TONGJI UNIVERSITY

UNIVERSITAT POLITÈCNICA DE CATALUNYA

Analysis, visualization and study of bike sharing dataset using Matlab.

Author:

David Hita Amorós

(UPC)

Directors:

Claudio Feijoo (UPM-Tongji)

José Luis Melús (UPC)

Table of Contents

Abstract

The purpose of this study is to demonstrate the ease of use of Matlab (with new releases) regarding the field of Data Analysis, especially for users who have little to no knowledge of complex programming languages in a field that is completely dominated by R.

The analysis focuses on analyzing a given dataset and studying the possible implementations of Machine Learning to extract useful data and test the Matlab algorithms and flexibility.

The given data comes from 'Bicing', the bike sharing system used in the city of Barcelona. It shows the dates and stations of removal and arrival, as well as the customer ID and the Bike ID for a period of time of only one week.

The study of the set will begin with the analysis of the raw data, the cleaning of it, and the consequent possible paths to be followed depending on these results.

1 Introduction

R has been always dominating the data analysis scene for a good reason, it's free, but it also has some defects. These defects make its use more difficult to people who don't have a strong background in sciences or engineering in general.

Matlab programming language is much easier to use and understand. It has a simple syntax based in matrix, which most of the problems can be put in this form. It also has an easier learning curve than R, which is messier.

It is also a mature system which has more than 20 years in use, a good community and a professional documentation and support. Toolboxes and functions are completely integrated in Matlab, which are tested regularly and are more stable than the R open source. Also Matlab has very strong parallel computing support and faster execution times.[3]

Study is oriented specially for students who are starting in the field of data analysis, who don't have a strong enough background for R but still want to use a multi-purpose software with powerful tools which can also have applications in many other uses in a degree.

1.1 Motivation

If you are looking for a career where your services will be in high demand, you should find something where you provide a scarce, complementary service to something that is getting ubiquitous and cheap. So what's getting ubiquitous and cheap? Data. And what is complementary to data? Analysis.

- Prof. Hal Varian, UC Berkeley, Chief Economist at Google [2]

As technology advances, data gathering is becoming much easier and cheaper, which leads into the analysis of it, with the objective of getting value out of things that previously was thought they didn't. Also with analysis it is possible to uncover hidden patterns that can explain relationships between variables that previously seemed uncorrelated, which in the business world it can lead to several advantages, like:

Competitive advantage - Data is a cheap resource that grows at an exponential rate, it is only natural that having data of more quality brings more value and advantage over competition.

Decision making - When you can analyze 23 variables instead of 4, it gets easier to predict outcomes, business trends…

Value of data - As volume of data continues to grow; new solutions are required to keep up with the constant increase of demand, storage and processing.

All these things can apply to the bike sharing dataset. It's important to extract conclusions of these kinds of data, as we can solve multiple problems at once.

Peak hours and public transport; many people need a transport to go to work. With increasing prices of public transport and the incommodity of congested roads during peak hours, sometimes without having to go through a long distance, a better alternative is needed.
In a growing city with increasing pollution, a bike sharing system also provides a transport system with a positive impact to the environment.

It also promotes a healthier lifestyle, which makes people more enthusiastic about it, especially with the increasing emphasis of the media in fitness and general dislike for obesity.

More importantly, it relieves from the responsibility of having to own a personal vehicle, and the problems that imply. When using bike sharing systems, people don't need to worry about having to spend extra money on maintenance and the fear of getting your vehicle stolen while you are absent, or the pain that is to find a parking place.

With analysis, it's possible to improve the planification needed to make the system successful and answer some important questions, like which zones requires more bikes, or which zones require new stations. All of these to give the customer a better experience of the system and continue growing competitively.

1.2 Research Structure

In the case of data analysis is important to define a methodology to follow in order to achieve conclusions. Since there are several ways to resolve a problem, it is often needed to analyze the previous results before continuing onwards. Many kinds of analysis tecniques exist, but not all of them are optimal, so the researcher usually needs to focus on the objective and solutions needed. [1]

The keypoints and guidelines that are followed in the research are as follow:

- Define Research problem
    - Objectives
- Develop analysis plan
    - Examining the data
        - Type of variable
        - Size
        - Dependency
        - Missing Values
        - Outliers

- Evaluate assumptions
    - Visualization of the data
        - Shape of distribution
        - Univariate profiling
        - Multivariate profiling
        - Linearity
        - Group differences
        - Requisites for analysis techniques
    -Obtaining data statistics
- Estimate the model
    - Selection of techniques
- Conclusions
- Validate the model

1.3 SOLUTION OVERVIEW

The original dataset was given in excel .xlsx format. Then extracted into Comma Separated Value format (.csv) and imported to Matlab 2014b version, where all the analysis is done. Since the raw data was not optimal for use, it suffered multiple transformations which resulted in a new, cleaner version of the same dataset, where variables without value were removed and other new ones were added. These transformations also include the elimination of certain statistical outliers and errors that skewed the model.

1.4 OBJECTIVE

There are two main objectives in the analysis. The first one is to discover if the latest releases of Matlab are viable in the data analysis field, especially when in hands of someone without much experience, to see if it can aid in simple analysis purposes without too many complications.

The second objective is to extract conclusions of the bike sharing dataset, given the set of variables, be able to get the most value out of the data and get useful information to get a better use of the system.

2 Examining the data

Examining the data is a very important step. It is important because it lays the foundations of the analysis. Knowing the behavior of variables can help decide which analysis techniques to follow and necessary requirements for their usage.

2.1 Data Set

The dataset consists in 11 variables recorded in a one week usage log of the bike sharing system 'Bicing' in the city of Barcelona. The biggest problem is that with only having one week worth of data, it's not possible to build a solid long term model but given the quantity of records, is still possible to get achieve some conclusions.

The original data is found in a Excel file(.xlsx) with 11 variables and a total of 342340 entries. The variables are constructed in the following way:

1. *BikeID* : An integer containing the ID of the Bike.
2. *CustID:* An integer containing the ID of the Customer.
3. *BikeSerialNumber*: Serial number of the bike consisting of several letters and numbers.
4. *Removal_date*: The date of removal of the bike in yyyy-mm-dd hh:mm:ss.ff format.
5. *Arrival_date:* The date of arrival of the bike in yyyy-mm-dd hh:mm:ss.ff format.
6. *Removal_station*: A number between 1 and 496, making reference to the physical station where the removal of the bike took place.
7. *Arrival_station:* A number between 1 and 496, making reference to the physical station where the arrival of the bike took place.
8. *Removal_type*: Apparently useless variable, it tells the type of removal, but it has a value of 'R' in all fields.
9. *Arrival_type:* Apparently useless variable, it tells the type of arrival, but it has a value of 'A' in all fields.
10. *Removal_slot:* A number associated to the station that specifies which slot the bike was removed.
11. *Arrival_slot*: A number associated to the station that specifies which slot the bike arrived.

2.2 Preliminary analysis

From all the given variables of the dataset its possible to remove some, due to the lack of value they contribute. These 'trash' variables, are:

Arrival & Removal type: They are always constant, so they don't give any information at all.

BikeSeriaNumber: Since it's tied with the *BikeID*, it's redundant information, thus doesn't bring any new value.

So the remaining variables are:
BikeID, CustID, Removal_date, Arrival_date, Removal_station, Arrival_station.

Not much information, considering that the stations only give a number. After some investigation, I got to the conclusion that I needed to create new variables that added value to the ones I already had. The first idea was to access the Bicing website, find where these stations were located and create a new file that contained the coordinates of these stations. With this, it's possible to build a map of the city which can help a lot in the visualization process of the analysis.

Bicing had a list of the streets where the stations where located, what was left was to look at them in Google maps and assign each one a pair of coordinates. Then, after each station got a pair of coordinates assigned, it was possible to determine the distance between two stations. Although the distance is the Euclidean distance between two points and doesn't contemplate real life obstacles and buildings in real life, it still provides a good approximation between the relationship of stations of removal and arrival and its dates.

2.3 Preparation of the data

Using Matlab, preparation of the data is probably one of the most important steps in the process of a multivariate analysis. Matlab always need inputs of data in form of a matrix to be able to use its algorithms and statistical packages.

Also, some kinds of analysis like factor analysis require not having missing values in the data matrix, so it is important to apply some kind of procedure to it. Depending to the extent of missing data in the matrix, different approaches must be taken. If the missing data is high enough (more than 10%) it becomes necessary to determine if our missing data is of random nature or it follows a pattern. If it's completely random, we could delete cases or variables without much impact (if only we have enough sample size), or apply interpolation methods to generate the missing values based on info we already have on the set based on individual means, but if the missing values follow a pattern, we could lose information related to a subset of the population, so a rethinking of the model it's needed.

Another thing to take into account are unique observations, outliers, that are different from all others, typically because they have too high or too low values to be able to fit in the model. If the dataset has a very low sample count, outliers could be of great impact to the results, in a positive or negative way, depending on their interpretation. There is several kinds of outliers, like the ones that come from errors on the entry of data, which should be treated as missing values, but we also have outliers that are the result of extraordinary events or observations, which sometimes don't have any explanation or are caused by rare causes.[1]

In this case, the variable that presented the most problems were the dates. Dates are the only kind of data that was of a complex nature, since all of the other ones were only composed by a sole string of numbers. My approach to the problem was to separate the dates variables into multiple smaller variables from the original one which was in the yyyy-mm-dd hh:mm:ss.ff format.

Since the whole dataset only contained one week worth of data, year and month were not necessary (data wasn't captured in the transition of two months). Also, capturing seconds and milliseconds felt unnecessary as the study doesn't require that much precision. In the end, two variables were created for each kind of date, day and time in dd and HHMM format respectively. So removal_date got changed into removal_day and removal_hour, analogously in arrival_dates.

The reason for putting the time into HHMM format and not HH:MM, is that with dots, it's not interpreted by a lot of Matlab functions, like Histogram and not compatible with the matrix format.

In the beginning, I tried using regular expressions and inline functions with success:

```
regexp(arrival_date,' ','split'); % Splits into yyyy-mm-dd and hh:mm:ss.ff in a 342340x1
cell containing 1x2 cells
fecha_arr = cellfun(@(x) x{1}, separa, 'uni', 0); %Split the 1x2 cells into two separate cells
hora_arr = cellfun(@(x) x{2}, separa, 'uni', 0);
separafecha = regexp(fecha_arr,'-','split'); %Does the same Split again, but now with
date, yyyy-mm-dd into yyyy mm and dd
dia_arr = cellfun(@(x) x{3}, separafecha, 'uni', 0); %Takes onle the dd of the date
separahora = regexp(hora_rem,':','split');
hhrem = cellfun(@(x) x{1}, separahora, 'uni', 0);
mmrem = cellfun(@(x) x{2}, separahora, 'uni', 0);
hhrem=cell2mat(hhrem); %Conversion from cell array to array
mmrem=cell2mat(mmrem); %Result of doing so is a in char format
hhmm=horzcat(hhrem,mmrem); %Concatenation of HH and MM.
hhmm=str2num(hhmm); %conversion from array of numeric chars to a single double.
```

But all of this seems overly complex, especially with the hassle that is to be converting formats all the time and manipulating cell arrays. So after a bit of investigation, it appears that Matlab has libraries for the purpose of manipulating date strings. All of the steps above could be simplified with the next statement:

```
arrivald=datevec(arrival_date,'yyyy-mm-dd HH:MM:SS.FFF');
removald=datevec(removal_date,'yyyy-mm-dd HH:MM:SS.FFF');
```

What *datevec* does is convert dates and times to a datetime array. A datetime array is an array that each component is the value of each kind of time. For example:

```
>> arrival_date(1,:)
        ans =
          '2012-09-21 00:08:52.000'
```

In this case, the output is a cell array, difficult to manipulate, as seen in previous cases.

```
>> arrivald(1,:)
```

```
ans=
 2012     9    21     0     8    52
```

Output is now a string of doubles. Simple and clean, no need for regular expressions or complex functions.

After applying these algorithms, all that is needed is to extract all useful data.

2.3.1 Creation of TimeElapsed

Also, after the creation of these variables, I created a new one, Time Elapsed, which is the difference between the date of arrival and the date of removal. This variable really comes into use after reading the rules of Bicing, which states that the first 30 minutes of usage of the service apply no extra cost, but after that period, a penalty applies, so an analysis of usage would be interesting.

But with the actual format, creating this new variable would become problematic. If trying to simply deduce the arrival date from the removal date in each column, at first glance would seem possible, for example:

```
arrivald(1,:)
ans =

     2012     9    21     0     8    52

removald(1,:)
ans =
     2012     9    21     0     0    24

Result
ans=
     0     0     0     0     8    28
```

But if the elapsed time is bigger than one hour in one of the variables, the calculation would fail. The solution would be to transform both dates into a seconds format, and then make the difference. Gratefully, there is a Matlab function that does this exact same thing.

```
for i = 1:342339
   t1=datevec(removal_date{i},'yyyy-mm-dd HH:MM:SS.FFF');
   t2=datevec(arrival_date{i},'yyyy-mm-dd HH:MM:SS.FFF');
Dt = etime(t2,t1); %transforms dates into seconds and calculate the elapsed time between dates
elapsed=datestr(Dt/86400,'HHMM'); %Since the output of etime is in seconds, it's necessary to
reconvert to a datetime, dividing by 86400 which is the number of seconds there is in one day.
histelapsed(i,:)=str2num(elapsed); % I called the variable like this, because now is in a numeric
format, prepared to be used in an histogram
end
```

2.3.2 Creation of coordinates and distance

Until now, the only real information of the set is the dates. It is needed something else to be able to study the data. Since no demographic information regarding the public that uses the system is available, I had to come up with some ideas.

Since it's not possible to directly use the physical station number of arrival and removal of the bike, to give any kind of quantifiable information, some supporting variables are needed.

The objective of this new variable, was to determine the distance between two points, and with help of the new created 'Elapsed' variable, it's possible to form a relationship between time and distance.

In order to calculate the distance between two points, it is needed the location of these points, which were extracted from the Bicing webpage. Using Google maps and inserting the location, it was possible to obtain the coordinates for each point of the list.

All of this process was made using an Excel spreadsheet, since the format of entering data is much more comfortable than Matlab. The only consideration to be taken is that all the columns need to have the same format in order to avoid syntax problems when manipulating in Matlab.

Bicing webpage also has a neighborhood assigned to each station, so I also added it like a supporting variable. It can be useful to find how the groups of stations are distributed.

2.4 Assembling the data

After going through multiple manipulations of the set, it felt necessary to organize it in a more comfortable way. Up to this moment, there is two separate things, the original .csv with original variables, and then the new created variables. Since it's chaotic to work with data that isn't homogeneous, a merge was needed.

The first point is to create a vector of equal length of the set to the new created variables. For example, the coordinate's one is only a list consisting in 500 rows with the name of each station and neighborhood, but the original set is 342339 rows. In each one of these rows is only listed the number of the station, so a new function is needed to translate between both sets, the number to the coordinates and then append the coordinates to the old set.

After the creation of all the vectors, using Matlab function *cswrite* is it possible to obtain a new more organized, cleaner and complete dataset.

But still, working with the whole dataset can be proven a bit inefficient, since now there are 14 different variables each one with a big number of rows. Working with this isn't considered a lot of data, but when working with a laptop it can prove to be a burden, since all the loaded variables in memory consume a lot of resources.

My solution to this is using the new datastore function from the latest releases of matlab. Datastore is a repository for data that is too large to be fitted on memory. This function is from the newer Bigdata/parallel computing libraries, which also can be used with MapReduce to compute large files of data in a very optimized way.

