

```
In [4]: import matplotlib.pyplot as plt
import pandas as pd
import numpy as np
from sklearn.impute import SimpleImputer
```

```
In [6]: df = pd.read_csv(r'C:\Users\SAHIL PANDITA\OneDrive\Documents\RAW DATA\covid_19_data.csv')
```

```
In [9]: df.head(50)
```

```
Out[9]:
```

	SNo	ObservationDate	Province/State	Country/Region	Last Update	Confirmed	Deaths	Recovered
0	1	01/22/2020	Anhui	Mainland China	1/22/2020 17:00	1.0	0.0	0.0
1	2	01/22/2020	Beijing	Mainland China	1/22/2020 17:00	14.0	0.0	0.0
2	3	01/22/2020	Chongqing	Mainland China	1/22/2020 17:00	6.0	0.0	0.0
3	4	01/22/2020	Fujian	Mainland China	1/22/2020 17:00	1.0	0.0	0.0
4	5	01/22/2020	Gansu	Mainland China	1/22/2020 17:00	0.0	0.0	0.0
5	6	01/22/2020	Guangdong	Mainland China	1/22/2020 17:00	26.0	0.0	0.0
6	7	01/22/2020	Guangxi	Mainland China	1/22/2020 17:00	2.0	0.0	0.0
7	8	01/22/2020	Guizhou	Mainland China	1/22/2020 17:00	1.0	0.0	0.0
8	9	01/22/2020	Hainan	Mainland China	1/22/2020 17:00	4.0	0.0	0.0
9	10	01/22/2020	Hebei	Mainland China	1/22/2020 17:00	1.0	0.0	0.0
10	11	01/22/2020	Heilongjiang	Mainland China	1/22/2020 17:00	0.0	0.0	0.0
11	12	01/22/2020	Henan	Mainland China	1/22/2020 17:00	5.0	0.0	0.0
12	13	01/22/2020	Hong Kong	Hong Kong	1/22/2020 17:00	0.0	0.0	0.0
13	14	01/22/2020	Hubei	Mainland China	1/22/2020 17:00	444.0	17.0	28.0
14	15	01/22/2020	Hunan	Mainland China	1/22/2020 17:00	4.0	0.0	0.0
15	16	01/22/2020	Inner Mongolia	Mainland China	1/22/2020 17:00	0.0	0.0	0.0
16	17	01/22/2020	Jiangsu	Mainland China	1/22/2020 17:00	1.0	0.0	0.0
17	18	01/22/2020	Jiangxi	Mainland China	1/22/2020 17:00	2.0	0.0	0.0

	SNo	ObservationDate	Province/State	Country/Region	Last Update	Confirmed	Deaths	Recovered
18	19	01/22/2020	Jilin	Mainland China	1/22/2020 17:00	0.0	0.0	0.0
19	20	01/22/2020	Liaoning	Mainland China	1/22/2020 17:00	2.0	0.0	0.0
20	21	01/22/2020	Macau	Macau	1/22/2020 17:00	1.0	0.0	0.0
21	22	01/22/2020	Ningxia	Mainland China	1/22/2020 17:00	1.0	0.0	0.0
22	23	01/22/2020	Qinghai	Mainland China	1/22/2020 17:00	0.0	0.0	0.0
23	24	01/22/2020	Shaanxi	Mainland China	1/22/2020 17:00	0.0	0.0	0.0
24	25	01/22/2020	Shandong	Mainland China	1/22/2020 17:00	2.0	0.0	0.0
25	26	01/22/2020	Shanghai	Mainland China	1/22/2020 17:00	9.0	0.0	0.0
26	27	01/22/2020	Shanxi	Mainland China	1/22/2020 17:00	1.0	0.0	0.0
27	28	01/22/2020	Sichuan	Mainland China	1/22/2020 17:00	5.0	0.0	0.0
28	29	01/22/2020	Taiwan	Taiwan	1/22/2020 17:00	1.0	0.0	0.0
29	30	01/22/2020	Tianjin	Mainland China	1/22/2020 17:00	4.0	0.0	0.0
30	31	01/22/2020	Tibet	Mainland China	1/22/2020 17:00	0.0	0.0	0.0
31	32	01/22/2020	Washington	US	1/22/2020 17:00	1.0	0.0	0.0
32	33	01/22/2020	Xinjiang	Mainland China	1/22/2020 17:00	0.0	0.0	0.0
33	34	01/22/2020	Yunnan	Mainland China	1/22/2020 17:00	1.0	0.0	0.0
34	35	01/22/2020	Zhejiang	Mainland China	1/22/2020 17:00	10.0	0.0	0.0
35	36	01/22/2020	NaN	Japan	1/22/2020 17:00	2.0	0.0	0.0
36	37	01/22/2020	NaN	Thailand	1/22/2020 17:00	4.0	0.0	2.0
37	38	01/22/2020	NaN	South Korea	1/22/2020 17:00	1.0	0.0	0.0
38	39	01/22/2020	Unknown	China	1/22/2020 17:00	0.0	0.0	0.0
39	40	01/23/2020	Anhui	Mainland China	1/23/20 17:00	9.0	0.0	0.0
40	41	01/23/2020	Beijing	Mainland China	1/23/20 17:00	22.0	0.0	0.0
41	42	01/23/2020	Chongqing	Mainland China	1/23/20 17:00	9.0	0.0	0.0
42	43	01/23/2020	Fujian	Mainland China	1/23/20 17:00	5.0	0.0	0.0

	SNo	ObservationDate	Province/State	Country/Region	Last Update	Confirmed	Deaths	Recovered
43	44	01/23/2020	Gansu	Mainland China	1/23/20 17:00	2.0	0.0	0.0
44	45	01/23/2020	Guangdong	Mainland China	1/23/20 17:00	32.0	0.0	2.0
45	46	01/23/2020	Guangxi	Mainland China	1/23/20 17:00	5.0	0.0	0.0
46	47	01/23/2020	Guizhou	Mainland China	1/23/20 17:00	3.0	0.0	0.0
47	48	01/23/2020	Hainan	Mainland China	1/23/20 17:00	5.0	0.0	0.0
48	49	01/23/2020	Hubei	Mainland China	1/23/20 17:00	444.0	17.0	28.0
49	50	01/23/2020	Heilongjiang	Mainland China	1/23/20 17:00	2.0	0.0	0.0

In [10]: `df.tail(50)`

Out[10]:

	SNo	ObservationDate	Province/State	Country/Region	Last Update	Confirmed	Deaths	Recovered
285257	285258	05/02/2021	Valle d'Aosta	Italy	2021-05-03 04:20:39	11013.0	459.0	9856.0
285258	285259	05/02/2021	Valle del Cauca	Colombia	2021-05-03 04:20:39	238813.0	7376.0	226123.0
285259	285260	05/02/2021	Valparaiso	Chile	2021-05-03 04:20:39	85927.0	2309.0	80180.0
285260	285261	05/02/2021	Varmland	Sweden	2021-05-03 04:20:39	16697.0	198.0	0.0
285261	285262	05/02/2021	Vasterbotten	Sweden	2021-05-03 04:20:39	20903.0	178.0	0.0
285262	285263	05/02/2021	Vasternorrland	Sweden	2021-05-03 04:20:39	23802.0	458.0	0.0
285263	285264	05/02/2021	Vastmanland	Sweden	2021-05-03 04:20:39	24678.0	343.0	0.0
285264	285265	05/02/2021	Vastra Gotaland	Sweden	2021-05-03 04:20:39	171187.0	2294.0	0.0
285265	285266	05/02/2021	Vaupes	Colombia	2021-05-03 04:20:39	1278.0	13.0	1229.0
285266	285267	05/02/2021	Veneto	Italy	2021-05-03 04:20:39	413142.0	11365.0	380037.0
285267	285268	05/02/2021	Veracruz	Mexico	2021-05-03 04:20:39	59870.0	9582.0	0.0
285268	285269	05/02/2021	Vermont	US	2021-05-03 04:20:39	23126.0	247.0	0.0
285269	285270	05/02/2021	Vichada	Colombia	2021-05-03 04:20:39	1552.0	23.0	1521.0
285270	285271	05/02/2021	Victoria	Australia	2021-05-03 04:20:39	20523.0	820.0	19682.0

	SNo	ObservationDate	Province/State	Country/Region	Last Update	Confirmed	Deaths	Recovered
285271	285272	05/02/2021	Vinnytsia Oblast	Ukraine	2021-05-03 04:20:39	68117.0	1516.0	52069.0
285272	285273	05/02/2021	Virgin Islands	US	2021-05-03 04:20:39	3138.0	27.0	0.0
285273	285274	05/02/2021	Virginia	US	2021-05-03 04:20:39	661314.0	10791.0	0.0
285274	285275	05/02/2021	Vladimir Oblast	Russia	2021-05-03 04:20:39	32055.0	1116.0	29599.0
285275	285276	05/02/2021	Volgograd Oblast	Russia	2021-05-03 04:20:39	57100.0	1119.0	54556.0
285276	285277	05/02/2021	Vologda Oblast	Russia	2021-05-03 04:20:39	44917.0	1010.0	41988.0
285277	285278	05/02/2021	Volyn Oblast	Ukraine	2021-05-03 04:20:39	57526.0	1003.0	49171.0
285278	285279	05/02/2021	Voronezh Oblast	Russia	2021-05-03 04:20:39	80874.0	2662.0	76252.0
285279	285280	05/02/2021	Wakayama	Japan	2021-05-03 04:20:39	2221.0	27.0	1822.0
285280	285281	05/02/2021	Wales	UK	2021-05-03 04:20:39	211573.0	5550.0	0.0
285281	285282	05/02/2021	Wallis and Futuna	France	2021-05-03 04:20:39	444.0	7.0	44.0
285282	285283	05/02/2021	Walloon Brabant	Belgium	2021-05-03 04:20:39	38008.0	0.0	0.0
285283	285284	05/02/2021	Washington	US	2021-05-03 04:20:39	404709.0	5499.0	0.0
285284	285285	05/02/2021	West Bengal	India	2021-05-03 04:20:39	845878.0	11447.0	717772.0
285285	285286	05/02/2021	West Flanders	Belgium	2021-05-03 04:20:39	89704.0	0.0	0.0
285286	285287	05/02/2021	West Virginia	US	2021-05-03 04:20:39	153918.0	2686.0	0.0
285287	285288	05/02/2021	Western Australia	Australia	2021-05-03 04:20:39	1008.0	9.0	972.0
285288	285289	05/02/2021	Wisconsin	US	2021-05-03 04:20:39	661685.0	7567.0	0.0
285289	285290	05/02/2021	Wyoming	US	2021-05-03 04:20:39	58142.0	707.0	0.0
285290	285291	05/02/2021	Xinjiang	Mainland China	2021-05-03 04:20:39	980.0	3.0	977.0
285291	285292	05/02/2021	Yamagata	Japan	2021-05-03 04:20:39	1537.0	34.0	1308.0
285292	285293	05/02/2021	Yamaguchi	Japan	2021-05-03 04:20:39	1890.0	46.0	1545.0
285293	285294	05/02/2021	Yamalo-Nenets Autonomous Okrug	Russia	2021-05-03 04:20:39	38581.0	416.0	37367.0
285294	285295	05/02/2021	Yamanashi	Japan	2021-05-03 04:20:39	1242.0	19.0	1076.0
285295	285296	05/02/2021	Yaroslavl Oblast	Russia	2021-05-03 04:20:39	38796.0	536.0	37038.0

	SNo	ObservationDate	Province/State	Country/Region	Last Update	Confirmed	Deaths	Recovered
285296	285297	05/02/2021	Yucatan	Mexico	2021-05-03 04:20:39	37319.0	3636.0	0.0
285297	285298	05/02/2021	Yukon	Canada	2021-05-03 04:20:39	81.0	2.0	79.0
285298	285299	05/02/2021	Yunnan	Mainland China	2021-05-03 04:20:39	344.0	2.0	291.0
285299	285300	05/02/2021	Zabaykalsky Krai	Russia	2021-05-03 04:20:39	42364.0	630.0	40903.0
285300	285301	05/02/2021	Zacatecas	Mexico	2021-05-03 04:20:39	30059.0	2750.0	0.0
285301	285302	05/02/2021	Zakarpattia Oblast	Ukraine	2021-05-03 04:20:39	60151.0	1527.0	54174.0
285302	285303	05/02/2021	Zaporizhia Oblast	Ukraine	2021-05-03 04:20:39	96531.0	1919.0	78700.0
285303	285304	05/02/2021	Zeeland	Netherlands	2021-05-03 04:20:39	26045.0	233.0	0.0
285304	285305	05/02/2021	Zhejiang	Mainland China	2021-05-03 04:20:39	1344.0	1.0	1322.0
285305	285306	05/02/2021	Zhytomyr Oblast	Ukraine	2021-05-03 04:20:39	84641.0	1597.0	68529.0
285306	285307	05/02/2021	Zuid-Holland	Netherlands	2021-05-03 04:20:39	359327.0	4138.0	0.0

```
In [15]: df.drop(['SNo', 'Last Update'], axis=1, inplace=True)
```

```
In [21]: df.rename(columns={'ObservationDate': 'Date', 'Province/State': 'State', 'Country/Region': 'Country'}, inplace=True)
```

```
In [22]: df['Date'] = pd.to_datetime(df['Date'])
```

```
In [24]: df.describe()
```

```
Out[24]:
```

	Confirmed	Deaths	Recovered
count	2.853070e+05	285307.000000	2.853070e+05
mean	7.662923e+04	1867.334661	4.478423e+04
std	2.466176e+05	5905.565551	1.788030e+05
min	-3.028440e+05	-178.000000	-8.544050e+05
25%	9.700000e+02	12.000000	1.100000e+01
50%	9.181000e+03	172.000000	1.552000e+03

	Confirmed	Deaths	Recovered
75%	4.561800e+04	1181.000000	1.793500e+04
max	5.605532e+06	112182.000000	6.399531e+06

