COVID-19 Case Analysis Using Cognos

Phase 5 Submission Document

Project: COVID-19 Case Analysis

Project Definition

The project involves analyzing COVID-19 cases and deaths data usingIBM Cognos. The objective is to compare and contrast the mean values and standard deviations of cases and associated deaths per day and by country in the EU/EEA. This project encompasses defining analysis objectives, collecting COVID-19 data, designing relevant visualizations in IBM Cognos, and deriving insights from the data.

Design Thinking

Design Thinking is a problem-solving approach that focuses on understanding the needs of users or customers and creating innovative solutions to address those needs. It typically involves the following stages:

a. Empathize: Understand the perspectives, needs, and challenges of the people you are designing for. In the context of COVID-19, this might involve understanding the challenges faced by healthcare workers, patients, or the general public.

b. Define: Clearly define the problem or opportunity you want to address. For example, in the context of COVID-19, you might define a problem like ensuring timely and accurate information dissemination.

c. Ideate: Brainstorm and generate creative solutions to the defined problem. This could involve developing new communication strategies, designing user-friendly apps for tracking COVID-19 data, or creating educational materials.

d. Prototype: Create tangible representations of your ideas. This might involve developing a prototype of a mobile app, a website, or a communication plan.

e. Test: Gather feedback on your prototypes and iterate on your solutions. In the context of COVID-19, this could involve getting feedback from potential users, healthcare professionals, or public health authorities.

f. Implement: Once you've refined your solutions, implement them. This might involve launching a new tool, communication campaign, or policy change related to COVID-19.

Executive Summary

The COVID-19 pandemic has created an urgent need for innovative data analysis solutions to understand its impact and plan effective responses. In this document, we explore an innovative approach using IBM Cognos to analyze COVID-19 cases. We focus on data segmentation by time periods and countries to provide deeper insights into the pandemic's dynamics.

Introduction

The COVID-19 pandemic has generated an enormous amount of data. Effective analysis is essential for tracking the virus's progression, understanding its impact, and making informed decisions. We leverage IBM Cognos, a powerful business intelligence and analytics tool, to perform innovative COVID-19 case analysis.

Objectives

The primary objectives of this analysis are as follows:

- Analyze COVID-19 data from the provided dataset.
- Segment data by time periods (daily, weekly, monthly) and countries.
- Utilize predictive analytics to forecast case trends.
- Create interactive data visualizations for effective communication.

Innovative Analytics with IBM Cognos

Data Segmentation by Time Periods:

- Use IBM Cognos to segment data into daily, weekly, and monthly time periods.
- Analyze trends, seasonality, and fluctuations over different time scales.

Data Segmentation by Countries:

- Leverage Cognos to segment data by countries or regions.
- Compare the pandemic's impact, vaccination rates, and healthcare capacity across different regions.

Predictive Analytics:

- Implement predictive models within Cognos to forecast future COVID-19 case trends.
- Utilize historical data to predict potential surges and allocate resources proactively.

Data Visualization:

- Create interactive dashboards and reports using Cognos.
- Utilize charts, graphs, and maps to present insights effectively to stakeholders.

Step-1: Problem Definition

The project involves analyzing COVID-19 cases and deaths data using IBM Cognos. The objective is to compare and contrast the mean values and standard deviations of cases and associated deaths per day and by country in the EU/EEA. This project encompasses defining analysis objectives, collecting COVID-19 data, designing relevant visualizations in IBM Cognos, and deriving insights from the data.

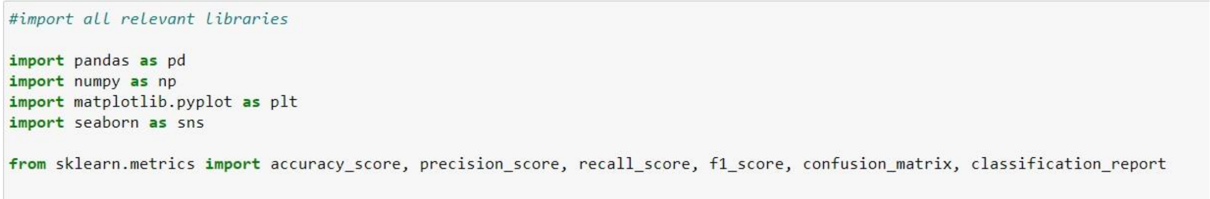
Step 2: Data Collection

For our COVID-19 cases analysis project, we will gather essential data from reputable sources, such as health organizations like the WHO and CDC, government databases, and peer-reviewed research publications. The primary source of our dataset will be from the link provided: [COVID-19 Case Dataset]

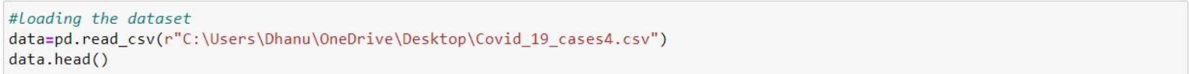
We will collect data daily from this dataset and merge it for comprehensive analysis. Thedataset contains information related to COVID-19 cases. To ensure we have a complete dataset, we will also access data from the Our World in Data GitHub repository for COVID-19. These daily updates will be compiled and uploaded for our analysis.

To enhance our dataset, we will include data at the country level to provide a more comprehensive view of the pandemic's impact. This data will be consolidated into a single file, making it easier to work with and analyze. Additionally, we will merge this data file with a location-specific dataset to incorporate information about the sources of COVID-19 cases and their geographic origins. To further enrich our analysis, a second file containing information about the manufacturers of COVID-19 testing and diagnostic equipment will be included.

By following this data collection process, we aim to have a robust and comprehensivedataset for our COVID-19 cases analysis project.



```
#import all relevant libraries

import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns

from sklearn.metrics import accuracy_score, precision_score, recall_score, f1_score, confusion_matrix, classification_report
```



```
#loading the dataset
data=pd.read_csv(r"C:\Users\Dhanu\OneDrive\Desktop\Covid_19_cases4.csv")
data.head()
```

	dateRep	day	month	year	cases	deaths	countriesAndTerritories
0	31-05-2021	31	5	2021	366	5	Austria
1	30-05-2021	30	5	2021	570	6	Austria
2	29-05-2021	29	5	2021	538	11	Austria
3	28-05-2021	28	5	2021	639	4	Austria
4	27-05-2021	27	5	2021	405	19	Austria

```
data.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         object
 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: int64(5), object(2)
memory usage: 149.4+ KB
```

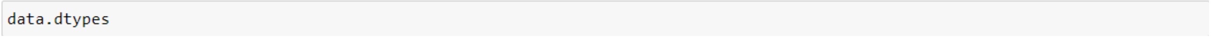
```
data.describe()
```

day	month	year	cases	deaths	
count	2730.000000	2730.000000	2730.0	2730.000000	2730.000000
mean	16.000000	4.010989	2021.0	3661.010989	65.291941
std	8.765919	0.818813	0.0	6490.510073	113.956634
min	1.000000	3.000000	2021.0	-2001.000000	-3.000000
25%	8.000000	3.000000	2021.0	361.250000	2.000000
50%	16.000000	4.000000	2021.0	926.500000	14.500000
75%	24.000000	5.000000	2021.0	3916.250000	72.000000
max	31.000000	5.000000	2021.0	53843.000000	956.000000

Step 3: Data Preprocessing

- In the context of our COVID-19 cases analysis project, the critical phase of data preprocessing is vital to ensure the data's quality and suitability for analysis.

