UNSUPERVISED LEARNING ANALYSIS OF CAUSES OF DEATH AMONG CHILDREN

Group Color: Gold
Members:
Tileshwar Narayan
Tejaswi Neelapu
Nidhi Trivedi

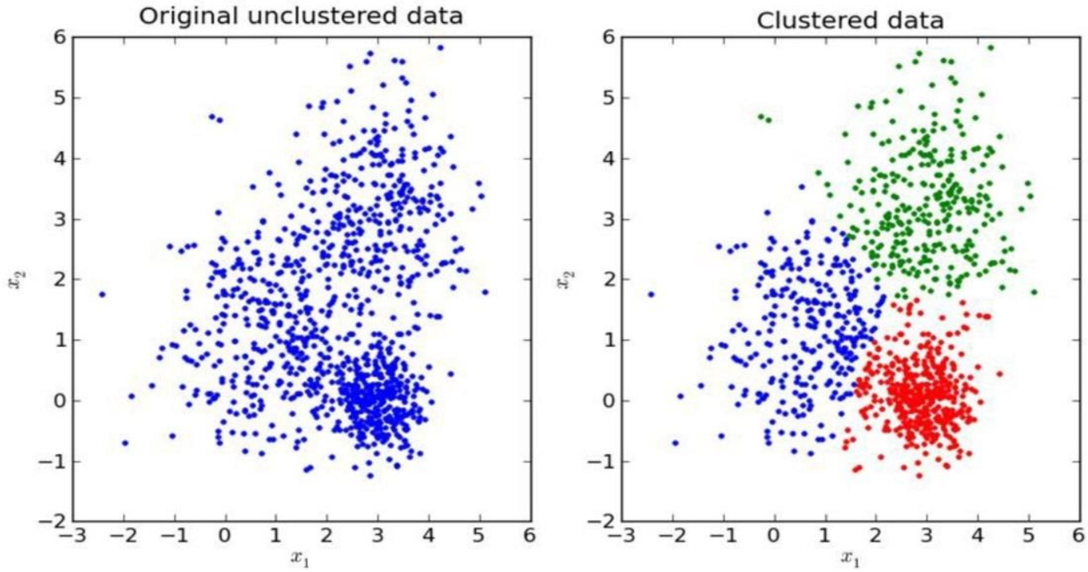
Abstract:
This report takes us through the application of unsupervised learning techniques for analyzing the causes of death in children under five years old, using data “Causes of death in children under five, 2019” from “Our World in Data” [1], a scientific online resource that focuses on large global problems. The data gives the count of deaths because of various diseases/causes in multiple countries for every year starting from 1990 to 2019. Techniques like Principal Component Analysis (PCA) and Singular Value Decomposition (SVD) are implemented for dimensionality reduction. Clustering techniques like K-Means and Hierarchical clustering were implemented for grouping untagged data. [2] The dimensionality reduction and clustering of the child mortality data around the world from 1990 to 2019 identified seven significant components and showed optimal clustering patterns, with imbalance between the countries such as India, Nigeria, and China. Main risk factors which have contributed to this trend were respiratory illnesses, preterm births, and stillbirths. The information presented gives an overview through which evidence, based pediatrics improvement strategies necessary for decreasing child mortality may be undertaken globally.

Introduction and Overview:
Children's lives, especially those under five years old, are still at risk across the world. Even though the healthcare sector has improved over the decades, decreasing child mortality remains a top concern to the global health agencies. [3] It’s important to know the factors associated with child mortality and its trends in order to design effective interventions and policies aimed at reducing mortality rates. Therefore, a discussion of causes and trends in child mortality across various countries in different years can be useful for understanding past and ongoing health issues, for informing the work of health care practitioners and policy makers. The dataset used in this study was taken from “Our World in Data” and includes data on deaths of children under the age of five years by diseases/causes across multiple countries from 1990 to 2019. This dataset provides a global picture of the child mortality rates with respect to the geographical location. The examination of such data can help identify the most important and distinctive trends in causes of child mortality across the globe.

Due to their ability to identify undetected patterns in the data without a clear categorization of the data set in advance, unsupervised learning techniques are especially useful in this context. These techniques will be useful in identifying other clusters and important features that are not very noticeable at first glance. Primarily, PCA and SVD are used for dimensionality reduction to focus on the most significant factors in the given dataset. However, clustering techniques such as K-Means and hierarchical clustering are applied to aggregate countries that share similar mortality ranks making it easier for comparison across the world. The main aim of this study is to use effective unsupervised learning methods to analyze a large and comprehensive dataset which is the causes of death in children under five years old. Mainly looking to identify achievable and strong patterns and associations, which could in turn help guide potential future health incorporation. It not only brings huge contributions to the discussion of child mortality in a report manner but also helps build up data-based solutions for this crucial area.

Theoretical Background:

Unsupervised learning is a category of machine learning algorithms that analyzes the data in a set that is not labeled and categorizes it into subsets, based on the similarities and differences of the data in a set. This type of learning does not require the participation of a specialist, unsupervised algorithms identify hidden structures in the information. Unsupervised learning is further defined as the process of using raw data in organizing new features or groups of patterns that are like each other. For example, to predict churn rate, the unlabeled data is given to the model for prediction by the author. No information is provided regarding customers. The model will analyze the data and find hidden patterns to categorize into two clusters of Churned customers and Non churned customers. [4]



[5]

Types of unsupervised learning:

Dimensionality Reduction: This is done to reduce the original high dimensional data into a lower form without leaving out any important features as much as possible. This technique is beneficial for enhancing the score of machine learning algorithms and for the data depiction method. There are several techniques that come under dimensionality reduction. [6]

Principal Component Analysis (PCA): PCA helps in decreasing the size of the data by reducing the dimensions. The process of PCA helps in coming up with new variables whose values are uncorrelated and whose variances add up to the variances of all the variables in the original set. It refines a lot of fine features in the data while keeping most of the data variability of the original data. [7]

To perform PCA, start with standardization, with a mean of zero and a standard deviation of one. Then a covariance matrix is done to get the relations between the variables. This matrix keeps the standard deviations of the features and correlations between them, which is used to find the eigenvectors of highest variability in the data. Then, eigenvalues and eigenvectors analyses are performed on the covariance matrix. The principal components are ranked based on variance, and commonly, the initial ones are important in capturing the most variance, in relation to the subsequent ones. Top principal components are then utilized to transform the data set from the original high dimensionality into the low dimensionality. This helps in keeping the basic pattern of the base data and is helpful for visualizations and other computations. Reduced dimensionality of the data can give various patterns, groups, and changes which are otherwise not easily understood in a high dimensional space.

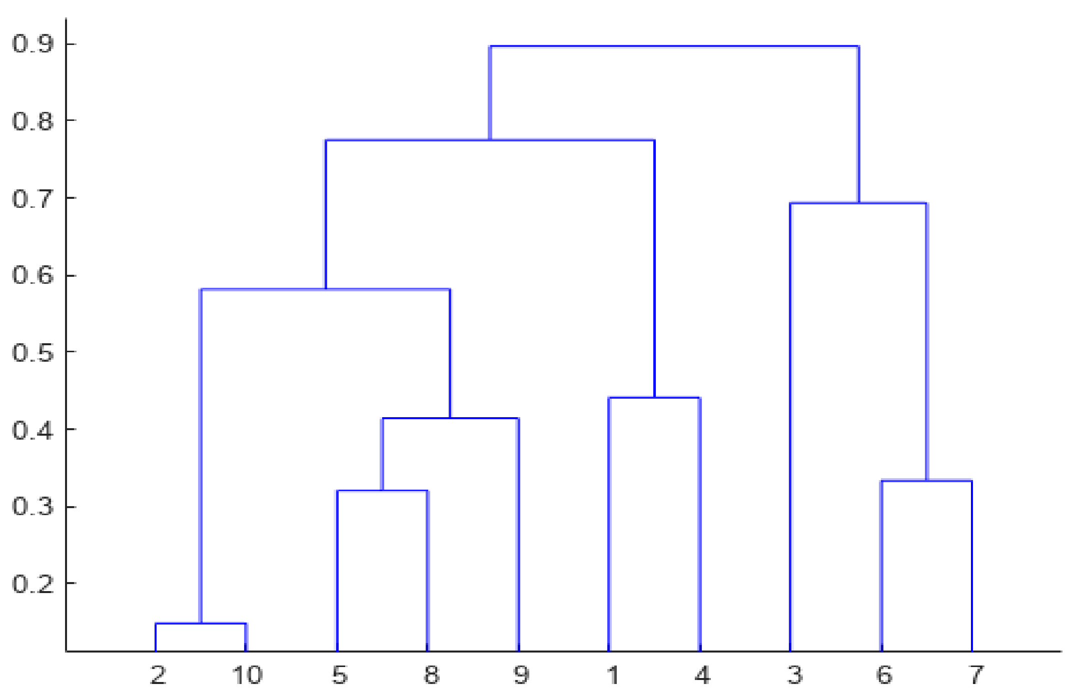
Singular Value Decomposition (SVD): SVD can be defined specifically as a matrix approximation procedure, which can break a given set into three more matrices to reveal some important structure in the data. It is useful specifically in data pre-processing and the most used in scenarios where it is required to decrease the number of features in data or filter out the noise from data.

To perform SVD process starts by breaking up a given matrix A into three smaller matrices: U, Σ, and V^T. Here, U is an orthogonal matrix whose columns are called the left singular vectors, Σ is a diagonal matrix containing the singular values arranged in descending order, and V^T is the transpose of an orthogonal matrix V (columns are called the right singular vectors). The diagonal elements of Σ are helpful for characterizing the basic dimension of the data, thus the dimensionality reductions can be easily done with the selection of the top k singular values and their corresponding singular vectors. This form of SVD divides the data into an original data matrix and a new matrix containing only significant eigenvectors, which works for tasks such as removing unwanted columns or compressing the data. The matrices U and V give a picture of the relationships within the data. The singular vectors on the left are the original data observations, the singular vectors on the right are the features, this gives proper analysis of the structure of the data.

Clustering: This is done to divide the original high dimensional data into similar sets, called clusters, where data in the same cluster have better similarities than the data in other clusters. Clustering algorithms is the pattern recognizer that helps in building structures without involving a training set making it an unsupervised learning. Some of the most used clustering techniques are K-Means and Hierarchical, which are effective when applied in different fields like customer grouping, identifying outliers or images categorization, and others.

K-means Clustering: This method is used to divide a dataset into subsets, by grouping patterns or features that have similarities to give a better understanding of data. The process of clustering starts by determining the numbers of desired clusters. It then proceeds by randomly assigning centroids to these clusters, the data points are then clustered into the nearest centroids and assigned the value of that cluster forming initial clusters. Next, centroids are recalculated from averages of all points within the cluster different from the centroid of the earlier iteration. This process of assigning different data points and recalculating centroid values continues until there are no significant changes being made in the position of centroids, and thus joining. This produces a series of clusters that have the least variation, so that all objects in the same cluster are more similar to each other than to objects of any other clusters. One of the most common uses of the K-Means algorithm is due to its basic and fast approach to cluster data into meaningful sets.

Hierarchical Clustering: This method is used to order the clusters in such a way to ensure that, when the data undergo clustering, it can provide insights into the hierarchical nature of the data and the number of clusters combined. The process starts by implementing every data point as a Cluster by itself. It then gradually combines the two clusters closest to each other using a selected distance from the Euclidean distance family. It continues until all the data points are grouped together into just one cluster which forms a tree-like diagram known as dendrogram. Another approach is opposite to the previous one, it can be described as a divide and conquer approach when all objects are initially assigned to a single cluster and then divided into more clusters. The format of viewing the data is shown in a dendrogram (tree diagram) and the number of clusters could then be determined based on the level of adaptability where the tree is sliced. Hierarchical Clustering can present considerable information regarding the relationships between the observations and does not need a numerical specification of the number of clusters.



[8]

Challenges of unsupervised learning:

- ➔ Overfitting or Underfitting
- ➔ Data Preprocessing
- ➔ Scalability
- ➔ Interpretability

Applications of unsupervised learning:

- ➔ Image and video analysis
- ➔ Customer division
- ➔ Social media analysis
- ➔ Recommendation systems
- ➔ Gene analysis

Methodology:

The primary goal of this project is to analyze and identify patterns in the causes of death among children under 5 years old across various countries using unsupervised learning techniques. The dataset utilized for this study is sourced from global health databases, detailing the number of deaths in children under 5 due to various causes for different countries. The dataset includes 32 columns, each representing cause of death, along with country names, codes and the year and total number of rows are 6841.

Data Preparation:

The process starts by loading the data from a csv file. To ensure a clean and usable dataset for analysis. Initially, a statistical summary of the numerical columns was generated to understand the basic characteristics of the data. The dataset was then checked for the presence of any null values, confirming that there were no missing entries. To maintain data integrity, any rows with null values were removed, ensuring that the analysis would be performed on a complete dataset without any missing values. The rows corresponding to the 'World' entity were removed from the Dataframe. These rows contained cumulative data across all countries, which would lead to duplication of data in our analysis. Since our analysis focuses on country-wise data, we dropped the 'World' rows to avoid duplicating global totals. This step is important to maintain the integrity and accuracy of our country-wise analysis. The column names in the original Dataframe were quite long and descriptive. To make the code clearer and easier to work with, a dictionary was created that mapped the original column names to shorter. The data is for the years from 1990 to 2019. The filtered data was grouped by the country name column and aggregation was done to sum the data across all years for each entity. This gave a new dataframe that contained the total number of deaths for each cause and each entity across the specified years. This aggregated data enables us to easily visualize and compare the total deaths across different countries and causes without having to deal with individual year-wise data points.

Model Fitting:

Principal Component Analysis (PCA):

While fitting the model, the dataset was first prepared for Principal Component Analysis (PCA). Firstly, the data is standardized by centering it around the mean and scaling it to have a standard deviation of 1. Secondly, the preprocessing was done because PCA is sensitive to the scale of the variables. This standardization was applied to the dataset, excluding the country name (‘entity’), country code ('Code'), and 'Year' columns as they were non-numerical.

After standardizing the data, PCA was performed to calculate the principal components. The number of principal components was found to be equal to the number of columns in the data, as PCA can create principal components as high as the number of columns. To show the principal components and understand the contribution of each original variable, a dataframe was created to display the coefficients of each principal component. These loadings represent the weights or importance of the original variables in determining the corresponding principal component. A scatter plot was used to see the relationship between the first two principal components, which usually find the most significant variations in the data. In this scatter plot, each data point represents a country, and its position was determined by the score on the first two principal components. This plot gives information about the distribution and potential clustering of countries in the reduced dimensionality space defined by the top two principal components.

Then the explained variance of each principal component was examined, which quantified how much of the total variance in the data was explained by that component. The explained variance ratio was calculated, representing the explained variance of each component divided by the total variance. To determine the appropriate number of principal components to keep, two-line plots were created. The first plot showed the proportion of variance explained by each principal component, while the second plot displayed the cumulative proportion of variance explained by the principal components. These plots helped in the selection of the minimum number of components needed to explain a desired value of total variance, typically 80% or 90%.

Singular Value Decomposition (SVD):

The SVD is now used on the dataset. The dataset is first scaled, and it is decomposed using SVD, which divides the dataset into three components: U, s (a vector of singular values), and V. For checking the imputation performance, a subset of values in the data were set as missing. This is done by randomly selecting a set of rows and columns and replacing these values with the column means. A copy of the dataset with missing values is created and various variables are initialized, including a threshold for the relative error, the initial relative error, and variables to track the mean squared error.

After the imputation process, two plots are created to interpret and evaluate the result. The scree plot displays the squared singular values of the principal components, which can help determine the appropriate number of components to keep. The cumulative variance explained plot shows how much of the total variation in the data is captured by adding each principal component. This helps in deciding how many components are needed to keep enough information from the original data.

K-means Clustering:

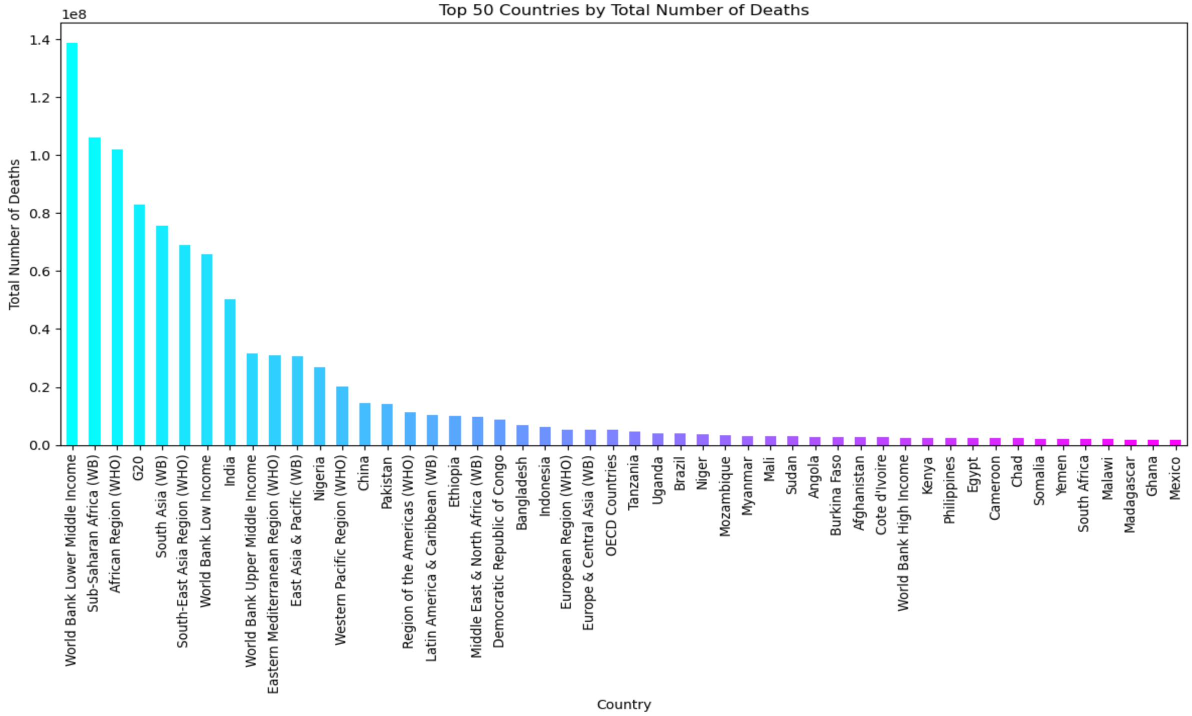
K-means clustering is used to group similar data points together into clusters. To find the optimal number of clusters for the dataset, the Elbow Method was used. This method involves running the K-Means algorithm multiple times with different values of K (the number of clusters) and calculating the inertia, which is a measure of the sum of squared distances between each data point and its assigned cluster centroid. The inertia values are then plotted against the corresponding K values, and the point at which the curve begins to flatten out (the "elbow") is considered as the optimal value of K. After finding the optimal value of K, the K-Means algorithm was executed with different values of K to visually explore the clustering patterns in the data. Scatter plots were generated, where each data point was colored according to its assigned cluster. These visualizations provided insights into the distribution and characteristics of the clusters. The analysis also investigated the impact of the "n_init" parameter on the clustering quality. The analysis compared the inertia values obtained with different combinations of K and n_init values, which allows the assessment of trade-off between clustering quality and computational complexity.

Hierarchical Clustering:

Hierarchical clustering is used here to find the groupings and structure within the dataset. The analysis begins by removing categorical columns such as country name, country code and year and creating a new dataframe. The resulting data frame is purely numeric data which will be used for clustering. Three different linkage methods were used: single, complete, and average linkage for plotting the dendrograms. The Centroid method was also used for comparison. These methods give the distance between the clusters. A custom function is created to plot the dendrograms for these linkage methods which was used to visualize the hierarchical clustering of the data. The same dendrogram plotting functions were then applied to the scaled data to observe any differences in the clustering results after standardizing the data. After plotting the dendrograms, the complete linkage method was chosen to cut the dendrograms and assign data points (countries) to clusters. The dendrograms were cut to form 5 clusters. A function was used to assign each country to a cluster and print the results. Each country was assigned to a cluster, and the results were printed. This helped in comparing the clustering results before and after scaling the data.

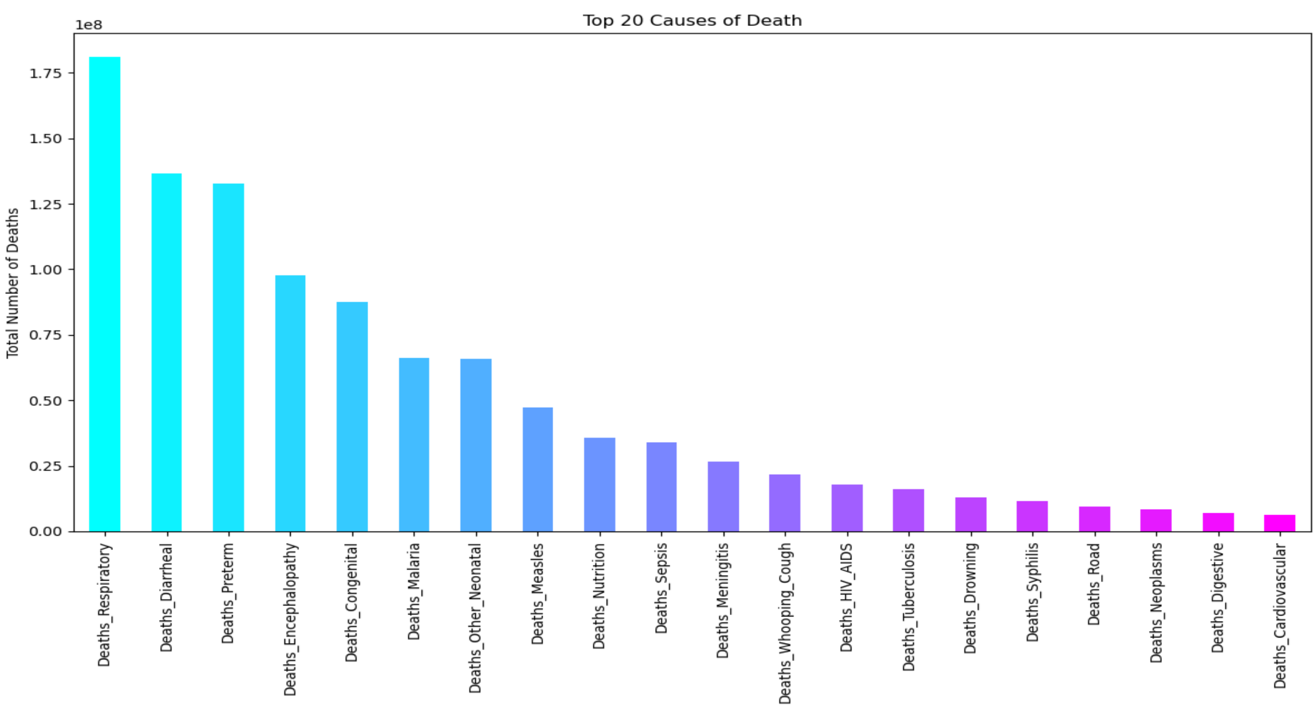
Computational Results:

Top 50 countries by total number of deaths:



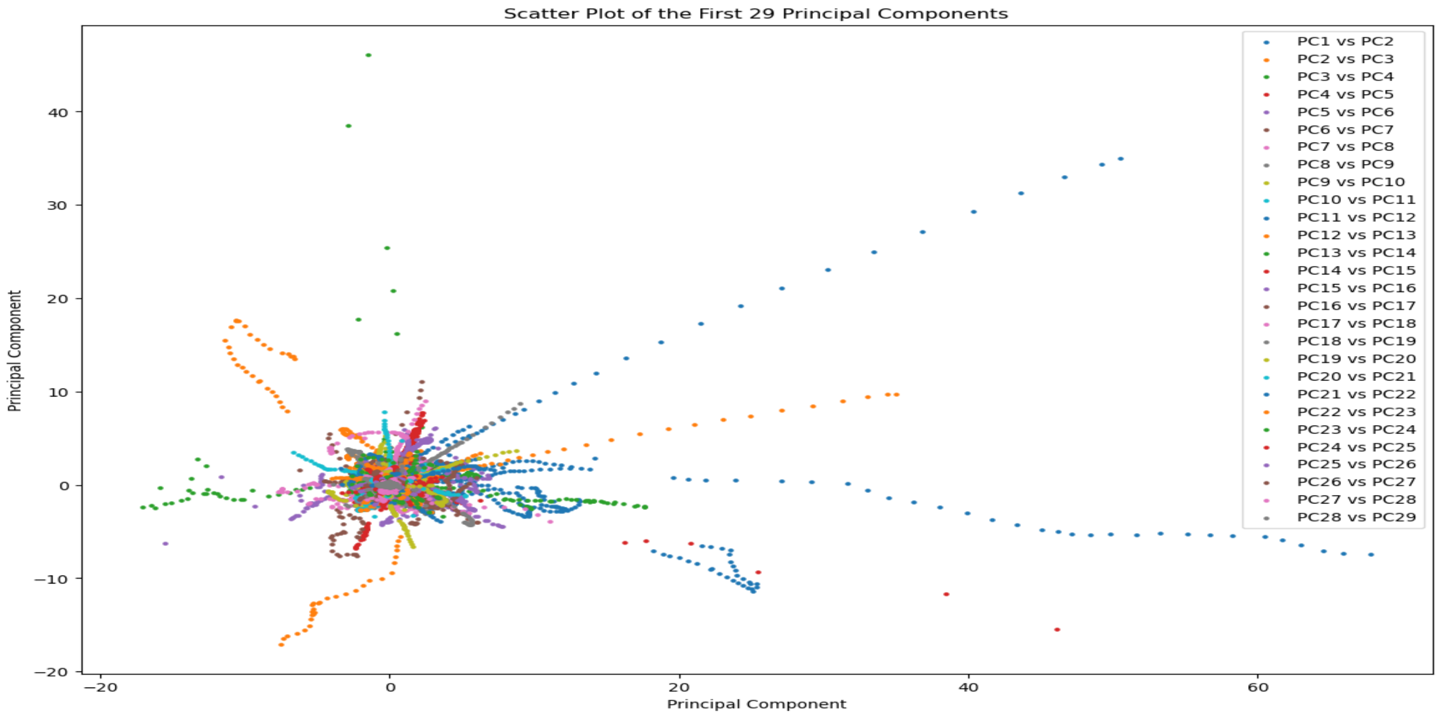
From all the countries/groups of countries around the world, the sum of all deaths of children under 5 from the years (1990-2019) are calculated and plotted to see which country/group of countries are present in the top 50.

Top 20 causes of deaths:

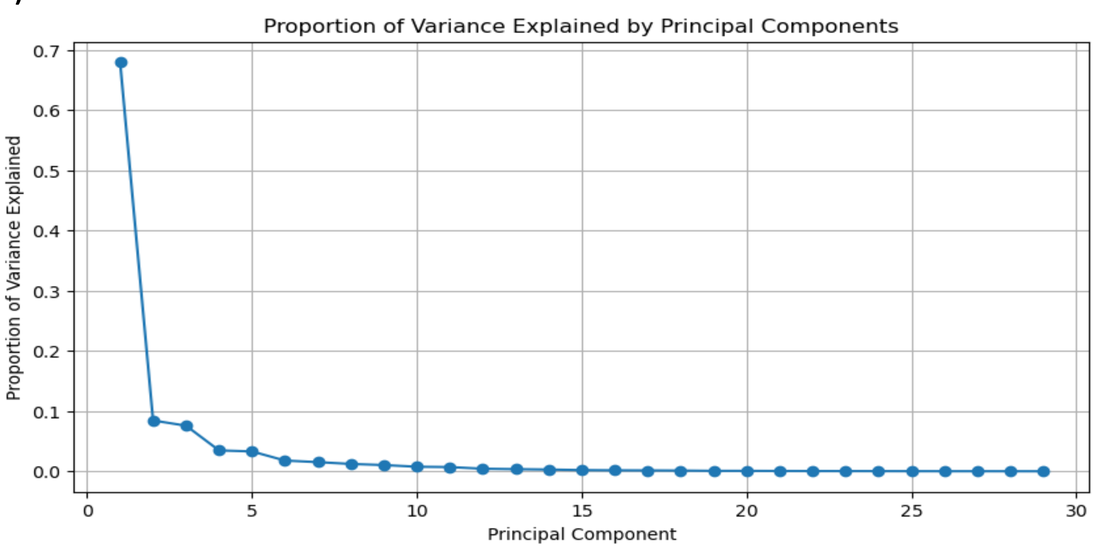


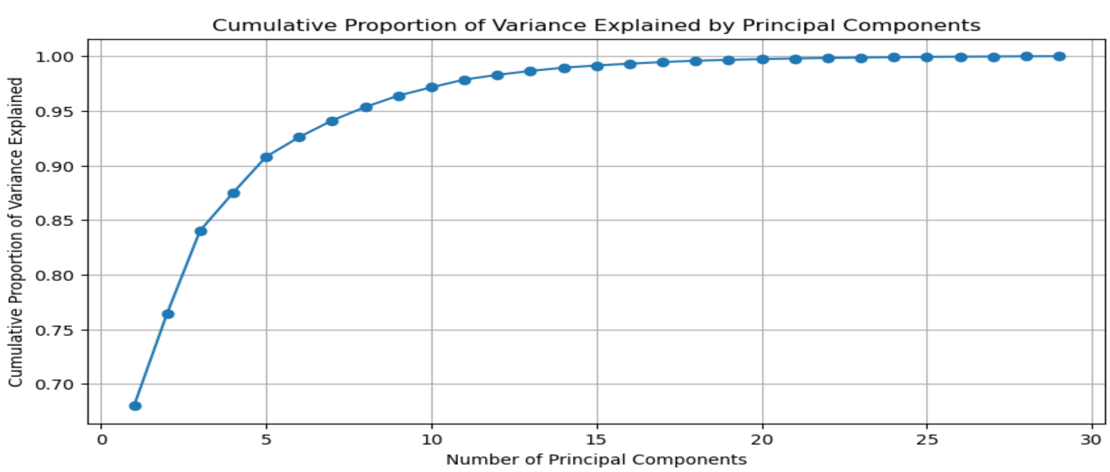
This plot was plotted to see which diseases/causes are causing most deaths around the world.

Scatter plot of principal components:

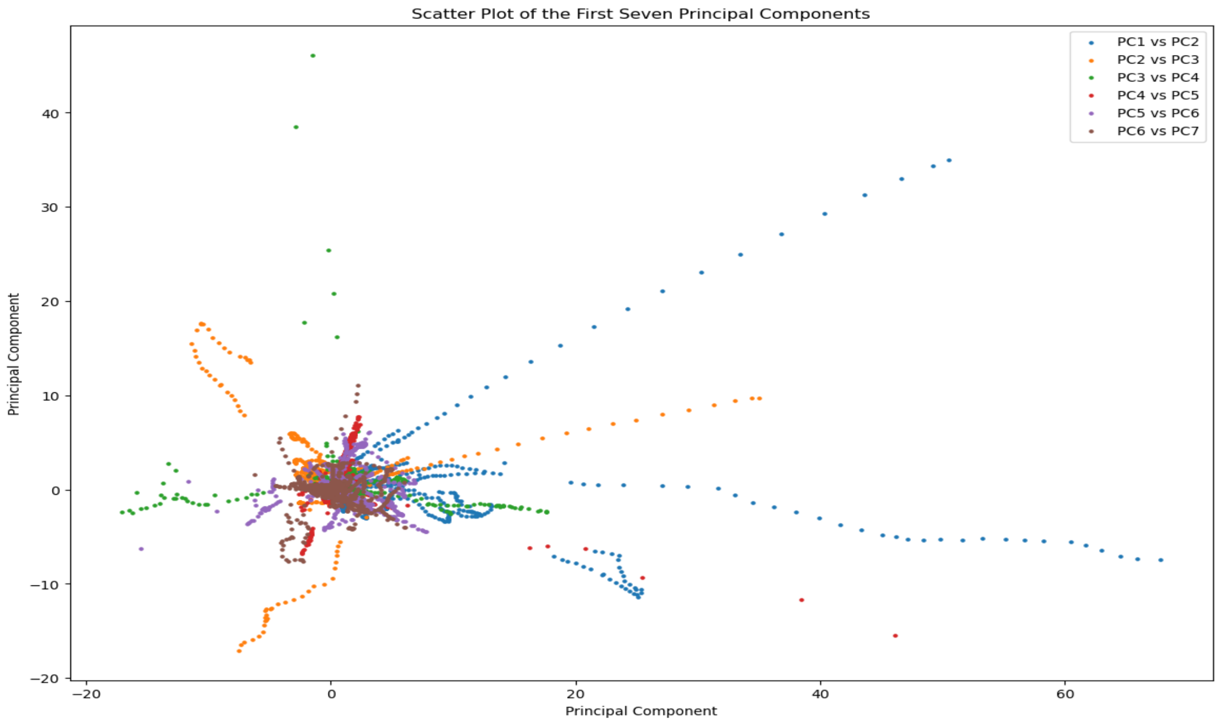


Scree plot (PCA):

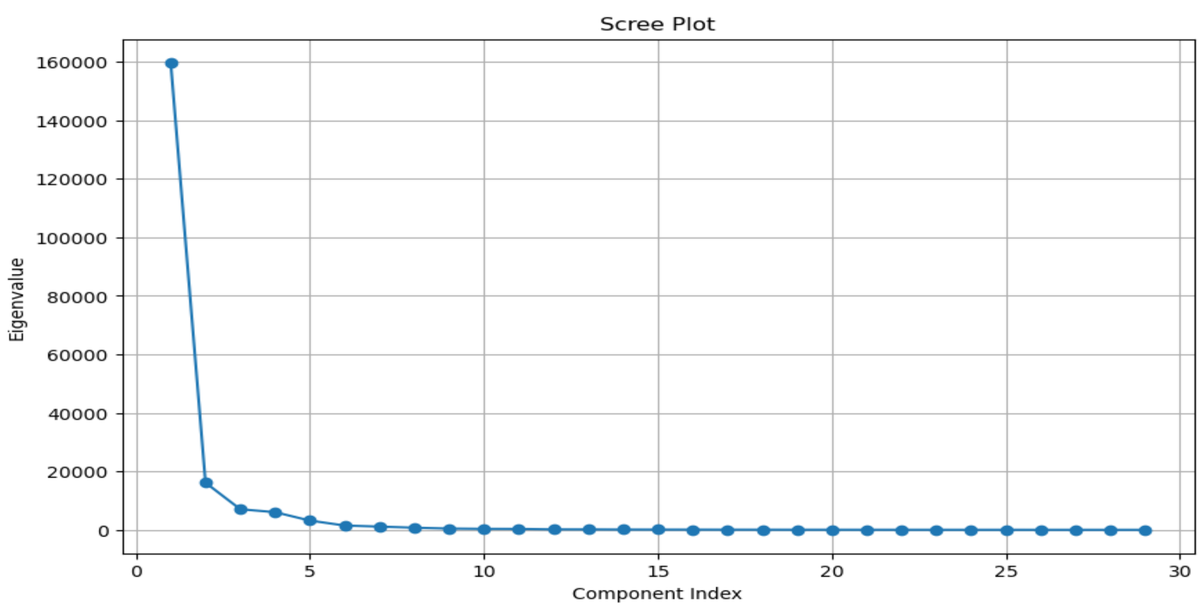


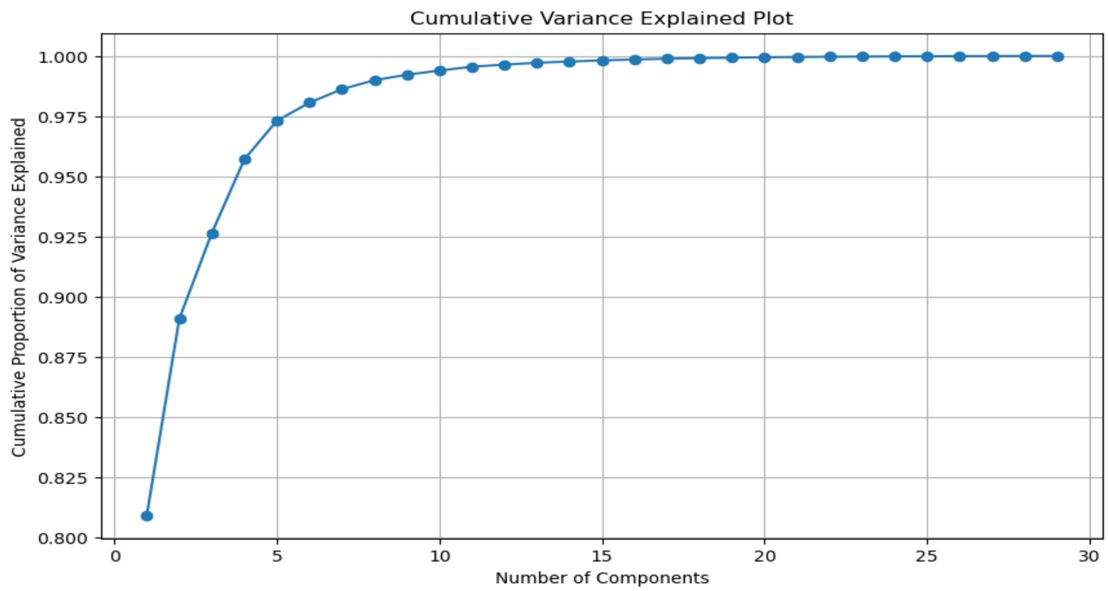


Scatter plot after Calculation of the Number of Principal Components:

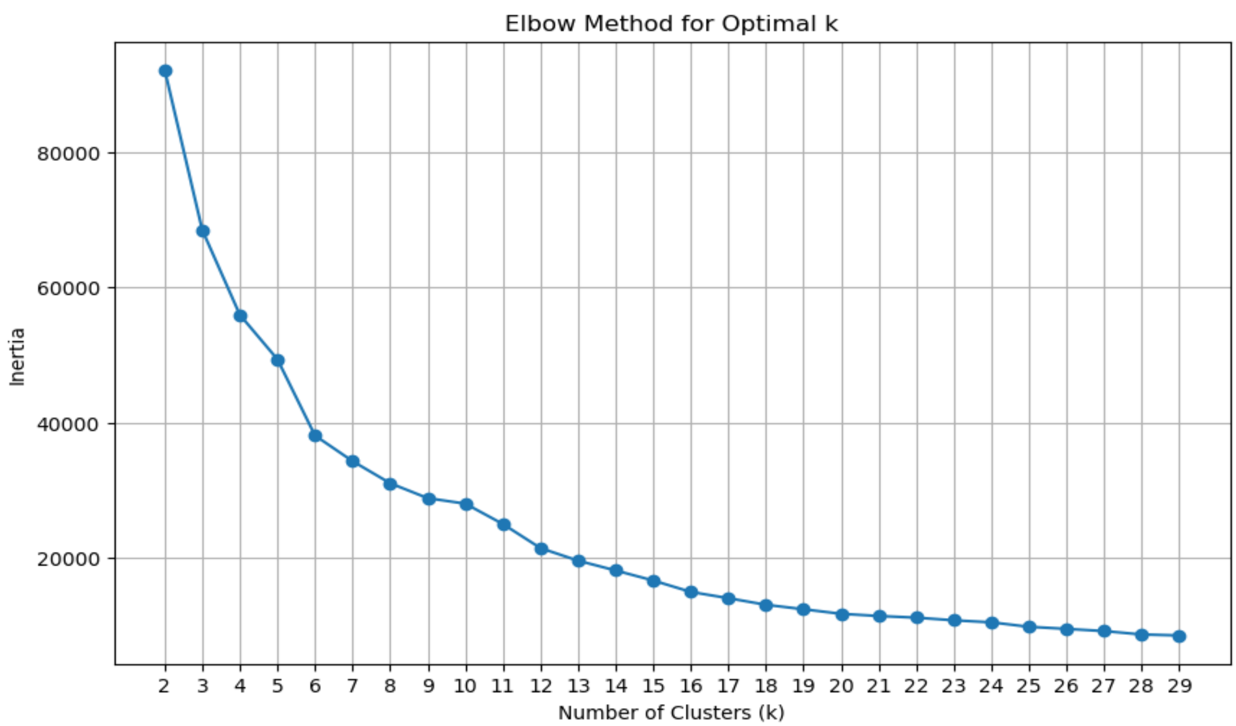


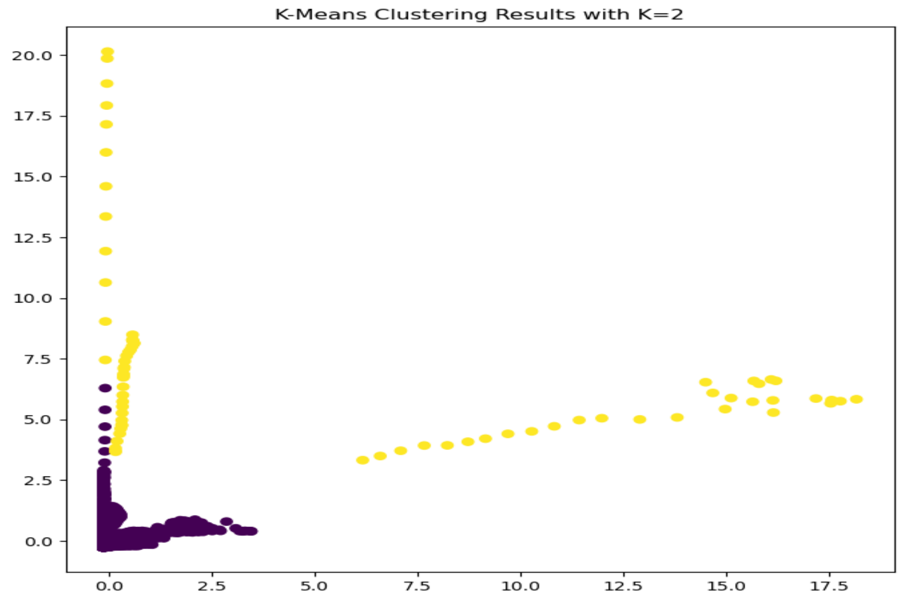
Scree plot (SVD):

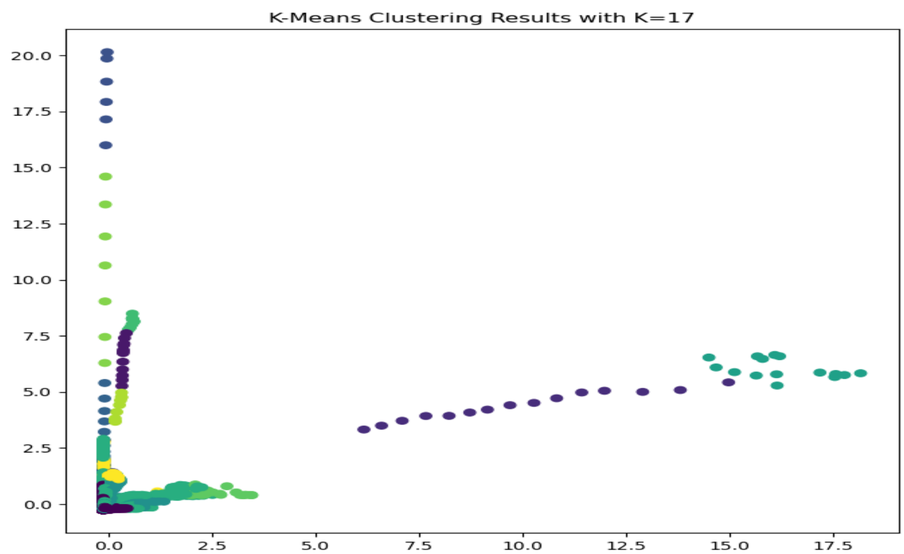


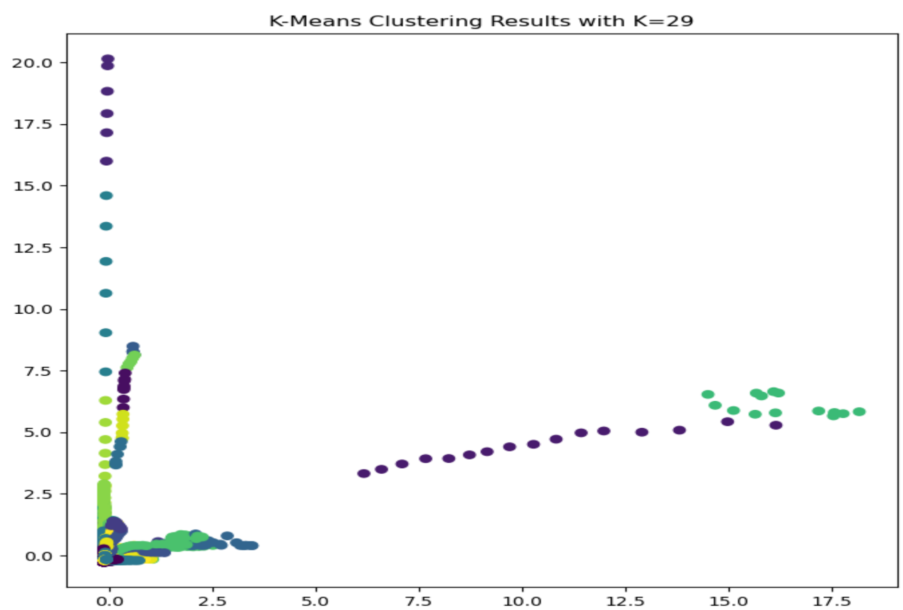


K-means Clustering:

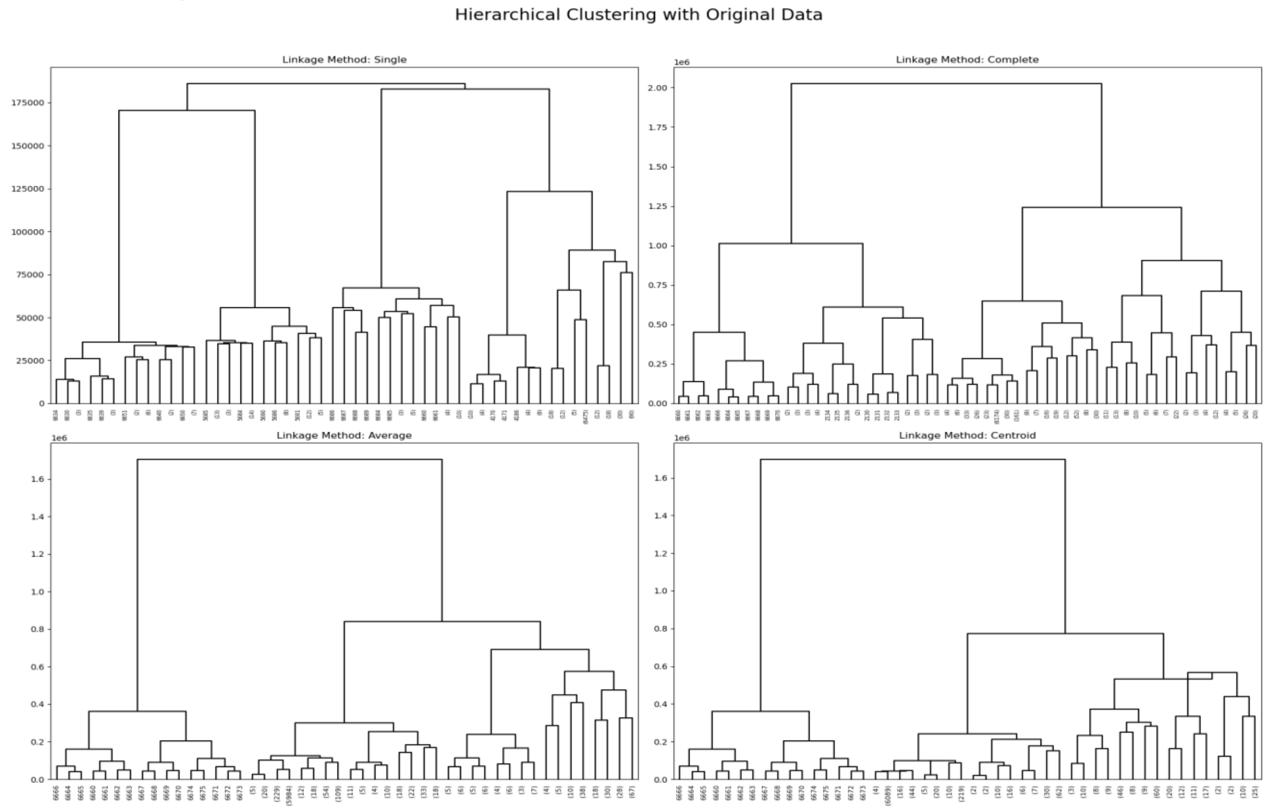


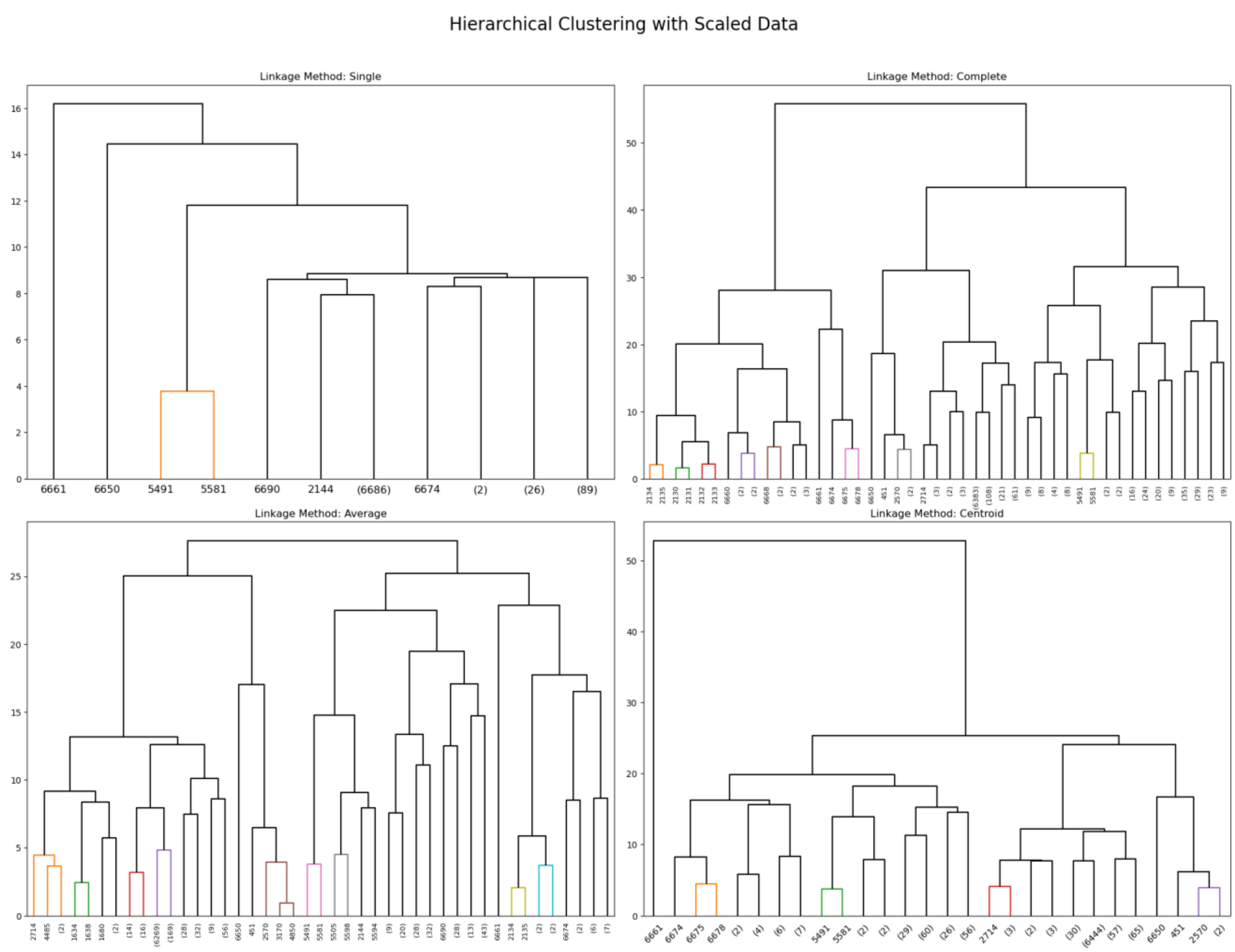

K-Means Clustering Results with K=2
20.0
17.5
15.0
12.5
10.0
7.5
5.0
2.5
0.0
0.0
2.5
5.0
7.5
10.0
12.5
15.0
17.5


K-Means Clustering Results with K=17
20.0
17.5
15.0
12.5
10.0
7.5
5.0
2.5
0.0
0.0
2.5
5.0
7.5
10.0
12.5
15.0
17.5


K-Means Clustering Results with K=29
20.0
17.5
15.0
12.5
10.0
7.5
5.0
2.5
0.0
0.0
2.5
5.0
7.5
10.0
12.5
15.0
17.5

Hierarchical Clustering:





Discussion:

The dataset provides numeric of mortality rate of children less than 5 years old worldwide which has a number of total deaths with 29 diseases/causes. The data is for 30 years from 1990 to 2019 per country. Some majorly affected continents were found to be Asia and Africa. India being the top country where the mortality rates are very high with an estimated deaths of around 50 million along with Nigeria with just lower than 30 million, China with 1.5 million, Pakistan with 1.5 million, and Ethiopia with around a million deaths from 1990 to 2019. The major cause of the deaths are through Respiratory diseases, diarrhea disorders and preterm births. All these factors can be linked to the air, water pollution in these countries. The plots show that the number of deaths are decreasing year by year. We can infer that the development of countries has led to improved access to essential services, improving living conditions.

While working on PCA, in the first analysis 29 principal components were used and each principal component has the values in each cell representing the contribution of the corresponding cause of death to each principal component. In the first component 'PC1', the highest number is with Respiratory deaths with 0.221859 followed by Preterm deaths with 0.212775 and Congenital deaths with 0.210769. The number suggests that these causes of death have a relatively strong influence on the principal component i.e., variations in these deaths have a notable impact on the overall mortality dynamics observed in the data. A scatter plot was plotted which provides the details regarding diseases that contribute more to the variance than others. It provides the relationships between pairs of principal components derived from the PCA analysis and the causes of deaths, each point represents a specific cause of death in the dataset, with its position determined by its scores on the corresponding principal components.

The variance has given a definite number of components with 7 being the best. The graph shows the drop of the trend at 7(0.205), it is almost 0 from there onwards. The model has been refitted using 7 principal components as these capture a significant amount of variance in the data interpretation. SVD was done to check the decomposing of the dataset into parts, which gives patterns and relationships between variables. But since we don't have any missing values in the current dataset, it provides insights into the structure of the data, identifies dominant features, and helps dimensionality reduction through the extraction of principal components. The result is identical to the result of the PCA model.

For K-means clustering, the elbow method was applied to determine the right number of clusters for dividing the data. The plot gave a much lesser inertia as the number of clusters increased. While varying the number of k-means clusters (K=2, K=17, K=29) and varying number of initializations (1 and 20), the scatter plots were plotted and are bell-shaped and cover the range of the x and y axis from negative to positive values. This distribution with an L-shape clearly indicates that the clusters are well separated from each other with slight overlapping. This infers that the data has a lot of variation, which were well divided into clusters. A higher inertia value of 92,144 only with two clusters indicates that the data points are away from each other and do not form dual clusters. Lower inertia values, like 17 clusters (14,188), indicate increased clustering and better placing of points in clusters. Reduced inertia values for 29 clusters (7,837) mean still higher clustering efficiency. The inertia values between different initialization strategies corresponding to n_init=1 and n_init=20 give a better understanding of the clustering stability and strength of the algorithms.

For hierarchical clustering, different linkage methods like 'single', 'complete', 'average' and 'centroid' and dendrograms enabled different instances of hierarchical structures which highlighted the distances between the data points. The dendrograms drawn on the Original data have the best model under 'centroid' and 'complete' while the dendrograms on scaled data have 'centroid' and 'complete' as the best model without any overlaps. Further clustering of the entities based on the developed structures gave significant results indicating sharp separations according to geographical classifications. For example, it can be observed from the original data sets that some of the identified clusters were around countries such as China or India, which involves its distinct characteristics and affinity to the neighboring areas. Likewise, the scaled data grouped countries that have features, whereby Bangladesh, Haiti, Indonesia, Iran, Myanmar, and Pakistan belonged to the same group. The results shown above indicate that this approach was useful for decision-making.

Conclusions:

This study looked at data on deaths of children under 5 years old from different countries over 30 years. The data showed that the countries with the highest number of child deaths were mostly in South Asia and Africa, like India, Nigeria, Pakistan, and Ethiopia. The main causes of death were respiratory diseases, diarrhea, and preterm birth issues, which could be linked to poor air quality and contaminated water in those regions. The study used advanced unsupervised learning techniques like Principal Component Analysis, K-Means clustering, and Hierarchical Clustering to find hidden patterns and groups in this large dataset. These methods helped reveal groups of countries with similar causes of child deaths, as well as outlier countries. The analyses also showed how some causes of death, like respiratory issues, contributed more to the overall patterns.

This study was able to find hidden patterns, trends, and relationships in the complex data on causes of death in children under 5 across countries globally. The findings from these analyses help policymakers and health organizations make more informed decisions about public health policies, resource allocation, and targeted programs focused on reducing child mortality rates and improving overall health outcomes in various countries and regions worldwide. The research shows how powerful unsupervised machine learning methods can be for exploratory data analysis and finding patterns in data, mainly in areas like health. Overall, this study helps us understand why child deaths are happening and what measures can be taken. By studying this report, one can find better ways to improve the health and well-being of children globally, which is important for the future.

Bibliography/References:

[1] Saloni Dattani, Fiona Spooner, Hannah Ritchie and Max Roser. *Diarrheal Diseases*. IHME, Global Burden of Disease (2019), Our World in Data.
https://ourworldindata.org/diarrheal-diseases#article-citation

[2] Data Camp, Introduction to Unsupervised Learning.
https://www.datacamp.com/blog/introduction-to-unsupervised-learning

[3] Muhammad Islam Satti, Mir Wajid Ali, Azeem Irshad, and Mohd Asif Shah. *Studying infant mortality: A demographic analysis based on data mining models*. National Library of Medicine, PMCID: PMC10358750. July, 2023.
https://aws.amazon.com/what-is/neural-network/#:~:text=A%20neural%20network%20is%20a,that%20resembles%20the%20human%20brain

[4] The Data Scientist, Machine Learning Tutorial: A Practical Guide Of Unsupervised Learning Algorithms. https://thedatascientist.com/machine-learning-tutorial-a-practical-guide-of-unsupervised-learning-algorithms/

[5] Deep AI, Unsupervised Learning.
https://images.app.goo.gl/EZPwrPKkyWoUbmZc8

[6] Geeks for geeks, Unsupervised Learning.
https://www.geeksforgeeks.org/ml-types-learning-part-2/

[7] IBM, What is Unsupervised Learning?
https://www.ibm.com/topics/unsupervised-learning

[8] MathWorks, Dendrogram.
https://www.mathworks.com/help/stats/dendrogram.html

6gdp8pkjz

May 27, 2024

[1]:
```
import pandas as pd
import seaborn as sns
from scipy.stats import zscore
import matplotlib.pyplot as plt
import numpy as np
import plotly.express as p
from sklearn.decomposition import PCA
from sklearn.preprocessing import StandardScaler
from sklearn.cluster import KMeans
from sklearn.cluster import AgglomerativeClustering
from scipy.cluster.hierarchy import dendrogram, cut_tree
from ISLP.cluster import compute_linkage
from scipy.cluster.hierarchy import linkage
```

[2]:
```
file_path = r"C:\Users\tiles\Downloads\causes-of-death-in-children-under-5.csv"
df = pd.read_csv(file_path)
df.head()
```

[2]:
```
        Entity Code  Year  \
0  Afghanistan  AFG  1990
1  Afghanistan  AFG  1991
2  Afghanistan  AFG  1992
3  Afghanistan  AFG  1993
4  Afghanistan  AFG  1994

   Deaths - Invasive Non-typhoidal Salmonella (iNTS) - Sex: Both - Age: Under 5
(Number)  \
0                                                 48
1                                                 55
2                                                 68
3                                                 78
4                                                 83

   Deaths - Interpersonal violence - Sex: Both - Age: Under 5 (Number)  \
0                                                105
1                                                130
2                                                155
```

```
3                                              178
4                                              194

   Deaths - Nutritional deficiencies - Sex: Both - Age: Under 5 (Number)  \
0                                               1779
1                                               1822
2                                               2069
3                                               2427
4                                               2649

   Deaths - Acute hepatitis - Sex: Both - Age: Under 5 (Number)  \
0                                               718
1                                               741
2                                               836
3                                               970
4                                              1063

   Deaths - Neoplasms - Sex: Both - Age: Under 5 (Number)  \
0                                               431
1                                               439
2                                               486
3                                               549
4                                               589

   Deaths - Measles - Sex: Both - Age: Under 5 (Number)  \
0                                              8649
1                                              8669
2                                              8539
3                                              8949
4                                             10642

   Deaths - Digestive diseases - Sex: Both - Age: Under 5 (Number)  …  \
0                                               477                 …
1                                               495                 …
2                                               554                 …
3                                               630                 …
4                                               681                 …

   Deaths - Other neonatal disorders - Sex: Both - Age: Under 5 (Number)  \
0                                              7112
1                                              7574
2                                              8614
3                                              9458
4                                              9823

   Deaths - Whooping cough - Sex: Both - Age: Under 5 (Number)  \
0                                              2455
```

```
1                                              2385
2                                              2370
3                                              2659
4                                              3187

   Deaths - Diarrheal diseases - Sex: Both - Age: Under 5 (Number)  \
0                                              3968
1                                              4650
2                                              5833
3                                              7800
4                                              7894

   Deaths - Fire, heat, and hot substances - Sex: Both - Age: Under 5 (Number)
\
0                                               131
1                                               129
2                                               137
3                                               155
4                                               170

   Deaths - Road injuries - Sex: Both - Age: Under 5 (Number)  \
0                                               802
1                                               781
2                                               821
3                                               923
4                                              1015

   Deaths - Tuberculosis - Sex: Both - Age: Under 5 (Number)  \
0                                               808
1                                               800
2                                               863
3                                               979
4                                              1064

   Deaths - HIV/AIDS - Sex: Both - Age: Under 5 (Number)  \
0                                                10
1                                                12
2                                                13
3                                                16
4                                                19

   Deaths - Drowning - Sex: Both - Age: Under 5 (Number)  \
0                                               776
1                                               748
2                                               777
3                                               872
4                                               961
```

```
   Deaths - Malaria - Sex: Both - Age: Under 5 (Number)  \
0                                                   21
1                                                   41
2                                                   51
3                                                   24
4                                                   52

   Deaths - Syphilis - Sex: Both - Age: Under 5 (Number)
0                                                  123
1                                                  132
2                                                  180
3                                                  239
4                                                  259

[5 rows x 32 columns]
```

[3]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 6840 entries, 0 to 6839
Data columns (total 32 columns):
 #   Column
Non-Null Count  Dtype
---  ------
--------------  -----
 0   Entity
6840 non-null   object
 1   Code
6150 non-null   object
 2   Year
6840 non-null   int64
 3   Deaths - Invasive Non-typhoidal Salmonella (iNTS) - Sex: Both - Age: Under
5 (Number)                    6840 non-null   int64
 4   Deaths - Interpersonal violence - Sex: Both - Age: Under 5 (Number)
6840 non-null   int64
 5   Deaths - Nutritional deficiencies - Sex: Both - Age: Under 5 (Number)
6840 non-null   int64
 6   Deaths - Acute hepatitis - Sex: Both - Age: Under 5 (Number)
6840 non-null   int64
 7   Deaths - Neoplasms - Sex: Both - Age: Under 5 (Number)
6840 non-null   int64
 8   Deaths - Measles - Sex: Both - Age: Under 5 (Number)
6840 non-null   int64
 9   Deaths - Digestive diseases - Sex: Both - Age: Under 5 (Number)
6840 non-null   int64
 10  Deaths - Cirrhosis and other chronic liver diseases - Sex: Both - Age:
```

```
Under 5 (Number)                6840 non-null   int64
 11  Deaths - Chronic kidney disease - Sex: Both - Age: Under 5 (Number)
6840 non-null   int64
 12  Deaths - Cardiovascular diseases - Sex: Both - Age: Under 5 (Number)
6840 non-null   int64
 13  Deaths - Congenital birth defects - Sex: Both - Age: Under 5 (Number)
6840 non-null   int64
 14  Deaths - Lower respiratory infections - Sex: Both - Age: Under 5 (Number)
6840 non-null   int64
 15  Deaths - Neonatal preterm birth - Sex: Both - Age: Under 5 (Number)
6840 non-null   int64
 16  Deaths - Environmental heat and cold exposure - Sex: Both - Age: Under 5
(Number)                      6840 non-null   int64
 17  Deaths - Neonatal sepsis and other neonatal infections - Sex: Both - Age:
Under 5 (Number)             6840 non-null   int64
 18  Deaths - Exposure to forces of nature - Sex: Both - Age: Under 5 (Number)
6840 non-null   int64
 19  Deaths - Diabetes mellitus - Sex: Both - Age: Under 5 (Number)
6840 non-null   int64
 20  Deaths - Neonatal encephalopathy due to birth asphyxia and trauma - Sex:
Both - Age: Under 5 (Number)  6840 non-null   int64
 21  Deaths - Meningitis - Sex: Both - Age: Under 5 (Number)
6840 non-null   int64
 22  Deaths - Other neonatal disorders - Sex: Both - Age: Under 5 (Number)
6840 non-null   int64
 23  Deaths - Whooping cough - Sex: Both - Age: Under 5 (Number)
6840 non-null   int64
 24  Deaths - Diarrheal diseases - Sex: Both - Age: Under 5 (Number)
6840 non-null   int64
 25  Deaths - Fire, heat, and hot substances - Sex: Both - Age: Under 5 (Number)
6840 non-null   int64
 26  Deaths - Road injuries - Sex: Both - Age: Under 5 (Number)
6840 non-null   int64
 27  Deaths - Tuberculosis - Sex: Both - Age: Under 5 (Number)
6840 non-null   int64
 28  Deaths - HIV/AIDS - Sex: Both - Age: Under 5 (Number)
6840 non-null   int64
 29  Deaths - Drowning - Sex: Both - Age: Under 5 (Number)
6840 non-null   int64
 30  Deaths - Malaria - Sex: Both - Age: Under 5 (Number)
6840 non-null   int64
 31  Deaths - Syphilis - Sex: Both - Age: Under 5 (Number)
6840 non-null   int64
dtypes: int64(30), object(2)
memory usage: 1.7+ MB
```

[4]:
```
df.describe()
```

[4]:
```
               Year  \
count  6840.000000
mean   2004.500000
std       8.656074
min    1990.000000
25%    1997.000000
50%    2004.500000
75%    2012.000000
max    2019.000000

       Deaths - Invasive Non-typhoidal Salmonella (iNTS) - Sex: Both - Age:
Under 5 (Number)  \
count                                        6840.000000
mean                                         1041.740789
std                                          5943.506061
min                                             0.000000
25%                                             0.000000
50%                                             1.000000
75%                                            42.000000
max                                         62334.000000

       Deaths - Interpersonal violence - Sex: Both - Age: Under 5 (Number)  \
count                                        6840.000000
mean                                          399.418567
std                                          1549.064285
min                                             0.000000
25%                                             1.000000
50%                                            10.000000
75%                                            77.000000
max                                         21223.000000

       Deaths - Nutritional deficiencies - Sex: Both - Age: Under 5 (Number)  \
count                                        6840.000000
mean                                         6392.636842
std                                         30815.191076
min                                             0.000000
25%                                             1.000000
50%                                            17.000000
75%                                           903.250000
max                                        524103.000000

       Deaths - Acute hepatitis - Sex: Both - Age: Under 5 (Number)  \
count                                        6840.000000
mean                                          826.106433
std                                          4399.035288
min                                             0.000000
25%                                             0.000000
```

```
50%                                                         2.000000
75%                                                        36.000000
max                                                     50184.000000

       Deaths - Neoplasms - Sex: Both - Age: Under 5 (Number)  \
count                                                6840.000000
mean                                                 1472.511550
std                                                  5794.139457
min                                                     0.000000
25%                                                     7.000000
50%                                                    44.000000
75%                                                   283.000000
max                                                 85197.000000

       Deaths - Measles - Sex: Both - Age: Under 5 (Number)  \
count                                              6840.000000
mean                                               8527.408772
std                                               43502.336767
min                                                   0.000000
25%                                                   0.000000
50%                                                   1.000000
75%                                                 655.000000
max                                              704288.000000

       Deaths - Digestive diseases - Sex: Both - Age: Under 5 (Number)  \
count                                                6840.000000
mean                                                 1235.561696
std                                                  5006.538348
min                                                     0.000000
25%                                                     2.000000
50%                                                    25.000000
75%                                                   290.250000
max                                                 77952.000000

       Deaths - Cirrhosis and other chronic liver diseases - Sex: Both - Age:
Under 5 (Number)  \
count                                                6840.000000
mean                                                  262.530409
std                                                  1119.050195
min                                                     0.000000
25%                                                     0.000000
50%                                                     4.000000
75%                                                    52.000000
max                                                 15916.000000

       Deaths - Chronic kidney disease - Sex: Both - Age: Under 5 (Number)  \
count                                                6840.000000
```

```
mean                                                304.512135
std                                                1208.144730
min                                                   0.000000
25%                                                   1.000000
50%                                                   7.000000
75%                                                  78.000000
max                                               18047.000000

       …  \
count  …
mean   …
std    …
min    …
25%    …
50%    …
75%    …
max    …

       Deaths - Other neonatal disorders - Sex: Both - Age: Under 5 (Number)  \
count                                              6840.000000
mean                                              11726.084795
std                                               51612.890640
min                                                   0.000000
25%                                                  18.750000
50%                                                 178.000000
75%                                                1773.250000
max                                              539952.000000

       Deaths - Whooping cough - Sex: Both - Age: Under 5 (Number)  \
count                                              6840.000000
mean                                               3876.111696
std                                               16817.425576
min                                                   0.000000
25%                                                   0.000000
50%                                                  21.000000
75%                                                 671.000000
max                                              240021.000000

       Deaths - Diarrheal diseases - Sex: Both - Age: Under 5 (Number)  \
count                                             6.840000e+03
mean                                              2.464864e+04
std                                               1.133132e+05
min                                               0.000000e+00
25%                                               3.000000e+00
50%                                               9.800000e+01
75%                                               3.822250e+03
max                                               1.649581e+06
```

```
       Deaths - Fire, heat, and hot substances - Sex: Both - Age: Under 5
(Number)  \
count                                        6840.000000
mean                                          535.363743
std                                          2156.814008
min                                             0.000000
25%                                             2.000000
50%                                            15.000000
75%                                           142.000000
max                                         35583.000000

       Deaths - Road injuries - Sex: Both - Age: Under 5 (Number)  \
count                                        6840.000000
mean                                         1718.877047
std                                          6934.211045
min                                             0.000000
25%                                             4.000000
50%                                            38.500000
75%                                           359.250000
max                                        115624.000000

       Deaths - Tuberculosis - Sex: Both - Age: Under 5 (Number)  \
count                                        6840.000000
mean                                         2892.662719
std                                         13333.898943
min                                             0.000000
25%                                             0.000000
50%                                            11.000000
75%                                           388.000000
max                                        209562.000000

       Deaths - HIV/AIDS - Sex: Both - Age: Under 5 (Number)  \
count                                        6840.000000
mean                                         3252.680702
std                                         18169.939174
min                                             0.000000
25%                                             0.000000
50%                                             7.000000
75%                                           298.250000
max                                        223680.000000

       Deaths - Drowning - Sex: Both - Age: Under 5 (Number)  \
count                                        6840.000000
mean                                         2331.720468
std                                         10832.408381
min                                             0.000000
```

```
25%                                                 3.000000
50%                                                27.000000
75%                                               291.250000
max                                            184096.000000

       Deaths - Malaria - Sex: Both - Age: Under 5 (Number)  \
count                                           6840.000000
mean                                           12045.209064
std                                            64858.902628
min                                                0.000000
25%                                                0.000000
50%                                                0.000000
75%                                              217.250000
max                                           631523.000000

       Deaths - Syphilis - Sex: Both - Age: Under 5 (Number)
count                                           6840.000000
mean                                            2107.161111
std                                             9180.674890
min                                                0.000000
25%                                                0.000000
50%                                               12.000000
75%                                              277.250000
max                                            99248.000000

[8 rows x 30 columns]
```

[5]:
```
df.isnull().sum()
```

[5]:
```
Entity
0
Code
690
Year
0
Deaths - Invasive Non-typhoidal Salmonella (iNTS) - Sex: Both - Age: Under 5
(Number)                              0
Deaths - Interpersonal violence - Sex: Both - Age: Under 5 (Number)
0
Deaths - Nutritional deficiencies - Sex: Both - Age: Under 5 (Number)
0
Deaths - Acute hepatitis - Sex: Both - Age: Under 5 (Number)
0
Deaths - Neoplasms - Sex: Both - Age: Under 5 (Number)
0
Deaths - Measles - Sex: Both - Age: Under 5 (Number)
0
```

```
Deaths - Digestive diseases - Sex: Both - Age: Under 5 (Number)
0
Deaths - Cirrhosis and other chronic liver diseases - Sex: Both - Age: Under 5
(Number)                    0
Deaths - Chronic kidney disease - Sex: Both - Age: Under 5 (Number)
0
Deaths - Cardiovascular diseases - Sex: Both - Age: Under 5 (Number)
0
Deaths - Congenital birth defects - Sex: Both - Age: Under 5 (Number)
0
Deaths - Lower respiratory infections - Sex: Both - Age: Under 5 (Number)
0
Deaths - Neonatal preterm birth - Sex: Both - Age: Under 5 (Number)
0
Deaths - Environmental heat and cold exposure - Sex: Both - Age: Under 5
(Number)                          0
Deaths - Neonatal sepsis and other neonatal infections - Sex: Both - Age: Under
5 (Number)                0
Deaths - Exposure to forces of nature - Sex: Both - Age: Under 5 (Number)
0
Deaths - Diabetes mellitus - Sex: Both - Age: Under 5 (Number)
0
Deaths - Neonatal encephalopathy due to birth asphyxia and trauma - Sex: Both -
Age: Under 5 (Number)      0
Deaths - Meningitis - Sex: Both - Age: Under 5 (Number)
0
Deaths - Other neonatal disorders - Sex: Both - Age: Under 5 (Number)
0
Deaths - Whooping cough - Sex: Both - Age: Under 5 (Number)
0
Deaths - Diarrheal diseases - Sex: Both - Age: Under 5 (Number)
0
Deaths - Fire, heat, and hot substances - Sex: Both - Age: Under 5 (Number)
0
Deaths - Road injuries - Sex: Both - Age: Under 5 (Number)
0
Deaths - Tuberculosis - Sex: Both - Age: Under 5 (Number)
0
Deaths - HIV/AIDS - Sex: Both - Age: Under 5 (Number)
0
Deaths - Drowning - Sex: Both - Age: Under 5 (Number)
0
Deaths - Malaria - Sex: Both - Age: Under 5 (Number)
0
Deaths - Syphilis - Sex: Both - Age: Under 5 (Number)
0
dtype: int64
```

[6]:
```
missing_code_entities_unique = df.loc[df['Code'].isnull(), 'Entity'].unique()
missing_code_entities_unique
```

[6]:
```
array(['African Region (WHO)', 'East Asia & Pacific (WB)',
       'Eastern Mediterranean Region (WHO)', 'England',
       'Europe & Central Asia (WB)', 'European Region (WHO)', 'G20',
       'Latin America & Caribbean (WB)',
       'Middle East & North Africa (WB)', 'North America (WB)',
       'Northern Ireland', 'OECD Countries',
       'Region of the Americas (WHO)', 'Scotland', 'South Asia (WB)',
       'South-East Asia Region (WHO)', 'Sub-Saharan Africa (WB)', 'Wales',
       'Western Pacific Region (WHO)', 'World Bank High Income',
       'World Bank Low Income', 'World Bank Lower Middle Income',
       'World Bank Upper Middle Income'], dtype=object)
```

[7]:
```
df = df.dropna()
```

[8]:
```
indices_to_drop = df[df['Entity'] == 'World'].index
df = df.drop(indices_to_drop)
```

[9]:
```
df
```

[9]:
```
           Entity Code  Year  \
0     Afghanistan  AFG  1990
1     Afghanistan  AFG  1991
2     Afghanistan  AFG  1992
3     Afghanistan  AFG  1993
4     Afghanistan  AFG  1994
…             …    …     …
6835     Zimbabwe  ZWE  2015
6836     Zimbabwe  ZWE  2016
6837     Zimbabwe  ZWE  2017
6838     Zimbabwe  ZWE  2018
6839     Zimbabwe  ZWE  2019

      Deaths - Invasive Non-typhoidal Salmonella (iNTS) - Sex: Both - Age: Under
5 (Number)  \
0                                                    48
1                                                    55
2                                                    68
3                                                    78
4                                                    83
…                                                   …
6835                                                106
6836                                                112
6837                                                111
6838                                                109
```

```
6839                                              108

      Deaths - Interpersonal violence - Sex: Both - Age: Under 5 (Number)  \
0                                                 105
1                                                 130
2                                                 155
3                                                 178
4                                                 194
…                                               …
6835                                               31
6836                                               32
6837                                               32
6838                                               31
6839                                               31

      Deaths - Nutritional deficiencies - Sex: Both - Age: Under 5 (Number)  \
0                                                1779
1                                                1822
2                                                2069
3                                                2427
4                                                2649
…                                               …
6835                                             1733
6836                                             1771
6837                                             1714
6838                                             1639
6839                                             1598

      Deaths - Acute hepatitis - Sex: Both - Age: Under 5 (Number)  \
0                                                 718
1                                                 741
2                                                 836
3                                                 970
4                                                1063
…                                               …
6835                                               17
6836                                               18
6837                                               17
6838                                               16
6839                                               15

      Deaths - Neoplasms - Sex: Both - Age: Under 5 (Number)  \
0                                                 431
1                                                 439
2                                                 486
3                                                 549
4                                                 589
```

```
…                                                 …
6835                                               56
6836                                               58
6837                                               58
6838                                               58
6839                                               57

      Deaths - Measles - Sex: Both - Age: Under 5 (Number)  \
0                                                  8649
1                                                  8669
2                                                  8539
3                                                  8949
4                                                 10642
…                                                 …
6835                                                615
6836                                                369
6837                                                261
6838                                                340
6839                                                349

      Deaths - Digestive diseases - Sex: Both - Age: Under 5 (Number)  …  \
0                                                   477                …
1                                                   495                …
2                                                   554                …
3                                                   630                …
4                                                   681                …
…                                                 …                …
6835                                                 92                …
6836                                                 95                …
6837                                                 94                …
6838                                                 91                …
6839                                                 89                …

      Deaths - Other neonatal disorders - Sex: Both - Age: Under 5 (Number)  \
0                                                  7112
1                                                  7574
2                                                  8614
3                                                  9458
4                                                  9823
…                                                 …
6835                                               2269
6836                                               2249
6837                                               2245
6838                                               2203
6839                                               2190

      Deaths - Whooping cough - Sex: Both - Age: Under 5 (Number)  \
```

```
0                                                    2455
1                                                    2385
2                                                    2370
3                                                    2659
4                                                    3187
…                                                   …
6835                                                  518
6836                                                  559
6837                                                  544
6838                                                  568
6839                                                  536

      Deaths - Diarrheal diseases - Sex: Both - Age: Under 5 (Number)  \
0                                                    3968
1                                                    4650
2                                                    5833
3                                                    7800
4                                                    7894
…                                                   …
6835                                                 1345
6836                                                 1286
6837                                                 1248
6838                                                 1136
6839                                                 1067

      Deaths - Fire, heat, and hot substances - Sex: Both - Age: Under 5
(Number)  \
0                                                     131
1                                                     129
2                                                     137
3                                                     155
4                                                     170
…                                                   …
6835                                                  114
6836                                                  119
6837                                                  117
6838                                                  114
6839                                                  112

      Deaths - Road injuries - Sex: Both - Age: Under 5 (Number)  \
0                                                     802
1                                                     781
2                                                     821
3                                                     923
4                                                    1015
…                                                   …
6835                                                  115
```

```
6836                                                  120
6837                                                  119
6838                                                  115
6839                                                  112

      Deaths - Tuberculosis - Sex: Both - Age: Under 5 (Number)  \
0                                                     808
1                                                     800
2                                                     863
3                                                     979
4                                                    1064
…                                                   …
6835                                                  799
6836                                                  787
6837                                                  745
6838                                                  693
6839                                                  661

      Deaths - HIV/AIDS - Sex: Both - Age: Under 5 (Number)  \
0                                                      10
1                                                      12
2                                                      13
3                                                      16
4                                                      19
…                                                   …
6835                                                 2178
6836                                                 1827
6837                                                 1658
6838                                                 1458
6839                                                 1394

      Deaths - Drowning - Sex: Both - Age: Under 5 (Number)  \
0                                                     776
1                                                     748
2                                                     777
3                                                     872
4                                                     961
…                                                   …
6835                                                  126
6836                                                  133
6837                                                  133
6838                                                  129
6839                                                  127

      Deaths - Malaria - Sex: Both - Age: Under 5 (Number)  \
0                                                      21
1                                                      41
```

```
2                                                          51
3                                                          24
4                                                          52
…                                                         …
6835                                                     1475
6836                                                     1219
6837                                                     1249
6838                                                     1213
6839                                                     1207

      Deaths - Syphilis - Sex: Both - Age: Under 5 (Number)
0                                                       123
1                                                       132
2                                                       180
3                                                       239
4                                                       259
…                                                    …
6835                                                    399
6836                                                    398
6837                                                    394
6838                                                    397
6839                                                    413

[6120 rows x 32 columns]
```

[10]:

```
rename_dict = {
    'Deaths - Invasive Non-typhoidal Salmonella (iNTS) - Sex: Both - Age: Under␣
↪5 (Number)': 'INTS_Deaths',
    'Deaths - Interpersonal violence - Sex: Both - Age: Under 5 (Number)':␣
↪'Violence_Deaths',
    'Deaths - Nutritional deficiencies - Sex: Both - Age: Under 5 (Number)':␣
↪'Nutrition_Deaths',
    'Deaths - Acute hepatitis - Sex: Both - Age: Under 5 (Number)':␣
↪'Hepatitis_Deaths',
    'Deaths - Neoplasms - Sex: Both - Age: Under 5 (Number)':␣
↪'Neoplasms_Deaths',
    'Deaths - Measles - Sex: Both - Age: Under 5 (Number)': 'Measles_Deaths',
    'Deaths - Digestive diseases - Sex: Both - Age: Under 5 (Number)':␣
↪'Digestive_Deaths',
    'Deaths - Cirrhosis and other chronic liver diseases - Sex: Both - Age:␣
↪Under 5 (Number)': 'Cirrhosis_Deaths',
    'Deaths - Chronic kidney disease - Sex: Both - Age: Under 5 (Number)':␣
↪'Kidney_Deaths',
    'Deaths - Cardiovascular diseases - Sex: Both - Age: Under 5 (Number)':␣
↪'Cardiovascular_Deaths',
```

```
    'Deaths - Congenital birth defects - Sex: Both - Age: Under 5 (Number)':␣
 ↪'Congenital_Deaths',
    'Deaths - Lower respiratory infections - Sex: Both - Age: Under 5 (Number)':
 ↪ 'Respiratory_Deaths',
    'Deaths - Neonatal preterm birth - Sex: Both - Age: Under 5 (Number)':␣
 ↪'Preterm_Deaths',
    'Deaths - Environmental heat and cold exposure - Sex: Both - Age: Under 5␣
 ↪(Number)': 'Heat_Cold_Deaths',
    'Deaths - Neonatal sepsis and other neonatal infections - Sex: Both - Age:␣
 ↪Under 5 (Number)': 'Sepsis_Deaths',
    'Deaths - Exposure to forces of nature - Sex: Both - Age: Under 5 (Number)':
 ↪ 'Nature_Deaths',
    'Deaths - Diabetes mellitus - Sex: Both - Age: Under 5 (Number)':␣
 ↪'Diabetes_Deaths',
    'Deaths - Neonatal encephalopathy due to birth asphyxia and trauma - Sex:␣
 ↪Both - Age: Under 5 (Number)': 'Encephalopathy_Deaths',
    'Deaths - Meningitis - Sex: Both - Age: Under 5 (Number)':␣
 ↪'Meningitis_Deaths',
    'Deaths - Other neonatal disorders - Sex: Both - Age: Under 5 (Number)':␣
 ↪'Other_Neonatal_Deaths',
    'Deaths - Whooping cough - Sex: Both - Age: Under 5 (Number)':␣
 ↪'Whooping_Cough_Deaths',
    'Deaths - Diarrheal diseases - Sex: Both - Age: Under 5 (Number)':␣
 ↪'Diarrheal_Deaths',
    'Deaths - Fire, heat, and hot substances - Sex: Both - Age: Under 5␣
 ↪(Number)': 'Fire_Heat_Deaths',
    'Deaths - Road injuries - Sex: Both - Age: Under 5 (Number)': 'Road_Deaths',
    'Deaths - Tuberculosis - Sex: Both - Age: Under 5 (Number)':␣
 ↪'Tuberculosis_Deaths',
    'Deaths - HIV/AIDS - Sex: Both - Age: Under 5 (Number)': 'HIV_AIDS_Deaths',
    'Deaths - Drowning - Sex: Both - Age: Under 5 (Number)': 'Drowning_Deaths',
    'Deaths - Malaria - Sex: Both - Age: Under 5 (Number)': 'Malaria_Deaths',
    'Deaths - Syphilis - Sex: Both - Age: Under 5 (Number)': 'Syphilis_Deaths'
}

df = df.rename(columns=rename_dict)
print(df.columns)
```

```
Index(['Entity', 'Code', 'Year', 'INTS_Deaths', 'Violence_Deaths',
       'Nutrition_Deaths', 'Hepatitis_Deaths', 'Neoplasms_Deaths',
       'Measles_Deaths', 'Digestive_Deaths', 'Cirrhosis_Deaths',
       'Kidney_Deaths', 'Cardiovascular_Deaths', 'Congenital_Deaths',
       'Respiratory_Deaths', 'Preterm_Deaths', 'Heat_Cold_Deaths',
       'Sepsis_Deaths', 'Nature_Deaths', 'Diabetes_Deaths',
       'Encephalopathy_Deaths', 'Meningitis_Deaths', 'Other_Neonatal_Deaths',
       'Whooping_Cough_Deaths', 'Diarrheal_Deaths', 'Fire_Heat_Deaths',
```

```
       'Road_Deaths', 'Tuberculosis_Deaths', 'HIV_AIDS_Deaths',
       'Drowning_Deaths', 'Malaria_Deaths', 'Syphilis_Deaths'],
      dtype='object')
```

[11]:
```
cols = df.drop(columns=['Entity', 'Code', 'Year']).select_dtypes(include=np.
 ↪number)
z_scores = cols.apply(zscore)
outlier_threshold = 3
outliers = (z_scores > outlier_threshold) | (z_scores < -outlier_threshold)
outlier_counts = outliers.sum()
print("Columns with outlier counts:")
print(outlier_counts[outlier_counts > 0])
```

```
Columns with outlier counts:
INTS_Deaths               38
Violence_Deaths           78
Nutrition_Deaths          55
Hepatitis_Deaths          33
Neoplasms_Deaths          70
Measles_Deaths            74
Digestive_Deaths         102
Cirrhosis_Deaths          78
Kidney_Deaths            141
Cardiovascular_Deaths     96
Congenital_Deaths         85
Respiratory_Deaths        73
Preterm_Deaths            58
Heat_Cold_Deaths          43
Sepsis_Deaths             75
Nature_Deaths             13
Diabetes_Deaths          177
Encephalopathy_Deaths    106
Meningitis_Deaths         97
Other_Neonatal_Deaths     48
Whooping_Cough_Deaths     83
Diarrheal_Deaths          57
Fire_Heat_Deaths          81
Road_Deaths              133
Tuberculosis_Deaths      125
HIV_AIDS_Deaths          165
Drowning_Deaths           68
Malaria_Deaths            82
Syphilis_Deaths          136
dtype: int64
```

[12]:
```
filtered_data = df[(df['Year'] >= 1990) & (df['Year'] <= 2019)]
aggregated_data = filtered_data.groupby('Entity').sum()
```

```python
aggregated_data = aggregated_data.drop(columns=['Year'])
aggregated_data
```

```
C:\Users\tiles\AppData\Local\Temp\ipykernel_24540\2972503105.py:2:
FutureWarning: The default value of numeric_only in DataFrameGroupBy.sum is
deprecated. In a future version, numeric_only will default to False. Either
specify numeric_only or select only columns which should be valid for the
function.
  aggregated_data = filtered_data.groupby('Entity').sum()
```

[12]:

```
                INTS_Deaths  Violence_Deaths  Nutrition_Deaths  \
Entity
Afghanistan            4355             6307             58382
Albania                   0               85               457
Algeria                 422              392              3812
American Samoa            0                0                17
Andorra                   0                0                 0
…                       …              …                 …
Venezuela                 0             2990             12437
Vietnam                 928             1055              9295
Yemen                  1924             1201             52547
Zambia                 1503             2487             68820
Zimbabwe               2005              771             33936

                Hepatitis_Deaths  Neoplasms_Deaths  Measles_Deaths  \
Entity
Afghanistan                23437             17974          199342
Albania                       13               798             196
Algeria                     1110              4393           20391
American Samoa                 0                 0              33
Andorra                        0                 0               0
…                            …               …              …
Venezuela                    129              4801              62
Vietnam                      667             11434           84565
Yemen                       6388              8405          100086
Zambia                      1015             13625           57746
Zimbabwe                     334              1183           28844

                Digestive_Deaths  Cirrhosis_Deaths  Kidney_Deaths  \
Entity
Afghanistan                20764              7046           8904
Albania                     1357               127            143
Algeria                     5265              1884           2848
American Samoa                 0                 0              0
Andorra                        0                 0              0
…                            …               …             …
Venezuela                   3363               439            716
```

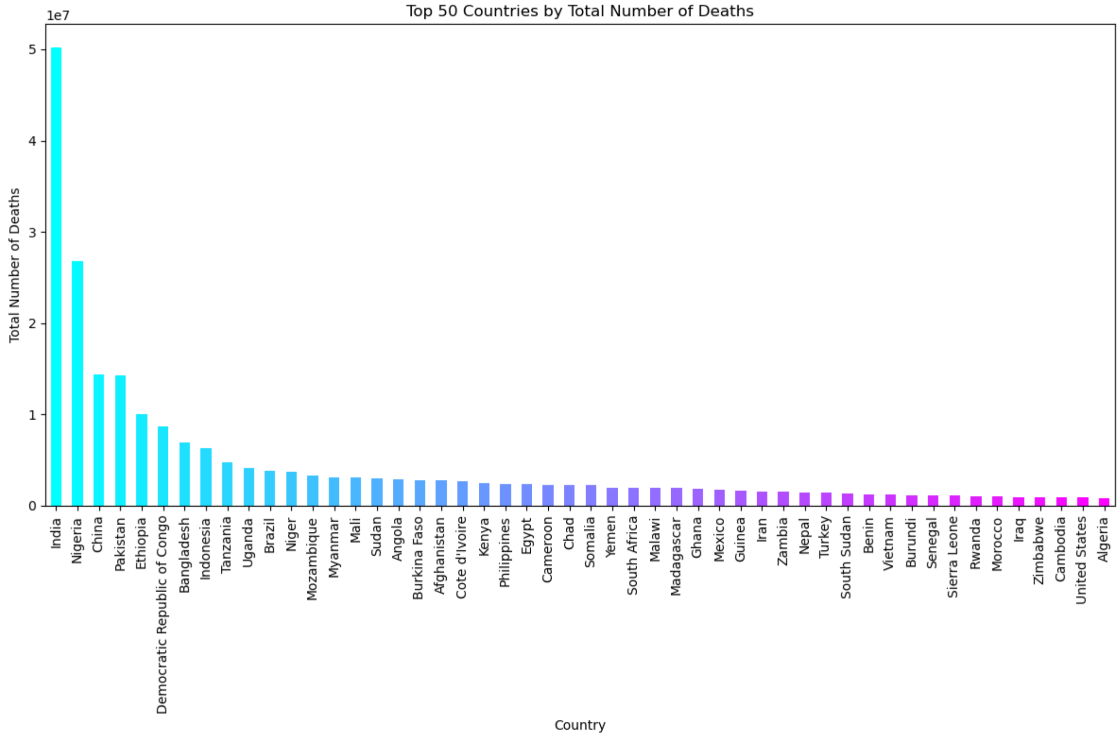
```
Vietnam                      10139               1777            2849
Yemen                        10464               3756            3108
Zambia                        8670               1752            2269
Zimbabwe                      2051                194             531

                Cardiovascular_Deaths  …  Other_Neonatal_Deaths  \
Entity                                 …
Afghanistan                      8038  …                 322226
Albania                          1073  …                   7584
Algeria                         10669  …                  78084
American Samoa                      0  …                     46
Andorra                             0  …                      0
…                                 …   …                      …
Venezuela                        1058  …                   9828
Vietnam                         12965  …                  27792
Yemen                           19717  …                 252675
Zambia                           5779  …                  59476
Zimbabwe                         3908  …                  62915

                Whooping_Cough_Deaths  Diarrheal_Deaths  Fire_Heat_Deaths  \
Entity
Afghanistan                    107240            236890              4528
Albania                           258               505               142
Algeria                          8150             26856              3659
American Samoa                     22                 4                 0
Andorra                             0                 0                 0
…                                 …                 …                 …
Venezuela                         686             38038               756
Vietnam                         33658             38757              3021
Yemen                           32201            397881              7836
Zambia                          27087            230238              4001
Zimbabwe                        12635             47178              2267

                Road_Deaths  Tuberculosis_Deaths  HIV_AIDS_Deaths  \
Entity
Afghanistan           26172                23411             1185
Albania                 324                   29                0
Algeria               26803                  959             1158
American Samoa            0                    0                0
Andorra                   0                    0                0
…                       …                    …                …
Venezuela              5376                  755             1534
Vietnam                9956                10835             2072
Yemen                 54648                 5338             1759
Zambia                 8373                25753           216174
Zimbabwe               2200                17075           236368
```

```
                Drowning_Deaths  Malaria_Deaths  Syphilis_Deaths
Entity
Afghanistan               25157            3310             8450
Albania                     426               0              150
Algeria                    5173               0             4200
American Samoa                0               0               55
Andorra                       0               0                0
…                           …               …                 …
Venezuela                  3369             523              432
Vietnam                   65895            1645             9328
Yemen                     12282            7081            10606
Zambia                     6805          124161            38809
Zimbabwe                   2297           56942            11729

[204 rows x 29 columns]
```

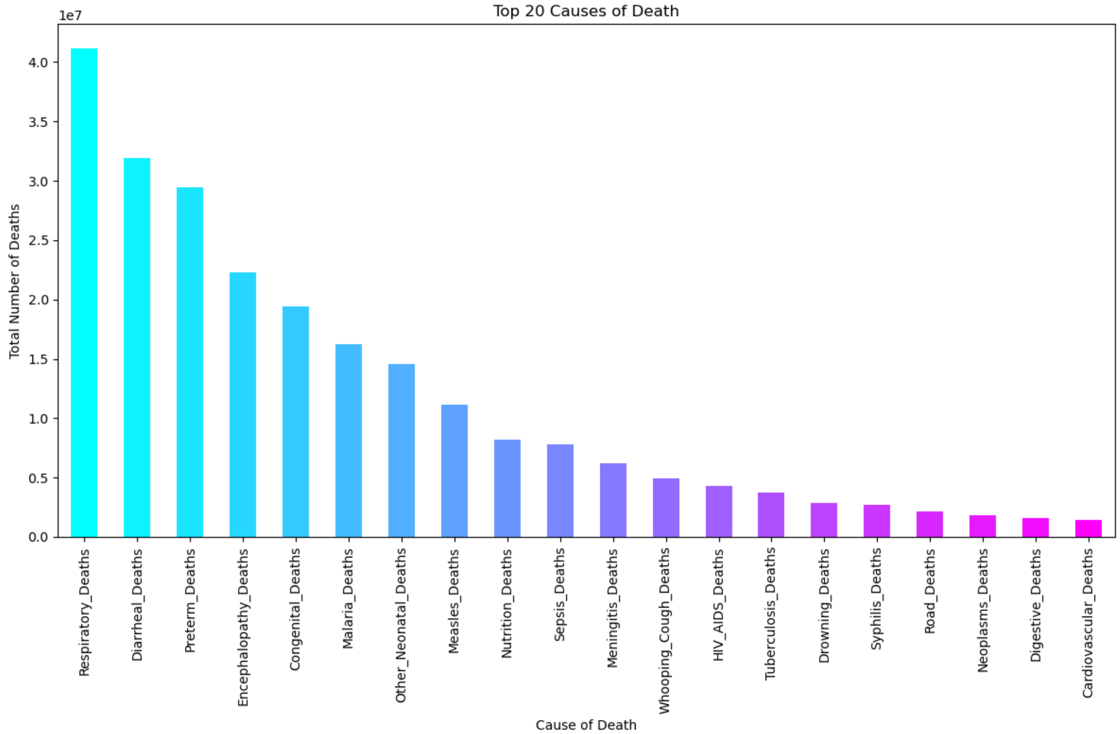
[13]:

```python
total_deaths_per_entity = df.groupby('Entity').sum().sum(axis=1)
top_50_countries = total_deaths_per_entity.sort_values(ascending=False).head(50)

colors = plt.cm.cool(np.linspace(0, 1, len(top_50_countries)))

plt.figure(figsize=(12, 8))
top_50_countries.plot(kind='bar', color=colors)
plt.title('Top 50 Countries by Total Number of Deaths')
plt.xlabel('Country')
plt.ylabel('Total Number of Deaths')
plt.xticks(rotation=90)
plt.tight_layout()
plt.show()
```

```
C:\Users\tiles\AppData\Local\Temp\ipykernel_24540\11496973.py:1: FutureWarning:
The default value of numeric_only in DataFrameGroupBy.sum is deprecated. In a
future version, numeric_only will default to False. Either specify numeric_only
or select only columns which should be valid for the function.
  total_deaths_per_entity = df.groupby('Entity').sum().sum(axis=1)
```



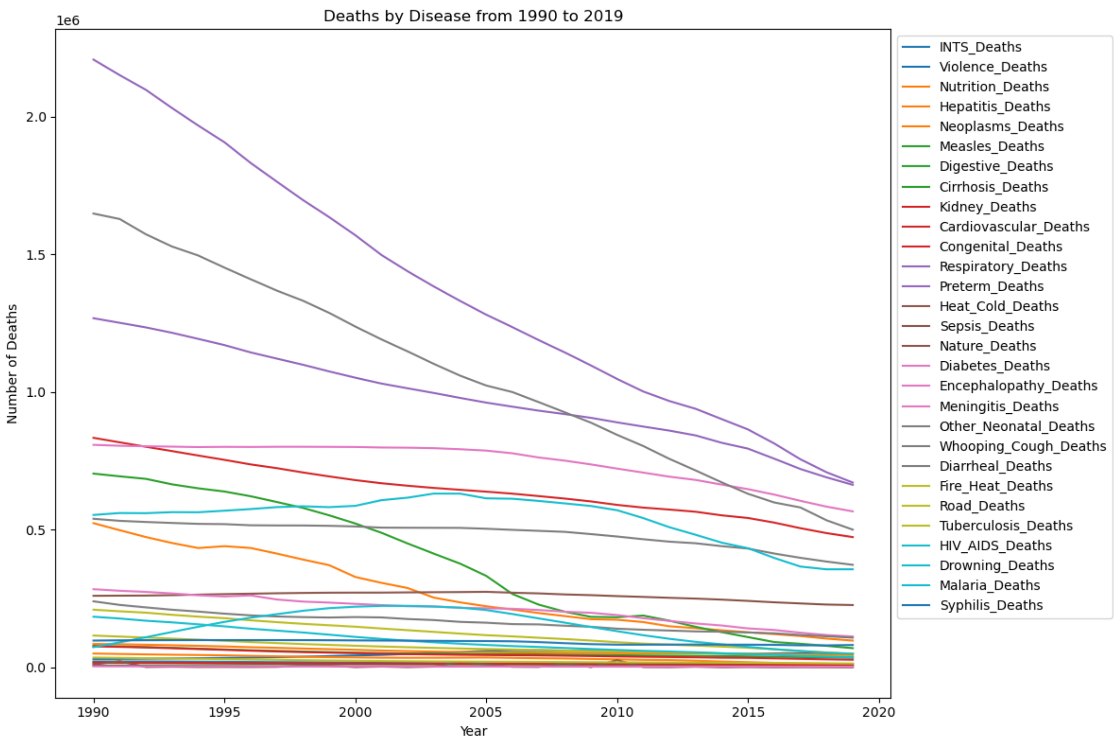
[14]:
```
causes_of_death = df.drop(columns=['Entity', 'Code', 'Year']).sum()
top_20_causes = causes_of_death.sort_values(ascending=False).head(20)

colors = plt.cm.cool(np.linspace(0, 1, len(top_20_causes)))

plt.figure(figsize=(12, 8))
top_20_causes.plot(kind='bar', color=colors)
plt.title('Top 20 Causes of Death')
plt.xlabel('Cause of Death')
plt.ylabel('Total Number of Deaths')
plt.xticks(rotation=90)
plt.tight_layout()
plt.show()
```



[15]:
```
df_map = df.copy()

df_map['Total_Deaths'] = df_map.drop(columns=['Entity', 'Code', 'Year']).
    ↪sum(axis=1)
yearly_deaths = df_map.groupby(['Entity', 'Year'])['Total_Deaths'].sum().
    ↪reset_index()

fig = px.choropleth(yearly_deaths,
                    locations="Entity",
                    locationmode='country names',
                    color="Total_Deaths",
                    hover_name="Entity",
                    animation_frame="Year",
                    color_continuous_scale='Reds',
                    title="Total Deaths from 1990 to 2019")

fig.update_layout(
    geo=dict(showframe=False, showcoastlines=False,␣
    ↪projection_type='equirectangular'),
    title=dict(x=0.5)
)

fig.show()
```

[16]:

```
import matplotlib.pyplot as plt

df_cause = df.drop(columns=['Entity', 'Code'])
df_yearly = df_cause.groupby('Year').sum().reset_index()

if 'Total_Deaths' in df_yearly.columns:
    df_yearly = df_yearly.drop(columns='Total_Deaths')

num_causes = len(df_yearly.columns) - 1
colors = plt.cm.tab10(np.linspace(0, 1, num_causes))

plt.figure(figsize=(12, 8))
for i, c in enumerate(df_yearly.columns[1:], start=1):
    plt.plot(df_yearly['Year'], df_yearly[c], label=c, color=colors[i %␣
↪num_causes])

plt.title('Deaths by Disease from 1990 to 2019')
plt.xlabel('Year')
plt.ylabel('Number of Deaths')
plt.legend(loc='upper left', bbox_to_anchor=(1, 1), ncol=1)
plt.tight_layout()
plt.show()
```



Models

[17]:
```
scaler = StandardScaler()
df_scaled = scaler.fit_transform(df.drop(columns=['Entity', 'Code', 'Year']))
pca = PCA()
pca_out = pca.fit_transform(df_scaled)
```

[18]:
```
print(pd.DataFrame({'Center': scaler.mean_, 'Scale': scaler.scale_}, index=df.
 ↪columns.drop(['Entity', 'Code', 'Year'])))
```

```
                           Center         Scale
INTS_Deaths            231.492320   1622.368214
Violence_Deaths         79.825817    303.457690
Nutrition_Deaths      1333.719118   6972.497881
Hepatitis_Deaths       160.559967   1616.560333
Neoplasms_Deaths       297.492974   1204.333079
Measles_Deaths        1814.687092   8421.696247
Digestive_Deaths       253.901797    938.604887
Cirrhosis_Deaths        54.085621    238.815144
Kidney_Deaths           63.699020    180.199557
Cardiovascular_Deaths  236.582190    898.611133
Congenital_Deaths     3175.634477  10591.726971
Respiratory_Deaths    6722.154902  30161.351947
Preterm_Deaths        4808.332026  20841.947352
Heat_Cold_Deaths        17.039706    123.145252
Sepsis_Deaths         1271.867157   4650.791920
Nature_Deaths           24.220098    505.365350
Diabetes_Deaths         16.040850     44.805750
Encephalopathy_Deaths 3647.752614  15066.581181
Meningitis_Deaths     1019.708007   4083.534715
Other_Neonatal_Deaths 2374.598366  14282.461832
Whooping_Cough_Deaths  807.175163   3630.187080
Diarrheal_Deaths      5213.969771  22619.833294
Fire_Heat_Deaths       111.063399    358.777053
Road_Deaths            352.145588   1244.571219
Tuberculosis_Deaths    611.118464   2500.691503
HIV_AIDS_Deaths        707.966013   2634.357974
Drowning_Deaths        469.680556   2781.121195
Malaria_Deaths        2654.809150  12169.034911
Syphilis_Deaths        448.185294   1475.692709
```

[19]:
```
print("Number of Principal Components:", pca.n_components_)
```

```
Number of Principal Components: 29
```

[20]:
```
components_df = pd.DataFrame(pca.components_.T, index=df.columns.
 ↪drop(['Entity', 'Code', 'Year']), columns=[f'PC{i+1}' for i in range(pca.
 ↪n_components_)])
```

```python
print(components_df)
```

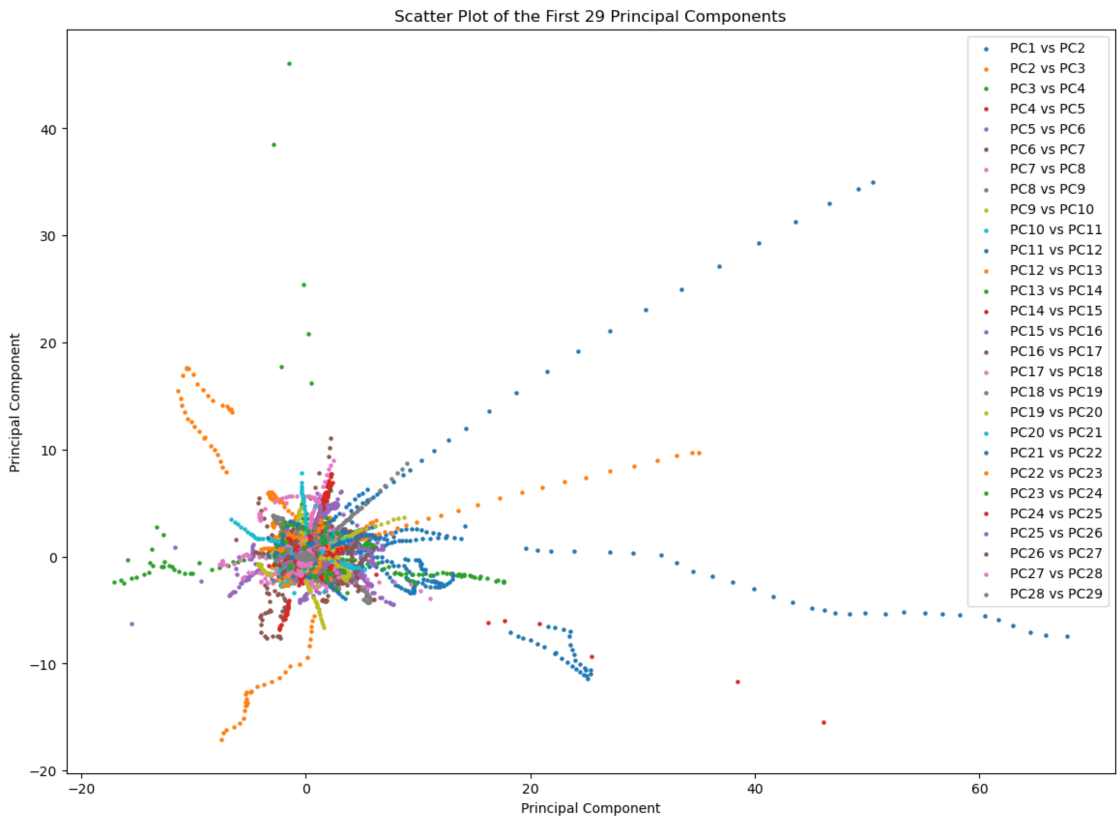
```
                           PC1       PC2       PC3       PC4       PC5  \
INTS_Deaths           0.094227 -0.279744  0.463189 -0.097988 -0.311449
Violence_Deaths       0.192083  0.222668  0.146339 -0.055789 -0.035394
Nutrition_Deaths      0.185846 -0.098552 -0.229165  0.030813  0.090948
Hepatitis_Deaths      0.181262 -0.071792 -0.355795 -0.059059 -0.015808
Neoplasms_Deaths      0.189168  0.308176  0.059227 -0.001661  0.086901
Measles_Deaths        0.190922 -0.195315 -0.058128  0.001188 -0.004194
Digestive_Deaths      0.204316  0.154755  0.155294 -0.032092 -0.091682
Cirrhosis_Deaths      0.207761 -0.142255 -0.120225 -0.020686 -0.116311
Kidney_Deaths         0.203901  0.060996  0.144088  0.068227 -0.049469
Cardiovascular_Deaths 0.153002  0.310854  0.170639 -0.037460 -0.036617
Congenital_Deaths     0.210769  0.182265  0.024624 -0.018932 -0.016313
Respiratory_Deaths    0.221859  0.003205 -0.048193 -0.041515 -0.054279
Preterm_Deaths        0.212775 -0.004188 -0.175389 -0.018307 -0.026728
Heat_Cold_Deaths      0.201447  0.150567 -0.138287 -0.088340 -0.017715
Sepsis_Deaths         0.204937 -0.181966 -0.100790  0.021038  0.033891
Nature_Deaths         0.021530  0.025108 -0.033360  0.933804 -0.320722
Diabetes_Deaths       0.151425  0.185858  0.211426  0.150184  0.291552
Encephalopathy_Deaths 0.205653 -0.058167 -0.058967  0.010093  0.001525
Meningitis_Deaths     0.200081 -0.191567  0.151517 -0.053601 -0.146360
Other_Neonatal_Deaths 0.186166 -0.095823 -0.326120 -0.005729  0.037272
Whooping_Cough_Deaths 0.216636 -0.095771 -0.121720 -0.004722  0.002849
Diarrheal_Deaths      0.204768 -0.223387  0.033048 -0.042289 -0.123416
Fire_Heat_Deaths      0.218313  0.060347  0.051362 -0.028369 -0.029486
Road_Deaths           0.182034  0.251374  0.167682 -0.015125 -0.015366
Tuberculosis_Deaths   0.205832 -0.118289 -0.014162  0.037863  0.044409
HIV_AIDS_Deaths       0.067126 -0.195497  0.223282  0.218046  0.743423
Drowning_Deaths       0.178385  0.319090 -0.047663  0.013700  0.006005
Malaria_Deaths        0.129861 -0.318591  0.386044 -0.031780 -0.070264
Syphilis_Deaths       0.193933 -0.157282 -0.001519  0.118707  0.260574

                           PC6       PC7       PC8       PC9      PC10  ...  \
INTS_Deaths          -0.150491 -0.160460 -0.168650  0.019196  0.068765  ...
Violence_Deaths      -0.211198 -0.073727 -0.162915 -0.251523  0.142197  ...
Nutrition_Deaths      0.162417  0.419810 -0.199878 -0.001383  0.167245  ...
Hepatitis_Deaths     -0.181882 -0.073030 -0.109682  0.225578  0.130812  ...
Neoplasms_Deaths     -0.085367  0.004181  0.014093 -0.241094 -0.110874  ...
Measles_Deaths        0.131766  0.354851  0.323575 -0.326737  0.191139  ...
Digestive_Deaths     -0.102466 -0.013309 -0.087361 -0.031306 -0.249631  ...
Cirrhosis_Deaths     -0.004083 -0.084571 -0.035769  0.175154 -0.093034  ...
Kidney_Deaths         0.238568 -0.130608  0.174932 -0.108956 -0.115535  ...
Cardiovascular_Deaths -0.069920  0.144762  0.567575  0.467288  0.356480  ...
Congenital_Deaths    -0.090481 -0.163793 -0.011526  0.023456 -0.043899  ...
Respiratory_Deaths   -0.076192  0.086761 -0.050587 -0.051147  0.123885  ...
Preterm_Deaths       -0.084886 -0.221464  0.010628  0.092853 -0.044653  ...
```

```
Heat_Cold_Deaths      -0.275361  0.125037 -0.239183  0.012007  0.136205  …
Sepsis_Deaths          0.034137 -0.287823 -0.004858  0.011011  0.111145  …
Nature_Deaths         -0.124072  0.033323 -0.017946  0.017272  0.030310  …
Diabetes_Deaths        0.524986 -0.306077 -0.212995  0.026004  0.468122  …
Encephalopathy_Deaths  0.027057 -0.265309  0.344594 -0.302319 -0.190070  …
Meningitis_Deaths     -0.045498 -0.004897  0.197662 -0.228545  0.005492  …
Other_Neonatal_Deaths -0.070486 -0.265279  0.060426  0.175507  0.013000  …
Whooping_Cough_Deaths -0.000840  0.023613 -0.063161  0.040207  0.011448  …
Diarrheal_Deaths      -0.003502  0.130329  0.078196 -0.001123  0.121607  …
Fire_Heat_Deaths      -0.076153  0.056770 -0.007674 -0.077277  0.032710  …
Road_Deaths            0.066890  0.188621  0.020785  0.382679 -0.417345  …
Tuberculosis_Deaths    0.300646  0.298881 -0.041852  0.027531 -0.235244  …
HIV_AIDS_Deaths       -0.472547  0.074565  0.070075 -0.014049 -0.019501  …
Drowning_Deaths        0.021078  0.156610 -0.273636 -0.205253 -0.059690  …
Malaria_Deaths         0.005307  0.110629 -0.258285  0.230162  0.052138  …
Syphilis_Deaths        0.234327 -0.102497 -0.021082  0.145453 -0.333198  …

                           PC20      PC21      PC22      PC23      PC24  \
INTS_Deaths            0.097433  0.001448 -0.074810 -0.094545 -0.097994
Violence_Deaths       -0.173820 -0.033710 -0.050523  0.083480  0.103140
Nutrition_Deaths      -0.127458 -0.044478  0.009286  0.094013 -0.186105
Hepatitis_Deaths      -0.236619 -0.216256 -0.077713  0.303019  0.177520
Neoplasms_Deaths      -0.092602  0.080114 -0.037015 -0.206970  0.142890
Measles_Deaths         0.053259 -0.115713 -0.123097 -0.015506 -0.142276
Digestive_Deaths      -0.055170 -0.379395  0.051284 -0.150414 -0.219953
Cirrhosis_Deaths      -0.092435 -0.357556 -0.195848 -0.003489  0.239556
Kidney_Deaths         -0.321186  0.343685 -0.127534 -0.012511 -0.199787
Cardiovascular_Deaths  0.055162 -0.101249  0.098490 -0.050023  0.073617
Congenital_Deaths     -0.166603  0.104662  0.302517 -0.228667  0.095590
Respiratory_Deaths    -0.110788  0.086398 -0.053321 -0.144675 -0.227351
Preterm_Deaths         0.108676 -0.137703  0.254089 -0.071234 -0.148213
Heat_Cold_Deaths       0.035002  0.207158 -0.148331 -0.029819 -0.384965
Sepsis_Deaths          0.647155  0.229967 -0.002104  0.026460 -0.036458
Nature_Deaths          0.001507 -0.001788  0.001029  0.002632 -0.003212
Diabetes_Deaths       -0.042636 -0.117215 -0.058150  0.066593 -0.010476
Encephalopathy_Deaths  0.009330 -0.222256 -0.415252  0.056801  0.066756
Meningitis_Deaths     -0.115091  0.092793  0.559466  0.541645 -0.002258
Other_Neonatal_Deaths -0.085579  0.118892  0.098032 -0.148898 -0.244146
Whooping_Cough_Deaths -0.121116  0.456805 -0.059503 -0.087762  0.564668
Diarrheal_Deaths       0.047733  0.118182 -0.224383 -0.040080  0.085454
Fire_Heat_Deaths       0.321166 -0.138664  0.057472 -0.110696  0.131770
Road_Deaths            0.150643  0.216489 -0.267263  0.347573 -0.100635
Tuberculosis_Deaths    0.081210 -0.069895  0.275319 -0.421664  0.135779
HIV_AIDS_Deaths        0.004064 -0.001969 -0.019463  0.000040  0.015408
Drowning_Deaths        0.311309 -0.091829  0.079973  0.269858  0.182744
Malaria_Deaths        -0.119786 -0.071422 -0.039554 -0.015174  0.069500
Syphilis_Deaths       -0.061982 -0.062689  0.100664  0.136883 -0.121650
```

	PC25	PC26	PC27	PC28	PC29
INTS_Deaths	-0.088373	0.044640	-0.002440	-0.108274	-0.045844
Violence_Deaths	0.057892	0.050430	0.056767	0.021887	0.028849
Nutrition_Deaths	0.060651	-0.031770	-0.038811	-0.111907	-0.029652
Hepatitis_Deaths	-0.400324	-0.138143	-0.279816	-0.366895	0.054902
Neoplasms_Deaths	-0.388595	0.351079	0.316007	-0.234096	0.146334
Measles_Deaths	0.086526	0.055086	0.052071	0.008244	-0.066508
Digestive_Deaths	0.307334	0.024007	-0.262471	-0.292995	-0.095235
Cirrhosis_Deaths	-0.147981	0.235191	0.326707	0.486375	-0.074721
Kidney_Deaths	-0.228333	-0.123653	0.026343	-0.201089	-0.040781
Cardiovascular_Deaths	-0.002784	-0.016565	0.019231	0.005018	-0.027578
Congenital_Deaths	-0.059441	-0.270984	-0.333200	0.332474	-0.250917
Respiratory_Deaths	0.009130	0.014471	-0.242095	0.348853	0.754397
Preterm_Deaths	0.220018	-0.408043	0.585638	-0.214154	0.257328
Heat_Cold_Deaths	-0.105954	-0.203076	0.198300	0.237287	-0.404382
Sepsis_Deaths	-0.236459	-0.000124	-0.138853	-0.011869	0.045211
Nature_Deaths	-0.004240	0.002690	-0.001426	0.004114	-0.000293
Diabetes_Deaths	0.077620	0.030078	0.022936	0.017752	-0.040112
Encephalopathy_Deaths	0.131748	-0.218342	-0.136616	0.111867	-0.068003
Meningitis_Deaths	-0.029798	0.122174	0.048276	0.108004	-0.035973
Other_Neonatal_Deaths	0.265599	0.635336	-0.080919	-0.078857	-0.164563
Whooping_Cough_Deaths	0.440254	-0.084923	0.022981	-0.074209	0.032438
Diarrheal_Deaths	0.022970	-0.035097	0.047334	-0.171974	-0.151255
Fire_Heat_Deaths	-0.004262	0.065565	-0.176340	-0.107244	0.071756
Road_Deaths	0.066116	0.072285	-0.019305	0.017310	0.088178
Tuberculosis_Deaths	-0.252701	-0.057748	-0.033566	-0.002233	-0.090820
HIV_AIDS_Deaths	-0.001889	-0.011943	-0.000415	0.002573	-0.014906
Drowning_Deaths	0.118622	0.073175	-0.012029	0.047145	-0.063032
Malaria_Deaths	0.074032	0.002485	0.052472	0.004714	0.040699
Syphilis_Deaths	-0.041984	-0.047687	0.012801	0.092254	0.028923

[29 rows x 29 columns]

[24]:
```
plt.figure(figsize=(14, 10))

for i in range(28):
    plt.scatter(pca_out[:, i], pca_out[:, i + 1], label=f'PC{i+1} vs PC{i+2}',
 ↪s=5)

plt.xlabel('Principal Component')
plt.ylabel('Principal Component')
plt.title('Scatter Plot of the First 29 Principal Components')
plt.legend()
plt.show()
```



[25]:
```
print("Explained Variance:", pca.explained_variance_)
```

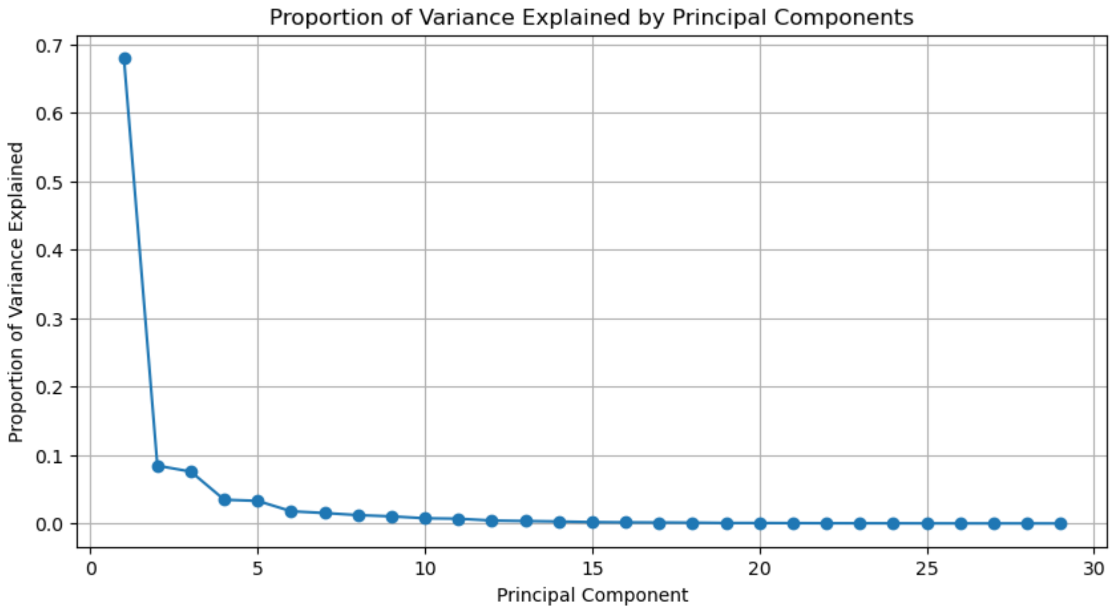
```
Explained Variance: [1.97196643e+01 2.45048380e+00 2.19999567e+00 1.00752567e+00
 9.58763219e-01 5.18799377e-01 4.43362442e-01 3.55857051e-01
 3.02347592e-01 2.21220670e-01 2.04948758e-01 1.25738712e-01
 1.04572661e-01 8.55214733e-02 5.88939451e-02 4.80105053e-02
 4.28070082e-02 3.57672799e-02 2.20169171e-02 2.13706380e-02
 1.66299190e-02 1.26423574e-02 1.15981840e-02 9.93009277e-03
 7.62041773e-03 5.67412152e-03 5.28917665e-03 4.89361606e-03
 2.79377675e-03]
```

[26]:
```
print("Explained Variance Ratio:", pca.explained_variance_ratio_)
```

```
Explained Variance Ratio: [6.79877314e-01 8.44856343e-02 7.58495239e-02
3.47365877e-02
 3.30553985e-02 1.78867106e-02 1.52858620e-02 1.22689277e-02
 1.04240755e-02 7.62705251e-03 7.06604378e-03 4.33510917e-03
 3.60536462e-03 2.94853446e-03 2.03049386e-03 1.65526415e-03
 1.47586254e-03 1.23315295e-03 7.59079984e-04 7.36798139e-04
 5.73351781e-04 4.35872127e-04 3.99872030e-04 3.42361042e-04
 2.62730088e-04 1.95627392e-04 1.82355600e-04 1.68717809e-04
 9.63213881e-05]
```

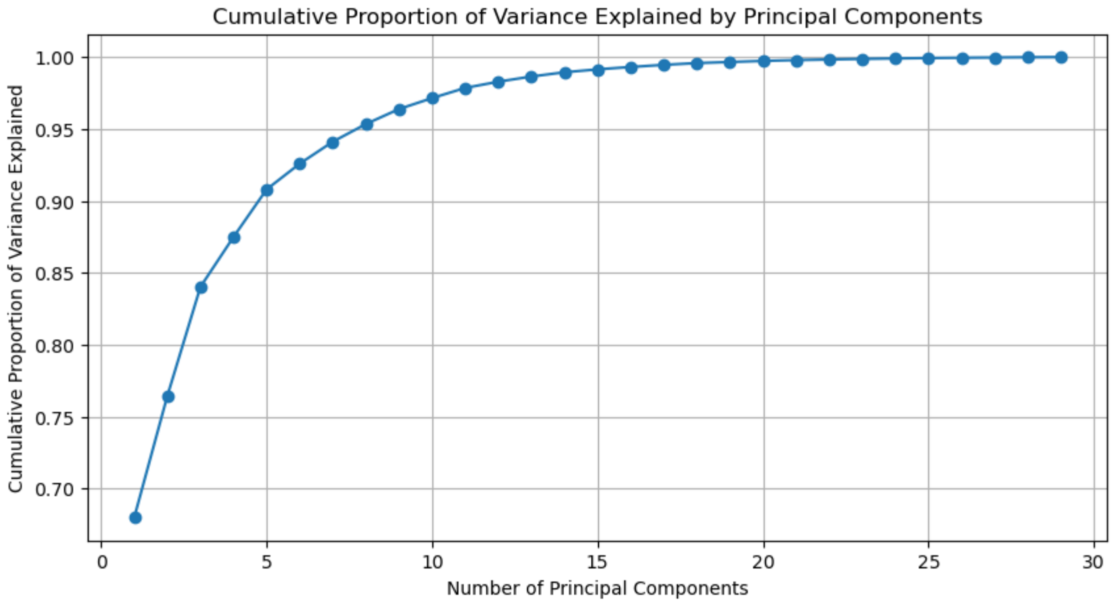
[27]:

```
plt.figure(figsize=(10, 5))
plt.plot(range(1, len(pca.explained_variance_ratio_) + 1), pca.
    ↪explained_variance_ratio_, marker='o')
plt.xlabel('Principal Component')
plt.ylabel('Proportion of Variance Explained')
plt.title('Proportion of Variance Explained by Principal Components')
plt.grid(True)
plt.show()
```



[28]:

```
cumulative_explained_variance = np.cumsum(pca.explained_variance_ratio_)
plt.figure(figsize=(10, 5))
plt.plot(range(1, len(cumulative_explained_variance) + 1),␣
    ↪cumulative_explained_variance, marker='o')
plt.xlabel('Number of Principal Components')
plt.ylabel('Cumulative Proportion of Variance Explained')
plt.title('Cumulative Proportion of Variance Explained by Principal Components')
plt.grid(True)
plt.show()
```



[29]:

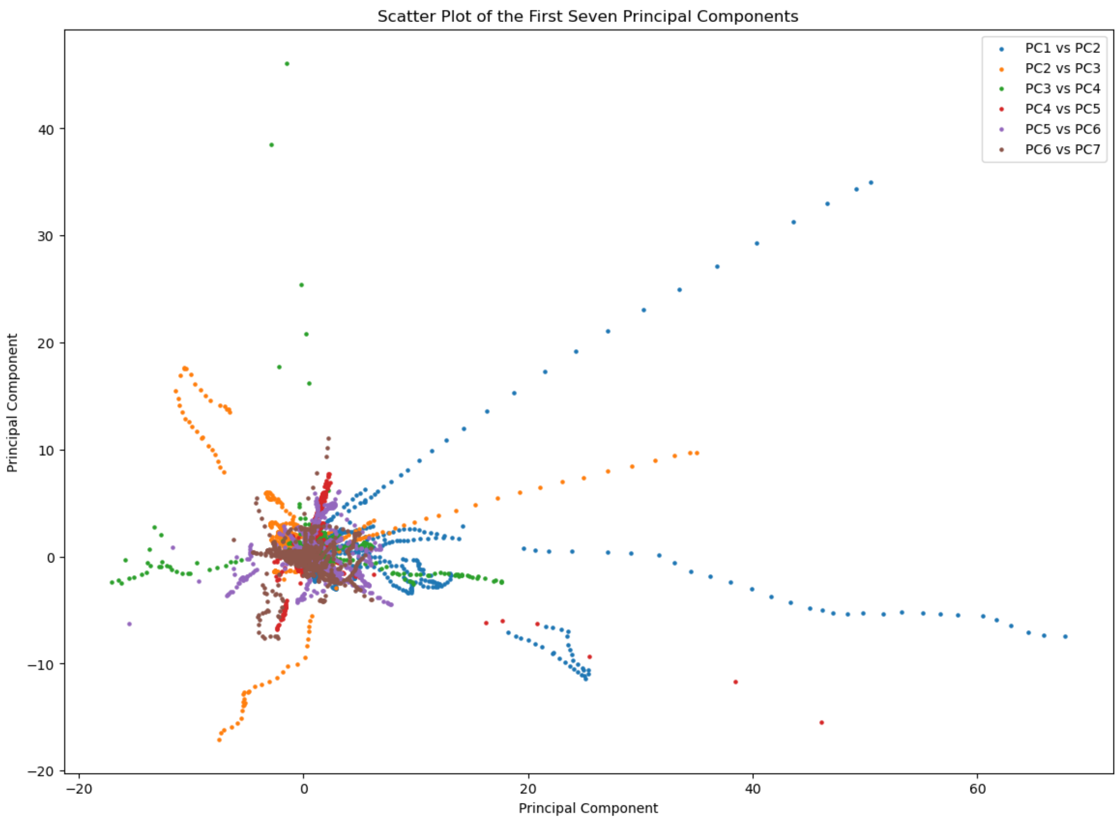
```
pca = PCA(n_components=7)
pca.fit(df_scaled)
components_df = pd.DataFrame(pca.components_.T, index=df.columns.
    ↪drop(['Entity', 'Code', 'Year']), columns=[f'PC{i+1}' for i in range(pca.
    ↪n_components_)])
print(components_df)
pca_out = pca.transform(df_scaled)
```

```
                          PC1       PC2       PC3       PC4       PC5  \
INTS_Deaths          0.094227 -0.279744  0.463189 -0.097988 -0.311449
Violence_Deaths      0.192083  0.222668  0.146339 -0.055789 -0.035394
Nutrition_Deaths     0.185846 -0.098552 -0.229165  0.030813  0.090948
Hepatitis_Deaths     0.181262 -0.071792 -0.355795 -0.059059 -0.015808
Neoplasms_Deaths     0.189168  0.308176  0.059227 -0.001661  0.086901
Measles_Deaths       0.190922 -0.195315 -0.058128  0.001188 -0.004194
Digestive_Deaths     0.204316  0.154755  0.155294 -0.032092 -0.091682
Cirrhosis_Deaths     0.207761 -0.142255 -0.120225 -0.020686 -0.116311
Kidney_Deaths        0.203901  0.060996  0.144088  0.068227 -0.049469
Cardiovascular_Deaths 0.153002  0.310854  0.170639 -0.037460 -0.036617
Congenital_Deaths    0.210769  0.182265  0.024624 -0.018932 -0.016313
Respiratory_Deaths   0.221859  0.003205 -0.048193 -0.041515 -0.054279
Preterm_Deaths       0.212775 -0.004188 -0.175389 -0.018307 -0.026728
Heat_Cold_Deaths     0.201447  0.150567 -0.138287 -0.088340 -0.017715
Sepsis_Deaths        0.204937 -0.181966 -0.100790  0.021038  0.033891
Nature_Deaths        0.021530  0.025108 -0.033360  0.933804 -0.320722
Diabetes_Deaths      0.151425  0.185858  0.211426  0.150184  0.291552
Encephalopathy_Deaths 0.205653 -0.058167 -0.058967  0.010093  0.001525
```

```
Meningitis_Deaths      0.200081 -0.191567  0.151517 -0.053601 -0.146360
Other_Neonatal_Deaths  0.186166 -0.095823 -0.326120 -0.005729  0.037272
Whooping_Cough_Deaths  0.216636 -0.095771 -0.121720 -0.004722  0.002849
Diarrheal_Deaths       0.204768 -0.223387  0.033048 -0.042289 -0.123416
Fire_Heat_Deaths       0.218313  0.060347  0.051362 -0.028369 -0.029486
Road_Deaths            0.182034  0.251374  0.167682 -0.015125 -0.015366
Tuberculosis_Deaths    0.205832 -0.118289 -0.014162  0.037863  0.044409
HIV_AIDS_Deaths        0.067126 -0.195497  0.223282  0.218046  0.743423
Drowning_Deaths        0.178385  0.319090 -0.047663  0.013700  0.006005
Malaria_Deaths         0.129861 -0.318591  0.386044 -0.031780 -0.070264
Syphilis_Deaths        0.193933 -0.157282 -0.001519  0.118707  0.260574

                            PC6       PC7
INTS_Deaths           -0.150491 -0.160460
Violence_Deaths       -0.211198 -0.073727
Nutrition_Deaths       0.162417  0.419810
Hepatitis_Deaths      -0.181882 -0.073030
Neoplasms_Deaths      -0.085367  0.004181
Measles_Deaths         0.131766  0.354851
Digestive_Deaths      -0.102466 -0.013309
Cirrhosis_Deaths      -0.004083 -0.084571
Kidney_Deaths          0.238568 -0.130608
Cardiovascular_Deaths -0.069920  0.144762
Congenital_Deaths     -0.090481 -0.163793
Respiratory_Deaths    -0.076192  0.086761
Preterm_Deaths        -0.084886 -0.221464
Heat_Cold_Deaths      -0.275361  0.125037
Sepsis_Deaths          0.034137 -0.287823
Nature_Deaths         -0.124072  0.033323
Diabetes_Deaths        0.524986 -0.306077
Encephalopathy_Deaths  0.027057 -0.265309
Meningitis_Deaths     -0.045498 -0.004897
Other_Neonatal_Deaths -0.070486 -0.265279
Whooping_Cough_Deaths -0.000840  0.023613
Diarrheal_Deaths      -0.003502  0.130329
Fire_Heat_Deaths      -0.076153  0.056770
Road_Deaths            0.066890  0.188621
Tuberculosis_Deaths    0.300646  0.298881
HIV_AIDS_Deaths       -0.472547  0.074565
Drowning_Deaths        0.021078  0.156610
Malaria_Deaths         0.005307  0.110629
Syphilis_Deaths        0.234327 -0.102497
```

[30]:
```
plt.figure(figsize=(14, 10))

for i in range(6):
```

```
    plt.scatter(pca_out[:, i], pca_out[:, i + 1], label=f'PC{i+1} vs PC{i+2}',␣
↪s=5)

plt.xlabel('Principal Component')
plt.ylabel('Principal Component')
plt.title('Scatter Plot of the First Seven Principal Components')
plt.legend()
plt.show()
```



0.0.1 SVD

[31]:
```
U, s, V = np.linalg.svd(df_scaled, full_matrices=False)
```

[32]:
```
np.round(V.T, 3)
```

[32]:
```
array([[ 0.094,  0.28 , -0.463,  0.098, -0.311,  0.15 , -0.16 ,  0.169,
        -0.019,  0.069, -0.101, -0.211,  0.041,  0.095,  0.002, -0.224,
        -0.349,  0.434,  0.163,  0.097,  0.001,  0.075,  0.095, -0.098,
        -0.088, -0.045, -0.002, -0.108, -0.046],
       [ 0.192, -0.223, -0.146,  0.056, -0.035,  0.211, -0.074,  0.163,
         0.252,  0.142,  0.175,  0.048,  0.298,  0.036, -0.169, -0.378,
```

```
 0.593,  0.092,  0.103, -0.174, -0.034,  0.051, -0.083,  0.103,
 0.058, -0.05 ,  0.057,  0.022,  0.029],
[ 0.186,  0.099,  0.229, -0.031,  0.091, -0.162,  0.42 ,  0.2  ,
 0.001,  0.167, -0.187, -0.241,  0.592, -0.232, -0.066,  0.116,
-0.133,  0.08 ,  0.055, -0.127, -0.044, -0.009, -0.094, -0.186,
 0.061,  0.032, -0.039, -0.112, -0.03 ],
[ 0.181,  0.072,  0.356,  0.059, -0.016,  0.182, -0.073,  0.11 ,
-0.226,  0.131,  0.092,  0.08 , -0.225,  0.034,  0.019, -0.044,
-0.038,  0.085,  0.091, -0.237, -0.216,  0.078, -0.303,  0.178,
-0.4  ,  0.138, -0.28 , -0.367,  0.055],
[ 0.189, -0.308, -0.059,  0.002,  0.087,  0.085,  0.004, -0.014,
 0.241, -0.111,  0.198, -0.09 ,  0.089,  0.04 ,  0.02 ,  0.39 ,
-0.174,  0.079, -0.068, -0.093,  0.08 ,  0.037,  0.207,  0.143,
-0.389, -0.351,  0.316, -0.234,  0.146],
[ 0.191,  0.195,  0.058, -0.001, -0.004, -0.132,  0.355, -0.324,
 0.327,  0.191,  0.114,  0.356, -0.248,  0.188, -0.295,  0.098,
-0.022,  0.231,  0.272,  0.053, -0.116,  0.123,  0.016, -0.142,
 0.087, -0.055,  0.052,  0.008, -0.067],
[ 0.204, -0.155, -0.155,  0.032, -0.092,  0.102, -0.013,  0.087,
 0.031, -0.25 , -0.181,  0.234,  0.07 ,  0.229,  0.338,  0.249,
 0.098, -0.091,  0.028, -0.055, -0.379, -0.051,  0.15 , -0.22 ,
 0.307, -0.024, -0.262, -0.293, -0.095],
[ 0.208,  0.142,  0.12 ,  0.021, -0.116,  0.004, -0.085,  0.036,
-0.175, -0.093, -0.353,  0.221,  0.12 ,  0.015,  0.054, -0.016,
-0.04 , -0.082,  0.05 , -0.092, -0.358,  0.196,  0.003,  0.24 ,
-0.148, -0.235,  0.327,  0.486, -0.075],
[ 0.204, -0.061, -0.144, -0.068, -0.049, -0.239, -0.131, -0.175,
 0.109, -0.116, -0.498,  0.007, -0.13 , -0.074, -0.144, -0.133,
 0.078, -0.278,  0.13 , -0.321,  0.344,  0.128,  0.013, -0.2  ,
-0.228,  0.124,  0.026, -0.201, -0.041],
[ 0.153, -0.311, -0.171,  0.037, -0.037,  0.07 ,  0.145, -0.568,
-0.467,  0.356, -0.084, -0.287,  0.042,  0.126,  0.047,  0.018,
 0.094,  0.004,  0.063,  0.055, -0.101, -0.098,  0.05 ,  0.074,
-0.003,  0.017,  0.019,  0.005, -0.028],
[ 0.211, -0.182, -0.025,  0.019, -0.016,  0.09 , -0.164,  0.012,
-0.023, -0.044,  0.002,  0.185,  0.062, -0.152, -0.348,  0.228,
-0.129,  0.251, -0.128, -0.167,  0.105, -0.303,  0.229,  0.096,
-0.059,  0.271, -0.333,  0.332, -0.251],
[ 0.222, -0.003,  0.048,  0.042, -0.054,  0.076,  0.087,  0.051,
 0.051,  0.124, -0.053, -0.044, -0.138,  0.042,  0.152, -0.023,
-0.017,  0.014, -0.123, -0.111,  0.086,  0.053,  0.145, -0.227,
 0.009, -0.014, -0.242,  0.349,  0.754],
[ 0.213,  0.004,  0.175,  0.018, -0.027,  0.085, -0.221, -0.011,
-0.093, -0.045, -0.005,  0.081, -0.052, -0.175, -0.11 , -0.038,
-0.032,  0.092, -0.024,  0.109, -0.138, -0.254,  0.071, -0.148,
 0.22 ,  0.408,  0.586, -0.214,  0.257],
[ 0.201, -0.151,  0.138,  0.088, -0.018,  0.275,  0.125,  0.239,
```

```
 -0.012,  0.136,  0.209, -0.082, -0.139,  0.169,  0.227, -0.016,
 -0.089, -0.229,  0.069,  0.035,  0.207,  0.148,  0.03 , -0.385,
 -0.106,  0.203,  0.198,  0.237, -0.404],
[ 0.205,  0.182,  0.101, -0.021,  0.034, -0.034, -0.288,  0.005,
 -0.011,  0.111, -0.103,  0.094,  0.205,  0.031,  0.066,  0.277,
  0.31 , -0.008,  0.16 ,  0.647,  0.23 ,  0.002, -0.026, -0.036,
 -0.236,  0.   , -0.139, -0.012,  0.045],
[ 0.022, -0.025,  0.033, -0.934, -0.321,  0.124,  0.033,  0.018,
 -0.017,  0.03 ,  0.068, -0.001,  0.003,  0.003,  0.008,  0.007,
  0.001,  0.002,  0.001,  0.002, -0.002, -0.001, -0.003, -0.003,
 -0.004, -0.003, -0.001,  0.004, -0.   ],
[ 0.151, -0.186, -0.211, -0.15 ,  0.292, -0.525, -0.306,  0.213,
 -0.026,  0.468,  0.123,  0.166, -0.075, -0.054,  0.196, -0.039,
 -0.166,  0.035, -0.055, -0.043, -0.117,  0.058, -0.067, -0.01 ,
  0.078, -0.03 ,  0.023,  0.018, -0.04 ],
[ 0.206,  0.058,  0.059, -0.01 ,  0.002, -0.027, -0.265, -0.345,
  0.302, -0.19 ,  0.221, -0.387,  0.035, -0.277,  0.139,  0.052,
 -0.053, -0.012,  0.033,  0.009, -0.222,  0.415, -0.057,  0.067,
  0.132,  0.218, -0.137,  0.112, -0.068],
[ 0.2  ,  0.192, -0.152,  0.054, -0.146,  0.045, -0.005, -0.198,
  0.229,  0.005,  0.087,  0.047,  0.034, -0.07 ,  0.285,  0.027,
 -0.092, -0.12 , -0.05 , -0.115,  0.093, -0.559, -0.542, -0.002,
 -0.03 , -0.122,  0.048,  0.108, -0.036],
[ 0.186,  0.096,  0.326,  0.006,  0.037,  0.07 , -0.265, -0.06 ,
 -0.176,  0.013,  0.068, -0.127, -0.155, -0.189, -0.11 , -0.122,
  0.044,  0.046, -0.005, -0.086,  0.119, -0.098,  0.149, -0.244,
  0.266, -0.635, -0.081, -0.079, -0.165],
[ 0.217,  0.096,  0.122,  0.005,  0.003,  0.001,  0.024,  0.063,
 -0.04 ,  0.011, -0.015,  0.01 ,  0.025,  0.24 ,  0.17 ,  0.033,
 -0.133, -0.022,  0.234, -0.121,  0.457,  0.06 ,  0.088,  0.565,
  0.44 ,  0.085,  0.023, -0.074,  0.032],
[ 0.205,  0.223, -0.033,  0.042, -0.123,  0.004,  0.13 , -0.078,
  0.001,  0.122, -0.045,  0.106, -0.004,  0.025,  0.051, -0.047,
  0.128,  0.072, -0.825,  0.048,  0.118,  0.224,  0.04 ,  0.085,
  0.023,  0.035,  0.047, -0.172, -0.151],
[ 0.218, -0.06 , -0.051,  0.028, -0.029,  0.076,  0.057,  0.008,
  0.077,  0.033,  0.111,  0.094,  0.125, -0.008, -0.314, -0.326,
 -0.379, -0.577, -0.048,  0.321, -0.139, -0.057,  0.111,  0.132,
 -0.004, -0.066, -0.176, -0.107,  0.072],
[ 0.182, -0.251, -0.168,  0.015, -0.015, -0.067,  0.189, -0.021,
 -0.383, -0.417,  0.194,  0.287,  0.052, -0.233, -0.011, -0.081,
 -0.078,  0.18 ,  0.046,  0.151,  0.216,  0.267, -0.348, -0.101,
  0.066, -0.072, -0.019,  0.017,  0.088],
[ 0.206,  0.118,  0.014, -0.038,  0.044, -0.301,  0.299,  0.042,
 -0.028, -0.235,  0.15 , -0.06 , -0.135, -0.184,  0.319, -0.323,
  0.197,  0.092,  0.102,  0.081, -0.07 , -0.275,  0.422,  0.136,
 -0.253,  0.058, -0.034, -0.002, -0.091],
```

```
       [ 0.067,  0.195, -0.223, -0.218,  0.743,  0.473,  0.075, -0.07 ,
         0.014, -0.02 , -0.186,  0.068, -0.052, -0.122,  0.071, -0.082,
        -0.038,  0.044,  0.006,  0.004, -0.002,  0.019, -0.   ,  0.015,
        -0.002,  0.012, -0.   ,  0.003, -0.015],
       [ 0.178, -0.319,  0.048, -0.014,  0.006, -0.021,  0.157,  0.274,
         0.205, -0.06 , -0.372, -0.329, -0.427, -0.02 , -0.13 ,  0.023,
         0.043,  0.142, -0.076,  0.311, -0.092, -0.08 , -0.27 ,  0.183,
         0.119, -0.073, -0.012,  0.047, -0.063],
       [ 0.13 ,  0.319, -0.386,  0.032, -0.07 , -0.005,  0.111,  0.258,
        -0.23 ,  0.052,  0.204, -0.155, -0.227, -0.227, -0.245,  0.411,
         0.244, -0.304,  0.081, -0.12 , -0.071,  0.04 ,  0.015,  0.07 ,
         0.074, -0.002,  0.052,  0.005,  0.041],
       [ 0.194,  0.157,  0.002, -0.119,  0.261, -0.234, -0.102,  0.021,
        -0.145, -0.333,  0.155, -0.267,  0.091,  0.637, -0.249, -0.046,
         0.032,  0.022, -0.135, -0.062, -0.063, -0.101, -0.137, -0.122,
        -0.042,  0.048,  0.013,  0.092,  0.029]])
```

[33]:
```python
pca_out = pca.transform(df_scaled)
pca_out
```

[33]:
```
array([[ 1.05030157,  0.32806136, -0.26212573, …, -0.27747282,
         0.35391138,  0.01280102],
       [ 1.13209969,  0.36087074, -0.26075422, …, -0.38430763,
         0.31305797, -0.04196272],
       [ 1.3725952 ,  0.42210871, -0.23694544, …, -0.32019865,
         0.36630812, -0.15413755],
       …,
       [-0.49005491, -0.1667606 , -0.07166429, …,  0.36594767,
        -0.11585224,  0.0633935 ],
       [-0.51580557, -0.15763107, -0.09296948, …,  0.32501016,
        -0.08830259,  0.05705652],
       [-0.52409014, -0.14791434, -0.10220481, …,  0.25663525,
        -0.09838061,  0.05373432]])
```

[34]:
```python
def fit_svd(X, M=1):
    U, s, V = np.linalg.svd(X, full_matrices=False)
    return U[:, :M] @ (np.diag(s[:M]) @ V[:M, :])
```

[35]:
```python
df_imputed = df_scaled.copy()
```

[36]:
```python
row_index = np.random.choice(len(df_imputed), size=20, replace=False)
column_index = np.random.choice(df_imputed.shape[1], size=20)
```

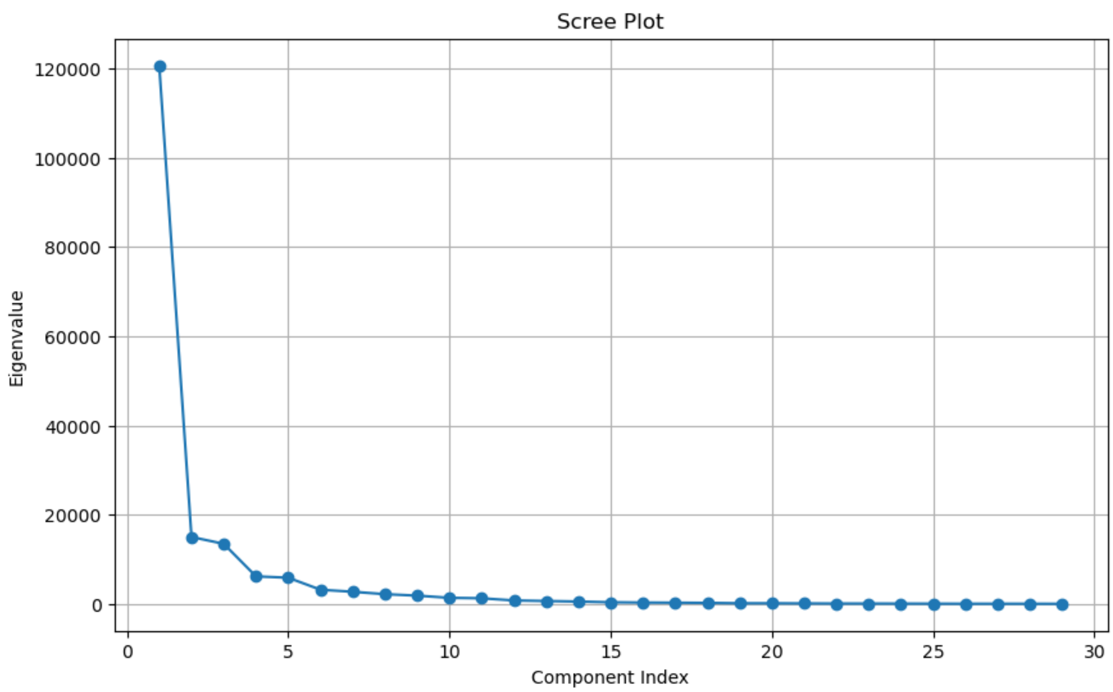
[37]:
```python
Xhat = df_imputed.copy()
xbar = np.nanmean(df_imputed, axis=0)
Xhat[row_index, column_index] = xbar[column_index]
```

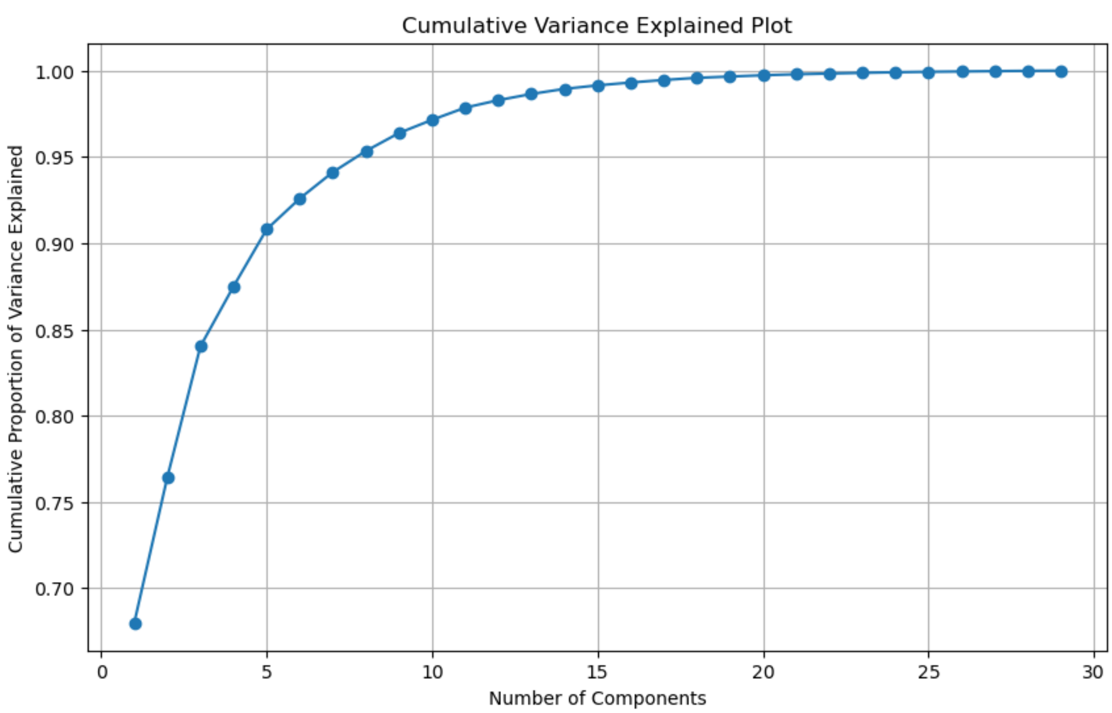
[38]:
```
thresh = 1e-7
rel_err = 1
iter_ = 0

ismiss = np.isnan(df_imputed)
Xscaled = (df_imputed - xbar) / np.sqrt(np.sum(~ismiss, axis=0))
Xscaled_nomiss = Xscaled[~ismiss]
mssold = np.mean(np.square(Xscaled_nomiss))
mss0 = np.mean(np.square(df_imputed[~ismiss]))
```

[39]:
```
while rel_err > thresh:
    iter_ += 1
    Xapp = fit_svd(Xhat, M=1)
    Xhat[ismiss] = Xapp[ismiss]
    mss = np.mean(np.square(df_imputed[~ismiss] - Xapp[~ismiss]))
    rel_err = (mssold - mss) / mss0
    mssold = mss

    print(f"Iter: {iter_}, MSS: {mss}, Rel. Err: {rel_err}")
```

```
Iter: 1, MSS: 0.3201319244821912, Rel. Err: -0.31996852578938073
```

[40]:
```
U, s, V = np.linalg.svd(df_scaled, full_matrices=False)

plt.figure(figsize=(10, 6))
plt.plot(range(1, len(s) + 1), s ** 2, marker='o', linestyle='-')
plt.title('Scree Plot')
plt.xlabel('Component Index')
plt.ylabel('Eigenvalue')
plt.grid(True)
plt.show()

cumulative_variance_explained = np.cumsum(s ** 2) / np.sum(s ** 2)
plt.figure(figsize=(10, 6))
plt.plot(range(1, len(s) + 1), cumulative_variance_explained, marker='o',␣
 ↪linestyle='-')
plt.title('Cumulative Variance Explained Plot')
plt.xlabel('Number of Components')
plt.ylabel('Cumulative Proportion of Variance Explained')
plt.grid(True)
plt.show()
```

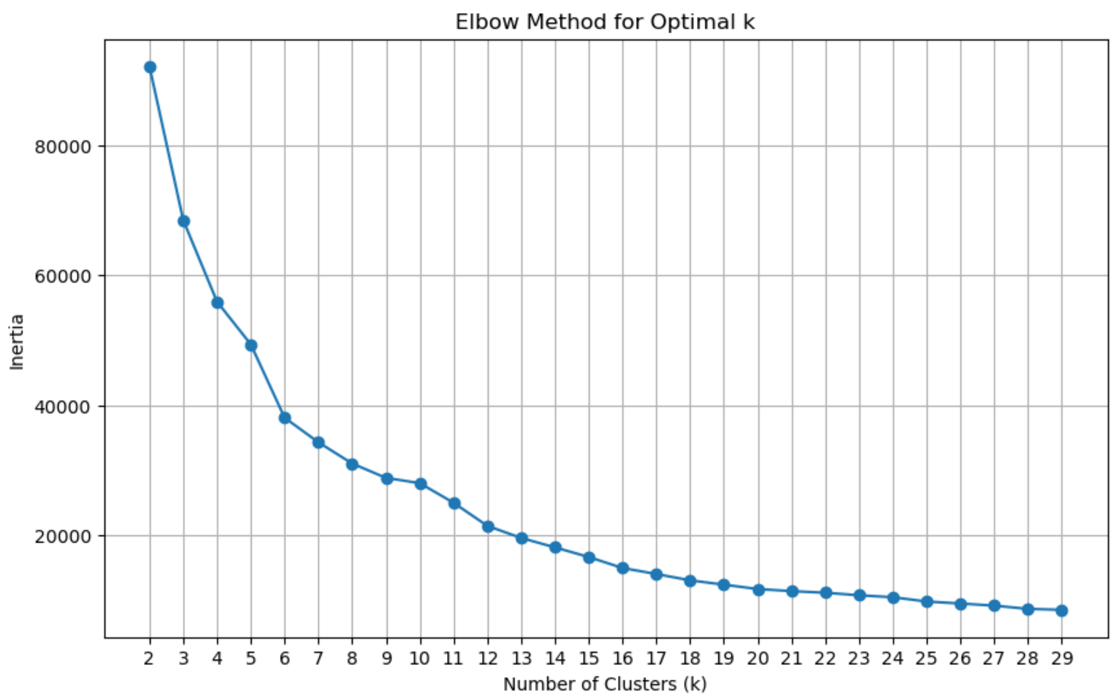

Scree Plot
Eigenvalue
Component Index
120000
100000
80000
60000
40000
20000
0
0
5
10
15
20
25
30


Cumulative Variance Explained Plot
Cumulative Proportion of Variance Explained
Number of Components
1.00
0.95
0.90
0.85
0.80
0.75
0.70
0
5
10
15
20
25
30

Clustering

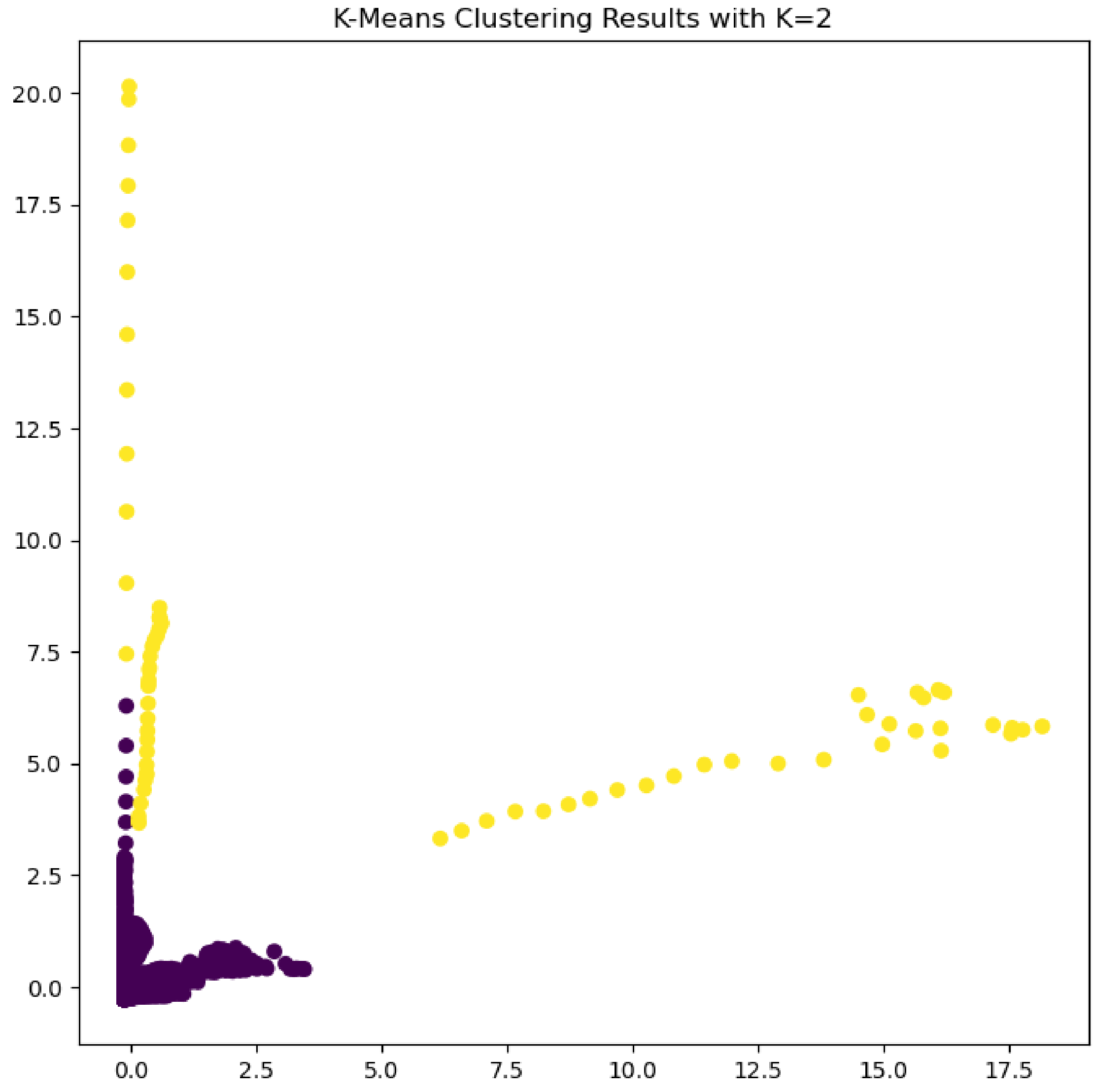
K Means

[41]:
```
inertia_values = []
k_values = range(2, 30)

for k in k_values:
    model = KMeans(n_clusters=k, random_state=42)
    model.fit(df_scaled)
    inertia_values.append(model.inertia_)

plt.figure(figsize=(10, 6))
plt.plot(k_values, inertia_values, marker='o')
plt.title('Elbow Method for Optimal k')
plt.xlabel('Number of Clusters (k)')
plt.ylabel('Inertia')
plt.xticks(k_values)
plt.grid(True)
plt.show()
```

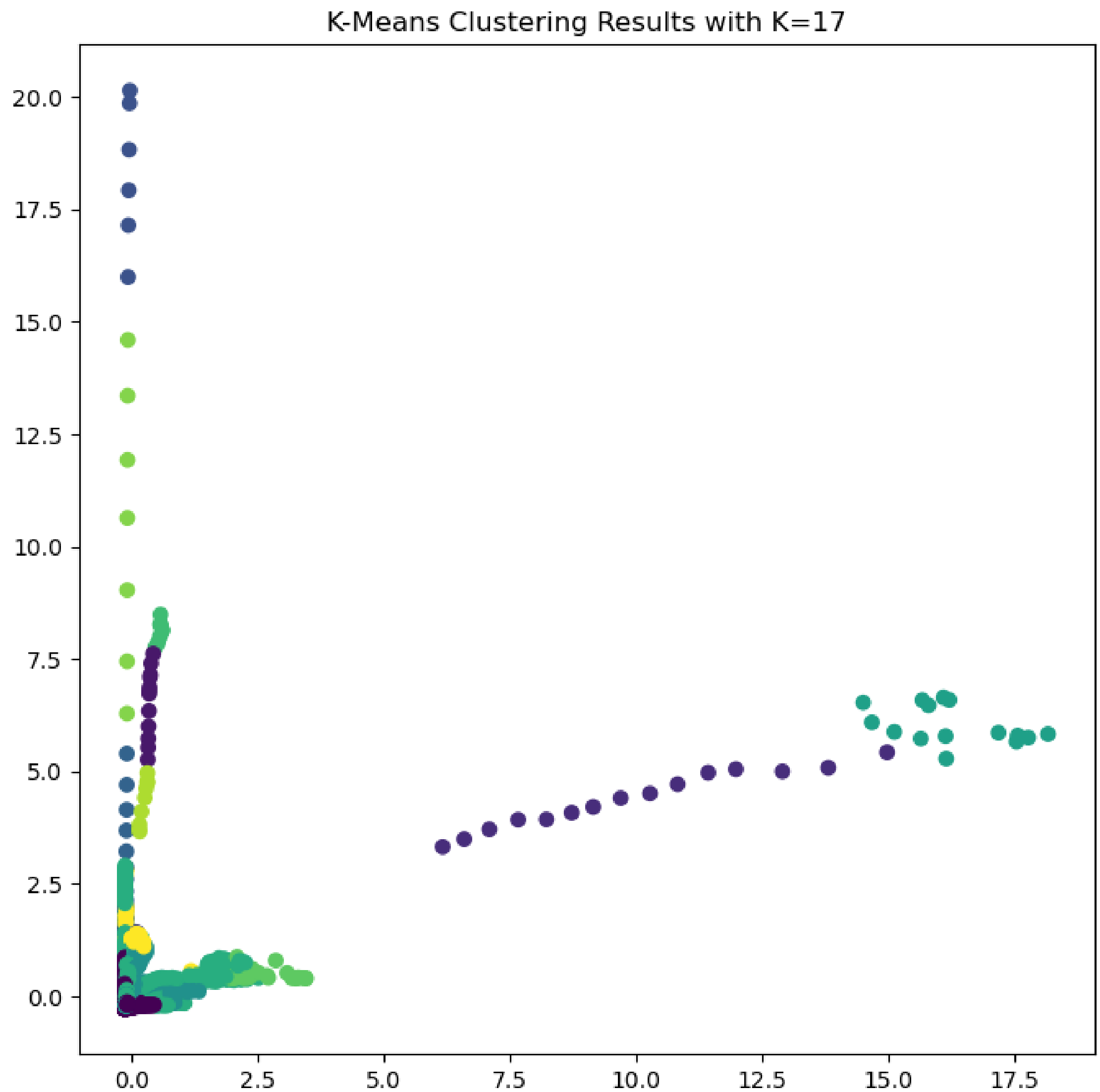


[42]:
```
kmeans_2 = KMeans(n_clusters=2, random_state=42)
kmeans_2.fit(df_scaled)
labels_2 = kmeans_2.labels_

fig, ax = plt.subplots(figsize=(8, 8))
ax.scatter(df_scaled[:, 0], df_scaled[:, 1], c=labels_2)
```

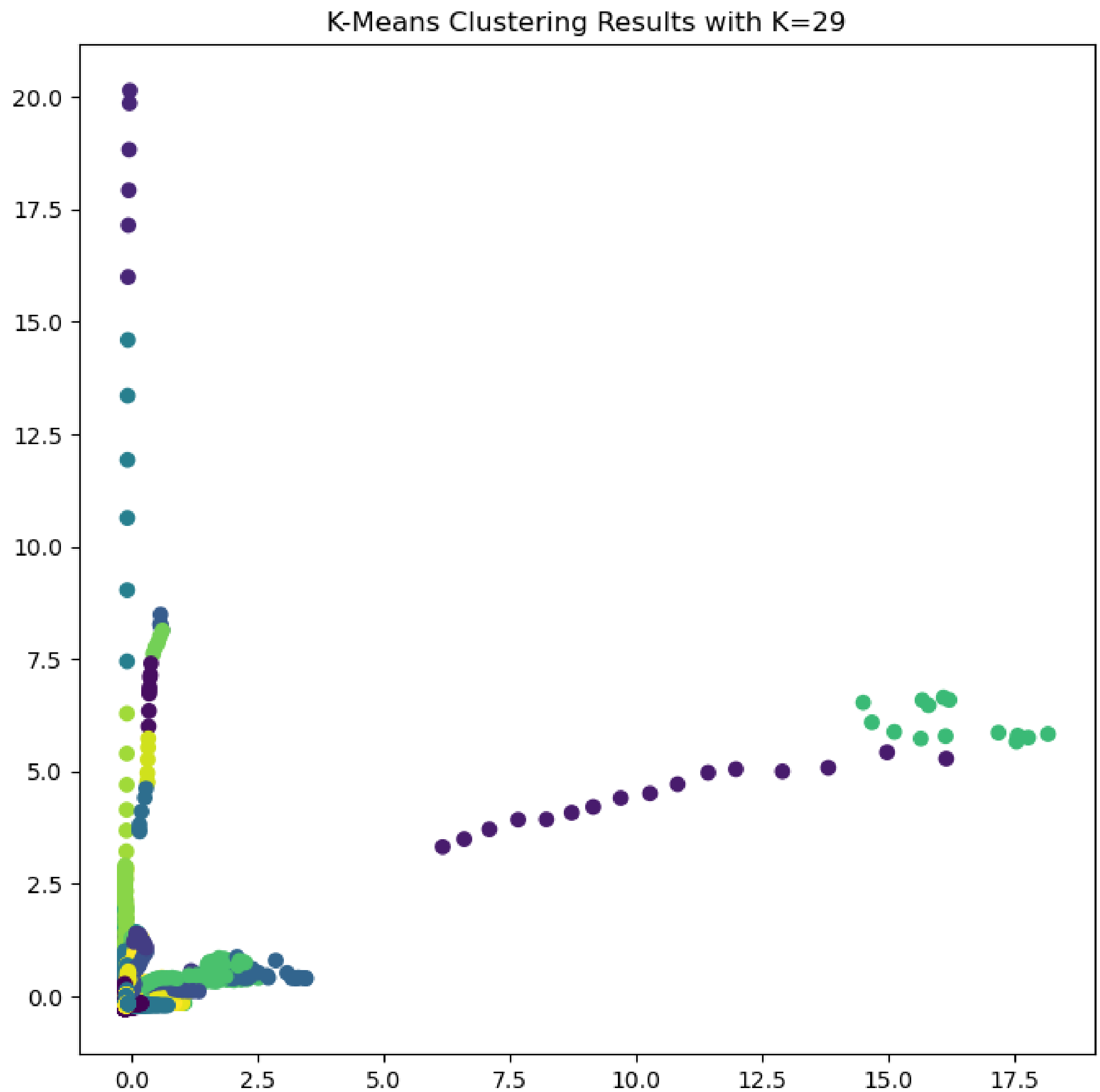
```python
ax.set_title("K-Means Clustering Results with K=2")
plt.show()
```



[43]:
```python
kmeans_17 = KMeans(n_clusters=17, random_state=42)
kmeans_17.fit(df_scaled)
labels_17 = kmeans_17.labels_

fig, ax = plt.subplots(figsize=(8, 8))
ax.scatter(df_scaled[:, 0], df_scaled[:, 1], c=labels_17)
ax.set_title("K-Means Clustering Results with K=17")
plt.show()
```



[44]:
```python
kmeans_29 = KMeans(n_clusters=29, random_state=42)
kmeans_29.fit(df_scaled)
labels_29 = kmeans_29.labels_

fig, ax = plt.subplots(figsize=(8, 8))
ax.scatter(df_scaled[:, 0], df_scaled[:, 1], c=labels_29)
ax.set_title("K-Means Clustering Results with K=29")
plt.show()
```



[45]:
```
kmeans_2_1 = KMeans(n_clusters=2, random_state=3, n_init=1)
kmeans_2_1.fit(df_scaled)
inertia_2_1 = kmeans_2_1.inertia_

kmeans_2_20 = KMeans(n_clusters=2, random_state=3, n_init=20)
kmeans_2_20.fit(df_scaled)
inertia_2_20 = kmeans_2_20.inertia_

print("Inertia for KMeans with 2 clusters and n_init=1:", inertia_2_1)
print("Inertia for KMeans with 2 clusters and n_init=20:", inertia_2_20)

kmeans_17_1 = KMeans(n_clusters=17, random_state=3, n_init=1)
kmeans_17_1.fit(df_scaled)
```

```
inertia_17_1 = kmeans_17_1.inertia_

kmeans_17_20 = KMeans(n_clusters=17, random_state=3, n_init=20)
kmeans_17_20.fit(df_scaled)
inertia_17_20 = kmeans_17_20.inertia_

print("Inertia for KMeans with 17 clusters and n_init=1:", inertia_17_1)
print("Inertia for KMeans with 17 clusters and n_init=20:", inertia_17_20)

kmeans_29_1 = KMeans(n_clusters=29, random_state=3, n_init=1)
kmeans_29_1.fit(df_scaled)
inertia_29_1 = kmeans_29_1.inertia_

kmeans_29_20 = KMeans(n_clusters=29, random_state=3, n_init=20)
kmeans_29_20.fit(df_scaled)
inertia_29_20 = kmeans_29_20.inertia_

print("Inertia for KMeans with 29 clusters and n_init=1:", inertia_29_1)
print("Inertia for KMeans with 29 clusters and n_init=20:", inertia_29_20)
```

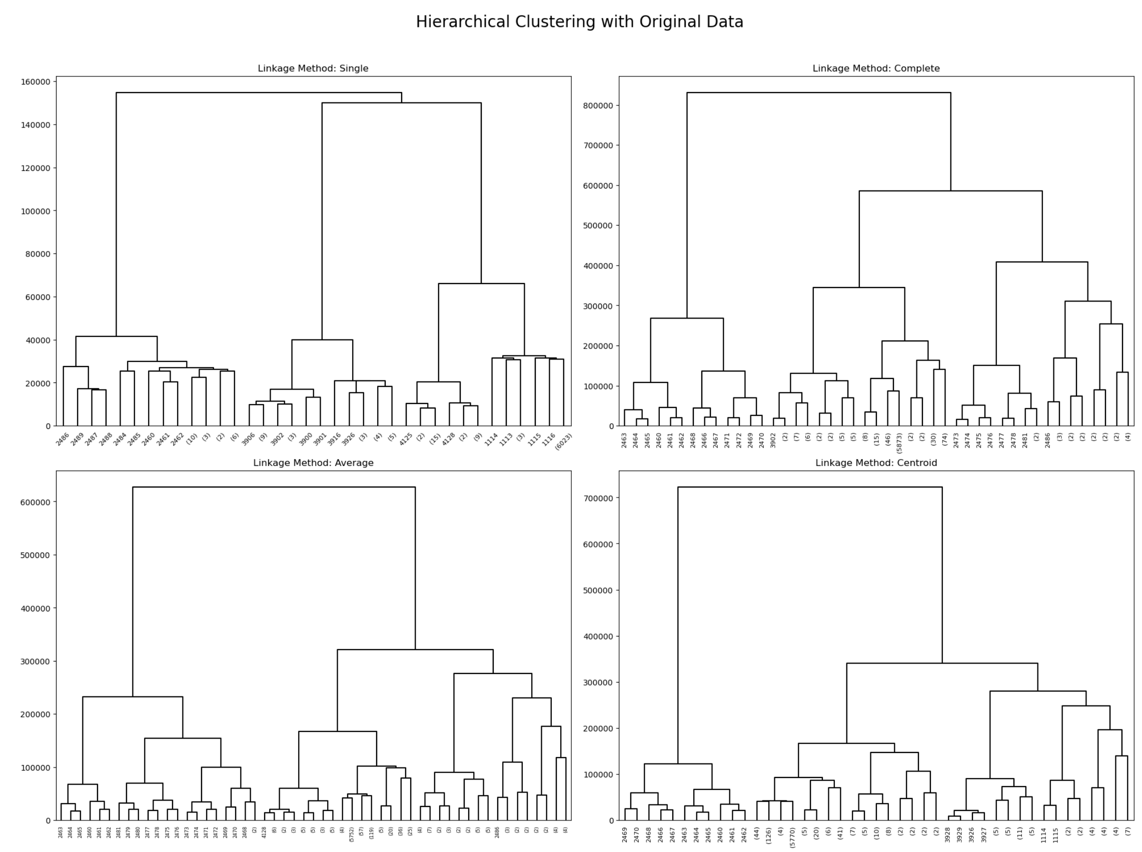
```
Inertia for KMeans with 2 clusters and n_init=1: 92144.82758027197
Inertia for KMeans with 2 clusters and n_init=20: 92144.82758027197
Inertia for KMeans with 17 clusters and n_init=1: 15774.632372538665
Inertia for KMeans with 17 clusters and n_init=20: 14187.884503011679
Inertia for KMeans with 29 clusters and n_init=1: 7970.138918026145
Inertia for KMeans with 29 clusters and n_init=20: 7837.581214293692
```

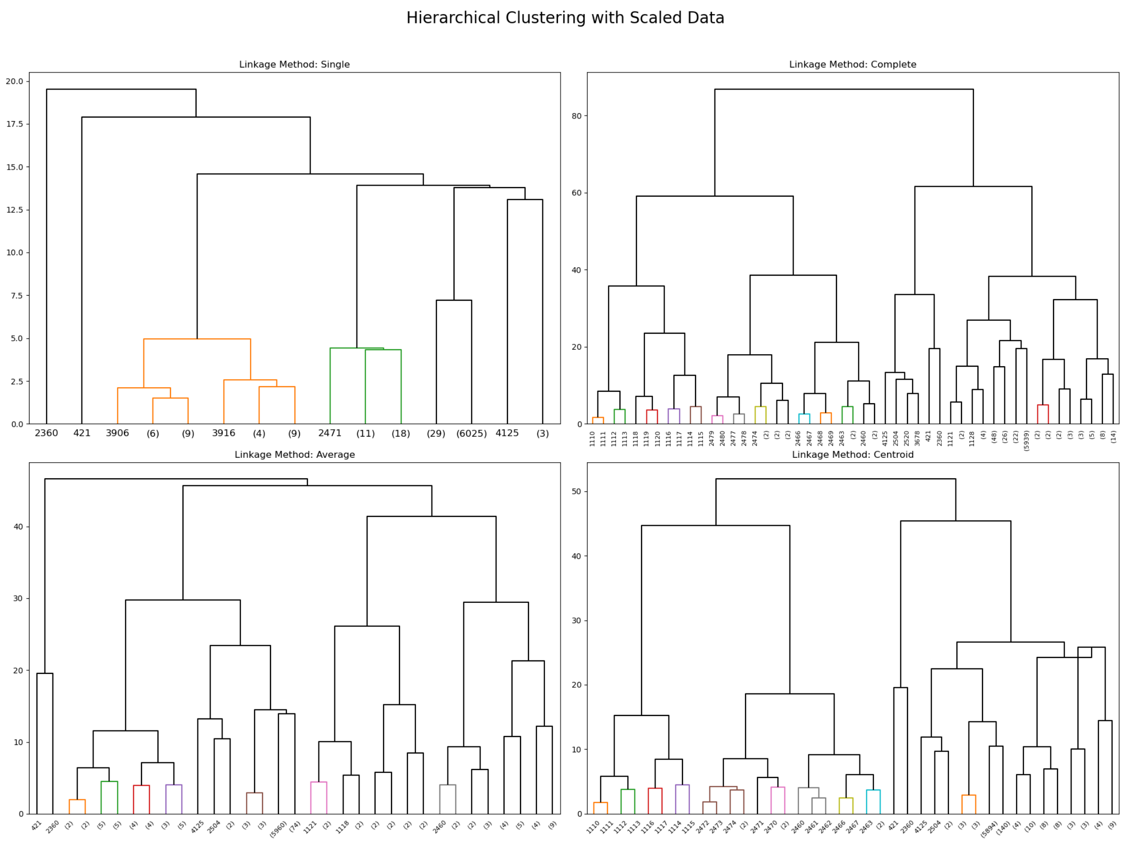
Hierarchical Clustering

[46]:
```
numeric_df = df.drop(columns=['Entity', 'Code', 'Year'])
```

[47]:
```
linkage_methods = ['single', 'complete', 'average']

def plot_dendrograms_sklearn(data, axes, title):
    for method, ax in zip(linkage_methods, axes.flatten()[:-1]):
        hc = AgglomerativeClustering(distance_threshold=0, n_clusters=None,␣
↪linkage=method)
        hc.fit(data)
        Z = linkage(data, method=method)
        dendrogram(Z, ax=ax, truncate_mode='level', p=5, color_threshold=5,␣
↪above_threshold_color='black')
        ax.set_title(f'Linkage Method: {method.capitalize()}')
    plt.suptitle(title, fontsize=20)
    plt.tight_layout(rect=[0, 0, 1, 0.96])

def plot_dendrogram_centroid(data, ax):
    Z = linkage(data, method='centroid')
```

```
    dendrogram(Z, ax=ax, truncate_mode='level', p=5, color_threshold=5,␣
 ↪above_threshold_color='black')
    ax.set_title('Linkage Method: Centroid')

fig, axes = plt.subplots(2, 2, figsize=(20, 15))
plot_dendrograms_sklearn(numeric_df, axes, "Hierarchical Clustering with␣
 ↪Original Data")
plot_dendrogram_centroid(numeric_df, axes.flatten()[-1])
plt.show()

fig, axes = plt.subplots(2, 2, figsize=(20, 15))
plot_dendrograms_sklearn(df_scaled, axes, "Hierarchical Clustering with Scaled␣
 ↪Data")
plot_dendrogram_centroid(df_scaled, axes.flatten()[-1])
plt.show()
```





[48]:
```python
def cut_and_print_clusters(data, Z, n_clusters, entity_column='Entity'):
    clusters = cut_tree(Z, n_clusters=n_clusters).flatten()
    data['Cluster'] = clusters
    for cluster in range(n_clusters):
        print(f"\nCluster {cluster + 1}:")
        print(data[data['Cluster'] == cluster][['Cluster', entity_column]])

print("Original Data Clusters:")
Z_complete_original = linkage(numeric_df, method='complete')
cut_and_print_clusters(df.copy(), Z_complete_original, n_clusters=5
                       )

print("\nScaled Data Clusters:")
Z_complete_scaled = linkage(df_scaled, method='complete')
scaled_df_with_clusters = pd.DataFrame(df_scaled, columns=numeric_df.columns)
scaled_df_with_clusters['Entity'] = df['Entity'].values
cut_and_print_clusters(scaled_df_with_clusters, Z_complete_scaled,␣
↪n_clusters=5, entity_column='Entity')
```

```
Original Data Clusters:
```

```
Cluster 1:
      Cluster       Entity
0           0  Afghanistan
1           0  Afghanistan
2           0  Afghanistan
3           0  Afghanistan
4           0  Afghanistan
…         …          …
6835        0     Zimbabwe
6836        0     Zimbabwe
6837        0     Zimbabwe
6838        0     Zimbabwe
6839        0     Zimbabwe

[6050 rows x 2 columns]

Cluster 2:
      Cluster Entity
1140        1  China
1141        1  China
1142        1  China
1143        1  China
1144        1  China
1145        1  China
1146        1  China
1147        1  China
1148        1  China
1149        1  China
2692        1  India
2693        1  India
2694        1  India
2695        1  India
2696        1  India
2697        1  India
2698        1  India
2699        1  India

Cluster 3:
      Cluster Entity
2670        2  India
2671        2  India
2672        2  India
2673        2  India
2674        2  India
2675        2  India
2676        2  India
2677        2  India
2678        2  India
```

```
2679        2  India
2680        2  India
2681        2  India
2682        2  India

Cluster 4:
      Cluster Entity
2683        3  India
2684        3  India
2685        3  India
2686        3  India
2687        3  India
2688        3  India
2689        3  India
2690        3  India
2691        3  India

Cluster 5:
      Cluster   Entity
4170        4  Nigeria
4171        4  Nigeria
4172        4  Nigeria
4173        4  Nigeria
4174        4  Nigeria
4175        4  Nigeria
4176        4  Nigeria
4177        4  Nigeria
4178        4  Nigeria
4179        4  Nigeria
4180        4  Nigeria
4181        4  Nigeria
4182        4  Nigeria
4183        4  Nigeria
4184        4  Nigeria
4185        4  Nigeria
4186        4  Nigeria
4187        4  Nigeria
4188        4  Nigeria
4189        4  Nigeria
4190        4  Nigeria
4191        4  Nigeria
4192        4  Nigeria
4193        4  Nigeria
4194        4  Nigeria
4195        4  Nigeria
4196        4  Nigeria
4197        4  Nigeria
4198        4  Nigeria
```

```
4199        4  Nigeria

Scaled Data Clusters:

Cluster 1:
      Cluster       Entity
0           0  Afghanistan
1           0  Afghanistan
2           0  Afghanistan
3           0  Afghanistan
4           0  Afghanistan
…         …            …
6115        0     Zimbabwe
6116        0     Zimbabwe
6117        0     Zimbabwe
6118        0     Zimbabwe
6119        0     Zimbabwe

[6082 rows x 2 columns]

Cluster 2:
      Cluster      Entity
421         1  Bangladesh
2360        1       Haiti
2504        1   Indonesia
2520        1        Iran
3678        1     Myanmar
4125        1    Pakistan

Cluster 3:
      Cluster Entity
1110        2  China
1111        2  China
1112        2  China
1113        2  China
1114        2  China
1115        2  China
1116        2  China
1117        2  China
1118        2  China
1119        2  China
1120        2  China

Cluster 4:
      Cluster Entity
2460        3  India
2461        3  India
2462        3  India
```

```
2463        3  India
2464        3  India
2465        3  India
2466        3  India
2467        3  India
2468        3  India
2469        3  India

Cluster 5:
      Cluster Entity
2470        4  India
2471        4  India
2472        4  India
2473        4  India
2474        4  India
2475        4  India
2476        4  India
2477        4  India
2478        4  India
2479        4  India
2480        4  India
```

[ ]:
```

```