COVID-19
PROJECT REPORT

Submitted by:

G.SATHWIK	A21126551018
J. TARUN	A21126551023
K.SREEJA	A21126551028
P.TEJESWARA RAO	A21126551040
S. LOCHAN KUMAR	A21126551057

In fulfillment of the project in

Computer Science & Engineering (AI & ML, DS)

ANIL NEERUKONDA INSTITUTE OF TECHNOLOGY AND SCIENCES

(Affiliated to Andhra University)

SANGIVALASA

VISAKHAPATNAM - 531162 2021-2025

DEPARTMENT OF

COMPUTER SCIENCE AND ENGINEERING (AI, ML&DS)

ANIL NEERUKONDA INSTITUTE OF TECHNOLOGY AND SCIENCES
(Affiliated to Andhra University)
SANGIVALASA, VISAKHAPATNAM – 531162
2021-2025

BONAFIDE CERTIFICATE

This is to certify that this Project Report of "COVID-19" is the bonafide work of **G.SATHWIK(A21126551018), J.TARUN(A21126551023), K.SREEJA(A21126551028), P.TEJESWARA RAO (A21126551040), S.LOCHAN KUMAR(A21126551057)** of ¾ CSD carried out the project work under my Supervision.

Dr. Appala Srinuvasu Muttipati	**Dr.K.S.Deepthi**
(Associate Professor)	Head of The
Department of CSE(AI,ML&DS)	Department of CSE(AI,ML&DS)
ANITS	ANITS

ACKNOWLEDGEMENT

An endeavor over a long period can be successful with the advice and support of many well- wishers. We take this opportunity to express our gratitude and appreciation to all of them.

We owe our tributes to Dr .K. S. Deepthi , Head of the Department, Computer Science & Engineering (AI&ML,DS), ANITS, for his valuable support and guidance during the period of the PROJECT.

We wish to express our sincere thanks and gratitude to our Curators srinuvasu sir who helped in stimulating discussions, and for guiding us throughout the project. We express our warm and sincere thanks for the encouragement, untiring guidance, and confidence they had shown in us. We also thank all the staff members of the Computer Science & Engineering (AI&ML,DS) department for their valuable advices. We also thank supporting staff for providing resources as and whenrequired.

G.SATHWIK	A21126551018
J. TARUN	A21126551023
K.SREEJA	A21126551028
P.TEJESWARA RAO	A21126551040
S. LOCHAN KUMAR	A21126551057

DECLARATION

We, G.SATHWIK(A21126551018),J.TARUN(A21126551023), K.SREEJA(A21126551028), P.TEJESWARA RAO (A21126551040),S.LOCHAN KUMAR (A21126551057) Solemnly declare that this minor project on Student performance analysis by using“Data Science with Python” is our original work, and we further declare that we have properly referred all outsourcing of materials used in this project and nothing is confidential in this project. We take the responsibility for all legal and ethical requirementsregarding this project.

G.SATHWIK

(A21126551017)

J.TARUN

(A21126551023)

K.SREEJA

(A21126551031)

P.TEJESWARARAO

(A21126551040)

S. LOCHANKUMAR

(A21126551057)

Table of Contents

1.INTRODUCTION

The exploration of COVID-19 datasets provides a compelling opportunity to unravel the intricate dynamics of the pandemic through data-driven insights. In this exploratory data analysis (EDA), we delve into a comprehensive dataset, aiming to extract meaningful patterns, trends, and correlations related to the spread and impact of the virus. As the world continues to grapple with the far-reaching consequences of COVID-19, a thorough examination of this dataset becomes a crucial step in understanding the factors influencing transmission rates, identifying vulnerable populations, and informing evidence-based strategies for public health interventions. This EDA not only serves as a lens into the past but also lays the foundation for proactive measures to shape future responses to global health crises.

Our analysis encompasses diverse dimensions, including temporal trends, geographical variations, and demographic impacts of the virus. Visualizations such as heatmaps, time series plots, and geographic maps offer a nuanced perspective on how the virus has evolved over time and across regions. Through statistical measures, we aim to identify critical points of interest, such as peak infection periods and disparities in healthcare outcomes. Furthermore, demographic breakdowns shed light on the differential impact on various age groups and socio-economic strata, guiding targeted interventions for the most vulnerable communities.

This exploratory data analysis (EDA) delves into a comprehensive COVID-19 dataset, providing a nuanced understanding of the pandemic's evolution, impacts, and potential avenues for intervention. Through a multidimensional exploration, including temporal, geographic, and demographic lenses, we unveil critical insights into the virus's behavior. Visualizations such as heatmaps, time series plots, and demographic breakdowns offer a detailed perspective on transmission patterns, regional disparities, and the differential impact on diverse populations. Statistical measures identify peak infection periods and guide targeted interventions for vulnerable communities. This EDA not only informs current public health assessments but lays the groundwork for proactive decision-making in future global health crises. As the world navigates the ongoing challenges of COVID-19, the insights gleaned from this analysis underscore the pivotal role of data-driven approaches in shaping resilient and adaptive public health strategies.

This introduction provides a foundation for further exploration of the field of data visualization, including its ideas, methods, resources, and the importance of using visual aids to draw insightful conclusions from data.

