Assignment 2

Michael Darmanis (7115152200004)
mdarm@di.uoa.gr

Vasilios Venieris (7115152200017)
vvenieris@di.uoa.gr

January 24, 2023

1 Introduction

The main goal of this report[1] is to provide recommendations to the CEO of HELP International on how to allocate the organisation's resources in a strategic and effective manner, based on the identification of countries in need of development aid. This will be achieved through the implementation of a clustering analysis on a country data set using MATLAB®. The analysis will consist of several stages, including the exploration of the data's characteristics and patterns, the selection and transformation of relevant features, the selection of appropriate clustering algorithms, the execution of these algorithms, and the characterization of the resulting clusters.

The initial stage of the analysis involves the examination of the individual features of the data, including their data type and range of values, as well as the distribution of these values. The relationships between different features will also be considered. Following this, relevant features will be selected and transformed in order to make their values comparable. Based on the characteristics of the data and the desired properties of the clusters, the appropriate clustering algorithms will then be chosen. These algorithms will then be executed with different parameter-values in order to identify persistent clusters. Finally, the clusters will be characterised based on the values of the features within each cluster.

2 "Feeling the data"

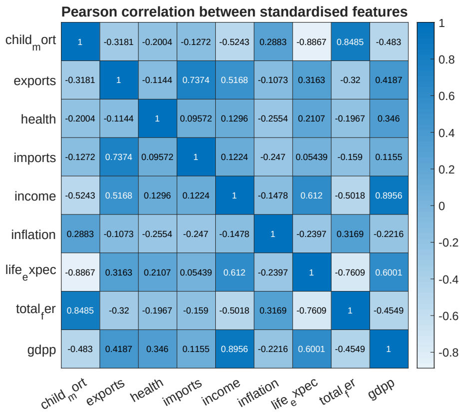
In this part of the analysis, characteristics of each individual feature in the dataset were examined in order to gain a better understanding of its nature. This involved determining the data type and range of values for each feature, as well as generating histograms to visualize the distribution of values. The mean and standard deviation were also calculated for each feature.

Feature	Type	Range	Mean	Std Dev
child_mort	float	[2.6000, 208.0000]	38.2701	40.3289
exports	float	[0.1090, 200.0000]	41.1090	27.4120
health	float	[1.8100, 17.9000]	6.8157	2.7468
imports	float	[0.0659, 174.0000]	46.8902	24.2096
income	float	[609.0000, 125000.0000]	17144.6886	19278.0677
inflation	float	[-4.2100, 104.0000]	7.7818	10.5707
life_expec	float	[32.1000, 82.8000]	70.5557	8.8932
total_fer	float	[1.1500, 7.4900]	2.9480	1.5138
gdpp	float	[231.0000, 105000.0000]	12964.1557	18328.7048

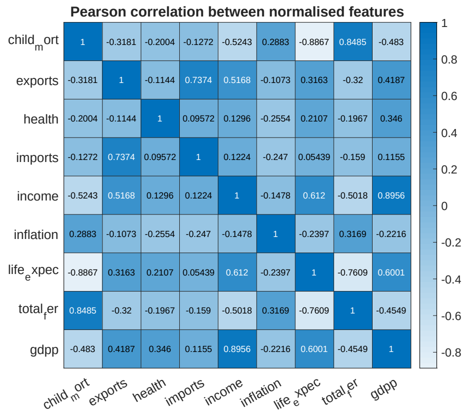
The linear dependence between each feature and all the others was calculated using the Pearson correlation coefficient in order to gain insight into the relationships between different features in the data set. Additionally, standard score normalization and min-max feature scaling normalization were performed. The linear dependence between the transformed features were calculated in each case.

It is evident in Figure 1 that linear relationships between features are not affected by the transfor-

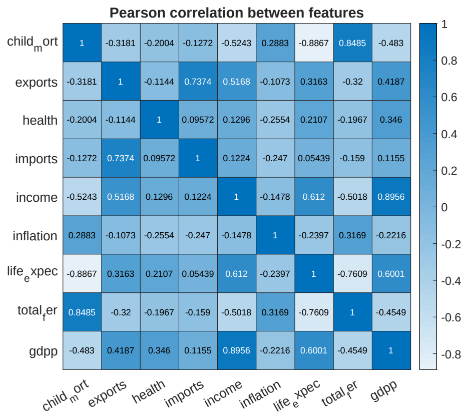
[1] as stated at https://www.kaggle.com/rohan0301/unsupervised-learning-on-country-data



(a) Standardised features.



(b) Min-max normalised features.



