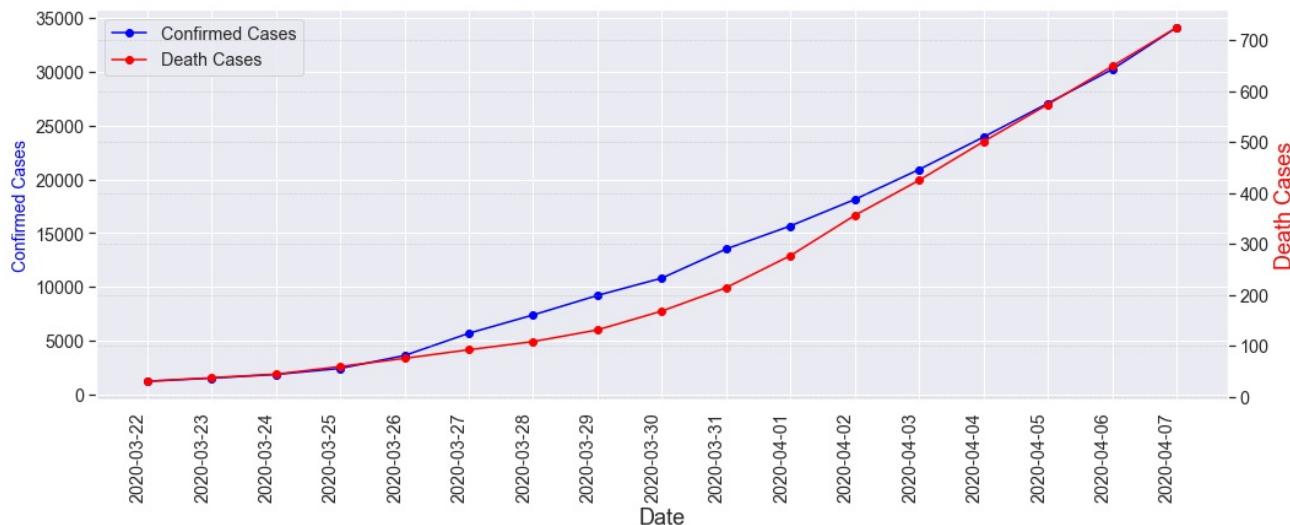
## Thailand's Growth Curve Alignment with COVID-19 Death Toll of At Least 25 Cases



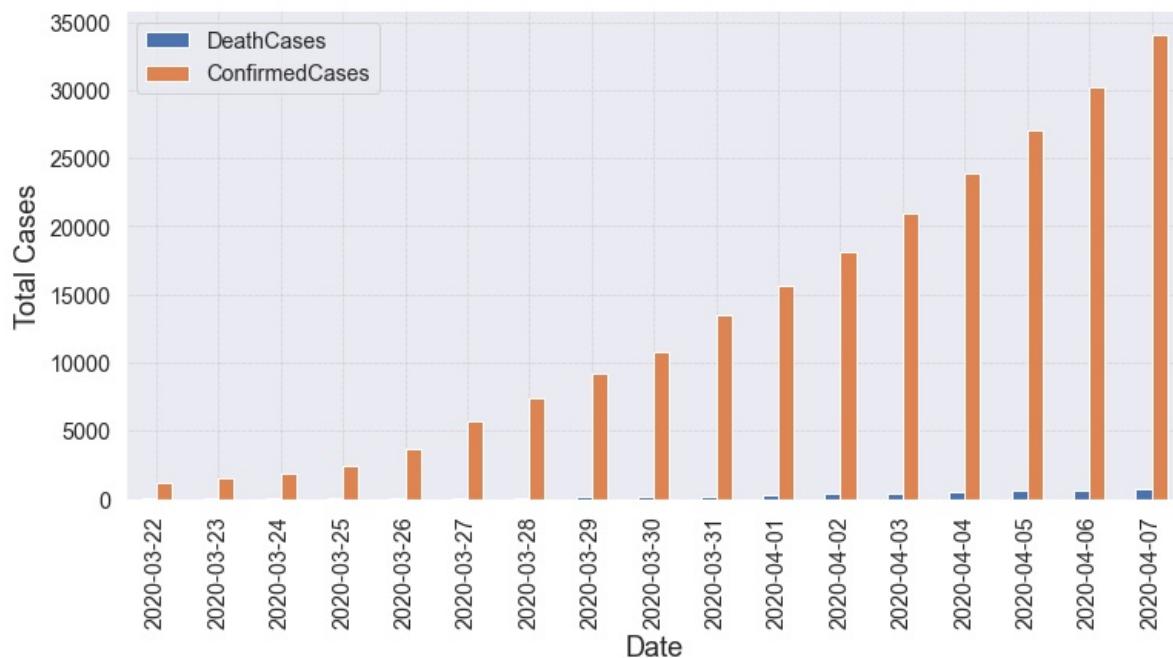
## Daily Cases and Death Count of COVID-19 from January - April in Thailand



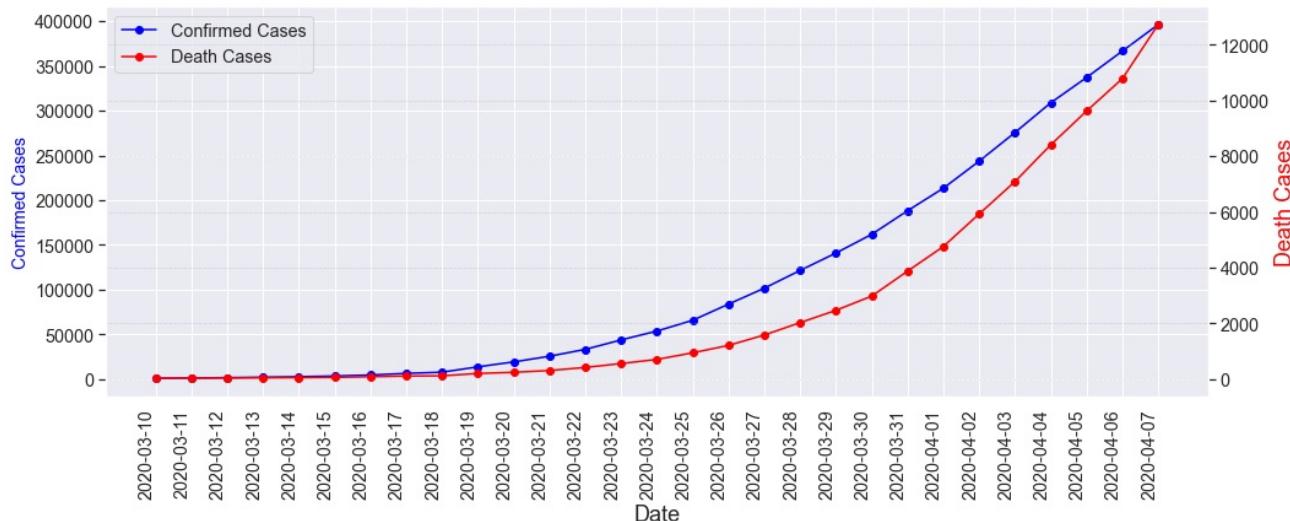
## Turkey's Growth Curve Alignment with COVID-19 Death Toll of At Least 25 Cases



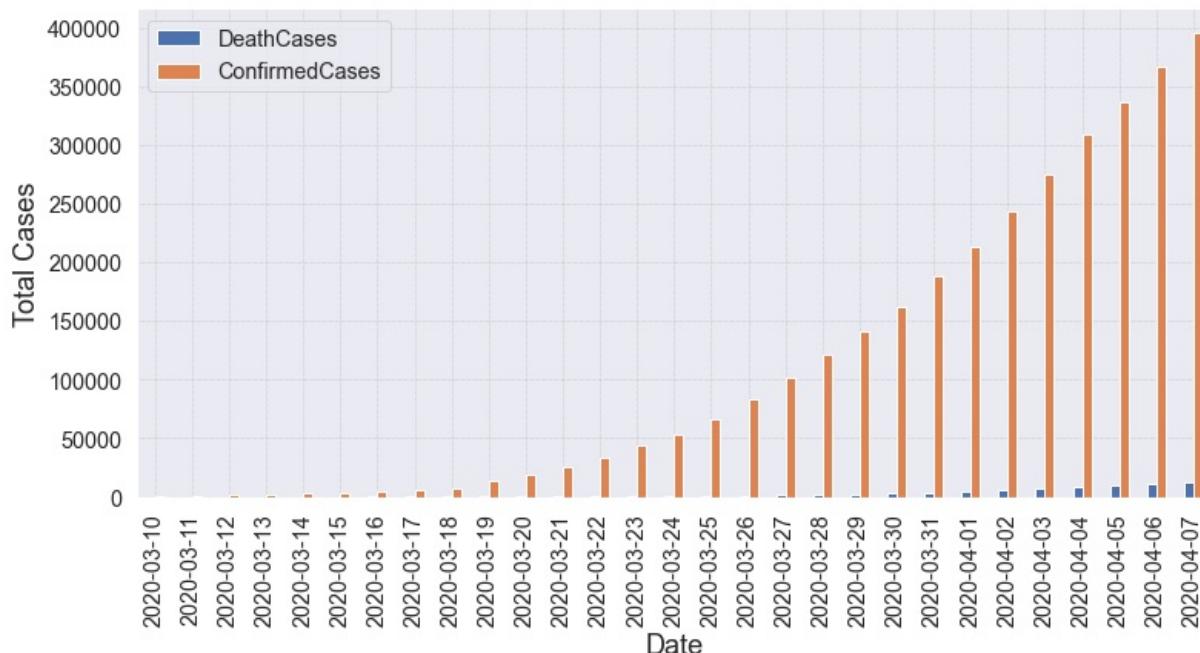
## Daily Cases and Death Count of COVID-19 from January - April in Turkey



