Cluster median problem

Optimization in Data Science

First Semester 2020
Andreu Meca Allende
Jan Leyva Massagué

INDEX

1. INTRODUCTION

This project is a comparison of the performance in terms of the optimal solution and time of three different models to cluster three different data sets. The aim of this project is to see the difference between an optimal model and a heuristic one. To do that, the optimal-solution algorithm used is a Cluster-Median, and the two heuristic ones are a Minimum Spanning Tree and a K-Means.

Clustering is the process of grouping a set of observations based on the similarity of the variables. Each group is called a cluster and, in this project, the total number of clusters is represented by the letter *k*.

The three data sets used have three different dimensions to see how the algorithms perform with different dimensions. The data is from the Open Data Barcelona and Kaggle web pages.

The hardware used in the project is a 2,2 GHz 6-Core Intel Core i7 with 16GB 2400MHz DDR4 of memory and MacOS Catalina Version 10.15.7.

2. DATA DESCRIPTION

On this project are used three different data sets, each one with a different size in order to check how Cluster-median, Minimum Cost Spanning Tree and K-means algorithm perform.

The first data set used is a two-dimensional variable which is composed by the location in terms of coordinates of almost all Green Points in Barcelona. Green Points are a network of environmental facilities to get rid of waste that can't be thrown into street containers. This data set consists of 25 observations with the longitude and latitude of each of these Green Points. The source of this data set is Open Data Barcelona.

The second largest data set is a three-dimensional variable which is composed of the average tax income classified by latitude and longitude. It consists of 73 observations corresponding to each neighborhoods in Barcelona. There are three variables, two of them are the coordinates (longitude and latitude) and the third one is the average tax income. The source of this dataset is Open Data Barcelona.

The largest data set is a seven-dimensional variable with 163 observations consisting of almost all countries in the world. The variables of the data set are child mortality, exports, imports, health expenditure, net import by person, inflation, life expectancy and the number of children that would be born per woman if the current age-fertility rates and GBP per capita remain the same. The source of this dataset is Kaggle

3. CONCEPTUAL FRAMEWORK

On this project, three models are performed. An integer one, the Cluster-median, and two heuristics, the Minimum Spanning Tree (MST) and K-Means. The first one is developed in Ampl and the two remaining in R.

3.1. Optimal cluster-median problem

The model will consider *m* clusters, although only *k* clusters are needed. Such that *m-k* clusters will be empty. The cluster that has as median the element j will be denoted "*cluster-j*".
The variables of the formulation are:
∀ i, j =1,…,m

$$x_{ij} = \{1 \;\; 0\}$$

If the element I belong to *cluster-j*, then in the solutions matrix will be 1. Otherwise will be a 0.

The model made *k* and only *k* clusters, every element of the data set belongs to one *cluster-j* and it made with the optimal result, every point with the medium total l_1 distance between points and the median.

The model definition is given by the following cost function and constraints:

$$min \sum_{i=1}^{m} \sum_{i=1}^{m} d_{ij}x_{ij} \qquad [Distance\ of\ all\ points\ to\ their\ medians]$$

$$Subject\ to \sum_{j=1}^{m} x_{ij} = 1 \quad i = 1, \dots, m \quad [Every\ point\ belongs\ to\ one\ cluster]$$

$$\sum_{j=1}^{m} x_{jj} = k \qquad [Exactly\ k\ clusters]$$

$$x_{jj} \geq x_{ij} \quad i, j = 1, \dots, m\ [A\ point\ belongs\ to\ cluster\ j\ if\ the\ cluster\ j\ exist]$$

On the project is developed the model with Ampl, introducing as input the Euclidean distance matrix of the data, the variables are not normalized. Even so, Euclidean distance matrix is used because the purpose of this project is to compare the three different models and their performance.
The distance matrix is recorded in the *.da*t files *(Cluster_Median_small.dat, Cluster_Median_medium.dat and Cluster_Median_large.dat*).
The model is done with the previous definition shown adapted to the software Ampl. This is recorded on the file *Cluster_Median.mod*, that will be used with the three different datasets. Also, there are three files *.run* each one for the dataset small, medium and large (with the same structure as the .dat files). The file run is used to call the model and data, as well as to choose the cplex solver, display the matrix results and the time needed to solve it. cplex solver is used because it has a better performance in integer problems.

3.2. K-means (heuristic algorithm)

The aim of the algorithm is to partition *n* observations into *k* clusters, which each observation belongs to the nearest cluster mean (centroid), such that the sum of squares from point to the assigned cluster center is minimized. The centroid is given randomly, this is one of the weaknesses of this algorithm because it depends on the centroid to do a good performance. To solve that is recommended to do more than one iteration and then choose the result with less variance. The steps will be the following:
Choose *k* initial means $\underline{x}^i = c_i \; i = 1, \dots, k$, for each cluster
Repeat
Assignment step: Assign each point to the cluster of closest mean we obtain partition $\{\{S_1, \dots, S_k\}$
Update step: compute the new means $\underline{x}^i = \frac{\sum_{j \in S_i} \; x^i}{|S_i|}$
Until the partition $\{S_i, \dots, S_k\}$ does not change.

As said before, K-means is computed with R using the function *Kmeans* that perform the algorithm. As input is given the data, number of *k* clusters and the number of iterations that choose a centroid to do a better performance.
On the R file contains the import of the three datasets, the algorithm k-means, the function to get the Euclidean distance matrix for cluster-median and at the end the plots to show the results.

3.3. Minimum Spanning Tree

The aim of the algorithm is to find a spanning tree of minimum cost. Once is found the MST that connects all the points without making a cycle is the moment to cut the *k-1* costly edges to get *k* clusters. This will guarantee a locally optimal solution to the clustering problem.

On this project is used the function *gclust* to get the MST and the function *cutree* in order to cut the costly edges and get the *k* clusters. The function used Kruskal Greedy algorithm. The steps are detailed below:
Order the arcs in non-decreasing order of cost.
T = 0
For I = 1 to m do
    If T + a does not have a cycle then
        T:=T+a_i
    end if;
end for
This algorithm satisfy optimality conditions

4. RESULTS

Once the models and the datasets are chosen, the next thing to do when solving a cluster problem is to decide the number of k clusters is the best for the data. To do that, in this project the number of clusters is decided on the error each k has based on a K-Means algorithm. A visualization of the errors can be seen in the figures 1, 2, and 3.

In order to reduce the error per number of clusters, the k cluster will be chosen differently for each data set. For the small data set will be used k=2, and for the two-remaining k=4. This decision is taken by the figures shown below.

In this project the 3 algorithms will be compared in a visual analysis first and then a time-cost comparison.

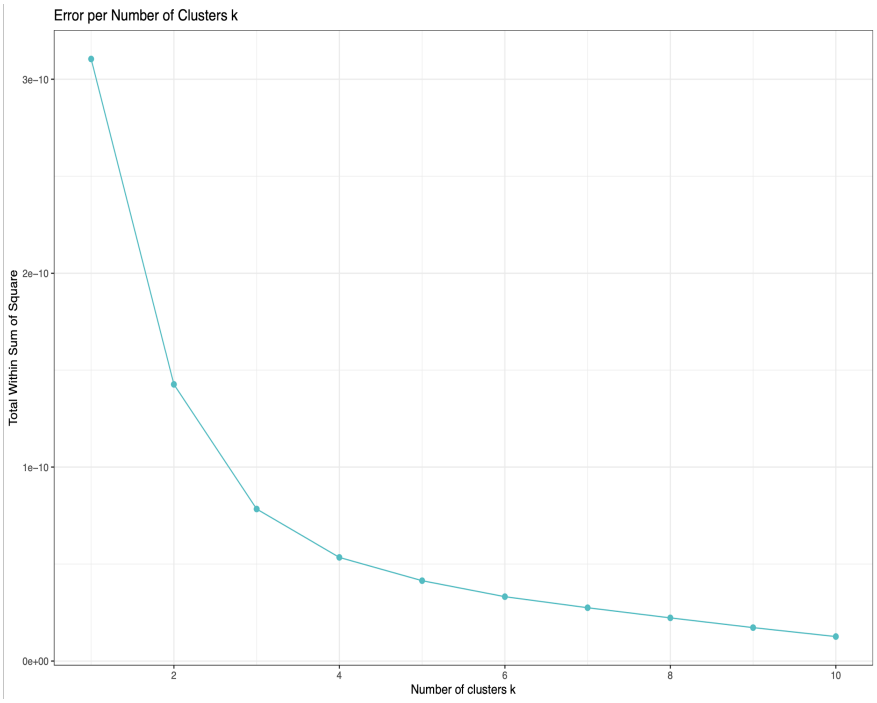

Figure 1. Error per number of Clusters k generated. Green Points by Coordinates in Barcelona dataset

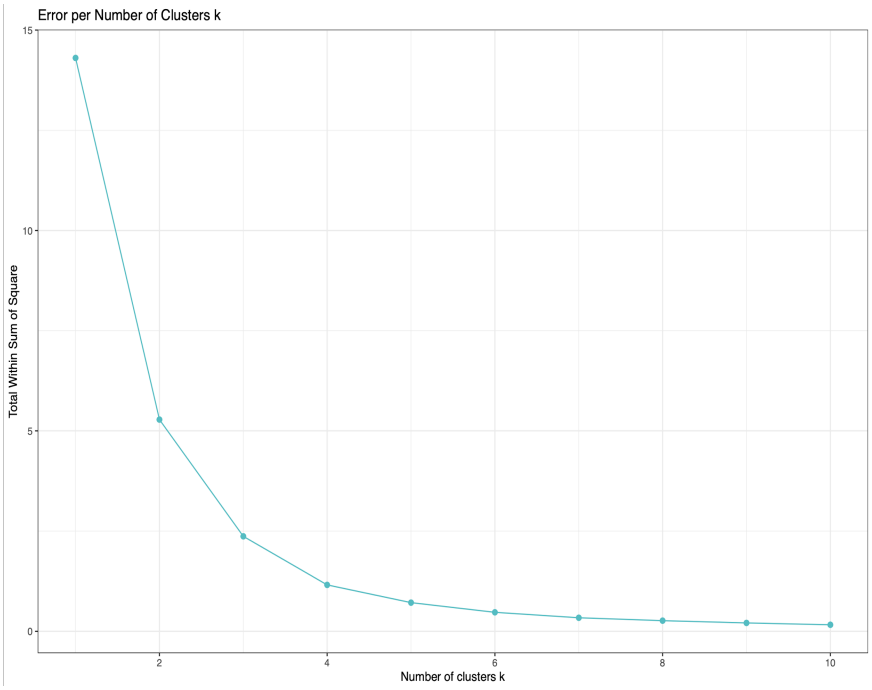

Figure 2. Error per number of Clusters k generated. Average Tax Income in Barcelona by Coordinates Dataset

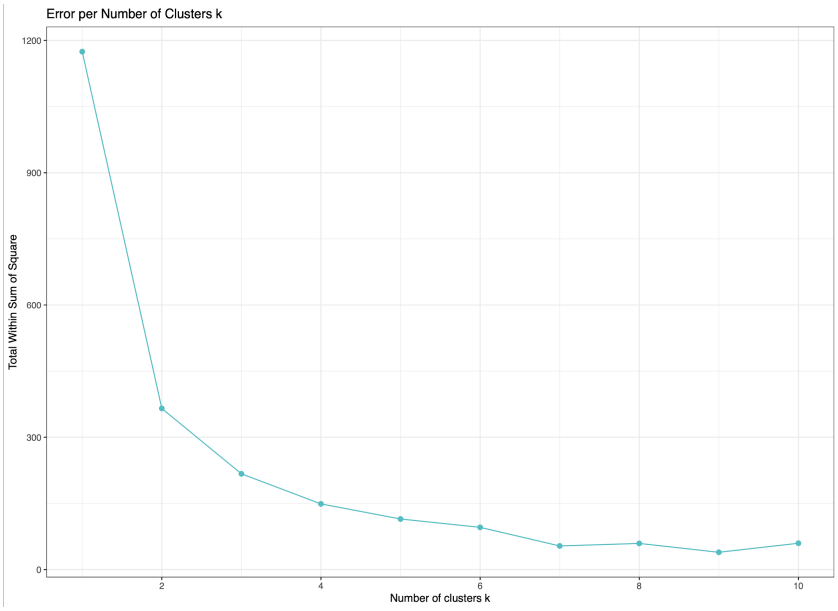

Figure 3. Error per number of Clusters k generated. Country Statistics Dataset

In the visual analysis comparison, based on figures from 4 to 9, it is possible to see a similarity between the three algorithms, but MST on the small data-set the clustering is a bit different. This is can be explained because the data have more weight on the points between 41.375 and 41.400 latitude and 2.1 and 2.175 longitude, and Cluster-Median and K-means perform really similar as one cluster, but MST finds the distance (cost between nodes) more costly and clustered the right side and left side as the two clusters. Here is possible to see the different performance between this algorithms, how the MST considers more decisive the cost of the edges and Cluster-Median and K-means minimize the distance in each cluster.

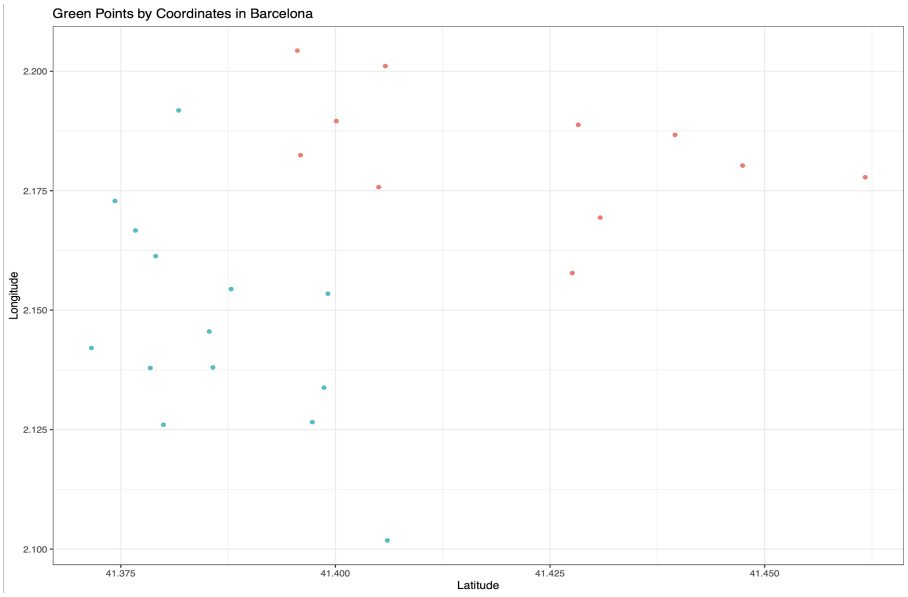

Figure 4. Cluster-Median Cluster Classification. Green Points by Coordinates in Barcelona dataset

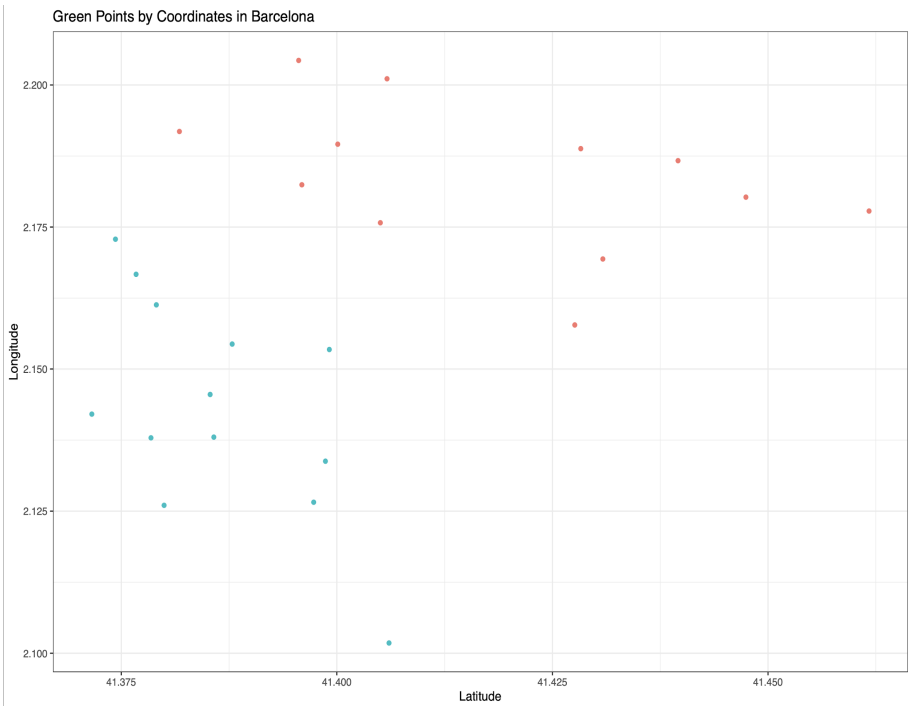

Figure 5. K-Means Cluster Classification. Green Points by Coordinates in Barcelona dataset

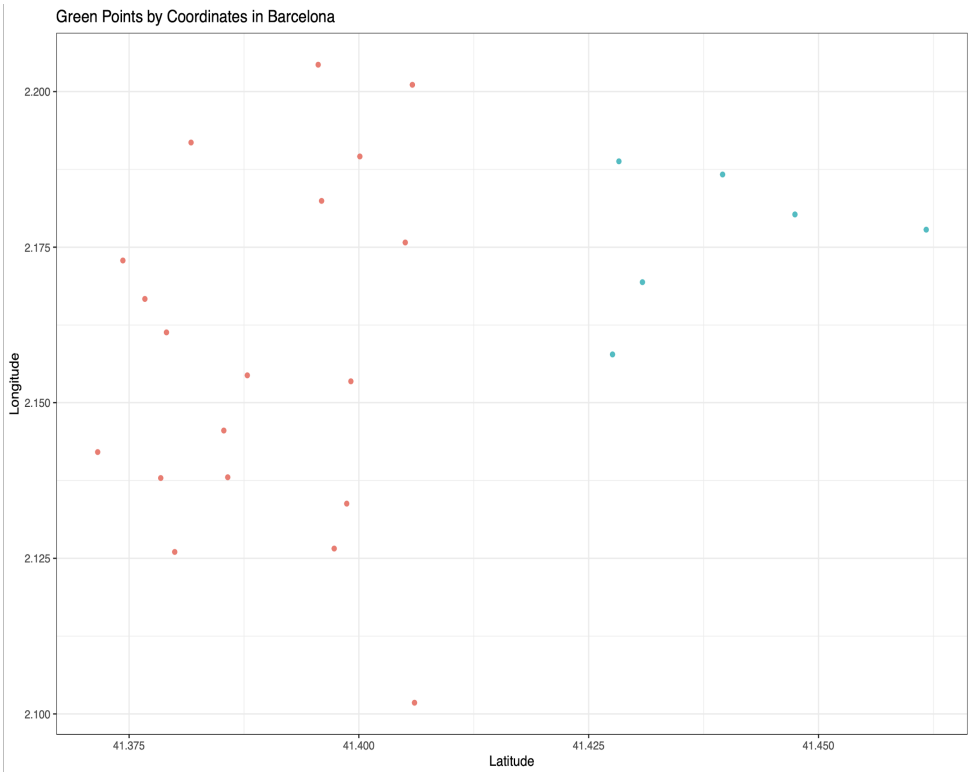

Figure 6. Minimum Spanning Tree Cluster Classification. Green Points by Coordinates in Barcelona dataset

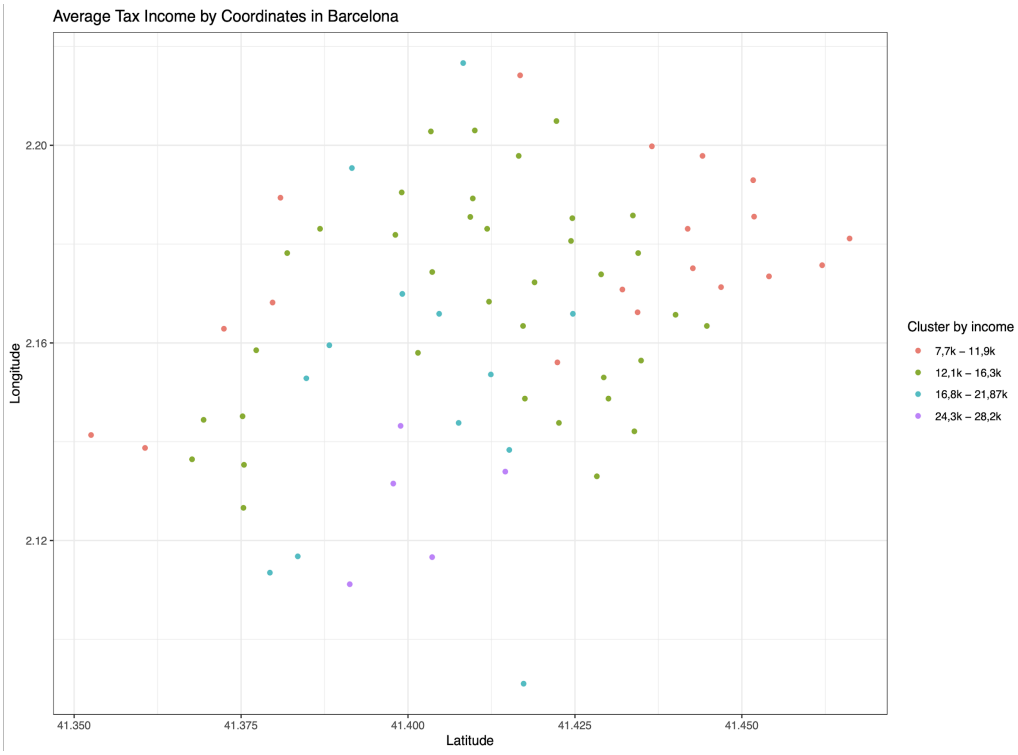

Figure 7. Cluster-Median Cluster Classification. Average Tax Income by Coordinates in Barcelona dataset

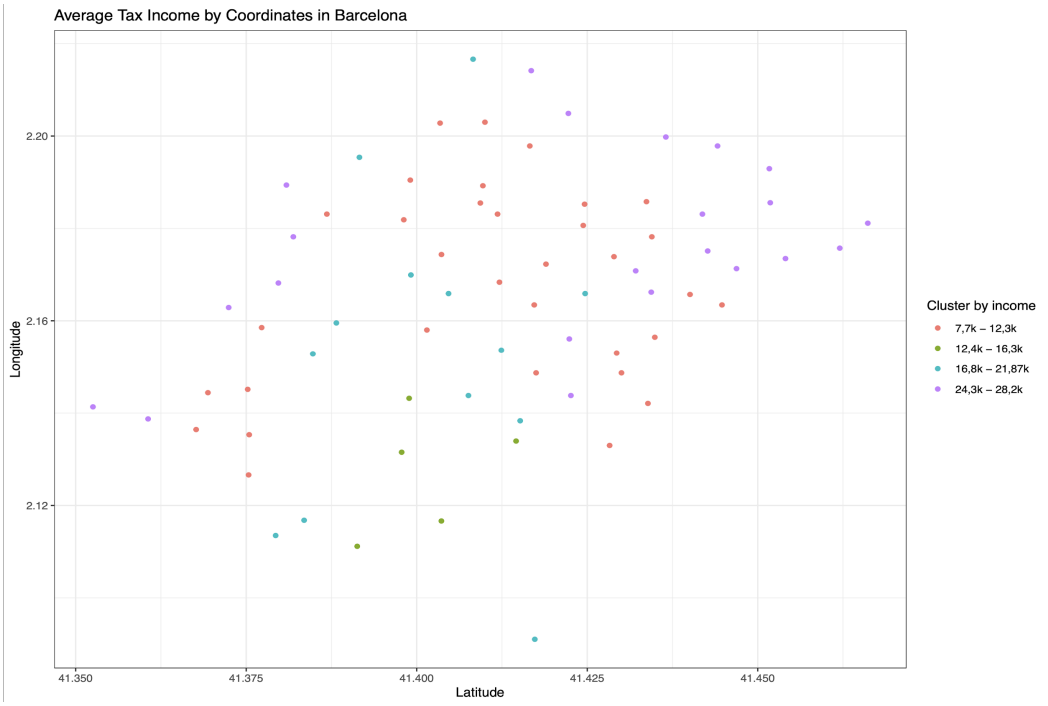

Figure 8. K-Means Cluster Classification. Average Tax Income by Coordinates in Barcelona dataset

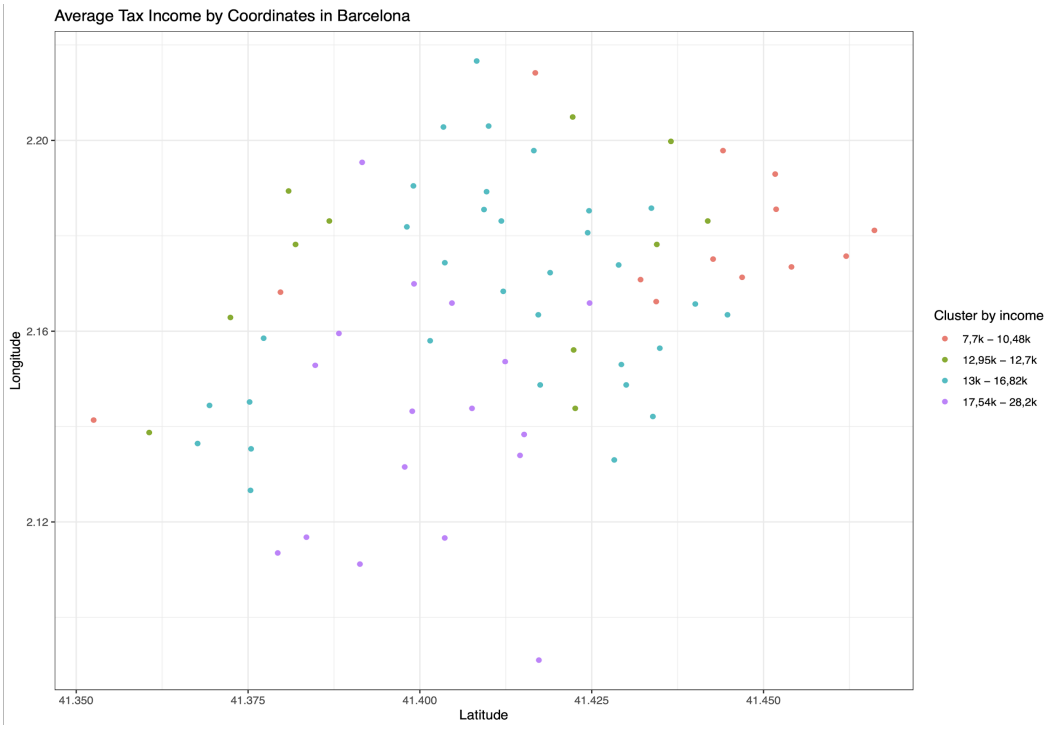

Figure 9. Minimum Spanning Tree Cluster Classification. Average Tax Income by Coordinates in Barcelona dataset

The second part of the comparison is to compute and compare the time each algorithm took to compute the clusters per each dataset, which is documented in figure 10. As expected, the optimal algorithm Cluster-Median took much longer than the other two algorithms, being the less expensive the K-Means algorithm. Also is possible to conclude how Cluster-Median as long as the *m* of the data set increase the time needed to solve the problem is higher, it is one of the weaknesses of this model because with a large data-set it will need a huge time and this time increase faster than with other two algorithms. MST, have a moderate increase as long as *m* increases and K-Means need a similar time to solve three clustering problems.

	Cluster-Median	**MST**	**K-Means**	**Total Average**
Green Points in Barcelona	0,03726	0,00379	0,00069	0,01391
Average Tax Income by Coordinates in Barcelona	0,38474	0,00823	0,00058	0,13118
Country Statistics	3,30104	0,01776	0,00071	1,10650
Total Average	1,24101	0,00993	0,00066	

Figure 10. Table with the time each algorithm took to compute the clusters per each dataset

In average, the Cluster-Median took 80 times more than the MST and 1771 more than the K-Means. As bigger the dataset is, the bigger the difference in time is between the optimal and the heuristic models.

5. CONCLUSIONS

This project confirms that the optimal model requires a much higher computational time to obtain the clusters for the same dataset than the heuristic models.

Also, it is possible to hypothesize that the K-Means algorithm is one of the most used algorithms in the data science world due to its low computational time and similar results as an optimal model such as the Cluster-Median. Knowing his weakness that needs more than one iteration to get a good clustering, but the time cost is less than performing the integer solutions.
On the project is possible to see how K-means needed similar time to solve three data-sets compared to Cluster-Median which increases almost 80 times his time to solve the largest data-set and MST needed more time too.

To sum up, Cluster-Median has a good performance clustering not large data-sets and it will be good to be used on these cases. But as long as the data set is large there are other algorithms that compute a similar solution with less time cost, such as MST and K-Means. Depending on the problem to solve, the decision will be MST for these problems that cost between nodes is the aim of the problem such as telecommunications problems or K-Means for the largest data-set.