covid-19-analysis-2

August 25, 2024

[1]:

```
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
```

[2]:

```
confirmed_cases = pd.read_csv('https://raw.githubusercontent.com/CSSEGISandData/
 ↪COVID-19/master/csse_covid_19_data/csse_covid_19_time_series/
 ↪time_series_covid19_confirmed_global.csv')
```

[3]:

```
confirmed_cases.head()
```

[3]:

```
  Province/State Country/Region      Lat     Long  1/22/20  1/23/20  1/24/20  \
0            NaN    Afghanistan  33.0000  65.0000        0        0        0
1            NaN        Albania  41.1533  20.1683        0        0        0
2            NaN        Algeria  28.0339   1.6596        0        0        0
3            NaN        Andorra  42.5063   1.5218        0        0        0
4            NaN         Angola -11.2027  17.8739        0        0        0

   1/25/20  1/26/20  1/27/20  …  4/18/20  4/19/20  4/20/20  4/21/20  \
0        0        0        0  …      933      996     1026     1092
1        0        0        0  …      548      562      584      609
2        0        0        0  …     2534     2629     2718     2811
3        0        0        0  …      704      713      717      717
4        0        0        0  …       24       24       24       24
```

```
   4/22/20  4/23/20  4/24/20  4/25/20  4/26/20  4/27/20
0     1176     1279     1351     1463     1531     1703
1      634      663      678      712      726      736
2     2910     3007     3127     3256     3382     3517
3      723      723      731      738      738      743
4       25       25       25       25       26       27

[5 rows x 101 columns]
```

[4]:
```
deaths_reported = pd.read_csv('https://raw.githubusercontent.com/CSSEGISandData/
  ↪COVID-19/master/csse_covid_19_data/csse_covid_19_time_series/
  ↪time_series_covid19_deaths_global.csv')
```

[5]:
```
deaths_reported.head()
```

[5]:
```
  Province/State Country/Region      Lat     Long  1/22/20  1/23/20  1/24/20  \
0            NaN    Afghanistan  33.0000  65.0000        0        0        0
1            NaN        Albania  41.1533  20.1683        0        0        0
2            NaN        Algeria  28.0339   1.6596        0        0        0
3            NaN        Andorra  42.5063   1.5218        0        0        0
4            NaN         Angola -11.2027  17.8739        0        0        0

   1/25/20  1/26/20  1/27/20  …  4/18/20  4/19/20  4/20/20  4/21/20  \
0        0        0        0  …       30       33       36       36
1        0        0        0  …       26       26       26       26
2        0        0        0  …      367      375      384      392
3        0        0        0  …       35       36       37       37
4        0        0        0  …        2        2        2        2

   4/22/20  4/23/20  4/24/20  4/25/20  4/26/20  4/27/20
0       40       42       43       47       50       57
1       27       27       27       27       28       28
2      402      407      415      419      425      432
3       37       37       40       40       40       40
4        2        2        2        2        2        2

[5 rows x 101 columns]
```

[6]:
```
recovered_cases = pd.read_csv('https://raw.githubusercontent.com/CSSEGISandData/
  ↪COVID-19/master/csse_covid_19_data/csse_covid_19_time_series/
  ↪time_series_covid19_recovered_global.csv')
```

[7]:
```
recovered_cases.head()
```

[7]:
```
  Province/State Country/Region      Lat     Long  1/22/20  1/23/20  1/24/20  \
0            NaN    Afghanistan  33.0000  65.0000        0        0        0
1            NaN        Albania  41.1533  20.1683        0        0        0
```

```
2            NaN        Algeria  28.0339   1.6596         0         0         0
3            NaN        Andorra  42.5063   1.5218         0         0         0
4            NaN         Angola -11.2027  17.8739         0         0         0

   1/25/20  1/26/20  1/27/20  …  4/18/20  4/19/20  4/20/20  4/21/20  \
0        0        0        0  …      112      131      135      150
1        0        0        0  …      302      314      327      345
2        0        0        0  …      894     1047     1099     1152
3        0        0        0  …      205      235      248      282
4        0        0        0  …        6        6        6        6

   4/22/20  4/23/20  4/24/20  4/25/20  4/26/20  4/27/20
0      166      179      188      188      207      220
1      356      385      394      403      410      422
2     1204     1355     1408     1479     1508     1558
3      309      333      344      344      344      385
4        6        6        6        6        6        6

[5 rows x 101 columns]
```

```
[8]: latest_data = pd.read_csv('https://raw.githubusercontent.com/CSSEGISandData/
     ↪COVID-19/master/csse_covid_19_data/csse_covid_19_daily_reports/04-25-2020.
     ↪csv')
```

```
[9]: latest_data.head()
```

```
[9]:       FIPS     Admin2  Province_State Country_Region          Last_Update  \
     0  45001.0  Abbeville  South Carolina             US  2020-04-26 02:30:51
     1  22001.0     Acadia       Louisiana             US  2020-04-26 02:30:51
     2  51001.0   Accomack        Virginia             US  2020-04-26 02:30:51
     3  16001.0        Ada           Idaho             US  2020-04-26 02:30:51
     4  19001.0      Adair            Iowa             US  2020-04-26 02:30:51

              Lat       Long_  Confirmed  Deaths  Recovered  Active  \
     0  34.223334  -82.461707         24       0          0      24
     1  30.295065  -92.414197        130       7          0     123
     2  37.767072  -75.632346        146       3          0     143
     3  43.452658 -116.241552        650      15          0     635
     4  41.330756  -94.471059          1       0          0       1

                           Combined_Key
     0  Abbeville, South Carolina, US
     1          Acadia, Louisiana, US
     2          Accomack, Virginia, US
     3                 Ada, Idaho, US
     4                Adair, Iowa, US
```

[10]:

```
# Fetching all the columns from confirmed dataset
cols = confirmed_cases.keys()
cols
```

[10]:

```
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',
       …
       '4/18/20', '4/19/20', '4/20/20', '4/21/20', '4/22/20', '4/23/20',
       '4/24/20', '4/25/20', '4/26/20', '4/27/20'],
      dtype='object', length=101)
```

[11]:

```
# 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]]
```

[12]:

```
confirmed
```

[12]:

```
     1/22/20  1/23/20  1/24/20  1/25/20  1/26/20  1/27/20  1/28/20  1/29/20  \
0          0        0        0        0        0        0        0        0
1          0        0        0        0        0        0        0        0
2          0        0        0        0        0        0        0        0
3          0        0        0        0        0        0        0        0
4          0        0        0        0        0        0        0        0
..       …        …        …        …        …        …        …        …
259        0        0        0        0        0        0        0        0
260        0        0        0        0        0        0        0        0
261        0        0        0        0        0        0        0        0
262        0        0        0        0        0        0        0        0
263        0        0        0        0        0        0        0        0

     1/30/20  1/31/20  …  4/18/20  4/19/20  4/20/20  4/21/20  4/22/20  \
0          0        0  …      933      996     1026     1092     1176
1          0        0  …      548      562      584      609      634
2          0        0  …     2534     2629     2718     2811     2910
3          0        0  …      704      713      717      717      723
4          0        0  …       24       24       24       24       25
..       …        … …      …        …        …        …        …
259        0        0  …        1        1        1        1        1
260        0        0  …        4        4        4        4        4
261        0        0  …        6        6        6        6        6
262        0        0  …        4        4        4        4        4
263        0        0  …        1        1        1        1        1

     4/23/20  4/24/20  4/25/20  4/26/20  4/27/20
0       1279     1351     1463     1531     1703
1        663      678      712      726      736
```

```
2       3007     3127     3256     3382     3517
3        723      731      738      738      743
4         25       25       25       26       27
..       …      …      …      …      …
259        1        1        1        1        1
260        5        5        5        6        6
261        6        6        6        6        6
262        4        4        4        4        4
263        1        1        1        1        1

[264 rows x 97 columns]
```

[13]:
```
dates = confirmed.keys()
world_cases = []
total_deaths = []
mortality_rate = []
recovery_rate = []
total_recovered = []
total_active = []

china_cases = []
italy_cases = []
us_cases = []
spain_cases = []
france_cases = []
germany_cases = []
uk_cases = []
russia_cases = []
india_cases = []

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


china_recoveries = []
italy_recoveries = []
us_recoveries = []
spain_recoveries = []
france_recoveries = []
germany_recoveries = []
```

```
uk_recoveries = []
russia_recoveries = []
india_recoveries = []
```

[14]:
```
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'][i].sum())
    italy_cases.append(confirmed_cases[confirmed_cases['Country/
 ↪Region']=='Italy'][i].sum())
    us_cases.append(confirmed_cases[confirmed_cases['Country/Region']=='US'][i].
 ↪sum())
    spain_cases.append(confirmed_cases[confirmed_cases['Country/
 ↪Region']=='Spain'][i].sum())
    france_cases.append(confirmed_cases[confirmed_cases['Country/
 ↪Region']=='France'][i].sum())
    germany_cases.append(confirmed_cases[confirmed_cases['Country/
 ↪Region']=='Germany'][i].sum())
    uk_cases.append(confirmed_cases[confirmed_cases['Country/Region']=='United␣
 ↪Kingdom'][i].sum())
    russia_cases.append(confirmed_cases[confirmed_cases['Country/
 ↪Region']=='Russia'][i].sum())
    india_cases.append(confirmed_cases[confirmed_cases['Country/
 ↪Region']=='India'][i].sum())


    china_deaths.append(deaths_reported[deaths_reported['Country/
 ↪Region']=='China'][i].sum())
    italy_deaths.append(deaths_reported[deaths_reported['Country/
 ↪Region']=='Italy'][i].sum())
    us_deaths.append(deaths_reported[deaths_reported['Country/
 ↪Region']=='US'][i].sum())
    spain_deaths.append(deaths_reported[deaths_reported['Country/
 ↪Region']=='Spain'][i].sum())
```

```
    france_deaths.append(deaths_reported[deaths_reported['Country/
↪Region']=='France'][i].sum())
    germany_deaths.append(deaths_reported[deaths_reported['Country/
↪Region']=='Germany'][i].sum())
    uk_deaths.append(deaths_reported[deaths_reported['Country/Region']=='United␣
↪Kingdom'][i].sum())
    russia_deaths.append(deaths_reported[deaths_reported['Country/
↪Region']=='Russia'][i].sum())
    india_deaths.append(deaths_reported[deaths_reported['Country/
↪Region']=='India'][i].sum())


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

```
[15]: world_cases
```

```
[15]: [555,
       654,
       941,
       1434,
       2118,
       2927,
       5578,
       6166,
       8234,
       9927,
       12038,
       16787,
       19881,
       23892,
       27635,
       30794,
       34391,
       37120,
       40150,
       42762,
       44802,
       45221,
       60368,
       66885,
       69030,
       71224,
       73258,
       75136,
       75639,
       76197,
       76819,
       78572,
       78958,
       79561,
       80406,
       81388,
       82746,
       84112,
       86011,
       88369,
       90306,
       92840,
       95120,
       97886,
       101801,
       105847,
       109821,
       113590,
       118620,
       125875,
       128352,
       145219,
       156116,
       167466,
       181603,
       197113,
       214846,
       242616,
       272247,
       304555,
        337018,
        378282,
        418079,
        467723,
        529701,
        593423,
        660824,
        720285,
        782490,
        857608,
        932638,
        1013458,
        1095876,
        1176059,
        1249737,
        1321427,
        1396438,
        1480200,
        1565538,
        1657929,
        1736025,
        1835164,
        1905192,
        1975581,
        2055506,
        2151872,
        2239723,
        2317339,
        2400843,
        2471759,
        2548821,
        2624107,
        2707742,
        2811603,
        2897624,
        2972363,
        3041764]
```

```
[16]: total_deaths
```

```
[16]: [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,
2007,
2122,
2247,
2251,
2458,
2469,
2629,
2708,
2770,
2814,
2872,
2941,
2996,
3085,
3160,
3254,
3348,
3460,
3558,
3802,
3988,
4262,
4615,
4720,
5404,
5819,
6440,
7126,
7905,
8733,
9867,
11300,
12973,
14652,
16506,
18626,
21182,
23972,
27202,
30656,
33929,
37590,
42109,
47191,
52999,
58800,
64626,
69389,
74596,
81953,
88380,
95541,
102573,
108551,
114123,
119529,
126071,
134234,
143853,
153897,
159615,
165081,
170013,
176729,
183180,
190858,
197174,
202868,
206568,
211167]
```

[17]:

```
confirmed_sum
```

[17]:

```
3041764
```

```
[18]: death_sum
```

```
[18]: 211167
```

```
[19]: recovered_sum
```

```
[19]: 893967
```

```
[20]: us_cases
```

```
[20]: [1,
       1,
       2,
       2,
       5,
       5,
       5,
       5,
       5,
       7,
       8,
       8,
       11,
       11,
       11,
       11,
       11,
       11,
       11,
       11,
       12,
       12,
       13,
       13,
       13,
       13,
       13,
       13,
       13,
       13,
       15,
       15,
       15,
       51,
       51,
       57,
       58,
       60,
       68,
       74,
       98,
       118,
       149,
       217,
       262,
       402,
       518,
       583,
       959,
       1281,
       1663,
       2179,
       2727,
       3499,
       4632,
       6421,
       7783,
       13747,
       19273,
       25600,
       33276,
       43843,
       53736,
       65778,
       83836,
       101657,
       121465,
       140909,
       161831,
       188172,
       213242,
       243622,
       275367,
       308650,
       336802,
       366317,
       397121,
       428654,
       462780,
       496535,
       526396,
       555313,
       580619,
       607670,
        636350,
        667592,
        699706,
        732197,
        758809,
        784326,
        811865,
        840351,
        869170,
        905358,
        938154,
        965785,
        988197]
```

[21]:
```
india_cases
```

[21]:
```
[0,
 0,
 0,
 0,
 0,
 0,
 0,
 0,
 1,
 1,
 1,
 2,
 3,
 3,
 3,
 3,
 3,
 3,
 3,
 3,
 3,
 3,
 3,
 3,
 3,
 3,
 3,
 3,
 3,
 3,
 3,
3,
3,
3,
3,
3,
3,
3,
3,
3,
5,
5,
28,
30,
31,
34,
39,
43,
56,
62,
73,
82,
102,
113,
119,
142,
156,
194,
244,
330,
396,
499,
536,
657,
727,
887,
987,
1024,
1251,
1397,
1998,
2543,
2567,
3082,
3588,
4778,
5311,
5916,
 6725,
 7598,
 8446,
 9205,
 10453,
 11487,
 12322,
 13430,
 14352,
 15722,
 17615,
 18539,
 20080,
 21370,
 23077,
 24530,
 26283,
 27890,
 29451]
```

[22]:
```
italy_recoveries
```

[22]:
```
[0,
 0,
 0,
 0,
 0,
 0,
 0,
 0,
 0,
 0,
 0,
 0,
 0,
 0,
 0,
 0,
 0,
 0,
 0,
 0,
 0,
 0,
 0,
 0,
 0,
0,
0,
0,
0,
0,
0,
1,
2,
1,
1,
3,
45,
46,
46,
83,
149,
160,
276,
414,
523,
589,
622,
724,
724,
1045,
1045,
1439,
1966,
2335,
2749,
2941,
4025,
4440,
4440,
6072,
7024,
7024,
8326,
9362,
10361,
10950,
12384,
13030,
14620,
15729,
16847,
18278,
19758,
20996,
21815,
22837,
24392,
26491,
28470,
30455,
32534,
34211,
35435,
37130,
38092,
40164,
42727,
44927,
47055,
48877,
51600,
54543,
57576,
60498,
63120,
64928,
66624]
```

[23]:

```
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
```

[24]:

```
# 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)
```

[25]:

```
spain_daily_increase
```

```
[25]: [0,
       0,
       0,
       0,
       0,
       0,
       0,
       0,
       0,
       0,
       1,
       0,
       0,
       0,
       0,
       0,
       0,
       0,
       1,
       0,
       0,
       0,
       0,
       0,
       0,
       0,
       0,
       0,
       0,
       0,
       0,
       0,
       0,
       0,
       4,
       7,
       2,
       17,
       13,
       39,
       36,
       45,
       57,
       37,
       141,
       100,
       173,
       400,
       622,
       582,
       0,
       2955,
       1159,
       1407,
       2144,
       1806,
       2162,
       4053,
       2447,
       4964,
       3394,
       6368,
       4749,
       9630,
       8271,
       7933,
       7516,
       6875,
       7846,
       7967,
       8195,
       7947,
       7134,
       6969,
       5478,
       5029,
       5267,
       6278,
       5002,
       5051,
       4754,
       3804,
       3268,
       2442,
       5103,
       7304,
       5891,
       887,
       6948,
       1536,
       3968,
       4211,
       4635,
       6740,
        3995,
        2870,
        2793]
```

[26]:
```
germany_daily_increase
```

[26]:
```
[0,
 0,
 0,
 0,
 0,
 1,
 3,
 0,
 0,
 1,
 3,
 2,
 2,
 0,
 0,
 0,
 1,
 0,
 1,
 0,
 2,
 0,
 0,
 0,
 0,
 0,
 0,
 0,
 0,
 0,
 0,
 0,
 0,
 0,
 1,
 10,
 19,
 2,
 31,
 51,
 29,
37,
66,
220,
188,
129,
241,
136,
281,
451,
170,
1597,
910,
1210,
1477,
1985,
3070,
2993,
4528,
2365,
2660,
4183,
3930,
4337,
6615,
6933,
6824,
4400,
4790,
4923,
6064,
6922,
6365,
4933,
4031,
3251,
4289,
5633,
4885,
3990,
2737,
2946,
2218,
1287,
3394,
2945,
3699,
1945,
 1842,
 1881,
 1226,
 2357,
 2481,
 1870,
 1514,
 1257,
 988]
```

[27]:

```
# 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)
```

[28]:

```
china_daily_death
```

[28]:

```
[17,
 1,
 8,
 16,
 14,
 26,
 49,
 2,
 38,
 42,
 46,
 102,
 64,
 66,
 72,
 70,
 85,
 87,
 100,
 107,
 100,
 5,
 252,
 152,
142,
103,
98,
139,
113,
122,
0,
205,
2,
150,
70,
52,
29,
44,
47,
35,
42,
33,
36,
32,
29,
28,
28,
23,
16,
22,
11,
8,
13,
10,
14,
13,
11,
8,
4,
6,
15,
0,
7,
4,
6,
5,
3,
5,
4,
1,
7,
 6,
 4,
 4,
 3,
 2,
 0,
 2,
 2,
 1,
 3,
 0,
 2,
 0,
 1,
 0,
 1290,
 0,
 0,
 0,
 0,
 0,
 0,
 0,
 0,
 1,
 0]
```

```
[29]: uk_daily_death
```

```
[29]: [0,
 0,
 0,
 0,
 0,
 0,
 0,
 0,
 0,
 0,
 0,
 0,
 0,
 0,
 0,
 0,
 0,
 0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
0,
1,
1,
0,
1,
1,
2,
2,
0,
0,
13,
0,
35,
0,
16,
66,
40,
56,
48,
54,
87,
43,
114,
 181,
 260,
 210,
 180,
 382,
 564,
 569,
 685,
 709,
 623,
 442,
 786,
 940,
 882,
 981,
 918,
 737,
 718,
 782,
 765,
 865,
 848,
 891,
 597,
 455,
 828,
 773,
 640,
 776,
 814,
 413,
 363]
```

[30]:

```
# 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)
```

[31]:

```
india_daily_recovery
```

```
[31]: [0,
       0,
       0,
       0,
       0,
       0,
       0,
       0,
       0,
       0,
       0,
       0,
       0,
       0,
       0,
       0,
       0,
       0,
       0,
       0,
       0,
       0,
       0,
       0,
       0,
       3,
       0,
       0,
       0,
       0,
       0,
       0,
       0,
       0,
       0,
       0,
       0,
       0,
       0,
       0,
       0,
       0,
       0,
       0,
       0,
       0,
       0,
0,
1,
0,
0,
0,
0,
9,
0,
1,
0,
1,
5,
3,
4,
0,
13,
3,
2,
28,
11,
11,
7,
21,
25,
43,
1,
37,
0,
146,
46,
85,
114,
154,
195,
111,
101,
178,
73,
336,
273,
422,
391,
419,
702,
395,
642,
486,
 441,
 584,
 614]
```

[32]:
```
world_daily_recovery
```

[32]:
```
[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,
2493,
2336,
1800,
1910,
2599,
1321,
1927,
2373,
3410,
2054,
2752,
2472,
1663,
2445,
4272,
6207,
452,
9649,
5787,
8363,
8765,
8500,
9667,
15484,
13468,
15143,
17086,
15533,
20356,
13860,
16503,
23539,
28607,
25314,
22121,
26014,
19612,
26933,
25606,
36758,
31088,
26236,
23976,
 31584,
 22002,
 34485,
 30112,
 28907,
 50672,
 27757,
 47895,
 28234]
```

[33]:

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

[33]:

```
['US',
 'Canada',
 'United Kingdom',
 'China',
 'Netherlands',
 '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',
 'Brazil',
 'Brunei',
 'Bulgaria',
 'Burkina Faso',
 'Burma',
'Burundi',
'Cabo Verde',
'Cambodia',
'Cameroon',
'Central African Republic',
'Chad',
'Chile',
'Colombia',
'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',
'Germany',
'Ghana',
'Greece',
'Grenada',
'Guatemala',
'Guinea',
'Guinea-Bissau',
'Guyana',
'Haiti',
'Holy See',
'Honduras',
'Hungary',
'Iceland',
'India',
'Indonesia',
'Iran',
'Iraq',
'Ireland',
'Israel',
'Italy',
'Jamaica',
'Japan',
'Jordan',
'Kazakhstan',
'Kenya',
'Korea, South',
'Kosovo',
'Kuwait',
'Kyrgyzstan',
'Laos',
'Latvia',
'Lebanon',
'Liberia',
'Libya',
'Liechtenstein',
'Lithuania',
'Luxembourg',
'MS Zaandam',
'Madagascar',
'Malawi',
'Malaysia',
'Maldives',
'Mali',
'Malta',
'Mauritania',
'Mauritius',
'Mexico',
'Moldova',
'Monaco',
'Mongolia',
'Montenegro',
'Morocco',
'Mozambique',
'Namibia',
'Nepal',
'New Zealand',
'Nicaragua',
'Niger',
'Nigeria',
'North Macedonia',
'Norway',
'Oman',
'Pakistan',
'Panama',
'Papua New Guinea',
'Paraguay',
'Peru',
'Philippines',
'Poland',
'Portugal',
'Qatar',
'Romania',
'Russia',
'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',
'Spain',
'Sri Lanka',
'Sudan',
'Suriname',
'Sweden',
'Switzerland',
'Syria',
'Taiwan*',
'Tanzania',
'Thailand',
'Timor-Leste',
'Togo',
'Trinidad and Tobago',
'Tunisia',
'Turkey',
'Uganda',
'Ukraine',
'United Arab Emirates',
'Uruguay',
'Uzbekistan',
'Venezuela',
'Vietnam',
'West Bank and Gaza',
'Western Sahara',
'Yemen',
'Zambia',
'Zimbabwe',
'Australia']
```

[34]:

```
country_confirmed_cases = []
country_death_cases = []
country_active_cases = []
country_recovery_cases = []
country_mortality_rate = []

no_cases = []
for i in unique_countries:
    cases = latest_data[latest_data['Country_Region']==i]['Confirmed'].sum()
    if cases > 0:
        country_confirmed_cases.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,␣
 ↪country_confirmed_cases), key=operator.itemgetter(1), reverse=True)]
for i in range(len(unique_countries)):
    country_confirmed_cases[i] =␣
 ↪latest_data[latest_data['Country_Region']==unique_countries[i]]['Confirmed'].
 ↪sum()
    country_death_cases.
 ↪append(latest_data[latest_data['Country_Region']==unique_countries[i]]['Deaths'].
 ↪sum())
    country_recovery_cases.
 ↪append(latest_data[latest_data['Country_Region']==unique_countries[i]]['Recovered'].
 ↪sum())
    country_active_cases.append(country_confirmed_cases[i] -␣
 ↪country_death_cases[i] - country_recovery_cases[i])
    country_mortality_rate.append(country_death_cases[i]/
 ↪country_confirmed_cases[i])
```

[35]:

```
country_df = pd.DataFrame({'Country Name': unique_countries, 'Number of␣
 ↪Confirmed Cases': country_confirmed_cases,
                           'Number of Deaths': country_death_cases, 'Number of␣
 ↪Recoveries' : country_recovery_cases,
                           'Number of Active Cases' : country_active_cases,
                           'Mortality Rate': country_mortality_rate})
# number of cases per country/region

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

[35]:

```
<pandas.io.formats.style.Styler at 0x204b72fe188>
```

[36]:

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

[37]:

```
province_confirmed_cases = []
province_country = []
province_death_cases = []
province_recovery_cases = []
province_mortality_rate = []

no_cases = []
for i in unique_provinces:
    cases = latest_data[latest_data['Province_State']==i]['Confirmed'].sum()
    if cases > 0:
        province_confirmed_cases.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,␣
 ↪province_confirmed_cases), key=operator.itemgetter(1), reverse=True)]
for i in range(len(unique_provinces)):
    province_confirmed_cases[i] =␣
 ↪latest_data[latest_data['Province_State']==unique_provinces[i]]['Confirmed'].
 ↪sum()
    province_country.
 ↪append(latest_data[latest_data['Province_State']==unique_provinces[i]]['Country_Region'].
 ↪unique()[0])
    province_death_cases.
 ↪append(latest_data[latest_data['Province_State']==unique_provinces[i]]['Deaths'].
 ↪sum())
    province_recovery_cases.
 ↪append(latest_data[latest_data['Province_State']==unique_provinces[i]]['Recovered'].
 ↪sum())
```

```
    province_mortality_rate.append(province_death_cases[i]/
 ↪province_confirmed_cases[i])
```

[38]:

```
# number of cases per province/state/city
province_df = pd.DataFrame({'Province/State Name': unique_provinces, 'Country':␣
 ↪province_country, 'Number of Confirmed Cases': province_confirmed_cases,
                           'Number of Deaths': province_death_cases, 'Number of␣
 ↪Recoveries' : province_recovery_cases,
                           'Mortality Rate': province_mortality_rate})
# number of cases per country/region

province_df.style.background_gradient(cmap='Reds')
```

[38]:

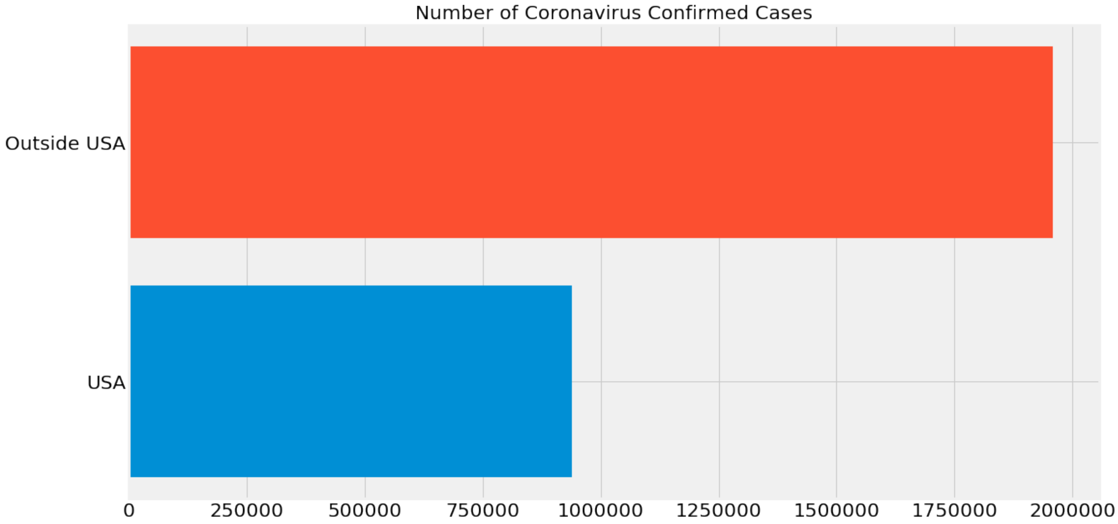
```
<pandas.io.formats.style.Styler at 0x204b7448b08>
```

[39]:

```
# 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)
province_confirmed_cases = list(province_confirmed_cases)

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

[40]:

```
USA_confirmed = latest_data[latest_data['Country_Region']=='US']['Confirmed'].
 ↪sum()
outside_USA_confirmed = np.sum(country_confirmed_cases) - 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()
```



[41]:
```
print('Outside USA {} cases:'.format(outside_USA_confirmed))
print('USA: {} cases'.format(USA_confirmed))
print('Total: {} cases'.format(USA_confirmed+outside_USA_confirmed))
```

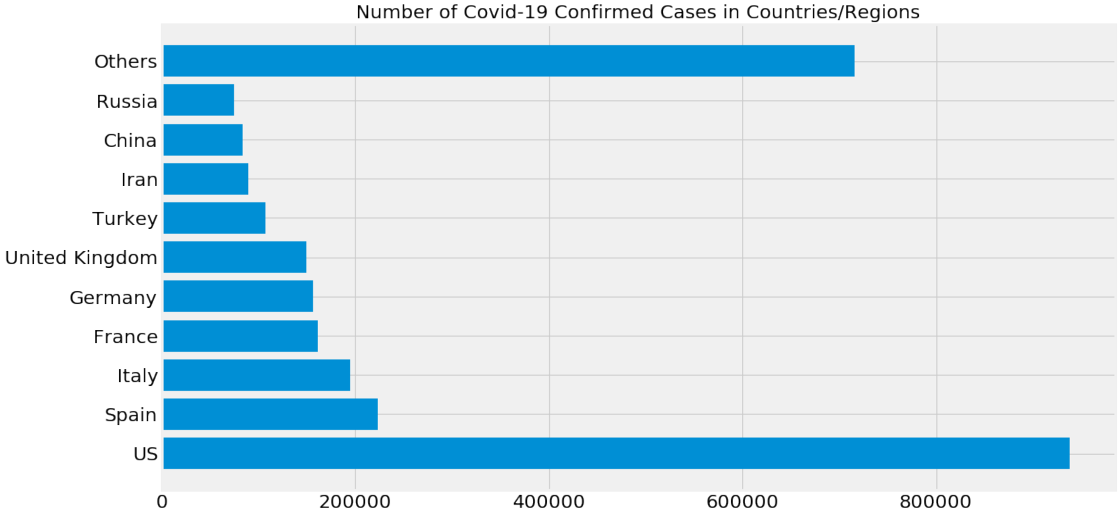
```
Outside USA 1958592 cases:
USA: 938154 cases
Total: 2896746 cases
```

[42]:
```
# Only show 10 countries with the most confirmed cases, the rest are grouped␣
↪into the other category
visual_unique_countries = []
visual_confirmed_cases = []
others = np.sum(country_confirmed_cases[10:])

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

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

[43]:
```
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()
```

[44]:

```
plot_bar_graphs(visual_unique_countries, visual_confirmed_cases, 'Number of␣
 ↪Covid-19 Confirmed Cases in Countries/Regions')
```

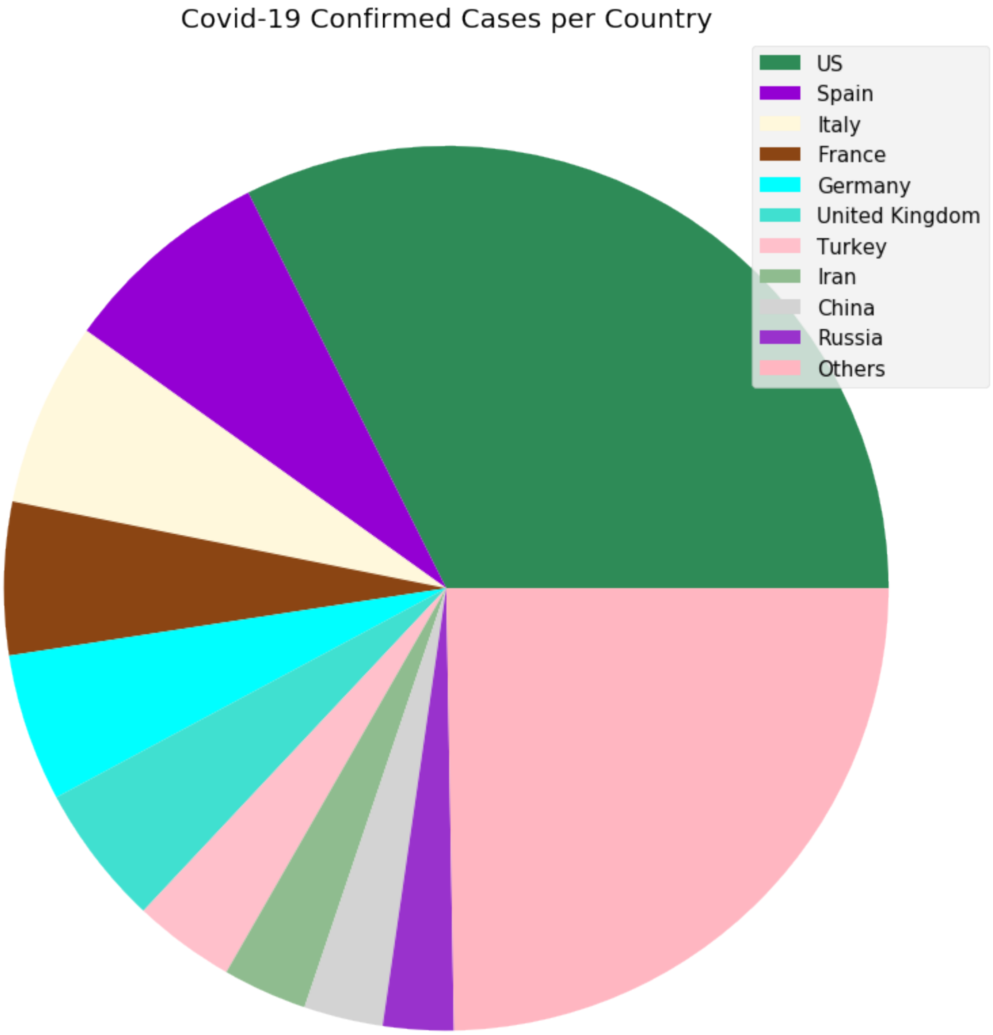


[45]:

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

[46]:

```
plot_pie_charts(visual_unique_countries, visual_confirmed_cases, 'Covid-19␣
 ↪Confirmed Cases per Country')
```

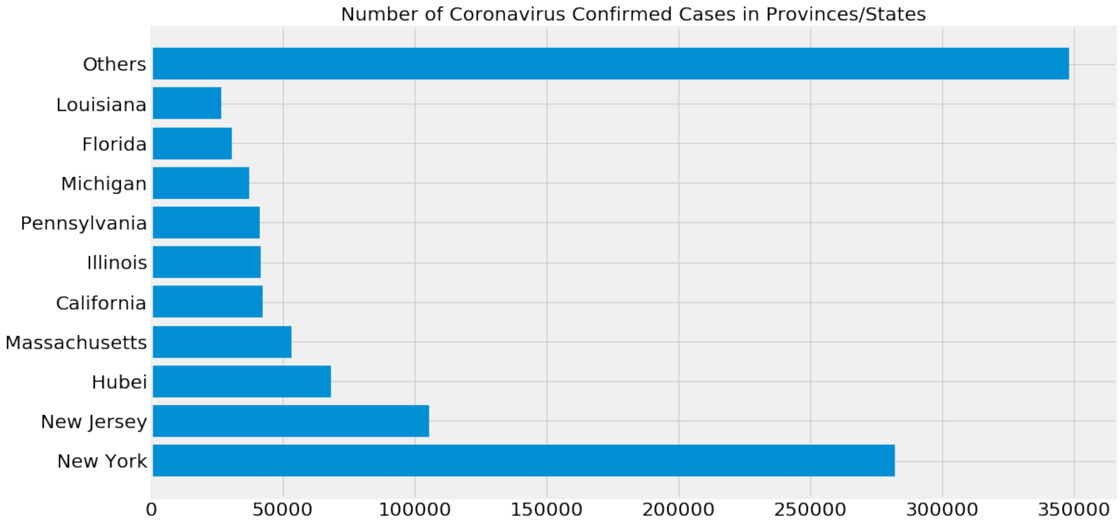


[47]:

```
# Only show 10 provinces with the most confirmed cases, the rest are grouped into the others category
visual_unique_provinces = []
visual_confirmed_cases2 = []
others = np.sum(province_confirmed_cases[10:])
for i in range(len(province_confirmed_cases[:10])):
    visual_unique_provinces.append(unique_provinces[i])
    visual_confirmed_cases2.append(province_confirmed_cases[i])

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

[48]:

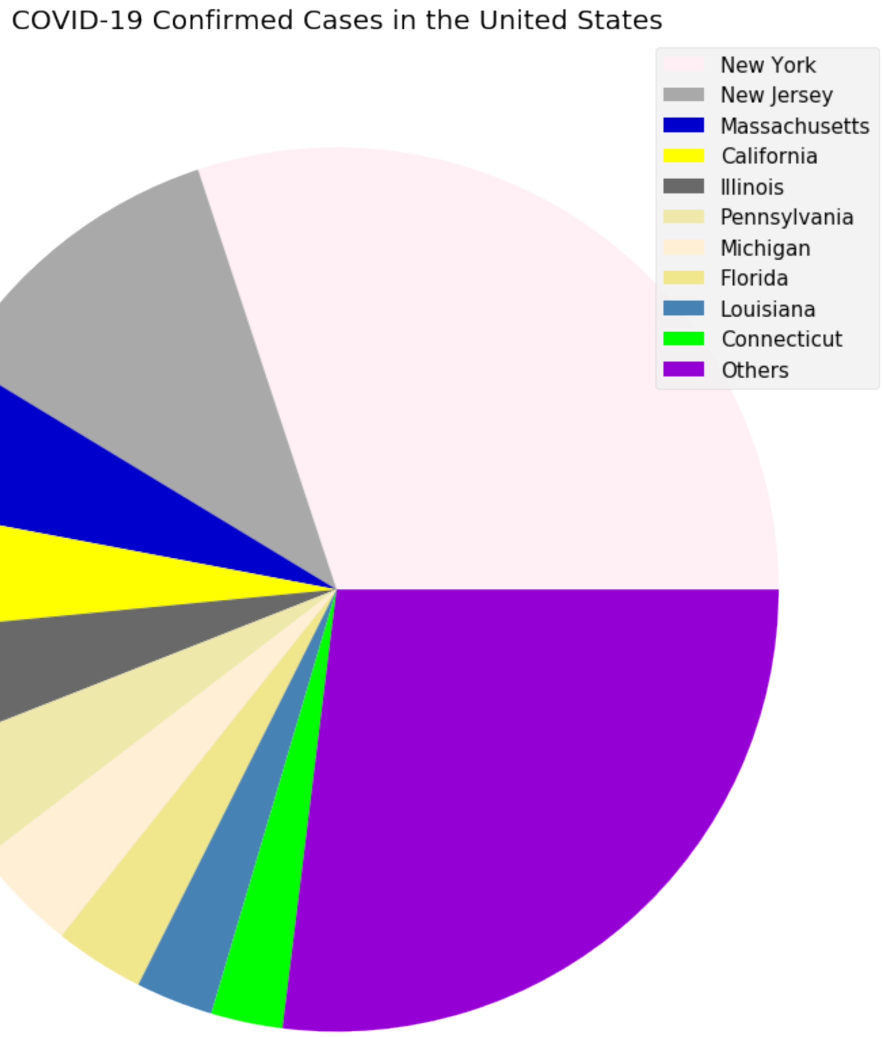
```
plot_bar_graphs(visual_unique_provinces, visual_confirmed_cases2, 'Number of␣
 ↪Coronavirus Confirmed Cases in Provinces/States')
```



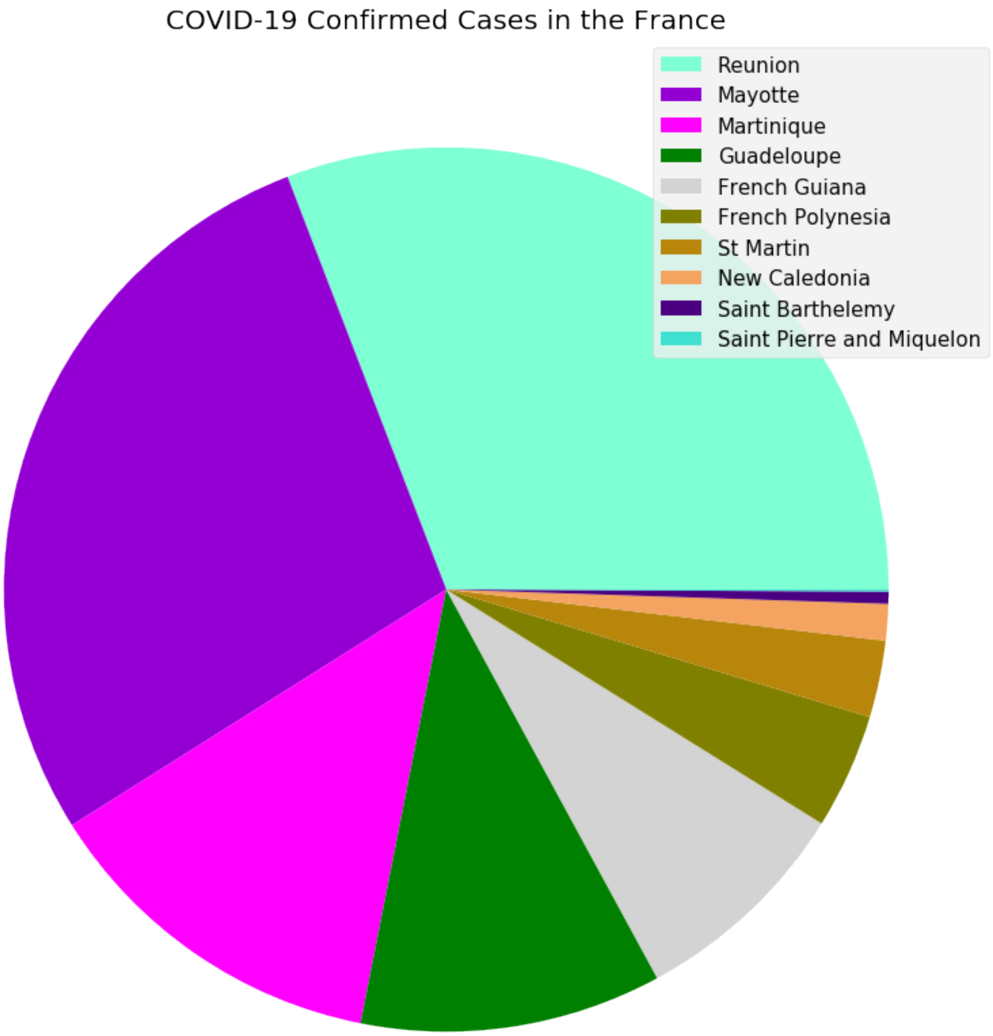
[49]:

```
def plot_pie_country_with_regions(country_name, title):
    regions =␣
 ↪list(latest_data[latest_data['Country_Region']==country_name]['Province_State'].
 ↪unique())
    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), reverse=True)]

    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)
```

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



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



```
[52]: 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)
```

```
[53]: 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]
```

[54]:

```
future_forecast
```

[54]:

```
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]])
```

[55]:

```
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'))
```

[56]:

```
X_train_confirmed, X_test_confirmed, y_train_confirmed, y_test_confirmed =␣
 ↪train_test_split(days_since_1_22, world_cases, test_size=0.25, shuffle=False)
```

[57]:

```
# transform our 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)
```

[58]:

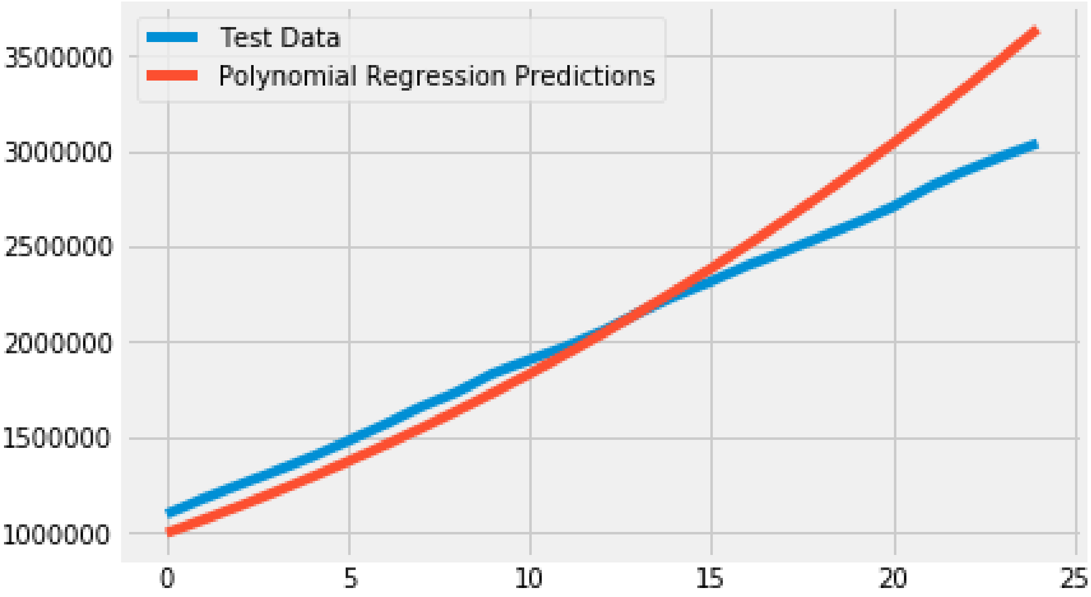
```
# polynomial regression
linear_model = LinearRegression(normalize=True, fit_intercept=False)
linear_model.fit(poly_X_train_confirmed, y_train_confirmed)
test_linear_pred = linear_model.predict(poly_X_test_confirmed)
linear_pred = linear_model.predict(poly_future_forecast)
```

```
print('MAE:', mean_absolute_error(test_linear_pred, y_test_confirmed))
print('MSE:',mean_squared_error(test_linear_pred, y_test_confirmed))
```

```
MAE: 172343.87626768733
MSE: 54097652301.452614
```

[59]:
```
plt.plot(y_test_confirmed)
plt.plot(test_linear_pred)
plt.legend(['Test Data', 'Polynomial Regression Predictions'])
```

[59]: `<matplotlib.legend.Legend at 0x204b7a75c88>`

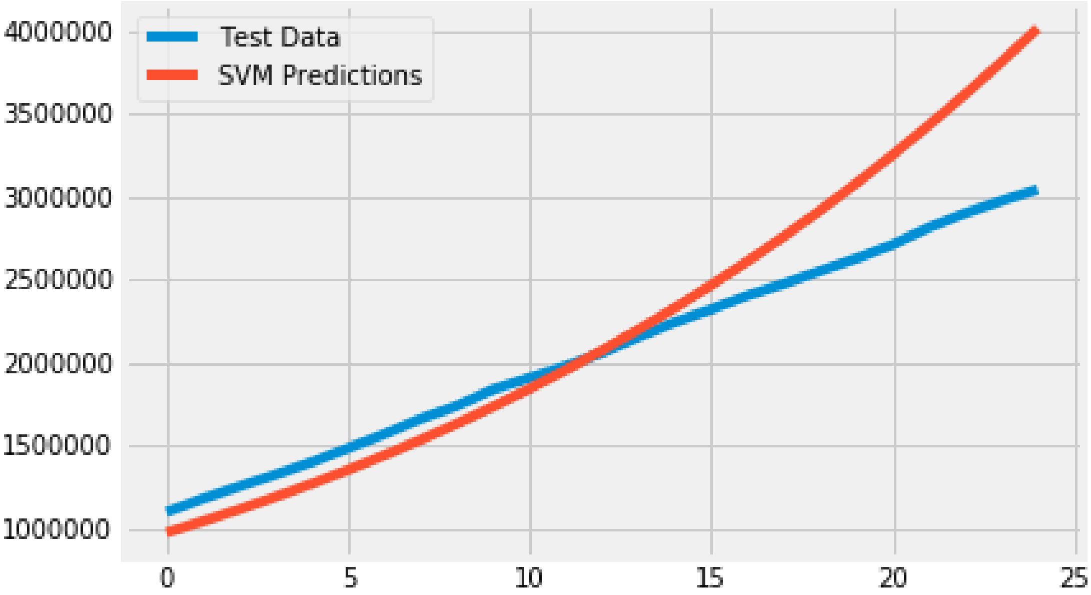


[60]:
```
# svm_confirmed = svm_search.best_estimator_
svm_confirmed = SVR(shrinking=True, kernel='poly',gamma=0.01,␣
 ↪epsilon=1,degree=5, C=0.1)
svm_confirmed.fit(X_train_confirmed, y_train_confirmed)
svm_pred = svm_confirmed.predict(future_forecast)
```

```
C:\ProgramData\Anaconda3\lib\site-packages\sklearn\utils\validation.py:724:
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().
  y = column_or_1d(y, warn=True)
```

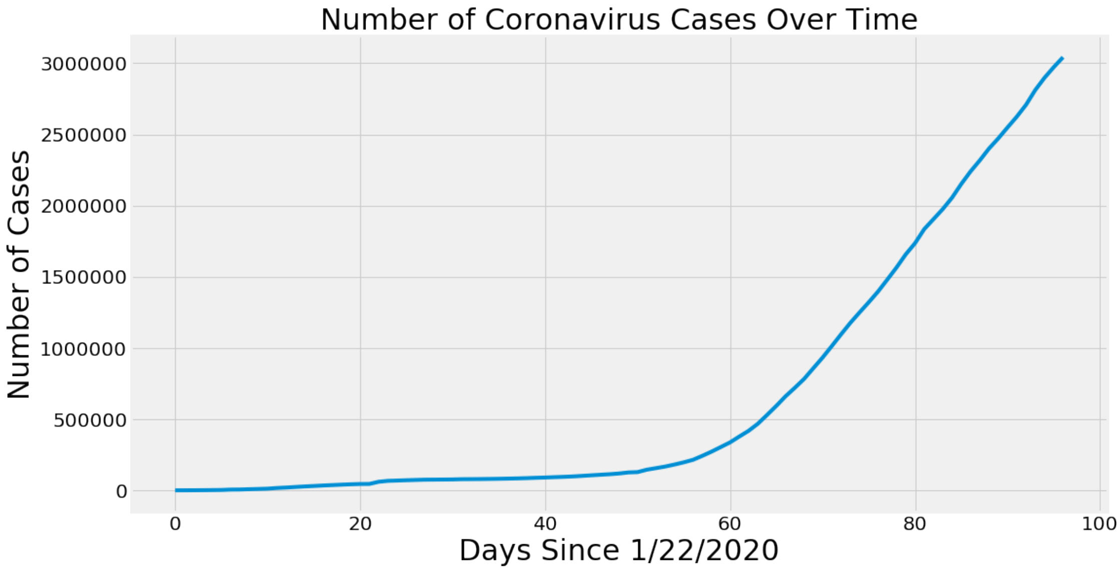
[61]:
```
svm_test_pred = svm_confirmed.predict(X_test_confirmed)
plt.plot(y_test_confirmed)
plt.plot(svm_test_pred)
plt.legend(['Test Data', 'SVM Predictions'])
print('MAE:', mean_absolute_error(svm_test_pred, y_test_confirmed))
print('MSE:',mean_squared_error(svm_test_pred, y_test_confirmed))
```

```
MAE: 265489.6552454628
MSE: 139718600180.29932
```



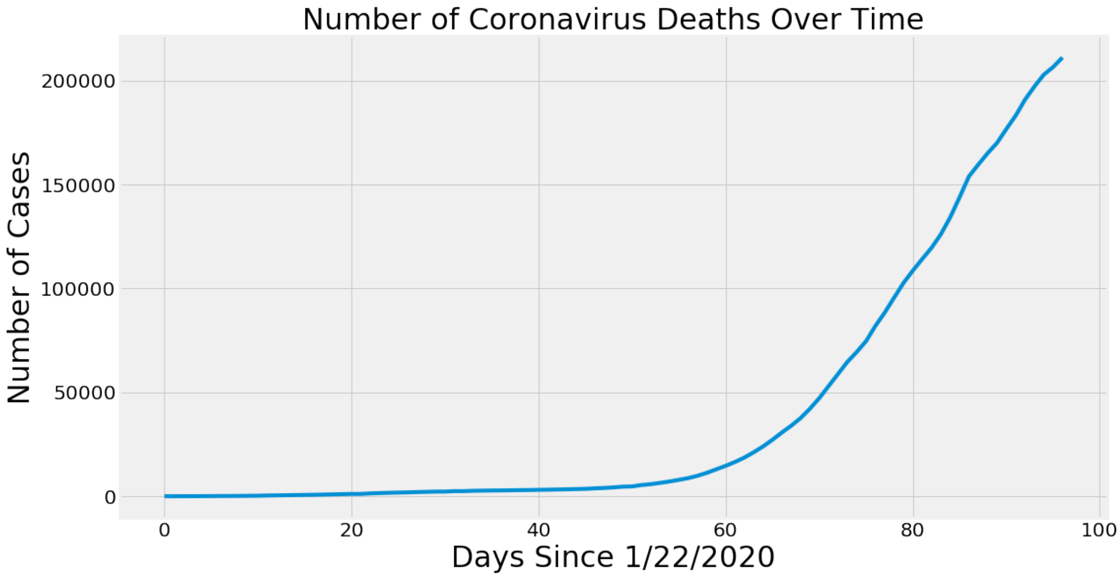
[62]:

```
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()
```



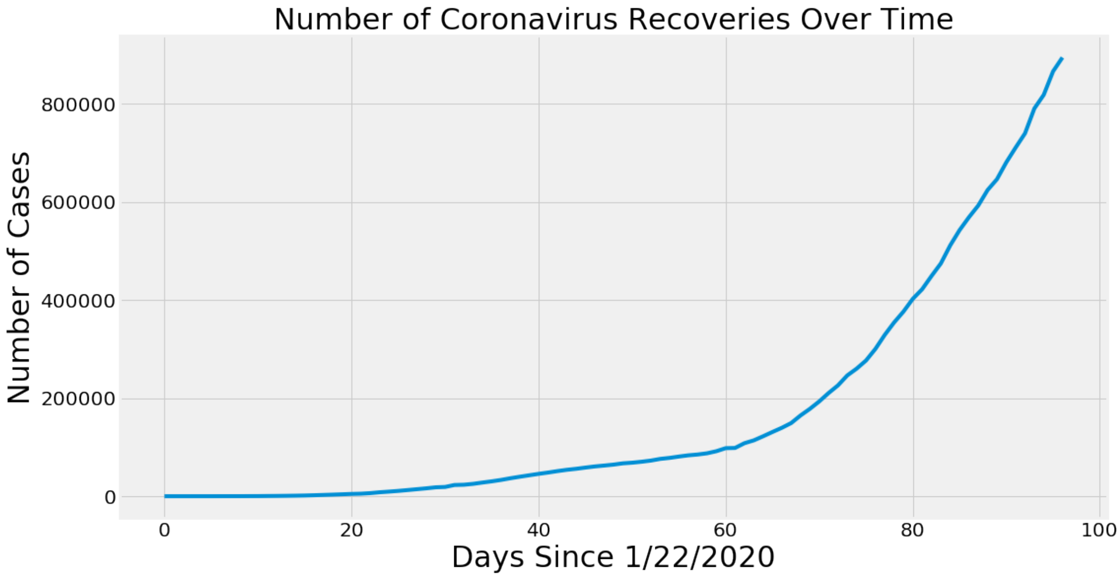
[63]:

```
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()
```



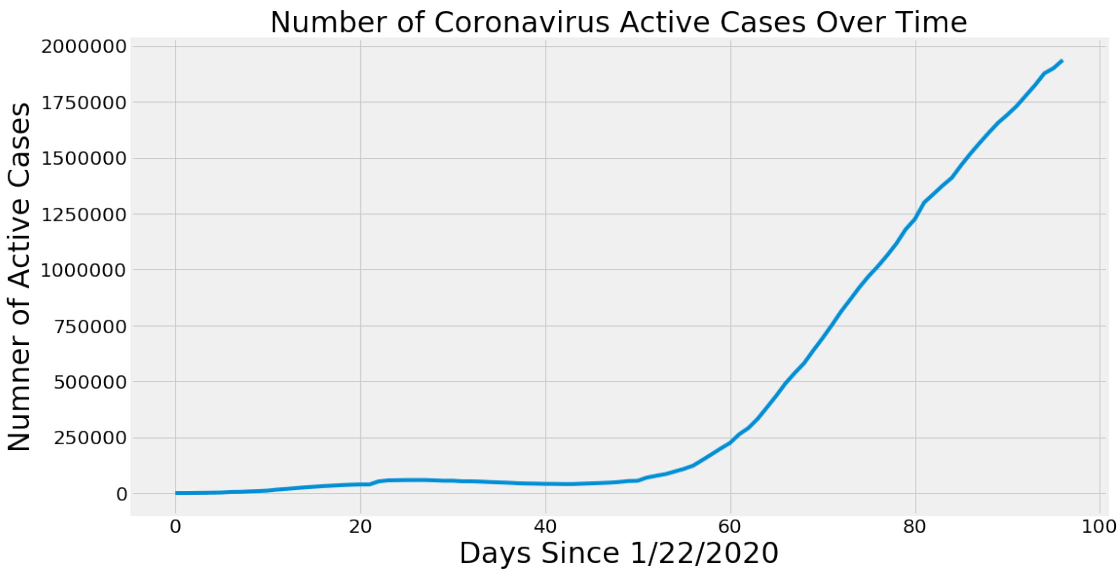
[64]:

```
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()
```

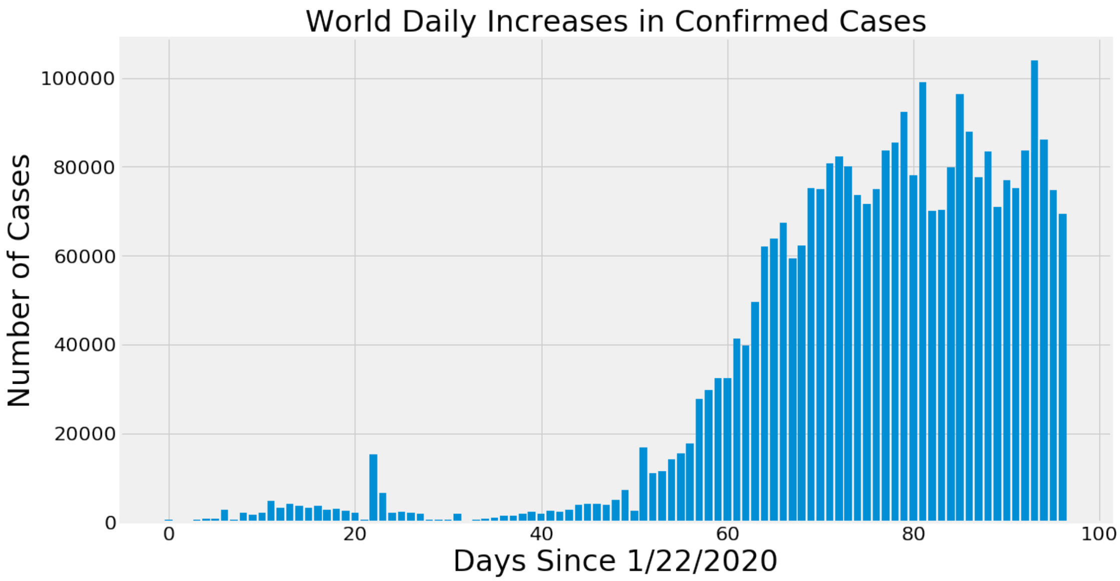


[65]:

```
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('Numner of Active Cases', size=30)
plt.xticks(size=20)
plt.yticks(size=20)
plt.show()
```

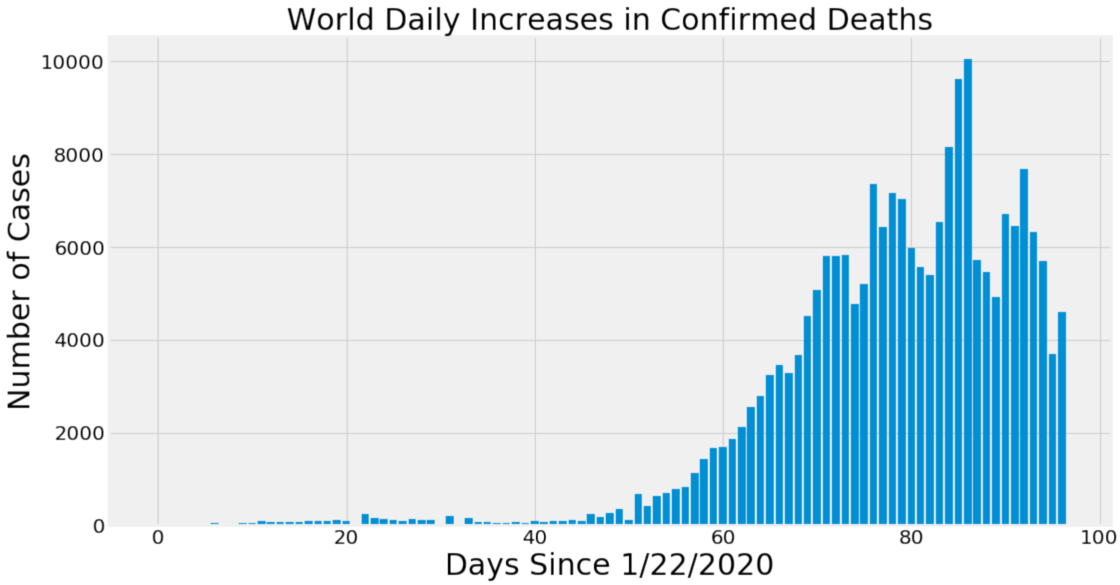


```
[66]: plt.figure(figsize=(16, 9))
      plt.bar(adjusted_dates, world_daily_increase)
      plt.title('World Daily Increases in Confirmed Cases', size=30)
      plt.xlabel('Days Since 1/22/2020', size=30)
      plt.ylabel('Number of Cases', size=30)
      plt.xticks(size=20)
      plt.yticks(size=20)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



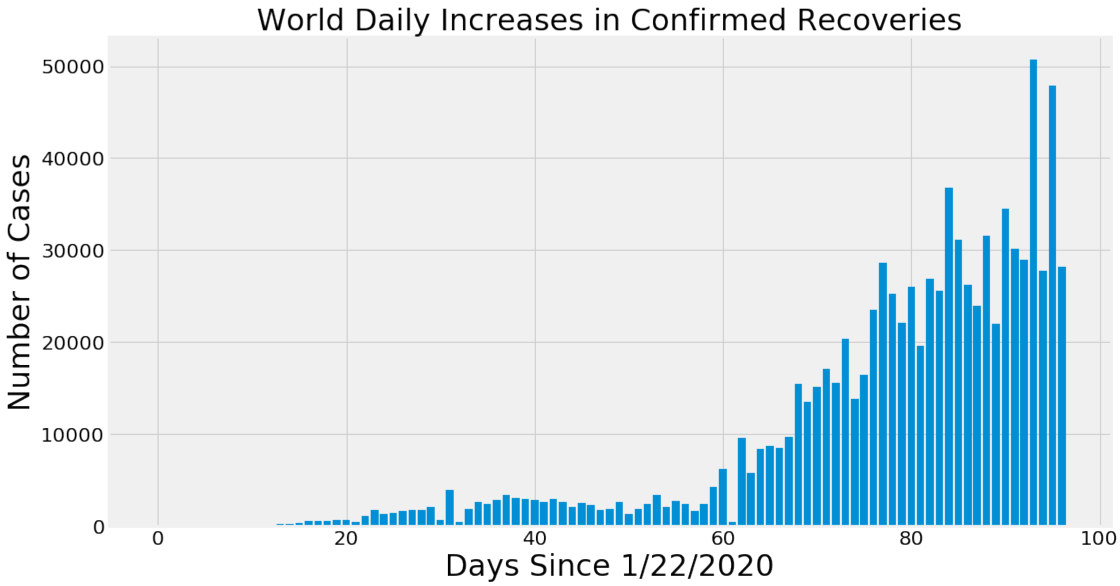
[67]:

```
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()
```



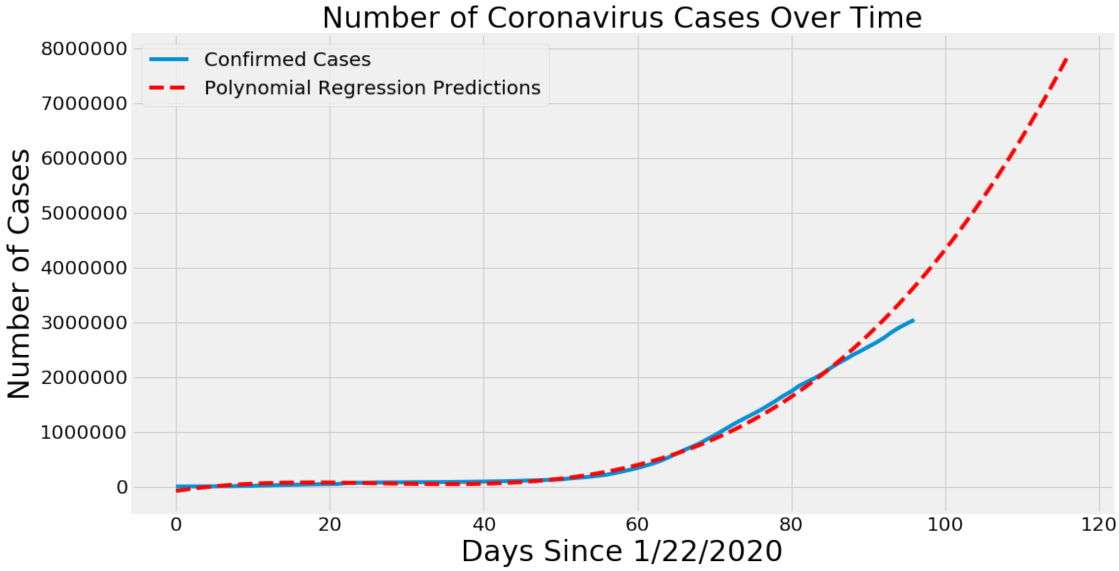
[68]:

```
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()
```

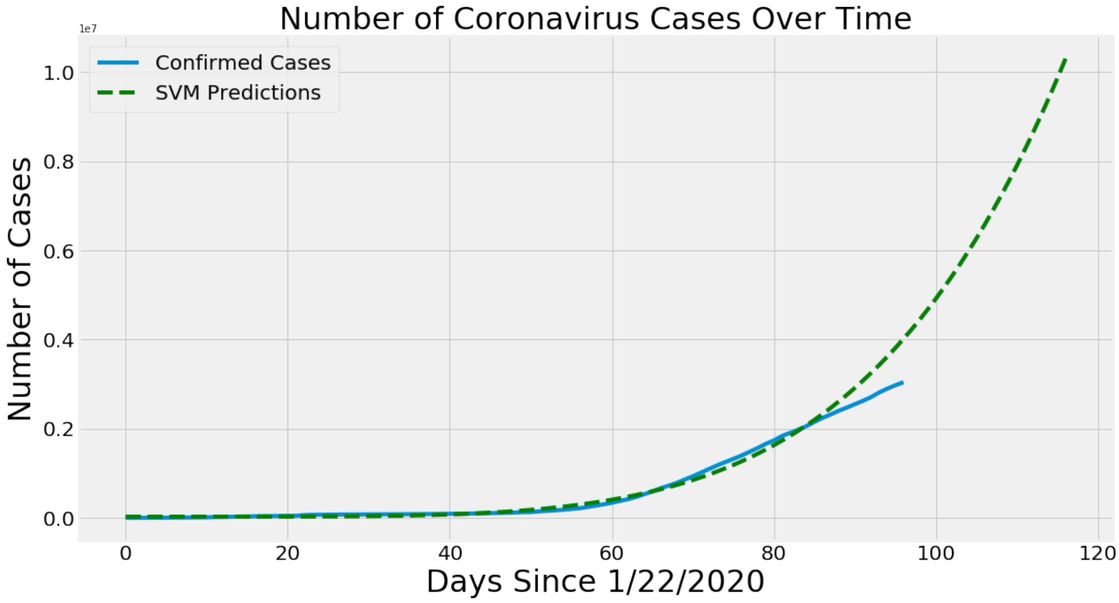


[69]:
```
def plot_predictions(x, y, pred, algo_name, color):
    plt.figure(figsize=(16, 9))
    plt.plot(x, y)
    plt.plot(future_forecast, pred, linestyle='dashed', color=color)
    plt.title('Number of Coronavirus Cases Over Time', size=30)
    plt.xlabel('Days Since 1/22/2020', size=30)
    plt.ylabel('Number of Cases', size=30)
    plt.legend(['Confirmed Cases', algo_name], prop={'size': 20})
    plt.xticks(size=20)
    plt.yticks(size=20)
    plt.show()
```

[70]:
```
plot_predictions(adjusted_dates, world_cases, linear_pred, 'Polynomial␣
 ↪Regression Predictions', 'red')
```



[71]:
```
plot_predictions(adjusted_dates, world_cases, svm_pred, 'SVM Predictions',␣
  ↪'green')
```



[72]:
```
# 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 Confirmed Cases Worldwide': np.round(linear_pred[-20:])})
poly_df
```

[72]:

```
          Date  Predicted number of Confirmed Cases Worldwide
0   04/28/2020                                      3805551.0
1   04/29/2020                                      3972542.0
2   04/30/2020                                      4144342.0
3   05/01/2020                                      4321018.0
4   05/02/2020                                      4502638.0
5   05/03/2020                                      4689268.0
6   05/04/2020                                      4880976.0
7   05/05/2020                                      5077830.0
8   05/06/2020                                      5279895.0
9   05/07/2020                                      5487241.0
10  05/08/2020                                      5699933.0
11  05/09/2020                                      5918039.0
12  05/10/2020                                      6141626.0
13  05/11/2020                                      6370762.0
14  05/12/2020                                      6605513.0
15  05/13/2020                                      6845948.0
16  05/14/2020                                      7092132.0
17  05/15/2020                                      7344134.0
18  05/16/2020                                      7602021.0
19  05/17/2020                                      7865860.0
```

[73]:

```
# Future predictions using SVM
svm_df = pd.DataFrame({'Date': future_forecast_dates[-20:], 'SVM Predicted # of␣
 ↪Confirmed Cases Worldwide': np.round(svm_pred[-20:])})
svm_df
```

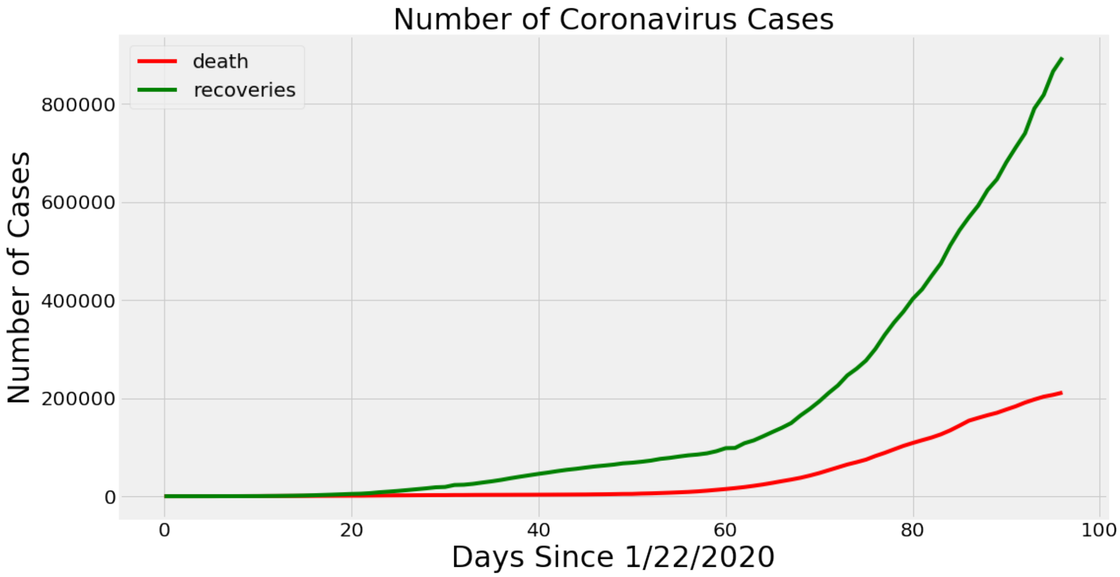
[73]:

```
          Date  SVM Predicted # of Confirmed Cases Worldwide
0   04/28/2020                                     4225560.0
1   04/29/2020                                     4446542.0
2   04/30/2020                                     4676730.0
3   05/01/2020                                     4916409.0
4   05/02/2020                                     5165869.0
5   05/03/2020                                     5425407.0
6   05/04/2020                                     5695325.0
7   05/05/2020                                     5975932.0
8   05/06/2020                                     6267541.0
9   05/07/2020                                     6570473.0
10  05/08/2020                                     6885054.0
11  05/09/2020                                     7211618.0
12  05/10/2020                                     7550504.0
13  05/11/2020                                     7902057.0
14  05/12/2020                                     8266628.0
15  05/13/2020                                     8644577.0
16  05/14/2020                                     9036267.0
17  05/15/2020                                     9442071.0
18  05/16/2020                                     9862366.0
```

```
19  05/17/2020                                 10297537.0
```

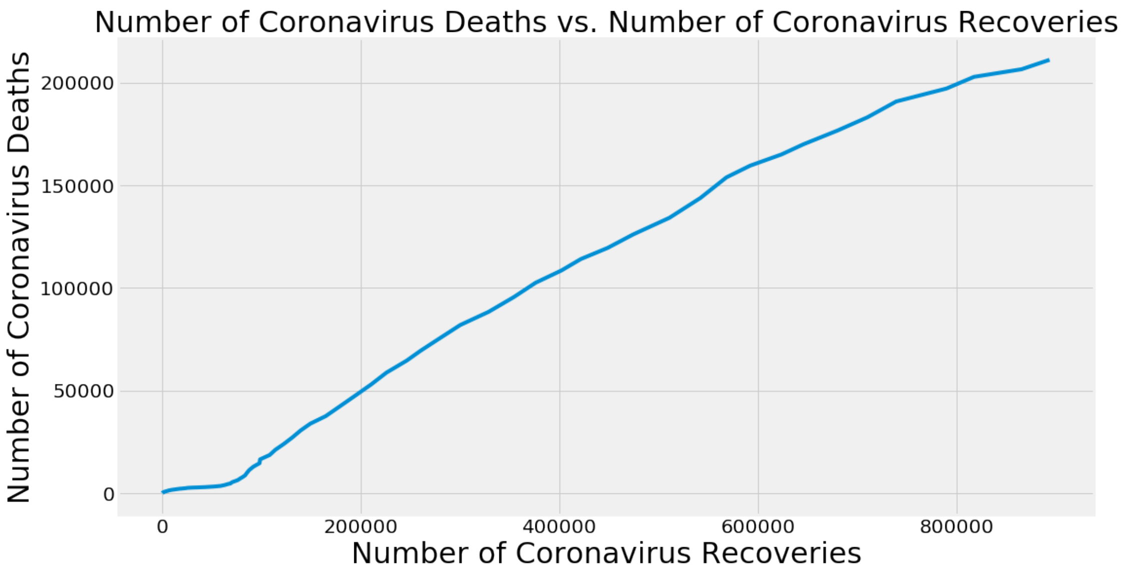
[74]:

```
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()
```



[75]:

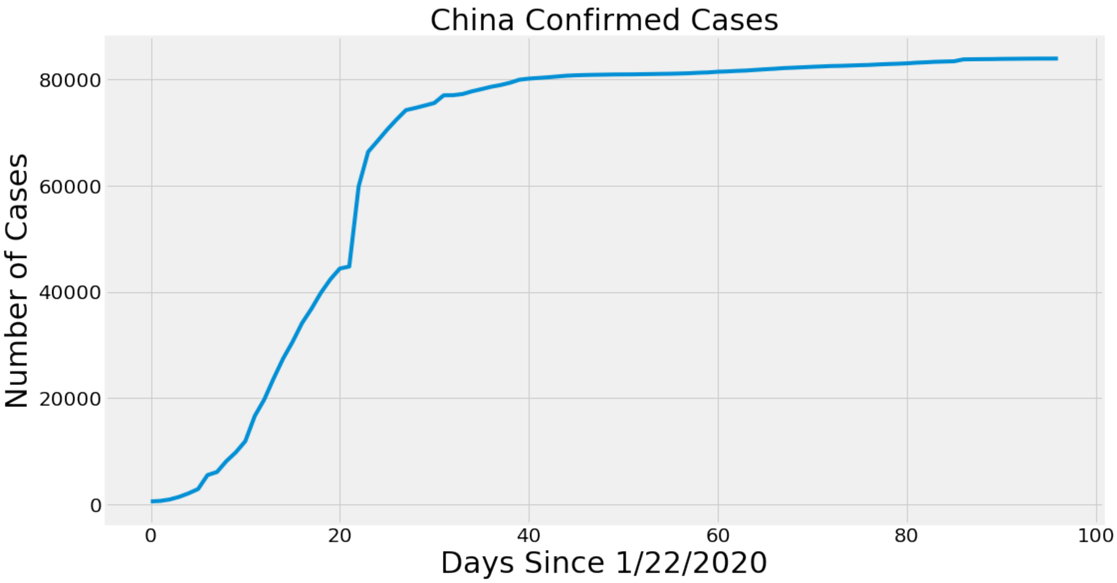
```
plt.figure(figsize=(16, 9))
plt.plot(total_recovered, total_deaths)
plt.title('Number of Coronavirus Deaths vs. Number of Coronavirus Recoveries',␣
 ↪size=30)
plt.xlabel('Number of Coronavirus Recoveries', size=30)
plt.ylabel('Number of Coronavirus Deaths', size=30)
plt.xticks(size=20)
plt.yticks(size=20)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

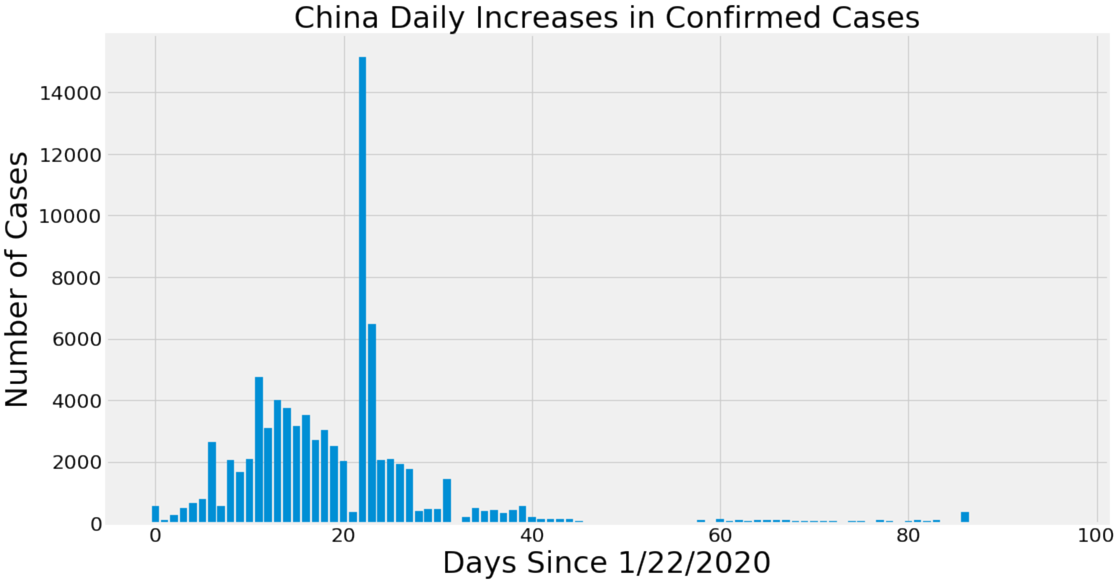


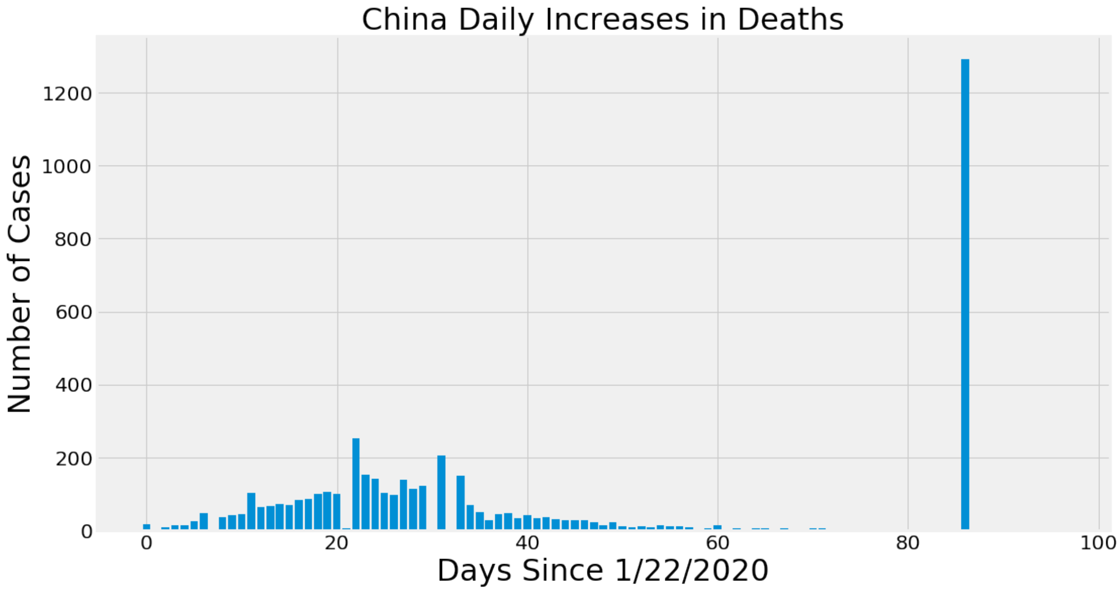
[76]:

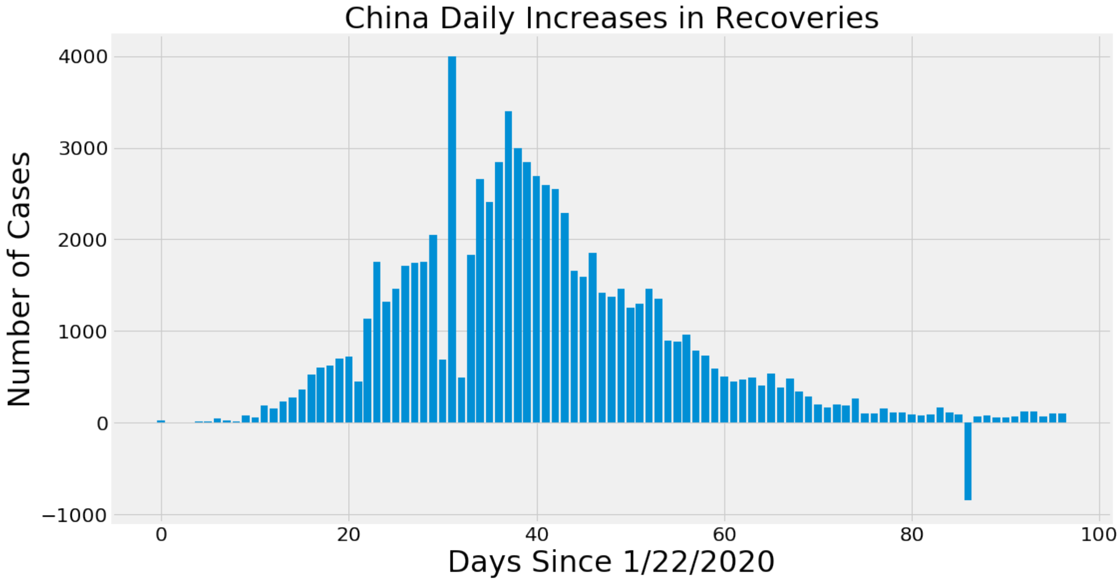
```python
def country_plot(x, y1, y2, y3, y4, country):
    plt.figure(figsize=(16, 9))
    plt.plot(x, y1)
    plt.title('{} Confirmed Cases'.format(country), size=30)
    plt.xlabel('Days Since 1/22/2020', size=30)
    plt.ylabel('Number of Cases', size=30)
    plt.xticks(size=20)
    plt.yticks(size=20)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

[77]:
```
country_plot(adjusted_dates, china_cases, china_daily_increase,␣
 ↪china_daily_death, china_daily_recovery, 'China')
```



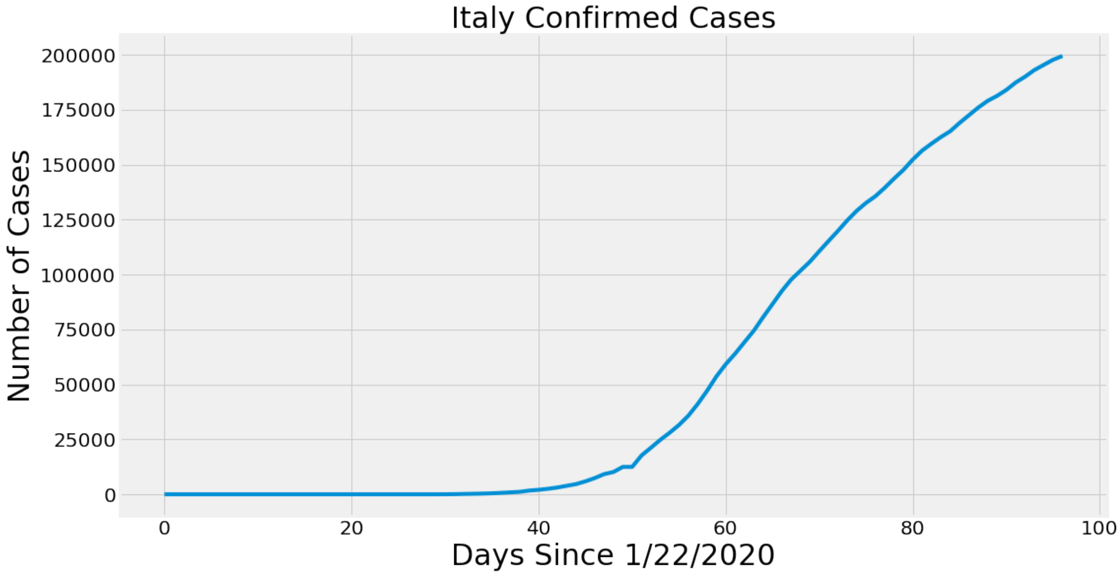


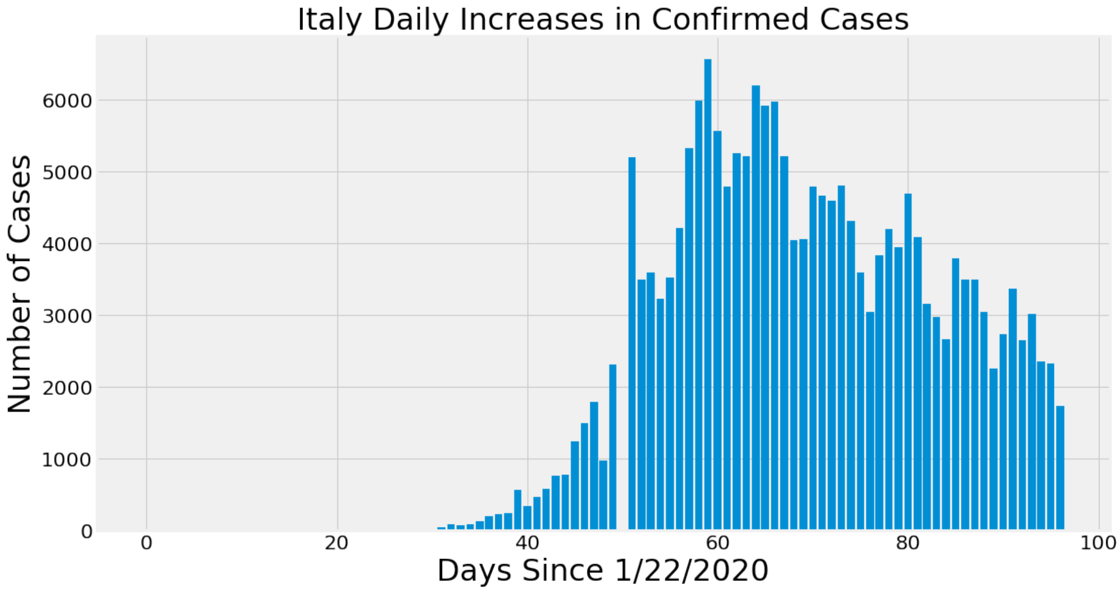


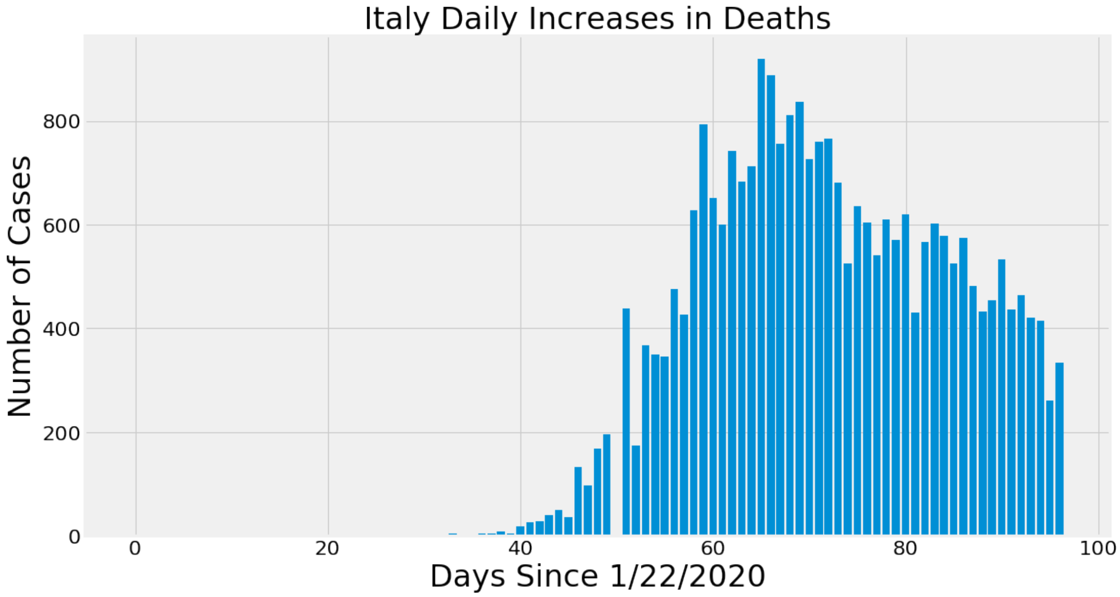


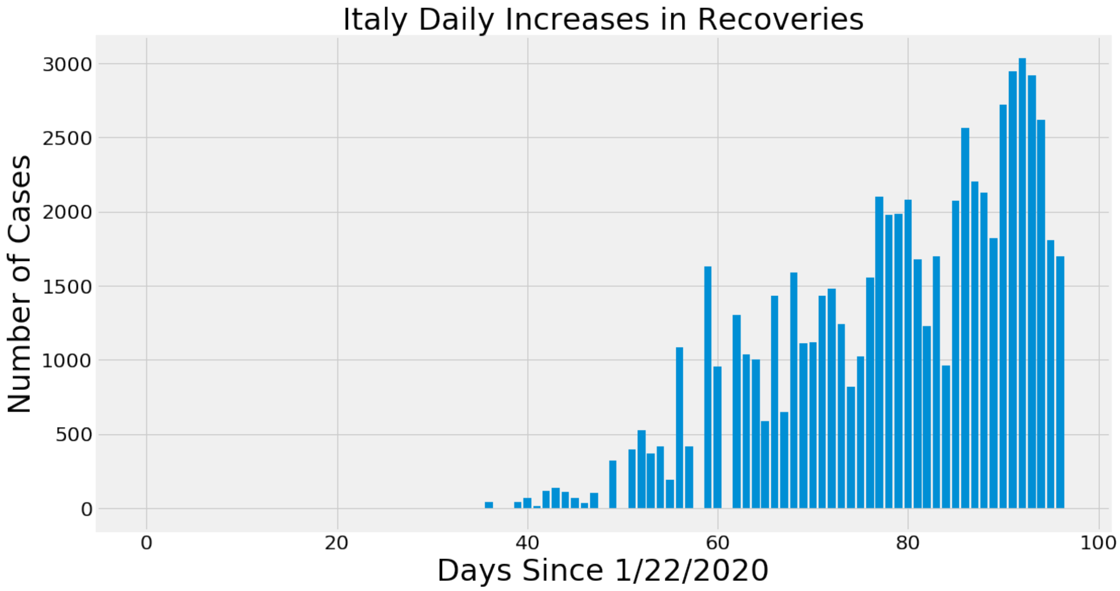
[78]:

```
country_plot(adjusted_dates, italy_cases, italy_daily_increase,␣
 ↪italy_daily_death, italy_daily_recovery, 'Italy')
```

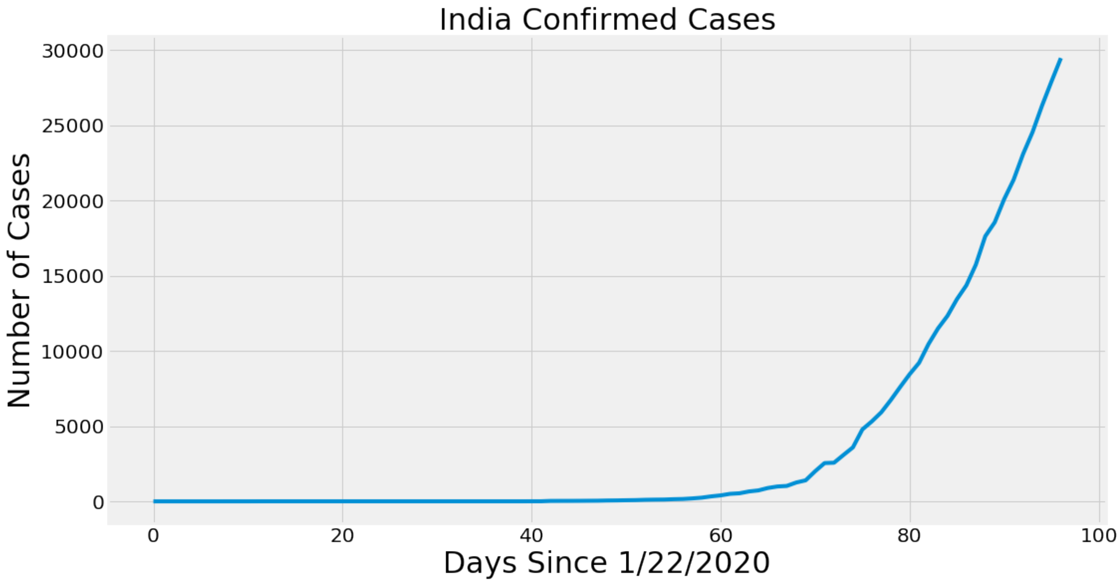

Italy Confirmed Cases
Number of Cases
200000
175000
150000
125000
100000
75000
50000
25000
0
0
20
40
60
80
100
Days Since 1/22/2020

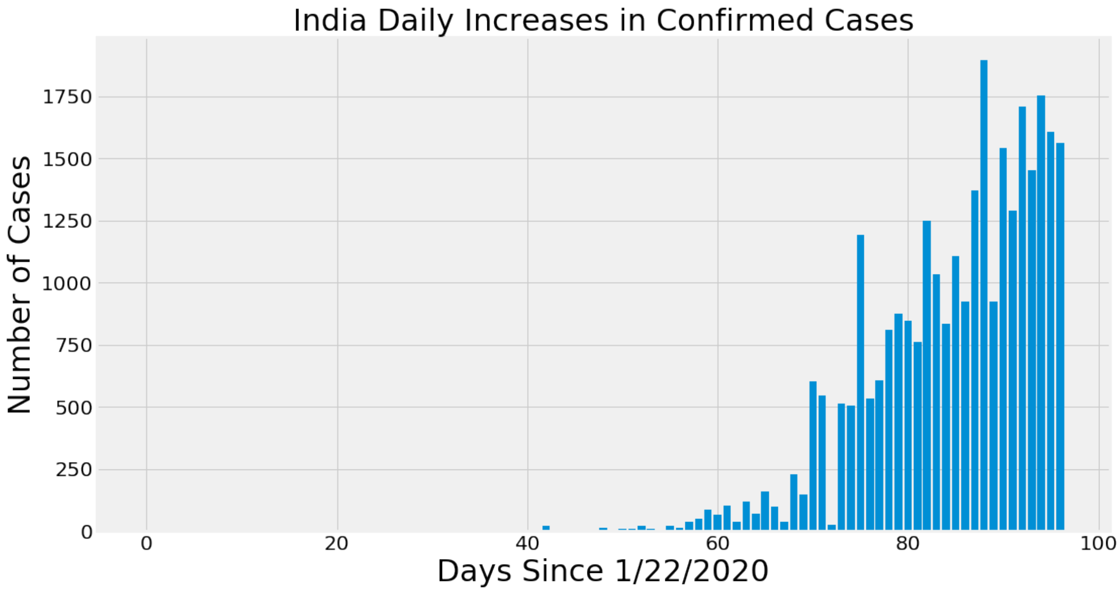

Italy Daily Increases in Confirmed Cases
Number of Cases
6000
5000
4000
3000
2000
1000
0
0
20
40
60
80
100
Days Since 1/22/2020

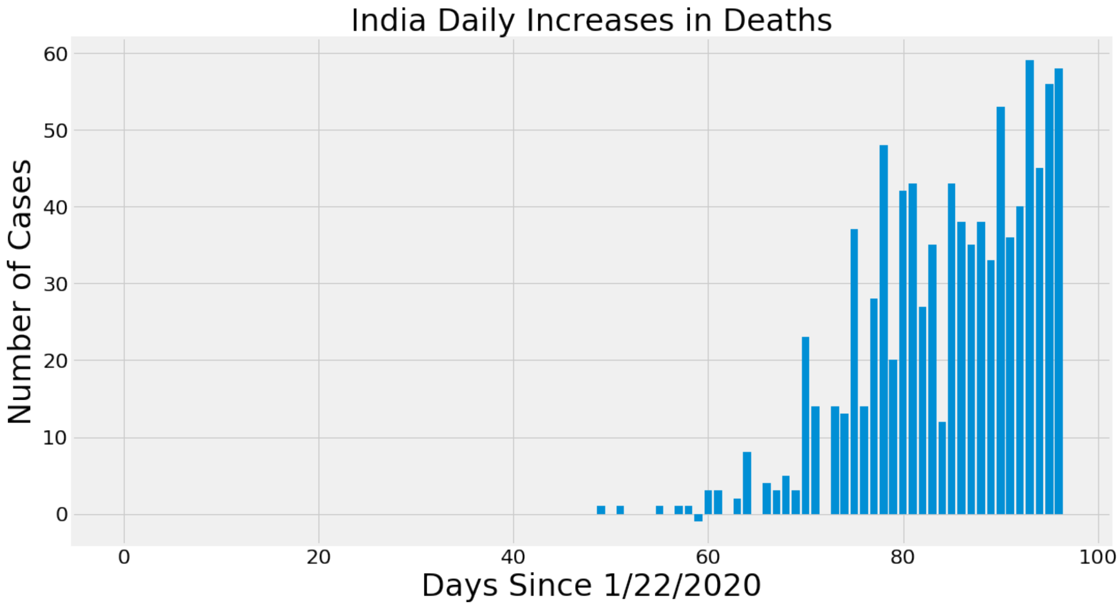


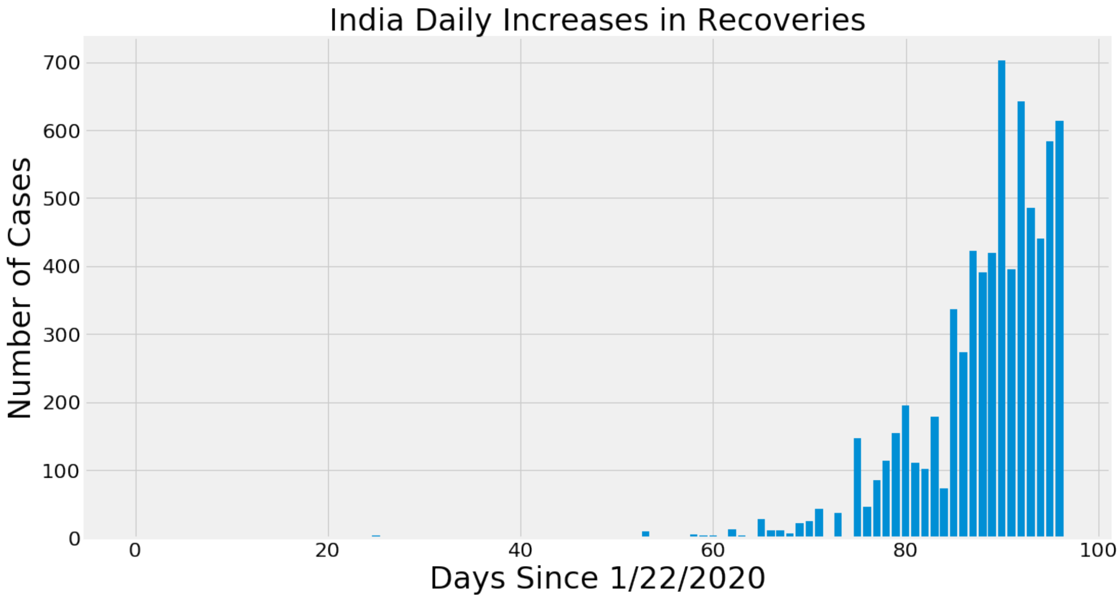


[79]:
```
country_plot(adjusted_dates, india_cases, india_daily_increase,␣
 ↪india_daily_death, india_daily_recovery, 'India')
```

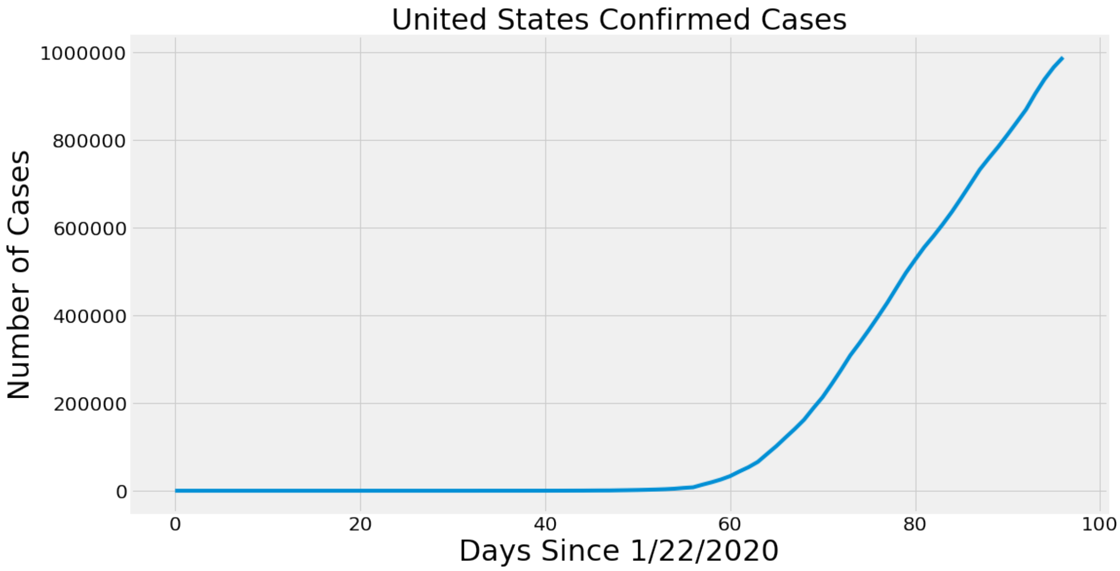

India Confirmed Cases
Number of Cases
30000
25000
20000
15000
10000
5000
0
0
20
40
60
80
100
Days Since 1/22/2020

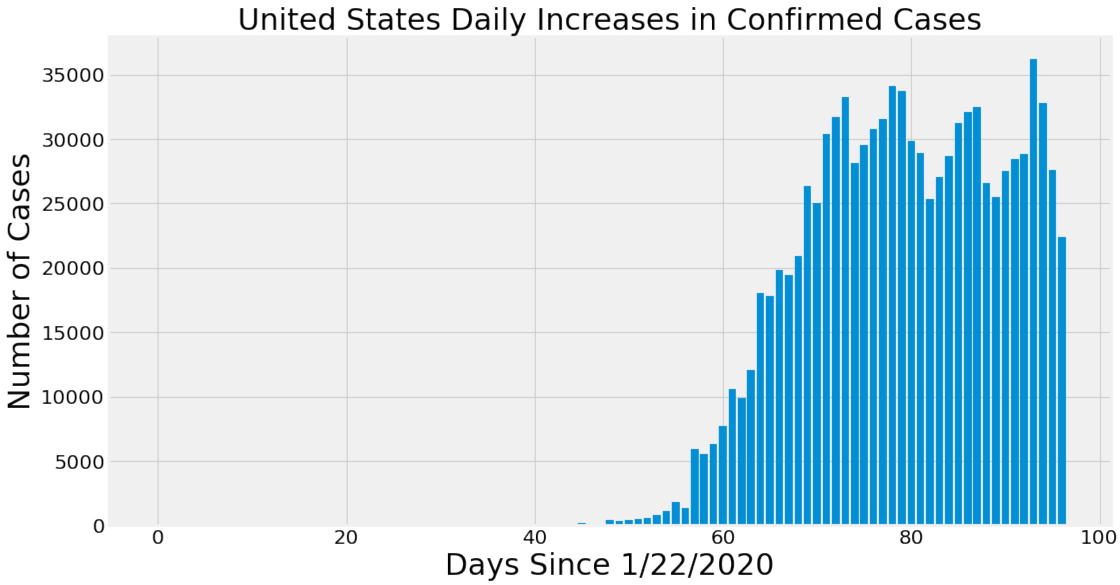

India Daily Increases in Confirmed Cases
Number of Cases
1750
1500
1250
1000
750
500
250
0
0
20
40
60
80
100
Days Since 1/22/2020

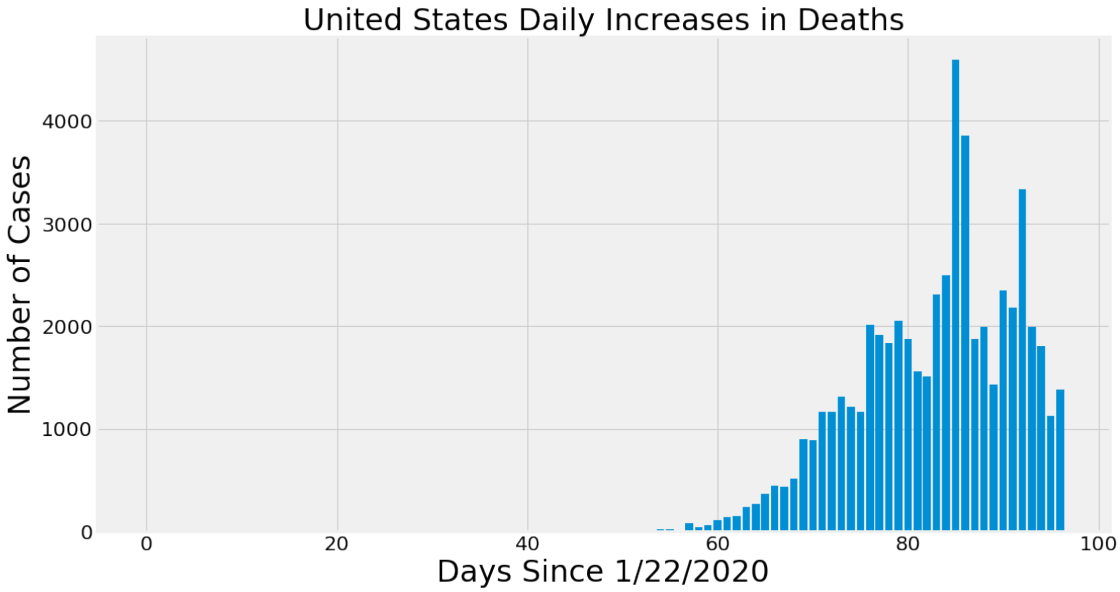


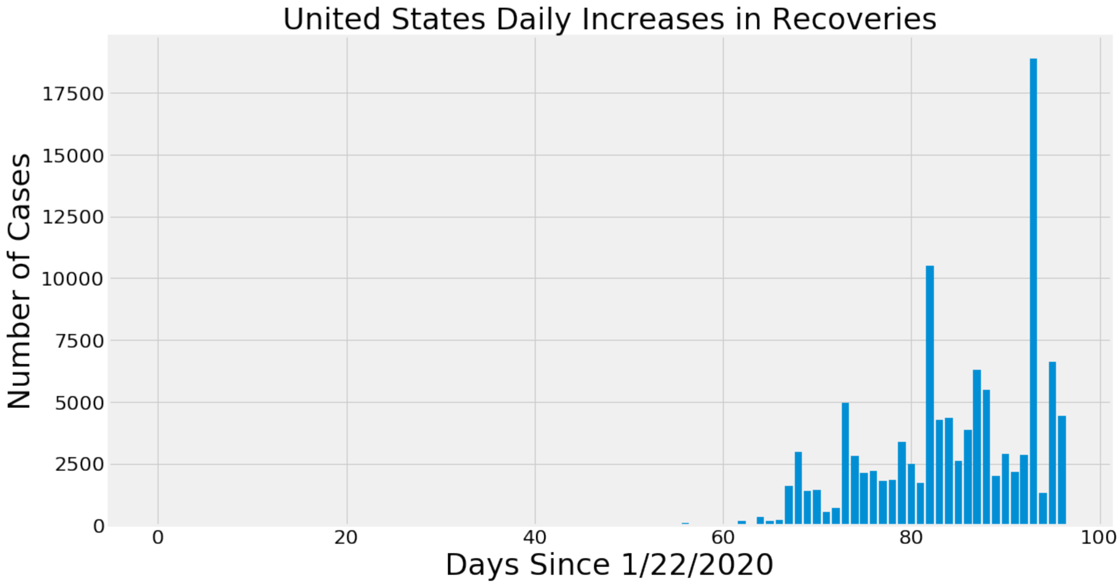


[80]:
```
country_plot(adjusted_dates, us_cases, us_daily_increase, us_daily_death,␣
 ↪us_daily_recovery, 'United States')
```

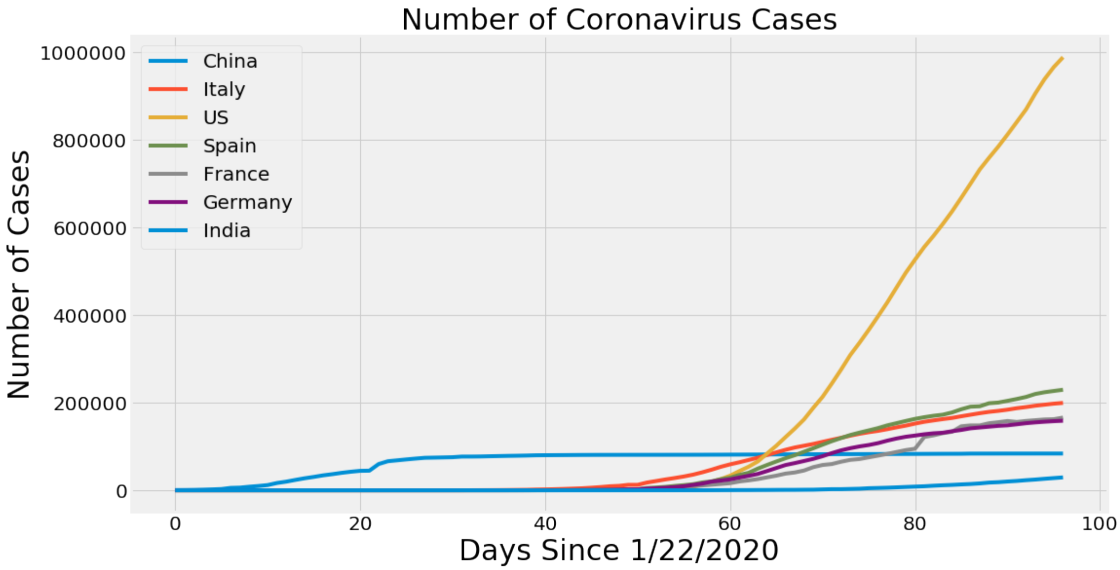

United States Confirmed Cases
Number of Cases
1000000
800000
600000
400000
200000
0
0
20
40
60
80
100
Days Since 1/22/2020


United States Daily Increases in Confirmed Cases
Number of Cases
35000
30000
25000
20000
15000
10000
5000
0
0
20
40
60
80
100
Days Since 1/22/2020

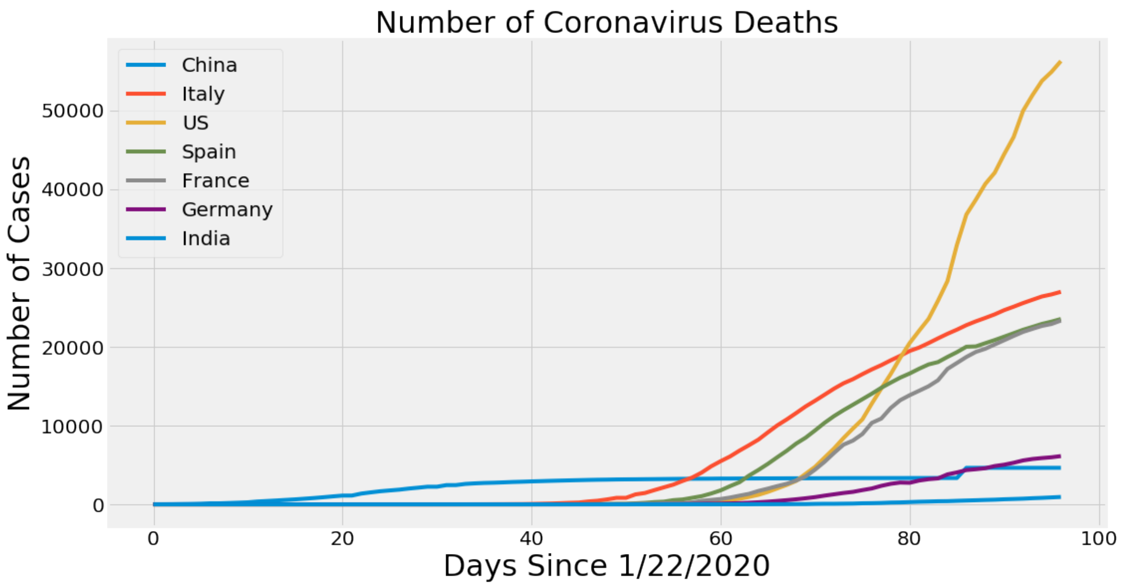




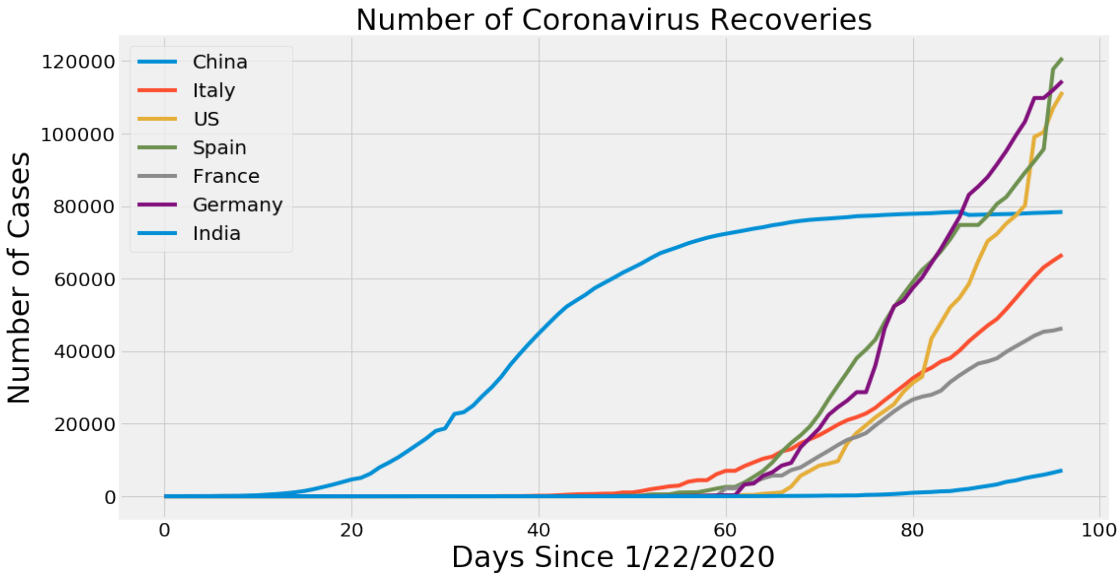
[81]:
```
plt.figure(figsize=(16, 9))
plt.plot(adjusted_dates, china_cases)
plt.plot(adjusted_dates, italy_cases)
plt.plot(adjusted_dates, us_cases)
plt.plot(adjusted_dates, spain_cases)
plt.plot(adjusted_dates, france_cases)
plt.plot(adjusted_dates, germany_cases)
plt.plot(adjusted_dates, india_cases)
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={'size': 20})
plt.xticks(size=20)
plt.yticks(size=20)
plt.show()
```



```
[82]: 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={'size': 20})
plt.xticks(size=20)
plt.yticks(size=20)
plt.show()
```



[83]:
```
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={'size': 20})
plt.xticks(size=20)
plt.yticks(size=20)
plt.show()
```


Number of Coronavirus Recoveries
China
Italy
US
Spain
France
Germany
India
120000
100000
80000
60000
40000
20000
0
Number of Cases
0
20
40
60
80
100
Days Since 1/22/2020

```
[ ]:
```