**Performing Different Clustering Algorithms for Classification of Provinces of Turkey According to Their Economic Development**

***By Didem Paloğlu***



# **Performing Different Clustering Algorithms for Classification of Provinces of Turkey According to Their Economic Development**

## **Introduction**

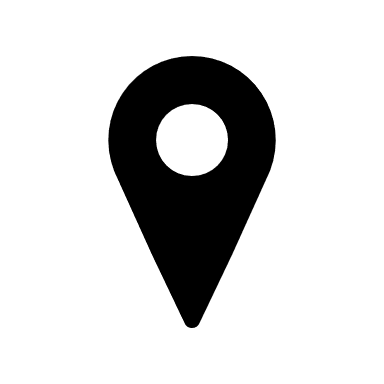
Clustering is a type of unsupervised learning and it is important to understand that clustering is used not for prediction about data sets, but is used for classify the dataset according to some feature of data. There are many clustering methods that is used to see pattern of data. Main types of clustering methods are partitional clustering (k-means and PAM), hierarchical clustering, density-based clustering and fuzzy clustering.

The main goal of this paper is to see how provinces of Turkey is grouped according to their several economic features and how similar provinces are from different regions and sizes. In this paper, PAM and hierarchical clustering methods are used and results are compared to see how different methods use different approach.

### ***Background Information about Data***

Turkey is divided into 7 regions and 81 provinces[[1]](#footnote-1) in total. Every province is listed according to their plate codes and the codes are sorted according to alphabetic order. The list of provinces with their plate codes and regions have been given in Appendix.

**Figure 1:** Turkey map with provinces



**İstanbul**

The data set including some economic indicators for each province have been taken from TÜİK (Turkish Statistical Institute) and given as a separate Excel file. The data is up to date and belongs to 2018[[2]](#footnote-2). All variables in this data set are numerical. It should be noted that İstanbul has the highest scores in total. Because it has a very special economic and geographic situation and therefore it creates its own cluster in all clustering algorithms.

The variables used in clustering algorithms were coded as follows:

**Table 1:** The variables in data set

|  |  |  |  |
| --- | --- | --- | --- |
| **Code** | **Explanation** | **Code** | **Explanation** |
| Province | The Name of the Province | Overnights | Number of Arrivals and Overnight Stays in Accommodation Facilities with Tourism Operation Certificate: Number of Overnight Stays / Total |
| GDP | GDP (Thousand $) | Net.Migration | Net Migration of the Province |
| Population | Total Population | Cars | Number of Motor Vehicles: Number of Automobiles per Thousand People |
| Export | Total Export (Thousand $) | Education | Population by Educational Level (15 Years and Older): Graduates of College or Faculty / Total |
| Agriculture | Total Cultivated Agricultural Area (Hectares) | Unemployment | Unemployment Rate (%) |

## **Methodology**

In this paper, R Programming Language have been used in order to apply two different clustering methodologies (PAM and hierarchical clustering). Related R codes have been given during the paper.

Before starting any clustering or preparing the data to clustering, there are some packages that are crucial to run the clustering algorithms. So, “cluster”, “factoextra”, “tidyverse” and “dendextend” packages have been installed.

*>- install.packages("cluster")*

*>- install.packages("factoextra")*

*>- install.packages("tidyverse") #data manipulation*

*>- install.packages("dendextend") #for comparing two dendrograms*

*>- library(cluster)*

*>- library(factoextra)*

*>- library(tidyverse)*

*>- library(dendextend)*

After installing packages, the data is ready to import. Related codes are as below:

