UAS - INTRODUCTION TO DATA SCIENCE - STAT6152016

Format nama files: nim_nama.ipynb; nim_nama.pdf; nim_nama.rar

NIM: 2540118656 Nama: Rafael Nicholas Tanaja

Link Video: https://drive.google.com/file/d/1FvYtRQ9NlniCUPvG6atGTMOqCqEWwpB-/view?usp=sharing

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
%matplotlib notebook
```

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 visualisai 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
0	NaN	Afghanistan	33.0000	65.0000	0	0	0	0	0	0	...	120	170
1	NaN	Albania	41.1533	20.1683	0	0	0	0	0	0	...	212	223
2	NaN	Algeria	28.0339	1.6596	0	0	0	0	0	0	...	511	584
3	NaN	Andorra	42.5063	1.5218	0	0	0	0	0	0	...	334	370
4	NaN	Angola	-11.2027	17.8739	0	0	0	0	0	0	...	7	7

11.2027

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

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
count	263.000000	263.000000	263.000000	263.000000	263.000000	263.000000	263.000000	263.000000	263.000000	263.
mean	21.339244	22.068133	2.110266	2.486692	3.577947	5.452471	8.053232	11.129278	21.209125	23.
std	24.779585	70.785949	27.434015	27.532888	34.275498	47.702207	66.662110	89.815834	220.427512	221.
min	-51.796300	-135.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.
25%	6.938500	-21.031300	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.
50%	23.634500	20.168300	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.
75%	41.178850	79.500000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.
max	71.706900	178.065000	444.000000	444.000000	549.000000	761.000000	1058.000000	1423.000000	3554.000000	3554.

8 rows × 79 columns

1. Jumlah kasus terkonfirmasi COVID-19 berdasarkan negara (Bobot: 15%)

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]:

```
data_country = raw_data_confirmed.groupby('Country/Region').sum()
data_country
```

Out[6]:

	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	...	3/29/20	3/30/20

Country/Region	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	...	3/29/20	3/30/20
Afghanistan	33.0000	65.0000	0	0	0	0	0	0	0	0	...	120	170
Country/Region													
Albania	41.1533	20.1683	0	0	0	0	0	0	0	0	...	212	223
Algeria	28.0339	1.6596	0	0	0	0	0	0	0	0	...	511	584
Andorra	42.5063	1.5218	0	0	0	0	0	0	0	0	...	334	370
Angola	-11.2027	17.8739	0	0	0	0	0	0	0	0	...	7	7
...	...	...	...	...	...	...	...	...	...	...	...	...	...
Vietnam	16.0000	108.0000	0	2	2	2	2	2	2	2	...	188	203
West Bank and Gaza	31.9522	35.2332	0	0	0	0	0	0	0	0	...	109	116
Western Sahara	24.2155	-12.8858	0	0	0	0	0	0	0	0	...	0	0
Zambia	-15.4167	28.2833	0	0	0	0	0	0	0	0	...	29	35
Zimbabwe	-20.0000	30.0000	0	0	0	0	0	0	0	0	...	7	7

184 rows × 79 columns

In [7]:

```
del data_country['Long']
del data_country['Lat']
```

In [8]:

```
confirmed_country = data_country
confirmed_country = confirmed_country.T
confirmed_country
```

Out[8]:

Country/Region	Afghanistan	Albania	Algeria	Andorra	Angola	Antigua and Barbuda	Argentina	Armenia	Australia	Austria	...	United Arab Emirates
1/22/20	0	0	0	0	0	0	0	0	0	0	...	0
1/23/20	0	0	0	0	0	0	0	0	0	0	...	0
1/24/20	0	0	0	0	0	0	0	0	0	0	...	0
1/25/20	0	0	0	0	0	0	0	0	0	0	...	0
1/26/20	0	0	0	0	0	0	0	0	4	0	...	0
...	...	...	...	...	...	...	...	...	...	...	...	...
4/3/20	281	304	1171	439	8	15	1265	736	5330	11524	...	1264
4/4/20	299	333	1251	466	10	15	1451	770	5550	11781	...	1505
4/5/20	349	361	1320	501	14	15	1451	822	5687	12051	...	1799
4/6/20	367	377	1423	525	16	15	1554	833	5797	12297	...	2076
4/7/20	423	383	1468	545	17	19	1628	853	5895	12639	...	2359

77 rows × 184 columns

Berdasarkan data tersebut bisa dilihat bahwa terdapat 184 negara yang terdapat di data.

2. Visualisasi kasus terkonfirmasi COVID-19 berdasarkan negara (Bobot: 15%)

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.

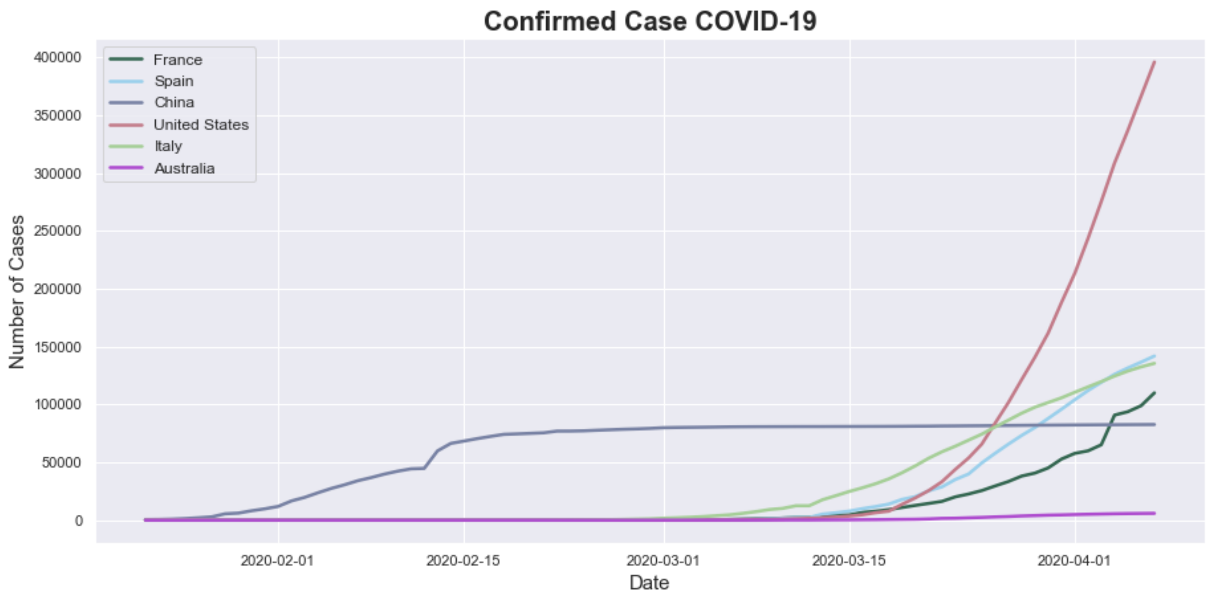
In [9]:

```
#untuk merapikan tanggal
confirmed_country.index = pd.to_datetime(confirmed_country.index,infer_datetime_format=T
rue)

#memasukkan masing-masing negara ke variable
a = confirmed_country['France']
b = confirmed_country['Spain']
c = confirmed_country['China']
d = confirmed_country['US']
e = confirmed_country['Italy']
f = confirmed_country['Australia']

#size chart
plt.figure(figsize = (15, 7))

#insert plot
plt.plot(a, color = "#346751", linewidth = 2.5)
plt.plot(b, color = "#9AD0EC", linewidth = 2.5)
plt.plot(c, color = "#7882A4", linewidth = 2.5)
plt.plot(d, color = "#C37B89", linewidth = 2.5)
plt.plot(e, color = "#A6CF98", linewidth = 2.5)
plt.plot(f, color = "#AE4CCF", linewidth = 2.5)
plt.legend(['France','Spain','China','United States','Italy','Australia'], fontsize = 12
)
plt.title('Confirmed Case COVID-19', fontsize = 20, fontweight = 'bold')
plt.xlabel('Date', fontsize = 15)
plt.ylabel('Number of Cases', fontsize = 15)
plt.show()
```



Berdasarkan hasil visualisasi diatas, United States memiliki total confirmed case covid-19 yang paling tinggi dan china memiliki jumlah confirmed casenya tidak melebihi 100.000.

3. Jumlah kasus kematian COVID-19 yang dilaporkan berdasarkan negara (Bobot: 15%)

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`

19. Tampilkan beberapa baris teratas/terbawah beserta informasi dari dataframe `raw_data_deaths`

In [10]:

```
# Beberapa baris teratas/terbawah dari dataframe raw_data_death

raw_data_deaths.head(5)
```

Out[10]:

	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
0	NaN	Afghanistan	33.0000	65.0000	0	0	0	0	0	0	...	4	4
1	NaN	Albania	41.1533	20.1683	0	0	0	0	0	0	...	10	11
2	NaN	Algeria	28.0339	1.6596	0	0	0	0	0	0	...	31	35
3	NaN	Andorra	42.5063	1.5218	0	0	0	0	0	0	...	6	8
4	NaN	Angola	-11.2027	17.8739	0	0	0	0	0	0	...	2	2

5 rows × 81 columns

In [11]:

```
# 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  
 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
```

Hasil analisis yang saya dapatkan adalah terdapat 263 baris data kematian COVID-19

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 [12]:

```
deaths_each_country = raw_data_deaths.groupby('Country/Region').sum()
del deaths_each_country['Long']
del deaths_each_country['Lat']
```

In [13]:

```
deaths_each_country = deaths_each_country.T
```

In [14]:

```
deaths_each_country.index
```

Out[14]:

```
Index(['1/22/20', '1/23/20', '1/24/20', '1/25/20', '1/26/20', '1/27/20',
       '1/28/20', '1/29/20', '1/30/20', '1/31/20', '2/1/20', '2/2/20',
       '1/28/20', '1/29/20', '1/30/20', '1/31/20', '2/1/20', '2/2/20',
       '2/3/20', '2/4/20', '2/5/20', '2/6/20', '2/7/20', '2/8/20', '2/9/20',
       '2/10/20', '2/11/20', '2/12/20', '2/13/20', '2/14/20', '2/15/20',
       '2/16/20', '2/17/20', '2/18/20', '2/19/20', '2/20/20', '2/21/20',
       '2/22/20', '2/23/20', '2/24/20', '2/25/20', '2/26/20', '2/27/20',
       '2/28/20', '2/29/20', '3/1/20', '3/2/20', '3/3/20', '3/4/20', '3/5/20',
       '3/6/20', '3/7/20', '3/8/20', '3/9/20', '3/10/20', '3/11/20', '3/12/20',
       '3/13/20', '3/14/20', '3/15/20', '3/16/20', '3/17/20', '3/18/20',
       '3/19/20', '3/20/20', '3/21/20', '3/22/20', '3/23/20', '3/24/20',
       '3/25/20', '3/26/20', '3/27/20', '3/28/20', '3/29/20', '3/30/20',
       '3/31/20', '4/1/20', '4/2/20', '4/3/20', '4/4/20', '4/5/20', '4/6/20',
       '4/7/20'],
      dtype='object')
```

In [15]:

```
deaths_each_country[0:5]
```

Out[15]:

Country/Region	Afghanistan	Albania	Algeria	Andorra	Angola	Antigua and Barbuda	Argentina	Armenia	Australia	Austria	...	United Arab Emirates
1/22/20	0	0	0	0	0	0	0	0	0	0	...	0
1/23/20	0	0	0	0	0	0	0	0	0	0	...	0
1/24/20	0	0	0	0	0	0	0	0	0	0	...	0
1/25/20	0	0	0	0	0	0	0	0	0	0	...	0
1/26/20	0	0	0	0	0	0	0	0	0	0	...	0

5 rows × 184 columns

Dari data yang telah ditampilkan, ditunjukan bahwa pertumbuhan jumlah kasus kematian COVID-19 di setiap negara terdapat kasus yang makin hari makin bertambah dan ada juga negara yang stabil.

4. Penyelarasan kurva pertumbuhan seluruh negara dengan jumlah kasus kematian COVID-19 ≥ 25 (Bobot: 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 [16]:

```
deaths = deaths_each_country[deaths_each_country >= 25]
deaths
```

Out[16]:

Country/Region	Afghanistan	Albania	Algeria	Andorra	Angola	Antigua and Barbuda	Argentina	Armenia	Australia	Austria	...	United Arab Emirates
1/22/20	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN
1/23/20	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN
1/24/20	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN
1/25/20	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN
1/26/20	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN
...	...	...	...	...	...	...	...	...	...	...	...	...
4/3/20	NaN	NaN	105.0	NaN	NaN	NaN	39.0	NaN	28.0	168.0	...	NaN

Country/Region	Afghanistan	Albania	Algeria	Andorra	Angola	Antigua and Barbuda	Argentina	Armenia	Australia	Austria	...	United Arab Emirates
4/4/20	NaN	NaN	130.0	NaN	NaN	NaN	43.0	NaN	30.0	186.0	...	NaN
4/5/20	NaN	NaN	152.0	NaN	NaN	NaN	44.0	NaN	35.0	204.0	...	NaN
4/6/20	NaN	NaN	173.0	NaN	NaN	NaN	48.0	NaN	40.0	220.0	...	NaN
4/7/20	NaN	NaN	193.0	NaN	NaN	NaN	56.0	NaN	45.0	243.0	...	NaN

77 rows × 184 columns

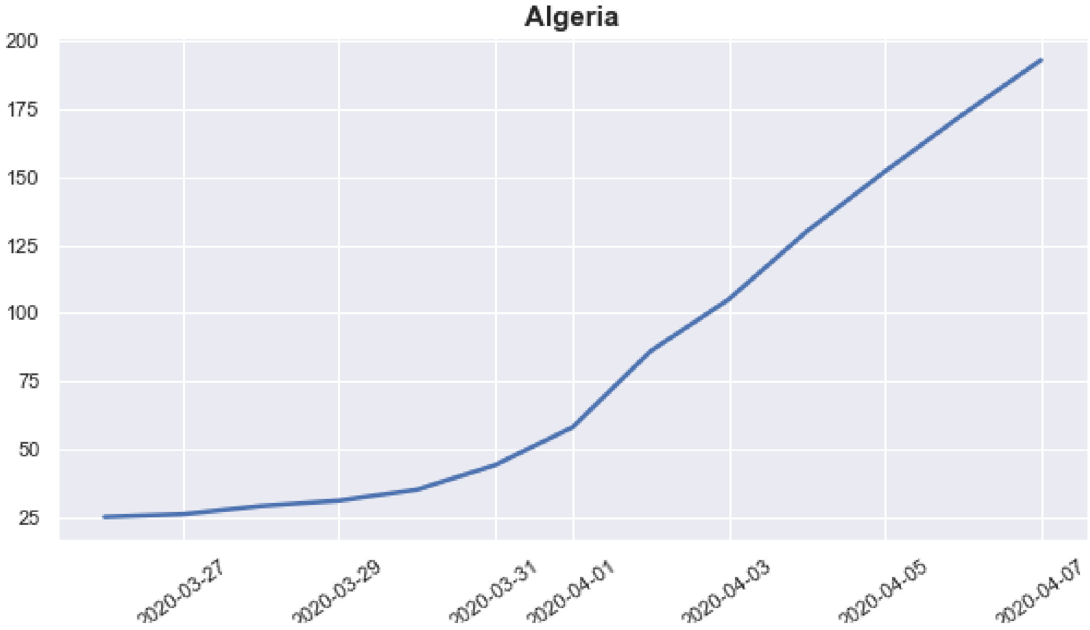
Visualisasikan hasil diatas dan berikan/atur judul, labels, dan spesifikasi (ukuran, warna, ketebalan, dll) yang sesuai sehingga plot/kurva yang dihasilkan rapi, menarik, dan mudah dipahami.

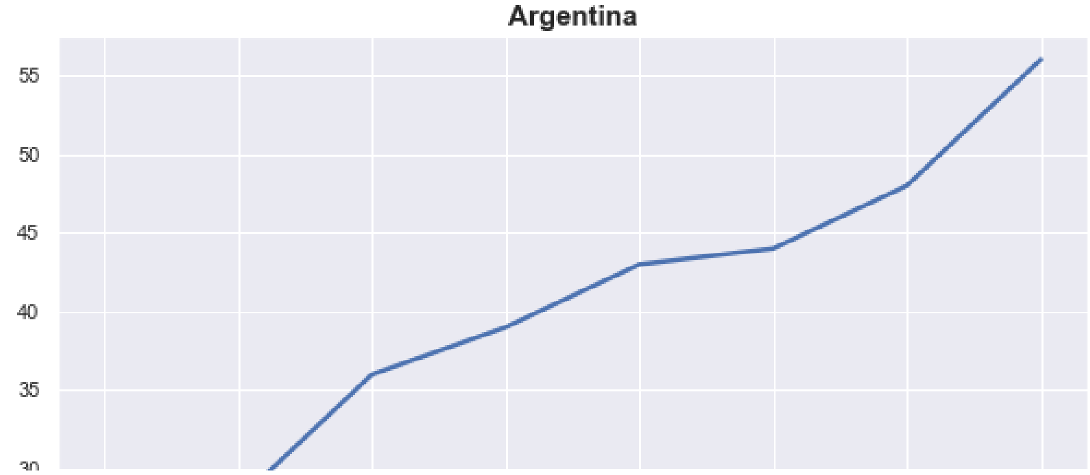
In [17]:

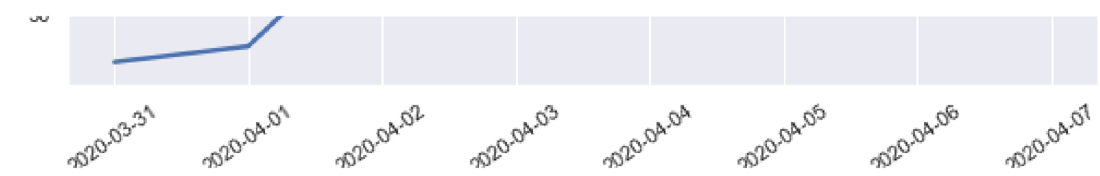
```
deaths.index = pd.to_datetime(deaths.index,infer_datetime_format=True)
```

In [18]:

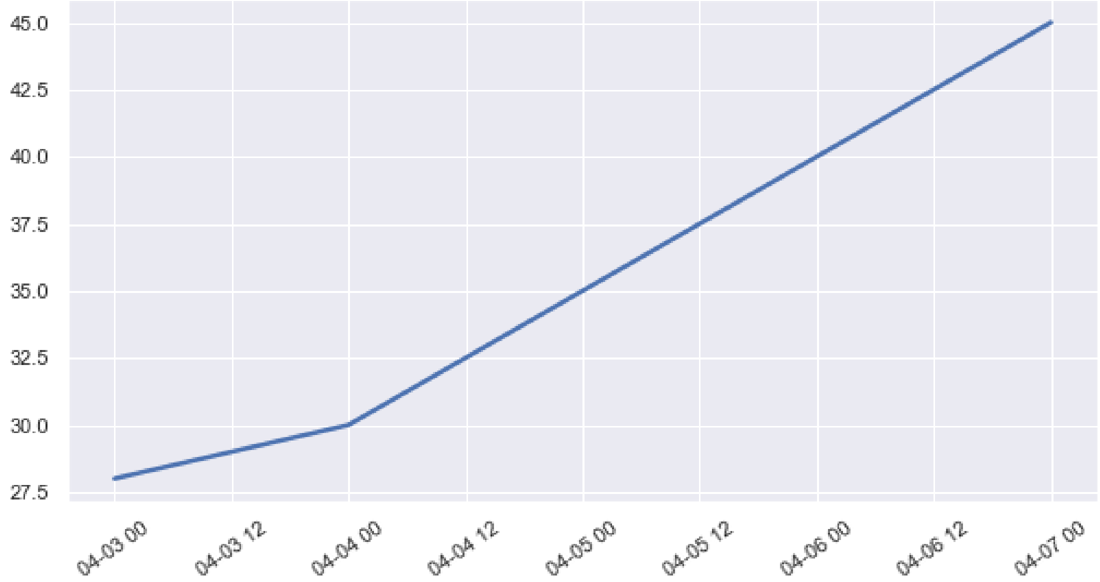
```
for i in deaths:
    test = deaths[i].sum()
    if test == 0:
        continue
    else:
        plt.figure(figsize = (10,5))
        plt.title(i, fontsize = 15, fontweight = 'bold')
        plt.plot(deaths[i], linewidth = 2.5)
        plt.xticks(rotation=35)
        plt.show()
```



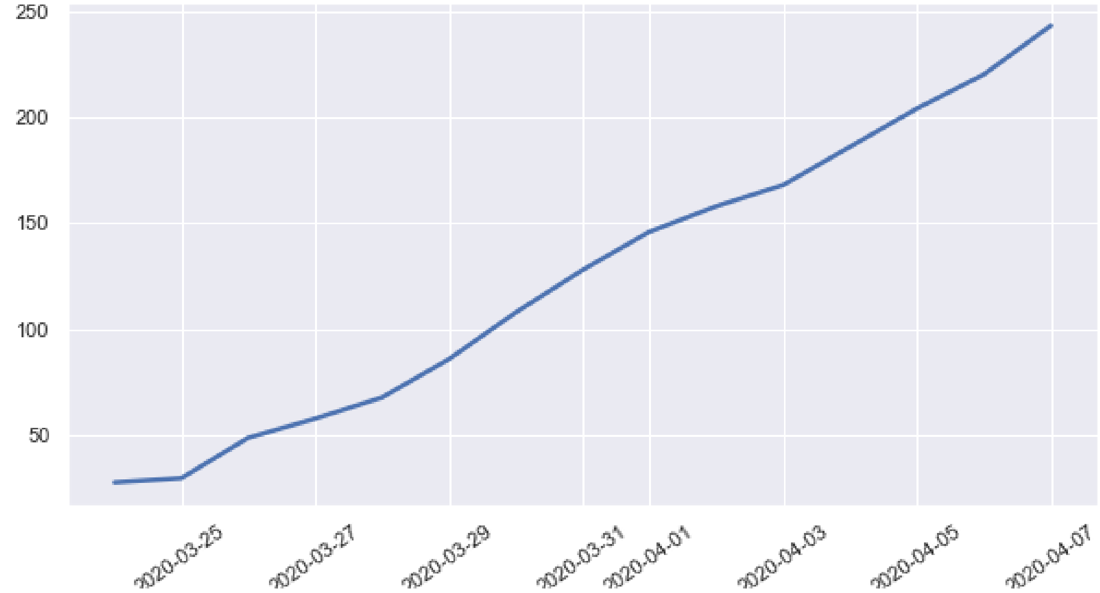




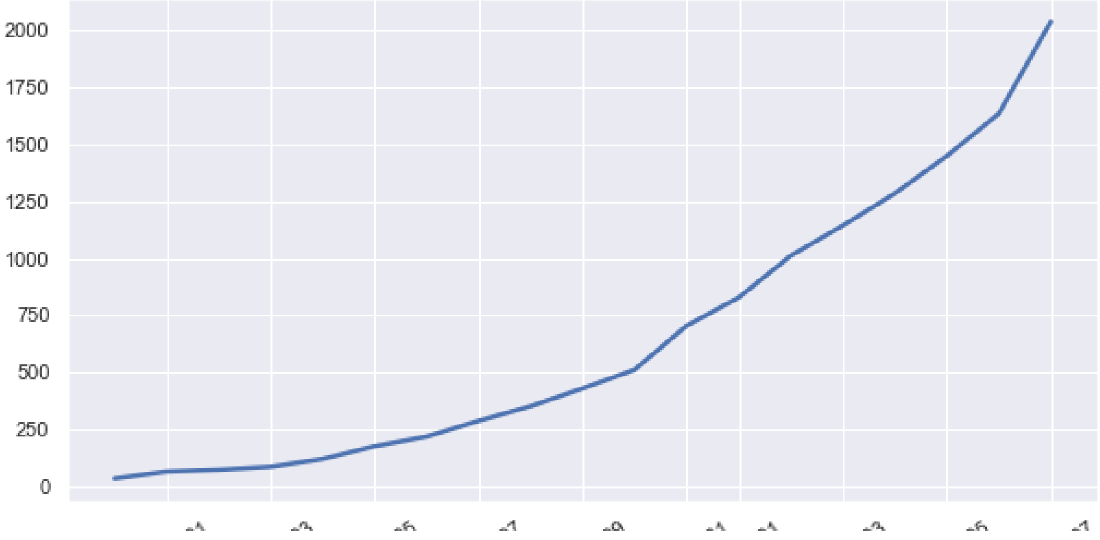
Australia

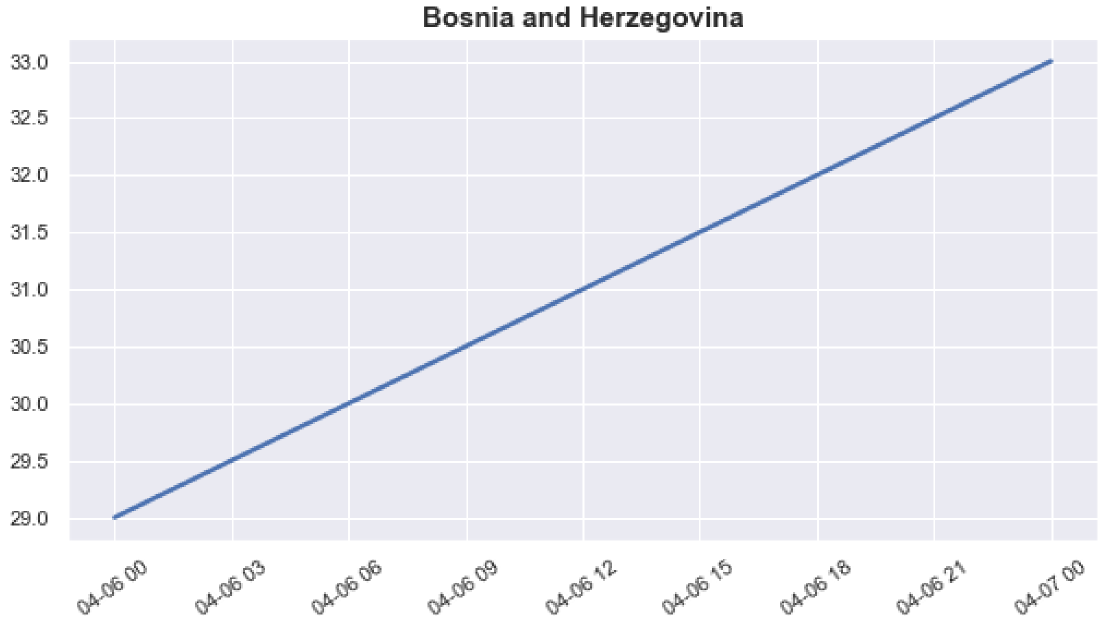


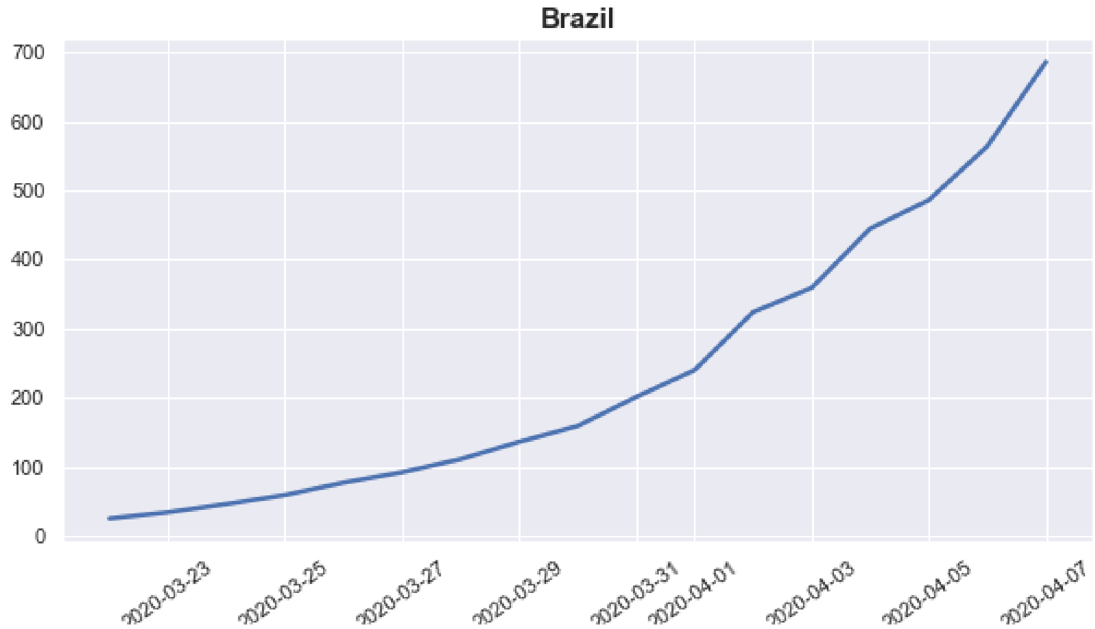
Austria

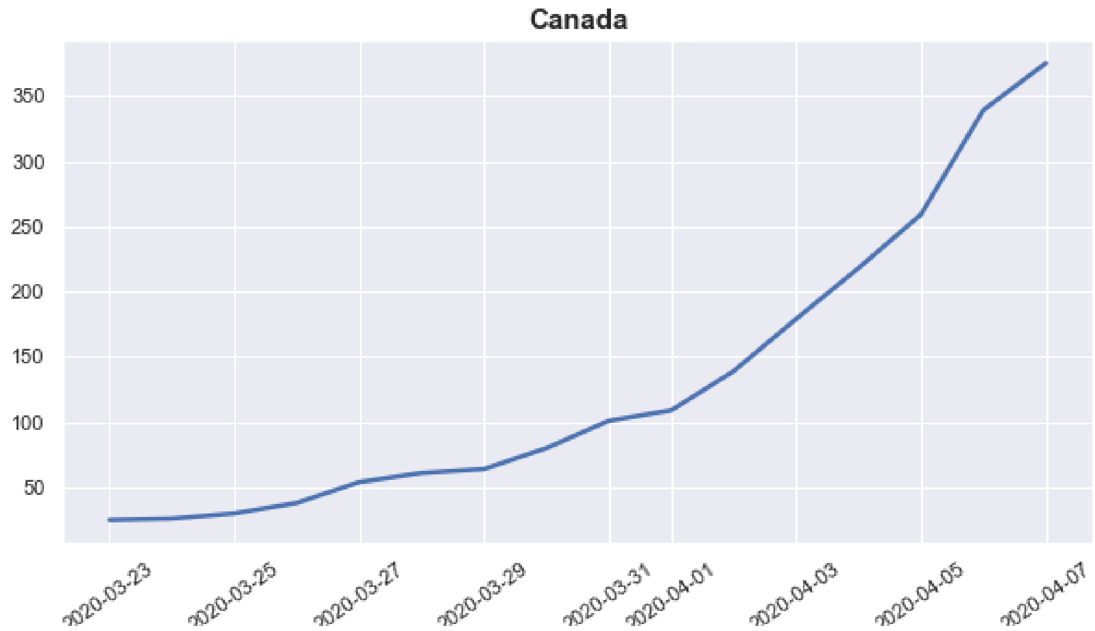


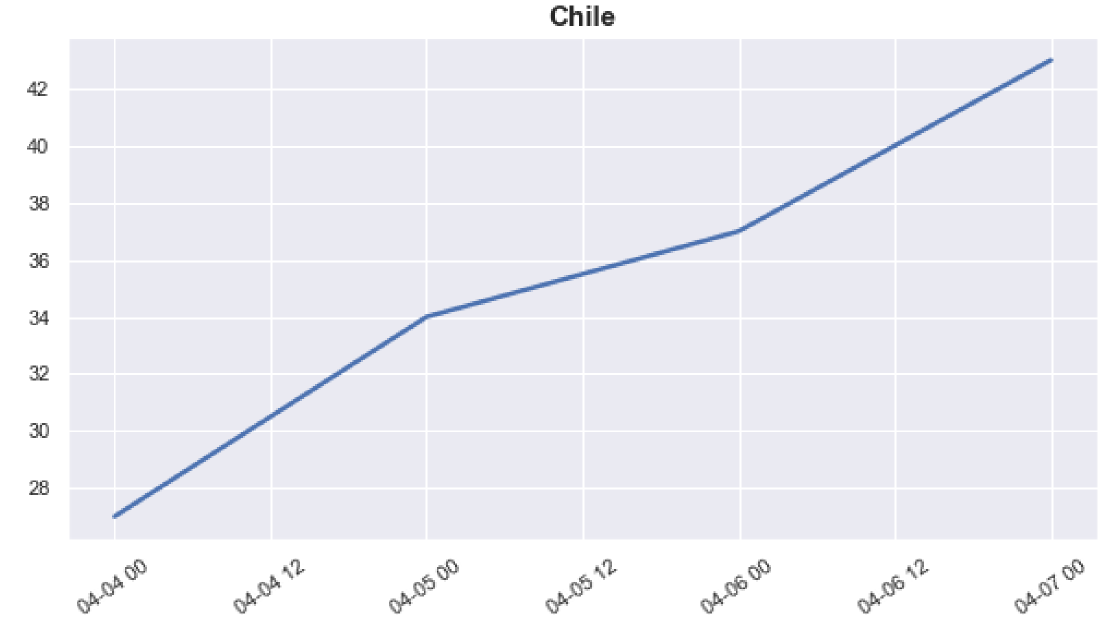
Belgium

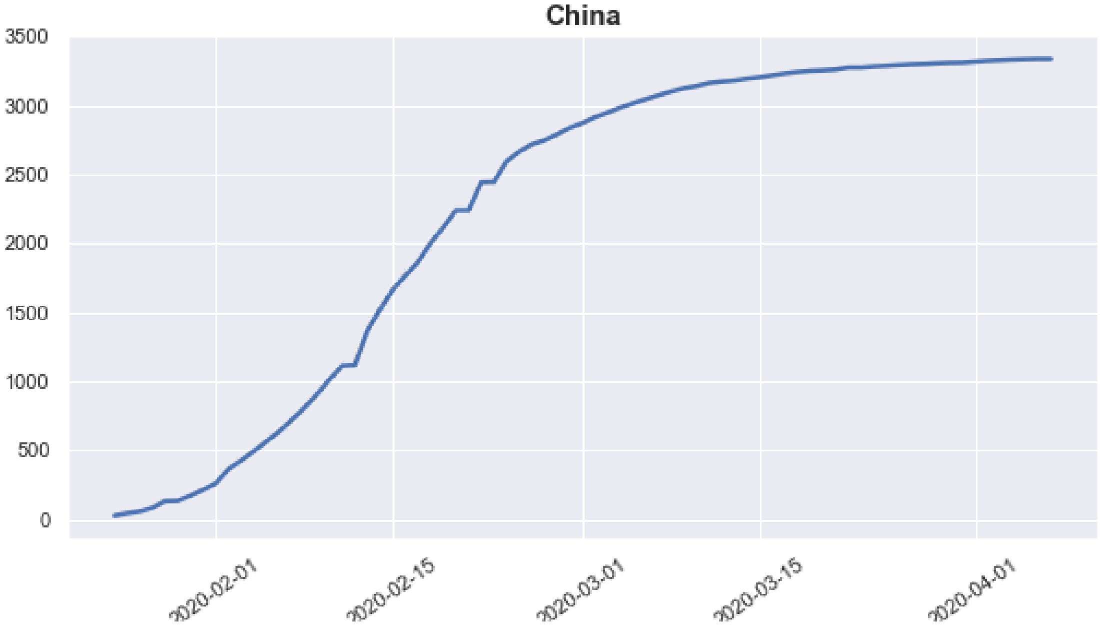


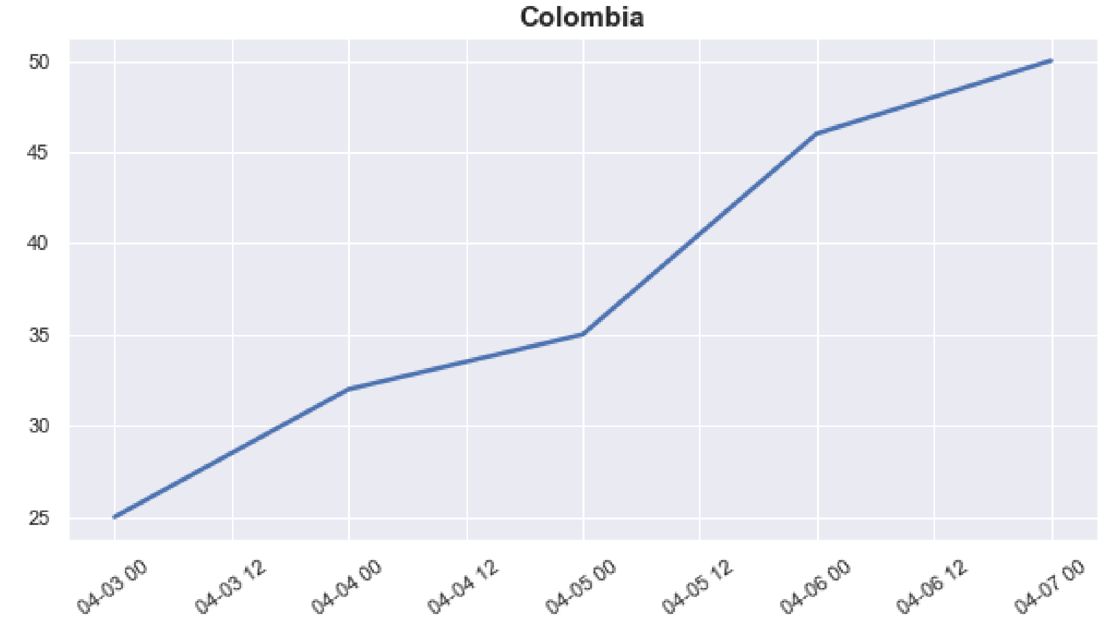

Bosnia and Herzegovina


Brazil

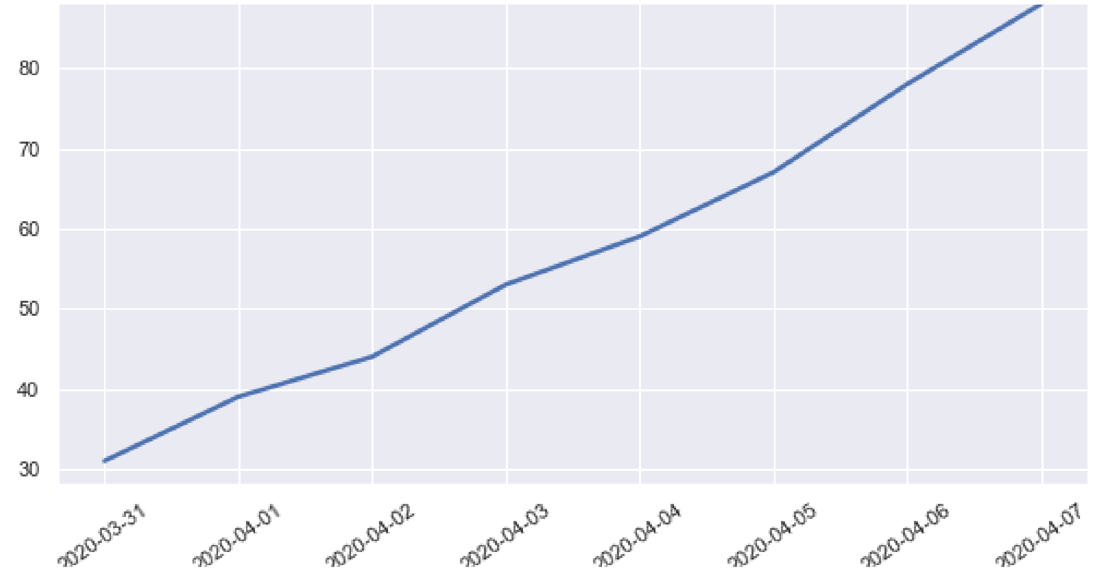

Canada


Chile
42
40
38
36
34
32
30
28
04-04 00
04-04 12
04-05 00
04-05 12
04-06 00
04-06 12
04-07 00

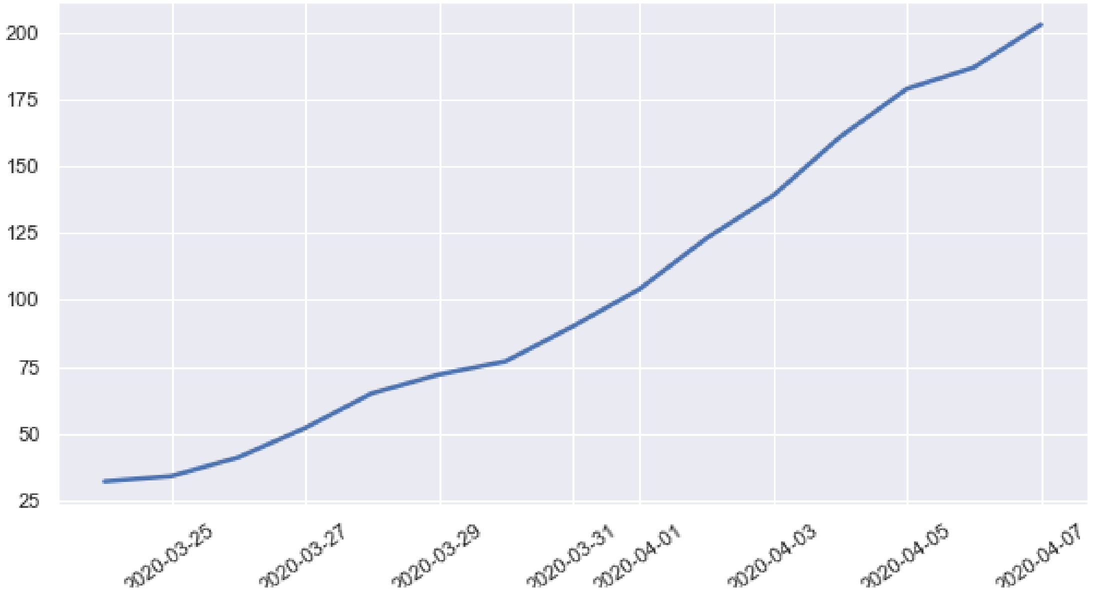

China
3500
3000
2500
2000
1500
1000
500
0
2020-02-01
2020-02-15
2020-03-01
2020-03-15
2020-04-01


Colombia
50
45
40
35
30
25
04-03 00
04-03 12
04-04 00
04-04 12
04-05 00
04-05 12
04-06 00
04-06 12
04-07 00

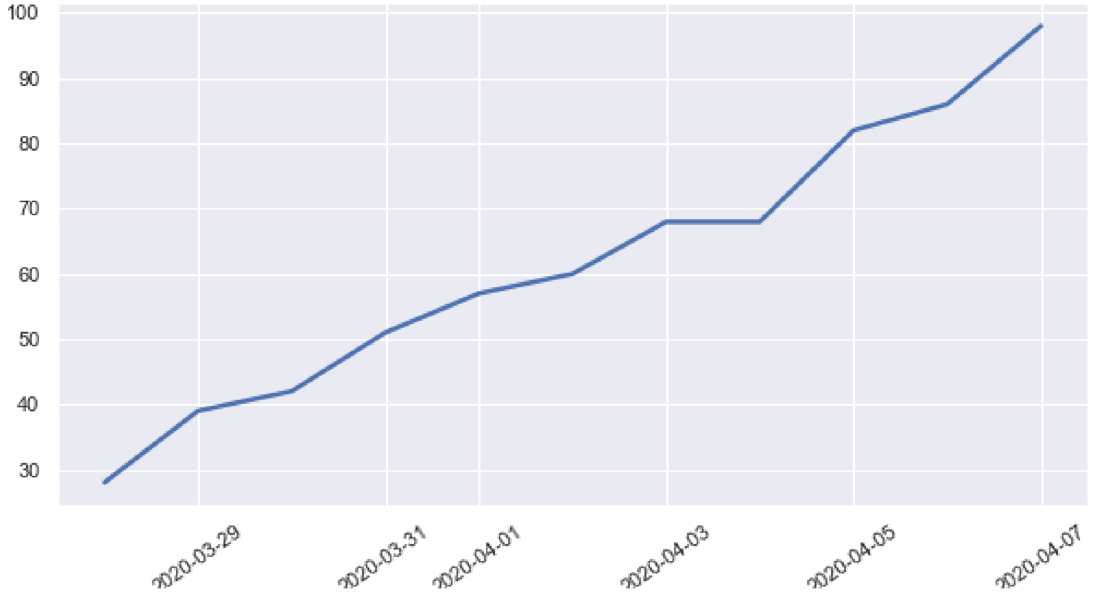
Czechia
90



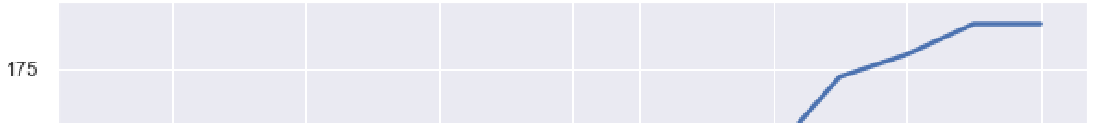
Denmark

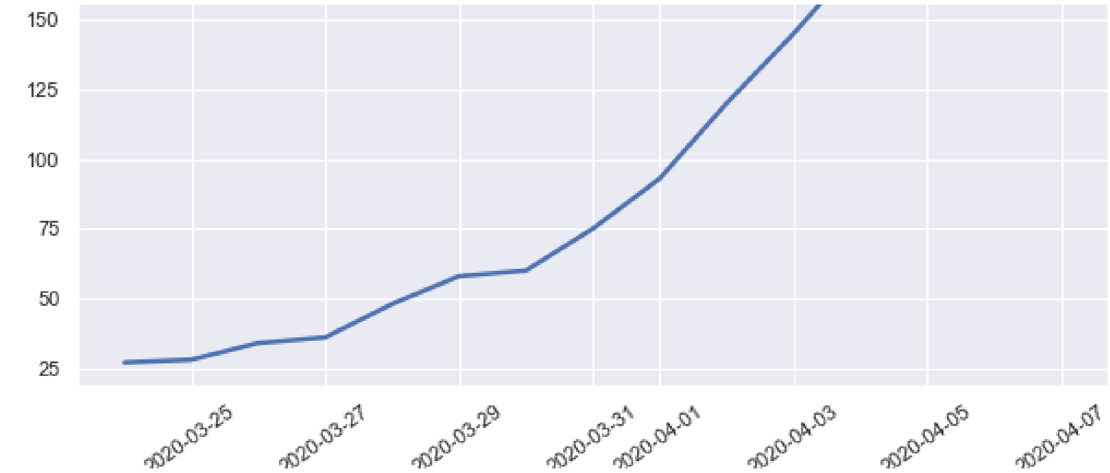


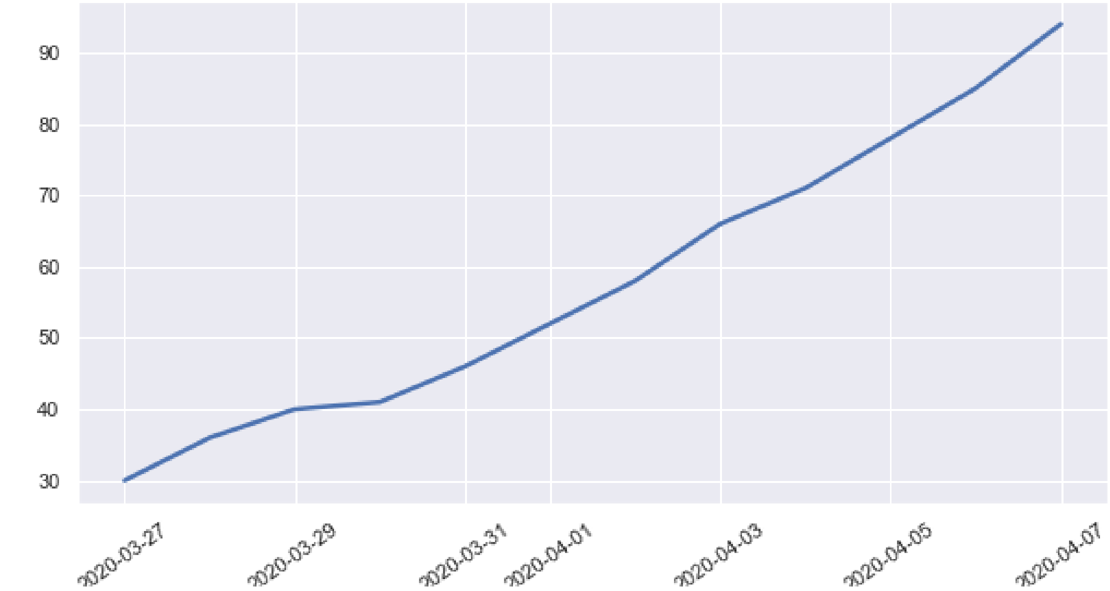
Dominican Republic

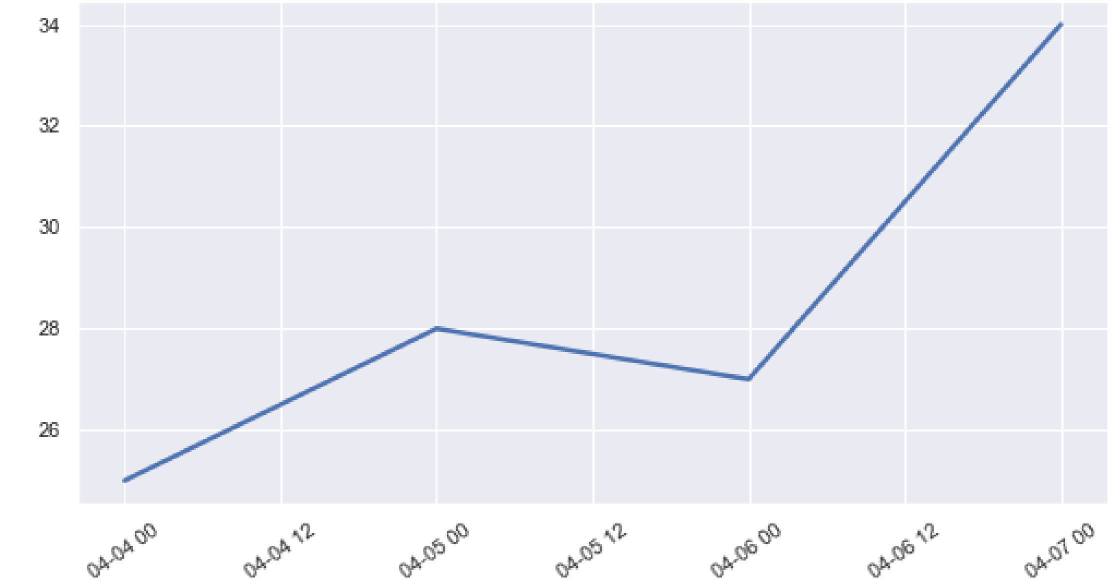


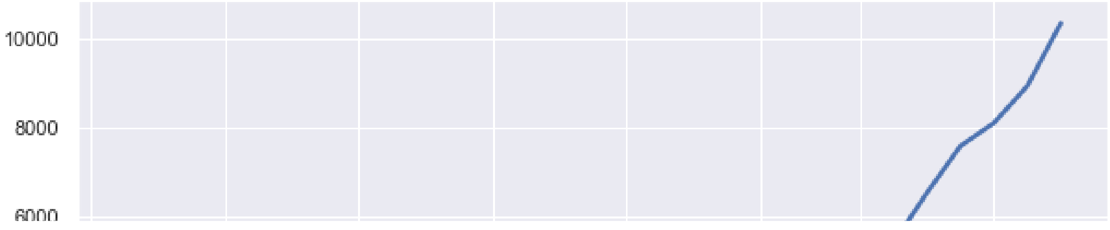
Ecuador

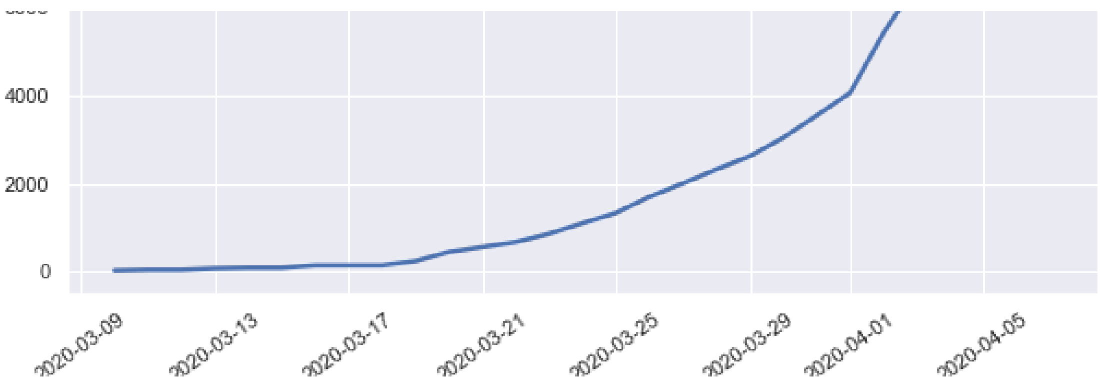


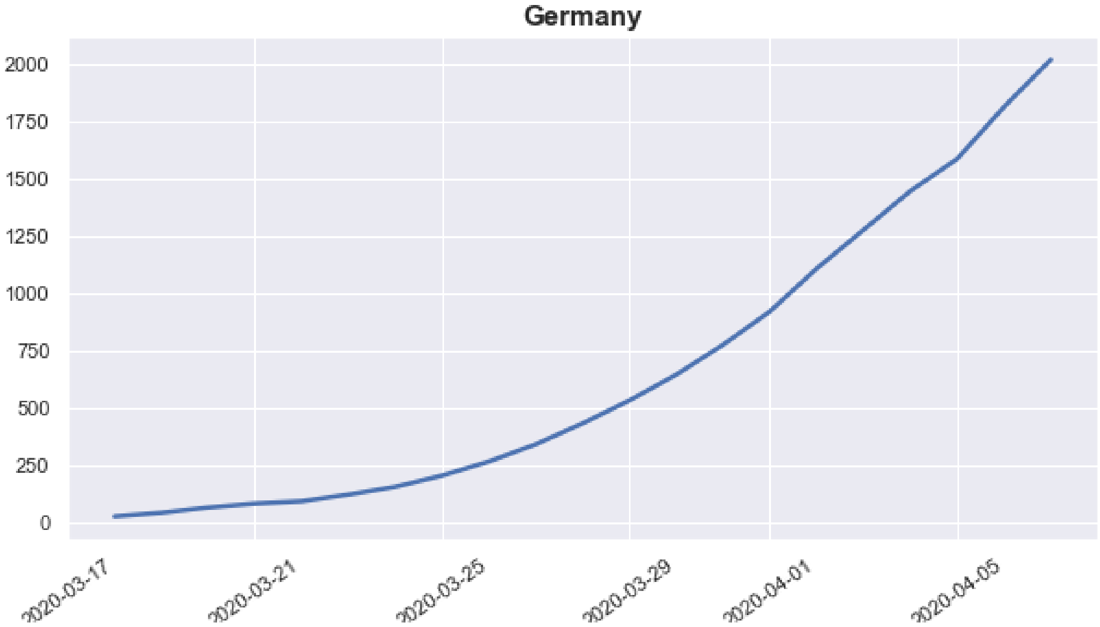



Egypt

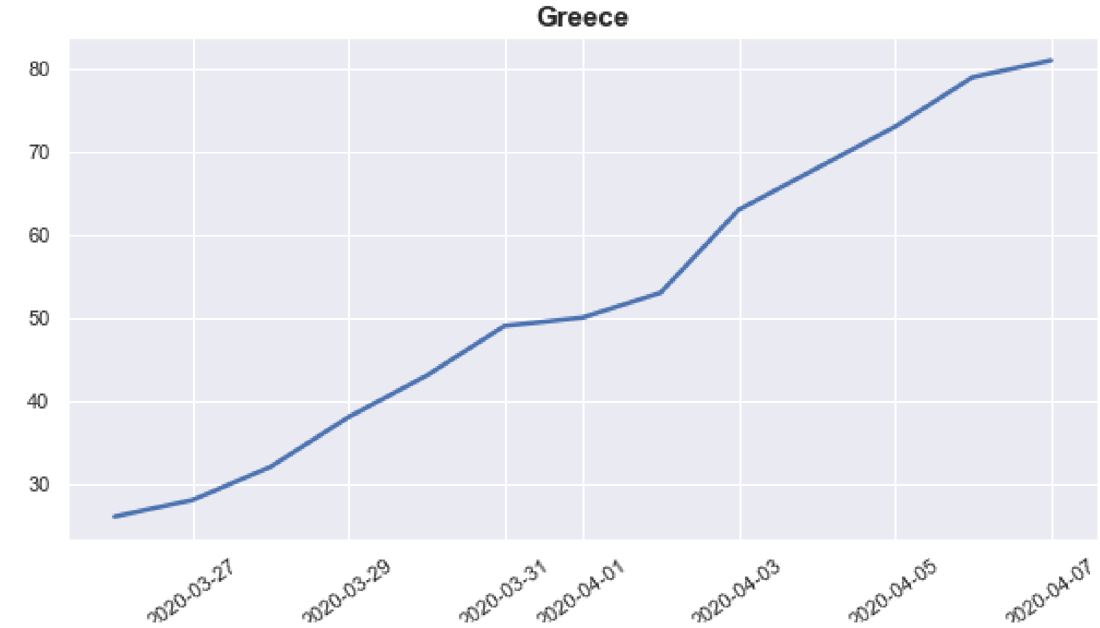

Finland

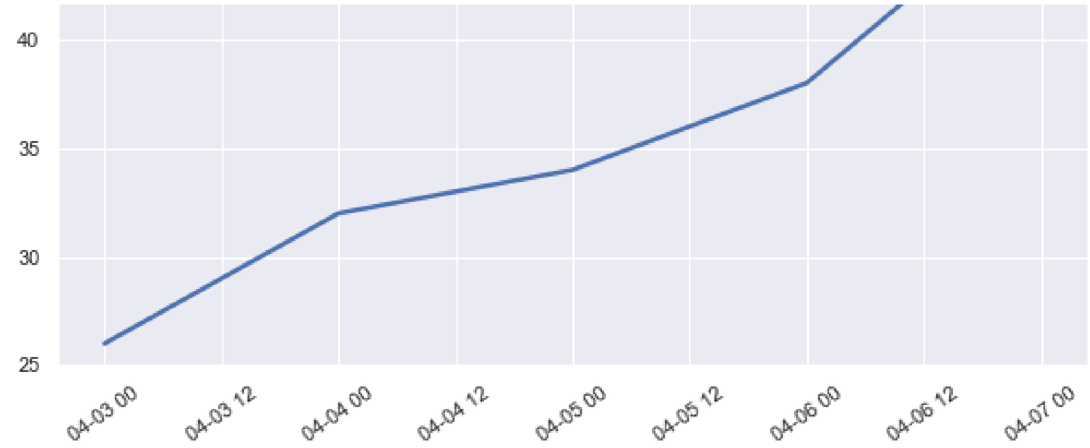

France



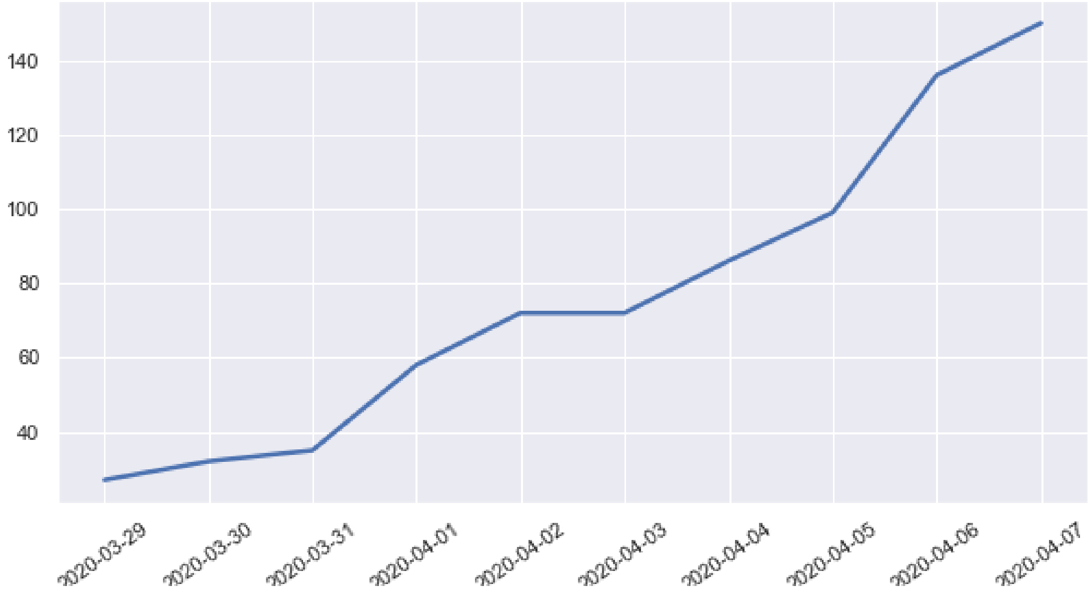


