# From Habits To Health:

## Understanding Cancer Clusters Across Countries by Death Rate, Lifestyle Factors and Geolocation

John Pauline Pineda

December 8, 2023

# OUTLINE

- Executive Summary
- Introduction
- Methodology
- Results Summary
  - Data Gathering and Description
  - Data Quality Assessment
  - Data Preprocessing
  - Data Exploration
  - Model Development, Selection and Post-Hoc Analysis
- Detailed Findings
- Discussion
  - Overall Findings and Implications
- Conclusion
- Appendix

# EXECUTIVE SUMMARY

- ## Study Objective
  - Conduct data analysis and modeling to identify inherent patterns, structures and groupings among countries based on their similarities and dissimilarities in terms of age-standardized death rates for the top nine cancer types (including lung, pancreatic, colorectal, stomach, esophageal, liver, prostate, cervical and breast cancers) as evaluated against lifestyle factors (smoking, overweight and alcohol prevalence) and geolocation data.
- ## Methodology and Tools
  - Data Quality Assessment using Python Pandas and NumPy APIs
  - Data Preprocessing using Python Pandas and Scikit-Learn APIs
  - Data Exploration using Python Matplotlib, Seaborn and SciPy APIs
  - Clustering Model Development and Analysis using Python Scikit-Learn and GeoPandas APIs
- ## Overall Findings
  - Data quality issues were identified and handled with the appropriate data preprocessing methods.
  - Clustering analysis was implemented for numeric descriptors (cancer death rates).
  - Post-hoc analysis was conducted for the formulated clusters against the target descriptors (lifestyle factors).
  - Post-hoc analysis was conducted for the formulated clusters against geolocation data (regions).

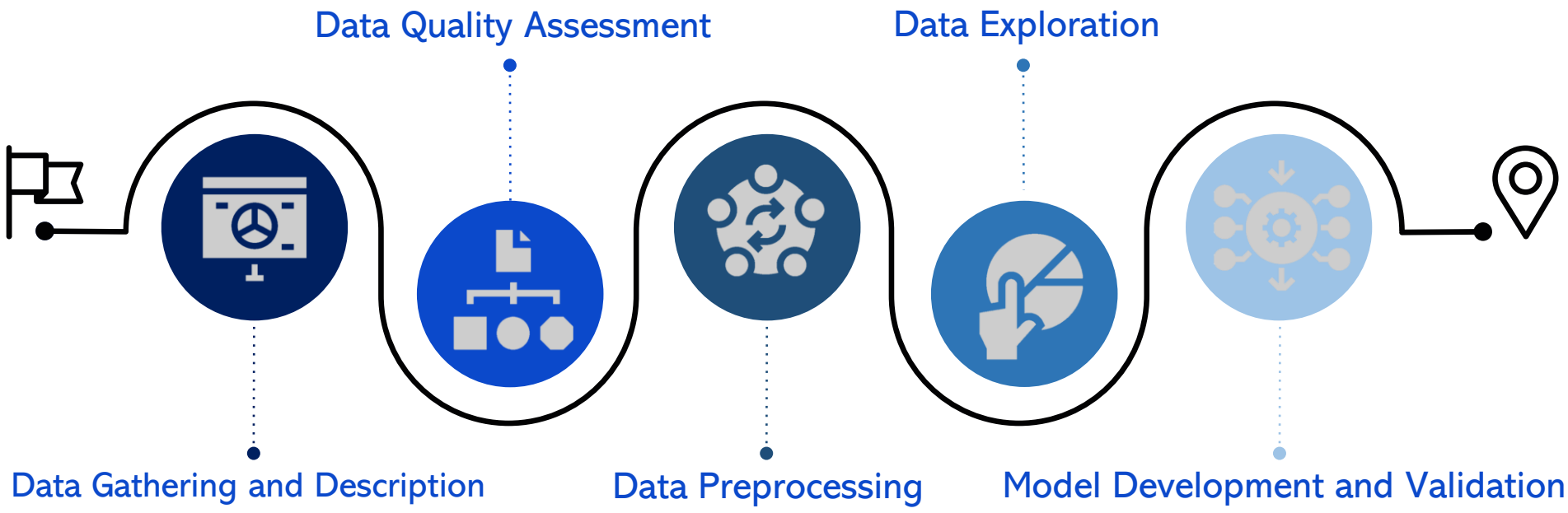
# INTRODUCTION

- Cancer – a complex group of diseases characterized by the uncontrolled growth and spread of abnormal cells, has emerged as a leading cause of morbidity and mortality worldwide.
- Analyzing cancer death rates allows for the identification of disparities in healthcare access, quality, and outcomes between countries. Understanding these disparities is essential for developing targeted interventions to reduce health inequities.
- This capstone project generally aims to develop a clustering model that could allow the segmentation of countries into distinct groups and provide a more granular view of relationships across different cancer death rates, lifestyle factors and geolocation data
    - In particular, multiple clustering models will be formulated with varying cluster numbers. The final model with the optimal number of clusters will be selected among candidates based on its ability to quantify the compactness and separation of clusters. A post-hoc analysis will be conducted on the formulated clusters to highlight specific characteristics common among groups of data points.

Section 1

# Methodology

# METHODOLOGY




- Source assessment
- Data download
- Variable description
- Data dimension
- Column types
- Numeric statistics
- Categorical statistics

- Row duplicates
- Column fill rate
- Row fill rate
- Low variance
- High data skew

- Data cleaning
- Outlier treatment
- Collinearity
- Transformation
- Centering and scaling

- Visual exploration
- Hypothesis testing

- Model development
- Hyperparameter tuning
- Model selection
- Post-hoc analysis

Section 2

# Results Summary

# RESULTS – DATA GATHERING

## Global Cancer Death Rate Estimates by Country

**Lung Cancer** [Source: OurWorldInData.Org]
**Pancreatic Cancer** [Source: OurWorldInData.Org]
**Colorectal Cancer** [Source: OurWorldInData.Org]
**Stomach Cancer** [Source: OurWorldInData.Org]
**Esophageal Cancer** [Source: OurWorldInData.Org]
**Liver Cancer** [Source: OurWorldInData.Org]
**Prostate Cancer** [Source: OurWorldInData.Org]
**Cervical Cancer** [Source: OurWorldInData.Org]
**Breast Cancer** [Source: OurWorldInData.Org]

## Lifestyle Factor Estimates

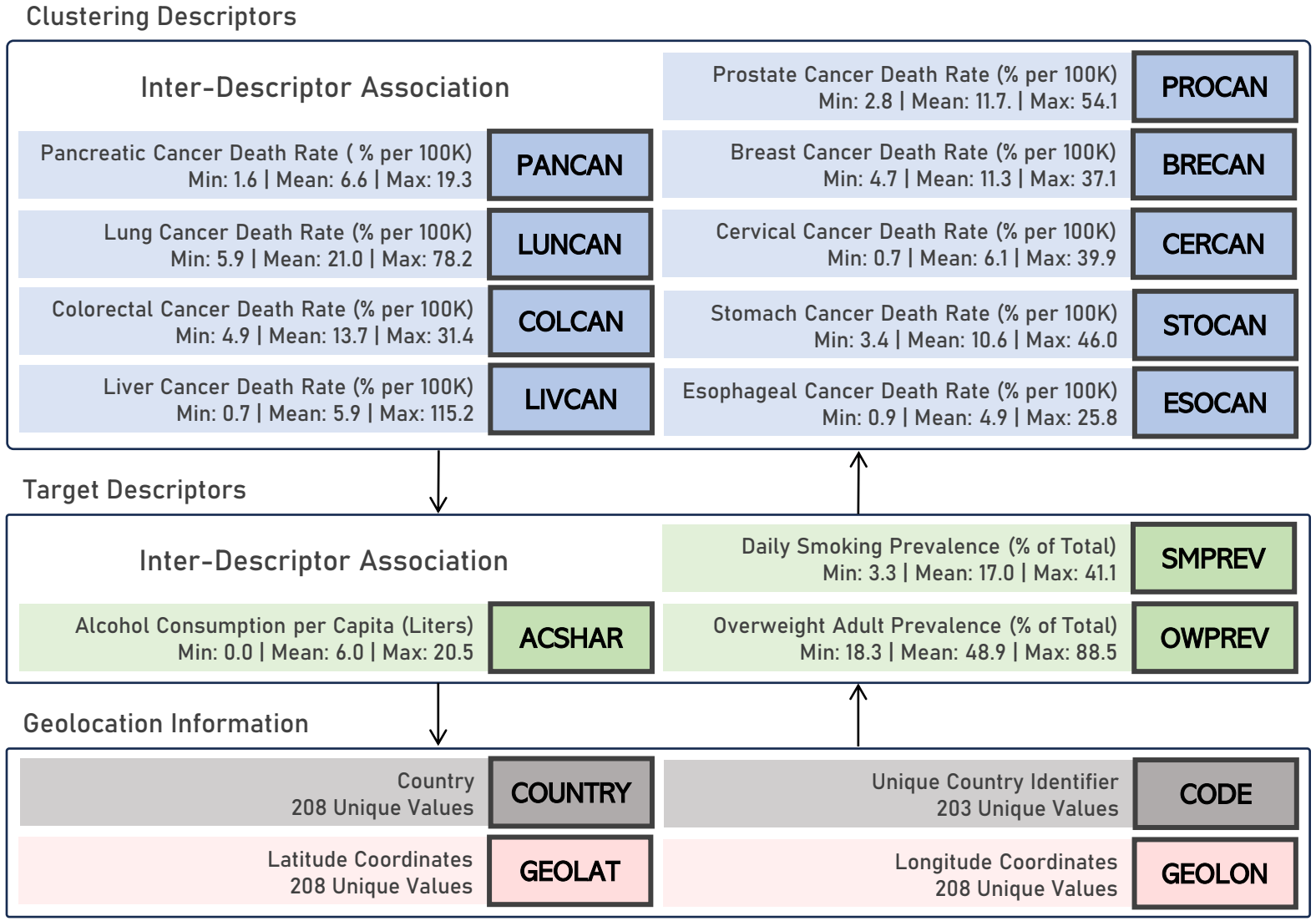
**Smoking Prevalence** [Source: OurWorldInData.Org]
**Overweight Prevalence** [Source: OurWorldInData.Org]
**Alcohol Consumption** [Source: OurWorldInData.Org]

## Geolocation Information

**Geographic Coordinates** [Source: GeoDatos.Net]
**Global Map Shape File** [Source: GeoJson-Maps]

- The study hypothesizes that **death rate estimates** among different cancer types (lung, pancreatic, colorectal, stomach, esophageal, liver, prostate, cervical, and breast) are sufficiently correlated to enable the presence of underlying structures or patterns within the data and allow the natural groupings or clusters in the data based on the similarity or dissimilarity of the observations. Additionally, the study assumes that the characteristics of these clusters are related to differences in **lifestyle factor estimates** and **geolocation information**.
- The objective of the study is to explore and prepare the dataset to investigate and understand the **cancer clusters** across countries by **death rate**, **lifestyle factors** and **geolocation**.

