AD699 Semester Project: Airbnb Listings in Copacabana

Shimony Agrawal, Gerardo Bastidas, Alberto Calderon, Benjamin Flavin, Oscar Villarreal Rojas

Boston University

MET AD699: Data Mining for Business Analytics (Summer II Term)

Table of Contents

AD699 Semester Project: Airbnb Listings in Copacabana

Copacabana is one of Rio's liveliest neighbourhoods drawing locals and tourists from all over the world. Known for its crescent shaped-beach, lively streets and at-par dining options, Copacabana is one of the most well-known tourist destinations in Brazil. In close proximity is Ipanema, which attracts tourists for its beaches, hippie markets and activities like surfing. Being the most sought-after tourist destination, Copacabana thrives on its beautiful Airbnb properties that give its tourists a local experience.

To improve the customer experience in Copacabana, Airbnb is working with its data analytics team to better understand the performance of Airbnb rentals in terms of their prices, instant bookings as well as developing groups of similar rentals to reach out to more targeted customers. The Airbnb data has 8000+ observations for features like neighbourhood overview, amenities, number of bedrooms, price, cleaning fee, reviews scores based on accuracy, location, cleanliness, number of reviews, host response rate, superhost, instant bookable, verification, property type, room type and cancellation policy.

To conduct the data analysis using various prediction, classification and clustering models, the data was first prepared to remove all N/A values and variables not needed for analysis. Following this, data exploration was carried out to better understand the variables using visualizations. Lastly, the various models helped in assessing the performance of Airbnb rentals in Copacabana. The results and analysis can be used by Airbnb to further improve its listings.

Data Preparation and Exploration

Missing Values

The first thing we did to manage the data with which we were going to work was to fix the structure of the columns (columns to factors, remove the "%", "$" and "," from columns with numbers and convert them to numeric and columns to dates). After, we filtered the data to our neighbourhood "Copacabana". Once this was done, we went to see which columns were unnecessary from our point of view and we began to eliminate them one by one. At the time we finished the steps mentioned, we moved on to analyse what to do with the missing data. We detected that a lot of data was missing in some of the columns from the review score columns (about 35% of the total), so eliminating those rows would be representing deleting a lot of information that can be valuable (i.e., price data), and after analysing their summary statistics, we believed that the best way to fix those missing values was to complement those empty spaces with the “mode” of the columns, and it would not be representative for the whole sample. Next, we decided to use the “median” of the data in the remaining columns, since we saw that there was a lot of positive bias that caused the average to increase, so we believed that it would not be the correct thing to do, these data represented about 15%. At the end we were left with columns that contained between 0.01-0.02% of missing values, so we decided to eliminate those rows with the function in R “na.omit”. This is how we handled the data and created the data frame that helped us to work throughout this project.

Summary Statistics

For the summary statistics, we started by viewing the summary statistics for all the columns in our data frame. When we called the function, we realized that it was very difficult to draw conclusions because it was still a lot of information, so we decided to focus on a couple and

start making filters to be able to make comparisons. We discovered that the number of host_id was repeated in several rows, meaning that a single host could have 1 or more properties on the data frame, so we first wanted to see who were the hosts with the most properties in our data set, and create a data frame that would contain the 10 hosts with more properties to analyse their statistics and compare them against everything (Figure 1). Immediately, we decided to see what type of property you could find the most within our data frame, where we did 3 researches based on this (for only "Apartment", putting together "Condominium", "Loft" and "Serviced Apartment" and finally for only "House").