```
<ipython-input-18-4c143f65e1d0>:6: RuntimeWarning: More than 20 figures have been opened.
Figures created through the pyplot interface (`matplotlib.pyplot.figure`) are retained un
til explicitly closed and may consume too much memory. (To control this warning, see the
rcParam `figure.max_open_warning`).
  plt.figure(figsize = (10,5))
```

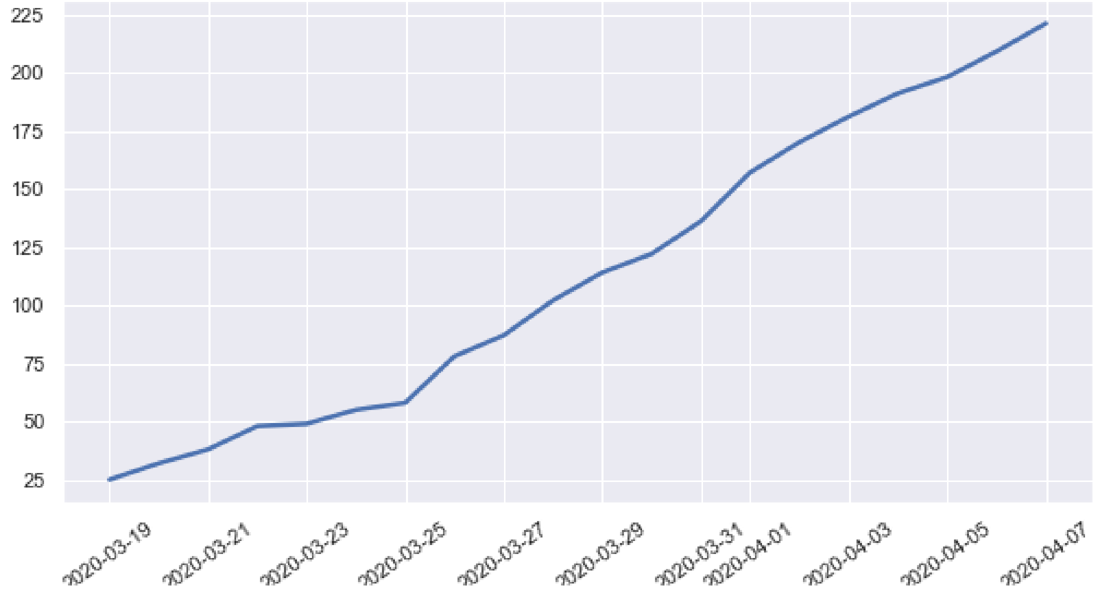




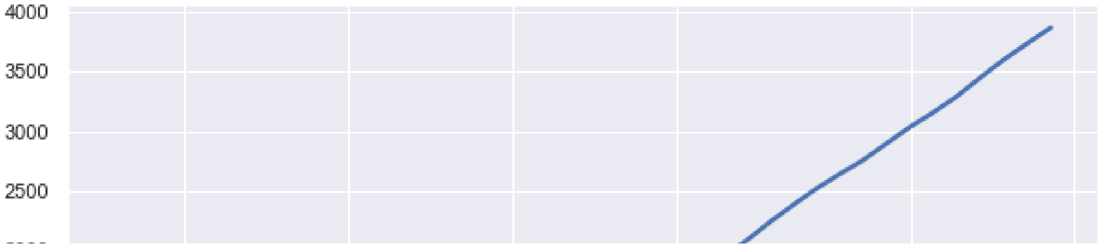
India

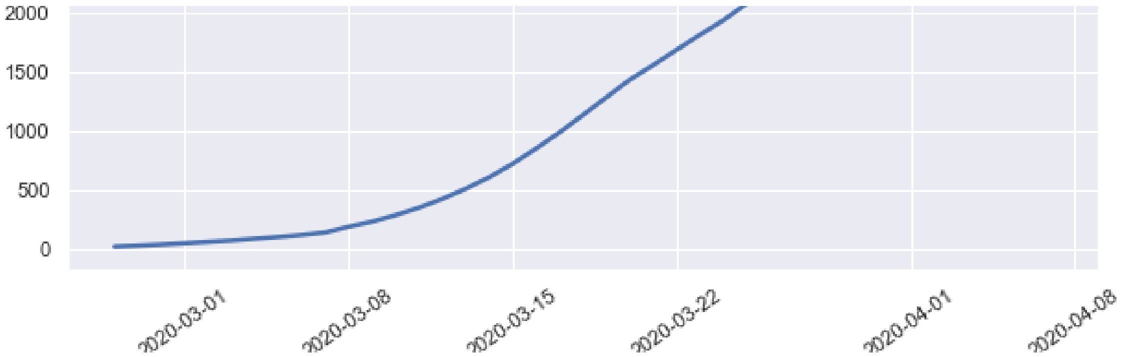


Indonesia

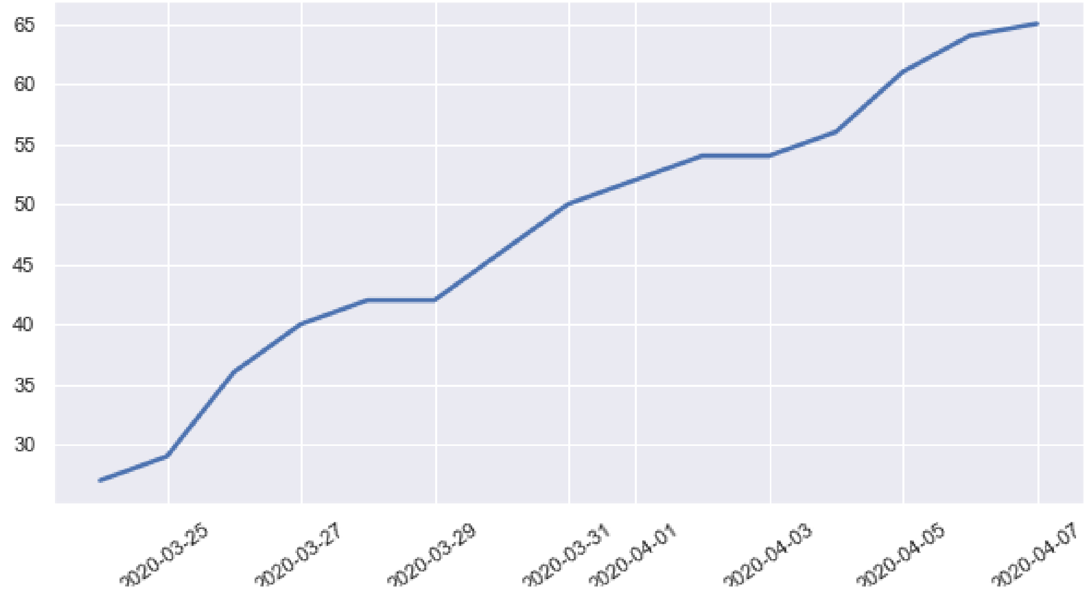


Iran

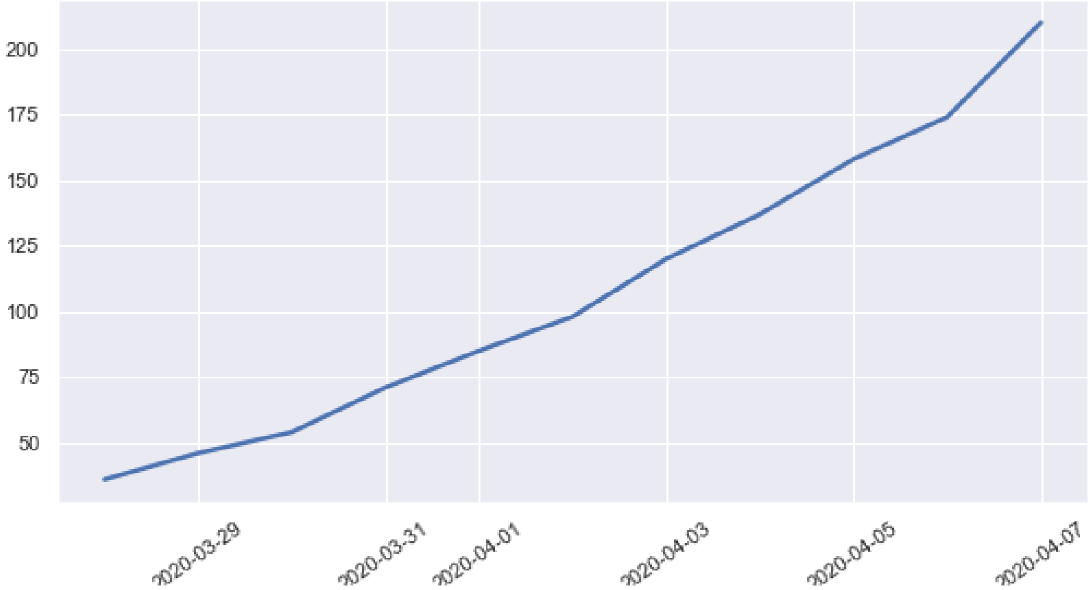




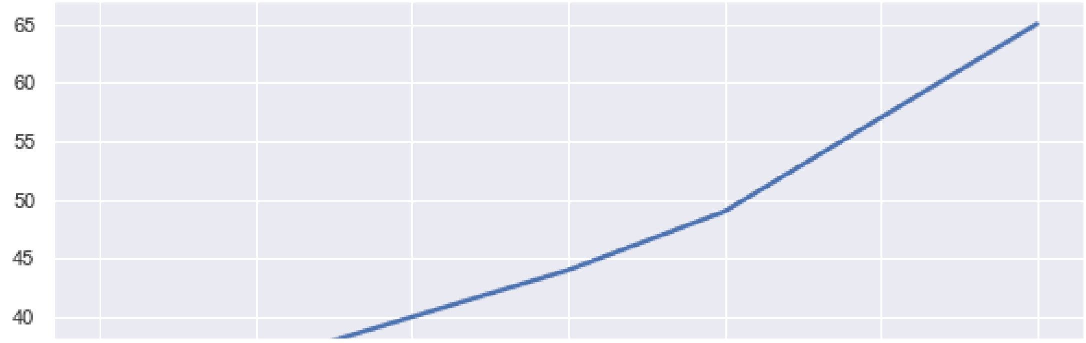
Iraq

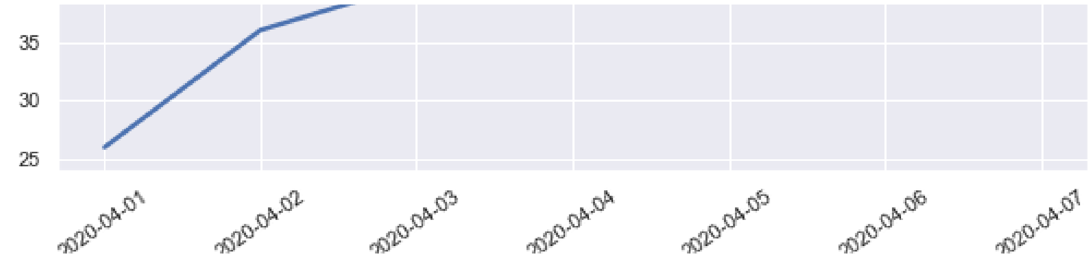