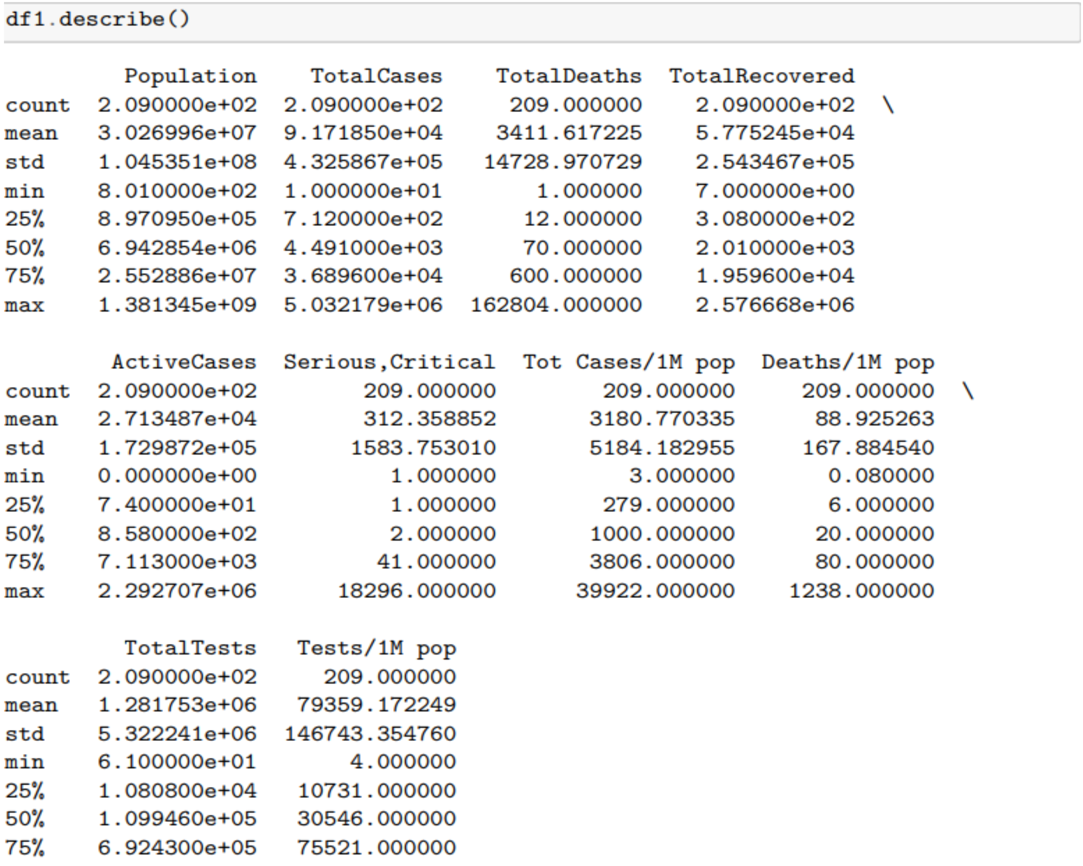
2.DATA SET WITH DESCRIPTION

2.1 ABOUT DATASET

The data set used in this project was taken from the kaggle platform .
It is created for learning purposes of the Covid-19.
link- https://www.kaggle.com/datasets/imdevskp/corona-virus-report/download?datasetVersionNumber=166

2.2 DATASET DESCRIPTION

Number of rows:209
Number of columns:16
Data format: comma separated values(csv)



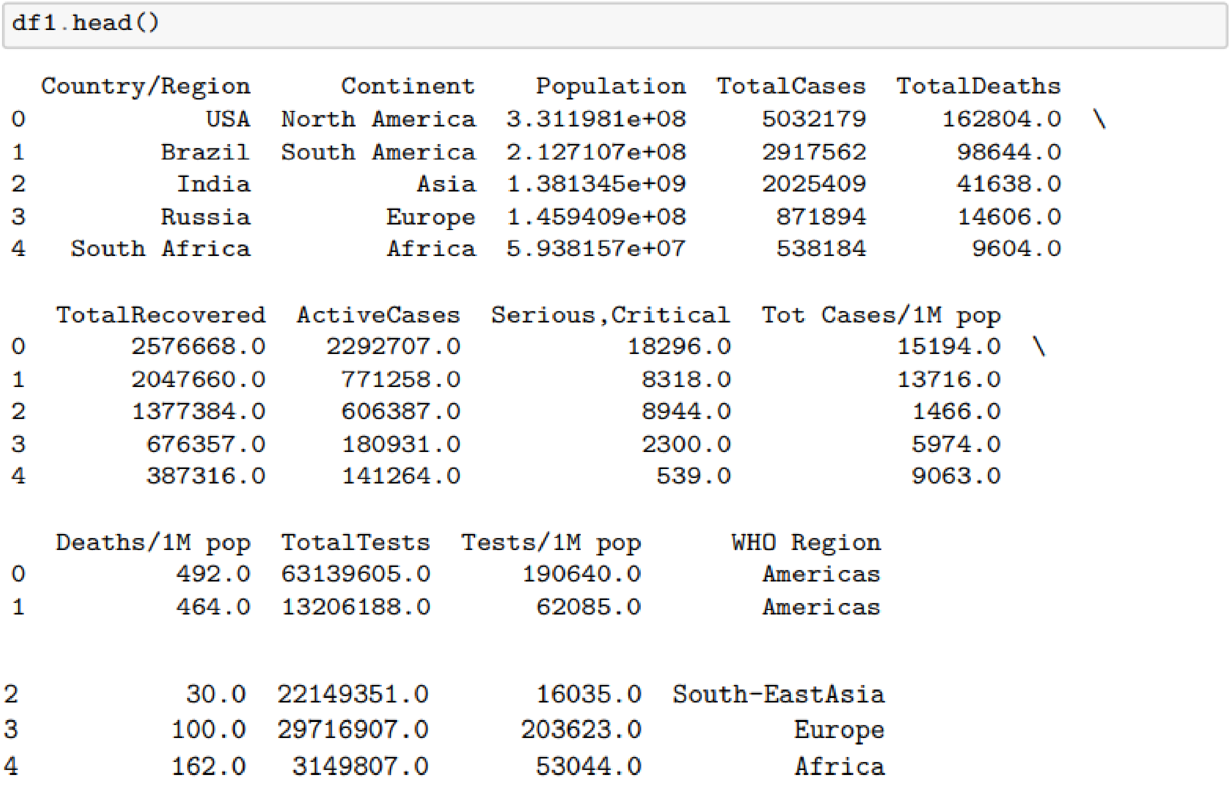
```
df1.describe()
```

```
         Population   TotalCases    TotalDeaths  TotalRecovered  \
count  2.090000e+02  2.090000e+02     209.000000    2.090000e+02
mean   3.026996e+07  9.171850e+04    3411.617225    5.775245e+04
std    1.045351e+08  4.325867e+05   14728.970729    2.543467e+05
min    8.010000e+02  1.000000e+01       1.000000    7.000000e+00
25%    8.970950e+05  7.120000e+02      12.000000    3.080000e+02
50%    6.942854e+06  4.491000e+03      70.000000    2.010000e+03
75%    2.552886e+07  3.689600e+04     600.000000    1.959600e+04
max    1.381345e+09  5.032179e+06  162804.000000    2.576668e+06

        ActiveCases  Serious,Critical  Tot Cases/1M pop  Deaths/1M pop  \
count  2.090000e+02        209.000000        209.000000     209.000000
mean   2.713487e+04        312.358852       3180.770335      88.925263
std    1.729872e+05       1583.753010       5184.182955     167.884540
min    0.000000e+00          1.000000          3.000000       0.080000
25%    7.400000e+01          1.000000        279.000000       6.000000
50%    8.580000e+02          2.000000       1000.000000      20.000000
75%    7.113000e+03         41.000000       3806.000000      80.000000
max    2.292707e+06      18296.000000      39922.000000    1238.000000

         TotalTests   Tests/1M pop
count  2.090000e+02     209.000000
mean   1.281753e+06   79359.172249
std    5.322241e+06  146743.354760
min    6.100000e+01       4.000000
25%    1.080800e+04   10731.000000
50%    1.099460e+05   30546.000000
75%    6.924300e+05   75521.000000
```

2.3 DATA SCHEMA

ATTRIBUTES	DATA TYPE	DESCRIPTION
Country/Region	String	Indicates the Name of the Country /Region
Continent	String	Indicates the Name of the continent
Population	Float	Total population of the specified country
TotalCases	Int	Count of Total no of covid cases
NewCases	Float	No of new cases occurred
TotalDeaths	Float	Count of Total no of Deaths
NewDeaths	Float	No of new Deaths occurred
TotalRecovered	Float	Specifies the count of all recovered cases
NewRecovered	Float	Count of newly recovered cases
ActiveCases	Float	Total no of cases currently present till date
Serious,Critical	Float	Total no of cases in serious and critical condition
Tot Cases/1M pop	Float	Total cases per 1 million population
Deaths/1M pop	Float	No of deaths per 1 million population
TotalTests	Float	No of tests conducted during pandemic
Tests/1M pop	Float	Tests per 1 million population
WHO Region	Float	World Health Organization Region

