

```
In [ ]: import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import matplotlib.colors as mcolors
import random
import math
import time
from sklearn.linear_model import LinearRegression
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import PolynomialFeatures
from sklearn.svm import SVR
from sklearn.metrics import mean_squared_error, mean_absolute_error
import datetime
import operator
plt.style.use('fivethirtyeight')
%matplotlib inline
```

```
In [ ]: confirmed_cases = pd.read_csv('https://raw.githubusercontent.com/CSSEGISandData/COV
```

```
In [ ]: confirmed_cases.head()
```

Out[ ]:	Province/State	Country/Region	Lat	Long	1/22/20	1/23/20	1/24/20	1/25/20	1/26/20
0	NaN	Afghanistan	33.93911	67.709953	0	0	0	0	0
1	NaN	Albania	41.15330	20.168300	0	0	0	0	0
2	NaN	Algeria	28.03390	1.659600	0	0	0	0	0
3	NaN	Andorra	42.50630	1.521800	0	0	0	0	0
4	NaN	Angola	-11.20270	17.873900	0	0	0	0	0

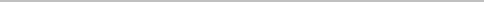
5 rows × 180 columns

◀ ▶

```
In [ ]: deaths_reported = pd.m
```

Out[ ]:	Province/State	Country/Region	Lat	Long	1/22/20	1/23/20	1/24/20	1/25/20	1/26/20
0	NaN	Afghanistan	33.93911	67.709953	0	0	0	0	0
1	NaN	Albania	41.15330	20.168300	0	0	0	0	0
2	NaN	Algeria	28.03390	1.659600	0	0	0	0	0
3	NaN	Andorra	42.50630	1.521800	0	0	0	0	0
4	NaN	Angola	-11.20270	17.873900	0	0	0	0	0

5 rows x 180 columns

```
In [ ]: recovered_cases = pd.m
```

Out[ ]:

	Province/State	Country/Region	Lat	Long	1/22/20	1/23/20	1/24/20	1/25/20	1/26/20
0	NaN	Afghanistan	33.93911	67.709953	0	0	0	0	0
1	NaN	Albania	41.15330	20.168300	0	0	0	0	0
2	NaN	Algeria	28.03390	1.659600	0	0	0	0	0
3	NaN	Andorra	42.50630	1.521800	0	0	0	0	0
4	NaN	Angola	-11.20270	17.873900	0	0	0	0	0

5 rows × 180 columns



In [ ]: latest\_data = pd.read\_csv('https://raw.githubusercontent.com/CSSEGISandData/COVID-19/master/csse\_covid\_19\_data/csse\_covid\_19\_time\_series/time\_series\_covid\_19\_confirmed.csv')

In [ ]: latest\_data.head()

Out[ ]:

	FIPS	Admin2	Province_State	Country_Region	Last_Update	Lat	Long_	Confirmed
0	45001.0	Abbeville	South Carolina		US 2020-07-16 04:44:59	34.223334	-82.461707	100
1	22001.0	Acadia	Louisiana		US 2020-07-16 04:44:59	30.295065	-92.414197	100
2	51001.0	Accomack	Virginia		US 2020-07-16 04:44:59	37.767072	-75.632346	100
3	16001.0	Ada	Idaho		US 2020-07-16 04:44:59	43.452658	-116.241552	100
4	19001.0	Adair	Iowa		US 2020-07-16 04:44:59	41.330756	-94.471059	100



In [ ]: # Fetching all the columns from confirmed dataset  
cols = confirmed\_cases.keys()  
cols

Out[ ]: Index(['Province/State', 'Country/Region', 'Lat', 'Long', '1/22/20', '1/23/20', '1/24/20', '1/25/20', '1/26/20', '1/27/20',  
 ...  
 '7/6/20', '7/7/20', '7/8/20', '7/9/20', '7/10/20', '7/11/20', '7/12/20',  
 '7/13/20', '7/14/20', '7/15/20'],  
 dtype='object', length=180)

In [ ]: # Extracting the date columns  
confirmed = confirmed\_cases.loc[:, cols[4]:cols[-1]]  
deaths = deaths\_reported.loc[:, cols[4]:cols[-1]]  
recoveries = recovered\_cases.loc[:, cols[4]:cols[-1]]

In [ ]: confirmed

Out[ ]: 1/22/20 1/23/20 1/24/20 1/25/20 1/26/20 1/27/20 1/28/20 1/29/20 1/30/20 1/31/20

	1/22/20	1/23/20	1/24/20	1/25/20	1/26/20	1/27/20	1/28/20	1/29/20	1/30/20	1/31/20
<b>241</b>	0	0	0	0	0	0	0	0	0	0
<b>242</b>	0	0	0	0	0	0	0	0	0	0
<b>243</b>	0	0	0	0	0	0	0	0	0	0
<b>244</b>	0	0	0	0	0	0	0	0	0	0
<b>245</b>	0	0	0	0	0	0	0	0	0	0
<b>246</b>	0	0	0	0	0	0	0	0	0	0
<b>247</b>	0	0	0	0	0	0	0	0	0	0
<b>248</b>	0	0	0	0	0	0	0	0	0	0
<b>249</b>	0	0	0	0	0	0	0	0	0	0
<b>250</b>	0	0	0	0	0	0	0	0	0	0
<b>251</b>	0	0	0	0	0	0	0	0	0	0
<b>252</b>	0	0	0	0	0	0	0	0	0	0
<b>253</b>	0	0	0	0	0	0	0	0	0	0
<b>254</b>	0	0	0	0	0	0	0	0	0	0
<b>255</b>	0	0	0	0	0	0	0	0	0	0
<b>256</b>	0	0	0	0	0	0	0	0	0	0
<b>257</b>	0	0	0	0	0	0	0	0	0	0
<b>258</b>	0	0	0	0	0	0	0	0	0	0
<b>259</b>	0	0	0	0	0	0	0	0	0	0
<b>260</b>	0	0	0	0	0	0	0	0	0	0
<b>261</b>	0	0	0	0	0	0	0	0	0	0
<b>262</b>	0	0	0	0	0	0	0	0	0	0
<b>263</b>	0	0	0	0	0	0	0	0	0	0
<b>264</b>	0	0	0	0	0	0	0	0	0	0
<b>265</b>	0	0	0	0	0	0	0	0	0	0

```
In [ ]: # Range of date
dates = confirmed.keys()

# Summary
world_cases = []
total_deaths = []
mortality_rate = []
recovery_rate = []
total_recovered = []
total_active = []

# Confirmed
china_cases = []
italy_cases = []
us_cases = []
spain_cases = []
```

```

france_cases = []
germany_cases = []
uk_cases = []
russia_cases = []
india_cases = []

# Death
china_deaths = []
italy_deaths = []
us_deaths = []
spain_deaths = []
france_deaths = []
germany_deaths = []
uk_deaths = []
russia_deaths = []
india_deaths = []

# Recovered
china_recoveries = []
italy_recoveries = []
us_recoveries = []
spain_recoveries = []
france_recoveries = []
germany_recoveries = []
uk_recoveries = []
russia_recoveries = []
india_recoveries = []

```

```

In [ ]: # Fill with the dataset
for i in dates:
    confirmed_sum = confirmed[i].sum()
    death_sum = deaths[i].sum()
    recovered_sum = recoveries[i].sum()

    world_cases.append(confirmed_sum)
    total_deaths.append(death_sum)
    total_recovered.append(recovered_sum)
    total_active.append(confirmed_sum-death_sum-recovered_sum)

    mortality_rate.append(death_sum/confirmed_sum)
    recovery_rate.append(recovered_sum/confirmed_sum)

    china_cases.append(confirmed_cases[confirmed_cases['Country/Region']=='China'])
    italy_cases.append(confirmed_cases[confirmed_cases['Country/Region']=='Italy'])
    us_cases.append(confirmed_cases[confirmed_cases['Country/Region']=='US'][i].sum())
    spain_cases.append(confirmed_cases[confirmed_cases['Country/Region']=='Spain'])
    france_cases.append(confirmed_cases[confirmed_cases['Country/Region']=='France'])
    germany_cases.append(confirmed_cases[confirmed_cases['Country/Region']=='Germany'])
    uk_cases.append(confirmed_cases[confirmed_cases['Country/Region']=='United Kingdom'])
    russia_cases.append(confirmed_cases[confirmed_cases['Country/Region']=='Russia'])
    india_cases.append(confirmed_cases[confirmed_cases['Country/Region']=='India'])

    china_deaths.append(deaths_reported[deaths_reported['Country/Region']=='China'])
    italy_deaths.append(deaths_reported[deaths_reported['Country/Region']=='Italy'])
    us_deaths.append(deaths_reported[deaths_reported['Country/Region']=='US'][i].sum())
    spain_deaths.append(deaths_reported[deaths_reported['Country/Region']=='Spain'])
    france_deaths.append(deaths_reported[deaths_reported['Country/Region']=='France'])
    germany_deaths.append(deaths_reported[deaths_reported['Country/Region']=='Germany'])
    uk_deaths.append(deaths_reported[deaths_reported['Country/Region']=='United Kingdom'])
    russia_deaths.append(deaths_reported[deaths_reported['Country/Region']=='Russia'])
    india_deaths.append(deaths_reported[deaths_reported['Country/Region']=='India'])

    china_recoveries.append(recovered_cases[recovered_cases['Country/Region']=='China'])
    italy_recoveries.append(recovered_cases[recovered_cases['Country/Region']=='Italy'])

```

```
us_recoveries.append(recovered_cases['Country/Region']=='US')[i]
spain_recoveries.append(recovered_cases['Country/Region']=='Spain')
france_recoveries.append(recovered_cases['Country/Region']=='France')
germany_recoveries.append(recovered_cases['Country/Region']=='Germany')
uk_recoveries.append(recovered_cases['Country/Region']=='United Kingdom')
russia_recoveries.append(recovered_cases['Country/Region']=='Russia')
india_recoveries.append(recovered_cases['Country/Region']=='India')
```

In [ ]: world\_cases

```
Out[ ]: [555,  
654,  
941,  
1434,  
2118,  
2927,  
5578,  
6166,  
8234,  
9927,  
12038,  
16787,  
19887,  
23898,  
27643,  
30802,  
34395,  
37129,  
40159,  
42768,  
44810,  
45228,  
60381,  
66908,  
69050,  
71234,  
73269,  
75151,  
75651,  
76211,  
76840,  
78601,  
78979,  
79543,  
80396,  
81373,  
82737,  
84114,  
86009,  
88386,  
90360,  
92938,  
95237,  
98028,  
101971,  
106009,  
109960,  
113860,  
118884,  
126547,  
132297,  
146704,  
157795,  
168941,  
183597,  
199434,  
219028,  
246949,  
276534,  
308842,  
342255,  
384420,  
425559,  
476376,
```

539336,  
603736,  
671344,  
730921,  
795560,  
871976,  
948197,  
1029598,  
1112752,  
1193128,  
1264846,  
1337518,  
1414391,  
1498166,  
1584791,  
1672449,  
1749169,  
1845951,  
1915545,  
1985472,  
2066301,  
2163013,  
2250737,  
2324083,  
2404790,  
2478129,  
2553289,  
2630185,  
2719198,  
2806138,  
2890259,  
2963206,  
3031819,  
3107209,  
3184143,  
3267936,  
3354984,  
3435571,  
3513207,  
3589284,  
3669273,  
3759295,  
3848381,  
3939898,  
4023533,  
4099779,  
4176020,  
4259383,  
4344371,  
4441476,  
4537825,  
4629638,  
4708147,  
4796431,  
4892855,  
4995697,  
5102217,  
5209117,  
5308100,  
5403426,  
5490389,  
5582911,  
5685511,  
5803600,

```
5924710,  
6053136,  
6160245,  
6256496,  
6377175,  
6496384,  
6623086,  
6755727,  
6881533,  
6993973,  
7097413,  
7220396,  
7354176,  
7492364,  
7621346,  
7755445,  
7888575,  
8010360,  
8149939,  
8325831,  
8464706,  
8645640,  
8803905,  
8934808,  
9072667,  
9238013,  
9405649,  
9583608,  
9774609,  
9952423,  
10117641,  
10273510,  
10447879,  
10664230,  
10872003,  
11076238,  
11269327,  
11452196,  
11620096,  
11829602,  
12041480,  
12268518,  
12498467,  
12717908,  
12910357,  
13104391,  
13324057,  
13554477]
```

```
In [ ]: total_deaths
```

```
Out[ ]: [17,  
18,  
26,  
42,  
56,  
82,  
131,  
133,  
171,  
213,  
259,  
362,  
426,  
492,  
564,  
634,  
719,  
806,  
906,  
1013,  
1113,  
1118,  
1371,  
1523,  
1666,  
1770,  
1868,  
2008,  
2123,  
2248,  
2252,  
2459,  
2470,  
2630,  
2710,  
2771,  
2814,  
2873,  
2942,  
2996,  
3085,  
3160,  
3255,  
3348,  
3460,  
3559,  
3803,  
3987,  
4266,  
4611,  
4916,  
5413,  
5830,  
6471,  
7151,  
7955,  
8852,  
9958,  
11436,  
13142,  
14840,  
16757,  
19027,  
21804,
```

24811,  
28329,  
32008,  
35481,  
39645,  
44489,  
50043,  
56348,  
62333,  
68174,  
73195,  
79027,  
86929,  
93664,  
101293,  
108565,  
114634,  
120365,  
126112,  
133011,  
141323,  
148606,  
157497,  
163955,  
168538,  
173981,  
181134,  
187893,  
194743,  
201417,  
206995,  
210878,  
215525,  
221990,  
228753,  
234720,  
239897,  
245222,  
248675,  
252803,  
258674,  
265343,  
270752,  
276320,  
280585,  
284151,  
287624,  
293168,  
298399,  
303667,  
308882,  
313053,  
316382,  
319673,  
324457,  
329337,  
334128,  
339412,  
343401,  
346541,  
347719,  
351922,  
357135,  
361847,

```
366580,  
370730,  
373599,  
376684,  
381518,  
387097,  
392246,  
397069,  
400890,  
403639,  
407395,  
412253,  
417463,  
422248,  
426545,  
430797,  
434143,  
437647,  
444439,  
449710,  
454721,  
460990,  
465260,  
469317,  
472888,  
478203,  
483377,  
489928,  
494773,  
499291,  
502450,  
506076,  
511268,  
516255,  
521366,  
526426,  
530780,  
534280,  
538058,  
544163,  
549468,  
554924,  
560209,  
565138,  
569128,  
573003,  
578646,  
584124]
```

```
In [ ]: confirmed_sum
```

```
Out[ ]: 13554477
```

```
In [ ]: death_sum
```

```
Out[ ]: 584124
```

```
In [ ]: recovered_sum
```

```
Out[ ]: 7559252
```

```
In [ ]: us_cases
```

```
Out[ ]: [1,
 1,
 2,
 2,
 5,
 5,
 5,
 5,
 5,
 7,
 8,
 8,
 11,
 11,
 11,
 11,
 11,
 11,
 11,
 11,
 11,
 12,
 12,
 13,
 13,
 13,
 13,
 13,
 13,
 13,
 13,
 13,
 13,
 15,
 15,
 15,
 15,
 15,
 15,
 15,
 15,
 16,
 16,
 24,
 30,
 53,
 73,
 104,
 174,
 222,
 337,
 451,
 519,
 711,
 1109,
 1561,
 2157,
 2870,
 2968,
 4360,
 6141,
 8917,
 14157,
 19479,
 25825,
 33761,
 43850,
 54112,
 66055,
```

84091,  
102276,  
122069,  
141205,  
162707,  
188724,  
214205,  
244610,  
276547,  
309699,  
337573,  
367215,  
397992,  
429686,  
464442,  
497943,  
527969,  
556522,  
581813,  
608878,  
637974,  
669272,  
701996,  
730337,  
756375,  
783716,  
809228,  
837422,  
871617,  
907908,  
940829,  
968518,  
990892,  
1015518,  
1042926,  
1072667,  
1106829,  
1136024,  
1161611,  
1184086,  
1208271,  
1233527,  
1261409,  
1288587,  
1314320,  
1334084,  
1352962,  
1374916,  
1396110,  
1423727,  
1449027,  
1474128,  
1493132,  
1514901,  
1535350,  
1558971,  
1584512,  
1608653,  
1630476,  
1651289,  
1670280,  
1689163,  
1707445,  
1730260,

```
1754764,  
1778995,  
1799124,  
1816479,  
1837374,  
1857332,  
1878683,  
1903907,  
1926639,  
1944370,  
1961785,  
1979912,  
2000706,  
2023656,  
2048986,  
2074542,  
2094366,  
2114026,  
2137731,  
2163290,  
2191099,  
2222579,  
2255328,  
2281767,  
2312303,  
2347491,  
2382426,  
2422299,  
2467554,  
2510259,  
2549294,  
2590668,  
2636414,  
2687588,  
2742049,  
2795361,  
2841241,  
2891124,  
2936077,  
2996098,  
3054699,  
3117946,  
3184573,  
3245925,  
3304942,  
3364157,  
3431574,  
3497847]
```

```
In [ ]: def daily_increase(data):  
    d = []  
    for i in range(len(data)):  
        if i == 0:  
            d.append(data[0])  
        else:  
            d.append(data[i]-data[i-1])  
    return d
```

```
In [ ]: # confirmed cases  
world_daily_increase = daily_increase(world_cases)  
china_daily_increase = daily_increase(china_cases)  
italy_daily_increase = daily_increase(italy_cases)  
us_daily_increase = daily_increase(us_cases)  
spain_daily_increase = daily_increase(spain_cases)
```

```
france_daily_increase = daily_increase(france_cases)
germany_daily_increase = daily_increase(germany_cases)
uk_daily_increase = daily_increase(uk_cases)
india_daily_increase = daily_increase(india_cases)
```

```
In [ ]: world_daily_increase
```

```
Out[ ]: [555,
 99,
 287,
 493,
 684,
 809,
 2651,
 588,
 2068,
 1693,
 2111,
 4749,
 3100,
 4011,
 3745,
 3159,
 3593,
 2734,
 3030,
 2609,
 2042,
 418,
 15153,
 6527,
 2142,
 2184,
 2035,
 1882,
 500,
 560,
 629,
 1761,
 378,
 564,
 853,
 977,
 1364,
 1377,
 1895,
 2377,
 1974,
 2578,
 2299,
 2791,
 3943,
 4038,
 3951,
 3900,
 5024,
 7663,
 5750,
 14407,
 11091,
 11146,
 14656,
 15837,
 19594,
 27921,
 29585,
 32308,
 33413,
 42165,
 41139,
 50817,
```

62960,  
64400,  
67608,  
59577,  
64639,  
76416,  
76221,  
81401,  
83154,  
80376,  
71718,  
72672,  
76873,  
83775,  
86625,  
87658,  
76720,  
96782,  
69594,  
69927,  
80829,  
96712,  
87724,  
73346,  
80707,  
73339,  
75160,  
76896,  
89013,  
86940,  
84121,  
72947,  
68613,  
75390,  
76934,  
83793,  
87048,  
80587,  
77636,  
76077,  
79989,  
90022,  
89086,  
91517,  
83635,  
76246,  
76241,  
83363,  
84988,  
97105,  
96349,  
91813,  
78509,  
88284,  
96424,  
102842,  
106520,  
106900,  
98983,  
95326,  
86963,  
92522,  
102600,  
118089,

```
121110,  
128426,  
107109,  
96251,  
120679,  
119209,  
126702,  
132641,  
125806,  
112440,  
103440,  
122983,  
133780,  
138188,  
128982,  
134099,  
133130,  
121785,  
139579,  
175892,  
138875,  
180934,  
158265,  
130903,  
137859,  
165346,  
167636,  
177959,  
191001,  
177814,  
165218,  
155869,  
174369,  
216351,  
207773,  
204235,  
193089,  
182869,  
167900,  
209506,  
211878,  
227038,  
229949,  
219441,  
192449,  
194034,  
219666,  
230420]
```

```
In [ ]: us_daily_increase
```



18036,  
18185,  
19793,  
19136,  
21502,  
26017,  
25481,  
30405,  
31937,  
33152,  
27874,  
29642,  
30777,  
31694,  
34756,  
33501,  
30026,  
28553,  
25291,  
27065,  
29096,  
31298,  
32724,  
28341,  
26038,  
27341,  
25512,  
28194,  
34195,  
36291,  
32921,  
27689,  
22374,  
24626,  
27408,  
29741,  
34162,  
29195,  
25587,  
22475,  
24185,  
25256,  
27882,  
27178,  
25733,  
19764,  
18878,  
21954,  
21194,  
27617,  
25300,  
25101,  
19004,  
21769,  
20449,  
23621,  
25541,  
24141,  
21823,  
20813,  
18991,  
18883,  
18282,  
22815,

```
24504,  
24231,  
20129,  
17355,  
20895,  
19958,  
21351,  
25224,  
22732,  
17731,  
17415,  
18127,  
20794,  
22950,  
25330,  
25556,  
19824,  
19660,  
23705,  
25559,  
27809,  
31480,  
32749,  
26439,  
30536,  
35188,  
34935,  
39873,  
45255,  
42705,  
39035,  
41374,  
45746,  
51174,  
54461,  
53312,  
45880,  
49883,  
44953,  
60021,  
58601,  
63247,  
66627,  
61352,  
59017,  
59215,  
67417,  
66273]
```

```
In [ ]: # deaths  
world_daily_death = daily_increase(total_deaths)  
china_daily_death = daily_increase(china_deaths)  
italy_daily_death = daily_increase(italy_deaths)  
us_daily_death = daily_increase(us_deaths)  
spain_daily_death = daily_increase(spain_deaths)  
france_daily_death = daily_increase(france_deaths)  
germany_daily_death = daily_increase(germany_deaths)  
uk_daily_death = daily_increase(uk_deaths)  
india_daily_death = daily_increase(india_deaths)
```

```
In [ ]: world_daily_death
```

```
Out[ ]: [17,  
1,  
8,  
16,  
14,  
26,  
49,  
2,  
38,  
42,  
46,  
103,  
64,  
66,  
72,  
70,  
85,  
87,  
100,  
107,  
100,  
5,  
253,  
152,  
143,  
104,  
98,  
140,  
115,  
125,  
4,  
207,  
11,  
160,  
80,  
61,  
43,  
59,  
69,  
54,  
89,  
75,  
95,  
93,  
112,  
99,  
244,  
184,  
279,  
345,  
305,  
497,  
417,  
641,  
680,  
804,  
897,  
1106,  
1478,  
1706,  
1698,  
1917,  
2270,  
2777,
```

3007,  
3518,  
3679,  
3473,  
4164,  
4844,  
5554,  
6305,  
5985,  
5841,  
5021,  
5832,  
7902,  
6735,  
7629,  
7272,  
6069,  
5731,  
5747,  
6899,  
8312,  
7283,  
8891,  
6458,  
4583,  
5443,  
7153,  
6759,  
6850,  
6674,  
5578,  
3883,  
4647,  
6465,  
6763,  
5967,  
5177,  
5325,  
3453,  
4128,  
5871,  
6669,  
5409,  
5568,  
4265,  
3566,  
3473,  
5544,  
5231,  
5268,  
5215,  
4171,  
3329,  
3291,  
4784,  
4880,  
4791,  
5284,  
3989,  
3140,  
1178,  
4203,  
5213,  
4712,

```
4733,  
4150,  
2869,  
3085,  
4834,  
5579,  
5149,  
4823,  
3821,  
2749,  
3756,  
4858,  
5210,  
4785,  
4297,  
4252,  
3346,  
3504,  
6792,  
5271,  
5011,  
6269,  
4270,  
4057,  
3571,  
5315,  
5174,  
6551,  
4845,  
4518,  
3159,  
3626,  
5192,  
4987,  
5111,  
5060,  
4354,  
3500,  
3778,  
6105,  
5305,  
5456,  
5285,  
4929,  
3990,  
3875,  
5643,  
5478]
```

```
In [ ]: us_daily_death
```