In [25]: `df.info()`

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 285307 entries, 0 to 285306
Data columns (total 6 columns):
#   Column      Non-Null Count  Dtype
---  -
0    Date        285307 non-null  datetime64[ns]
1    State       212318 non-null  object
2    Country     285307 non-null  object
3    Confirmed   285307 non-null  float64
4    Deaths     285307 non-null  float64
5    Recovered   285307 non-null  float64
dtypes: datetime64[ns](1), float64(3), object(2)
memory usage: 13.1+ MB
```

In [27]: `imputer=SimpleImputer(strategy='constant')`
`df2=pd.DataFrame(imputer.fit_transform(df),columns=df.columns)`

In [30]: `df3=df2.groupby(['Country','Date'])[['Country','Date','Confirmed','Deaths','Recovered']].sum().reset_index()`

In [31]: `df3.head(3)`

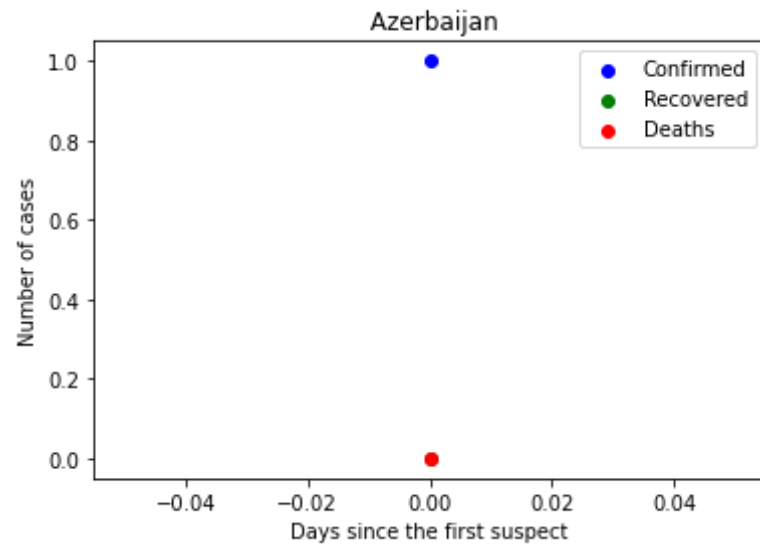
Out[31]:

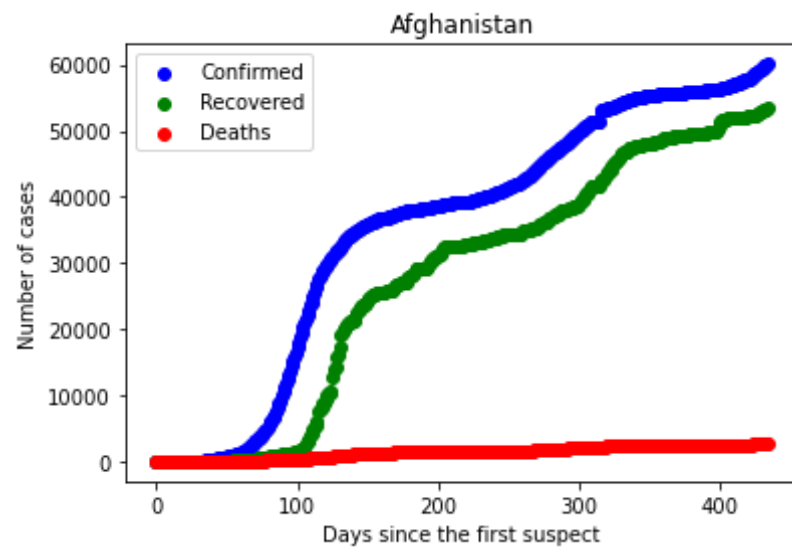
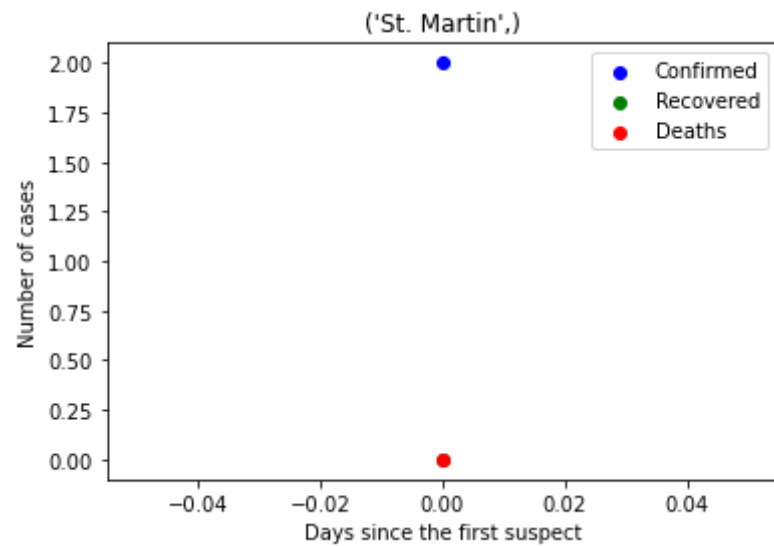
	Country	Date	Confirmed	Deaths	Recovered
0	Azerbaijan	2020-02-28	1.0	0.0	0.0
1	('St. Martin',)	2020-03-10	2.0	0.0	0.0
2	Afghanistan	2020-02-24	1.0	0.0	0.0

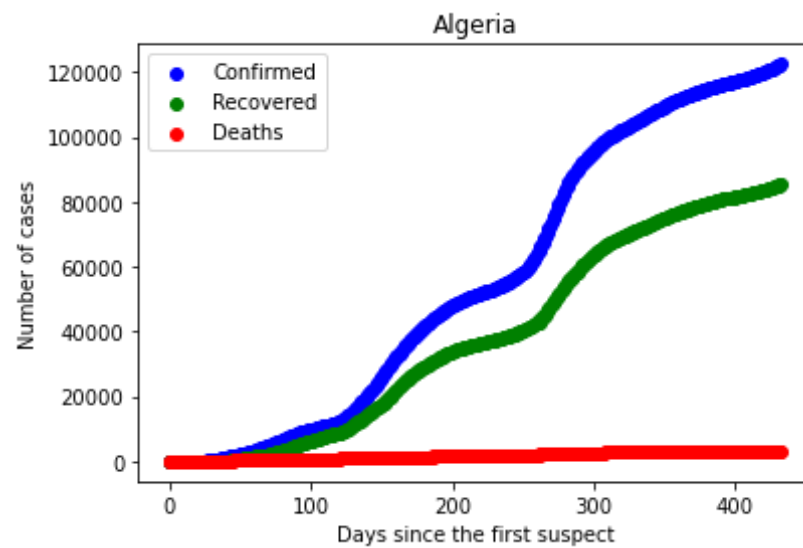
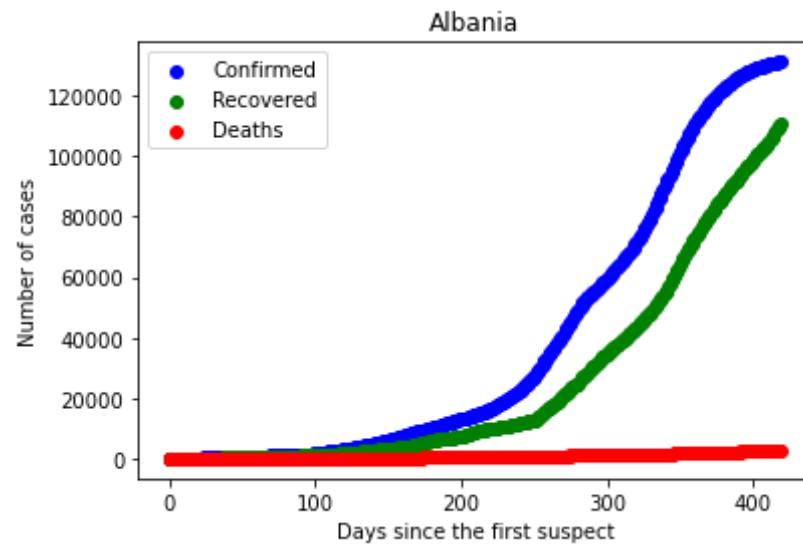
In [32]: `Countries=df3['Country'].unique()`
`len(Countries)`

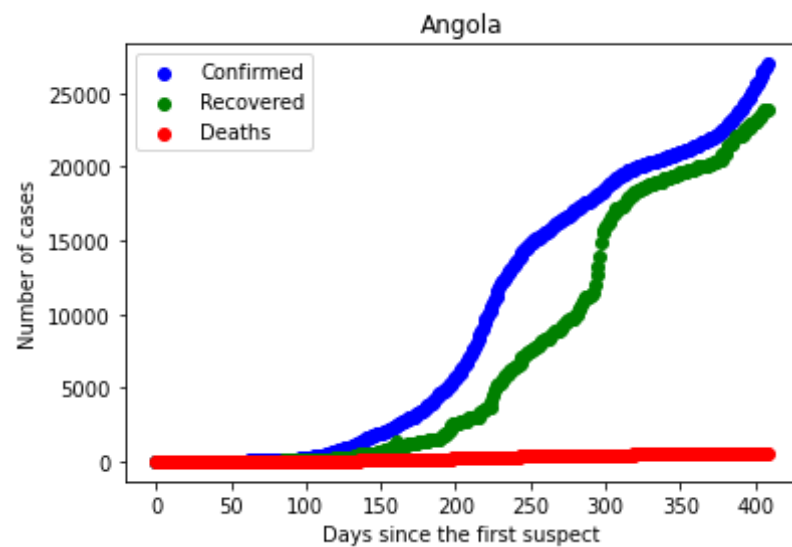
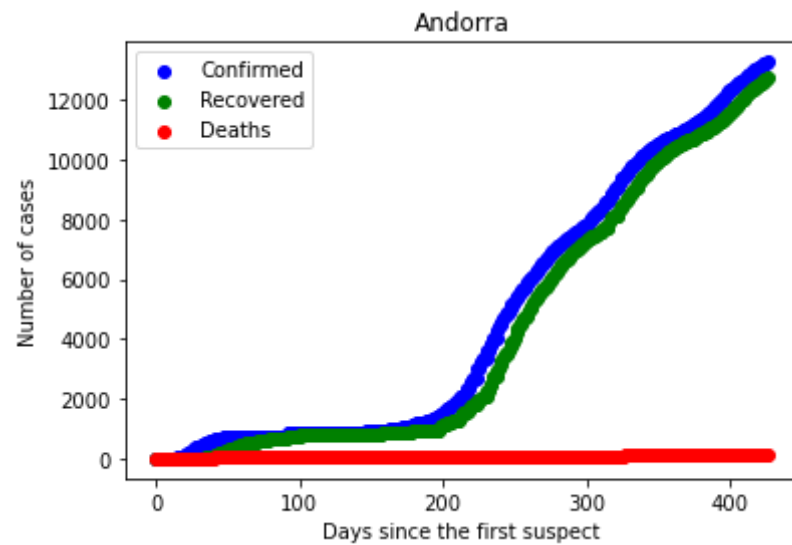
Out[32]: 228

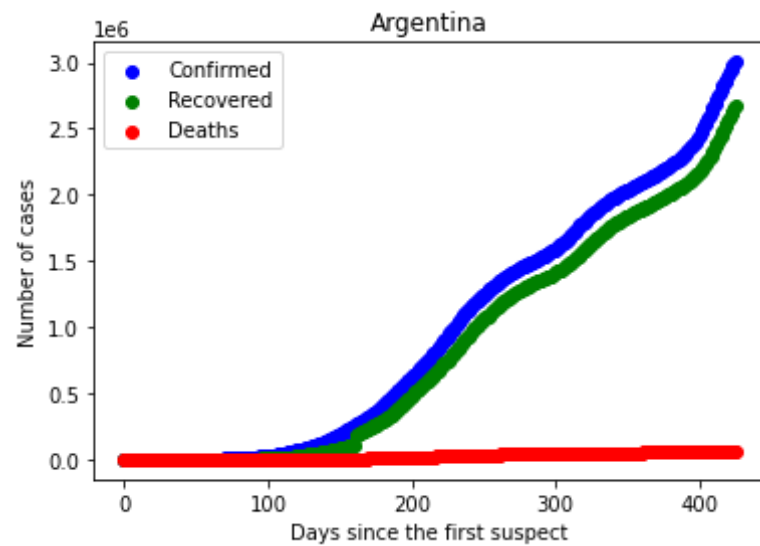
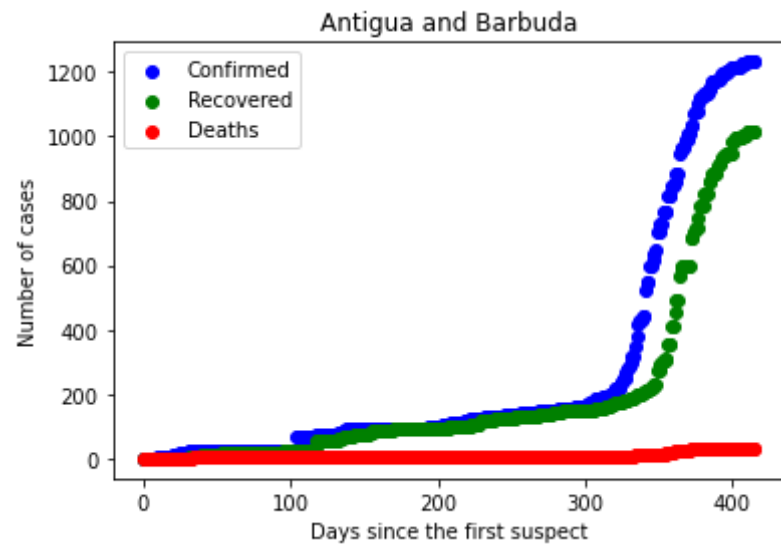
```
In [35]: for idx in range(0, len(Countries)):
        c = df3[df3['Country'] == Countries[idx]].reset_index()
        plt.scatter(np.arange(0, len(c)), c['Confirmed'], color='blue', label='Confirmed')
        plt.scatter(np.arange(0, len(c)), c['Recovered'], color='green', label='Recovered')
        plt.scatter(np.arange(0, len(c)), c['Deaths'], color='red', label='Deaths')
        plt.title(Countries[idx])
        plt.xlabel('Days since the first suspect')