# RESULTS – DATA DESCRIPTION



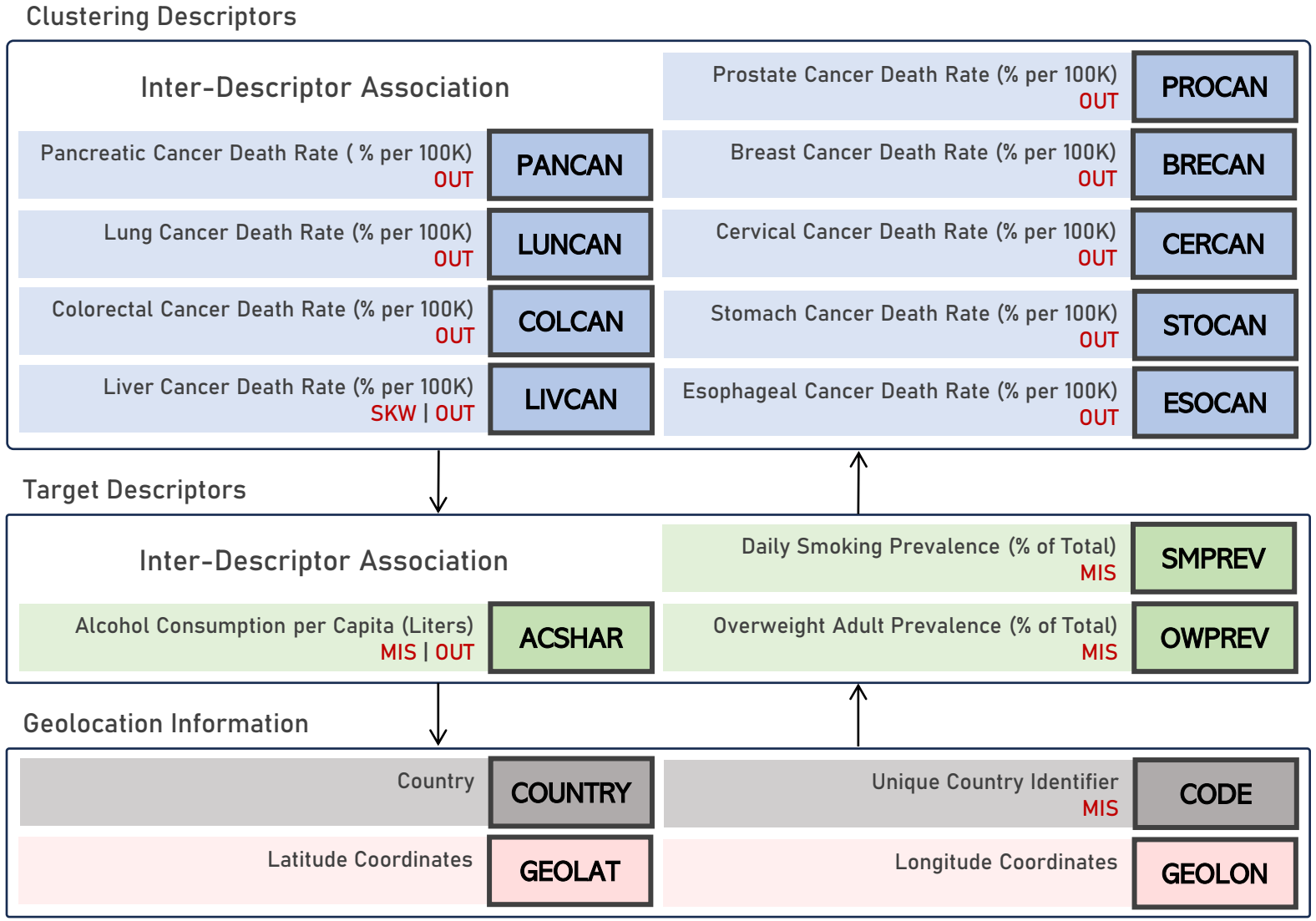

- Original data consisted of:
  - 208 observation rows
  - 9 numeric clustering descriptor columns
  - 3 numeric target descriptor columns
  - 2 object metadata column
  - 2 numeric metadata column
- Numeric data are of different scales.
- Clustering and target descriptors are assumed to be internally associated and related with geolocation information

**16** Descriptors + Metadata

**208** Observations

- Numeric Clustering Descriptor
- Numeric Target Descriptor
- Object Metadata
- Numeric Metadata

# RESULTS – DATA QUALITY ASSESSMENT




- Data quality issues identified included:
  - Missing data
  - Skewed distributions
  - High outlier ratio
- Data preprocessing is needed to address data quality issues.
- Correlation analysis is needed to confirm sufficient association between descriptors to enable the presence of underlying structures within the data and allow the natural groupings based on the similarity or dissimilarity of the observations

**16** Descriptors + Metadata

**208** Observations

- Numeric Clustering Descriptor
- Numeric Target Descriptor
- Object Metadata
- Numeric Metadata

MIS High Missing Data
SKW Skewed Distribution
NZV Near-Zero Variance
MUL High Multicollinearity
OUT High Outlier Ratio

# RESULTS – DATA PREPROCESSING

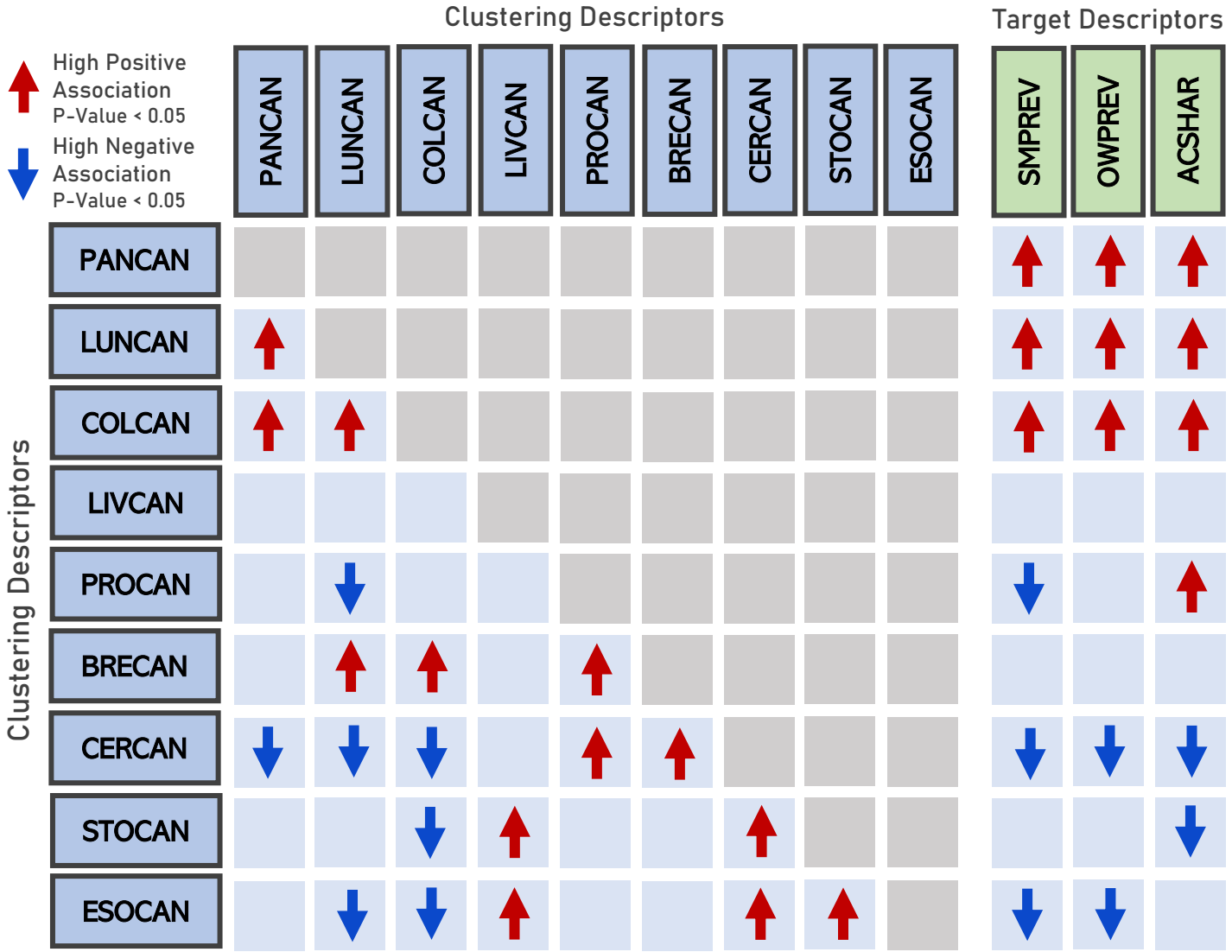
| Descriptors + Metadata | DATA PREPROCESSING STEPS | Observations |
|---|---|---|
| 16 | | 208 |
| 16 | High Column-Wise Missing Data: Column Fill Rate<100% \| High Row-Wise Missing Data[A]: Row Missing Rate>0% | 183 |
| 16 | Multicollinearity Detection: Pairwise Correlation Coefficient>0.90 | 183 |
| 16 | Skewness Detection \| Outlier Detection: Skewness>\|3\|, Value>75$^{th}$ Percentile+1.5 IQR, Value<25$^{th}$ Percentile-1.5 IQR | 183 |
| 16 | Shape Transformation: Yeo-Johnson Transformation for Skewness and Outlier Reduction | 183 |
| 16 | Post-Transformation Outlier \| Skewness Detection: Outlier Count>25, Skewness>\|3\| | 183 |
| 16 | Centering \| Scaling: Standardization for Comparable Range | 183 |

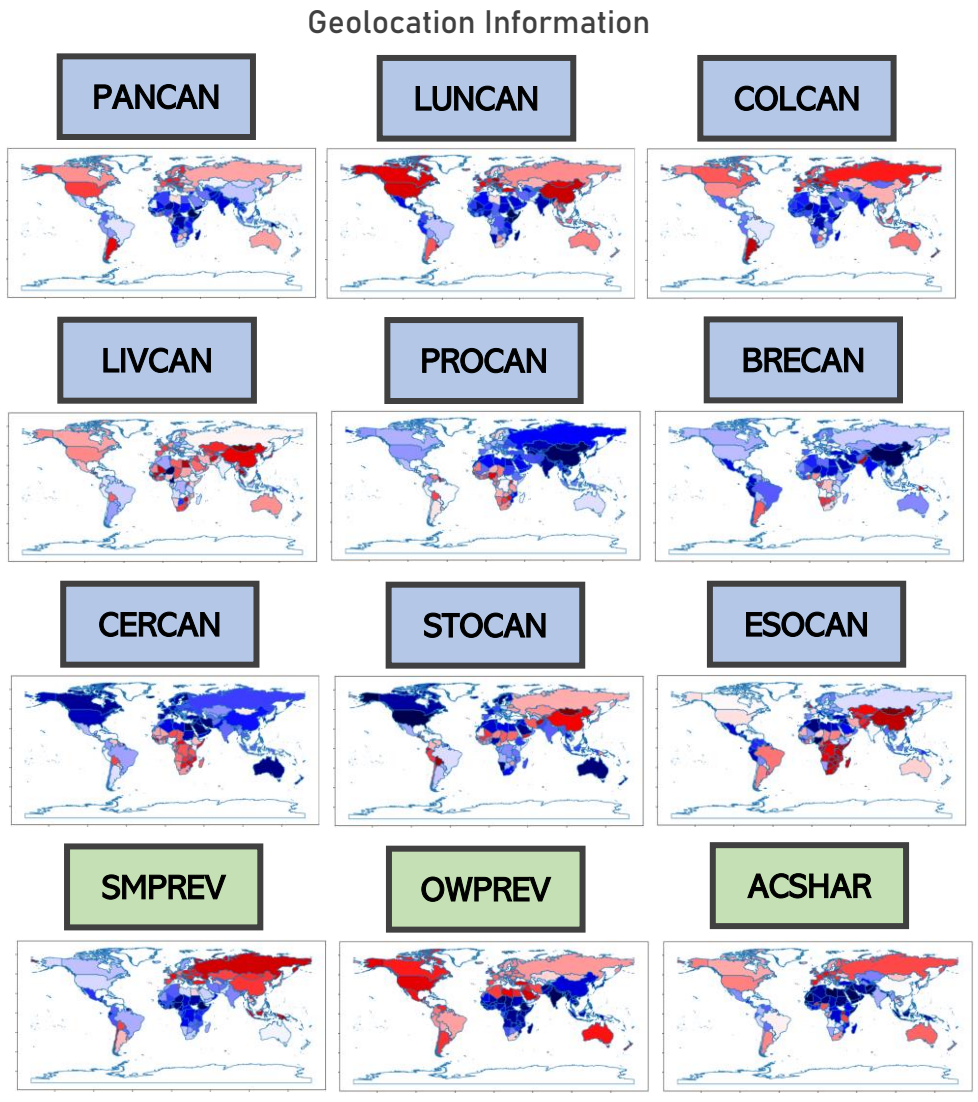
Removed 0 Descriptor
All data quality issues were addressed by all the pre-processing steps applied on the data set.

Removed 25 Observations

[A]COUNTRY: Wales, Northern Ireland, England, Tokelau, Scotland, United States Virgin Islands, South Sudan, San Marino, Puerto Rico, Bermuda, American Samoa, Monaco, Guam, Greenland, Northern Mariana Islands, Niue, Palau, Palestine, Taiwan, Cook Islands, Nauru, Saint Kitts and Nevis, Micronesia, Marshall Islands, Tuvalu

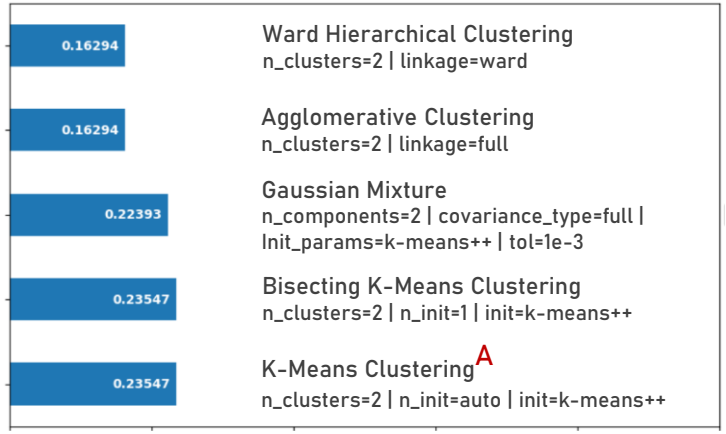
# RESULTS – DATA EXPLORATION



# RESULTS – MODEL DEVELOPMENT

## Silhouette Score by Candidate Models




## [A]OPTIMAL MODEL

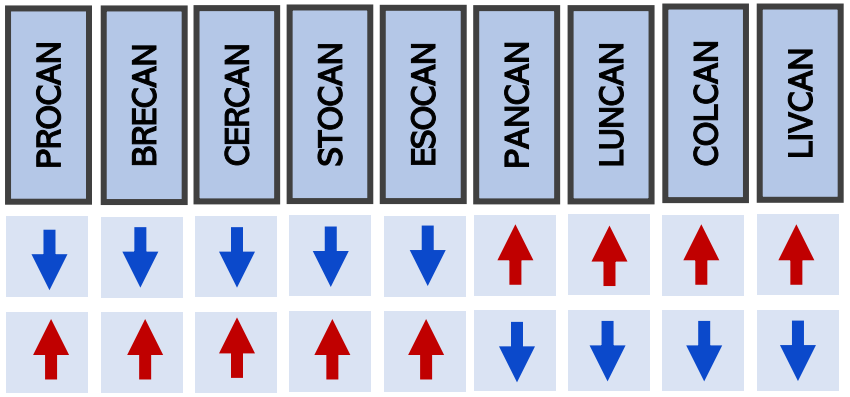
**K-Means**

n_clusters=2
n_init=auto
init=k-means++

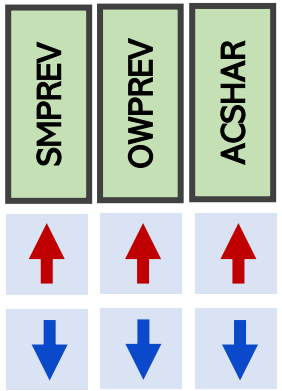
Cluster 0 :
HIGH_PAN_LUN_COL_LIV_CAN
Higher Values | Lower Values