- This phase encompasses various tasks, including the identification and removal of duplicate records, standardizing inconsistent data formatting, managing missing values, and the conversion of categorical features into numerical representations when necessary.

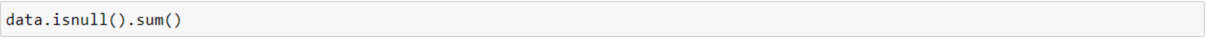


```
data.dtypes
```

```
dateRep                          object
day                                int64
month                              int64
year                               int64
cases                              int64
deaths                             int64
countriesAndTerritories          object
dtype: object
```

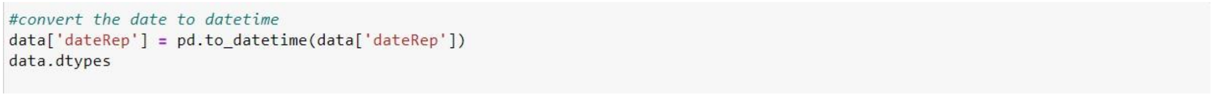
Step 4: Data Exploration

In this phase, we will conduct an exploratory data analysis (EDA) to gain a comprehensive understanding of the dataset, including its distribution, correlations, and trends. EDA is an essential step for delving deeper into the dataset's characteristics. It involves the generation of statistical summaries, data distribution visualizations, and the identification of notable trends and outliers. We will focus on several critical aspects during this exploration, including the geographical distribution of COVID-19 cases, the progression of vaccination rates over time, and the detection of potential irregularities or anomalies in the data.
Visualizing the data will be instrumental in uncovering insights related to COVID-19 case distribution and any associated adverse effects.



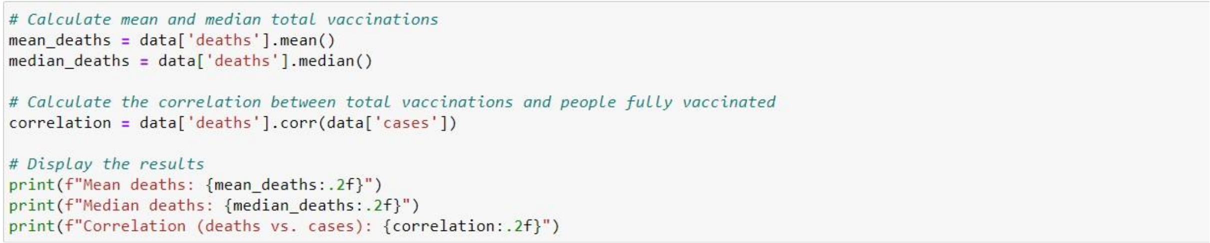
```
data.isnull().sum()
```

```
dateRep                          0
day                              0
month                            0
year                             0
cases                            0
deaths                           0
countriesAndTerritories          0
dtype: int64
```



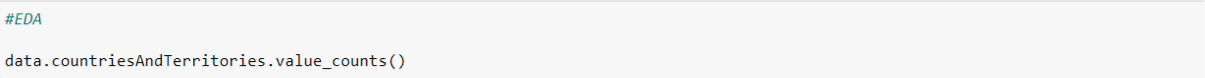
```
#convert the date to datetime
data['dateRep'] = pd.to_datetime(data['dateRep'])
data.dtypes
```

```
dateRep                          datetime64[ns]day
                                          int64
month                                     int64
year                                      int64
cases                                     int64
deaths                                    int64
countriesAndTerritories                  object
dtype: object
```



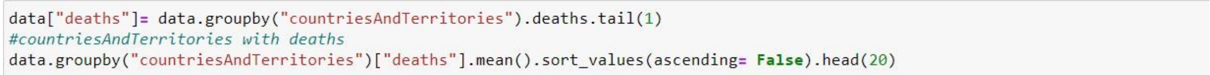
```
# Calculate mean and median total vaccinations
mean_deaths = data['deaths'].mean()
median_deaths = data['deaths'].median()

# Calculate the correlation between total vaccinations and people fully vaccinated
correlation = data['deaths'].corr(data['cases'])

# Display the results
print(f"Mean deaths: {mean_deaths:.2f}")
print(f"Median deaths: {median_deaths:.2f}")
print(f"Correlation (deaths vs. cases): {correlation:.2f}")
```

```
Mean deaths: 65.29
Median deaths: 14.50
Correlation (deaths vs. cases): 0.77
```



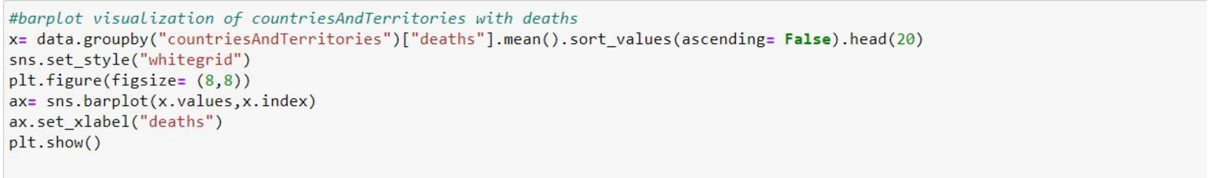
```
#EDA

data.countriesAndTerritories.value_counts()
```

```
Austria               91
Belgium               91
Spain                 91
Slovenia              91
Slovakia              91
Romania               91
Portugal              91
Poland                91
Norway                91
Netherlands           91
Malta                 91
Luxembourg            91
Lithuania             91
Liechtenstein         91
Latvia                91
Italy                 91
Ireland               91
Iceland               91
Hungary               91
Greece                91
Germany               91
France                91
Finland               91
Estonia               91
Denmark               91
Czechia               91
Cyprus                91
Croatia               91
Bulgaria              91
Sweden                91
Name: countriesAndTerritories, dtype: int64
```