        plt.ylabel('Number of cases')
        plt.legend()
        plt.show()
```

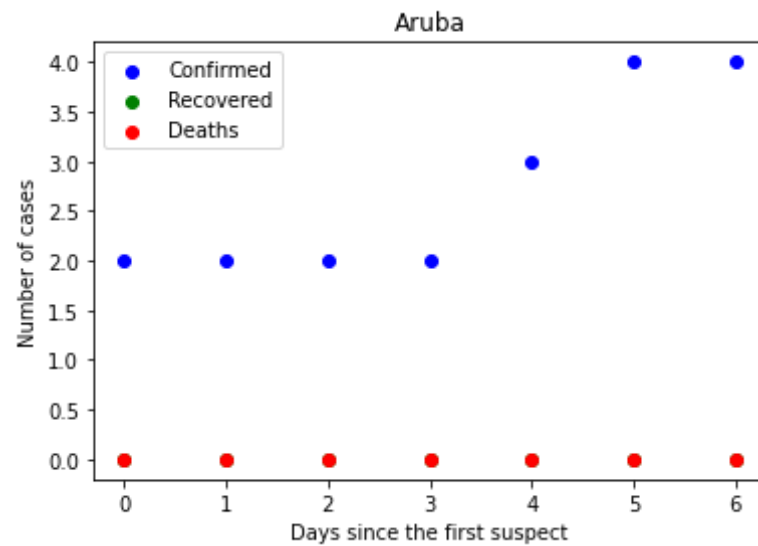
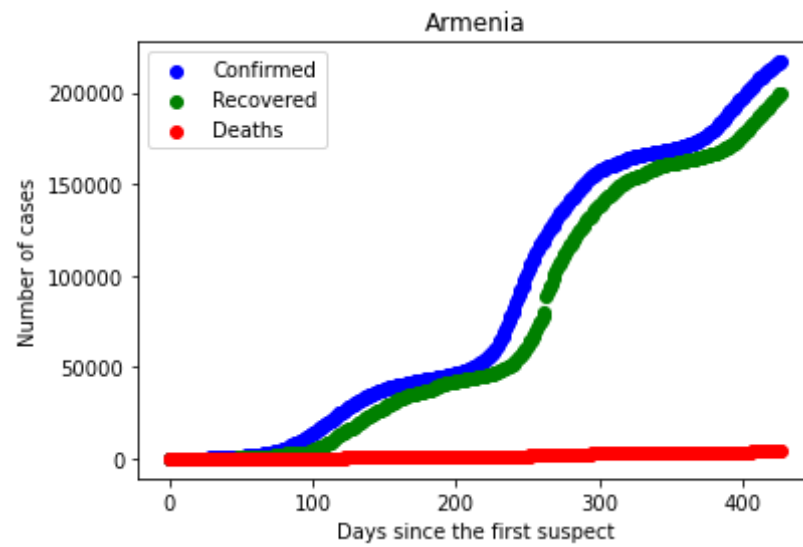


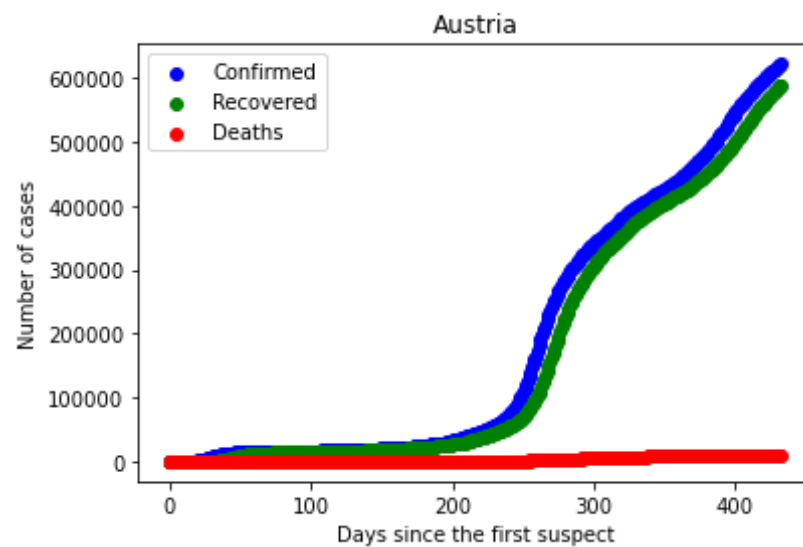
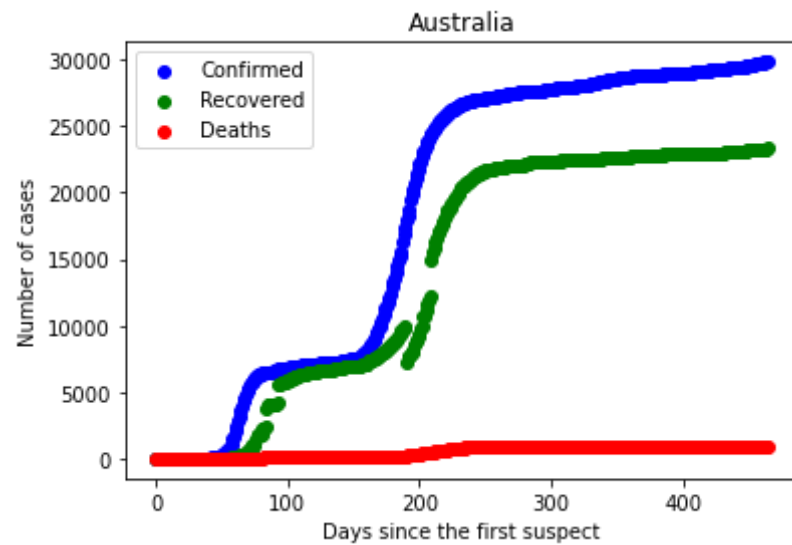


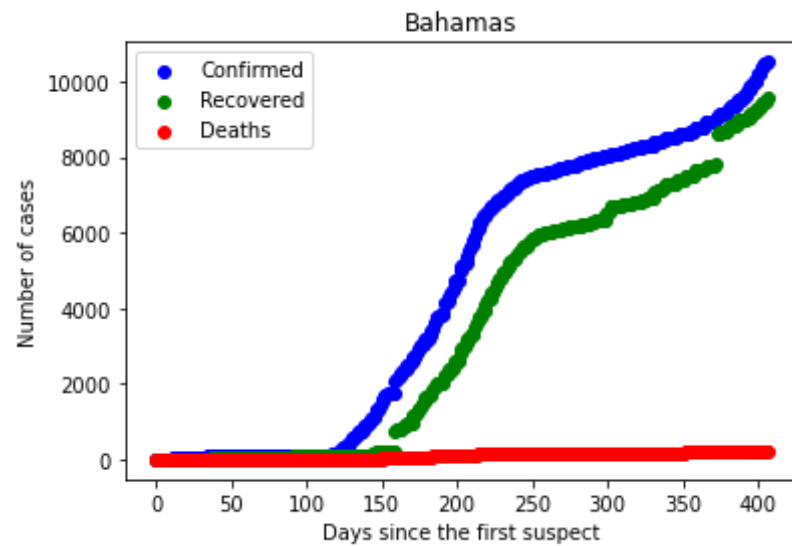
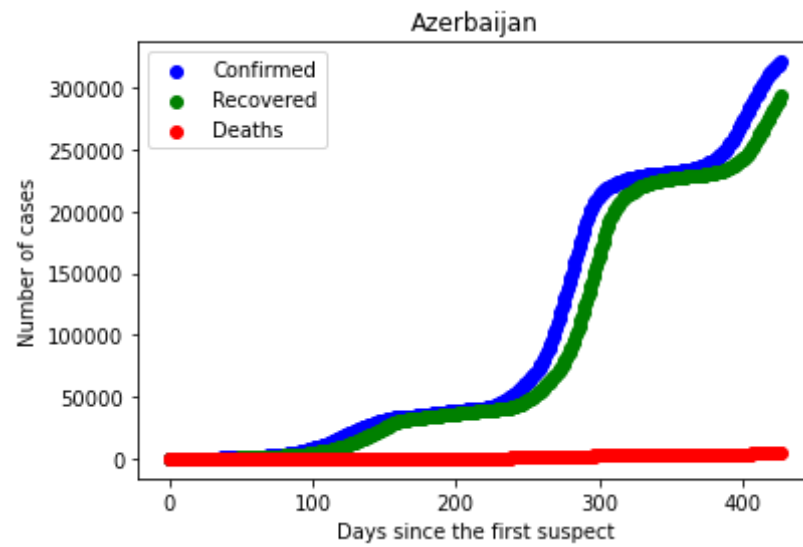


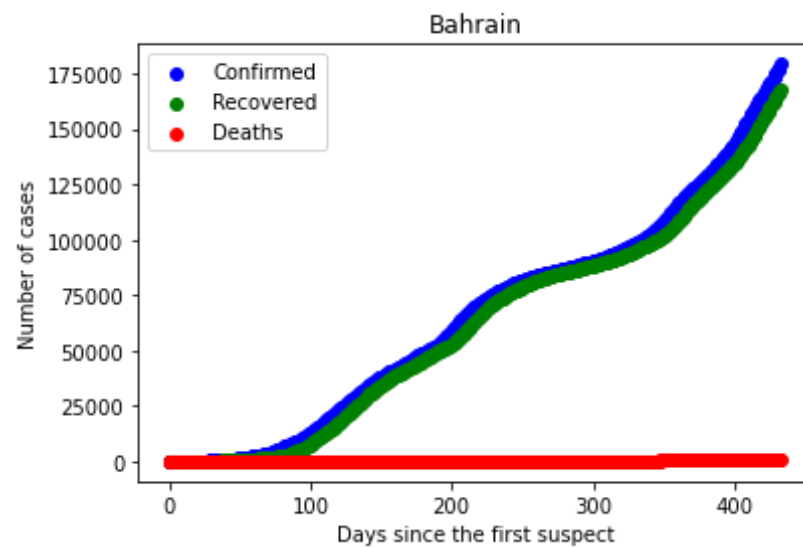
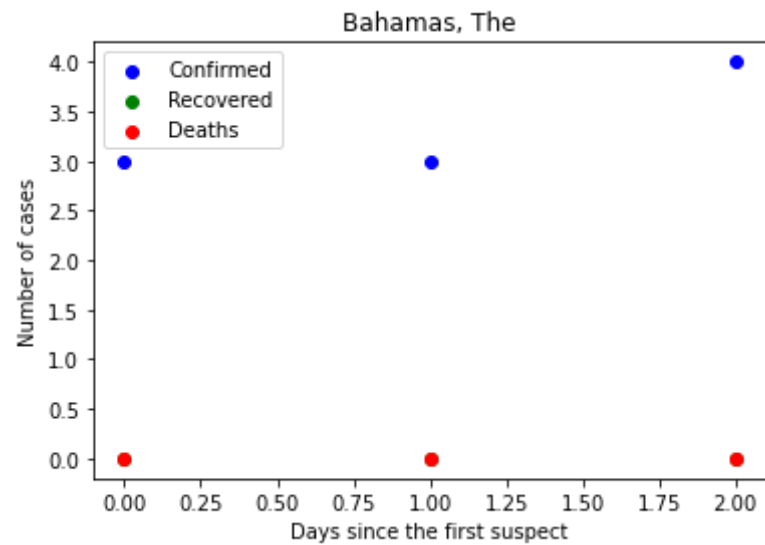


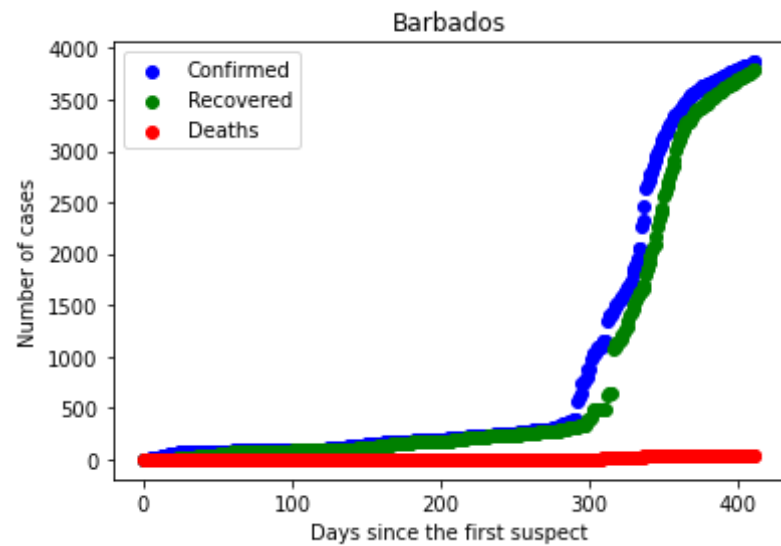
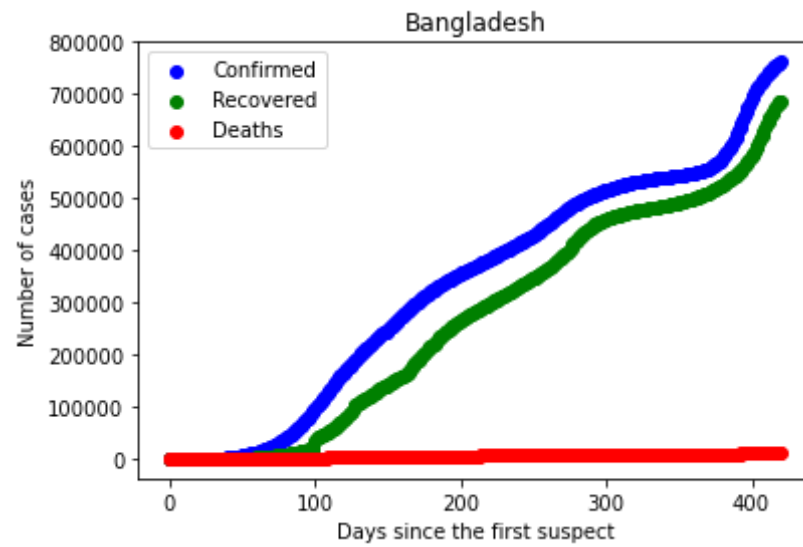


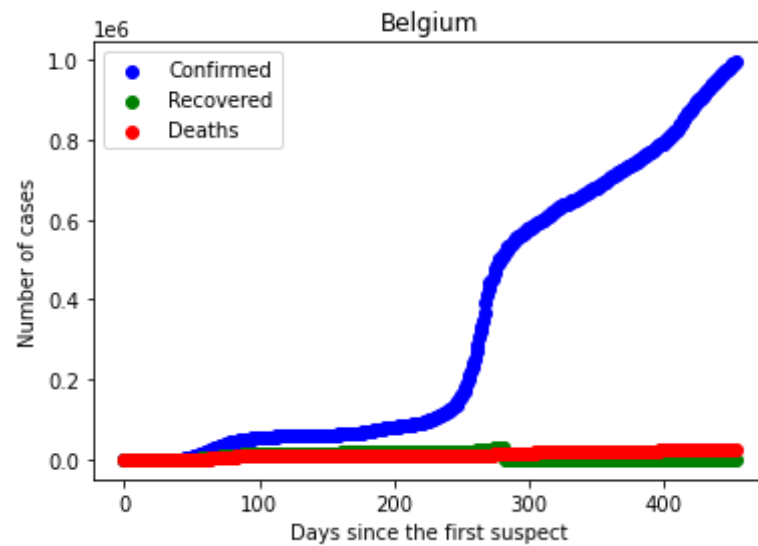
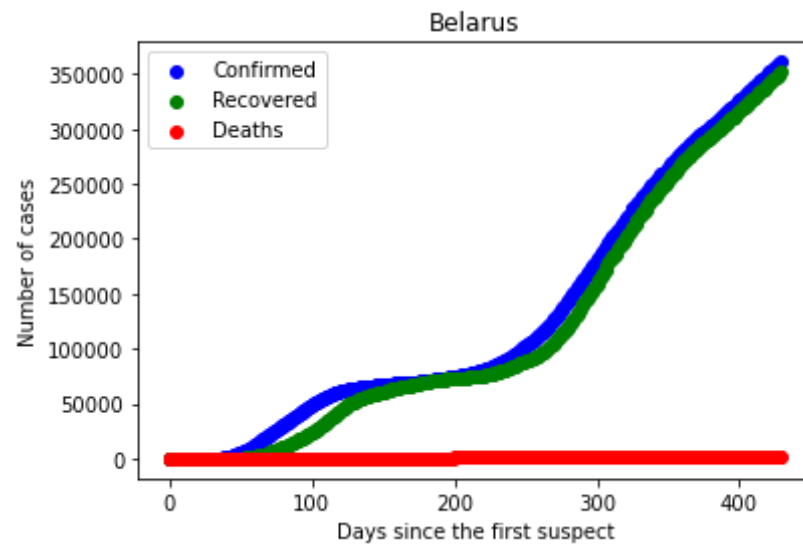


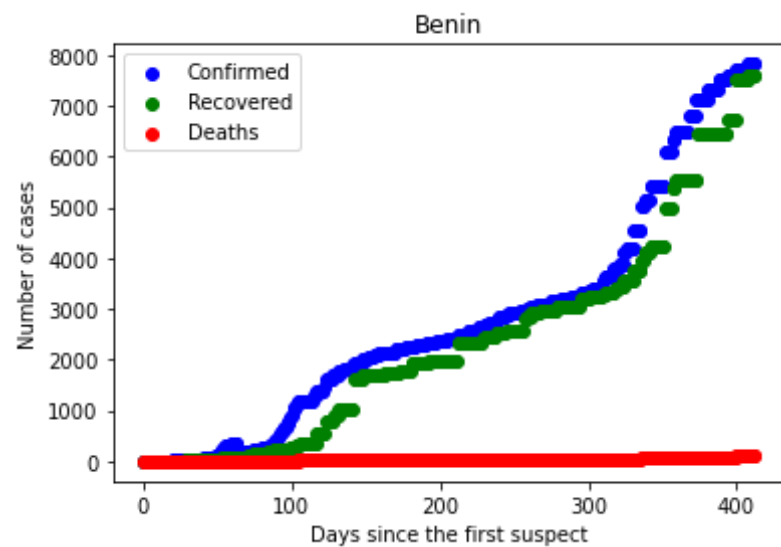
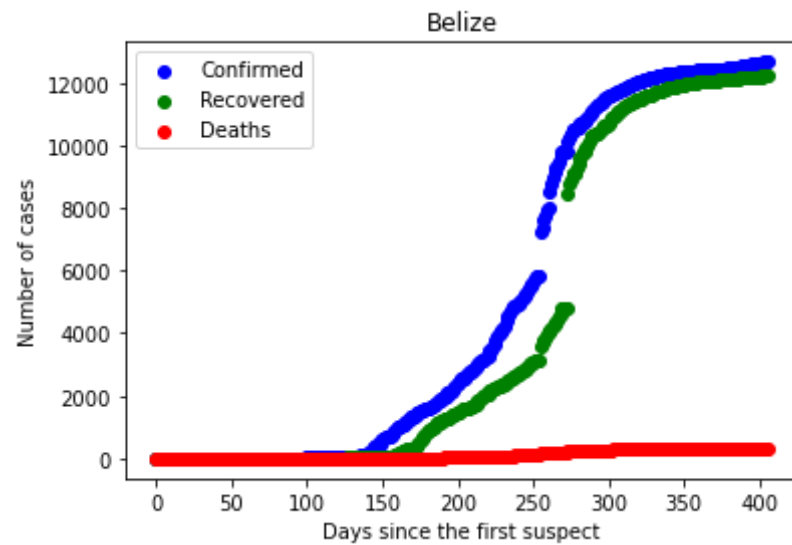


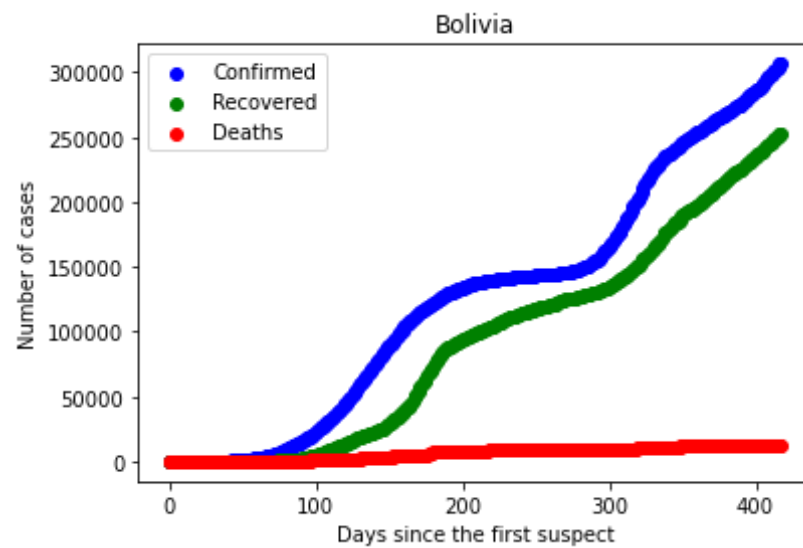
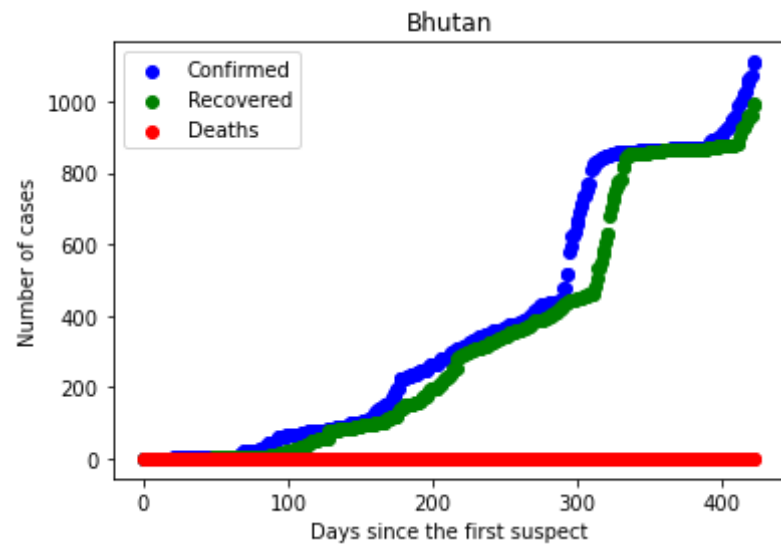


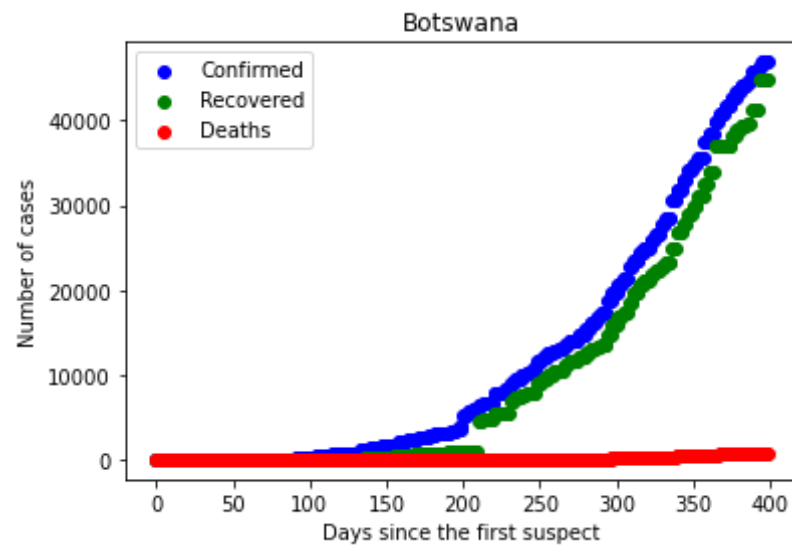
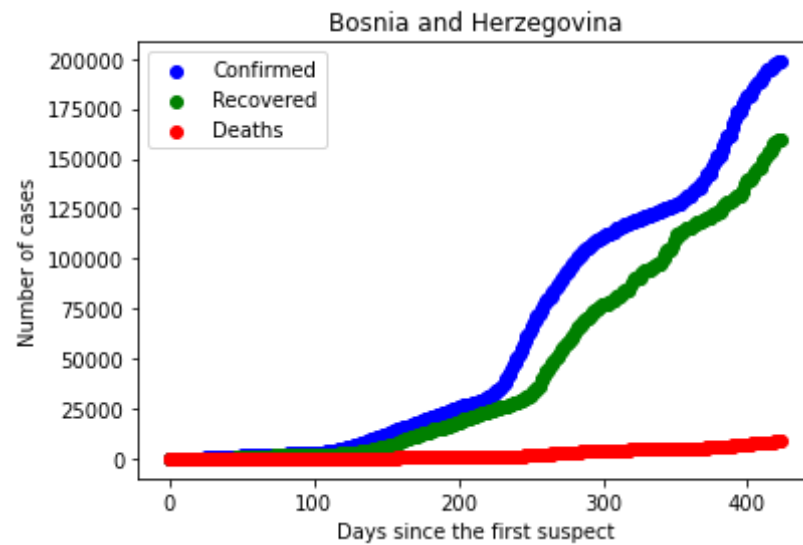


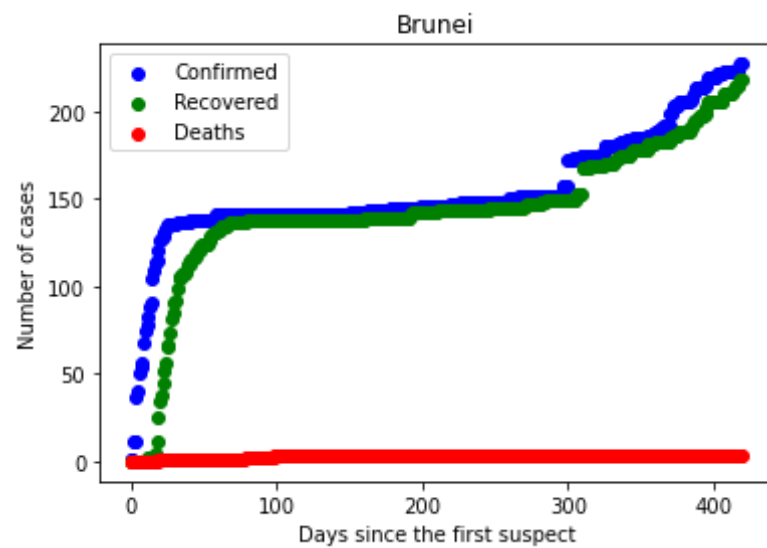
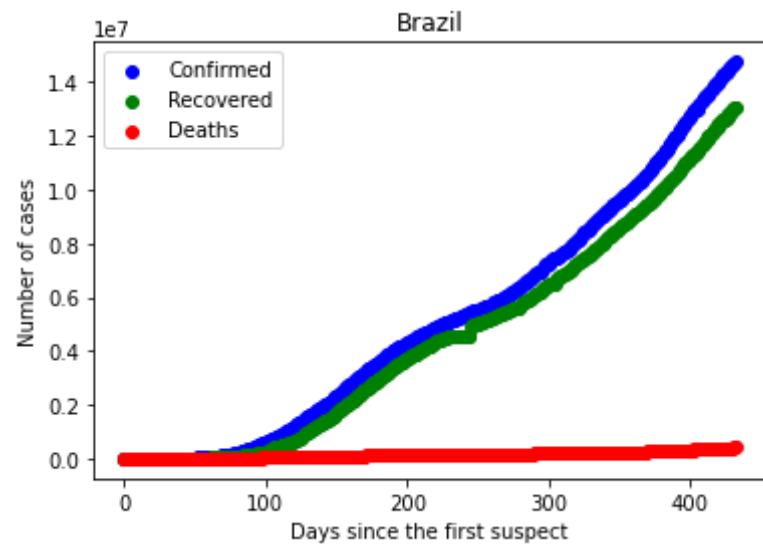


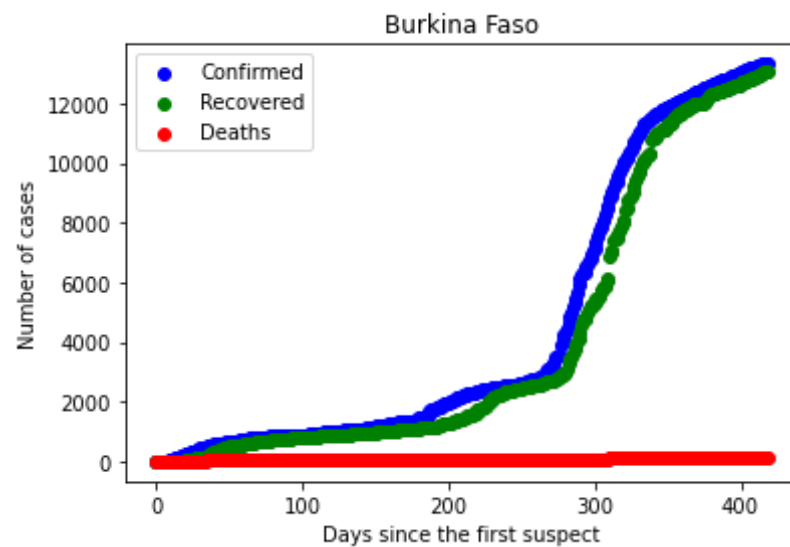
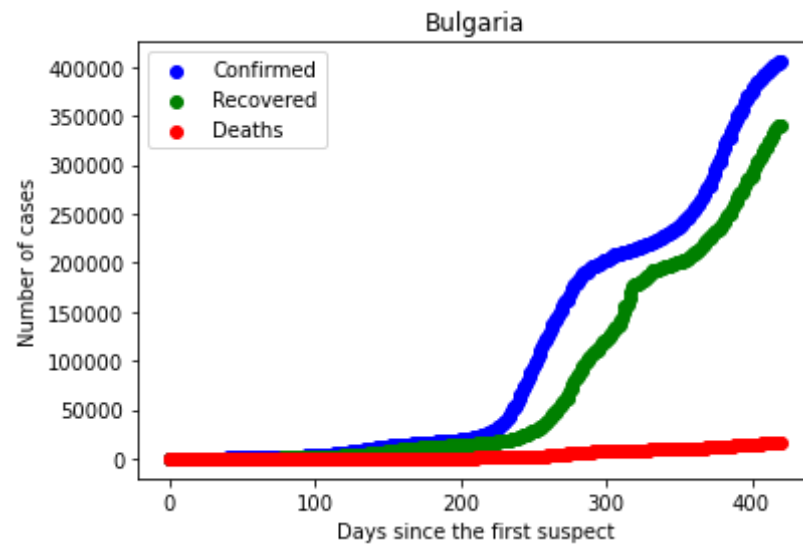


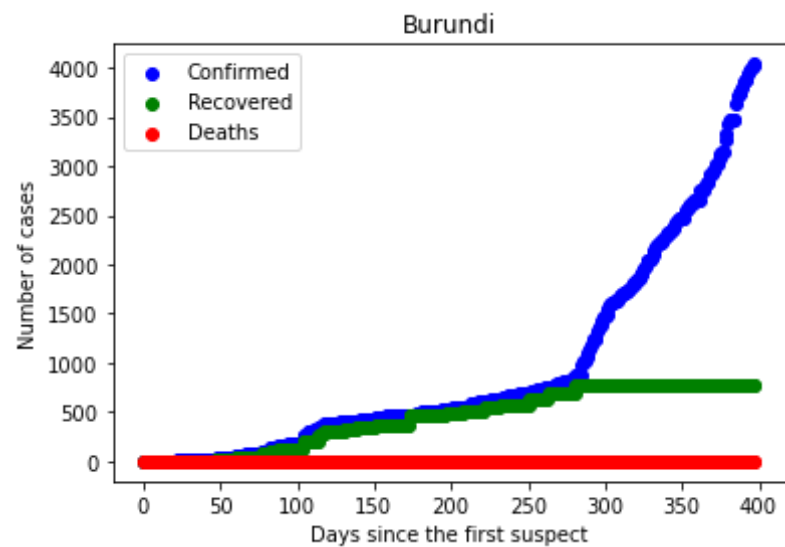
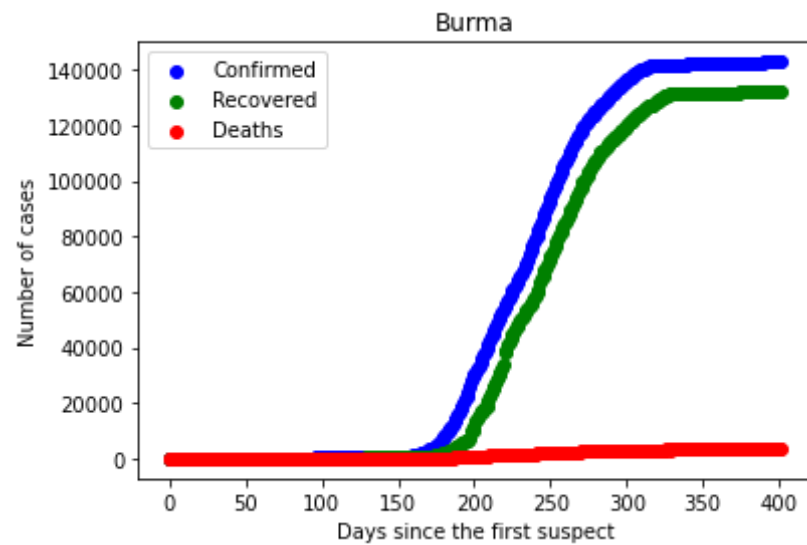


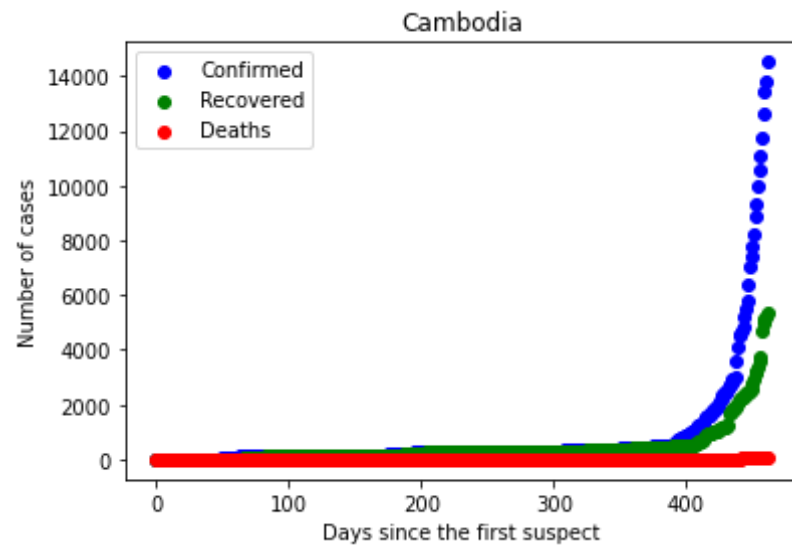
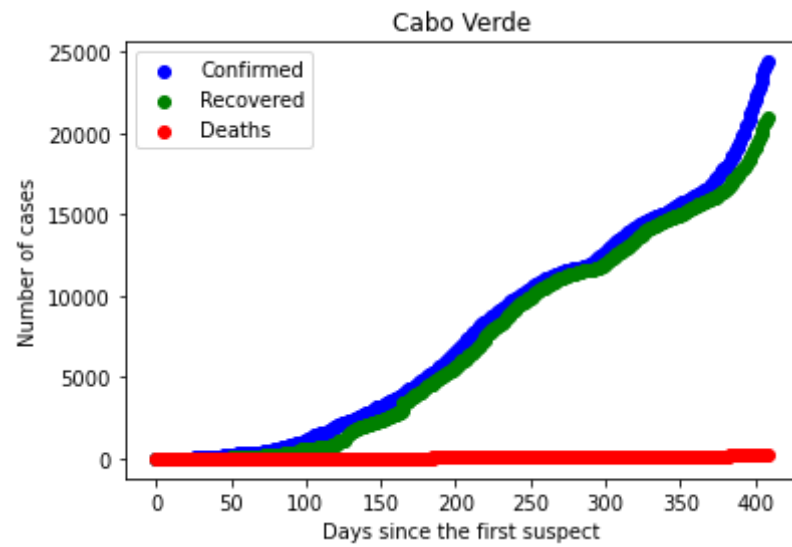


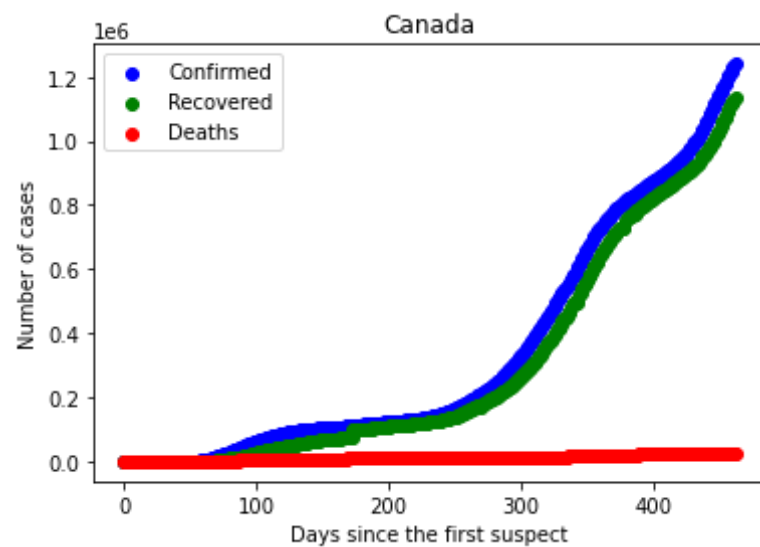
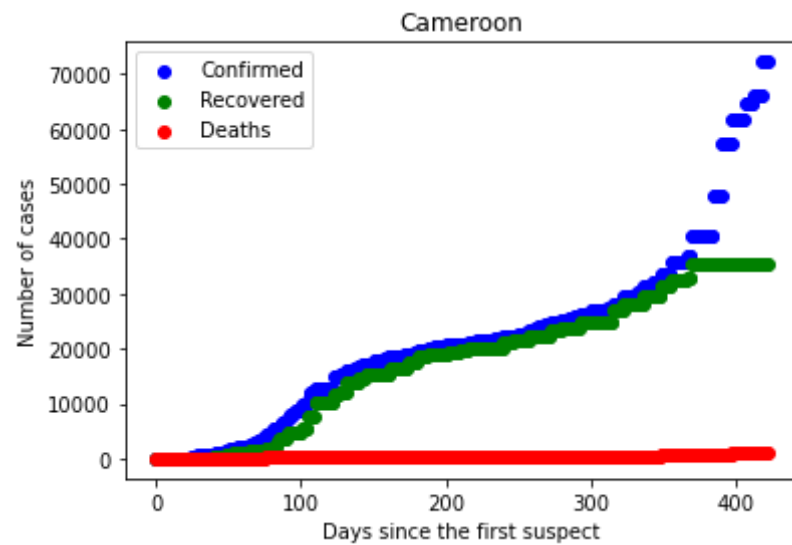


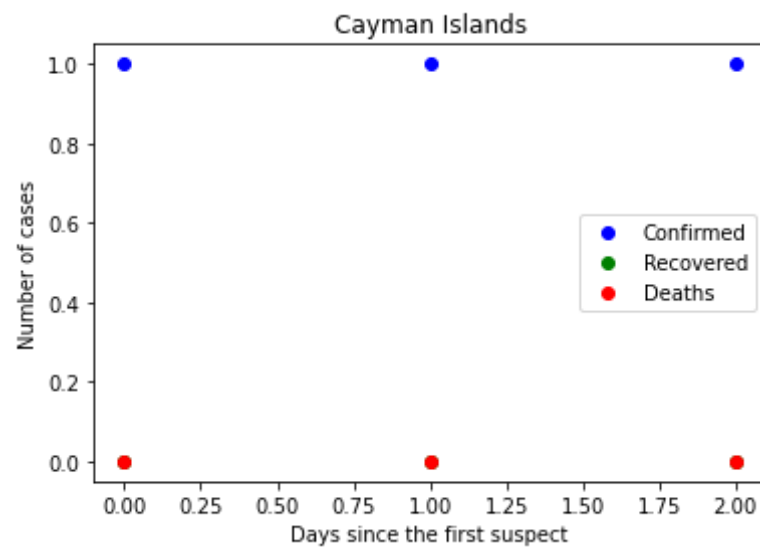
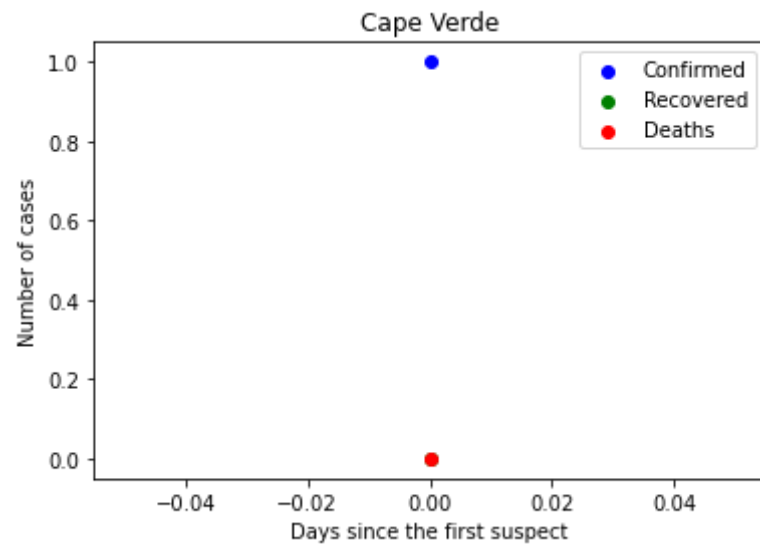


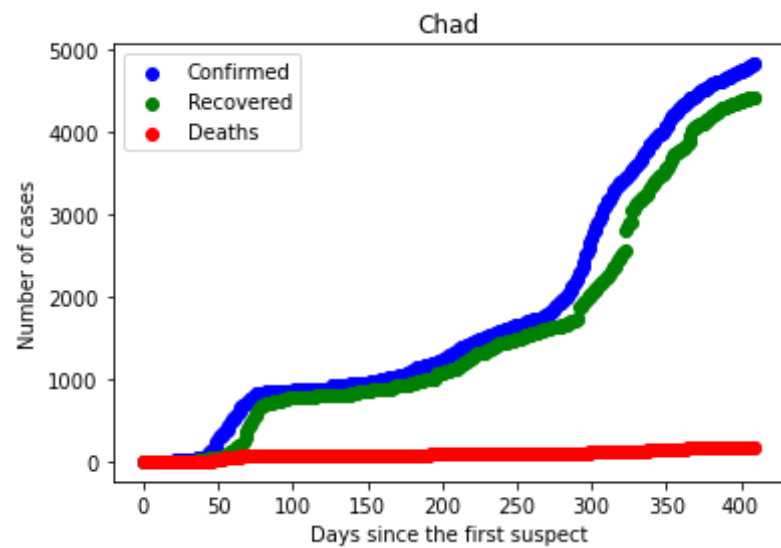
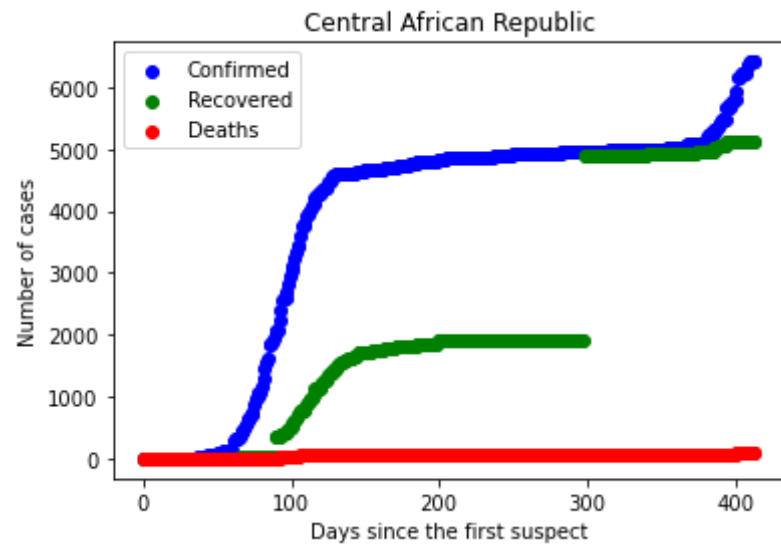


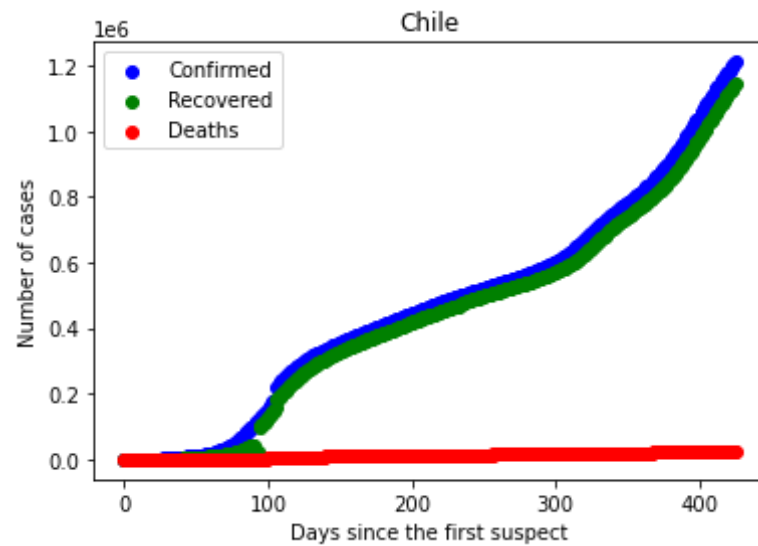
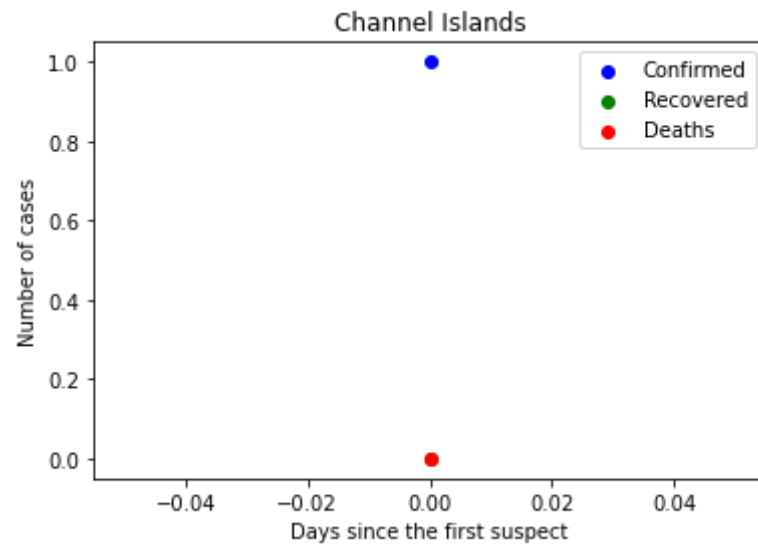


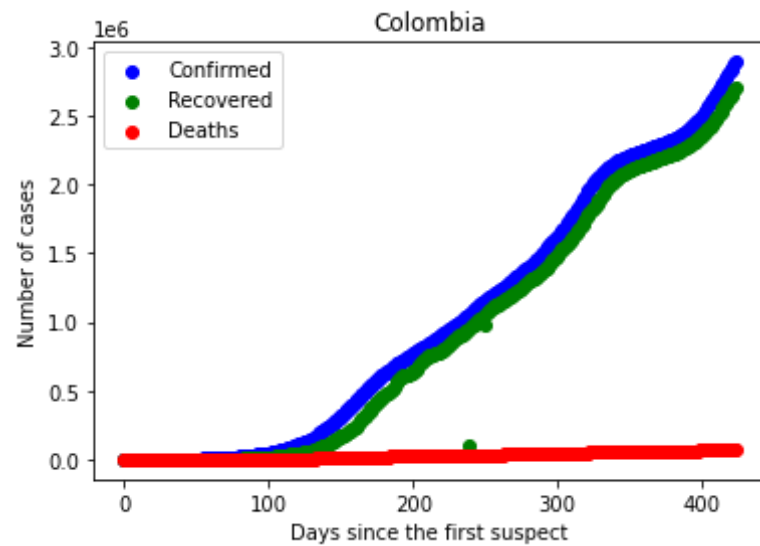
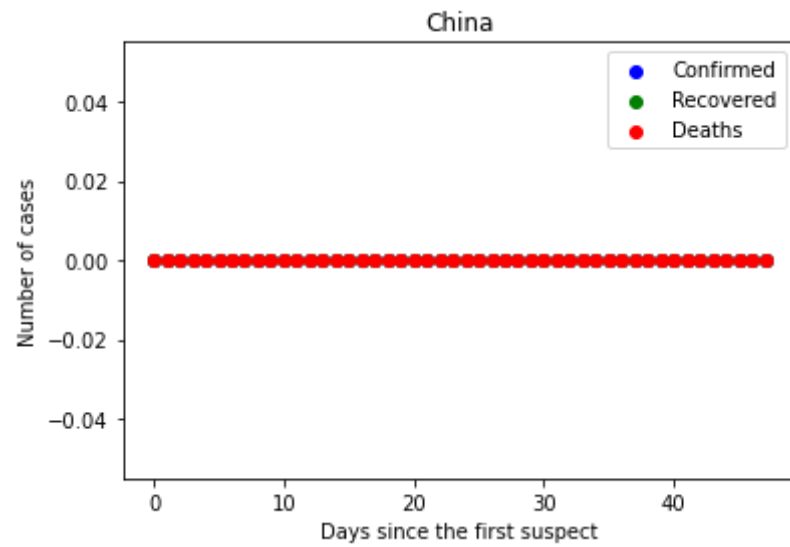


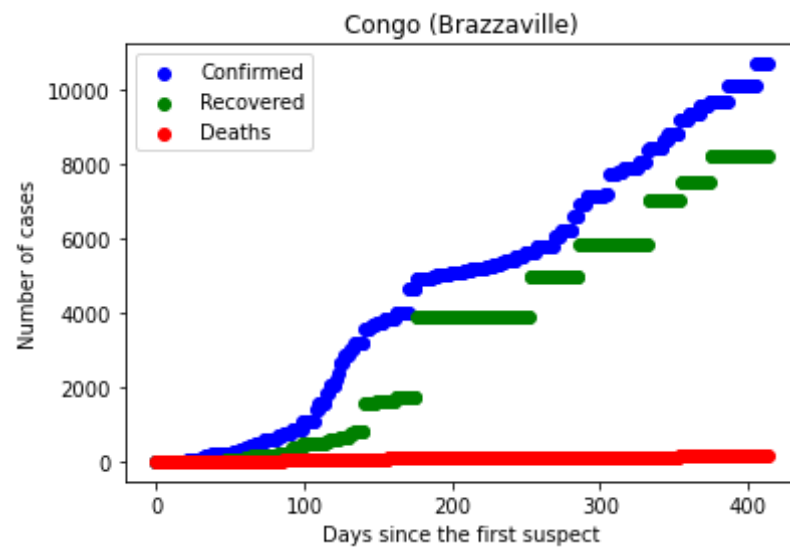
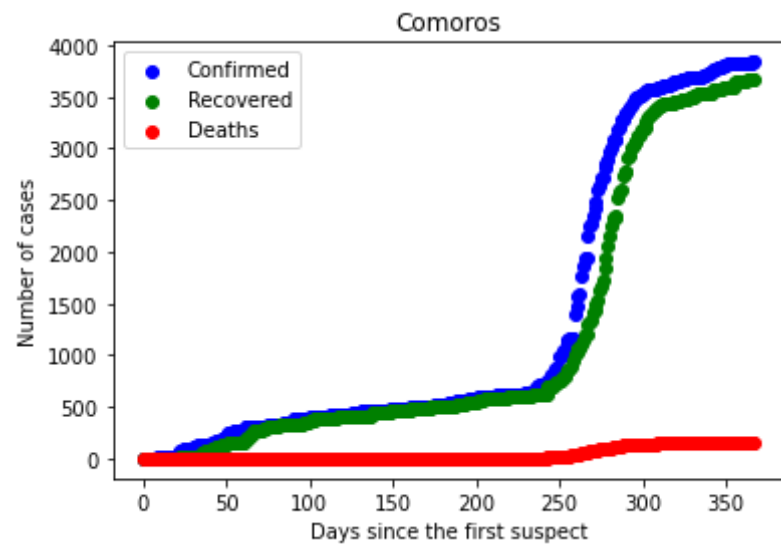


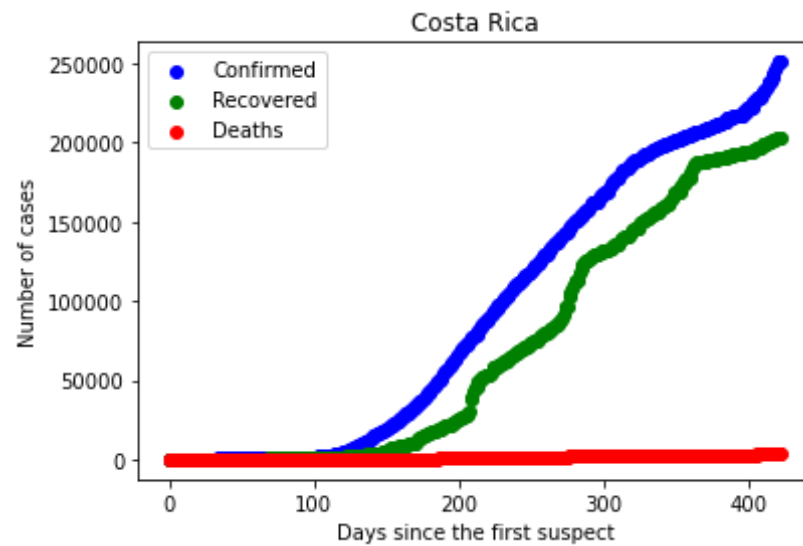
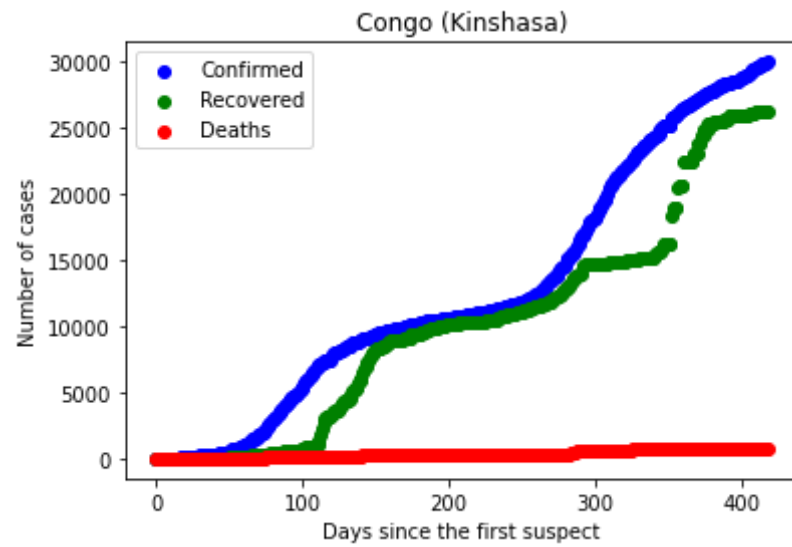


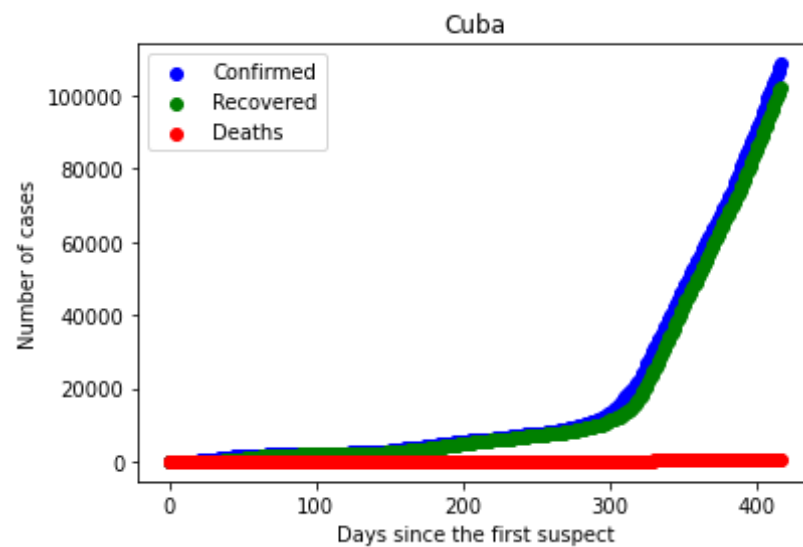
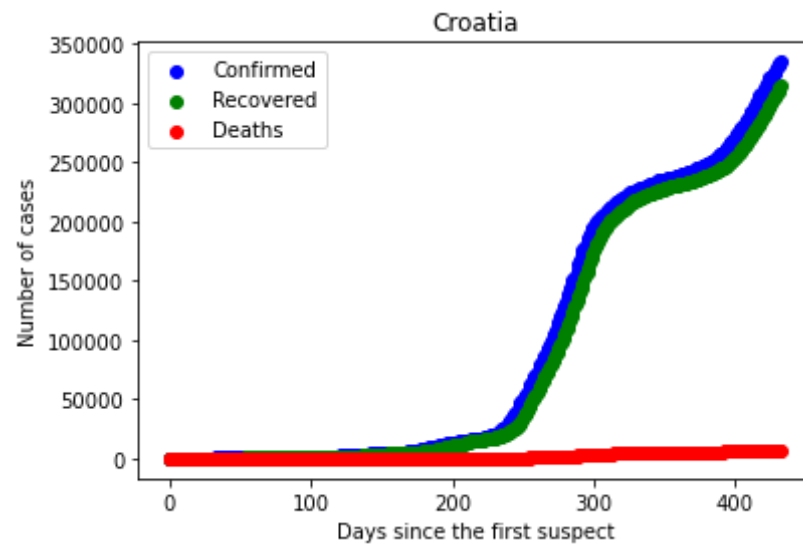


