Observing Newest Covid19 Confirmed Statistics

In [1]:
```
from datetime import date
from datetime import timedelta
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
```

In [2]:
```
class CovidNineteen:
    def get_latest_daily_report(self):
        """
        Get latest daily report(world) from:
        https://github.com/CSSEGISandData/COVID-19/tree/master/csse_covid_19_data/csse_covid_19_daily_reports
        """
        data_date = date.today()
        data_date_delta = timedelta(days=1)
        daily_report_url_no_date = "https://raw.githubusercontent.com/CSSEGISandData/COVID-19/master/csse_covid_19_data/csse_covid_19_daily_reports/{}.csv"
        while True:
            data_date_str = date.strftime(data_date, '%m-%d-%Y')
            daily_report_url = daily_report_url_no_date.format(data_date_str)
            try:
                print("Trying to get {} daily report.".format(data_date_str))
                daily_report = pd.read_csv(daily_report_url)
                print("The file exists，got {} daily report.".format(data_date_str))
                break
            except:
                print("{} hasn't uploaded yet.".format(data_date_str))
                data_date -= data_date_delta # data_date = data_date - data_date_delta
        return daily_report

    def get_time_series(self):
        """
        Get time series data from:
        https://github.com/CSSEGISandData/COVID-19/tree/master/csse_covid_19_data/csse_covid_19_time_series
        """
        time_series = pd.read_csv("https://raw.githubusercontent.com/CSSEGISandData/COVID-19/master/csse_covid_19_data/csse_covid_19_time_series/time_series_covid19_
        return time_series
```

In [3]:
```
covid_19 = CovidNineteen()
daily_report = covid_19.get_latest_daily_report()
time_series = covid_19.get_time_series()
```

```
Trying to get 02-20-2021 daily report.
02-20-2021 hasn't uploaded yet.
Trying to get 02-19-2021 daily report.
The file exists，got 02-19-2021 daily report.
```

In [4]:
```
daily_report.head()
```

Out[4]:

	FIPS	Admin2	Province_State	Country_Region	Last_Update	Lat	Long_	Confirmed	Deaths	Recovered	Active	Combined_Key	Incident_Rate	Case_Fatality_Ratio
0	NaN	NaN	NaN	Afghanistan	2021-02-20 05:22:51	33.93911	67.709953	55575	2430	48803	4342.0	Afghanistan	142.762313	4.372470
1	NaN	NaN	NaN	Albania	2021-02-20 05:22:51	41.15330	20.168300	97909	1636	61605	34668.0	Albania	3402.216971	1.670939
2	NaN	NaN	NaN	Algeria	2021-02-20 05:22:51	28.03390	1.659600	111600	2954	76797	31849.0	Algeria	254.497937	2.646953
3	NaN	NaN	NaN	Andorra	2021-02-20 05:22:51	42.50630	1.521800	10645	107	10146	392.0	Andorra	13777.260079	1.005167
4	NaN	NaN	NaN	Angola	2021-02-20 05:22:51	-11.20270	17.873900	20478	498	18991	989.0	Angola	62.307044	2.431878

In [5]:
```
time_series.head()
```

Out[5]:

	Province/State	Country/Region	Lat	Long	1/22/20	1/23/20	1/24/20	1/25/20	1/26/20	1/27/20	...	2/10/21	2/11/21	2/12/21	2/13/21	2/14/21	2/15/21	2/16/21	2/17/21	2/18/21	2/19/2
0	NaN	Afghanistan	33.93911	67.709953	0	0	0	0	0	0	...	55402	55420	55445	55473	55492	55514	55518	55540	55557	5557
1	NaN	Albania	41.15330	20.168300	0	0	0	0	0	0	...	88671	89776	90835	91987	93075	93850	94651	95726	96838	9790
2	NaN	Algeria	28.03390	1.659600	0	0	0	0	0	0	...	109782	110049	110303	110513	110711	110894	111069	111247	111418	1116C
3	NaN	Andorra	42.50630	1.521800	0	0	0	0	0	0	...	10352	10391	10427	10463	10503	10538	10555	10583	10610	1064
4	NaN	Angola	-11.20270	17.873900	0	0	0	0	0	0	...	20210	20261	20294	20329	20366	20381	20389	20400	20452	2047

5 rows × 399 columns

In [6]:
```
idVars  =['Province/State', 'Country/Region', 'Lat', 'Long']
time_series_long = pd.melt(time_series, id_vars=idVars, var_name='Date', value_name='Confirmed')
time_series_long
```

Out[6]:

	Province/State	Country/Region	Lat	Long	Date	Confirmed
0	NaN	Afghanistan	33.939110	67.709953	1/22/20	0
1	NaN	Albania	41.153300	20.168300	1/22/20	0
2	NaN	Algeria	28.033900	1.659600	1/22/20	0
3	NaN	Andorra	42.506300	1.521800	1/22/20	0
4	NaN	Angola	-11.202700	17.873900	1/22/20	0
...	...	...	...	...	...	...
107830	NaN	Vietnam	14.058324	108.277199	2/19/21	2362
107831	NaN	West Bank and Gaza	31.952200	35.233200	2/19/21	171717
107832	NaN	Yemen	15.552727	48.516388	2/19/21	2157
107833	NaN	Zambia	-13.133897	27.849332	2/19/21	73203
107834	NaN	Zimbabwe	-19.015438	29.154857	2/19/21	35710

107835 rows × 6 columns

In [7]:
```
time_series_long['Date'] = pd.to_datetime(time_series_long['Date'])
time_series_long
```