Cluster 1 :
HIGH_PRO_BRE_CER_STO_ESO_CAN
Higher Values | Lower Values

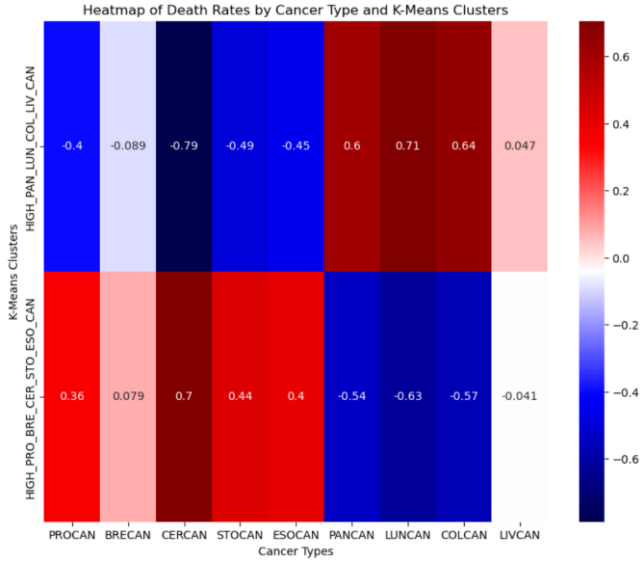
## Clustering Descriptors



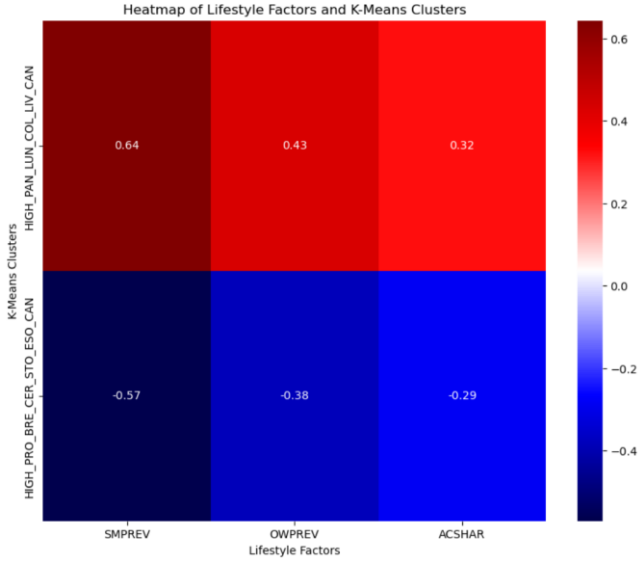

## Target Descriptors




## Cluster Characteristics by Clustering Descriptors



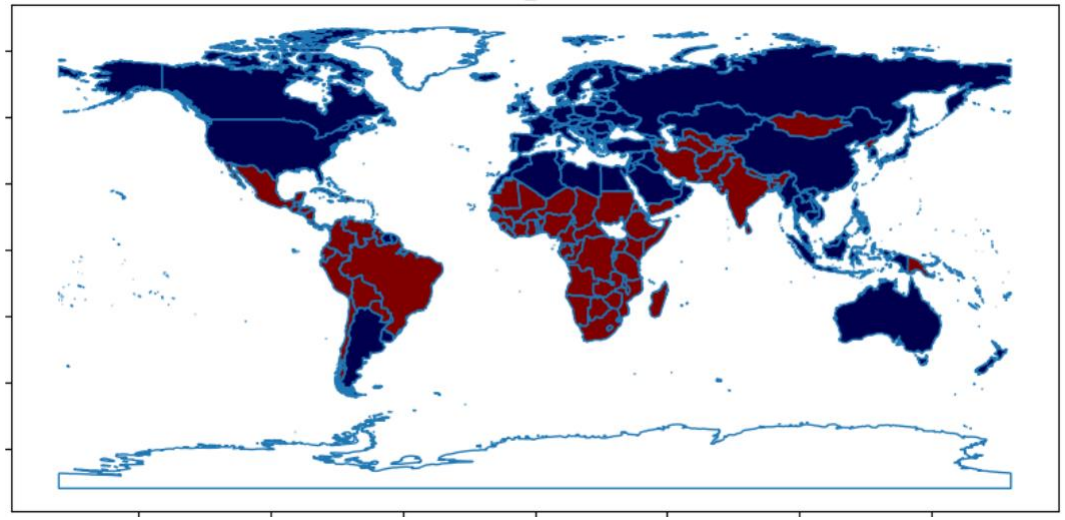

## Cluster Characteristics by Target Descriptors




## Cluster Characteristics by Geolocation Information

Cluster 0 :
North America, Europe, Non-South Asia, Australia

Cluster 1 :
Central America, South America, Africa, South Asia



Section 3

# Detailed Findings

# RESULTS – DATA QUALITY ASSESSMENT

- **Findings** ( GitHub URL: Python Notebook | Markdown Presentation )

1. No duplicated rows observed.
2. Missing data noted for 4 variables with Null.Count>0 and Fill.Rate<1.0.
   - CODE: Null.Count = 5, Fill.Rate = 0.976
   - SMPREV: Null.Count = 22, Fill.Rate = 0.894
   - OWPREV: Null.Count = 17, Fill.Rate = 0.918
   - ACSHAR: Null.Count = 21, Fill.Rate = 0.899
3. Missing data noted for 25 observations noted with Missing.Rate>0.0.
   - COUNTRY=Wales: Missing.Rate= 0.250
   - COUNTRY=Northern Ireland: Missing.Rate= 0.250
   - COUNTRY=England: Missing.Rate= 0.250
   - COUNTRY=Tokelau: Missing.Rate= 0.250
   - COUNTRY=Scotland: Missing.Rate= 0.250
   - COUNTRY=American Samoa: Missing.Rate= 0.187
   - COUNTRY=United States Virgin Islands: Missing.Rate= 0.187
   - COUNTRY=South Sudan: Missing.Rate= 0.187
   - COUNTRY=San Marino: Missing.Rate= 0.187
   - COUNTRY=Puerto Rico: Missing.Rate= 0.187
   - COUNTRY=Bermuda: Missing.Rate= 0.187
   - COUNTRY=Northern Mariana Islands: Missing.Rate= 0.187
   - COUNTRY=Monaco: Missing.Rate= 0.187
   - COUNTRY=Guam: Missing.Rate= 0.187
   - COUNTRY=Greenland: Missing.Rate= 0.187
   - COUNTRY=Niue: Missing.Rate= 0.125
   - COUNTRY=Palau: Missing.Rate= 0.125
   - COUNTRY=Palestine: Missing.Rate= 0.125
   - COUNTRY=Taiwan: Missing.Rate= 0.125
   - COUNTRY=Cook Islands: Missing.Rate= 0.125
   - COUNTRY=Nauru: Missing.Rate= 0.062
   - COUNTRY=Micronesia: Missing.Rate= 0.062
   - COUNTRY=Saint Kitts and Nevis: Missing.Rate= 0.062
   - COUNTRY=Marshall Islands: Missing.Rate= 0.062
   - COUNTRY=Tuvalu: Missing.Rate= 0.062
4. No low variance observed for any variable with First.Second.Mode.Ratio>5.
5. No low variance observed for any variable with Unique.Count.Ratio>10.
6. High skewness observed for 1 variable with Skewness>3 or Skewness<(-3).
   - LIVCAN: Skewness = +9.113

# RESULTS – DATA CLEANING

- **Findings** ( GitHub URL: Python Notebook | Markdown Presentation )

1. Subsets of rows with high rates of missing data were removed from the dataset:
   - 25 rows with Missing.Rate>0.0 were exluded for subsequent analysis.
     - COUNTRY=Wales: Missing.Rate= 0.250
     - COUNTRY=Northern Ireland: Missing.Rate= 0.250
     - COUNTRY=England: Missing.Rate= 0.250
     - COUNTRY=Tokelau: Missing.Rate= 0.250
     - COUNTRY=Scotland: Missing.Rate= 0.250
     - COUNTRY=American Samoa: Missing.Rate= 0.187
     - COUNTRY=United States Virgin Islands: Missing.Rate= 0.187
     - COUNTRY=South Sudan: Missing.Rate= 0.187
     - COUNTRY=San Marino: Missing.Rate= 0.187
     - COUNTRY=Puerto Rico: Missing.Rate= 0.187
     - COUNTRY=Bermuda: Missing.Rate= 0.187
     - COUNTRY=Northern Mariana Islands: Missing.Rate= 0.187
     - COUNTRY=Monaco: Missing.Rate= 0.187
     - COUNTRY=Guam: Missing.Rate= 0.187
     - COUNTRY=Greenland: Missing.Rate= 0.187
     - COUNTRY=Niue: Missing.Rate= 0.125
     - COUNTRY=Palau: Missing.Rate= 0.125
     - COUNTRY=Palestine: Missing.Rate= 0.125
     - COUNTRY=Taiwan: Missing.Rate= 0.125
     - COUNTRY=Cook Islands: Missing.Rate= 0.125
     - COUNTRY=Nauru: Missing.Rate= 0.062
     - COUNTRY=Micronesia: Missing.Rate= 0.062
     - COUNTRY=Saint Kitts and Nevis: Missing.Rate= 0.062
     - COUNTRY=Marshall Islands: Missing.Rate= 0.062
     - COUNTRY=Tuvalu: Missing.Rate= 0.062
2. No variables were removed due to zero or near-zero variance.

2. No variables were removed due to zero or near-zero variance.
3. The cleaned dataset is comprised of:
   - **183 rows** (observations)
   - **16 columns** (variables)
     - **2/16 metadata** (object)
       - COUNTRY
       - CODE
     - **2/16 metadata** (numeric)
       - GEOLAT
       - GEOLON
     - **9/16 clustering descriptors** (numeric)
       - PROCAN
       - BRECAN
       - CERCAN
       - STOCAN
       - ESOCAN
       - PANCAN
       - LUNCAN
       - COLCAN
       - LIVCAN
     - **3/16 target descriptors** (numeric)
       - SMPREV
       - OWPREV
       - ACSHAR

# RESULTS – OUTLIER TREATMENT

- **Findings** ( GitHub URL: Python Notebook | Markdown Presentation )

1. High number of outliers observed for 2 numeric variables with Outlier.Ratio>0.10 and marginal to high Skewness.
   - ESOCAN: Outlier.Count = 24, Outlier.Ratio = 0.131, Skewness=+2.092
   - LIVCAN: Outlier.Count = 19, Outlier.Ratio = 0.104, Skewness=+8.716
2. Minimal number of outliers observed for 8 numeric variables with Outlier.Ratio<0.10 and normal Skewness.
   - PROCAN: Outlier.Count = 11, Outlier.Ratio = 0.060, Skewness=+2.246
   - BRECAN: Outlier.Count = 8, Outlier.Ratio = 0.044, Skewness=+1.958
   - STOCAN: Outlier.Count = 6, Outlier.Ratio = 0.033, Skewness=+2.086
   - CERCAN: Outlier.Count = 2, Outlier.Ratio = 0.011, Skewness=+1.989
   - LUNCAN: Outlier.Count = 2, Outlier.Ratio = 0.011, Skewness=+0.857
   - COLCAN: Outlier.Count = 2, Outlier.Ratio = 0.011, Skewness=+0.820
   - PANCAN: Outlier.Count = 1, Outlier.Ratio = 0.006, Skewness=+0.599
   - ACSHAR: Outlier.Count = 1, Outlier.Ratio = 0.006, Skewness=+0.337

# RESULTS – COLLINEARITY

- **Findings** ( GitHub URL: Python Notebook | Markdown Presentation )

1. Majority of the numeric variables reported moderate to high correlation which were statistically significant.
2. Among pairwise combinations of numeric variables on cancer death rates, high Pearson.Correlation.Coefficient values were noted for:
   - PANCAN and COLCAN: Pearson.Correlation.Coefficient = +0.754
   - LUNCAN and COLCAN: Pearson.Correlation.Coefficient = +0.701
3. Among the numeric variables on cancer death rates, the highest Pearson.Correlation.Coefficient values against the SMPREV variable were noted for:
   - SMPREV and LUNCAN: Pearson.Correlation.Coefficient = +0.642
   - SMPREV and COLCAN: Pearson.Correlation.Coefficient = +0.413
4. Among the numeric variables on cancer death rates, the highest Pearson.Correlation.Coefficient values against the OWPREV variable were noted for:
   - OWPREV and PANCAN: Pearson.Correlation.Coefficient = +0.521
   - OWPREV and COLCAN: Pearson.Correlation.Coefficient = +0.410
5. Among the numeric variables on cancer death rates, the highest Pearson.Correlation.Coefficient values against the ACSHAR variable were noted for:
   - ACSHAR and COLCAN: Pearson.Correlation.Coefficient = +0.582
   - ACSHAR and PANCAN: Pearson.Correlation.Coefficient = +0.575
6. No any variable was removed due to extreme multicollinearity.