2.4 SAMPLE DATA



```
df1.head()
```

```
  Country/Region      Continent    Population  TotalCases  TotalDeaths  \
0            USA  North America  3.311981e+08     5032179     162804.0   
1         Brazil  South America  2.127107e+08     2917562      98644.0   
2          India           Asia  1.381345e+09     2025409      41638.0   
3         Russia         Europe  1.459409e+08      871894      14606.0   
4   South Africa         Africa  5.938157e+07      538184       9604.0   

   TotalRecovered  ActiveCases  Serious,Critical  Tot Cases/1M pop  \
0       2576668.0    2292707.0           18296.0           15194.0   
1       2047660.0     771258.0            8318.0           13716.0   
2       1377384.0     606387.0            8944.0            1466.0   
3        676357.0     180931.0            2300.0            5974.0   
4        387316.0     141264.0             539.0            9063.0   

   Deaths/1M pop  TotalTests  Tests/1M pop       WHO Region  
0          492.0  63139605.0      190640.0         Americas  
1          464.0  13206188.0       62085.0         Americas  


2           30.0  22149351.0       16035.0  South-EastAsia
3          100.0  29716907.0      203623.0          Europe
4          162.0   3149807.0       53044.0          Africa
```

3.PYTHON PACKAGES

3.1 PANDAS

[1] https://pandas.pydata.org/

Pandas were initially developed by Wes McKinney in 2008 while he was working at AQR Capital Management. He convinced the AQR to allow him to open source the Pandas. Another AQR employee, Chang She, joined as the second major contributor to the library in 2012. Over time many versions of pandas have been released.Pandas is a library in Python that is made mainly for working with relational or labeled data both easily and intuitively. It provides various data structures and operations for manipulating numerical data and time series.

3.2 MATPLOTLIB

[2] https://matplotlib.org/

Goal: Matplotlib is a feature-rich Python data visualization package.

For the purpose of visualizing data, it offers a large selection of excellent, fully customisable plots and charts.

Important characteristics:

Different Plot kinds: Matplotlib allows for a number of plot kinds, such as line, scatter, and bar charts.

Histograms, charts, and more.

Customization: You have complete control over the colors, styles, labels, and other elements of your plots,comments.

Plots can be exported in a variety of formats (such as PNG and PDF) for use in reports spoken or written works.

3.3 SEABORN

[3] https://seaborn.pydata.org/

Goal : Its main objective is to offer a sophisticated interface for making eye - catching and educational statistical visuals.

Important characteristics:

Elegant Plots: Seaborn includes default themes and color schemes that are visually appealing, making plots that are visually appealing is simple.

Statistical Plots : It provides tools for producing specific statistical illustrations , such as box

violin plots, pair plots, and storylines.

Simple to Use : Seaborn makes it easier to create intricate statistics charts and visualizations.

4.DIFFERENT GRAPHICAL VISUALS

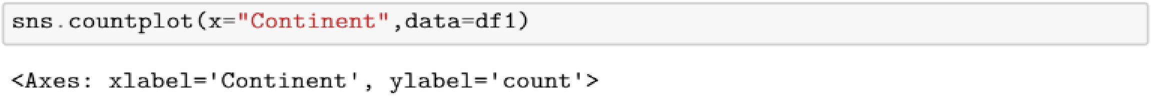
4.1 COUNTPLOT

DESCRIPTION

A count plot is a type of data visualization that displays the number of occurrences of categorical data in a dataset. It is particularly useful for understanding the distribution of categorical variables and identifying patterns or trends within them.

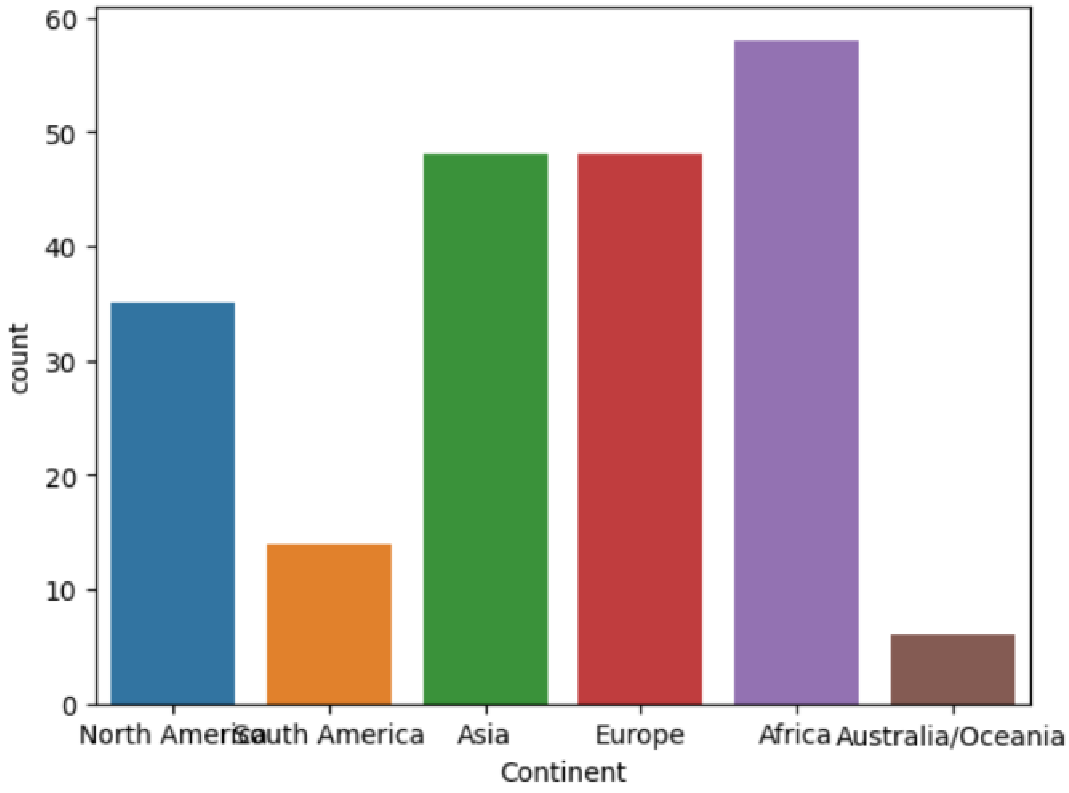
CODE AND OUTPUT

Here Count Plot is drawn for Continent and the Total count of cases.We can identify the total count in each respective continent from this count plot



```
sns.countplot(x="Continent",data=df1)
```

```
<Axes: xlabel='Continent', ylabel='count'>
```



A count plot is a compelling and intuitive data visualization technique employed in statistical analysis and data exploration. At its core, this graphical representation serves the purpose of illustrating the distribution of categorical variables within a dataset. While it might seem deceptively simple with its bars and counts, a count plot offers a wealth of information and insights, making it a valuable tool in the hands of data analysts and researchers.

Understanding the Basics

At a fundamental level, a count plot operates by displaying the frequency or count of each category in a categorical variable. Imagine a dataset containing information about the types of fruits in a grocery store. A count plot could visually represent the number of occurrences of each fruit, using bars to indicate the count. For instance, if there are 20 apples, 15 oranges, and 10 bananas, the count plot would have bars corresponding to each fruit, with heights proportional to their counts.

The simplicity of a count plot makes it an excellent choice for initial data exploration. Its straightforward design facilitates a quick grasp of the distribution of categories, enabling analysts to identify dominant and minority groups. This clarity is particularly useful when dealing with large datasets or when attempting to communicate findings to a non-technical audience.