As shown in Figure 2, we first used the filter for “Apartment” and review the summary statistics for the number of reviews in the ltm and ratings reviews, and then compare by adding to the apartment filter, the filter for the 10 hosts with the most properties and review the summary statistics of the above inspected. Regarding the rating of review scores, the average went down from 96 in the data frame with all the rows to 93 where we filtered by the 10 hosts with the most properties, meaning that not necessarily those 10 hosts are having good comments, assuming that perhaps for having more properties, they don't go into the details as much to create a better user experience. The same conclusion applies to the part where we group "Condominium", "Loft" and "Serviced Apartment" where the average goes from 95 to 88 and where we analyse only House, where the average drops from 95 to 87 in the review score. Regarding the analysis of the number of reviews, we see that we have significant positive biases that do not allow us to draw an adequate conclusion regarding the comparison of media since if there is a lot of difference, except where we filter by “House”, we can see a similar range (in the whole vs 10 hosts with more properties) and we see a higher average where it is filtered by the

10 hosts, which could mean that more people who have been staying in the houses of those 10 hosts, for which they have received in average more comments in the last 12 months.

Data Visualization

Figure 3 shows the first visualisation depicting Number of Reviews by Type of Host, displays the leadership in the number of reviews of superhosts over normal hosts. The first quartile of superhosts is higher than the mean in the number of reviews of normal hosts.

Figure 4 illustrates the second graph showing that frequency per room type and superhost. It can be inferred that entire homes/apartments are the most popular type of room, followed by private rooms, where shared rooms and hotel rooms have just a slight presence in the offering. Also, in most cases, hosts are not superhosts as seen in the figure.

Figure 5 shows the Frequency per Property of the Airbnb rentals in Copacabana. As seen, apartments dominate the property types with more than 6,000 apartments being rented. The rest of the property types have little to no representation.

Figure 6 illustrates two histograms that show the price distribution. The first histogram includes all prices with a binwidth of 1,000, whereas the second histogram excludes outliers by filtering the price and showing just the ones smaller than 2,500, which allows for a smaller binwidth of 50, without compromising the clarity of the plot. It is visible that the filter helped to ignore outliers and put the focus in the prices with most representation.

Finally, figure 7 depicts a set of histograms displaying the price distribution by room type which makes it simple to see the longer right tail in prices of entire homes/apartments. A surprising finding is how the hotel price distribution ends around 1,250 and private room distribution ends at 700, while the entire home/apartments has records around 2,500. One would

think that the additional services, amenities and renown of certain hotels would reflect higher prices.

As an extra visualization shown in figure 8, we developed a word cloud to highlight the most required amenities by customers of Airbnb rentals in Copacabana, Brazil. This would help analyze the most prominent requirements and can be included by all Airbnb rentals across Copacabana. Some of the most useful amenities are friendly hosts, TV, Wi-Fi, parking, waterbed, dryer, silverware cooking, pillows, suitable for families and children, extra linens, smoking, blankets, microwaves, private workspace, allowing pets and books. It can be seen that friendly hosts top the list which can be used by Airbnb to better train its hosts with guidelines on interacting with their guests. A fully functioning kitchen is another requirement that hosts can add in their listings.

Mapping

The map shown in Figure 9 was generated to display a “heat map” of rental pricing throughout Copacabana. The intent was to highlight regions of more expensive listings and less expensive listings and to illustrate geographically some factors that may have gone into the pricing decisions landlords made. First, we can look at some of the pricier units on the market. Praia de Copacabana (Copacabana Beach) is a light blue dot indicating the most expensive price class located on the coastal side of Copacabana. Conventional wisdom dictates that waterfront property will always be expensive relative to its less-maritime neighbours, which is reflected in the plot using a gradient colour scheme (i.e. the shoreline is lighter blue than the inland neighbours). One can also see expensive real estate in the downtown theatre district and by the Parque Estadual da Chacrinha, the park in the northeast quadrant of the city. One additional stand-out property of note is located just north of the peninsula containing the Forte de

Copacabana (or the Copacabana Fort). This site, depicted on the map in light blue, is within easy striking distance of the Estátua de Carlos Drummond de Andrade, a local landmark. If a potential guest was looking for a cheaper listing, they might try the southwest quadrant of Copacabana bordering Ipanema where the gradient turns dark blue; a "value" indicator on this map.

