MSc Data Science & Information Technologies

Course: Clustering Algorithms

Professor: Konstantinos Koutroumbas

Clustering Algorithms 2nd Homework

Name	*Student ID*
Konstantinos Giatras	7115152300005
Vasiliki Koumarela	7115152200009

DEPARTMENT OF INFORMATICS & TELECOMMUNICATIONS

HELLENIC REPUBLIC
National and Kapodistrian University of Athens
EST. 1837

2023-2024

Contents

1 Problem Statement

The objective of this assignment is to cluster countries using socio-economic and health factors that determine the overall development of a country and to characterize each resulting cluster (and, consequently, the countries it comprises) based on the relevant values of the available factors for each country, which are the following:

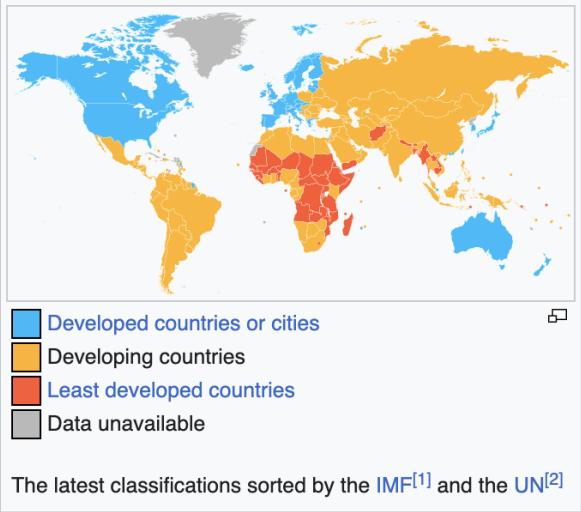
1. **Child_mortality**: Death of children under 5 years of age per 1000 live births
2. **Exports**: Exports of goods and services per capita. Given as %age of the GDP per capita
3. **Health**: Total health spending per capita. Given as %age of GDP per capita
4. **Imports**: Imports of goods and services per capita. Given as %age of the GDP per capita
5. **Income**: Net income per person
6. **Inflation**: The measurement of the annual growth rate of the Total GDP
7. **Life_expectancy**: The average number of years a new born child would live if the current mortality patterns are to remain the same
8. **Total_fertility**: The number of children that would be born to each woman if the current age-fertility rates remain the same
9. **GDPP**: The GDP per capita (Calculated as the Total GDP divided by the total population)

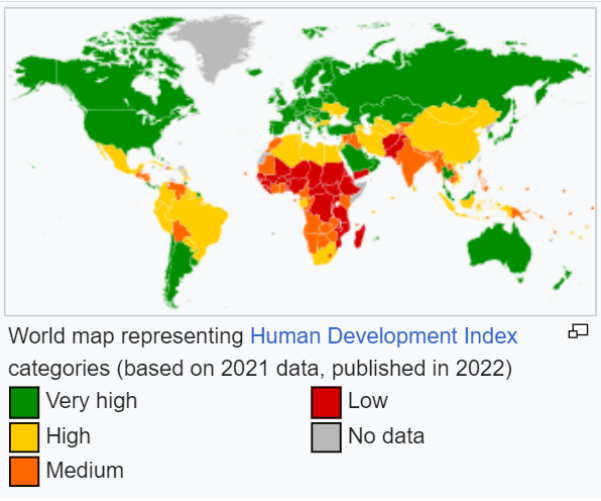
The relevant data set consists of data from ***167 countries*** and for each country, the above ***9 features*** are available. The data are given in a 167×9 matrix, called "***Countrydata***", where each row corresponds to a country and each column to a feature. In addition, the array 167×1 column vector "***country***" contains the names of countries corresponding to the lines of the matrix "***Countrydata***". Both "***Countrydata***" and "***country***" are included in the file named "***data_country***".

The data are from https://www.kaggle.com/rohan0301/unsupervised-learning-on-country-data.

1.1 Domain knowledge

Clustering countries based on a combination of economic and health characteristics can lead to the identification of distinct groups representing countries according to their development. Utilizing indicators such as economic indices, and healthcare metrics, the countries can be categorized into three groups: "Developed countries", "Developing countries", and "Least developed countries". A more up-to-date classification can be achieved through utilizing the Human Development Index (HDI), where countries can have a very high, high, medium or low score. [4] Additionally, the World Bank's classification of economies into four income groups (high, upper-middle, lower-middle, and low income) [9] provides valuable insights. suggesting that we can anticipate forming **clusters ranging from three to four** based on this information.





1. **Child_mortality**: According to the World Health Organization [10], child mortality varies depending on where children are born, especially the two regions with the highest child mortality rates are sub-Saharan Africa and south and central Asia (2021 data).
2. **Exports**: Exports are affected from geographical location, transportation and workforce, so we can provide a general interpretation. For example, some regions in Africa may have limited exports compared to more industrialized nations, while China is a major global exporter, known for its manufacturing capabilities and trade dominance in various industries. [2]
3. **Health**: Researchers from Humboldt University of Berlin started use the term of "Global South" [8] to describe a grouping of countries along the lines of socio-economic and political characteristics such as lower incomes, deficient health system, etc. Low-and-Middle-Income countries typically expose large inequalities in affordability and accessibility of medical care [3]. It is worth to refer that people in developing countries usually have a **lower life expectancy** than people in developed countries, reflecting both lower income levels and poorer public health.
4. **Imports**: **Imports and exports are interlinked aspects** of international trade, embodying the movement of goods and services across borders. The correlation between them is encapsulated by the trade balance and **influences a country's Gross Domestic Product (GDP)**. [1]
5. **Income**: The connection between income per person and healthcare lies in the relationship between economic prosperity and access to quality medical services. Individuals with higher incomes are more likely to reside in developed countries, where healthcare infrastructure and services are generally more advanced and accessible, creating a **positive correlation between income levels and healthcare quality**. [7]
6. **Inflation**: Inflation is intricately connected to other economic features and health indicators through its impact on purchasing power and overall economic stability. High inflation rates can lead to increased costs of living, affecting both **economic and health-related factors**. The GDP deflator is computed using the following formula [5]:

$$\text{Inflation Rate} = \left(\frac{\text{GDP Deflator in Year 2} - \text{GDP Deflator in Year 1}}{\text{GDP Deflator in Year 1}} \right) * 100$$

7. **Life_expectancy**: Life expectancy indicator is directly connected to the quality of healthcare in the respective countries and are influenced by various factors, including the prevalence of diseases. The Organization for Economic Cooperation and Development (OECD) library states that, apart from the high correlation between these variables and GDP per capita (GDPP), there are notable differences in life expectancy among countries with similar income per capita. For example, Japan and Spain exhibit higher life expectancies, while Luxembourg, the United States, and the Russian Federation show lower life expectancies than would be predicted by their GDP per capita alone. [6]
8. **Total_fertility**: Predicting the relationship between total fertility rate and other features is complex, as it doesn't consistently align with the expectation that good healthcare and a strong economy would lead to higher fertility rates. Cultural, social, and individual factors can play significant roles in shaping fertility patterns, making it challenging to establish a straightforward correlation.
9. **GDPP**: GDP per capita serves as a crucial indicator as it encapsulates various economic factors and provides a comprehensive representation of a country's economic health. Its inclusion in the analysis allows for a concise representation of economic indicators within this feature group. The GDPP is calculated based on the following formula [1]:

$$GDPP = \left(\frac{\text{Consumption} + \text{Investment} + \text{Government Spending} + (\text{Exports - Imports}))}{\text{Population}} \right)$$

2 "Feeling the Data"

In this stage, our goal is to perform various simple operations on the data, in order to by aware of their nature and extract important information for each feature.

More specifically, we will consider the features in groups, check for the existence of missing values and then, individually for each feature:

- Determine its kind (in terms of whether it is discrete or continuous-valued)
- Find its range of values (min, max)
- Approximate and visualize the probability density function (mean, median, standard deviation, histogram)
- Detect outliers
- Perform standard score normalization
- Perform minmax feature scaling normalization
- Discover possible linear dependence with other features (Pearson correlation coefficient heatmaps)

2.1 Data Exploration

All features in our dataset are continuous positive valued (except for inflation, which can also be negative). The features can be categorized into two main groups: Socio-economic Indicators and Health and Well-being Indicators. This grouping, based on the nature of the features, will assist us in selecting representative features from each group.

1. Socio-economic Indicators:
   - Exports
   - Imports
   - Income
   - Inflation
   - GDPP

   These variables represent aspects of a country's socio-economic performance, including income levels, trade activities (exports and imports), and the overall economic output per capita (GDPP).
2. Health and Well-being Indicators:
   - Child mortality
   - Health
   - Life expectancy
   - Total fertility

   These variables focus on health and demographic indicators, including child mortality rates, overall health indicators, life expectancy, and fertility rates.

2.2 Missing values

Before continuing on the data exploration, it is vital to check for missing values within the dataset. The presence of missing data can significantly impact the reliability and validity of analytical results. With the following code, we ensured that there were **no missing data** (but if there were, one way of handling that could be to remove the vectors with missing values):

```
% Missing values per column
missing_values_cols = any(isnan(X), 1); % where X is the dataset
disp('Missing Values Along Columns:');
disp(missing_values_cols);
```

2.3 Descriptive Statistics and Plots of Features

We use the following code to find the value range (minimum and maximum values), mean value, median value, standard deviation and percentiles for each feature:

```
% Descriptive statistics for X
[X_min, X_max, X_mean, X_median, X_var, X_std, X_percentiles] = data_statistics(X);

function [min_X, max_X, mean_X, median_X, var_X, std_X, percentiles_X] = data_statistics
    (X)
    min_X = min(X);
    max_X = max(X);
    mean_X = mean(X);
    median_X = median(X);
    var_X = var(X);
    std_X = std(X);
    percentiles = [25, 50, 75];
    percentiles_X = prctile(X, percentiles);
end
```

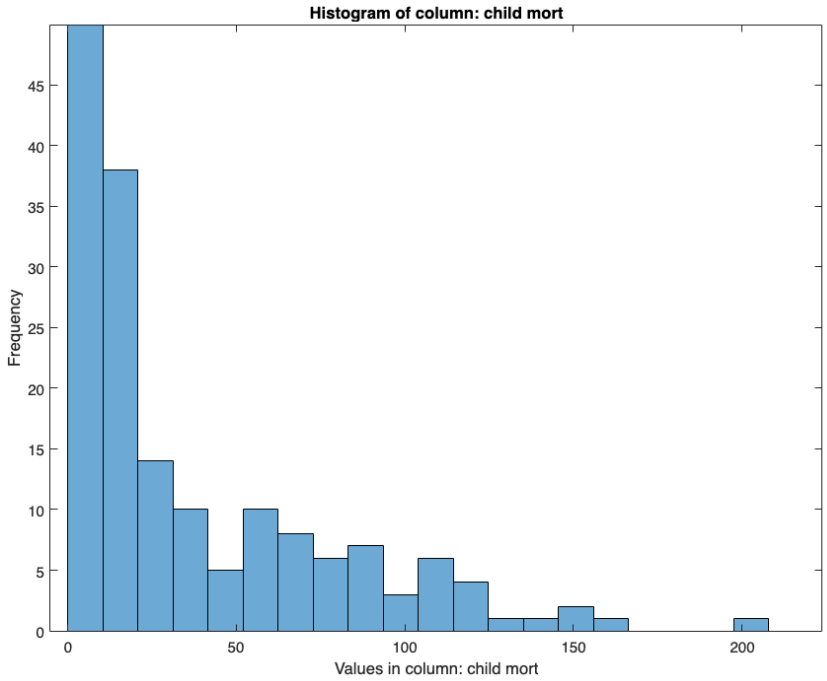
	child mort	exports	health	imports	income	inflation	life expec	total fer	gdpp
min	2.6	0.109	1.81	0.0659	609	-4.21	32.1	1.15	231
max	208	200	17.9	174	125000	104	82.8	7.49	105000
mean	38.2701	41.109	6.8157	46.8902	17145	7.7818	70.5557	2.948	12964
median	19.3	35	6.32	43.3	9960	5.39	73.1	2.41	4660
variance	1626.42	751.42	7.5451	586.10	371643894.16	111.74	79.09	2.29	335941419.96
std	40.3289	27.412	2.7468	24.2096	19278	10.5707	8.8932	1.5138	18329
25%	8.075	23.8	4.915	30.1	3347.5	1.79	65.3	1.7925	1320
50%	19.3	35	6.32	43.3	9960	5.39	73.1	2.41	4660
75%	62.15	51.375	8.625	58.825	22850	10.825	76.8	3.895	14325

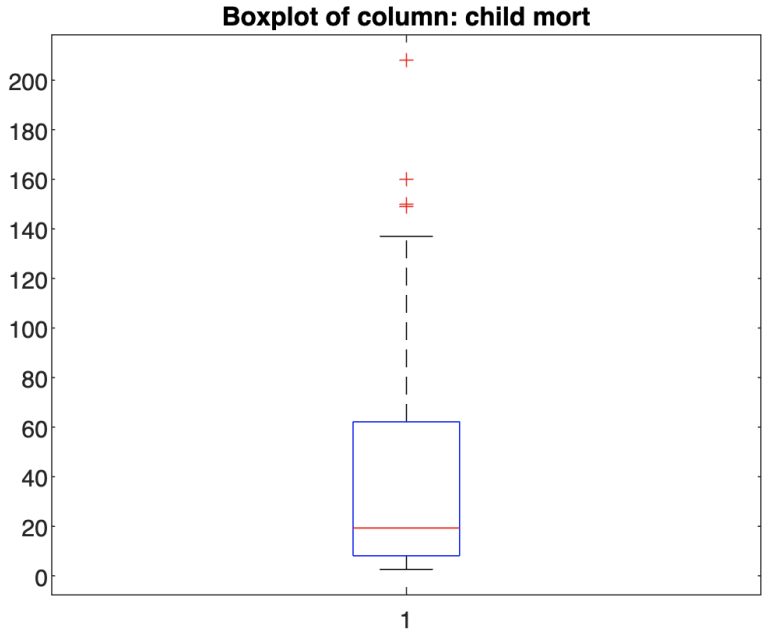
Table 1: Statistics on the original data

Then, to visualize the probability density function and detect possible outliers for each feature, we use the following code to create histograms and boxplots respectively:

```
% Display histograms
labels = {'child mort', 'exports', 'health', 'imports', 'income', 'inflation', 'life
    expec', 'total fer', 'gdpp'};
cols = size(X, 2);
for i = 1:cols
    current_col = X(:, i);
    if DISPLAY_FIGURES
        figure;
        histogram(current_col, 20);
        title(['Histogram of column: ', labels{i}]);
        xlabel(['Values in column: ', labels{i}]);
        ylabel('Frequency');
    end
end

% ------------------------------------------------------------------

% Display boxplots
for i = 1:cols
    current_col = X(:, i);
    if DISPLAY_FIGURES
        figure;
        boxplot(current_col);
        title(['Boxplot of column: ', labels{i}]);
    end
end
```

1. **Child_mortality**:

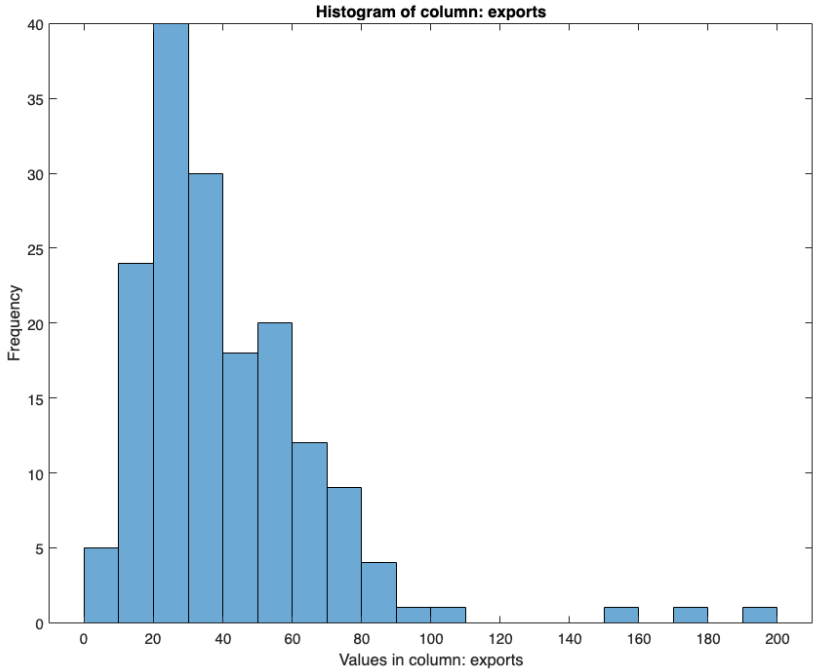


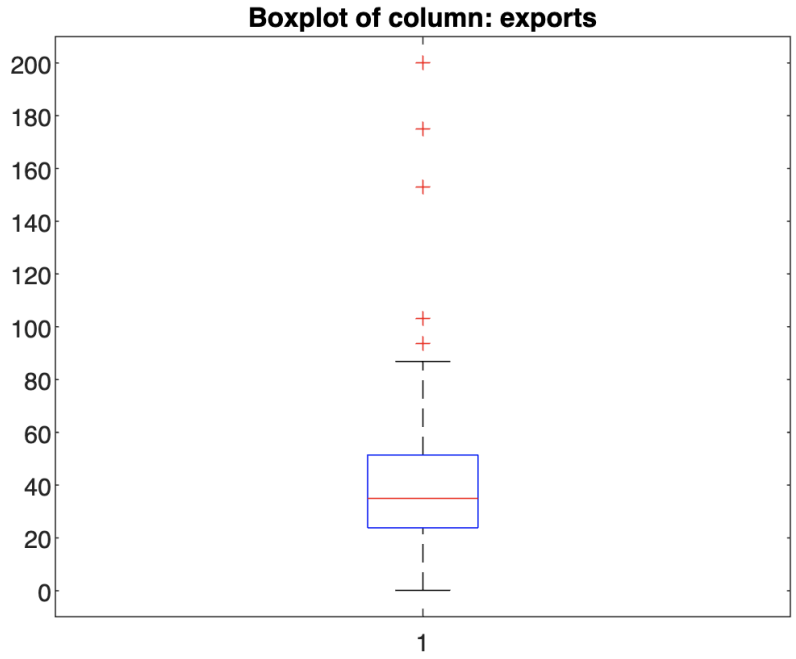


The minimum child mortality rate is 2.6 while the maximum value is 208, meaning that the relatively low and high child mortality rates result in significant variation. Also, the standard deviation is high (40.33), indicating considerable dispersion in child mortality rates among countries. The 75th percentile (75%) is significantly higher than the median (50%), resulting in the positively skewed (left skewed) distribution. The boxplot reveals some possible outliers for values greater than 140.

