ERASMUS UNIVERSITY ROTTERDAM
Erasmus School of Economics

Bachelor Thesis Econometrics & Operations Research

Clustering European Countries Based on Energy Dependence

Author:
Rosalie Matla
449555

Supervisor:
C. Cavicchia
Second assessor:
prof. dr. P.H.B.F. Franses

July 3, 2020

Abstract

The aim of this thesis is to create homogeneous groups of the member states of the European Union based on their energy dependence levels and import dependence, in order to help shape long-term energy policies for the different created groups. K-means and K-harmonic means are implemented as Machine learning techniques for clustering the data. Ward's hierarchical clustering method combined with the squared Euclidean distance as well as the silhouette method are used to determine the optimal number of clusters, that needs to be pre-determined for both algorithms. In contradiction to previous research, we find that K-harmonic means is still sensitive to its centre initialization, however, it is less sensitive than K-means, which makes it a more robust clustering algorithm. As a result of the analysis, eight clusters are formed using K-means and thirteen clusters for K-harmonic means.

Contents

1 Introduction

The import and export of energy between countries in the European Union is very important. More than half of the fossil fuels (primary energy) consumed by the European Union are imported. The dependence on energy in Europe has significantly increased since 1990 (European Commission, 2018). Energy dependence shows how much a country depends on other countries to meet its needs. The aim is to analyse the energy dependence and resources of member states of the European Union and to find groups of countries that are homogenous, which can be a base for shaping long-term energy policies for the different groups (Bluszcz, 2017). Therefore, the main research question is: *How can we make groups of the member states of the European Union based on its energy dependence from 2016?*

In order to answer our research question and to find the optimal groups of EU countries, we use various Machine learning techniques, namely the K-means clustering algorithm and the K-harmonic means (KHM) clustering algorithm. K-means is one of the most popular clustering methods, as it has a fast and easy implementation. KHM differs from K-means because of its different objective function using the harmonic average, as well as assigning weights to the datapoints and using this in the calculation. Both algorithms need an initialization of the number of clusters, which we choose based on the outcomes of the silhouette method and Ward's method. Because KHM is said to be insensitive to its centre initialization (K-means is known for being sensitive and might converge into a local minimum), we perform a small simulation to check for this sensitiveness.

We find a grouping of 8 clusters of EU countries when using K-means, combined with Ward's method and the silhouette method. For KHM, 8 groups of countries does not seem to be the optimal grouping, as we found 13 clusters as the optimal number after comparing the results of the silhouette method. We find that KHM, compared to what is said by Güngör & Ünler (2008), is still sensitive to the centre initialization and gives us different cluster assignments when using different random starting points for cluster centres. However, we do find it is less sensitive than K-means.

In Section 2, a brief review of important papers and research is given, which is followed by a short overview of our data. Next, the methods used in order to answer our research question are explained in Section 4, followed by the clustering results and simulation example. We end with a small conclusion on the formed groups.

2 Literature Review

A lot of research has been done on different methods of clustering data. Cluster analysis is a Machine learning technique and is an important concept for data analysis. There are multiple types of clustering, for example hierarchical clustering, partioning methods, model-based clustering and fuzzy clustering. Partitioning methods need a pre-specificied number of clusters and then divides the dataset into these clusters. Hierarchical clustering does not need a pre-specified number of clusters. It starts with each datapoint in one cluster and then merging the most similar clusters into one new cluster. The result is a tree-based representation (dendrogram). Model-based clustering considers the data as coming from a mixture of probability distributions and uses certain models for clusters. Lastly, datapoints in fuzzy clustering can be part of multiple clusters and are given a membership value for each cluster (Kassambara, 2017).

The most popular partitioning method is the K-means clustering, once introduced by MacQueen (1967). Nazeer & Sebastian (2009) find that different sets of initial values as centres produces different results of clusters using the K-means algorithm, meaning K-means is very sensitive to its cluster initialization. To overcome this sensitivity of the initialization of centres, Zhang et al. (1999) propose an extension to the original K-means algorithm, namely the K-harmonic means (KHM). Güngör & Ünler (2008) find that KHM is robust to the initialization of the centres, which makes it a better clustering method. However, KHM can still converge into a local minimum, which is another problem (Bouyer & Farajzadeh, 2015).

Another way of clustering is by using a hierarchical agglomerative cluster method. There are multiple ways of forming the clusters, with each linkage form yielding an unique hierarchical method. One of these popular linking methods is Ward's method, proposed by Ward (1963). This method takes both between-cluster distances as well as within-cluster distances into consideration. The assumption is made that in the merging process there are only two points at minimal linkage value of eachother that are then merged. However, it is not specified what to do when there are more than two points suitable for merging in a certain step (Carlsson & Mémoli, 2010).

Bluszcz (2017) uses the hierarchical agglomerative cluster method to determine the number of clusters that the K-means and KHM need as input to form the clusters based on energy dependence of European countries. However, the created dendrogram from which the decision is made on how many clusters are needed, is just a graphical insight and gives no explanation on why a certain value of K clusters is chosen. A direct method to indicate the number of clusters is the silhouette method, which uses the silhouette value to show how well an observation is clustered. Kaufman & Rousseeuw (1990) propose to choose the number of clusters for which the silhouette value is maximized.

Overall, we use the clustering methods used by Bluszcz (2017) as a base to then implement methods that could improve the research and the formed clusters, by using KHM as an extension on K-means for its robustness as well as the silhouette method for deciding the optimal number of clusters.

3 Data

The data we use is retrieved from Eurostat, in the statistical book "Energy, Transport and Environment Indicators (2019)". The import dependence (ID) is calculated as follows (European Commission, 2013):

$$ID = \frac{M_j - X_j}{GIC_j + Bunk_j}, \tag{1}$$

where M is the import, X is the export, j is the energy product, GIC is the gross inland consumption and $Bunk$ is the consumption of international bunkers and the unit is %. This shows how much a country is dependent on energy imports to meet its needs. Using (1), the energy dependence of solid fuels and derivates, total petroleum products, and natural gas is calculated per country for the year 2016, as shown in Table 1. Because of missing values for the countries Cyprus and Malta, they are deleted from the dataset and are not used for clustering.

It can be seen that Luxembourg, Portugal and Sweden are highly dependent on the import of all three products, for example 102.3% and 108.4% for solid fuels in Portugal and Sweden respectively. Croatia is very dependent on the import for solid fuels as well (102.0%), but is less dependent on natural gas (33.5%). Poland and Czech Republic are self-sufficient in solid fuels (-12.0% and -0.9%), as well as Denmark and the Netherlands are for natural gas (-44.4% and -32.7%). Remarkably, none of the European countries is self-sufficient for petroleum products, meaning that the level of dependence of the EU countries on the import of petroleum products like oil is very high.

Table 1: Energy dependence level of EU member states in 2016

	Solid fuels and derivatives	Total petroleum products	Natural gas
Austria	95.5	92.0	85.8
Belgium	94.9	98.8	100.6
Bulgaria	9.8	96.1	96.5
Croatia	102.0	76.9	33.5
Cyprus	N/A	N/A	N/A
Czech Republic	-0.9	97.2	95.7
Denmark	84.6	2.8	-44.4
Estonia	0.0	48.7	100.0
Finland	60.1	94.7	99.7
France	93.5	97.5	98.9
Germany	49.6	96.4	88.6
Greece	4.4	99.6	99.2
Hungary	34.5	89.3	78.9
Ireland	54.7	98.4	40.1
Italy	97.5	91.0	91.8
Latvia	84.4	109.1	83.4
Lithuania	89.2	97.9	100.6
Luxembourg	100.0	100.0	99.3
Malta	N/A	N/A	N/A
Netherlands	91.0	95.6	-32.7
Poland	-12.0	92.8	78.4
Portugal	102.3	96.9	99.1
Romania	20.2	56.7	13.0
Slovakia	83.3	91.8	92.8
Slovenia	17.2	100.3	99.4
Spain	76.0	99.2	98.7
Sweden	108.4	95.9	99.2
United Kingdom	50.8	33.9	46.5

4 Methods

In this section we discuss the methods used by Bluszcz (2017) and propose different methods as an extension. Before using the data mentioned in Section 3, we standardise it according to the following formula (Johnson & Wichern, 2014):

$$z_i = \frac{x_i - \bar{x}}{S_x}, \tag{2}$$

where $\bar{x}$ is the mean and S_x is the standard deviation of the corresponding variable. The standardized data can be found in Table 7 in Appendix A.