(c) Raw features.

Figure 1: Correlation matrices between features.

mations. Pearson's correlation measures the linear component of association so it comes to no surprise that linear transformations of data (like mim-max normalisation and standardisation) did not affect the correlations between the features.

In Figure 2 it can readily be observed that the feature `life_expec` exhibits a negative skew in its distribution, while the `health` exhibits a normal distribution. On the other hand, all other measured quantities show a positive skew in their distributions. It should be noted that the distribution of the categorical feature "country", which consists solely of text data and exhibits the same number of unique values as the total length of the data set, was not analysed since it makes little sense to do so.

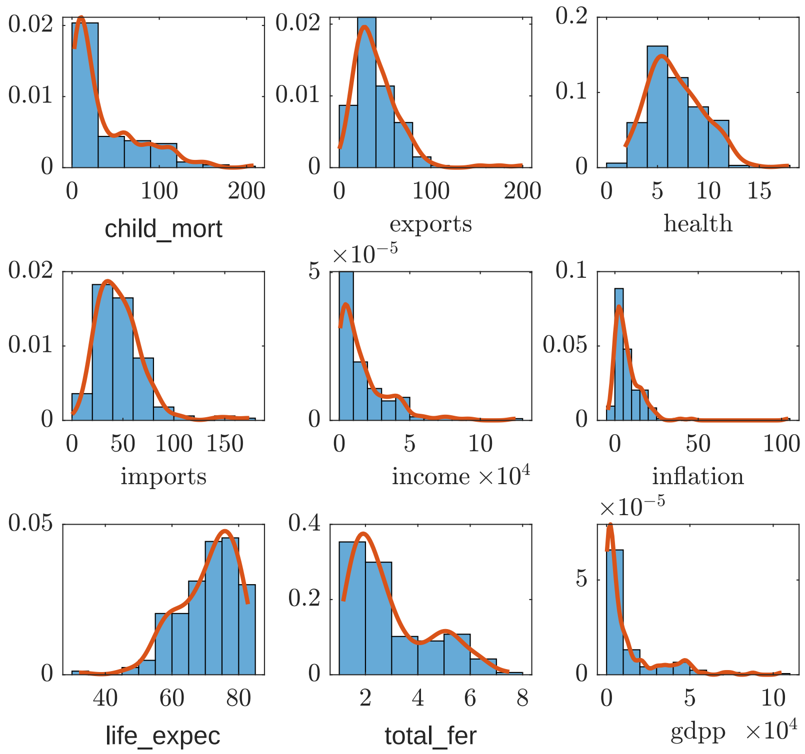


Figure 2: Histograms and distributions of features

The analysis of the dataset also reveals several relationships between features. The relationship between child mortality and economic conditions is of particular significance, as the data indicates that child mortality tends to increase as income, gross domestic product (GDP), and exports decrease. Inflation also appears to have a negative impact on child mortality. This suggests that economic factors, such as income and GDP, may be influential in determining child mortality rates. The relationship between exports and other economic indicators is also noteworthy, as an increase in exports tends to lead to an increase in GDP, income, and imports, implying that exports may be a key contributor to economic growth.

In addition, the data suggests that spending on health has a positive effect on life expectancy and a negative effect on child mortality. Higher levels of income and GDP are correlated with higher life expectancy and lower child mortality, indicating a possible relationship between these factors and spending on health. Furthermore, high levels of inflation appear to be detrimental to economic conditions, as they have a negative effect on various economic indicators, including income, GDP, and total fertility rate. Finally, the data suggests that higher life expectancy is correlated with lower total fertility rates, a relationship that may be influenced by factors such as GDP and spending on health.

3 Feature selection/transformation

Based on the aformentioned data relationships, it is clear that some features are closely related to specific categories, namely: health, trade, and finance. Therefore, features will be grouped up into these categories and then normalised using a min-max scheme. The three categories of features in the dataset are: health (child mortality, health, life expectancy, total fertility rate), trade (imports, exports), and finance (income, inflation, GDP).

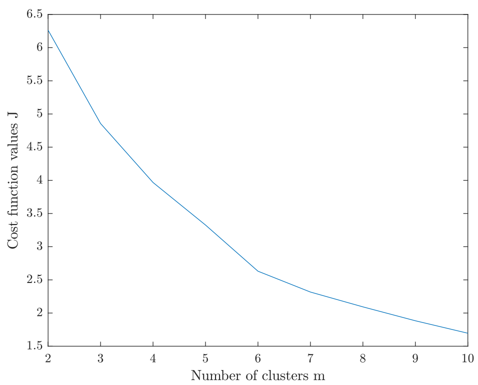
By combining multiple features into fewer ones facilitates the comprehension of relationships between different features and provides a more intuitive understanding of the data. It also captures broader or more general relationships in the data, and it may be possible to improve the generalisability of the clustering results to new, unseen data.

