An Exploration Of Pandas For A New User, And Visualising What Is Going On With Coronavirus

Craig Godden-Payne
Mar 11 · 9 min read ★

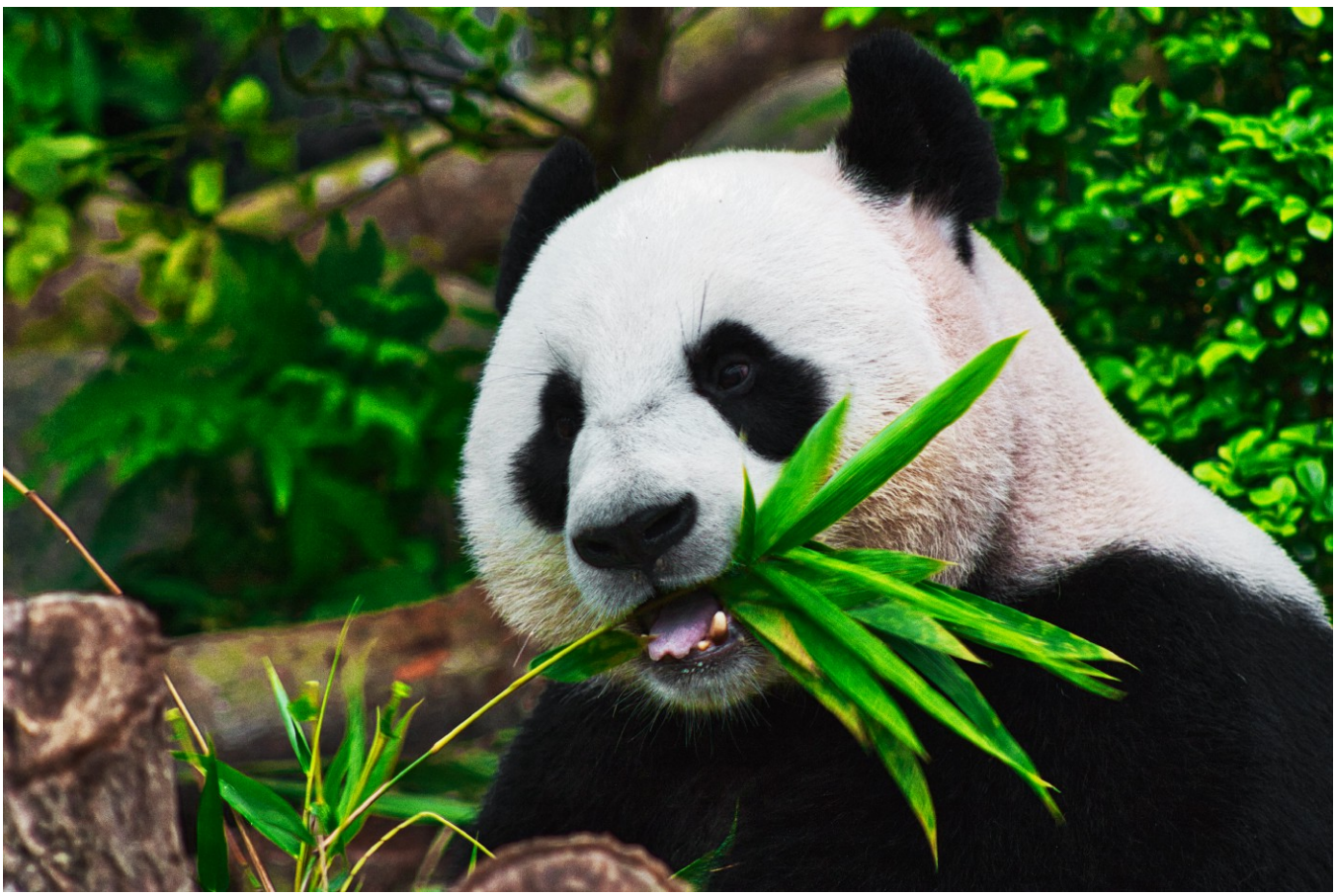


Photo by Sid Balachandran on Unsplash

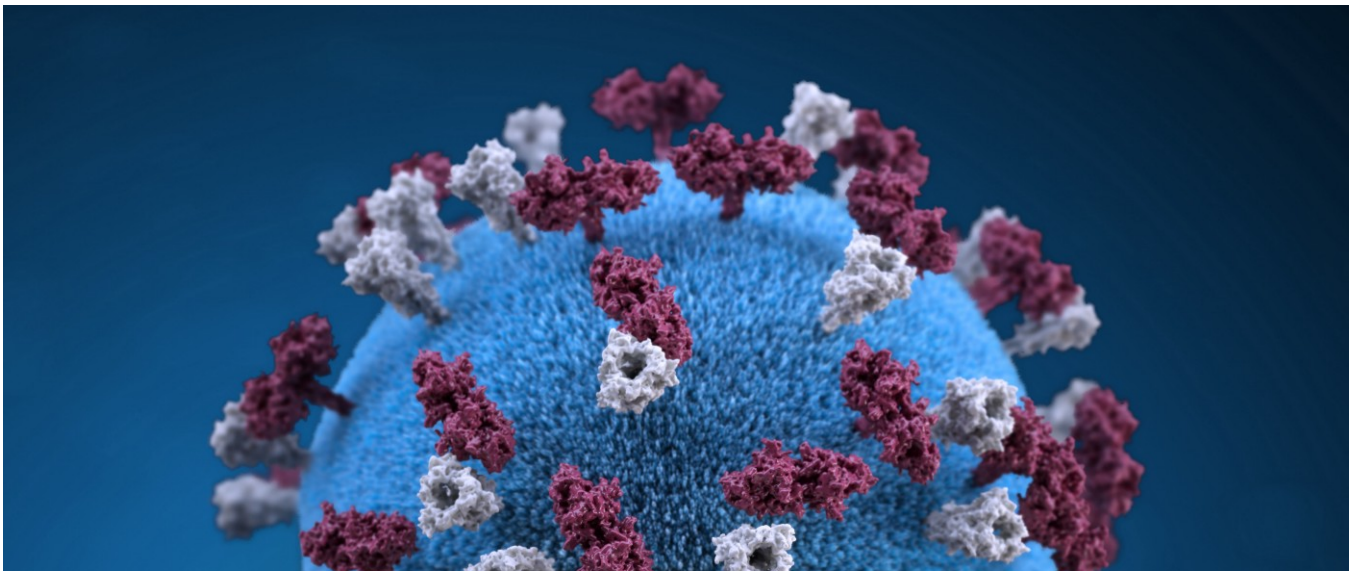
I've been learning Python for a few weeks, and recently been looking more into Pandas.

> *In computer programming,* ***pandas*** *is a software library written for the* ***Python*** *programming language for data manipulation and analysis. In particular, it offers data structures and operations for manipulating numerical tables and time series. It is free software released under the three-clause BSD license.*

Since coronavirus is all over the news, I thought it would be good to try and visualise some of this data.