	Min	**Median**	**Max**
Country	Iceland	Iran	Haiti
Value	2.6	19.3	208

2. **Exports**:

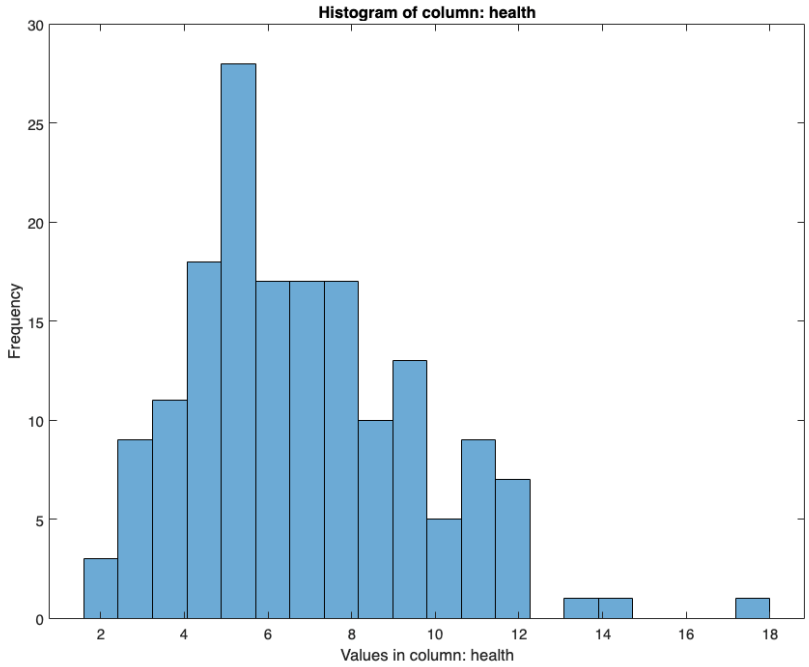


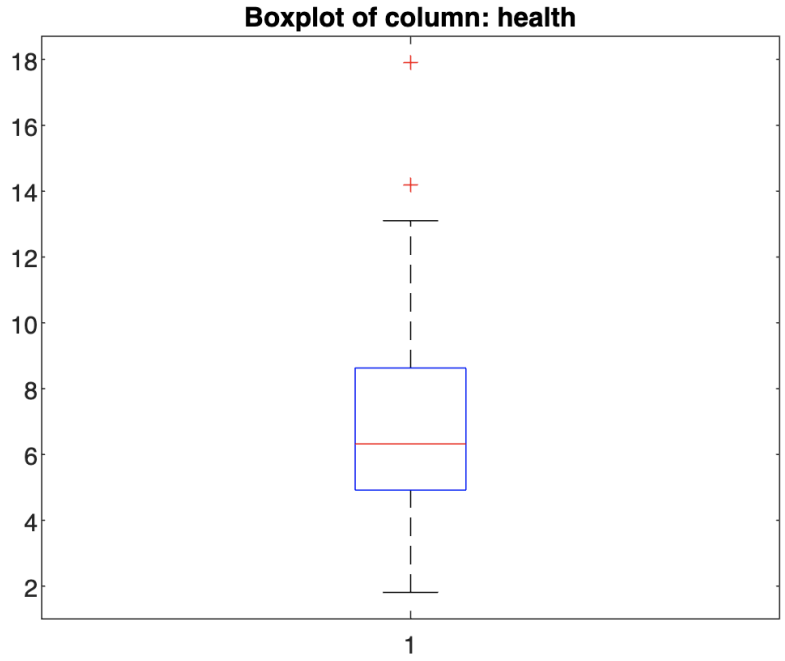


The value range for exports is [0.109, 200], indicating a wide range of export values across countries. The mean (41.109) and median (35) suggest a moderately skewed distribution, with a standard deviation of 27.412. The 75th percentile is 51.375, higher than the median, suggesting a positively skewed distribution. The boxplot shows potential outliers for values greater than 90.

	Min	**Median**	**Max**
Country	Myanmar	Bahamas	Singapore
Value	0.109	35	200

3. **Health**:

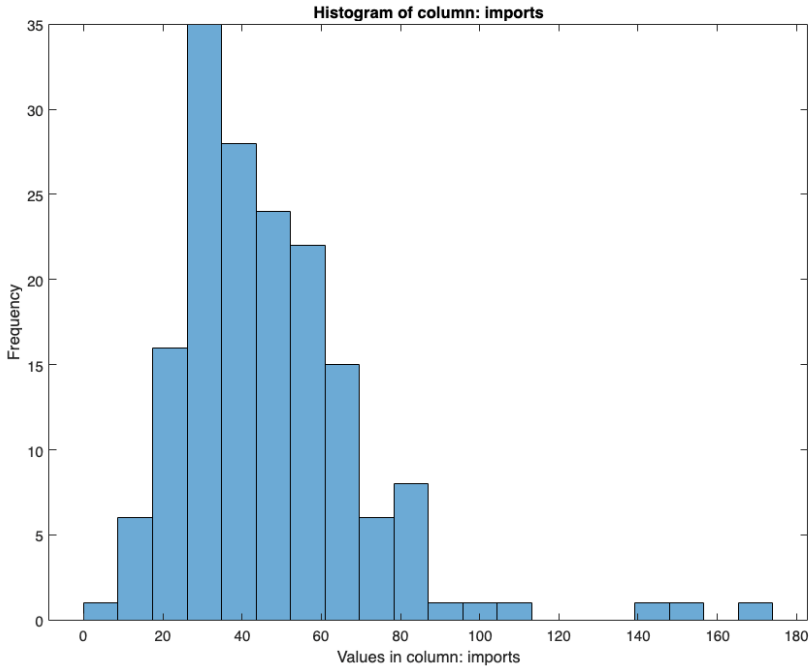


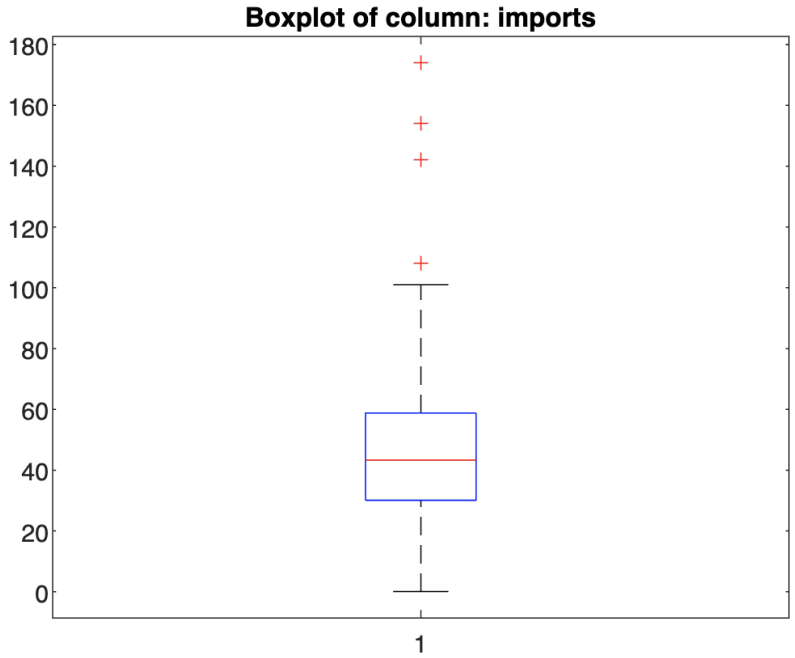


The health indicator ranges from 1.81 to 17.9, showcasing diversity among countries. The mean (6.8157) and median (6.32) are close, indicating a relatively symmetric distribution. The standard deviation is 2.7468, suggesting moderate dispersion. The 75th percentile is higher than the median, indicating a positively skewed distribution.

	Min	**Median**	**Max**
Country	Qatar	Sudan	United States
Value	1.81	6.32	17.9

4. **Imports**:

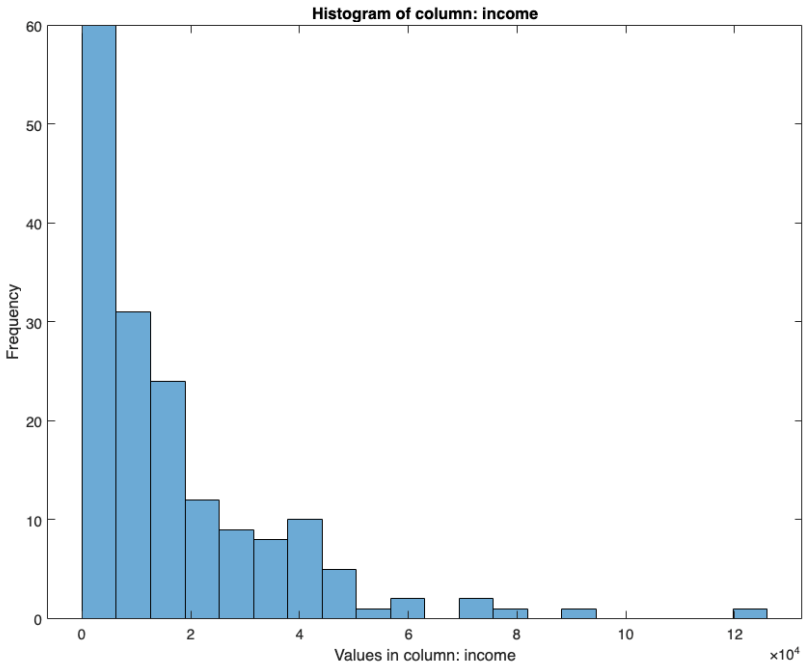


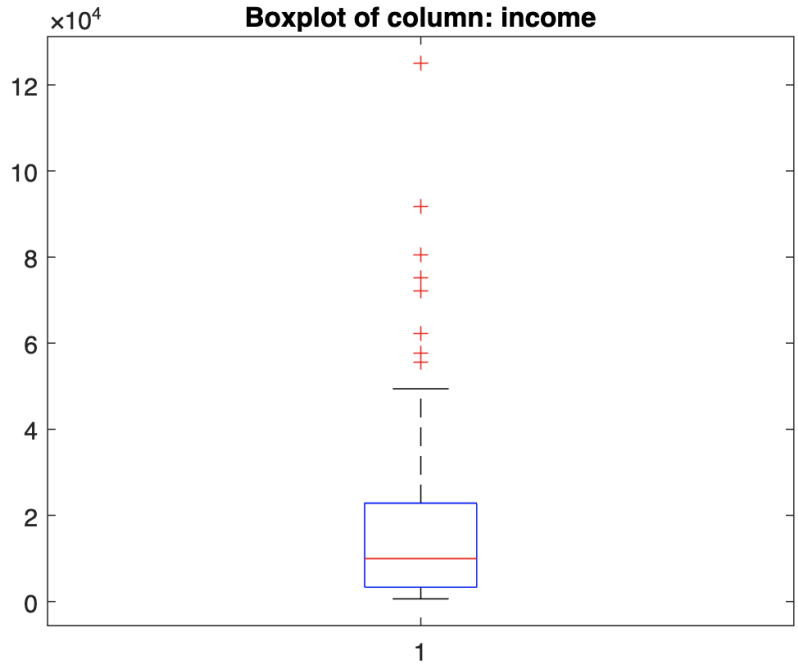


The import values range from 0.0659 to 174, indicating considerable variability. The mean (46.8902) is higher than the median (43.3), suggesting a positively skewed distribution. The standard deviation is 24.2096, reflecting moderate dispersion. The boxplot reveals potential outliers for values greater than 110.

	Min	**Median**	**Max**
Country	Myanmar	Cote d'Ivoire	Singapore
Value	0.0659	43.3	174

5. **Income**:

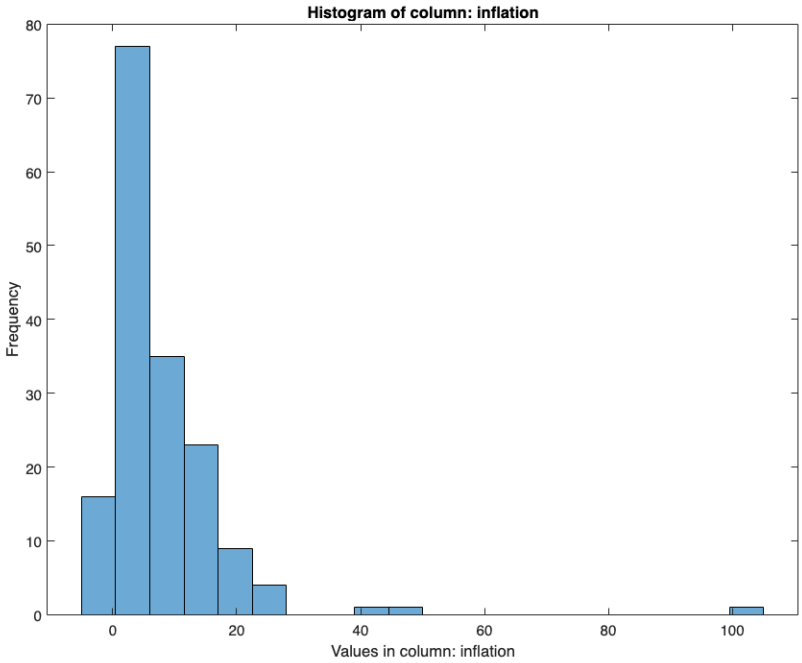


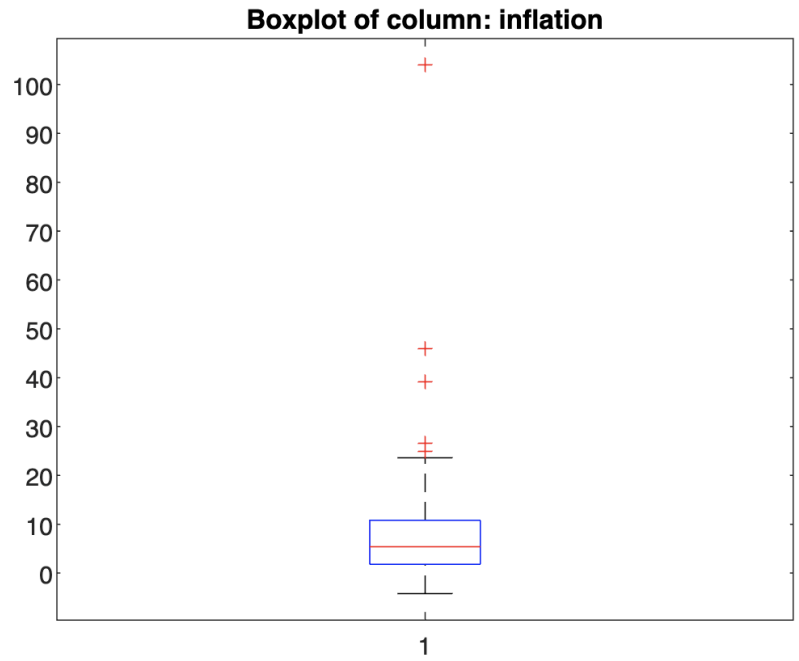


Income levels range from 609 to 125000, illustrating significant diversity. The mean (17145) and median (9960) suggest a positively skewed distribution. The standard deviation is 19278, indicating substantial dispersion. The boxplot shows potential outliers for values greater than 50000.

	Min	**Median**	**Max**
Country	Congo Dem Rep	Peru	Qatar
Value	609	9960	125000

6. **Inflation**:

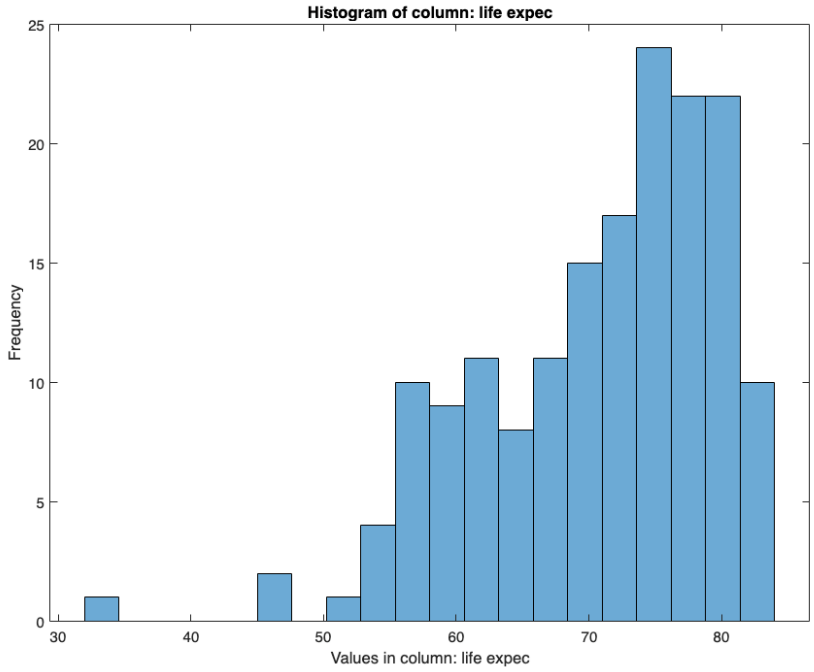


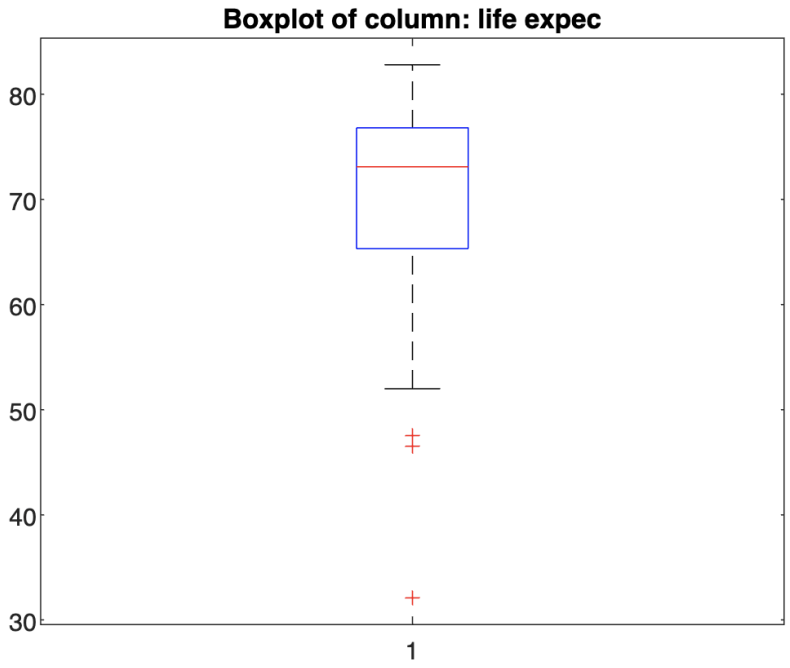


Inflation rates range from -4.21 to 104, showcasing diverse economic conditions. The mean (7.7818) and median (5.39) suggest a positively skewed distribution. The standard deviation is 10.5707, indicating considerable variability. The boxplot shows potential outliers for values greater than 25.

	Min	**Median**	**Max**
Country	Seychelles	Cote d'Ivoire	Nigeria
Value	-4.21	5.39	104

7. **Life_expectancy**:

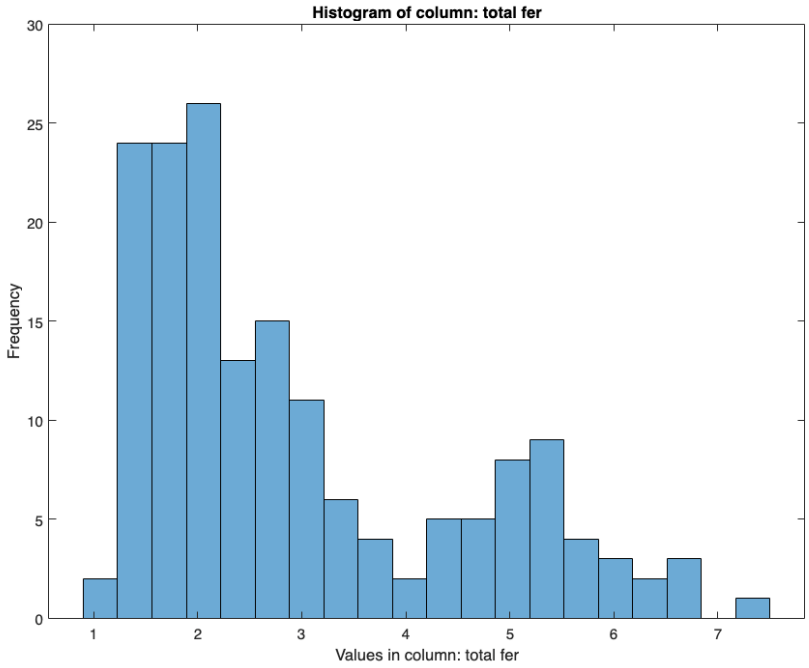


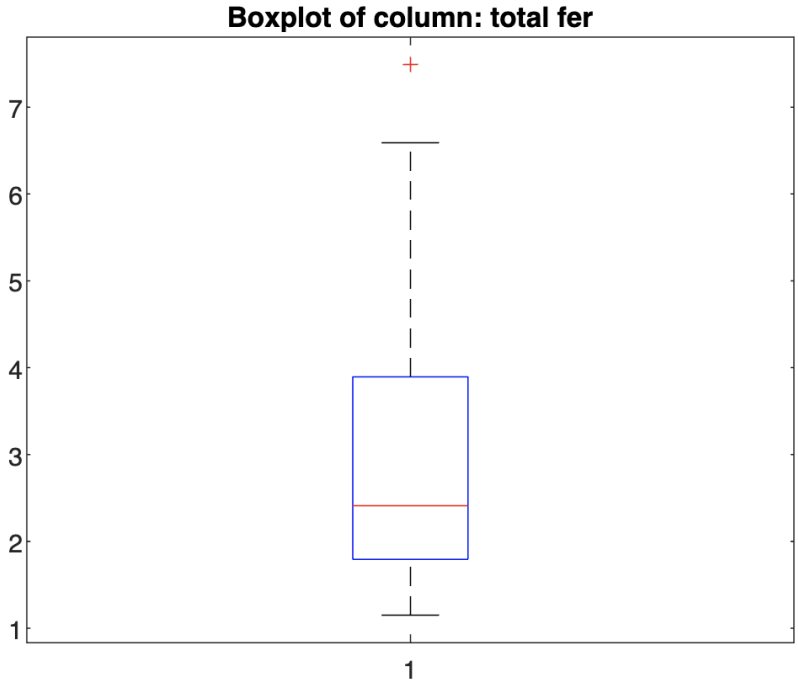


Life expectancy varies from 32.1 to 82.8, indicating diverse healthcare conditions. The mean (70.5557) and median (73.1) suggest a slightly negatively skewed distribution. The standard deviation is 8.8932, reflecting moderate variability. The boxplot shows potential outliers for values less than 50.