Beyond its aesthetic appeal, a count plot provides a visual summary of categorical data, aiding in the detection of patterns or anomalies. It serves as a starting point for more in-depth analyses, offering a snapshot of the data's composition. When multiple count plots are used in conjunction, they can reveal intricate relationships and dependencies between different categorical variables, paving the way for more nuanced interpretations.

Applications and Best Practices

The applications of count plots extend across various domains, from business analytics to scientific research. In marketing, for example, a count plot could depict the distribution of customer preferences among different product categories. Researchers studying survey data might utilize count plots to visualize responses to multiple-choice questions, gaining insights into participant preferences or opinions.

One notable strength of count plots is their adaptability to different types of categorical variables. Whether dealing with nominal variables without a specific order or ordinal variables with a defined sequence, count plots can effectively represent the distribution. The versatility of this visualization tool makes it a staple in the data analyst's toolkit.

However, like any analytical tool, using count plots requires an understanding of best practices. Care must be taken to choose an appropriate color scheme and labeling to ensure clarity and avoid misinterpretation. Additionally, when dealing with large datasets, aggregating or grouping categories may be necessary to prevent clutter and enhance readability.

In conclusion, the count plot stands as a powerful and accessible means of exploring and communicating categorical data. Its simplicity belies its effectiveness, providing a visual narrative that lays the foundation for more intricate analyses. Whether used in isolation for a quick overview or in combination with other visualization techniques, the count plot remains a valuable asset in the realm of data analysis.

4.2 BOXPLOT

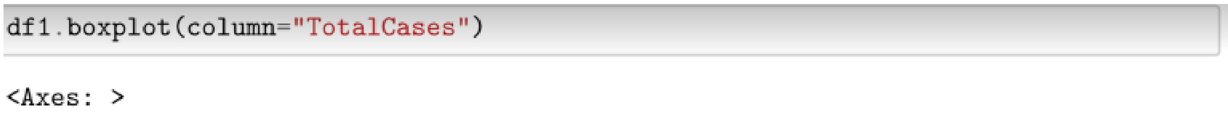
DESCRIPTION

A box plot, also known as a box-and-whisker plot, is a statistical visualization that provides a summary of the distribution of a dataset. It displays the median, quartiles, and potential outliers of the data. The plot consists of a rectangular "box" that represents the interquartile range (IQR) and "whiskers" that extend from the box to show the range of the data. Outliers may be plotted as individual points.

4.2.1 Box Plot on Total Cases

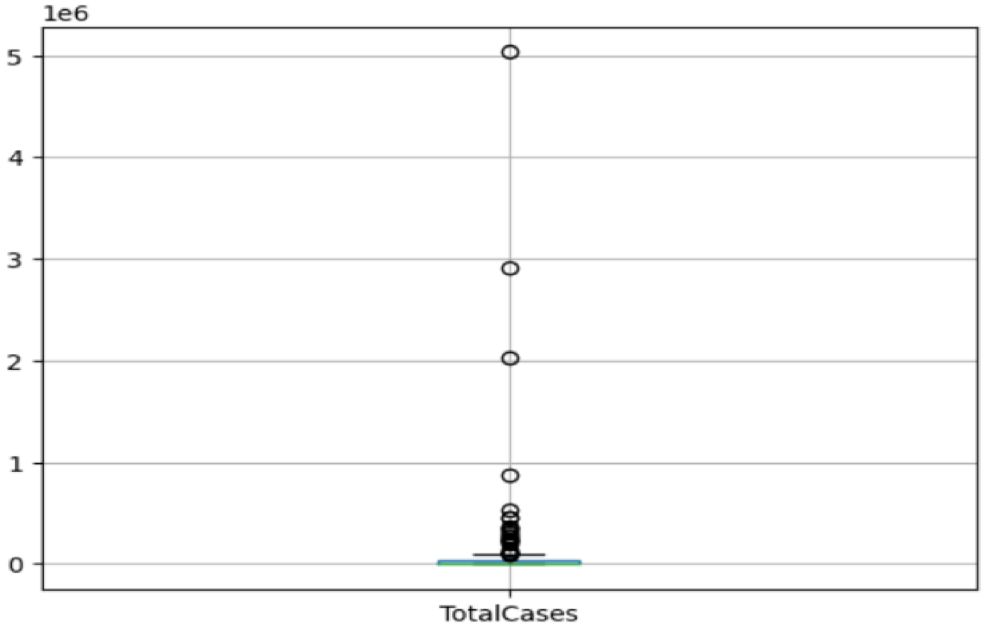
CODE AND OUTPUT

The Box Plot is drawn for the Total Cases of Covid-19 ,that helps us to get a proper knowledge on the count. We have Considered total cases on x-axis.



```
df1.boxplot(column="TotalCases")
```

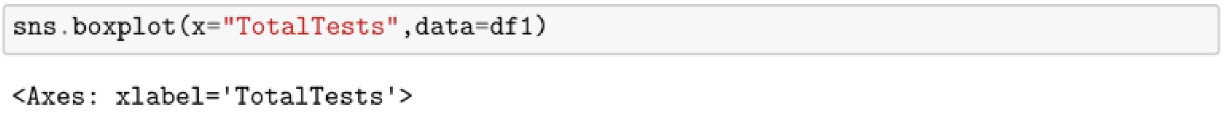
```
<Axes: >
```



4.2.2 BoxPlot on Total Tests

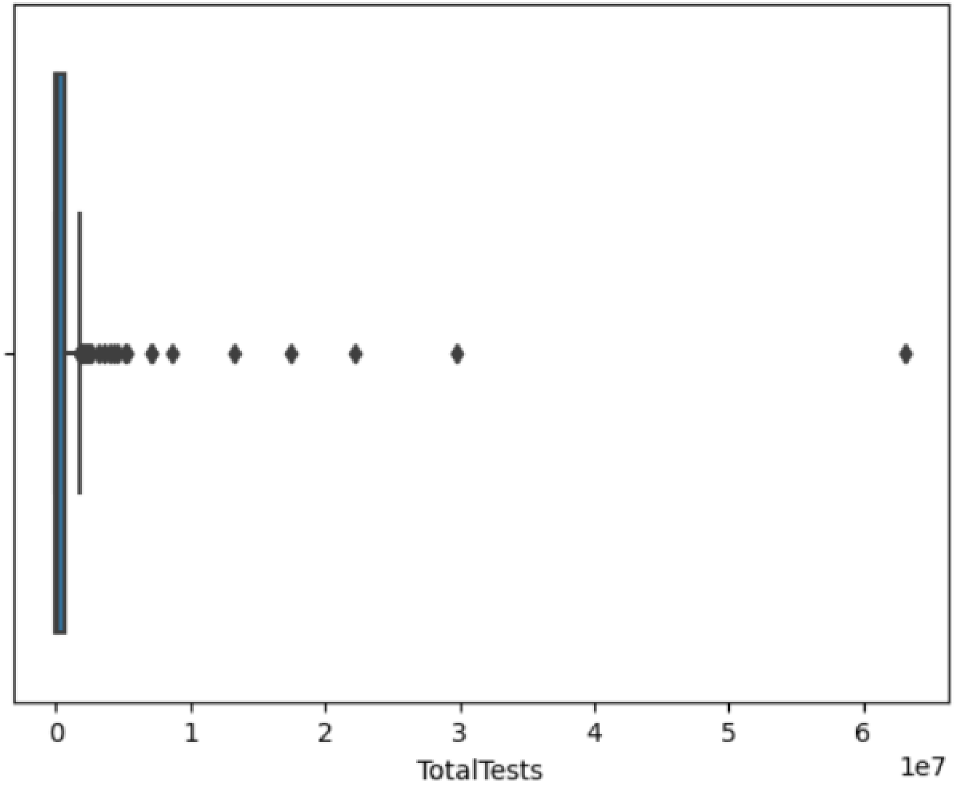
CODE AND OUTPUT

The Box Plot is drawn for the Total Cases of Covid-19 , that helps us to get a proper knowledge on the count. We have Considered total cases on x-axis.



```
sns.boxplot(x="TotalTests",data=df1)
```

```
<Axes: xlabel='TotalTests'>
```