Out[7]:

	Province/State	Country/Region	Lat	Long	Date	Confirmed
0	NaN	Afghanistan	33.939110	67.709953	2020-01-22	0
1	NaN	Albania	41.153300	20.168300	2020-01-22	0
2	NaN	Algeria	28.033900	1.659600	2020-01-22	0
3	NaN	Andorra	42.506300	1.521800	2020-01-22	0
4	NaN	Angola	-11.202700	17.873900	2020-01-22	0
...	...	...	...	...	...	...
107830	NaN	Vietnam	14.058324	108.277199	2021-02-19	2362
107831	NaN	West Bank and Gaza	31.952200	35.233200	2021-02-19	171717
107832	NaN	Yemen	15.552727	48.516388	2021-02-19	2157
107833	NaN	Zambia	-13.133897	27.849332	2021-02-19	73203
107834	NaN	Zimbabwe	-19.015438	29.154857	2021-02-19	35710

107835 rows × 6 columns

In [8]:
```
country_confirmed_groupby = time_series_long.groupby(['Date', 'Country/Region'])['Confirmed'].sum()
df_country_confirmed = pd.DataFrame(country_confirmed_groupby).reset_index()
df_country_confirmed
```

Out[8]:

	Date	Country/Region	Confirmed
0	2020-01-22	Afghanistan	0
1	2020-01-22	Albania	0
2	2020-01-22	Algeria	0
3	2020-01-22	Andorra	0
4	2020-01-22	Angola	0
...	...	...	...
75835	2021-02-19	Vietnam	2362
75836	2021-02-19	West Bank and Gaza	171717
75837	2021-02-19	Yemen	2157
75838	2021-02-19	Zambia	73203
75839	2021-02-19	Zimbabwe	35710

75840 rows × 3 columns

In [9]:
```
country_confirmed = df_country_confirmed.sort_values('Confirmed', ascending=True)
```

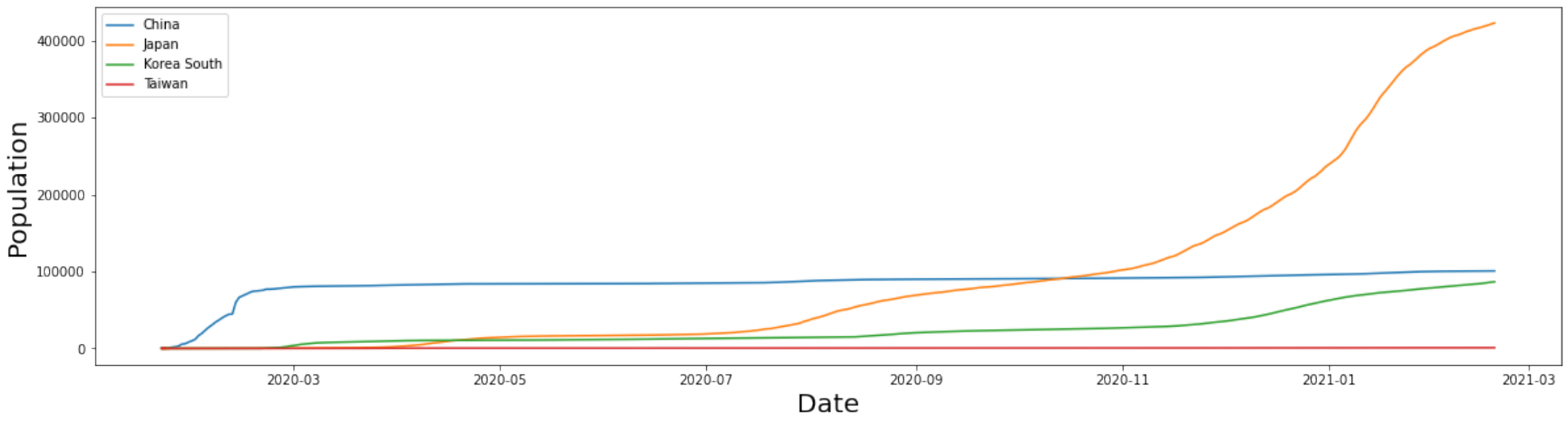
In [10]:
```
#us = country_confirmed[country_confirmed['Country/Region'].str.contains('US')]
#If we use USA plot on this pic, we won't see the other Country's progress.

cn = country_confirmed[country_confirmed['Country/Region'].str.contains('China')]
jpn = country_confirmed[country_confirmed['Country/Region'].str.contains('Japan')]
kr = country_confirmed[country_confirmed['Country/Region'].str.contains('Korea, South')]
tw = country_confirmed[country_confirmed['Country/Region'].str.contains('Taiwan')]

plt.figure(figsize=(20, 5))
plt.title('Covid 19 Confirmed Results', fontsize=20)
plt.xlabel('Date', fontsize=20)
plt.ylabel('Population', fontsize=20)

#plt.plot(us['Date'], us['Confirmed'], label='USA')
plt.plot(cn['Date'], cn['Confirmed'], label='China')
plt.plot(jpn['Date'], jpn['Confirmed'], label='Japan')
plt.plot(kr['Date'], kr['Confirmed'], label='Korea South')
plt.plot(tw['Date'], tw['Confirmed'], label='Taiwan')
plt.legend(loc=2)

plt.show()
```


Covid 19 Confirmed Results
China
Japan
Korea South
Taiwan
Population
400000
300000
200000
100000
0
2020-03
2020-05
2020-07
2020-09
2020-11
2021-01
2021-03
Date