In [1]:
```
import requests,datetime,os
import json
import pandas as pd
import numpy as np
from io import BytesIO
import datetime as dt
```

In [2]:
```
#! pip install ...
import folium
%matplotlib inline

import matplotlib
import matplotlib.pyplot as plt
import chart_studio as plotly
import chart_studio.plotly as py
import plotly.express as px
import plotly.graph_objects as go
import random
```

In [3]:
```
print("COVID19 Analysis")
url_1 = 'https://api.covid19api.com/summary'
response_summary = requests.get(url_1)
summary = response_summary.json()
```

```
COVID19 Analysis
```

In [4]:
```
New_Confirmed = summary['Global']['NewConfirmed']
Total_Confirmed = summary['Global']['TotalConfirmed']
New_Deaths = summary['Global']['NewDeaths']
Total_Deaths = summary['Global']['TotalDeaths']
New_Recovered = summary['Global']['NewRecovered']
Total_Recovered = summary['Global']['TotalRecovered']
print("New COVID19 Reports")
Overall_df = pd.DataFrame({'NewConfirmed': [New_Confirmed],'TotalConfirmed': [Total_
Overall_df
```

```
New COVID19 Reports
```

Out[4]:

	NewConfirmed	TotalConfirmed	NewDeaths	TotalDeaths	NewRecovered	TotalRecovered
0	1773740	319532570	4664	5516838	0	0

In [5]:
```
def GetCountries():
    response = requests.get('https://api.covid19api.com/summary')
    data_dict = response.json()
    countries = data_dict['Countries']
    return countries
countries = GetCountries()
country_df = pd.DataFrame(countries)
print(country_df.head(5))
```

```
                                     ID      Country CountryCode         Slug  \
0  523c5020-c674-4171-994e-3290905b19b0  Afghanistan          AF  afghanistan
1  1ccaa233-392f-4df4-946e-fff12cac938a      Albania          AL      albania
2  e3abfa50-1037-488f-8fdc-af14dcc88104      Algeria          DZ      algeria
3  cff5f4d1-ab47-4a98-8d26-c9c6df9457a2      Andorra          AD      andorra
4  124a2495-6b4c-49fe-abba-60dd8aca7dcc       Angola          AO       angola
```

```
   NewConfirmed  TotalConfirmed  NewDeaths  TotalDeaths  NewRecovered  \
0             0          158602          0         7376             0
1             0          226598          0         3255             0
2             0          224383          0         6383             0
3             0           28899          0          141             0
4             0           92581          0         1847             0

   TotalRecovered                      Date Premium
0               0  2022-01-14T16:43:30.993Z      {}
1               0  2022-01-14T16:43:30.993Z      {}
2               0  2022-01-14T16:43:30.993Z      {}
3               0  2022-01-14T16:43:30.993Z      {}
4               0  2022-01-14T16:43:30.993Z      {}
```

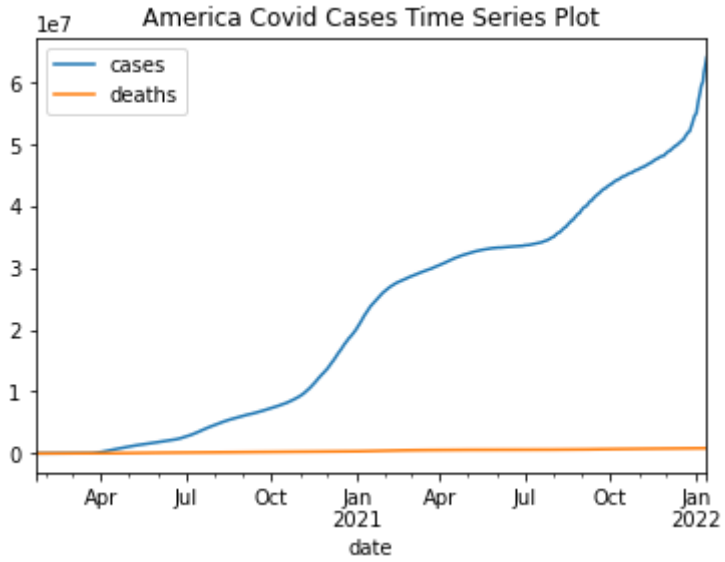
In [6]:

```
# COVID-19 Datasets
github_url1 = 'https://github.com/nytimes/covid-19-data/blob/master/us.csv?raw=true'
github_url2 = 'https://github.com/nytimes/covid-19-data/blob/master/us-states.csv?ra
github_url3 = 'https://github.com/nytimes/covid-19-data/blob/master/us-counties.csv?
github_url4 = 'https://github.com/nytimes/covid-19-data/blob/master/us-counties-rece
df1 = pd.read_csv(github_url1)
df2 = pd.read_csv(github_url2)
print(df1.head(5))
```

```
         date  cases  deaths
0  2020-01-21      1       0
1  2020-01-22      1       0
2  2020-01-23      1       0
3  2020-01-24      2       0
4  2020-01-25      3       0
```

