# Airbnb rentals in Barcelona

## PMAAD - GIA

**Professors:**
Karina Gibert
Dante Conti
Sergi Ramírez

**Students:**
Àlex Miquel Casanovas Cordero.
Daniel Cejas Vázquez.
Gerard Gómez Izquierdo.
Víctor Molina Díez.
Pol Rion Solé.

**Course:**
Q4 Primavera 2022-2023

# Index

# 1. The Data

Our Database contains information about all the apartments, houses and private rooms located in Barcelona which have an appearance in the Airbnb website. As it is specified in the *Metadata file* that we propose for the D1 delivery, there is information about: the characteristics and specifications of the "structure" of the house, different scores given by previous clients and personal information about the host.

## 1.1 Links to the Databases

Databases extracted from: http://insideairbnb.com/get-the-data

This is the link where we've taken the AirBnB Barcelona Database from. If you 'click' in this link, and go to the Barcelona - Spain section, you'll find the original csv file of our Database. The names of the Databases used are: *listings.csv*, *reviews.csv*. We will use two different Databases: the first one, containing all the numerical and categorical variables, will be used to make clusters, linear models…; the second one contains the textual and time-referenced data necessary for the Time Series and textual analysis part of this project.

## 1.2. How did we obtain the data

Each member of the group has searched individually for a dataset that accomplishes the necessary requirements to develop the practical work of the subject. After long sessions of searching this type of dataset and not succeeding, we ended up deciding we would investigate the same theme of the previous quadrimester, using the same dataset but treating it the way this course tells us. In this way, at the end, we will be able to compare the results obtained last quadrimester with those obtained in this one. We also made this decision due to recommendations made by the teachers, who mentioned that the aspect of the comparison would be interesting.

The theme of our practical work will be the Airbnb rentals in Barcelona, and the website from which we extracted the dataset is the same as the one from which we did last time.

For the additional dataset we found one with a temporal register of all the reviews made on Airbnb rentals in Barcelona. The website offers open datasets, so we only had to click in a link to download the dataset.

## 1.3 Database Structure

In this section we will specify the basic structure of our dataset, mentioning basic aspects of it such

as the number of rows, variables and the types of variables we are going to use. Moreover, we will show a table with the details of the missing values in each one of them.

To start, it's important to remark that our dataset has 15778 rows and 75 variables. At this moment of the work, where we present the characteristics of the dataset, we are going to provide information about each variable. Maybe in the future, when we start with the development of the work at all, some variables will be eliminated by reasons of redundancy or lack of information provided by them. This will be specified in the future when the decisions are made.

Globally, our dataset has 1183350 cells, from which 1124148 are not missing values and 59202 are missing values. For that reason, the percentage of missing values represents approximately 5% of the dataset. We will work with a dataset that contains 36 character variables, 23 integer variables, 2 logical variables and 14 numeric variables.

In the following table we can see information about the missing values in each variable in more detail:

| VARIABLE | NOMBRE DE DADES MANCANTS | PERCENTATGE DE DADES MANCANTS |
|---|---|---|
| id | 0 | 0% |
| listing_url | 0 | 0% |
| scrape_id | 0 | 0% |
| last_scraped | 0 | 0% |
| source | 0 | 0% |
| name | 0 | 0% |
| description | 0 | 0% |
| neighborhood_overview | 0 | 0% |
| host_id | 0 | 0% |
| host_url | 0 | 0% |
| host_name | 0 | 0% |

| | | |
|---|---|---|
| host_since | 0 | 0% |
| host_location | 0 | 0% |
| host_about | 0 | 0% |
| host_response_time | 0 | 0% |
| host_response_rate | 0 | 0% |
| host_acceptance_rate | 0 | 0% |
| host_is_superhost | 0 | 0% |
| host_thumbnail_url | 0 | 0% |
| host_picture_url | 0 | 0% |

| | | |
|---|---|---|
| host_neighbourhood | 0 | 0% |
| host_listing_count | 2 | 0,013% |
| host_total_listing_count | 2 | 0,013% |
| host_verifications | 0 | 0% |
| host_has_profile_pic | 0 | 0% |

| | | |
|---|---|---|
| host_identity_verified | 0 | 0% |
| neighbourhood | 0 | 0% |
| neighbourhood_cleansed | 0 | 0% |
| neighbourhood_group_cleansed | 0 | 0% |
| latitude | 0 | 0% |
| longitude | 0 | 0% |
| property_type | 0 | 0% |
| room_type | 0 | 0% |
| accomodates | 0 | 0% |
| bathrooms | 15778 | 100% |
| bedrooms | 581 | 3,682% |
| beds | 274 | 1,736% |
| amenities | 0 | 0% |
| price | 0 | 0% |
| minimum_nights | 0 | 0% |
| maximum_nights | 0 | 0% |
| minimum_minimum_nights | 1 | 0,006% |

| | | |
|---|---|---|
| maximum_minimum_nights | 1 | 0,006% |
| minimum_maximum_nights | 1 | 0,006% |
| maximum_maximum_nights | 1 | 0,006% |
| minimum_nights_avg_ntm | 1 | 0,006% |
| maximum_nights_avg_ntm | 1 | 0,006% |
| calendar_updated | 15778 | 100% |
| has_availability | 0 | 0% |
| availability_30 | 0 | 0% |
| availability_90 | 0 | 0% |
| availability_180 | 0 | 0% |
| availability_365 | 0 | 0% |
| calendar_last_scraped | 0 | 0% |
| number_of_reviews | 0 | 0% |
| number_of_reviews_ltm | 0 | 0% |
| number_of_reviews_l30d | 0 | 0% |
| first_review | 0 | 0% |

| | | |
|---|---|---|
| last_review | 0 | 0% |
| review_scores_rating | 3277 | 20.77% |
| review_scores_accuracy | 3370 | 21.36% |
| review_scores_cleanliness | 3369 | 33.69% |

| | | |
|---|---|---|
| review_scores_checkin | 3374 | 21.38% |
| review_scores_communication | 3368 | 21.35% |
| review_scores_location | 3373 | 21.38% |
| review_scores_value | 3373 | 21.38% |
| license | 0 | 0% |
| instant_bookable | 0 | 0% |
| calculated_host_listing_count | 0 | 0% |
| calculated_host_listing_count_entired_rooms | 0 | 0% |
| calculated_host_listing_count_privated_rooms | 0 | 0% |
| calculated_host_listing_count_shared_rooms | 0 | 0% |
| reviews_per_month | 3277 | 20.77% |

# 2. Preprocessing

The first step after getting the data is to preprocess it. In our case we need to do multiple things in order to get our data ready to make the prediction models. In our data we have more rows than desired and also a lot of variables (75) so we will need to apply dimensionality reduction techniques to reduce the complexity of our data and try to eliminate all the noisy variables.

## 2.1 Dimensionality Reduction.

First of all we did a random sample to get 5000 rows from the original 15778 rows. We needed to reduce the quantity of observations to adequate the data to the project requirements. Furthermore we decided to do this sample random to get our data distributions the most similar to the one in the original dataset. Another option was applying a filter, for example getting only the rows whose neighborhood is l'Eixample. But we thought that by doing this kind of filter we could lose interesting factors to analyze, so we ended up doing a random sample.

### 2.1.1 Manual Feature Selection

To start with the dimensionality reduction we erased all the features that didn't give any relevant information. For example the id of the property or the id of the host, that only act as identifiers, or the url of the property. Those are the variables that we erased:

- id
- listing_url
- scrape_id
- last_scraped
- picture_url
- host_id
- host_url
- host_thumbnail_url
- host_picture_url
- calendar_updated
- calendar_last_scraped
- license
- host_listings_count
- host_total_listings_count
- bathrooms

Apart from the identifiers, there are some features that we also decided to erase:

**Last_scraped and Calendar_last_scraped:** Shows the date the data was collected/updated. As it only includes a range of a day or two, it's not useful as temporal data.

**Calendar_updated and Bathrooms:** Those two features were completely null. In the case of bathroom we have another column that gives us that information but in text format. We will do feature engineering to get the numerical data from this column.

**Host_listings_count and Host_total_listings_count:** Those variables give information about how many properties the host has in total in Airbnb. This variable is important, but the number is the sum of all the properties around the world, so if the host has a property in Madrid or London, the number will appear here. We erased that feature because we have another variable that gives the same information but only of the properties the host has in Barcelona, so we consider it is better for our project.

After this first step of feature selection we erased 16 features. Now we have 60 features left to analyze.

### 2.1.3 Feature Engineering:

Before beginning with the Feature Selection we did a transformation to some variables in order to make them more useful.

#### Amenities:

This feature is not really informative right now. It describes all the amenities that the rental has. The problem is that as there are a lot of amenities it has so many modalities, because the combinations of amenities an apartment can have is very huge. For that reason we created a new feature that indicates the number of amenities. Doing this we can get a more informative feature and eliminate the old one.

#### host_location:

For this feature we had to rename the modalities because it had a lot of different locations. We made the decision to classify the locations in 3 modalities: "in_cat", "out_sp", "in _sp_out_cat".

#### Bathrooms_txt:

We had to adapt the format of this feature, because the number of bathrooms was expressed as a text, for example: "1 bathroom" so we had to remove the text and show the result as an integer.

#### Gender:

Since we started to work with this data we wanted to add some kind of gender perspective to our data, to see how gender may affect the different variables on our database (price, listing_count, etc.).
We don't have the "gender" feature in our data. So how are we supposed to know the gender of the host of an apartment? The answer was the name of the host. We have this feature, and we thought that we could try to extract the gender of the host with a machine learning model. In that way we enhance

the relevance of AI in our project, connecting this subject with the knowledge we are gaining in the degree.

Once we decided that, we searched for a ready-made model to extract the gender from a name. And we found a company[1] that does that, and its free plan was of 5000 names/month, so it fitted perfectly our database.

The model has over 8 million names processed and 22 alphabets supported. Furthermore, it returns not only the predicted gender but a calculated probability of how sure it is it's the correct gender.

Once it predicted our names we did a rapid check and found out that there were not only person names, but also company names and compound hosts like "Anna & Mario". So we needed a manual preprocessing of the names first, because although a lot of companies got a relatively low hit rate, there were some hotel names like: "Sant Jordi Hostels" that were predicted as a male with a very high probability of success.

For that reason we created 2 more modalities: Companies and Unisex. To select the companies, we did an exhaustive search in the database looking for keywords like: "Hotel", "Hostel","Apartment","Barcelona",etc. The Unisex modality was for those names that are predicted with less than 80% success rate, here we have names that are the same for male and female or compound host names as mentioned above. After doing all this preprocessing we ended up with a new feature that explains the gender of the host. This feature has the following frequency distribution:

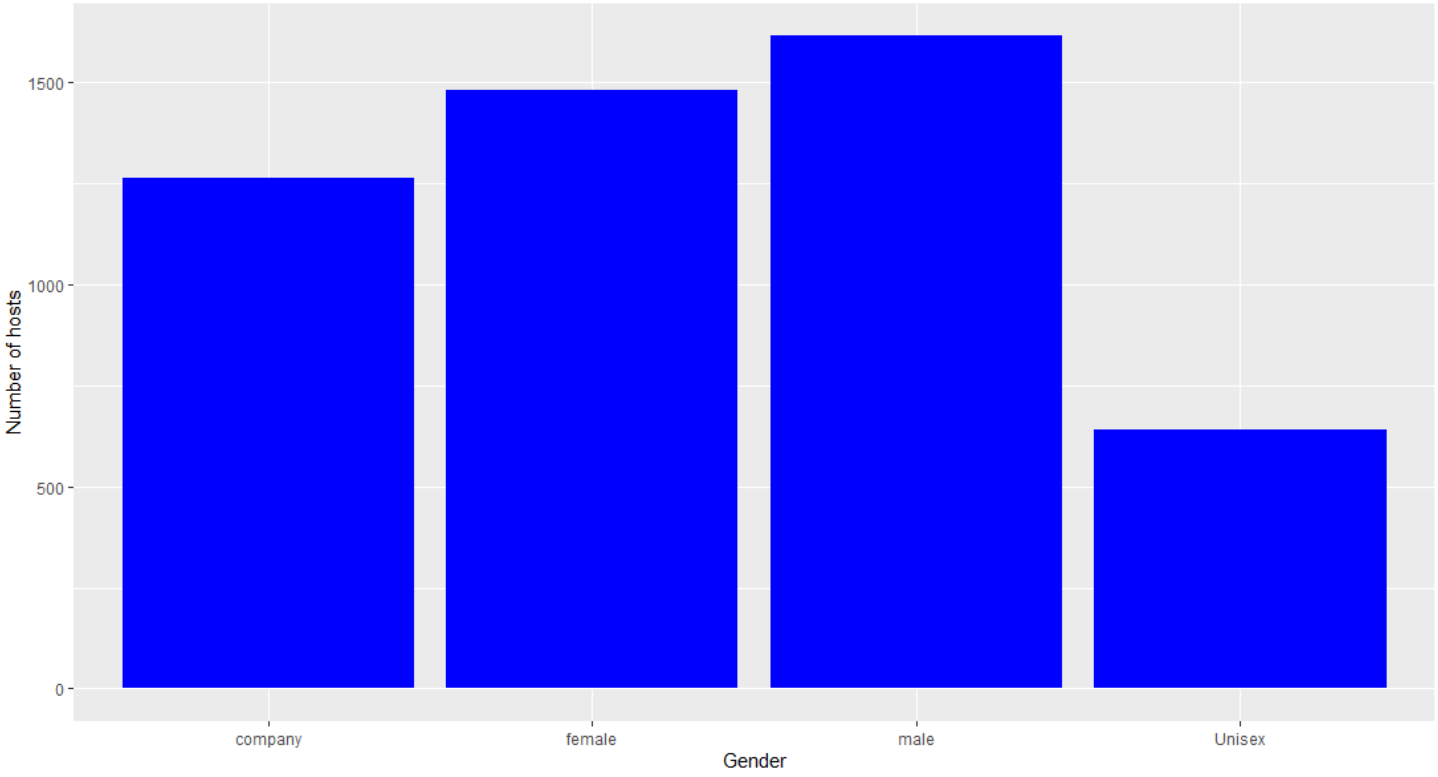



Figure 1: Gender feature countplot

[1] The company is GenderGuesser and its web is: https://gender-guesser.com/

As we can see the number of males and females is very close but we still have more male hosts than female. The number of companies is high too, which makes sense because each row is a different listing and a company may offer different properties or apartments. For the last, we have approximately 600 unisex hosts, which include multiple hosts that may be a man and a woman, or names that works for both genders or maybe non-binary names.

### 2.1.3 Feature Selection: Filter Methods

**Numerical Features:**

The next step was going one step further using a correlation matrix to make a selection of numerical variables. However we also tried to make a PCA to check the relations between the features. Unfortunately, the information inertia in the first and second dimension was not even 30% as we can see in the following graph.

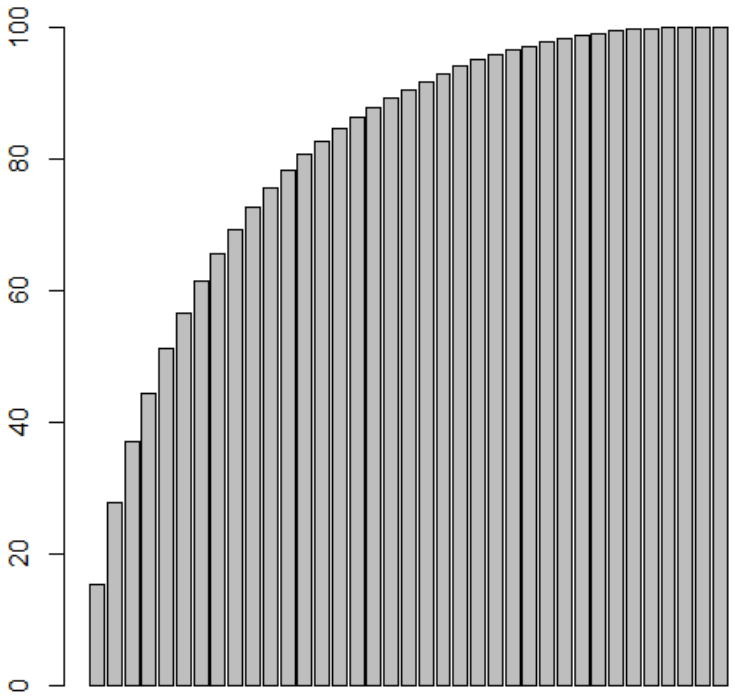



Figure 2: Cumulative information inertia

So with these bad results, we preferred to base all the feature selection in the correlation matrix.

After checking the correlation matrix we found different groups of variables that are very correlated among them:

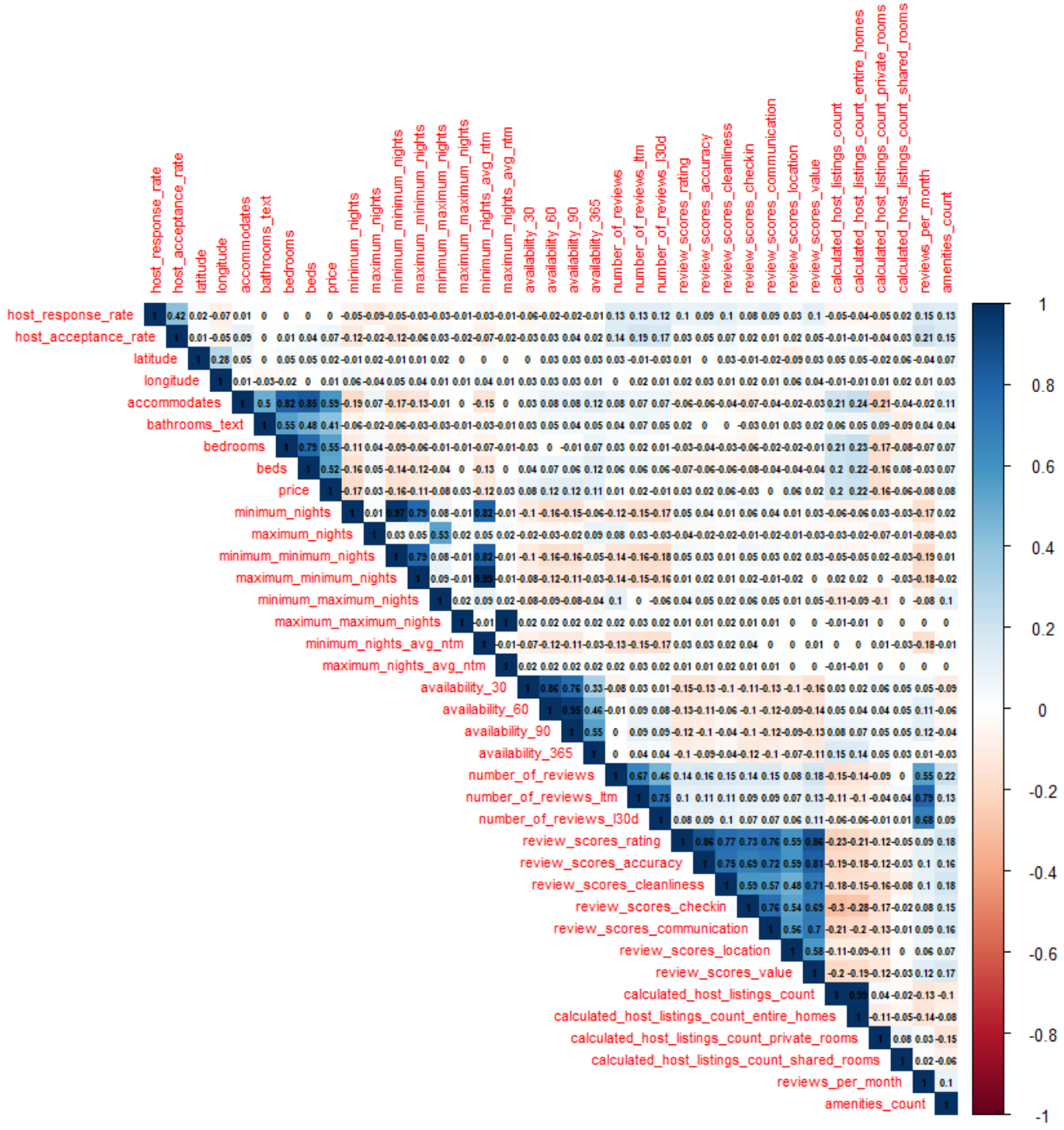



Figure 3: Correlation matrix

As we can see in the correlation matrix, there are several groups of variables very correlated. Now we will analyze each of these groups to select the features that are not important for the project.

**Accommodates, Bedrooms and Beds:**

These three features are highly correlated, with a correlation coefficient of 0.83 and 0.87 (accommodates with beds). These values really make sense because as the number of bedrooms and beds increase, the capacity of the house increases too. Now we have to choose which variable will stay in the database. If we keep the bedrooms, there can be a property with a lot of bedrooms and one bed per bedroom or few bedrooms with 2 or more beds in it. If we keep the beds, it can be that two houses have the same number of beds but one has only individual beds, and the other has all king size beds. For that reason we decided to keep the accommodates variable, because it represents better the requirements of the property.

We could also talk about the bathroom_text variable. In this case, we did not eliminate it because the correlation was not that high, only 0,5 correlation, so it still gives us new information.

**Minimum_nights:**

This group of features is also very correlated, and it gives very similar or redundant information. This happens because a property can have different minimum night values. For example, in summer the minimum nights can be 3 nights, and the rest of the year 7 nights. For this reason we decided to use only minimum_nights_avg, to have the mean of minimum nights of the listing along the year, to not be conditioned by the time the data was collected.

**Maximum_nights:**

With this group of variables the case should be the same as the minimum_nights. However, if we look at the correlation matrix, the values are not correlated at all, which didn't seem normal to us. So we thought about looking for outliers in this variables that could make those variables not correlated among them.

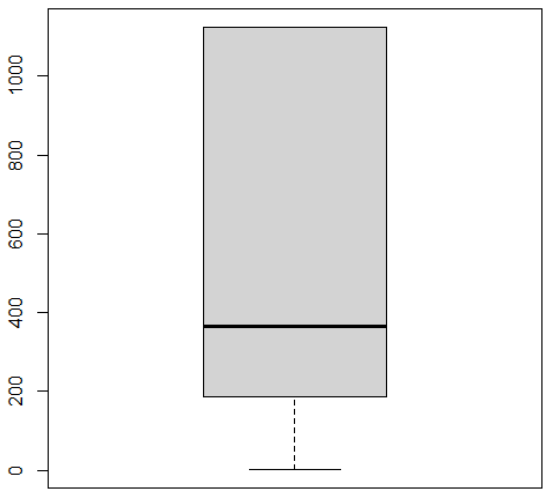



Figure 4: Maximum_nights

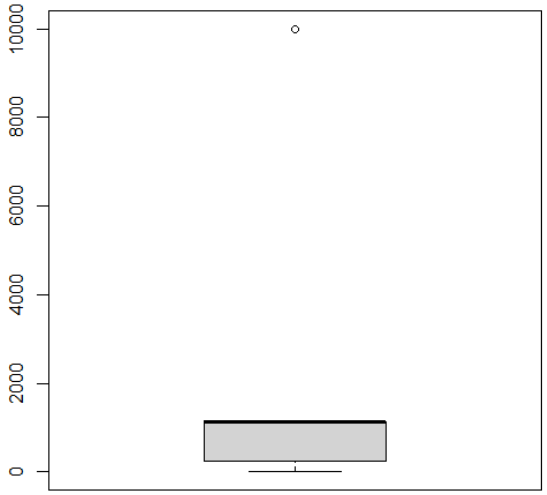



Figure 5: Maximum_minimum_nights

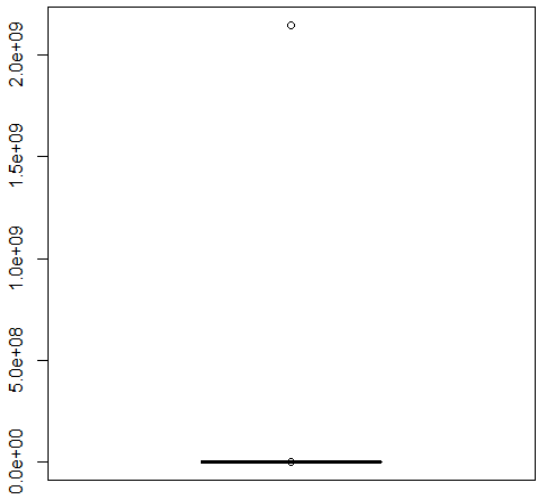



Figure 6: Maximum_maximum_nights

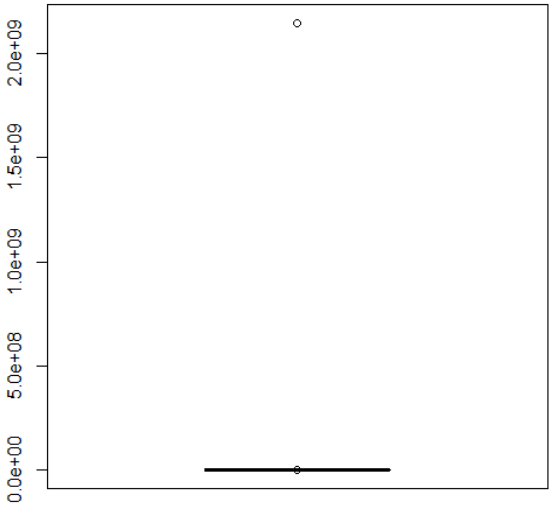



Figure 7: Maximum_nights_avg_ntm

As we can see there are outliers in all the variables except the “maximum_nights”. And some of them are so big that they may alter the correlation between those variables. Furthermore, if we look at the outliers, they must come from an error, because the values are so gigantic that they make no sense, or a value of 9999, which usually means “missing value”. For those reasons we will make these values NA and we will treat them as this in the missing values imputation section (2.2).

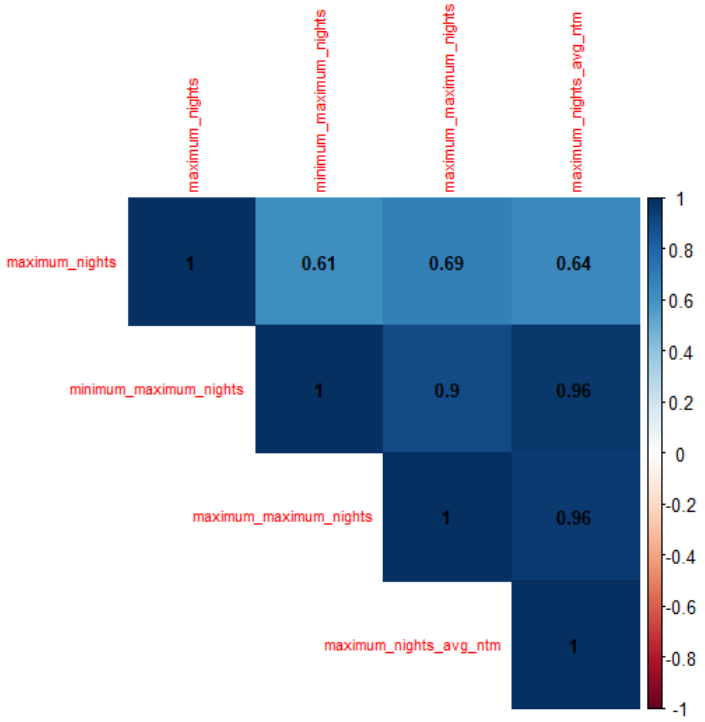



Figure 8: Correlation Matrix (Maximum_nights)

As we can see now, those features are highly correlated. So we will proceed to erase all except the one that has the average values, as we did with the minimum night variable.

**Availability:**

Those variables are also highly correlated, after all, they all show the same information in different time intervals. We decided to keep the availability_365, because it gives us more information, because we can see if an apartment is highly demanded or maybe if it only is available a part of the year. With the others we could only see 30/60/90 days forward, and depending on the date the data was collected we could get wrong information and extract wrong conclusions in consequence.

**Number_of_reviews:**

As with the other groups of features, in this group almost all correlations were high. The only correlation that was a bit low was between "number of reviews" and "number of reviews last 30 days", which makes sense because the last 30 days do not represent the total reviews because maybe that month the apartment was not available or it was a bad month. For those reason we again keep the feature with the average values, to maintain all the maximum information and because this variable was highly correlated with the other three.

**Reviews_score:**

For the "reviews score" the conclusion is very similar. We keep the "reviews_score_rating" because it's the mean of the other scores, so it is highly correlated with all the others. Doing this we can get the global rating of the listing with only one feature.

**Calculated_host_listings_count:**

If we look at the correlation matrix this group has no correlations except for the one between "calculated_host_listings_count" and "calculated_host_listings_count_entire_homes". Our first hypothesis was that almost all the listings are entire_homes, so we plotted the categorical variable that shows that information:

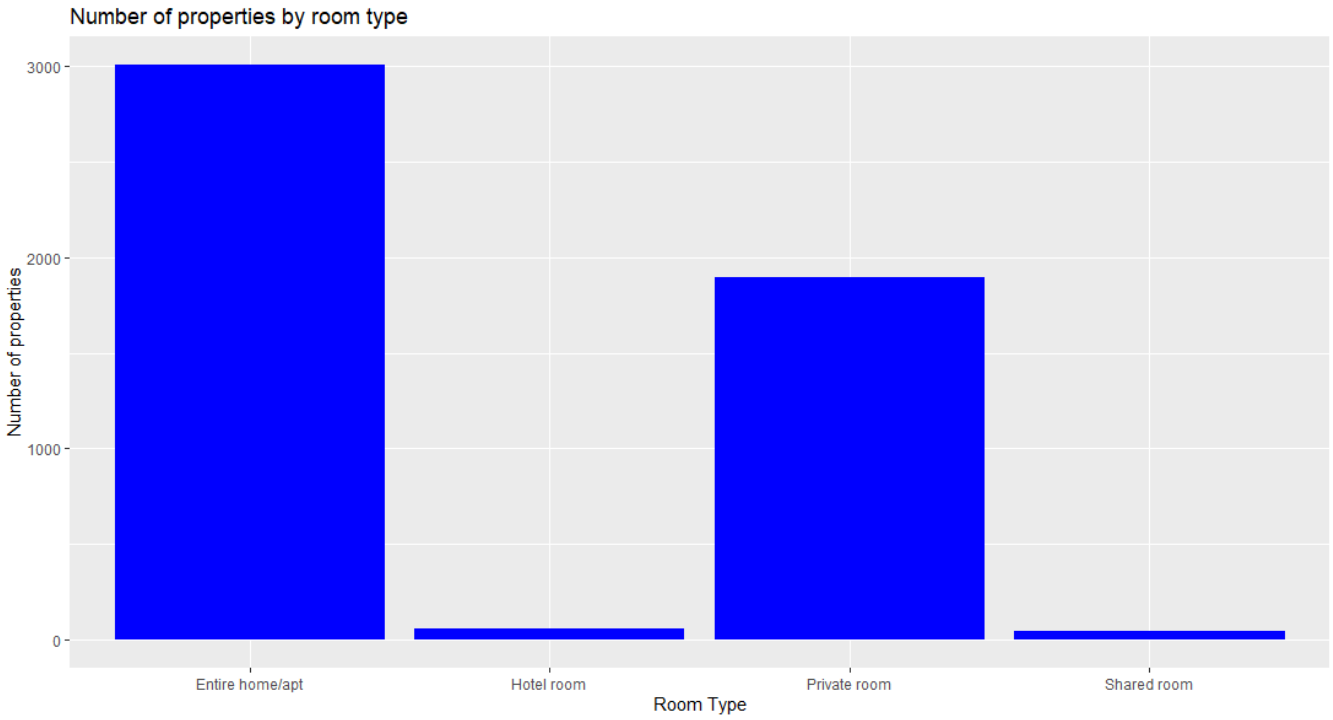



Figure 9: Room_type frequency plot

As we can see, it is true that entire_homes are the most freqüent modality, but private_room has also a very high frequency. After seeing that we decided to look at the means of both features:

```
> mean(dd$calculated_host_listings_count)
[1] 22.6016
> mean(dd$calculated_host_listings_count_entire_homes)
[1] 19.453
> mean(dd$calculated_host_listings_count_private_rooms)
[1] 2.9618
```

Now we see that the mean is much higher in the entire_homes one, and much more near to the mean of the total count, that's why it is really correlated. The reason behind these numbers is that those features count how many properties a host has, so every time that a property has the same host, the value of those features is the same. For example, if a host has 110 entire_homes and 3 private_rooms, the 110 will appear 110 times so it will increase the mean.

After seeing that we can confirm that the features are related between them, so now we can keep only the one that is the sum of the others.

**Final result:**
After doing all these changes we reduced the numerical variables from 37 to 14, which is a great reduction. Here is the correlation matrix after the elimination of those features:

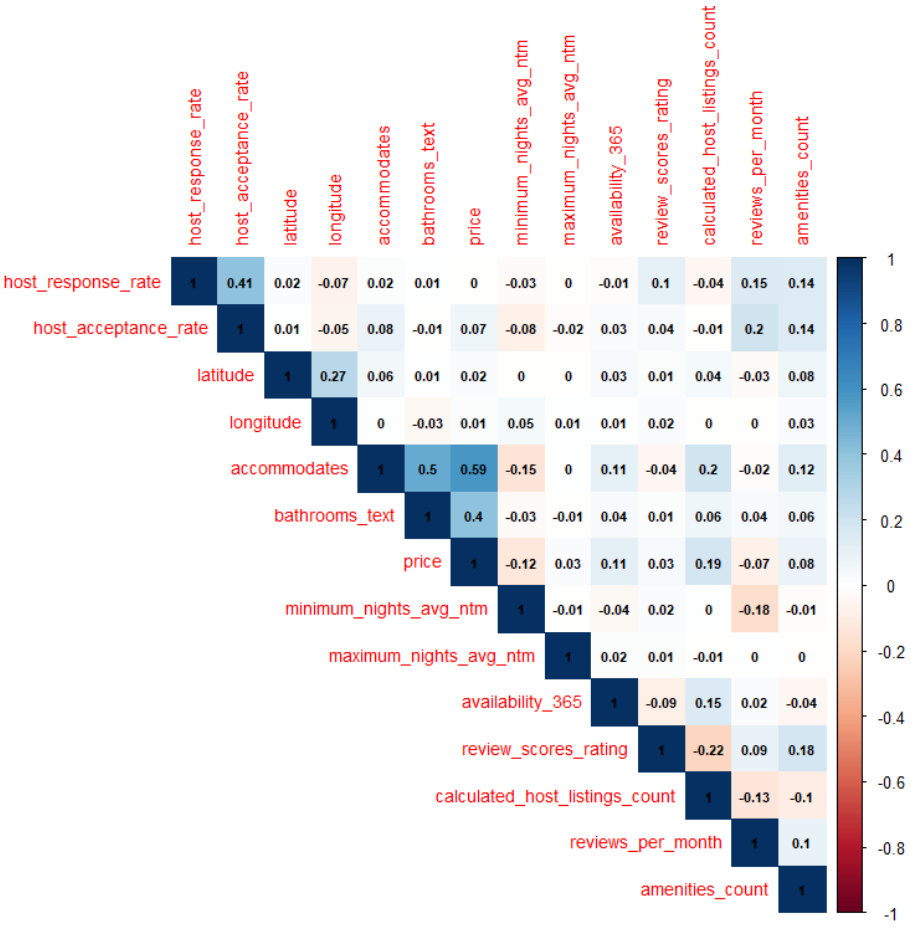



Figure 10: New Correlation Matrix

**Categorical Features:**

Once we did the feature selection for numerical features, we need to do the same for the categorical ones. In this case we have much less features, exactly 9 categorical and 5 binary features.

To extract the relationships between all those variables we did an MCA to look at the most important features to represent our data and to see which ones are so related that we can keep only one. Apart from this, we will also look at the meaning of the features themselves, because there are features that may not seem related because the number of modalities is too high but in fact they are related. The next plot is the plot of the first two dimensions of the MCA where we can extract some patterns:

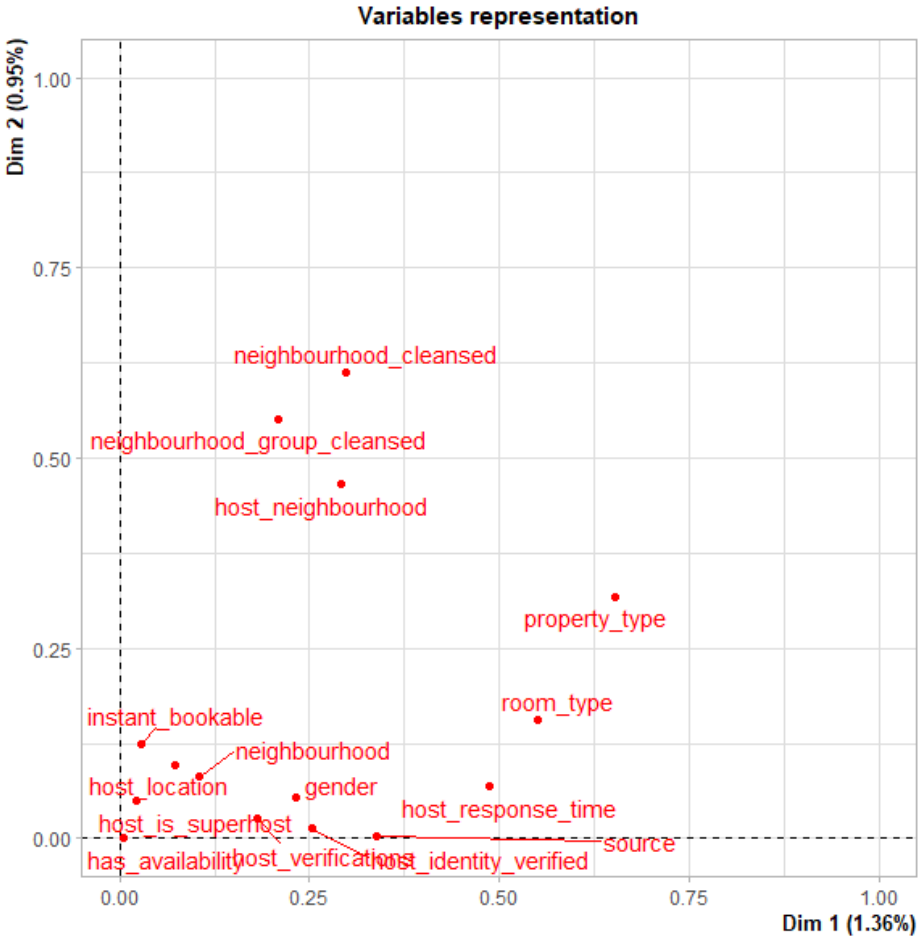



Figure 11: MCA Dim1/Dim2 Plot

If we analyze the plot we can try to identify the meaning of each dimension.

**Dim1:** If we look at the most important features in this dimension we see that room_type, property_type and host_response_time have a big impact on the meaning of the dimension. For that reason this dimension may represent the **characteristics of the listing or the host.**

**Dim2:** In this dimension we have all the information related with the **location** either from the host or the listing.

Now that we know that we can proceed to analyze which features should we erase. Starting from the first dimension we have that property_type and room_type are really related. That's because property_type is a more specific variable that tells information about the property. For that reason we will erase this feature, because it has a lot of modalities that can be resumed in the ones in room_type. We will keep the other important features of this dimension ("host_response_time", "host_identity_verified", "source", "gender","host_verifications" and "host_is_superhost"). We decided to keep these features because it gives us great information about the host behavior and reliance, also we can see that the gender feature that we created has an impact in this dimension.

With the other dimension we can see that the features that tell us about the neighbourhood the listing is in have a lot of impact. We also can see that the neighbourhood of the host has a big impact, but now we will go into that. On the one hand, we have 3 features that are really related: "neighbourhood_cleansed","neighbourhood_group_cleansed", as it's shown in the plot, but also "neighbourhood". Those three features represent the same information but at different degrees of precision. That is why the neighbourhood doesn't seem to be related with the others, because it is way more specific than the other two. For those reasons we will keep only the "neighbourhood_group_cleansed" because it is the more general feature, and has less modalities which will help in posterior analysis.

The feature "host_neighbourhood" is really related with the neighborhood_group of the listing because most of the hosts are from Catalonia, specifically from Barcelona, so the neighbour may coincide with the neighbour of the listing. For this reason, we will erase this variable and keep host_location, which gives more general information about the host and has less modalities.

To end with these feature selection, we will also erase the feature "has_availability" because it has almost 0 variance in those dimensions above all, because we have a numerical variable that gives us more information about availability.

### 2.1.4 Feature and Modality Rename

To end with this first step of preprocessing, we renamed the names of the features and the modalities of the categorical features. We made that because some names were really large, and for a further analysis that may cause overlapping names and difficult the visualization of some plots.

## 2.2 Missings Imputation

After we decided which variables of the database were going to be used in our practical work, we started the analysis of the missings that it presented.

First of all, we replaced the missings represented with an empty string with NA, so in this way they can be detected by the program. After that, thinking of the plots we wanted to show of the missings found, we simplified the names of the variables to make them more stylish and understandable.

Once we had the dataset prepared, we initated the missings analysis. To realize this analysis, we considered the missings per variable and the missings per row. It's important to know how to treat the missings found, considering at the same time the variable where it belongs and the individual from who it is. Given a missing, if it belongs to an individual with several other missings, this individual

may be erased from the dataset due to lack of information. This type of missings probably are not worth to impute because, in that way we would work with a high quantity of synthetic data which may not correspond to reality. Otherwise, if this missing doesn't belong to a row in that conditions, it could be interesting to study its imputation to complete and augmentate the information given by a variable. In this case, we also should consider that this missing doesn't belong to a variable with several other missings, which would have a similar effect as the incomplete row of what we wrote before.

Knowing our goals and warning concepts, we started computing a counter for missings per variable. The results obtained are the following ones:

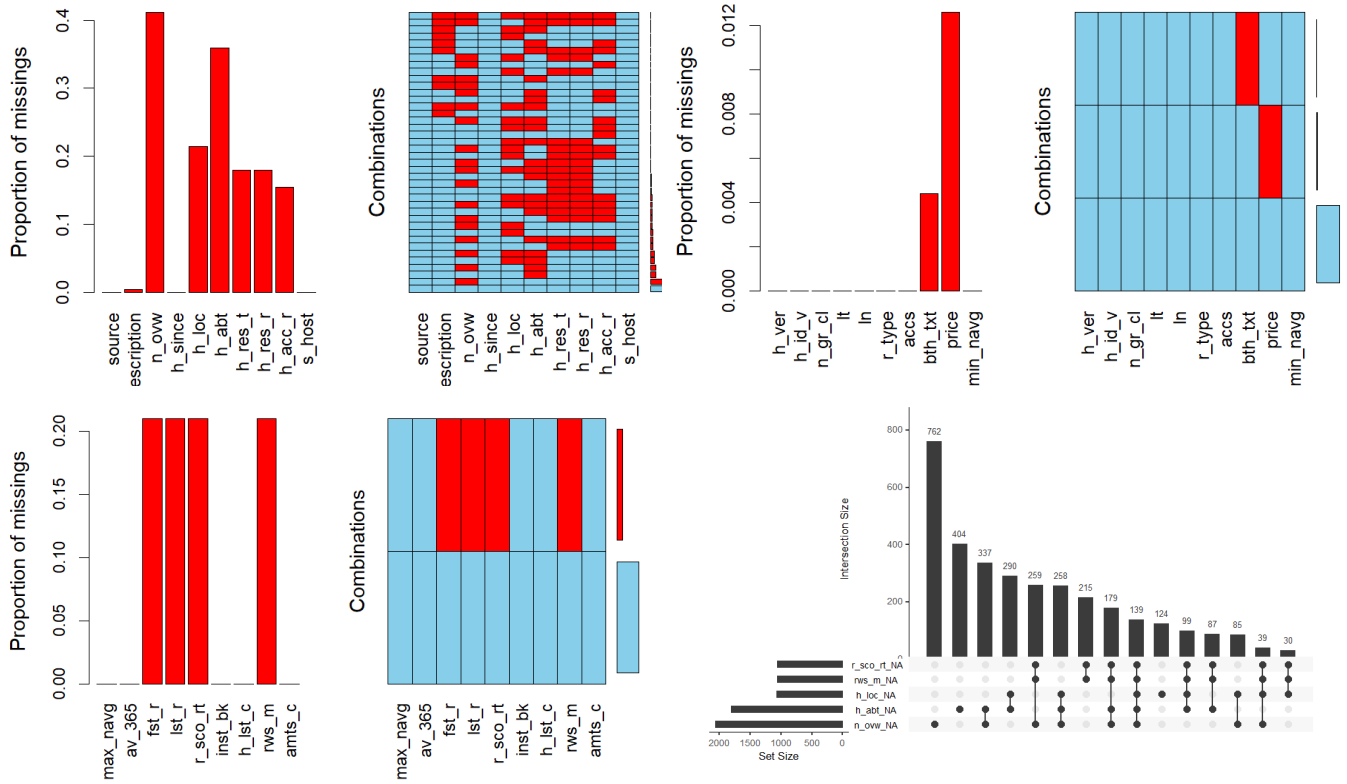



Figure 12: Missings distribution on the dataset

As we can see in the previous plots, we are working with some problematic variables, because they contain a high proportion of missing values. We can highlight *n_ovw*, *h_loc*, *h_abt*, *fst_r*, *lst_r*, *r_sco_r*, *rws_m* among others. To see in a numerical way the pronounced presence of missing in this variables compared with others, we counted them in each variable. The results obtained are the following ones.

Missings per variable:

| Variable | Count |
|---|---|
| source | 0 |

| | |
|---|---|
| description | 21 |
| n_ovw | 2060 |
| h_since | 0 |
| h_loc | 1069 |
| h_abt | 1798 |
| h_res_t | 896 |
| h_res_r | 896 |
| h_acc_r | 772 |
| s_host | 0 |
| h_ver | 0 |
| h_id_v | 0 |
| n_gr_cl | 0 |
| lt | 0 |
| ln | 0 |
| r_type | 0 |
| accs | 0 |
| bth_txt | 22 |
| price | 63 |
| min_navg | 0 |
| max_navg | 0 |
| av_365 | 0 |
| fst_r | 1051 |
| lst_r | 1051 |
| r_sco_rt | 1051 |
| inst_bk | 0 |
| h_lst_c | 0 |
| rws_m | 1051 |

| | |
|---|---|
| amts_c | 0 |

To continue with the analysis we computed a counter of missings per row. We added an extra column to the database which indicated the number of missings observed in the corresponding row. This counter helped us to observe if there existed individuals with a low relevance and information contribution in our database. It's important to note that, to make this calculus, we considered those NAs that belong to all variables except those which are date or text characterized. We had to establish a threshold that indicated to us if the row was relevant enough or not. We decided to use 6 missings as the limit to accept a row. After the calculus, we obtained 4 rows with 7 missings, so they were eliminated. When the work had been done, the extra column that counted NAs was eliminated to continue with the analysis.

After that, we had to deal with the rest of the missing values, trying to identify from which type they were. To start with this identification, we decided to make a MCAR test on the numerical variables to see if the missing values that they presented were MCAR classifiable. Before this test was made, we had to make some modifications in the variable *max_navg* in order to avoid the influence of outliers in the test. Once we made the test, we obtained a p value of $0 < 0.05$, so we can say that the missing weren't MCAR classifiable. Under these circumstances, we had to study our dataset and try to identify patterns visually. Appreciating the individuals with missing values, we saw that the individuals that had a missing value on *first_review* had missing values also in the other variables related with reviews. Probably these individuals were the apartments that started recently with their business and didn't have any review registered yet. These missings showed a non random pattern, so they were MNAR type. Due to this, we decided to remove the individuals that presented them.

Before starting with the imputation of missing values in the numerical variables, we treated the missings in the categorical ones. The text columns contained lots of missing values, but we didn't need to treat them because these columns weren't going to be used to perform models. The missings of the other categorical variables, which represented classes or categories, were substituted with a new category named "unknown".

At this point, there were only numerical variables with missing values, specifically *h_res_r*,*h_acc_r*,*bth_txt* and *price* so we needed to start with its imputation. To go ahead with this phase we had two options: MIMMI imputation or MICE imputation. We wanted to choose the best imputation option, so we decided to make the two imputations, each one in an independent way of the other, with the intention of comparing the results obtained later and choosing the most appropriated method.

To start, we imputed the missing data with MIMMI. As MIMMI works with clusters to identify the values with which it has to impute a determined empty cell of the dataset, we had to choose the number of clusters that we wanted to work with. Appreciating the dendrogram we can see below, we chose 6 clusters ($k = 6$).

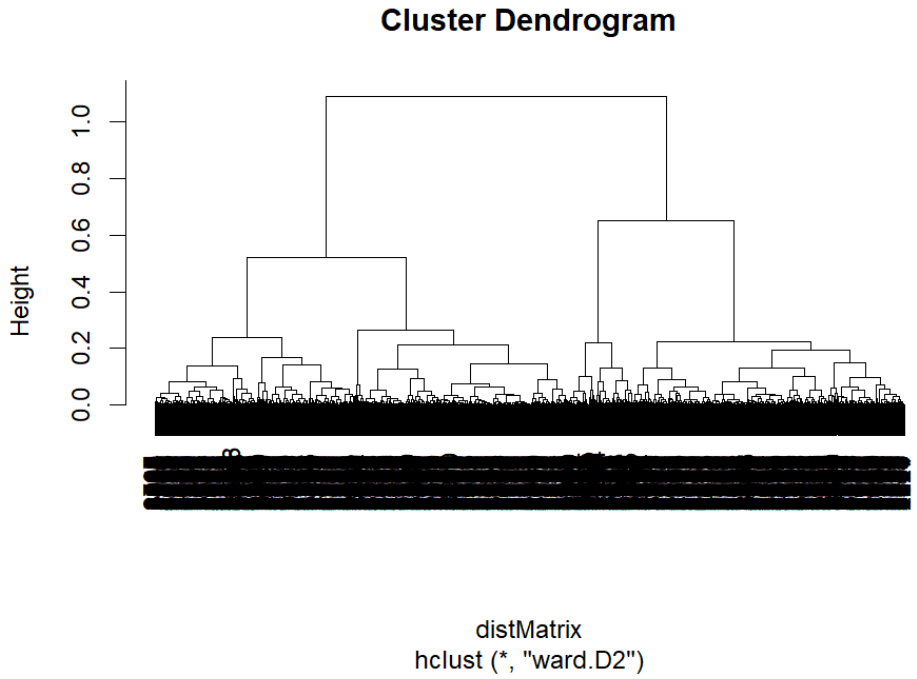



Figure13: Cluster Dendrogram

Later, we could see the values with which we imputed the missing information of each individual, depending on which cluster it was classified. These values were the following ones:

| | h_res_r | h_acc_r | bth_txt | price |
|---|---|---|---|---|
| c1 | 100 | 100 | 1 | 50 |
| c2 | 100 | 100 | 1 | 50 |
| c3 | 100 | 100 | 1 | 90 |
| c4 | 97 | 99 | 1 | 92 |
| c5 | 100 | 100 | 1 | 120 |
| c6 | 100 | 100 | 1 | 50 |

On the other hand, and in an independent way of the previous imputation, we imputed the missings with MICE. For each iteration of the algorithm we impute the missing values of each variable, for all variables, and this process is done ***m*** times, trying to reach a moment where the imputed values don't change. The typical used values for ***m*** are ***m=5*** or ***m=10***. These operations are repeated in each iteration, so assuming that for each iteration we construct a different set of imputed values, at the end of the execution we have as different sets as the number of iterations we did. To finish with the imputation, these different sets are combined by average or votation technique to obtain a unique estimation for each missing value.

Here we can appreciate a stripplot that lets us better understand the mice algorithm, showing the evolution of the imputation of the variable *price* made in an iteration. We can see that we imputed it ***m=5*** times. The red points represent the imputed values, and we can see that in each imputation they decrease their changeness.