*>- setwd("C:\\Users\\user\\Desktop")*

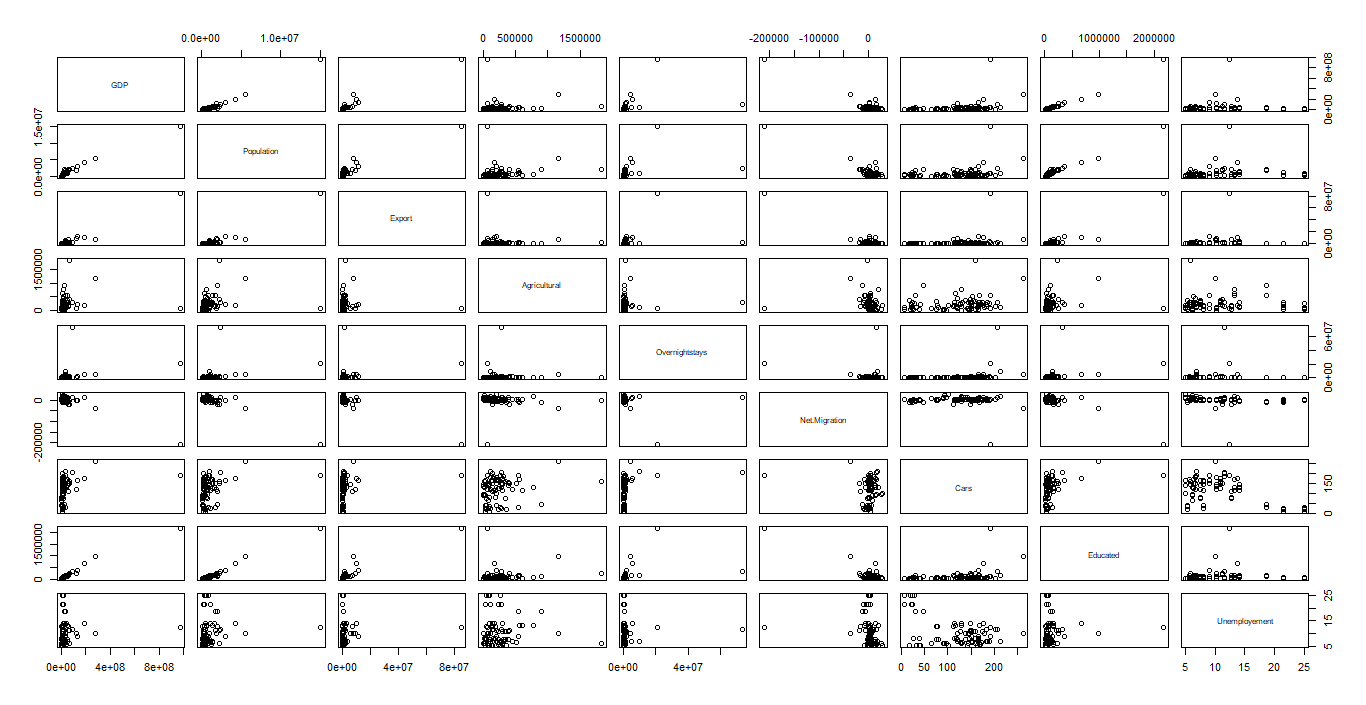
*>- mydata <- read.csv("TÜİK-Raw Data.csv",sep=";", header=T)*

*>- mydata.use <- mydata[,-1] #The province names will not be useful in cluster analysis, so it have been removed and new dataset will be used in following steps*

Now, to get an idea about data, there are some codes that summarize the data visually and statistically as below:

*>- plot(mydata.use)*

**Figure 2:** Plot of the data set



*>- head(mydata)*

Province GDP Population Export Agricultural Overnightstays Net.Migration Cars Educated

1 Istanbul 970188957 15067724 84913687 69606 20983823 -210321 192 2176161

2 Tekirdag 46964486 1029927 1288671 385637 333924 12885 139 112810

3 Edirne 12755222 411528 52244 309743 306162 1630 167 49518

4 Kirklareli 13666040 360860 179701 234251 69214 2191 167 44813

5 Balikesir 38568113 1226575 608271 298558 1206584 15210 169 155825

6 Canakkale 20107429 540662 152723 234497 850521 8405 170 74074

Unemployement

1 12.5

2 7.5

3 7.5

4 7.5

5 5.7

6 5.7

*>- colnames(mydata)*

[1] "Province" "GDP" "Population" "Export"

[5] "Flights" "Agricultural" "Overnightstays" "Net.Migration"

[9] "Flight.Passengers" "Cars" "Educated"

*>- sum(is.na(mydata))*

[1] 0

*>- summary(mydata.use)*

GDP Population Export Agricultural Overnightstays

Min. : 1812028 Min. : 82274 Min. : 208 Min. : 542 Min. : 14781

1st Qu.: 7561833 1st Qu.: 288878 1st Qu.: 52470 1st Qu.: 78930 1st Qu.: 82549

Median : 12662300 Median : 536483 Median : 197234 Median : 187356 Median : 185986

Mean : 38352306 Mean : 1012394 Mean : 2073076 Mean : 243495 Mean : 1782905

3rd Qu.: 30781062 3rd Qu.: 1029927 3rd Qu.: 741965 3rd Qu.: 298558 3rd Qu.: 540587

Max. :970188957 Max. :15067724 Max. :84913687 Max. :1833311 Max. :73689106

Net.Migration Cars Educated Unemployement

Min. :-210321 Min. : 8.0 Min. : 7914 Min. : 5.10

1st Qu.: -709 1st Qu.: 81.0 1st Qu.: 29596 1st Qu.: 6.40

Median : 2191 Median :138.0 Median : 49842 Median : 8.00

Mean : 0 Mean :122.3 Mean : 120426 Mean :10.38

3rd Qu.: 5499 3rd Qu.:166.0 3rd Qu.: 112810 3rd Qu.:12.50

Max. : 28027 Max. :262.0 Max. :2176161 Max. :25.00

Data set consists of 9 (the province names are removed) variables and each variable contribute an economic value to the related province. Before starting clustering methods, data needs to be prepared to analyze. This means that, in order to apply clustering methodology, the data has to meet some requirements. These are listed as follows:

* Rows must represent ‘observations’ and ‘columns’ should represent variables. **(It holds)**
* There must be no missing value in data. If there are any, they must be omitted or imputed. ***(It holds)***
* Data must be standardized, in other words scaled. This makes the variables comparable and scaled variable has 0 mean and 1 standard deviation. **(!)**

When we look at the variables, it can be understood that variables are at different scale. So, in order to make data standardized, we need to use scale function.