# RESULTS – SHAPE TRANSFORMATION

- Findings ( GitHub URL: Python Notebook | Markdown Presentation )

1. A Yeo-Johnson transformation was applied to all numeric variables to improve distributional shape.
2. All variables achieved symmetrical distributions with minimal outliers after transformation.

# RESULTS – CENTERING AND SCALING

- Findings ( GitHub URL: Python Notebook | Markdown Presentation )

1. All numeric variables were transformed using the standardization method to achieve a comparable scale between values.

# RESULTS – EXPLORATORY DATA ANALYSIS

- **Findings** ( GitHub URL: Python Notebook | Markdown Presentation )

1. Bivariate analysis identified individual descriptors with generally linear relationship to the target descriptor based on visual inspection.
2. Linear relationships for the following descriptors and SMPREV variables were observed:
   - LUNCAN
   - COLCAN
3. Linear relationships for the following descriptors and OWPREV variables were observed:
   - PANCAN
   - COLCAN
4. Linear relationships for the following descriptors and ACSHAR variables were observed:
   - COLCAN
   - PANCAN

# RESULTS – HYPOTHESIS TESTING

- ## Findings ( GitHub URL: Python Notebook | Markdown Presentation )

1. The relationship between the numeric descriptors to the SMPREV, OWPREV and ACSHAR target descriptors were statistically evaluated using the following hypotheses:
   - **Null**: Pearson correlation coefficient is equal to zero
   - **Alternative**: Pearson correlation coefficient is not equal to zero
2. There is sufficient evidence to conclude of a statistically significant linear relationship between the SMPREV target descriptor and 6 of the 9 numeric descriptors given their high Pearson correlation coefficient values with reported low p-values less than the significance level of 0.05.
   - LUNCAN: Pearson.Correlation.Coefficient=+0.684, Correlation.PValue=0.000
   - CERCAN: Pearson.Correlation.Coefficient=-0.494, Correlation.PValue=0.000
   - PROCAN: Pearson.Correlation.Coefficient=-0.438, Correlation.PValue=0.000
   - COLCAN: Pearson.Correlation.Coefficient=+0.420, Correlation.PValue=0.000
   - PANCAN: Pearson.Correlation.Coefficient=+0.343, Correlation.PValue=0.000
   - ESOCAN: Pearson.Correlation.Coefficient=-0.262, Correlation.PValue=0.001
3. There is sufficient evidence to conclude of a statistically significant linear relationship between the OWPREV target descriptor and 5 of the 9 numeric descriptors given their high Pearson correlation coefficient values with reported low p-values less than the significance level of 0.05.
   - PANCAN: Pearson.Correlation.Coefficient=+0.684, Correlation.PValue=0.000
   - CERCAN: Pearson.Correlation.Coefficient=-0.494, Correlation.PValue=0.000
   - LUNCAN: Pearson.Correlation.Coefficient=+0.438, Correlation.PValue=0.000
   - COLCAN: Pearson.Correlation.Coefficient=+0.420, Correlation.PValue=0.000
   - ESOCAN: Pearson.Correlation.Coefficient=-0.343, Correlation.PValue=0.000
4. There is sufficient evidence to conclude of a statistically significant linear relationship between the ACSHAR target descriptor and 5 of the 9 numeric descriptors given their high Pearson correlation coefficient values with reported low p-values less than the significance level of 0.05.
   - PANCAN: Pearson.Correlation.Coefficient=+0.567, Correlation.PValue=0.000
   - COLCAN: Pearson.Correlation.Coefficient=+0.564, Correlation.PValue=0.000
   - LUNCAN: Pearson.Correlation.Coefficient=+0.399, Correlation.PValue=0.000
   - PROCAN: Pearson.Correlation.Coefficient=+0.201, Correlation.PValue=0.010
   - BRECAN: Pearson.Correlation.Coefficient=+0.174, Correlation.PValue=0.026

# RESULTS – PRE-MODELLING DATA ANALYSIS

- **Findings** ( GitHub URL: Python Notebook | Markdown Presentation )

1. Among the 9 numeric descriptors, LIVCAN and STOCAN have not demonstrated a statistically significant linear relationship between the SMPREV, OWPREV nad , ACSHAR target descriptors.
2. All 9 numeric descriptors were however retained for the clustering analysis.

# RESULTS – K-MEANS CLUSTERING

- ## Findings ( GitHub URL: Python Notebook | Markdown Presentation )

K-Means Clustering groups similar data points together into clusters by minimizing the mean distance between geometric points. The algorithm iteratively partitions data sets into a fixed number of non-overlapping k subgroups or clusters wherein each data point belongs to the cluster with the nearest mean cluster center. The process begins by initializing all the coordinates into a pre-defined k number of cluster centers. With every pass of the algorithm, each point is assigned to its nearest cluster center. The cluster centers are then updated to be the centers of all the points assigned to it in that pass. This is performed by re-calculating the cluster centers as the average of the points in each respective cluster. The algorithm repeats until there's a minimum change of the cluster centers from the last iteration.

Silhouette Score assesses the quality of clusters created by a clustering algorithm. It measures how well-separated the clusters are and how similar each data point in a cluster is to the other points in the same cluster compared to the nearest neighboring cluster. The silhouette score ranges from -1 to 1, where a higher value indicates better-defined clusters. The silhouette method requires the computation of the silhouette scores for each data point which is the average dissimilarity of the data point with all other data points in the next-nearest cluster minus the average dissimilarity of the data point to points in the same cluster and divided by the larger of the two numbers. The overall silhouette score for the clustering is the average of the silhouette scores for all data points.

1. The k-means clustering model from the **sklearn.cluster** Python library API was implemented.
2. The model contains 3 hyperparameters:
   - n_clusters = number of clusters to form as well as the number of centroids to generate made to vary between 2 to 9
   - n_init = number of times the k-means algorithm is run with different centroid seeds held constant at a value of auto
   - init = method for initialization held constant at a value equal to k-means++
3. Hyperparameter tuning was conducted on the data with optimal model performance using the silhouette score determined for:
   - n_clusters = 2
   - n_init = auto
   - init = k-means++
4. The apparent model performance of the optimal model is summarized as follows:
   - **Silhouette Score** = 0.2355

# RESULTS – BISECTING K-MEANS CLUSTERING

- ## Findings ( GitHub URL: Python Notebook | Markdown Presentation )

Bisecting K-Means Clustering is a variant of the traditional K-Means algorithm which iteratively splits clusters into two parts until the desired number of clusters is reached. It is a hierarchical clustering approach that uses a divisive strategy to build a hierarchy of clusters. The algorithm starts with the entire dataset as the initial cluster. The standard K-Means algorithm is implemented to the selected cluster, splitting it into two sub-clusters. Both steps are repeated until the desired number of clusters is reached. In cases when there are multiple clusters present, the algorithm selects the cluster with the largest variance. This results in a hierarchical structure of clusters, and the process can be stopped at any desired level of granularity.

Silhouette Score assesses the quality of clusters created by a clustering algorithm. It measures how well-separated the clusters are and how similar each data point in a cluster is to the other points in the same cluster compared to the nearest neighboring cluster. The silhouette score ranges from -1 to 1, where a higher value indicates better-defined clusters. The silhouette method requires the computation of the silhouette scores for each data point which is the average dissimilarity of the data point with all other data points in the next-nearest cluster minus the average dissimilarity of the data point to points in the same cluster and divided by the larger of the two numbers. The overall silhouette score for the clustering is the average of the silhouette scores for all data points.

1. The bisecting k-means clustering model from the **sklearn.cluster** Python library API was implemented.
2. The model contains 3 hyperparameters:
   - n_clusters = number of clusters to form as well as the number of centroids to generate made to vary between 2 to 9
   - n_init = number of time the inner k-means algorithm will be run with different centroid seeds in each bisection held constant at a value of 1
   - init = method for initialization held constant at a value equal to k-means++
3. Hyperparameter tuning was conducted on the data with optimal model performance using the silhouette score determined for:
   - n_clusters = 2
   - n_init = 1
   - init = k-means++
4. The apparent model performance of the optimal model is summarized as follows:
   - **Silhouette Score** = 0.2355

# RESULTS – GAUSSIAN MIXTURE CLUSTERING

- ## Findings ( GitHub URL: Python Notebook | Markdown Presentation )

Gaussian Mixture Clustering is a probabilistic model that assumes all the data points are generated from a mixture of a finite number of Gaussian distributions with unknown parameters incorporating information about the covariance structure of the data as well as the centers of the latent Gaussians. The algorithm involves initializing the parameters of the Gaussian components using K-means clustering to get initial estimates for the means and the identity matrix as a starting point for the covariance matrices. The expectation-maximization process is applied by calculating the probability of each data point belonging to each Gaussian component using the Bayes' theorem for the expectation step, and updating the parameters of the Gaussian components based on the weighted sum of the data points based on the probabilities determined for the maximization step. Convergence is checked by evaluating whether the log-likelihood of the data has stabilized or reached a maximum. Both steps are iterated until the criteria is met. After convergence, each data point is assigned to the cluster with the highest probability.

Silhouette Score assesses the quality of clusters created by a clustering algorithm. It measures how well-separated the clusters are and how similar each data point in a cluster is to the other points in the same cluster compared to the nearest neighboring cluster. The silhouette score ranges from -1 to 1, where a higher value indicates better-defined clusters. The silhouette method requires the computation of the silhouette scores for each data point which is the average dissimilarity of the data point with all other data points in the next-nearest cluster minus the average dissimilarity of the data point to points in the same cluster and divided by the larger of the two numbers. The overall silhouette score for the clustering is the average of the silhouette scores for all data points.

1. The gaussian mixture clustering model from the **sklearn.mixture** Python library API was implemented.
2. The model contains 4 hyperparameters:
   - n_components = number of mixture components made to vary between 2 to 9
   - covariance_type = type of covariance parameters to use held constant at a value equal to full (each component has its own general covariance matrix)
   - init_params = method for initialization held constant at a value equal to k-means++
   - tol = convergence threshold held constant at a value of 1e-3
3. Hyperparameter tuning was conducted on the data with optimal model performance using the silhouette score determined for:
   - n_components = 2
   - covariance_type = full
   - init_params = k-means++
   - tol = 1e-3
4. The apparent model performance of the optimal model is summarized as follows:
   - **Silhouette Score** = 0.2239

# RESULTS – AGGLOMERATIVE CLUSTERING

- ## Findings ( GitHub URL: Python Notebook | Markdown Presentation )

Agglomerative Clustering builds a hierarchy of clusters. In this algorithm, each data point starts as its own cluster, and the algorithm merges clusters iteratively until a stopping criterion is met. The algorithm starts with each data point as a singleton cluster with the number of initial clusters is equal to the number of data points. The pairwise distance matrix is calculated between all clusters using complete linkage determined as the maximum distance between any two points in the two clusters. The two clusters that have the minimum distance according to the linkage criterion are identified and merged in the next step. The distances between new clusters and all other clusters are recalculated. All previous steps are repeated until the desired number of clusters is reached or until a stopping criterion is met.

Silhouette Score assesses the quality of clusters created by a clustering algorithm. It measures how well-separated the clusters are and how similar each data point in a cluster is to the other points in the same cluster compared to the nearest neighboring cluster. The silhouette score ranges from -1 to 1, where a higher value indicates better-defined clusters. The silhouette method requires the computation of the silhouette scores for each data point which is the average dissimilarity of the data point with all other data points in the next-nearest cluster minus the average dissimilarity of the data point to points in the same cluster and divided by the larger of the two numbers. The overall silhouette score for the clustering is the average of the silhouette scores for all data points.

