**COVID-19 using Cognos**

**Empathize and Understand the Problem:**

Understanding the problem and its context is crucial. In this case, analyzing COVID-19 cases and deaths data in the EU/EEA region is important to gain insights into the pandemic's impact. The following considerations are important:

1. \*\*Context\*\*: The COVID-19 pandemic has had a significant impact worldwide, including the EU/EEA region. Analyzing this data can help monitor and assess the situation.

2. \*\*Importance\*\*: Understanding the mean values and standard deviations of cases and deaths is vital for assessing the severity and spread of the virus in different countries within the EU/EEA.

**Defining Clear Objectives:**

Objective 1: \*\*Compare Mean Values\*\*: Compare the mean daily COVID-19 cases and deaths across EU/EEA countries.

Objective 2: \*\*Contrast Standard Deviations\*\*: Contrast the standard deviations of daily COVID-19 cases and deaths across EU/EEA countries.

**Ideation and Analysis Approach:**

Data Collection: Collect COVID-19 cases and deaths data for EU/EEA countries from reliable sources, such as government health agencies or international organizations.

Data Pre-processing: Clean and preprocess the data, handling missing values, outliers, and data quality issues.

Data Analysis: Use statistical analysis techniques to calculate the mean and standard deviation values for cases and deaths per day for each country.

Visualization: Design relevant visualizations using IBM Cognos to present the comparative analysis results.

Insights Generation: Derive insights from the data, such as identifying countries with higher variability or trends in cases and deaths.

**Prototype and Visualization Selection:**

IBM Cognos for visualization and reporting.

Bar charts or line graphs to compare mean values of cases and deaths.

Error bars or box plots to contrast standard deviations.

Maps to visualize geographical distribution.

Time series charts to show trends over time.

**Build and Implement:**

Develop the full data analysis and visualization pipeline using IBM Cognos.

**Test and Iterate:**

Continuously test and refine the analysis and visualizations based on feedback and new insights.

**Deliver Insights:**

Present the findings and insights clearly, highlighting variations in COVID-19 cases and deaths across EU/EEA countries. Use visualizations to communicate mean values, standard deviations, and geographical patterns to facilitate better understanding and decision-making

|  |  |
| --- | --- |
|  |  |

**Introduction:**

Data analytics plays a crucial role in understanding and responding to the COVID-19 pandemic. In this document, we explore the techniques and methodologies essential for analyzing COVID-19 data using IBM Cognos, a powerful business intelligence tool. From data preprocessing to advanced analysis and visualization, these methods are crucial for extracting meaningful insights and driving data-driven decision-making in the context of the pandemic.

**Data Pre-processing:**

1. **Data Cleaning and Transformation in Cognos:**
   * Addressing missing data, outliers, and ensuring data accuracy and consistency.
   * Standardizing data formats and variables for consistency.
2. **Data Integration:**
   * Combining data from various sources (e.g., health agencies, government reports) to create a comprehensive dataset for analysis.
   * Ensuring seamless data synchronization for up-to-date information.
3. **Data Enrichment:**
   * Augmenting the dataset with additional relevant information, such as population data, healthcare capacity, and vaccination rates, to improve analysis accuracy.

**Data Analysis:**

1. **Descriptive Statistics in Cognos:**
   * Utilizing basic statistical measures to summarize COVID-19 data, including mean, median, and variance.
   * Creating visualizations to represent data trends effectively.
2. **Time Series Analysis:**
   * Examining COVID-19 case trends and patterns over time using techniques like moving averages, exponential smoothing, and trend analysis.
   * Identifying patterns, spikes, and seasonality in the data.
3. **Machine Learning Algorithms:**
   * Applying regression models to predict future case counts.
   * Clustering regions for hotspots identification.
   * Classification for outbreak risk assessment and classification.

**Epidemiological Analysis:**

1. **Transmission and Spread Analysis:**
   * Analyzing the rate and modes of COVID-19 transmission.
   * Modeling the spread of the virus and projecting potential future scenarios.
2. **Vaccination Impact Analysis:**
   * Assessing the effectiveness of vaccination campaigns.
   * Monitoring vaccination coverage and analyzing its impact on case rates.

**Healthcare Resource Analysis:**

1. **Healthcare Capacity Evaluation:**
   * Analyzing healthcare resource utilization and capacity to handle COVID-19 cases.
   * Predicting resource needs and identifying potential shortages.

