Data Mining for Airbnb in New York

Data Mining

Airbnb in New York

Mario Font Blanc

Ramon Ribas Domingo

Xavier Marti Llull

David Daniel Streuli

Ricard Guixaró Trancho

UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH

Index

Motivation and general description 3

Data Source presentation 3

Formal description of Data structure and metadata 4

Complete Data Mining process 7

Preprocessing and data preparation 7

- Deleting rows and columns of the table 7
- Redefining the type of the variables 7
- Applying the KNN method 8
- Recording missing data as a new modality and redefining levels 8
- Outliers 8

Basic statistical descriptive analysis 10

- Univariate for all the variables included in the study 10
- Bivariate 29

PCA analysis 33

Hierarchical Clustering 39

Profiling of clusters 41

- Profiling variables 42
  - Neighbourhood group 42
  - Latitude and longitude 43
  - Instant Bookable 44
  - Cancellation policy 45
  - Room type 46
  - Minimum nights 47
  - Number of reviews and reviews per month 48
  - Calculated listing counts 49
  - availability .365 49
  - Price and service fee 50
- Cluster description 53
- PCA and Clustering comparison 54

Conclusions 54

Working plan 56

R Scripts 58

Redefining types script 58
Recoding missings and imputing 1nn 58
PCA script 60
Clustering 63
Profiling 65

Motivation and general description

This is the first practical work for the data mining course at UPC Barcelona. In this work we were tasked with finding a dataset on which we have to perform a complete data mining process including a formal description of the data, preprocessing, a basic statistical descriptive analysis of the data, a PCA analysis and finally clustering the data and profiling it.

To find an appropriate dataset we scanned Kaggle for interesting datasets that meet the course requirements. This part turned out to be more difficult than expected, however we found a dataset for Airbnb in New York that met all the requirements and is also interesting to work with.

Using this dataset our goal was then to analyse, structure and visualise this data. The motivation behind this first practical work is to get an understanding of the complete data mining process and overview of the different algorithms that are used in data mining. Furthermore we should learn to work with RStudio and develop a feel for good practices in writing code for data mining projects.

Data Source presentation

The dataset we are going to use for this project was extracted from *kaggle*. It is a specific dataset meant to apply a data cleaning process on it, being a perfect fit for the requirements of our work. The set also covers the variable types that we needed. To get the data we simply downloaded the CSV file from *kaggle.* Lastly, having our set on RStudio, we deleted some of the rows to make a more manageable set with 5000 rows.

The data is information from Airbnb stays in New York city. Every row is a different Airbnb entry where there is information about the place, such as the name, the location, its unique id...

Data source including the url involved:

https://www.kaggle.com/datasets/arianazmoudeh/airbnbopendata

Number of records	5000

Number of variables	21
Number of numerical variables	13
Number of binary variables	2
Number of date variables	1
Number of qualitative variables	5

Formal description of Data structure and metadata

In this section we give a formal description of our data including a metadata table. Our dataset has 5000 rows and 21 columns (variables). The rows have been reduced to 5000 in order to meet the course requirements and also for efficiency reasons. Thirteen of the variables are numerical, five are qualitative, two are binary and one is a date variable. This metadata file specifies how many missings each variable of our initial database has (analysed with R). If we sum all the missings we see that only a **1.62%** of the whole data matrix is missing.

Number and % of missing data per each variable: (id: 0 -> 0%), (host id: 0 -> 0%), (host_identity_verified: 73 -> 1.46%), (host name: 20 -> 0.4%), (neighborhood group: 27 -> 0.54%), (neighborhood: 16 -> 0.32%), (lat: 8 -> 0.16%), (long: 8 -> 0.16%), (instant_bookable: 79 -> 1.58%), (cancellation_policy: 50 -> 1%), (room type: 0 -> 0%), (construction year: 135 -> 2.7%), (price: 0 -> 0%), (service fee: 0 -> 0%), (minimum nights: 84 -> 1.68%), (number of reviews: 8 -> 0.16%), (last review: 475 -> 9.5%), (reviews_per_month: 461 -> 9.22%), (review_rate_number: 93 -> 1.86%), (calculated_host_listings_count: 23 -> 0.46%), (availability.365: 148 -> 2.96%).
% of missing data in the whole data matrix: 1.62%

Variable	Modalities	meaning	Type	Measuring unit	Missing code	Measuring procedure	Range	Role
id		Advert's	Numerical		There's no			

		identification			missing code			
host id		Identification of the advert's owner	Numerical		There's no missing code			
host_identity_verified		Wether the account is assigned to a verified person or not	Boolean		"" (73)			Explanatory
	U	Unconfirmed						
	V	Verified						
host name		Name of the host	Qualitative		"" (20)			Explanatory
neighbourhood group		District name	Qualitative		"" (27)			Explanatory
neighbourhood		Neighbourhood name	Qualitative		"" (16)			Explanatory
lat		Latitude	Numerical	Degrees	NA (8)	Airbnb takes the coordinates of the house location	[-90,90]	Explanatory
long		Longitude	Numerical	Degrees	NA (8)	Airbnb takes the coordinates of the house location	[-180,180]	Explanatory
instant_bookable		Wether the household is instant bookable or not	Boolean		NA (79)			Explanatory
	T	True						
	F	False						
cancellation		Policy of	Qualitative		"" (50)			Explanator

_policy		cancellation						y
	S	Strict						
	M	Moderate						
	F	Flexible						
room type		Type of space to be booked	Qualitative		There's no missing code			Explanatory
Construction year		Year of construction	Numerical	Years	NA (135)			Explanatory
price		Renting price	Numerical		There's no missing code			Explanatory
service fee		Airbnb's service fee	Numerical	Dollars	There's no missing code			Explanatory
minimum nights		Minimum of nights to stay	Numerical		NA (84)			Explanatory
number of reviews		Number of advert's reviews	Numerical		NA (8)			Explanatory
last review		Last review date	Date		NA (475)			Explanatory
reviews per month		Number of reviews per month	Numerical		NA (461)			Explanatory
review rate number		Review rate score	Numerical		NA (93)	Airbnb takes the guest's ratings that go from 0 to 5 and makes an average	[1,5]	Explanatory
calculated host listings count		Total number of listings made by hosts	Numerical		NA (23)			Explanatory
availability		Available	Numerical		NA (148)	Airbnb asks	[0,365]	Explanator

365		days during the year				it to the owner		y

Complete Data Mining process

In this section we present the full data mining process.

Preprocessing and data preparation

In this section we describe the steps included in the preprocessing of our data in order to prepare it for further analysis and treatment.

Deleting rows and columns of the table

First of all, we have deleted many rows of our initial dataset to make it not that huge, now only 5000 of them are left. In addition, we have deleted that columns represented by variables that did not have relevance and were not important for our analysis. Those were name (name of the listing), country and country.code (which were redundant because we are dealing with adverts only from New York), house_rules (which only contained opinions) and license (which was empty). In conclusion, our database has 5000 rows and 21 columns.

Redefining the type of the variables

After remodelling the data matrix, we have redefined the type of those variables that R has interpreted with a type that we had not expected using as.'type'('variable'). Concretely, we have redefined variables service.fee and price to numerical (they were qualitative due to the dollar symbol) and last.review from qualitative to date, giving it the correct format.

```
price       <- as.numeric(price)
service.fee <- as.numeric(service.fee)
last.review <- as.Date(last.review, format ="%m/%d/%Y")
```

Applying the KNN method

Once the type of the variables has been redefined, we have applied 1-nn to the missing values from numerical variables with the aim to exterminate any missing from those variables with the aim to improve the quality of our posterior analysis.

```
#built indexes of numerical variables that require inputation

uncompletevars<-c(7,8,16,20,15,19,12,21,18)

#better if you sort them by increasing number of missing values

fullvariables<-c(1,2,13,14)
aux<-dd[,fullvariables]
dim(aux)
names(aux)

for (k in uncompletevars){
  aux1 <- aux[!is.na(dd[,k]),]
  dim(aux1)
  aux2 <- aux[is.na(dd[,k]),]
  dim(aux2)

  RefValues<- dd[!is.na(dd[,k]),k]
  #Find nns for aux2
  knn.values = knn(aux1,aux2,RefValues)

  #CARE: neither aux1 nor aux2 can contain NAs

  #CARE: knn.ing is generated as a factor.
  #Be sure to retrieve the correct values

  dd[is.na(dd[,k]),k] = as.numeric(as.character(knn.values))
  fullvariables<-c(fullvariables, k)
  aux<-dd[,fullvariables]
}
```

Recording missing data as a new modality and redefining levels

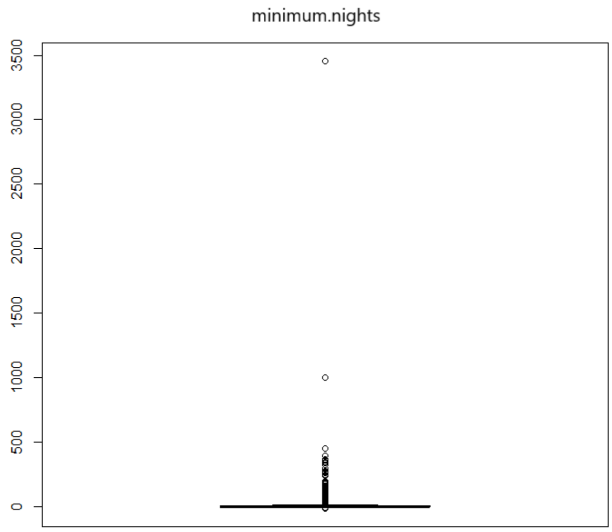
Since our qualitative variables' missings are random and we can not estimate their values, we have kept them as a new modality with an "unknown" tag and we have redefined the levels of this qualitative variables adding that tag as a new level.

```
host.name[host.name==""]<-"Unk_hName"
levels(host.name)<-c(levels(host.name),"Unk_hName")
neighbourhood.group[neighbourhood.group==""]<-"Unk_neighG"
levels(neighbourhood.group)<-c(levels(neighbourhood.group),"Unk_neighG")
neighbourhood[neighbourhood==""]<-"Unk_neigh"
levels(neighbourhood)<-c(levels(neighbourhood),"Unk_neigh")
cancellation_policy[cancellation_policy==""]<-"Unk_cPolicy"
levels(cancellation_policy)<-c(levels(cancellation_policy),"Unk_cPolicy")
```

Outliers

To identify the outliers we mainly used the *summary* function that R has. That function is very useful to get the minimum, the maximum and the median of a column. With this information we were able to see if any of the values of our variables were out of the expected range. For example, our variable *availability.365* that represents the number of days that an Airbnb is available during the year. This variable had negative values, so we had to change them to unknown ones.

We also used the *boxplot* function in R to identify outliers that were not totally wrong. That was very helpful for our *minimum.nights* variable that had extreme values that were not correct for an airbnb.



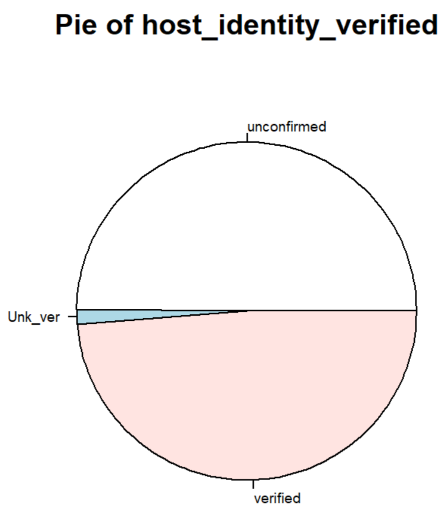
Basic statistical descriptive analysis

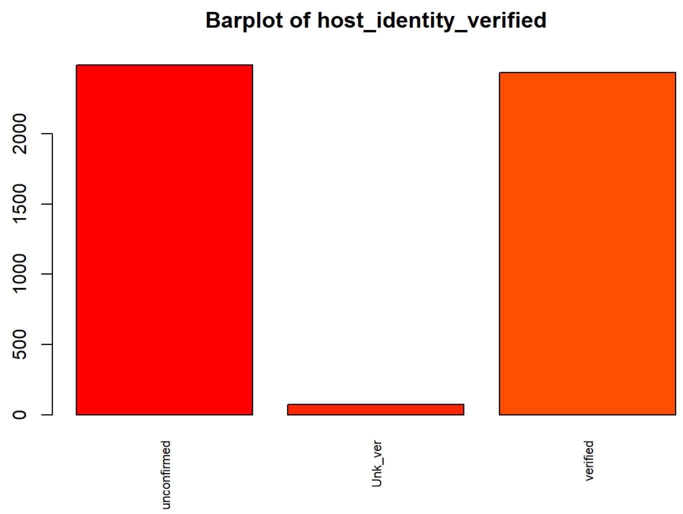
In this section we provide a univariate and bivariate analysis for selected variables of our dataset.

Univariate for all the variables included in the study

Name of the variable: **host_identity_verified**

Number of modalities: 3





Figures 2.1 & 2.2: Pie-chart and bar plot of ***host_identity_verified***

Modalities	Frequency	Proportion
unconfirmed	2491	0.4982
verified	2436	0.4872
Unk_ver	73	0.0146

Table 1: Frequency and relative frequency of ***host_identity_verified***

The table and plots show a minimal difference between unconfirmed and verified data.

Name of the variable: **host.name**

Number of modalities: 2091

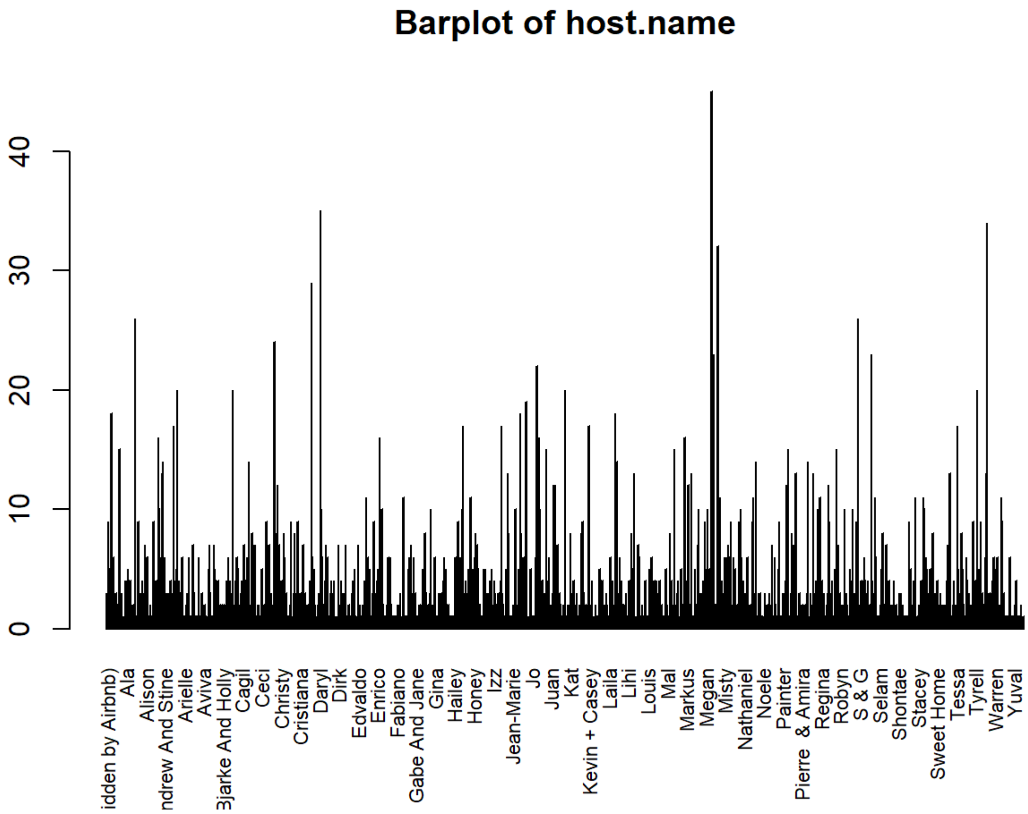


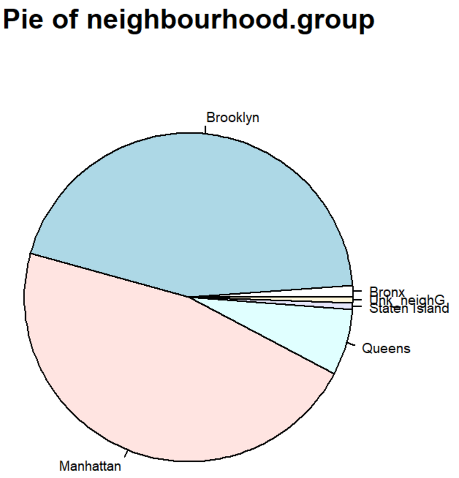
Figure 2.3: Bar plot of ***host.name***

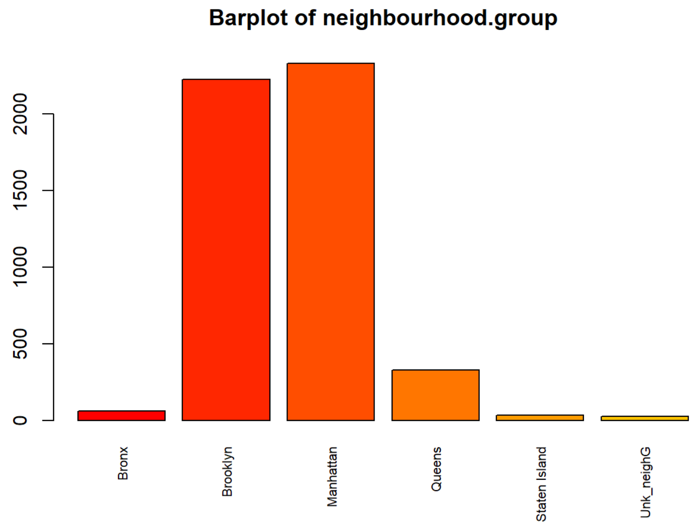
Modalities	**Frequency**	**Proportion**
Michael	45	0.0090
David	35	0.0070
Vida	34	0.0068
Mike	31	0.0062
Daniel	29	0.0058
Alex	26	0.0052
Ryan	26	0.0052
...		

Table 2: Frequency and relative frequency of ***host.name***

Name of the variable: **neighbourhood.group**

Number of modalities: 6





Figures 2.4 & 2.5: Pie-chart and bar plot of ***neighbourhood.group***

Modalities	Frequency	Proportion
Manhattan	2330	0.4660
Brooklyn	2224	0.4448
Queens	328	0.0656
Bronx	59	0.0118
Staten Island	32	0.0064
Unk_neighG	27	0.0056
...		

Table 3: Frequency and relative frequency of ***neighbourhood.group***

As we can see, the majority of the Airbnb's are located either in Brooklyn or Manhattan.

Name of the variable: **neighbourhood**

Number of modalities: 151

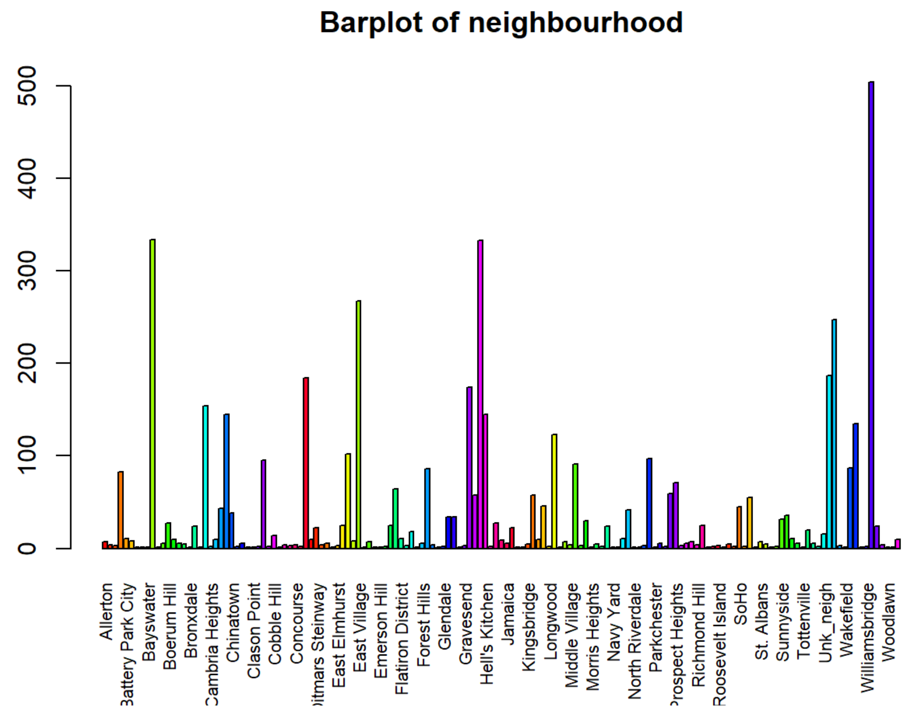


Figure 2.6: Bar plot of ***neighbourhood***

Modalities	**Frequency**	**Proportion**
Williamsburg	504	0.1006
Bedford-Stuyvesant	334	0.0668
Harlem	333	0.0666
East Village	267	0.0532
Upper West Side	247	0.0488
Upper East Side	187	0.0374
Crown Heights	184	0.0368
Greenpoint	174	0.0348
...		

Table 4: Frequency and relative frequency of ***neighbourhood***

As we can see, most of the Airbnb are located either in Brooklyn or Manhattan.