	Min	**Median**	**Max**
Country	Haiti	Latvia	Japan
Value	32.1	73.1	82.8

8. **Total_fertility**:

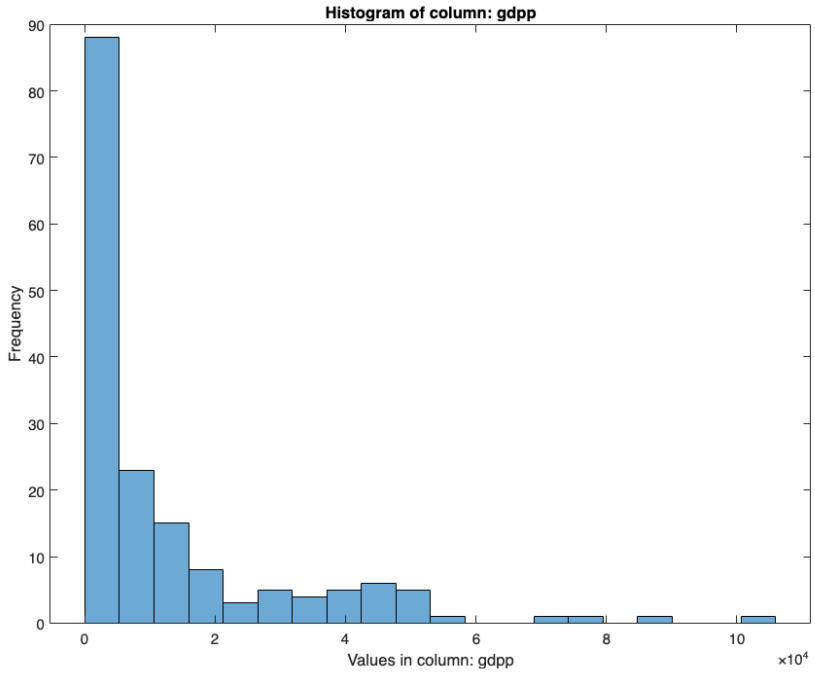


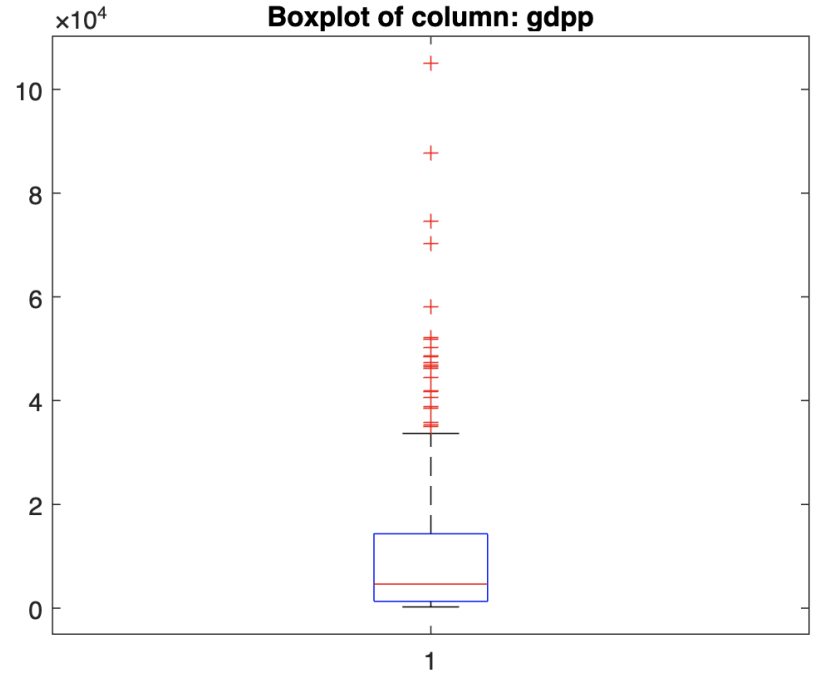


Total fertility rates range from 1.15 to 7.49, indicating diverse demographics. The mean (2.948) and median (2.41) suggest a slightly positively skewed distribution. The standard deviation is 1.5138, reflecting moderate variability. The boxplot shows potential outliers for values greater than 4.5.

	Min	**Median**	**Max**
Country	Singapore	Libya	Niger
Value	1.15	2.41	7.49

9. **GDPP**:





GDP per capita varies from 231 to 105000, indicating significant economic disparities. The mean (12964) and median (4660) suggest a positively skewed distribution. The standard deviation is 18329, indicating substantial variability. The boxplot shows potential outliers for values greater than 30000.

	Min	**Median**	**Max**
Country	Burundi	Ecuador	Luxembourg
Value	231	4660	105000

To find the countries that correspond to the minimum, median and maximum values of each feature, we used the following code:

```
% Finding corresponding country names
[country_min, country_median, country_max] = find_country_names(country, X, X_min,
    X_median, X_max);

function [country_min, country_median, country_max] = find_country_names(country, X,
    min_X, median_X, max_X)
    for i = 1:size(X,2) % Loop through each feature
        [~, min_index] = min(X(:,i)); % Find index of min value for feature i
        [~, median_index] = min(abs(X(:,i)-median_X(i))); % Find index closest to median
     for feature i
        [~, max_index] = max(X(:,i)); % Find index of max value for feature i

        country_min(i) = country(min_index); % Find country for min value
        country_median(i) = country(median_index); % Find country for median value
        country_max(i) = country(max_index); % Find country for max value
    end
end
```

2.3.1 Outlier detection

Outliers can be found in the above boxplots, which draw points as outliers if they are greater than $q_3 + w(q_3 - q_1)$ or less than $q_1 - w(q_3 - q_1)$. The w is the multiplier Whisker (corresponds to approximately $+/-2.7\sigma$ and 99.3 percent coverage if the data are normally distributed), and q_1 and q_3 are the 25th and 75th percentiles of the sample data, respectively.

In the provided boxplots, outliers for most features tend to be situated at the upper limit of the values, as evidenced by the red crosses in the upper range. Notably, the "life_expec" feature displays outliers at the lower limit, indicating values that fall below the general distribution. All features contain outliers, but GDPP contains the most. Despite the presence of outliers in the features, it is recommended not to remove them indiscriminately, as they contain valuable information necessary to accurately group countries based on their characteristics.

2.4 Standard Score Normalization

The standard score normalization or the z-score normalizaton is the process of normalizing every value in a dataset such that the mean of all of the values is 0 and the standard deviation is 1:

$$z = \frac{x - \mu}{\sigma},$$

where μ is the mean of the population, σ is the standard deviation of the population.

```
% Normalizing each row as a zero mean unit variance distributions
X_stand_norm = (X-ones(N,1)*mean(X)) ./ (ones(N,1)*std(X));
% Descriptive statistics for X_stand_norm
[X_stand_norm_min, X_stand_norm_max, X_stand_norm_mean, X_stand_norm_median,
    X_stand_norm_var, X_stand_norm_std, X_stand_norm_percentiles] = data_statistics(
    X_stand_norm);
```

	child mort	exports	health	imports	income	inflation	life expec	total fer	gdpp
min	-0.8845	-1.4957	-1.8223	-1.9341	-0.8577	-1.1344	-4.3242	-1.1877	-0.6947
max	4.2086	5.7964	4.0353	5.2504	5.5947	9.1023	1.3768	3.0003	5.0214
mean	2.1274e-17	-4.2547e-17	-1.3296e-16	2.6592e-17	-8.5095e-17	1.0903e-16	-1.2445e-15	2.9783e-16	3.1911e-17
median	-0.4704	-0.2229	-0.1805	-0.1483	-0.3727	-0.2263	0.2861	-0.3554	-0.4531
variance	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
std	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
25%	-0.7487	-0.6314	-0.6920	-0.6935	-0.7157	-0.5668	-0.5910	-0.7633	-0.6353
50%	-0.4704	-0.2229	-0.1805	-0.1483	-0.3727	-0.2263	0.2861	-0.3554	-0.4531
75%	0.5921	0.3745	0.6587	0.4930	0.2959	0.2879	0.7021	0.6256	0.0742

Table 2: Statistics on the transformed data after z-score normalization

2.5 Minmax Feature Scaling Normalization

Min-max rescaling is the simplest method and consists in rescaling the range of features to scale the range in [0, 1]:

$$x' = \frac{x - min(x)}{max(x) - min(x)}$$

```
% Normalizing each row in the min-max range
X_minmax_norm = (X-ones(N,1)*min(X)) ./ (ones(N,1)*(max(X)-min(X)));
% Descriptive statistics for X_minmax_norm
[X_minmax_norm_min, X_minmax_norm_max, X_minmax_norm_mean, X_minmax_norm_median,
    X_minmax_norm_var, X_minmax_norm_std, X_minmax_norm_percentiles] = data_statistics(
    X_minmax_norm);
```

	child mort	exports	health	imports	income	inflation	life expec	total fer	gdpp
min	0	0	0	0	0	0	0	0	0
max	1	1	1	1	1	1	1	1	1
mean	0.1737	0.2051	0.3111	0.2692	0.1329	0.1108	0.7585	0.2836	0.1215
median	0.0813	0.1746	0.2803	0.2486	0.0752	0.0887	0.8087	0.1987	0.0423
variance	0.0386	0.0188	0.0291	0.0194	0.0240	0.0095	0.0308	0.0570	0.0306
std	0.1963	0.1371	0.1707	0.1392	0.1550	0.0977	0.1754	0.2388	0.1749
25%	0.0267	0.1185	0.1930	0.1727	0.0220	0.0554	0.6548	0.1013	0.0104
50%	0.0813	0.1746	0.2803	0.2486	0.0752	0.0887	0.8087	0.1987	0.0423
75%	0.2899	0.2565	0.4236	0.3378	0.1788	0.1389	0.8817	0.4330	0.1345

Table 3: Statistics on the transformed data after min-max ([0,1]) normalization

2.6 Correlation Coefficient Heatmaps

To discover possible linear dependence between features, we use the following code:

```
% Correlation coefficients heatmap
correlation_matrix = corrcoef(X);
if DISPLAY_FIGURES
    colormap('jet');
    figure;
    heatmap(correlation_matrix, 'XDisplayLabels', labels, 'YDisplayLabels', labels);
    title('Correlation Heatmap');
end

% ---------------------------------------------------------------------

% Correlation coefficients heatmap after normalization using unit variance distributions
corr_stand_norm = corrcoef(X_stand_norm);
if DISPLAY_FIGURES
    colormap('jet');
    figure;
    heatmap(corr_stand_norm, 'XDisplayLabels', labels, 'YDisplayLabels', labels);
    title('Correlation Heatmap after normalization using unit variance distributions');
end

% ---------------------------------------------------------------------

% Correlation coefficients heatmap after normalization using min-max range
corr_minmax_norm = corrcoef(X_minmax_norm);
if DISPLAY_FIGURES
    colormap('jet');
    figure;
    heatmap(corr_minmax_norm, 'XDisplayLabels', labels, 'YDisplayLabels', labels);
    title('Correlation Heatmap after normalization using min-max range');
end
```


Correlation Heatmap
child mort 1 -0.3181 -0.2004 -0.1272 -0.5243 0.2883 -0.8867 0.8485 -0.483
exports -0.3181 1 -0.1144 0.7374 0.5168 -0.1073 0.3163 -0.32 0.4187
health -0.2004 -0.1144 1 0.09572 0.1296 -0.2554 0.2107 -0.1967 0.346
imports -0.1272 0.7374 0.09572 1 0.1224 -0.247 0.05439 -0.159 0.1155
income -0.5243 0.5168 0.1296 0.1224 1 -0.1478 0.612 -0.5018 0.8956
inflation 0.2883 -0.1073 -0.2554 -0.247 -0.1478 1 -0.2397 0.3169 -0.2216
life expec -0.8867 0.3163 0.2107 0.05439 0.612 -0.2397 1 -0.7609 0.6001
total fer 0.8485 -0.32 -0.1967 -0.159 -0.5018 0.3169 -0.7609 1 -0.4549
gdpp -0.483 0.4187 0.346 0.1155 0.8956 -0.2216 0.6001 -0.4549 1
child mort exports health imports income inflation life expec total fer gdpp
1 0.8 0.6 0.4 0.2 0 -0.2 -0.4 -0.6 -0.8


Correlation Heatmap after normalization using unit variance distributions
child mort 1 -0.3181 -0.2004 -0.1272 -0.5243 0.2883 -0.8867 0.8485 -0.483
exports -0.3181 1 -0.1144 0.7374 0.5168 -0.1073 0.3163 -0.32 0.4187
health -0.2004 -0.1144 1 0.09572 0.1296 -0.2554 0.2107 -0.1967 0.346
imports -0.1272 0.7374 0.09572 1 0.1224 -0.247 0.05439 -0.159 0.1155
income -0.5243 0.5168 0.1296 0.1224 1 -0.1478 0.612 -0.5018 0.8956
inflation 0.2883 -0.1073 -0.2554 -0.247 -0.1478 1 -0.2397 0.3169 -0.2216
life expec -0.8867 0.3163 0.2107 0.05439 0.612 -0.2397 1 -0.7609 0.6001
total fer 0.8485 -0.32 -0.1967 -0.159 -0.5018 0.3169 -0.7609 1 -0.4549
gdpp -0.483 0.4187 0.346 0.1155 0.8956 -0.2216 0.6001 -0.4549 1
child mort exports health imports income inflation life expec total fer gdpp
1 0.8 0.6 0.4 0.2 0 -0.2 -0.4 -0.6 -0.8

> **!** To calculate the difference between the correlation on the original data and the correlation on the data after normalization applications, we will not assert the equality "corr original data == corr standardized data" or "corr original data == corr min-max normalized data" due to potential floating-point precision issues. Instead, we will compute the absolute value of the subtraction.

We observe that the differences between correlation values are negligible (feature scaling only changes the values of the features, not the relationships between them):

```
diff_corr_stand_data = abs(corr_stand_data - corr_original_data);
% diff_corr_stand_data_min = 0
% diff_corr_stand_data_max = 8.8818e-16

diff_corr_minmax_data = abs(corr_minmax_data - corr_original_data);
% diff_corr_minmax_data_min = 0
% diff_corr_minmax_data_max = 1.3323e-15
```

Based on the correlation heatmap, the highly positively correlated features above 0.60 are:

1. <income, gdpp> with value 0.8956
2. <child mort, total fer> with value 0.8485
3. <imports, exports> with value 0.7374
4. <life expec, income> with value 0.612
5. <life expec, gdpp> with value 0.6001

And the highly negatively correlated features under -0.60 are:

1. <life expec, child mort> with value -0.8867
2. <life expec, total fer> with value -0.7609

3 Feature Selection/Transformation

Based on the analysis of the previous stage, we have to decide:

- Which features we will employ to represent the entities involved in the current problem (in our case, the countries)
- Whether we need to apply certain transformations on the chosen features (e.g., if two features have significantly different range of values, they should be transformed, so that to have comparable ranges of values)

3.1 PCA-based Feature Selection

Before initiating the feature selection process, we will apply Principal Component Analysis (PCA) [11] to identify the most significant features. Initially, the data will be **transformed using standard score normalization** (since we want to preserve the outliers of our dataset as much as possible), followed by the application of the MATLAB PCA method:

```
X_stand_norm = (X-ones(N,1)*mean(X)) ./ (ones(N,1)*std(X));
[coeff, score, ~, ~, explained] = pca(X_stand_norm);
```

To determine the appropriate number of principal components to retain, we aim to capture at least 95% of the total eigenvalue variance:

```
cumulative = cumsum(explained);
components = find(cumulative >= 95, 1);
```

child mort	exports	health	imports	income	inflation	life expec	total fer	gdpp
45.9517	63.1334	76.1376	87.1908	94.5310	97.0152	98.2757	99.2569	100.0000

Table 4: Total eigenvalue variance values of transformed features after applying PCA

The selected features are:

(**EXPERIMENT A**): GDPP, Inflation, Total Fertility, Life Expectancy

```
X_exp_A = score(:, components);
labels_exp_A = labels([6, 7, 8, 9]);
```

3.2 Correlation-based Feature Selection

- We choose "GDPP" over "Income" because they are highly correlated (0.8956) and also income is included in the "Consumption" term of the numerator of GDPP
- "Imports" and "Exports" are included in the "GDPP" formula, so we can exclude both of them
- We keep "Inflation" because it is a complex standalone economic feature
- We choose "Life expectancy" over "Child mortality" because they are highly correlated (-0.8867) and because the former is provides a broader, more general version of the information provided by the latter
- We keep "Total fertility", even though it is correlated with "Life expectancy" (-0.7609), because it holds some information about population growth which is unique to this feature
- We keep "Health", because it has low correlation with every other feature and thus holds information unique to this feature (amount of money spent on healthcare)

Based on the above points, we will choose the following features to perform the clustering:

(**EXPERIMENT B**): GDPP, Inflation, Total fertility, Health, Life Expectancy

```
X_exp_B = X_stand_norm(:, [3, 6, 7, 8, 9]);
labels_exp_B = labels([3, 6, 7, 8, 9]);
```

4 Selection of the Clustering Algorithm(s)

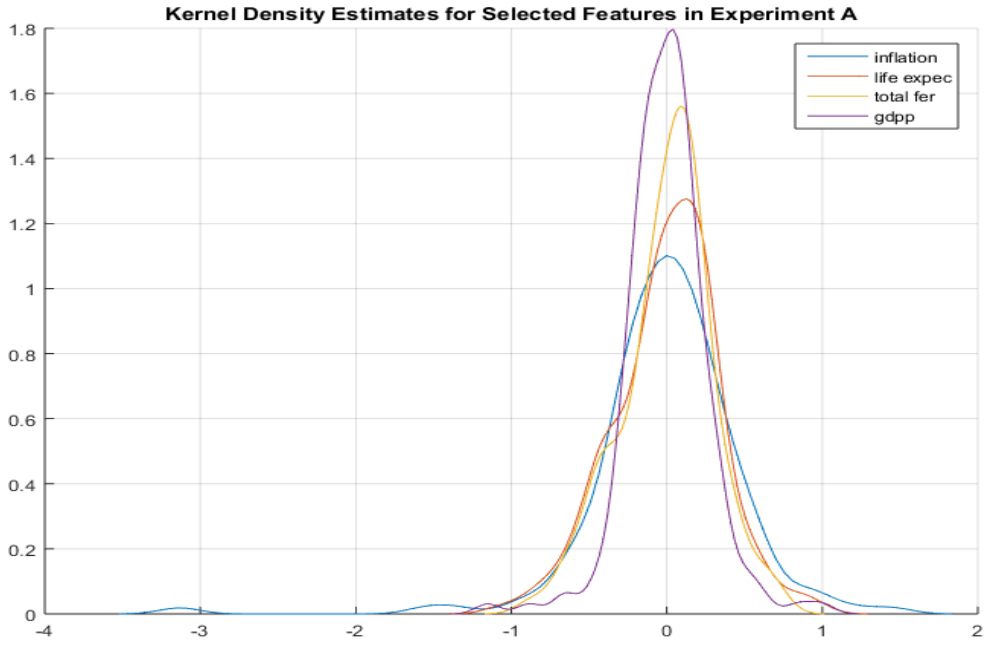
Based on the data exploration we previously performed, we need to select the proper cost function optimization clustering algorithm(s) that would be appropriate in our framework.