The Bicing set isn't as big as to be needed MapReduce to be analyzed, but Datastore provides some inbuilt functions that are very useful in the analysis, like Preview, which returns the first ten rows of the dataset, or the possibility of selection of which variables you want to load, since working with different functions, not all of them are needed in all cases and can slow your computers performance.

Usually, in my scripts, I start with creating a datastore object from a .csv, then selecting the needed variables for the operation needed.

```
ds=datastore('cleandata4.csv')

ds.VariableNames={'BikeID','CustID','removal_slot','arrival_slot','removal_station','arrival_station','BarRem','BarArr','DiaRem','DiaArr','HoraRemHHMM','HoraArrHHMM','Elapsed','Distkm'}
preview(ds)
```

removal_station	arrival_station	DiaArr	DiaRem	HoraRemHHMM	HoraArrHHMM	Elapsed	Distkm
99	237	21	21	0	8	8	1.3516
51	113	21	21	0	6	6	0.88985
124	22	21	21	0	27	27	3.0115
126	51	21	21	0	12	11	1.1205
419	34	21	21	0	5	5	0.79644

Fig1

```
ds.SelectedVariableNames = {'Elapsed','BarRem'};

T = readall(ds);
```

In this case, T would only be a table with variables Elapsed and BarRem.

3-Visualization

As the volume of data increases, it gets more difficult to see obvious things. Before getting to work with specific statistics, it's always better to have a preliminary idea of the available data.

Using visualization tools is important to see patterns easily and give an interpretation before proceeding to work on a more a complex analysis. Visualization helps to simplify the data and understand the relationships that variables have in a functional and clear way.

The process of visualization starts first analyzing only variables without taking anything else into consideration, to look for their shape and behavior, it is important to know that a condition for several kinds of analysis is that variables have a normal distribution.
After this process is completed, it is then proceeded to analyze several variables put together. In here, its possible to see the relationship between them, often looking for lineal or elliptic patterns. If the result is completely at random, then it would be needed also to know why.

3.1 Histogram

Histogram is a type of plot that represents the distribution and frequency of the data. Different numerical values are separated into bins with different heights, being the higher the height, the higher frequency it appears in the data.
It is an important visualization tool because it allows an easy judgment of the quality of the distribution. Based on the geometry of the distribution, it also helps to visible see any groupings of data.

Example of usage:

```
histogram(T.Distkm),title('Distance in Km')
```

3.1.1 Arrival and removal time

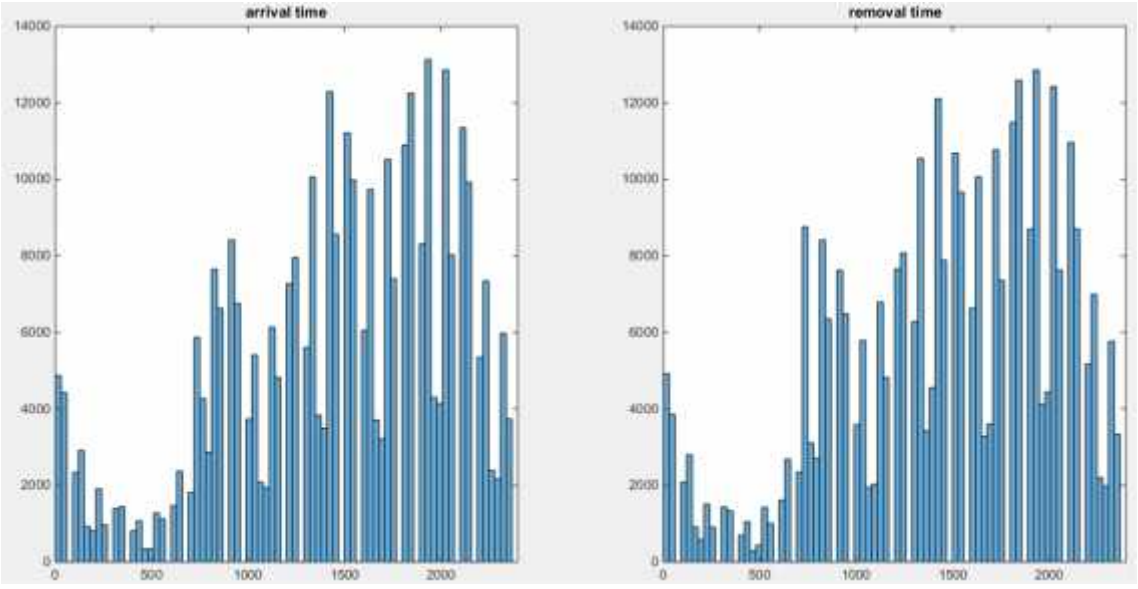


Fig2

At first glance, it is possible to conclude that the biggest peak hours are in the following periods:

-Around 8-10h
-Around 14-15h
-Around 19-20h

Also, it's possible to see that the removal time on early hours, compared to the arrival time, there is a notable difference in time. One possible explanation would be that the system is mainly used when people go to work, which is consistent with the peak hours.

In this case, the shape of the distribution looks weird because histogram

3.1.2 Elapsed time

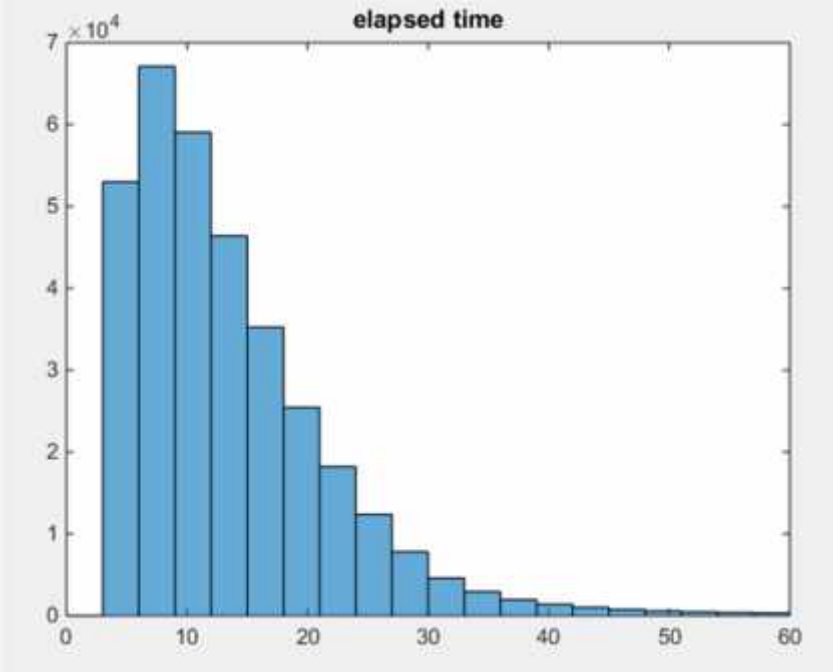


Fig3

Shows a normal distribution where it appears that the majority of people only use the service for a period of approximately 10 minutes. After 30 minutes, it decreases considerably, being consistent with the penalization of the service, that when you surpass 30 minutes limit time, you have to pay an extra fee.

3.1.3 Distance

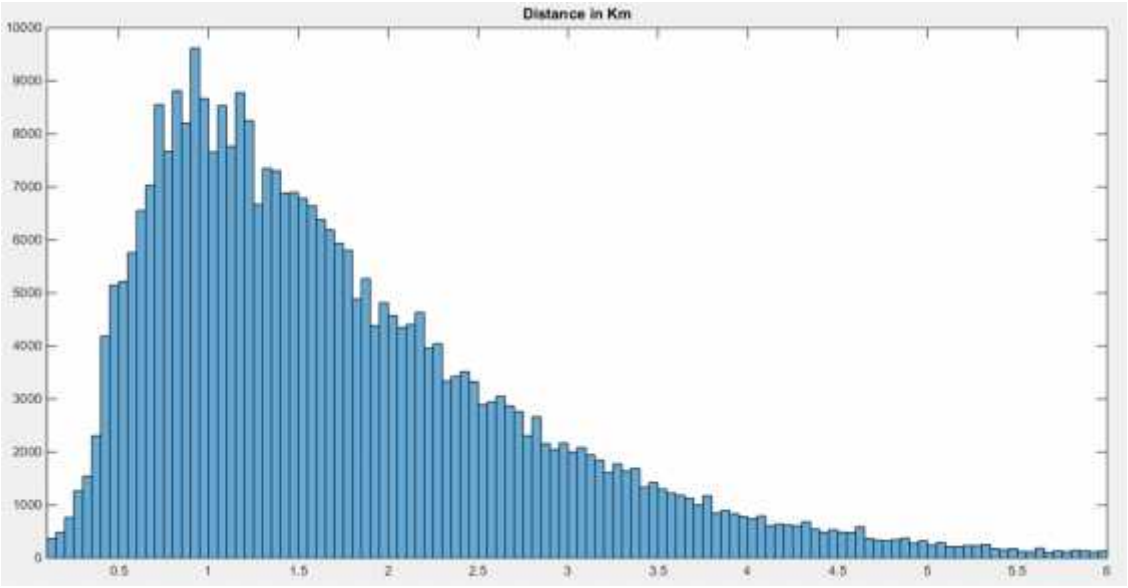


Fig4

Another clear normal distribution where the most popular distance is 1 km. These results explain that the system is probably only used to go from point to point inside the city, in distances that is maybe not long enough to be worth it to take a car or it's not desirable to use the public transport, but not short enough to go walking comfortably.

3.1.4 Neighborhoods of removal and arrival

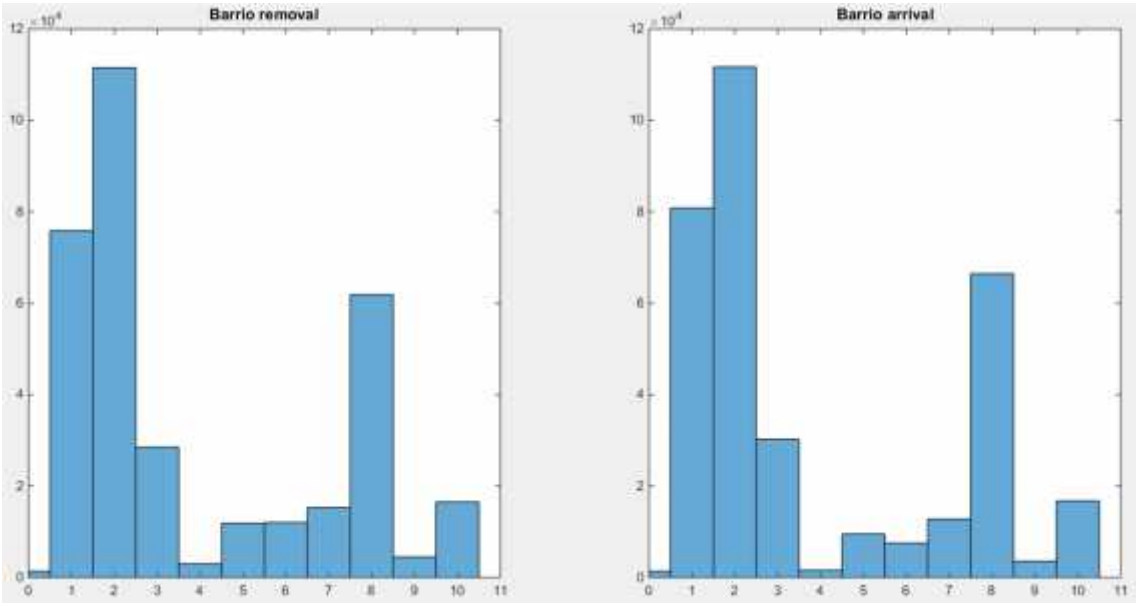


Fig5

Not all the neighborhoods have the same ratios of arrival and removal; apparently some of them take the biggest share of the load. Eixample(2), Ciutat Vella(1) and Sant

Martí(8) are the top busiest ones, while Horta-Guinardó(4), Poble Nou(9) and Sarrià-St Gervasi(6) are the ones with less overall usage.

The 0 in the plot is referencing stations that have been eliminated from the system, but since the grouping of stations from bicing webpage it's newer than the data, they still remain in the set.

3.1.5 Number of stations per neighborhood

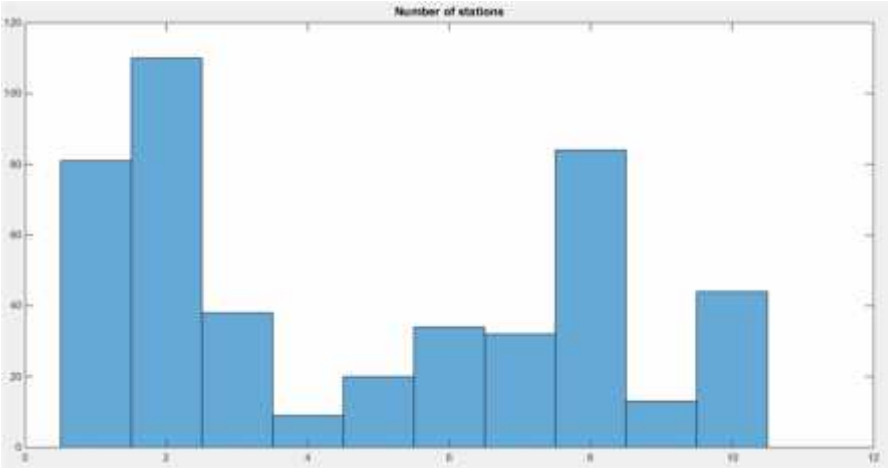


Fig6

But the previous conclusions are also evident, since the number of stations per neighborhood also varies significantly. It is to notice that comparing to the previous histograms, even if two neighborhoods have a similar number of stations, they have a significant different ratio of usage. In this case, Ciutat Vella(1) and Sant Martí(8) have similar number of stations, but seeing previous plot, Ciutat Vella has a lot more traffic.

3.1.6 Removal vs Arrival

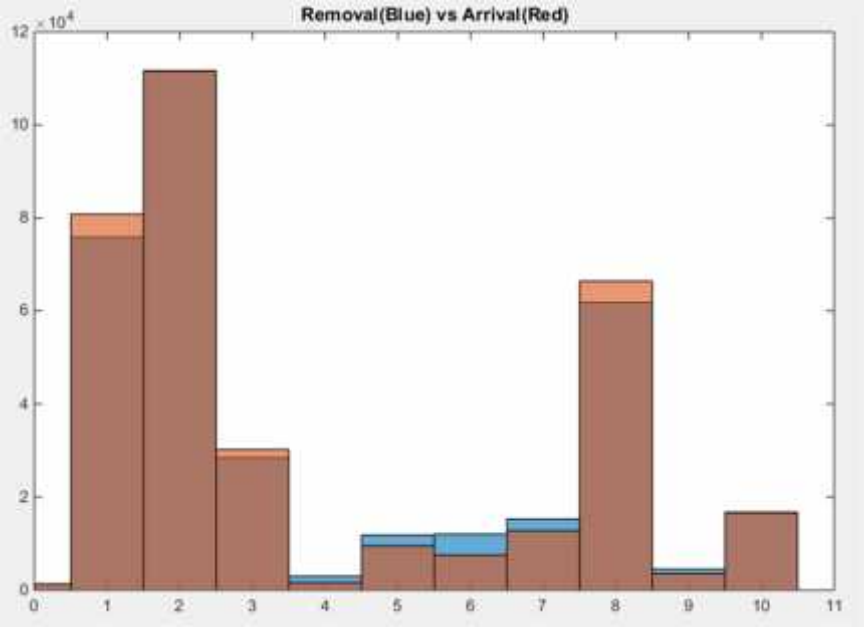


Fig7

Here it's easier to understand the flow of the system. There are also neighborhoods which are more prone to arrival than removal or vice versa. The ones with the biggest discrepancy should be the ones that the system has to look after the most, to provide or remove enough bikes for optimal usage or create more stations.