Name of the variable: **lat**

Minimum value: 40.15

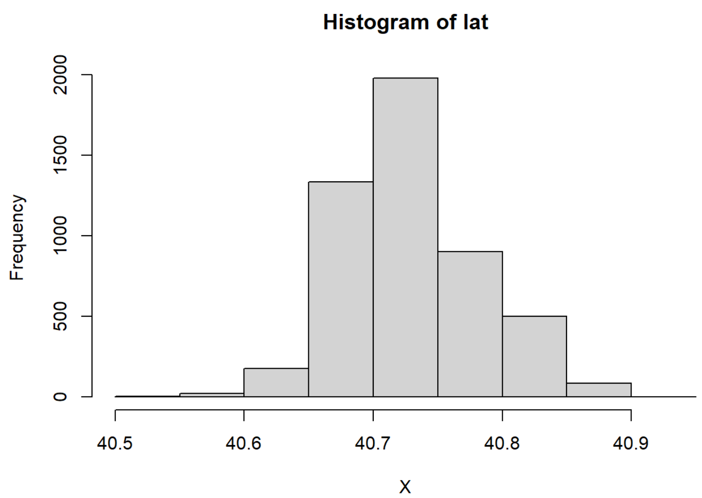
Maximum value: 40.91

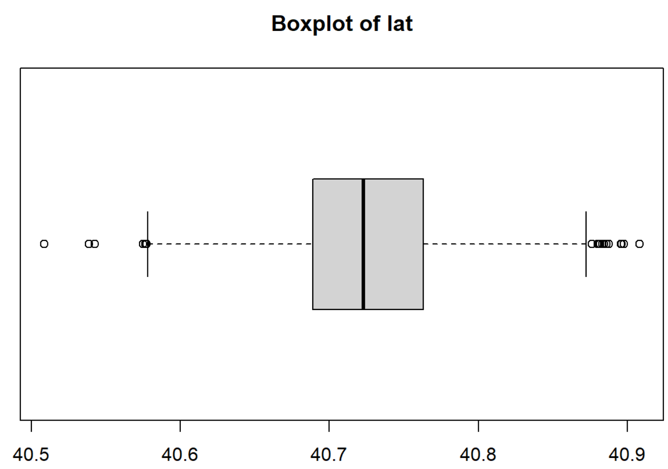
Mean: 40.73

Median: 40.72

Variance: 0.0513

Standard deviation: 0.0012

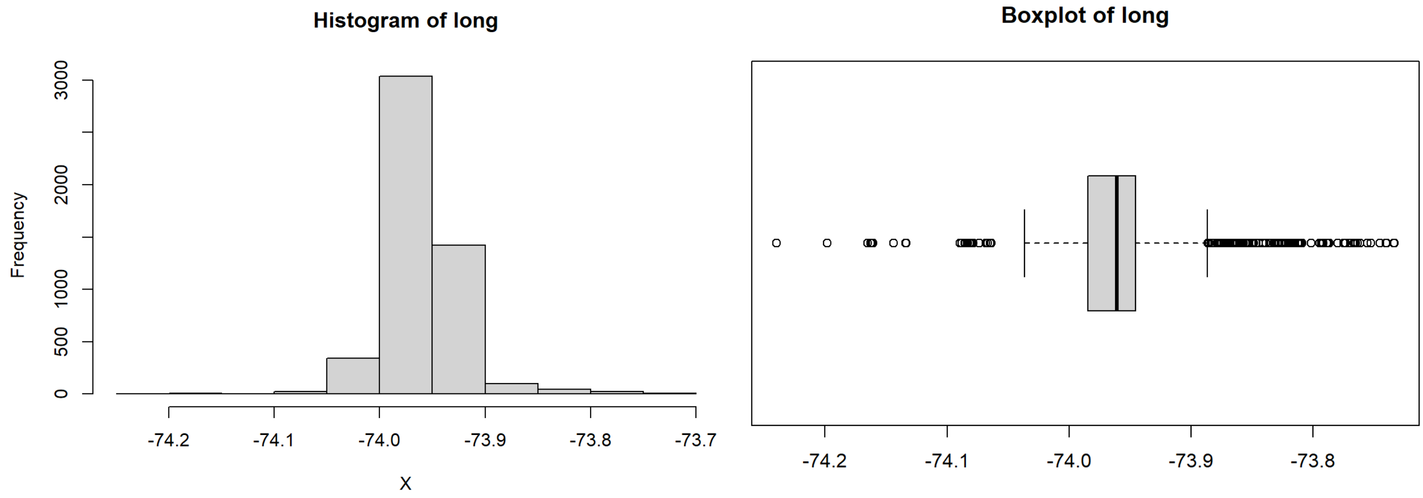




Figures 2.7 & 2.8: Histogram and box plot of ***lat***

On a map, the average latitude (*40.73*) would be between Manhattan and Brooklyn, which is logical given the results obtained from the variable *neighbourhood.group*.

Name of the variable:	**long**
Minimum value:	-74.24
Maximum value:	-73.73
Mean:	-73.96
Median:	-73.96
Variance:	-0.0004
Standard deviation:	0.0352

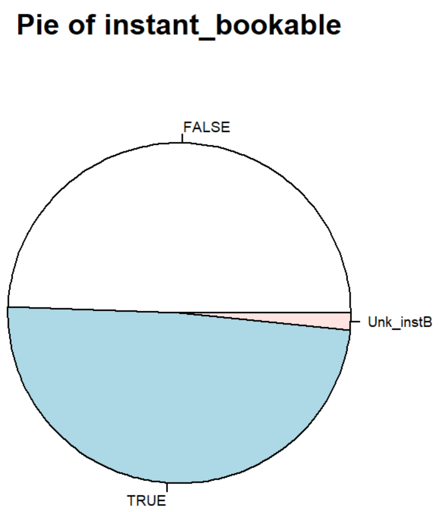


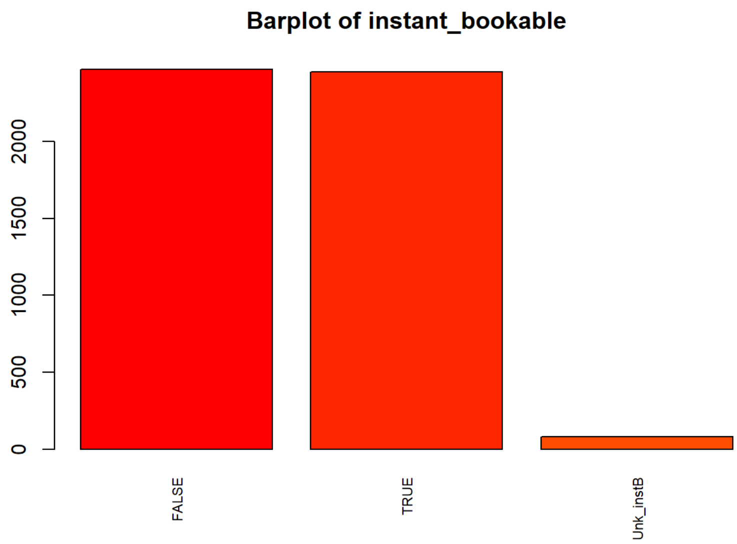
Figures 2.9 & 2.10: Histogram and box plot of ***long***

If we were to draw a line in the average latitude, it would cross the East River, which separates Manhattan from Brooklyn.

Name of the variable: **instant_bookable**

Number of modalities: 3





Figures 2.11 & 2.12: Pie-chart and bar plot of ***instant_ bookable***

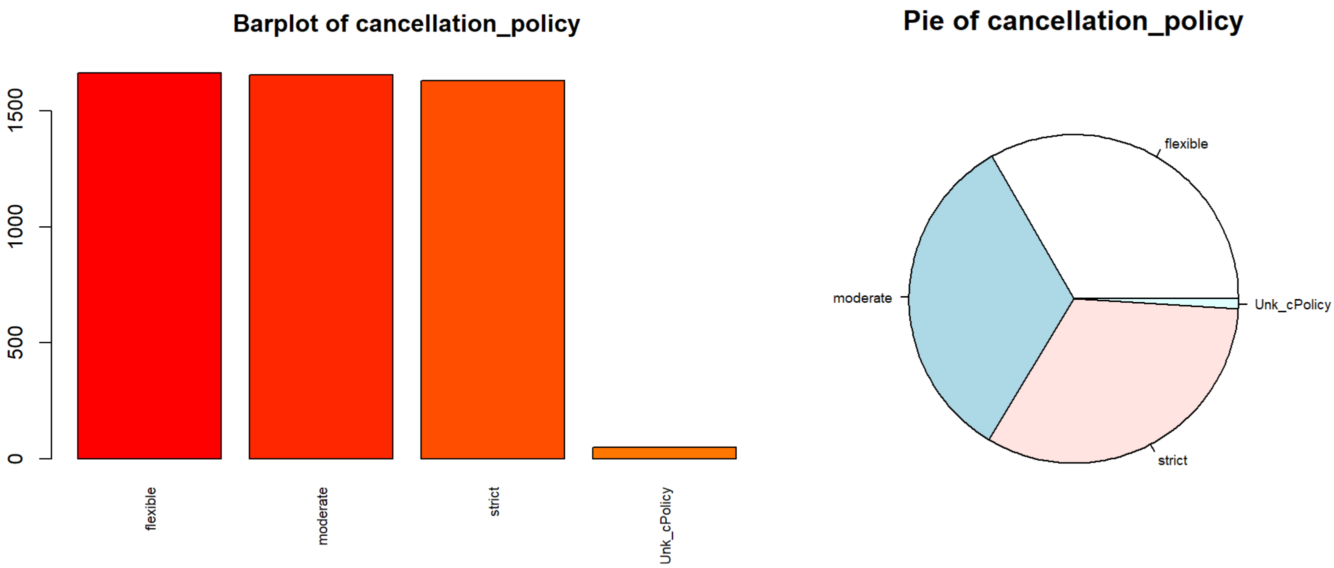
Modalities	**Frequency**	**Proportion**
FALSE	2469	0.4938
TRUE	2452	0.4904
Unk_instB	79	0.0158

Table 5: Frequency and relative frequency of ***instant_ bookable***

It is clear that these two values are almost perfectly balanced.

Name of the variable: **cancellation_policy**

Number of modalities: 4



Figures 2.13 & 2.14: Pie-chart and bar plot of ***cancellation_policy***

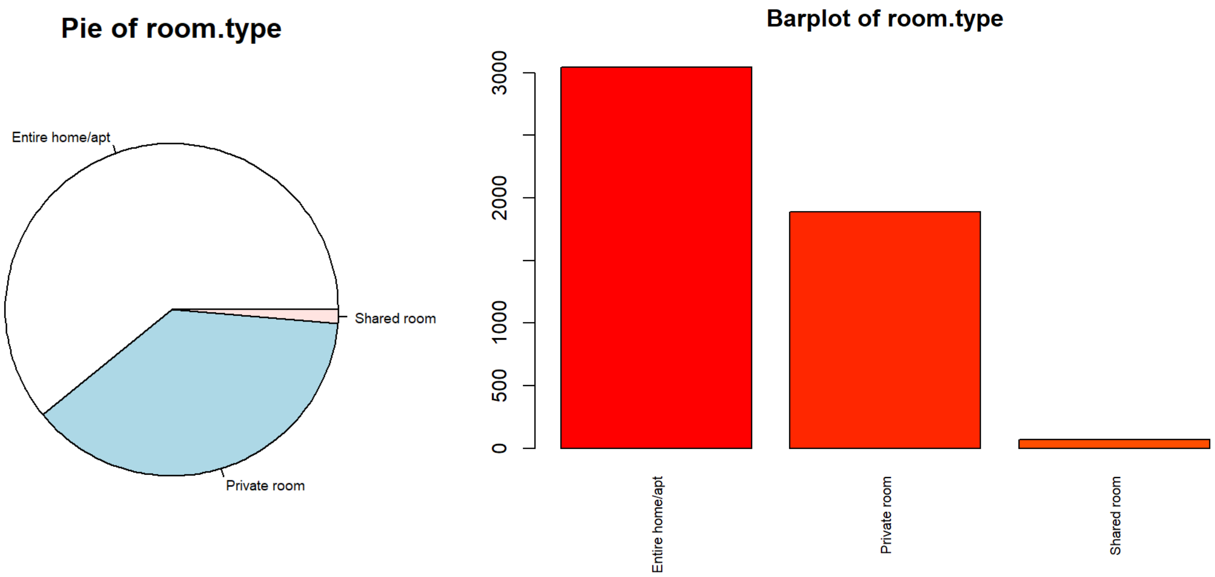
Modalities	**Frequency**	**Proportion**
flexible	1664	0.3328
moderate	1655	0.3310
strict	1631	0.3262
Unk_cPolicy	50	0.0100

Table 6: Frequency and relative frequency of ***cancellation_policy***

There is no apparent dominant reason why bookings are cancelled.

Name of the variable: **room_type**

Number of modalities: 3



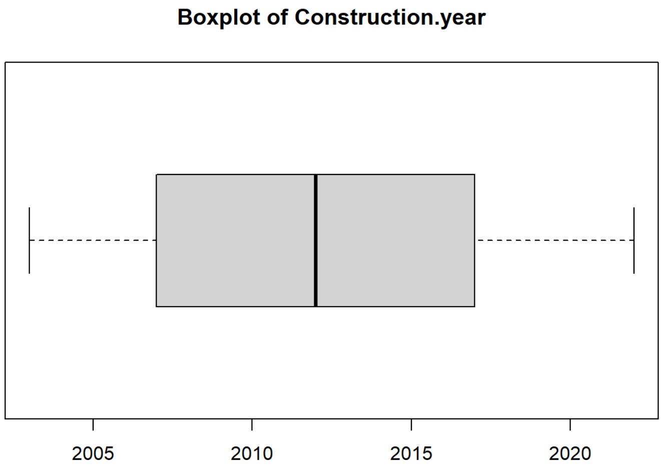
Figures 2.15 & 2.16: Pie-chart and bar plot of **room_type**

Modalities	**Frequency**	**Proportion**
Entire home/apt	3045	0.6090
Private room	1887	0.3774
Shared room	68	0.0136

Table 7: Frequency and relative frequency of **room_type**

The vast majority of the booked Airbnb's include an entire home or apartment.

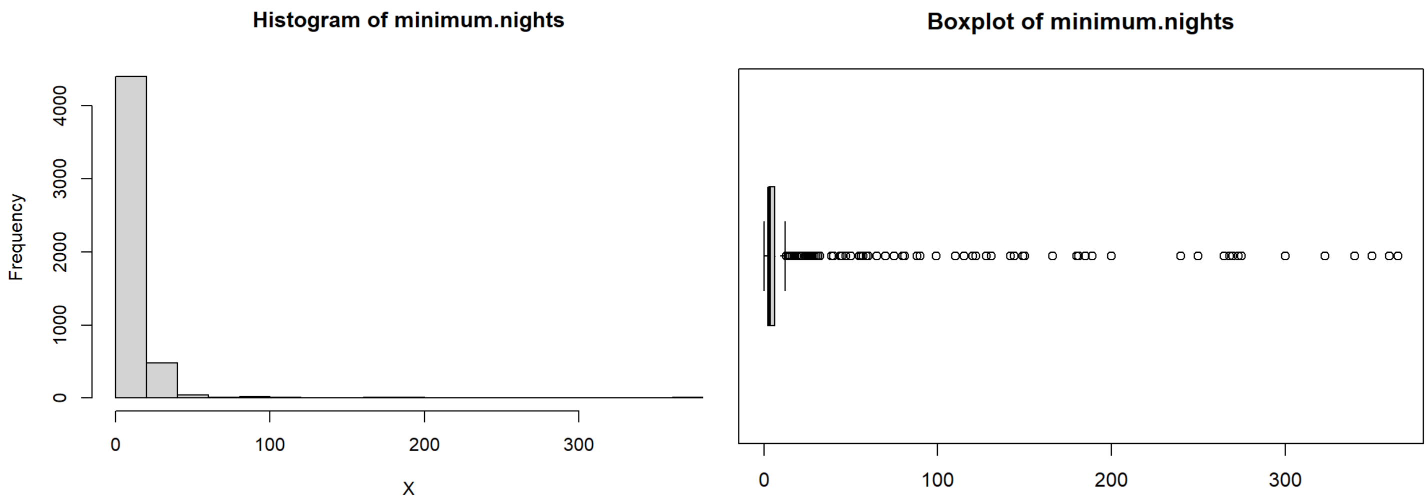
Name of the variable:	**construction.year**
Minimum value:	2003
Maximum value:	2022
Mean:	2012
Median:	2012
Variance:	0.0028
Standard deviation:	5.7818



Figures 2.17 & 2.18: Histogram and box plot of ***construction.year***

The average (2012) shows that it was during the first years of the 2010 decade that more apartments were built.

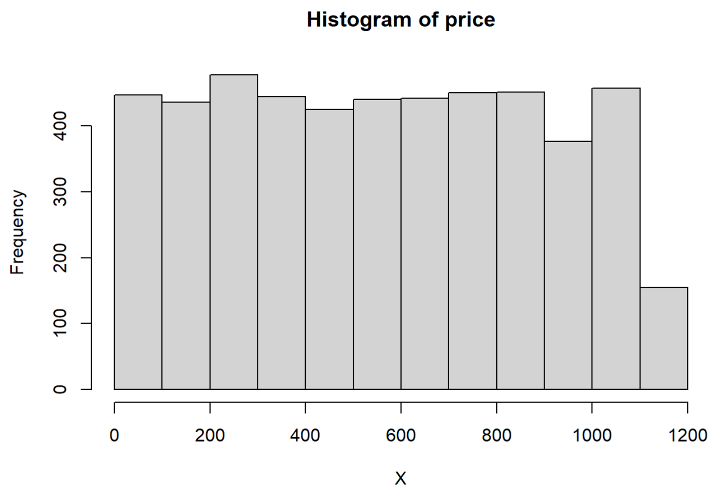
Name of the variable:	**minimum.nights**
Minimum value:	0
Maximum value:	365
Mean:	9.389
Median:	3
Variance:	2.9234
Standard deviation:	27.448

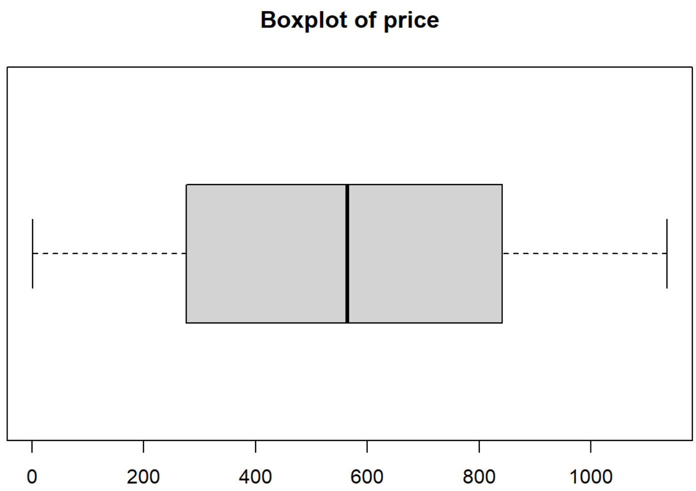


Figures 2.19 & 2.20: Histogram and box plot of ***minimum.nights***

The average minimum nights required to book an Airbnb apartment in New York is 3.

Name of the variable:	**price**
Minimum value:	$1
Maximum value:	$1136
Mean:	$563.1
Median:	$564.5
Variance:	0.5822
Standard deviation:	327.84

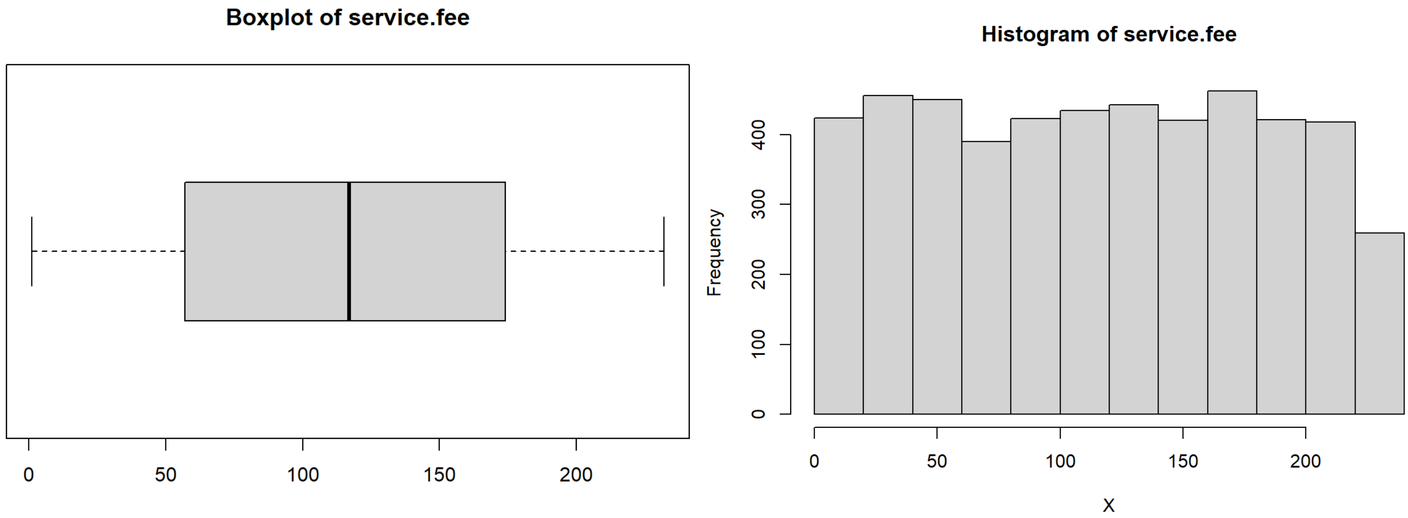




Figures 2.21 & 2.22: Histogram and box plot of ***price***

Based on the average price ($550) and the minimum amount of nights required to book an apartment (9), the cost per night is approximately $60.