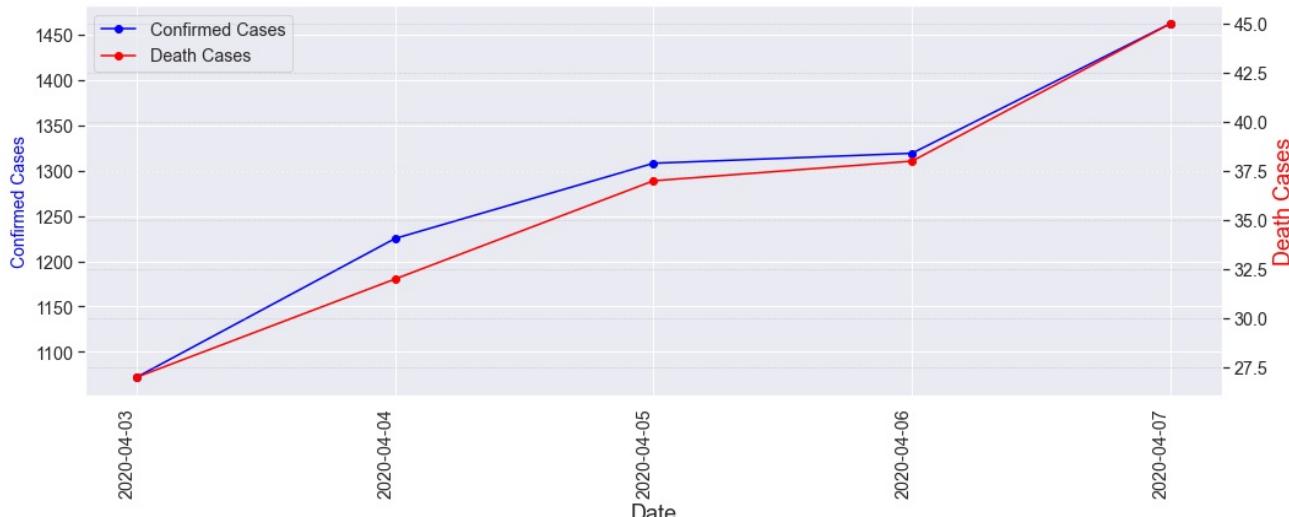
## US's Growth Curve Alignment with COVID-19 Death Toll of At Least 25 Cases



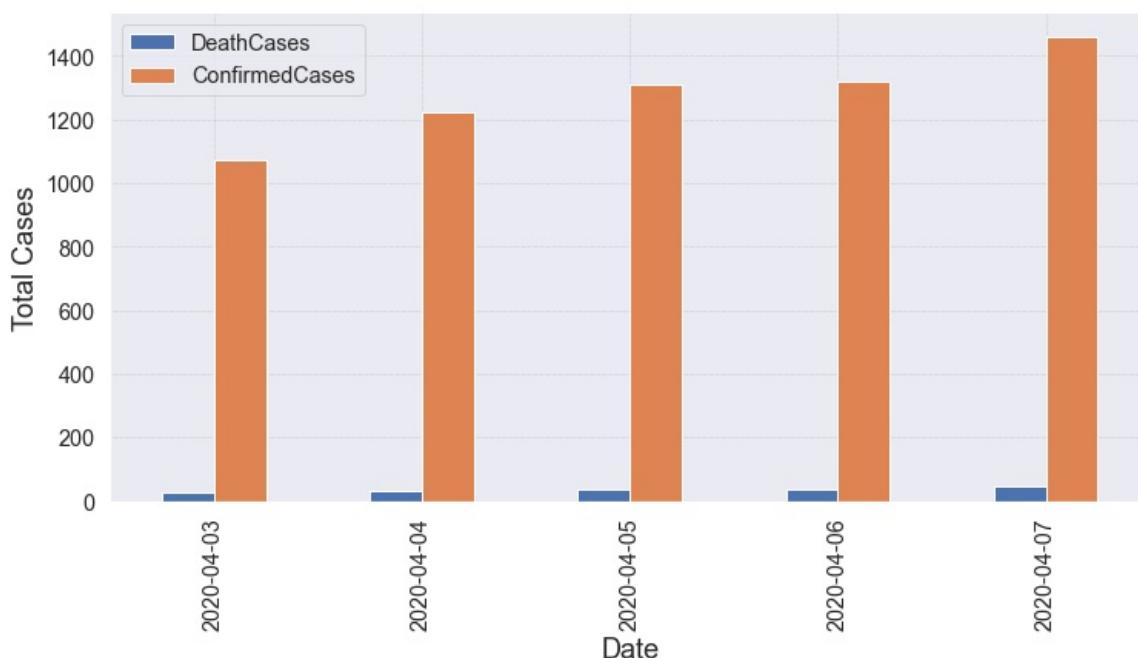
## Daily Cases and Death Count of COVID-19 from January - April in US



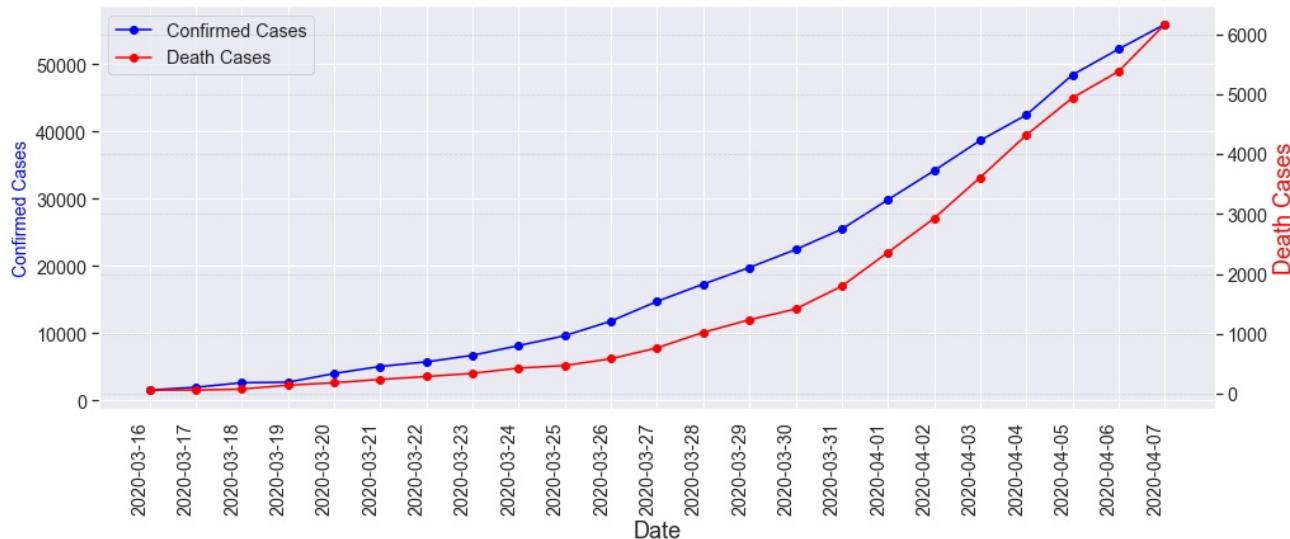
## Ukraine's Growth Curve Alignment with COVID-19 Death Toll of At Least 25 Cases



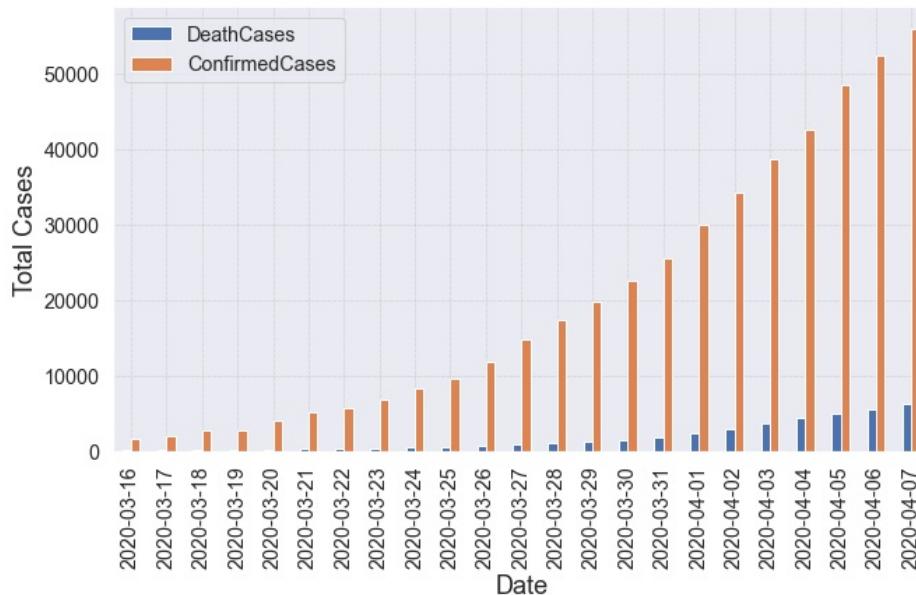
## Daily Cases and Death Count of COVID-19 from January - April in Ukraine