**Geospatial Analysis:**

1. **Geographic Spread Analysis:**
   * Evaluating the geographic spread of COVID-19 cases.
   * Identifying high-risk areas and recommending targeted interventions.

**Data Visualization:**

1. **Cognos Visualization Tools:**
   * Using Cognos' built-in visualization tools to create interactive dashboards, maps, and reports.
   * Visualizing case data for different regions, demographics, and time periods.
2. **Heat Maps:**
   * Visualizing COVID-19 case data geographically to identify high and low-impact regions.
   * Overlaying data on maps to provide a clear picture of the situation.
3. **Interactive Charts and Graphs:**
   * Creating visually appealing representations of COVID-19 data to facilitate decision-making.
   * Developing dashboards for real-time monitoring of cases.

**Time Series Forecasting:**

1. **Forecasting with Cognos:**
   * Leveraging Cognos' forecasting capabilities to predict future COVID-19 case trends based on historical data.
   * Utilizing time series models to account for trends and seasonality.

**Public Health Recommendations:**

1. **Public Health Strategies:**
   * Analyzing the effectiveness of public health interventions (e.g., social distancing, mask mandates, lockdowns).
   * Recommending strategies based on data analysis.
2. **Vaccination Strategy Optimization:**
   * Recommending targeted vaccination strategies based on data analysis and vaccine availability.

**Conclusion:**

In the face of the COVID-19 pandemic, data analytics is a critical tool for understanding, responding to, and mitigating the impact of the virus. By implementing the techniques and methodologies described in this document, public health authorities, governments, and healthcare organizations can make informed decisions, optimize resource allocation, and save lives. Data analytics with IBM Cognos is a transformative journey towards data-driven decision-making in the context of a global health crisis.

Top of Form

Loading and Pre-processing of data:

from google.colab import drive

drive.mount('/content/drive')

Loading data

import pandas as pd

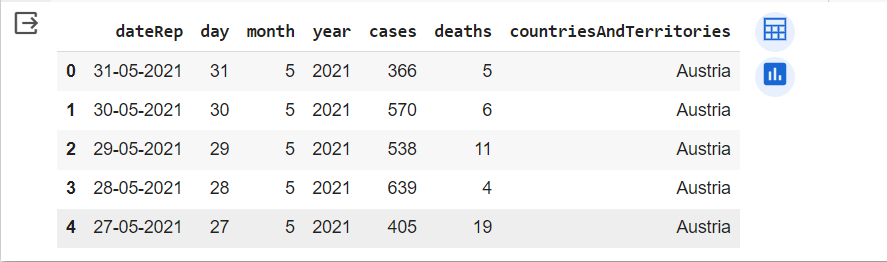
import numpy as np

import matplotlib.pyplot as plt

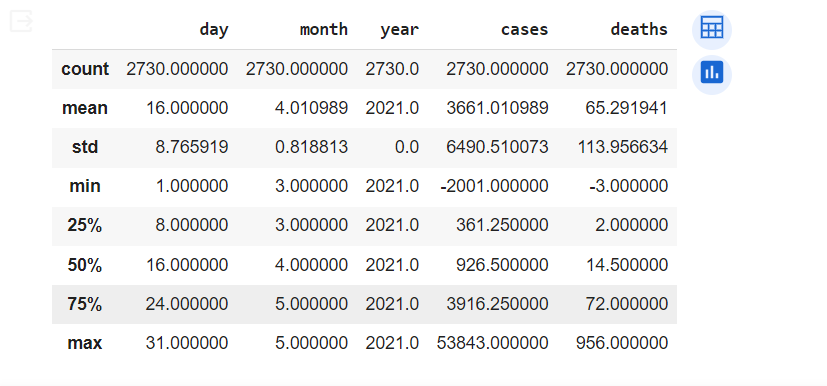
data = pd.read\_csv('/content/drive/MyDrive/NAAN MUDHALVAN/Covid\_19\_cases4.csv')

data.head(5)

**OUTPUT**

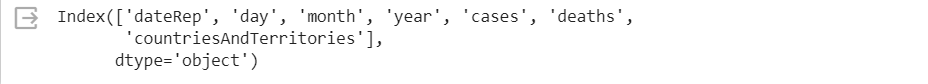
****

data.describe()