Name of the variable:	**service.fee**
Minimum value:	$1
Maximum value:	$232
Mean:	$116.3
Median:	$117.0
Variance:	0.5751
Standard deviation:	66.873



Figures 2.23 & 2.24: Histogram and box plot of ***service.fee***

In addition to the cost of the night, to book an Airbnb apartment in New York, users are also charged an average of $116 based on the service fee.

Name of the variable: **number.of.reviews**

Minimum value: 0

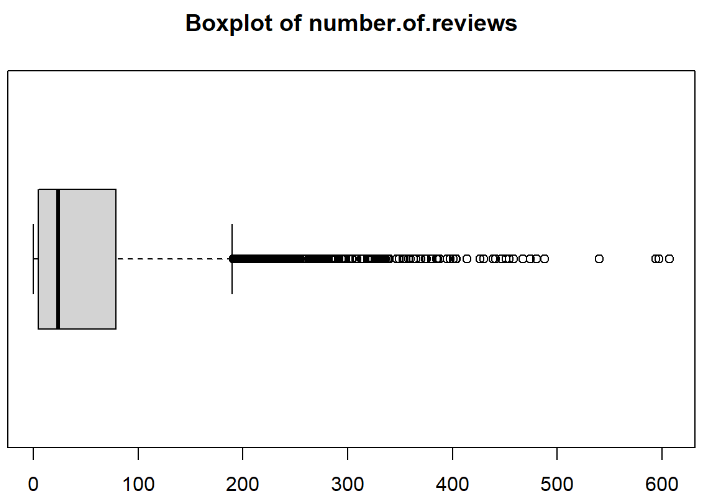
Maximum value: 607

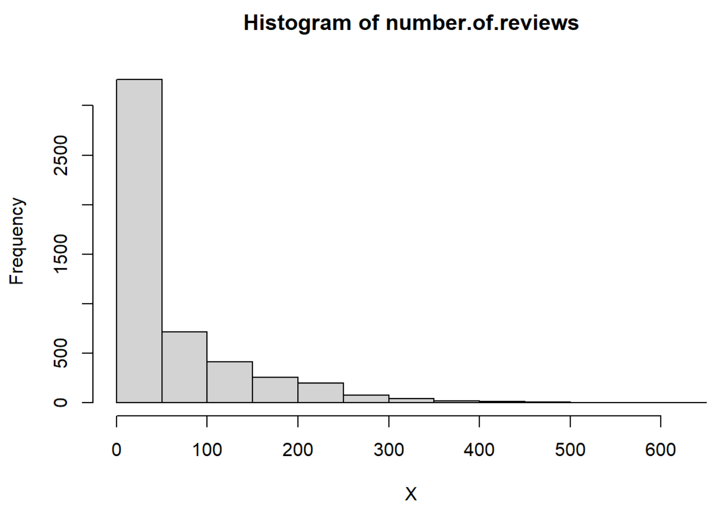
Mean: 57.24

Median: 24

Variance: 1.3464

Standard deviation: 77.064





Figures 2.25 & 2.26: Histogram and box plot of ***number.of.reviews***

Despite an analysis of 5,000 cases, only 30% of these bookings have received more than average reviews.

Name of the variable: **last.review**

Number of modalities: 1174

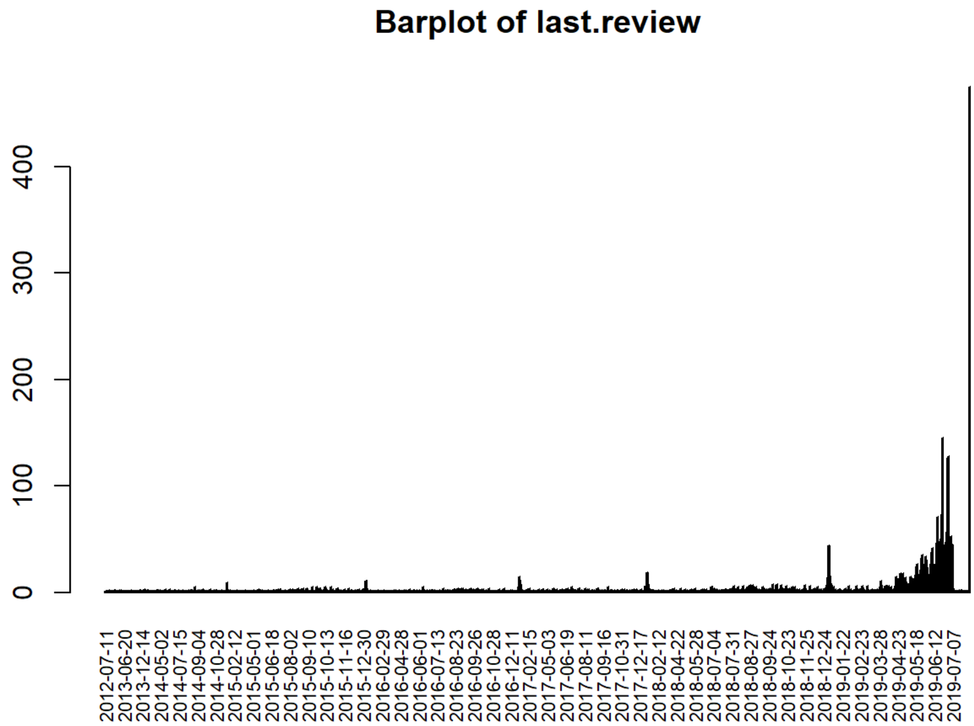


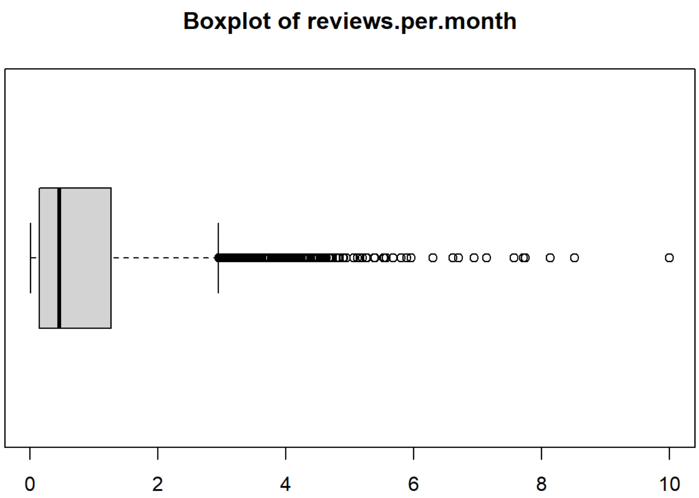
Figure 2.27: Bar plot of ***last.review***

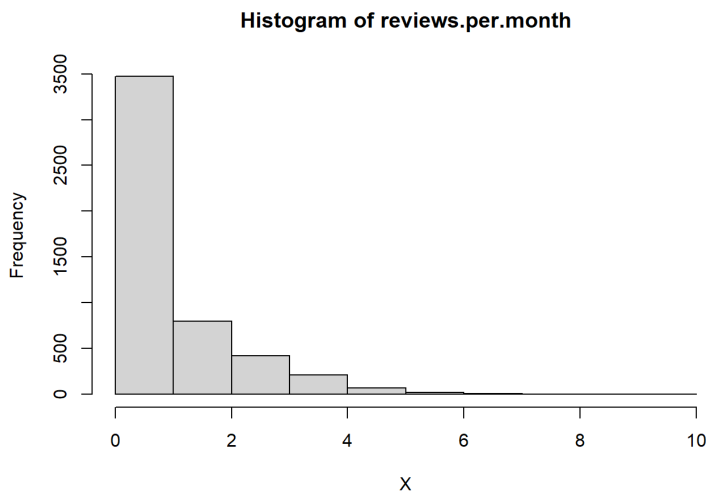
Modalities	Frequency	Proportion
Unk_lReview	474	0.0948
6/23/2019	145	0.0290
7/1/2019	128	0.0256
6/30/2019	126	0.0252
6/24/2019	100	0.0200
6/22/2019	73	0.0146
...		

Table 6: Frequency and relative frequency of ***last.review***

The number of reviews submitted in 2019 is far superior to any other year in the dataset.

Name of the variable:	**reviews.per.month**
Minimum value:	0.0100
Maximum value:	10
Mean:	0.9057
Median:	0.46
Variance:	1.2052
Standard deviation:	1.0915

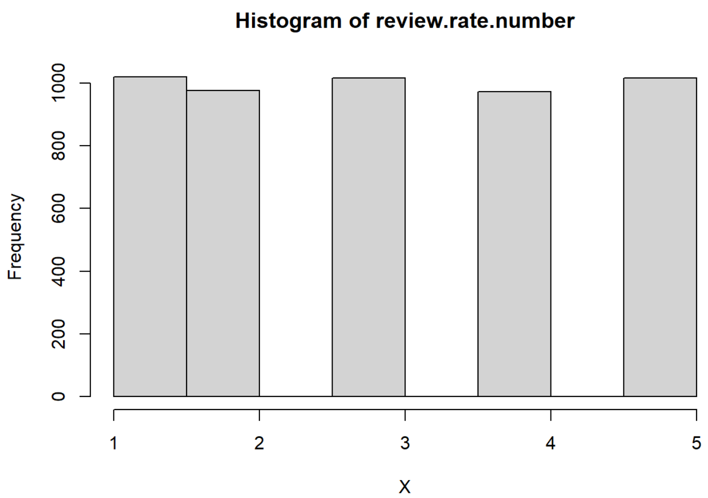


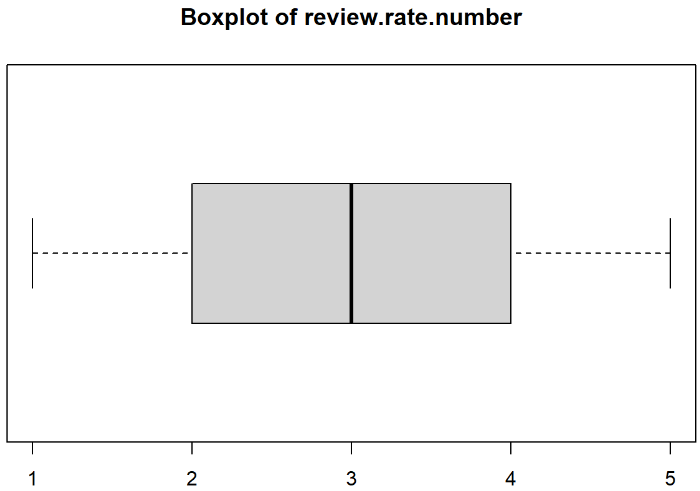


Figures 2.27 & 2.28: Histogram and box plot of ***reviews.per.month***

The average number of monthly reviews on an apartment is slightly less than 1.

Name of the variable:	**review.rate.number**
Minimum value:	1
Maximum value:	5
Mean:	2.998
Median:	3
Variance:	0.4738
Standard deviation:	1.4205





Figures 2.29 & 2.30: Histogram and box plot of ***review.rate.number***

More than 800 apartments have received a review with the minimum score possible (1).

Name of the variable: **calculated.host.listings.count**

Minimum value: 1

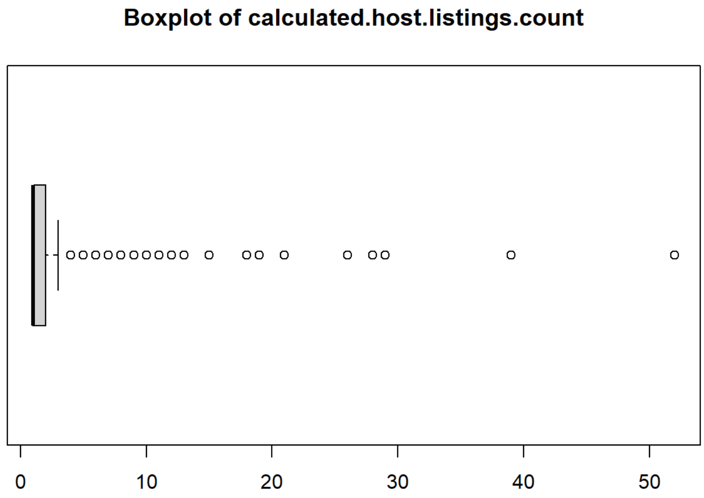
Maximum value: 52

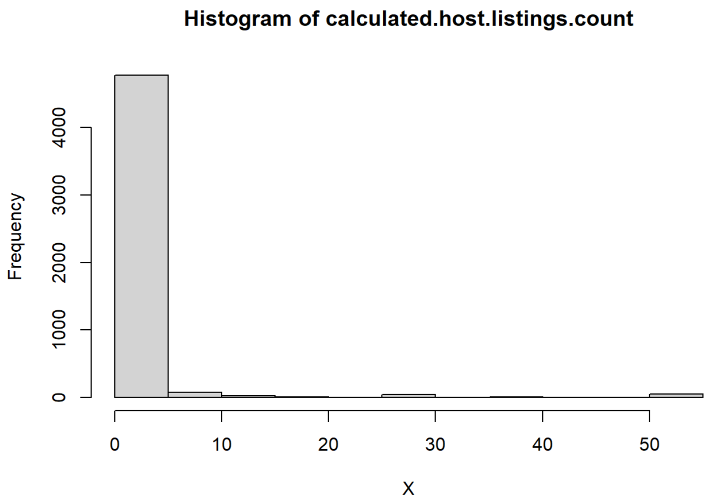
Mean: 2.458

Median: 1

Variance: 2.4967

Standard deviation: 6.1360

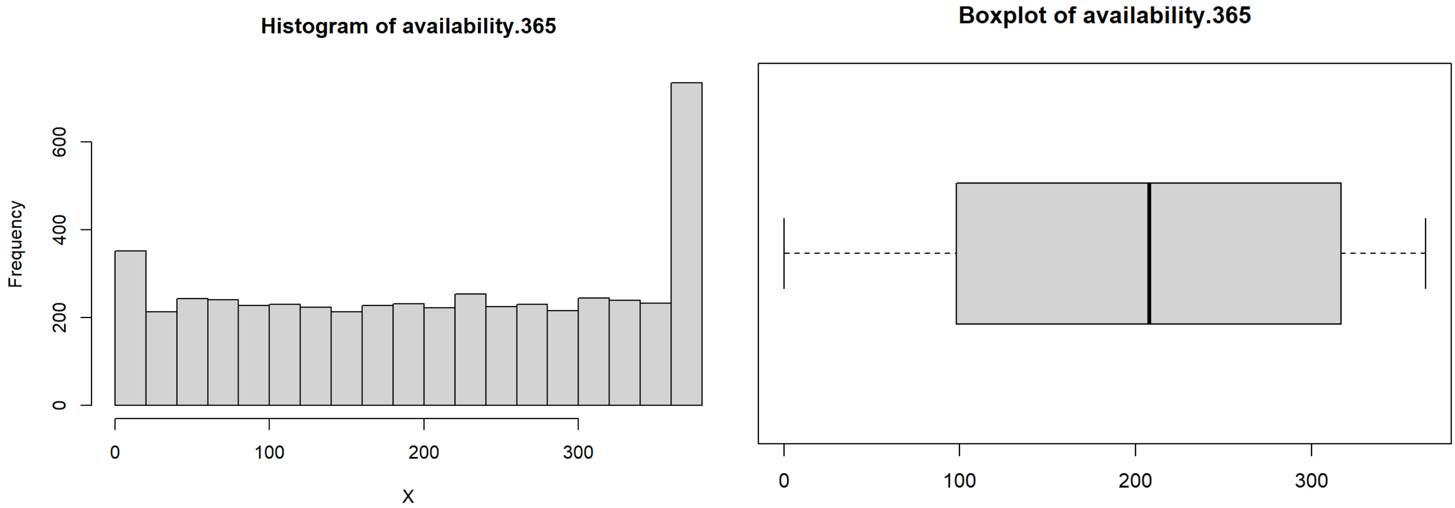




Figures 2.31 & 2.32: Histogram and box plot of ***calculated.host.listings.count***

After analysing the computed.host.listings.count variable, we can state that each host makes between 2 and 3 listings, with a few exceptions, although they are likely outliers.

Name of the variable:	**availability.365**
Minimum value:	0
Maximum value:	365
Mean:	203.9
Median:	208
Variance:	0.5858
Standard deviation:	119.42



Figures 2.33 & 2.34: Histogram and box plot of ***availability.365***

During the year, most of the apartments are available for about 200 days, and if we look at the average of the *minimum nights* required to make a reservation (9), each host completes around 22 contracts per year.

Bivariate

For bivariate analysis, we have decided to compare longitude vs latitude to have a first view about the apartments' locations in New York, and we obtain a curious set of points that represent more or less the map of New York.

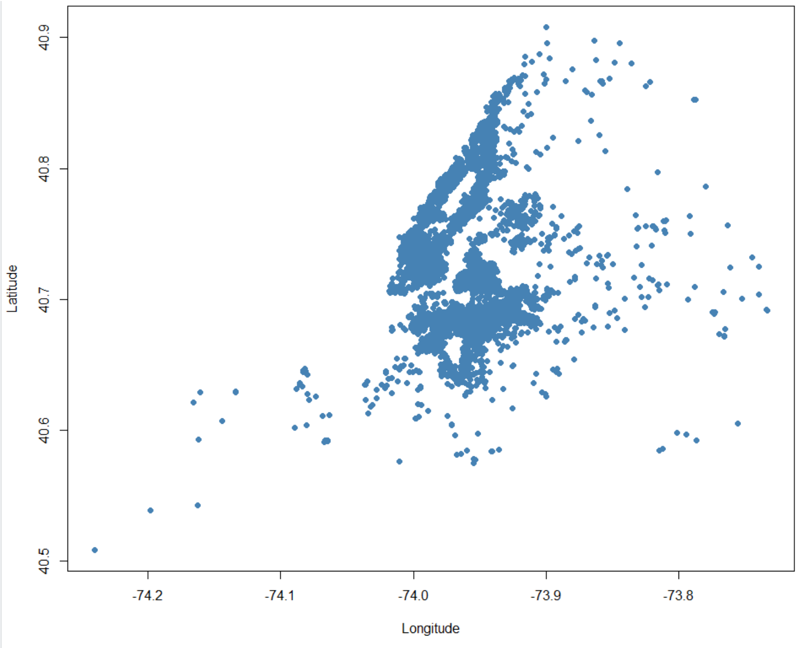


Figure 2.35: Histogram of ***latitude vs longitude***

If we translate it to a real map, we see that the part that has more apartments is the one below.

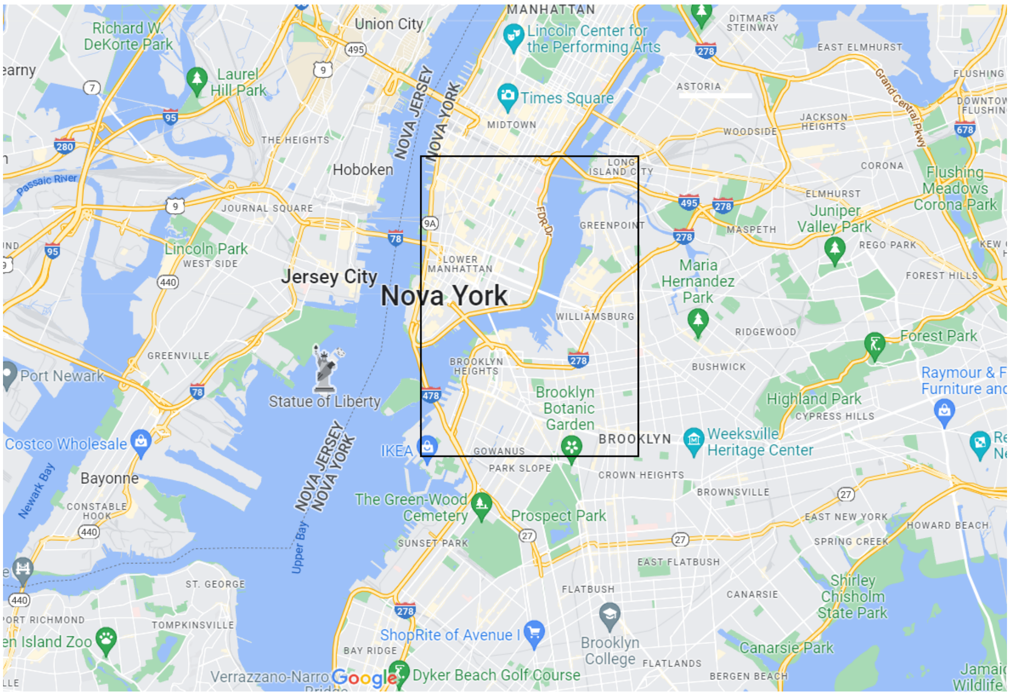


Figure 2.36: zone of most abundance of apartments

We then attempted to find a significant relationship between the rental price and the type of room. As the scatter plot below shows, those apartments that are meant to be shared, generally have a lower price. There is little difference between the private room and the ones that include the entire apartment.

