Capstone Project Report

Samanta Rana, #0810971, Data Analytics, Capstone project

St. Clair College, Mississauga

Objective:

"Unveiling COVID-19's Impact: Data Analytics on Death and Vaccination Datasets."

Table of Contents

Abstract:

This research study presents a comprehensive analysis of a global COVID-19 dataset, encompassing COVID-19 deaths and vaccinations, as well as factors such as hospital capacity, bed availability, vaccination success rates, and survival rates. The objective is to examine the interplay between these significant factors and understand their impact on the COVID-19 pandemic worldwide.

The dataset used for this analysis covers a wide range of countries and regions, offering a global perspective on the COVID-19 crisis. Various statistical and data mining techniques can be employed to uncover insights and patterns within the data.

The analysis begins by investigating the relationship between COVID-19 deaths and vaccinations. By examining the vaccination rates and their effectiveness, the study aims to assess the influence of vaccination campaigns on reducing mortality rates. Additionally, the research explores the factors associated with successful vaccination programs, including distribution strategies, public awareness campaigns, and healthcare infrastructure.

Furthermore, the study examines the role of hospitals and their capacity in mitigating the impact of COVID-19. The analysis includes the assessment of available beds, ICU capacity, and the strain on healthcare systems. The research also investigates the relationship between hospital resources and survival rates, considering factors such as medical staff-to-patient ratios, access to critical care equipment, and the availability of specific treatments.

Other significant factors considered in this analysis include socioeconomic variables, population density, government policies, and public health measures implemented in different regions. By incorporating these factors into the analysis, the study aims to provide a holistic view of the COVID-19 pandemic and its multifaceted impact on global health.

The findings of this research have implications for policymakers, healthcare professionals, and public health organizations worldwide. The insights derived from this analysis can contribute to informed decision-making, resource allocation, and the development of effective strategies to combat the ongoing COVID-19 crisis. Ultimately, the research aims to provide valuable insights into mitigating the impact of the pandemic and improving global health outcomes.

Research Question :

1. Can the current vaccination rates and success rates be used to predict future trends in COVID-19 deaths globally?

2. How does the strain on healthcare systems affect COVID-19 outcomes and survival rates?

3. What insights can be derived from the analysis to inform decision-making and the development of effective strategies to combat the ongoing COVID-19 crisis?

4. How can the insights from the analysis be leveraged to develop predictive models that aid decision-making, resource allocation, and the formulation of effective strategies to combat the ongoing COVID-19 crisis globally

5. Can we develop a predictive model that takes into account hospital capacity, including available beds and ICU capacity, to estimate the impact of COVID-19 on healthcare systems and predict future strain levels?

Keywords: COVID-19 · Measures · Social distancing · Mobility · Travel behavior

Click here for : Data Link

Github repository : *https://github.com/SamantaRana11/CapstoneProject.git*

Introduction

Title:

Unveiling the Insights of the Covid Dataset:
A Comprehensive Analysis of Covid Deaths and Vaccinations

The Covid-19 pandemic has left an indelible mark on humanity, challenging societies worldwide and transforming the way we live and interact. As this unprecedented crisis unfolded, researchers and policymakers sought to harness the power of data to understand its complexities better and devise effective strategies to combat the virus. In this context, our Capstone project takes on the monumental task of excavating and analyzing the vast Covid dataset, with a keen focus on two paramount aspects: Covid Deaths and Vaccinations.

The dataset at the heart of our study stands as a testament to the magnitude of the pandemic's impact, encompassing an impressive 10,868,428 instances. Each instance represents a unique snapshot of time, capturing various attributes that detail the evolution of the pandemic from its inception in January 2020 to the present day. The richness of this dataset empowers us to delve into a plethora of factors that have shaped the trajectory of Covid-19, allowing us to draw meaningful correlations and glean profound insights.

Crucial to our investigation are the two pivotal dimensions: Covid Deaths and Covid Vaccinations. The former focuses on the sobering reality of lives lost during the pandemic. By meticulously analyzing the Covid Deaths subset, we aim to unearth patterns and trends that shed light on the mortality rate under different circumstances. Our exploration encompasses critical phases of the pandemic, from its early emergence to the aftermath of vaccination campaigns. Understanding how the death rate has evolved over time and across regions is essential to comprehend the virus's behavior and adapt response strategies accordingly.

Furthermore, our analysis delves into the demographic aspect of Covid Deaths, probing how age distribution influences mortality rates. This knowledge can inform targeted measures to protect vulnerable age groups. Moreover, we investigate the socio-economic impact on mortality, as disparities in healthcare access and resources may have affected outcomes for different economic strata.

The second vital dimension, Covid Vaccinations, is central to understanding our path to recovery. Within the CovidVaccinations subset, we explore the efficacy of vaccination efforts in curbing the pandemic's spread. Delving

into the time taken by officials to develop and distribute vaccines provides invaluable insights into the magnitude of the challenge faced by researchers, policymakers, and healthcare providers in rapidly responding to the crisis.

An equally vital aspect of CovidVaccinations is the public's response to vaccination campaigns. We analyze data points that reflect the duration and efficacy of public awareness campaigns aimed at fostering vaccine acceptance. Understanding the dynamics of vaccine hesitancy and acceptance is crucial for future vaccination initiatives, ensuring that accurate information reaches the masses and fostering trust in vaccination as a vital tool in combating infectious diseases.

Additionally, our study recognizes the tireless efforts of healthcare professionals, hospitals, and researchers. Their role in spearheading vaccination drives, administering shots, and studying vaccine efficacy has been instrumental in the battle against Covid-19. We endeavor to highlight their contributions and acknowledge their sacrifices, which have been pivotal in safeguarding communities and saving lives.

In conclusion, our research project seeks to navigate the intricacies of the Covid dataset to glean profound insights into Covid Deaths and Vaccinations. By analyzing this wealth of information, we aim to contribute significantly to the body of knowledge surrounding the pandemic's impact and the efficacy of vaccination efforts. Our findings hold the potential to inform public health strategies, guide policy decisions, and foster resilience in the face of future health crises.

Data Dictionary

The list of categorical attributes :

S. no	Column names	Non-null count	DTypes
1.	iso_code	313267	category
2.	continent	298368	category
3.	location	313267	category
4.	date	313267	category
5.	tests_units	313267	category

The list of numerical attributes :

S. no	Column names	Non-null count	DTypes
1.	population	277045	numerical
2.	total_cases	256295	numerical
3.	total_deaths	79387	numerical
4.	total_tests	304385	numerical
5.	new_cases	304385	numerical
6.	new_cases_per_million	303121	numerical
7.	new_cases_smoothed_per_million	75403	numerical
8.	new_tests	79387	numerical
9.	new_tests_per_thousand	75403	numerical
10.	total_tests_per_thousand	103965	numerical
11.	new_tests_smoothed	103965	numerical
12.	new_tests_smoothed_per_thousand	95927	numerical
13.	positive_rate	94348	numerical
14.	tests_per_case	106788	numerical
15.	total_vaccinations	75332	numerical
16.	people_vaccinated	72143	numerical
17.	people_fully_vaccinated	68671	numerical
18.	total_boosters	43905	numerical
19.	new_vaccinations	169269	numerical
20	new_vaccinations_smoothed	169120	numerical
21.	total_vaccinations_per_hundred	169120	numerical
S. no.	Column names	Non-null count	DTypes
22.	people_vaccinated_per_hundred	72143	numerical
23.	people_fully_vaccinated_per_hundred	68671	numerical
24.	total_boosters_per_hundred	43905	numerical
25.	new_vaccinations_smoothed_per_million	169269	numerical
26.	new_people_vaccinated_smoothed	169120	numerical
27.	new_people_vaccinated_smoothed_per_hundred	169120	numerical
28.	stringency_index	197651	numerical
29.	population_density	165823	numerical
30.	median_age	247214	numerical
31.	aged_65_older	238587	numerical
32.	aged_70_older	244733	numerical
33.	gdp_per_capita	242298	numerical
34.	extreme_poverty	156082	numerical
35.	cardiovasc_death_rate	242726	numerical
36.	diabetes_prevalence	255124	numerical
37.	female_smokers	182104	numerical
38.	male_smokers	179624	numerical
39.	handwashing_facilities	118879	numerical
40.	hospital_beds_per_thousand	214292	numerical
41.	life_expectancy	288068	numerical
42.	human_development_index	235314	numerical

43.	excess_mortality_cumulative_absolute	10916	numerical
44.	excess_mortality_cumulative	10916	numerical
45.	excess_mortality	10916	numerical
46.	excess_mortality_cumulative_per_million	10916	numerical

Statistical Description of Dataset

The Dataset, as mentioned above, comprised of varied data-types of the data points, namely, categorical and numerical. Hence, the numerical data points can be observed better with a statistical description.

The statistical description of the dataset is defined as a comprehensive summary of the numerical attributes, aiming to reveal the distribution, central tendency, and dispersion of the data. For the numerical attributes in the dataset, various statistical measures will be computed to gain a deeper understanding of their characteristics.

Measures such as mean, median, and mode will offer insights into the central tendency, indicating the typical or average value of the data. Additionally, measures of dispersion, including the range, variance, and standard deviation, will provide information about the spread or variability of the numerical data points.

The statistical description will also encompass graphical representations, such as histograms, box plots, and scatter plots, which visually depict the distribution and potential outliers in the data. By employing these statistical techniques, we aim to unravel hidden patterns and gain valuable insights from the numerical attributes, contributing to a comprehensive analysis of the dataset and aiding in informed decision-making and data-driven conclusions.

	mean	std	min	25%	50%	75%	max	Description
total_cases	5813693.1	36255067.6	1.0	6607.0	62627.0	647602.0	766894311.0	umber of covid cases
new_cases	10679.3	102719.6	0.0	0.0	16.0	516.0	7460817.0	ew cases
new_cases_smoothed	10722.1	99962.9	0.0	1.0	37.1	625.6	6410666.9	e of daily count
total_deaths	80886.6	417669.6	1.0	122.0	1237.0	10711.0	6935876.0	otal deaths
new_deaths	95.2	599.1	0.0	0.0	0.0	6.0	20027.0	of deaths
new_deaths_smoothed	95.6	589.7	0.0	0.0	0.3	6.6	14677.9	laily count of deaths
total_cases_per_million	88706.4	139152.2	0.0	2064.9	21312.0	109074.3	737554.5	otal cases
new_cases_per_million	161.1	1124.2	0.0	0.0	2.3	67.7	228872.0	ber of new cases per million
new_cases_smoothed_per_million	161.8	632.9	0.0	0.2	10.5	102.3	37241.8	new cases per million
total_deaths_per_million	814.3	1056.2	0.0	51.1	339.8	1248.2	6477.6	er of total deaths per million
new_deaths_per_million	1.0	4.7	0.0	0.0	0.0	0.4	603.7	of deaths
new_deaths_smoothed_per_million	1.0	2.9	0.0	0.0	0.0	0.7	148.6	daily count of deaths per million
reproduction_rate	0.9	0.4	-0.1	0.7	1.0	1.1	5.9	births
icu_patients	697.1	2213.1	0.0	23.0	103.0	453.0	28891.0	ts

	mean	std	min	25%	50%	75%	max	Description
icu_patients_per_million	16.8	23.3	0.0	2.9	7.5	21.2	180.7	ts per million
hosp_patients	4070.6	10207.7	0.0	227.0	800.0	3141.0	154497.0	s in hospital
hosp_patients_per_million	138.8	156.6	0.0	36.9	86.4	181.9	1526.8	# of patients in hospital per million
weekly_icu_admissions	362.5	538.8	0.0	28.0	129.0	468.0	4838.0	nt of new patients
weekly_icu_admissions_per_million	11.0	14.1	0.0	2.3	5.8	14.2	225.0	w patients per million
weekly_hosp_admissions	4480.2	11327.9	0.0	278.2	937.0	4172.0	153977.0	hospital admittals
weekly_hosp_admissions_per_million	91.0	90.6	0.0	28.8	65.9	122.8	708.4	hospital per million
total_tests	21104573	84098694.3	0.0	364654.0	2067330	10248451.5	9214000000	
new_tests	67285.4	247734.0	1.0	2244.0	8783.0	37229.0	35855632.0	#
total_tests_per_thousand	924.3	2195.4	0.0	43.6	234.1	894.4	32925.8	er thousand
new_tests_per_thousand	3.3	9.0	0.0	0.3	1.0	2.9	531.1	er thousand
new_tests_smoothed	142178.4	1138214.7	0.0	1486.0	6570.0	32205.0	14769984.0	ew test # per thousand
new_tests_smoothed_per_thousand	2.8	7.3	0.0	0.2	0.9	2.6	147.6	new tests per thousand
positive_rate	0.1	0.1	0.0	0.0	0.1	0.1	1.0	ive cases
tests_per_case	2403.6	33443.7	1.0	7.1	17.5	54.6	1023631.9	d by # cases
total_vaccinations	382663569	1437893783.7	0.0	1489246.0	11093138	82238759.0	1339007476	ns
people_vaccinated	170715184	638040952.4	0.0	859788.0	5628383	41159127.5	5582392608	ccinated
people_fully_vaccinated	154275220	583835406.4	1.0	783535.5	4883115	34802538.0	5128292804	ly vaccinated
total_boosters	94039981	326648468.3	1.0	354596.0	3722696	27375984.0	2762734522	ters
new_vaccinations	837818.9	3392168.1	0.0	3093.0	27856.5	223513.8	49673470.0	nations
new_vaccinations_smoothed	322758.5	2058980.6	0.0	374.0	4634.0	36840.0	43692997.0	w vaccinations
total_vaccinations_per_hundred	115.0	84.1	0.0	35.4	113.3	184.8	406.4	ns per hundred
people_vaccinated_per_hundred	51.0	29.9	0.0	23.7	59.0	76.7	129.1	ccinated per hundred
people_fully_vaccinated_per_hundred	46.0	29.6	0.0	16.9	53.0	72.7	126.9	ly vaccinated per hundred
total_boosters_per_hundred	33.0	30.0	0.0	3.6	30.2	55.7	150.5	ters per hundred
new_vaccinations_smoothed_per_million	2095.1	3271.5	0.0	175.0	831.0	2802.0	117113.0	w vaccinations per million
new_people_vaccinated_smoothed	119458.5	837657.1	0.0	68.0	1092.0	11669.2	21071228.0	ople vaccinated
new_people_vaccinated_smoothed_per_hundred	0.1	0.2	0.0	0.0	0.0	0.1	11.7	ccinated per hundred
stringency_index	42.7	24.9	0.0	22.2	42.6	62.0	100.0	metrics
population_density	410.8	1873.9	0.1	37.7	90.7	222.9	20546.8	opulation
median_age	30.5	9.1	15.1	22.2	29.7	38.7	48.2	
aged_65_older	8.7	6.1	1.1	3.5	6.4	13.9	27.0	or older
aged_70_older	5.5	4.1	0.5	2.1	3.9	8.6	18.5	or older
gdp_per_capita	19016.6	19997.0	661	3823.2	12294.9	27216.4	116935.6	ita
extreme_poverty	13.8	20.1	0.1	0.6	2.5	21.4	77.6	rty
cardiovasc_death_rate	264.3	120.9	79.4	175.7	245.5	333.4	724.4	s due to cardiovescular
diabetes_prevalence	8.6	4.9	1.0	5.4	7.2	10.8	30.5	diabetes
female_smokers	10.8	10.8	0.1	1.9	6.3	19.3	44.0	okers

	mean	std	min	25%	50%	75%	max	Description
male_smokers	32.9	13.6	7.7	22.6	33.1	41.3	78.1	kers
handwashing_facilities	50.8	32.0	1.2	20.9	49.8	83.2	100.0	ing facilities
hospital_beds_per_thousand	3.1	2.5	0.1	1.3	2.5	4.2	13.8	eds per thousand
life_expectancy	73.7	7.4	53.3	69.6	75.0	79.5	86.8	ncy
human_development_index	0.7	0.1	0.4	0.6	0.7	0.8	1.0	lopment index
population	128372721	660555533.2	47.0	449002.0	5882259	28301700.0	7975105024	n
excess_mortality_cumulative_absolute	47665.2	139183.4	3776	32.3	4666.8	32395.6	1288358.4	r of excess deaths
excess_mortality_cumulative	9.5	12.8	-44.2	0.6	7.8	15.2	76.6	r of excess deaths
excess_mortality	12.3	26.2	-95.9	-1.4	6.3	17.7	377.4	r of excess deaths
excess_mortality_cumulative_per_million	1500.6	1854.5	2142	26.5	925.7	2450.2	10292.0	r of excess deaths

This dataset encompasses a wide range of variables that provide valuable insights into the COVID-19 pandemic's impact on different regions, populations, and healthcare systems. Researchers can use this dataset to explore correlations, derive patterns, conduct predictive modeling, and inform evidence-based decision-making to combat the ongoing COVID-19 crisis and prepare for future health challenges.

First phase CRISP - DM

Understanding the business problem :-

Here are some potential research goals:

1. **Predictive Modeling**:
   - **COVID-19 Spread Prediction**: Develop models to predict the spread of COVID-19 cases over time, considering various factors like vaccination rates, population density, and mobility.

2. **Vaccination Analysis**
   - **Vaccine Efficacy**: Evaluate the effectiveness of different COVID-19 vaccines in preventing infection and severe outcomes.
   - **Vaccination Impact**: Assess the impact of vaccination campaigns on reducing case numbers, hospitalizations, and deaths.

3. **Epidemiological Studies**:

- **Disease Trends**: Analyze trends and patterns in COVID-19 cases, deaths, and recoveries.
- **Hotspot Identification**: Identify regions or areas with a higher risk of outbreaks.

4. **Healthcare Resource Allocation**:
   - **Hospital Capacity Analysis**: Study the availability and utilization of hospital facilities and resources, and make recommendations for resource allocation.
   - **Optimizing Healthcare Response**: Develop models to optimize the allocation of healthcare resources during a surge in cases.

5. **Public Policy and Interventions**:
   - **Impact of Interventions**: Evaluate the effectiveness of various public health measures and interventions (e.g., lockdowns, mask mandates, social distancing).
   - **Policy Recommendations**: Provide evidence-based recommendations for policymakers to manage and mitigate the impact of the pandemic.

6. **Demographic and Socioeconomic Analysis**:
   - **Vulnerability Analysis**: Identify demographic and socioeconomic factors that correlate with a higher risk of infection or poor outcomes.
   - **Equity and Access**: Assess disparities in vaccine distribution and healthcare access.

7. **Mutations and Variants**:
   - **Genomic Analysis**: Study the genetic mutations and variants of the virus and their implications for transmission and severity.

8. **Behavioral Insights**:
   - **Public Behavior Analysis**: Examine how public behavior and compliance with guidelines impact the spread of the virus.

9. **Surveillance and Early Warning Systems**:
   - **Early Detection**: Develop models for early detection of potential outbreaks and emerging variants.
   - **Surveillance and Monitoring**: Implement a system for continuous monitoring of COVID-19 data and trends.

10. **Vaccine Deployment Strategy**:
    - **Optimal Distribution**: Determine the optimal strategy for vaccine distribution, considering factors like population density, vulnerability, and vaccine availability.

11. **Educational Campaigns**:
    - **Assessing Education Impact**: Evaluate the impact of public health education campaigns on public behavior and vaccine acceptance.

12. **Long-Term Effects**:
    - **Study Long-Term Health Effects**: Investigate the potential long-term health consequences for individuals who have had COVID-19.

Second phase CRISP - DM

The Cross-Industry Standard Process for Data Mining (CRISP-DM) is a widely used framework for guiding data mining and machine learning projects. It consists of six phases. The second phase of CRISP-DM is "Data Understanding."

In the Data Understanding phase, you work on gaining a better understanding of the data you're going to use for your project. This typically involves:

1. **Collecting the data:** The dataset is collected from the following link :- Data Link

2. **Describing the data:** The data has a total of (313267 * 67) entries.

   Exploring the data to get a basic understanding of its structure, contents, and quality. This involves generating summary statistics, identifying missing values, and visualizing the data.

	count	mean	std	min	25%	50%	75%	max
total_cases	313267.0	5.141476e+06	3.414532e+07	0.0000	1587.000	34658.000	472594.0	7.668943e+08
new_cases	313267.0	1.037648e+04	1.012684e+05	0.0000	0.000	13.000	471.0	7.460817e+06
new_cases_smoothed	313267.0	1.037485e+04	9.834910e+04	0.0000	0.429	30.571	572.5	6.410667e+06
total_deaths	313267.0	6.617621e+04	3.790719e+05	0.0000	13.000	396.000	6860.0	6.935876e+06
new_deaths	313267.0	9.256791e+01	5.907791e+02	0.0000	0.000	0.000	5.0	2.002700e+04
...	...	...	...	...	...	...	...	...
population	313267.0	1.283727e+08	6.605555e+08	47.0000	449002.000	5882259.000	28301700.0	7.975105e+09

	count	mean	std	min	25%	50%	75%	max
excess_mortality_cumulative lute	313267.0	1.660926e+03	2.741133e+04	-37726.0980	0.000	0.000	0.0	1.288358e+06
excess_mortality_cumulative	313267.0	3.313608e-01	2.958736e+00	-44.2300	0.000	0.000	0.0	7.655000e+01
excess_mortality	313267.0	4.294640e-01	5.379763e+00	-95.9200	0.000	0.000	0.0	3.773700e+02
excess_mortality_cumulative_per_million	313267.0	5.228864e+01	4.422206e+02	-2142.3403	0.000	0.000	0.0	1.029202e+04

3. **Exploring the data:** Further exploring the data to identify patterns, relationships, or anomalies, which was later used for data mining processes, for example, observing the patterns for filling the null values using ‘rolling values averages.