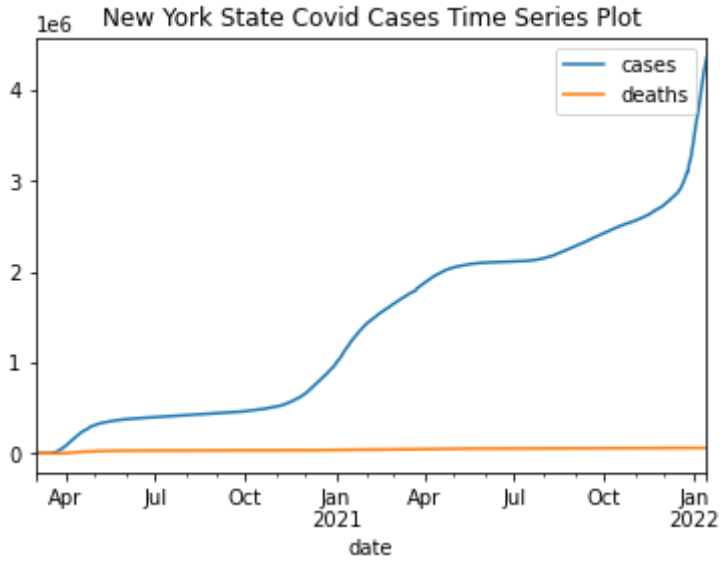
In [7]:

```
from pandas import read_csv
from matplotlib import pyplot
# America Covid cases time series plot
series = read_csv('https://github.com/nytimes/covid-19-data/blob/master/us.csv?raw=t
series.plot(title="America Covid Cases Time Series Plot")
pyplot.show()
```



In [8]:

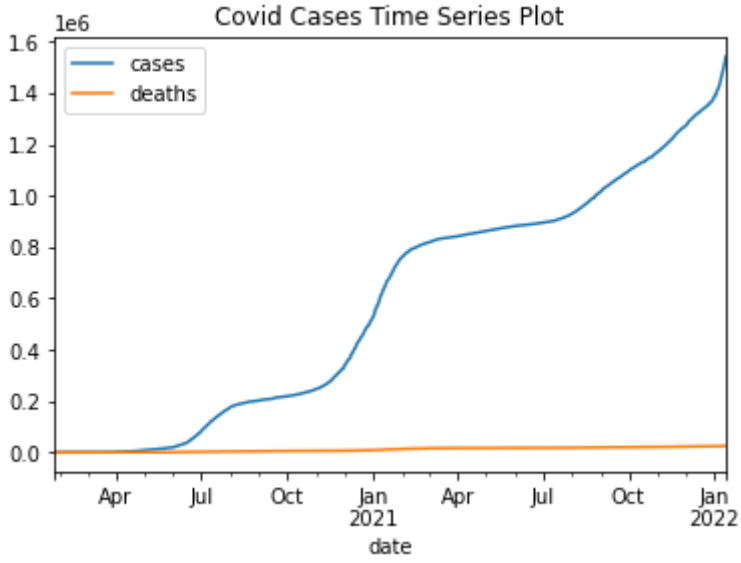
```
#
df2 = df2.drop('fips', 1)
nydata = df2.query('state=="New York"')
nydata.to_csv(r'nydata.csv',index = False)
series = read_csv('nydata.csv', header=0, index_col=0, parse_dates=True, squeeze=Tru
series.plot(title="New York State Covid Cases Time Series Plot")
pyplot.show()
```



In [10]:

```
statedata = df2.query(input('Please enter state=="state you want to query"'))
statedata.to_csv(r'statedata.csv',index = False)
series = read_csv('statedata.csv', header=0, index_col=0, parse_dates=True, squeeze=
series.plot(title="Covid Cases Time Series Plot")
pyplot.show()
```