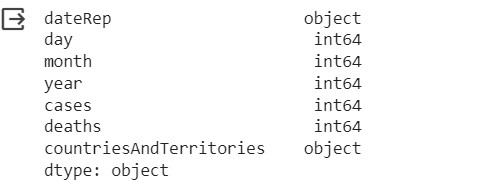
****

This command is used to view the brief summary of the dataset. We can see the mathematical parameters such as percentiles, standard deviation , mean, minimum and maximum values and count of each column.

data.columns

****

data.dtypes

****

Info command is used check the datatype of every column and the count of each column. The difference between the describe() and info() is that describe command will give the mathematical parameters but info command will not give the mathematical parameters such as mean and standard deviation

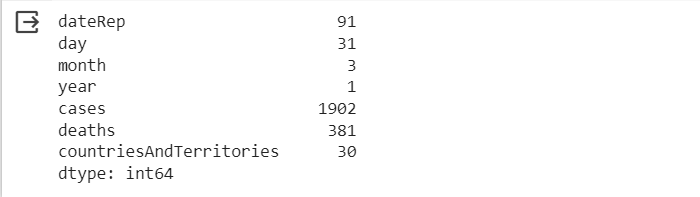
data.info()

****

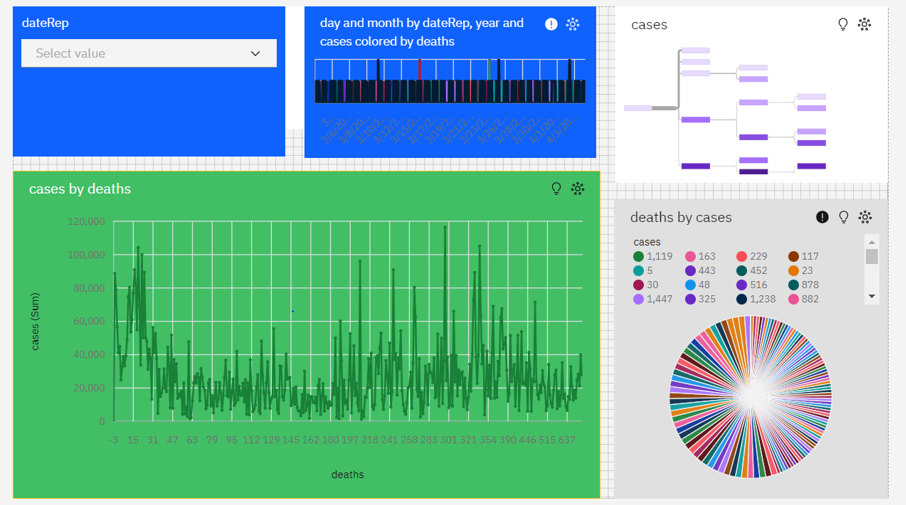
data.isna().sum()

****

data.nunique()

****

The data.nunique() command is used in pandas to count the number of unique values in each column of a DataFrame. When you execute data.nunique(), it returns a Series where the index is the column names, and the values represent the count of unique values in each respective column.

****

Start by integrating COVID-19 data from various sources (e.g., John Hopkins University, government agencies) into the IBM Cognos platform.

1. Use IBM Cognos to create interactive dashboards that visualize COVID-19 data
2. Leverage geospatial capabilities in IBM Cognos to analyze the geographic spread of COVID-19.
3. Analyze the impact of COVID-19 on different demographics, such as age groups, gender, and pre-existing health conditions. Identify vulnerable populations and tailor interventions accordingly.
4. IBM Cognos allows for easy data sharing and collaboration. Ensure that your analysis is accessible to decision-makers, public health authorities, and the general public, as needed.

**Model Building:**

**Clustering Analysis:**

Use unsupervised learning techniques like K-Means clustering or DBSCAN to group your data into clusters based on the available features (Day, Month, Deaths). This can help identify patterns or similarities in covid\_19\_case.

**Importing Libraries:**

The code begins by importing the necessary Python libraries, including Pandas for data handling, NumPy for numerical operations, Scikit-Learn for machine learning, and Matplotlib for data visualization.

