

In [1]:

```
# Data collected from "https://covid19.who.int/WHO-COVID-19-global-data.csv"  
# pip install matplotlib  
# pip install pandas  
# pip install numpy
```

In [2]:

```
import os  
import urllib  
import matplotlib.pyplot as Mat  
import pandas as pd  
import numpy as np
```

In [3]:

```
url = "https://covid19.who.int/WHO-COVID-19-global-data.csv"  
file_path = os.path.join("data", "covid")
```

In [4]:

```
os.makedirs(file_path, exist_ok=True)  
csv_path = os.path.join(file_path, "WHO-COVID-19-Data.csv")  
urllib.request.urlretrieve(url, csv_path)
```

Out[4]:

```
('data\\covid\\WHO-COVID-19-Data.csv',  
<http.client.HTTPMessage at 0x1dd6a4b3100>)
```

In [5]:

```
DataF = pd.read_csv(csv_path)
```

In [6]:

DataF

Out[6]:

	Date_reported	Country_code	Country	WHO_region	New_cases	Cumulative_cases
0	2020-01-03	AF	Afghanistan	EMRO	0	0
1	2020-01-04	AF	Afghanistan	EMRO	0	0
2	2020-01-05	AF	Afghanistan	EMRO	0	0
3	2020-01-06	AF	Afghanistan	EMRO	0	0
4	2020-01-07	AF	Afghanistan	EMRO	0	0
...
150964	2021-09-26	ZW	Zimbabwe	AFRO	371	129505
150965	2021-09-27	ZW	Zimbabwe	AFRO	120	129625
150966	2021-09-28	ZW	Zimbabwe	AFRO	0	129625
150967	2021-09-29	ZW	Zimbabwe	AFRO	647	130272
150968	2021-09-30	ZW	Zimbabwe	AFRO	213	130485

150969 rows × 8 columns

In [7]:

```
DataF_index = DataF.index
DataF_index
```

Out[7]:

RangeIndex(start=0, stop=150969, step=1)

In [8]:

```
DataF_columns = DataF.columns
DataF_columns
```

Out[8]:

```
Index(['Date_reported', 'Country_code', 'Country', 'WHO_region', 'New_case
s',
      'Cumulative_cases', 'New_deaths', 'Cumulative_deaths'],
      dtype='object')
```

In [9]:

DataF_index.values

Out[9]:

array([0, 1, 2, ..., 150966, 150967, 150968], dtype=int64)

In [10]:

DataF.values

Out[10]:

```
array([[ '2020-01-03', 'AF', 'Afghanistan', ..., 0, 0, 0],
       [ '2020-01-04', 'AF', 'Afghanistan', ..., 0, 0, 0],
       [ '2020-01-05', 'AF', 'Afghanistan', ..., 0, 0, 0],
       ...,
       [ '2021-09-28', 'ZW', 'Zimbabwe', ..., 129625, 0, 4604],
       [ '2021-09-29', 'ZW', 'Zimbabwe', ..., 130272, 11, 4615],
       [ '2021-09-30', 'ZW', 'Zimbabwe', ..., 130485, 1, 4616]],
      dtype=object)
```

In [11]:

DataF.dtypes

Out[11]:

```
Date_reported      object
Country_code       object
Country            object
WHO_region         object
New_cases          int64
Cumulative_cases   int64
New_deaths         int64
Cumulative_deaths  int64
dtype: object
```

In [12]:

DataF.shape

Out[12]:

(150969, 8)

In [13]:

DataF.head()

Out[13]:

	Date_reported	Country_code	Country	WHO_region	New_cases	Cumulative_cases	New_
0	2020-01-03	AF	Afghanistan	EMRO	0	0	
1	2020-01-04	AF	Afghanistan	EMRO	0	0	
2	2020-01-05	AF	Afghanistan	EMRO	0	0	
3	2020-01-06	AF	Afghanistan	EMRO	0	0	
4	2020-01-07	AF	Afghanistan	EMRO	0	0	

