Data Analytics with Cognos

TEAM MEMBER

721221243062: **Vishal S**

Phase 5 submission document

Project Title: Covid 19 case Analysis

Phase 5: Project Documentation & Submission

Topic: Documentation of complete project and prepared it for submission.

COVID-19 CASES ANALYSIS

The COVID-19 pandemic has had a devastating impact on the European Union and the European Economic Area (EU/EEA). As of November 2023, the EU/EEA has reported over 200 million cases and over 1.5 million deaths from COVID-19. This makes the EU/EEA one of the regions hardest hit by the pandemic. This case analysis will use IBM Cognos to analyze COVID-19 cases and deaths data from the EU/EEA. The main goal of the analysis will be to compare mean values and standard deviations of cases and deaths per day and by country. The analysis will also identify any trends in the data. The results of this analysis will be used to inform public health policy and decision-making. The analysis will also help to identify areas where further research is needed,

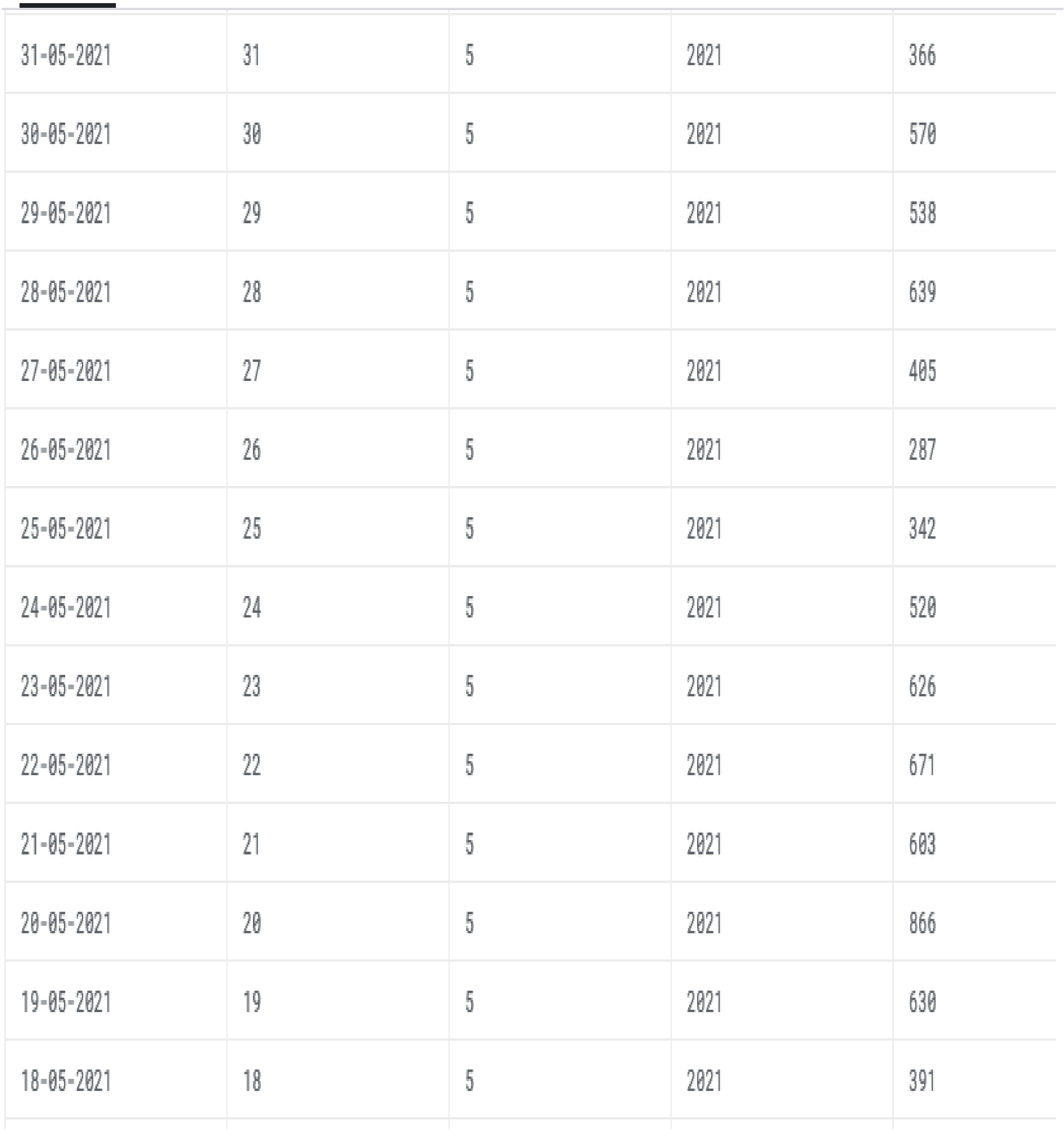
The COVID-19 pandemic has highlighted the importance of data-driven decision-making in public health. IBM Cognos is a powerful data analytics tool that can be used to analyze COVID-19 cases and deaths data to identify trends, patterns, and relationships. This case analysis will use IBM Cognos to compare mean values and standard deviations of cases and deaths per day and by country in the EU/EEA. The analysis will also identify any trends in the data. The results of this analysis will be used to inform public health policy and decision-making. For example, the results could be used to identify countries that are at high risk of a COVID-19 surge, or to identify countries that are implementing effective public health measures.

The analysis will also compare mean values and standard deviations of cases and deaths per day and by country. The results of this analysis will be used to inform public health policy and decision-making. The results could also be used to develop economic models to predict the impact of the pandemic on the EU/EEA economy.