4. **Verifying data quality:** Assessing data quality, which includes checking for inconsistencies, errors, and missing values. You may need to perform data cleaning and preprocessing.

5. **Assessing data relevancy:** Determining whether the data you have collected is relevant to the goals of your project. It's important to ensure that the data is suitable for your intended analysis.

The Data Understanding phase is crucial for building a solid foundation for the rest of the project. It helps you identify potential issues and gain insights into the data you will be working with, allowing you to make informed decisions about data preparation, feature engineering, and modeling in the subsequent phases of CRISP-DM.

Third phase CRISP - DM

In the third phase of CRISP-DM, the dataset is prepared to find some valuable insights and relationship between categorical and numerical variable, for which , chi-square test is used for finding the strength of relationship between the categorical variables and for evaluating the same insights in terms of mean of two variables at a time, t-test is used. The Chi-Square Statistic is 42275.23 with 24 degrees of freedom, resulting in a p-value of 0.0 which shows that the observed data significantly differs from the expected frequencies in the contingency table.

Third phase of CRISP-DM deals with preparation of data for further analysis, since transformation of data has already been done in the previous steps, hence, this phase highlighted the classification and clustering analysis of the dataset of COVID. For that, elbow method is used to find an optimal value of 'k' to use for k-Means clustering to primarily analyze the total cases versus the total deaths ration with the help of clustering.

Moreover, since the dataset contain features of varied scale , the process of standardization is performed, which helps to set the maximum and minimum point of the dataset as fixed so that all the values of features lies within those limits and it becomes easy for the analyst to analyze the dataset visually.

In the next analysis, Linear Regression is performed, to assess the number of cases given various factors on which it depends (observed using chi-square test and t-test) and at last ANOVA test is applied as ANOVA, or Analysis of Variance, is a statistical method used to analyze the differences among group means in a sample. It is a powerful and flexible tool.

All the models that have been performed are evaluated based on the their statistics and the accuracy and also the errors. After the deployment of these models they have been evaluated for their valuable insights and then they have been reiterated and rebalanced for their parameters which at last has provided better accuracy and better results to analyse the data set for any valuable knowledge of the data set required.

Fourth phase CRISP - DM

For the analyses in the fourth DM given the priority as compared to the. Analysis. The whole data set is divided into various sub categories containing on the basis of. Continents and one data set that contains the data points of the whole world. Then these excel files are analyzed separately so that valuable insights can be. Find out from these files and visualize for better presentation.

Further various interesting graphs and charts have been employed in the Jupiter notebook file for the analysis for example donut chart line graphs bar graphs histograms and comparative scatter plots. These charts and graphs have provided valuable insights into the data sets and how the economy of any continent has effected the total cases of COVID as compared to the hospital facilities available in that continent and the total cases people who are fully vaccinated the total tests that have been performed on those people and finally and unfortunately how many total deaths have taken place for those particular continents.

Data Mining processes and Insights

The first step while initiating analysing the data was to clean it thoroughly, because -

1) it is a live data and so, can contain many impurities.

2) during the initial phase of Covid, the entries which got registered were not uniform and timely.

3) the data is of varied measures; implying normalization is required.

Another aspect of this dataset is that it contains over 3 million entries, hence, the project has started from cleaning only the attributes required for analysis.

Post-preprocessing, once the dataset is cleaned and prepared for the analysis, the first and most generalized analyses is carried out on the total number of cases as against the total deaths occurred. The trend was varied when the nations from all around the globe are considered. Following is the review of the analysis of total number of cases versus total number of deaths.

Impact of Healthcare System Strain on COVID-19 Outcomes and Survival Rates:

The strain on healthcare systems has emerged as a critical factor influencing COVID-19 outcomes and survival rates. The rapid and overwhelming surge in cases during the pandemic has put immense pressure on healthcare infrastructure worldwide. Overburdened hospitals face challenges in providing timely and adequate care to COVID-19 patients, leading to potential delays in treatment and an increased risk of adverse outcomes.

Research has shown that regions with higher hospitalization rates and limited intensive care unit (ICU) capacities experience higher COVID-19 fatality rates. Limited availability of critical medical resources, such as ventilators and ICU beds, can compromise patient care and contribute to excess mortality.

Healthcare system strain also has implications beyond COVID-19 care. As hospitals prioritize COVID-19 patients, non-COVID-19 medical procedures may be delayed, impacting patients with other health conditions. This situation can lead to increased morbidity and mortality for non-COVID-19 patients, further contributing to excess mortality.

Non-pharmaceutical interventions (NPIs) have been crucial in mitigating healthcare system strain and improving COVID-19 outcomes. Measures such as social distancing, mask-wearing, and travel restrictions help slow the spread of the virus, reducing the number of severe cases and preventing overwhelming surges in hospital admissions.

To address healthcare system strain effectively, governments and health authorities need to implement proactive strategies. Increasing hospital capacity, enhancing the availability of medical resources, and adopting adaptive healthcare policies that balance COVID-19 care with other medical needs are essential in managing the ongoing pandemic.

Data Pre-processing :

The dataset in concern contains numerous null values, count is different for varied fields, and hence the handling of null values is also different for all the fields. Also, since the dataset contain over 3 million entries, with the scope of complexity and chances of loss of data, complete data at once is not cleaned rather the fields required as per the research questions is cleaned accordingly.

Four major processes involved in cleaning are :-

1) The whole data contained null data cells at the beginning, implying that the early data was not recorded and probably not registered due to the sudden outbreak of Covid. Hence, those entries were fille with ‘0’ .

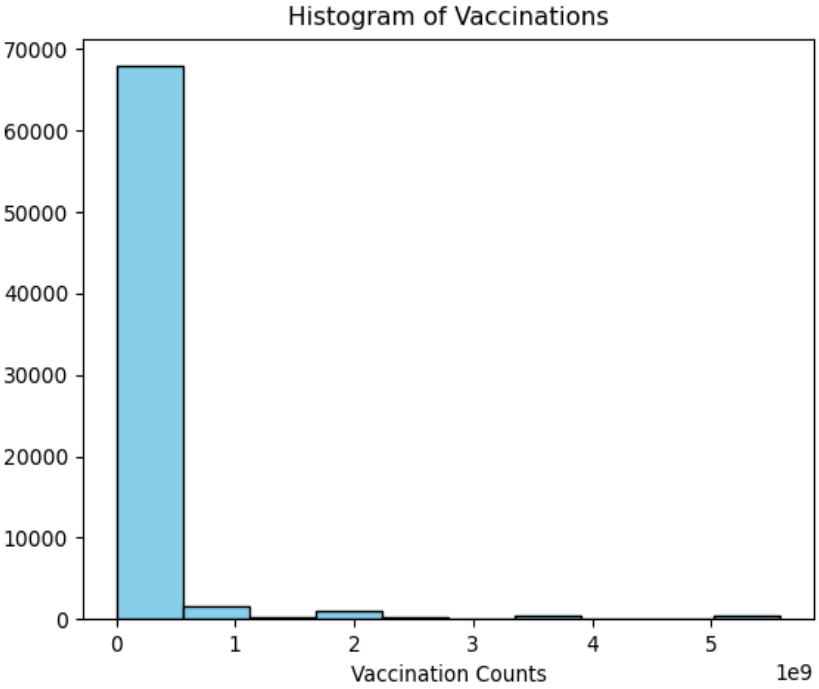
2) Then the whole data is filtered year and then month wise and ‘rolling means’ is employed to fill in the null values after observing the definite pattern in the dataset. Since, all columns have different pattern to them,

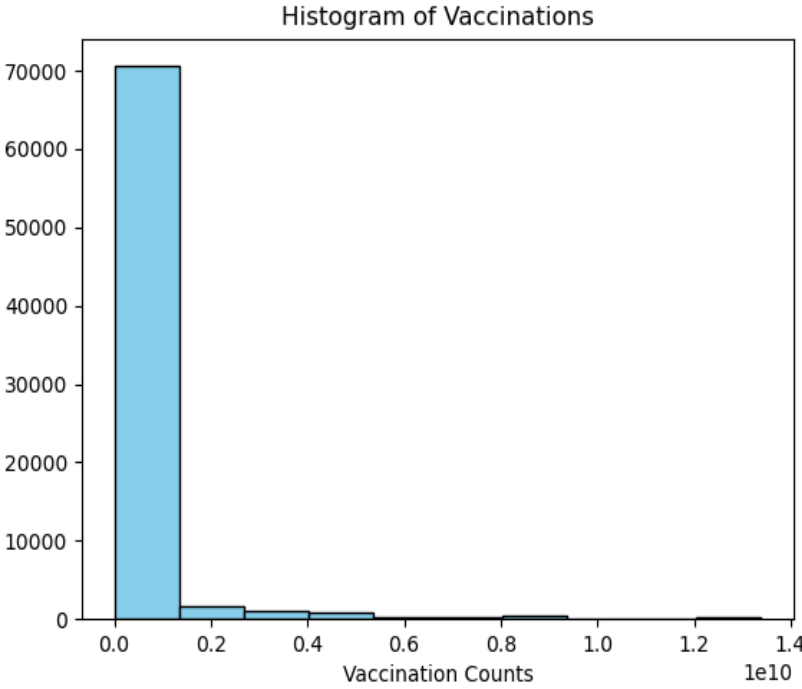
so, rolling mean of window 4- 18 and for the column vaccinations, window of 50 was taken too. The priority here is kept for the definite pattern to be observed so that no wrong analysis can be done.

3) to eradicate the null values for the column = 'population', firstly the data is divided nation-wise and then the mean of the national population is used to fill the null values.

4) for filling up the missing age of the demographic from the column = 'age', the dataset is divided into sub-groups of age and then median is taken to fill up the null values of the age.

The left histogram represents the number of people who got vaccinated, however, the following right histogram reflects the total vaccinations produced.





Insights for Decision-Making and Effective Strategies in Combating COVID-19:

The comprehensive COVID-19 dataset offers valuable insights that inform decision-making and the development of effective strategies to combat the ongoing crisis. Researchers have analyzed data to identify patterns related to COVID-19 transmission, severity, and mortality rates. These insights have guided the formulation of evidence-based policies and interventions to control the spread of the virus and reduce mortality rates.

Vaccination campaigns have proven to be a critical tool in reducing COVID-19 deaths and achieving pandemic control. Data analysis has enabled public health authorities to identify regions with low vaccination coverage and prioritize vaccine distribution. Insights from data have also informed strategies for mass vaccination drives, outreach to vulnerable populations, and addressing vaccine hesitancy.

Non-pharmaceutical interventions (NPIs) have played a crucial role in curbing the spread of the virus. Data analysis has provided evidence of the effectiveness of measures such as lockdowns, mask mandates, and social distancing policies. Decision-makers have used these insights to implement targeted NPIs based on regional disease prevalence and healthcare system capacity.

Furthermore, data analysis has underscored the importance of adopting a multidimensional approach in combating COVID-19. Strategies that combine vaccination efforts with NPIs and socioeconomic support have proven more effective in reducing the burden of the disease and promoting equitable access to healthcare resources.

Data-driven decision-making has also been instrumental in early detection and rapid response to emerging outbreaks. Real-time analysis of COVID-19 data allows for prompt identification of regions experiencing surges in cases or deaths, enabling health authorities to implement timely containment measures.

Literature extraction

Before the full text assessment, we also reviewed reference lists for relevant literature and discovered additional relevant articles through forward and backward reference tracing, adding them to the search lists to complement the literature identified through database searches. Subsequently, duplicates were removed, and the remaining studies were further screened for relevance and scope by examining each article's abstract, introduction, and conclusion. After filtering, 364 articles remained for the final analysis. We thoroughly reviewed each study's content and conducted thematic analyses to categorize studies based on their topics and study perspectives. This approach effectively identified each study's purpose, data, and results, and grouped them into major topics and sub-topics. When the study subject and transport means were similar across multiple studies, those not significantly meaningful to this study's review subject were not included in the analysis. The articles relevant to each subject were extracted and summarized in tables that included publication details (author(s) and year), study area, research objective and method, data type, and transport mode type.

Government's measures

This chapter reviews the literatures on the overall impacts of the COVID-19 outbreak on mobility regardless of the presence or absence of the government's specific measures. Specifically, existing studies are discussed under the following topics:

(1) the relationship between human contact and COVID-19 transmission;

(2) the impact of COVID-19 on overall mortality based on observed data;

(3) the impact on public health services;

(4) other impacts; and

(5) changes in personal hygiene behavior based on survey data.

Research -A Global Perspective

1. Introduction

The COVID-19 pandemic has unleashed a profound global health crisis, challenging nations across the world to grapple with its devastating impact. Amidst efforts to comprehend the intricate dynamics of the virus's spread and severity, a surprising pattern has emerged in the relationship between total COVID-19 cases and total deaths. Unlike conventional infectious diseases, the progression of COVID-19 cases and deaths seems to follow a unique trajectory that varies across nations. This report aims to explore this curious relationship and examine potential factors contributing to the observed patterns. By conducting a comprehensive analysis of COVID-19 data from diverse countries, this study endeavors to unveil insights into the enigmatic association between total cases and total deaths.

2. Methodology

To unravel the mysteries surrounding the intriguing relationship between total COVID-19 cases and total deaths, a systematic and data-driven approach was adopted. A rich dataset was curated from reputable sources such as the World Health Organization (WHO), national health departments, and prominent research databases. The dataset incorporated COVID-19 case and death data from various countries and regions, enabling an expansive global analysis. Advanced statistical techniques, data visualization, and regression analysis were employed to discern underlying trends and correlations between total cases and total deaths.

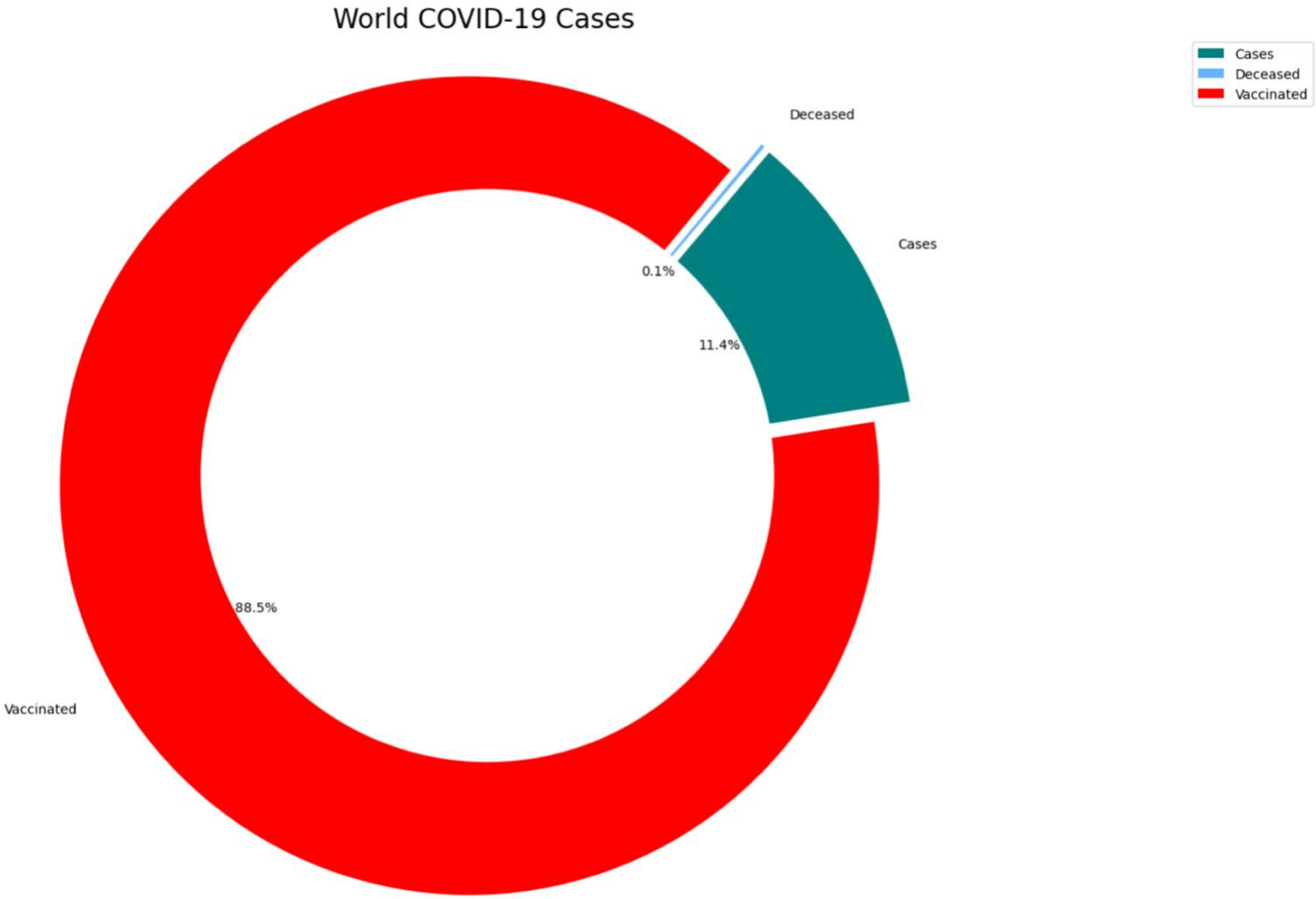
The analysis of the dataset marks with finding the correlation among the input variables, the heat map for which is shown below, the scale by the side of the map depicts the strength of relation between the variables, the lighter the tone the positive and strong the correlation is and the darker the shade implies the negative but still strong correlation. This heat map depicts the direct and inverse correlation among the attributes in concern.

