Section 4 Charts and APIs and Data Analysis

You dont need anything other than the basic pre requisite libraries for this section :)

In [ ]:

```
#Importing libraries esp requests
import requests
import pandas as pd
import matplotlib.pyplot as plt
import matplotlib.ticker as ticker


#Listing all urls to call here
dayone = "https://api.covid19api.com/dayone/country/singapore"
todayglobalsummary = "https://api.covid19api.com/summary"
sgdailycases = "https://api.covid19api.com/dayone/country/singapore/status/recovered"
sgrunningtotal = "https://api.covid19api.com/total/dayone/country/singapore/status/recovered"
```

Calling the covid 19 API to scrape data and pull them into a dataframe using pandas

- Data is looking good so far, although we can be quite sure the cases displayed here are CUMULATIVE, not daily.

In [ ]:

```
response = requests.request("GET", dayone)

singapore = pd.DataFrame(response.json())

#Returns Cumulative Total for Singapore for all confirmed/active/deaths/recovered cases each day beginnin
g 01 Jan 2020
singapore
```

	ID	Country	CountryCode	Province	City	CityCode	Lat	Lon	Confirmed	Deaths	Recovered	Active	
0	f1471591-909d-4c12-9e8b-26ce43fa3b4e	Singapore	SG				1.35	103.82	1	0	0	1	2020 23T0
1	ad7bf6e4-27b0-46e9-bdc5-998458be7e6c	Singapore	SG				1.35	103.82	3	0	0	3	2020 24T0
2	25bb999d-2443-4505-925d-27f21af143f6	Singapore	SG				1.35	103.82	3	0	0	3	2020 25T0
3	297dd8d4-5417-4922-9ac9-a93b9902d4c8	Singapore	SG				1.35	103.82	4	0	0	4	2020 26T0
4	bee2a95e-2cf6-46e5-b2a5-9f4b4cca9613	Singapore	SG				1.35	103.82	5	0	0	5	2020 27T0
...	...	...	...	...	...	...	...	...	...	...	...	...	...
680	691eaac8-5a1c-46f0-b532-199607de0e93	Singapore	SG				1.35	103.82	267916	744	0	267172	2021 03T0
681	a6c9290f-66cb-4395-972a-54d27c5acf05	Singapore	SG				1.35	103.82	268659	746	0	267913	2021 04T0
682	3edd9b12-f2e3-413e-8625-a9661bf66388	Singapore	SG				1.35	103.82	269211	759	0	268452	2021 05T0
683	9fb7b3f3-3e05-4a46-8e2b-9357f44159ed	Singapore	SG				1.35	103.82	269873	763	0	269110	2021 06T0
684	741fc758-f044-4adb-b72d-d8a70a06b502	Singapore	SG				1.35	103.82	269873	763	0	269110	2021 07T0

685 rows × 13 columns

Since this is cumulative, i want additional columns taking the difference of the latest row, with the row before so i know the daily cases for each status

In [ ]:

```
for i in range(1, len(singapore)):
    singapore.loc[i, 'Daily_Confirmed'] = singapore.loc[i, 'Confirmed'] - singapore.loc[i-1, 'Confirmed']
    singapore.loc[i, 'Daily_Deaths'] = singapore.loc[i, 'Deaths'] - singapore.loc[i-1, 'Deaths']
    singapore.loc[i, 'Daily_Recovered'] = singapore.loc[i, 'Recovered'] - singapore.loc[i-1, 'Recovered']
    singapore.loc[i, 'Daily_Active'] = singapore.loc[i, 'Active'] - singapore.loc[i-1, 'Active']

singapore
```