## United Kingdom's Growth Curve Alignment with COVID-19 Death Toll of At Least 25 Cases



## Daily Cases and Death Count of COVID-19 from January - April in United Kingdom



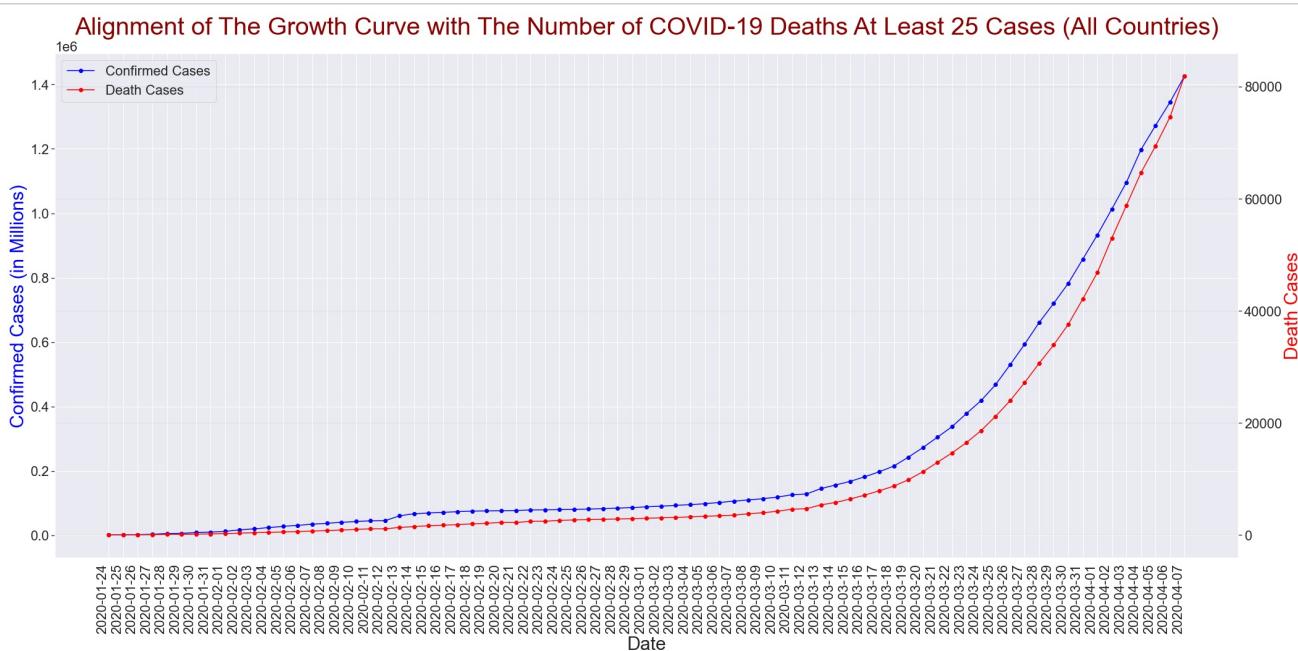
In [30]:

```
sns.set(font_scale = 2)
fig, ax = plt.subplots()
ax2=ax.twinx()
ax.plot(all_ctry['Date'], all_ctry['ConfirmedCases'], color="blue", marker="o", label="Confirmed Cases")
ax.set_xlabel("Date", fontsize=30)
ax.set_ylabel("Confirmed Cases (in Millions)", color="blue", fontsize=30)

ax2.plot(all_ctry['Date'], all_ctry['DeathCases'], color="red", marker="o", label="Death Cases")
ax2.set_ylabel("Death Cases", color="red", fontsize=30)

lines, labels = ax.get_legend_handles_labels()
lines2, labels2 = ax2.get_legend_handles_labels()
ax.legend(lines + lines2, labels + labels2, loc=0)

plt.title("Alignment of The Growth Curve with The Number of COVID-19 Deaths At Least 25 Cases (All Countries)",
          loc='center', pad=30, fontsize=40, color='darkred')
ax.set_xticklabels(all_ctry['Date'], rotation=90, ha='right')
plt.grid(color='darkgray', linestyle=':', linewidth=0.5)
plt.gcf().set_size_inches(30,15)
fig.tight_layout()
plt.show()
```



#### Hasil Analisis:

Jika dilihat dari grafik masing-masing negara, ditemukan bahwa kasus perkembangan COVID-19 dan kasus kematian di setiap negara memiliki korelasi yang tinggi. Rata-rata, setiap negara menghasilkan grafik yang serupa antara kurva pertumbuhan dan kurva kematian. Hal ini menunjukkan bahwa apabila kasus yang terkonfirmasi meningkat, maka, angka kematian juga akan meningkat. Sebaliknya, jika kasus pertumbuhan mengalami penurunan, maka, kasus kematian juga akan berkurang. Untuk mendukung hasil analisis, tersedia pula grafik total kasus yang dihitung dari jumlah kasus seluruh negara. Grafik tersebut juga menarik kesimpulan yang sama, dimana kedua grafik memiliki pola dan bentuk yang mirip, yang berarti bahwa angka pertumbuhan COVID-19 sangat mempengaruhi angka kematian pada suatu negara.

## 5. Visualisasi jumlah pasien yang sembuh (recovered people) (Bobot: 25%)

Anda perlu mendapatkan informasi mengenai total pasien COVID-19 yang dinyatakan sembuh seperti pada kasus terkonfirmasi dan kasus kematian tapi, kali ini lakukan hal tersebut dengan membuat user-defined functions yang diperlukan bukan mengetik ulang barisan code yang sama seperti di nomor sebelumnya.

In [31]:

```
# check duplicates
print(raw_data_recovered.shape)
print(raw_data_recovered[raw_data_recovered.duplicated()].shape)
```

(265, 725)  
(0, 725)

In [32]:

```
#type your function codes here
def newdf(idx, col, values):
    table = pd.crosstab(index=idx,
                          columns=col,
                          values=values,
                          aggfunc='sum',
                          margins=True,
                          margins_name="TotalCases")
    return table

def dropCol(df):
    return df.drop(columns="TotalCases")

def dropIdx(df):
    return df.drop(index="TotalCases")

def dropTC(df):
    return df.drop(index="TotalCases", columns="TotalCases")

def fixplot():
    sns.set(font_scale = 1.3)
    plt.xlabel('Date', fontsize = 18)
    plt.ylabel('Total Cases', fontsize = 18)
    plt.grid(color='darkgray', linestyle=':', linewidth=0.5)
    plt.gcf().set_size_inches(15,7)
    plt.tight_layout()
    plt.show()

def c6plotting(total_recovered, country):
    total_recovered = newdf(rec_ctry["Yearmonth"], "Total", rec_ctry[country])
    total_recovered = dropTC(total_recovered).reset_index()
    total_recovered["Yearmonth"] = total_recovered["Yearmonth"].astype(str)

    sns.lineplot(data=total_recovered,
                 x="Yearmonth",
                 y="Total")
    plt.title('Total Recovered Cases of COVID19 in {} (in Thousands)'.format(country),
              loc='center', pad=30, fontsize=25, color='darkred')
    plt.xticks(rotation=90)
    plt.ylim(ymin=0)
    labels, locations = plt.yticks()
    plt.yticks(labels, (labels/1000).astype(int))
    fixplot()

    sns.barplot(data=total_recovered,
                 x="Yearmonth",
                 y="Total")
    plt.xticks(rotation=90)
    plt.ylim(ymin=0)
    labels, locations = plt.yticks()
    plt.yticks(labels, (labels/1000).astype(int))
    fixplot()
```

In [33]:

```
rdf = raw_data_recovered.melt(id_vars=["Province/State", "Country/Region", "Lat", "Long"],
                               var_name="Date",
                               value_name="Cases")
rdf["Date"] = pd.to_datetime(rdf["Date"])
rdf["Month"] = rdf["Date"].dt.month_name()
rdf["Yearmonth"] = rdf["Date"].dt.to_period('M')
rdf["Date"] = rdf["Date"].dt.to_period('D')
rdf.head()
```

Out[33]:

	Province/State	Country/Region	Lat	Long	Date	Cases	Month	Yearmonth
0	NaN	Afghanistan	33.93911	67.709953	2020-01-22	0	January	2020-01
1	NaN	Albania	41.15330	20.168300	2020-01-22	0	January	2020-01
2	NaN	Algeria	28.03390	1.659600	2020-01-22	0	January	2020-01
3	NaN	Andorra	42.50630	1.521800	2020-01-22	0	January	2020-01
4	NaN	Angola	-11.20270	17.873900	2020-01-22	0	January	2020-01

Periksa fungsi-fungsi yang sudah anda buat

In [34]:

```
# write your codes to check your functions here
```

```
#data per hari
```

```
recovered_country = newdf(rdf["Date"], rdf["Country/Region"], rdf["Cases"])
recovered_country
```

Out[34]:

Country/Region	Afghanistan	Albania	Algeria	Andorra	Angola	Antigua and Barbuda	Argentina	Armenia	Australia	Austria	...	Uruguay
Date												
2020-01-22	0	0	0	0	0	0	0	0	0	0	...	0
2020-01-23	0	0	0	0	0	0	0	0	0	0	...	0
2020-01-24	0	0	0	0	0	0	0	0	0	0	...	0
2020-01-25	0	0	0	0	0	0	0	0	0	0	...	0
2020-01-26	0	0	0	0	0	0	0	0	0	0	...	0
...	...	...	...	...	...	...	...	...	...	...	...	...
2022-01-08	0	0	0	0	0	0	0	0	0	0	...	0
2022-01-09	0	0	0	0	0	0	0	0	0	0	...	0
2022-01-10	0	0	0	0	0	0	0	0	0	0	...	0
2022-01-11	0	0	0	0	0	0	0	0	0	0	...	0
TotalCases	18289880	22652652	25670233	3085649	5895800	191813	711610324	51634388	8821094	131744371	...	36788937

722 rows × 197 columns

In [35]:

```
#data per bulan
```

```
recovered_country_m = newdf(rdf["Yearmonth"], rdf["Country/Region"], rdf["Cases"])
recovered_country_m
```

Out[35]:

Country/Region	Afghanistan	Albania	Algeria	Andorra	Angola	Antigua and Barbuda	Argentina	Armenia	Australia	Austria	...	Uruguay
Yearmonth												
2020-01	0	0	0	0	0	0	0	0	4	0	...	0
2020-02	0	0	0	0	0	0	0	0	195	0	...	0
2020-03	26	241	594	37	1	0	875	205	2307	2883	...	41
2020-04	2927	7902	24072	5636	129	125	19590	12116	91483	241616	...	7760
2020-05	24129	21760	109804	18505	466	561	84929	63541	195488	447388	...	17353
2020-06	184768	32742	233399	23239	1668	629	335157	222445	205100	481557	...	23361
2020-07	688117	67686	467472	24877	5708	1551	1556787	671719	256329	539229	...	28141
2020-08	843438	120294	826991	26677	22283	2558	6029870	1059630	381635	645006	...	37229
2020-09	949104	200469	1026971	32128	41899	2743	13413943	1251850	614861	851070	...	47202
2020-10	1039280	305370	1177057	69118	94176	3163	24249610	1490885	679910	1521700	...	65274
2020-11	1058724	424181	1373852	143348	196092	3745	33912316	2395256	669764	4042516	...	100833
2020-12	1207801	819127	1902624	209606	286751	4343	41776579	3964417	697478	8916607	...	242149
2021-01	1403684	1250570	2181520	253747	473400	4906	48805346	4695829	703156	11328881	...	730662
2021-02	1358287	1622660	2126923	277194	527106	5943	51092582	4507424	640112	11533469	...	1204812
2021-03	1541980	2525857	2481861	336906	618101	18051	62045827	5186198	711784	14360090	...	1982760
2021-04	1566429	3010039	2495622	361236	668606	28597	70049590	5573536	694114	16404817	...	3752012
2021-05	1710784	3795097	2710001	407093	802500	35462	92046916	6432713	727969	18990884	...	6517701
2021-06	1887517	3894898	2805260	408532	924203	36515	112898859	6461740	710896	19029231	...	9204071
2021-07	2492541	4032655	3257465	430536	1077983	37977	134893683	6764038	741869	19831920	...	11331854
2021-08	330344	521104	468745	57234	154728	4944	18397865	880846	96640	2575507	...	1495722
2021-09	0	0	0	0	0	0	0	0	0	0	...	0
2021-10	0	0	0	0	0	0	0	0	0	0	...	0
2021-11	0	0	0	0	0	0	0	0	0	0	...	0
2021-12	0	0	0	0	0	0	0	0	0	0	...	0
2022-01	0	0	0	0	0	0	0	0	0	0	...	0
TotalCases	18289880	22652652	25670233	3085649	5895800	191813	711610324	51634388	8821094	131744371	...	36788937