In [14]:

DataF.tail()

Out[14]:

	Date_reported	Country_code	Country	WHO_region	New_cases	Cumulative_cases	
150964	2021-09-26	ZW	Zimbabwe	AFRO	371	129505	
150965	2021-09-27	ZW	Zimbabwe	AFRO	120	129625	
150966	2021-09-28	ZW	Zimbabwe	AFRO	0	129625	
150967	2021-09-29	ZW	Zimbabwe	AFRO	647	130272	
150968	2021-09-30	ZW	Zimbabwe	AFRO	213	130485	

In [15]:

DataF.info()

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 150969 entries, 0 to 150968
Data columns (total 8 columns):
#   Column                Non-Null Count  Dtype
---  -
0   Date_reported         150969 non-null object
1   Country_code          150332 non-null object
2   Country               150969 non-null object
3   WHO_region            150969 non-null object
4   New_cases             150969 non-null int64
5   Cumulative_cases     150969 non-null int64
6   New_deaths            150969 non-null int64
7   Cumulative_deaths    150969 non-null int64
dtypes: int64(4), object(4)
memory usage: 9.2+ MB
```

In [16]:

DataF.describe()

Out[16]:

	New_cases	Cumulative_cases	New_deaths	Cumulative_deaths
count	150969.000000	1.509690e+05	150969.000000	150969.000000
mean	1544.265028	3.362740e+05	31.605217	7724.168723
std	9615.481780	1.979656e+06	170.884896	38866.115217
min	-32952.000000	0.000000e+00	-60.000000	0.000000
25%	0.000000	4.000000e+01	0.000000	0.000000
50%	12.000000	4.510000e+03	0.000000	68.000000
75%	371.000000	8.094900e+04	5.000000	1339.000000
max	414188.000000	4.296694e+07	8786.000000	688099.000000

In [17]:

```
DataF["Country"]
```

Out[17]:

```
0      Afghanistan
1      Afghanistan
2      Afghanistan
3      Afghanistan
4      Afghanistan
```

...

```
150964    Zimbabwe
150965    Zimbabwe
150966    Zimbabwe
150967    Zimbabwe
150968    Zimbabwe
```

Name: Country, Length: 150969, dtype: object

In [18]:

```
DataF["Country"].unique()
```

Out[18]:

```
array(['Afghanistan', 'Albania', 'Algeria', 'American Samoa', 'Andorra',
      'Angola', 'Anguilla', 'Antigua and Barbuda', 'Argentina',
      'Armenia', 'Aruba', 'Australia', 'Austria', 'Azerbaijan',
      'Bahamas', 'Bahrain', 'Bangladesh', 'Barbados', 'Belarus',
      'Belgium', 'Belize', 'Benin', 'Bermuda', 'Bhutan',
      'Bolivia (Plurinational State of)', 'Bonaire',
      'Bosnia and Herzegovina', 'Botswana', 'Brazil',
      'British Virgin Islands', 'Brunei Darussalam', 'Bulgaria',
      'Burkina Faso', 'Burundi', 'Cabo Verde', 'Cambodia', 'Cameroon',
      'Canada', 'Cayman Islands', 'Central African Republic', 'Chad',
      'Chile', 'China', 'Colombia', 'Comoros', 'Congo', 'Cook Islands',
      'Costa Rica', 'Côte d'Ivoire', 'Croatia', 'Cuba', 'Curaçao',
      'Cyprus', 'Czechia', 'Democratic People's Republic of Korea',
      'Democratic Republic of the Congo', 'Denmark', 'Djibouti',
      'Dominica', 'Dominican Republic', 'Ecuador', 'Egypt',
      'El Salvador', 'Equatorial Guinea', 'Eritrea', 'Estonia',
      'Eswatini', 'Ethiopia', 'Falkland Islands (Malvinas)',
      'Faroe Islands', 'Fiji', 'Finland', 'France', 'French Guiana',
      'French Polynesia', 'Gabon', 'Gambia', 'Georgia', 'Germany',
      'Ghana', 'Gibraltar', 'Greece', 'Greenland', 'Grenada',
      'Guadeloupe', 'Guam', 'Guatemala', 'Guernsey', 'Guinea',
      'Guinea-Bissau', 'Guyana', 'Haiti', 'Holy See', 'Honduras',
      'Hungary', 'Iceland', 'India', 'Indonesia',
      'Iran (Islamic Republic of)', 'Iraq', 'Ireland', 'Isle of Man',
      'Israel', 'Italy', 'Jamaica', 'Japan', 'Jersey', 'Jordan',
      'Kazakhstan', 'Kenya', 'Kiribati', 'Kosovo[1]', 'Kuwait',
      'Kyrgyzstan', 'Lao People's Democratic Republic', 'Latvia',
      'Lebanon', 'Lesotho', 'Liberia', 'Libya', 'Liechtenstein',
      'Lithuania', 'Luxembourg', 'Madagascar', 'Malawi', 'Malaysia',
      'Maldives', 'Mali', 'Malta', 'Marshall Islands', 'Martinique',
      'Mauritania', 'Mauritius', 'Mayotte', 'Mexico',
      'Micronesia (Federated States of)', 'Monaco', 'Mongolia',
      'Montenegro', 'Montserrat', 'Morocco', 'Mozambique', 'Myanmar',
      'Namibia', 'Nauru', 'Nepal', 'Netherlands', 'New Caledonia',
      'New Zealand', 'Nicaragua', 'Niger', 'Nigeria', 'Niue',
      'North Macedonia',
      'Northern Mariana Islands (Commonwealth of the)', 'Norway',
      'occupied Palestinian territory, including east Jerusalem', 'Oman',
      'Other', 'Pakistan', 'Palau', 'Panama', 'Papua New Guinea',
      'Paraguay', 'Peru', 'Philippines', 'Pitcairn Islands', 'Poland',
      'Portugal', 'Puerto Rico', 'Qatar', 'Republic of Korea',
      'Republic of Moldova', 'Réunion', 'Romania', 'Russian Federation',
      'Rwanda', 'Saba', 'Saint Barthélemy', 'Saint Helena',
      'Saint Kitts and Nevis', 'Saint Lucia', 'Saint Martin',
      'Saint Pierre and Miquelon', 'Saint Vincent and the Grenadines',
      'Samoa', 'San Marino', 'Sao Tome and Principe', 'Saudi Arabia',
      'Senegal', 'Serbia', 'Seychelles', 'Sierra Leone', 'Singapore',
      'Sint Eustatius', 'Sint Maarten', 'Slovakia', 'Slovenia',
      'Solomon Islands', 'Somalia', 'South Africa', 'South Sudan',
      'Spain', 'Sri Lanka', 'Sudan', 'Suriname', 'Sweden', 'Switzerland',
      'Syrian Arab Republic', 'Tajikistan', 'Thailand',
      'The United Kingdom', 'Timor-Leste', 'Togo', 'Tokelau', 'Tonga',
      'Trinidad and Tobago', 'Tunisia', 'Turkey', 'Turkmenistan',
      'Turks and Caicos Islands', 'Tuvalu', 'Uganda', 'Ukraine',
      'United Arab Emirates', 'United Republic of Tanzania',
```

```
'United States of America', 'United States Virgin Islands',
'Uruguay', 'Uzbekistan', 'Vanuatu',
'Venezuela (Bolivarian Republic of)', 'Viet Nam',
'Wallis and Futuna', 'Yemen', 'Zambia', 'Zimbabwe'], dtype=object)
```

In [19]:

```
DataF["Country_code"].unique()
```

Out[19]:

```
array(['AF', 'AL', 'DZ', 'AS', 'AD', 'AO', 'AI', 'AG', 'AR', 'AM', 'AW',
'AU', 'AT', 'AZ', 'BS', 'BH', 'BD', 'BB', 'BY', 'BE', 'BZ', 'BJ',
'BM', 'BT', 'BO', 'XA', 'BA', 'BW', 'BR', 'VG', 'BN', 'BG', 'BF',
'BI', 'CV', 'KH', 'CM', 'CA', 'KY', 'CF', 'TD', 'CL', 'CN', 'CO',
'KM', 'CG', 'CK', 'CR', 'CI', 'HR', 'CU', 'CW', 'CY', 'CZ', 'KP',
'CD', 'DK', 'DJ', 'DM', 'DO', 'EC', 'EG', 'SV', 'GQ', 'ER', 'EE',
'SZ', 'ET', 'FK', 'FO', 'FJ', 'FI', 'FR', 'GF', 'PF', 'GA', 'GM',
'GE', 'DE', 'GH', 'GI', 'GR', 'GL', 'GD', 'GP', 'GU', 'GT', 'GG',
'GN', 'GW', 'GY', 'HT', 'VA', 'HN', 'HU', 'IS', 'IN', 'ID', 'IR',
'IQ', 'IE', 'IM', 'IL', 'IT', 'JM', 'JP', 'JE', 'JO', 'KZ', 'KE',
'KI', 'XK', 'KW', 'KG', 'LA', 'LV', 'LB', 'LS', 'LR', 'LY', 'LI',
'LT', 'LU', 'MG', 'MW', 'MY', 'MV', 'ML', 'MT', 'MH', 'MQ', 'MR',
'MU', 'YT', 'MX', 'FM', 'MC', 'MN', 'ME', 'MS', 'MA', 'MZ', 'MM',
nan, 'NR', 'NP', 'NL', 'NC', 'NZ', 'NI', 'NE', 'NG', 'NU', 'MK',
'MP', 'NO', 'PS', 'OM', ' ', 'PK', 'PW', 'PA', 'PG', 'PY', 'PE',
'PH', 'PN', 'PL', 'PT', 'PR', 'QA', 'KR', 'MD', 'RE', 'RO', 'RU',
'RW', 'XC', 'BL', 'SH', 'KN', 'LC', 'MF', 'PM', 'VC', 'WS', 'SM',
'ST', 'SA', 'SN', 'RS', 'SC', 'SL', 'SG', 'XB', 'SX', 'SK', 'SI',
'SB', 'SO', 'ZA', 'SS', 'ES', 'LK', 'SD', 'SR', 'SE', 'CH', 'SY',
'TJ', 'TH', 'GB', 'TL', 'TG', 'TK', 'TO', 'TT', 'TN', 'TR', 'TM',
'TC', 'TV', 'UG', 'UA', 'AE', 'TZ', 'US', 'VI', 'UY', 'UZ', 'VU',
'VE', 'VN', 'WF', 'YE', 'ZM', 'ZW'], dtype=object)
```

In [20]:

```
DataF.columns = [col.strip() for col in DataF.columns]
DataF.columns
```

Out[20]:

```
Index(['Date_reported', 'Country_code', 'Country', 'WHO_region', 'New_cases',
'Cumulative_cases', 'New_deaths', 'Cumulative_deaths'],
dtype='object')
```

In [21]:

```
DataF.Country
```

Out[21]:

```
0      Afghanistan
1      Afghanistan
2      Afghanistan
3      Afghanistan
4      Afghanistan
...
150964  Zimbabwe
150965  Zimbabwe
150966  Zimbabwe
150967  Zimbabwe
150968  Zimbabwe
Name: Country, Length: 150969, dtype: object
```

In [22]:

```
DataF.loc[1:4, "Country"]
```

Out[22]:

```
1      Afghanistan
2      Afghanistan
3      Afghanistan
4      Afghanistan
Name: Country, dtype: object
```

In [23]:

```
DataF.loc[1:8, ["Country", "New_cases"]]
```

Out[23]:

	Country	New_cases
1	Afghanistan	0
2	Afghanistan	0
3	Afghanistan	0
4	Afghanistan	0
5	Afghanistan	0
6	Afghanistan	0
7	Afghanistan	0
8	Afghanistan	0

In [24]:

```
DataF.Country == "India"
```

Out[24]:

```
0      False
1      False
2      False
3      False
4      False
...
150964 False
150965 False
150966 False
150967 False
150968 False
Name: Country, Length: 150969, dtype: bool
```

In [25]:

```
DataF[DataF.Country == "India"]
```

Out[25]:

	Date_reported	Country_code	Country	WHO_region	New_cases	Cumulative_cases	New
61152	2020-01-03	IN	India	SEARO	0	0	
61153	2020-01-04	IN	India	SEARO	0	0	
61154	2020-01-05	IN	India	SEARO	0	0	
61155	2020-01-06	IN	India	SEARO	0	0	
61156	2020-01-07	IN	India	SEARO	0	0	
...
61784	2021-09-26	IN	India	SEARO	28326	33652745	
61785	2021-09-27	IN	India	SEARO	26041	33678786	
61786	2021-09-28	IN	India	SEARO	18795	33697581	
61787	2021-09-29	IN	India	SEARO	18870	33716451	
61788	2021-09-30	IN	India	SEARO	23529	33739980	

637 rows × 8 columns



In [26]:

DataF[DataF.New_deaths > 1000]

Out[26]:

	Date_reported	Country_code	Country	WHO_region	New_cases	Cumulative_cases
5370	2020-10-03	AR	Argentina	AMRO	14001	765002
15537	2020-09-08	BO	Bolivia (Plurinational State of)	AMRO	528	120769
17975	2020-05-21	BR	Brazil	AMRO	17408	271628
17977	2020-05-23	BR	Brazil	AMRO	18508	310087
17978	2020-05-24	BR	Brazil	AMRO	20803	330890
...
144593	2021-09-25	US	United States of America	AMRO	122403	42434301
144594	2021-09-26	US	United States of America	AMRO	134541	42568842
144595	2021-09-27	US	United States of America	AMRO	116041	42684883
144597	2021-09-29	US	United States of America	AMRO	94879	42859539
144598	2021-09-30	US	United States of America	AMRO	107399	42966938

756 rows × 8 columns



In [27]:

```
DataF.loc[(DataF.New_deaths > 1000) & (DataF.Country_code=="IN"),["Date_reported","Country_
```

Out[27]:

	Date_reported	Country_code	Country	New_deaths	New_cases
61318	2020-06-17	IN	India	2003	10974
61354	2020-07-23	IN	India	1129	45720
61372	2020-08-10	IN	India	1007	62064
61376	2020-08-14	IN	India	1007	64553
61381	2020-08-19	IN	India	1092	64531
...
61693	2021-06-27	IN	India	1258	50040
61697	2021-07-01	IN	India	1005	48786
61706	2021-07-10	IN	India	1206	42766
61709	2021-07-13	IN	India	2020	32906
61717	2021-07-21	IN	India	3998	42015

120 rows × 5 columns

In [28]:

```
DataF.loc[DataF.Country_code == "IN",["New_cases"]].max()
```

Out[28]:

```
New_cases    414188
dtype: int64
```

In [29]:

```
DataF.loc[DataF.Country_code == "IN",["New_deaths"]].max()
```

Out[29]:

```
New_deaths    6148
dtype: int64
```

In [30]:

```
DataF.loc[DataF.Country_code == "IN",["New_deaths"]].sum()
```

Out[30]:

```
New_deaths    448062
dtype: int64
```

In [31]:

```
#DataC = pd.read_csv("c:\\Users\\yurik\\data\\covid\\WHO-COVID-19-Data.csv")
```

In [32]:

```
#DataC = pd.DataFrame(DataC)
```

In [33]:

```
# Ploting Data
#DataCountry = DataC["Country"]
#DataNCases = DataC["New_cases"]
#DataCCases = DataC["Cumulative_cases"]
#DataNDeaths = DataC["New_deaths"]
#DataCDeaths = DataC["Cumulative_deaths"]
```