	ID	Country	CountryCode	Province	City	CityCode	Lat	Lon	Confirmed	Deaths	Recovered	Active	
0	f1471591-909d-4c12-9e8b-26ce43fa3b4e	Singapore	SG				1.35	103.82	1	0	0	1	2020 23T0
1	ad7bf6e4-27b0-46e9-bdc5-998458be7e6c	Singapore	SG				1.35	103.82	3	0	0	3	2020 24T0
2	25bb999d-2443-4505-925d-27f21af143f6	Singapore	SG				1.35	103.82	3	0	0	3	2020 25T0
3	297dd8d4-5417-4922-9ac9-a93b9902d4c8	Singapore	SG				1.35	103.82	4	0	0	4	2020 26T0
4	bee2a95e-2cf6-46e5-b2a5-9f4b4cca9613	Singapore	SG				1.35	103.82	5	0	0	5	2020 27T0
...	...	...	...	...	...	...	...	...	...	...	...	...	...
680	691eaac8-5a1c-46f0-b532-199607de0e93	Singapore	SG				1.35	103.82	267916	744	0	267172	2021 03T0
681	a6c9290f-66cb-4395-972a-54d27c5acf05	Singapore	SG				1.35	103.82	268659	746	0	267913	2021 04T0
682	3edd9b12-f2e3-413e-8625-a9661bf66388	Singapore	SG				1.35	103.82	269211	759	0	268452	2021 05T0
683	9fb7b3f3-3e05-4a46-8e2b-9357f44159ed	Singapore	SG				1.35	103.82	269873	763	0	269110	2021 06T0
684	741fc758-f044-4adb-b72d-d8a70a06b502	Singapore	SG				1.35	103.82	269873	763	0	269110	2021 07T0

685 rows × 17 columns

In [ ]:

```
singapore.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 685 entries, 0 to 684
Data columns (total 17 columns):
 #   Column           Non-Null Count  Dtype  
---  ------           --------------  -----  
 0   ID               685 non-null    object 
 1   Country          685 non-null    object 
 2   CountryCode      685 non-null    object 
 3   Province         685 non-null    object 
 4   City             685 non-null    object 
 5   CityCode         685 non-null    object 
 6   Lat              685 non-null    object 
 7   Lon              685 non-null    object 
 8   Confirmed        685 non-null    int64  
 9   Deaths           685 non-null    int64  
 10  Recovered        685 non-null    int64  
 11  Active           685 non-null    int64  
 12  Date             685 non-null    object 
 13  Daily_Confirmed  684 non-null    float64
 14  Daily_Deaths     684 non-null    float64
 15  Daily_Recovered  684 non-null    float64
 16  Daily_Active     684 non-null    float64
dtypes: float64(4), int64(4), object(9)
memory usage: 91.1+ KB
```

In [ ]:

```
singapore.dtypes
```

```
ID                 object
Country            object
CountryCode        object
Province           object
City               object
CityCode           object
Lat                object
Lon                object
Confirmed           int64
Deaths              int64
Recovered           int64
Active              int64
Date               object
Daily_Confirmed   float64
Daily_Deaths      float64
Daily_Recovered   float64
Daily_Active      float64
dtype: object
```

So far so good, data is as expected so far, now time to drop unnecessary columns and drop missing values and also changing index to datetime

In [ ]:

```
singapore['Date'] = pd.to_datetime(singapore['Date'], infer_datetime_format=True)
singapore = singapore.set_index('Date')
singapore = singapore.drop(['Province','City','CityCode','CountryCode','Lat','Lon'],axis = 1)
singapore.dropna(inplace=True)
singapore
```