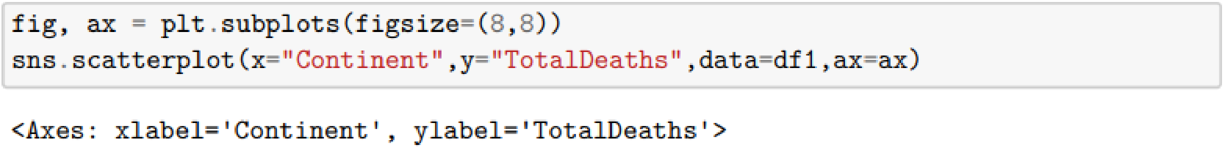
4.3 SCATTER PLOT

DESCRIPTION

A scatter plot is a type of data visualization that displays individual data points on a two-dimensional graph. Each point on the graph represents the values of two variables, one plotted along the x-axis and the other along the y-axis. Scatter plots are useful for visualizing the relationship or correlation between two continuous variables.

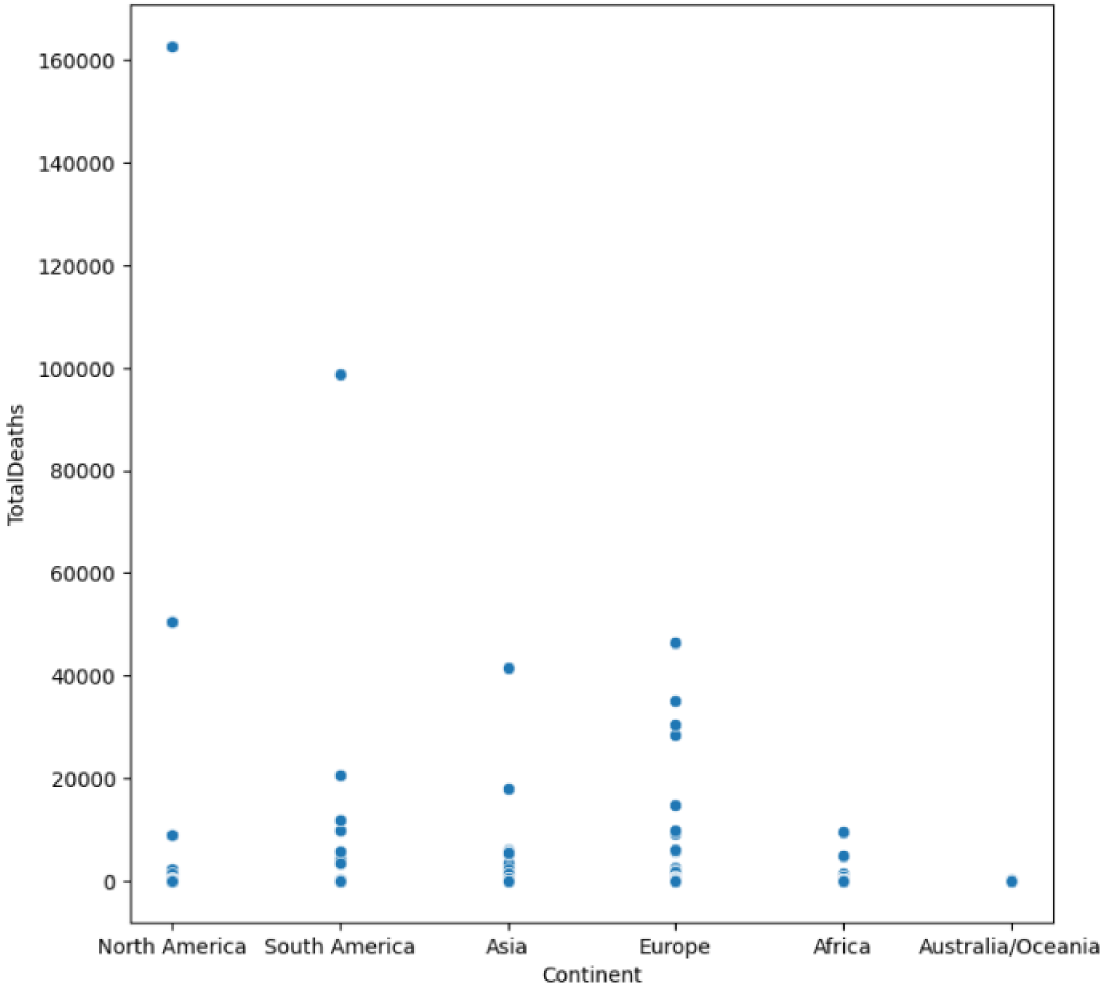
CODE AND OUTPUT

Helps us to understand the relation between continent and no of deaths Taken place.on x-axis we have taken continent name and on y-axis no of total deaths respectively



```
fig, ax = plt.subplots(figsize=(8,8))
sns.scatterplot(x="Continent",y="TotalDeaths",data=df1,ax=ax)
```

```
<Axes: xlabel='Continent', ylabel='TotalDeaths'>
```