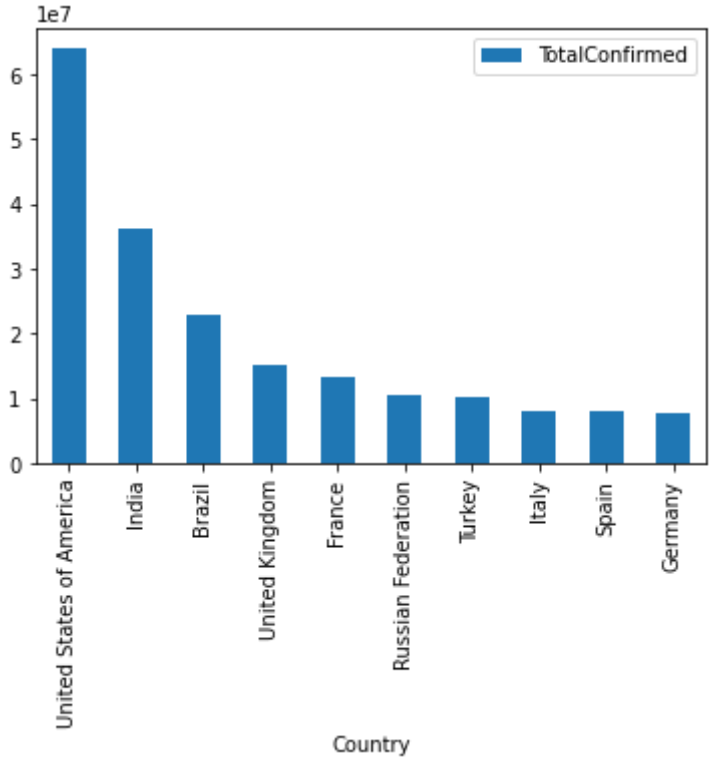
```
Please enter state=="state you want to query"state=="Arizona"
```



In [12]:

```
case_type = {'TotalConfirmed', 'NewConfirmed', 'NewDeaths', 'TotalDeaths', 'NewRecov
print(case_type)
you_want = input('Enter or copy one of the case types listed here: ')
try:
    top = country_df.sort_values(input('Enter the same case type again to view bar c
    top.head(10).plot.bar(y = {you_want} ,x = 'Country')
except KeyError:
    print('You entered wrong case type!')
```

```
{'TotalRecovered', 'NewRecovered', 'TotalConfirmed', 'TotalDeaths', 'NewDeaths', 'Ne
wConfirmed'}
Enter or copy one of the case types listed here: TotalConfirmed
Enter the same case type again to view bar chart of top10 countries: TotalConfirmed
```



In [13]:

```
print("COVID19 in America")
url = 'https://api.covid19api.com/live/country/united%20states%20of%20america/status
def US():
    response = requests.get('https://api.covid19api.com/live/country/united%20states
    data_dict = response.json()
    return data_dict
data_dict1 = US()
us_df = pd.DataFrame(data_dict1)
```

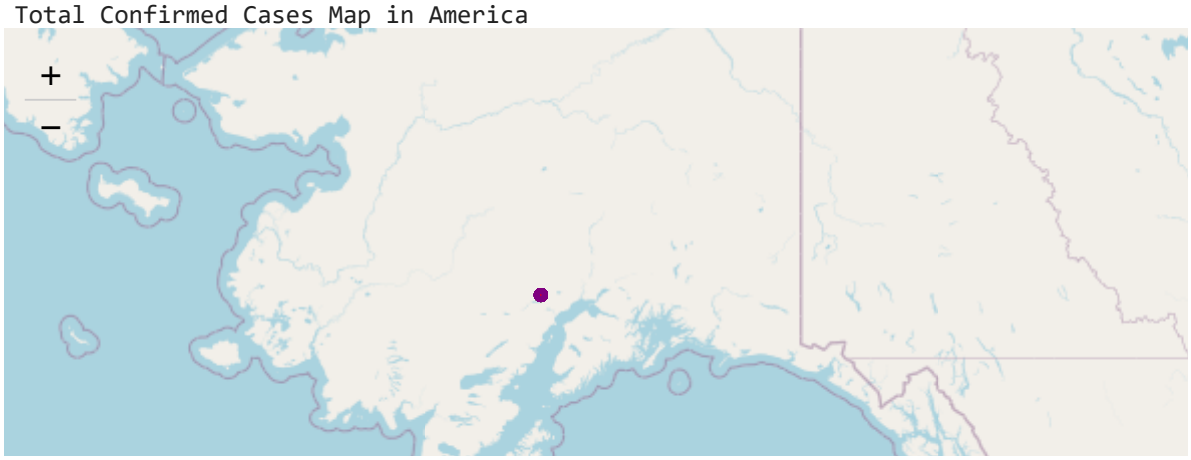
```
COVID19 in America
```

In [14]:

```
print("Total Confirmed Cases Map in America")
CENTER_US = (39.8333333,-98.585522)
london = (51.5074, -0.1278)
map = folium.Map(location=CENTER_US, zoom_start=4)
for row in us_df.to_records():
    pos = (row['Lat'],row['Lon'])
    marker = folium.CircleMarker(location=pos, radius=row['Confirmed']/80000, color=
                    popup="%s,%s" % (row['Province'],row['Confirmed'])
                          )
    map.add_child(marker)
map
#Clicking on the dots will show the exact number.
```

```
Total Confirmed Cases Map in America
```