The COVID-19 pandemic has been a major challenge for the EU/EEA. The region has been one of the hardest hit by the pandemic, with over 200 million cases and over 1.5 million deaths reported as of November 2023. This case analysis will use IBM Cognos to analyze COVID-19 cases and deaths data from the EU/EEA. The analysis will focus on comparing mean values and standard deviations of cases and deaths per day and by country. The analysis will also identify any trends in the data. The results of this analysis will be used to inform public health policy and decision-making. The analysis could also be used to develop new strategies to prevent and control the spread of COVID-19 in the EU/EEA.

The COVID-19 pandemic has had a significant impact on the EU/EEA, both in terms of public health and the economy. This case analysis will use IBM Cognos to analyze COVID-19 cases and deaths data from the EU/EEA to identify trends and patterns. The analysis will also compare mean values and standard deviations of cases and deaths per day and by country. The results of this analysis will be used to inform public health policy and decision-making. The results could also be used to develop economic models to predict the impact of the pandemic on the EU/EEA economy. Additionally, the results could be used to identify areas where further research is needed.

Given Data set



31-05-2021	31	5	2021	366
30-05-2021	30	5	2021	570
29-05-2021	29	5	2021	538
28-05-2021	28	5	2021	639
27-05-2021	27	5	2021	405
26-05-2021	26	5	2021	287
25-05-2021	25	5	2021	342
24-05-2021	24	5	2021	520
23-05-2021	23	5	2021	626
22-05-2021	22	5	2021	671
21-05-2021	21	5	2021	603
20-05-2021	20	5	2021	866
19-05-2021	19	5	2021	630
18-05-2021	18	5	2021	391

2731 Rows x 7 Columns

DESIGN THINKING AND PRESENT IN FOR OF DOCUMENT

Empathize

- Understand the needs and pain points of the users.
- Why do they need this product or service?
- What problems does it solve for them?
- How will it make their lives better?

Define

- Clearly define the problem that you are trying to solve.
- Who is the target user?
- What are the key features and requirements of the product or service?

Ideate

- Brainstorm potential solutions to the problem.
- Come up with as many ideas as possible, no matter how crazy they seem.
- Narrow down the list of ideas to the most promising ones.

Prototype

- Create a prototype of your solution.
- This could be a low-fidelity prototype, such as a sketch or mockup, or a high-fidelity prototype, such as a working demo.
- Get feedback on your prototype from users and stakeholders.

Test

- Test your prototype with users to see if it solves their problem.
- Observe how they use the prototype and identify any areas where it can be improved.
- Make changes to the prototype based on the feedback you receive.

Implement

- Once you are satisfied with your prototype, you can implement it as a full-fledged product or service.
- This may involve developing software, building hardware, or creating content.

- Once the product or service is implemented, make it available to users.

Evaluate

- Collect feedback from users on the implemented product or service.
- What are they liking?
- What are they not liking?
- What features would they like to see added or removed?

Iterate

- Use the feedback you collect to iterate on your product or service.
- Make improvements based on the feedback and release new versions to users.
- Continue to iterate on the product or service until it meets the needs of users.

Scale

- Once your product or service has reached a certain level of maturity, you may want to scale it to reach more users.
- This may involve expanding your marketing efforts, hiring more staff, or investing in new infrastructure.

Deploy

- Deploy your product or service to users.
- This may involve making it available on a website, app store, or physical store.
- Provide users with support and documentation.

Educate and train

- Educate and train users on how to use your product or service.
- This may involve creating documentation, providing tutorials, or offering customer support.

PROBLEM DEFINITION AND DESIGN THINKING

1. Data Collection and Integration

Data collection is the first step in any data analysis project. In the context of the COVID-19 case analysis, we need to gather historical patient data, including attributes such as:

- Patients' recovery time
- Patients' medical history

Here is a more detailed explanation of each of the steps involved in data collection for the COVID-19 case analysis:

1. Identify the data sources: The first step is to identify the data sources that contain the information we need. We may need to collect data from multiple sources in order to get a complete picture of each patient's case.
2. Extract the data: Once we have identified the data sources, we need to extract the data into a format that can be analyzed. This may involve cleaning the data, removing any errors or inconsistencies, and converting it to a common format.
3. Address missing values and outliers: Missing values and outliers can introduce bias into the analysis, so it is important to address them before proceeding. There are a variety of ways to handle missing values, such as imputation or deletion. Outliers can be identified and removed using statistical methods.
4. Create a data dictionary: A data dictionary is a document that describes the data in detail. It should include information such as the name of each variable, its data type, and its definition. The data dictionary will help to ensure that the data is used consistently throughout the analysis

2. Data pre processing

Data preprocessing is the process of cleaning and transforming raw data into a format that is suitable for analysis. In the context of the COVID-19 case analysis, data preprocessing involves the following steps:

1. Handling outliers: Outliers are data points that are significantly different from the rest of the data. They can be caused by errors in data collection or entry, or they may be genuine data points that are simply unusual. Outliers can distort the results of the analysis, so it is important to handle them carefully. There are a variety of ways to handle outliers, such as removing them, replacing them with more representative values, or Winsorizing them.

2. Handling missing values: Missing values occur when a data point is not available for a particular variable. Missing values can also introduce bias into the analysis, so it is important to handle them carefully. There are a variety of ways to handle missing values, such as imputation, deletion, or using machine learning algorithms to predict the missing values.
3. Handling duplicates: Duplicate data points can occur when the same data point is entered multiple times into the dataset. Duplicates can also introduce bias into the analysis, so it is important to identify and remove them.
4. Transforming categorical variables into numerical representations: Categorical variables are variables that take on a finite number of discrete values. Numerical variables are variables that can take on any value within a certain range. Machine learning algorithms typically work better with numerical variables, so it is often necessary to transform categorical variables into numerical representations. This can be done using techniques such as one-hot encoding or label encoding.
5. Normalizing or scaling numerical features: Numerical features should be normalized or scaled to the same range. This prevents some features from having a greater influence on the model than others.