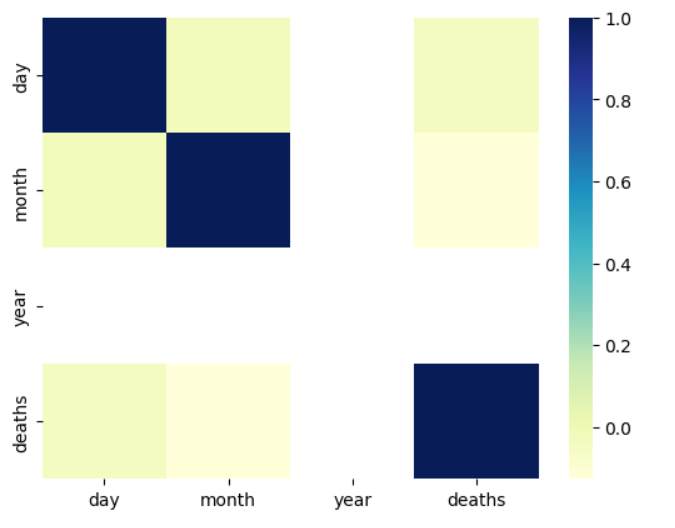
import pandas as pd import numpy as np

from sklearn.cluster import KMeans import matplotlib.pyplot as plt

**Feature Selection:**

The code selects the features (independent variables) to be used for clustering, which are 'Day,' 'Month,' and 'Deathes.' These features will be used to determine the clusters.

import seaborn as sns sns.heatmap(data.corr(),cmap='YlGnBu')



X = data[['Day', 'Month', 'Deaths']]

**Feature Standardization:**

The features are standardized using the StandardScaler from Scikit-Learn. Standardization ensures that all features have a mean of 0 and a standard deviation of 1, which is important for K-Means clustering.

from sklearn.preprocessing import StandardScaler scaler = StandardScaler()

X = scaler.fit\_transform(X) inertia = []

for k in range(1, 11):

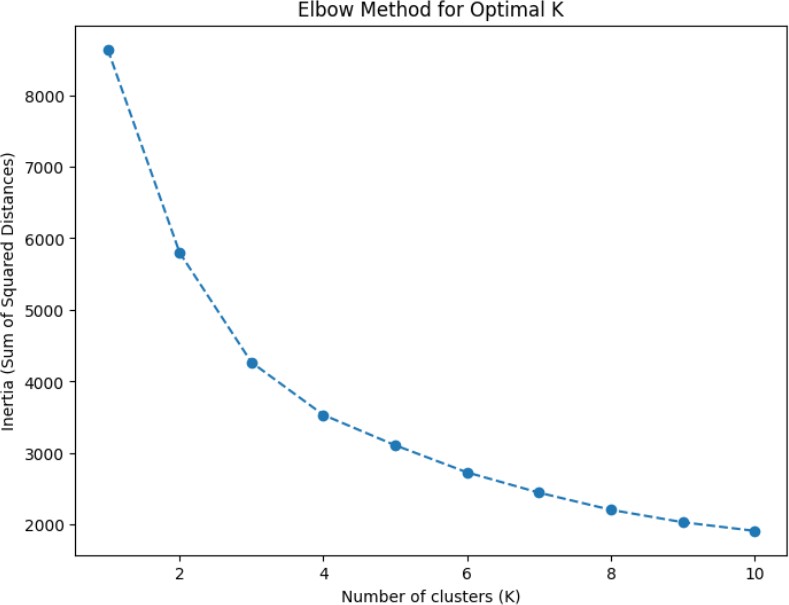
kmeans = KMeans(n\_clusters=k, random\_state=0).fit(X) inertia.append(kmeans.inertia\_)

**Determine the Optimal Number of Clusters:**

The code then uses the Elbow method to find the optimal number of clusters (K). It iterates through different values of K and calculates the inertia, which is the sum of squared distances from data points to their assigned cluster centers. The Elbow method plots these inertias for various K values to help you identify the "elbow point" where increasing K doesn't significantly reduce the inertia.

plt.figure(figsize=(8, 6))