Additionally, by creating new features that capture more relevant or subtle relationships in the data, it may be possible to improve the performance of the clustering algorithm itself.

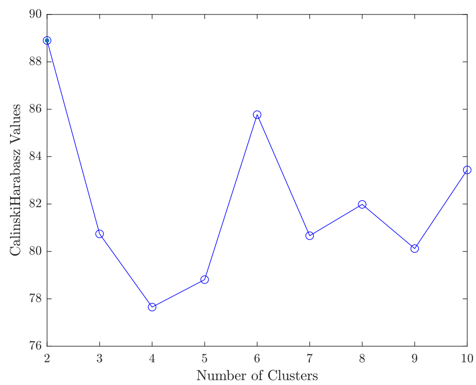
4 Selection and execution of the clustering algorithms

After normalising and combining & renormalising the data, the k-means clustering algorithm was used for performing the clustering analysis in order to effectively group countries in terms of their needs. The k-means algorithm is a popular choice for clustering due to its simplicity and efficiency, making it well-suited for large datasets (not that ours is, at this point anyway). Additionally, the algorithm produces clear and distinct clusters, which can be beneficial when attempting to partition the countries into groups. Overall, the use of k-means clustering in this context can provide valuable insights and aid in the strategic allocation of resources.

Determining the appropriate number of initial clusters in a k-means clustering operation can be achieved through the use of various measures such as the "elbow method" or the variance-ratio criterion[1]. The "elbow method" for approximating the correct value of k is to run the algorithm for increasing values of k, until there is a minimal decrease in the chosen measure of cluster cohesion (the cost function in our case) between two values[5]. Alternative measures such as the Bayesian Information Criterion and Gap statistics are also suggested as preferable options[7]. It is important to note that the "elbow method", which involves identifying a point of inflection in a plot of the measure of cluster cohesion versus k, has been criticized in the literature [6, 4] and should, generally speaking, be avoided.



(a) "Elbow" method.



(b) Variance-ratio criterion.

Figure 3: Picking the right number of initial clusters k.

Both the "elbow method" (see Figure 3a) and the variance-ratio criterion (see Figure 3b) agreed that the hidden structure underlying the data was expressed in three clusters. The the k-means algorithm was initialised using three clusters.

In order to pick initial points that have a good chance of being in different clusters, k-means centroids were randomly initialized from Gaussian noise of the data as described in Coates et al. [2]. The algorithm was then applied 100 times to the dataset, and the results with the smallest cost function were selected. The resulting clusters are shown in Figure 4.