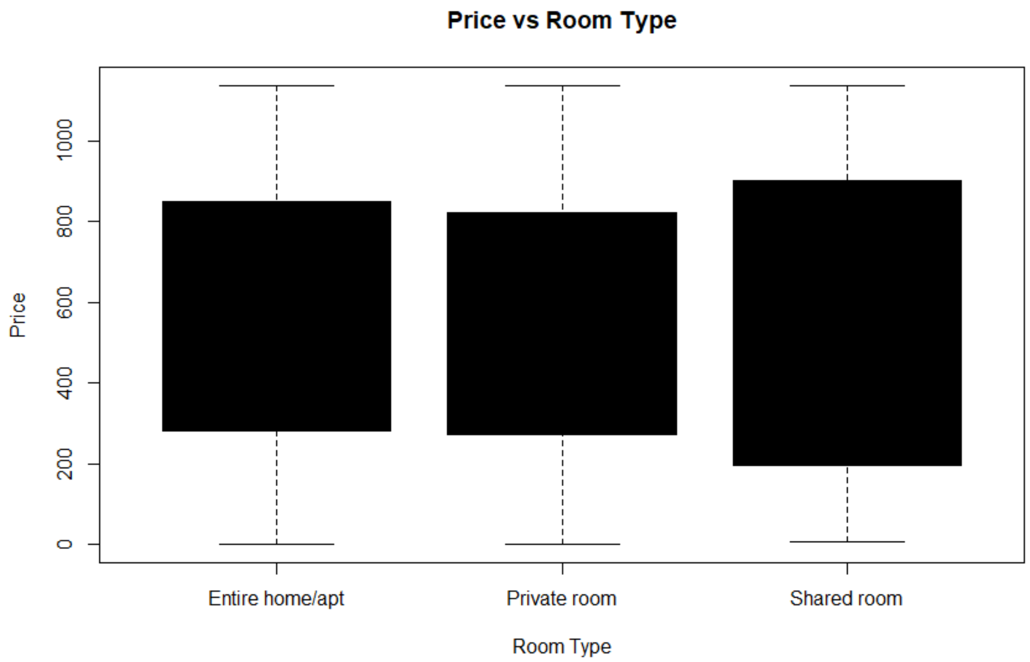
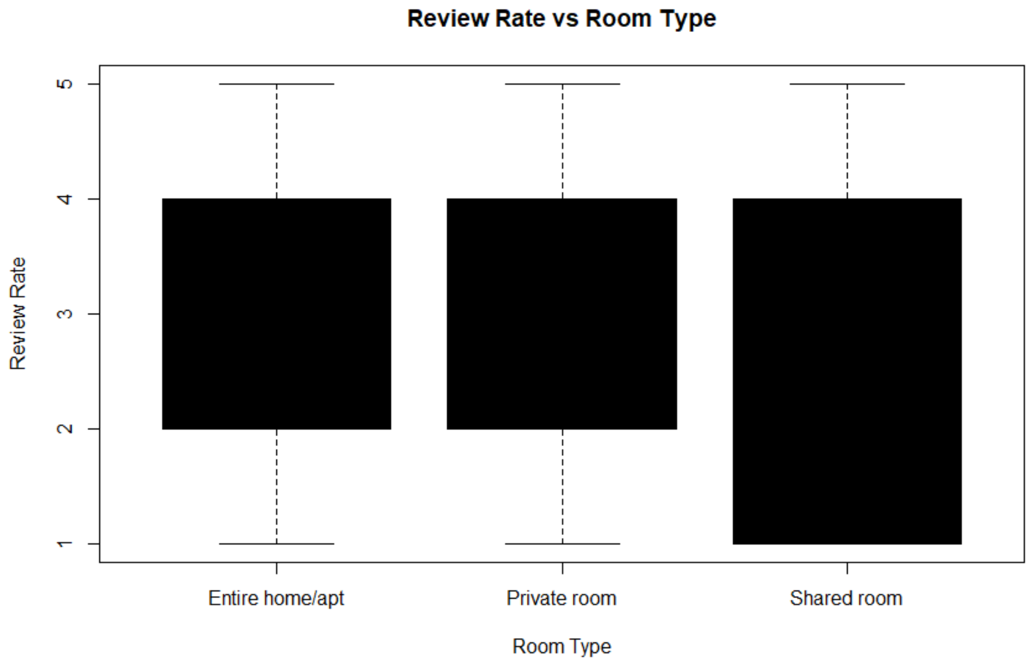


Figure 2.37: Boxplot of price vs Room Type

We also looked at a correlation between an apartment's review rate and its type. In this case, it is quite clear that the shared rooms get much lower rates than other apartments.



Figures 2.38: Boxplot of review.rate vs room.type

Next, we tried to connect the price of the apartment and its locations, defined by the neighbourhood group. In this case, it is only in Staten Island that we find a "significant" difference, the price is usually lower than in other parts of the city.

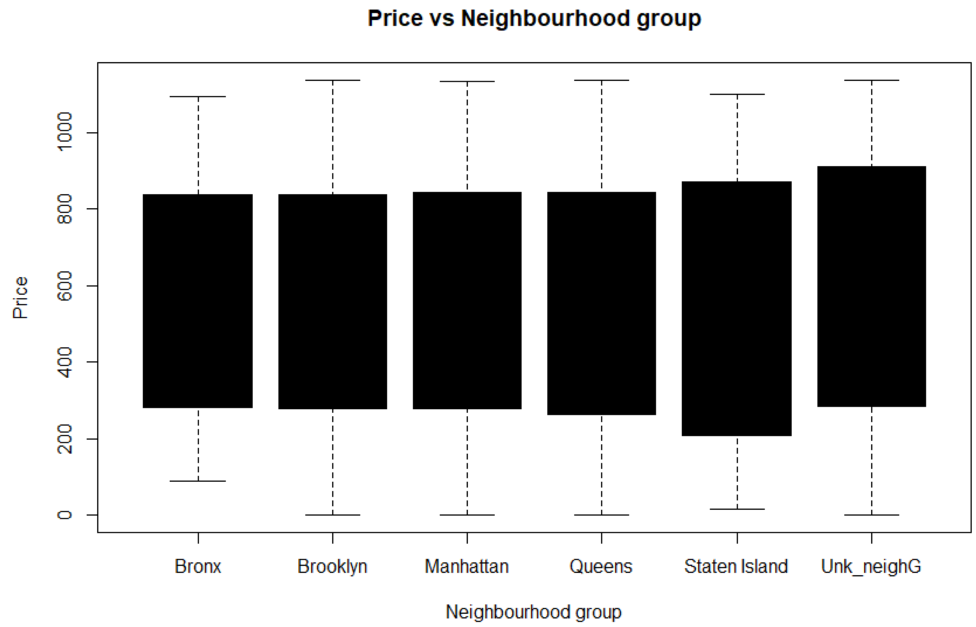
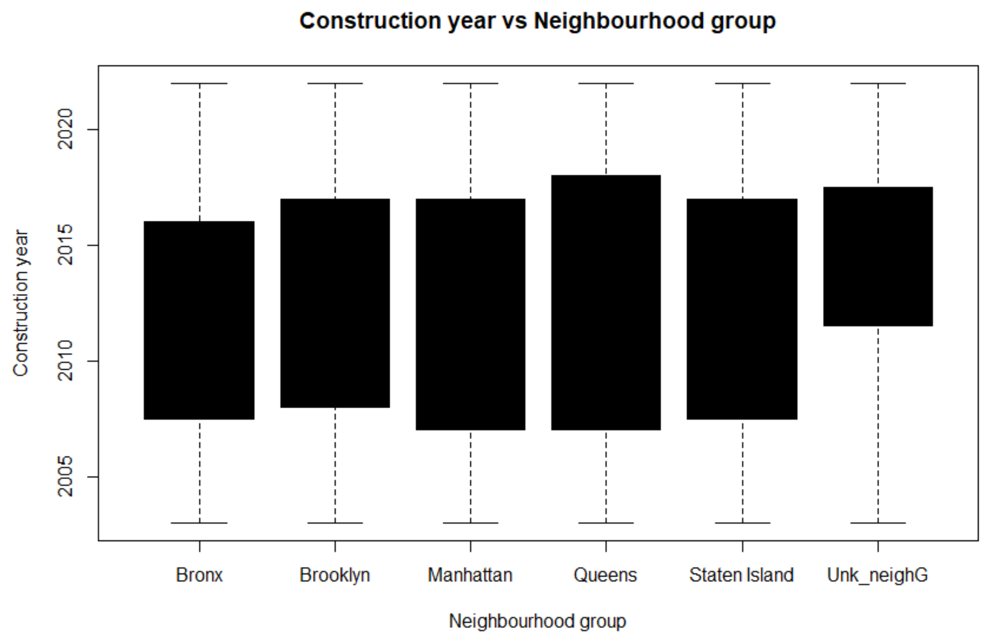


Figure 2.39: Box plot of *neighbourhood.group*

To finish with the bivariate analysis, we once again analysed the neighbourhood group, but this time to find a relationship with its year of construction. In the chart obtained, we can see more differences than in the previous one. The Bronx, Manhattan and Queens built more flats in the closing years of the 2000s. However, it is also in Queens, where more and more apartments are being built today.



Figures 2.40: Boxplot of ***Construction.year vs neighbourhood.group***

Conclusions

In conclusion, our initial database had very interesting variables, but also some that would not contribute to our analysis, so we deleted them. Despite this, we still have 21 variables, where 13 are numerical, 5 are qualitative, 2 are binary and one is a date variable. In fact, it is probable that we do not use some variables such as id, host.id and host.name, but by the moment we will keep them to have an identification for every row of the table. Now we also have our numerical values without missing values and without outliers, what will make our analysis be better.

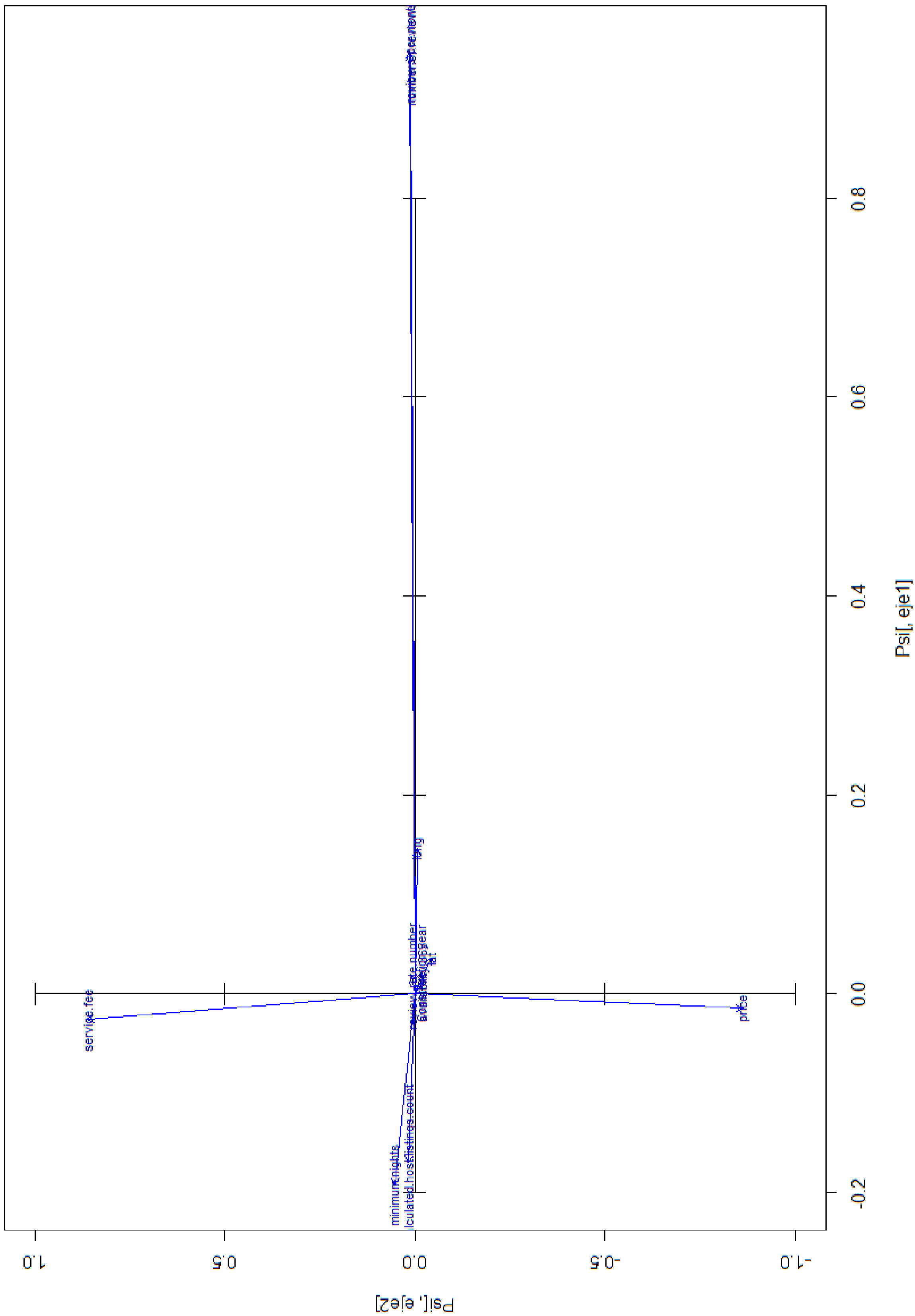
PCA analysis

To visualise how meaningful numeric variables are, we will use the PCA analysis. It will also show how they affect each other. The variables we have selected to calculate its PCA are: long, lat, price, number.of.reviews, review.rate.number, service.fee, calculated.host.listings.count, Construction.year, minimum.nights, reviews.per.month and availability.365.

It is observable how two distinct axis have formed. On the vertical axis there is price and service fee parallel and inverse, signifying that the value of each house is a differentiator between rented houses, and that cheaper houses tend to have more expensive fees.

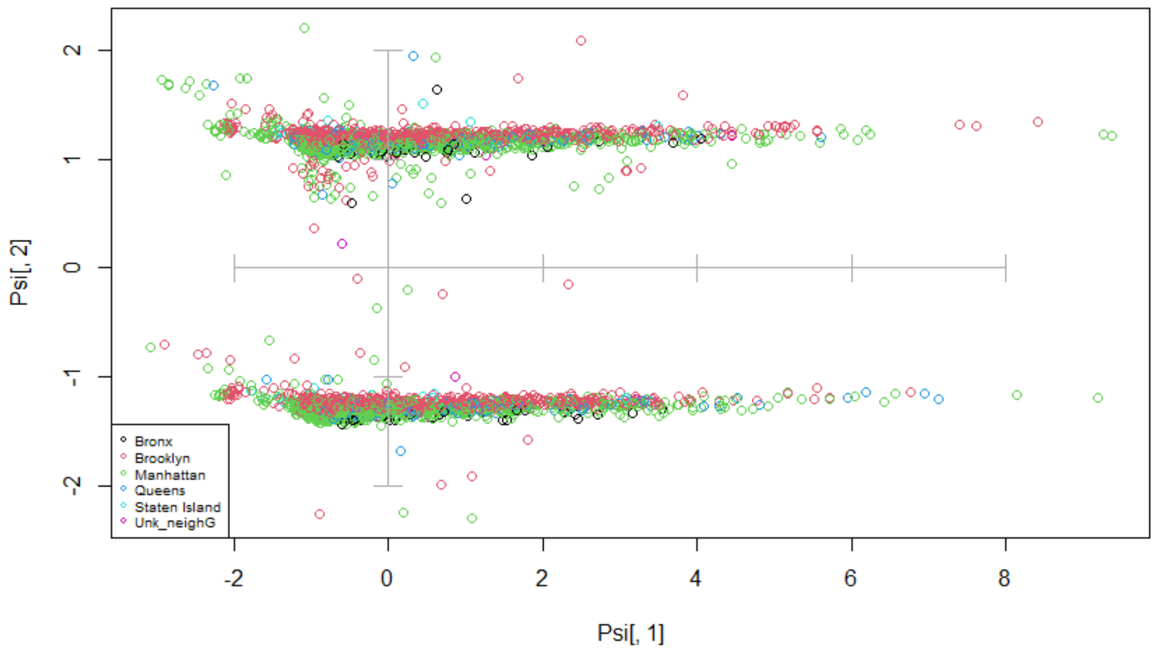
On the horizontal axis, we can see the inverse relationship between minimum number of nights and number of reviews or reviews per month, and the direct relationship between the last two. We can assume that a bigger number of reviews is equivalent to a bigger number of hosts, so this helps us visualise how a minimum number of nights attracts longer lasting hosts, while less restricting houses bring a more dynamic occupation. It could also mean that a higher number of minimum nights could deter hosts, lessening their numbers. We can also note that houses with more hosts also have a smaller calculated host listing count.

PCA can also be used to visualise the effects of qualitative variables painting the individual elements of the data base in the map.



Psi[, eje1]
Psi[, eje2]
service.fee
price
long
lat
minimum.nights
-0.2
0.0
0.2
0.4
0.6
0.8
-1.0
-0.5
0.5
1.0

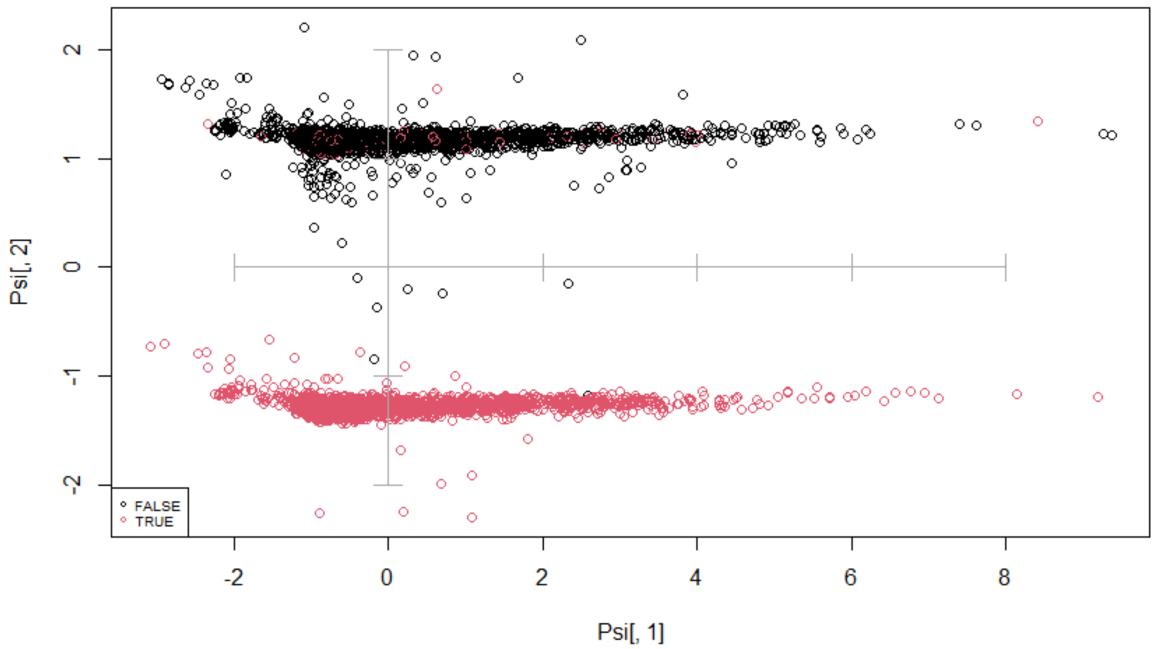
PCA analysis of numeric variables



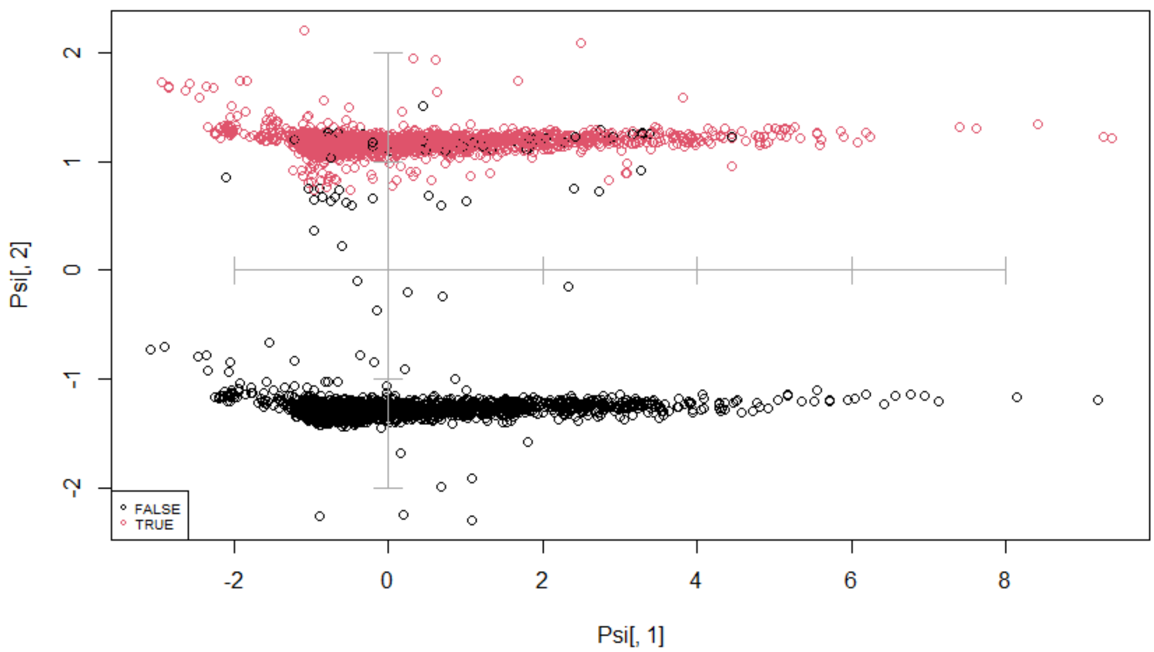
elements painted by its neighbourhood group

Two stripes of elements can be observed in the plot. There is reason to believe that some qualitative variable can be dividing its elements into them. But after trying everyone, none seems to have any notable effect, with the exception of neighbourhood.group, which divides each stripe into two, with most of the upper elements belonging to Brooklyn and the lower to Manhattan.