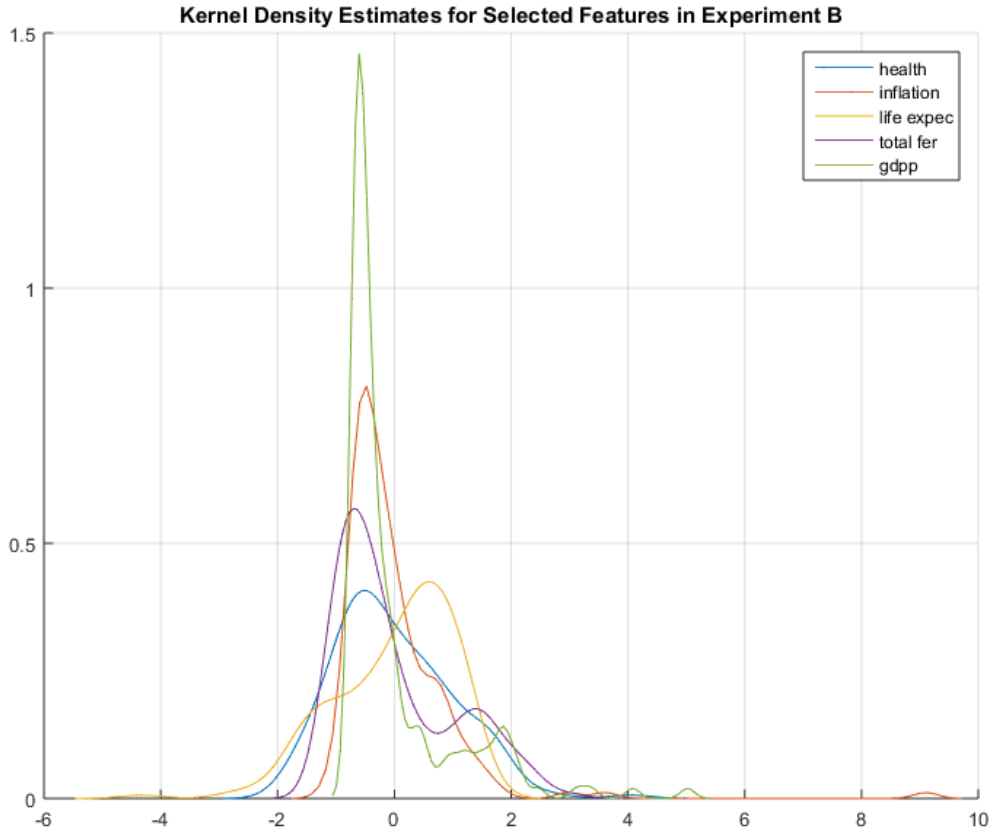
As we mentioned before, the goal of the analysis is to cluster each country into a distinct group, whose representative will provide information about the overall development of the countries included in that group. For that reason, we chose to represent the countries using the following feature combinations:

1. **Experiment A**: GDPP, Inflation, Total Fertility, Life Expectancy (PCA-based)
2. **Experiment B**: GDPP, Inflation, Total Fertility, Health, Life Expectancy (Correlation-based)

To identify the type of clusters we should expect, we plot the probability density function for each normalized feature (since they are continuous valued), in each experiment, using the following code:

```
if DISPLAY_FIGURES
    figure;
    grid on;
    hold on;
    for i = 1:4
        [f,xi] = ksdensity(X_exp_A(:,i));
        plot(xi, f, 'DisplayName', labels_exp_A{i});
    end
    hold off;
    legend('show');
    title('Kernel Density Estimates for Selected Features in Experiment A');
    drawnow;
end

if DISPLAY_FIGURES
    figure;
    grid on;
    hold on;
    for i = 1:5
        [f,xi] = ksdensity(X_exp_B(:,i));
        plot(xi, f, 'DisplayName', labels_exp_B{i});
    end
    hold off;
    legend('show');
    title('Kernel Density Estimates for Selected Features in Experiment B');
    drawnow;
end
```





In both cases, we observe a major overlap between the PDFs of the chosen features. This is an indication that we should be expecting **compact clusters** and, thus, we choose to represent them using **point representatives**.

To categorize the countries into groups based on the values of their features, we first consider following the Generalized Hard Clustering Algorithmic Scheme (GHAS), where each country exclusively belongs to a single cluster. Some popular approaches in this case are the following:

- **The K-Means Clustering Algorithm**: seeks to minimize the variance within each cluster by iteratively assigning each data point to the nearest cluster representative/centroid and then recomputing the centroids based on the current cluster memberships
- **The K-Medians Clustering Algorithm**: similar to K-Means but uses the median in each dimension to define the center of a cluster, which can make it more robust to outliers than K-Means
- **The K-Medoids Clustering Algorithm**: similar to K-Means but uses actual data points as the centers (medoids) instead of the mean and is more robust to noise and outliers compared to K-Means

Since we do not wish to restrict the cluster representatives to be data points from the dataset, we will not follow the K-Medoids approach. Our algorithm of choice will be the **K-Medians**, since the dataset is small and we do not want to be too affected by outliers. We will, however, utilize both **K-Means** and **K-Medians** (since they perform the calculations for the cluster representatives differently) in order to compare the results of the two different approaches and determine the number of physical clusters.

5 Execution of the Clustering Algorithm(s)

In this section, we will run the K-Means and K-Medians hard CFO clustering algorithms for various values of its parameters in order to determine the physical clusters formed by the data vectors and we will run the selected K-Medians algorithm to find the best possible clustering(s) each of our experiments. The specific implementations of the K-Means and K-Medians algorithms used here were taken from (c) 2010 S. Theodoridis, A. Pikrakis, K. Koutroumbas, D. Cavouras.

5.1 Identification of the Optimal Number of Clusters

First, the selected clustering algorithms will be run for a wide range of cluster numbers (2-20) and seed numbers (10, 200, 1000) to generate the elbow curve and perform the evalclusters analysis for a subset of the number of clusters. This approach enables the identification of the optimal number of clusters (K) that likely corresponds to the number of physical clusters in our dataset. Subsequently, we will employ the silhouette method to refine our evaluation for K. Finally, the K-Medians clustering algorithm will be re-run using the determined optimal number of clusters.

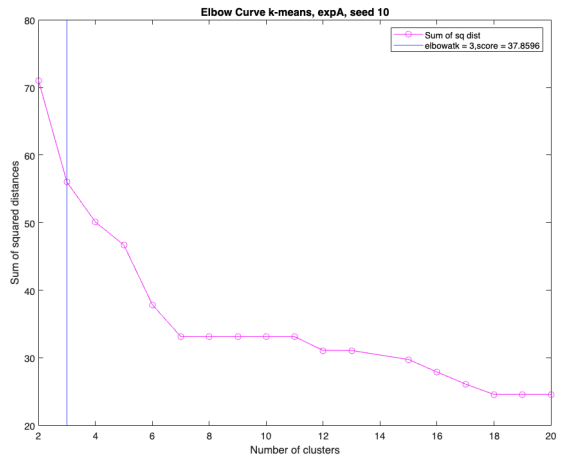
5.1.1 Elbow Method

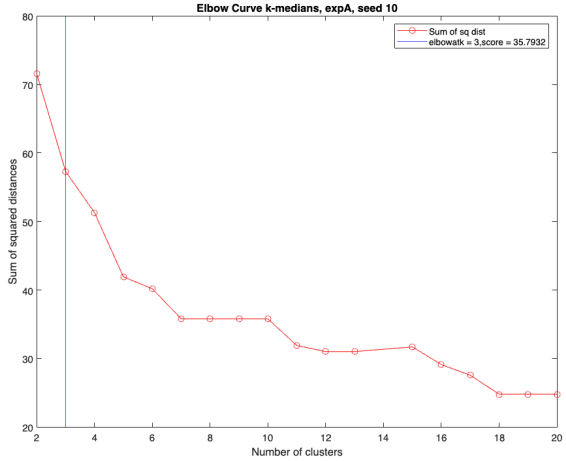
The Elbow method includes the plot of the the sum of the square distance between points and the representative of the cluster and the number of clusters for each iteration. The first featured "elbow" point of the curve will help us to define the number of physical clusters.

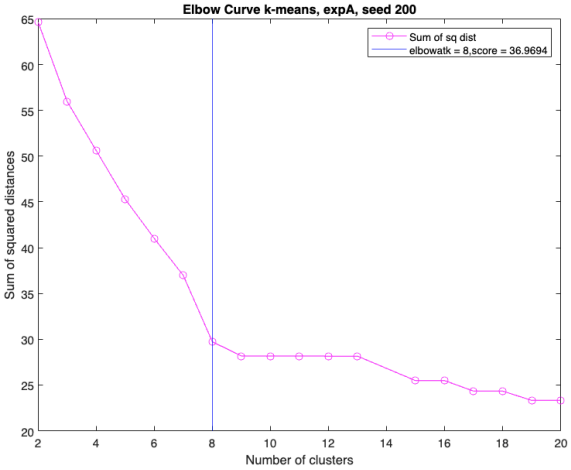
As there is no direct method for calculating the distortion score in the current library, it was computed by averaging the sums of squared distances. Distortion considers only the tightness of the clusters, where lower distortion indicates that the points within a cluster are closer to the center. The "elbow" point was identified by combining the local minimum and the difference between values of the sum mentioned above (similar to angle based Knee Detection for discrete dataset [12]).

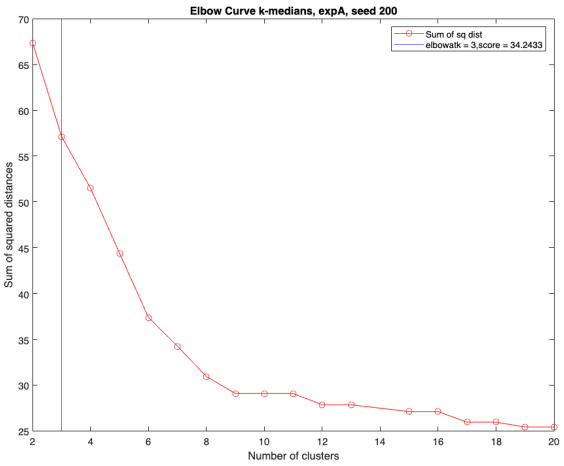
```
function [] = display_elbow_curve(clusters_matrix, J_history, plot_title, plot_style)
    distortion_score = mean(J_history);
    [~, closest_index] = min(abs(J_history - distortion_score));
    closest_value = J_history(closest_index);
    elbow_point = find_elbow_point(clusters_matrix, J_history);
    optimal_K = elbow_point;
    clusters_local_min = clusters_matrix(find(islocalmin(J_history),1));
    if clusters_local_min < optimal_K
        optimal_K = clusters_local_min;
    end
    figure;
    plot(clusters_matrix, J_history, plot_style);
    xline(optimal_K, '-b')
    xlabel('Number of clusters');
    ylabel('Sum of squared distances');
    title(plot_title);
    legend('Sum of sq dist', ['elbowatk = ' num2str(optimal_K) ',score = ' num2str(
    closest_value)])
    drawnow;
end

function optimal_k = find_elbow_point(k_values, sum_of_distances)
    diff1 = diff(sum_of_distances);
    diff2 = diff(diff1);
    [~, idx] = max(diff2);
    optimal_k = k_values(idx + 1);
end
```

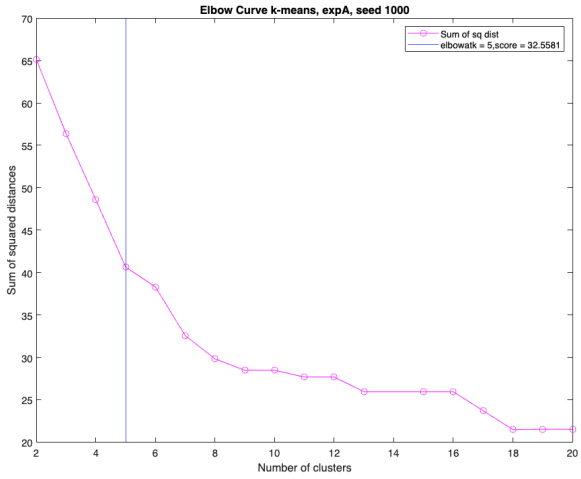
Experiment A:

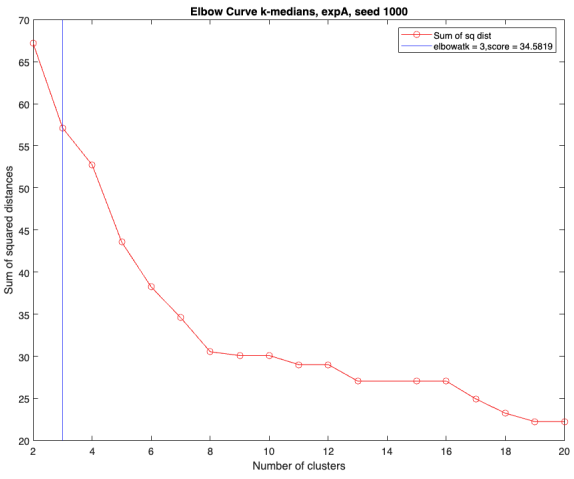






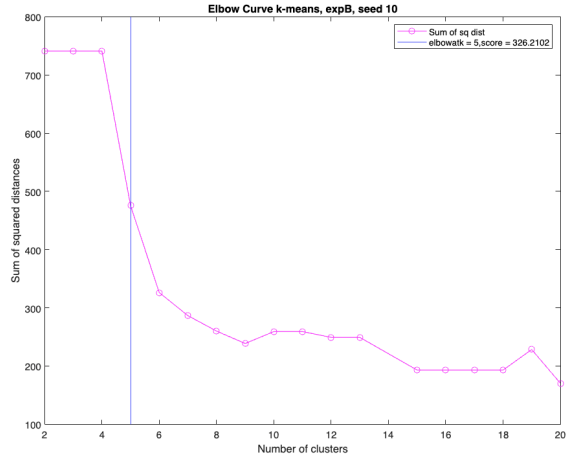


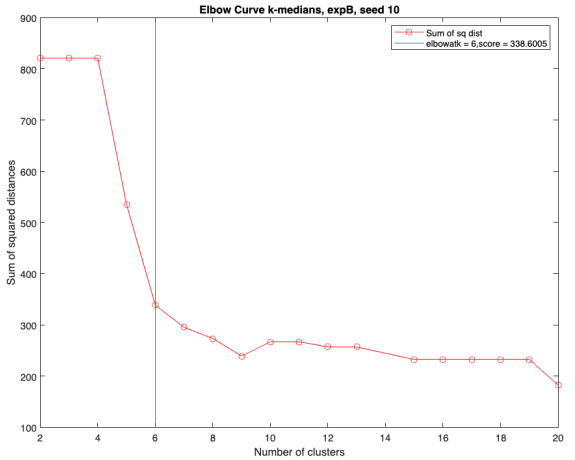


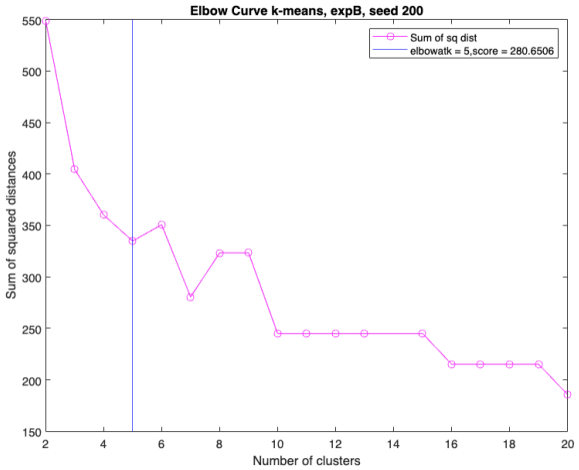


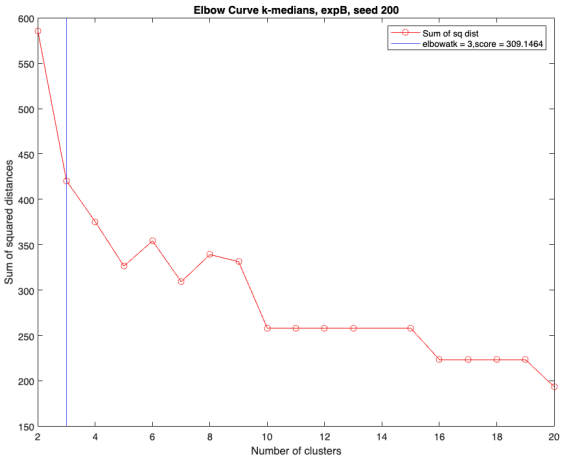
It's easy to observe that k-means exhibits higher sensitivity to the initial values of cluster representatives (theta) across different seed numbers compared to k-medians, leading to variations in the optimal K based on initialization. Additionally, the distortion score in k-means tends to be more variable compared to the same metric in k-medians. In contrast, k-medians utilizes the median value of data points within a cluster as the new centroid, providing greater stability in determining the optimal number of clusters, which in this case is 3.

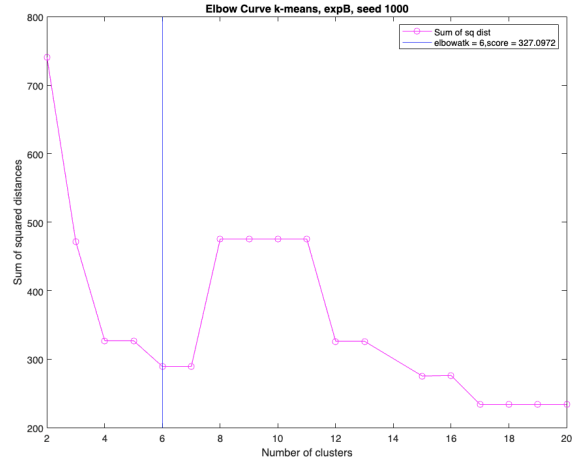
Experiment B:

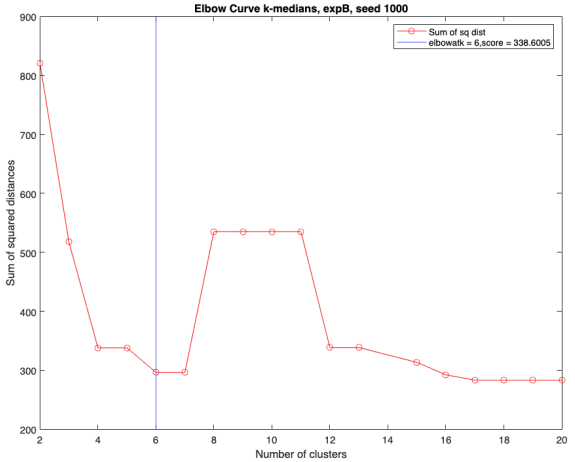












Using more features in this experiment, it becomes apparent that the optimal K is not as stable in both algorithms. Instead, this variable fluctuates within the range of [3, 4, 5, 6], and the previously mentioned mathematical solution fails to assist in determining the best number of clusters. This happens due to the sum of squared distances being a larger number and the mean of this value be more variable quantity (distortion score). Nevertheless, if we set the selected seed number to be 200, we can identify the elbow points that emerge, particularly for K = 3.

After the two experiments, the optimal number of clusters is likely 3, as an "elbow" is spotted there on average among the plots. However, in all of the plots, these "elbows" are not sufficiently defined, so we cannot be confident in our choice of number of clusters using this method alone. Therefore, we may need to consider additional methods to determine the optimal number of clusters.

! In many cases where the elbow curve is insufficient for identifying the appropriate 'K,' the silhouette method proves more robust. This is because the silhouette method takes into account both the cohesion within clusters and the separation between clusters.

5.1.2 Evalclusters

Following the above methods of determining the optimal K, we found the MATLAB's evalclusters method that allows us to evaluate clustering solutions for varying numbers of clusters (K) and aids in identifying the optimal K. The selected criteria to perform our experiments are: 'CalinskiHarabasz', 'DaviesBouldin', 'silhouette':

- **CalinskiHarabasz**: The Calinski-Harabasz Index, or Variance Ratio Criterion, measures the sum of between-cluster dispersion against the sum of within-cluster dispersion, where dispersion is the sum of distance squared.
- **DaviesBouldin**: The Davies-Bouldin Index is defined as the average similarity measure of each cluster with its most similar cluster, where similarity is the ratio of within-cluster distances to between-cluster distances. Thus, clusters which are farther apart and less dispersed will result in a better score.
- **silhouette**: The silhouette value is a measure of how similar an object is to its own cluster (cohesion) compared to other clusters (separation). The silhouette ranges from -1 to +1, where a high value indicates that the object is well matched to its own cluster and poorly matched to neighboring clusters.