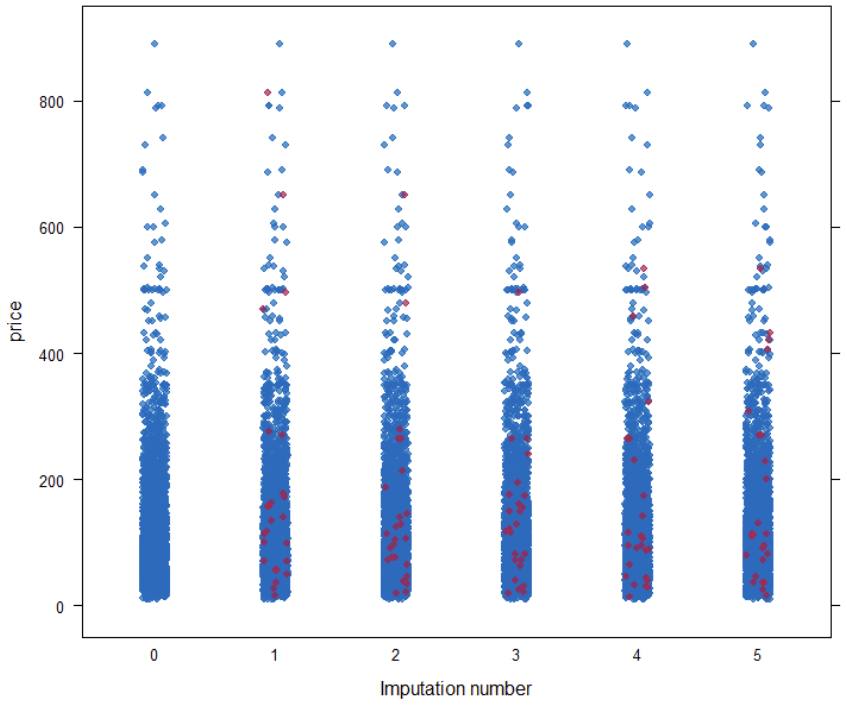



Figure14: Missing imputation by MICE

When we had the two imputations done, we proceeded to compare them using plots to show the variables imputed. The following boxplots allow to see the different phases of imputation:

1. The variable with missings
2. The variable after removing the MNAR missings
3. The variable after imputing with MIMMI or MICE

To decide which imputation method is better, we compared the different variables obtained in phase 3 (MIMMI-imputed and MICE-imputed) with the variable in phase 1 (the variable before any imputation). Visually, we tried to find significant differences in the plot of their information. The criteria to decide the appropriateness of a method was the similarity that the resulting imputed variable showed with the variable before imputing. The less are the differences, the less we are distorting our variable.

The boxplots of the different phases for each variable to impute are the following ones:

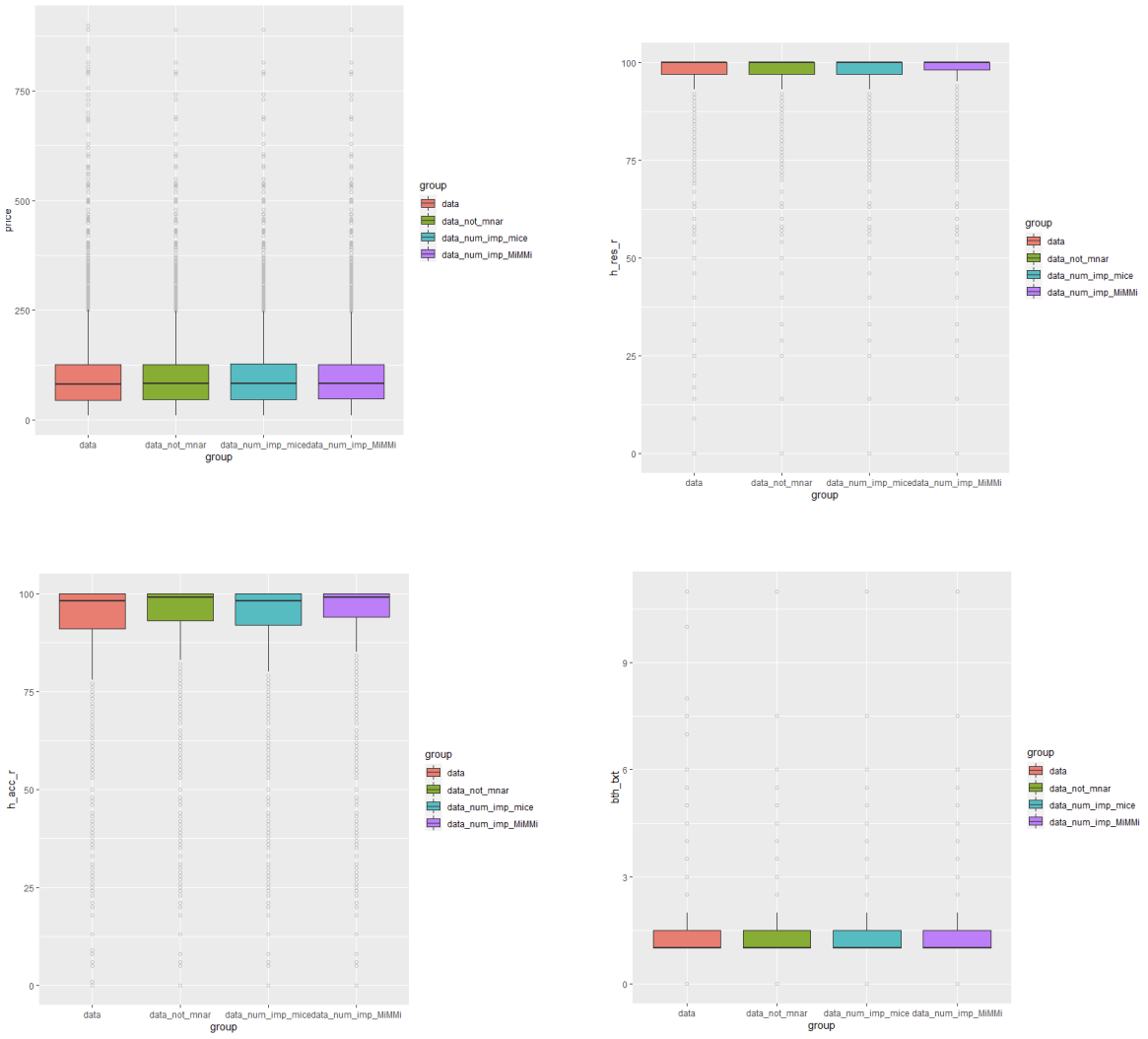



Figure 15: Boxplots of the variable distribution in the different phases of imputation, for each variable to impute

As we can see in the previous graphs, the imputation method that most dissimilarity presents with the original information is MIMMI, so MICE, a priori, is a more confident method to impute.

To corroborate our conclusions, we decided to make a Kolmogorov Smirnov test. To interpret and understand the results of the test we must take into account two values:

- **D:** indicates the distance between the imputed variable and the variable with missings. The lower the distance is, the more similar they are.
- **P-value:** if it is lower than 0.05, we can affirm that the imputed variable and the original one present different distributions, else we can't affirm it.

The results obtained for each variable are the next ones:

**<u>Price</u>**

*MICE*

```
> ks.test(data$price,data_num_imp_mice$price)

        Two-sample Kolmogorov-Smirnov test

data:  data$price and data_num_imp_mice$price
D = 0.018578, p-value = 0.4355
alternative hypothesis: two-sided
```

*MIMMI*

```
> ks.test(data$price,data_num_imp_MiMMi$price)

        Two-sample Kolmogorov-Smirnov test

data:  data$price and data_num_imp_MiMMi$price
D = 0.016297, p-value = 0.6051
alternative hypothesis: two-sided
```

In this case, for the variable price, we can see that the best imputing option is MIMMI because it presents a lower D and a higher p-value in comparison with MICE imputation.

***H_res_r***

*MICE*

```
> ks.test(data$h_res_r,data_num_imp_mice$h_res_r)

        Two-sample Kolmogorov-Smirnov test

data:  data$h_res_r and data_num_imp_mice$h_res_r
D = 0.026573, p-value = 0.1167
alternative hypothesis: two-sided
```

*MIMMI*

```
> ks.test(data$h_res_r,data_num_imp_MiMMi$h_res_r)

        Two-sample Kolmogorov-Smirnov test

data:  data$h_res_r and data_num_imp_MiMMi$h_res_r
D = 0.086138, p-value = 2.167e-13
alternative hypothesis: two-sided
```

For this variable, the best imputation is MICE due to it has a lower D and a higher p-value in comparison with the MIMMI imputation.

***H_acc_r***

*MICE*

```
> ks.test(data$h_acc_r,data_num_imp_mice$h_acc_r)

        Two-sample Kolmogorov-Smirnov test

data:  data$h_acc_r and data_num_imp_mice$h_acc_r
D = 0.023144, p-value = 0.2243
alternative hypothesis: two-sided
```

*MIMMI*

```
> ks.test(data$h_acc_r,data_num_imp_MiMMi$h_acc_r)

        Two-sample Kolmogorov-Smirnov test

data:  data$h_acc_r and data_num_imp_MiMMi$h_acc_r
D = 0.098846, p-value < 2.2e-16
alternative hypothesis: two-sided
```

In this case, the best imputation option is MICE again, because as the same case as before, the D shown is lower and the p-value shown is higher in comparison with the MIMMI imputation.

***Bth_txt***

*MICE*

```
> ks.test(data$bth_txt,data_num_imp_mice$bth_txt)

        Two-sample Kolmogorov-Smirnov test

data:  data$bth_txt and data_num_imp_mice$bth_txt
D = 0.0062886, p-value = 1
alternative hypothesis: two-sided
```

*MIMMI*

```
> ks.test(data$bth_txt,data_num_imp_MiMMi$bth_txt)

        Two-sample Kolmogorov-Smirnov test

data:  data$bth_txt and data_num_imp_MiMMi$bth_txt
D = 0.0065419, p-value = 1
alternative hypothesis: two-sided
```

In the last variable, we can see that the best imputation method is MICE again because the p-values are equal but the D is lower in MICE in comparison with MIMMI imputation.

In conclusion, collecting all the results obtained, we can affirm that the MICE imputation method is more appropriate for our dataset than MIMMI. It satisfies the requisites for more variables than the MIMMI does. For this reason, MICE was the imputation method chosen to continue with our practical work and finally conclude this part of missing treatment.

## 2.3 Outlier detection and treatment

### 2.3.1 Univariate Outlier detection and Previous Analysis

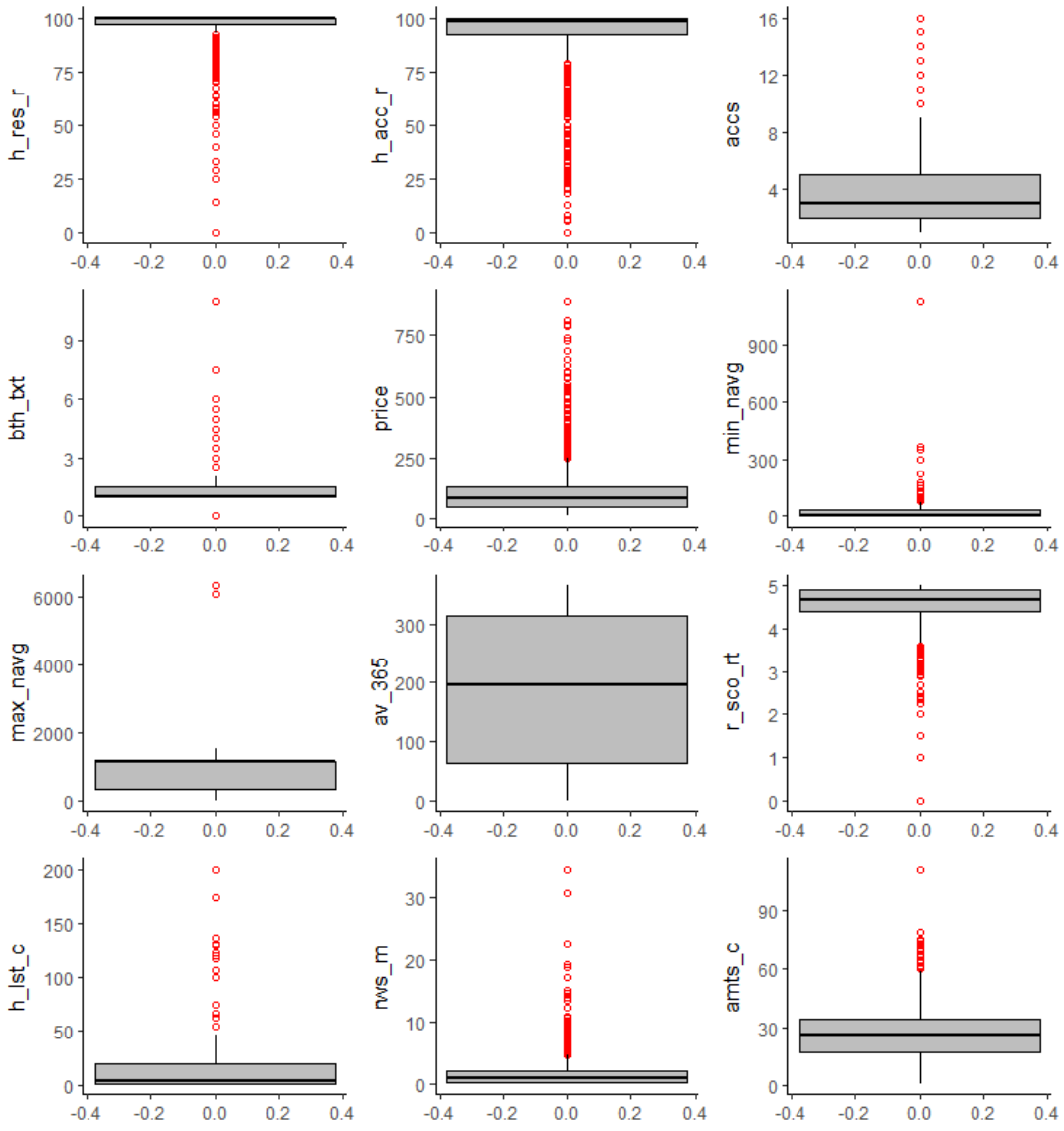



Figure 16: Boxplots from all numerical variables except latitude and longitude

Initial variables observations:

- The ‘Host Response Rate’ and ‘Host Acceptance Rate’ are typically high values, close to 1. However, there are some very low values, close to 0, that require further investigation to determine if they are outliers or not.

- In the "accommodates" and "baths" variables, we can observe that some apartments have unusual values. However, these apartments could be spacious enough to accommodate all the occupants. Therefore, these values are not considered outliers but rather uncommon.

- The ‘price’ variable range is quite extensive, and we must examine whether the most costly

apartments are simply expensive or if they represent incorrect values that could lead to erroneous predictions in our future models. In theory, these values do not appear to be outliers.

- The ‘minimum and maximum night average’ have 3 very large values that differ significantly from the mean. These values should be considered outliers, and we will need to study and treat those samples after the multivariate outlier detection.

- The review score rating variable is similar to the host acceptance rate, as it also has uncommon values that diverge from the mean. In some cases, the value is 0, and therefore we need to study these cases to determine whether it is a genuine value, a missing mark, or an outlier.

- The calculated count of host listings and reviews per month have unusual values, but they do not appear to be outliers as they are not significantly different from the mean.

- The count of amenities holds significant value, but it may be an outlier. Therefore, we need to examine this sample in the dataset to determine if the other variables correspond to an apartment with that number of amenities.

In conclusion, we cannot assume that all high or low uncommon values are outliers and need to be deleted. We will now study all the dataframe outliers using the Mahalanobis distance, a multivariate outlier detection technique, and treat them.

### 2.3.2 Multivariate Outlier detection

To study our data in depth we will perform a Multivariate Outlier detection to identify values that diverge massively from the common values. In addition to Univariate analysis we will take into account the interaction between variables.

**Mahalanobis distance technique**

The Mahalanobis distance is a statistical measure that calculates the distance between each sample and a distribution, taking into account the interaction between variables. Specifically the covariance between the variables of the distribution. The Mahalanobis distance between a point ‘x’ and a distribution with mean ‘m’ and covariance matrix ‘C’ is calculated as follows:

$$D^2 = (x - m)^T \cdot C^{-1} \cdot (x - m)$$

Figure 17: Mahalanobis distance formula

After calculating the Mahalanobis distance for the dataset this is the resulting plot:

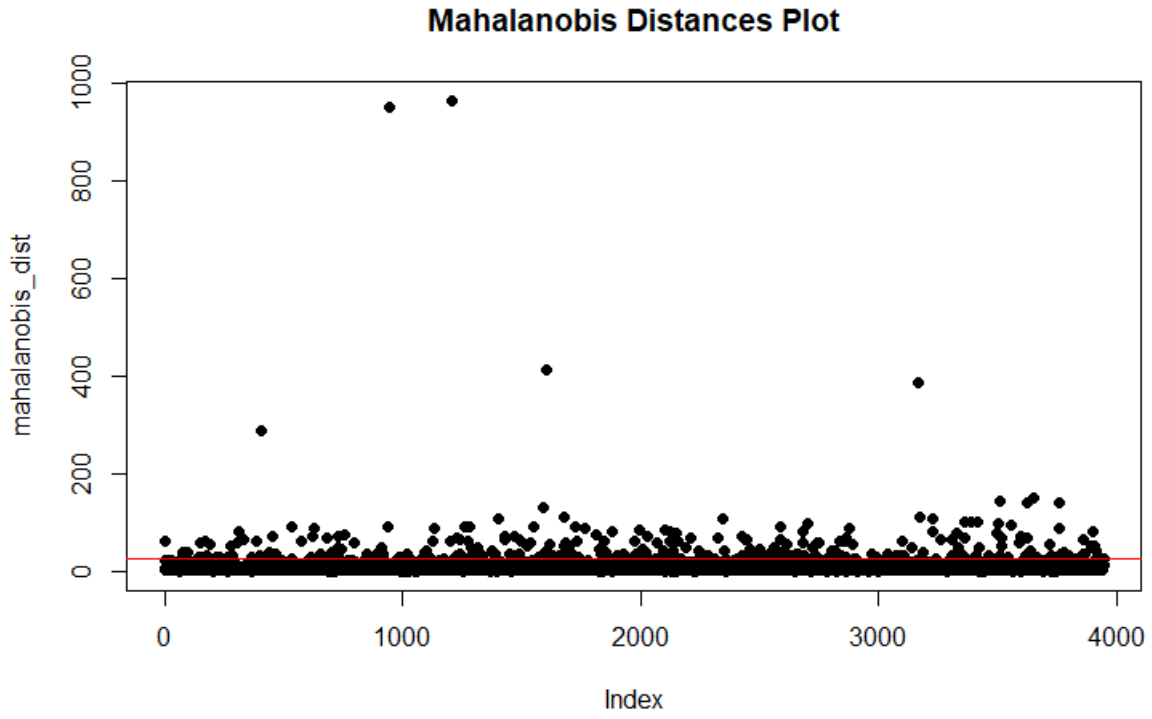



Figure 18: Plot of Mahalanobis distances with threshold set to 0.01 critical region of Chisq distribution

In this plot, we can see that each row representing the X-axis has a distance assigned to the Y-axis. As we can see, the majority of points have a low Mahalanobis distance, which means that the data point is not far away from the center of the distribution. Instead, there are some other values that have big values. This means that those data points are significantly different from the rest of the dataset, indicating that they could be outliers.

To determine which data points are outliers, we must establish a threshold that indicates at what Mahalanobis distance value data points are considered outliers. When dealing with a complex dataset, a common and effective approach is to assume the chi-squared distribution with the degrees of freedom equal to the number of columns of the dataset to determine this threshold. The concept is to assume that the Mahalanobis distance distribution follows the chi-squared distribution with ncol degrees of freedom and select a critical value of the chi-squared distribution to a certain level of significance. This value will act as a threshold, with commonly used values being 0.01, 0.025, or 0.05.

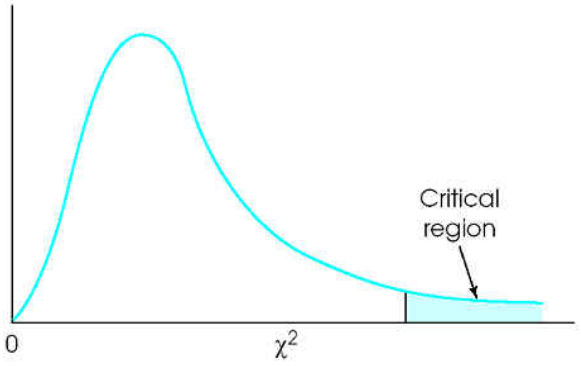



Figure 19: Chisq distribution with marked critical region

```
# Find the threshold for outlier detection using chi-square distribution
threshold <- qchisq(0.99, df = ncol(num_data))

# Identify outliers
outliers <- which(mahalanobis_dist > threshold)
```

Figure 20: Lines used in R to select data points with bigger value than our threshold:

In our case, we chose the value of 0.01. However, as shown in Figure 3, the red threshold line with 0.01 selects too many data points as outliers (286). This poses a problem for treating these outliers later. To address this issue, we manually increased the previous threshold, which was set based on the 0.01 critical value of the chisq distribution. The previous threshold was set at 26.22, but after examining the Mahalanobis distances plot and the number of outliers selected, we decided to set the threshold to a distance of 125. The plot below shows the new threshold.

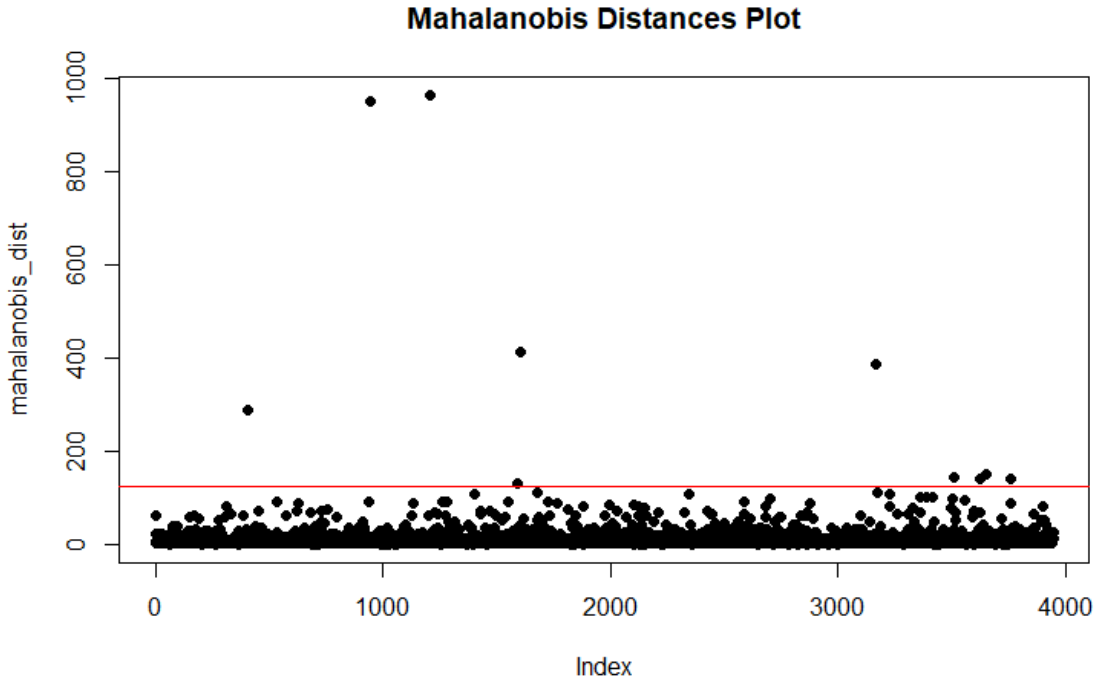



Figure 21: Mahalanobis distance plot with threshold set to 90

After updating the threshold, we have a total of 10 selected outliers. The next step is to treat them.

## 2.3.3 Outlier treatment

Once we have identified which samples are outliers, we need to determine which variables are causing the outliers to have a large Mahalanobis distance. One way to do this is to manually study the 10 samples that are considered outliers and mark the variables that are causing them to be outliers as NA. After selecting and marking the values as NA, we will perform a MICE imputation, as we did in the missing values preprocessing step.

To select and mark the values of the individuals with NA, we will take into account the boxplots from Figure 1. This will help us to see which are the uncommon values that are causing the outliers.

| | h_res_r | h_acc_r | accs | bth_txt | price | min_navg | max_navg | av_365 | r_sco_rt | h_lst_c | rws_m | amts_c |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **409** | 100 | 100 | 2 | 1.0 | 88 | 1.1 | 7.0 | 258 | 4.73 | 3 | 30.65 | 30 |
| **948** | 100 | 96 | 2 | 1.0 | 70 | 1125.0 | 1125.0 | 363 | 5.00 | 9 | 0.70 | 58 |
| **1206** | 79 | 100 | 1 | 1.0 | 106 | 1124.0 | 1125.0 | 0 | 4.55 | 1 | 1.31 | 28 |
| **1592** | 96 | 94 | 3 | 1.0 | 142 | 5.2 | 6079.4 | 306 | 4.64 | 6 | 2.45 | 36 |
| **1608** | 100 | 100 | 1 | 5.0 | 18 | 1.0 | 6.9 | 362 | 4.45 | 6 | 34.46 | 21 |
| **3165** | 70 | 64 | 1 | 11.0 | 32 | 32.0 | 180.0 | 364 | 3.00 | 120 | 0.16 | 22 |
| **3510** | 0 | 100 | 2 | 1.0 | 56 | 180.0 | 999.0 | 344 | 0.00 | 10 | 0.02 | 34 |
| **3628** | 100 | 91 | 4 | 1.0 | 136 | 5.2 | 6347.4 | 307 | 4.68 | 4 | 1.90 | 43 |
| **3653** | 100 | 100 | 2 | 1.0 | 106 | 1.1 | 1125.0 | 321 | 4.73 | 1 | 22.52 | 34 |
| **3762** | 100 | 0 | 10 | 7.5 | 500 | 5.0 | 31.0 | 75 | 4.00 | 1 | 0.02 | 14 |

Figure 22: Samples considered outliers

As we previously mentioned, we will examine in each sample which value is causing an individual to have a large Mahalanobis distance. By analyzing the table in Figure 7 and the boxplots in Figure 1, we can identify the outlier or outliers for each sample.

Samples:

- **409**: Value: 30.65 from reviews per month (rws_m) variable.
- **948**: Value: 1125.0 from minimum nights average (min_navg) variable.
- **1206**: Value: 1124.0 from minimum nights average (min_navg) variable.
- **1592**: Value: 6079.4 from maximum nights average (max_navg) variable.
- **1608**: Value: 34.46 from reviews per month (rws_m) variable.
- **3165**: Value: 11 from bathrooms variable (bth_txt) and Value: 120 from host listings count (h_lst_c) variable.
- **3510**: Value: 0 from host response rate variable (h_res_r) and Value: 0 from review score rating (r_sco_rt) variable.
- **3628**: Value: 6347.4 from maximum nights average (max_navg) variable.
- **3653**: Value: 22.52 from reviews per month (rws_m) variable
- **3762**: Value: 0 from host acceptance rate (h_acc_r) variable.

After identifying all the values, we will replace them with NA to impute them using MICE, the method used in the missing preprocessing step.

| | h_res_r | h_acc_r | accs | bth_txt | price | min_navg | max_navg | av_365 | r_sco_rt | h_lst_c | rws_m | amts_c |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **409** | 100 | 100 | 2 | 1.0 | 88 | 1.1 | 7.0 | 258 | 4.73 | 3 | 4.48 | 30 |
| **948** | 100 | 96 | 2 | 1.0 | 70 | 1.0 | 1125.0 | 363 | 5.00 | 9 | 0.70 | 58 |
| **1206** | 79 | 100 | 1 | 1.0 | 106 | 3.0 | 1125.0 | 0 | 4.55 | 1 | 1.31 | 28 |
| **1592** | 96 | 94 | 3 | 1.0 | 142 | 5.2 | 1125.0 | 306 | 4.64 | 6 | 2.45 | 36 |
| **1608** | 100 | 100 | 1 | 5.0 | 18 | 1.0 | 6.9 | 362 | 4.45 | 6 | 4.27 | 21 |
| **3165** | 70 | 64 | 1 | 2.0 | 32 | 32.0 | 180.0 | 364 | 3.00 | 1 | 0.16 | 22 |
| **3510** | 100 | 100 | 2 | 1.0 | 56 | 180.0 | 999.0 | 344 | 4.67 | 10 | 0.02 | 34 |
| **3628** | 100 | 91 | 4 | 1.0 | 136 | 5.2 | 1125.0 | 307 | 4.68 | 4 | 1.90 | 43 |
| **3653** | 100 | 100 | 2 | 1.0 | 106 | 1.1 | 1125.0 | 321 | 4.73 | 1 | 3.89 | 34 |
| **3762** | 100 | 100 | 10 | 7.5 | 500 | 5.0 | 31.0 | 75 | 4.00 | 1 | 0.02 | 14 |

Figure 23: Samples considered outliers after MICE imputation. Values imputed underlined.

## 2.3.4 Univariate Final Analysis

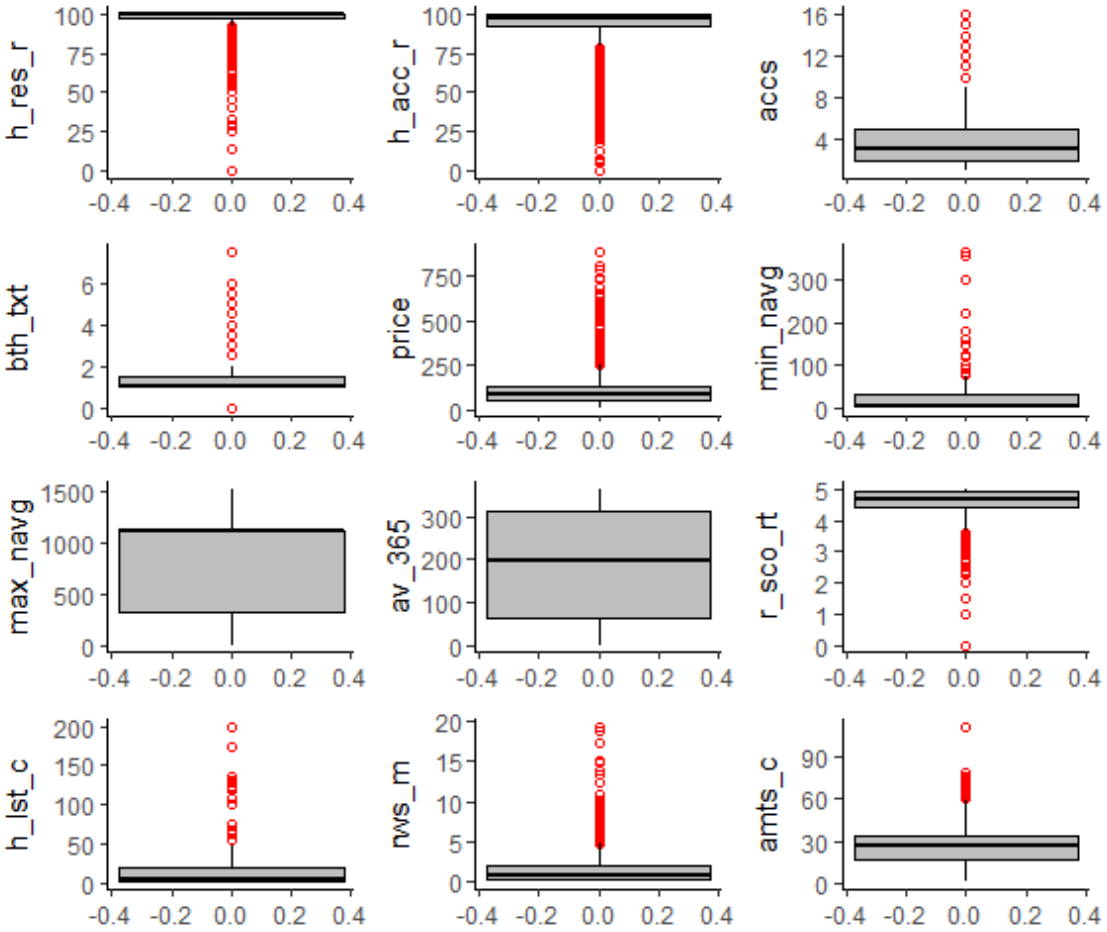



Figure 24: Boxplots of numerical variables after treating outliers.

After treating the outliers, we can now see that there are fewer values that diverge extremely from the mean. Specifically, in variables like 'max_navg', where we have deleted the only two values that were significantly larger than the mean of the variable. Additionally, some of the ranges of the variables in

the plots have decreased, indicating that the samples are now closer together and less divergent.

### 2.3.5 Multivariate Final Analysis

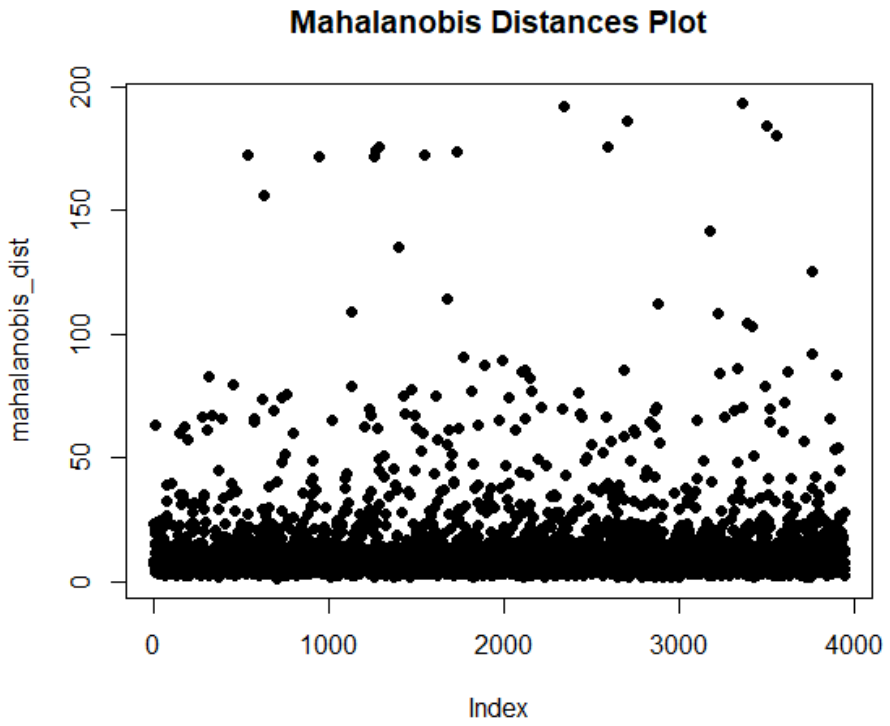



Figure 25: Mahalanobis distance recalculated

If we recalculate the Mahalanobis distance, we can observe that the new distances are now below 200. This suggests that individuals are now more similar compared to before, when there were samples with a distance of nearly 1000. Additionally, we can observe that a significant number of samples exceed the threshold of 125 that we set in the previous step. This is because the distance calculation is now performed on a dataframe without the previously imputed outliers, which causes the distances to vary.

### 2.3.6 Outlier treatment conclusions

After carefully cleaning and imputing our data, we have successfully dealt with outliers using Mahalanobis distance and MICE imputation techniques. To ensure the quality of our results, we also visually inspected our data using Mahalanobis distance plot and Boxplots plots. However, if in the future we notice any issues with our predictive models or any other analysis that relies on this data, we can revisit our outlier treatment and consider imputing additional samples to obtain more accurate results. It is always important to keep an open mind and be willing to revisit our data cleaning techniques to improve the quality of our analysis.

## 2.4 Preprocessing Conclusions

After completing the outlier imputation step using Mahalanobis distance and MICE, we can now conclude that our Airbnb dataset has been completely preprocessed and is now ready for future tasks. By imputing the main outliers, we have solved one of the main problems that can affect the accuracy and reliability of our results.

We have also ensured that the dataset is complete and free of any missing values, which is essential for conducting any meaningful analysis. With this preprocessing step completed, we can now move on to the next phase of our analysis, which may involve running statistical models, analyzing our data with advanced clustering, developing ACM and working with time series. By taking the time to thoroughly preprocess our data, we can be confident that our results will be robust and reliable.

# 3. Multiple Correspondence Analysis (MCA)

Once we have our dataset fully preprocessed, we move forward to Multiple Correspondence Analysis (MCA), the analysis of multiple qualitative variables, whose purpose is to detect and represent underlying structures in the dataset. MCA represents data as points in a low-dimensional space, similar to PCA, but works with categorical variables.

The reason for using categorical variables in MCA is that they cannot be analyzed using traditional statistical methods due to their non-numerical nature. By using MCA, we can identify patterns in the data and summarize them into a smaller number of dimensions, making it easier to interpret the results. Moreover, MCA allows us to identify relationships between different categories and visualize the data in a meaningful way.

To begin with, we will select the principal components and use visualization techniques such as factorial maps for individuals, modalities, and categories to determine similarities between variables and different aspects, helping us understand the database and its various relations among all the variables.

## 3.1. Selection of principal components

Our first step in the MCA analysis is to select the categorical variables. We will use the 10 categorical variables available in our dataset: source, h_loc, h_res_t, s_host (binary), h_ver, h_id_v (binary), n_gr_cl, r_type, inst_bk (binary), and gender, as active categorical features. Additionally, we will include all the quantitative variables as supplementary variables to provide additional information to the analysis. These variables do not define the dimensions of the analysis but can aid in the interpretation of the results by helping to identify factors that contribute to the observed patterns in the data. We will also visualize the data using factorial maps from individuals, modalities, and categories to determine similarities between variables and gain a better understanding of the database and its different relations among all the variables.

Once we have identified the variables, we will proceed to perform the MCA analysis using the FactoMiner library. We will use the default "indicator" method for the MCA, instead of the "burt" method, as we are not working with a large amount of data where a burt table may be more suitable for reducing complexity. With the "indicator" method, each category from our categorical variables will be treated as a binary variable.

The MCA was performed using 10 active categorical variables (p) and 40 different modalities (k), giving us a maximum number of dimensions of (k-p), which is 30. To select the dimensions for the analysis, we need to consider the variance from the eigenvalues. We will choose the last dimension where the variance is higher than 1/p (1/10), which means it is interesting to see up to 13 dimensions. However, we will perform our analysis with only 3 dimensions, as this is sufficient to identify patterns, and using all 13 dimensions may be too tedious.

In the eigenvalues of the MCA, we can see the variance of each dimension, as well as the percentage of the total explained variance. The total inertia of the data, which is the sum of the explanatory variances of all dimensions, is 3. As shown in the plot below, the variance of the first dimension is

significantly higher than the other dimensions, suggesting that visualizing it may be sufficient for the analysis. However, since 13 dimensions is a large number to analyze, we will focus on the first 3 dimensions, which contain a substantial percentage of the total inertia.

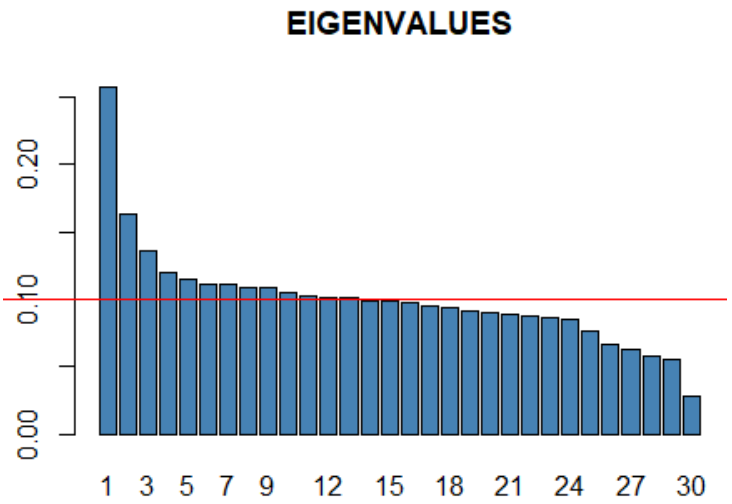



Figure 26: Plot that shows the variance of each dimensions provided by the eigenvalues.

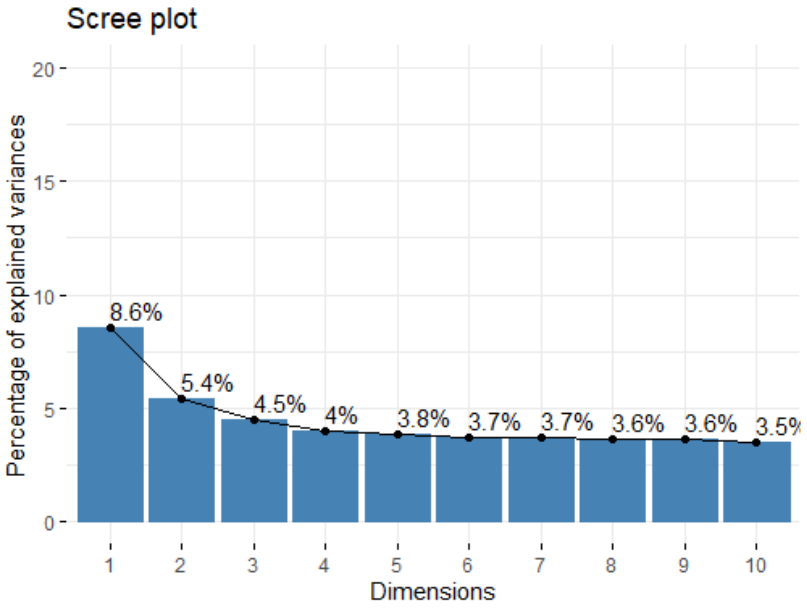



Figure 27: Plot that shows the percentage of explained variance of the first 10 dimensions.

## 3.2. Results and analysis

### 3.2.1. Contributions

Once the MCA is performed, it is time to see the results and analyze them. First of all, we are going to look at the contributions of the modalities of the variables in the dimensions. In the following plots,

we are going to see the 10 modalities that explain the highest percentage of inertia in each combination of dimensions, which means that they are the categories most related to each dimension.

***dimensions 1&2***

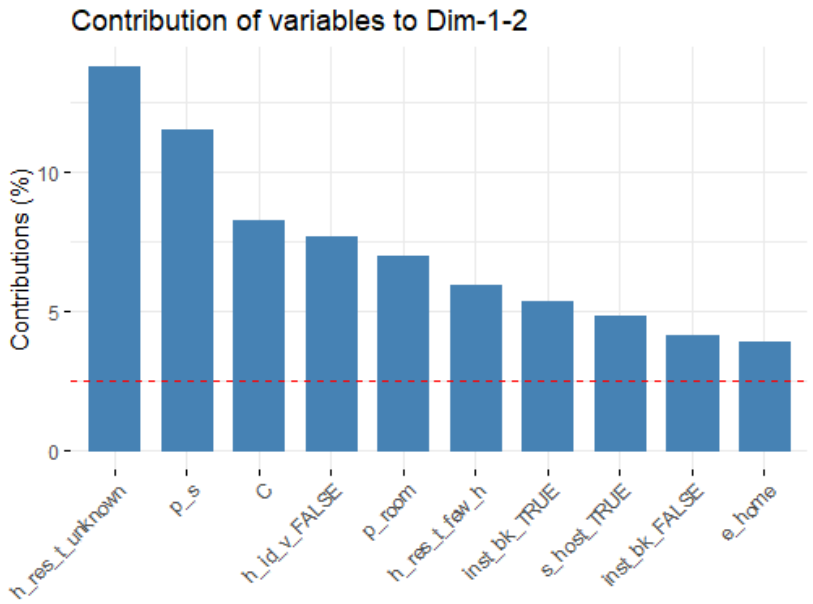



Figure 28: Plot that shows the 10 categories that have a better percentage of contribution to dimension 1 and 2.

***dimensions 1&3***

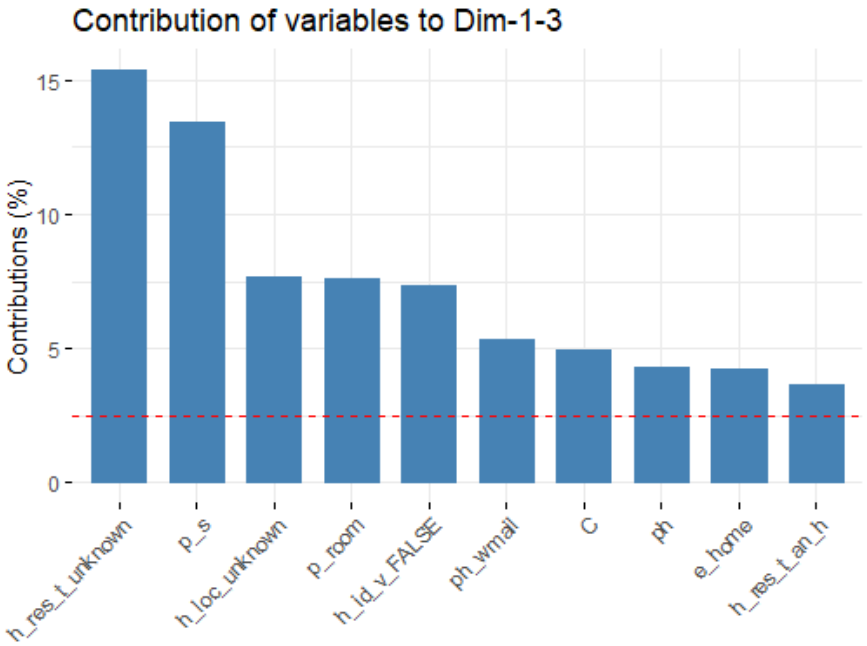



Figure 29: Plot that shows the 10 categories that have a better percentage of contribution to dimension 1 and 3.

***dimensions 2&3***

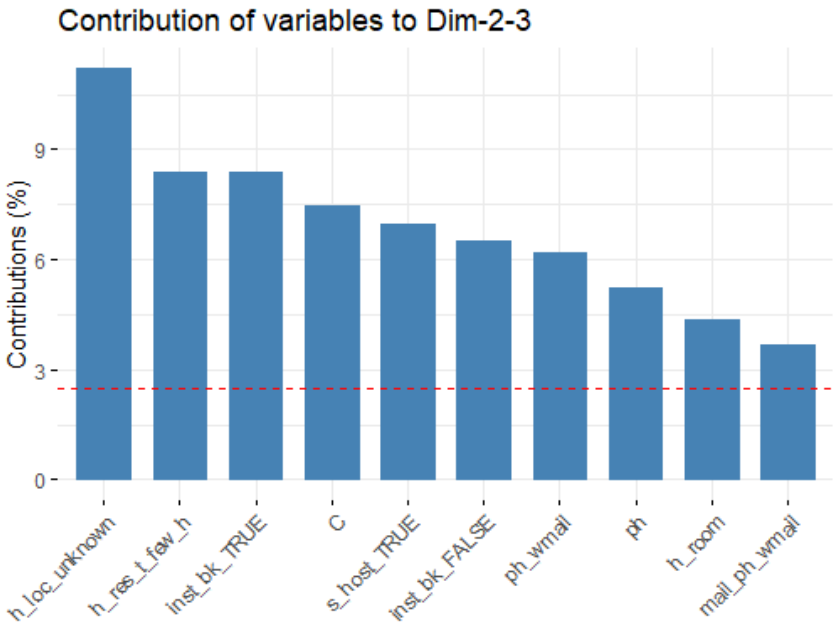



Figure 30: Plot that shows the 10 categories that have a better percentage of contribution to dimension 2 and 3.

In the first combination of dimensions, the top five modalities are: the host is a company renting out the property, the host's ID is not verified, the response time is unknown, the property is a private room, and it has been previously booked. For the second combination of dimensions, the top five modalities are: the host's response time is unknown, the property has been previously booked, the host's location is unknown, the property is a private room, and the host's ID is not verified. Finally, the last combination shows the top five categories as follows: the host's location is unknown, the host's response time is a few hours, the property is instant bookable, the host is a company, and the host is a superhost.

## 3.2.2. Factorial analysis

Our next step consists of a factorial analysis to identify the different patterns and relationships among the variables. For each combination of dimensions, we will first plot the categories to visualize them accurately. Then, we will show the individuals. Additionally, the plot of the categories will show the contribution of each factor.

***dimensions 1&2***

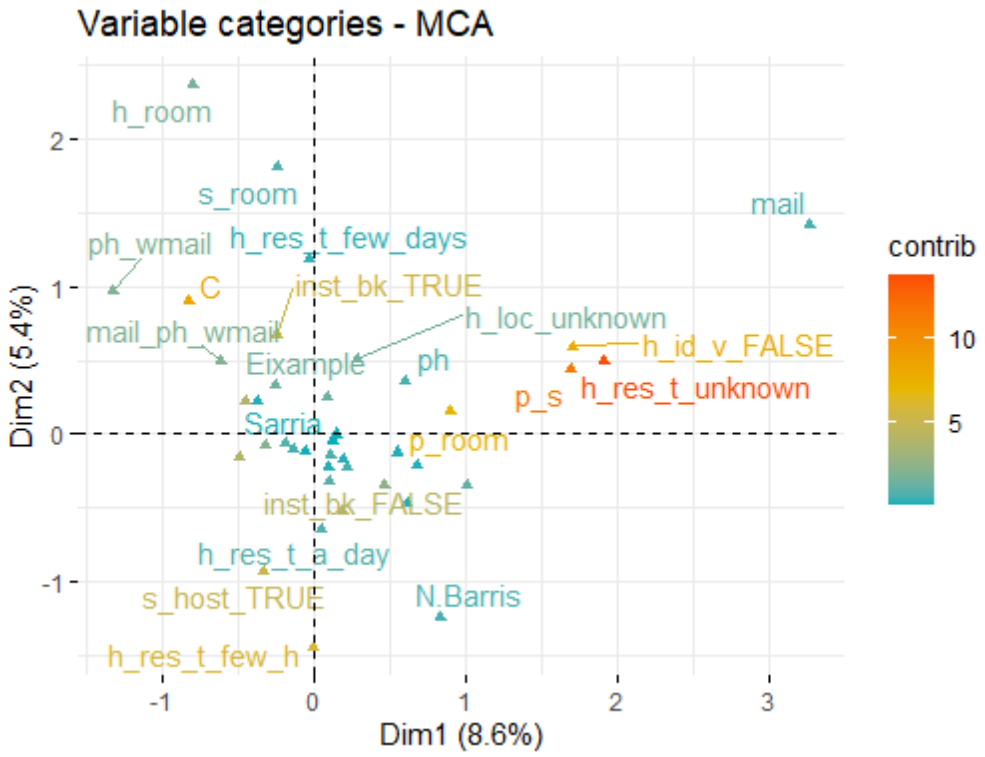



Figure 31: Factorial map from categories in dimension 1 and 2.

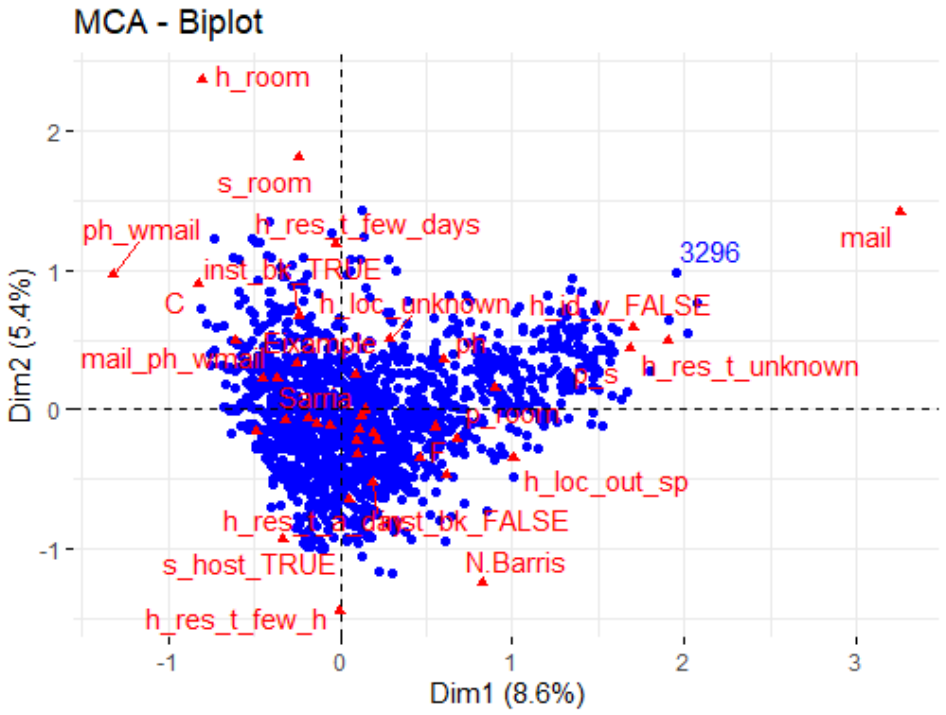



Figure 32: Factorial map from categories with individuals in dimension 1 and 2.