4.1 Ward's method

Ward's method is an agglomerative hierarchical clustering method that starts with n singleton clusters, with n the number of observations, and stops when one big cluster of size n is created (Ward, 1963). Based on an optimal value of an objective function, clusters merge at each step. Ward's method chooses the error sum of squares (ESS) as a criterion for this objective function:

$$ESS = \sum_{p_j \in C_i} dist(p_j, m_i)^2, \tag{3}$$

where $dist(p_j, m_i)$ is the distance of datapoint p_j to a centre m_i of cluster C_i, with $i = 1, ..., k$ and $j = 1, .., n$. It minimizes the total within-cluster sum of squared distances. At each stage, the link is created which has the least increase in the ESS after merging the clusters.

The squared Euclidean distance is used as a measure for the distance between the datapoints (Ronald C. Henry, 1991):

$$D_{ij} = \sum_{v=1}^{m} (x_{iv} - x_{jv})^2, \tag{4}$$

where D_{ij} is the distance between points x_{iv} and x_{jv}, with $v = 1, ..., m$ the dimension of the data. Using the data from Section 3, $m = 3$.

Let d_{ik}, d_{jk} and d_{ij} be the distances between clusters C_i, C_j and C_k and let $d_{(ij)k}$ be the distance between cluster C_k and the new cluster $C_i \cup C_j$.The recursive formula for the hierarchical clustering using Ward's method is then (Szekely & Rizzo, 2005):

$$d_{(ij)k} = \alpha_i d_{ik} + \alpha_j d_{jk} + \beta d_{ij} + \gamma \left| d_{ik} - d_{jk} \right|, \tag{5}$$

with the following parameters of transformation:

$$\begin{aligned} \alpha_i &= \frac{n_1 + n_3}{n_1 + n_2 + n_3} \\ \alpha_j &= \frac{n_2 + n_3}{n_1 + n_2 + n_3} \\ \beta &= \frac{-n_3}{n_1 + n_2 + n_3} \\ \gamma &= 0, \end{aligned}$$

with n_1, n_2 and n_3 the cluster sizes of clusters C_i, C_j and C_k respectively. The dendrogram that is then created shows the hierarchical relation of the clusters and how the objects are connected to create the clusters.

4.2 K-means

After indicating how many clusters are formed with the hierarchical Ward's method in terms of the level and quality of the energy dependence, the K-means algorithm is used to optimize the initial grouping. The algorithm is one of the most common clustering methods in the current Machine learning literature. It is a non-model based algorithm, meaning it does not require a distribution of the data. K-means is popular in its use because of its low time complexity. The algorithm tries to minimize a criterion function $e(k)$ by finding a division of the datapoints between K clusters. The optimal partition for K-means minimizes the sum of the squared error within each cluster (Larose, 2004):

$$e(k) = \sum_{i=1}^{K} \sum_{p_j \in C_i} dist(p_j, m_i)^2, \tag{6}$$

where p_j is the datapoint in R^m space ($m = 3$ for our dataset), m_i is the centroid of cluster C_i and $dist(p_j, m_i)$ is the Euclidean distance $\|p_j - m_i\|$ between object p_j and the centroid m_i of the nearest cluster C_i. The K-means algorithm is then as follows (James et al., 2014):

1. Randomly assign a number, from 1 to K, to each datapoint. These serve as K initial cluster assignments;
2. Repeat steps (a) and (b) until there are no more changes in the cluster assignments:
   (a) For each cluster C_i, compute the cluster centroid m_i;
   (b) Assign each object $p_j \in D$ to the cluster C_i whose centroid m_i is closest (based on the Euclidean distance).

K-means clustering is commonly used because it is an easy algorithm, but it does need the initalization of the number of clusters K and the centroids. Often, the centroids are chosen randomly. However, K-means is known to be very sensitive to the initialization of the cluster centroids, meaning different random initializations lead to different clustering results. This non-robustness is the biggest drawback of the K-means algorithm (Ahmad & Hashmi, 2016).

4.3 Extensions

4.3.1 K-harmonic means

As an extension to K-means explained in Section 4.2, we introduce the K-harmonic means (KHM) method proposed by Zhang et al. (1999). The KHM algorithm is essentially insensitive to the initialization of the centres (Güngör & Ünler, 2008), which has been a problem before with the standard K-means, as it is very sensitive to its initalization. Therefore, KHM can be a more robust clustering algorithm than K-means.

Where the K-means algorithm uses the minimum distance from a datapoint to the centres, the KHM algorithm uses the harmonic average (HA) of the distances from all points in N to all the centres K. The HA is small if one of the distances is small and the HA is big if all distances to K are big. This way it behaves like a minimum function while taking all datapoints into account and giving them weights. If two or more centres are close to a datapoint, KHM will move one of these centres to an area where there is a datapoint with no close centre, which will lower the value of the objective function. The objective function using the HA is defined as follows:

$$KHM(P,M) = \sum_{j=1}^{n} \frac{k}{\sum_{i=1}^{k} \frac{1}{\left\|p_j - m_i\right\|^w}}, \tag{7}$$

where w is an input parameter which is typically $w \geq 2$ (Güngör & Ünler, 2007).

We need to compute the membership $m(m_i|p_j)$ for each point p_j in each centre m_i:

$$m(m_i|p_j) = \frac{\left\|p_j - m_i\right\|^{-w-2}}{\sum_{i=1}^{k} \left\|p_j - m_i\right\|^{-w-2}}, \tag{8}$$

as well as computing its weight $w(p_j)$:

$$w(p_j) = \frac{\sum_{i=1}^{k} \left\|p_j - m_i\right\|^{-w-2}}{\left(\sum_{i=1}^{k} \left\|p_j - m_i\right\|^{-w}\right)^2}. \tag{9}$$

The function assigns a small weight to points that are close to one or more centres, and assigns a large weight to points that are not close to any centre. This way, KHM overcomes the problem of highly densely areas of cluster centres and can still move away a cluster centre that is positioned in such areas. This is also the reason why KHM is less sensitive to its centre initializations then K-means, because K-means tends to have cluster centres trapped in these densed areas.
The KHM algorithm is then as follows (Yang et al., 2009):

1. Randomly choose the initial centres m_i for cluster C_i
2. Repeat steps (a)-(c) until the objective function KHM(P,M) does not change significantly:

(a) Calculate the objective function given in (7)

(b) For each point p_j, compute the membership $m(m_i|p_j)$ and its weight $w(p_j)$

(c) For each centre m_i, re-compute its location from all points p_j:

$$m_i = \frac{\sum_{j=1}^{n} m(m_i|p_j)w(p_j)p_j}{\sum_{j=1}^{n} m(m_i|p_j)w(p_j)} \tag{10}$$

3. Assign point p_j to cluster c_j with the biggest membership.

The KHM algorithm reduces the weakness of the K-means algorithm by using the conditional probability of cluster centres to datapoints and by using weights. Therefore we expect the KHM to perform better than the K-means and giving us different results (Jiang et al., 2010).

4.3.2 Choosing K clusters - Silhouette method

The K-means and K-harmonic means algorithms need an initialization of the number of clusters. We choose this number of clusters based on the dendogram created from Ward's method in Section 4.1. However, this number is based purely by 'looking' at the dendrogram, which means there is no solid explanation on choosing the number of clusters. We would like to investigate if using a direct method like the silhouette method indicates a different number of K as the optimal number of clusters.

The silhouette method uses the silhouette value to measure how similar an observation is to the cluster it is in compared to the other clusters. These values can then be plotted to give a graphical insight on how well an observation is clustered and to see for which number of clusters most of the observations are clustered well (Wang et al., 2017).