413,  
554,  
634,  
627,  
820,  
1224,  
1241,  
1586,  
1315,  
1284,  
1439,  
1668,  
2309,  
2116,  
2075,  
2093,  
2046,  
1771,  
1828,  
2404,  
2549,  
2113,  
2614,  
2386,  
1229,  
1795,  
2453,  
2393,  
2437,  
2118,  
1650,  
1293,  
1423,  
2221,  
2501,  
2301,  
1869,  
1556,  
1096,  
1309,  
2307,  
2458,  
1919,  
1727,  
1479,  
879,  
1013,  
1612,  
1754,  
1774,  
1661,  
1223,  
752,  
770,  
1551,  
1563,  
1228,  
1269,  
1121,  
608,  
513,  
685,  
1509,  
1182,

```
1166,  
964,  
591,  
772,  
1033,  
990,  
1009,  
970,  
680,  
451,  
505,  
940,  
917,  
881,  
822,  
750,  
298,  
391,  
839,  
751,  
704,  
672,  
610,  
255,  
406,  
837,  
757,  
2422,  
605,  
489,  
240,  
351,  
721,  
673,  
698,  
639,  
247,  
271,  
325,  
1195,  
820,  
990,  
802,  
685,  
428,  
361,  
900,  
941]
```

```
In [ ]: # recoveries  
world_daily_recovery = daily_increase(total_recovered)  
china_daily_recovery = daily_increase(china_recoveries)  
italy_daily_recovery = daily_increase(italy_recoveries)  
us_daily_recovery = daily_increase(us_recoveries)  
spain_daily_recovery = daily_increase(spain_recoveries)  
france_daily_recovery = daily_increase(france_recoveries)  
germany_daily_recovery = daily_increase(germany_recoveries)  
uk_daily_recovery = daily_increase(uk_recoveries)  
india_daily_recovery = daily_increase(india_recoveries)
```

```
In [ ]: world_daily_recovery
```

```
Out[ ]: [28,  
 2,  
 6,  
 3,  
 13,  
 9,  
 46,  
 19,  
 17,  
 79,  
 62,  
 188,  
 151,  
 229,  
 272,  
 363,  
 524,  
 605,  
 628,  
 702,  
 737,  
 467,  
 1145,  
 1763,  
 1337,  
 1470,  
 1718,  
 1769,  
 1769,  
 2056,  
 713,  
 3996,  
 508,  
 1833,  
 2678,  
 2479,  
 2893,  
 3434,  
 3071,  
 2934,  
 2886,  
 2626,  
 2942,  
 2626,  
 2069,  
 2494,  
 2335,  
 1799,  
 1911,  
 2598,  
 1321,  
 1927,  
 2371,  
 3410,  
 2054,  
 2752,  
 2483,  
 1637,  
 2445,  
 4264,  
 6215,  
 465,  
 9639,  
 5783,
```

8365,  
8769,  
8494,  
9467,  
15437,  
13486,  
15090,  
17041,  
15447,  
20413,  
13839,  
16633,  
23388,  
28716,  
25336,  
21819,  
26247,  
19403,  
27166,  
25093,  
36667,  
30828,  
26120,  
23923,  
31654,  
22002,  
34827,  
30429,  
28791,  
50033,  
27779,  
28603,  
27803,  
33266,  
42168,  
64971,  
38231,  
40917,  
32323,  
34064,  
36605,  
45918,  
39497,  
36522,  
53549,  
33594,  
46994,  
37035,  
55842,  
39735,  
47989,  
56591,  
40886,  
52898,  
52165,  
58913,  
51265,  
108651,  
54971,  
55440,  
63723,  
55214,  
63393,  
66857,

```
77327,  
70472,  
76308,  
54895,  
104123,  
79142,  
70029,  
69156,  
72202,  
55093,  
151598,  
82260,  
79139,  
85889,  
79716,  
85941,  
70778,  
80207,  
97831,  
118786,  
81144,  
95008,  
115825,  
68696,  
91705,  
104058,  
115727,  
92803,  
106636,  
106307,  
89035,  
94914,  
117392,  
115929,  
284872,  
109841,  
195718,  
119441,  
123620,  
145030,  
157951,  
134517,  
139397,  
125778,  
111658,  
140412,  
142105,  
159778]
```

```
In [ ]: us_daily_recovery
```



320,  
188,  
203,  
1593,  
2979,  
1380,  
1450,  
527,  
706,  
4945,  
2796,  
2133,  
2182,  
1796,  
1851,  
3380,  
2480,  
1718,  
10494,  
4281,  
4333,  
2607,  
3842,  
6295,  
5497,  
1992,  
2875,  
2162,  
2837,  
18876,  
1293,  
6616,  
4436,  
4512,  
4784,  
33227,  
10068,  
11367,  
4770,  
7028,  
2611,  
119,  
5126,  
3957,  
13541,  
3635,  
16564,  
-2446,  
13143,  
2984,  
4333,  
17629,  
3889,  
10913,  
6214,  
4920,  
4106,  
51717,  
11104,  
5497,  
12421,  
5745,  
6606,  
8483,

```
6455,  
10015,  
28297,  
13473,  
5637,  
15390,  
5744,  
6704,  
9143,  
5518,  
12155,  
6333,  
8649,  
6788,  
7094,  
9220,  
5210,  
14518,  
7169,  
8688,  
6924,  
7600,  
10745,  
4673,  
18065,  
7350,  
8613,  
7401,  
7247,  
8499,  
5856,  
20039,  
15428,  
9363,  
51976,  
8434,  
103921,  
12438,  
17385,  
12328,  
16986,  
15649,  
14074,  
12391,  
10750,  
25613,  
17159,  
26784]
```

```
In [ ]: unique_countries = list(latest_data['Country_Region'].unique())  
unique_countries
```

```
Out[ ]: ['US',
 'Italy',
 'Brazil',
 'Russia',
 'Mexico',
 'Japan',
 'Canada',
 'Colombia',
 'Peru',
 'Spain',
 'India',
 'United Kingdom',
 'China',
 'Chile',
 'Netherlands',
 'Australia',
 'Pakistan',
 'Germany',
 'Sweden',
 'Ukraine',
 'Denmark',
 'France',
 'Afghanistan',
 'Albania',
 'Algeria',
 'Andorra',
 'Angola',
 'Antigua and Barbuda',
 'Argentina',
 'Armenia',
 'Austria',
 'Azerbaijan',
 'Bahamas',
 'Bahrain',
 'Bangladesh',
 'Barbados',
 'Belarus',
 'Belgium',
 'Belize',
 'Benin',
 'Bhutan',
 'Bolivia',
 'Bosnia and Herzegovina',
 'Botswana',
 'Brunei',
 'Bulgaria',
 'Burkina Faso',
 'Burma',
 'Burundi',
 'Cabo Verde',
 'Cambodia',
 'Cameroon',
 'Central African Republic',
 'Chad',
 'Comoros',
 'Congo (Brazzaville)',
 'Congo (Kinshasa)',
 'Costa Rica',
 "Cote d'Ivoire",
 'Croatia',
 'Cuba',
 'Cyprus',
 'Czechia',
 'Diamond Princess',
```

'Djibouti',  
'Dominica',  
'Dominican Republic',  
'Ecuador',  
'Egypt',  
'El Salvador',  
'Equatorial Guinea',  
'Eritrea',  
'Estonia',  
'Eswatini',  
'Ethiopia',  
'Fiji',  
'Finland',  
'Gabon',  
'Gambia',  
'Georgia',  
'Ghana',  
'Greece',  
'Grenada',  
'Guatemala',  
'Guinea',  
'Guinea-Bissau',  
'Guyana',  
'Haiti',  
'Holy See',  
'Honduras',  
'Hungary',  
'Iceland',  
'Indonesia',  
'Iran',  
'Iraq',  
'Ireland',  
'Israel',  
'Jamaica',  
'Jordan',  
'Kazakhstan',  
'Kenya',  
'Korea, South',  
'Kosovo',  
'Kuwait',  
'Kyrgyzstan',  
'Laos',  
'Latvia',  
'Lebanon',  
'Lesotho',  
'Liberia',  
'Libya',  
'Liechtenstein',  
'Lithuania',  
'Luxembourg',  
'MS Zaandam',  
'Madagascar',  
'Malawi',  
'Malaysia',  
'Maldives',  
'Mali',  
'Malta',  
'Mauritania',  
'Mauritius',  
'Moldova',  
'Monaco',  
'Mongolia',  
'Montenegro',  
'Morocco',

```
'Mozambique',
'Namibia',
'Nepal',
'New Zealand',
'Nicaragua',
'Niger',
'Nigeria',
'North Macedonia',
'Norway',
'Oman',
'Panama',
'Papua New Guinea',
'Paraguay',
'Philippines',
'Poland',
'Portugal',
'Qatar',
'Romania',
'Rwanda',
'Saint Kitts and Nevis',
'Saint Lucia',
'Saint Vincent and the Grenadines',
'San Marino',
'Sao Tome and Principe',
'Saudi Arabia',
'Senegal',
'Serbia',
'Seychelles',
'Sierra Leone',
'Singapore',
'Slovakia',
'Slovenia',
'Somalia',
'South Africa',
'South Sudan',
'Sri Lanka',
'Sudan',
'Suriname',
'Switzerland',
'Syria',
'Taiwan*',
'Tajikistan',
'Tanzania',
'Thailand',
'Timor-Leste',
'Togo',
'Trinidad and Tobago',
'Tunisia',
'Turkey',
'Uganda',
'United Arab Emirates',
'Uruguay',
'Uzbekistan',
'Venezuela',
'Vietnam',
'West Bank and Gaza',
'Western Sahara',
'Yemen',
'Zambia',
'Zimbabwe']
```

```
In [ ]: confirmed_by_country = []
death_by_country = []
active_by_country = []
```

```

recovery_by_country = []
mortality_rate_by_country = []

no_cases = []
for i in unique_countries:
    cases = latest_data[latest_data['Country_Region']==i]['Confirmed'].sum()
    if cases > 0:
        confirmed_by_country.append(cases)
    else:
        no_cases.append(i)

for i in no_cases:
    unique_countries.remove(i)

# sort countries by the number of confirmed cases
unique_countries = [k for k, v in sorted(zip(unique_countries, confirmed_by_country),
                                         key=lambda x: x[1], reverse=True)]
for i in range(len(unique_countries)):
    confirmed_by_country[i] = latest_data[latest_data['Country_Region']==unique_countries[i]]['Confirmed'].sum()
    death_by_country.append(latest_data[latest_data['Country_Region']==unique_countries[i]]['Deaths'].sum())
    recovery_by_country.append(latest_data[latest_data['Country_Region']==unique_countries[i]]['Recovered'].sum())
    active_by_country.append(confirmed_by_country[i] - death_by_country[i] - recovery_by_country[i])
    mortality_rate_by_country.append(death_by_country[i]/confirmed_by_country[i])

```

In [ ]:

```

country_df = pd.DataFrame({'Country Name': unique_countries, 'Number of Confirmed Cases': confirmed_by_country,
                           'Number of Deaths': death_by_country, 'Number of Recovered Cases': recovery_by_country,
                           'Number of Active Cases' : active_by_country,
                           'Mortality Rate': mortality_rate_by_country})
# number of cases per country/region

country_df.style.background_gradient(cmap='Blues')

```

Out[ ]:

	Country Name	Number of Confirmed Cases	Number of Deaths	Number of Recoveries	Number of Active Cases	Mortality Rate
0	US	3497847	137407	1075882	2284558	0.0392833
1	Brazil	1966748	75366	1350098	541284	0.0383201
2	India	968857	24914	612768	331175	0.0257148
3	Russia	745197	11753	522375	211069	0.0157717
4	Peru	337751	12417	226400	98934	0.0367638
5	Chile	321205	7186	292085	21934	0.022372
6	Mexico	317635	36906	252368	28361	0.11619
7	South Africa	311049	4453	160693	145903	0.0143161
8	United Kingdom	293469	45138	1386	246945	0.153808
9	Iran	264561	13410	227561	23590	0.0506877
10	Pakistan	257914	5426	178737	73751	0.021038
11	Spain	257494	28413	150376	78705	0.110344
12	Italy	243506	34997	196016	12493	0.143721
13	Saudi Arabia	240474	2325	183048	55101	0.0096684
14	Turkey	215940	5419	197733	12788	0.0250949
15	France	210568	30123	78945	101500	0.143056
16	Germany	200890	9080	186000	5810	0.0451989
17	Bangladesh	193590	2457	105523	85610	0.0126918
18	Colombia	159898	5969	68806	85123	0.03733
19	Argentina	111146	2050	47298	61798	0.0184442
20	Canada	110693	8857	74067	27769	0.0800141
21	Qatar	104983	151	101637	3195	0.00143833
22	China	85246	4644	80005	597	0.0544776
23	Egypt	84843	4067	26135	54641	0.0479356
24	Iraq	83867	3432	52621	27814	0.0409219
25	Indonesia	80094	3797	39050	37247	0.0474068
26	Sweden	76492	5572	0	70920	0.0728442
27	Ecuador	70329	5158	30641	34530	0.073341
28	Belarus	65443	480	56379	8584	0.00733463
29	Kazakhstan	65188	375	39066	25747	0.00575259
30	Belgium	62872	9788	17242	35842	0.155681
31	Oman	61247	281	39038	21928	0.00458798
32	Philippines	58850	1614	20976	36260	0.0274257
33	Kuwait	56877	399	46897	9581	0.00701514
34	Ukraine	56779	1444	29005	26330	0.0254319

	Country Name	Number of Confirmed Cases	Number of Deaths	Number of Recoveries	Number of Active Cases	Mortality Rate
35	United Arab Emirates	55848	335	46418	9095	0.00599842
36	Bolivia	52218	1942	16357	33919	0.0371902
37	Netherlands	51471	6155	193	45123	0.119582
38	Panama	49243	982	25417	22844	0.0199419
39	Dominican Republic	47671	929	23459	23283	0.0194877
40	Portugal	47426	1676	32110	13640	0.0353393
41	Singapore	46878	27	42988	3863	0.000575963
42	Israel	44188	376	19989	23823	0.0085091
43	Poland	38721	1594	28492	8635	0.0411663
44	Afghanistan	34994	1094	22456	11444	0.0312625
45	Bahrain	34560	117	30320	4123	0.00338542
46	Nigeria	34259	760	13999	19500	0.022184
47	Romania	34226	1952	22049	10225	0.0570327
48	Switzerland	33148	1968	29800	1380	0.0593701
49	Armenia	33005	592	21348	11065	0.0179367
50	Guatemala	32074	1350	4624	26100	0.0420902
51	Honduras	30036	825	3379	25832	0.027467
52	Ireland	25683	1748	23364	571	0.0680606
53	Azerbaijan	25672	326	16695	8651	0.0126987
54	Ghana	25430	139	21511	3780	0.00546599
55	Japan	23172	984	18126	4062	0.042465
56	Algeria	20770	1040	14792	4938	0.0500722
57	Moldova	20040	659	13298	6083	0.0328842
58	Serbia	19334	429	14047	4858	0.0221889
59	Austria	19154	710	17175	1269	0.037068
60	Nepal	17177	39	11025	6113	0.00227048
61	Morocco	16262	259	13821	2182	0.0159267
62	Cameroon	15173	359	11928	2886	0.0236604
63	Uzbekistan	14581	71	8655	5855	0.00486935
64	Korea, South	13612	291	12396	925	0.0213782
65	Czechia	13475	355	8507	4613	0.0263451
66	Cote d'Ivoire	13403	87	7146	6170	0.00649108
67	Denmark	13293	610	12383	300	0.0458888
68	Kyrgyzstan	12282	165	3712	8405	0.0134343

	Country Name	Number of Confirmed Cases	Number of Deaths	Number of Recoveries	Number of Active Cases	Mortality Rate
69	Kenya	11252	209	3068	7975	0.0185745
70	Australia	10810	113	8035	2662	0.0104533
71	El Salvador	10645	286	6120	4239	0.0268671
72	Sudan	10527	668	5601	4258	0.0634559
73	Venezuela	10428	100	3050	7278	0.00958957
74	Norway	9011	253	8138	620	0.0280768
75	Costa Rica	8986	40	2551	6395	0.00445137
76	Malaysia	8734	122	8526	86	0.0139684
77	North Macedonia	8530	393	4565	3572	0.0460727
78	Senegal	8369	153	5605	2611	0.0182818
79	Ethiopia	8181	146	2430	5605	0.0178462
80	Congo (Kinshasa)	8163	192	3983	3988	0.0235208
81	Bulgaria	7877	289	3841	3747	0.0366891
82	Bosnia and Herzegovina	7411	235	3335	3841	0.0317096
83	Finland	7296	328	6880	88	0.0449561
84	West Bank and Gaza	7064	44	1084	5936	0.00622877
85	Haiti	6831	143	3283	3405	0.020934
86	Tajikistan	6695	56	5383	1256	0.00836445
87	Guinea	6276	38	4981	1257	0.00605481
88	Gabon	6121	46	3664	2411	0.00751511
89	Madagascar	5605	43	2811	2751	0.00767172
90	Mauritania	5564	149	2830	2585	0.0267793
91	Kosovo	5237	112	2462	2663	0.0213863
92	Luxembourg	5122	111	4247	764	0.0216712
93	Djibouti	4985	56	4765	164	0.0112337
94	Central African Republic	4362	53	1261	3048	0.0121504
95	Hungary	4263	595	3126	542	0.139573
96	Croatia	3953	120	2629	1204	0.0303567
97	Greece	3910	193	1374	2343	0.0493606
98	Albania	3752	101	2091	1560	0.026919
99	Thailand	3232	58	3092	82	0.0179455
100	Paraguay	3198	25	1338	1835	0.00781739
101	Nicaragua	3147	99	2282	766	0.0314585

	Country Name	Number of Confirmed Cases	Number of Deaths	Number of Recoveries	Number of Active Cases	Mortality Rate
102	Somalia	3083	93	1425	1565	0.0301654
103	Equatorial Guinea	3071	51	842	2178	0.016607
104	Maldives	2831	14	2321	496	0.00494525
105	Sri Lanka	2671	11	2001	659	0.00411831
106	Malawi	2614	43	1005	1566	0.0164499
107	Lebanon	2542	38	1455	1049	0.0149489
108	Cuba	2438	87	2277	74	0.035685
109	Mali	2433	121	1764	548	0.0497328
110	Congo (Brazzaville)	2222	47	589	1586	0.0211521
111	South Sudan	2153	41	1175	937	0.0190432
112	Estonia	2016	69	1901	46	0.0342262
113	Slovakia	1927	28	1507	392	0.0145304
114	Iceland	1911	10	1885	16	0.00523286
115	Zambia	1895	42	1412	441	0.0221636
116	Lithuania	1882	79	1582	221	0.0419766
117	Slovenia	1878	111	1501	266	0.0591054
118	Guinea-Bissau	1842	26	773	1043	0.0141151
119	Cabo Verde	1780	19	850	911	0.0106742
120	Sierra Leone	1668	64	1200	404	0.0383693
121	Libya	1589	43	373	1173	0.027061
122	New Zealand	1548	22	1499	27	0.0142119
123	Yemen	1526	433	694	399	0.283748
124	Eswatini	1489	20	724	745	0.0134318
125	Rwanda	1435	4	752	679	0.00278746
126	Benin	1378	26	557	795	0.0188679
127	Mozambique	1330	9	375	946	0.00676692
128	Tunisia	1319	50	1091	178	0.0379075
129	Montenegro	1287	24	330	933	0.018648
130	Jordan	1201	10	1016	175	0.00832639
131	Latvia	1178	31	1022	125	0.0263158
132	Niger	1100	69	993	38	0.0627273
133	Zimbabwe	1089	20	395	674	0.0183655
134	Liberia	1056	51	447	558	0.0482955
135	Uganda	1043	0	1004	39	0

	Country Name	Number of Confirmed Cases	Number of Deaths	Number of Recoveries	Number of Active Cases	Mortality Rate
136	Burkina Faso	1038	53	882	103	0.0510597
137	Cyprus	1025	19	839	167	0.0185366
138	Uruguay	1009	31	909	69	0.0307235
139	Georgia	1004	15	873	116	0.0149402
140	Namibia	960	2	31	927	0.00208333
141	Chad	885	75	799	11	0.0847458
142	Andorra	862	52	803	7	0.0603248
143	Suriname	837	18	573	246	0.0215054
144	Jamaica	763	10	647	106	0.0131062
145	Togo	740	15	534	191	0.0202703
146	Sao Tome and Principe	737	14	322	401	0.0189959
147	Diamond Princess	712	13	651	48	0.0182584
148	San Marino	699	42	656	1	0.0600858
149	Malta	674	9	661	4	0.0133531
150	Angola	576	27	124	425	0.046875
151	Tanzania	509	21	183	305	0.0412574
152	Syria	458	22	140	296	0.0480349
153	Taiwan*	451	7	440	4	0.0155211
154	Botswana	399	1	38	360	0.00250627
155	Vietnam	381	0	353	28	0
156	Mauritius	343	10	331	2	0.0291545
157	Burma	337	6	266	65	0.0178042
158	Comoros	321	7	302	12	0.0218069
159	Guyana	313	18	156	139	0.057508
160	Burundi	269	1	207	61	0.00371747
161	Mongolia	261	0	209	52	0
162	Lesotho	256	3	48	205	0.0117188
163	Eritrea	232	0	107	125	0
164	Cambodia	166	0	133	33	0
165	Brunei	141	3	138	0	0.0212766
166	Trinidad and Tobago	133	8	124	1	0.0601504
167	Bahamas	119	11	91	17	0.092437
168	Monaco	109	4	97	8	0.0366972
169	Barbados	104	7	90	7	0.0673077

	Country Name	Number of Confirmed Cases	Number of Deaths	Number of Recoveries	Number of Active Cases	Mortality Rate
170	Seychelles	100	0	27	73	0
171	Bhutan	84	0	78	6	0
172	Liechtenstein	84	1	81	2	0.0119048
173	Antigua and Barbuda	74	3	57	14	0.0405405
174	Gambia	64	3	34	27	0.046875
175	Belize	39	2	21	16	0.0512821
176	Saint Vincent and the Grenadines	35	0	29	6	0
177	Fiji	26	0	18	8	0
178	Timor-Leste	24	0	24	0	0
179	Grenada	23	0	23	0	0
180	Saint Lucia	22	0	19	3	0
181	Laos	19	0	19	0	0
182	Dominica	18	0	18	0	0
183	Saint Kitts and Nevis	17	0	15	2	0
184	Holy See	12	0	12	0	0
185	Papua New Guinea	11	0	8	3	0
186	Western Sahara	10	1	8	1	0.1
187	MS Zaandam	9	2	0	7	0.222222

```
In [ ]: unique_provinces = list(latest_data['Province_State'].unique())
```

```
In [ ]: confirmed_by_province = []
country_by_province = []
death_by_province = []
recovery_by_province = []
mortality_rate_by_province = []

no_cases = []
for i in unique_provinces:
    cases = latest_data[latest_data['Province_State']==i]['Confirmed'].sum()
    if cases > 0:
        confirmed_by_province.append(cases)
    else:
        no_cases.append(i)

# remove areas with no confirmed cases
for i in no_cases:
    unique_provinces.remove(i)

unique_provinces = [k for k, v in sorted(zip(unique_provinces, confirmed_by_province))]
for i in range(len(unique_provinces)):
    confirmed_by_province[i] = latest_data[latest_data['Province_State']==unique_provinces[i]]['Confirmed'].sum()
    country_by_province.append(latest_data[latest_data['Province_State']==unique_provinces[i]]['Country_Region'].values[0])
    death_by_province.append(latest_data[latest_data['Province_State']==unique_provinces[i]]['Deaths'].sum())
    recovery_by_province.append(latest_data[latest_data['Province_State']==unique_provinces[i]]['Recovered'].sum())
    mortality_rate_by_province.append(latest_data[latest_data['Province_State']==unique_provinces[i]]['Mortality_Rate'].values[0])
```

```
recovery_by_province.append(latest_data[latest_data['Province_State']==unique_p  
mortality_rate_by_province.append(death_by_province[i]/confirmed_by_province[i])
```

```
In [ ]: # number of cases per province/state/city  
province_df = pd.DataFrame({'Province/State Name': unique_provinces, 'Country': cou  
'Number of Deaths': death_by_province, 'Number of Recover  
'Mortality Rate': mortality_rate_by_province})  
# number of cases per country/region  
  
province_df.style.background_gradient(cmap='Reds')
```

Out[ ]:

	Province/State Name	Country	Number of Confirmed Cases	Number of Deaths	Number of Recoveries	Mortality Rate
0	New York	US	404006	32427	0	0.0802637
1	Sao Paulo	Brazil	393176	18640	246941	0.0474088
2	California	US	354885	7375	0	0.0207814
3	Florida	US	301810	4521	0	0.0149796
4	Texas	US	289837	3498	0	0.0120689
5	Maharashtra	India	275640	10928	152613	0.0396459
6	England	United Kingdom	250885	40462	0	0.161277
7	Metropolitana	Chile	241345	5957	224418	0.0246825
8	Moscow	Russia	231270	4234	167810	0.0183076
9	Lima	Peru	181131	5763	0	0.0318168
10	New Jersey	US	176278	15634	0	0.0886895
11	Illinois	US	157825	7427	0	0.0470585
12	Tamil Nadu	India	151820	2167	102310	0.0142735
13	Ceara	Brazil	141248	7030	115426	0.0497706
14	Rio de Janeiro	Brazil	134449	11757	114351	0.0874458
15	Arizona	US	131354	2434	0	0.0185301
16	Para	Brazil	130834	5337	115757	0.0407921
17	Georgia	US	127838	3091	0	0.024179
18	Delhi	India	116993	3487	95699	0.0298052
19	Bahia	Brazil	112993	2638	86708	0.0233466
20	Massachusetts	US	112347	8368	0	0.0744835
21	Sindh	Pakistan	108913	1888	70292	0.0173349
22	Maranhao	Brazil	102469	2572	77815	0.0251003
23	Pennsylvania	US	102269	6957	0	0.0680265
24	Punjab	India	97338	2272	70114	0.0233413
25	Lombardia	Italy	95236	16765	70936	0.176036
26	Amazonas	Brazil	92911	3279	74157	0.0352918
27	North Carolina	US	91837	1589	0	0.0173024
28	Louisiana	US	84131	3461	0	0.0411382
29	Minas Gerais	Brazil	82010	1752	56001	0.0213632
30	Michigan	US	78913	6330	0	0.0802149
31	Distrito Federal	Brazil	75379	1001	59900	0.0132796
32	Maryland	US	75016	3341	0	0.0445372
33	Pernambuco	Brazil	74960	5772	51095	0.0770011

	Province/State Name	Country	Number of Confirmed Cases	Number of Deaths	Number of Recoveries	Mortality Rate
34	Virginia	US	73527	1992	0	0.0270921
35	Madrid	Spain	72797	8444	40736	0.115994
36	Ohio	US	69311	3075	0	0.0443653
37	Tennessee	US	69061	783	0	0.0113378
38	Hubei	China	68135	4512	63623	0.0662215
39	Espirito Santo	Brazil	66352	2097	44519	0.0316042
40	Catalonia	Spain	65852	5678	26203	0.0862237
41	Paraiba	Brazil	63939	1383	22468	0.02163
42	South Carolina	US	62245	998	0	0.0160334
43	Moscow Oblast	Russia	61059	1022	42013	0.0167379
44	Ciudad de Mexico	Mexico	59670	7910	47891	0.132562
45	Alabama	US	59067	1211	0	0.0205021
46	Quebec	Canada	56859	5636	26097	0.0991224
47	Indiana	US	53370	2785	0	0.0521829
48	Capital District	Colombia	53131	1299	19075	0.0244449
49	Bayern	Germany	49427	2613	45870	0.0528658
50	Santa Catarina	Brazil	47976	569	37027	0.0118601
51	Alagoas	Brazil	47864	1331	40876	0.027808
52	Connecticut	US	47636	4380	0	0.0919473
53	Karnataka	India	47253	928	18466	0.019639
54	Parana	Brazil	47124	1200	15383	0.0254647
55	Nordrhein-Westfalen	Germany	45233	1708	41386	0.03776
56	Gujarat	India	44552	2079	31286	0.0466646
57	Mexico	Mexico	44117	5439	36392	0.123286
58	Minnesota	US	43742	1558	0	0.0356179
59	Washington	US	43046	1421	0	0.0330112
60	Rio Grande do Sul	Brazil	42239	1101	35388	0.026066
61	Uttar Pradesh	India	41383	1012	25743	0.0244545
62	Rio Grande do Norte	Brazil	40654	1473	3258	0.0362326
63	Sergipe	Brazil	40139	1054	23439	0.0262588
64	Telangana	India	39342	386	25999	0.0098114
65	Ontario	Canada	38918	2778	34502	0.0713809
66	Goias	Brazil	38895	927	11965	0.0238334
67	Wisconsin	US	38727	827	0	0.0213546
68	Mississippi	US	38567	1290	0	0.0334483

	Province/State Name	Country	Number of Confirmed Cases	Number of Deaths	Number of Recoveries	Mortality Rate
69	Colorado	US	38137	1744	0	0.0457299
70	Iowa	US	36536	777	0	0.0212667
71	Baden-Wurttemberg	Germany	36237	1838	33980	0.0507216
72	Atlantico	Colombia	35649	2039	16317	0.0571966
73	Andhra Pradesh	India	35451	452	18378	0.01275
74	Piaui	Brazil	35445	1019	33551	0.0287488
75	West Bengal	India	34427	1000	20680	0.029047
76	Amapa	Brazil	32408	488	21108	0.015058
77	Piemonte	Italy	31515	4118	26510	0.130668
78	Khyber Pakhtunkhwa	Pakistan	31217	1120	22260	0.0358779
79	Utah	US	30891	233	0	0.00754265
80	Nevada	US	30468	618	0	0.0202836
81	Mato Grosso	Brazil	30319	1174	11660	0.0387216
82	Arkansas	US	30297	335	0	0.0110572
83	Missouri	US	30057	1119	0	0.0372293
84	Emilia-Romagna	Italy	28989	4271	23548	0.147332
85	Saint Petersburg	Russia	28406	1676	21746	0.0590016
86	Rondonia	Brazil	27917	668	16888	0.0239281
87	Rajasthan	India	26437	530	19502	0.0200477
88	Roraima	Brazil	23681	403	8194	0.0170179
89	Haryana	India	23306	319	17667	0.0136875
90	Oklahoma	US	22814	432	0	0.0189357
91	Stockholm	Sweden	22382	2372	0	0.105978
92	Nebraska	US	21979	291	0	0.0132399
93	Nizhny Novgorod Oblast	Russia	20763	312	15022	0.0150267
94	Kentucky	US	20677	645	0	0.0311941
95	Bihar	India	20612	180	13462	0.00873278
96	Kansas	US	20349	304	0	0.0149393
97	Castilla y Leon	Spain	19836	2793	8716	0.140805
98	Madhya Pradesh	India	19643	682	13908	0.0347197
99	Veneto	Italy	19441	2043	16960	0.105087
100	Callao	Peru	19307	735	0	0.0380691
101	Piura	Peru	19096	853	0	0.044669
102	Assam	India	18666	46	12173	0.00246437

	Province/State Name	Country	Number of Confirmed Cases	Number of Deaths	Number of Recoveries	Mortality Rate
103	Castilla - La Mancha	Spain	18399	3031	6392	0.164737
104	Scotland	United Kingdom	18373	2490	0	0.135525
105	Rhode Island	US	17640	987	0	0.0559524
106	Vastra Gotaland	Sweden	16983	798	0	0.0469882
107	Wales	United Kingdom	16854	1545	0	0.0916696
108	Acre	Brazil	16672	446	9198	0.0267514
109	Sverdlovsk Oblast	Russia	16635	175	10614	0.01052
110	Tocantins	Brazil	16031	271	9823	0.0169047
111	Tabasco	Mexico	15909	1485	12789	0.0933434
112	New Mexico	US	15841	557	0	0.0351619
113	Lambayeque	Peru	15308	805	0	0.0525869
114	Valle del Cauca	Colombia	15132	625	7233	0.0413032
115	Veracruz	Mexico	15073	2058	11490	0.136536
116	Valparaiso	Chile	15016	347	12844	0.0231087
117	Odisha	India	14898	77	10476	0.00516848
118	Puebla	Mexico	14876	1864	12134	0.125303
119	Mato Grosso do Sul	Brazil	14631	183	9401	0.0125077
120	Islamabad	Pakistan	14402	156	11486	0.0108318
121	Pais Vasco	Spain	14043	1563	16160	0.111301
122	Niedersachsen	Germany	13848	644	12806	0.0465049
123	Andalusia	Spain	13573	1435	10671	0.105725
124	Guanajuato	Mexico	13329	668	10198	0.0501163
125	Oregon	US	13081	247	0	0.0188823
126	Delaware	US	13050	521	0	0.0399234
127	Khanty-Mansi Autonomous Okrug	Russia	12993	96	7530	0.00738859
128	Sonora	Mexico	12712	1198	10952	0.0942417
129	Bolivar	Colombia	12596	484	7300	0.0384249
130	Idaho	US	12449	110	0	0.00883605
131	La Libertad	Peru	12430	827	0	0.0665326
132	Antofagasta	Chile	12389	232	10392	0.0187263
133	C. Valenciana	Spain	11854	1432	9970	0.120803
134	Jammu and Kashmir	India	11666	206	6337	0.0176582
135	Balochistan	Pakistan	11322	127	8002	0.0112171

	Province/State Name	Country	Number of Confirmed Cases	Number of Deaths	Number of Recoveries	Mortality Rate
136	Baja California	Mexico	11247	2336	8434	0.2077
137	Hessen	Germany	11217	514	10301	0.0458233
138	District of Columbia	US	11026	571	0	0.0517867
139	Rostov Oblast	Russia	11019	172	8451	0.0156094
140	Krasnoyarsk Krai	Russia	10889	235	7407	0.0215814
141	Nuevo Leon	Mexico	10830	590	8389	0.0544783
142	Sinaloa	Mexico	10748	1714	8440	0.159472
143	Irkutsk Oblast	Russia	10685	107	5278	0.010014
144	Antioquia	Colombia	10576	124	3089	0.0117247
145	Puerto Rico	US	10379	171	0	0.0164756
146	Toscana	Italy	10338	1127	8898	0.109015
147	Loreto	Peru	10093	364	0	0.0360646
148	Liguria	Italy	10042	1561	8285	0.155447
149	Ancash	Peru	9926	565	0	0.0569212
150	Jalisco	Mexico	9760	1105	7317	0.113217
151	Tamaulipas	Mexico	9687	639	8174	0.0659647
152	Ica	Peru	9674	644	0	0.0665702
153	Voronezh Oblast	Russia	9631	69	6910	0.00716437
154	Arequipa	Peru	9560	474	0	0.0495816
155	Kerala	India	9553	35	4634	0.00366377
156	Galicia	Spain	9471	619	9204	0.0653574
157	Chelyabinsk Oblast	Russia	9164	88	5212	0.00960279
158	Alberta	Canada	8994	163	8127	0.0181232
159	Ucayali	Peru	8932	170	0	0.0190327
160	Biobio	Chile	8792	80	7481	0.00909918
161	O Higgins	Chile	8755	134	7267	0.0153055
162	Berlin	Germany	8698	221	8114	0.0254081
163	Tokyo	Japan	8644	325	6336	0.0375983
164	Dagestan Republic	Russia	8602	419	7019	0.0487096
165	Yamalo-Nenets Autonomous Okrug	Russia	8572	57	3535	0.00664956
166	Lazio	Italy	8376	847	6631	0.101122
167	Guerrero	Mexico	8149	1128	6378	0.138422
168	Oaxaca	Mexico	8029	760	6795	0.0946569
169	Coahuila	Mexico	7988	401	6395	0.0502003

	Province/State Name	Country	Number of Confirmed Cases	Number of Deaths	Number of Recoveries	Mortality Rate
170	Saratov Oblast	Russia	7874	48	4463	0.00609601
171	Volgograd Oblast	Russia	7694	50	4271	0.00649857
172	Tula Oblast	Russia	7688	166	6273	0.0215921
173	South Dakota	US	7652	111	0	0.014506
174	Novosibirsk Oblast	Russia	7608	152	5557	0.019979
175	Murmansk Oblast	Russia	7597	39	4149	0.00513361
176	Michoacan	Mexico	7430	591	6464	0.0795424
177	Lviv Oblast	Ukraine	7425	193	1167	0.0259933
178	Ulyanovsk Oblast	Russia	7382	54	4595	0.00731509
179	Arkhangelsk Oblast	Russia	7370	113	4699	0.0153324
180	Maule	Chile	7341	96	6221	0.0130772
181	Tarapaca	Chile	7266	115	6500	0.0158271
182	Rheinland-Pfalz	Germany	7232	236	6737	0.0326327
183	Aragon	Spain	7111	915	3772	0.128674
184	Krasnodar Krai	Russia	6912	89	5877	0.0128762
185	Bryansk Oblast	Russia	6906	35	5210	0.00506806
186	Marche	Italy	6805	987	5661	0.14504
187	San Martin	Peru	6797	188	0	0.0276593
188	Kaluga Oblast	Russia	6742	52	5053	0.00771284
189	Yucatan	Mexico	6674	612	4796	0.0916991
190	Altai Krai	Russia	6540	54	4142	0.00825688
191	Kiev	Ukraine	6497	126	2160	0.0193936
192	Stavropol Krai	Russia	6394	123	4159	0.0192368
193	French Guiana	France	6299	33	3738	0.00523893
194	New Hampshire	US	6113	394	0	0.0644528
195	Samara Oblast	Russia	6078	70	3745	0.0115169
196	Bashkortostan Republic	Russia	6077	19	4255	0.00312654
197	Chuvashia Republic	Russia	5998	47	3253	0.00783595
198	Penza Oblast	Russia	5844	68	5208	0.0116359
199	Northern Ireland	United Kingdom	5799	556	0	0.0958786
200	Ryazan Oblast	Russia	5766	22	4735	0.00381547
201	Kursk Oblast	Russia	5734	48	3982	0.00837112
202	Ivanovo Oblast	Russia	5691	59	4015	0.0103672

	Province/State Name	Country	Number of Confirmed Cases	Number of Deaths	Number of Recoveries	Mortality Rate
203	Kabardino-Balkarian Republic	Russia	5686	61	5193	0.0107281
204	Khabarovsk Krai	Russia	5637	43	2796	0.00762817
205	Navarra	Spain	5627	528	3905	0.0938333
206	Tyva Republic	Russia	5537	42	2366	0.00758534
207	Quintana Roo	Mexico	5533	744	3834	0.134466
208	Junin	Peru	5524	241	0	0.0436278
209	Sachsen	Germany	5478	225	5220	0.0410734
210	Yaroslavl Oblast	Russia	5387	22	4544	0.00408391
211	Tambov Oblast	Russia	5361	28	3987	0.00522291
212	Chiapas	Mexico	5308	786	4338	0.148078
213	Belgorod Oblast	Russia	5271	42	4687	0.00796813
214	Primorsky Krai	Russia	5236	60	3963	0.0114591
215	Hamburg	Germany	5232	261	4900	0.0498853
216	Smolensk Oblast	Russia	5212	81	3129	0.0155411
217	Chernivtsi Oblast	Ukraine	5208	201	2998	0.0385945
218	Leningrad Oblast	Russia	5177	42	4289	0.00811281
219	Orel Oblast	Russia	5153	75	3714	0.0145546
220	Orenburg Oblast	Russia	5126	32	3869	0.00624268
221	Tatarstan Republic	Russia	5093	16	3979	0.00314157
222	Vladimir Oblast	Russia	5048	96	4113	0.0190174
223	Hidalgo	Mexico	5035	822	3726	0.163257
224	San Luis Potosi	Mexico	5002	285	3854	0.0569772
225	Sakha (Yakutiya) Republic	Russia	4976	37	4428	0.00743569
226	Omsk Oblast	Russia	4949	44	2849	0.00889068
227	P.A. Trento	Italy	4881	405	4452	0.0829748
228	Campania	Italy	4787	432	4096	0.0902444
229	Rivne Oblast	Ukraine	4782	83	3101	0.0173568
230	Victoria	Australia	4750	29	2591	0.00610526
231	Narino	Colombia	4660	155	2778	0.0332618
232	Perm Krai	Russia	4618	97	3618	0.0210048
233	North Dakota	US	4565	88	0	0.0192771
234	West Virginia	US	4557	98	0	0.0215054
235	Puglia	Italy	4541	547	3926	0.120458
236	Chhattisgarh	India	4539	20	3324	0.00440626

	Province/State Name	Country	Number of Confirmed Cases	Number of Deaths	Number of Recoveries	Mortality Rate
237	Jonkoping	Sweden	4505	177	0	0.0392897
238	Cundinamarca	Colombia	4472	114	2129	0.0254919
239	Komi Republic	Russia	4416	39	2444	0.00883152
240	Jharkhand	India	4320	38	2485	0.0087963
241	Huanuco	Peru	4307	117	0	0.0271651
242	Lipetsk Oblast	Russia	4287	19	3056	0.004432
243	North Ossetia - Alania Republic	Russia	4287	67	3740	0.0156286
244	Tyumen Oblast	Russia	4286	21	2687	0.00489967
245	Coquimbo	Chile	4277	42	3508	0.00981997
246	Tver Oblast	Russia	4153	105	3544	0.0252829
247	Zakarpattia Oblast	Ukraine	4112	142	1379	0.0345331
248	Karachay-Cherkess Republic	Russia	4109	20	2548	0.00486736
249	La Rioja	Spain	4107	366	3107	0.0891161
250	Chihuahua	Mexico	4092	774	3128	0.18915
251	Astrakhan Oblast	Russia	4013	38	2800	0.00946923
252	Mordovia Republic	Russia	3938	29	2847	0.00736414
253	Kirov Oblast	Russia	3933	49	3509	0.0124587
254	Cajamarca	Peru	3857	134	0	0.034742
255	Uttarakhand	India	3785	50	2948	0.01321
256	Zabaykalsky Krai	Russia	3767	46	2887	0.0122113
257	Araucania	Chile	3651	49	3368	0.013421
258	Maine	US	3578	114	0	0.0318614
259	New South Wales	Australia	3527	49	2988	0.0138928
260	Tlaxcala	Mexico	3499	486	2725	0.138897
261	Brandenburg	Germany	3490	173	3300	0.0495702
262	Buryatia Republic	Russia	3481	27	2676	0.00775639
263	Tumbes	Peru	3477	129	0	0.0371009
264	Uppsala	Sweden	3472	238	0	0.0685484
265	Morelos	Mexico	3456	757	2507	0.219039
266	Novgorod Oblast	Russia	3450	56	1735	0.0162319
267	Ostergotland	Sweden	3426	224	0	0.0653824
268	Mari El Republic	Russia	3380	30	2925	0.00887574
269	Ingushetia Republic	Russia	3344	71	2529	0.0212321
270	Friuli Venezia Giulia	Italy	3339	345	2901	0.103324

	Province/State Name	Country	Number of Confirmed Cases	Number of Deaths	Number of Recoveries	Mortality Rate
271	Abruzzo	Italy	3331	467	2740	0.140198
272	Skane	Sweden	3330	258	0	0.0774775
273	Thuringen	Germany	3304	182	3100	0.0550847
274	Pskov Oblast	Russia	3291	39	2022	0.0118505
275	Schleswig-Holstein	Germany	3235	154	3000	0.0476043
276	Extremadura	Spain	3208	519	2652	0.161783
277	Campeche	Mexico	3182	320	2573	0.100566
278	Ivano-Frankivsk Oblast	Ukraine	3178	113	1246	0.0355557
279	Tomsk Oblast	Russia	3173	25	1771	0.00787898
280	British Columbia	Canada	3149	189	2753	0.0600191
281	Kiev Oblast	Ukraine	3128	58	2054	0.0185422
282	Volyn Oblast	Ukraine	3117	74	1840	0.0237408
283	Sicilia	Italy	3115	283	2695	0.0908507
284	Aguascalientes	Mexico	3083	193	2558	0.0626014
285	Nuble	Chile	3032	40	2536	0.0131926
286	Magdalena	Colombia	3001	214	1175	0.0713096
287	Gavleborg	Sweden	2989	147	0	0.0491803
288	Goa	India	2951	18	1674	0.00609963
289	Durango	Mexico	2924	215	2288	0.0735294
290	Kamchatka Krai	Russia	2840	29	1957	0.0102113
291	Saarland	Germany	2823	174	2592	0.0616366
292	Kharkiv Oblast	Ukraine	2787	111	1508	0.0398278
293	Mayotte	France	2743	37	2581	0.0134889
294	Arica y Parinacota	Chile	2735	31	2025	0.0113346
295	Sucre	Colombia	2723	170	607	0.0624311
296	P.A. Bolzano	Italy	2677	292	2288	0.109077
297	Queretaro	Mexico	2667	389	1954	0.145857
298	Kaliningrad Oblast	Russia	2648	42	1827	0.015861
299	Baja California Sur	Mexico	2603	108	1943	0.0414906
300	Kemerovo Oblast	Russia	2553	14	1326	0.00548374
301	Madre de Dios	Peru	2540	81	0	0.0318898
302	Vastmanland	Sweden	2537	175	0	0.0689791
303	Nayarit	Mexico	2527	264	1814	0.104472
304	Orebro	Sweden	2480	168	0	0.0677419

	Province/State Name	Country	Number of Confirmed Cases	Number of Deaths	Number of Recoveries	Mortality Rate
305	Canarias	Spain	2467	162	1537	0.0656668
306	Cusco	Peru	2467	29	0	0.0117552
307	Los Lagos	Chile	2456	24	1960	0.00977199
308	Khakassia Republic	Russia	2450	19	1844	0.0077551
309	Asturias	Spain	2442	334	1063	0.136773
310	Ayacucho	Peru	2442	39	0	0.0159705
311	Cantabria	Spain	2378	216	2287	0.0908326
312	Sormland	Sweden	2336	248	0	0.106164
313	Odessa Oblast	Ukraine	2314	30	1031	0.0129646
314	Unknown	Chile	2293	0	226958	0
315	Ternopil Oblast	Ukraine	2292	34	1446	0.0148342
316	Adygea Republic	Russia	2291	14	1843	0.00611087
317	Choco	Colombia	2280	81	1095	0.0355263
318	Baleares	Spain	2271	224	1533	0.098635
319	Tripura	India	2268	3	1604	0.00132275
320	Amur Oblast	Russia	2227	30	1234	0.013471
321	Kalmykia Republic	Russia	2220	27	1312	0.0121622
322	Vinnitsia Oblast	Ukraine	2194	38	1623	0.01732
323	Osaka	Japan	2126	86	1806	0.0404516
324	Montana	US	2096	34	0	0.0162214
325	Wyoming	US	1985	22	0	0.0110831
326	Vologda Oblast	Russia	1972	19	1535	0.00963489
327	Sachsen-Anhalt	Germany	1931	62	1815	0.0321077
328	Kostroma Oblast	Russia	1886	24	1256	0.0127253
329	Dalarna	Sweden	1884	163	0	0.086518
330	Udmurt Republic	Russia	1880	19	1394	0.0101064
331	Chechen Republic	Russia	1863	28	1275	0.0150295
332	Kanagawa	Japan	1832	98	1488	0.0534934
333	Karelia Republic	Russia	1792	3	990	0.00167411
334	Azad Jammu and Kashmir	Pakistan	1771	46	1061	0.025974
335	Murcia	Spain	1766	148	2180	0.0838052
336	Gilgit-Baltistan	Pakistan	1750	38	1389	0.0217143
337	Atacama	Chile	1722	10	1263	0.0058072
338	Bremen	Germany	1700	55	1616	0.0323529

	Province/State Name	Country	Number of Confirmed Cases	Number of Deaths	Number of Recoveries	Mortality Rate
339	Manipur	India	1700	0	989	0
340	Cordoba	Colombia	1695	224	408	0.132153
341	Meta	Colombia	1669	20	1093	0.0119832
342	Halland	Sweden	1657	77	0	0.0464695
343	Guangdong	China	1650	8	1636	0.00484848
344	Alaska	US	1631	17	0	0.0104231
345	Vasternorrland	Sweden	1628	122	0	0.0749386
346	Puducherry	India	1596	21	889	0.0131579
347	Hong Kong	China	1588	10	1241	0.00629723
348	Magallanes	Chile	1587	18	1467	0.0113422
349	Cesar	Colombia	1584	41	889	0.0258838
350	Saitama	Japan	1568	67	1103	0.0427296
351	Sakhalin Oblast	Russia	1568	0	564	0
352	Zhytomyr Oblast	Ukraine	1550	34	1204	0.0219355
353	Kurgan Oblast	Russia	1546	3	722	0.00194049
354	Zacatecas	Mexico	1516	147	1055	0.0969657
355	Tolima	Colombia	1511	33	594	0.0218398
356	Norrbotten	Sweden	1507	70	0	0.0464499
357	Moquegua	Peru	1480	33	0	0.0222973
358	Umbria	Italy	1452	80	1362	0.0550964
359	Tacna	Peru	1446	13	0	0.00899032
360	Santander	Colombia	1431	31	455	0.0216632
361	Pasco	Peru	1404	32	0	0.022792
362	Sardegna	Italy	1376	134	1230	0.0973837
363	Himachal Pradesh	India	1341	11	979	0.00820283
364	Vermont	US	1318	56	0	0.0424886
365	Hokkaido	Japan	1308	102	1132	0.0779817
366	Hawaii	US	1292	22	0	0.0170279
367	Henan	China	1276	22	1254	0.0172414
368	Zhejiang	China	1270	1	1267	0.000787402
369	Calabria	Italy	1218	97	1062	0.0796388
370	Puno	Peru	1197	38	0	0.031746
371	Valle d'Aosta	Italy	1196	146	1048	0.122074
372	Chiba	Japan	1193	46	930	0.0385583