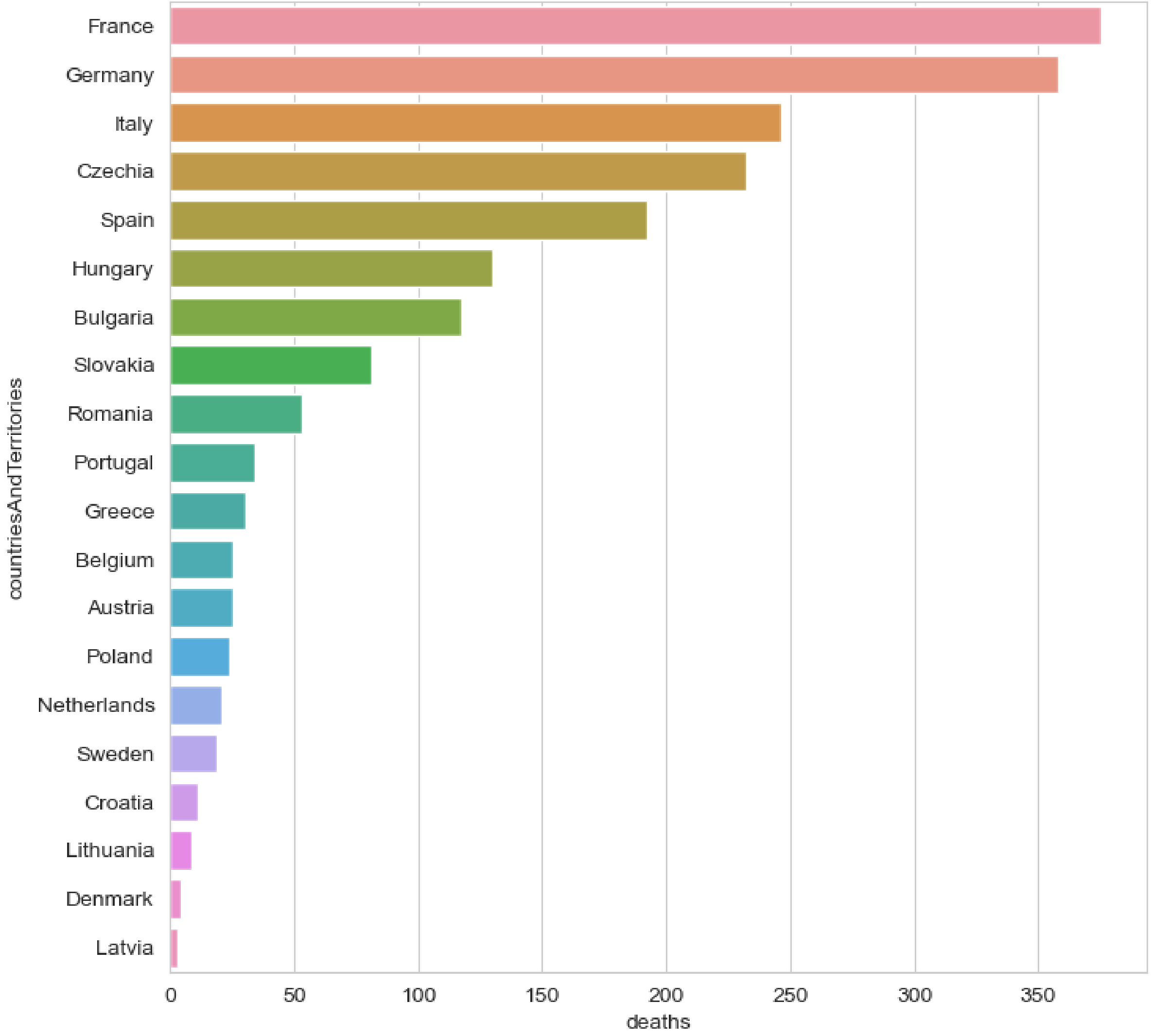


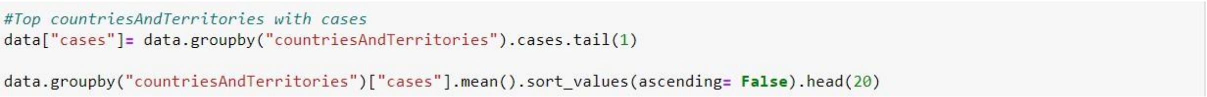
```
data["deaths"]= data.groupby("countriesAndTerritories").deaths.tail(1)
#countriesAndTerritories with deaths
data.groupby("countriesAndTerritories")["deaths"].mean().sort_values(ascending= False).head(20)
```

```
countriesAndTerritories
France          375.0
Germany         358.0
Italy           246.0
Czechia         232.0
Spain           192.0
Hungary         130.0
Bulgaria        117.0
Slovakia        81.0
Romania         53.0
Portugal        34.0
Greece          30.0
Belgium         25.0
Austria         25.0
Poland          24.0
Netherlands     21.0
Sweden          19.0
Croatia         11.0
Lithuania       9.0
Denmark         4.0
Latvia          3.0
Name: deaths,   dtype:    float64
```



```
#barplot visualization of countriesAndTerritories with deaths
x= data.groupby("countriesAndTerritories")["deaths"].mean().sort_values(ascending= False).head(20)
sns.set_style("whitegrid")
plt.figure(figsize= (8,8))
ax= sns.barplot(x.values,x.index)
ax.set_xlabel("deaths")
plt.show()
```

