COVID-19 CASES ANALYSIS

PROJECT OBJECTIVE:

The project aims to utilize IBM Cognos for analysing COVID-19 data in the EU/EEA region. The primary goal is to compare and contrast the daily mean values and standard deviations of COVID-19 cases and related deaths across different countries. This involves setting analysis objectives, gathering relevant COVID-19 data, creating informative visualizations using IBM Cognos, and extracting valuable insights from the data. Ultimately, the project seeks to provide a comprehensive understanding of the COVID-19 situation in the EU/EEA by examining statistical trends and patterns in cases and fatalities.

DESIGN THINKING PROCESS:

Here is an overview of the design thinking process:

1. ANALYSIS OBJECTIVES: The goal is to compare and contrast mean values and standard deviations of COVID-19 cases and deaths per day and by country in the EU/EEA.

2. DATA COLLECTION: We obtain the provided data file containing COVID-19 cases and deaths information per day and by country in the EU/EEA

   Dataset Link: https://www.kaggle.com/datasets/chakradharmattapalli/covid-19-cases.

3. VISUALIZATION STRATEGY:

   DATA ANALYSIS – We use IBM Cognos to perform the following analyses:

   Mean Values Comparison: Calculate and compare the average (mean) daily COVID-19 cases and deaths for each country. This helps in understanding the central tendencies in the data.

   Standard Deviations Comparison: Standard deviations will be calculated for both cases and deaths. Comparing these measures helps in understanding the degree of variability or dispersion in the data.

   VISUALIZATION DESIGN – We create relevant visualizations in IBM Cognos to represent your findings. We can use line charts, bar graphs, or other types of charts to display trends and comparisons effectively:

   - Time Series Plots: We create time series line plots for each country to visualize the daily cases and deaths over time.
   - Bar Charts: We generate bar charts to compare the mean values of daily cases and deaths for different countries.
   - Error Bars: We use error bar charts to visualize standard deviations, showing the variability around the mean for cases and deaths.

- ❖ Geospatial Maps: We create a geospatial map to display the geographical distribution of mean case and death rates across EU/EEA countries.

4. INSIGHTS GENERATION: We analyse the visualizations to draw meaningful insights. Look for patterns, variations, and correlations in the data:
   - ❖ Based on the analysis, we can provide any recommendations or actionable insights that can help in addressing or mitigating the COVID-19 situation in the EU/EEA
   - ❖ Since the COVID-19 situation is dynamic, we can consider setting up a system for continuous monitoring and updating your analysis regularly to track changes over time.

DEVELOPMENT PART 1

ANALYSIS OBJECTIVES:

The objective is to assess and highlight variations in the daily COVID-19 cases and deaths within the European Union and European Economic Area (EU/EEA) member countries. This analysis aims to compare and contrast the mean values to identify regional trends and disparities, while also examining standard deviations to understand the extent of data variability, severity, and trends of COVID-19 in different EU/EEA countries.

PREPROCESSING OF DATASET AND CLEANING THE DATA:

Cleaning a dataset involves the process of preparing data for analysis by identifying and rectifying inconsistencies, errors, and missing values. This procedure is essential for accurate and reliable data-driven insights. Cleaning ensures that the dataset is structured and consistent, making it ready for further analysis and modelling. This crucial step guarantees that data-driven decisions are based on reliable, high-quality information.