Date	ID	Country	Confirmed	Deaths	Recovered	Active	Daily_Confirmed	Daily_Deaths	Daily_Recovered	Daily_
2020-01-24	ad7bf6e4-27b0-46e9-bdc5-998458be7e6c	Singapore	3	0	0	3	2.0	0.0	0.0	2.0
2020-01-25	25bb999d-2443-4505-925d-27f21af143f6	Singapore	3	0	0	3	0.0	0.0	0.0	0.0
2020-01-26	297dd8d4-5417-4922-9ac9-a93b9902d4c8	Singapore	4	0	0	4	1.0	0.0	0.0	1.0
2020-01-27	bee2a95e-2cf6-46e5-b2a5-9f4b4cca9613	Singapore	5	0	0	5	1.0	0.0	0.0	1.0
2020-01-28	64b738cc-1d17-41c1-a9aa-d60900d57140	Singapore	7	0	0	7	2.0	0.0	0.0	2.0
...	...	...	...	...	...	...	...	...	...	...
2021-12-03	691eaac8-5a1c-46f0-b532-199607de0e93	Singapore	267916	744	0	267172	766.0	9.0	0.0	757.0
2021-12-04	a6c9290f-66cb-4395-972a-54d27c5acf05	Singapore	268659	746	0	267913	743.0	2.0	0.0	741.0
2021-12-05	3edd9b12-f2e3-413e-8625-a9661bf66388	Singapore	269211	759	0	268452	552.0	13.0	0.0	539.0
2021-12-06	9fb7b3f3-3e05-4a46-8e2b-9357f44159ed	Singapore	269873	763	0	269110	662.0	4.0	0.0	658.0
2021-12-07	741fc758-f044-4adb-b72d-d8a70a06b502	Singapore	269873	763	0	269110	0.0	0.0	0.0	0.0

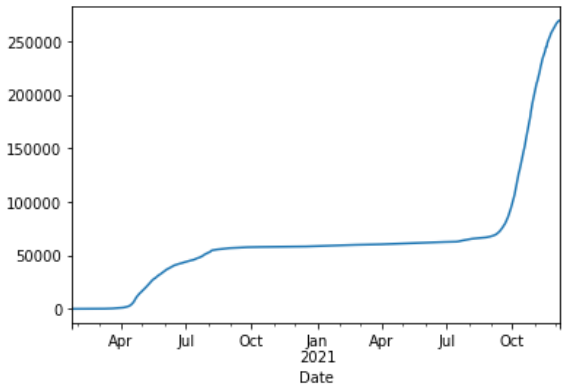
684 rows × 10 columns

First plot - Singapore Cumulative Confirmed COVID-19 Cases

In [ ]:

```
singapore['Confirmed'].plot()
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x242653e75c0>
```