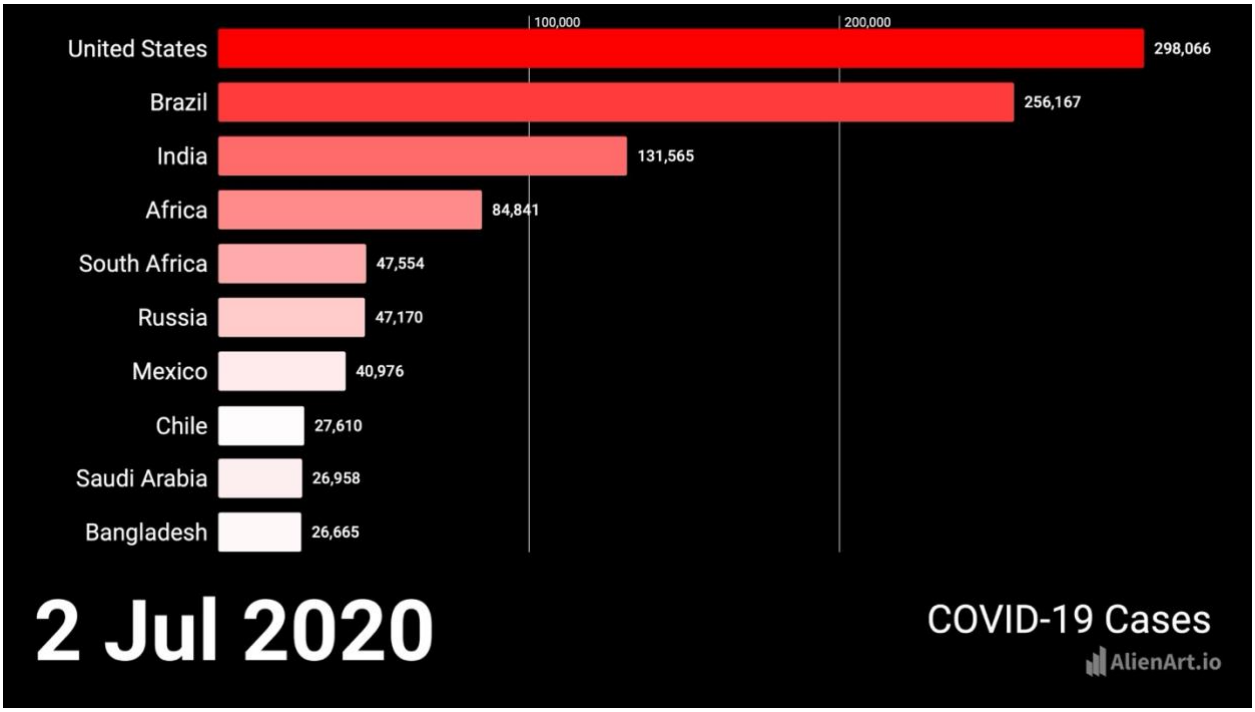
3. Results

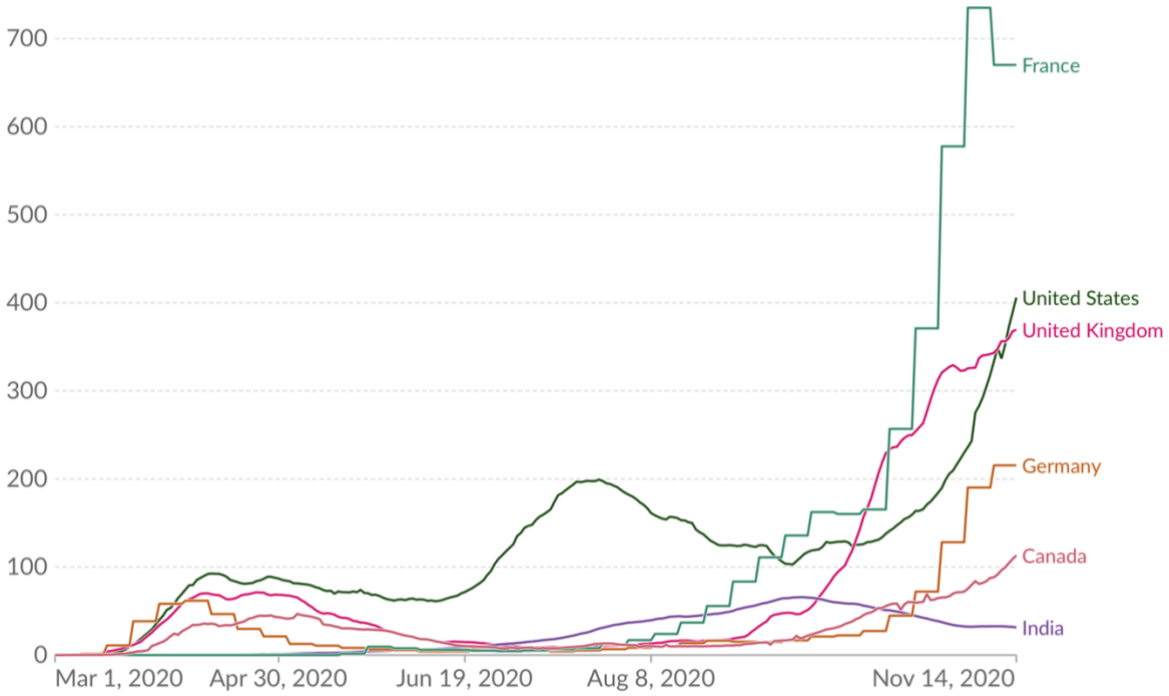
The analysis of the dataset unraveled a remarkable pattern in the relationship between total COVID-19 cases and total deaths across diverse nations. Notably, the curve illustrating this association exhibited unexpected variations, suggesting that the pandemic's impact differs significantly from country to country. Intriguingly, some regions displayed a disproportionate increase in total cases without a corresponding surge in total deaths, while others exhibited a more traditional progression. The presence of outliers highlighted the complexity of the phenomenon, requiring a closer examination of influencing factors.

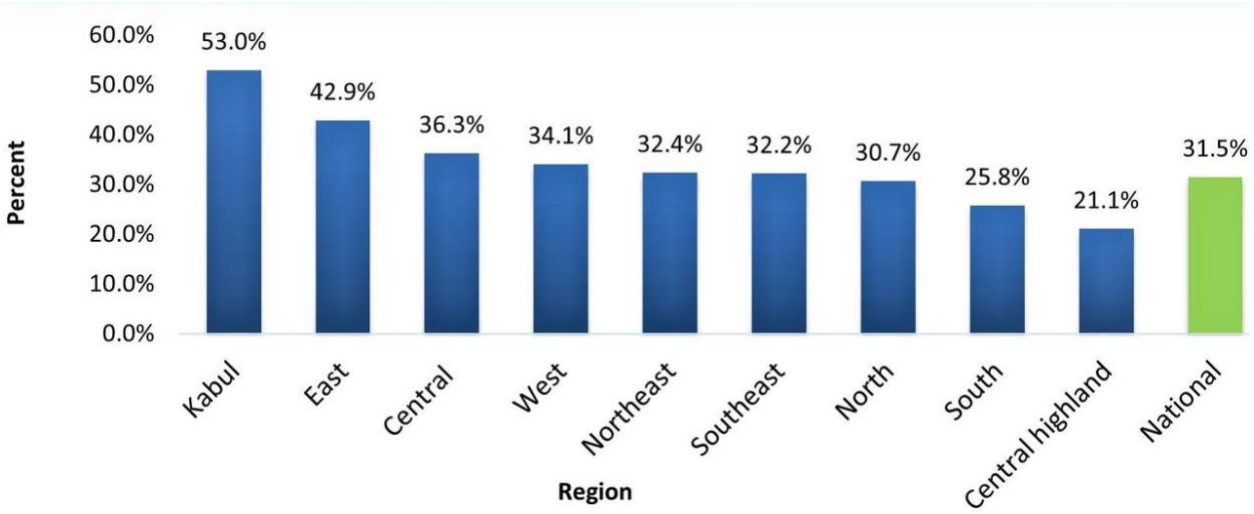
4. Discussion

In light of the surprising findings, several potential hypotheses were formulated to shed light on the observed relationship between total COVID-19 cases and total deaths. The effectiveness of early containment measures, healthcare system capacity and preparedness, the prevalence of new variants, and vaccination efforts emerged as key factors warranting in-depth investigation. Regions that swiftly implemented stringent public health measures and sustained high vaccination rates appeared to have managed the pandemic more effectively, reflected in the unique curves in their case-to-death trajectory.


World COVID-19 Cases
Cases
Deceased
Vaccinated
Deceased
Cases
0.1%
11.4%
88.5%
Vaccinated


100,000
200,000
United States
298,066
Brazil
256,167
India
131,565
Africa
84,841
South Africa
47,554
Russia
47,170
Mexico
40,976
Chile
27,610
Saudi Arabia
26,958
Bangladesh
26,665
2 Jul 2020
COVID-19 Cases
AlienArt.io


700
600
500
400
300
200
100
0
Mar 1, 2020
Apr 30, 2020
Jun 19, 2020
Aug 8, 2020
Nov 14, 2020
France
United States
United Kingdom
Germany
Canada
India



5. Implications and Recommendations

The exploration of the curious relationship between total COVID-19 cases and total deaths bears significant implications for public health policies and strategies. Policymakers must consider tailored interventions to address the varying impact of the pandemic on different regions. Enhancing healthcare system capacities, bolstering early containment measures, and prioritizing equitable vaccine distribution remain critical in mitigating the pandemic's consequences. Data-driven decision-making will continue to be essential in guiding evidence-based interventions and formulating adaptive responses to this evolving crisis.

6. Conclusion

In conclusion, the analysis of the unique relationship between total COVID-19 cases and total deaths has offered valuable insights into the pandemic's dynamics. The observed variations in the case-to-death trajectory underscore the multifaceted nature of the COVID-19 crisis, urging further research and cooperation among nations to combat the pandemic effectively. By understanding the contributing factors and crafting targeted interventions, the global community can enhance preparedness and resilience to confront this unprecedented health challenge.

To support the above report, here is an instance from two different coutries at two different time-spans -

location	**Afghanistan**
date	2020-03-25
total_cases	42.0
total_deaths	1.0
Death_Percentage	**2.380952**

location	**Albania**
date	2020-10-06
total_cases	14266.0
total_deaths	396.0
Death_Percentage	**2.775831**

location	**Fiji**
date	2020-09-17
total_cases	32.0
total_deaths	2.0
Death_Percentage	**6.25**

location	**Europe**
date	2020-08-10
total_cases	3024803.0
total_deaths	218887.0
Death_Percentage	**7.236405**

This varied percentage of death count reveals that the impact of Covid was not uniform around the globe and the number or ratio of total death counts were different as mentioned in an instance below :-

TotalDeathCount

	TotalDeathCount
location	
United States	1127152.0
Brazil	702421.0
India	531843.0
Russia	398919.0
Mexico	334079.0
United Kingdom	225852.0
Peru	220561.0
Italy	190242.0
Germany	174032.0
France	163437.0
Indonesia	161701.0
Iran	146230.0
Colombia	142741.0

From the actual table containing 228 entries,

United States suffered the most deaths -

United States	1127152.0

And Nauru suffered the least -

Nauru	1.0

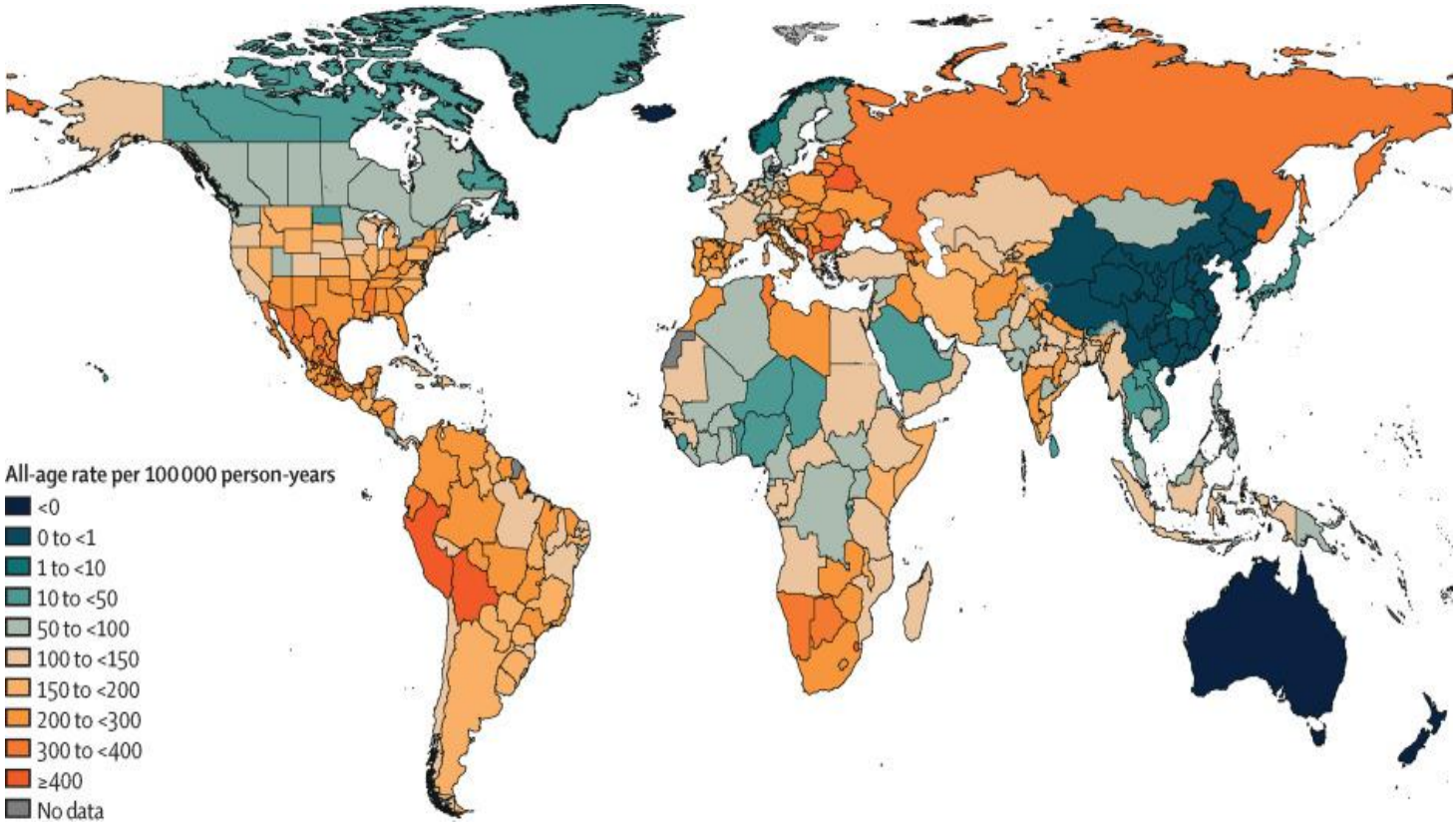
and all the other nations swinged in between, other than economic mobility the other reason that could be stated for this chance of events is freedom of expression and lack of control of government official bodies.

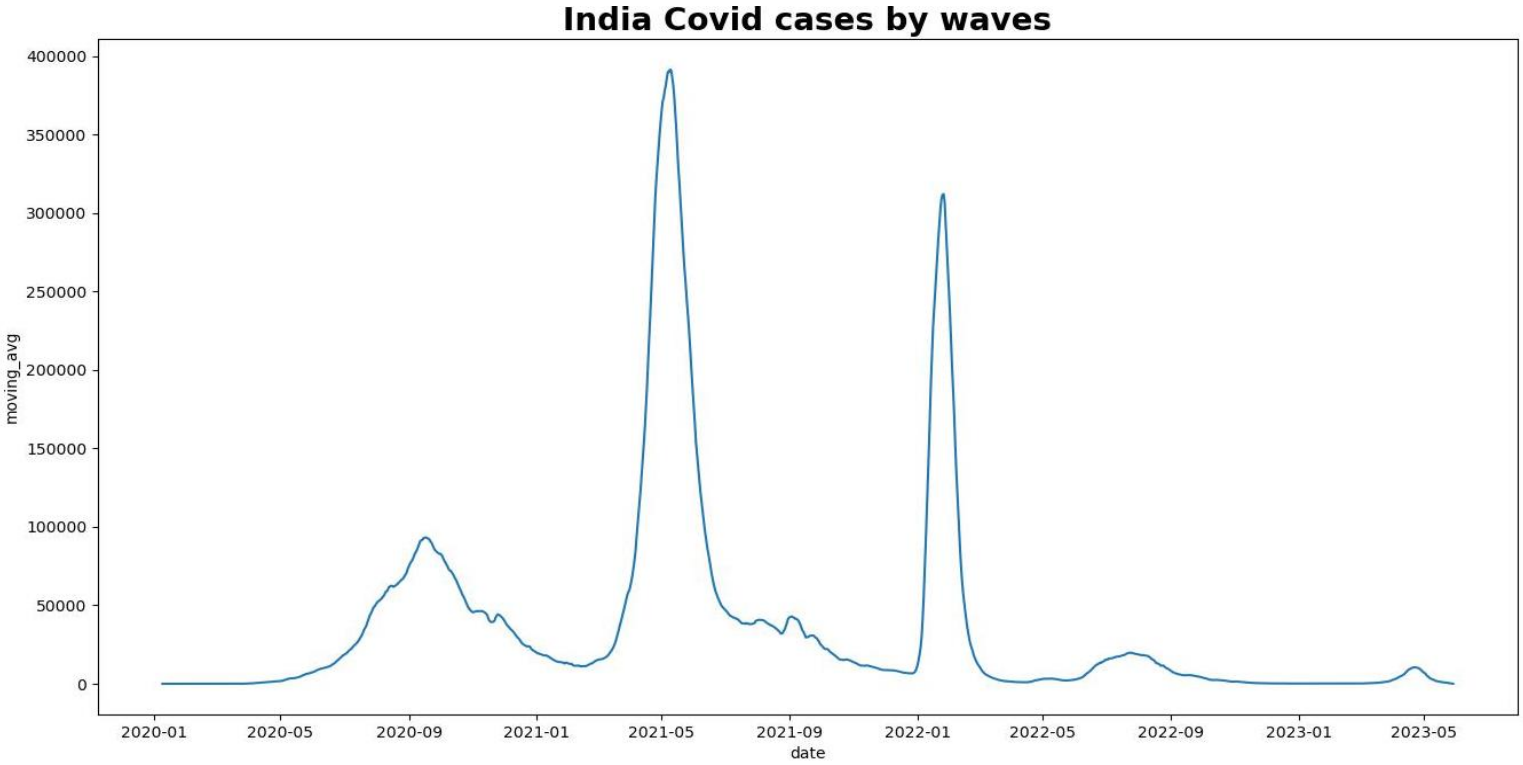
Due to freedom of expression, masses were protesting against government policies at large extent which remained the major cause of spread of Covid, due to unawareness of the spread and adversities caused by this virus at a large extent and vulnerability of the unknown along with the lack of government initiation/slow application of law and policies/unable to read the situation properly were some drawbacks on the official's side.

The same studies have been done in accordance to taking into consideration a single nation and to observe how the spread of Covid boomed over the passing days. For an instance, if the nation into consideration is, India, for example. Then, we know although the Covid marked its presence from 2019 but due to some unknown or unverified reasons, the data from the initial days has not been recorded, however, starting from January 2020, the records for all parameters were recorded appropriately.

The values for the parameters for India followed a curve which is theoretically justified from January 2020 - January 2022, however, post this timeframe a quick surge in the Covid cases were again seen accompanied by the birth of new variant, it was unfortunate that even after controlling the Covid after an year of its spread, its new variant had the impacts the nationals similarly, although from the graph it is clear that the extent of impact was comparatively low, however,
Lack of controlling the circumstances was marked as a failure and similar trend is seen for all the nations.

Below is the : Global distribution of estimated excess mortality rate due to the COVID-19 pandemic, for the cumulative period





Executive Summary :

The COVID-19 pandemic has presented unprecedented challenges to global public health and economies. This report provides a comprehensive analysis of key insights derived from data to inform decision-making and the development of effective strategies to combat the ongoing COVID-19 crisis. The insights cover epidemiological trends, vulnerable populations, healthcare system capacity, vaccination effectiveness, testing, and contact tracing, public compliance, economic impact, and the importance of global cooperation. Based on these insights, the report recommends targeted interventions, vaccination campaigns, healthcare preparedness, testing and tracing improvements, public awareness, travel restrictions, economic support, research and development, and global collaboration.

Data Driven Recommendations

1. Epidemiological Trends:

- Analyze the spread of the virus over time to identify hotspots and potential areas of concern.
- Monitor the reproductive number (R0) and the impact of different variants to guide targeted interventions.
- Use modeling and forecasting to anticipate potential future trends and prepare for different scenarios.

Recommendation:

Implement targeted interventions in areas with high transmission rates, and closely monitor areas experiencing outbreaks.

2. Vulnerable Populations:

- Identify demographic groups most vulnerable to severe outcomes from COVID-19.
- Prioritize vaccination and resource allocation based on vulnerability.

Recommendation:

Launch vaccination campaigns focusing on high-risk groups, such as elderly individuals and individuals with underlying health conditions.

3. Healthcare System Capacity:

- Monitor hospitalization rates, ICU admissions, and medical resource availability.
- Plan for resource allocation and surge capacity.

Recommendation:

Build healthcare system capacity to handle surges and emergencies effectively.

4. Vaccination Effectiveness:

- Analyze vaccination data to assess vaccine effectiveness and the need for booster shots.
- Monitor the overall impact of vaccination on reducing severe cases and deaths.

Recommendation:

Ensure equitable distribution of vaccines and prioritize booster shots as needed.

5. Testing and Contact Tracing:

- Improve testing rates and effectiveness of contact tracing efforts.
- Early detection and containment are crucial in managing outbreaks.

Recommendation:

Enhance testing capacity and contact tracing efforts to identify and isolate cases promptly.

6. Public Compliance:

- Analyze data on public adherence to preventive measures (e.g., mask-wearing, social distancing).
- Evaluate the effectiveness of public health campaigns and messaging.

Recommendation:

Improve public health messaging to encourage preventive behaviors and combat misinformation.

7. Impact on Economy and Society:

- Assess the economic and social impact of the pandemic.
- Identify sectors and vulnerable populations requiring targeted support.

Recommendation:

Provide financial aid and support to affected businesses and individuals.

8. Travel Restrictions and Quarantine Measures:

- Analyze data from high-risk regions to implement travel restrictions and quarantine protocols.

Recommendation:

Implement travel restrictions and quarantine measures based on data from high-risk regions.

9. Global Collaboration:

- Identify successful strategies and areas requiring international cooperation.
- Strengthen global collaboration to share knowledge, resources, and best practices.

