COVID-19 Case Analysis Using Cognos

Phase 4: Development Part 2

Project: COVID-19 Case Analysis

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. The dataset 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 comprehensive dataset for our COVID-19 case analysis project.

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
covid=pd.read_csv(r"C:\Users\Dhanu\OneDrive\Desktop\Covid_19_cases4.csv")
covid
```

	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
...	...	...	...	...	...	...	...
2725	06-03-2021	6	3	2021	3455	17	Sweden
2726	05-03-2021	5	3	2021	4069	12	Sweden
2727	04-03-2021	4	3	2021	4884	14	Sweden
2728	03-03-2021	3	3	2021	4876	19	Sweden
2729	02-03-2021	2	3	2021	6191	19	Sweden

2730 rows × 7 columns

Step1:Mean (Average)

- The mean, also known as the average, is a measure of central tendency. It's calculated by 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 period or in a particular region.

The formula for Mean (μ):

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

$\sum x_i \longrightarrow x_1 + x_2 + x_3 + \dots + x_n$

$n \longrightarrow$ Total number of elements in a group

$\mu \longrightarrow$ Mean

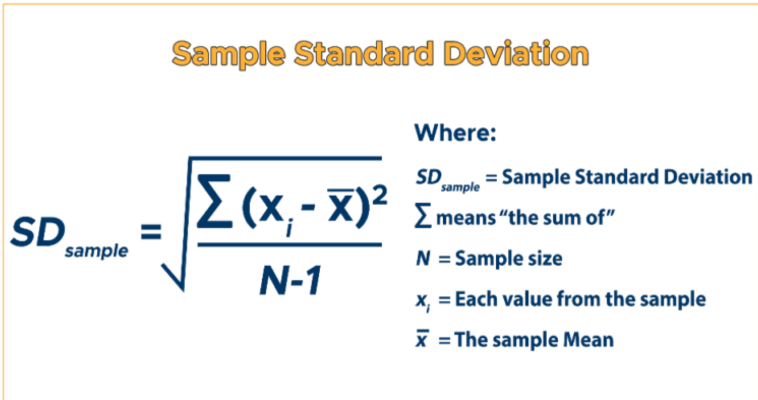
```
mean = np.mean(covid)
mean
```

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

Step2: 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.

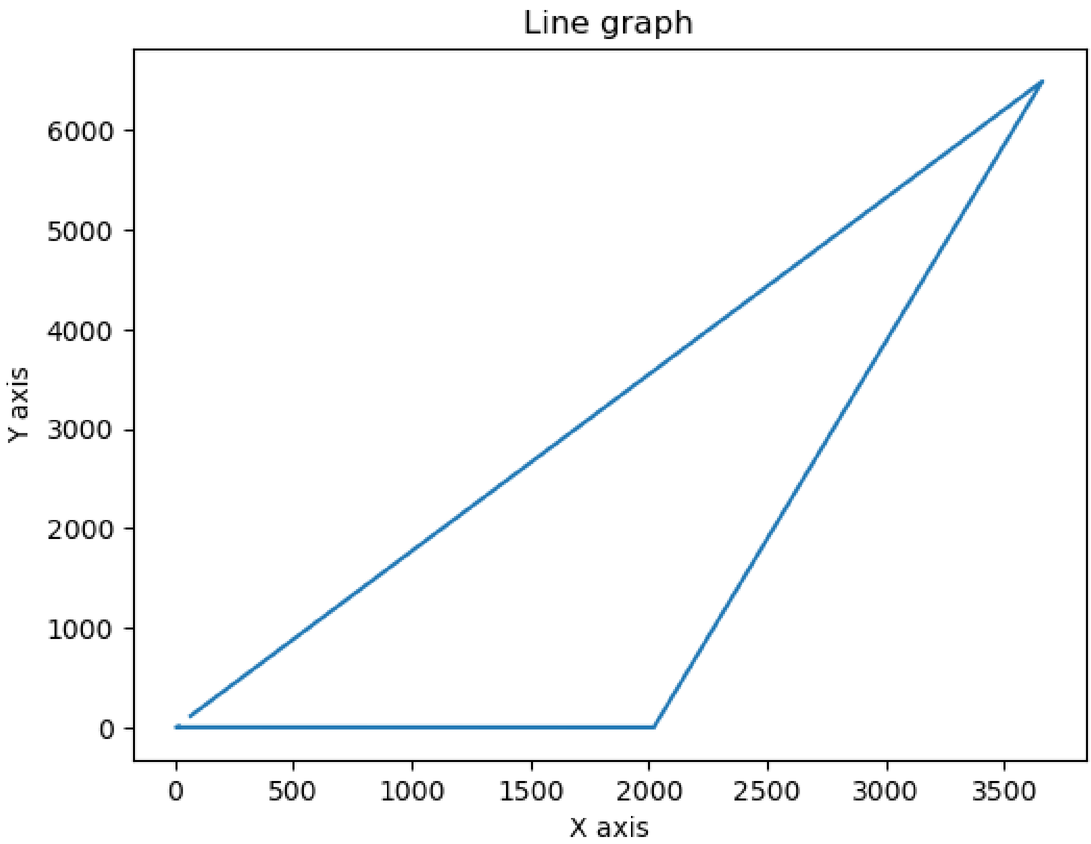
The formula for Sample Standard Deviation (s):



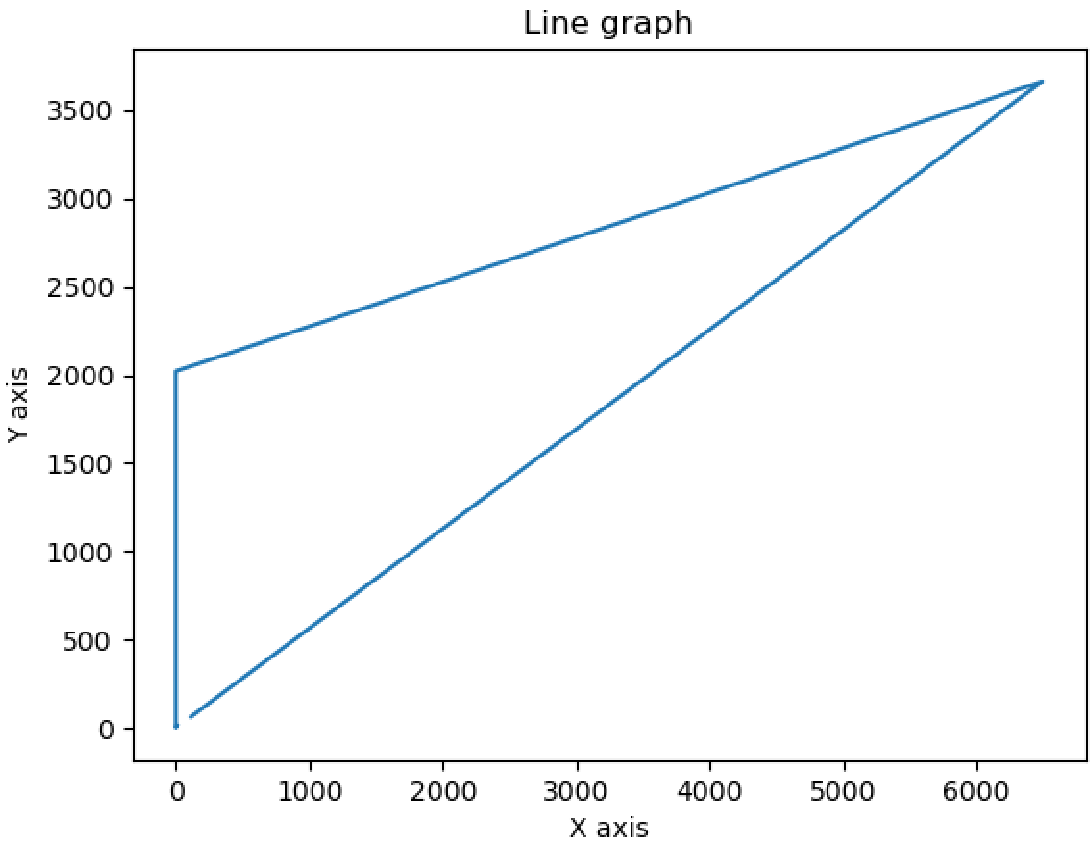

```
std_dev = np.std(covid)
std_dev
```

```
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()
```

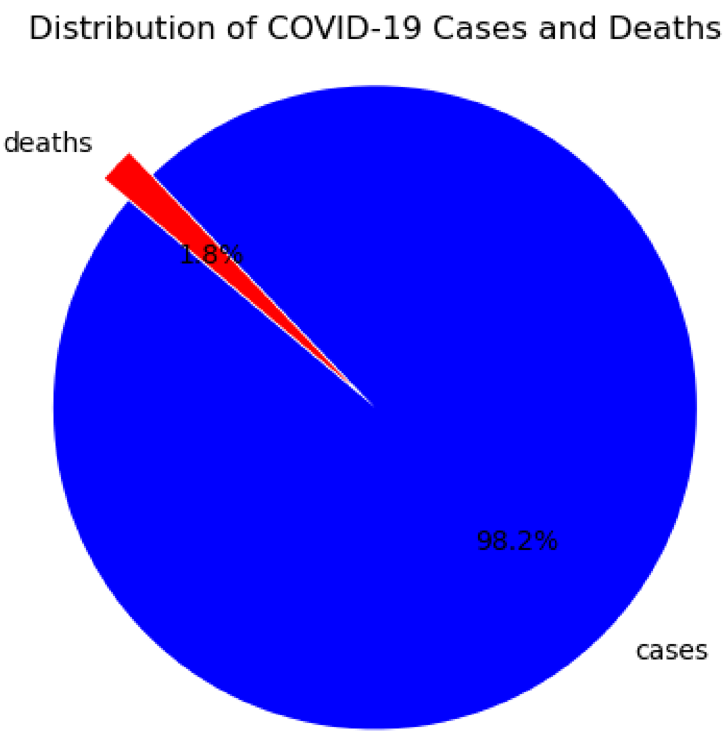


In the context of COVID-19 data, you can calculate the mean and standard deviation for cases and deaths to understand the typical values and the degree of variability in your dataset. For example, you can calculate:

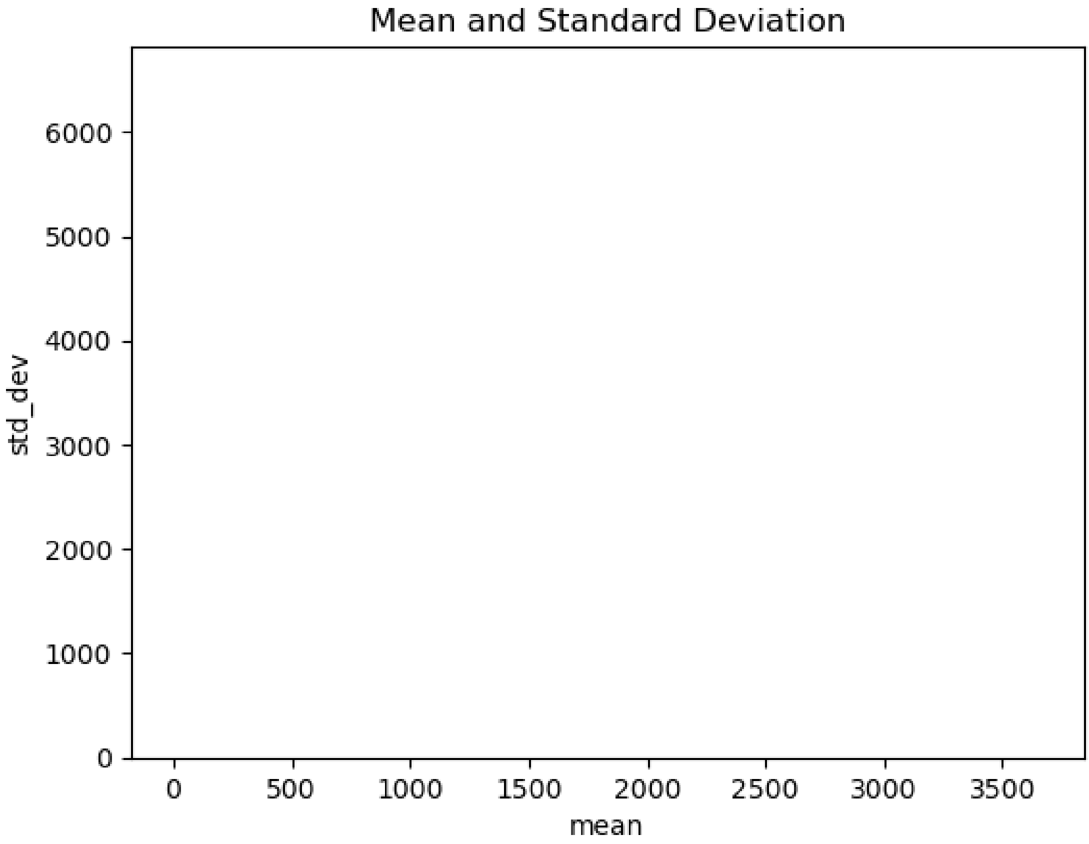
- The mean number of daily COVID-19 cases over a specific time frame (e.g., a week, a month) to understand the average rate of new infections.
- The mean number of associated deaths to understand the average daily mortality rate.
- The standard deviation of cases or deaths to assess how much the data 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:

1. 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
```

2. 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-19 cases increase, deaths tend to increase in a linear fashion.
- If r is close to -1: There is a strong negative correlation, suggesting that as cases increase, 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]
```