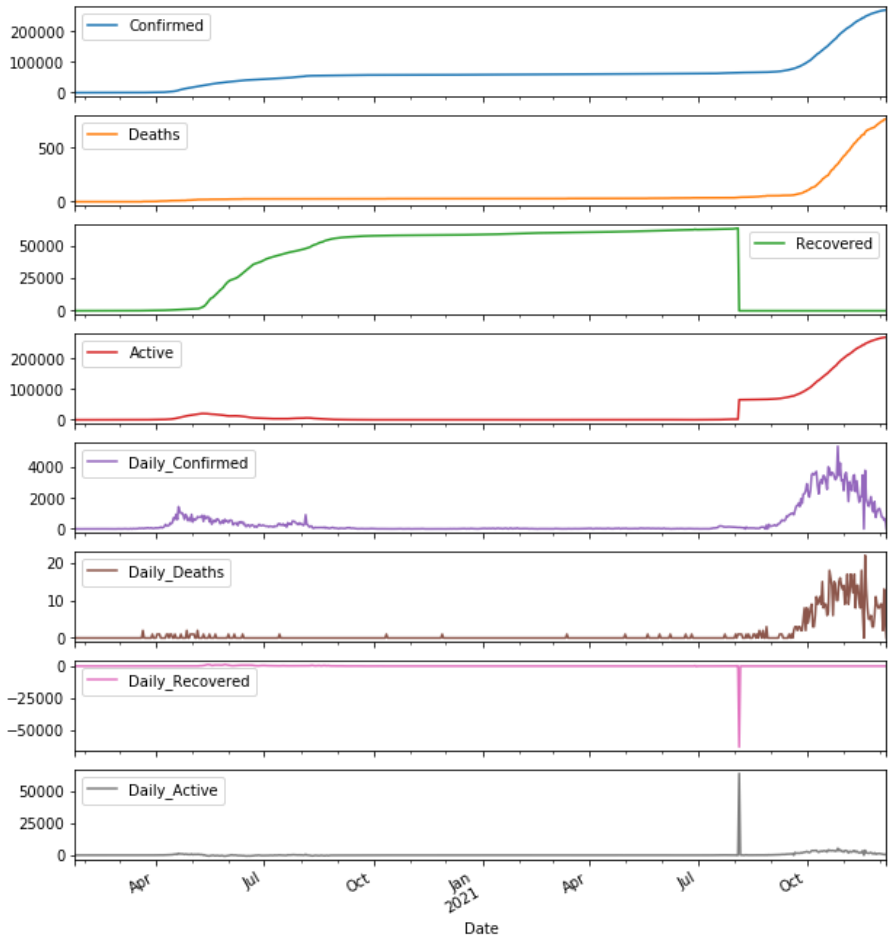


Second plot - All subplots for both cumulative status and daily statuses. Looks like there could be something wrong!

In [ ]:

```
singapore.plot(subplots=True, figsize=(10,12))
```

```
array([<matplotlib.axes._subplots.AxesSubplot object at 0x0000024265505390>,
       <matplotlib.axes._subplots.AxesSubplot object at 0x000002426553BFD0>,
       <matplotlib.axes._subplots.AxesSubplot object at 0x0000024265572470>,
       <matplotlib.axes._subplots.AxesSubplot object at 0x00000242655A08D0>,
       <matplotlib.axes._subplots.AxesSubplot object at 0x00000242655CFD30>,
       <matplotlib.axes._subplots.AxesSubplot object at 0x000002426560B1D0>,
       <matplotlib.axes._subplots.AxesSubplot object at 0x0000024265638630>,
       <matplotlib.axes._subplots.AxesSubplot object at 0x0000024265669AC8>],
      dtype=object)
```



There could be something wrong with the data above

- It seems that at 2021-08-04 the **API Counter resetted** and **total recovered cases suddenly dropped to 0** and total active cases suddenly increased from 2014 to 65468 cases on 2021-08-05 hmmm
- Lets attempt to verify this further before proceeding
- This returns the days where something went wrong with the API, as you can see, the daily recovered cases are negative, which means somehow the cumulative recovered cases all dropped to 0 on the 2021-08-05

In [ ]:

```
## This returns the days where something went wrong with the API, as you can see, the daily recovered cas
es are negative, which means somehow the cumulative recovered cases all dropped to 0 on the 2021-08-05
singapore[singapore.Daily_Recovered < 0]
```

Date	ID	Country	Confirmed	Deaths	Recovered	Active	Daily_Confirmed	Daily_Deaths	Daily_Recovered	Daily_
2021-06-30	815a14db-566e-4c1f-a0ee-94310472b688	Singapore	62579	36	62228	315	16.0	0.0	-335.0	7.0
2021-08-05	c0d0131e-4312-484f-8f83-42fba788ad84	Singapore	65508	40	0	65468	98.0	1.0	-63357.0	63454.

- This returns one day before the previous 2021-08-05 for 2021-08-04 where you can see there were 63357 covid cases cumulative who have recovered but the next day this counter has resetted to 0.
- How strange! Perhaps the count was resetted to zero due to some API error?
- Lets verify this with the other API endpoints as well.