I started off by installing jupyter notebooks and creating a new project. I then looked for a source to scrape data from. I settled on this source: https://raw.githubusercontent.com/datasets/covid-19/master/time-series-19-covid-combined.csv which lists fields I was interested in:

- Country
- Date
- Confirmed
- Recovered
- Deaths



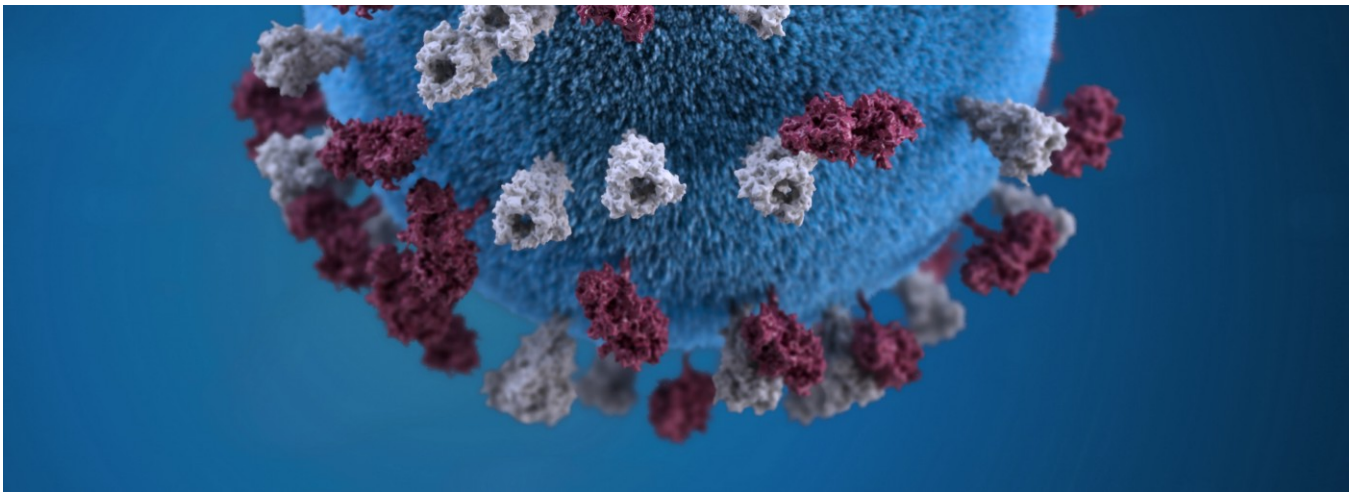


Photo by CDC on Unsplash

I've been flicking back and forth between jupyter and the pandas documentation, and I did manage to get some interesting information out, some of it a bit weird but I guess I need to play with it a bit more.

```
import pandas as pd
import matplotlib.pyplot as plt
import io
import requests
url="https://raw.githubusercontent.com/datasets/covid-
19/master/time-series-19-covid-combined.csv"
s=requests.get(url).content
dataset=pd.read_csv(io.StringIO(s.decode('utf-8')))

dataset.columns
```

This outputted a list of the columns the dataframe

```
['Province/State', 'Country/Region', 'Lat', 'Long', 'Date',
'Confirmed', 'Recovered', 'Deaths']
```

I then wanted to look at the countries in the dataframe, just to check the information I am working with

```
dataset['Country/Region'].unique()
```

Which resulted in

```
['Mainland China', 'Thailand', 'Japan', 'South Korea', 'Taiwan',
'US', 'Macau', 'Hong Kong', 'Singapore', 'Vietnam', 'France',
'Nepal', 'Malaysia', 'Canada', 'Australia', 'Cambodia', 'Sri Lanka',
'Germany', 'Finland', 'United Arab Emirates', 'Philippines',
'India', 'Italy', 'UK', 'Russia', 'Sweden', 'Spain', 'Belgium',
'Others', 'Egypt', 'Iran', 'Lebanon', 'Iraq', 'Oman', 'Afghanistan',
'Bahrain', 'Kuwait', 'Algeria', 'Croatia', 'Switzerland', 'Austria',
'Israel', 'Pakistan', 'Brazil', 'Georgia', 'Greece', 'North
Macedonia', 'Norway', 'Romania', 'Denmark', 'Estonia',
'Netherlands', 'San Marino', 'Belarus', 'Iceland', 'Lithuania',
'Mexico', 'New Zealand', 'Nigeria', 'Ireland', 'Luxembourg',
'Monaco', 'Qatar', 'Ecuador', 'Azerbaijan', 'Czech Republic',
'Armenia', 'Dominican Republic', 'Indonesia', 'Portugal', 'Andorra',
'Latvia', 'Morocco', 'Saudi Arabia', 'Senegal', 'Argentina',
'Chile', 'Jordan', 'Ukraine', 'Saint Barthelemy', 'Hungary', 'Faroe
Islands', 'Gibraltar', 'Liechtenstein', 'Poland', 'Tunisia',
'Palestine', 'Bosnia and Herzegovina', 'Slovenia', 'South Africa',
'Bhutan', 'Cameroon', 'Colombia', 'Costa Rica', 'Peru', 'Serbia',
'Slovakia', 'Togo', 'Vatican City', 'French Guiana', 'Malta',
'Martinique', 'Bulgaria', 'Maldives', 'Bangladesh', 'Moldova',
'Paraguay', 'Albania', 'Cyprus', 'St. Martin', 'Brunei']
```

Photo by T.H. Chia on Unsplash

Awesome, so I thought I would limit the columns to what I wanted, so I ran this:

```
a =
dataset[['Country/Region','Date','Confirmed','Recovered','Deaths']]
a.head().to_csv()
```

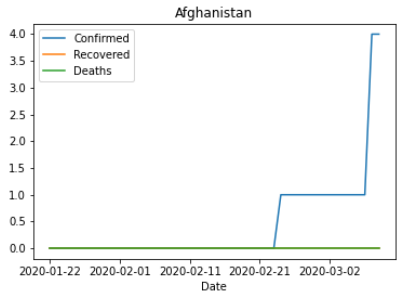
Which resulted in

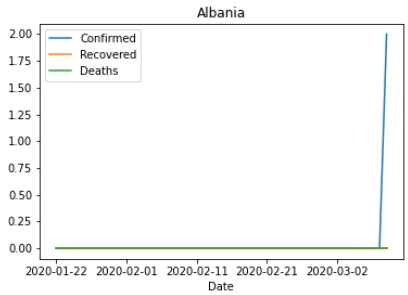
```
,Country/Region,Date,Confirmed,Recovered,Deaths
0,Mainland China,2020-01-22,1,0,0
1,Mainland China,2020-01-23,9,0,0
2,Mainland China,2020-01-24,15,0,0
3,Mainland China,2020-01-25,39,0,0
4,Mainland China,2020-01-26,60,0,0
```

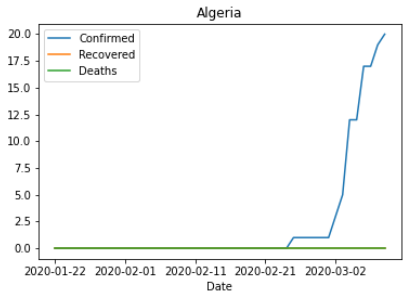
Now I need to change the index to be the date, and then for each country in the dataframe, plot a graph showing the Confirmed, Recovered and Deaths

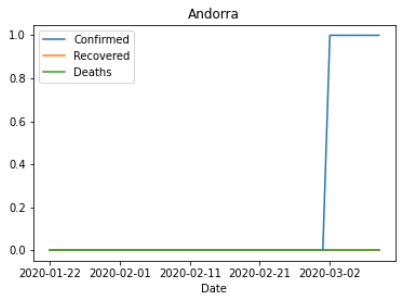
```
b=a.set_index('Date')
for country in b['Country/Region'].unique():
    c=b[b['Country/Region']==country]
    c.plot(title=country)
    plt.savefig(country+'.png')
```

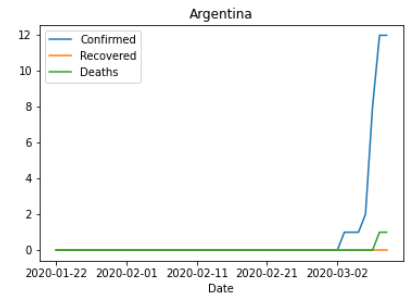
I found it quite cool for my first jump into Pandas, anyway here are the graphs I produced!