In the first plot, we can see the relations with the categories more clearly. For example, we can see Sarria or Eixample, which are two common districts of the city, so close to each other. The factors that are far away from the origin have more contribution to the total variance of the dataset. For example, we can see the factor "mail," which is not close to any other factor, and this could mean that the apartments whose hosts have the verification of the mail are relevant to the price or are so different

from the other ones. Other important factors are the host response time in few hours, apartments located in Nou Barris, the host response time unknown, and apartments that are shared rooms. The categories that are the furthest and are closer to each other, such as host verification id false and host response time unknown are so related. Another fact to analyze is that factors like hotel room and host response time few hours are negatively correlated, which means that an hotel room, its host doesn't delay few hours in order to response.

As we can see in the individual plot, those factors that are far away from the center are not similar to the ones that are close to the origin. Almost all individuals share categories that are closer to the origin. In terms of contribution, we can observe that almost all the factors that are far away from the origin are the most contributory. However, we can also see that the factor "mail" is not as contributory as we might expect. This could be because this factor is strongly correlated with another factor that contributes a lot, such as host response time unknown.

***dimensions 1&3***

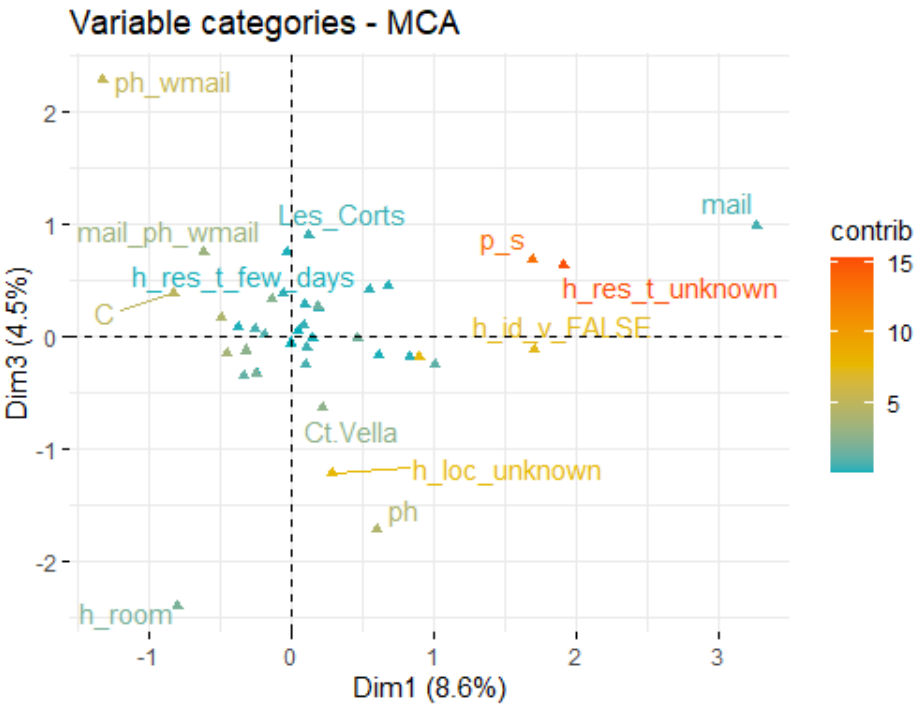



Figure 33: Factorial map from categories in dimension 1 and 3.

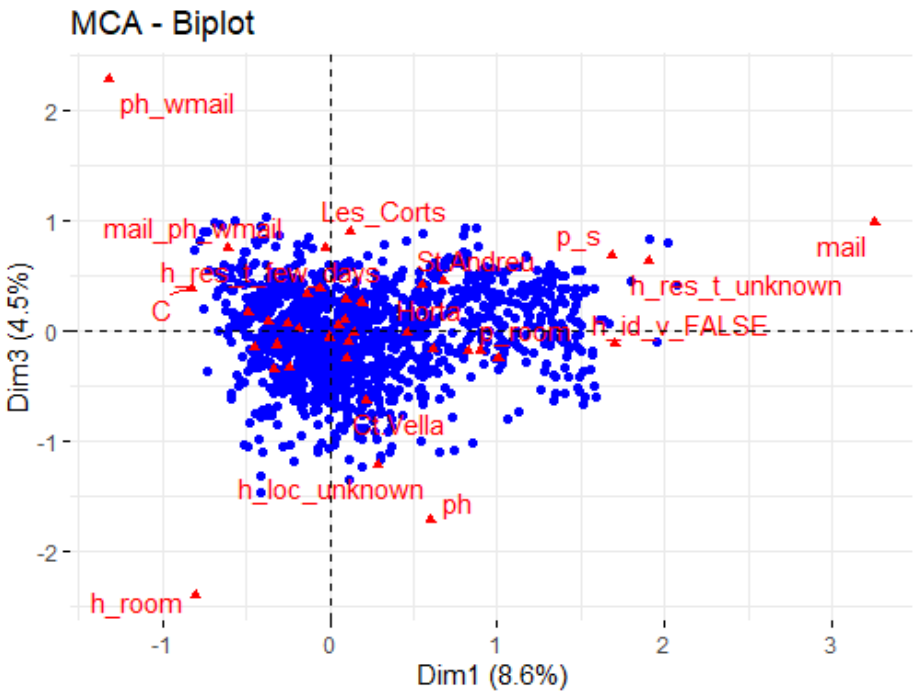



Figure 34: Factorial map from categories with individuals in dimension 1 and 3.

In the plots for dimensions 1 and 3, we can see similar results as in the previous plots. The factor mail is still far from the origin, and other factors such as whether the host has verified their phone and work email or whether the apartment is a hotel room are also far away. In this case, there are more factors closer to the individuals, which could mean that the factors that are farther away are more important, as we can also see from their contributions. Moreover, we can see how negatively correlated are hotel room and phone and work email verification, that could mean that an hotel room doesn't have this verification.

***dimensions 2&3***

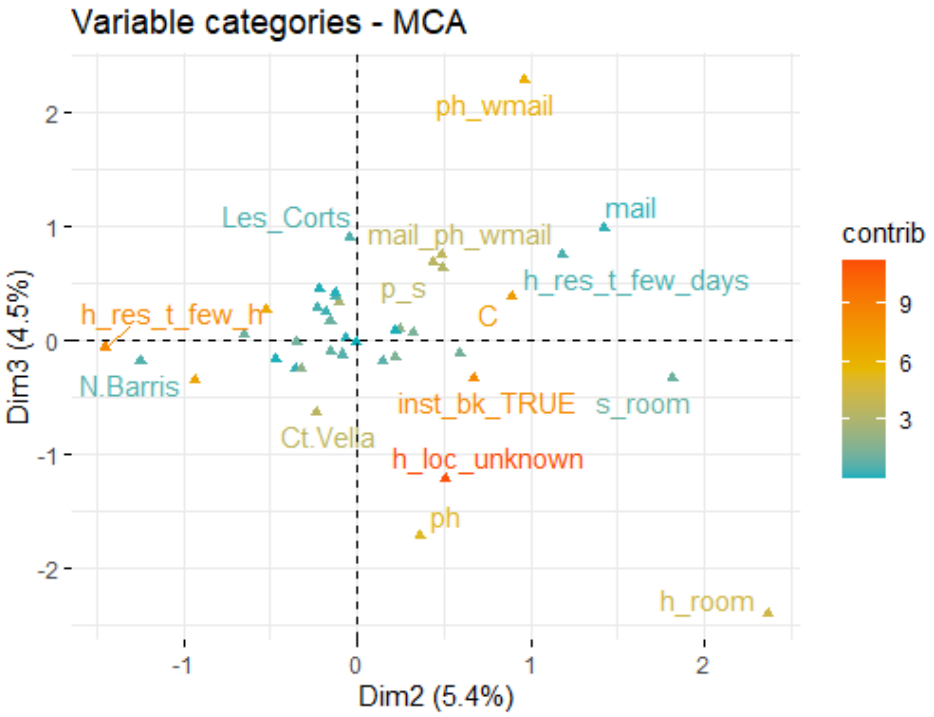



Figure 35: Factorial map from categories in dimension 1 and 2.

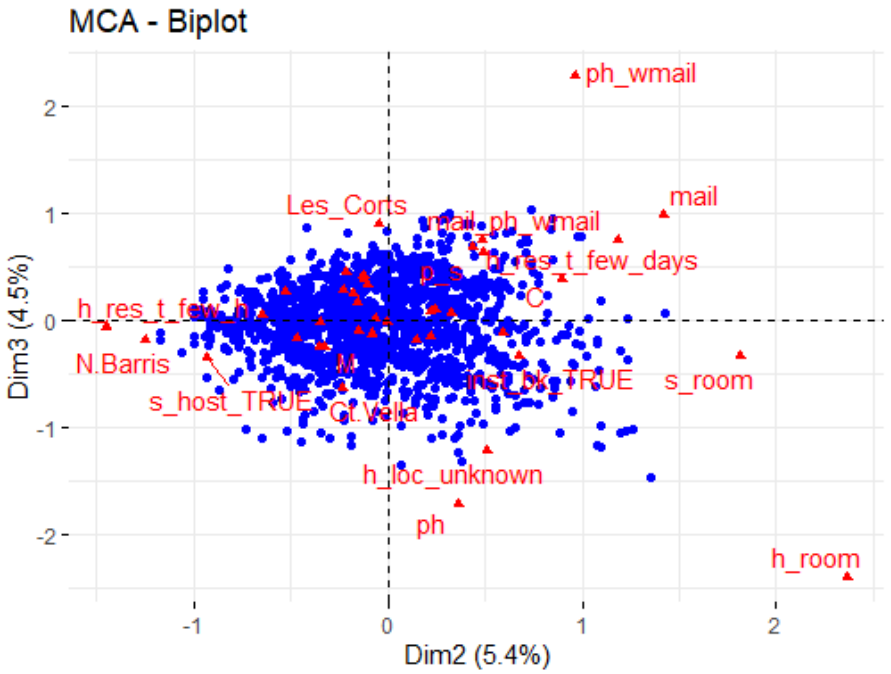



Figure 36: Factorial map from categories with individuals in dimension 2 and 3.

Finally, looking at the dimensions 2 and 3, we can see more clearly 5 factors that are far away from the origin who are hotel room, shared room, mail verification , host response time in a few hours and phone and work mail verification. Moreover, factors host response time in a few hours and Nou Barris are strongly correlated, which could mean that the apartments from that district, its owners have a delay of a few hours to respond. Another time hotel room has a negative correlation with phone and work mail verification.

In conclusion, the most important factors are related to variables about verifications and the response time of the host. Indeed, factors like host response time unknown or host id verification false are strongly correlated, as well as some factors of verifications are negatively correlated to hotel room. In the following plots, we will analyze the similarities of the variables for each dimension and try to determine the different aspects we have observed in the factor maps. We will also include some qualitative variables in our analysis.

***dimensions 1&2***

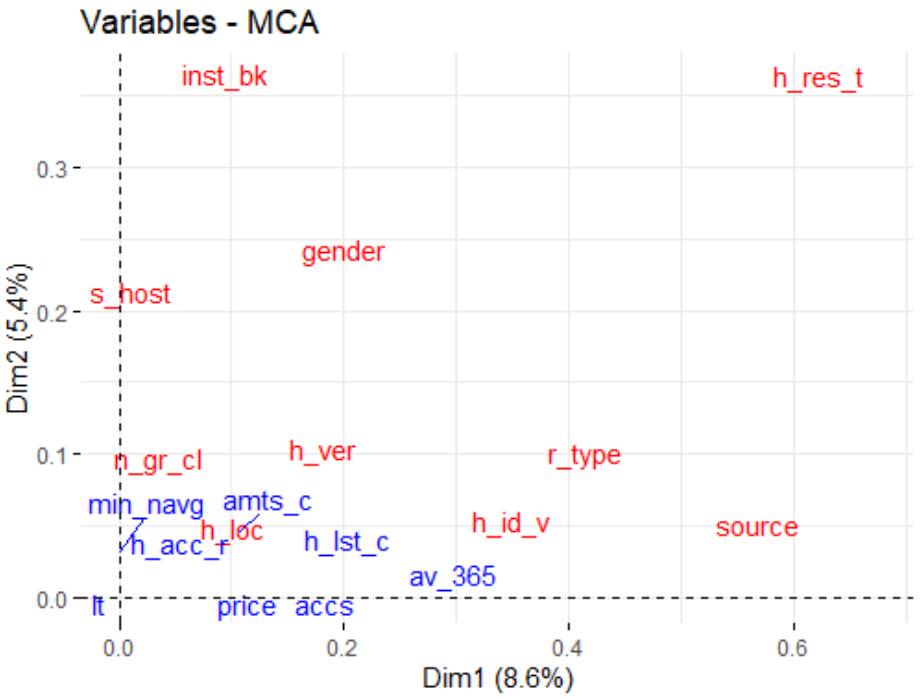



Figure 37: Plot from variables in dimension 1 and 2.

***dimensions 1&3***

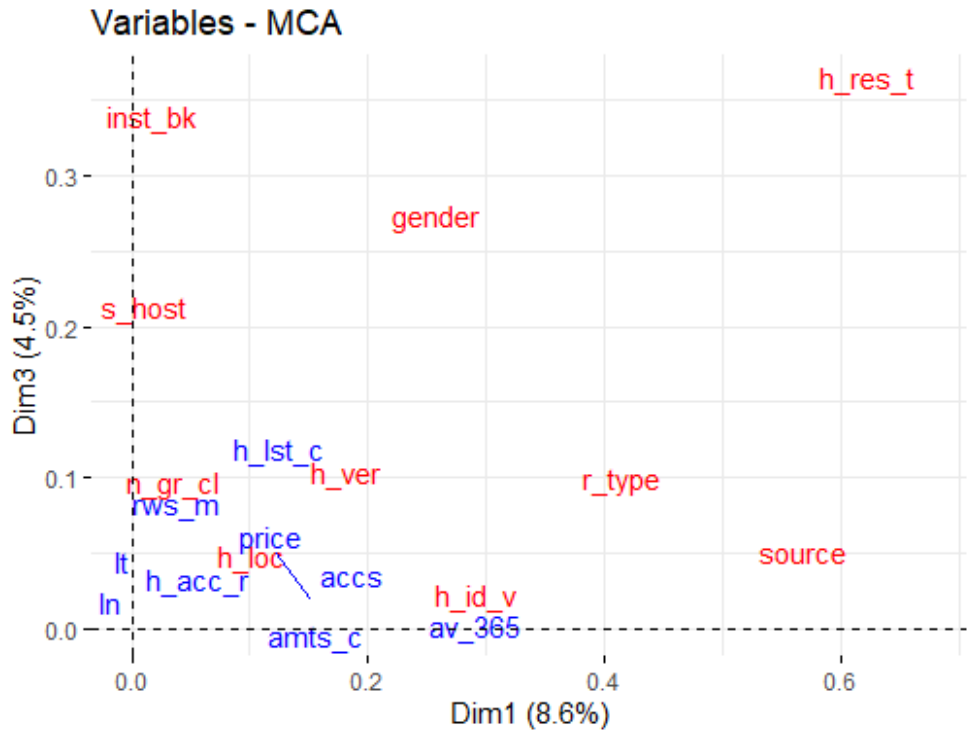



Figure 38: Plot from variables in dimension 1 and 3.

***dimensions 2&3***

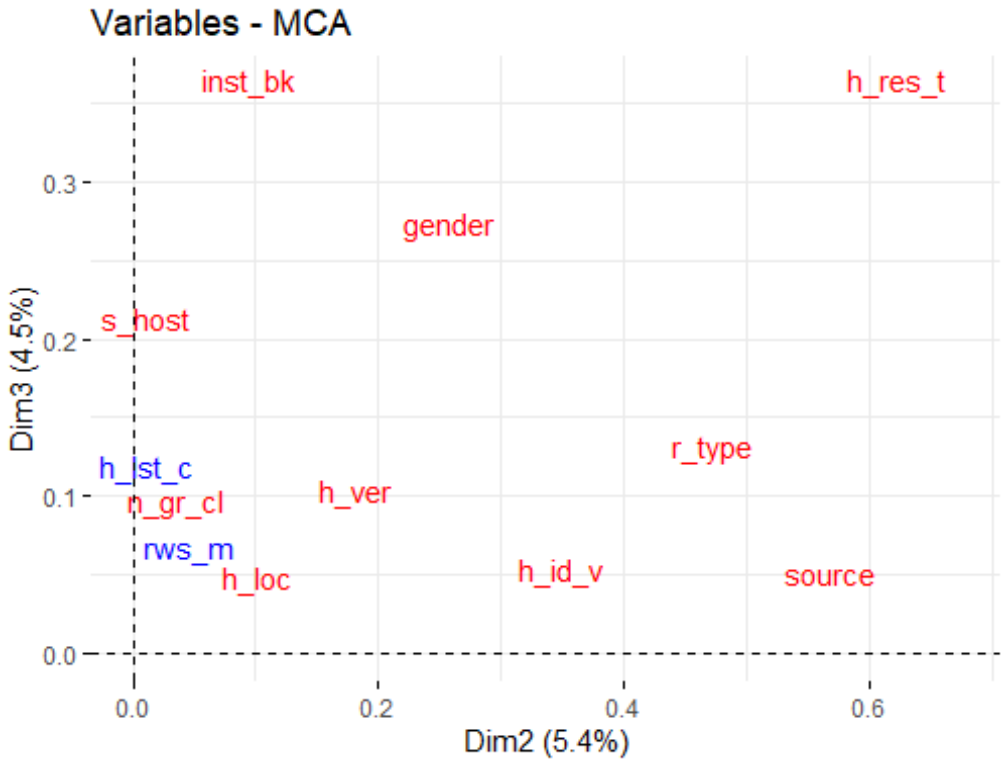



Figure 39: Plot from variables in dimension 2 and 3.

As we have seen in the plots above, there are many important variables, with the most important being host response time, followed by instant bookable, gender, room type, and source. It is interesting to note that the variable that was supposed to be far away from the center, host verification, is actually one of the closest. This suggests that factors like mail may not be as important as previously thought.

On the other hand, the least important variables are the district of the apartments and the location of the host. However, the variables are not close enough to each other to affirm that they are important between them.

### 3.2.3. Plot of the individuals

In addition to what we have done, we have also plotted every categorical variable for dimension 1 and 2 to see its behavior.

***source***

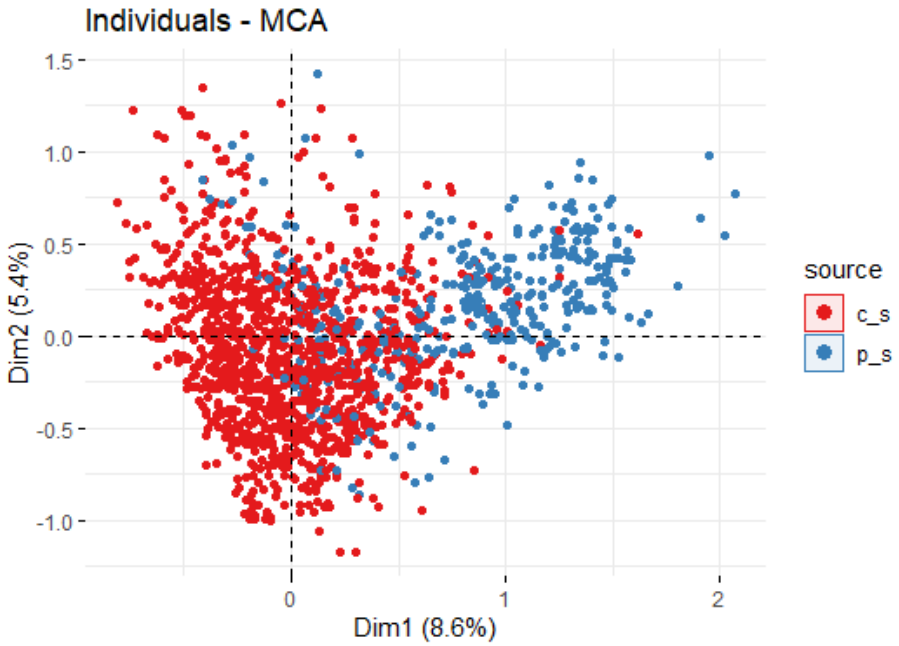



Figure 40: Plot from the individuals of the variable source.

This variable shows that the category "previous scrape" is the one that is the farthest away, meaning that it is the most important. This category has also appeared in our factor analysis.

***h_loc***

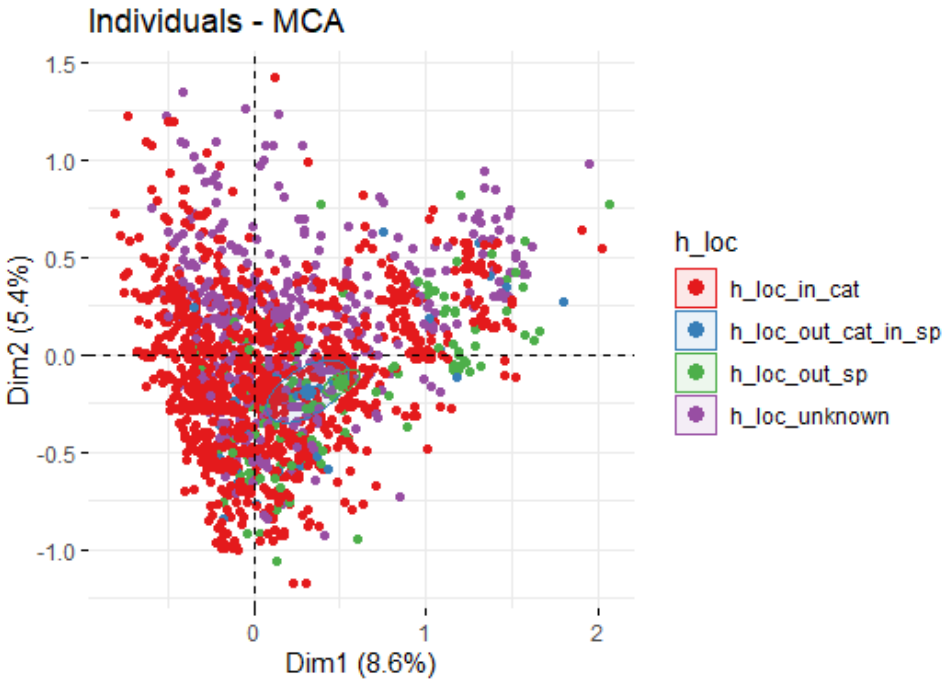



Figure 41: Plot from the individuals of the variable host location.

This variable was one of the less important and as we can see, all categories are centered, an aspect that affirms what was said before.

***h_res_t***

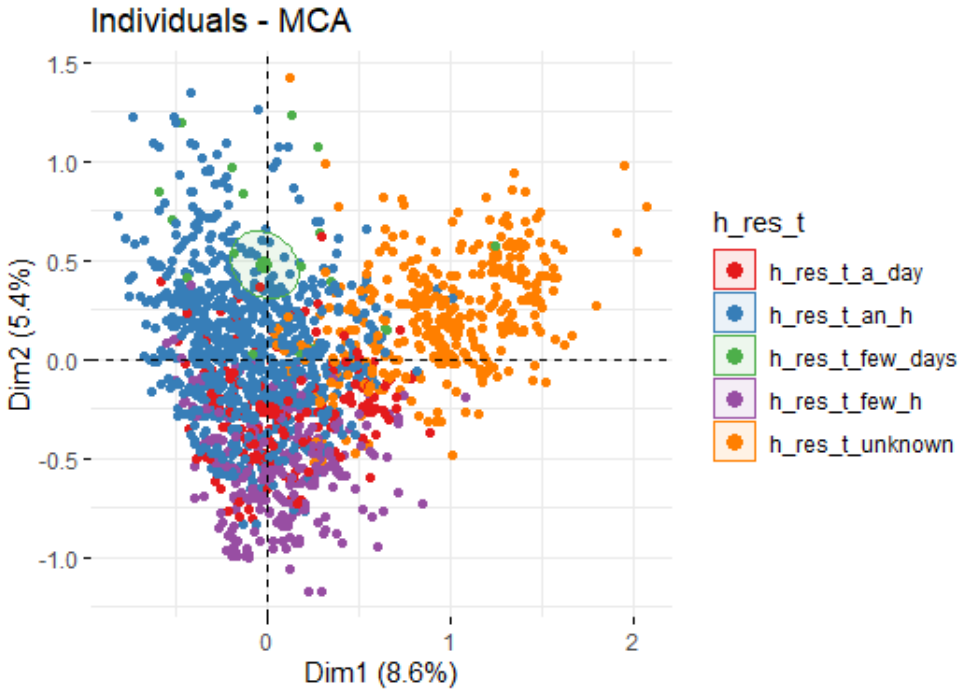



Figure 42: Plot from the individuals of the variable host response time.

This variable shows that the categories "host response time unknown" and "host response time in few hours" are the most uncentralized, which corresponds to what we have seen before as they were the most contributors.

***s_host***

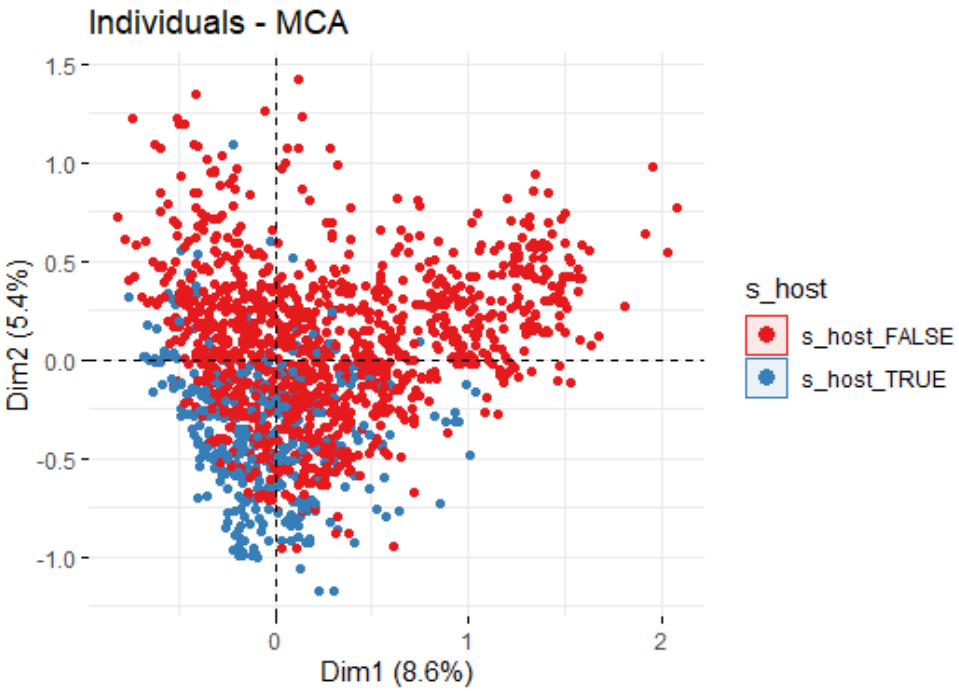



Figure 43: Plot from the individuals of the variable host is superhost.

This variable doesn't show anything to appreciate.

***h_ver***

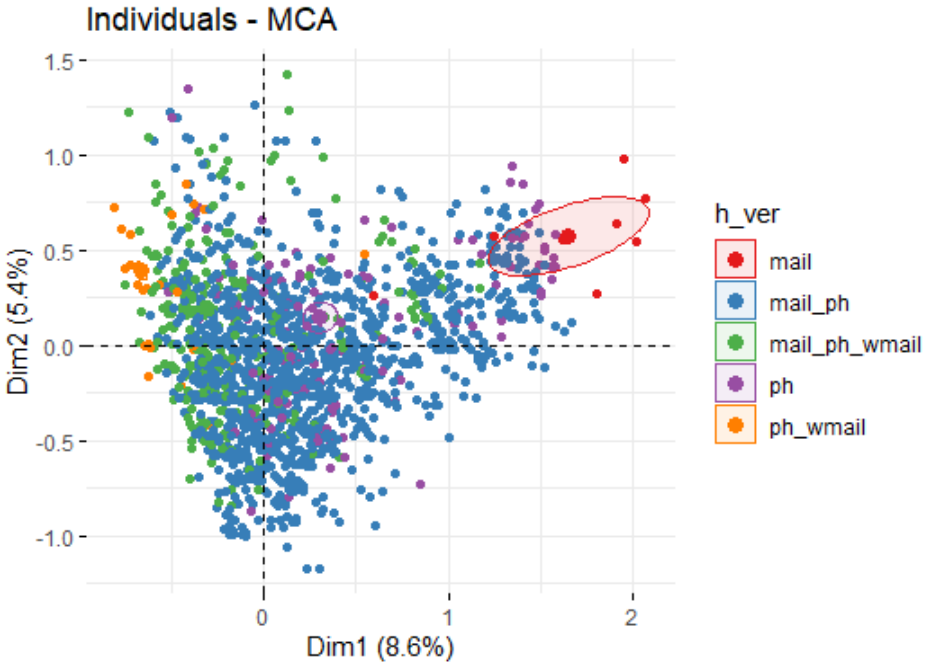



Figure 44: Plot from the individuals of the variable host verifications.

In this variable, we can see how perfectly the individuals with the mail category are far away from the center. This phenomenon could happen because apartments with this aspect are so different from the others.

***h_id_v***

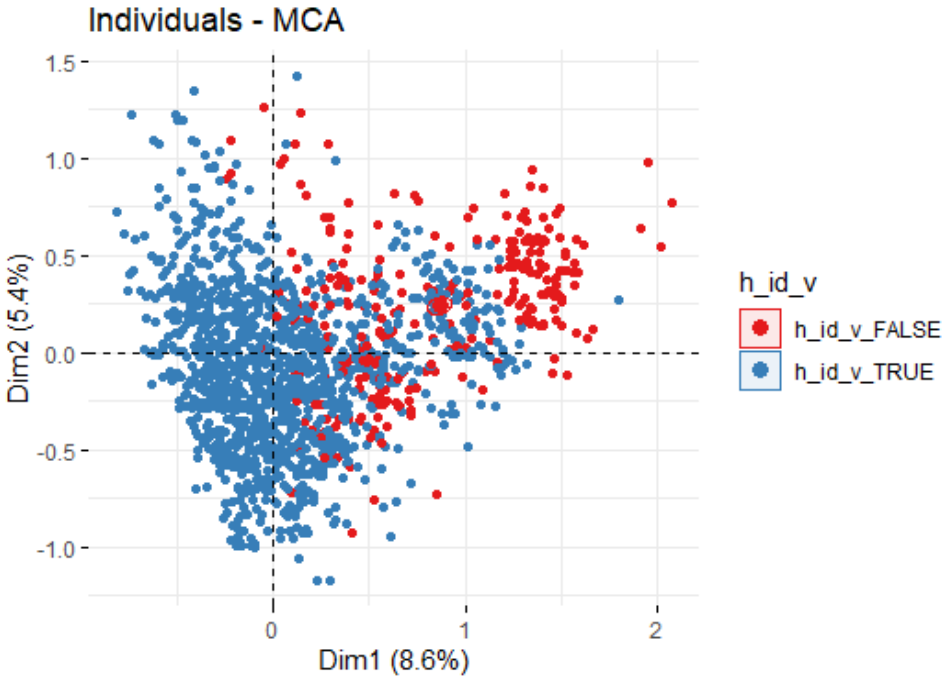



Figure 45: Plot from the individuals of the variable host id verificated.

In this variable, the category that is the furthest away from the center is "host has not verified their ID", as we have seen before.

***n_gr_cl***

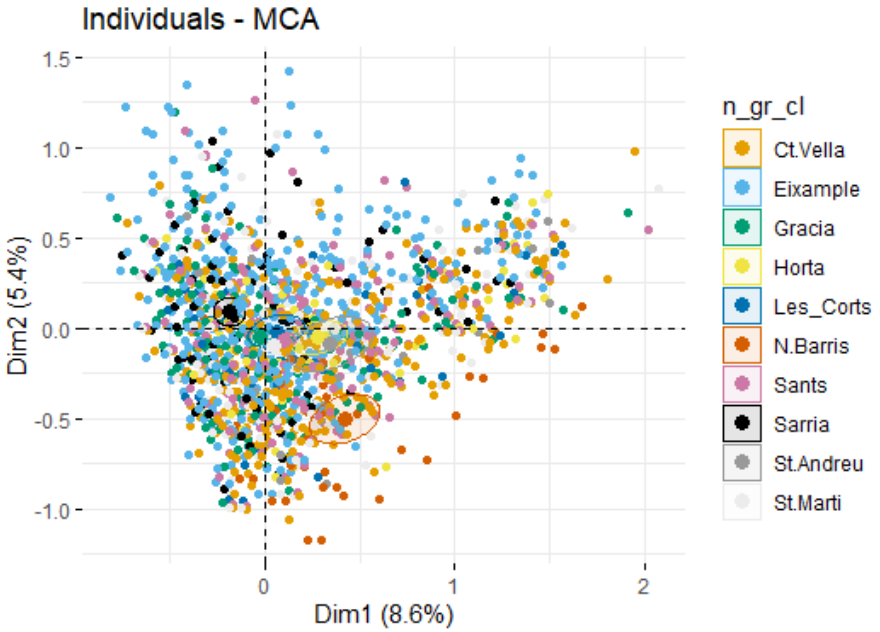



Figure 46: Plot from the individuals of the variable district.

This variable does not indicate much about the importance, but the individuals that are farthest away are from Nou Barris.

***r_type***

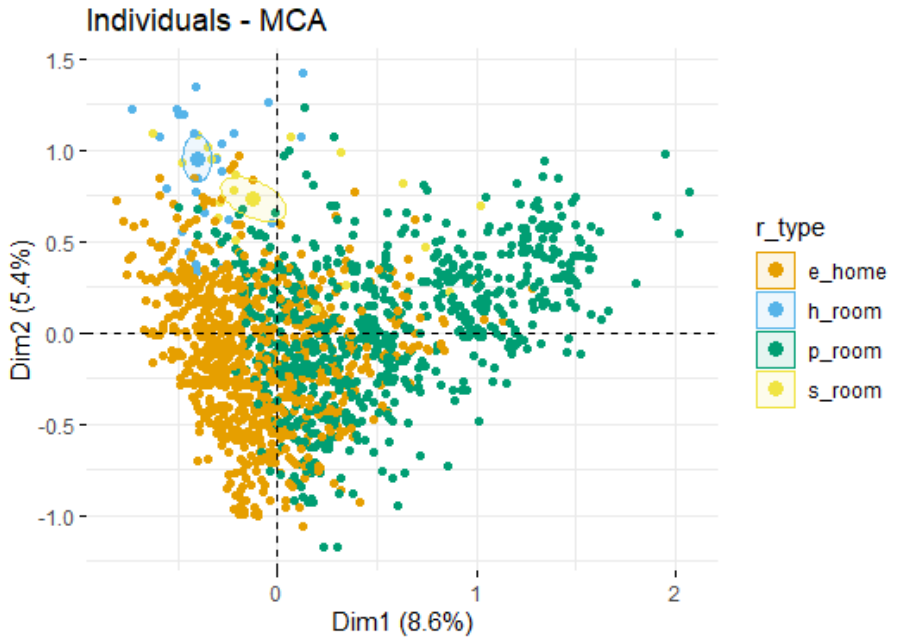



Figure 47: Plot from the individuals of the variable room type.

This variable shows how the categories shared room and hotel room are the most important as they are the furthest away from the origin.

***inst_bk***

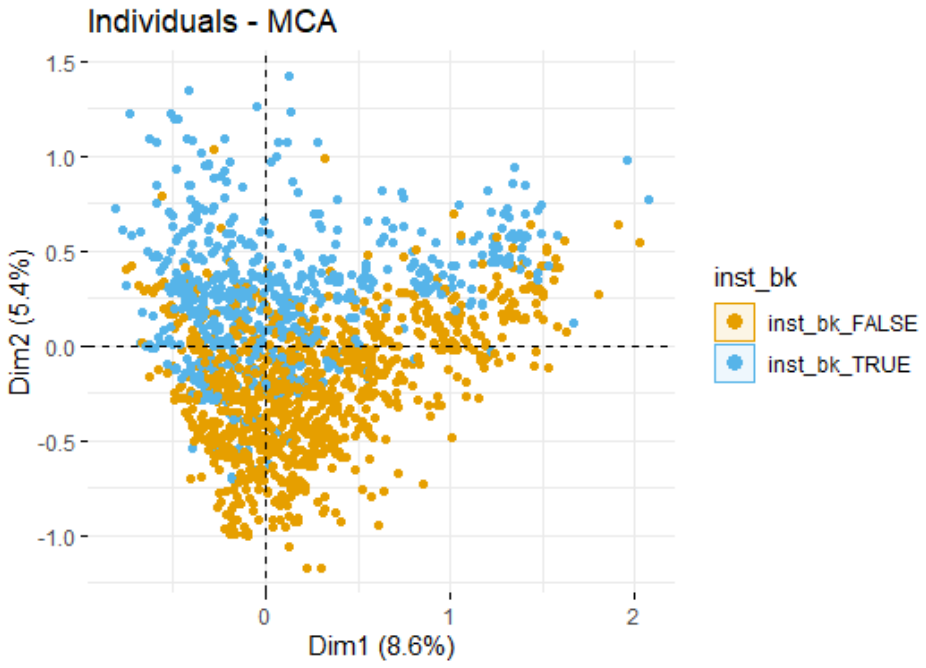



Figure 48: Plot from the individuals of the variable instant bookable.

In this variable, we can observe a negative correlation between both categories, indicating that the individuals of each category are dissimilar.

***gender***

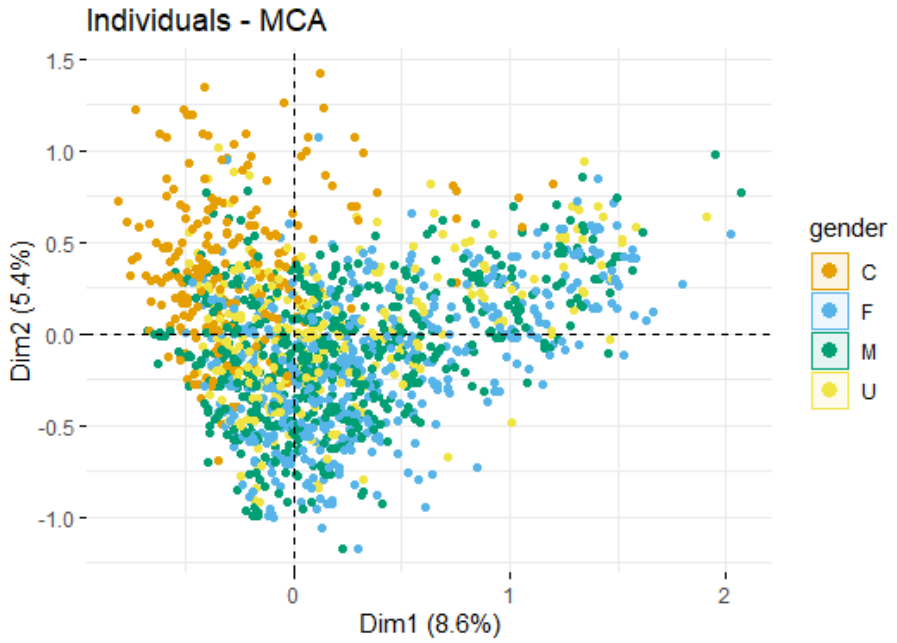



Figure 49: Plot from the individuals of the variable gender.

Finally, in this last variable, the gender variable, we can see that the category with the most individuals far from the origin is the "Company" category, which means that this category is the most important.

## 3.3. Conclusions

Summing up, this MCA analysis has shown many different and interesting aspects. Firstly, the most important variables in determining the price of an Airbnb listing are host response time, instant bookable, gender, room type, and source. These variables are closely related to the verification and response time of the host, and the ease of booking for guests. On the other hand, the district of the apartment and the location of the host seem to have a lower impact on the price.

Secondly, the factor analysis revealed that some categories related to host verification and response time are strongly correlated between them and we have shown that apartments without the response time of the host as unknown, this owner has not the id verified. Moreover, categories such as mail verification, phone and work mail verification, and host response time in few hours were the most significant contributors to the variance in the dataset.

Finally, factors about the verifications of the host are negatively correlated to the hotel room one, aspect that could mean that hotels do not have these types of verifications and they have others like just the phone verification.

In conclusion, the analysis suggests that factors related to host verification and response time, as well as ease of booking, are the most important in determining the price of an Airbnb listing. These

findings can be useful for hosts looking to optimize their listing and for guests looking for the best value for their money.

# 4. Advanced Clustering

Our next step to analyze our data will be making a clustering to classify our samples in different groups trying to get some patterns about the different types of listings of Airbnb in Barcelona.

To do this we will need to try different clustering techniques to find the one that best fits our data. After selecting the one that fits better, we will perform an analysis of the results and use the TLP technique to make an understanding conclusion of the clustering we made, helping the experts to comprehend the results.

## 4.1 Construction of the Clusters

The first step is making the different clusters by trying different clustering methods. We will add every cluster that we get as a column to our database for a further analysis in section 4.2.

### 4.1.1 Baseline

Like in every data science method where we need to compare different proposals to solve the same problem, we need to fix a baseline point. A first proposal to establish which is the minimum level of performance from which we have to improve by using different methods or settings for each method.

In this case, we are facing an advanced clustering task, which we have to try and solve in the best possible way. Following what we have said before, our baseline will be a simple clustering method called **K-means**, being k = 5 as an standard value for this parameter.

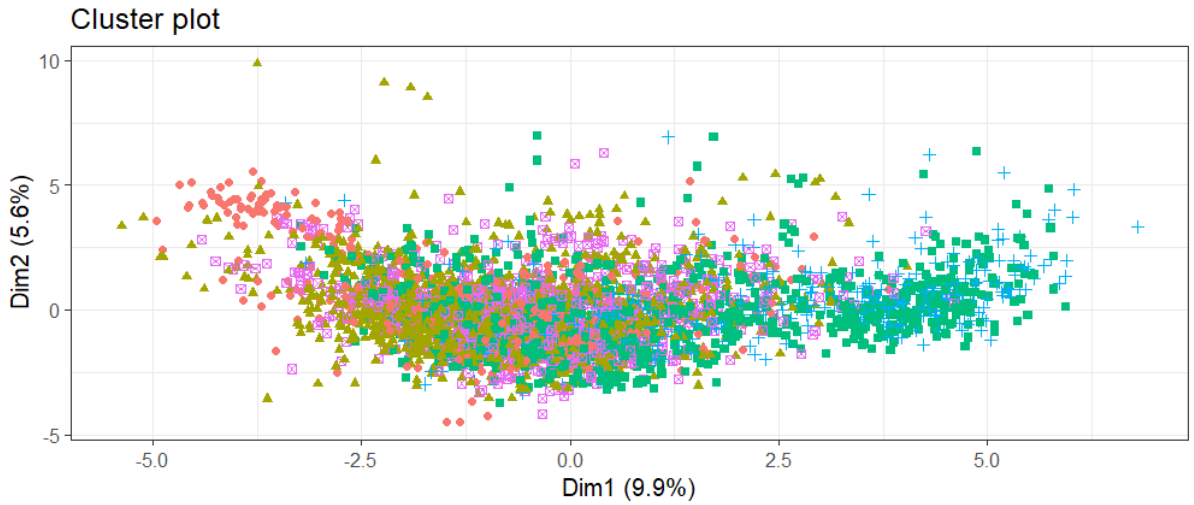



Figure 50: Plot that shows the PCA applied to the clustered data with K-means. We can see that only the 15% of the original information is gathered with this two dimensions.

Figure 1 shows the first two dimensions of PCA applied to our data after applying the K-Means algorithm. We are doing PCA so that we can visually analyze in just two dimensions the performance of this clustering method by gathering the maximum possible information that the original 30 variables gave to us. Before starting to discuss whether this clustering has functioned in a good or bad way, we have to say that the poor variance collected by these two dimensions (15%) tells us that this way of validating the clustering method is not robust enough to extract any conclusions. It will be just a guide.

Having said this, from Figure 1 we could infer that data is not well separated by spatial similarity, not as it is demanded from clustering methods. We can see that most of the clusters are overlapped, meaning that this clustering doesn't separate well the different regions that the variables draw.

From now on, we are going to try and improve this baseline clustering with two different but similar methods: DBSCAN and OPTICS.

### 4.1.2 DBSCAN

As we already know, this clustering method is based on spatial density of data points. It searches high density related spatial regions from which it will form clusters. This method is specially suitable for data that does not draw a circular nor an oval shape, because of its features, it looks for dense regions by recursively visiting all of the points and classifying them based on the other data points surrounding it.

Since our data is not drawing any special shape, but an oval one as we can see in Figure 1, this method would not be the perfect one to face this task, because it adds an unnecessary extra computational cost to the process of clustering that is only useful when we have non-circular shapes. Even so, we will apply it to learn how it fits our data.

First of all, we should mention that this method includes the setting of two sensitive parameters, **epsilon** and **min_pts**. The first one is an upper bound for the distance at which another point (looking from the one that we are treating) is considered to belong to the same dense region as the one that we are treating. The second one is a lower-bound that establishes the minimum points that a dense-region should have to be considered dense and, in consequence, a cluster itself. This is to avoid noise-points and outliers becoming a cluster itself.

To determine the optimal value for min_pts, as the literature stands, we have to calculate the 0.25% of all the samples from our data and round this number to the upper integer. This is the minimum portion that should be considered a cluster itself. If we do so, as we have a total number of 3.947 samples, we would get a number of **15** min_pts, this will be our optimal value.

Now, the last thing to determine is the value of epsilon. To optimize this value and get the best DBSCAN clustering, we will do K-NN being k = 5, as stated in the literature. We will do this to extract an elbow plot. This plot shows in the X-axis the samples in the dataset in ascending distance order and in the Y-axis the corresponding distance. This gives us a global vision of the distances that we should consider short and large using our dataset as the reference.

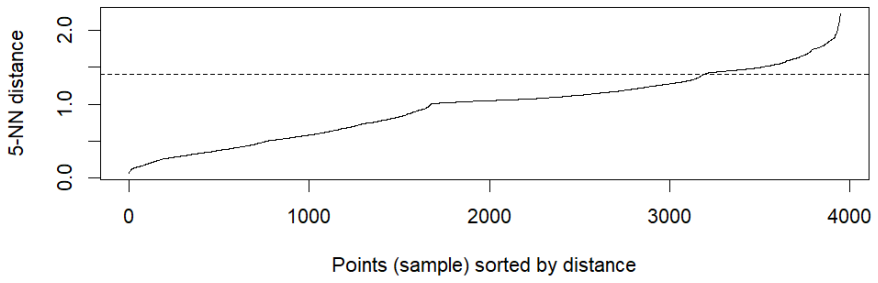
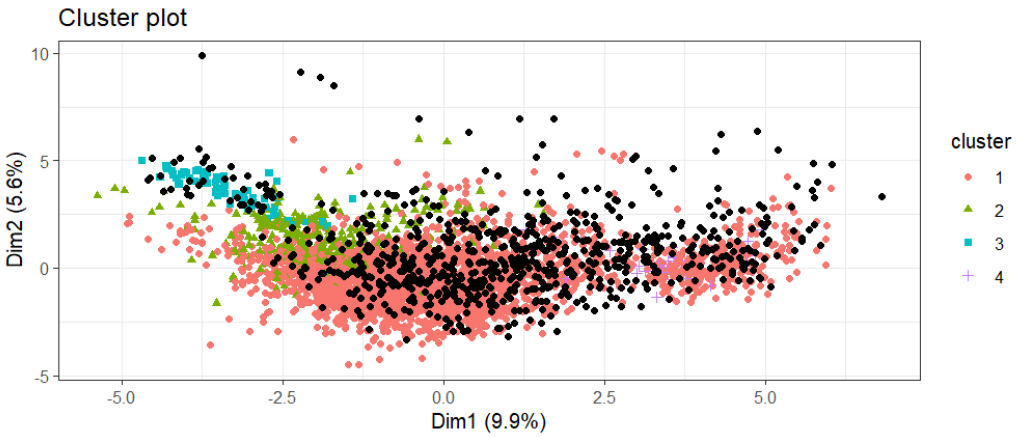



Figure 51: Elbow plot extracted from applying a 5-NN to our Dataset. As we can see, the distance between samples grows exponentially since d = 1.4.

As we can see in the elbow plot of Figure 2, all distances from samples 0 to 3.000 grow in a linear way. Since this point, samples are distanced by a much larger distance as we can see that the distance values grow exponentially. This point mentioned before is our “elbow”. The distance at which we will consider a point to be far, our epsilon. This value, as we can see in the dashed line, is **1.4**.

Having optimized our hyper-parameters, we apply the DBSCAN algorithm to our standardized numerical variables and we get the following clustering:



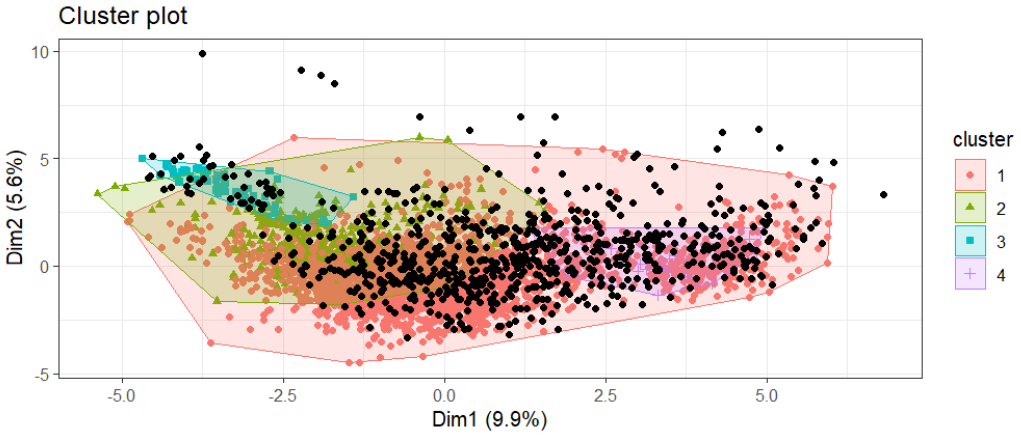



Figure 52: Upper image shows the PCA plot after applying DBSCAN clustering to our dataset. We get 4 clusters. Lower image shows the same, but with the convex hull closing every cluster.

As we can see in Figure 3, DBSCAN results in 4 clusters. We can distinguish a big cluster represented by the red dots, and three minor clusters represented in green, blue and purple. Black dots are the samples considered noise.

Bearing in mind what we said before about PCA, we can say that this clustering method substantially improves the K-means clustering as it doesn't significantly overlap the clusters. We can say that, visually, it separates the samples in space in a good way.

### 4.1.3 OPTICS

This is another alternative method of advanced clustering based on density. It is a generalization of DBSCAN where it generalizes to other ranges. Like DBSCAN, we should set the epsilon and min_pts hyper-parameters. In the case of OPTICS, epsilon becomes an upper-bound of neighborhood recognition. To test other strategies, we will try two different alternatives of optimizing the values of the hyper-parameters.

#### Grid searching for epsilon and min_pts

This method consists of creating a grid of all possible combinations of values for epsilon and min_pts within a finite range and computing, for each possible combination the **silhouette** value of the resulting clustering. The silhouette is a metric that computes the wellness of the clusterization of each point in the dataset. For every point, it calculates within a range of [-1,1] how good it fits in its cluster, based on the mean distance. From all the combinations, the one which maximizes this value is the chosen setting for epsilon and min_pts.

| | V1 | V2 | V3 |
|---|---|---|---|
| result.1 | 0.1 | 5 | -0.171998421 |
| result.2 | 0.2 | 5 | -0.344979182 |
| result.3 | 0.3 | 5 | -0.300383454 |
| result.4 | 0.4 | 5 | -0.233066073 |
| result.5 | 0.5 | 5 | -0.185241049 |
| result.6 | 0.6 | 5 | -0.112558320 |
| result.7 | 0.7 | 5 | -0.072571045 |
| result.8 | 0.8 | 5 | -0.025285681 |
| result.9 | 0.9 | 5 | -0.003868106 |
| result.10 | 1.0 | 5 | 0.009494346 |

Figure 53: Small part of the grid of all combinations of values of epsilon and min_pts. The third column represents the value of the Silhouette metric for the corresponding combination.