1. The agglomerative clustering model from the **sklearn.cluster** Python library API was implemented.
2. The model contains 2 hyperparameters:
   - n_cluster = number of clusters to find made to vary between 2 to 9
   - linkage = linkage criterion used to determine which distance to use between sets of observation held constant at a value equal to full (minimizes the maximum distance between observations of pairs of clusters)
3. Hyperparameter tuning was conducted on the data with optimal model performance using the silhouette score determined for:
   - n_cluster = 2
   - linkage = full
4. The apparent model performance of the optimal model is summarized as follows:
   - **Silhouette Score** = 0.1629

# RESULTS – WARD HIERARCHICAL CLUSTERING

- ## Findings ( GitHub URL: Python Notebook | Markdown Presentation )

Ward Hierarchical Clustering creates compact, well-separated clusters by minimizing the variance within each cluster during the merging process. In this algorithm, each data point starts as its own cluster, and the algorithm merges clusters iteratively until a stopping criterion is met. The algorithm starts with each data point as a singleton cluster with the number of initial clusters is equal to the number of data points. The pairwise distance matrix is calculated between all clusters and used as a measure of dissimilarity. For each cluster, the within-cluster variance is computed which evaluates how tightly the data points within a cluster are grouped. The two clusters that, when merged, result in the smallest increase in the within-cluster variance are identified and merged in the next step. The within-cluster variance for the newly formed cluster are recalculated and the pairwise distance matrix updated. All previous steps are repeated until the desired number of clusters is reached or until a stopping criterion is met.

Silhouette Score assesses the quality of clusters created by a clustering algorithm. It measures how well-separated the clusters are and how similar each data point in a cluster is to the other points in the same cluster compared to the nearest neighboring cluster. The silhouette score ranges from -1 to 1, where a higher value indicates better-defined clusters. The silhouette method requires the computation of the silhouette scores for each data point which is the average dissimilarity of the data point with all other data points in the next-nearest cluster minus the average dissimilarity of the data point to points in the same cluster and divided by the larger of the two numbers. The overall silhouette score for the clustering is the average of the silhouette scores for all data points.

1. The ward hierarchical clustering model from the **sklearn.cluster** Python library API was implemented.
2. The model contains 2 hyperparameters:
   - n_cluster = number of clusters to find made to vary between 2 to 9
   - linkage = linkage criterion used to determine which distance to use between sets of observation held constant at a value equal to ward (minimizes the sum of squared differences and variance within all clusters)
3. Hyperparameter tuning was conducted on the data with optimal model performance using the silhouette score determined for:
   - n_cluster = 2
   - linkage = ward
4. The apparent model performance of the optimal model is summarized as follows:
   - **Silhouette Score** = 0.2148

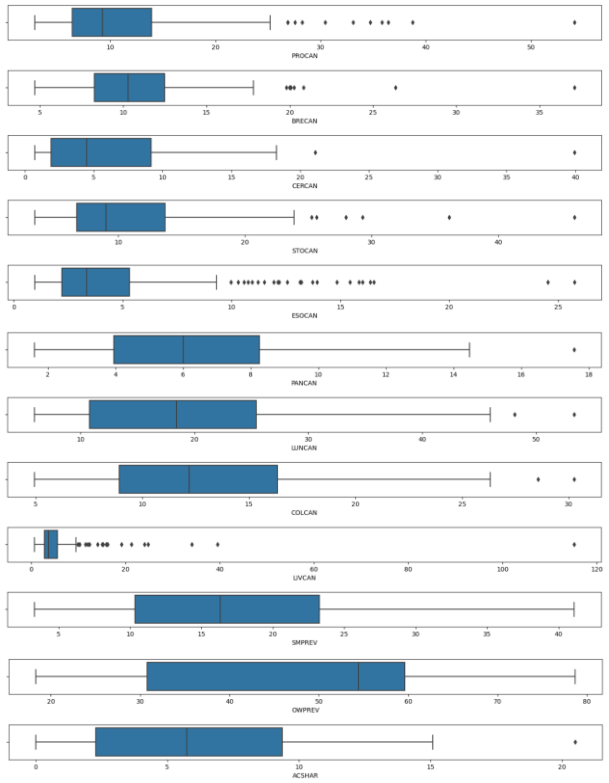
# RESULTS – MODEL SELECTION | POST-HOC

- ## Findings ( GitHub URL: Python Notebook | Markdown Presentation )

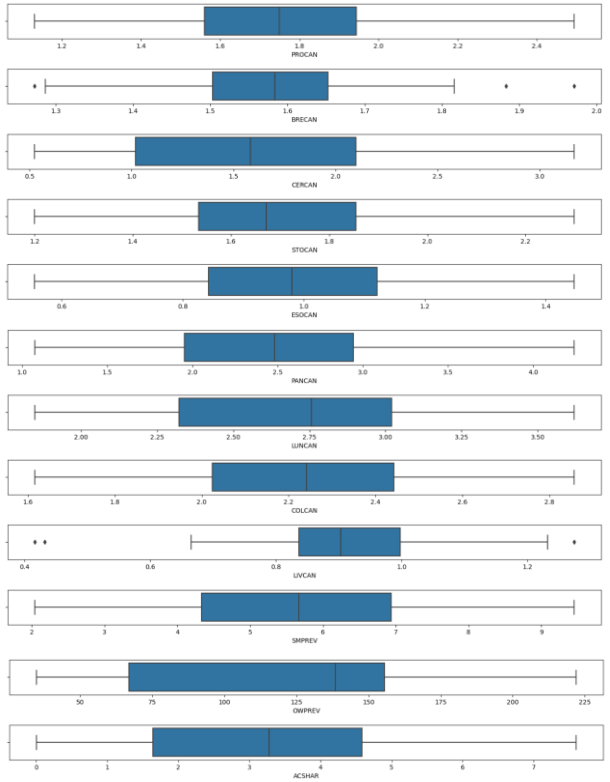
1. Among the range of cluster counts evaluated, the k-means clustering model with 2 clusters provided the most compact intra-cluster and differential inter-cluster segmentation of countries:
   - **Silhouette Score** = 0.2355
2. Among the range of cluster counts evaluated, the bisecting k-means clustering model with 2 clusters provided the most compact intra-cluster and differential inter-cluster segmentation of countries:
   - **Silhouette Score** = 0.2355
3. Among the range of cluster counts evaluated, the gaussian mixture clustering model with 2 clusters provided the most compact intra-cluster and differential inter-cluster segmentation of countries:
   - **Silhouette Score** = 0.2239
4. Among the range of cluster counts evaluated, the agglomerative clustering model with 2 clusters provided the most compact intra-cluster and differential inter-cluster segmentation of countries:
   - **Silhouette Score** = 0.1629
5. Among the range of cluster counts evaluated, the ward hierarchical clustering model with 2 clusters provided the most compact intra-cluster and differential inter-cluster segmentation of countries:
   - **Silhouette Score** = 0.2148
6. Comparing all results from the clustering models formulated, the k-means clustering model which demonstrated the highest silhouette score was selected as the final model for segmenting the countries based on similar characteristics.
   - **Silhouette Score** = 0.2355
7. Given the final model, the following segmented groupings were observed from the formulated clusters:
   - **Cluster 0: HIGH_PAN_LUN_COL_LIV_CAN** composed of countries characterized by:
     - Higher death rates for pancreatic, lung, colon and liver cancers
     - Lower death rates for prostate, breast, cervical, stomach and esophageal cancers
     - Higher smoking prevalence, overweight prevalence and alcohol consumption
     - Predominantly from North America, Europe, West Asia, Central Asia, East Asia, Southeast Asia and Australia regions
   - **Cluster 1: HIGH_PRO_BRE_CER_STO_ESO_CAN** composed of countries characterized by:
     - Higher death rates for prostate, breast, cervical, stomach and esophageal cancers
     - Lower death rates for pancreatic, lung, colon and liver cancers
     - Lower smoking prevalence, overweight prevalence and alcohol consumption
     - Predominantly from the South America, South Asia and Africa regions