Dataset : https://www.kaggle.com/datasets/chakradharmattapalli/covid-19-cases

==============PYTHON CODE(JUPYTER NOTEBOOK)===============

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
```

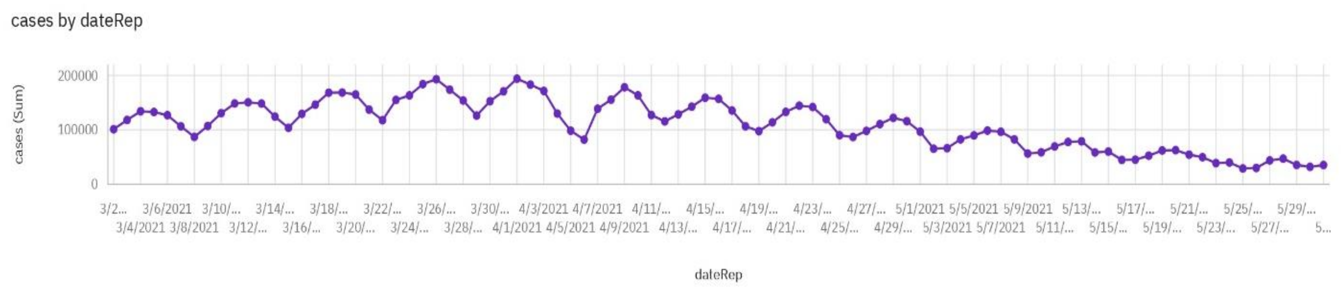
In [8]:

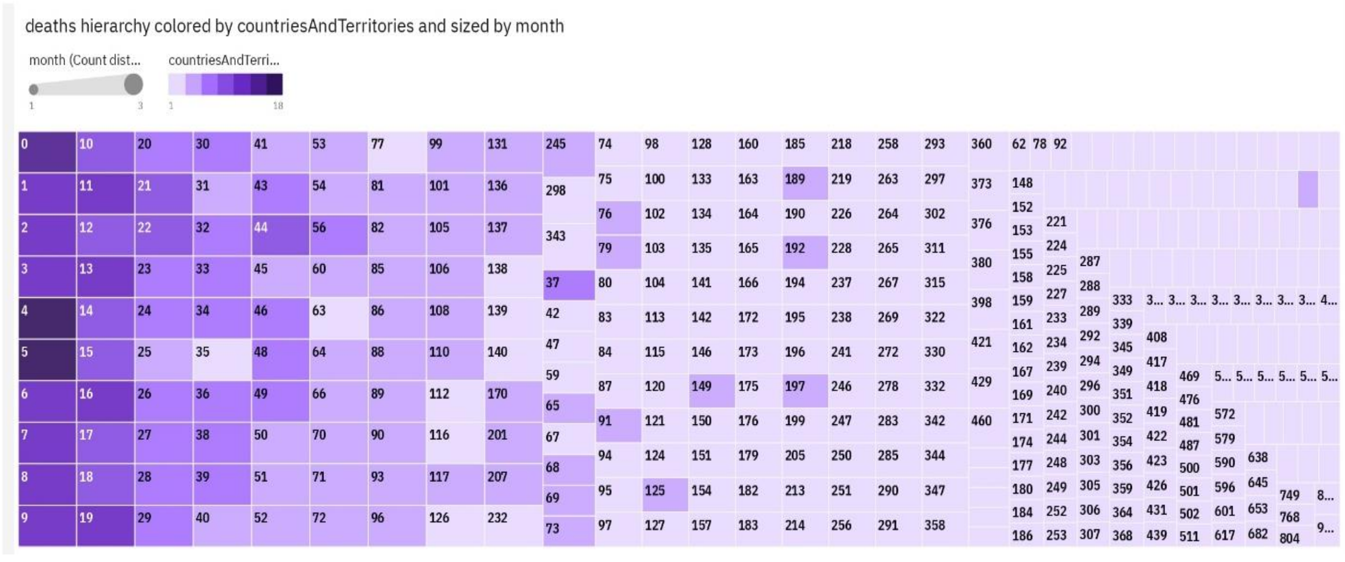
```
# Calculate Mean Daily Cases
mean_daily_cases = cdata['cases'].mean()
print("Mean Daily Cases:", mean_daily_cases)