! Using the above method (evalclusters), an error may occur when one or more clusters have zero elements. To address this issue, we added a try-catch block to prevent the script from failing in such instances.

```
eval_criterions = {'CalinskiHarabasz', 'DaviesBouldin', 'silhouette'};
function [] = preview_eval_clusters(data, clusters_history, eval_criterions)
    for criterion=1:size(eval_criterions, 2)
        try
            eva = evalclusters(data, clusters_history, eval_criterions(criterion));
            disp(eva)
        catch
            disp('Error: at least one cluster contains 0 elements.')
        end
    end
end
```

	Num Observations	Inspected K	CriterionValues	Optimal K
k-means, expA, seed 200, CalinskiHarabasz	167	[3 4 5]	[37.6927 33.3772 32.5521]	3
k-medians, expA, seed 200, CalinskiHarabasz	167	[3 4 5]	[38.4783 34.2476 36.6916]	3
k-means, expA, seed 200, DaviesBouldin	167	[3 4 5]	[1.6170 1.6316 1.4206]	5
k-medians, expA, seed 200, DaviesBouldin	167	[3 4 5]	[1.4683 1.3753 1.3231]	5
k-means, expA, seed 200, Silhouette	167	[3 4 5]	[0.2871 0.2280 0.2840]	3
k-medians, expA, seed 200, Silhouette	167	[3 4 5]	[0.2605 0.2366 0.2668]	5
k-means, expB, seed 200, CalinskiHarabasz	167	[3 4 5]	[86.3114 70.8686 59.8821]	3
k-medians, expB, seed 200, CalinskiHarabasz	167	[3 4 5]	[85.7295 69.4793 65.8254]	3
k-means, expB, seed 200, DaviesBouldin	167	[3 4 5]	[1.0477 1.1423 1.2118]	3
k-medians, expB, seed 200, DaviesBouldin	167	[3 4 5]	[1.0532 1.1310 1.1783]	3
k-means, expB, seed 200, Silhouette	167	[3 4 5]	[0.5037 0.5052 0.4710]	4
k-medians, expB, seed 200, Silhouette	167	[3 4 5]	[0.4966 0.4969 0.3495]	4

Table 5: Optimal number of clusters determined using the evalclusters MATLAB function.

The above results show that the most frequently observed optimal K is 3 when considering the selected number of clusters [3, 4, 5] as input. The second most frequently is the number 5 and the last one is the number 4.

5.1.3 Silhouette Method

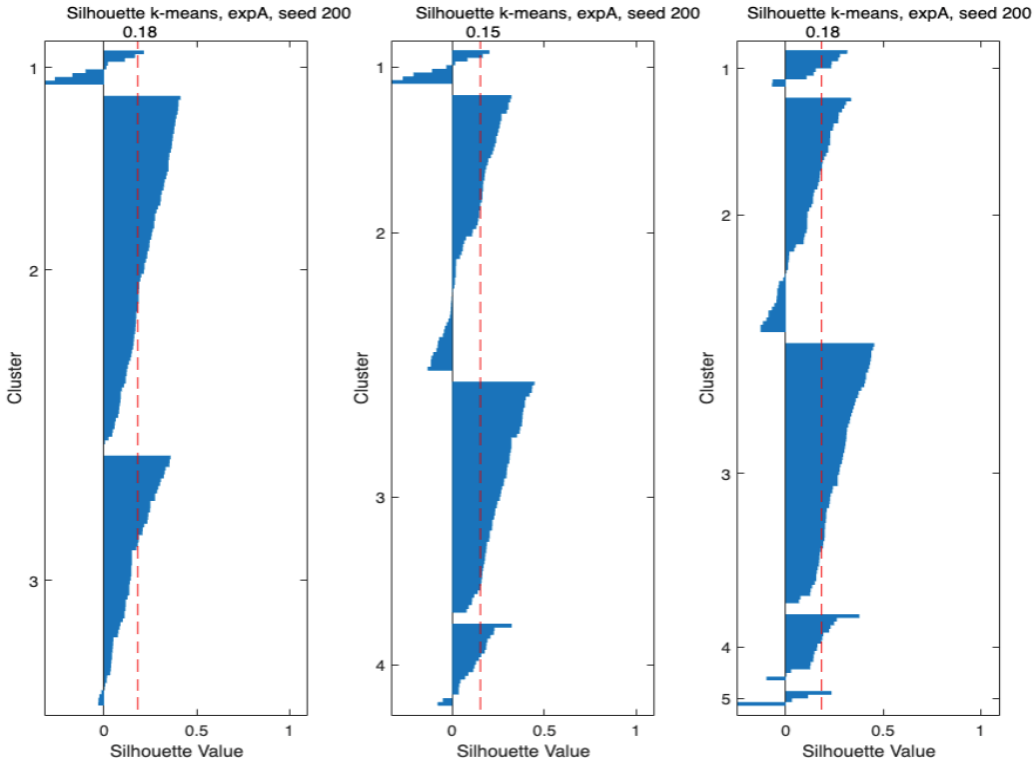
As above mentioned, the silhouette method measures how similar each point is to other points in the same cluster, compared to points in other clusters. We can calculate it using the following formula:

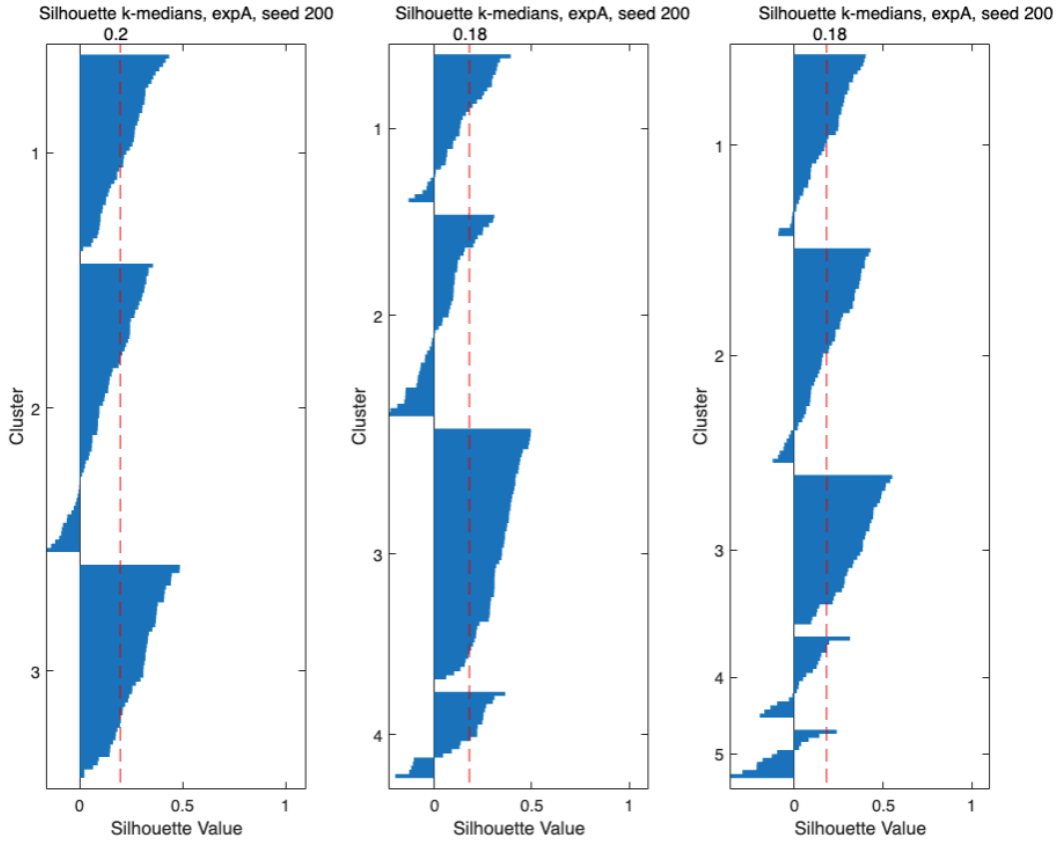
$$s_i = \frac{b_i - a_i}{max(a_i, b_i)},$$

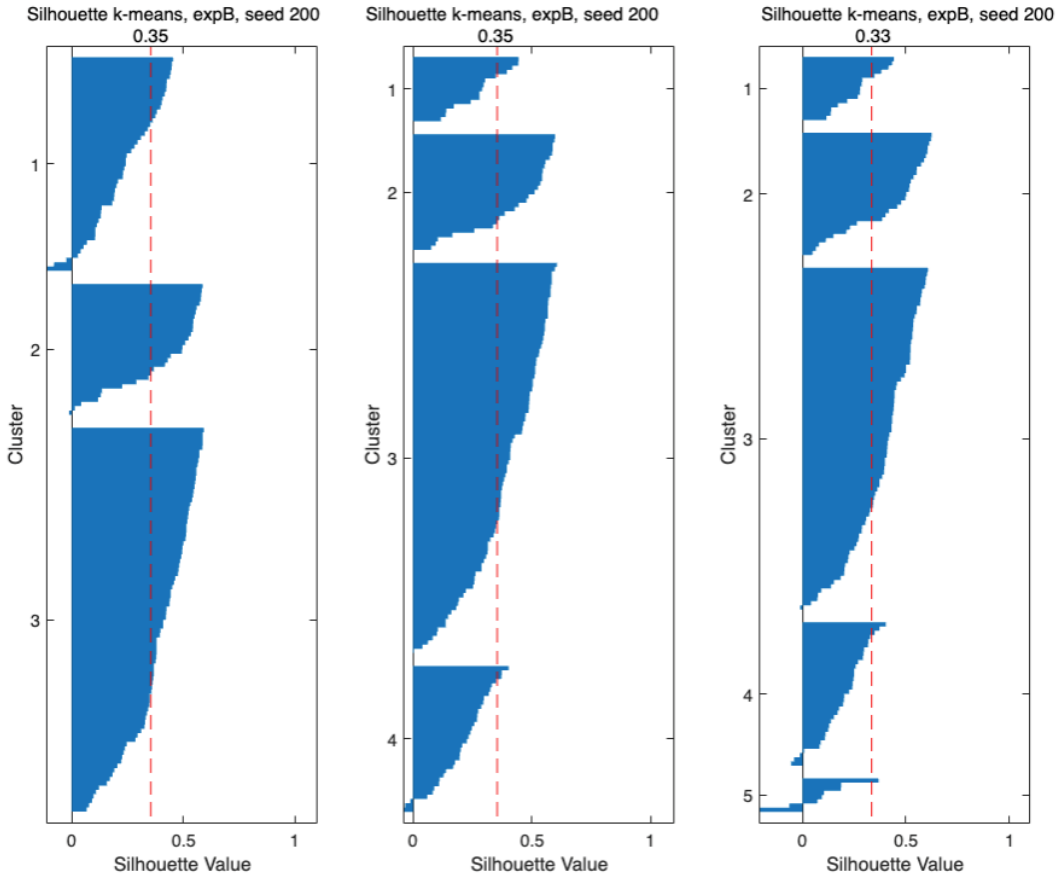
where a_i is the average distance from the i^{th} point to the other points in the same cluster as i, and b_i is the minimum average distance from the i^{th} point to points in a different cluster, minimized over the clusters. The silhouette values range from –1 to 1, where a high value indicates that the object is well matched to its own cluster and poorly matched to neighboring clusters. If most points have a high silhouette value, then the clustering solution is appropriate. On the other hand, if many points have a low or negative silhouette value, then the clustering solution might have too many or too few clusters.

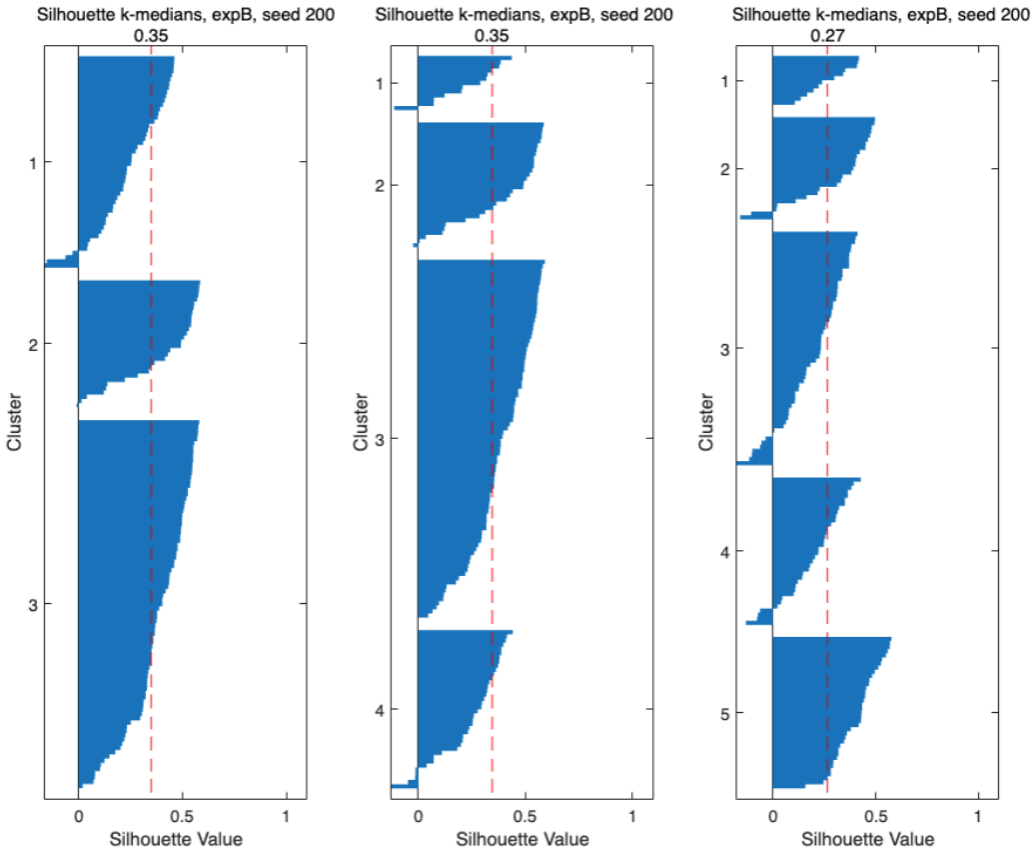
```
function [] = display_silhouette(data, clusters_history, title)
    num_columns = size(clusters_history, 2);
    plot_cols = int32(3);
    plot_rows = idivide(num_columns, plot_cols) + mod(num_columns, plot_cols);
    figure;
    for K=1:num_columns
         subplot(double(plot_rows), double(plot_cols), K)
         [s, h] = silhouette(data, clusters_history(:, K),'Euclidean');
         avg = mean(s);
         x = [avg avg];
         y = [0 500];
         line(x,y,'Color','red','LineStyle','--')
         text(avg - 0.08 , -2, num2str(avg,2))
         text(-0.2, -7, title)
    end
end
```

We executed the silhouette method with the chosen seed number 200, setting the range of the number of clusters [3, 4, 5] for both Experiment A (4 features) and Experiment B (5 features).




Silhouette k-medians, expA, seed 200
0.2
Cluster
Silhouette Value
Silhouette k-medians, expA, seed 200
0.18
Cluster
Silhouette Value
Silhouette k-medians, expA, seed 200
0.18
Cluster
Silhouette Value


Silhouette k-means, expB, seed 200
0.35
Cluster
Silhouette Value
Silhouette k-means, expB, seed 200
0.35
Cluster
Silhouette Value
Silhouette k-means, expB, seed 200
0.33
Cluster
Silhouette Value



In the following silhouette plots, the x-axis shows the silhouette values and the height of each silhouette indicates the number of points in the corresponding cluster. The red line shows the average silhouette value for all the clusters and shows us the balance of the overall cohesion and separation in all the clusters. Therefore, we're looking for the clustering with a higher mean silhouette value, ideally close to 1, but we already know from the Table 5, that the silhouette scores are low around 0.5. Contrary to the interpretation from this table, in this section we will observe the mean of the silhouette scores to take into account the overall performance of the clustering algorithms.

- **k-means expA**:
  The highest mean silhouette score is associated with K = 3 and K = 5. Upon closer inspection, we observe that for K = 3, the heights of clusters are more similar compared to K = 5, where cluster 5 exhibits a higher proportion of negative values than positive ones.
- **k-medians expA**:
  The k-medians algorithm yields higher mean silhouette scores than k-means. Due to the reasons mentioned earlier, the optimal number of clusters remains at 3. However, it's worth noting that for K = 2, in cluster 2, there are instances of negative silhouette values, indicating potential misalignment of points within this cluster.
- **k-means expB**:
  Observably, the highest mean silhouette scores are attained for K = 3 and K = 4. It is worth to mention that there is a subtle indication that cluster 1 for K = 3 is divided into two subclusters for K = 4 because the height of cluster 2 and 3 remained the same. Therefore, the optimal cluster number is still 3, as we aim for equi-dispersion between clusters.
- **k-medians expB**:
  This experiment has results similar to the previous one, favoring K = 3 and K = 4 due to their larger mean silhouette values. However, a slight difference is noted, with more and higher negative values indicating that some samples might have been assigned to the wrong cluster, as another cluster appears more similar.

The results reveal relatively weak clusters, with the maximum silhouette score reaching approximately 0.5. This outcome is somewhat expected given the limited data (167 samples) and the presence of numerous features that need to be considered in the clustering process.

The code that was used to execute the k-means and k-medians algorithms and to generate the graphs:

```
% ------------------------ k-means ------------------------

% EXPERIMENT A: execute clustering algorithms
[ theta_k_means_a, clusters_k_means_a, J_k_means_a, results_k_means_a] = execute_k_means
    (country, X_exp_A', clusters_num, seed_number);
exp_title = ['k-means, expA, seed ' num2str(seed_number)];
if DISPLAY_FIGURES
    display_silhouette(X_exp_A, clusters_k_means_a, ['Silhouette ' exp_title]);
    display_elbow_curve(clusters_num, J_k_means_a', ['Elbow Curve ' exp_title], '-om');
end
preview_eval_clusters(X_exp_A, clusters_k_means_a, eval_criterions)

% EXPERIMENT B: execute clustering algorithms
[ theta_k_means_b, clusters_k_means_b, J_k_means_b, results_k_means_b] = execute_k_means
    (country, X_exp_B', clusters_num, seed_number);
exp_title = ['k-means, expB, seed ' num2str(seed_number)];
if DISPLAY_FIGURES
    display_silhouette(X_exp_B, clusters_k_means_b, ['Silhouette ' exp_title]);
    display_elbow_curve(clusters_num, J_k_means_b', ['Elbow Curve ' exp_title], '-om');
end
preview_eval_clusters(X_exp_B, clusters_k_means_b, eval_criterions)

% ------------------------ k-medians ------------------------

% EXPERIMENT A: execute clustering algorithms
[ theta_k_medians_a, clusters_k_medians_a, J_k_medians_a, results_k_medians_a] =
    execute_k_medians(country, X_exp_A', clusters_num, seed_number);
exp_title = ['k-medians, expA, seed ' num2str(seed_number)];
if DISPLAY_FIGURES
    display_silhouette(X_exp_A, clusters_k_medians_a, ['Silhouette ' exp_title]);
    display_elbow_curve(clusters_num, J_k_medians_a', ['Elbow Curve ' exp_title], '-or')
    ;
end
preview_eval_clusters(X_exp_A, clusters_k_medians_a, eval_criterions)

% EXPERIMENT B: execute clustering algorithms
[ theta_k_medians_b, clusters_k_medians_b, J_k_medians_b, results_k_medians_b] =
    execute_k_medians(country, X_exp_B', clusters_num, seed_number);
exp_title = ['k-medians, expB, seed ' num2str(seed_number)];
if DISPLAY_FIGURES
    display_silhouette(X_exp_B, clusters_k_medians_b, ['Silhouette ' exp_title]);
    display_elbow_curve(clusters_num, J_k_medians_b', ['Elbow Curve ' exp_title], '-or')
    ;
end
preview_eval_clusters(X_exp_B, clusters_k_medians_b, eval_criterions)
```

The functions execute_k_means and execute_k_medians are shown below:

```
% ------------------------ k-means ------------------------
function [ theta, clusters_history, J_history, cluster_assignments] = execute_k_means(
    country, data, clusters_matrix, seed_number)
    cluster_assignments = cell(length(country), length(clusters_matrix) + 1);
    cluster_assignments(:, 1) = country;
    clusters_history = zeros(length(data), length(clusters_matrix));
    J_history = zeros(length(clusters_matrix), 1);
    for index = 1:length(clusters_matrix)
        clusters_num = clusters_matrix(index);
        theta_init = rand_init(data, clusters_num, seed_number);
        [ theta, clusters, J ] = k_means(data, theta_init);
        J_history(index) = J;
        for i = 1:length(clusters)+1
            cluster_assignments(clusters == i, index+1) = num2cell(i);
            clusters_history(:,index) = clusters(:);
        end
    end
end

```

```
function [ theta, clusters_history, J_history, cluster_assignments] = execute_k_medians(
    country, data, clusters_matrix, seed_number)
    cluster_assignments = cell(length(country), length(clusters_matrix) + 1);
    cluster_assignments(:, 1) = country;
    clusters_history = zeros(length(data), length(clusters_matrix));
    J_history = zeros(length(clusters_matrix), 1);
    for index = 1:length(clusters_matrix)
        clusters_num = clusters_matrix(index);
        theta_init = rand_init(data, clusters_num, seed_number);
        [ theta, clusters, J ] = k_medians(data, theta_init);
        J_history(index) = J;
        for i = 1:clusters+1
            cluster_assignments(clusters == i, index+1) = num2cell(i);
            clusters_history(:,index) = clusters(:);
        end
    end
end
```

5.2 Execution of the K-Medians Algorithm for the Optimal Number of Clusters

Having determined the optimal number of clusters (3), we will execute the K-Medians algorithm again for each of our experiments:

1. **Experiment A**: GDPP, Inflation, Total Fertility, Life Expectancy (PCA-based feature selection)
2. **Experiment B**: GDPP, Inflation, Total Fertility, Health, Life Expectancy (Correlation-based feature selection)

In each experiment, to figure out which features contribute more to the formation of the clusters, we will pairwise plot and compare the clustered features of the high-dimensional clustering (4D for experiment A, 5D for experiment B) with the corresponding 2-dimensional clustering, using 3 cluster representatives.