By doing this process we get the values of:

- eps = 1.5
- min_pts = 5

And we get the following Reachability plot:

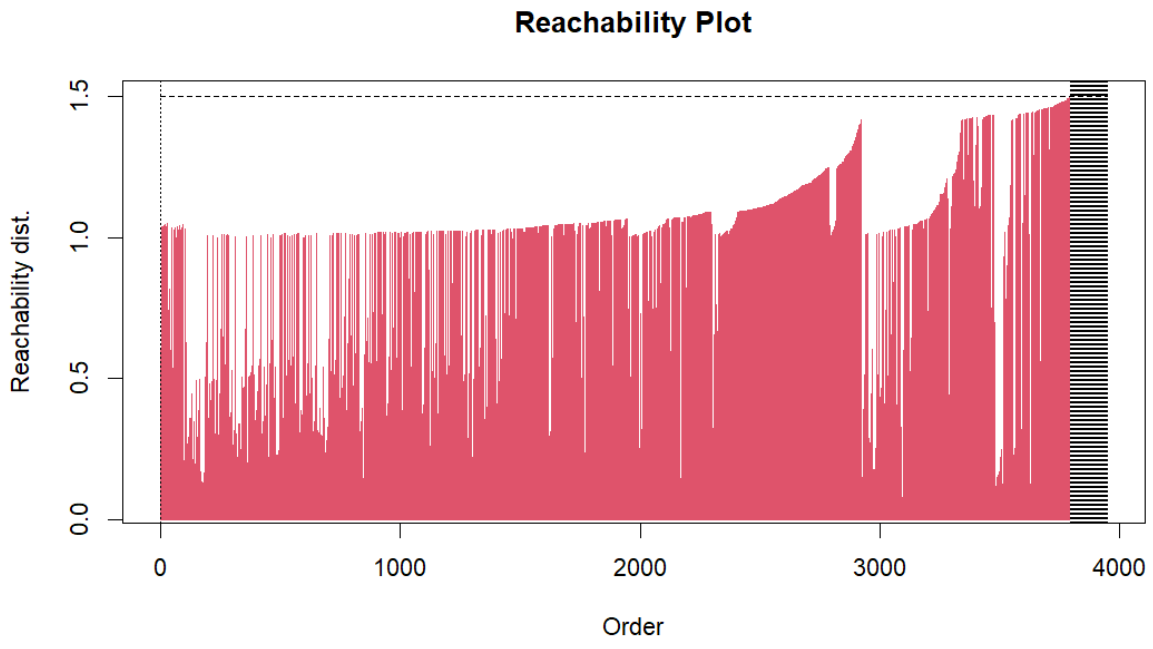



Figure 54: Reachability plot returned by OPTICS with eps = 1.5 and min_pts = 5.

Since the optimal value for the epsilon is set at 1.5, we just get a single cluster containing the whole dataset (without some noise points). From the reachability plot we can see that there are a lot of reachable regions (potential clusters) below 0.5 reachability distance, so we can conclude that this method is not returning an optimal clustering.

There's a functionality that enables us to extract a real clustering from this reachability plot. It is the following one:

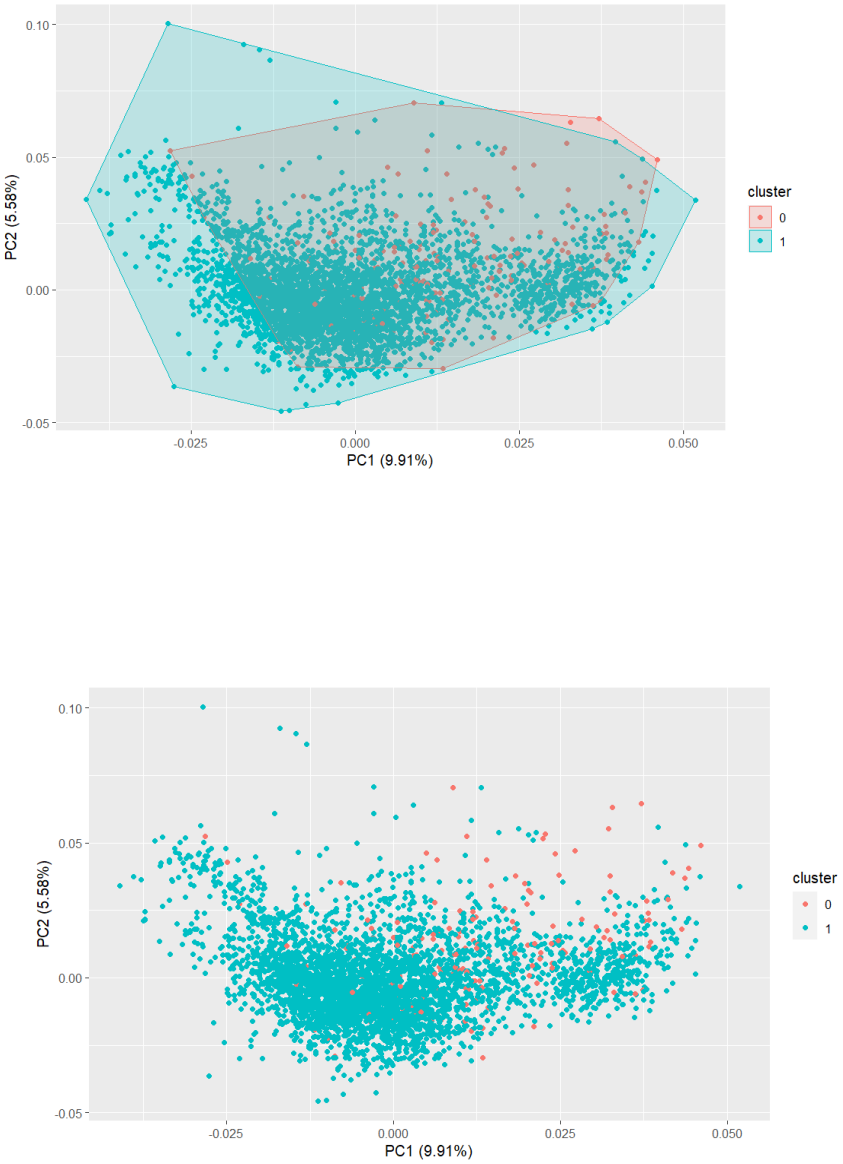



Figure 55: Upper image shows the PCA plot after applying OPTICS clustering to our dataset. We get one single cluster. Lower image shows the same, but without the convex hull closing every cluster.

### Silhouette method for obtaining the value for epsilon with min_pts fixed

In this case we will apply a similar process to the previous one, by executing OPTICS with different values for epsilon and extracting a plot of the Silhouette value in function of the epsilon value. We will choose the epsilon that maximizes this Silhouette value with min_pts fixed to 5 as the literature stands. We get the following plot:

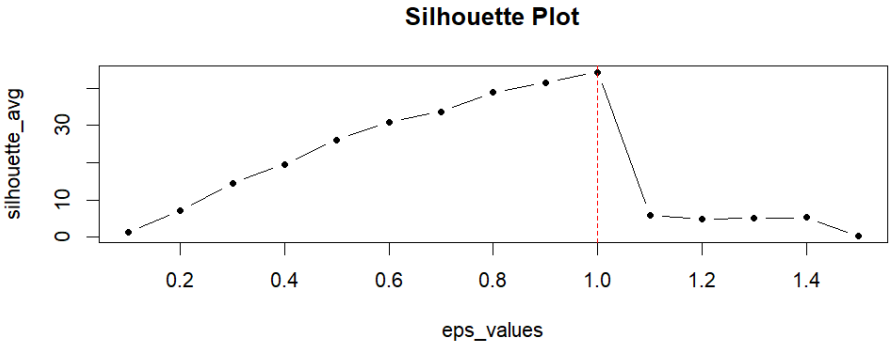



Figure 56: Silhouette plot for a min_pts value fixed to 5.

Looking at the plot in Figure 7, we can see that the epsilon value that maximizes Silhouette is eps = 1. We execute OPTICS with this particular settings and we get the following reachability plot:

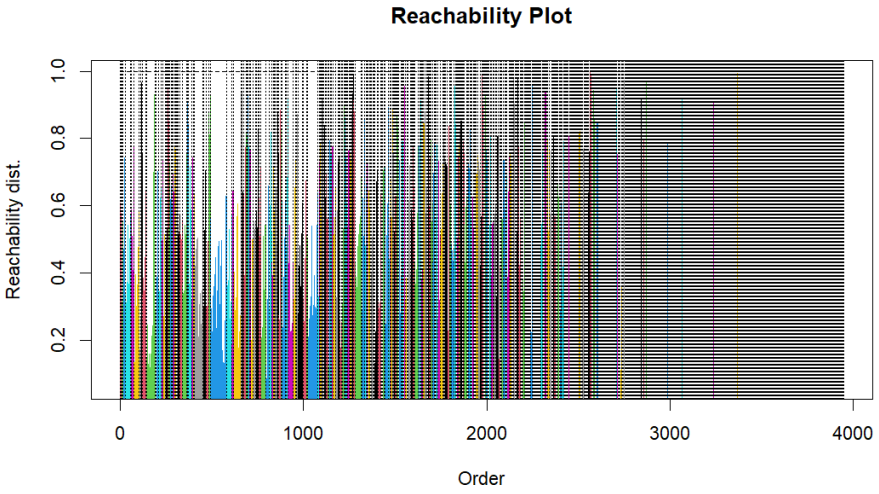



Figure 57: Reachability plot returned by OPTICS with eps = 1 and min_pts = 5.

It is important to note that two different methods of optimizing the hyperparameters have resulted in similar, but different values for the epsilon hyper-parameter. This is a noticeable and strange fact.

In Figure 8 we can see that, in contrast with the first alternative, multiple clusters have been formed as the reachability distance has decreased. Since the plot has a lot of fluctuations, we can say that it forms too many clusters. A total number of 190 clusters. It is not an optimal clustering. Let's see at the PCA, to see if it has separated the samples in a good way:

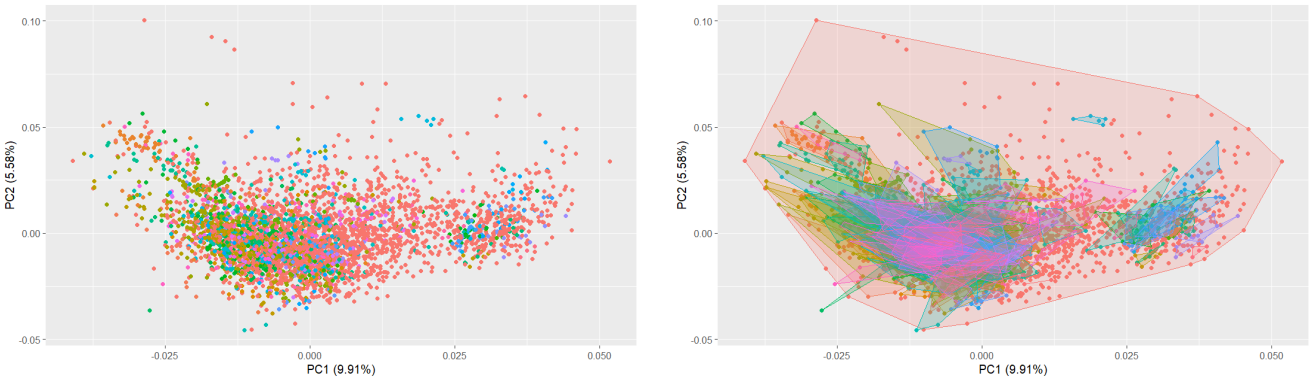



Figure 58: Left image shows the PCA plot after applying OPTICS clustering to our dataset. We get a total of 190 clusters and nearly 2.000 noise points. Right image shows the same, but with the convex hull closing every cluster.

A total of 190 clusters and 2.000 noise points is not a good cluster. We can say it is worse than the K-means clustering. The reason for getting so many clusters is because of the small numbers used either for epsilon and min_pts. A small value for epsilon tends to create more clusters because more values for distances are considered “far”, and small values for min_pts also tends to make small clusters, because the restriction to declare a region “dense” is softer.

### 4.1.4 Hierarchical clustering

Now that we have seen the density based clustering we will try some distance based clustering techniques. First we will try the classical hierarchical clustering to see how well it clusters our data. Then we will try CURE which is a method that uses hierarchical clustering on a sample of the data and then calculate the centroids with the most representative individuals to assign a cluster to the rest of the data.

Before doing the cluster we had to make our categorical variables dummies to be able to use them in the cluster calculations. Once we had the dummies, we also scaled the data, because there are variables like reviews or price that have large numbers and others that have lower values. If we did not scale the data, the large variables would have more impact than the smaller ones.

When we have the data ready, we proceed with the hierarchical clustering which is easy to do and also gives us a good baseline for the CURE. We will be using the euclidean distance and the ward method to do the cluster.

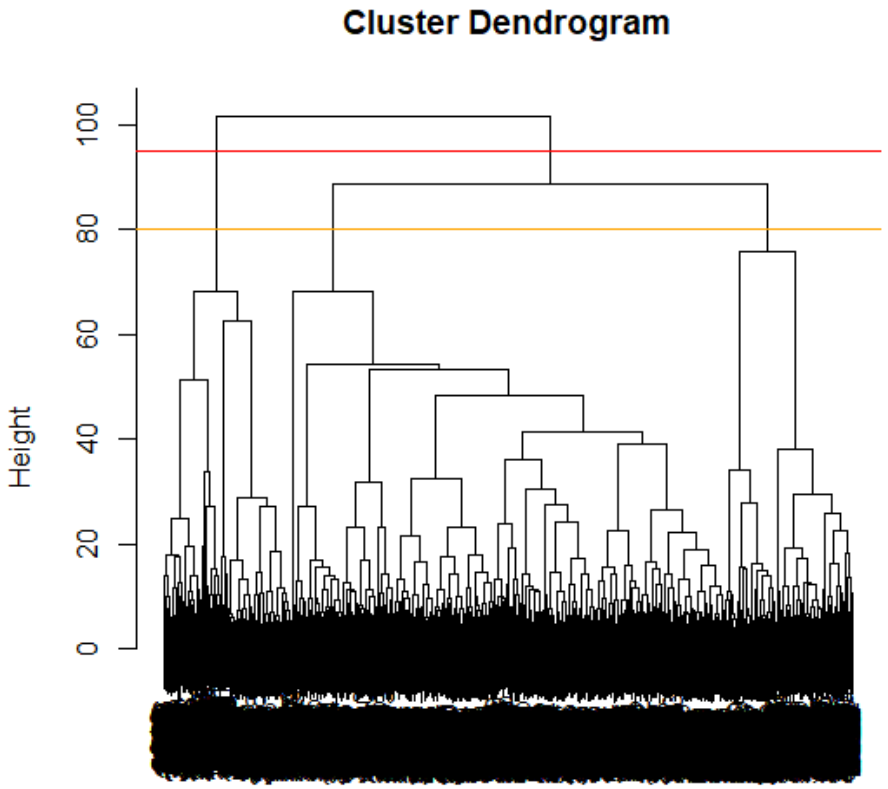
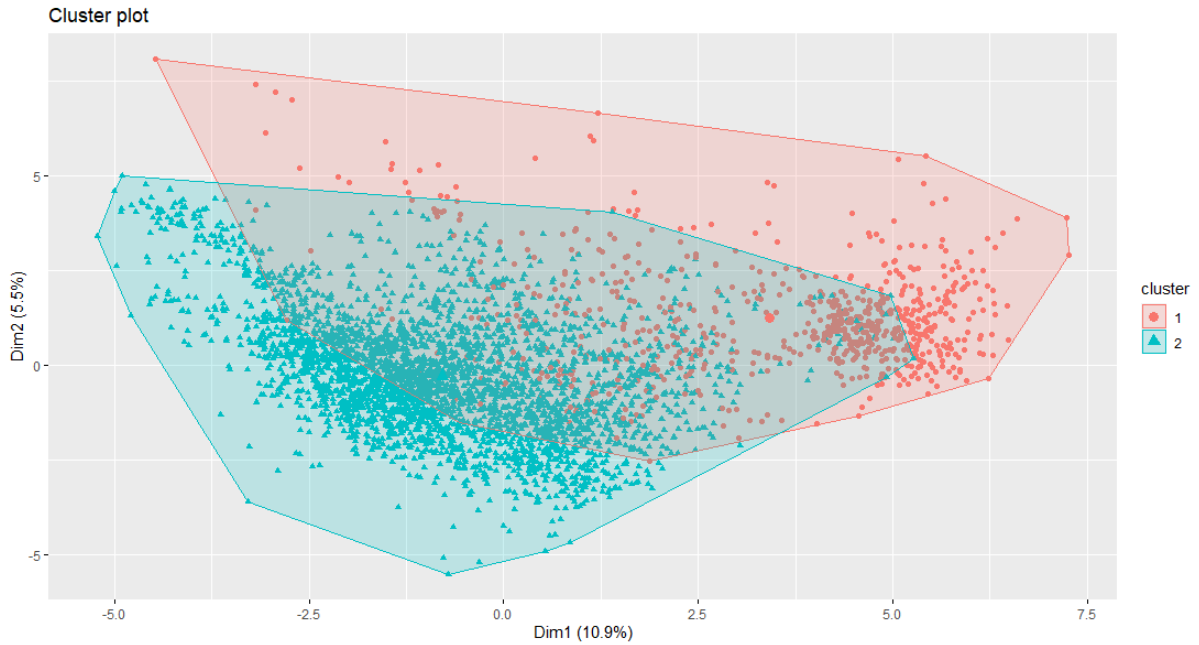



Figure 59: Cluster Dendrogram

As we can see in the dendrogram, the optimal k must be k=2 or k=3. That's because those have the larger distance between clusters and as the distance are very similar we will try both k.

As we did in the other clustering methods we will visualize the results with fviz_cluser, although the information shown in the graph will be low, so it's not the best representation of how good the clustering is.



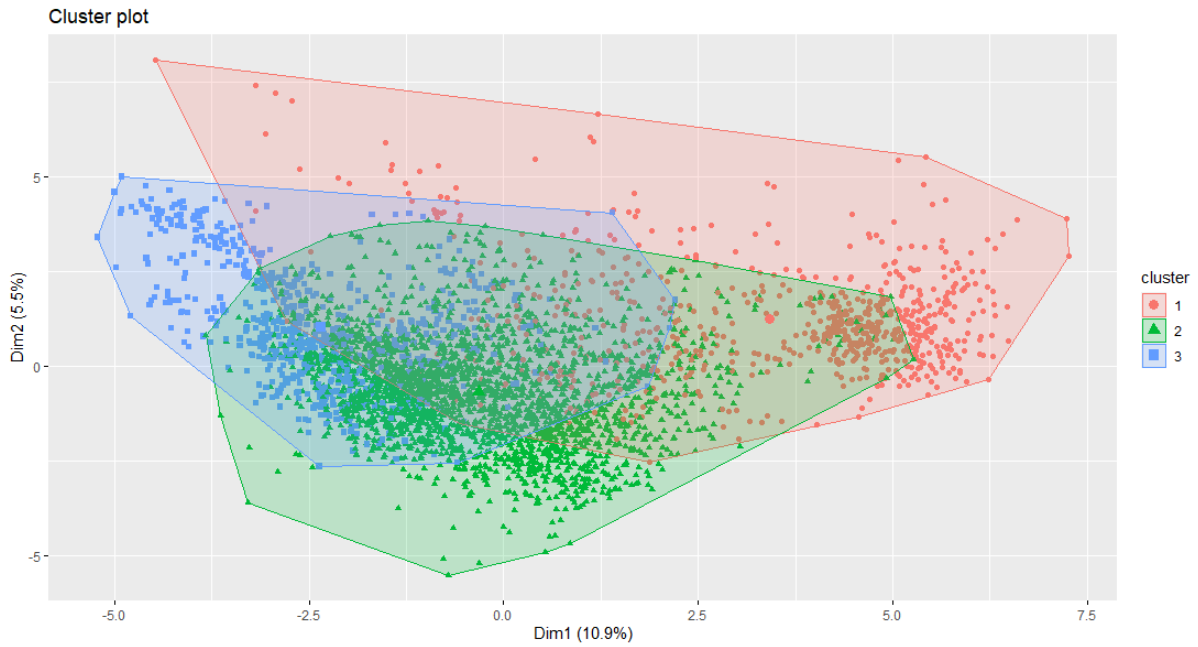



Figure 60: Clustering Results with PCA (k=2, k=3)

If we look at those plots, it seems that the hierarchical clustering works pretty well. We can see that every cluster really collects different data points. However we will have to make a deeper analysis to be sure that those clusters are correct, because the information shown by this PCA plot is the 15% of the total which is very low and we can't be sure that are good clusters even if the graph seems to look good.

### 4.1.5 CURE

Once we have done the hierarchical clustering, we will try with the CURE method. We will try with k=2 and k=3 as in the hierarchical clustering those were the optimal. The reduction actor will be r = 0.2 and we will apply the method on a sample of the 30% of our data.

First of all we will perform a hierarchical clustering on the sample of 30% of the data, to get the centroids that we will use in order to choose the most representative individuals.

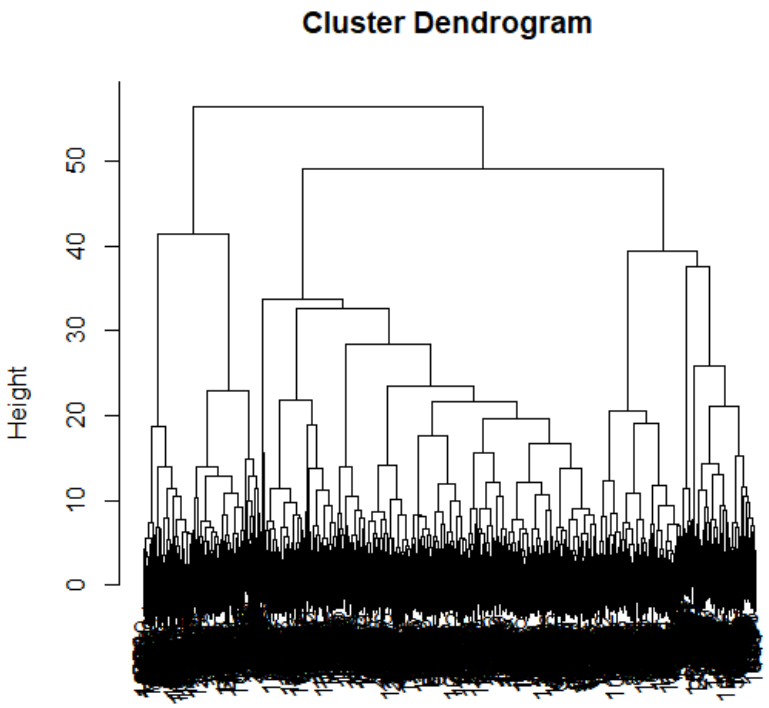



Figure 61: Cluster Dendrogram

This is the dendrogram after using the hierarchical clustering in our sample. As we see the dendrogram is not the same as before, but the optimal k is still 2 or 3.

Once we have this dendrogram we proceed with the calculus of the centroids of each cluster. After that, we will take the 20% most representative individuals (the closest to the centroid) of each cluster. Then we will recalculate the centroids based only on the representative individuals (to avoid outliers). Now, with those new centroids we will assign a cluster to the 70% of the database that was not used in the clustering process.

After all this process, the results of the clustering are the following:

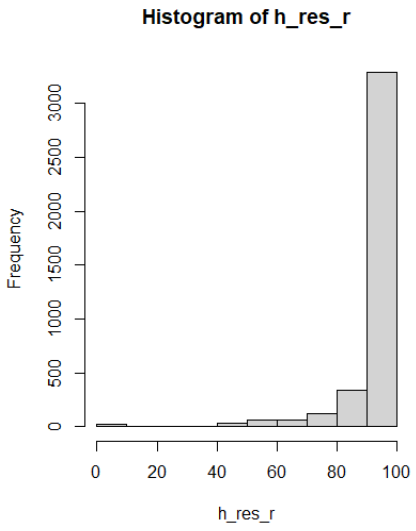

Figure 62: Clustering Results with PCA (k=2, k=3)

As we can see, these plots also seem to be pretty good. We can observe the differences between the individuals of each cluster. The results are not exactly the same as hierarchical clustering, but seem also good so we will analyze both methods deeper in the next section. In addition we can say that we expected good results in CURE too, after all, it also uses hierarchical clustering in a smaller sample. However this is a great method for large databases, which is not our case. That is the reason why we think that the classical hierarchical clustering with k = 3 will give better results when we do the advanced profiling.

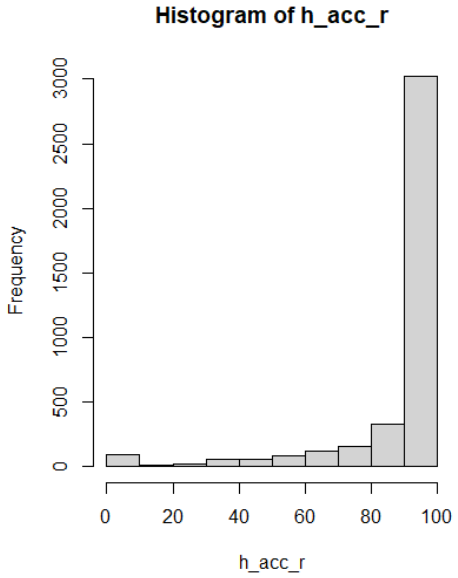
# 5. Advanced Profiling based on Hierarchical Clustering with k = 3

## 5.1 Distributions from variables and statistics

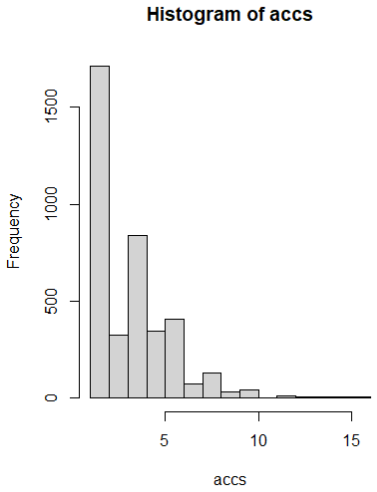
Numerical Variables:



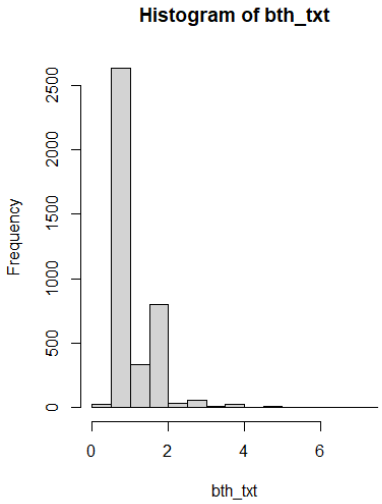

```
Summary statistics for h_res_r:
  Mean: 95.1451735495313
  Median: 100
  Minimum: 0
  Maximum: 100
  1st Quartile: 97
  3rd Quartile: 100
```



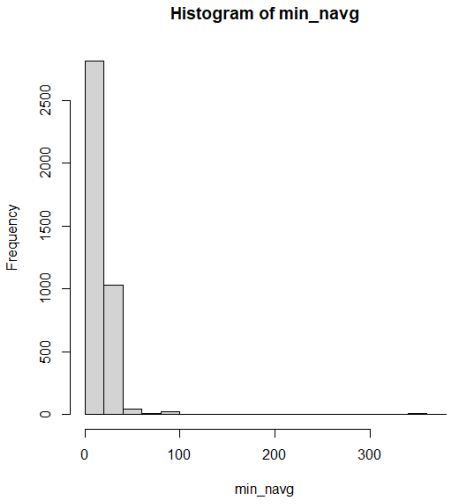

```
Summary statistics for h_acc_r:
  Mean: 90.0471243982772
  Median: 98
  Minimum: 0
  Maximum: 100
  1st Quartile: 92
  3rd Quartile: 100
```



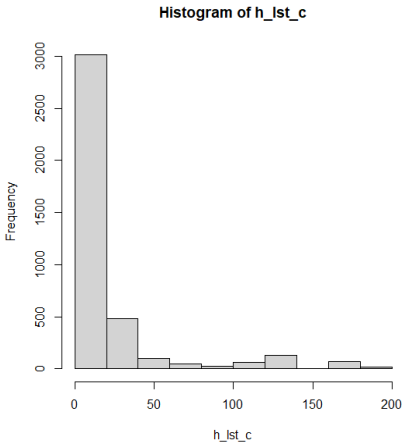

```
Summary statistics for accs:
  Mean: 3.6133772485432
  Median: 3
  Minimum: 1
  Maximum: 16
  1st Quartile: 2
  3rd Quartile: 5
```



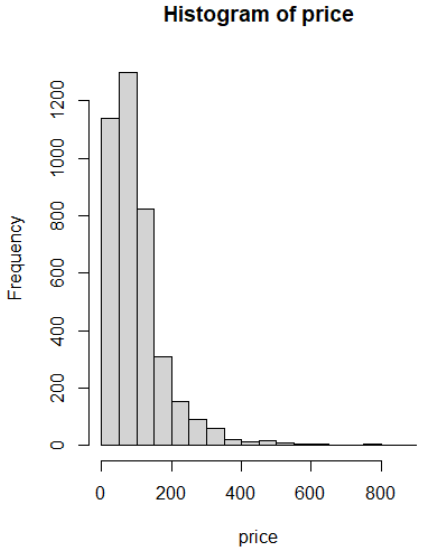

```
Summary statistics for bth_txt:
  Mean: 1.33139092982012
  Median: 1
  Minimum: 0
  Maximum: 7.5
  1st Quartile: 1
  3rd Quartile: 1.5
```



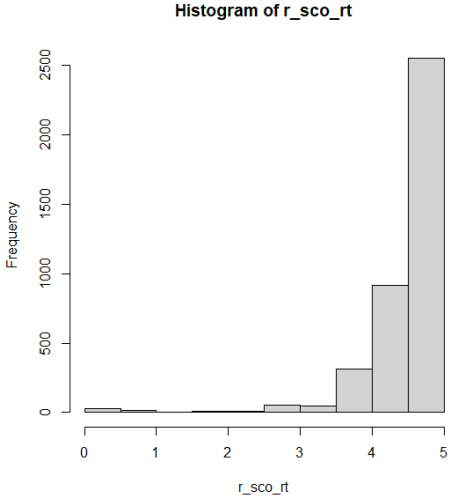

```
Summary statistics for min_navg:
  Mean: 13.2909804915125
  Median: 3.1
  Minimum: 1
  Maximum: 365
  1st Quartile: 2
  3rd Quartile: 30.85
```



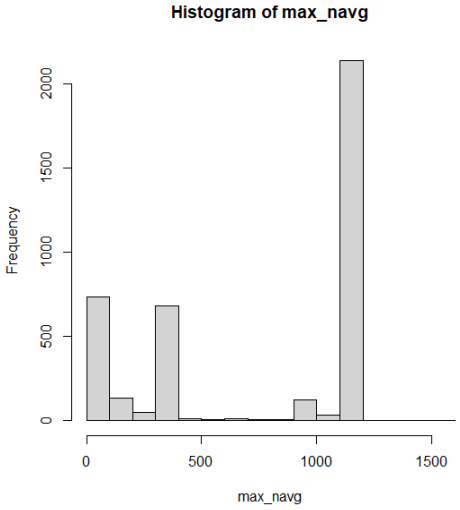

```
Summary statistics for h_lst_c:
  Mean: 19.7628578667342
  Median: 4
  Minimum: 1
  Maximum: 200
  1st Quartile: 1
  3rd Quartile: 20
```



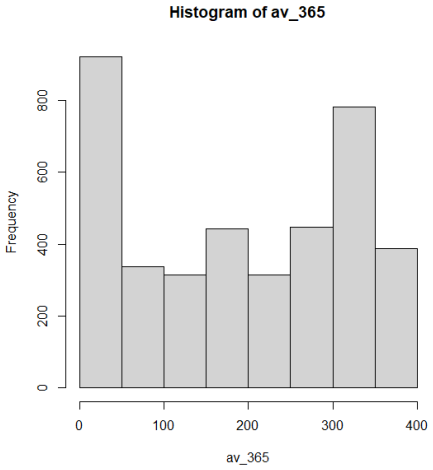

```
Summary statistics for price:
  Mean: 103.48493539397
  Median: 83
  Minimum: 10
  Maximum: 890
  1st Quartile: 47.5
  3rd Quartile: 128
```



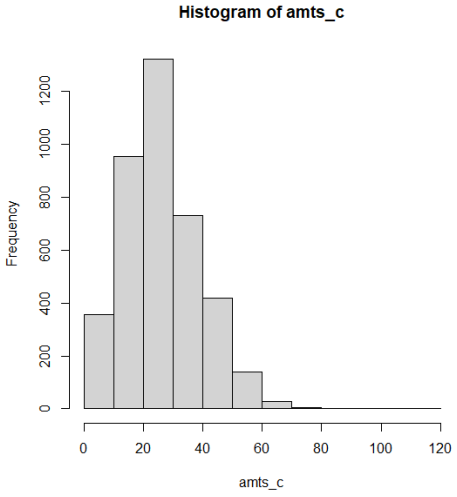

```
Summary statistics for r_sco_rt:
  Mean: 4.52990119077781
  Median: 4.67
  Minimum: 0
  Maximum: 5
  1st Quartile: 4.39
  3rd Quartile: 4.89
```



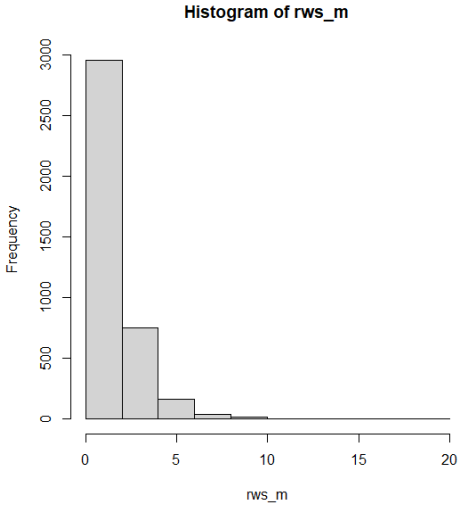

```
Summary statistics for max_navg:
  Mean: 734.583354446415
  Median: 1125
  Minimum: 1
  Maximum: 1533
  1st Quartile: 330
  3rd Quartile: 1125
```




```
Summary statistics for av_365:
  Mean: 188.17050924753
  Median: 195
  Minimum: 0
  Maximum: 365
  1st Quartile: 62
  3rd Quartile: 314
```

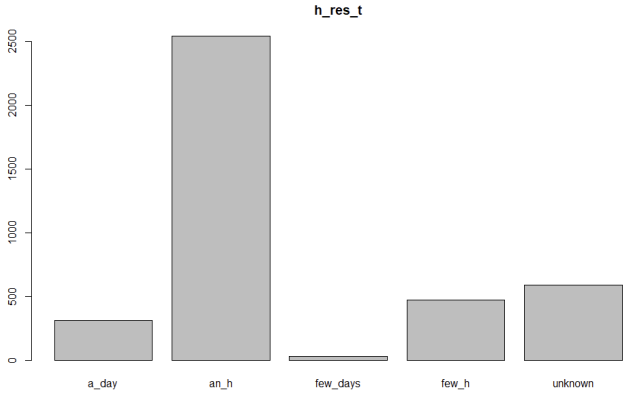


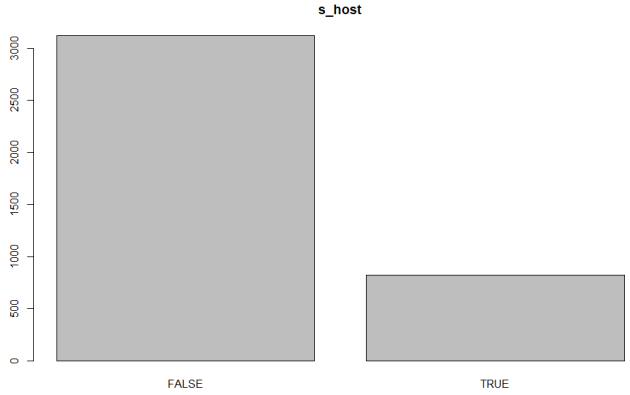

```
Summary statistics for amts_c:
  Mean: 26.649860653661
  Median: 26
  Minimum: 1
  Maximum: 111
  1st Quartile: 17
  3rd Quartile: 34
```

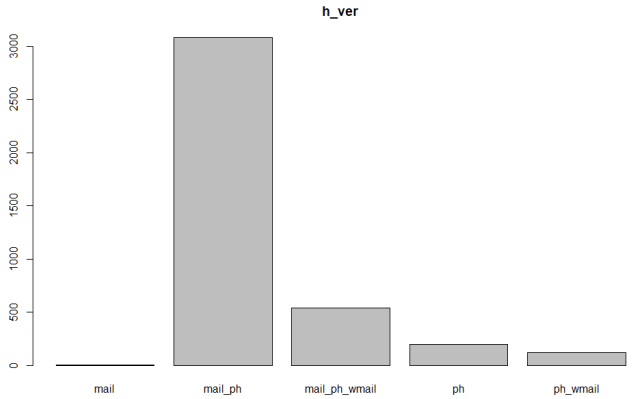


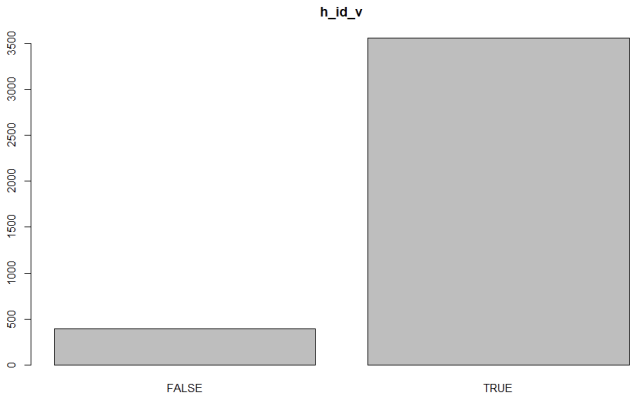

```
Summary statistics for rws_m:
  Mean: 1.3717050924753
  Median: 0.83
  Minimum: 0.01
  Maximum: 19.27
  1st Quartile: 0.23
  3rd Quartile: 2.015
```

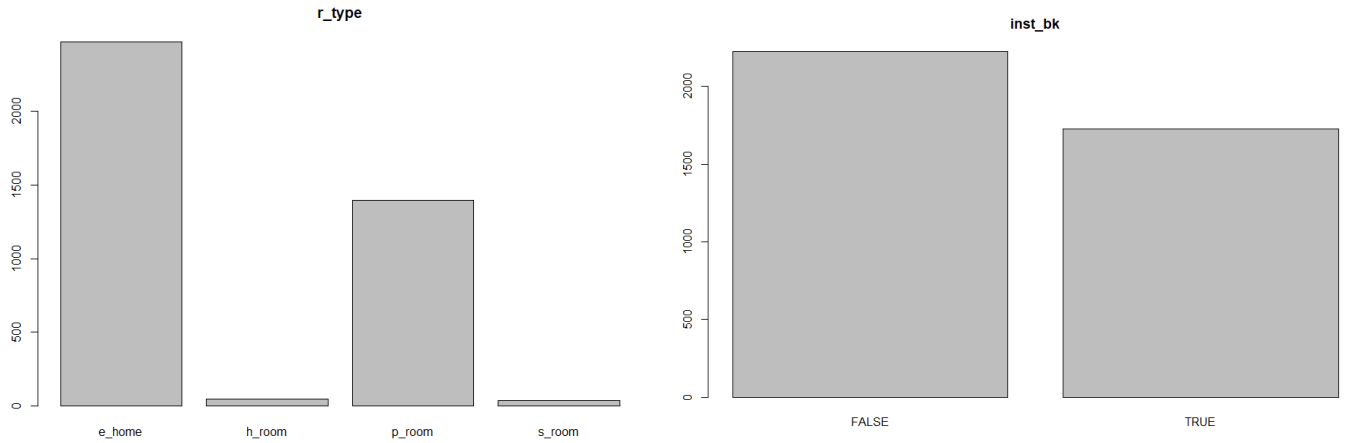
## Categorical Variables:






## 5.2 Interpretation of traffic light colors

In the previous plots and statistics, we can see the distributions and principal statistics for the numerical and categorical variables that we will take into account when doing the TLP and CGP. This will help us decide the color of the cell in both panels. To determine the color of each cell, we will look at whether the mean of the variable from the individuals in each cluster falls, under the mean, over the mean, or near the mean of the original distribution of the variable.

To color our cells, we will use three colors: green, yellow, and red. In green and red cases we will use lighter green and red to indicate that positiveness or negativeness is not very strong.

Green will be associated with a positive characteristic from the individuals in the cluster.
Yellow will be related to a neutral feature from the individuals in the cluster.
Red will represent a negative attribute from the cluster.
It is important to note that the criteria to determine negative or positive properties have been set by us based on our knowledge and experience.

**Positive and negative interpretation table (numerical variables):**

| **Variable** | **Low (under the mean)** | **High (over the mean)** |
|---|---|---|
| *h_res_r* | Negative | Positive |
| *h_acc_r* | Negative | Positive |
| *accs* | Negative | Positive |

| | | |
|---|---|---|
| *bth_txt* | Negative | Positive |
| *price* | Positive | Negative |
| *min_navg* | Positive | Negative |
| *max_navg* | Negative | Positive |
| *av_365* | Negative | Positive |
| *r_sco_rt* | Negative | Positive |
| *h_lst_c* | Negative | Positive |
| *rws_m* | Negative | Positive |
| *amts_c* | Negative | Positive |

We are now going to specify the interpretation that must be done of the coloring of our categorical variables cells. This explanation will be done differently than the one done with numerical variables due to the fact that categorical variables don't share any empiric characteristics between them that we can use to limit their positiveness and negativeness. In the coloring of numerical variables we used statistical information to establish quantity limits distinguishing different sections in the range of each variable, and we assigned colors to each section depending on the sensations that they transmitted. Then, looking at the mean of the members of each cell, we painted it according to the color of the section where the value was located in the original variable range. The interpretation of the categorical variables will be done individually because in this case we can't generalize by quartiles to decide the sense of the value, we must go deep into the modalities of each one and relate colors to the sensations that they produce. It's important to take into account that the definitive color of the cell is going to be decided by looking at the distribution of the original variable, which includes all the individuals of all clusters. The original distribution is going to be understood as the average value, considered neutral. When we observe the distribution of each cluster, we may see changes in the modalities predominance. The color association to modalities will permit us to see if the distribution has changed to better or worse conditions, and the sense of this change will be the one that decides the definitive color of the cell.

**Positive and negative interpretation of categorical variables:**

In the explanation of each variable, the colors mentioned before will be used to associate the sentiments that they represent to the different modalities of the variable. This will be done by highlighting the name of the modality with the corresponding color.

***h_res_t***

This variable shows the response delay of the host and it contains the following modalities:

- within an hour
- within a few hours

- within a day
- a few days or more
- unknown

Is evident that the lower the response delay is, the better are the communication conditions, the confidence and the implication that the host shows. It's important to highlight that the category unknown is interpreted negatively, because it doesn't show implication of the host and doesn't transmit confidence.

***s_host***
This binary variable shows if the host of the apartment is considered a super host or not. The unique modalities it has are:

- True
- False

If the host of the apartment is considered a super host, it denotes that he or she has experience with these deals, it has received good critiques and he has done his job well. Due to these reasons it transmits confidence to its clients.

***h_ver***
This variable shows a list of verifications that the host has. The things from which the host can be verified are email, phone, photographer and work_email. The modalities of this variable are subsets of these elements. In this case, to make the color assignment to the modalities, we won't base on which are the things the host is verified from, we will base on how many are the things from which the host is verified. Thi is a categorical variable that will be used as a numerical for its interpretation.

- 4
- 3
- 2
- 1

The more verifications the host has, the more is the confidence that he or she transmits.

***h_id_v***
This binary variable shows if the identity of the host is verified or not. the only modalities that it has are:

- True
- False

If the identity of the host is verified, it is a positive point for him, this fact shows that he has previous experience and it is recognized, so it transmits more confidence to the renters.

***r_type***
This variable identifies the type of room that we are renting. The different types of room that this variable talks about are:

- Entire place
- Private room
- Hotel room
- Shared room

This assignment of colors may be different from another perspective, because it depends so much on the desires and preferences of each person, so it's very subjective. The criteria used to classify the modalities in this way are the high valuation of exclusivity, luxury and privacy. Following these conditions, an entire place includes all the positive characteristics, being a private place with more space to be more independent and comfortable. Private rooms and hotel rooms provide privacy, but are not as exclusive as an entire apartment, and also don't provide the same independence and space. Finally, shared rooms don't provide any of the positive characteristics mentioned before.

***inst_bk***
This binary variable shows if the apartment is instant bookable or not. The only modalities that it has are:

- True
- False

If the renter can book the apartment without the approbation of the host, the rental conditions are better and faster.

## 5.3 Results: CPG and TLP

To interpret and perform our profiling, we will represent our data using CGP and TLP on our best clustering performed, hierarchical clustering with k = 3.

**CGP** is a way to represent our data in an easy and understandable way. It consists of creating a dataframe where each cell is the distribution of a variable from the individuals of a specific cluster. With this matrix, we are able to explore each variable in each cluster and define with colors (red, green, or yellow) whether the samples in this cluster have a negative or positive value for that variable. As a result, we can create a profile of each cluster more easily. In our analysis red implies negative connotation, green indicates positiveness and yellow represents neutrality.

**TLP** is a similar way of representing the clusters. As before, we have a table where rows are clusters and columns are variables. The difference is that now the columns are grouped into four categories: About Host, Apartment Characteristics, Reviews, and Apartment Rental Conditions. This way, we can profile our clusters using a different approach

## 5.3.1 CPG for numerical variables


*Figure 63: CGP from numerical variables*

## 5.3.2 CPG for categorical variables

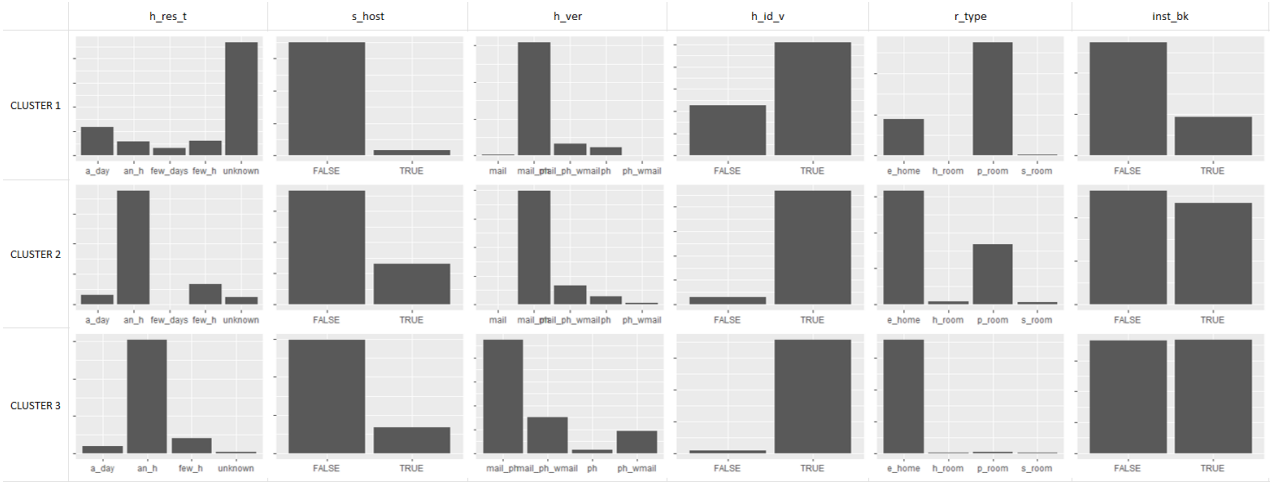



*Figure 64: CGP from categorical variables*

## 5.3.3 TLP

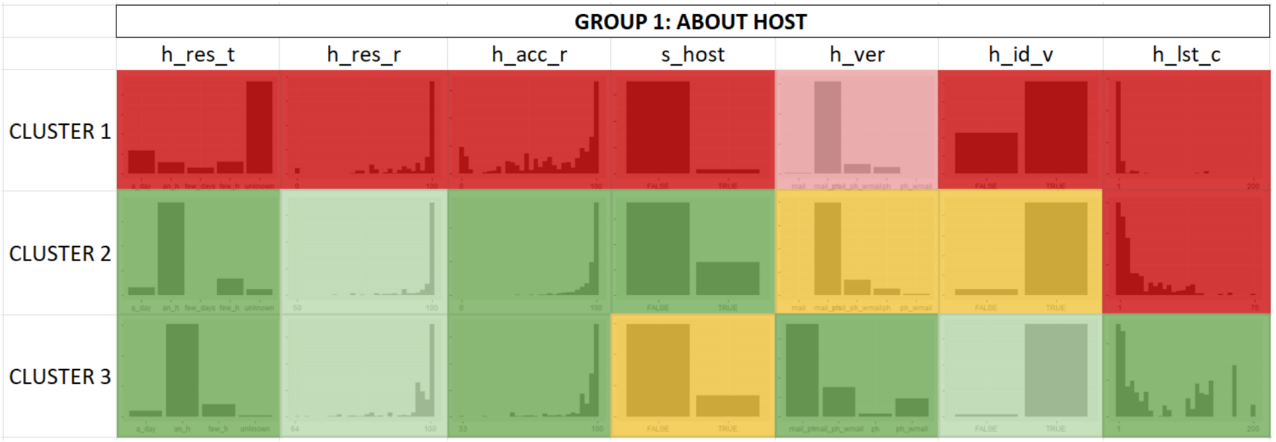



*Figure 65: TLP from group 1*

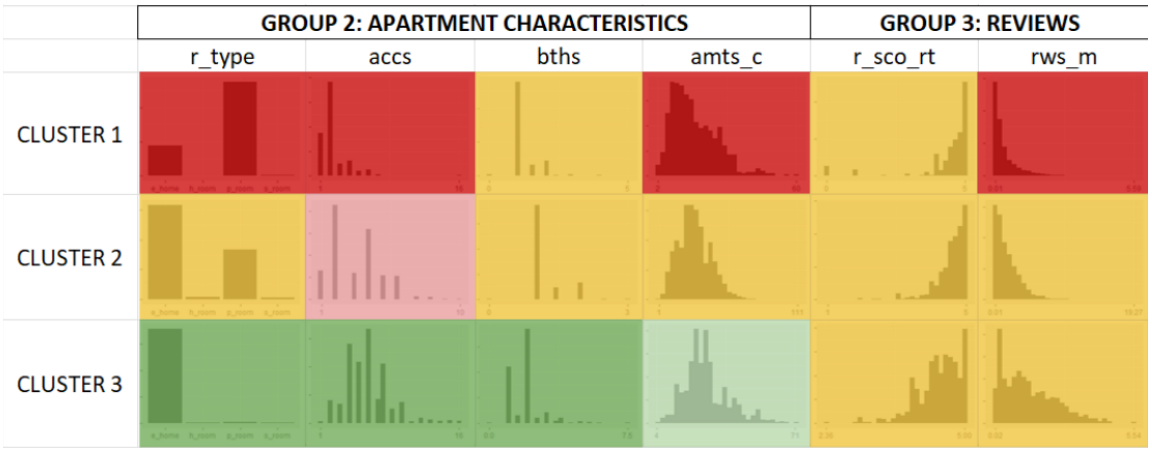



*Figure 66: TLP from group 2 & 3*

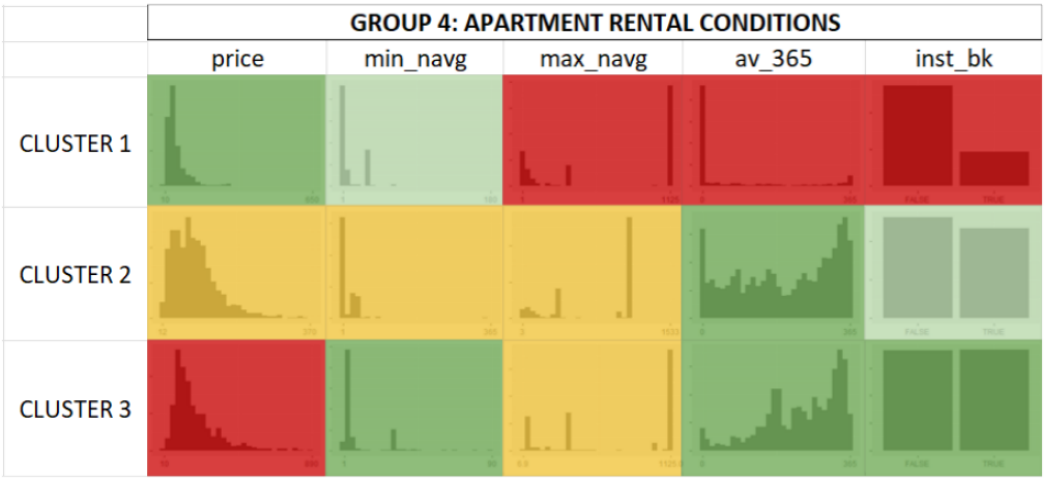



*Figure 67: TLP from group 4*

## 5.4 Conclusions

These previous graphs allow us to see differences between the clusters of our dataset. The colors provide a fast and visual analysis of the characteristics of each cluster and permit us to do a mental distinguish and classification. The TLP graph allows us to compare the clusters through the different aspects of an apartment by grouping variables that inform about the same aspect. In the following lines, we will highlight the characteristics of each cluster in every aspect of the apartment at the same time that we compare them with the characteristics of other clusters.