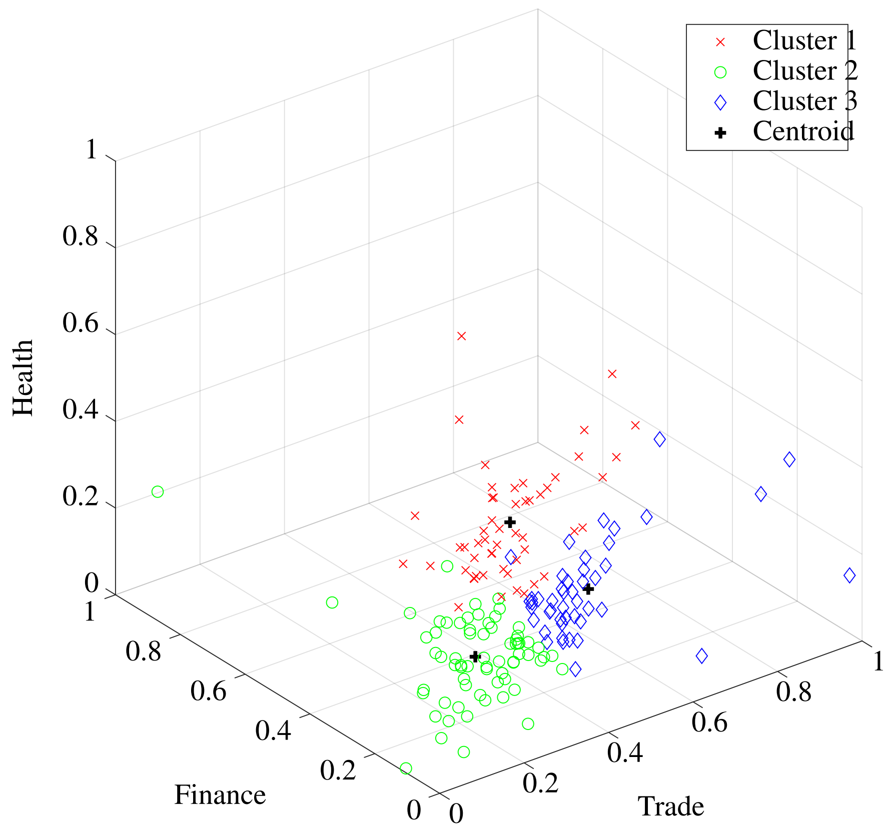


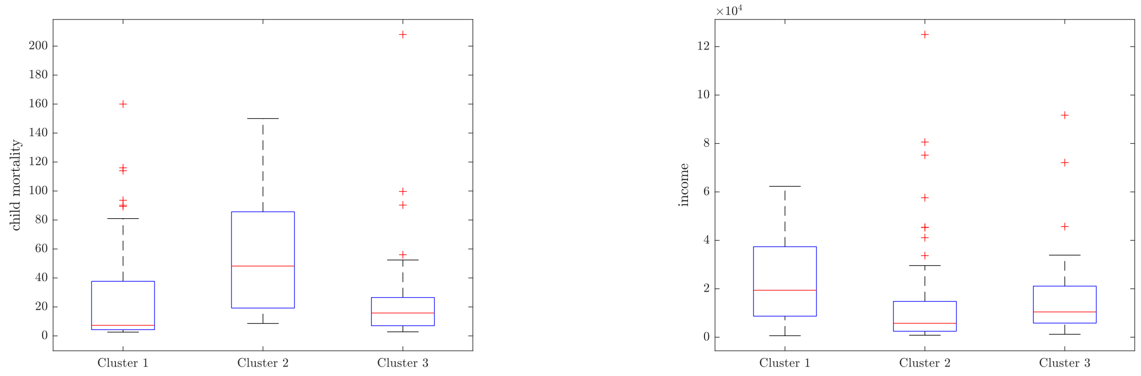
Figure 4: Clusters

It is important to note that running the algorithm multiple times helps to make the k-means more tolerant to outliers and results in the final centroid leaning towards the denser are of points. If that were not the case, then the algorithm would have been sensitive to outliers, and a variant such as k-medians ought have been used instead as described in Theodoridis and Koutroumbas [8].

5 Characterisation of clusters

As seen in Section 2, a strong correlation exists between low income and high child mortality. This relationship is widely acknowledged within the fields of economics and public health (see Appendix A). Low income is frequently considered an indicator of economic underdevelopment, and high child mortality reflects the overall health and well-being of a population.

Based on the strong correlation between low income and high child mortality, it is safe to assume that countries with low income and high child mortality rates are likely to have less developed economies and weaker healthcare systems, thereby being in need of funding. Figure 5 showcases the income and child mortality rates with respect to labelled clusters. By examining Figures 5a and 5b, we can make an initial assumption and say that the countries belonging to Cluster 1 are the ones in more need of development aid.



(a) Child mortality. (b) Net income.

Figure 5: Death of children under 5 years of age per 1000 live births and net income per person of all the countries within each Cluster.

By examining the countries within Cluster 1, we can observe that even developed countries such as the United Kingdom are included in this cluster. This suggests that the clustering process and feature engineering[2] should be re-evaluated in order to accurately identify which countries are in need of funding.

```
       Cluster 1           Cluster 2              Cluster 3
      ----------          ----------             ----------
     Afghanistan              Angola                  Benin
    Burkina Faso             Burundi                Albania
         Algeria Antigua and Barbuda              Argentina
         Armenia           Australia                Austria
         Bahrain             Belgium                 Brunei
               .                   .                      .
               .                   .                      .
               .                   .                      .
        Tanzania         Timor-Leste                   Togo
          Uganda              Zambia                Uruguay
      Uzbekistan             Vanuatu                Vietnam
           Yemen         Switzerland United Arab Emirates
  United Kingdom       United States              Venezuela
```

Prima facie evidence suggests that there could be a case for further investigation, to establish whether or not further clustering analysis should be put in hand. Nevertheless, it should be stressed that, an unsupervised analysis is limited and relevant facts could be difficult to establish with any degree of certainty.

References

[1] Tadeusz Calinski. "A dendrite method for cluster analysis". In: *Communication in statistics* 3 (1974), pp. 1–27.