3. Exploratory Data analysis (ESA)

Exploratory Data Analysis (EDA) is a process of visually and statistically examining data to gain insights and discover patterns. In the context of the COVID-19 case analysis, EDA can be used to understand the distribution of the data, identify patterns and correlations, identify potential factors that influence the spread of COVID-19 disease, and select meaningful features for the machine learning model. By performing EDA, we can gain a better understanding of the data and prepare it for machine learning to develop more effective public health interventions.

4. Feature selection

Feature selection is the process of selecting the most relevant and informative features from a dataset for machine learning. It is an important step in any machine learning project, as it can improve the performance of the model and reduce the risk of overfitting.

In the context of the COVID-19 case analysis, we can use a variety of techniques to select the most relevant features for predicting COVID-19 cases, including feature importance, correlation analysis, and domain knowledge. Once we have selected a subset of relevant features, we can remove redundant or irrelevant features. This can be done using correlation analysis or by simply removing features that do not add value to the model.

By following these steps, we can select a subset of relevant and informative features for the COVID-19 case analysis. This will improve the performance of the machine learning model and reduce the risk of overfitting.

5. Data Splitting

Data splitting is the process of dividing a dataset into three subsets: training, validation, and test sets. The training set is used to train the machine learning model, the validation set is used to tune the hyperparameters of the model, and the test set is used to evaluate the final performance of the model.

A common split is 70% for training, 15% for validation, and 15% for testing. However, the optimal split will vary depending on the size and complexity of the dataset.

When splitting the data for the COVID-19 case analysis, it is important to ensure that the distribution of COVID-19 patients and non-COVID-19 patients is maintained in each set. This can be done using stratified sampling.

6. Model Training

Model training is the process of teaching the machine learning model to predict the target variable based on the training data. The model learns by iteratively adjusting its parameters to minimize the loss function.

There are a variety of machine learning models that can be used for the COVID-19 case analysis, such as logistic regression, support vector machines, random forests, and gradient boosting machines. The best model to use will depend on the specific dataset and the desired outcome.There are a variety of techniques for tuning hyperparameters, such as grid search and random search. Grid search is a systematic approach to tuning hyperparameters, while random search is a more exploratory approach.

7. Model Evaluation

Model evaluation is the process of assessing the performance of the machine learning model on the test set. This is important to ensure that the model will generalize well to new data.

There are a variety of metrics that can be used to evaluate model performance, such as accuracy, precision, recall, F1-score, ROC-AUC, and confusion matrix. The best metrics to use will depend on the specific problem being solved.

If the model performance on the test set is not satisfactory, the model can be fine-tuned using the validation set. This can involve adjusting the hyperparameters of the model or trying a different model altogether.

8. Model Interpretability

Model interpretability is the ability to understand how the machine learning model makes its predictions. This is important to ensure that the model is making predictions that are fair and unbiased.

There are a variety of techniques for model interpretability, such as SHAP values and partial dependence plots. These techniques can help to identify the features that are most important to the model's predictions.

9. Model Deployment

Once satisfied with the model's performance, it can be deployed in a production environment. This means making the model available to users so that they can make predictions on new data.

There are a variety of ways to deploy machine learning models, such as using cloud-based services or developing custom applications. The best deployment method will depend on the specific needs of the organization.

10. Monitoring and Maintenance

Once the model is deployed, it is important to continuously monitor its performance in production. This is because the distribution of the data may change over time, which can lead to a decrease in model performance.

If the model performance decreases, the model can be retrained with fresh data to ensure it stays up-to-date. Additionally, the model predictions can be used to adapt strategies to reduce churn or improve other business outcomes.