Ireland

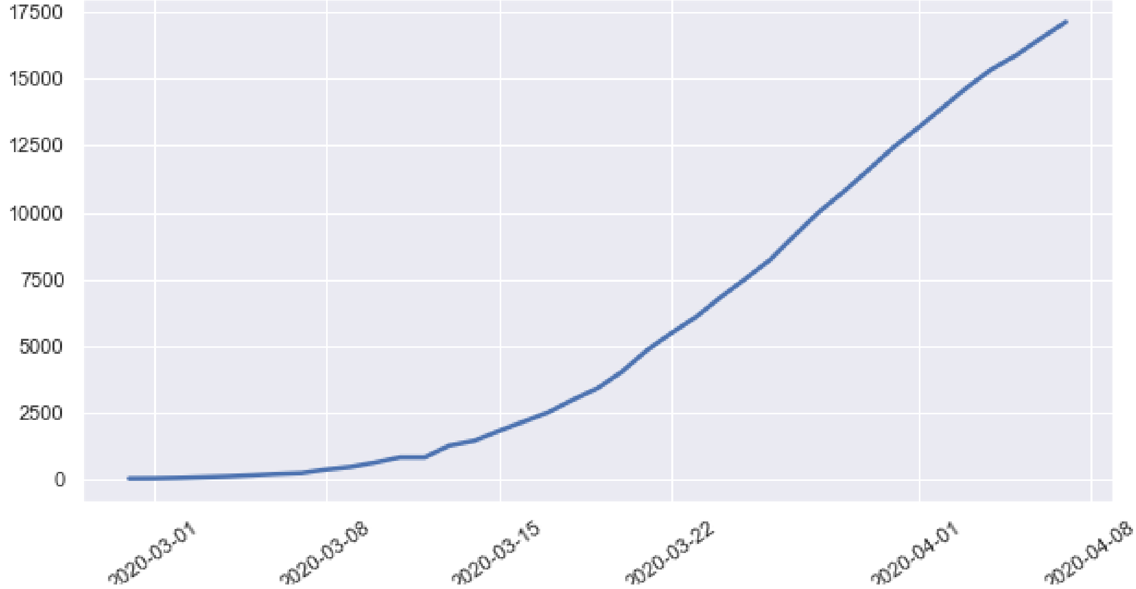


Israel

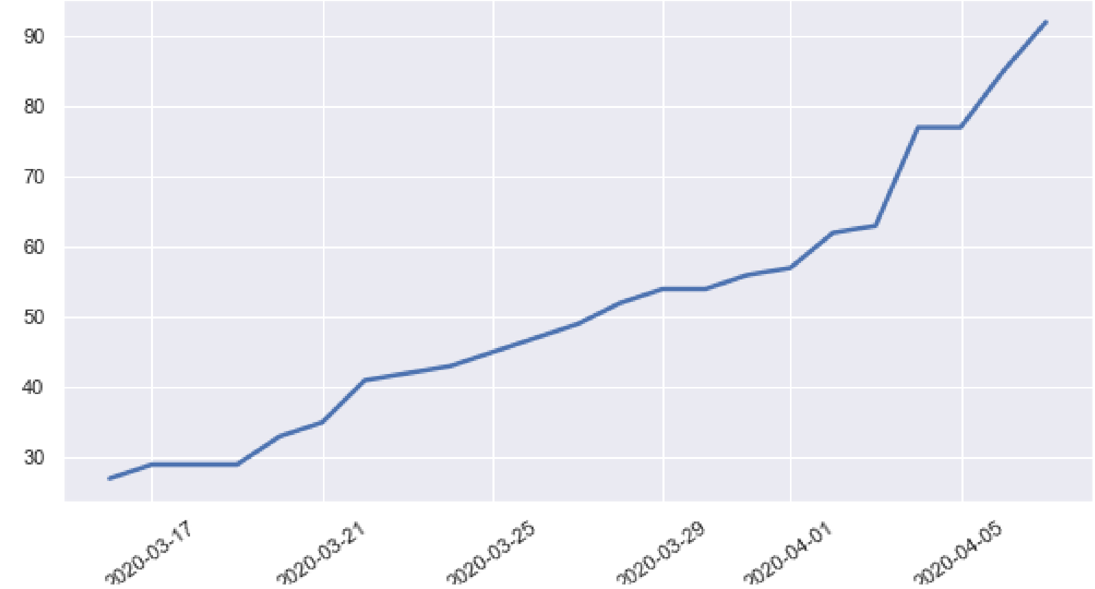




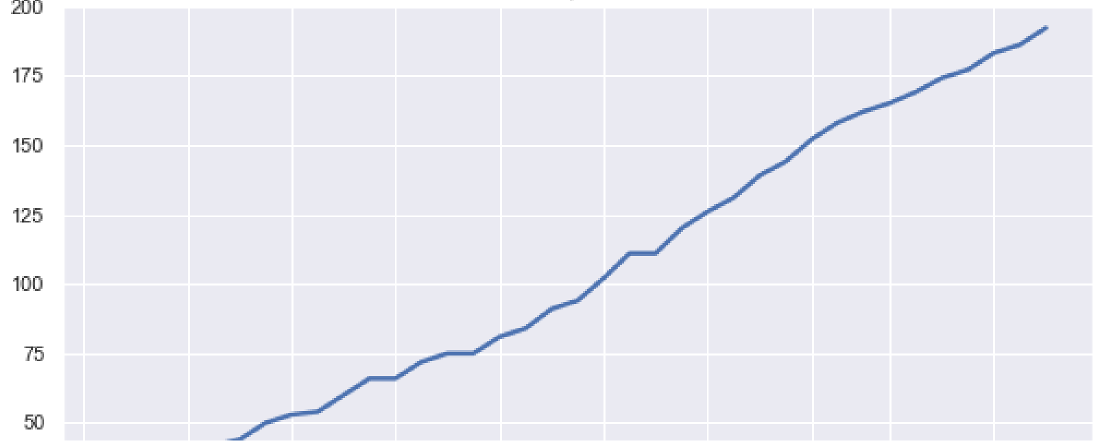
Italy

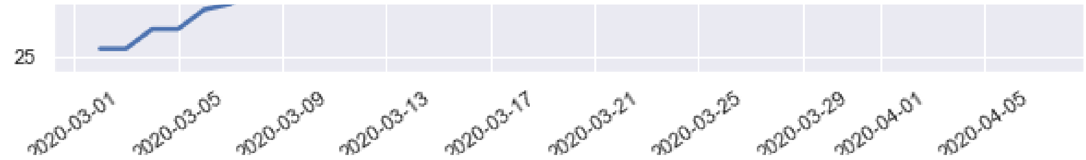


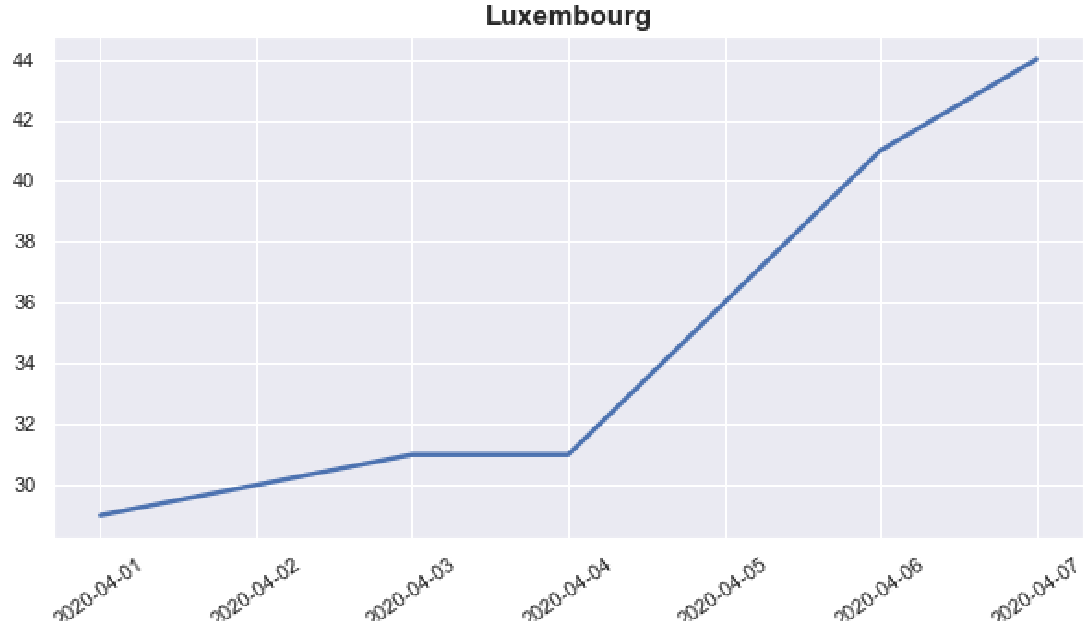
Japan

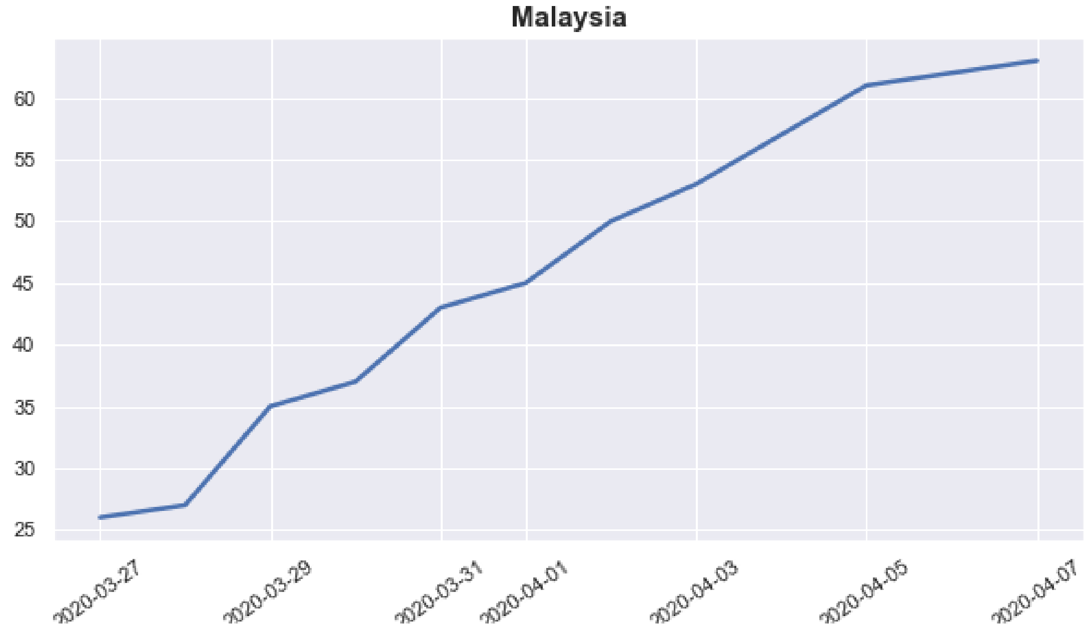


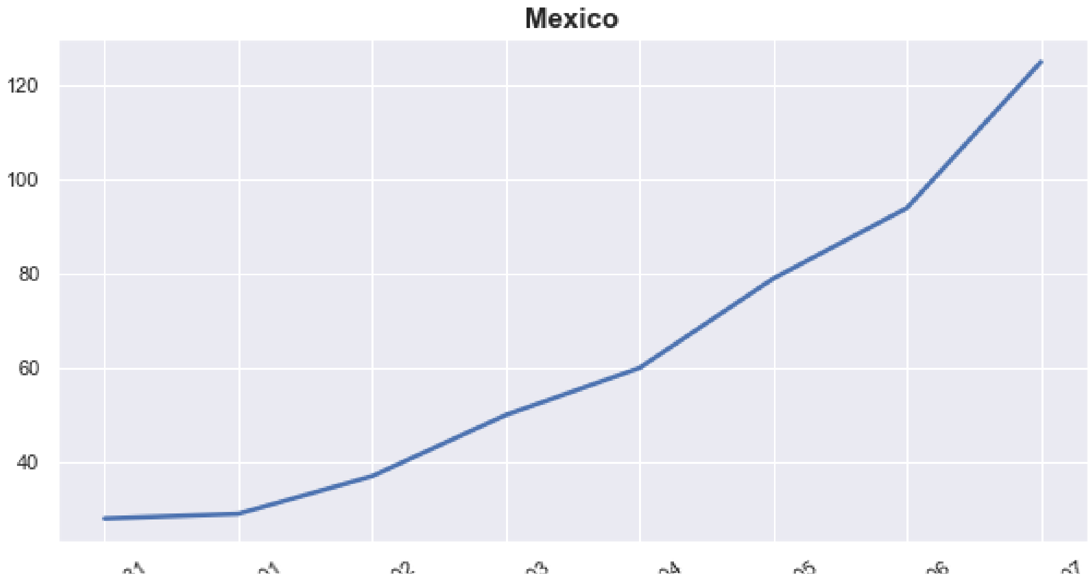
Korea, South

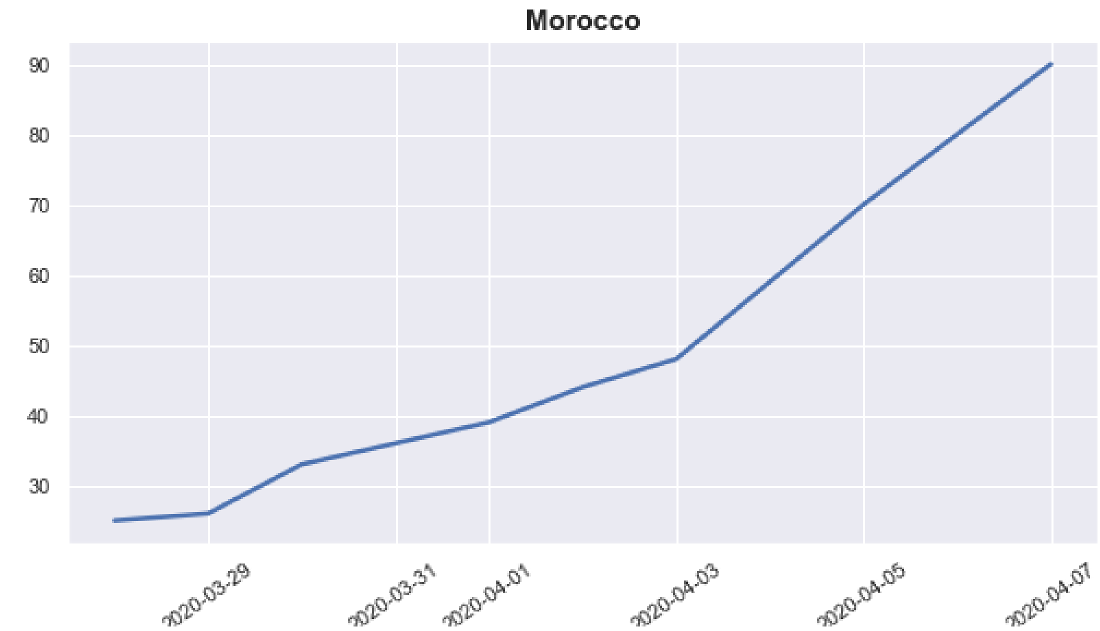


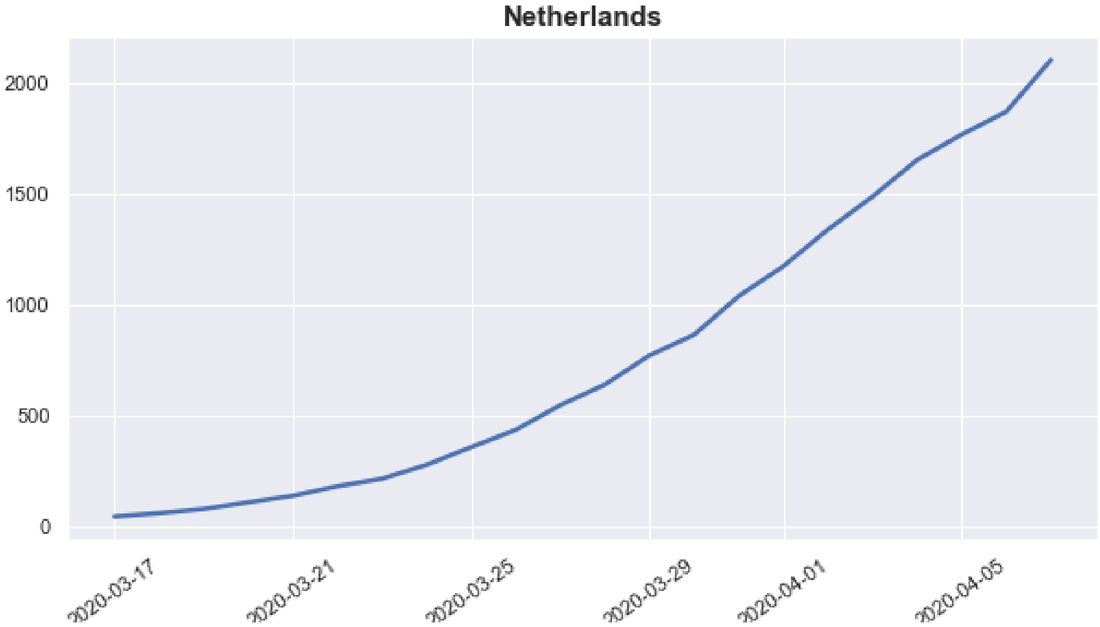


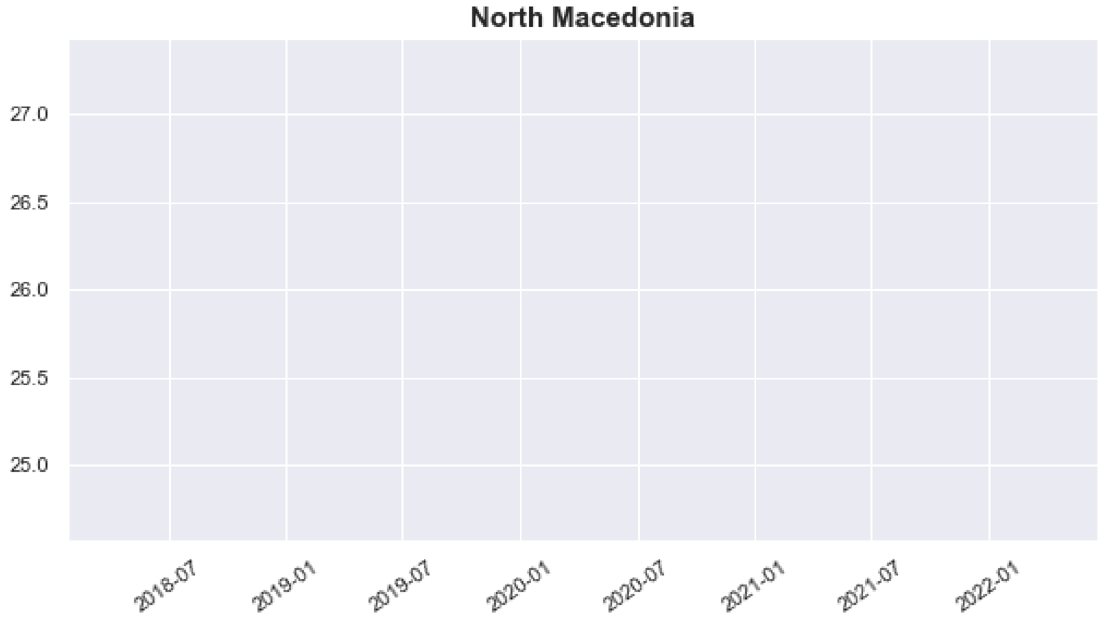

Luxembourg

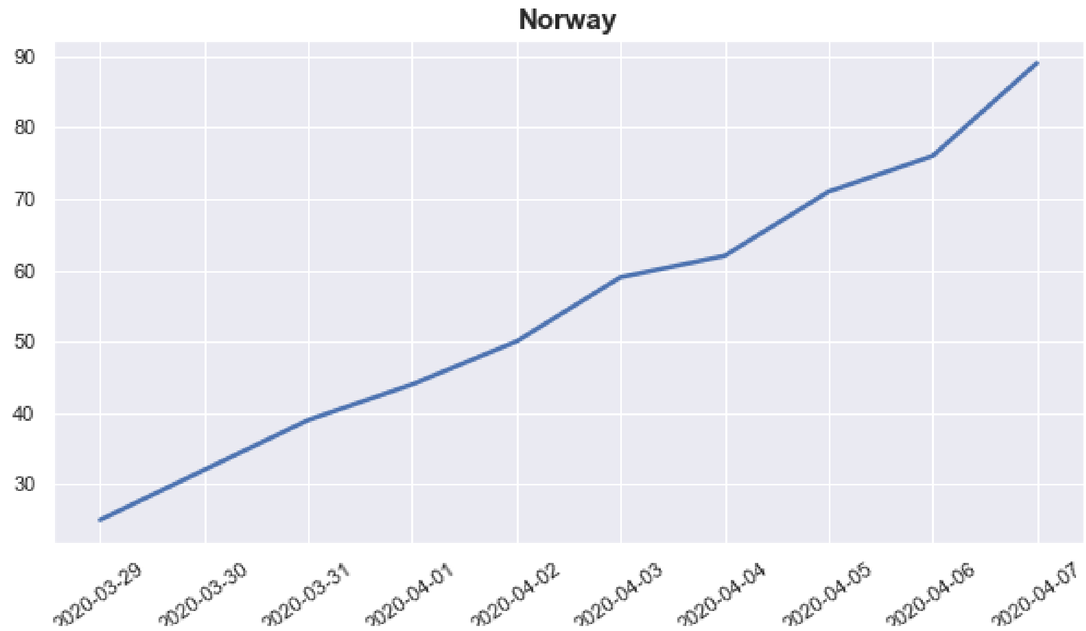

Malaysia


Mexico

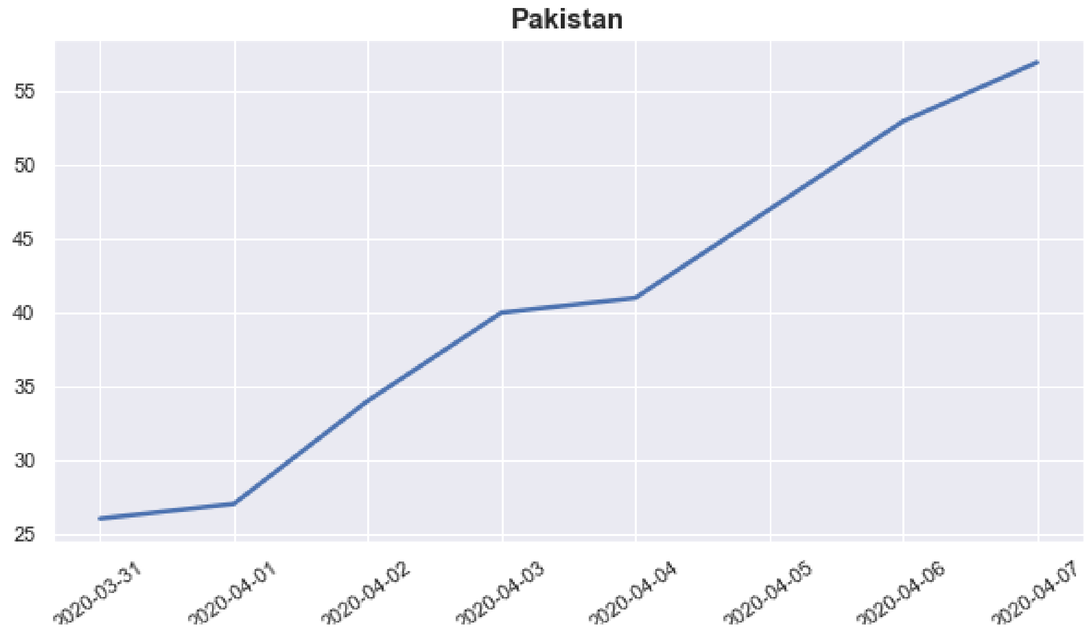

Morocco


Netherlands

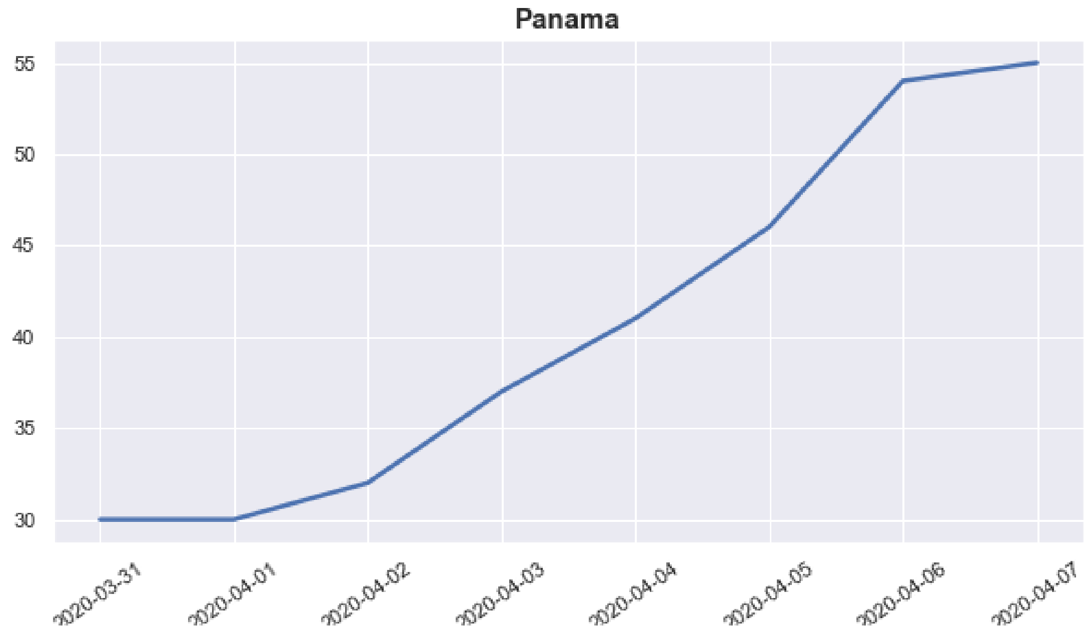