26 rows × 197 columns

Selanjutnya, tampilkan data total pasien COVID-19 yang sembuh dalam bentuk grafik yang sesuai. Berikan judul, labels, dan spesifikasi (ukuran, warna, ketebalan, dll) yang sesuai sehingga plot yang dihasilkan rapi, menarik, dan mudah dipahami.

In [36]:

```
#type your codes here
recovered = dropIdx(recovered_country_m).reset_index()
recovered
```

Out[36]:

Country/Region	Yearmonth	Afghanistan	Albania	Algeria	Andorra	Angola	Antigua and Barbuda	Argentina	Armenia	Australia	...	Uruguay	U...
0	2020-01	0	0	0	0	0	0	0	0	4	...	0	
1	2020-02	0	0	0	0	0	0	0	0	195	...	0	
2	2020-03	26	241	594	37	1	0	875	205	2307	...	41	
3	2020-04	2927	7902	24072	5636	129	125	19590	12116	91483	...	7760	
4	2020-05	24129	21760	109804	18505	466	561	84929	63541	195488	...	17353	
5	2020-06	184768	32742	233399	23239	1668	629	335157	222445	205100	...	23361	
6	2020-07	688117	67686	467472	24877	5708	1551	1556787	671719	256329	...	28141	
7	2020-08	843438	120294	826991	26677	22283	2558	6029870	1059630	381635	...	37229	
8	2020-09	949104	200469	1026971	32128	41899	2743	13413943	1251850	614861	...	47202	
9	2020-10	1039280	305370	1177057	69118	94176	3163	24249610	1490885	679910	...	65274	
10	2020-11	1058724	424181	1373852	143348	196092	3745	33912316	2395256	669764	...	100833	
11	2020-12	1207801	819127	1902624	209606	286751	4343	41776579	3964417	697478	...	242149	
12	2021-01	1403684	1250570	2181520	253747	473400	4906	48805346	4695829	703156	...	730662	
13	2021-02	1358287	1622660	2126923	277194	527106	5943	51092582	4507424	640112	...	1204812	
14	2021-03	1541980	2525857	2481861	336906	618101	18051	62045827	5186198	711784	...	1982760	
15	2021-04	1566429	3010039	2495622	361236	668606	28597	70049590	5573536	694114	...	3752012	
16	2021-05	1710784	3795097	2710001	407093	802500	35462	92046916	6432713	727969	...	6517701	
17	2021-06	1887517	3894898	2805260	408532	924203	36515	112898859	6461740	710896	...	9204071	
18	2021-07	2492541	4032655	3257465	430536	1077983	37977	134893683	6764038	741869	...	11331854	
19	2021-08	330344	521104	468745	57234	154728	4944	18397865	880846	96640	...	1495722	
20	2021-09	0	0	0	0	0	0	0	0	0	...	0	
21	2021-10	0	0	0	0	0	0	0	0	0	...	0	
22	2021-11	0	0	0	0	0	0	0	0	0	...	0	
23	2021-12	0	0	0	0	0	0	0	0	0	...	0	
24	2022-01	0	0	0	0	0	0	0	0	0	...	0	

25 rows × 198 columns

In [37]:

```
total_recovered = newdf(recovered[ "Yearmonth" ], "Total", recovered[ "TotalCases" ])
total_recovered = dropTC(total_recovered).reset_index()
total_recovered[ "Yearmonth" ] = total_recovered[ "Yearmonth" ].astype(str)
total_recovered
```

Out[37]:

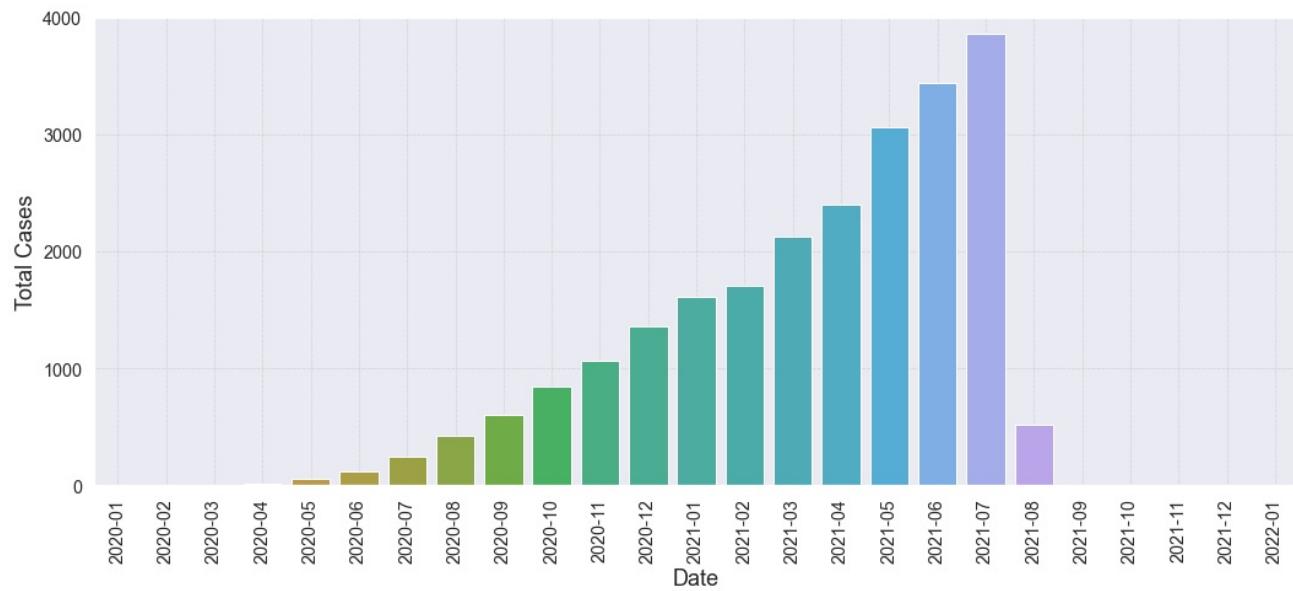
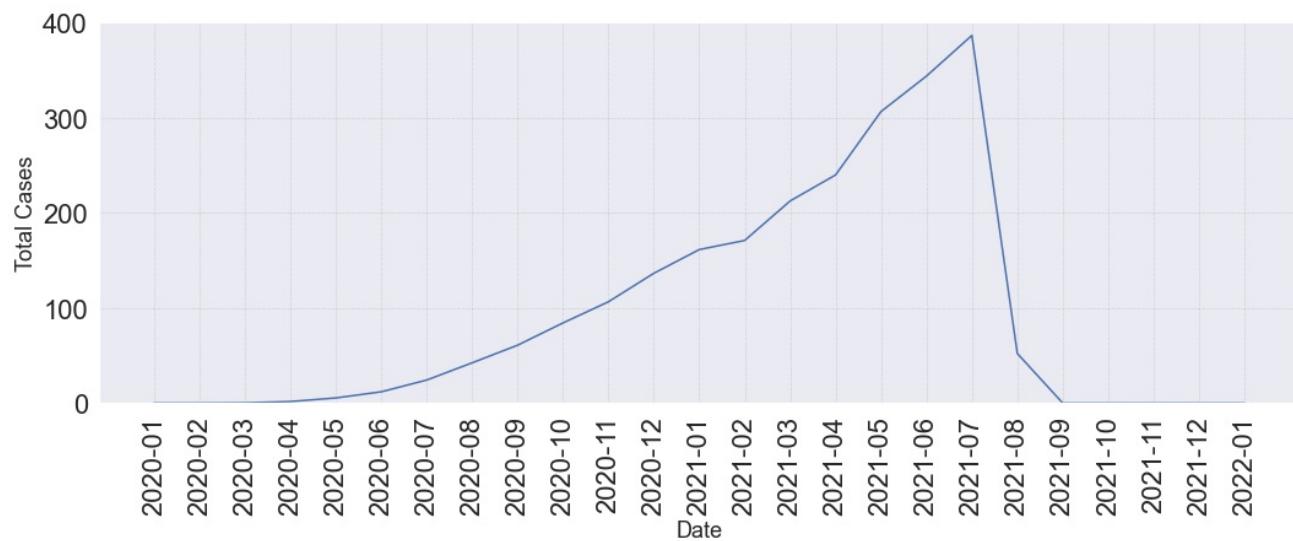
col_0	Yearmonth	Total
0	2020-01	869
1	2020-02	380884
2	2020-03	2705110
3	2020-04	16314536
4	2020-05	54100868
5	2020-06	118880540
6	2020-07	242238461
7	2020-08	423525878
8	2020-09	608664203
9	2020-10	843271736
10	2020-11	1065636228
11	2020-12	1365751320
12	2021-01	1615078084
13	2021-02	1711133925
14	2021-03	2126494619
15	2021-04	2400178967
16	2021-05	3066997601
17	2021-06	3440040018
18	2021-07	3868309116
19	2021-08	521752984
20	2021-09	0
21	2021-10	0
22	2021-11	0
23	2021-12	0
24	2022-01	0