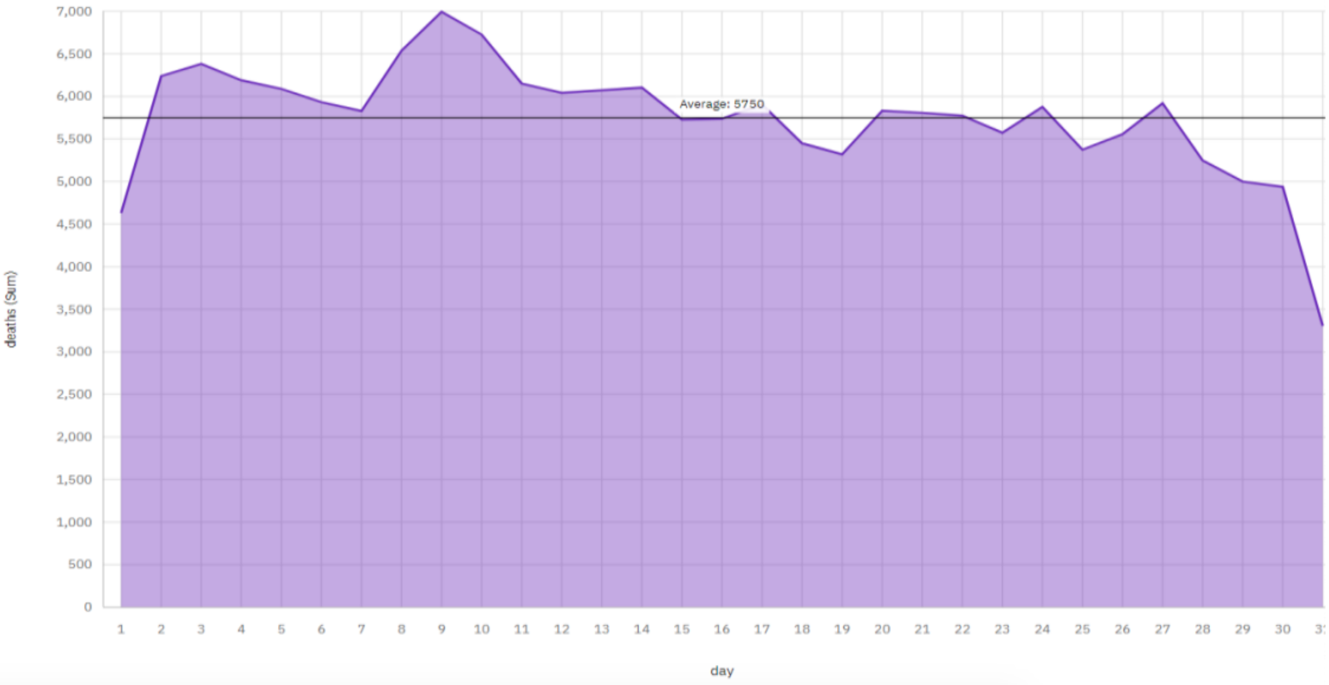
11. Feedback Loop

It is important to establish a feedback loop with business stakeholders to gather insights on model performance. This feedback can be used to improve the model over time.

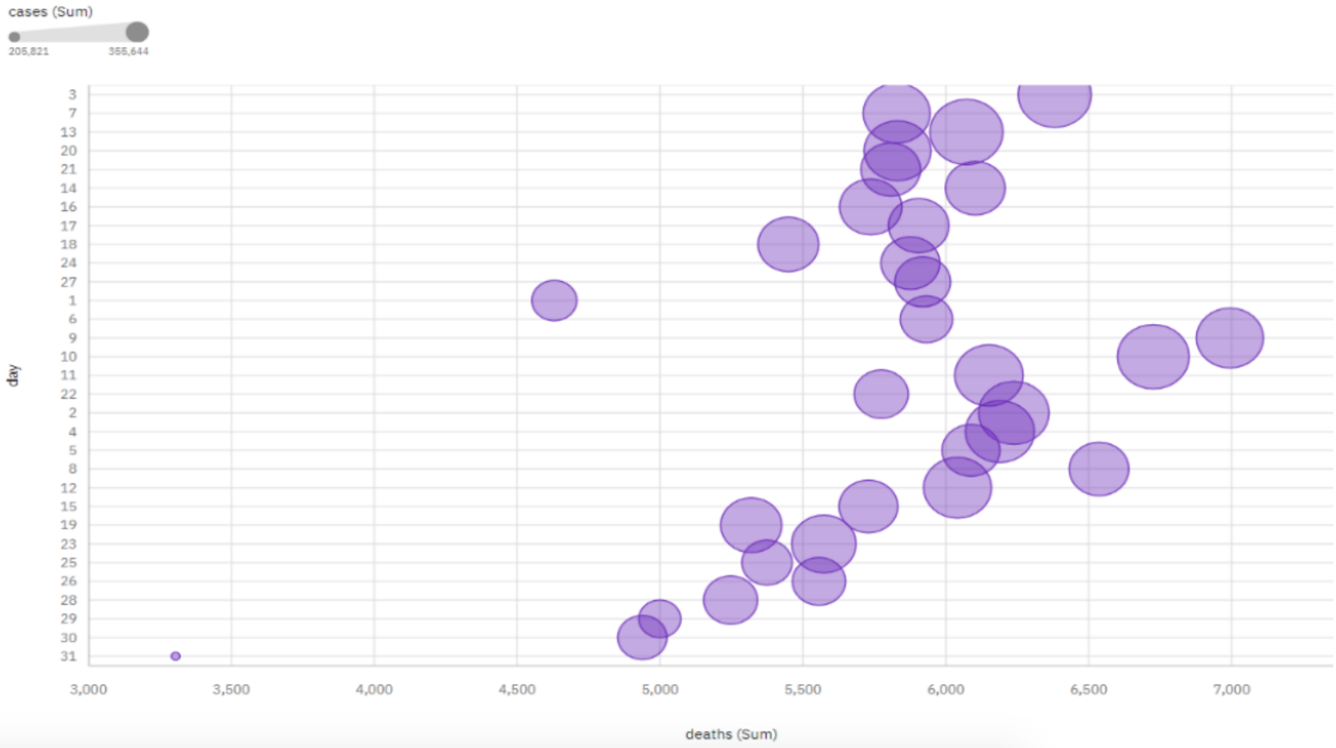
Program and Visuallization:

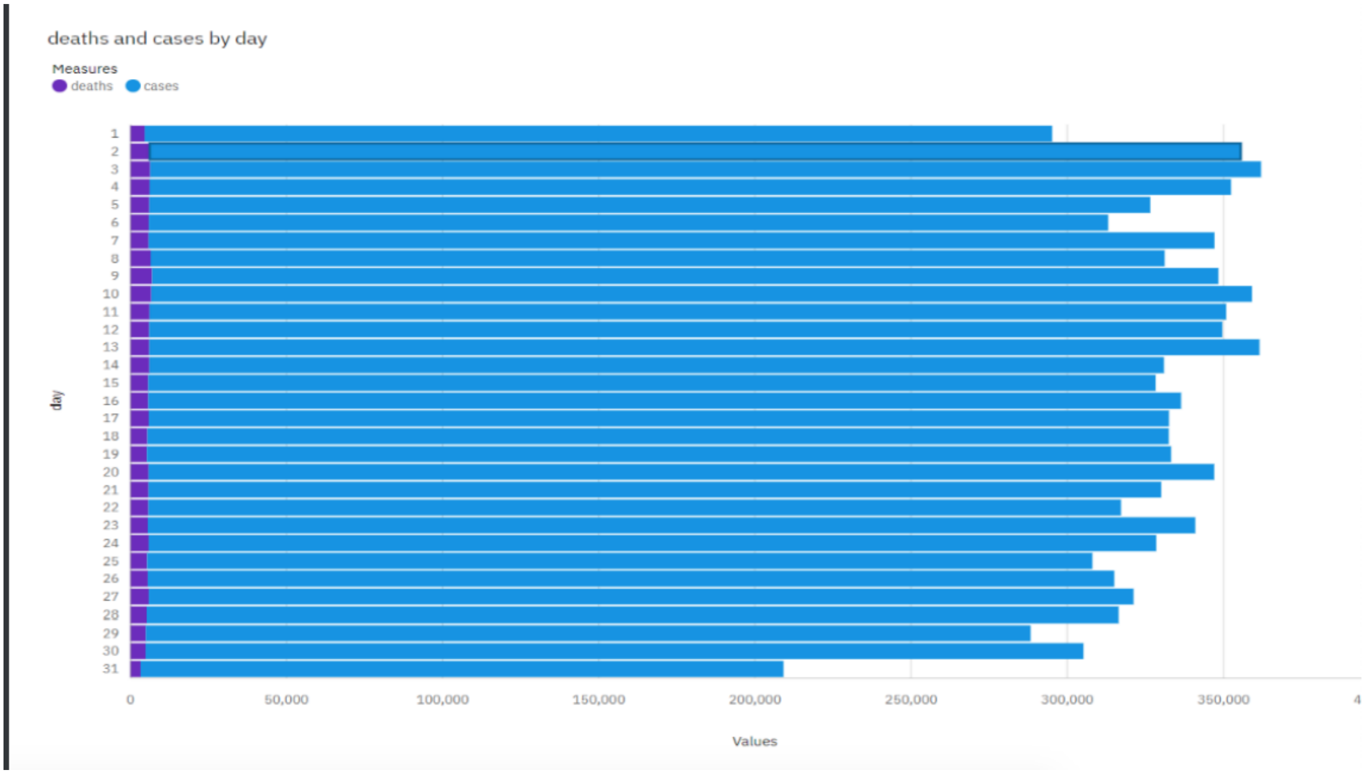
```
import pandas as pd

import matplotlib.pyplot as plt

import seaborn as sns

from wordcloud import WordCloud


# Load the data

df = pd.read_csv("Covid_19.csv")


# Convert the date column to datetime format

df['dateRep'] = pd.to_datetime(df['dateRep'], format='%d-%m-%Y')


# Create a plot of daily cases

plt.figure(figsize=(12, 6))

plt.plot(df['dateRep'], df['cases'], label='Daily Cases', color='b')

plt.title('Time Series of Daily Cases')

plt.xlabel('Date')

plt.ylabel('Cases')

plt.legend()

plt.show()


# Create a plot of daily deaths

plt.figure(figsize=(12, 6))

plt.plot(df['dateRep'], df['deaths'], label='Daily Deaths', color='r')

plt.title('Time Series of Daily Deaths')

plt.xlabel('Date')
plt.ylabel('Deaths')
plt.legend()
plt.show()

# Create a plot of monthly cases and deaths
monthly_data = df.groupby(['year', 'month'])[['cases', 'deaths']].sum().reset_index()
plt.figure(figsize=(12, 6))
sns.set_palette("Set1")
sns.lineplot(x='year', y='cases', hue='month', data=monthly_data, style='month', markers=True)
sns.lineplot(x='year', y='deaths', hue='month', data=monthly_data, style='month', markers=True)
plt.title('Monthly Cases and Deaths')
plt.xlabel('Year')
plt.ylabel('Count')
plt.xticks(rotation=45)
plt.legend(loc='upper left')
plt.show()

# Create a bar chart of the top 10 dates with the highest cases
top_10_dates_cases = df.nlargest(10, 'cases')
plt.figure(figsize=(12, 6))
sns.barplot(x='dateRep', y='cases', data=top_10_dates_cases, palette="Blues_d")
plt.title('Top 10 Dates with the Highest Cases')
plt.xlabel('Date')
plt.ylabel('Cases')
plt.xticks(rotation=45)
plt.show()

# Create a bar chart of the top 10 dates with the highest deaths
top_10_dates_deaths = df.nlargest(10, 'deaths')
plt.figure(figsize=(12, 6))
sns.barplot(x='dateRep', y='deaths', data=top_10_dates_deaths, palette="Reds_d")
plt.title('Top 10 Dates with the Highest Deaths')
plt.xlabel('Date')
plt.ylabel('Deaths')
plt.xticks(rotation=45)
plt.show()

# Create a boxplot of daily cases by month
plt.figure(figsize=(10, 6))
sns.boxplot(x='month', y='cases', data=df, palette='viridis')
plt.title('Boxplot of Daily Cases by Month')
plt.xlabel('Month')
plt.ylabel('Daily Cases')
plt.show()

# Create a scatter plot of cases vs. deaths
plt.figure(figsize=(12, 6))
sns.scatterplot(x='cases', y='deaths', data=df, hue='countriesAndTerritories', palette='viridis')
plt.title('Scatter Plot of Cases vs. Deaths')
plt.xlabel('Cases')
plt.ylabel('Deaths')
plt.legend(loc='upper left')
plt.show()

# Create a pairplot of the data
sns.pairplot(df[['cases', 'deaths', 'day', 'month', 'year']], diag_kind='kde', palette='coolwarm')
plt.suptitle('Pairplot for Correlations')
plt.show()

# Create a histogram of the cases distribution
plt.figure(figsize=(10, 6))
sns.histplot(df['cases'], kde=True, color='skyblue')
plt.title('Histogram of Cases Distribution')
plt.xlabel('Cases')
plt.ylabel('Frequency')
plt.show()

# Create a histogram of the deaths distribution
plt.figure(figsize=(10, 6))