3.1.7 Arrival and Removal station

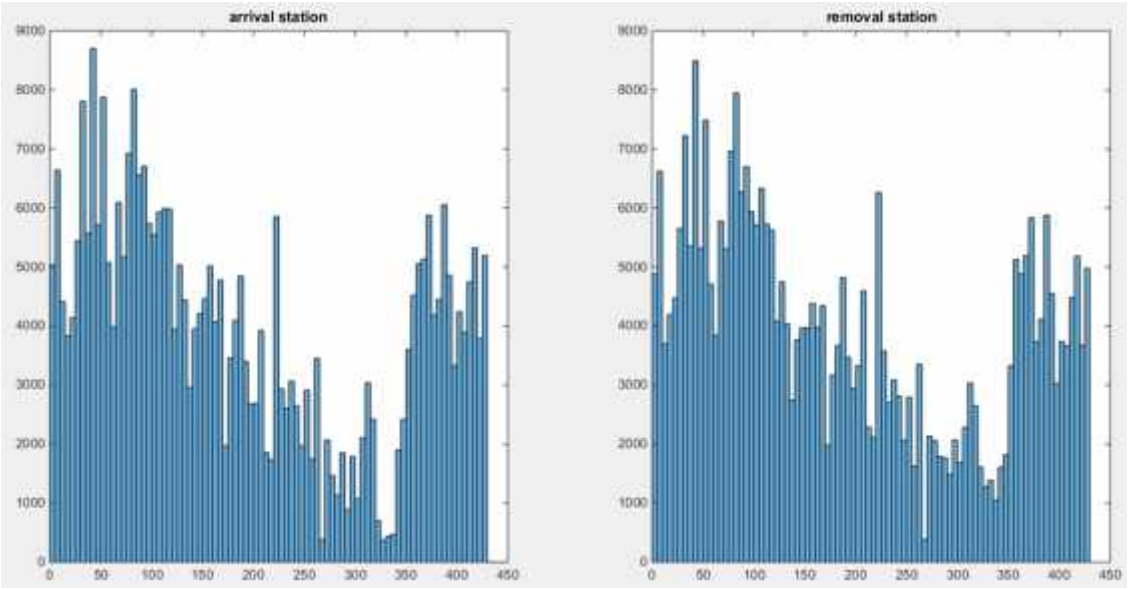


Fig8

This is the representation of the most used stations, without taking in consideration the neighborhood they are in. The numbers of the stations can seem random in the beginning, but when the system first started, they did only a limited amount of them.

The first stations are actually grouped in range by the neighborhood in sequence, but as they went updating the system, they started adding stations when they needed it the most. This is clearly visible after the 300 station mark, where suddenly it's possible to visualize a great increase in usage.

3.1.8 Arrival and Removal slot

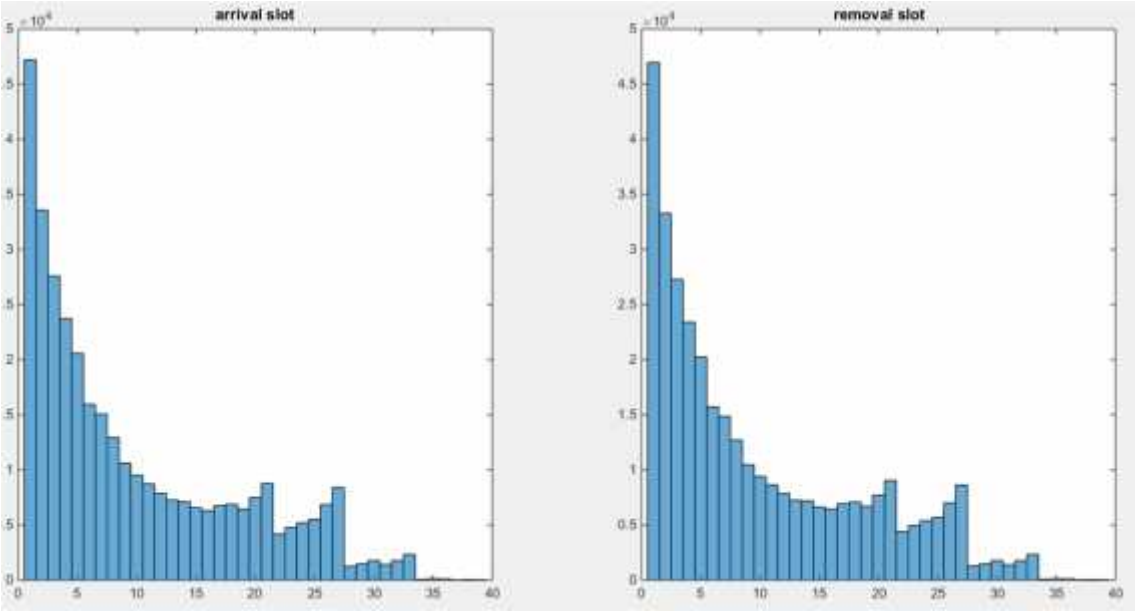


Fig9

The slot makes reference for the position of the bicing machine system the bike is stored. Not all the stations have the same storage capacity, but it's curious to see that the majority of people prefer to use slot 1, compared to 2 or 3, which are still present in all the systems. Also there are peaks around 20 and 27, which imply that people always prefer to take the bikes that are in the corners of the storage machines. This one is not really an important variable.

3.2 Boxplot

The boxplot is another useful type of visualization tool that separates the data by several equally divided subsets. There are exactly four subsets of the total data; two represent the inside of the box, two outside the box. Each segment represents a 25% of the range of the data, and the box itself is the range between 25% and 75%, being the red line the median value.
The values that are represented outside of the total range of the boxplot are the outliers, and each one is plotted as an individual point.
Advantages of the boxplot respect other kind of visualization tools are that you can actually group the data and see the variance of the same variable in the different subsets.

Example of usage:

boxplot(T.Elapsed,T.BarRem),axis([0.5 11.5 -1 40]),title('Time Elapsed by Neighborhood of Removal')

3.2.1 Time elapsed & Distance

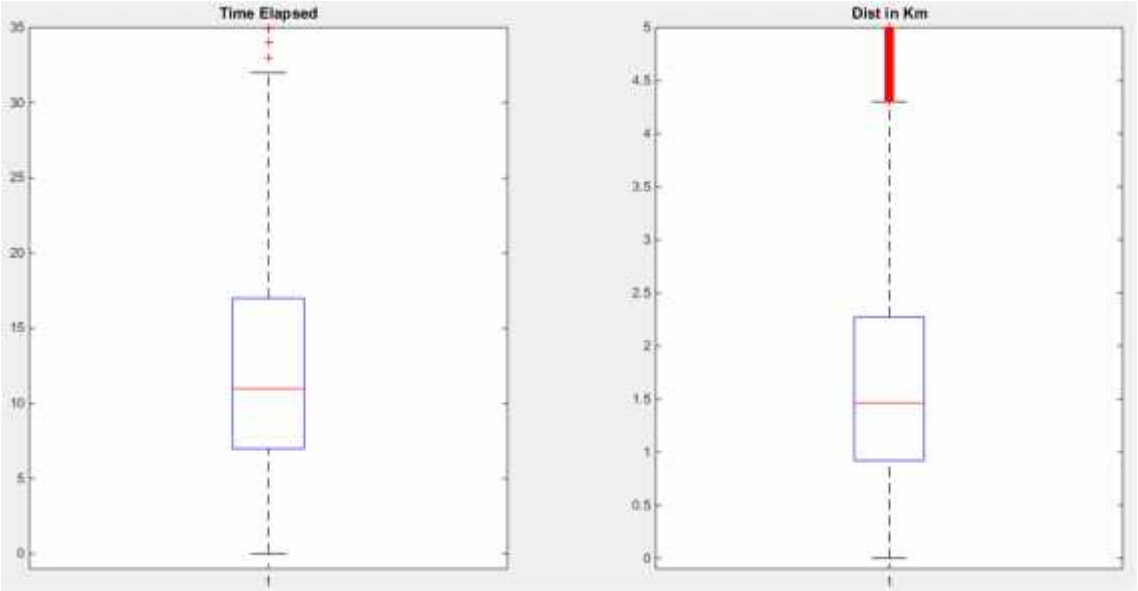


Fig10

This case is the equivalent in boxplot of the previous case of histograms. Compared to them, here it's easier to see statistic parameters more easily, like the median value, the variance and especially the outliers. In time elapsed, values over 32 minutes and in distance, over 4 km, are considered outliers in the dataset, and shouldn't be taken into account.
The red line represents the median value, which separates the insides of the box. The upper and lower insides of the box don't have the same space but they represent the same amount of data, which means, the smaller the box, the more frequent is the data inside of it.

3.2.3 Time elapsed & Distance by neighborhood of removal

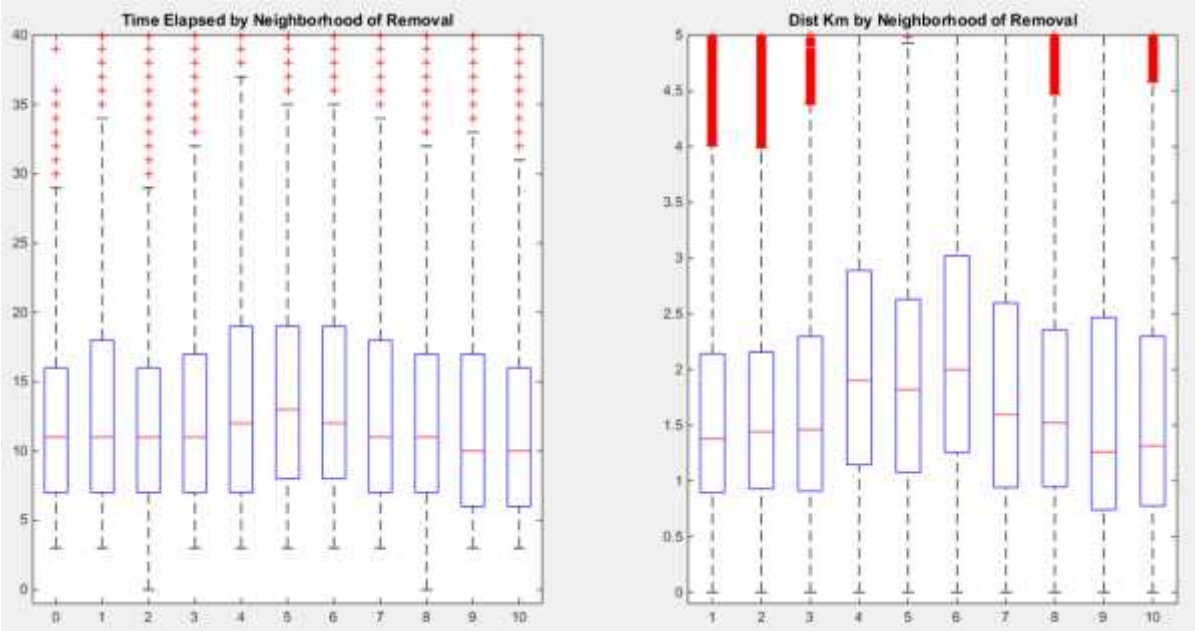


Fig11

It is also possible to study the variance of the time elapsed or the distance by neighborhood. This plot is really useful, in this case it's possible to visualize that the neighborhoods that show the most time elapsed, are also the ones that present the biggest distance, which means that the variables are correlated (which should be obvious).
This also is a positive thing, because it reinforces the hypothesis that there are neighborhoods that more residential than others, so they should travel a longer distance to go work, in comparison to others.

3.2.2 Time elapsed & Distance by day of the week

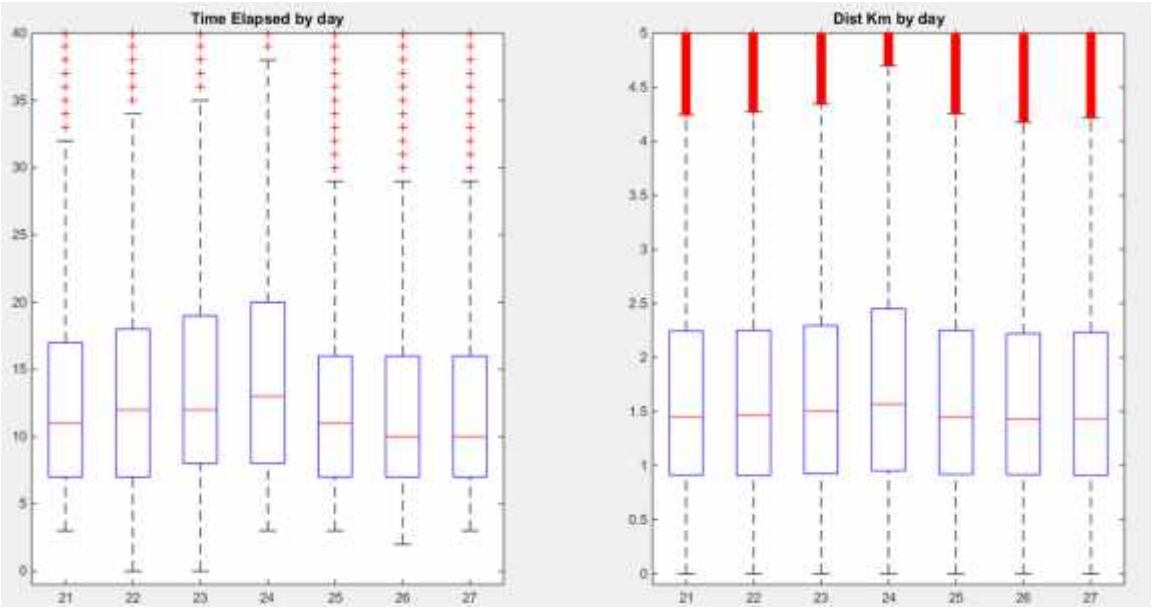


Fig12

Another good thing to watch should be the day of the week of service usage. At first glance, in some boxes there is even a variance of 5 minutes in time elapsed, which hints that the day of the week heavily matters. Incidentally, the biggest differences in boxes, from day 24 and 25, are due to being Friday and Saturday respectively, which explains that the system is a lot less congested in free days.
But in counterpart, the day of the week doesn't seem to affect too much the distance, which looks pretty much the same in all days.

3.3 Scatter

The scatterplot is another kind of plot that takes the next step after the histogram. While the histogram only plots information about one variable to look for its behavior, the Scatterplot is used for two different ones. It helps look how they are possibly correlated and grouped.