As the Y axis referred to the economical value of the house, we decided to create two new qualitative variables. One referring to the service fee and the other to price. We will define these variables by whether the element is more expensive than the mean of all elements.



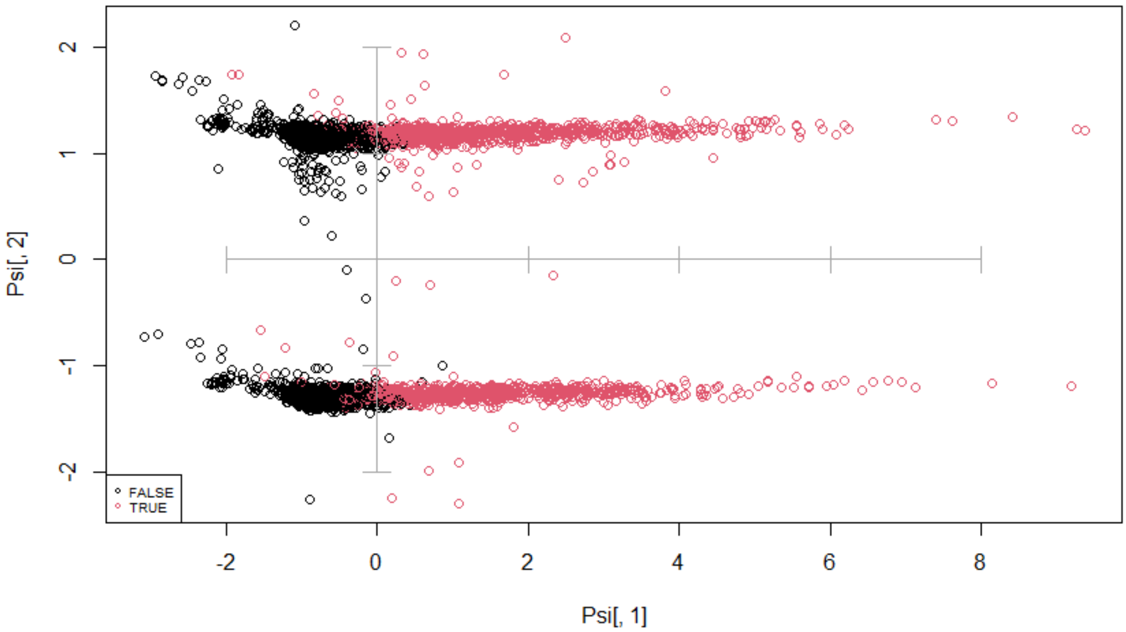
elements painted by above.mean.price



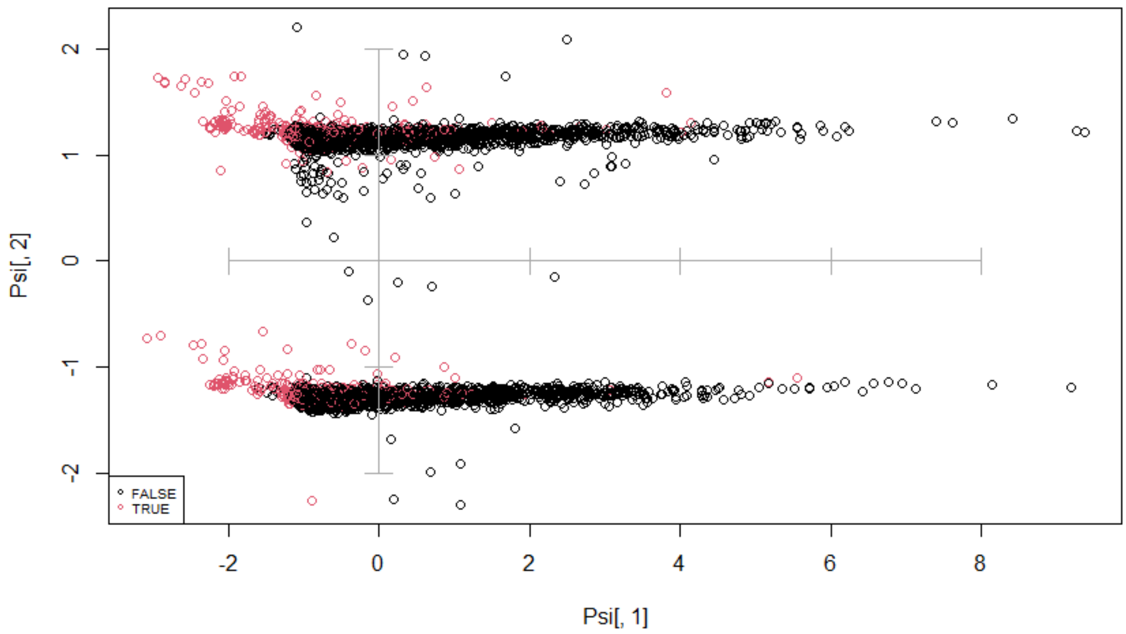
elements painted by above.mean.service.fee

The two new qualitative variables perfectly explains the existence of the two stripes in the Y axis. With the elements in the higher stripe having high price and low service fee and the elements in the lower having low price and high service fee. The elements left in the middle have both low price and low service fee. This conclusion puts sense on the first plot, as Manhattan has a higher land value than Brooklyn.

As we realised in previous stages of the PCA, the X axis depends on the number of hosts, so if we create a new variable for it we will have a similar result.



elements painted by above.mean.reviews.per.month



elements painted by above.mean.minimum.nights

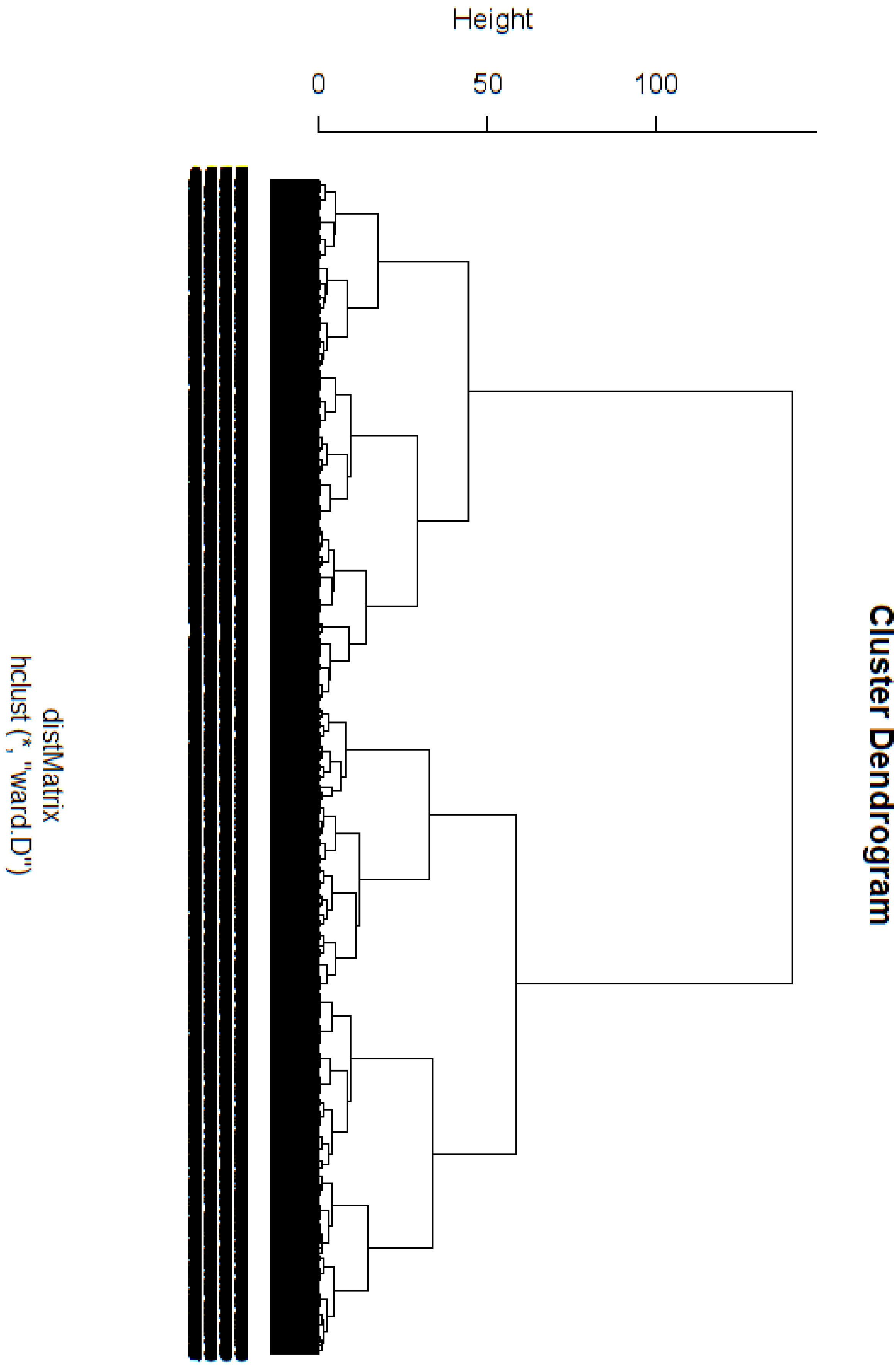
We can observe how the X axis is highly dependent on the number of reviews per month of a house, with the elements in the left of the plot having long lasting hosts, or less hosts in general, and the ones in the right having a higher number of hosts.

In conclusion, a small set of numeric variables create a meaningful difference between all other variables to the elements of the dataset. Creating this effect of the two groups divided by their cost, and moving in the x axis depending on the number of hosts.

Hierarchical Clustering

To perform hierarchical clustering we used every variable in our dataset except the *id* column, which was not useful for the analysis. For the similarity metric we used the Gower dissimilarity coefficient to the square, very useful with heterogeneous variables. The aggregation criteria used by the *hclust* R function is the Ward's criterion.

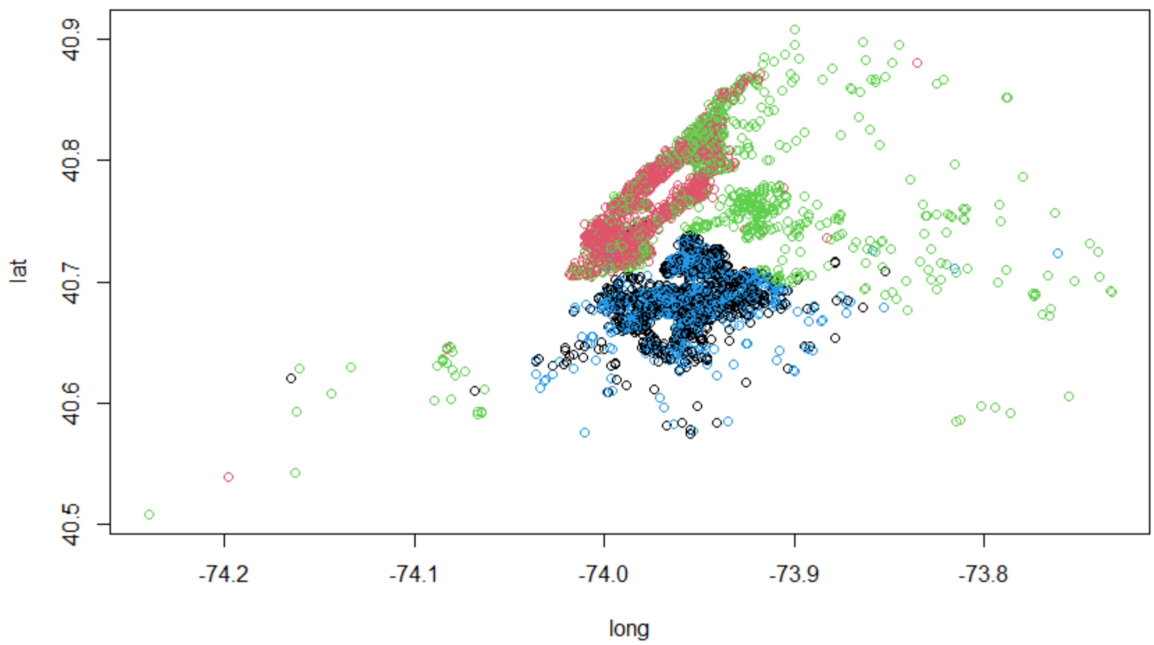
The resulting dendrogram after performing the clustering was this one below:


Cluster Dendrogram
Height
0
50
100
distMatrix
hclust (*, "ward.D")

After observing the results of our dendrogram, we decided that the best partition was 4 classes. Then we got these sizes on our four different groups:

```
c2
   1    2    3    4
 806 1537 1213 1444
```

Generating plots on R we could observe how the pairs of our variables were related to one another. The most impressive plot was the one that related *longitude* with *latitude*. As we can see in the picture below we got a pretty accurate New York city map with how our clusters are more or less the three biggest neighbourhoods: Manhattan, Brooklyn and Queens.



Profiling of clusters

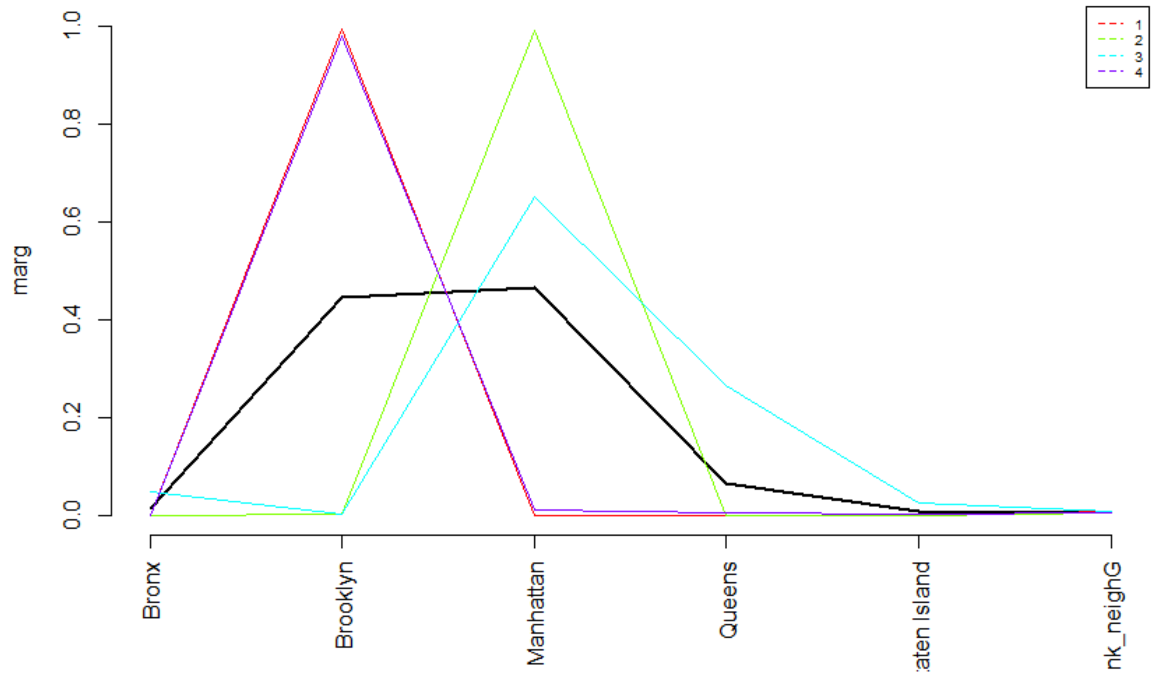
Once we have created the clusters, we must use profiling to define the characteristics of each class and analyse the patterns found during hierarchical clustering.

We shall accomplish this through the use of box plots, CPG and bivariate plots.

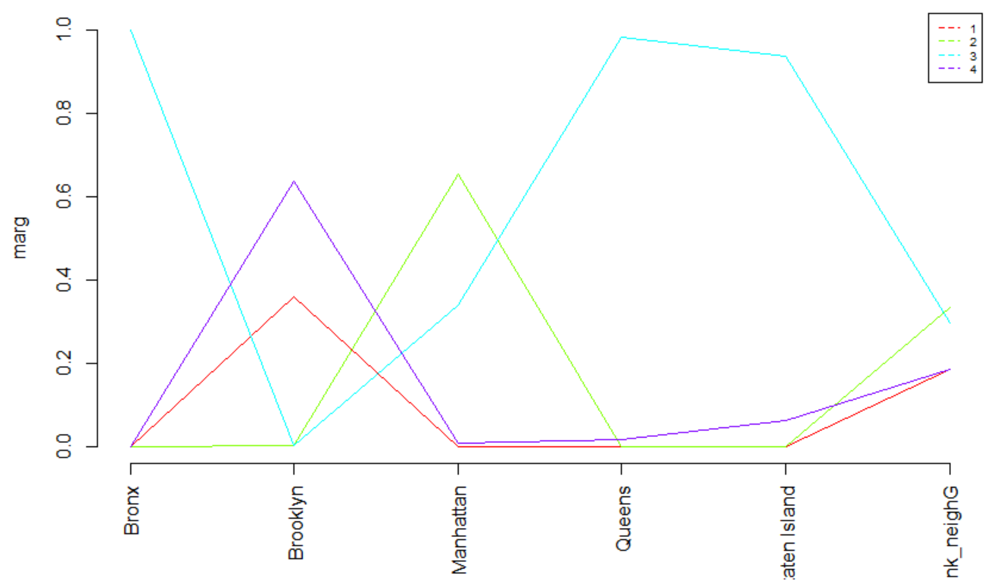
Profiling variables

Neighbourhood group

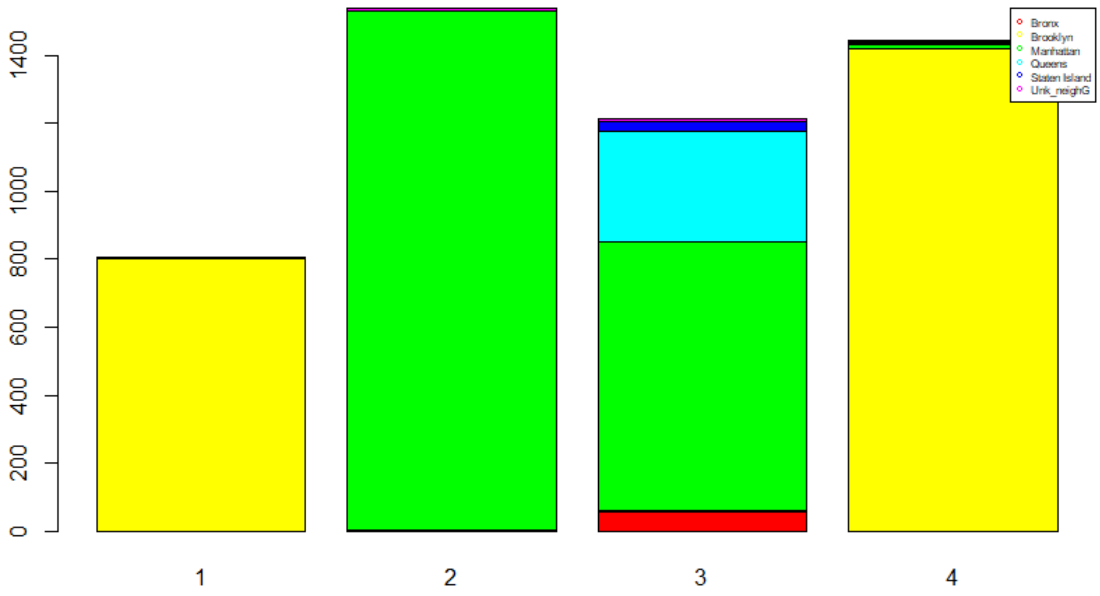
The first differentiator of classes is its neighbourhood group, we can observe it in the following plots:



Distribution by class



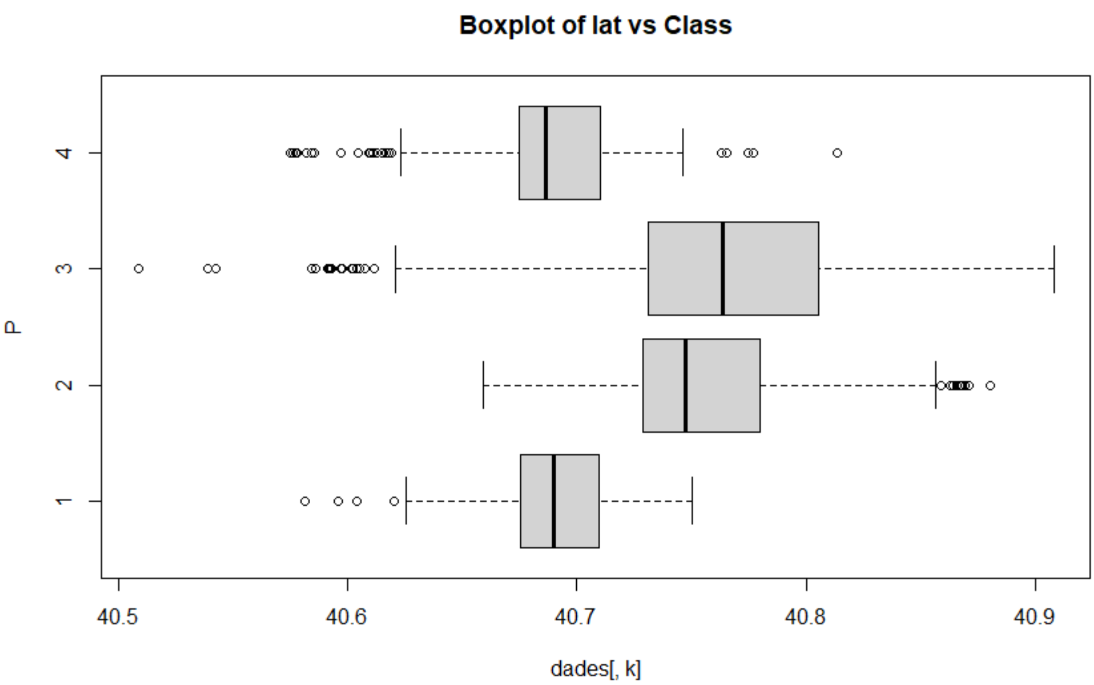
P value of classes

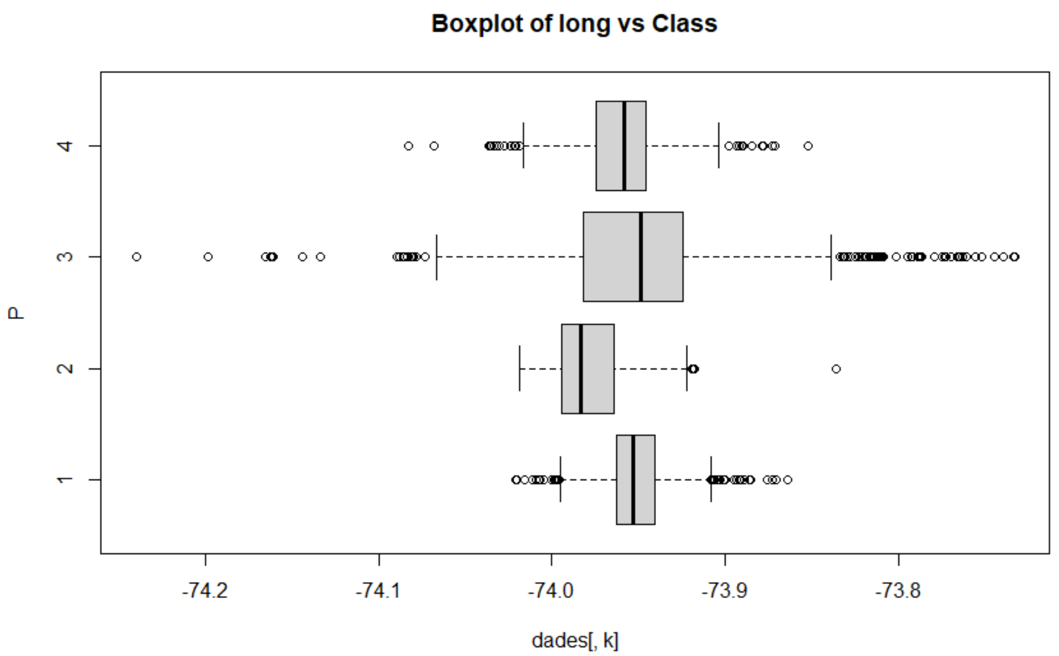


Class histogram

Latitude and longitude

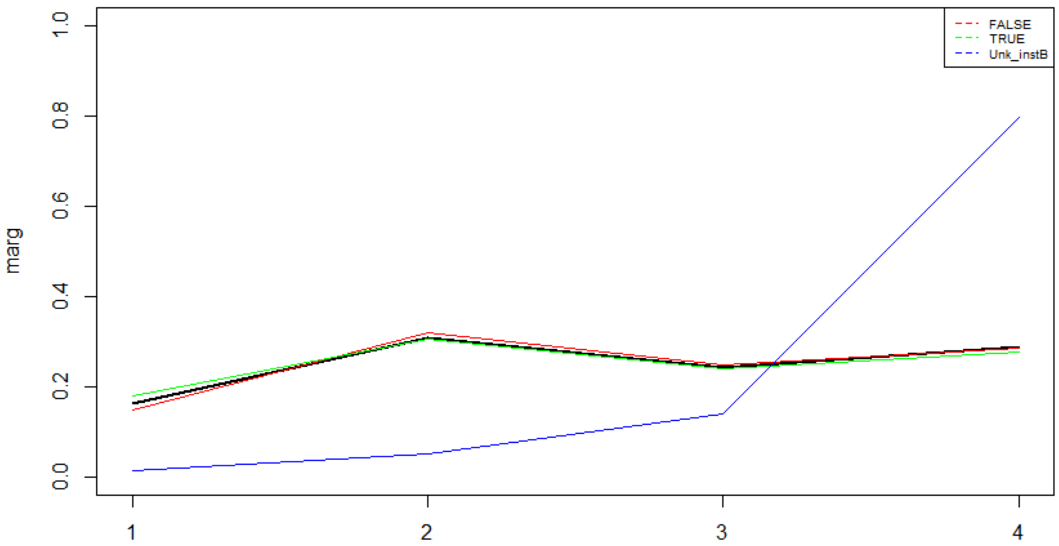
Knowing its geographical location is important for its class distribution, latitude and longitude are both divisive variables. The 3rd cluster seems to have a more varied location.



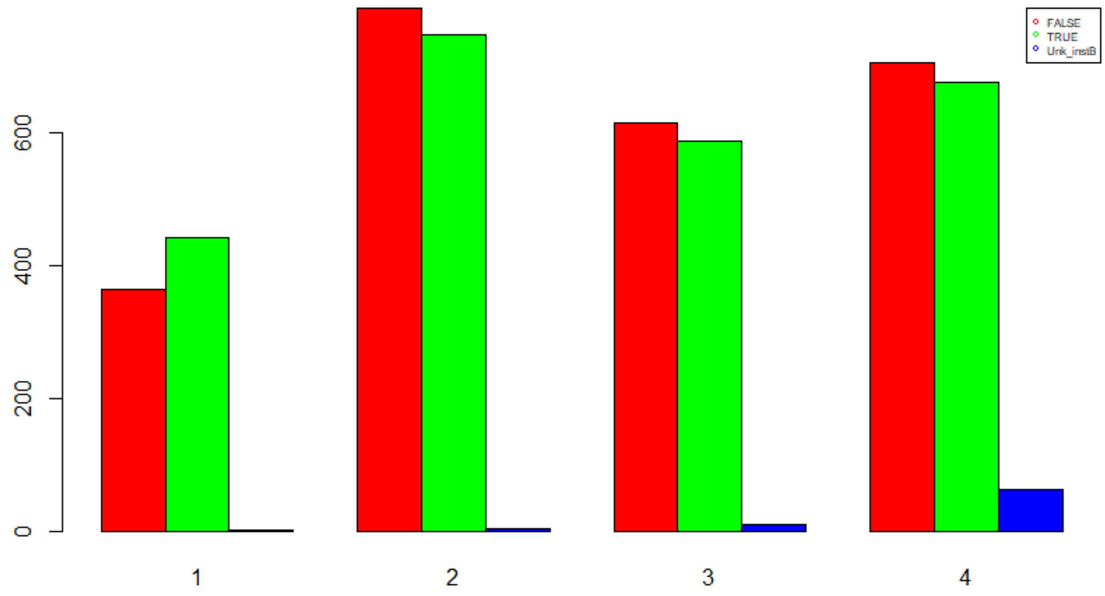


Instant Bookable

Another significant value is instant bookable. We can observe how most unknowns belong to the 4th class. Another observation is that the 1st class is the only class with more Trues than Falses.



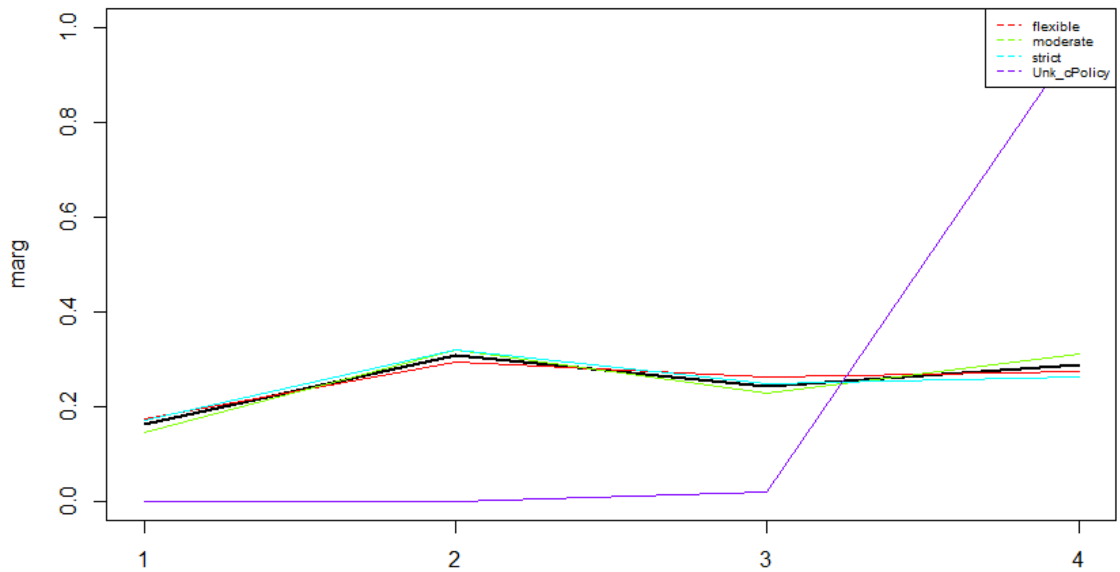
P value of classes



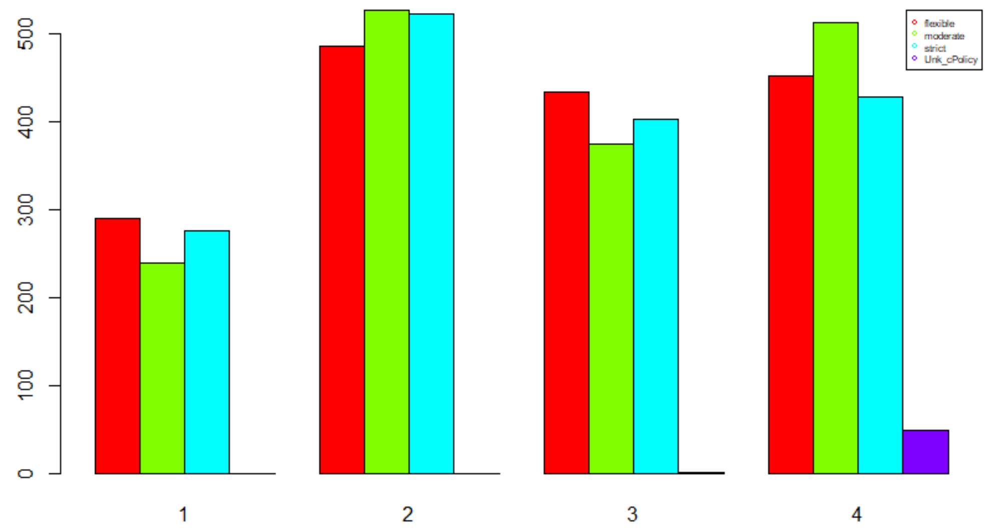
Class histogram

Cancellation policy

Similarly to instant bookable, all missing data elements belong to the 4th class. We can also see that 1st and 3rd classes have more flexible cancellation policies, while 2nd and 4th have more moderate policies; 2nd having even more strict policies than flexible ones.



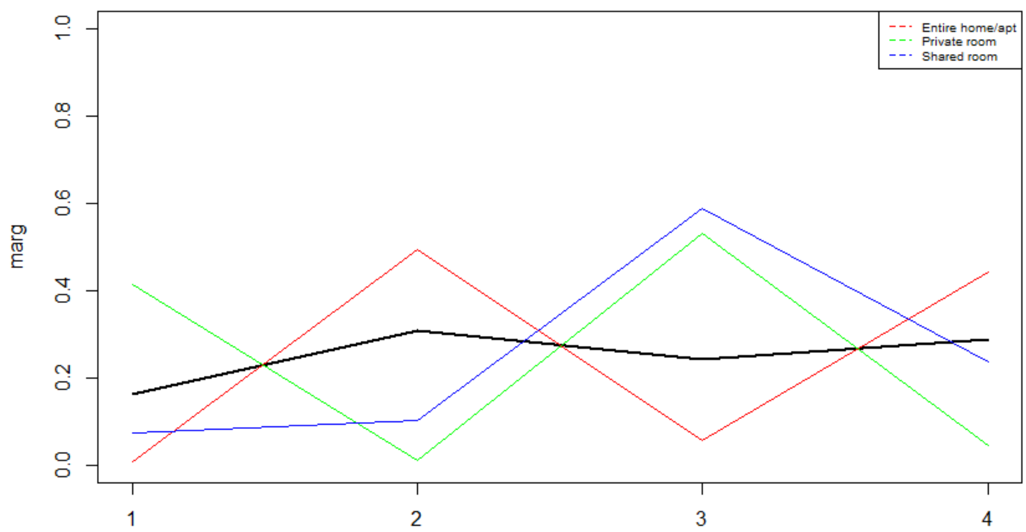
P value of classes



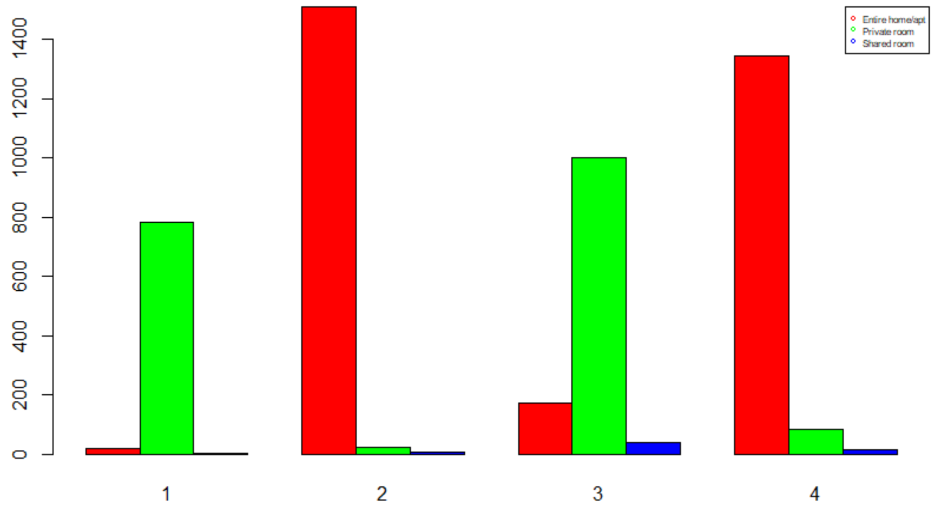
Class histogram

Room type

The kind of room rented is also a significant value to observe. We can still see the differences between 1st and 3rd with 2nd and 4th classes. The first set has more private rooms, 3rd having most shared rooms. Whilst the 2nd and 4th has more entire home apartments.



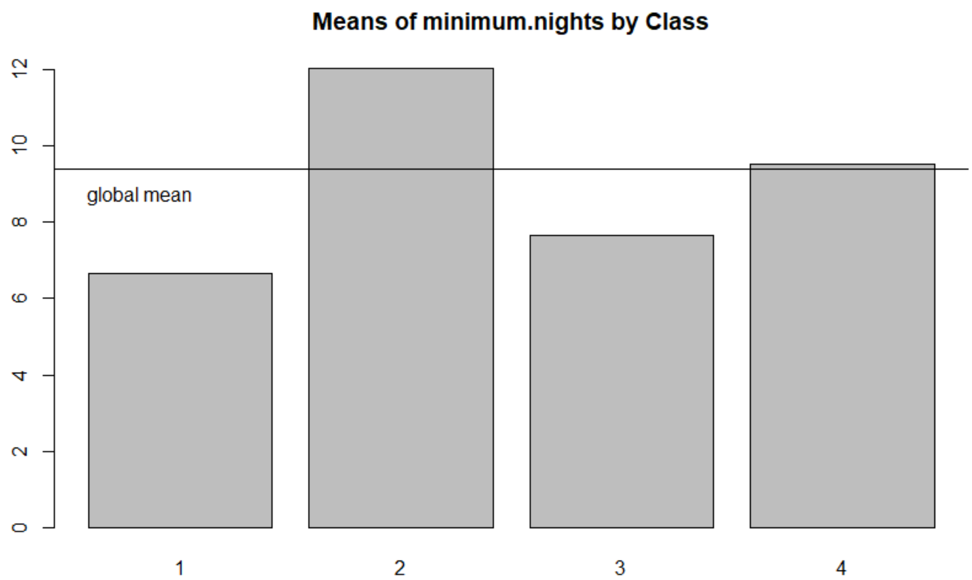
P value of classes



Class histogram

Minimum nights

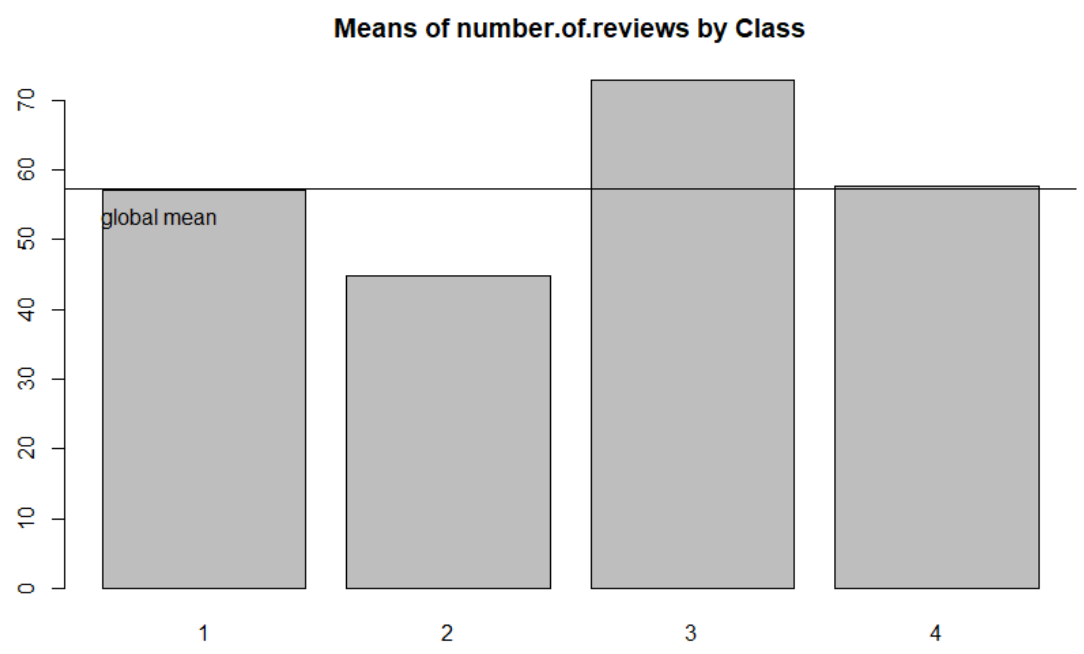
The minimum of nights to rent is also a very important variable for representing clusters. We can still observe the trend of 2nd and 4th having more strict policies while 1st and 3rd have more dynamic occupation.