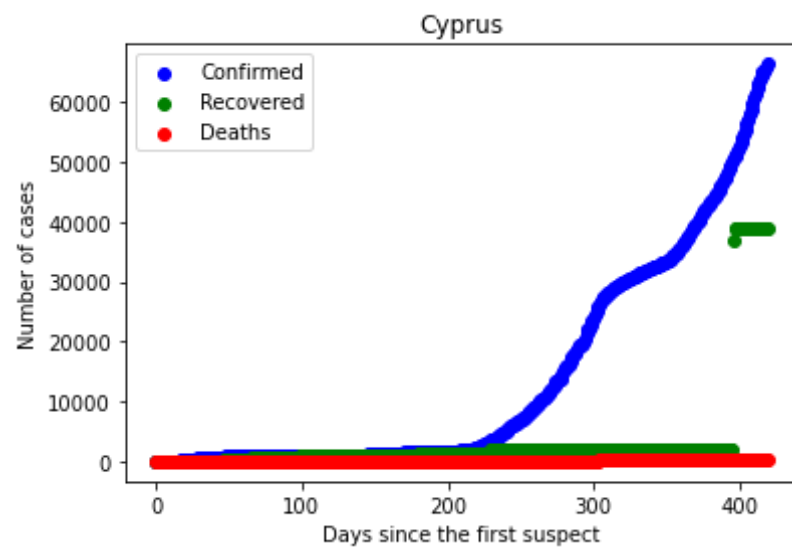
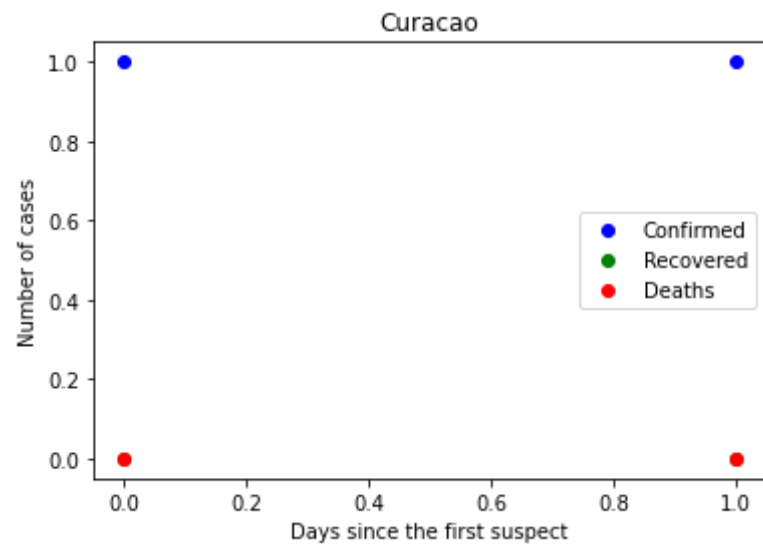


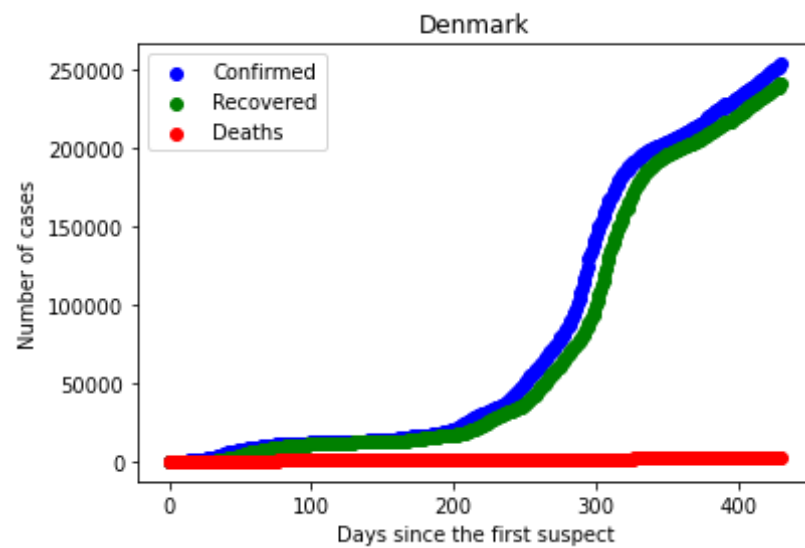
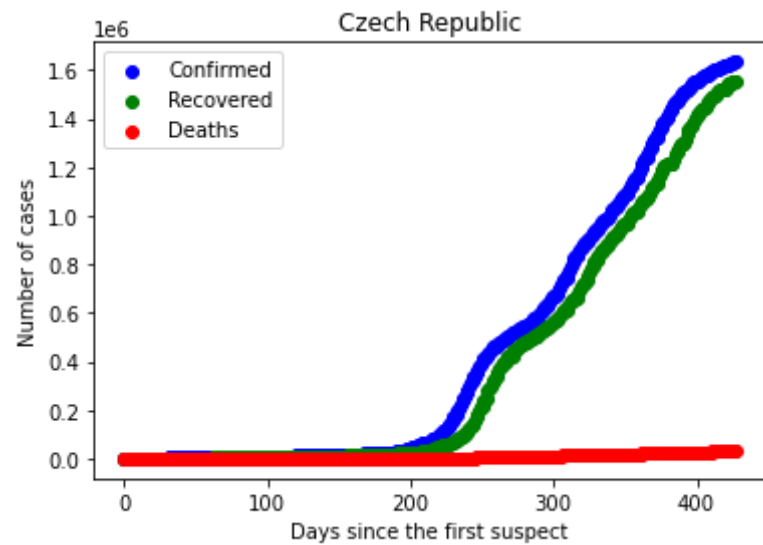


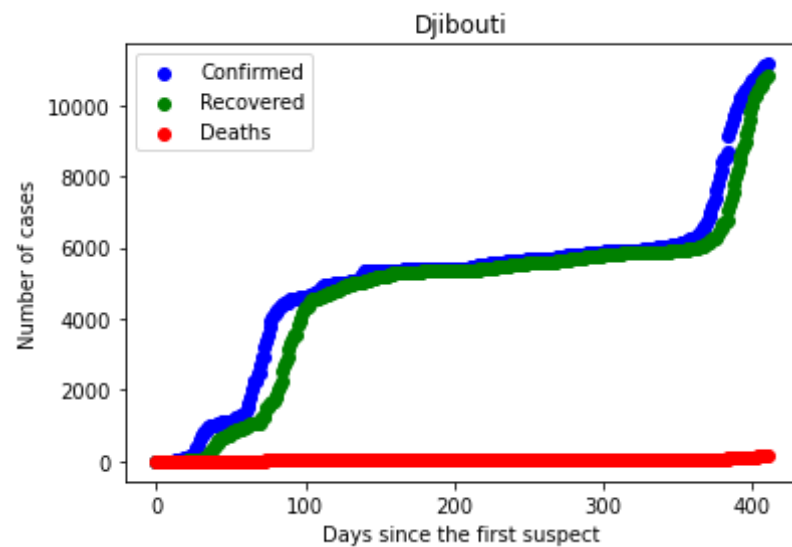
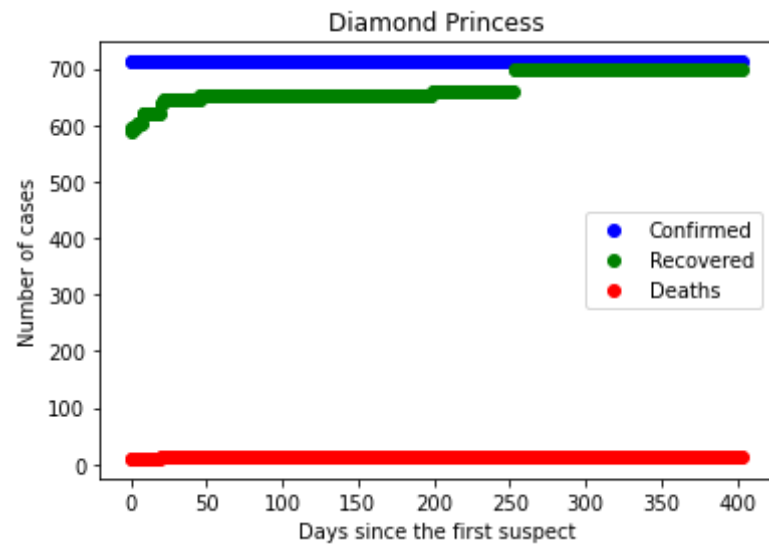


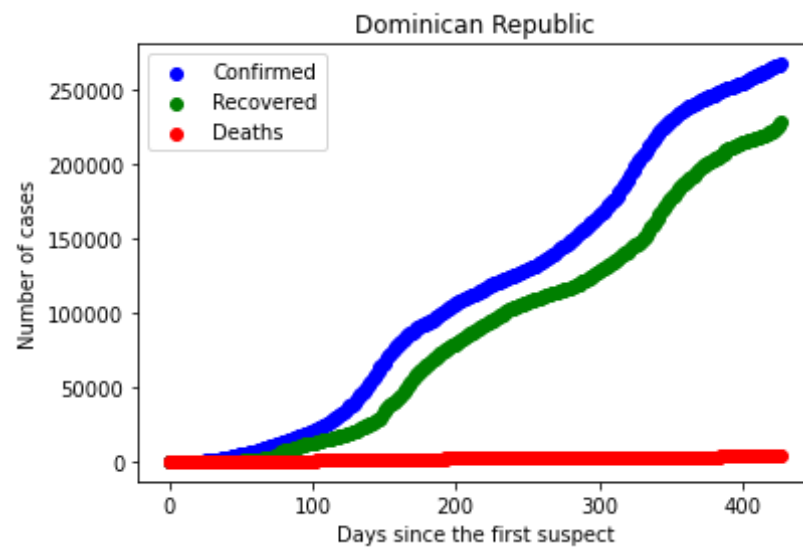
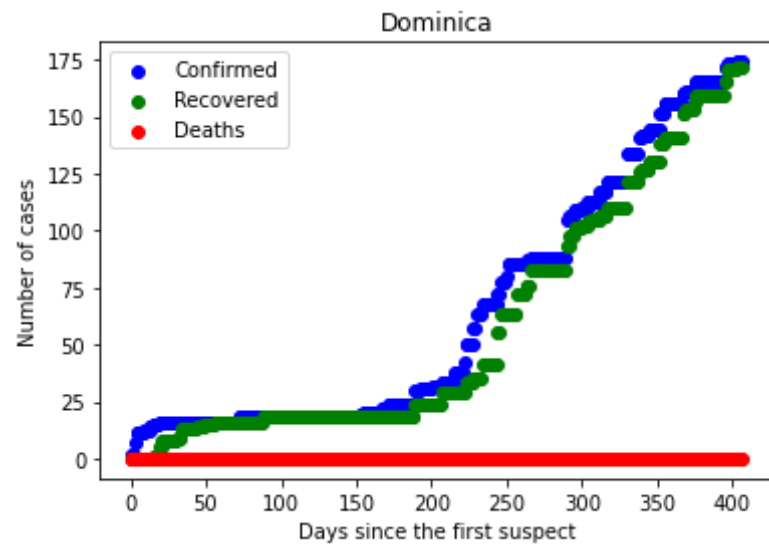


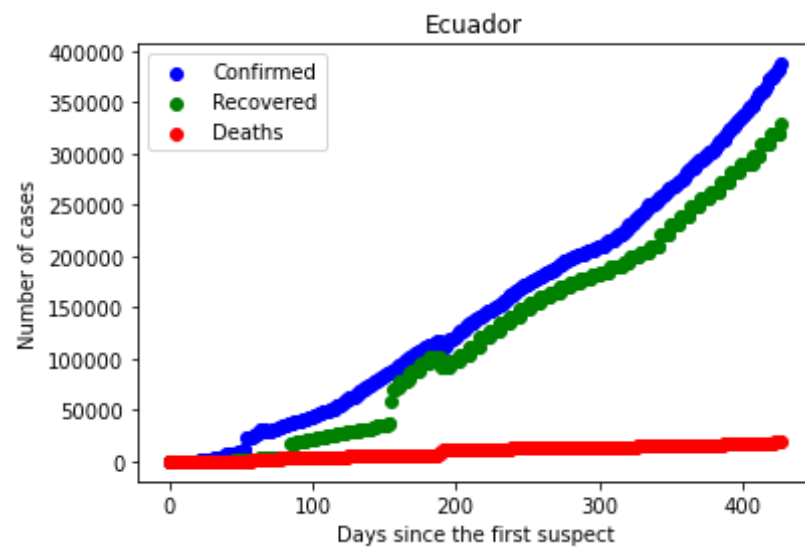
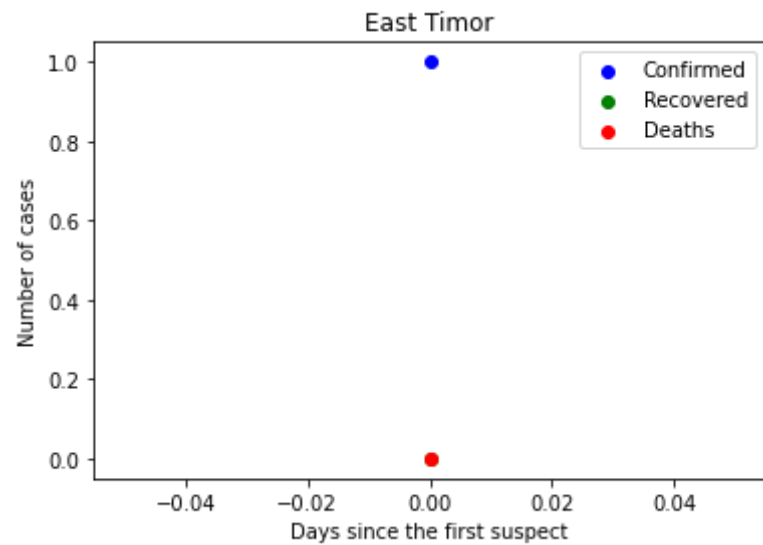


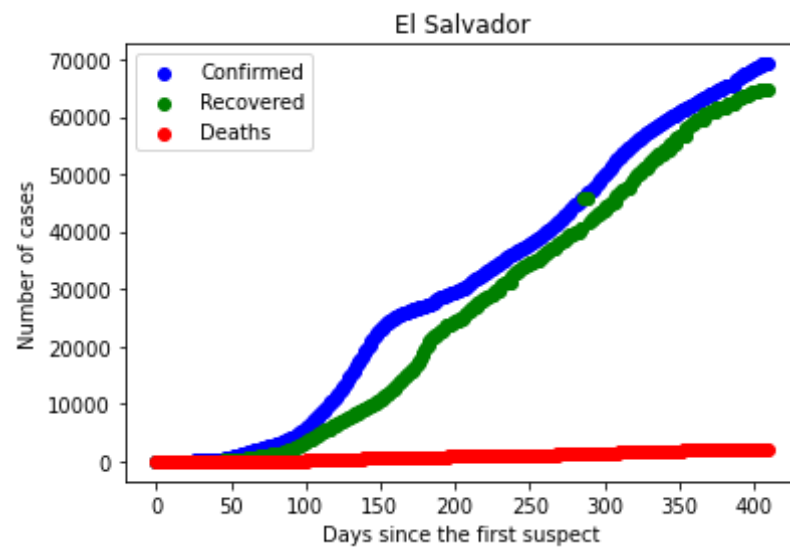
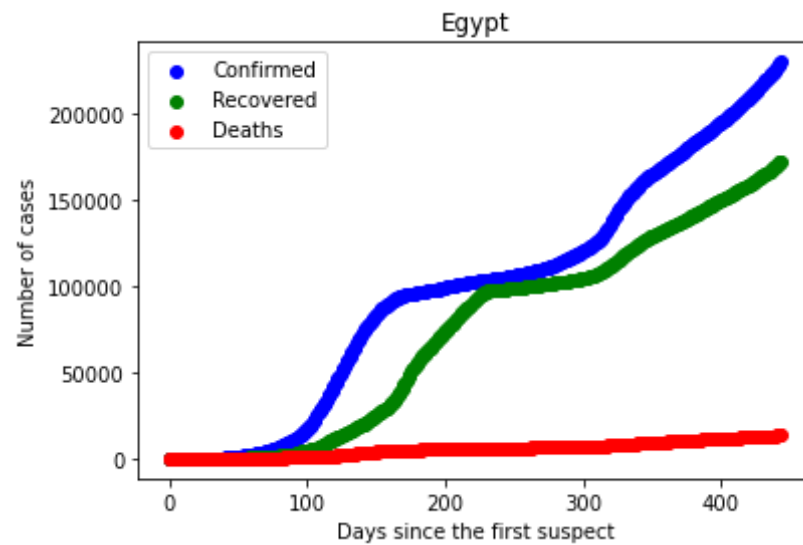


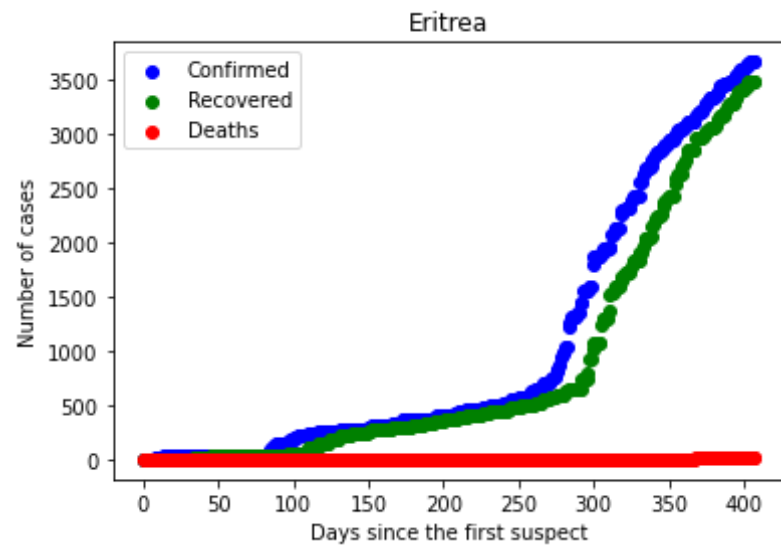
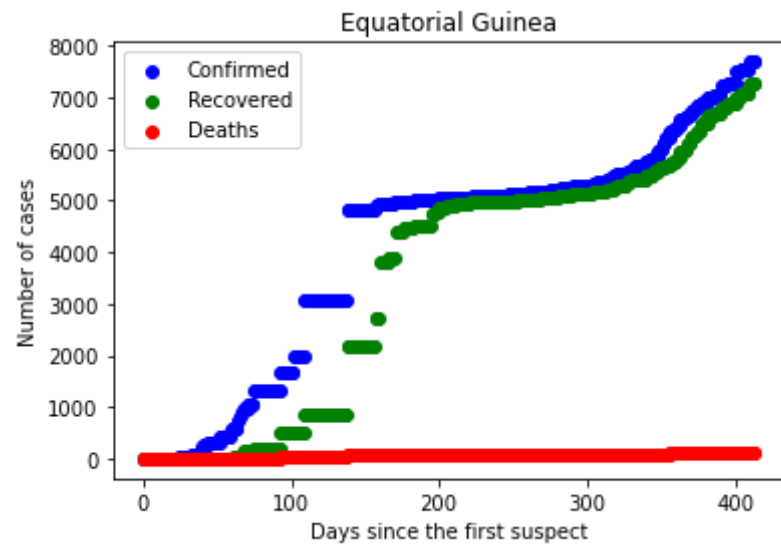


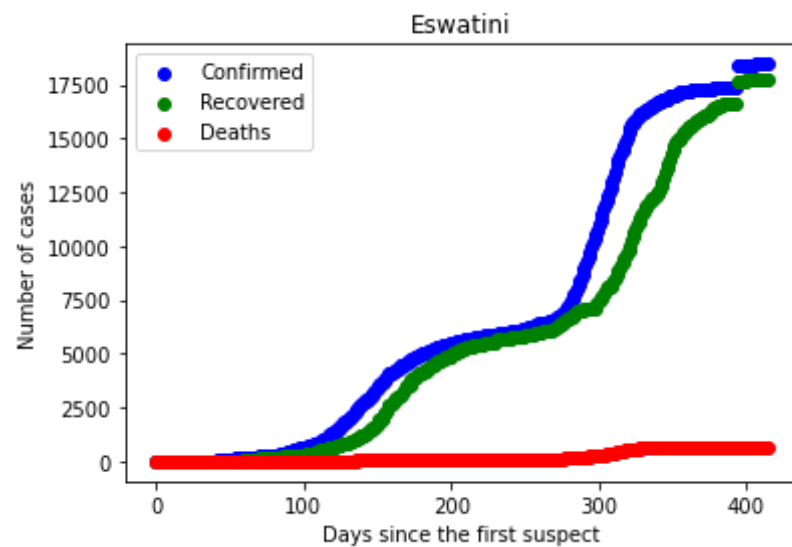
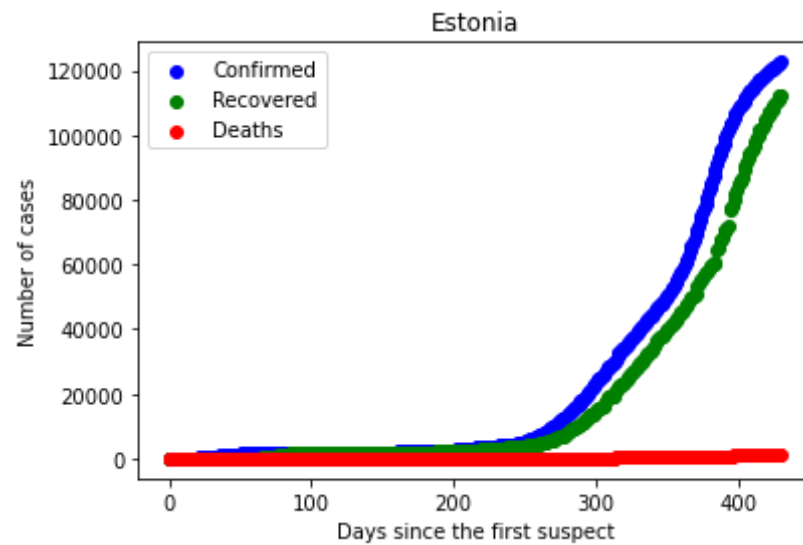


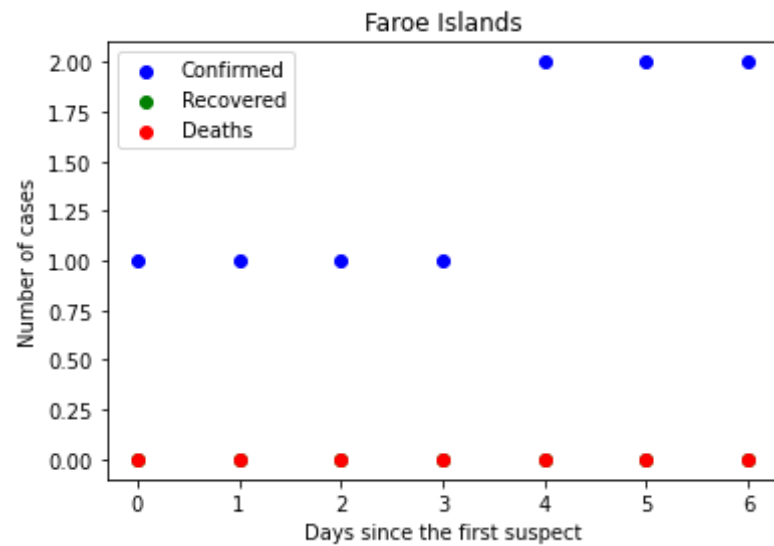
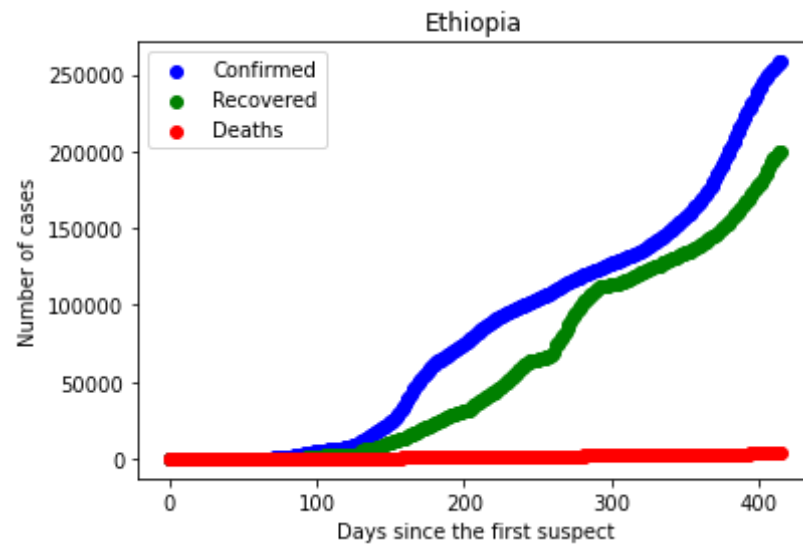


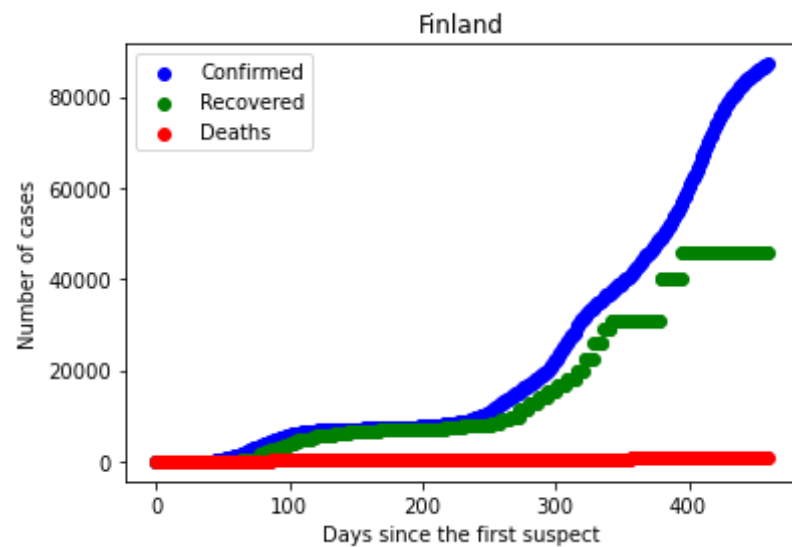
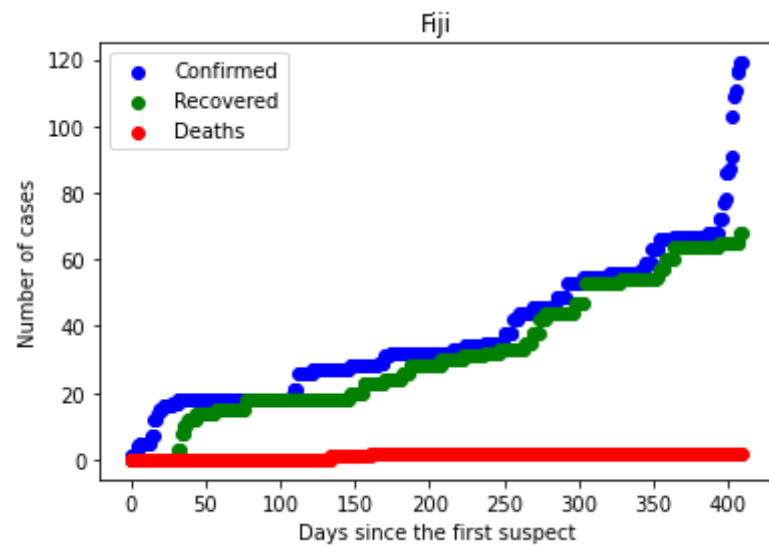


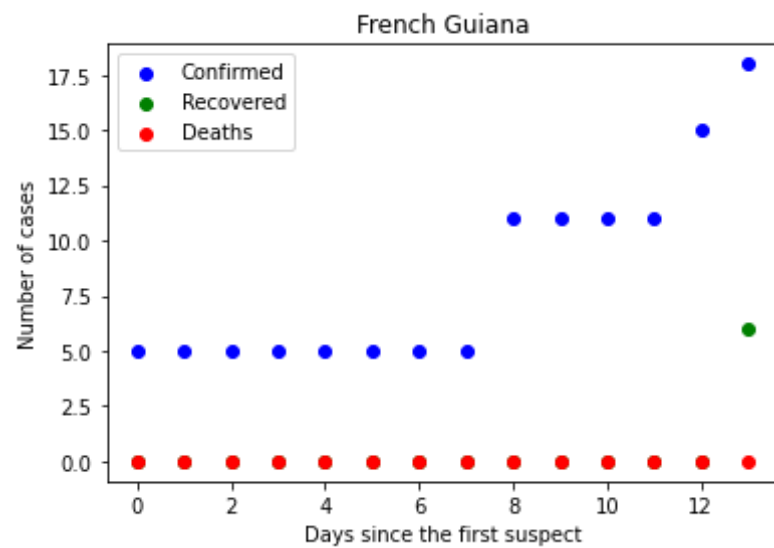
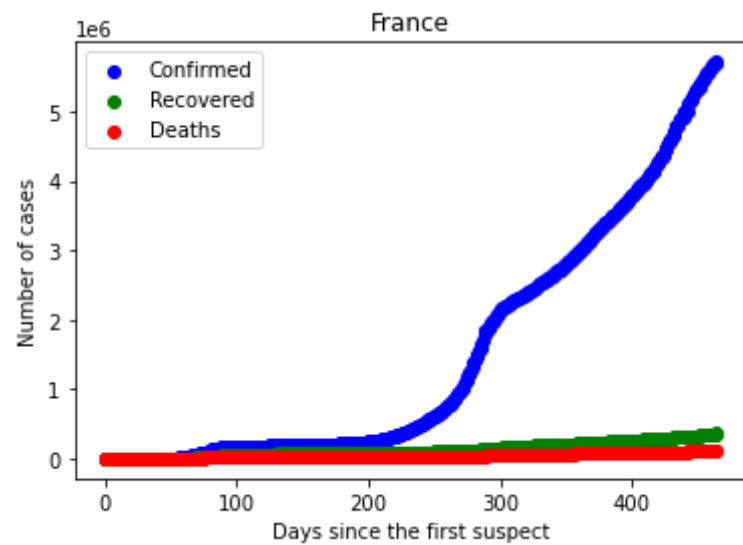


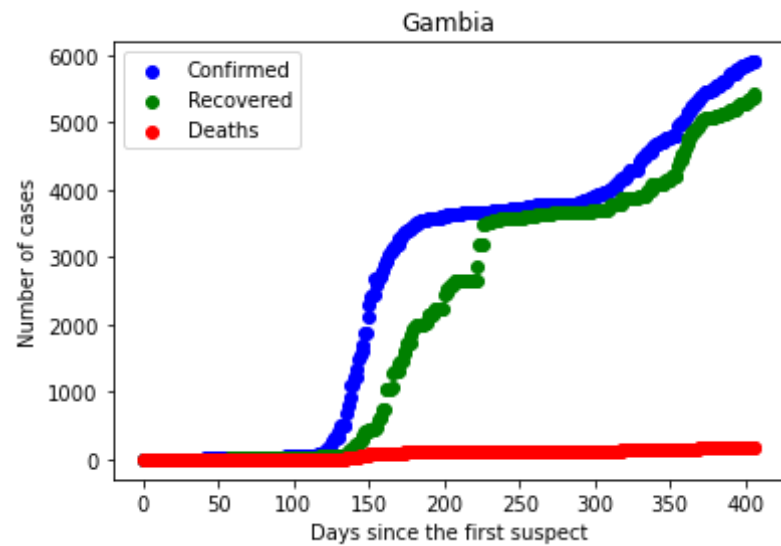
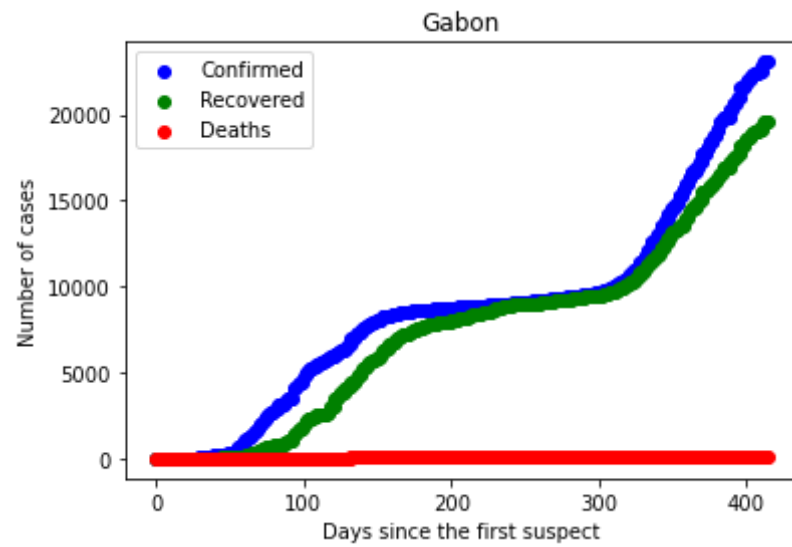


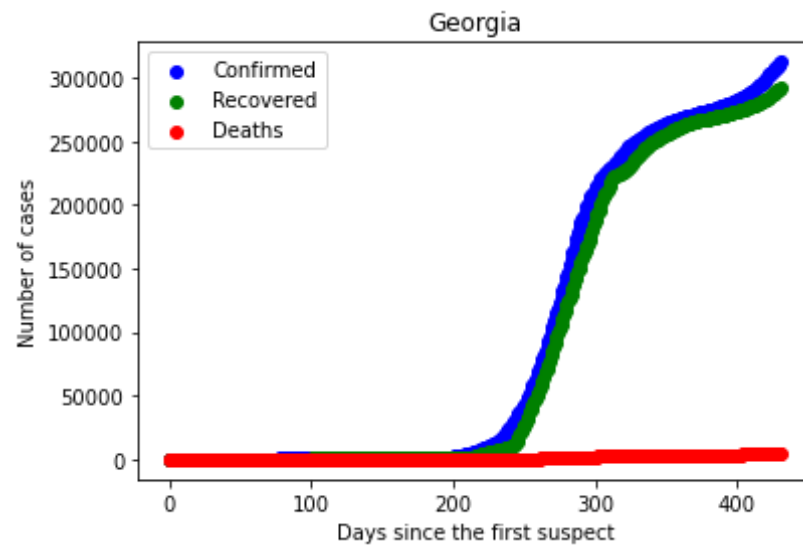
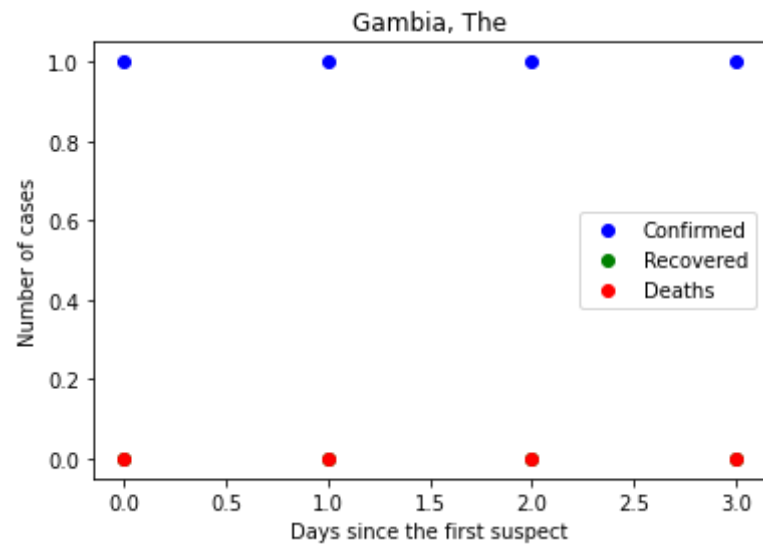


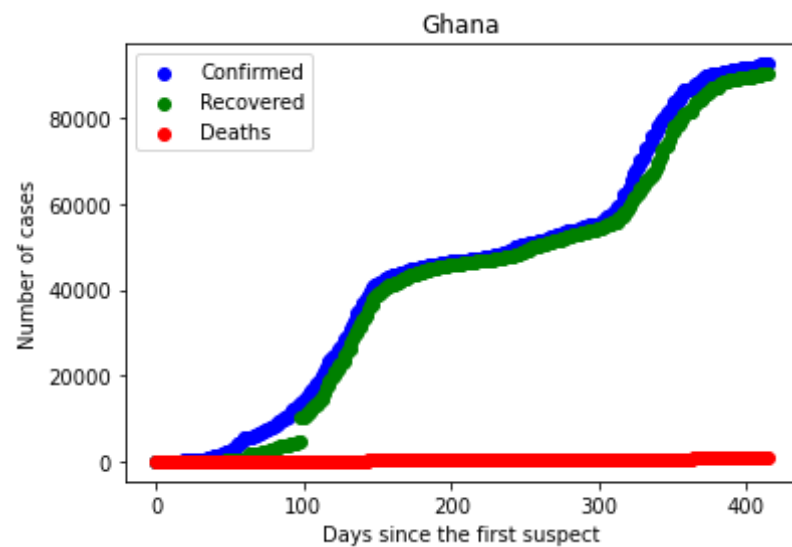
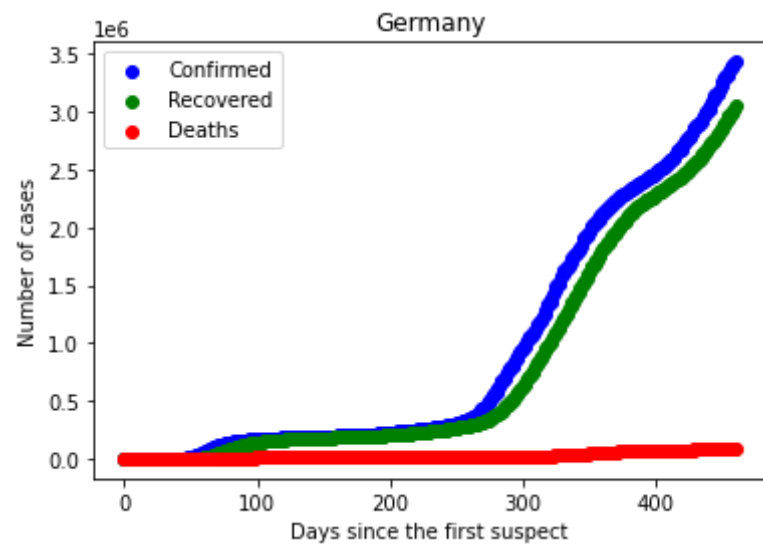


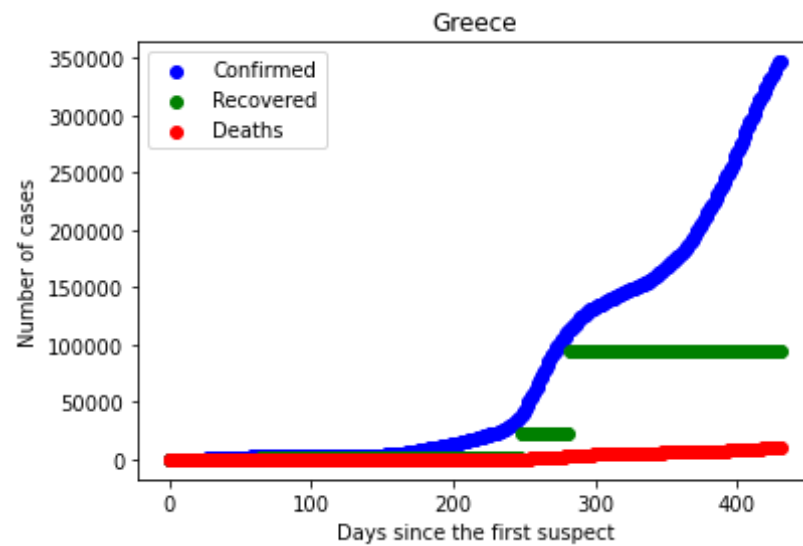
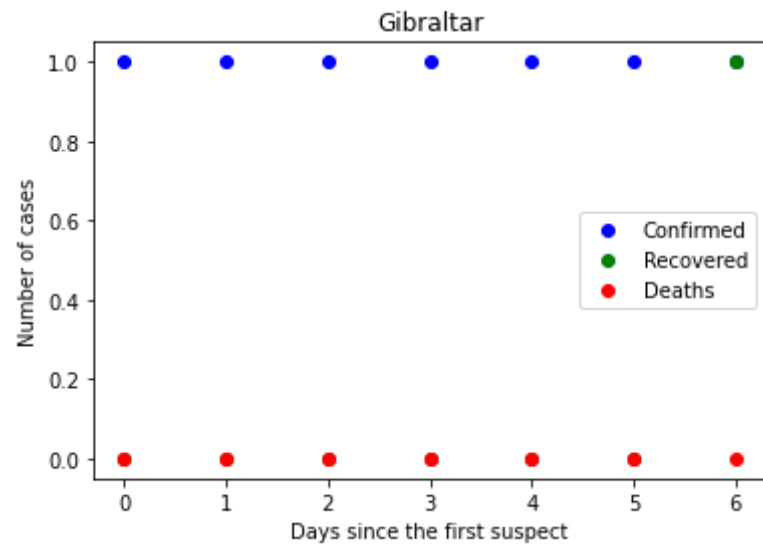


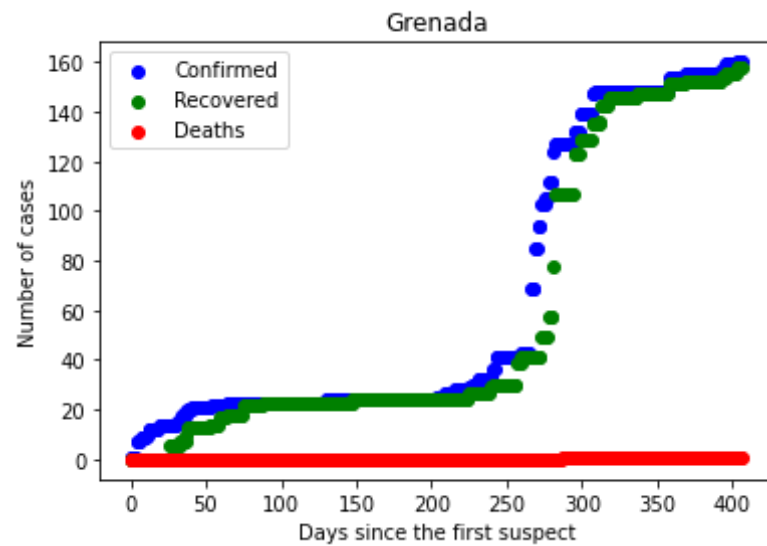
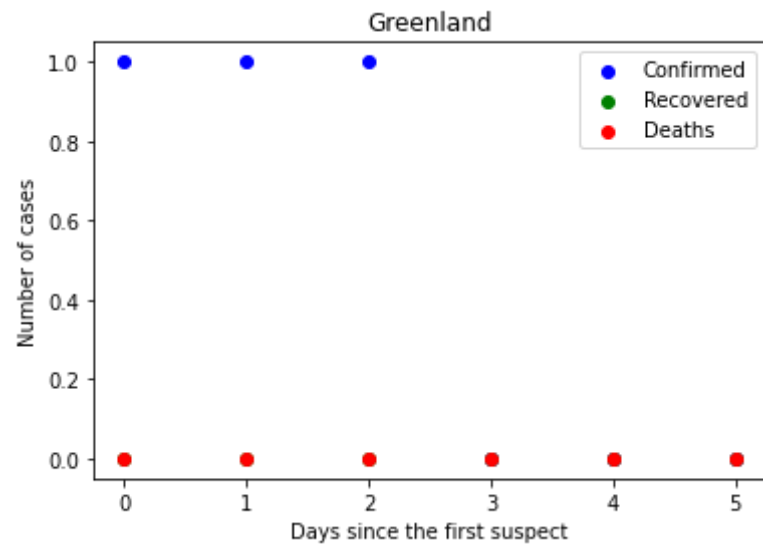


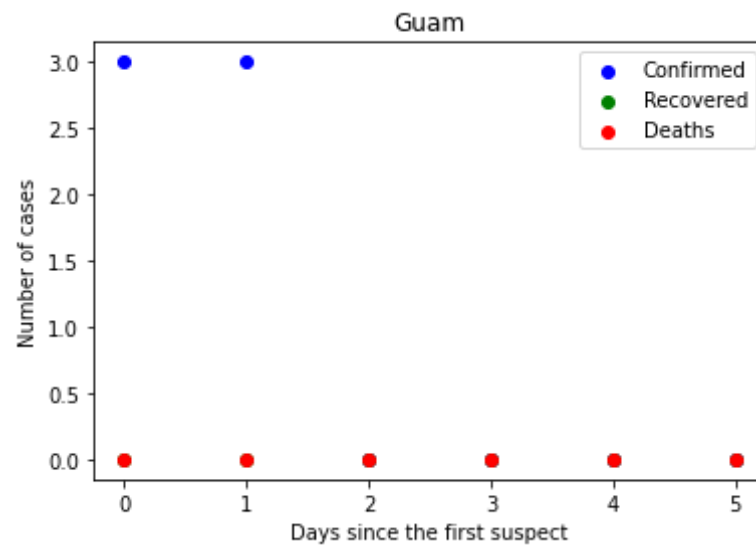
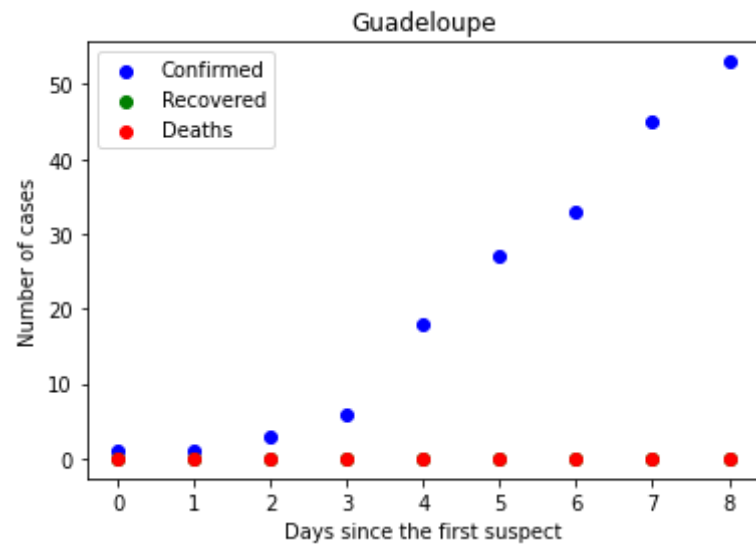


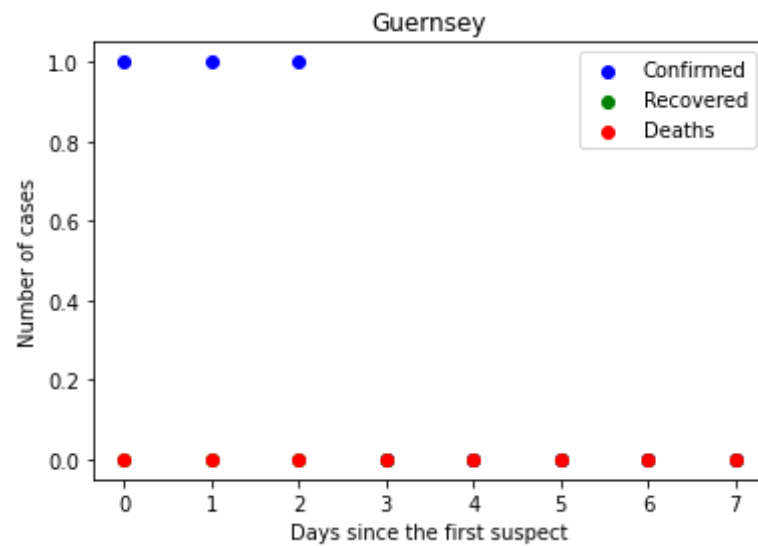
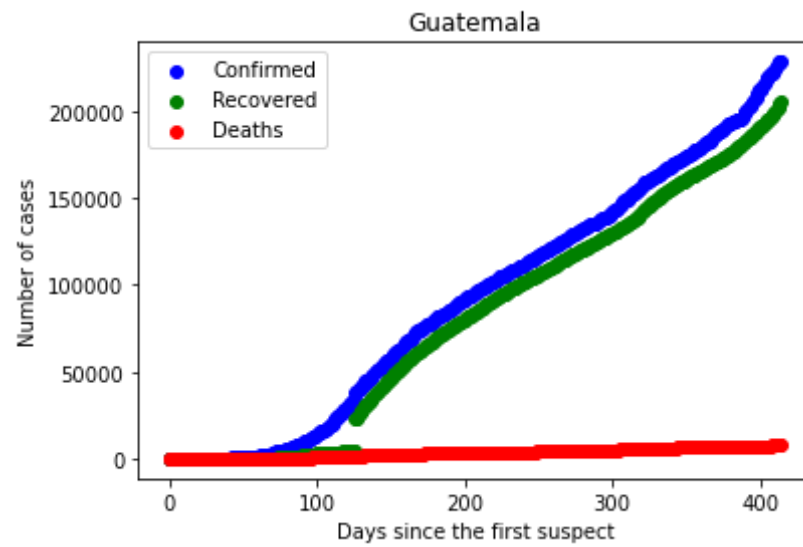


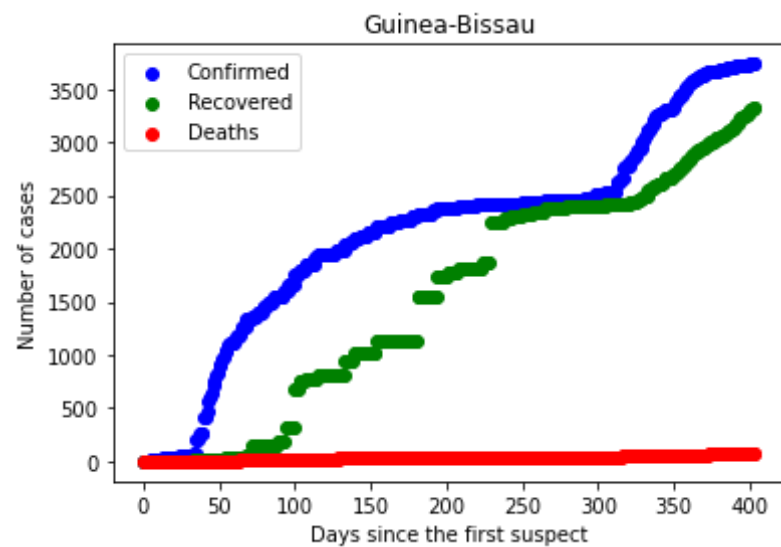
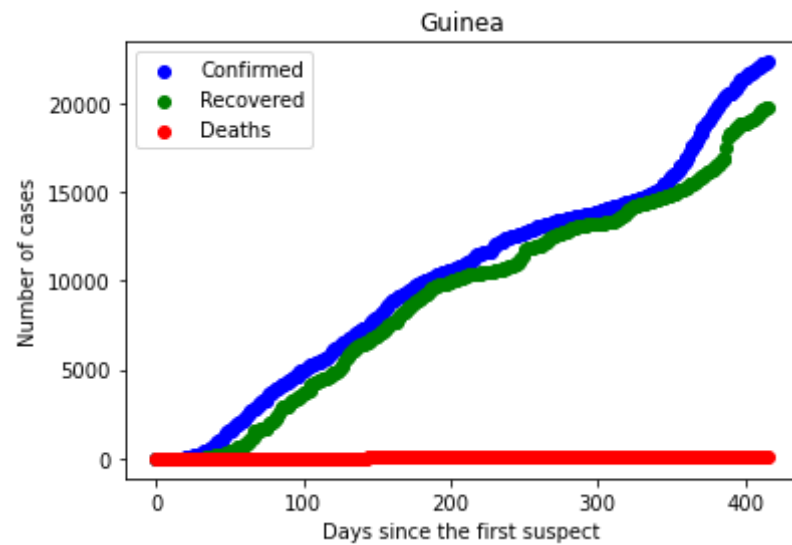


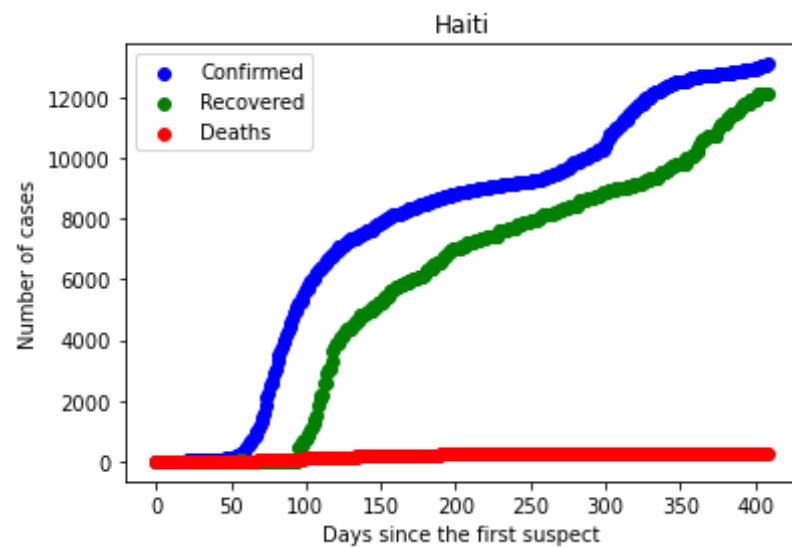
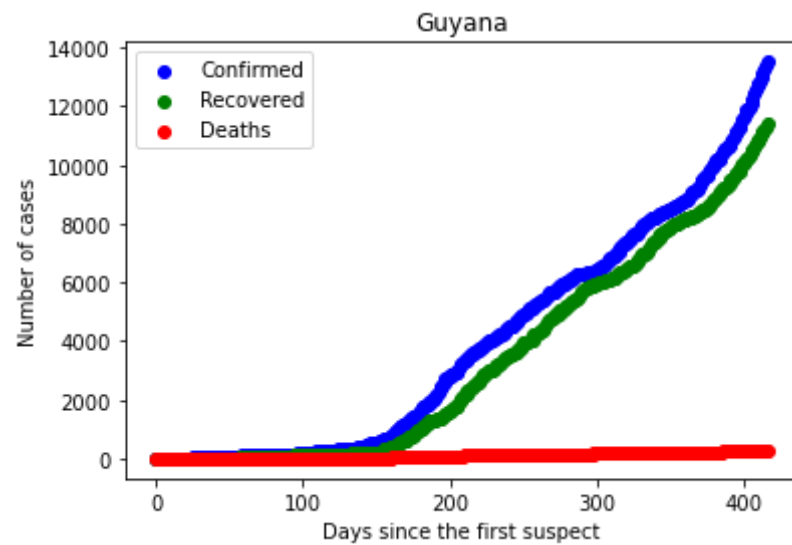


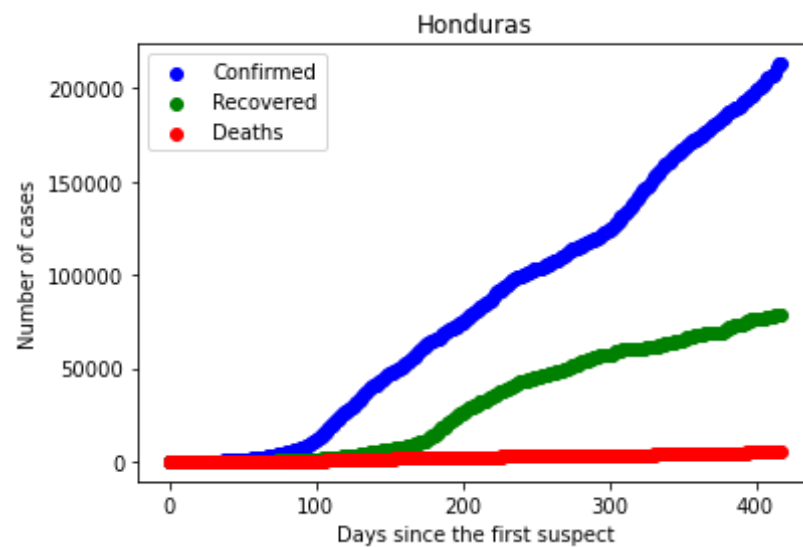
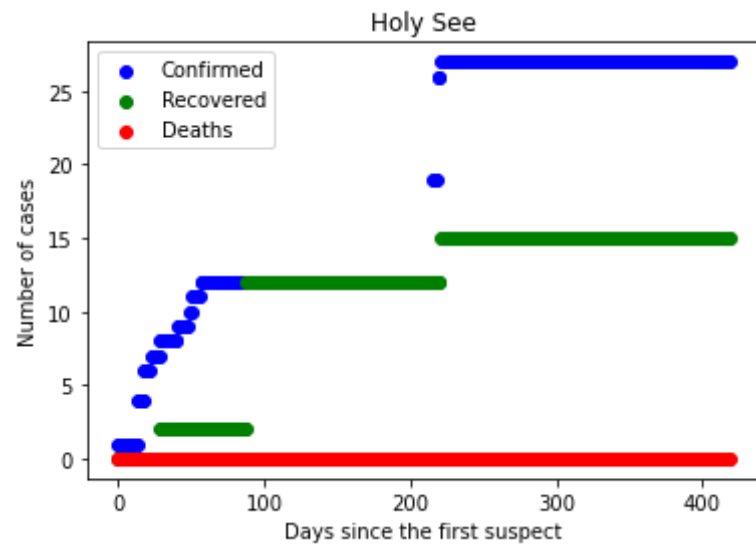


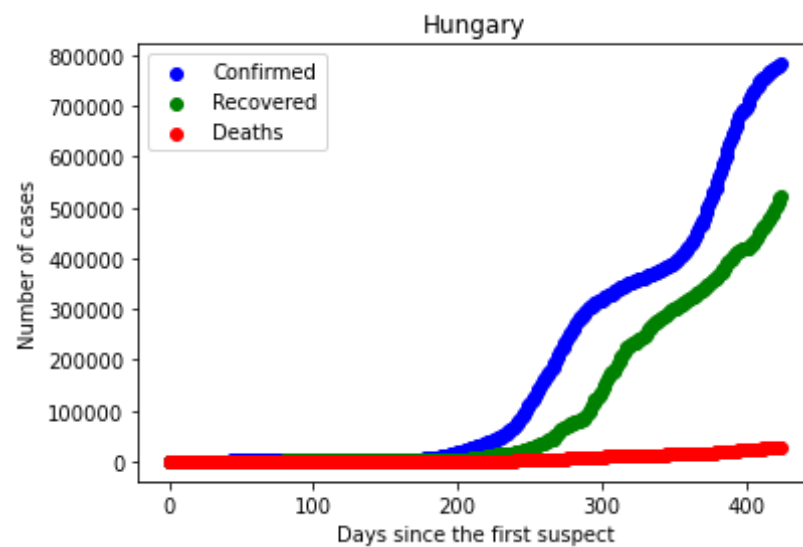
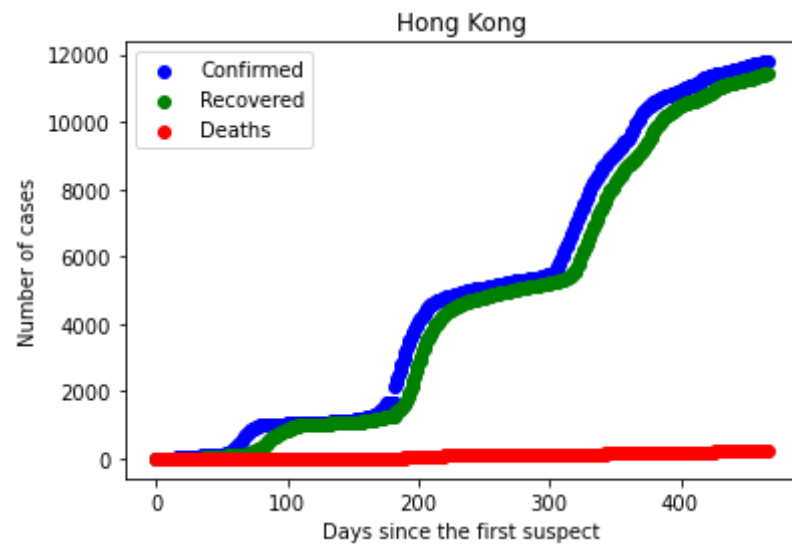