*>- means <- apply(mydata.use,2,mean) #computes means of each column*

*>- std <- apply(mydata.use,2,sd) #computes standard deviations of each column*

*>- mydata.use <- scale(mydata.use,center=means,scale=std) #standardization*

*>- head(mydata.use)*

GDP Population Export Agricultural Overnightstays Net.Migration Cars

[1,] 8.263280330 7.715912385 8.6392364 -0.63887619 2.25567074 -8.29488313 1.1939219

[2,] 0.076370532 0.009625255 -0.0818036 0.52223716 -0.17022222 0.50817355 0.2859490

[3,] -0.226988150 -0.329855381 -0.2107474 0.24339889 -0.17348362 0.06428583 0.7656328

[4,] -0.218911257 -0.357670440 -0.1974553 -0.03396241 -0.20131962 0.08641119 0.7656328

[5,] 0.001913723 0.117578515 -0.1527608 0.20230465 -0.06770454 0.59986959 0.7998959

[6,] -0.161790727 -0.258965078 -0.2002687 -0.03305859 -0.10953384 0.33148612 0.8170275

Educated Unemployement

[1,] 7.63868989 0.4025630

[2,] -0.02829917 -0.5480959

[3,] -0.26347926 -0.5480959

[4,] -0.28096208 -0.5480959

[5,] 0.13153574 -0.8903331

[6,] -0.17223420 -0.8903331

Now, after scaling the data set, it is ready for cluster analysis. The following sections apply two different clustering algorithm and compare each other.

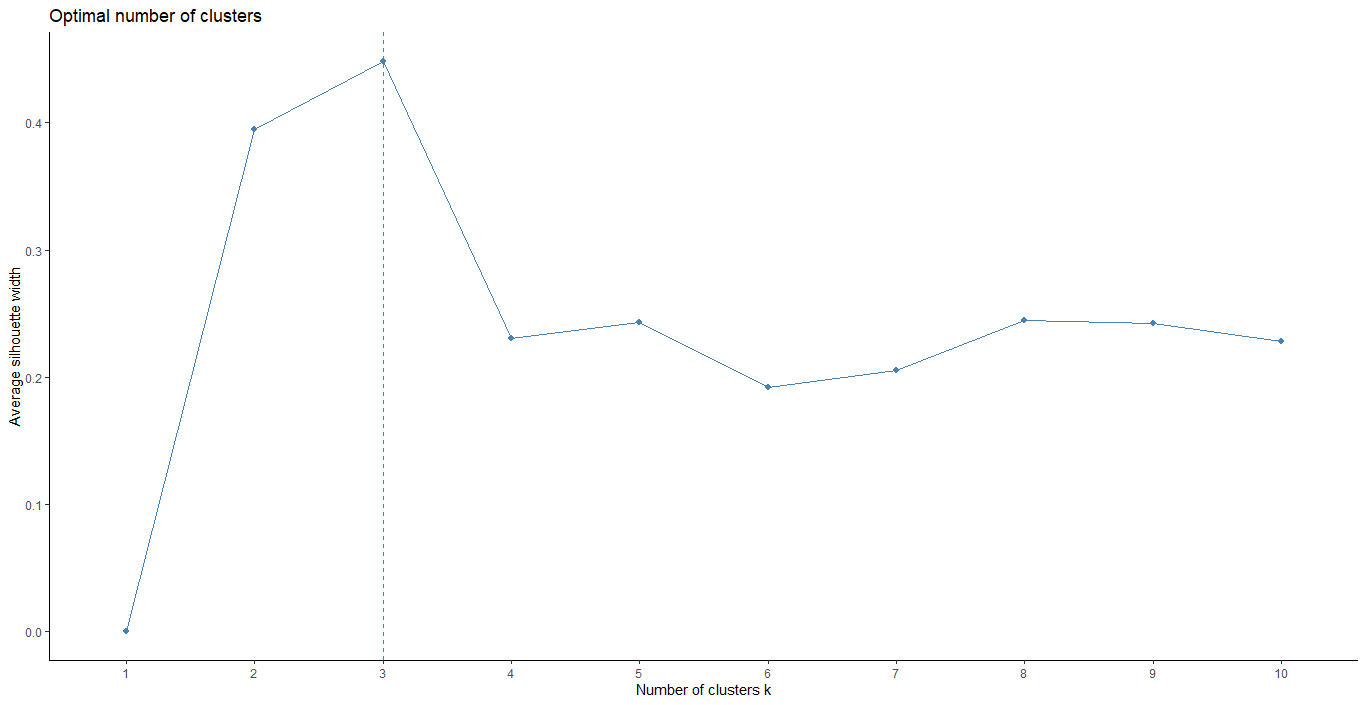
### **PAM Clustering Algorithm**

After the data is prepared to clustering, now the first method can be applied. PAM clustering, in other words k-medoids, is an adoption of k-means clustering. It is more robust than k-means because where k-means is easily affected from outliers, PAM is not sensitive to outliers and noises. In PAM algorithm, the specified observations (medoids) are selected as centers.