North Macedonia



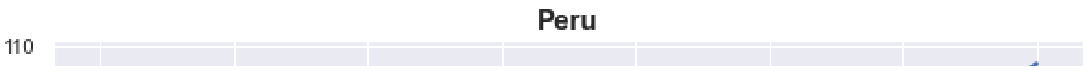
Norway



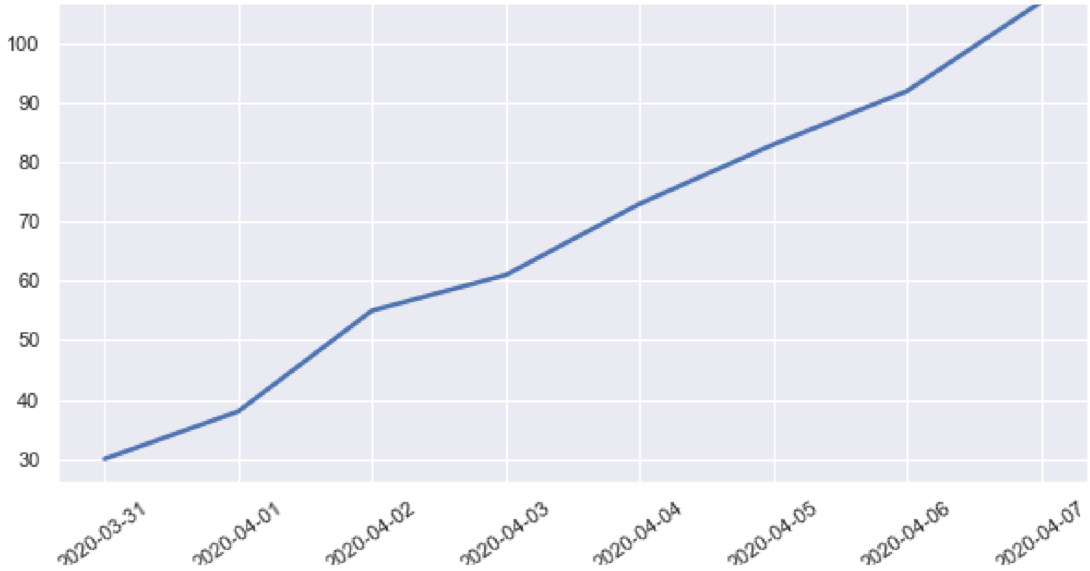
Pakistan



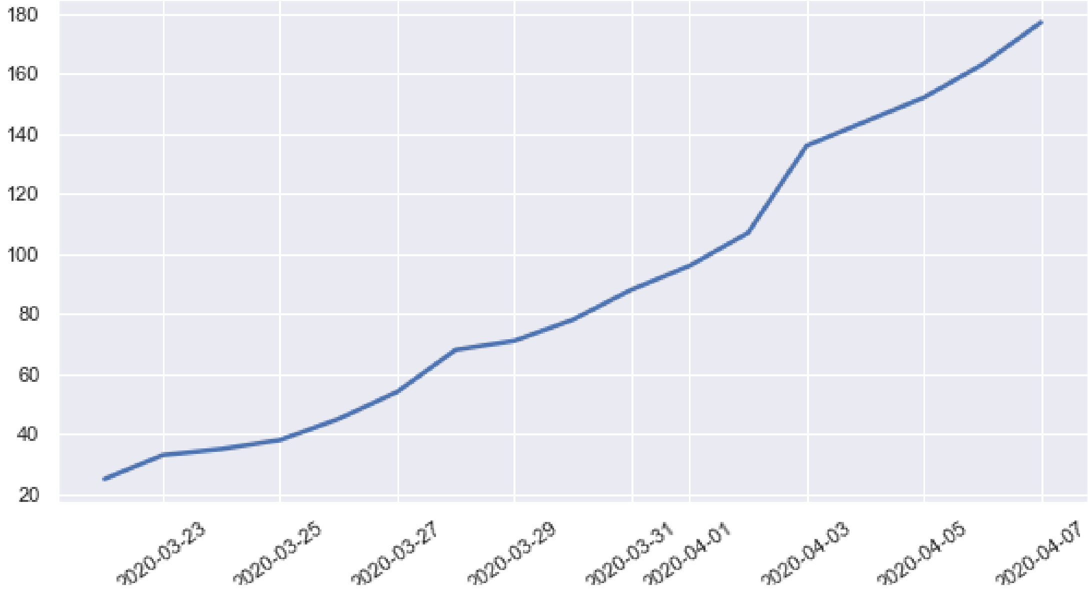
Panama



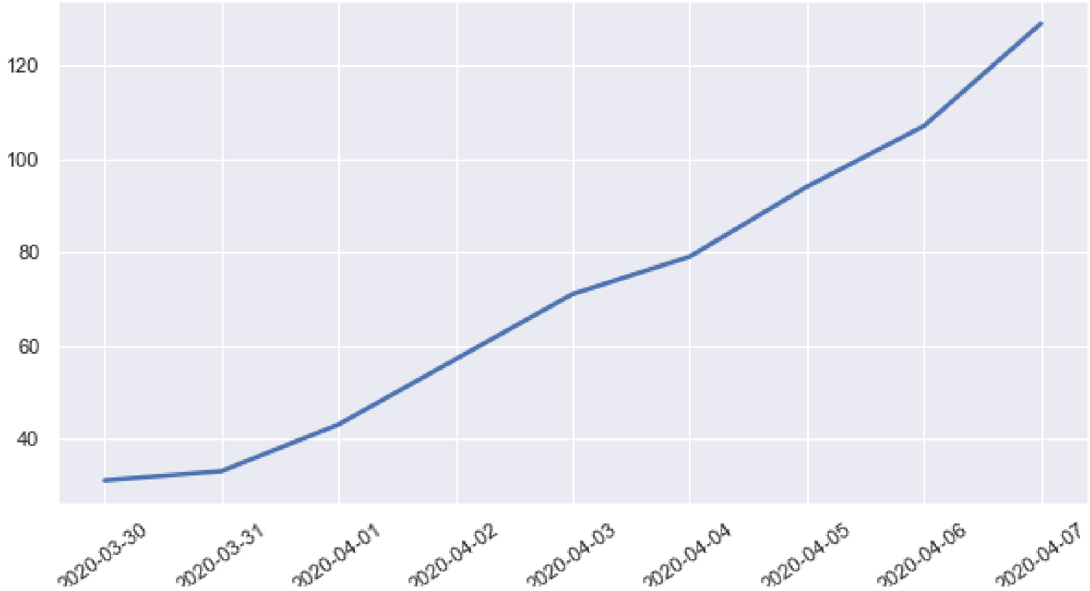
Peru



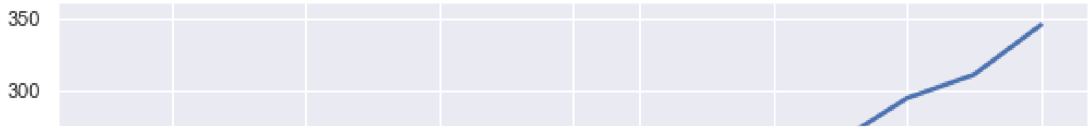
Philippines

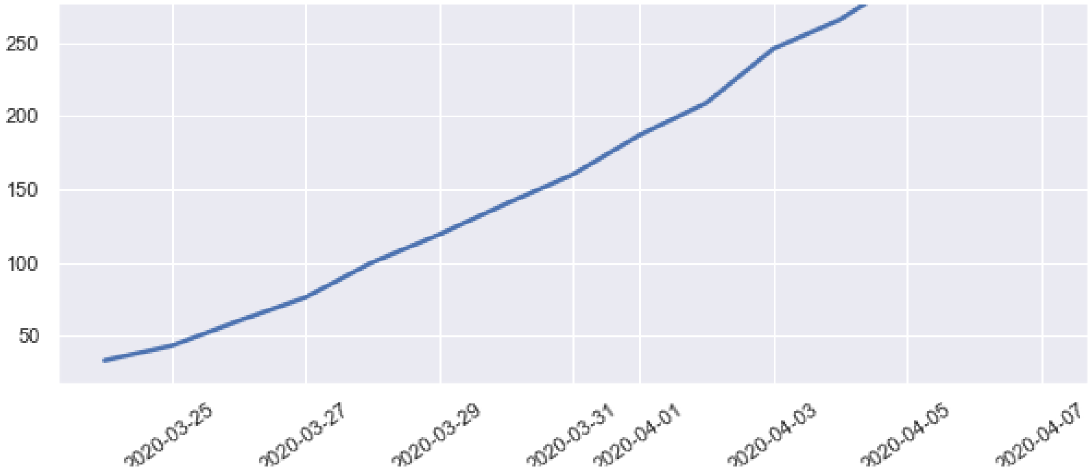


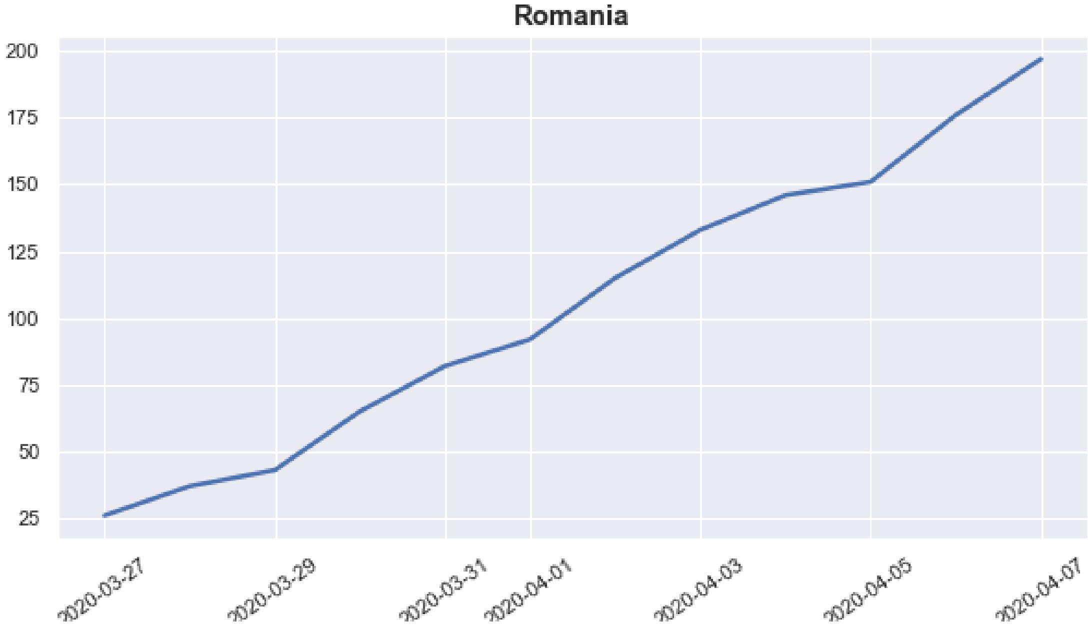
Poland

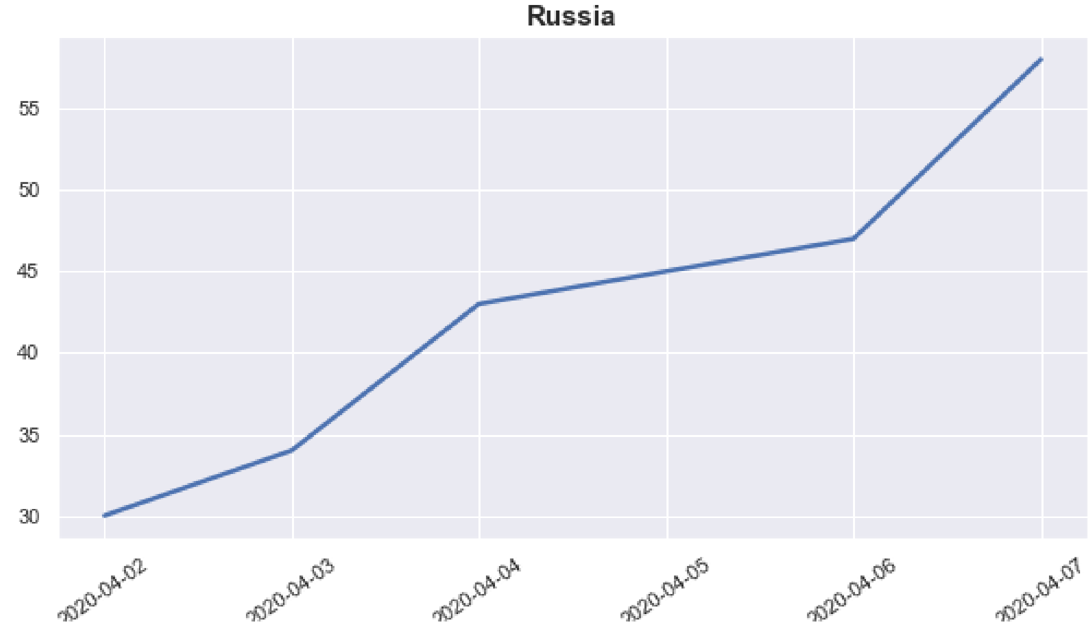


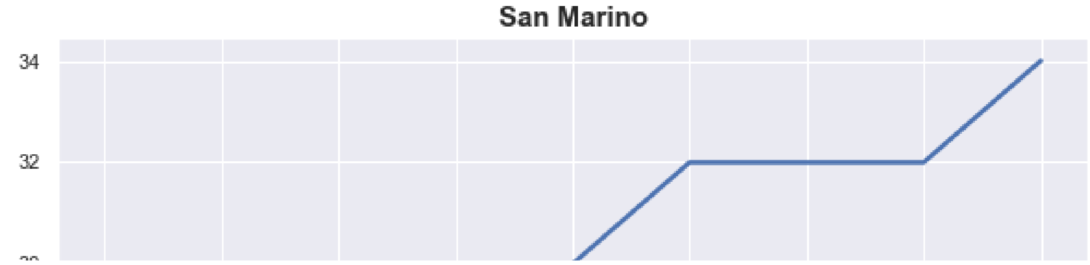
Portugal

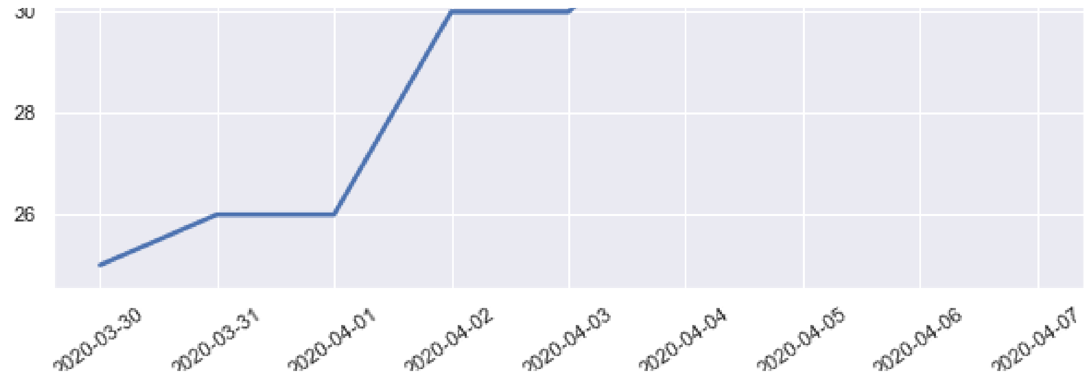




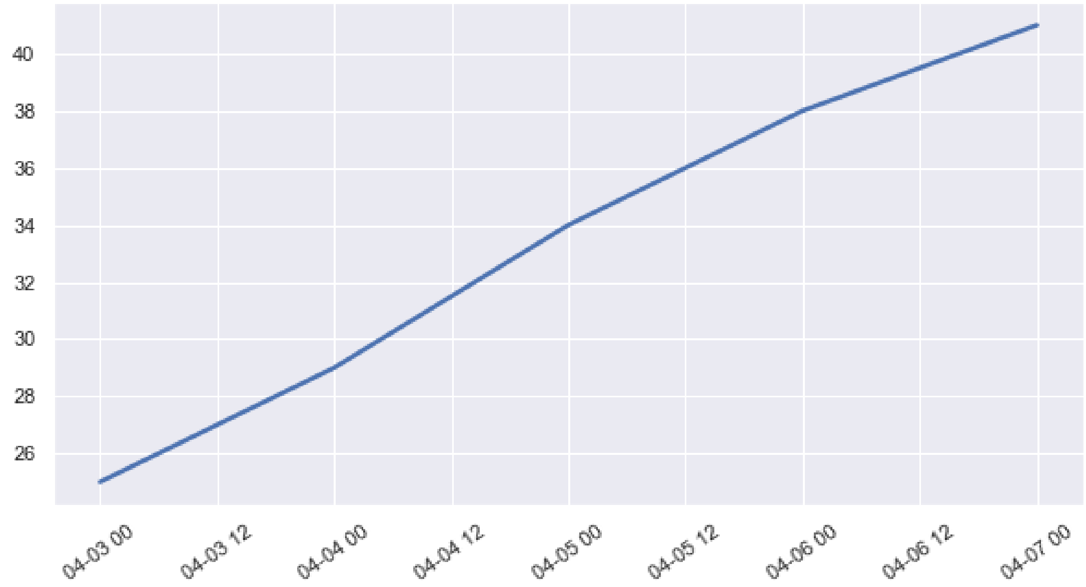

Romania


Russia

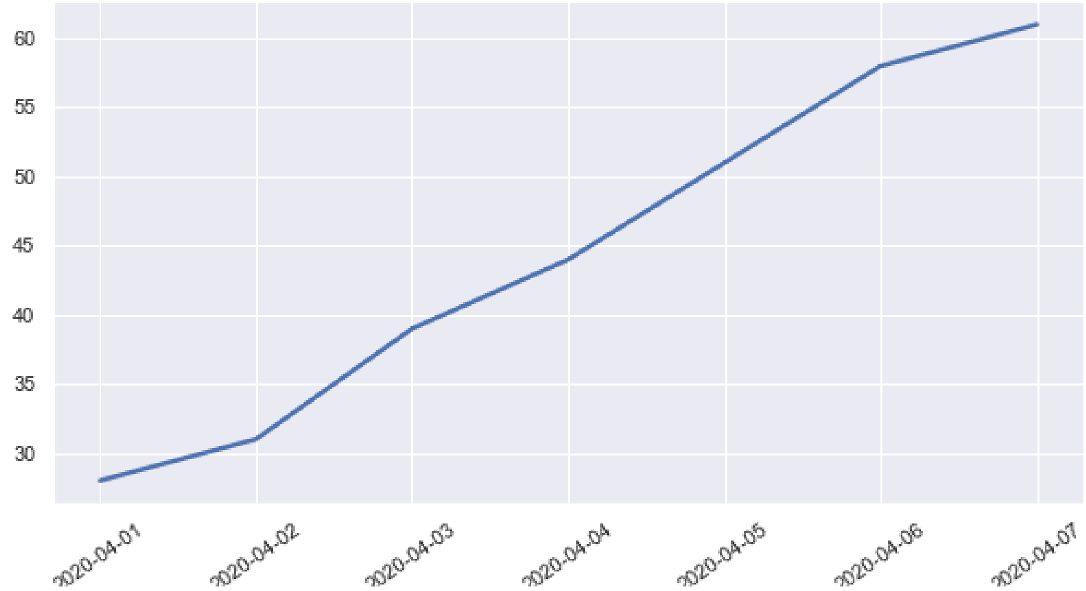

San Marino