5.2.1 Experiment A

We use the following code:

```
% Number of features in Experiment A
num_features_A = 4;
feature_indices_A = 1:num_features_A; % Assuming features are the first 4
labels_A = labels_exp_A; % Assuming these are the labels for Experiment A

% Generate all combinations of 2 features for Experiment A
combinations_A = nchoosek(feature_indices_A, 2);

% Iterate over each combination for Experiment A
for i = 1:size(combinations_A, 1)
    % Select features for this combination
    features_2D_A = X_exp_A(:, combinations_A(i, :));

    % Running k-medians on these two features
    [theta_k_medians_a_2D, clusters_k_medians_a_2D, ~, ~] = execute_k_medians(country,
    features_2D_A', [3], seed_number);

    % Running k-medians for 4D data in Experiment A
    [theta_k_medians_a_4D, clusters_k_medians_a_4D, J_k_medians_a_4D,
    results_k_medians_a_4D] = execute_k_medians(country, X_exp_A', [3], seed_number);

    % Visualizing clustering results for Experiment A
    if DISPLAY_FIGURES
        figure; % Create a new figure for each pair of features
        set(gcf, 'Position', [100, 100, 1024, 512]); % Set a wider figure size for this
    figure

        % High-dimensional clustering (4D data)
        subplot('Position', [0.05, 0.1, 0.4, 0.8]); % Position as [left, bottom, width,
    height]
        scatter(X_exp_A(:, combinations_A(i, 1)), X_exp_A(:, combinations_A(i, 2)), 10,
    clusters_k_medians_a_4D, 'filled');
```

```
        title(['4D Clustering for Experiment A: ', labels_A{combinations_A(i, 1)}, ' & ', labels_A{combinations_A(i, 2)}]);
        xlabel(labels_A{combinations_A(i, 1)});
        ylabel(labels_A{combinations_A(i, 2)});
        axis square; % Make the current axes region square

        % 2D clustering for the same features
        subplot('Position', [0.55, 0.1, 0.4, 0.8]); % Position as [left, bottom, width, height]
        scatter(features_2D_A(:,1), features_2D_A(:,2), 10, clusters_k_medians_a_2D, 'filled');
        title(['2D Clustering for Experiment A: ', labels_A{combinations_A(i, 1)}, ' & ', labels_A{combinations_A(i, 2)}]);
        xlabel(labels_A{combinations_A(i, 1)});
        ylabel(labels_A{combinations_A(i, 2)});
        axis square; % Make the current axes region square

        drawnow;
    end
end
```

By comparing each 4 dimensional clustered pair with its corresponding 2 dimensional (produced by the above code but not shown in this report because they are not worth including), we observed that the shape of the clusters is generally not preserved in any feature pair. This means that, under the chosen conditions of this experiment, we cannot make strong claims about which of these features contribute more to the formation of these clusters.

5.2.2 Experiment B

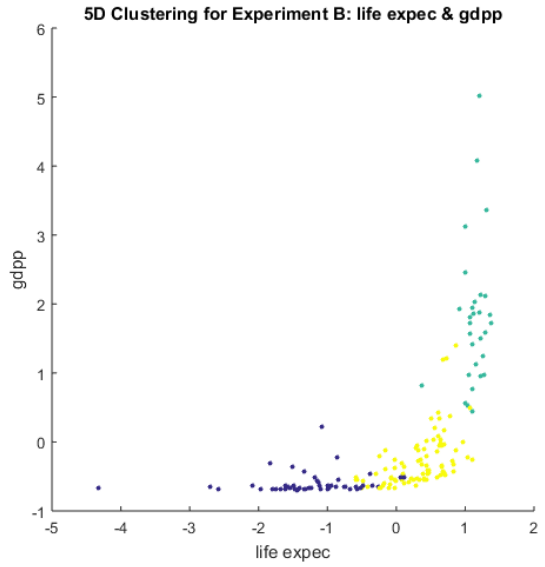
We use the following code:

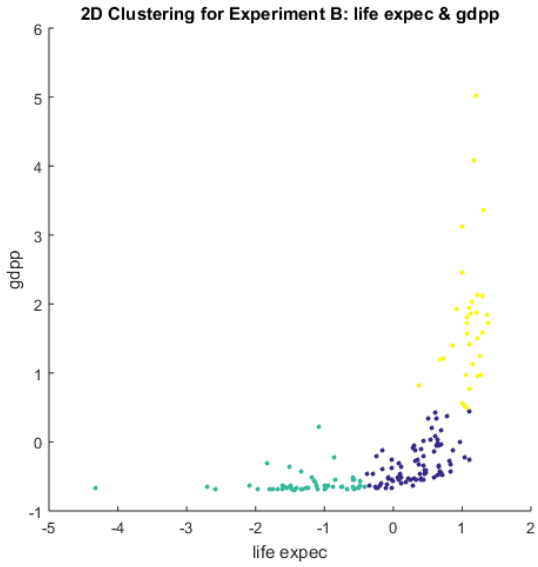
```
% Number of features in Experiment B
num_features_B = 5;
feature_indices_B = 1:num_features_B; % Assuming features are the first 5
labels_B = labels_exp_B; % Assuming these are the labels for Experiment B

% Generate all combinations of 2 features for Experiment B
combinations_B = nchoosek(feature_indices_B, 2);

% Iterate over each combination for Experiment B
for i = 1:size(combinations_B, 1)
    % Select features for this combination
    features_2D_B = X_exp_B(:, combinations_B(i, :));

    % Running k-medians on these two features
    [theta_k_medians_b_2D, clusters_k_medians_b_2D, ~, ~] = execute_k_medians(country, features_2D_B', [3], seed_number);

    % Running k-medians for 5D data in Experiment B
    [theta_k_medians_b_5D, clusters_k_medians_b_5D, J_k_medians_b_5D, results_k_medians_b_5D] = execute_k_medians(country, X_exp_B', [3], seed_number);


    % Visualizing clustering results for Experiment B
    if DISPLAY_FIGURES
        figure; % Create a new figure for each pair of features
    set(gcf, 'Position', [100, 100, 1024, 512]); % Set a wider figure size for this figure

    % High-dimensional clustering (5D data)
    subplot('Position', [0.05, 0.1, 0.4, 0.8]); % Position as [left, bottom, width, height]
    scatter(X_exp_B(:, combinations_B(i, 1)), X_exp_B(:, combinations_B(i, 2)), 10, clusters_k_medians_b_5D, 'filled');
    title(['5D Clustering for Experiment B: ', labels_B{combinations_B(i, 1)}, ' & ', labels_B{combinations_B(i, 2)}]);
    xlabel(labels_B{combinations_B(i, 1)});
    ylabel(labels_B{combinations_B(i, 2)});
    axis square; % Make the current axes region square

    % 2D clustering for the same features
```

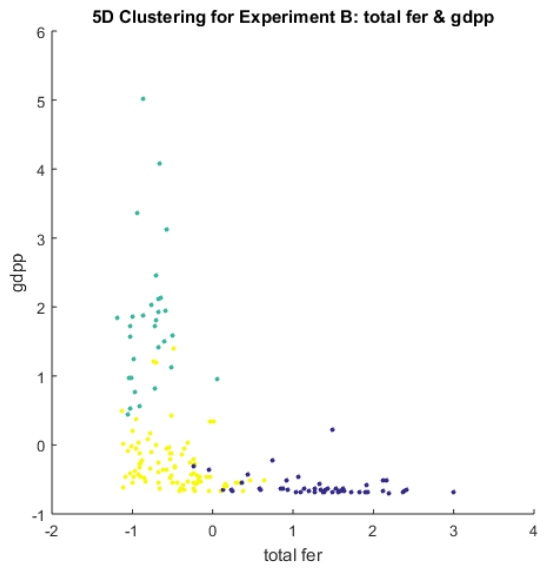
```
    subplot('Position', [0.55, 0.1, 0.4, 0.8]); % Position as [left, bottom, width,
    height]
    scatter(features_2D_B(:,1), features_2D_B(:,2), 10, clusters_k_medians_b_2D, 'filled
    ');
    title(['2D Clustering for Experiment B: ', labels_B{combinations_B(i, 1)}, ' & ',
    labels_B{combinations_B(i, 2)}]);
    xlabel(labels_B{combinations_B(i, 1)});
    ylabel(labels_B{combinations_B(i, 2)});
    axis square; % Make the current axes region square

    drawnow;
    end
end
```

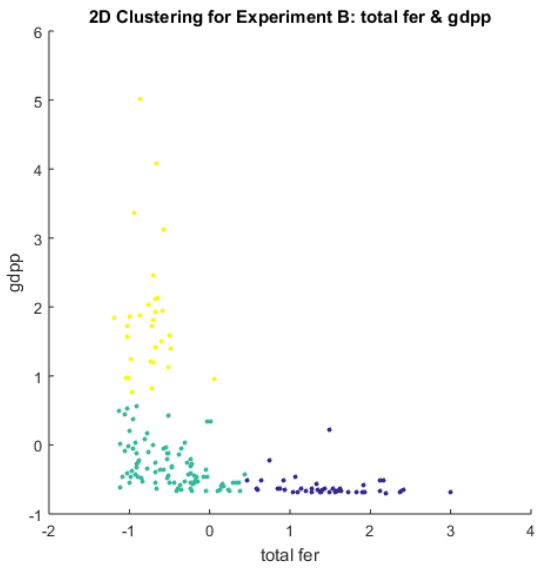
By comparing each 5 dimensional clustered pair with its corresponding 2 dimensional, we observed that the shape of the clusters is generally preserved in the following feature pairs:

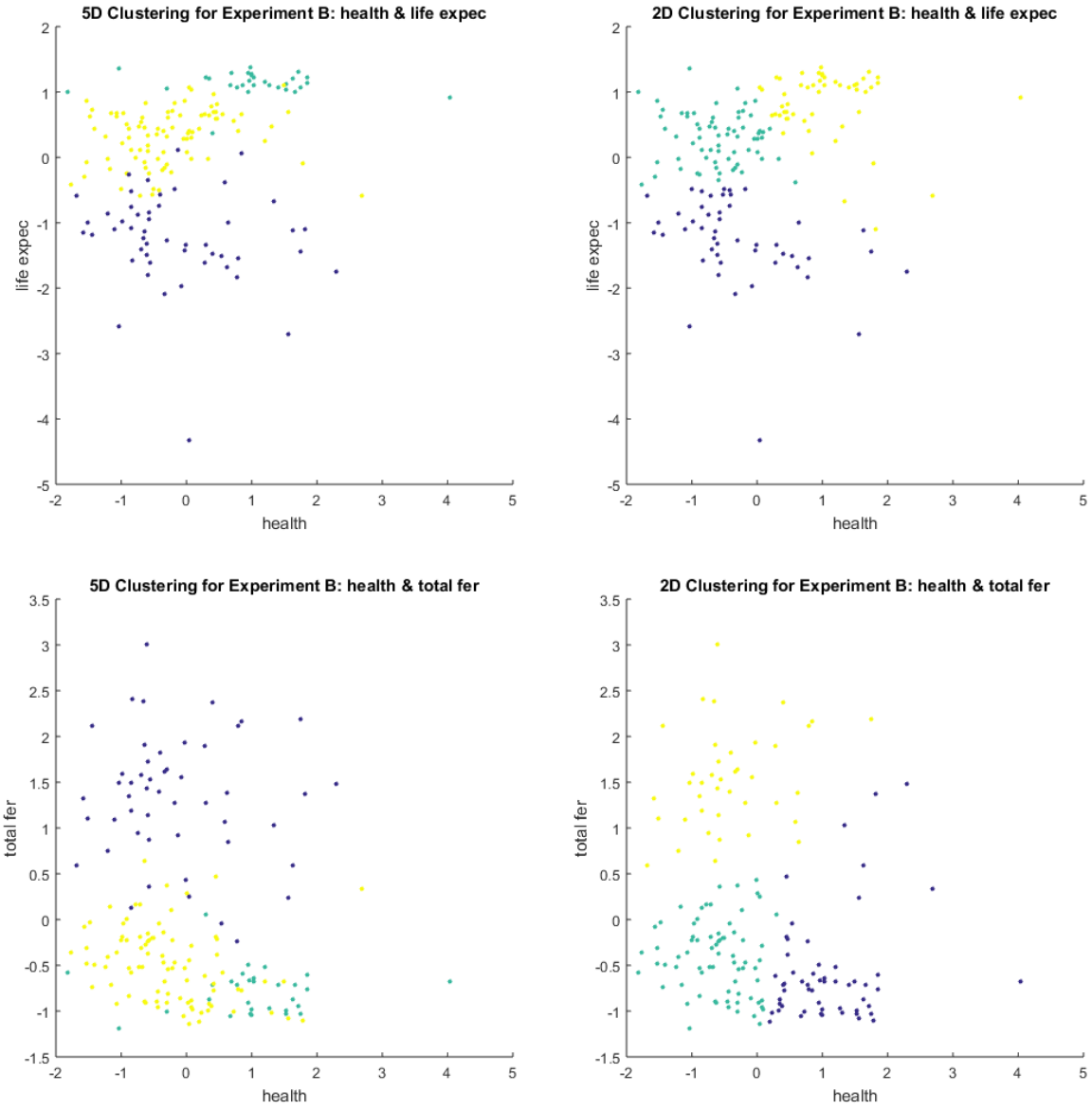
- Life Expectancy - GDPP
- Total Fertility - GDPP
- Health - Life Expectancy
- Health - Total Fertility











This means that these features pairs have strong relationships and, thus, contribute significantly into the clustering results, since the shape of their clusters is preserved even in higher dimensions.

6 Characterization of the Clusters

Having determined the clusters from the previous stage, we confirm that they are indeed compact. In this section we:

- Characterize each cluster, based on the values of the features that are encountered among its data points (this includes descriptive statistics and approximation and visualization of the pdf for each cluster)
- Compare the different clusters that resulted from our above analysis, based on their characterization

6.1 Experiment A

We start by identifying the final cluster representatives:

```
% Identify the final cluster representatives
final_representatives_A = theta_k_medians_a_4D; % Final representatives
```

Features	Cluster 1 Representative	Cluster 2 Representative	Cluster 3 Representative
inflation	0.0013	-0.2039	0.3615
life expec	-0.3587	0.1606	0.0921
total fer	0.1091	-0.0489	0.0870
gdpp	0.0758	-0.0185	-0.0455

Table 6: Final Cluster Representatives

We proceed by analyzing each of the clusters:

Cluster 1:

We calculate the descriptive statistics for each feature of the cluster:

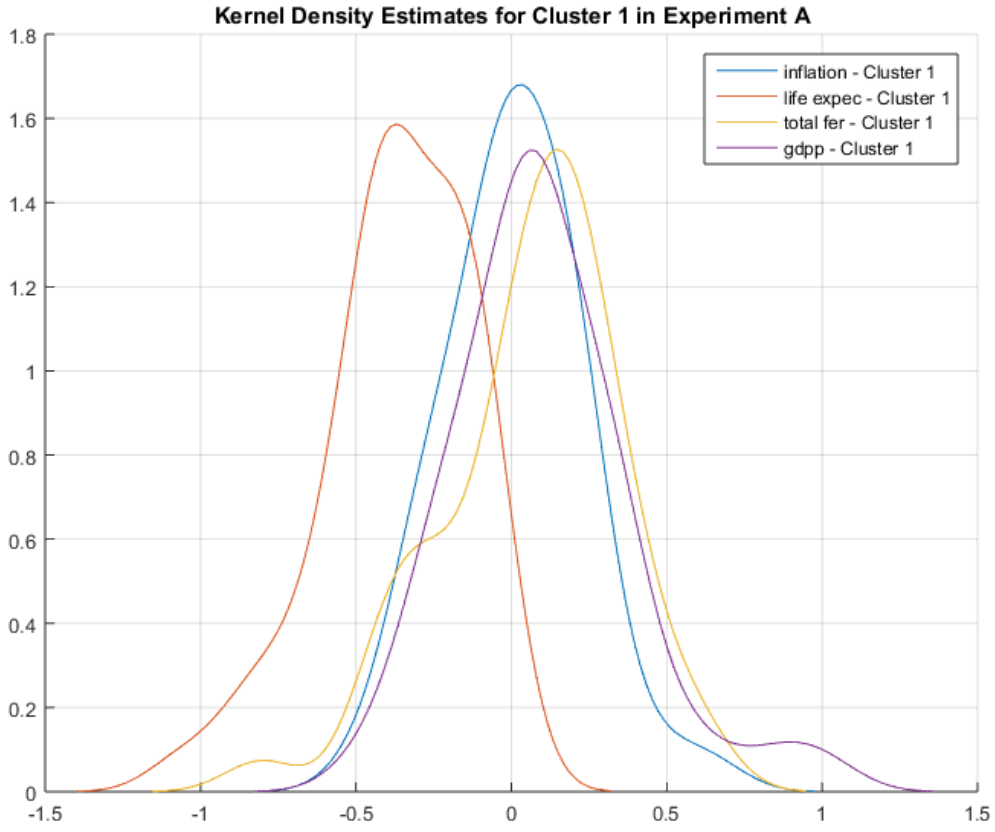
```
% Calculate detailed statistics for each feature within each cluster
% Preallocate statistics for each cluster
cluster_stats_A = cell(3, 1);

% Loop through each cluster to compute and store statistics
for k = 1:3
    % Extract the data points belonging to cluster k
    cluster_data = X_exp_A(clusters_k_medians_a_4D == k, :);

    % Get detailed statistics for each feature in cluster k
    [min_vals, max_vals, mean_vals, median_vals, var_vals, std_vals, percentiles] =
    data_statistics(cluster_data);

    % Combine the statistics into a matrix and store in the cell array
    cluster_stats_A{k} = [min_vals; max_vals; mean_vals; median_vals; var_vals; std_vals
    ; percentiles];
end
```

	inflation	life expec	total fer	gdpp
min	-0.4678	-1.0599	-0.8046	-0.4609
max	0.6312	-0.0107	0.6041	0.9957
mean	0.0111	-0.3657	0.0620	0.1017
median	0.0013	-0.3587	0.1091	0.0758
variance	0.0467	0.0548	0.0823	0.0802
std	0.2162	0.2340	0.2868	0.2832
25%	-0.1103	-0.4601	-0.0804	-0.0759
50%	0.0013	-0.3587	0.1091	0.0758
75%	0.1630	-0.1706	0.2189	0.2877