In [38]:

```
sns.lineplot(data=total_recovered,
              x="Yearmonth",
              y="Total")
plt.title('Total Recovered Cases of COVID19 (in Millions)',
          loc='center', pad=30, fontsize=25, color='darkred')
plt.xticks(rotation=90)
plt.ylim(ymax=4000000000)
labels, locations = plt.yticks()
plt.yticks(labels, (labels/10000000).astype(int))
fixplot()

sns.barplot(data=total_recovered,
             x="Yearmonth",
             y="Total")
plt.xticks(rotation=90)
plt.ylim(ymax=4000000000)
labels, locations = plt.yticks()
plt.yticks(labels, (labels/1000000).astype(int))
fixplot()
```

Total Recovered Cases of COVID19 (in Millions)



#### Hasil Analisis:

Berdasarkan hasil visualisasi di atas, diketahui bahwa total kasus kesembuhan di seluruh negara mulai meningkat pada bulan Mei 2020 hingga Juli 2021. Hal ini memberikan pertanda baik karena banyak masyarakat yang mengalami kesembuhan. Tetapi, dapat diasumsikan pula, bahwa kasus kematian lebih besar daripada kasus kesembuhan, yang secara keseluruhan berarti dapat memberikan pengaruh buruk. Selanjutnya, kasus kesembuhan kembali menurun pada bulan Agustus 2021. Dengan adanya grafik penurunan yang jatuh secara drastis ini, saya juga menemukan 2 asumsi. Pertama, dapat diasumsikan bahwa kasus terkonfirmasi dari COVID-19 sudah membaik, sehingga semakin dikit masyarakat yang mengalami kesembuhan akibat penurunan kasus. Namun, pada kondisi kedua, dapat diasumsikan bahwa kasus COVID-19 semakin memburuk dan kasus kematian yang terjadi lebih banyak dibandingkan kasus kesembuhannya. Dengan adanya 2 asumsi ini, saya tidak dapat menarik kesimpulan penuh hanya dengan melihat data dari kasus kesembuhan pada bulan Januari 2020 - Januari 2022.

Lakukan visualisasi yang sama pada beberapa negara berikut (France, Spain, China, US, Italy, and Australia). Berikan judul, labels, dan spesifikasi (ukuran, warna, ketebalan, dll) yang sesuai sehingga plot yang dihasilkan rapi, menarik, dan mudah dipahami.

In [39]:

```
#type your codes here
rec_ctry = dropTC(recovered_country_m).reset_index()
rec_ctry
```

Out[39]:

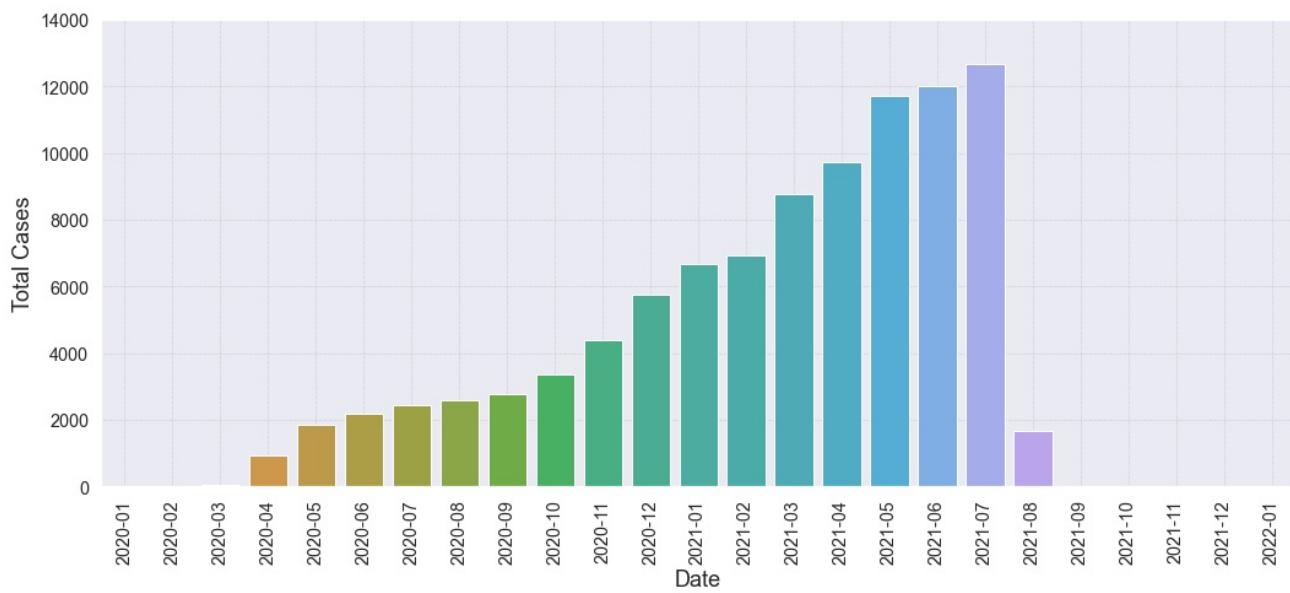
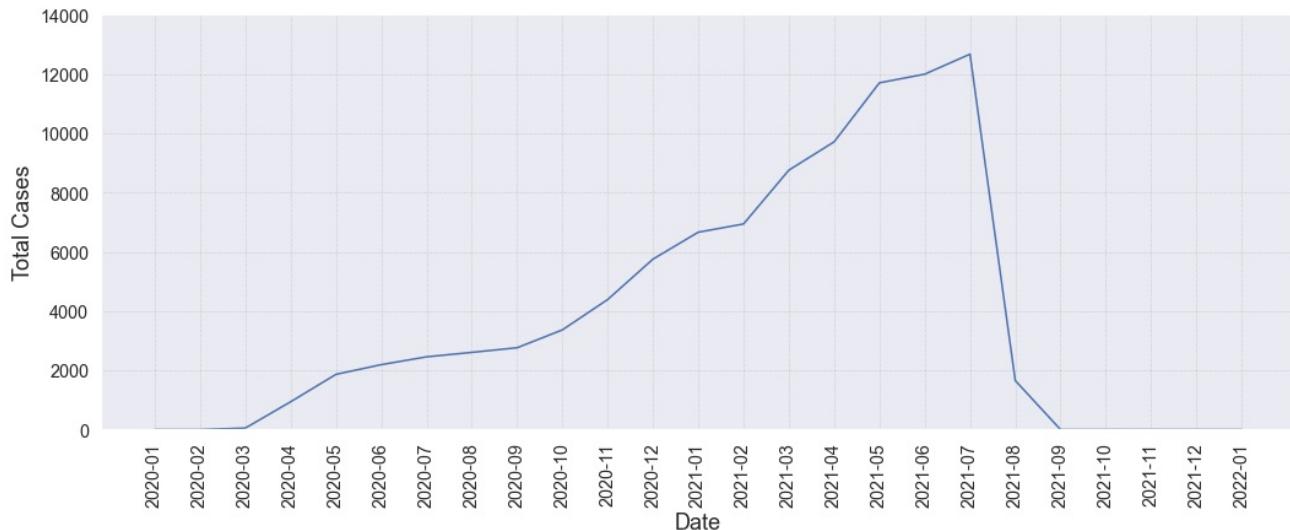
Country/Region	Yearmonth	Afghanistan	Albania	Algeria	Andorra	Angola	Antigua and Barbuda	Argentina	Armenia	Australia	...	United Kingdom
0	2020-01	0	0	0	0	0	0	0	0	0	4	0
1	2020-02	0	0	0	0	0	0	0	0	0	195	...
2	2020-03	26	241	594	37	1	0	875	205	2307	...	1929
3	2020-04	2927	7902	24072	5636	129	125	19590	12116	91483	...	14880
4	2020-05	24129	21760	109804	18505	466	561	84929	63541	195488	...	32790
5	2020-06	184768	32742	233399	23239	1668	629	335157	222445	205100	...	38818
6	2020-07	688117	67686	467472	24877	5708	1551	1556787	671719	256329	...	43441
7	2020-08	843438	120294	826991	26677	22283	2558	6029870	1059630	381635	...	46765
8	2020-09	949104	200469	1026971	32128	41899	2743	13413943	1251850	614861	...	62290
9	2020-10	1039280	305370	1177057	69118	94176	3163	24249610	1490885	679910	...	79916
10	2020-11	1058724	424181	1373852	143348	196092	3745	33912316	2395256	669764	...	95104
11	2020-12	1207801	819127	1902624	209606	286751	4343	41776579	3964417	697478	...	132440
12	2021-01	1403684	1250570	2181520	253747	473400	4906	48805346	4695829	703156	...	242466
13	2021-02	1358287	1622660	2126923	277194	527106	5943	51092582	4507424	640112	...	303091
14	2021-03	1541980	2525857	2481861	336906	618101	18051	62045827	5186198	711784	...	376948
15	2021-04	1566429	3010039	2495622	361236	668606	28597	70049590	5573536	694114	...	415975
16	2021-05	1710784	3795097	2710001	407093	802500	35462	92046916	6432713	727969	...	471500
17	2021-06	1887517	3894898	2805260	408532	924203	36515	112898859	6461740	710896	...	468037
18	2021-07	2492541	4032655	3257465	430536	1077983	37977	134893683	6764038	741869	...	528517
19	2021-08	330344	521104	468745	57234	154728	4944	18397865	880846	96640	...	92778
20	2021-09	0	0	0	0	0	0	0	0	0	...	0
21	2021-10	0	0	0	0	0	0	0	0	0	...	0
22	2021-11	0	0	0	0	0	0	0	0	0	...	0
23	2021-12	0	0	0	0	0	0	0	0	0	...	0
24	2022-01	0	0	0	0	0	0	0	0	0	...	0

25 rows x 197 columns

In [40]:

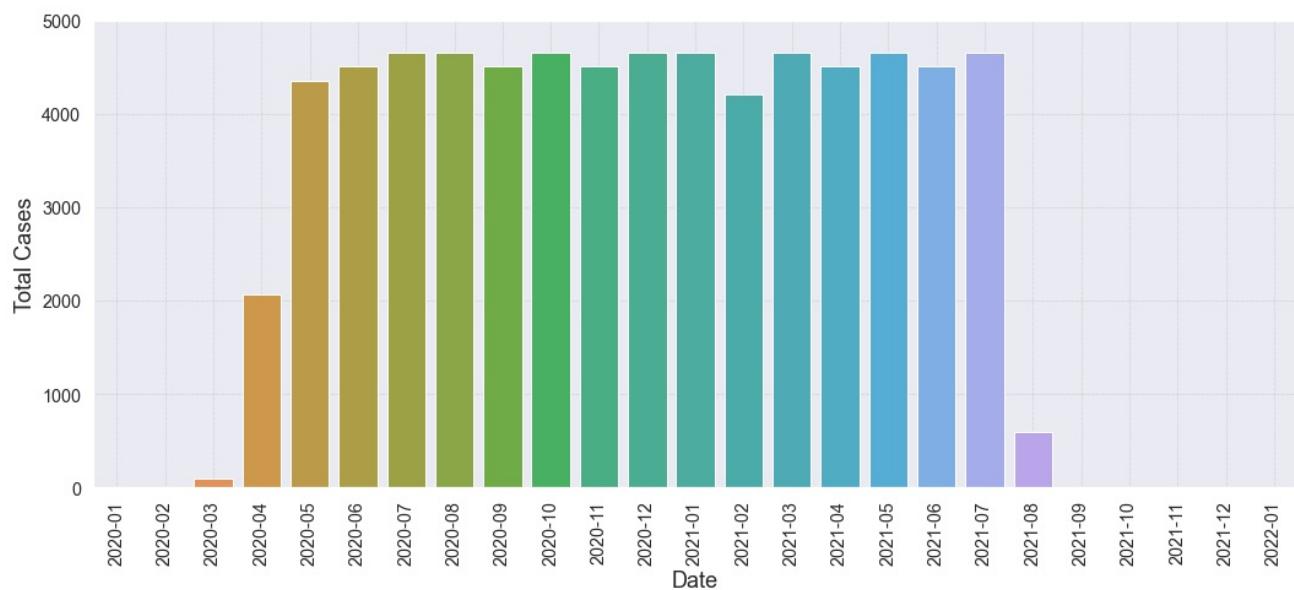
```
c6plotting("France", "France")
c6plotting("Spain", "Spain")
c6plotting("China", "China")
c6plotting("US", "US")
c6plotting("Italy", "Italy")
c6plotting("Australia", "Australia")
```

### Total Recovered Cases of COVID19 in France (in Thousands)

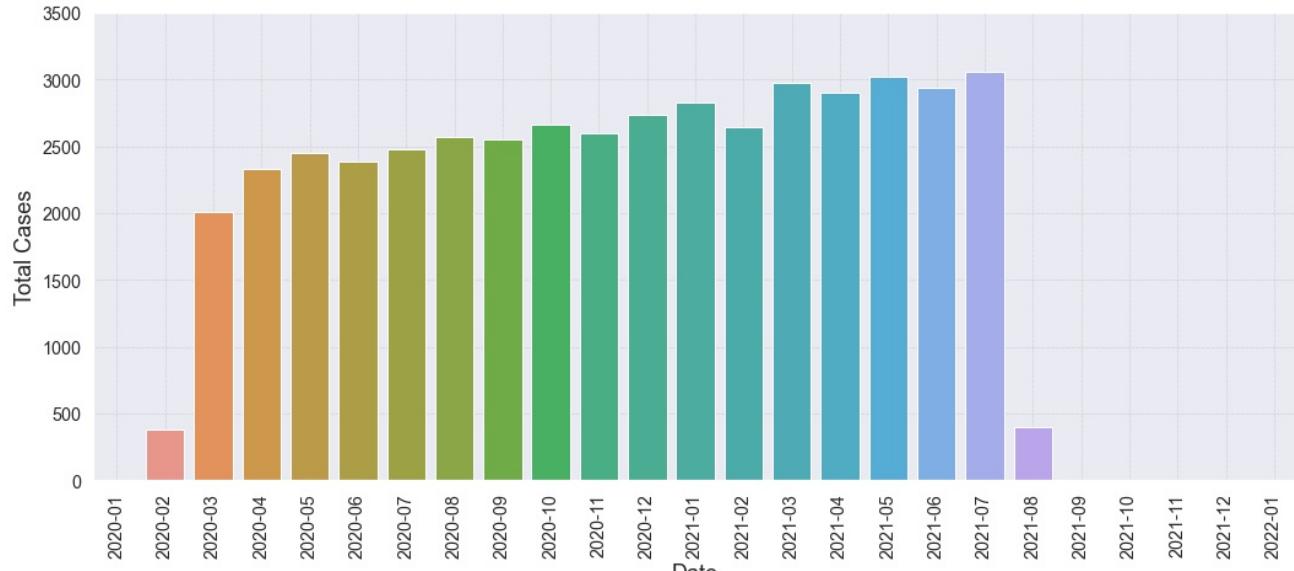
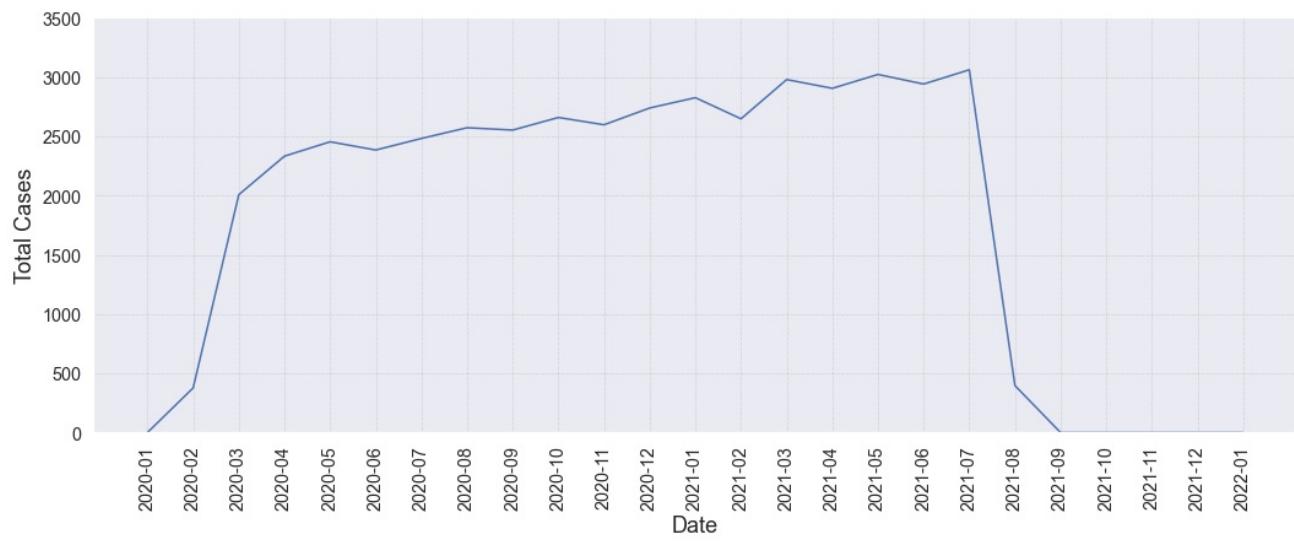


### Total Recovered Cases of COVID19 in Spain (in Thousands)

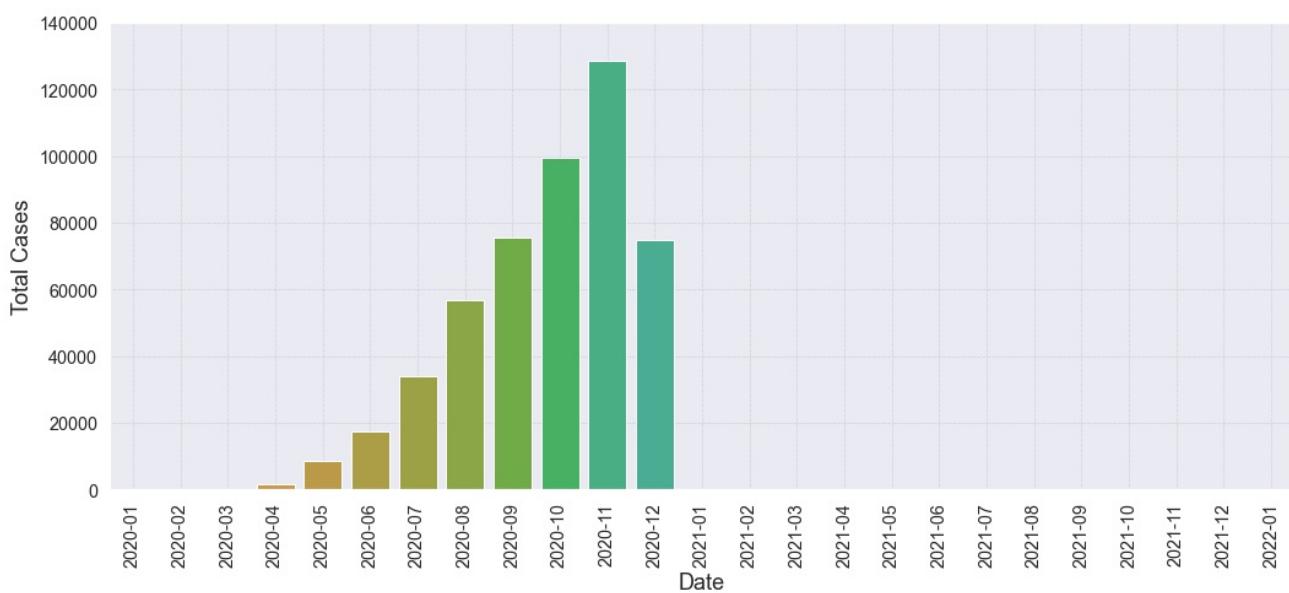
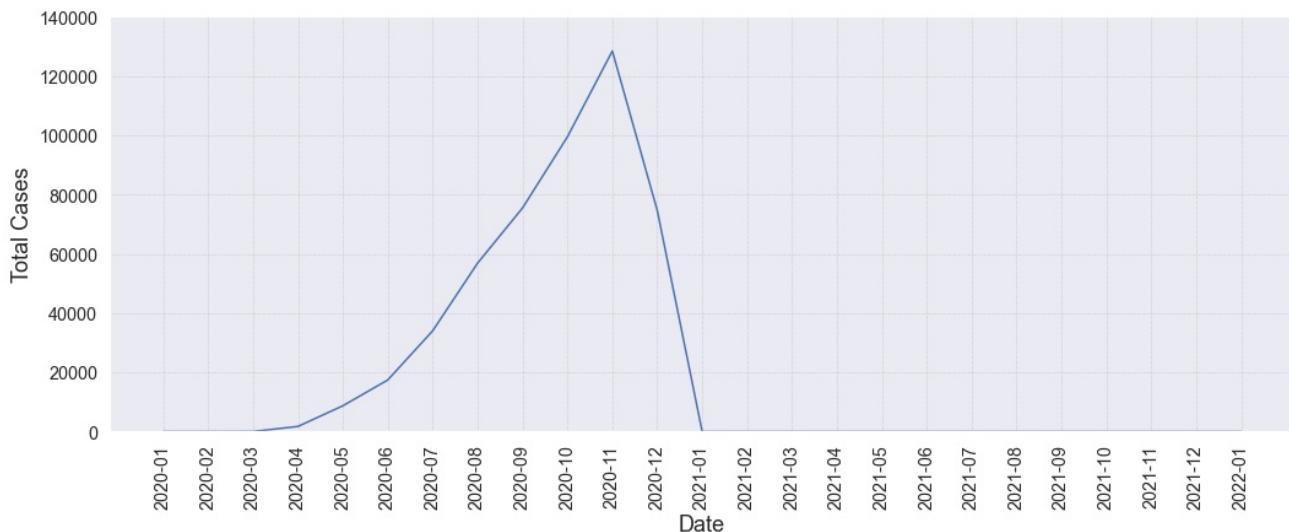