We introduce a certain silhouette value s_i, that we compute for each datapoint i and later on are combined in a plot. In case of dissimilarities, take any datapoint i and denote A as the cluster where point i is in. If there are more objects than only i in cluster A (if i is the only object in A, $a(i)$ is set to zero), we can compute (Rousseeuw, 1987):

$$a(i) = \text{average dissimilarity of } i \text{ to all other objects in } A.$$

For all clusters $C \neq A$, we compute:

$$d(i, C) = \text{average dissimilarity of } i \text{ to all objects in } C.$$

We select $b(i) = \min_{C \neq A} d(i, C)$, which then lets us compute the silhouette value $s(i)$ as follows:

$$s_i = \frac{b(i) - a(i)}{\max\{a(i), b(i)\}}. \tag{11}$$

In case of similarities, we define $a'(i)$ and $d'(i, C)$ as the average similarities, and we select $b'(i) = \max_{C \neq A} d'(i, C)$. Then, $s(i)$ is defined as:

$$s(i) = \begin{cases} 1 - \frac{b'(i)}{a'(i)} & \text{if } a'(i) > b'(i) \\ 0 & \text{if } a'(i) = b'(i) \\ \frac{a'(i)}{b'(i)} - 1 & \text{if } a'(i) < b'(i). \end{cases} \tag{12}$$

Plotting s_i for all i in A in decreasing order gives us the silhouette of A. Observations with a large silhouette value are well clustered, a negative value means its cluster is wrong, and a value around zero means the observation is between clusters.

To determine the best number of clusters using this method, we perform the K-means and KHM for multiple choices of K, and check for the best average silhouette of the observations. We want the mean silhouette value $\bar{s}$ to be as close to one as possible and we want the plot of each cluster to be above the mean. If having clusters of approximately the same size is important as well, the width of the plot has to be as uniform as possible. Taking all these things into account, we compare the silhouette plots and choose the optimal number of clusters (Kaufman & Rousseeuw, 1990).

5 Results

The research question is: *How can we make groups of the member states of the European Union based on its energy dependence from 2016?* We perform the K-means algorithm, as well as an extension by performing the K-harmonic means algorithm to find the optimal grouping in terms of energy dependence. As both algorithms need an initalization of the number of K, Ward's method is used to create a dendrogram to choose the number of clusters. A more direct method that is also used to choose K is the silhouette method.

5.1 Ward's method

Figure 1 shows the dendrogram created by Ward's hierarchical clustering method. Based on the dendrogram, we decide to choose eight clusters, as this looks like a good grouping of the countries based on the binding and linkage. Each colour forms a cluster of multiple countries and the black nodes form a cluster of one country each.

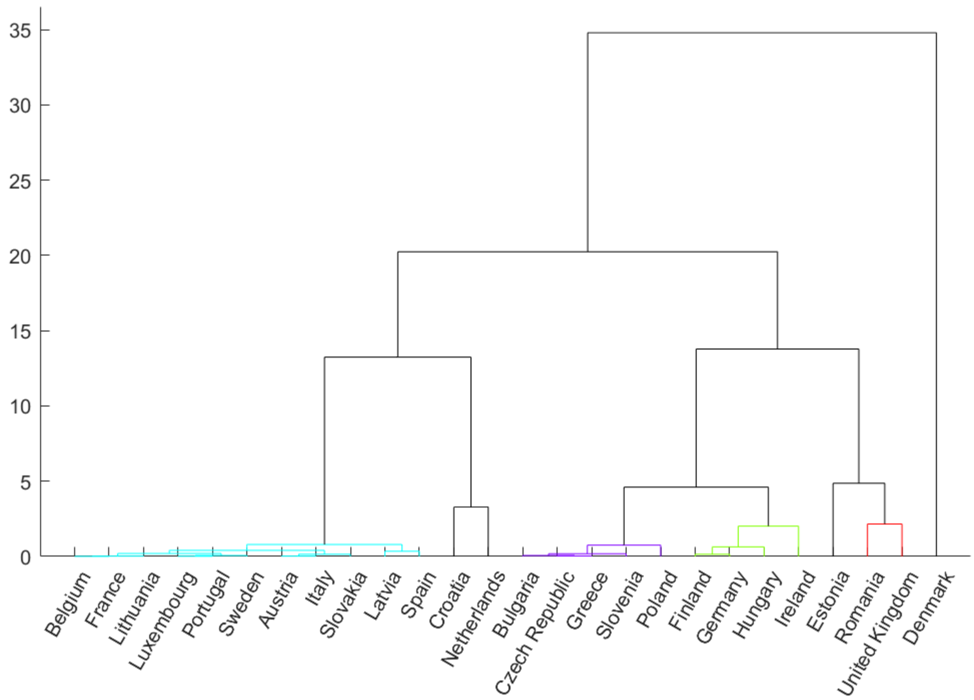


Figure 1: Dendrogram of grouping EU countries according to their energy dependence level

5.2 K-means

The next step is to find the optimal clustering by using K-means. After performing Ward's method and choosing eight clusters in Section 5.1, we perform the K-means algorithm with the choice of $K = 8$. The formed clusters are presented in Table 2, as well as the distance from each datapoint to the centre of its cluster and the within-cluster sums of these distances per cluster.

Four clusters are formed with only one country (Cluster 3: Denmark, Cluster 5: Netherlands, Cluster 6: Croatia, Cluster 7: Estonia). In cluster 1, the datapoint furthest from its centre is Finland (0.6130). As is shown in the dendrogram in Figure 1, Finland is in a different cluster. However, cluster 1 seems to be the correct cluster for Finland (smallest Euclidean distance).

Cluster 4 and cluster 8 have the biggest mean distance from the centroid of the cluster. To give a better graphical insight, a plot of all datapoints in R^3 space including the clusters and its centres is shown in Figure 2.

Table 2: Optimisation of clusters with K-means where K=8

Country	Distance from centroid of cluster	Within-cluster sums of point-to-centroid distances
Cluster 1		1.8877
Austria	0.1201	
Belgium	0.0316	
Finland	0.6130	
France	0.0121	
Italy	0.1045	
Latvia	0.3614	
Lithuania	0.0157	
Luxembourg	0.0811	
Portugal	0.0976	
Slovakia	0.0854	
Spain	0.1472	
Sweden	0.2180	
Cluster 2		0.5601
Bulgaria	0.0303	
Czech Republic	0.0157	
Greece	0.0272	
Poland	0.3343	
Slovenia	0.1526	
Cluster 3		0.0
Denmark	0.0	
Cluster 4		1.0785
Romania	0.5393	
United Kingdom	0.5393	
Cluster 5		0.0
Netherlands	0.0	
Cluster 6		0.0
Croatia	0.0	
Cluster 7		0.0
Estonia	0.0	
Cluster 8		1.0053
Hungary	0.1951	
Ireland	0.5738	
Germany	0.2364	