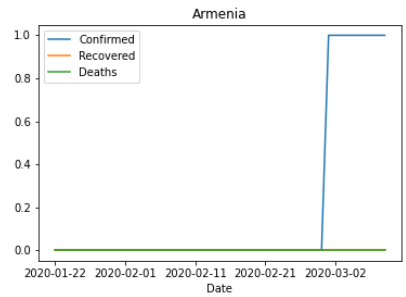


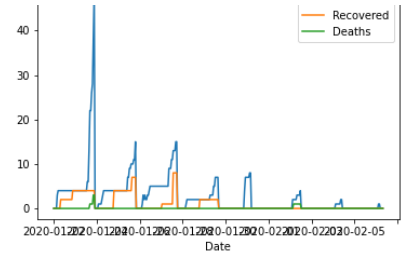


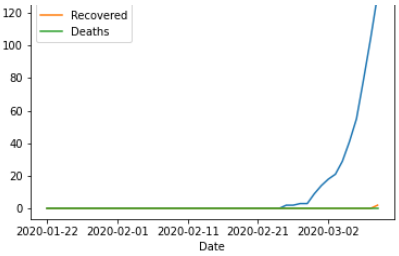


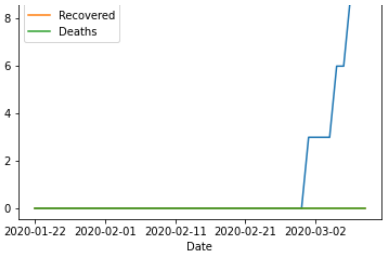


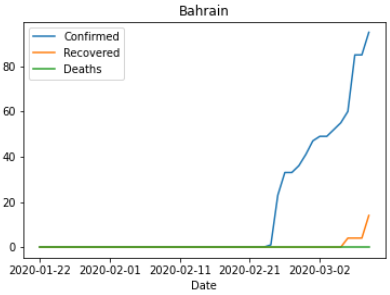


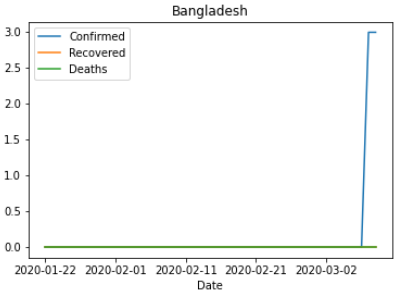


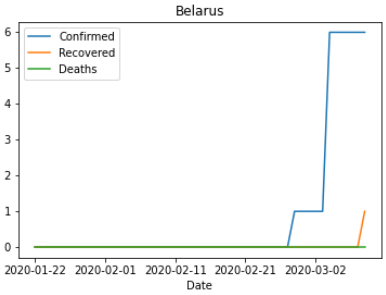

Recovered
Deaths
Date

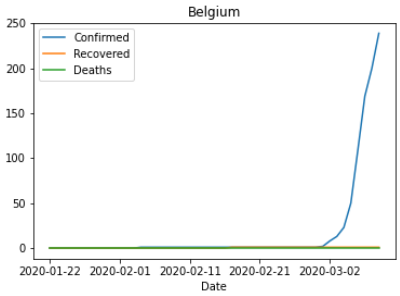

Recovered
Deaths
Date

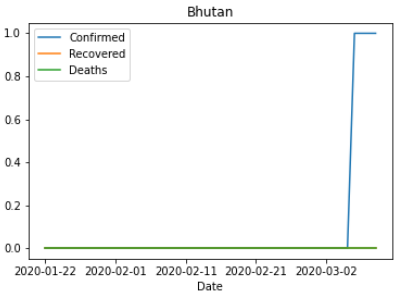

Recovered
Deaths
Date

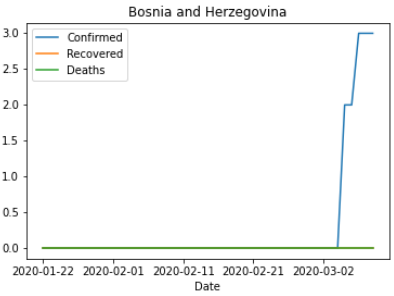

Bahrain
Confirmed
Recovered
Deaths
Date

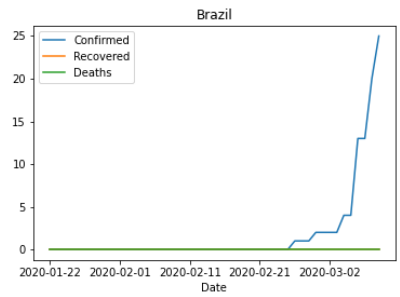

Bangladesh
Confirmed
Recovered
Deaths
Date

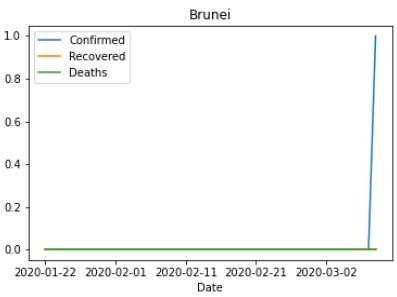

Belarus
Confirmed
Recovered
Deaths
Date

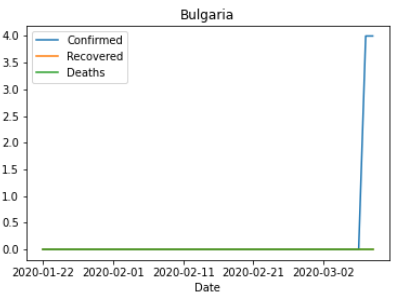

Belgium
Confirmed
Recovered
Deaths
Date

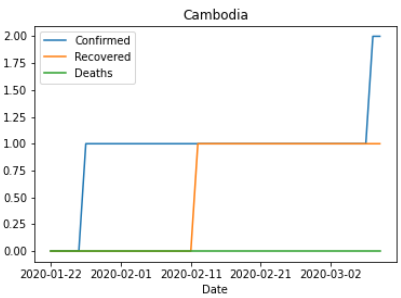

Bhutan
Confirmed
Recovered
Deaths
Date

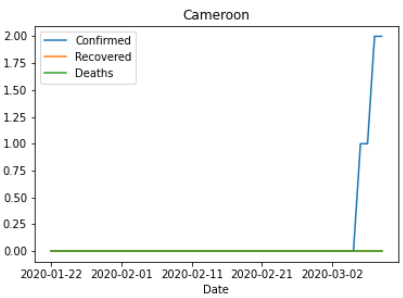

Bosnia and Herzegovina
Confirmed
Recovered
Deaths
Date

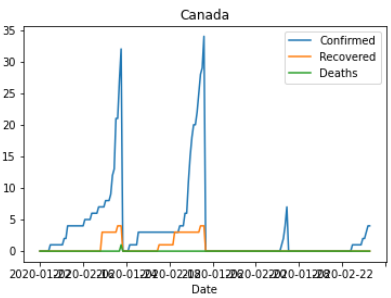

Brazil
Confirmed
Recovered
Deaths
Date

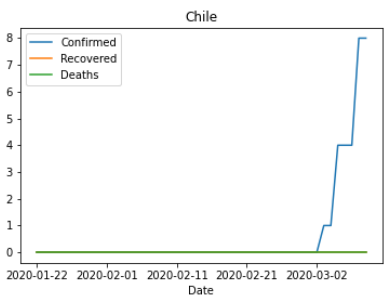

Brunei
Confirmed
Recovered
Deaths
Date

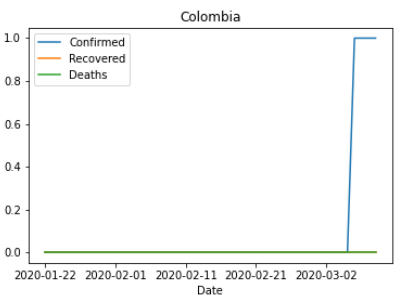

Bulgaria
Confirmed
Recovered
Deaths
Date

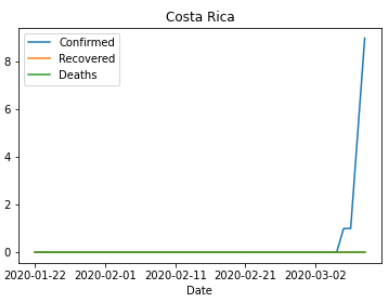

Cambodia
Confirmed
Recovered