[2] Adam Coates and Andrew Y Ng. "Learning feature representations with k-means". In: *Neural networks: Tricks of the trade*. Springer, 2012, pp. 561–580. URL: https://link.springer.com/chapter/10.1007/978-3-642-35289-8_30.

[2] an alternative is to discard highly correlated data altogether and keep only low-valued correlations; such a case is seen in [3]

[3] Tanmay Deshpande. *Clustering: PCA | K-Means - DBSCAN - Hierarchical |*. 2022. URL: https://www.kaggle.com/code/tanmay111999/clustering-pca-k-means-dbscan-hierarchical/notebook.

[4] David J Ketchen and Christopher L Shook. "The application of cluster analysis in strategic management research: an analysis and critique". In: *Strategic management journal* 17.6 (1996), pp. 441–458.

[5] Jure Leskovec, Anand Rajaraman, and Jeffrey David Ullman. "Clustering". In: *Mining of massive data sets*. Cambridge university press, 2020. Chap. 7, pp. 240–280. URL: http://mmds.org/#ver30.

[6] Glenn W Milligan and Martha C Cooper. "An examination of procedures for determining the number of clusters in a data set". In: *Psychometrika* 50.2 (1985), pp. 159–179.

[7] Erich Schubert. "Stop using the elbow criterion for k-means and how to choose the number of clusters instead". In: *arXiv preprint arXiv:2212.12189* (2022).

[8] Sergios Theodoridis et al. "Clustering". In: *Introduction to Pattern Recognition: A Matlab Approach*. Academic Press, 2010. Chap. 7, pp. 159–208.

Appendix A

According to the World Bank, low income is defined as a gross national income (GNI) per capita of less than $1,035 per year, while high child mortality is defined as a mortality rate of more than 43 deaths per 1,000 live births. Countries with low income and high child mortality rates tend to have less developed economies and weaker healthcare systems, which can contribute to higher mortality rates among children.

Many factors can contribute to low income and high child mortality in a country, including poverty, lack of access to education and healthcare, lack of infrastructure and resources, and political instability. Developing countries often face these challenges to a greater extent than developed countries, which can lead to higher rates of child mortality and lower levels of economic development.

Several sources discuss the relationship between low income and high child mortality, including:

- The World Health Organization (WHO) states that "poverty is the single greatest threat to child survival" and that "most of the 10.6 million deaths among children under five occur in developing countries." (https://www.who.int/child_adolescent_health/topics/poverty/en/)
- The World Bank's World Development Indicators database provides data on income levels and child mortality rates for countries around the world. (https://data.worldbank.org/indicator/SP.DYN.IMRT.IN)
- The United Nations Children's Fund (UNICEF) reports that "more than half of all child deaths occur in just five countries – India, Nigeria, Pakistan, Democratic Republic of the Congo and China – with India alone accounting for about one third." (https://www.unicef.org/sowc2014/numbers/)