	Province/State Name	Country	Number of Confirmed Cases	Number of Deaths	Number of Recoveries	Mortality Rate
373	Dnipropetrovsk Oblast	Ukraine	1155	24	1045	0.0207792
374	Huancavelica	Peru	1154	24	0	0.0207972
375	Ladakh	India	1142	1	964	0.000875657
376	Altai Republic	Russia	1139	1	650	0.000877963
377	Kronoberg	Sweden	1098	101	0	0.0919854
378	Jamtland Harjedalen	Sweden	1093	54	0	0.0494053
379	Magadan Oblast	Russia	1074	5	626	0.00465549
380	Queensland	Australia	1071	6	1058	0.00560224
381	Nova Scotia	Canada	1067	63	1002	0.059044
382	Hunan	China	1019	4	1015	0.00392542
383	Varmland	Sweden	1003	71	0	0.0707876
384	Anhui	China	991	6	985	0.00605449
385	La Guajira	Colombia	984	61	223	0.0619919
386	Colima	Mexico	980	118	643	0.120408
387	Heilongjiang	China	947	13	934	0.0137276
388	Fukuoka	Japan	933	33	822	0.0353698
389	Jiangxi	China	932	1	931	0.00107296
390	Beijing	China	929	9	752	0.00968784
391	Khmelnitskyi Oblast	Ukraine	922	22	778	0.0238612
392	Nagaland	India	902	0	348	0
393	Crimea Republic*	Ukraine	889	12	664	0.0134983
394	Saskatchewan	Canada	881	15	791	0.0170261
395	Cauca	Colombia	806	30	244	0.0372208
396	Mecklenburg-Vorpommern	Germany	805	20	784	0.0248447
397	Risaralda	Colombia	800	22	422	0.0275
398	Shandong	China	793	7	785	0.00882724
399	Vasterbotten	Sweden	784	31	0	0.0395408
400	Los Rios	Chile	781	11	718	0.0140845
401	Cherkasy Oblast	Ukraine	771	32	605	0.0415045
402	Donetsk Oblast	Ukraine	767	10	281	0.0130378
403	Hyogo	Japan	760	45	663	0.0592105
404	Kalmar	Sweden	734	58	0	0.0790191
405	Shanghai	China	732	7	691	0.00956284
406	Kirovohrad Oblast	Ukraine	674	33	593	0.0489614

	Province/State Name	Country	Number of Confirmed Cases	Number of Deaths	Number of Recoveries	Mortality Rate
407	Apurimac	Peru	668	23	0	0.0344311
408	Jiangsu	China	654	0	654	0
409	Western Australia	Australia	646	9	610	0.0139319
410	Norte de Santander	Colombia	645	30	235	0.0465116
411	Zaporizhia Oblast	Ukraine	631	20	526	0.0316957
412	Chandigarh	India	625	11	459	0.0176
413	Chernihiv Oblast	Ukraine	624	14	292	0.0224359
414	Reunion	France	608	3	472	0.00493421
415	Sichuan	China	599	3	590	0.00500835
416	Chongqing	China	583	6	576	0.0102916
417	Channel Islands	United Kingdom	581	47	512	0.080895
418	Aichi	Japan	552	34	483	0.0615942
419	Dadra and Nagar Haveli and Daman and Diu	India	520	1	310	0.00192308
420	Boyaca	Colombia	517	21	292	0.040619
421	Mykolaiv Oblast	Ukraine	494	12	389	0.0242915
422	Jewish Autonomous Okrug	Russia	491	6	403	0.01222
423	Kyoto	Japan	477	18	366	0.0377358
424	Blekinge	Sweden	475	14	0	0.0294737
425	Arunachal Pradesh	India	462	3	153	0.00649351
426	Molise	Italy	446	23	414	0.0515695
427	South Australia	Australia	444	4	439	0.00900901
428	Huila	Colombia	434	17	331	0.0391705
429	Port Quarantine	Japan	432	1	201	0.00231481
430	Basilicata	Italy	405	27	373	0.0666667
431	Fujian	China	363	1	361	0.00275482
432	Hebei	China	349	6	340	0.017192
433	Meghalaya	India	346	2	66	0.00578035
434	Sumy Oblast	Ukraine	346	6	300	0.017341
435	Poltava Oblast	Ukraine	337	13	305	0.0385757
436	Isle of Man	United Kingdom	336	24	312	0.0714286
437	Manitoba	Canada	330	7	318	0.0212121
438	Shaanxi	China	321	3	315	0.00934579

	Province/State Name	Country	Number of Confirmed Cases	Number of Deaths	Number of Recoveries	Mortality Rate
439	Guam	US	313	5	0	0.0159744
440	Caldas	Colombia	305	8	226	0.0262295
441	Ishikawa	Japan	300	27	267	0.09
442	Sevastopol*	Ukraine	283	5	210	0.0176678
443	Newfoundland and Labrador	Canada	262	3	258	0.0114504
444	Martinique	France	255	15	98	0.0588235
445	Guangxi	China	254	2	252	0.00787402
446	Nenets Autonomous Okrug	Russia	254	0	44	0
447	Inner Mongolia	China	249	1	237	0.00401606
448	Virgin Islands	US	243	6	0	0.0246914
449	Mizoram	India	238	0	159	0
450	Toyama	Japan	230	22	205	0.0956522
451	Tasmania	Australia	228	13	215	0.0570175
452	Sikkim	India	220	0	87	0
453	Ibaraki	Japan	209	10	171	0.0478469
454	Kherson Oblast	Ukraine	205	3	186	0.0146341
455	Quindio	Colombia	204	6	145	0.0294118
456	Cayman Islands	United Kingdom	203	1	200	0.00492611
457	Tianjin	China	203	3	195	0.0147783
458	Shanxi	China	201	0	198	0
459	Guadeloupe	France	190	14	157	0.0736842
460	Hiroshima	Japan	190	3	166	0.0157895
461	Gotland	Sweden	189	6	0	0.031746
462	Faroe Islands	Denmark	188	0	188	0
463	Yunnan	China	188	2	183	0.0106383
464	Gibraltar	United Kingdom	180	0	180	0
465	Andaman and Nicobar Islands	India	176	0	130	0
466	Hainan	China	171	6	165	0.0350877
467	New Brunswick	Canada	168	2	163	0.0119048
468	Gansu	China	167	2	165	0.011976
469	Gifu	Japan	166	7	148	0.0421687
470	Ceuta	Spain	164	4	163	0.0243902

	Province/State Name	Country	Number of Confirmed Cases	Number of Deaths	Number of Recoveries	Mortality Rate
471	Liaoning	China	164	2	150	0.0121951
472	Gunma	Japan	161	19	134	0.118012
473	Kagoshima	Japan	158	0	11	0
474	Jilin	China	155	2	153	0.0129032
475	Okinawa	Japan	152	7	139	0.0460526
476	Bermuda	United Kingdom	150	9	137	0.06
477	Arauca	Colombia	147	0	62	0
478	Guizhou	China	147	2	145	0.0136054
479	Chukotka Autonomous Okrug	Russia	145	1	141	0.00689655
480	Casanare	Colombia	131	1	61	0.00763359
481	Nara	Japan	131	2	91	0.0152672
482	Melilla	Spain	128	2	125	0.015625
483	Fukui	Japan	125	8	114	0.064
484	Caqueta	Colombia	124	3	25	0.0241935
485	Grand Princess	US	116	3	13	0.0258621
486	Australian Capital Territory	Australia	113	3	105	0.0265487
487	Putumayo	Colombia	113	10	13	0.0884956
488	Miyagi	Japan	112	1	94	0.00892857
489	Shiga	Japan	107	1	100	0.00934579
490	Tochigi	Japan	107	0	82	0
491	Aruba	Netherlands	106	3	99	0.0283019
492	Shizuoka	Japan	98	1	82	0.0102041
493	Luhansk Oblast	Ukraine	97	1	74	0.0103093
494	Niigata	Japan	88	0	83	0
495	Fukushima	Japan	84	0	82	0
496	Wakayama	Japan	84	3	61	0.0357143
497	Ehime	Japan	82	5	77	0.0609756
498	Nagano	Japan	79	0	76	0
499	Sint Maarten	Netherlands	78	15	63	0.192308
500	Kochi	Japan	76	3	71	0.0394737
501	Xinjiang	China	76	3	73	0.0394737
502	Yamanashi	Japan	76	1	74	0.0131579
503	Ningxia	China	75	0	75	0

	Province/State Name	Country	Number of Confirmed Cases	Number of Deaths	Number of Recoveries	Mortality Rate
504	Yamagata	Japan	74	1	68	0.0135135
505	Turks and Caicos Islands	United Kingdom	72	2	12	0.0277778
506	French Polynesia	France	62	0	60	0
507	Aysen	Chile	60	0	46	0
508	Oita	Japan	60	1	59	0.0166667
509	Vaupes	Colombia	59	1	21	0.0169492
510	Mie	Japan	54	1	45	0.0185185
511	Diamond Princess	Canada	49	1	0	0.0204082
512	Kumamoto	Japan	48	3	45	0.0625
513	Macau	China	46	0	45	0
514	Saga	Japan	45	0	45	0
515	St Martin	France	43	3	37	0.0697674
516	Guaviare	Colombia	42	0	37	0
517	Yamaguchi	Japan	37	0	37	0
518	Northern Mariana Islands	US	36	2	0	0.0555556
519	Prince Edward Island	Canada	36	0	27	0
520	Aomori	Japan	34	1	26	0.0294118
521	Kagawa	Japan	32	0	28	0
522	Nagasaki	Japan	32	1	17	0.03125
523	Okayama	Japan	32	0	26	0
524	Northern Territory	Australia	31	0	29	0
525	San Andres y Providencia	Colombia	29	1	21	0.0344828
526	Curacao	Netherlands	26	1	24	0.0384615
527	Shimane	Japan	25	0	24	0
528	New Caledonia	France	22	0	21	0
529	Miyazaki	Japan	20	0	17	0
530	Qinghai	China	18	0	18	0
531	Akita	Japan	16	0	16	0
532	Guainia	Colombia	14	1	11	0.0714286
533	Falkland Islands (Malvinas)	United Kingdom	13	0	13	0
534	Greenland	Denmark	13	0	13	0
535	Montserrat	United Kingdom	12	1	10	0.0833333

	Province/State Name	Country	Number of Confirmed Cases	Number of Deaths	Number of Recoveries	Mortality Rate
536	Yukon	Canada	11	0	11	0
537	Tokushima	Japan	10	1	4	0.1
538	Bonaire, Sint Eustatius and Saba	Netherlands	9	0	7	0
539	British Virgin Islands	United Kingdom	8	1	7	0.125
540	Saint Barthelemy	France	6	0	6	0
541	Northwest Territories	Canada	5	0	5	0
542	Tottori	Japan	5	0	3	0
543	Anguilla	United Kingdom	3	0	3	0
544	Saint Pierre and Miquelon	France	2	0	1	0
545	Tibet	China	1	0	1	0
546	Vichada	Colombia	1	0	1	0

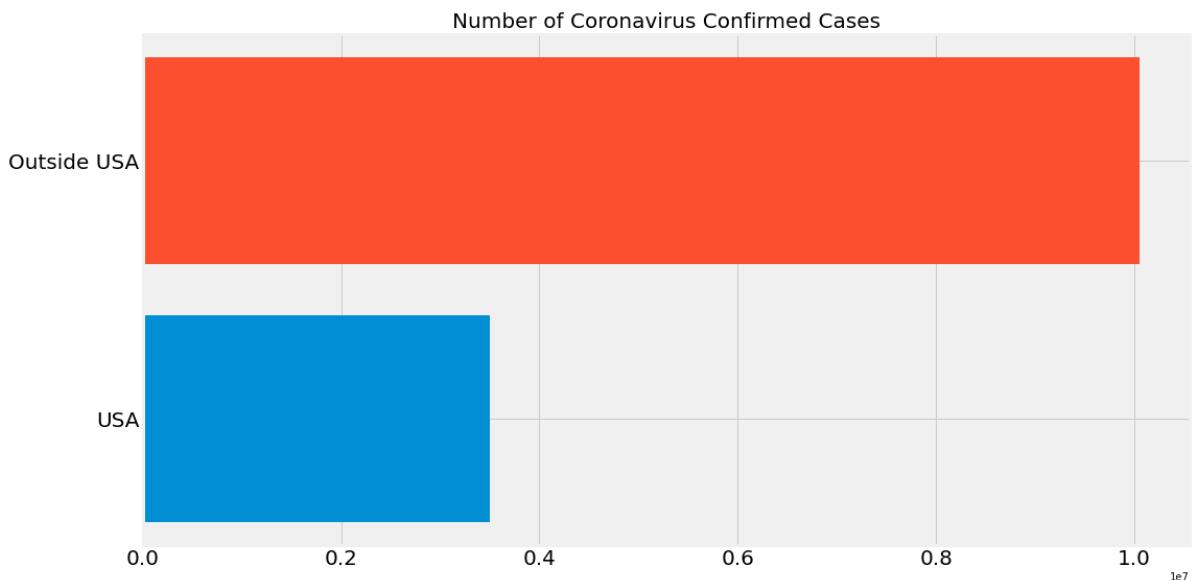
```
In [ ]: # Dealing with missing values
nan_indices = []
```

```
# handle nan if there is any, it is usually a float: float('nan')
for i in range(len(unique_provinces)):
    if type(unique_provinces[i]) == float:
        nan_indices.append(i)

unique_provinces = list(unique_provinces)
confirmed_by_province = list(confirmed_by_province)

for i in nan_indices:
    unique_provinces.pop(i)
    confirmed_by_province.pop(i)
```

```
In [ ]: USA_confirmed = latest_data[latest_data['Country_Region']=='US']['Confirmed'].sum()
outside_USA_confirmed = np.sum(confirmed_by_country) - USA_confirmed
plt.figure(figsize=(16, 9))
plt.barh('USA', USA_confirmed)
plt.barh('Outside USA', outside_USA_confirmed)
plt.title('Number of Coronavirus Confirmed Cases', size=20)
plt.xticks(size=20)
plt.yticks(size=20)
plt.show()
```



```
In [ ]: print('Outside USA: {} cases'.format(outside_USA_confirmed))
print('USA: {} cases'.format(USA_confirmed))
print('Total: {} cases'.format(USA_confirmed+outside_USA_confirmed))
```

```
Outside USA: 10056630 cases
USA: 3497847 cases
Total: 13554477 cases
```

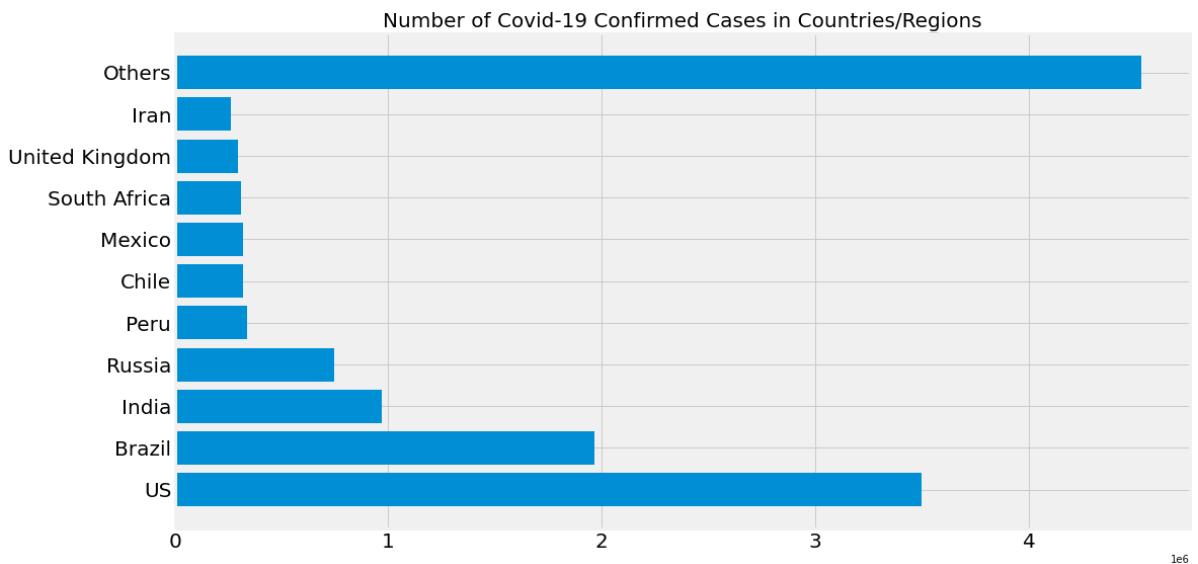
```
In [ ]: # Only show 10 countries with the most confirmed cases, the rest are grouped into 'Others'
visual_unique_countries = []
visual_confirmed_cases = []
others = np.sum(confirmed_by_country[10:])

for i in range(len(confirmed_by_country[:10])):
    visual_unique_countries.append(unique_countries[i])
    visual_confirmed_cases.append(confirmed_by_country[i])

visual_unique_countries.append('Others')
visual_confirmed_cases.append(others)
```

```
In [ ]: def plot_bar_graphs(x, y, title):
    plt.figure(figsize=(16, 9))
    plt.barh(x, y)
    plt.title(title, size=20)
    plt.xticks(size=20)
    plt.yticks(size=20)
    plt.show()
```

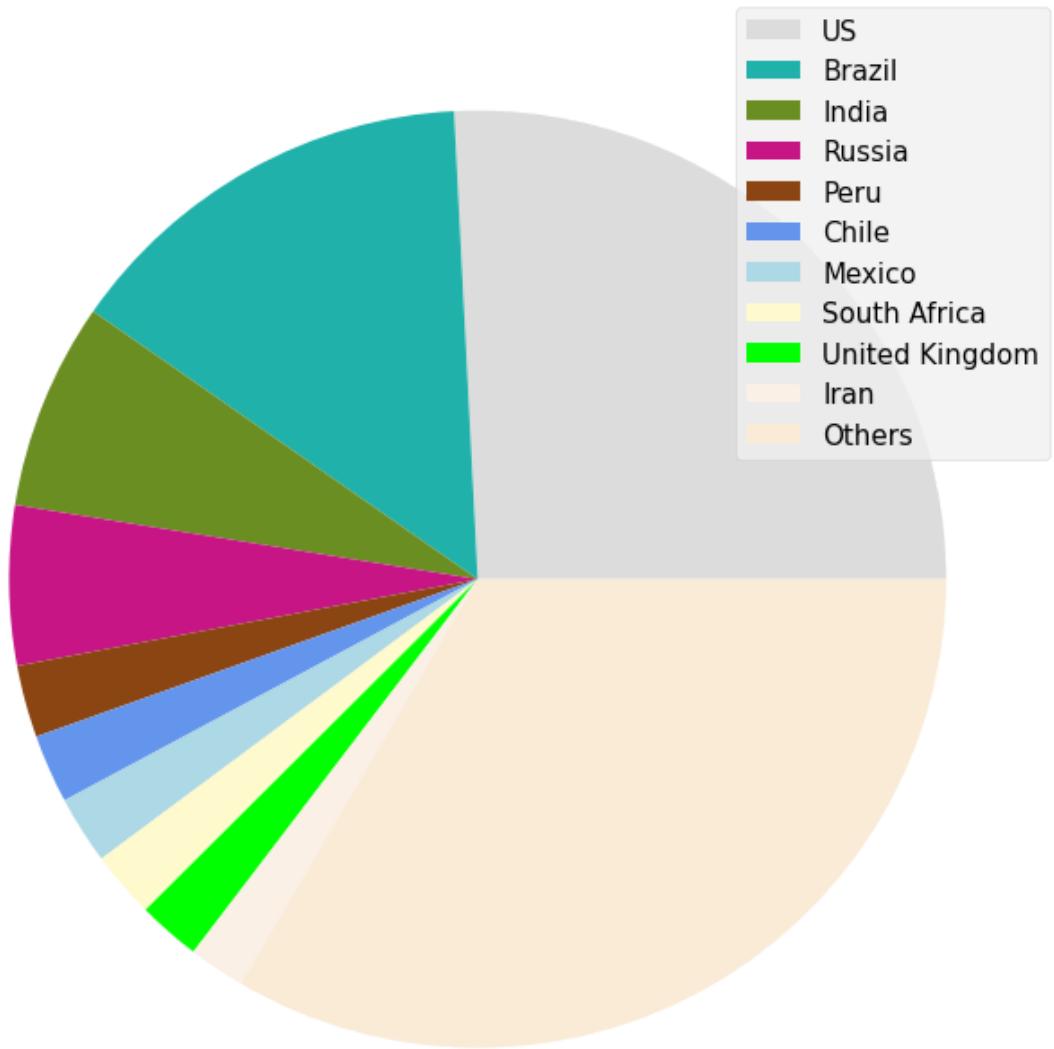
```
In [ ]: plot_bar_graphs(visual_unique_countries, visual_confirmed_cases, 'Number of Covid-19 Confirmed Cases')
```



```
In [ ]: def plot_pie_charts(x, y, title):
    c = random.choices(list(mcolors.CSS4_COLORS.values()), k = len(unique_countries))
    plt.figure(figsize=(12,12))
    plt.title(title, size=20)
    plt.pie(y, colors=c)
    plt.legend(x, loc='best', fontsize=15)
    plt.show()
```

```
In [ ]: plot_pie_charts(visual_unique_countries, visual_confirmed_cases, 'Covid-19 Confirmed Cases in Countries/Regions')
```