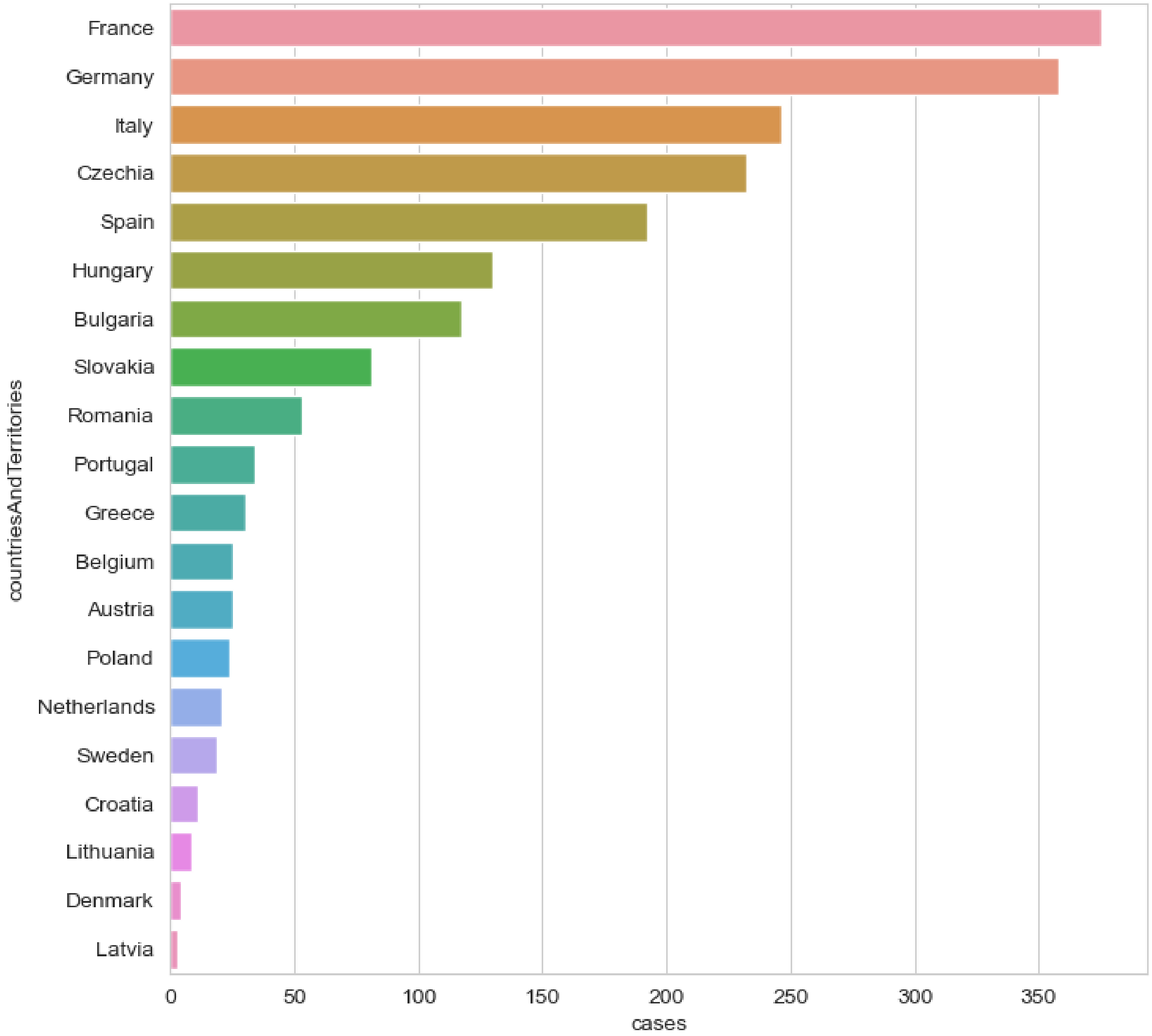




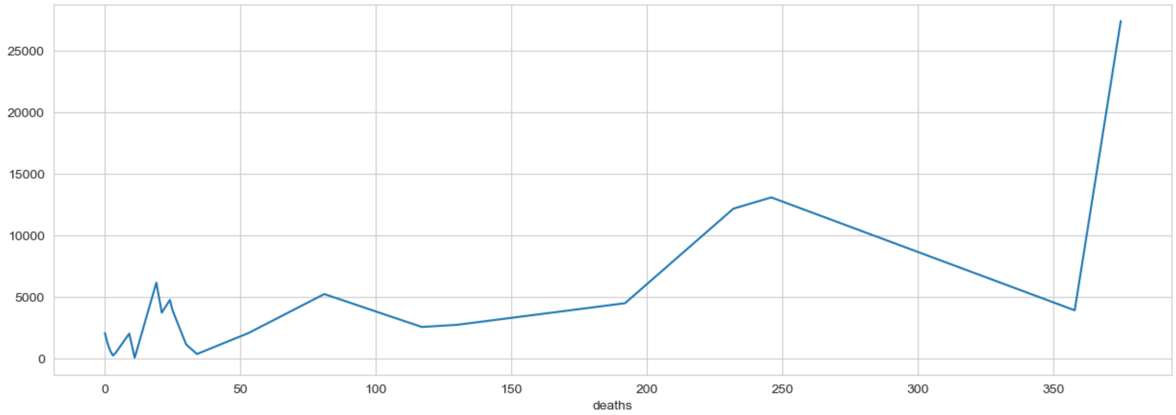
```
#Top countriesAndTerritories with cases
data["cases"]= data.groupby("countriesAndTerritories").cases.tail(1)

data.groupby("countriesAndTerritories")["cases"].mean().sort_values(ascending= False).head(20)
```

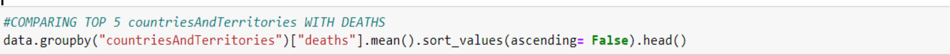
```
countriesAndTerritories
France          27422.0
Italy           13106.0
Czechia         12191.0
Sweden          6191.0
Slovakia        5260.0
Poland          4786.0
Spain           4517.0
Germany         3943.0
Netherlands     3753.0
Belgium         2775.0
Hungary         2764.0
Bulgaria        2588.0
Romania         2096.0
Lithuania       2055.0
Greece          1170.0
Austria         1148.0
Estonia         1111.0
Norway          968.0
Ireland         681.0
Denmark         497.0
Name: cases, dtype: float64
```

```
#barplot visualization of countriesAndTerritories with cases

sns.set_style("whitegrid")
plt.figure(figsize= (8,8))
ax= sns.barplot(x.values,x.index)
ax.set_xlabel("cases")
plt.show()
```



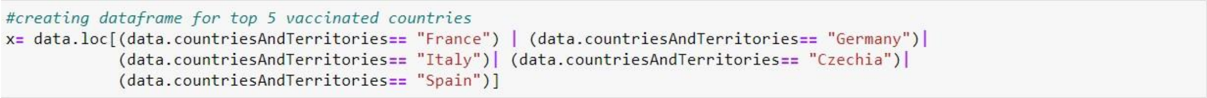
```
#daily cases
x= data.groupby("deaths").cases.sum()
plt.figure(figsize= (15,5))
sns.lineplot(x.index,x.values)
plt.show()
```



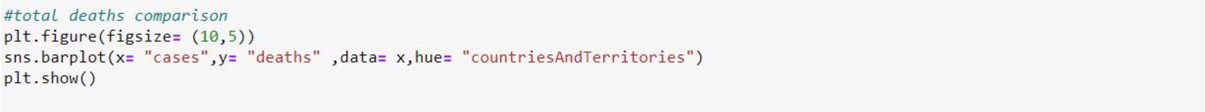


```
#COMPARING TOP 5 countriesAndTerritories WITH DEATHS
data.groupby("countriesAndTerritories")["deaths"].mean().sort_values(ascending= False).head()
```