# PLOTS – OUTLIER ANALYSIS

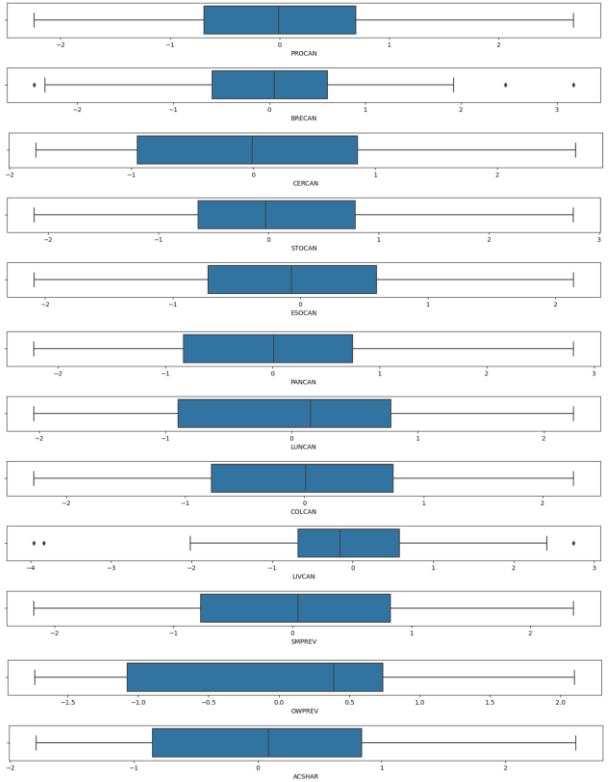
- Original Data




- Transformed Data

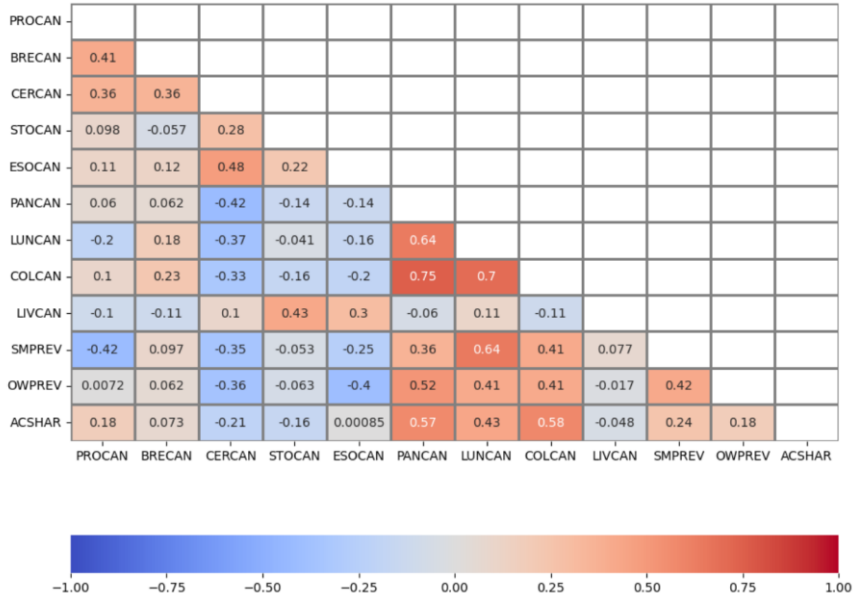



- Centered and Scaled Data

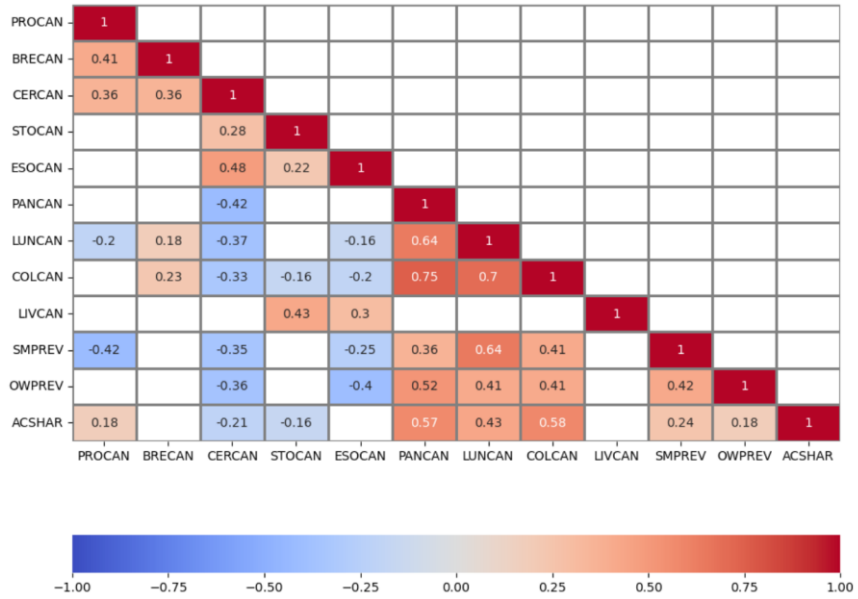


# PLOTS – CORRELATION ANALYSIS

- All Correlations

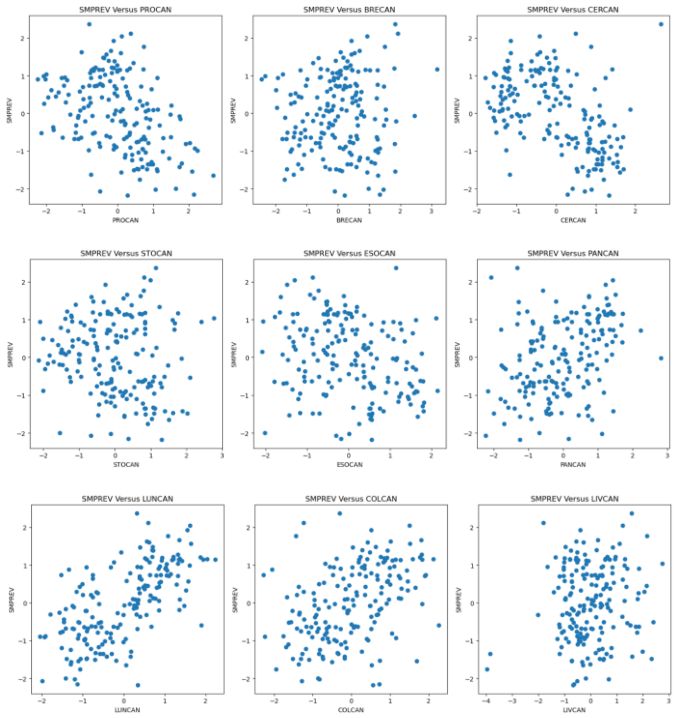


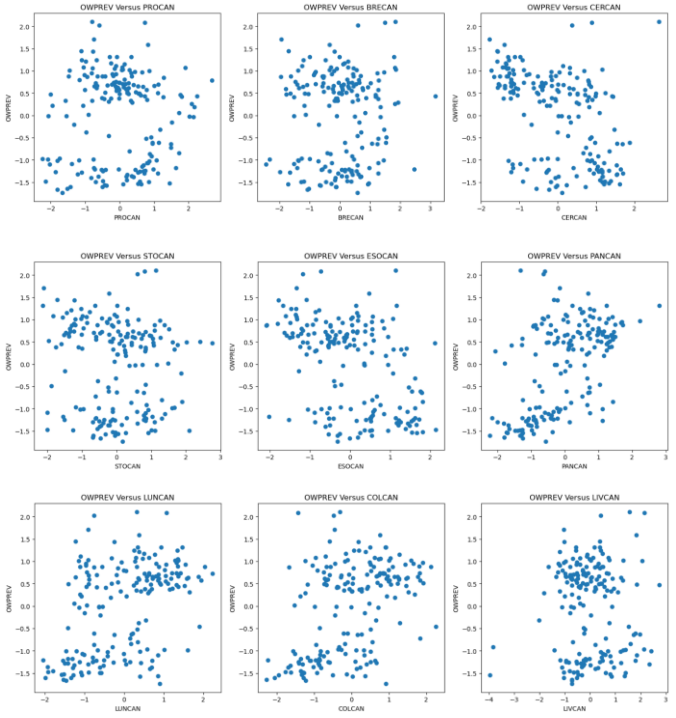

- Statistically Significant Correlations

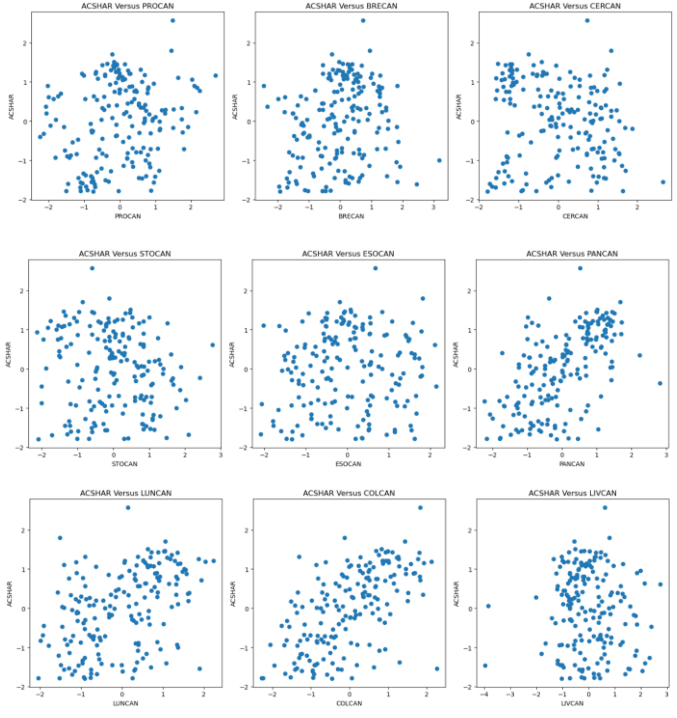


# PLOTS – EXPLORATORY DATA ANALYSIS

- Relationship Scatterplots

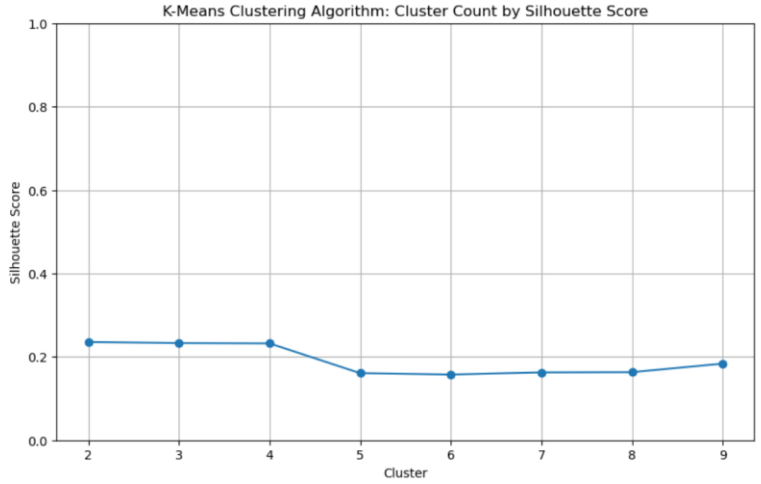


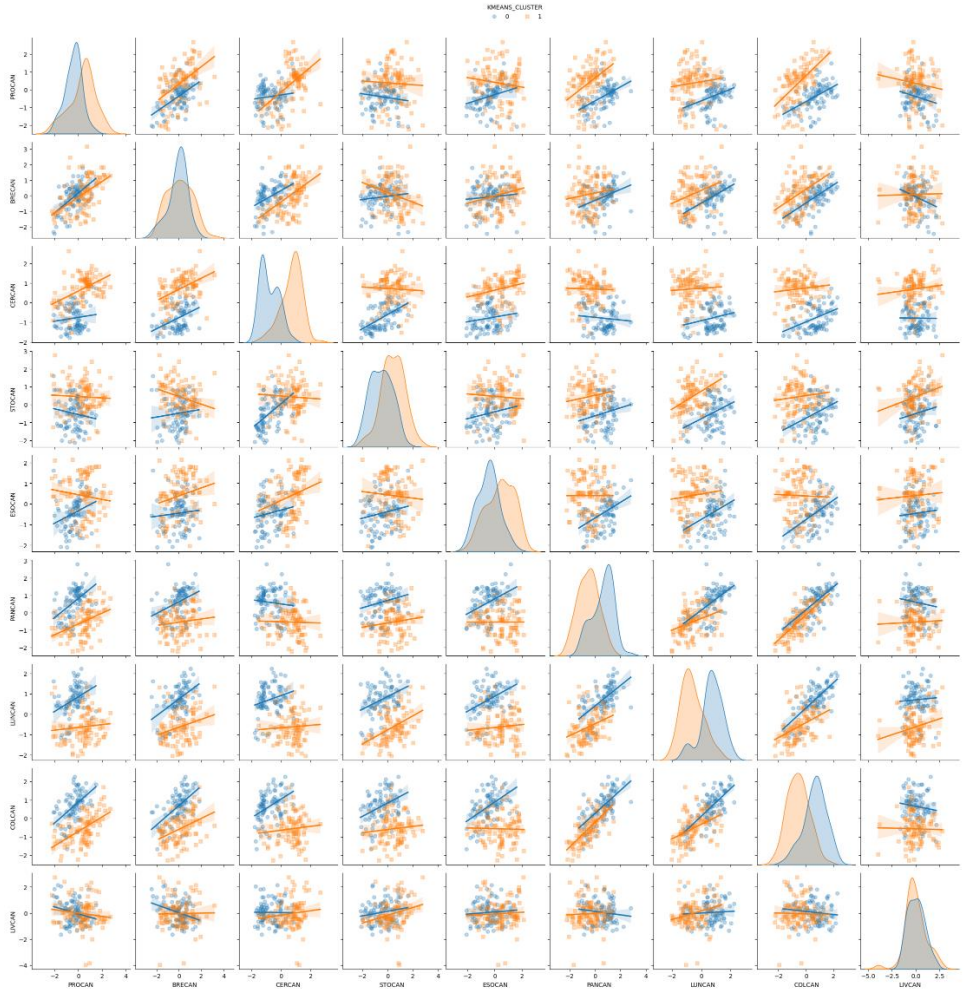



# PLOTS – CLUSTERING MODEL FORMULATION

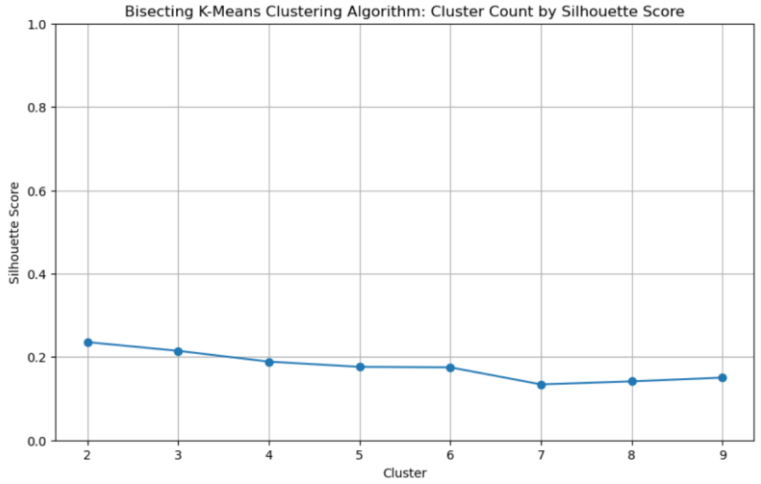
- K-Means Clustering

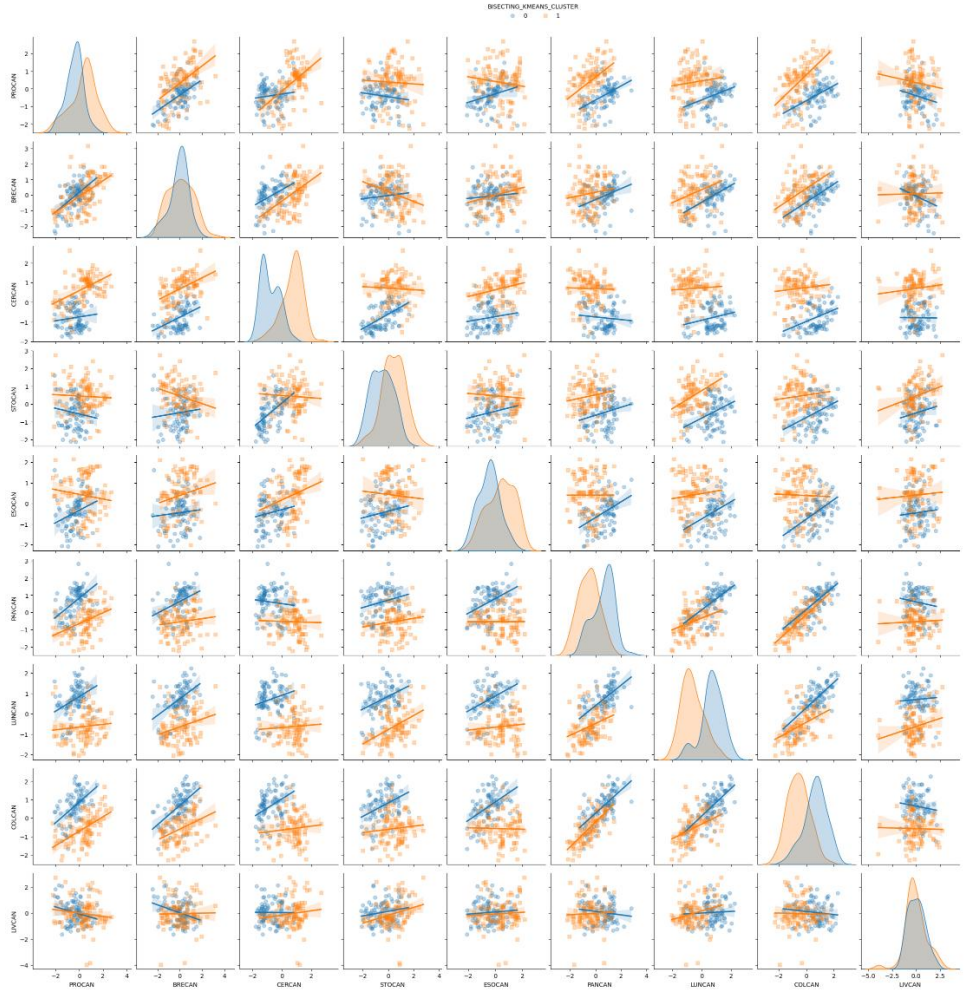


# PLOTS – CLUSTERING MODEL FORMULATION

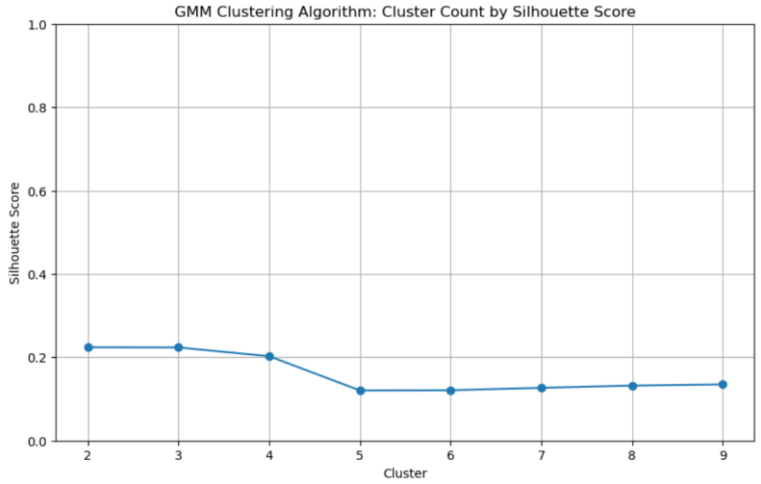
- Bisecting K-Means Clustering

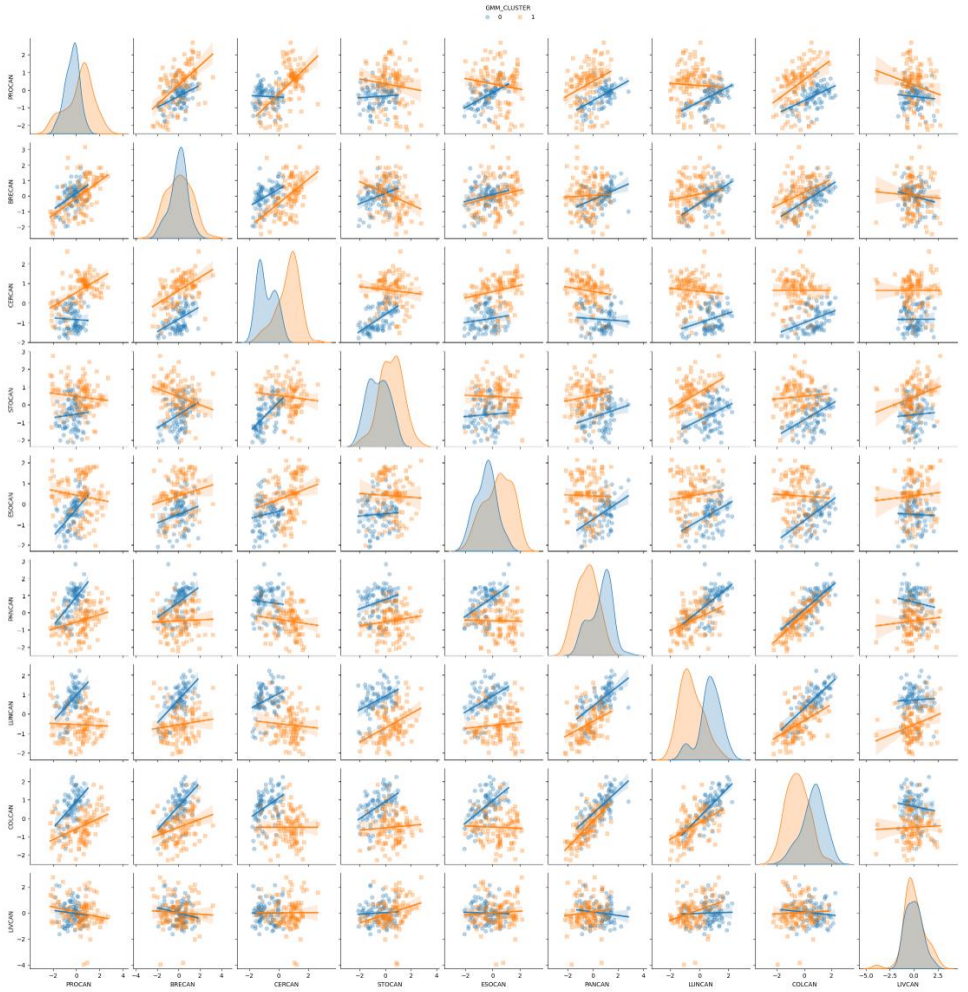


# PLOTS – CLUSTERING MODEL FORMULATION

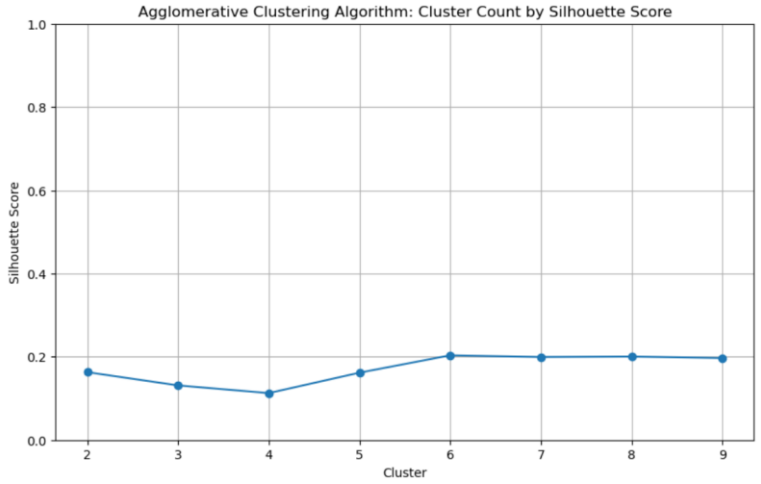
- Gaussian Mixture Model Clustering

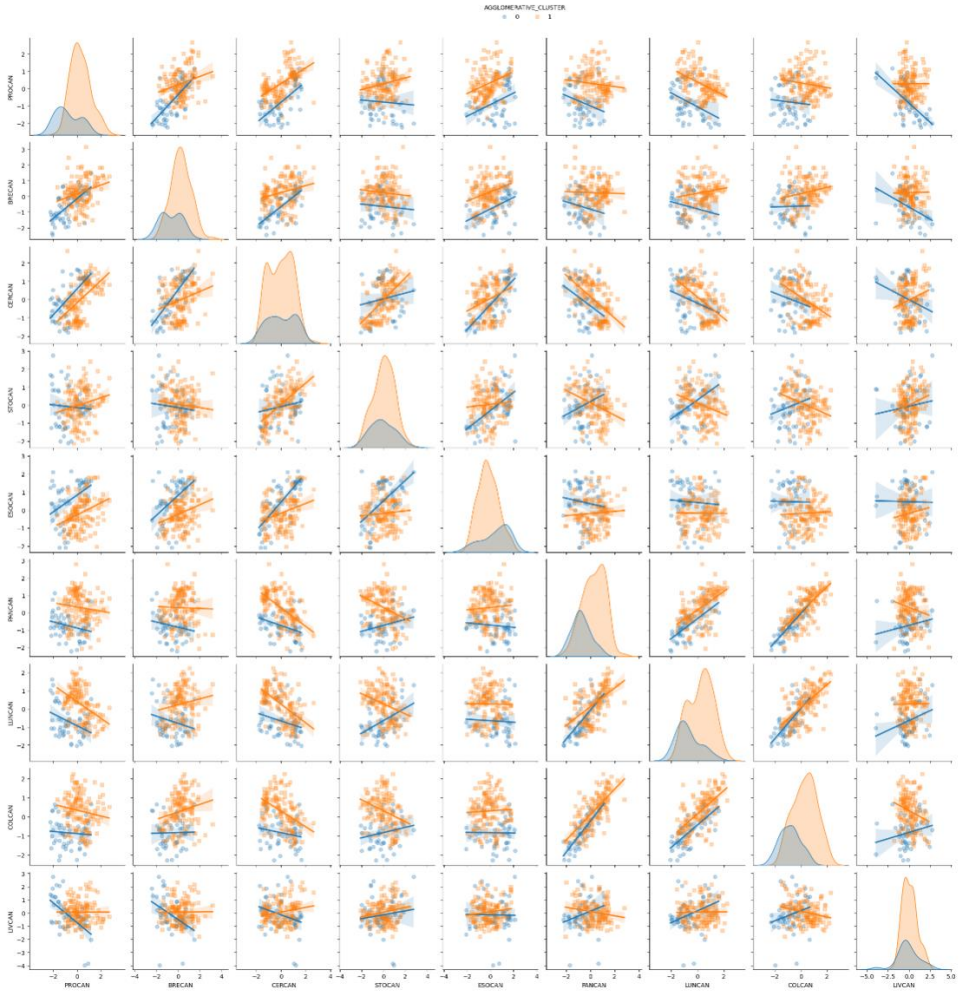


# PLOTS – CLUSTERING MODEL FORMULATION

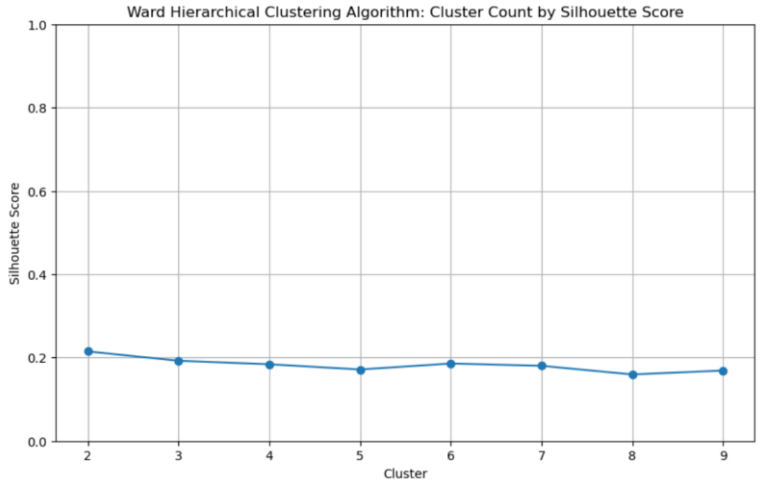
- Agglomerative Clustering

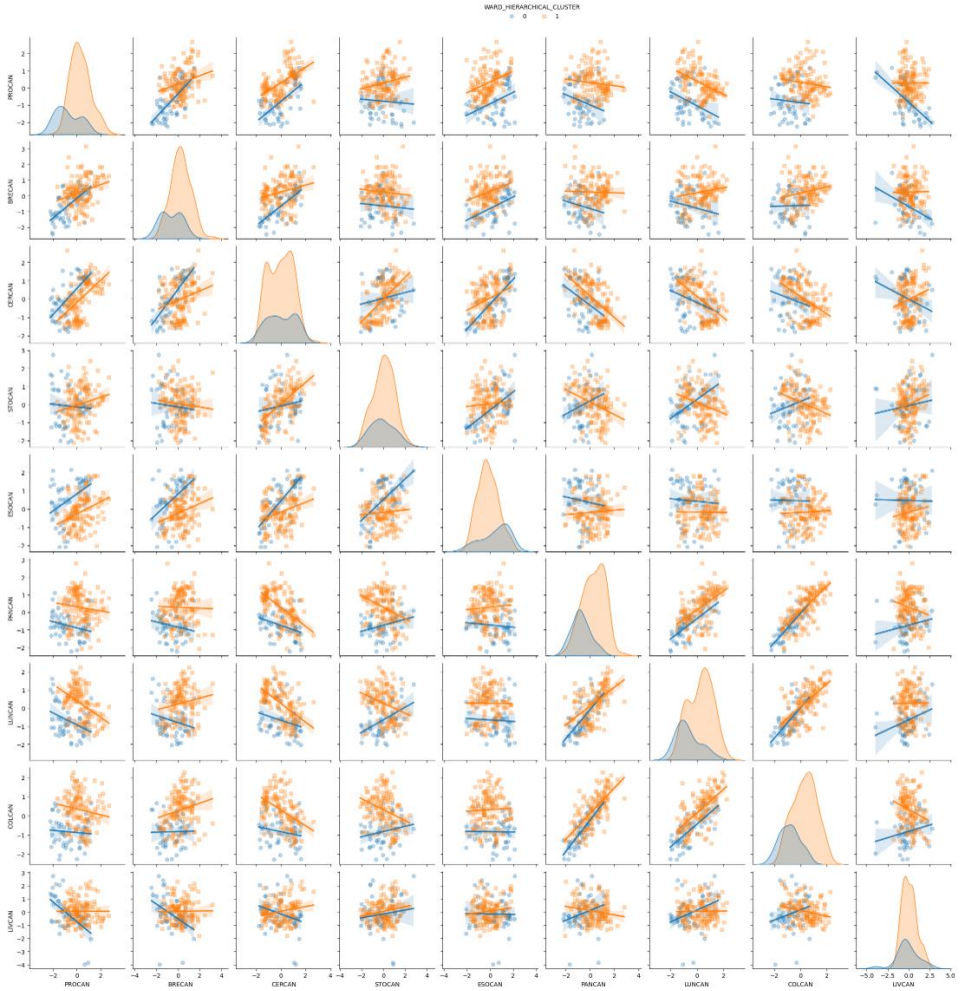


# PLOTS – CLUSTERING MODEL FORMULATION

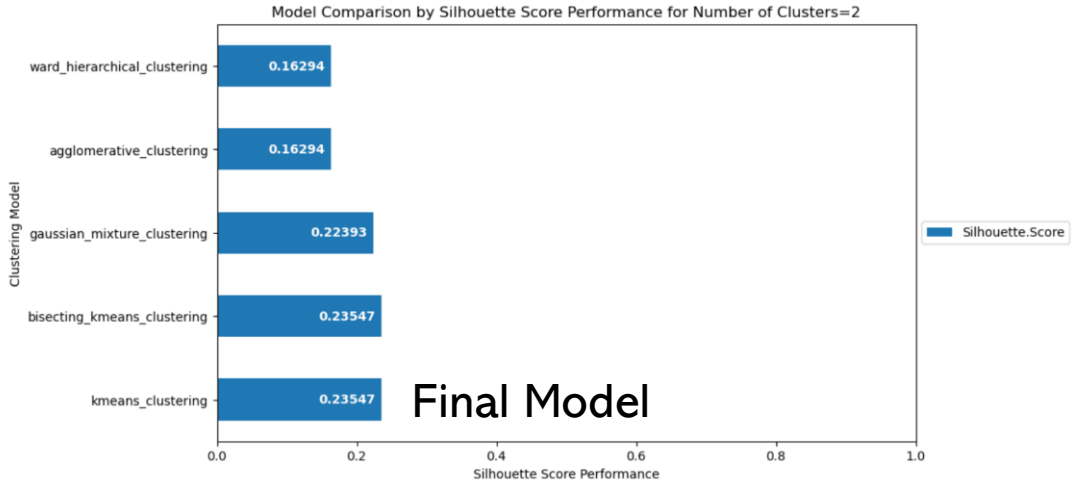
- Ward Hierarchical Clustering



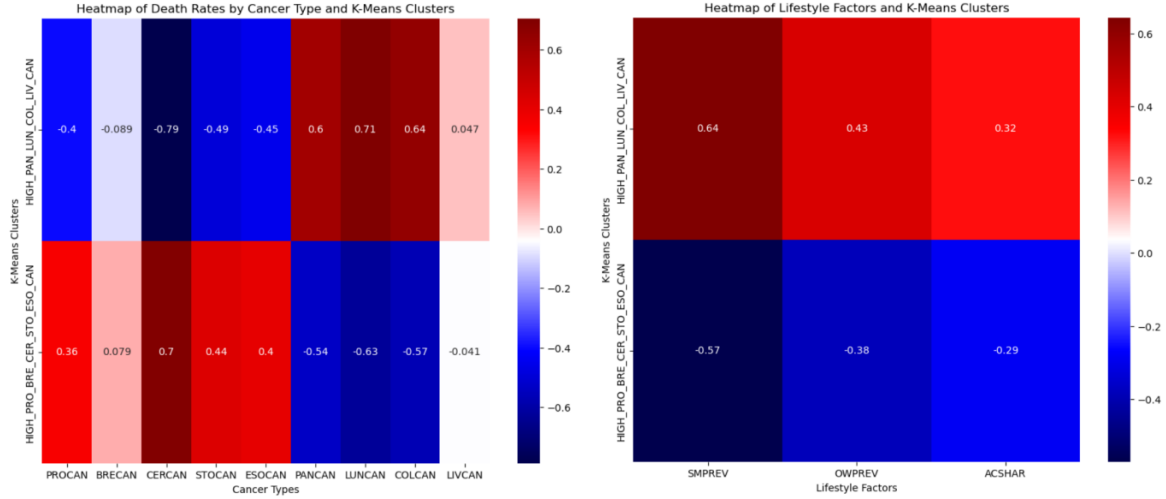



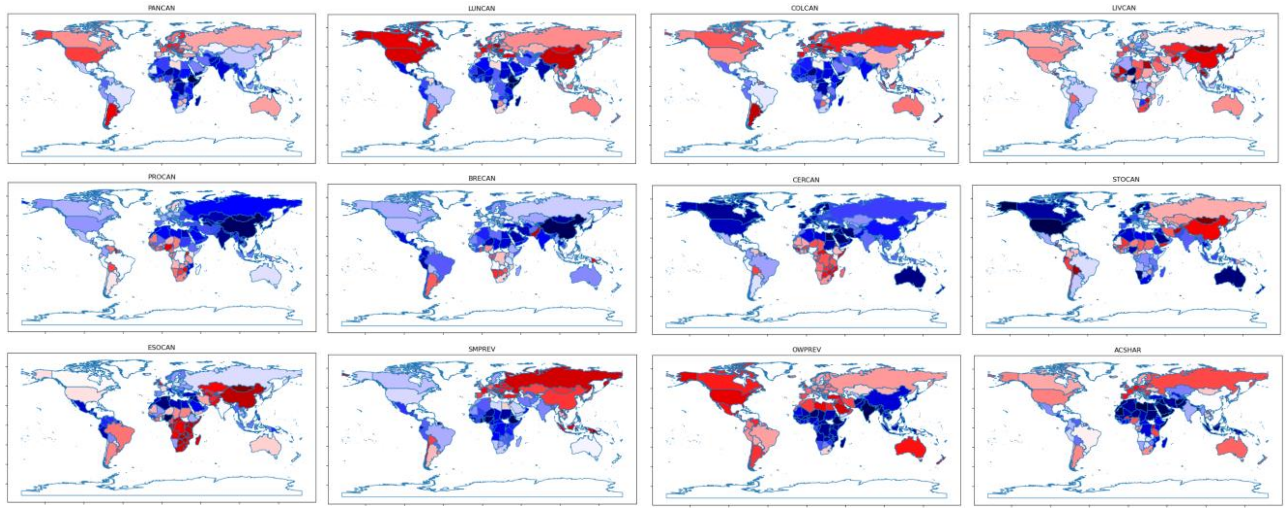
# PLOTS – MODEL SELECTION | POST-HOC

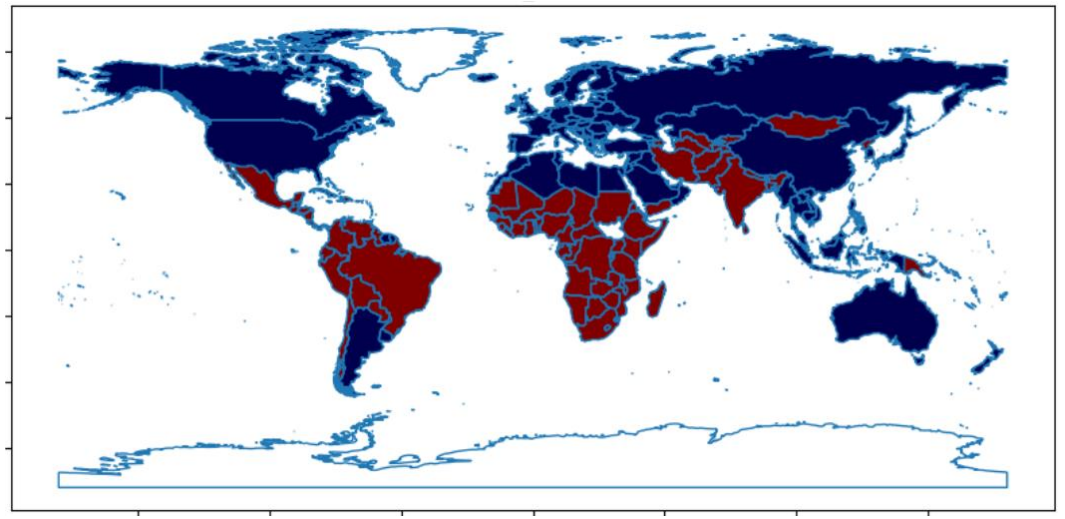
- Pattern Identification from Formulated Clusters




Final Model



Section 4

# Summary

# OVERALL FINDINGS AND IMPLICATIONS

- Key Findings
  - Using the silhouette score, the best-performing clustering model is **K-Means with 2 clusters**.
  - Given the final model, the segmented groups were observed with characteristics as follows:
    - **Cluster 0 : HIGH_PAN_LUN_COL_LIV_CAN** composed of countries characterized by:
      - Higher death rates for pancreatic, lung, colon and liver cancers
      - Lower death rates for prostate, breast, cervical, stomach and esophageal cancers
      - Higher smoking prevalence, overweight prevalence and alcohol consumption
      - Predominantly from North America, Europe, Non-South Asia and Australia regions
    - **Cluster 1: HIGH_PRO_BRE_CER_STO_ESO_CAN** composed of countries characterized by:
      - Higher death rates for prostate, breast, cervical, stomach and esophageal cancers