Deaths
Date


Cameroon
Confirmed
Recovered
Deaths
Date

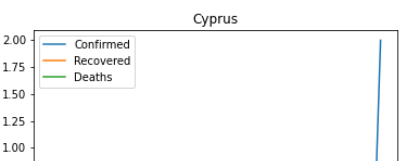

Canada
Confirmed
Recovered
Deaths
Date


Chile
Confirmed
Recovered
Deaths
Date

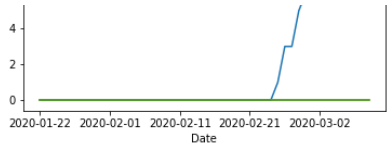

Colombia
Confirmed
Recovered
Deaths
Date

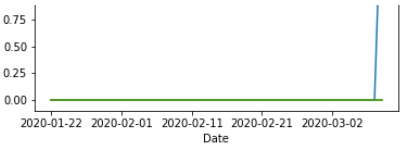

Costa Rica
Confirmed
Recovered
Deaths
Date

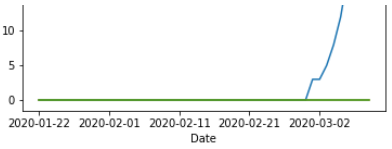

Croatia
Confirmed
Recovered
Deaths

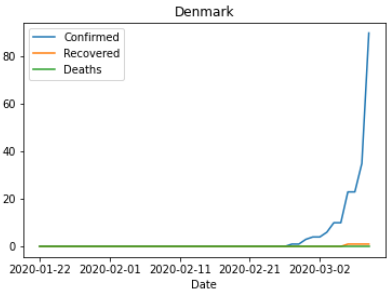

Cyprus
Confirmed
Recovered
Deaths

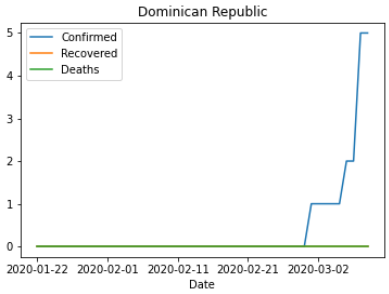

Czech Republic
Confirmed
Recovered
Deaths

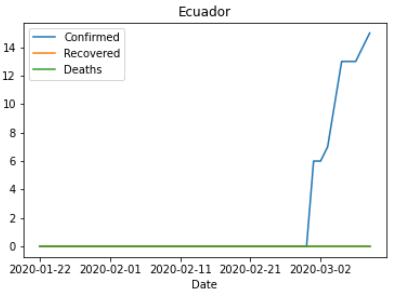

Confirmed
Recovered
Deaths
Date

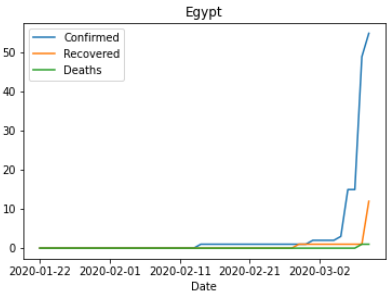

Confirmed
Recovered
Deaths
Date

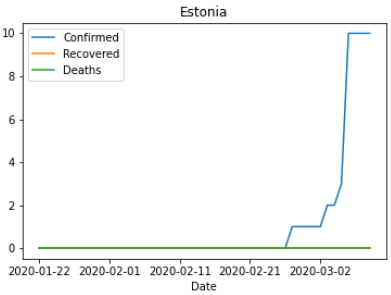

Confirmed
Recovered
Deaths
Date

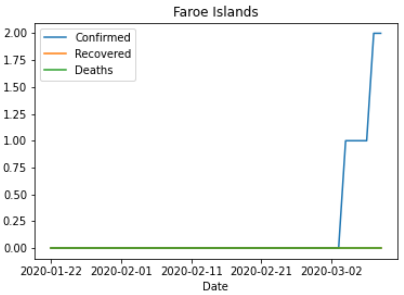

Denmark
Confirmed
Recovered
Deaths
Date

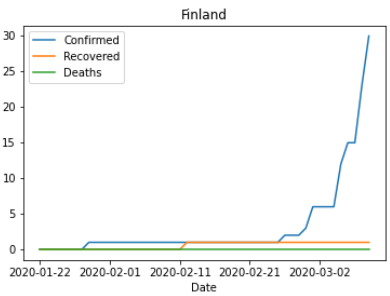

Dominican Republic
Confirmed
Recovered
Deaths
Date

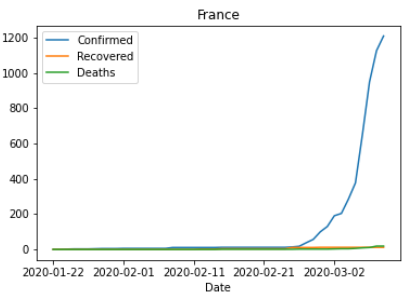

Ecuador
Confirmed
Recovered
Deaths
Date

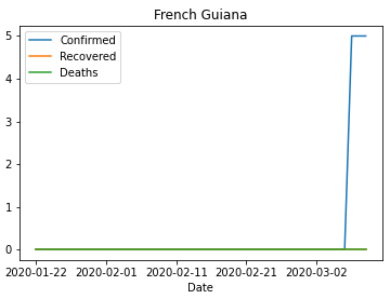

Egypt
Confirmed
Recovered
Deaths
Date

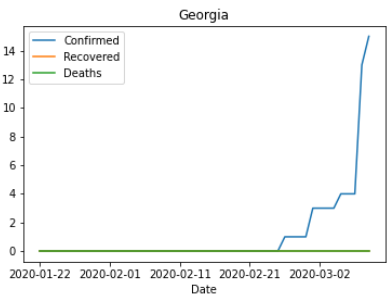

Estonia
Confirmed
Recovered
Deaths
Date

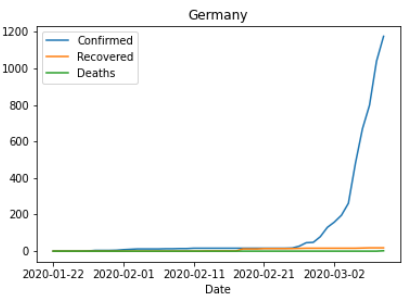