Word Cloud

The word cloud show in Figure 10 highlights the neighbourhood overview that has been provided by the customers. The overview has short descriptions of the experiences of the users and what they liked about the Airbnb rental. The most prominent words used are: Copacabana, praia (beach), Ipanema, restaurantes (restaurants), bares (bars), famous, beautiful, perto (near) etc. Copacabana being a tourist destination in Rio de Janeiro, the most frequent words are beautiful, beaches, restaurants and bars showing that the tourists come to Copacabana for its beaches, restaurants and bars and its night life. Another word that is prominent is Ipanema, a beach near Copacabana which indicates that tourists coming to Copacabana also visit Ipanema.

Prediction

Multiple Linear Regression

For the second step of the final project, we developed a multiple regression model to predict the price of an Airbnb with different characteristics. The variables that we choose are of different types, numeric variables, and dummies variables. The numeric variables are host response rate, latitude and longitude, accommodates, bathrooms, bedrooms, beds, price, guests included, extra people, minimum nights, maximum nights, number of reviews, review scores rating, review scores cleanliness, review scores communication, review scores location and

review scores value. On the other hand, our dummies variables are host is super host, room type, bed type, and cancellation policy.

We use the alpha.csv dataset to select the previous variables, ending with a database of 8415 observations. Then, we decided to analyze the distribution of the price, where we found that there were many outliers, which could affect the predictions of the model; therefore, we filter our dataset to keep the observations where the price is equal or lower to 2,500 dollars. By filtering the dataset, we reduce the number of observations to 8265. Next, we divided this final dataset into two different databases, a training set with 60% of the observations, and a validation set with the remaining 40%.

Once we have our training and validation set, we proceed to calculate the correlation of all the numeric variables of our database. By analyzing each variable's relationship, we notice that "accommodates" has a correlation higher than 0.70 with bedrooms and beds as shown in Figure 11. We decide to remove "accommodates" and create a multiple linear regression model with the remaining variables, which shows the results that we can see in Figure 12.

Then, by looking at the p-value of each variable, we noticed that many variables were insignificant for our model's purpose. Therefore, we carried out a backward elimination process to eliminate the variables that were not contributing any value to the model.

Finally, we create our final model with the remaining variables of the backward elimination process. These variables are host response rate, host is super host, latitude and longitude, room type, bathrooms, bedrooms, beds, guests included, extra people, number of reviews, review scores rating, review scores cleanliness, review scores location, and cancellation policy. The results of the model can be seen in Figure 13. Here we can see that the equation of our model is:

new variable = 3.13 - 6.35(xHost response rate) - 5.93 (xHost is super host) - 3.5 (xLatitude) + 2.59 (xLongitude) - 2.11 (xRoom type hotel) - 1.70 (xRoom type private) - 1.33 (xRoom type shared) + 1.11 (xBathrooms) + 1.32 (xBedrooms) - 7.72 (xBeds) - 2.27 (xGuest included) + 3.39 (xExtra people) - 1.25 (xNumber of reviews) + 5.23 (xReview rating) + 1.57 (xReview cleanliness) - 2.08 (xReview location) - 1.24 (xCancellation moderate) - 1.05 (xCancellation strict) + 1.04 (xCancellation super strict 30) + 2.53 (xCancellation super strict 60).

If we want to know the price of a new Airbnb, we must substitute the x values with the data of our Airbnb. One of the things that we noticed was that the r squared predicted only 35% of the observations. Although we tried to increase it by using different variables, we failed. Consequently, we think that if we want to improve the r squared of the model, we must gather information about new variables that could contribute to the model, like the square feet of the property.

Lastly, we calculated the accuracy of our model compared to the training and validating dataset, where we got an RMSE of 316 and 287, respectively. Despite the fact that the errors from both databases are pretty similar, the model was not doing a good job predicting the sales of new variables.