Example of usage:

```
gscatter(T.Distkm,T.Elapsed,T.BarRem,'rgbymck','oxsd'), axis([0 6 0 60])
```

3.3.1 Distance x Time Elapsed

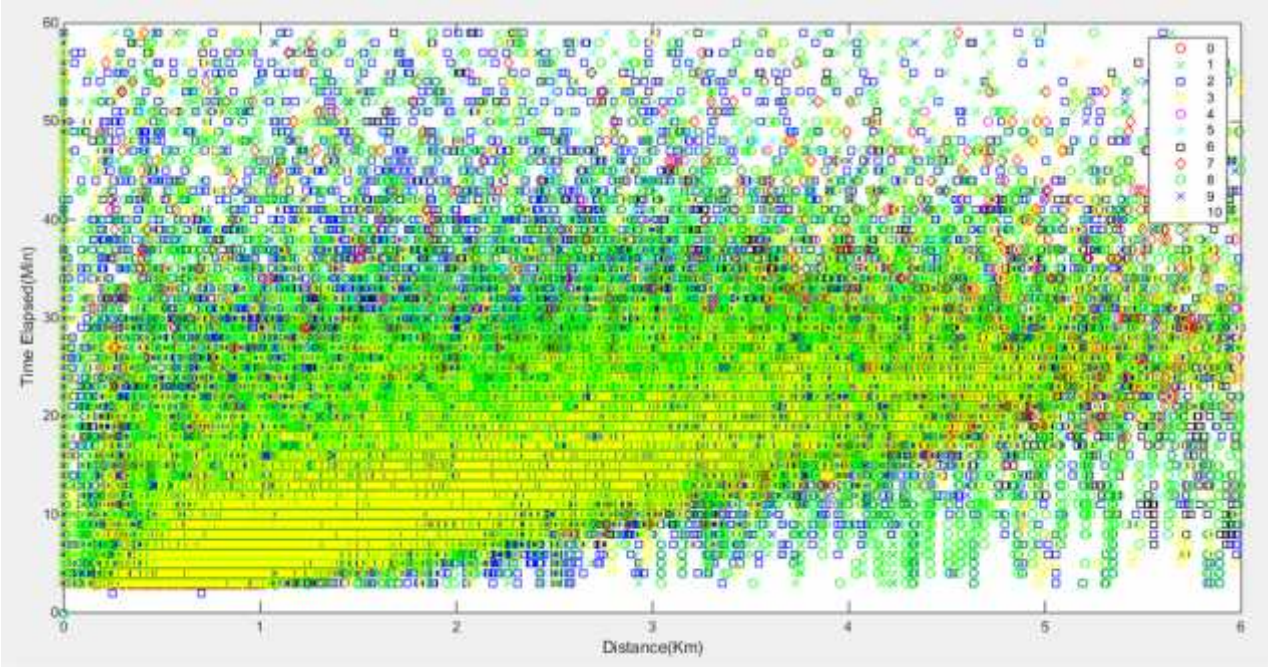


Fig13

The different colors in the scatterplot are the selected grouping, which is by neighborhood. Since there is so many points, it's not a really a good indicator because it's very difficult to extract any conclusions based on that. What is possible to determine is the behavior of the two variables, since it doesn't appear at random. It looks like they are in a linear dependency, which means that by using later a linear

regression it would be possible to determine and predict the values that time elapsed is going to take for every value of distance.

Also, it's possible to use it to confirm previous hypothesis. Based in the histograms, it was possible to determine the time of the day the system was most used, but with the use of the scatterplot, it's possible to link the usage of the system to the mean time it takes to make a trip.

3.3.2 Time elapsed x Removal hour

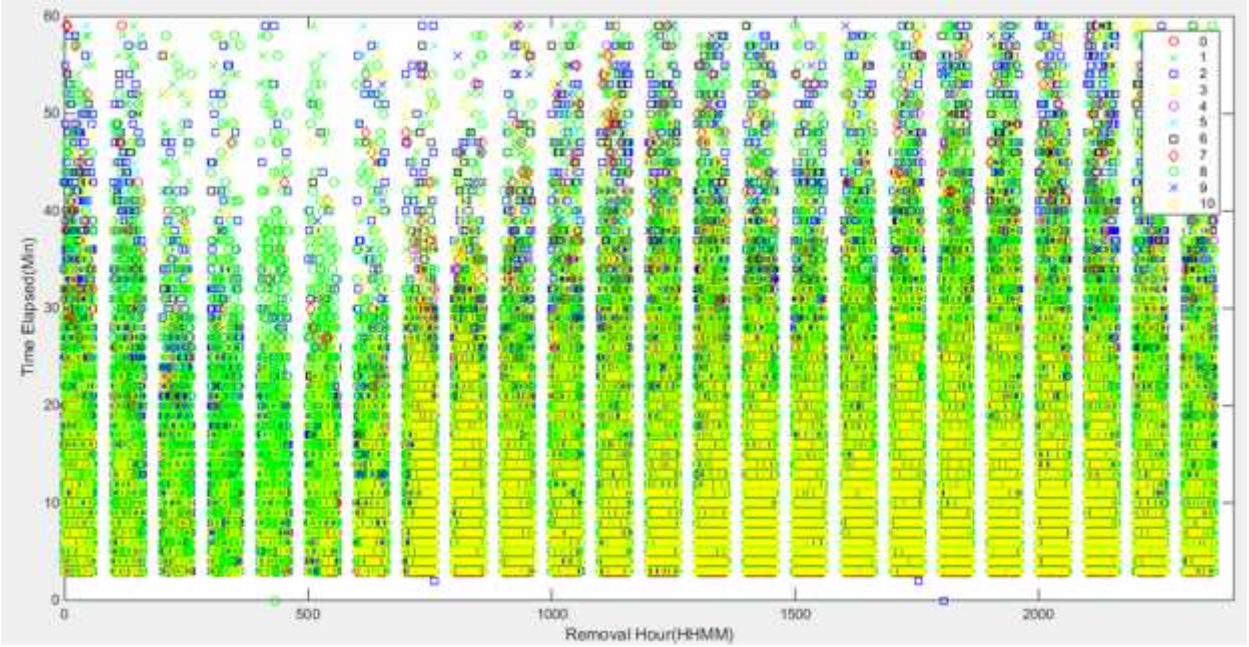


Fig14

The conclusion is also that elapsed time is correlated to Removal Hour, experiencing an increase usage between 7AM to 10PM. Since it is now proven that Distance and time elapsed are lineal dependent, I didn't feel the necessity to plot a distance x removal hour scatter.

3.3.3 Mapping

Another possible use for the scatterplot is the plotting of coordinates. Since it plots the relationship between two variables, when putting together latitudes and altitudes, the result is a map. In this case, since each pair of coordinates represents a station, the result is the map of Bicing stations in Barcelona, grouped by neighborhood. Thanks to this, it is also possible to see mistakes made during the creation of the new variables, since there are points assigned to neighborhood that are obviously wrong, because they don't match the points near to them.

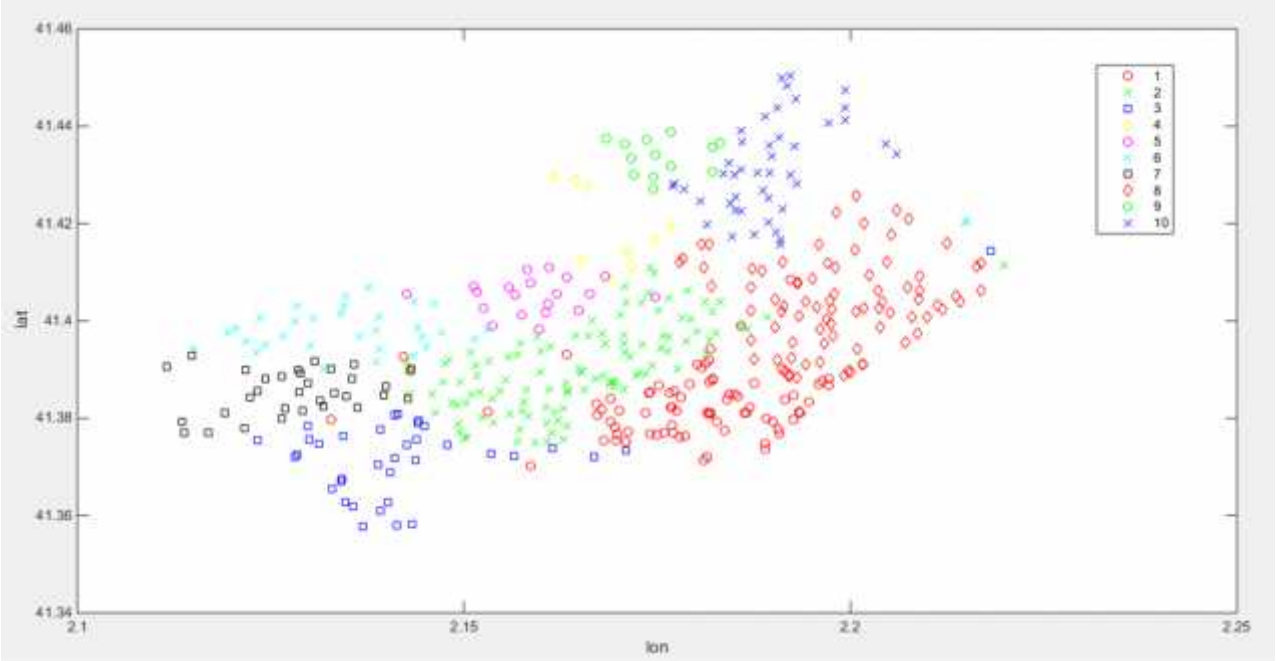


Fig15

Apparently, most of the stations are concentrated in the middle area of the map, which are the neighborhoods 1&2. Most of the surrounding neighborhoods are considered residential areas, so it is expected they have less traffic and stations, some of these areas with less stations are also in a higher elevation and they already have well located metro stations.

4 Data statistics

After the process of visualization, it's time to put numbers on the information obtained previously. The only problem in this part is to filter the data to be obtained, since applying functions in Matlab is very easy.

Let's start with the most notable variables to analyze, Time Elapsed and Distance:

Example of usage:

```
n = size(T.Distkm) %Size of the observations vector.
m = mean(T.Distkm) %Vector with the mean of each variable. It always refers to the mean value of a TRIP.
med = median(T.Distkm) % Median value
v=var(T.Distkm) % variance
```

	Distance (km)	**Time Elapsed (min)**
Size	342339	342339
Mean	1.7332	16.0501
Median	1.4643	11
Variance	1.3262	3251

But these are the global statistics, without taking in consideration the outliers. In the case of Bicing, I would take into consideration that an outlier is any case that the elapsed time is superior to 30 minutes, since it is penalized by the system. But first, let's see how much percentage of the data has an elapsed time superior to 30 min:

```
out=T(find(T.Elapsed>30),:);
len=length(out.Elapsed)

>len =15853
```

Dividing by the size, it's 4.63% of the total. So, how would look the statistics of the users that are using the system without getting any penalization?

	Distance (km)	**Time Elapsed (min)**
Size	326486	326486
Mean	1.6888	11.9453
Median	1.4454	11
Variance	1.1794	41.7860

So obviously, between the users who use the system in a normal fashion, the mean is reduced by 5 minutes from the previous case, which is largely spiked by the users who use the system for hours, which shouldn't be the standard.

4.1 Number of stations by neighborhood

Ciutat Vella	**Eixample**	**Sants Montjuic**	**Horta**	**Gràcia**	**Sarrià**	**Corts**	**San Martí**	**Nou Barrius**	**San Andreu**
81	110	38	4	20	34	32	84	13	44

Notice that the three biggest neighborhoods are mostly located in the center of the city and near the sea, while the smaller ones are mostly in the surrounding area.

4.2 Distance by bike week/day

We would obtain the unique Bikes, then for each one of them, calculate their usage, then calculate the global mean

```
y=unique(T.BikeID); %A vector with the unique elements

X=T(T.BikeID==y(i),:); % This would mean a new dataset for each unique bike
```

	Unique number	Mean Distance/week	Mean Time/week	Mean Usage/week
Bikes	4434	133.8159 km	1239 min	77.2077 uses
Users	28927	20.5116 km	189.9463 min	11.8346 uses

	Unique number	Mean Distance/day	Mean Time/day	Mean Usage/day
Bikes	4434	19.1166 km	177.0286 min	11.0297 uses
Users	28927	2.9302 km	27.1352 min	1.6907 uses

In a more realistic approximation, probably most users either don't use the system, or use it two times in a day, or use it one time and then come back with another kind of transport.

Bicing webpage states that there are 6000 bikes in the system, yet only 4434 appeared in this week's data.

4.3 Filtering by day of the week

```
out=T(find(T.DiaRem==21),:);
out=T(find(T.DiaRem==22),:);
…
```

	Length	**Mean Elapsed**	**Mean Distance**
21 - Friday	56281	14.0769	1.714
22 - Saturday	41082	16.4962	1.7020
23 - Sunday	36632	18.8515	1.7445
24 - Monday	40864	18.2092	1.8292
25 - Tuesday	57596	15.0917	1.7291
26 - Wednesday	53194	15.0980	1.7102
27 - Thursday	56690	16.1863	1.7206

So why is there such a big difference between days? Days 21,25,26,27 appear to have similar stats, while days 22,23,24 have all lower length and higher mean distance. The explanation is because day 22 and 23 are Saturday and Sunday respectively, so what is the problem with 24? It appears that the 24 of September is a local holiday in Barcelona, la Mercè. So this explains that either on weekends or in holidays, the use of the service is lower.

4.4 Filtering by neighborhoods

	1	2	3	4	5	6	7	8	9	10
Removal	2	1	8	3	10	7	6	5	9	4
Count	111501	75884	61864	28512	16525	15290	12037	11862	4513	3000
Arrival	2	1	8	3	10	7	5	6	9	4
Count	111687	80820	66466	30222	16765	12779	9551	7540	3573	1585

4.5 Filtering by Neighborhood of removal

	Length	Percentage	Mean Elapsed	Mean Distance
0 – Deleted Stations	1351	0.3946	14.7121	0
1 – Ciutat Vella	75884	22.1663	16.8090	1.6176
2 – Eixample	111501	32.5703	15.0895	1.6499
3 – Sants-Montjuic	28512	8.3286	16.0980	1.7615
4 – Horta Guinardó	3000	0.8763	17.3783	2.1104
5 - Gràcia	11862	3.4650	16.1788	1.9161
6 – Sarrià St Gervasi	12037	3.5161	17.0825	2.4036
7 – Les corts	15290	4.4663	15.6917	1.9573
8 – Sant Martí	61864	18.0710	16.2423	1.8084
9 – Nou Barris	4513	1.3183	17.2883	1.8071
10 – Sant Andreu	16525	4.8271	17.2626	1.7215

Again, Eixample, Ciutat Vella and Sant Martí are the most used neighborhoods of removal by a very large margin, with a total of 72.8% of the share, while the lowest neighborhoods, Horta Guinardó, Nou Barris and Gràcia only account for a 5.65%.

4.6 Filtering by Neighborhood of arrival

	Length	Percentage	Mean Elapse	Mean Distance
0 – Deleted Stations	1351	0.3946	14.7121	0
1 – Ciutat Vella	80820	23.6082	16.4472	1.6815
2 – Eixample	111687	32.6247	15.1054	1.6660
3 – Sants-Montjuic	30222	8.8281	16.1386	1.7649
4 – Horta Guinardó	1585	0.4630	19.5918	1.9375
5 - Gràcia	9551	2.7899	17.1610	1.8166
6 – Sarrià St Gervasi	7540	2.2025	17.9841	2.1370
7 – Les corts	12779	3.7328	15.4764	1.7479
8 – Sant Martí	66466	19.4153	16.4202	1.8876
9 – Nou Barris	3573	1.0437	16.9096	1.5281
10 – Sant Andreu	16765	4.8972	17.3267	1.6846

In the same fashion as neighborhood of removal, Eixample, Ciutat Vella and Sant Martí, top the charts with a total of 75.64% of arrivals. Horta Guinardó, Nou Barris and Sarrià St Gervasi with a total of 3.70%.