**Cluster 1**

In cluster 1, we can clearly see how the worst results were obtained in the four groups.

In group 1, as shown in the TLP plot, all variables related to host attendance have a negative connotation. For example, the h_res_t (host_response_time) variable's predominant class is "response time unknown," which clearly has a negative meaning when we talk about a host's attendance. Another example is the variable h_acc_r (host_acceptance_rate), which, in cluster 1, was clearly below the general mean. This implies that hosts in this cluster tend to accept fewer people in their apartments. The same negative behavior of the hosts is observed in the other variables of this group when we study the apartments from cluster 1.

In group 2, we have included all variables related to apartment characteristics, such as r_type (room type), accs (maximum number of accommodates (people) allowed), number of baths, and number of amenities (amts_c). In the room type variable, the number of entire apartments is reduced compared to the original mean, while the rent of private rooms increases. In the variables accs (accommodates) and amts_c (amenities count), the mean of the samples from cluster 1 falls below the original mean, implying that the facilities are worse. In the variable baths, the mean of cluster 1 is the same as the original mean.

In group 3, we can see variables related to reviews. We can observe that the score of the reviews is in the mean, but the number of reviews per month falls below the mean, which implies that fewer people tend to review these apartments, perhaps because they are rented less frequently.

Finally, in group 4, we find all variables related to the rental conditions of the apartment. Here, we can highlight how the mean price of apartments in this cluster is much lower than the original mean. We can also see that the availability (av_365) during the year is very low, that the minimum nights limit is lower than the mean, which is a good characteristic, and that the limit of maximum nights is also lower than the mean, which is a bad characteristic. In addition, the variable "instant bookable" has more false samples than the mean, which is also a bad characteristic.

We can conclude that cluster 1 consists of low-cost apartments where the host is less attentive, and the apartment has worse conditions and facilities.

**Cluster 2**

As shown in the CPG, cluster 2 is an average cluster. From the TLP, we can infer that apartments in this cluster have good host service. If we take a look at the variables such as host response time, host response rate, and host acceptance rate, we can see that they have positive values.

Regarding the topic of apartment conditions in group 2 and the topic of reviews in group 3, we can see that almost all variables in these groups are close to the original mean and distributions.

Finally, in group 4, we can observe that some rental conditions such as price, maximum nights, and minimum nights fall within the mean. However, conditions such as availability

during the year and whether the apartment is instant bookable have improved compared to the original distributions.

We can conclude that cluster 2 consists of average apartments in all groups except for host attendance, where we find good responses from the hosts.

**Cluster 3**

Cluster 3 is visually which has the better color combination in comparison to the other clusters. This is because this cluster englobes high-class apartments, most of which show a high number of facilities, a good host and flexible conditions in its renting. These factors contribute to rising prices.

In the host aspect, it's clearly distinguishable that this cluster predominates positively with respect to the others. The host profile transmits confidence and denotes that he has previous experience and he or she implies with the rental. The response and acceptance that he or she provides are better than the mean of all the hosts, in the same way that the verifications and the number of other rentals that he has.

Looking now at the apartment characteristics, it can be seen again that the positiveness is concentrated in this cluster again in comparison to the other ones. The type of apartments that it englobes is highly marked by the enormous predominance of entire apartments. We can say that this cluster is characterized by luxury and exclusiveness. Moreover, the facilities that these apartments provide are better than the facilities average of all apartments, in terms of amenities, baths, and maximum of people accepted. These conditions provide more comfort and are highly desired when searching for a rental.

Talking about the reviews, we can see that the apartments of this cluster have similar conditions to the average of all the apartments. In general, the three clusters obtained are very similar to the average, and in this terms are not so distinguishable.

Finally, looking at rental conditions, we can see general positiveness but with the exception of the price. The conditions that they provide show a high flexibility, facility and speedness of the rental added to the high disponibility of the apartment. These characteristics make the rental so much more comfortable in comparison to the conditions in other clusters. However, all the good characteristics that these apartments concentrate have a notable influence in their prices, which is classified as red in the TLP. The luxury, exclusiveness and facilities that they provide accord to the price that they are valued with.

In conclusion, cluster 3 consists of high-class expensive renting apartments with an attentive and experienced host, high-quality conditions and a high number of facilities.

# 6. Time series clustering

From now on, we will focus on the special data on our dataset, such as temporal, textual or geospatial data. We will first work with the temporal data, but unlike the last subject (ME) we will be doing a clustering approach to the time series.

To make a clustering with the time series we need several time series, which our dataset doesn't have. The time series we were using showed the number of reviews per month. With this information we can get a good value about the use of airbnb in Barcelona over time.

Knowing that, we came up with a great data to analyze with the time series clustering. We will be getting this data from different cities around the world. Doing this we will have multiple time series to analyze and to clusterize in order to group the most similar cities. Grouping big and popular cities or maybe different countries or continents.

The cities we selected are the following. We tried to maintain a balance between continents in order to analyze if there are relations between continents too:

1. Barcelona
2. Berlin
3. Buenos Aires
4. Cape Town
5. Hong Kong
6. Istanbul
7. Los Angeles
8. Mexico
9. New York
10. Rio de Janeiro
11. Rome
12. Sydney
13. Taipei
14. Tokyo


Figure 68: Map with the cities selected for the Time Series Clustering

As we see, we have chosen cities from every continent, but we had some troubles about some countries. The African Continent only had data about Cape Town, which is the one we picked as a representative of Africa, but it may not be a good representative because it is a very touristic city. Besides, we also had problems particularly with China, because other cities such as Shanghai or Beijing appeared on the web to download the data, but the databases were empty.

Once we selected the cities to make the analysis we plotted the time series before clustering them, but we needed all of them to have the same length. For this reason we had to do a little preprocessing:

- Erase the last row of the dataset. That is because the data it's not scrapped at the end of the month, and also not all the cities have been scrapped on the same date. Eliminating the last month we get rid of the bad values got from the scrapping.

- Substitute the NA for 0. The time series don't have the same length, so NA were imputed when a time series didn't have a value for a particular month. Those NA only appeared at the beginning of the columns, meaning that represented 0 reviews on those months, so we decided to impute them as 0.

After doing this preprocessing, the different plots looked like this:

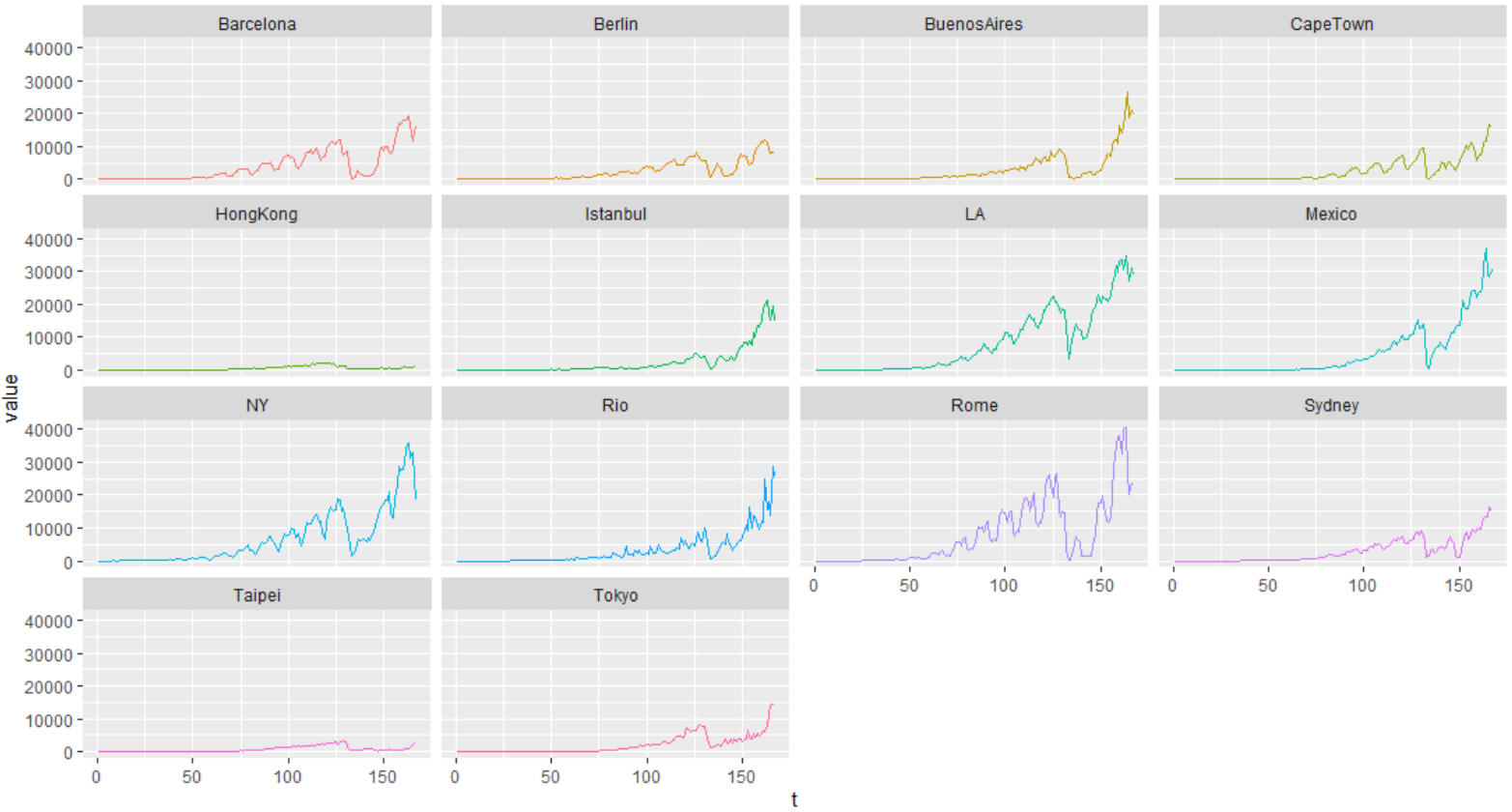



Figure 69: Plot of the time series of the different cities

Seeing the plot of all the countries we can get some hypothesis or preprocessing to do before strating with the clustering:

1. COVID-19 affected, to a greater or lesser extent, every country. We can see that because all the plots have a big fall in the number of reviews at the same time (March 2020). The covid altered the time series of almost every data. For this reason we will be analyzing the data in two different parts: Pre-Covid and Post-Covid.

2. We can see some plots that are very similar among them, like Hong Kong and Taipei or Tokyo and Sydney. Seeing this, we could expect the clustering to give good results joining those countries together.

Before doing the clustering we need to separate the data in two epochs, as we commented before. To do this we had to make a decision to set the break point. In fact, we actually will split the data in three parts: Pre-Covid, Covid and Post-Covid, but we will not analyze the Covid one, because the data is not stable because of the different government measures applied on this period or the well-known infection waves.

The decision of setting which period was the covid was not easy, because we cities from all around the world. And the pandemic did not start in all the world at the same time, so we decided to take the time point where all the countries started to get a lot less reviews. This date is March 2020 as we can observe in the following extract from the database:

| Date | Barcelona | Berlin | BuenosAires | CapeTown | HongKong | Istanbul | LA | Mexico | NY | Rio |
|---|---|---|---|---|---|---|---|---|---|---|
| 2019-12 | 7260 | 5746 | 8951 | 8138 | 1081 | 3587 | 17182 | 12777 | 16138 | 5733 |
| 2020-01 | 8108 | 5544 | 8242 | 9129 | 1135 | 4124 | 18749 | 12770 | 11518 | 10124 |
| 2020-02 | 8628 | 5702 | 7264 | 9381 | 503 | 3556 | 18245 | 13937 | 9904 | 8814 |
| 2020-03 | 4920 | 3454 | 5570 | 7000 | 377 | 2611 | 12995 | 11356 | 7409 | 5261 |
| 2020-04 | 195 | 399 | 524 | 359 | 345 | 266 | 3390 | 1842 | 1406 | 651 |
| 2020-05 | 227 | 814 | 449 | 293 | 351 | 274 | 6508 | 520 | 2061 | 635 |
| 2020-06 | 571 | 2024 | 428 | 595 | 376 | 739 | 9102 | 2508 | 3227 | 1042 |

Figure 70: Subset of the database showing the March 2020 review fall.

On the other hand, setting the date to end the Covid period was trickier, because some countries recovered earlier than others, or there were more falls due to the covid state in the countries. After some time, we selected February 2021, because from that day on, almost all the countries started to increase the number of reviews, but the difference wasn't as clear as the start date, as we can observe:

| Date | Barcelona | Berlin | BuenosAires | CapeTown | HongKong | Istanbul | LA | Mexico | NY | Rio |
|---|---|---|---|---|---|---|---|---|---|---|
| 2020-10 | 1482 | 3134 | 597 | 2875 | 502 | 4152 | 12385 | 7353 | 6697 | 4779 |
| 2020-11 | 825 | 1408 | 1125 | 3372 | 395 | 3729 | 12034 | 8582 | 6398 | 5602 |
| 2020-12 | 1021 | 1107 | 1812 | 5119 | 601 | 2884 | 9250 | 7661 | 5736 | 4980 |
| 2021-01 | 1149 | 970 | 2021 | 4825 | 488 | 3105 | 9606 | 6805 | 6846 | 8237 |
| 2021-02 | 1136 | 1020 | 1934 | 3177 | 481 | 2920 | 10254 | 6515 | 6255 | 5721 |
| 2021-03 | 1522 | 1510 | 2268 | 4641 | 355 | 4051 | 13568 | 8373 | 7805 | 4461 |
| 2021-04 | 1902 | 1614 | 2074 | 5193 | 482 | 3893 | 15625 | 9593 | 9218 | 3075 |
| 2021-05 | 3395 | 2426 | 1397 | 4313 | 443 | 2793 | 18271 | 11549 | 12286 | 3906 |
| 2021-06 | 4619 | 3256 | 1404 | 3453 | 427 | 4125 | 19276 | 11643 | 13355 | 4941 |
| 2021-07 | 8103 | 5450 | 2119 | 2791 | 563 | 5728 | 21162 | 13234 | 15199 | 5603 |
| 2021-08 | 9994 | 7667 | 2228 | 3909 | 724 | 7283 | 22795 | 13442 | 16838 | 6008 |
| 2021-09 | 8482 | 6792 | 2523 | 5048 | 523 | 7309 | 20172 | 13368 | 17088 | 7492 |
| 2021-10 | 9837 | 7456 | 3887 | 6932 | 510 | 8457 | 22576 | 16013 | 18395 | 8428 |

Figure 71: Subset of the database showing the February 2021 gradually increase.

## 6.1 Pre-Covid Clustering

We will start by clustering the time series before covid. First of all, let's see the time series plot before Covid:

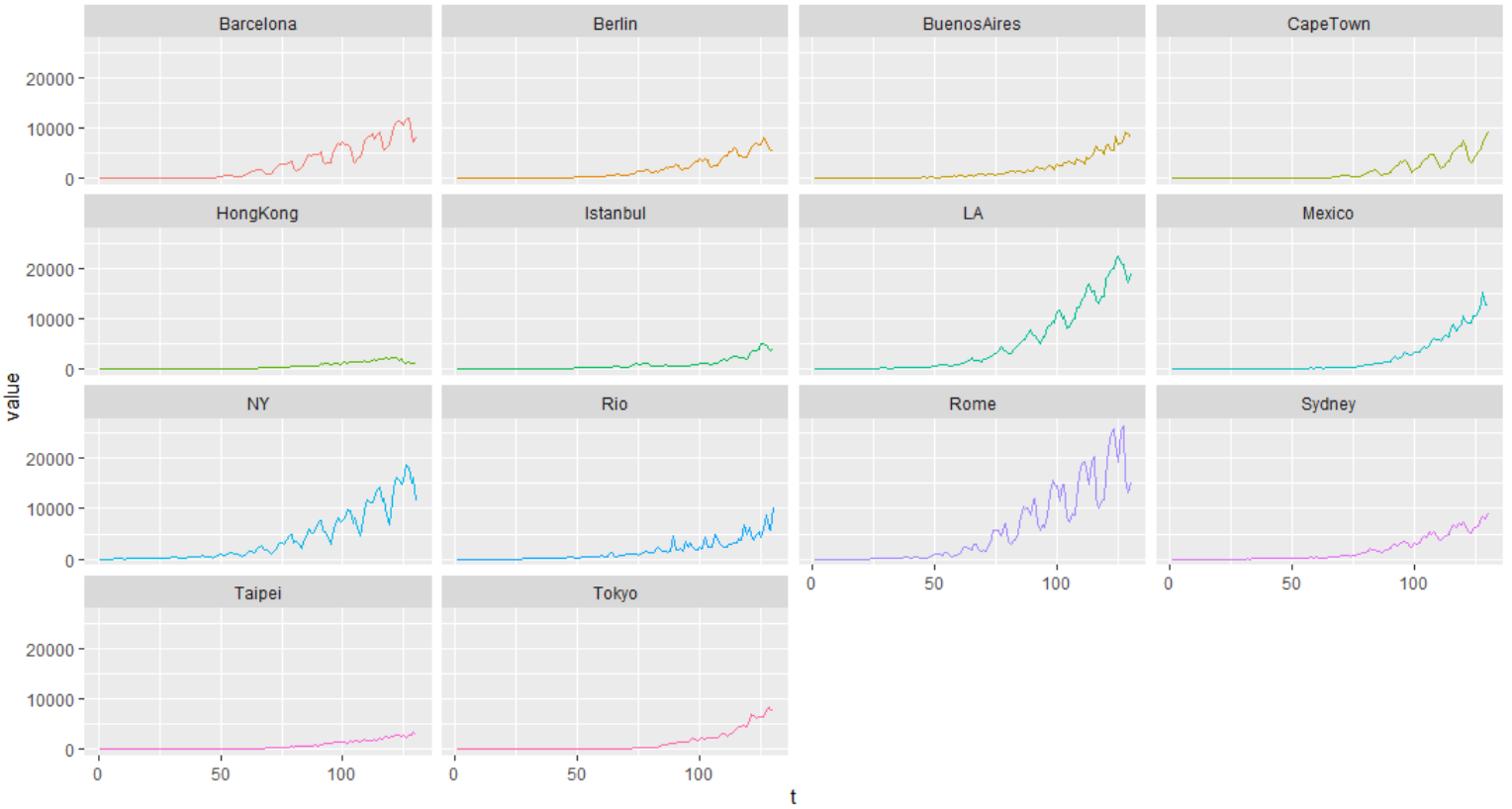



Figure 72: Plot of the time series of the different cities before Covid.

Performing a visual analysis of the plots we can see several things:

1. All the charts have an ascendant tendency, with more or less slope. For example, NY, Rome and LA have a very tilted slope, while Hong Kong, Taipei and Istanbul have a very soft slope.

2. We can see a very clear seasonality in some charts, where the falls coincide with the low-season, were people don't travel, while the peaks represent the high-season.

3. Airbnb didn't start at the same time in all the cities. In NY or Rome, the series starts to increase very soon, while in places like Mexico or Tokyo starts later.

Once said that. Let's dive into the clustering. We will be doing hierarchical clustering using the complete method. The distance method we will be using is "DTW" (Dynamic Time Warping), which is a very used distance to compare time series, it also work for time series of different length, but in out case, all the series have the same length, so we could have also used other distances like the "cosine" one.

Once we did the clustering, the dendrogram looked like this:

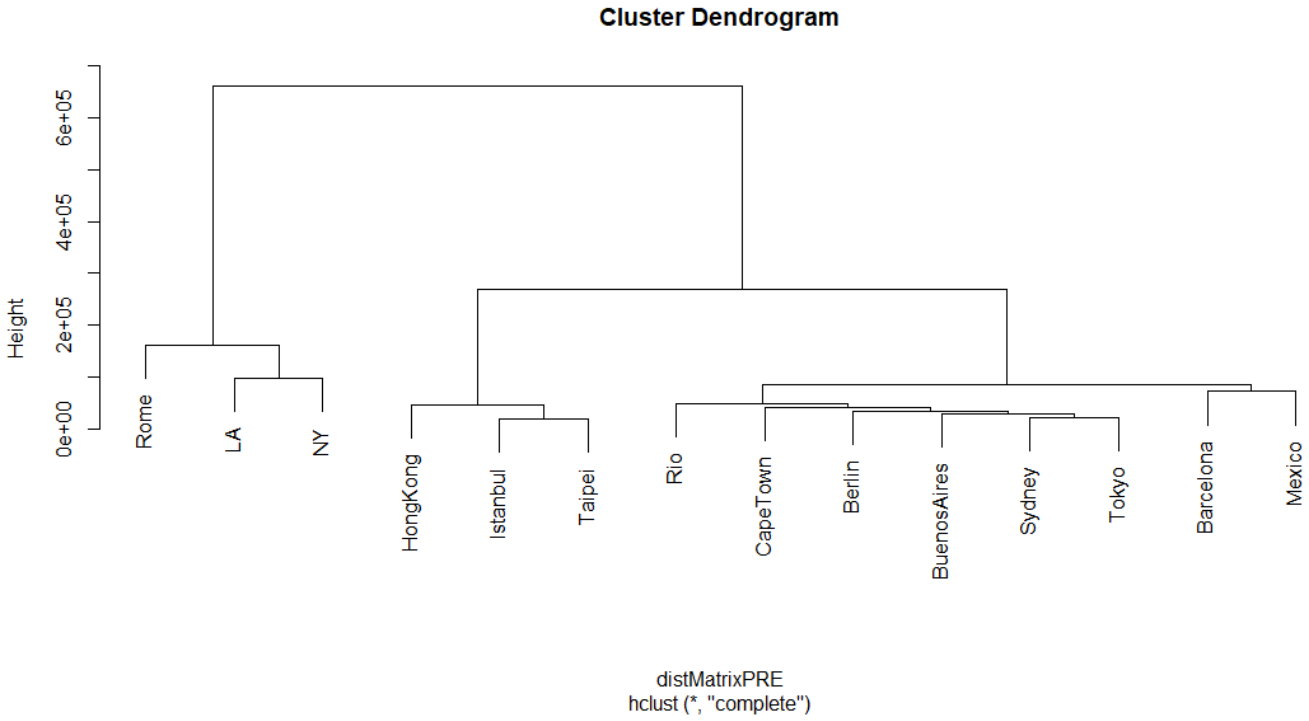



Figure 73: Cluster Dendrogram before Covid

Looking at the clustering dendrogram we can get interesting information about the relation of the series. Looking at the distances the optimal k for this cluster should be 2. However we think that a most appropriate k, taking into account the groups formed is 3. So now let's analyze the 3 clusters:

1. **Big popular cities:** In this first cluster we have New York, Los Angeles and Rome. Those are the cities with the highest number of reviews of all the series. It makes sense because those are one of the most popular cities in the world and also big cities. In addition, Airbnb was

founded in the United States, so it makes sense that the most famous cities of the country are the ones with the biggest number of reviews.

2. **Average Cities:** In this group we get the cities that have a considerable amount of reviews but not as many as the ones in the cluster one. We also have to say that in this cluster we can observe two distinguished groups. On the one hand, Barcelona and Mexico. If we look at the plots in the figure 72, we can see that they have a great amount of reviews which makes them really similar. On the other hand, we have cities like: Rio de Janeiro, Cape Town, Berlin, Buenos Aires, Sydney or Tokyo. Despite the fact that these cities are also important and touristic the usage of airbnb is not so high, we also can see that Tokyo and Sydney are very similar according to the dendrogram and as we stated before looking at the figure 72 plots.

3. **Least Popular cities:** The last cluster represents the cities with the lowest number of reviews and the least tendency slope. Those cities are Hong Kong, Taipei and Istanbul. Hong Kong and Taipei are two Asian cities, where the culture is a bit different than in the rest of the world, and maybe Airbnb is not a very used application there. However, Japan had bigger numbers despite the fact that it is an Asian city too, but probably the culture and tourism there makes Airbnb still a decent application to rent apartments. Besides, Istanbul is a city on the border of both continents, and had a big tourism increase in the last few years. However, it seems that before covid Airbnb was not in use.

Those are the clustering results before Covid-19, now let's plot the time series separated by colors to visually represent the clustering results:

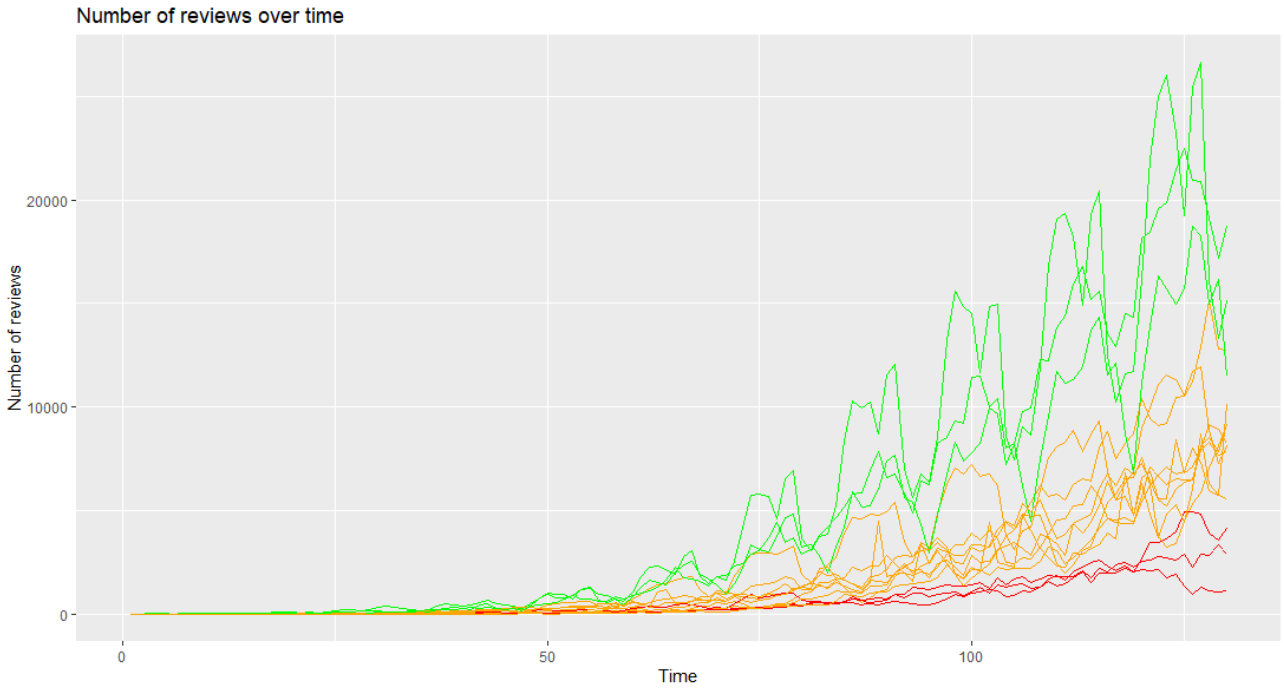



Figure 74: Plot of the different time series clustered. Cluster1 = Green, Cluster2 = Orange, Cluster3 = Red

## 6.2 Post-Covid Clustering

Once we have seen how the different time series are clustered before covid, let's see if the covid has changed how the cities are clustered together or if the cluster still looks the same after the pandemic. First of all let's check the plots of the different cities:


Figure 75: Plot of the time series of the different cities after Covid.

Here we can see some differences with the time series before covid. First of all, the shape is not the same as before, this is due to the fact that this time series are only 2 years long so we have only 25 observations, which is pretty low to start getting patterns. Nevertheless, we can start to see some seasonality in the majority of countries. For example in Barcelona we can see two peaks and two falls which represent the two summers and two winters of 2021 and 2022. On the other hand we have Hong Kong or Taipei which have not recovered from the damage caused by the pandemic.

Once we have take a first look at the plots, we can start performing the hierarchical clustering to see if the cluster members have changed:

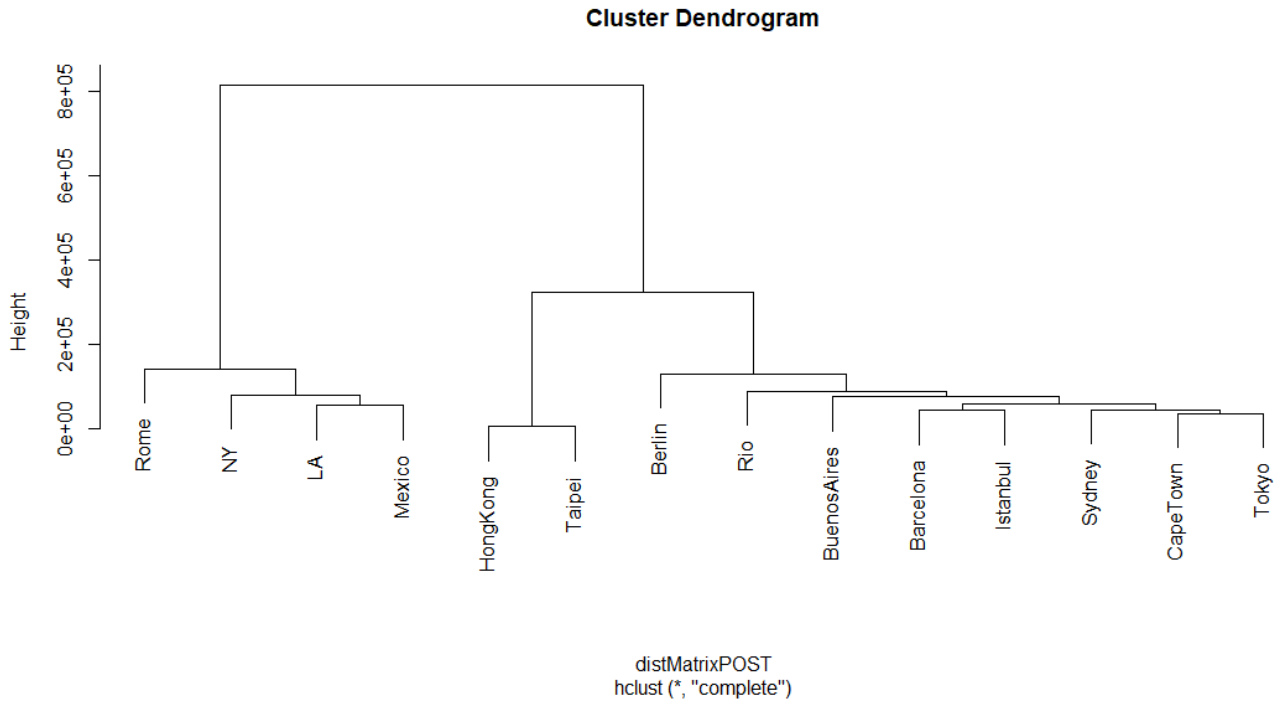



Figure 76: Cluster Dendrogram after Covid

As we can see, the cluster distribution is very similar to the ones before covid. However there are some changes in the members of every cluster. As with the pre-covid dendrogram, the optimal k should be 2, but we think that k=3 is a better approach to extract useful information and knowledge.

1. **Big popular cities:** In this cluster we still have the members that were here before the covid. This means that even with the damages caused by the covid, those cities still have a lot of tourism, and looking at the figure 72 and 75 we can see that the number of reviews after covid has surpassed by far the numbers before the pandemic. Furthermore, there is another city that joined this cluster, Mexico. We can see that before the covid it was having an exponential growth, that has not been stopped by the pandemic, taking the city to one of the most popular cities in Airbnb.

2. **Average Cities:** This group has remained the same, with the addition of Istanbul, which has gained popularity in the last few years, becoming one of the most popular cities of this cluster.

3. **Least Popular cities:** This group remains the same except for Istanbul that we already commented on. However, these two cities remaining have a characteristic that the other cities don't have. They are the only two cities that have not recovered from the pandemic. The rest of the cities have reached higher values than before the pandemic, while Hong Kong and Taipei remain practically flat. We think that this may be due to the strong policy against covid in those countries, where the zero-Covid policy was still up at the end of 2022.

Those are the clustering results after Covid-19, now let’s plot the time series separated by colors to visually represent the clustering results:

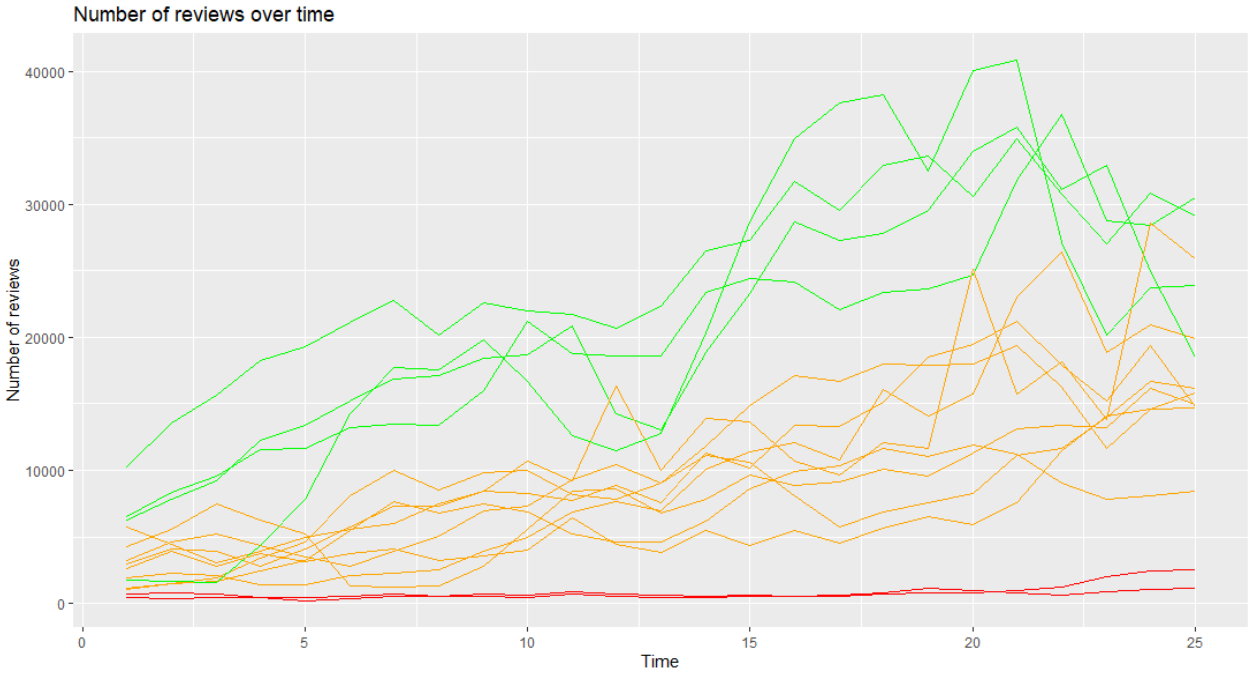



Figure 77: Plot of the different time series clustered. Cluster1 = Green, Cluster2 = Orange, Cluster3 = Red

We can visualize the clusters clearly separated, especially the Hong Kong and Taipei lines, that remain with very low reviews until the end of 2022.

# 7. Clustering apartment descriptions

## 7.1 Introduction

In this section we applied text mining knowledge to classify Airbnb apartment descriptions. The main objective of this was to find different themes of what the costumers talk about when describing an apartment and group the descriptions following these themes.

## 7.2 Preprocessing

It must be highlighted that we only worked with descriptions written in English, in this way we will only focus in one language, setting an equality environment among the descriptions.

The first step that we must follow when working with text is the preprocessing of it. This is essential to guarantee a good processing of the texts and obtain coherent results. Now we are going to see the different preprocessing steps that have been done and how they transformed the text we are working with.

The raw description we are going to take as an example is the following one:

```
[2] "Hostel One Ramblas has a great atmosphere and is an easy place to meet other people. We offer a wide range of tours and special events throughout Barcelona, as well as free entrance to the best clubs in Barcelona and amazing free dinners everynight. So what are you waiting for! Book now and become apart of the Hostel One Family!<br /><br /><b>License number</b><br />AJ-000563"
```

As we can see, there are some terms and symbols that only apport irregularity, noise, and complexity to the text.

- Elimination of <br/> symbols or similar, and the context of license numbers.

  ```
  [2] "Hostel One Ramblas has a great atmosphere and is an easy place to meet other people. We offer a wide range of tours and special events throughout Barcelona, as well as free entrance to the best clubs in Barcelona and amazing free dinners everynight. So what are you waiting for! Book now and become apart of the Hostel One Family!"
  ```

  The < … > symbols didn't have any semantic meaning and License Numbers didn't apport any significant difference between descriptions.

- Lower all the capital letters in the text.

  ```
  [2] "hostel one ramblas has a great atmosphere and is an easy place to meet other people. we offer a wide range of tours and special events throughout barcelona, as well as free entrance to the best clubs in barcelona and amazing free dinners everynight. so what are you waiting for! book now and become apart of the hostel one family!"
  ```

  With this step we provide uniformity to the text and we ease the following processing of it.

- Remove stopwords

  ```
  [2] "hostel one ramblas  great atmosphereeasy place  meet  people. offer  wide range  tours  special events throughout barcelona,  well free  entrance  best  clubs  barcelona  amazing  free  dinners everynight. waiting ! book now  become apart  hostel one family!"
  ```

  This step removes all the 'empty' words, all those words that have no relevant meaning to develop the analysis. We can say that with this step we keep the key words, those that apport meaning and characteristics to the text.

- Remove punctuation

  ```
  [2] "hostel one ramblas  great atmosphereeasy place  meet  people offer  wide range  tours  special events throughout barcelona  well free  entrance  best clubs  barcelona  amazing free dinners everynight  waiting  book now  become apart  hostel one family”
  ```

  Punctuation doesn’t apport any characterization to the text.

- Remove numbers

  ```
  [2] "hostel one ramblas  great atmosphereeasy place  meet  people offer  wide range  tours  special events throughout barcelona  well free  entrance  best clubs  barcelona  amazing free dinners everynight  waiting  book now  become apart  hostel one family"
  ```

  It’s a similar case as the punctuation, the numbers don’t highlight any important characteristic of the text.

- Remove excessive empty spaces

  ```
  [2] "hostel one ramblas great atmosphere easy place meet people offer wide range tours special events throughout barcelona well free entrance best clubs barcelona amazing free dinners everynight waiting book now become apart hostel one family"
  ```

  This step apports uniformity to the text and a more aesthetic appearance.

- Remove general and highly repeated words

  ```
  [2] "hostel one ramblas great atmosphere easy place meet people offer wide range tours special events throughout  well free entrance best clubs  amazing free dinners everynight waiting book now become apart hostel one family"
  ```

  The words that are highly repeated are not good to distinguish the descriptions later, as they will appear with a high frequency in all of them, not allowing us to see the main theme that the groups of descriptions talk about. The words that we eliminated were apartment, Barcelona and room. They all represent general characteristics of the apartments that all descriptions could mention, not apporting distinctive character among descriptions themes.

## 7.3 Clustering

Once we preprocessed all the text, leaving it prepared to be processed and workable with, we started with the clustering of the descriptions.

A key element that helped us with the clustering was the Document Term Matrix. This element consists of a matrix of m·n dimension where m is the number of documents (descriptions in our case) and n is the number of different words that we can observe in the set of all descriptions. In conclusion, we have many rows as many descriptions we have and many columns as many different words are in all descriptions. The values of the matrix are integers that represent how many times a certain description contain a certain word. We must consider that not all word present in this matrix were relevant in the analysis we did. To depurate the matrix and guarantee better results, removed the less frequent words to focus only on those that have real impact in the descriptions.

After this depuration, we calculated a distance matrix and applied a hierarchical clustering to agrupate the descriptions following its similarity. The dendrogram obtained was the following one:

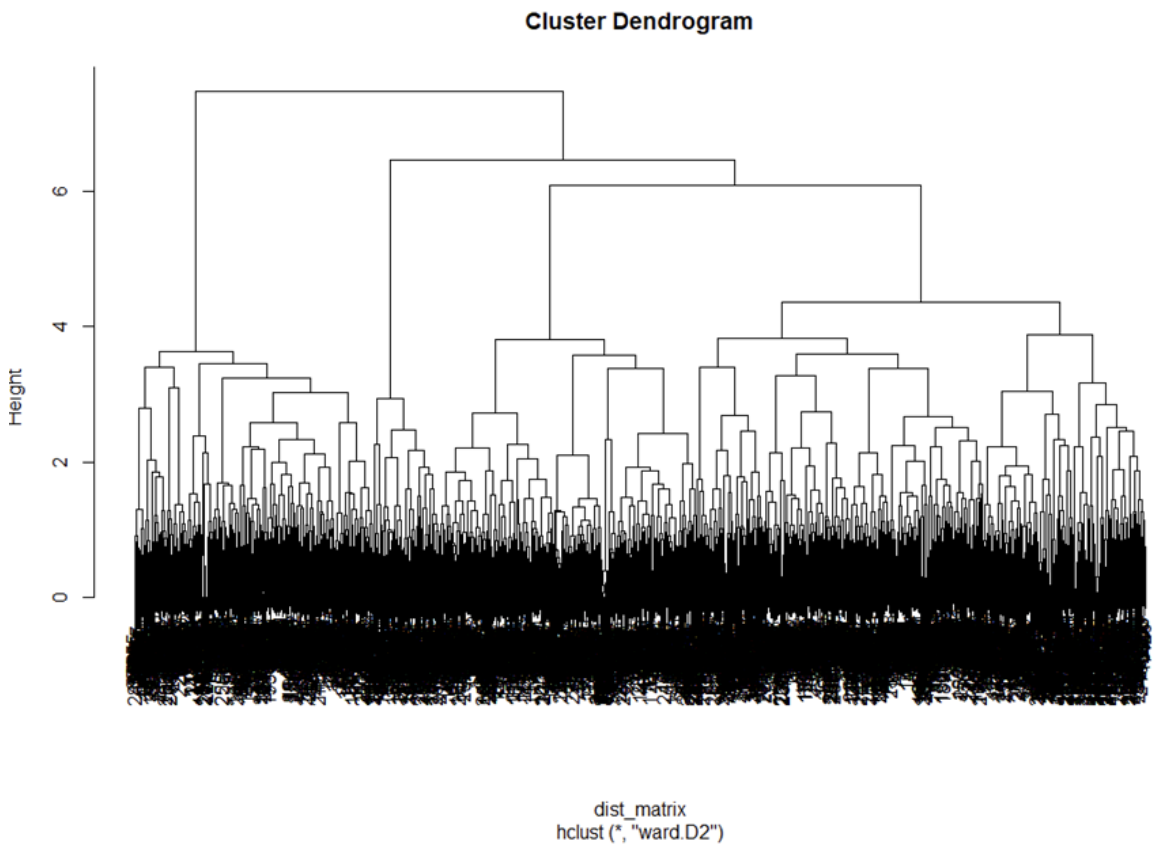



Figure 78: Dendrogram of the clustering of the Airbnb apartments descriptions

As we can see, k = 3 was a good value to cut the dendrogram and make a good classification of the descriptions.

The only thing that left to do was to observe the characteristics that presented the groups of descriptions we created trying to find some patterns that allowed us to see how the classification had been done. To do this, we splitted the Document Term Matrix in three subgroups, observing the cluster where the description was classified and calculating the most frequent terms that they presented. The results were appreciated by creating wordclouds.

**Cluster 1**


Figure 78: cluster 1 of descriptions

**Cluster 2**

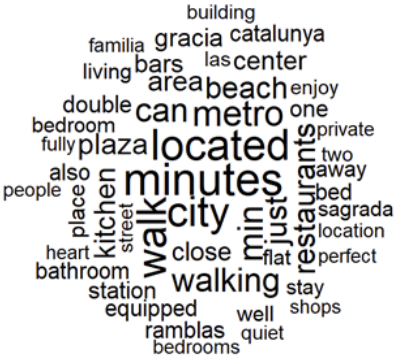



Figure 79: cluster 2 of descriptions

**Cluster 3**

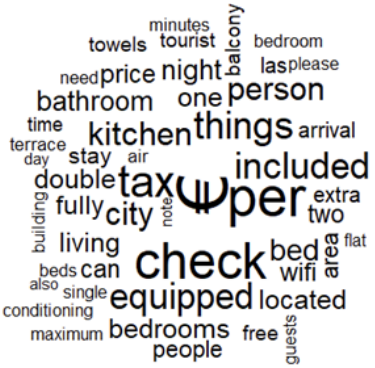



Figure 80: cluster 3 of descriptions

As we can see, the three clusters obtained show different most used words. Observing the characteristics of each wordcloud, we can set the following hypothesis:

- Cluster 1 → includes descriptions which main theme is the accessories, facilities and physical characteristics that the apartment has.
- Cluster 2 → includes descriptions which main theme is the location of the apartment, mentioning also closeness of the apartment to determined places, times and methods of displacement.
- Cluster 3 → includes descriptions which main theme is the price of the apartment and other economic related aspects.

In conclusion, in this section we observed how we can apply text mining knowledge to cluster different descriptions of the apartments that we have in our Airbnb dataset. We also understood in a better way the Document Term Matrix and its working. Finally, we appreciated three different groups of descriptions in our dataset. They talk about the facilities of the apartment, location of the apartment and prices of the apartment respectively.

# 8. Sentiment analysis

Our next step in the project is sentiment analysis, where we analyzed all the reviews of the apartments in the dataset to extract conclusions about the negativity and positivity. First of all, we had to prepare the data. Since each apartment has multiple reviews and each review is a different row in the dataset, we grouped all the reviews of the same apartment. We only kept English reviews, as they were the majority, and omitted rows where we had NA's in the reviews, as well as missing and outlier data.

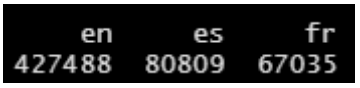



Figure 81: top 3 idioms in our reviews

Once we have all the reviews grouped by the listing ID, it's time to preprocess the text. We followed the same steps as in the description clustering, but this time we added lemmatization and performed all the steps on a corpus. First, we created a corpus to collect all the reviews, enabling us to conduct additional analysis beyond sentiment analysis, such as topic modeling. Then, we converted all the letters to lowercase, removed numbers and punctuation marks, as well as stop words. Finally, we applied lemmatization to reduce words to their basic lemma in order to use those words with a future lexicon. We preprocessed the texts in our corpus and only needed to replace the review texts in the dataset with those from the corpus. We kept the corpus for further analysis, however, at this stage, it was only useful for preprocessing.

Following the preprocessing, we had to keep the variables that we deemed important for the analysis. Therefore, we retained the apartment ID, the preprocessed reviews, the review score rate, and the district. We selected both the review score rate and district for specific reasons. Firstly, we wanted to compare the rates we would calculate using sentiment analysis with the ones already provided by Airbnb. Secondly, we aimed to classify the rating scores by district and examine any notable differences in review scores among apartments in different districts.

When we had all the selected data, it was time to load the AFINN lexicon. This dataset contains numerous sentiment English words scored between (-4,4), where a lower score indicates a negative sentiment and a higher score indicates a positive sentiment. For each word in all the reviews from all apartments, we assigned its corresponding value based on the AFINN lexicon. Words with a value greater than 1 were labeled as positive, while others were labeled as negative. In the upcoming plots, we will be able to observe the top words that appear in our reviews.

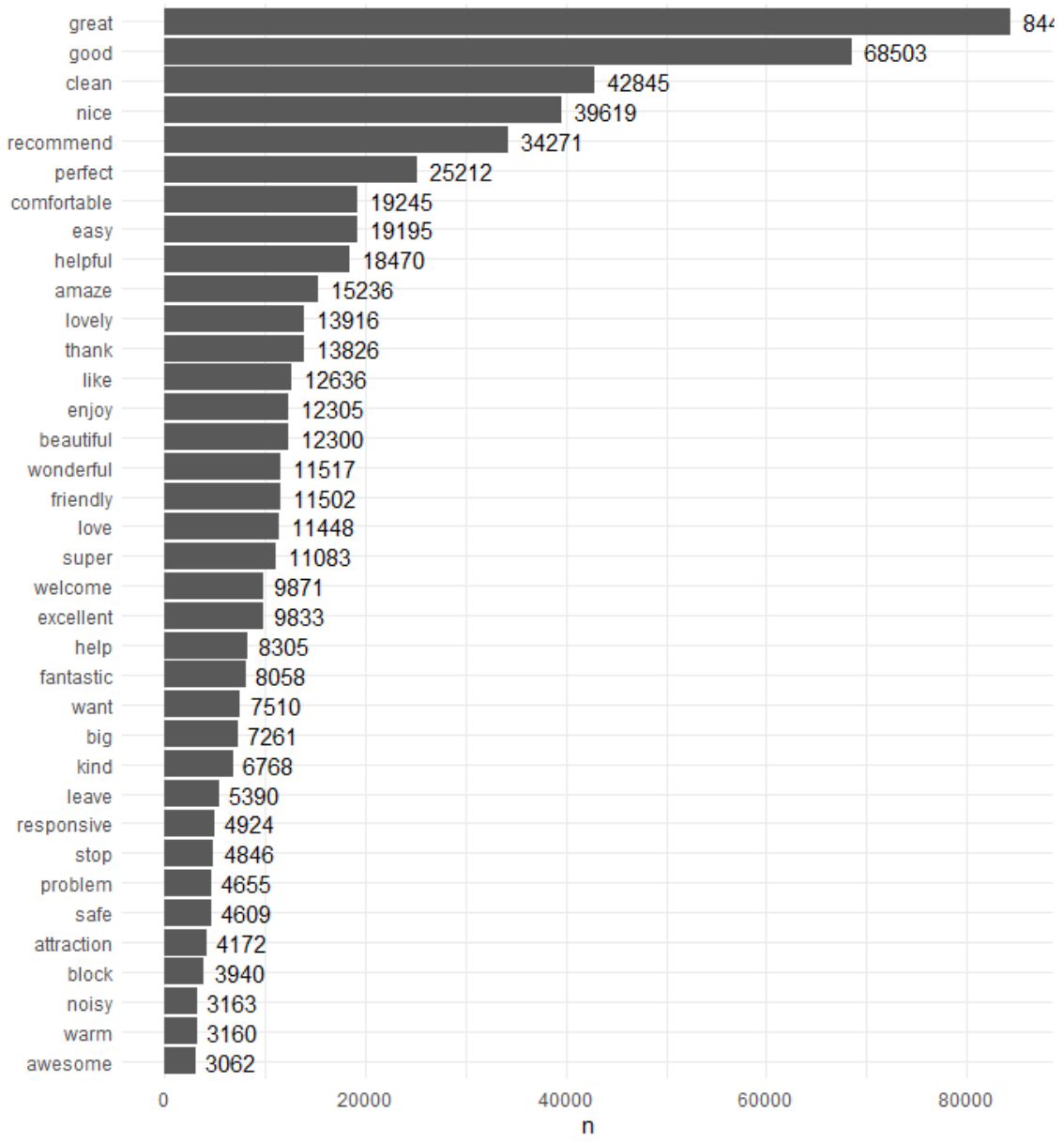



Figure 82: words with the most appearances in the reviews

As we can see, 'great' is the most common word in our reviews, followed by 'good,' 'clean,' and 'nice.' These results could indicate that the majority of the reviews are positive, an aspect that we will likely explore further.


Figure 83: word cloud with the words that appear the most

In this word cloud, we can visualize the words that appear most frequently in the reviews in a unique and visually appealing manner.


Figure 84: word cloud with the most positive and negative words

In this other word cloud, we can observe the top words for each class: positive and negative. In the positive class, 'great' has the highest frequency, followed by 'clean,' 'recommend,' 'good,' 'perfect,' and 'nice.' In the negative class, ‘leave’, 'stop,' 'block,' and 'problem' are the most common words.

Our next step was to aggregate the reviews by district. Our objective was to identify patterns that differentiate the districts from each other based on the sentiments expressed in the reviews. To accomplish this, we created plots to visualize the most positive and negative words used in each district by the users who reviewed the apartments.


Figure 85: top positive words by district

For the positive reviews, we can observe that 'great' and ‘good’ are the most common words in all districts, followed by 'clean,' 'nice,' and 'recommend.' We can highlight that in Nou Barris, Horta and Sant Andreu, the words 'recommend','clean' and ‘nice’ are frequently used.