      - Lower death rates for pancreatic, lung, colon and liver cancers
      - Lower smoking prevalence, overweight prevalence and alcohol consumption
      - Predominantly from South America, South Asia and Africa regions
- Overall Implications
  - Clustering analysis showed disparities in cancer death rates among countries with associations to lifestyle factors and geographical locations. These findings are essential for developing targeted interventions to reduce health inequities and improve cancer mortality overall.

# CONCLUSION

- Overall Summary
  - Data collection for the analysis involved cancer death rates, lifestyle factors and geolocation data by country which were hypothesized to be sufficiently correlated to enable the discovery of natural groupings based on the similarity or dissimilarity of the observations..
  - The quality of gathered data was assessed and potential issues were identified.
  - Appropriate pre-processing methods including remedial procedures to address duplicate, missing, outlying, and non-normalized data were applied to prepare the data for subsequent analysis. Additional data scaling and transformation were implemented.
  - EDA using visualization and statistical testing presented the various significant associations among cancer death rates, lifestyle factors and geolocation data.
  - A final clustering model was selected among candidates which allowed the segmentation of countries into distinct groups and provided a more granular view of relationships across different cancer death rates, lifestyle factors and geolocation data.
  - Overall analysis findings were discussed and their practical implications were highlighted.

Section 5

# Appendix

# APPENDIX

- Source Data
  - Cancer Deaths by Type: OurWorldInData.Org
  - Prevalence of Overweight Among Adults: OurWorldInData.Org
  - Prevalence of Daily Smoking Among Adults: OurWorldInData.Org
  - Total Alcohol Consumption: OurWorldInData.Org
  - Geographic Coordinates: Geodatos
  - Global Map Shape File: GeoJson-Maps

# APPENDIX

- Python Notebooks | Codes
  - GitHub URL: Data Background
  - GitHub URL: Data Description
  - GitHub URL: Data Quality Assessment
  - GitHub URL: Data Preprocessing
    - GitHub URL: Data Cleaning
    - GitHub URL: Outlier Treatment
    - GitHub URL: Collinearity
    - GitHub URL: Shape Transformation