Classification

K-Nearest Neighbors

Using k-nearest neighbours model shown in Figure 14, we predicted the type of cancellation that a rental unit would have in our Copacabana neighbourhood. The first thing we did for this part was to remove all the columns that were not numeric, leaving only as non-numeric our variable that we were going to predict (cancellation policy). Once we were done, we

divided our data frame in 2 (validation and training), setting a seed and then partitioning the data 60% -40% with the sample_n and slice functions. Later, we normalize our data to be able to test and obtain our most accurate k of the model. Using all the variables from our data frame, we obtained that our most accurate k was 22, with an accuracy of 57.04%. It is worth mentioning that we did many tests with different variables in our data frame and the model was not able to identify a better result for the k-nearest neighbours test, so we concluded that 22-k with 57% precision, using all the variables, was how we would be able to get the most precise type of cancellation policy that will have for a certain rental unit in Copacabana.

We carry out a quick test by creating a fictitious place and through the runif function, making a data frame to test what result we would obtain in terms of the type of location cancellation policy, obtaining, with 57% of accuracy,"Strict_14_with_grace_period" as response with k = 22.

Naïve Bayes

Using Naive Bayes model, we predicted the probabilities that an Airbnb rental in Copacabana was instantly bookable or not. Since the Bayesian classifier works only with categorical predictors, we converted our chosen variables into factors by binning them. The variables we chose for the model were: number of bedrooms, reviews based on cleanliness and location, property type, room type, if the host was superhost or not, requires guest phone verification and their cancellation policy. We selected these variables on the basis that anyone who books an Airbnb property will go through these stages to select the rental. For instance, bedrooms are the first filter people would use on the Airbnb website followed by the reviews given for each rental. Additionally, property type and room type offer the user more choices to the users. The other 2 variables of whether the host is superhost and guest verification give the

users the idea that the property is keeping in mind the authenticity of the host and the safety of its guests. Lastly, cancellation policy is another important factor for users. The more flexible the policy, the more likely people would book that rental. Given the uncertain times like these, having a flexible cancellation policy is an added benefit. Based on the factors like user convenience, safety and credibility of the host we shortlisted the above-mentioned variables.

Figure 15 shows the Naive Bayes model used to predict whether the rental in Copacabana is instantly bookable. As seen, the a-priori probability of the rental being instantly bookable is 39% as compared to the rental not being bookable i.e. 60%. This can be due to a variety of reasons like poor guest reviews, strict cancellation policies or bad experiences with the host. Figure 16 shows the fictional rental property we created based on our variables. We assessed the probability of the property being instantly bookable by 2 methods. Firstly, we used the predict function on the current model as shown in Figure 16.1. Our outcome was that the fictional rental was not bookable instantly. The model predicted that there was 55% chance of it not being bookable versus a 44% chance of it being bookable. Secondly, we decided to test the model on our using the conditional probabilities we found and it yielded similar results - 54% chance of not being bookable and 45% chance of being bookable as shown in Figure 16.2.We also tested the training and validation sets to assess the accuracies of the model as well as the confusion matrix. Figure 17 shows the outcome of the confusion matrix. As seen, the training set has an accuracy of 64% while the validation set has an accuracy of 61% which indicates an average performance.

Based on the results obtained, it can be concluded that Airbnb rentals in Copacabana are not instantly bookable. By introducing new features and tools for its hosts, Airbnb can work on increasing its probability of being instantly bookable. Airbnb has a feature of Instant Book that

allows guests to directly book the listing without waiting for an approval from the host. For the hosts, by using the Instant feature they increase their visibility and achieve a status of superhost as well as cancelling a listing if the guest is not trustworthy. Given the higher rate of not being able to instantly book, Airbnb has introduced new features and tools like a more detailed calendar for hosts to keep track of their listings and adding a 1-day period for instant bookings.

Classification Tree

The development of a classification tree with a decent accuracy was quite a challenge. After several trials using diverse variables that made sense to us, the accuracy rate was around 35 to 40 percent. As a team, we decided to do the process again, but with a regression model as a start point. In that sense, the first step was to determine which variables are strongly correlated to the output variable: cleaning fee. The first regression model included all variables of the "alpha" data frame, and from there backwards elimination was applied until the end result was an optimal model with the key variables for the classification tree. However, even though the model included the most influential variables with a p-value smaller than 0.05, the R-2 was quite weak with 0.2582. Such a weak R-2 is an early indicator that the classification tree accuracy might be weak as well. It was surprising to discover that the variables had a weak effect among themselves.