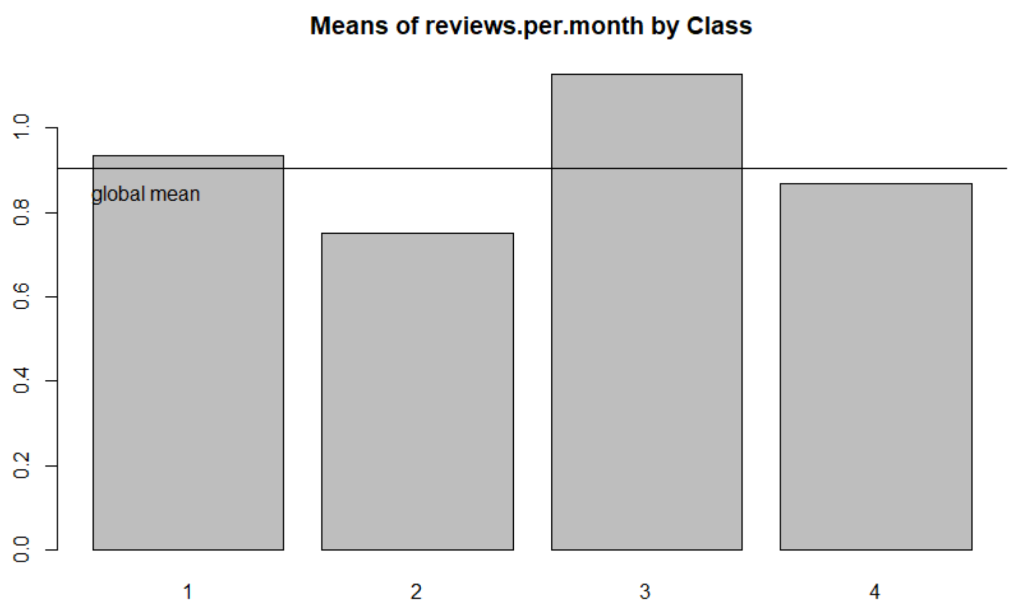
Mean histogram

Number of reviews and reviews per month

As shown in the PCA analysis, both variables show the same distinctiveness between elements, and both are important too. 1st and 3rd show more number of reviews per month compared to 2nd 4th. Specially in the 3rd.



Means of number of reviews histogram



Mean of reviews per month histogram

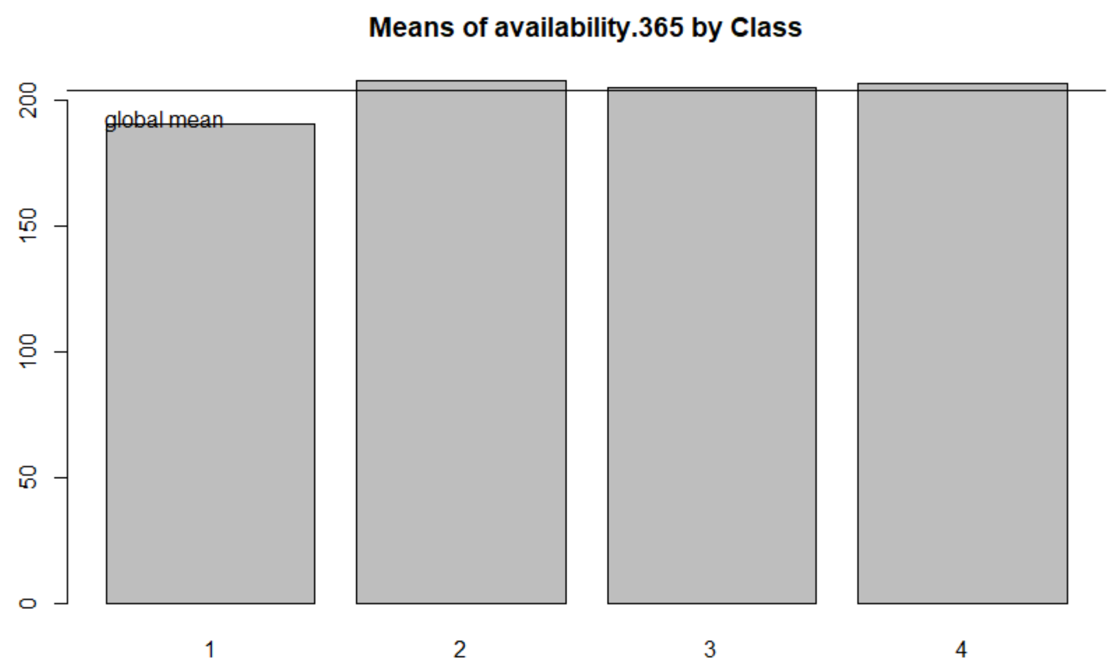
Calculated listing counts

It is observable that clusters with long lasting hosts also have a bigger amount of listings. This could mean that hosts look up more houses when they are planning a long rent.

Mean of calculated.host.listings.count histogram

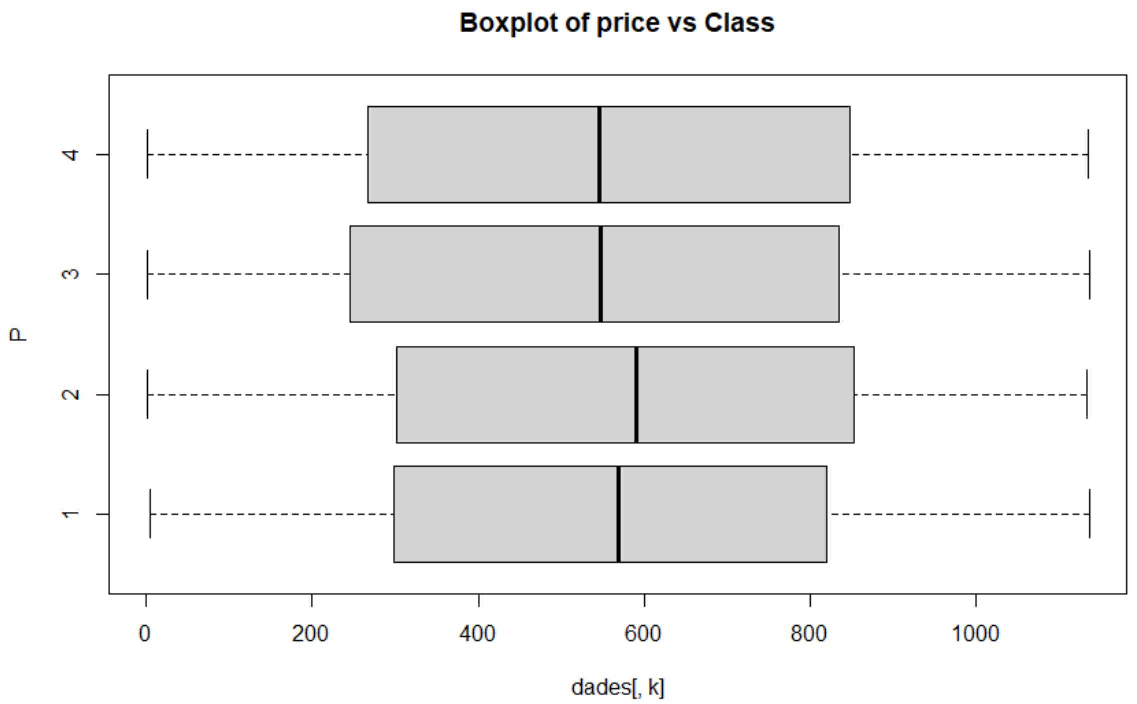
availability .365

The availability shows a little change to the clusters, with less availability in the houses with more occupation.

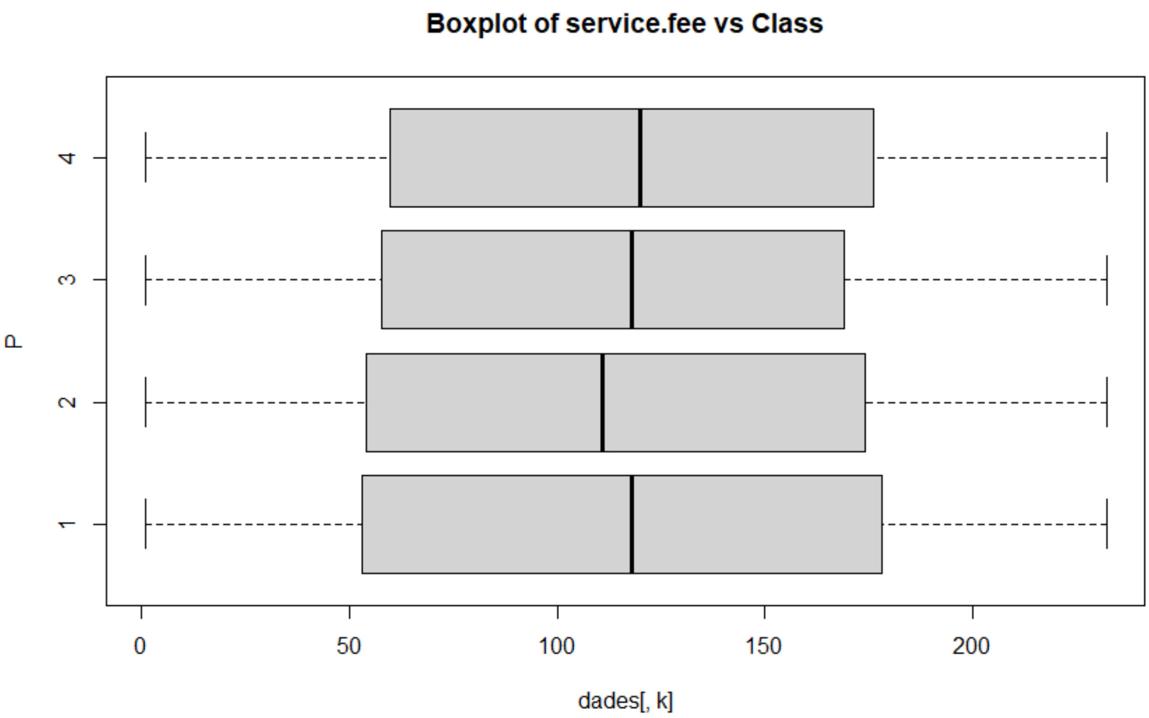


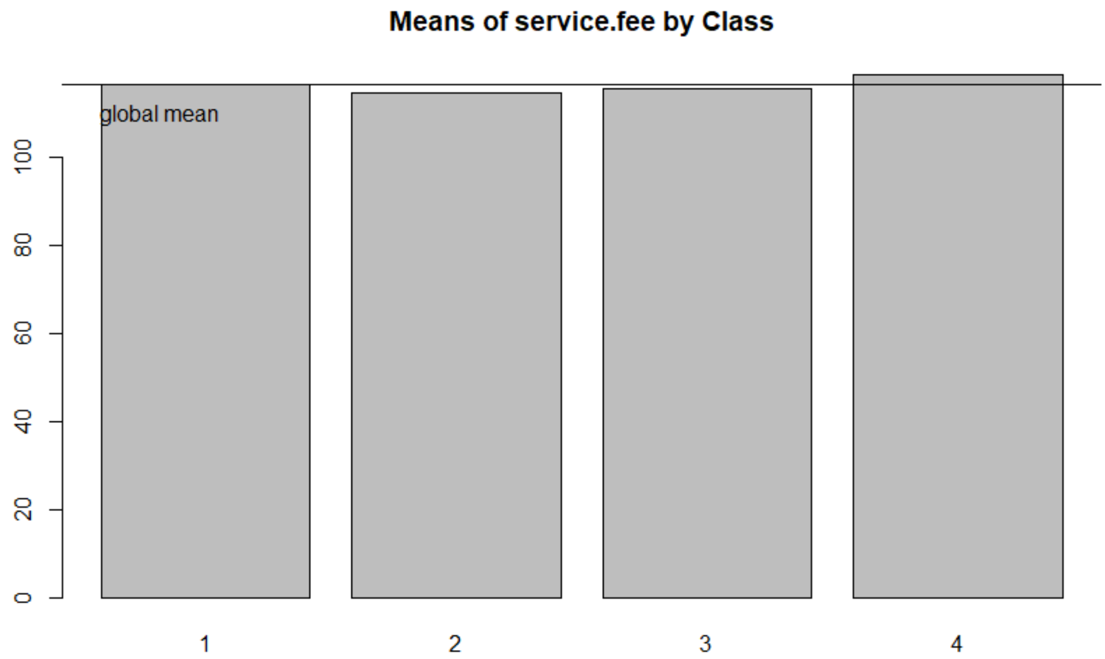
Price and service fee

Even though the PCA price and service fee resulted in dividing the elements very well, they ended up being not very significant to describe clusters. A very small difference is observed describing the patterns mentioned before but it is not of much importance.


Boxplot of price vs Class
P
4
3
2
1
0
200
400
600
800
1000
dades[, k]


Means of price by Class
global mean
500
400
300
200
100
0
1
2
3
4


Boxplot of service.fee vs Class
P
1
2
3
4
0
50
100
150
200
dades[, k]


Means of service.fee by Class
global mean
0
20
40
60
80
100
1
2
3
4

Cluster description

After the differences between classes, we reached the following conclusions:

- Cluster 1: Primarily located in Brooklyn, is characterised by having a dynamic occupation. It probably targets tourism, due to being mostly private rooms, their low availability 365 and the high number of hosts per month. Their low minimum nights and instant bookability support this conclusion.

- Cluster 2: With virtually all elements in Manhattan, the second cluster describes a more long lasting, less tourist oriented kind of target host. It has the greatest minimum number of nights and the lowest reviews per month. They are also primarily apartments, which might be appealing to hosts with long stays.

- Cluster 3: This cluster has the most diversity of location, with its elements distributed between Manhattan, Queens, Staten Island Bronx, and a small portion of them in Brooklyn. It also describes a tourist target, or at least more precarious due to being mostly private rooms and containing most shared rooms while being the cheapest. They also have the biggest amount of reviews, which supports the idea of it being targeted to tourists.

- Cluster 4: Having most of their elements in Brooklyn, this cluster seems to be a more economic version of cluster 2. Being mostly apartments, having higher minimum nights and low reviews per month; it might also be oriented to attract long lasting hosts. The fact that it has the biggest amount of listings also supports the idea of a more humble, long lasting host, as it might be an important decision to the supposed user. This cluster could also be described as the dumpster, as it contains most elements with unknown data.

PCA and Clustering comparison

In the PCA analysis we found two main axis of difference. One referring to the price and the other referring to the dynamism of occupation. Clustering uses the amount of users per month as a significant differentiator between classes.

On the other hand, it doesn't give much importance to the axis describing the price. Instead it uses more neighbourhood groups. But price still makes a small difference between neighbourhood groups, so it is not that far from the conclusions of the PCA.

Conclusions

In this section we draw conclusions about the previous sections and the whole project. In summary we have preprocessed the data in order to obtain a dataset ready for further analysis, we have analysed the data descriptively, did a PCA analysis and hierarchical clustering on the data and finally did profiling of the clusters.

After finding a suitable dataset we have imputed missing data, treated outliers and reduced the number of rows such that the algorithms used for data analysis work efficiently. This step is very important as the quality of the further data analysis is highly dependent on the data itself. This is also why this procedure usually takes up a considerable amount of time during the whole data mining process. The preprocessed data can then be used to draw further conclusions about the data.

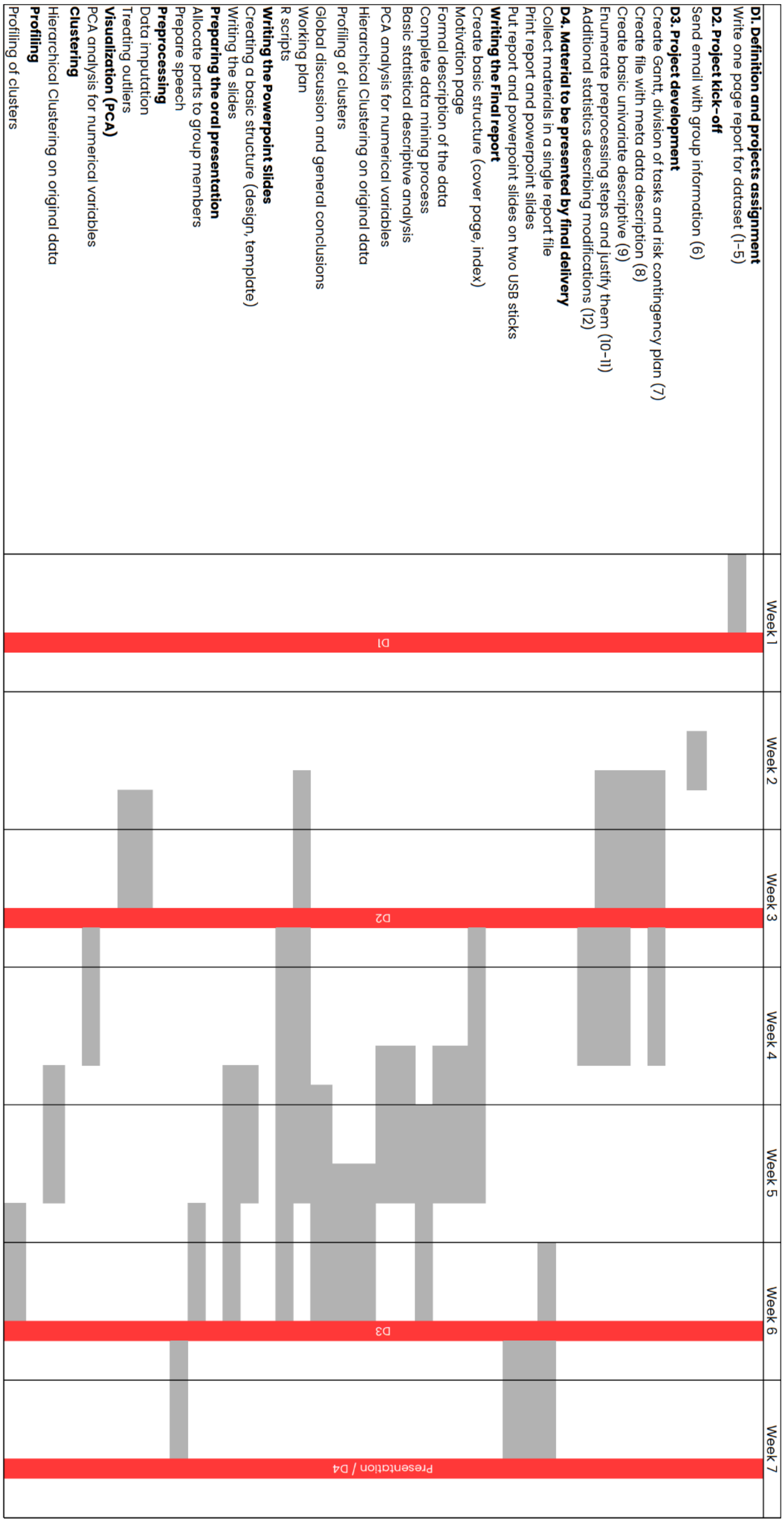
On the preprocessed data we have then performed a basic descriptive statistical analysis using different statistical visualisation techniques. This step provided a good overview of the whole dataset and its variables.

In the next step we then did the principal component analysis to obtain information about how much information the various numeric variables contain. The PCA allowed us to visualise this metainformation on the numeric variables. In some projects this information could be used to do a dimensionality reduction and drop dimensions with less information for more efficient data processing.

In a next step we performed a hierarchical clustering on the data to group similar data points. From the dendrogram we learned that a good number of clusters is four. These clusters when mapped out on a graph with latitude and longitude also presented the different neighbourhoods of New York. In a final step we then did the profiling of these clusters.

In conclusion we learned what the most important dimensions of our data are, how the variables are related to each other and what information each variable contains. This information could be used for market research in the tourism industry or for city planning for the city of New York.

Working plan



	Xavi	Ramamon	Mario	Rykart	David
D1. Definition and projects assignment					
Write one page report for dataset (1-5)		x	x	x	
D2. Project kick-off					
Send email with group information (6)	x	x		x	
D3. Project development					
Create Gantt, division of tasks and risk contingency plan (7)			x		x
Create file with meta data description (8)	x		x	x	
Create basic univariate descriptive (9)	x			x	
Enumerate preprocessing steps and justify them (10-11)	x		x		
Additional statistics describing modifications (12)	x		x		
D4. Material to be presented by final delivery					
Collect materials in a single report file	x				x
Print report and powerpoint slides	x	x			
Put report and powerpoint slides on two USB sticks	x	x			
Writing the Final report					
Create basic structure (cover page, index)		X			x
Motivation page			x		x
Formal description of the data			x		x
Data source presentation			x		x
Complete data mining process				x	x
Basic statistical descriptive analysis	x			x	
PCA analysis for numerical variables		x		x	
Hierarchical Clustering on original data		x	x		
Profiling of clusters	x	x			
Global discussion and general conclusions	x	X			x
Working plan	x				x
R scripts	x	x	x	x	x
Writing the Powerpoint Slides					
Creating a basic structure (design, template)	x				x
Writing the slides	x				x
Preparing the oral presentation					
Allocate parts to group members	x	x	x	x	x
Prepare speech	x	x	x	x	x
Preprocessing					
Data imputation	x		x		
Treating outliers	x		x		
Visualization (PCA)					
PCA analysis for numerical variables		x	x	x	
Clustering					
Hierarchical Clustering on original data	x	x	x		
Profiling					
Profiling of clusters		x	x		x

Risk	How to prevent	How to manage
A team member leaves the course	All tasks have at least two members assigned	Pending work reassigned to rebalance efforts
Data has weak structures and models do not perform well	Ensure that technical assumptions of models hold on data	Change to models without hypothesis that do not fit the data
Missing a deadline	Add deadlines to personal calendar	Watch calendar
Too much work in the end	Create Gantt and manage time	Stick to time schedule according to the working plan
Lack of knowledge	Exchange knowledge in team, attend lectures and listen carefully	Ask questions to fill knowledge gaps
Forgetting material for presentation	Set reminders	Have material ready the day before presentation

R Scripts

The descriptive script will not be included because only consisted in changing the used table to ours'.