Since the dataset is from 2009 but the data from stations in neighborhoods is actual, there are some discrepancies between old and new stations, since the system has experienced changes, thus resulting in the Deleted stations statistic.

4.7 Most popular destinations by neighborhood of removal

1 - Ciutat Vella	1	2	8	3	5	7	6	10	4	9
2 - Eixample	2	1	8	3	5	7	6	10	4	9
3 - Sans Montjuic	3	2	1	7	8	6	5	10	4	9
4 - Horta Guinardó	2	8	10	1	5	9	4	6	3	7
5 - Gràcia	2	1	5	8	6	7	3	4	10	9
6 - Sarrià St Gervasi	2	7	6	1	3	5	8	4	10	9
7 - Les Corts	2	7	3	6	1	5	8	10	4	9
8 - St Martí	8	2	1	10	5	3	9	6	4	7
9 - Nou Barris	10	9	8	2	4	1	5	3	6	7
10 - San Andreu	10	8	2	9	1	4	5	6	3	7

This table represents the destinations ordered from more to less. For example, the most popular destination of service if you start at Ciutat Vella is Ciutat Vella, then Eixample. So here it's an interesting case, the neighborhoods which use more the system, tend to stay in the same neighborhood, while smaller ones travel to Eixample. Eixample is also always in one of the top destinations of every neighborhood, which can be explained by its centrality in the network. Users don't usually travel more than one neighborhood apart, probably because of the distance; they also can use other methods, like subway. Also, generally, users are more disposed to travel from north to south and from west to east, which would mean that they go from the mountain to the beach.

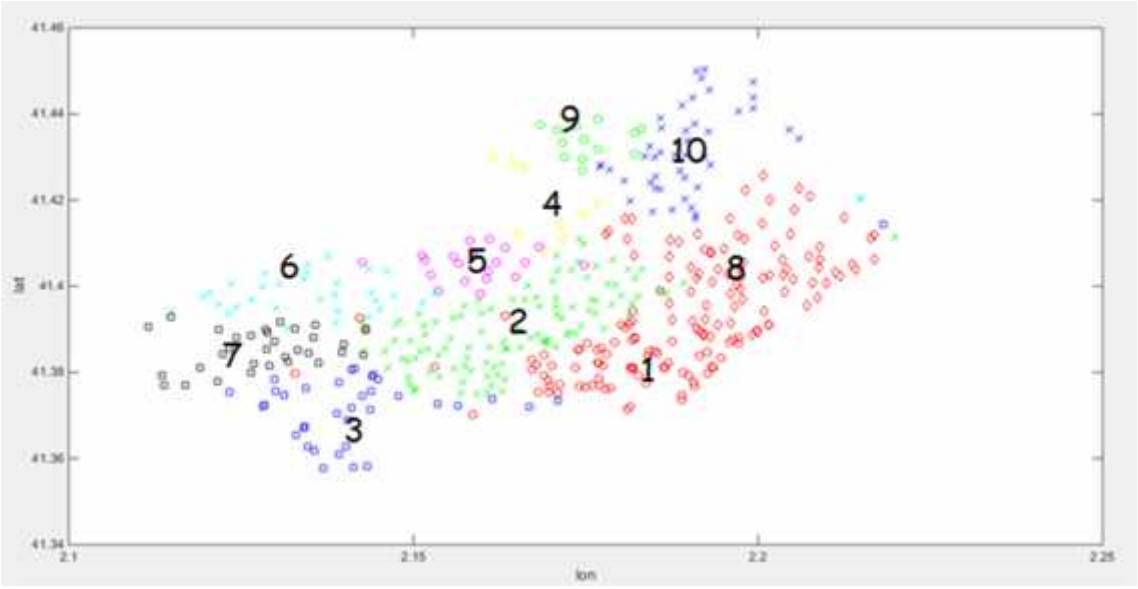


Fig16

4.8 Top 10 stations

	1	2	3	4	5	6	7	8	9	10
Removal	78	427	53	93	42	29	368	89	209	390
Count	3517	3177	2213	2211	2192	2054	2046	2036	1997	1984
Neighb.	2	1	1	2	8	2	2	2	2	1
Arrival	78	427	53	42	165	93	401	368	390	64
Count	3473	2467	2272	2256	2195	2180	2158	2121	2106	2064
Neighb.	2	1	1	8	8	2	1	2	1	2

This is a funny case, because the top station in the system has itself more usages than some whole neighborhoods, with a total of 1.02% use in a single station. It's not a surprise that all of the most used stations are part of the three top neighborhoods.

4.9 Topless 10 stations

	1	2	3	4	5	6	7	8	9	10
Removal	269	271	337	341	295	62	328	270	285	267
Count	42	50	73	107	118	139	140	141	154	156
Neighb.	10	10	6	10	9	0	6	10	10	10
Arrival	338	332	337	329	330	269	356	336	294	271
Count	28	30	32	38	41	48	50	53	54	55
Neighb.	6	6	6	6	6	10	5	6	9	10

Sarrià and San Andreu dominate with the less used stations. The most probable cause is the location of these stations, being in higher elevation, more mountain like zone than the others.

5 Analysis

Up to this point, from the statistics we have, we know that:

- Distance and time elapsed are variables that have a normal distribution and have a lineal dependency.
- Neighborhood and Day of the week are grouping variables, used to measure the effect of the previous ones.
- Stations and dates are due to being difficult to be treated, are used to create the previous distance and time elapsed.
- BikeID and CustID could also be grouping variables, but not usable like that due being so large in size, only option would be analysis one by one.

With this information we can ask several questions:

- How dependent is distance with time elapsed? Is it possible to know how long it's going to take to make a trip beforehand?

- Do the neighborhood and the day of the week really have an impact on the distance or time elapsed?
- Why are some stations more prone of usage than others? Is there some kind of user behavior depending on the neighborhood of origin?

To solve these questions it is needed to choose the best method of approach. This is done by selecting between some Machine Learning and Multivariate analysis techniques. It is important to choose well, sine techniques are not all optimal to achieve the same result.

5.1 Linear regression

Linear Regression is a method mainly used for prediction. The main objective is to determine the value a dependent variable will take with the use of several other independent variables. In the case of linear regression, the model is a straight line.

After developing the model, if a new value of an independent variable is given, without the pairing value of de dependent variable, the value can be predicted by looking at the fitted model.

The line is often fitted by the least squares approach method, which tends to look for the minimized distance from each point to the fitted straight line.[3]

In this example, the objective is to predict the value of Time Elapsed using two stations. The problem of using two stations as predictor variables is that in the set needs enough training data to be able to make the regression, also, make the permutation between all the stations to calculate the model is not a simple task. For this reason, is easier to calculate the elapsed time based on the distance, so then, when having two stations, just obtain the distance between them and predict the elapsed time for it.

The input for function, *fitlm*, is a table, where the last column is the dependent variable we want to predict. For this reason, we have to modify a bit the function of Datastore to only include the variables to use:

```
ds.SelectedVariableNames = {'Distkm','Elapsed'};
```

In this simple approach, it's not necessary to use any other variables. It is obvious that the day of the week matters or the neighborhood of origin, but it makes no sense to add them in the regression, because it would make different regressions for each one of these variables. Also, with the number of available observations, would make the computational time exponentially larger.

```
mdl = fitlm(T, 'Elapsed ~ 1 + Distkm','Intercept',false)
```

I disable the intercept in this case, because the intercept is the point where the linear regression crosses the Y axis, but if the distance is zero, the time elapsed should also be a zero. Using the command above gives the next result:

```
Linear regression model:
    Elapsed ~ Distkm

Estimated Coefficients:
              Estimate       SE       tStat     pValue
              ________    ________    ______    ______

    Distkm    7.5914      0.046892    161.89    0
```

The *pValue* refers to the statistical significance level. Usually, if the value is under 0,1, or 0,01 in some cases, it means it is significant. Since the prediction is only with the usage of one variable, it would be strange if it wasn't significant.

To visualize the results:

plot(mdl),axis([0 12 0 60])

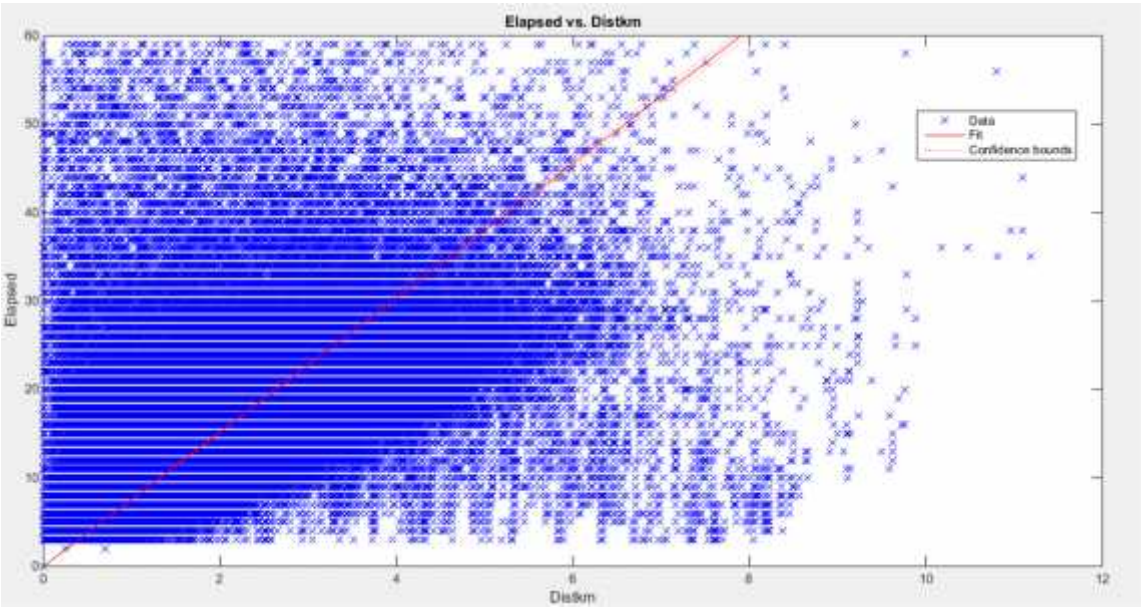


Fig17

Since there are so many points, the plot appears like that, but actually, most of them are considered outliers and not taken into account by the regression model. Also remember that after 30 minutes, there is only a 4.63% of usage of the system, so actually, most of the points are concentrated before that mark.

For the prediction part, another useful tool that Matlab brings to the visualization of the lineal regression is *PlotSlice*. It is an interactive plot that lets the user input the value of the independent variables to obtain a prediction value of the independent one.

plotSlice(mdl)

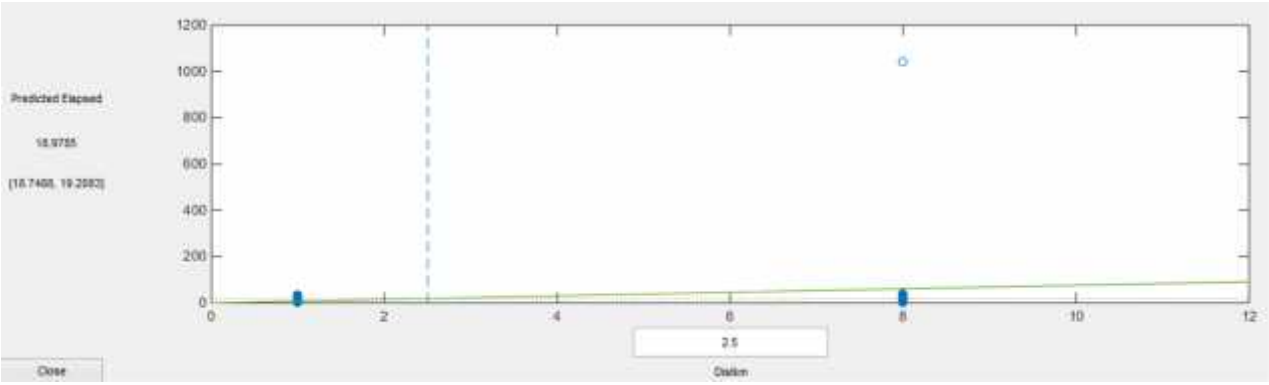


Fig18

This particular tool consumes lots of resources, so it's always possible to use only the predict function, which would return the value of elapsed time

Y=predict(mdl,2.5);

>Y=18.9785

This way it's possible to obtain new values with the previously obtained trained data. This is an interesting functionality that could be added to the Bicing service webpage, like most of the transport prediction trip apps that exist in the moment. Another useful thing that could be used to improve the prediction is the weather information, the day of the week, the hour of the day…given enough computational capacity. But even in the best cases, the difference would only account for a 1-2 minutes difference, as seen previously in the statistics.

It would also be greatly influenced by a personal profile, since each individual greatly varies from another, with some people in the system being able to make the same distance in half the elapsed time as others in a consistent manner.
Studying the individual usage on BikeID or CustomerID of the system could also help discover differences between destinations depending on the day of the week.

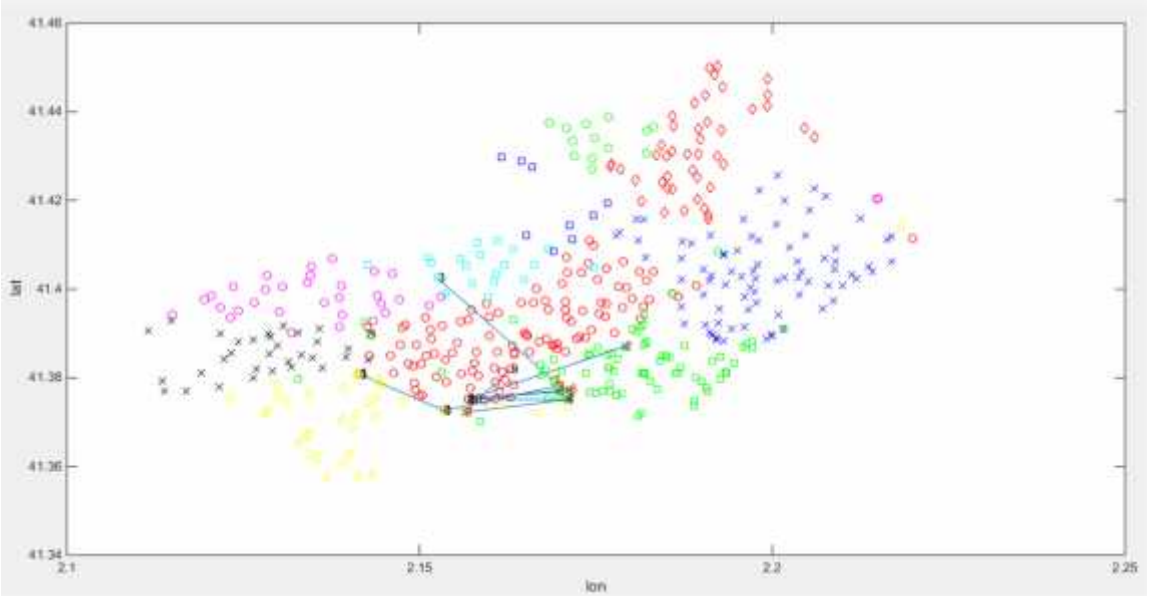


Fig19

These are the trips of a random user. Time elapsed could be tailored by distance and user, given the data. Notice that there are two areas that concentrate most of trips. One possible theory could be their home and work, while the others are just some other sporadic usages.