According to the regression model, the variables that have an influence in the cleaning fee are: Price, Host Identity Verified, Property type, Room type, Bathrooms, Bedrooms, Security deposit, Guests included, Extra people, Minimum nights, Review scores value, Cancellation policy, Require guest profile picture, Require guest phone verification, Reviews per month.

After having the key variables with the most influence, the classification tree was developed. For this, the cleaning fee was divided in four breaks: Low, Average, Expensive, Very

Expensive. The data frame was divided, as well, into two sets: Training and Validation. The first model for the classification tree used the training set and a complexity parameter (cp) of 0. It gave an accuracy of 71.12% against the training set and of 57.08% against the validation set. From the model shown in Figure 18, the complexity parameter was further analysed to identify the best cp and ideal size. The cp of 0.00164880 provided the best option as it has the lowest Xerror, but the number of splits is quite high with 25. The cp of 0.00288541 provides a clearer decision tree, 9 splits, at a similar Xerror. After changing the cp to 0.00164880, the accuracy of the validation set went slightly higher with 58.39% accuracy.

Clustering

In creating custom variables for the k-means cluster, an analyst might consider the perspective on a potential guest and the attributes that they would value in a rental. These could include amenities, proximity to local attractions, “bang for the buck”, etc. We opted to use the latter to cluster rentals by cost given the maximum occupancy level of the rental. By such a method, for example, one might find the highest occupancy rentals for the cheapest price. In order to create these custom variables, we had to consider which variables in the dataset contributed to cost and how “maxing out” the occupancy of each rental might impact price. To create the “Max. Cost” variable, we had to multiply the base rate by the minimum amount of nights a guest can stay in order to establish a baseline cost. We then added the cost to “max out” the occupancy of the rental which involved subtracting the guests included in the base price from the total occupancy (the “accommodates” variable) and multiplying the remaining guests not covered under the base price by the cost per additional guest (“the extra_people” variable). Finally, we added the security deposit and cleaning fees to the above to get the sum cost for full occupancy for a given rental. The “Max. Guests variable was a much simpler calculation; after

checking to make the sure “accommodates” variable (a.k.a. The maximum occupancy) was roughly twice that of the count of beds per rental (the logic being you could theoretically sleep two to a bed) we simply assigned the variable “accommodates” to “Max. Guests.” Had “accommodates” been lower than two times the bed count, then we would have rewritten the calculation for “Max. Guests” as twice the count of beds.

After plotting the results (Figure 19-20) and trying several different cluster values, we decided it was most logical (and easiest to read) when there were four distinct clusters. This enabled us to define clusters according to our hypothetical guest question regarding maximum occupancy and relative cost. The “Best Value” cluster included rentals that were of average cost but greater to significantly greater maximum occupancy. This cluster would appeal to guests wanting to know “how do me and all my friends find the cheapest possible place to stay under one roof?” By contrast, the “Overpriced” cluster sleeps anywhere from an under-average to over-average amount of guests but at a significantly greater cost. There may be other reasons a guest may want to stay in these units, but from a purely cost-per-head standpoint these units would be considered too expensive. The “Average Value” cluster is centered around the mean in terms of occupancy and price, hence the name. Finally, “Crash Pad” includes units that are of average to slightly over-average price but sleep fewer than average guests. One might stay somewhere in the “Crash Pad” cluster if they were traveling alone or in a very small group and did not feel like spending a high premium for their stay.