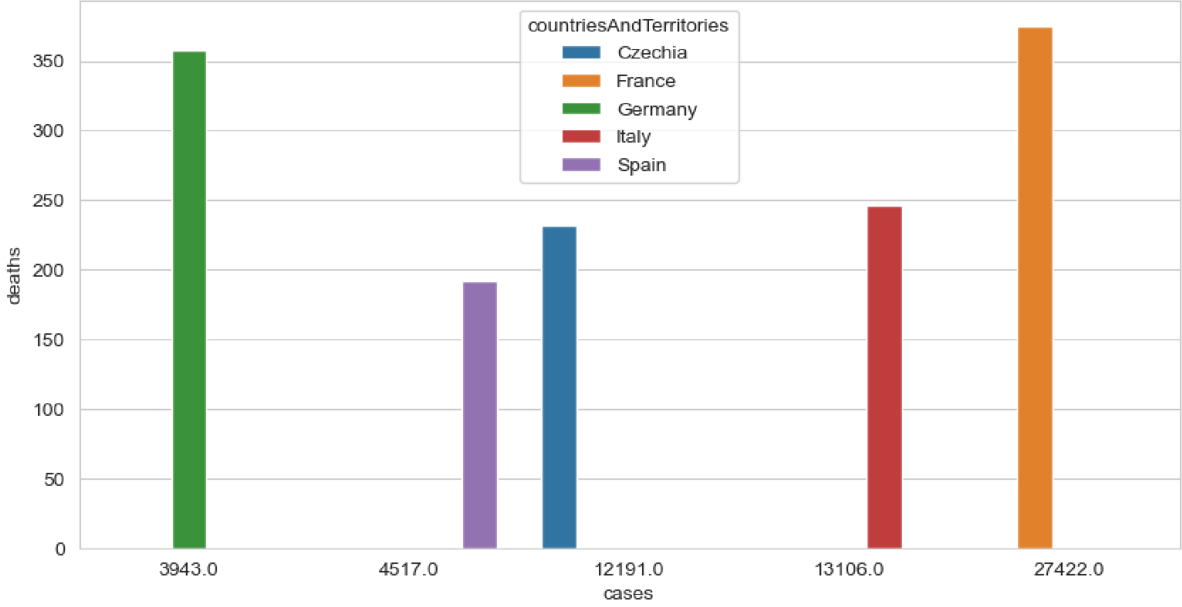
```
countriesAndTerritories
France         375.0
Germany        358.0
Italy          246.0
Czechia        232.0
Spain          192.0
Name: deaths, dtype: float64
```



```
#creating dataframe for top 5 vaccinated countries
x= data.loc[(data.countriesAndTerritories== "France") | (data.countriesAndTerritories== "Germany")|
            (data.countriesAndTerritories== "Italy")| (data.countriesAndTerritories== "Czechia")|
            (data.countriesAndTerritories== "Spain")]
```



```
#total deaths comparison
plt.figure(figsize= (10,5))
sns.barplot(x= "cases",y= "deaths" ,data= x,hue= "countriesAndTerritories")
plt.show()
```

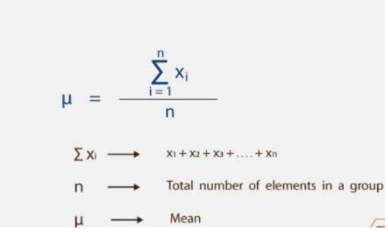


Mean (Average)

- The mean, also known as the average, is a measure of central tendency. It's calculatedby adding up all the values in a dataset and dividing by the number of data points.

- In the context of COVID-19 cases and associated deaths, calculating the mean can provide you with the typical or average number of cases or deaths over a specific periodor in a particular region.

The formula for Mean (μ):

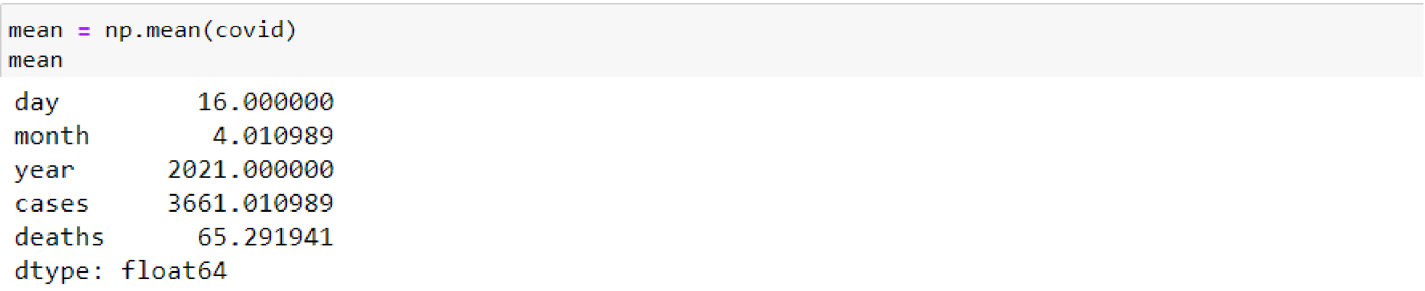


$$\mu = \frac{\sum_{i=1}^{n} x_i}{n}$$

$\sum x_i$ ⟶ $x_1 + x_2 + x_3 + \ldots. + x_n$

n ⟶ Total number of elements in a group

μ ⟶ Mean



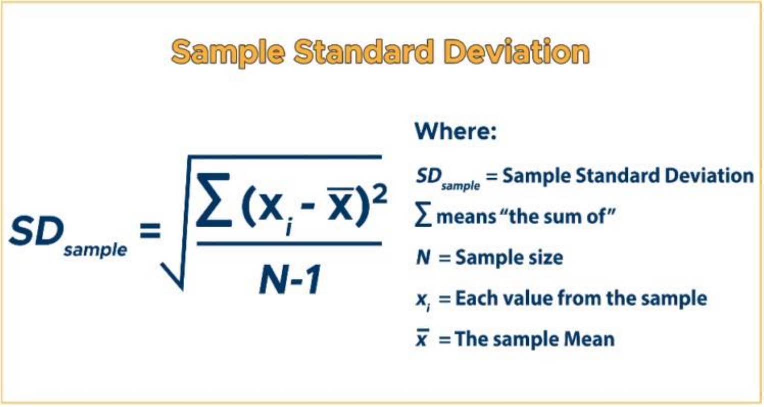
```
mean = np.mean(covid)
mean
```

```
day          16.000000
month         4.010989
year       2021.000000
cases      3661.010989
deaths       65.291941
dtype: float64
```

Standard Deviation

- The standard deviation measures a dataset's variation or dispersion. It tells you how spread out the data is around the mean.
- A higher standard deviation indicates greater variability in the data, while a lower standard deviation suggests that the data points are close to the mean.

The formula for Sample Standard Deviation (s):



Sample Standard Deviation

$$SD_{sample} = \sqrt{\frac{\sum (x_i - \overline{x})^2}{N-1}}$$

Where:

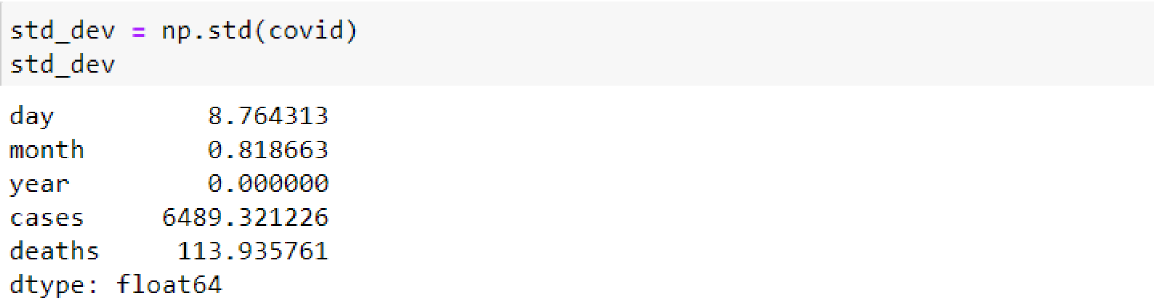
SD_{sample} = Sample Standard Deviation

$\sum$ means "the sum of"