Appendix B

Listing 1: Clustering analysis script

```
%% Clustering Algorithms, Homework 2
%  Clustering analysis on country data in order to determine which
%  group of countries are in need of financial aid.
%
%  This script makes use of the following provided functions:
%
```

```
%     rand_data_init.m
%     k_means.m
%
%  Two further functions, written by the authors, were also used
%  for editting plot variables, namely:
%
%     PlotDimensions.m
%     ChangeInterpreter.m
%

%% Initialisation
clear; close all; clc

%% ================= Part 1: Feeling the data ========================

% Import the CSV file
data = readtable('Country-data.csv');

% Extract the labels from the first column
countryNames = data{:, 1};

% Extract the column names and keep only feature labels
featureNames = data.Properties.VariableNames;
featureNames = featureNames(1, 2:end);

% Convert the table-data to an array
data = table2array(data(:, 2:end));

% Determine the dimensions of the data set
[numRows, numCols] = size(data);

% Preallocate cell array for feature type
featureType = cell(1, 9);

% Print a header row for the table
fprintf('%-13s %-9.5s %-30s %-11s %-11s\n', 'Feature',...
    'Type', 'Range', 'Mean', 'Std Dev');
fprintf('%-13s %-8s  %-30s %-11s %-11s\n',...
    '------------', '---------',...
    '------------------------------', '-----------', '-----------');

% Create a grid of subplots, with one subplot for each feature
figure(1);
subplot(3, 3, 1:numCols);

for i = 1:numCols
    uniqueVal = unique(data(:, i));
    if isstring(uniqueVal)
        featureType(1, i) = {'categorical'};
    elseif isinteger(uniqueVal)
        featureType(1, i) = {'integer'};
```

```
    elseif isfloat(uniqueVal)
        featureType(1, i) = {'float'};
    end

    % Determine the range of values, mean, and standard deviation
    % for the current column
    minVal  = min(data(:, i));
    maxVal  = max(data(:, i));
    meanVal = mean(data(:, i));
    stdVal  = std(data(:, i));

    % Print a row for the current column
    fprintf(' %-11s  %-8s  [%12.4f, %12.4f] %12.4f %12.4f\n',...
        featureNames{i}, featureType{i}, minVal, maxVal,...
        meanVal, stdVal);

    % Select the subplot for the current feature
    subplot(3, 3, i);

    % Extract the data for the current feature
    featureData = data(:, i);

    % Create a histogram for the current feature
    histogram(featureData, 'Normalization', 'pdf');

    % Allow the distribution plot to be superimposed on the
       histogram
    hold on;
    x = linspace(min(featureData), max(featureData), 100);
    y = ksdensity(featureData, x);
    plot(x, y, 'LineWidth', 2);
    xlabel(featureNames(i), 'Interpreter', 'none');

    % Reset the hold state
    hold off;
end
PlotDimensions(figure(1), 'centimeters', [15.747, 14], 12)
ChangeInterpreter(figure(1), 'latex')

% Plot image in pdf format
h = figure(1);
set(h,'Units','Inches');
pos = get(h,'Position');
set(h, 'PaperPositionMode', 'Auto', 'PaperUnits', 'Inches',...
    'PaperSize', [pos(3), pos(4)])
print(h, 'histogram', '-dpdf', '-r0')

% Calculate the Pearson correlation coefficient between
% each pair of features
corrMatrix = corr(data);

% Create a heatmap of the correlation matrix
```

```
figure(2);
heatmap(featureNames, featureNames, corrMatrix);
title('Pearson correlation between features')
PlotDimensions(figure(2), 'centimeters', [15.747, 14], 12)
ChangeInterpreter(figure(2), 'latex')

% Plot image in pdf format
h = figure(2);
set(h,'Units','Inches');
pos = get(h,'Position');
set(h, 'PaperPositionMode', 'Auto', 'PaperUnits', 'Inches',...
    'PaperSize', [pos(3), pos(4)])
print(h, 'corr1', '-dpdf', '-r0')

% Calculate the mean and standard deviation of each column
meanVals = mean(data, 1);
stdDevs = std(data, 0, 1);

% Perform standard score normalization on each feature
standardizedData = (data - meanVals) ./ stdDevs;

% Calculate the Pearson correlation coefficient between
% each pair of standardised features
standardizedCorrMatrix = corr(standardizedData);

% Create a heatmap of the standardised correlation matrix
figure(3);
heatmap(featureNames, featureNames, standardizedCorrMatrix);
title('Pearson correlation between standardised features')
PlotDimensions(figure(3), 'centimeters', [15.747, 14], 12)
ChangeInterpreter(figure(3), 'latex')

% Plot image in pdf format
h = figure(3);
set(h,'Units','Inches');
pos = get(h,'Position');
set(h, 'PaperPositionMode', 'Auto', 'PaperUnits', 'Inches',...
    'PaperSize', [pos(3), pos(4)])
print(h, 'corr2', '-dpdf', '-r0')

% Find the minimum and maximum values in each column of the data
minVals = min(data);
maxVals = max(data);

% Normalize the data using max-min normalization
normalisedData = (data - minVals) ./ (maxVals - minVals);

% Calculate the Pearson correlation coefficient between
% each pair of normalised features
minMaxCorrMatrix = corr(normalisedData);

% Create a heatmap of the min-max correlation matrix
```

```
figure(4);
heatmap(featureNames, featureNames, minMaxCorrMatrix);
title('Pearson correlation between normalised features')
PlotDimensions(figure(4), 'centimeters', [15.747, 14], 12)
ChangeInterpreter(figure(4), 'latex')

% Plot image in pdf format
h = figure(4);
set(h, 'Units', 'Inches');
pos = get(h, 'Position');
set(h, 'PaperPositionMode', 'Auto', 'PaperUnits', 'Inches',...
    'PaperSize', [pos(3), pos(4)])
print(h, 'corr3', '-dpdf', '-r0')

%% ================= Part 2: Feature selection
   ========================

% Normalize the columns by their mean values
%dataNorm = data ./ mean(data, 1);

% Create the first new feature by adding the normalized
% columns 2 and 4
%trade = dataNorm(:, 2) + dataNorm(:, 4);
trade = data(:, 4);
% Create the second new feature by adding the normalized
% columns 5, 6, and 9
%finance = dataNorm(:, 5) + dataNorm(:, 6) + dataNorm(:, 9);
finance = data(:, 6);
% Create the third new feature by concatenating the remaining
   columns
%health = dataNorm(:, 1) + dataNorm(:, 3) + dataNorm(:, 7)...
%    + dataNorm(:, 8);
health = data(:, 3);
% Concatenate the new features
newFeatures = [trade finance health];

% Normalise the new features using max-min normalization
dataFinal = (newFeatures - min(newFeatures)) ./...
    (max(newFeatures) - min(newFeatures));


%% ===== Part 3: Selection and execution of clustering algorithms
   ======

% Transpose dataFinal for input to k_means function
dataFinal = dataFinal';

% Set number of runs and range of values for m
nRuns = 40;
mMin = 2;
mMax = 10;