## Covid-19 Confirmed Cases per Country



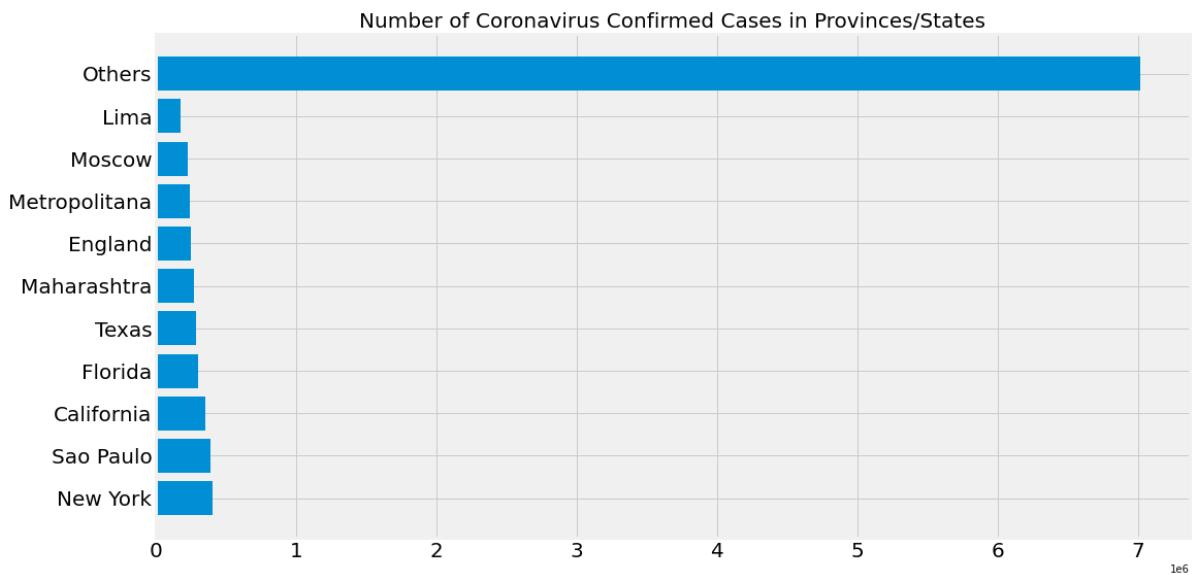
```
In [ ]:
```

```
# Only show 10 provinces with the most confirmed cases, the rest are grouped into 'Others'
visual_unique_provinces = []
visual_confirmed_cases2 = []
others = np.sum(confirmed_by_province[10:])

for i in range(len(confirmed_by_province[:10])):
    visual_unique_provinces.append(unique_provinces[i])
    visual_confirmed_cases2.append(confirmed_by_province[i])

visual_unique_provinces.append('Others')
visual_confirmed_cases2.append(others)
```

```
In [ ]: plot_bar_graphs(visual_unique_provinces, visual_confirmed_cases2, 'Number of Corona
```



```
In [ ]: def plot_pie_country_with_regions(country_name, title):
regions = list(latest_data[latest_data['Country_Region']==country_name]['Province_State'])
confirmed_cases = []
no_cases = []

for i in regions:
    cases = latest_data[latest_data['Province_State']==i]['Confirmed'].sum()
    if cases > 0:
        confirmed_cases.append(cases)
    else:
        no_cases.append(i)

# remove areas with no confirmed cases
for i in no_cases:
    regions.remove(i)

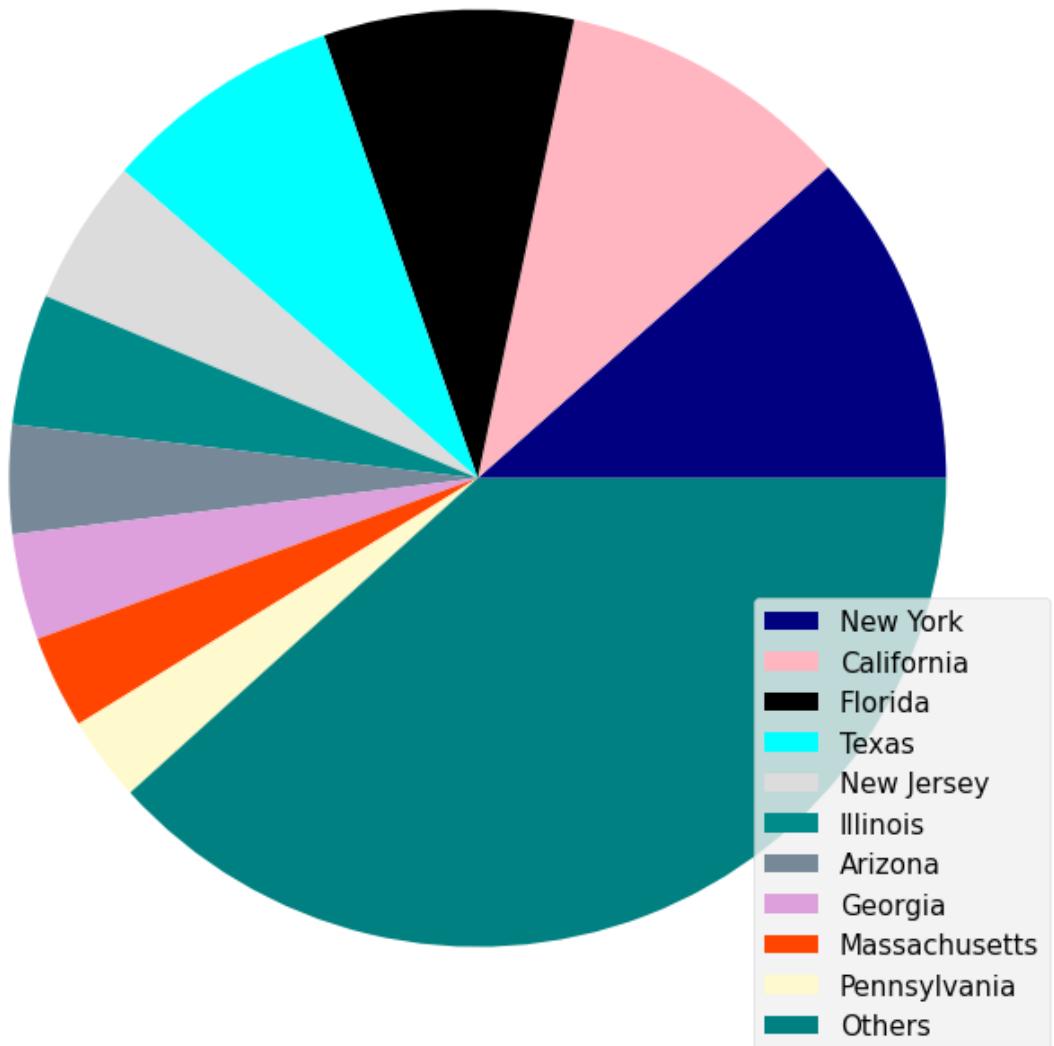
# only show the top 10 states
regions = [k for k, v in sorted(zip(regions, confirmed_cases), key=operator.itemgetter(1))[:10]]

for i in range(len(regions)):
    confirmed_cases[i] = latest_data[latest_data['Province_State']==regions[i]]['Confirmed'].sum()

# additional province/state will be considered "others"
if(len(regions)>10):
    regions_10 = regions[:10]
    regions_10.append('Others')
    confirmed_cases_10 = confirmed_cases[:10]
    confirmed_cases_10.append(np.sum(confirmed_cases[10:]))
    plot_pie_charts(regions_10,confirmed_cases_10, title)
else:
    plot_pie_charts(regions,confirmed_cases, title)
```

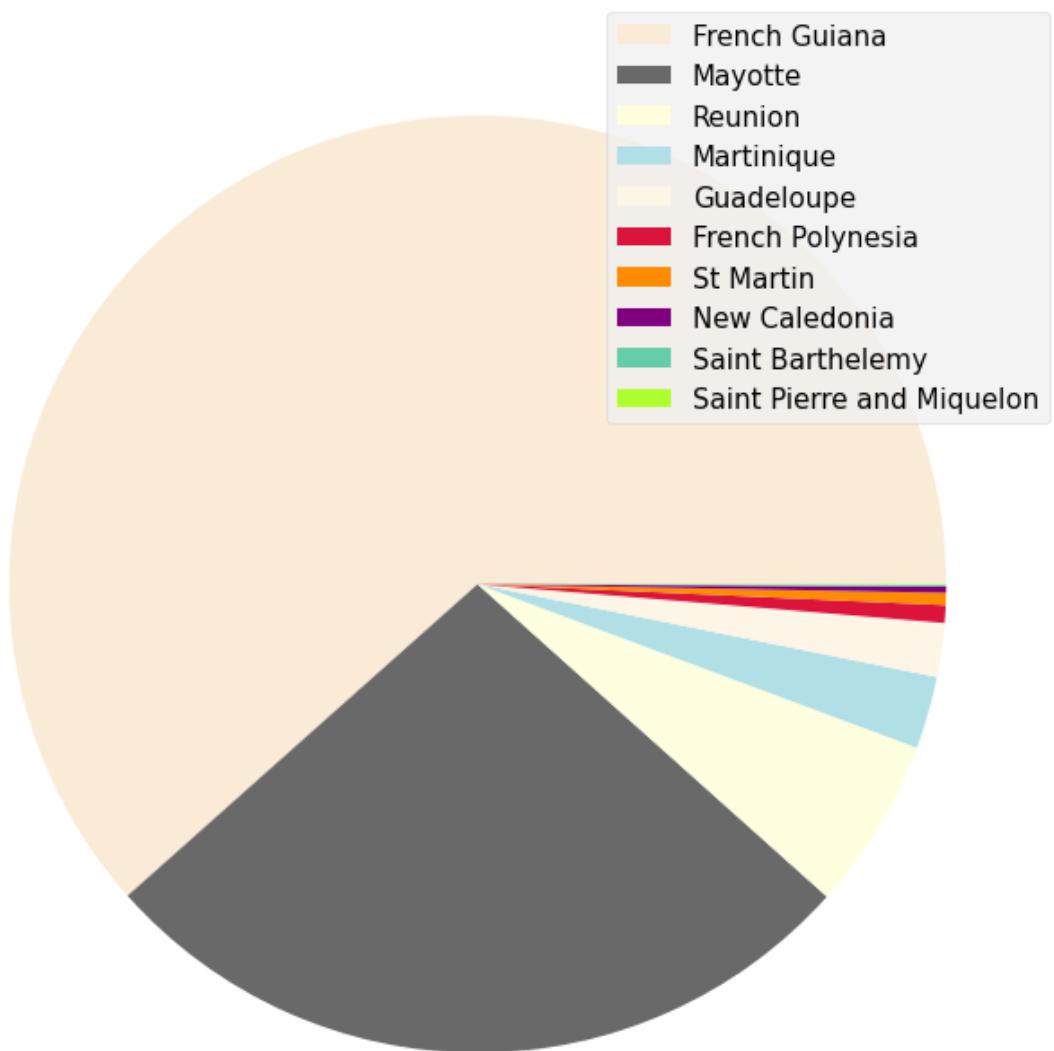
```
In [ ]: plot_pie_country_with_regions('US', 'COVID-19 Confirmed Cases in the United States')
```

## COVID-19 Confirmed Cases in the United States



```
In [ ]: plot_pie_country_with_regions('France', 'COVID-19 Confirmed Cases in the France')
```

## COVID-19 Confirmed Cases in the France



```
In [ ]: # Predicting the future
```

```
In [ ]: days_since_1_22 = np.array([i for i in range(len(dates))]).reshape(-1, 1)
world_cases = np.array(world_cases).reshape(-1, 1)
total_deaths = np.array(total_deaths).reshape(-1, 1)
total_recovered = np.array(total_recovered).reshape(-1, 1)
```

```
In [ ]: days_in_future = 20
future_forecast = np.array([i for i in range(len(dates)+days_in_future)]).reshape(-1, 1)
adjusted_dates = future_forecast[:-20]
```

```
In [ ]: future_forecast
```

```
Out[ ]: array([[ 0],  
 [ 1],  
 [ 2],  
 [ 3],  
 [ 4],  
 [ 5],  
 [ 6],  
 [ 7],  
 [ 8],  
 [ 9],  
 [ 10],  
 [ 11],  
 [ 12],  
 [ 13],  
 [ 14],  
 [ 15],  
 [ 16],  
 [ 17],  
 [ 18],  
 [ 19],  
 [ 20],  
 [ 21],  
 [ 22],  
 [ 23],  
 [ 24],  
 [ 25],  
 [ 26],  
 [ 27],  
 [ 28],  
 [ 29],  
 [ 30],  
 [ 31],  
 [ 32],  
 [ 33],  
 [ 34],  
 [ 35],  
 [ 36],  
 [ 37],  
 [ 38],  
 [ 39],  
 [ 40],  
 [ 41],  
 [ 42],  
 [ 43],  
 [ 44],  
 [ 45],  
 [ 46],  
 [ 47],  
 [ 48],  
 [ 49],  
 [ 50],  
 [ 51],  
 [ 52],  
 [ 53],  
 [ 54],  
 [ 55],  
 [ 56],  
 [ 57],  
 [ 58],  
 [ 59],  
 [ 60],  
 [ 61],  
 [ 62],  
 [ 63],
```

[ 64],  
[ 65],  
[ 66],  
[ 67],  
[ 68],  
[ 69],  
[ 70],  
[ 71],  
[ 72],  
[ 73],  
[ 74],  
[ 75],  
[ 76],  
[ 77],  
[ 78],  
[ 79],  
[ 80],  
[ 81],  
[ 82],  
[ 83],  
[ 84],  
[ 85],  
[ 86],  
[ 87],  
[ 88],  
[ 89],  
[ 90],  
[ 91],  
[ 92],  
[ 93],  
[ 94],  
[ 95],  
[ 96],  
[ 97],  
[ 98],  
[ 99],  
[100],  
[101],  
[102],  
[103],  
[104],  
[105],  
[106],  
[107],  
[108],  
[109],  
[110],  
[111],  
[112],  
[113],  
[114],  
[115],  
[116],  
[117],  
[118],  
[119],  
[120],  
[121],  
[122],  
[123],  
[124],  
[125],  
[126],  
[127],

[128],  
[129],  
[130],  
[131],  
[132],  
[133],  
[134],  
[135],  
[136],  
[137],  
[138],  
[139],  
[140],  
[141],  
[142],  
[143],  
[144],  
[145],  
[146],  
[147],  
[148],  
[149],  
[150],  
[151],  
[152],  
[153],  
[154],  
[155],  
[156],  
[157],  
[158],  
[159],  
[160],  
[161],  
[162],  
[163],  
[164],  
[165],  
[166],  
[167],  
[168],  
[169],  
[170],  
[171],  
[172],  
[173],  
[174],  
[175],  
[176],  
[177],  
[178],  
[179],  
[180],  
[181],  
[182],  
[183],  
[184],  
[185],  
[186],  
[187],  
[188],  
[189],  
[190],  
[191],

```
[192],  
[193],  
[194],  
[195]))
```

```
In [ ]: start = '1/22/2020'  
start_date = datetime.datetime.strptime(start, '%m/%d/%Y')  
future_forecast_dates = []  
for i in range(len(future_forecast)):  
    future_forecast_dates.append((start_date + datetime.timedelta(days=i)).strftime('%m/%d/%Y'))
```

```
In [ ]: X_train_confirmed, X_test_confirmed, y_train_confirmed, y_test_confirmed = train_test_split(X, y, test_size=0.2, random_state=42)
```

```
In [ ]: # transform data for polynomial regression  
poly = PolynomialFeatures(degree=3)  
poly_X_train_confirmed = poly.fit_transform(X_train_confirmed)  
poly_X_test_confirmed = poly.fit_transform(X_test_confirmed)  
poly_future_forecast = poly.fit_transform(future_forecast)
```

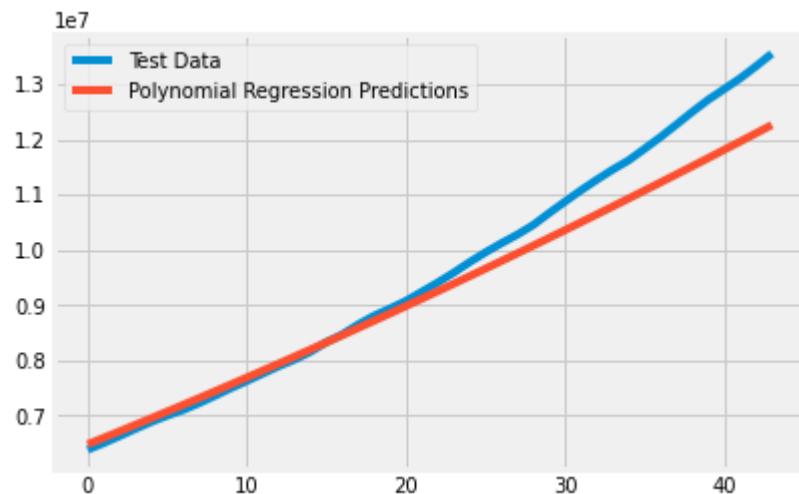
```
In [ ]: # polynomial regression  
linear_model = LinearRegression(normalize=True, fit_intercept=False)  
linear_model.fit(poly_X_train_confirmed, y_train_confirmed)  
test_linear_pred = linear_model.predict(poly_X_test_confirmed)  
linear_pred = linear_model.predict(poly_future_forecast)  
print('MAE:', mean_absolute_error(test_linear_pred, y_test_confirmed))  
print('MSE:', mean_squared_error(test_linear_pred, y_test_confirmed))
```

```
MAE: 370757.7867551611
```

```
MSE: 287366330493.7956
```

```
In [ ]: plt.plot(y_test_confirmed)  
plt.plot(test_linear_pred)  
plt.legend(['Test Data', 'Polynomial Regression Predictions'])
```

```
Out[ ]: <matplotlib.legend.Legend at 0x7facc0228710>
```



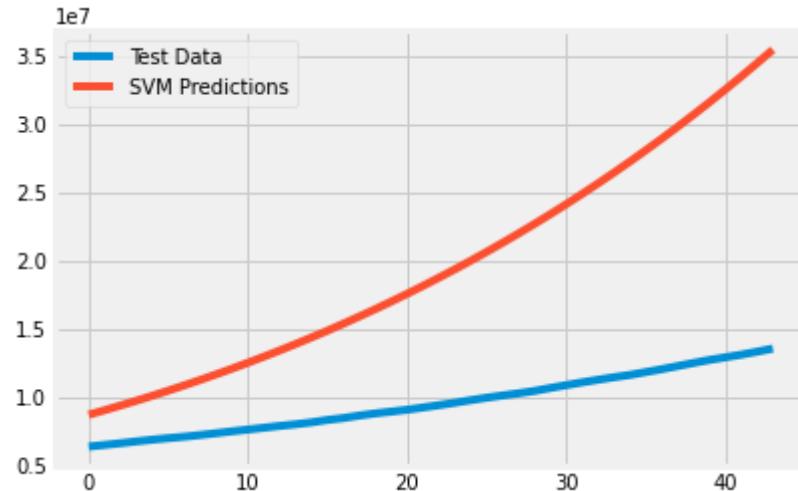
```
In [ ]: # svm_confirmed = svm_search.best_estimator_  
svm_confirmed = SVR(shrinking=True, kernel='poly', gamma=0.01, epsilon=1, degree=5, C=1000)  
svm_confirmed.fit(X_train_confirmed, y_train_confirmed)  
svm_pred = svm_confirmed.predict(future_forecast)
```

```
/opt/venv/lib/python3.7/site-packages/sklearn/utils/validation.py:73: DataConversionWarning:  
A column-vector y was passed when a 1d array was expected. Please change the shape of y to (n_samples, ), for example using ravel().  
return f(**kwargs)
```

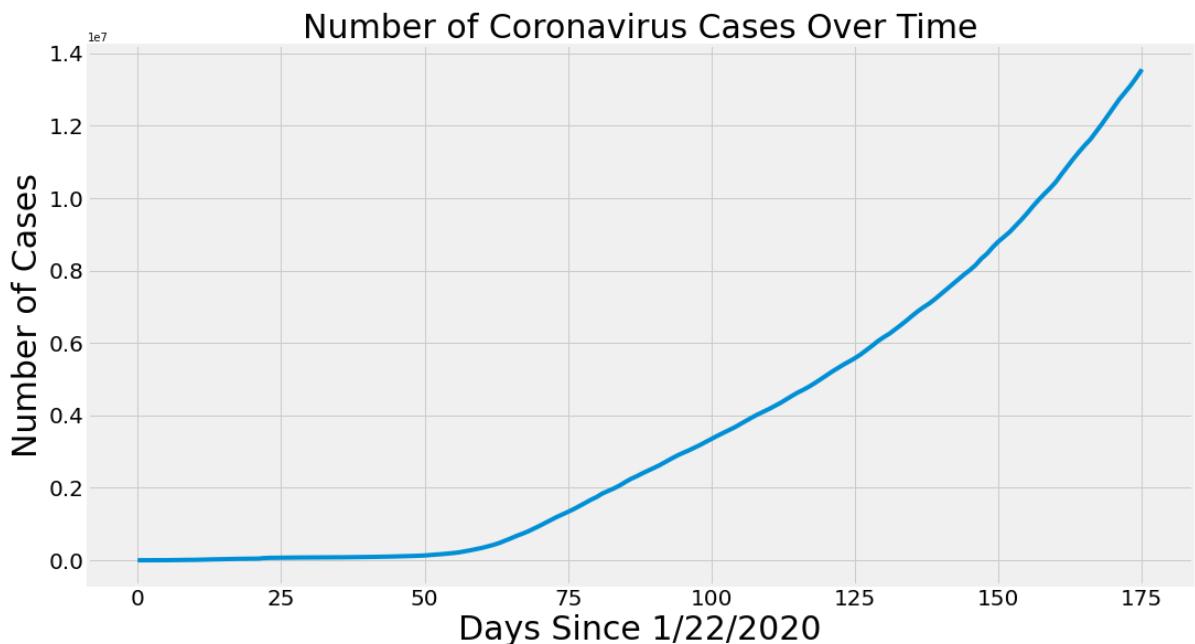
```
In [ ]: svm_test_pred = svm_confirmed.predict(X_test_confirmed)
plt.plot(y_test_confirmed)
plt.plot(svm_test_pred)
plt.legend(['Test Data', 'SVM Predictions'])
print('MAE:', mean_absolute_error(svm_test_pred, y_test_confirmed))
print('MSE:', mean_squared_error(svm_test_pred, y_test_confirmed))
```

MAE: 10132321.457088307

MSE: 135669512706535.75

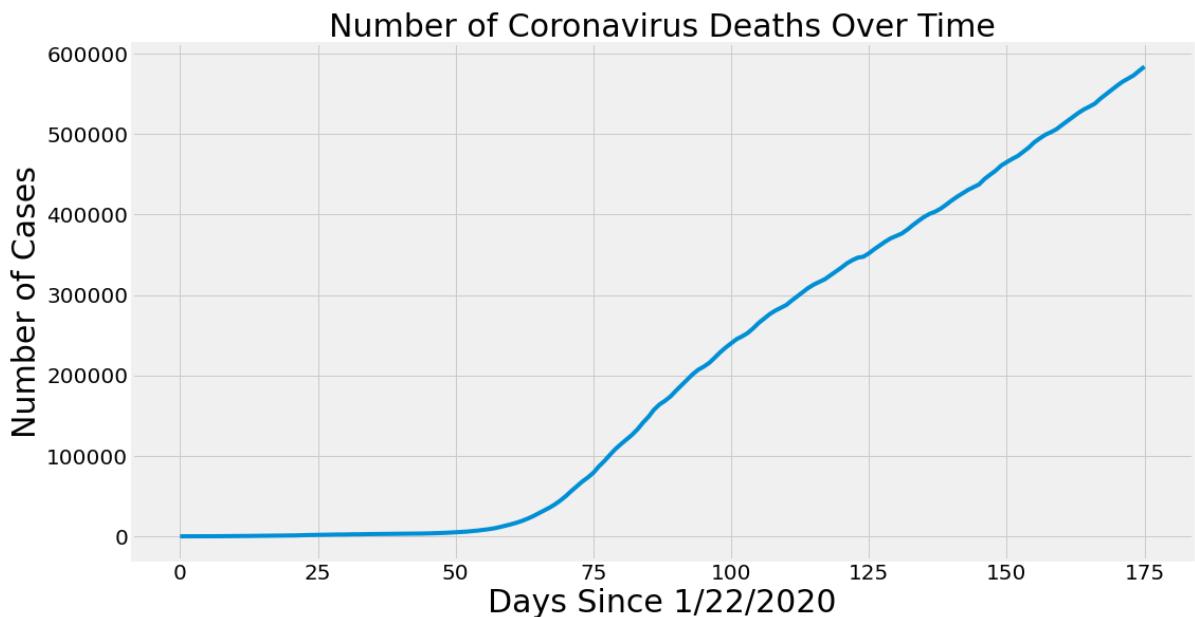


```
In [ ]: adjusted_dates = adjusted_dates.reshape(1, -1)[0]
plt.figure(figsize=(16, 9))
plt.plot(adjusted_dates, world_cases)
plt.title('Number of Coronavirus Cases Over Time', size=30)
plt.xlabel('Days Since 1/22/2020', size=30)
plt.ylabel('Number of Cases', size=30)
plt.xticks(size=20)
plt.yticks(size=20)
plt.show()
```

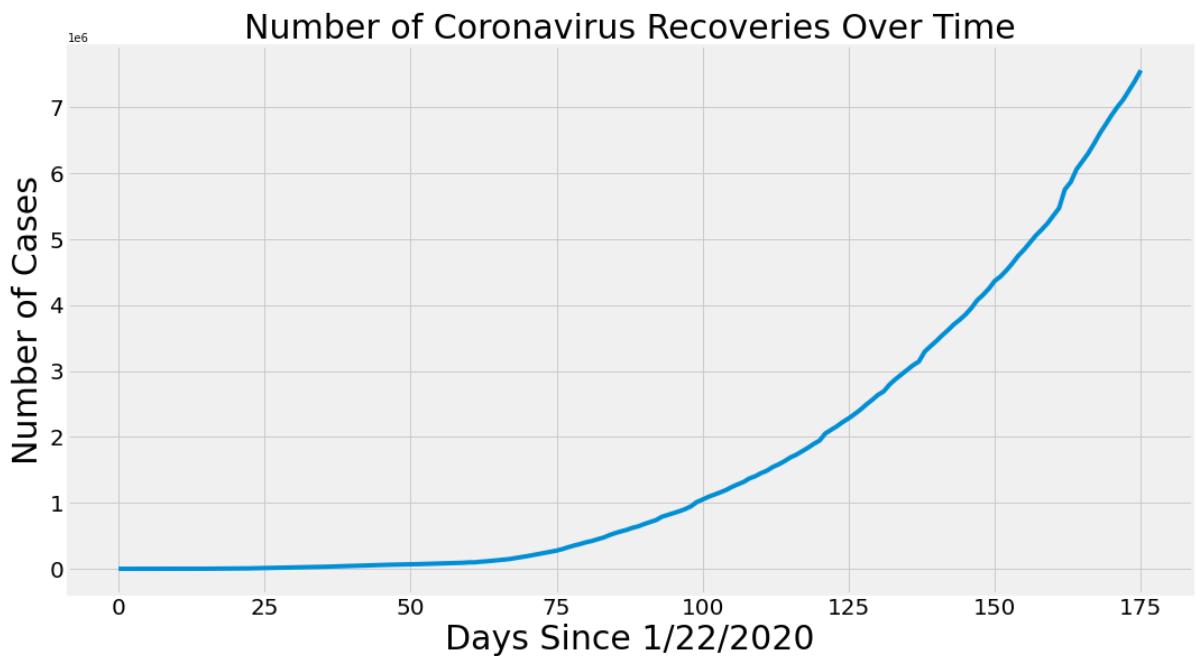


```
In [ ]: plt.figure(figsize=(16, 9))
plt.plot(adjusted_dates, total_deaths)
plt.title('Number of Coronavirus Deaths Over Time', size=30)
plt.xlabel('Days Since 1/22/2020', size=30)
plt.ylabel('Number of Cases', size=30)
plt.xticks(size=20)
```