Faroe Islands
Confirmed
Recovered
Deaths
Date

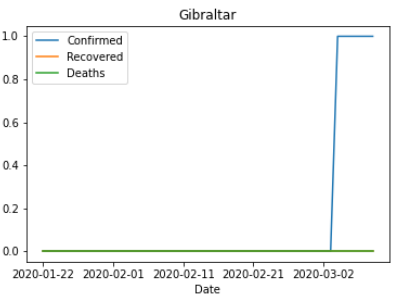

Finland
Confirmed
Recovered
Deaths
Date

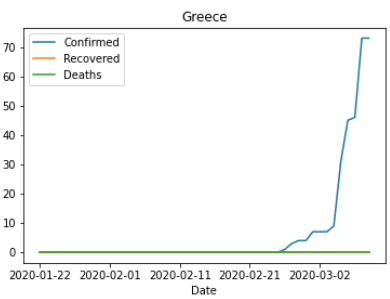

France
Confirmed
Recovered
Deaths
Date

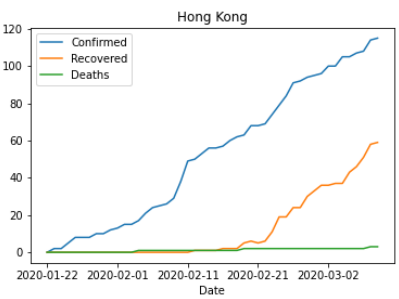

French Guiana
Confirmed
Recovered
Deaths
Date

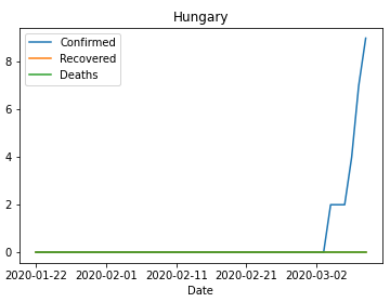

Georgia
Confirmed
Recovered
Deaths
Date

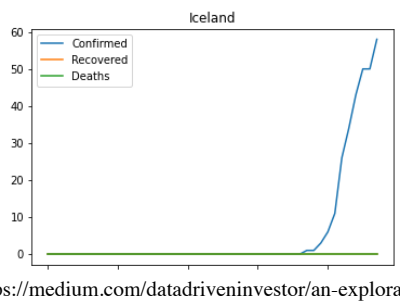

Germany
Confirmed
Recovered
Deaths
Date

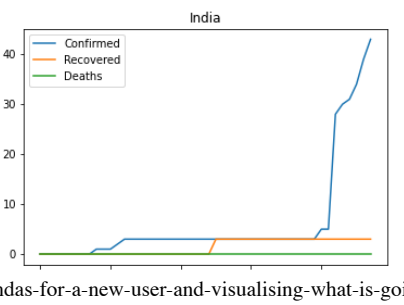

Gibraltar
Confirmed
Recovered
Deaths
Date

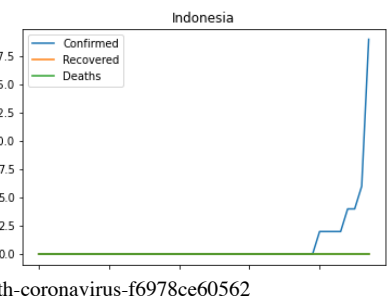

Greece
Confirmed
Recovered
Deaths
Date

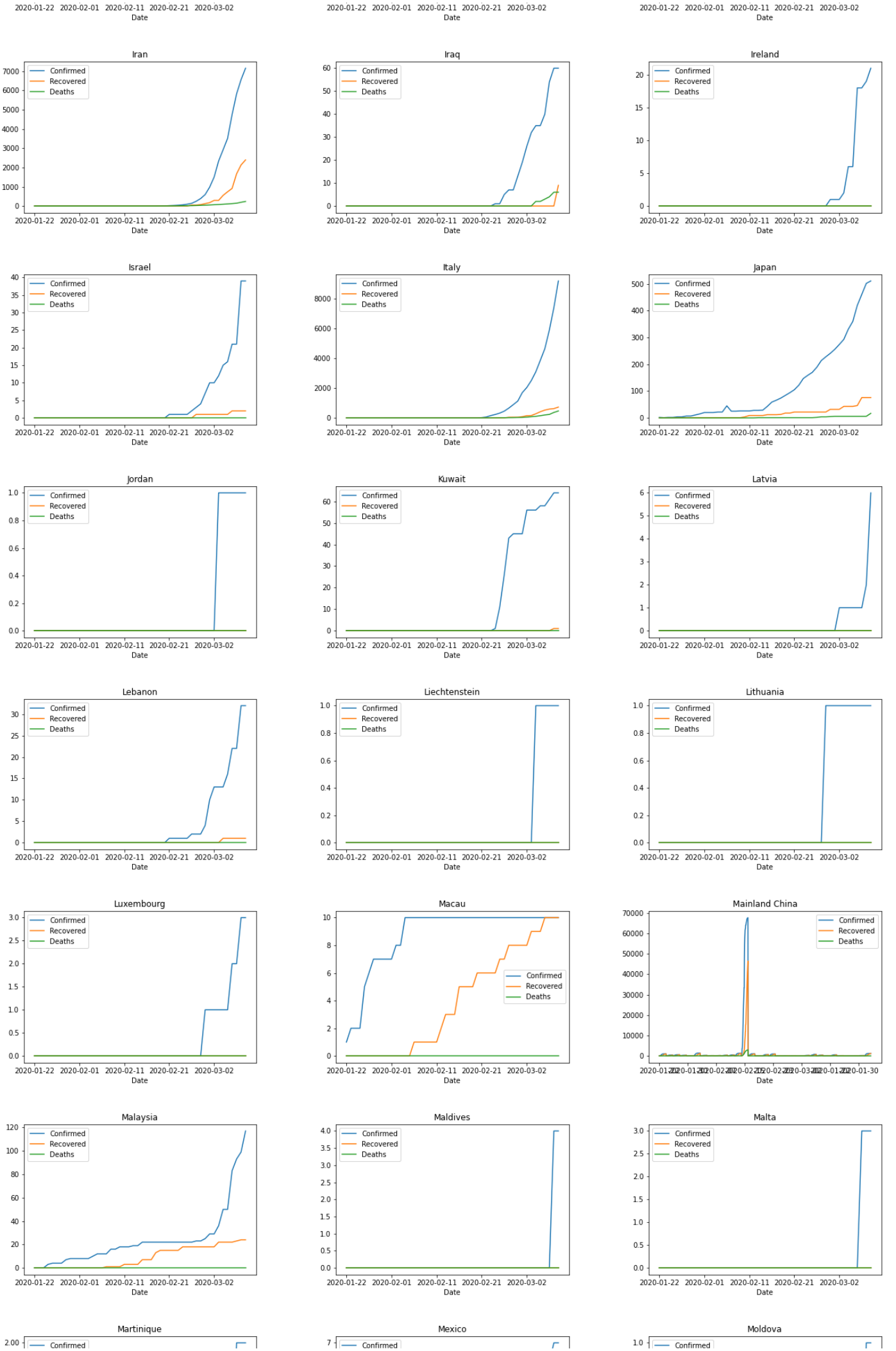

Hong Kong
Confirmed
Recovered
Deaths
Date

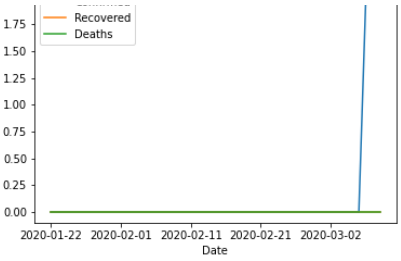

Hungary
Confirmed
Recovered
Deaths
Date

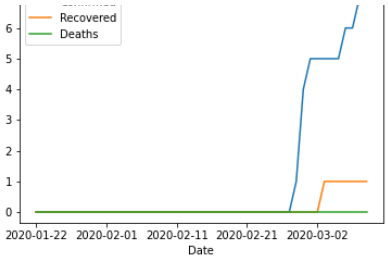

Iceland
Confirmed
Recovered
Deaths

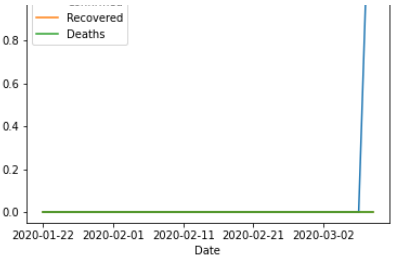

India
Confirmed
Recovered
Deaths

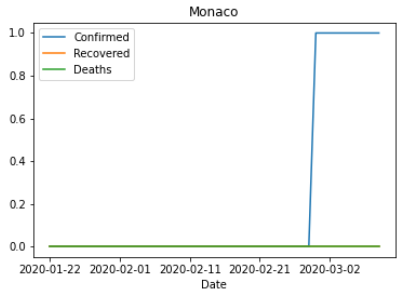

Indonesia
Confirmed
Recovered
Deaths

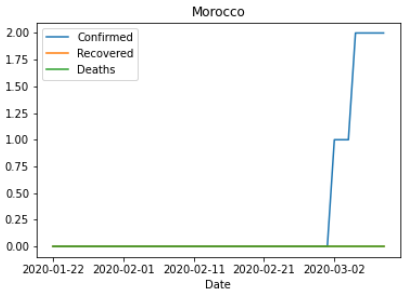

Iran
Iraq
Ireland
Israel
Italy
Japan
Jordan
Kuwait
Latvia
Lebanon
Liechtenstein
Lithuania
Luxembourg
Macau
Mainland China
Malaysia
Maldives
Malta
Martinique
Mexico
Moldova
Confirmed
Recovered
Deaths
Date