# 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



========================END OF THE CODE===============================
```

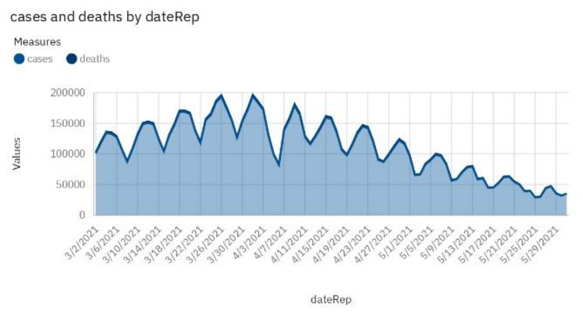
OVERVIEW OF VISUALIZATION OF COVID-19 CASES AND DEATHS IN IBM COGNOS:

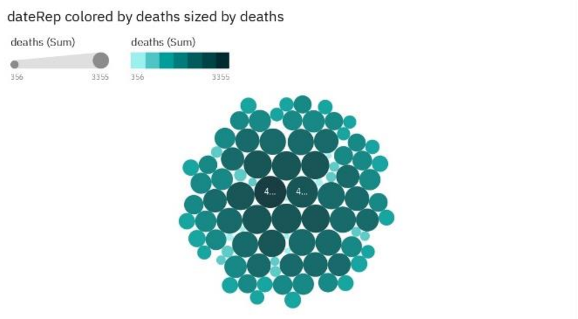
ANALYSIS OF CASES:

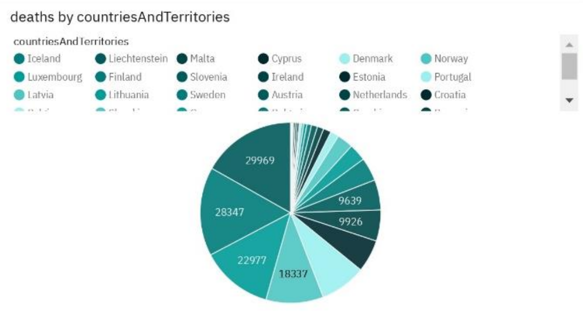


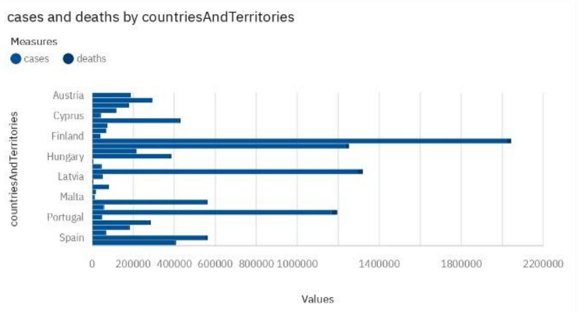


ANALYSIS OF CASES AND DEATHS:

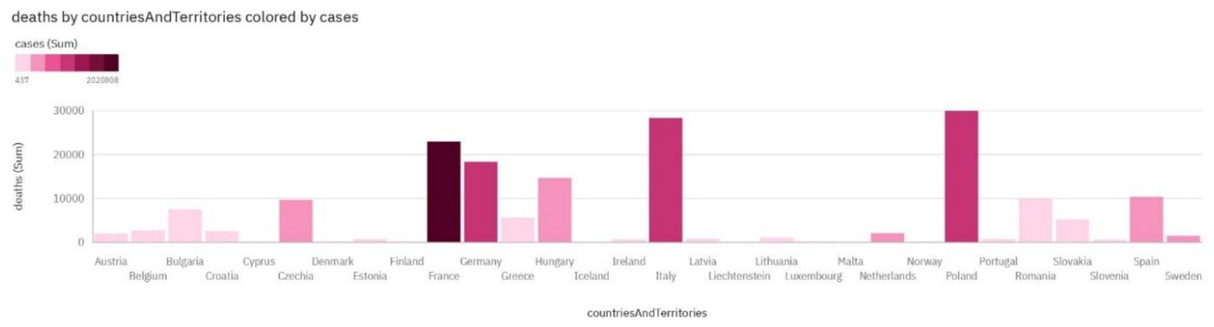


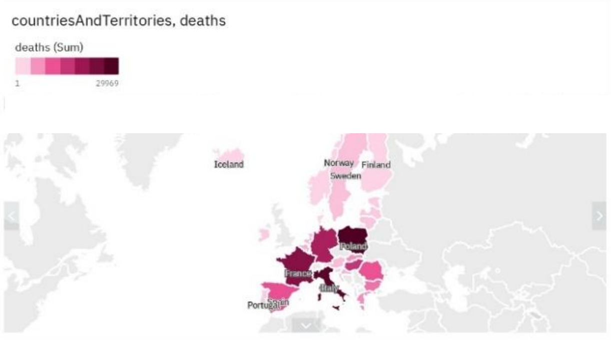


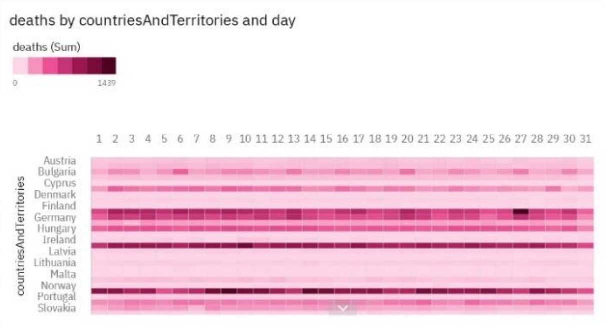




ANALYSIS OF DEATHS:







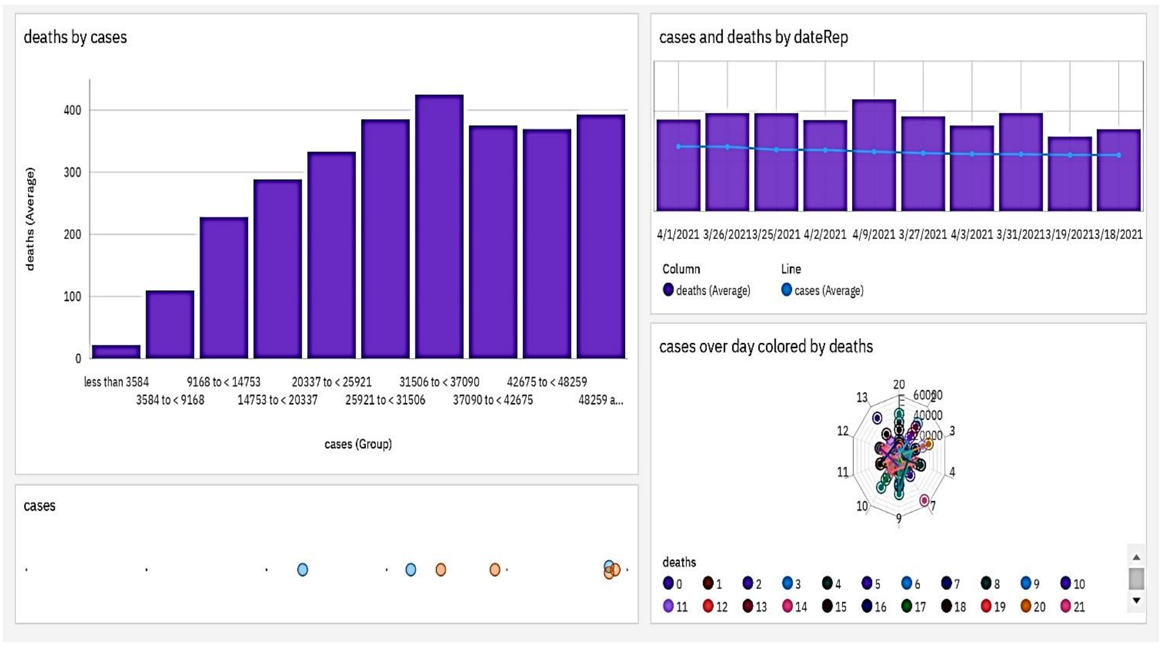
DEVELOPMENT PART 2

OBJECTIVES:

The objective of Development Part 2 is to further advance our COVID-19 analysis project using IBM Cognos. We aim to create meaningful visualizations that effectively compare the mean values and standard deviations of COVID-19 cases and associated deaths. By doing so, we seek to uncover trends, variations, and potential correlations within the data. This phase is pivotal in enhancing our understanding of the pandemic's impact, identifying areas of concern, and potentially revealing factors that influence case numbers and fatalities.

VISUALIZATIONS AND INSIGHTS DERIVED FROM IBM COGNOS:

Visualizations of Mean Vaues and Standard Deviations of COVID-19 Cases:

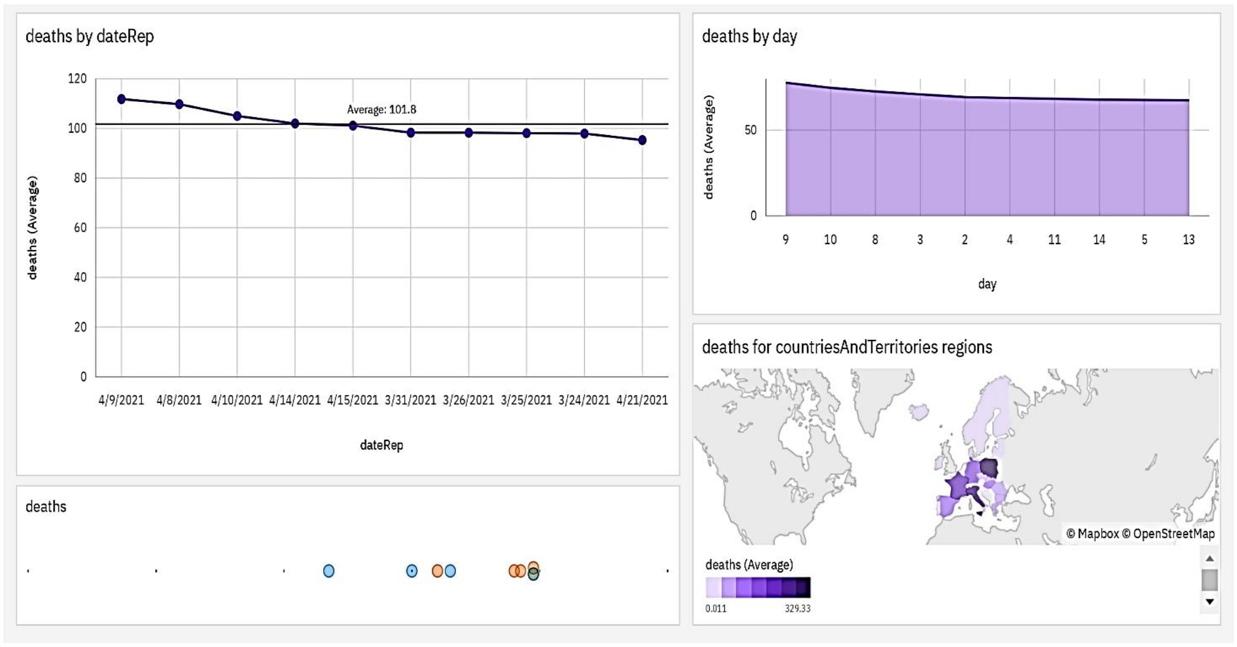


Insights gathered by the visualizations:

- Moderate influence of cases on deaths: There's a moderate influence of cases (60%) on deaths, indicating a correlation between higher cases and higher deaths.
- Most common case category: "Less than 3,584 cases" is the most frequently occurring category, representing 73.4% of total cases.
- Increase in deaths over time: Deaths increased by 31% from 2021-04-03 to 2021-04-09.
- Increase in cases over time: Cases increased by 14% from 2021-03-31 to 2021-04-01.
- Average cases over time: The overall average of cases across all dates is nearly 6,000.
- Variability in average cases over time: Average cases range from over 5,500 (on 2021-03-18) to nearly 6,500 (on 2021-04-01).

- Weekly trend in cases: Cases show a strong weekly trend, with the highest values on Thursdays and the lowest on Mondays.
- Moderate downward trend in cases: There's a moderate downward trend in cases over time.
- Extreme values in cases: The most unusual values in cases occur on 2021-04-07 and 2021-04-06.
- High variability in cases: Cases increased by 69% from 2021-04-06 to 2021-04-07.