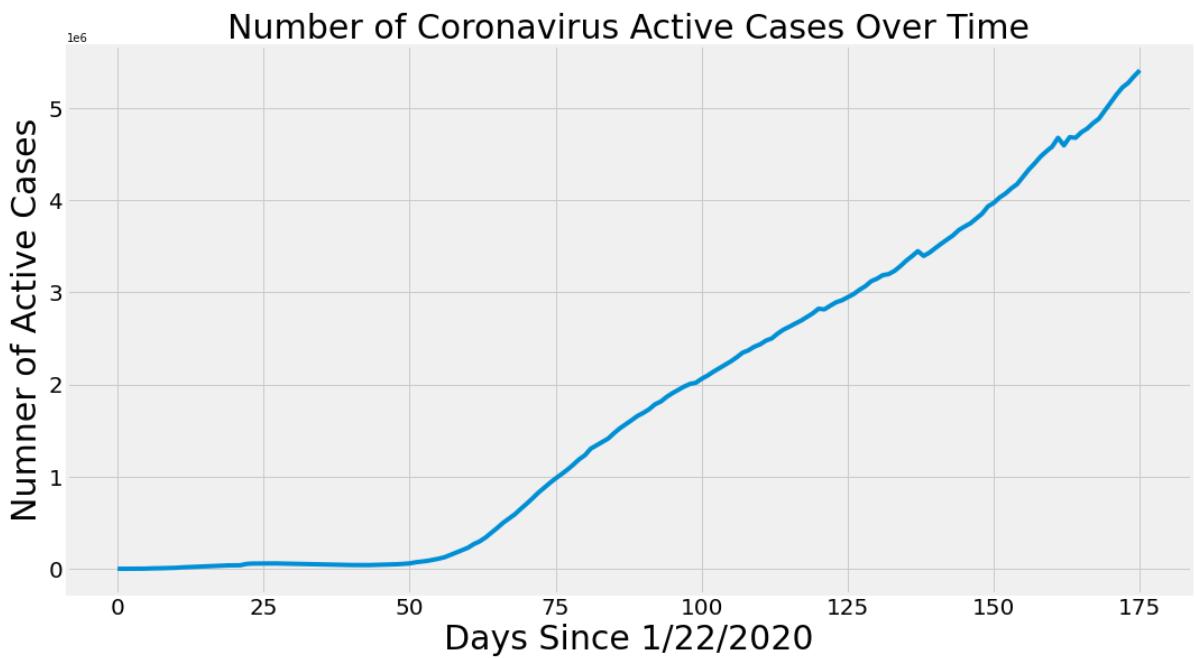
```
plt.yticks(size=20)  
plt.show()
```



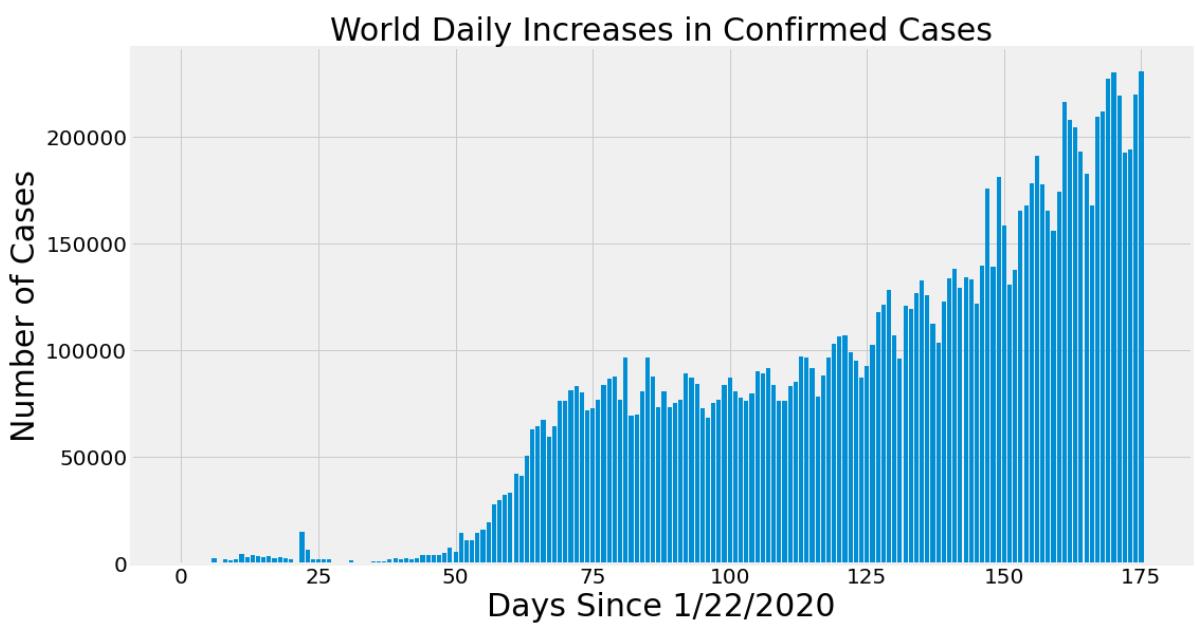
```
In [ ]: plt.figure(figsize=(16, 9))  
plt.plot(adjusted_dates, total_recovered)  
plt.title('Number of Coronavirus Recoveries Over Time', size=30)  
plt.xlabel('Days Since 1/22/2020', size=30)  
plt.ylabel('Number of Cases', size=30)  
plt.xticks(size=20)  
plt.yticks(size=20)  
plt.show()
```



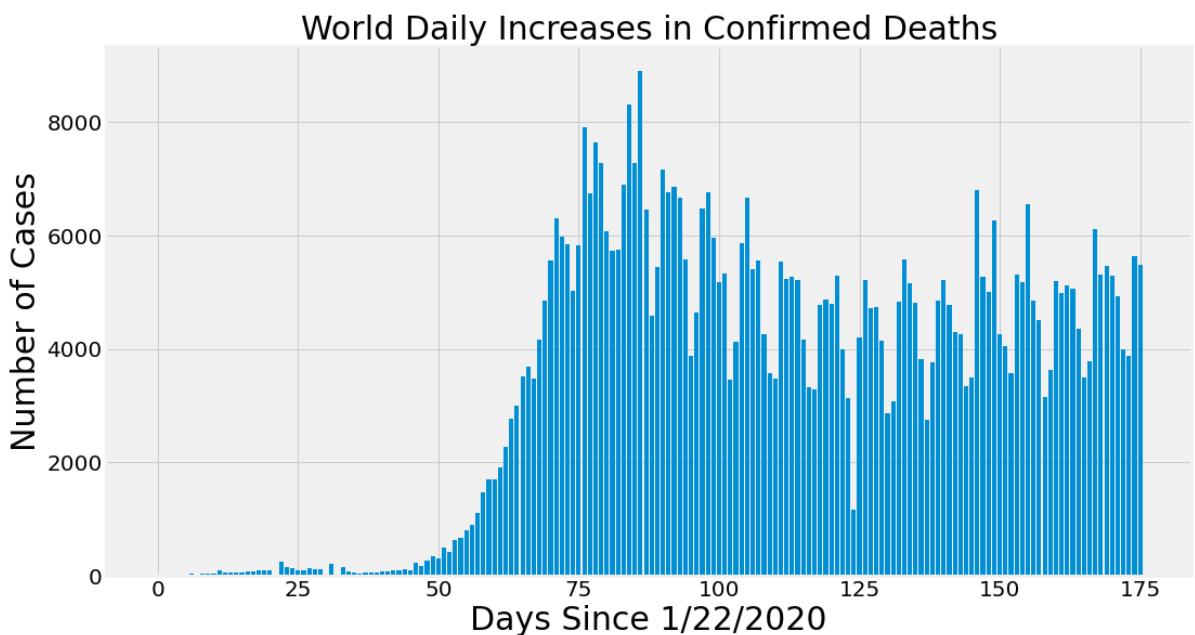
```
In [ ]: plt.figure(figsize=(16, 9))  
plt.plot(adjusted_dates, total_active)  
plt.title('Number of Coronavirus Active Cases Over Time', size=30)  
plt.xlabel('Days Since 1/22/2020', size=30)  
plt.ylabel('Number of Active Cases', size=30)  
plt.xticks(size=20)  
plt.yticks(size=20)  
plt.show()
```



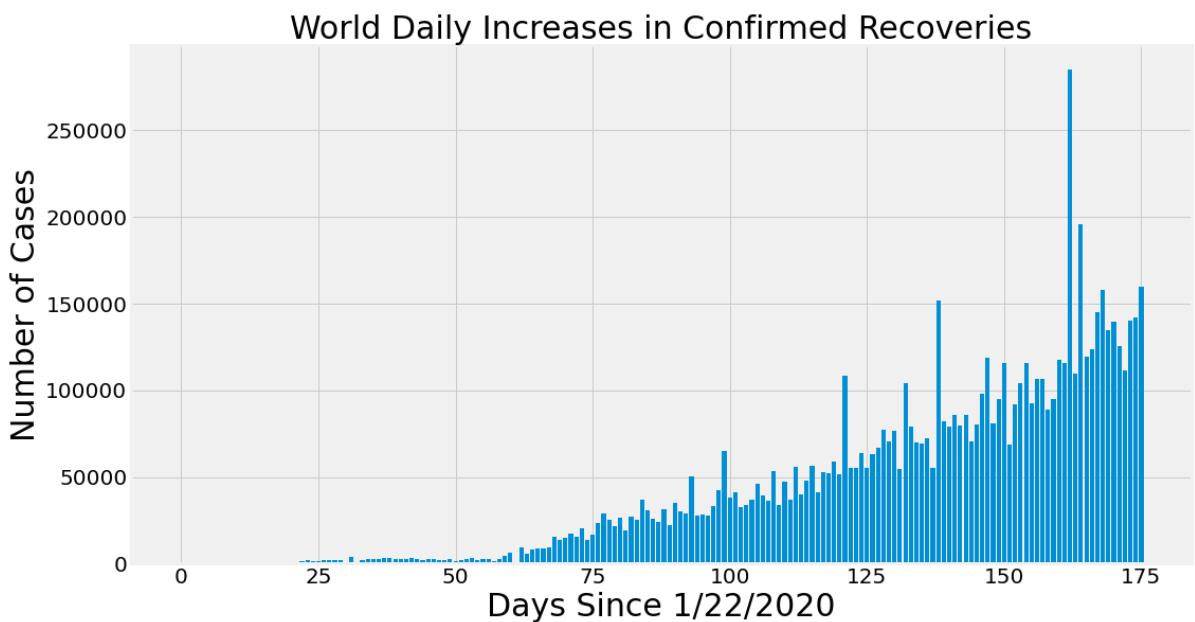
```
In [ ]: plt.figure(figsize=(16, 9))
plt.bar(adjusted_dates, world_daily_increase)
plt.title('World Daily Increases in Confirmed Cases', size=30)
plt.xlabel('Days Since 1/22/2020', size=30)
plt.ylabel('Number of Cases', size=30)
plt.xticks(size=20)
plt.yticks(size=20)
plt.show()
```



```
In [ ]: plt.figure(figsize=(16, 9))
plt.bar(adjusted_dates, world_daily_death)
plt.title('World Daily Increases in Confirmed Deaths', size=30)
plt.xlabel('Days Since 1/22/2020', size=30)
plt.ylabel('Number of Cases', size=30)
plt.xticks(size=20)
plt.yticks(size=20)
plt.show()
```

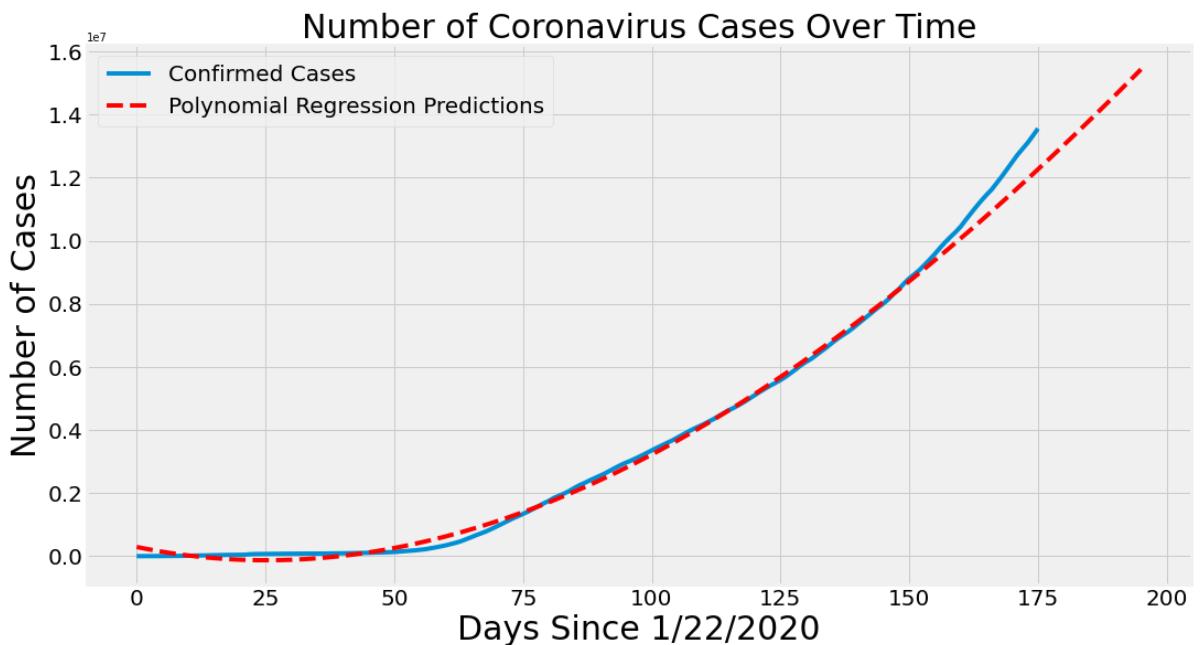


```
In [ ]: plt.figure(figsize=(16, 9))
plt.bar(adjusted_dates, world_daily_recovery)
plt.title('World Daily Increases in Confirmed Recoveries', size=30)
plt.xlabel('Days Since 1/22/2020', size=30)
plt.ylabel('Number of Cases', size=30)
plt.xticks(size=20)
plt.yticks(size=20)
plt.show()
```

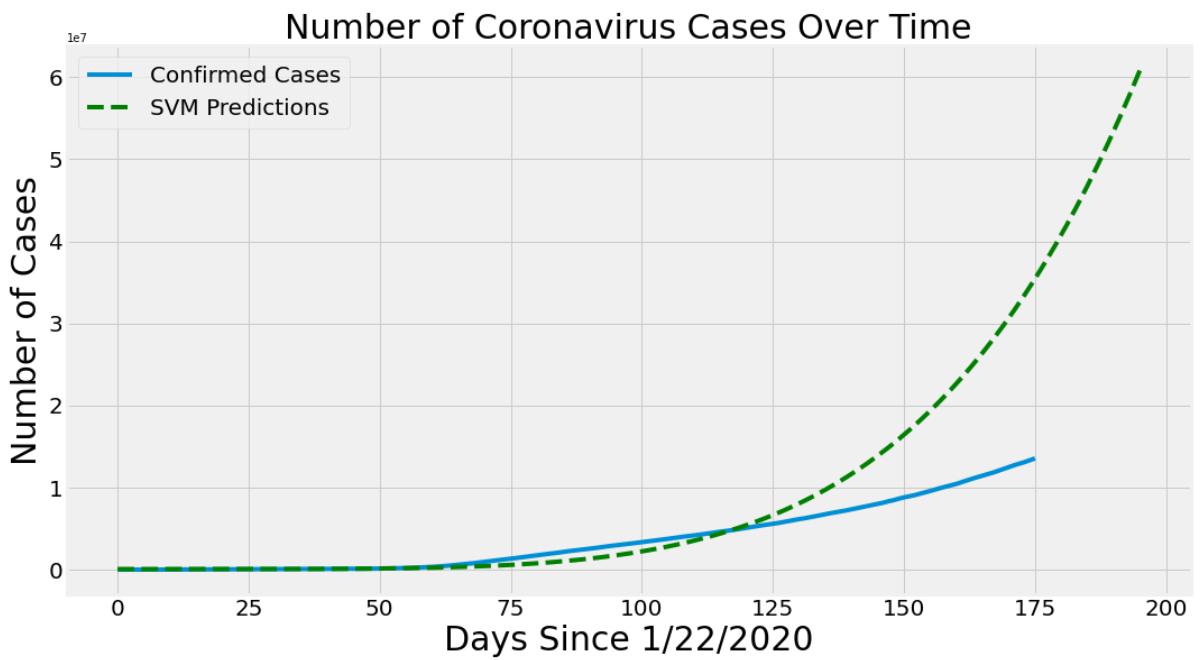


```
In [ ]: def plot_predictions(x, y, pred, algo_name, color):
    plt.figure(figsize=(16, 9))
    plt.plot(x, y)
    plt.plot(future_forecast, pred, linestyle='dashed', color=color)
    plt.title('Number of Coronavirus Cases Over Time', size=30)
    plt.xlabel('Days Since 1/22/2020', size=30)
    plt.ylabel('Number of Cases', size=30)
    plt.legend(['Confirmed Cases', algo_name], prop={'size': 20})
    plt.xticks(size=20)
    plt.yticks(size=20)
    plt.show()
```

```
In [ ]: plot_predictions(adjusted_dates, world_cases, linear_pred, 'Polynomial Regression F
```



```
In [ ]: plot_predictions(adjusted_dates, world_cases, svm_pred, 'SVM Predictions', 'green')
```



```
In [ ]: # Future predictions using polynomial regression
linear_pred = linear_pred.reshape(1,-1)[0]
poly_df = pd.DataFrame({'Date': future_forecast_dates[-20:], 'Predicted number of cases': linear_pred})
poly_df
```

Out[ ]:

	Date	Predicted number of Confirmed Cases Worldwide
<b>0</b>	07/16/2020	12413063.0
<b>1</b>	07/17/2020	12565602.0
<b>2</b>	07/18/2020	12718864.0
<b>3</b>	07/19/2020	12872846.0
<b>4</b>	07/20/2020	13027542.0
<b>5</b>	07/21/2020	13182951.0
<b>6</b>	07/22/2020	13339068.0
<b>7</b>	07/23/2020	13495889.0
<b>8</b>	07/24/2020	13653411.0
<b>9</b>	07/25/2020	13811630.0
<b>10</b>	07/26/2020	13970541.0
<b>11</b>	07/27/2020	14130142.0
<b>12</b>	07/28/2020	14290429.0
<b>13</b>	07/29/2020	14451397.0
<b>14</b>	07/30/2020	14613044.0
<b>15</b>	07/31/2020	14775365.0
<b>16</b>	08/01/2020	14938357.0
<b>17</b>	08/02/2020	15102015.0
<b>18</b>	08/03/2020	15266337.0
<b>19</b>	08/04/2020	15431318.0

In [ ]:

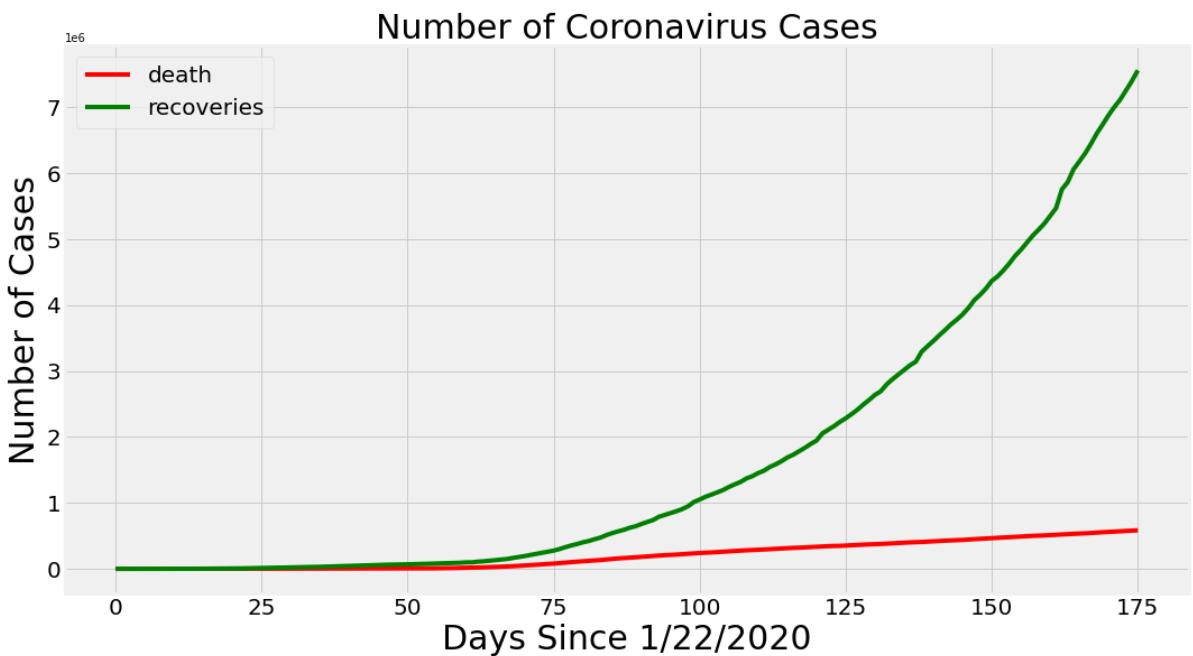
```
# Future predictions using SVM
svm_df = pd.DataFrame({'Date': future_forecast_dates[-20:], 'SVM Predicted # of Cor
svm_df
```

Out[ ]:

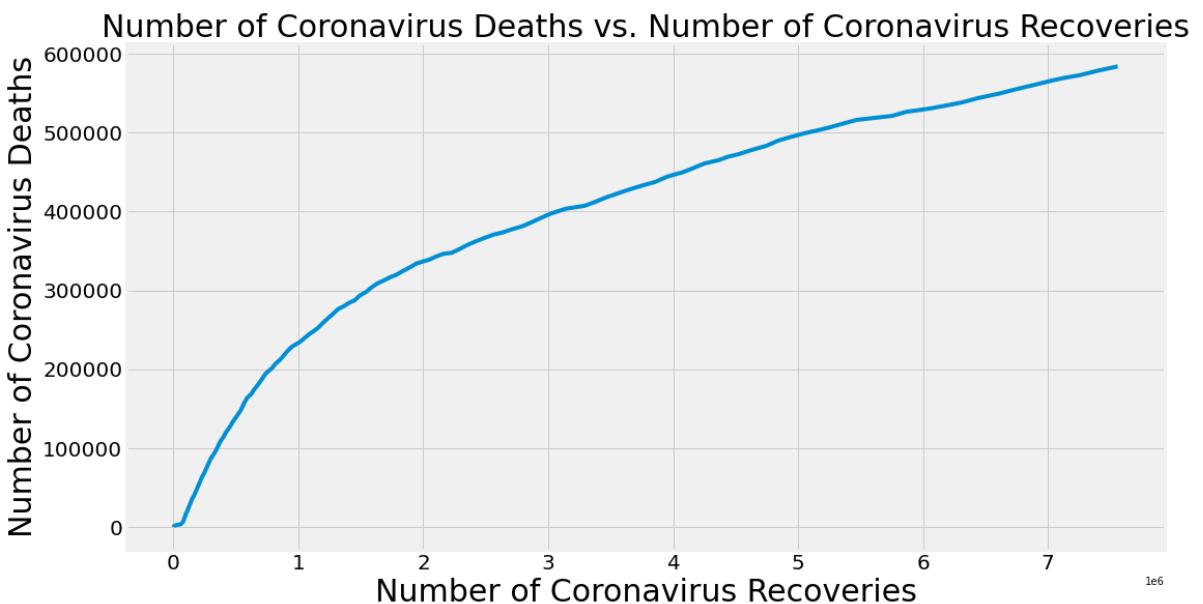
	Date	SVM Predicted # of Confirmed Cases Worldwide
0	07/16/2020	36460620.0
1	07/17/2020	37506189.0
2	07/18/2020	38575656.0
3	07/19/2020	39669428.0
4	07/20/2020	40787917.0
5	07/21/2020	41931539.0
6	07/22/2020	43100716.0
7	07/23/2020	44295873.0
8	07/24/2020	45517442.0
9	07/25/2020	46765857.0
10	07/26/2020	48041558.0
11	07/27/2020	49344991.0
12	07/28/2020	50676605.0
13	07/29/2020	52036855.0
14	07/30/2020	53426199.0
15	07/31/2020	54845103.0
16	08/01/2020	56294036.0
17	08/02/2020	57773471.0
18	08/03/2020	59283887.0
19	08/04/2020	60825770.0