sns.hist

# Create a violin plot of cases distribution by month
plt.figure(figsize=(12, 6))
sns.violinplot(x='month', y='cases', data=df, palette='coolwarm')
plt.title('Violin Plot of Cases Distribution by Month')
plt.xlabel('Month')
plt.ylabel('Cases')
plt.show()

# Create a kernel density estimation (KDE) plot for cases and deaths
plt.figure(figsize=(12, 6))
sns.kdeplot(data=df['cases'], shade=True, color='blue', label='Cases')
sns.kdeplot(data=df['deaths'], shade=True, color='red', label='Deaths')
plt.title('Kernel Density Estimation Plot for Cases and Deaths')
plt.xlabel('Count')
plt.ylabel('Density')
plt.legend()
plt.show()

# Create a pie chart of total cases by month
monthly_cases = df.groupby('month')['cases'].sum()
labels = [f'Month {month}' for month in monthly_cases.index]
plt.figure(figsize=(8, 8))
plt.pie(monthly_cases, labels=labels, autopct='%1.1f%%', startangle=140, colors=sns.color_palette('pastel'))
plt.title('Pie Chart of Total Cases by Month')
plt.show()

# Create a word cloud of countries and territories
country_text = " ".join(df['countriesAndTerritories'])
wordcloud = WordCloud(width=800, height=400, background_color='white').generate(country_text)
plt.figure(figsize=(10, 5))
plt.imshow(wordcloud, interpolation='bilinear')
plt.axis('off')
plt.title('Word Cloud of Countries and Territories')
plt.show()

# Create a donut plot of cases by month
monthly_cases = df.groupby('month')['cases'].sum()
labels = [f'Month {month}' for month in monthly_cases.index]
plt.figure(figsize=(8, 8))
plt.pie(monthly_cases, labels=labels, autopct='%1.1f%%', startangle=140, colors=sns.color_palette('pastel', len(monthly_cases)))
centre_circle = plt.Circle((0,0),0.70,fc='white')
fig = plt.gcf()
fig.gca().add_artist(centre_circle)
plt.title('Donut Plot of Cases by Month')
plt.show()

# Create a radar plot of monthly cases
monthly_cases = df.groupby('month')['cases'].sum()
months = [f'Month {month}' for month in monthly_cases.index]
values = monthly_cases.values
N = len(months)
angles = [n / float(N) * 2 * 3.14159265358979323846 for n in range(N)]
angles += angles[:1]
values = list(values)
values += values[:1]
plt.figure(figsize=(10, 10))
plt.polar(angles, values, marker='o')
plt.fill(angles, values, 'teal', alpha=0.1)
plt.xticks(angles[:-1], months)
plt.title('Radar Plot of Monthly Cases')
plt.show()


# Create a parallel coordinates plot
df_normalized = (df[['cases', 'deaths']] - df[['cases', 'deaths']].mean()) / df[['cases', 'deaths']].std()
df_normalized['month'] = df['month']
plt.figure(figsize=(12, 6))
parallel_coordinates(df_normalized, 'month', colormap='viridis')
plt.title('Parallel Coordinates Plot')
plt.xlabel('Feature')
plt.ylabel('Normalized Value')
plt.show()


# Create a waterfall chart for cumulative cases
df['cumulative_cases'] = df['cases'].cumsum()
plt.figure(figsize=(12, 6))
plt.bar(df['dateRep'], df['cumulative_cases'], color='lightblue', label='Cases')
plt.title('Waterfall Chart for Cumulative Cases')
plt.xlabel('Date')
plt.ylabel('Cumulative Cases
```