Table 7: Statistics on Cluster 1 of Experiment A

We then plot the probability density functions (PDFs) for each feature in the cluster:

```
% Calculate detailed statistics for each feature within each cluster
% Preallocate statistics for each cluster
cluster_stats_A = cell(3, 1);

% Loop through each cluster to compute and store statistics
for k = 1:3
    % Extract the data points belonging to cluster k
    cluster_data = X_exp_A(clusters_k_medians_a_4D == k, :);

    % Get detailed statistics for each feature in cluster k
    [min_vals, max_vals, mean_vals, median_vals, var_vals, std_vals, percentiles] =
    data_statistics(cluster_data);

    % Combine the statistics into a matrix and store in the cell array
    cluster_stats_A{k} = [min_vals; max_vals; mean_vals; median_vals; var_vals; std_vals
    ; percentiles];
end
```



We then compare the standard deviations and mean values for each feature of the cluster to the original standardized dataset:

```
% Compare the standard deviations and mean values for each feature of each cluster to
    the original standardized dataset
% Initialize the storage for the comparison metrics
cluster_comparison_A = cell(3, 1);

for k = 1:3
    % Extract the data for cluster k
    cluster_k_data = X_exp_A(clusters_k_medians_a_4D == k, :);

    % Calculate the statistics for cluster k
    [cluster_min, cluster_max, cluster_mean, cluster_median, cluster_var, cluster_std,
    cluster_percentiles] = data_statistics(cluster_k_data);

```

```
    % Calculate the differences from the original standardized data
    mean_difference = cluster_mean; % Since the original mean is 0 for all features
    std_difference = cluster_std - 1; % Since the original std is 1 for all features

    % Store the results
    cluster_comparison_A{k}.mean_difference = mean_difference;
    cluster_comparison_A{k}.std_difference = std_difference;
end
```

	inflation	**life expec**	**total fer**	**gdpp**
std	-0.7838	-0.7660	-0.7132	-0.7168
mean	0.0111	-0.3657	0.0620	0.1017

Table 8: Standard Deviation and Mean Value Differences between Cluster 1 of Experiment A and the Original Standardized Dataset

- **Inflation**: big negative change in standard deviation (-0.7838, values more concentrated around the mean), very small positive change in the mean value (0.0111, cluster center shifted towards higher values)
- **Life Expectancy**: big negative change in standard deviation (-0.7660, values more concentrated around the mean), small negative change in the mean value (-0.3657, cluster center shifted towards lower values)
- **Total Fertility**: big negative change in standard deviation (-0.7132, values more concentrated around the mean), very small positive change in the mean value (0.0620, cluster center shifted towards higher values)
- **GDPP**: big negative change in standard deviation (-0.7168, values more concentrated around the mean), very small positive change in the mean value (0.1017, cluster center shifted towards higher values)

So this cluster is mostly characterized by countries with a little higher than average inflation, lower than average life expectancy, a little higher than average total fertility and a little higher than average GDPP across all of them.

Finally, we list all the countries belonging to this cluster:

```
% List the countries belonging to each cluster
% Initialize a cell array to store the country names for each cluster
countries_in_clusters_A = cell(3, 1);

for k = 1:3
    % Find the indices of the countries belonging to the k-th cluster
    indices = find(clusters_k_medians_a_4D == k);

    % Store the country names in the cell array
    countries_in_clusters_A{k} = country(indices);
end
```

Countries in Cluster 1 (Experiment A): 'Afghanistan', 'Albania', 'Antigua and Barbuda', 'Armenia', 'Australia', 'Bahamas', 'Bangladesh', 'Bhutan', 'Canada', 'Cape Verde', 'Comoros', 'Cyprus', 'Denmark', 'El Salvador', 'Eritrea', 'Finland', 'Ghana', 'Grenada', 'Guinea', 'Guyana', 'Italy', 'Jamaica', 'Japan', 'Kiribati', 'Kyrgyz Republic', 'Lebanon', 'Liberia', 'Luxembourg', 'Madagascar', 'Micronesia Fed Sts', 'Mongolia', 'Montenegro', 'Nepal', 'Nigeria', 'Norway', 'Pakistan', 'Seychelles', 'Solomon Islands', 'Sri Lanka', 'St. Vincent and the Grenadines', 'Sweden', 'Switzerland', 'Tajikistan', 'Tonga', 'United Arab Emirates', 'United Kingdom', 'Venezuela'

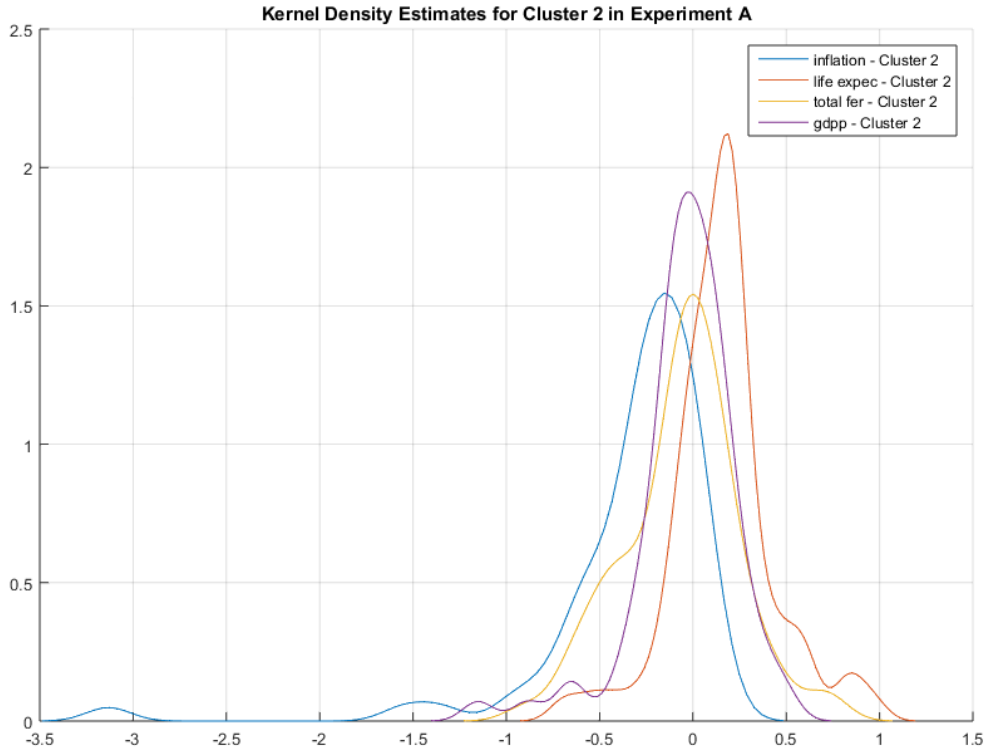
Cluster 2:

We calculate the descriptive statistics for each feature of the cluster:

	inflation	**life expec**	**total fer**	**gdpp**
min	-3.1323	-0.6901	-0.8949	-1.1518
max	0.1334	0.9609	0.7386	0.4963
mean	-0.3195	0.1565	-0.0641	-0.0409
median	-0.2039	0.1606	-0.0489	-0.0185
variance	0.2177	0.0772	0.0952	0.0742
std	0.4666	0.2779	0.3086	0.2724
25%	-0.3996	0.0221	-0.2178	-0.1289
50%	-0.2039	0.1606	-0.0489	-0.0185
75%	-0.0487	0.2545	0.1152	0.1234

Table 9: Statistics on Cluster 2 of Experiment A

We then plot the probability density functions (PDFs) for each feature in the cluster:



We then compare the standard deviations and mean values for each feature of the cluster to the original standardized dataset:

	inflation	**life expec**	**total fer**	**gdpp**
std	-0.5334	-0.7221	-0.6914	-0.7276
mean	-0.3195	0.1565	-0.0641	-0.0409

Table 10: Standard Deviation and Mean Value Differences between Cluster 2 of Experiment A and the Original Standardized Dataset

- **Inflation**: medium negative change in standard deviation (-0.5334, values more concentrated around the mean), small negative change in the mean value (-0.3195, cluster center shifted towards lower values)
- **Life Expectancy**: big negative change in standard deviation (-0.7221, values more concentrated around the mean), very small positive change in the mean value (0.1565, cluster center shifted towards higher values)
- **Total Fertility**: big negative change in standard deviation (-0.6914, values more concentrated around the mean), very small negative change in the mean value (-0.0641, cluster center shifted towards lower values)
- **GDPP**: big negative change in standard deviation (-0.7276, values more concentrated around the mean), very small negative change in the mean value (-0.0409, cluster center shifted towards lower values)

So this cluster is mostly characterized by countries with a little lower than average inflation (but more spread out compared to cluster 1), a little higher than average life expectancy, a little lower than average total fertility and a little lower than average GDPP across all of them.

Finally, we list all the countries belonging to this cluster:

Countries in Cluster 2 (Experiment A): 'Argentina', 'Austria', 'Azerbaijan', 'Bahrain', 'Barbados', 'Belarus', 'Bosnia and Herzegovina', 'Botswana', 'Brazil', 'Brunei', 'Bulgaria', 'Cambodia', 'Cameroon', 'Central African Republic', 'China', 'Colombia',' Cote d"Ivoire', 'Croatia', 'Czech Republic', 'Equatorial Guinea', 'Estonia', 'Fiji', 'FYROM', 'Gabon', 'Georgia', 'Germany', 'Greece', 'Guinea-Bissau', 'Haiti', 'Hungary', 'India', 'Indonesia', 'Iran', 'Kazakhstan', 'Kuwait', 'Lao', 'Latvia', 'Lesotho', 'Lithuania', 'Malawi', 'Malaysia', 'Mauritius', 'Moldova', 'Mozambique', 'Myanmar', 'Namibia', 'Poland', 'Portugal', 'Qatar', 'Romania', 'Russia', 'Saudi Arabia', 'Serbia', 'Sierra Leone', 'Slovak Republic', 'South Africa', 'South Korea', 'Spain', 'Suriname', 'Thailand', 'Togo', 'Turkmenistan', 'Ukraine', 'United States', 'Uruguay', 'Uzbekistan', 'Vanuatu', 'Vietnam', 'Zambia'

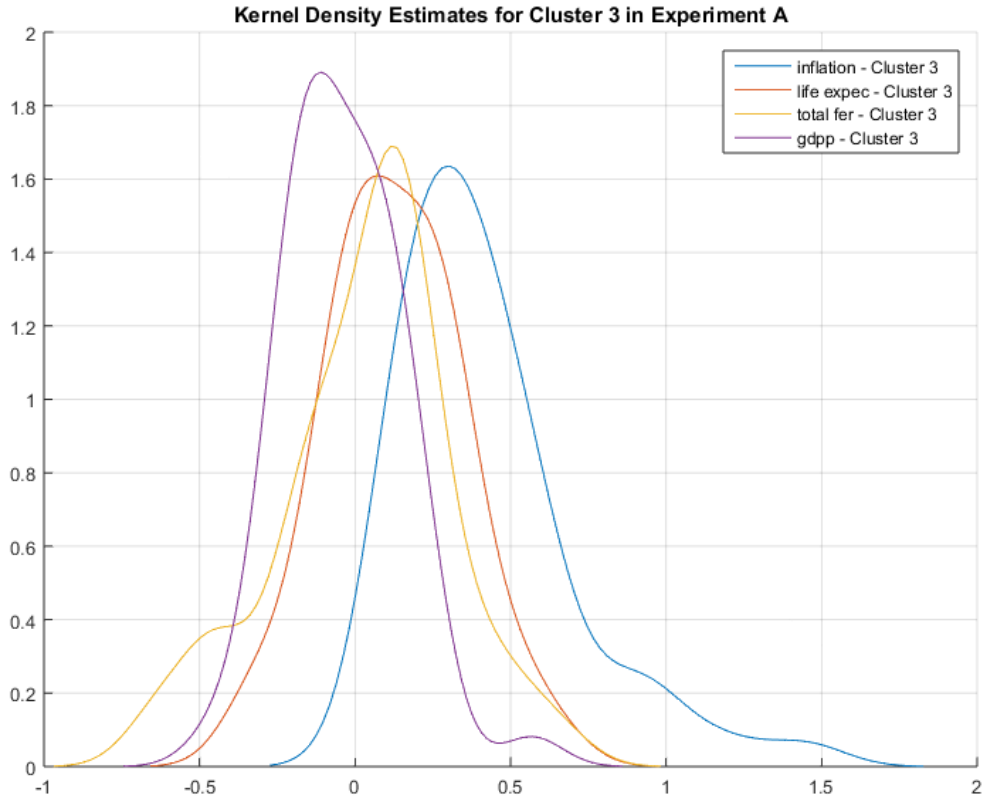
Cluster 3:

We calculate the descriptive statistics for each feature of the cluster:

	inflation	**life expec**	**total fer**	**gdpp**
min	0.1082	-0.3284	-0.6499	-0.4517
max	1.4477	0.6483	0.6639	0.5693
mean	0.4420	0.1253	0.0296	-0.0385
median	0.3615	0.0921	0.0870	-0.0455
variance	0.0806	0.0457	0.0778	0.0338
std	0.2839	0.2137	0.2790	0.1838
25%	0.2208	-0.0353	-0.1229	-0.1789
50%	0.3615	0.0921	0.0870	-0.0455
75%	0.5436	0.2711	0.1893	0.0946

Table 11: Statistics on Cluster 3 of Experiment A

We then plot the probability density functions (PDFs) for each feature in the cluster:



We then compare the standard deviations and mean values for each feature of the cluster to the original standardized dataset:

	inflation	**life expec**	**total fer**	**gdpp**
std	-0.7161	-0.7863	-0.7210	-0.8162
mean	0.4220	0.1253	0.0296	-0.0385

Table 12: Standard Deviation and Mean Value Differences between Cluster 3 of Experiment A and the Original Standardized Dataset

- **Inflation**: big negative change in standard deviation (-0.7161, values more concentrated around the mean), medium positive change in the mean value (0.4220, cluster center shifted towards higher values)
- **Life Expectancy**: big negative change in standard deviation (-0.7863, values more concentrated around the mean), very small positive change in the mean value (0.1253, cluster center shifted towards higher values)
- **Total Fertility**: big negative change in standard deviation (-0.7210, values more concentrated around the mean), very small positive change in the mean value (0.0296, cluster center shifted towards higher values)
- **GDPP**: very big negative change in standard deviation (-0.8162, values more concentrated around the mean), very small negative change in the mean value (-0.0385, cluster center shifted towards lower values)

So this cluster is mostly characterized by countries with a higher than average inflation, a little higher than average life expectancy, a little higher than average total fertility and a little lower than average GDPP across all of them.

Finally, we list all the countries belonging to this cluster:

Countries in Cluster 3 (Experiment A): 'Algeria', 'Angola', 'Belgium', 'Belize', 'Benin', 'Bolivia', 'Burkina Faso', 'Burundi', 'Chad', 'Chile', 'Congo Dem Rep', 'Congo Rep', 'Costa Rica', 'Dominican Republic', 'Ecuador', 'Egypt', 'France', 'Gambia', 'Guatemala', 'Iceland', 'Iraq', 'Ireland', 'Israel', 'Jordan', 'Kenya', 'Libya', 'Maldives', 'Mali', 'Malta', 'Mauritania', 'Morocco', 'Netherlands', 'New Zealand', 'Niger', 'Oman', 'Panama', 'Paraguay', 'Peru', 'Philippines', 'Rwanda', 'Samoa', 'Senegal', 'Singapore', 'Slovenia', 'Sudan', 'Tanzania', 'Timor-Leste', 'Tunisia', 'Turkey', 'Uganda', 'Yemen'

Overall interpretation:

- **Cluster 1** seems to represent countries with relatively higher economic development (higher GDPP) but lower life expectancy
- **Cluster 2** seems to consist of countries with lower inflation and slightly better life expectancy, possibly indicating stable economies
- **Cluster 3** seems to characterize countries with higher inflation rates, suggesting more economic variability or instability

6.2 Experiment B

We start by identifying the final cluster representatives:

```
% Identify the final cluster representatives
final_representatives_B = theta_k_medians_b_5D; % Final representatives
```

Features	Cluster 1 Representative	Cluster 2 Representative	Cluster 3 Representative
health	-0.4153	0.9900	-0.3443
inflation	0.0410	-0.6396	-0.1695
life expec	-1.1588	1.1294	0.3985
total fer	1.3786	-0.7418	-0.5139
gdpp	-0.6537	1.7178	-0.4100

Table 13: Final Cluster Representatives

We proceed by analyzing each of the clusters:

Cluster 1:

We calculate the descriptive statistics for each feature of the cluster:

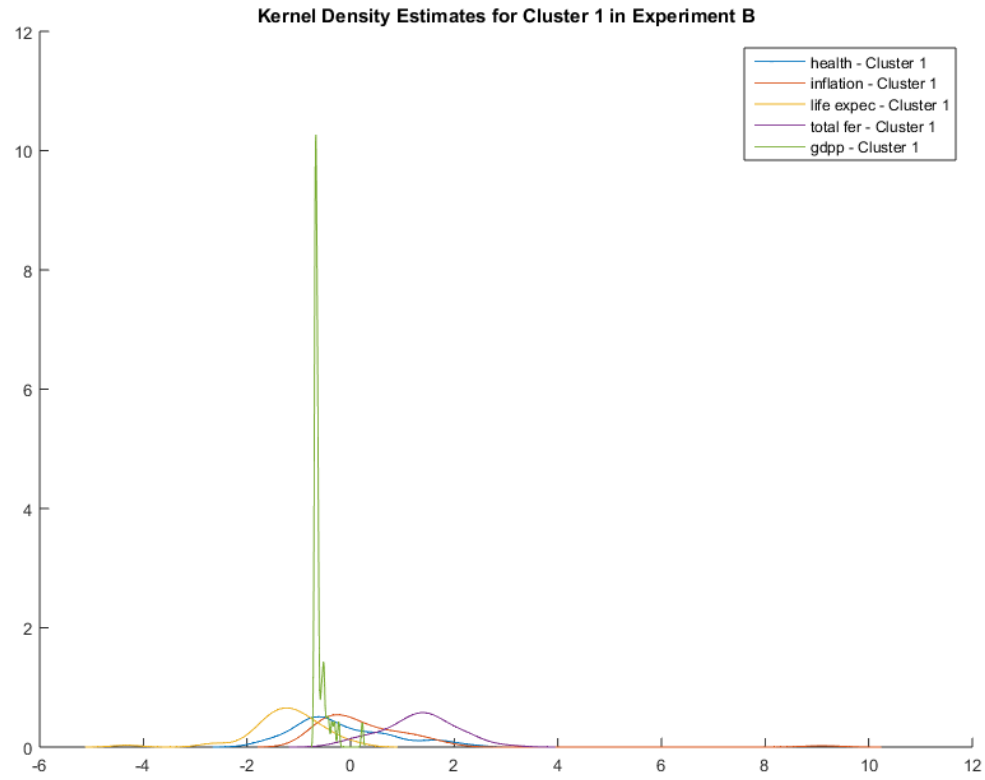
```
% Calculate detailed statistics for each feature within each cluster
% Preallocate statistics for each cluster
cluster_stats_B = cell(3, 1);

% Loop through each cluster to compute and store statistics
for k = 1:3
    cluster_data = X_exp_B(clusters_k_medians_b_5D == k, :);
    [min_vals, max_vals, mean_vals, median_vals, var_vals, std_vals, percentiles] =
    data_statistics(cluster_data);
    cluster_stats_B{k} = [min_vals; max_vals; mean_vals; median_vals; var_vals; std_vals
    ; percentiles];
end
```

	health	inflation	life expec	total fer	gdpp
min	-1.6804	-0.6524	-4.3242	-0.2365	-0.6947
max	2.2878	9.1023	0.1062	3.0003	0.2256
mean	-0.1475	0.3538	-1.2364	1.3220	-0.6003
median	-0.4153	0.0410	-1.1588	1.3786	-0.6537
variance	0.8978	2.0592	0.5355	0.4722	0.0246
std	0.9475	1.4350	0.7318	0.6872	0.1570
25%	-0.8321	-0.3994	-1.5468	0.9195	-0.6771
50%	-0.4153	0.0410	-1.1588	1.3786	-0.6537
75%	0.5404	0.8342	-0.8496	1.7254	-0.6260