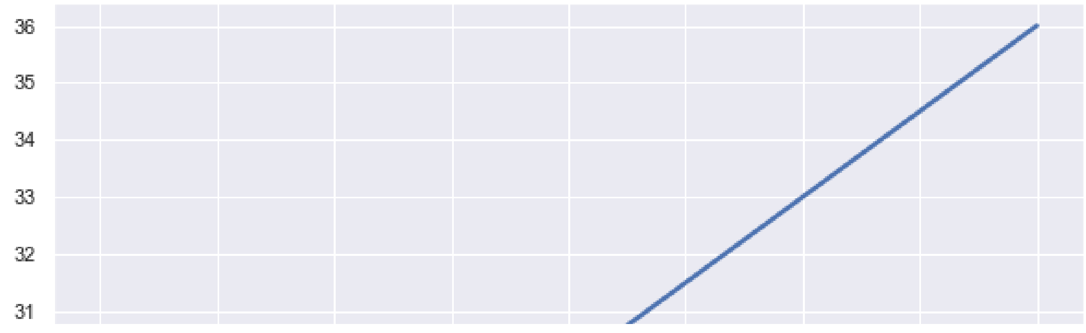
Saudi Arabia

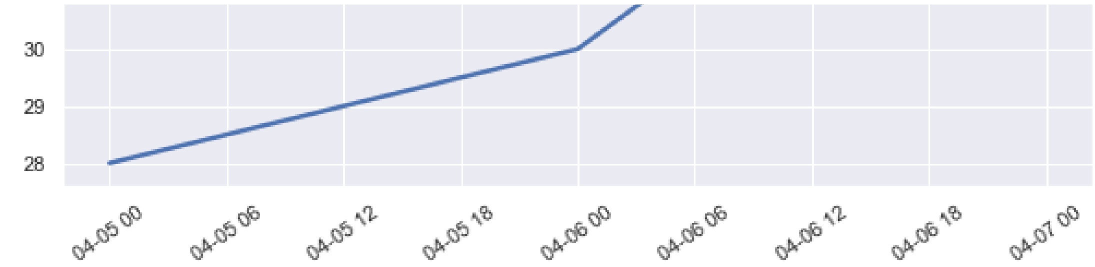


Serbia

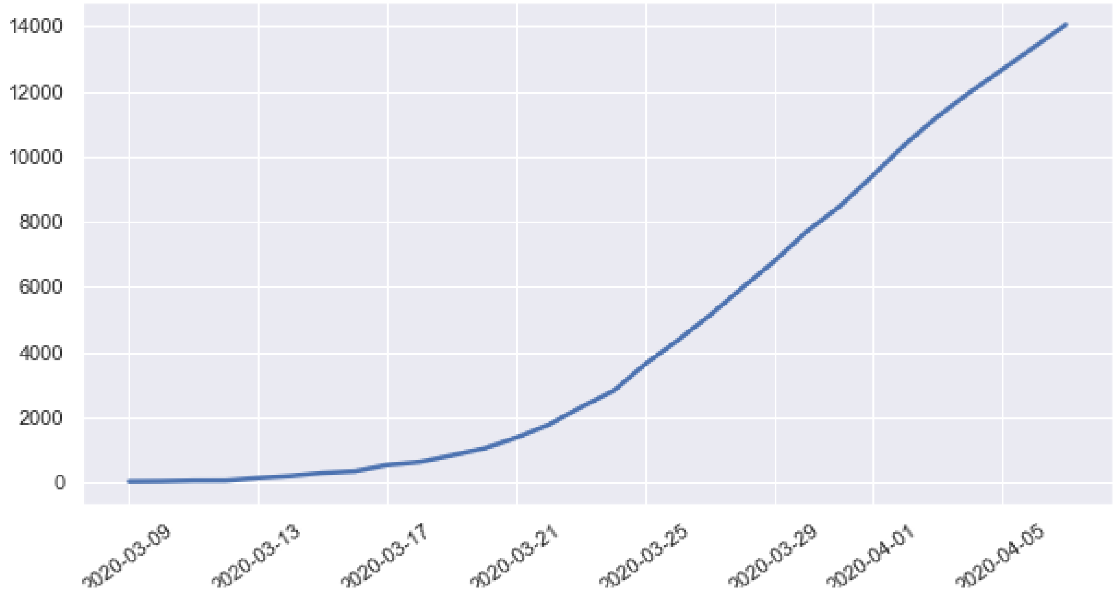


Slovenia

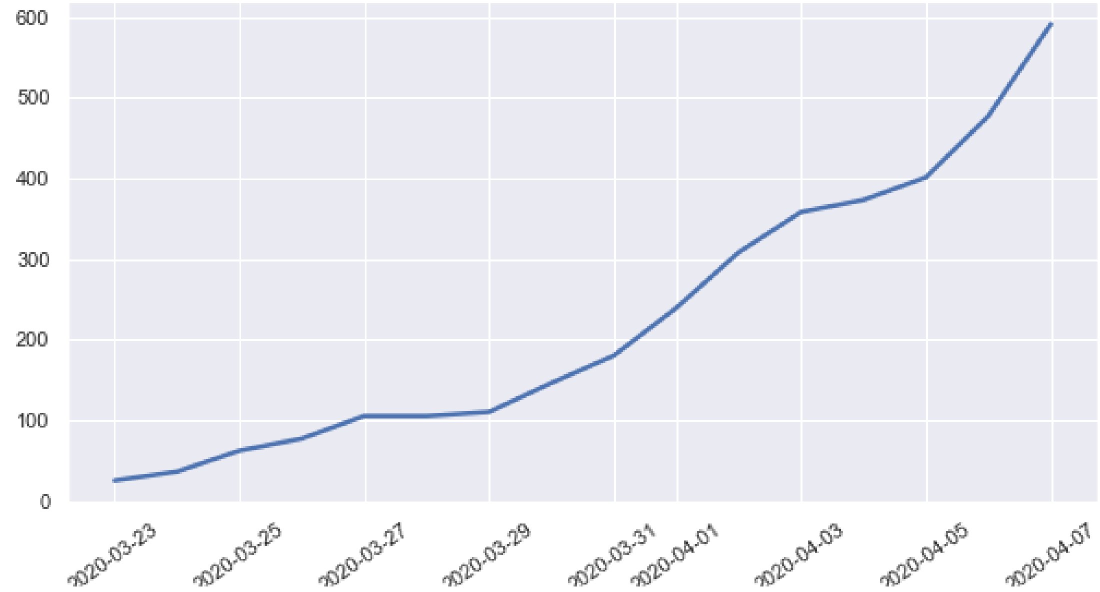




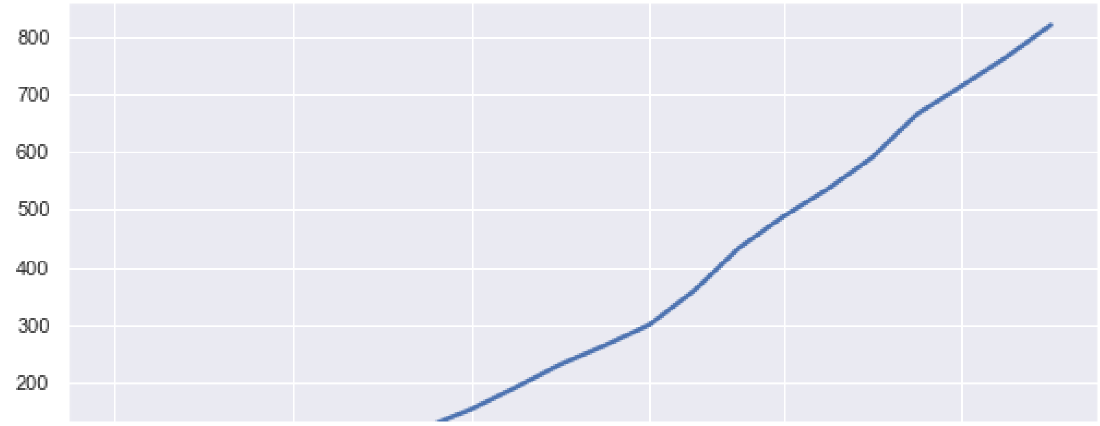
Spain

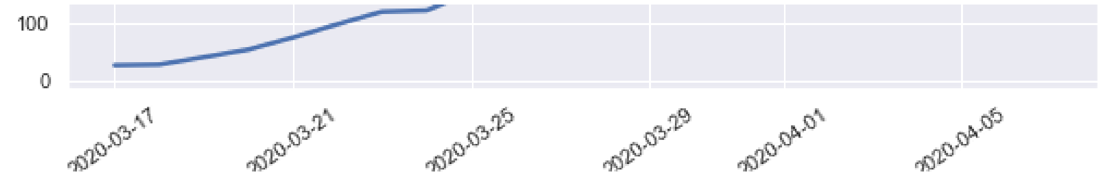


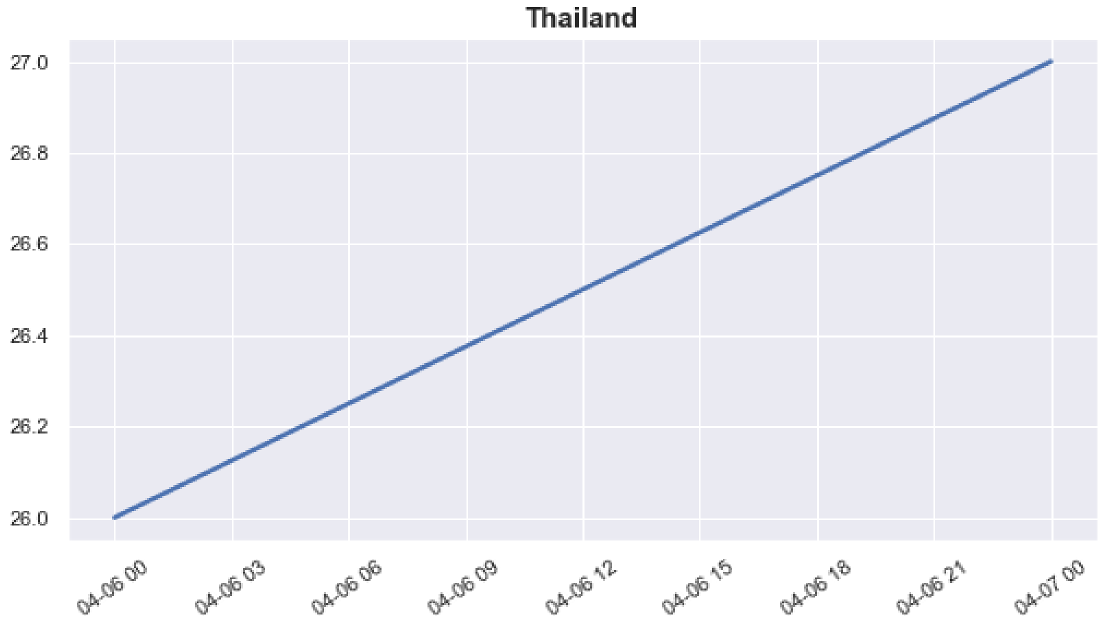
Sweden

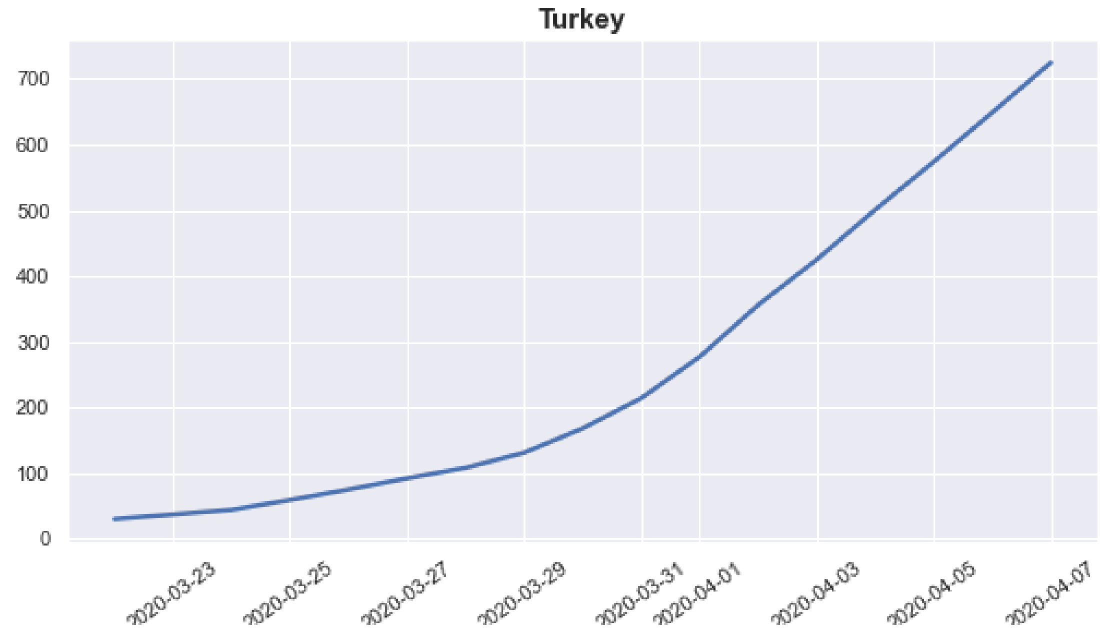


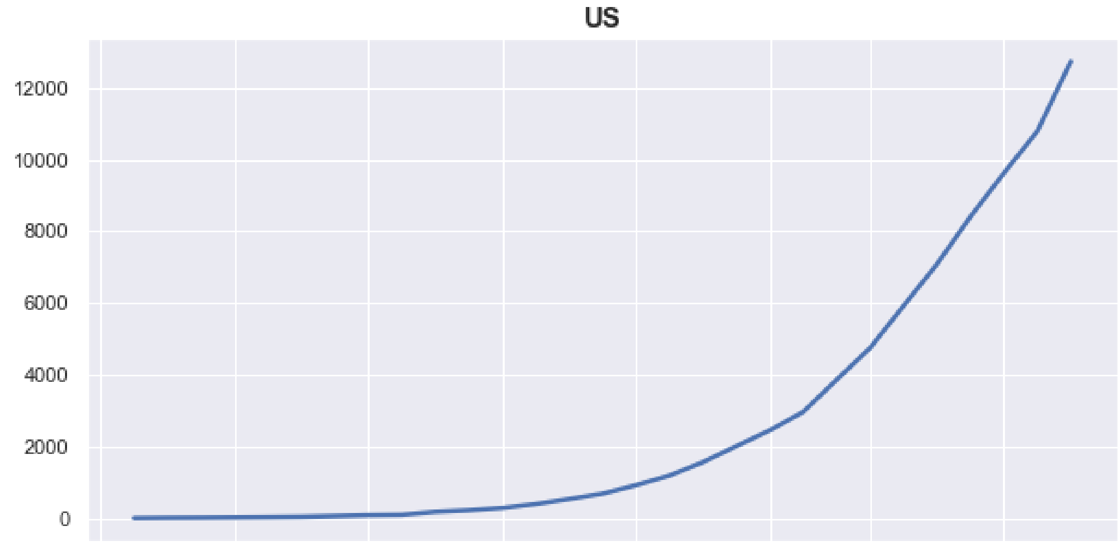
Switzerland

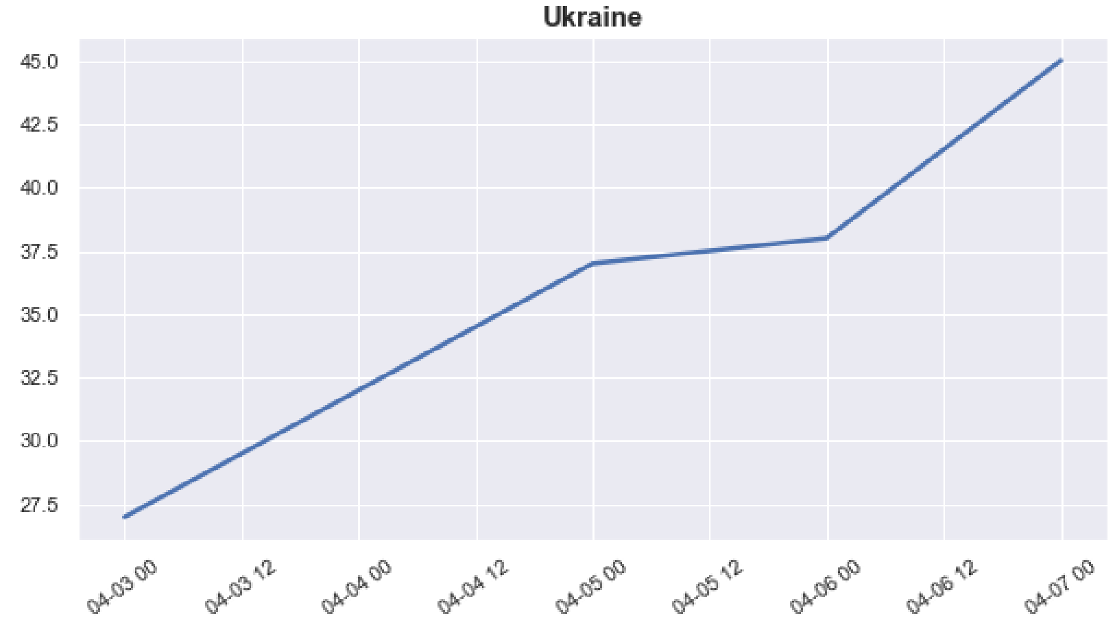


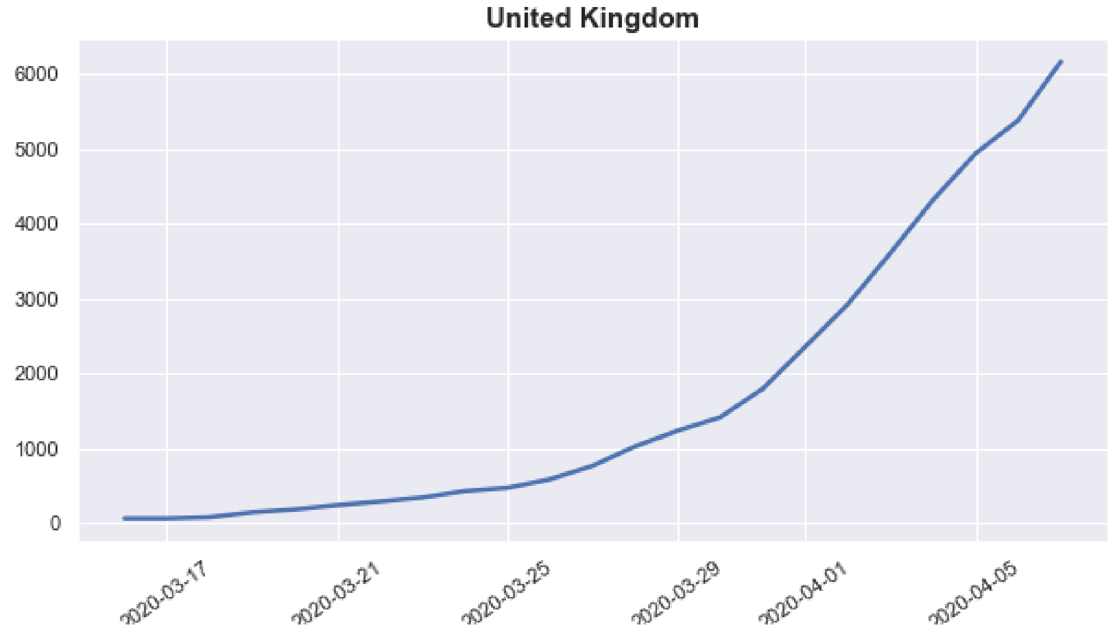



Thailand


Turkey


US





Kematian di dunia paling banyak dialami oleh US dan China.

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 [19]:

```
#type your function codes here
def pasien_sembuh():
    global recovered
    global world
    total = 0
    recovered = raw_data_recovered.groupby('Country/Region').sum()
    del recovered['Long']
    del recovered['Lat']
    recovered = recovered.T
    world = recovered[:].sum(axis = 1)

    max = recovered.max()
    for i in max:
        total += i

    display(recovered.head(5))
    display(recovered.tail(5))

    print("")
    print('\033[1m' + "Total Pasien COVID-19 yang dinyatakan sembuh adalah", total)
```

Periksa fungsi-fungsi yang sudah anda buat

In [20]:

```
pasien_sembuh()
```

Country/Region	Afghanistan	Albania	Algeria	Andorra	Angola	Antigua and Barbuda	Argentina	Armenia	Australia	Austria	...	United Kingdom
1/22/20	0	0	0	0	0	0	0	0	0	0	...	0
1/23/20	0	0	0	0	0	0	0	0	0	0	...	0
1/24/20	0	0	0	0	0	0	0	0	0	0	...	0
1/25/20	0	0	0	0	0	0	0	0	0	0	...	0
1/26/20	0	0	0	0	0	0	0	0	0	0	...	0

5 rows × 196 columns

Country/Region	Afghanistan	Albania	Algeria	Andorra	Angola	Antigua and Barbuda	Argentina	Armenia	Australia	Austria	...	United Kingdom
1/7/22	0	0	0	0	0	0	0	0	0	0	...	0
1/8/22	0	0	0	0	0	0	0	0	0	0	...	0
1/9/22	0	0	0	0	0	0	0	0	0	0	...	0
1/10/22	0	0	0	0	0	0	0	0	0	0	...	0
1/11/22	0	0	0	0	0	0	0	0	0	0	...	0

5 rows × 196 columns

```
Total Pasien COVID-19 yang dinyatakan sembuh adalah 137249981
```

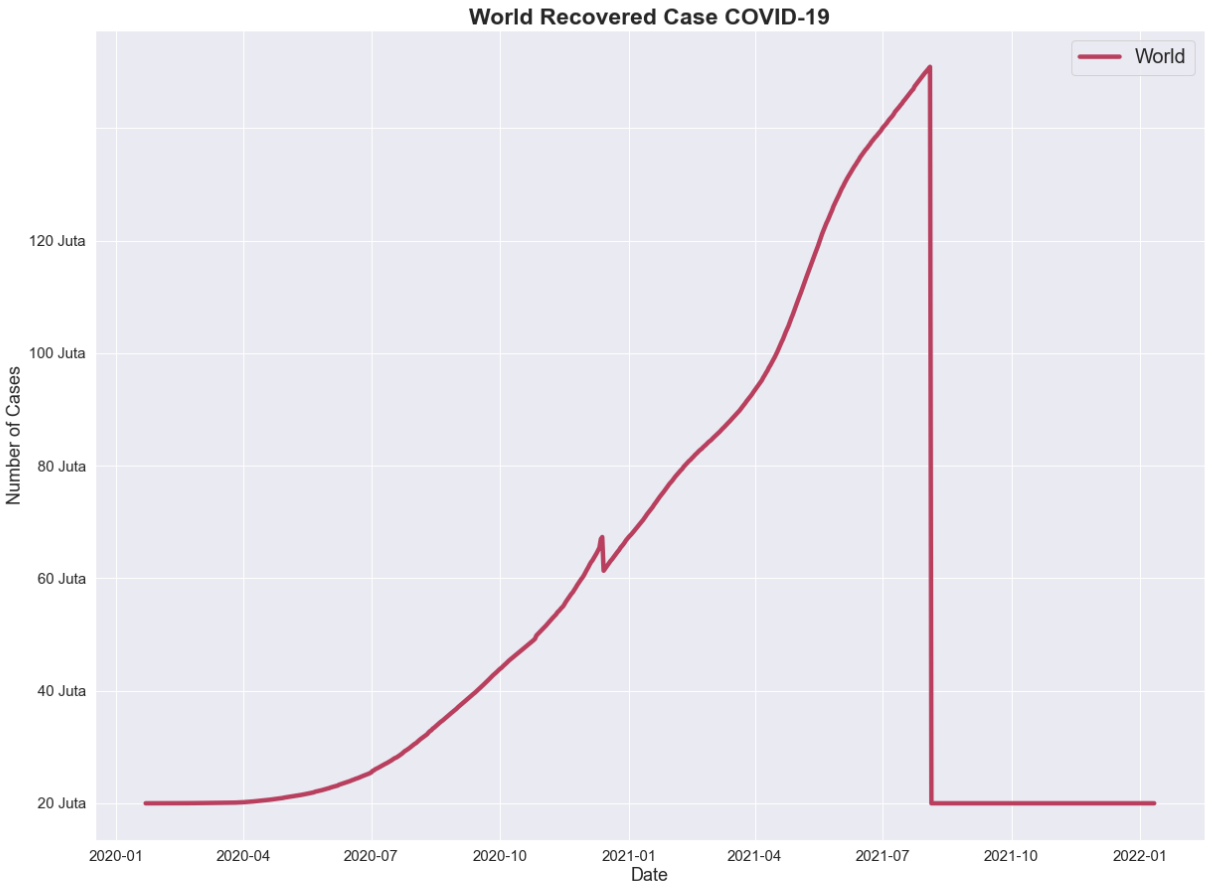
Selanjutnya,tampilkan data total pasien COVID-19 yang sembuh dalam bentuk grafik yang sesuai. Berikan judul, labels, dan spesifikasi (ukuran, warna, ketebalan, dll) yang sesuai sehingga plot yang dihasilkan rapi, menarik, dan mudah dipahami.

In [21]:

```
world.index = pd.to_datetime(world.index,infer_datetime_format=True)

#size chart
fig,ax = plt.subplots(figsize = (20, 15))

#insert plot
ax.plot(world, color = "#B8405E", linewidth = 4.5)
ax.legend(['World'], fontsize = 20)
ax.set_title('World Recovered Case COVID-19', fontsize = 23, fontweight = 'bold')
ax.set_xlabel('Date', fontsize = 18)
ax.set_ylabel('Number of Cases', fontsize = 18)
plt.xticks(size = 15)
ax.set_yticklabels(['0', '20 Juta', '40 Juta', '60 Juta', '80 Juta', '100 Juta', '120 Juta'], size = 15)
plt.show()
```



```
<ipython-input-21-90d314b7bee1>:13: UserWarning: FixedFormatter should only be used together with FixedLocator
  ax.set_yticklabels(['0', '20 Juta', '40 Juta', '60 Juta', '80 Juta', '100 Juta', '120 Juta'], size = 15)
```

Jumlah kasus yang sembuh dari covid-19 semakin banyak dan memiliki puncak di bulan 8 - 2021

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 [22]:

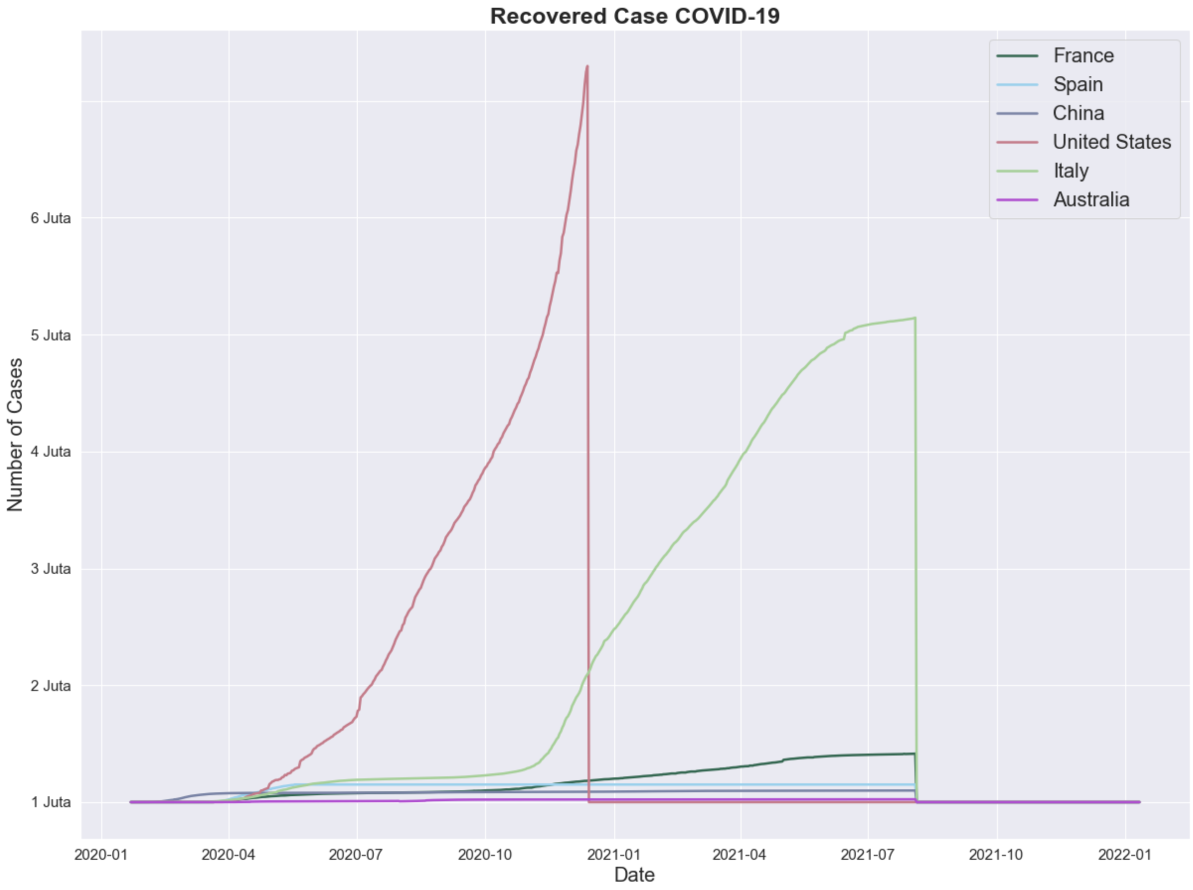
```
recovered.index = pd.to_datetime(recovered.index,infer_datetime_format=True)
```

In [23]:

```
prancis = recovered['France']
spanyol = recovered['Spain']
china = recovered['China']
us = recovered['US']
itali = recovered['Italy']
australia = recovered['Australia']

#size chart
fig,ax1 = plt.subplots(figsize = (20, 15))
```

```
#insert plot
plt.plot(prancis, color = "#346751", linewidth = 2.5)
plt.plot(spanyol, color = "#9AD0EC", linewidth = 2.5)
plt.plot(china, color = "#7882A4", linewidth = 2.5)
plt.plot(us, color = "#C37B89", linewidth = 2.5)
plt.plot(itali, color = "#A6CF98", linewidth = 2.5)
plt.plot(australia, color = "#AE4CCF", linewidth = 2.5)
plt.legend(['France','Spain','China','United States','Italy','Australia'], fontsize = 20
)
plt.title('Recovered Case COVID-19', fontsize = 23, fontweight = 'bold')
plt.xlabel('Date', fontsize = 20)
plt.ylabel('Number of Cases', fontsize = 20)
ax1.set_yticklabels(['0', '1 Juta', '2 Juta', '3 Juta', '4 Juta', '5 Juta', '6 Juta'], s
ize = 15)
plt.xticks(size = 15)
plt.show()
```



```
<ipython-input-23-df9114c49e46>:22: UserWarning: FixedFormatter should only be used toget
her with FixedLocator
  ax1.set_yticklabels(['0', '1 Juta', '2 Juta', '3 Juta', '4 Juta', '5 Juta', '6 Juta'],
size = 15)
```

Amerika memiliki kasus recovered COVID-19 tertinggi dari bulan 4-2020 hingga bulan 12-2020