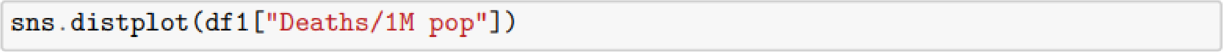
4.4 DISTPLOT

DESCRIPTION

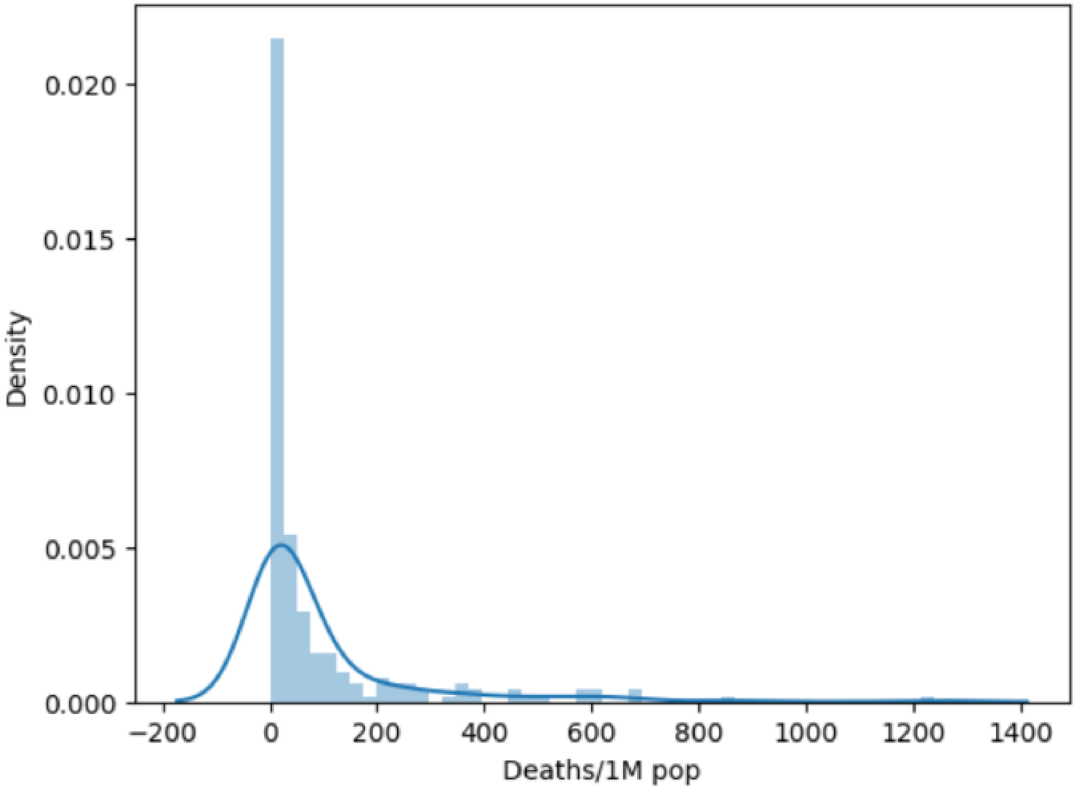
Distplot stands for "distribution plot," and it is a term often associated with the Seaborn library in Python. A distplot in Seaborn is a data visualization that combines the features of a histogram with a kernel density estimate (KDE). It provides a way to visualize the distribution of a univariate (single variable) dataset.

CODE AND OUTPUT

This plot is used to compare the relationship between the death/1   million population and rate of density.



```
sns.distplot(df1["Deaths/1M pop"])
```



```
<Axes: xlabel='Deaths/1M pop', ylabel='Density'>
```

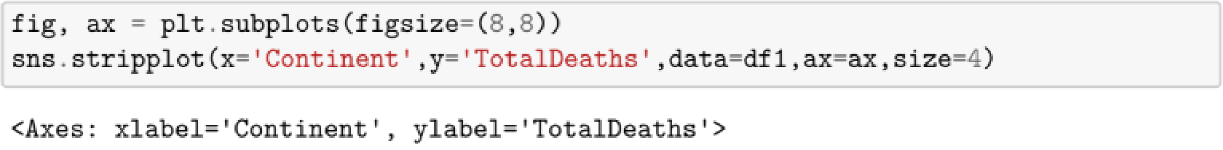
4.5 STRIPPLOT

DESCRIPTION

A Stripplot is a type of data visualization that displays individual data points along a one-dimensional axis. It is particularly useful for visualizing the distribution of a dataset and showing the density of points at different values. Strip plots are often used when dealing with categorical data or when there is an interest in understanding the distribution of a single continuous variable.

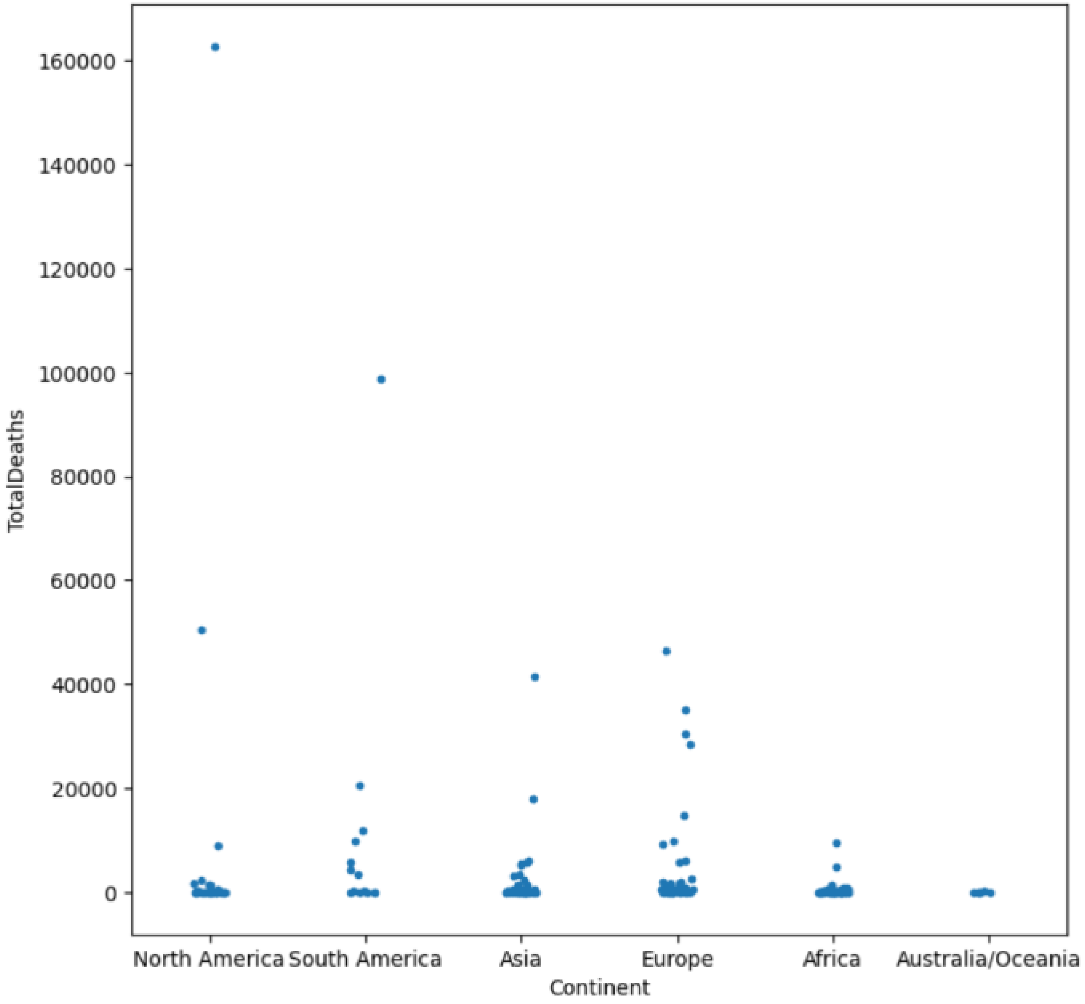
CODE AND OUTPUT

It gives us a clear view and understand about the total deaths that took place due to the Covid-19 in each continent.



```
fig, ax = plt.subplots(figsize=(8,8))
sns.stripplot(x='Continent',y='TotalDeaths',data=df1,ax=ax,size=4)
```

```
<Axes: xlabel='Continent', ylabel='TotalDeaths'>
```