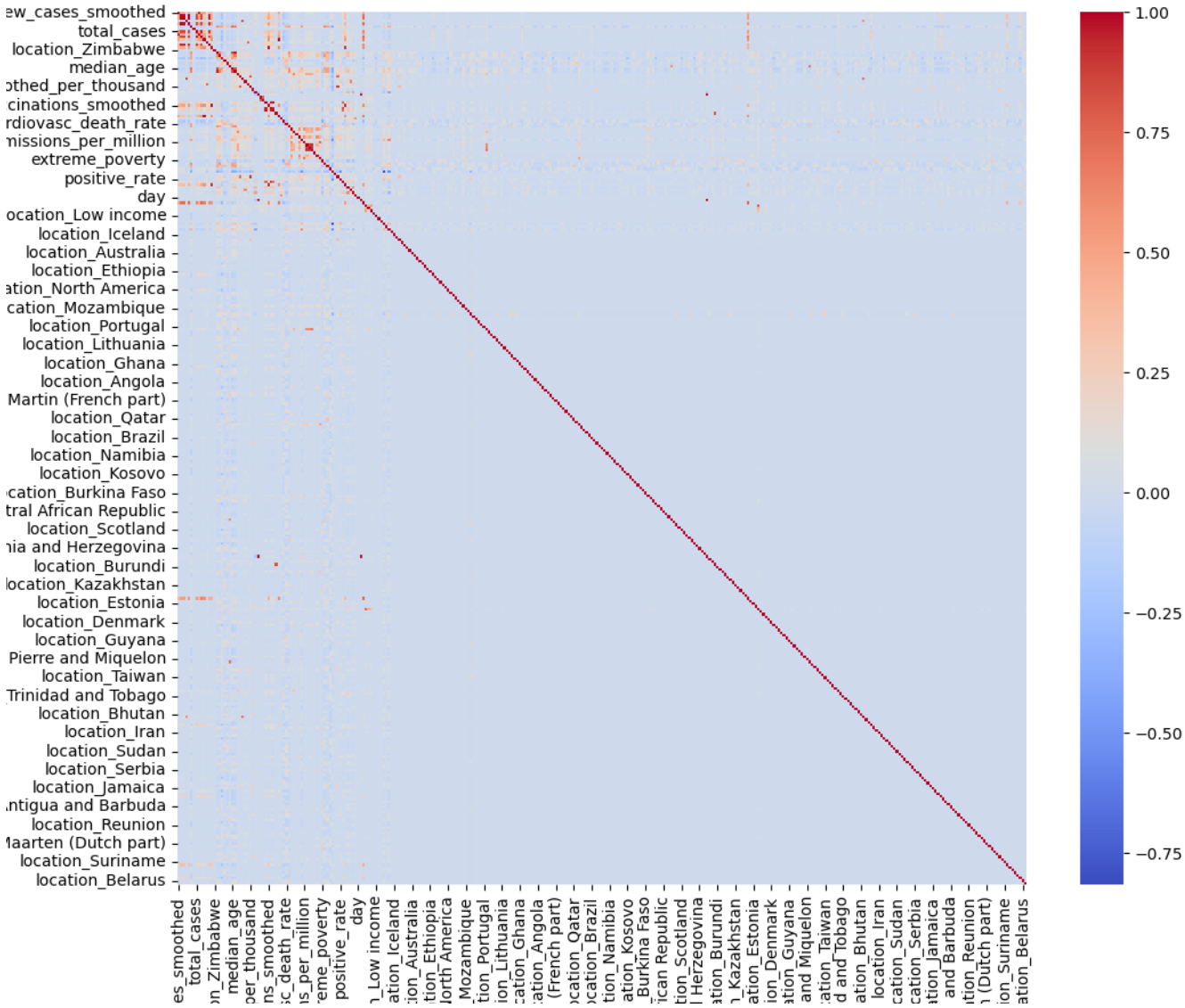
Conclusion:

Data-driven insights are invaluable in combating the ongoing COVID-19 crisis. This report has highlighted key insights that can guide decision-making and the development of effective strategies. By implementing targeted interventions, prioritizing vaccination campaigns, improving healthcare capacity, enhancing testing and tracing efforts, and promoting public awareness, we can work together to navigate through these challenging times and minimize the impact of the pandemic on public health and societies worldwide. Continuous monitoring and evaluation of data will be crucial in making informed adjustments as the situation evolves.

Appendix

Feature engineering for column - 'date'

In [15]:
```
# Convert the 'date' column to a datetime data type
data['date'] = pd.to_datetime(data['date'])

# Extract year, month, and day features
data['year'] = data['date'].dt.year
data['month'] = data['date'].dt.month
data['day'] = data['date'].dt.day

# Extract day of the week as a numerical value (0=Monday, 6=Sunday)
data['day_of_week'] = data['date'].dt.dayofweek

# Extract week of the year
data['week_of_year'] = data['date'].dt.isocalendar().week

# Extract quarter
data['quarter'] = data['date'].dt.quarter

# Extract whether it's a weekend (1) or a weekday (0)
data['is_weekend'] = data['day_of_week'].apply(lambda x: 1 if x in [5, 6] else 0)

# Extract the day of the year
data['day_of_year'] = data['date'].dt.dayofyear
```



In [6]:
```
# Create a contingency table
contingency_table = pd.crosstab(crosstab_data['continent'], crosstab_data['tests_units'])
```

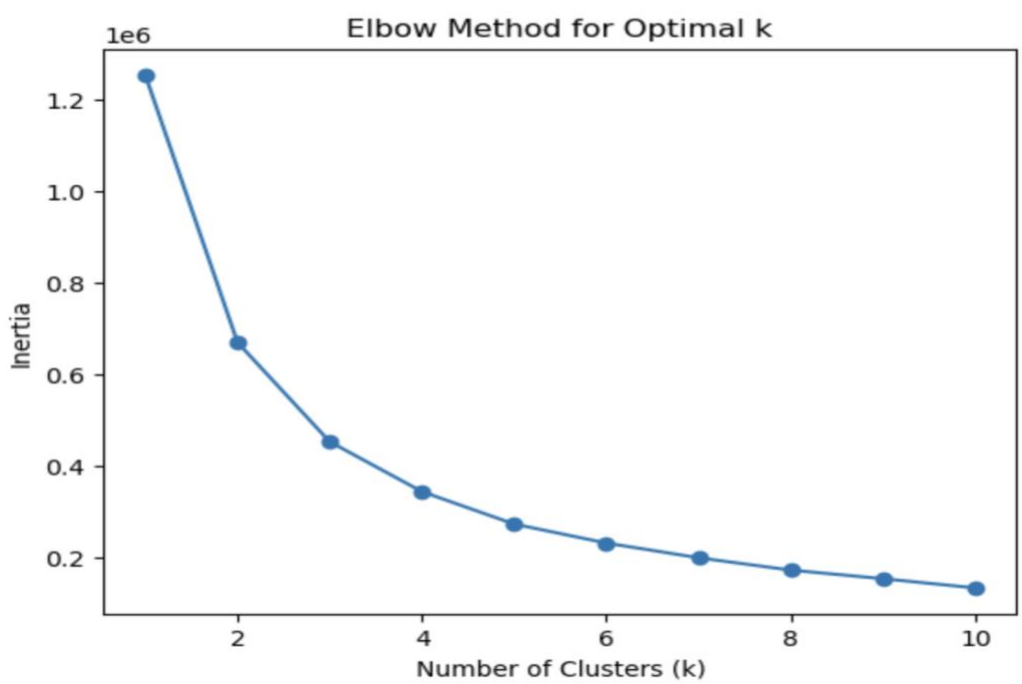
In [7]:
```
# Perform the chi-square test
chi2, p, dof, expected = chi2_contingency(contingency_table)

# Output the test results
print("Chi-Square Statistic:", chi2)
print("P-Value:", p)
print("Degrees of Freedom:", dof)
print("Expected Frequencies Table:")
print(expected)
```

```
Chi-Square Statistic: 42275.2313136625
P-Value: 0.0
Degrees of Freedom: 24
Expected Frequencies Table:
[[9.82015540e+03 7.73184035e+02 4.56148618e+02 3.80951393e+03
  3.99980177e+01]
 [4.65117730e+04 3.66207650e+03 2.16048322e+03 1.80432223e+04
  1.89444937e+02]
 [4.09350004e+04 3.22299266e+03 1.90144077e+03 1.58798357e+04
  1.66730444e+02]
 [4.47275485e+04 3.52159666e+03 2.07760556e+03 1.73510716e+04
  1.82177695e+02]
 [3.34632720e+04 2.63471061e+03 1.55437716e+03 1.29813425e+04
  1.36297695e+02]
 [1.95842860e+04 1.54195699e+03 9.09694871e+02 7.59729428e+03
  7.97678434e+01]
 [1.14369647e+04 9.00482540e+02 5.31249803e+02 4.43671963e+03
  4.65833682e+01]]
```

The Chi-Square Statistic is 42275.23 with 24 degrees of freedom, resulting in a p-value of 0.0. The observed data significantly differs from the expected frequencies in the contingency table.

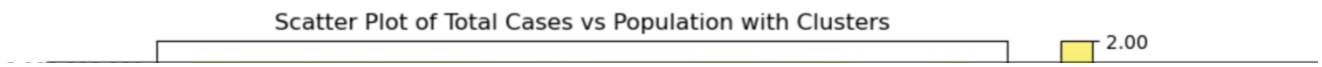
In [11]:
```
# Apply k-means clustering for different values of k
inertia = []
for k in range(1, 11):
    kmeans = KMeans(n_clusters=k, random_state=42)
    kmeans.fit(features_scaled)
    inertia.append(kmeans.inertia_)
```

In [12]:
```
# Plot the elbow curve
plt.plot(range(1, 11), inertia, marker='o')
plt.title('Elbow Method for Optimal k')
plt.xlabel('Number of Clusters (k)')
plt.ylabel('Inertia')
plt.show()
```



In [15]:
```
data['cluster_label'] = kmeans.labels_
```

In [33]:
```
import matplotlib.pyplot as plt
from matplotlib.ticker import FuncFormatter

# Assuming 'data' is your DataFrame with 'population' not in scientific notation
plt.figure(figsize=(10, 6))

# Scatter plot with color-coded clusters
scatter = plt.scatter(data['total_cases'], data['population'], c=data['cluster_label'], cmap='viridis',alpha=0.6, s=

# Add labels and title
plt.xlabel('Total Cases')
plt.ylabel('population_density')
plt.title('Scatter Plot of Total Cases vs Population with Clusters')

# Add colorbar
colorbar = plt.colorbar(scatter)
colorbar.set_label('Cluster Label')

# Format the population axis tick labels
plt.gca().yaxis.set_major_formatter(FuncFormatter(lambda x, _: '{:,.0f}'.format(x)))

# Show the plot
plt.show()
```



Classification

In [19]:
```
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score
```

In [20]:
```
# Randomly select 10,000 rows

df_sample = data.sample(n=10000, random_state=42)  # Set random_state for reproducibility
```

In [21]:
```
# Assuming 'target' is your target variable
target = df_sample['total_deaths']
```

In [22]:
```
input_variables = df_sample[['total_cases', 'new_cases', 'population']]
```

In [23]:
```
# Splitting the data into training and testing sets

X_train, X_test, y_train, y_test = train_test_split(input_variables, target, test_size=0.2)
```

In [24]:
```
#creating logistic regression model

clf = LogisticRegression(max_iter=100)
```

In [25]:
```
from sklearn.preprocessing import StandardScaler

scaler = StandardScaler()
X_train_scaled = scaler.fit_transform(X_train)
```

In [26]:
```
from sklearn.pipeline import make_pipeline

logistic_regression_model = make_pipeline(StandardScaler(), LogisticRegression())
```

In [27]:
```
X_train_scaled.mean(axis=0)
```

Out[27]:
```
array([-5.32907052e-18,  5.55111512e-18,  1.33226763e-17])
```

In [28]:
```
clf.fit(X_train, y_train)
```

```
/Applications/anaconda3/lib/python3.9/site-packages/sklearn/linear_model/_logistic.py:444: ConvergenceWarning: lbfg
s failed to converge (status=1):
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html
Please also refer to the documentation for alternative solver options:
    https://scikit-learn.org/stable/modules/linear_model.html#logistic-regression
  n_iter_i = _check_optimize_result(
```

Out[28]:
```
LogisticRegression()
```

In a Jupyter environment, please rerun this cell to show the HTML representation or trust the notebook.
On GitHub, the HTML representation is unable to render, please try loading this page with nbviewer.org.

In [29]:
```
y_pred = clf.predict(X_test)
```

In [30]:
```
#evaluating the model

accuracy = accuracy_score(y_test, y_pred)

print("Accuracy:", accuracy)
```

```
Accuracy: 0.1755
```

Second Phase of CRISP-DM

Objectives : 1. Data Transformation 2. Feature engineering 3. Co-relation analysis

Data Transformation

```
# Data Transformation

import pandas as pd
import numpy as np
data = pd.read_csv('cleaned_dataset.csv')
data.head()
```

	iso_code	continent	location	date	total_cases	new_cases	new_cases_smoothed	total_deaths	new_deaths	new_deaths_smoothed	...	male_smokers	handwashing_facilities	hospital_beds_per_thousand	life_expectancy	human_development_index	population	excess_mortality_cumulative_absolute	excess_mortality_cumulative	excess_mortality	excess_mortality_cumulative_per_million
0	AFG	Asia	Afghanistan	2020-01-03	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	37.746	0.5	64.83	0.511	41128772.0	0.0	0.0	0.0	0.0
1	AFG	Asia	Afghanistan	2020-01-04	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	37.746	0.5	64.83	0.511	41128772.0	0.0	0.0	0.0	0.0
2	AFG	Asia	Afghanistan	2020-01-05	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	37.746	0.5	64.83	0.511	41128772.0	0.0	0.0	0.0	0.0

	iso_code	continent	location	date	total_cases	new_cases	new_cases_smoothed	total_deaths	new_deaths	new_deaths_smoothed	...	male_smokers	handwashing_facilities	hospital_beds_per_thousand	life_expectancy	human_development_index	population	excess_mortality_cumulative_absolute	excess_mortality_cumulative	excess_mortality	excess_mortality_cumulative_per_million
3	AFG	Asia	Afghanistan	2020-01-06	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	37.746	0.5	64.83	0.511	41128772.0	0.0	0.0	0.0	0.0
4	AFG	Asia	Afghanistan	2020-01-07	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	37.746	0.5	64.83	0.511	41128772.0	0.0	0.0	0.0	0.0

5 rows × 67 columns

```
data.isnull().sum().sum()
0
data.info()
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 313267 entries, 0 to 313266
Data columns (total 67 columns):
 #   Column                                      Non-Null Count   Dtype
---  ------                                      --------------   -----
 0   iso_code                                    313267 non-null  object
 1   continent                                   313267 non-null  object
 2   location                                    313267 non-null  object
 3   date                                        313267 non-null  object
 4   total_cases                                 313267 non-null  float64
 5   new_cases                                   313267 non-null  float64
 6   new_cases_smoothed                          313267 non-null  float64
 7   total_deaths                                313267 non-null  float64
 8   new_deaths                                  313267 non-null  float64
 9   new_deaths_smoothed                         313267 non-null  float64
 10  total_cases_per_million                     313267 non-null  float64
 11  new_cases_per_million                       313267 non-null  float64
 12  new_cases_smoothed_per_million              313267 non-null  float64
 13  total_deaths_per_million                    313267 non-null  float64
 14  new_deaths_per_million                      313267 non-null  float64
 15  new_deaths_smoothed_per_million             313267 non-null  float64
 16  reproduction_rate                           313267 non-null  float64
 17  icu_patients                                313267 non-null  float64
```

```
 18  icu_patients_per_million                     313267 non-null  float64
 19  hosp_patients                                313267 non-null  float64
 20  hosp_patients_per_million                    313267 non-null  float64
 21  weekly_icu_admissions                        313267 non-null  float64
 22  weekly_icu_admissions_per_million            313267 non-null  float64
 23  weekly_hosp_admissions                       313267 non-null  float64
 24  weekly_hosp_admissions_per_million           313267 non-null  float64
 25  total_tests                                  313267 non-null  float64
 26  new_tests                                    313267 non-null  float64
 27  total_tests_per_thousand                     313267 non-null  float64
 28  new_tests_per_thousand                       313267 non-null  float64
 29  new_tests_smoothed                           313267 non-null  float64
 30  new_tests_smoothed_per_thousand              313267 non-null  float64
 31  positive_rate                                313267 non-null  float64
 32  tests_per_case                               313267 non-null  float64
 33  tests_units                                  313267 non-null  object
 34  total_vaccinations                           313267 non-null  float64
 35  people_vaccinated                            313267 non-null  float64
 36  people_fully_vaccinated                      313267 non-null  float64
 37  total_boosters                               313267 non-null  float64
 38  new_vaccinations                             313267 non-null  float64
 39  new_vaccinations_smoothed                    313267 non-null  float64
 40  total_vaccinations_per_hundred               313267 non-null  float64
 41  people_vaccinated_per_hundred                313267 non-null  float64
 42  people_fully_vaccinated_per_hundred          313267 non-null  float64
 43  total_boosters_per_hundred                   313267 non-null  float64
 44  new_vaccinations_smoothed_per_million        313267 non-null  float64
 45  new_people_vaccinated_smoothed               313267 non-null  float64
 46  new_people_vaccinated_smoothed_per_hundred  313267 non-null  float64
 47  stringency_index                             313267 non-null  float64
 48  population_density                           313267 non-null  float64
 49  median_age                                   313267 non-null  float64
 50  aged_65_older                                313267 non-null  float64
 51  aged_70_older                                313267 non-null  float64
 52  gdp_per_capita                               313267 non-null  float64
 53  extreme_poverty                              313267 non-null  float64
 54  cardiovasc_death_rate                        313267 non-null  float64
 55  diabetes_prevalence                          313267 non-null  float64
 56  female_smokers                               313267 non-null  float64
 57  male_smokers                                 313267 non-null  float64
 58  handwashing_facilities                       313267 non-null  float64
 59  hospital_beds_per_thousand                   313267 non-null  float64
 60  life_expectancy                              313267 non-null  float64
 61  human_development_index                      313267 non-null  float64
 62  population                                   313267 non-null  float64
 63  excess_mortality_cumulative_absolute         313267 non-null  float64
 64  excess_mortality_cumulative                  313267 non-null  float64
 65  excess_mortality                             313267 non-null  float64
 66  excess_mortality_cumulative_per_million      313267 non-null  float64
dtypes: float64(62), object(5)
memory usage: 160.1+ MB
data.describe
<bound method NDFrame.describe of        iso_code continent     location        date  total_
cases  new_cases  \
0           AFG      Asia  Afghanistan  2020-01-03          0.0        0.0
1           AFG      Asia  Afghanistan  2020-01-04          0.0        0.0
2           AFG      Asia  Afghanistan  2020-01-05          0.0        0.0
3           AFG      Asia  Afghanistan  2020-01-06          0.0        0.0
```

```
4           AFG      Asia  Afghanistan  2020-01-07          0.0        0.0
...         ...       ...          ...         ...          ...        ...
313262      ZWE    Africa     Zimbabwe  2023-05-20     264848.0        0.0
313263      ZWE    Africa     Zimbabwe  2023-05-21     264848.0        0.0
313264      ZWE    Africa     Zimbabwe  2023-05-22     264848.0        0.0
313265      ZWE    Africa     Zimbabwe  2023-05-23     264848.0        0.0
313266      ZWE    Africa     Zimbabwe  2023-05-24     264848.0        0.0

        new_cases_smoothed  total_deaths  new_deaths  new_deaths_smoothed  \
0                    0.000           0.0         0.0                  0.0
1                    0.000           0.0         0.0                  0.0
2                    0.000           0.0         0.0                  0.0
3                    0.000           0.0         0.0                  0.0
4                    0.000           0.0         0.0                  0.0
...                    ...           ...         ...                  ...
313262               3.857        5690.0         0.0                  0.0
313263               1.000        5690.0         0.0                  0.0
313264               0.000        5690.0         0.0                  0.0
313265               0.000        5690.0         0.0                  0.0
313266               0.000        5690.0         0.0                  0.0

        ...  male_smokers  handwashing_facilities  hospital_beds_per_thousand  \
0       ...           0.0                  37.746                         0.5
1       ...           0.0                  37.746                         0.5
2       ...           0.0                  37.746                         0.5
3       ...           0.0                  37.746                         0.5
4       ...           0.0                  37.746                         0.5
...     ...           ...                     ...                         ...
313262  ...          30.7                  36.791                         1.7
313263  ...          30.7                  36.791                         1.7
313264  ...          30.7                  36.791                         1.7
313265  ...          30.7                  36.791                         1.7
313266  ...          30.7                  36.791                         1.7

        life_expectancy  human_development_index  population  \
0                 64.83                    0.511  41128772.0
1                 64.83                    0.511  41128772.0
2                 64.83                    0.511  41128772.0
3                 64.83                    0.511  41128772.0
4                 64.83                    0.511  41128772.0
...                 ...                      ...         ...
313262            61.49                    0.571  16320539.0
313263            61.49                    0.571  16320539.0
313264            61.49                    0.571  16320539.0
313265            61.49                    0.571  16320539.0
313266            61.49                    0.571  16320539.0

        excess_mortality_cumulative_absolute  excess_mortality_cumulative  \
0                                        0.0                          0.0
1                                        0.0                          0.0
2                                        0.0                          0.0
3                                        0.0                          0.0
4                                        0.0                          0.0
...                                      ...                          ...
313262                                   0.0                          0.0
313263                                   0.0                          0.0
313264                                   0.0                          0.0
313265                                   0.0                          0.0
```

```
313266                                  0.0                           0.0

        excess_mortality  excess_mortality_cumulative_per_million
0                    0.0                                      0.0
1                    0.0                                      0.0
2                    0.0                                      0.0
3                    0.0                                      0.0
4                    0.0                                      0.0
...                  ...                                      ...
313262               0.0                                      0.0
313263               0.0                                      0.0
313264               0.0                                      0.0
313265               0.0                                      0.0
313266               0.0                                      0.0

[313267 rows x 67 columns]>
data = data.drop_duplicates()
data.shape
(313267, 67)
# checking the columns for the presence of categorical values

categorical_columns = data.select_dtypes(include=['object', 'category'])

# Print the list of categorical columns

print("Categorical Columns:")
print(categorical_columns.columns)
Categorical Columns:
Index(['iso_code', 'continent', 'location', 'date', 'tests_units'], dtype='object')
#converting categorical values into numerical except 'date'

from sklearn.preprocessing import LabelEncoder

label_encoder = LabelEncoder()
#for iso_code using label encoding

data['iso_code'] = label_encoder.fit_transform(data['iso_code'])
#for continent using label encoding

data['continent'] = label_encoder.fit_transform(data['continent'])
#for loaction using one-hot encoding

data = pd.get_dummies(data, columns=['location'])
#for test_units using one-hot encoding

data = pd.get_dummies(data, columns=['tests_units'])
```