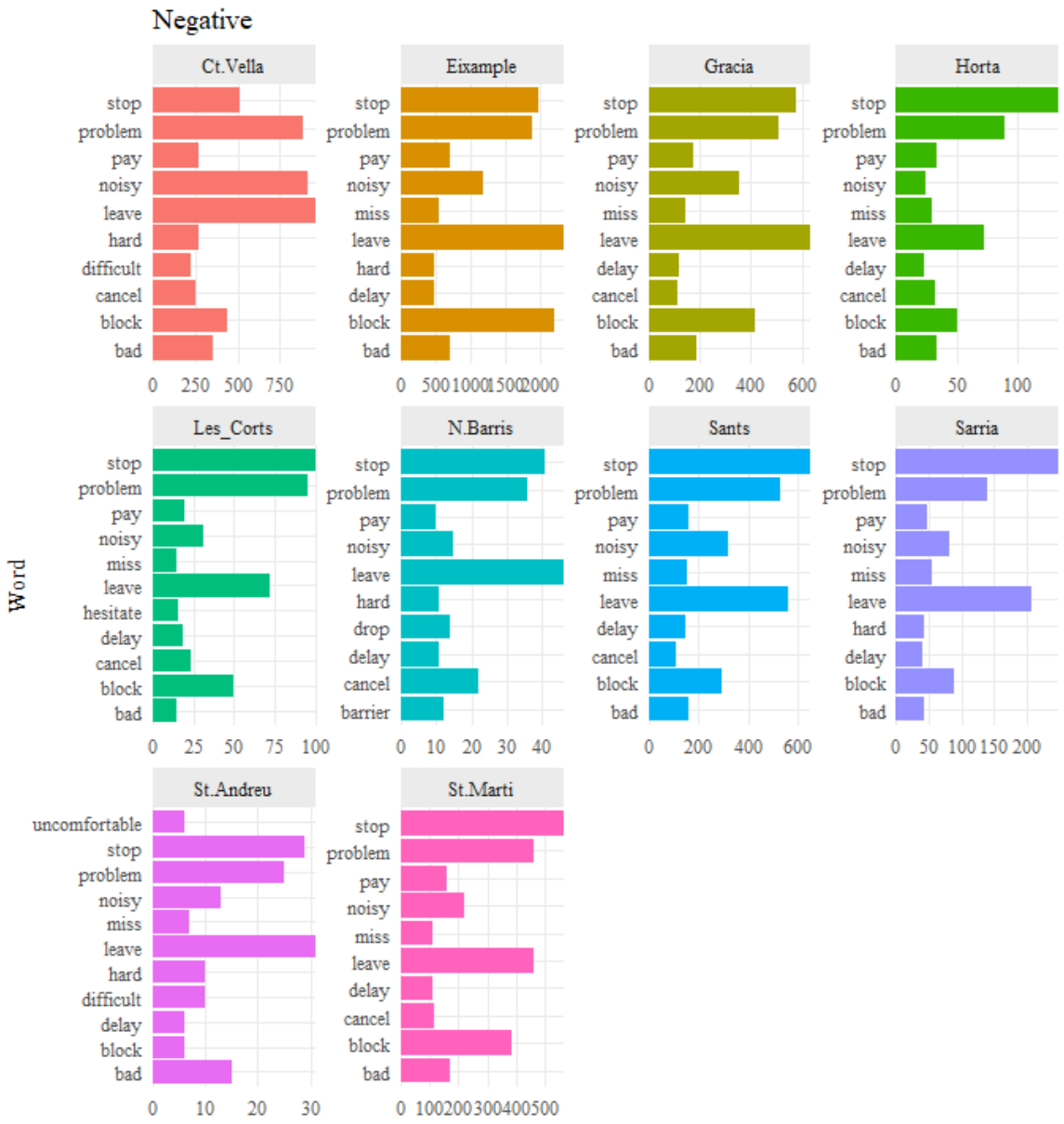



Figure 86: top negative words by district

For the negative reviews, 'stop', 'problem' and ‘leave’ are the most frequently used words by a significant margin. In districts like Eixample, Gracia, and Sant Marti, the word 'block' appears frequently. Additionally, in Sant Andreu, we observe a considerable number of instances of the word 'bad,' while in Nou Barris, the word 'cancel' stands out. In the following plot, we have represented the percentage of positive and negative words by district in the dataset.

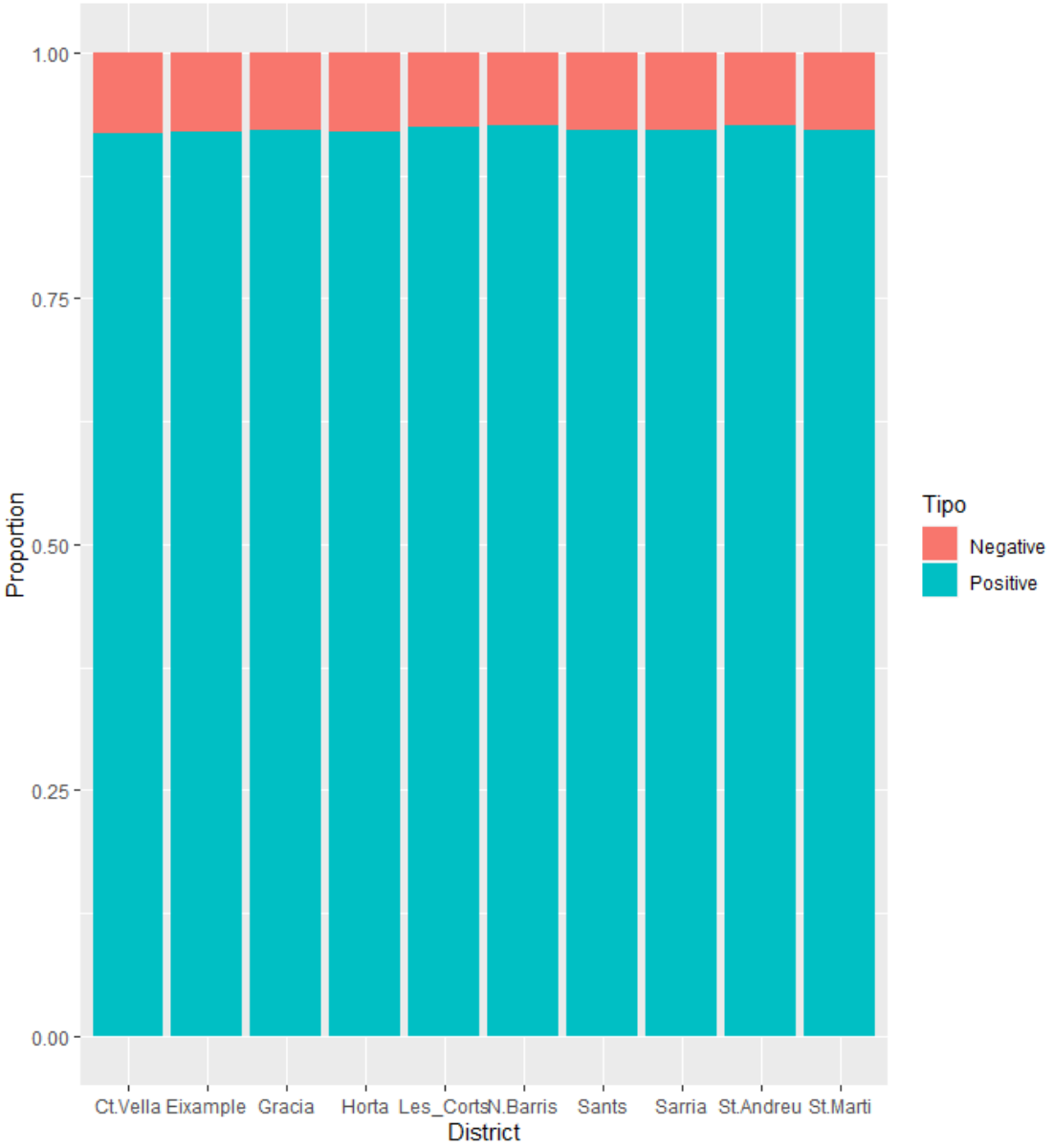



Figure 87: proportion positive/negative words by district

The plot perfectly illustrates that the distribution is almost similar in all districts, with around 90% of positive words and the remaining percentage being negative. Eixample, Horta, and Ciutat Vella have a slightly higher percentage of negative words, while Gracia, Nou Barris, and Sant Andreu have a slightly lower percentage. However, the difference is minimal.

After analyzing the words, it was time to analyze the sentences. We calculated the mean value for the reviews by summing the values of all the words and dividing it by the length of the reviews. In the next plot, we will observe the distribution of the calculated values.

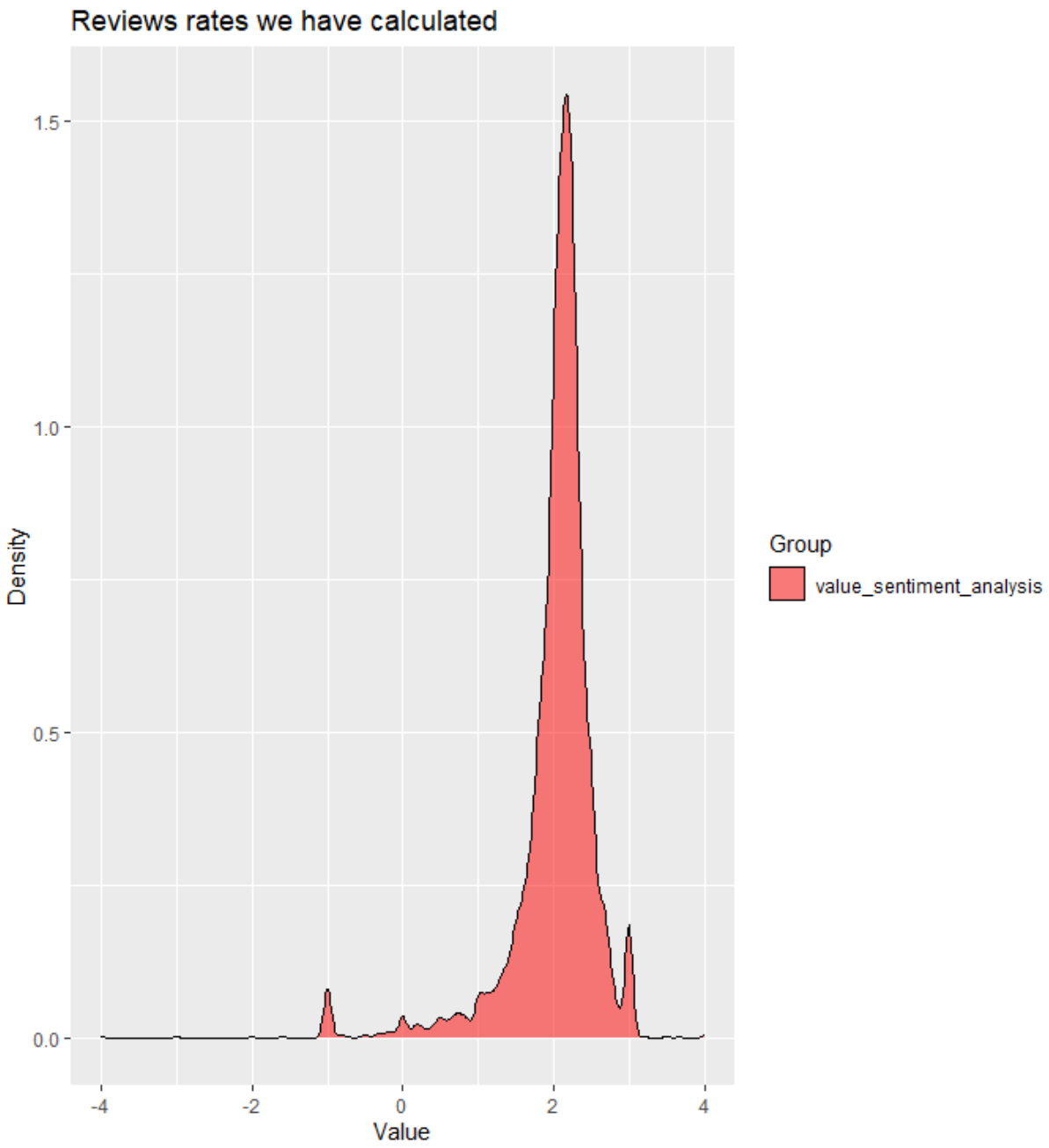



Figure 88: reviews rates of the apartments that we have calculated

As we already know, when the value is higher than 0, it indicates that the review is positive. There is an abundance of positive reviews, while the number of negative reviews is very small. Afterward, we scaled these values between 0 and 5 to facilitate comparison with the scores provided by Airbnb.

Now, in the next plot, we will observe the distributions of the reviews we have calculated versus the ones provided by Airbnb, both on the same scale. The distribution of the values provided by Airbnb is much closer to 5, whereas the distribution of the values we have calculated is centered around 3.5.

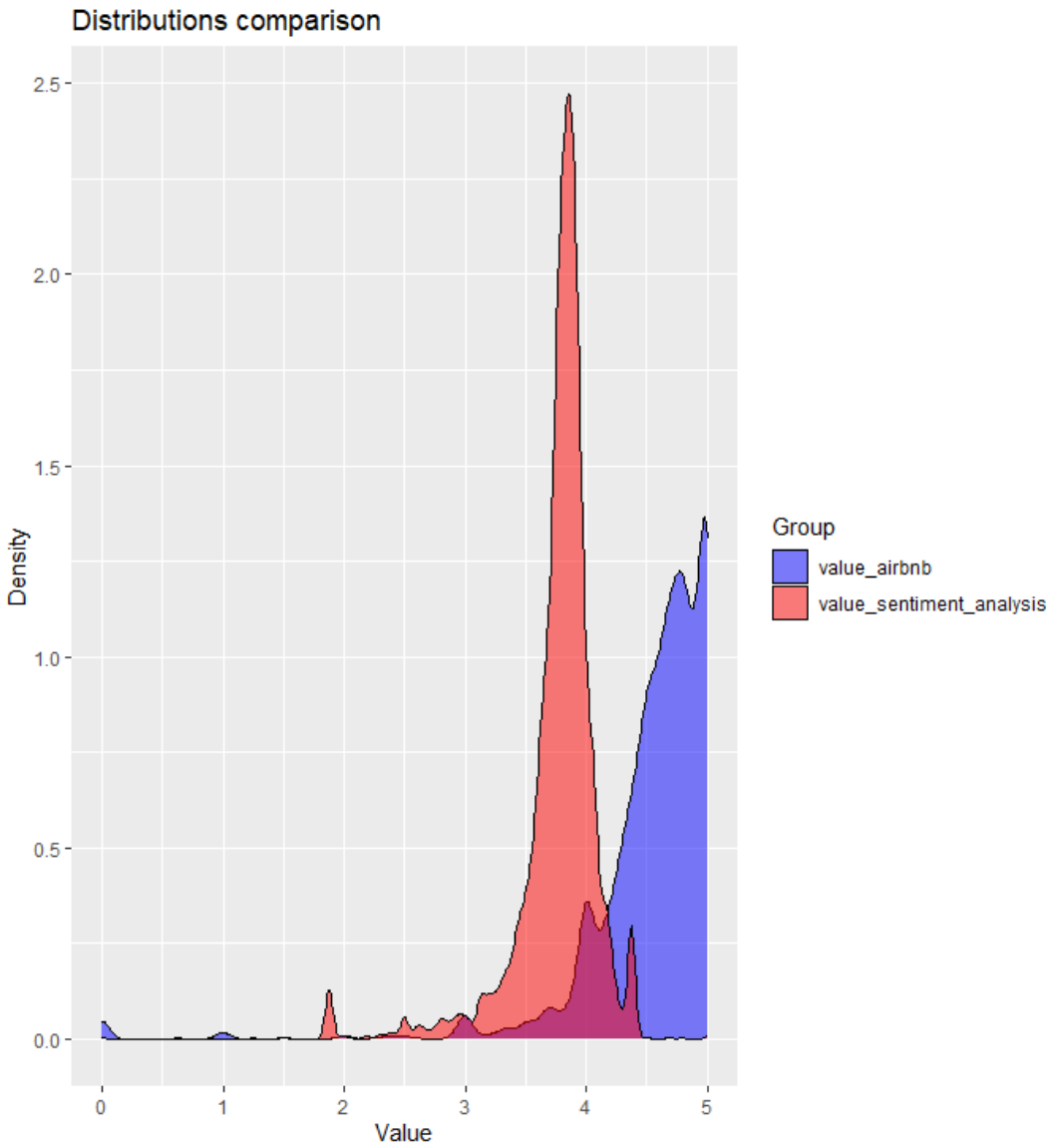



Figure 89: reviews rates of the apartments that we have calculated vs the ones provided by Airbnb

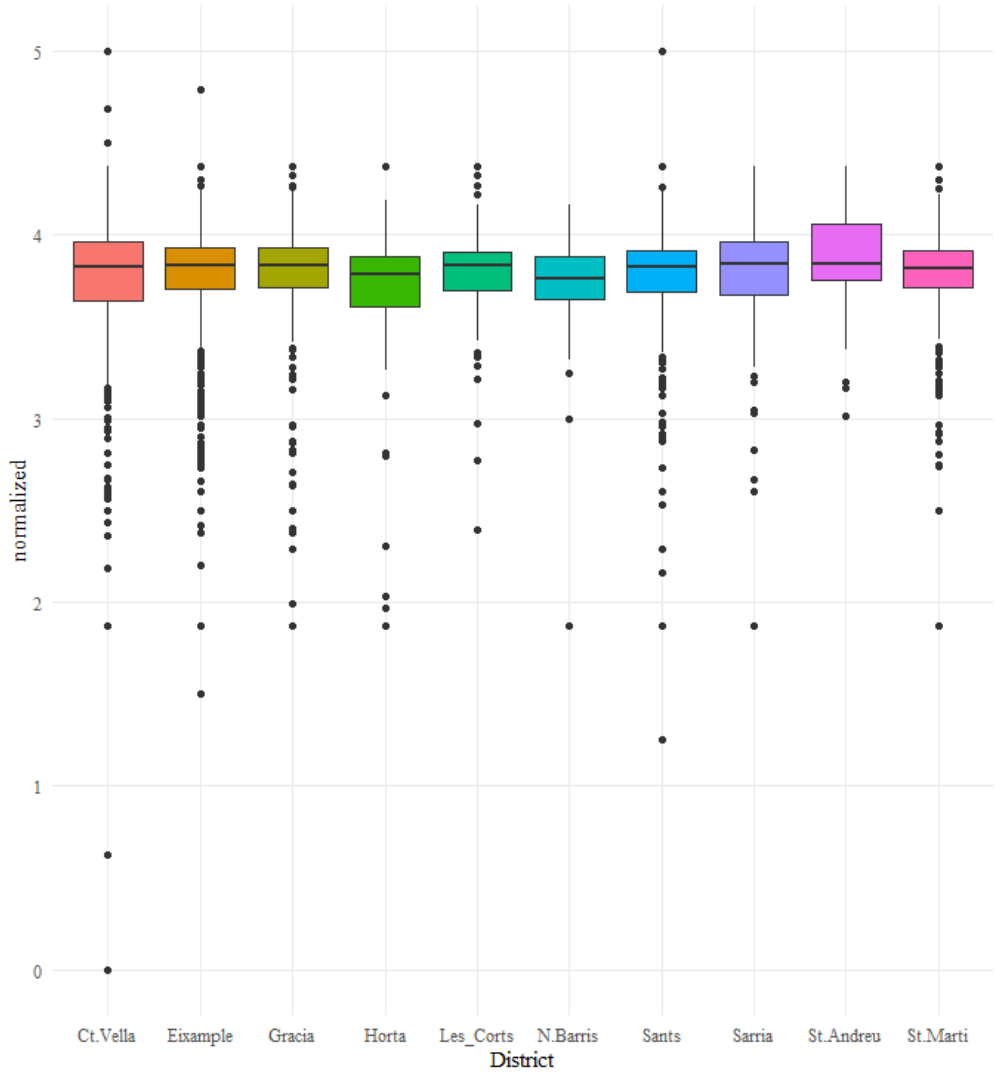



Figure 90: boxplots of the reviews rates we have calculated grouped by district

In these boxplots, we can observe the review scores that we have calculated by district. Nou Barris is the district with the fewest outliers, indicating that the ratings are very similar among apartments. On the other hand, Ciutat Vella has the most outliers, representing a wide range of review scores, including both the best and worst-rated apartments. Moreover, we can see that the majority of the reviews fall between 3.5 and 4, with only a few apartments receiving a score higher than 4. Now, it is time to examine the values provided by Airbnb.

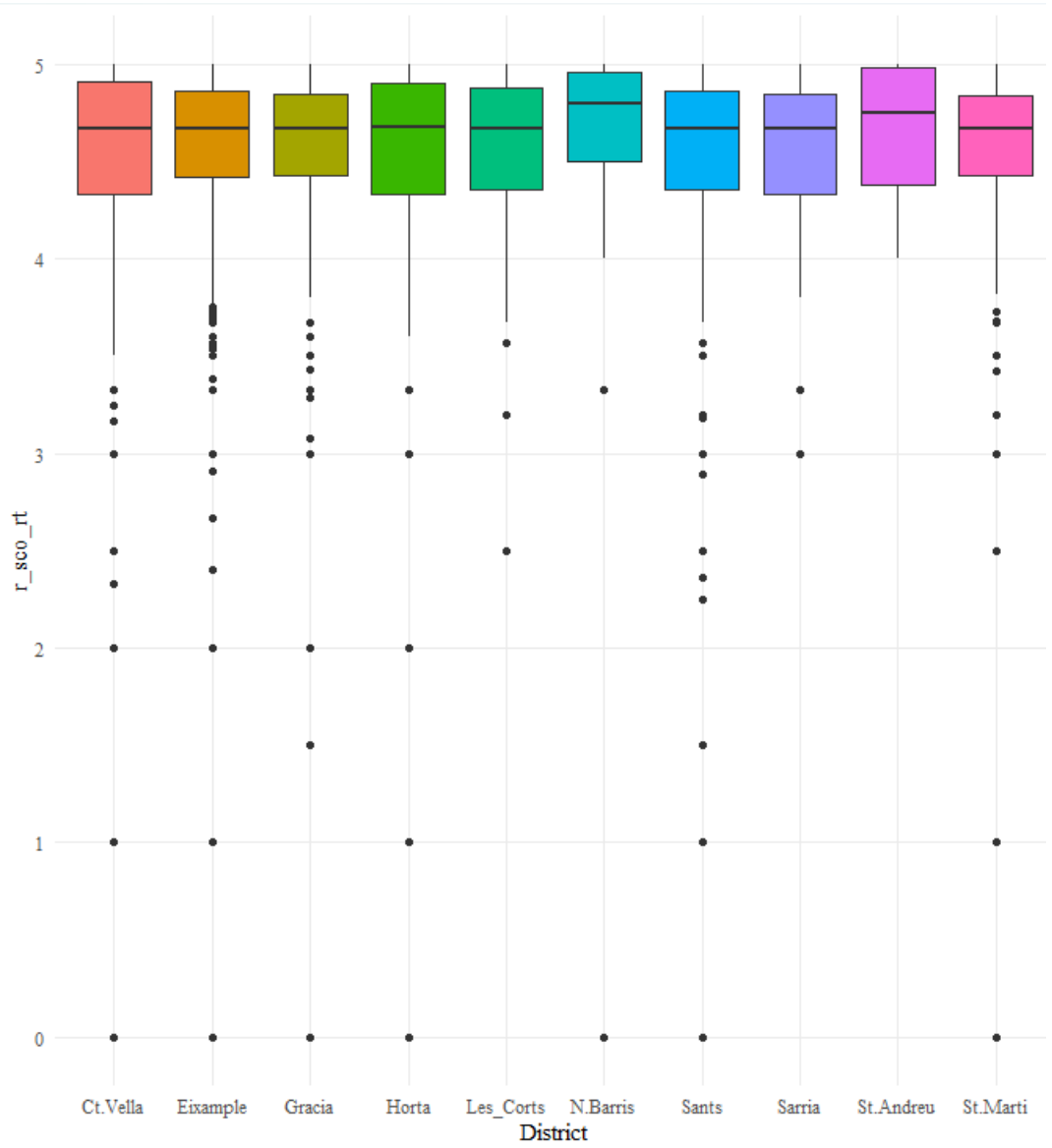



Figure 91: boxplots of the reviews provided by Airbnb grouped by district

This plot displays results that differ significantly from the scores we have calculated. Almost all the apartments have scores between 4 and 5, which are much higher compared to the other scores. Upon observing this, we proceeded to conduct a t-test, a statistical test that utilizes the means from two groups to compare them and determine if there is a significant difference between them. In our test, our null hypothesis (H0) stated that there was no difference between the means of the distributions, while the alternative hypothesis (H1) posited that a significant difference did exist. The results of the test were as follows:

```
        Welch Two Sample t-test

data:  value_airbnb and value_sentiment_analysis
t = 67.436, df = 6169.2, p-value < 2.2e-16
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 0.7497673 0.7946635
sample estimates:
mean of x mean of y
 4.535476  3.763261
```

Figure 92: results from the t-test

Since we have a p-value smaller than 0.05 (less than **2.26e-16**), we reject the null hypothesis, indicating that there is indeed a significant difference between the means of the distributions. Furthermore, the t-value of **67.436** indicates a substantial magnitude of the observed difference between the means of the compared groups, signifying a large effect size.

After examining the distributions and analyzing the result of the t-test, we aimed to determine if there was a correlation between the variables. The correlation value we obtained was **0.6345971**, which suggests a relationship between the variables. This indicates that the calculations we performed using the AFINN lexicon correspond to the scores provided by Airbnb.

In conclusion, there are several key points to highlight from this sentiment analysis:

- First of all, the majority of words used by users convey a positive sentiment, resulting in high review scores.
- There is not much variation in terms of positivity and negativity between the different districts.
- The scores we calculated from the reviews have lower values compared to the scores provided by Airbnb. This could be attributed to the fact that we are using a lexicon with associated values for words. It is possible that users may use words that our lexicon considers negative, which could lower the average review score, despite the user giving a 5-star rating. However, despite the lower numerical values, since the distributions appear similar, our sentiment calculation appears to be reliable.
- Lastly, the high correlation between the two distributions indicates that the values we calculated correspond to the values provided by Airbnb.

# Correspondence Analysis applied to textual data

## Creating Term document matrix

With our dataset merged and the text of our reviews preprocessed in the sentiment analysis section, we proceed to create our term document matrix for correspondence analysis applied to textual data and topic modeling.

Initially, when we created our term document matrix (TDM), due to the abundance of reviews and processed text, despite our efforts in stemming the words and eliminating stopwords, we ended up with an excessively sparse matrix containing 61,201 variables (words). This posed

a problem as our computer experienced significant slowness while working with such a large matrix. Furthermore, this sparsity would cause issues in the future when plotting all these words for our factorial analysis.

To address this problem, we made the decision to remove less frequent terms from the TDM. Using the following R command: tdm <- removeSparseTerms(tdm, sparse = 0.95), we retained only terms that occurred in at least 5% of the documents. By doing so, we eliminated words that likely contained errors, irrelevant proper nouns, and other sporadic words that introduced noise. For instance, words such as "diana," "churchyou," "brright," "bathroomshow," and "apartmentsbrbrgener" were removed.

After applying the previous approach, we have successfully reduced noise and improved the quality of our term document matrix (TDM). Here are the key advantages of this step:

- **Reduction of Noise:** The elimination of less frequent words helps to minimize noise or outlier occurrences that may not contribute significant meaningful information to our analysis. This results in a cleaner dataset.

- **Focus on More Representative Words:** By removing less frequent words, our analysis can now focus on words that occur more frequently across the documents. These more common words are likely to be more representative of the overall content and themes present in our dataset.

- **Dimensionality Reduction:** The removal of less frequent words has reduced the number of unique terms in the TDM, leading to a reduction in dimensionality. This reduction in the number of variables makes subsequent analyses more computationally efficient and easier to interpret.

- **Enhanced Interpretability:** With fewer less frequent words, the remaining words in the TDM are now more interpretable and meaningful. This enhancement in interpretability facilitates the understanding and interpretation of results, this will be very useful when doing correspondence analysis applied to textual data.

As a result of these improvements, our TDM now consists of 3718 observations and 1197 words, representing a substantial reduction from the initial 61,201 variables. This refined TDM sets the stage for more efficient and insightful analysis of our data.

The objective of the following correspondence analysis applied to textual data is to explore and visualize the relationship between words and other variables of our data. This will help us to uncover patterns and associations between words and variables that then can be used to improve apartment descriptions to attract more people to an apartment.

## CA

The first step we need to take in our Correspondence Analysis is to examine the explained variance in each dimension of the CA. This enables us to comprehend the contribution of each dimension to the overall variation in the data. Here we can see the variance of the first 10 dimensions:

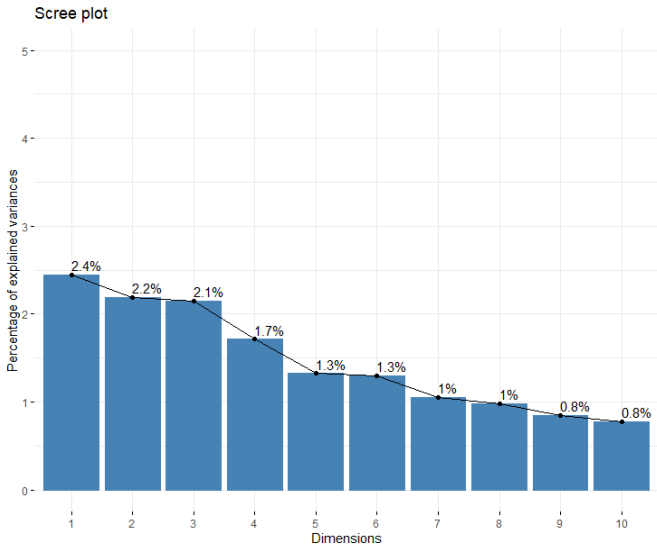



Figure 93: explained variance for the first 10 dimensions

As we can see in the plot, the most significant dimensions are 1,2,3 and 4. This means that these dimensions contain more meaningful patterns and relationships between the words and documents in the analysis.

Now we will look at which 10 words have higher contribution in the 4 dimensions. This analysis helps us understand the specific words that play a significant role in defining the patterns and relationships captured by those dimensions. These words have a greater influence on shaping the content, themes, or characteristics represented by each dimension. They provide valuable insights into the underlying meaning and interpretation of the dimensions.

Here are the results:

| **Word** | **Dim1** | **Dim2** |
|---|---|---|
| autom | 1.0901475 | 12.05617633 |
| post | 0.8659712 | 9.77076182 |
| hostel | 8.9584409 | 1.05866971 |
| cancel | 0.9315402 | 8.76510059 |
| room | 4.6358293 | 4.30803221 |
| reserv | 0.6796424 | 6.97369016 |
| staff | 6.7868724 | 0.10616309 |

| | | |
|---|---|---|
| apart | 0.7891013 | 3.19325425 |
| hotel | 3.0371675 | 0.09263415 |
| host | 0.3267757 | 2.09881908 |

Figure 94: table with the contribution of the most contributive words in dimensions 1 and 2

| **Word** | **Dim3** | **Dim4** |
|---|---|---|
| autom | 14.5513077 | 1.7917203 |
| post | 11.8747542 | 1.4351602 |
| view | 0.4046374 | 11.9779077 |
| cancel | 10.8883131 | 1.3298985 |
| reserv | 8.5933077 | 1.1054741 |
| staff | 4.8637665 | 4.3490689 |
| terrac | 0.4262910 | 8.1648020 |
| hostel | 6.3823843 | 1.4307672 |
| room | 5.4081880 | 0.3288287 |
| hotel | 2.7547548 | 2.8616231 |

Figure 95: table with the contribution of the most contributive words in dimensions 3 and 4

From the results obtained, we can see that in Dim 1 the words that have more contribution are hostel, staff and room. In Dim 2 are the words autom, post, cancel and reserv. In Dim 3 we can see that the words are autom, post, cancel, reserv, hostel and room, this Dim is similar to Dim 2. Finally in Dim 4 we can see that the most contributive words are view and terrac. This preview of the most significant words in each dimension will help us to interpret the following plots of the CA factor map.

**Dimensions 1 and 2**

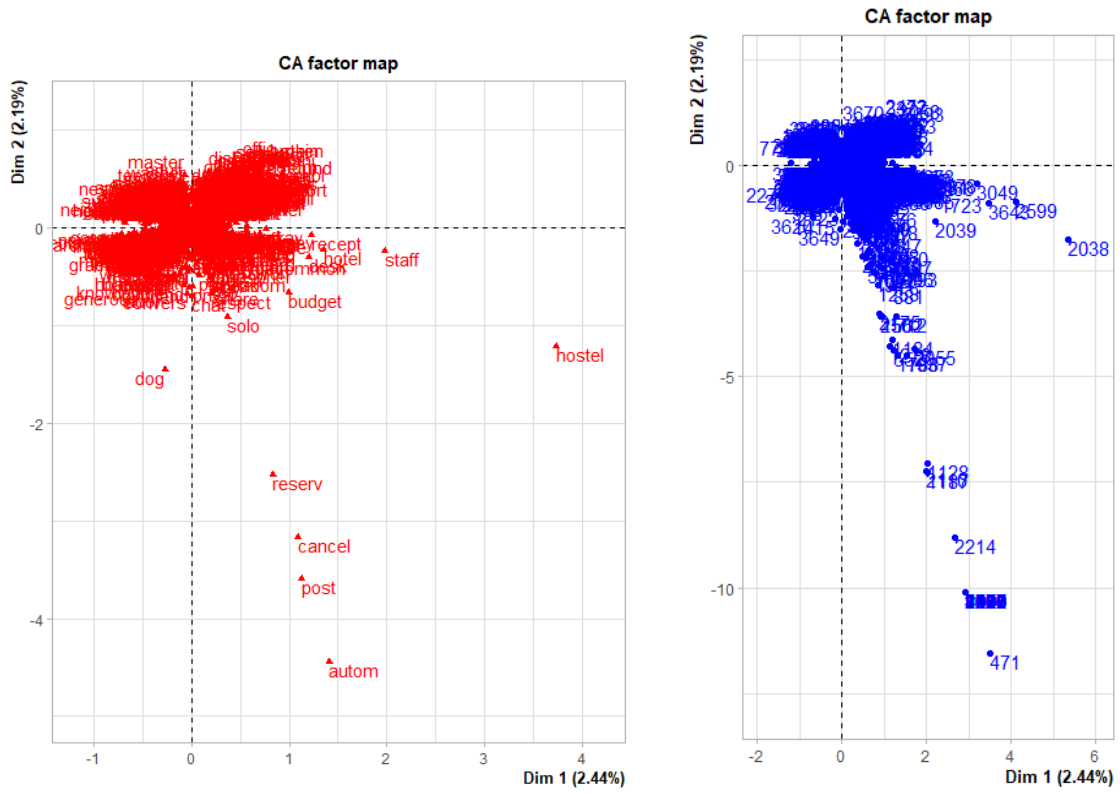



Figure 96: CA factor map for words and documents in dimensions 1 and 2

In the left-side plot, it is evident that there is a cluster of words that is difficult to interpret. However, when we focus on the non-overlapping words, we can observe that the word "hostel" has a significant influence on the Dim 1 axis. Beyond that, there are no other noteworthy highlights from Dimension 1. Moving on to Dimension 2, we can observe that words like "autom," "post," "cancel," and "reserve" have a considerable impact. It is possible that this dimension corresponds to reviews that are more closely associated with cancellations, reservations, automatic check-ins, or administrative matters.

Another important observation from this plot is the presence of four distinct clusters in both the cloud of words and the cloud of documents. This suggests that, in the future, when we perform LDA (Latent Dirichlet Allocation), we may consider utilizing four topics for topic modeling. However, a more detailed analysis of this will be conducted in the LDA section.

### Dimensions 3 and 4

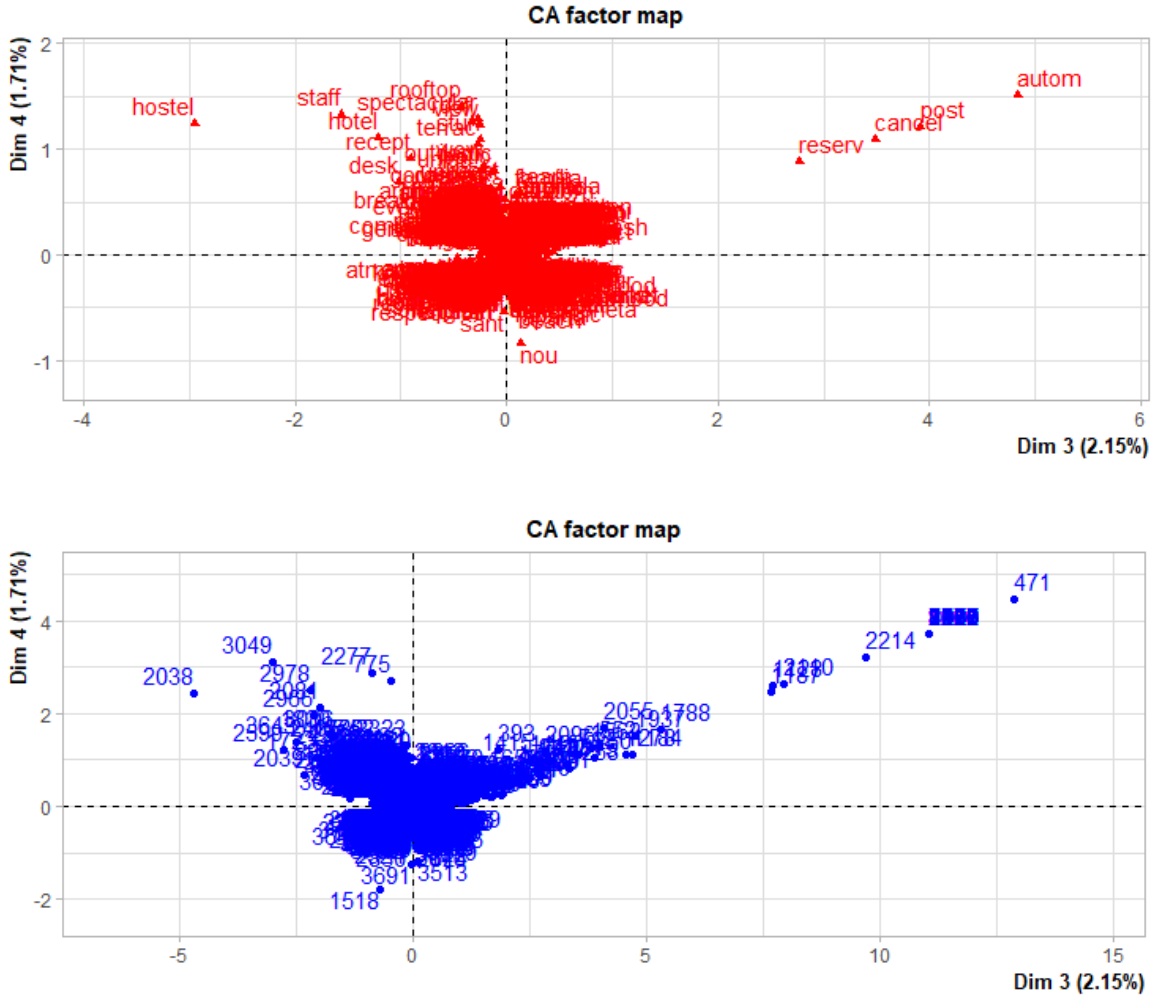



Figure 97: CA factor map for words and documents in dimensions 3 and 4

In these plots, we can once again observe the contribution of each word (represented in red) and document (represented in blue) to dimensions 3 and 4. Similar to dimension 2, we can see that words such as "autom," "post," "cancel," and "reserv" have a significant contribution to dimension 3. Additionally, in this case, the word "hostel" also has a contribution, this time negative. Moving on to Dimension 4, we can observe the positive impact of words like "rooftop," "autom," "terrac," "staff," and "spectac". It is evident that these words are related to the same topic as they appear together in the factorial map.

Furthermore, we once again notice the presence of what could potentially be four clusters within the cloud of overlapping words. This observation is important to consider for future approaches to topic modeling. Taking these clusters into account can help guide and inform our topic modeling analysis.

To conclude this initial Correspondence Analysis, it is evident that the results have not provided us with the anticipated insights. The presence of a large number of words has led to significant overlap, making it difficult to discern meaningful patterns. The dimensions exhibit minimal variation, indicating a lack of substantial information. Even the words that made significant contributions did not reveal any intriguing relationships between them.

Moving forward, we will now proceed with a Correspondence Analysis using GALT (Generalised Aggregated Lexical Tables). We have high expectations that this approach will yield more relevant information regarding the relationship between the modality of different variables and the words present in the apartment reviews.

# CA GALT (Generalised Aggregated Lexical Tables)

To conduct our Generalised Aggregated Lexical Tables (GALT) analysis, we will augment our term document matrix with additional variables of interest. This will enable us to explore the relationship between words and the various modalities of these variables.

We have identified several variables to incorporate: neighborhood, host is super host, host response time, and room type. We have chosen these variables because we believe their modalities are likely to be associated with the words found in the reviews. For instance, we anticipate discovering words that capture the characteristics of specific neighborhoods near the corresponding modalities of that neighborhood in the factorial map. This will provide information about what aspects of Barcelona's neighborhoods are of greatest importance to people. Furthermore, we expect to observe words that represent attentive hosts in close proximity to the "True" modality of superhost and also to see what people write about the type of room that they had.

Now we proceed to perform our CA GALT with the R function *CaGalt().*

After performing de CaGalt we proceed to look at the eigenvalues to see the cumulative percentage of variance that we have among dimensions. This is what we obtain when looking into the eigenvalues.

| | **Percentage of variance** | **Cumulative percentage of variance** |
|---|---|---|
| Dim 1 | 30.21491 | 30.21491 |
| Dim 2 | 15.84219 | 46.05710 |
| Dim 3 | 12.06646 | 58.12357 |
| Dim 4 | 10.41381 | 68.53738 |
| Dim 5 | 7.6767964 | 76.21418 |
| Dim 6 | 5.1337348 | 81.34792 |

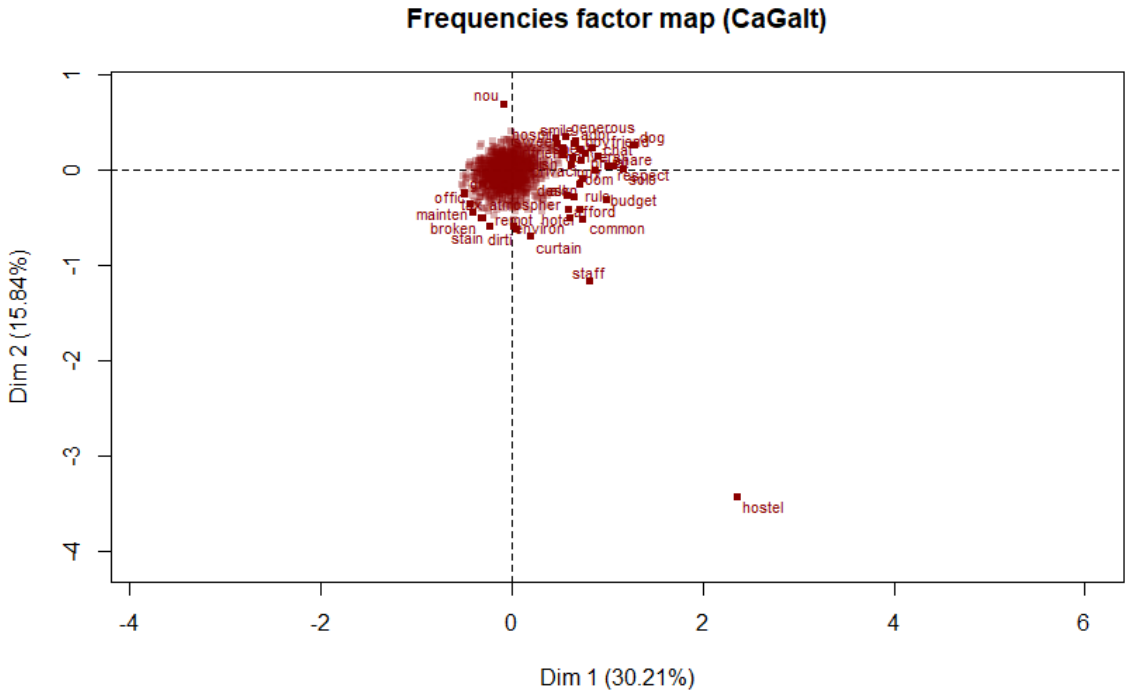
Figure 98: Table with the percentage and cumulative percentage of variance of the first 6 dimensions

This table contains only the first 6 dimensions from the CaGalt. We have considered studying these 6 dimensions because they have a cumulative percentage of variance of

81.35% which implies that they are very informative in representing the relationships between the variables and words.

Now, we will proceed to examine all the plots generated by our Correspondence Analysis and thoroughly analyze the relationships between words and modalities. This comprehensive analysis will allow us to gain a deeper understanding of the associations and connections within our data.

**Dimensions 1 and 2:**



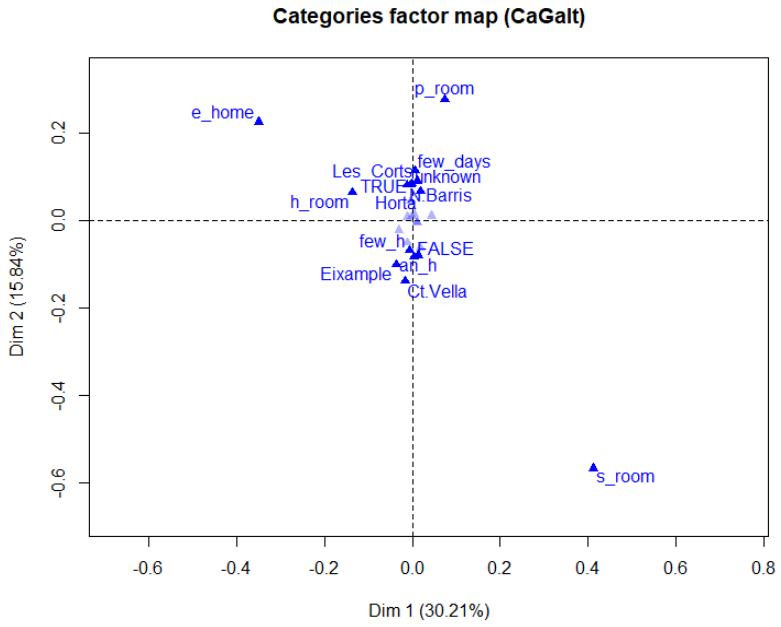
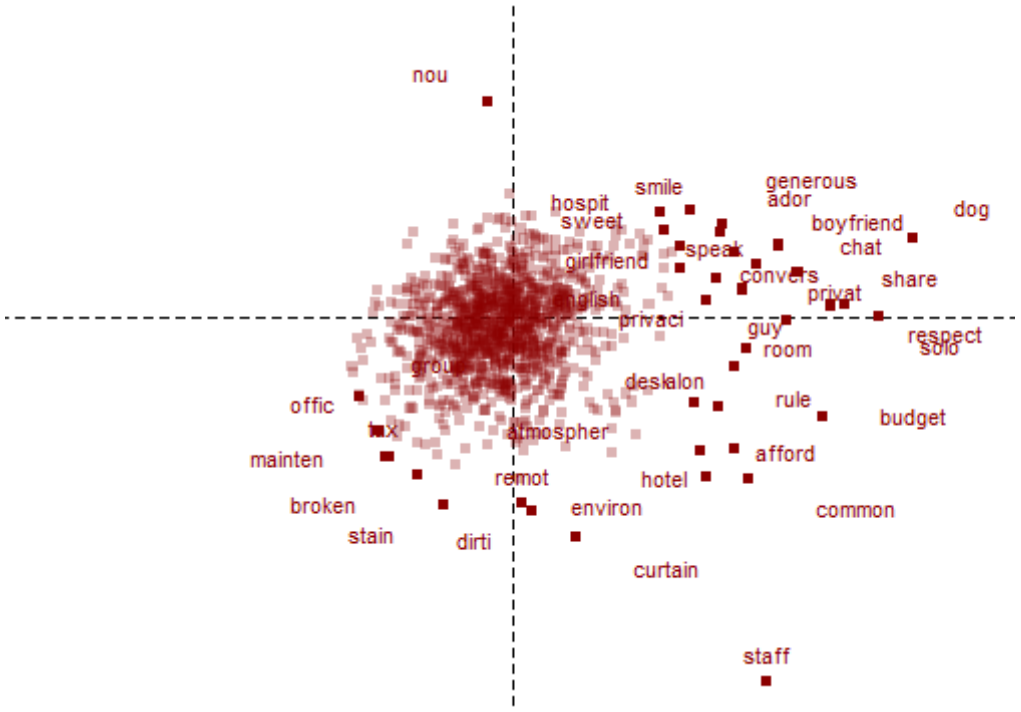



Figure 99: CaGalt factor map for words and modalities of variables in dimensions 1 and 2

Due to the large number of words, the visibility of details is limited as they overlap on the plots. However, from the first two plots, we can observe an association between the modality "shared room" and the word "hostel". They appear in close proximity to each other, indicating a strong relationship between them. It suggests that individuals who stay in shared rooms often mention hostels in their reviews. We can also see the relationship between word nou and private room. It seems like dimension 2 gives information about if an apartment is private or shared.

Now we are going to zoom on the overlapped regions to see if we can extract more relevant information.



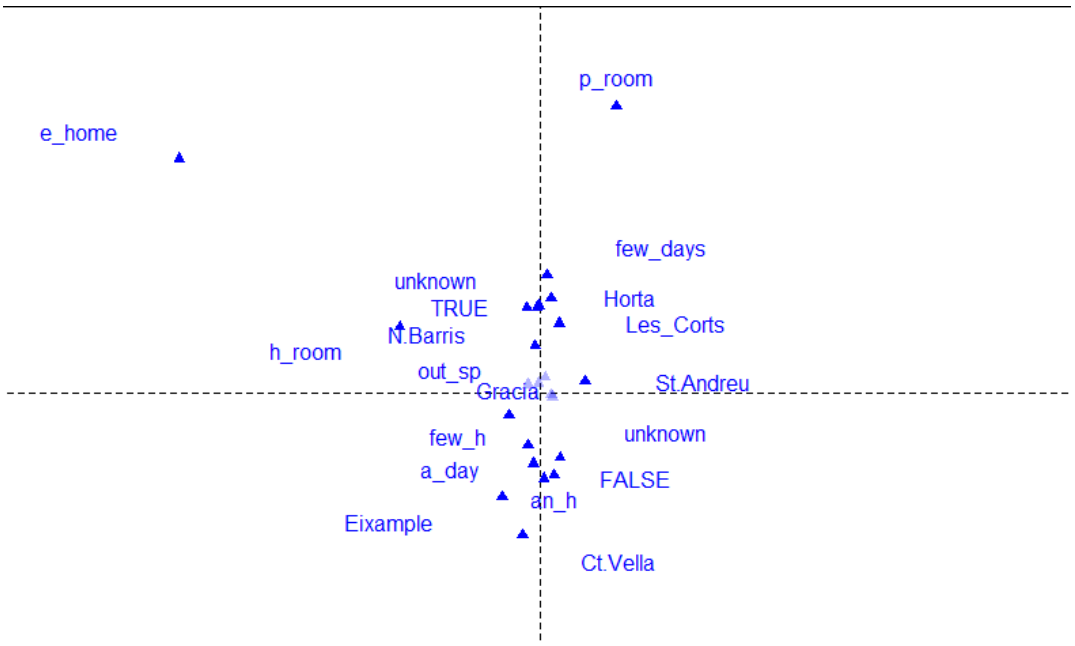
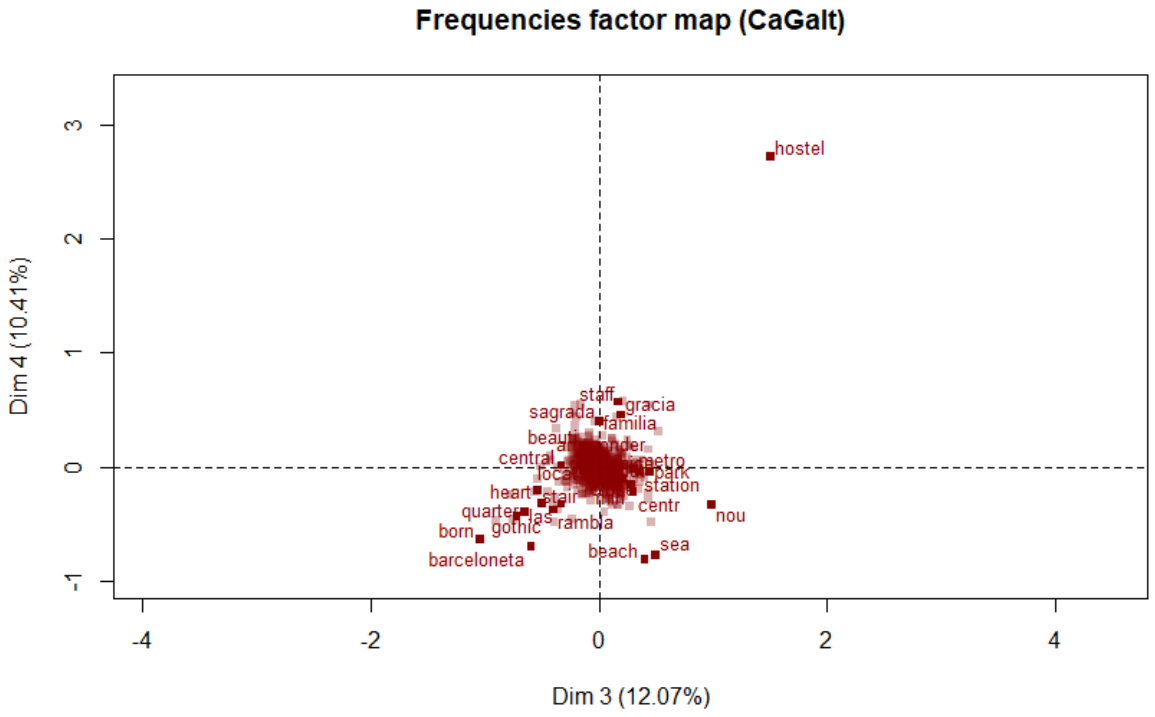



Figure 100: Zoomed CaGalt factor map for words and modalities of variables in dimensions 1 and 2

In these initial two dimensions, it is challenging to identify any notable relationships between words and variables. However, we can observe that the words "dirti" and "stain" are in close proximity to the neighborhoods "CtVella" and "Eixample." This alignment seems logical as these neighborhoods are known to have cleanliness issues.