In [ ]:

```
plt.figure(figsize=(16, 9))
plt.plot(adjusted_dates, total_deaths, color='r')
plt.plot(adjusted_dates, total_recovered, color='green')
plt.legend(['death', 'recoveries'], loc='best', fontsize=20)
plt.title('Number of Coronavirus Cases', size=30)
plt.xlabel('Days Since 1/22/2020', size=30)
plt.ylabel('Number of Cases', size=30)
plt.xticks(size=20)
plt.yticks(size=20)
plt.show()
```



```
In [ ]: plt.figure(figsize=(16, 9))
plt.plot(total_recovered, total_deaths)
plt.title('Number of Coronavirus Deaths vs. Number of Coronavirus Recoveries', size=30)
plt.xlabel('Number of Coronavirus Recoveries', size=30)
plt.ylabel('Number of Coronavirus Deaths', size=30)
plt.xticks(size=20)
plt.yticks(size=20)
plt.show()
```



```
In [ ]: def country_plot(x, y1, y2, y3, y4, country):
    plt.figure(figsize=(16, 9))
    plt.plot(x, y1)
    plt.title('{} Confirmed Cases'.format(country), size=30)
    plt.xlabel('Days Since 1/22/2020', size=30)
    plt.ylabel('Number of Cases', size=30)
    plt.xticks(size=20)
    plt.yticks(size=20)
    plt.show()

    plt.figure(figsize=(16, 9))
    plt.bar(x, y2)
    plt.title('{} Daily Increases in Confirmed Cases'.format(country), size=30)
    plt.xlabel('Days Since 1/22/2020', size=30)
```

```

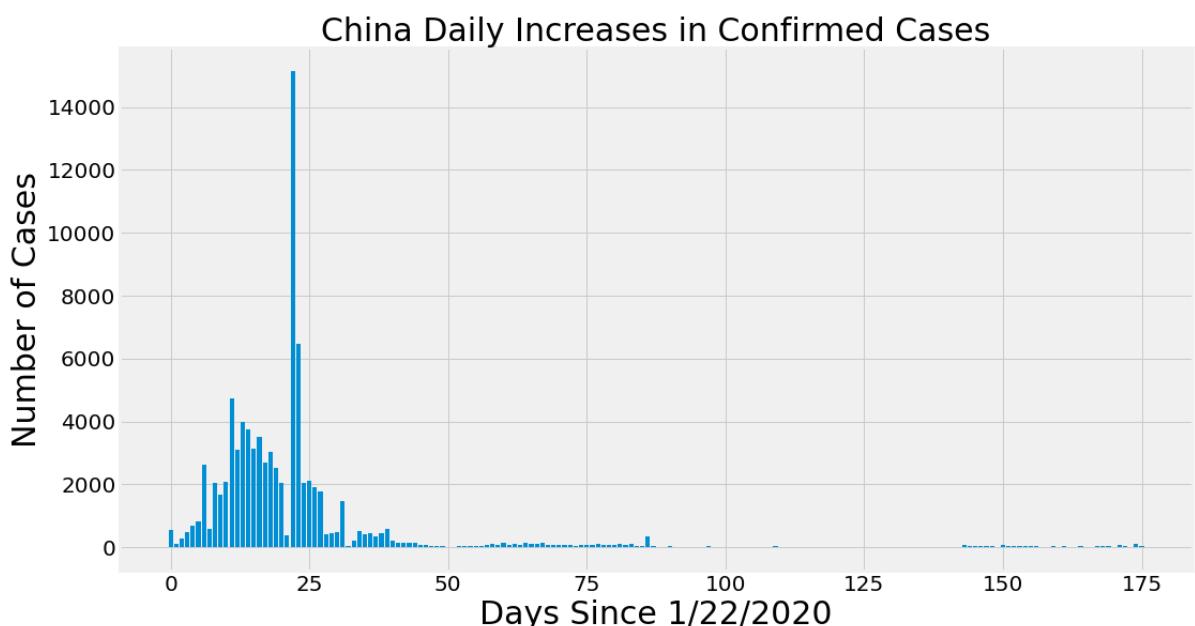
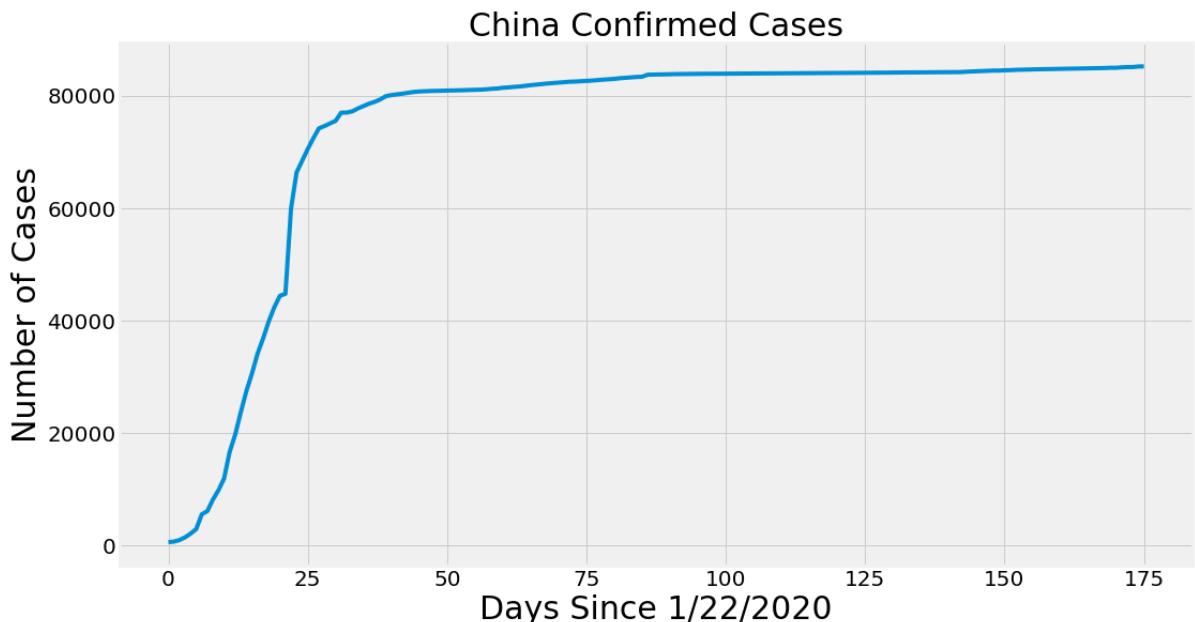
plt.ylabel('Number of Cases', size=30)
plt.xticks(size=20)
plt.yticks(size=20)
plt.show()

plt.figure(figsize=(16, 9))
plt.bar(x, y3)
plt.title('{} Daily Increases in Deaths'.format(country), size=30)
plt.xlabel('Days Since 1/22/2020', size=30)
plt.ylabel('Number of Cases', size=30)
plt.xticks(size=20)
plt.yticks(size=20)
plt.show()

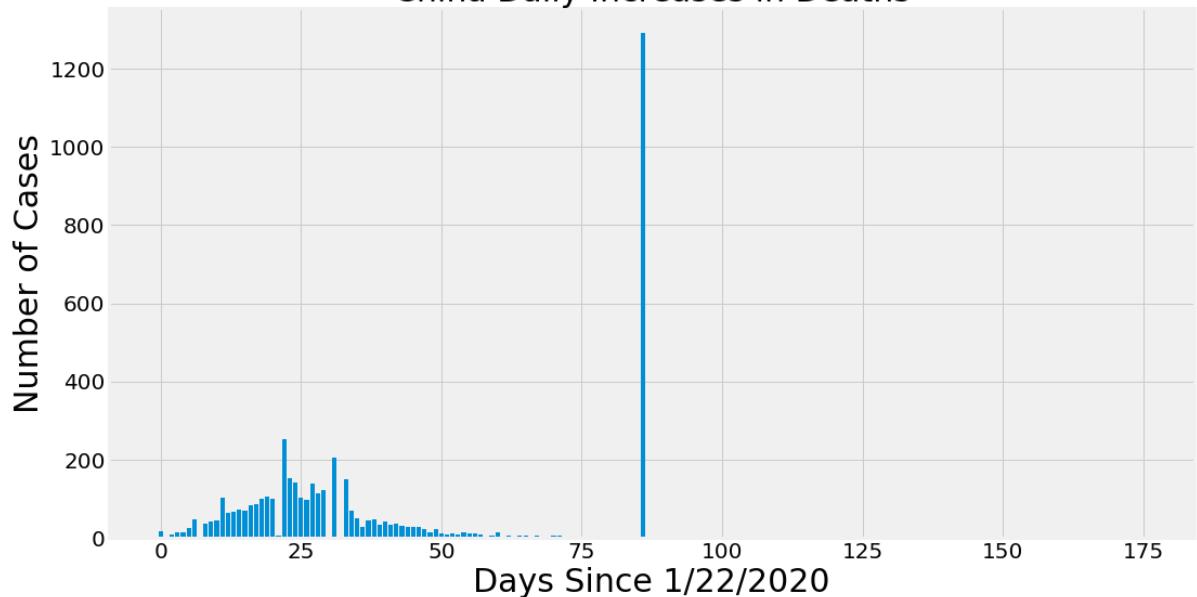
plt.figure(figsize=(16, 9))
plt.bar(x, y4)
plt.title('{} Daily Increases in Recoveries'.format(country), size=30)
plt.xlabel('Days Since 1/22/2020', size=30)
plt.ylabel('Number of Cases', size=30)
plt.xticks(size=20)
plt.yticks(size=20)
plt.show()

```

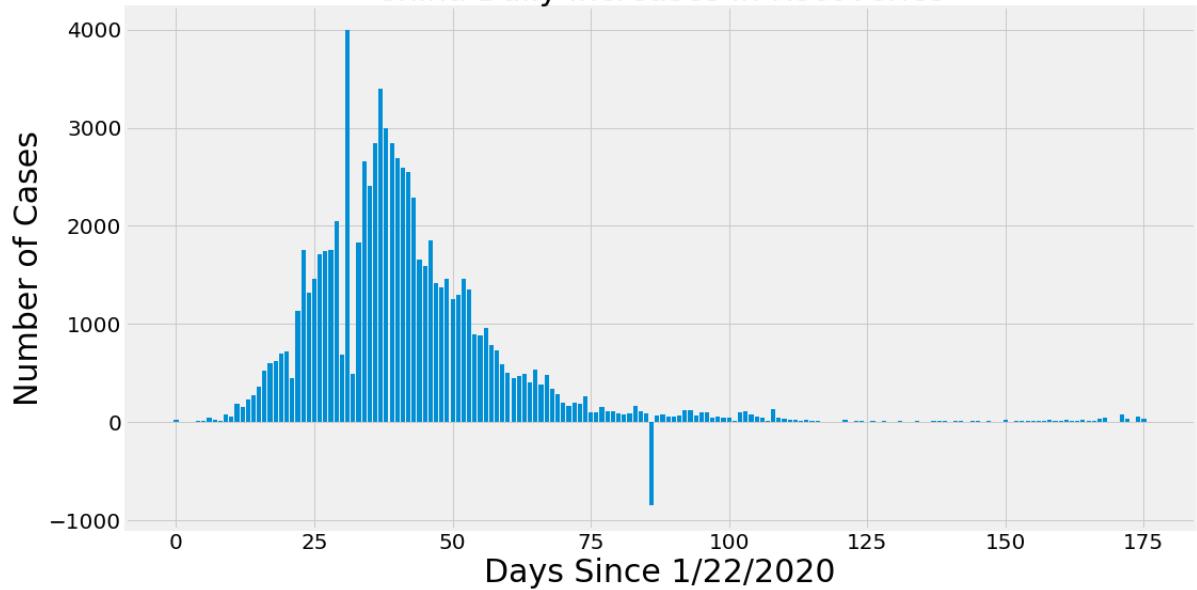
In [ ]: country\_plot(adjusted\_dates, china\_cases, china\_daily\_increase, china\_daily\_death,



### China Daily Increases in Deaths

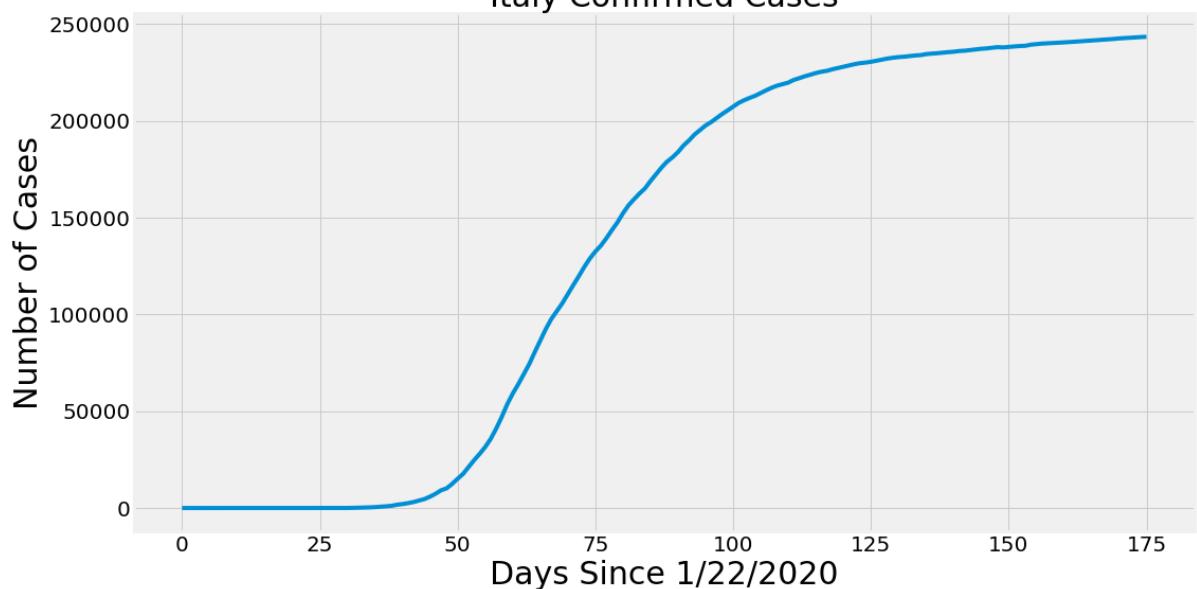


### China Daily Increases in Recoveries

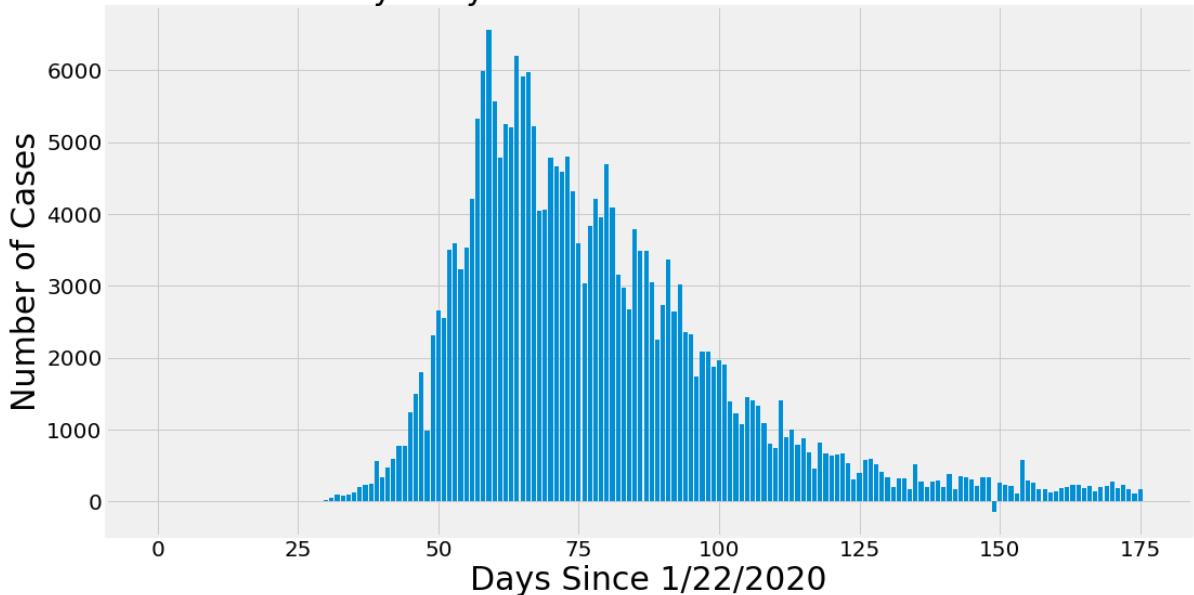


```
In [ ]: country_plot(adjusted_dates, italy_cases, italy_daily_increase, italy_daily_death,
```

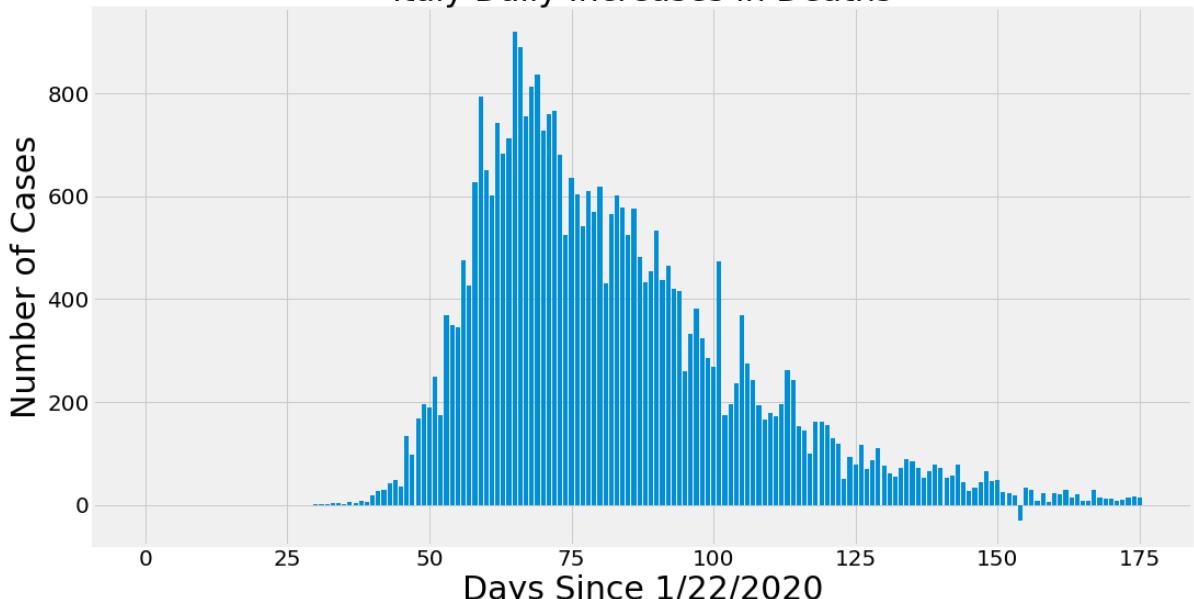
### Italy Confirmed Cases



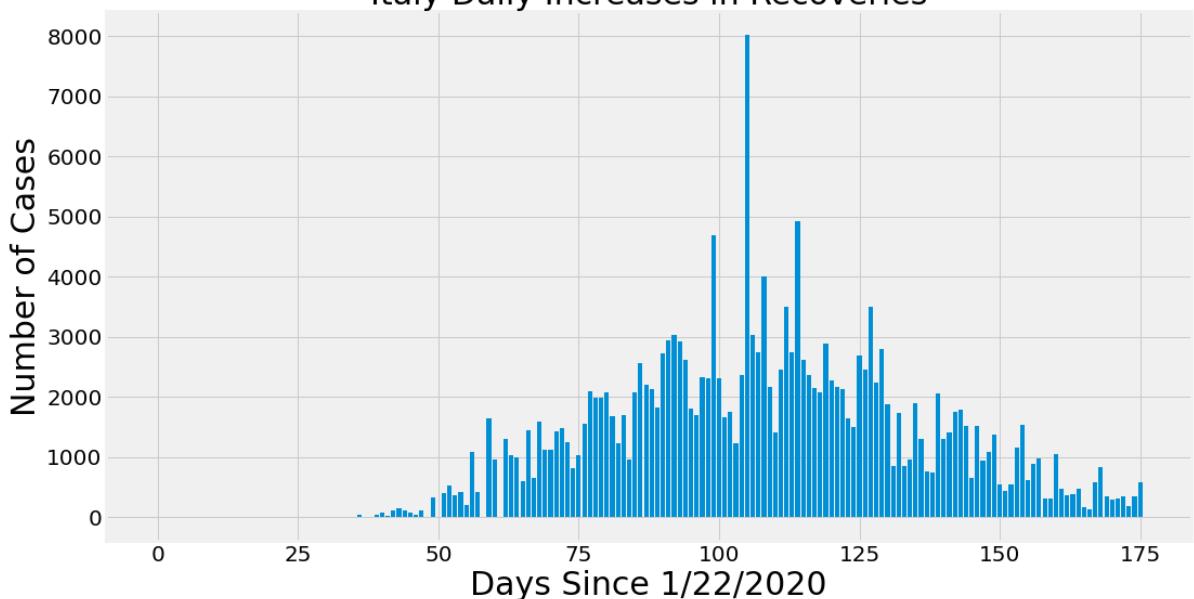
### Italy Daily Increases in Confirmed Cases