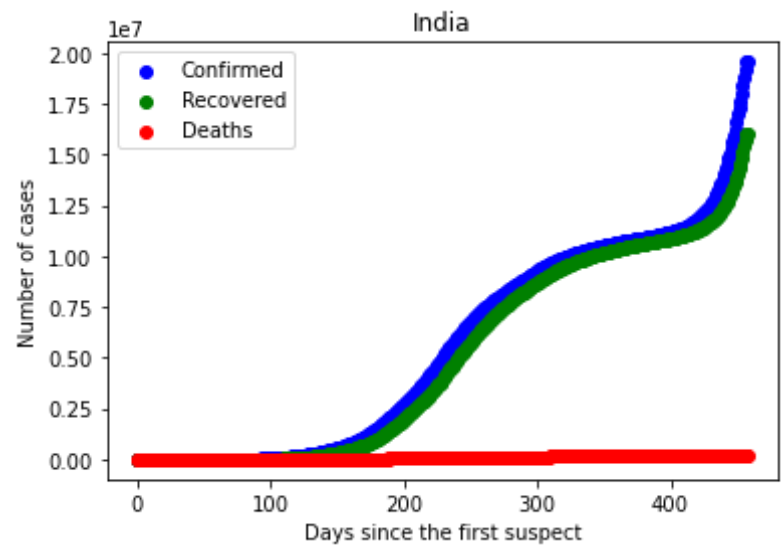
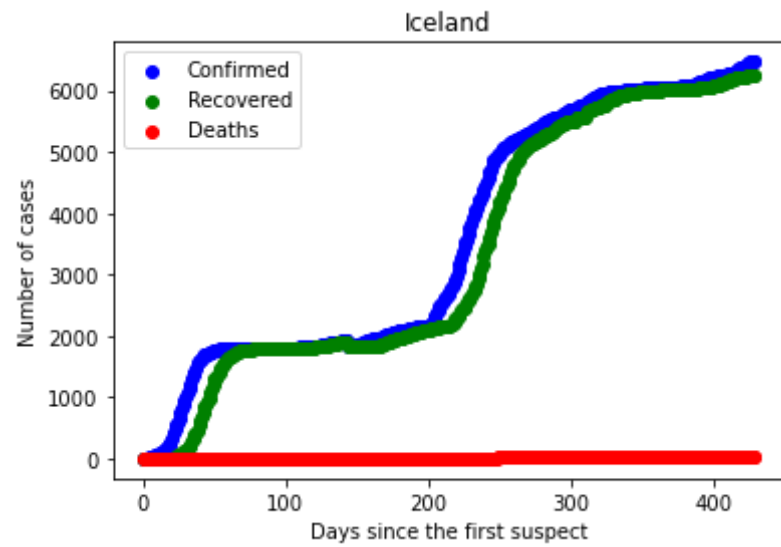


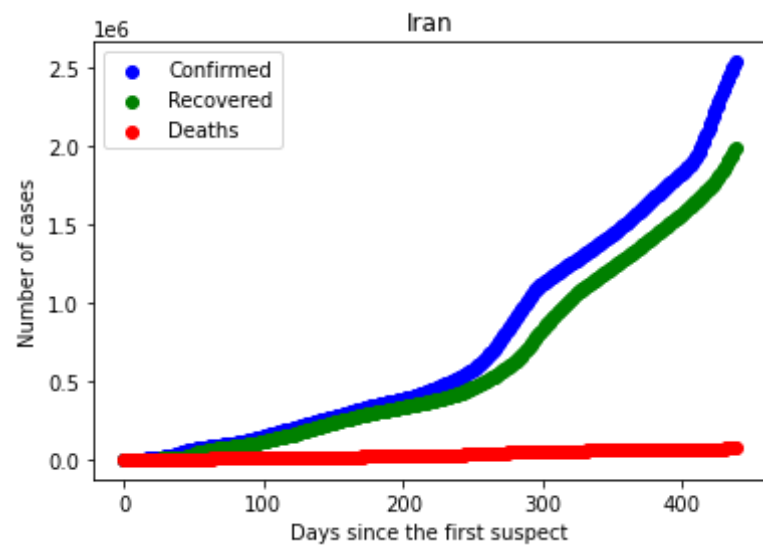
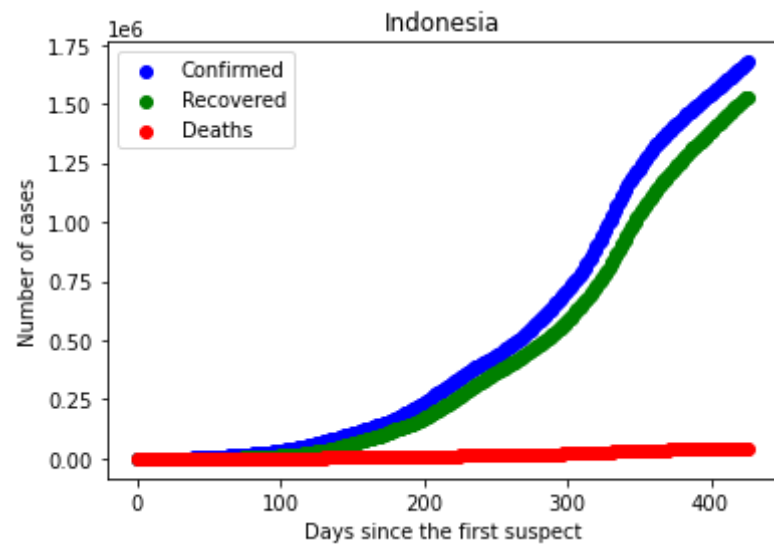


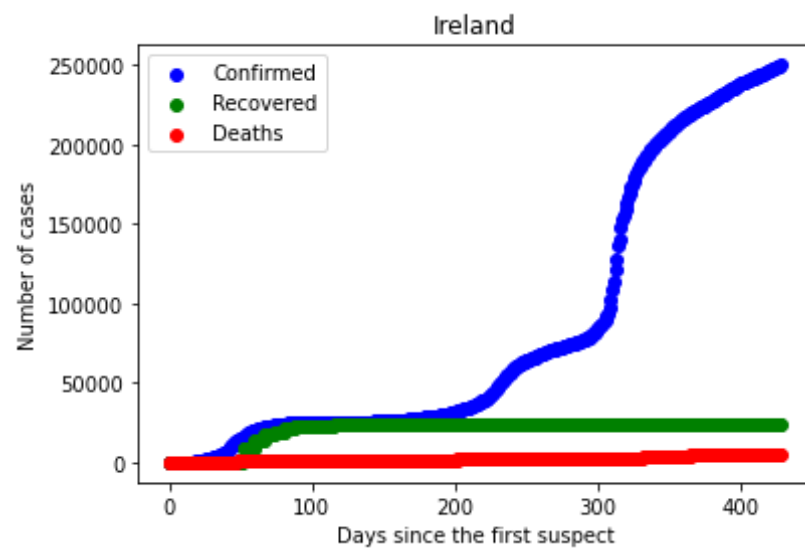
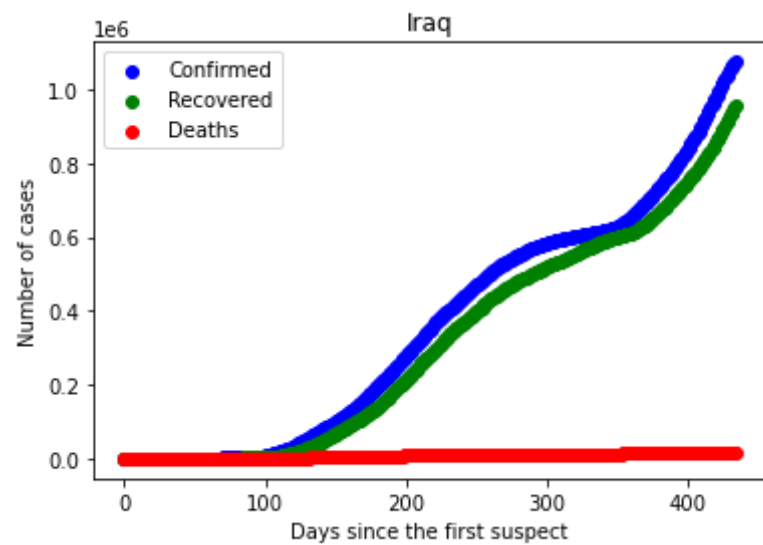


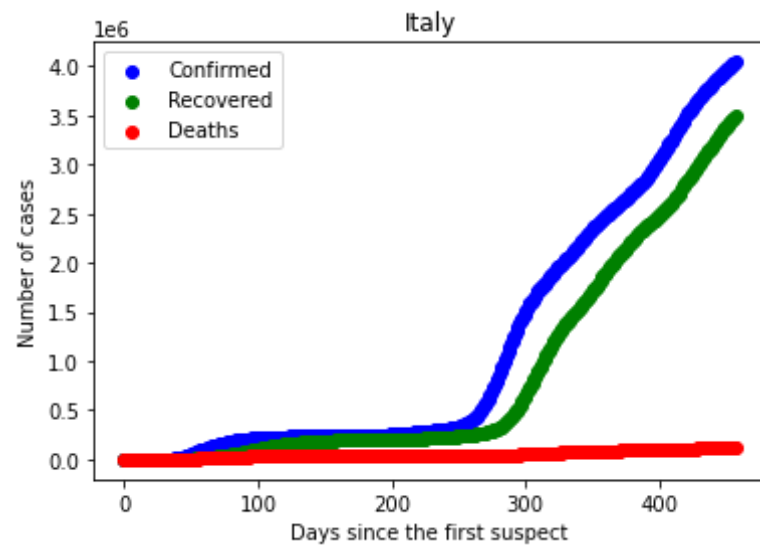
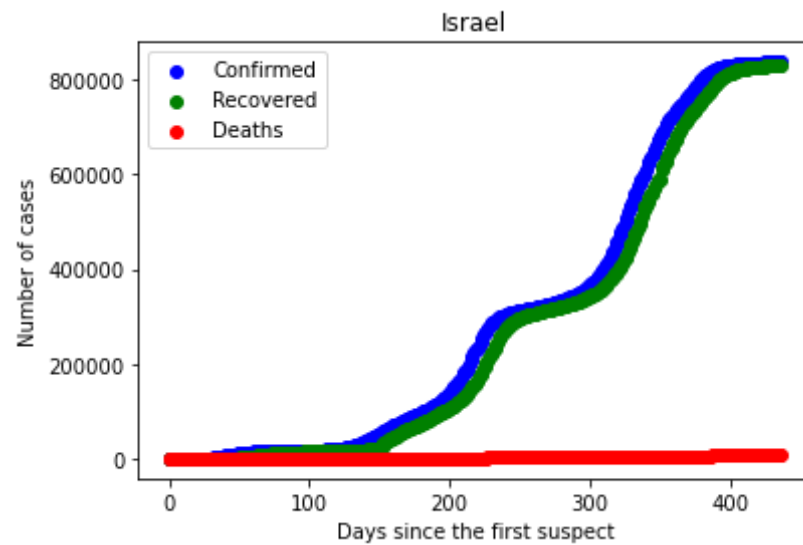


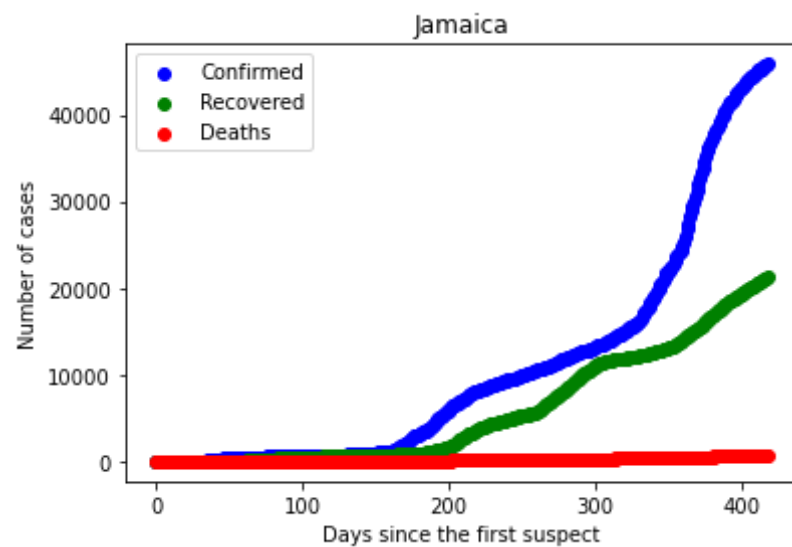
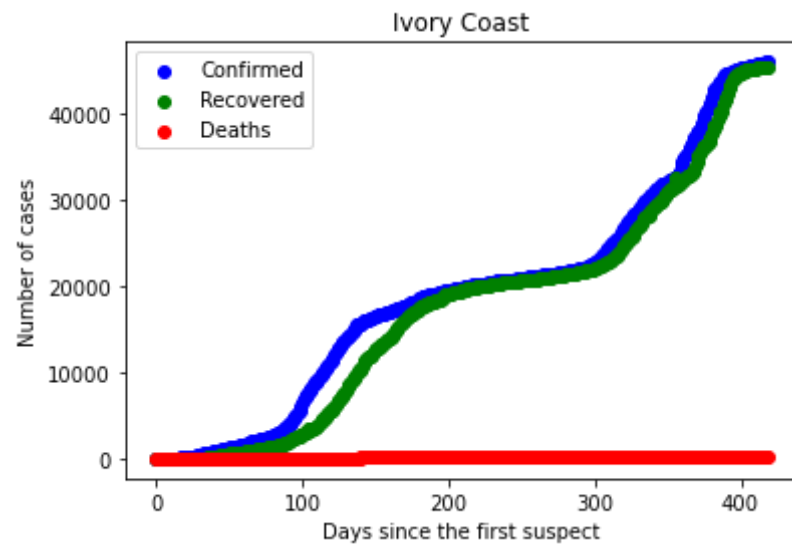


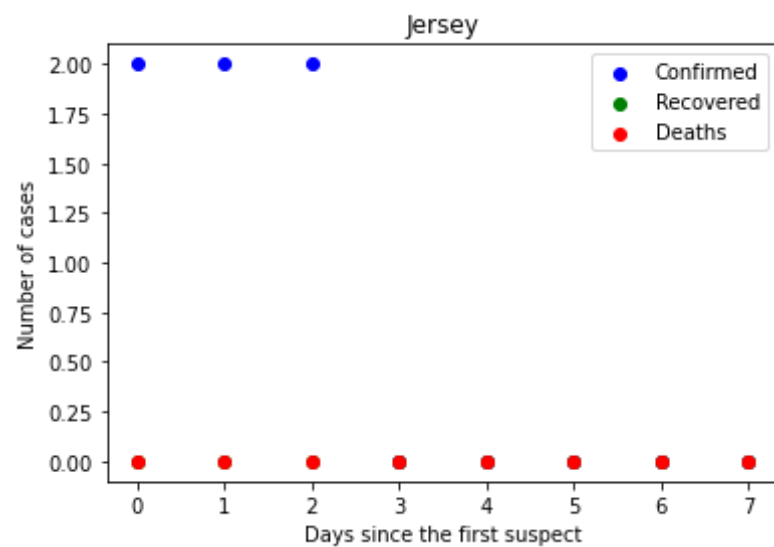
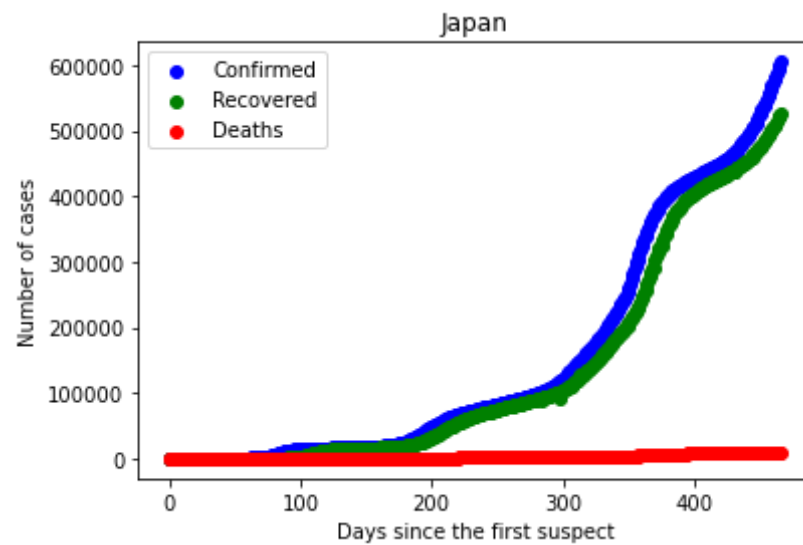


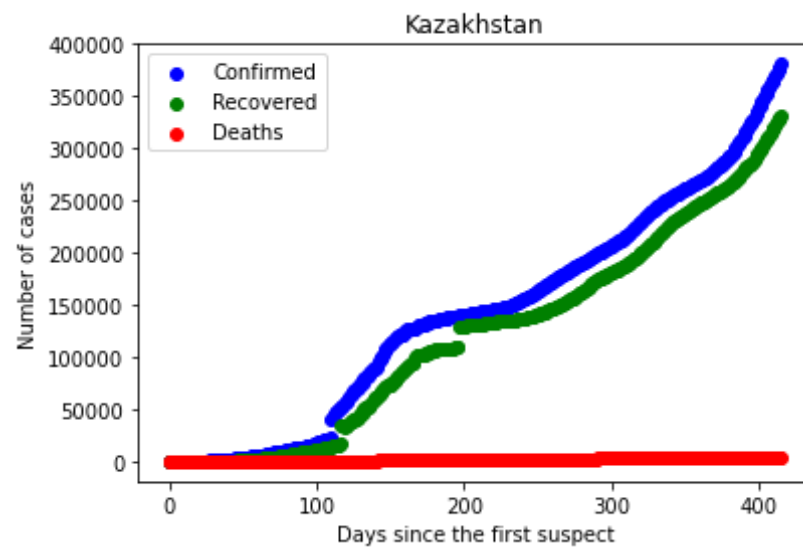
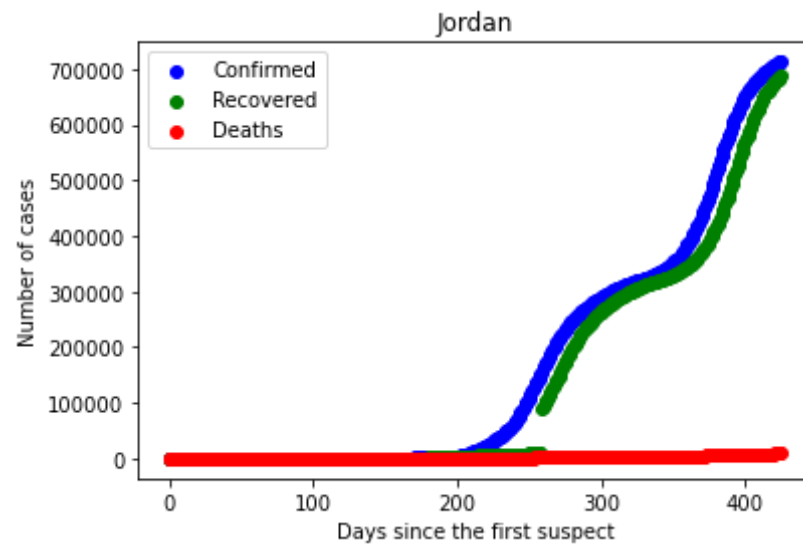


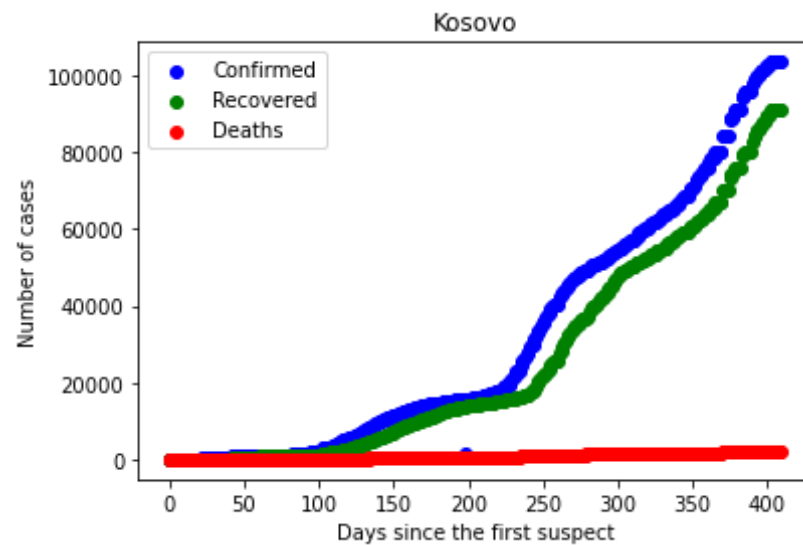
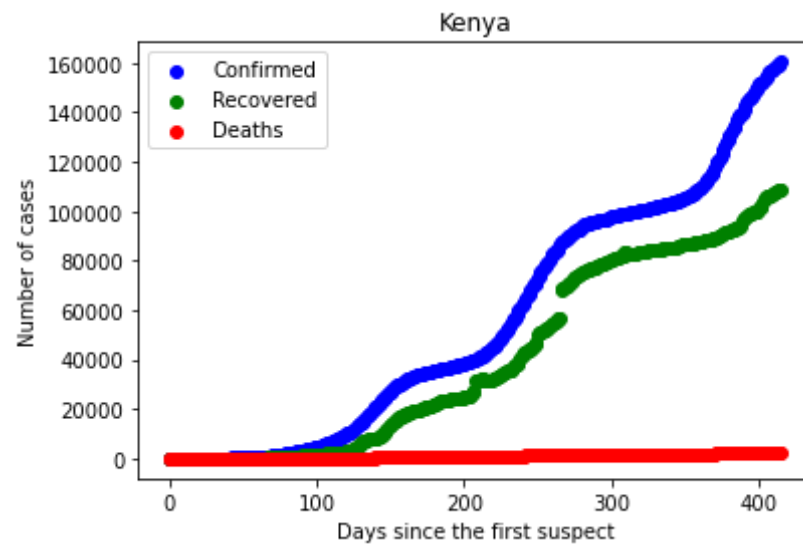


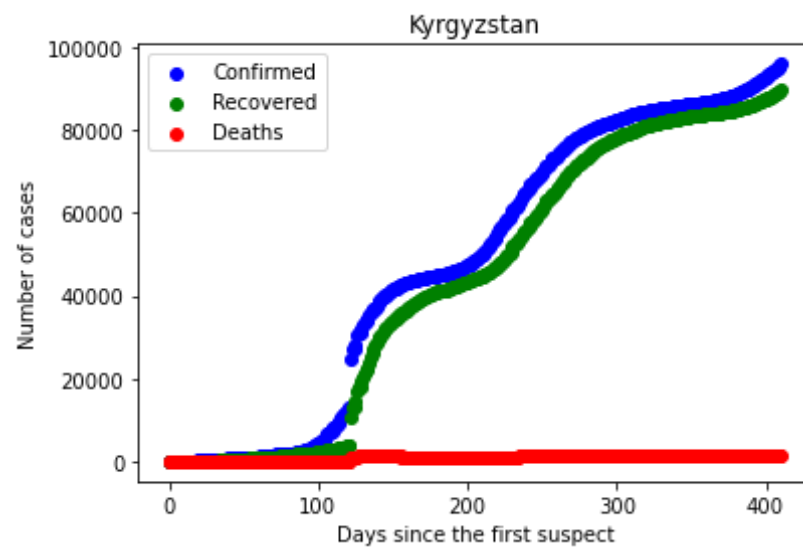
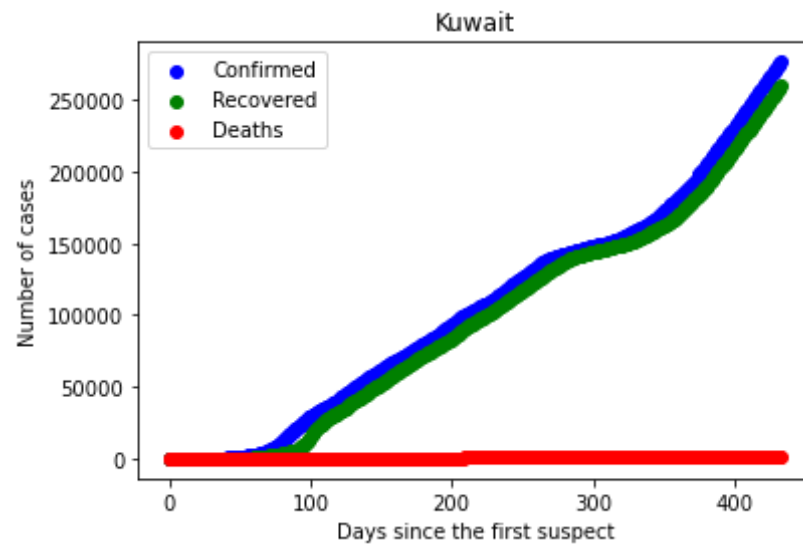


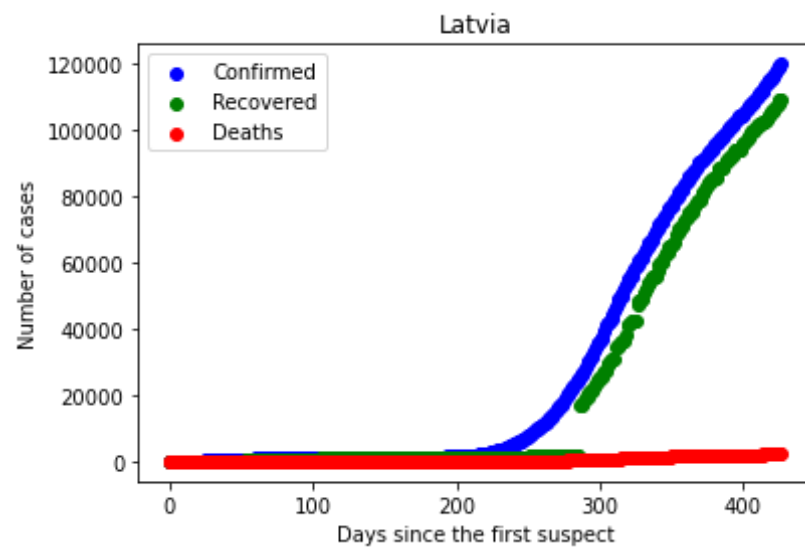
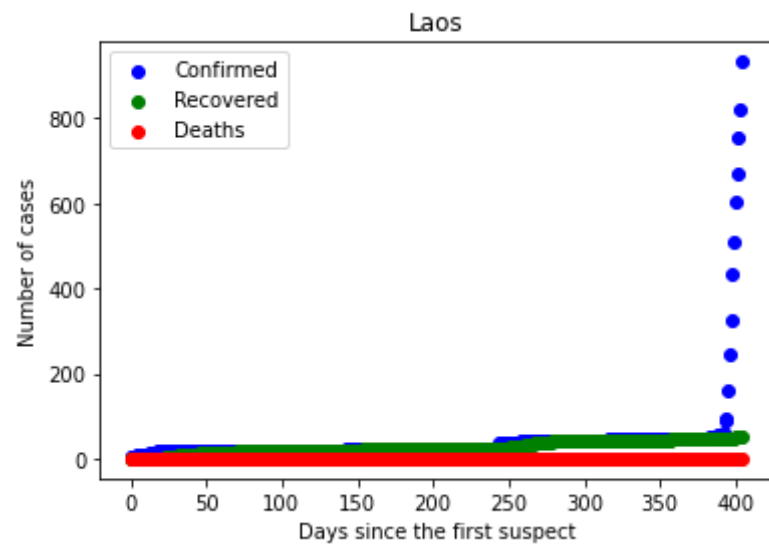


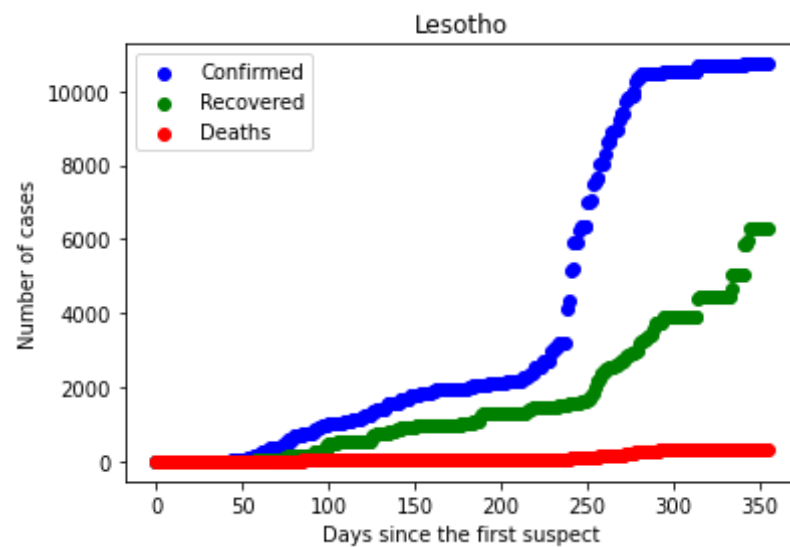
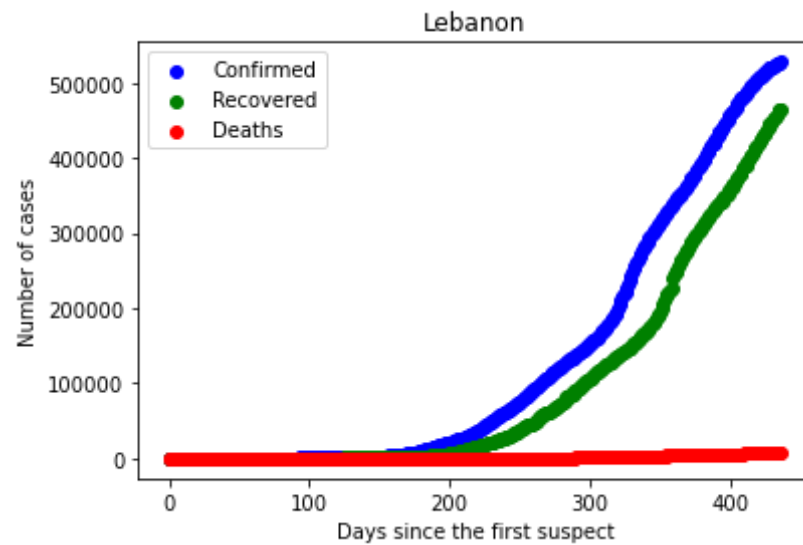


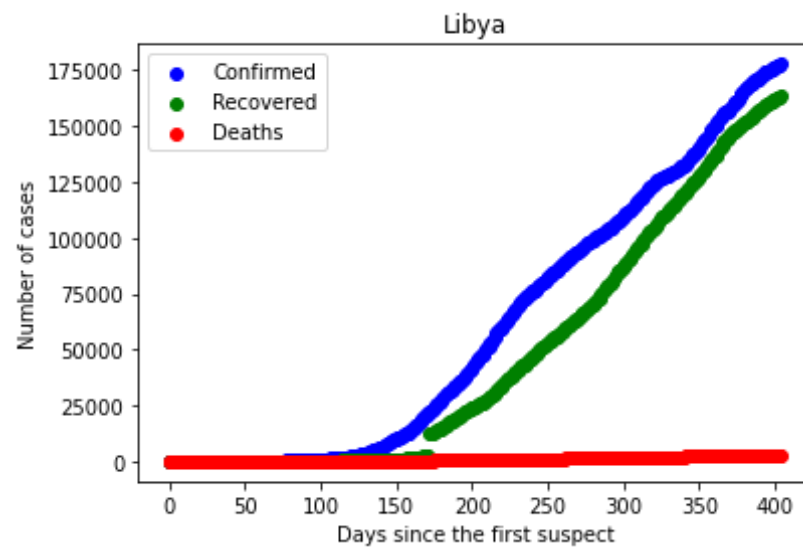
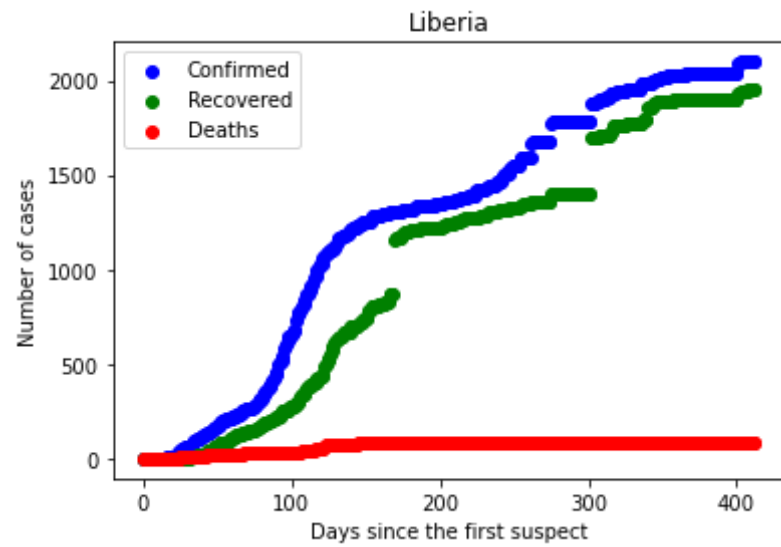


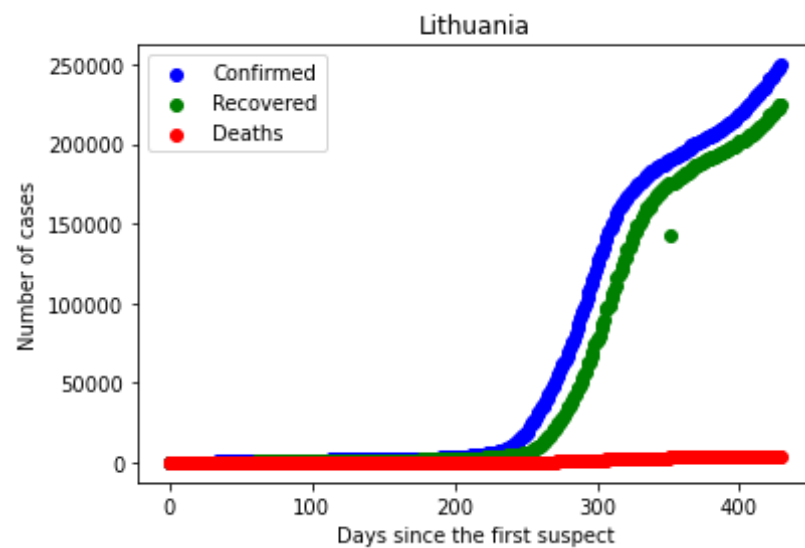
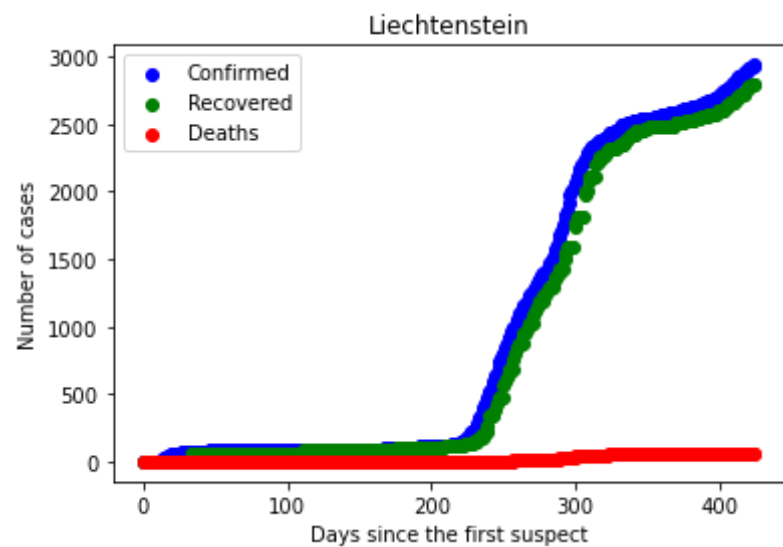


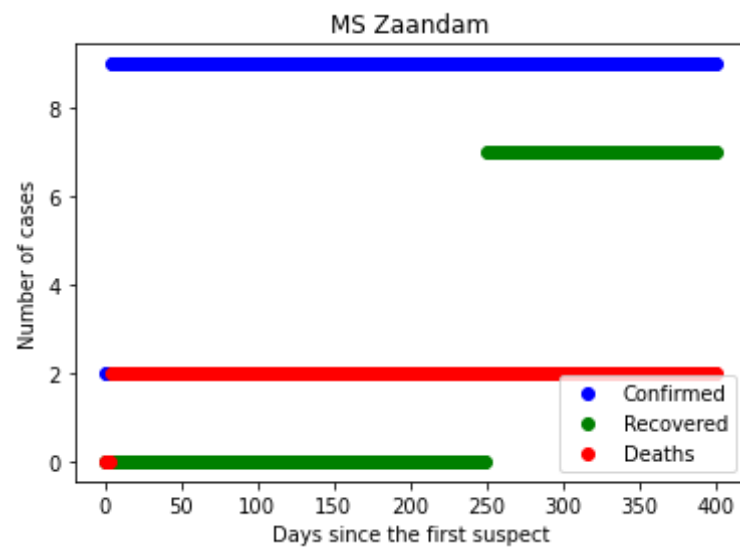
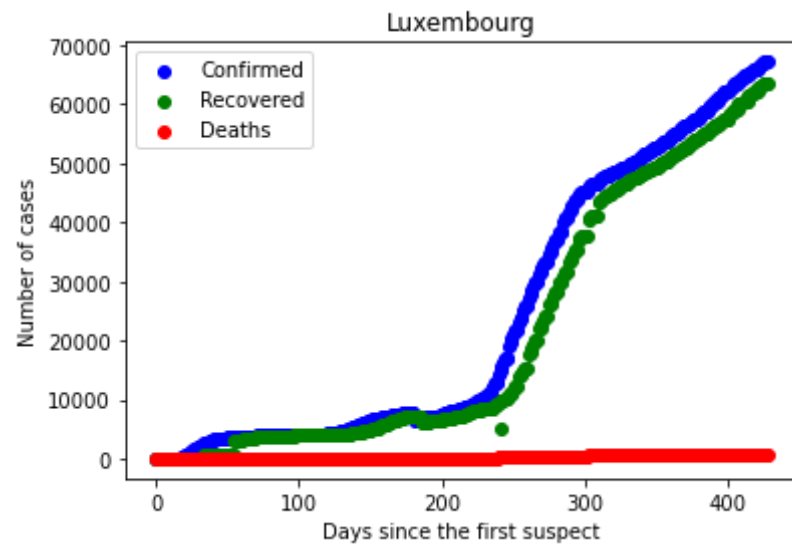


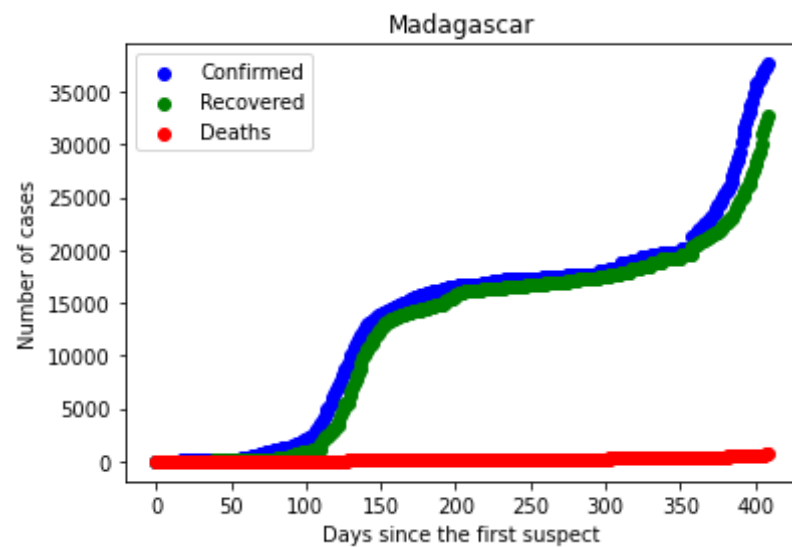
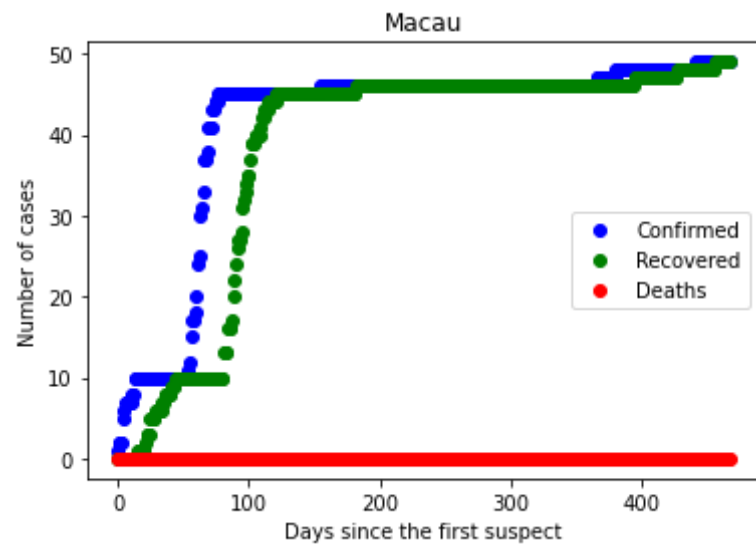


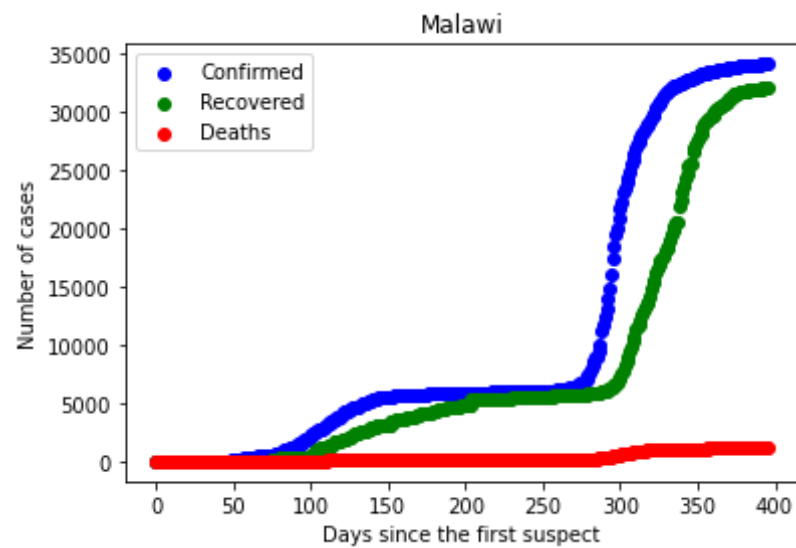
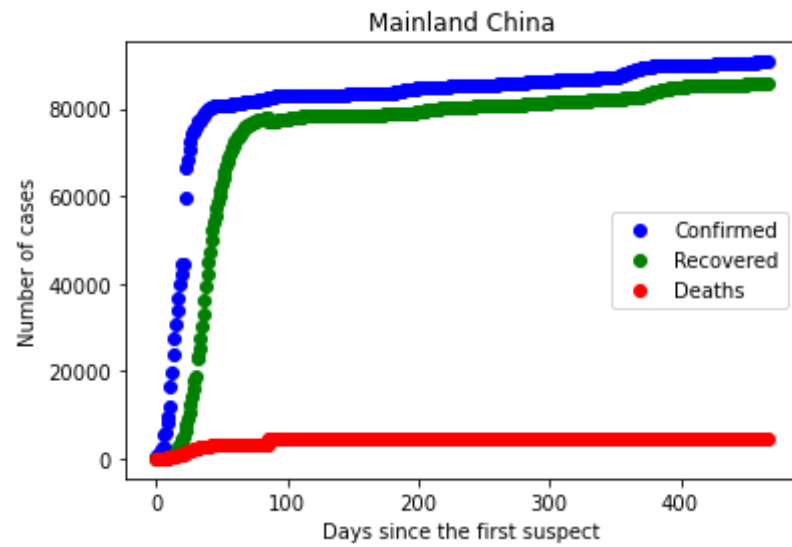


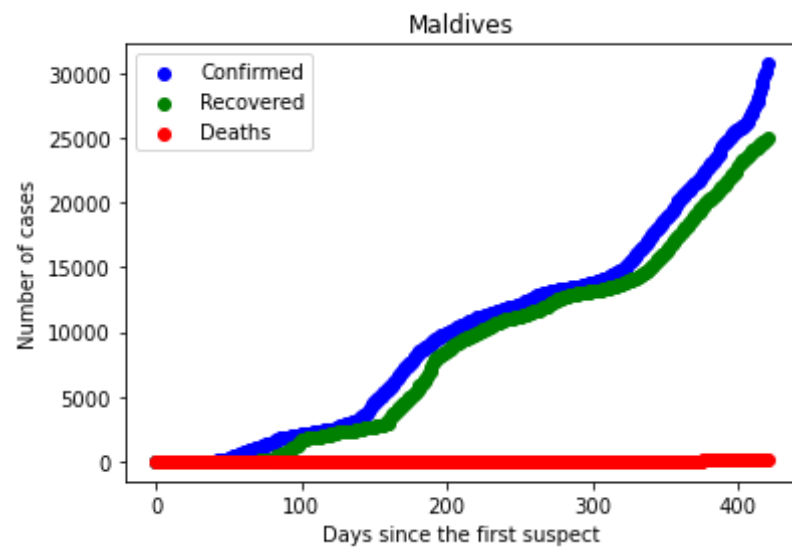
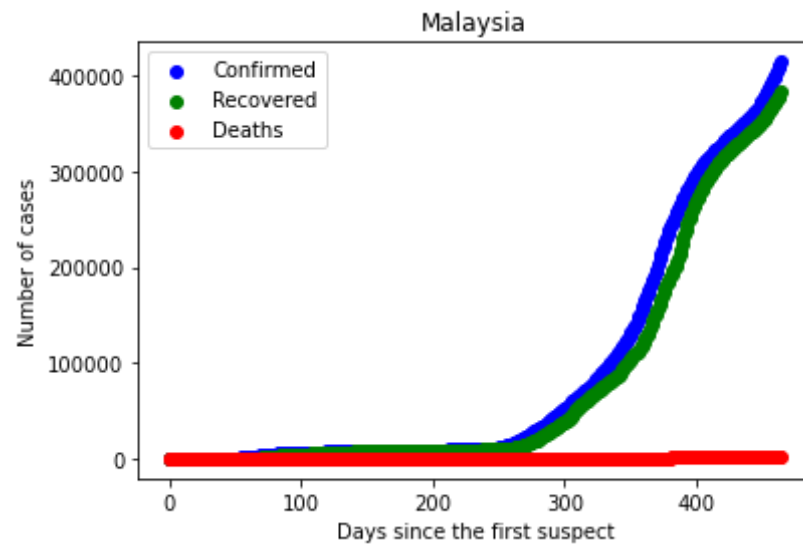


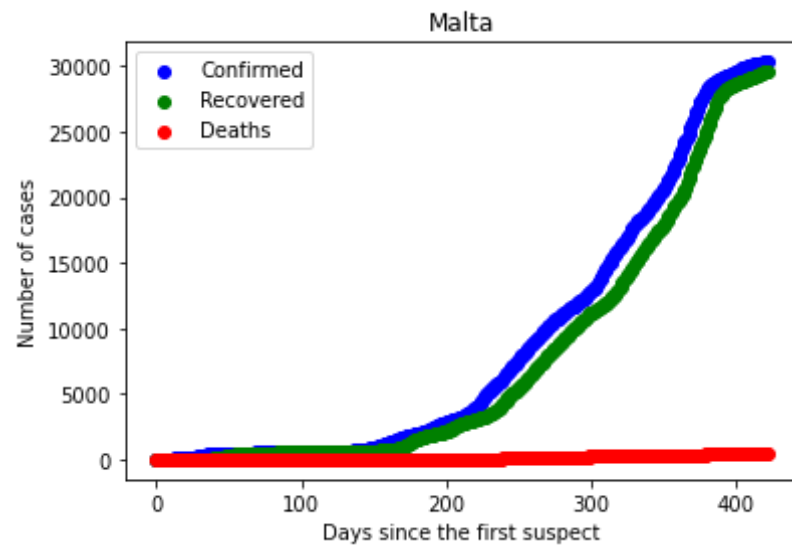
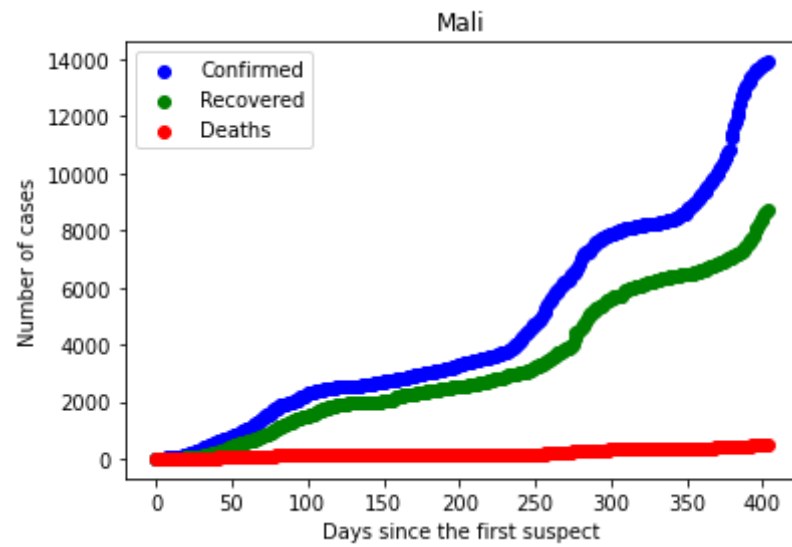


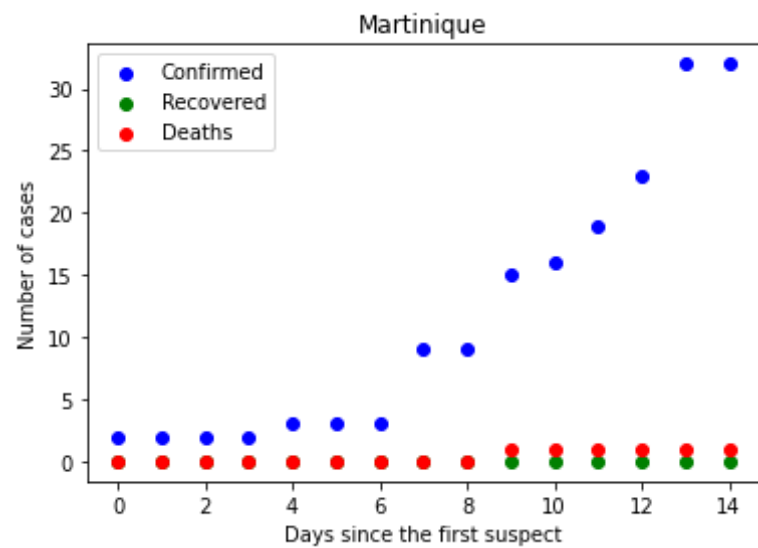
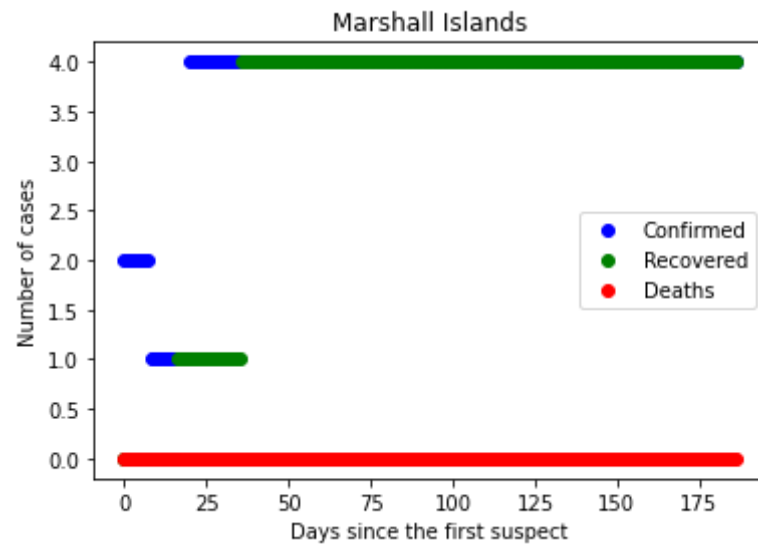


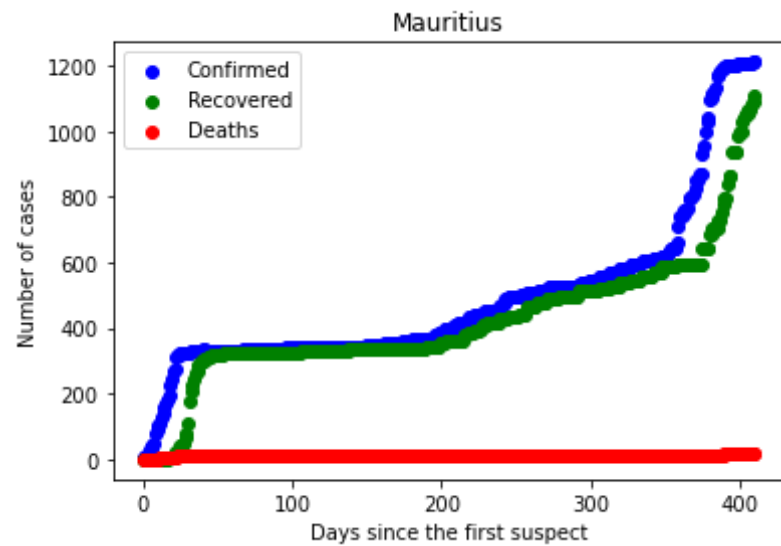
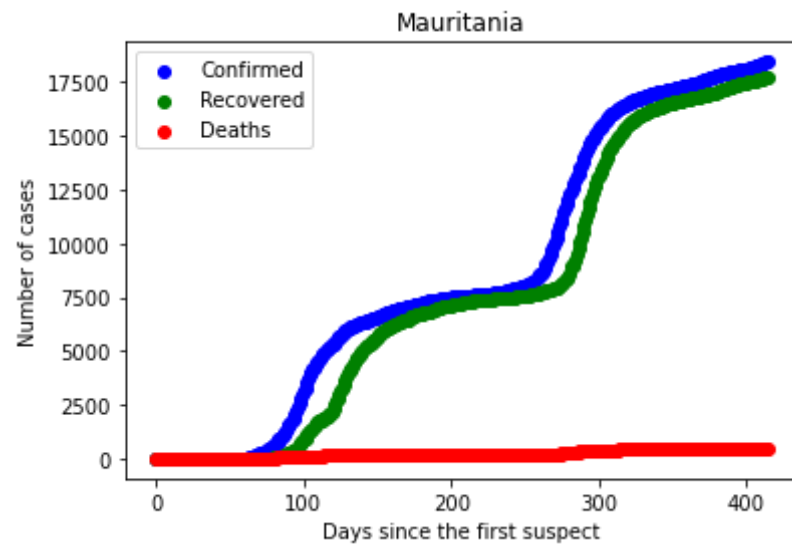


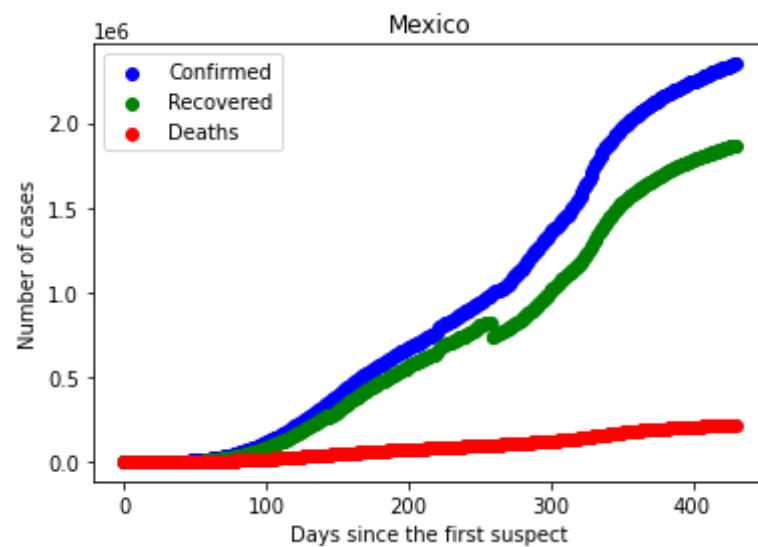
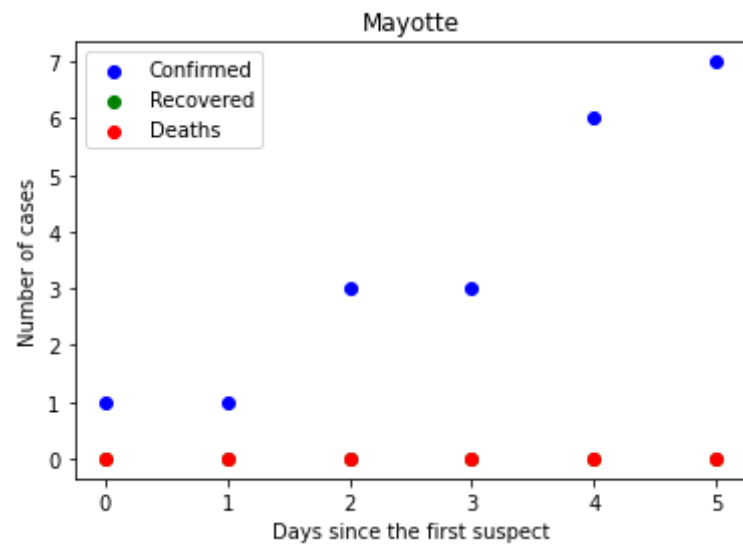


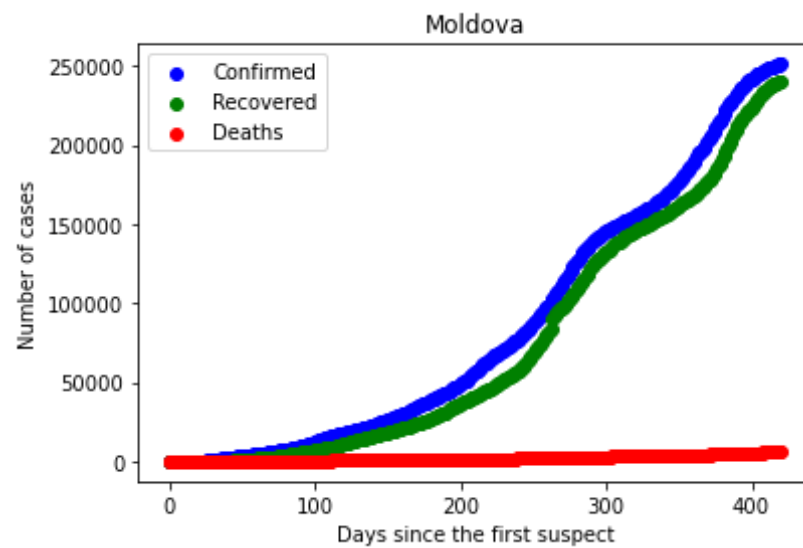
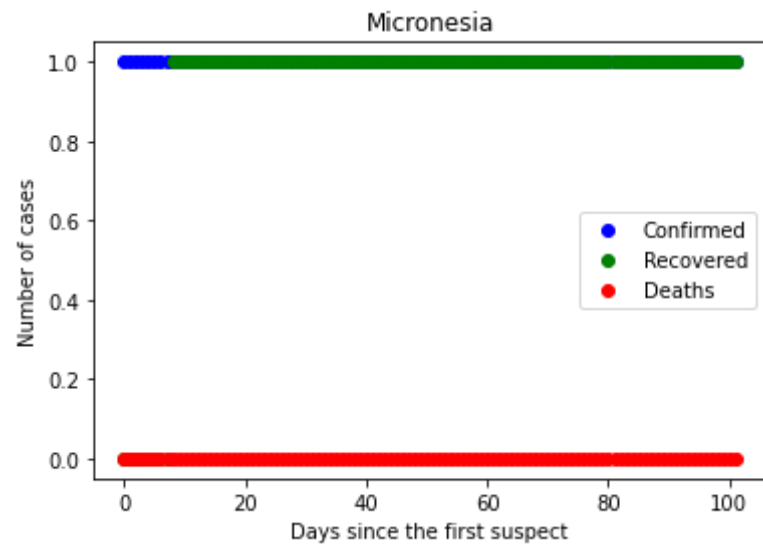