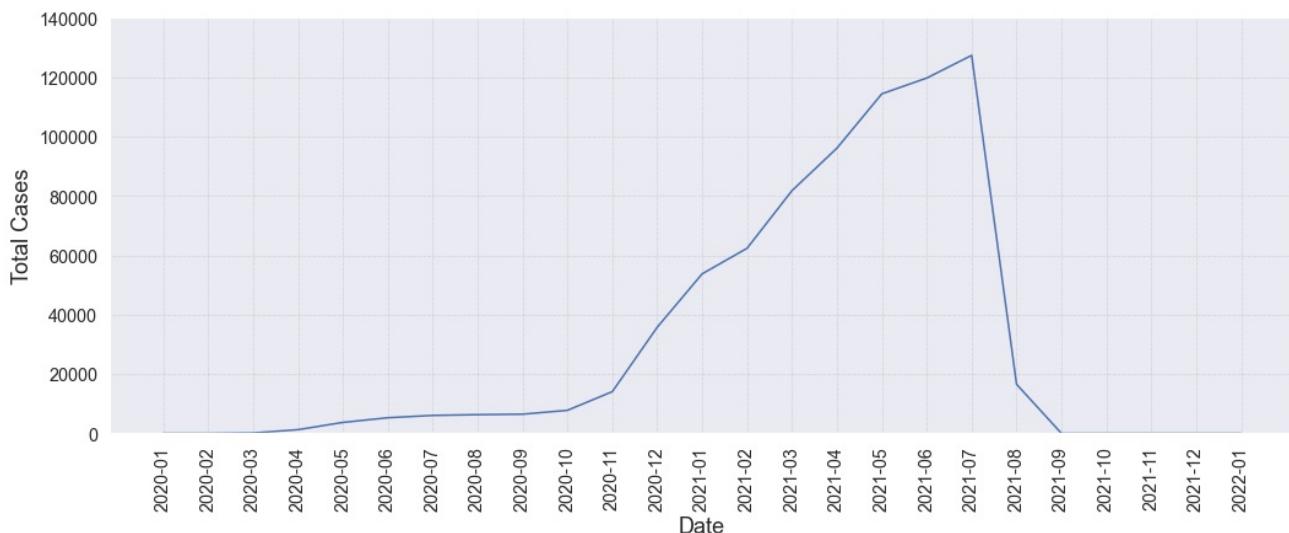
**Total Recovered Cases of COVID19 in China (in Thousands)**

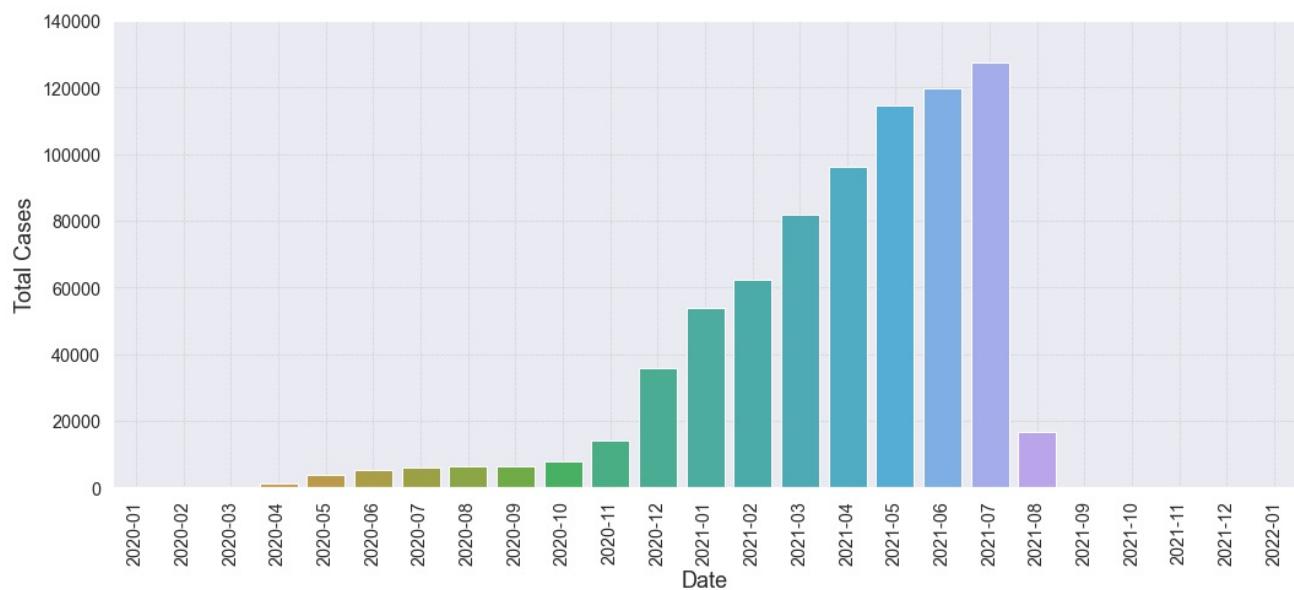


### Total Recovered Cases of COVID19 in US (in Thousands)

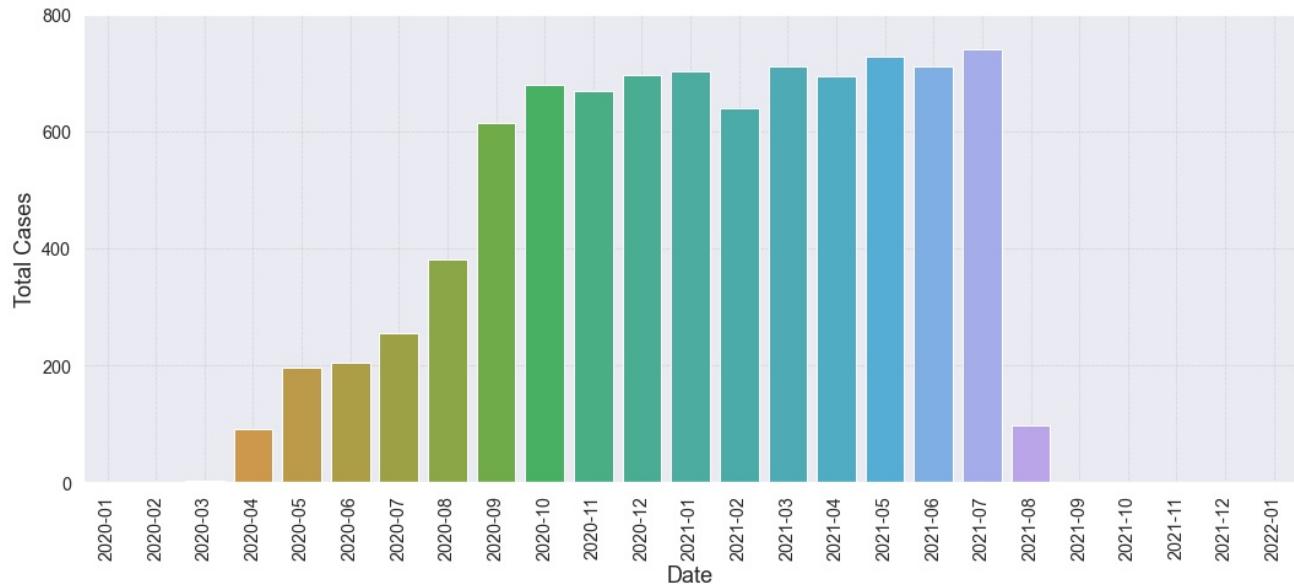
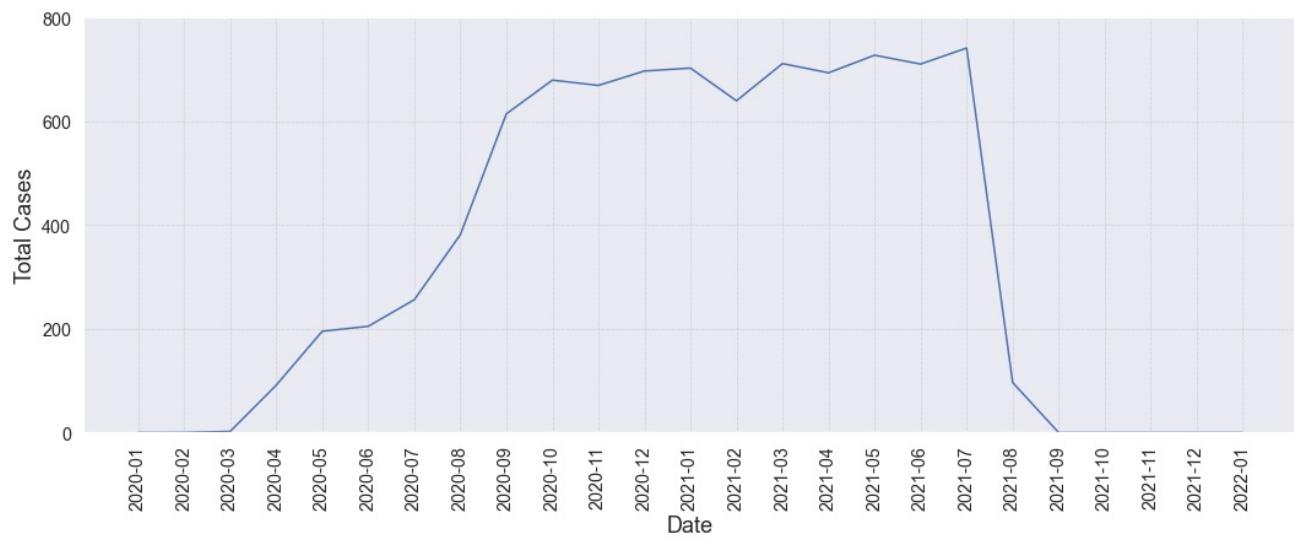