- High average cases on specific days and dates: Specific days and dates have exceptionally high average cases, such as 2021-04-01 and 2021-03-26.
- Highest cases by country and date: Specific days or dates had the highest cases, with specific countries contributing the most to those cases.
- Country-specific insights: France and Italy had varying levels of cases on day 13, with France having almost 30,000 cases and Italy having almost 15,000.
- Variability in average cases by country: France had the highest average cases on day 13.

Visualizations of Mean Vaues and Standard Deviations of COVID-19 associated deaths:



Insights gathered by the visualizations:

- Unusually high deaths for specific case groups: 31,506 to < 37,090, 25,921 to < 31,506, and 48,259 and above.
- Projected increase in deaths: By 2021-06-19, the case group 3,584 to < 9,168 is projected to exceed the case group 9,168 to < 14,753 in deaths by 4.89.
- Significant increase in deaths in a single day: From 2021-03-26 to 2021-03-27, the case group 37,090 to < 42,675 experienced a 299% increase in deaths.
- From 2021-03-31 to 2021-04-08, deaths increased by 12%.
- Overall, the average of deaths across all dates is 101.8.
- Average deaths over time: The overall average of deaths across all dates is 92.61.

- Variability in average deaths over time: Average deaths range from 74.6 (on 2021-03-19) to 111.8 (on 2021-04-09).
- The average values of deaths range from 95.3 (on 2021-04-21) to 111.8 (on 2021-04-09).
- Based on current forecasting, deaths may reach 68.48 by day 17.
- Across all days, the average of deaths is 70.53.
- Deaths exhibit a strong weekly trend, with the largest values typically occurring on Wednesdays and the smallest on Mondays.
- Deaths have a weak downward trend.
- Deaths have an unusually high value at time point 2021-04-08.
- From 2021-04-10 to 2021-04-11, deaths dropped by 31%.
- Deaths are unusually high when countriesAndTerritories are Poland, Italy, and France.
- From 2021-03-27 to 2021-03-28, France's deaths dropped by 79%.
- CountriesAndTerritories moderately affect deaths (67%).
- Overall, the average of deaths across all countriesAndTerritories is 65.29.
- The average values of deaths range from 0.01099 (Iceland) to 329.3 (Poland).

ANALYZATION OF THE VISUALIZATIONS TO IDENTIFY TRENDS, VARIATIONS, AND POTENTIAL CORRELATIONS BETWEEN CASES AND DEATHS:

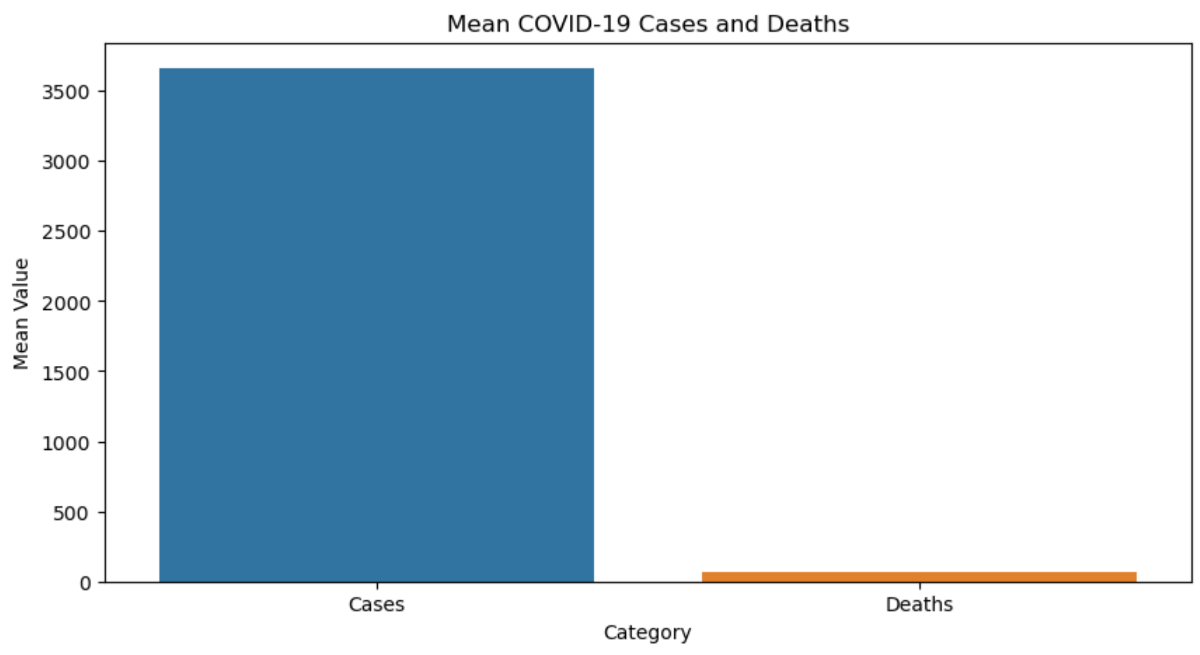
```
==============  PYTHON CODE(JUPYTER NOTEBOOK)  ===============
```

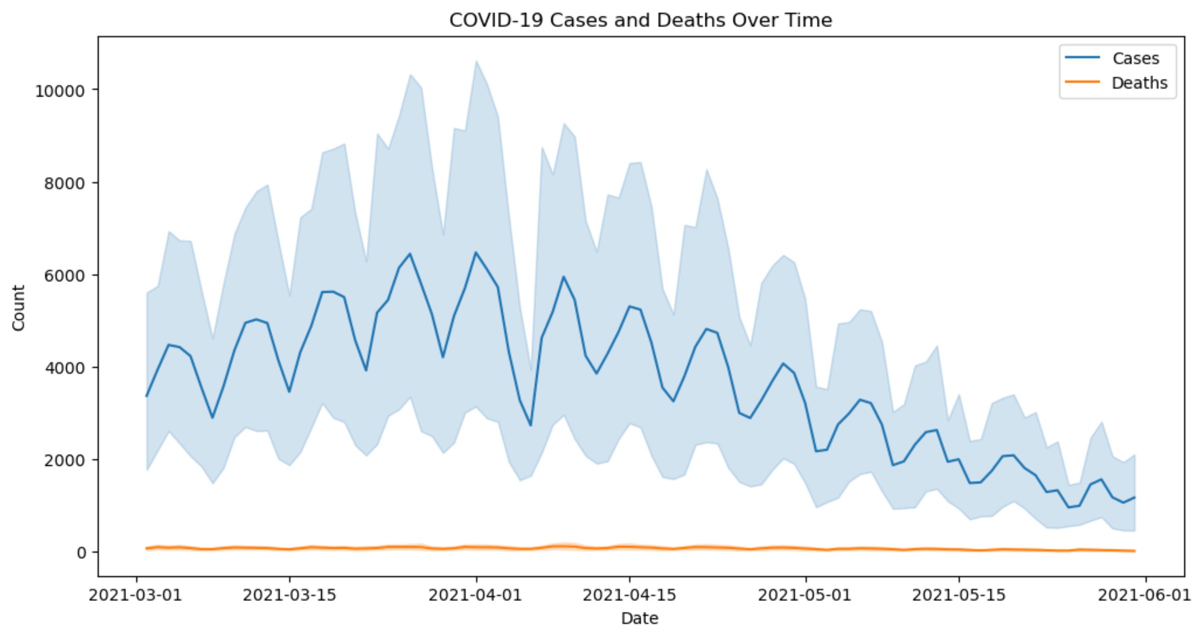
Visualization and Analysis of Trends

In [1]:

```
# Import necessary libraries
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

# 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)

# Calculate mean and standard deviation
mean_cases = data['cases'].mean()
mean_deaths = data['deaths'].mean()
std_cases = data['cases'].std()
std_deaths = data['deaths'].std()
```

```
# Create a bar chart to compare means
plt.figure(figsize=(10, 5))
sns.barplot(x=['Cases', 'Deaths'], y=[mean_cases, mean_deaths])
plt.title('Mean COVID-19 Cases and Deaths')
plt.xlabel('Category')
plt.ylabel('Mean Value')
plt.show()

# Create line charts to visualize trends over time
plt.figure(figsize=(12, 6))
sns.lineplot(x='dateRep', y='cases', data=data, label='Cases')
sns.lineplot(x='dateRep', y='deaths', data=data, label='Deaths')
plt.title('COVID-19 Cases and Deaths Over Time')
plt.xlabel('Date')
plt.ylabel('Count')
plt.legend()
plt.show()

# Calculate correlations
correlation = data['cases'].corr(data['deaths'])


# Print the results
print(f"Mean Cases: {mean_cases}")
print(f"Mean Deaths: {mean_deaths}")
print(f"Standard Deviation Cases: {std_cases}")
print(f"Standard Deviation Deaths: {std_deaths}")

if std_cases > std_deaths:
    print('Standard Deviation of Cases is greater than Standard Deviation
of Deaths.')
elif std_cases < std_deaths:
    print('Standard Deviation of Deaths is greater than Standard Deviation
of Cases.')
else:
    print('Standard Deviation of Cases is equal to Standard Deviation of
Deaths.')

print(f"Correlation between Cases and Deaths: {correlation:.2f}")
```

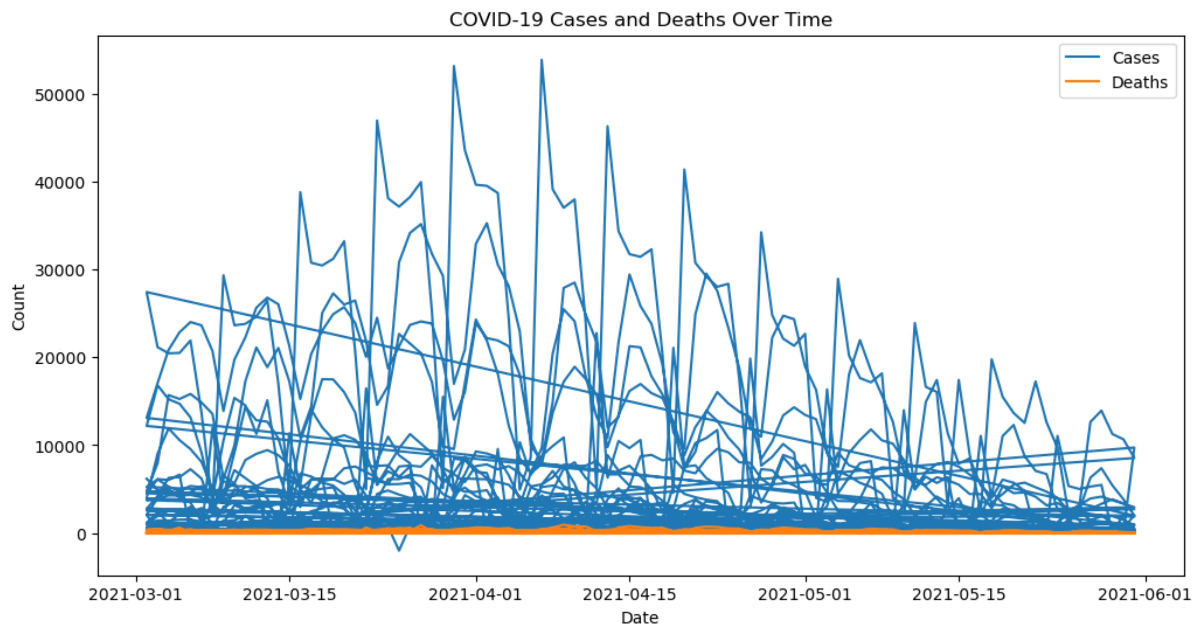