Table 14: Statistics on Cluster 1 of Experiment B

We then plot the probability density functions (PDFs) for each feature in the cluster:

```
% Plot the probability density functions (PDFs) of features for each cluster
for k = 1:3
    if DISPLAY_FIGURES
        figure;
        hold on;
        for j = 1:num_features_B
            data_feature = X_exp_B(clusters_k_medians_b_5D == k, j);
            [f,xi] = ksdensity(data_feature);
            plot(xi, f, 'DisplayName', [labels_exp_B{j} ' - Cluster ' num2str(k)]);
        end
        hold off;
        legend('show');
        title(['Kernel Density Estimates for Cluster ' num2str(k) ' in Experiment B']);
        drawnow;
    end
end
```



We then compare the standard deviations and mean values for each feature of the cluster to the original standardized dataset:

```
% Compare the standard deviations and mean values for each cluster to the original
    standardized dataset
cluster_comparison_B = cell(3, 1);

for k = 1:3
    cluster_k_data = X_exp_B(clusters_k_medians_b_5D == k, :);
    [cluster_min, cluster_max, cluster_mean, cluster_median, cluster_var, cluster_std,
    cluster_percentiles] = data_statistics(cluster_k_data);
    mean_difference = cluster_mean; % Since the original mean is 0 for all features
    std_difference = cluster_std - 1; % Since the original std is 1 for all features
    cluster_comparison_B{k}.mean_difference = mean_difference;
    cluster_comparison_B{k}.std_difference = std_difference;
end
```

	health	**inflation**	**life expec**	**total fer**	**gdpp**
std	-0.0525	0.4350	-0.2682	-0.3128	-0.8430
mean	-0.1475	0.3538	-1.2364	1.3220	-0.6003

Table 15: Standard Deviation and Mean Value Differences between Cluster 1 of Experiment B and the Original Standardized Dataset

- **Health**: very small negative change in standard deviation (-0.0525, values more concentrated around the mean), very small negative change in the mean value (-0.1475, cluster center shifted towards lower values)
- **Inflation**: medium positive change in standard deviation (0.4350, values more spread out around the mean), small positive change in the mean value (0.3538, cluster center shifted towards higher values)
- **Life Expectancy**: small negative change in standard deviation (-0.2682, values more concentrated around the mean), extreme negative change in the mean value (-1.2364, cluster center shifted towards lower values)
- **Total Fertility**: small negative change in standard deviation (-0.3128, values more concentrated around the mean), extreme positive change in the mean value (1.3220, cluster center shifted towards higher values)
- **GDPP**: very big negative change in standard deviation (-0.8430, values more concentrated around the mean), big negative change in the mean value (-0.6003, cluster center shifted towards lower values)

So this cluster is mostly characterized by countries with a little lower than average health spending with values a little more concentrated around the mean, higher than average inflation with values more spread out around the mean, a lot lower than average life expectancy with values a little more concentrated around the mean, a lot higher than average total fertility with values a little more concentrated around the mean and much lower than average GDPP with values much more concentrated around the mean.

Finally, we list all the countries belonging to this cluster:

```
% List the countries belonging to each cluster
countries_in_clusters_B = cell(3, 1);

for k = 1:3
    indices = find(clusters_k_medians_b_5D == k);
    countries_in_clusters_B{k} = country(indices);
end
```

Countries in Cluster 1 (Experiment B): 'Afghanistan', 'Angola', 'Benin', 'Botswana', 'Burkina Faso', 'Burundi', 'Cameroon', 'Central African Republic', 'Chad', 'Comoros', 'Congo Dem Rep', 'Congo Rep', 'Cote d"Ivoire', 'Equatorial Guinea', 'Eritrea', 'Gabon', 'Gambia', 'Ghana', 'Guinea', 'Guinea-Bissau', 'Haiti', 'Iraq', 'Kenya', 'Kiribati', 'Lao', 'Lesotho', 'Liberia', 'Madagascar', 'Malawi', 'Mali', 'Mauritania', 'Mozambique', 'Namibia', 'Niger', 'Nigeria', 'Pakistan', 'Rwanda', 'Samoa', 'Senegal', 'Sierra Leone', 'Solomon Islands', 'South Africa', 'Sudan', 'Tanzania', 'Timor-Leste', 'Togo', 'Uganda', 'Vanuatu', 'Yemen', 'Zambia'

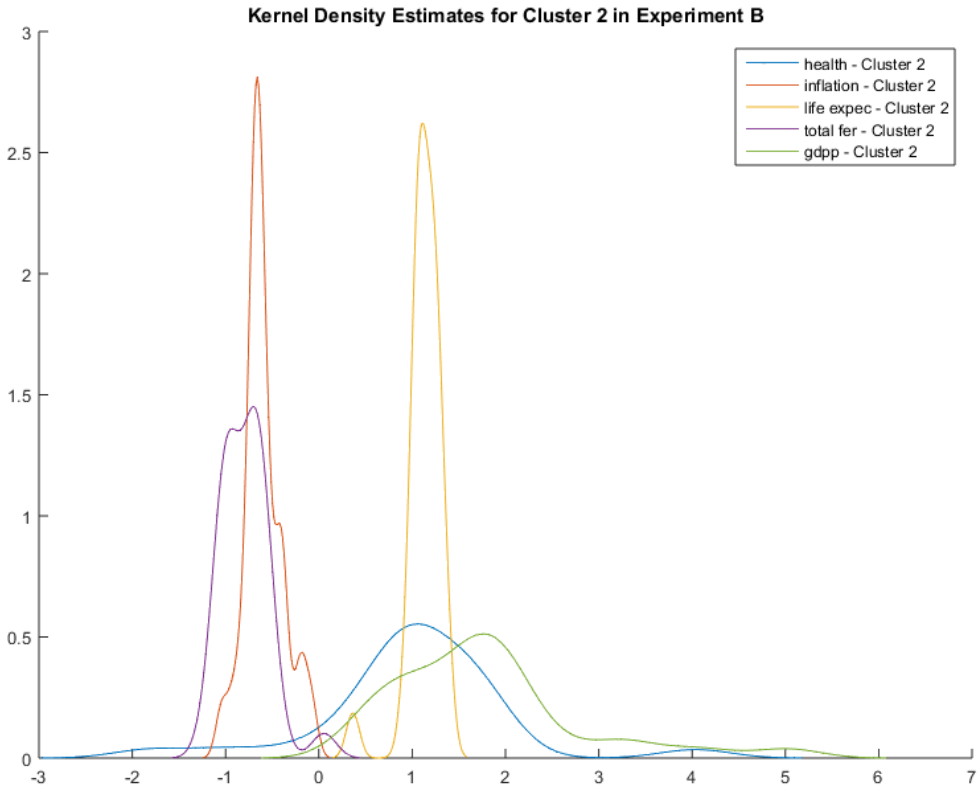
Cluster 2:

We calculate the descriptive statistics for each feature of the cluster:

	health	**inflation**	**life expec**	**total fer**	**gdpp**
min	-1.8223	-1.0408	0.3648	-1.1877	0.4439
max	4.0353	-0.0759	1.3768	0.0542	5.0214
mean	1.0048	-0.5920	1.1276	-0.7867	1.7839
median	0.9900	-0.6396	1.1294	-0.7418	1.7178
variance	0.9929	0.0436	0.0339	0.0609	1.0606
std	0.9964	0.2087	0.1842	0.2468	1.0298
25%	0.6969	-0.7060	1.0620	-0.9961	0.9731
50%	0.9900	-0.6396	1.1294	-0.7418	1.7178
75%	1.5233	-0.4647	1.2306	-0.6592	2.0370

Table 16: Statistics on Cluster 2 of Experiment B

We then plot the probability density functions (PDFs) for each feature in the cluster:



We then compare the standard deviations and mean values for each feature of the cluster to the original standardized dataset:

	health	**inflation**	**life expec**	**total fer**	**gdpp**
std	-0.0036	-0.7913	-0.8158	-0.7532	0.0298
mean	1.0048	-0.5920	1.1276	-0.7867	1.7839

Table 17: Standard Deviation and Mean Value Differences between Cluster 2 of Experiment B and the Original Standardized Dataset

- **Health**: very small negative change in standard deviation (-0.0036, values more concentrated around the mean), extreme positive change in the mean value (1.0048, cluster center shifted towards higher values)
- **Inflation**: big negative change in standard deviation (-0.7913, values more concentrated around the mean), medium negative change in the mean value (-0.5920, cluster center shifted towards lower values)
- **Life Expectancy**: very big negative change in standard deviation (-0.8158, values more concentrated around the mean), extreme positive change in the mean value (1.1276, cluster center shifted towards higher values)
- **Total Fertility**: big negative change in standard deviation (-0.7532, values more concentrated around the mean), big negative change in the mean value (-0.7867, cluster center shifted towards lower values)
- **GDPP**: very small positive change in standard deviation (0.0298, values more spread out around the mean), extreme positive change in the mean value (1.7839, cluster center shifted towards higher values)

So this cluster is mostly characterized by countries with a lot higher than average health spending with values a little more concentrated around the mean, lower than average inflation with values much more concentrated around the mean, a lot higher than average life expectancy with values a lot more concentrated around the mean, much lower than average total fertility with values much more concentrated around the mean and a lot higher than average GDPP with values a little more spread out around the mean.

Finally, we list all the countries belonging to this cluster:

Countries in Cluster 2 (Experiment B): 'Australia', 'Austria', 'Bahamas', 'Belgium', 'Canada', 'Cyprus', 'Denmark', 'Finland', 'France', 'Germany', 'Greece', 'Iceland', 'Ireland', 'Israel', 'Italy', 'Japan', 'Luxembourg', 'Malta', 'Netherlands', 'New Zealand', 'Norway', 'Portugal', 'Qatar', 'Singapore', 'Slovenia', 'Spain', 'Sweden', 'Switzerland', 'United Kingdom', 'United States'

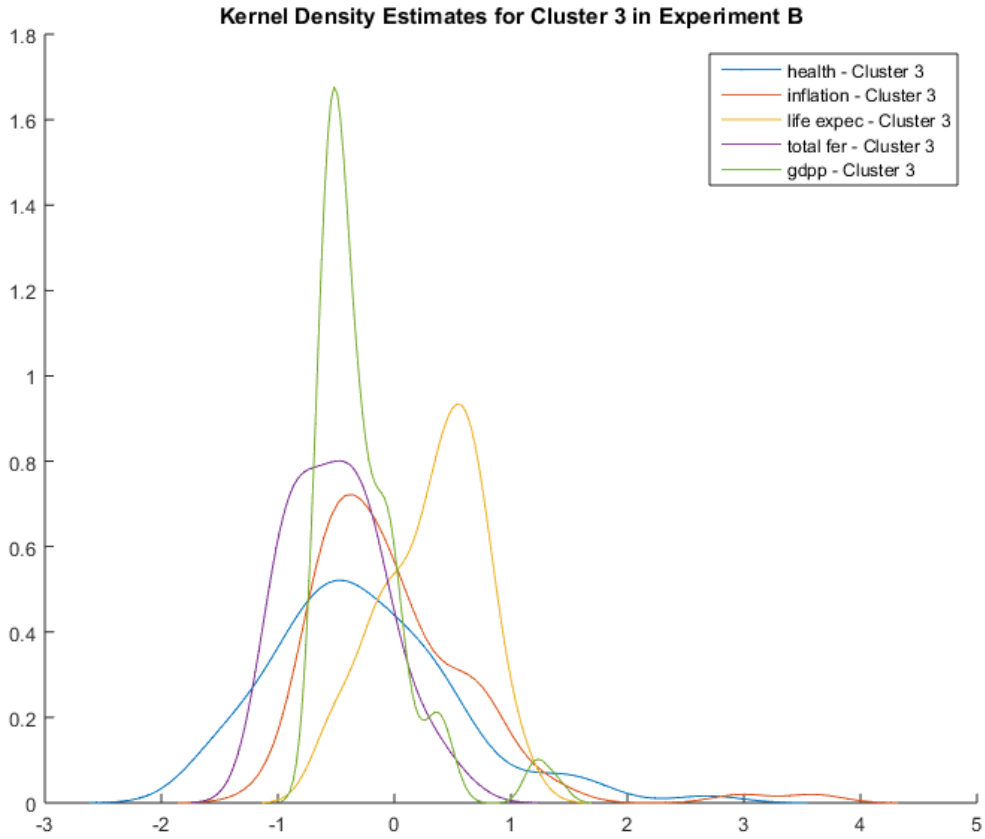
Cluster 3:

We calculate the descriptive statistics for each feature of the cluster:

	health	**inflation**	**life expec**	**total fer**	**gdpp**
min	-1.7641	-1.1344	-0.5910	-1.1348	-0.6750
max	2.6883	3.6060	1.1070	0.6355	1.3932
mean	-0.2617	0.0008	0.3218	-0.4885	-0.2701
median	-0.3443	-0.1695	0.3985	-0.5139	-0.4100
variance	0.6498	0.5472	0.1766	0.1674	0.1652
std	0.8061	0.7398	0.4203	0.4091	0.4064
25%	-0.7711	-0.5061	-0.0175	-0.8508	-0.5435
50%	-0.3443	-0.1695	0.3985	-0.5139	-0.4100
75%	0.1654	0.3210	0.6544	-0.2183	-0.1017

Table 18: Statistics on Cluster 3 of Experiment B

We then plot the probability density functions (PDFs) for each feature in the cluster:



We then compare the standard deviations and mean values for each feature of the cluster to the original standardized dataset:

	health	**inflation**	**life expec**	**total fer**	**gdpp**
std	-0.1939	-0.2602	-0.5797	-0.5909	-0.5936
mean	-0.2617	0.0008	0.3218	-0.4885	-0.2701

Table 19: Standard Deviation and Mean Value Differences between Cluster 3 of Experiment A and the Original Standardized Dataset

- **Health**: very small negative change in standard deviation (-0.1939, values more concentrated around the mean), small negative change in the mean value (-0.2617, cluster center shifted towards lower values)
- **Inflation**: small negative change in standard deviation (-0.2602, values more concentrated around the mean), very small positive change in the mean value (0.0008, cluster center shifted towards higher values)
- **Life Expectancy**: medium negative change in standard deviation (-0.5797, values more concentrated around the mean), small positive change in the mean value (0.3218, cluster center shifted towards higher values)
- **Total Fertility**: medium negative change in standard deviation (-0.5909, values more concentrated around the mean), medium negative change in the mean value (-0.4885, cluster center shifted towards lower values)
- **GDPP**: medium negative change in standard deviation (-0.5936, values more concentrated around the mean), small negative change in the mean value (-0.2701, cluster center shifted towards lower values)

So this cluster is mostly characterized by countries with a little lower than average health spending with values a little more concentrated around the mean, a little higher than average inflation with values a little more concentrated around the mean, a little higher than average life expectancy with values more concentrated around the mean, lower than average total fertility with values more concentrated around the mean and a little lower than average GDPP with values more concentrated around the mean.

Finally, we list all the countries belonging to this cluster:

Countries in Cluster 3 (Experiment B): 'Albania', 'Algeria', 'Antigua and Barbuda', 'Argentina', 'Armenia', 'Azerbaijan', 'Bahrain', 'Bangladesh', 'Barbados', 'Belarus', 'Belize', 'Bhutan', 'Bolivia', 'Bosnia and Herzegovina', 'Brazil', 'Brunei', 'Bulgaria', 'Cambodia', 'Cape Verde', 'Chile', 'China', 'Colombia', 'Costa Rica', 'Croatia', 'Czech Republic', 'Dominican Republic', 'Ecuador', 'Egypt', 'El Salvador', 'Estonia', 'Fiji', 'FYROM', 'Georgia', 'Grenada', 'Guatemala', 'Guyana', 'Hungary', 'India', 'Indonesia', 'Iran', 'Jamaica', 'Jordan', 'Kazakhstan', 'Kuwait', 'Kyrgyz Republic', 'Latvia', 'Lebanon', 'Libya', 'Lithuania', 'Malaysia', 'Maldives', 'Mauritius', 'Micronesia Fed Sts', 'Moldova', 'Mongolia', 'Montenegro', 'Morocco', 'Myanmar', 'Nepal', 'Oman', 'Panama', 'Paraguay', 'Peru', 'Philippines', 'Poland', 'Romania', 'Russia', 'Saudi Arabia', 'Serbia', 'Seychelles', 'Slovak Republic', 'South Korea', 'Sri Lanka', 'St. Vincent and the Grenadines', 'Suriname', 'Tajikistan', 'Thailand', 'Tonga', 'Tunisia', 'Turkey', 'Turkmenistan', 'Ukraine', 'United Arab Emirates', 'Uruguay', 'Uzbekistan', 'Venezuela', 'Vietnam'

Overall interpretation:

- **Cluster 1** seems to represent countries with challenging economic and health conditions, characterized by lower health expenditure, higher inflation, lower life expectancy, higher fertility rates, and lower GDP per capita (least developed countries)
- **Cluster 2** seems to represent countries with better economic and health conditions, with higher health expenditure, lower and stable inflation, higher life expectancy, lower fertility rates, and higher GDP per capita (developed countries)
- **Cluster 3** seems to represent a mix of countries, possibly including emerging economies, with modest health expenditure, average inflation, slightly better life expectancy, lower fertility rates, and modest GDP per capita (developing countries)

6.3 Conclusion

Experiment A focuses on a limited set of variables: Inflation, Life Expectancy, Total Fertility, and GDPP. Its simplicity allows for straightforward interpretation of basic economic and demographic patterns but omits critical dimensions like health expenditure, a key factor in understanding a nation's socio-economic fabric. As a result, while informative in economic and demographic contexts, Experiment A's clusters don't provide a comprehensive view of a country's overall development.

In contrast, Experiment B incorporates Health expenditure into its analysis, alongside Inflation, Life Expectancy, Total Fertility, and GDPP. This broader approach weaves in a crucial aspect of socio-economic development, resulting in richer and more nuanced categorization of countries. It accurately mirrors real-world complexities where health is intertwined with economic and demographic indicators, offering more realistic and actionable insights. This methodology significantly enhances our socio-economic analysis, providing more accurate insights for policymakers and researchers. Experiment B's comprehensive approach captures the complex interplay between health, economic, and demographic factors, making it a superior tool for socio-economic planning and evaluation, and offering a fuller understanding of a nation's socio-economic status.

References

[1] Britannica: gross domestic product. https://www.britannica.com/money/topic/gross-domestic-product.

[2] Economic transformation and progress towards the sdgs through trade. https://sdgpulse.unctad.org/trade-developing-economies/.

[3] The global south's long-term challenge of sustainable healthcare in the lmics' health systems. https://www.biomedcentral.com/collections/GSSH.

[4] Human development index (hdi). https://hdr.undp.org/data-center/human-development-index#/indicies/HDI.

[5] Inflation rate formula: How to calculate inflation rate using gdp deflator? https://www.marca.com/en/lifestyle/us-news/personal-finance/2022/11/01/63618eb8268e3ebb788b45e5.html.

[6] Ocd library: Life expectancy. https://www.oecd-ilibrary.org/docserver/pension{_}glance-2017-21-en.pdf.

[7] Relationship between income and health. https://www.health.org.uk/evidence-hub/money-and-resources/income/relationship-between-income-and-health.

[8] The use of the concept "global south" in social science and humanities. https://www.academia.edu/7917466.

[9] World bank country and lending groups. https://datahelpdesk.worldbank.org/knowledgebase/articles/906519-world-bank-country-and-lending-groups#:~:text=For%20the%20current%202024%20fiscal,those%20with%20a%20GNI%20per.

[10] World health organization: Child mortality and causes of death. https://www.who.int/data/gho/data/themes/topics/topic-details/GHO/child-mortality-and-causes-of-death/.

[11] Ian T. Jolliffe and Jorge Cadima. Principal component analysis: a review and recent developments. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 374(2065):20150202, April 2016.

[12] David Irwin Barath Raghavan Ville Satopaa, Jeannie Albrecht. Finding a kneedle in a haystack: Detecting knee points in system behavior. *Distributed Computing Systems Workshops (ICDCSW)*, July 2011.