Feature engineering for column - 'date'

```
# Convert the 'date' column to a datetime data type
data['date'] = pd.to_datetime(data['date'])

# Extract year, month, and day features
data['year'] = data['date'].dt.year
data['month'] = data['date'].dt.month
data['day'] = data['date'].dt.day

# Extract day of the week as a numerical value (0=Monday, 6=Sunday)
data['day_of_week'] = data['date'].dt.dayofweek
```

```
# Extract week of the year
data['week_of_year'] = data['date'].dt.isocalendar().week

# Extract quarter
data['quarter'] = data['date'].dt.quarter

# Extract whether it's a weekend (1) or a weekday (0)
data['is_weekend'] = data['day_of_week'].apply(lambda x: 1 if x in [5, 6] else 0)

# Extract the day of the year
data['day_of_year'] = data['date'].dt.dayofyear
data.head()
```

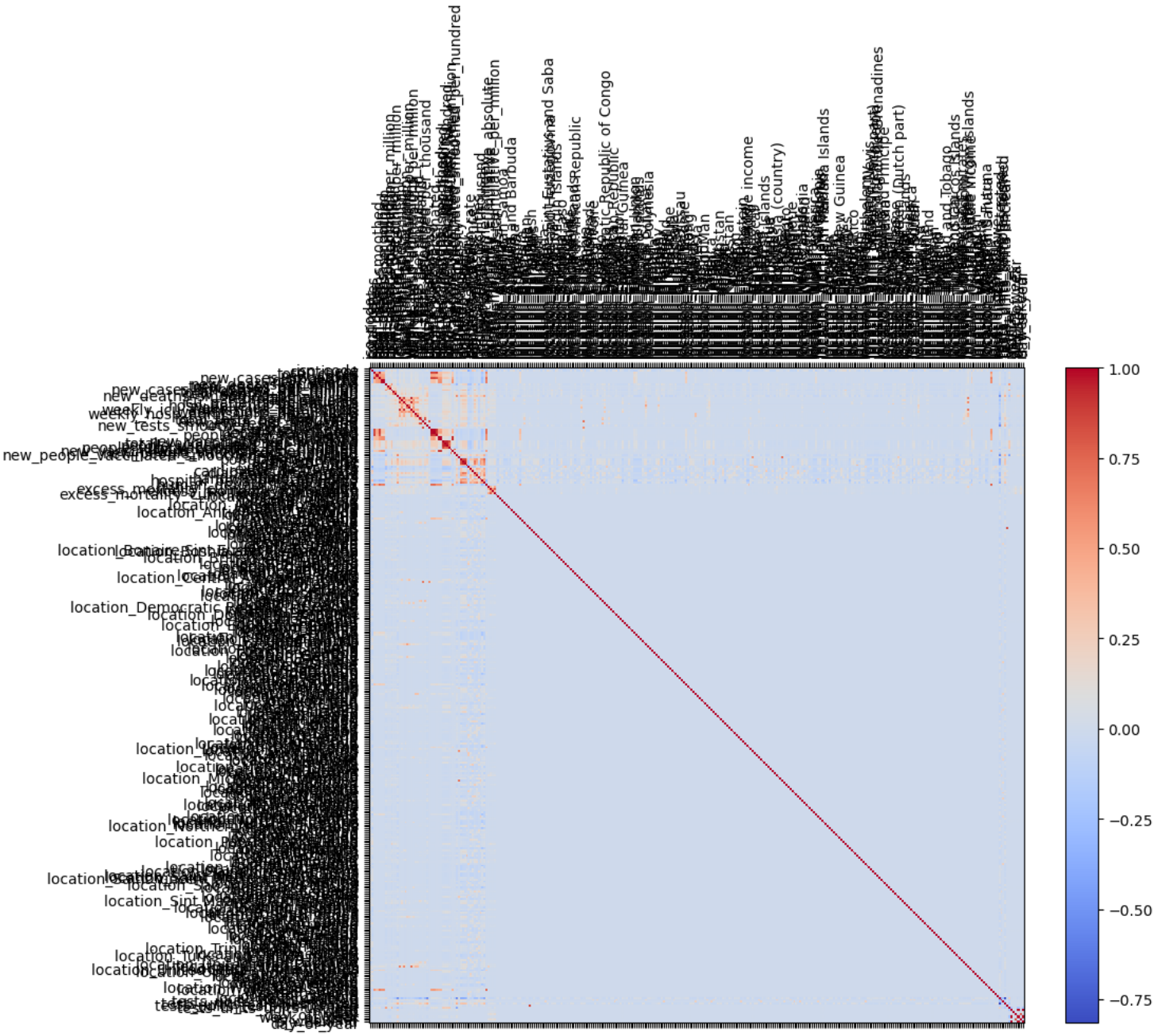
	iso_code	continent	date	total_cases	new_cases	new_cases_smoothed	total_deaths	new_deaths	new_deaths_smoothed	total_cases_per_million	...	tests_units_tests performed	tests_units_units unclear	year	month	day	day_of_week	week_of_year	quarter	is_weekend	day_of_year
0	1	2	2020-01-03	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0	0	2020	1	3	4	1	1	0	3
1	1	2	2020-01-04	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0	0	2020	1	4	5	1	1	1	4
2	1	2	2020-01-05	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0	0	2020	1	5	6	1	1	1	5
3	1	2	2020-	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0	0	2020	1	6	0	2	1	0	6

	iso_code	continent	date	total_cases	new_cases	new_cases_smoothed	total_deaths	new_deaths	new_deaths_smoothed	total_cases_per_million	...	tests_units_tests_tests performed	tests_units_units unclear	year	month	day	day_of_week	week_of_year	quarter	is_weekend	day_of_year
			01-06																		
4	1	2	2020-01-07	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0	0	2020	1	7	1	2	1	0	7

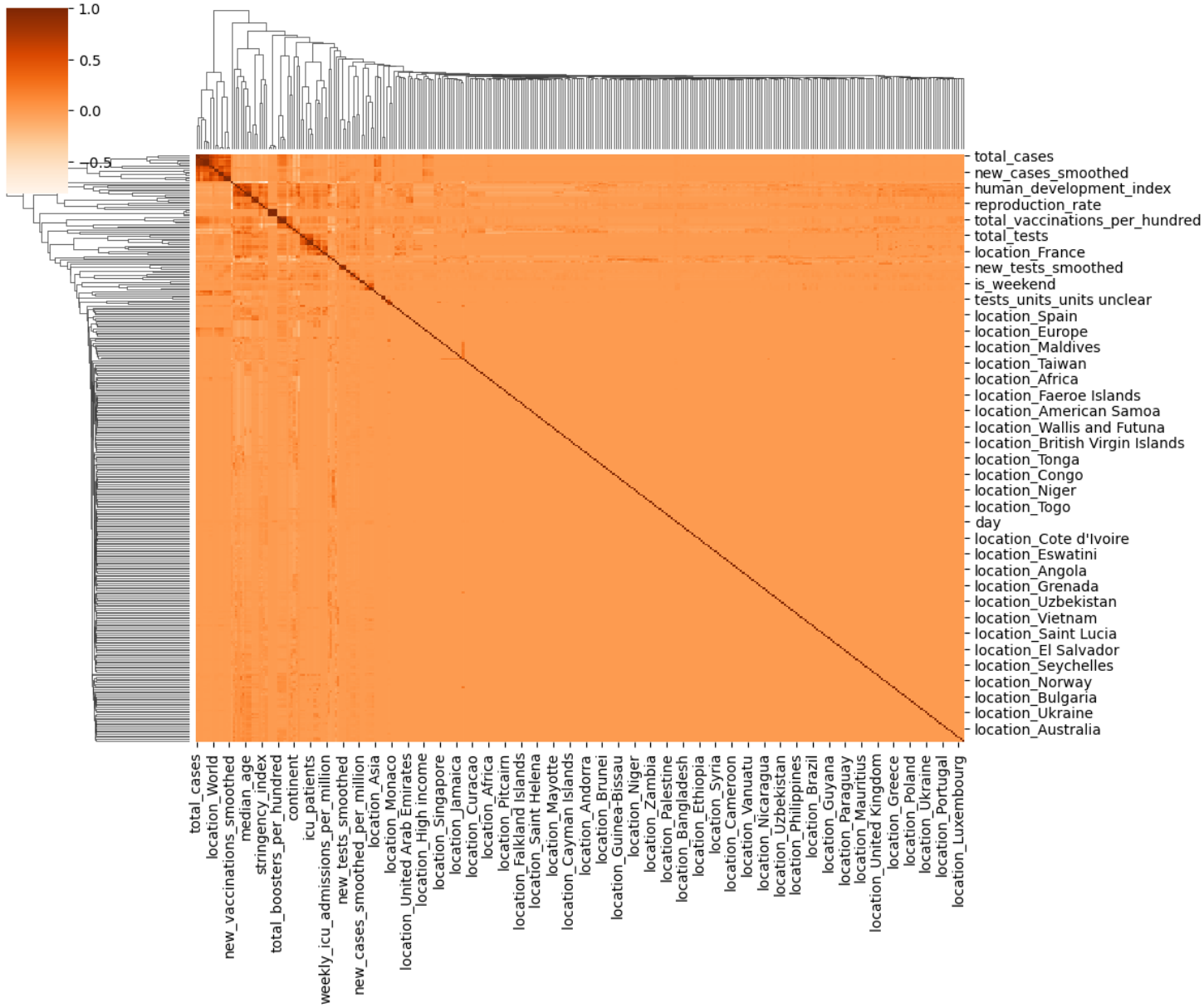
5 rows × 333 columns

Correlation Analysis

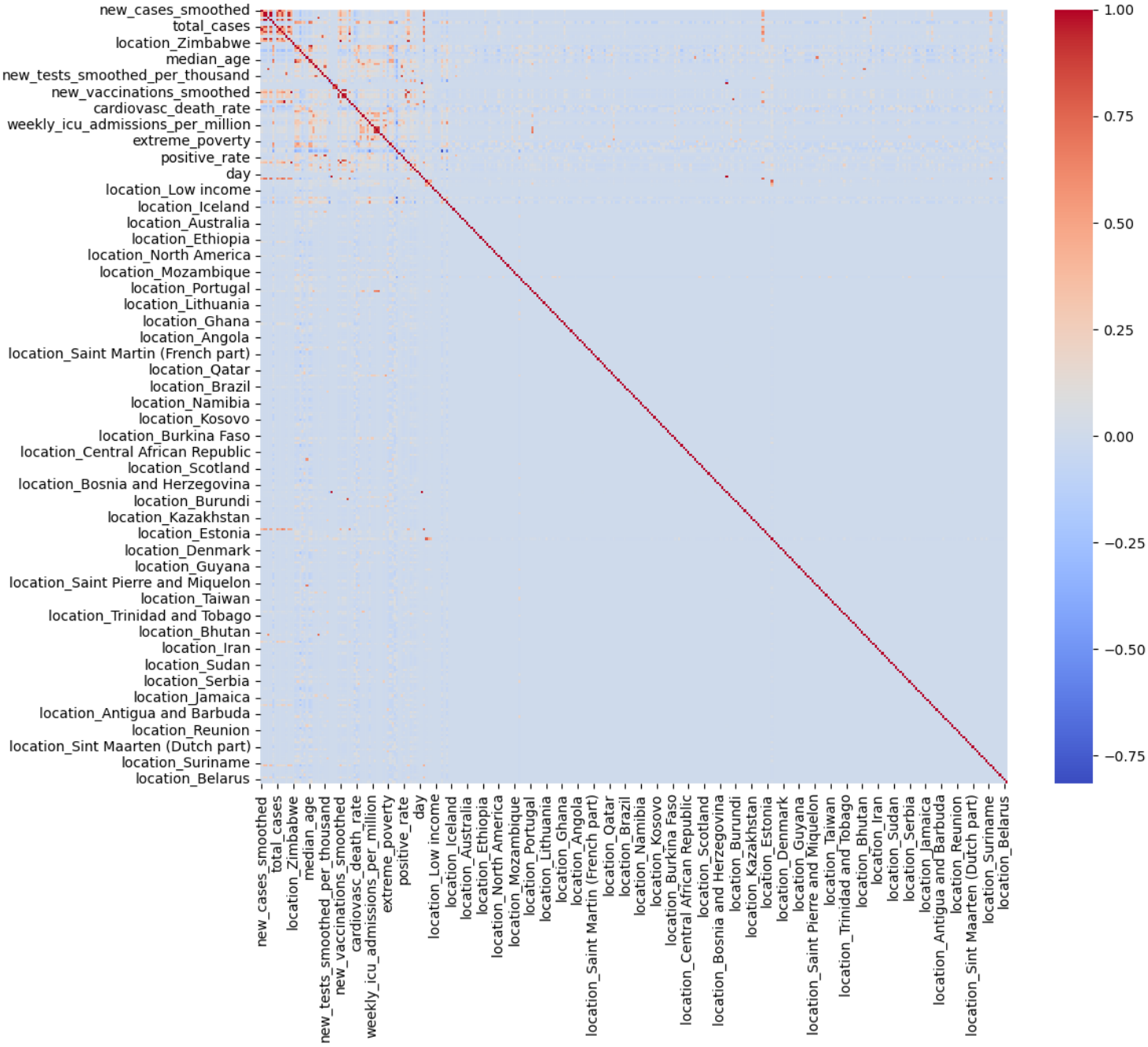
```
import matplotlib.pyplot as plt
# Calculating the Pearson correlation matrix
correlation_matrix = data.corr()
# Ploting a correlation matrix heatmap

plt.figure(figsize=(10, 8))
plt.matshow(correlation_matrix, cmap='coolwarm', fignum=1)
plt.colorbar()
plt.xticks(range(len(correlation_matrix.columns)), correlation_matrix.columns, rotation=90)
plt.yticks(range(len(correlation_matrix.columns)), correlation_matrix.columns)
plt.show()
```




```
import seaborn as sns
# Create a hierarchical clustering of columns

g = sns.clustermap(correlation_matrix, cmap='Oranges', figsize=(12, 10))
```



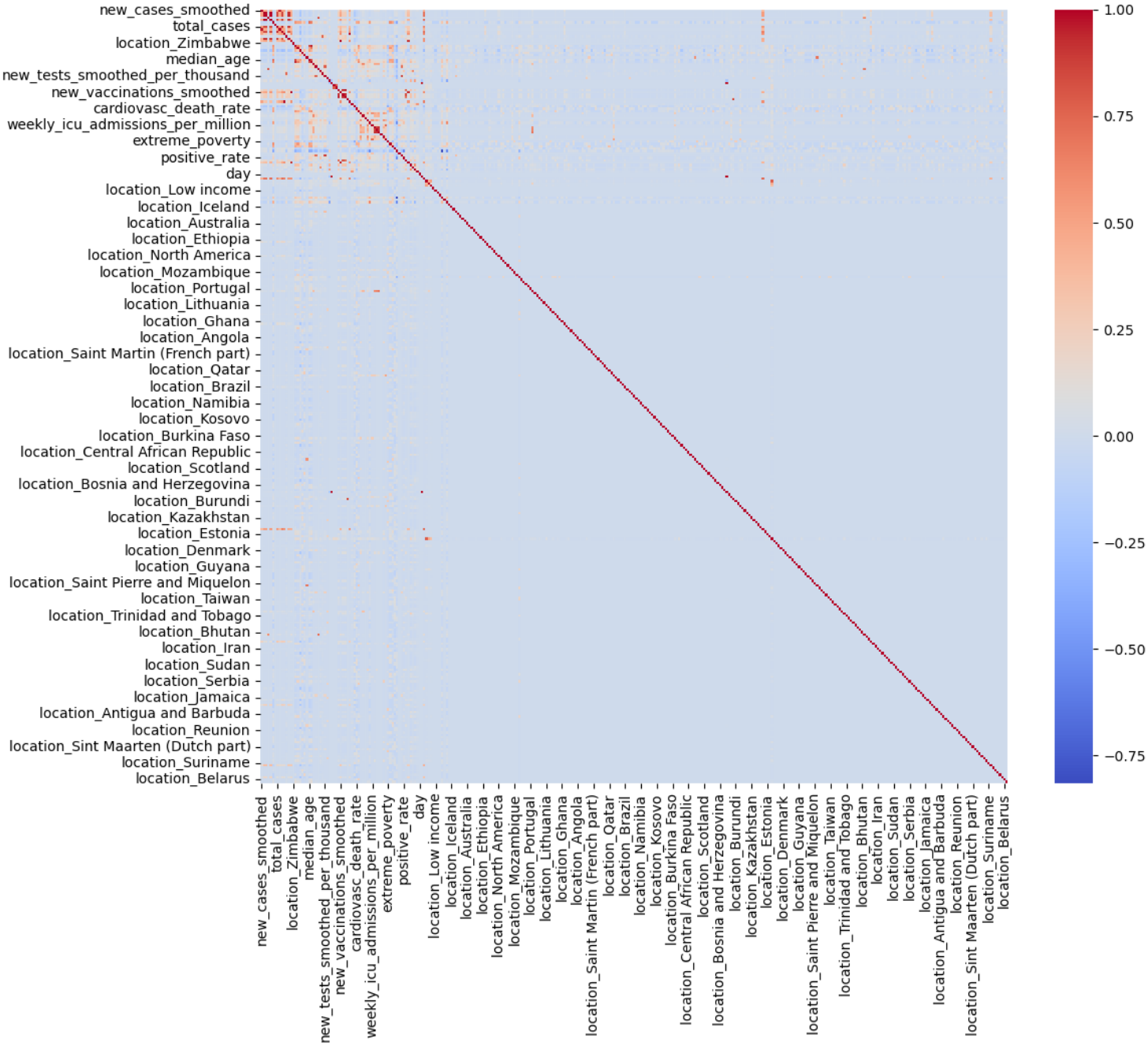

```
# Reordering the columns based on clustering

reordered_columns = data.columns[g.dendrogram_row.reordered_ind]
data = data[reordered_columns]
# Plotting the reordered correlation matrix

correlation_matrix_reordered = data.corr()
plt.figure(figsize=(12, 10))
sns.heatmap(correlation_matrix_reordered, cmap='coolwarm', annot=False)

# Save the heatmap
plt.savefig('correlation_heatmap.png')

plt.show()
```




```
import seaborn as sns
import matplotlib.pyplot as plt
from IPython.display import display, HTML

# Assuming you have a DataFrame named 'data' with the correlation matrix
correlation_matrix_reordered = data.corr()

# Plotting the reordered correlation matrix
plt.figure(figsize=(12, 10))
sns.heatmap(correlation_matrix_reordered, cmap='coolwarm', annot=False)

# Save the plot as an image file
plt.savefig('correlation_heatmap.png')

# Display a download link
display(HTML('<a href="correlation_heatmap_1.png" download>Click here to download the
heatmap</a>'))
```

```
# Show the plot
plt.show()
```

Click here to download the heatmap



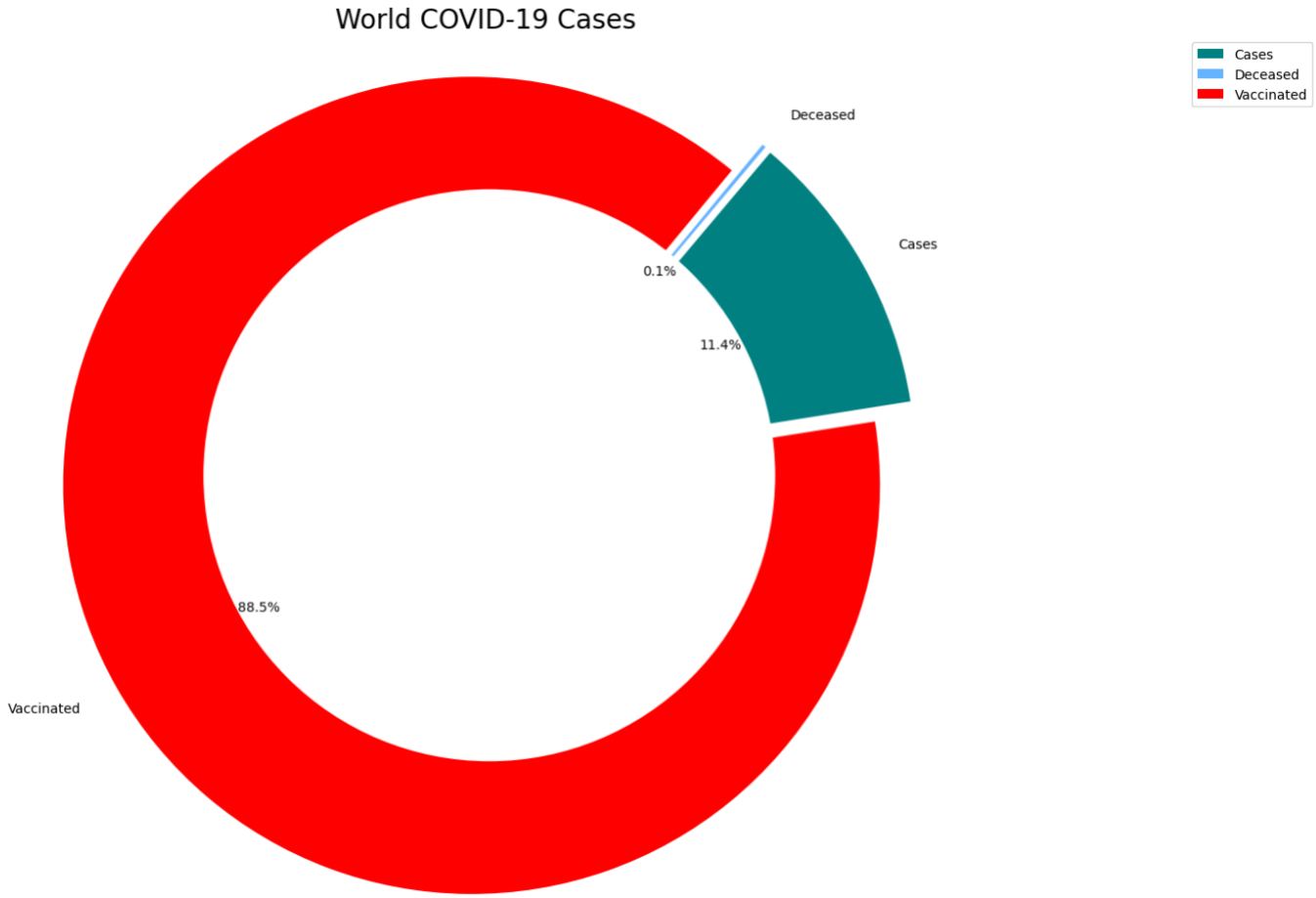

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
%matplotlib inline
world_data = pd.read_csv('world_covid_dataset.csv')
world_data.head(3)
```

	iso_code	continent	location	date	total_cases	new_cases	new_cases_smoothed	total_deaths	new_deaths	new_deaths_smoothed	...	male_smokers	handwashing_facilities	hospital_beds_per_thousand	life_expectancy	human_development_index	population	excess_mortality_cumulative_absolute	excess_mortality_cumulative	excess_mortality	excess_mortality_cumulative_per_million
0	OWID_WRL	World	World	2020-01-03	0	0	0.0	0	0	0.0	...	34.635	60.13	2.705	72.58	0.737	7975105024	0	0	0	0
1	OWID_WRL	World	World	2020-01-04	3	3	0.0	0	0	0.0	...	34.635	60.13	2.705	72.58	0.737	7975105024	0	0	0	0
2	OWID_WRL	World	World	2020-01-05	3	0	0.0	0	0	0.0	...	34.635	60.13	2.705	72.58	0.737	7975105024	0	0	0	0

3 rows × 50 columns

```
world_data.isnull().sum().sum()
0
world_cases = world_data['total_cases'].sum()
world_deaths = world_data['total_deaths'].sum()
world_vaccinated = world_data['people_fully_vaccinated'].sum()


labels = ['Cases','Deceased','Vaccinated']
sizes = [world_cases,world_deaths,world_vaccinated]
color= ['teal','#66b3ff','red']
explode = []

for i in labels:
    explode.append(0.05)

plt.figure(figsize= (15,10))
```

```
plt.pie(sizes, labels=labels, autopct='%1.1f%%', startangle=9, explode =explode,colors =
color)
centre_circle = plt.Circle((0,0),0.70,fc='white')

fig = plt.gcf()
fig.gca().add_artist(centre_circle)
plt.legend(labels, loc='upper right', bbox_to_anchor=(1.2, 1))
plt.title('World COVID-19 Cases',fontsize = 20)
plt.axis('equal')
plt.tight_layout()
```




```
africa_data = pd.read_csv('Africa_covid_dataset.csv')
africa_data.head(3)
```

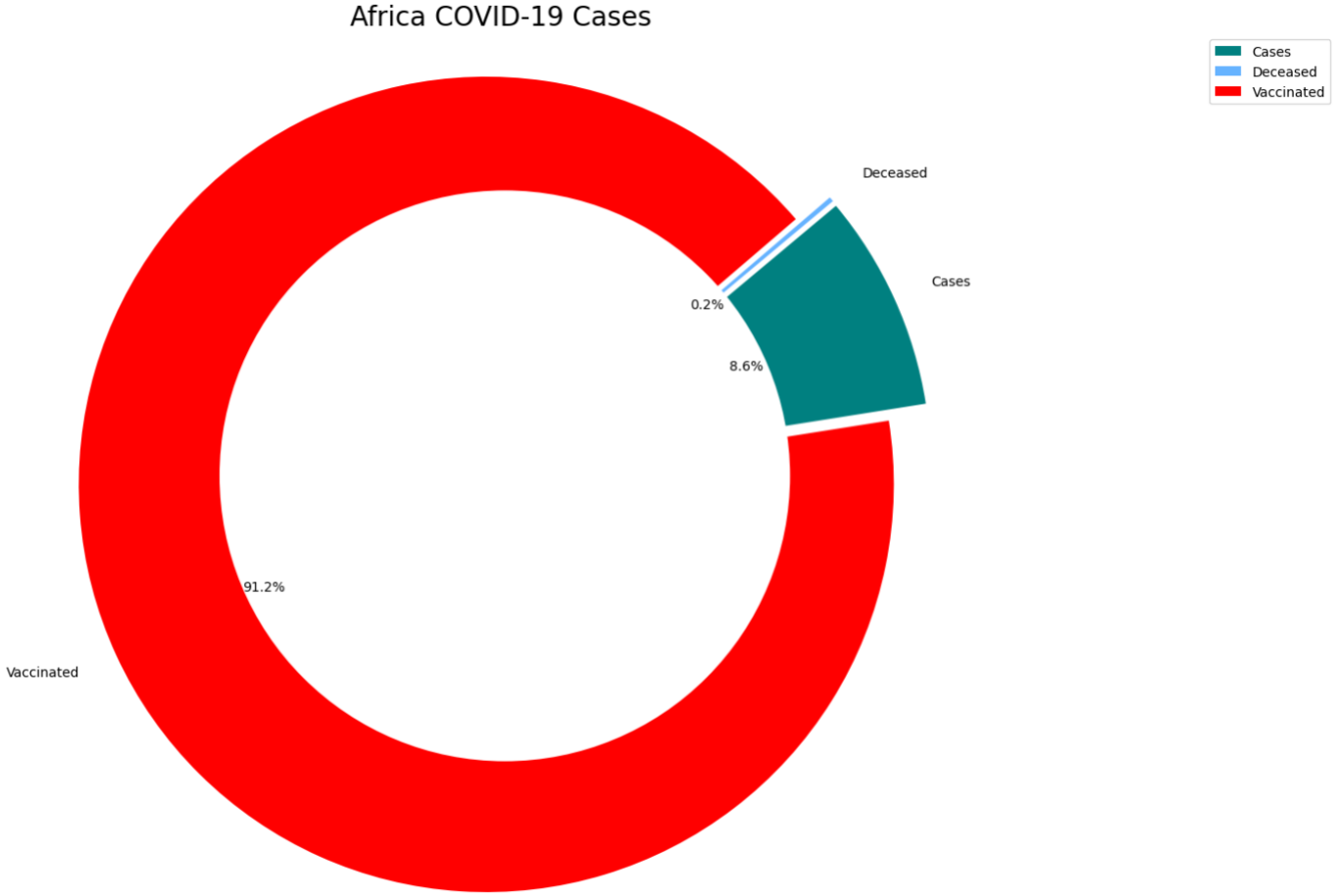
	iso_code	continent	location	date	total_cases	new_cases	new_cases_smoothed	total_deaths	new_deaths	new_deaths_smoothed	...	male_smokers	handwashing_facilities	hospital_beds_per_thousand	life_expectancy	human_development_index	population	excess_mortality_cumulative_absolute	excess_mortality_cumulative	excess_mortality	excess_mortality_cumulative_per_million
0	OWID_A	Africa	Africa	2020-01-	0	0	0.0	0	0	0.0	...	0.0	0.0	0.0	0.0	0.0	1426736614	0.0	0.0	0.0	0.0

	iso_code	continent	location	date	total_cases	new_cases	new_cases_smoothed	total_deaths	new_deaths	new_deaths_smoothed	...	male_smokers	handwashing_facilities	hospital_beds_per_thousand	life_expectancy	human_development_index	population	excess_mortality_cumulative_absolute	excess_mortality_cumulative	excess_mortality	excess_mortality_cumulative_per_million
	FR			03																	
1	OWID_AFR	Africa	Africa	2020-01-04	0	0	0.0	0	0	0.0	...	0.0	0.0	0.0	0.0	0.0	1426736614	0.0	0.0	0.0	0.0
2	OWID_AFR	Africa	Africa	2020-01-05	0	0	0.0	0	0	0.0	...	0.0	0.0	0.0	0.0	0.0	1426736614	0.0	0.0	0.0	0.0

3 rows × 67 columns

```
africa_data.isnull().sum().sum()
0
africa_cases = africa_data['total_cases'].sum()
africa_deaths = africa_data['total_deaths'].sum()
africa_vaccinated = africa_data['people_fully_vaccinated'].sum()


labels = ['Cases','Deceased','Vaccinated']
sizes = [africa_cases,africa_deaths,africa_vaccinated]
color= ['teal','#66b3ff','red']
explode = []

for i in labels:
    explode.append(0.05)

plt.figure(figsize= (15,10))
plt.pie(sizes, labels=labels, autopct='%1.1f%%', startangle=9, explode =explode,colors =
color)
centre_circle = plt.Circle((0,0),0.70,fc='white')

fig = plt.gcf()
fig.gca().add_artist(centre_circle)
```

```
plt.legend(labels, loc='upper right', bbox_to_anchor=(1.2, 1))
plt.title('Africa COVID-19 Cases',fontsize = 20)
plt.axis('equal')
plt.tight_layout()
```



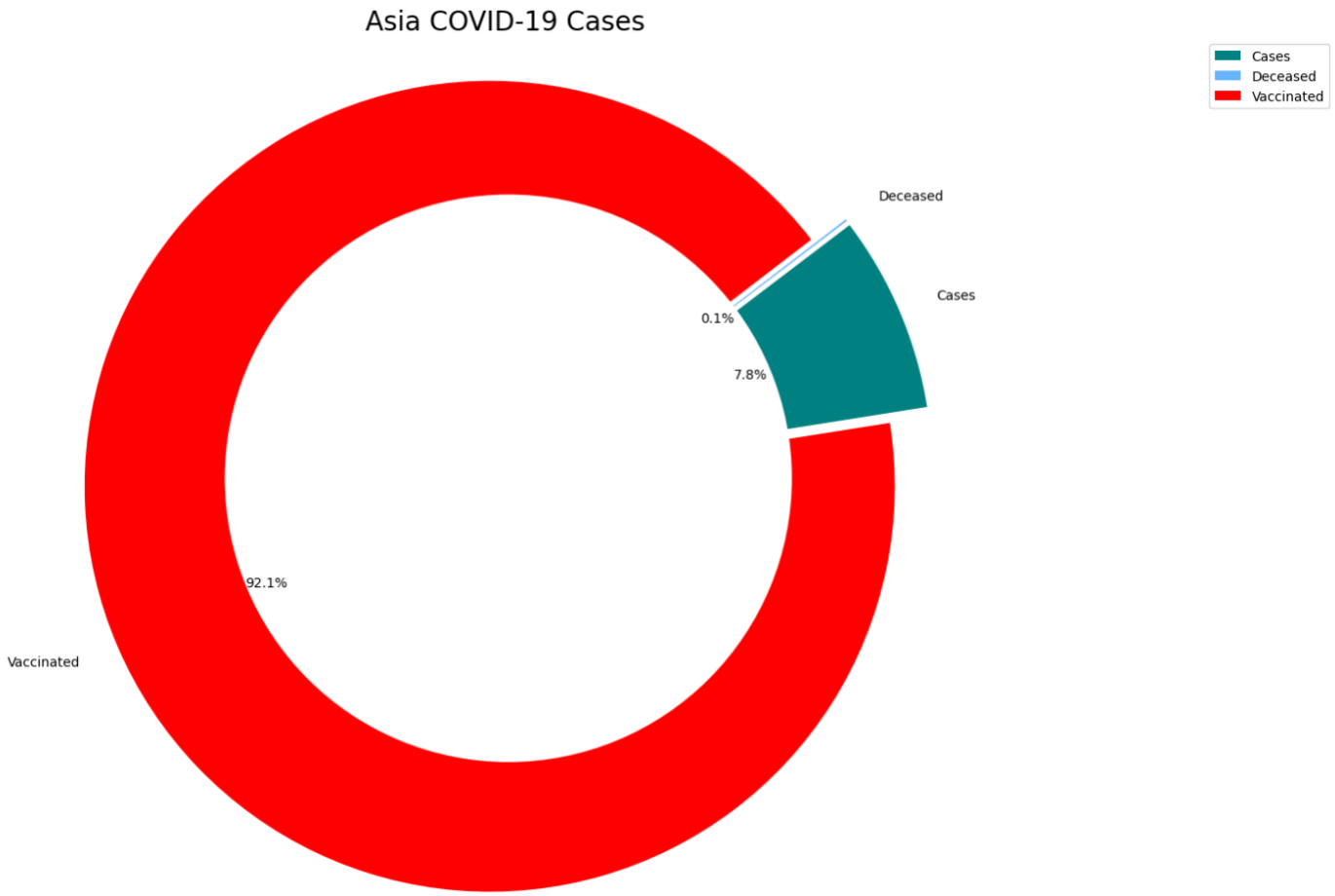

```
asia_data = pd.read_csv('Asia_covid_dataset.csv')
asia_data.head(3)
```

	iso_code	continent	location	date	total_cases	new_cases	new_cases_smoothed	total_deaths	new_deaths	new_deaths_smoothed	...	male_smokers	handwashing_facilities	hospital_beds_per_thousand	life_expectancy	human_development_index	population	excess_mortality_cumulative_absolute	excess_mortality_cumulative	excess_mortality	excess_mortality_cumulative_per_million
0	AFG	Asia	Afghanistan	2020-01-03	0	0	0.0	0	0	0.0	...	0.0	37.746	0.5	64.83	0.511	41128772	0.0	0.0	0.0	0.0

	iso_code	continent	location	date	total_cases	new_cases	new_cases_smoothed	total_deaths	new_deaths	new_deaths_smoothed	...	male_smokers	handwashing_facilities	hospital_beds_per_thousand	life_expectancy	human_development_index	population	excess_mortality_cumulative_absolute	excess_mortality_cumulative	excess_mortality	excess_mortality_cumulative_per_million
1	AFG	Asia	Afghanistan	2020-01-04	0	0	0.0	0	0	0.0	...	0.0	37.746	0.5	64.83	0.511	41128772	0.0	0.0	0.0	0.0
2	AFG	Asia	Afghanistan	2020-01-05	0	0	0.0	0	0	0.0	...	0.0	37.746	0.5	64.83	0.511	41128772	0.0	0.0	0.0	0.0

3 rows × 67 columns

```
asia_data.isnull().sum().sum()
0
asia_cases = asia_data['total_cases'].sum()
asia_deaths = asia_data['total_deaths'].sum()
asia_vaccinated = asia_data['people_fully_vaccinated'].sum()


labels = ['Cases','Deceased','Vaccinated']
sizes = [asia_cases,asia_deaths,asia_vaccinated]
color= ['teal','#66b3ff','red']
explode = []

for i in labels:
    explode.append(0.05)

plt.figure(figsize= (15,10))
plt.pie(sizes, labels=labels, autopct='%1.1f%%', startangle=9, explode =explode,colors =
color)
centre_circle = plt.Circle((0,0),0.70,fc='white')

fig = plt.gcf()
fig.gca().add_artist(centre_circle)
plt.title('Asia COVID-19 Cases',fontsize = 20)
plt.legend(labels, loc='upper right', bbox_to_anchor=(1.2, 1))
plt.axis('equal')
plt.tight_layout()
```



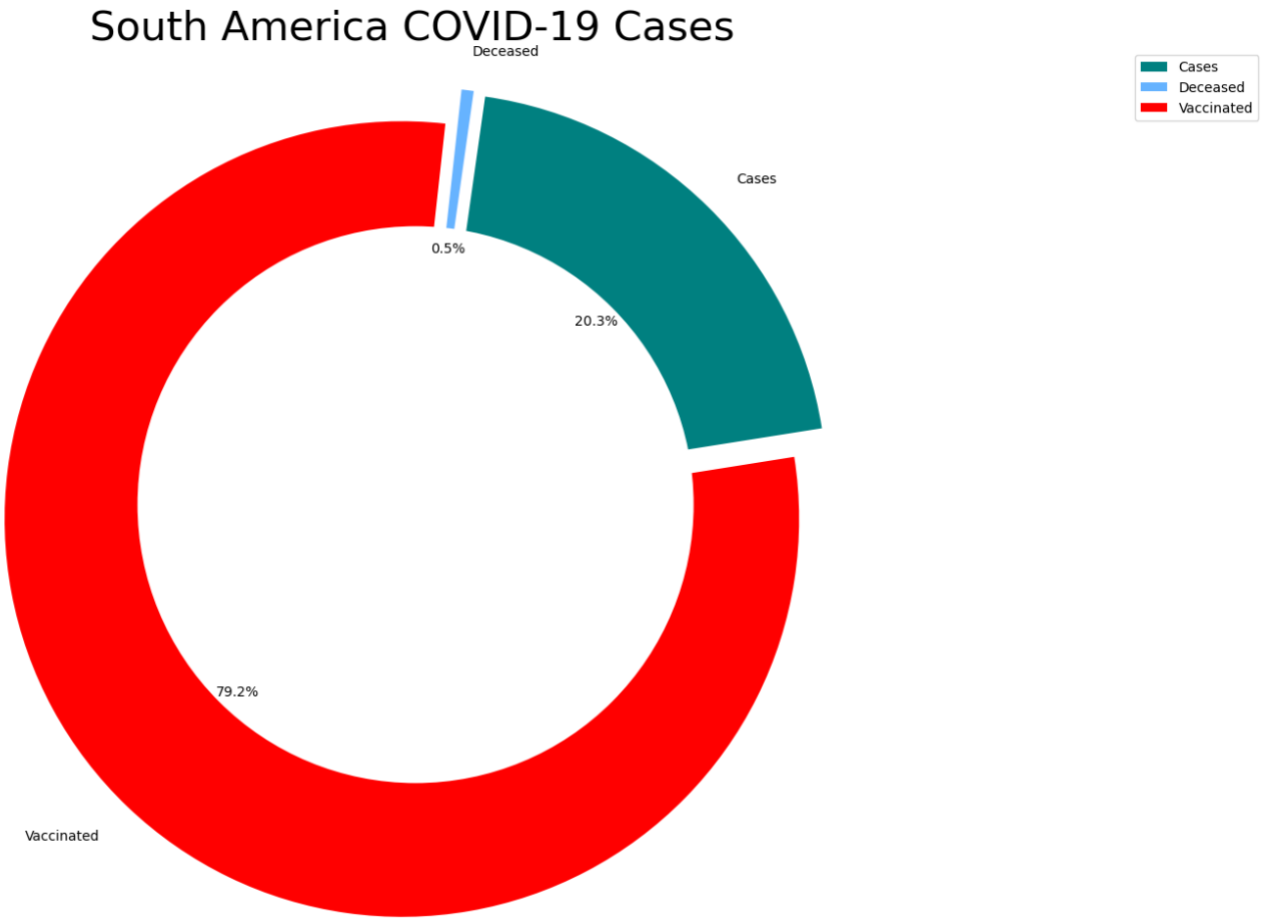

```
sa_data = pd.read_csv('South_America_covid_dataset.csv')
sa_data.head(3)
```

	iso_code	continent	location	date	total_cases	new_cases	new_cases_smoothed	total_deaths	new_deaths	new_deaths_smoothed	...	male_smokers	handwashing_facilities	hospital_beds_per_thousand	life_expectancy	human_development_index	population	excess_mortality_cumulative_absolute	excess_mortality_cumulative	excess_mortality	excess_mortality_cumulative_per_million
0	ARG	South America	Argentina	2020-01-01	0	0	0.0	0	0	0.0	...	27.7	0.0	5.0	76.67	0.845	45510324	0.0	0.0	0.0	0.0
1	ARG	South Amer	Argent	2020-01	0	0	0.0	0	0	0.0	...	27.7	0.0	5.0	76.67	0.845	45510 3	0.0	0.0	0.0	0.0

	iso_code	continent	location	date	total_cases	new_cases	new_cases_smoothed	total_deaths	new_deaths	new_deaths_smoothed	...	male_smokers	handwashing_facilities	hospital_beeds_per_thousand	life_expectancy	human_development_index	population	excess_mortality_cumulative_absolute	excess_mortality_cumulative	excess_mortality	excess_mortality_cumulative_per_million
		ica	ina	-02													24				
2	ARG	South America	Argentina	2020-01-03	0	0	0.0	0	0	0.0	...	27.7	0.0	5.0	76.67	0.845	45510324	0.0	0.0	0.0	0.0

3 rows × 67 columns

```
sa_data.isnull().sum().sum()
0
sa_cases = sa_data['total_cases'].sum()
sa_deaths = sa_data['total_deaths'].sum()
sa_vaccinated = sa_data['people_fully_vaccinated'].sum()


labels = ['Cases','Deceased','Vaccinated']
sizes = [sa_cases,sa_deaths,sa_vaccinated]
color= ['teal','#66b3ff','red']
explode = []

for i in labels:
    explode.append(0.05)

plt.figure(figsize= (15,10))
plt.pie(sizes, labels=labels, autopct='%1.1f%%', startangle=9, explode =explode,colors =
color)
centre_circle = plt.Circle((0,0),0.70,fc='white')

fig = plt.gcf()
fig.gca().add_artist(centre_circle)
plt.subplots_adjust(top=0.85)
plt.legend(labels, loc='upper right', bbox_to_anchor=(1.2, 1))
plt.title('South America COVID-19 Cases',fontsize = 30)
plt.axis('equal')
plt.tight_layout()
```



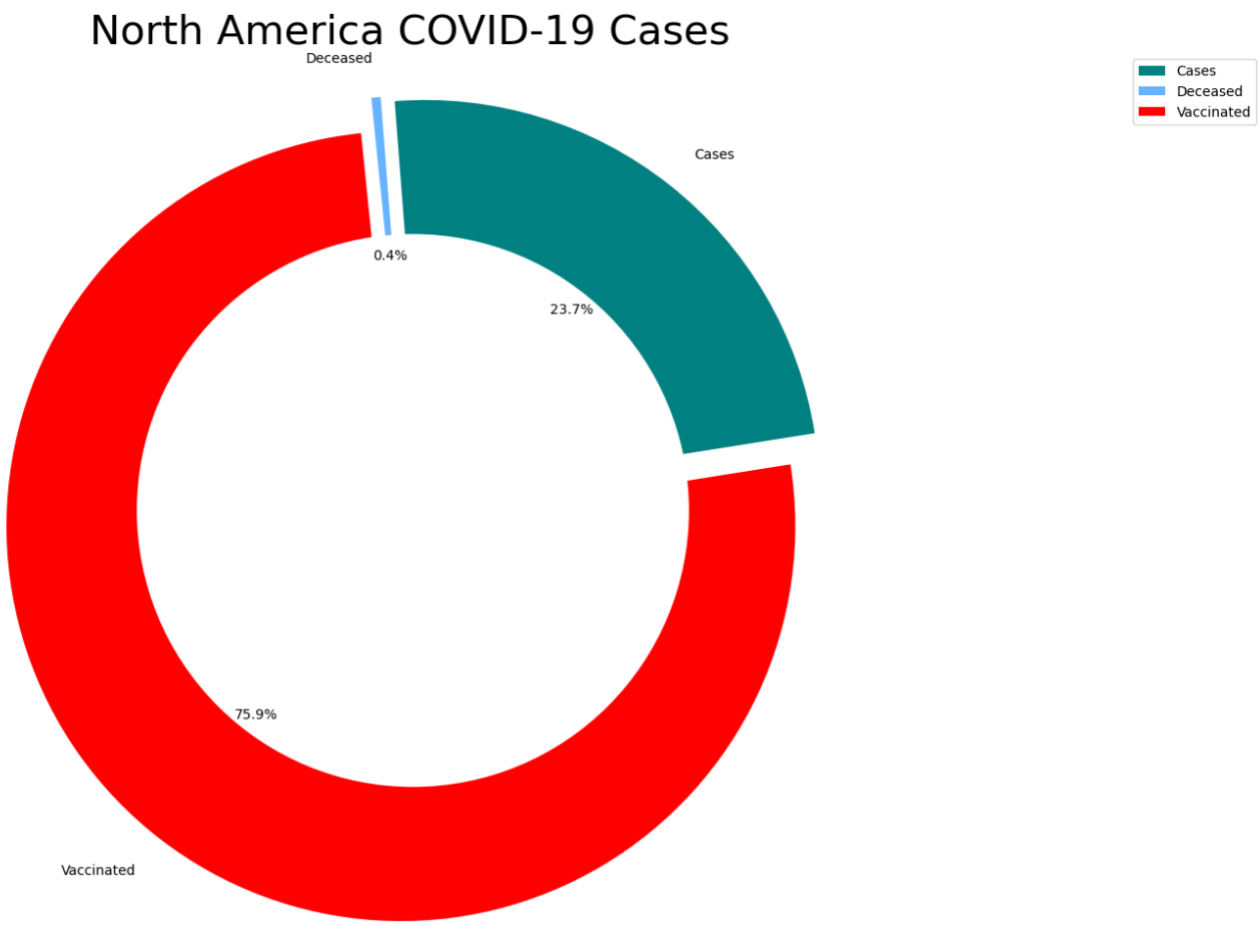

```
na_data = pd.read_csv('North_America_covid_dataset.csv')
na_data.head(3)
```

	iso_code	continent	location	date	total_cases	new_cases	new_cases_smoothed	total_deaths	new_deaths	new_deaths_smoothed	...	male_smokers	handwashing_facilities	hospital_beds_per_thousand	life_expectancy	human_development_index	population	excess_mortality_cumulative_absolute	excess_mortality_cumulative	excess_mortality	excess_mortality_cumulative_per_million
0	AIA	North America	Anguilla	2020-01-03	0	0	0.0	0	0	0.0	...	0.0	0.0	0.0	81.88	0.0	15877	0.0	0.0	0.0	0.0
1	AIA	North Amer	Anguil	2020-01	0	0	0.0	0	0	0.0	...	0.0	0.0	0.0	81.88	0.0	15877	0.0	0.0	0.0	0.0

	iso_code	continent	location	date	total_cases	new_cases	new_cases_smoothed	total_deaths	new_deaths	new_deaths_smoothed	...	male_smokers	handwashing_facilities	hospital_beds_per_thousand	life_expectancy	human_development_index	population	excess_mortality_cumulative_absolute	excess_mortality_cumulative	excess_mortality	excess_mortality_cumulative_per_million
		ica	la	-04																	
2	AIA	North America	Anguilla	2020-01-05	0	0	0.0	0	0	0.0	...	0.0	0.0	0.0	81.88	0.0	15877	0.0	0.0	0.0	0.0

3 rows × 67 columns

```
na_data.isnull().sum().sum()
0
na_cases = na_data['total_cases'].sum()
na_deaths = na_data['total_deaths'].sum()
na_vaccinated = na_data['people_fully_vaccinated'].sum()


labels = ['Cases','Deceased','Vaccinated']
sizes = [na_cases,na_deaths,na_vaccinated]
color= ['teal','#66b3ff','red']
explode = []

for i in labels:
    explode.append(0.05)

plt.figure(figsize= (15,10))
plt.pie(sizes, labels=labels, autopct='%1.1f%%', startangle=9, explode =explode,colors =
color)
centre_circle = plt.Circle((0,0),0.70,fc='white')

fig = plt.gcf()
fig.gca().add_artist(centre_circle)
plt.subplots_adjust(top=0.85)
plt.legend(labels, loc='upper right', bbox_to_anchor=(1.2, 1))
plt.title('North America COVID-19 Cases',fontsize = 30)
plt.axis('equal')
plt.tight_layout()
```




```
oceania_data = pd.read_csv('Oceania_covid_dataset.csv')
oceania_data.isnull().sum().sum()
0
oceania_data.head(3)
```

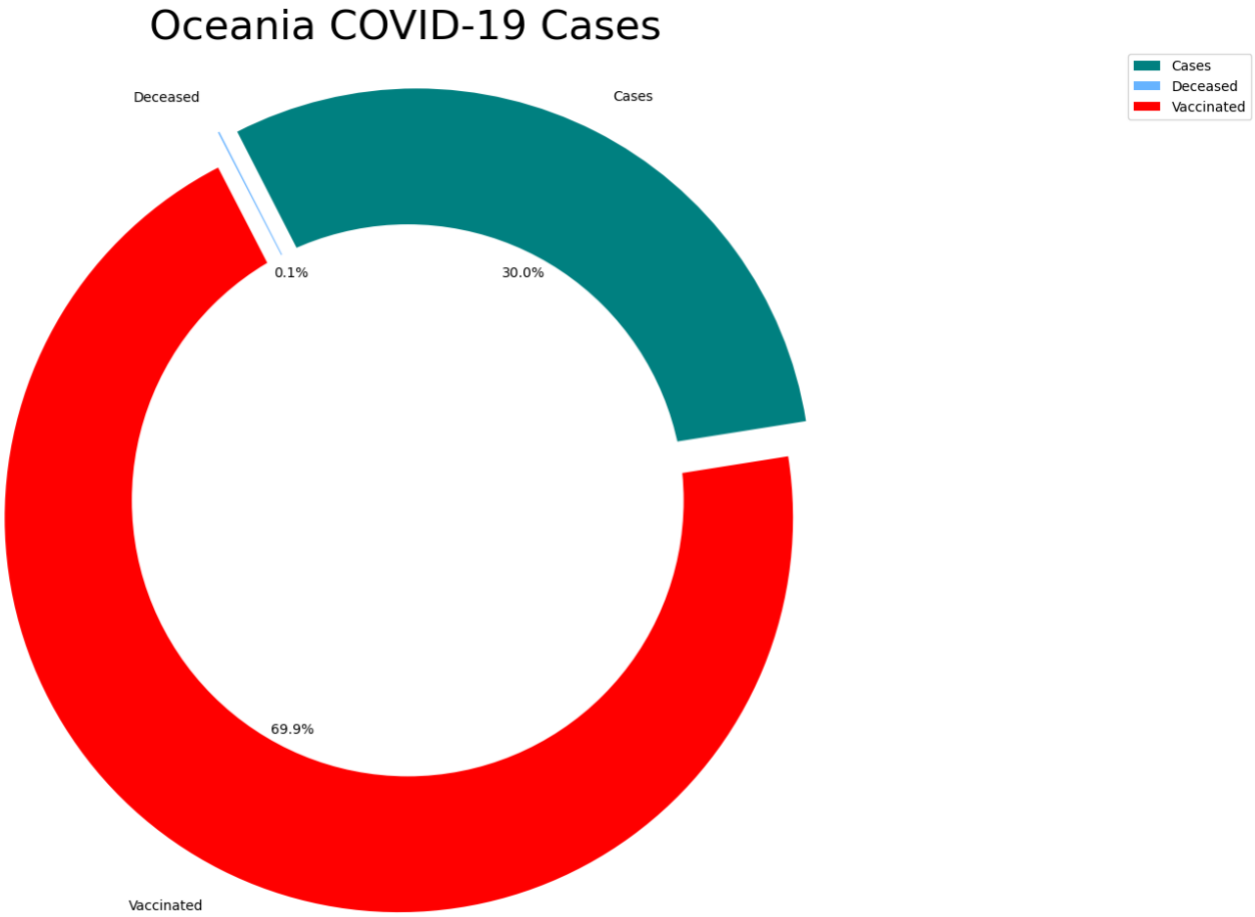
	iso_code	continent	location	date	total_cases	new_cases	new_cases_smoothed	total_deaths	new_deaths	new_deaths_smoothed	...	male_smokers	handwashing_facilities	hospital_beds_per_thousand	life_expectancy	human_development_index	population	excess_mortality_cumulative_absolute	excess_mortality_cumulative	excess_mortality	excess_mortality_cumulative_per_million
0	ASM	Oceania	American Samoa	2020-01-03	0	0	0.0	0	0	0.0	...	0.0	0.0	0.0	73.74	0.0	44295	0.0	0.0	0.0	0.0

	iso_code	continent	location	date	total_cases	new_cases	new_cases_smoothed	total_deaths	new_deaths	new_deaths_smoothed	...	male_smokers	handwashing_facilities	hospital_beds_per_thousand	life_expectancy	human_development_index	population	excess_mortality_cumulative_absolute	excess_mortality_cumulative	excess_mortality	excess_mortality_cumulative_per_million
1	ASM	Oceania	American Samoa	2020-01-04	0	0	0.0	0	0	0.0	...	0.0	0.0	0.0	73.74	0.0	44295	0.0	0.0	0.0	0.0
2	ASM	Oceania	American Samoa	2020-01-05	0	0	0.0	0	0	0.0	...	0.0	0.0	0.0	73.74	0.0	44295	0.0	0.0	0.0	0.0

3 rows × 67 columns

```
oceania_cases = oceania_data['total_cases'].sum()
oceania_deaths = oceania_data['total_deaths'].sum()
oceania_vaccinated = oceania_data['people_fully_vaccinated'].sum()


labels = ['Cases','Deceased','Vaccinated']
sizes = [oceania_cases,oceania_deaths,oceania_vaccinated]
color= ['teal','#66b3ff','red']
explode = []

for i in labels:
    explode.append(0.05)

plt.figure(figsize= (15,10))
plt.pie(sizes, labels=labels, autopct='%1.1f%%', startangle=9, explode =explode,colors =
color)
centre_circle = plt.Circle((0,0),0.70,fc='white')

fig = plt.gcf()
fig.gca().add_artist(centre_circle)
plt.subplots_adjust(top=0.85)
plt.legend(labels, loc='upper right', bbox_to_anchor=(1.2, 1))
plt.title('Oceania COVID-19 Cases',fontsize = 30)
plt.axis('equal')
```

```
plt.tight_layout()
```



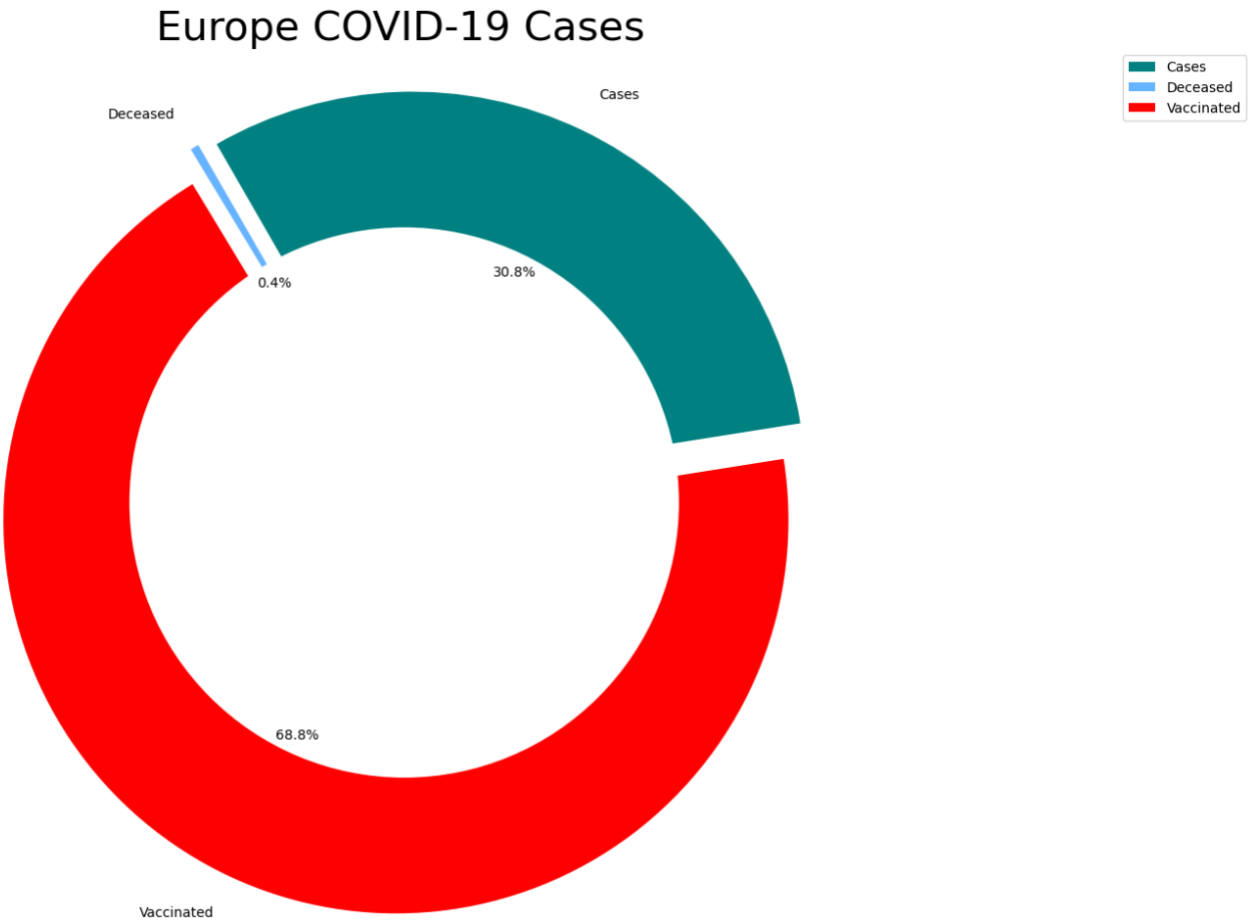

```
europe_data = pd.read_csv('Europe_covid_dataset.csv')
europe_data.head(3)
```

	iso_code	continent	location	date	total_cases	new_cases	new_cases_smoothed	total_deaths	new_deaths	new_deaths_smoothed	...	male_smokers	handwashing_facilities	hospital_beds_per_thousand	life_expectancy	human_development_index	population	excess_mortality_cumulative_absolute	excess_mortality_cumulative	excess_mortality	excess_mortality_cumulative_per_million
0	ALB	Europe	Albania	2020-01-03	0	0	0.0	0	0	0.0	...	51.2	0.0	2.89	78.57	0.795	2842318	0.0	0.0	0.0	0.0
1	ALB	Euro	Alban	2020-0	0	0	0.0	0	0	0.0	...	51.2	0.0	2.89	78.57	0.795	28423	0.0	0.0	0.0	0.0

	iso_code	continent	location	date	total_cases	new_cases	new_cases_smoothed	total_deaths	new_deaths	new_deaths_smoothed	...	male_smokers	handwashing_facilities	hospital_beds_per_thousand	life_expectancy	human_development_index	population	excess_mortality_cumulative_absolute	excess_mortality_cumulative	excess_mortality	excess_mortality_cumulative_per_million
		pe	ia	1-04													18				
2	ALB	Europe	Albania	2020-01-05	0	0	0.0	0	0	0.0	...	51.2	0.0	2.89	78.57	0.795	2842318	0.0	0.0	0.0	0.0

3 rows × 67 columns

```
europe_data.isnull().sum().sum()
0
europe_cases = europe_data['total_cases'].sum()
europe_deaths = europe_data['total_deaths'].sum()
europe_vaccinated = europe_data['people_fully_vaccinated'].sum()


labels = ['Cases','Deceased','Vaccinated']
sizes = [europe_cases,europe_deaths,europe_vaccinated]
color= ['teal','#66b3ff','red']
explode = []

for i in labels:
    explode.append(0.05)

plt.figure(figsize= (15,10))
plt.pie(sizes, labels=labels, autopct='%1.1f%%', startangle=9, explode =explode,colors =
color)
centre_circle = plt.Circle((0,0),0.70,fc='white')

fig = plt.gcf()
fig.gca().add_artist(centre_circle)
plt.subplots_adjust(top=0.85)
plt.legend(labels, loc='upper right', bbox_to_anchor=(1.2, 1))
plt.title('Europe COVID-19 Cases',fontsize = 30)
plt.axis('equal')
plt.tight_layout()
```




```
data= pd.read_csv('final_covid_dataset.csv')
data.head(3)
```

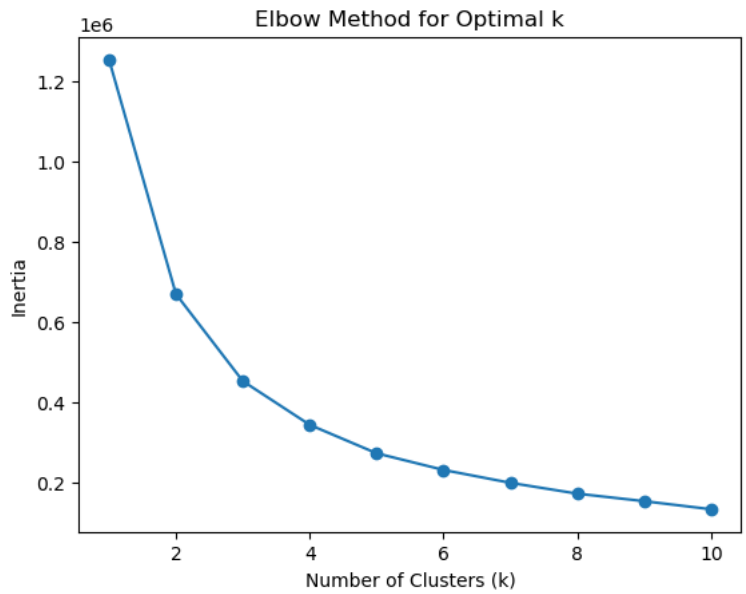
	iso_code	continent	location	date	total_cases	new_cases	new_cases_smoothed	total_deaths	new_deaths	new_deaths_smoothed	...	male_smokers	handwashing_facilities	hospital_beds_per_thousand	life_expectancy	human_development_index	population	excess_mortality_cumulative_absolute	excess_mortality_cumulative	excess_mortality	excess_mortality_cumulative_per_million
0	AFG	Asia	Afghanistan	2020-01-03	0	0	0.0	0	0	0.0	...	0.0	37.746	0.5	64.83	0.511	41128772	0.0	0.0	0.0	0.0
1	AFG	Asia	Afghanistan	2020-01	0	0	0.0	0	0	0.0	...	0.0	37.746	0.5	64.83	0.511	41128	0.0	0.0	0.0	0.0

	iso_code	continent	location	date	total_cases	new_cases	new_cases_smoothed	total_deaths	new_deaths	new_deaths_smoothed	...	male_smokers	handwashing_facilities	hospital_beds_per_thousand	life_expectancy	human_development_index	population	excess_mortality_cumulative_absolute	excess_mortality_cumulative	excess_mortality	excess_mortality_cumulative_per_million
				-04													72				
2	AFG	Asia	Afghanistan	2020-01-05	0	0	0.0	0	0	0.0	...	0.0	37.746	0.5	64.83	0.511	41128772	0.0	0.0	0.0	0.0

3 rows × 67 columns

```
data.isnull().sum().sum()
0
categorical_columns = data.select_dtypes(include=['object']).columns.tolist()

print("Columns with Categorical Values:", categorical_columns)
Columns with Categorical Values: ['iso_code', 'continent', 'location', 'date', 'tests_units'
]
numerical_columns = data.select_dtypes(include=['number']).columns.tolist()

print("Columns with Numerical Values:", numerical_columns)
Columns with Numerical Values: ['total_cases', 'new_cases', 'new_cases_smoothed', 'total_dea
ths', 'new_deaths', 'new_deaths_smoothed', 'total_cases_per_million', 'new_cases_per_million
', 'new_cases_smoothed_per_million', 'total_deaths_per_million', 'new_deaths_per_million', '
new_deaths_smoothed_per_million', 'reproduction_rate', 'icu_patients', 'icu_patients_per_mil
lion', 'hosp_patients', 'hosp_patients_per_million', 'weekly_icu_admissions', 'weekly_icu_ad
missions_per_million', 'weekly_hosp_admissions', 'weekly_hosp_admissions_per_million', 'tota
l_tests', 'new_tests', 'total_tests_per_thousand', 'new_tests_per_thousand', 'new_tests_smoo
thed', 'new_tests_smoothed_per_thousand', 'positive_rate', 'tests_per_case', 'total_vaccinat
ions', 'people_vaccinated', 'people_fully_vaccinated', 'total_boosters', 'new_vaccinations',
'new_vaccinations_smoothed', 'total_vaccinations_per_hundred', 'people_vaccinated_per_hundre
d', 'people_fully_vaccinated_per_hundred', 'total_boosters_per_hundred', 'new_vaccinations_s
moothed_per_million', 'new_people_vaccinated_smoothed', 'new_people_vaccinated_smoothed_per_
hundred', 'stringency_index', 'population_density', 'median_age', 'aged_65_older', 'aged_70_
older', 'gdp_per_capita', 'extreme_poverty', 'cardiovasc_death_rate', 'diabetes_prevalence',
'female_smokers', 'male_smokers', 'handwashing_facilities', 'hospital_beds_per_thousand', 'l
ife_expectancy', 'human_development_index', 'population', 'excess_mortality_cumulative_absol
ute', 'excess_mortality_cumulative', 'excess_mortality', 'excess_mortality_cumulative_per_mi
llion']
binary_columns = []

for column in numerical_columns:
    unique_values = data[column].unique()
```

```
    if len(unique_values) == 2 and set(unique_values) <= {0, 1}:
        binary_columns.append(column)

print("Binary Columns:", binary_columns)
Binary Columns: []
from sklearn.cluster import KMeans
import matplotlib.pyplot as plt
features = data[['total_cases', 'new_cases', 'total_deaths', 'population']]
# Standardize the data (optional but recommended for k-means)
from sklearn.preprocessing import StandardScaler
scaler = StandardScaler()
features_scaled = scaler.fit_transform(features)
# Apply k-means clustering for different values of k
inertia = []
for k in range(1, 11):
    kmeans = KMeans(n_clusters=k, random_state=42)
    kmeans.fit(features_scaled)
    inertia.append(kmeans.inertia_)
# Plot the elbow curve
plt.plot(range(1, 11), inertia, marker='o')
plt.title('Elbow Method for Optimal k')
plt.xlabel('Number of Clusters (k)')
plt.ylabel('Inertia')
plt.show()
```



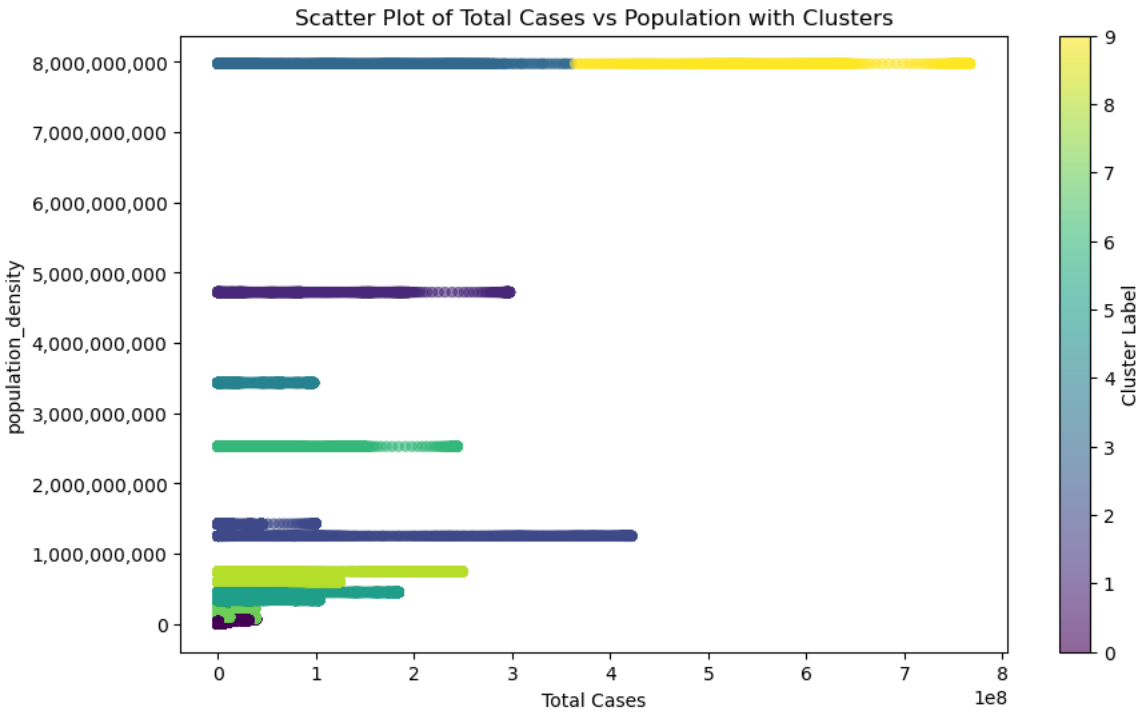

```
#from the elbow graph abive

k = 3
#creating a kmeans model

kmeans = KMeans(n_clusters = k)

kmeans.fit(features)
```

```
KMeans
```

```
KMeans(n_clusters=10)
```

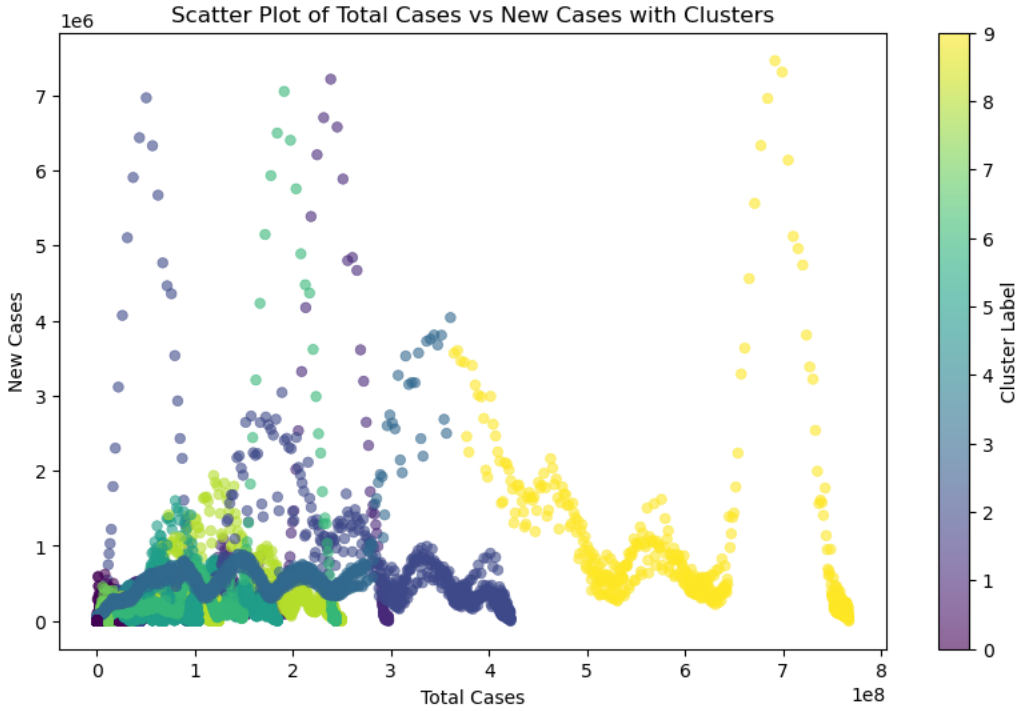
```
data['cluster_label'] = kmeans.labels_
from matplotlib.ticker import FuncFormatter

# Assuming 'data' is your DataFrame with 'population' not in scientific notation
plt.figure(figsize=(10, 6))
```

```
# Scatter plot with color-coded clusters
scatter = plt.scatter(data['total_cases'], data['population'], c=data['cluster_label'],
cmap='viridis',alpha=0.6, s=30, linewidths=0.5)

# Add labels and title
plt.xlabel('Total Cases')
plt.ylabel('population_density')
plt.title('Scatter Plot of Total Cases vs Population with Clusters')

# Add colorbar
colorbar = plt.colorbar(scatter)
colorbar.set_label('Cluster Label')

# Format the population axis tick labels
plt.gca().yaxis.set_major_formatter(FuncFormatter(lambda x, _: '{:,.0f}'.format(x)))

# Show the plot
plt.show()
```

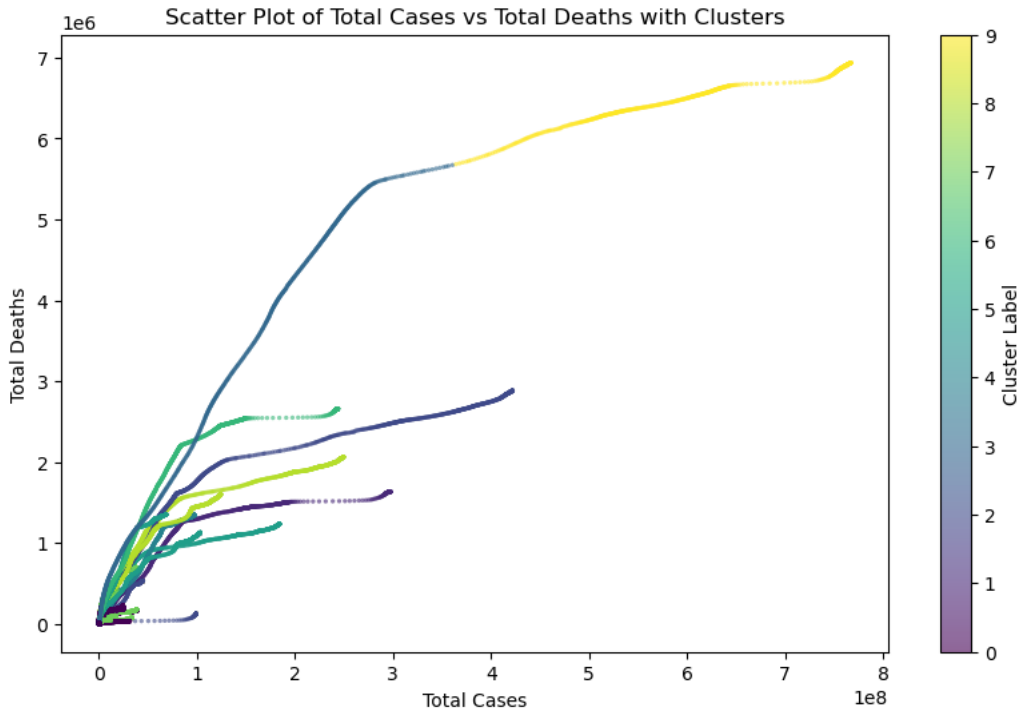



```
plt.figure(figsize=(10, 6))

# Scatter plot with color-coded clusters
scatter = plt.scatter(data['total_cases'], data['new_cases'], c=data['cluster_label'],
cmap='viridis',alpha=0.6, s=30, linewidths=0.5)

# Add labels and title
plt.xlabel('Total Cases')
plt.ylabel('New Cases')
plt.title('Scatter Plot of Total Cases vs New Cases with Clusters')

# Add colorbar
colorbar = plt.colorbar(scatter)
colorbar.set_label('Cluster Label')

# Show the plot
```

```
plt.show()
```




```
plt.figure(figsize=(10, 6))

# Scatter plot with color-coded clusters
scatter = plt.scatter(data['total_cases'], data['total_deaths'], c=data['cluster_label'],
cmap='viridis',alpha=0.6, s=5, linewidths=0.1)

# Add labels and title
plt.xlabel('Total Cases')
plt.ylabel('Total Deaths')
plt.title('Scatter Plot of Total Cases vs Total Deaths with Clusters')

# Add colorbar
colorbar = plt.colorbar(scatter)
colorbar.set_label('Cluster Label')

# Show the plot
plt.show()
```



Applying chi-squared test

```
from scipy.stats import chi2_contingency
contingency_table = pd.crosstab(data['location'], data['tests_units'])

print(contingency_table)
tests_units        0  people tested  samples tested  tests performed  \
location
Afghanistan     1092              0               0              146
Africa          1238              0               0                0
Albania          389              0               0              849
Algeria         1237              0               0                1
American Samoa  1238              0               0                0
...              ...            ...             ...              ...
Western Sahara     0              0               0                1
World           1243              0               0                0
Yemen           1150              0               0               88
Zambia           419              0               0              819
Zimbabwe         460              0               0              778

tests_units     units unclear
location
Afghanistan                 0
Africa                      0
Albania                     0
Algeria                     0
American Samoa              0
...                       ...
Western Sahara              0
World                       0
Yemen                       0
Zambia                      0
Zimbabwe                    0
```

```
[255 rows x 5 columns]
stat, p, dof, expected = chi2_contingency(contingency_table)

print("Chi-square Statistic:", stat)
print("P-value:", p)
print("Degrees of Freedom:", dof)
print("Expected Frequencies Table:")
print(expected)
Chi-square Statistic: 728090.7786153554
P-value: 0.0
Degrees of Freedom: 1016
Expected Frequencies Table:
[[815.98445416  64.24604571  37.90267727 316.54327459   3.32354828]
 [815.98445416  64.24604571  37.90267727 316.54327459   3.32354828]
 [815.98445416  64.24604571  37.90267727 316.54327459   3.32354828]
 ...
 [815.98445416  64.24604571  37.90267727 316.54327459   3.32354828]
 [815.98445416  64.24604571  37.90267727 316.54327459   3.32354828]
 [815.98445416  64.24604571  37.90267727 316.54327459   3.32354828]]
alpha = 0.05
print("p value is " + str(p))
if p <= alpha:
    print('Dependent (reject H0)')
else:
    print('Independent (H0 holds true)')
p value is 0.0
Dependent (reject H0)
contingency_table_1 = pd.crosstab(data['date'], data['tests_units'])
print(contingency_table_1)


stat, p, dof, expected = chi2_contingency(contingency_table_1)

alpha = 0.05
print("p value is " + str(p))
if p <= alpha:
    print('Dependent (reject H0)')
else:
    print('Independent (H0 holds true)')
tests_units    0  people tested  samples tested  tests performed  \
date
2020-01-01     0              1               0                1
2020-01-02     0              1               0                1
2020-01-03   244              1               0                1
2020-01-04   243              1               0                2
2020-01-05   243              1               0                2
...          ...            ...             ...              ...
2023-05-25    38              0               0                0
2023-05-26    35              0               0                0
2023-05-27    32              0               0                0
2023-05-28    30              0               0                0
2023-05-29    23              0               0                0

tests_units  units unclear
date
2020-01-01               0
2020-01-02               0
```

```
2020-01-03               0
2020-01-04               0
2020-01-05               0
...                    ...
2023-05-25               0
2023-05-26               0
2023-05-27               0
2023-05-28               0
2023-05-29               0

[1245 rows x 5 columns]
p value is 0.0
Dependent (reject H0)
contingency_table_2 = pd.crosstab(data['continent'], data['tests_units'])
print(contingency_table_2)


stat, p, dof, expected = chi2_contingency(contingency_table_2)

alpha = 0.05
print("p value is " + str(p))
if p <= alpha:
    print('Dependent (reject H0)')
else:
    print('Independent (H0 holds true)')
tests_units         0  people tested  samples tested  tests performed  \
continent
0                6210              0               0                0
Africa          50428           2162            1643            17572
Asia            35972           5791            5425            15320
Europe          36399           3343             842            28519
European Union   1243              0               0                0
North America   39233           3366            1681             7731
Oceania         26501              9               0             4441
South America   10493           1586               0             6516

tests_units     units unclear
continent
0                           0
Africa                      0
Asia                      841
Europe                      0
European Union              0
North America               0
Oceania                     0
South America               0
p value is 0.0
Dependent (reject H0)
contingency_table_3 = pd.crosstab(data['weekly_hosp_admissions'],
data['people_fully_vaccinated'])

stat, p, dof, expected = chi2_contingency(contingency_table_3)

alpha = 0.05
print("p value is " + str(p))
if p <= alpha:
    print('Dependent (reject H0)')
else:
```

```
    print('Independent (H0 holds true)')
p value is 0.0
Dependent (reject H0)
contingency_table_4 = pd.crosstab(data['continent'], data['total_deaths'].max())
print(contingency_table_4)

stat, p, dof, expected = chi2_contingency(contingency_table_4)

alpha = 0.05
print("p value is " + str(p))
if p <= alpha:
    print('Dependent (reject H0)')
else:
    print('Independent (H0 holds true)')
col_0           6935876
continent
0                  6210
Africa            71805
Asia              63349
Europe            69103
European Union     1243
North America     52011
Oceania           30951
South America     18595
p value is 1.0
Independent (H0 holds true)
```

ANOVA testing

```
import statsmodels.api as sm
from statsmodels.formula.api import ols
# Assuming 'data' is your DataFrame containing the specified columns
formula = 'total_cases ~ new_cases+ total_deaths+ reproduction_rate + icu_patients+
total_tests + life_expectancy'

# Fit the ANOVA model
model = ols(formula, data=data).fit()

# Print ANOVA table
anova_table = sm.stats.anova_lm(model, typ=2)
print(anova_table)
                        sum_sq        df             F         PR(>F)
new_cases          7.284521e+16       1.0  4.942548e+02  2.055598e-109
total_deaths       2.248246e+20       1.0  1.525435e+06   0.000000e+00
reproduction_rate  7.027783e+16       1.0  4.768351e+02  1.251642e-105
icu_patients       2.161156e+16       1.0  1.466345e+02   9.598173e-34
total_tests        8.116292e+15       1.0  5.506905e+01   1.166611e-13
life_expectancy    1.717780e+17       1.0  1.165514e+03  5.631224e-255
Residual           4.616949e+19  313260.0           NaN            NaN
import scipy.stats as stats
data['tests_units'] = pd.DataFrame(data['tests_units'])
from scipy.stats import f_oneway
stat.f_oneway(data['tests_units'])
```