Maybe in this case it's not really worth it to apply a lineal regression, since the model is so simple it doesn't tell us anything new at all. Because it is influenced by other factors that don't appear in the data, the actual model is already an approximation and the scale of time varies so little, it wouldn't be really necessary to calculate the time. Also, the user is probably already using other kinds of transport if the trip is longer than a certain distance or already knows the time if it's a destination they frequently travel to.

5.2 ANOVA

ANOVA (ANalysis Of the VAriance) is a statistical test that helps determine if the means of several groups have a significant difference, in this case, to see if it really is an influencing factor the day of the week or the neighborhood is taken. If the variable doesn't 'pass' the ANOVA test, would mean that the variations of mean regarding different grouping are only because random effects or noise.[5]

In Matlab, using the ANOVA function will return a boxplot with the variable classified by the different groups (the same obtained previously with the boxplot function) and a table with some stats, like in section 3.2.3.

```
[p,table,stats] = anova1(T.Elapsed,T.DiaRem)
```

ANOVA Table

Source	SS	df	MS	F	Prob>F
Groups	1.21441e+06	6	202401.8	62.32	1.26192e-77
Error	1.11191e+09	342332	3248		
Total	1.11312e+09	342338			

Fig20

The key to determine if the variation of the mean, is random or not, is the F-Statistic.

$$F = \frac{\text{between-group variability}}{\text{within-group variability}}$$

If the F-parameter is high, it explains that the variation between sample means dominates over the variation within groups, which in this cases translates that the variation of elapsed time between days is larger than the variation of distances within the same day, confirming the hypothesis that the elapsed time of a trip is closely related to the day of usage.

P-Value is also another value of interest, it indicates if the results are statistically significant, normally they are if p-value is less than 0,1, or 0,01 in some cases.

Another way to visualize the data is the use of *multcompare*, a function that groups data with similar means:

multcompare(stats)

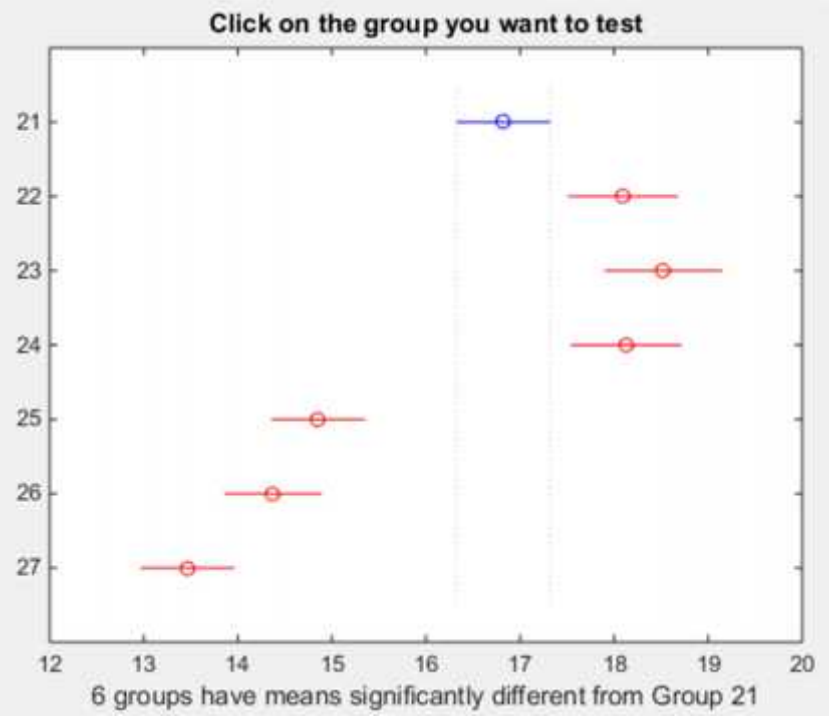


Fig21

Remember that 22 and 23 is the weekend and day 24 is holiday, meaning that in these days, the mean elapsed time per trip is higher than in weekdays. My theory would be that people don't feel as rushed to go to work or come home and use the service in a more relaxed way. Day 21, being Friday, would be explained as a mix of weekdays and weekend.

We could also study the effect of several variables in the elapsed time, for example, the day of usage and the neighborhood of removal with:

```
[p,table,stats] = anovan(T.Elapsed,{T.DiaRem,T.BarRem})
```

Analysis of Variance

Source	Sum Sq.	d.f.	Mean Sq.	F	Prob>F
X1	1205453.1	6	200908.8	61.86	4.80074e-77
X2	193904.4	10	19390.4	5.97	4.12457e-09
Error	1111712595.1	342322	3247.6		
Total	1113120910.4	342338			

Fig22

But the results are very similar to the ones without taking into account the neighborhood, since it appears that the day of the week carries a lot more weight to the variance, as can be seen in the F parameter.

5.3 Principal Components Analysis

Principal component analysis or PCA in short, is a type of statistical test used to analyze the relationships between large numbers of variables and try to reduce the number of variables in smaller set of factors while minimizing the loss of information, very similar to factor analysis, which objective is to detect an structure of the data.

This technique could be either used for exploratory analysis, or hypothesis testing. It studies the structure of the correlations among the variables by defining sets of variables that are interrelated, called principal components. These values should represent a new concept described by the aggrupation of the variables they are composed.

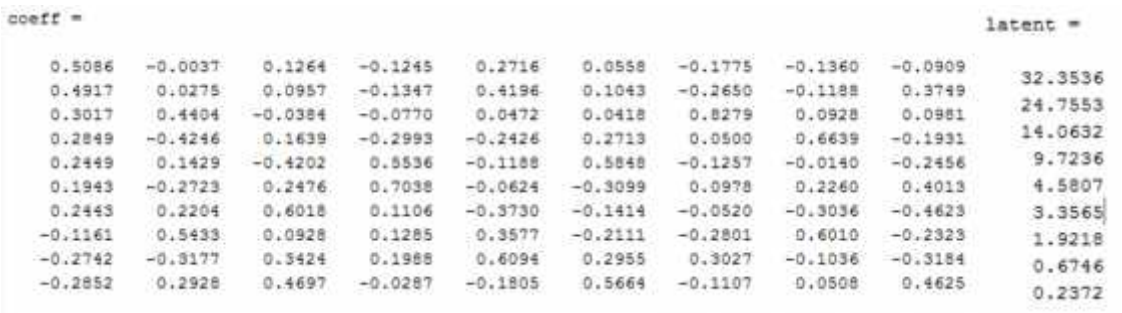
To do this, PCA algorithm uses a transformation where the first component has the largest variance, and the next component has the largest variance but being orthogonal to the one before. These components are orthogonal because they are the Eigen values of the correlation or covariance matrix. The transformation and coefficients are obtained by using a singular value decomposition of the matrix.[4]

In this particular case, PCA will be applied to the previous "Most popular destinations by neighborhood of removal" to try to find if there are any underlying reasons and relationship of why these destinations are as they are. A .csv with this data was created

only for this case, consisting in a 10x10 matrix with the neighborhoods and the 10 more popular destinations, sorted from top to bottom.

There are two main methods for obtaining the principal components, the correlation and covariance method. Both yield similar results, but covariance is better if the values in our data are in different scales, while correlation is better if they are similar. In the PCA function in Matlab, by default utilizes the correlation method, which will be the used in this example.

[coeff,score,latent,~,explained] = pca(X)



coeff =

0.5086	-0.0037	0.1264	-0.1245	0.2716	0.0558	-0.1775	-0.1360	-0.0909
0.4917	0.0275	0.0957	-0.1347	0.4196	0.1043	-0.2650	-0.1188	0.3749
0.3017	0.4404	-0.0384	-0.0770	0.0472	0.0418	0.8279	0.0928	0.0981
0.2849	-0.4246	0.1639	-0.2993	-0.2426	0.2713	0.0500	0.6639	-0.1931
0.2449	0.1429	-0.4202	0.5536	-0.1188	0.5848	-0.1257	-0.0140	-0.2456
0.1943	-0.2723	0.2476	0.7038	-0.0624	-0.3099	0.0978	0.2260	0.4013
0.2443	0.2204	0.6018	0.1106	-0.3730	-0.1414	-0.0520	-0.3036	-0.4623
-0.1161	0.5433	0.0928	0.1285	0.3577	-0.2111	-0.2801	0.6010	-0.2323
-0.2742	-0.3177	0.5424	0.1988	0.6094	0.2955	0.3027	-0.1036	-0.3184
-0.2852	0.2928	0.4697	-0.0287	-0.1805	0.5664	-0.1107	0.0508	0.4625

latent =

32.3536
24.7553
14.0632
9.7236
4.5807
3.3565
1.9218
0.6746
0.2372

Fig23

The coefficients matrix is composed by the representation of the new created components in the columns, and the weights of each variable, in the rows, in this case, the different neighborhoods. So there is a total of 9 principal components, which is consistent with the objective of data reduction which is one of the main purposes of PCA.

The latent values express the weight of each component in the global representation of the data, meaning they include most of the information about the relationship of the variables, and the explained variable normalizes the latent values. For example, in this case, the first 4 components, explain 89% of the data. The point of it is to find a compromise between the number of components and the % of data to be preserved, meaning, the more components, better representation of the data, but more complexity. The plot that explains the variation of the components is called scree plot.

pareto(explained)

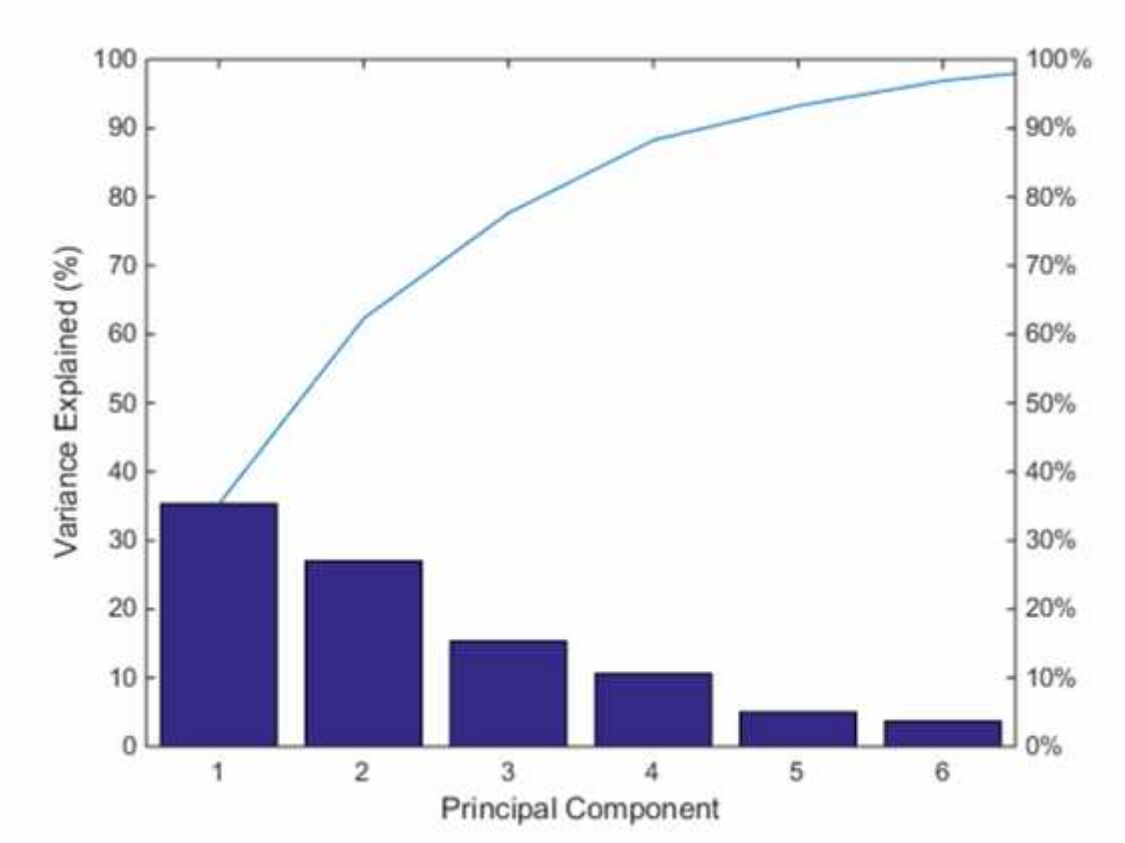


Fig24

So analyzing the coefficients, taking the first four columns which represent the first four components:

- The first one has high values on neighborhoods 1 and 2.
- The second one, on 8 and a bit of 3.
- The third one, only in 7.
- And the fourth one, in 5 and 6.

So my hypothesis would be that PCA is actually grouping the neighborhoods by similar behavior of destinations.

- Actually, neighborhoods 1 and 2, have almost exactly the same destinations, also being one next to the other, and being both the most popular neighborhood destinations in Bicing.
- In the same line of thought, 3 and 8 are next in popularity by destination, they are also after 1 and 2 in terms of usage, but they are comparatively near 2, which is the most popular destination.
- 7 being alone would mean that it has a more unique behavior than the rest.

- Then 5 and 6, which are also next to each other. They are similar in that sense to 1 and 2, but being a bit far away from 2, which is still the most popular destination.

The other factors that I didn't consider, take into account the lowest of usage of neighborhoods, which would be 4, 9 and 10, which are also the one with the least usage.

The conclusion of the analysis is that neighborhood destination is actually explained most likely by the singular position of the neighborhoods and their distance to 2 and 1, being them the center of the destinations.

5.4 Cluster Analysis

Cluster analysis makes reference to a set of techniques that helps group the available information into smaller sets in a way that the objects in the same set have a maximal association; while with objects in other sets have a minimal association. It can help discover hidden structures in the data without providing any previous explanation, therefore, is often used in exploratory analysis and the results require some form of interpretation.

There are several algorithms mainly used for clustering, k-means, tree clustering, two-way joining… But my algorithm of choice is k-means.
K-means is probably the simplest clustering algorithm there is and the easiest one to implement. It also works really well with multivariate data.

What k-means does is, first, assigns points of the data randomly into k clusters, then calculates the distance (Euclidean, sum of squares) from every point to the cluster. If the point is the closest to its cluster, then it stays in the same cluster, if not, change it. Then, with all the points in the same cluster, it computes it mean distance position, which is the centroid. Then the process is repeated taking the centroid as comparing position in the cluster. The algorithm is then repeated until equilibrium.
Since the initial choosing of clustering points is random, sometimes it is possible to obtain different clusters of data.[6]

The practical use of this kind of analysis in this case, was to determine if it exist any kind of grouping of neighborhood by its behavior in destinations. For this purpose, I used the matrix obtained previously in the statistics section.