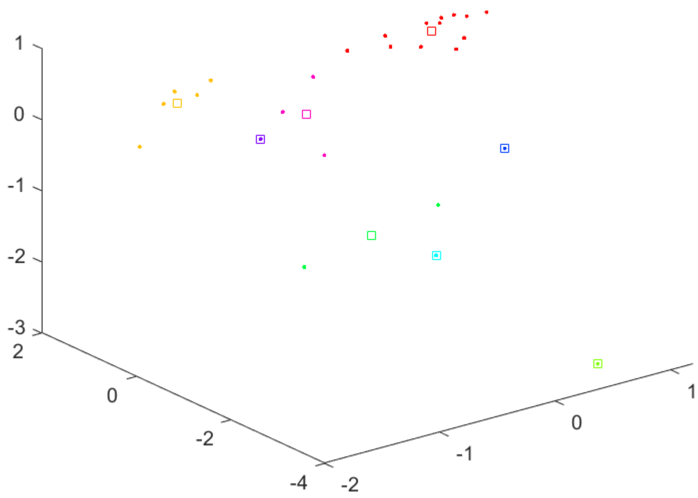


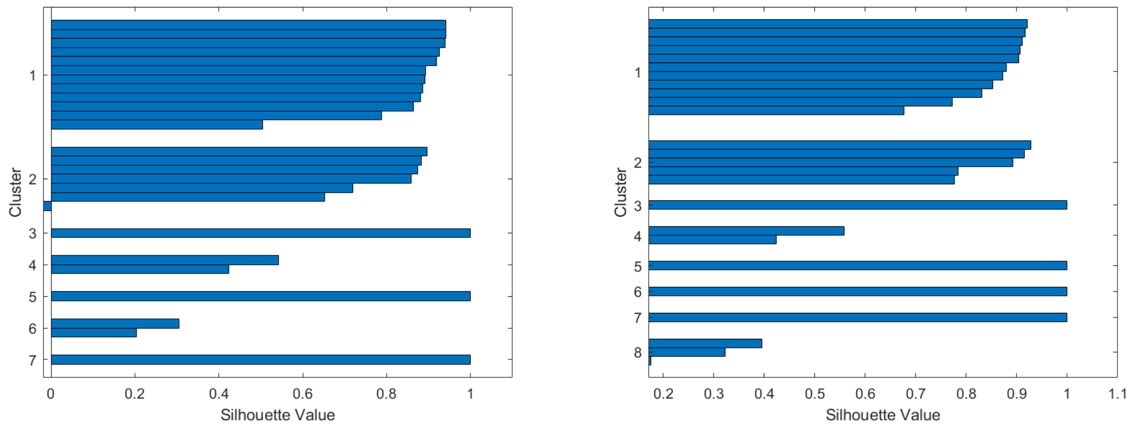
Figure 2: Plot for clusters K-means with K=8

5.2.1 Silhouette method

For choosing the optimal number for K, the silhouette method gives us the silhouette value s_i which tells us how well each datapoint is clustered. We perform the K-means algorithm for $K = 6, ..., K = 13$ and plot the silhouette values. The mean silhouette values $\bar{s}$ are shown in Table 3. The silhouette plots for $K = 7$ and $K = 8$ are presented in Figure 3. The silhouette plots for the other number of clusters can be found in Figure 6 in Appendix B.

Table 3: Mean Silhouette Value of K-means

Number of clusters	*K=6*	*K=7*	*K=8*	*K=9*	*K=10*	*K=11*	*K=12*	*K=13*
Mean silhouette value	0.7184	0.7582	0.7610	0.6200	0.6591	0.5793	0.7015	0.7050



(a) K=7

(b) K=8

Figure 3: Silhouette plots

As stated in Section 4.3.2, to determine the optimal number for K, we want the mean silhouette value to be as close to one as possible. This is for $K = 8$, with a mean silhouette value of 0.7610. When looking at the silhouette plot, we want the width to be as uniform as possible as well as the plot of each cluster to be above the mean silhouette value. When looking at the silhouette plot for $K = 8$, we see in Figure 3b that the width is not very uniform. However, this is because we only have a small number of datapoints to cluster (26 countries) which leads to four clusters with only one datapoint. In our research, the uniformness of the clusters is difficult to obtain and less important.

The silhouette values for cluster 4 and cluster 8 are below the mean silhouette value. The closer the silhouette value is to zero, the closer an observation is to the decision boundary between two neighboring clusters. However, no silhouette value is below zero, meaning it is in the wrong cluster. This is the case for $K = 7$, which can be seen in Figure 3a. Cluster 2 contains a datapoint with a negative value which can be a sign that it is not in the correct cluster. Taking this into account, as well as that the values of cluster 4 and cluster 6 are below the mean for $K = 7$, we choose $K = 8$ based on the silhouette method.

5.3 K-harmonic means

We perform the KHM algorithm with the same number of clusters, $K = 8$. The formed clusters, the distance from each datapoint to the centre of its cluster and the within-cluster sums of these distances are presented in Table 4.

Table 4: Optimisation of clusters with KHM where K=8

Country	Distance from centroid of cluster	Within-cluster sums of point-to-centroid distances
Cluster 1		2.1062
Belgium	0.0656	
France	0.0653	
Latvia	0.6548	
Lithuania	0.1777	
Luxembourg	0.1339	
Portugal	0.1704	
Spain	0.5079	
Sweden	0.3306	
Cluster 2		
Netherlands	0.0904	0.0904
Cluster 3		
Germany	0.1002	0.9755
Finland	0.3860	
Hungary	0.4893	
Cluster 4		1.4065
Bulgaria	0.1365	
Czech Republic	0.1514	
Greece	0.1483	
Poland	0.6155	
Slovenia	0.3548	
Cluster 5		0.5473
Austria	0.1389	
Italy	0.1887	
Slovakia	0.2197	
Cluster 6		0.0624
Denmark	0.0624	
Cluster 7		3.4643
Croatia	1.3846	
Ireland	0.1811	
Romania	1.8986	
Cluster 8		1.9460
Estonia	0.2143	
United Kingdom	1.7317	

Where K-means has four clusters formed of only one country, KHM only gives us two clusters with one country (Cluster 2: Netherlands, Cluster 6: Denmark). Also, KHM has a more uniform division between the clusters. The biggest cluster is of size 8, which is size 12 for K-means. The other clusters are either of size 5, 3, 2 or 1. In addition, the within-cluster sums of point-to-centroid distances from the KHM clusters are larger than for the K-means. The average within-cluster sum for KHM is 1.3248, for K-means this is 0.5665. One reason for this is that for K-means, the datapoint in a cluster of size 1 is also the centre of that cluster. This is not the case for KHM, where for Cluster 2 and Cluster 6 the distance from the datapoint to the centre is not equal to zero. This is because KHM takes all datapoints into account while calculating a new centre. K-means calculates the minimum distance of only the datapoints in that particular cluster.

To give a better graphical insight in the formed clusters and centres, a plot of all datapoints in R^3 space is shown in Figure 4.

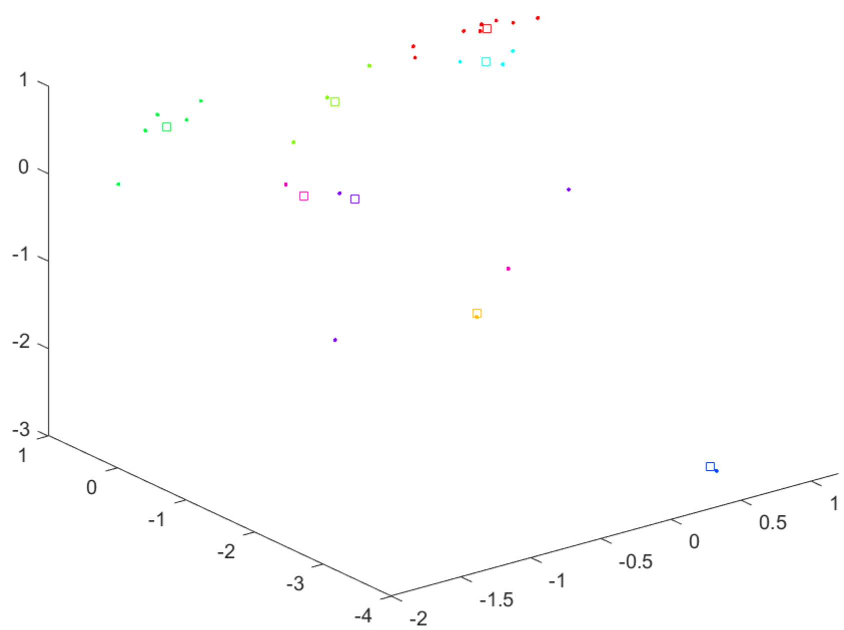


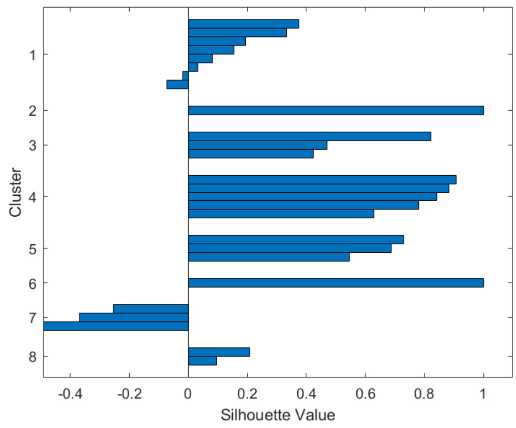
Figure 4: Plot for clusters KHM with K=8

5.3.1 Silhouette method

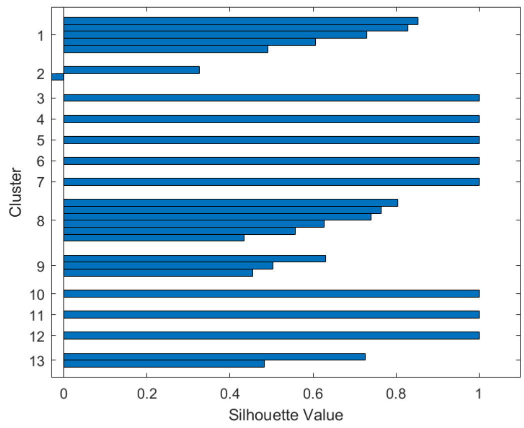
To choose the optimal number of clusters for the KHM algorithm, we performed KHM for $K = 6, ..., K = 15$ and plot the silhouette values, which can be found in Figure 8 in Appendix B. The mean silhouette values $\bar{s}$ are shown in Table 5. The plots for $K = 8$ and $K = 13$ are presented in Figure 5.