    - GitHub URL: Centering and Scaling
    - GitHub URL: Preprocessed Data Description
  - GitHub URL: Data Exploration
    - GitHub URL: Exploratory Data Analysis
    - GitHub URL: Hypothesis Testing

# APPENDIX

- Python Notebooks | Codes
  - GitHub URL: Model Development
    - GitHub URL: Premodelling Data Description
    - GitHub URL: K-Means Clustering
    - GitHub URL: Bisecting K-Means Clustering
    - GitHub URL: Gaussian Mixture Clustering
    - GitHub URL: Agglomerative Clustering
    - GitHub URL: Ward Hierarchical Clustering
  - GitHub URL: Consolidated Findings

# APPENDIX

- References
  - [Book] Data Preparation for Machine Learning: Data Cleaning, Feature Selection, and Data Transforms in Python by Jason Brownlee
  - [Book] Feature Engineering and Selection: A Practical Approach for Predictive Models by Max Kuhn and Kjell Johnson
  - [Book] Feature Engineering for Machine Learning by Alice Zheng and Amanda Casari
  - [Book] Applied Predictive Modeling by Max Kuhn and Kjell Johnson
  - [Book] Data Mining: Practical Machine Learning Tools and Techniques by Ian Witten, Eibe Frank, Mark Hall and Christopher Pal
  - [Book] Data Cleaning by Ihab Ilyas and Xu Chu
  - [Book] Data Wrangling with Python by Jacqueline Kazil and Katharine Jarmul
  - [Book] RegressionFinding Groups in Data: An Introduction to Cluster Analysis by Leonard Kaufman and Peter Rousseeuw
  - [Book] The Elements of Statistical Learning by Trevor Hastie, Robert Tibshirani and Jerome Friedman
  - [Book] Training Systems using Python Statistical Modeling by Curtis Miller
  - [Book] Python Data Science Handbook by Jake VanderPlas
  - [Book] Theory of Agglomerative Hierarchical Clustering by Sadaaki Miyamoto
  - [Python Library API] NumPy by NumPy Team
  - [Python Library API] pandas by Pandas Team
  - [Python Library API] seaborn by Seaborn Team
  - [Python Library API] matplotlib.pyplot by MatPlotLib Team
  - [Python Library API] itertools by Python Team
  - [Python Library API] operator by Python Team
  - [Python Library API] sklearn.preprocessing by Scikit-Learn Team
  - [Python Library API] sklearn.metrics by Scikit-Learn Team
  - [Python Library API] sklearn.cluster by Scikit-Learn Team
  - [Python Library API] sklearn.mixture by Scikit-Learn Team
  - [Python Library API] SciPy by scipy Team
  - [Python Library API] GeoPandas by GeoPandas Team

# Thank You!