The only thing to consider in the use of this algorithm is the initial number of k clusters. For this task, it's possible to use the results obtained from the principal components analysis. Viewing the scree plot can give us an idea of the optimal number of clusters to be used without losing too much information or either over fitting the data.[7]

I considered that 4-5 clusters was a pretty good approach, since the changes are also minimal. To call for the clustering algorithm:

```
[idx,C] = kmeans(X,4);
```

In this case, the important thing is obtaining idx, which assigns each observation to a cluster. After applying a dictionary function to map the neighborhoods to clusters, this are the results obtained:

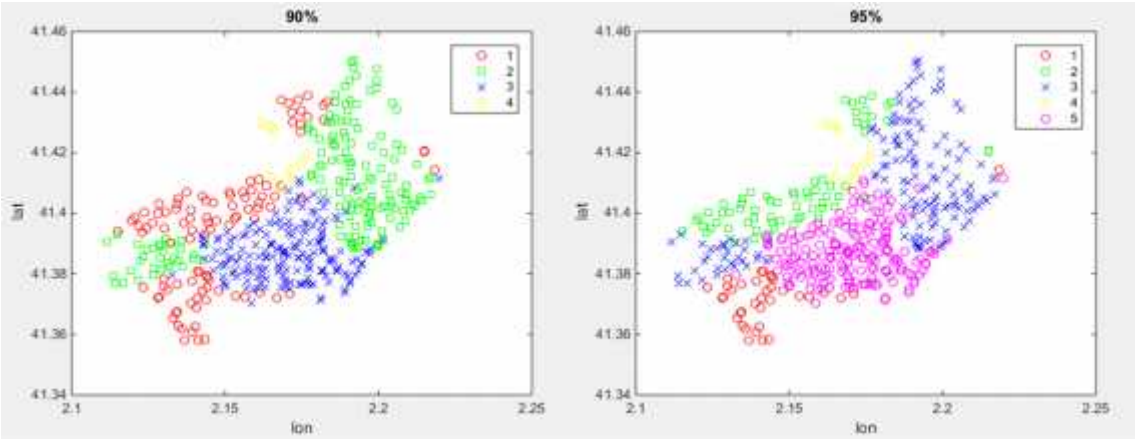


Fig25

- There is always one big cluster which groups neighborhoods 1 and 2. They are the most centric ones, also the favorite destination by all others, even themselves.
- Another big group which groups 8, 7 and 10, which still are a prominent destination after the first group, even if they are not the most popular destination.
- A group containing lesser neighborhoods, 4, 5, 6. They are not so popular destinations and are smaller. Usually this group doesn't have priority to travel inside the same group.
- A small group containing 9, which is the least popular destination, also the neighborhood with the least number of stations.

Why is this happening?

- Ciutat Vella and Eixample are the core of Barcelona. In these neighborhoods it is possible to find the majority of the most emblematic and touristic places of the city. It is also one of the most populated areas of the city and includes lots of destinations for entertainment, shop, the sea…
- Les Corts and San Martí are also important destinations, but for other reasons. In Les Corts it is possible to find the popular destination El camp nou, as well as several universities and the main offices of several companies located in the city. San Martí on the other hand, it is a highly populated district, which thanks to an initiative by the city council, it promotes the development of companies related to new technologies, IT, medical field, energy…

- The low popularity of Les Corts but higher traffic than other areas could be explained due to the majority of usage is because internal use of the system inside the same neighborhood, while people from other neighborhoods prefer to come using metro stations, since they are very well positioned.
- Sans Monjuic, Gràcia, Sarrià are mainly residential areas. Gràcia has a prominent nightlife scene. Sarrià one of the richest. These neighborhoods are also a bit more into the mountains that the previous ones. Higher income is also a reality in these neighborhoods, which could explain people using other kinds of transports.
- Nou Barris, it is probably one of the poorest neighborhoods of Barcelona, which can explain why is one of the least popular destinations.

6 Conclusions

About Matlab

- Strong documentation with powerful examples, especially in the use of the inbuilt functions about machine learning and multivariate data analysis. Most of the time, it's only needed to call a function with default parameters to obtain significant results.
- It is very intuitive to know the reason of why you get an error; it helps a lot specially if not having strong prior knowledge in coding.
- The biggest problems encountered while using Matlab are while the cleaning and organizing the data, since it's mostly intuitive. Once the data is ready, getting to use functions is very easy. Sometimes a lot of time is lost because you don't know a function that does something exists already for it.
- Knowing the different formats of data in Matlab is very important, especially when manipulating cell strings that contain dates, which are usually very common in datasets.

About the dataset

- There is a 4.63% of users who use the system for more than 30 minutes, which is the time that the penalization starts.
- For the people who use the service for less than 30 min, the average distance of a trip is 1,7km and 12 minutes.
- Eixample, Ciutat Vella and San Martí are the neighborhoods with the most stations (110,81,84), also the ones who get more traffic.
- Nou Barris, Horta and Sarrià are the ones with the least traffic, but Sarrià has a lot more stations than the others (13,4,34).
- The times of the day with more usage of the system are between 7-10h, 13-15h and 18-20h, which match with working schedules.
- Number of bikes is 4434, each one does 19km a day, 177 minutes of usage in a total of 11 trips.

- There are 28927 unique users in this week, who do 2.9km a day, in 27 min and use the system 1.6 times a day.
- During week days, the system is more used than on weekends or holidays.
- Some neighborhoods regularly do longer trips than others.
- Eixample and Ciutat Vella are the most popular destinations, both in choice and traffic.
- Nou Barris is the least popular destination choice, followed by Les Corts, but traffic isn't comparable, since most of Les Corts traffic is to itself.
- In general, trips go from mountain to the sea.
- Trips don't usually last for more than two neighborhoods of separation
- The most used station in the system, 78 in Plaza Universitat, has 1.02% of the total traffic with 1000 uses/day (500 arrivals and 500 removals), it's almost all the traffic that the whole Nou Barris has, with a 1.04% and 13 stations.
- On weekends and holidays, the time of a trip is reduced by 4-5 minutes in some cases.

7 Bibliography

[1]Pearson - Multivariate Data Analysis, 7th edition

[2] MAD Skills: New Analysis Practices for Big Data -2009

[3]Matlab documentation

[4]A Tutorial on Principal Component Analysis – 2005
http://www.cs.cmu.edu/~elaw/papers/pca.pdf

[5]One way ANOVA -
http://www.statstutor.ac.uk/resources/uploaded/onewayanova.pdf

[6]Cluster Analysis - http://www-users.cs.umn.edu/~kumar/dmbook/ch8.pdf

[7] Cluster Analysis - http://www.statsoft.com/Textbook/Cluster-Analysis#vfold

[8] Applied Multivariate Statistical Analysis -
https://onlinecourses.science.psu.edu/stat505/node/1

8 Acknowledgements

I would like to express my gratitude for my supervisors Claudio Feijoo and José Luis Melús. Without their guidance and suggestions this thesis wouldn't be possible and they also taught me things I could not know.

My thanks goes also to all the team in the Sino-Spanish campus in Tongji University for letting me use their office and materials, and answering some of my questions, also, for the free coffee.

I thank all the friends I have made in China, who have made my stay at the country one of the best experiences in my life and supported me while there.
Last, I would like to thank my family for making all these possible and supporting me until now.

9 Code

Statistics

```
load ('coordsv3.mat')
ds=datastore('cleandata4.csv')

ds.VariableNames={'BikeID','CustID','removal_slot','arrival_slot','removal_station','arrival_station','BarRem','BarArr','DiaRem','DiaArr','HoraRemHHMM','HoraArrHHMM','Elapsed','Distkm'}

ds.SelectedVariableNames =
{'CustID','Distkm','removal_station','arrival_station','DiaRem','HoraRemHHMM','HoraArrHHMM','Elapsed','BarRem','BarArr'};
preview(ds)
T = readall(ds);
%%
n = size(T.Distkm) %Size of the observations vector.
m = mean(T.Distkm) %Vector with the mean of each variable
med = median(T.Distkm)
v=var(T.Distkm) % variance
%%
n = size(T.Elapsed) %Size of the observations vector.
m = mean(T.Elapsed) %Vector with the mean of each variable
med = median(T.Elapsed)
v=var(T.Elapsed) % variance
%%
c = cov(T.Distkm,1) %Covariance matrix
e = sqrt(diag(c))
Correlation = c./(e*e') %Correlation matrix
%%
%> 30 min
out=T(find(T.Elapsed>30),:);
len=length(out.Elapsed)
%%
%< 30 min
out=T(find(T.Elapsed<31),:);
n = size(out.Elapsed) %Size of the observations vector.
m = mean(out.Elapsed) %Vector with the mean of each variable
```

```
med = median(out.Elapsed)
v=var(out.Elapsed) % variance
%%
n = size(out.Distkm) %Size of the observations vector.
m = mean(out.Distkm) %Vector with the mean of each variable
med = median(out.Distkm)
v=var(out.Distkm) % variance
%%
 X=T(T.DiaArr==21,:);%viernes
 length(X.Elapsed) %56281
 mean(X.Elapsed)%14.0769
 mean(X.Distkm)%1.7174
 %%
 X=T(T.DiaArr==22,:);%sabado
 length(X.Elapsed)%41082
 mean(X.Elapsed)%16.4962
 mean(X.Distkm)%1.7020
  %%
 X=T(T.DiaArr==23,:);%domingo
 length(X.Elapsed)%36632
 mean(X.Elapsed)%18.8515
 mean(X.Distkm)%1.7445
  %%
 X=T(T.DiaArr==24,:);%lunes Barcelona, 20 de maig i 24 de setembre http://laboral.cat/wp-
content/uploads/2012/12/Calendari-de-festes-locals-2013-a-Catalunya1.pdf
 length(X.Elapsed)%40864
 mean(X.Elapsed)%18.2092
 mean(X.Distkm)%1.8292
  %%
 X=T(T.DiaArr==25,:);%martes
 length(X.Elapsed)%57596
 mean(X.Elapsed)%15.0917
 mean(X.Distkm)%1.7291
 %%
 X=T(T.DiaArr==26,:);%miercoles
 length(X.Elapsed)%53194
 mean(X.Elapsed)%15.0980
 mean(X.Distkm)%1.7102
  %%
 X=T(T.DiaArr==27,:);%jueves
 length(X.Elapsed)%56690
 mean(X.Elapsed)%16.1863
 mean(X.Distkm)%1.7206
 %%
  X=T(T.BarRem==0,:);%Estaciones eliminadas
 length(X.Elapsed)%1351
 mean(X.Elapsed)%14.7121
 mean(X.Distkm)% 0
 %%
   X=T(T.BarRem==1,:);%Ciutat Vella
 length(X.Elapsed)%75884
```

```
mean(X.Elapsed)%16.8090
mean(X.Distkm)% 1.6176
%%
  X=T(T.BarRem==2,:);%Eixample
length(X.Elapsed)%111501
mean(X.Elapsed)%15.0895
mean(X.Distkm)% 1.6499
%%
   X=T(T.BarRem==3,:);%Sants-Monjuic
length(X.Elapsed)%28512
mean(X.Elapsed)%16.0980
mean(X.Distkm)% 1.7615
%%
   X=T(T.BarRem==4,:);%Horta Guinardo
length(X.Elapsed)% 3000
mean(X.Elapsed)%17.3783
mean(X.Distkm)% 2.1104
 %%
   X=T(T.BarRem==5,:);%Gracia
length(X.Elapsed)%11862
mean(X.Elapsed)%16.1788
mean(X.Distkm)% 1.9161
 %%
   X=T(T.BarRem==6,:);%Sarria Sant Gervasi
length(X.Elapsed)%12037
mean(X.Elapsed)%17.0825
mean(X.Distkm)% 2.4036
 %%
   X=T(T.BarRem==7,:);%Les Corts
length(X.Elapsed)%15290
mean(X.Elapsed)%15.6917
mean(X.Distkm)% 1.9573
 %%
   X=T(T.BarRem==8,:);%Sant Marti
length(X.Elapsed)%61864
mean(X.Elapsed)%16.2423
mean(X.Distkm)% 1.8084
 %%
   X=T(T.BarRem==9,:);%Nou Barris
length(X.Elapsed)%4513
mean(X.Elapsed)%17.2883
mean(X.Distkm)% 1.8071
 %%
   X=T(T.BarRem==10,:);%San Andreu
length(X.Elapsed)%16525
mean(X.Elapsed)%17.2626
mean(X.Distkm)% 1.7215
 %%
 X=T(T.BarArr==0,:);%Estaciones eliminadas
length(X.Elapsed)%1351
mean(X.Elapsed)%14.7121
```

```
mean(X.Distkm)% 0
%%
  X=T(T.BarArr==1,:);%Ciutat Vella
length(X.Elapsed)%80820
mean(X.Elapsed)%16.4472
mean(X.Distkm)% 1.6815
%%
   X=T(T.BarArr==2,:);%Eixample
length(X.Elapsed)%111687
mean(X.Elapsed)%15.1054
mean(X.Distkm)% 1.6660
%%
    X=T(T.BarArr==3,:);%Sants-Monjuic
length(X.Elapsed)%30222
mean(X.Elapsed)%16.1386
mean(X.Distkm)% 1.7649
%%
    X=T(T.BarArr==4,:);%Horta Guinardo
length(X.Elapsed)%  1585
mean(X.Elapsed)%19.5918
mean(X.Distkm)% 1.9375
 %%
    X=T(T.BarArr==5,:);%Gracia
length(X.Elapsed)%9551
mean(X.Elapsed)%17.1610
mean(X.Distkm)% 1.8166
 %%
 X=T(T.BarArr==6,:);%Sarria Sant Gervasi
length(X.Elapsed)% 7540
mean(X.Elapsed)%17.9841
mean(X.Distkm)% 2.1370
 %%
    X=T(T.BarArr==7,:);%Les Corts
length(X.Elapsed)%12779
mean(X.Elapsed)%15.4764
mean(X.Distkm)% 1.7479
 %%
    X=T(T.BarArr==8,:);%Sant Marti
length(X.Elapsed)%66466
mean(X.Elapsed)%16.4202
mean(X.Distkm)% 1.8876
 %%
    X=T(T.BarArr==9,:);%Nou Barris
length(X.Elapsed)%3573
mean(X.Elapsed)%16.9096
mean(X.Distkm)% 1.5281
 %%
    X=T(T.BarArr==10,:);%San Andreu
length(X.Elapsed)%16765
mean(X.Elapsed)%17.3267
mean(X.Distkm)% 1.6846
```

```
%%
C = unique(T.BikeID);
length(C)
%%
C = unique(T.CustID);
length(C)
%%
%CALCULATE TOP 10
y=unique(T.removal_station); %find unique elements of vector
N=histcounts(T.removal_station,y); %calc frequency of appeareance
S=sort(N,'descend'); %sort by freq
S=S(1:10)
for i=1:10
   topRem(i)=y(find(N==S(i)));    %show top 10
end
topRem
%%
y=unique(T.arrival_station);
N=histcounts(T.arrival_station,y);
S=sort(N,'descend');
S=S(1:10)
for i=1:10
   topArr(i)=y(find(N==S(i)));
end
topArr
%%
y=unique(T.BarArr);
N=histcounts(T.BarArr,11);
S=sort(N,'descend')
for i=1:10
   topArr(i)=y(find(N==S(i)));
end
topArr
%%
%https://www.mathworks.com/matlabcentral/newsreader/view_thread/94653
y=unique(T.BarRem);
N=histcounts(T.BarRem,11);
S=sort(N,'descend')
for i=1:10
   topRem(i)=y(find(N==S(i)));
end
topRem
%%
   X=T(T.BarRem==5,:);
y=unique(X.BarArr);
N=histcounts(X.BarArr,10);
S=sort(N,'descend')
   for i=1:length(S)
   topArrb(i)=y(find(N==S(i)));
   end
topArrb
```

```
%%
%TOP 10 neighborhoods
for j=1:10
   X=T(T.BarRem==j,:);
y=unique(X.BarArr);
N=histcounts(X.BarArr,10);
S=sort(N,'descend')
   for i=1:length(S)
   topArr(j,i)=y(find(N==S(i)))
   end
end
topArr
%%
% BIKE STATS
y=unique(T.BikeID);
for i=1:length(y)
 X=T(T.BikeID==y(i),:);
 usos(i)=length(X.Elapsed);
 sumela(i)=sum(X.Elapsed);
 sumdis(i)=sum(X.Distkm);
end
meanela=mean(sumela)% 1.2392e+03 mean bike elapsed time in 1 week
meandis=mean(sumdis)% 133.8159 mean distance 1 bike in 1 week
meanlen=mean(usos)%77.2077 mean usages 1 bike 1 week

%%
%Customer stats
y=unique(T.CustID);
for i=1:length(y)
 X=T(T.CustID==y(i),:);
 usos(i)=length(X.Elapsed);
 sumela(i)=sum(X.Elapsed);
 sumdis(i)=sum(X.Distkm);
end
meanela=mean(sumela)% 189.9463
meandis=mean(sumdis)% 20.5116
meanlen=mean(usos)%11.8346
%%
%FACTOR ANALYSIS
load ('barriosFA.csv') % .csv with table of neighbourhood destinations
X=[CVE,EIX,SMJ,HTG,GRA,STG,COR,SMT,NOU,SAN]; %load neighborhoods into a table
[wcoeff,score,latent,tsquared,explained] = pca(X) % pca function
pareto(explained) %plot pca
xlabel('Principal Component')
ylabel('Variance Explained (%)')
%%
%Anova analysis
[p,table,stats] = anova1(T.Elapsed,T.DiaRem)
multcompare(stats)
%%
[p,table,stats] = anovan(T.Elapsed,{T.DiaRem,T.BarRem})
```