Table 5: Mean Silhouette Value of KHM

Number of clusters	*K=6*	*K=7*	*K=8*	*K=9*	*K=10*	*K=11*	*K=12*	*K=13*	*K=14*
Mean silhouette value	0.6644	0.6452	0.3841	0.3999	0.5306	0.5852	0.5674	0.7125	0.6761



(a) K=8



(b) K=13

Figure 5: Silhouette plots

The biggest mean silhouette value is for $K = 13$, which is 0.7125. Where Ward's method as well as the silhouette method for K-means say $K = 8$ is a good choice of cluster size, the mean silhouette value for KHM is only 0.3841, meaning it is not a good choice here. Looking at only the silhouette value, we would choose $K = 13$ clusters.

When looking at the plots, we see that for $K = 8$ there are a few negative values and some very small positive values. This means some datapoints are in the wrong cluster, as well as some datapoints are very close to the decision boundary between two clusters. Looking at the plot for $K = 13$, we see that there is only one datapoint with a negative value. Almost all clusters are above the mean silhouette value of 0.7125, except for cluster 2 and cluster 9.

Lastly, while looking at the uniformness of the plots, we can see that for $K = 13$ there are 8 clusters of size 1. Because we have a small dataset of only 26 observations, the uniformness is hard to obtain.

Taking everything into account and also comparing the plots for $K = 13$ with the other plots in Appendix B, we decide that 13 clusters is the best choice according to the silhouette method for performing the KHM algorithm. However, we would prefer a smaller number of clusters, as we only have to cluster 26 datapoints. The choice for $K = 13$ is purely based on this method and its criteria.

5.4 Sensitivity of cluster centre initializations

To check the insensitivity of the KHM algorithm, as mentioned in Section 4.3.1, we perform a small simulation. For 1000 random cluster initializations, we check how many unique cluster assignments we get from both KHM and K-means with $K = 8$. The results are presented in Table 6. Out of 1000 runs of the K-means algorithm with different random cluster centre initializations, we get 996 unique cluster assignments. This shows the sensitivity of K-means for the initialization, as we get 996 different outcomes.

For the KHM algorithm, we get 837 unique cluster assignments, for 1000 different random initializations. This is in contradiction with what is stated by Güngör & Ünler (2008), saying KHM solves the initialization problem of K-means. We still get 837 different outcomes of cluster assignments, meaning it is still sensitive to initialization. However, comparing it to K-means, we can state it is less sensitive.

Table 6: Cluster initialization sensitivity simulation

	K-means	**KHM**
Unique cluster assignments	996	837
Total runs	1000	1000

6 Conclusion

We want to investigate multiple ways of clustering data, in order to find homogeneous groups of EU countries, based on their energy dependence. We perform two partitioning methods of clustering, namely K-means and K-harmonic means. As both of these methods need a predetermined number of clusters, we choose the best number of clusters by using the silhouette method and the hierarchical Ward's method.

The first step is performing Ward's method, using the squared Euclidean distance. After creating the dendrogram, we find that $K = 8$ is a good choice of the number of clusters for our data. We decide to perform both K-means and K-harmonic means with initially eight clusters.

The K-means algorithm clusters the datapoints based on minimizing the sum of the squared error within each cluster. Out of the eight clusters, we find four clusters of size 1, all other clusters have sizes 12, 5, 3 and 2. The K-harmonic means makes clusters using the harmonic average of all datapoints. We find only two clusters of size 1, and other clusters of size 8, 5, 3 and 2. There is a small difference noticeable between the formed clusters of K-means and KHM. Comparing the distances, we find a bigger average within-cluster sum of point-to-centroid distances for KHM.

Lastly, we perform the silhouette method as a way to determine the best number of clusters. After running K-means for $K = 6, ..., K = 13$, we find the best mean silhouette value for $K = 8$, and also after comparing the silhouette plots we decide $K = 8$ as the best number of clusters. This is in line with the decision based on Ward's method. However, when performing KHM for $K = 6, ..., K = 15$, we find that eight clusters is not the best choice, as it has the lowest mean silhouette value, as well as a few negative values, meaning certain datapoints are clustered wrong. We choose $K = 13$ for KHM based on the highest mean silhouette value and the best looking silhouette plot.

In order to check the sensitivity of both algorithms, we perform a small simulation and check how many unique cluster assignments are given. It is said for KHM to be insensitive to its centre initialization, however, after performing a simulation of 1000 runs, we get 837 unique cluster assignments, meaning it is still sensitive to random centre initalization. For K-means, we find 996 unique cluster assignments, showing us the already known sensitivity. We do see that KHM is less sensitive, as it has less unique assignments.

In conclusion, both K-means and K-harmonic means are good partitioning methods for clustering European countries into homogenous groups based on their energy dependence, where KHM is more robust for its initialization. However, we do find a difference in the best number of clusters for both methods after performing the silhouette method. It would be interesting to see if a different method of choosing the number K for KHM would give us another optimal number of clusters. We prefer a lower number than 13 clusters, given that we only have 26 datapoints to cluster, which is an average of two points per cluster. Taking our goal of the clustering into account (help shaping long-term energy policies), we should reconsider if 13 clusters is helpful for this goal. However, based on the criteria for the silhouette method, 13 clusters really comes out as the optimal number. For further research, hierarchical aggregation could be interesting to lower the number of clusters, or using a different method like the elbow method to choose the best number of K. We would prefer a solution lower than the given solution of the silhouette method.

Considering the sensitivity of both K-means and KHM, it would be useful to look into other algorithms that do not have this sensitivity. An interesting option would be to extend the KHM using a kernel, or by using particle swarm optimization.

For the data, we only have information on three given variables, namely solid fuels and derivates, total petroleum products, and natural gas. It would be interesting to add more country-dependent variables that can help explain the energy dependence level and capture more information to help create good clusters. An example could be to add information about the current state of the economy of a country.

References

Ahmad, A., & Hashmi, S. (2016). K-Harmonic Means type Clustering algorithm for Mixed Datasets. *Applied Soft Computing*, *48*(1), 39-49.

Bluszcz, A. (2017). European Economies in Terms of Energy Dependence. *Quality & Quantity: International Journal of Methodology*, *51*(4), 1531-1548.

Bouyer, A., & Farajzadeh, N. (2015). An Optimized K-Harmonic Means Algorithm Combined with Modified Particle Swarm Optimization and Cuckoo Search Algorithm. *Journal of Intelligent Systems*, *29*(1), 1-18.

Carlsson, G., & Mémoli, F. (2010). Characterization, Stability and Convergence of Hierarchical Clustering Methods. *Journal of Machine Learning Research*, *11*(47), 1425-1470.

European Commission. (2013). European Economy Member States' Energy Dependence: An Indicator—Based Assessment. *Occassional Papers 145*.

European Commission. (2018). EU Energy in Figures. *Publications Office of the European Union*.

Güngör, Z., & Ünler, A. (2007). K-Harmonic Means Data Clustering with Simulated Annealing Heuristic. *Applied Mathematics and Computation*, *184*(2), 199-209.