```

```
% Preallocate array to store results
jM = zeros(1, mMax - mMin + 1);

% Loop over values of m
for m = mMin:mMax
    % Initialize temporary minimum value
    jTempMin = inf;

    % Loop over number of runs
    for t = 1:nRuns
        % Generate initial theta values using randDataInit function
        thetaIni = rand_data_init(dataFinal, m);

        % Run kMeans function and store results
        [theta, bel, j] = k_means(dataFinal, thetaIni);

        % Update temporary minimum value if necessary
        if jTempMin > j
            jTempMin = j;
        end
    end

    % Append minimum value to jM array
    jM(m - mMin + 1) = jTempMin;
end

% Define m values for plot
m = mMin:mMax;

% Create figure and plot jM versus m
figure(5), plot(m, jM);
xlabel("Number of clusters m");
ylabel("Cost function values J");
ChangeInterpreter(figure(5), 'latex')

% Plot image in pdf format
h = figure(5);
set(h,'Units','Inches');
pos = get(h,'Position');
set(h, 'PaperPositionMode', 'Auto', 'PaperUnits', 'Inches',...
    'PaperSize', [pos(3), pos(4)])
print(h, 'elbow', '-dpdf', '-r0')

evaluation = evalclusters(dataFinal', "kmeans",...
    "CalinskiHarabasz", "KList", 1:10);
figure(6), plot(evaluation)
ChangeInterpreter(figure(6), 'latex')

% Plot image in pdf format
h = figure(6);
set(h,'Units','Inches');
pos = get(h,'Position');
```

```
set(h, 'PaperPositionMode', 'Auto', 'PaperUnits', 'Inches',...
    'PaperSize', [pos(3), pos(4)])
print(h, 'eval', '-dpdf', '-r0')

% Set fixed value of m
m = 3;

% Preallocate theta and bel arrays
nRuns = 100;
theta = zeros(m, size(dataFinal, 1), nRuns);
bel = zeros(1, size(dataFinal, 2), nRuns);

% Initialize temporary minimum value
jTempMin = inf;

% Loop over number of runs
for t = 1:nRuns
    % Generate initial theta values using randDataInit function
    thetaIni = rand_data_init(dataFinal, m);

    % Run kMeans function and store results
    [theta(:, :, t), bel(:, :, t), j] = k_means(dataFinal, thetaIni)
       ;

    % Update temporary minimum value and corresponding
    % outputs if necessary
    if jTempMin > j
        jTempMin = j;
        thetaMin = theta(:, :, t);
        belMin = bel(:, :, t);
    end
end

% Plot the clusters
figure(7), plot3(dataFinal(1, belMin==1),...
    dataFinal(2, belMin==1), dataFinal(3, belMin==1),'rx',...
    dataFinal(1, belMin==2), dataFinal(2, belMin==2),...
    dataFinal(3, belMin==2),'go', dataFinal(1, belMin==3),...
    dataFinal(2, belMin==3), dataFinal(3, belMin==3),'bd');
hold on
plot3(thetaMin(1,:), thetaMin(2,:), thetaMin(3,:), 'k+', 'LineWidth'
   , 2)
xlabel("Trade")
ylabel("Finance")
zlabel("Health")
legend("Cluster 1", "Cluster 2", "Cluster 3", "Centroid",...
    'Location', 'NorthEast')
grid on
hold off

ChangeInterpreter(figure(7), 'latex')
PlotDimensions(figure(7), 'centimeters', [18, 18], 12)
```

```
Plot2LaTeX(figure(7), 'test')

%% ============= Part 4: Characterisation of clusters
    ================

cluster1 = countryNames(belMin == 1);
cluster2 = countryNames(belMin == 2);
cluster3 = countryNames(belMin == 3);

% Get the lengths of the cell arrays
len1 = length(cluster1);
len2 = length(cluster2);
len3 = length(cluster3);

% Sort the strings in each cell array in alphabetical order
cluster1 = sort(cluster1);
cluster2 = sort(cluster2);
cluster3 = sort(cluster3);

% Print the cluster labels
fprintf('%20s\t%20s\t%20s\n', 'Cluster 1', 'Cluster 2', 'Cluster 3')
    ;
fprintf('%20s\t%20s\t%20s\n', '----------', '----------',...
    '----------');

% Print the first 5 strings of each cell array
fprintf('%20s\t%20s\t%20s\n', cluster1{1:5}, ...
    cluster2{1:5}, cluster3{1:5});

% Print the dots if there are more than 10 strings
% in any of the cell arrays
if len1 > 10 || len2 > 10 || len3 > 10
    fprintf('%20s\t%20s\t%20s\n', '.', '.', '.');
    fprintf('%20s\t%20s\t%20s\n', '.', '.', '.');
    fprintf('%20s\t%20s\t%20s\n', '.', '.', '.');
end

% Print the last 5 strings of each cell array
fprintf('%20s\t%20s\t%20s\n', cluster1{max(1, len1-4):len1},...
    cluster2{max(1, len2-4):len2}, cluster3{max(1, len3-4):len3});

clusters = ["Cluster 1", "Cluster 2", "Cluster 3"];
clusterData1 = data(belMin' == 1, :);
clusterData2 = data(belMin' == 2, :);
clusterData3 = data(belMin' == 3, :);

% Concatenate row vectors and append data to grouping variables
childMortality = [clusterData1(:, 1)' clusterData2(:, 1)'...
    clusterData3(:, 1)'];
income = [clusterData1(:, 5)' clusterData2(:, 5)'...
    clusterData3(:, 5)'];
grp = [zeros(1, length(clusterData1(:, 1)')),...
```

```
    ones(1, length(clusterData2(:, 1)')),...
    2 * ones(1, length(clusterData3(:, 1)'))];

% Boxplot of child mortality for all clusters
figure(8), boxplot(childMortality, grp, 'Labels', clusters);
ylabel("child mortality")
ChangeInterpreter(figure(8), 'latex')

% Plot image in pdf format
h = figure(8);
set(h,'Units','Inches');
pos = get(h,'Position');
set(h, 'PaperPositionMode', 'Auto', 'PaperUnits', 'Inches',...
    'PaperSize', [pos(3), pos(4)])
print(h, 'childmortbox', '-dpdf', '-r0')

% Boxplot of income for all clusters
figure(9), boxplot(income, grp, 'Labels', clusters);
ylabel("income")
ChangeInterpreter(figure(9), 'latex')

% Plot image in pdf format
h = figure(9);
set(h,'Units','Inches');
pos = get(h,'Position');
set(h, 'PaperPositionMode', 'Auto', 'PaperUnits', 'Inches',...
    'PaperSize', [pos(3), pos(4)])
print(h, 'healthbox', '-dpdf', '-r0')
```