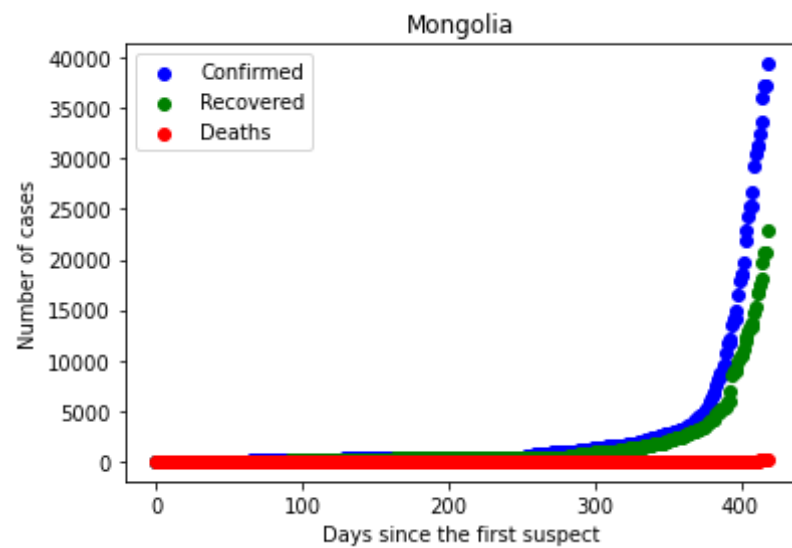
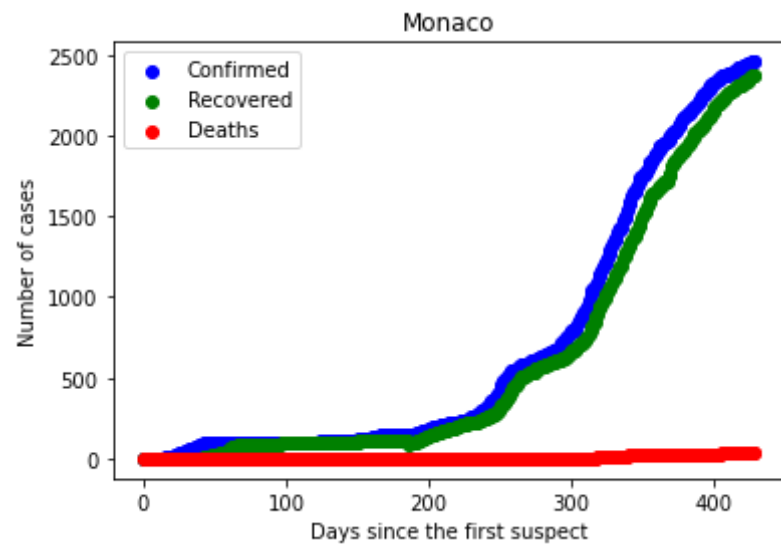


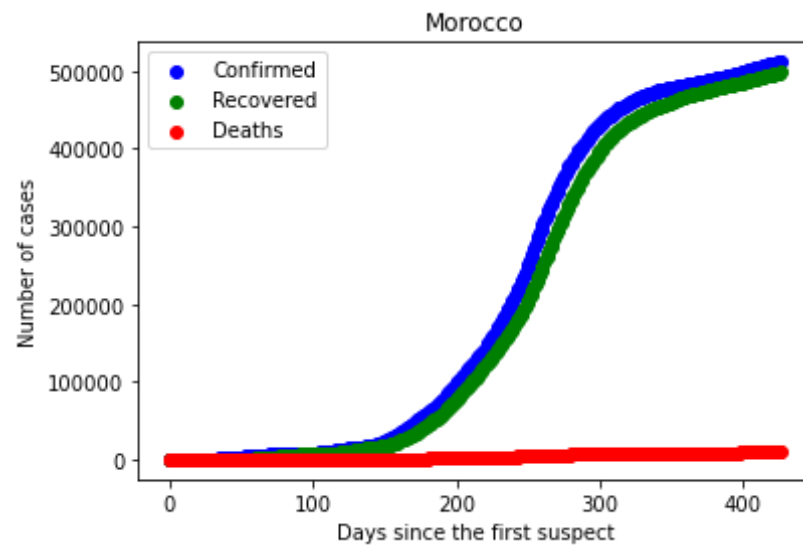
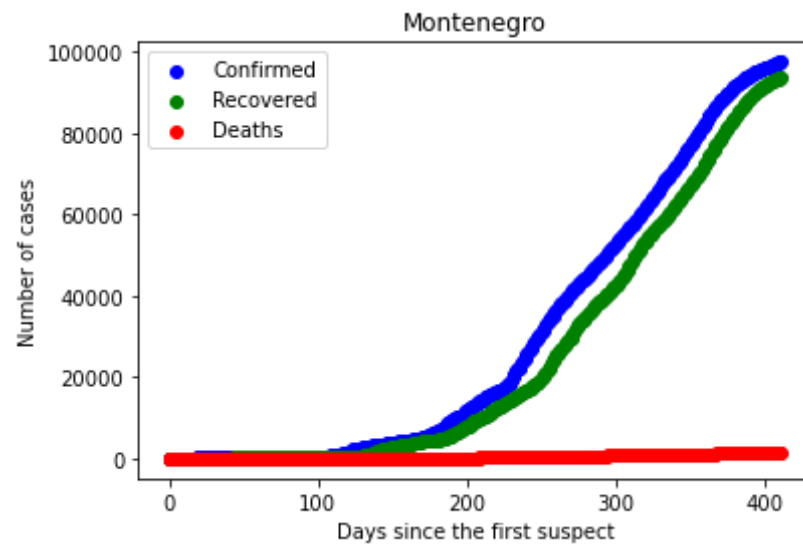


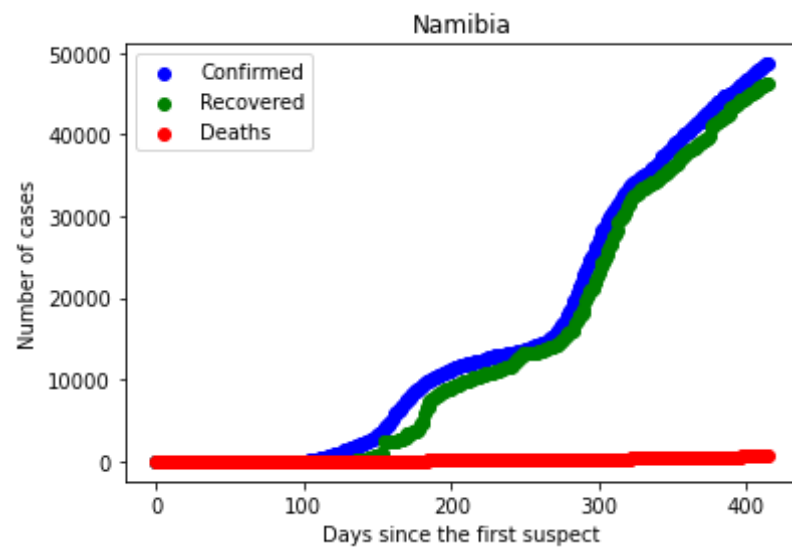
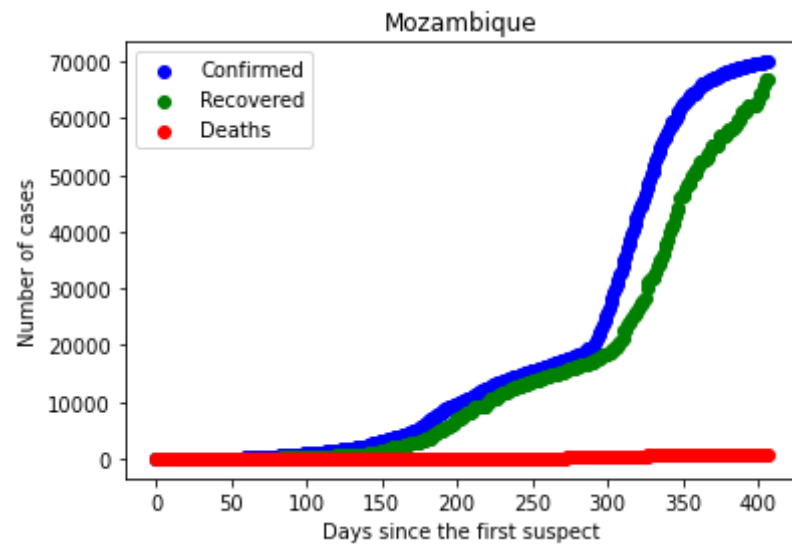


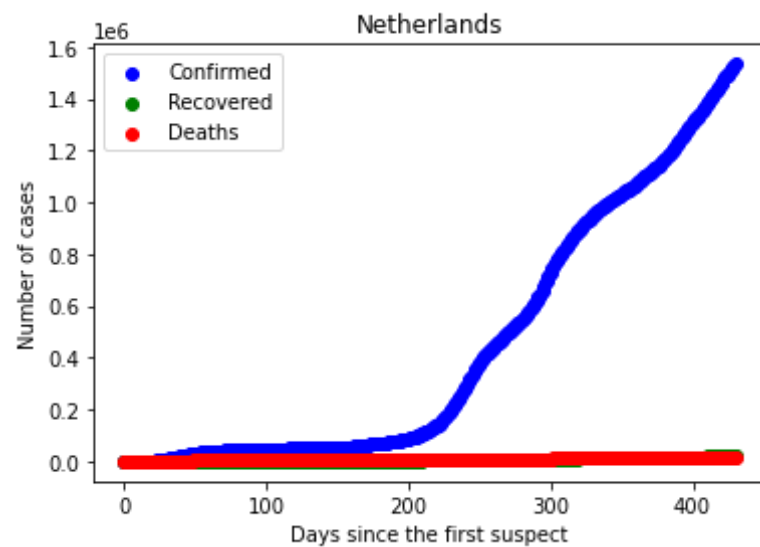
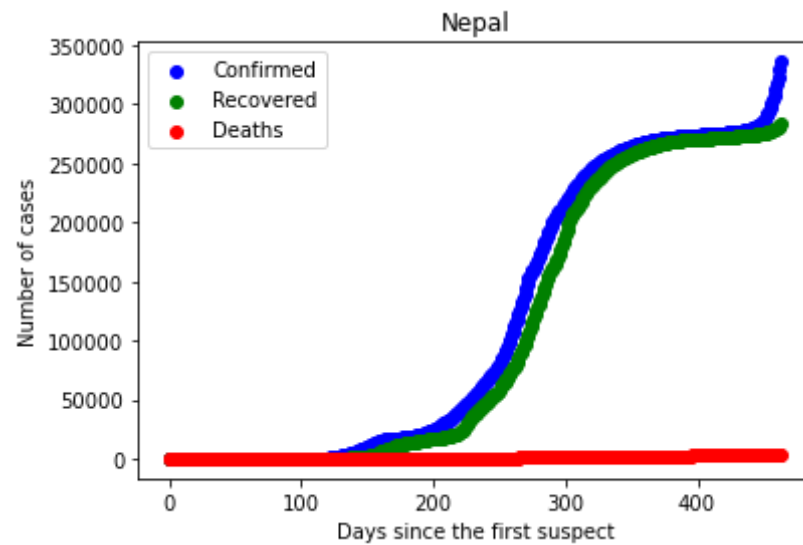


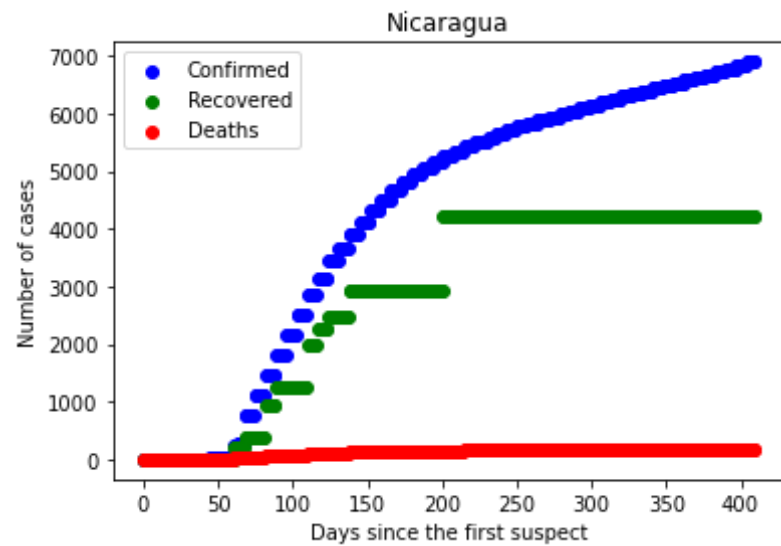
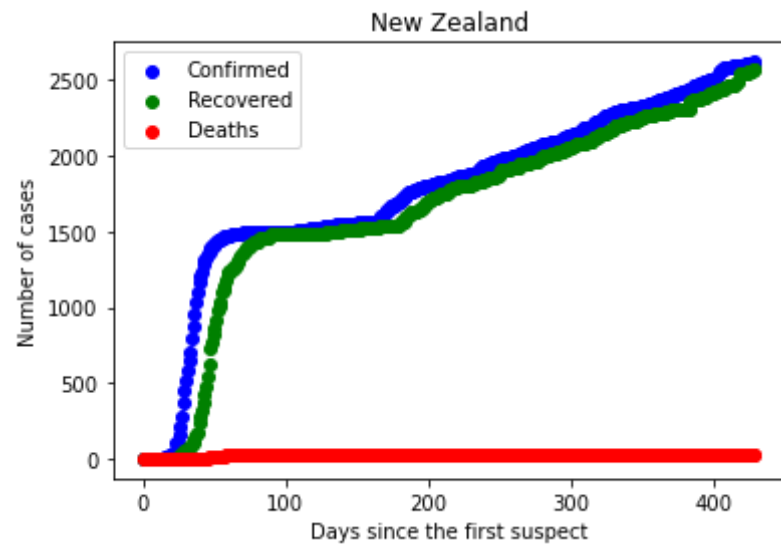


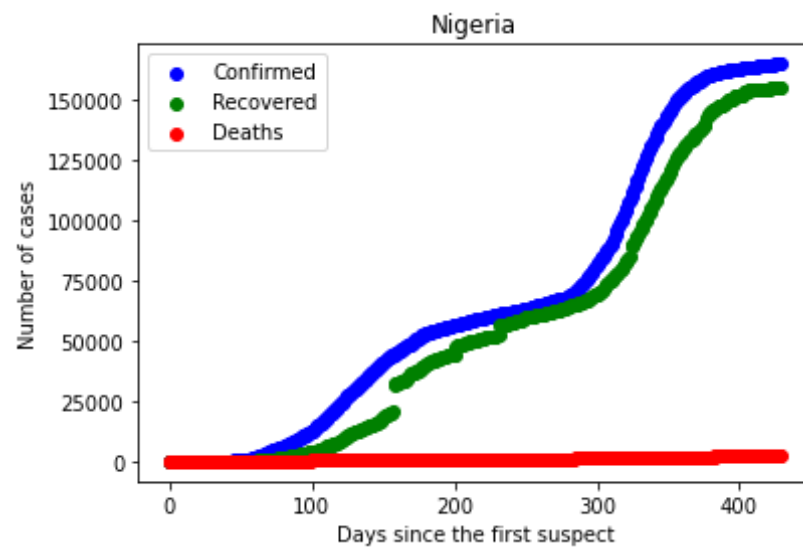
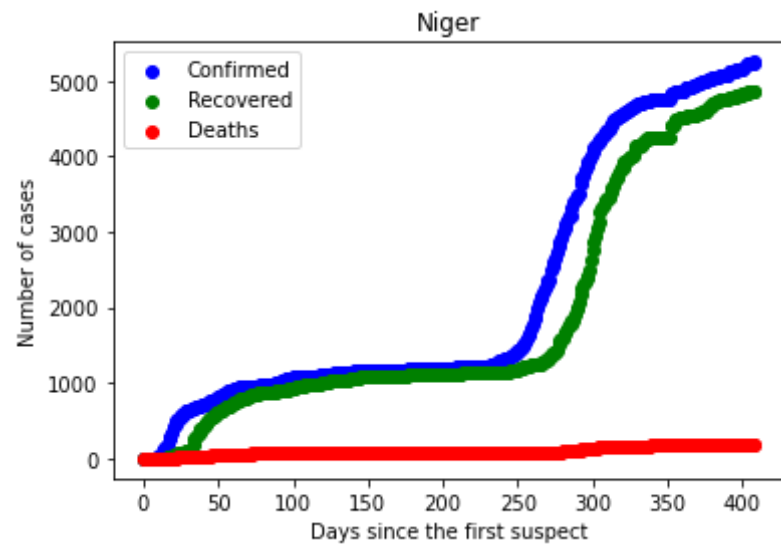


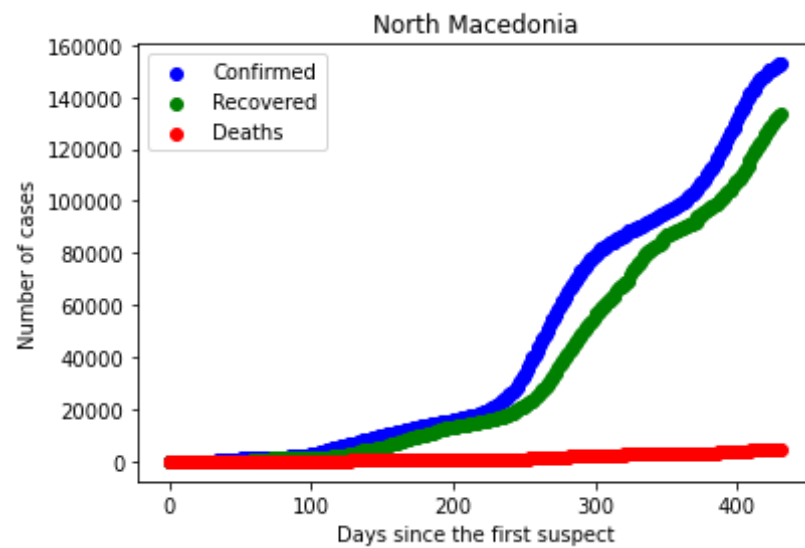
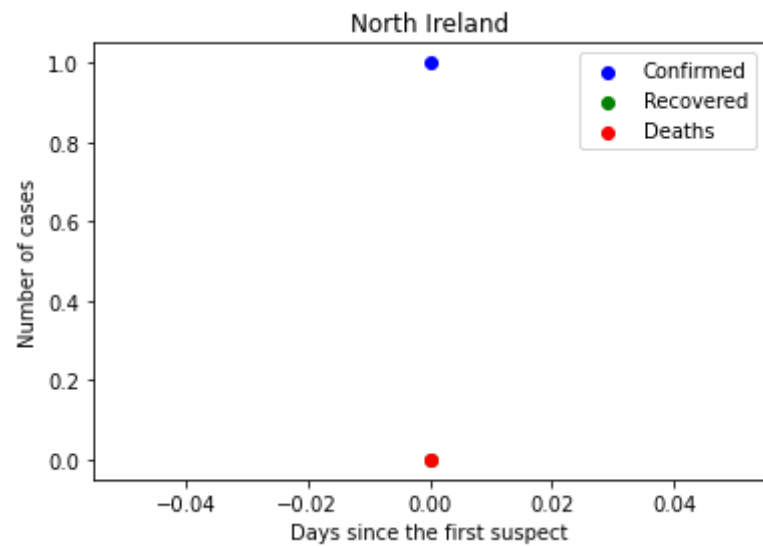


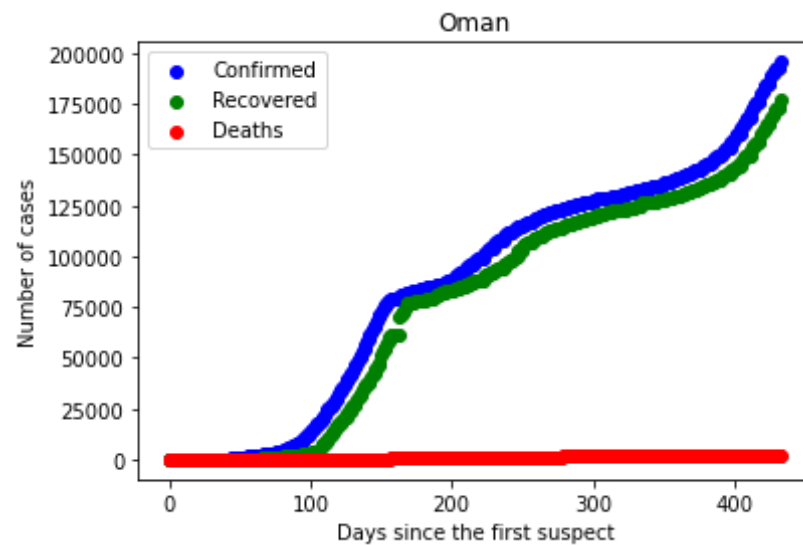
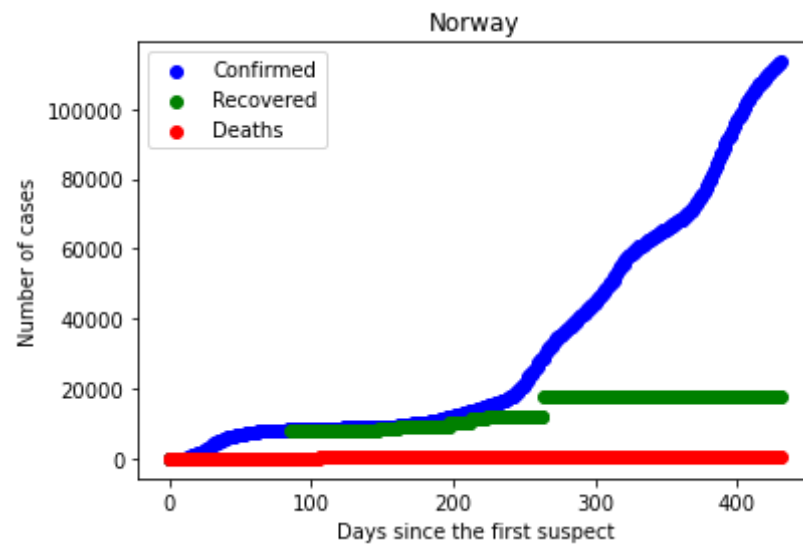


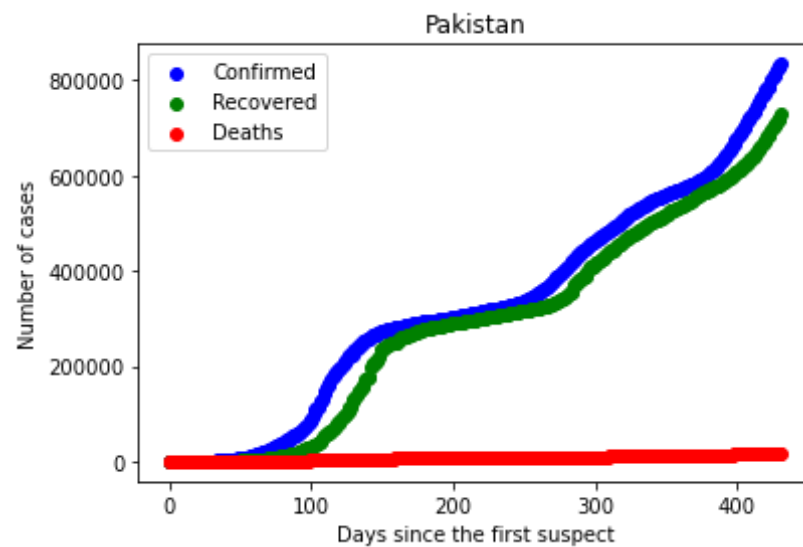
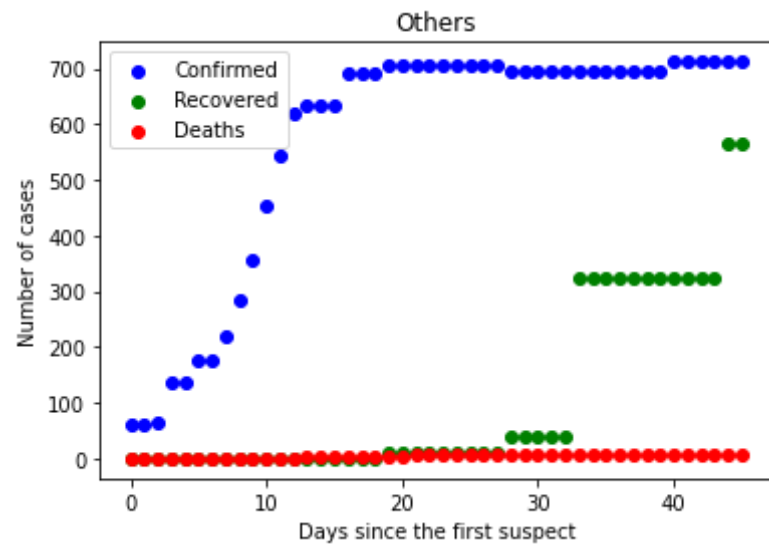


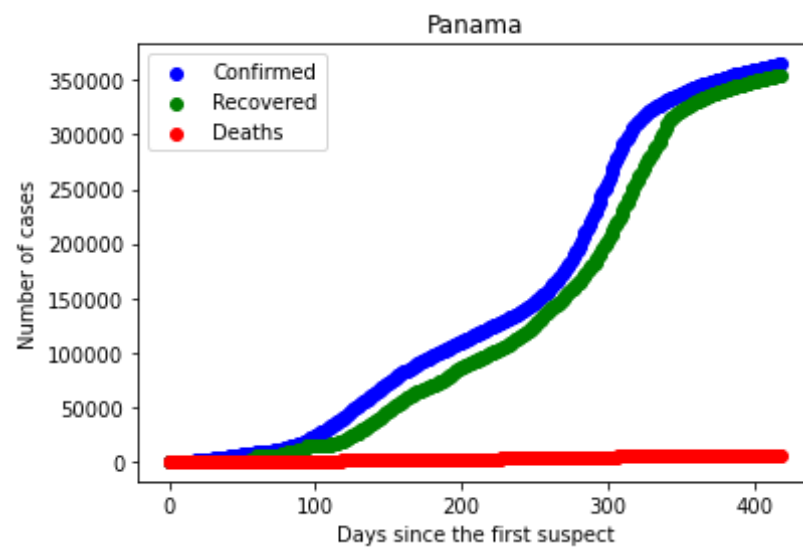
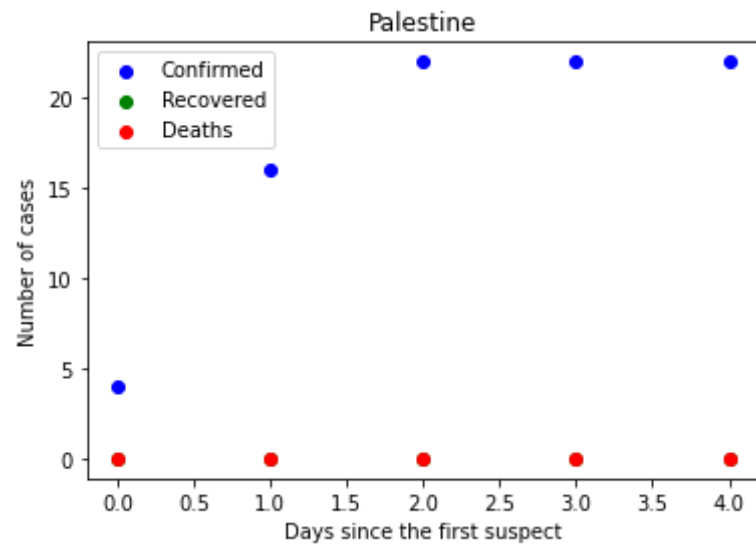


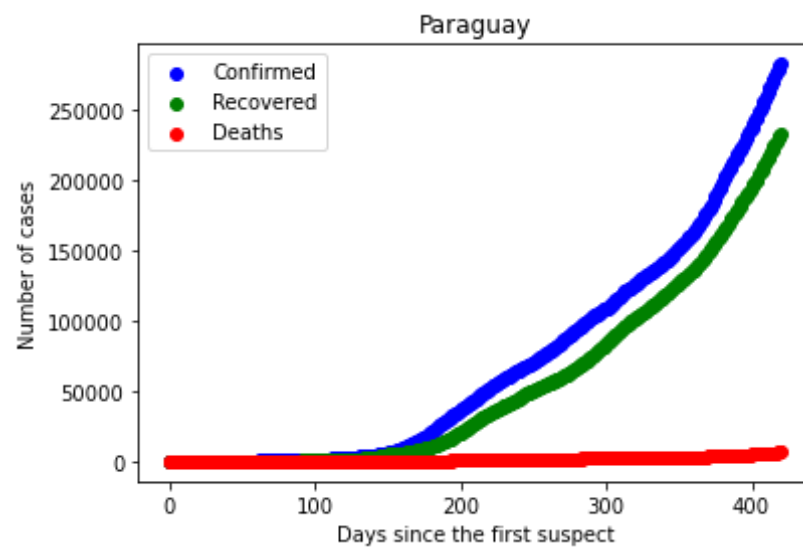
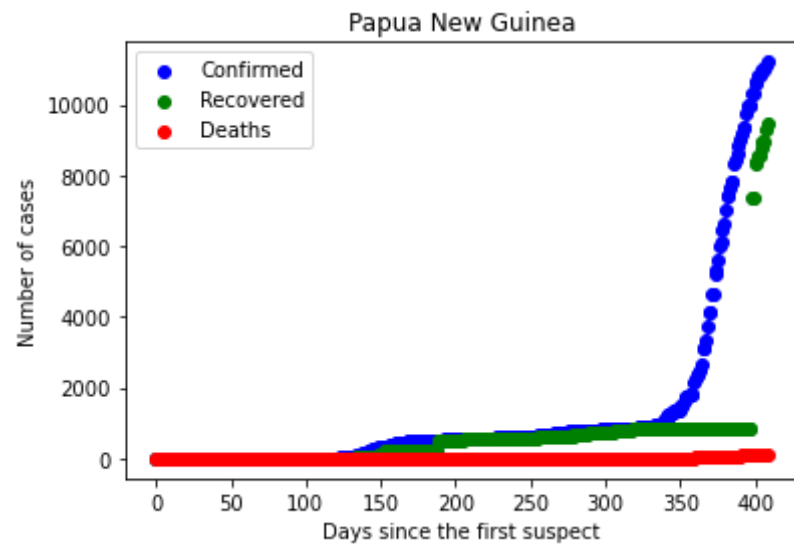


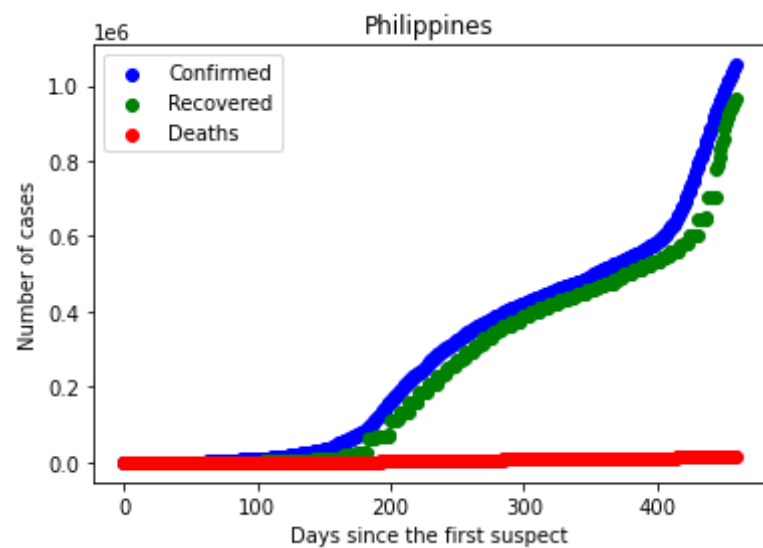
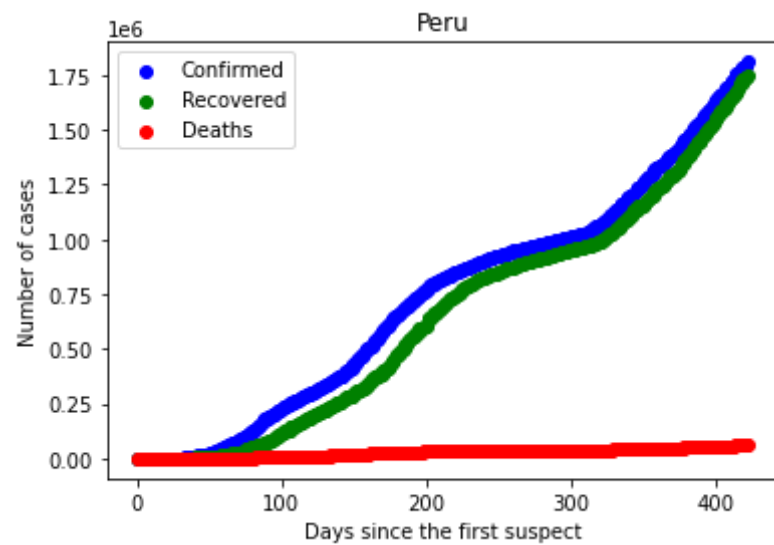


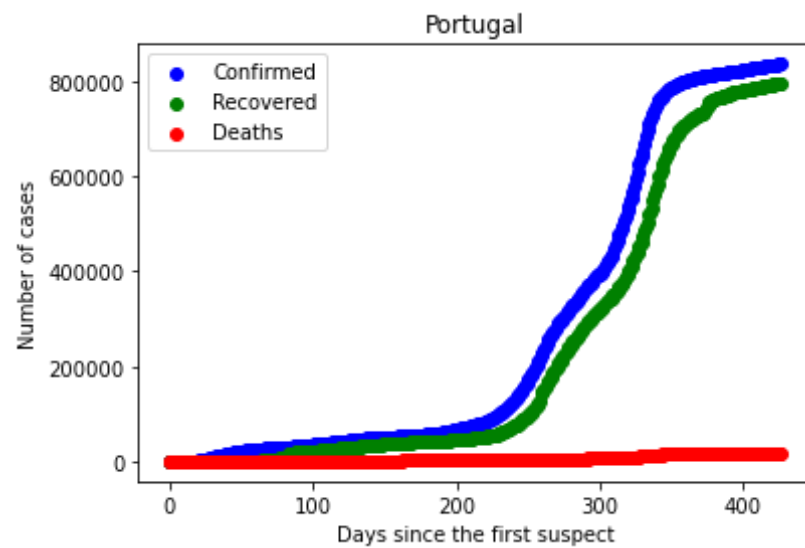
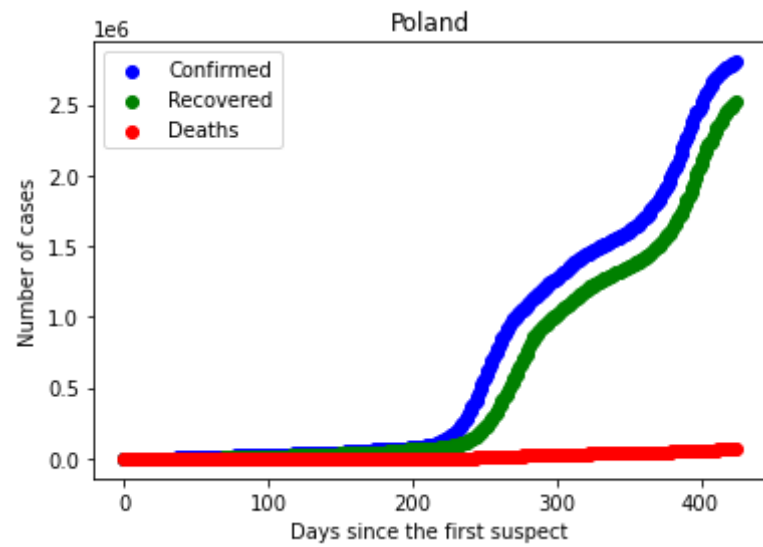


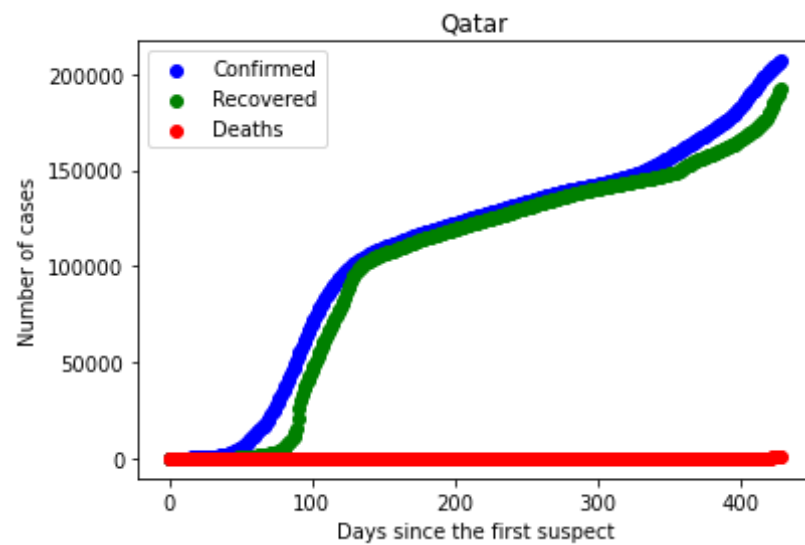
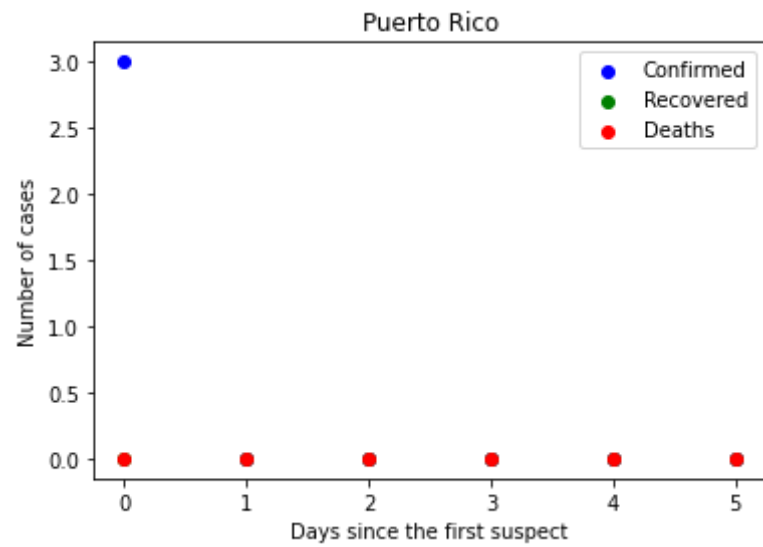


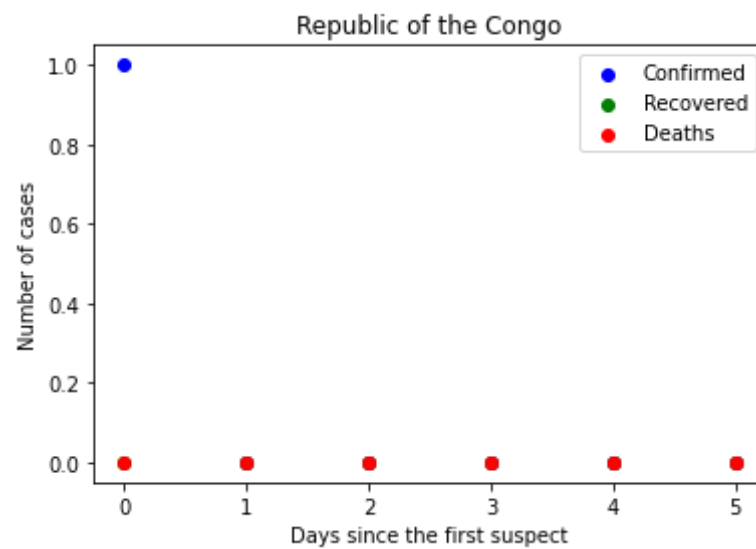
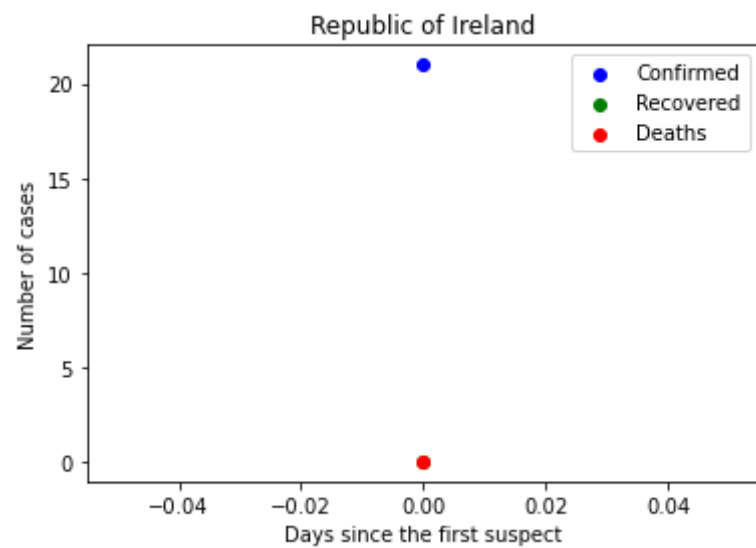


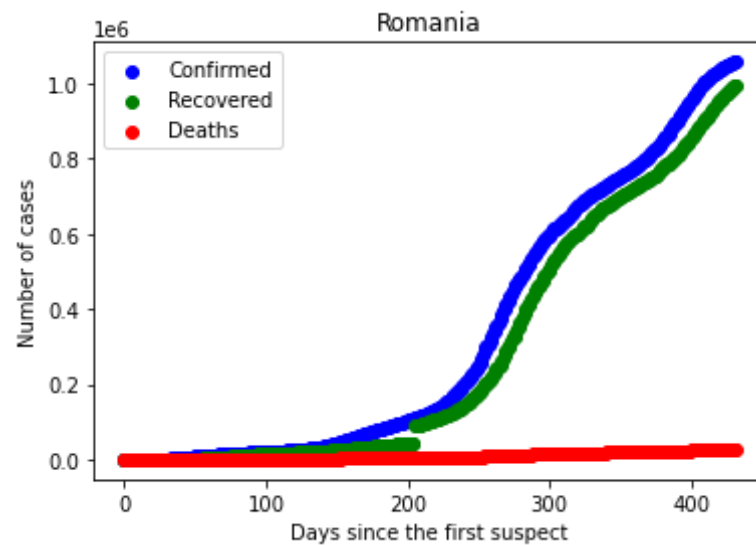
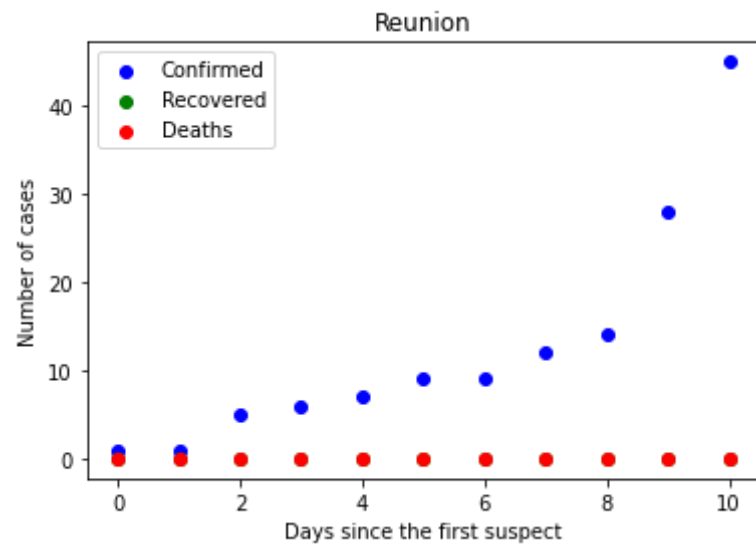


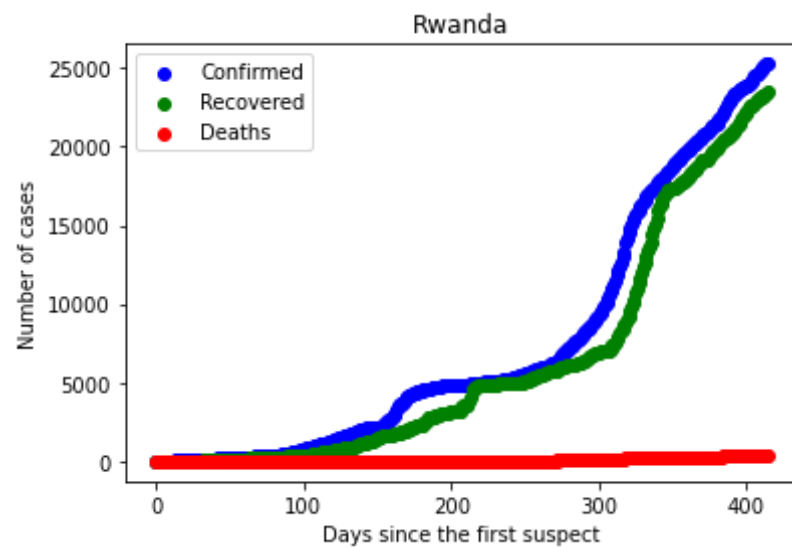
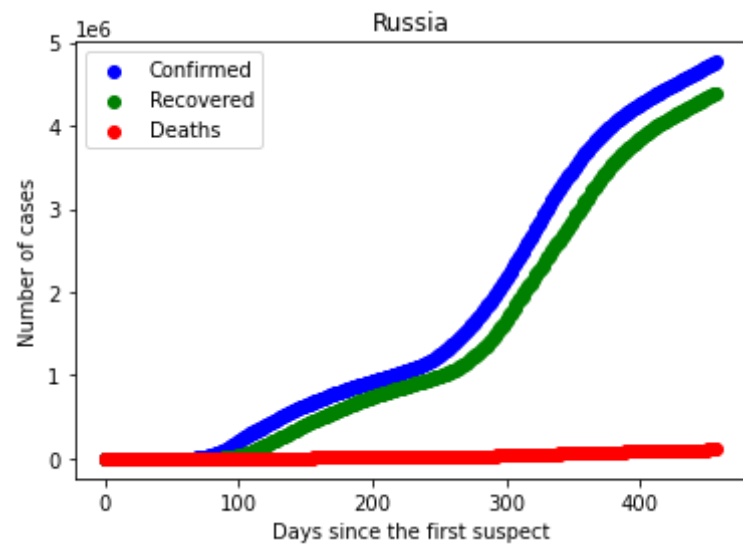


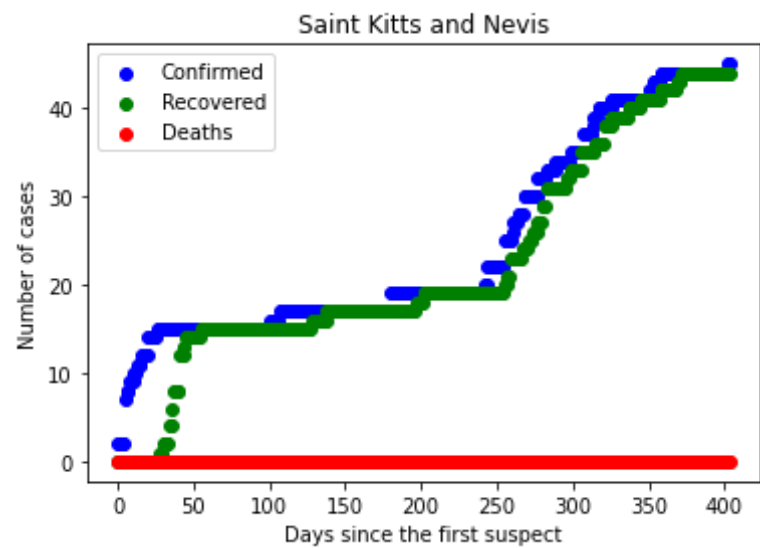
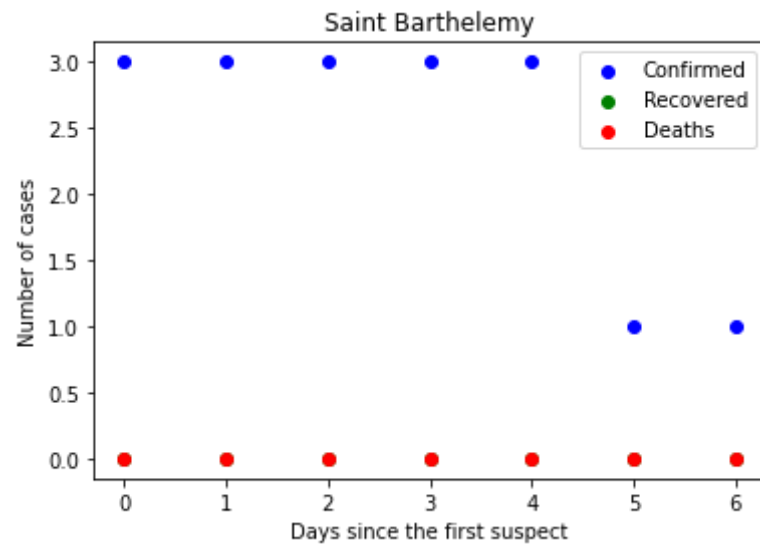


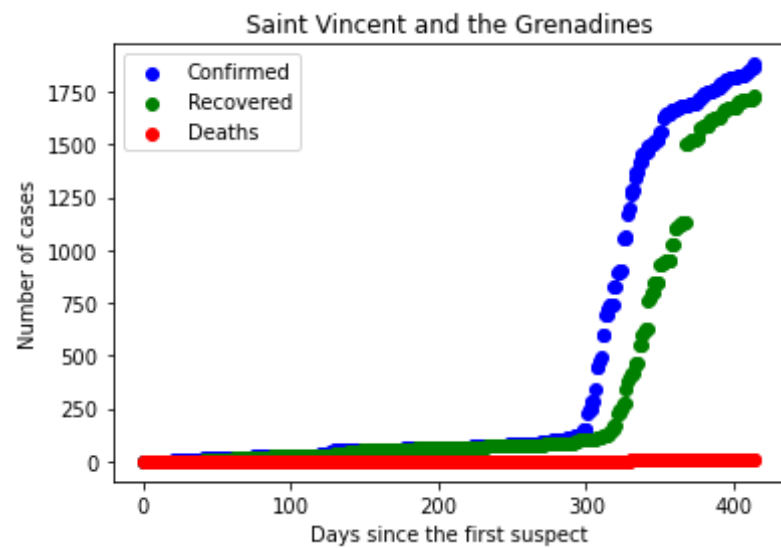
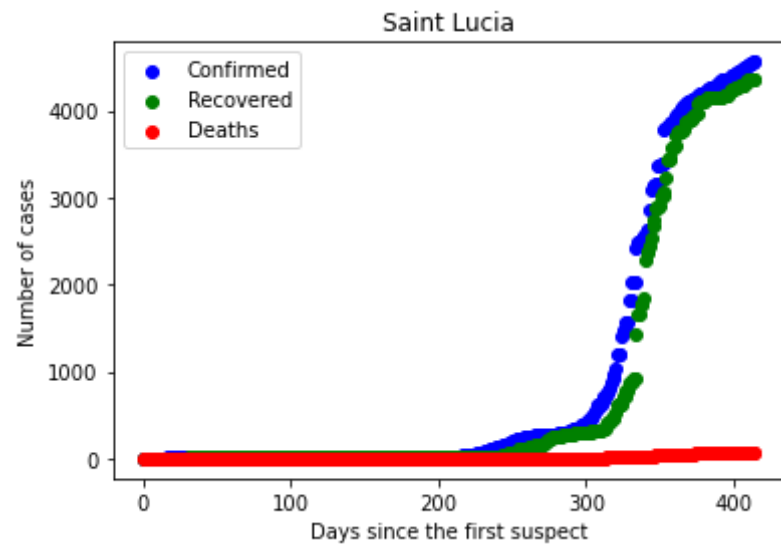


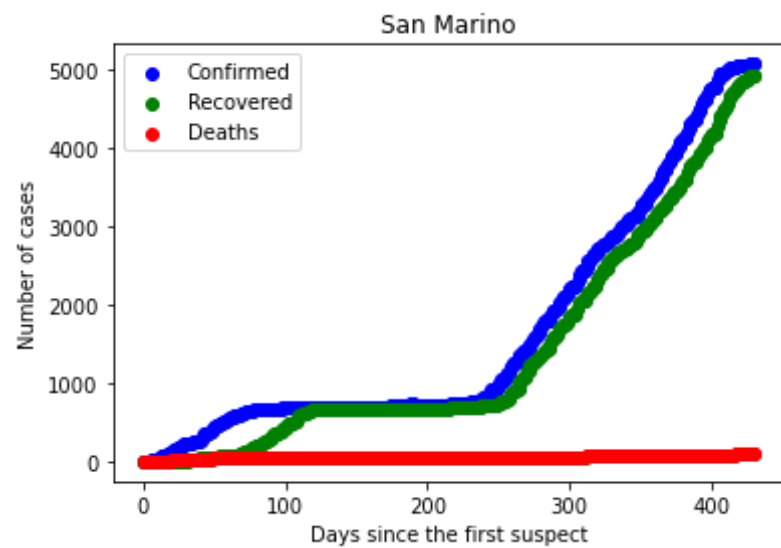
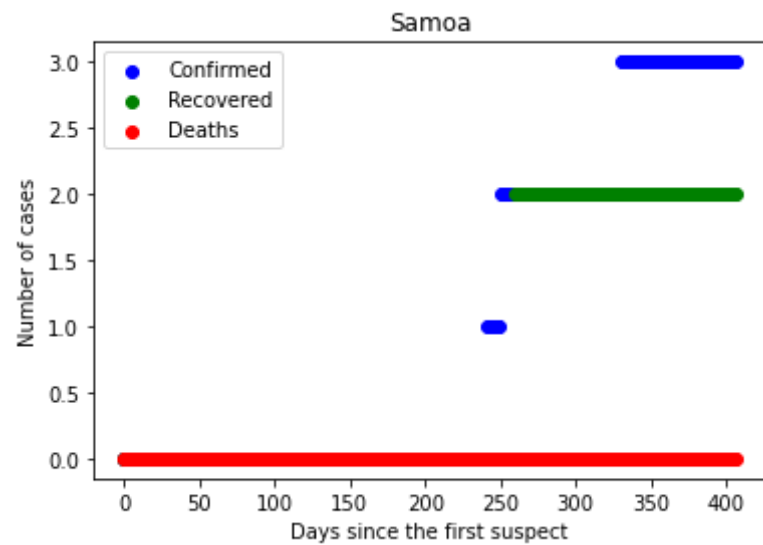


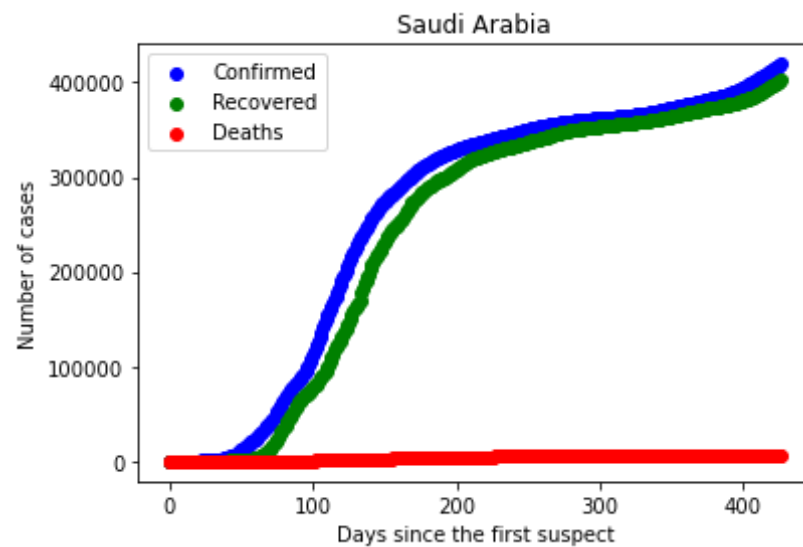
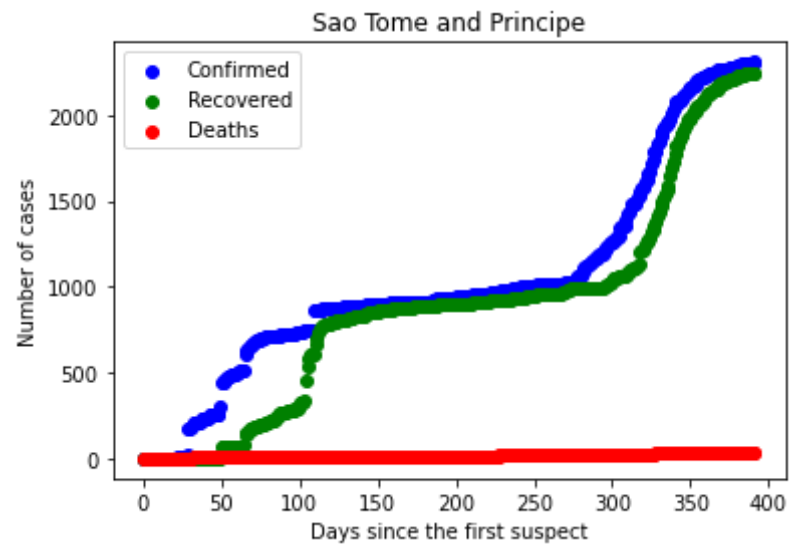


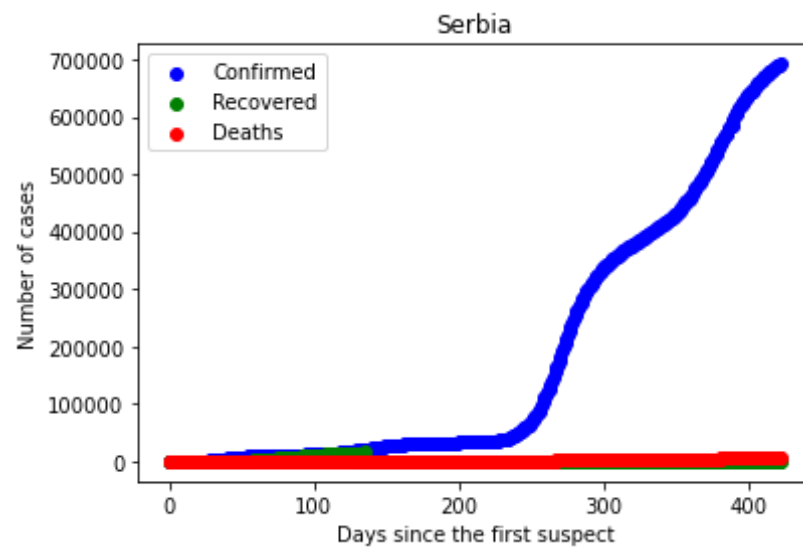
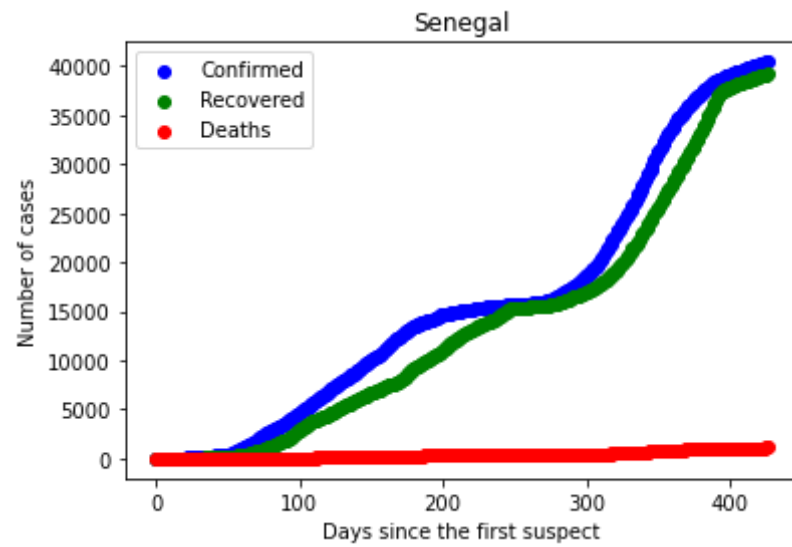