Güngör, Z., & Ünler, A. (2008). K-Harmonic Means Data Clustering with Tabu-search Method. *Applied Mathematical Modelling*, *32*(6), 1115-1125.

James, G., Witten, D., Hastie, T., & Tibshirani, R. (2014). An Introduction to Statistical Learning: With Applications in R. In (7th ed., p. 386 - 390). Springer.

Jiang, H., Yi, S., Li, J., Yang, F., & Hu, X. (2010). Ant Clustering Algorithm with K-Harmonic Means Clustering. *Expert Systems with Applications*, *37*(12), 8679-8684.

Johnson, R., & Wichern, D. (2014). Applied Multivariate Statistical Analysis. In (6th ed., p. 11-17). Harlow: Pearsons Education Limited.

Kassambara, A. (2017). *Practical Guide to Cluster Analysis in R: Unsupervised Machine Learning* (Vol. 1). STHDA.

Kaufman, L., & Rousseeuw, P. J. (1990). Finding Groups in Data: An Introduction to Cluster Analysis. In (1st ed., p. 83-88). Wiley.

Larose, D. T. (2004). Discovering Knowledge in Data: An Introduction to Data Mining. In (1st ed., p. 128-146). Wiley.

MacQueen, J. (1967). Some methods for classification and analysis of multivariate observations. In *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, Volume 1: Statistics* (pp. 281–297). University of California Press.

Nazeer, K. A., & Sebastian, M. (2009). Improving the Accuracy and Efficiency of the K-means Clustering Algorithm. In *Proceedings of the world congress on engineering* (Vol. 1, pp. 1–3).

Ronald C. Henry. (1991). Multivariate Receptor Models. In *Receptor modeling for air quality management* (Vol. 7, p. 117 - 147). Elsevier.

Rousseeuw, P. J. (1987). Silhouettes: A Graphical Aid to the Interpretation and Validation of Cluster Analysis. *Journal of Computational and Applied Mathematics*, *20*(1), 53-65.

Szekely, G. J., & Rizzo, M. L. (2005). Hierarchical Clustering via Joint Between-Within Distances: Extending Ward's Minimum Variance Method. *Journal of Classification*, *22*(2), 151-183.

Wang, F., Franco-Penya, H.-H., Kelleher, J. D., Pugh, J., & Ross, R. (2017). An Analysis of the Application of Simplified Silhouette to the Evaluation of K-means Clustering Validity. In *Machine learning and data mining in pattern recognition* (p. 291-305). Springer.

Ward, J. H. (1963). Hierarchical Grouping to Optimize an Objective Function. *Journal of the American Statistical Association*, *58*(301), 236-244.

Yang, F., Sun, T., & Zhang, C. (2009). An Efficient Hybrid Data Clustering Method based on K-Harmonic Means and Particle Swarm Optimization. *Expert Systems with Applications*, *36*(6), 9847-9852.

Zhang, B., Hsu, M., & Dayal, U. (1999). K-Harmonic Means-A Data Clustering Algorithm. *Hewlett-Packard Labs Technical Report HPL-1999-124*, *55*.

Appendix

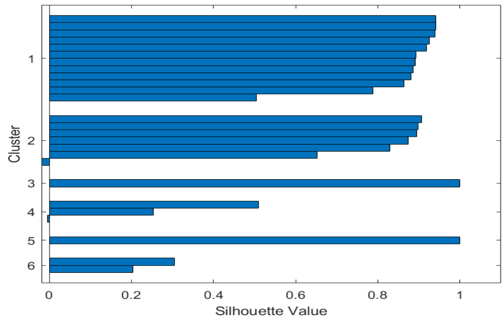
A Data

Table 7: Data after standardization

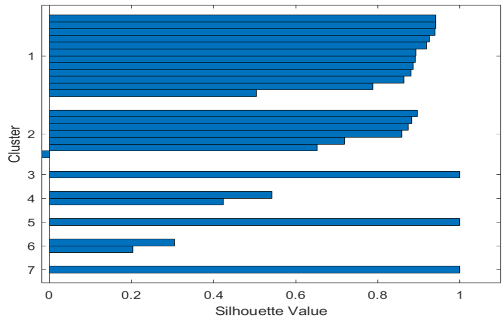
	Solid fuels and derivatives	Total petroleum products	Natural gas
Austria	0.8723	0.2259	0.2706
Belgium	0.8571	0.5061	0.6319
Bulgaria	-1.3067	0.3948	0.5318
Croatia	1.0376	-0.3964	-1.0061
Czech Republic	-1.5788	0.4402	0.5122
Denmark	0.5952	-3.4503	-2.9077
Estonia	-1.5559	-1.5586	0.6172
Finland	-0.0278	0.3372	0.6099
France	0.8215	0.4525	0.5904
Germany	-0.2947	0.4072	0.3389
Greece	-1.444	0.5391	0.5977
Hungary	-0.6787	0.1146	0.1021
Ireland	-0.1651	0.4896	-0.8450
Italy	0.9232	0.1847	0.4170
Latvia	0.5901	0.9306	0.2120
Lithuania	0.7121	0.4690	0.6319
Luxembourg	0.9867	0.5556	0.6001
Netherlands	0.7579	0.3742	-2.6221
Poland	-1.861	0.2588	0.0899
Portugal	1.0452	0.4278	0.5952
Romania	-1.0423	-1.2289	-1.5065
Slovakia	0.5621	0.2176	0.4415
Slovenia	-1.1186	0.5679	0.6026
Spain	0.3765	0.5226	0.5855
Sweden	1.2003	0.3866	0.5977
United Kingdom	-0.2642	-2.1686	-0.6888

Note: Cyprus and Malta have been deleted because no data was available.