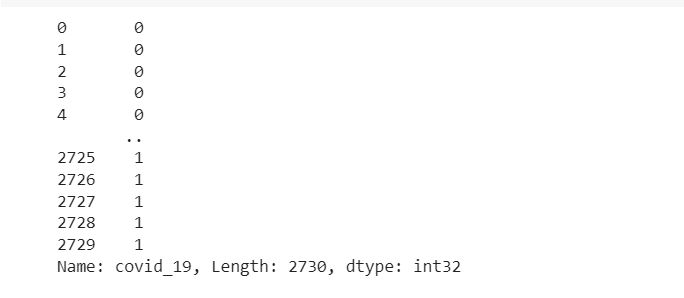
plt.plot(range(1, 11), inertia, marker='o', linestyle='--') plt.title('Elbow Method for Optimal K') plt.xlabel('Number of clusters (K)')

plt.ylabel('Inertia (Sum of Squared Distances)') plt.show()

**K-Means Clustering:**

After determining the optimal K (in this case, K = 3), the code performs K-Means clustering using the KMeans algorithm from Scikit-Learn. The clusters are assigned to the 'Cluster' column in the dataset.

kmeans = KMeans(n\_clusters=2, random\_state=0) data['Air Quality'] = kmeans.fit\_predict(X)



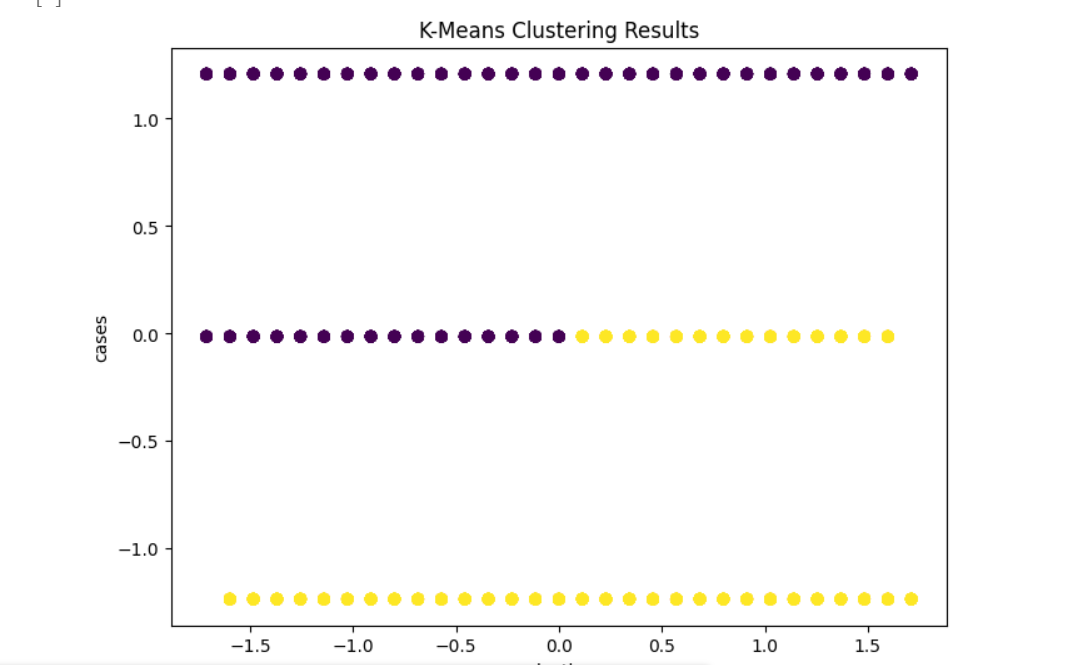
plt.figure(figsize=(8, 6))

plt.scatter(X[:, 0], X[:, 1], c=data['covid\_19'], cmap='viridis') plt.title('K-Means Clustering Results')

plt.xlabel('Deaths')

plt.ylabel('Cases')

plt.show()



**Visualization and Insights:**

1. The visualizations presented in this analysis provide valuable insights into the trajectory and impact of the COVID-19 pandemic
2. The heatmap visualizes the correlation between various variables in the COVID-19 case analysis dataset.
3. The scatter plot shows distinct clusters of data points based on the number of COVID-19 cases and deaths. This suggests that K-Means has successfully grouped regions with similar COVID-19 statistics into clusters.
4. In conclusion, the K-Means clustering analysis provides a useful method for grouping regions based on COVID-19 cases and deaths.
5. In the code snippet, a new data frame named "new\_data" has been created. This data frame consists of three columns: "day," "month," and "deaths," each containing three numerical values