**Dimensions 3 and 4:**



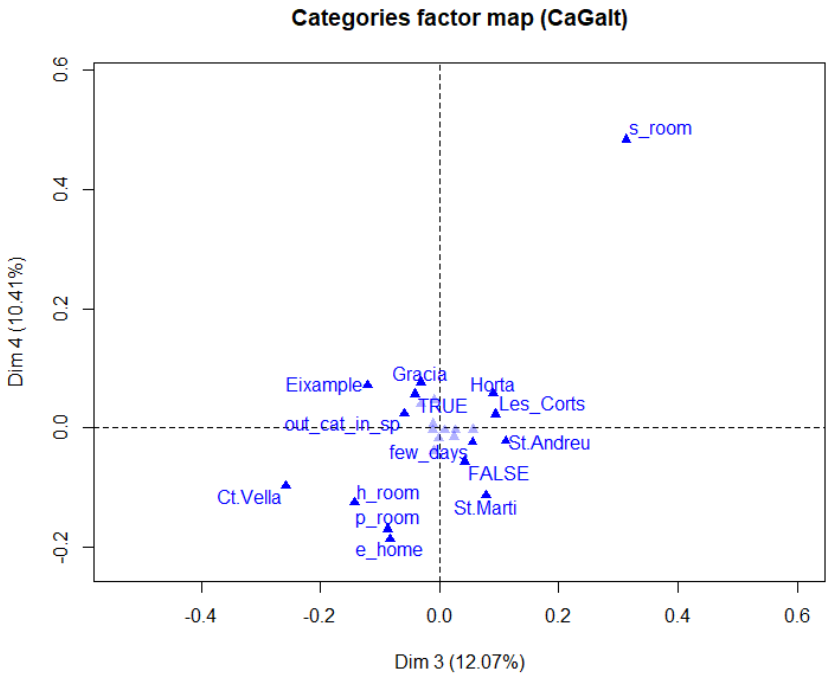
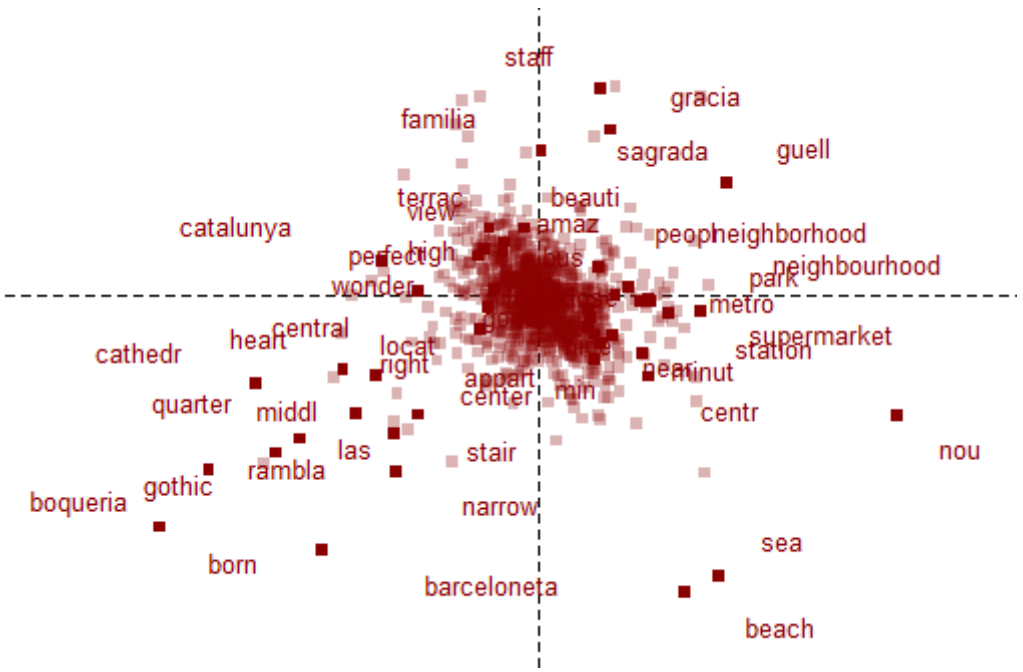



Figure 101: CaGalt factor map for words and modalities of variables in dimensions 3 and 4

We can identify again the relation between shared room modality and the word hostel. Before starting the analysis of the rest of relationships let's zoom into the cloud of words and modalities.



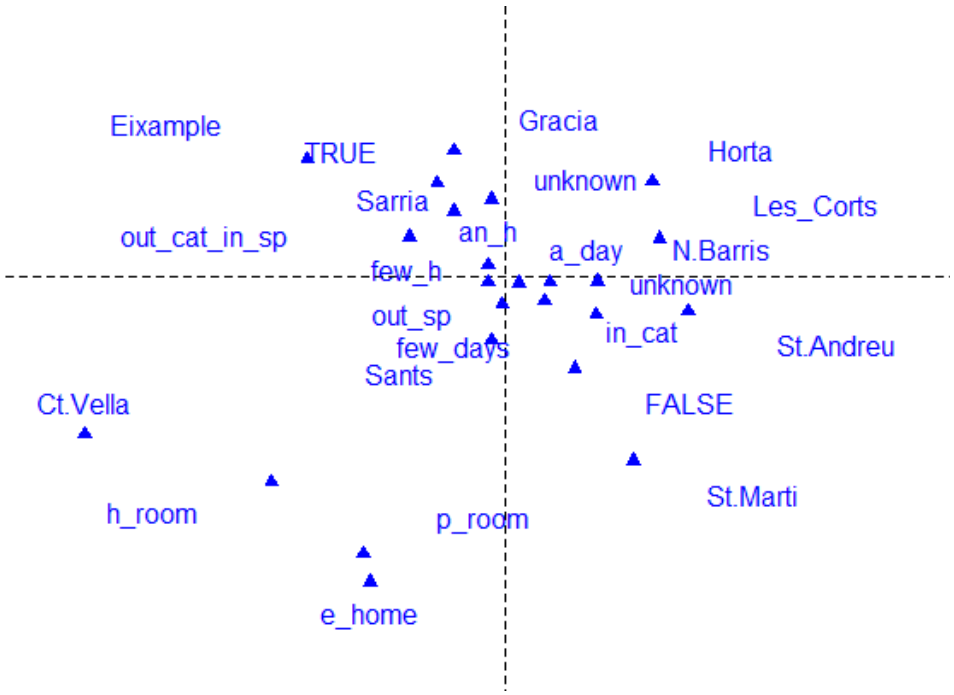



Figure 102: Zoomed CaGalt factor map for words and modalities of variables in dimensions 3 and 4

We can observe several intriguing relationships in these two plots. Firstly, it is evident that individuals staying in the St. Marti neighborhood frequently mention the sea and beach in their reviews, indicating a strong association between these aspects.

Additionally, near the Ct. Vella modality, we find a cluster of interesting words that appear to represent notable places in this neighborhood. Examples include "boqueria" (referring to Mercat de la Boqueria, a renowned market in Barcelona), "born" (referring to the historic El Born neighborhood), "gothic," "rambla," and "cathed." These words likely capture discussions about famous landmarks and features of the Ct. Vella neighborhood, such as its central location. Moreover, words like "heart" and "middl" may reflect the central and vibrant nature of this area.

Near the Eixample modality, we come across the word "catalunya", which likely refers to Plaça de Catalunya, the city's most iconic square and also ‘sagrada’ and ‘familia’. This suggests that individuals staying in the Eixample neighborhood often mention this significant landmarks in their reviews.

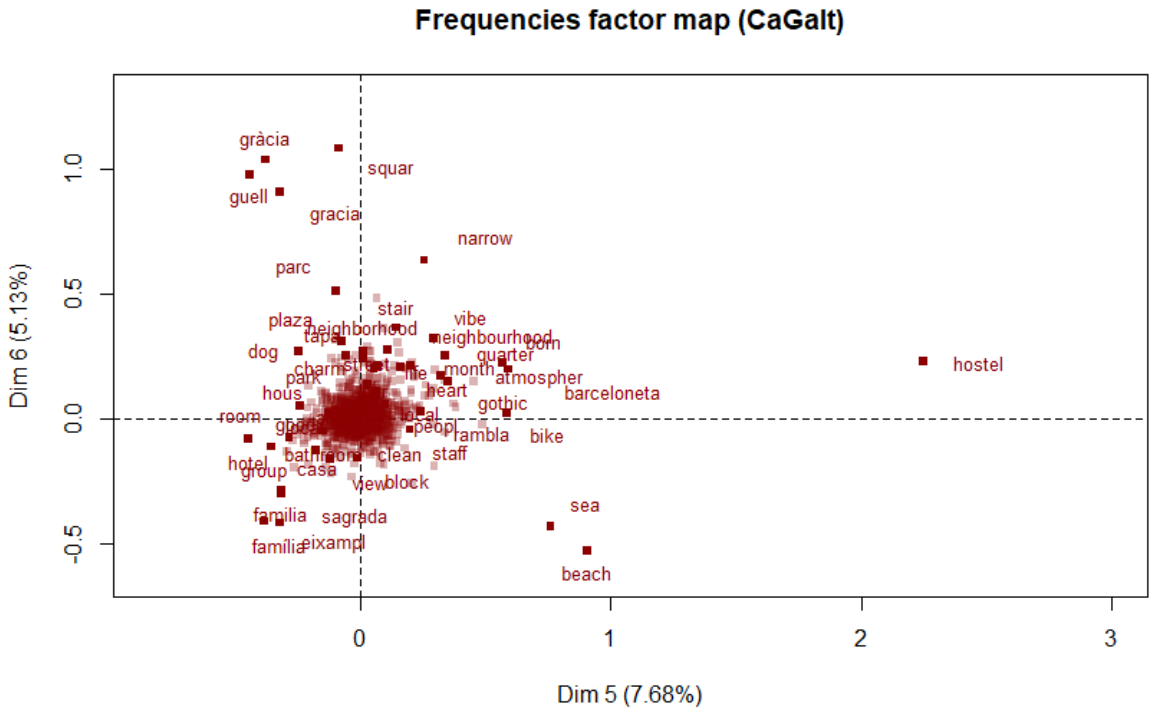
Moving on to the Sarria modality, we find words like "perfect," "high," "terrac," and "view." These words indicate that people who choose to stay in Sarria highly appreciate the neighborhood's scenic views and advantageous location.

In the vicinity of the Gracia modality, we encounter words such as "sagrada" (referring to the Sagrada Familia), "guell" (alluding to Parc Güell), and "gracia" (representing the neighborhood itself). These words likely reflect reviews about the famous attractions and the charm of the Gracia neighborhood.

Lastly, near the St. Andreu modality, we find the words "station" and "metro," which are crucial in this neighborhood as they represent the transportation connections to the city center.

Overall, in dimensions 3 and 4 the relationships between words and modalities provide valuable insights into the preferences, experiences, and attractions associated with each neighborhood in Barcelona.

**Dimensions 5 and 6:**



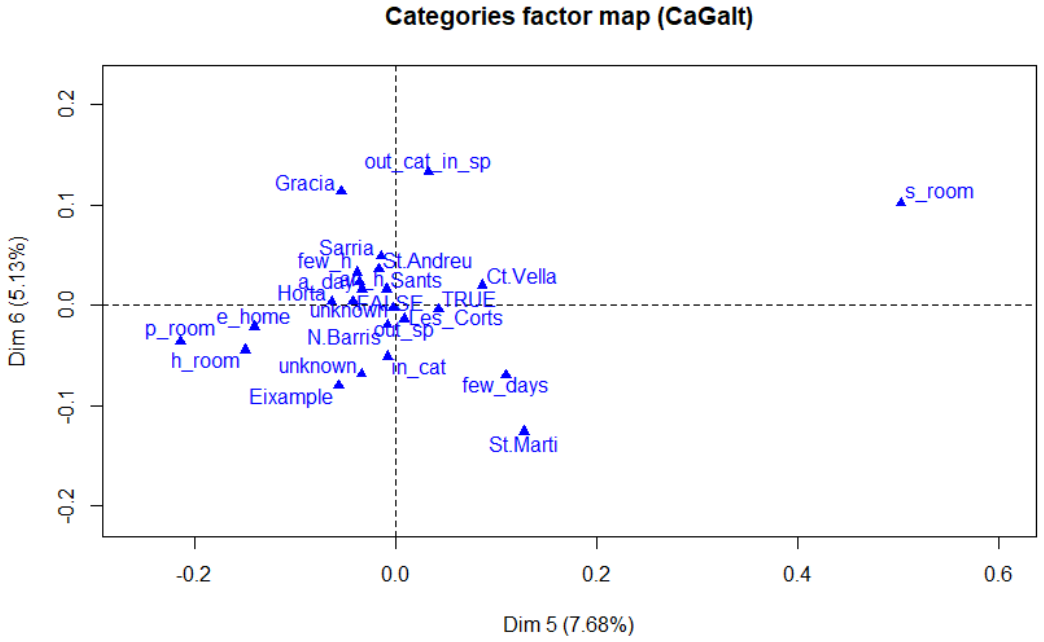
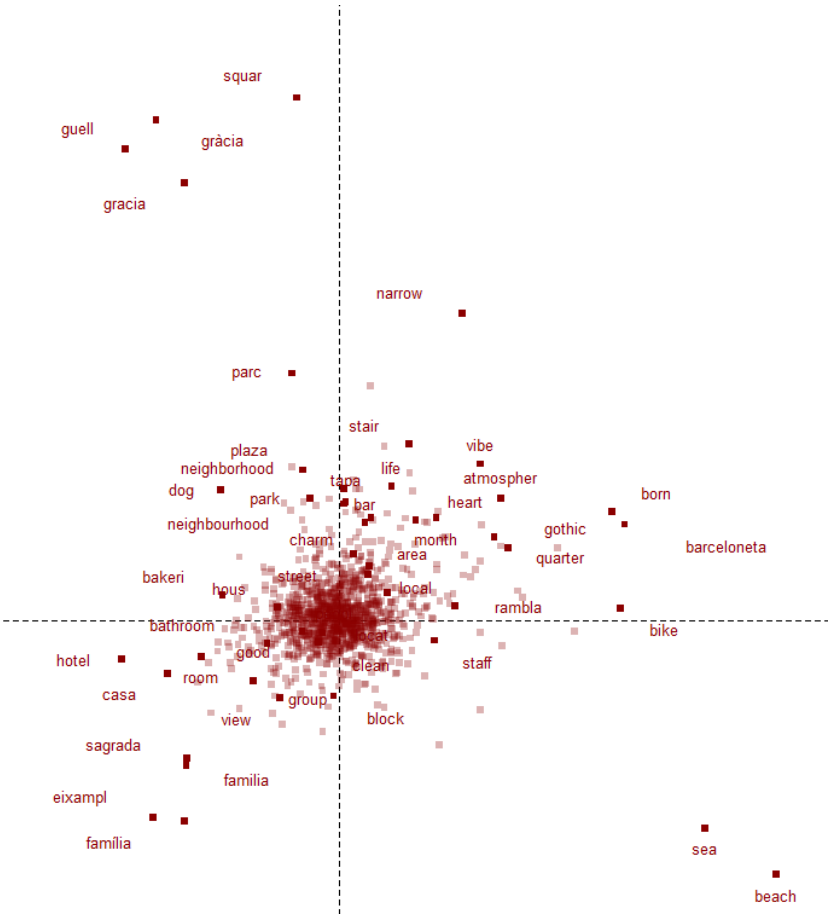



Figure 103: CaGalt factor map for words and modalities of variables in dimensions 5 and 6

We can identify again the relation between shared room modality and the word hostel. Before starting the analysis of the rest of relationships let's zoom into the cloud of words and modalities.



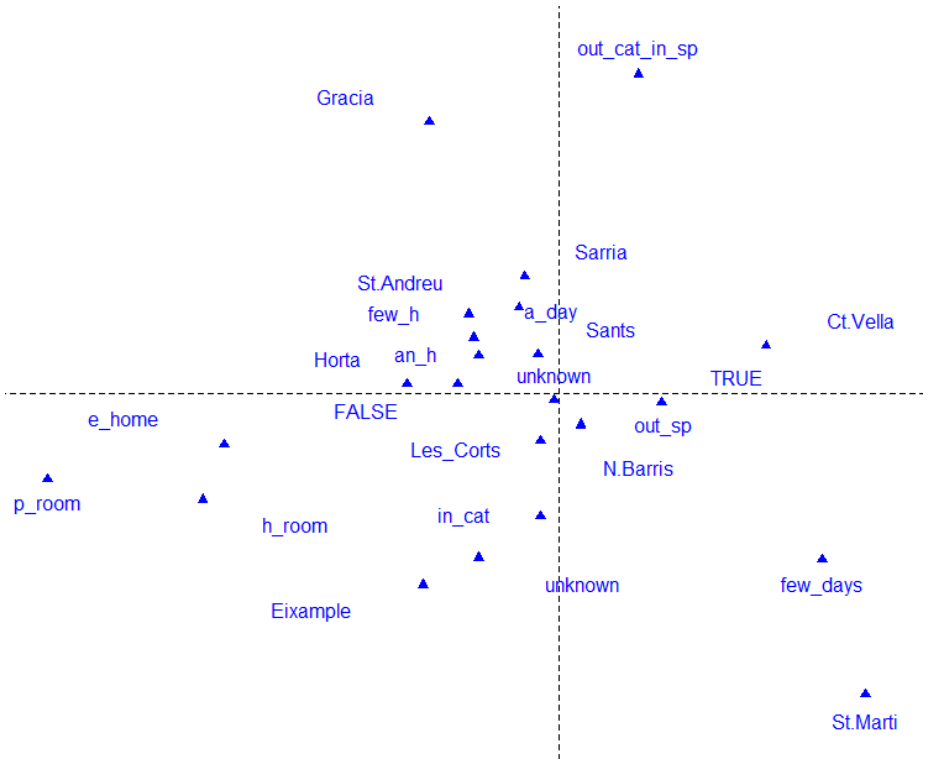



Figure 104: Zoomed CaGalt factor map for words and modalities of variables in dimensions 5 and 6

Once again, we observe the correlation between neighborhoods and their prominent landmarks. In dimensions 5 and 6, we notice that Barceloneta emerges as a notable topic within the Ciutat Vella neighborhood. This indicates that individuals often mention Barceloneta when discussing their experiences in this particular area.

Furthermore, in these dimensions, we find that the "entire home" modality is closely associated with the "private room" and "hotel room" modalities. This proximity aligns with the presence of words such as "hotel," "casa," and "room."

However, we cannot extract any further substantial information from these two dimensions.

In summary, the analysis of dimensions 5 and 6 reinforces the relationship between neighborhoods and their famous landmarks, while also highlighting the associations between different modalities and the corresponding words used in the reviews.

**CA conclusion**

In conclusion, this correspondence analysis applied to review text provides valuable insights for property owners who aim to optimize their apartment rentals. It enables them to understand the aspects that people prioritize when choosing accommodations. For instance, if an apartment is located in the Ciutat Vella neighborhood, the analysis suggests that highlighting its proximity to landmarks such as El Born, the Boqueria Market, and the Gothic Quarter in the location description could attract potential renters.

By incorporating the prominent features and attractions identified through the analysis, property owners can effectively market their apartments to align with the preferences and

interests of potential guests. This knowledge can help them emphasize the unique characteristics of their properties, making them more appealing to the target audience.

# 9. Topic modeling with LSA

Before going into topic modeling with LDA we want to do a brief analysis of our data using LSA.

Latent Semantic Analysis (LSA) works on the idea that words with similar meanings often show up in similar situations or contexts. By studying the statistical patterns of how words appear together in a large collection of documents, LSA can uncover these patterns. It then represents words and documents in a special space called a "concept space," which has fewer dimensions than the original data.

In this concept space, each word and document is represented as a vector. The different dimensions of these vectors correspond to various concepts or topics. When words or documents have similar meanings or content, their vector representations will be similar, and they will be located closer to each other in the concept space.

Now we will compare different words in our corpus based on cosine similarity. This similarity determines the cosine of the angle between the two vectors that represent the words, representing how similar or related they are in terms of their orientation in the vector space.

During the previous correspondence analysis, we observed that there were several words that tended to appear together or be far apart in the corpus. Now, let's examine the cosine similarity between these words using the LSA approach and let’s analyze the results.

| **Word 1** | **Word 2** | **Cosine similarity** |
|---|---|---|
| Sagrada | Familia | 0.9905062 |
| Hostel | Dog | 0.08027841 |
| Hostel | Beauti | 0.2590059 |
| Hotel | Beauti | 0.6668833 |
| Reserv | Cancel | 0.902933 |

In this table, we can observe that words with a cosine similarity near 1 share a similar semantic meaning in our corpus. For instance, the words "sagrada" and "familia" exhibit a very high similarity because they frequently appear together in reviews related to Barcelona, specifically in reference to the Sagrada Familia. On the other hand, we have "hostel" and "dog," which have a very low cosine similarity. This may be attributed to the fact that people who visit hostels generally do not discuss whether pets are allowed or not. The table also presents other interesting cases, such as the distinction between "hostel" and "beauti," as well as "hotel" and "beauti." The similarity to "beauti" significantly increases when the word is "hotel" rather than "hostel," indicating that people may have more positive reviews for hotels.

Lastly, we can observe a substantial similarity between "reserv" and "cancel" since they are typically used in the same context.

Now we are going to use the function *plot_neighbors()* from R and we are going to take a look at the 10 most similar words that this function gives us based on cosine similarity.

**Word: gracia**

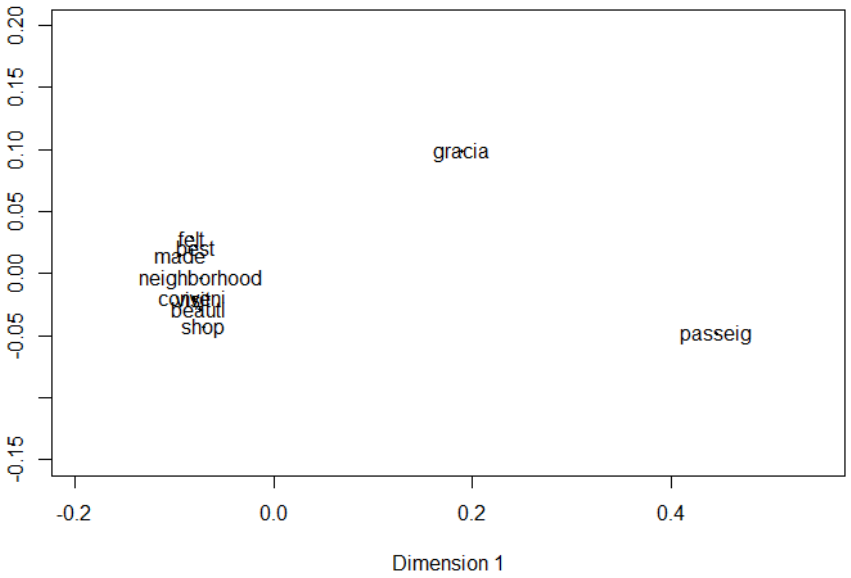



Figure 105: LSA 10 words with highest cosine similarity associated with gracia

In this plot, we can observe that "gracia" exhibits a significant similarity with "passeig," "neighborhood," "beauty," "shop," and more. This sequence of words appears to describe the Passeig de Gràcia in Barcelona. Hence, we can gain insights into what people say about this famous street.

**Word: nois**

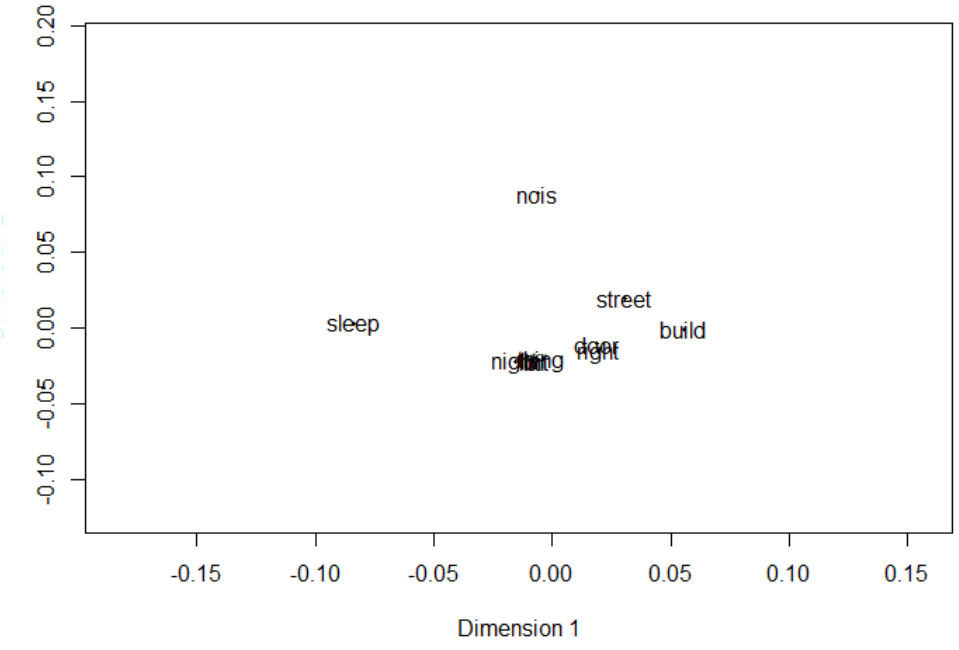


Figure 106: LSA 10 words with highest cosine similarity associated with nois

Here, we can observe that the most similar words to "noise" are "sleep," "street," "building," "little," "night," "door," and "thing." It appears that there are apartments in Barcelona where people frequently complain about noise during the night and from the street. This observation is interesting because it highlights a common problem in Barcelona. It also suggests that when we conduct LDA analysis in the following section, we may discover a topic related to this issue.

Once we have analyzed the words from the corpus, let's examine how documents are related based on their Pearson correlation coefficient. This correlation allows us to measure the linear relationship between two documents and determine if they are closely related or not. In the context of LSA, this coefficient provides insights into the similarity between two documents in terms of their semantic meaning.

Now, let's examine two different reviews from two distinct apartments and calculate the correlation coefficient between them. It's important to note that the Pearson correlation coefficient ranges from -1 to 1. A value of -1 indicates a perfect negative linear relationship between the documents, while a value of 1 suggests a perfect positive linear relationship. When the correlation coefficient is close to 0, it indicates that there is no linear relationship between the documents.

We examined the correlation coefficients between Apartment 1 and Apartment 14, as well as between Apartment 1 and Apartment 21. The correlation coefficient between Apartment 1 and 14 is 0.004115765, while the coefficient between Apartment 1 and 21 is 0.1766911. This indicates that Apartment 1 and 21 share more similarities in their reviews compared to Apartment 1 and 14. To further investigate the factors contributing to this correlation coefficient change, we will analyze the reviews of these three apartments. By identifying the distinguishing features, this way we can extract valuable information for topic differentiation in the future LDA analysis.

After analyzing the reviews from these three apartments, we can see that in Apartment 1 and Apartment 21, people love its facilities and characteristics. They do not mention any negative points about their stay and mention that they would choose the same apartment again if they come to Barcelona.

On the other hand, in Apartment 14, people say that the apartment is basic. Some reviewers mention that they think it is too expensive for what it offers and mention some problems with the internet, moisture in the walls of the bathroom, cleanness, and other issues.

From this analysis, we can predict that in the next topic modeling, we will find a topic related to problems in the apartment and another topic with the good things. Now we proceed with our LDA section.

# 10. Topic modeling with LDA

In this part of our work, we're going to do topic modeling using LDA (Latent Dirichlet Allocation). We will use this algorithm to uncover the different types of reviews that apartments receive from customers.

Our main goal here is to assign a topic to each apartment in our database. By doing so, we will be able to create a new variable that captures the most significant information from people who have stayed in those apartments. Once we've got this new variable in place, our aim is to use it with the apartment's score rating. This way, we can dig deeper into the characteristics that have the greatest impact on the rating rating of apartments in Barcelona.

By doing this analysis, we can gain valuable insights into the key features of each topic. These insights can then be used to improve the overall performance of apartment rentals in Barcelona.

## Model Calculation

As known, LDA is an unsupervised algorithm that works with the hyperparameter "k," which indicates the number of different topics that the algorithm searches for. To decide on the value of the number of topics, we will use the R function FindTopicsNumber from the package 'ldatuning'. This function works by evaluating the coherence of the topics generated by LDA for a given range of numbers of k and selects the number that gives the highest coherence score. The coherence metrics "CaoJuan2009" and "Deveaud2014" ensure the quality and interpretation of the topics. We will set the range of k from 5 to 29 advancing to by 2. This is the resulting plot from the metrics.

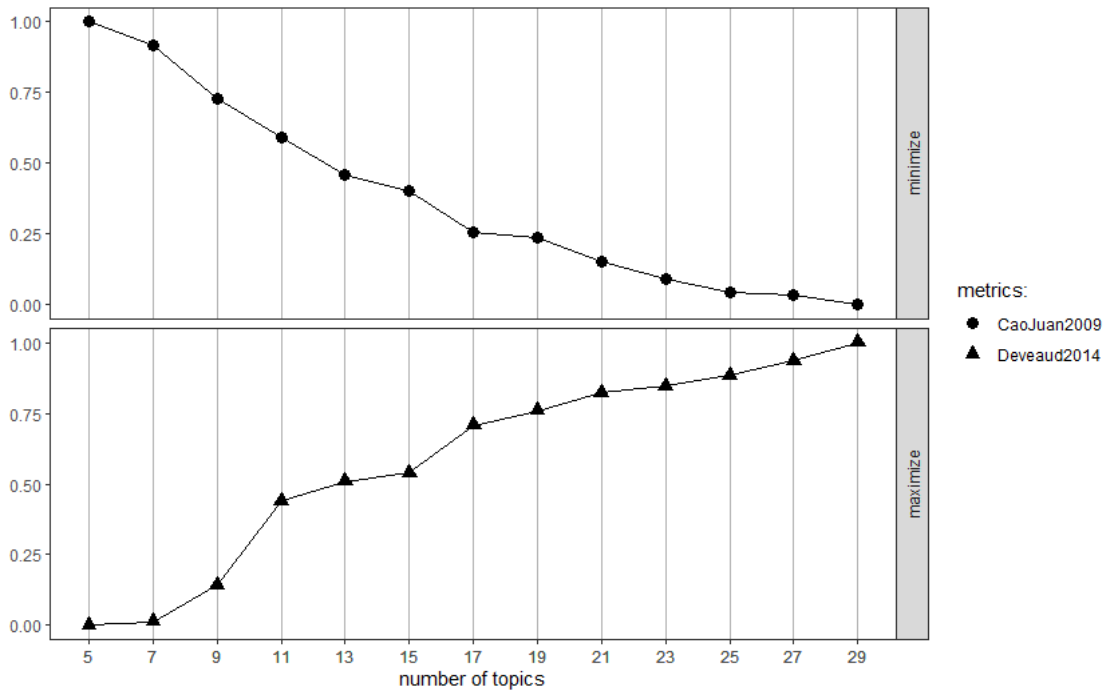



Figure 107: Results of metrics after"CaoJuan2009" and "Deveaud2014" using FindTopicsNumber() R function

The plot shows both of the mentioned metrics. The number of topics is considered better when the CaoJuan2009 metric is minimized and the Deveaud2014 metric is maximized. Based on this plot and the metrics from the R function, the optimal number of topics is determined to be 29. These were too many topics to handle, so we made the decision to start with 17 topics. If we need to raise them during the postprocessing step, we will increase the number.

After executing the LDA with 17 topics we will take the 3 documents studied in the LSA (apartemnts 1, 14 adn 21) and we will see how influential every topic is in each document.

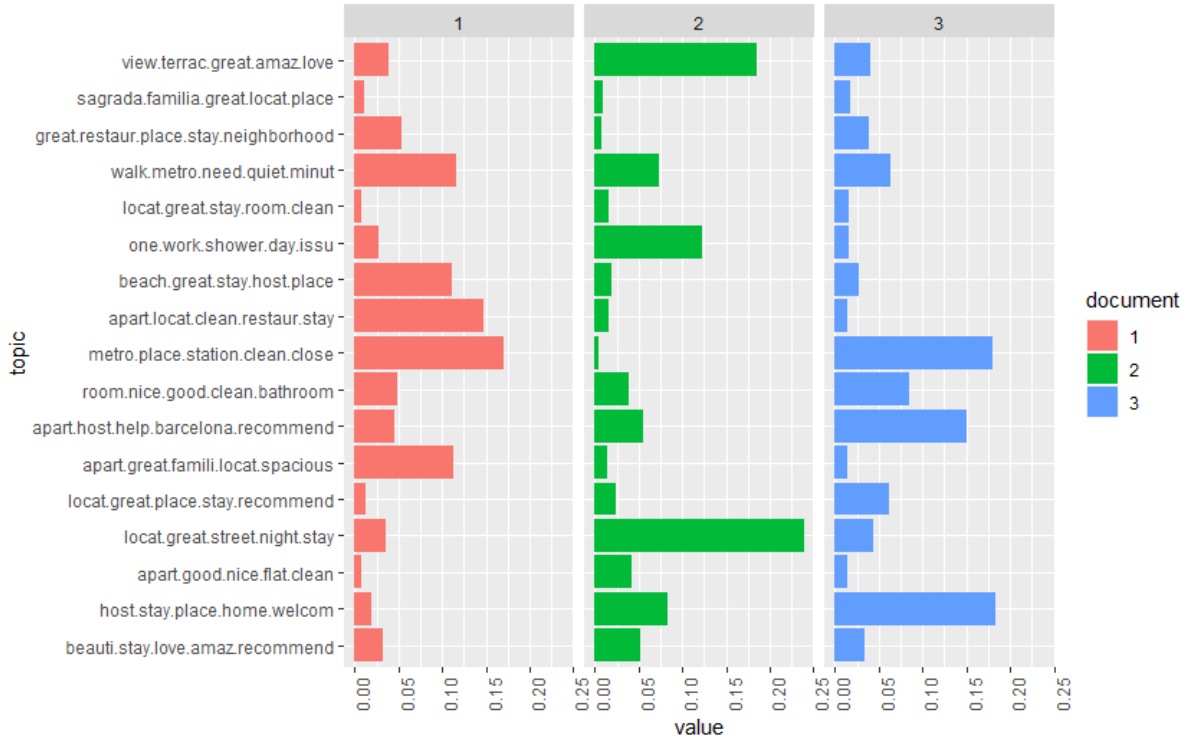



Figure 108: Plot that represents the presence of each topic computed with LDA with alpha 2.94 in the 3 sample documents 1, 14 and 21

Here we can see how there are several topics that influence each document. To solve this we will do the same LDA again but this time with alpha = 1.5. This parameter was previously set to 2.94 and its function is to control the sparsity of the document-topic matrix. A higher alpha value results in documents that are more likely to contain a mixture of many topics, while a lower alpha value biases the documents towards containing fewer topics. This is why we reduced it. Now we obtain the following data.

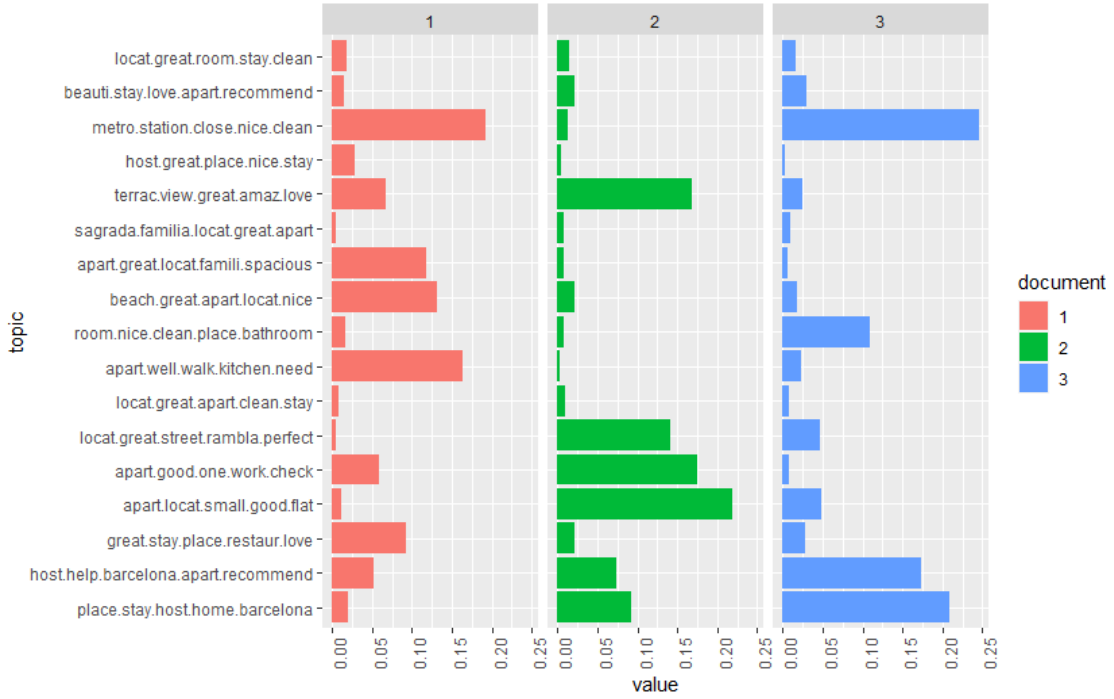



Figure 109: Plot that represents the presence of each topic computed with LDA with alpha 1.5 in the 3 sample documents 1, 14 and 21

As we can see, topics that had a smaller value, now have a lower one, and topics that had a big value, now have a bigger one. But we can continue seeing that there are still ambiguous topics for the same document.

## Post Processing

Now, to know if we need to add or remove topics, we will do a post processing step where we will look at the correlation between documents and topics. This analysis will help us to identify those cases where topics are too similar or overlapping, indicating that the value of k should be increased for better topic differentiation. On the other hand, if topics are too diverse or unrelated, it may indicate that the value of k should be reduced to capture more coherent themes.

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|---|---|---|---|---|---|---|---|---|---|
| 1 | 1.00000000 | 0.100222865 | -4.220619e-02 | -1.276125e-01 | -0.27146538 | -0.08795656 | -0.178665782 | -0.233054575 | 0.22728724 |
| 2 | 0.10022286 | 1.000000000 | 5.711497e-02 | -1.349996e-01 | -0.29166304 | -0.08468494 | 0.021759566 | 0.053466532 | -0.07053401 |
| 3 | -0.04220619 | 0.057114973 | 1.000000e+00 | 2.357495e-05 | -0.16782391 | -0.05506752 | 0.001076845 | 0.096660719 | -0.13666762 |
| 4 | -0.12761255 | -0.134999646 | 2.357495e-05 | 1.000000e+00 | 0.07598736 | 0.07116540 | -0.112419779 | -0.013785090 | -0.07534540 |
| 5 | -0.27146538 | -0.291663043 | -1.678239e-01 | 7.598736e-02 | 1.00000000 | -0.07593521 | -0.087491343 | 0.011782595 | -0.12688126 |
| 6 | -0.08795656 | -0.084684944 | -5.506752e-02 | 7.116540e-02 | -0.07593521 | 1.00000000 | 0.027290822 | -0.051524904 | 0.02432651 |
| 7 | -0.17866578 | 0.021759566 | 1.076845e-03 | -1.124198e-01 | -0.08749134 | 0.02729082 | 1.000000000 | 0.002459378 | -0.19542811 |
| 8 | -0.23305457 | 0.053466532 | 9.666072e-02 | -1.378509e-02 | 0.01178260 | -0.05152490 | 0.002459378 | 1.000000000 | -0.33017740 |
| 9 | 0.22728724 | -0.070534008 | -1.366676e-01 | -7.534540e-02 | -0.12688126 | 0.02432651 | -0.195428109 | -0.330177396 | 1.00000000 |
| 10 | -0.02080409 | -0.054479075 | -3.403540e-02 | -1.073363e-02 | -0.08296773 | -0.05308314 | -0.139383732 | -0.010575036 | -0.08348134 |
| 11 | -0.25336883 | -0.080519339 | -6.105069e-02 | -5.504679e-02 | 0.06707480 | -0.03350756 | 0.032314059 | 0.050617424 | -0.20648743 |
| 12 | -0.09228819 | -0.009037112 | -1.029299e-01 | -5.627458e-02 | -0.06197601 | -0.17880912 | -0.131507342 | 0.013752559 | -0.06692700 |
| 13 | -0.02491563 | -0.085257440 | -4.297008e-02 | -3.955332e-03 | -0.10462657 | -0.11069358 | -0.095280290 | -0.048064438 | -0.13456415 |
| 14 | 0.04943247 | -0.075966116 | -8.826359e-02 | -3.365749e-02 | -0.14385936 | -0.07207785 | -0.109630971 | -0.167139217 | 0.09087066 |
| 15 | 0.04941102 | 0.098145791 | -1.821178e-02 | -7.335470e-02 | -0.17498190 | -0.23496337 | -0.026386175 | 0.025778128 | 0.02890837 |
| 16 | 0.00120124 | 0.034547166 | 4.087633e-02 | -1.440785e-01 | -0.17764508 | 0.01400678 | 0.116063697 | 0.053585921 | -0.22743350 |
| 17 | -0.10969388 | -0.200385166 | -1.331529e-01 | -1.038665e-01 | -0.06115592 | 0.03411680 | -0.107204796 | -0.220488679 | 0.05618208 |

| | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 |
|---|---|---|---|---|---|---|---|---|
| 1 | -0.02080409 | -0.25336883 | -0.092288191 | -0.024915635 | 0.04943247 | 0.04941102 | 0.00120124 | -0.10969388 |
| 2 | -0.05447908 | -0.08051934 | -0.009037112 | -0.085257440 | -0.07596612 | 0.09814579 | 0.03454717 | -0.20038517 |
| 3 | -0.03403540 | -0.06105069 | -0.102929938 | -0.042970082 | -0.08826359 | -0.01821178 | 0.04087633 | -0.13315285 |
| 4 | -0.01073363 | -0.05504679 | -0.056274575 | -0.003955332 | -0.03365749 | -0.07335470 | -0.14407846 | -0.10386650 |
| 5 | -0.08296773 | 0.06707480 | -0.061976009 | -0.104626570 | -0.14385936 | -0.17498190 | -0.17764508 | -0.06115592 |
| 6 | -0.05308314 | -0.03350756 | -0.178809116 | -0.110693577 | -0.07207785 | -0.23496337 | 0.01400678 | 0.03411680 |
| 7 | -0.13938373 | 0.03231406 | -0.131507342 | -0.095280290 | -0.10963097 | -0.02638617 | 0.11606370 | -0.10720480 |
| 8 | -0.01057504 | 0.05061742 | 0.013752559 | -0.048064438 | -0.16713922 | 0.02577813 | 0.05358592 | -0.22048868 |
| 9 | -0.08348134 | -0.20648743 | -0.066927004 | -0.134564151 | 0.09087066 | 0.02890837 | -0.22743350 | 0.05618208 |
| 10 | 1.00000000 | -0.06924547 | -0.091652084 | -0.015004886 | 0.01930323 | -0.06642624 | -0.09296592 | -0.08438234 |
| 11 | -0.06924547 | 1.00000000 | 0.014824690 | -0.020074560 | -0.11962724 | -0.10766437 | 0.07389542 | -0.11754737 |
| 12 | -0.09165208 | 0.01482469 | 1.000000000 | 0.054252874 | -0.04734352 | -0.06610353 | -0.05679207 | -0.04429155 |
| 13 | -0.01500489 | -0.02007456 | 0.054252874 | 1.000000000 | -0.06169916 | -0.11073154 | 0.03999356 | 0.01514536 |
| 14 | 0.01930323 | -0.11962724 | -0.047343516 | -0.061699165 | 1.00000000 | 0.01545026 | -0.13002005 | -0.10575672 |
| 15 | -0.06642624 | -0.10766437 | -0.066103531 | -0.110731538 | 0.01545026 | 1.00000000 | -0.22025999 | -0.09650872 |
| 16 | -0.09296592 | 0.07389542 | -0.056792067 | 0.039993562 | -0.13002005 | -0.22025999 | 1.00000000 | -0.03956024 |
| 17 | -0.08438234 | -0.11754737 | -0.044291547 | 0.015145360 | -0.10575672 | -0.09650872 | -0.03956024 | 1.00000000 |

Figure 110: Correlation matrix between the 17 topics

After examining the correlation matrix, it becomes evident that there are no significant correlations between topics. The correlation values fall within the range of (-0.3, 0.3), indicating that there is no need to increase the number of topics (K). Now, our attention will be directed towards the topics generated by LDA, specifically by analyzing the top 10 words with the highest beta values in each topic.

# Visualization of words and topics

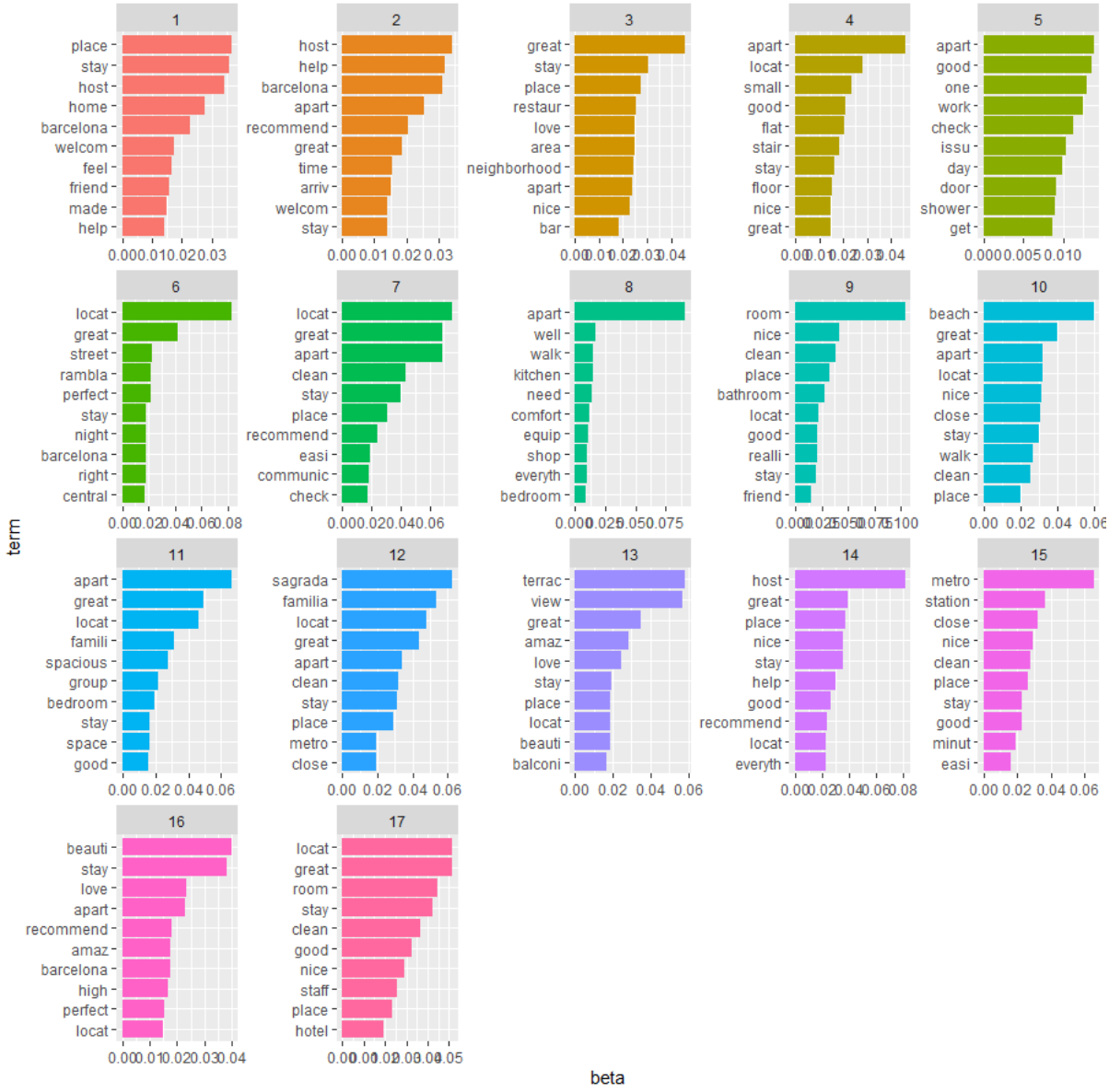



Figure 111: 10 words sorted by beta value in the 17 documents

In the above plot we can see all the topics with their more representative words. Using this information we can label each topic automatically selecting the first five words of each topic. If we do this we obtain the following result.

| | |
|---|---|
| **Topic 1** | place stay host home barcelona |
| **Topic 2** | host help barcelona apart recommend |
| **Topic 3** | great stay place restaur love |
| **Topic 4** | apart locat small good flat |
| **Topic 5** | apart good one work check |

| | |
|---|---|
| **Topic 6** | locat great street rambla perfect |
| **Topic 7** | locat great apart clean stay |
| **Topic 8** | apart well walk kitchen need |
| **Topic 9** | room nice clean place bathroom |
| **Topic 10** | beach great apart locat nice |
| **Topic 11** | apart great locat famili spacious |
| **Topic 12** | sagrada familia locat great apart |
| **Topic 13** | terrac view great amaz love |
| **Topic 14** | host great place nice stay |
| **Topic 15** | metro station close nice clean |
| **Topic 16** | beauti stay love apart recommend |
| **Topic 17** | locat great room stay clean |

Figure 112: Table with the automatic names generated using the 5 words with bigger beta value in each topic

As observed in both the table and the figure displaying the 10 most significant words per topic, there is a wide variety of topics. However, some of them overlap to a certain extent, making it difficult to distinguish between them. For instance, topics 1 and 2 share very similar words, as do topics 6 and 7. One possible solution to address this issue is to increase the value of K, but this would lead to a larger number of topics, which can be challenging to handle.

On the other hand, there are numerous topics that are well-differentiated and offer valuable insights for our project. For instance, topic 4 pertains to small, well-located apartments with positive reviews. Topic 13 focuses on apartments with terraces and views, topic 15 relates to apartments in close proximity to metro or train stations, and topic 10 references apartments located near the beach. These are just a few examples that can be found in the table. Here are the word clouds with 20 words from the topics that we've found most interesting.


Figure 113 : Topic 4 word cloud


Figure 114 : Topic 10 word cloud


Figure 115: Topic 13 word cloud


Figure 116: Topic 15 word cloud


Figure 117: Topic 12 word cloud


Figure 118 : Topic 14 word cloud


Figure 119 : Topic 9 word cloud


Figure 120: Topic 16 word cloud

## Topic ranking

As mentioned before, there are some topics that, by their names, do not provide valuable information because these words are very common throughout the entire corpus and are not specific to that topic. To solve this problem, we use what is called re-ranking. The concept of re-ranking terms shares similarities with the concept of TF-IDF. When a term appears frequently in the top levels relative to its probability, it becomes less meaningful for describing the topic. Consequently, the scoring approach prioritizes terms that are more relevant for representing a topic.