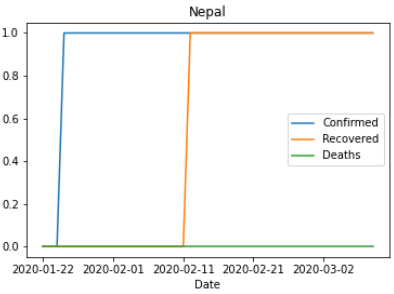

Recovered
Deaths
Date

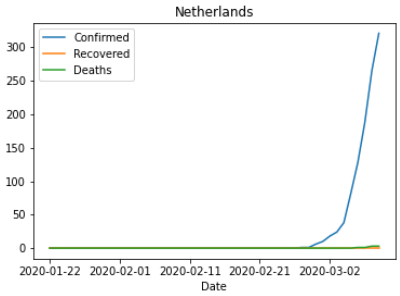

Recovered
Deaths
Date

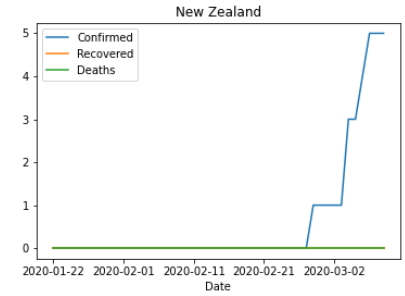

Recovered
Deaths
Date

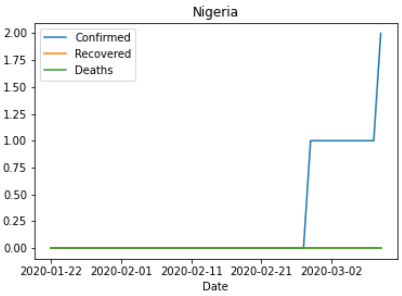

Monaco
Confirmed
Recovered
Deaths
Date

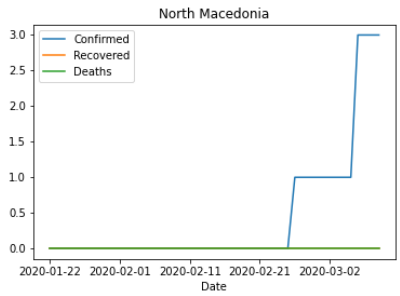

Morocco
Confirmed
Recovered
Deaths
Date

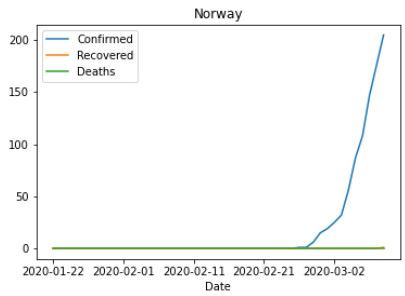

Nepal
Confirmed
Recovered
Deaths
Date

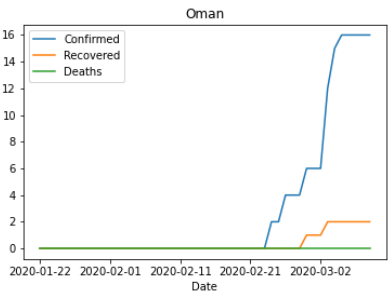

Netherlands
Confirmed
Recovered
Deaths
Date

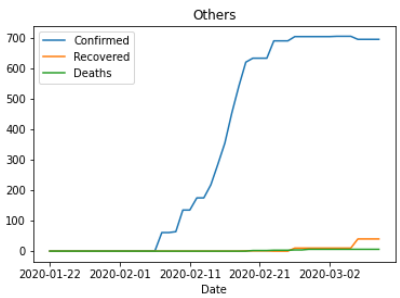

New Zealand
Confirmed
Recovered
Deaths
Date

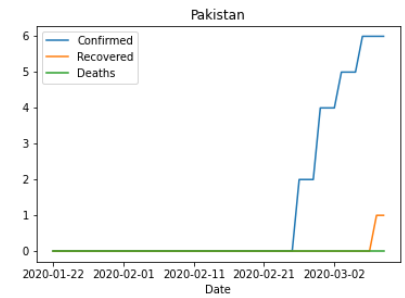

Nigeria
Confirmed
Recovered
Deaths
Date

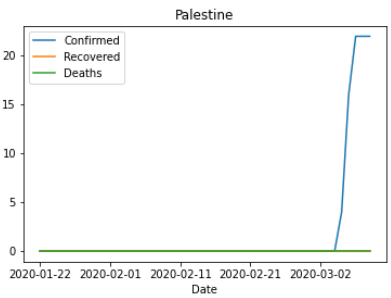

North Macedonia
Confirmed
Recovered
Deaths
Date

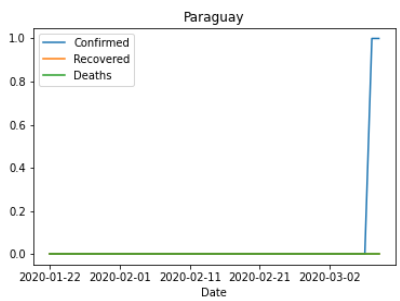

Norway
Confirmed
Recovered
Deaths
Date

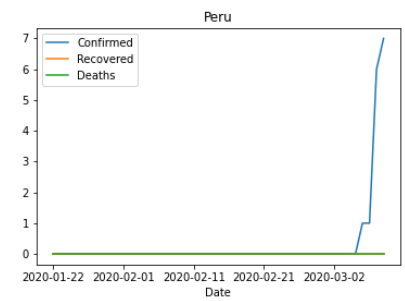

Oman
Confirmed
Recovered
Deaths
Date

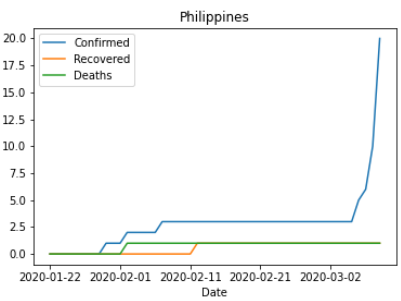

Others
Confirmed
Recovered
Deaths
Date


Pakistan
Confirmed
Recovered