In [ ]:

```
## This returns one day before the previous 2021-08-05 for 2021-08-04 where you can see there were 63357
 covid cases cumulative who have recovered but the next day this counter has resetted to 0.
singapore.loc['20210804']
```

```
ID                 170da623-3a7a-4225-9fe3-fa83c8b81694
Country                                       Singapore
Confirmed                                         65410
Deaths                                               39
Recovered                                         63357
Active                                             2014
Daily_Confirmed                                      95
Daily_Deaths                                          1
Daily_Recovered                                     105
Daily_Active                                        -11
Name: 2021-08-04 00:00:00, dtype: object
```

In [ ]:

```
## This confirms my suspicions that even using a different API endpoint, the results for recovered and ac
tive cases seem to be the same.

response = requests.request("GET", sgrunningtotal)
verification = pd.DataFrame(response.json())
verification[verification.Cases == 63357]
```

	Country	CountryCode	Province	City	CityCode	Lat	Lon	Cases	Status	Date
559	Singapore					0	0	63357	recovered	2021-08-04T00:00:00Z

In [ ]:

```
## All API endpoints return the same data for 2021-08-05 when cumulative data drops to 0 for all recoverd cases

response = requests.request("GET", sgdailycases)
test = pd.DataFrame(response.json())
test.loc[560]
```

```
Country                 Singapore
CountryCode                    SG
Province
City
CityCode
Lat                           1.35
Lon                         103.82
Cases                            0
Status                   recovered
Date           2021-08-05T00:00:00Z
Name: 560, dtype: object
```

As suspected all API endpoints return the same data, the problem could be due to the API itself, unlikely to be a result of Singapore's COVID cases

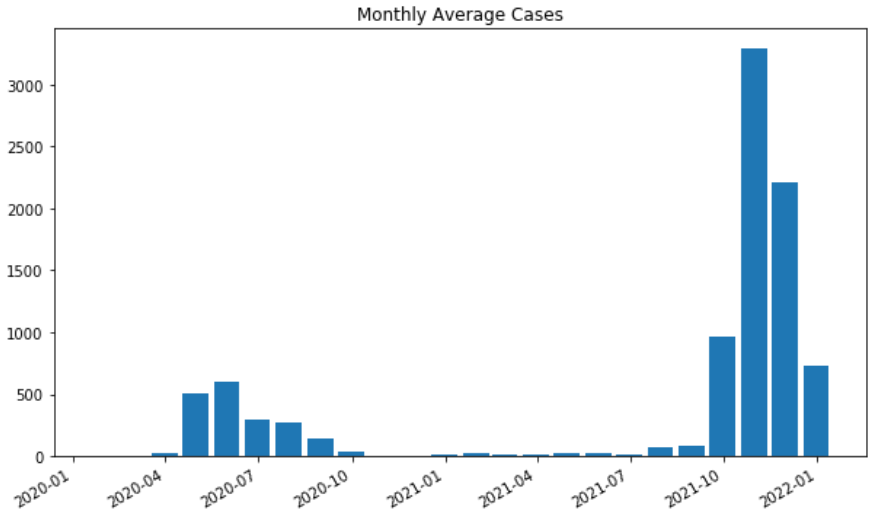
Now it seems like we have two choices, we can try to rectify the above data by replacing erranous data with an averaged value, or we could perhaps focus on daily_deaths and daily_cpnfirmed cases. I will focus on the latter. I am going to add more features into our dataset, so we can plot moving averages for confirmed covid cases. First, i will try to find seasonality within daily_confirmed cases. From this point onwards i will only be looking at confirmed status.

Seasonality

Resampling for months or weeks and making bar plots is another very simple and widely used method of finding seasonality. Here I am making a bar plot of month data for 2020 and 2021. For the index, I will use [2020:]. Because our dataset contains data until 2021. So, 2020 to end should bring 2020 and 2021.

In [ ]:

```
singapore_month = singapore.resample("M").mean()
fig, ax = plt.subplots(figsize=(10, 6))
ax.bar(singapore_month['2020':].index, singapore_month.loc['2020':, "Daily_Confirmed"], width=25, align='center')
ax.xaxis_date()
plt.title('Monthly Average Cases')
fig.autofmt_xdate()
```



