Open in Colab
(https://colab.research.google.com/github/Elzawawy/covid-case-estimator/blob/master/Worldometer_Daily_Data_Exploration.ipynb)

In [1]:

```
#imports cell
from shutil import copyfile
import pandas as pd
import numpy as np
import pickle

# mount google drive to copy files from repo into drive.
from google.colab import drive
drive.mount('/content/drive')
```

```
Go to this URL in a browser: https://accounts.google.com/o/oauth2/auth?client_id=947318989803-6
bn6qk8qdgf4n4g3pfee6491hc0brc4i.apps.googleusercontent.com&redirect_uri=urn%3aietf%3awg%3aoauth
%3a2.0%3aoob&response_type=code&scope=email%20https%3a%2f%2fwww.googleapis.com%2fauth%2fdocs.te
st%20https%3a%2f%2fwww.googleapis.com%2fauth%2fdrive%20https%3a%2f%2fwww.googleapis.com%2fauth%
2fdrive.photos.readonly%20https%3a%2f%2fwww.googleapis.com%2fauth%2fpeopleapi.readonly

Enter your authorization code:
··········
Mounted at /content/drive
```

Clone Repoistory to Temporary Drive and Sync latest data with the Storage directory on permanent google drive storage.

In [3]:

```
# download repoistory from Github and Sync with Drive data.
!git clone https://github.com/bumbeishvili/covid19-daily-data
REPO_DIR = "/content/covid19-daily-data/"
STORAGE_DIR = "/content/drive/My Drive/COVID-19/worldometer-daily-data/"
CONFIRMED_FILE = "time_series_19-covid-Confirmed.csv"
RECOVERED_FILE = "time_series_19-covid-Recovered.csv"
DEATHS_FILE = "time_series_19-covid-Deaths.csv"

# copy files from temproary repo directory to permanent drive directory.
copyfile(REPO_DIR+CONFIRMED_FILE, STORAGE_DIR+CONFIRMED_FILE);
copyfile(REPO_DIR+RECOVERED_FILE, STORAGE_DIR+RECOVERED_FILE);
copyfile(REPO_DIR+DEATHS_FILE, STORAGE_DIR+DEATHS_FILE);
```

```
fatal: destination path 'covid19-daily-data' already exists and is not an empty directory.
```

Let's Begin by understanding the Daily Confirmed Cases Table.

- Originally, it has 4 main columns **(Province/State - Country/Region - Long - Lat)** and more variable columns based on days.
- Each day constitues a column, thus since each day this dataset gets updated we basically get a new column added each day.
- The dataset starts at day **1/22/2020** which when Corona Vrius started appearing for the first time in China and ends at the day you will run this Notebook at while the Github Repoistory keeps updating each day.

In [4]:

```
confirmed_cases_dataframe = pd.read_csv(STORAGE_DIR+CONFIRMED_FILE)
confirmed_cases_dataframe.head()
```

Out[4]:

	Province/State	Country/Region	Long	Lat	1/22/20	1/23/20	1/24/20	1/25/20	1/26/20	1/27/20	1/28/2
0	Afghanistan	Afghanistan	65.0000	33.0000	0	0	0	0	0	0	
1	Albania	Albania	20.1683	41.1533	0	0	0	0	0	0	
2	Algeria	Algeria	1.6596	28.0339	0	0	0	0	0	0	
3	Andorra	Andorra	1.5218	42.5063	0	0	0	0	0	0	
4	Angola	Angola	17.8739	-11.2027	0	0	0	0	0	0	

5 rows × 118 columns

What we need to do to this dataset to match our needs ?

- We can drop the (Long - Lat - Province/State) columns since no need for them for us. **Why ?**
    - Longitude and lattiude is something that probably won't affect the coronavirus spead.
    - We are only interested in countries not states inside them.
- We will need to separate a dataset **for each country** with days as row instances and the cell values are the number of daily cases at this country **(1-D arrays: number of cases)**. Date of the day can be implicit itself from the bigger dataset.
- We will do this for each country and put them in a **Map (dict) structure** with country as key and it's 1-D array as value.

In [0]:

```
def create_cases_dict(cases_dataframe):
  cases_dataframe = cases_dataframe.drop(columns=["Long","Lat","Province/State"])
  country_cases = {}
  # iterate on each row, put the Country Name in key and rest of day values as 1-D array value.
  for row in cases_dataframe.itertuples(index=False):
    country_cases[row[0]] = np.array([day for day in row[1:]])
  return country_cases

def save_dict_to_pickle(dict, pickle_file):
  with open(pickle_file, 'wb') as handle:
    pickle.dump(dict, handle, protocol=pickle.HIGHEST_PROTOCOL)
```

In [0]:

```
confirmed_cases_dict = create_cases_dict(confirmed_cases_dataframe)
save_dict_to_pickle(confirmed_cases_dict, STORAGE_DIR+"confirmed_cases_dict.pickle")
```

We now repeat the same steps for Recovered Cases.

In [0]:

```
recovered_cases_dataframe = pd.read_csv(STORAGE_DIR+RECOVERED_FILE)
recovered_cases_dict = create_cases_dict(recovered_cases_dataframe)
save_dict_to_pickle(recovered_cases_dict, STORAGE_DIR+"recovered_cases_dict.pickle")
```

Lastly, repeat the same steps for Death Cases.

In [0]:

```
deaths_cases_dataframe = pd.read_csv(STORAGE_DIR+DEATHS_FILE)
deaths_cases_dict = create_cases_dict(deaths_cases_dataframe)
save_dict_to_pickle(deaths_cases_dict, STORAGE_DIR+"deaths_cases_dict.pickle")
```