# COVID-19 Cases Analysis

In [1]:

```
# Reading the Excel file into a Pandas DataFrame
import pandas as pd
file_path = r'C:\Users\sankar\Desktop\Covid_19_cases4.xlsx'

# Load the Excel file into a Pandas DataFrame
data = pd.read_excel(file_path)
print(data)
```

```
        dateRep  day  month  year  cases  deaths countriesAndTerritories
0    2021-05-31   31      5  2021    366       5                 Austria
1    2021-05-30   30      5  2021    570       6                 Austria
2    2021-05-29   29      5  2021    538      11                 Austria
3    2021-05-28   28      5  2021    639       4                 Austria
4    2021-05-27   27      5  2021    405      19                 Austria
...         ...  ...    ...   ...    ...     ...                     ...
2725 2021-03-06    6      3  2021   3455      17                  Sweden
2726 2021-03-05    5      3  2021   4069      12                  Sweden
2727 2021-03-04    4      3  2021   4884      14                  Sweden
2728 2021-03-03    3      3  2021   4876      19                  Sweden
2729 2021-03-02    2      3  2021   6191      19                  Sweden
[2730 rows x 7 columns]
```

In [2]:

```
# Creating copy of original data
cdata=data.copy()
```

In [3]:

```
# Structure of the dataset
cdata.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2730 entries, 0 to 2729
Data columns (total 7 columns):
 #   Column                  Non-Null Count  Dtype
---  ------                  --------------  -----
 0   dateRep                 2730 non-null   datetime64[ns]
 1   day                     2730 non-null   int64
 2   month                   2730 non-null   int64
 3   year                    2730 non-null   int64
 4   cases                   2730 non-null   int64
 5   deaths                  2730 non-null   int64
 6   countriesAndTerritories  2730 non-null   object
dtypes: datetime64[ns](1), int64(5), object(1)
memory usage: 149.4+ KB
```

In [4]:

```
# Summary of numerical variables
summary_num = cdata.describe()
print(summary_num)
```

```
           day    month     year     cases   deaths
count 2730.000 2730.000 2730.000  2730.000 2730.000
mean    16.000    4.011 2021.000  3661.011   65.292
std      8.766    0.819    0.000  6490.510  113.957
min      1.000    3.000 2021.000 -2001.000   -3.000
25%      8.000    3.000 2021.000   361.250    2.000
50%     16.000    4.000 2021.000   926.500   14.500
75%     24.000    5.000 2021.000  3916.250   72.000
max     31.000    5.000 2021.000 53843.000  956.000
```

In [5]:

```
#Summary of categorical variables
summary_cate = cdata.describe(include = "O")
print(summary_cate)
```

```
       countriesAndTerritories
count                     2730
unique                      30
top                    Austria
freq                        91
```

In [6]:

```
# Removing duplicate records
cdata.drop_duplicates(keep='first',inplace=True)
```

In [7]:

```
# Check for missing values
cdata.isnull()
print('Data columns with null values:\n', cdata.isnull().sum())
```

```
Data columns with null values:
 dateRep                   0
day                        0
month                      0
year                       0
cases                      0
deaths                     0
countriesAndTerritories    0
dtype: int64
```

In [8]:

```
# Calculate Mean Daily Cases
mean_daily_cases = cdata['cases'].mean()
print("Mean Daily Cases:", mean_daily_cases)

# Calculate Mean Daily Deaths
mean_daily_deaths = cdata['deaths'].mean()
print("Mean Daily Deaths:", mean_daily_deaths)

# Calculate Standard Deviation of Daily Cases
std_daily_cases = cdata['cases'].std()
print("Standard Deviation of Daily Cases:", std_daily_cases)

# Calculate Standard Deviation of Daily Deaths
std_daily_deaths = cdata['deaths'].std()
print("Standard Deviation of Daily Deaths:", std_daily_deaths)
```

```
Mean Daily Cases: 3661.010989010989
Mean Daily Deaths: 65.29194139194139
Standard Deviation of Daily Cases: 6490.510073102111
Standard Deviation of Daily Deaths: 113.95663405806982
```