Deaths
Date


Palestine
Confirmed
Recovered
Deaths
Date

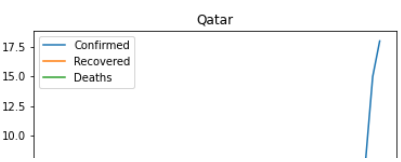

Paraguay
Confirmed
Recovered
Deaths
Date

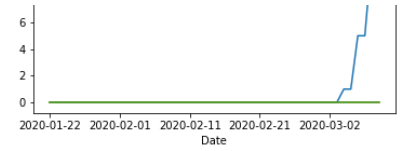

Peru
Confirmed
Recovered
Deaths
Date

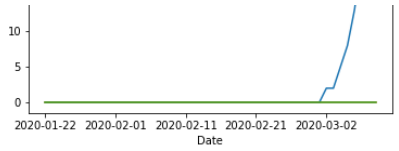

Philippines
Confirmed
Recovered
Deaths
Date

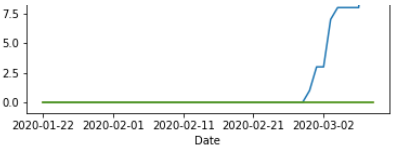

Poland
Confirmed
Recovered
Deaths

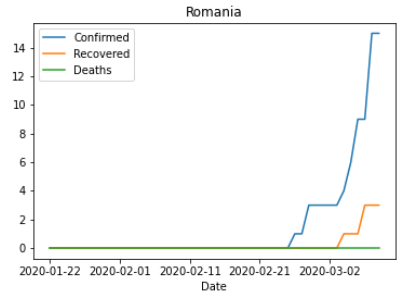

Portugal
Confirmed
Recovered
Deaths

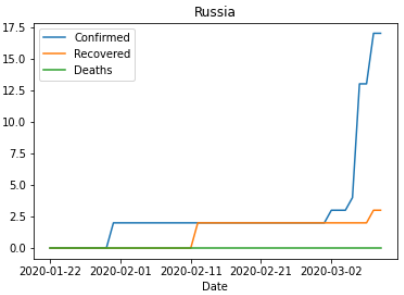

Qatar
Confirmed
Recovered
Deaths

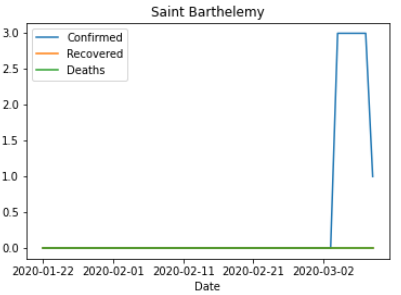

Date

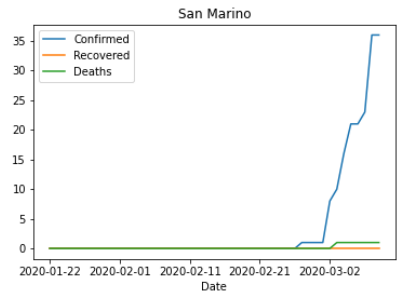

Date

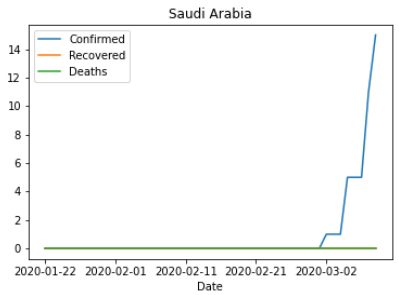

Date

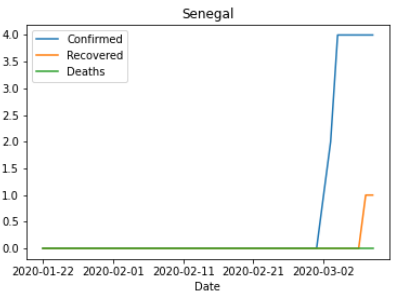

Romania
Confirmed
Recovered
Deaths
Date

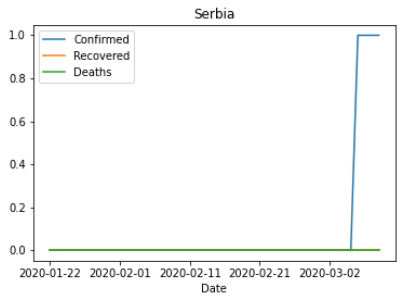

Russia
Confirmed
Recovered
Deaths
Date

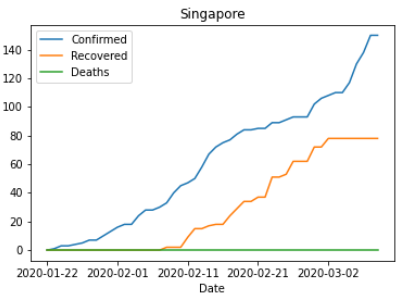

Saint Barthelemy
Confirmed
Recovered
Deaths
Date

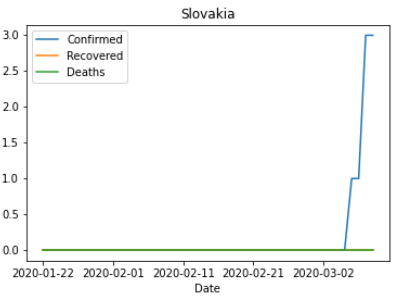

San Marino
Confirmed
Recovered
Deaths
Date

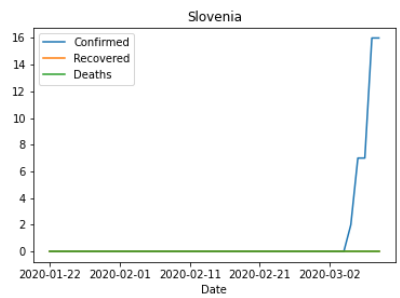

