Máster Universitario en

Tratamiento Estadístico-Computacional

de la Información (TECI)

Curso 2013-2014

Trabajo Fin de Máster

"*Experimental design to measure the impact of online advertising on the sales of a car manufacturer's dealer network*"

AUTOR: **Juan José Carín Mansilla**

TUTORA: **Macarena Estévez (CONENTO)**

PONENTE: **María del Mar Fenoy Muñoz (EIO-UCM)**

Francisco Ballesteros Olmo (DMAT-UPM)

Table of Contents

1. Problem statement, Objective and Resources

1.1. Problem statement

> *Imagine that an advert of a store is published in a local newspaper, or broadcasted on radio. There are certain people in the city interested in the kind of products sold at that store: some do buy at that store, and others do not. Not all buyers were exposed to the ad, and some non-buyers were. So what are the differences between all these people? If a buyer were not exposed to the ad, would he or she still buy at that store? What would happen if both types of ads were used? What if the radio ad were broadcasted more frequently?*

Advertisers need to know the effectiveness of advertising—e.g., how and how many people are influenced by it—, in order to plan and optimize their corresponding budget—i.e., how and how much they want to spend.

This project is focused on online advertising, through YouTube (owned by Google), although what is mentioned above applies to any kind of advertising, through any type of platform.

And it is also focused on offline sales, for two reasons:

1. The impact of online advertising on online sales is relatively easy to quantify, as it is possible to track down the visits to a website—and also the phone calls received, in the case of mobile search advertising—coming from Google. Anyway this kind of SEM activity is likely to drive also in-store sales: those potential customers who prefer to buy from a physical store (even if they have called the retailer or visited its website) may end up buying influenced by the online ads they were exposed to.
2. Moreover, some retailers like the one under study in this project, a car manufacturer—and its associated dealer network—, still do not offer their products online.

Therefore, the question is how many of a retailer's (offline) customers were exposed to, and influenced by advertising (in YouTube), and how this can be measured.

The answer to this question is important, not only to advertisers, but also to Google (and similar companies): YouTube is currently the third most visited website on the Internet—behind Google and Facebook—, so all the agents involved would like to know what is the impact of advertising in it, not only in terms of building brand awareness, but also of sales uplift.

For this reason, Google Spain proposed an experiment to one of its clients, a car manufacturer, which was to be developed by Conento, a consulting firm specializing in marketing analytics.

This capstone project covers part of this experiment, namely its first two phases, as described in Chapter 2, Subsections 2.2.1 and 2.2.2.

1.2. Objective

As previously mentioned, the main goal of this Master's Capstone Project is **to measure the impact of online advertising on the offline sales of a retailer**.

Impact is defined as the sales uplift attributable only to online advertising campaigns. The retailer under study is **a car manufacturer's dealer network**, and the advertising whose impact is to be measured is the one made through YouTube, and it is focused on the best-selling car model of this manufacturer.

For simplicity, hereafter we will refer to this specific model as **MUS** (Model Under Study).

1.3. Resources

In order to achieve the mentioned objective, data on multiple variables are required. These variables are listed and described in Chapter 2, Section 2.3 (Methodology – Variables).

A statistical software or programming language is also required to analyze the data. The chosen tool is R (version 3.1.0), a free software environment for statistical computing and graphics. Additional programs like IBM SPSS version 19, QGIS version 2.2 and Microsoft Office 2013 pack were also used.

The equipment used was a laptop with an Intel® Core™ i5 processor at 1.8 GHz, 8 GB RAM, running 64-bit Windows 8.1 operating system.

2. Methodology

2.1. Introduction

Google is a U.S. multinational corporation specializing in Internet-related services and products. These include search, cloud computing, software, and online advertising technologies. Most of its profits are derived from AdWords, which is the online advertising service that, for instance, places advertising copy at the top or bottom of, or beside, the list of results Google displays for a particular search query. The choice and placement of the ads is based in part on a proprietary determination of the relevance of the search query to the advertising copy.

YouTube, which is a video-sharing website created in February 2005 that allows users to upload, view, and share videos, has been owned by Google since late 2006. As part of AdWords' services, it is possible for advertisers to place pre-roll advertisements—ones that are shown before the searched video starts—on YouTube. Those adverts will only be displayed to potential customers, based on their profile, likes, location, etc., and the advertisers only pay when someone chooses to watch their adverts.

Many methods have been used to address the need to measure the effectiveness of advertising, but one of the most rigorous and trusted of these is based in randomized experiments, which involve randomly assigning experimental units to Test and Control conditions. This is the approach selected for this experiment, though the Test and Control cells are created based on their similarity, rather than randomly.

Google AdWords allows advertising to be targeted by geographical region. By assigning Geographic Marketing Areas (**GMAs**) to test or control conditions, it is possible to employ AdWords' geo-targeted advertising capabilities to increase or decrease the regional advertising spend accordingly.

Though strictly speaking the term GMA usually refers to a much broader area—e.g., Google has 46 GMAs in Spain, which cover its whole territory—, for the purpose of this project a GMA will refer to the unit used in this experiment: a city. I.e., a GMA will be a city where it is possible for Google to launch a geo-targeted advertising campaign in YouTube. Based on this definition, there were 443 GMAs in Spain by the time this project was started.

Hence, the first goal is to analyze the GMAs (and the car dealers' stores located in them), in order to select the largest possible number of them, and to divide that group into two comparable subgroups—Test and Control—, as similar as possible to each other. Once these groups have been determined, the retailer shall increase the video ads spend in the GMAs belonging to the Test group, compared to the level of spend in the Control group, during a certain test period. By measuring the difference in sales of stores located in each group, a

metric of the impact of online advertising would be obtained.

2.2. Main phases

The main phases of the whole experiment are describe below. As previously mentioned, this capstone project covers the first two phases, which are fully described in Chapter 3.

2.2.1. Matching the car dealers' stores with the GMAs

Only the car dealers' stores which are located in a GMA (and hence the ones that will clearly benefit from a geo-targeted advertising campaign in those GMAs[1]) will be analyzed.

A car dealer—i.e., a company that sells cars from one or more manufacturers—may have several stores or Points of Sales (**POSs**). As it is further explained in Chapter 3, Section 3.1 (Project Phases – Data cleansing and Sample selection), in case a certain dealer have POSs located in different cities—whether or not GMAs—, that dealer should not be considered as all the relevant variables (e.g., sales and visits) are measured on a dealer basis, not considering the specific POS. Therefore, the GMAs of interest may contain one car dealer (and hence at least one POS) or more, but all of the POSs of each car dealer must be located in that GMA.

For instance, let's suppose that there are only 3 cities (*A*, *B* and *C*) and 3 car dealers (*X*, *Y* and *Z*) in the area under study. If *X* has one store in *A*, *Y* has two stores, one in *B* and one in *C*, and *Z* has two stores, both in *C*, and only cities *B* and C are GMAs, then only the GMA *C*—and the dealer *Z*—could be considered, as the store of *X* is not located in a GMA, and *Y* has stores in more than one city (and only the total amount of sales is known, not how many of them were closed in *B* or *C*).

2.2.2. Definition of the Test and Control groups

In order to assign each GMA (and the valid POSs it may contain) to a group, different variables (see Chapter 2, Section 2.3, for further details) must be analyzed. Data on these variables should cover a (pre-test) period of at least 1 year, ideally 2 years.

Both groups—Test and Control—must have the same size (in terms of GMAs) and be as similar as possible. In order to define a measure of similarity a specific metric must be created.

[1] Of course, people who watch a YouTube advert while they are in a GMA of interest, may visit a car dealer's store in a nearby city (it is a very common practice in this market to compare between different dealers), but any other city that is not a GMA shall not be considered, as it is not possible to know the number of those people.

2.2.3. Proposal of increase in advertising spend in the Test group

The increase in advertising spend—during a test period of 2 to 3 weeks—needed to achieve a significant enough sales uplift in the GMAs in the Test group must be calculated. That increase was based on an econometric model, which is outside of the scope of this project.

2.2.4. Test period

Based in the previous proposal, the advertising spend during the test period will be increased in the GMAs belonging to the Test group, and it will remain equal in the GMAs belonging to the Control group. As YouTube campaigns are not usually conducted on a permanent basis (at least in the case of the car manufacturer under study), the advertising spend in any GMA rather than the ones belonging to the Test group will be null.

By the time this project was finished, and as the car manufacturer wants the YouTube campaign to run simultaneously with another one on TV, scheduled in October, 2014, the test period has not taken place, and hence there are no real values to be analyzed.

2.2.5. Analysis of results

The increased exposure to advertising of customers inside the GMAs belonging to the Test group may result in an increase in sales *during* the test period and also *immediately after*. Therefore, sales in both groups should be analyzed not only during the test period but also until sales in the Test group stabilize to previous levels. Sales in Test group could also be compared to the forecasted values in the absence of that stimulus[2].

2.3. Variables

In order to solve the mentioned problem, data on multiple variables are considered—at least initially. We could classify them according to their nature (static[3] and dynamic—i.e., time series—) and source (Google, the car manufacturer, and others). All of these variables will be analyzed during the pre-test period, but only the dynamic ones need to be measured also

[2] Although the two groups have been designed to be as comparable as possible, the difference in sales between them may not be completely attributable to difference in advertising exposure.

[3] Of course, no variable should be considered strictly static. Variables like the population of each GMA are indeed dynamic ones, but their variability over time is relatively small, and hence are measured only once or twice a year. The location of the POSs is fixed, at least during the last part of the pre-test period, and the whole test period.

during the test period, in order to analyze the results of the experiment. The test period should cover the 2 or 3 weeks during which the advertising spend is increased in the Test group, plus a few additional days, to consider the whole sales uplift due to (additional) exposure to advertising.

The pre-test period covers 109 weeks, from week 18, 2012 to week 22, 2014 (i.e., from April 30, 2012, to June 1, 2014).

> As this test period—i.e., the YouTube campaign—was postponed, the analysis that has been made should be extended and improved, also considering the weeks between the original pre-test period and the test period.

The static variables were measured at the end of the pre-test period (or as close to that time as possible, which depended on the source), and the dynamic ones were measured on a weekly basis (thus, we have 109 observations of each one). All of them are segmented by city, which is the subject or experimental unit of our study, except those provided by the car manufacturer, that are segmented by dealer (but not by POS). As a dealer may have POSs located in different cities, it is not possible in those cases to link the data from the car manufacturer to a city, and hence some dealers cannot be included in our analysis.

2.3.1. Dynamic variables

For the MUS as well as for all the car manufacturer's models, except where otherwise indicated:

- From the car manufacturer (segmented by dealer rather than by city):
  - Registrations
  - Sales
  - Orders
  - Cancelled orders

    In order to achieve their quota or keep their stock, it is a common practice that dealers artificially increase these numbers at the end of the month—and then offset these increases, by cancelling orders, keeping a stock of registered vehicles, etc.—. Therefore, none of the variables above is a perfect indicator of a dealer's success over time.

  - **Traffic**: number of potential customers' visits to a POS of the dealer. This is the variable that best reflects the success of an advertising campaign, and hence the target variable (and whenever we refer to sales uplift it must be understood as traffic uplift instead).
  - Tests

- From the car manufacturer's media agency (for more than 5,000 cities of Spain):
  - Visits to the website (of the MUS or the car manufacturer's brand), depending on their origin:
    - Total
    - Direct
    - SEA
    - SEO
    - **Display**: this is our origin of most interest, as it is the one that includes video adverts.

      The car manufacturer's media agency started to measure visits coming from Video ads separately 2 weeks before the end of the pre-test period. Therefore, data for those 2 weeks were added to the Display values in that period.

    - Referrals
    - Affiliation
    - Emailing
    - Social media
  - Visits to the website of the MUS (hence, only for it, and not for all the car manufacturer's models), where its name was a keyword, based on their origin:
    - SEM
    - SEO

As mentioned above, the most important variables are **Traffic** and **Display Visits**. The main expected effect of this experiment will be an increase of Visits—the explanatory variable—, which should be linked to an increase in Traffic—the dependent or response variable. By measuring how a change in the explanatory variable affects the Traffic, we can estimate the effect of advertising, i.e., at what extent it affects the variable most related to car manufacturer's success.

So both terms, Traffic and Visits, will be often mentioned along this document. It is important to keep in mind, to avoid any confusion, and since each term could also be used in the other context, that here **Traffic** refers to the number of visits to a POS, while **Visits** refers to the number of visits to the MUS website (and hence, Display Visits indicates that the source are Display adverts—video ads, banners, etc.). Unless otherwise indicated, when Traffic and Display Visits are mentioned in this document, they refer to the MUS values, not to all the car manufacturer's models.

Since all these data from the car manufacturer (or its media agency) are confidential, these time series have been transformed as follows: First, a matrix is built for each dynamic variable,

in which the row indicates the week of the pre-test period, and the column indicates the GMA. Then, a new matrix is built by dividing each cell of the former one by the total sum of its cells—i.e., the total amount of that variable throughout the whole pre-test period and for all the GMAs in the final sample—, and multiplying each by 100. This way we obtain a matrix in which the sum of all rows of a column is the percentage of that GMA on the total, and the sum of all columns of a row is the percentage of that week on the total. Thus, we preserve the information about how these variables evolve in each city.

Table 1. Example of variable transformation: first and last 6 rows and columns of the Display Visits matrix.

	A Coruña	Albacete	Algeciras	Alicante	Avila	Barcelona	Teruel	Tortosa	Valencia	Valladolid	Vitoria	Zaragoza	
w1	0.008	0.004	0.002	0.010	0.002	0.071	0.001	0.000	0.030	0.006	0.004	0.013	**0.437**
w2	0.001	0.000	0.000	0.001	0.000	0.009	0.000	0.000	0.003	0.001	0.000	0.001	**0.043**
w3	0.001	0.000	0.000	0.001	0.000	0.007	0.000	0.000	0.001	0.000	0.000	0.001	**0.029**
w4	0.004	0.002	0.001	0.006	0.002	0.041	0.001	0.000	0.015	0.005	0.001	0.006	**0.265**
w5	0.013	0.004	0.002	0.018	0.002	0.108	0.002	0.000	0.047	0.011	0.005	0.021	**0.664**
w6	0.031	0.011	0.006	0.039	0.004	0.235	0.002	0.001	0.114	0.025	0.013	0.050	**1.512**
w104	0.003	0.001	0.001	0.005	0.000	0.021	0.001	0.000	0.012	0.002	0.002	0.006	**0.159**
w105	0.003	0.001	0.000	0.003	0.000	0.018	0.000	0.000	0.009	0.002	0.001	0.003	**0.118**
w106	0.001	0.001	0.000	0.002	0.000	0.014	0.000	0.000	0.006	0.002	0.001	0.004	**0.096**
w107	0.014	0.009	0.005	0.023	0.003	0.262	0.002	0.001	0.083	0.016	0.012	0.033	**1.145**
w108	0.037	0.018	0.008	0.037	0.007	0.523	0.004	0.002	0.168	0.038	0.025	0.077	**2.464**
w109	0.025	0.013	0.005	0.027	0.005	0.327	0.004	0.001	0.114	0.021	0.018	0.053	**1.550**
	1.798	**0.783**	**0.365**	**2.079**	**0.297**	**17.910**	**0.194**	**0.065**	**7.055**	**1.674**	**0.832**	**3.020**	**100.000**

2.3.2. Static variables

- From the National Statistics Institute of Spain (INE):
  - Population of the 8,118 towns of Spain, as at January 1, 2013.
- From the Economic Yearbook of Spain (2013), of the financial institution La Caixa (which contains a set of statistical data and economic indicators of the 3,245 towns with more than 1,000 inhabitants existing in Spain as at January 1, 2012, whose population accounts for 96.8% of the entire population):
  - Extension (in square kilometers)
  - Unemployment rate or unemployment-to-population ratio
  - Land lines
  - Percent change in number of land lines since 2007
  - Broadband lines
  - Cars
  - Percent change in number of cars since 2007
  - Economic indicators
    - Market share
    - Economy activity index
    - Industrial index
    - Industrial activity index

- Wholesale trade index
- Wholesale trade activity index
- Retail trade index
- Retail trade activity index
- Trade index
- Tourist index

In order to analyze a combination of static and dynamic variables, single values (per GMA) of the latter ones are required. Both the average value during the last 104 weeks of the pre-test period and the slope of its regression line—to have an estimation of its trend—are used in that case. When analyzing both static and dynamic variables, all of them (or at least the latter ones, whose values are confidential) are standardized in the usual way—subtracting the mean from each value and dividing by the sample standard deviation, hence obtaining new variables with a mean of zero and a standard deviation of one.

Other static variables related to the automotive market in Spain—such as total number or car registrations—, but per Autonomous Community rather than per city, were also used just to analyze some interesting results, like the performance of the car manufacturer as compared to its competitors. Those results are briefly mentioned in the Annex of this document, Section A2.

2.4. Units of analysis

As previously mentioned, our main subject or unit of analysis is the city, and more specifically, the GMA, which is the city where it is possible for Google to launch a geo-targeted advertising campaign in YouTube.

Based on this, we only need data from cities which are GMAs. By the time this project was finished, there were 443 GMAs, and all of them are included in the cities for which data of the static variables, as well as of the variables measured by the car manufacturer's media agency, exist. Some variables associated to each GMA are its geographical coordinates, province, INE code, etc.

The secondary units of analysis are the car manufacturer's dealers and their POSs. Not every dealer—but just 178 of them—collect data. And, as those data are segmented by dealer, but not by POS, only the dealers with POSs in a single city must be considered, since otherwise there is no way to know how many sales, for instance, were closed in each POS. Some variables associated to each POS are its trade name—i.e., the dealer or company which owns that POS—, address, zip code, geographical coordinates, province, etc.

3. Project phases

In this chapter all the work corresponding to the first two stages of the whole experiment (see Chapter 2, Subsections 2.2.1 and 2.2.2) is described.

3.1. Data cleansing and Sample selection

The original sample, after collecting all the information from the different sources, consisted of more than 32 million data spread across about 250 files. It included not every dealer of the car manufacturer under study, but only the 178 of them which collect weekly data[4].

It was necessary to gather all that information in just a few tables, linking the interrelated variables for each unit of analysis—be it a GMA or a dealer—and discarding the data non-useful for this project. That included, for example, data from cities that are not GMAs—in the case of static variables and variables measured by the car manufacturer's media agency—as only this type of cities must be included in the final sample.

Besides, some other data had to be excluded for the final sample to be valid for our study:

- Data from dealers for which there is no traffic data for at least the last 104 of the 109-week pre-test period[5]—this way, dealers that ceased operating before the beginning of the pre-test period, or started operating over 5 weeks after the end of it, are discarded.
- Data from dealers for which there is not enough significant traffic data[6]—hence, dealers not considered to be successful enough are discarded.

This reduced the number of possible dealers to be analyzed from 178 to 119 (with 163 POSs in 140 cities, scattered throughout the country[7]).

4 According to the car manufacturer, these 178 dealers represent "the majority of them". The total number of dealers, their location, and their traffic data were unknown.

5 Though the pre-test period lasts 109 weeks, it was assumed that it is enough to have data from last 104 of them (about 2 years). Dealers which started operating recently may distort the results, and of course it makes no sense to include dealers which are no longer operating.

6 It was assumed that a dealer for which traffic was zero—i.e., nobody interested in the MUS visited a dealer's POS in a whole week—in more than 26 of the last 104 weeks of the pre-test period—i.e., 25%—did not have enough significant data.

7 With the exception of the Canary Islands—none of the 178 dealers that collect data are located there—and some provinces.

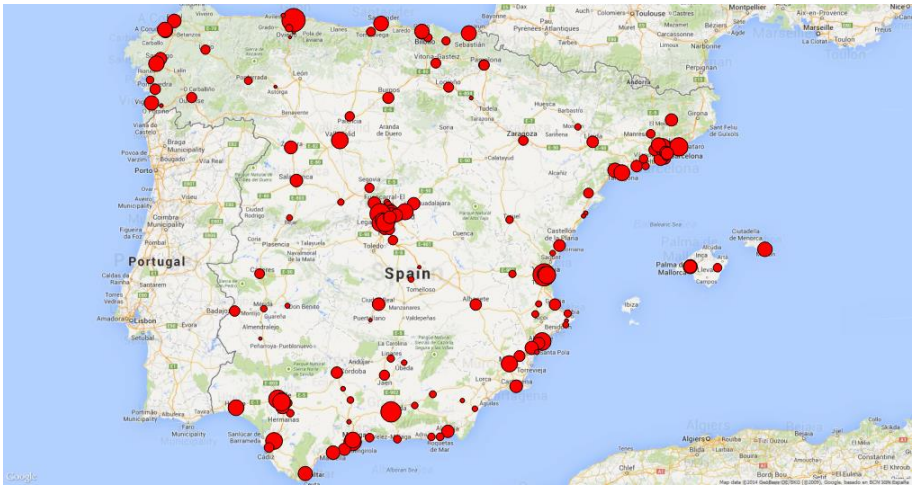


Figure 1. Geographical location of the POSs of the 119 dealers after the first filtering (size is proportional to average weekly traffic, weighted by population when a dealer has POSs in different cities).

The following step involved discarding the dealers:

1. with POSs in more than one city, and
2. whose POSs are not located in a GMA.

Thus obtaining the final sample, i.e., the GMAs that satisfy all the requested conditions (in summary—there are POSs of the car manufacturer located in them, and there are known and relevant data for at least the total of all those POSs[8]), and hence shall be included either in the Test group or the Control group.

This final sample contains 69 dealers, with 76 POSs in **57 GMAs**, with the traffic in them representing 56% of the original sample[9]. These 57 GMAs (in alphabetical order) are: A Coruña, Albacete, Algeciras, Alicante, Avila, Barcelona, Bejar, Burgos, Caceres, Camas, Cartagena, Cordoba, Coslada, Eibar, El Puerto de Santa Maria, Fuenlabrada, Gandia, Getafe, Gijon, Granada, Guadalajara, Huelva, Jerez de la Frontera, Leganes, Leioa, Lleida, Lugo, Madrid, Mahon, Malaga, Manacor, Marbella, Mataro, Mollet del Valles, Murcia, Orihuela, Ourense, Palencia, Palma de Mallorca, Pamplona, Parla, Pontevedra, Requena, Reus, San Sebastian, Sant Cugat del Valles, Santander, Santiago de Compostela, Segovia, Seville, Tarragona, Teruel, Tortosa, Valencia, Valladolid, Vitoria, and Zaragoza.

As seen in the figure below, this final sample includes cities of many different sizes and regions. Almost 32% of the Spanish population live in these 57 cities.

8 An additional condition is that there are (valid) data for any other variable apart from traffic. Unfortunately, that was not whose the case of the city of Zamora—that otherwise could have been included as the 58th GMA in the final sample—, display visits data were not reliable.

9 Bear in mind that not every dealer collects data, so the actual traffic percentage of the dealers included in the final sample may be lower.

Figure 2. Geographical location of the 57 GMAs of the final sample (size is proportional to average weekly traffic).

Figure 3. Screenshot of the final sample database (confidential values are blurred).

3.2. Exploratory data analysis

3.2.1. Descriptive statistics

Now we will focus only in the variables per GMA mentioned in the related Section. As it was mentioned there, when dynamic variables are to be analyzed together with the static ones, only their mean and the slope of their regression lines will be considered. Besides, those variables are standardized to keep their confidentiality.

Table 2. Descriptive statistics of the static variables in the 57 GMAs.

	Mean	Standard Error	Median	Standard Deviation	Kurtosis	Skewness	Range	Minimum	Maximum	Sum	Confidence Level(95.0%)
Population	259,883.772	62,509.072	133,545.000	471,933.143	28.492	4.966	3,192,967.000	14,280.000	3,207,247.000	14,813,375.000	125,220.718
Extension (km2)	271.754	48.679	135.000	367.521	5.013	2.263	1,741.000	9.000	1,750.000	15,490.000	97.516
Unemployment rate	126,136.754	33,026.505	63,846.000	249,344.648	30.276	5.185	1,695,317.000	7,570.000	1,702,887.000	7,189,795.000	66,160.040
Market share	13.279	1.169	12.800	8.822	0.998	0.119	49.600	-12.000	37.600	756.900	2.341
Land lines	70,543.754	17,545.893	35,663.000	132,468.584	29.430	5.080	899,051.000	4,290.000	903,341.000	4,020,994.000	35,148.646
Change in #land lines since 2007 (%)	117,920.316	28,078.443	57,256.000	211,987.592	32.508	5.275	1,485,017.000	7,025.000	1,492,042.000	6,721,458.000	56,247.879
Broadband lines	5.168	0.630	5.200	4.755	1.070	-0.199	27.900	-10.000	17.900	294.600	1.262
Cars	14.560	0.446	14.200	3.367	0.708	0.646	17.000	7.200	24.200	829.900	0.893
Change in # cars since 2007 (%)	553.789	133.962	301.000	1,011.390	28.714	4.998	6,845.000	33.000	6,878.000	31,566.000	268.358
Industrial activity index	2,948.070	724.897	1,598.000	5,472.855	28.363	5.030	36,714.000	126.000	36,840.000	168,040.000	1,452.144
Wholesale trade activity index	1,005.614	272.766	462.000	2,059.340	24.057	4.725	12,896.000	50.000	12,946.000	57,320.000	546.416
Retail trade activity index	4,503.860	979.281	2,612.000	7,393.413	21.093	4.335	45,941.000	326.000	46,267.000	256,720.000	1,961.736
Industrial index	478.035	99.021	226.000	747.595	20.878	4.130	4,785.000	9.000	4,794.000	27,248.000	198.364
Economy activity index	736.649	176.605	406.000	1,333.336	21.076	4.353	8,215.000	28.000	8,243.000	41,989.000	353.782
Wholesale trade index	690.649	181.491	316.000	1,370.225	21.895	4.486	8,384.000	29.000	8,413.000	39,367.000	363.570
Retail trade index	772.070	173.664	455.000	1,311.134	20.067	4.201	8,085.000	27.000	8,112.000	44,008.000	347.891
Tourist index	669.193	214.071	132.000	1,616.202	19.972	4.327	9,399.000	1.000	9,400.000	38,144.000	428.836
Trade index	720.947	213.714	311.000	1,613.506	27.248	5.027	10,471.000	17.000	10,488.000	41,094.000	428.121

Table 3. Descriptive statistics of the (standardized) dynamic variables in the 57 GMAs.

	Standard Error	Median	Kurtosis	Skewness	Range	Minimum	Maximum	Confidence Level(95.0%)
MUS-Visits-All	0.132	-0.070	0.217	0.441	4.527	-1.851	2.676	0.265
Slope MUS-Visits-All	0.132	0.124	50.448	-6.872	8.404	-7.204	1.200	0.265
MUS-Visits-AFF	0.132	-0.245	0.536	1.063	3.917	-0.907	3.010	0.265
Slope MUS-VISITS-AFF	0.132	0.229	43.168	-6.356	7.217	-6.929	0.287	0.265
MUS-VISITS-DIS	0.132	0.030	0.282	0.425	4.711	-1.847	2.863	0.265
Slope MUS-VISITS-DIS	0.132	-0.182	37.948	5.796	7.718	-0.995	6.723	0.265
MUS-VISITS-DRT	0.132	-0.218	9.087	2.567	5.889	-1.252	4.637	0.265
Slope MUS-VISITS-DRT	0.132	0.144	56.733	-7.524	7.650	-7.409	0.241	0.265
MUS-VISITS-EML	0.132	-0.080	0.561	0.752	4.260	-1.678	2.582	0.265
Slope MUS-VISITS-EML	0.132	0.215	26.606	-4.783	7.139	-6.145	0.994	0.265
MUS-VISITS-REF	0.132	-0.164	-0.778	0.289	3.895	-1.818	2.077	0.265
Slope MUS-VISITS-REF	0.132	-0.194	41.953	6.146	7.671	-0.777	6.895	0.265
MUS-VISITS-SEA	0.132	-0.099	-0.535	0.243	4.113	-1.882	2.231	0.265
Slope MUS-VISITS-SEA	0.132	0.224	19.767	-3.672	8.099	-5.695	2.405	0.265
MUS-VISITS-SEO	0.132	-0.190	-0.285	0.349	4.182	-1.818	2.364	0.265
Slope MUS-VISITS-SEO	0.132	0.237	34.608	-5.397	8.022	-6.586	1.436	0.265
MUS-VISITS-SOC	0.132	-0.047	0.098	0.680	4.208	-1.624	2.584	0.265
Slope MUS-VISITS-SOC	0.132	-0.258	41.757	6.141	7.287	-0.399	6.888	0.265
MUS-SEO	0.132	-0.117	0.198	0.521	4.735	-1.761	2.974	0.265
Slope MUS-SEO	0.132	0.240	39.202	-5.980	7.106	-6.771	0.336	0.265
MUS-SEM	0.132	-0.248	-0.041	0.513	4.493	-1.789	2.704	0.265
Slope MUS-SEM	0.132	0.251	33.545	-5.496	7.297	-6.497	0.800	0.265
Manufacturer-VISITS-ALL	0.132	-0.041	0.845	0.580	4.745	-1.844	2.900	0.265
Slope Manufacturer-VISITS-ALL	0.132	0.086	51.126	-6.913	8.686	-7.226	1.460	0.265
Manufacturer-VISITS-AFF	0.132	-0.194	-0.298	0.379	4.169	3.918	2.440	0.265
Slope Manufacturer-VISITS-AFF	0.132	0.236	33.785	-5.583	6.838	-6.486	0.352	0.265
Manufacturer-VISITS-DIS	0.132	0.053	0.459	0.477	4.710	-1.810	2.899	0.265
Slope Manufacturer-VISITS-DIS	0.132	-0.277	28.415	5.045	6.631	-0.392	6.239	0.265
Manufacturer-VISITS-DRT	0.132	-0.239	6.328	2.134	5.632	-1.286	4.346	0.265
Slope Manufacturer-VISITS-DRT	0.132	0.138	56.895	-7.540	7.632	-7.414	0.218	0.265
Manufacturer-VISITS-EML	0.132	-0.103	33.700	5.190	7.572	-1.017	6.555	0.265
Slope Manufacturer-VISITS-EML	0.132	0.063	14.662	1.322	8.449	-3.323	5.126	0.265
Manufacturer-VISITS-REF	0.132	-0.112	-0.378	0.252	4.338	-1.835	2.504	0.265
Slope Manufacturer-VISITS-REF	0.132	-0.316	19.240	4.162	6.234	-0.782	5.453	0.265
Manufacturer-VISITS-SEA	0.132	-0.028	-0.519	0.215	4.127	-1.869	2.258	0.265
Slope Manufacturer-VISITS-SEA	0.132	0.230	30.923	-5.045	7.930	-6.406	1.524	0.265
Manufacturer-VISITS-SEO	0.132	-0.063	0.414	0.519	4.703	-1.805	2.898	0.265
Slope Manufacturer-VISITS-SEO	0.132	0.157	33.303	-5.043	8.329	-6.519	1.811	0.265
Manufacturer-VISITS-SOC	0.132	-0.215	0.186	0.839	4.269	-1.682	2.587	0.265
Slope Manufacturer-VISITS-SOC	0.132	-0.234	43.604	6.331	7.363	-0.406	6.957	0.265
MUS-TRAFFIC	0.132	-0.216	17.857	3.882	6.355	-0.827	5.528	0.265
Slope MUS-TRAFFIC	0.132	-0.240	7.194	2.332	5.568	-1.209	4.358	0.265
MUS-REGISTRATIONS	0.132	-0.229	16.701	3.894	5.688	-0.801	4.887	0.265
Slope MUS-REGISTRATIONS	0.132	-0.278	7.603	2.503	5.438	-1.252	4.186	0.265
MUS-TESTS	0.132	-0.261	10.819	2.937	5.684	-0.919	4.766	0.265
Slope MUS-TESTS	0.132	-0.233	2.117	1.328	5.226	-2.217	3.009	0.265
MUS-ORDERS	0.132	-0.250	15.848	3.784	5.610	-0.804	4.806	0.265
Slope MUS-ORDERS	0.132	-0.247	6.463	2.328	5.396	-1.348	4.049	0.265
MUS-CANCELLED ORDERS	0.132	-0.292	16.853	3.587	6.285	-0.741	5.543	0.265
Slope MUS-CANCELLED ORDERS	0.132	-0.259	7.203	2.221	6.027	-1.555	4.472	0.265
Manufacturer-TRAFFIC	0.132	-0.261	16.776	3.750	6.268	-0.839	5.428	0.265
Slope Manufacturer-TRAFFIC	0.132	-0.211	8.294	2.673	5.224	-1.124	4.100	0.265
Manufacturer-REGISTRATIONS	0.132	-0.288	11.732	3.174	5.650	-0.913	4.737	0.265
Slope Manufacturer-REGISTRATIONS	0.132	-0.229	7.990	2.571	5.672	-1.408	4.265	0.265
Manufacturer-TESTS	0.132	-0.283	10.835	2.760	6.059	-1.026	5.033	0.265
Slope Manufacturer-TESTS	0.132	-0.307	6.328	2.283	5.546	-1.452	4.094	0.265
Manufacturer-ORDERS	0.132	-0.266	10.878	3.058	5.589	-0.918	4.672	0.265
Slope Manufacturer-ORDERS	0.132	-0.216	7.388	2.462	5.532	-1.451	4.081	0.265
Manufacturer-CANCELLED ORDERS	0.132	-0.236	27.252	4.620	6.935	-0.697	6.238	0.265
Slope Manufacturer-CANCELLED ORDERS	0.132	-0.191	6.230	2.403	4.778	-1.114	3.663	0.265

3.2.2. Fitting distributions statistics

Normality tests demonstrate that the distribution of the most important variables (the dynamic ones, which are the ones related to the car manufacturer) is not normal in any case[10]. For example, these are the p-values when the Shapiro-Wilk tests are performed on the two most important variables (as mentioned in Chapter 2, Subsection 2.3.1)—Traffic and Display Visits:

```
shapiro.test(MUS$Traffic)$p.value
## [1] 1.375e-08
shapiro.test(MUS$Disp_Visits)$p.value
## [1] 9.849e-15
```

As the p-value is so close to zero in both cases, we must reject the null-hypothesis (the sample is normally distributed).

Other distributions were also tested (using the *fitdistr* function included in *MASS* package), with no results. So no usual distribution seems to fit the final sample.

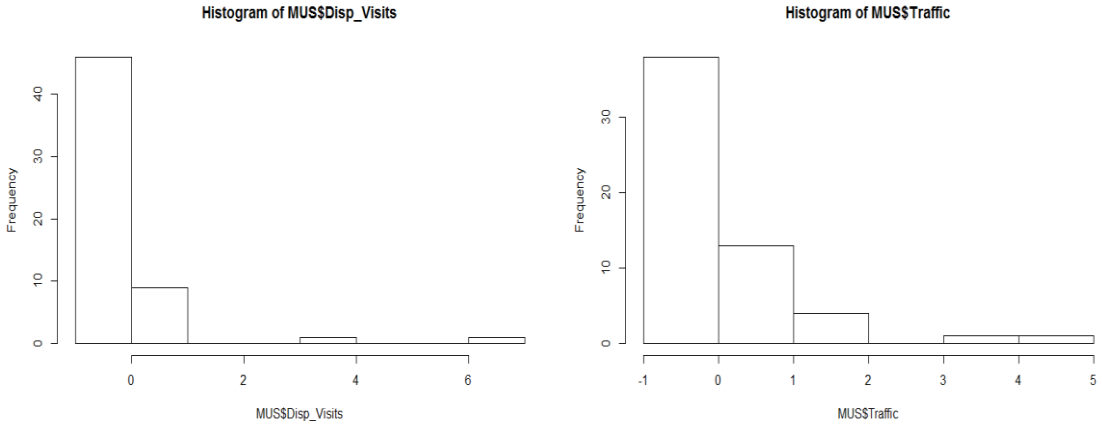


Figure 4. Histograms of standardized Traffic and Display Visits biennial averages per GMA.

This is probably due to the presence of outliers, especially Madrid and Barcelona[11], as seen in the figure below.

10 Apart from those tests, some results shown in Table 2 and Table 3 already indicate the non-normality—the median is different to the mean, and the skewness and kurtosis are not even close to zero.

11 It could be argued that 57 observations are not enough to fit a distribution to them. But the final sample contains the 8 most populated cities of Spain. So it is unlikely that including other cities in the final sample would change the histogram significantly.

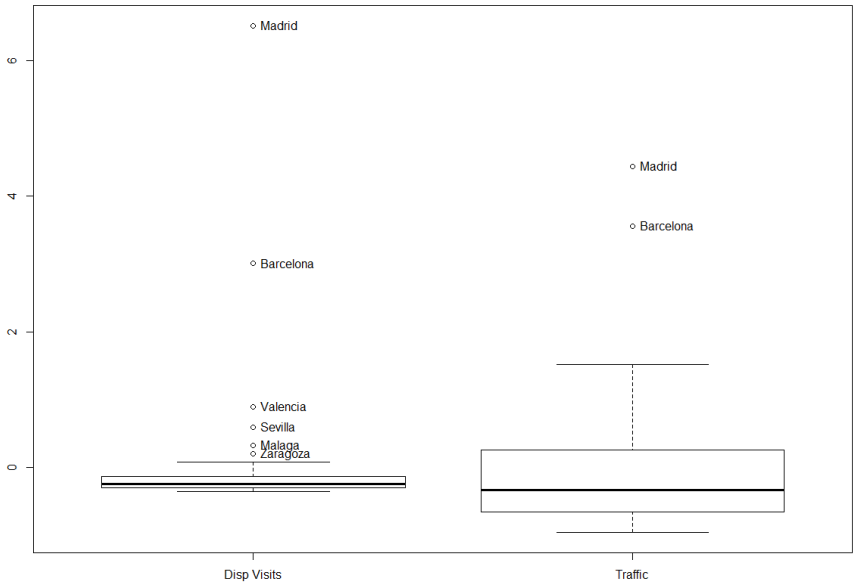
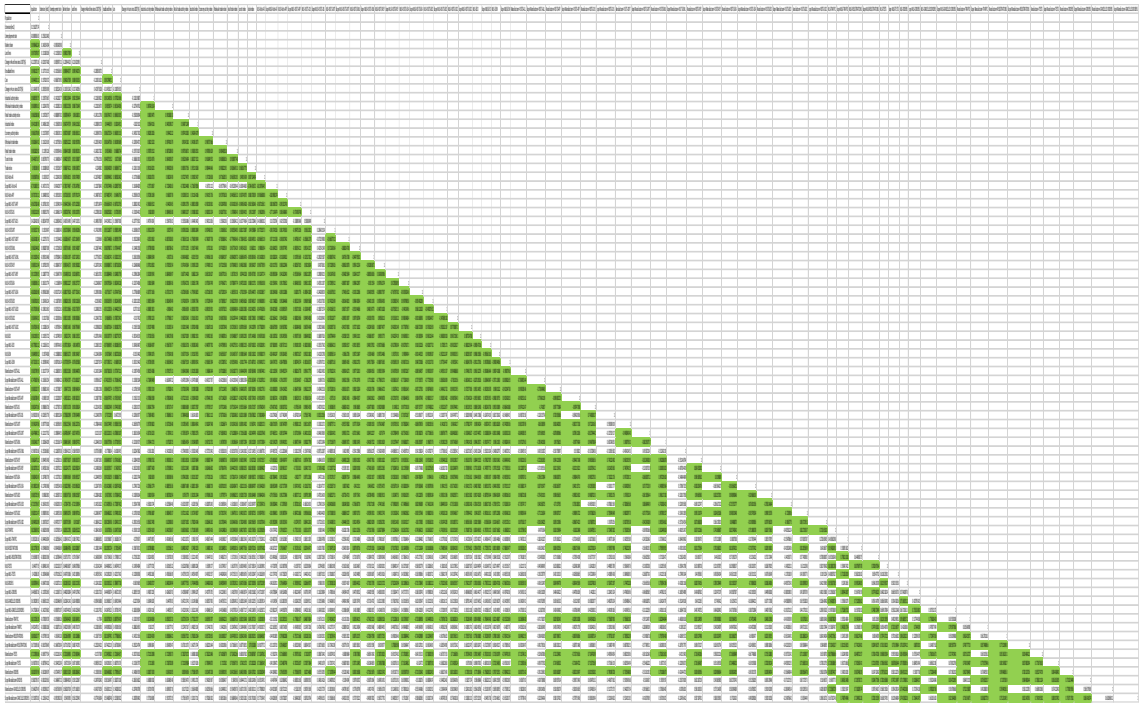


Figure 5. Boxplots of standardized Traffic and Display Visits biennial averages per GMA.

3.2.3. Correlations

As expected, the correlation between each pair of variables is quite high in most cases, due to the proportionality to population[12].

Table 4. Correlation matrix of the variables (green background cells indicate a correlation greater than or equal to 0.8).



12 For example, the correlation between Display Visits and population is 0.985 (i.e., both city variables are almost proportional to each other), and the correlation between Traffic and population is 0.826.

Table 5. First rows and columns of the correlation matrix.

	Population	Extension (km2)	Unemployment rate	Market share	Land lines	Change in #land lines since 2007 (%)	Broadband lines	Cars	Change in # cars since 2007 (%)	MUS-Visits-All	Slope MUS-Visits-All	MUS-Visits-AFF	Slope MUS-VISITS-AFF	MUS-VISITS-DIS	Slope MUS-VISITS-DIS
Population	1.000														
Extension (km2)	0.152	1.000													
Unemployment rate	-0.090	0.254	1.000												
Market share	1.000	0.146	-0.090	1.000											
Land lines	0.997	0.131	-0.112	0.998	1.000										
Change in #land lines since 2007 (%)	-0.223	-0.022	0.086	-0.221	-0.214	1.000									
Broadband lines	0.998	0.138	-0.102	0.999	1.000	-0.209	1.000								
Cars	0.994	0.157	-0.069	0.994	0.990	-0.210	0.992	1.000							
Change in # cars since 2007 (%)	-0.314	0.294	0.193	-0.319	-0.317	0.418	-0.315	-0.311	1.000						
MUS-Visits-All	0.986	0.103	-0.121	0.988	0.992	-0.217	0.989	0.981	-0.317	1.000					
Slope MUS-Visits-All	-0.791	-0.146	0.041	-0.789	-0.793	0.121	-0.790	-0.830	0.161	-0.820	1.000				
MUS-Visits-AFF	0.970	0.105	-0.140	0.972	0.980	-0.195	0.975	0.965	-0.296	0.994	-0.830	1.000			
Slope MUS-VISITS-AFF	-0.963	-0.108	0.134	-0.964	-0.971	0.188	-0.967	-0.967	0.281	-0.987	0.882	-0.994	1.000		
MUS-VISITS-DIS	0.985	0.098	-0.124	0.987	0.991	-0.228	0.988	0.973	-0.331	0.996	-0.771	0.986	-0.971	1.000	
Slope MUS-VISITS-DIS	0.428	-0.085	-0.217	0.437	0.447	-0.150	0.442	0.360	-0.238	0.437	0.134	0.425	-0.339	0.501	1.000

The correlation between Traffic and Display Visits is 0.835.

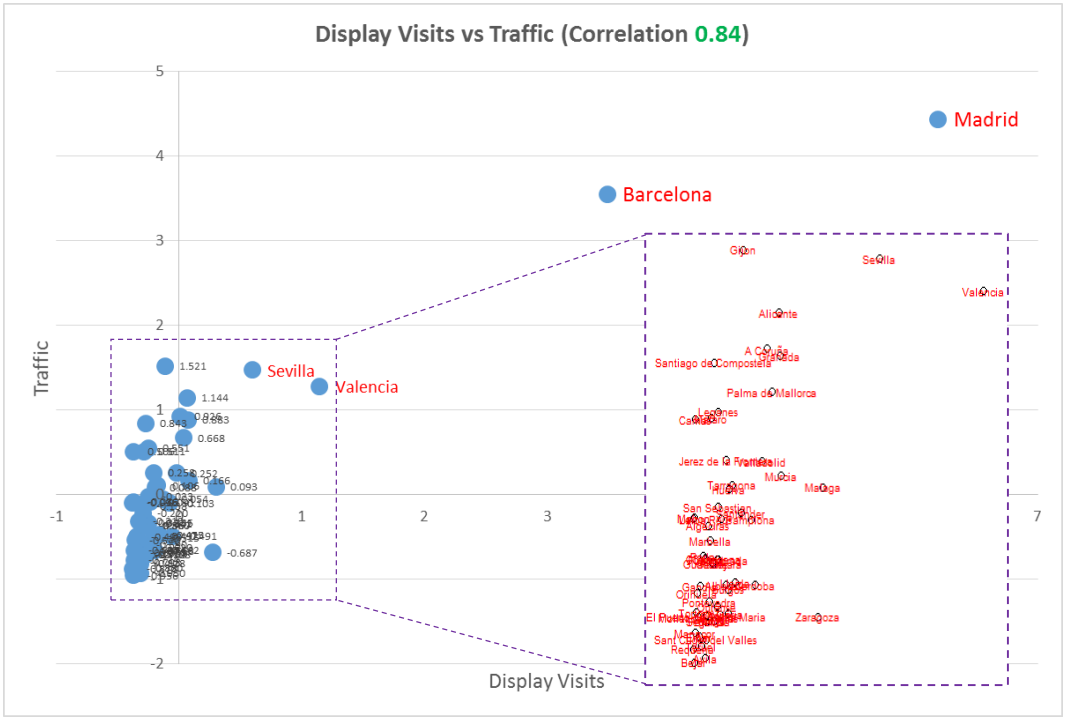


Figure 6. Scatterplot of Visits vs Traffic.

As expected, the more people from a city visit the MUS website, the more they visit any POS in that city. And the more people live in a city, the more they visit the website or a POS. But it should be noticed (the scatterplot above hints to it) that this correlation is reduced to only 0.595 if Madrid and Barcelona—the main outliers—are extracted from the sample.

3.2.4. Per capita values

In order to avoid the effect of population, per capita values were calculated. The intention was to transform the outliers into more typical data, so it would be easier to match cities, from relationships like "On average, about X out of 100 people in both cities A and B visited the MUS website per week".

But the population effect did not disappear completely (Madrid and Barcelona were still

outliers, and some new ones, like Camas, Mahón or Leioa, appeared), and now the correlation was low and negative, which does not seem to make much sense.

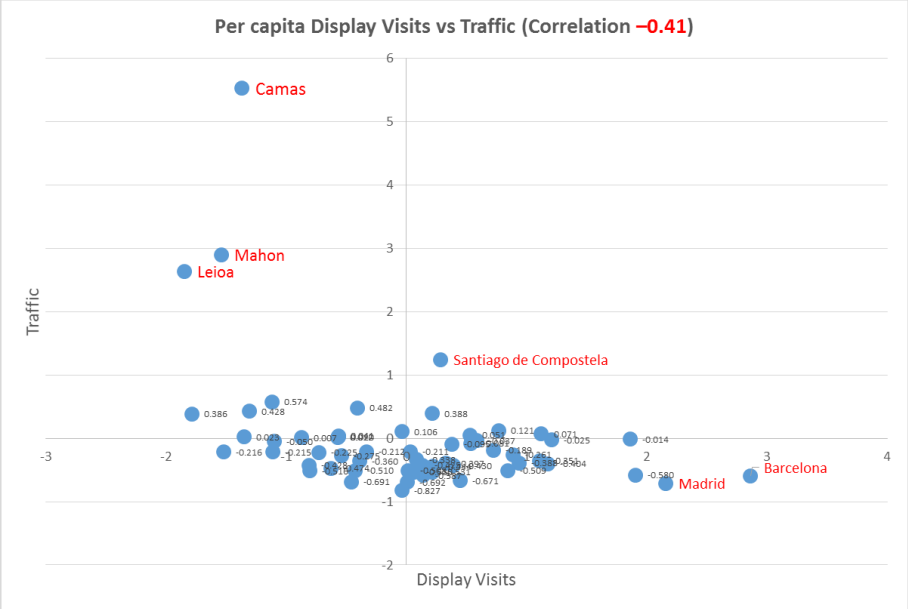


Figure 7. Scatterplot of per capita Display Visits vs Traffic.

Explaining this situation would certainly require to include many other variables[13]. The fact is that per capita values should not be used.

A possible solution could be to divide the Display Visits not by population but by a better proxy for Internet penetration. Unfortunately, there are no data (at least on a city basis) about this, and the only information available—number of land lines or broadband lines—is not updated enough, and would not consider the growing amount of people that currently use their mobiles and tablets to access the Internet.

3.2.5. Time series

Graphs of the response variable—Traffic—and the most important explanatory variable—Display Visits—, aggregated and per GMA, as well as their decompositions, can be found in the following pages. As mentioned, their values have been transformed or scaled—so the area under each total variable curve is equal to 100—in a way that preserves the aspect of each curve.

13 For example, the reason why per capita Visits are so high in Madrid and Barcelona could be that the Internet penetration is also higher in these cities. And maybe per capita Traffic is so high in Camas, Mahón or Leioa due to a lack of nearby alternatives.

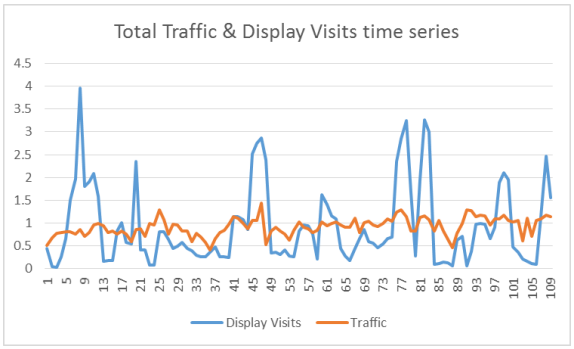


Figure 8. Graph of total Traffic and Display Visits time series.

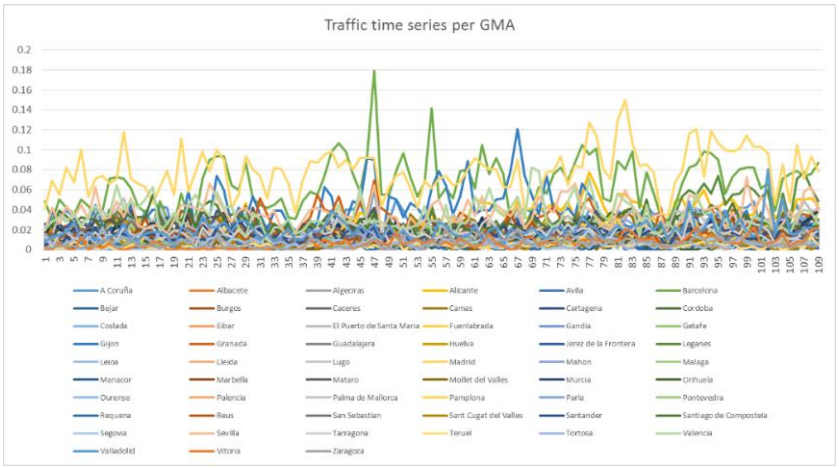


Figure 9. Graph of Traffic time series per GMA.

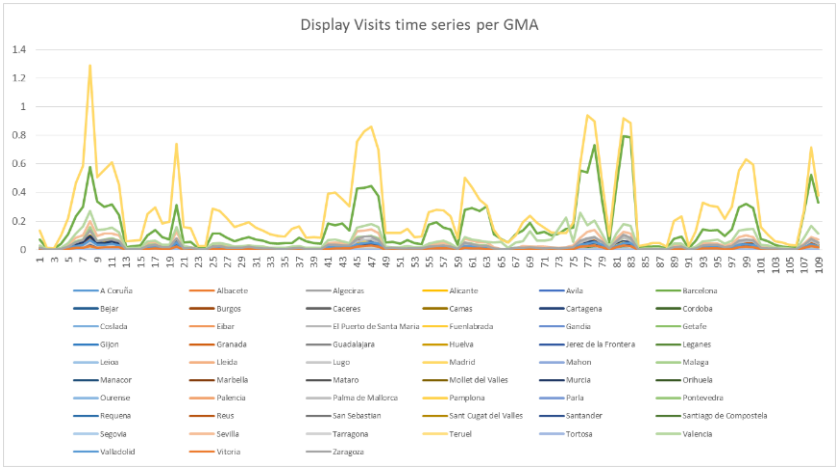


Figure 10. Graph of Display Visits time series per GMA.

As seen in Figure 10, the Display Visits time series in each GMA are quite similar, as those visits have been mainly the effect of specific advertising campaigns run at different moments. The same is not true for the Traffic time series (see Figure 9). And the goal of this project (as it is explained in more detail later) is to split the sample in 2 groups, so that the aggregate Traffic time series of a group is as similar to the other's as possible. How to achieve this is the core of this project.

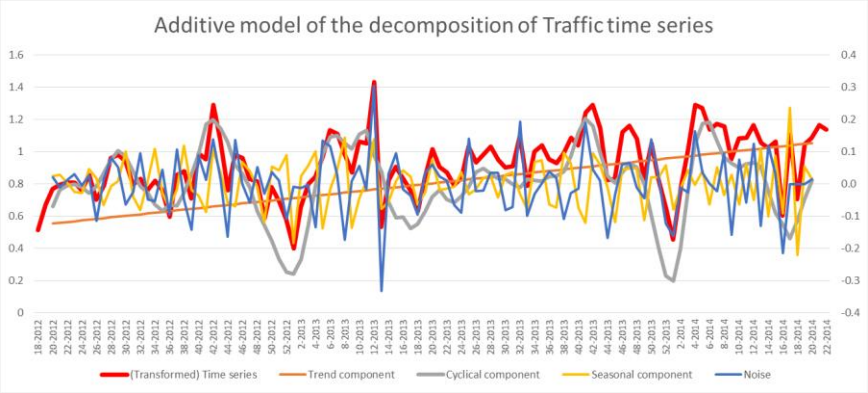


Figure 11. Decomposition of the Traffic time series.

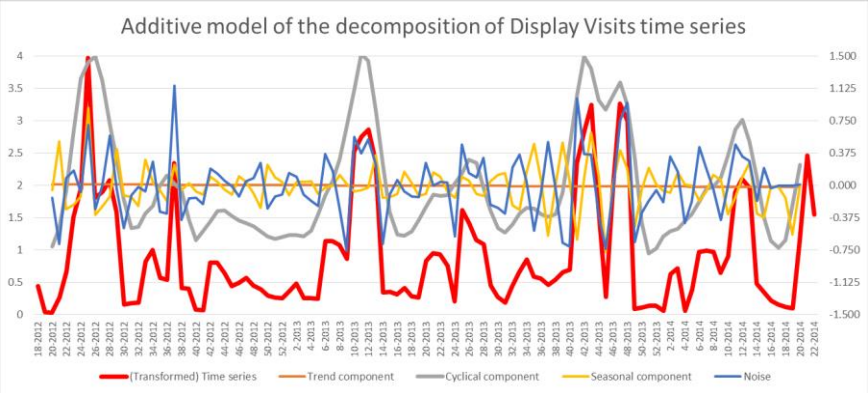


Figure 12. Decomposition of the Display Visits time series.

The following results can be extracted from the two figures above:

- Traffic:
  - Positive trend.
  - Strong cyclical component. There is a significant growth of Traffic during the first 2 or 3 months of the year, then it becomes more or less stable until summer, when it increases, and finally it decreases from the end of it until the end of the year.
- Display Visits:
  - Negative trend, close to zero.
  - Strong cyclical component. In this case there are several peaks, and Display Visits are quite low during the rest of the time. But those peaks—highly dependent on the MUS marketing actions—do not always occur in the same weeks of the year (that may depend on when the car manufacturer launches new versions of the MUS, its plans for other models, and many other factors).

Both time series have a small seasonal component (comparable to the noise). A possible reason may be that the time units are weeks instead of months (an increase at the end of each month would be expected).

3.3. First approach – Cluster analysis

The first approach was based on the idea that, based on their characteristics, it may be possible to allocate cities to certain categories. Hence, it involved detecting clusters within the sample, and for each one, randomly assigning half of its elements to the Test group, and the other half to the Control group. Thus, both groups would contain the same number of elements of each cluster or category.

Ideally, the clusters should be very well-defined, i.e., very dense (small within-cluster variations) and far apart from each other (large within-cluster variations). Unfortunately, this is not the case. Another desirable attribute is that clusters contain at least two elements, and preferably an even number of them (otherwise there will be one element that cannot be assigned to any group).

The idea was to use as many available variables as possible, in order to better categorize each city, so cities within the same cluster would be similar in many aspects. As single values per GMA are required, only the (standardized values) of the mean during the last 104 weeks of the pre-test period and the slope of its regression line are used in the case of dynamic variables.

3.3.1. Using Traffic and Display Visits

The first analyses only consider Traffic and Display Visits of the MUS. Just examining Figure 6 (in Subsection 3.2.4), at first glance it seems that there are 3 or 4 well-defined clusters:

- The 2 outliers (Madrid and Barcelona, the most populated cities of Spain) form their own clusters, or a single one (but with a quite high within-cluster variation).
- Most of the other elements, which are small to medium-size cities, could form a third one.
- The remaining elements, all of them in the top 20 of Spanish most populated cities (Valencia, Seville, Zaragoza, Malaga, Gijon, Alicante…) would require a deeper—and statistically valid—analysis.

The R script developed (Clusters_Traffic_and_Visits.R) can be found in the Annex of this document, Subsection A1.1. In all cases the optimal number of clusters between 1 and 28 (half of the sample size, since we would like to find clusters with at least 2 elements) was searched. As this is just a first approach, only some of the main results of this clustering are mentioned below.

First, **hierarchical clusters** were tested. Based on the silhouette of them, 3 is the optimal number of clusters, regardless of the method used—Madrid and Barcelona form their own

clusters[14], and the remaining GMAs the third one.

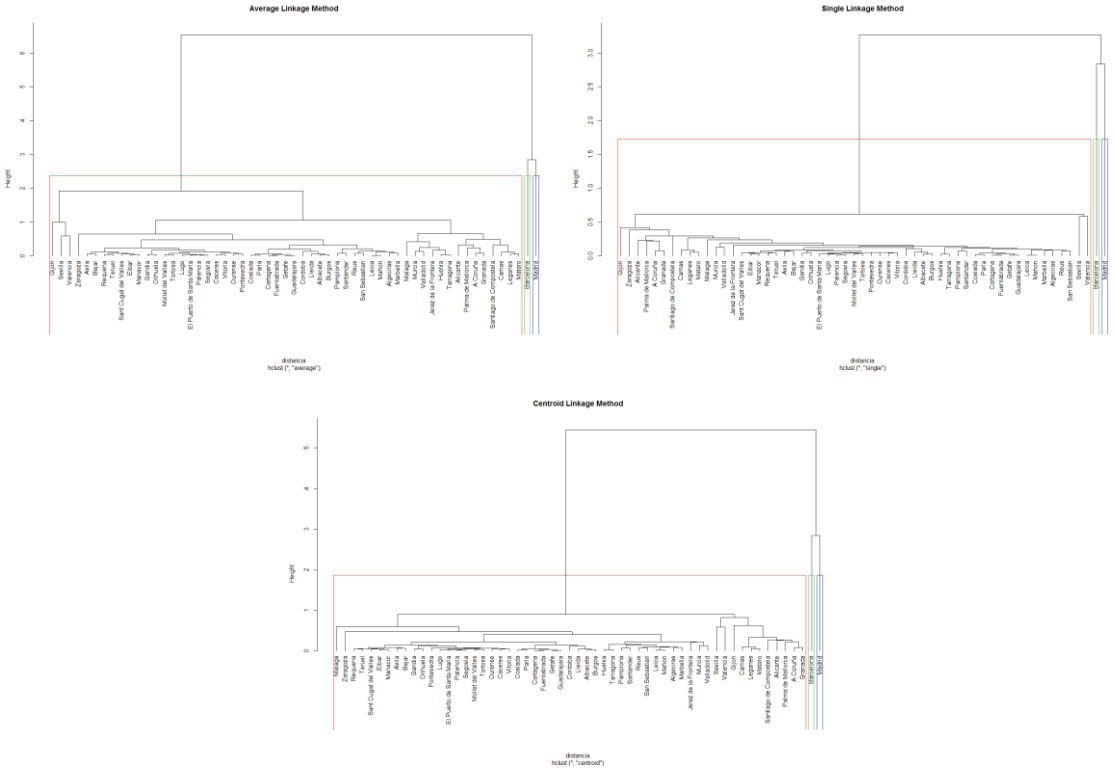


Figure 13. Hierarchical clustering dendograms using different methods.

K-means clustering was also tested. In this case the optimal number of clusters[15] was chosen based on the Calinski-Harabasz index, to have a small total within-cluster variation and a large between-cluster variation.

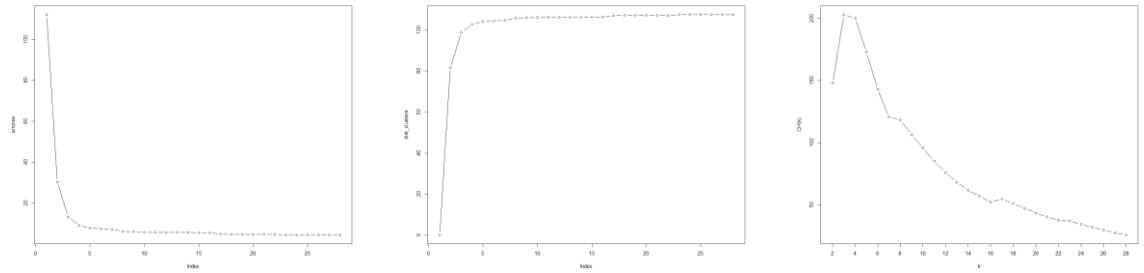


Figure 14. Within- and between-cluster variations, and Calinski-Harabasz indices for k-means clustering (k=1…28).

Again, the optimal numbers of clusters is 3, but now Madrid and Barcelona form a single

[14] The best situation would be that they are part of a cluster with more elements, in order to assign at least one of these cities to the Test group. Since they are the most populated cities of Spain, the influence of the stimulus—the YouTube campaign—would be more evident in them than in any other city.

[15] Of course the results depend on the initial starting point values, so when assessing different values of *k*, the same seed needs to be used. Different seeds were used for *k=3*, and the clusters obtained were always the same.

cluster, some of the quite large cities that we mention at the beginning of this Subsection form another one, and the remaining 40 cities form the third cluster.

```
## [[1]]
## Barcelona    Madrid
##         1         1
##
## [[2]]
##                 Albacete                Algeciras                    Avila
##                        2                        2                        2
##                    Bejar                   Burgos                  Caceres
##                        2                        2                        2
##                Cartagena                  Cordoba                  Coslada
##                        2                        2                        2
##                    Eibar El Puerto de Santa Maria              Fuenlabrada
##                        2                        2                        2
##                  Gandia                   Getafe               Guadalajara
##                        2                        2                        2
##                  Huelva                    Leioa                   Lleida
##                        2                        2                        2
##                    Lugo                    Mahon                  Manacor
##                        2                        2                        2
##                Marbella         Mollet del Valles                  Orihuela
##                        2                        2                        2
##                 Ourense                 Palencia                 Pamplona
##                        2                        2                        2
##                   Parla                Pontevedra                  Requena
##                        2                        2                        2
##                    Reus             San Sebastian     Sant Cugat del Valles
##                        2                        2                        2
##                Santander                  Segovia                Tarragona
##                        2                        2                        2
##                  Teruel                  Tortosa                   Vitoria
##                        2                        2                        2
##                Zaragoza
##                        2
##
## [[3]]
##                A Coruña                 Alicante                    Camas
##                       3                        3                        3
##                   Gijon                  Granada    Jerez de la Frontera
##                       3                        3                        3
##                Leganes                   Malaga                   Mataro
##                       3                        3                        3
##                 Murcia       Palma de Mallorca Santiago de Compostela
##                       3                        3                        3
##                Sevilla                  Valencia                Valladolid
##                       3                        3                        3
```

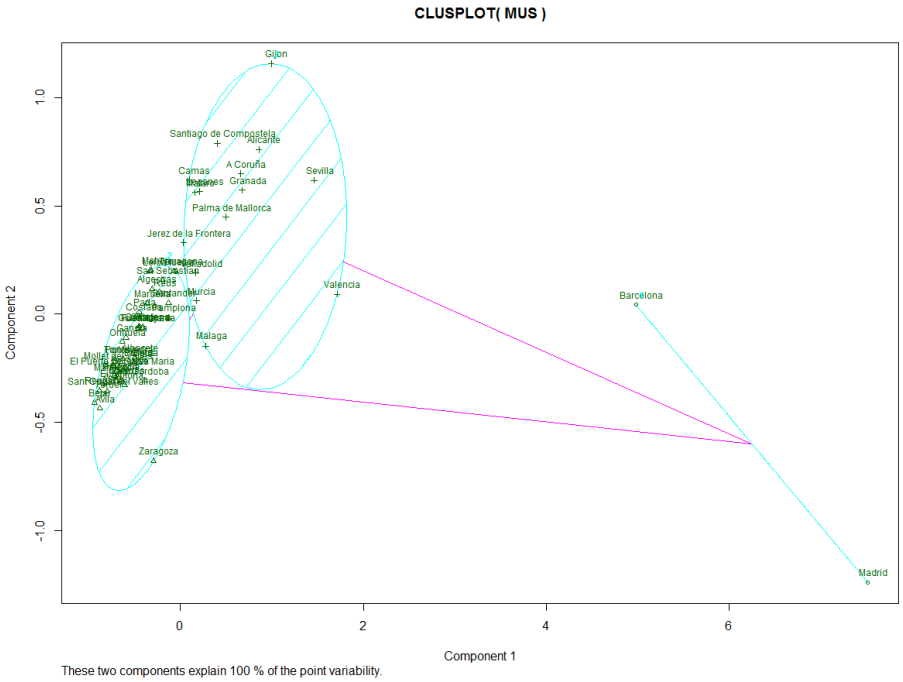


Figure 15. Plot of k-means clustering.

There is an R function, *NbClust*—in the package with the same name—which provides 30 indices to determine the best number of clusters and proposes the best clustering scheme from the different results obtained by varying all combinations of number of clusters, distance measures, and clustering methods. It was used using k-means, but can also be used with other algorithms, like the hierarchical ones. According to that function (and its associated indices), the best number of cluster is 2—Madrid and Barcelona in one, and the remaining cities in the other—, which does not seem to be a solution to our problem.

```
## *****************************************************************
## * Among all indices:
## * 9 proposed 2 as the best number of clusters
## * 7 proposed 3 as the best number of clusters
## * 1 proposed 5 as the best number of clusters
## * 1 proposed 6 as the best number of clusters
## * 1 proposed 8 as the best number of clusters
## * 2 proposed 21 as the best number of clusters
## * 2 proposed 24 as the best number of clusters
## * 3 proposed 26 as the best number of clusters
## * 2 proposed 27 as the best number of clusters
##
##                   ***** Conclusion *****
##
## * According to the majority rule, the best number of clusters is  2
##
##
## *****************************************************************
## $All.index
##         KL     CH Hartigan      CCC Scott Marriot   TrCovW  TraceW
## 2   2.4232 148.09  71.7722   0.8950 115.4  501.11 131.9596 30.3319
## 3   6.8440 202.80  24.1258   2.5011 185.0  332.21  16.5876 13.1595
## 4   1.1086 199.87   8.5279   3.5976 235.3  244.57  10.0910  9.0957
## 5   0.5151 172.83   2.8152   2.4728 258.0  256.20  10.0275  7.8350
##    Friedman   Rubin Cindex     DB Silhouette     Duda Pseudot2   Beale
## 2     10.85   3.692 0.2730 0.3246     0.8640   0.3470  99.7331  1.8469
## 3     17.62   8.511 0.1485 0.5946     0.5941   0.7375  16.0149  0.3468
## 4     30.92  12.313 0.1096 0.6478     0.5186   1.1236  -1.4296 -0.0943
## 5     40.47  14.295 0.0916 0.7477     0.4552   1.2424  -2.9264 -0.1561
##    Ratkowsky    Ball Ptbiserial   Gap    Frey McClain  Gamma  Gplus    Tau
## 2     0.6011 15.1660     0.9026 1.833 12.5460  0.0088 1.0000  0.000 102.42
## 3     0.5424  4.3865     0.5182 2.246  3.7957  0.1626 0.8767 24.296 345.56
## 4     0.4792  2.2739     0.4129 2.311  3.1841  0.3120 0.8684 24.632 325.17
## 5     0.4312  1.5670     0.3520 2.148 10.1027  0.4475 0.8683 21.226 279.79
##      Dunn Hubert SDindex Dindex   SDbw
## 2  1.1511 0.0196  160.76 0.5921 0.8085
## 3  0.0548 0.0185  119.38 0.3778 0.6861
## 4  0.0350 0.0190  150.57 0.2909 0.5930
## 5  0.0498 0.0193   46.39 0.2573 0.5335
##
## $Best.nc
##                     KL    CH Hartigan    CCC Scott Marriot TrCovW TraceW
## Number_clusters  6.00  21.0     24.0 21.000  27.0      26    3.0   3.00
## Value_Index     31.46 504.8    449.1  8.069 290.1    2571  115.4  13.11
##                    Friedman  Rubin Cindex     DB Silhouette   Duda PseudoT2
## Number_clusters       27   26.0 24.000 2.0000      2.000 3.0000     3.00
## Value_Index         1243 -719.3  0.041 0.3246      0.864 0.7375    16.01
##                    Beale Ratkowsky  Ball PtBiserial   Gap Frey McClain Gamma
## Number_clusters 2.000    2.0000  3.00     2.0000 3.000  5.0  2.0000     2
## Value_Index     1.847    0.6011 10.78     0.9026 2.246 10.1  0.0088     1
##                    Gplus   Tau  Dunn Hubert SDindex Dindex    SDbw
## Number_clusters     2   3.0 2.000      0     8.0      0 26.0000
## Value_Index         0 345.6 1.151      0    40.5      0  0.0031
```

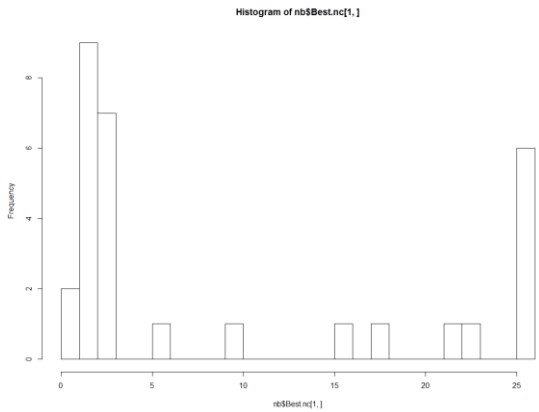


Figure 16. Histogram with the optimal number of clusters, based on the k-means algorithm and the 30 indices provided by function NbClust.

After that, the **PAM** (Partitioning Around Medioids) technique, which is the most common realization of **k-medoid clustering** algorithm, was used. Based on the optimum average silhouette width, the optimal number of cluster would be 28—the maximum value considered. But half of them only contain one element, which is not useful for our purposes.

So we assign *k* the value of 3, obtaining the following results:

```
## Medoids:
##                   ID Disp_Visits Traffic
## Palma de Mallorca 39     0.03513  0.6683
## Pontevedra        42    -0.29673 -0.5901
## Madrid            28     6.19081  4.4379
## Clustering vector:
##                 A Coruña                Albacete               Algeciras
##                        1                       2                       2
##                Alicante                   Avila               Barcelona
##                        1                       2                       3
##                   Bejar                  Burgos                 Caceres
##                        2                       2                       2
##                   Camas               Cartagena                 Cordoba
##                        1                       2                       2
##                 Coslada                  Eibar El Puerto de Santa Maria
##                        2                       2                       2
##             Fuenlabrada                  Gandia                   Getafe
##                        2                       2                       2
##                   Gijon                 Granada             Guadalajara
##                        1                       1                       2
##                  Huelva     Jerez de la Frontera                 Leganes
##                        1                       1                       1
##                   Leioa                  Lleida                    Lugo
##                        2                       2                       2
##                  Madrid                   Mahon                  Malaga
##                        3                       2                       1
##                 Manacor                 Marbella                  Mataro
##                        2                       2                       1
##       Mollet del Valles                  Murcia                 Orihuela
##                        2                       1                       2
##                 Ourense                Palencia        Palma de Mallorca
##                        2                       2                       1
##                Pamplona                   Parla              Pontevedra
##                        2                       2                       2
##                 Requena                    Reus           San Sebastian
##                        2                       2                       2
##   Sant Cugat del Valles               Santander   Santiago de Compostela
##                        2                       2                       1
##                 Segovia                 Sevilla               Tarragona
##                        2                       1                       1
##                  Teruel                 Tortosa                Valencia
##                        2                       2                       1
##             Valladolid                 Vitoria                Zaragoza
##                        1                       2                       2
```

The clusters are almost the same than those obtained using k-means clustering—only 2 cities (Huelva and Tarragona) are moved from the largest cluster to the other that previously

contained 15 elements.

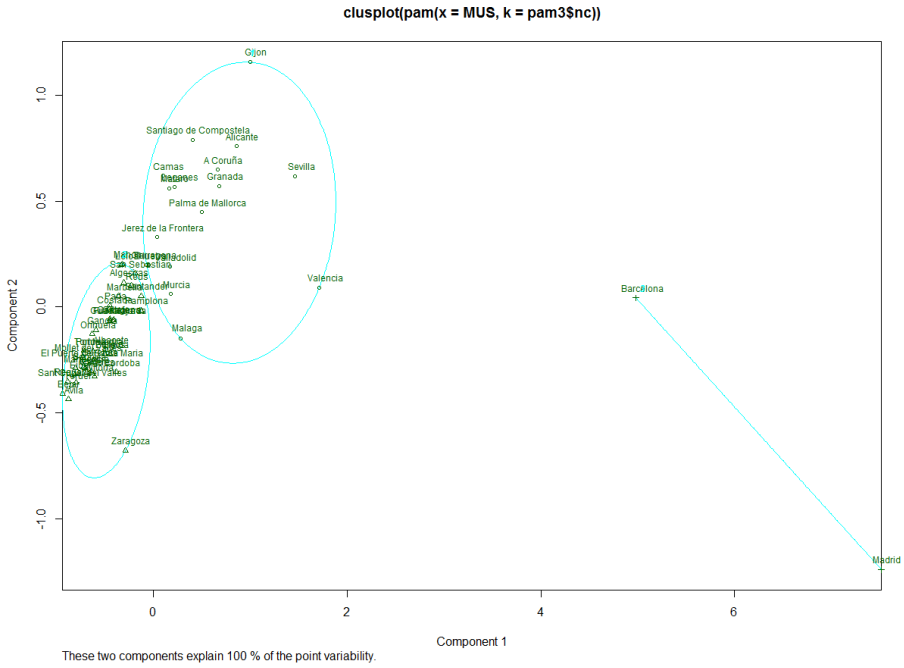


Figure 17. Plot of k-medoids clustering.

Another technique that was tested is based on **Expectation Maximization** (EM), whose implementation in R includes the following features:

- Normal mixture modeling (EM).
- EM initialization through a hierarchical clustering approach.
- Estimation of the number of clusters based on the BIC.

The best model according to this technique is the one with diagonal distribution, variable volumes, equal shapes and coordinate axes. The optimal number of clusters is, again, 3. The main differences between them and the ones previously detected are that Valencia and Seville are in the same cluster than Madrid and Barcelona, and the largest cluster is now slightly smaller (33 elements).

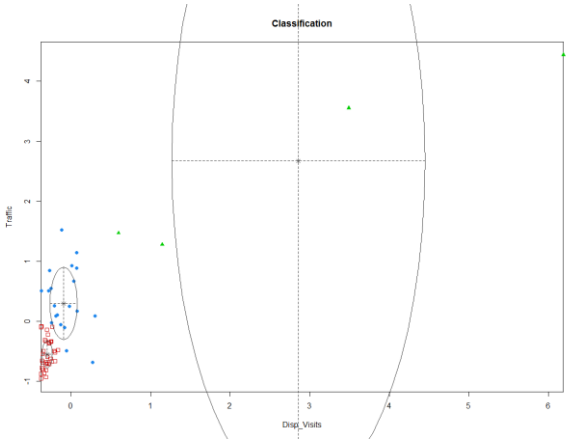


Figure 18. Plot of clustering based on EM technique.

```
##                A Coruña                 Albacete                 Algeciras
##                       1                        2                         2
##                Alicante                    Avila                 Barcelona
##                       1                        2                         3
##                   Bejar                   Burgos                   Caceres
##                       2                        2                         2
##                   Camas                 Cartagena                   Cordoba
##                       1                        2                         1
##                 Coslada                    Eibar El Puerto de Santa Maria
##                       2                        2                         2
##              Fuenlabrada                   Gandia                    Getafe
##                       2                        2                         2
##                   Gijon                  Granada               Guadalajara
##                       1                        1                         2
##                  Huelva     Jerez de la Frontera                  Leganes
##                       1                        1                         1
##                   Leioa                   Lleida                     Lugo
##                       2                        2                         2
##                  Madrid                   Mahon                   Malaga
##                       3                        2                         1
##                 Manacor                 Marbella                   Mataro
##                       2                        2                         1
##        Mollet del Valles                   Murcia                  Orihuela
##                       2                        1                         2
##                 Ourense                 Palencia         Palma de Mallorca
##                       2                        2                         1
##                Pamplona                    Parla                Pontevedra
##                       1                        2                         2
##                 Requena                     Reus             San Sebastian
##                       2                        2                         1
##    Sant Cugat del Valles                 Santander   Santiago de Compostela
##                       2                        1                         1
##                 Segovia                  Sevilla                 Tarragona
##                       2                        3                         1
##                  Teruel                  Tortosa                  Valencia
##                       2                        2                         3
##              Valladolid                  Vitoria                  Zaragoza
##                       1                        2                         1
```

The final technique that was tested was **Affinity Propagation**, a clustering algorithm based on the concept of "message passing" between data points. Unlike clustering algorithms such as k-means or k-medoids, Affinity Propagation does not require the number of clusters to be determined or estimated before running the algorithm. Like k-medoids, Affinity Propagation finds "exemplars", members of the input set that are representative of clusters.

With this technique, 8 clusters are found. Madrid and Barcelona form their own clusters, and the other ones respectively contain 2, 4, 5, 12, 13 and 19 elements.

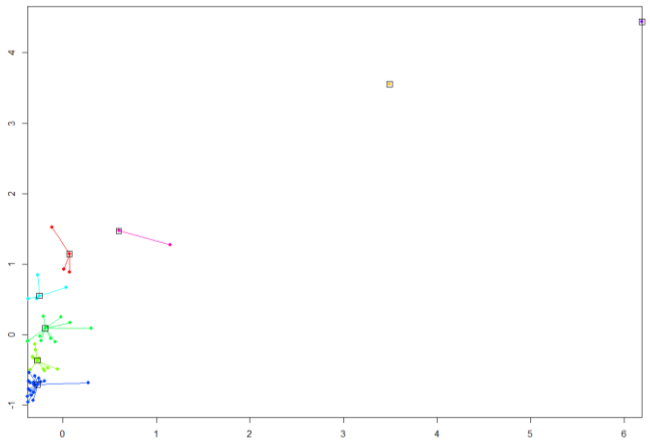


Figure 19. Plot of clustering based on Affinity Propagation.

```
## Clusters:
##    Cluster 1, exemplar Alicante:
##       A Coruña Alicante Gijon Granada
##    Cluster 2, exemplar Barcelona:
##       Barcelona
```

```
##    Cluster 3, exemplar Guadalajara:
##       Albacete Algeciras Burgos Cartagena Cordoba Coslada Fuenlabrada
##       Gandia Getafe Guadalajara Lleida Marbella Parla
##    Cluster 4, exemplar Huelva:
##       Huelva Jerez de la Frontera Leioa Mahon Malaga Murcia Pamplona Reus
##       San Sebastian Santander Tarragona Valladolid
##    Cluster 5, exemplar Leganes:
##       Camas Leganes Mataro Palma de Mallorca Santiago de Compostela
##    Cluster 6, exemplar Lugo:
##       Avila Bejar Caceres Eibar El Puerto de Santa Maria Lugo Manacor
##       Mollet del Valles Orihuela Ourense Palencia Pontevedra Requena
##       Sant Cugat del Valles Segovia Teruel Tortosa Vitoria Zaragoza
##    Cluster 7, exemplar Madrid:
##       Madrid
##    Cluster 8, exemplar Sevilla:
##       Sevilla Valencia
```

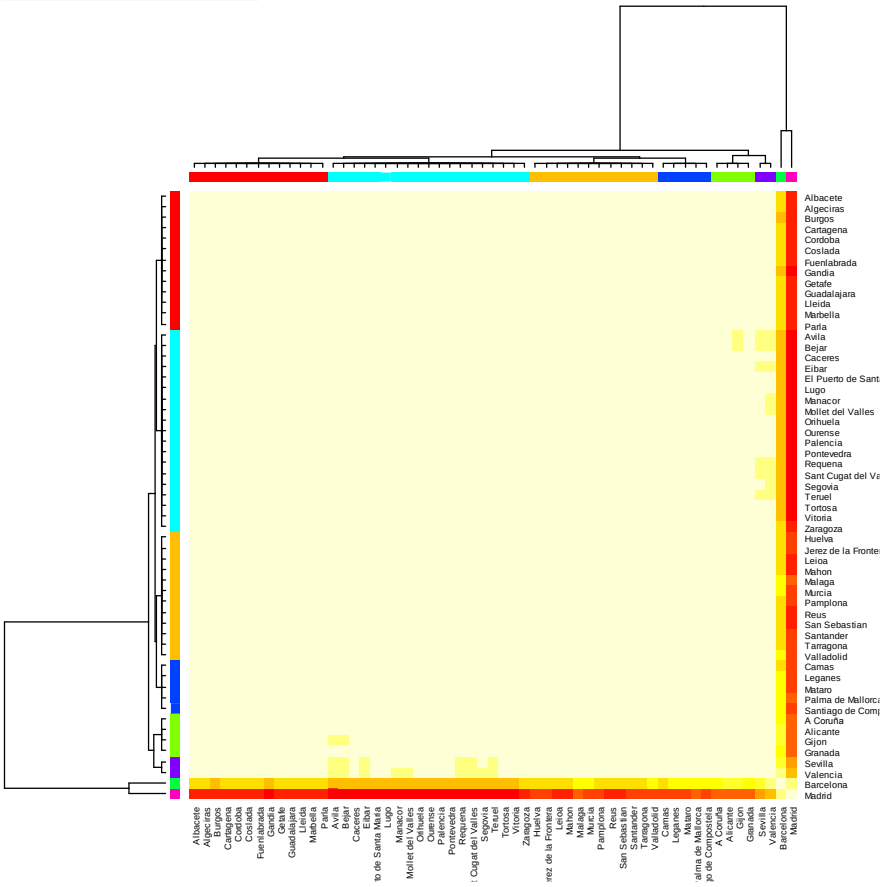


Figure 20. Heatmap of clustering based on Affinity Propagation.

3.3.2. Using all the variables

After performing clustering analysis using only the two main variables, all of them—previously standardized—were used. As previously mentioned, the correlation between the MUS Traffic and many of the other variables is usually very high (the variables whose correlation with that response variable is higher than 0.8 are shown below).

```
##                          Population                       Unemployment.rate
##                              0.8259                                  0.8283
##                          Land.lines Change.in..land.lines.since.2007....
##                              0.8304                                  0.8140
##      Change.in...cars.since.2007....             Industrial.activity.index
##                              0.8290                                  0.8148
##       Wholesale.trade.activity.index          Retail.trade.activity.index
##                              0.8410                                  0.8598
##               Economy.activity.index                Wholesale.trade.index
##                              0.8496                                  0.8409
##                  Retail.trade.index                          Tourist.index
```

```
##                      0.8526                      0.8401
##                Trade.index                MUS.Visits.All
##                      0.8251                      0.8204
##             MUS.VISITS.DIS              MUS.VISITS.EML
##                      0.8353                      0.8211
##             MUS.VISITS.REF              MUS.VISITS.SEA
##                      0.8243                      0.8263
##             MUS.VISITS.SEO              MUS.VISITS.SOC
##                      0.8264                      0.8203
##                    MUS.SEO                    MUS.SEM
##                      0.8025                      0.8167
##       Manufacturer.VISITS.ALL       Manufacturer.VISITS.AFF
##                      0.8158                      0.8182
##       Manufacturer.VISITS.DIS       Manufacturer.VISITS.EML
##                      0.8305                      0.8124
##       Manufacturer.VISITS.REF       Manufacturer.VISITS.SEA
##                      0.8370                      0.8257
##       Manufacturer.VISITS.SEO       Manufacturer.VISITS.SOC
##                      0.8268                      0.8227
##                  MUS.TRAFFIC            MUS.REGISTRATIONS
##                      1.0000                      0.9546
##                   MUS.TESTS                  MUS.ORDERS
##                      0.8108                      0.9558
##      Manufacturer.TRAFFIC       Manufacturer.REGISTRATIONS
##                      0.9884                      0.9407
##        Manufacturer.TESTS         Manufacturer.ORDERS
##                      0.8580                      0.9386
```

Though it was not detailed in the corresponding Subsection, and due to the correlation with population, almost every variable has a high variance and several outliers—almost the same in every case. I.e., if a city differs from another in one aspect, they will likely differ in others, and vice versa. If this happens, it may have the effect that distant cities (in terms of similarity) would be even more separated—so the within-cluster variations would be larger—, and similar cities would be even closer.

But variables must be carefully selected. If an explanatory variable is not related to the target variable, adding it to the model would have the effect of spreading the elements. In our case, none of the variables are even marginally normally distributed, hence any pair of them are not jointly normally distributed, so uncorrelation does not imply independence. Otherwise we could have thought of discarding those variables uncorrelated with the Traffic.

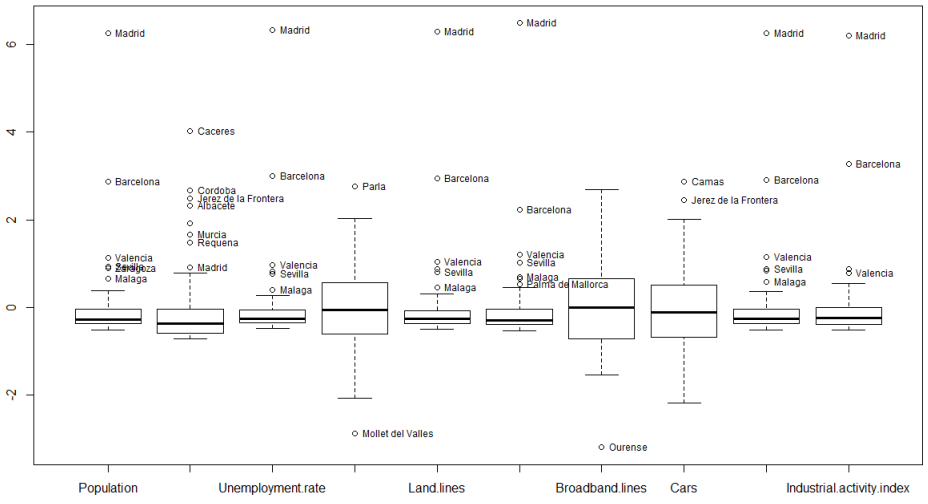


Figure 21. Boxplots of some of the socio-demographic and economic variables per GMA.

The R script developed (Clusters_all.R) can be found in the Annex of this document, Subsection A1.2. Since its results are not definite—and in some cases similar to the previous ones—, just a few of them are mentioned below.

Hierarchical clusters are almost the same than when using only Traffic and Display Visits: the optimal number of clusters, regardless of the method used, is 3: Madrid and Barcelona form their own clusters, and the remaining GMAs the third one.

Using **K-means clustering**, the best number of clusters would be 5. But, as seen in the figure below, two of them—cluster 2 and 4—overlap, so the classification is not optimal.

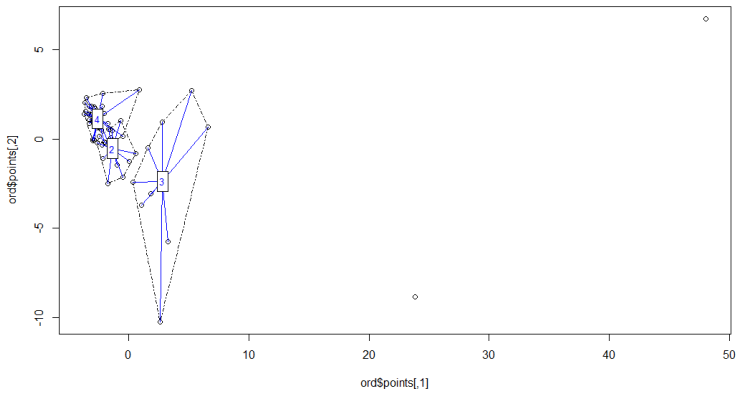


Figure 22. Plot of k-means clustering, when using all the variables.

The **PAM** technique offers better results. Based on the optimum average silhouette width, the optimal number of cluster would be 28, but most of them with a single element. If we assign *k* the value of 5, non-overlapping clusters are detected (but Madrid and Barcelona form their own clusters).

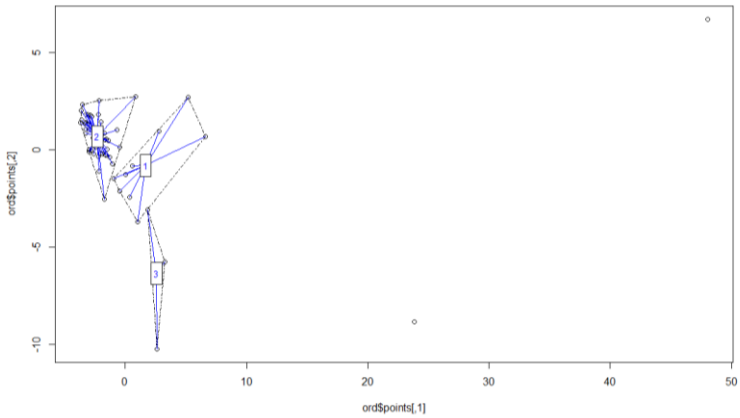


Figure 23. Plot of k-medoids clustering, when using all the variables.

The best model according to the **Expectation Maximization** technique is now the one with diagonal distribution, equal volumes and shapes, and coordinate axes. The optimal number of clusters is 13. But some of them overlap each other, and some others only contain a single element. The **Affinity Propagation** technique gives very similar results.

3.3.3. Using Principal Components of all the variables

In order to reduce the number of variables, Principal Component Analysis (PCA) was used. Since the number of original variables was actually greater than the number of observations, PCA could not be directly performed. Instead, new matrices were built, one per category of variables (Demography, Cars, Internet, Economy, Web and Dealers). The PCA was then performed in each of these matrices, and the Principal Components that explained the highest amount of the variation in the corresponding dataset were added to a new matrix. If the PCA did not reduce the number of variables in a significant way, the original variables with were still used—since their meaning is more intuitive than that of built indices.
In any case, the results remain almost the same.

3.3.4. Conclusions

Clustering analysis did not provide satisfactory results, regardless of the number of variables used. Several GMAs—and unfortunately the main ones; see Footnote 14—form their own cluster, so this analysis does not provide information about which other GMA should be assigned to the other group. And the size of the clusters with several elements is sometimes large, so the distance between elements may vary significantly, and hence randomly splitting them into two groups may lead to not-so-similar groups.
Moreover, even if the slope of the regression lines of the main variables is considered, we lose information about how these variables vary over time. Their variation is far from being linear, and it is what best characterizes these variables. That led us to try a different—and final—approach.

3.4. Distance measures for time series and nearest neighbors

3.4.1. Distance measure between time series

As briefly mentioned in the previous Subsection, we want a measure of similarity between the Traffic time series of any two GMAs. That metric should consider not only the Traffic average during the pre-test period but also how it changes.
There are several distance or dissimilarity measures for time series, many of them implemented in an R package called *TSclust*. The three methods that seemed more appropriate to the Traffic time series, due to their nature, were:

1. Dynamic Time Warping (DTW), which is—or used to be—one of the most used algorithm for measuring dissimilarity between time series, especially in speech recognition.

2. Temporal Correlation and Raw Values method. Using it, between two time series covers dissimilarity on both raw values and temporal correlation behaviors. A parameter controlling the weight of the dissimilarity between dynamic behaviors is required, as well as a method for the raw data discrepancy. That method can be either the Euclidean distance, the Fréchet distance, or DTW (using Manhattan as local distance). The latter method was the one used in this project.
3. **Discrete Wavelet Transform** (DWT). A wavelet transform is the representation of a function by wavelets, i.e., mathematical functions used to divide a given function or continuous-time signal into different scale components. Usually one can assign a frequency range to each scale component. Each scale component can then be studied with a resolution that matches its scale. Wavelet transforms are now being adopted for a vast number of applications, often replacing Fourier transforms, since the former have many advantages over the latter: temporal resolution—they capture not only frequency but also location information (location in time)—, they represent functions that have discontinuities and sharp peaks much better, and they allow to accurately deconstruct and reconstruct finite, non-periodic and/or non-stationary signals.

The three methods mentioned above were used and, as expected, DWT gave better results. The problem with DTW is that it assumes that one time series is a non-linear time-stretched version of the other—so there is a temporal alignment—and that actual values are on the same scale.

So in this Section only the definite results, achieved using DWT, are explained. The first step is to build two distance matrices for Traffic and Display Visits.

Table 6. First rows and columns of the distance matrix based on DWT.

row.names	A.Coru\u00e3a	Albacete	Algeciras	Alicante	Avila	Barcelona	Bejar	Burgos
A.Coru\u00e3a	0.00000000	0.211993163	0.16504905	0.08225513	0.27542747	0.3860368	0.27873733	0.21575057
Albacete	0.21199316	0.000000000	0.05837873	0.25695999	0.06908688	0.5925827	0.07002201	0.03813601
Algeciras	0.16504905	0.058378726	0.00000000	0.22077782	0.12473128	0.5436209	0.12494490	0.08147452
Alicante	0.08225513	0.256959991	0.22077782	0.00000000	0.31163663	0.3665533	0.31670883	0.25053901
Avila	0.27542747	0.069086879	0.12473128	0.31163663	0.00000000	0.6555205	0.01048438	0.06325555
Barcelona	0.38603679	0.592582681	0.54362090	0.36655332	0.65552051	0.0000000	0.65949745	0.59404920
Bejar	0.27873733	0.070022013	0.12494490	0.31670883	0.01048438	0.6594975	0.00000000	0.06934295
Burgos	0.21575057	0.038136007	0.08147452	0.25053901	0.06325555	0.5940492	0.06934295	0.00000000

To have a better idea of what these distances mean, the following step is to normalize them with respect to the maximum distance between any pair of GMAs. The two GMAs more distant from each other are Madrid and Bejar, in terms of traffic, and Madrid and Leioa, in terms of Display Visits. The distance between those pairs of GMAs is now 1 (and any other distance will be lower).

The figure displays the representation the location of each GMA. As the distance matrix defines an n-dimensional space, where n=57, a dimensionality reduction is required. PCA or Factorial Analysis, for example, could have been used, but Multidimensional Scaling (MDS),

also known as Principal Coordinates Analysis, was preferred, since it best preserves a priori the neighborhood relationships. This scaling takes a set of dissimilarities and returns a set of points such that the distances between the points are approximately equal to the dissimilarities.

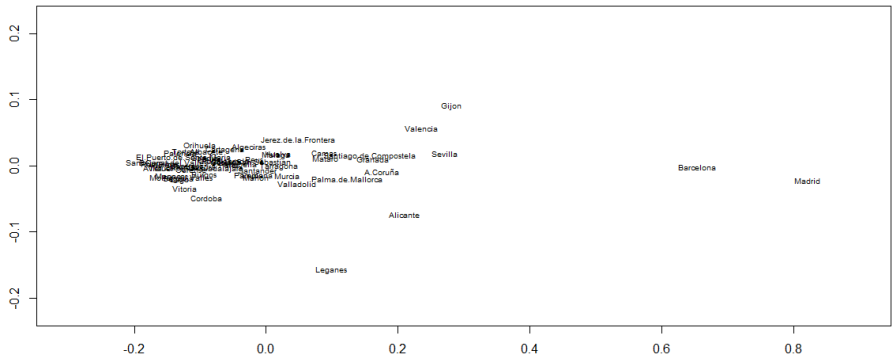


Figure 24. Location of each GMA based on their Traffic time series (since the dimensionality has been reduced, the real location of each GMA with respect to the other may differ slightly).

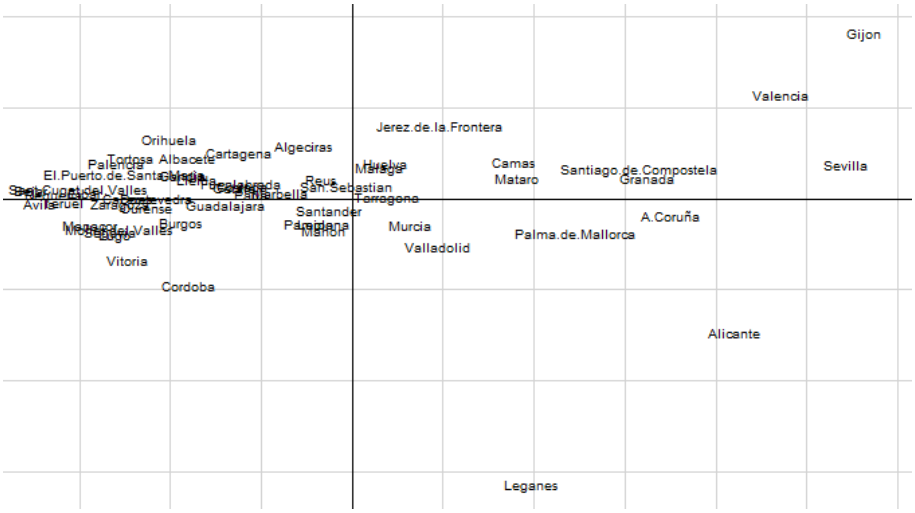
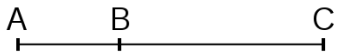


Figure 25. Detail of the location of each GMA—except Madrid and Barcelona—based on their Traffic time series (since the dimensionality has been reduced, the real location of each GMA with respect to the other may differ slightly).

3.4.2. Matching nearest neighbors

After that, we look for nearest neighbors, associating each GMA to its nearest one. To form pairs, we want this relation to be mutual—i.e., for two GMAs to be their respective neighbors, the distance between them must be lower than the distance between one of them and any other GMA, and both GMAs must satisfy that condition. To put an example, let's suppose we are in $\Re$ instead that in $\Re^{57}$, where any three cities—that we call *A*, *B* and *C*—would be on a line. If *B* is between *A* and *C*, and closer to *A*, the nearest city to *A* will be *B*, the nearest city to *B* will be *C*, and the nearest city to *C* will be *B*. Then, the only pair of nearest neighbors is the one formed by *A* and *B*, which are hence matched. *C* would be unmatched, as its nearest

neighbor is *B*, but the nearest neighbor of *B* is not *C*.



Using the distance matrix of Traffic, 14 pairs of GMAs are matched:

```
##         Ciudad                Vecina
## 1     A.Coruña               Granada
## 2     Albacete                Gandia
## 3        Avila                 Bejar
## 4    Barcelona                Madrid
## 5        Camas                Mataro
## 6        Eibar Sant.Cugat.del.Valles
## 7  Fuenlabrada                 Parla
## 8       Huelva                Malaga
## 9         Lugo     Mollet.del.Valles
## 10       Mahon             Santander
## 11      Murcia            Valladolid
## 12     Ourense            Pontevedra
## 13    Palencia               Tortosa
## 14     Sevilla               Valencia
```

3.4.3. Assignment to Test and Control groups

Now we are able to assign one GMA of each couple to the Test group, and the other to the Control group. The question is which one. For the first couple, it doesn't matter. For the second and subsequent couples, since the objective is that the aggregate time series of a group is as similar to the other's as possible, we should test all the possible pairs of groups that could be created. E.g., if the couples we have found are $\{a_1,b_1\}$, $\{a_2,b_2\}$ and $\{a_3,b_3\}$, let's say we assign *A* to the Test group, and *B* to the Control group. Then we should compare four pairs of groups:

$\{a_1,a_2,a_3\}$ and $\{b_1,b_2,b_3\}$

$\{a_1,a_2,b_3\}$ and $\{b_1,b_2,a_3\}$

$\{a_1,b_2,a_3\}$ and $\{b_1,a_2,b_3\}$

$\{a_1,b_2,b_3\}$ and $\{b_1,a_2,a_3\}$

If we don't consider the first couple—whose first element always belongs to the Test group—and only the first element of the remaining couples (a_2 and a_3), the possible Test groups may contain none, one or the two of these elements. For n couples, the possible Test groups may contain all the combinations of 0 to n-1 elements of the set $\{a_2,\ldots,a_n\}$:

$$\sum_{k=0}^{n-1} \binom{n-1}{k} = 2^{n-1}$$

This is the number of ways the Test group can be created (the Control group contains the remaining n elements of the couples). An R function was created to calculate the DWT distances between the time series of the possible Test and Control groups[16], and select the

[16] The time series of a group refers to the time series of the sum of Traffic of all the GMAs that belongs to it.

groups for which that distance is minimum[17]. After applying that function to the 14 couples detected, the Test and Control groups that appear in the table below were detected. 57% of the final sample Traffic (and hence 32% of the original sample) comes from the GMAs included in these first Test and Control groups.

Table 7. First Test and Control groups.

TEST			CONTROL			OUT		
CITY	AVERAGE WEEKLY TRAFFIC	TRAFFIC SLOPE	CITY	AVERAGE WEEKLY TRAFFIC	TRAFFIC SLOPE	CITY	AVERAGE WEEKLY TRAFFIC	TRAFFIC SLOPE
A Coruña	18.30	0.0481	Granada	17.91	0.0536	Algeciras	8.87	-0.0331
Albacete	5.72	-0.0173	Gandia	5.67	-0.0013	Alicante	20.23	0.1979
Avila	1.85	0.0171	Bejar	1.62	0.0019	Burgos	5.52	0.0496
Barcelona	41.58	0.1547	Madrid	49.42	0.1240	Caceres	4.16	0.0192
Camas	14.57	-0.0001	Mataro	14.62	0.0067	Cartagena	7.06	-0.0223
Eibar	3.01	0.0008	Sant Cugat del Valles	2.84	-0.0056	Cordoba	5.73	0.0935
Mahon	9.33	0.0562	Santander	9.61	0.0452	Coslada	7.15	0.0060
Malaga	10.91	0.0086	Huelva	10.87	0.0078	El Puerto de Santa M	4.08	-0.0097
Mollet del Valles	4.01	0.0319	Lugo	3.81	0.0418	Getafe	6.89	0.0064
Parla	7.31	0.0192	Fuenlabrada	7.12	0.0119	Gijon	23.57	0.0086
Pontevedra	4.86	0.0147	Ourense	4.59	0.0149	Guadalajara	6.84	0.0299
Sevilla	23.13	0.0304	Valencia	21.39	-0.0156	Jerez de la Frontera	12.38	-0.0198
Tortosa	4.27	-0.0354	Palencia	3.85	-0.0280	Leganes	14.97	0.2598
Valladolid	12.32	0.0620	Murcia	11.56	0.0380	Leioa	9.23	0.0435
						Lleida	5.88	-0.0094
						Manacor	3.22	0.0346
						Marbella	8.13	0.0240
						Orihuela	5.30	-0.0310
						Palma de Mallorca	16.01	0.0454
						Pamplona	9.17	0.0661
						Requena	2.31	0.0000
						Reus	9.28	0.0112
						San Sebastian	9.88	0.0030
						Santiago de Compost	17.56	-0.0130
						Segovia	3.78	0.0515
						Tarragona	11.03	0.0285
						Teruel	2.46	0.0160
						Vitoria	4.22	0.0667
						Zaragoza	4.00	0.0279
	Sum: 161.14 (28%)	Mean: 0.0279		Sum: 164.86 (29%)	Mean: 0.0211		Sum: 248.90 (43%)	Mean: 0.0328

Then the results—not only of Traffic but also of Display Visits—are compared to those that are obtained if those 28 GMAs would have been randomly assigned to the Test and Control groups—even without matching couples—, and also to the results obtained considering 56 of the 57 GMAs—since the maximum number of GMAs that could be matched must be even.

17 Since the total number of GMAs is 57, the maximum number of iterations is $2^{\lfloor 57/2 \rfloor - 1} = 134{,}217{,}728$.

Table 8. Comparison between optimal and random assignment of the first 14 couples to Test and Control groups[18].

	TRAFFIC		
	Random assignment of 56 GMAs	Random assignment of the first 28 GMAs selected	Optimal assignment of the first 14 couples detected
Distance	1.595	0.564	0.229
Correlation	0.831	0.773	0.809

	DISPLAY VISITS		
	Random assignment of 56 GMAs	Random assignment of the first 28 GMAs selected	Optimal assignment of the first 14 couples detected
Distance	2.088	0.468	0.513
Correlation	0.992	0.982	0.981

As seen in the table above, the results obtained are good: the distance between the time series of Test and Control groups when the assignment is optimal is less than half than that distance when the assignment is random—remember that, after normalization, the maximum distance between any pair of GMAs is 1. When considering Display Visits, the results are not so good (compared to random assignment), because the optimization was based on Traffic.

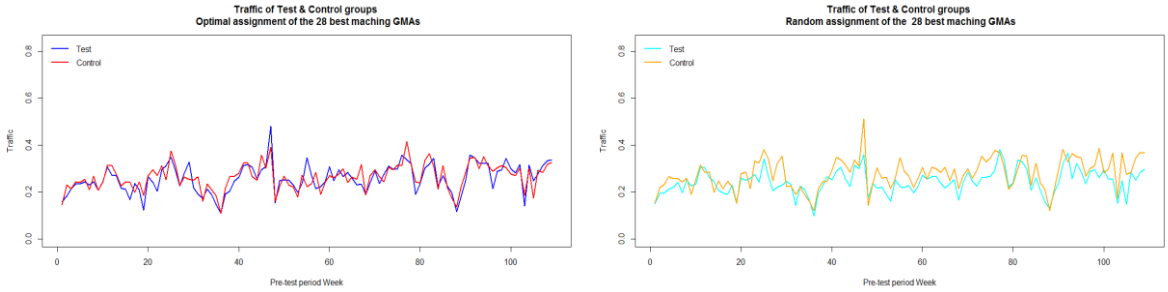


Figure 26. Plots of Traffic of Test and Control groups formed by the 28 best matching GMAs when they are either optimally or randomly assigned to each group.

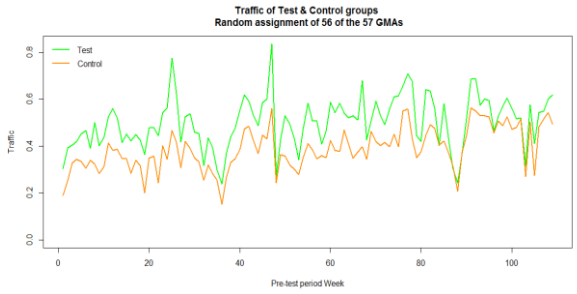


Figure 27. Plot of Traffic of Test and Control groups formed by 56 GMAs (the 57th is randomly discarded) when they are randomly assigned to each group.

18 Bear in mind that distance—based on GMA—is a much better indicator of similarity than correlation, which is not affected, for instance, by the amplitude differences.

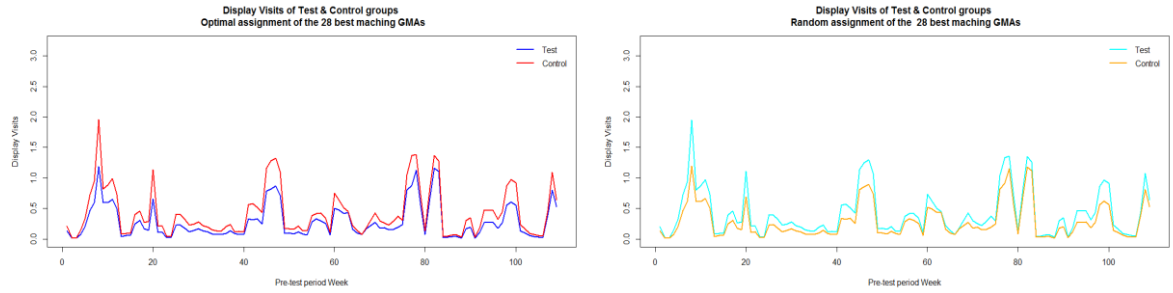


Figure 28. Plots of Display Visits of Test and Control groups formed by the 28 best matching GMAs when they are either optimally or randomly assigned to each group.

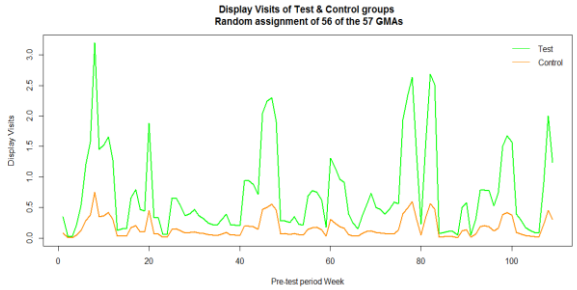


Figure 29. Plot of Display Visits of Test and Control groups formed by 56 GMAs (the 57th is randomly discarded) when they are randomly assigned to each group.

3.4.4. Obtaining more couples

Now that we have 28 GMAs whose Traffic, as mentioned, is the 57% of the final sample—and hence the 32% of the original sample—, the question is whether it is possible to increase this number and hence the size of the Test and Control groups, while still maintaining a low distance between their time series (i.e., without increasing the distance between them).

Two methods were tested, that were applied iteratively, both based on the idea that the remaining 29 cities—or at least most of them—are very close to others. If we look for those cities (see Table 7) in Figure 25, we'll notice that most of them form a quite dense cluster (possible exceptions are Alicante, Leganes, and some others to a lesser degree).

The first method involves a **dimensionality reduction** using MDS, which is the one that was previously used—with a Goodness of Fit of 0.969—to be able to plot the location of the GMAs based on the distances between them. Using MDS we obtained a pair of coordinates for each GMA. The distances—this time Euclidean instead of based on DWT—between each pair of those coordinates are calculated, and nearest neighbors are searched in the matrix obtained. Again, it is possible to detect the best matching couples, although—since we have lost or distorted information when reducing the dimensionality—they are not the same that were previously detected. In particular, now 16 couples are obtained: 7 couples had been previously detected, 4 are new couples, and the other 5 are incompatible (one of the partners is a member of the previous couples), and hence discarded. The 4 new couples are added to the 14

previous ones, so now 65% of the final sample Traffic (and hence 37% of the original sample) comes from the 36 GMAs we are currently considering.

The DWT distances between the time series of all the possible Test and Control groups were tested again, in two ways:

1. Starting from the beginning, as if the previous Test group had not been already detected—i.e., testing 2^{18-1}=131,072 possible Test groups.
2. Starting with the previous Test group and testing the 2^{4-1}=8 possible Test groups that could be formed when adding 4 additional GMAs.

The results were the same, so the second option was chosen thereafter. That may not work so well in successive iterations, but it drastically reduces the computation time, and the results, if not optimal anymore, will be good enough. They are certainly much better results—not only regarding to Traffic, but now also to Display Visits—than those obtained when randomly assigning the 36 GMAs to Test and Control groups, as seen below.

Table 9. Test and Control groups after adding a second set of GMAs.

TEST			CONTROL			OUT		
CITY	AVERAGE WEEKLY TRAFFIC	TRAFFIC SLOPE	CITY	AVERAGE WEEKLY TRAFFIC	TRAFFIC SLOPE	CITY	AVERAGE WEEKLY TRAFFIC	TRAFFIC SLOPE
A Coruña	18.30	0.0481	Granada	17.91	0.0536	Algeciras	8.87	-0.0331
Albacete	5.72	-0.0173	Gandia	5.67	-0.0013	Alicante	20.23	0.1979
Avila	1.85	0.0171	Bejar	1.62	0.0019	Burgos	5.52	0.0496
Barcelona	41.58	0.1547	Madrid	49.42	0.1240	Cartagena	7.06	-0.0223
Camas	14.57	-0.0001	Mataro	14.62	0.0067	Cordoba	5.73	0.0935
Eibar	3.01	0.0008	Sant Cugat del Valles	2.84	-0.0056	Coslada	7.15	0.0060
Mahon	9.33	0.0562	Santander	9.61	0.0452	El Puerto de Santa M	4.08	-0.0097
Malaga	10.91	0.0086	Huelva	10.87	0.0078	Getafe	6.89	0.0064
Mollet del Valles	4.01	0.0319	Lugo	3.81	0.0418	Gijon	23.57	0.0086
Pamplona	9.17	0.0661	Leioa	9.23	0.0435	Guadalajara	6.84	0.0299
Parla	7.31	0.0192	Fuenlabrada	7.12	0.0119	Jerez de la Frontera	12.38	-0.0198
Pontevedra	4.86	0.0147	Ourense	4.59	0.0149	Leganes	14.97	0.2598
Requena	2.31	0.0000	Teruel	2.46	0.0160	Lleida	5.88	-0.0094
San Sebastian	9.88	0.0030	Reus	9.28	0.0112	Manacor	3.22	0.0346
Sevilla	23.13	0.0304	Valencia	21.39	-0.0156	Marbella	8.13	0.0240
Tortosa	4.27	-0.0354	Palencia	3.85	-0.0280	Orihuela	5.30	-0.0310
Valladolid	12.32	0.0620	Murcia	11.56	0.0380	Palma de Mallorca	16.01	0.0454
Zaragoza	4.00	0.0279	Caceres	4.16	0.0192	Santiago de Compost	17.56	-0.0130
						Segovia	3.78	0.0515
						Tarragona	11.03	0.0285
						Vitoria	4.22	0.0667
	Sum: 186.51 (32%)	Mean: 0.0271		Sum: 189.99 (33%)	Mean: 0.0214		Sum: 198.40 (35%)	Mean: 0.0364

Table 10. Comparison between optimal and random assignment of the first 18 couples to Test and Control groups.

	TRAFFIC		
	Random assignment of 56 GMAs	Random assignment of the first 36 GMAs selected	Optimal assignment of the first 18 couples detected
Distance	1.595	2.360	0.221
Correlation	0.831	0.799	0.820

	DISPLAY VISITS		
	Random assignment of 56 GMAs	Random assignment of the first 36 GMAs selected	Optimal assignment of the first 18 couples detected
Distance	2.088	1.827	0.398
Correlation	0.992	0.987	0.985

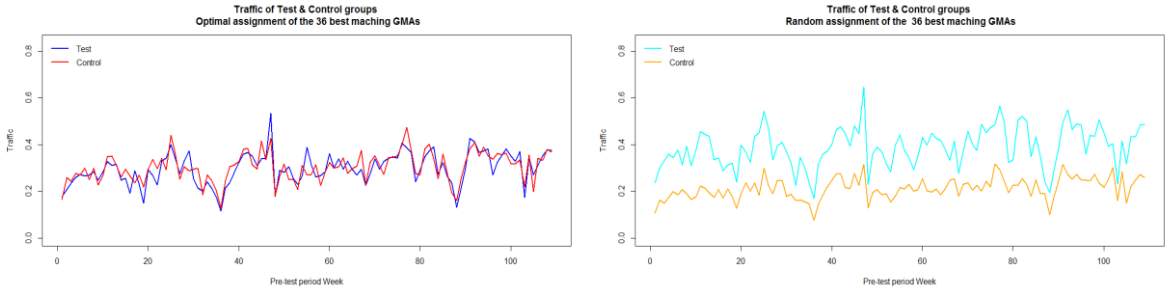


Figure 30. Plots of Traffic of Test and Control groups formed by the 36 best matching GMAs when they are either optimally or randomly assigned to each group.

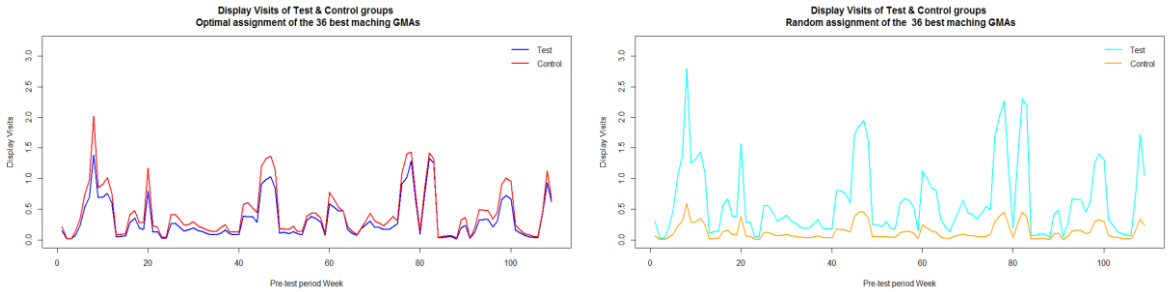


Figure 31. Plots of Display Visits of Test and Control groups formed by the 36 best matching GMAs when they are either optimally or randomly assigned to each group.

By **discarding the couples already assigned** to the Test and Control groups, the second method tries to add more couples. There were probably some GMAs between the remaining 21 ones, shown below, but as most of them—either reaming or already selected—form a quite dense cluster, the distances between most of these 21 GMAs are still relatively short.

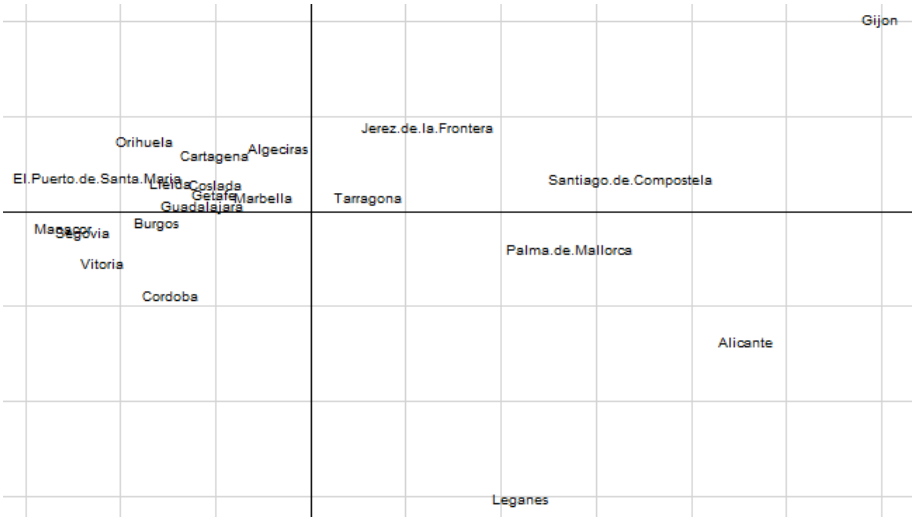


Figure 32. Location of the remaining 21 GMAs based on their Traffic time series.

This way 4 new couples are detected. Now we have 44 GMAs, and 79% of the final sample Traffic (and hence 45% of the original sample) comes from them.

Once again, the results are much better—Test and Control time series are much more similar—than those obtained when randomly assigning the 44 GMAs to Test and Control groups, as seen below.

Table 11. Test and Control groups after adding a third set of GMAs.

TEST			CONTROL			OUT		
CITY	AVERAGE WEEKLY TRAFFIC	TRAFFIC SLOPE	CITY	AVERAGE WEEKLY TRAFFIC	TRAFFIC SLOPE	CITY	AVERAGE WEEKLY TRAFFIC	TRAFFIC SLOPE
A Coruña	18.30	0.0481	Granada	17.91	0.0536	Algeciras	8.87	-0.0331
Albacete	5.72	-0.0173	Gandia	5.67	-0.0013	Alicante	20.23	0.1979
Avila	1.85	0.0171	Bejar	1.62	0.0019	Burgos	5.52	0.0496
Barcelona	41.58	0.1547	Madrid	49.42	0.1240	Cordoba	5.73	0.0935
Camas	14.57	-0.0001	Mataro	14.62	0.0067	Coslada	7.15	0.0060
Eibar	3.01	0.0008	Sant Cugat del Valles	2.84	-0.0056	El Puerto de Santa M	4.08	-0.0097
Getafe	6.89	0.0064	Cartagena	7.06	-0.0223	Gijon	23.57	0.0086
Jerez de la Frontera	12.38	-0.0198	Tarragona	11.03	0.0285	Guadalajara	6.84	0.0299
Mahon	9.33	0.0562	Santander	9.61	0.0452	Leganes	14.97	0.2598
Malaga	10.91	0.0086	Huelva	10.87	0.0078	Lleida	5.88	-0.0094
Manacor	3.22	0.0346	Segovia	3.78	0.0515	Marbella	8.13	0.0240
Mollet del Valles	4.01	0.0319	Lugo	3.81	0.0418	Orihuela	5.30	-0.0310
Pamplona	9.17	0.0661	Leioa	9.23	0.0435	Vitoria	4.22	0.0667
Parla	7.31	0.0192	Fuenlabrada	7.12	0.0119			
Pontevedra	4.86	0.0147	Ourense	4.59	0.0149			
Requena	2.31	0.0000	Teruel	2.46	0.0160			
San Sebastian	9.88	0.0030	Reus	9.28	0.0112			
Santiago de Compost	17.56	-0.0130	Palma de Mallorca	16.01	0.0454			
Sevilla	23.13	0.0304	Valencia	21.39	-0.0156			
Tortosa	4.27	-0.0354	Palencia	3.85	-0.0280			
Valladolid	12.32	0.0620	Murcia	11.56	0.0380			
Zaragoza	4.00	0.0279	Caceres	4.16	0.0192			
Sum: 226.56 (39%)		Mean: 0.0225	Sum: 227.87 (40%)		Mean: 0.0222	Sum: 120.48 (21%)		Mean: 0.0502

Table 12. Comparison between optimal and random assignment of the first 22 couples to Test and Control groups.

	TRAFFIC		
	Random assignment of 56 GMAs	Random assignment of the first 44 GMAs selected	Optimal assignment of the first 22 couples detected
Distance	1.595	2.360	0.170
Correlation	0.831	0.826	0.854

	DISPLAY VISITS		
	Random assignment of 56 GMAs	Random assignment of the first 44 GMAs selected	Optimal assignment of the first 22 couples detected
Distance	2.088	0.199	0.456
Correlation	0.992	0.984	0.988

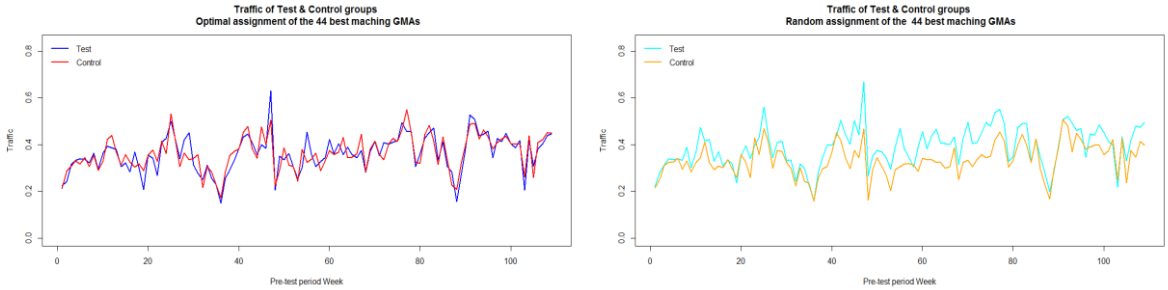


Figure 33. Plots of Traffic of Test and Control groups formed by the 44 best matching GMAs when they are either optimally or randomly assigned to each group.

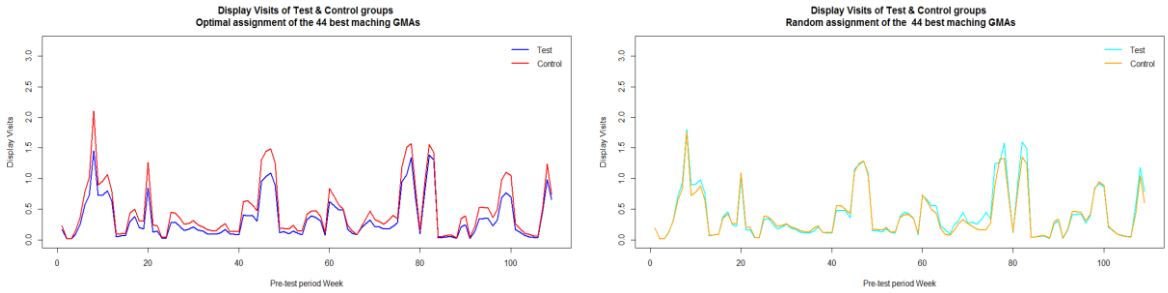


Figure 34. Plots of Display Visits of Test and Control groups formed by the 44 best matching GMAs when they are either optimally or randomly assigned to each group.

These two methods were applied iteratively, obtaining the following results:

Applying dimensionality reduction to the 21 remaining GMAs we had before the previous step, no new couples are found, so the 8 new GMAs that were added to the Test and Control groups in that step are discarded, and the remaining 13 GMAs are analyzed.

Figure 35. Location of the remaining 21 GMAs based on their Traffic time series.

3 new couples are detected, making it a total of 50 GMAs. 89% of the final sample Traffic (and hence 50% of the original sample) comes from them.

Table 13. Test and Control groups after adding a fourth set of GMAs.

TEST			CONTROL			OUT		
CITY	AVERAGE WEEKLY TRAFFIC	TRAFFIC SLOPE	CITY	AVERAGE WEEKLY TRAFFIC	TRAFFIC SLOPE	CITY	AVERAGE WEEKLY TRAFFIC	TRAFFIC SLOPE
Alicante	20.23	0.1979	Leganes	14.97	0.2598	Algeciras	8.87	-0.0331
A Coruña	18.30	0.0481	Granada	17.91	0.0536	Burgos	5.52	0.0496
Albacete	5.72	-0.0173	Gandia	5.67	-0.0013	Cordoba	5.73	0.0935
Avila	1.85	0.0171	Bejar	1.62	0.0019	Gijon	23.57	0.0086
Barcelona	41.58	0.1547	Madrid	49.42	0.1240	Lleida	5.88	-0.0094
Camas	14.57	-0.0001	Mataro	14.62	0.0067	Marbella	8.13	0.0240
Coslada	7.15	0.0060	Guadalajara	6.84	0.0299	Vitoria	4.22	0.0667
Eibar	3.01	0.0008	Sant Cugat del Valles	2.84	-0.0056			
El Puerto de Santa M	4.08	-0.0097	Orihuela	5.30	-0.0310			
Getafe	6.89	0.0064	Cartagena	7.06	-0.0223			
Jerez de la Frontera	12.38	-0.0198	Tarragona	11.03	0.0285			
Mahon	9.33	0.0562	Santander	9.61	0.0452			
Malaga	10.91	0.0086	Huelva	10.87	0.0078			
Manacor	3.22	0.0346	Segovia	3.78	0.0515			
Mollet del Valles	4.01	0.0319	Lugo	3.81	0.0418			
Pamplona	9.17	0.0661	Leioa	9.23	0.0435			
Parla	7.31	0.0192	Fuenlabrada	7.12	0.0119			
Pontevedra	4.86	0.0147	Ourense	4.59	0.0149			
Requena	2.31	0.0000	Teruel	2.46	0.0160			
San Sebastian	9.88	0.0030	Reus	9.28	0.0112			
Santiago de Compos	17.56	-0.0130	Palma de Mallorca	16.01	0.0454			
Sevilla	23.13	0.0304	Valencia	21.39	-0.0156			
Tortosa	4.27	-0.0354	Palencia	3.85	-0.0280			
Valladolid	12.32	0.0620	Murcia	11.56	0.0380			
Zaragoza	4.00	0.0279	Caceres	4.16	0.0192			
Sum: 258.02 (45%)		Mean: 0.0276	Sum: 254.97 (44%)		Mean: 0.0299	Sum: 61.91 (11%)		Mean: 0.0285

Table 14. Comparison between optimal and random assignment of the first 25 couples to Test and Control groups.

	TRAFFIC		
	Random assignment of 56 GMAs	Random assignment of the first 50 GMAs selected	Optimal assignment of the first 25 couples detected
Distance	1.595	1.453	0.251
Correlation	0.831	0.839	0.866

	DISPLAY VISITS		
	Random assignment of 56 GMAs	Random assignment of the first 50 GMAs selected	Optimal assignment of the first 25 couples detected
Distance	2.088	1.431	0.406
Correlation	0.992	0.982	0.990

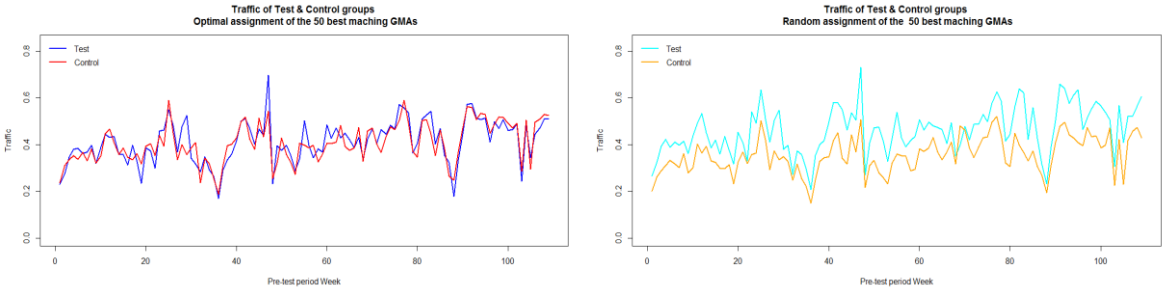


Figure 36. Plots of Traffic of Test and Control groups formed by the 50 best matching GMAs when they are either optimally or randomly assigned to each group.

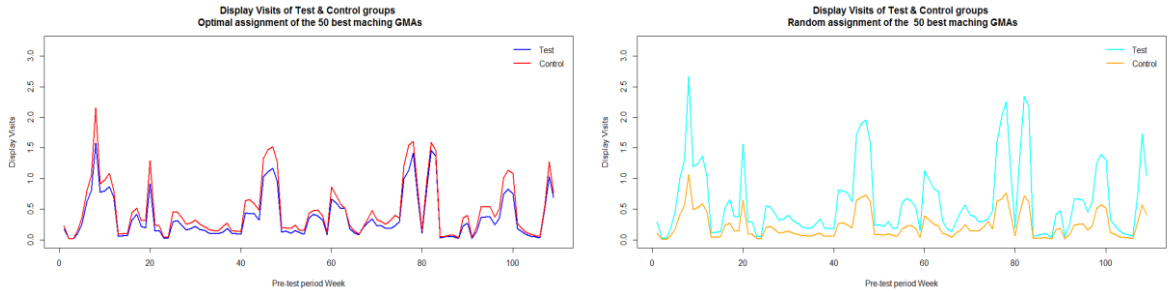


Figure 37. Plots of Display Visits of Test and Control groups formed by the 50 best matching GMAs when they are either optimally or randomly assigned to each group.

Applying dimensionality reduction to the 13 remaining GMAs we had before the previous step, 2 new couples are found. 94% of the final sample Traffic (and hence 53% of the original sample) comes from the 54 GMAs that we currently have.

Table 15. Test and Control groups after adding a fifth set of GMAs.

TEST			CONTROL			OUT		
CITY	AVERAGE WEEKLY TRAFFIC	TRAFFIC SLOPE	CITY	AVERAGE WEEKLY TRAFFIC	TRAFFIC SLOPE	CITY	AVERAGE WEEKLY TRAFFIC	TRAFFIC SLOPE
Alicante	20.23	0.1979	Leganes	14.97	0.2598	Cordoba	5.73	0.0935
A Coruña	18.30	0.0481	Granada	17.91	0.0536	Gijon	23.57	0.0086
Albacete	5.72	-0.0173	Gandia	5.67	-0.0013	Vitoria	4.22	0.0667
Avila	1.85	0.0171	Bejar	1.62	0.0019			
Barcelona	41.58	0.1547	Madrid	49.42	0.1240			
Camas	14.57	-0.0001	Mataro	14.62	0.0067			
Coslada	7.15	0.0060	Guadalajara	6.84	0.0299			
Eibar	3.01	0.0008	Sant Cugat del Valles	2.84	-0.0056			
El Puerto de Santa M	4.08	-0.0097	Orihuela	5.30	-0.0310			
Getafe	6.89	0.0064	Cartagena	7.06	-0.0223			
Jerez de la Frontera	12.38	-0.0198	Tarragona	11.03	0.0285			
Lleida	5.88	-0.0094	Burgos	5.52	0.0496			
Mahon	9.33	0.0562	Santander	9.61	0.0452			
Malaga	10.91	0.0086	Huelva	10.87	0.0078			
Manacor	3.22	0.0346	Segovia	3.78	0.0515			
Marbella	8.13	0.0240	Algeciras	8.87	-0.0331			
Mollet del Valles	4.01	0.0319	Lugo	3.81	0.0418			
Pamplona	9.17	0.0661	Leioa	9.23	0.0435			
Parla	7.31	0.0192	Fuenlabrada	7.12	0.0119			
Pontevedra	4.86	0.0147	Ourense	4.59	0.0149			
Requena	2.31	0.0000	Teruel	2.46	0.0160			
San Sebastian	9.88	0.0030	Reus	9.28	0.0112			
Santiago de Compost	17.56	-0.0130	Palma de Mallorca	16.01	0.0454			
Sevilla	23.13	0.0304	Valencia	21.39	-0.0156			
Tortosa	4.27	-0.0354	Palencia	3.85	-0.0280			
Valladolid	12.32	0.0620	Murcia	11.56	0.0380			
Zaragoza	4.00	0.0279	Caceres	4.16	0.0192			
Sum: 272.03 (47%)		Mean: 0.0261	Sum: 269.36 (47%)		Mean: 0.0283	Sum: 33.52 (6%)		Mean: 0.0562

Table 16. Comparison between optimal and random assignment of the first 27 couples to Test and Control groups.

	TRAFFIC		
	Random assignment of 56 GMAs	Random assignment of the first 54 GMAs selected	Optimal assignment of the first 27 couples detected
Distance	1.595	1.349	0.242
Correlation	0.831	0.844	0.873

	DISPLAY VISITS		
	Random assignment of 56 GMAs	Random assignment of the first 54 GMAs selected	Optimal assignment of the first 27 couples detected
Distance	2.088	1.831	0.401
Correlation	0.992	0.984	0.990

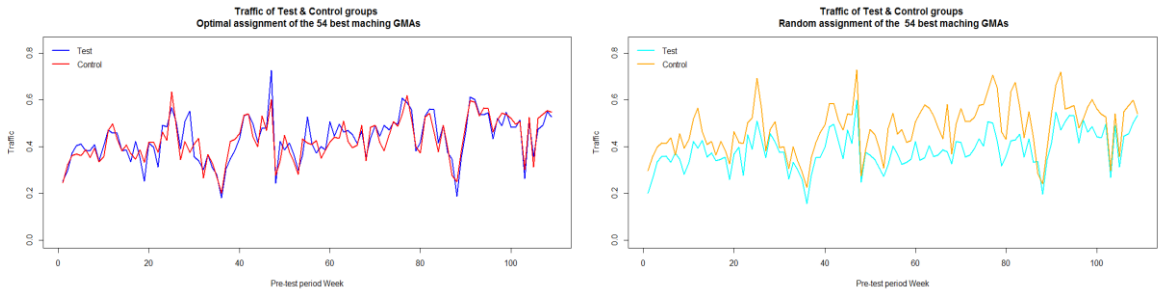


Figure 38. Plots of Traffic of Test and Control groups formed by the 54 best matching GMAs when they are either optimally or randomly assigned to each group.

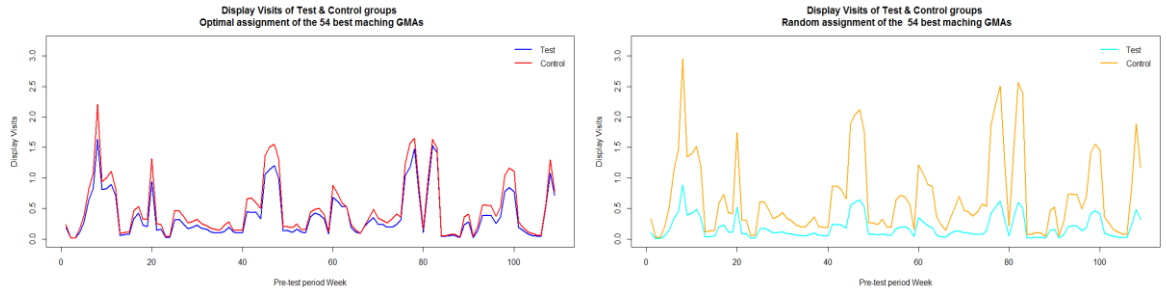


Figure 39. Plots of Display Visits of Test and Control groups formed by the 54 best matching GMAs when they are either optimally or randomly assigned to each group.

Now there is only a possible couple. Hence we analyze the remaining 3 GMAs, detecting that it is Gijon which cannot be paired. 96% of the final sample Traffic (and hence 54% of the original sample) comes from the other 56 GMAs.

Table 17. Test and Control groups after adding a sixth set of GMAs.

TEST			CONTROL			OUT		
CITY	AVERAGE WEEKLY TRAFFIC	TRAFFIC SLOPE	CITY	AVERAGE WEEKLY TRAFFIC	TRAFFIC SLOPE	CITY	AVERAGE WEEKLY TRAFFIC	TRAFFIC SLOPE
Alicante	20.23	0.1979	Leganes	14.97	0.2598	Gijon	23.57	0.0086
A Coruña	18.30	0.0481	Granada	17.91	0.0536			
Albacete	5.72	-0.0173	Gandia	5.67	-0.0013			
Avila	1.85	0.0171	Bejar	1.62	0.0019			
Barcelona	41.58	0.1547	Madrid	49.42	0.1240			
Camas	14.57	-0.0001	Mataro	14.62	0.0067			
Coslada	7.15	0.0060	Guadalajara	6.84	0.0299			
Eibar	3.01	0.0008	Sant Cugat del Valles	2.84	-0.0056			
El Puerto de Santa M	4.08	-0.0097	Orihuela	5.30	-0.0310			
Getafe	6.89	0.0064	Cartagena	7.06	-0.0223			
Jerez de la Frontera	12.38	-0.0198	Tarragona	11.03	0.0285			
Lleida	5.88	-0.0094	Burgos	5.52	0.0496			
Mahon	9.33	0.0562	Santander	9.61	0.0452			
Malaga	10.91	0.0086	Huelva	10.87	0.0078			
Manacor	3.22	0.0346	Segovia	3.78	0.0515			
Marbella	8.13	0.0240	Algeciras	8.87	-0.0331			
Mollet del Valles	4.01	0.0319	Lugo	3.81	0.0418			
Pamplona	9.17	0.0661	Leioa	9.23	0.0435			
Parla	7.31	0.0192	Fuenlabrada	7.12	0.0119			
Pontevedra	4.86	0.0147	Ourense	4.59	0.0149			
Requena	2.31	0.0000	Teruel	2.46	0.0160			
San Sebastian	9.88	0.0030	Reus	9.28	0.0112			
Santiago de Compost	17.56	-0.0130	Palma de Mallorca	16.01	0.0454			
Sevilla	23.13	0.0304	Valencia	21.39	-0.0156			
Tortosa	4.27	-0.0354	Palencia	3.85	-0.0280			
Valladolid	12.32	0.0620	Murcia	11.56	0.0380			
Vitoria	4.22	0.0667	Cordoba	5.73	0.0935			
Zaragoza	4.00	0.0279	Caceres	4.16	0.0192			
Sum: 276.25 (48%)		Mean: 0.0212	Sum: 275.09 (48%)		Mean: 0.0221	Sum: 23.57 (4%)		Mean: 0.0086

Table 18. Comparison between optimal and random assignment of the 28 possible couples to Test and Control groups.

	TRAFFIC		
	Random assignment of 56 GMAs	Random assignment of the 56 GMAs selected	Optimal assignment of the 28 couples detected
Distance	1.595	3.081	0.251
Correlation	0.831	0.859	0.876

	DISPLAY VISITS		
	Random assignment of 56 GMAs	Random assignment of the 56 GMAs selected	Optimal assignment of the 28 couples detected
Distance	2.088	1.944	0.422
Correlation	0.992	0.991	0.990

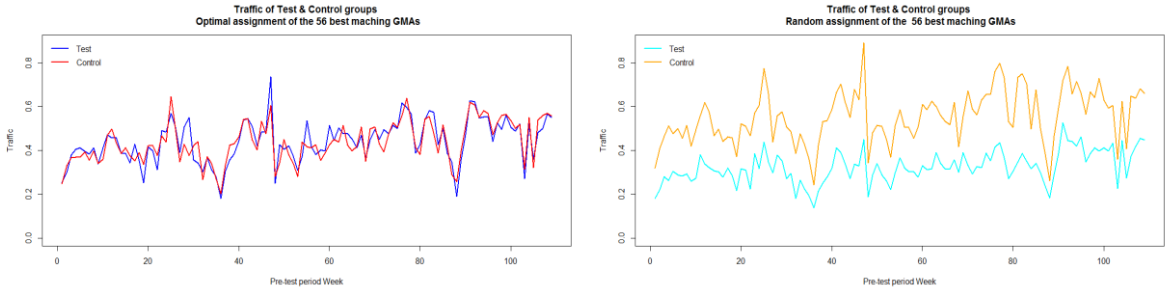


Figure 40. Plots of Traffic of Test and Control groups formed by the 56 best matching GMAs when they are either optimally or randomly assigned to each group.

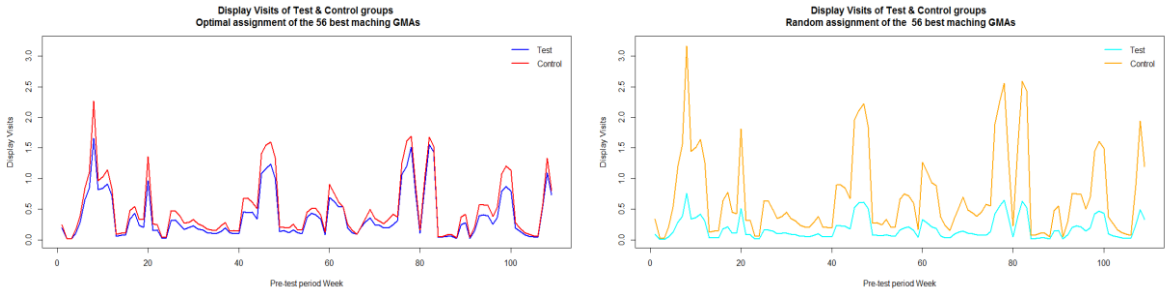


Figure 41. Plots of Display Visits of Test and Control groups formed by the 56 best matching GMAs when they are either optimally or randomly assigned to each group.

3.4.5. Final conclusions

If the 6 pairs of Test and Control groups that have been selected are compared, we detect that adding more GMAs to each group initially results in a shorter distance between them, but the distance increases as we add more GMAs. In particular, the distance decreases from 28 up to 44 GMAs, and it increases when more 44 GMAs are considered.

Table 19. Comparison between the different results.

	TRAFFIC						
	Optimal assignment						Random assignment
	28 GMAs	36 GMAs	44 GMAs	50 GMAs	54 GMAs	56 GMAs	56 GMAs
Distance	0.229	0.221	0.170	0.251	0.242	0.251	1.595
Correlation	0.809	0.820	0.854	0.866	0.873	0.876	0.831

	DISPLAY VISITS						
	Optimal assignment						Random assignment
	28 GMAs	36 GMAs	44 GMAs	50 GMAs	54 GMAs	56 GMAs	56 GMAs
Distance	0.513	0.398	0.456	0.406	0.401	0.422	2.088
Correlation	0.981	0.985	0.988	0.990	0.990	0.990	0.992

This seems a reasonable result, since we started with the most similar couples, and then we added less similar couples in each additional step. Therefore, we have to reach a **compromise** between number of GMAs and similarity between groups—the more GMAs we consider, the less similar Test and Control groups will be, and vice versa.

The **proposed solution**—if advertising budget is not an issue—is to launch the YouTube campaign in largest Test group possible (i.e., containing 28 GMAs), and make a double analysis after the test-period: considering both the whole groups and only the first 22 GMAs—the most similar ones to other—of each.

As the different graphs of Display Visits of Test and Control groups showed, both groups—regardless of their size—are not so similar in terms of that variable. That is not a problem at all, since the impact of a YouTube campaign on the Display Visits is already demonstrated—we have strong evidence that the Display Visits distance between the groups will change (it will increase or decrease, depending on which group previously had more Display Visits) during the test period. The effectiveness of YouTube advertising will be tested by checking if the Traffic distance between the groups, which was as small as possible, increases during the test period or not (assuming no other different stimuli in both groups).

> By the time this project was finished, the car manufacturer planned to run the YouTube campaign at the same time that another one on TV. That TV campaign would be run in the whole country, so it shouldn't involve an increase of the distance between groups, since that distance was close to zero during the pre-test period, when other TV campaigns had previously run.

3.4.6. Next steps

The function that detects the best possible groups, rather than starting from the beginning each time, considers the previous Test group and tests the possible new ones that could be formed when adding new GMAs from the new couples. Hence, $2^{13}+2^{4}+2^{4}+2^{3}+2^{2}+2^{1}=8{,}238$ instead of $2^{13}+2^{17}+2^{21}+2^{24}+2^{26}+2^{27}=220{,}340{,}224$ iterations—which would have taken several

weeks with the resources used—are made. But the first option might offer slightly better results[19], so an improvement could be made to perform this in a more optimal way.

As previously mentioned, this project only covered the first two stages of the whole experiment (see Chapter 2, Section 2.2 for further details). Prior to launching the YouTube campaign, the pre-test period should be extended up to the test period, and the analyses described in this Section—as well as the proposal of increase in advertising spend at the GMAs belonging to the Test group—should be made again, in order to detect any possible changes in the Test and Control groups.

In the future, having an indicator of Internet penetration per GMA could help to improve the analysis, removing the effect of the difference in population.

[19] E.g., let's say the first couples found are $\{a_1,b_1\}$ and $\{a_2,b_2\}$, and after reducing the dimensionality an additional couple, $\{a_3,b_3\}$, is detected. If the first iterations detected that the optimal Test group for that pair or couples is $\{a_1,b_2\}$, the R function that have been employed only tests the following possibilities: $\{a_1,b_2,a_3\}$ and $\{a_1,b_2,b_3\}$. But strictly speaking, these 4 possibilities should be tested: $\{a_1,a_2,a_3\}$, $\{a_1,a_2,b_3\}$, $\{a_1,b_2,a_3\}$ and $\{a_1,b_2,b_3\}$. If could be the case that we select the 4th choice, and it's the 1st choice the one with minimum distance.

List of Figures

List of Tables

List of Abbreviations and Acronyms

BIC	Bayesian Information Criteria
DTW	Dynamic Time Warping
DWT	Discrete Wavelet Transform
EM	Expectation Maximization
GIS	Geographic Information System
GMA	Geographic Marketing Area
INE	*Instituto Nacional de Estadística*
MDS	MultiDimensional Scaling
MUS	Model Under Study
PAM	Partitioning Around Medioids
PCA	Principal Component Analysis
POS	Point of Sale

SEA	Search Engine Advertising (equivalent to SEM)
SEM	Search Engine Marketing
SEO	Search Engine Optimization

Bibliography and Online Sources

Montero Manso, P. "A package for stationary time series clustering". 11 Jan. 2013. <http://eio.usc.es/pub/mte/descargas/ProyectosFinMaster/Proyecto_769.pdf>

Wang, X. et al . "Experimental comparison of representation methods and distance measures for time series". 26 Apr. 2010.

"Google." *Wikipedia*. Web. 10 Mar. 2014. <http://en.wikipedia.org/wiki/Google>

"Google AdWords – Online Advertising by Google." *Google.* Web. 10 Mar. 2014. <http://adwords.google.com>

Vaver, John, and Jim Koehler. "Measuring ad effectiveness using geo experiments." *Google*. Web. 10 Mar. 2014. <http://services.google.com/fh/files/blogs/geo_experiments_final_version.pdf>

"Estadística del Padrón Continuo a 1 de enero de 2013. Datos por municipios." *INE.* Web. 1 Jul. 2014. <http://www.ine.es/jaxi/tabla.do?path=/t20/e245/p05/a2013/l0/&file=00000001.px&type=pcaxis&L=0>

"Anuario Económico de España 2013." *La Caixa.* Web. 1 Jul. 2014. <http://www.anuarioeco.lacaixa.comunicacions.com/java/X?cgi=caixa.le_DEM.pattern&CLEAR=YES>

"Cluster analysis in R: determine the optimal number of clusters." *Stack Overflow.* Web. 12 Jul. 2014. <http://stackoverflow.com/questions/15376075/cluster-analysis-in-r-determine-the-optimal-number-of-clusters/15376462#15376462>

"Determining the number of clusters in a data set." *Wikipedia.* Web. 12 Jul. 2014. <http://en.wikipedia.org/wiki/Determining_the_number_of_clusters_in_a_data_set>

"CRAN Task View: Cluster Analysis & Finite Mixture Models." *CRAN.* Web. 12 Jul. 2014. <http://cran.r-project.org/web/views/Cluster.html>

"Data Mining Algorithms in R/Clustering/Expectation Maximization (EM)." *Wikibooks.* Web. 12 Jul. 2014. <http://en.wikibooks.org/wiki/Data_Mining_Algorithms_In_R/Clustering/Expectation_Maximization_(EM)>

"Affinity Propagation." *Wikipedia.* Web. 12 Jul. 2014. <http://en.wikipedia.org/wiki/Affinity_propagation>

"TSclust Dissimilarity Computation." *R Documentation.* Web. 20 Jul. 2014. <http://127.0.0.1:21986/library/TSclust/html/diss.html>

"Advertise – YouTube." *YouTube.* Web. 26 Aug. 2014. <https://www.youtube.com/yt/advertise/>

ANNEX

A1. Main R scripts (Chapter 3)

A1.1. Clusters_Traffic_and_Visits.R (Subsection 3.3.1)

```
rm(list=ls())
ruta<-"C:/Users/JuanJose/Google Drive/ANL/MASTERS/TECI/TFM/Universidad"
# Cambiar ruta segun proceda
setwd(ruta)


######## CARGA DATOS ########

MUS<-read.csv(file="traffic_and_visits.csv")
# Tabla con las 57 ciudades con la media semanal de
   # Trafico y Visitas de MUS estandarizadas (media y
   # desv. de la muestra)
rownames(MUS)<-MUS[,1]
ciudades<-rownames(MUS)
colnames(MUS)
MUS<-MUS[,-1]
colnames(MUS)<-c("Disp_Visits","Traffic")
cor(MUS)
# Alta correlacion entre ambas variables
summary(MUS)
boxplot(x=MUS)
# El Trafico presenta mayor dispersion que las Visitas
# Las Visitas presentan mas Outliers (y mas alejados)

# Para detectar los Outliers
boxdata <- with(MUS,boxplot(MUS),range=4)
for(i in 1:length(boxdata$group)){
   text(boxdata$group[i],boxdata$out[i],
        ciudades[if(which(MUS==boxdata$out[i])==
                       dim(MUS)[1])dim(MUS)[1]
                 else
                    which(MUS==
                             boxdata$out[i])%%dim(MUS)[1]],
        pos=4)
}
# Los principales Outliers para ambas variables son
   # Madrid y Barcelona, sobre todo en Visitas
# Visitas presenta menos Variabilidad, pero aparecen 4
   # Outliers mas: Valencia, Sevilla, Malaga y Zaragoza
   # (las otras ciudades mas pobladas, casi por orden
   # (se invierten Zaragoza y Malaga))

plot(MUS,xlab="Display Visits",ylab="Traffic")
text(x=MUS$Disp_Visits,y=MUS$Traffic,
     labels=rownames(MUS),cex=0.6,col="red")
# Se aprecia de nuevo que Madrid y Barcelona son
```

```
   # Outliers (seguidos de Valencia y Sevilla)

######## CLUSTERS JERARQUICOS ########

library(cluster)
distance<-dist(MUS)

# A la vista de la representacion anterior y
   # considerando los objetivos, se descarta los
   # metodos de:
      # WARD: muy sensible a Outliers y no buscamos
         # Clusters con similar nº de individuos
      # COMPLETO: tampoco interesa que los Clusters
         #tengan diametros similares
# Y en su lugar se consideraran:
      # MEDIO: Clusters con Varianzas similares
      # SIMPLE: Clusters irregulares. Busca el vecino
         # mas proximo.
      # CENTROIDE: la distancia entre Clusters es el
         # cuadrado de la distancia euclidea entre sus
         # centroides

fit_medio<-hclust(distance,method="average")
plot(fit_medio,hang=-1,main="Average Linkage Method")

# Una limitacion del Clustering Jerarquico es que
   # clasifica observaciones pero no explica en que
   # difiere cada clase/Cluster

# ¿Que numero de Clusters es el adecuado?
   # Podemos definir como metrica una usada en
      # Particiones: la Anchura de la SILUETA de los
      # Clusters.
      # Para cada observacion se compara su cercania al
      # resto de miembros de su mismo Cluster con su
      # cercania a miembros de otros Clusters.
   # Valores proximos a 1 indican que la observacion
      # esta bien asignada a su Cluster.
   # Valores proximos a 0 indican que deberia asignarse
      # a otro Cluster.
   # Usaremos en primer lugar la Anchura Media de las
      # Siluetas: si es superior a 0.7 indicara una
      # fuerte estructura; si es inferior a 0.5, que la
      # estructura es debil o inexistente.

silueta<-numeric(28)
silueta[1]<-1
for (i in 2:28){
   silueta[i]<-summary(silhouette(cutree(fit_medio,k=i),
                                  distance))$avg.width
   }
plot(silueta,type="b")
# La Anchura Media de la silueta baja de 0.7 a partir
   # de 3 Clusters.
```

```
# Pero además, a partir de 5 Clusters no hay buen
   # ajuste de algunas observaciones.
i<-5
plot(silhouette(cutree(fit_medio,k=i),distance))
# Parece razonable quedarse con 3 Clusters, incluyendo
   # el 2º y el 3º a los Outliers Madrid y Barcelona
   # (por separado)
        # addmargins(table(rownames(MUS),cutree(fit_medio,k=3)))
        # cutree(fit_medio, k=3)
plot(fit_medio,hang=-1,main="Average Linkage Method")
rect.hclust(fit_medio,k=3, border=2:4)

fit_simple<-hclust(distance,method="single")
plot(fit_simple,hang=-1,main="Single Linkage Method")
silueta<-numeric(28)
silueta[1]<-1
for (i in 2:28){
   silueta[i]<-summary(silhouette(cutree(fit_simple,k=i),
                                  distance))$avg.width
}
plot(silueta,type="b")
# DE NUEVO La Anchura Media de la silueta baja de 0.7
   # a partir de 3 Clusters.
# Pero además, a partir de 4 Clusters no hay buen
   # ajuste de algunas observaciones.
i<-4
plot(silhouette(cutree(fit_simple,k=i),distance))
# Parece razonable quedarse con 3 Clusters, incluyendo
   # el 2º y el 3º a los Outliers Madrid y Barcelona
   # (por separado)
plot(fit_simple,hang=-1,main="Single Linkage Method")
rect.hclust(fit_simple,k=3, border=2:4)

fit_centroide<-hclust(distance,method="centroid")
plot(fit_centroide,hang=-1,main="Centroid Linkage Method")
silueta<-numeric(28)
silueta[1]<-1
for (i in 2:28){
   silueta[i]<-summary(silhouette(cutree(fit_centroide,
                                         k=i),
                                  distance))$avg.width
}
plot(silueta,type="b")
# DE NUEVO La Anchura Media de la silueta baja de 0.7
   # a partir de 3 Clusters.
# Pero además, a partir de 5 Clusters no hay buen
   # ajuste de algunas observaciones.
i<-5
plot(silhouette(cutree(fit_centroide,k=i),distance))
# Parece razonable quedarse con 3 Clusters, incluyendo
   # el 2º y el 3º a los Outliers Madrid y Barcelona
   # (por separado)
plot(fit_centroide,hang=-1,
     main="Centroid Linkage Method")
```

```
rect.hclust(fit_centroide,k=3, border=2:4)


######## CLUSTERS NO JERARQUICOS ########
######## K-MEDIAS ########

# Para determinar el nº optimo de Clusters analizamos
   # la suma de cuadrados de las distancias de las
   # observaciones en un mismo Cluster, para todos los
   # Clusters: tendremos asi una medida de lo compacto
   # de cada Cluster
# Probamos hasta 28 Clusters ya que interesan Clusters
   # con al menos 1 par de observaciones
errores<-numeric(28)
for(i in 1:28){
   set.seed(2096730329)
   errores[i]<-kmeans(MUS,centers=i)$tot.withinss}
plot(errores,type="b")
# El punto de inflexion parece estar en 3 Clusters, a
   # partir de ahí la agrupación se hace más compleja
   # sin un alto impacto en la minimizacion del error
kmeans(MUS,centers=3)$cluster

# Tambien podemos comprobar las distancias entre
   # Clusters
dist_clusters<-numeric(28)
for(i in 1:28){
   set.seed(2096730329)
  dist_clusters[i]<-kmeans(MUS,centers=i)$betweenss
}
plot(dist_clusters,type="b")

# O, mejor aun, calcular el indice CH (Calinski-
   # Harabasz), que nos permite calcular el numero k
   # de clusters que maximiza la relacion entre una
   # distancia total intra-cluster baja y una distancia
   # inter-cluster alta
CH_index<-function(datos,k){
   set.seed(2096730329)
   fit<-kmeans(datos,k)
   CH_index<-(fit$betweenss/(k-1))/
      (fit$tot.withinss/(dim(datos)[1]-k))
}
CH_k<-numeric(28)
CH_k<-sapply(1:28,function(i) CH_index(MUS,i))
which(CH_k==max(CH_k))
plot(c(1:28),CH_k,type="b",xlab="k",ylab="CH(k)",xlim=c(2,28), xaxt='n')
axis(1,at=seq(0, 28, 2))

# Nos quedamos por tanto con 3 Clusters
set.seed(2096730329)
fit_kmedias<-kmeans(MUS,3)
summary(fit_kmedias)
str(fit_kmedias)
```

```
fit_kmedias$cluster
fit_kmedias$withins
addmargins(table(rownames(MUS),fit_kmedias$cluster))
Clusters<-sapply(1:3,function(i){
   fit_kmedias$cluster[fit_kmedias$cluster==i]})
print(Clusters)
library(fpc)
#plotcluster(MUS,fit_kmedias$cluster)
clusplot(MUS,fit_kmedias$cluster,color=F,shade=T,
         labels=2,cex=0.75)

######### 30 indices for determining the number of
# clusters and proposes to user the best clustering
# scheme from the different results obtained by
# varying all combinations of number of clusters,
# distance measures, and clustering methods

library(NbClust)
(nb <- NbClust(MUS, diss="NULL", distance = "euclidean",
               min.nc=2, max.nc=28, method = "kmeans",
               index = "alllong", alphaBeale = 0.1))
hist(nb$Best.nc[1,], breaks = max(na.omit(nb$Best.nc[1,])))

######## PAM ########

pamk.best<-pamk(data=MUS,krange=28,criterion="asw",
                critout=T)
plot(pam(MUS,pamk.best$nc))
plot(pam(MUS,2))
pam3<-pamk(data=MUS,krange=3,criterion="asw",
           critout=T)
pam(MUS,pam3$nc)
pamk.best$pamobject[[3]]
pam3$pamobject[[3]][pam3$pamobject[[3]]==1]
clusplot(pam(MUS,pam3$nc),lines=0,color=F,shade=F,
         labels=2,cex=0.75)

######### The optimal model according to BIC for EM
   # initialized by hierarchical clustering for
   # parameterized Gaussian mixture models####

library(mclust)
d_clust<-Mclust(as.matrix(MUS),G=1:28)
m.best<-dim(d_clust$z)[2]
cat("model-based optimal number of clusters:",m.best,
    "\n")
plot(d_clust)
d_clust$classification

######### AFFINITY PROPAGATION CLUSTERING ####
   # Affinity Propagation clusters data using a set of
      # real-valued pairwise data point similarities as
      # input. Each cluster is represented by a cluster
```

```
      # center data point (the so-called exemplar). The
      # method is iterative and searches for clusters
      # maximizing an objective function called net
      # similarity.

library(apcluster)
fit_AP<-apcluster(negDistMat(r=2),x=MUS,details=T)
cat("affinity propogation optimal number of clusters:",
    length(fit_AP@clusters), "\n")
# 8
fit_AP
fit_AP@sim
windows()
apcluster::heatmap(fit_AP)
plot(fit_AP,MUS)

## Cambiar la preferencia al cuantil 10% de
   # similaridades
fit_AP2<-apcluster(s=fit_AP@sim,q=0.1)
show(fit_AP2)
plot(fit_AP2,MUS)

## now try the same with RBF kernel
sim <- expSimMat(MUS, r=2)
apres <- apcluster(s=sim, q=0.1)
show(apres)
plot(apres,MUS)
```

A1.2. Clusters_All.R (Subsection 3.3.2)

```
rm(list=ls())
ruta<-"C:/Users/JuanJose/Google Drive/ANL/MASTERS/TECI/TFM/Universidad"
# Cambiar ruta segun proceda
setwd(ruta)


######## CARGA DATOS ########

MUS<-read.csv(file="all.csv")
# Tabla con las 57 ciudades con todas las variables
   # estandarizadas (media y desv. de la muestra)
rownames(MUS)<-MUS[,1]
ciudades<-rownames(MUS)
colnames(MUS)
MUS<-MUS[,-1]
summary(MUS)
# for (k in seq(1,71,10)){
#    if (k<71) kfin<-k+9 else kfin<-k+7
#    boxdata <- with(MUS,boxplot(MUS[,c(k:kfin)]),range=4)
#    for(i in 1:length(boxdata$group)){
#       text(boxdata$group[i],boxdata$out[i],
```

```
#             ciudades[if(
#                which(MUS[,c(k:kfin)]==
#                         boxdata$out[i])%%dim(MUS)[1]==0)
#                which(MUS[,c(k:kfin)]==boxdata$out[i])
#                else
#                   which(MUS[,c(k:kfin)]==
#                            boxdata$out[i])%%dim(MUS)[1]],
#             pos=4,cex=.75)
#    }
# }
Corr_MUS<-cor(MUS)["MUS.TRAFFIC",]
Corr_MUS[Corr_MUS>0.8]
#MUS<-MUS[,names(Corr_MUS[Corr_MUS>0.8])]

######## CLUSTERS JERARQUICOS ########

library(cluster)
distance<-dist(MUS)

# A la vista de la representacion anterior y
   # considerando los objetivos, se descarta los
   # metodos de:
      # WARD: muy sensible a Outliers y no buscamos
         # Clusters con similar nº de individuos
      # COMPLETO: tampoco interesa que los Clusters
         #tengan diametros similares
# Y en su lugar se consideraran:
      # MEDIO: Clusters con Varianzas similares
      # SIMPLE: Clusters irregulares. Busca el vecino
         # mas proximo.
      # CENTROIDE: la distancia entre Clusters es el
         # cuadrado de la distancia euclidea entre sus
         # centroides

fit_medio<-hclust(distance,method="average")
plot(fit_medio,hang=-1,main="Average Linkage Method")

# Una limitacion del Clustering Jerarquico es que
   # clasifica observaciones pero no explica en que
   # difiere cada clase/Cluster

# ¿Que numero de Clusters es el adecuado?
   # Podemos definir como metrica una usada en
      # Particiones: la Anchura de la SILUETA de los
      # Clusters.
      # Para cada observacion se compara su cercania al
      # resto de miembros de su mismo Cluster con su
      # cercania a miembros de otros Clusters.
   # Valores proximos a 1 indican que la observacion
      # esta bien asignada a su Cluster.
   # Valores proximos a 0 indican que deberia asignarse
      # a otro Cluster.
   # Usaremos en primer lugar la Anchura Media de las
      # Siluetas: si es superior a 0.7 indicara una
```

```
      # fuerte estructura; si es inferior a 0.5, que la
      # estructura es debil o inexistente.

silueta<-numeric(28)
silueta[1]<-1
for (i in 2:28){
   silueta[i]<-summary(silhouette(cutree(fit_medio,k=i),
                                  distance))$avg.width
   }
plot(silueta,type="b")
# La Anchura Media de la silueta baja de 0.7 a partir
   # de 3 Clusters.
# Pero además, a partir de 4 Clusters no hay buen
   # ajuste de algunas observaciones.
i<-4
plot(silhouette(cutree(fit_medio,k=i),distance))
# Parece razonable quedarse con 3 Clusters, incluyendo
   # el 2º y el 3º a los Outliers Madrid y Barcelona
   # (por separado)
         # addmargins(table(rownames(MUS),cutree(fit_medio,k=3)))
         # cutree(fit_medio, k=3)
plot(fit_medio,hang=-1,main="Average Linkage Method")
rect.hclust(fit_medio,k=3, border=2:4)

fit_simple<-hclust(distance,method="single")
plot(fit_simple,hang=-1,main="Single Linkage Method")
silueta<-numeric(28)
silueta[1]<-1
for (i in 2:28){
   silueta[i]<-summary(silhouette(cutree(fit_simple,k=i),
                                  distance))$avg.width
}
plot(silueta,type="b")
# DE NUEVO La Anchura Media de la silueta baja de 0.7
   # a partir de 3 Clusters.
# Pero además, a partir de 4 Clusters no hay buen
   # ajuste de algunas observaciones.
i<-3
plot(silhouette(cutree(fit_simple,k=i),distance))
# Parece razonable quedarse con 3 Clusters, incluyendo
   # el 2º y el 3º a los Outliers Madrid y Barcelona
   # (por separado)
plot(fit_simple,hang=-1,main="Single Linkage Method")
rect.hclust(fit_simple,k=3, border=2:4)

fit_centroide<-hclust(distance,method="centroid")
plot(fit_centroide,hang=-1,main="Centroid Linkage Method")
silueta<-numeric(28)
silueta[1]<-1
for (i in 2:28){
   silueta[i]<-summary(silhouette(cutree(fit_centroide,
                                         k=i),
                                  distance))$avg.width
}
```

```
plot(silueta,type="b")
# DE NUEVO La Anchura Media de la silueta baja de 0.7
   # a partir de 3 Clusters.
# Pero además, a partir de 5 Clusters no hay buen
   # ajuste de algunas observaciones.
i<-5
plot(silhouette(cutree(fit_centroide,k=i),distance))
# Parece razonable quedarse con 3 Clusters, incluyendo
   # el 2º y el 3º a los Outliers Madrid y Barcelona
   # (por separado)
plot(fit_centroide,hang=-1,
     main="Centroid Linkage Method")
rect.hclust(fit_centroide,k=3, border=2:4)


######## CLUSTERS NO JERARQUICOS ########
######## K-MEDIAS ########

# Para determinar el nº optimo de Clusters analizamos
   # la suma de cuadrados de las distancias de las
   # observaciones en un mismo Cluster, para todos los
   # Clusters: tendremos asi una medida de lo compacto
   # de cada Cluster
# Probamos hasta 28 Clusters ya que interesan Clusters
   # con al menos 1 par de observaciones
errores<-numeric(28)
for(i in 1:28){
   set.seed(2096730329)
   errores[i]<-kmeans(MUS,centers=i)$tot.withinss}
plot(errores,type="b")
# El punto de inflexion parece estar en 5 Clusters, a
   # partir de ahí la agrupación se hace más compleja
   # sin un alto impacto en la minimizacion del error
kmeans(MUS,centers=3)$cluster

# Tambien podemos comprobar las distancias entre
   # Clusters
dist_clusters<-numeric(28)
for(i in 1:28){
   set.seed(2096730329)
  dist_clusters[i]<-kmeans(MUS,centers=i)$betweenss
}
plot(dist_clusters,type="b")
# La grafica vuelve a sugerir 5 clusters

# O, mejor aun, calcular el indice CH (Calinski-
   # Harabasz), que nos permite calcular el numero k
   # de clusters que maximiza la relacion entre una
   # distancia total intra-cluster baja y una distancia
   # inter-cluster alta
CH_index<-function(datos,k){
   set.seed(2096730329)
   fit<-kmeans(datos,k)
   CH_index<-(fit$betweenss/(k-1))/
```

```
      (fit$tot.withinss/(dim(datos)[1]-k))
}
CH_k<-numeric(28)
CH_k<-sapply(1:28,function(i) CH_index(MUS,i))
which(CH_k==max(CH_k))
plot(c(1:28),CH_k,type="b",xlab="k",ylab="CH(k)",xlim=c(2,28), xaxt='n')
axis(1,at=seq(0, 28, 2))
# En puridad el maximo se da con 2 clusters, pero vamos
   # a quedarnos con el sugiente pico, en k=5 o 6
      # El indice CH funciona bien con clusters del
         # mismo tamaño, y aqui no es el caso

# Nos quedamos por tanto con 5 Clusters
set.seed(2096730329)
fit_kmedias<-kmeans(MUS,4)
summary(fit_kmedias)
str(fit_kmedias)
fit_kmedias$cluster
fit_kmedias$withins
addmargins(table(rownames(MUS),fit_kmedias$cluster))
Clusters<-sapply(1:5,function(i){
   fit_kmedias$cluster[fit_kmedias$cluster==i]})
print(Clusters)
library(fpc)
plotcluster(MUS,fit_kmedias$cluster)
clusplot(MUS,fit_kmedias$cluster,color=F,shade=T,
         labels=2,cex=0.75)

ord <- cmdscale(distance,eig=T)
plot(ord$points)

ordihull(ord,fit_kmedias$cluster,lty=10)
ordispider(ord,fit_kmedias$cluster,col="blue",
           label=TRUE)

######## PAM ########

pamk.best<-pamk(data=MUS,krange=28,criterion="asw",
                critout=T)
pam3<-pamk(data=MUS,krange=5,criterion="asw",
           critout=T)
pam(MUS,pam3$nc)
pamk.best$pamobject[[3]]
pam3$pamobject[[3]]
clusplot(pam(MUS,pam3$nc),lines=0,color=F,shade=F,
         labels=2,cex=0.75)


ord <- cmdscale(distance,eig=T)
plot(ord$points)

ordihull(ord,pam3$pamobject[[3]],lty=10)
ordispider(ord,pam3$pamobject[[3]],col="blue",
           label=TRUE)
```

```
######### The optimal model according to BIC for EM
   # initialized by hierarchical clustering for
   # parameterized Gaussian mixture models####

library(mclust)
d_clust<-Mclust(as.matrix(MUS),G=1:28)
m.best<-dim(d_clust$z)[2]
cat("model-based optimal number of clusters:",m.best,
    "\n")
windows()
plot(d_clust)
d_clust$classification

ord <- cmdscale(distance,eig=T)
plot(ord$points)
ordihull(ord,d_clust$classification,lty=10)
ordispider(ord,d_clust$classification,col="blue",
           label=TRUE)

######### AFFINITY PROPAGATION CLUSTERING ####
   # Affinity Propagation clusters data using a set of
      # real-valued pairwise data point similarities as
      # input. Each cluster is represented by a cluster
      # center data point (the so-called exemplar). The
      # method is iterative and searches for clusters
      # maximizing an objective function called net
      # similarity.

library(apcluster)
fit_AP<-apcluster(negDistMat(r=2),x=MUS,details=T)
cat("affinity propogation optimal number of clusters:",
    length(fit_AP@clusters), "\n")
# 13
fit_AP
fit_AP@sim
windows()
apcluster::heatmap(fit_AP)
plot(fit_AP,MUS)
```

A1.3. TimeSeries_Neighbors.R (Section 3.4)

```
######## INICIALIZACION ########

rm(list=ls())
ruta<-"C:/Users/JuanJose/Google Drive/ANL/MASTERS/TECI/TFM/Universidad"
# Cambiar ruta segun proceda
setwd(ruta)

######## CARGA LIBRERIAS ########

library(cluster)
```

```
library(dtw)
library(TSclust)
library(fpc)
library(ade4)


######## CARGA FUNCIONES ########

posicion<-function(distancia,coord,N){
   if (coord%%N==0){
      fila<-dim(distancia)[1];
      columna<-coord/N} else {
         fila<-coord%%N;
         columna<-floor(coord/N)+1}
   lista<-list(fila,columna)
}


empareja<-function(vecinos,N){
   parejas<-data.frame()
   j<-0
   for (i in 1:N){
   if(vecinos[rownames(vecinos)==vecinos[i,],]==
         rownames(vecinos)[i]){
      if(j==0){
         j<-j+1
         parejas[j,1]<-rownames(vecinos)[i]
         parejas[j,2]<-vecinos[rownames(vecinos)[i],]
      }else if (sum(sapply(rownames(vecinos)[i]==
                              parejas[c(1:j),2],
                           prod))!=1){
         j<-j+1
         parejas[j,1]<-rownames(vecinos)[i]
         parejas[j,2]<-vecinos[rownames(vecinos)[i],]}
   }
   }
   colnames(parejas)<-c("Ciudad","Vecina")
   parejas[,2]<-as.character(parejas[,2])
   parejas
}


num2bin<-function(number,noBits){
   binary_vector=rev(as.numeric(intToBits(number)))
   if(missing(noBits))
      binary_vector
   else
      binary_vector[-(1:(length(binary_vector)-noBits))]
}


optimiza<-function(parejas1,parejas2,num,numBits,maximo){
   distTC<-Inf
   d<-NULL
```

```
   for (i in 0:num){
      Test_loop<-as.matrix(parejas1[,1])
      Ctrl_loop<-as.matrix(parejas1[,2])
      d_aux<-Inf
      for (j in 1:numBits){
         if(num2bin(i,numBits)[j]==0){
            Test_loop<-rbind(Test_loop,parejas2[j,1]);
            Ctrl_loop<-rbind(Ctrl_loop,parejas2[j,2])
         }else{
            Test_loop<-rbind(Test_loop,parejas2[j,2]);
            Ctrl_loop<-rbind(Ctrl_loop,parejas2[j,1])}
      }
      GTest<-TSTrafico[,Test_loop]
      GCtrl<-TSTrafico[,Ctrl_loop]
      d_aux<-diss(as.data.frame(cbind(rowSums(GTest),
                                      rowSums(GCtrl))),
             "DWT",diag=T)/maximo
      distTC<-min(distTC,d_aux)
      if(d_aux==distTC){
         Test<-Test_loop;
         Ctrl<-Ctrl_loop}
      d<-cbind(d,d_aux)
   }
   lista<-list(distTC,d,Test,Ctrl)
}

representa<-function(GTestTrafico,GCtrlTrafico,
                     GTestTrafico_rnd,GCtrlTrafico_rnd,
                     GTestTrafico_rnd_TOTAL,
                     GCtrlTrafico_rnd_TOTAL,
                     GTestVisitas,GCtrlVisitas,
                     GTestVisitas_rnd,GCtrlVisitas_rnd,
                     GTestVisitas_rnd_TOTAL,
                     GCtrlVisitas_rnd_TOTAL){
   ymax<-colMaxs(rowMaxs(cbind(
      rowSums(GTestTrafico),
      rowSums(GCtrlTrafico),
      rowSums(GTestTrafico_rnd),
      rowSums(GCtrlTrafico_rnd),
      rowSums(GTestTrafico_rnd_TOTAL),
      rowSums(GCtrlTrafico_rnd_TOTAL))))
   ymax2<-colMaxs(rowMaxs(cbind(
      rowSums(GTestVisitas),
      rowSums(GCtrlVisitas),
      rowSums(GTestVisitas_rnd),
      rowSums(GCtrlVisitas_rnd),
      rowSums(GTestVisitas_rnd_TOTAL),
      rowSums(GCtrlVisitas_rnd_TOTAL))))

   texto1<-"Optimal assignment of the"
   texto2<-"Random assignment of the "
   texto_fin<-"best maching GMAs"
   texto3<-"Random assignment of 56 of the 57 GMAs"
   numero<-as.character(2*ncol(GTestTrafico))
```

```
plot(rowSums(GTestTrafico),type="l",col="blue",
     ylim=c(0,ymax),lwd=2,
     main=paste("Traffic of Test & Control groups",
                "\n",texto1,numero,texto_fin),
     xlab="Pre-test period Week",ylab="Traffic")
lines(rowSums(GCtrlTrafico),col="red",lwd=2)
legend("topleft",legend = c("Test","Control"),lty=1,
       col=c("blue","red"),bty="n",lwd=2)

plot(rowSums(GTestTrafico_rnd),type="l",col="cyan",
     ylim=c(0,ymax),lwd=2,
     main=paste("Traffic of Test & Control groups",
                "\n",texto2,numero,texto_fin),
     xlab="Pre-test period Week",ylab="Traffic")
lines(rowSums(GCtrlTrafico_rnd),col="orange",lwd=2)
legend("topleft",legend = c("Test","Control"),lty=1,
       col=c("cyan","orange"),bty="n",lwd=2)

plot(rowSums(GTestTrafico_rnd_TOTAL),type="l",
     col="green",ylim=c(0,ymax),lwd=2,
     main=paste("Traffic of Test & Control groups",
                "\n",texto3),
     xlab="Pre-test period Week",
     ylab="Traffic")
lines(rowSums(GCtrlTrafico_rnd_TOTAL),
      col="darkorange",lwd=2)
legend("topleft",legend = c("Test","Control"),lty=1,
       col=c("green","darkorange"),bty="n",lwd=2)

plot(rowSums(GTestVisitas),type="l",col="blue",
     ylim=c(0,ymax2),lwd=2,
     main=paste("Display Visits of Test & Control groups",
                "\n",texto1,numero,texto_fin),
     xlab="Pre-test period Week",
     ylab="Display Visits")
lines(rowSums(GCtrlVisitas),col="red",lwd=2)
legend("topright",legend = c("Test","Control"),lty=1,
       col=c("blue","red"),bty="n",lwd=2)
plot(rowSums(GTestVisitas_rnd),type="l",col="cyan",
     ylim=c(0,ymax2),lwd=2,
     main=paste("Display Visits of Test & Control groups",
                "\n",texto2,numero,texto_fin),
     xlab="Pre-test period Week",
     ylab="Display Visits")
lines(rowSums(GCtrlVisitas_rnd),col="orange",lwd=2)
legend("topright",legend = c("Test","Control"),lty=1,
       col=c("cyan","orange"),bty="n",lwd=2)
plot(rowSums(GTestVisitas_rnd_TOTAL),type="l",
     col="green",ylim=c(0,ymax2),lwd=2,
     main=paste("Display Visits of Test & Control groups",
                "\n",texto3),
     xlab="Pre-test period Week",
```

```
        ylab="Display Visits")
   lines(rowSums(GCtrlVisitas_rnd_TOTAL),col="darkorange",
         lwd=2)
   legend("topright",legend = c("Test","Control"),lty=1,
          col=c("green","darkorange"),bty="n")
}


######## CARGA DATOS ########

TSTrafico<-read.csv(file="TimeSeriesTrafico.csv")
NCiudades<-ncol(TSTrafico)
Nsemanas<-nrow(TSTrafico)
TSVisitas<-read.csv(file="TimeSeriesVisitasDisp.csv")

plot(c(1:Nsemanas),rowSums(TSTrafico),type="l",
     xlab="Semana",ylab="Trafico Conc. y Visitas Web DISP (normalizados)",
     main="Trafico Conc. y Visitas Web DISP en las 57 ciudades",
     col="blue",ylim=c(0,4),lwd=2)
lines(c(1:Nsemanas),rowSums(TSVisitas),col="red",lwd=2)
legend("topright",legend = c("Trafico","Visitas DISP"),lty=1,
       col=c("blue","red"),cex=.75,lwd=2)

sum(rowSums(TSTrafico))
sum(rowSums(TSVisitas))
# La suma de la fila iesima da el % de la Var.
   # en l semana i, para todas las Ciudades
# La suma de la columna iesima da el % de la Var.
   # en la Ciudad i, para todo el periodo de Estudio


######## DISTANCIAS: DWT (Wavelet) ########

distTrafico<-as.data.frame(
   as.matrix(diss(TSTrafico,"DWT",diag=T)))
distVisitas<-as.data.frame(
   as.matrix(diss(TSVisitas,"DWT",diag=T)))

var(rowSums(TSTrafico))
# 0.04
var(rowSums(TSVisitas))
# 0.75
sum_distTrafico<-sum(rowSums(distTrafico))/2
# 239.585
sum_distVisitas<-sum(rowSums(distVisitas))/2
# 506.842
max_distTrafico<-max(distTrafico[,1:NCiudades])
# 0.791
max_distVisitas<-max(distVisitas[,1:NCiudades])
# 3.580
# Para tener una mejor noción de lo que suponen las
   # distancias, normalizaremos respecto a la maxima
   # distancia entre 2 Ciudades
coord_max_distTrafico<-which(distTrafico==
```

```
                                  max(distTrafico))[1]
lista<-posicion(distTrafico,coord_max_distTrafico,
                NCiudades)
row<-lista[[1]];col<-lista[[2]]
rownames(distTrafico)[row]
colnames(distTrafico)[col]
coord_max_distVisitas<-which(distVisitas==
                                  max(distVisitas))[1]
lista<-posicion(distVisitas,coord_max_distVisitas,
                NCiudades)
row<-lista[[1]];col<-lista[[2]]
rownames(distVisitas)[row]
colnames(distVisitas)[col]
# MADRID es la Ciudad más alejada de cualquier otra
   # en ambos casos: de BEJAR en Trafico y de LEIOA en
   # Visitas
distTrafico<-distTrafico/max_distTrafico
distVisitas<-distVisitas/max_distVisitas
sum_distTrafico<-sum(rowSums(distTrafico))/2
sum_distVisitas<-sum(rowSums(distVisitas))/2
# Si dividimos por maxTrafico (o maxVisitas) las
   # Series Temporales, se obtiene tambien una
   # distancia maxima = 1 entre la pareja de Ciudades
   # citada


######## VECINOS + PROXIMOS Y PAREJAS ########

Vecinos<-sapply(1:NCiudades,function(i){
   rownames(distTrafico)[
      which(distTrafico[i,]==min(
         distTrafico[i,distTrafico[i,]!=0]))]})
Vecinos<-as.data.frame(Vecinos)
rownames(Vecinos)<-rownames(distTrafico)

k<-3
knumVecinos<-matrix(0,ncol=k,nrow=NCiudades)
kVecinos<-knumVecinos
for (i in 1:NCiudades){
   knumVecinos[i,]<-order(distTrafico[i,])[2:(k+1)]
   for (j in 1:k)
      kVecinos[i,j]<-colnames(distTrafico[1,])[
         knumVecinos[i,j]]
}
rownames(kVecinos)<-colnames(distTrafico)
kVecinos<-as.data.frame(kVecinos)

parejas<-empareja(Vecinos,NCiudades)

num<-2^(dim(parejas)[1]-1)-1
numBits<-dim(parejas)[1]-1

p1<-parejas[1,]
p2<-parejas[c(2:dim(parejas)[1]),]
```

```
lista<-optimiza(p1,p2,num,numBits,max_distTrafico)
distTCTrafico<-lista[[1]]
d<-lista[[2]]
Test<-lista[[3]]
Ctrl<-lista[[4]]
GTestTrafico<-TSTrafico[,Test]
GCtrlTrafico<-TSTrafico[,Ctrl]
GTestVisitas<-TSVisitas[,Test]
GCtrlVisitas<-TSVisitas[,Ctrl]
(cor(rowSums(GTestTrafico),rowSums(GCtrlTrafico)))
(cor(rowSums(GTestVisitas),rowSums(GCtrlVisitas)))
distTCVisitas<-diss(as.data.frame(cbind(
   rowSums(GTestVisitas),rowSums(GCtrlVisitas))),
   "DWT",diag=T)/max_distVisitas
plot(1:(num+1),d,type="l",xlab="#iteración",
     ylab="distancia",main="Distancia entre Test y Ctrl")
plot(1:(num+1),sort(d,decreasing=T),type="l",
     xlab="#iteración",ylab="distancia",
     main="Distancia entre Test y Ctrl")


######## SELECCION ALEATORIA ########

array_parejas<-c(parejas[,1],as.character(parejas[,2]))
GTotalTrafico_rnd<-data.frame(
   row.names=row.names(TSTrafico))
GTotalVisitas_rnd<-GTotalTrafico_rnd
for (i in 1:length(array_parejas)){
   GTotalTrafico_rnd[,i]<-
      TSTrafico[,array_parejas[i]]
   GTotalVisitas_rnd[,i]<-
      TSVisitas[,array_parejas[i]]
   colnames(GTotalTrafico_rnd)[i]<-
      array_parejas[i]
   colnames(GTotalVisitas_rnd)[i]<-
      array_parejas[i]
}
set.seed(123456789)
v1<-sample(length(array_parejas),
           length(array_parejas)/2,replace=FALSE)
v2<-c(1:length(array_parejas))[-v1]
GTestTrafico_rnd<-GTotalTrafico_rnd[v1]
GCtrlTrafico_rnd<-GTotalTrafico_rnd[v2]
GTestVisitas_rnd<-GTotalVisitas_rnd[v1]
GCtrlVisitas_rnd<-GTotalVisitas_rnd[v2]
distTCTrafico_rnd<-diss(as.data.frame(
   cbind(rowSums(GTestTrafico_rnd),
         rowSums(GCtrlTrafico_rnd))),"DWT",
   diag=T)/max_distTrafico
cor(rowSums(GTestTrafico_rnd),
    rowSums(GCtrlTrafico_rnd))
distTCTrafico_rnd
distTCTrafico
distTCVisitas_rnd<-diss(as.data.frame(
```

```
   cbind(rowSums(GTestVisitas_rnd),
         rowSums(GCtrlVisitas_rnd))),"DWT",
   diag=T)/max_distVisitas
cor(rowSums(GTestVisitas_rnd),
    rowSums(GCtrlVisitas_rnd))
distTCVisitas_rnd
distTCVisitas

set.seed(123456789)
v1<-sample(NCiudades,NCiudades/2,replace=F)
v2<-c(1:NCiudades)[-v1]
length(sort(c(v1,v2)))
if(NCiudades%%2!=0)
   v2<-v2[randconf(NCiudades/2+1,NCiudades/2)]
GTestTrafico_rnd_TOTAL<-TSTrafico[v1]
GCtrlTrafico_rnd_TOTAL<-TSTrafico[v2]
GTestVisitas_rnd_TOTAL<-TSVisitas[v1]
GCtrlVisitas_rnd_TOTAL<-TSVisitas[v2]
dTCTrafico_rnd_TOTAL<-diss(as.data.frame(
   cbind(rowSums(GTestTrafico_rnd_TOTAL),
         rowSums(GCtrlTrafico_rnd_TOTAL))),"DWT",
   diag=T)/max_distTrafico
cor(rowSums(GTestTrafico_rnd_TOTAL),
    rowSums(GCtrlTrafico_rnd_TOTAL))
dTCVisitas_rnd_TOTAL<-
   diss(as.data.frame(
      cbind(rowSums(GTestVisitas_rnd_TOTAL),
            rowSums(GCtrlVisitas_rnd_TOTAL))),"DWT",
      diag=T)/max_distVisitas
cor(rowSums(GTestVisitas_rnd_TOTAL),
    rowSums(GCtrlVisitas_rnd_TOTAL))


######## REPRESENTACION ########

representa(GTestTrafico,GCtrlTrafico,
           GTestTrafico_rnd,GCtrlTrafico_rnd,
           GTestTrafico_rnd_TOTAL,
           GCtrlTrafico_rnd_TOTAL,
           GTestVisitas,GCtrlVisitas,
           GTestVisitas_rnd,GCtrlVisitas_rnd,
           GTestVisitas_rnd_TOTAL,
           GCtrlVisitas_rnd_TOTAL)

mds_coord<-cmdscale(distTrafico,eig=T,x.ret=T,add=F)
mds_coord$GOF
max(mds_coord$points[,1])
min(mds_coord$points[,1])
max(mds_coord$points[,2])
min(mds_coord$points[,2])
plot(NULL,xlim=c(-.3,0.9),ylim=c(-0.1,0.1),asp=1,axes=T,lty=2)
s.label(mds_coord$points,xax=1,yax=2,
        label=row.names(distTrafico),clabel = 0.6,
```

```
        boxes=F,xlim=c(-.2,0.85),ylim=c(-0.2,0.1),
        cpoint=1,add.plot=T)
s.label(mds_coord$points,xax=1,yax=2,
        label=row.names(distTrafico),clabel = 0.6,
        boxes=F,xlim=c(-.2,0.3),ylim=c(-0.2,0.15),
        cpoint=1)


######## VECINOS Y PAREJAS AL REDUCIR LA DIM. ########

distTrafico_MDS<-as.data.frame(as.matrix(
   dist(mds_coord$points)))
# distTrafico_MDS["Madrid","Bejar"]
# Se preserva en gran medida la distancia unitaria
   # Madrid-Bejar: 0.9985133

Vecinos2<-sapply(1:NCiudades,function(i){
   rownames(distTrafico_MDS)[
      which(distTrafico_MDS[i,]==min(
         distTrafico_MDS[i,distTrafico_MDS[i,]!=0]))]})
Vecinos2<-as.data.frame(Vecinos2)
rownames(Vecinos2)<-rownames(distTrafico_MDS)

posibles_parejas<-empareja(Vecinos2,NCiudades)
parejas2<-parejas
for (i in 1:dim(posibles_parejas)[1]){
   if(
      length(which(parejas==
                      posibles_parejas[i,1]))==0 &
         length(which(parejas==
                         posibles_parejas[i,2]))==0)
         parejas2<-rbind(parejas2,posibles_parejas[i,])
}
rownames(parejas2)<-c(1:dim(parejas2)[1])
nuevas_parejas<-parejas2[-c(1:dim(parejas)[1]),]

num<-2^(dim(nuevas_parejas)[1])-1
numBits<-dim(nuevas_parejas)[1]

p1<-cbind(as.character(Test),as.character(Ctrl))
p2<-nuevas_parejas
lista<-optimiza(p1,p2,num,numBits,max_distTrafico)
distTCTrafico2<-lista[[1]]
d2<-lista[[2]]
Test2<-lista[[3]]
Ctrl2<-lista[[4]]
GTestTrafico2<-TSTrafico[,Test2]
GCtrlTrafico2<-TSTrafico[,Ctrl2]
GTestVisitas2<-TSVisitas[,Test2]
GCtrlVisitas2<-TSVisitas[,Ctrl2]
(cor(rowSums(GTestTrafico2),rowSums(GCtrlTrafico2)))
(cor(rowSums(GTestVisitas2),rowSums(GCtrlVisitas2)))
distTCVisitas2<-diss(as.data.frame(cbind(
```

```
   rowSums(GTestVisitas2),rowSums(GCtrlVisitas2))),
   "DWT",diag=T)/max_distVisitas
plot(1:(num+1),d2,type="l",xlab="#iteración",
     ylab="distancia",main="Distancia entre Test y Ctrl")
plot(1:(num+1),sort(d2,decreasing=T),type="l",
     xlab="#iteración",ylab="distancia",
     main="Distancia entre Test y Ctrl")


######## SELECCION ALEATORIA ########

array_parejas2<-c(parejas2[,1],as.character(parejas2[,2]))
GTotalTrafico_rnd2<-data.frame(
   row.names=row.names(TSTrafico))
GTotalVisitas_rnd2<-GTotalTrafico_rnd2
for (i in 1:length(array_parejas2)){
   GTotalTrafico_rnd2[,i]<-
      TSTrafico[,array_parejas2[i]]
   GTotalVisitas_rnd2[,i]<-
      TSVisitas[,array_parejas2[i]]
   colnames(GTotalTrafico_rnd2)[i]<-
      array_parejas2[i]
   colnames(GTotalVisitas_rnd2)[i]<-
      array_parejas2[i]
}
set.seed(123456789)
v1<-sample(length(array_parejas2),
           length(array_parejas2)/2,replace=FALSE)
v2<-c(1:length(array_parejas2))[-v1]
GTestTrafico_rnd2<-GTotalTrafico_rnd2[v1]
GCtrlTrafico_rnd2<-GTotalTrafico_rnd2[v2]
GTestVisitas_rnd2<-GTotalVisitas_rnd2[v1]
GCtrlVisitas_rnd2<-GTotalVisitas_rnd2[v2]
distTCTrafico_rnd2<-diss(as.data.frame(
   cbind(rowSums(GTestTrafico_rnd2),
         rowSums(GCtrlTrafico_rnd2))),"DWT",
   diag=T)/max_distTrafico
cor(rowSums(GTestTrafico_rnd2),
    rowSums(GCtrlTrafico_rnd2))
distTCTrafico_rnd2
distTCTrafico2
distTCVisitas_rnd2<-diss(as.data.frame(
   cbind(rowSums(GTestVisitas_rnd2),
         rowSums(GCtrlVisitas_rnd2))),"DWT",
   diag=T)/max_distVisitas
cor(rowSums(GTestVisitas_rnd2),
    rowSums(GCtrlVisitas_rnd2))
distTCVisitas_rnd2
distTCVisitas2


######## REPRESENTACION ########

representa(GTestTrafico2,GCtrlTrafico2,
```

```
            GTestTrafico_rnd2,GCtrlTrafico_rnd2,
            GTestTrafico_rnd_TOTAL,
            GCtrlTrafico_rnd_TOTAL,
            GTestVisitas2,GCtrlVisitas2,
            GTestVisitas_rnd2,GCtrlVisitas_rnd2,
            GTestVisitas_rnd_TOTAL,
            GCtrlVisitas_rnd_TOTAL)


######## VECINOS + PROXIMOS Y PAREJAS en las ciudades restantes ########

ciudades<-sort(colnames(TSTrafico))
ciudades2<-sort(array_parejas2)
ciudades_Restantes<-ciudades
for (i in 1:length(array_parejas2)){
   ciudades_Restantes<-
      ciudades_Restantes[which(
         ciudades_Restantes!=ciudades2[i])]
}
NCiudades_Restantes<-length(ciudades_Restantes)

distTrafico_21<-distTrafico[ciudades_Restantes,
                            ciudades_Restantes]
sum_distTrafico_21<-rowSums(distTrafico_21)
names(sum_distTrafico_21[sum_distTrafico_21==
                            max(sum_distTrafico_21)])
# Gijon es la ciudad mas alejada del resto

Vecinos3<-sapply(1:NCiudades_Restantes,function(i){
   rownames(distTrafico_21)[
      which(distTrafico_21[i,]==min(
         distTrafico_21[i,distTrafico_21[i,]!=0]))]})
Vecinos3<-as.data.frame(Vecinos3)
rownames(Vecinos3)<-rownames(distTrafico_21)

nuevas_parejas2<-empareja(Vecinos3,NCiudades_Restantes)
num<-2^(dim(nuevas_parejas)[1])-1
numBits<-dim(nuevas_parejas)[1]

p1<-cbind(as.character(Test2),as.character(Ctrl2))
p2<-nuevas_parejas2
lista<-optimiza(p1,p2,num,numBits,max_distTrafico)
distTCTrafico3<-lista[[1]]
d3<-lista[[2]]
Test3<-lista[[3]]
Ctrl3<-lista[[4]]
GTestTrafico3<-TSTrafico[,Test3]
GCtrlTrafico3<-TSTrafico[,Ctrl3]
GTestVisitas3<-TSVisitas[,Test3]
GCtrlVisitas3<-TSVisitas[,Ctrl3]
(cor(rowSums(GTestTrafico3),rowSums(GCtrlTrafico3)))
(cor(rowSums(GTestVisitas3),rowSums(GCtrlVisitas3)))
distTCVisitas3<-diss(as.data.frame(cbind(
   rowSums(GTestVisitas3),rowSums(GCtrlVisitas3))),
```

```
   "DWT",diag=T)/max_distVisitas
plot(1:(num+1),d3,type="l",xlab="#iteración",
     ylab="distancia",main="Distancia entre Test y Ctrl")
plot(1:(num+1),sort(d3,decreasing=T),type="l",
     xlab="#iteración",ylab="distancia",
     main="Distancia entre Test y Ctrl")


######## SELECCION ALEATORIA ########

parejas3<-rbind(parejas2,nuevas_parejas2)
array_parejas3<-c(parejas3[,1],as.character(parejas3[,2]))
GTotalTrafico_rnd3<-data.frame(
   row.names=row.names(TSTrafico))
GTotalVisitas_rnd3<-GTotalTrafico_rnd3
for (i in 1:length(array_parejas3)){
   GTotalTrafico_rnd3[,i]<-
      TSTrafico[,array_parejas3[i]]
   GTotalVisitas_rnd3[,i]<-
      TSVisitas[,array_parejas3[i]]
   colnames(GTotalTrafico_rnd3)[i]<-
      array_parejas3[i]
   colnames(GTotalVisitas_rnd3)[i]<-
      array_parejas3[i]
}
set.seed(123456789)
v1<-sample(length(array_parejas3),
           length(array_parejas3)/2,replace=FALSE)
v2<-c(1:length(array_parejas3))[-v1]
GTestTrafico_rnd3<-GTotalTrafico_rnd3[v1]
GCtrlTrafico_rnd3<-GTotalTrafico_rnd3[v2]
GTestVisitas_rnd3<-GTotalVisitas_rnd3[v1]
GCtrlVisitas_rnd3<-GTotalVisitas_rnd3[v2]
distTCTrafico_rnd3<-diss(as.data.frame(
   cbind(rowSums(GTestTrafico_rnd3),
         rowSums(GCtrlTrafico_rnd3))),"DWT",
   diag=T)/max_distTrafico
cor(rowSums(GTestTrafico_rnd3),
    rowSums(GCtrlTrafico_rnd3))
distTCTrafico_rnd3
distTCTrafico3
distTCVisitas_rnd3<-diss(as.data.frame(
   cbind(rowSums(GTestVisitas_rnd3),
         rowSums(GCtrlVisitas_rnd3))),"DWT",
   diag=T)/max_distVisitas
cor(rowSums(GTestVisitas_rnd3),
    rowSums(GCtrlVisitas_rnd3))
distTCVisitas_rnd3
distTCVisitas3


######## REPRESENTACION ########

representa(GTestTrafico3,GCtrlTrafico3,
```

```
            GTestTrafico_rnd3,GCtrlTrafico_rnd3,
            GTestTrafico_rnd_TOTAL,
            GCtrlTrafico_rnd_TOTAL,
            GTestVisitas3,GCtrlVisitas3,
            GTestVisitas_rnd3,GCtrlVisitas_rnd3,
            GTestVisitas_rnd_TOTAL,
            GCtrlVisitas_rnd_TOTAL)

mds_coord2<-cmdscale(distTrafico_21,eig=T,x.ret=T,add=F)
mds_coord2$GOF
max(mds_coord2$points[,1])
min(mds_coord2$points[,1])
max(mds_coord2$points[,2])
min(mds_coord2$points[,2])
plot(NULL,xlim=c(-.3,0.9),ylim=c(-0.2,0.1),asp=1,axes=T,lty=2)
s.label(mds_coord2$points,xax=1,yax=2,
        label=row.names(distTrafico_21),clabel = 0.6,
        boxes=F,xlim=c(-.2,0.85),ylim=c(-0.2,0.1),cpoint=1,add.plot=T)
s.label(mds_coord2$points,xax=1,yax=2,
        label=row.names(distTrafico_21),clabel = 0.6,
        boxes=F,xlim=c(-.2,0.3),ylim=c(-0.2,0.15),cpoint=1)


######## VECINOS Y PAREJAS AL REDUCIR LA DIM. ########

distTrafico_MDS_21<-as.data.frame(as.matrix(
   dist(mds_coord2$points)))

Vecinos4<-sapply(1:NCiudades_Restantes,function(i){
   rownames(distTrafico_MDS_21)[
      which(distTrafico_MDS_21[i,]==min(
         distTrafico_MDS_21[i,distTrafico_MDS_21[i,]!=0]))]})
Vecinos4<-as.data.frame(Vecinos4)
rownames(Vecinos4)<-rownames(distTrafico_MDS_21)

posibles_parejas2<-empareja(Vecinos4,NCiudades_Restantes)
parejas4<-parejas3
for (i in 1:dim(posibles_parejas2)[1]){
   if(
      length(which(parejas3==
                      posibles_parejas2[i,1]))==0 &
         length(which(parejas3==
                         posibles_parejas2[i,2]))==0)
      parejas4<-rbind(parejas4,posibles_parejas2[i,])
}
rownames(parejas4)<-c(1:dim(parejas4)[1])
nuevas_parejas3<-parejas4[-c(1:dim(parejas3)[1]),]
nuevas_parejas3
rm(nuevas_parejas3)
rm(Vecinos4)
rm(parejas4)
# NO SE AÑADEN NUEVAS PAREJAS AL REDUCIR LA DIMENSION
```

```
######## VECINOS + PROXIMOS Y PAREJAS en las ciudades restantes ########

ciudades3<-sort(array_parejas3)
ciudades_Restantes2<-ciudades
for (i in 1:length(array_parejas3)){
   ciudades_Restantes2<-
      ciudades_Restantes2[which(
         ciudades_Restantes2!=ciudades3[i])]
}
NCiudades_Restantes2<-length(ciudades_Restantes2)

distTrafico_13<-distTrafico[ciudades_Restantes2,
                            ciudades_Restantes2]

Vecinos4<-sapply(1:NCiudades_Restantes2,function(i){
   rownames(distTrafico_13)[
      which(distTrafico_13[i,]==min(
         distTrafico_13[i,distTrafico_13[i,]!=0]))]})
Vecinos4<-as.data.frame(Vecinos4)
rownames(Vecinos4)<-rownames(distTrafico_13)

nuevas_parejas3<-empareja(Vecinos4,NCiudades_Restantes2)
num<-2^(dim(nuevas_parejas3)[1])-1
numBits<-dim(nuevas_parejas3)[1]

p1<-cbind(as.character(Test3),as.character(Ctrl3))
p2<-nuevas_parejas3
lista<-optimiza(p1,p2,num,numBits,max_distTrafico)
distTCTrafico4<-lista[[1]]
d4<-lista[[2]]
Test4<-lista[[3]]
Ctrl4<-lista[[4]]
GTestTrafico4<-TSTrafico[,Test4]
GCtrlTrafico4<-TSTrafico[,Ctrl4]
GTestVisitas4<-TSVisitas[,Test4]
GCtrlVisitas4<-TSVisitas[,Ctrl4]
(cor(rowSums(GTestTrafico4),rowSums(GCtrlTrafico4)))
(cor(rowSums(GTestVisitas4),rowSums(GCtrlVisitas4)))
distTCVisitas4<-diss(as.data.frame(cbind(
   rowSums(GTestVisitas4),rowSums(GCtrlVisitas4))),
   "DWT",diag=T)/max_distVisitas
plot(1:(num+1),d4,type="l",xlab="#iteración",
     ylab="distancia",main="Distancia entre Test y Ctrl")
plot(1:(num+1),sort(d4,decreasing=T),type="l",
     xlab="#iteración",ylab="distancia",
     main="Distancia entre Test y Ctrl")


######## SELECCION ALEATORIA ########

parejas4<-rbind(parejas3,nuevas_parejas3)
array_parejas4<-c(parejas4[,1],as.character(parejas4[,2]))
GTotalTrafico_rnd4<-data.frame(
```

```
   row.names=row.names(TSTrafico))
GTotalVisitas_rnd4<-GTotalTrafico_rnd4
for (i in 1:length(array_parejas4)){
   GTotalTrafico_rnd4[,i]<-
      TSTrafico[,array_parejas4[i]]
   GTotalVisitas_rnd4[,i]<-
      TSVisitas[,array_parejas4[i]]
   colnames(GTotalTrafico_rnd4)[i]<-
      array_parejas4[i]
   colnames(GTotalVisitas_rnd4)[i]<-
      array_parejas4[i]
}
set.seed(123456789)
v1<-sample(length(array_parejas4),
           length(array_parejas4)/2,replace=FALSE)
v2<-c(1:length(array_parejas4))[-v1]
GTestTrafico_rnd4<-GTotalTrafico_rnd4[v1]
GCtrlTrafico_rnd4<-GTotalTrafico_rnd4[v2]
GTestVisitas_rnd4<-GTotalVisitas_rnd4[v1]
GCtrlVisitas_rnd4<-GTotalVisitas_rnd4[v2]
distTCTrafico_rnd4<-diss(as.data.frame(
   cbind(rowSums(GTestTrafico_rnd4),
         rowSums(GCtrlTrafico_rnd4))),"DWT",
   diag=T)/max_distTrafico
cor(rowSums(GTestTrafico_rnd4),
    rowSums(GCtrlTrafico_rnd4))
distTCTrafico_rnd4
distTCTrafico4
distTCVisitas_rnd4<-diss(as.data.frame(
   cbind(rowSums(GTestVisitas_rnd4),
         rowSums(GCtrlVisitas_rnd4))),"DWT",
   diag=T)/max_distVisitas
cor(rowSums(GTestVisitas_rnd4),
    rowSums(GCtrlVisitas_rnd4))
distTCVisitas_rnd4
distTCVisitas4


######## REPRESENTACION ########

representa(GTestTrafico4,GCtrlTrafico4,
           GTestTrafico_rnd4,GCtrlTrafico_rnd4,
           GTestTrafico_rnd_TOTAL,
           GCtrlTrafico_rnd_TOTAL,
           GTestVisitas4,GCtrlVisitas4,
           GTestVisitas_rnd4,GCtrlVisitas_rnd4,
           GTestVisitas_rnd_TOTAL,
           GCtrlVisitas_rnd_TOTAL)

mds_coord3<-cmdscale(distTrafico_13,eig=T,x.ret=T,add=F)
mds_coord3$GOF
max(mds_coord3$points[,1])
min(mds_coord3$points[,1])
max(mds_coord3$points[,2])
```

```
min(mds_coord3$points[,2])
plot(NULL,xlim=c(-.3,0.9),ylim=c(-0.2,0.1),asp=1,axes=T,lty=2)
s.label(mds_coord3$points,xax=1,yax=2,
        label=row.names(distTrafico_13),clabel = 0.6,
        boxes=F,xlim=c(-.2,0.85),ylim=c(-0.2,0.1),cpoint=1,add.plot=T)
s.label(mds_coord3$points,xax=1,yax=2,
        label=row.names(distTrafico_13),clabel = 0.6,
        boxes=F,xlim=c(-.2,0.3),ylim=c(-0.2,0.15),cpoint=1)


######## VECINOS Y PAREJAS AL REDUCIR LA DIM. ########

distTrafico_MDS_13<-as.data.frame(as.matrix(
   dist(mds_coord3$points)))

Vecinos5<-sapply(1:NCiudades_Restantes2,function(i){
   rownames(distTrafico_MDS_13)[
      which(distTrafico_MDS_13[i,]==min(
         distTrafico_MDS_13[i,distTrafico_MDS_13[i,]!=0]))]})
Vecinos5<-as.data.frame(Vecinos5)
rownames(Vecinos5)<-rownames(distTrafico_MDS_13)

posibles_parejas3<-empareja(Vecinos5,NCiudades_Restantes2)
parejas5<-parejas4
for (i in 1:dim(posibles_parejas3)[1]){
   if(
      length(which(parejas4==
                      posibles_parejas3[i,1]))==0 &
         length(which(parejas4==
                         posibles_parejas3[i,2]))==0)
      parejas5<-rbind(parejas5,posibles_parejas3[i,])
}
rownames(parejas5)<-c(1:dim(parejas5)[1])
nuevas_parejas4<-parejas5[-c(1:dim(parejas4)[1]),]
nuevas_parejas4

num<-2^(dim(nuevas_parejas4)[1])-1
numBits<-dim(nuevas_parejas4)[1]

p1<-cbind(as.character(Test4),as.character(Ctrl4))
p2<-nuevas_parejas4
lista<-optimiza(p1,p2,num,numBits,max_distTrafico)
distTCTrafico5<-lista[[1]]
d5<-lista[[2]]
Test5<-lista[[3]]
Ctrl5<-lista[[4]]
GTestTrafico5<-TSTrafico[,Test5]
GCtrlTrafico5<-TSTrafico[,Ctrl5]
GTestVisitas5<-TSVisitas[,Test5]
GCtrlVisitas5<-TSVisitas[,Ctrl5]
(cor(rowSums(GTestTrafico5),rowSums(GCtrlTrafico5)))
(cor(rowSums(GTestVisitas5),rowSums(GCtrlVisitas5)))
distTCVisitas5<-diss(as.data.frame(cbind(
```

```
   rowSums(GTestVisitas5),rowSums(GCtrlVisitas5))),
   "DWT",diag=T)/max_distVisitas
plot(1:(num+1),d5,type="l",xlab="#iteración",
     ylab="distancia",main="Distancia entre Test y Ctrl")
plot(1:(num+1),sort(d5,decreasing=T),type="l",
     xlab="#iteración",ylab="distancia",
     main="Distancia entre Test y Ctrl")


######## SELECCION ALEATORIA ########

array_parejas5<-c(parejas5[,1],as.character(parejas5[,2]))
GTotalTrafico_rnd5<-data.frame(
   row.names=row.names(TSTrafico))
GTotalVisitas_rnd5<-GTotalTrafico_rnd5
for (i in 1:length(array_parejas5)){
   GTotalTrafico_rnd5[,i]<-
      TSTrafico[,array_parejas5[i]]
   GTotalVisitas_rnd5[,i]<-
      TSVisitas[,array_parejas5[i]]
   colnames(GTotalTrafico_rnd5)[i]<-
      array_parejas5[i]
   colnames(GTotalVisitas_rnd5)[i]<-
      array_parejas5[i]
}
set.seed(123456789)
v1<-sample(length(array_parejas5),
           length(array_parejas5)/2,replace=FALSE)
v2<-c(1:length(array_parejas5))[-v1]
GTestTrafico_rnd5<-GTotalTrafico_rnd5[v1]
GCtrlTrafico_rnd5<-GTotalTrafico_rnd5[v2]
GTestVisitas_rnd5<-GTotalVisitas_rnd5[v1]
GCtrlVisitas_rnd5<-GTotalVisitas_rnd5[v2]
distTCTrafico_rnd5<-diss(as.data.frame(
   cbind(rowSums(GTestTrafico_rnd5),
         rowSums(GCtrlTrafico_rnd5))),"DWT",
   diag=T)/max_distTrafico
cor(rowSums(GTestTrafico_rnd5),
    rowSums(GCtrlTrafico_rnd5))
distTCTrafico_rnd5
distTCTrafico5
distTCVisitas_rnd5<-diss(as.data.frame(
   cbind(rowSums(GTestVisitas_rnd5),
         rowSums(GCtrlVisitas_rnd5))),"DWT",
   diag=T)/max_distVisitas
cor(rowSums(GTestVisitas_rnd5),
    rowSums(GCtrlVisitas_rnd5))
distTCVisitas_rnd5
distTCVisitas5


######## REPRESENTACION ########

representa(GTestTrafico5,GCtrlTrafico5,
```

```
            GTestTrafico_rnd5,GCtrlTrafico_rnd5,
            GTestTrafico_rnd_TOTAL,
            GCtrlTrafico_rnd_TOTAL,
            GTestVisitas5,GCtrlVisitas5,
            GTestVisitas_rnd5,GCtrlVisitas_rnd5,
            GTestVisitas_rnd_TOTAL,
            GCtrlVisitas_rnd_TOTAL)


######## VECINOS + PROXIMOS Y PAREJAS en las ciudades restantes ########

ciudades4<-sort(array_parejas5)
ciudades_Restantes3<-ciudades
for (i in 1:length(array_parejas5)){
   ciudades_Restantes3<-
      ciudades_Restantes3[which(
         ciudades_Restantes3!=ciudades4[i])]
}
NCiudades_Restantes3<-length(ciudades_Restantes3)

distTrafico_3<-distTrafico[ciudades_Restantes3,
                           ciudades_Restantes3]

Vecinos6<-sapply(1:NCiudades_Restantes3,function(i){
   rownames(distTrafico_3)[
      which(distTrafico_3[i,]==min(
         distTrafico_3[i,distTrafico_3[i,]!=0]))]})
Vecinos6<-as.data.frame(Vecinos6)
rownames(Vecinos6)<-rownames(distTrafico_3)

nuevas_parejas5<-empareja(Vecinos6,NCiudades_Restantes3)
num<-2^(dim(nuevas_parejas5)[1])-1
numBits<-dim(nuevas_parejas5)[1]

p1<-cbind(as.character(Test5),as.character(Ctrl5))
p2<-nuevas_parejas5
lista<-optimiza(p1,p2,num,numBits,max_distTrafico)
distTCTrafico6<-lista[[1]]
d6<-lista[[2]]
Test6<-lista[[3]]
Ctrl6<-lista[[4]]
GTestTrafico6<-TSTrafico[,Test6]
GCtrlTrafico6<-TSTrafico[,Ctrl6]
GTestVisitas6<-TSVisitas[,Test6]
GCtrlVisitas6<-TSVisitas[,Ctrl6]
(cor(rowSums(GTestTrafico6),rowSums(GCtrlTrafico6)))
(cor(rowSums(GTestVisitas6),rowSums(GCtrlVisitas6)))
distTCVisitas6<-diss(as.data.frame(cbind(
   rowSums(GTestVisitas6),rowSums(GCtrlVisitas6))),
   "DWT",diag=T)/max_distVisitas
plot(1:(num+1),d6,type="l",xlab="#iteración",
     ylab="distancia",main="Distancia entre Test y Ctrl")
plot(1:(num+1),sort(d6,decreasing=T),type="l",
     xlab="#iteración",ylab="distancia",
```

```
     main="Distancia entre Test y Ctrl")


######## SELECCION ALEATORIA ########

parejas6<-rbind(parejas5,nuevas_parejas5)
array_parejas6<-c(parejas6[,1],as.character(parejas6[,2]))
GTotalTrafico_rnd6<-data.frame(
   row.names=row.names(TSTrafico))
GTotalVisitas_rnd6<-GTotalTrafico_rnd6
for (i in 1:length(array_parejas6)){
   GTotalTrafico_rnd6[,i]<-
      TSTrafico[,array_parejas6[i]]
   GTotalVisitas_rnd6[,i]<-
      TSVisitas[,array_parejas6[i]]
   colnames(GTotalTrafico_rnd6)[i]<-
      array_parejas6[i]
   colnames(GTotalVisitas_rnd6)[i]<-
      array_parejas6[i]
}
set.seed(123456789)
v1<-sample(length(array_parejas6),
           length(array_parejas6)/2,replace=FALSE)
v2<-c(1:length(array_parejas6))[-v1]
GTestTrafico_rnd6<-GTotalTrafico_rnd6[v1]
GCtrlTrafico_rnd6<-GTotalTrafico_rnd6[v2]
GTestVisitas_rnd6<-GTotalVisitas_rnd6[v1]
GCtrlVisitas_rnd6<-GTotalVisitas_rnd6[v2]
distTCTrafico_rnd6<-diss(as.data.frame(
   cbind(rowSums(GTestTrafico_rnd6),
         rowSums(GCtrlTrafico_rnd6))),"DWT",
   diag=T)/max_distTrafico
cor(rowSums(GTestTrafico_rnd6),
    rowSums(GCtrlTrafico_rnd6))
distTCTrafico_rnd6
distTCTrafico6
distTCVisitas_rnd6<-diss(as.data.frame(
   cbind(rowSums(GTestVisitas_rnd6),
         rowSums(GCtrlVisitas_rnd6))),"DWT",
   diag=T)/max_distVisitas
cor(rowSums(GTestVisitas_rnd6),
    rowSums(GCtrlVisitas_rnd6))
distTCVisitas_rnd6
distTCVisitas6


######## REPRESENTACION ########

representa(GTestTrafico6,GCtrlTrafico6,
           GTestTrafico_rnd6,GCtrlTrafico_rnd6,
           GTestTrafico_rnd_TOTAL,
           GCtrlTrafico_rnd_TOTAL,
           GTestVisitas6,GCtrlVisitas6,
           GTestVisitas_rnd6,GCtrlVisitas_rnd6,
```

```
           GTestVisitas_rnd_TOTAL,
           GCtrlVisitas_rnd_TOTAL)

mds_coord4<-cmdscale(distTrafico_3,eig=T,x.ret=T,add=F)
mds_coord4$GOF
max(mds_coord4$points[,1])
min(mds_coord4$points[,1])
max(mds_coord4$points[,2])
min(mds_coord4$points[,2])
plot(NULL,xlim=c(-.3,0.9),ylim=c(-0.2,0.1),asp=1,axes=T,lty=2)
s.label(mds_coord4$points,xax=1,yax=2,
        label=row.names(distTrafico_3),clabel = 0.6,
        boxes=F,xlim=c(-.2,0.85),ylim=c(-0.2,0.1),cpoint=1,add.plot=T)
s.label(mds_coord4$points,xax=1,yax=2,
        label=row.names(distTrafico_3),clabel = 0.6,
        boxes=F,xlim=c(-.2,0.3),ylim=c(-0.2,0.15),cpoint=1)
```

A2. Some complementary results (Section 3.2)

As mentioned, some results on an Autonomous Community level were also obtained. A few of them are displayed and briefly commented below.

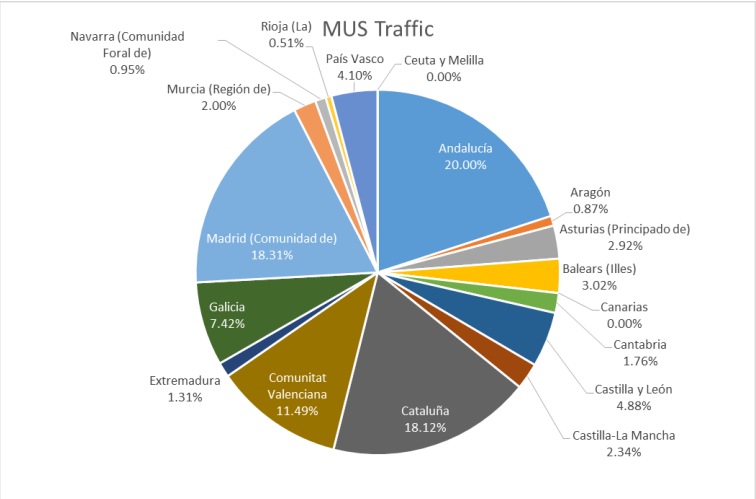


Figure 42. MUS traffic per Autonomous Community.

As expected, the MUS traffic is proportional to the Autonomous Community population

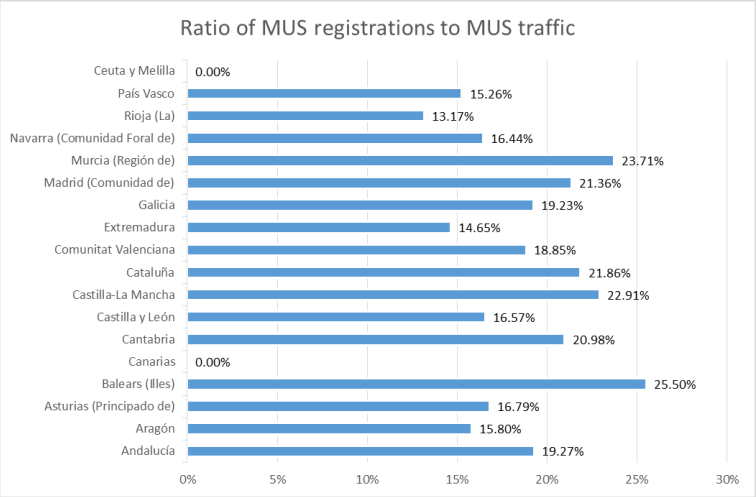


Figure 43. Ratio of MUStraffic, per Autonomous Community.

When interpreting the figure above, it must be considered that not all the dealers collect data—this is the reason why the MUS registrations in some regions are zero. But that figure give us an idea of the sales closing ratio in each Autonomous Community.

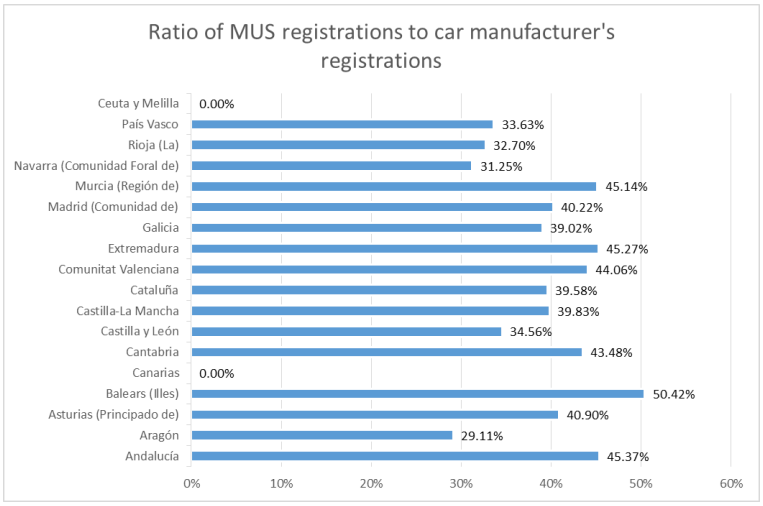


Figure 44. Ratio of MUSregistrations to car manufacturer's registrations, per Autonomous Community.

As seen in the figure above, the MUS is the best-selling model of the manufacturer under study—41.1% of total, at least in the dealers which collect data—, especially in some Autonomous Communities where it represents almost half of it sales, or even more.

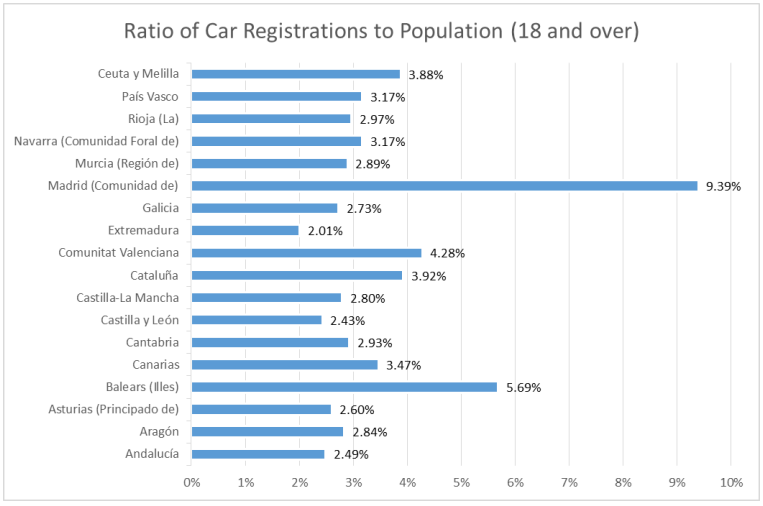


Figure 45. Ratio of car registrations to population (18 and over), per Autonomous Community.

Finally, the ratio of car registrations (considering all brands) to driving-age population is quite similar in all the Autonomous Communities, with some notorious exceptions, like Madrid.