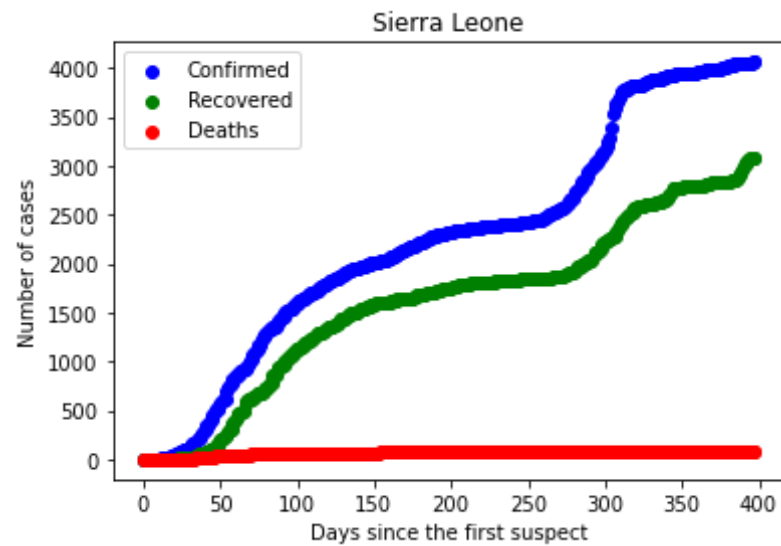
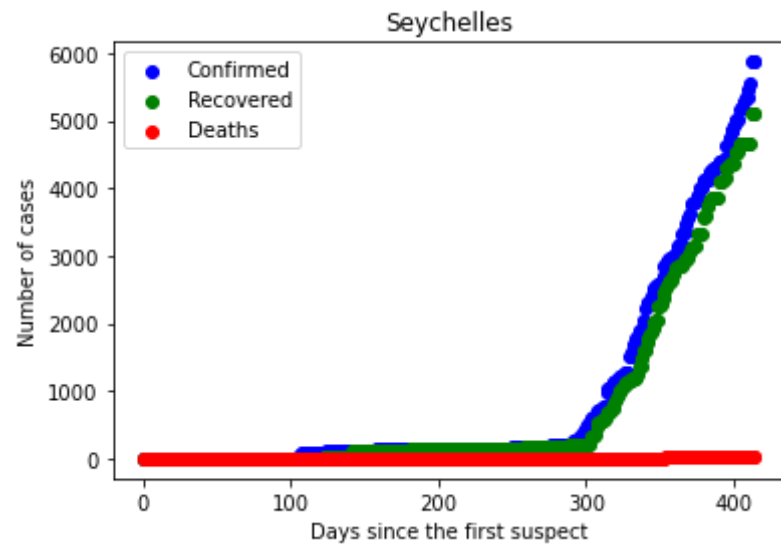


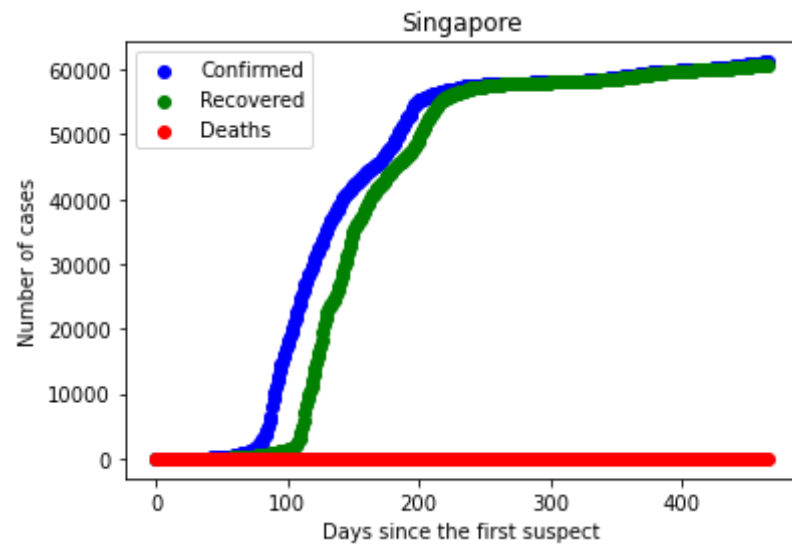


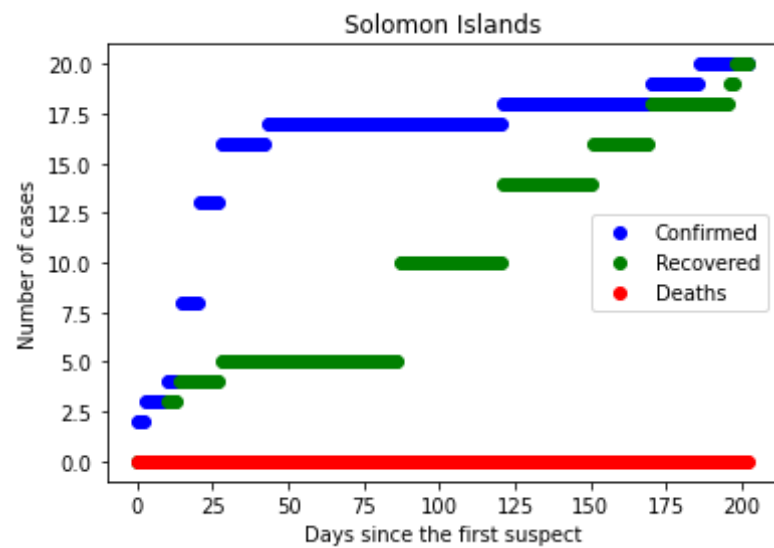
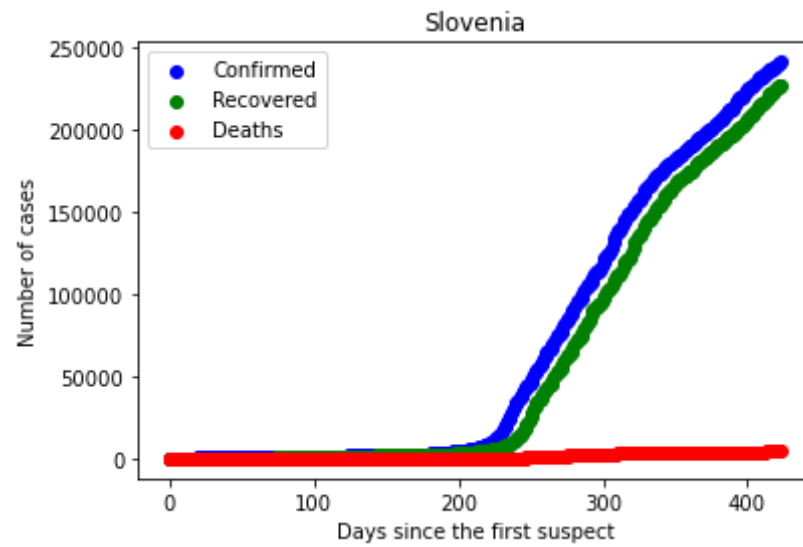


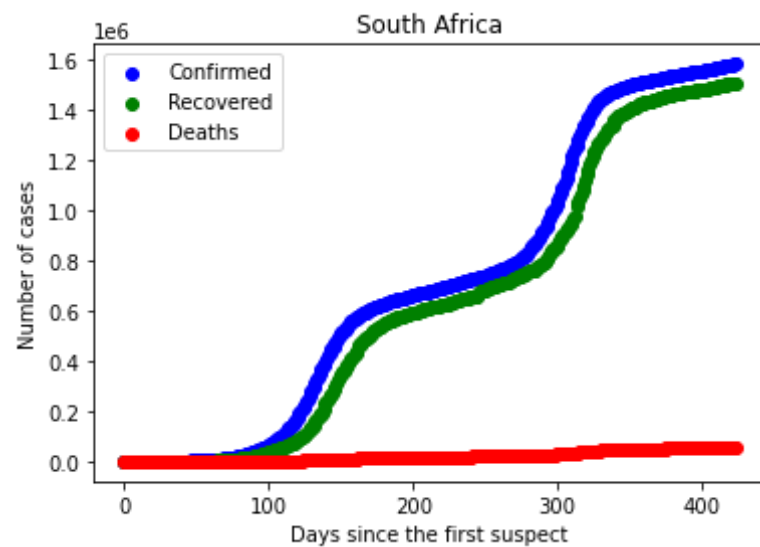
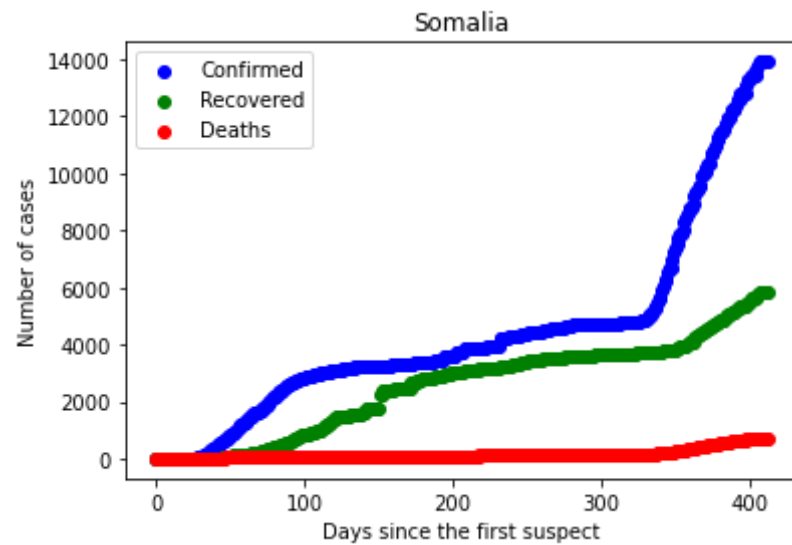


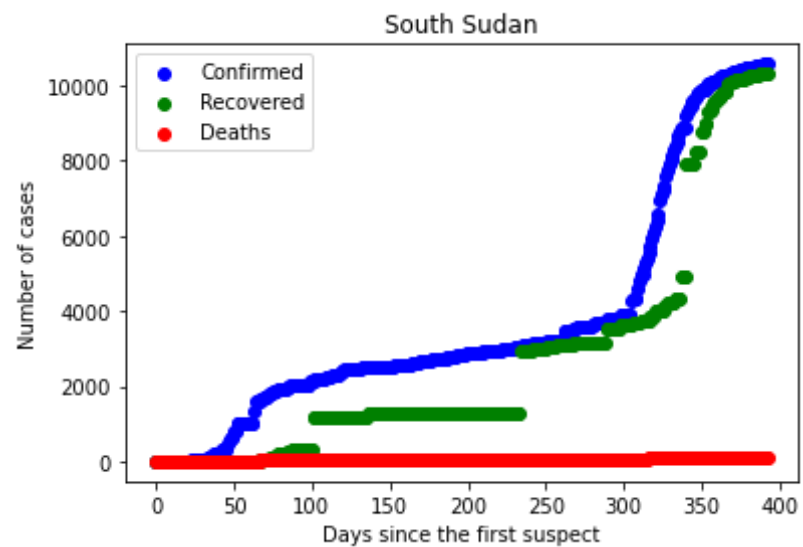
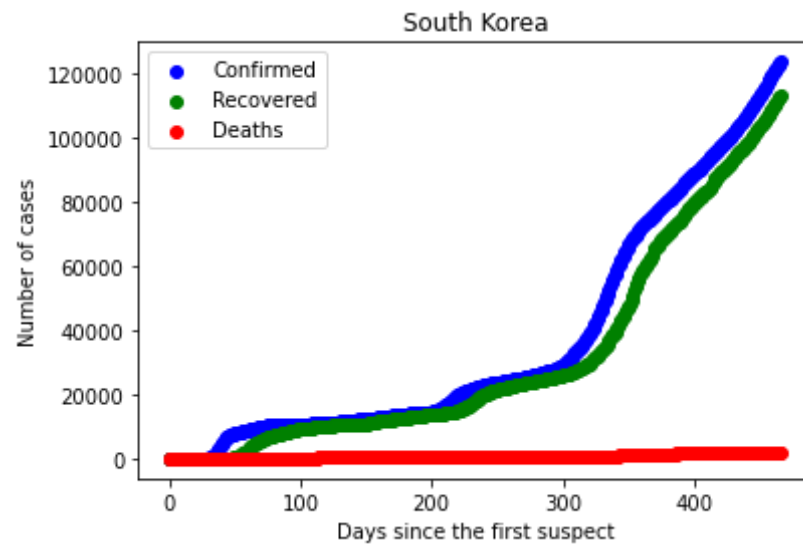


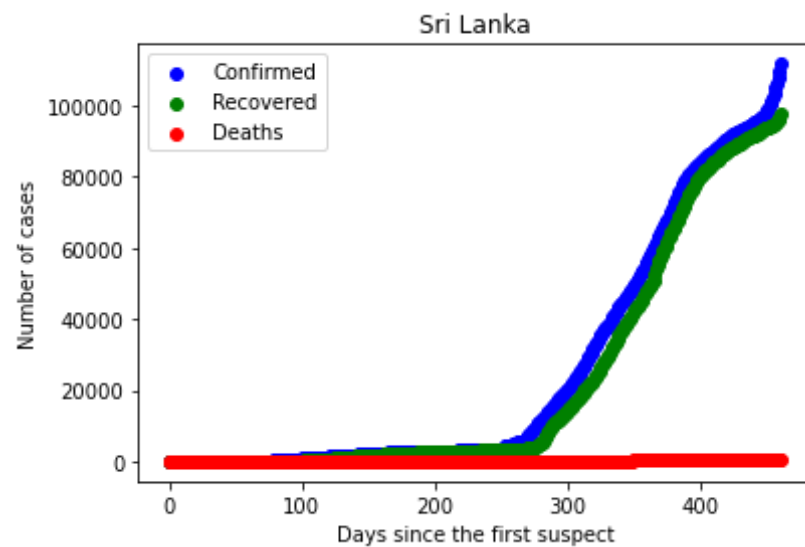
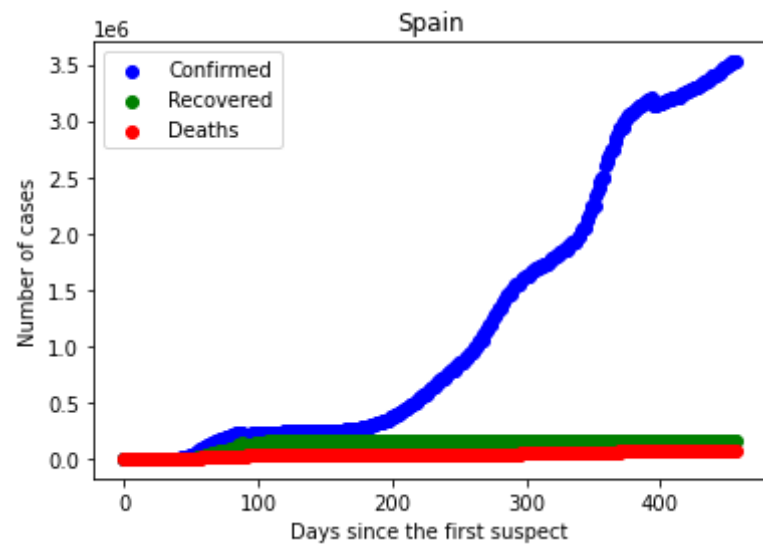


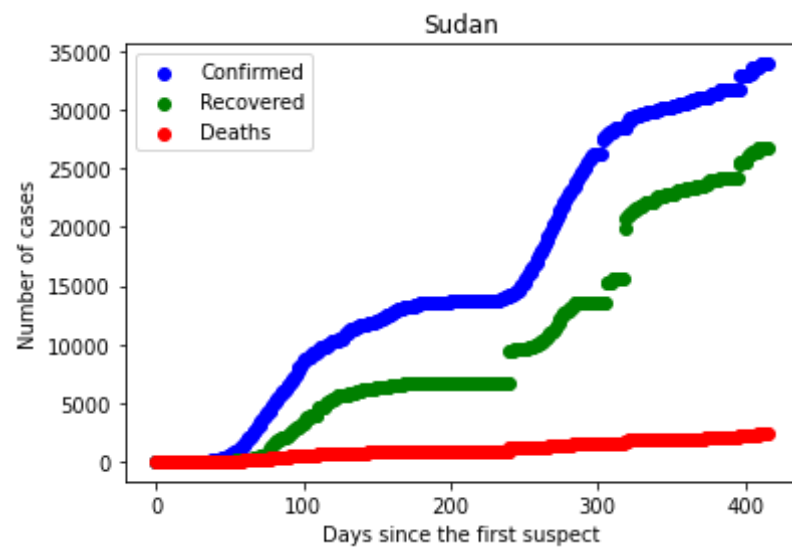
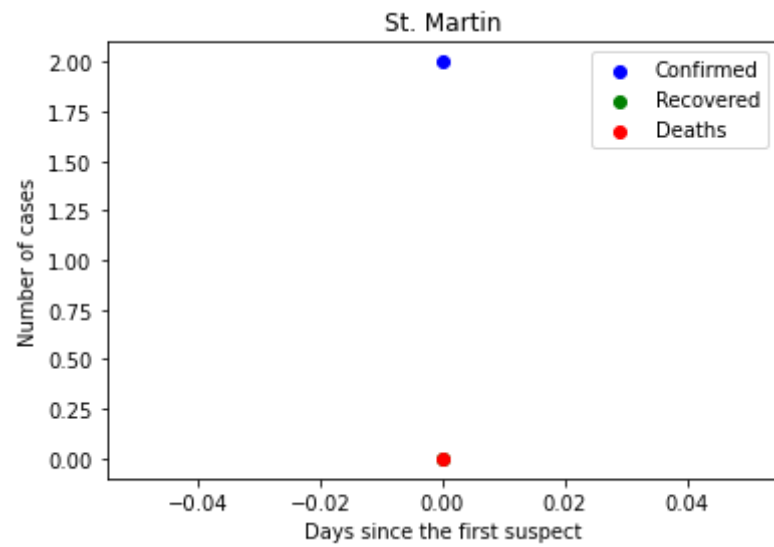


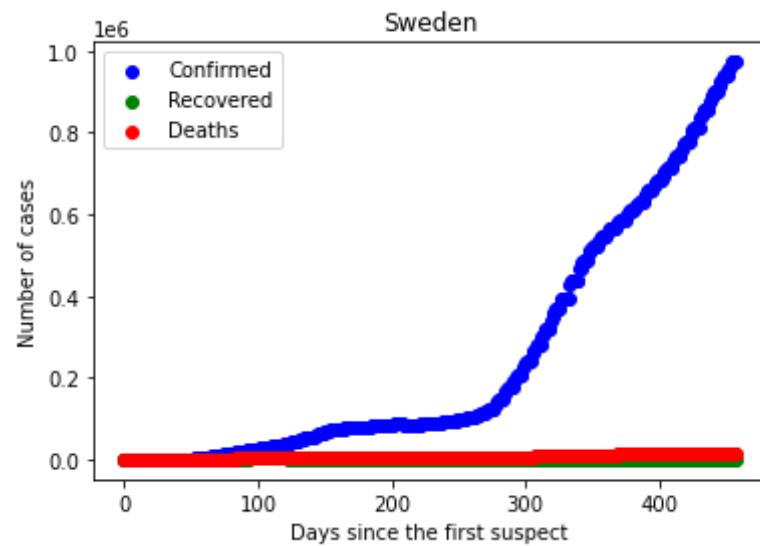
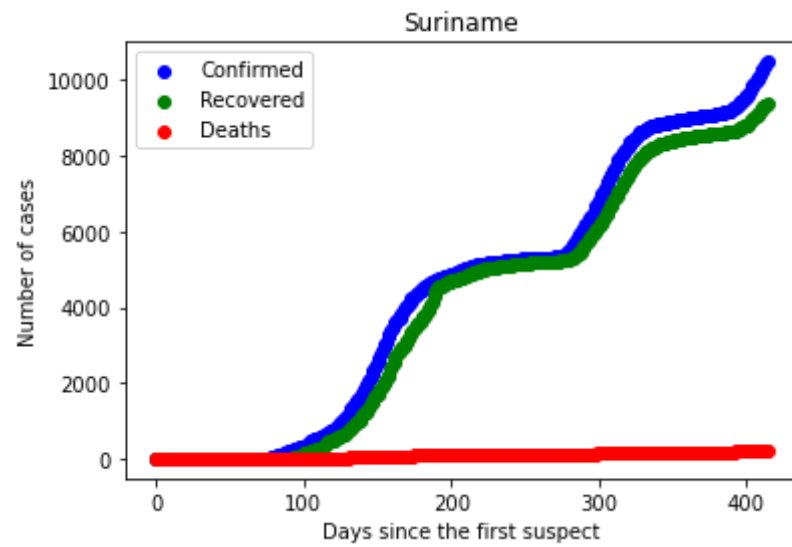


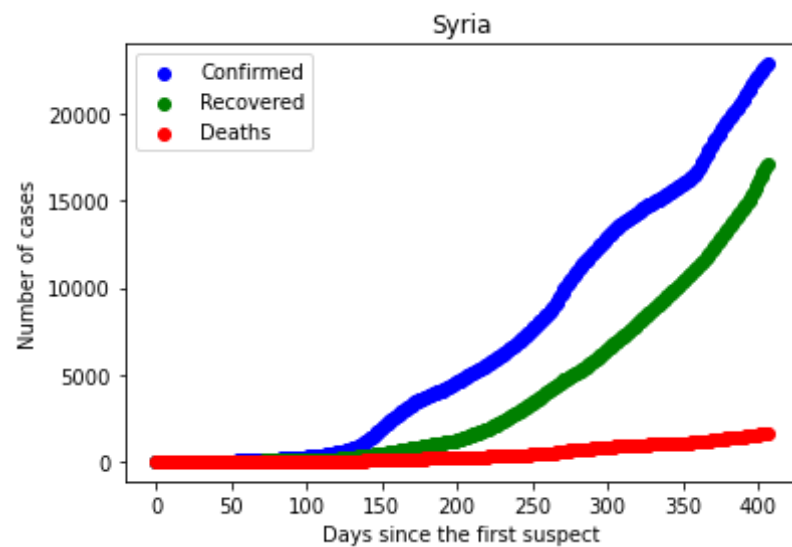
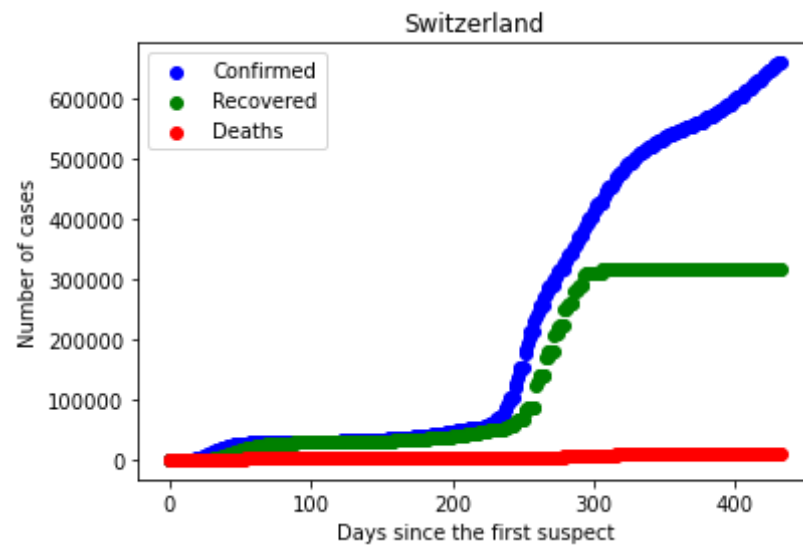


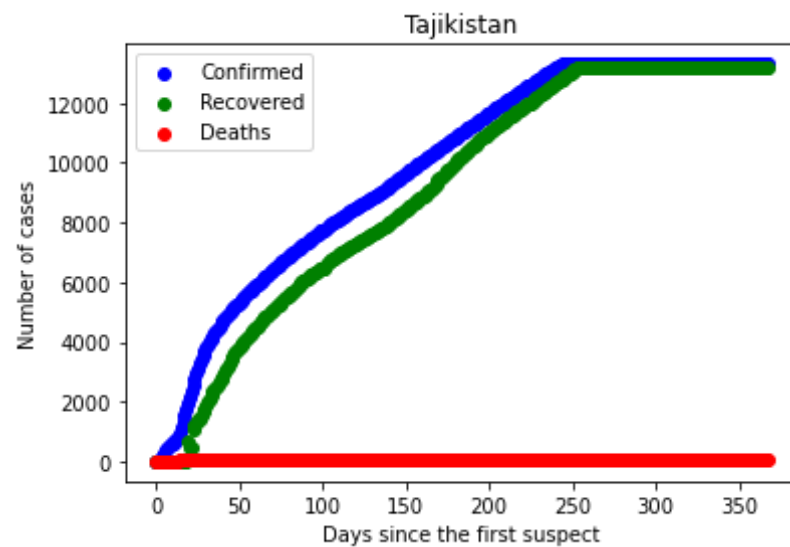
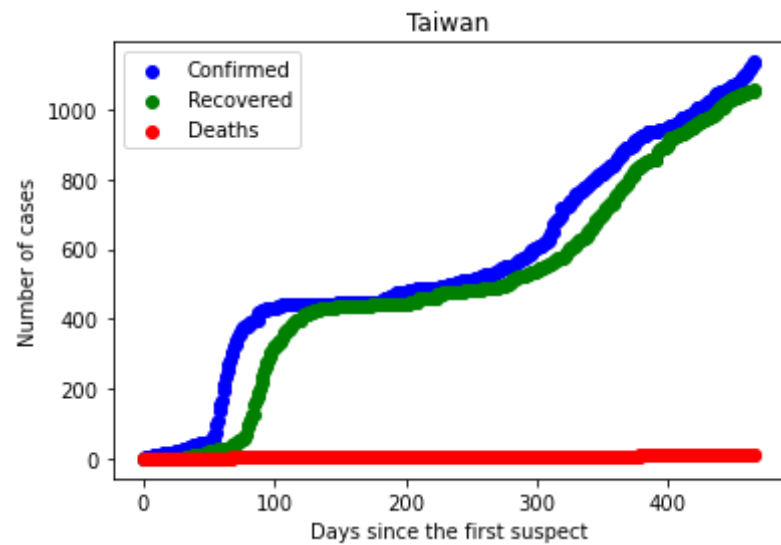


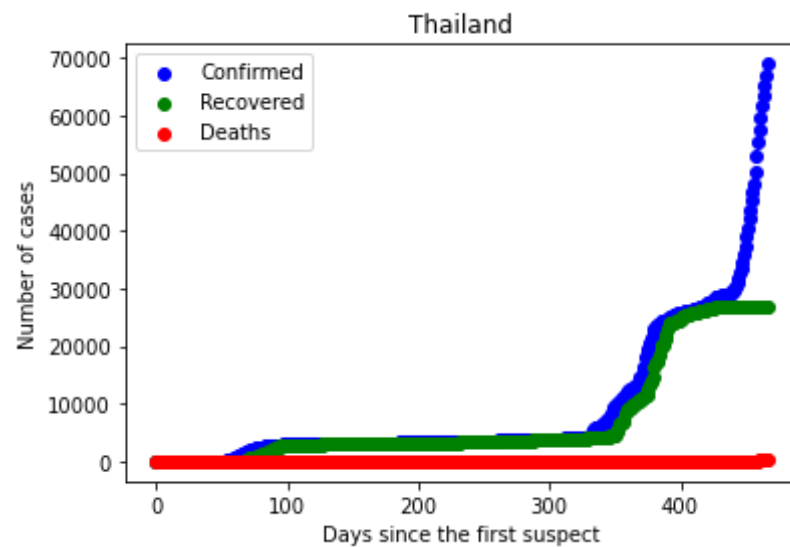
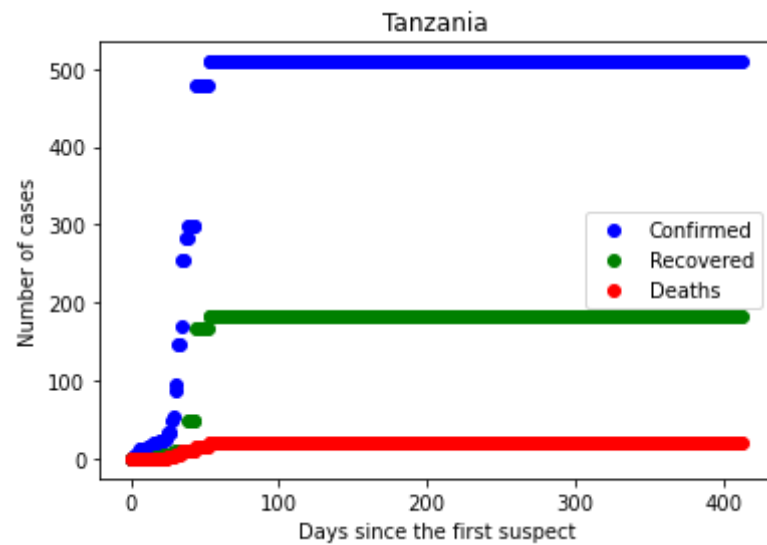


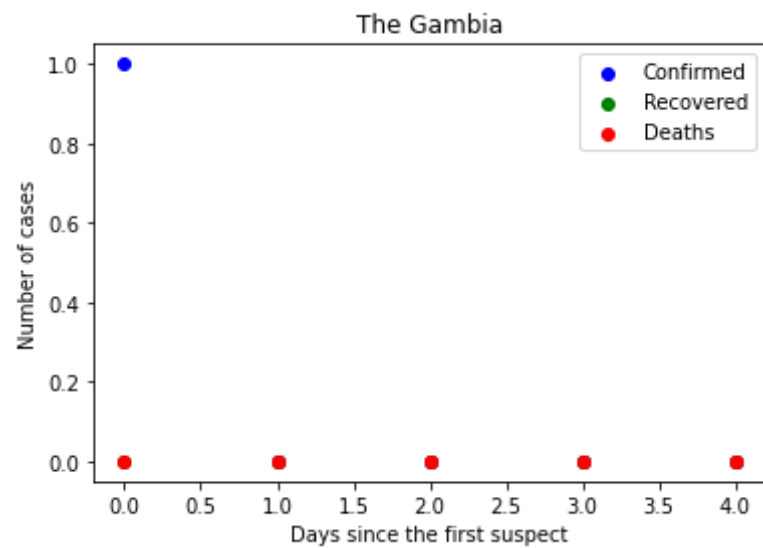
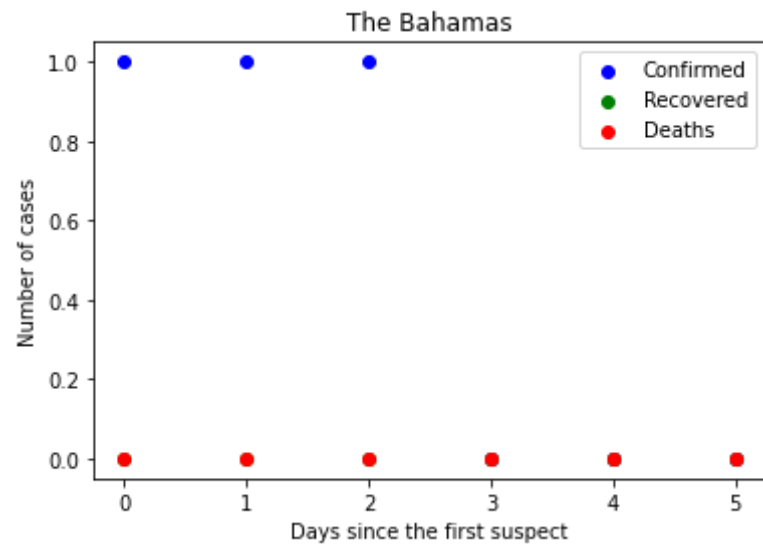


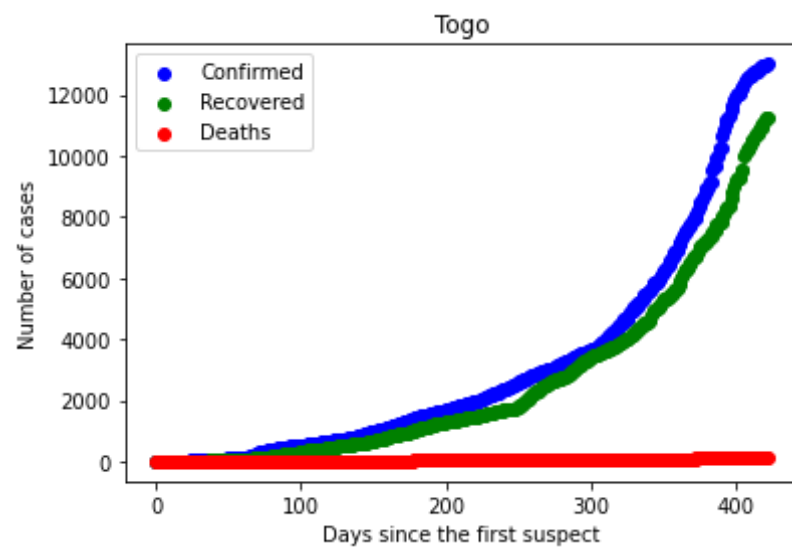
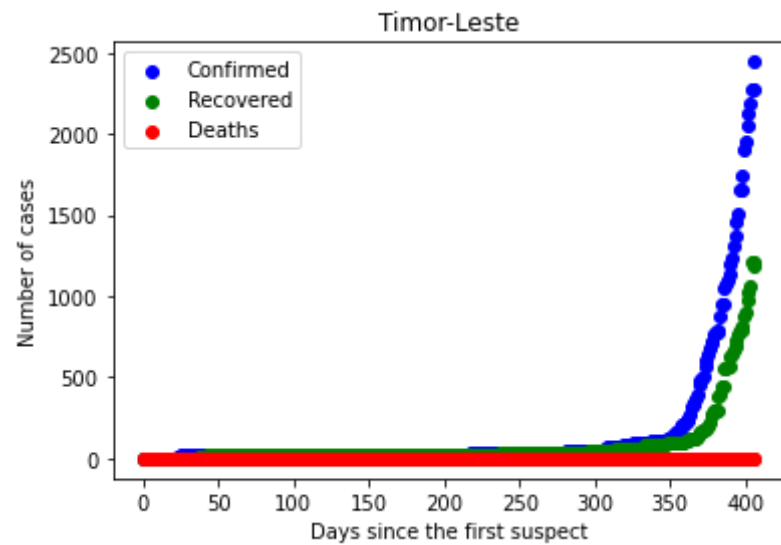


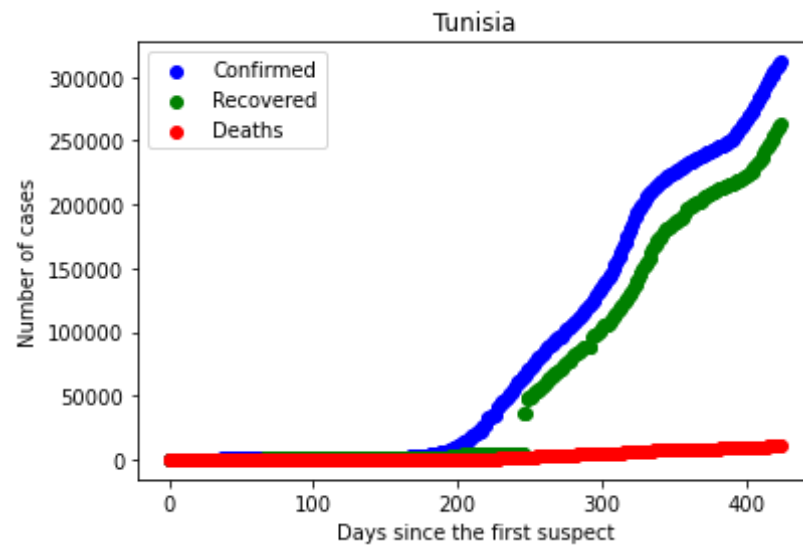
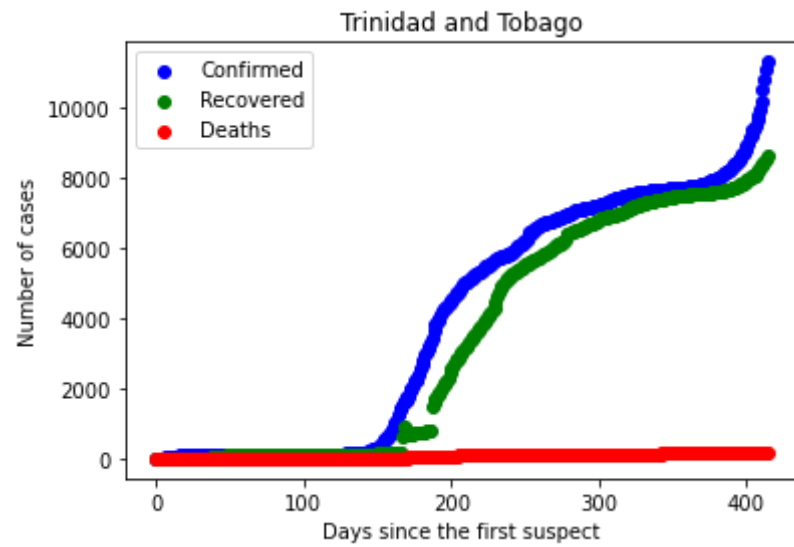


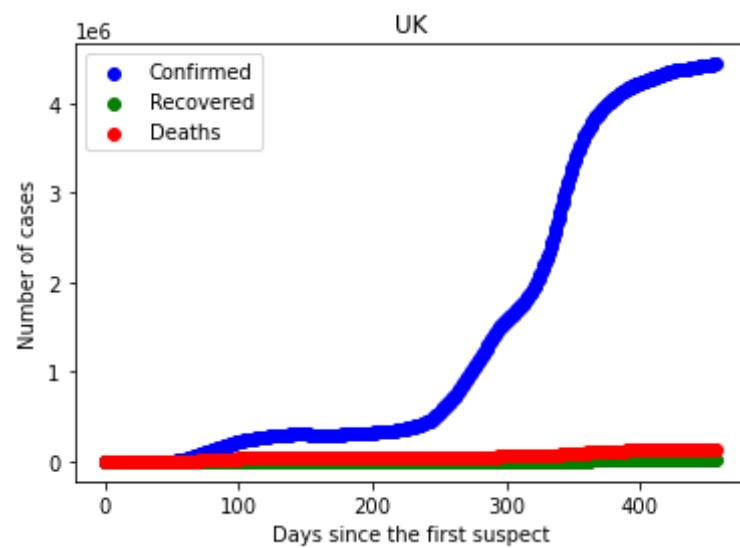
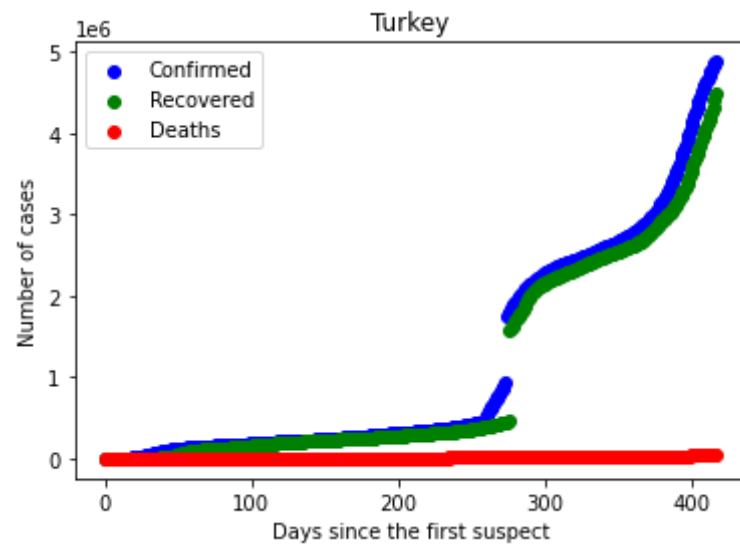


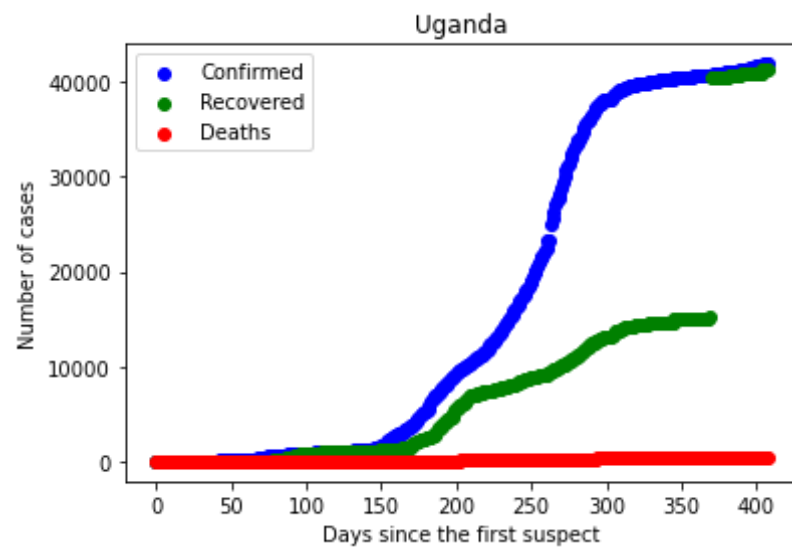
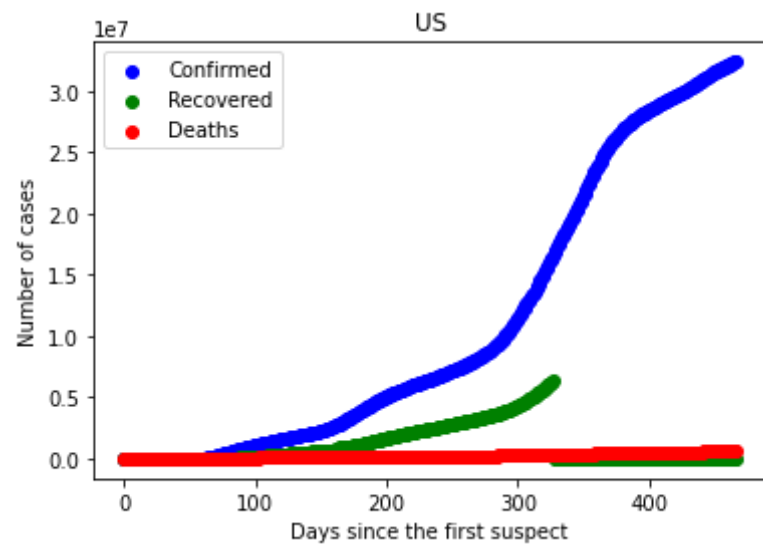


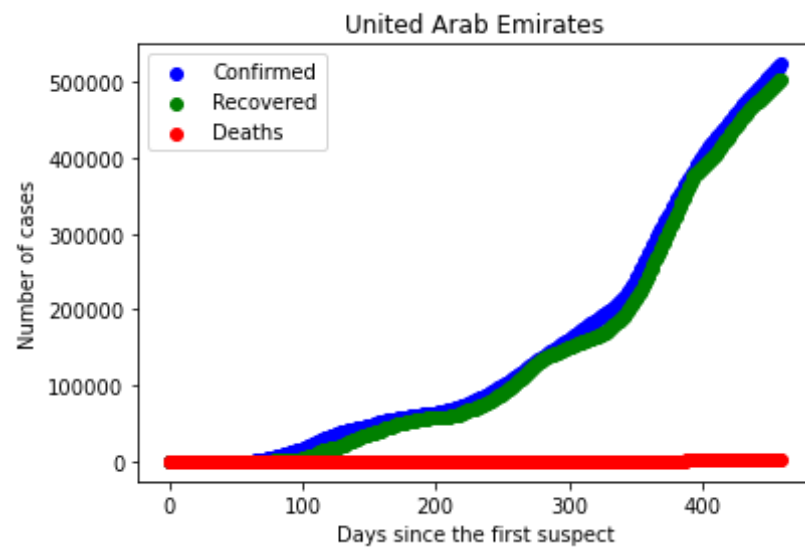
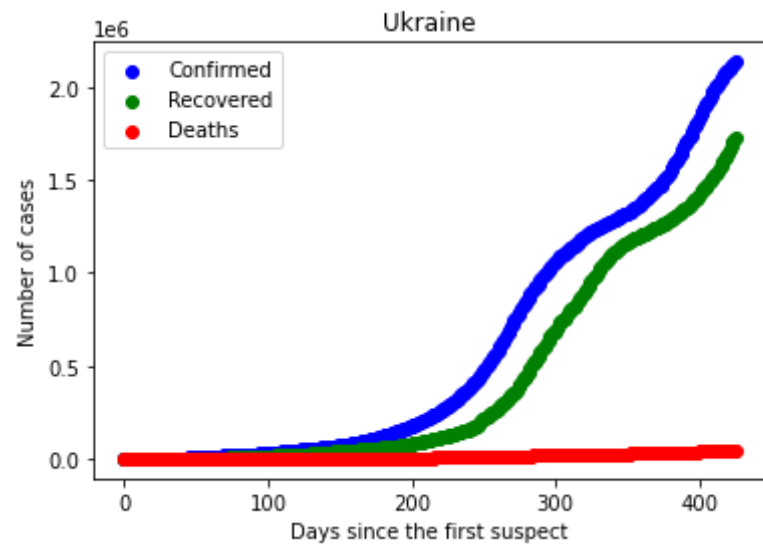


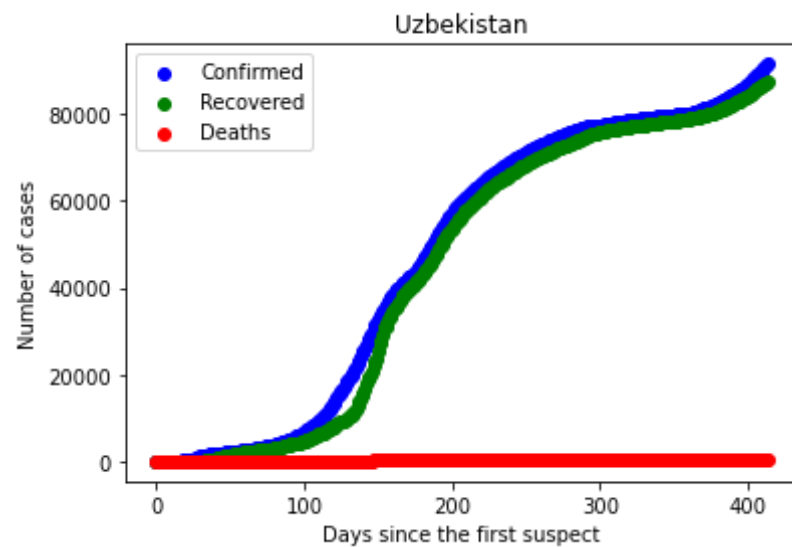
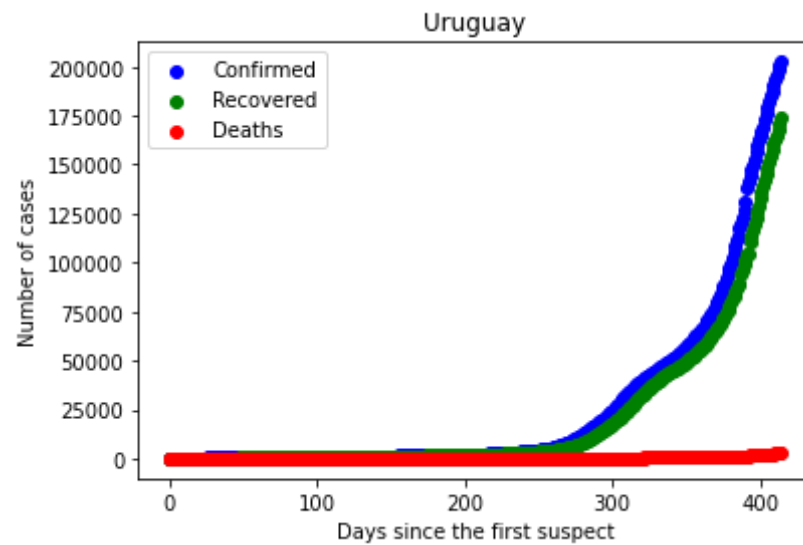


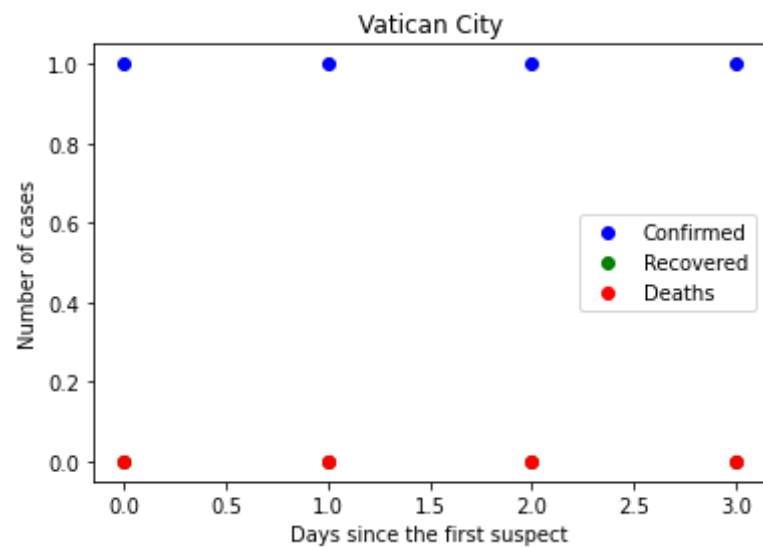
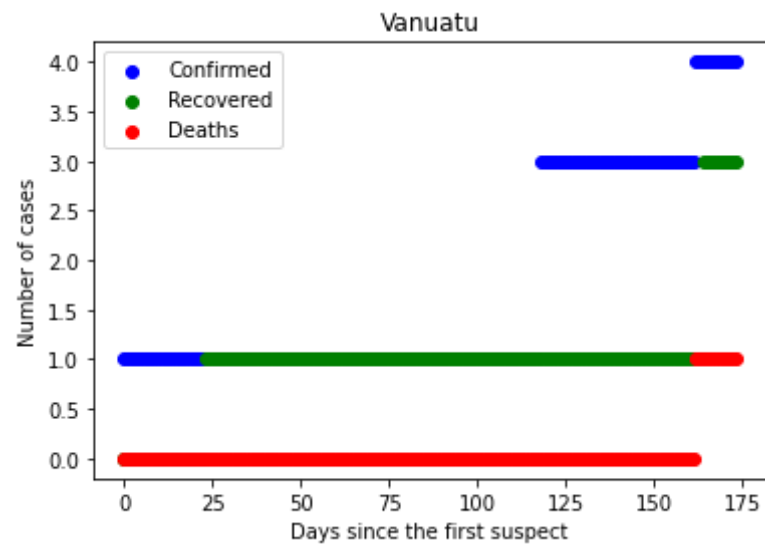


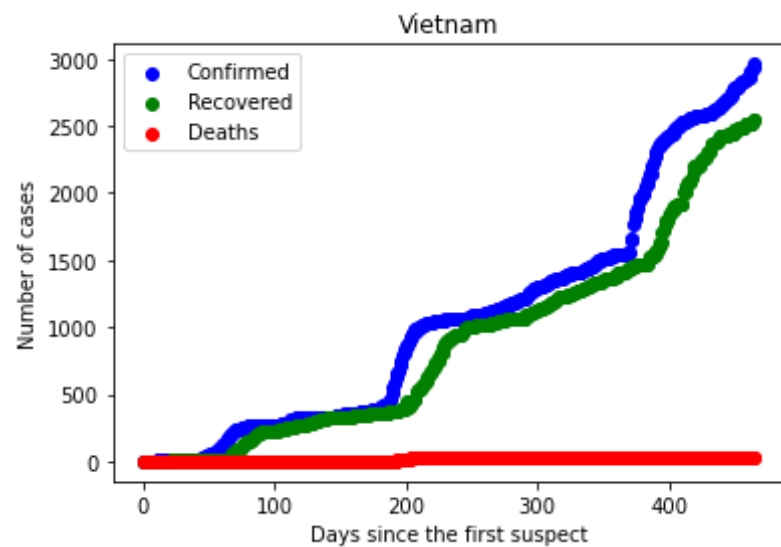
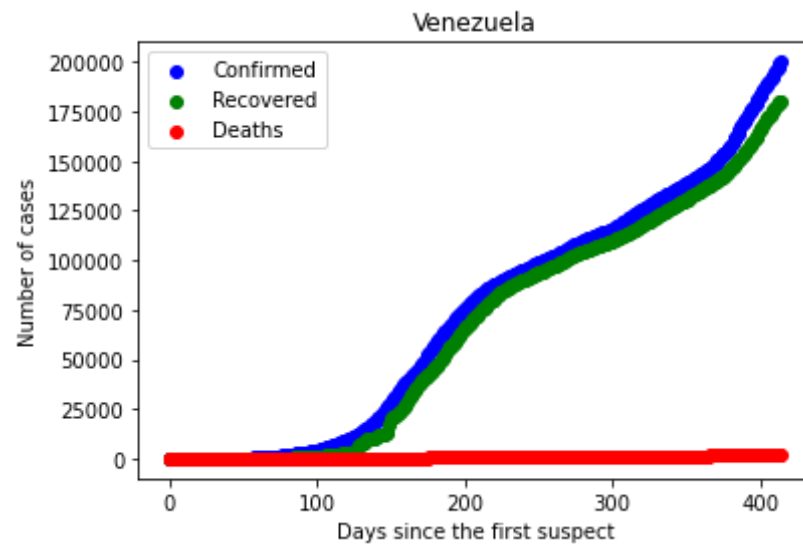


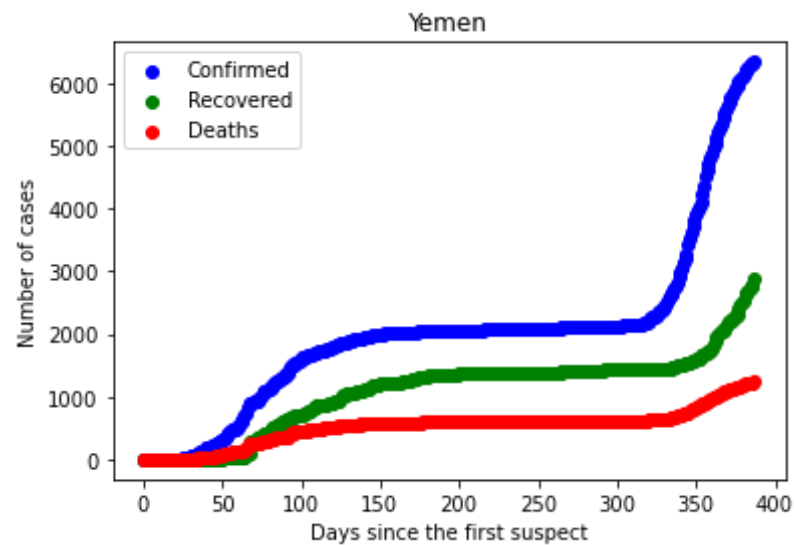
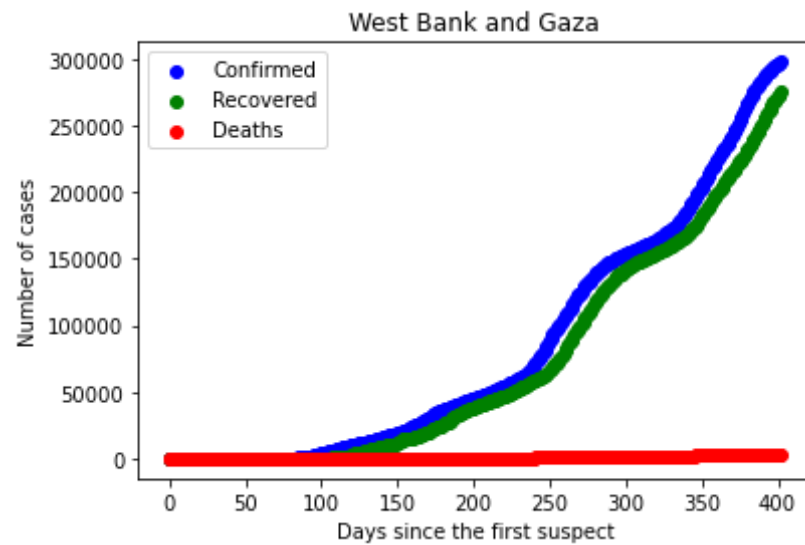


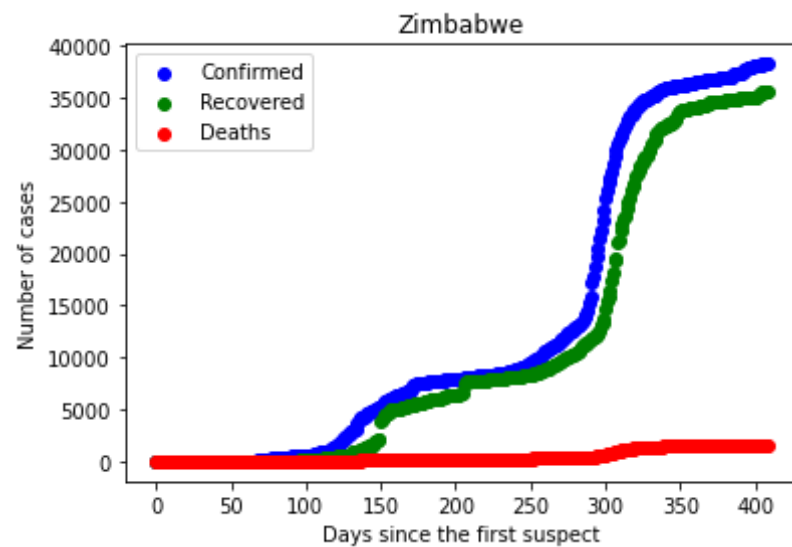
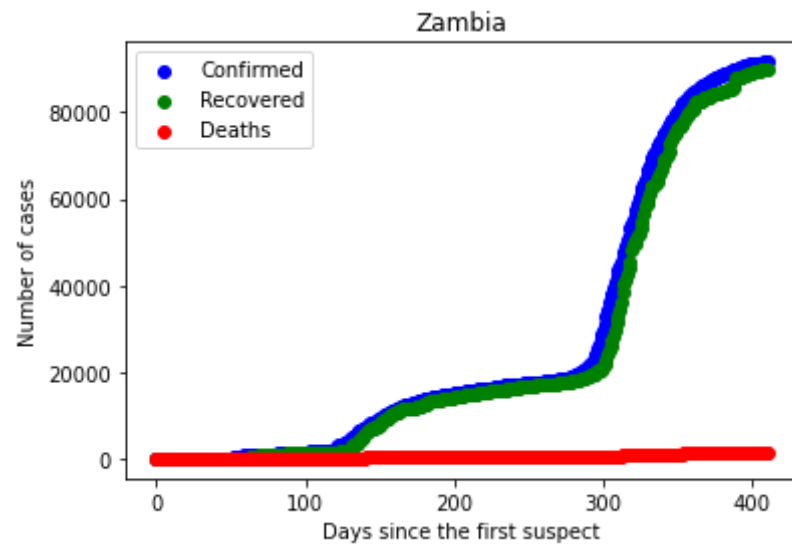


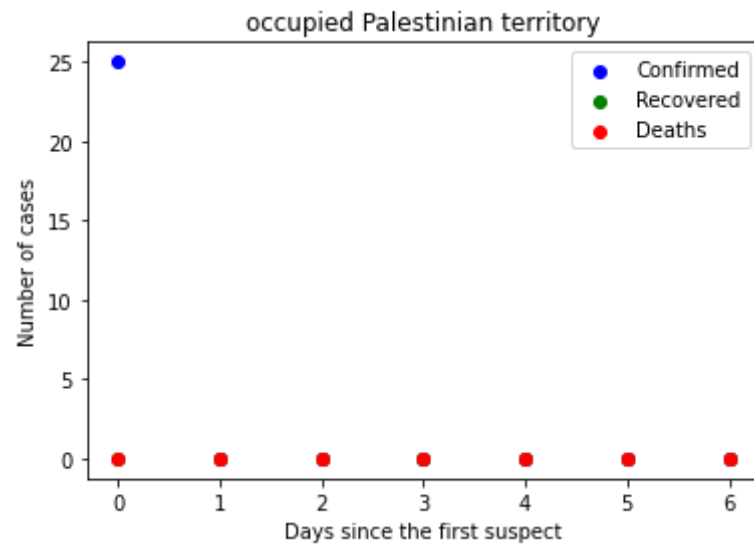