Redefining types script

```
setwd("C:/Users/xavim/Desktop")
dd <- read.csv("Airbnb_Open_Data.csv", header=T, stringsAsFactors=TRUE)

n<-dim(dd)[1]
n
K<-dim(dd)[2]
K

names(dd)
summary(dd)
attach(dd)
sapply(dd, class)

price    <- as.numeric(price)
service.fee <- as.numeric(service.fee)
last.review <- as.Date(last.review, format ="%m/%d/%Y")

dd[,13]  <-price
dd[,14]  <-service.fee
dd[,17]  <-last.review

write.table(dd, file = "Airbnb_Open_Data.csv", sep = ",", na = "NA", dec = ".", row.names = FALSE, col.names = TRUE)
```

Recoding missings and imputing 1nn

```
setwd("C:/Users/xavim/Desktop/md")
dd <- read.csv("Airbnb_Open_Data.csv", header=T)
attach(dd)
#Missing data treatment

#Detect
names(dd)
table(dd[,1]=="")
table(dd[,2]=="")
table(dd[,3]=="")
table(dd[,4]=="")
table(dd[,5]=="")
table(dd[,6]=="")
table(is.na(dd[ , 7]))
table(is.na(dd[ , 8]))
table(is.na(dd[ , 9]))
```

```
table(dd[,10]=="")
table(dd[,11]=="")
table(is.na(dd[ , 12]))
table(dd[,13]=="")
table(dd[,14]=="")
table(is.na(dd[ , 15]))
table(is.na(dd[ , 16]))
table(is.na(dd[ , 17]))
table(is.na(dd[ , 18]))
table(is.na(dd[ , 19]))
table(is.na(dd[ , 20]))
table(is.na(dd[ , 21]))

#For non structural missings in qualitative variables, just keep as a new modality. Only if required inpute or describe appart
#you already have from previous treatment of factors
levels(host_identity_verified)<-c(levels(host_identity_verified),"Unk_ver")
host_identity_verified[host_identity_verified==""]<-"Unk_ver"
levels(instant_bookable)<-c(levels(instant_bookable),"Unk_instB")
instant_bookable[is.na(instant_bookable)]<-"Unk_instB"
levels(last.review)<-c(levels(last.review),"Unk_lReview")
last.review[is.na(last.review)]<-"Unk_lReview"
host.name[host.name==""]<-"Unk_hName"
levels(host.name)<-c(levels(host.name),"Unk_hName")
neighbourhood.group[neighbourhood.group==""]<-"Unk_neighG"
levels(neighbourhood.group)<-c(levels(neighbourhood.group),"Unk_neighG")
neighbourhood[neighbourhood==""]<-"Unk_neigh"
levels(neighbourhood)<-c(levels(neighbourhood),"Unk_neigh")
cancellation_policy[cancellation_policy==""]<-"Unk_cPolicy"
levels(cancellation_policy)<-c(levels(cancellation_policy),"Unk_cPolicy")

# Recode missing data to NA

host_identity_verified[host_identity_verified==""] <- NA
host.name[host.name==""] <- NA
neighbourhood.group[neighbourhood.group==""] <- NA
neighbourhood[neighbourhood==""] <- NA
cancellation_policy[cancellation_policy==""]<- NA

#start substituting the structural missing values.
#with remaining, impute: Knn, MIMMI, MICE (multiple imputation, only if you know well)

# IMPUTATION By THE 1NN

library(class)

# FOR EVERY INDIVIDUAL WITH MISSING LOOK FOR THE MOST SIMILAR INDIVIDUAL
# wrt REMAINING VARIABLES
# For more robustness average the values of k-NN in general (with small k)

#For several Variables:

#built indexes of numerical variables that require inputation

uncompletevars<-c(7,8,16,20,15,19,12,21,18)

#better if you sort them by increasing number of missing values

fullvariables<-c(1,2,13,14)
aux<-dd[,fullvariables]
```

```
dim(aux)
names(aux)

for (k in uncompletevars){
  aux1 <- aux[!is.na(dd[,k]),]
  dim(aux1)
  aux2 <- aux[is.na(dd[,k]),]
  dim(aux2)

  RefValues<- dd[!is.na(dd[,k]),k]
  #Find nns for aux2
  knn.values = knn(aux1,aux2,RefValues)

  #CARE: neither aux1 nor aux2 can contain NAs

  #CARE: knn.ing is generated as a factor.
  #Be sure to retrieve the correct values

  dd[is.na(dd[,k]),k] = as.numeric(as.character(knn.values))
  fullVariables<-c(fullVariables, k)
  aux<-dd[,fullVariables]
}

dim(dd)
summary(dd)

#check for outliers
#how?

# SAVING THE TRANSFORMATIONS IN A INTERNAL R FILE

save.image("Airbnb")

#saving the dataframe in an external file
write.table(dd, file = "Airbnb_Open_Data.csv", sep = ",", na = "NA", dec = ".", row.names = FALSE, col.names = TRUE)
```

PCA script

```
#Neighbourhood Group
varcat=as.factor(dd[,5])
plot(Psi[,1],Psi[,2],col = varcat)
axis(side=1, pos= 0, labels = F, col="darkgray")
axis(side=3, pos= 0, labels = F, col="darkgray")
axis(side=2, pos= 0, labels = F, col="darkgray")
axis(side=4, pos= 0, labels = F, col="darkgray")
legend("bottomleft",levels(varcat),pch=1,col = 1:length(levels(varcat)), cex=0.6)


#Mean Price
varcat=as.factor(dd[,22])
plot(Psi[,1],Psi[,2],col = varcat)
```

```
axis(side=1, pos= 0, labels = F, col="darkgray")
axis(side=3, pos= 0, labels = F, col="darkgray")
axis(side=2, pos= 0, labels = F, col="darkgray")
axis(side=4, pos= 0, labels = F, col="darkgray")
legend("bottomleft",levels(varcat),pch=1,col = 1:length(levels(varcat)), cex=0.6)

#Mean Service Fee
varcat=as.factor(dd[,23])
plot(Psi[,1],Psi[,2],col = varcat)
axis(side=1, pos= 0, labels = F, col="darkgray")
axis(side=3, pos= 0, labels = F, col="darkgray")
axis(side=2, pos= 0, labels = F, col="darkgray")
axis(side=4, pos= 0, labels = F, col="darkgray")
legend("bottomleft",levels(varcat),pch=1,col = 1:length(levels(varcat)), cex=0.6)

#Mean Reviews per month
varcat=as.factor(dd[,24])
plot(Psi[,1],Psi[,2],col = varcat)
axis(side=1, pos= 0, labels = F, col="darkgray")
axis(side=3, pos= 0, labels = F, col="darkgray")
axis(side=2, pos= 0, labels = F, col="darkgray")
axis(side=4, pos= 0, labels = F, col="darkgray")
legend("bottomleft",levels(varcat),pch=1,col = 1:length(levels(varcat)), cex=0.6)

#Mean Miminum Nights
varcat=as.factor(dd[,25])
plot(Psi[,1],Psi[,2],col = varcat)
axis(side=1, pos= 0, labels = F, col="darkgray")
axis(side=3, pos= 0, labels = F, col="darkgray")
axis(side=2, pos= 0, labels = F, col="darkgray")
axis(side=4, pos= 0, labels = F, col="darkgray")
legend("bottomleft",levels(varcat),pch=1,col = 1:length(levels(varcat)), cex=0.6)

#all qualitative together
plot(Psi[,eje1],Psi[,eje2],type="n")
axis(side=1, pos= 0, labels = F, col="cyan")
axis(side=3, pos= 0, labels = F, col="cyan")
axis(side=2, pos= 0, labels = F, col="cyan")
axis(side=4, pos= 0, labels = F, col="cyan")
```

```
#nominal qualitative variables

dcat<-c(5,11,22,23,24,25)
#divide categoricals in several graphs if joint representation saturates

#build a palette with as much colors as qualitative variables

#colors<-c("blue","red","green","orange","darkgreen")
#alternative
colors<-rainbow(length(dcat))

c<-1
for(k in dcat){
  seguentColor<-colors[c]
fdic1 = tapply(Psi[,eje1],dd[,k],mean)
fdic2 = tapply(Psi[,eje2],dd[,k],mean)

text(fdic1,fdic2,labels=levels(dd[,k]),col=seguentColor, cex=0.6)
c<-c+1
}
legend("bottomleft",names(dd)[dcat],pch=1,col=colors, cex=0.6)

#determine zoom level
#use the scale factor or not depending on the position of centroids
# ES UN FACTOR D'ESCALA PER DIBUIXAR LES FLETXES MES VISIBLES EN EL GRAFIC
#fm = round(max(abs(Psi[,1])))
# fm=20

#scale the projected variables
# X<-fm*U[,eje1]
# Y<-fm*U[,eje2]

#represent numerical variables in background
plot(Psi[,eje1],Psi[,eje2],type="n",xlim=c(-1,1), ylim=c(-3,1))
plot(X,Y,type="none",xlim=c(-2,1), ylim=c(-2,2))
 axis(side=1, pos= 0, labels = F, col="cyan")
 axis(side=3, pos= 0, labels = F, col="cyan")
 axis(side=2, pos= 0, labels = F, col="cyan")
```

```
 axis(side=4, pos= 0, labels = F, col="cyan")

#add projections of numerical variables in background
 arrows(ze, ze, X, Y, length = 0.07,col="lightgray")
 text(X,Y,labels=etiq,col="gray", cex=0.7)

#add centroids
 c<-1
 for(k in dcat){
  seguentColor<-colors[c]

  fdic1 = tapply(Psi[,eje1],dd[,k],mean)
  fdic2 = tapply(Psi[,eje2],dd[,k],mean)

 #points(fdic1,fdic2,pch=16,col=seguentColor, abels=levels(dd[,k]))
 text(fdic1,fdic2,labels=levels(dd[,k]),col=seguentColor, cex=0.6)
  c<-c+1
}
legend("bottomleft",names(dd)[dcat],pch=1,col=colors, cex=0.6)
```

Clustering

```
setwd("C:/Users/xavier.marti.llull/Desktop")
dd <- read.csv("OUTLIERS_OUT.csv",header=T, sep=",", stringsAsFactors=TRUE);
names(dd)
dim(dd)
summary(dd)
attach(dd)

#set a list of numerical variables
dcon <- data.frame
(lat,price,number.of.reviews,review.rate.number,long,service.fee,last.review,calculated.host.listings.count,Construction.year,minimu
m.nights,reviews.per.month,availability.365)
dim(dcon)

# CLUSTERING
#move to Gower mixed distance to deal
#simoultaneously with numerical and qualitative data
```

```
library(cluster)
#dissimilarity matrix
actives<-c(2:16)
dissimMatrix <- daisy(dd[,actives], metric = "gower", stand=TRUE)
distMatrix<-dissimMatrix^2
h1 <- hclust(distMatrix,method="ward.D")  # NOTICE THE COST

plot(h1)
c2 <- cutree(h1,4)
#class sizes
table(c2)

#comparing with other partitions
names(dd)
# service.fee
boxplot(dd[,14]~c2, horizontal=TRUE)
#lat
boxplot(dd[,7]~c2, horizontal=TRUE)
#long
boxplot(dd[,8]~c2, horizontal=TRUE)
# price
boxplot(dd[,13]~c2, horitzontal=TRUE)
# Construction.year
boxplot(dd[,12]~c2, horitzontal=TRUE)
# minimum.nights
boxplot(dd[,15]~c2, horitzontal=TRUE)
# number.of.reviews
boxplot(dd[,16]~c2, horitzontal=TRUE)
# last.review
boxplot(dd[,17]~c2, horitzontal=TRUE)
# reviews.per.month
boxplot(dd[,18]~c2, horitzontal=TRUE)
# review.rate.number
boxplot(dd[,19]~c2, horitzontal=TRUE)
# calculated.host.listings.count"
boxplot(dd[,20]~c2, horitzontal=TRUE)
# availability.365
boxplot(dd[,21]~c2, horitzontal=TRUE)
```

```
pairs(dcon[,1:12], col=c2)

plot(lat,long,col=c2,main="Clustering of credit data in 3 classes")
legend("topright",levels(c2),pch=1,col=c(1:4), cex=0.6)
```

Profiling

```
#Read variables

dd <- read.csv("Airbnb_clean.csv",header=T, sep=",", stringsAsFactors=TRUE);
names(dd)
dim(dd)
summary(dd)

attach(dd)

#set a list of numerical variables
names(dd)

dcon <- data.frame
(lat,price,number.of.reviews,review.rate.number,long,service.fee,last.review,calculated.host.listings.count,Construction.year,minimu
m.nights,reviews.per.month,availability.365)
dim(dcon)

#
# QUICK CLUSTERING

library(cluster)

#dissimilarity matrix

actives<-c(2:16)
dissimMatrix <- daisy(dd[,actives], metric = "gower", stand=TRUE)

distMatrix<-dissimMatrix^2

h1 <- hclust(distMatrix,method="ward.D")  # NOTICE THE COST
```

```
plot(h1)

c2 <- cutree(h1,4)

#Dictamen                <- as.factor(Dictamen)
#levels(Dictamen) <- c(NA, "positiu","negatiu")

#Profyling

#Calcula els valor test de la variable Xnum per totes les modalitats del factor P
ValorTestXnum <- function(Xnum,P){
 #freq dis of fac
 nk <- as.vector(table(P));
 n <- sum(nk);
 #mitjanes x grups
 xk <- tapply(Xnum,P,mean);
 #valors test
 txk <- (xk-mean(Xnum))/(sd(Xnum)*sqrt((n-nk)/(n*nk)));
 #p-values
 pxk <- pt(txk,n-1,lower.tail=F);
 for(c in 1:length(levels(as.factor(P)))){if (pxk[c]>0.5){pxk[c]<-1-pxk[c]}}
 return (pxk)
}

ValorTestXquali <- function(P,Xquali){
 taula <- table(P,Xquali);
 n <- sum(taula);
 pk <- apply(taula,1,sum)/n;
 pj <- apply(taula,2,sum)/n;
 pf <- taula/(n*pk);
 pjm <- matrix(data=pj,nrow=dim(pf)[1],ncol=dim(pf)[2], byrow=TRUE);
 dpf <- pf - pjm;
 dvt <- sqrt(((1-pk)/(n*pk))%*%t(pj*(1-pj)));
 #i hi ha divisions iguals a 0 dona NA i no funciona
 zkj <- dpf
 zkj[dpf!=0]<-dpf[dpf!=0]/dvt[dpf!=0];
 pzkj <- pnorm(zkj,lower.tail=F);
 for(c in 1:length(levels(as.factor(P)))){for (s in 1:length(levels(Xquali))){if (pzkj[c,s]> 0.5){pzkj[c,s]<-1- pzkj[c,s]}}}
 return (list(rowpf=pf,vtest=zkj,pval=pzkj))
```

```
}

#source("file")
#dades contain the dataset
dades<-dd
#dades<-dd[filtro,]
#dades<-df
K<-dim(dades)[2]
par(ask=TRUE)

#P must contain the class variable
#P<-dd[,5]
P<-c2
#P<-dd[,18]
nameP<-"Classes"
#P<-df[,33]

nc<-length(levels(factor(P)))
nc
pvalk <- matrix(data=0,nrow=nc,ncol=K, dimnames=list(levels(P),names(dades)))
nameP<-"Class"
n<-dim(dades)[1]

for(k in 1:K){
 if (is.numeric(dades[,k])){
        print(paste("AnÃ lisi per classes de la Variable:", names(dades)[k]))

        boxplot(dades[,k]~P, main=paste("Boxplot of", names(dades)[k], "vs", nameP ), horizontal=TRUE)

        barplot(tapply(dades[[k]], P, mean),main=paste("Means of", names(dades)[k], "by", nameP ))
        abline(h=mean(dades[[k]]))
        legend(0,mean(dades[[k]]),"global mean",bty="n")
        print("EstadÃstics per groups:")
        for(s in levels(as.factor(P))) {print(summary(dades[P==s,k]))}
        o<-oneway.test(dades[,k]~P)
        print(paste("p-valueANOVA:", o$p.value))
        kw<-kruskal.test(dades[,k]~P)
        print(paste("p-value Kruskal-Wallis:", kw$p.value))
        pvalk[,k]<-ValorTestXnum(dades[,k], P)
```

```
print("p-values ValorsTest: ")
print(pvalk[,k])
}else{
if(class(dd[,k])=="Date"){
print(summary(dd[,k]))
print(sd(dd[,k]))
#decide breaks: weeks, months, quarters...
hist(dd[,k],breaks="weeks")
}else{
#qualitatives
print(paste("Variable", names(dades)[k]))
table<-table(P,dades[,k])
#   print("Cross-table")
#   print(table)
rowperc<-prop.table(table,1)

colperc<-prop.table(table,2)
#  print("Distribucions condicionades a files")
# print(rowperc)

#ojo porque si la variable es true o false la identifica amb el tipus Logical i
#aquest no te levels, por tanto, coertion preventiva

dades[,k]<-as.factor(dades[,k])

marg <- table(as.factor(P))/n
print(append("Categories=",levels(as.factor(dades[,k]))))

#from next plots, select one of them according to your practical case
plot(marg,type="l",ylim=c(0,1),main=paste("Prop. of pos & neg by",names(dades)[k]))
paleta<-rainbow(length(levels(dades[,k])))
for(c in 1:length(levels(dades[,k]))){lines(colperc[,c],col=paleta[c]) }

#with legend
plot(marg,type="l",ylim=c(0,1),main=paste("Prop. of pos & neg by",names(dades)[k]))
paleta<-rainbow(length(levels(dades[,k])))
for(c in 1:length(levels(dades[,k]))){lines(colperc[,c],col=paleta[c]) }
legend("topright", levels(dades[,k]), col=paleta, lty=2, cex=0.6)
```

```
#condicionades a classes
print(append("Categories=",levels(dades[,k])))
plot(marg,type="n",ylim=c(0,1),main=paste("Prop. of pos & neg by",names(dades)[k]))
paleta<-rainbow(length(levels(dades[,k])))
for(c in 1:length(levels(dades[,k]))){lines(rowperc[,c],col=paleta[c]) }

#with legend
plot(marg,type="n",ylim=c(0,1),main=paste("Prop. of pos & neg by",names(dades)[k]))
paleta<-rainbow(length(levels(dades[,k])))
for(c in 1:length(levels(dades[,k]))){lines(rowperc[,c],col=paleta[c]) }
legend("topright", levels(dades[,k]), col=paleta, lty=2, cex=0.6)

#amb variable en eix d'abcisses
marg <-table(dades[,k])/n
        print(append("Categories=",levels(dades[,k])))
        plot(marg,type="l",ylim=c(0,1),main=paste("Prop. of pos & neg by",names(dades)[k]), las=3)
        #x<-plot(marg,type="l",ylim=c(0,1),main=paste("Prop. of pos & neg by",names(dades)[k]), xaxt="n")
#text(x=x+.25, y=-1, adj=1, levels(CountryName), xpd=TRUE, srt=25, cex=0.7)
paleta<-rainbow(length(levels(as.factor(P))))
        for(c in 1:length(levels(as.factor(P)))){lines(rowperc[c,],col=paleta[c]) }

        #with legend
        plot(marg,type="l",ylim=c(0,1),main=paste("Prop. of pos & neg by",names(dades)[k]), las=3)
        for(c in 1:length(levels(as.factor(P)))){lines(rowperc[c,],col=paleta[c])}
        legend("topright", levels(as.factor(P)), col=paleta, lty=2, cex=0.6)

#condicionades a columna
plot(marg,type="n",ylim=c(0,1),main=paste("Prop. of pos & neg by",names(dades)[k]), las=3)
paleta<-rainbow(length(levels(as.factor(P))))
for(c in 1:length(levels(as.factor(P)))){lines(colperc[c,],col=paleta[c]) }

#with legend
plot(marg,type="n",ylim=c(0,1),main=paste("Prop. of pos & neg by",names(dades)[k]), las=3)
for(c in 1:length(levels(as.factor(P)))){lines(colperc[c,],col=paleta[c])}
legend("topright", levels(as.factor(P)), col=paleta, lty=2, cex=0.6)

        table<-table(dades[,k],P)
        print("Cross Table:")
        print(table)
```

```
		print("Distribucions condicionades a columnes:")
		print(colperc)

		#diagrames de barres apilades

		paleta<-rainbow(length(levels(dades[,k])))
		barplot(table(dades[,k], as.factor(P)), beside=FALSE,col=paleta )

		barplot(table(dades[,k], as.factor(P)), beside=FALSE,col=paleta )
		legend("topright",levels(as.factor(dades[,k])),pch=1,cex=0.5, col=paleta)

  #diagrames de barres adosades
		barplot(table(dades[,k], as.factor(P)), beside=TRUE,col=paleta )

		barplot(table(dades[,k], as.factor(P)), beside=TRUE,col=paleta)
		legend("topright",levels(as.factor(dades[,k])),pch=1,cex=0.5, col=paleta)

		print("Test Chi quadrat: ")
		print(chisq.test(dades[,k], as.factor(P)))

		print("valorsTest:")
		print( ValorTestXquali(P,dades[,k]))
  #calcular els pvalues de les quali
  }
		}
}#endfor

#descriptors de les classes mÃ©s significatius. Afegir info qualits
for (c in 1:length(levels(as.factor(P)))) {
 if(!is.na(levels(as.factor(P))[c])){
		print(paste("P.values per class:",levels(as.factor(P))[c]));
		print(sort(pvalk[c,]), digits=3)
 }
}

#afegir la informacio de les modalitats de les qualitatives a la llista de pvalues i fer ordenacio global

#saving the dataframe in an external file
#write.table(dd, file = "Airbnb_Clean.csv", sep = ";", na = "NA", dec = ".", row.names = FALSE, col.names = TRUE)
```