### Total Recovered Cases of COVID19 in Italy (in Thousands)





**Total Recovered Cases of COVID19 in Australia (in Thousands)**

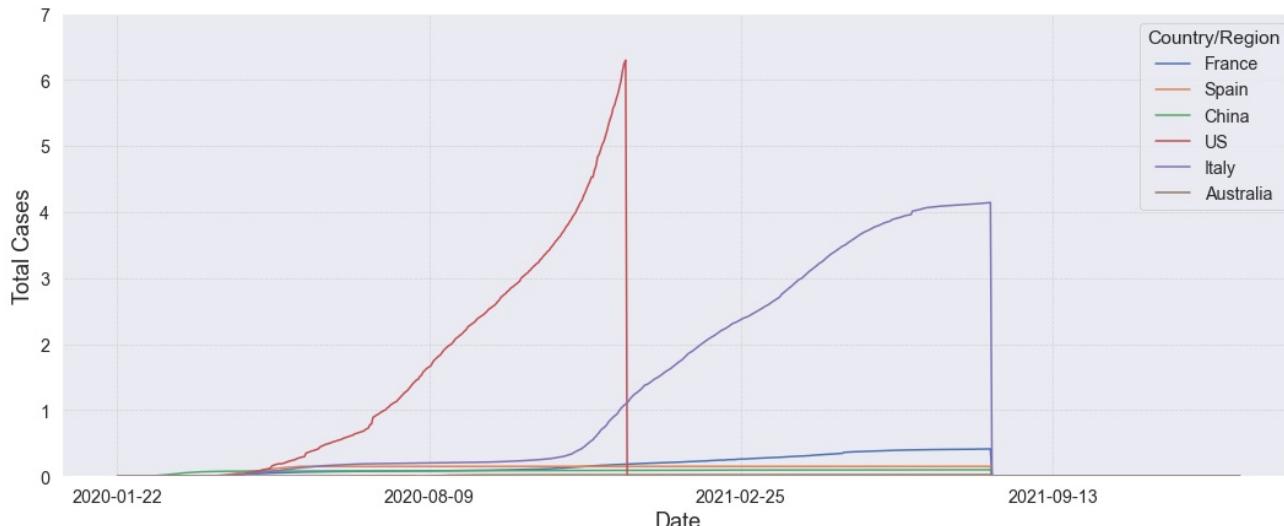


In [41]:

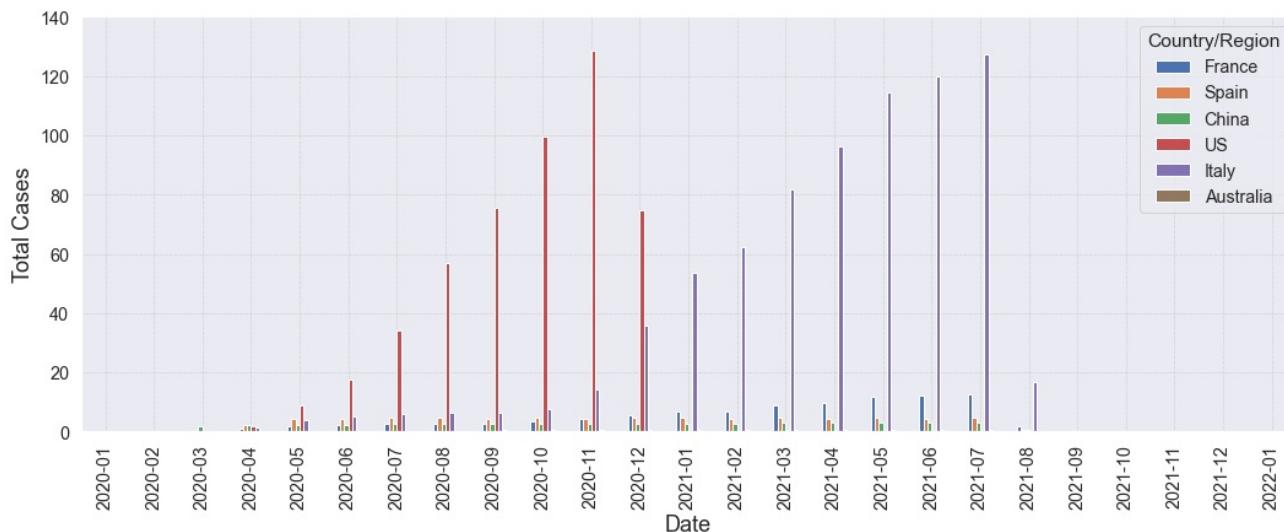
```
dropTC(recovered_country).loc[:, ['France', 'Spain', 'China', 'US', 'Italy', 'Australia']].plot()
plt.title('Total Recovered Daily Cases of COVID19 in 6 Countries (in Millions)',
          loc='center', pad=30, fontsize=25, color='darkred')
plt.ylim(ymin=0)
labels, locations = plt.yticks()
plt.yticks(labels, (labels/1000000).astype(int))
fixplot()

dropTC(recovered_country_m).loc[:, ['France', 'Spain', 'China', 'US', 'Italy', 'Australia']].plot.bar()
plt.title('Total Recovered Monthly Cases of COVID19 in 6 Countries (in Millions)',
          loc='center', pad=30, fontsize=25, color='darkred')
plt.ylim(ymin=0)
labels, locations = plt.yticks()
plt.yticks(labels, (labels/1000000).astype(int))
fixplot()
```

Total Recovered Daily Cases of COVID19 in 6 Countries (in Millions)



Total Recovered Monthly Cases of COVID19 in 6 Countries (in Millions)



#### Hasil Analisis:

Dari grafik-grafik di atas, diketahui bahwa Amerika Serikat memiliki kasus kesembuhan tertinggi, dengan puncaknya pada bulan November 2020. Sedangkan, negara yang memiliki kasus kesembuhan tertinggi kedua adalah Italia, yang terjadi pada bulan Juli 2021. Namun, di saat keduanya sedang mengalami kesembuhan sebanyak-banyaknya, pada bulan selanjutnya, kasus kesembuhan menurun secara drastis dan penurunan ini terus berlanjut hingga pada bulan Januari 2022. Sedangkan, kasus kesembuhan paling sedikit ada pada negara Australia dengan total kesembuhan hampir 800.000 kasus pada bulan Juli 2021. Dengan ini, dapat disimpulkan bahwa rata-rata kasus kesembuhan terus terjadi selama kurang lebih 1 tahun (Maret 2020-Agustus 2021).

In [42]:

```
"""
# VISUALISASI TAMBAHAN (Choropleth)
rdf_map = dropTC(recovered_country).loc[:, ['France', 'Spain', 'China', 'US', 'Italy', 'Australia']]
rdf_map = rdf_map.reset_index().melt(id_vars="Date")
rdf_map['Date'] = rdf_map['Date'].astype(str)
rdf_map = rdf_map.rename(columns={'value': 'RecoveredCases'})
rdf_map = rdf_map.groupby(['Date', 'Country/Region'], as_index=False)['RecoveredCases'].sum()

rdf_map['iso_alpha'] = rdf_map.apply(lambda x: obj.fetch_iso3(x['Country/Region']), axis=1)
rdf_map['log(RecoveredCases)'] = np.log(rdf_map.RecoveredCases + 1)

px.choropleth(rdf_map,
              locations="iso_alpha",
              color="log(RecoveredCases)",
              hover_name="Country/Region",
              hover_data=["RecoveredCases"] ,
              animation_frame="Date",
              color_continuous_scale=px.colors.sequential.dense,
              title='Total Recovered Cases Growth of COVID-19 (Logarithmic Scale)')
"""

Out[42]:
```

```
'n# VISUALISASI TAMBAHAN (Choropleth)nrdf_map = dropTC(recovered_country).loc[:, ['France', 'Spain', 'China', 'US', 'Italy', 'Australia']]nrdf_map = rdf_map.reset_index().melt(id_vars="Date")nrdf_map['Date'] = rdf_map['Date'].astype(str)nrdf_map = rdf_map.rename(columns={'value': 'RecoveredCases'})nrdf_map = rdf_map.groupby(['Date', 'Country/Region'], as_index=False)[['RecoveredCases']].sum()nrdf_map['iso_alpha'] = rdf_map.apply(lambda x: obj.fetch_iso3(x['Country/Region']), axis=1)nrdf_map['log(RecoveredCases)'] = np.log(rdf_map.RecoveredCases + 1)npx.choropleth(rdf_map, n locations="iso_alpha", n color="log(RecoveredCases)", n hover_name="Country/Region", n hover_data=["RecoveredCases"] ,n animation_frame="Date", n color_continuous_scale=px.colors.sequential.dense, n title='Total Recovered Cases Growth of COVID-19 (Logarithmic Scale)')'
```