```
In [36]: df4=df2.groupby(['Date'])[['Date', 'Confirmed', 'Deaths', 'Recovered']].sum().reset_index()
```

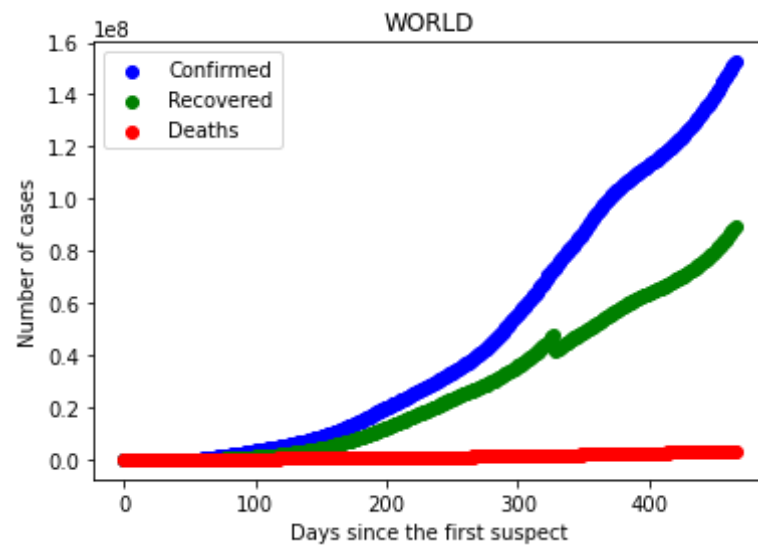
```
In [37]: df4.head(5)
```

```
Out[37]:
```

	Date	Confirmed	Deaths	Recovered
0	2020-01-22	557.0	17.0	30.0
1	2020-01-23	1097.0	34.0	60.0
2	2020-01-24	941.0	26.0	39.0
3	2020-01-25	1437.0	42.0	42.0
4	2020-01-26	2118.0	56.0	56.0

```
In [39]: C=df4
plt.scatter(np.arange(0,len(C)),C['Confirmed'],color='blue',label='Confirmed')
plt.scatter(np.arange(0,len(C)),C['Recovered'],color='green',label='Recovered')
plt.scatter(np.arange(0,len(C)),C['Deaths'],color='red',label='Deaths')
plt.title('WORLD')
plt.xlabel('Days since the first suspect')
plt.ylabel('Number of cases')
```

```
plt.legend()  
plt.show()
```



In []: