# Covid-19 Data Analysis

# 1

### *Download the data from the given link*

# 2

### *Create a DataFrame from the CSV file.*

In [104]:

```
import pandas as pd
data = pd.read_csv("time-series-19-covid-combined.csv")
df=pd.DataFrame(data)
```

In [105]:

```
df
```

Out[105]:

| | Date | Country/Region | Province/State | Confirmed | Recovered | Deaths |
|---|---|---|---|---|---|---|
| **0** | 2020-01-22 | Afghanistan | NaN | 0 | 0.0 | 0 |
| **1** | 2020-01-23 | Afghanistan | NaN | 0 | 0.0 | 0 |
| **2** | 2020-01-24 | Afghanistan | NaN | 0 | 0.0 | 0 |
| **3** | 2020-01-25 | Afghanistan | NaN | 0 | 0.0 | 0 |
| **4** | 2020-01-26 | Afghanistan | NaN | 0 | 0.0 | 0 |
| **...** | ... | ... | ... | ... | ... | ... |
| **154561** | 2021-07-24 | Zimbabwe | NaN | 97277 | 64628.0 | 3050 |
| **154562** | 2021-07-25 | Zimbabwe | NaN | 97894 | 65913.0 | 3094 |
| **154563** | 2021-07-26 | Zimbabwe | NaN | 99944 | 67827.0 | 3173 |
| **154564** | 2021-07-27 | Zimbabwe | NaN | 101711 | 70496.0 | 3280 |
| **154565** | 2021-07-28 | Zimbabwe | NaN | 103567 | 71383.0 | 3340 |

154566 rows × 6 columns

# 3

### *Merge the data for countries with multiple regions in order to provide a single time-series for each country.*

In [89]:

```
df_new = df.groupby(['Date', 'Country/Region'])["Confirmed", "Recovered",'Deaths'].appl
```

In [90]:

```
df_new
```

Out[90]:

```
                                    Confirmed  Recovered  Deaths
      Date       Country/Region
2020-01-22          Afghanistan           0.0        0.0     0.0
                        Albania           0.0        0.0     0.0
                        Algeria           0.0        0.0     0.0
                        Andorra           0.0        0.0     0.0
                         Angola           0.0        0.0     0.0
       ...                  ...           ...        ...     ...
2021-07-28              Vietnam      123640.0    27457.0   630.0
             West Bank and Gaza      316328.0   311622.0  3601.0
                          Yemen        7027.0     4168.0  1374.0
                         Zambia      193432.0   183957.0  3338.0
                       Zimbabwe      103567.0    71383.0  3340.0

108030 rows × 3 columns
```

# 4

***Print the total number of confirmed cases and number of deaths in each country in the last reported day. What are the 10 countries with the highest number of confirmed COVID-19 cases? What are the 10 countries with the highest number of deaths?***

In [37]:

```
df_new.reset_index(level=0, inplace=True)
latest=df_new['Date'].max()
df_1 = df_new[df_new['Date'] == latest]
```

In [38]:

```
df_1
```

Out[38]:

| | Date | Confirmed | Recovered | Deaths |
|---|---|---|---|---|
| **Country/Region** | | | | |
| **Afghanistan** | 2021-07-28 | 145552.0 | 82586.0 | 6577.0 |
| **Albania** | 2021-07-28 | 132952.0 | 130174.0 | 2457.0 |
| **Algeria** | 2021-07-28 | 167131.0 | 112900.0 | 4161.0 |
| **Andorra** | 2021-07-28 | 14586.0 | 14113.0 | 127.0 |
| **Angola** | 2021-07-28 | 42288.0 | 35964.0 | 1000.0 |
| **...** | ... | ... | ... | ... |
| **Vietnam** | 2021-07-28 | 123640.0 | 27457.0 | 630.0 |
| **West Bank and Gaza** | 2021-07-28 | 316328.0 | 311622.0 | 3601.0 |
| **Yemen** | 2021-07-28 | 7027.0 | 4168.0 | 1374.0 |
| **Zambia** | 2021-07-28 | 193432.0 | 183957.0 | 3338.0 |
| **Zimbabwe** | 2021-07-28 | 103567.0 | 71383.0 | 3340.0 |

195 rows × 4 columns

In [21]:

```
df_1['Confirmed'].nlargest(10)
```

Out[21]:

```
Country/Region
US                  34672690.0
India               31484605.0
Brazil              19797086.0
France               6116853.0
Russia               6116249.0
United Kingdom       5797445.0
Turkey               5660469.0
Argentina            4891810.0
Colombia             4757139.0
Spain                4395602.0
Name: Confirmed, dtype: float64
```

In [22]:

```
df_1['Deaths'].nlargest(10)
```

Out[22]:

```
Country/Region
US                  611801.0
Brazil              553179.0
India               422022.0
Mexico              239616.0
Peru                196138.0
Russia              153620.0
United Kingdom      129718.0
Italy               128010.0
Colombia            119801.0
France              111923.0
Name: Deaths, dtype: float64
```

# 5

***Plot a graph of the number of confirmed cases over time for each country. Which countries present exponential growth in the number of cases and which countries are already leaving exponential growth?***

In [ ]:

```
# dividing the column into 16 equal arrays for subplotting purposes
```

In [91]:

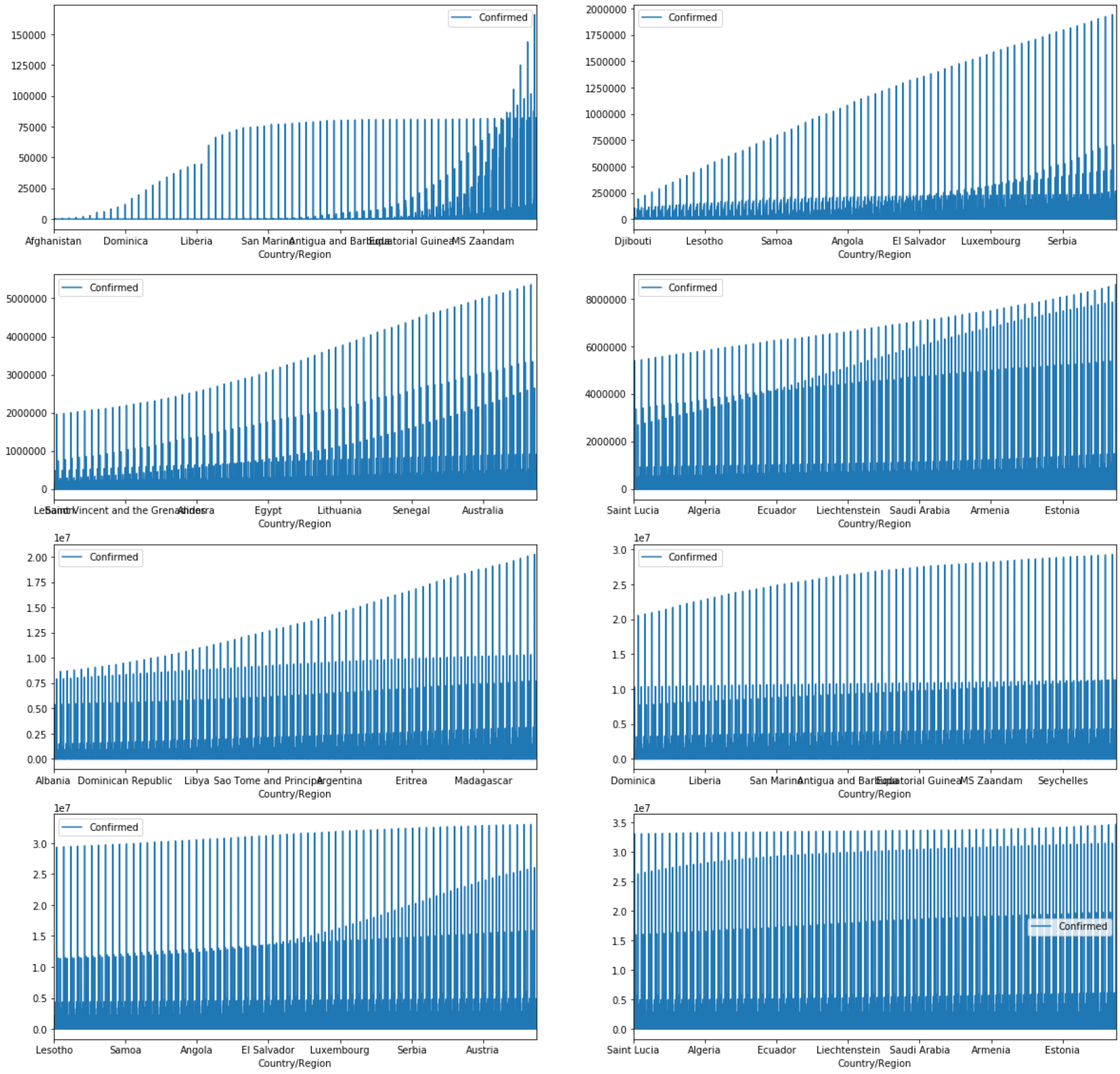
```
df_new.reset_index(level=0, inplace=True)
```

In [93]:

```
import numpy as np
df1, df2, df3, df4, df5, df6, df7, df8 = np.array_split(df_new[['Date','Confirmed']], 8
```

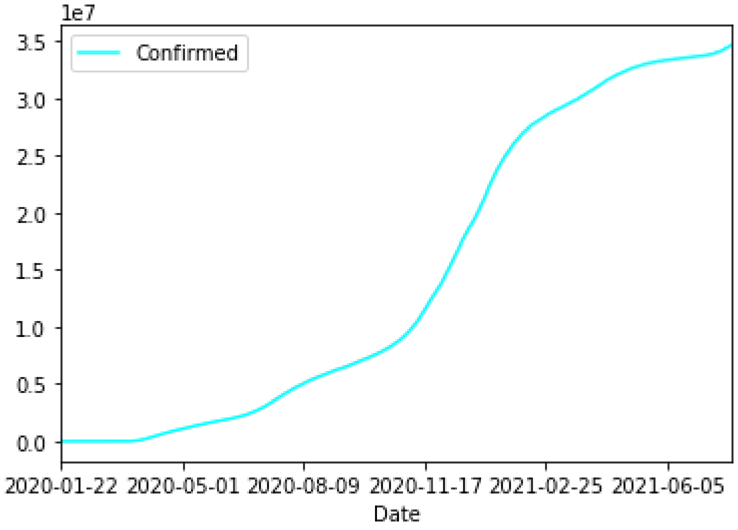
In [94]:

```
# plotting the graphs together by using subplot method
df_list = [df1 ,df2, df3, df4, df5, df6, df7, df8]
fig, axes = plt.subplots(4, 2)

#counter
count=0
for i in range(4):
    for j in range(2):
        df_list[count].plot(ax=axes[i,j], figsize=(20,20))
        count+=1
```



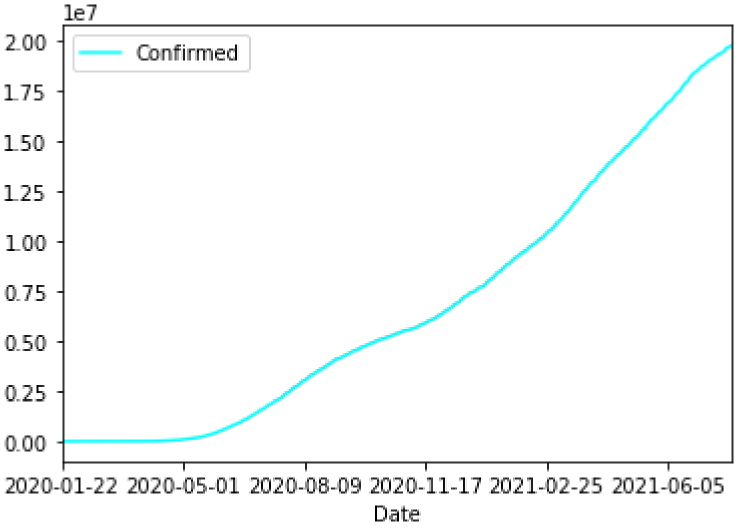
In [106]:

```
# to make plots look comprehensible, i tried plotting them one at a time for one countr

import matplotlib.pyplot as plt
df_US=df[df['Country/Region'] == 'US']
df_US.plot( 'Date' ,'Confirmed', color='cyan' )
plt.show()
```



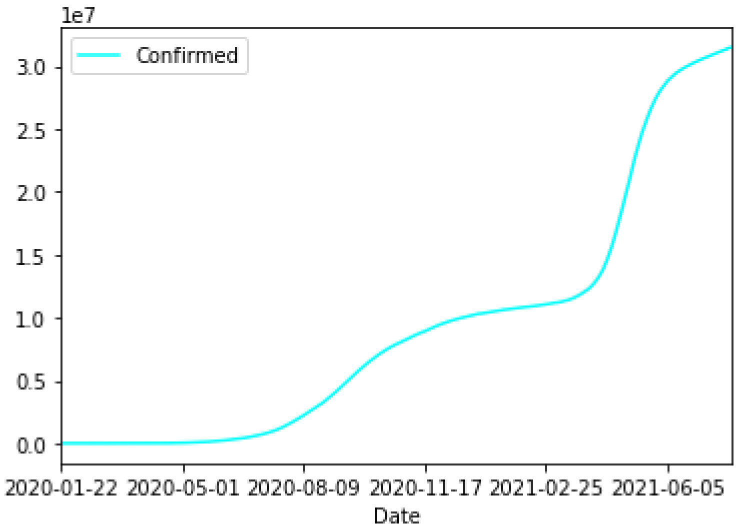
In [108]:

```
df_Brazil=df[df['Country/Region'] == 'Brazil']
df_Brazil.plot( 'Date' ,'Confirmed', color='cyan' )
plt.show()
```



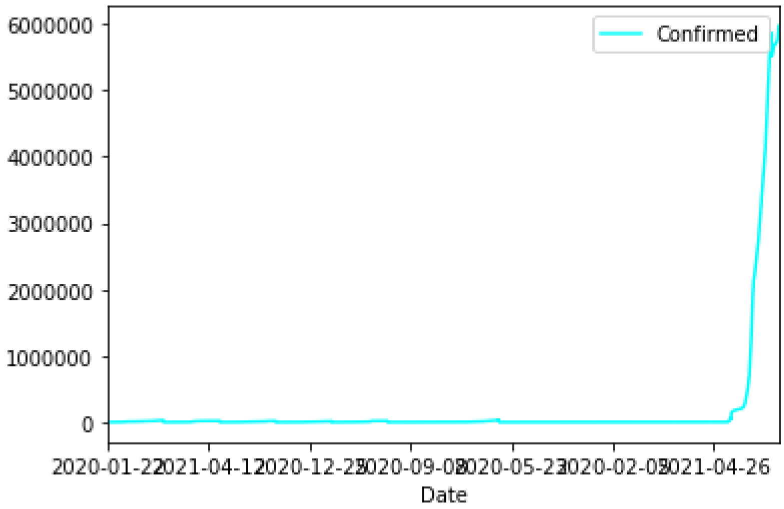
In [109]:

```
df_India=df[df['Country/Region'] == 'India']
df_India.plot( 'Date' ,'Confirmed', color='cyan' )
plt.show()
```



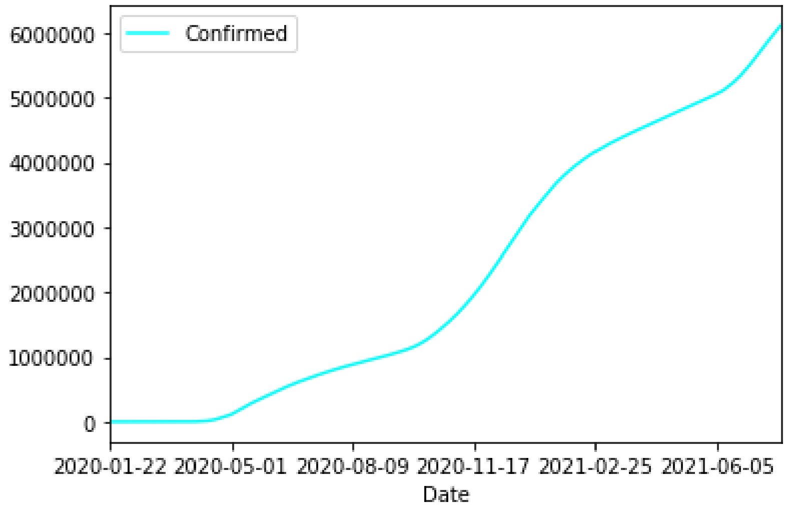
In [110]:

```
df_France=df[df['Country/Region'] == 'France']
df_France.plot( 'Date' ,'Confirmed', color='cyan' )
plt.show()
```



In [111]:

```
df_Russia=df[df['Country/Region'] == 'Russia']
df_Russia.plot( 'Date' ,'Confirmed', color='cyan' )
plt.show()
```



Through these plots we found out that, while countries like US, Brazil, India, Russia had a similar increase in the covid confirmed cases throughout the duration, there were countries like France which encountered a sudden upheaval in the cases around April,2021.

# 6

***Create a bar plot that shows the number of deaths per 100 confirmed cases (observed case-fatality ratio) for the 20 most affected countries. See Figure 1 for an example.***

In [39]:

```
df_6=df_1
df_6.fillna(0)
df_6['Death_percent_confirmed']=df_6['Deaths']*100/df_6['Confirmed']
del df_6['Date']
```

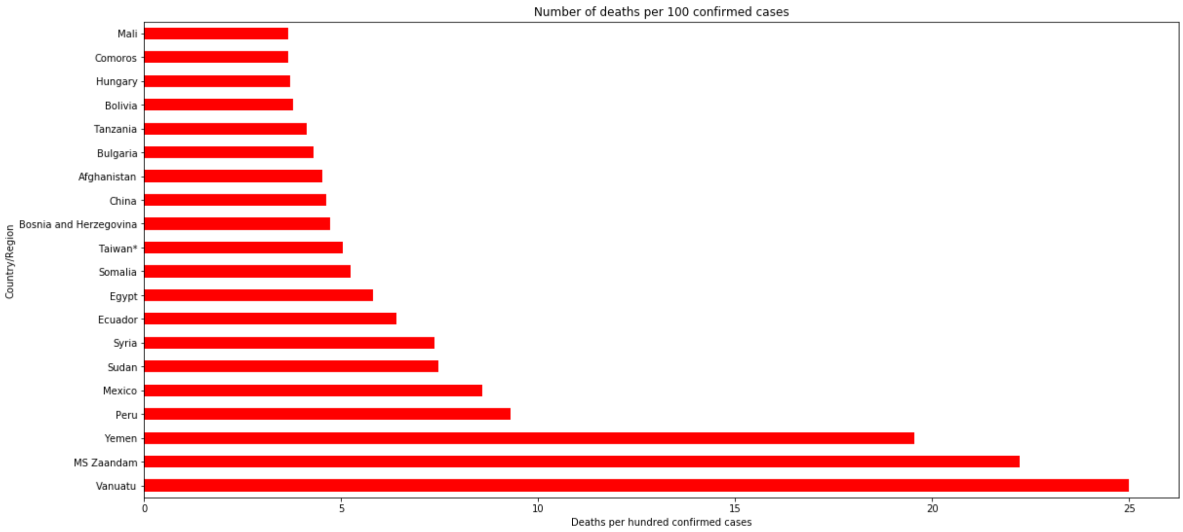
...

In [46]:

```
# plotting a horizontal bar graph
import matplotlib.pyplot as plt

plt.title('Number of deaths per 100 confirmed cases')
plt.xlabel('Deaths per hundred confirmed cases')
plt.ylabel('Country/Region')
df_6['Death_percent_confirmed'].nlargest(20).plot.barh(figsize=(19,9), color='red')
```

Out[46]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x14764aa1248>
```



# 7

***Compute the ratio between the total number of confirmed cases and the population size of each country. The file worldpopulation.json contains data on the population size of each country.***

### Data Cleaning

In [49]:

```
population=pd.read_json('worldpopulation.json')
df_6.reset_index(level=0, inplace=True)
df_7=df_6
```

In [50]:

```
# Finding out the non-matching countries in the two datasets
x=pd.DataFrame()
x=population['country'].sort_values()
x=x.reset_index(drop=True)
data=[df_7['Country/Region'], x]
names = ["country_1", "country_2"]
df_check= pd.concat(data, axis=1, keys=names)
print(list(set(df_check.country_1)-set(df_check.country_2)))
```

```
['West Bank and Gaza', 'Taiwan*', 'US', 'Burma', 'Summer Olympics 2020', 'Sa
int Vincent and the Grenadines', 'Vietnam', 'Kosovo', 'United Kingdom', "Cot
e d'Ivoire", 'Congo (Brazzaville)', 'Eswatini', 'Korea, South', 'North Maced
onia', 'Diamond Princess', 'Congo (Kinshasa)', 'MS Zaandam', 'Czechia']
```

In [52]:

```
# preprocessing the data
# changing the names of countries so that they match
update_df=pd.DataFrame()
update_df['Country/Region']=df_7['Country/Region'].replace("Czechia", "Czech Republic",
update_df['Country/Region']=df_7['Country/Region'].replace("North Macedonia", "North Ko
update_df['Country/Region']=df_7['Country/Region'].replace("Taiwan*", "TFYR Macedonia",
update_df['Country/Region']=df_7['Country/Region'].replace("US", "U.S.",inplace=True)
update_df['Country/Region']=df_7['Country/Region'].replace("United Kingdom", "U.K.",inp
update_df['Country/Region']=df_7['Country/Region'].replace("Vietnam", "Viet Nam",inplac

# removing the rows with countries that are only present in one dataset
update_df = df_7.drop(df_7.index[[40, 42, 48, 58, 92, 93, 106, 148, 167]])
update_pop= population.drop(population.index[[54, 45, 15, 24, 118, 25, 150, 178, 180, 1

update_df = update_df.reset_index(drop=True)
update_pop = update_pop.reset_index(drop=True)
update_df = update_df.set_index('Country/Region')
update_df = update_df.reindex(index=population['country'])
update_df = update_df.reset_index()
update_df= update_df.dropna()
update_df = update_df.reset_index(drop=True)
update_pop= update_pop.dropna()
update_pop = update_pop.reset_index(drop=True)

# creating a new dataframe to check whether all the country names are same or not
data2=[update_df['country'], update_pop['country']]
names2= ["country_1", "country_2"]
df_check2= pd.concat(data2, axis=1, keys=names2)
```

In [55]:

```
# displaying the dataframe containing both the columns of countries to compare them
with pd.option_context('display.max_rows', None,
                       'display.max_columns', None,
                       'display.precision', 3,
                       ):
    print(df_check2)
```

```
                    country_1                 country_2
0                       China                     China
1                       India                     India
2                        U.S.                      U.S.
3                   Indonesia                 Indonesia
4                      Brazil                    Brazil
5                    Pakistan                  Pakistan
6                     Nigeria                   Nigeria
7                  Bangladesh                Bangladesh
8                      Russia                    Russia
9                      Mexico                    Mexico
10                      Japan                     Japan
11                   Ethiopia                  Ethiopia
12                Philippines               Philippines
13                   Viet Nam                  Viet Nam
14                      Egypt                     Egypt
15                       Iran                      Iran
16                    Germany                   Germany
17                     Turkey                    Turkey
18                   Thailand                  Thailand
19                       U.K.                      U.K.
20                     France                    France
21                      Italy                     Italy
22                   Tanzania                  Tanzania
23               South Africa               South Korea
24                   Colombia                  Colombia
25                      Kenya                     Kenya
26                      Spain                     Spain
27                    Ukraine                   Ukraine
28                  Argentina                 Argentina
29                      Sudan                     Sudan
30                     Uganda                    Uganda
31                    Algeria                   Algeria
32                       Iraq                      Iraq
33                     Poland                    Poland
34                     Canada                    Canada
35                    Morocco                   Morocco
36                Afghanistan               Afghanistan
37               Saudi Arabia              Saudi Arabia
38                       Peru                      Peru
39                  Venezuela                 Venezuela
40                   Malaysia                  Malaysia
41                 Uzbekistan                Uzbekistan
42                 Mozambique                     Nepal
43                      Nepal                     Ghana
44                      Ghana                     Yemen
45                      Yemen                    Angola
46                     Angola                Madagascar
47                North Korea               North Korea
48                  Australia                 Australia
49                   Cameroon                  Cameroon
```

```
50                   Niger                  Niger
51               Sri Lanka              Sri Lanka
52                 Romania                Romania
53            Burkina Faso           Burkina Faso
54                   Syria                  Syria
55                    Mali                   Mali
56                   Chile                  Chile
57                  Malawi                 Malawi
58              Kazakhstan             Kazakhstan
59                  Zambia                 Zambia
60             Netherlands            Netherlands
61               Guatemala              Guatemala
62                 Ecuador                Ecuador
63                Zimbabwe               Zimbabwe
64                Cambodia               Cambodia
65                 Senegal                Senegal
66                    Chad                   Chad
67                  Guinea                 Guinea
68             South Sudan            South Sudan
69                  Rwanda                 Rwanda
70                 Burundi                Burundi
71                 Tunisia                Tunisia
72                   Benin                  Benin
73                 Belgium                Belgium
74                 Somalia                Somalia
75                    Cuba                   Cuba
76                 Bolivia                Bolivia
77                   Haiti                  Haiti
78                  Greece                 Greece
79      Dominican Republic     Dominican Republic
80          Czech Republic         Czech Republic
81                Portugal               Portugal
82              Azerbaijan             Azerbaijan
83                  Sweden                 Sweden
84                 Hungary                Hungary
85                 Belarus                Belarus
86    United Arab Emirates   United Arab Emirates
87              Tajikistan             Tajikistan
88                  Serbia                 Serbia
89                 Austria                Austria
90             Switzerland            Switzerland
91                  Israel                 Israel
92                Honduras               Honduras
93        Papua New Guinea       Papua New Guinea
94                  Jordan                 Jordan
95                    Togo                   Togo
96                Bulgaria               Bulgaria
97                    Laos                   Laos
98                Paraguay               Paraguay
99            Sierra Leone           Sierra Leone
100                  Libya                  Libya
101              Nicaragua              Nicaragua
102            El Salvador            El Salvador
103             Kyrgyzstan                Lebanon
104                Lebanon              Singapore
105              Singapore                Denmark
106                Denmark                Finland
107                Finland           Turkmenistan
108                Eritrea                Eritrea
109               Slovakia               Slovakia
110                 Norway                 Norway
```

```
111  Central African Republic  Central African Republic
112                Costa Rica                Costa Rica
113                   Ireland                     Congo
114                      Oman                   Ireland
115                   Liberia                      Oman
116               New Zealand                   Liberia
117                Mauritania               New Zealand
118                   Croatia                Mauritania
119                    Kuwait                   Croatia
120                   Moldova                    Kuwait
121                    Panama                   Moldova
122                   Georgia                    Panama
123    Bosnia and Herzegovina                   Georgia
124                   Uruguay    Bosnia and Herzegovina
125                  Mongolia                   Uruguay
126                   Armenia                  Mongolia
127                   Albania                   Armenia
128                 Lithuania                   Albania
129                   Jamaica                 Lithuania
130                   Namibia                   Jamaica
131                  Botswana                   Namibia
132                     Qatar                  Botswana
133                   Lesotho                     Qatar
134                    Gambia                   Lesotho
135            TFYR Macedonia                    Gambia
136                  Slovenia            TFYR Macedonia
137                    Latvia                  Slovenia
138             Guinea-Bissau                    Latvia
139                     Gabon             Guinea-Bissau
140                   Bahrain                   Bahrain
141       Trinidad and Tobago       Trinidad and Tobago
142                   Estonia                   Estonia
143                 Mauritius                 Mauritius
144               Timor-Leste               Timor-Leste
145                    Cyprus                    Cyprus
146                  Djibouti                  Djibouti
147                      Fiji                      Fiji
148         Equatorial Guinea         Equatorial Guinea
149                   Comoros                   Comoros
150                    Bhutan                    Bhutan
151                    Guyana                    Guyana
152                Montenegro                Montenegro
153           Solomon Islands           Solomon Islands
154                Luxembourg                Luxembourg
155                  Suriname                  Suriname
156                Cabo Verde                Cabo Verde
157                    Brunei                    Brunei
158                     Malta                     Malta
159                   Bahamas                   Bahamas
160                  Maldives                  Maldives
161                    Belize                    Belize
162                   Iceland                   Iceland
163                  Barbados                  Barbados
164                   Vanuatu                   Vanuatu
165     Sao Tome and Principe     Sao Tome and Principe
166                     Samoa                     Samoa
167               Saint Lucia               Saint Lucia
168                  Kiribati                  Kiribati
169                   Grenada                   Grenada
170                Micronesia                Micronesia
171                Seychelles                Seychelles
```

```
172       Antigua and Barbuda       Antigua and Barbuda
173                  Dominica                  Dominica
174                   Andorra                   Andorra
175     Saint Kitts and Nevis     Saint Kitts and Nevis
176          Marshall Islands          Marshall Islands
177             Liechtenstein             Liechtenstein
178                    Monaco                    Monaco
179                San Marino                San Marino
180                  Holy See                     Palau
181                       NaN                  Holy See
```

### Finding the Ratio

In [59]:

```
df_ratio=pd.DataFrame()
df_ratio['Cases_per_capita']=update_df['Confirmed']/update_pop['population']
df_ratio['Country']=update_pop['country']
```

In [60]:

```
df_ratio
```

Out[60]:

| | Cases_per_capita | Country |
|---|---|---|
| **0** | 0.000076 | China |
| **1** | 0.023452 | India |
| **2** | 0.106204 | U.S. |
| **3** | 0.012477 | Indonesia |
| **4** | 0.093717 | Brazil |
| **...** | ... | ... |
| **177** | 0.081085 | Liechtenstein |
| **178** | 0.074533 | Monaco |
| **179** | 0.159793 | San Marino |
| **180** | 0.001243 | Palau |
| **181** | NaN | Holy See |

182 rows × 2 columns

### *What are the 10 countries with the highest number of confirmed COVID-19 cases per capita?*

In [57]:

```
df_per_capita=pd.DataFrame()
df_per_capita['Cases_per_capita']=update_df['Confirmed']*100/update_pop['population']
df_per_capita['Country']=update_pop['country']
```

In [58]:

```
df_per_capita.nlargest(10,'Cases_per_capita')
```

Out[58]:

| | Cases_per_capita | Country |
|---|---|---|
| **174** | 21.222791 | Andorra |
| **160** | 20.510979 | Maldives |
| **140** | 18.947561 | Bahrain |
| **171** | 18.517721 | Seychelles |
| **152** | 16.206946 | Montenegro |
| **179** | 15.979317 | San Marino |
| **80** | 15.850274 | Czech Republic |
| **154** | 12.625171 | Luxembourg |
| **136** | 12.425623 | TFYR Macedonia |
| **83** | 11.075866 | Sweden |

# 8

***In this part we would like to test the hypothesis that the spread of the virus is slowed down by warm weather. Plot a graph of the monthly number of confirmed cases vs. the average monthly temperature for a few selected countries, and analyze the correlation between these two factors.***

### Data Processing

In [64]:

```
import json
from pandas.io.json import json_normalize
with open('climate.json') as file:
    data=json.load(file)
climate=json_normalize(data, 'monthlyAvg', ['country'], errors='ignore')
climate.columns=['high','low',"dryDays","snowDays",'rainfall', 'Country/Region']
cl=pd.DataFrame(climate)
```

In [65]:

```
# taking the average of all the temperatures of each country
average_high_temp=[]
for j in cl['Country/Region']:
    cl_new= cl[cl['Country/Region'] == j]
    cl_new=cl.groupby(['Country/Region'])['high'].mean().sort_values().reset_index(name
    cl_new2=cl.groupby(['Country/Region'])['low'].mean().sort_values().reset_index(name
    cl_merge=pd.merge(cl_new, cl_new2, on= 'Country/Region')
```

In [66]:

```
# finding the confirmed cases per month and averaging all the confirmed cases per month
monthly_confirmed=[]
for i in df_1['Country/Region']:
    df_us = df[df['Country/Region'] == i]
    df_us['Date'] = pd.to_datetime(df['Date'])
    df_us=df_us.groupby(df_us['Date'].dt.strftime('%B'))['Confirmed'].sum().sort_values
    x=df_us.mean()
    monthly_confirmed.append(x)
```

...

In [67]:

```
# merging all the processed data into a single DataFrame
df_climate=pd.DataFrame()
df_climate['Country/Region']=df_1['Country/Region']
df_climate['average_monthly_confirmed']=monthly_confirmed

final_data=pd.merge(df_climate, cl_merge, on='Country/Region')
final_data['avg_temp']=(final_data['high']+final_data['low'])/2
```

In [68]:

```
final_data
```

Out[68]:

| | Country/Region | average_monthly_confirmed | high | low | avg_temp |
|---|---|---|---|---|---|
| **0** | Argentina | 6.408577e+07 | 22.166667 | 14.500000 | 18.333333 |
| **1** | Australia | 9.423037e+05 | 22.041667 | 11.104167 | 16.572917 |
| **2** | Austria | 1.141650e+07 | 14.916667 | 5.583333 | 10.250000 |
| **3** | Belgium | 2.035460e+07 | 14.416667 | 7.000000 | 10.708333 |
| **4** | Brazil | 3.127759e+08 | 29.055556 | 18.916667 | 23.986111 |
| **5** | Bulgaria | 6.818600e+06 | 16.416667 | 4.833333 | 10.625000 |
| **6** | Canada | 2.324708e+07 | 12.130952 | 1.797619 | 6.964286 |
| **7** | Chile | 2.708727e+07 | 23.666667 | 7.333333 | 15.500000 |
| **8** | China | 4.143917e+06 | 19.625000 | 10.375000 | 15.000000 |
| **9** | Czech Republic | 0.000000e+00 | 13.833333 | 5.083333 | 9.458333 |
| **10** | Denmark | 5.041026e+06 | 12.583333 | 5.750000 | 9.166667 |
| **11** | France | 1.005386e+08 | 18.041667 | 8.791667 | 13.416667 |
| **12** | Germany | 6.234787e+07 | 13.250000 | 6.000000 | 9.625000 |
| **13** | Greece | 5.864084e+06 | 19.666667 | 13.666667 | 16.666667 |
| **14** | Hungary | 1.228024e+07 | 16.000000 | 6.750000 | 11.375000 |
| **15** | Iceland | 1.776009e+05 | 8.000000 | 2.416667 | 5.208333 |
| **16** | India | 4.198417e+08 | 30.472222 | 19.333333 | 24.902778 |
| **17** | Indonesia | 3.362925e+07 | 31.666667 | 23.000000 | 27.333333 |
| **18** | Ireland | 4.956613e+06 | 12.416667 | 5.916667 | 9.166667 |
| **19** | Israel | 1.699681e+07 | 23.458333 | 14.333333 | 18.895833 |
| **20** | Italy | 7.437087e+07 | 20.250000 | 10.083333 | 15.166667 |
| **21** | Japan | 1.146843e+07 | 20.875000 | 11.583333 | 16.229167 |
| **22** | Malaysia | 8.298775e+06 | 32.833333 | 24.000000 | 28.416667 |
| **23** | Mexico | 5.108411e+07 | 24.083333 | 10.500000 | 17.291667 |
| **24** | Morocco | 1.140396e+07 | 26.250000 | 12.583333 | 19.416667 |
| **25** | Netherlands | 2.829252e+07 | 12.833333 | 7.583333 | 10.208333 |
| **26** | New Zealand | 8.345533e+04 | 16.888889 | 9.888889 | 13.388889 |
| **27** | Norway | 2.094937e+06 | 10.666667 | 3.666667 | 7.166667 |
| **28** | Poland | 4.611836e+07 | 13.000000 | 4.166667 | 8.583333 |
| **29** | Portugal | 1.627157e+07 | 18.166667 | 11.875000 | 15.020833 |
| **30** | Russia | 1.049129e+08 | 9.333333 | 1.041667 | 5.187500 |
| **31** | Singapore | 2.103296e+06 | 31.416667 | 24.666667 | 28.041667 |
| **32** | South Africa | 3.877769e+07 | 23.500000 | 10.583333 | 17.041667 |
| **33** | Spain | 7.354313e+07 | 21.000000 | 10.833333 | 15.916667 |

| | Country/Region | average_monthly_confirmed | high | low | avg_temp |
|---|---|---|---|---|---|
| **34** | Sweden | 1.728602e+07 | 10.583333 | 2.666667 | 6.625000 |
| **35** | Switzerland | 1.317681e+07 | 14.666667 | 5.000000 | 9.833333 |
| **36** | Thailand | 2.023818e+06 | 32.722222 | 22.722222 | 27.722222 |
| **37** | Turkey | 7.868205e+07 | 20.416667 | 9.083333 | 14.750000 |
| **38** | United Arab Emirates | 1.011001e+07 | 33.500000 | 22.750000 | 28.125000 |

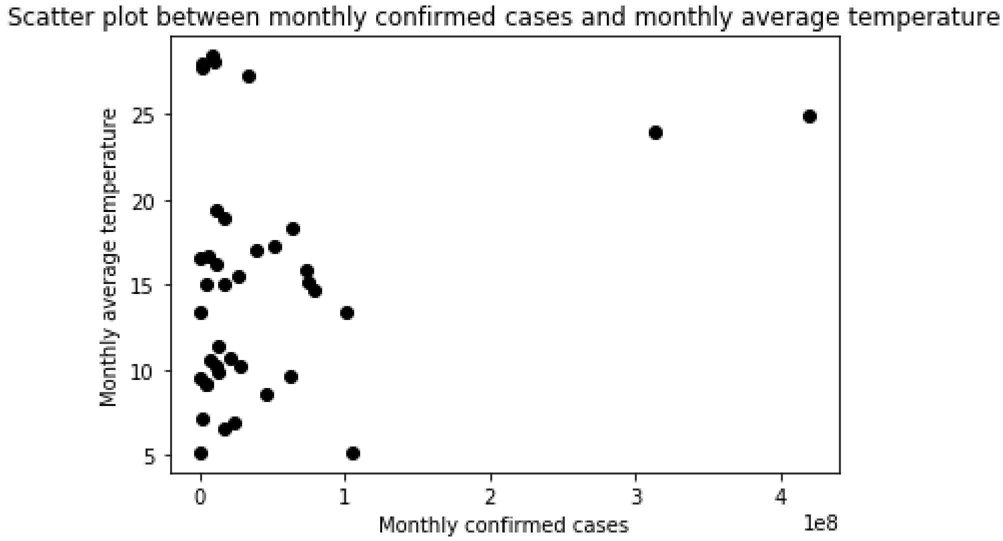
### Visualization

In [83]:

```
import matplotlib.pyplot as plt
plt.title('Scatter plot between monthly confirmed cases and monthly average temperature
plt.xlabel('Monthly confirmed cases')
plt.ylabel('Monthly average temperature')
plt.scatter(final_data['average_monthly_confirmed'],final_data['avg_temp'], color='blac
```

Out[83]:

```
<matplotlib.collections.PathCollection at 0x1475bd9e448>
```

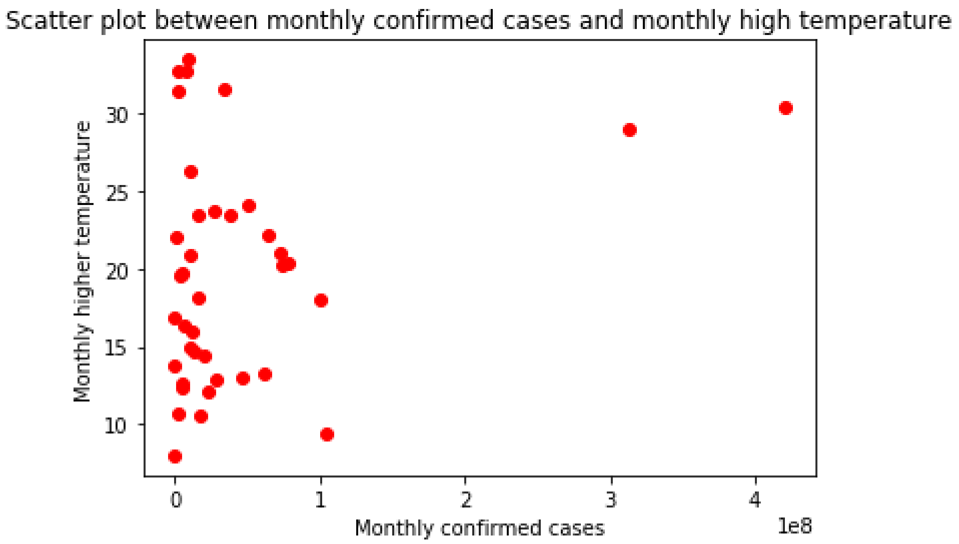


In [85]:

```
plt.title('Scatter plot between monthly confirmed cases and monthly high temperature')
plt.xlabel('Monthly confirmed cases')
plt.ylabel('Monthly higher temperature')
plt.scatter(final_data['average_monthly_confirmed'],final_data['high'], color='red')
```

Out[85]:

```
<matplotlib.collections.PathCollection at 0x1475cbfa288>
```

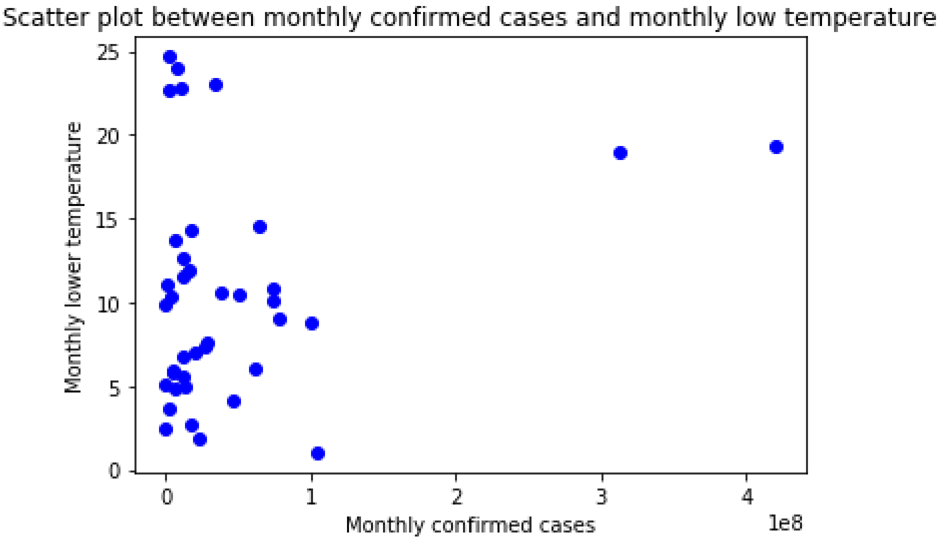


In [86]:

```
plt.title('Scatter plot between monthly confirmed cases and monthly low temperature')
plt.xlabel('Monthly confirmed cases')
plt.ylabel('Monthly lower temperature')
plt.scatter(final_data['average_monthly_confirmed'],final_data['low'], color='blue')
```

Out[86]:

```
<matplotlib.collections.PathCollection at 0x1475bdd05c8>
```



Through this plot we can say that the covid cases does not seems to be decreasing with the increase in temperature.

# 9

***Articulate your own research question related to COVID-19 and try to provide an answer to it using the given data set.***

The effect of population level of a country on the amount of confirmed cases

In [115]:

```
data3=[update_df['Confirmed'], update_pop['population']]
names3= ["Confirmed", "population"]
df_population= pd.concat(data3, axis=1, keys=names3)
```

In [116]:

```
df_population
```

Out[116]:

| | Confirmed | population |
|---|---|---|
| **0** | 104850.0 | 1388232693 |
| **1** | 31484605.0 | 1342512706 |
| **2** | 34672690.0 | 326474013 |
| **3** | 3287727.0 | 263510146 |
| **4** | 19797086.0 | 211243220 |
| **...** | ... | ... |
| **177** | 3083.0 | 38022 |
| **178** | 2833.0 | 38010 |
| **179** | 5130.0 | 32104 |
| **180** | 27.0 | 21726 |
| **181** | NaN | 801 |

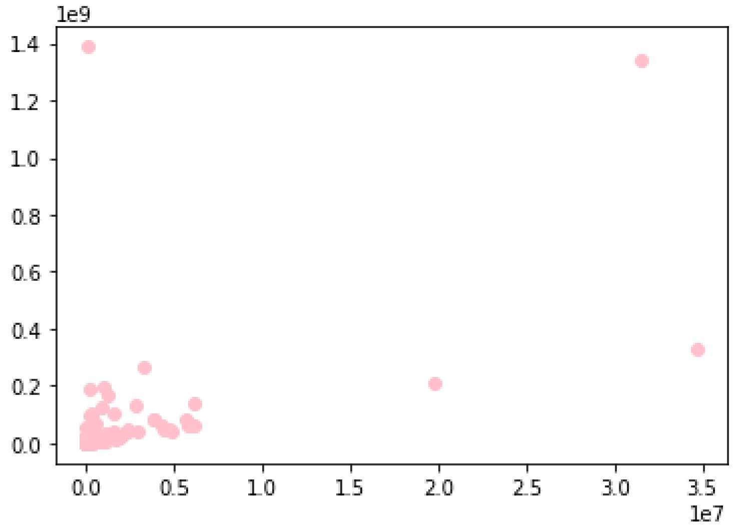
182 rows × 2 columns

In [121]:

```
plt.scatter(df_population['Confirmed'],df_population['population'], color='pink')
```

Out[121]:

```
<matplotlib.collections.PathCollection at 0x1476242aa88>
```



We can see a slight correlation between the population and the confirmed cases. It shows that the number of confirmed cases in a country does depend on the population of the particular country.