In [1]:
```
import os, re, glob
import pandas as pd
import numpy as np

import matplotlib.pyplot as plt
import folium
from sklearn.preprocessing import StandardScaler
from sklearn.cluster import KMeans
from IPython.display import IFrame
from IPython.display import Image
from tqdm import tqdm
import warnings
warnings.filterwarnings('ignore')

%matplotlib inline
```

This workshop would analyse the COVID -19 cases in the winter of 2022. Peak of winter is in January and February. the report from this workshop would be compared to the summer of the previous year to see if there is a difference in the upsurge of COVID 19 cases at both times.

In [2]:
```
root = 'C:\\Users\\User\\Documents\\COVID'
recent_date = "01-01-2022"
previous_date = "02-01-2022"

duplicate_columns = {"Lat": "Latitude",
                     "Long_": "Longitude",
                     "Incidence_Rate": "Incident_Rate",
                     "Case-Fatality_Ratio": "Case_Fatality_Ratio",
                     "Province/State": "Province_State",
                     "Country/Region": "Country_Region",
                     "Last Update": "Last_Update"}

recent_df = pd.read_csv(os.path.join(root, (recent_date + ".csv")))
previous_df = pd.read_csv(os.path.join(root, (previous_date + ".csv")))

for key, value in duplicate_columns.items():
    if key in recent_df.columns:
        recent_df = recent_df.rename(columns={key: value})
    if key in previous_df.columns:
        previous_df = previous_df.rename(columns={key: value})
```

In [3]:
```
recent_df.head()
```

Out[3]:

	FIPS	Admin2	Province_State	Country_Region	Last_Update	Latitude	Longitude	Confirmed	Deaths	Recovered	Active	Combined_Key	Incident_R
0	NaN	NaN	NaN	Afghanistan	2022-01-02 04:20:52	33.93911	67.709953	158107	7356	NaN	NaN	Afghanistan	406.148
1	NaN	NaN	NaN	Albania	2022-01-02 04:20:52	41.15330	20.168300	210224	3217	NaN	NaN	Albania	7305.024
2	NaN	NaN	NaN	Algeria	2022-01-02 04:20:52	28.03390	1.659600	218818	6284	NaN	NaN	Algeria	499.002
3	NaN	NaN	NaN	Andorra	2022-01-02 04:20:52	42.50630	1.521800	23740	140	NaN	NaN	Andorra	30725.425
4	NaN	NaN	NaN	Angola	2022-01-02 04:20:52	-11.20270	17.873900	82398	1772	NaN	NaN	Angola	250.706

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

Out[4]:

	FIPS	Admin2	Province_State	Country_Region	Last_Update	Latitude	Longitude	Confirmed	Deaths	Recovered	Active	Combined_Key	Incident_R
0	NaN	NaN	NaN	Afghanistan	2022-02-02 04:21:09	33.93911	67.709953	163555	7417	NaN	NaN	Afghanistan	420.143
1	NaN	NaN	NaN	Albania	2022-02-02 04:21:09	41.15330	20.168300	258543	3346	NaN	NaN	Albania	8984.050
2	NaN	NaN	NaN	Algeria	2022-02-02 04:21:09	28.03390	1.659600	253520	6593	NaN	NaN	Algeria	578.139
3	NaN	NaN	NaN	Andorra	2022-02-02 04:21:09	42.50630	1.521800	35958	145	NaN	NaN	Andorra	46538.536
4	NaN	NaN	NaN	Angola	2022-02-02 04:21:09	-11.20270	17.873900	98226	1895	NaN	NaN	Angola	298.865

In [5]:
```
current_df = pd.DataFrame(columns=['Province_State','Country_Region','Confirmed','Deaths'])
current_df['Province_State'] = recent_df['Province_State']
current_df['Country_Region'] = recent_df['Country_Region']
current_df['Confirmed'] = recent_df['Confirmed'] - previous_df['Confirmed']
current_df['Deaths'] = recent_df['Deaths'] - previous_df['Deaths']
```

In [6]:
```
current_df.shape
```

Out[6]:
```
(4016, 4)
```

In [7]:
```
current_df.head()
```

Out[7]:

	Province_State	Country_Region	Confirmed	Deaths
0	NaN	Afghanistan	-5448	-61
1	NaN	Albania	-48319	-129
2	NaN	Algeria	-34702	-309
3	NaN	Andorra	-12218	-5
4	NaN	Angola	-15828	-123

In [8]:
```
#we cannot have negative values, so we get the absolute values of the negative values in the Confirmed and Death columns
current_df['Confirmed'] = current_df['Confirmed'].abs()
current_df['Deaths'] = current_df['Deaths'].abs()
```

In [9]:
```
current_df.head()
```

Out[9]:

	Province_State	Country_Region	Confirmed	Deaths
0	NaN	Afghanistan	5448	61
1	NaN	Albania	48319	129
2	NaN	Algeria	34702	309
3	NaN	Andorra	12218	5
4	NaN	Angola	15828	123

In [10]:
```
name_number = 'DekeAdeleye_2229810b.csv'
current_df.to_csv(name_number, index=False)
```

In [11]:
```
data = pd.read_csv(name_number)
```

In [12]:
```
data.head()
```

Out[12]:

	Province_State	Country_Region	Confirmed	Deaths
0	NaN	Afghanistan	5448	61
1	NaN	Albania	48319	129
2	NaN	Algeria	34702	309
3	NaN	Andorra	12218	5
4	NaN	Angola	15828	123

In [13]:
```
print(data.shape)
```

```
(4016, 4)
```

In [14]:
```
print(data.count())
```

```
Province_State    3837
Country_Region    4016
Confirmed         4016
Deaths            4016
dtype: int64
```

In [15]:
```
#Q1.  To print how many null values exist in the dataset.
print(data.isnull().sum().sum())
```

```
179
```

In [16]:
```
data.apply(lambda x: sum(x.isnull()), axis = 0)
```

Out[16]:
```
Province_State    179
Country_Region      0
Confirmed           0
Deaths              0
dtype: int64
```

In [17]:
```
data.loc[data['Province_State'].isnull(),'Province_State'] = data['Country_Region']
```

In [18]:
```
data.head()
```

Out[18]:

	Province_State	Country_Region	Confirmed	Deaths
0	Afghanistan	Afghanistan	5448	61
1	Albania	Albania	48319	129
2	Algeria	Algeria	34702	309
3	Andorra	Andorra	12218	5
4	Angola	Angola	15828	123

In [19]:
```
states = data['Province_State'].unique()
print("Number of unique States - ", len(states))
```

```
Number of unique States -  774
```

In [20]:
```
#Q2. How many unique countries exist in the dataset using a similar approach.
countries = data['Country_Region'].unique()
print("Number of unique countries - ", len(countries))
```

```
Number of unique countries -  201
```

In [21]:
```
import datetime, time, requests
from time import sleep
from geopy.geocoders import Nominatim


def get_lat_lon(place):
    geolocator = Nominatim(user_agent=name_number)
    location = geolocator.geocode(place)
    lat_lon = location.latitude, location.longitude

    output = [float(i) for i in lat_lon]
    return output
```

In [22]:
```
#data['Province_State'].value_counts()
```

In [23]:
```
from tqdm import tqdm

geo_lat = []
geo_lon = []

not_found = []
found = []
for state in tqdm(states):
    time.sleep(0.2)
    lat_lon = [None, None]
    try:
        lat_lon = get_lat_lon(state)
        found.append(state)
    except:
        not_found.append(state)

    geo_lat.append(lat_lon[0])
    geo_lon.append(lat_lon[1])

if len(not_found) > 0:
    print("Locations are not found for - ", not_found)
else:
    print("Found all the locations")

#if len(found) > 0:
#    print("Locations are found for - ", found)
```

```
100%|██████████| 774/774 [06:35<00:00,  1.95it/s]

Locations are not found for -  ['Repatriated Travellers', 'W.P. Kuala Lumpur', 'Sakha (Yakutiya) Republic', 'Summer Olympi
cs 2020']
```

In [24]:
```
states_list = states.tolist() #converting states to list to index list's items
lats = []
lons = []
for i, r in data.iterrows():
    state = r['Province_State']
    index_list = states_list.index(state)
    lats.append(geo_lat[index_list])
    lons.append(geo_lon[index_list])


data['Latitude'] = lats
data['Longitude'] = lons
```

In [25]:
```
data.head()
```

Out[25]:

	Province_State	Country_Region	Confirmed	Deaths	Latitude	Longitude
0	Afghanistan	Afghanistan	5448	61	33.768006	66.238514
1	Albania	Albania	48319	129	41.000028	19.999962
2	Algeria	Algeria	34702	309	28.000027	2.999983
3	Andorra	Andorra	12218	5	42.540717	1.573203
4	Angola	Angola	15828	123	-11.877577	17.569124

In [ ]:
```

```

In [26]:
```
data.head()
```

Out[26]:

	Province_State	Country_Region	Confirmed	Deaths	Latitude	Longitude
0	Afghanistan	Afghanistan	5448	61	33.768006	66.238514
1	Albania	Albania	48319	129	41.000028	19.999962
2	Algeria	Algeria	34702	309	28.000027	2.999983
3	Andorra	Andorra	12218	5	42.540717	1.573203
4	Angola	Angola	15828	123	-11.877577	17.569124

In [27]:
```
data.shape
```

Out[27]:
```
(4016, 6)
```

In [28]:
```
#remove null values from the longitude and latitude columns.
#we will use the mean value to replace the null values
data['Latitude'].fillna(data['Latitude'].mean(), inplace = True)
data['Longitude'].fillna(data['Longitude'].mean(), inplace = True)
```

In [29]:
```
data.head()
```

Out[29]:

	Province_State	Country_Region	Confirmed	Deaths	Latitude	Longitude
0	Afghanistan	Afghanistan	5448	61	33.768006	66.238514
1	Albania	Albania	48319	129	41.000028	19.999962
2	Algeria	Algeria	34702	309	28.000027	2.999983
3	Andorra	Andorra	12218	5	42.540717	1.573203
4	Angola	Angola	15828	123	-11.877577	17.569124

In [30]:
```
clustering_data = data[["Confirmed", "Deaths"]]
```

In [31]:
```
clustering_data.head()
```

Out[31]:

	Confirmed	Deaths
0	5448	61
1	48319	129
2	34702	309
3	12218	5
4	15828	123

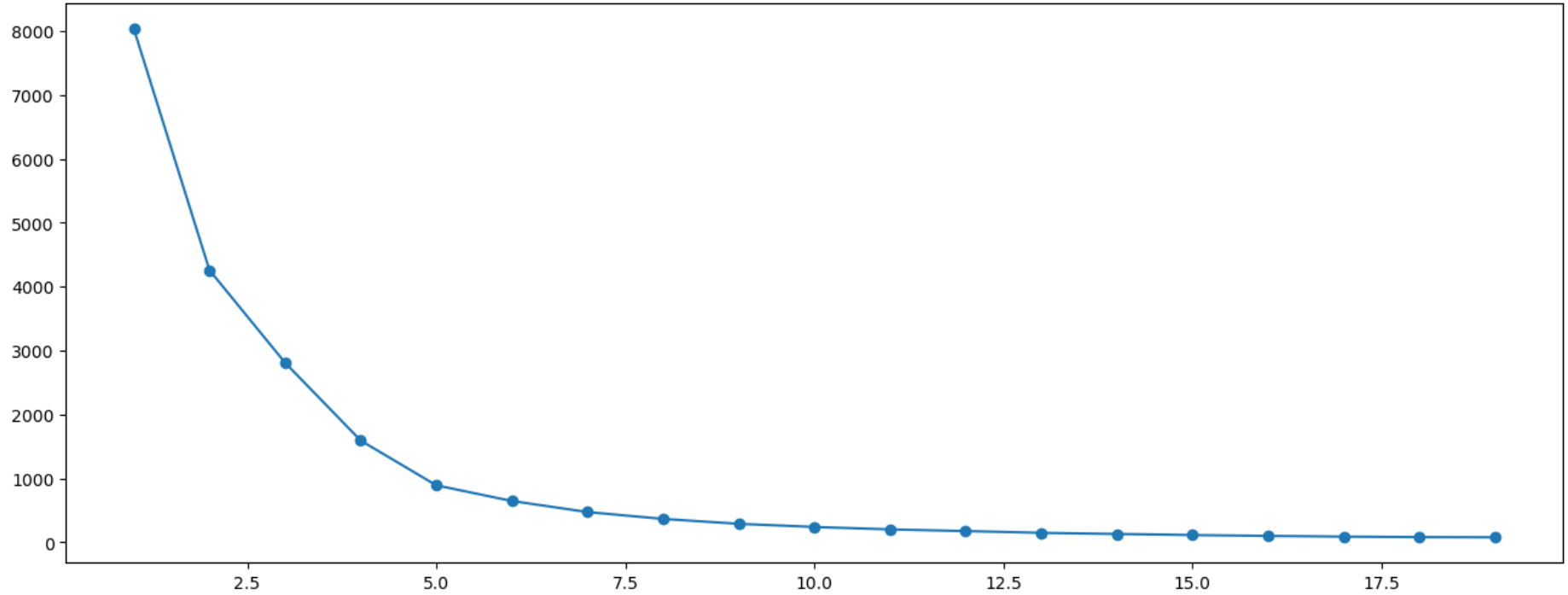
In [32]:
```
scaler = StandardScaler()
X_scaled = scaler.fit(clustering_data).transform(clustering_data.astype(np.float))
```

In [33]:
```
cluster_range = range( 1, 20 )
cluster_errors = []

for num_clusters in cluster_range:
    clusters = KMeans( num_clusters )
    clusters.fit( X_scaled )
    cluster_errors.append( clusters.inertia_ )

clusters_df = pd.DataFrame( { "num_clusters":cluster_range,
                              "cluster_errors": cluster_errors } )

plt.figure(figsize=(16,6))
plt.plot( clusters_df.num_clusters, clusters_df.cluster_errors, marker = "o" );
```



In [34]:
```
# Fitting K-Means to the dataset
kmeans = KMeans(n_clusters = 4, init = 'k-means++', random_state = 10)
y_kmeans = kmeans.fit_predict(X_scaled)

#beginning of  the cluster numbering with 1 instead of 0
y_kmeans1=y_kmeans+1

# New list called cluster
cluster = list(y_kmeans1)
# Adding cluster to our data set
clustering_data['cluster'] = cluster
```

In [35]:
```
clustering_data.head(10)
```

Out[35]:

	Confirmed	Deaths	cluster
0	5448	61	1
1	48319	129	1
2	34702	309	1
3	12218	5	1
4	15828	123	1
5	2344	8	1
6	2753350	4332	2
7	25942	81	1
8	32104	11	1
9	869983	780	1

In [36]:
```
kmeans_mean_cluster = pd.DataFrame(round(clustering_data.groupby('cluster').mean(),1))
kmeans_mean_cluster
```

Out[36]:

cluster	Confirmed	Deaths
1	18050.8	57.0
2	1821248.0	5954.2
3	33664.0	24101.0
4	9131815.0	7256.0

In [37]:
```
data['cluster'] = cluster
clusters = data[['Province_State', 'cluster']]
clusters.loc[clusters['cluster'] == 2]
```

Out[37]:

	Province_State	cluster
6	Argentina	2
265	Kerala	2
510	Poland	2
672	Turkey	2
3986	England	2
4001	Vietnam	2

In [38]:
```
data['cluster'] = cluster
clusters = data[['Province_State', 'cluster']]
clusters.loc[clusters['cluster'] == 3]
```

Out[38]:

	Province_State	cluster
1070	Florida	3

In [39]:
```
data['cluster'] = cluster
clusters = data[['Province_State', 'cluster']]
clusters.loc[clusters['cluster'] == 4]
```

Out[39]:

	Province_State	cluster
216	France	4

In [40]:
```
data.head()
```

Out[40]:

	Province_State	Country_Region	Confirmed	Deaths	Latitude	Longitude	cluster
0	Afghanistan	Afghanistan	5448	61	33.768006	66.238514	1
1	Albania	Albania	48319	129	41.000028	19.999962	1
2	Algeria	Algeria	34702	309	28.000027	2.999983	1
3	Andorra	Andorra	12218	5	42.540717	1.573203	1
4	Angola	Angola	15828	123	-11.877577	17.569124	1

In [41]:
```
def get_color(cluster_id):
    if cluster_id == 2:
        return 'darkred'
    if cluster_id == 1:
        return 'green'
    if cluster_id == 3:
        return 'orange'
    if cluster_id == 4:
        return 'yellow'

data["color"] = data["cluster"].apply(lambda x: get_color(x))
```

In [42]:
```
data.head(10)
```

Out[42]:

	Province_State	Country_Region	Confirmed	Deaths	Latitude	Longitude	cluster	color
0	Afghanistan	Afghanistan	5448	61	33.768006	66.238514	1	green
1	Albania	Albania	48319	129	41.000028	19.999962	1	green
2	Algeria	Algeria	34702	309	28.000027	2.999983	1	green
3	Andorra	Andorra	12218	5	42.540717	1.573203	1	green
4	Angola	Angola	15828	123	-11.877577	17.569124	1	green
5	Antigua and Barbuda	Antigua and Barbuda	2344	8	17.223472	-61.955461	1	green
6	Argentina	Argentina	2753350	4332	-34.996496	-64.967282	2	darkred
7	Armenia	Armenia	25942	81	40.769627	44.673665	1	green
8	Australian Capital Territory	Australia	32104	11	-35.488350	149.002694	1	green
9	New South Wales	Australia	869983	780	-31.875984	147.286949	1	green

In [43]:
```
#create a map
this_map = folium.Map(location =[data["Latitude"].mean(),
                                 data["Longitude"].mean()], zoom_start=5)

def plot_dot(point):
    '''input: series that contains a numeric named latitude and a numeric named longitude
    this function creates a CircleMarker and adds it to your this_map'''
    folium.CircleMarker(location=[point.Latitude, point.Longitude],
                        radius=2,
                        color=point.color,
                        weight=1).add_to(this_map)




#clustered_full.apply(,axis=1) #use this to iterate through every row in your dataframe
data.apply(plot_dot, axis = 1)


#Set the zoom to the maximum possible
this_map.fit_bounds(this_map.get_bounds())

#Save the map to an HTML file
this_map.save(os.path.join('covid_mapb.html'))
```

cluster 1 has the lowest number of cases with 18050 confirmd cases and 58 deaths, this shows a significant decrease in COVID-19 cases in the countries in these cluster. For instance, Afghanistan had a further decline in confirmed cases and deaths in winter 2022 compared to summer 2021.
Cluster 2 still had high number of cases in winter 2022, with 1821488 confirmed cases and 5954 deaths. The number of confirmed cases increased compared to summer 2021, even though the deaths reduced. One of the countries in cluster 2, Argentina moved from recording a moderate number of cases and deaths in summer 2021 to an upsurge in the cases. As at Jan 6, it was recorded that Argentina broke its record for COVID 19 infections, facing its third wave of the pandemic (Reuters, 2022).
Cluster 3 had as much confirmed cases as deaths. The number of confirmed cases and deaths were very minimal. Most likely, the countries in this cluster is not much, as it brought out only Florida. In Winter 2022, Florida witnessed an increase in the number of COVID 19 cases of the Omicron variant. Accrding to January 2022 was the month with the highest average cases, while September 2021 was the month with the highest average deaths in Florida (The New York Times, 2023).

```
Cluster 4 had an upsurge in the number of confirmed cases and deaths with 9131815 confirmed cases and 7256 deaths.
Countries in cluster 4 include France, which has been recorded as witnessing a high increase in number of COVID cases
during the winter.
```

Reuters (2022) Argentina breaks COVID-19 case record as daily infections near 100,000. Reuters[online]. 6 January 2022.[Accessed 03 March 2023] Available at: https://www.reuters.com/business/healthcare-pharmaceuticals/argentina-breaks-covid-19-case-record-daily-infections-near-100000-2022-01-06/ (https://www.reuters.com/business/healthcare-pharmaceuticals/argentina-breaks-covid-19-case-record-daily-infections-near-100000-2022-01-06/) The New York Times (2023) Tracking Coronavirus in Florida: Latest Map and Case Count. The New York Times [online]. 9 March.[Accessed 03 March 2023] Available at: https://www.nytimes.com/interactive/2021/us/florida-covid-cases.html (https://www.nytimes.com/interactive/2021/us/florida-covid-cases.html)