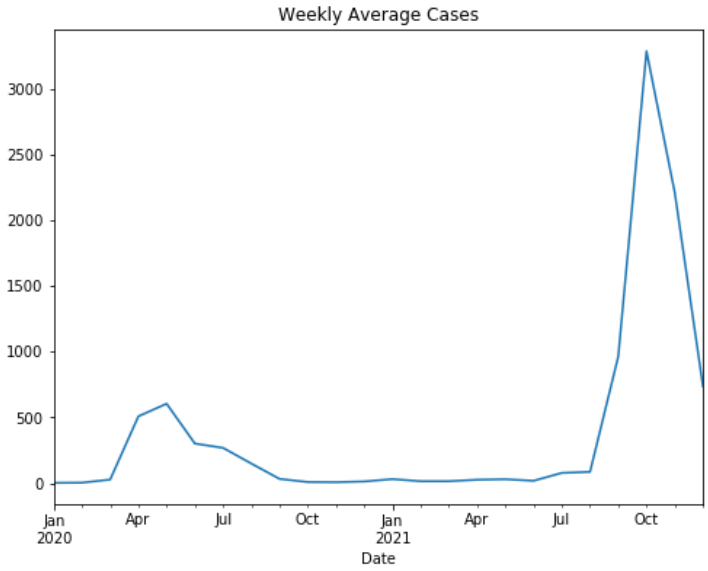
Third plot - Resampling daily cases to give a monthly average as a bar chart

In [ ]:

```
singapore_month["Daily_Confirmed"].plot(figsize=(8, 6))

ax.set_ylabel('Cases')
ax.xaxis_date()
ax.xaxis.set_major_locator(ticker.MultipleLocator(21))
fig.autofmt_xdate()
plt.title('Weekly Average Cases')
ax.legend()
plt.show()
```

```
No handles with labels found to put in legend.
```



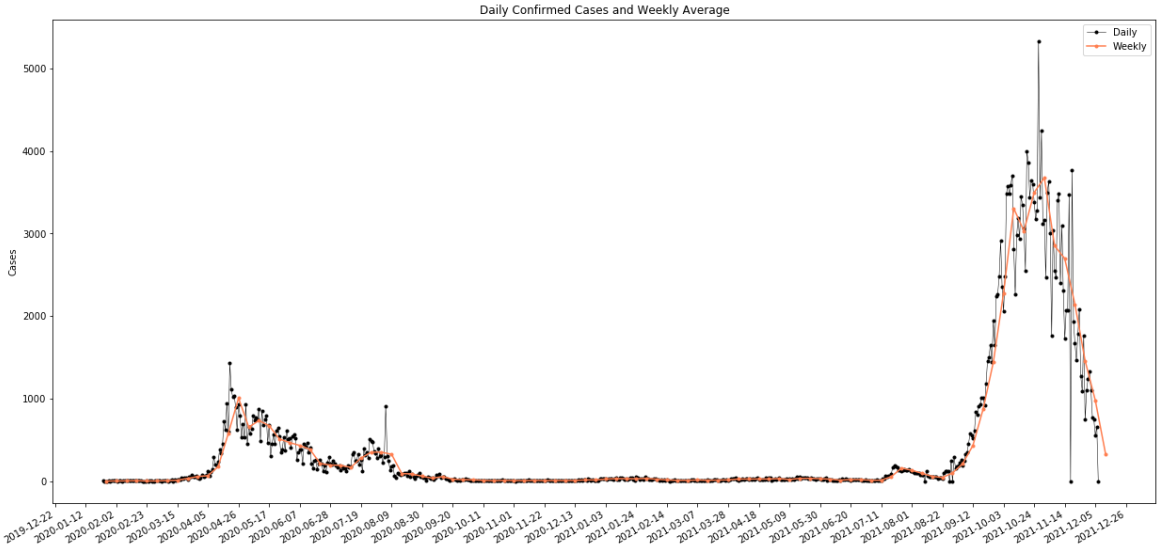
Fourth Plot - Resample daily cases to obtain weekly average as line chart

In [ ]:

```
singapore_week = singapore.resample("W").mean()

start, end = '2020-01', '2021-12'
fig, ax = plt.subplots()
ax.plot(singapore.loc[start:end, 'Daily_Confirmed'], marker='.', linestyle='-', linewidth = 0.5, label='Daily', color='black')
ax.plot(singapore_week.loc[start:end, 'Daily_Confirmed'], marker='o', markersize=3.0, linestyle='-', label='Weekly', color='coral')

ax.set_ylabel('Cases')
ax.xaxis_date()
ax.xaxis.set_major_locator(ticker.MultipleLocator(21))
fig.autofmt_xdate()
fig.set_size_inches(21, 10.5)
plt.title('Daily Confirmed Cases and Weekly Average')
ax.legend()
plt.show()
```



Fifth plot - Daily confirmed cases and weekly average cases

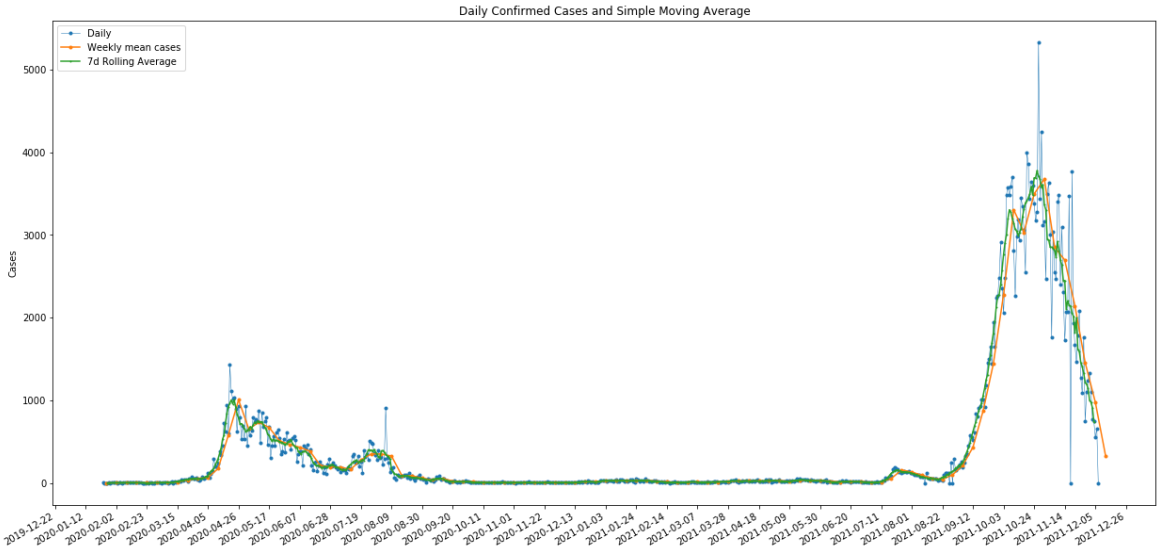
In [ ]:

```
singapore_7d_rolling = singapore.rolling(7, center=True).mean()
start, end = '2020-01', '2021-12'

fig, ax = plt.subplots()

ax.plot(singapore.loc[start:end, 'Daily_Confirmed'], marker='.', linestyle='-',
        linewidth=0.5, label='Daily')
ax.plot(singapore_week.loc[start:end, 'Daily_Confirmed'], marker='o', markersize=3,
        linestyle='-', label = 'Weekly mean cases')
ax.plot(singapore_7d_rolling.loc[start:end, 'Daily_Confirmed'], marker='.', markersize = 2, linestyle='-',
label='7d Rolling Average')


ax.set_ylabel('Cases')
ax.xaxis_date()
ax.xaxis.set_major_locator(ticker.MultipleLocator(21))
fig.autofmt_xdate()
fig.set_size_inches(21, 10.5)
plt.title('Daily Confirmed Cases and Simple Moving Average')
ax.legend()
plt.show()
```



Sixth plot - Daily confirmed cases, Weekly average confirmed cases and Simple moving average over 7 days