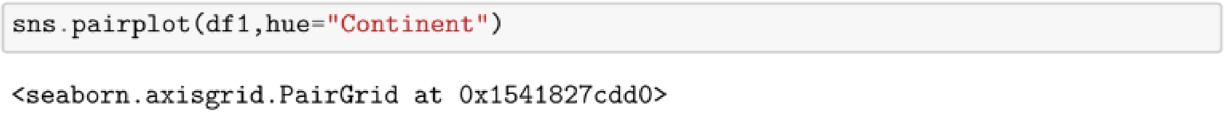


4.6 PAIRPLOT

DESCRIPTION

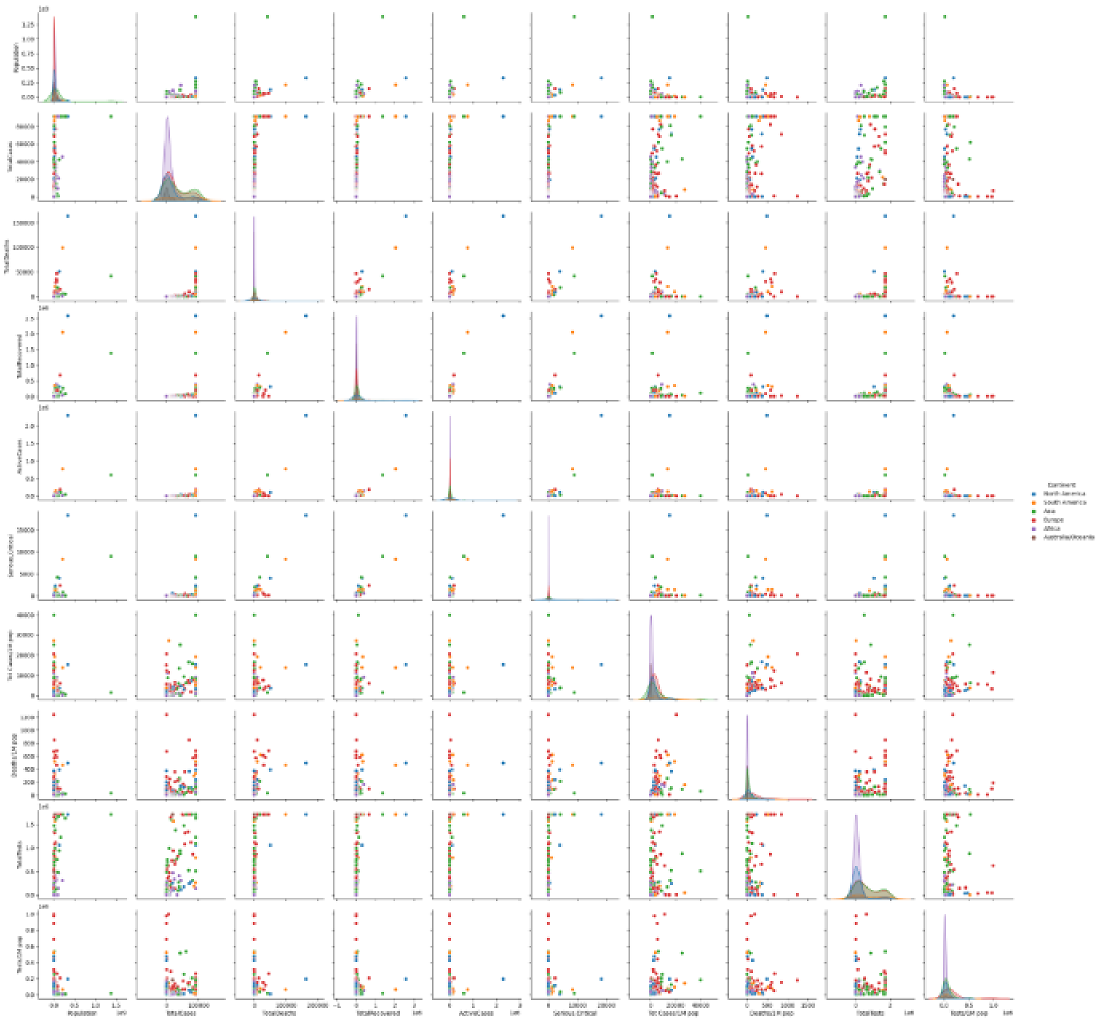
A pair plot, often created using Seaborn in Python, is a matrix of scatterplots and histograms that allows you to visualize the relationships between multiple variables in a dataset. It is particularly useful for exploring pairwise relationships and identifying patterns or trends. Each combination of variables is represented in a scatterplot, and histograms along the diagonal show the univariate distribution of each variable.

CODE AND OUTPUT



```
sns.pairplot(df1,hue="Continent")
```

```
<seaborn.axisgrid.PairGrid at 0x1541827cdd0>
```

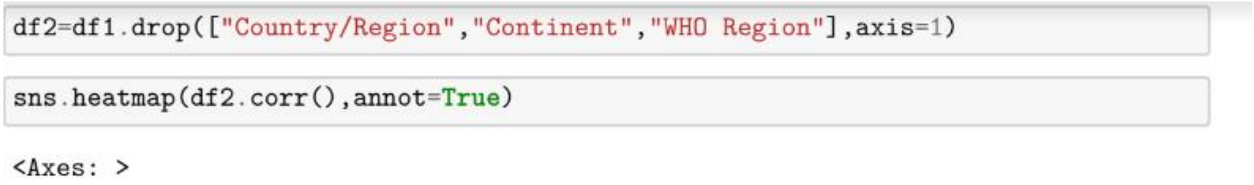


4.7 HEATMAP

DESCRIPTION

A heatmap is a graphical representation of data where values in a matrix are represented as colors. Heatmaps are commonly used to visualize complex data sets, such as the correlation matrix of variables or the distribution of values in a two-dimensional space. The colors in a heatmap represent the magnitude of the values, making it easier to identify patterns and trends in the data.

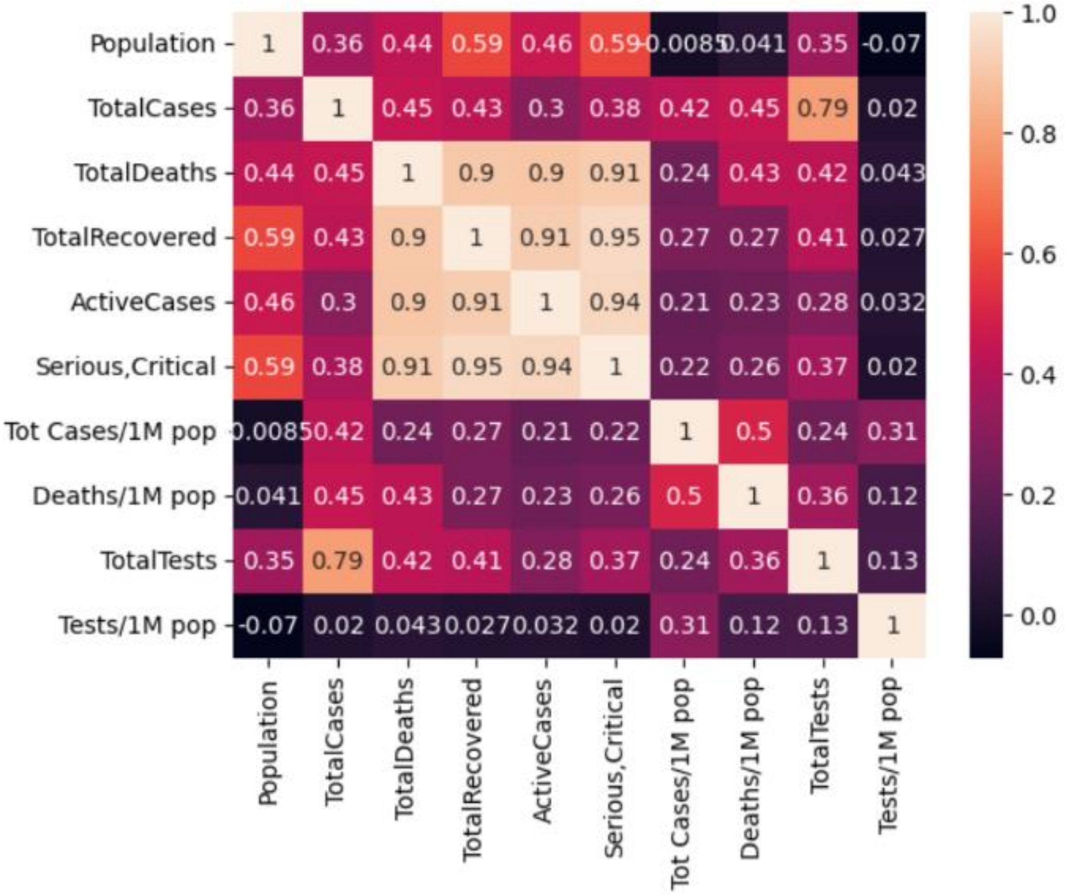
CODE AND OUTPUT



```
df2=df1.drop(["Country/Region","Continent","WHO Region"],axis=1)
```

```
sns.heatmap(df2.corr(),annot=True)
```

```
<Axes: >
```



	Population	TotalCases	TotalDeaths	TotalRecovered	ActiveCases	Serious,Critical	Tot Cases/1M pop	Deaths/1M pop	TotalTests	Tests/1M pop
Population	1	0.36	0.44	0.59	0.46	0.59	-0.0085	0.041	0.35	-0.07
TotalCases	0.36	1	0.45	0.43	0.3	0.38	0.42	0.45	0.79	0.02
TotalDeaths	0.44	0.45	1	0.9	0.9	0.91	0.24	0.43	0.42	0.043
TotalRecovered	0.59	0.43	0.9	1	0.91	0.95	0.27	0.27	0.41	0.027
ActiveCases	0.46	0.3	0.9	0.91	1	0.94	0.21	0.23	0.28	0.032
Serious,Critical	0.59	0.38	0.91	0.95	0.94	1	0.22	0.26	0.37	0.02
Tot Cases/1M pop	0.0085	0.42	0.24	0.27	0.21	0.22	1	0.5	0.24	0.31
Deaths/1M pop	0.041	0.45	0.43	0.27	0.23	0.26	0.5	1	0.36	0.12
TotalTests	0.35	0.79	0.42	0.41	0.28	0.37	0.24	0.36	1	0.13
Tests/1M pop	-0.07	0.02	0.043	0.027	0.032	0.02	0.31	0.12	0.13	1

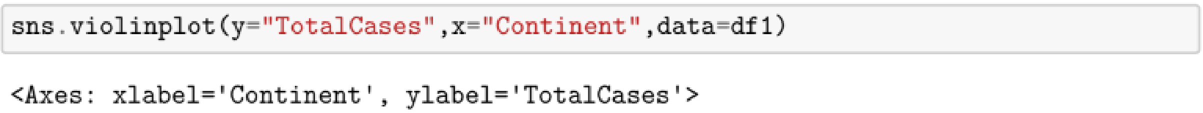
4.8 VIOLINPLOT

DESCRIPTION

A violin plot is a data visualization that combines aspects of a box plot and a kernel density plot. It is used to visualize the distribution of a continuous variable across different categories or groups. The violin plot displays the probability density of the data at different values, providing insights into both the central tendency and the spread of the data.

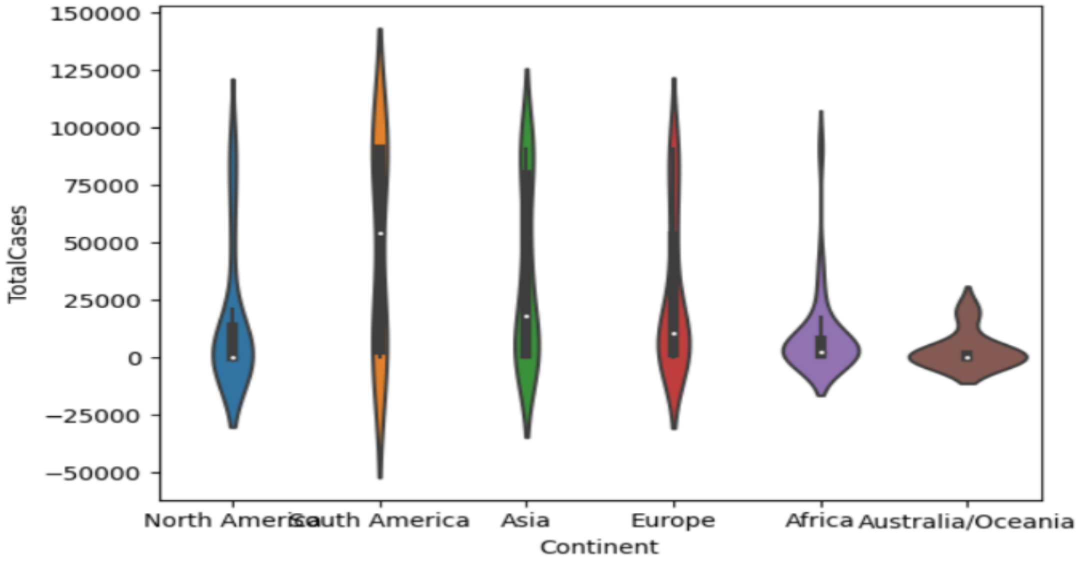
CODE AND DESCRIPTION

This plot is drawn between the total no of Covid cases(y-axis) in continent(x-axis) specified.



```
sns.violinplot(y="TotalCases",x="Continent",data=df1)
```

```
<Axes: xlabel='Continent', ylabel='TotalCases'>
```



5.CONCLUSION

In conclusion, the exploratory data analysis of the COVID-19 dataset has offered valuable glimpses into the multifaceted aspects of the pandemic. Through visualizations, statistical measures, and exploratory techniques, we've uncovered insights that contribute to a deeper understanding of the virus's behavior. This analysis not only aids in assessing the current state of the pandemic but also lays the groundwork for proactive decision-making and resource allocation. As we navigate the evolving landscape of COVID-19, continued exploration and analysis of datasets remain paramount for informed policymaking and effective public health responses. The insights gained from this EDA contribute to the collective knowledge essential for addressing the ongoing challenges posed by the pandemic, reinforcing the pivotal role of data-driven approaches in shaping resilient and adaptive public health strategies.

A violin plot is a data visualization that combines aspects of a box plot and a kernel density plot. It is used to visualize the distribution of a continuous variable across different categories or groups. The violin plot displays the probability density of the data at different values, providing insights into both the central tendency and the spread of the data.

6.REFERENCES

1. https://pandas.pydata.org/

2. https://matplotlib.org/

3. https://seaborn.pydata.org/