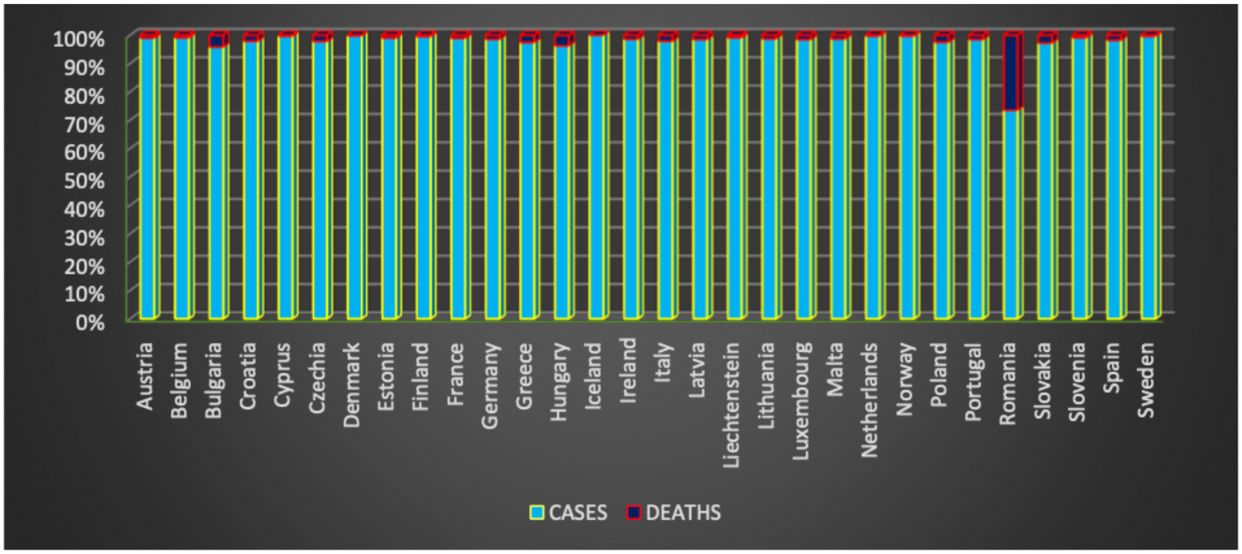
deaths by day

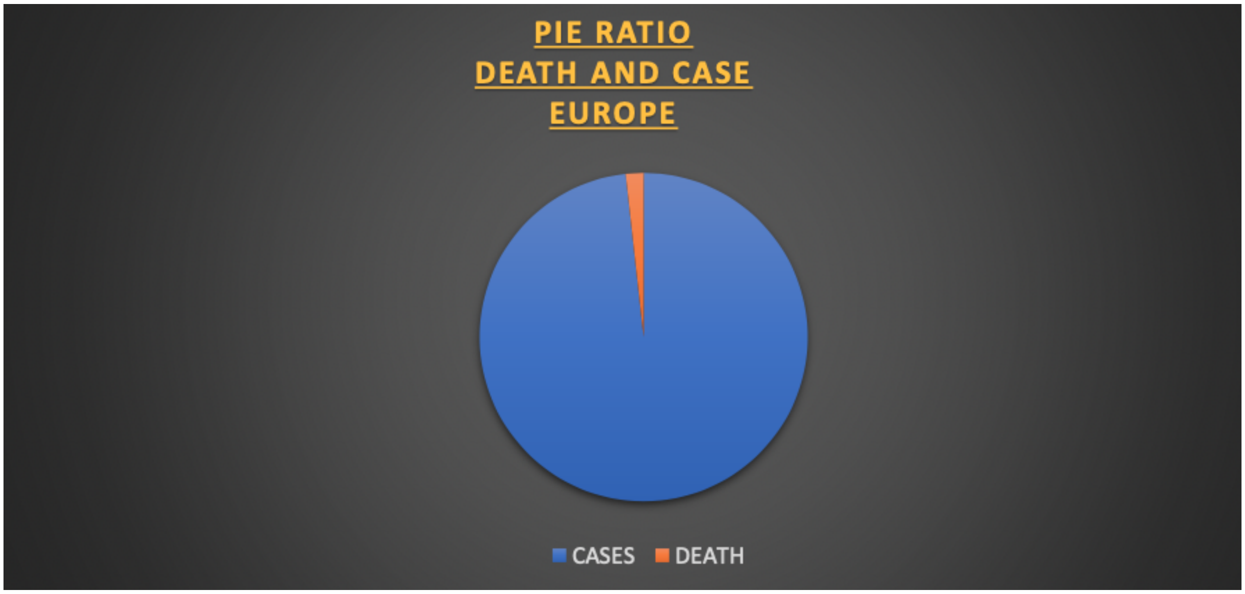


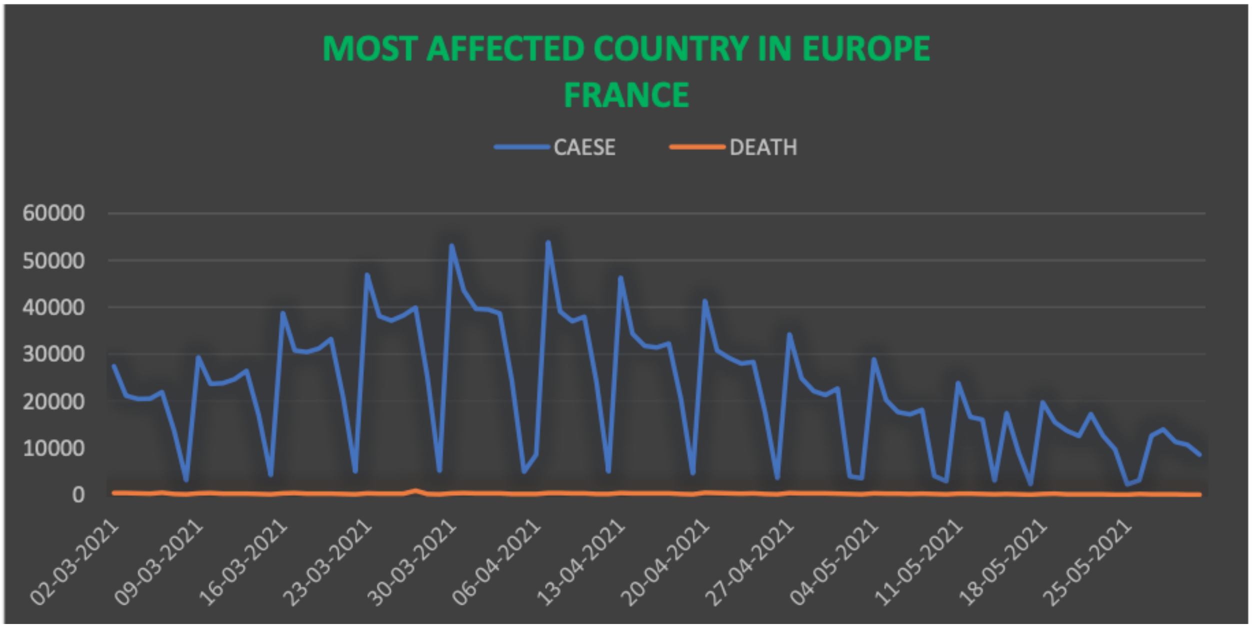
deaths by day sized by cases

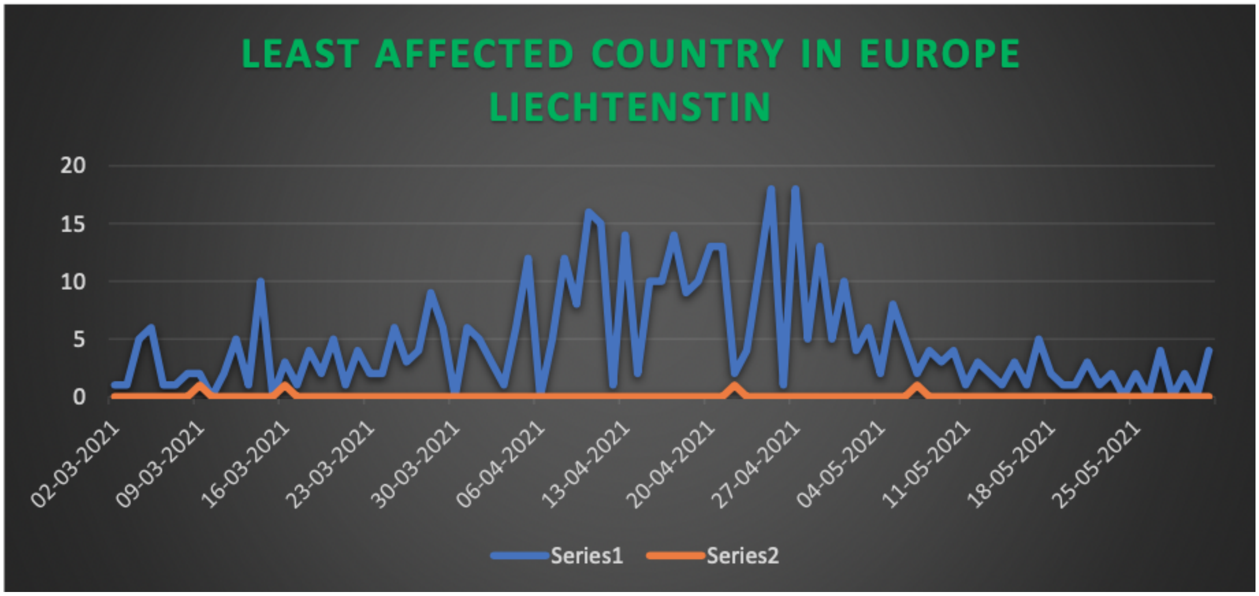



deaths and cases by day
Measures
● deaths
● cases
day
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
0
50,000
100,000
150,000
200,000
250,000
300,000
350,000
4
Values


100%
90%
80%
70%
60%
50%
40%
30%
20%
10%
0%
Austria
Belgium
Bulgaria
Croatia
Cyprus
Czechia
Denmark
Estonia
Finland
France
Germany
Greece
Hungary
Iceland
Ireland
Italy
Latvia
Liechtenstein
Lithuania
Luxembourg
Malta
Netherlands
Norway
Poland
Portugal
Romania
Slovakia
Slovenia
Spain
Sweden
□ CASES
■ DEATHS


PIE RATIO
DEATH AND CASE
EUROPE
CASES
DEATH


MOST AFFECTED COUNTRY IN EUROPE
FRANCE
CAESE
DEATH
60000
50000
40000
30000
20000
10000
0
02-03-2021
09-03-2021
16-03-2021
23-03-2021
30-03-2021
06-04-2021
13-04-2021
20-04-2021
27-04-2021
04-05-2021
11-05-2021
18-05-2021
25-05-2021


LEAST AFFECTED COUNTRY IN EUROPE
LIECHTENSTIN
20
15
10
5
0
02-03-2021
09-03-2021
16-03-2021
23-03-2021
30-03-2021
06-04-2021
13-04-2021
20-04-2021
27-04-2021
04-05-2021
11-05-2021
18-05-2021
25-05-2021
Series1
Series2

Insights :

- Cases are expected to exceed 270,000 by day 38, but they have been trending slightly downward.
- There is an unusually low number of cases at time point 31. This may be due to incomplete data or a recent event that requires further investigation.
- Deaths range from 0 to 956, with the highest number of deaths occurring when cases are high.
- Over 178,000 deaths have been reported in total.
- There was a 4,450% increase in deaths from May 9 to 10, 2021.
- Deaths are unusually high when cases are 27,890 and 39,932.

Conclusion

The COVID-19 pandemic is still ongoing and that the number of cases and deaths is expected to continue to increase. However, the data also shows that the number of cases and deaths has been trending slightly downward in recent weeks. This suggests that the pandemic may be peaking, but it is important to remain vigilant and continue to take precautions to prevent the spread of the virus.

The unusually high number of deaths at certain time points may be due to a number of factors, such as incomplete data, recent events, or changes in the virus itself. It is important to investigate these factors further to better understand the dynamics of the pandemic and to develop effective public health interventions.

Overall, the data suggests that the COVID-19 pandemic is still a serious threat, but there are signs that the pandemic may be peaking. It is important to continue to monitor the situation closely and to take precautions to prevent the spread of the virus.