Listing 2: MATLAB® function for changing the interpreter of all objects within a figure.

```
function ChangeInterpreter(h, Interpreter)
% ChangeInterpreter() changes the interpreter of figure h.

    % Find all string type objects
    TexObj = findall(h, 'Type', 'Text');
    LegObj = findall(h, 'Type', 'Legend');
    AxeObj = findall(h, 'Type', 'Axes');
    ColObj = findall(h, 'Type', 'Colorbar');

    Obj = [TexObj; LegObj]; % Tex and Legend opbjects can be treated similarly
    n_Obj = length(Obj);
    for i = 1:n_Obj
        Obj(i).Interpreter = Interpreter;
    end

    Obj = [AxeObj; ColObj]; % Axes and ColorBar objects can be treated similarly
    n_Obj = length(Obj);
    for i = 1:n_Obj
        Obj(i).TickLabelInterpreter = Interpreter;
```

```
    end
end
```

Listing 3: MATLAB® function for configuring a figure's appearance options.

```
function PlotDimensions(h, Units, Plotsize, Fontsize)
% PlotDimensions() changes the string units, the fontsize
% and the unit size of the figure h.

    h.Units = Units; % measurement units
    h.Position(2) = (h.Position(2) - 8.5); % bottom-left corner of plot
    h.Position((3:4)) = Plotsize; % usually [15.747, 9]
    set(findall(h, '-property', 'FontSize'), 'FontSize', Fontsize); % fontsize
end
```