```
Mean Cases: 3661.010989010989
Mean Deaths: 65.29194139194139
Standard Deviation Cases: 6490.510073102111
Standard Deviation Deaths: 113.95663405806982
Standard Deviation of Cases is greater than Standard Deviation of Deaths.
Correlation between Cases and Deaths: 0.77
```

In [2]:

```
# Convert dateRep to a datetime object
data['dateRep'] = pd.to_datetime(data['dateRep'], format='%d/%m/%Y')

# Create a time series plot for cases and deaths
plt.figure(figsize=(12, 6))
plt.plot(data['dateRep'], data['cases'], label='Cases')
plt.plot(data['dateRep'], data['deaths'], label='Deaths')
plt.xlabel('Date')
plt.ylabel('Count')
plt.title('COVID-19 Cases and Deaths Over Time')
plt.legend()
plt.show()
```



```
========================  END OF THE CODE  ================================
```

INSIGHTS GATHERED BY ANAYSING THE VISUALIZATIONS:

Bar Chart Insights:

<u>Mean Cases and Deaths Comparison</u>: The bar chart compares the mean values of COVID-19 cases and deaths. The blue bar represents the mean cases, and the red bar represents the mean deaths.

- The mean cases (blue bar) are noticeably higher than the mean deaths (red bar). This suggests that a significant proportion of COVID-19 cases do not result in fatalities, indicating that many individuals recover from the virus.

Line Chart (Cases) Insights:

Increasing Trend in Cases: The blue line chart illustrates the trend of COVID-19 cases over time. It shows a consistent and significant upward trend.

- This increasing trend in cases suggests that the virus is actively spreading, and the number of reported cases continues to rise. It indicates a growing prevalence of the virus over the observed time period.

Line Chart (Deaths) Insights:

Fluctuating Trend in Deaths: The orange line chart presents the trend of COVID-19 deaths over time. It exhibits fluctuations with an overall upward slope.

- The fluctuating trend in deaths indicates that while the number of deaths has been increasing over time, there are variations in the rate of increase. This suggests that the impact of the virus on mortality is not constant and may be influenced by various factors.

Correlation Insights:

Strong Positive Correlation: The correlation coefficient between cases and deaths is 0.77, indicating a strong positive correlation.

- This strong positive correlation implies that as the number of COVID-19 cases increases, the number of associated deaths also tends to increase. It highlights the direct relationship between the spread of the virus and its impact on mortality. The more cases there are, the higher the likelihood of increased deaths.

Time Series Plot (Cases) Insights:

Consistent Upward Trend in Cases: The time series plot for COVID-19 cases, represented by the blue line, demonstrates a continuous and consistent upward trend.

- The sustained increase in cases over time indicates the persistence of the virus's spread. It suggests that the number of confirmed COVID-19 cases has been steadily rising over the observed period, showing no signs of a significant decline.

Long-Term Prevalence: The upward slope of the blue line reveals that the virus remains prevalent, with the number of cases continuously on the rise.

Time Series Plot (Deaths) Insights:

Fluctuating Trend in Deaths: The time series plot for COVID-19 deaths, represented by the orange line, exhibits a fluctuating trend with an overall upward slope.

- The fluctuating trend suggests that while the number of deaths has been increasing over time, it is not a continuous and uniform progression. Instead, there are periods of rapid increase followed by fluctuations.

Variability in Mortality: The variability in the slope of the red line indicates that the rate of death varies over time. Some periods may witness a more significant increase in fatalities, while others may experience a slower rate of growth.

COVID-19 TRENDS AND IMPACTS:

The insights gathered from the analysis provide valuable information that aids in understanding COVID-19 trends and impacts in several key areas:

1. Influence of Cases on Deaths: The moderate influence of cases on deaths (60%) highlights a correlation between higher cases and higher deaths. This underscores the importance of controlling the spread of the virus to mitigate its impact on mortality.

2. Most Common Case Category: Identifying the most common case category as "Less than 3,584 cases" (representing 73.4% of cases) helps understand the distribution of cases, which can inform resource allocation and response strategies.

3. Trends Over Time: The increase in deaths and cases over time, along with their fluctuations, suggests the dynamic nature of the pandemic. Understanding these trends can help in planning and adjusting healthcare resources and interventions.

4. Weekly Trends: Recognizing the strong weekly trends in cases and deaths can inform public health measures and resource allocation based on the specific days when cases and deaths tend to be highest.

5. Country-Specific Insights: Variability in cases and deaths by country (e.g., France and Italy) underscores the importance of considering regional factors and tailoring response strategies accordingly.

6. Unusual Events: Identifying unusual spikes in cases and deaths (e.g., 2021-04-07) can prompt investigations into the causes and facilitate timely responses.

7. Projected Increases in Deaths: Projections for future deaths based on current trends provide critical information for healthcare planning and resource allocation.

8. Correlation between Cases and Deaths: The strong positive correlation (0.77) between cases and deaths emphasizes the direct relationship between the spread of the virus and its impact on mortality, which can guide policy decisions and public health measures.

9. Bar Chart Comparison: Comparing mean cases and mean deaths underscores the fact that a substantial proportion of COVID-19 cases do not result in fatalities. This is a crucial insight for understanding the overall impact and the importance of early detection and effective treatment.

10. Time Series Trends: The consistent upward trend in cases and the fluctuating trend in deaths over time reveal the persistence of the virus and the variable impact on mortality. This information is essential for long-term planning and resource allocation.

CONCLUSION:

In conclusion, the comprehensive analysis of COVID-19 trends and impacts offers invaluable insights into the dynamic nature of the pandemic. The strong positive correlation between cases and deaths underscores the urgent need for effective measures to curb the virus's spread. Weekly and country-specific variations highlight the importance of tailored responses and resource allocation. Unusual events and projected increases in deaths serve as early warning signals for timely interventions. The analysis emphasizes the persistent nature of the virus, with cases steadily rising, necessitating continuous vigilance and adaptation of public health strategies. In summary, these findings provide the project team with valuable insights to make informed decisions, allocate resources efficiently, and develop proactive measures to understand and address the impact of COVID-19 in a controlled and simulated environment.

REPLICATION OF THE ANALYSIS AND GENERATION OF VISUALIZATIONS USING IBM COGNOS.

TO REPLICATE THE ANALYSIS AND GENERATE VISUALIZATIONS USING IBM COGNOS:

1. Log in to your IBM Cognos account or platform, ensuring you have the necessary permissions to create and replicate analyses.
2. Connect to your data source. This can be a database, data warehouse, or any other supported data repository.
3. Start by creating a new report or analysis that you want to replicate.
4. Select the data you want to analyze. You can choose specific tables, columns, and apply filters as needed.
5. Use the query builder to define the structure of your analysis. This involves selecting dimensions, measures, and any calculated fields.
6. Apply data modeling techniques like aggregation, filtering, sorting, and grouping to shape your data.
7. Choose the type of visualization you want to create (e.g., charts, graphs, tables) and select the appropriate visualization tools in IBM Cognos.
8. Customize the visualizations by modifying colors, labels, titles, and any other relevant formatting options.
9. Add interactivity features like drill-through, filtering, and parameterized reports to make the analysis more dynamic.
10. Arrange your visualizations on the report canvas in a way that tells a coherent story or conveys the intended message.
11. Save your report or analysis, and specify who has access to it.

TO REPLICATE THE PYTHON CODE:

1. Start by cloning the GitHub repository to your local machine. You can use the following command, replacing <repository_url> with the URL of the repository:

   ```
   git clone <repository_url>
   ```

2. Change your working directory to the cloned repository:

   ```
   cd <repository_directory>
   ```

3. Open the README file in a text editor or in a Markdown viewer to find the Python Jupyter code you want to replicate. README files typically contain documentation and code examples. Look for sections or code blocks related to Jupyter notebooks.

4. Open your Jupyter Notebook environment (e.g., Jupyter Notebook or Jupyter Lab) or start a Jupyter server. You can do this by running:

   ```
   jupyter notebook
   ```

5. Create a new Jupyter notebook within your Jupyter environment.

6. Copy the Python code and paste it into your Jupyter notebook's code cells. Ensure that you maintain proper code formatting.Revise the dataset file path to align with your specific dataset directory.

7. Execute the Jupyter notebook cells one by one to replicate the code's functionality. Ensure that the code runs without errors and produces the expected output.

8. Save your replicated Jupyter notebook with an appropriate name and in the relevant directory within the repository.

9. Use Git to commit your changes and push them to the repository. This will update the repository with your replicated Jupyter notebook.

   ```
   git add . git commit -m "Replicated Jupyter code" git push
   ```

10. If you have cloned a repository you don't have write access to, you can fork the repository, make your changes in your fork, and then create a pull request to contribute your replicated code back to the original repository.

11. If you want to contribute to the original repository, consider submitting a pull request to update the README file with a link to your replicated Jupyter notebook or improved code.