Saudi Arabia
Confirmed
Recovered
Deaths
Date

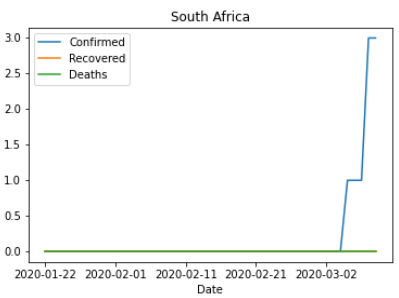

Senegal
Confirmed
Recovered
Deaths
Date

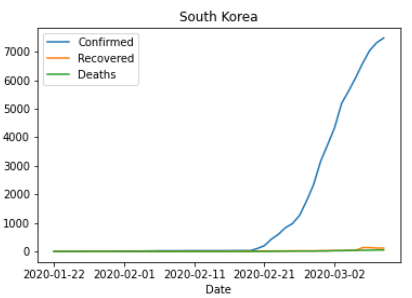

Serbia
Confirmed
Recovered
Deaths
Date

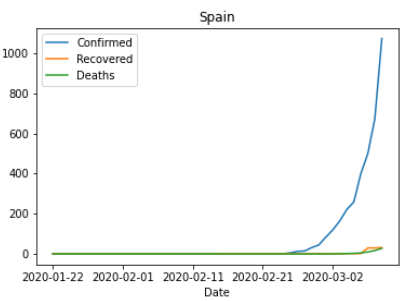

Singapore
Confirmed
Recovered
Deaths
Date

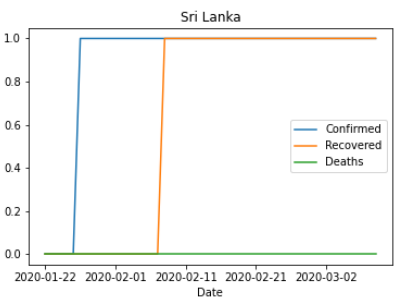

Slovakia
Confirmed
Recovered
Deaths
Date

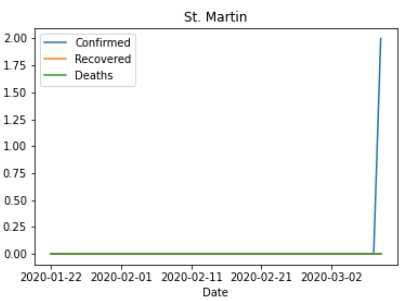

Slovenia
Confirmed
Recovered
Deaths
Date

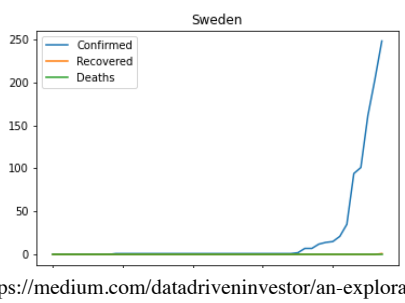

South Africa
Confirmed
Recovered
Deaths
Date

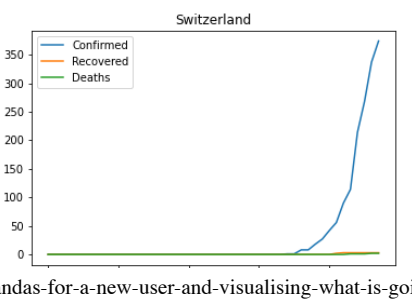

South Korea
Confirmed
Recovered
Deaths
Date

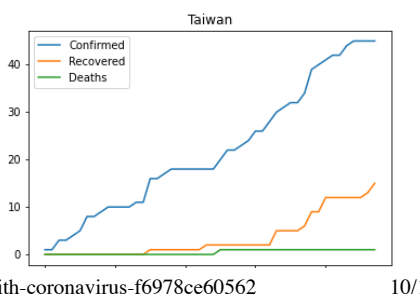

Spain
Confirmed
Recovered
Deaths
Date

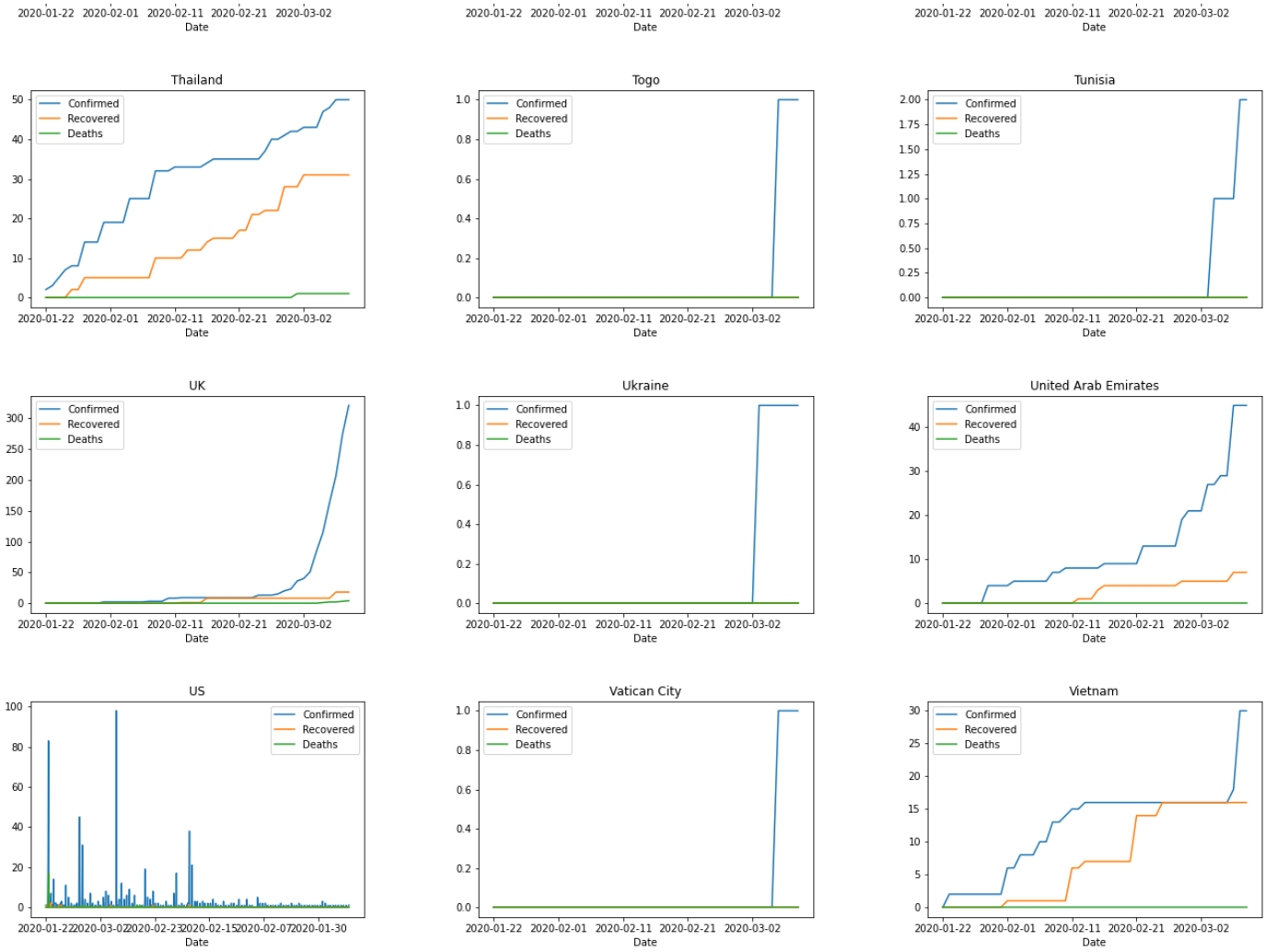

Sri Lanka
Confirmed
Recovered
Deaths
Date


St. Martin
Confirmed
Recovered
Deaths
Date


Sweden
Confirmed
Recovered
Deaths


Switzerland
Confirmed
Recovered
Deaths


Taiwan
Confirmed
Recovered
Deaths



I then tried something a bit different

4.2K signups

Pandas Data Analysis Visualization Data Science