The results are given below:

*>- fviz\_nbclust(mydata.use, pam, method = "silhouette") + theme\_classic()*

Figure 3: Optimal number of clusters in PAM



Above graph shows the optimal number of clusters and there are 3 clusters selected. The PAM algorithm has been constructed based on 3 clusters. The summary of clustering algorithms has been applied to data set and the results are as below:

*>- mydata.pam = pam(mydata.dist,3)*

*>- print(mydata.pam)*

Medoids:

ID

[1,] 1 1

[2,] 66 66

[3,] 70 70

Clustering vector:

[1] 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2

[49] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 3 3 3 3 2 2 2 3 3 3 3 3 3

Objective function:

build swap

1.424488 1.424488

Available components:

[1] "medoids" "id.med" "clustering" "objective" "isolation" "clusinfo" "silinfo"

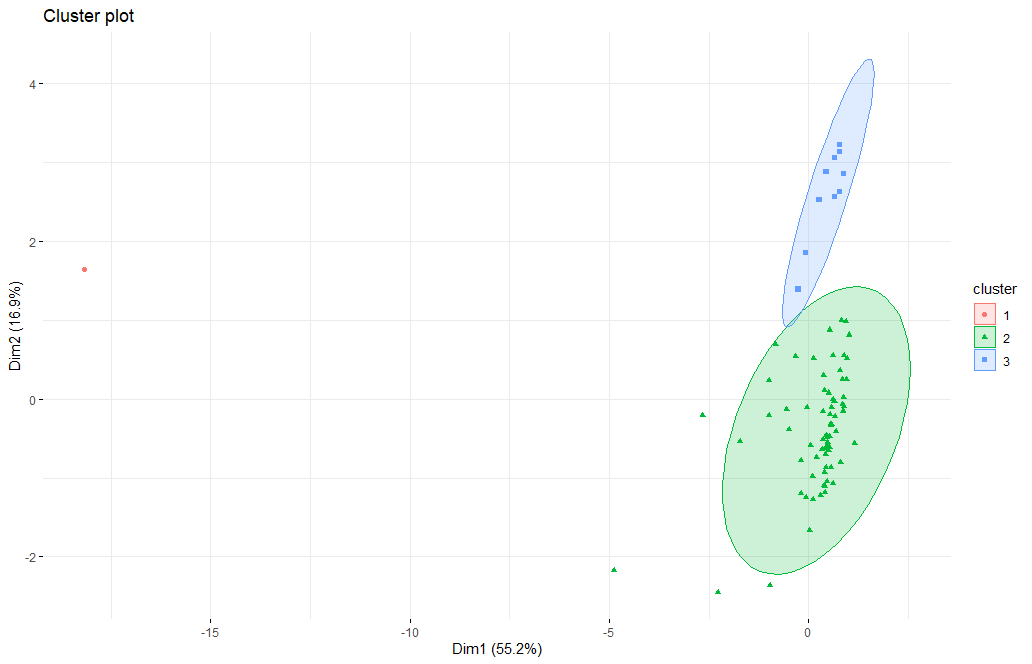
[8] "diss" "call"

After determining the number of clusters, the structure of them can be shown as graphically. As seen in graph, one observation owns its cluster. İstanbul is the reason of it. Because advanced socio-economic structure of it, it differs very much from other provinces. The elliptic visualization is as below:

*>- pam.res <- eclust(mydata.use, "pam", k = 3, graph = FALSE)*

*>- fviz\_cluster(pam.res, geom = "point", ellipse.type = "norm", ggtheme = theme\_minimal())*

**Figure 4:** Cluster plot with PAM



One can check the clustering tendency of the data. Hopkin Statistics measures how well the data can be clustered. It uses the hypothesis that:

H0: The data is uniformly distributed (i.e. there is no meaningful clustering)

H1: The data is not uniformly distributed (i.e. there is a meaningful clustering)

*>- hopkins(mydata.use, n=nrow(mydata.use)-1)*

$H

[1] 0.08078091

*>- res <- get\_clust\_tendency(mydata.use, n = nrow(mydata.use)-1, graph = FALSE) #another way to get the same result using by different code*

*>- res$hopkins\_stat*

[1] 0.08078091

In this example, 1 – H has been used as Hopkin statistics and it is equal to approximately 0.92. So, it can be said that the null hypothesis is rejected and the data has some meaningful clusters.

Another thing to check is ‘*silhouette value’.* It is a measure of how similar an object is to its own cluster and how far it is to other clusters. It takes the values between -1 and 1. If it is close to 1, it means that the observations in a cluster is well fitted. The results is as below:

*>- fviz\_silhouette(mydata.pam)*

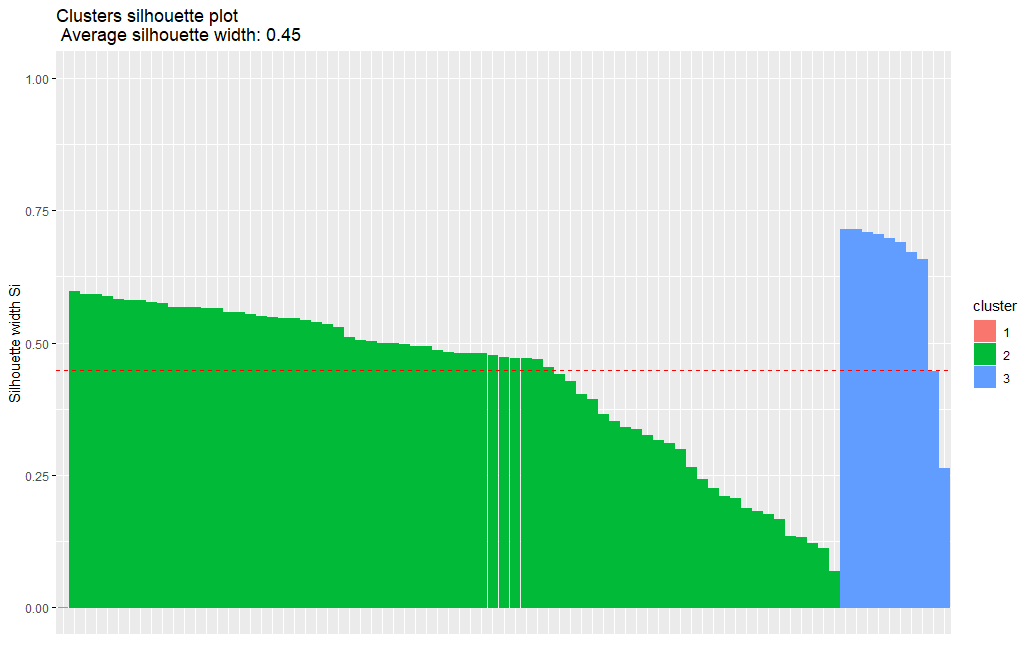
cluster size ave.sil.width

1 1 1 0.00

2 2 70 0.43

3 3 10 0.63

**Figure 5:** Clusters silhouette plot



The average silhouette value is 0.45. Third cluster is best matched cluster (silhouette = 0.63) with observations in it. First cluster has a 0-silhouette value. It is because of outlier and that cluster has only one observation.

### **Hierarchical Clustering Algorithm**

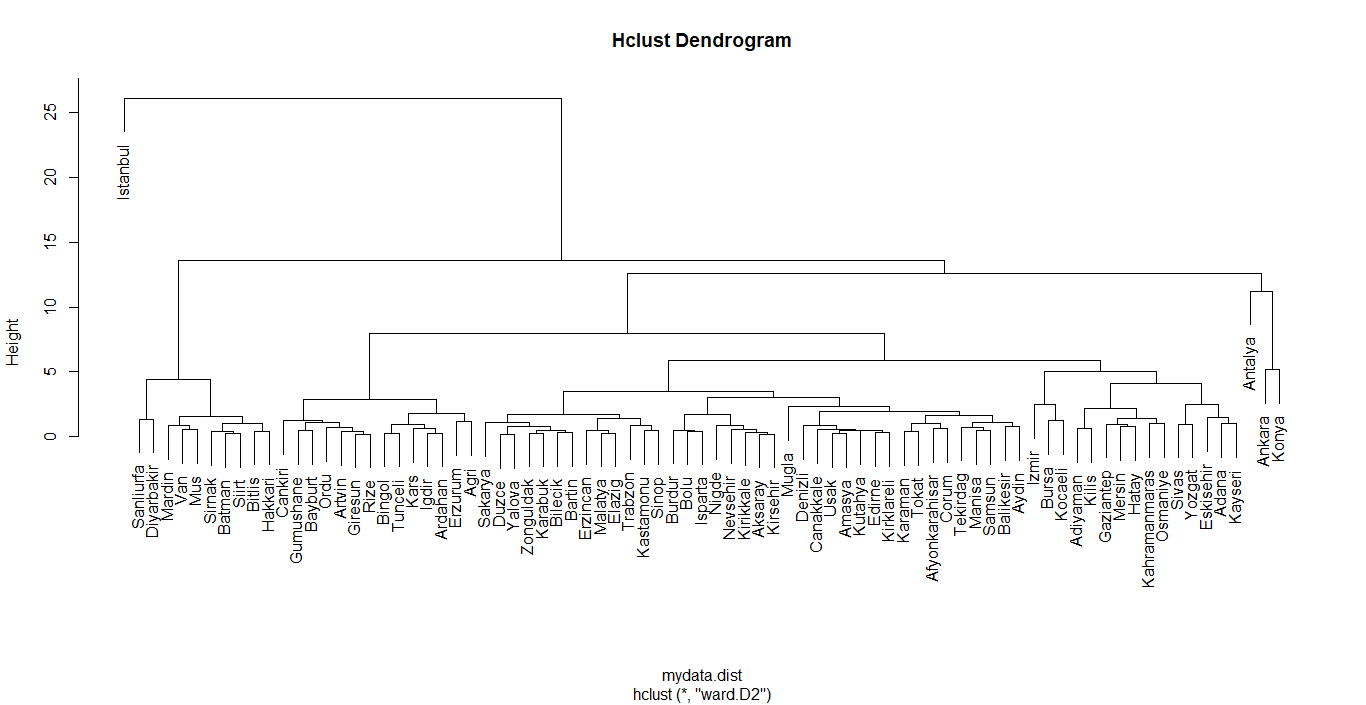
Hierarchical clustering is another simple and effective method to cluster the data. It has some advantages such as there is no need to specify the number of clusters and dendrogram produced during the algorithm is very useful visualization to understand the data. In hierarchical clustering, there are also different distance determination methods. As in k-means, “Euclidean” distance is used in hierarchical clustering.

First step in hierarchical clustering is calculating the distance matrix. Distance matrix is a nxn matrix with n observations data and (i,j)th observation of the matrix gives the distance between observations i and j. In this paper, ‘dist’ function have been used to calculate the distance matrix (there is also ‘daisy’ function for calculation of distance matrix but in this project, it has not been used).

*>- mydata.dist <- dist(mydata.use) #distance matrix*

*>- mydata.hclust <- hclust(mydata.dist, method= "ward.D2") #ward has been selected as linkage method*

*>- plot(mydata.hclust,labels= mydata$Province,main='Hclust Dendrogram') #simple plotting dendrogram*

**Figure 6:** Hierarchical Clustering Dendrogram

Another visualization method for showing dendrogram is fviz\_dend function. It is a nicer way to see the clusters in dendrograms. The code is as follow:

*>- fviz\_dend(mydata.hclust, k = 6, # Cut in six groups*

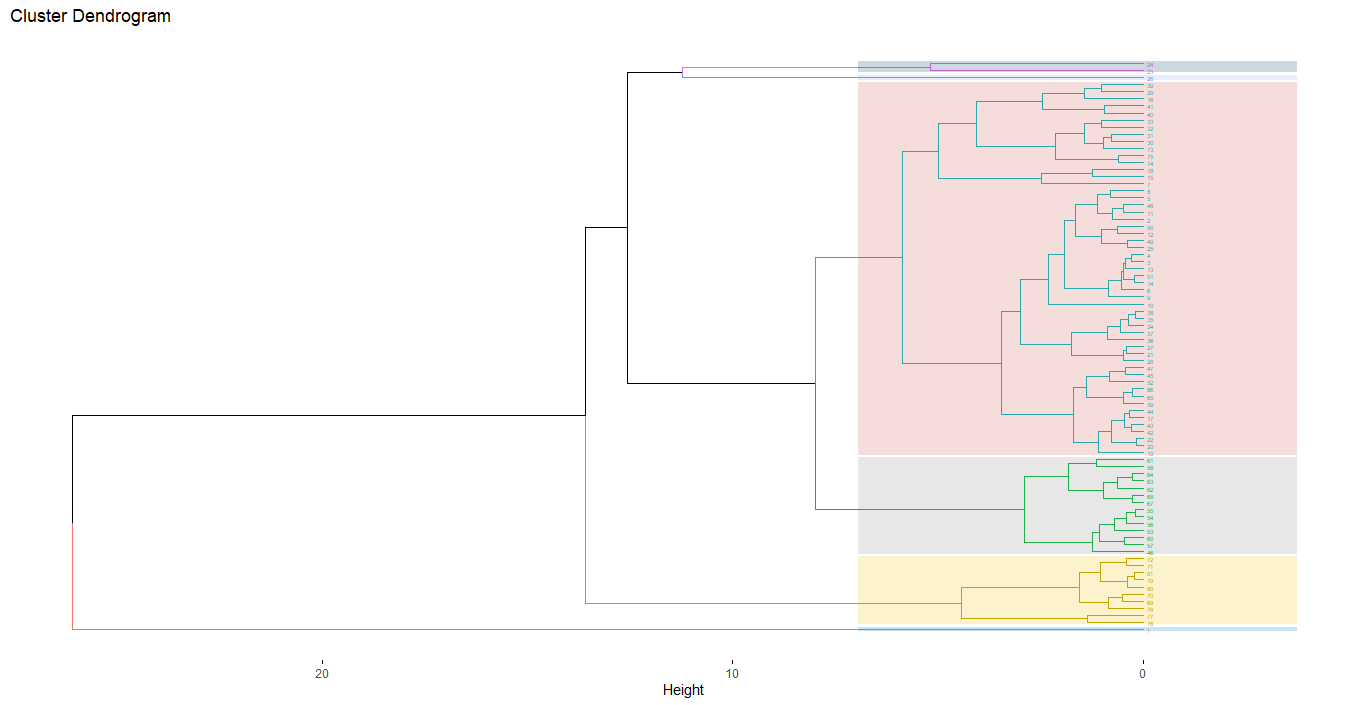
*cex = 0.3, #label size*

*horiz=TRUE,*

*color\_labels\_by\_k = TRUE,*

*, rect=TRUE, rect\_border="jco", rect\_fill = TRUE, show\_labels = TRUE)*

**Figure 7:** Hierarchical clustering horizontal dendrogram



After seeing how dendrograms look like with different plotting methods, there is one thing to be needed to explain. “ward.D2” have been used as linkage method in this clustering. This method is used because the data set has some outlier observations. Because some provinces in Turkey is very developed, such as İstanbul and Ankara, in economic way and the observations belonging them is highly different than other provinces. As a result, it creates some outliers. In this point, “ward” method helps to construct clusters with more equal sizes. The below codes compare different linkage methods. The highest score implies the more reliable method. It can be understood that ward is most suitable method for linkage as highlighted.

*>- m <- c( "average", "single", "complete", "ward")*

*>- names(m) <- c( "average", "single", "complete", "ward")*

*>- ac <- function(x) {*

*+ agnes(mydata.use, method = x)$ac*

*+ }*

*>- map\_dbl(m, ac)*

average single complete ward

0.9807417 0.9818170 0.9818260 0.9831148

To see which provinces are in which clusters, we can use the below codes. 6 clusters are selected in total and *“sapply”* function have been used in order to see each province in each clusters.

*>- groups.6 = cutree(mydata.hclust,6)*

*>- sapply(unique(groups.6),function(g)mydata$Province[groups.6 == g])*

|  |
| --- |
| **[[1]]**  [1] Istanbul  **[[2]]**  [1] Tekirdag Edirne Kirklareli Balikesir Canakkale Izmir  [7] Aydin Denizli Mugla Manisa Afyonkarahisar Kutahya  [13] Usak Bursa Eskisehir Bilecik Kocaeli Sakarya  [19] Duzce Bolu Yalova Karaman Isparta Burdur  [25] Adana Mersin Hatay Kahramanmaras Osmaniye Kirikkale  [31] Aksaray Nigde Nevsehir Kirsehir Kayseri Sivas  [37] Yozgat Zonguldak Karabuk Bartin Kastamonu Sinop  [43] Samsun Tokat Corum Amasya Trabzon Erzincan  [49] Malatya Elazig Gaziantep Adiyaman Kilis  **[[3]]**  [1] Ankara Konya  **[[4]]**  [1] Antalya  **[[5]]**  [1] Cankiri Ordu Giresun Rize Artvin Gumushane Erzurum Bayburt Agri  [10] Kars Igdir Ardahan Bingol Tunceli  **[[6]]**  [1] Van Mus Bitlis Hakkari Sanliurfa Diyarbakir Mardin Batman  [9] Sirnak Siirt |

## **Results and Interpretation**

Two clustering methods have been used in this paper. In PAM clustering algorithm, the optimal number of clusters can be determined with algorithm. 3 clusters have been selected as optimal and these clusters have been showed graphically. Hopkins statistics has been used to see how well the data is appropriate for clustering. As a result, the data shows it can be clusterable. Another measure is silhouette statistics. It has given how observations in the clusters are similar to their clusters. As a result, with 0.45 average silhoutte value, the observations are matched approximately well.

In hierarchical clustering, there has been 6 clusters. In selecting clusters, dendrograms have been very helpful and it has helped to understand the data structure. In linkage method, “ward” has been used as a result of comparing all linkage methods.

In both two methods, İstanbul has created its own cluster. It is not a surprise because all the variables that İstanbul has have extreme values rather than other provinces in Turkey. Istanbul could be omitted but I preferred not to omit it and want to see how data sets with outliers can be clustered. And the results have given the methodology section.

## **Conclusion**

This paper covers the application of different clustering algorithm to the Turkey’s provinces regarding their economic development. PAM (k-medoids) and hierarchical clustering methods have been applied and results have been interpreted. PAM method has been used because in data set, there has been some outliers and PAM is not sensitive to outliers rather than k-means. As a result of applying the PAM algorithm, there has been 3 clusters determined. This number has been determined by the algorithm itself.

In hierarchical clustering, there are 6 clusters determined but, in this time, clusters were determined according the dendrogram tree. It is not a surprise that İstanbul is outlier city because it is most developed country in Turkey and it makes the highest contribution to the Turkish economy.

With the result of two different method, it can be shown that the number of clusters differs from each other.

# **APPENDIX**

**Table 2:** Provinces of Turkey, their plate codes and regions

|  |  |  |
| --- | --- | --- |
| **Name of The Province** | **Plate Code** | **Region** |
| 01 | Adana | Mediterranean |
| 02 | Adıyaman | Southeastern Anatolia |
| 03 | Afyonkarahisar | Aegean |
| 04 | Ağrı | Eastern Anatolia |
| 05 | Amasya | Black Sea |
| 06 | Ankara | Central Anatolia |
| 07 | Antalya | Mediterranean |
| 08 | Artvin | Black Sea |
| 09 | Aydin | Aegean |
| 10 | Balıkesir | Marmara |
| 11 | Bilecik | Marmara |
| 12 | Bingöl | Eastern Anatolia |
| 13 | Bitlis | Eastern Anatolia |
| 14 | Bolu | Black Sea |
| 15 | Burdur | Mediterranean |
| 16 | Bursa | Marmara |
| 17 | Çanakkale | Marmara |
| 18 | Çankırı | Central Anatolia |
| 19 | Çorum | Black Sea |
| 20 | Denizli | Aegean |
| 21 | Diyarbakır | Southeastern Anatolia |
| 22 | Edirne | Marmara |
| 23 | Elazığ | Eastern Anatolia |
| 24 | Erzincan | Eastern Anatolia |
| 25 | Erzurum | Eastern Anatolia |
| 26 | Eskişehir | Central Anatolia |
| 27 | Gaziantep | Southeastern Anatolia |
| 28 | Giresun | Black Sea |
| 29 | Gümüşhane | Black Sea |
| 30 | Hakkâri | Eastern Anatolia |
| 31 | Hatay | Mediterranean |
| 32 | Isparta | Mediterranean |
| 33 | Mersin | Mediterranean |
| 34 | İstanbul | Marmara |
| 35 | İzmir | Aegean |
| 36 | Kars | Eastern Anatolia |
| 37 | Kastamonu | Black Sea |
| 38 | Kayseri | Central Anatolia |
| 39 | Kırklareli | Marmara |
| 40 | Kırşehir | Central Anatolia |
| 41 | Kocaeli (İzmit) | Marmara |
| 42 | Konya | Central Anatolia |
| 43 | Kütahya | Aegean |
| 44 | Malatya | Eastern Anatolia |
| 45 | Manisa | Aegean |
| 46 | Kahramanmaraş | Mediterranean |
| 47 | Mardin | Southeastern Anatolia |
| 48 | Muğla | Aegean |
| 49 | Muş | Eastern Anatolia |
| 50 | Nevşehir | Central Anatolia |
| 51 | Niğde | Central Anatolia |
| 52 | Ordu | Black Sea |
| 53 | Rize | Black Sea |
| 54 | Sakarya (Adapazarı) | Marmara |
| 55 | Samsun | Black Sea |
| 56 | Siirt | Southeastern Anatolia |
| 57 | Sinop | Black Sea |
| 58 | Sivas | Central Anatolia |
| 59 | Tekirdağ | Marmara |
| 60 | Tokat | Black Sea |
| 61 | Trabzon | Black Sea |
| 62 | Tunceli | Eastern Anatolia |
| 63 | Şanlıurfa | Southeastern Anatolia |
| 64 | Uşak | Aegean |
| 65 | Van | Eastern Anatolia |
| 66 | Yozgat | Central Anatolia |
| 67 | Zonguldak | Black Sea |
| 68 | Aksaray | Central Anatolia |
| 69 | Bayburt | Black Sea |
| 70 | Karaman | Central Anatolia |
| 71 | Kırıkkale | Central Anatolia |
| 72 | Batman | Southeastern Anatolia |
| 73 | Şirnak | Southeastern Anatolia |
| 74 | Bartın | Black Sea |
| 75 | Ardahan | Eastern Anatolia |
| 76 | Iğdır | Eastern Anatolia |
| 77 | Yalova | Marmara |
| 78 | Karabük | Black Sea |
| 79 | Kilis | Southeastern Anatolia |
| 80 | Osmaniye | Mediterranean |
| 81 | Düzce | Black Sea |

1. Names of the provinces, their plate codes and their regions are given in Appendix. [↑](#footnote-ref-1)
2. Only GDP variable belongs to 2017, since 2018 data of “GDP of provinces” has not published yet. [↑](#footnote-ref-2)