Out[14]:

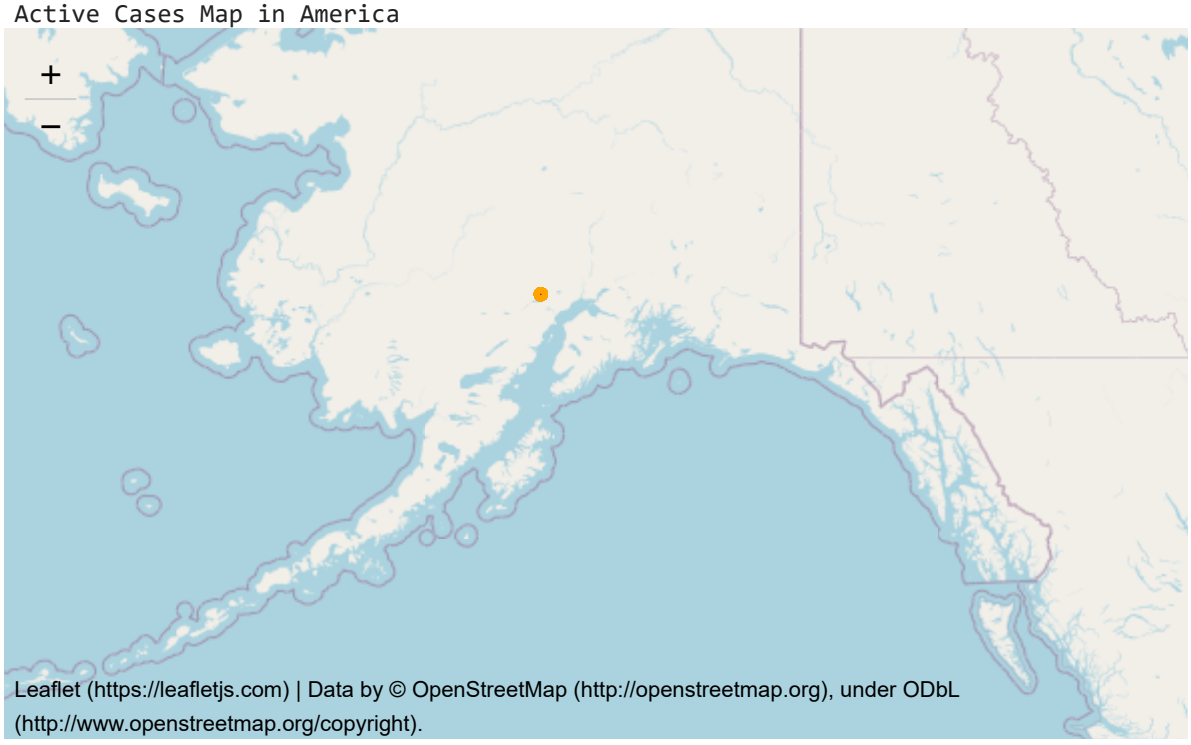


In [15]:

```
print("Active Cases Map in America")
CENTER_US = (39.8333333,-98.585522)
london = (51.5074, -0.1278)
map = folium.Map(location=CENTER_US, zoom_start=4)
for row in us_df.to_records():
    pos = (row['Lat'],row['Lon'])
    marker = folium.CircleMarker(location=pos, radius=row['Active']/80000, color='or
                    popup="%s,%s" % (row['Province'],row['Active'])
                           )
    map.add_child(marker)
map
#Clicking on the dots will show the exact number.
```

Active Cases Map in America

Out[15]:



In [16]:

```
print("Total Deaths Map in America")
CENTER_US = (39.8333333,-98.585522)
london = (51.5074, -0.1278)
map = folium.Map(location=CENTER_US, zoom_start=4)
for row in us_df.to_records():
    pos = (row['Lat'],row['Lon'])
    marker = folium.CircleMarker(location=pos, radius=row['Deaths']/5000, color='bla
                    popup="%s,%s" % (row['Province'],row['Deaths'])
                           )
    map.add_child(marker)
map
#Clicking on the dots will show the exact number.
```

Total Deaths Map in America

Out[16]: Make this Notebook Trusted to load map: File -> Trust Notebook

+

In [ ]:

In [ ]:

In [ ]: