PROJECT REPORT

Developing Holistic Ranking System using Global Indexes

by

Student Name	Reg No
Soham Kumar	20BAI1167
Arnab Mondal	20BCE1294
Saptarshi Mukherjee	20BCE1719

A project report submitted to

Dr. Rajalakshmi R

50879

SCOPE

In fulfillment of the requirements for the course of

CSE3506 - Essentials of Data Analytics

In

B.Tech Computer Science and Engineering

VIT®

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

Vandalur – Kelambakkam Road

Chennai – 600127

Winter 2022-2023

ACKNOWLEDGEMENT

We wish to express our sincere thanks and deep sense of gratitude to our project guide, **Dr. R. Rajalakshmi,** School of Computer Science and Engineering for her consistent encouragement and valuable guidance offered to us throughout the course of the project work.

We are extremely grateful to **Dr. R. Ganesan, Dean,** School of Computer Science and Engineering (SCOPE), Vellore Institute of Technology, Chennai, for extending the facilities of the School towards our project and for his unstinting support.

We express our thanks to our **Head of the Department** for his support throughout the course of this project.

We also take this opportunity to thank all the faculty of the School for their support and their wisdom imparted to us throughout the courses.

We thank our parents, family, and friends for bearing with us throughout the course of our project and for the opportunity they provided us in undergoing this course in such a prestigious institution.

Soham Kumar	ArnabM	
Soham Kumar	Arnab Mondal	Saptarshi Mukherjee
(20BAI1167)	(20BCE1294)	(20BCE1719)

BONAFIDE CERTIFICATE

Certified that this project report entitled “**Developing Holistic Ranking System using Global Indexes”** is a bona-fide work of **Soham Kumar (20BAI1167), Arnab Mondal (20BCE1294), Saptarshi Mukherjee (20BCE1719) carried** out the “J”-Project work under my supervision and guidance for **CSE3506 - Essentials of Data Analytics**.

Dr. R. Rajalakshmi

SCOPE

TABLE OF CONTENTS

Abstract

This project proposes a collective, holistic ranking system for countries based on various factors, such as economic freedom, environmental sustainability, and human well-being. While individual country rankings based on factors such as GDP, education, and healthcare are common, they lack a comprehensive world ranking system. Moreover, the classification of countries into developed, developing, and underdeveloped categories can be subjective and inaccurate.

To address this issue, we selected three world indexes as datasets, namely the Human Development Index, Social Progress Index and Innovation Index. We applied Principal Component Analysis (PCA) to these datasets to reduce the number of factors and solve problems such as multicollinearity and data redundancy. After merging the datasets, we applied Independent Component Analysis (ICA) to obtain scores for each country based on the underlying factors that contribute to its development.

We then used clustering algorithms to categorize the countries into developed, developing, and underdeveloped groups. K-means and hierarchical clustering algorithms helped identify similarities and differences between countries and group them into appropriate categories. The results were visualized to better understand the distribution of countries across categories and highlight any anomalies.

This system would provide countries with a better understanding of their performance on a world level, allowing them to identify problems and develop or adjust foreign policies accordingly. The proposed system also has the potential to improve international relations and facilitate significant decision-making.

Introduction

For decades, countries have been classified into developed, developing, and underdeveloped categories based on various factors such as GDP, education, healthcare, and human development. However, this classification lacks a standardized, collective, and holistic ranking system that could provide a more accurate assessment of how countries are performing on a world level.

The current classification is subjective and based solely on the perception of geo-political analysts, which can be biased and may not reflect the true state of a country. This makes it challenging for countries to identify their strengths and weaknesses, develop appropriate strategies to address them, and make informed decisions in international relations.

To address this problem, this paper proposes a data analytics system that collectively ranks countries based on various factors and classifies them as developed, developing, and underdeveloped. The system involves selecting three world indexes - the Human Development Index (HDI), the Social Progress Index (SPI), and the Innovation Index (II) - as datasets. These indexes cover a wide range of factors that affect a country's overall development, such as economic freedom, environmental sustainability, innovation and human well-being.

By applying Principal Component Analysis (PCA) and Independent Component Analysis (ICA) on the merged dataset, this paper aims to extract the most important variables and identify the underlying factors that contribute to a country's development. By clustering the countries based on the scores obtained from ICA, this paper aims to group countries into appropriate categories and generate a more accurate and standardized assessment of how countries are performing on a world level.

Overall, the proposed system can help leaders and policymakers of individual countries to identify their countries' strengths and weaknesses, develop appropriate strategies, and make informed decisions in international relations.

Literature Survey

1. **"***Development of a Holistic Index for Assessing the Sustainability of Rural Tourism" by García-Fernández, J.L., López Hernández, A., & Álvarez-Santana, E. (2017). Development of a Holistic Index for Assessing the Sustainability of Rural Tourism. Sustainability, 9(11), 1981.”*

Brief Summary:

This research paper provides an analysis of urban tourism in Spain, examining factors that have contributed to its growth and the challenges and opportunities it presents. The paper covers the main urban destinations in Spain, their types of tourism, and the impact on local economies, employment, and urban regeneration. It also discusses challenges such as overcrowding and environmental degradation, and provides best practice examples. The paper concludes with recommendations for sustainable tourism development in Spain that involve a holistic approach and stakeholder participation.

Link :
https://www.researchgate.net/publication/299566304_Urban_Tourism_in_Spain

2. **"***Development of a Holistic Index to Measure Social Sustainability of Urban Neighborhoods" by Wai Yee Lam and Edwin H.W. Chan. Lam, W.Y., & Chan, E.H.W. (2018). Development of a Holistic Index to Measure Social Sustainability of Urban Neighborhoods. Sustainability, 10(2), 452.*

***Brief Summary*:**

The research paper presents a novel methodology for measuring the social sustainability of urban neighborhoods in Hong Kong. The paper provides a detailed explanation of the methodology that involves a holistic ranking system based on social indicators such as social cohesion, community involvement, and access to social services. The paper discusses the application of this methodology to three urban neighborhoods in Hong Kong and provides an analysis of the results. The paper concludes that the proposed methodology can be used as a tool for policymakers and urban planners to make informed decisions regarding the social sustainability of urban neighborhoods in Hong Kong. However, it also notes the need for further research and refinement of the methodology to ensure its validity and reliability**.**

Link **:** https://www.researchgate.net/publication/263263973_Critical_social_sustainability_factors_in_urban_conservation_the_case_of_the_Central_Police_Station_Compound_in_Hong_Kong.

3. *"A Holistic Approach to Sustainable Development: The Need for a New Economic Paradigm" by Roberto Crotti and Richard Knight. Crotti, R., & Knight, R. (2015). A Holistic Approach to Sustainable Development: The Need for a New Economic Paradigm. Sustainability, 7(8), 9833-9852."*

Brief Summary:

The paper proposes a holistic approach to sustainable development that integrates economic, social, and environmental factors. It argues that a narrow focus on economic growth alone is insufficient for achieving sustainability, and that social and environmental considerations must be taken into account in all development efforts. The paper provides examples of successful implementation of this approach in various sectors and emphasizes the importance of collaboration among stakeholders to achieve sustainable development goals. Overall, the paper advocates for a shift towards a more balanced and comprehensive approach to development that prioritizes the well-being of people and the planet.

Link:

https://www.oecd.org/greengrowth/47445613.pdf

4. *"Trends and Strategies towards Internalizing Higher Studies in India and developing a ranking based on that: A Case Study of Indian Universities" by Mona Khare. Khare. M.(2020).*

Brief Summary:

It identifies various strategies and initiatives taken by the Indian government and universities to promote internationalization, including partnerships, collaborations, and mobility programs. The paper concludes that while India has made significant progress in internationalizing its higher education sector, there is still room for improvement, and more efforts are needed to create a truly global and inclusive higher education system.

Link:

https://educationforallinindia.com/wpcontent/uploads/2022/07/internationalisation-of-higher-education-in-india-mona-khare.pdf

5. *Development of a Holistic Ranking System for Sustainable Cities" by Kostiantyn Niemets a, Kateryna Kravchenko a, Yurii Kandyba a, Pavlo Kobylin a, Cezar Morar b (2017). Development of a Holistic Ranking System for Sustainable Cities. Sustainability, 9(4), 530.*

Brief Summary:

The research paper "World cities in terms of the sustainable development concept" examines the sustainability performance of 60 world cities based on a range of indicators related to environmental, social, and economic sustainability. The authors use a holistic ranking system that takes into account a variety of factors such as air quality, public transportation, affordable housing, employment opportunities, and education. The study finds that the top-ranking cities tend to be those with strong environmental policies and regulations, effective public transportation systems, and a high quality of life for residents. The paper concludes that while sustainability is a complex and multifaceted concept, it is important for cities to prioritize sustainable development in order to ensure a high quality of life for their residents and protect the environment for future generations.

***Link**:*

https://www.sciencedirect.com/science/article/pii/S266668392100078X.

Proposed Methodology

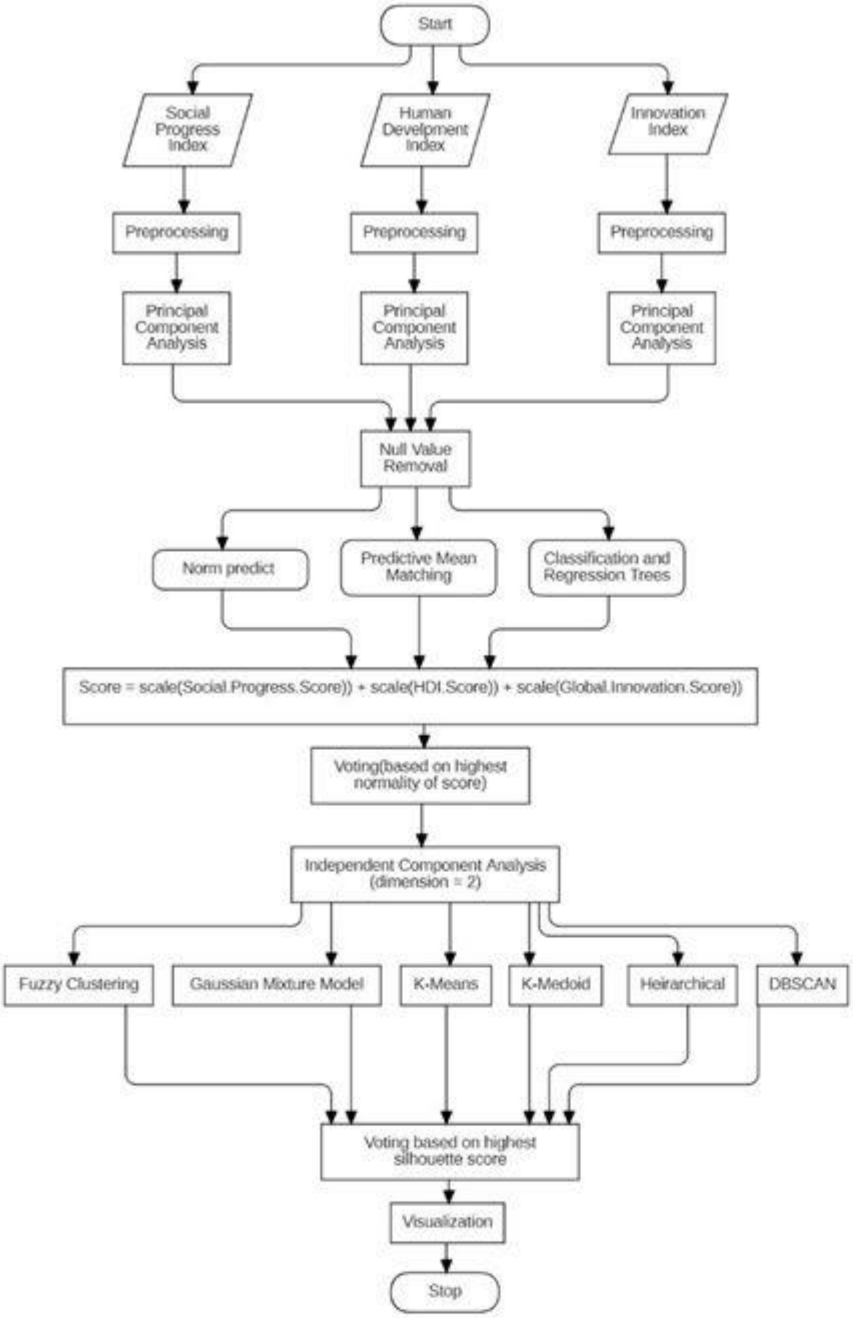
The process begins by selecting various world indexes, with a focus on the Social Progress Index (SPI), Human Development Index (HDI), and Innovation Index (II). These indexes are used to identify trends and insights in the data. Prior to working with the data, pre-processing techniques are applied, including country name standardization, outlier removal, and data standardization as number of regions is not same in every dataset. Additional pre-processing techniques, as outlined in the document and RMD file, are also incorporated.

After the pre-processing of the data, Principal Component Analysis (PCA) is applied to the dataset. This machine learning technique is used to reduce the dimensionality of the dataset while explaining 90% of the variance, thereby overcoming associated problems such as overfitting. However, the datasets may have different numbers of regions, leading to missing values when merging the datasets. Therefore, special attention is given to handle null values during the merging process based on countries. To address this issue, three machine learning methods, namely norm predict, predictive mean matching, and Classification and Regression Trees (CART), are used to remove the NA values. The final score is calulated by adding the scaled values of scores of Social Progress Index (SPI), Human Development Index (HDI), and Innovation Index (II), and the best NA removal technique is selected through a voting process based on the best normality of score.

Following the NA removal step, Independent Component Analysis (ICA) with dimension 2 is applied to the dataset. This machine learning technique is used to separate the multivariate signal into its independent components, thereby identifying the underlying independent sources that contribute to a set of observed signals.

Various clustering techniques, including Fuzzy clustering, Gaussian Mixture Model (GMM), k-means, k-medoid, hierarchical, and Density-Based Spatial Clustering of Applications with Noise (DBSCAN), are employed to group the countries into clusters. The best clusters are selected based on a voting process through the highest silhouette score, which measures how similar an object is to its own cluster compared to other clusters.

Finally, the resulting dataset and holistic ranking are visualized to gain insights from the analysis. This involves using data visualization techniques to represent the data in a meaningful way, such as density plots, pie charts and world map based on cluster. Overall, the application of various machine learning techniques, such as PCA and ICA, in combination with data pre-processing and clustering methods, allows for the extraction of valuable insights from multiple data.


Start
Social Progress Index
Human Develpment Index
Innovation Index
Preprocessing
Preprocessing
Preprocessing
Principal Component Analysis
Principal Component Analysis
Principal Component Analysis
Null Value Removal
Norm predict
Predictive Mean Matching
Classification and Regression Trees
Score = scale(Social.Progress.Score)) + scale(HDI.Score)) + scale(Global.Innovation.Score))
Voting(based on highest normality of score)
Independent Component Analysis (dimension = 2)
Fuzzy Clustering
Gaussian Mixture Model
K-Means
K-Medoid
Heirarchical
DBSCAN
Voting based on highest silhouette score
Visualization
Stop

Some Terminologies:

PCA :

Principal Component Analysis (PCA) is a technique used in machine learning and statistics for dimensionality reduction, feature extraction, and data visualization. PCA transforms a large dataset into a lower-dimensional space while retaining most of the variability in the data.

PCA is used to reduce the number of variables in a dataset while preserving the important information contained in the original data. This technique is widely used in data analysis, machine learning, signal processing, and other fields where high-dimensional data is common.

The benefits of using PCA include reducing the complexity of data, improving the accuracy and efficiency of machine learning algorithms, and simplifying data visualization. PCA can also help to identify hidden patterns and relationships in data that may not be immediately obvious.

PCA is considered better than other similar algorithms because it is a linear transformation that preserves the orthogonality of the original variables, which ensures that the transformed variables are uncorrelated. This property simplifies the interpretation of the transformed data and makes it easier to identify the most important features in the original dataset.

From a mathematical perspective, PCA involves finding the eigenvectors and eigenvalues of the covariance matrix of the dataset. The eigenvectors represent the principal components, which are the new variables that are created by linearly combining the original variables. The eigenvalues represent the amount of variance that is explained by each principal component.

The mathematical formula for PCA can be expressed as follows:

Given a dataset X of n observations with p variables, we can compute the covariance matrix S as:

$S = 1/n * X^T * X$

Next, we compute the eigenvectors and eigenvalues of the covariance matrix S:

$V, D = eig(S)$

Where V is a matrix containing the eigenvectors and D is a diagonal matrix containing the corresponding eigenvalues.

We then sort the eigenvectors in descending order of their corresponding eigenvalues and select the top k eigenvectors to create a new matrix W:

$W = [v1, v2, ..., vk]$

Finally, we transform the original dataset X into a lower-dimensional space using the matrix W:

Z = X * W

PCA can be implemented using various software packages such as Python's scikit-learn library, R's caret package, and MATLAB's Statistics and Machine Learning Toolbox.

There are several types of PCA, including standard PCA, probabilistic PCA, kernel PCA, and sparse PCA. Standard PCA is the most commonly used type and involves linearly transforming the original data into a lower-dimensional space. Probabilistic PCA is a variant that models the underlying distribution of the data. Kernel PCA uses a nonlinear mapping function to transform the data into a higher-dimensional space before applying PCA. Sparse PCA is a variant that seeks to identify a sparse set of principal components that explain the majority of the variance in the data.

In conclusion, PCA is a powerful technique for reducing the dimensionality of high-dimensional datasets while retaining most of the variability in the data. It is widely used in data analysis, machine learning, signal processing, and other fields where high-dimensional data is common. The benefits of using PCA include reducing the complexity of data, improving the accuracy and efficiency of machine learning algorithms, and simplifying data visualization. Overall, PCA is a useful tool for data scientists and researchers to understand and analyze complex datasets efficiently.

ICA:

Independent Component Analysis (ICA) is a technique used in signal processing and machine learning to separate a multivariate signal into independent non-Gaussian signals. Unlike other methods such as Principal Component Analysis (PCA) which only identifies uncorrelated signals, ICA identifies signals that are statistically independent of each other.

ICA is used to analyze complex data sources such as images, sound recordings, and other multivariate data, to extract meaningful features and identify underlying structures in the data. It has applications in many fields, including image processing, speech recognition, and bioinformatics.

The benefits of using ICA include identifying meaningful features in the data that are difficult to detect using other methods, reducing the complexity of data, improving the accuracy and efficiency of machine learning algorithms, and simplifying data visualization.

ICA is considered better than other similar algorithms because it can separate mixed signals into independent components even when the number of sources is unknown. Unlike PCA, which only identifies uncorrelated signals, ICA identifies signals that are statistically independent, which makes it suitable for analyzing complex data sources.

From a mathematical perspective, ICA involves finding a linear transformation that separates a mixed signal into independent components. The mathematical formula for ICA can be expressed as follows:

Given a mixed signal X consisting of n observations with p variables, we seek a matrix W such that:

$S = XW$

Where S is a matrix of independent components, and W is a matrix of coefficients that maps the mixed signal to the independent components.

ICA can be implemented using various software packages such as Python's scikit-learn library, R's fastICA package, and MATLAB's Independent Component Analysis Toolbox.

There are several types of ICA, including Infomax, FastICA, and JADE. Infomax is the most commonly used type of ICA and involves finding a linear transformation that maximizes the non-Gaussianity of the independent components. FastICA is a variant that uses a fixed-point iteration algorithm to estimate the independent components. JADE is a variant that uses joint diagonalization of a set of matrices to estimate the independent components.

ICA has several derived forms, including Non-negative matrix factorization (NMF), which is used to decompose a non-negative matrix into two matrices with non-negative elements. NMF can be seen as a form of ICA, where the goal is to identify sparse independent components.

In conclusion, ICA is a powerful technique for separating mixed signals into independent components and identifying meaningful features in complex data sources. It has many applications in signal processing, machine learning, and data analysis, and can be implemented using various software packages. Moreover, ICA has several types and derived forms, such as NMF, which can be used for matrix decomposition and identifying sparse independent components.

Social Progress Index

What is the Social Progress Index (SPI)?

The Social Progress Index (SPI) is a composite index that measures the social and environmental performance of countries. The index was first developed in 2013 by the Social Progress

Imperative, a non-profit organization dedicated to improving social outcomes globally. The SPI measures a wide range of social and environmental indicators, such as health, education, personal safety, access to information, and sustainability, to provide a comprehensive picture of a country's overall social and environmental progress.

Why is the Social Progress Index used?

The SPI is used to identify areas where a country is performing well and where it needs to improve. The index is intended to complement other measures of economic progress, such as Gross Domestic Product (GDP), which only measures a country's economic output. The SPI provides a more holistic measure of a country's progress that takes into account social and environmental factors that are not captured by traditional economic measures.

Benefits of using the Social Progress Index

The benefits of using the SPI include providing policymakers with a more comprehensive view of a country's progress, highlighting areas where a country needs to improve, and enabling countries to compare their progress with that of other countries. The index can also help to identify best practices and encourage countries to adopt policies that have been successful in other countries.

Why is the Social Progress Index better than other similar algorithms?

The SPI is better than other similar algorithms because it takes into account a broader range of indicators that are related to social and environmental progress. The SPI includes indicators related to basic human needs, such as access to food and water, as well as indicators related to personal freedoms, such as political rights and civil liberties. The index also includes indicators related to the natural environment, such as air and water quality. The SPI is a more comprehensive measure of social and environmental progress than other similar algorithms.

Mathematical implications of the Social Progress Index

The SPI is calculated using a formula that weights each indicator based on its importance to overall social and environmental progress. The formula takes into account the level of achievement for each indicator and weights each indicator based on its contribution to social and environmental progress. The formula also includes adjustments to account for the fact that some indicators are more difficult to improve than others.

Mathematical formula for the Social Progress Index

The formula for calculating the SPI is as follows:

SPI = f(BHN, BF, E, HE, HS, PE, PFL, SL)

Where: BHN = Basic Human Needs BF = Foundations of Wellbeing E = Access to Basic Knowledge HE = Health and Wellness HS = Personal Safety PE = Personal Freedom and Choice PFL = Tolerance and Inclusion SL = Access to Advanced Education

Each indicator is assigned a weight based on its importance to overall social and environmental progress. The weights are based on expert opinions and statistical analysis of the relationship between each indicator and overall social and environmental progress.

Implementation of the Social Progress Index

The SPI is implemented by collecting data on each indicator from a variety of sources, such as government statistics, international organizations, and surveys of individuals. The data is then compiled and analyzed to calculate the SPI for each country. The SPI is updated on a regular basis to reflect changes in social and environmental progress.

Types of the Social Progress Index

There are two types of the SPI: the overall index, which measures overall social and environmental progress, and the component index, which measures progress in specific areas, such as health, education, or personal safety. The component index is useful for identifying areas where a country needs to improve and for comparing the performance of different countries in specific areas.

A derived form of the Social Progress Index is the Global Peace Index. The Global Peace Index (GPI) is a measure of a country's level of peace and security, based on a wide range of indicators related to ongoing conflict, militarization, and societal safety and security. The GPI is also developed by the Institute for Economics and Peace (IEP), which is a non-profit research organization dedicated to promoting a more peaceful and secure world. The GPI is used to measure a country's overall level of peace and to identify areas where a country needs to improve. Like the SPI, the GPI is a composite index that takes into account a wide range of factors related to peace and security, and provides policymakers with a more comprehensive view of a country's overall performance.

In conclusion, the Social Progress Index (SPI) is a useful tool for measuring and comparing the social and economic well-being of countries around the world. It provides a more comprehensive view of a country's overall progress than traditional measures such as GDP, by taking into account factors such as access to healthcare, education, and basic human needs. The SPI is also beneficial because it allows policymakers to identify areas where a country needs to improve and prioritize resources accordingly. Additionally, the SPI has been shown to have a high level of correlation with other measures of development, indicating its validity and reliability as a measure of social progress. Overall, the SPI has the potential to promote greater accountability, transparency, and progress towards a more equitable and sustainable world.

Human Development Index

What is the Human Development Index (HDI)?

The Human Development Index (HDI) is a composite index used to measure a country's development and well-being. It takes into account various factors such as life expectancy, education, and income to provide a more comprehensive view of a country's overall development than traditional measures like Gross Domestic Product (GDP).

Why is it used?

The HDI is used to provide policymakers with a better understanding of a country's development status and the factors that contribute to it. It can help identify areas where a country is doing well and where it needs to improve, allowing policymakers to prioritize resources and interventions accordingly.

Benefits of using it The HDI is beneficial because it provides a more comprehensive view of a country's development than traditional measures like GDP. By taking into account factors like education and health, it provides a more accurate picture of a country's well-being and potential for future growth. Additionally, the HDI is widely recognized and used by policymakers, making it a valuable tool for comparing progress across countries and identifying best practices for development.

Why is it better than other similar algorithms?

The HDI is unique in that it takes into account a wide range of factors that contribute to a country's development, rather than just economic growth. This allows for a more accurate picture of a country's well-being and potential for future growth. Additionally, the HDI is widely recognized and used by policymakers, making it a valuable tool for cross-country comparisons and identifying best practices for development.

Mathematical implications The HDI is a composite index calculated by taking into account three dimensions of development: health, education, and income. Each dimension is weighted based on its importance to overall development. The mathematical implications of the HDI depend on the specific indicators used and the weights assigned to each dimension.

Mathematical Formula The formula for the HDI is calculated as follows:

HDI = (Health Index + Education Index + Income Index) / 3

Where:

Health Index = (Life expectancy - 20) / (85 - 20) Education Index = (Expected years of schooling - 0) / (15 - 0) Income Index = ln(Gross National Income per capita) - ln($100)

Implementation The HDI is typically calculated by the United Nations Development Programme (UNDP) and is updated annually. Data for the HDI is collected from various sources, including government statistics, international organizations, and academic research.

Types if present and their description There are several variations of the HDI, including the Gender Development Index (GDI) and the Multidimensional Poverty Index (MPI). The GDI measures gender-based inequalities in health, education, and income, while the MPI measures poverty based on several indicators related to health, education, and standard of living.

Derived form where The HDI has not been derived from another index or measure.

Conclusion In conclusion, the Human Development Index is a valuable tool for measuring a country's development and well-being. By taking into account various factors related to health, education, and income, the HDI provides a more accurate and comprehensive view of a country's progress than traditional measures like GDP. The HDI is widely recognized and used by policymakers, making it a valuable tool for cross-country comparisons and identifying best practices for development.

Global Innovation Index

The Human Development Index (HDI) is a composite statistical measure of human development that was created by the United Nations Development Programme (UNDP) in 1990. It is used to rank countries according to their level of human development based on three dimensions: health, education, and living standards.

The HDI is used to measure progress towards achieving the United Nations Sustainable Development Goals, particularly Goal 3: Good Health and Well-being, Goal 4: Quality Education, and Goal 8: Decent Work and Economic Growth. The index provides a comprehensive picture of a country's development status and helps policymakers identify areas for improvement.

One of the benefits of using the HDI is that it takes into account not only economic factors but also social factors such as education and health. It provides a more comprehensive and nuanced view of human development than traditional measures of economic development such as Gross Domestic Product (GDP). Additionally, the HDI is useful for comparing countries with different levels of economic development.

The HDI is better than other similar algorithms because it takes into account multiple dimensions of development rather than focusing solely on economic indicators. It also allows for comparison between countries with vastly different economic situations, providing a more accurate picture of overall development.

The mathematical formula for calculating the HDI is as follows:

HDI = (1/3) * (Health Index + Education Index + Standard of Living Index)

The Health Index is calculated using life expectancy at birth, while the Education Index is calculated using a combination of mean years of schooling and expected years of schooling. The Standard of Living Index is calculated using Gross National Income (GNI) per capita, adjusted for purchasing power parity.

The HDI is implemented by collecting data from national statistical agencies and other sources. The data is then used to calculate the index for each country.

There are several types of HDI, including the Inequality-Adjusted HDI (IHDI), which takes into account inequality within a country in addition to the three dimensions of the HDI. The IHDI provides a more accurate picture of a country's development by taking into account differences in access to education, healthcare, and income.

The HDI is derived from the work of economist Amartya Sen, who argued that development should be measured not just by economic indicators but also by social indicators such as health and education.

In conclusion, the HDI is an important tool for measuring human development and progress towards achieving the United Nations Sustainable Development Goals. Its multidimensional approach provides a more comprehensive view of development than traditional measures of economic growth, and it allows for comparison between countries with vastly different economic situations. The HDI is continually evolving, with new types and methodologies being developed to improve its accuracy and usefulness.

Classification and Regression Trees:

What is Classification and Regression Trees (CART)?

Classification and Regression Trees (CART) is a type of decision tree algorithm used in data mining and machine learning for classification and prediction tasks. CART constructs a binary tree of decisions and conditions that are used to predict the target variable based on the input variables.

Why is CART used?

CART is used for both classification and regression tasks. It is particularly useful for exploratory data analysis and generating hypotheses for further research. CART can also be used for predictive modeling and decision-making applications, such as determining credit risk or identifying potential customers.

Benefits of using CART

Some of the benefits of using CART include:

1. Easy to interpret: CART produces a tree-like structure that is easy to visualize and interpret.
2. Non-parametric: CART does not make any assumptions about the distribution of the data or the relationship between the input and target variables.
3. Handles mixed data types: CART can handle a mix of continuous and categorical data types.
4. Handles missing data: CART can handle missing data by using surrogate splits.
5. Efficient: CART is computationally efficient and can handle large datasets.

Why is CART better than other similar algorithms?

CART has some advantages over other similar algorithms, such as:

1. Flexibility: CART can handle both categorical and continuous data, whereas some other decision tree algorithms only work with one type of data.
2. Robustness: CART is less sensitive to outliers than some other machine learning algorithms.
3. Adaptability: CART can handle different types of data, such as binary, nominal, and continuous data.

Mathematical implications

CART uses a greedy algorithm to recursively split the dataset into smaller subsets based on the input variables. The splitting process is based on the reduction in impurity, which is measured by the Gini index or entropy.

Mathematical formula

The mathematical formula for CART is:

1. For classification problems, the Gini index is used to measure the impurity of a split. The Gini index is defined as:

G=∑i=1cpi(1−pi)G=∑i=1cpi(1−pi)

where c is the number of classes, and p_i is the proportion of instances in class i.

2. For regression problems, the mean squared error (MSE) is used to measure the impurity of a split. The MSE is defined as:

MSE=1n∑i=1n(yi−yi^)2MSE=n1∑i=1n(yi−yi^)2

where n is the number of instances, y_i is the observed value of the target variable, and $\hat{y_i}$ is the predicted value.

Implementation

CART can be implemented in various programming languages, such as R, Python, and SAS. There are also several open-source and commercial software packages that provide CART algorithms, such as the R package rpart and the SAS procedure PROC TREE.

Types if present and their description

There are two types of CART:

1. Classification trees: used for classification tasks where the target variable is categorical.
2. Regression trees: used for regression tasks where the target variable is continuous.

Derived form where

CART is derived from decision trees, which are a type of machine learning algorithm that uses a tree-like structure to represent a set of decisions and their possible consequences.

Conclusion

CART is a versatile and easy-to-use algorithm that can handle a wide range of data types and is suitable for both classification and regression tasks. Its ability to handle mixed data types, missing data, and outliers makes it a popular choice for exploratory data analysis and predictive modeling applications.

Predictive Mean Matching

What is Predictive Mean Matching? Predictive Mean Matching (PMM) is a technique used in missing data imputation. It is a multiple imputation method that is commonly used to fill in missing values in continuous data when the missing data is assumed to be missing at random. In PMM, the imputation is done by finding the observed value closest to the predicted value from a model and substituting that value in place of the missing value.

Why is PMM used? Missing data is a common problem in statistical analysis, and it can lead to biased results if not handled properly. PMM is used to impute missing values in continuous data when the missing data is assumed to be missing at random. It is a popular imputation method because it is simple, fast, and can be easily implemented in most statistical software packages.

Benefits of using PMM:

- PMM produces imputations that are valid and efficient.

- PMM imputations tend to be more accurate than other imputation methods, particularly when the missingness is related to the observed data.
- PMM produces imputations that are unbiased and consistent under a variety of conditions.
- PMM can handle missing data in both continuous and categorical variables.

Why is PMM better than other similar algorithms? PMM is a popular imputation method because it is easy to implement and generally performs well in practice. It is particularly useful when the missing data is related to the observed data. Other similar algorithms, such as regression imputation or mean imputation, may not perform as well in these situations.

Mathematical implications: PMM is based on the idea of finding the observed value closest to the predicted value from a model and using that value to impute the missing value. The algorithm involves the following steps:

1. Fit a model to the observed data.
2. Predict the missing values using the model.
3. For each missing value, find the observed value closest to the predicted value.
4. Substitute the observed value in place of the missing value.

Mathematical Formula: There is no single mathematical formula for PMM, as it is implemented differently depending on the software package and the specific model being used.

Implementation: PMM can be implemented in most statistical software packages, including R, SAS, and Stata. In R, PMM can be implemented using the mice package.

Types if present and their description: There are different variations of PMM, such as fractional imputation, where the probability of observing the missing data is estimated and incorporated into the imputation model. Another variation is robust PMM, which is used to handle outliers in the data.

Derived form where: PMM is derived from the more general concept of multiple imputation, which involves generating multiple imputed datasets and using them to obtain estimates of the missing data.

Conclusion: Predictive Mean Matching is a powerful imputation method that can be used to fill in missing values in continuous data when the missing data is assumed to be missing at random. It produces imputations that are valid and efficient, and tend to be more accurate than other imputation methods in certain situations. PMM is easy to implement in most statistical software packages and can handle missing data in both continuous and categorical variables.

Fuzzy Clustering

What is Fuzzy Clustering? Fuzzy clustering is a data clustering technique that assigns a data point to one or more clusters, based on its degree of membership in each cluster. Unlike traditional clustering techniques that assign a data point to only one cluster, fuzzy clustering allows for the possibility of partial membership in multiple clusters. The degree of membership is represented by a value between 0 and 1, where 0 means the data point is not a member of the cluster, and 1 means the data point is a full member of the cluster.

Why is it used? Fuzzy clustering is used in situations where data points do not clearly belong to one cluster or the other, but rather belong to multiple clusters to varying degrees. It is particularly useful when dealing with complex data that cannot be easily classified into discrete groups, such as images, text, and sensor data.

Benefits of using it: Fuzzy clustering has several benefits, including:

1. Flexibility: Fuzzy clustering is flexible in that it allows for partial membership in multiple clusters, which is often more realistic than hard clustering techniques that assign a data point to only one cluster.
2. Robustness: Fuzzy clustering is robust to noise and outliers in the data, as it accounts for the uncertainty in the data by allowing for partial membership in multiple clusters.
3. Interpretability: Fuzzy clustering provides more information than traditional clustering techniques, as it not only identifies the clusters to which a data point belongs, but also the degree of membership in each cluster.

Why better than other similar algorithms: Fuzzy clustering is often preferred over traditional clustering techniques because it provides a more nuanced view of the data by allowing for partial membership in multiple clusters. Other similar algorithms, such as K-means clustering, do not allow for partial membership and can be more rigid in their classification of data points.

Mathematical implications: Fuzzy clustering is based on fuzzy set theory, which allows for partial membership in a set. In fuzzy clustering, each data point is assigned a membership grade to each cluster, which is represented by a value between 0 and 1. The membership grade indicates the degree to which the data point belongs to the cluster.

Mathematical Formula: The mathematical formula for fuzzy clustering is as follows: Let X = {x1, x2, ..., xn} be the set of n data points. Let C = {c1, c2, ..., cm} be the set of m clusters. Let uij be the degree of membership of data point xi in cluster cj. Then, the objective function of fuzzy clustering is given by:

minimize: J = ∑i=1 to n ∑j=1 to m (uij)^m ||xi-cj||^2

subject to: ∑j=1 to m uij = 1 for all i, 0 <= uij <= 1 for all i and j.

The parameter m is a tuning parameter that controls the degree of fuzziness in the clustering. A larger value of m results in a more fuzzy clustering, while a smaller value of m results in a more rigid clustering.

Algorithm:

Input:

- A set of n data points, D = {x1, x2, ..., xn}
- The number of clusters k
- A fuzzifier parameter m
- The maximum number of iterations
- A convergence threshold
- An initial set of k centroids {c1, c2, ..., ck}

Output:

- A set of k centroids {c1, c2, ..., ck}
- A fuzzy partition matrix U, where uij represents the degree of membership of data point i to cluster j

Steps:

1. Initialize the fuzzy partition matrix U with random values between 0 and 1, where each row sums up to 1.
2. Calculate the distance between each data point and each centroid using a distance metric such as Euclidean distance.
3. Calculate the degree of membership of each data point to each cluster using the following formula:

   uij = 1 / (∑k=1 to c (dij/dkj)^(2/(m-1)))

   where dij is the distance between data point i and centroid j, dkj is the distance between data point k and centroid j, and m is the fuzzifier parameter.

4. Update the centroids of each cluster using the following formula:

   cj = (∑i=1 to n (uij)^m * xi) / (∑i=1 to n (uij)^m)

   where xi is the ith data point and j is the cluster.

5. Calculate the objective function, which measures the quality of the clustering, using the following formula:

   J(U,C) = ∑i=1 to n ∑j=1 to k (uij)^m * dij^2

6. Check if the convergence criteria is met, if not, go back to step 2.
7. Return the set of k centroids and the fuzzy partition matrix U.

Implementation: Fuzzy clustering can be implemented using various software packages, such as MATLAB, Python's scikit-fuzzy, and R's fuzzyclus. The implementation involves specifying the number of clusters and the value of the tuning parameter m, and then applying an iterative algorithm, such as the Fuzzy C-Means (FCM) algorithm, to optimize the objective function.

Types of Fuzzy Clustering:

1. Fuzzy C-Means (FCM): It is the most common and widely used fuzzy clustering algorithm. It aims to minimize the sum of squared distances between each data point and the cluster center, with membership degrees indicating the degree of belongingness of each point to each cluster.
2. Gustafson-Kessel (GK): This algorithm uses the Mahalanobis distance instead of the Euclidean distance used in FCM. It is useful when the data has different variances and covariances.
3. Possibilistic C-Means (PCM): In this algorithm, the membership degrees indicate the degree of possibility of each point belonging to a cluster, rather than the degree of belongingness as in FCM.
4. Fuzzy-Expectation-Maximization (FEM): This algorithm is a fuzzy clustering version of the Expectation-Maximization algorithm, which is commonly used in statistical inference and unsupervised learning. It is useful for dealing with missing or incomplete data.

Derived form of Fuzzy Clustering:

Fuzzy clustering can be used in a wide range of applications, including image segmentation, pattern recognition, data mining, and bioinformatics. One derived form of fuzzy clustering is called Fuzzy Decision Trees (FDT), which combine the advantages of fuzzy clustering and decision trees. FDT can handle fuzzy data and generate fuzzy rules for decision making.

Conclusion:

Fuzzy clustering is a powerful unsupervised learning technique that is widely used in various applications. It can handle uncertain and incomplete data, and provide valuable insights into the structure of the data. Different types of fuzzy clustering algorithms have their own strengths and weaknesses, and the choice of algorithm depends on the specific application and data characteristics. The derived form of fuzzy clustering, such as Fuzzy Decision Trees, can provide additional benefits and insights into decision making.

Gaussian Mixture Model

What is Gaussian Mixture Model (GMM)? Gaussian Mixture Model (GMM) is a probabilistic model used to represent the distribution of data points in a dataset. It is a type of unsupervised learning algorithm that assumes the data points are generated from a mixture of several Gaussian distributions with unknown parameters. GMM is a type of clustering algorithm that clusters data points based on their similarity in distribution.

Why it is used? GMM is used for several purposes, including:

1. Clustering: GMM is used to cluster data points into different groups based on their similarity in distribution.
2. Density estimation: GMM is used to estimate the probability density function of the data points in a dataset.
3. Outlier detection: GMM is used to identify the outliers in a dataset based on their probability density.

Benefits of using GMM:

1. Flexibility: GMM can model complex distributions with multiple modes and can capture the underlying structure of the data.
2. Scalability: GMM can handle large datasets and is computationally efficient.
3. Unsupervised learning: GMM can learn from data without the need for labels, making it suitable for unsupervised learning tasks.

Why is GMM better than other similar algorithms?

1. Can model complex distributions: GMM can model complex distributions with multiple modes, which is not possible with algorithms such as K-means.
2. Probabilistic output: GMM provides a probabilistic output, which is useful for understanding the uncertainty associated with the clustering results.

Mathematical implications: GMM assumes that the data points are generated from a mixture of Gaussian distributions. The parameters of the GMM include the means and variances of the Gaussian distributions and the mixing coefficients that specify the probability of each Gaussian distribution generating a data point.

Mathematical formula: The probability density function of a GMM is given by:

$p(x) = \Sigma k=1 \text{ to } K (\pi k * N(x \mid \mu k, \Sigma k))$

where πk is the mixing coefficient of the k-th Gaussian distribution, $N(x \mid \mu k, \Sigma k)$ is the probability density function of the k-th Gaussian distribution, and K is the number of Gaussian distributions.

Algorithm:
Input:

- A set of n data points, D = {x1, x2, ..., xn}
- The number of clusters k
- The maximum number of iterations
- A convergence threshold
- An initial set of k Gaussian distributions {G1, G2, ..., Gk}, each with a mean, covariance matrix, and weight

Output:

- A set of k Gaussian distributions {G1, G2, ..., Gk}, each with a mean, covariance matrix, and weight
- A set of k probabilities {p1, p2, ..., pk} for each cluster

Steps:

1. Initialize the parameters of k Gaussian distributions {G1, G2, ..., Gk} with random values, or using some initialization method.
2. Calculate the probability of each data point belonging to each Gaussian distribution using Bayes' theorem.
3. Calculate the posterior probability of each Gaussian distribution given the data points and the current parameters.
4. Update the mean, covariance matrix, and weight of each Gaussian distribution based on the posterior probability using the maximum likelihood estimation.
5. Check if the convergence criteria is met, if not, go back to step 2.
6. Assign each data point to the Gaussian distribution with the highest posterior probability.
7. Return the set of k Gaussian distributions and their respective probabilities.

Implementation: GMM can be implemented using algorithms such as Expectation-Maximization (EM) algorithm, Variational Bayes (VB) algorithm, and Gibbs sampling algorithm.

Types if present and their description:

1. Diagonal covariance GMM: In this type of GMM, the covariance matrices of the Gaussian distributions are diagonal, which means that the features of the data points are assumed to be independent.
2. Full covariance GMM: In this type of GMM, the covariance matrices of the Gaussian distributions are full, which means that the features of the data points are assumed to be correlated.

Derived form where: GMM can be extended to other models such as Hidden Markov Model (HMM), where the GMM is used to model the emission probability distribution of the HMM.

Conclusion: Gaussian Mixture Model (GMM) is a powerful unsupervised learning algorithm that can model complex distributions and cluster data points based on their similarity in distribution. It has several benefits, including flexibility, scalability, and the ability to handle large datasets. GMM can be implemented using algorithms such as Expectation-Maximization (EM) algorithm, and it has different types such as diagonal covariance GMM and full covariance GMM.

K-Means

What is k-means? k-means is a popular unsupervised learning algorithm used for clustering data. The algorithm is based on the concept of partitioning a dataset into a fixed number of clusters (k), where each cluster represents a group of similar data points. The algorithm works iteratively by first initializing k centroids randomly and assigning each data point to the closest centroid. The centroids are then updated by computing the mean of all data points assigned to the cluster. The process is repeated until the centroids no longer change, or the maximum number of iterations is reached.

Why is it used? k-means is commonly used for a variety of applications, including customer segmentation, image segmentation, and anomaly detection. It is a useful tool for data exploration and understanding, as it can uncover patterns and relationships in complex datasets that may not be visible through manual inspection. Additionally, it can be used as a pre-processing step for supervised learning algorithms, as clustering can help identify features that are most relevant for classification.

Benefits of using it Some of the benefits of using k-means include its simplicity, scalability, and efficiency. The algorithm is easy to understand and implement, making it accessible to users with varying levels of expertise. Additionally, it is highly scalable, meaning it can handle large datasets with millions of data points. Furthermore, the algorithm is computationally efficient, making it suitable for real-time applications where speed is critical.

Why better than other similar algorithms k-means has several advantages over other clustering algorithms. One of the main advantages is its simplicity, as it is easy to understand and implement. Additionally, it is highly scalable and efficient, making it suitable for large datasets and real-time applications. Furthermore, k-means is highly interpretable, as it produces clusters that are easily understandable and can be visualized.

Mathematical implications The mathematical foundations of k-means are rooted in optimization theory and linear algebra. The algorithm attempts to minimize the sum of squared distances between each data point and its assigned centroid. This objective function can be solved using numerical optimization techniques such as gradient descent or the Lloyd's algorithm.

Algorithm:
Input:

- A set of n data points, D = {p1, p2, ..., pn}
- The number of clusters k
- A distance measure, d(p, q), to compute the distance between two data points p and q

Output:

- A set of k clusters with their centroids

Steps:

1. Initialize k centroids by randomly selecting k data points from D as the initial centroids.
2. Assign each data point in D to the closest centroid based on the chosen distance metric.
3. Recalculate the centroids of each cluster as the mean of all data points assigned to that cluster.
4. Repeat steps 2-3 until the centroids no longer change or a maximum number of iterations is reached.

Implementation k-means can be implemented using a variety of programming languages and libraries, including Python's scikit-learn, MATLAB, and R. The algorithm requires tuning of hyperparameters, such as the number of clusters (k) and the maximum number of iterations.

Types if present and their description There are several variations of k-means, including:

1. K-Medoids: Instead of using the mean of the data points in a cluster to update the centroid, K-Medoids uses the median point as the centroid. This approach is more robust to outliers than k-means.
2. Mini-Batch K-Means: This variation of k-means uses a random subset of the data points (a mini-batch) to update the centroids at each iteration. This makes the algorithm more computationally efficient and allows for real-time clustering of large datasets.
3. Hierarchical K-Means: This approach uses a hierarchical clustering algorithm to determine the number of clusters and their structure. The algorithm starts with each data point as its own cluster and then merges clusters based on their distance.

Derived forms of k-means algorithm include:

1. K-medoids: K-medoids is a variant of k-means that uses medoids (i.e., data points) as cluster centers instead of means. Medoids are chosen as the data points that minimize the sum of dissimilarities between them and other points in the same cluster. This makes K-medoids more robust to outliers than k-means.
2. Fuzzy k-means: Fuzzy k-means is a variant of k-means that allows data points to belong to multiple clusters with different degrees of membership. Instead of assigning each data point to a single cluster, fuzzy k-means assigns each point a membership grade for each cluster. This makes fuzzy k-means more flexible than k-means, but also more computationally intensive.
3. Hierarchical k-means: Hierarchical k-means is a hierarchical clustering algorithm that uses k-means as its base clustering method. The algorithm starts with each data point as its own cluster and then recursively merges clusters until a stopping criterion is met. This results in a hierarchical tree-like structure that can be cut at different levels to obtain different numbers of clusters.

Conclusion: K-means is a popular and widely used clustering algorithm that is simple, efficient, and effective for many clustering tasks. Its ability to handle large datasets and its interpretability make it a valuable tool in data analysis and machine learning. However, it is important to choose the appropriate number of clusters and to pre-process the data carefully to obtain good results with k-means. In addition, there are several variants of k-means that can be used to handle different types of data or to obtain more flexible or robust clustering results.

K-Medoid

What is k-medoids?

K-medoids is a clustering algorithm that partitions a dataset into k number of clusters. The algorithm is similar to the k-means algorithm, but instead of computing the mean of each cluster, it chooses a representative point of each cluster, called a medoid. The medoid is the data point that minimizes the sum of distances to all other points in the cluster.

Why is it used?

K-medoids is used to identify patterns in data and group similar items together. It is particularly useful when the dataset is small or when the data is not normally distributed, as it can handle non-continuous and non-linear data.

Benefits of using it

K-medoids has several benefits:

1. It can handle non-linear and non-continuous data.

2. It is less sensitive to outliers than k-means.
3. It can be more computationally efficient than k-means for small datasets.
4. It provides a representative point for each cluster, which can be useful in interpretation and visualization of the results.

Why better than other similar algorithm

Compared to k-means, k-medoids is more robust to outliers and noise in the data. It also provides a representative point for each cluster, which can be useful in interpretation and visualization of the results. However, k-means is generally faster and more scalable than k-medoids for large datasets.

Algorithm:
Input:

- A set of n data points, D = {p1, p2, ..., pn}
- The number of clusters k
- A distance measure, d(p, q), to compute the distance between two data points p and q

Output:

- A set of k clusters with their medoids

Steps:

1. Initialize k medoids by randomly selecting k data points from D as the initial medoids.
2. Assign each data point in D to the closest medoid based on the chosen distance metric.
3. Compute the total dissimilarity of all data points to their assigned medoids, which is the objective function to be minimized.
4. For each medoid m: a. For each non-medoid data point p that belongs to the cluster of m: i. Compute the total dissimilarity of all data points to p as the new total dissimilarity of the cluster. ii. If the new total dissimilarity is less than the current total dissimilarity of the cluster, swap m and p and update the total dissimilarity of the cluster.
5. Repeat steps 2-4 until the assignment of data points to medoids no longer changes or a maximum number of iterations is reached.

Implementation

K-medoids can be implemented in various programming languages, such as Python and R. There are also several packages available that implement the algorithm, such as scikit-learn in Python and cluster in R.

Types if present and their description

There are several variations of the k-medoids algorithm, including:

1. Partitioning Around Medoids (PAM): This is the most common variation of k-medoids and is based on the algorithm described above.
2. CLARA: This variation uses multiple random samples of the dataset to compute the medoids and then assigns each point to its closest medoid across all samples.
3. CLARANS: This variation is similar to CLARA, but uses a more efficient search strategy to find the optimal medoids.

Derived form where

K-medoids is derived from k-means, which is a popular clustering algorithm that partitions a dataset into k number of clusters by minimizing the sum of squared distances between each point and its assigned centroid.

In conclusion, k-medoid is a clustering algorithm that is used to group similar data points together. It is particularly useful in situations where the distance metric used to determine the similarity between data points is non-Euclidean or the data contains noise or outliers. K-medoid is a robust algorithm that is less sensitive to outliers than k-means, and it is also computationally efficient. It has several applications in various fields, including image analysis, bioinformatics, and marketing. Overall, k-medoid is a powerful clustering algorithm that is widely used due to its simplicity, efficiency, and robustness.

Hierarchical Clustering

What is hierarchical clustering:

Hierarchical clustering is a type of clustering algorithm used to group similar data points together. It involves creating a hierarchy of clusters where smaller clusters are nested within larger clusters. The algorithm works by starting with each data point as its own cluster and then merging the closest pairs of clusters until all the data points are in a single cluster.

Why it is used:

Hierarchical clustering is used in a variety of applications such as market segmentation, image processing, and bioinformatics. It is particularly useful when the data being analyzed does not have a predetermined number of clusters or when the relationships between data points are complex and difficult to define.

Benefits of using it:

One of the main benefits of hierarchical clustering is that it provides a visual representation of the data in the form of a dendrogram, which can be useful in identifying the optimal number of clusters. It also allows for the identification of nested clusters, which can reveal underlying structures in the data.

Why better than other similar algorithm:

Hierarchical clustering is often preferred over other clustering algorithms such as k-means because it does not require the number of clusters to be specified in advance, and can handle non-linear relationships between data points.

Mathematical implications:

Hierarchical clustering involves measuring the similarity between data points using a distance metric such as Euclidean distance, and then merging the closest pairs of clusters based on the distance between their centroids. The algorithm continues until all the data points are in a single cluster.

Mathematical Formula:

There are different ways to calculate the distance between data points, such as the single linkage, complete linkage, and average linkage methods. The most commonly used method is the complete linkage method, which calculates the distance between clusters as the maximum distance between any two points in the clusters.

Implementation:

Hierarchical clustering can be implemented using various software packages such as R, Python, and MATLAB. The algorithm can also be customized to incorporate different distance metrics and linkage methods based on the needs of the analysis.

Algorithm:
Input:

- A set of n data points, D = {p1, p2, ..., pn}
- A distance measure, d(p, q), to compute the distance between two data points p and q
- A linkage criterion to measure the distance between clusters, such as single linkage, complete linkage, or average linkage

Output:

- A dendrogram or a set of clusters

Steps:

1. Create n clusters, one for each data point.

2. Compute the pairwise distance matrix between all data points using the chosen distance metric.
3. While there is more than one cluster: a. Compute the pairwise distance matrix between all clusters using the chosen linkage criterion. b. Merge the two closest clusters into a new cluster. c. Update the distance matrix to include the new cluster and its distances to all other clusters.
4. Stop when there is only one cluster left, or when the desired number of clusters has been reached.

Types if present and their description:

There are two main types of hierarchical clustering: agglomerative and divisive. Agglomerative clustering is the most commonly used and involves starting with each data point as its own cluster and then merging the closest pairs of clusters until all the data points are in a single cluster. Divisive clustering involves starting with all the data points in a single cluster and then recursively dividing the cluster into smaller sub-clusters.

Derived form where:

Hierarchical clustering is derived from the field of statistics and has been widely used in various disciplines such as computer science, biology, and social sciences.

Conclusion:

Hierarchical clustering is a powerful tool for exploring and analyzing complex datasets. Its ability to identify nested clusters and provide a visual representation of the data makes it a popular choice for data scientists and researchers. However, its performance can be affected by the choice of distance metric and linkage method, and it may not be suitable for very large datasets.

DBSCAN

What is it? Density-Based Spatial Clustering of Applications with Noise (DBSCAN) is a clustering algorithm used in machine learning for discovering clusters of data points in a dataset. It is particularly useful for spatial databases or data with noise and irregularities.

Why is it used? DBSCAN is used to identify clusters of data points that are close to each other based on their distance or density. It can also identify outliers, which are data points that do not belong to any cluster. DBSCAN is often used in fields such as image analysis, computer vision, and anomaly detection.

Benefits of using it: DBSCAN has several advantages over other clustering algorithms, including:

1. It does not require specifying the number of clusters in advance, making it more flexible than other algorithms.
2. It can handle clusters of different shapes and sizes.
3. It can identify noise and outliers, which can be useful in many applications.

Why better than other similar algorithm? DBSCAN is often preferred over other clustering algorithms such as K-means or hierarchical clustering because it can handle data with noise and outliers more effectively. It can also identify clusters of different shapes and sizes, which other algorithms may struggle with.

Mathametical implications: DBSCAN uses the concepts of density and distance to identify clusters in a dataset. It uses two parameters: epsilon (ε), which defines the radius around a data point to search for other data points, and MinPts, which defines the minimum number of data points required to form a cluster.

Mathematical Formula: DBSCAN uses the following formulas:

1. Epsilon (ε): Defines the radius around a data point to search for other data points.
2. MinPts: Defines the minimum number of data points required to form a cluster.
3. Core points: Data points that have at least MinPts data points within a distance of ε.
4. Border points: Data points that are within ε distance of a core point but have less than MinPts data points within ε.
5. Noise points: Data points that are neither core nor border points.

Implementation: DBSCAN can be implemented using various programming languages such as Python, R, Java, and MATLAB. There are also several libraries available for implementing DBSCAN, such as scikit-learn in Python and fpc in R.

Types if present and their description: There are two types of DBSCAN:

1. Epsilon DBSCAN: This type of DBSCAN uses a fixed value of ε for all data points.
2. Variable-Radius DBSCAN: This type of DBSCAN uses a variable ε based on the density of data points.

Algorithm:
Input:

- A set of n data points, D = {p1, p2, ..., pn}
- Epsilon (ε): maximum distance between two points to be considered as neighbors
- Minimum points (MinPts): minimum number of points needed to form a dense region

Output:

- A set of clusters C1, C2, ..., Ck and noise points

Steps:

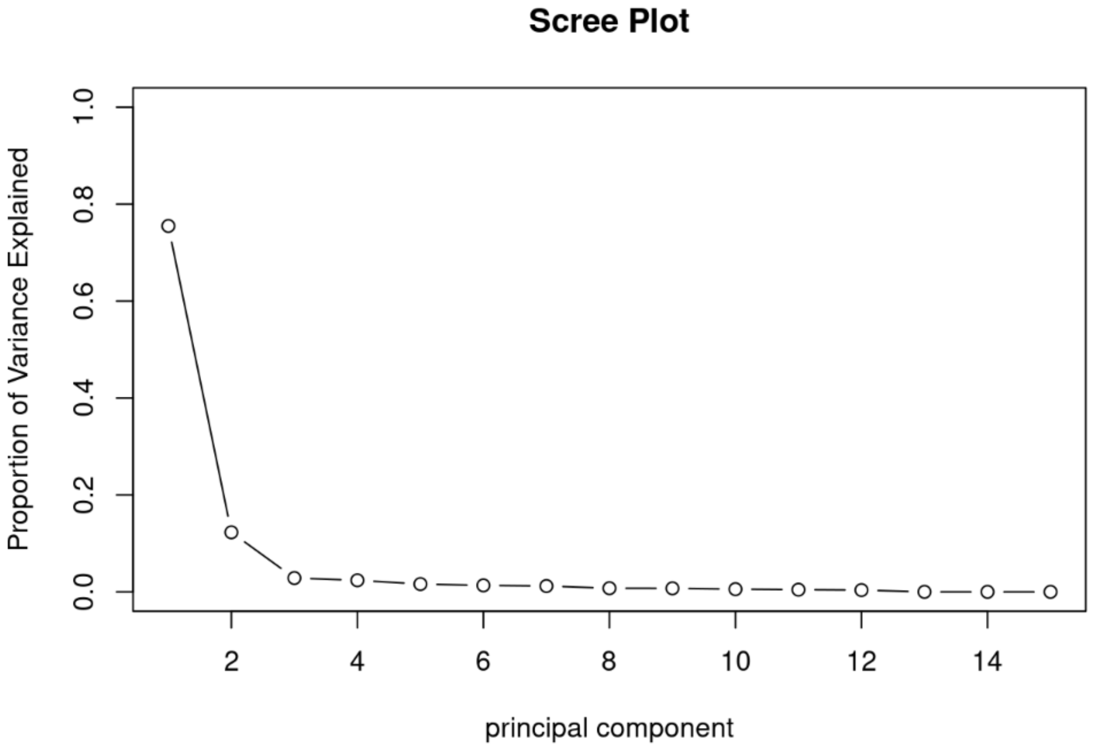
1. Randomly select a data point p from D.
2. Retrieve all points within ε distance from p to form a neighborhood N.
3. If the number of points in N is less than MinPts, mark point p as noise and go to step 7.
4. Otherwise, mark p as a core point and add p and all the points in N to a new cluster Ci.
5. For each point q in N that is not already assigned to a cluster, retrieve all points within ε distance from q to form a new neighborhood Nq.
6. If the number of points in Nq is greater than or equal to MinPts, add all the points in Nq to cluster Ci. If q is a core point, this process will be recursively repeated for its neighborhood.
7. Repeat steps 1-6 until all points have been either assigned to a cluster or marked as noise.

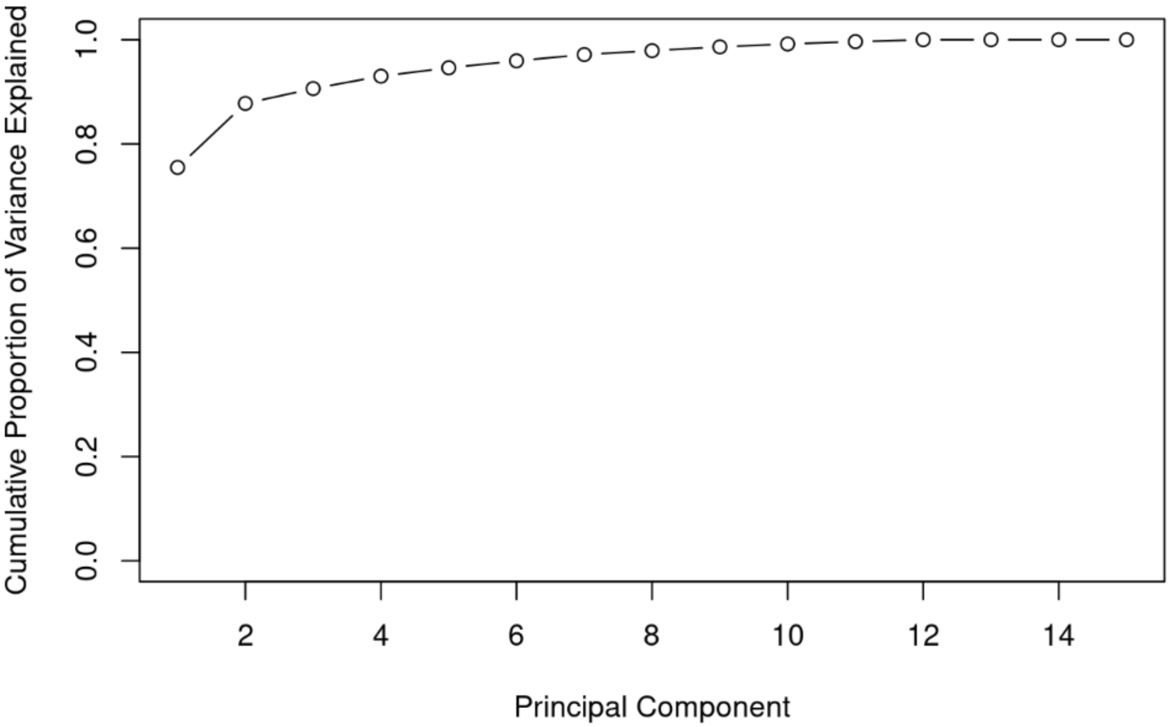
Derived form where: DBSCAN has several derived forms such as Hierarchical DBSCAN, Distributed DBSCAN, and OPTICS.

Conclusion: DBSCAN is a powerful clustering algorithm that can handle data with noise and outliers effectively. It does not require specifying the number of clusters in advance, making it more flexible than other clustering algorithms. DBSCAN can be implemented using various programming languages and libraries, and there are several derived forms of DBSCAN available.

RESULTS AND DISCUSSION

Analysis for Social Progress score dataset:

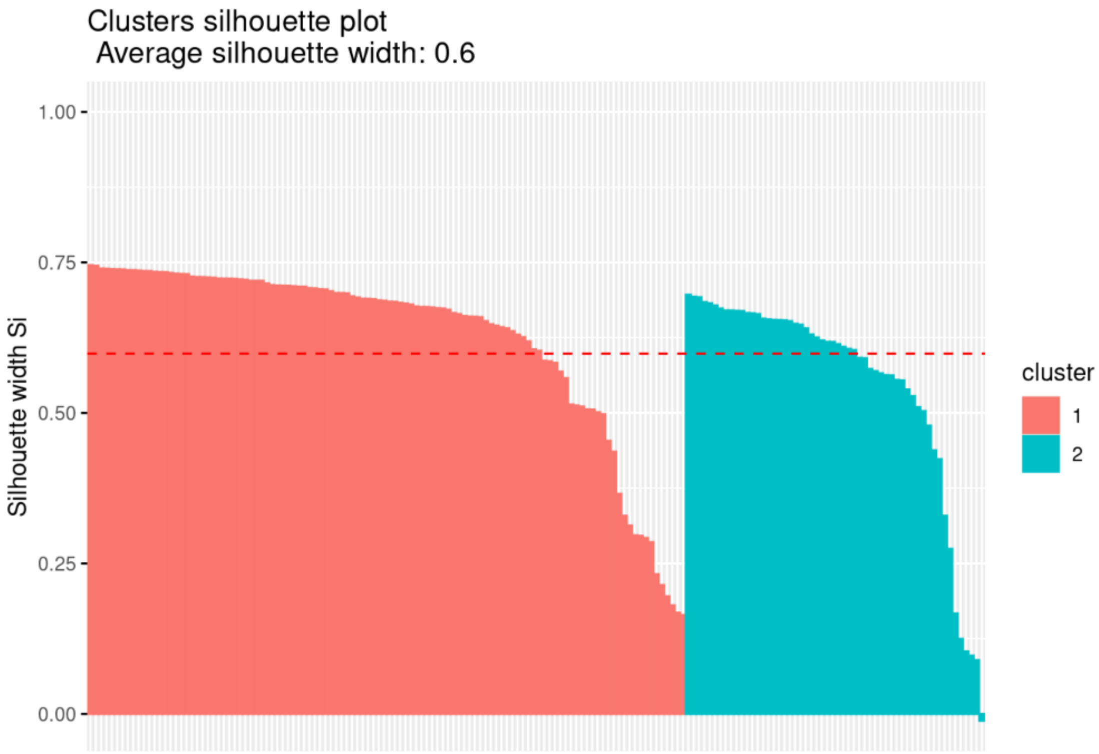


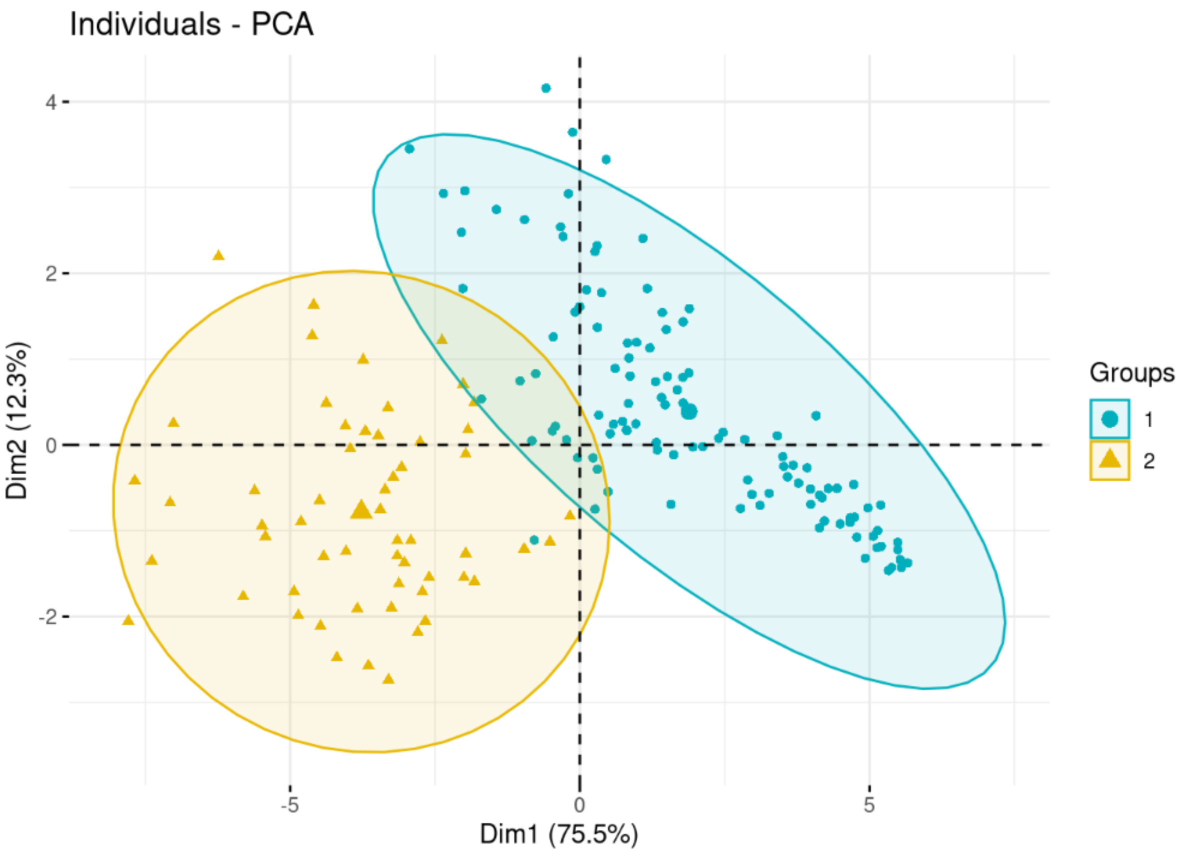


Inference:

The above Scree plots explains that 90-100% of cumulative proportion of variance is being explained by the dataset of Global Innovation scores which means that our dataset is good.

Plots based on Clustering:

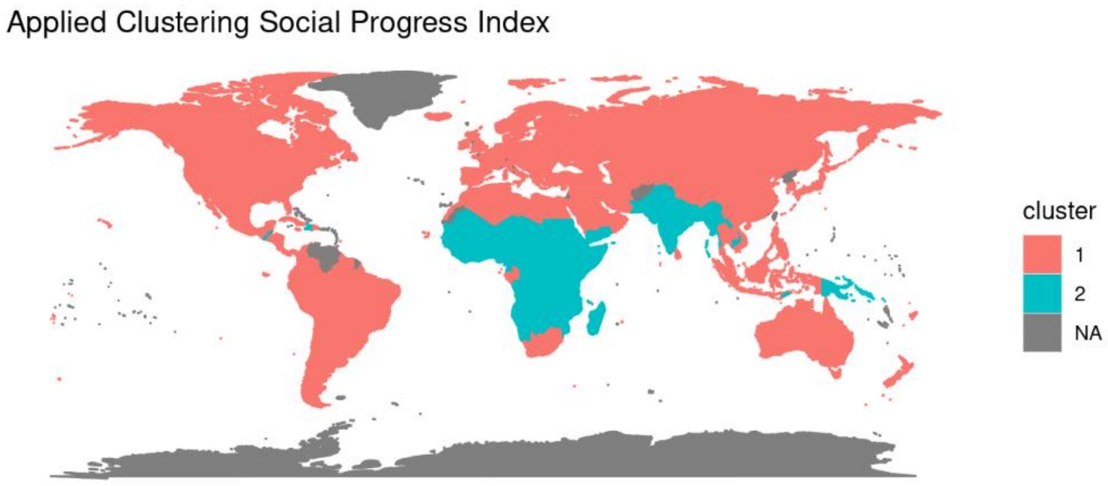

Clusters silhouette plot
Average silhouette width: 0.6
Silhouette width Si
1.00
0.75
0.50
0.25
0.00
cluster
1
2



Inference:

The cluster silhouette plot above shows a silhouette width of 0.6 which explains that the clusters obtained by us are good and they don’t overlap with each other and are uniformly separated. No. of clusters formed = 2.

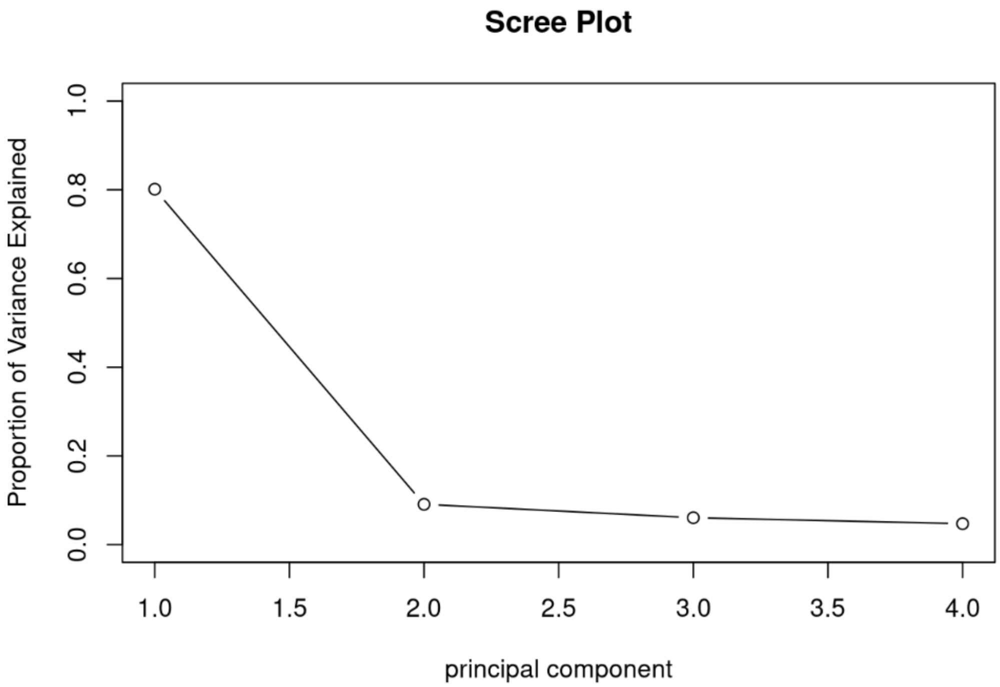
World map:

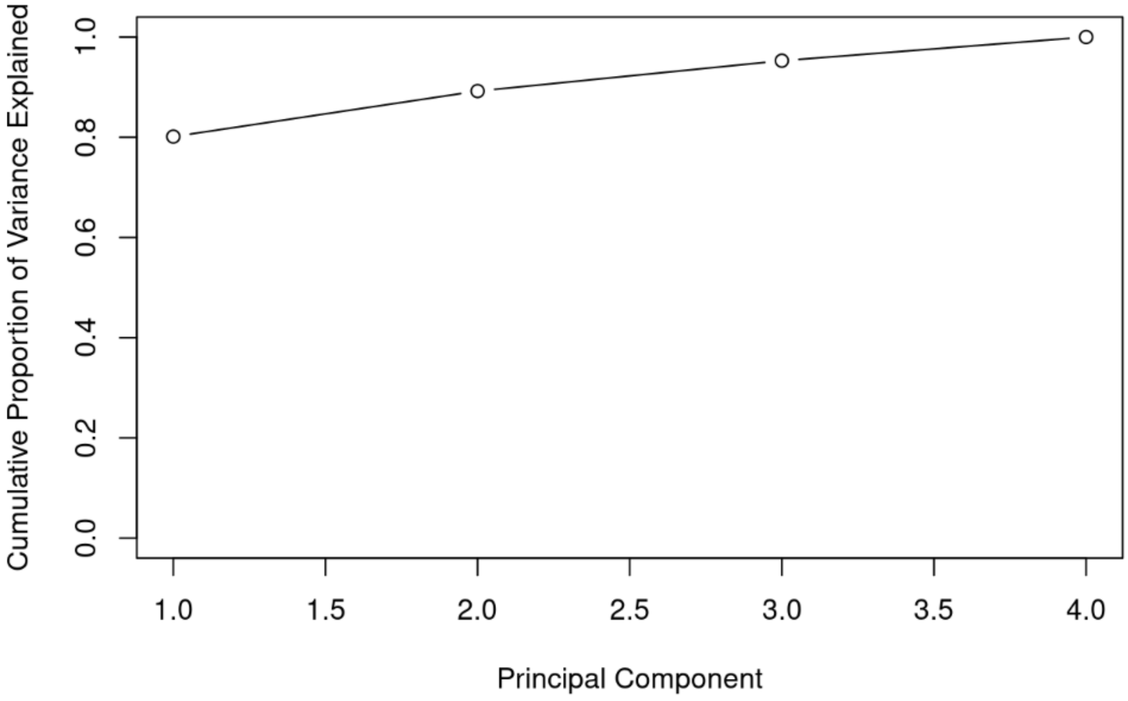


Inference:

The clusters 1 and 2 as clearly visible from the above figure shows that the clusters are uniformly separated among one another in the globe.

Analysis for Human Development Index scores:

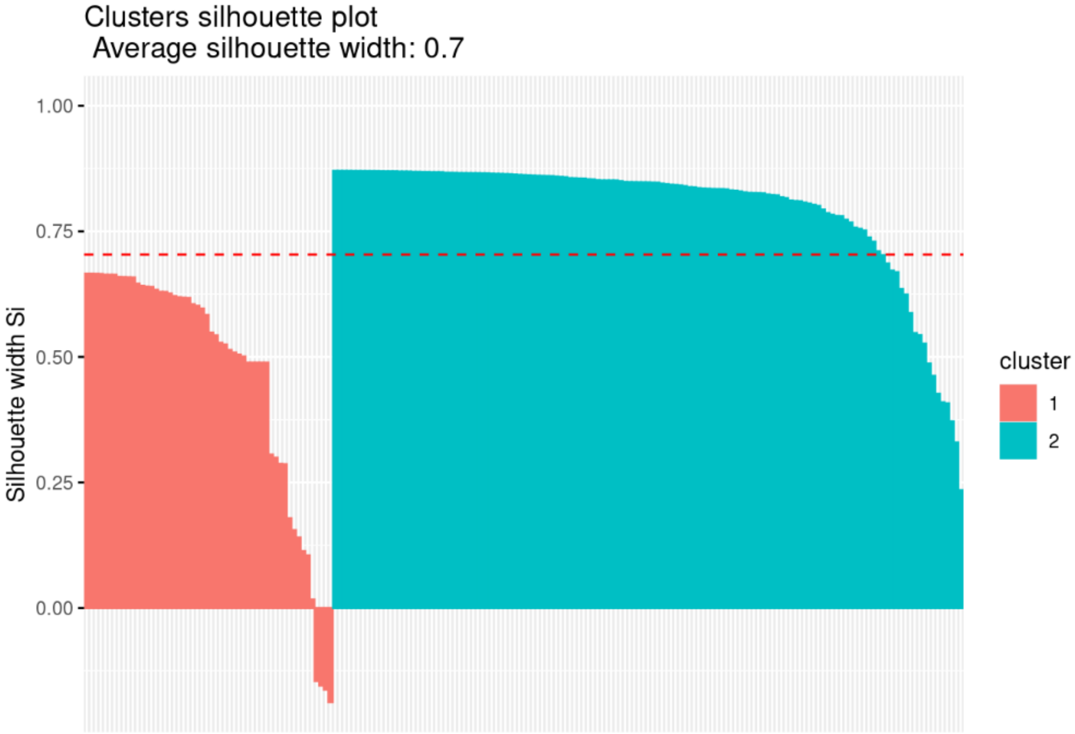

Scree Plot
Proportion of Variance Explained
0.0
0.2
0.4
0.6
0.8
1.0
1.0
1.5
2.0
2.5
3.0
3.5
4.0
principal component

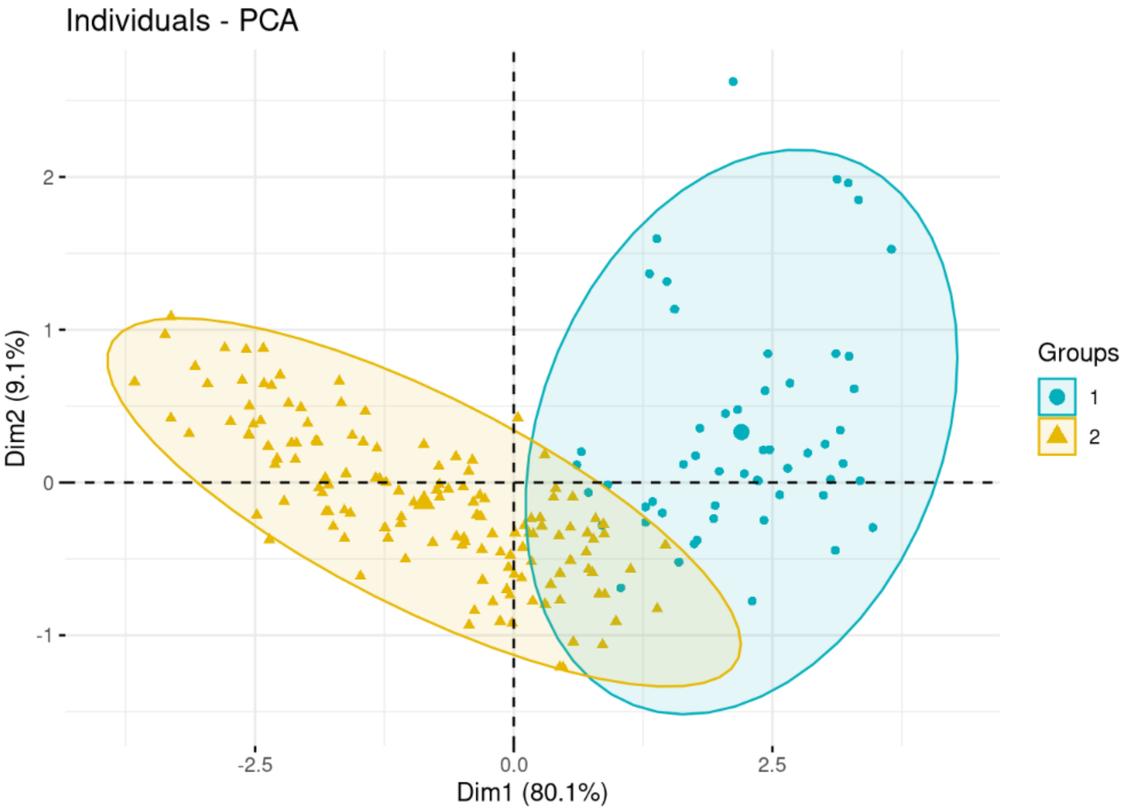

Cumulative Proportion of Variance Explained
0.0
0.2
0.4
0.6
0.8
1.0
1.0
1.5
2.0
2.5
3.0
3.5
4.0
Principal Component

Inference:

The above Scree plots explains that 90-100% of cumulative proportion of variance is being explained by the dataset of Global Innovation scores which means that our dataset is good.

Cluster-plots:

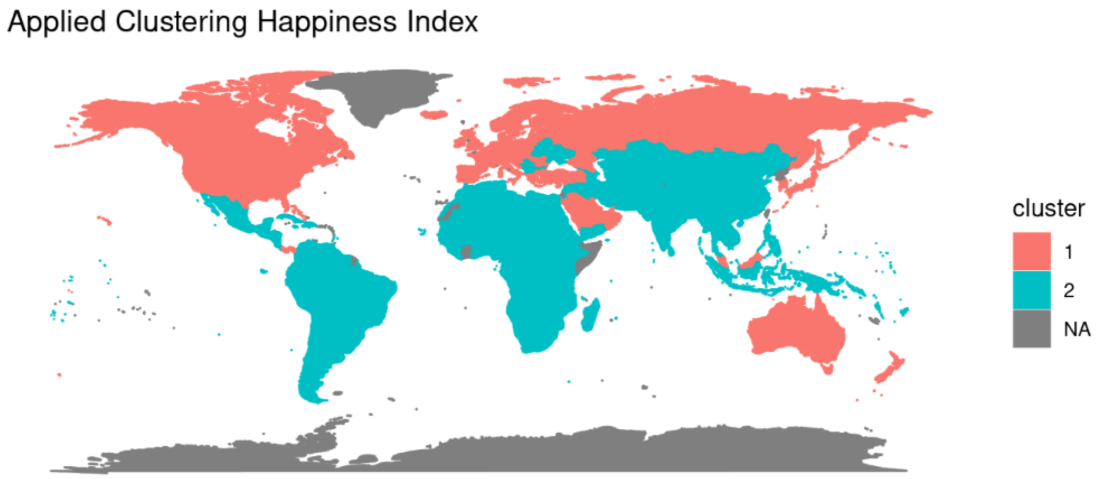




Inference:

The above cluster plots explains that we have a silhouette width score of 0.7 which means that both the clusters are very nicely separated and there is uniformity obtained through the separation of clusters. No. of clusters formed = 2.

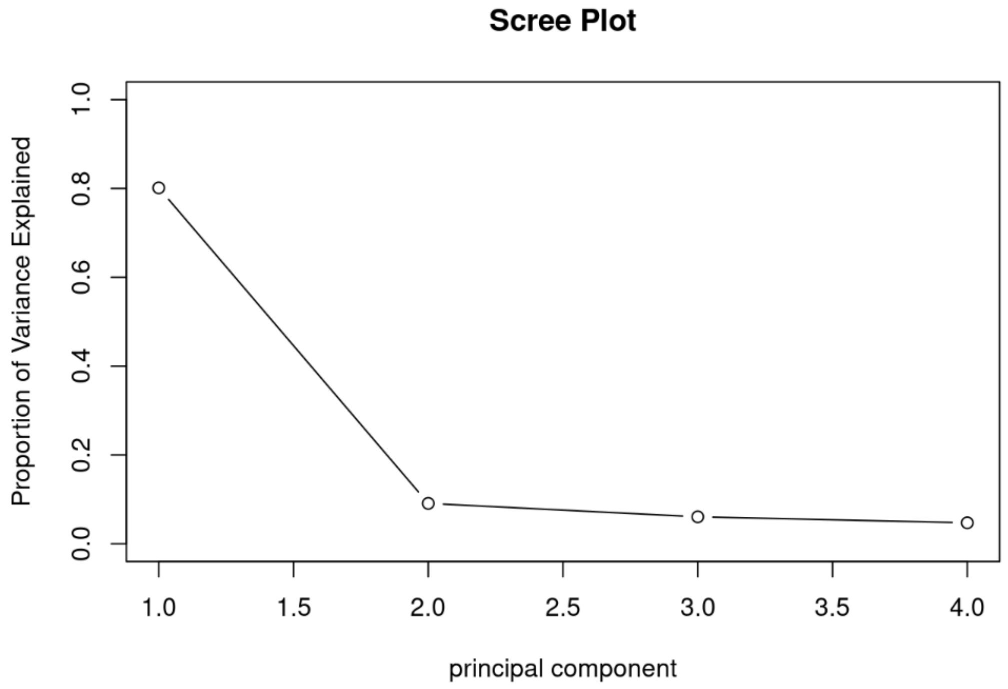
World Map

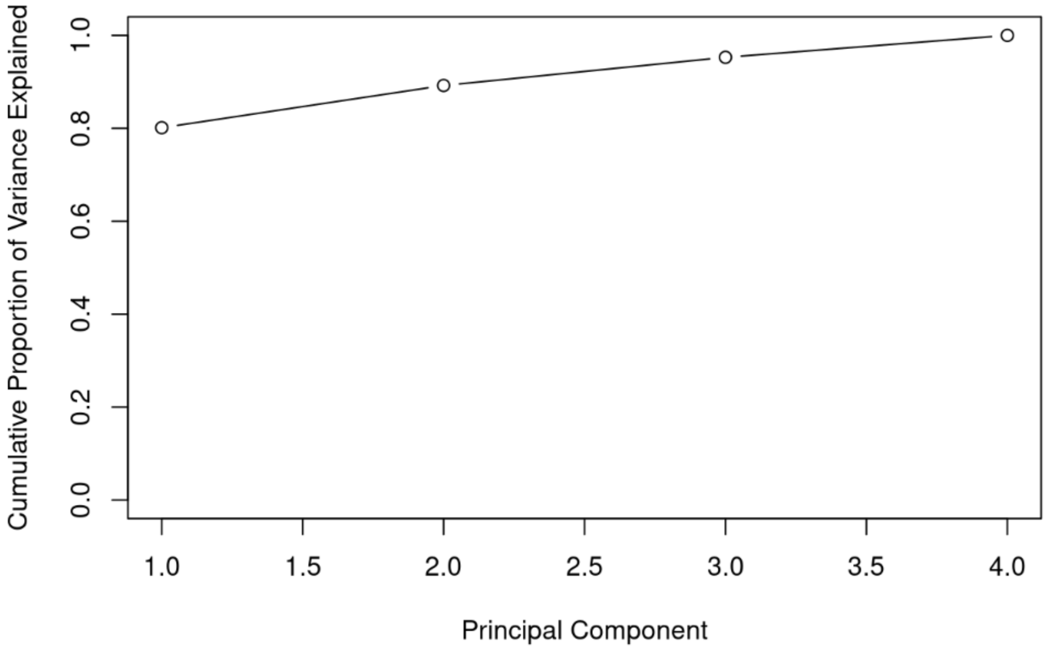


Inference:

The clusters are uniformly separated and distributed among various parts of the globe as observed from the above world map.

Analysis for Global Innovation scores:

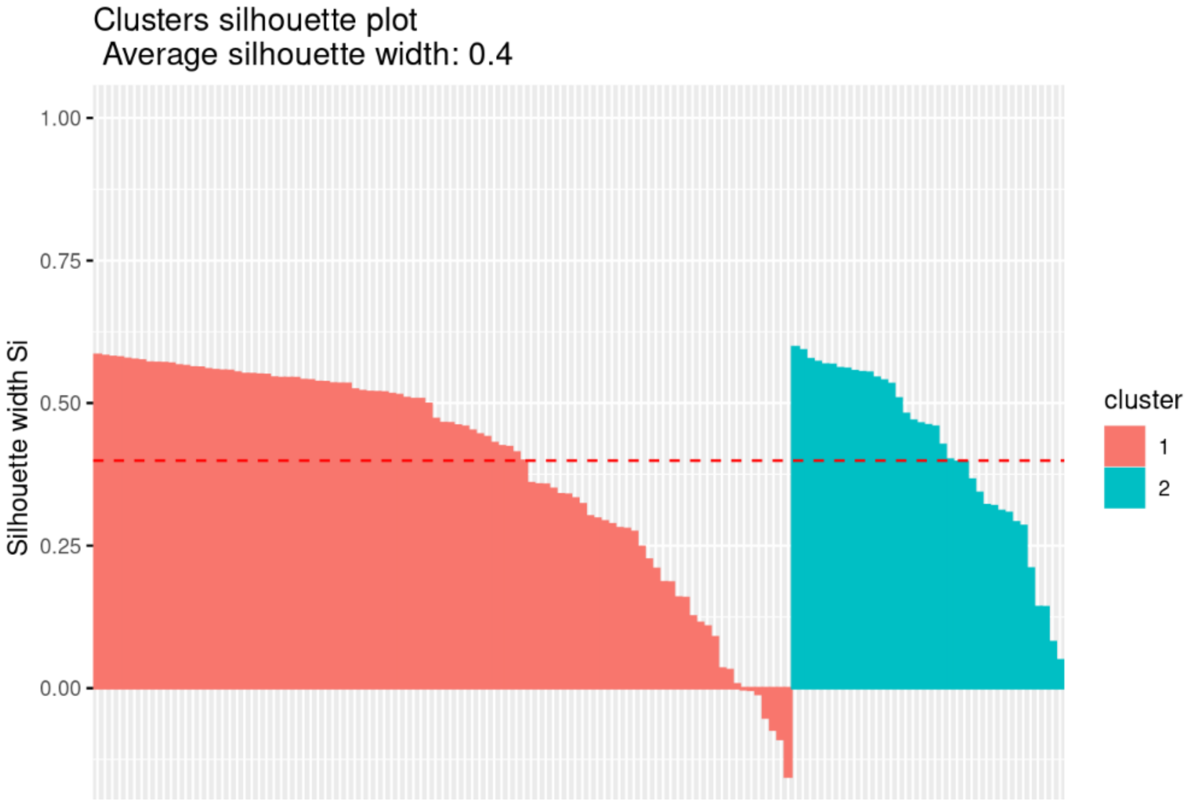

Scree Plot
Proportion of Variance Explained
0.0
0.2
0.4
0.6
0.8
1.0
1.0
1.5
2.0
2.5
3.0
3.5
4.0
principal component

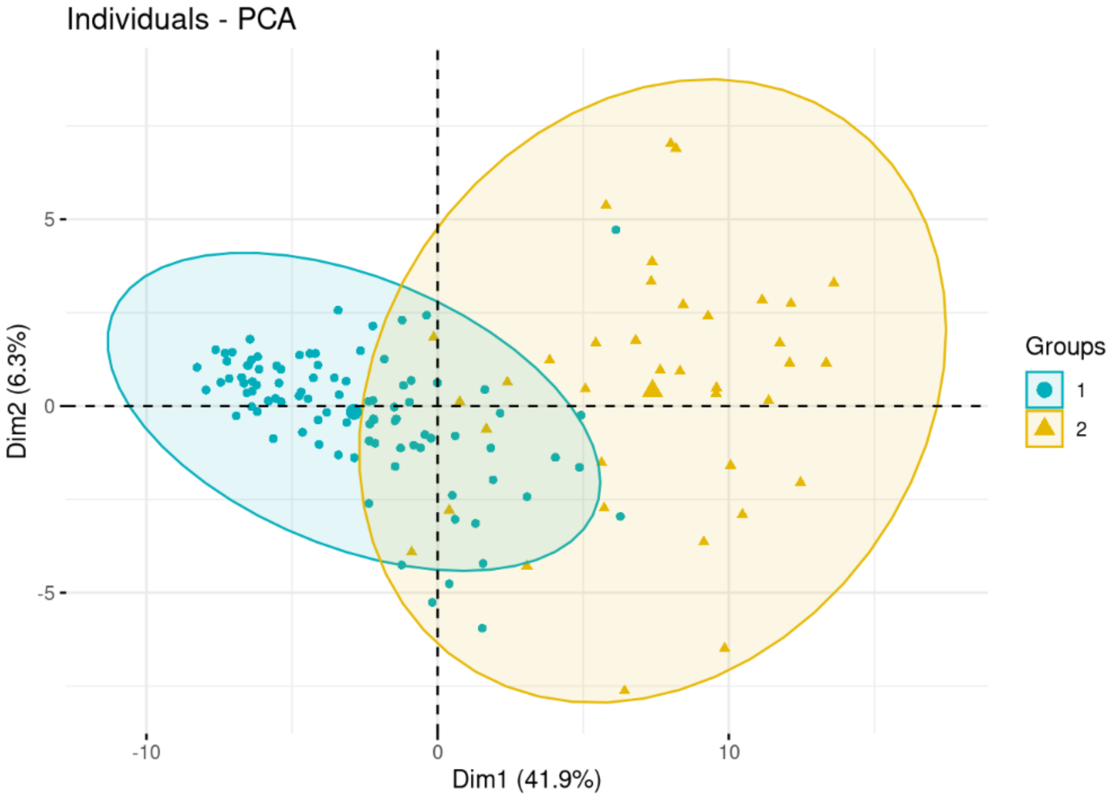

Cumulative Proportion of Variance Explained
0.0
0.2
0.4
0.6
0.8
1.0
1.0
1.5
2.0
2.5
3.0
3.5
4.0
Principal Component

Inference:

The above Scree plots explains that 90-100% of cumulative proportion of variance is being explained by the dataset of Global Innovation scores which means that our dataset is good.

Cluster plots:

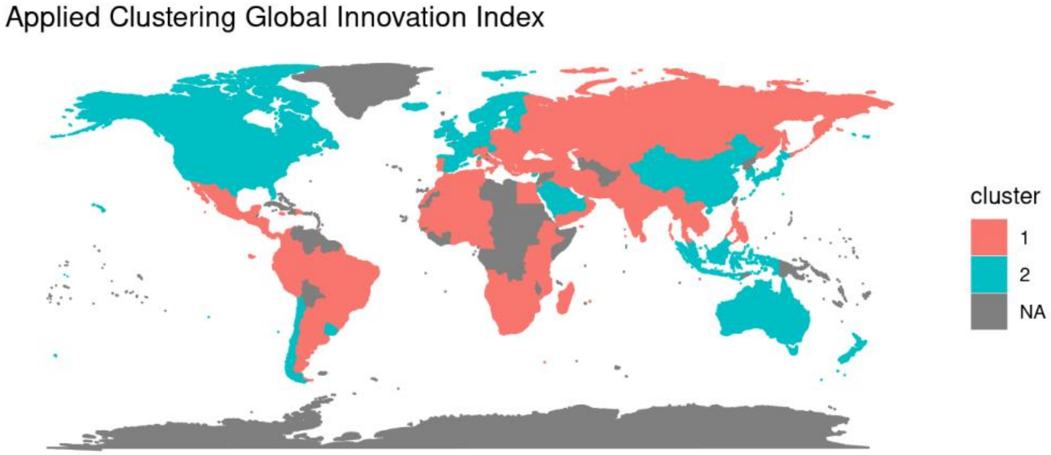




Inference:

The above plot explains that the average silhouette width is 0.4 and 2 clusters are formed.

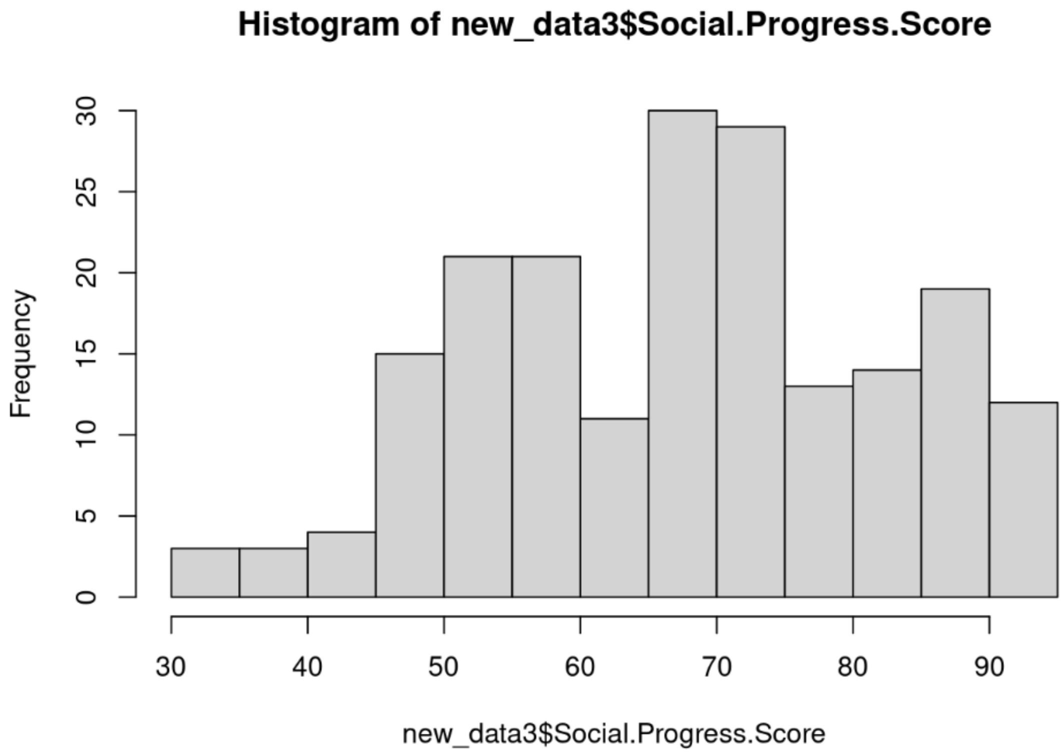
World map:



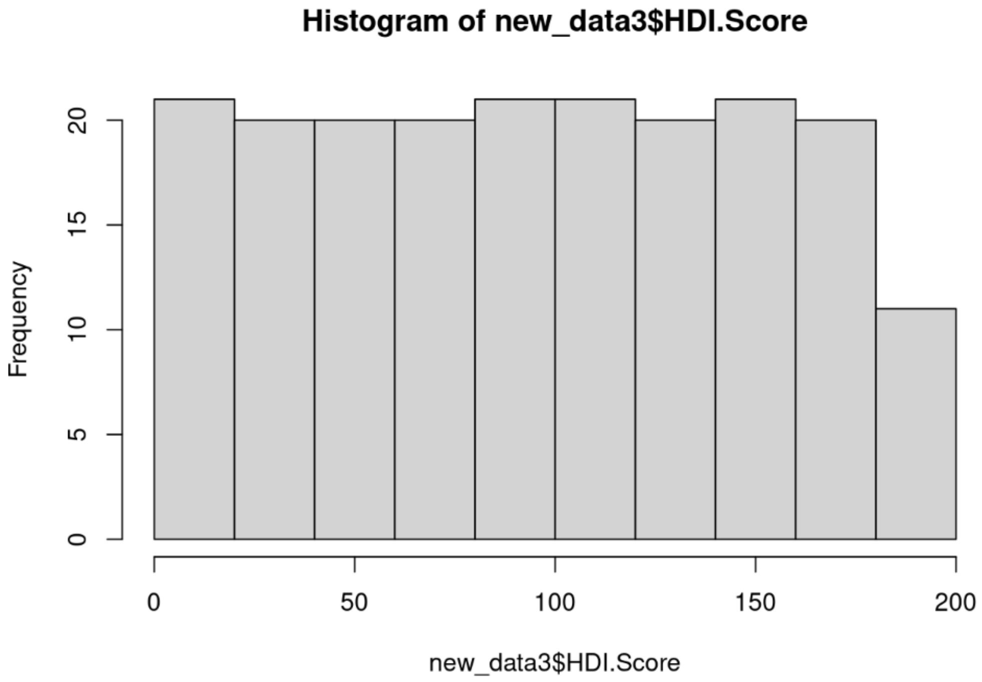
Inference:

Both the clusters have been aptly distributed across various portions of the globe as it can be seen from the above world map.

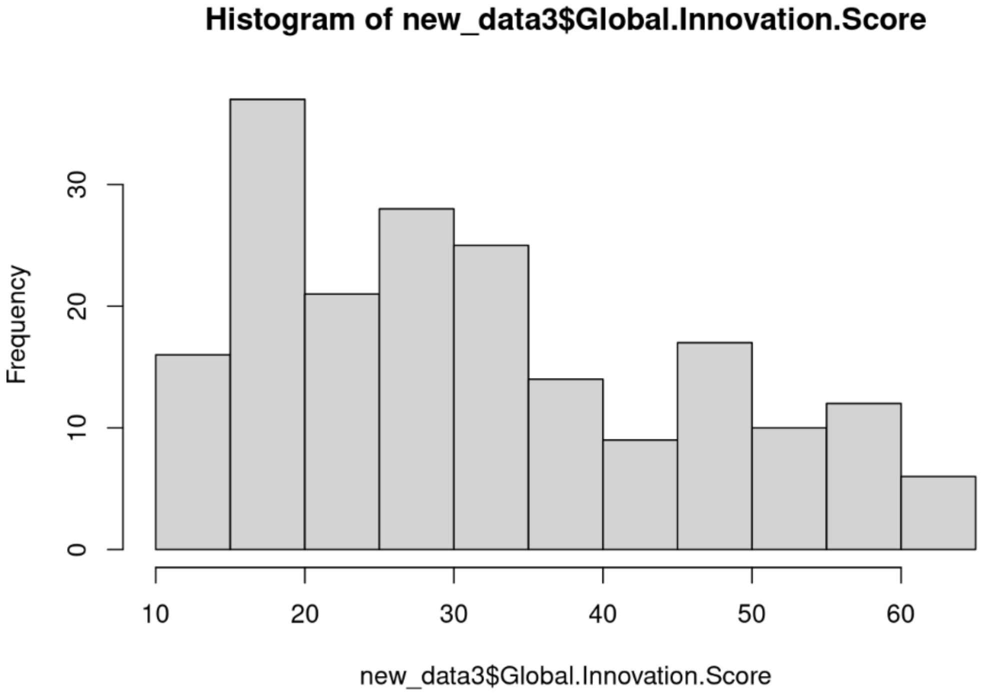
Histogram for Social Progress Score:



Histogram for Happiness Index Score:



Histogram for Global Innovation Score



Selecting Dataset with Best Normality of Scores:

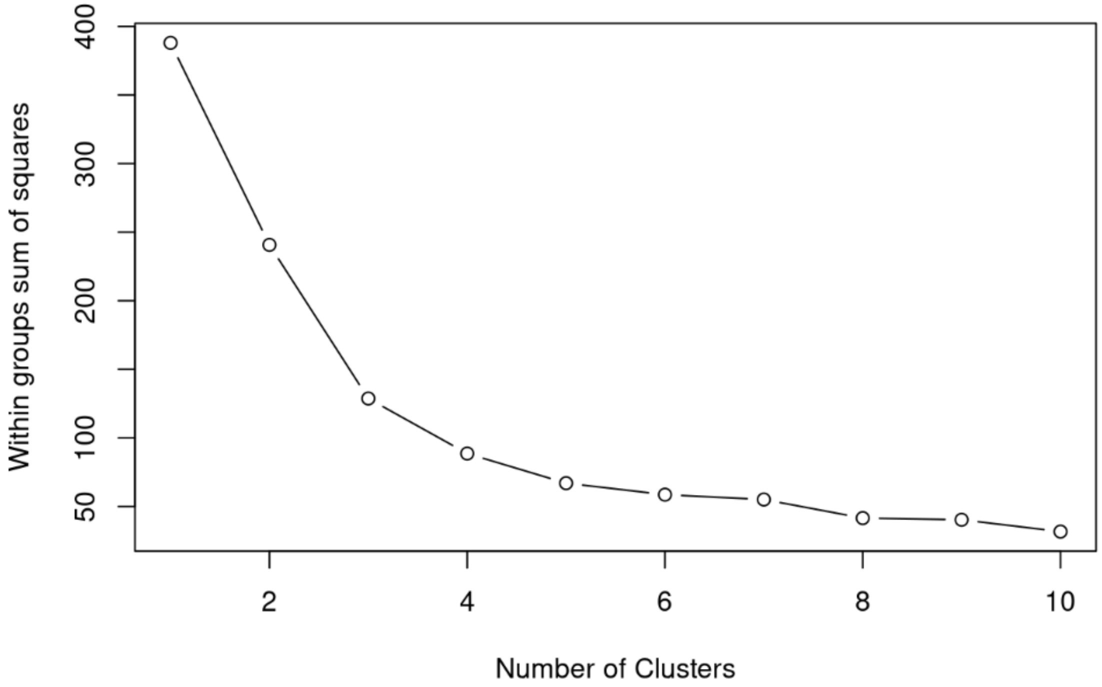
After removal of NULL values using Norm predict, Predictive mean matching and CART on the datasets and then applying Shapiro wilk test to obtain the following results:

Dataset	**W-value**	**P-value**
1	**0.9601**	**2.579e^-5**
2	**0.96102**	**3.237e^-5**
3	**0.9692**	**0.000276**

We see that the P-value of dataset 3 is the highest. Hence it, shall be selected for further computations.

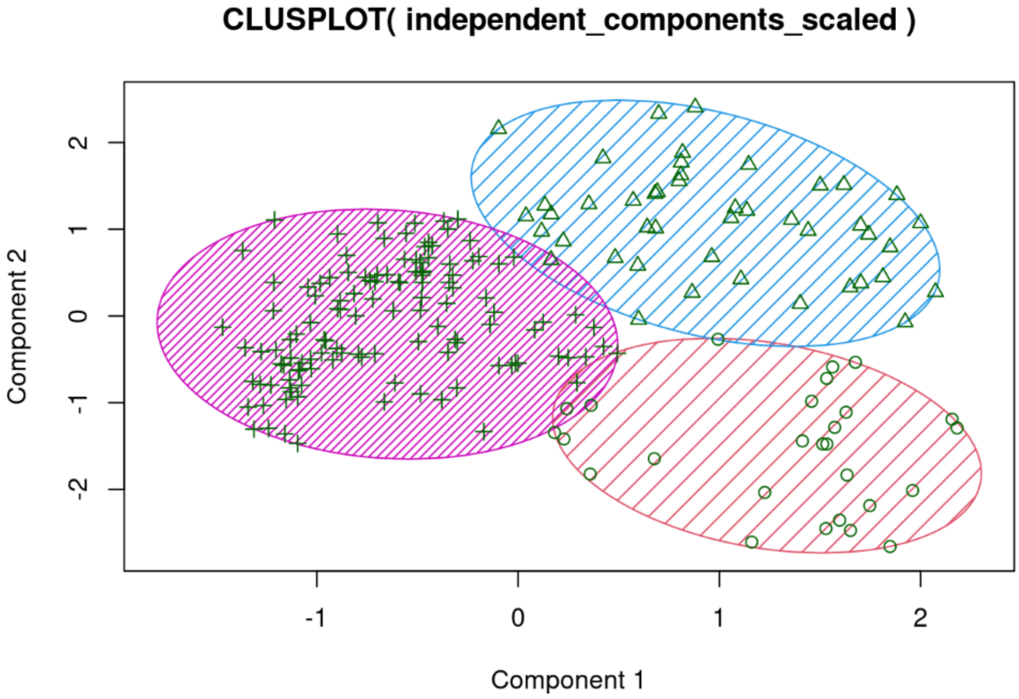
INDEPENDENT COMPONENT ANALYSIS

Plot depicting the Number of clusters vs Group Sum of Squares using K-Means clustering Algorithm:

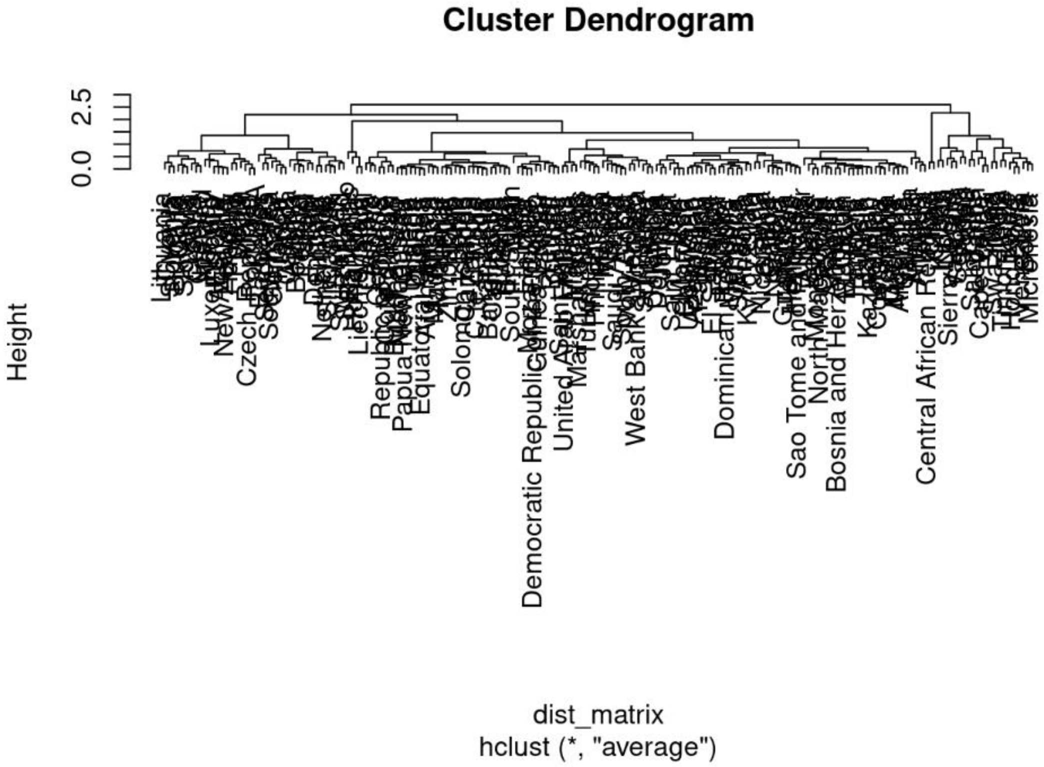


After applying the K-Means clustering Algorithm, we obtain that the score is highest for **4 clusters**. The corresponding score obtained is **0.486337.**

Cluster plot:

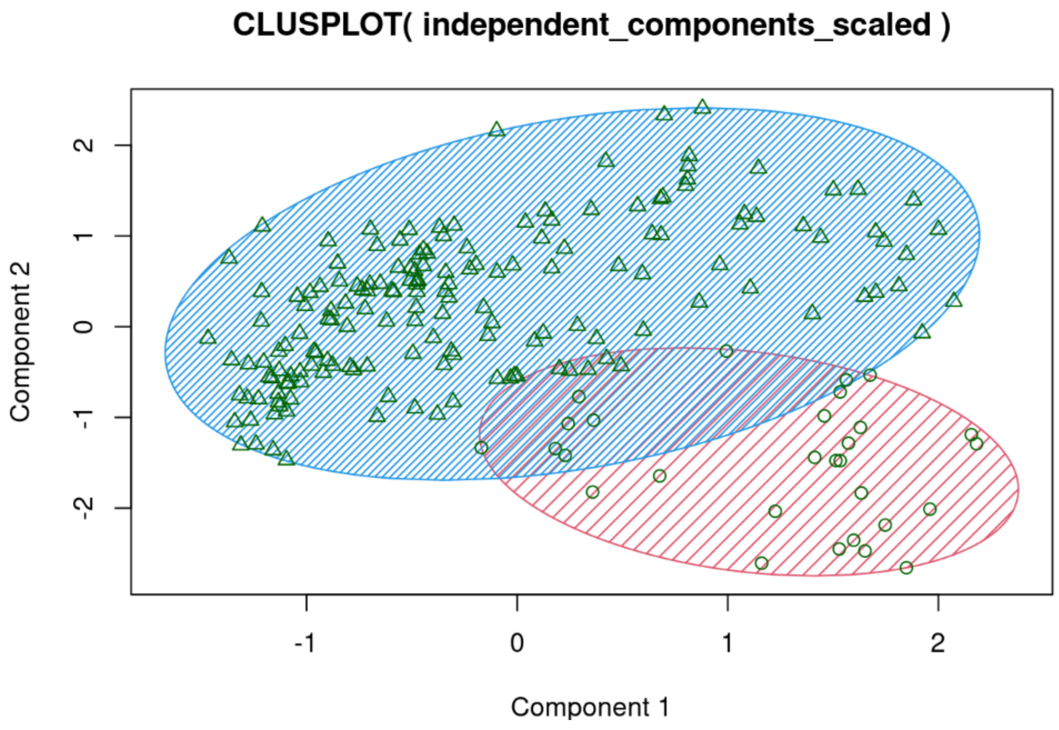


Hierarchical Clustering (Average Linkage):

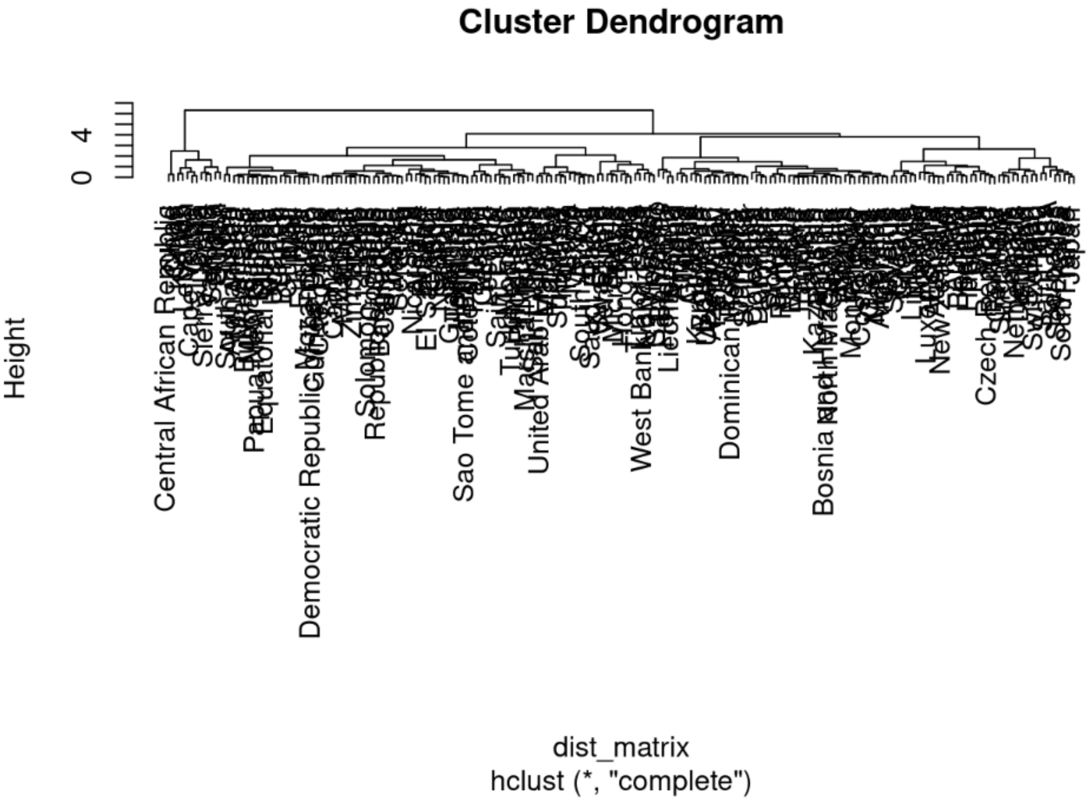


The maximum Silhouette Score obtained is **0.4333979** with **2 clusters.**

Cluster-Plot:

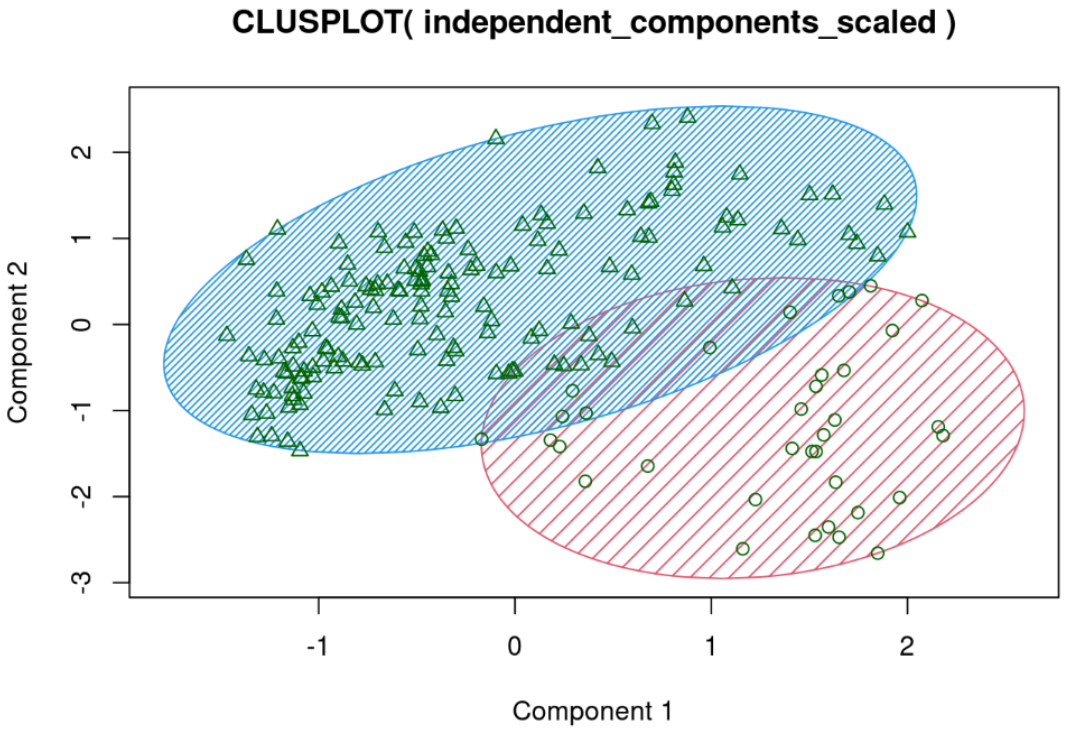


Hierarchical Clustering (Complete Linkage):



The silhouette score obtained is **0.4354677** with **2 clusters**.

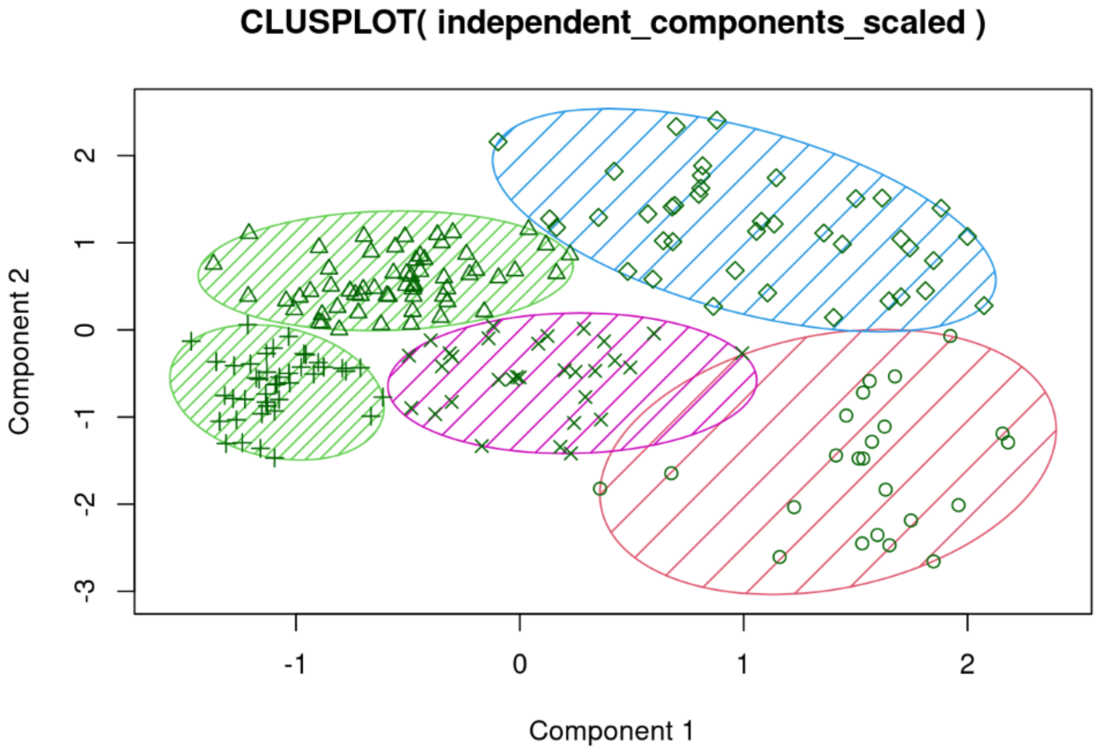
Cluster plot:



K-Medoid Clustering:

We observe that the Maximum score obtained is **0.4683219** with **2 clusters**.

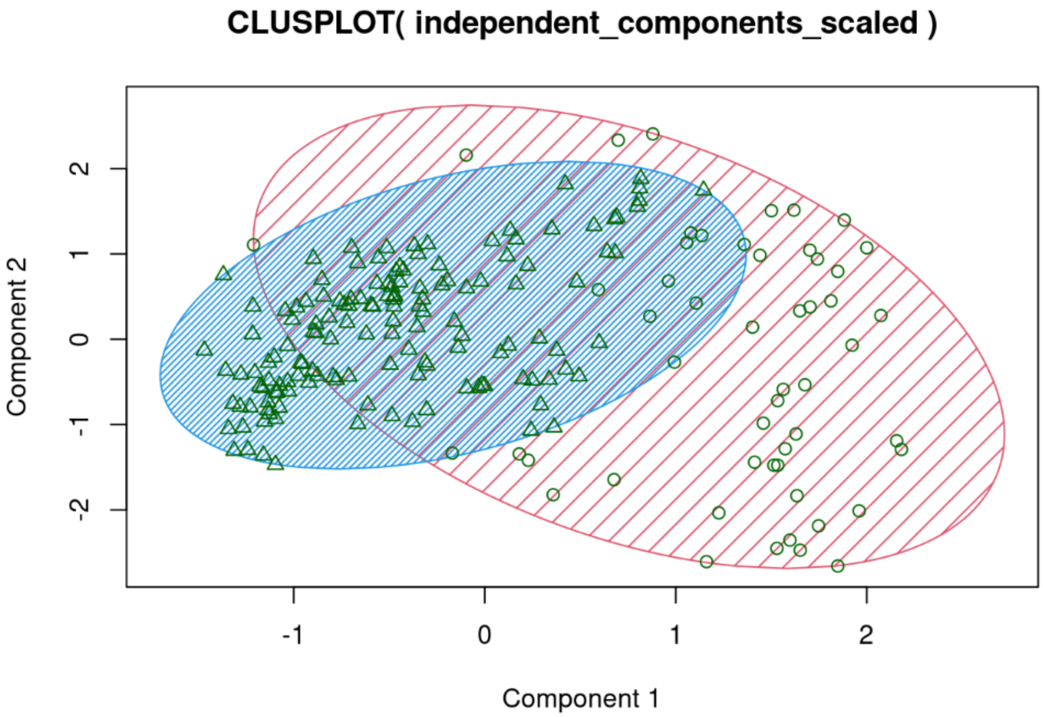
Cluster Plot:



DBSCAN Clustering Algorithm:

We observe that the Maximum score obtained is **0.417542** with **2 clusters.**

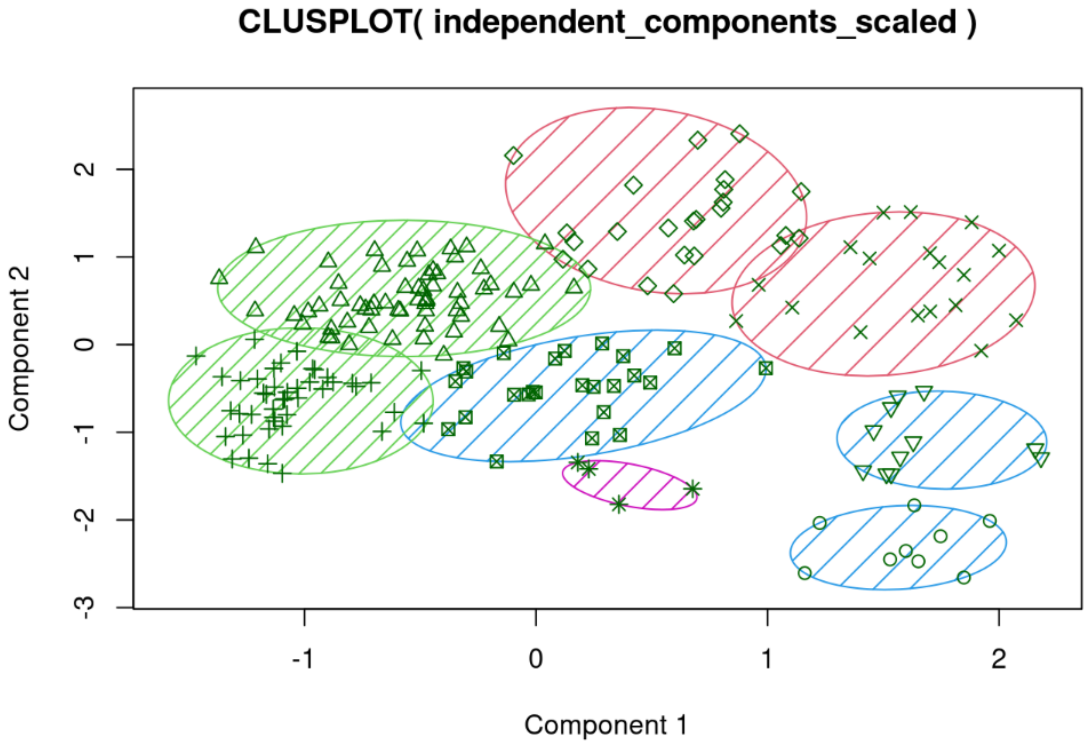
Cluster Plot:



GMM Clustering:

We observe that the maximum score obtained is **0.4547116** with **7 clusters**.

Cluster-plot:



Fuzzy Clustering Algorithm:

We observe that the maximum score obtained is **0.462874** with **6 clusters**.

Cluster-plot:

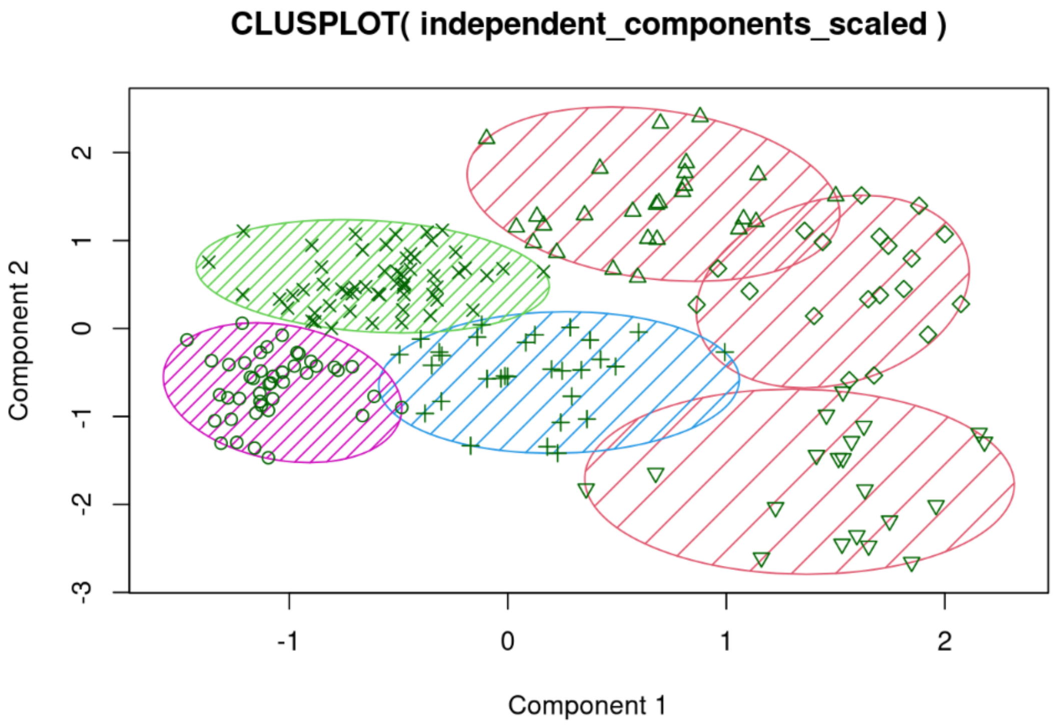
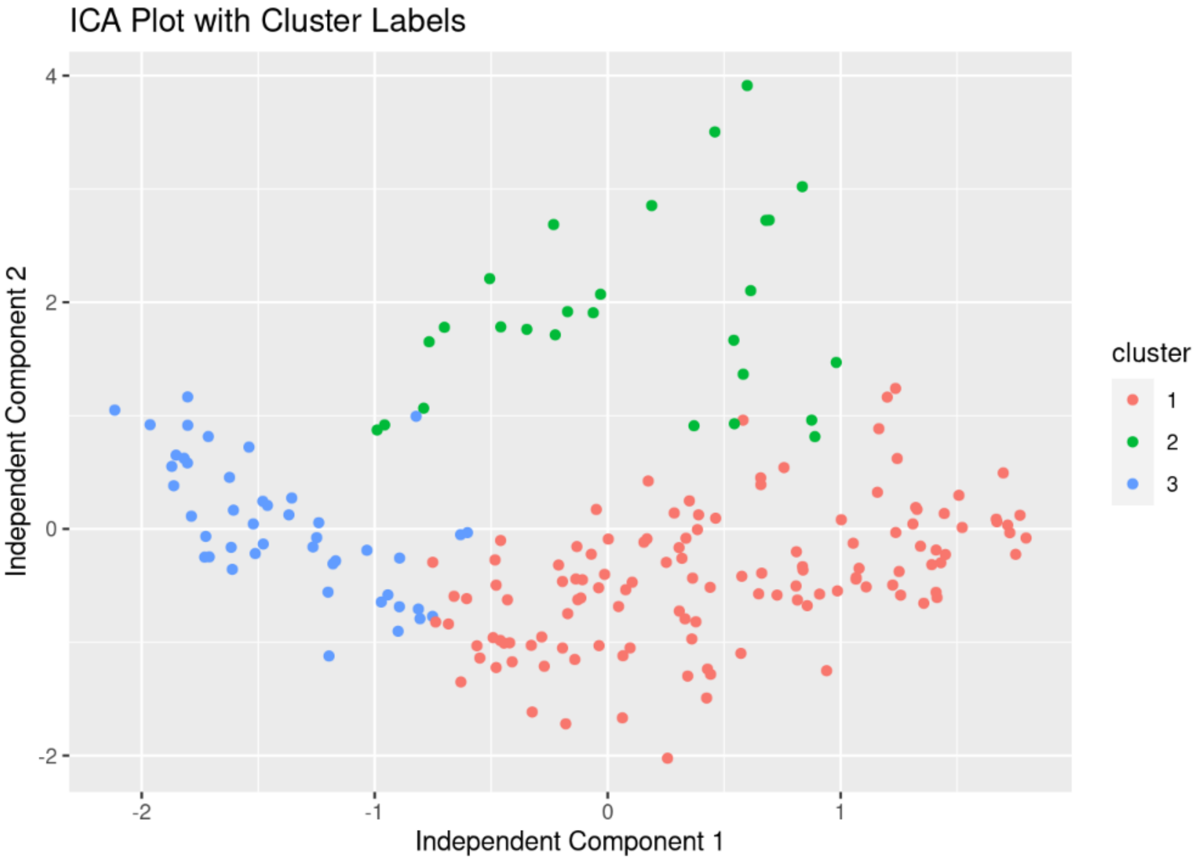


TABLE:

SL.NO	**CLUSTERING TECHNIQUE**	**SILHOUETTE SCORE**
1	K-MEANS	0.486337
2	AVERAGE LINKAGE	0.4333979
3	COMPLETE LINKAGE	0.4354677
4	K-MEDOID	0.4683219

5	GMM	0.4542116
6	DBSCAN	0.417542
7	FUZZY	0.462874

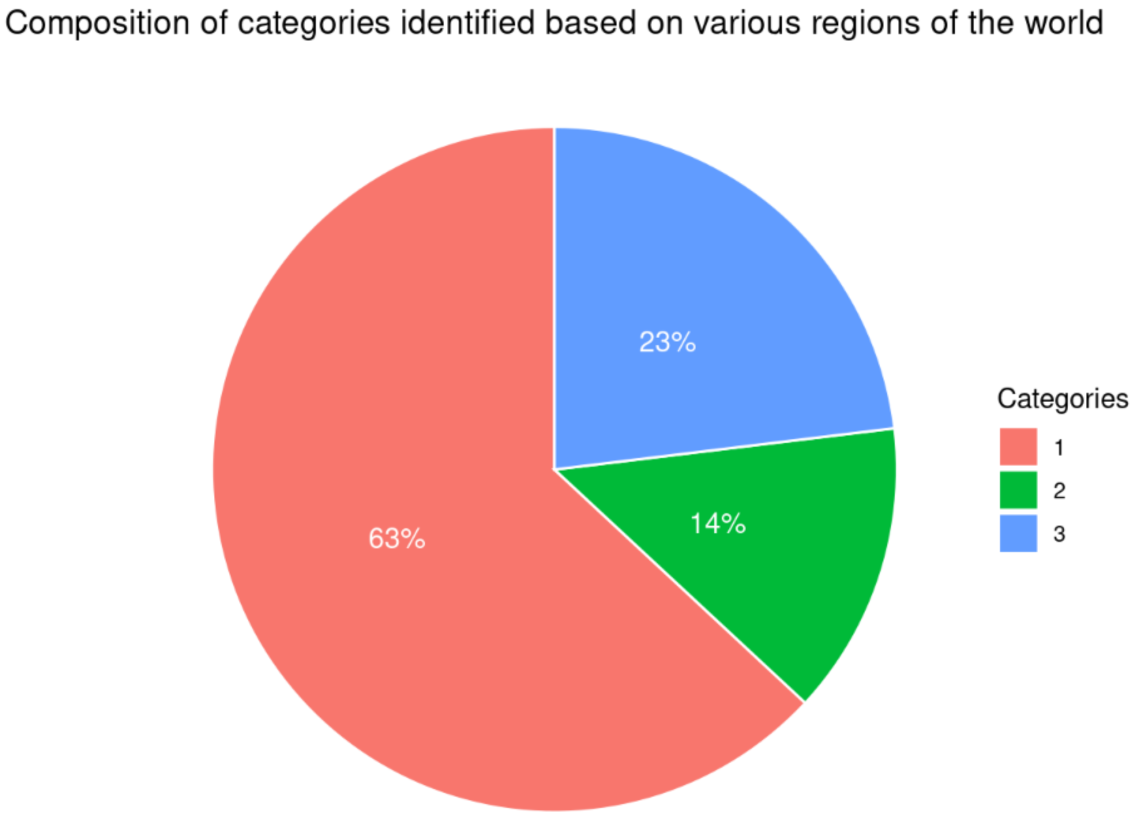
VISUALISATIONS:



Inference:

The above plot shows that all the clusters are well and uniformly distributed from our dataset.

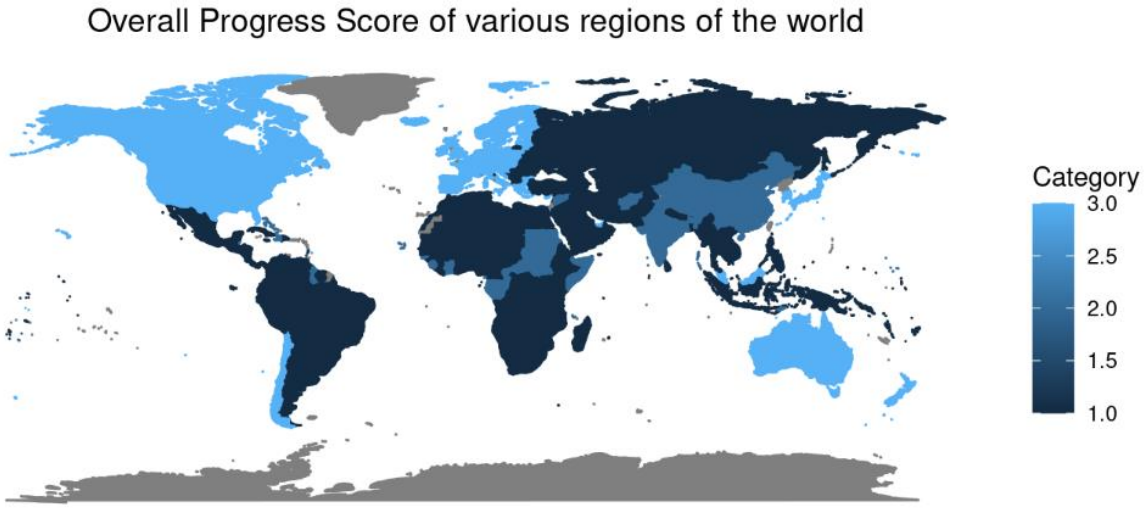
Plot depicting the concentration of various categories in the Globe



Inference:

The above plot depicts that 63% of the globe falls under Category-1(Under-developed), 14% of the globe falls under Category-2(Developing) and 14% falls under Category-3 (Developed)

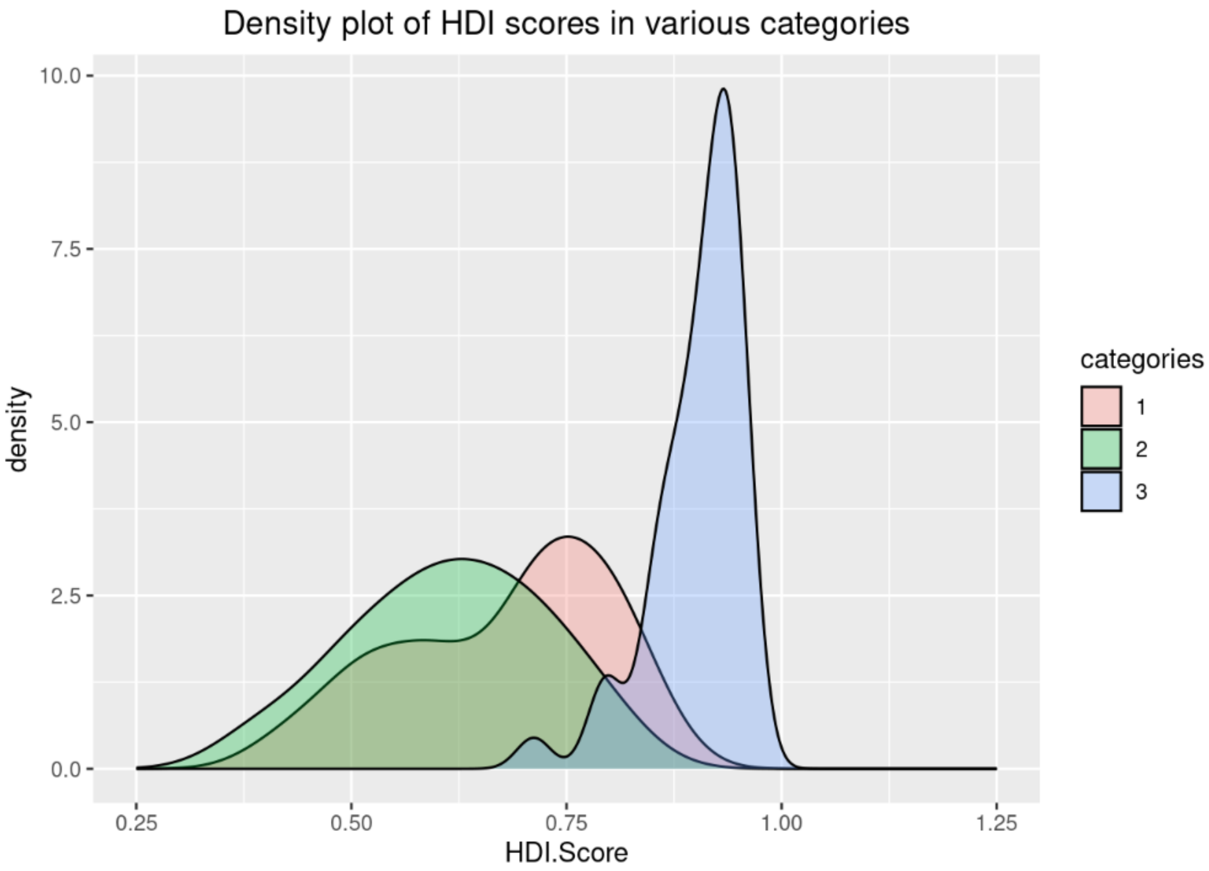
World plot depicting the different categories:



Inference:

The light blue color in the world-map above depicts the developed nations, dark blue color depicts the developing nations and the blackish color depicts the under-developed nations.

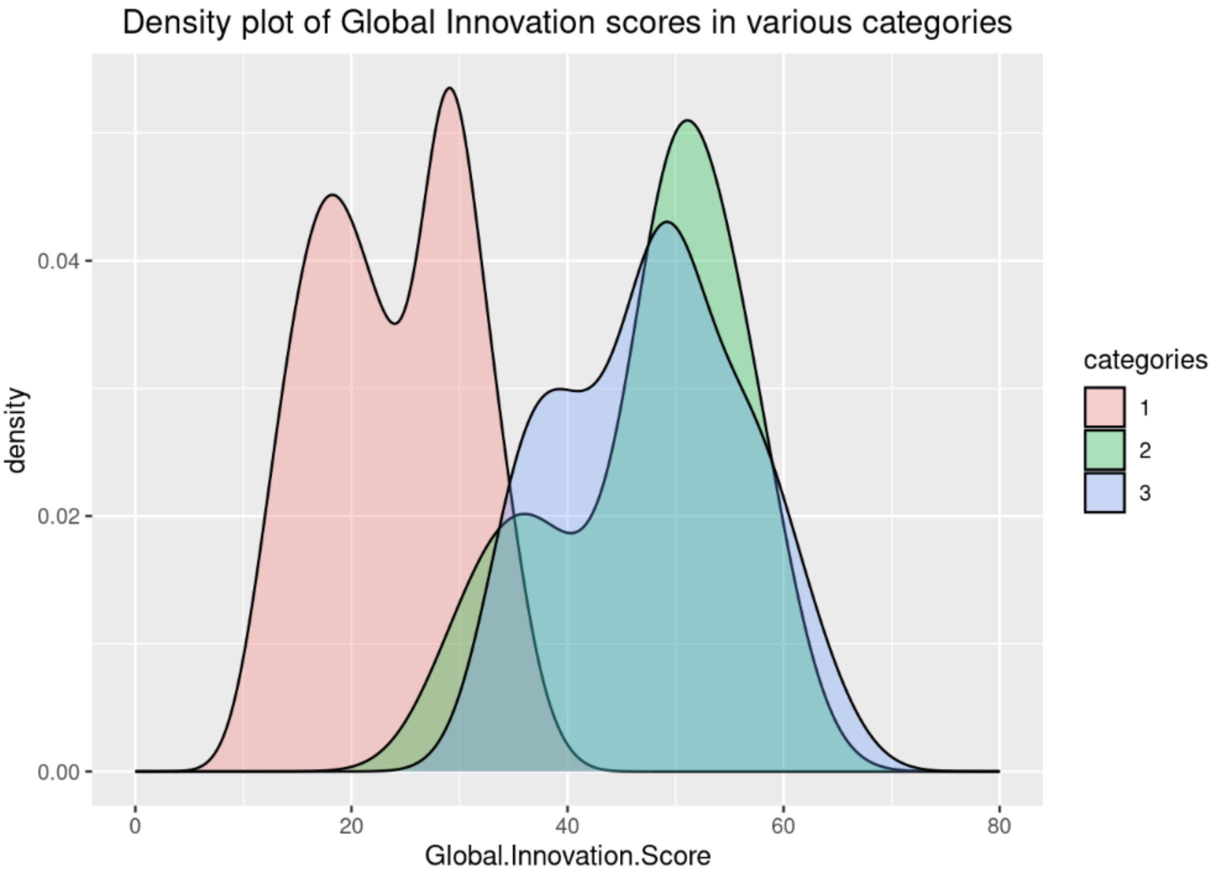
Density plot of HDI Scores based on various categories:



Inference:

The **highest HDI** score lies for all the **Category-3**. Then comes the **Category-1** followed by the **Category-2.**

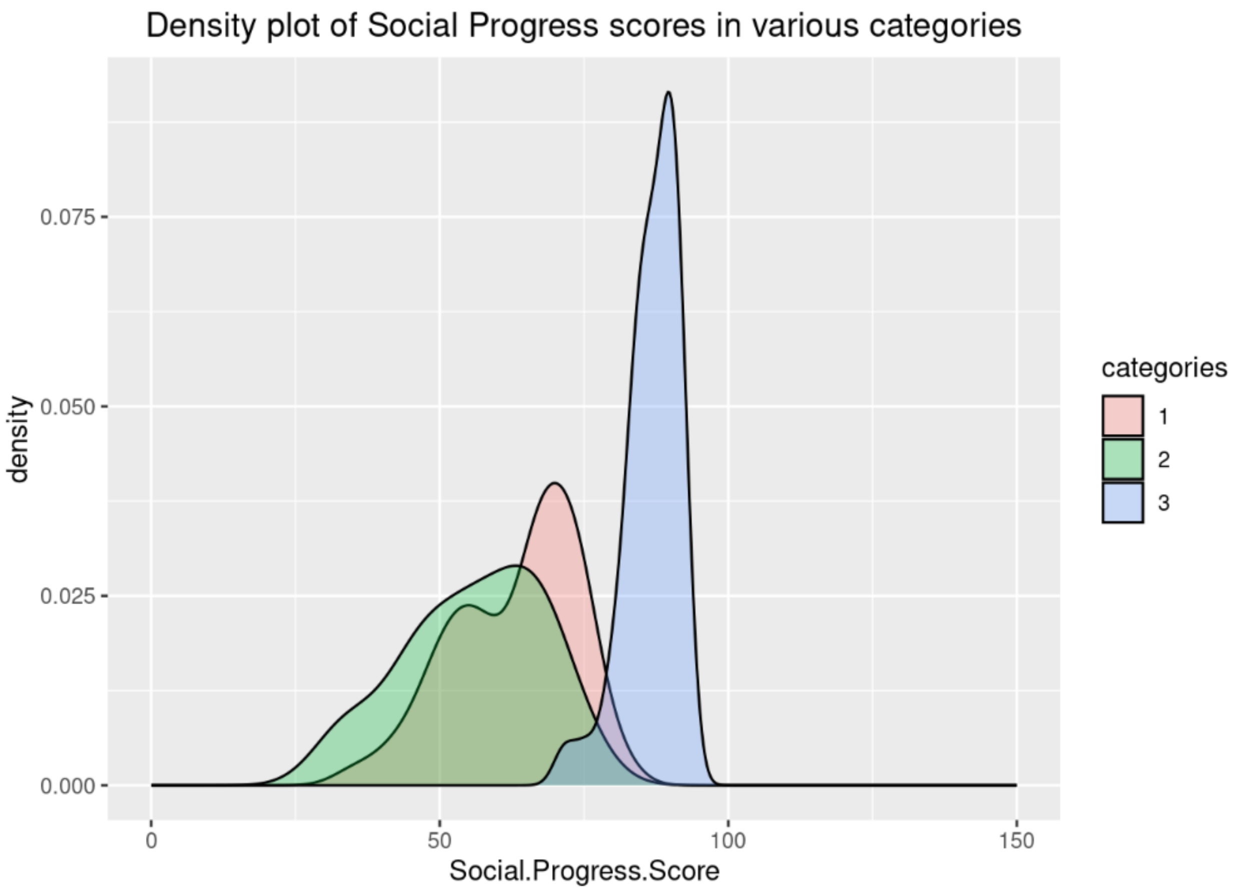
Density plot of Global Innovation scores with respect to various categories:



Inference:

The **highest Global Innovation scores** lies for all the **Category-2** countries followed by the **Category-3** and then the **Category-1 nations** respectively.

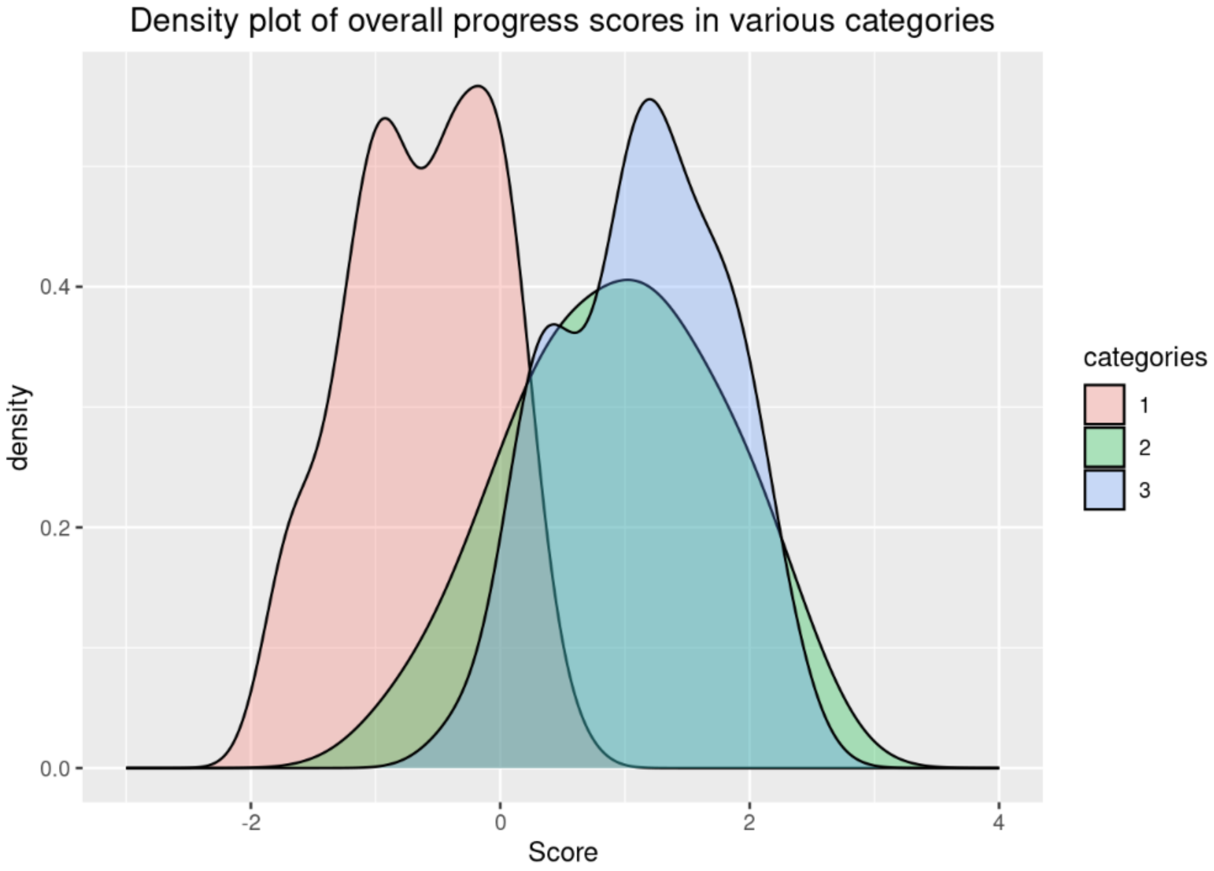
Density Plot depicting the Social-progress scores across various categories:



Inference:

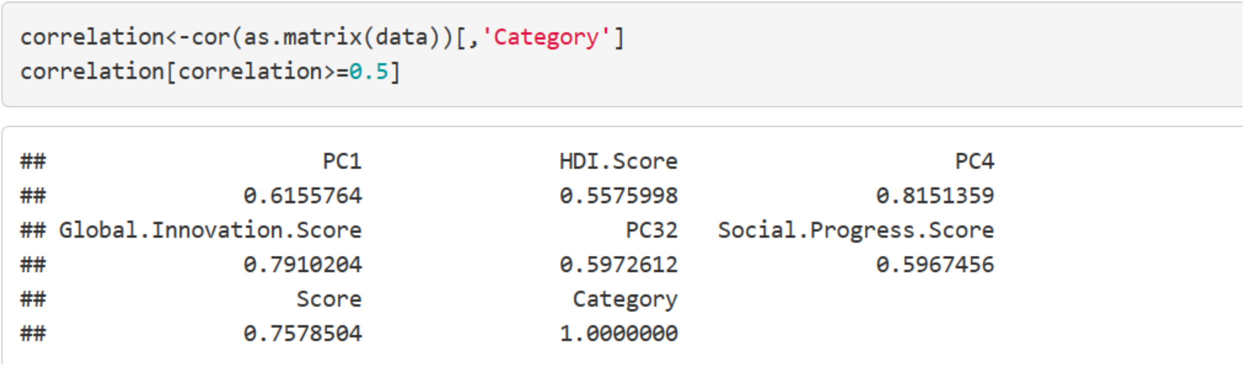
The **Highest** Social Progress score lies for the **Category-3 nations**, followed by the **Category-1 nations** and then the **Category-2 nations**.

Density plot of Overall scores across various categories:



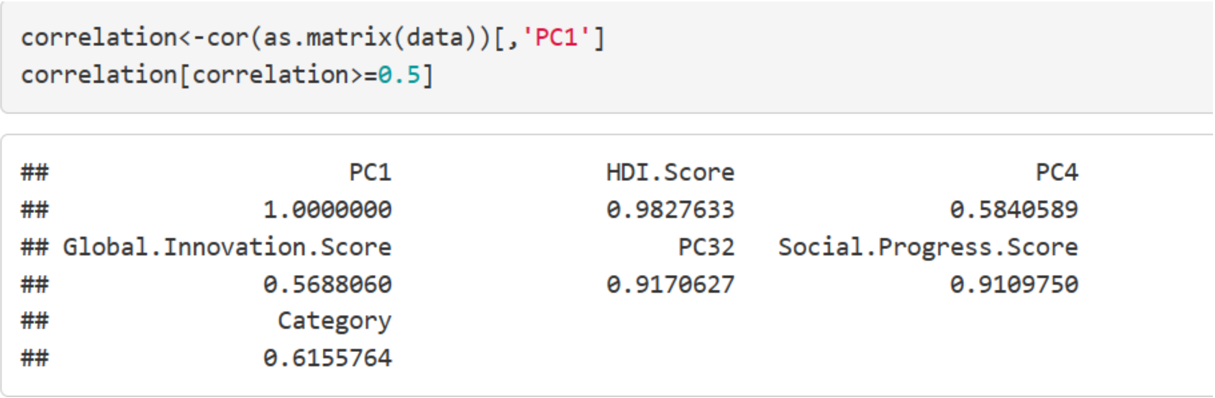
Inference:

The **Highest overall progress** score lies for the **Category-3** nations, followed by the **Category-2** nations and then the **Category-1** nations.



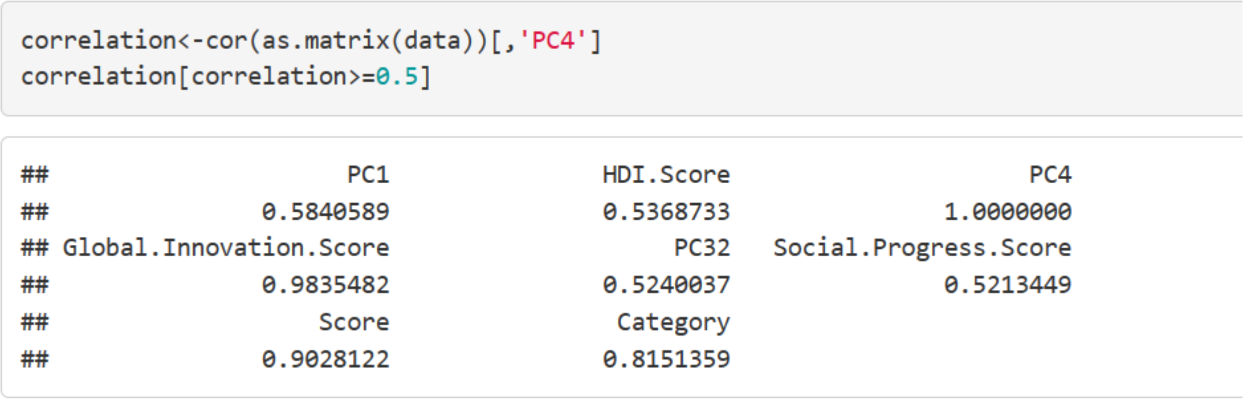
```r
correlation<-cor(as.matrix(data))[,'Category']
correlation[correlation>=0.5]
```

```
##                   PC1             HDI.Score                   PC4 
##             0.6155764             0.5575998             0.8151359 
## Global.Innovation.Score                  PC32 Social.Progress.Score 
##             0.7910204             0.5972612             0.5967456 
##                 Score              Category 
##             0.7578504             1.0000000
```



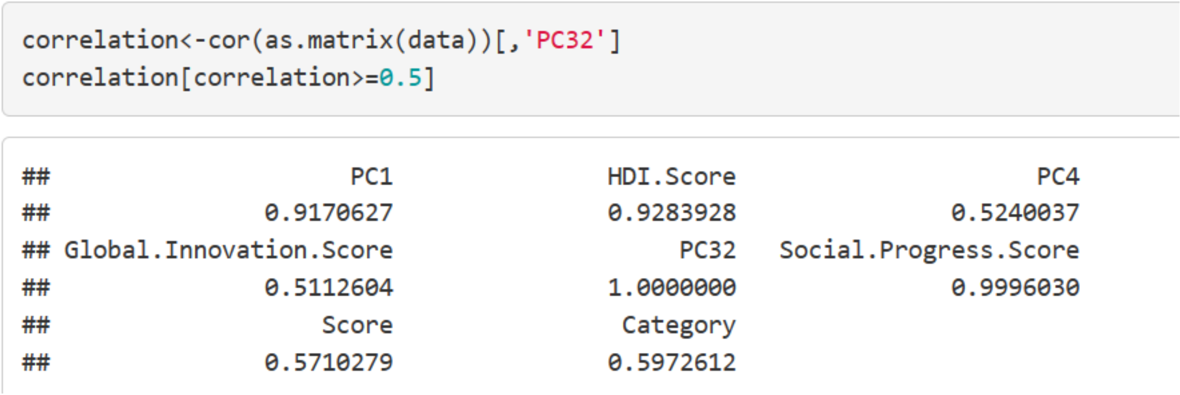
```r
correlation<-cor(as.matrix(data))[,'PC1']
correlation[correlation>=0.5]
```

```
##                   PC1             HDI.Score                   PC4 
##             1.0000000             0.9827633             0.5840589 
## Global.Innovation.Score                  PC32 Social.Progress.Score 
##             0.5688060             0.9170627             0.9109750 
##              Category 
##             0.6155764
```



```r
correlation<-cor(as.matrix(data))[,'PC4']
correlation[correlation>=0.5]
```

```
##                   PC1             HDI.Score                   PC4 
##             0.5840589             0.5368733             1.0000000 
## Global.Innovation.Score                  PC32 Social.Progress.Score 
##             0.9835482             0.5240037             0.5213449 
##                 Score              Category 
##             0.9028122             0.8151359
```



```r
correlation<-cor(as.matrix(data))[,'PC32']
correlation[correlation>=0.5]
```

```
##                   PC1             HDI.Score                   PC4 
##             0.9170627             0.9283928             0.5240037 
## Global.Innovation.Score                  PC32 Social.Progress.Score 
##             0.5112604             1.0000000             0.9996030 
##                 Score              Category 
##             0.5710279             0.5972612
```

Inference:

The categories obtained are dependent on PC1, PC4, and PC32 compared to the principal components. We found that PC1 is highly dependent on the Human Development Score and Social Progress Score, PC4 is highly dependent on the Global Innovation Score, and PC32 is highly dependent on the Social Progress Score and Human Development Score.

Conclusion:

We have successfully ranked countries based on their Social Progress Score, Human Development Score, and Global Innovation Score. Finally, we categorized the countries into three clusters, with a silhouette score of 0.48. Based on our observations, North America, Australia, New Zealand, Japan, South Korea, Europe, and Chile fall under category 3, which shows a higher band of scores and are marked as developed countries. Countries like India, China, South Sudan, Republic of Congo, and Central African Republic fall into category 2, which falls into the moderate score band and are marked as developing countries. Regions like most of Africa, South America, Central Asia, and Southeast Asia fall into the lowest score band and are marked as poor-performing countries.

The categories obtained are dependent on PC1, PC4, and PC32 compared to the principal components. We found that PC1 is highly dependent on the Human Development Score and Social Progress Score, PC4 is highly dependent on the Global Innovation Score, and PC32 is highly dependent on the Social Progress

Score and Human Development Score. There is further scope for exploring the dataset based on the 34 principal components, the obtained categories, and the overall score calculated.

References

I. "Development of a Holistic Index for Assessing the Sustainability of Rural Tourism" by García-Fernández, J.L., López Hernández, A., & Álvarez-Santana, E. (2017). Development of a Holistic Index for Assessing the Sustainability of Rural Tourism. Sustainability, 9(11), 1981."

II. "Development of a Holistic Index to Measure Social Sustainability of Urban Neigh- borhoods" by Wai Yee Lam and Edwin H.W. Chan. Lam, W.Y., & Chan, E.H.W. (2018). Development of a Holistic Index to Measure Social Sustainability of Urban Neighborhoods. Sustainability, 10(2), 452.

III. "A Holistic Approach to Sustainable Development: The Need for a New Economic Paradigm" by Roberto Crotti and Richard Knight. Crotti, R., & Knight, R. (2015). A Holistic Approach to Sustainable Development: The Need for a New Economic Para- digm. Sustainability, 7(8), 9833-9852."

IV. “Trends and Strategies towards Internalizing Higher Studies in India and developing a ranking based on that: A Case Study of Indian Universities" by Mona Khare. Khare. M.(2020).

V. “Development of a Holistic Ranking System for Sustainable Cities" by Kostiantyn Nie- mets a, Kateryna Kravchenko a, Yurii Kandyba a, Pavlo Kobylin a, Cezar Morar b (2017). Development of a Holistic Ranking System for Sustainable Cities. Sustainabil- ity, 9(4), 530.”

VI. United Nations Development Programme (UNDP). (2019). Human Development In- dices and Indicators: 2019 Statistical Update.

VII. Social Progress Imperative. (2021). 2021 Social Progress Index.

VIII. Global Innovation Index. (2021) WIPO