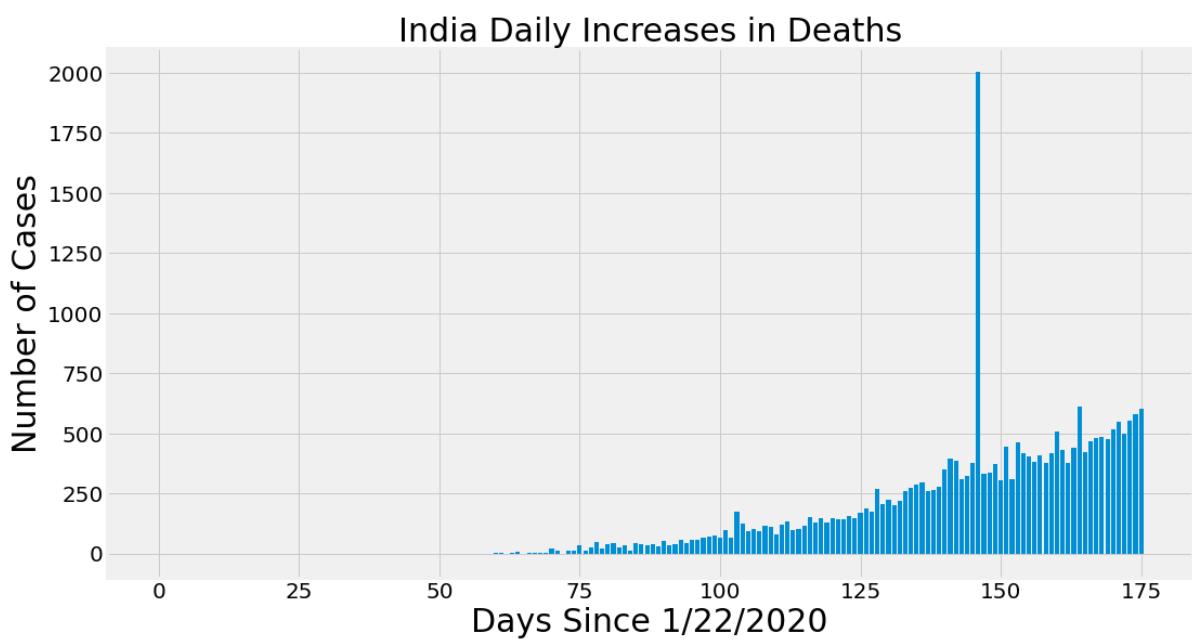
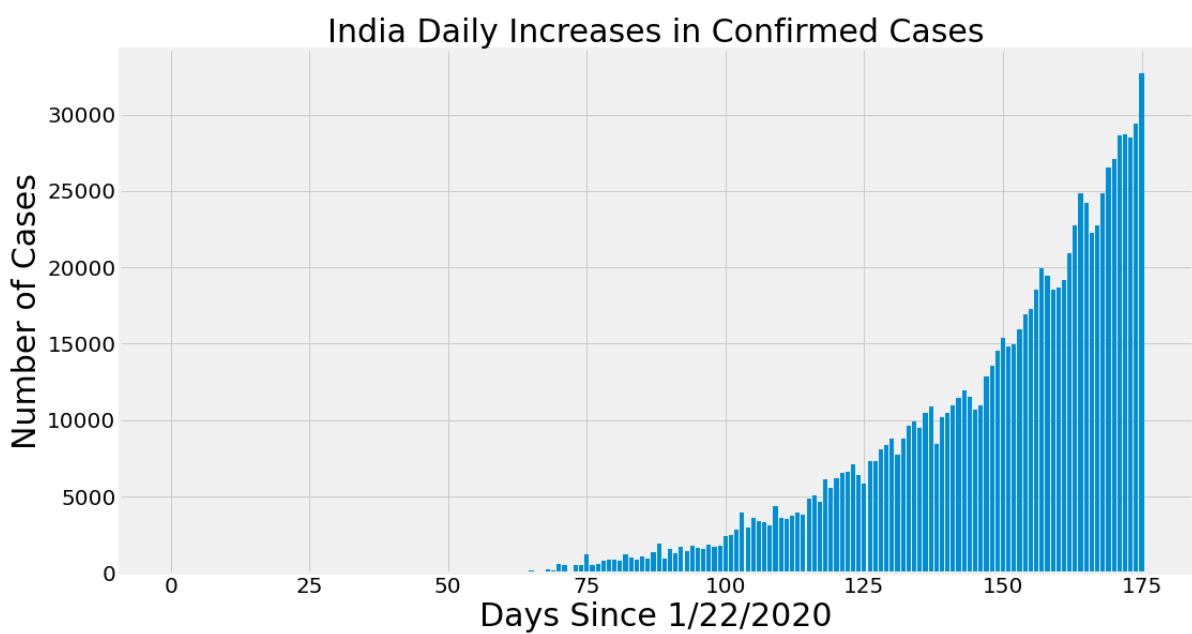
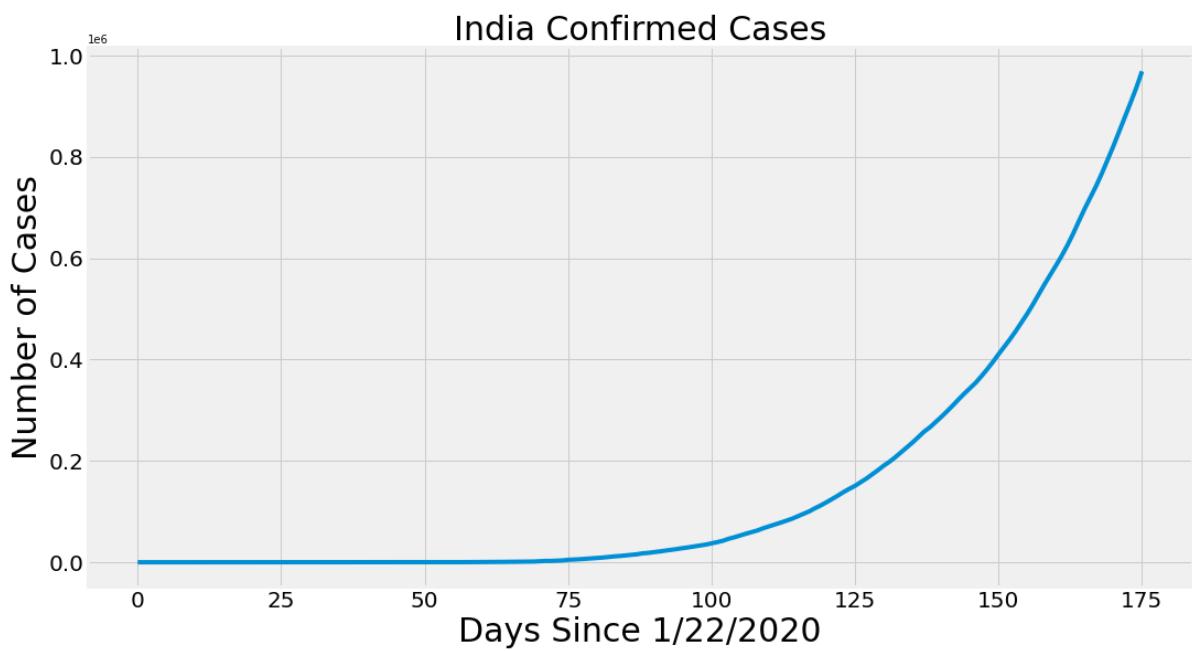
### Italy Daily Increases in Deaths



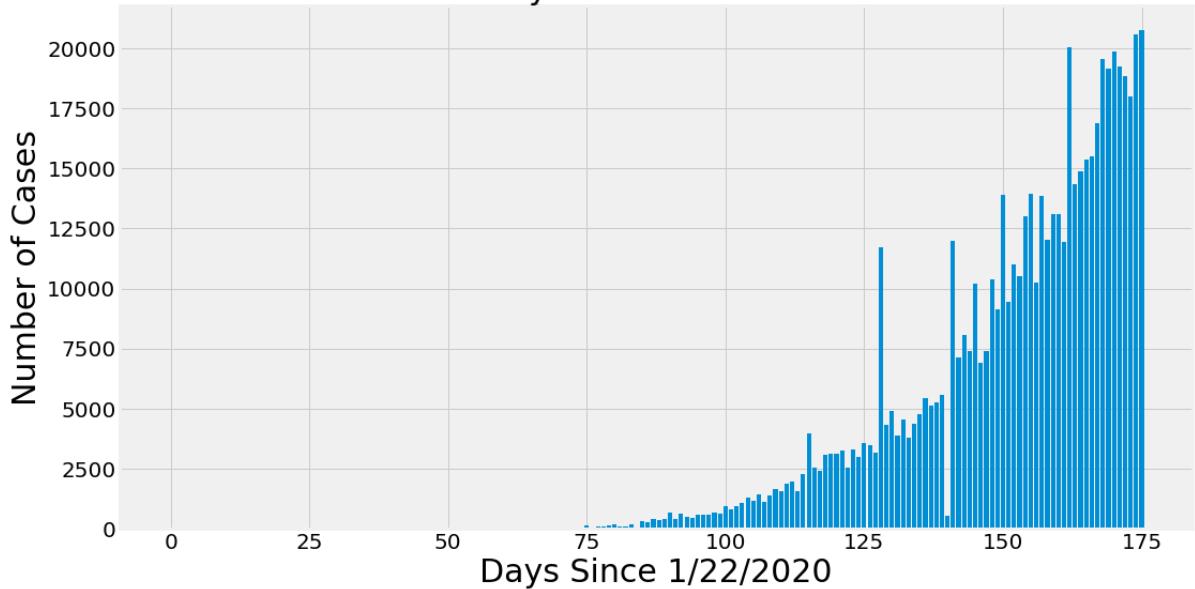
### Italy Daily Increases in Recoveries



```
In [ ]: country_plot(adjusted_dates, india_cases, india_daily_increase, india_daily_death,
```

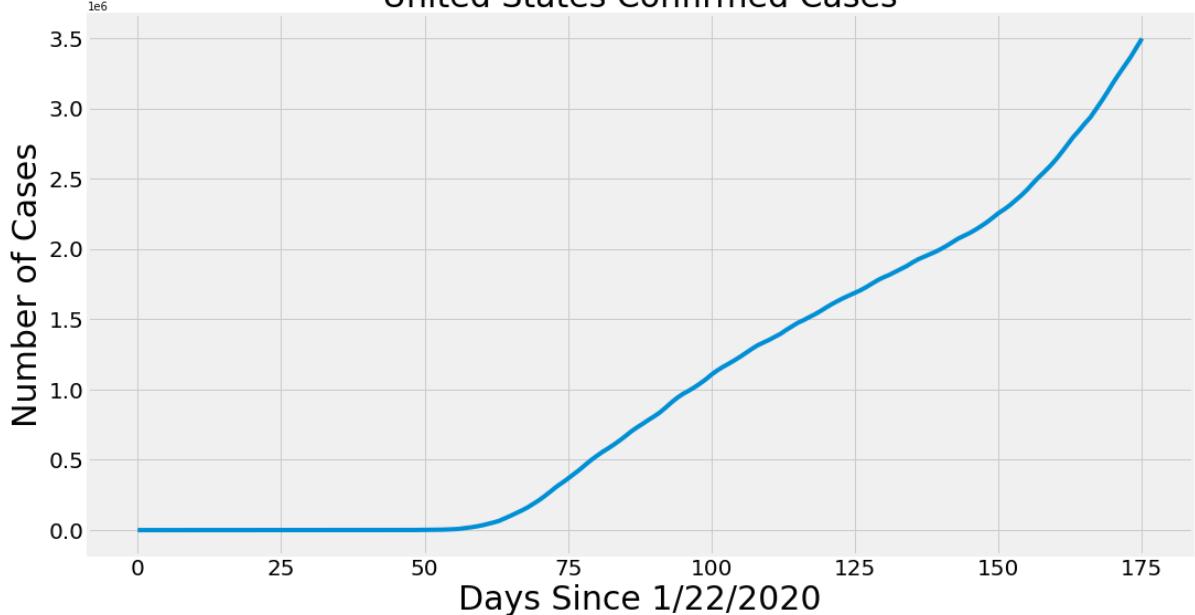


### India Daily Increases in Recoveries

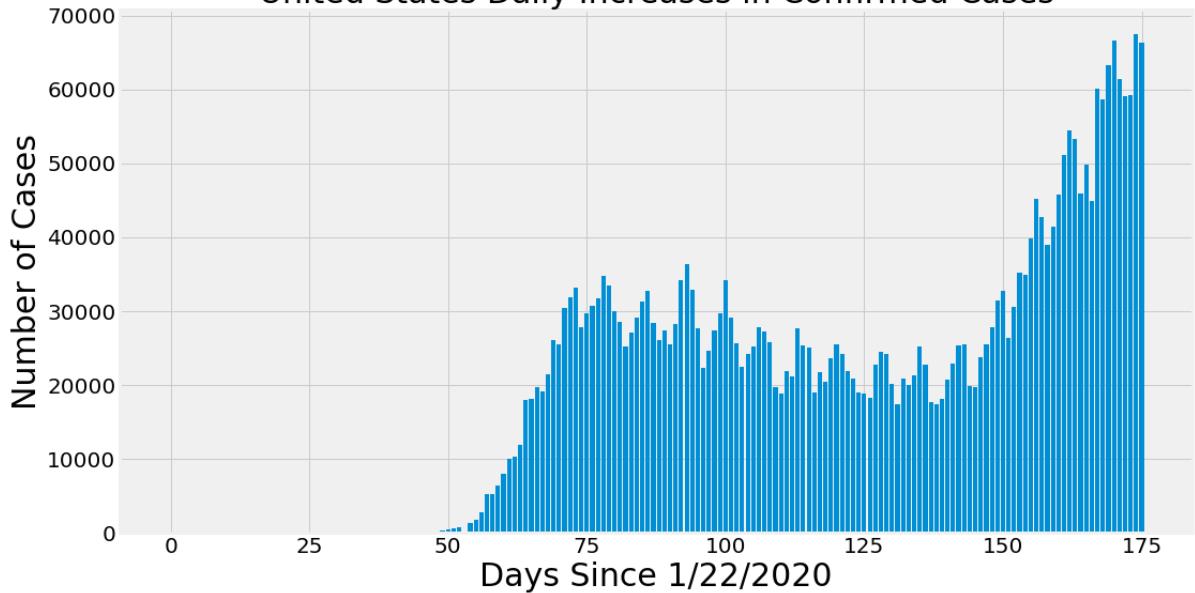


```
In [ ]: country_plot(adjusted_dates, us_cases, us_daily_increase, us_daily_death, us_daily_
```

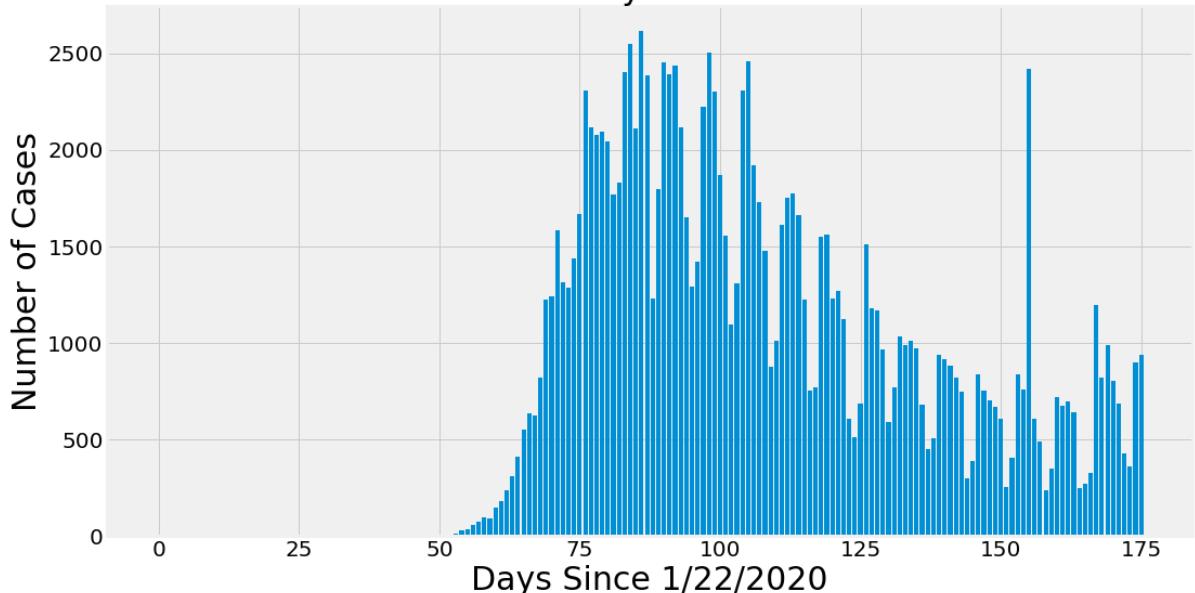
### United States Confirmed Cases



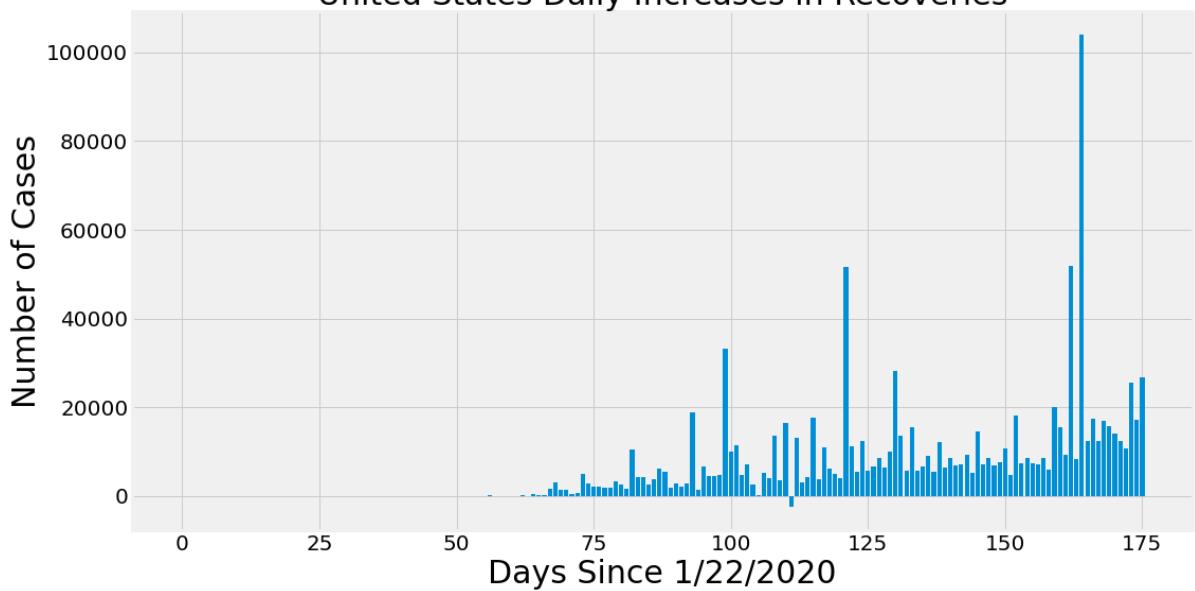
### United States Daily Increases in Confirmed Cases



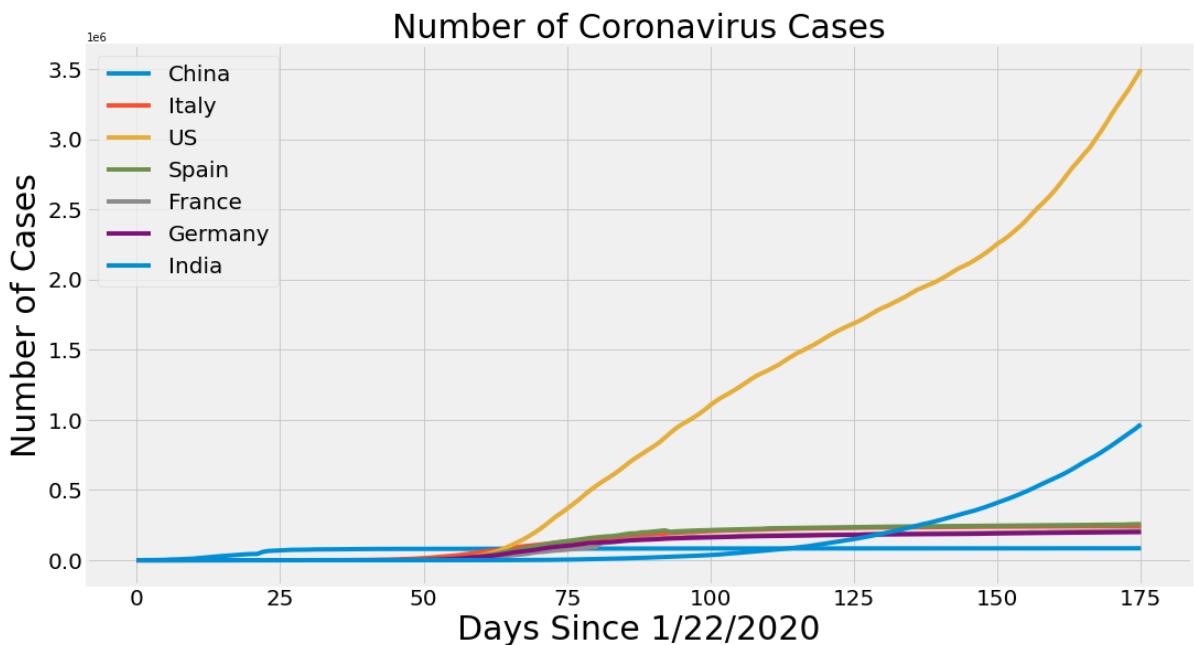
### United States Daily Increases in Deaths



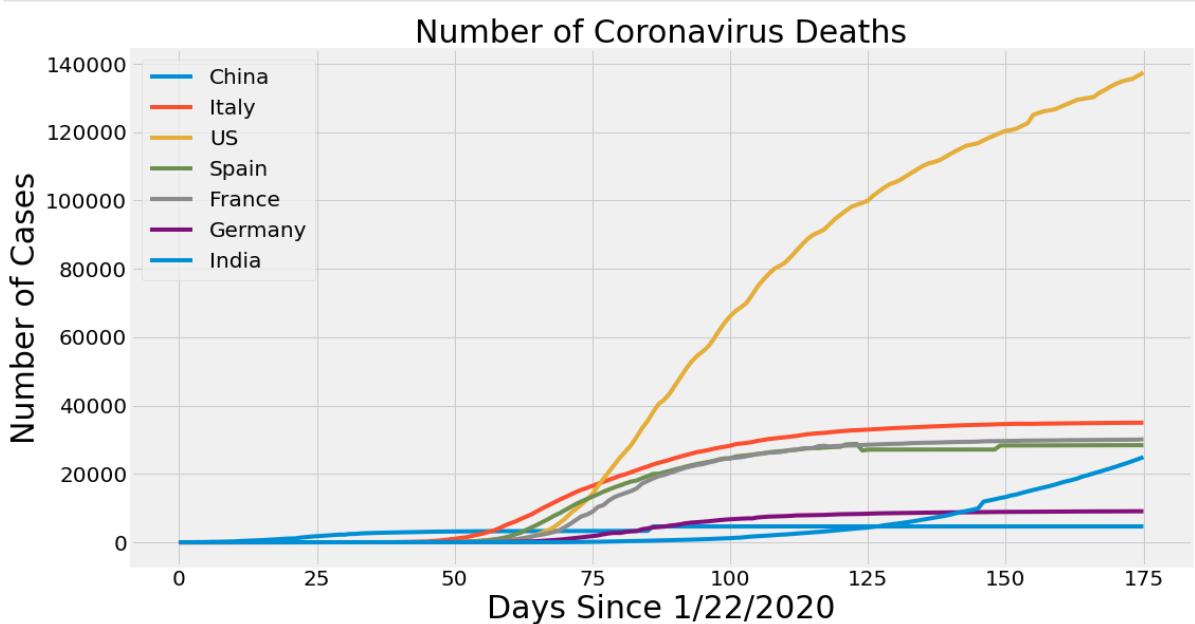
### United States Daily Increases in Recoveries



```
In [ ]: plt.figure(figsize=(16, 9))
plt.plot(adjusted_dates, china_cases)
plt.plot(adjusted_dates, italy_cases)
plt.plot(adjusted_dates, us_cases)
plt.plot(adjusted_dates, spain_cases)
plt.plot(adjusted_dates, france_cases)
plt.plot(adjusted_dates, germany_cases)
plt.plot(adjusted_dates, india_cases)
plt.title('Number of Coronavirus Cases', size=30)
plt.xlabel('Days Since 1/22/2020', size=30)
plt.ylabel('Number of Cases', size=30)
plt.legend(['China', 'Italy', 'US', 'Spain', 'France', 'Germany', 'India'], prop={})
plt.xticks(size=20)
plt.yticks(size=20)
plt.show()
```



```
In [ ]: plt.figure(figsize=(16, 9))
plt.plot(adjusted_dates, china_deaths)
plt.plot(adjusted_dates, italy_deaths)
plt.plot(adjusted_dates, us_deaths)
plt.plot(adjusted_dates, spain_deaths)
plt.plot(adjusted_dates, france_deaths)
plt.plot(adjusted_dates, germany_deaths)
plt.plot(adjusted_dates, india_deaths)
plt.title('Number of Coronavirus Deaths', size=30)
plt.xlabel('Days Since 1/22/2020', size=30)
plt.ylabel('Number of Cases', size=30)
plt.legend(['China', 'Italy', 'US', 'Spain', 'France', 'Germany', 'India'], prop={})
plt.xticks(size=20)
plt.yticks(size=20)
plt.show()
```



```
In [ ]: plt.figure(figsize=(16, 9))
plt.plot(adjusted_dates, china_recoveries)
plt.plot(adjusted_dates, italy_recoveries)
plt.plot(adjusted_dates, us_recoveries)
plt.plot(adjusted_dates, spain_recoveries)
plt.plot(adjusted_dates, france_recoveries)
plt.plot(adjusted_dates, germany_recoveries)
```

```
plt.plot(adjusted_dates, india_recoveries)
plt.title('Number of Coronavirus Recoveries', size=30)
plt.xlabel('Days Since 1/22/2020', size=30)
plt.ylabel('Number of Cases', size=30)
plt.legend(['China', 'Italy', 'US', 'Spain', 'France', 'Germany', 'India'], prop={})
plt.xticks(size=20)
plt.yticks(size=20)
plt.show()
```