N = Sample size

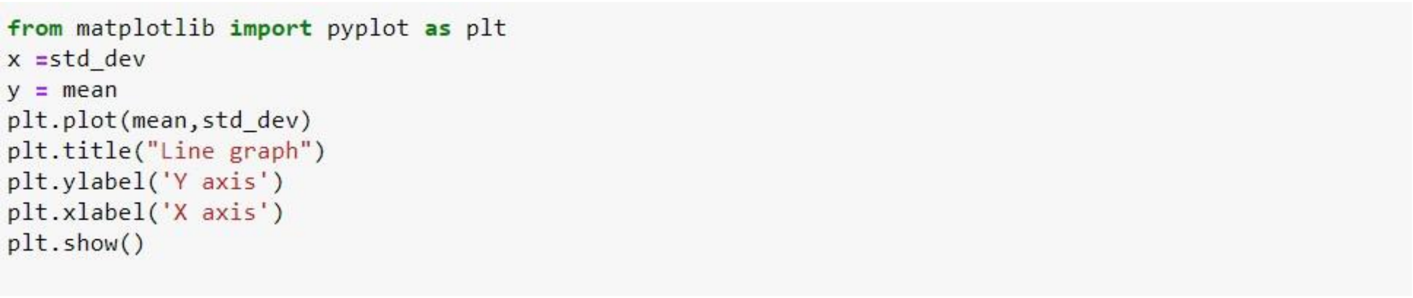
x_i = Each value from the sample

$\overline{x}$ = The sample Mean

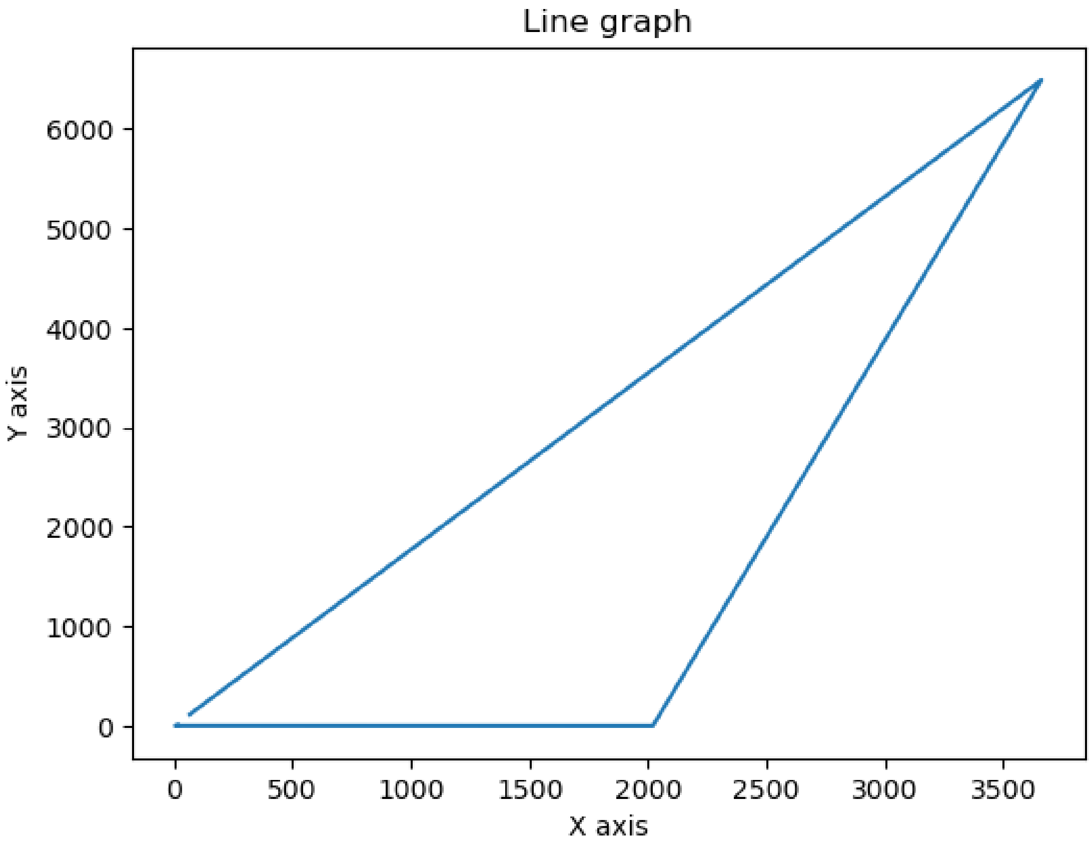


```
std_dev = np.std(covid)
std_dev
```

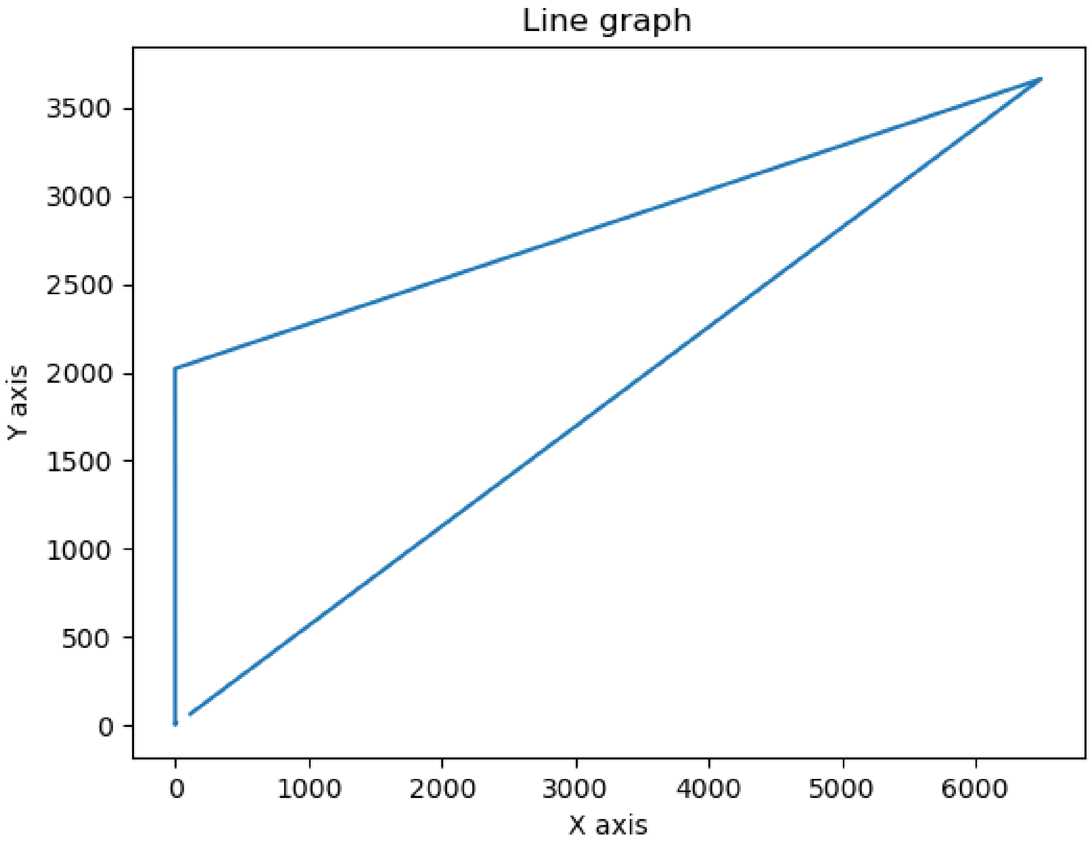
```
day          8.764313
month        0.818663
year         0.000000
cases     6489.321226
deaths     113.935761
dtype: float64
```



```
from matplotlib import pyplot as plt
x =std_dev
y = mean
plt.plot(mean,std_dev)
plt.title("Line graph")
plt.ylabel('Y axis')
plt.xlabel('X axis')
plt.show()
```



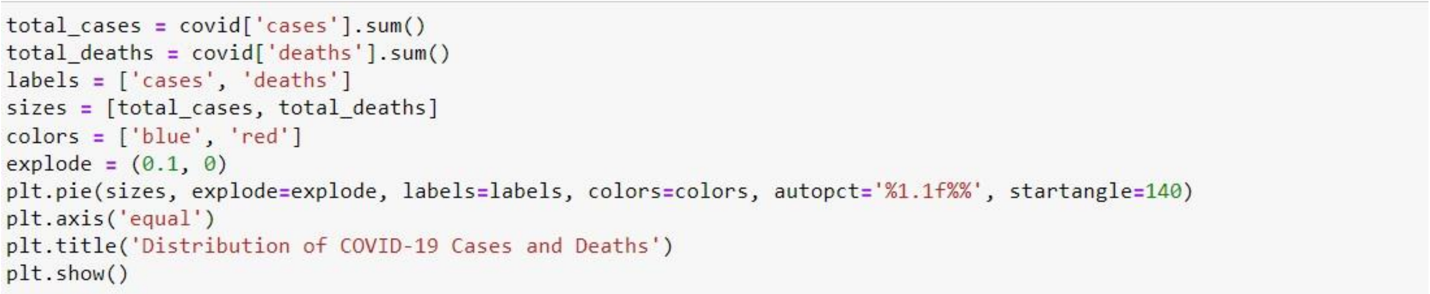
```
from matplotlib import pyplot as plt
x =std_dev
y = mean
plt.plot(std_dev,mean)
plt.title("Line graph")
plt.ylabel('Y axis')
plt.xlabel('X axis')
plt.show()
```



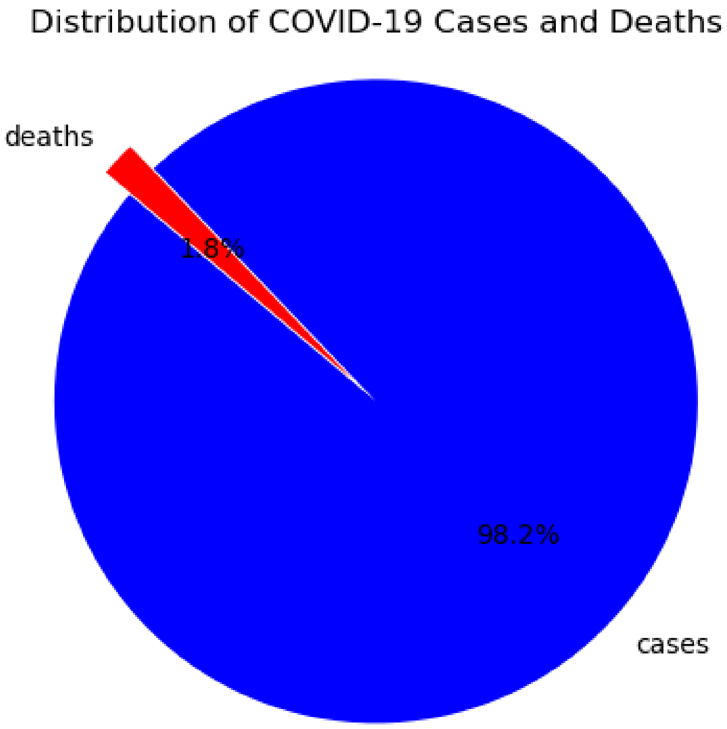
In the context of COVID-19 data, you can calculate the mean and standard deviation for cases and deaths to understand the typical values and the degree of variability in your dataset. For example, you can calculate:

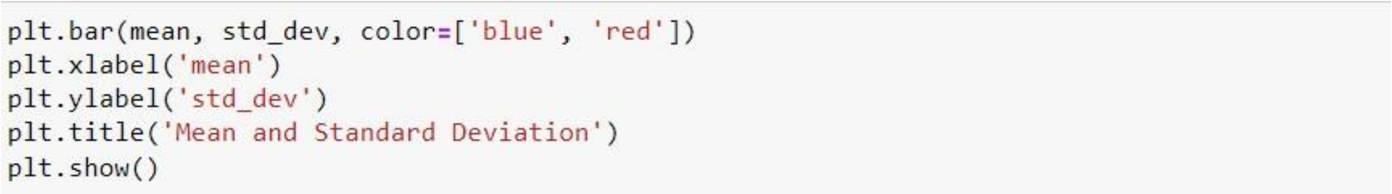
- The mean number of daily COVID-19 cases over a specific time frame (e.g., a week, a month) to understand the average rate of new infections.
- The mean number of associated deaths to understand the average daily mortality rate.
- The standard deviation of cases or deaths to assess how much the data varies from the mean, which can help in identifying days or regions with significant fluctuations.

These statistical measures can provide valuable insights into the spread and impact of COVID-19, helping you to make informed decisions and respond effectively to the pandemic.

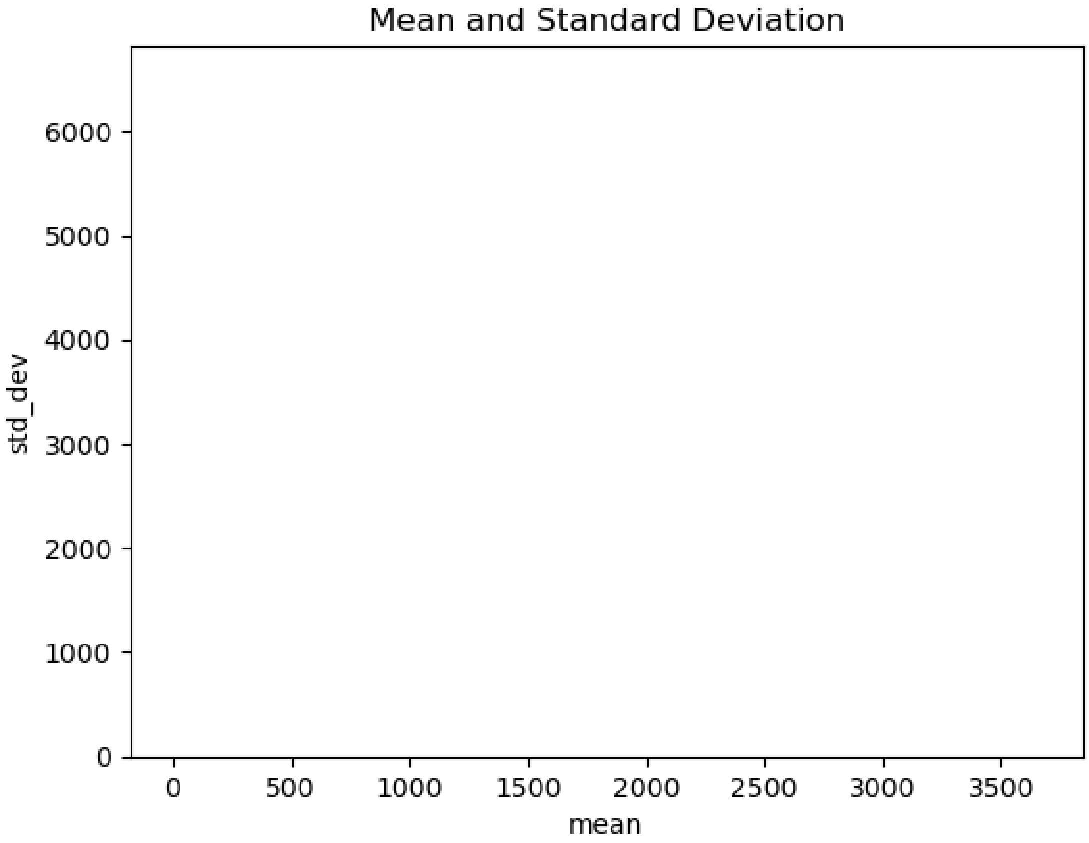


```
total_cases = covid['cases'].sum()
total_deaths = covid['deaths'].sum()
labels = ['cases', 'deaths']
sizes = [total_cases, total_deaths]
colors = ['blue', 'red']
explode = (0.1, 0)
plt.pie(sizes, explode=explode, labels=labels, colors=colors, autopct='%1.1f%%', startangle=140)
plt.axis('equal')
plt.title('Distribution of COVID-19 Cases and Deaths')
plt.show()
```





```
plt.bar(mean, std_dev, color=['blue', 'red'])
plt.xlabel('mean')
plt.ylabel('std_dev')
plt.title('Mean and Standard Deviation')
plt.show()
```



To continue building your analysis using IBM Cognos and derive insights from the data, it's important to understand the definition of correlation between cases and deaths. Correlation is a statistical measure that quantifies the degree to which two variables are related or associated. In the context of COVID-19 data, you can calculate the correlation between cases and deaths to determine how changes in one variable (cases) are related to changes in another variable (deaths).

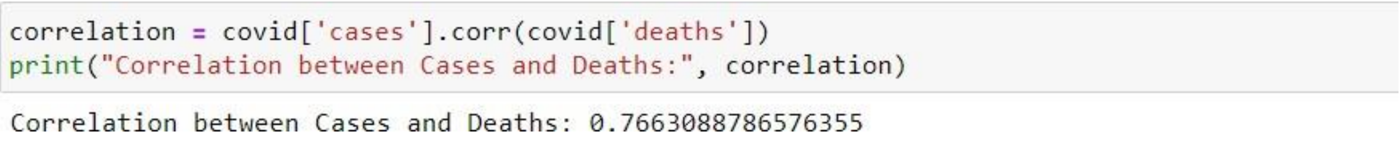
There are different ways to calculate the correlation coefficient, with Pearson correlation being one of the most common methods. The Pearson correlation coefficient (r) measures the linear relationship between two variables, and it ranges from -1 to 1, with the following interpretations:

- r = 1: Perfect positive correlation - As cases increase, deaths also increase in a linear fashion.
- r = 0: No correlation - There is no linear relationship between cases and deaths.
- r = -1: Perfect negative correlation - As cases increase, deaths decrease in a linear fashion.

Here's how you can calculate and interpret the correlation between COVID-19 cases and associated deaths using IBM Cognos:

Calculate Pearson Correlation:

Create a calculated field in Cognos to calculate the Pearson correlation coefficient (r)between cases and deaths. You can use the CORREL function in Cognos to do this.

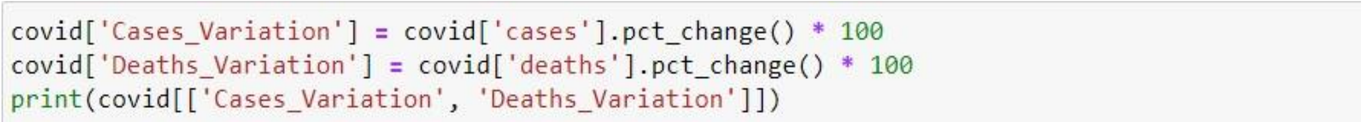


```
correlation = covid['cases'].corr(covid['deaths'])
print("Correlation between Cases and Deaths:", correlation)
```

```
Correlation between Cases and Deaths: 0.7663088786576355
```

Interpret the Correlation Coefficient:

- Once you've calculated the correlation coefficient, interpret it as follows:
- If r is close to 1: There is a strong positive correlation, indicating that as COVID-19cases increase, deaths tend to increase in a linear fashion.
- If r is close to -1: There is a strong negative correlation, suggesting that as casesincrease, deaths tend to decrease in a linear fashion.
- If r is close to 0: There is little to no linear relationship between cases and deaths.



```
covid['Cases_Variation'] = covid['cases'].pct_change() * 100
covid['Deaths_Variation'] = covid['deaths'].pct_change() * 100
print(covid[['Cases_Variation', 'Deaths_Variation']])
```

```
     Cases_Variation  Deaths_Variation
0                NaN               NaN
1          55.737705         20.000000
2          -5.614035         83.333333
3          18.773234        -63.636364
4         -36.619718        375.000000
...              ...               ...
2725      138.111647        -29.166667
2726       17.771346        -29.411765
2727       20.029491         16.666667
2728       -0.163800         35.714286
2729       26.968827          0.000000
```

[2730 rows x 2 columns]

Insights and Recommendations

Based on the innovative analysis:

- Identify countries with successful pandemic management strategies.
- Make data-driven decisions to allocate healthcare resources.
- Understand the impact of vaccination campaigns on case rates.
- Provide policymakers with actionable insights for better pandemic response.

Conclusion

Incorporating IBM Cognos into COVID-19 case analysis offers innovative solutions for understanding and responding to the pandemic. By segmenting data by time periods and countries, using predictive analytics, and creating informative visualizations, we enhance our ability to address the ongoing challenges posed by COVID-19.

References

Include references to data sources, research papers, and relevant publications used in the analysis.

This document outlines an innovative approach to COVID-19 case analysis using IBM Cognos. By leveraging the power of this analytics tool and incorporating data segmentation, we can gain deeper insights into the pandemic's dynamics, improving our response strategies and decision-making processes.

- ❖ Dataset Link: COVID-19 Cases Dataset

Software used: Jupiter notebook (or) vs code.

Language used: python.