Topics names with re-ranked words to be more specific in each topic:

| | |
|---|---|
| **Topic 1** | home host welcom feel hous |
| **Topic 2** | host help gave arriv welcom |
| **Topic 3** | neighborhood restaur bar great gracia |
| **Topic 4** | apart stair small floor flat |
| **Topic 5** | work issu key door check |
| **Topic 6** | locat rambla street right great |
| **Topic 7** | apart locat great clean easi |
| **Topic 8** | apart kitchen equip bedroom street |
| **Topic 9** | room bathroom nice clean share |

| | |
|---|---|
| **Topic 10** | beach close nice apart great |
| **Topic 11** | apart famili group spacious locat |
| **Topic 12** | sagrada familia locat apart clean |
| **Topic 13** | terrac view amaz beauti balconi |
| **Topic 14** | host nice good alway help |
| **Topic 15** | metro station close nice clean |
| **Topic 16** | beauti stylish decor wonder amaz |
| **Topic 17** | room staff locat hotel great |

Figure 121: Table with the automatic names generated using the 5 words with most importance in each topic using the re-ranking method

As shown in the table, the topic names that were previously too general have become more specific, providing us with more information about each topic. For instance, topic 16, which previously had the name 'beauti stay love apart recommend', has been updated to 'beauti stylish decor wonder amaz', which clearly emphasizes the beauty and style of the apartment. Another example is topic 5, which used to be named 'apart good one work check' and is now 'work issu key door check'. We can observe how its meaning has changed, indicating that this topic may contain reviews related to issues and material problems.

The same improvement can be seen across all topics. The words are now less overlapping and similar, allowing us to better identify the specific information associated with each topic. This re-ranking technique has greatly enhanced our topic labels. This step will be crucial when analyzing the apartment's score rating in relation to their respective topics, as it will enable us to draw more accurate conclusions. In addition, now we can clearly see how the majority of topics are related to good things except topic 5.

At this moment we have finally set our topics labels. Now we will apply what is called method Rank-1 to sort our topics based on how often they appear as primary topics in the reviews from the apartments.

```
 [1] "578 : work issu key door check"              "333 : apart locat great clean easi"
 [3] "304 : room bathroom nice clean share"        "300 : home host welcom feel hous"
 [5] "236 : host nice good alway help"             "223 : metro station close nice clean"
 [7] "217 : room staff locat hotel great"          "180 : neighborhood restaur bar great gracia"
 [9] "176 : beach close nice apart great"          "171 : apart famili group spacious locat"
[11] "166 : host help gave arriv welcom"           "162 : sagrada familia locat apart clean"
[13] "160 : apart kitchen equip bedroom street"    "149 : locat rambla street right great"
[15] "127 : apart stair small floor flat"          "119 : terrac view amaz beauti balconi"
[17] "117 : beauti stylish decor wonder amaz"
```

Figure 122: Sorted topics based on Rank-1 method

As we can observe in the image above, the most frequent topic is 'work issu key door check'. This is quite interesting because it appears to be the only negative topic, and we can extract valuable information by examining if the score rating variable for apartments labeled with this topic is lower compared to the others. Regarding the remaining topics, we can see that all of

them have a significant representation. There are no topics with very low or extremely high representation.

After completing the creation and analysis of our topics, we will add a topic variable to our dataset. Subsequently, we will conduct an analysis to explore how other variables are influenced by this new topic variable.

## Topics combined with original data

The first thing we've done in this section is to add the finded topics with LDA to our dataset as a new variable. Now we are going to look at how the mean of the score rating is influenced by the topic of the apartment, we expect to find interesting relationships in this analysis.

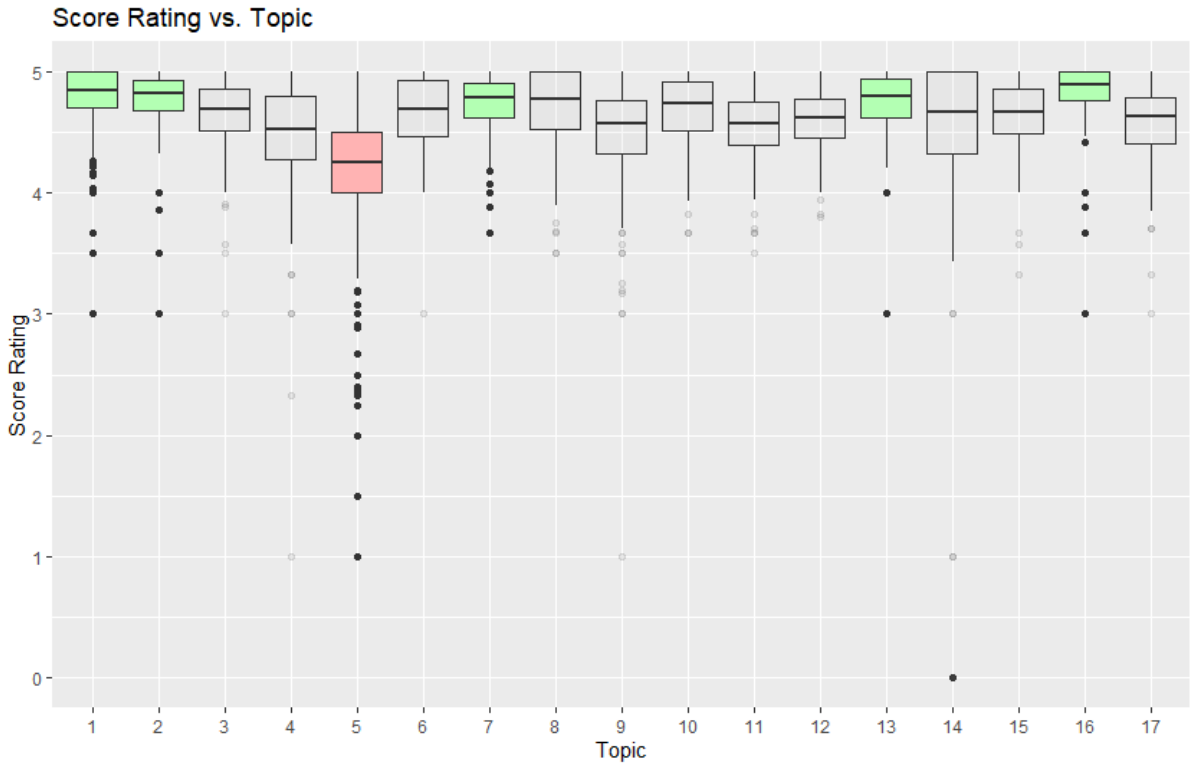



Figure 123: Boxplots of the r_sco_rt variable depending on the topic, highest and lowest scores highlighted

Here, we can observe that the mean of all scores in all topics falls between 4 and 5. This happens because the mean score of all the apartments in the dataset is also very high, ranging between 4 and 5. However, significant changes can be seen in certain topics, specifically in topics 1, 5, and 16. In topics 1 and 16, we can observe a positive influence on the ratings, indicating that when apartments belong to topic 1 or 16, the reviews tend to be higher compared to other topics. Conversely, the opposite occurs with topic 5. As mentioned earlier, this topic is related to problems in apartments, and it is reflected in our plot. We can clearly see that when apartments belong to topic 5, the value of their score rating decreases.

We can also observe that topics 2 (named 'host help gave arriv welcom') , 7 (named 'apart locat great clean easi') , and 13 (named 'terrac view amaz beauti balconi') have a positive impact on the score rating. The remaining topics do not show a significant change in the

rating; all of them seem to be positive topics, so it is understandable that the reviews are highly rated.

Now, let's delve into the highlighted topics to specifically examine the terms mentioned by people in their reviews. This way, we can extract valuable information about what apartment owners should focus on to increase their rating and attract more people. To study these topics we are going to look at the 20 most influential words after doing re-ranking to give more importance to representative words from the topic.

Let's see the word cloud from the **negative** topic 5 named 'work issu key door check':

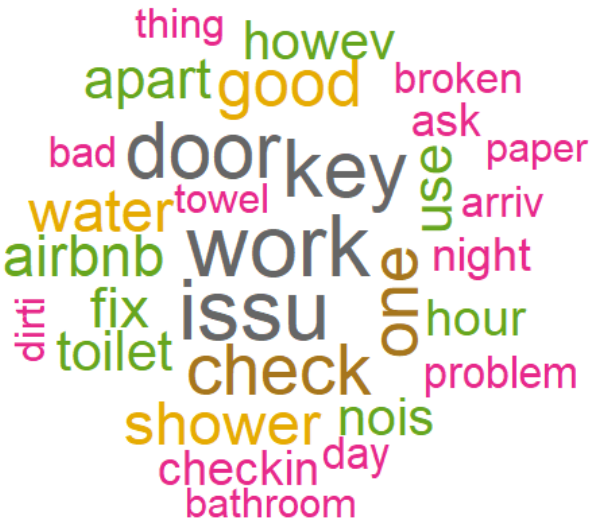



Figure 124: Word cloud of size 30 from topic 5 with re-ranked order in words

As we can see in this topic, we have access to the most influential words. As shown in the plot, we can observe people's complaints regarding words such as "fix," "issue," "broken," "dirty," and "noise." Additionally, we can identify the specific material things they complain about, including the door, key, shower, water, dirty areas, check-in process, bathroom, toilet, and others.

Now that we have gathered all this information, we can provide advice to apartment owners regarding what is important to keep under control in their apartments if they want to avoid receiving bad reviews. From these 30 words, we can deduce that problems with the key, door, and check-in process are common reasons for negative reviews. Similarly, issues related to cleanliness, broken amenities, and noise are also frequently mentioned.

Now we are going to study positive topics, in particular topics 1, 7, 13 and 16. To study them we are going to visualize the 10 most important words of each topic.


Figure 125: Word cloud topic 1


Figure 126: Word cloud topic 7


Figure 127: Word cloud topic 13


Figure 128: Word cloud topic 16

After examining the most relevant words for each topic, we can understand the reasons why each one receives a good rating on Airbnb. In the first topic, it is evident that the good rating is attributed to the host's treatment of guests. We can observe positive terms such as host, home, welcome, feel, kind, make, hospitality, wonder, and others.

In the seventh topic, the influential words are location, apartment, clean, easy, and communication. This suggests that apartments belonging to this topic receive high ratings due to their favorable location, ease of access, cleanliness, and effective communication.

The thirteenth topic focuses on words like terrace, view, balcony, patio, roof, and top. These words indicate that apartments in this topic offer beautiful views, which contributes to their high ratings.

Lastly, the sixteenth topic is related to apartment decor and style. As reflected in the word cloud, important terms include beautiful, stylish, decor, wonderful, amazing, spacious design, and similar expressions.

Once we have all this information, we can provide advice to apartment owners on how to maximize their ratings. To achieve that, we would recommend focusing on improving two key factors: the apartment's decor and the treatment of guests. These two aspects are not overly costly and can greatly enhance the ratings of their apartments.

## Real applications of our LDA

Finally, we will mention two possible uses of our LDA analysis. Firstly, as we have mentioned throughout the work, one practical application would be to assist apartment owners in Barcelona in improving their apartment reviews and ratings, thereby attracting more guests and maximizing their occupancy.

The second valuable application of our LDA analysis would involve utilizing the newly created topics variable in our dataset to construct predictive models for relevant data.

# 11. Geospatial analysis and Geostatistics

## Type-1 Data Analysis

In this subsection we are going to present the work that we have done in the field of geostatistics. First of all, we are going to talk about Type 1 data and its modelisation. Type 1 data is the data that is considered to be continuous along a limited or unlimited region of space such that between two points $P_1$ and $P_2$ there are infinite points. That's the definition of continuity. The aim of studying this kind of variables and its relation with spatial data or space is to have an intuitive understanding of whether this variable and its values depends on spatial location or not.

To do so, two main processes are applied in this study:

- The construction of a model variogram based on the empirical variogram and the theoretical variogram to end up having a function that relates the distance between two points and the variability of the variable that we are studying at each point.
- The application of a kriging predictor to predict, in a continuous space, which is the variance of the variable at each of the points.

## Semivariogram

To compute the distance between two points another position scale is needed (far from longitude and latitude scale) so that the model could be able to build the semivariogram/variogram given a certain continuous variable In our case, we wanted to study whether the **Price of the apartments** is related to their geospatial position. This variable fits well with the needs that we have because it is continuous and it is somehow logical to think that it could be related to space.

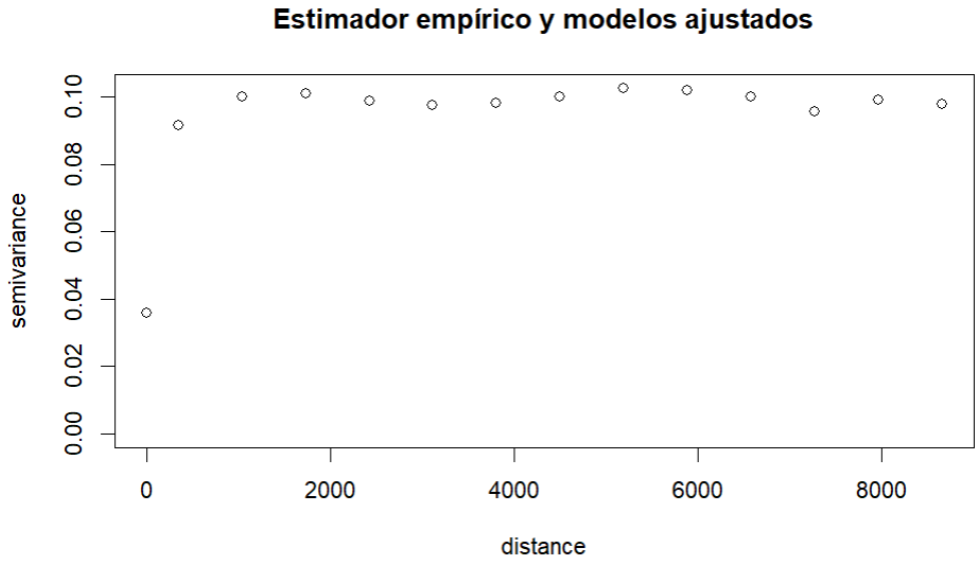
First of all we transformed our Latitude/Longitude spatial data to UTM 31N Zone (referring to Barcelona) scale to be able to compute the distance between two points. After this, we log-transformed our **Price** variable to avoid having the scalability problem where some

points have more relevance than others just by their high value and not by their informative meaning.

After doing this, we did a statistical test to check whether the price variable is independent from space or not. Its null hypothesis is that it is independent so if we get a *p-value <= 0.05* it would mean that it is dependent. The results, as shown in the Figure below, show that there is not a significant dependency between price and space. This means that the further study that we are going to apply to this data would not fit in the real world, but it could be useful as a case study to improve our knowledge in the field.

```
> sm.variogram(coordinates(dd), dd$logprice, model = "independent")
Warning: weights overwritten by binning
Test of spatial independence: p =  0.589
```

After this, we can plot the variogram of some distance-representative instances of our data that shows how the variability changes when going from one distance to another. The figure below shows this first empirical variogram.

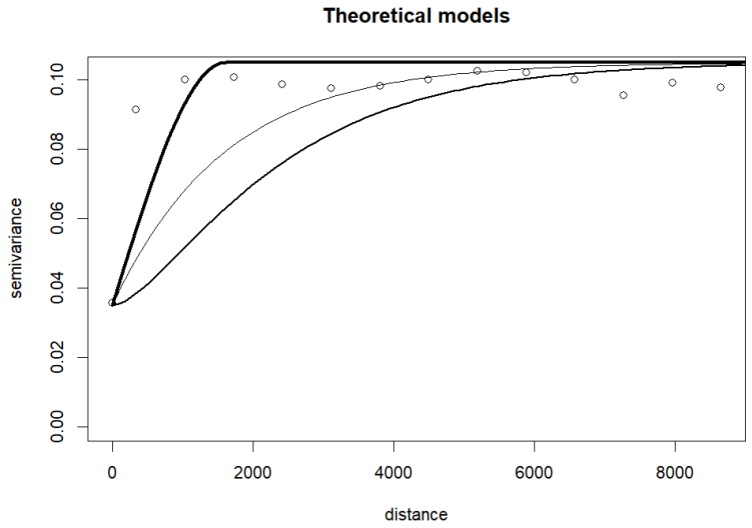



When plotting the empirical variogram (the one based on observed data) we can see whether the variable shows some stationarity or some trend. In our case it is stationary because there exists some point at which the variability stabilizes at some semivariance value. We can also see the following features:

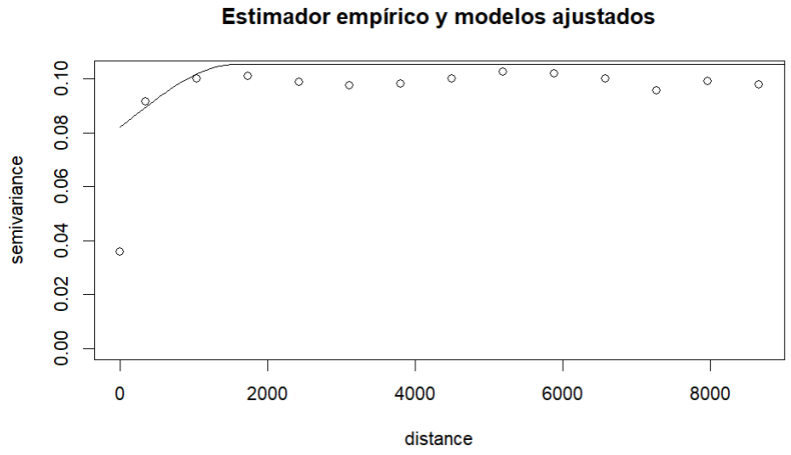
- **Range = 1600m**
- **Nugget = 0.035**
- **Total-sill = 0.1**
- **Partial sill = Total-sill - Nugget = 0.07**

With this information one could plot different theoretical variograms (spherical,exponential,normal…) to see which is the one that fits better with our empirical variogram. In our case is the spherical theoretical variogram (with the feature mentioned

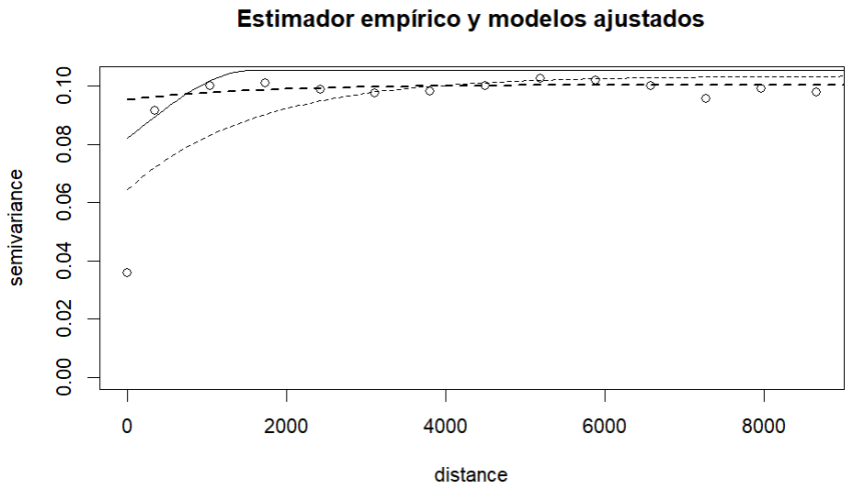
above) the one that fits better with our data, as it's shown below (the one with the widest line). The thinnest one represents the exponential model and the medium one represents de Gaussian model.




Once we have set the best theoretical model, thus being the **Spherical** one, we must adjust a variogram model to fit our empirical model with the theoretical chosen characteristics. The optimiser that we have chosen to do so is the **ML (maximum likelihood)** estimator to better fit the data. Having done this, the adjusted line looks as following:

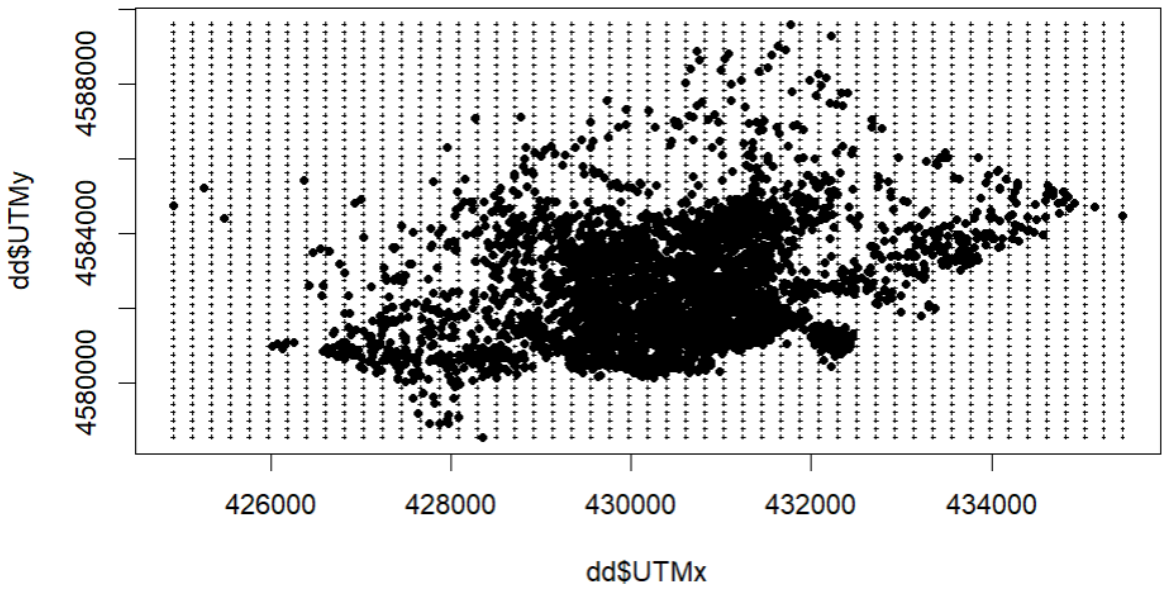



As we can see in the next Figure below some other models have been tried to be adjusted but the Spherical conditions with the adjusted model is the one which fits better to our empirical data.

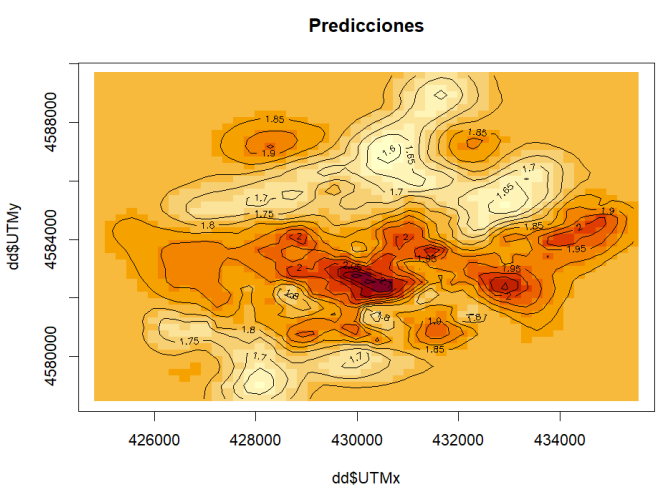


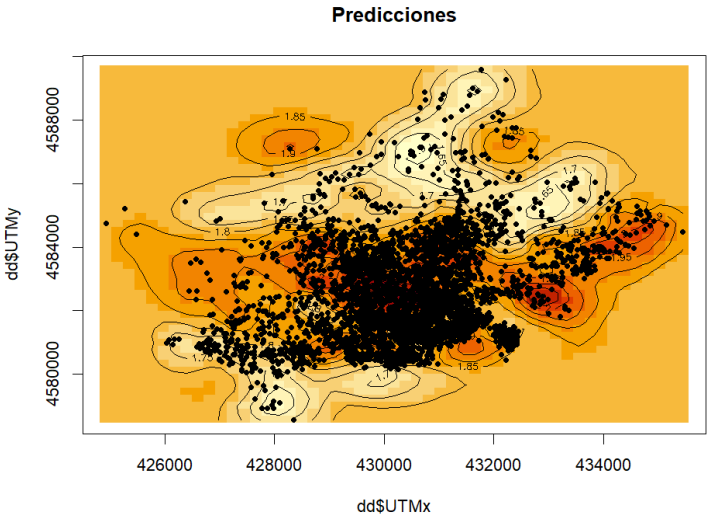

## Kriging Predictor

Once we have set our adjusted variogram model to predict the variance at any point in the region of study we must create and inference with our predictor. Before doing so, what we have done is a grid covering all the box of points determined by the minimum and maximum values of x-position and y-position. This grid of points will be the points at where we will make a prediction about the log price value and the variance at those points given the information of the known points. At the image below we can see how does the grid of points look versus the observations.



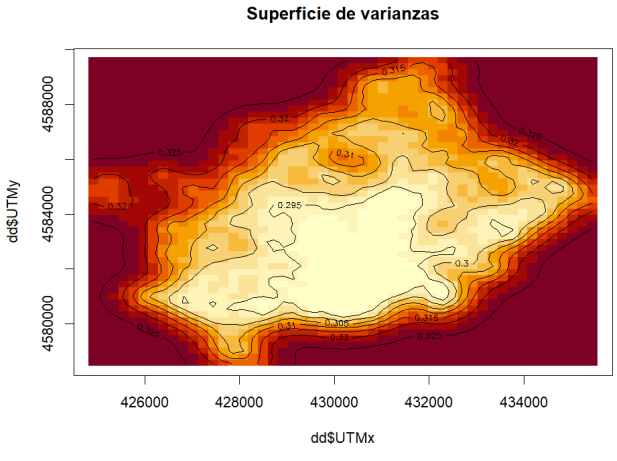

With this said, we must apply an ordinary Kriging prediction on the grid points to see which are their predicted log-price values. The results are shown in the heatmap below, with the contour values.

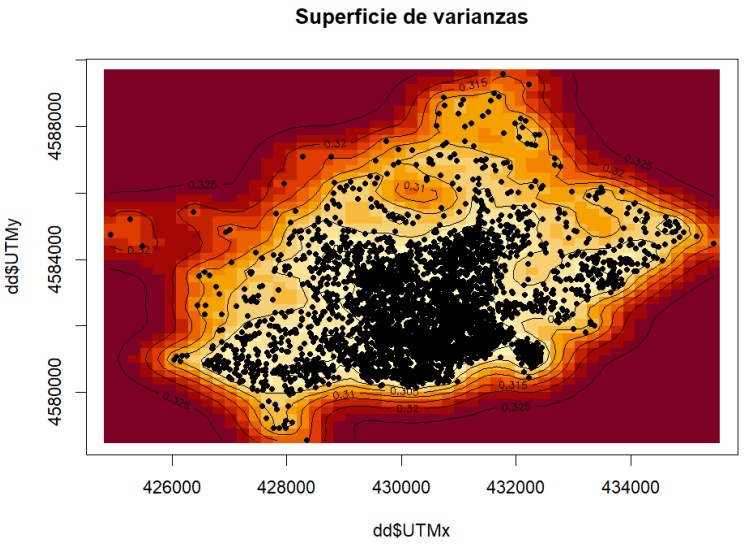



As we can see in the two plots above, the places with higher levels of log-prices values are the ones that in a Map would correspond to beach-near positions and centric places. Also, it does match with those places with a higher level of observed values, thus meaning with a higher level of certainty and empirical knowledge. As we have said before, these results would not be analyzable in the real world due to the lack of dependency in space.

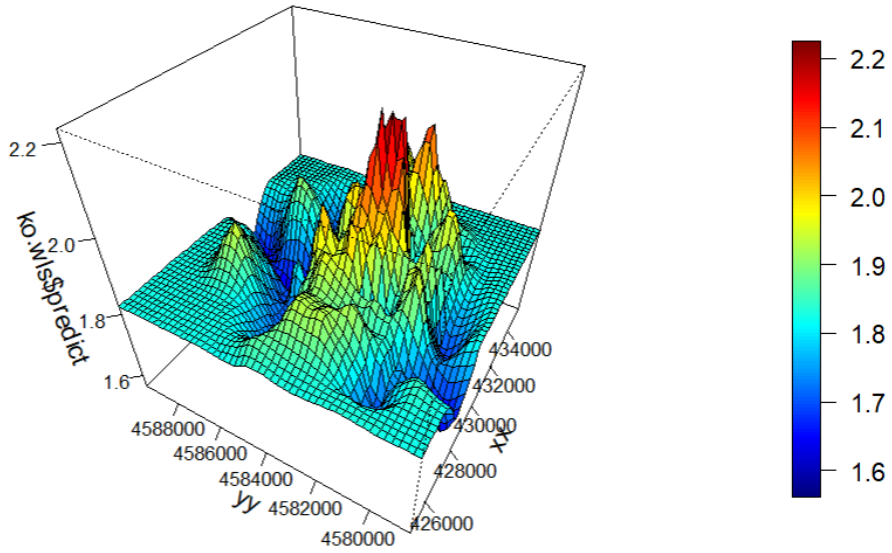
Moreover, we could plot the area of variances, thus representing which zones have predictions with more certainty and low variance (yellow) and which ones have more uncertainty and more variance (the red ones)




As we can see in the two plots above, the places with more certainty perfectly match those places where there are a lot of empirical observations, places where this certainty is obvious. Out of these radius of certainty we no longer encounter places with high certainty, thus reflecting the lack of spatial dependency in our data.

At last but not least, we can see a 3D plot of the predicted log-price levels against each one of the position components. It is interesting to see that in this plot the highest values for the log-price also match with the more expensive and demanded places such as the more observed ones.




## Type-2 Data Analysis

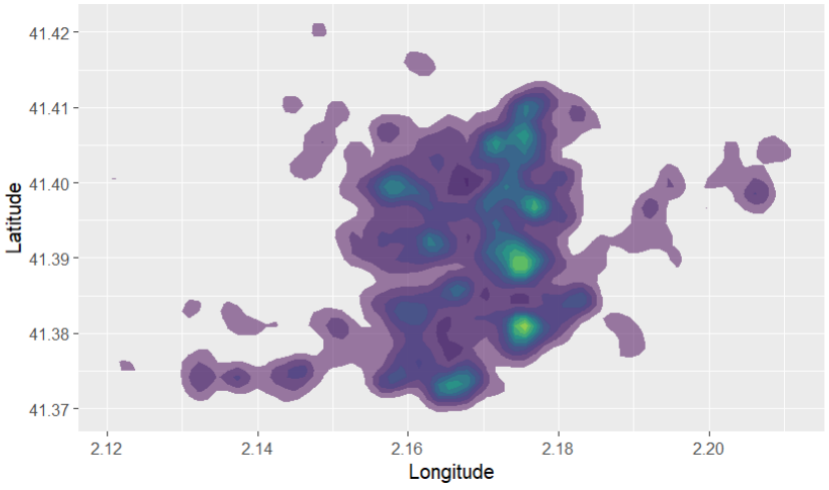
In this second part of the geospatial and geostatistical topics we are going to work with Type-2 Data. This kind of data is characterized by being an event-representing data or punctual-successes data such as crimes, deaths, reviews (that is our case). It handles discrete data (in contraposition with Type-2 Data). This kind of data doesn't have a very complicated background (statistically talking) but it does give us a lot of information like the intensity or the density of some kind of countable event/action as the same time as it is plotted in a detailed map where an expert could watch and observe some of the relations between density of the events and spatial information (neighborhood, temperature of that specific place…).

In our case we have decided to work with the event **REVIEW** of an apartment. This decision follows two reasons: the first one is because this kind of data fits perfectly with the description of Type-2 Data as we could plot the density of the reviews distributed in Barcelona (the region of the space that we are actually working in); the second one is because we have time data about these reviews, which could give us the eventual possibility to add a third dimension to our work and see how thesis distribution of densities and intensity has evolved during the time. To do so, we first need to have some kind of Dataset that is not usual. We need a **transactional** Dataset, which means that we need to have rows

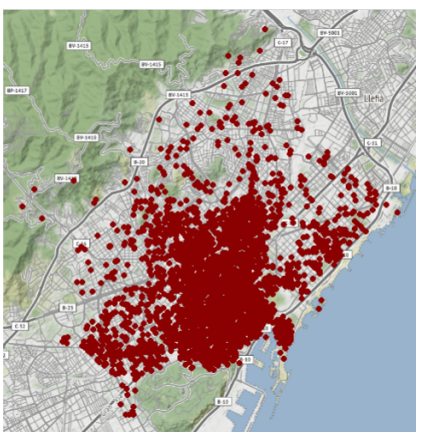
representing each of the events in the form: LOCATION; EVENT; DATE-TIME. This has been resolved by merging two open datasets provided by airbnb: one having LOCATION + EVENT + ID_REVIEW and the other one having EVENT+ID_EVENT+DATE-TIME.

| | id | lt | ln | date |
|---|---|---|---|---|
| 1 | 10009907 | 41.39039 | 2.18012 | 2017-03-13 |
| 2 | 10009907 | 41.39039 | 2.18012 | 2017-06-30 |
| 3 | 1002372 | 41.39732 | 2.20693 | 2018-01-20 |
| 4 | 1002372 | 41.39732 | 2.20693 | 2017-06-04 |
| 5 | 1002372 | 41.39732 | 2.20693 | 2017-07-04 |
| 6 | 1002372 | 41.39732 | 2.20693 | 2017-02-11 |
| 7 | 1002372 | 41.39732 | 2.20693 | 2016-06-20 |
| 8 | 1002372 | 41.39732 | 2.20693 | 2015-03-28 |
| 9 | 1002372 | 41.39732 | 2.20693 | 2014-06-02 |
| 10 | 1002372 | 41.39732 | 2.20693 | 2015-10-17 |
| 11 | 1002372 | 41.39732 | 2.20693 | 2017-06-01 |
| 12 | 1002372 | 41.39732 | 2.20693 | 2017-05-23 |

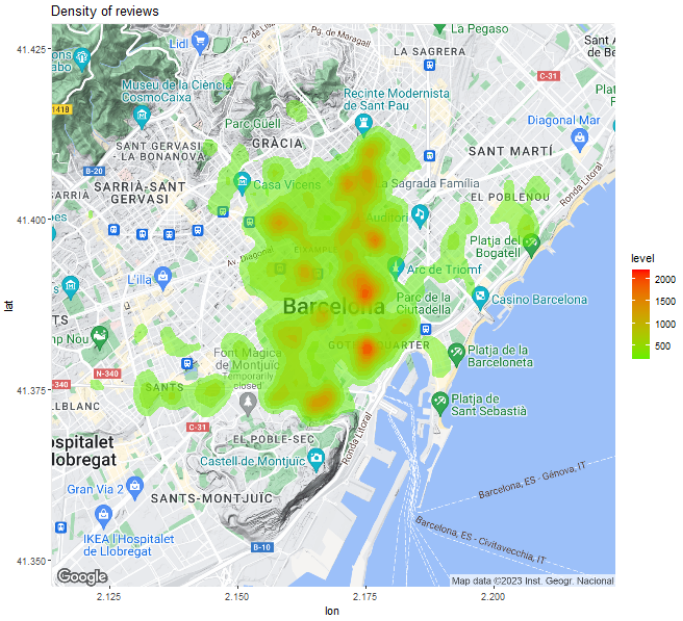
Once we have our transactional data, we can now plot in two normal axis how does the distribution of all the events looks, alongside with its density and intensity:




To gain some informativity with the plot we could get a terrain map of Barcelona and plot the reviews as points (not as a intensity/density cloud) where we could see:

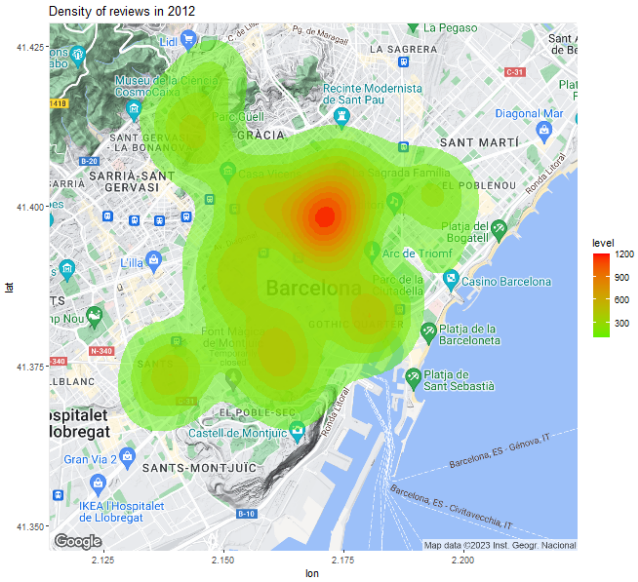


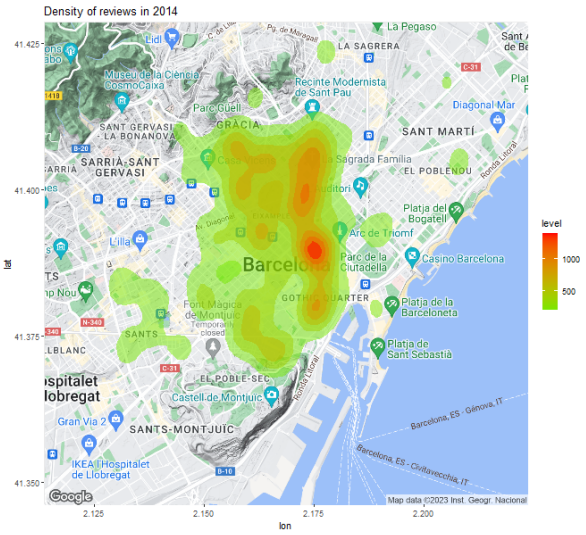
But in this plot we can't even see the city properly nor the neighborhoods, which can be a good part of discussion here. Thus, we have registered to the Google API to get some Google Maps Static API photographs of Barcelona thus having a lot more detail in the spatial analysis. In the next plot we can see the result of the density plot in this Map given by Google Maps:

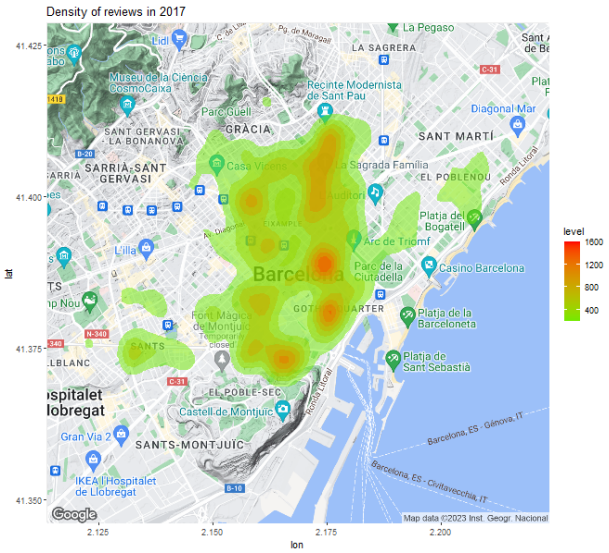


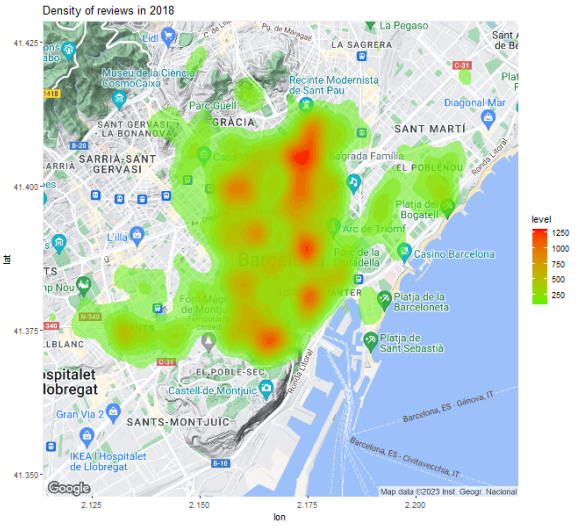

In the Map of the above Figure we can clearly visualize which have been the most reviewed points of the city between the years 2011 and 2023. We can see that the Gothic Quarters are so visited as they're center-positioned and near to the beach. We can also see that places like Eixample and Sarrià have had high levels of reviews during all these years. Furthermore it is remarkable the intensity of reviews that have happened in places like Ciutadella, Platja del Bogatell or Barceloneta, which are very touristic places. Apart from that, in low levels we see more residential or non-touristical places that have less reviews because there are not much changes between the hostages or they're not so visited.

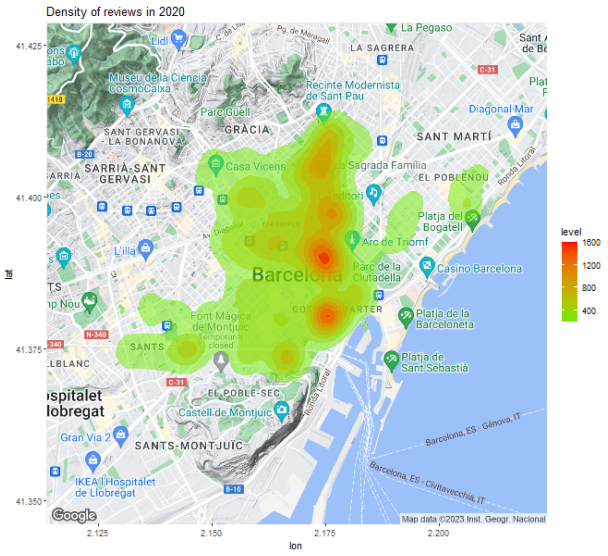
As we said before, we have another kind of Data that could be interesting to analyse: the time. With this, we could make different geostatistical analysis (e.g every different year) to see how have the reviewed points evolved/changed along the last ten or twelve years. This kind of study is so interesting because with so little knowledge and so little amount of data we can extract so much information. In our case, before presenting different pictures of the review-density graphs during the years in Barcelona, we must say that we have created some kind of video (images put together) to visually be able to see how the density cloud has evolved in years as we look at its movement. This video is obviously attached to the delivery. Below we show the most relevant results from all the years:

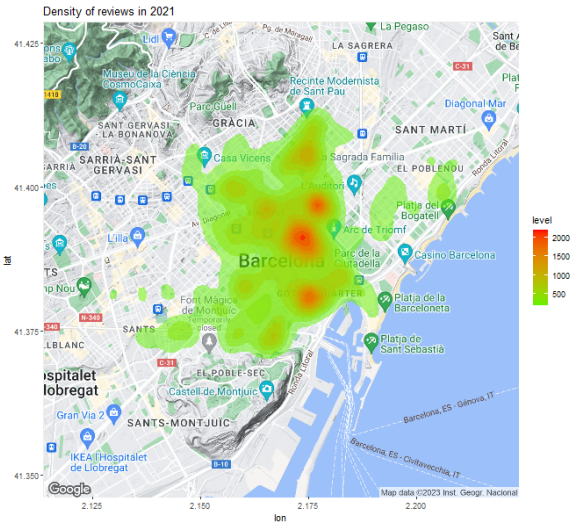


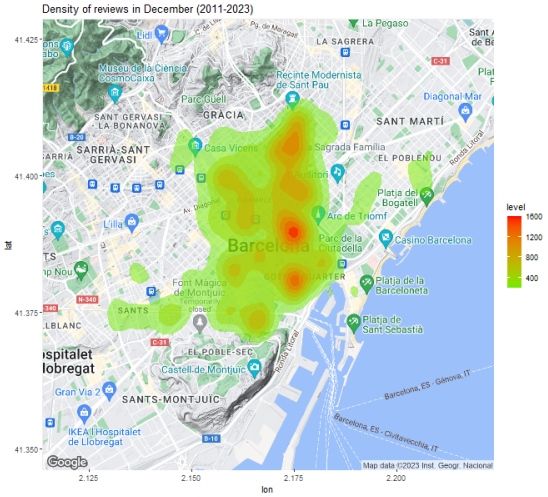



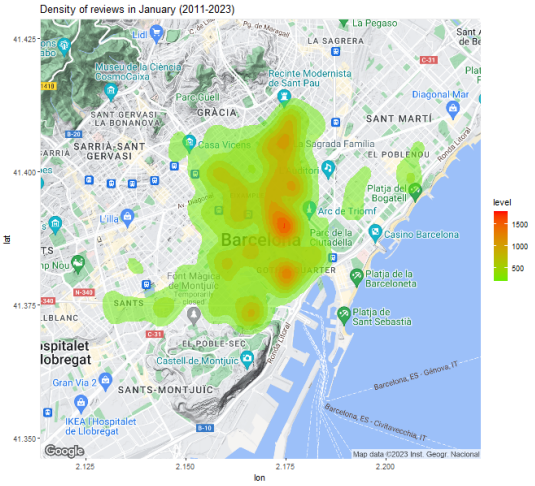



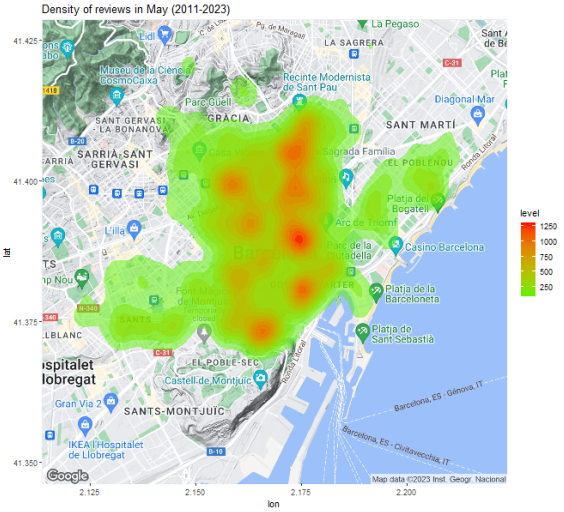

As we can see in the previous plots, there is not a clear tendency of change between the

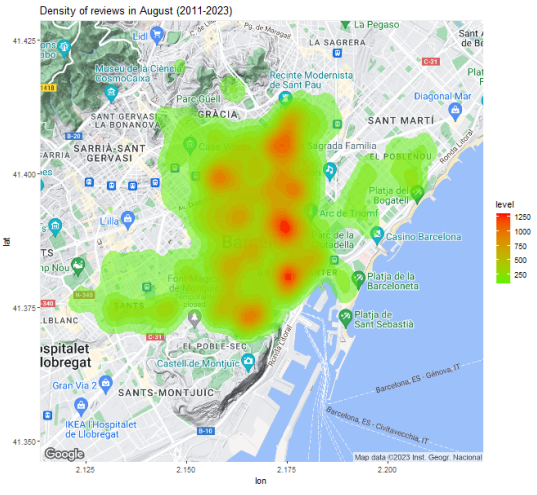
most reviewed points along all the years. What we can mention is that in the early years of study the density of reviews was more sparse and has turned more specific through the years, ending with a lot of intensity in Eixample, Barceloneta and Sarrià. Another amazing thing to see is that the Covid effect is not as substantial as other studies say because as we can see, the level and the map has remain constant during the pandemic years.

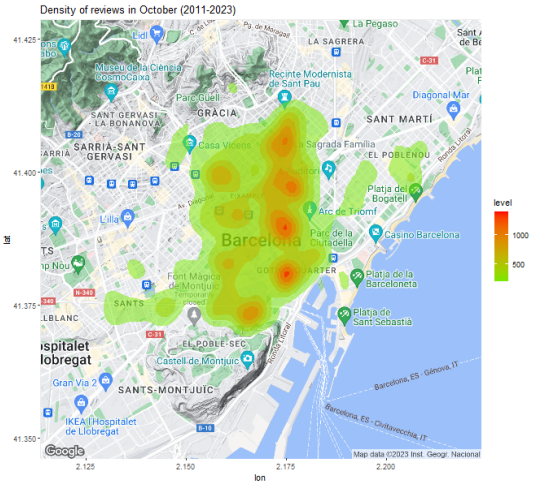
Another interesting matter of study is the seasonal impact in the reviewed points in Barcelona city. As we have the data of the different months along all the years what we can do is to collect all the reviews done in (January, February,...,December) for all the years to see if seasonality changes the position of the high and low densities. The images below show this study:

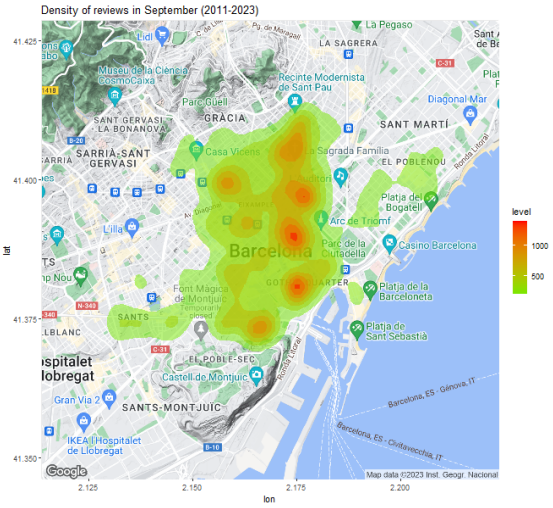





As we can see, there is an important effect of seasonality in the density or presence of reviews in beach zones such as Barceloneta, Ciutadella or Bogatell: there is a lot more presence from May-August than from December-January and more or less from October-September. We can also see that from October to December the maximum general level is lower than in other seasons of the year, thus meaning that in those months Barcelona has less visitors and tourists. As we said before, the GIF with all the images is disposed in the delivery.

# 12. Conclusions

After this project we got a lot of knowledge about different data types and preprocessing. To conclude with the project we want to capture the knowledge obtained from every part of this project starting from the preprocessing.

The preprocessing is probably the most important part when we work with real data. Because almost always, the data contains errors or wrong values. First of all we have to mention the dimensionality reduction, which we had to incorporate to our project due to the high dimensionality of the data (75 features). We learned that filter methods are a good method to do feature selection, and that PCA works well if it accumulates a lot of inertia.

In the outlier section we have used the multivariate approach of Mahalanobis distance; this method has enabled us to treat data points that were outliers and were not detected by the univariate approach due to the lack of information about the interaction between variables. After this step we had our preprocessing successfully completed.

The MCA analysis revealed that host response time, instant bookable option, gender, room type, and source are the most important factors influencing Airbnb listing prices. These variables are closely tied to host verification, response efficiency, and guest booking convenience. The impact of the apartment's district and host location on price was found to be relatively low. Factor analysis showed strong correlations among categories related to host verification and response time. Apartments

without recorded host response times were associated with unverified IDs. Categories such as mail verification, phone and work mail verification, and host response time within a few hours contributed significantly to the dataset's variance.

After all of this, we needed to do clustering and a profiling to our data in order to extract clear knowledge about it. First of all, we tested with several clustering algorithms such as DBSCAN and OPTICS (density-based) and Hierarchical Clustering and CURE (distance-based). After checking the results, the clustering method that fitted our data better was the distance-based, in particular Hierarchical Clustering. We think that this may be because the volume of data is not really big.

Once we have learned more advanced techniques to preprocess and to evaluate our categorical and numerical data, we started to learn about other types of data. The first one was the Time Series, in particular the clustering of several time series. To apply this to our data, we decided to compare the reviews per month in several cities around the world. With this we end up with very interesting results: Airbnb hasn’t fully arrived to Africa; Covid had a negative impact in all the time series; The countries with covid-zero policy didn’t recover the values before the pandemic.

In textual analysis, we need to highlight what we observed in the sentiment analysis. The values of the reviews that we calculated using the AFINN lexicon were slightly lower than those provided by Airbnb. However, both variables were correlated, which means there is a relationship between them. This aspect leads us to believe that the difference in values is caused by a few words with high negative values in some reviews, which lower the overall average and ultimately make the review values tend to be similar. Finally, almost all the reviews were positive.

In correspondence analysis and topic modeling, we have been able to extract valuable data about what is mentioned in the reviews. Specifically, in CA GALT, we have observed the most important landmarks of each neighborhood mentioned in the reviews. This information is extremely useful if you own an apartment and want to identify nearby points of interest to include in the description and attract more guests.

Additionally, through topic modeling, we have successfully generated 17 distinct and coherent topics. Using these topics, we conducted an analysis of ratings based on each topic. This approach has enabled us to uncover the words commonly used in apartments with low ratings and those with high ratings.

In geospatial and geostatistical modeling we have encountered a lot of challenges and exciting learning. We have learned how to predict the value of some continuous random point in space based on some empirical observations and a function of adjusted variogram. We have checked that our results were coherent with our propper reality, which is that the variable used to construct the study was not dependent on space. In geostatistical modeling we have seen how the intensity and density of the reviews are influenced by the season, being the Summer and the Spring the seasons where there are more demands in beach-near places. We also have checked that there is no significant change of the most reviewed points during the last ten years, from 2011 to 2023.