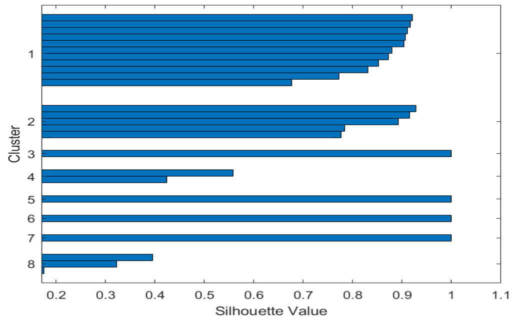
B Silhouette Method



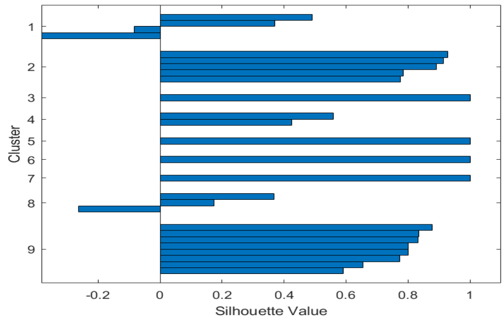
(a) K=6



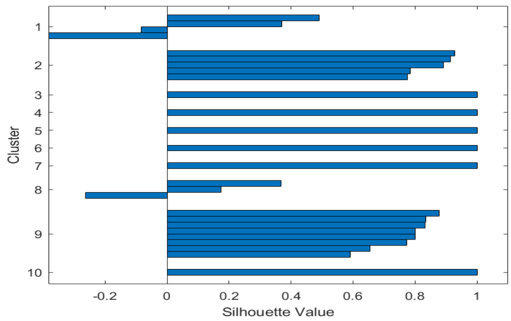
(b) K=7



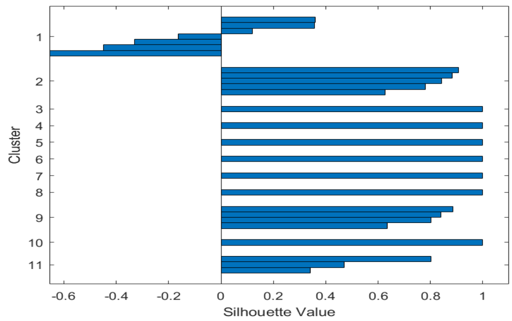
(c) K=8



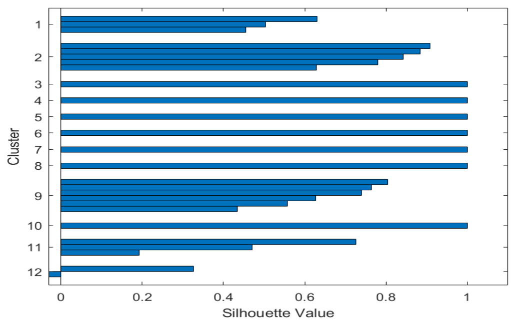
(d) K=9



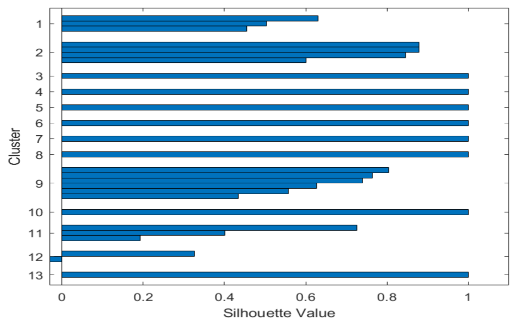
(e) K=10



(f) K=11

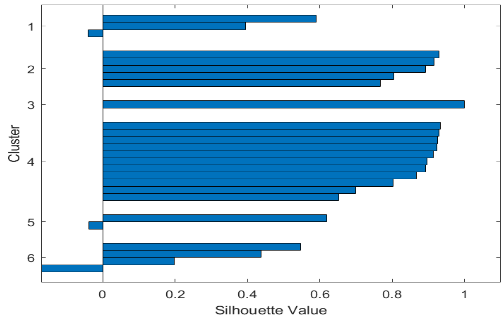


(g) K=12

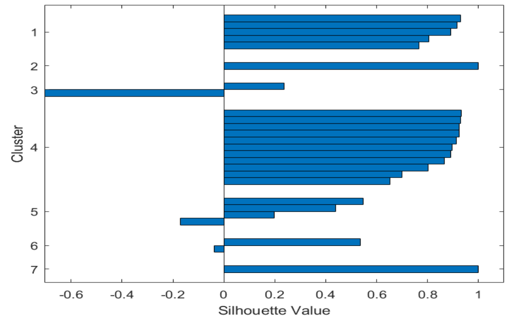


(h) K=13

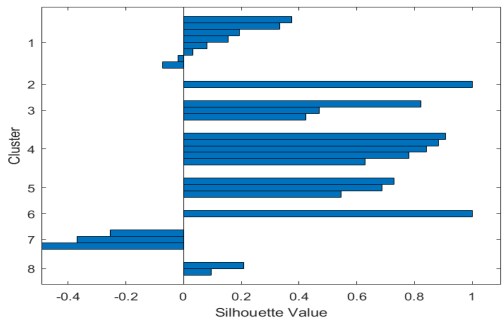
Figure 6: Silhouette plots for K-means



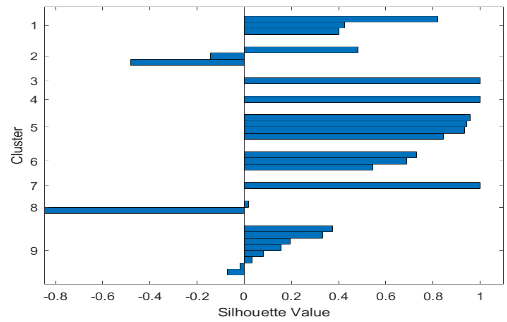
(a) K=6



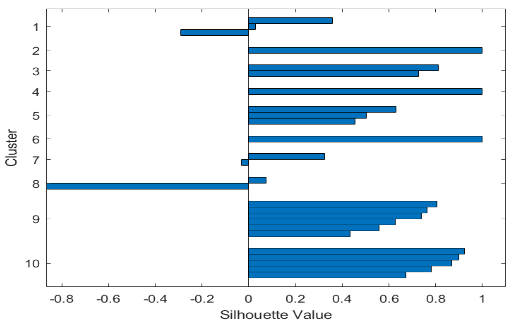
(b) K=7



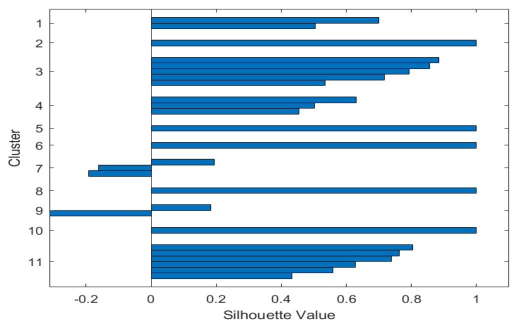
(c) K=8



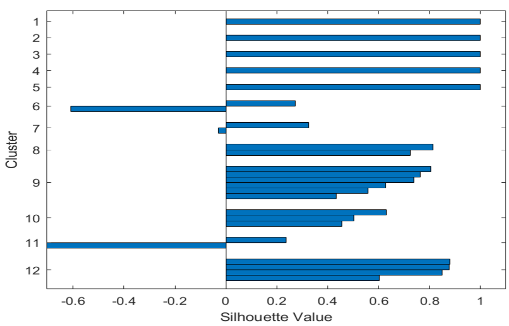
(d) K=9



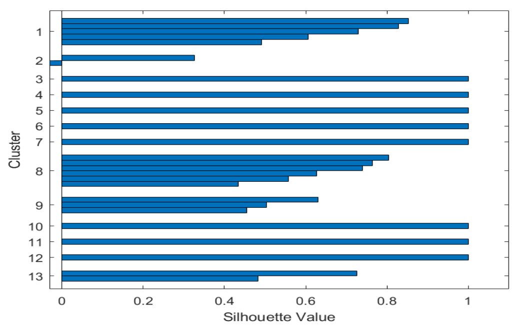
(e) K=10



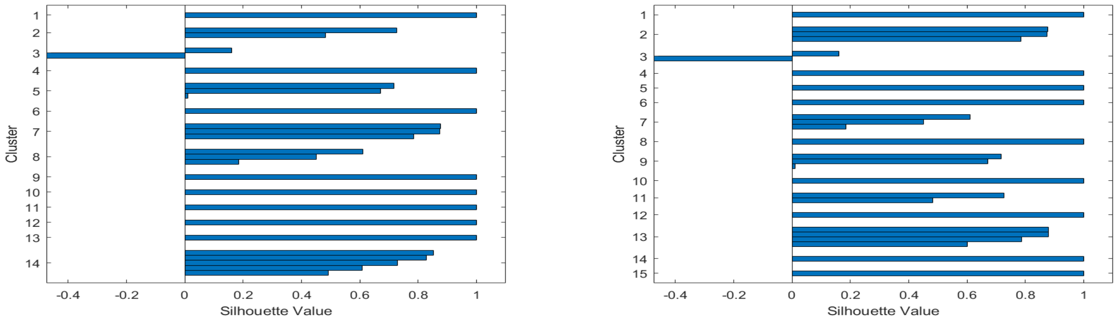
(f) K=11



(g) K=12



(h) K=13



(a) K=14 (b) K=15

Figure 8: Silhouette plots for KHM

C Codes

C.1 Ward's method, K-means, Silhouette method

```
rng('default')
load('dataset.mat')
Z = zscore(dataset);
Y = linkage(Z,'ward', 'squaredeuclidean');  %Ward's method
%dendrogram(Y)                              %Dendrogram
%y=1000;                                    %Simulation
%iter=1;                                    %Simulation
%K=[];                                      %Simulation

%while iter<=y                              %Simulation
[idx,C, sumD, D]=kmeans(Z,13);               %K-means
D=(min(D'))';                               %Distance from datapoint to cluster
    centre
%iter=iter+1;                               %Simulation

%K=[K idx];                                 %Simulation
%end                                        %Simulation
silhouette(Z,idx);                         %Silhouette plot
%PlotClusters(Z,idx,C);                     %ClusterPlot
s=silhouette(Z,idx);                       % SIlhouette values
means=mean(s);                             %Mean Silhouette value
```

C.2 K-harmonic means

```
clear
clc
%load Dataset
load 'Dataset.mat'
DataStand=zscore(dataset);
%(n samples) (p dimensions)
[n,p]=size(DataStand);
x=DataStand;
S=[];                                        %For storing all assignments
                                             %in simulation
y=1000;
itersim=1;

while itersim<=y

k=8;
c=zeros(k,p);
u=zeros(n,k);
w=zeros(n,1);
delta = 1e-5;
d=1000;
iter=1;
q=2;
Objective=[10000;9999];                     % While loop until objective
                                            %doesn't change significantly
                                            %anymore -- these values
                                            %are just for starting, make
                                            %sure the algorithm starts.
                                            %Saves the objective
                                            %function after in vector
DistCentrPoint=[];


%initialize random cluster centers
for i =1:k
    %randomly select vector from the DataStand
    c=-3+(3+3)*rand(k,p);
end
```

```
%clustering loop

while((Objective(end-1)-Objective(end))>delta)
J=0;
%Calculate the objective function(J)
for j=1:n
    JJ=0;
    for i=1:k
        dist = pdist([x(j,:);c(i,:)],'euclidean');
        JJ = dist^(-2)+JJ;
    end
     J=J+(k/JJ);
end
Objective=[Objective;J];

%determine membership matrix(U)
 for j=1:n
        for i=1:k
            KK=0;
            for l=1:k
                dist_som = pdist([x(j,:);c(l,:)],'euclidean');
                if dist_som == 0
                    KK=KK+0;
                else
                    KK=KK+dist_som^(-q-2);
                end
            end
            distm = pdist([x(j,:);c(i,:)],'euclidean');
            if distm ==0
                u(j,i) = 0;
            else
                u(j,i)=distm^(-q-2)/KK;
            end

        end
  end

%Calculate weight function (W)
 for j=1:n
     dist1=0;
     dist2=0;
        for i=1:k
```

```
            distw = pdist([x(j,:);c(i,:)],'euclidean');
            if distw == 0
            dist1=dist1+0;
            dist2=dist2+0;
            else
            dist1 = dist1+distw^(-q-2);
            dist2=dist2+distw^(-q);
            end
        end
        w(j,:)=dist1/(dist2^2);
 end

%update cluster center

for i=1:k
    teller=0;
    noemer=0;

    for j=1:n
        teller = (u(j,i)*w(j)*x(j,:))+teller;
        noemer = (u(j,i)*w(j))+noemer;
    end
    c(i,:)=teller/noemer;
end
c=sort(c);
iter = iter +1;
end

%assign point pj to cluster with biggest membership
for j=1:n
 [~,Assignment]= max(u,[],2);
end

%Compute distances from centroid of cluster to datapoint
for j=1:n
D = pdist([x(j,:);c(Assignment(j),:)],'euclidean');
DistCentrPoint=[DistCentrPoint;D];
end

%Store all assignments for simulation
S=[S Assignment];
```

```
itersim=itersim+1;
end
%silhouette(x,Assignment);
%s=silhouette(x,Assignment);
%means=mean(s);
DifferentClusterAssignments=unique(S', 'rows');
```

C.3 Plot clusters function

```
%PlotClusters   A function to plot clusters with different colors
%   PlotClusters(DATA, IDX) plots the m-by-d matrix of data points, DATA,
%   with the associated cluster m-elements index array, IDX. IDX can be
%   obtained by a clustering function like kmeans. The function finds the
%   centroid of each cluster and plots it as well. Default colors are
%   assigned to each cluster.
%
%   PlotClusters(DATA, IDX, CENTERS) additionally, allows you to specify
%   the position of the cluster centers, CENTERS, as a c-by-d matrix, where
%   c is the number of unique clusters in IDX.
%
%   PlotClusters(DATA, IDX, CENTERS, COLORS) additionally, allows you to
%   specify the colors to use, COLORS, as a c-by-3 array similar to the one
%   given by the command hsv.
%
% Example:
% --------
%   % Use kmeans to get clustered data.
%   X = [randn(20,2)+10*ones(20,2); randn(20,2)-10*ones(20,2)];
%   [idx, ctrs] = kmeans(X, 2);
%   PlotClusters(X, idx, ctrs)
%
% See also: kmeans, plot, plot3

%Author: Elad Kivelevitch
%Version: 2.0
%Date: 19 August, 2017

function []=PlotClusters(Data,IDX,Centers,Colors)
%Checking inputs
switch nargin
    case 1 %Not enough inputs
        error('Clustering data is required to plot clusters. Usage: PlotClusters(
```

```
        Data,IDX,Centers,Colors)')
case 2 %Need to calculate cluster centers and color scheme
    [NumOfDataPoints,Dimensions]=size(Data);
    if Dimensions~=2 && Dimensions~=3 %Check ability to plot
        error('It is only possible to plot in 2 or 3 dimensions.')
    end
    if length(IDX)~=NumOfDataPoints %Check that each data point is assigned to a
         cluster
        error('The number of data points in Data must be equal to the number of
            indices in IDX.')
    end
    NumOfClusters=max(IDX);
    Centers=zeros(NumOfClusters,Dimensions);
    NumOfCenters=NumOfClusters;
    NumOfPointsInCluster=zeros(NumOfClusters,1);
    for i=1:NumOfDataPoints
        Centers(IDX(i),:)=Centers(IDX(i),:)+Data(i,:);
        NumOfPointsInCluster(IDX(i))=NumOfPointsInCluster(IDX(i))+1;
    end
    for i=1:NumOfClusters
        Centers(i,:)=Centers(i,:)/NumOfPointsInCluster(i);
    end
    Colors=hsv(NumOfClusters);
case 3 %Need to calculate color scheme
    [NumOfDataPoints,Dimensions]=size(Data);
    if Dimensions~=2 && Dimensions~=3 %Check ability to plot
        error('It is only possible to plot in 2 or 3 dimensions.')
    end
    if length(IDX)~=NumOfDataPoints %Check that each data point is assigned to a
         cluster
        error('The number of data points in Data must be equal to the number of
            indices in IDX.')
    end
    NumOfClusters=max(IDX);
    [NumOfCenters,Dims]=size(Centers);
    if Dims~=Dimensions
        error('The number of dimensions in Data should be equal to the number of
             dimensions in Centers')
    end
    if NumOfCenters<NumOfClusters %Check that each cluster has a center
        error('The number of cluster centers is smaller than the number of
            clusters.')
```

```
        elseif NumOfCenters>NumOfClusters %Check that each cluster has a center
            disp('There are more centers than clusters, all will be plotted')
        end
        Colors=hsv(NumOfCenters);
    case 4 %All data is given just need to check consistency
        [NumOfDataPoints,Dimensions]=size(Data);
        if Dimensions~=2 && Dimensions~=3 %Check ability to plot
            error('It is only possible to plot in 2 or 3 dimensions.')
        end
        if length(IDX)~=NumOfDataPoints %Check that each data point is assigned to a
             cluster
            error('The number of data points in Data must be equal to the number of
                indices in IDX.')
        end
        NumOfClusters=max(IDX);
        [NumOfCenters,Dims]=size(Centers);
        if Dims~=Dimensions
            error('The number of dimensions in Data should be equal to the number of
                 dimensions in Centers')
        end
        if NumOfCenters<NumOfClusters %Check that each cluster has a center
            error('The number of cluster centers is smaller than the number of
                clusters.')
        elseif NumOfCenters>NumOfClusters %Check that each cluster has a center
            disp('There are more centers than clusters, all will be plotted')
        end
        [NumOfColors,RGB]=size(Colors);
        if RGB~=3 || NumOfColors<NumOfCenters
            error('Colors should have at least the same number of rows as number of
                clusters and 3 columns')
        end
end
%Data is ready. Now plotting
if Dimensions==2
    for i=1:NumOfCenters
        plot(Data(IDX == i,1),Data(IDX == i,2),'.','Color',Colors(i,:))
        hold on
        plot(Centers(i,1),Centers(i,2),'s','Color',Colors(i,:))
    end
else
    for i=1:NumOfCenters %plot data points
        plot3(Data(IDX == i,1),Data(IDX == i,2),Data(IDX == i,3),'.','Color',Colors(
```

```
            i,:))
        hold on
        plot3(Centers(i,1),Centers(i,2),Centers(i,3),'s','Color',Colors(i,:))
    end
end
```