Cluster Analysis

```
load ('coordsv3.mat') %Coordinatess of stations in latitude and longitude
load ('barriosFA.csv')
X=[CVE,EIX,SMJ,HTG,GRA,STG,COR,SMT,NOU,SAN];
 %Load neighborhoods into a table
%%
%representation of stations in a map
gscatter(lon,lat,bar2,'rgbymck','osxd'), axis([2.1 2.25 41.34 41.46])
%%
plot(lon,lat,'o')
%%
X=[lat lon bar2]
%%
%Dictionary function translate 10 neighborhods into k number of members in the cluster
  [idx,C] = kmeans(X,5);

for i=1:length(bar2)
   for j=1:10
      if bar2(i)==j
      bar3(i,1)=idx(j);
      end
   end
end
 %plot cluster in map
 gscatter(lon,lat,bar3,'rgbymck','osxd'), axis([2.1 2.25 41.34 41.46])

 title '95%';
 %%
[silh2,h] = silhouette(X,idx);
```

Translation functions

```
load ('coordsv3.mat') %Coordinates of stations in latitude and longitude
load ('cleandata4.csv') %Full dataset
% Assignates stations existing on the dataset to neighborhoods existing on the coordinates .csv,
0 mean deleted stations.
for i=1:342339

   if find(stat==removal_station(i)) & find(stat==arrival_station(i))
  barrem(i,:)=bar2(find(stat==removal_station(i)),:);
  bararr(i,:)=bar2(find(stat==arrival_station(i)),:);

   else
            barrem(i,:)=0;
      bararr(i,:)=0;

   end
end
```

```
barrem2=mat2cell(barrem,342339,3);
bararr2=mat2cell(bararr,342339,3);
```

Visualization

```
load ('coordsv3.mat')
ds=datastore('cleandata4.csv')

ds.VariableNames={'BikeID','CustID','removal_slot','arrival_slot','removal_station','arrival_statio
n','BarRem','BarArr','DiaRem','DiaArr','HoraRemHHMM','HoraArrHHMM','Elapsed','Distkm'}

%ds.SelectedVariableNames =
{'CustID','Distkm','removal_station','arrival_station','DiaRem','HoraRemHHMM','HoraArrHHM
M','Elapsed','BarRem','BarArr'};

preview(ds)
T = readall(ds);
%%
%Histograms
histogram(T.Distkm),title('Distance in Km'),axis([0.1 6 0 10000])
%%
histogram(T.Distkm),title('Distance in Km'),axis([0.1 6 0 10000])
%%
subplot(121),histogram(T.BarRem);title('Barrio removal'),axis([0 11 0 120000])
subplot(122),histogram(T.BarArr);title('Barrio arrival'),axis([0 11 0 120000])
%%
subplot(121),histogram(T.arrival_station); title('arrival station'),axis([0 450 0 9000]),hold on
subplot(122),histogram(T.removal_station); title('removal station'),axis([0 450 0 9000]),hold on

subplot(121),histogram(T.arrival_station(T.BarArr == 2)); title('arrival station in
Eixample'),axis([0 450 0 9000])
subplot(122),histogram(T.removal_station(T.BarRem == 2)); title('removal station in
Eixample'),axis([0 450 0 9000])
%%
histogram(histelapsed),title('elapsed time'),axis([0 60 0 70000])
%%
histogram(histelapsed),axis([0 120 0 70000])
%%
subplot(121),histogram(T.arrival_station);hold on
subplot(122),histogram(T.removal_station);

subplot(121),histogram(T.arrival_station(T.BarArr == 2));
subplot(122),histogram(T.removal_station(T.BarRem == 2));
%%
histogram(T.removal_station); hold on
%%
histogram(T.BarRem),title('Removal(Blue) vs Arrival(Red)'),axis([0 11 0 120000]),hold on
histogram(T.BarArr),axis([0 11 0 120000])
%%
subplot(121),histogram(T.arrival_slot),title('arrival slot')
```

```
subplot(122),histogram(T.removal_slot),title('removal slot')
%%
histogram(bar2),title('Number of stations')
%%
%BOXPLOTS
figure
boxplot([histhoraarr,histhorarem]),hold on
figure
boxplot(histelapsed),hold on
%%
subplot(121),boxplot(T.Elapsed),axis([0.5 1.5 -1 35]),title('Time Elapsed')
subplot(122),boxplot(T.Distkm),axis([0.5 1.5 -0.1 5]),title('Dist in Km')
%%
subplot(121),boxplot(T.Elapsed,T.BarRem),axis([0.5 11.5 -1 40]),title('Time Elapsed by
Neighborhood of Removal')
subplot(122),boxplot(T.Distkm,T.BarRem),axis([1.5 11.5 -0.1 5]),title('Dist Km by Neighborhood
of Removal')
%%
subplot(121),boxplot(T.Elapsed,T.DiaRem),axis([0.5 7.5 -1 40]),title('Time Elapsed by day')
subplot(122),boxplot(T.Distkm,T.DiaRem),axis([0.5 7.5 -0.1 5]),title('Dist Km by day')
%%
boxplot(T.Elapsed,T.HoraRemHHMM),axis([0.5 2401 -1 40]),title('Time Elapsed by hour')
%%
boxplot(T.BarArr,T.BarRem),axis([1.5 11.5 0 11])
%%
figure
boxplot(histelapsed),hold on

%%
%SCATTERPLOTS
gscatter(T.HoraRemHHMM,T.Elapsed,T.BarRem,'rgbymck','oxsd'), axis([0 2401 0 60])
%%
gscatter(T.Distkm,T.Elapsed,T.DiaRem,'rgbymck','oxsd'), axis([0 6 0 60])
xlabel 'Distance(Km)';
ylabel 'Time Elapsed(Min)';
%%
gscatter(T.HoraRemHHMM,T.Elapsed,T.BarRem,'rgbymck','oxsd'), axis([0 2401 0 60])
xlabel 'Removal Hour(HHMM)';
ylabel 'Time Elapsed(Min)';
%%
gscatter(lon,lat,bar,'rgbymck','oxsd'), axis([2.1 2.25 41.34 41.46])
```

Creation of dates and cleaning

```
%Data loaded is from original .xlsx through matlab gui
%SLOW METHOD
separa = regexp(arrival_date,' ','split'); % crea cell que contiene en cada celda un 1x2 cell

fecha_arr = cellfun(@(x) x{1}, separa, 'uni', 0);%separa el cell interno en 2 distintos
hora_arr = cellfun(@(x) x{2}, separa, 'uni', 0);
separafecha = regexp(fecha_arr,'-','split');
dia_arr = cellfun(@(x) x{3}, separafecha, 'uni', 0);
```

```
separa = regexp(removal_date,' ','split'); % crea cell que contiene en cada celda un 1x2 cell
fecha_rem = cellfun(@(x) x{1}, separa, 'uni', 0);%separa el cell interno en 2 distintos
hora_rem = cellfun(@(x) x{2}, separa, 'uni', 0);
separafecha = regexp(fecha_rem,'-','split');
dia_rem = cellfun(@(x) x{3}, separafecha, 'uni', 0);
%%
separahora = regexp(hora_rem,':','split');
hhrem = cellfun(@(x) x{1}, separahora, 'uni', 0);
mmrem = cellfun(@(x) x{2}, separahora, 'uni', 0);
%%
hhrem=cell2mat(hhrem);
mmrem=cell2mat(mmrem);
%%
hhmm=horzcat(hhrem,mmrem);
hhmm=cell2mat(hhmm);
hhmm=str2num(hhmm);
%%
%FAST METHOD
arrivald=datevec(arrival_date,'yyyy-mm-dd HH:MM:SS.FFF');
removald=datevec(removal_date,'yyyy-mm-dd HH:MM:SS.FFF');
%%
histhoraarr=horzcat(arrivald(:,4),arrivald(:,5),arrivald(:,6));
histhorarem=horzcat(removald(:,4),removald(:,5),removald(:,6));
%%
DiaArr=datestr(arrivald,'dd');
DiaRem=datestr(removald,'dd');
%%
DiaArr=str2num(DiaArr);
DiaRem=str2num(DiaRem);
%%
histhoraarr=datestr(arrivald,'HHMM');
histhorarem=datestr(removald,'HHMM');

histhoraarr=str2num(histhoraarr);
histhorarem=str2num(histhorarem);
%%
%http://stackoverflow.com/questions/16990762/matlab-how-to-calculate-the-number-of-
seconds-between-two-date-strings
%Creation of time elapsed for histogram usage GOOD METHOD
for i = 1:342339
t1 = datevec(hora_rem{i},'HH:MM:SS.FFF');
t2 = datevec(hora_arr{i},'HH:MM:SS.FFF');
Dt = etime(t2,t1);
elapsed=datestr(Dt/86400,'HHMM');
histelapsed(i,:)=str2num(elapsed);
end
%%
%creating time elsaped SLOW METHOD
asd{i,1}=mat2cell(elapsed,1);
kek=cell2mat(asd{i,1});
```

```
expression = ['(\d+):(\d+)'];
[tokens,matches] = regexp(kek,expression,'tokens','match');
hora=cellfun(@(x) x{1},tokens, 'uni', 0);
min=cellfun(@(x) x{2},tokens, 'uni', 0);
hora=cell2mat(hora);
min=cell2mat(min);
elapsed=horzcat(hora,min)
%%
cell2mat(asd);
expression = ['(\d+):(\d+)'];
[tokens,matches] = regexp(kek,expression,'tokens','match');
hora=cellfun(@(x) x{1},tokens, 'uni', 0);
min=cellfun(@(x) x{2},tokens, 'uni', 0);
hora=cell2mat(hora);
min=cell2mat(min);
elapsed=horzcat(hora,min)
```

User following

```
%Following of user or bike through a map
load ('coordsv3.mat')
ds=datastore('cleandata4.csv')

ds.VariableNames={'BikeID','CustID','removal_slot','arrival_slot','removal_station','arrival_statio
n','BarRem','BarArr','DiaRem','DiaArr','HoraRemHHMM','HoraArrHHMM','Elapsed','Distkm'}

ds.SelectedVariableNames =
{'BikeID','CustID','Distkm','removal_station','arrival_station','HoraRemHHMM','HoraArrHHM
M','Elapsed'};

preview(ds)
T=readall(ds);
%%

%%
var=find(T.BikeID==21695) % random bike
dat= T(var,[4,5])
X=[lat,lon];
dat=table2array(dat)
%%
var=find(T.CustID==364180)% 365340 %364180 %313420 69196 16281
dat= T(var,[4,5]) % removal stations and arrival station
X=[lat,lon];
dat=table2array(dat)
%%
 %Plot points of stations of a user or bike into the map and draw lines to join them
for i=1:size(lols)
   if find(stat==lols(i,1)) & find(stat==lols(i,2))
  coordsrem(i,:)=X(find(stat==lols(i,1)),:)
```

```
  coordsarr(i,:)=X(find(stat==lols(i,2)),:)
   end
end

for i=1:size(lols)
   if find(stat==lols(i,1)) & find(stat==lols(i,2))
  barrem(i,:)=bar(find(stat==lols(i,1)),:)
  bararr(i,:)=bar(find(stat==lols(i,2)),:)
   end
end

latr=coordsrem(:,1)
lonr=coordsrem(:,2)
lata=coordsarr(:,1)
lona=coordsarr(:,2)
rem=[lonr,lona]
arr=[latr,lata]
Y=[lonr;lona]
X=[latr;lata]
barrr=[bararr;barrem]
gscatter(Y,X,barrr,'rgbymck','osxd'), axis([2.1 2.25 41.34 41.46]),hold on
for i=1:length(rem)
line(rem(i,:),arr(i,:)),hold on
text(rem(i,1),arr(i,1),num2str(i))
kk=text(rem(i,2),arr(i,2),num2str(i))
kk.Color='r';
end

gscatter(lon,lat,bar,'rgbymck','osxd'), axis([2.1 2.25 41.34 41.46])
```

Creation of Neighborhood variables and distance

```
ds=datastore('cleandata4.csv')

ds.VariableNames={'BikeID','CustID','removal_slot','arrival_slot','removal_station','arrival_statio
n','BarRem','BarArr','DiaRem','DiaArr','HoraRemHHMM','HoraArrHHMM','Elapsed','Distkm'}
preview(ds)
T = readall(ds);
%%

X=[lat,lon];

%%
%calculate distance between stations
X=[lat,lon];
for i=1:length(removal_station)
  X1=X(find(stat==removal_station(i)),:)
  X2=X(find(stat==arrival_station(i)),:)

  distancia(i,:)= distance(X1,X2);
```

```
end
distkm=deg2km(distancia);

%%
%Creation of distance variable in dataset
for i=1:length(removal_station)
   if find(stat==removal_station(i)) & find(stat==arrival_station(i))
  coordsrem(i,:)=X(find(stat==removal_station(i)),:);
  coordsarr(i,:)=X(find(stat==arrival_station(i)),:);
distancia(i,:)=distance(coordsrem(i,1),coordsrem(i,2),coordsarr(i,1),coordsarr(i,2));
   else
      distancia(i,:)=0;
   end
end
distkm=deg2km(distancia);
%%
%Creation of neigbhoorhood variable in the dataset
X=[lat,lon];
for i=1:length(removal_station)
   if find(stat==removal_station(i)) & find(stat==arrival_station(i))
  barrem(i,:)=bar(find(stat==removal_station(i)),:);
  bararr(i,:)=bar(find(stat==arrival_station(i)),:);
   else
            barrem(i,:)=cellstr('NaN');;
      bararr(i,:)=cellstr('NaN');

   end
end
```

Linear regression

```
ds=datastore('cleandata4.csv')

ds.VariableNames={'BikeID','CustID','removal_slot','arrival_slot','removal_station','arrival_statio
n','BarRem','BarArr','DiaRem','DiaArr','HoraRemHHMM','HoraArrHHMM','Elapsed','Distkm'}

%ds.SelectedVariableNames = {'CustID','Distkm','removal_station','arrival_station','Elapsed'};
ds.SelectedVariableNames = {'removal_station','arrival_station','Distkm','Elapsed'};

mdl = fitlm(T, 'Elapsed ~ 1 + Distkm','Intercept',false)

%%
plot(mdl),axis([0 12 0 60])
%%
plotSlice(mdl)
%%
Y=predict(mdl,2.5);
```