Conclusion

Data cleansing and preparation is a really important step because it improves your data quality and in doing so, increases overall productivity. Summary statistics can be very helpful at

the beginning to have a more accurate perception of what we are dealing with, and when the sets are too big we think that the best approach is as we did it, digging in a couple of variables to find out interesting things. Data Visualisation further helps in identify relationships between variables for further analysis. For instance, we found the most popular property types and room types which helped us in filtering those features to get relevant results. Moreover, upon visualising the price distribution we identified various outliers which we then filter in the models later. The price distribution by room type gave us insight into price distribution for “Entire home/Apt” goes till $2,500 indicating profitable room-type.

To guests and owners alike, a price map of Copacabana can be a useful tool. Guests may want to know about the value they are getting when securing a given rental. This could include the price compared to neighbouring units, the proximity of the unit to local landmarks or attractions, or simply the average price of the local market in general as this may influence a guest’s decision to confirm travel plans or look at another market. The price map is exactly the type of resource that would help Airbnb guests in such an endeavour. Indeed, as both precedence and proof-of-concept, if one were to look at travel sites like TripAdvisor, Travelocity, Trivago, Expedia, or any other leading travel site for that matter, one can view an interactive map of available units that often includes price. This map would also be beneficial to unit owners looking to ensure their pricing is competitive in the local market.

K-NN is a very interesting model since it let us know what the best classification of a certain variable is, for this project, we didn’t obtain a very accurate K (around 57%) but is enough to get a result, and conclude that we will have to collect more data in order to increase that accuracy.

According to the Naïve Bayes classifier, the Copacabana Airbnb rentals are not instantly bookable. Actually, “Instant Book” is a feature Airbnb is testing out since 2015. As per Airbnb, 60% of its bookings are now being done via Instant Book. It is still improving the feature by adding new tools like updated calendars, 1-day booking times for hosts to approve the guests and addressing safety concerns for the guests. Additionally, statistics show that 52% more with Instant Book while improvising more features to qualify as superhost like maximum and minimum nights, number of bedrooms and increasing credibility of their listings through ID verification. While numbers are still low, with improvisation and better strategies, Instant Book will be a success for Airbnb.

The K-means clustering, conducted here to generate four distinct clusters of rentals, will really begin to demonstrate its value when it comes time to push rental suggestions to particular market segments of Airbnb guests. After collaborative filtering helps us determine the desirable rental attributes for a given customer or customer segment (using either a user-based approach by gathering data on new guests or item-based approach by looking at what types of rentals a guest has viewed or rented before) we would use these clusters to help make a match between guest and rental. The clusters could also be helpful in an active-search capacity for Airbnb’s user interface. For instance, there could be an FAQ page that includes the prompt “I’m looking for a place to host my 30th birthday with friends,” which would naturally suggest rentals in the “best value” cluster (i.e. high capacity, low price-per-head).

References

Ting, D. (2017, June 14). Airbnb Ramps Up Push to Get More Hosts to Choose Instant Booking. Retrieved August 11, 2020, from https://skift.com/2017/06/14/airbnb-ramps-up-push-to-get-more-hosts-to-choose-instant-booking/

Should You Use Instant Book on Airbnb? The Data Says Yes! (2019, October 23). Retrieved August 11, 2020, from https://www.airdna.co/blog/airbnb-instant-book-or-not

What Is Airbnb Instant Book and How to Use It Effectively. (2020, May 26). Retrieved August 11, 2020, from https://www.igms.com/what-is-instant-book-on-airbnb/

Figures

	host_id	n
1	91654021	93
2	81876389	48
3	13580277	45
4	224192	43
5	31275569	42
6	66877715	41
7	74463624	41
8	1500426	25
9	1381764	24
10	2915201	24
11	4255118	22
12	66039372	22
13	3413189	20
14	13664216	20

	property_type	n
1	Apartment	7557
2	Condominium	253
3	Loft	196
4	Serviced apartment	120
5	House	82
6	Aparthotel	35
7	Guest suite	31
8	Bed and breakfast	25
9	Guesthouse	25
10	Hotel	24
11	Hostel	21
12	Other	14
13	Boutique hotel	11

Figure 1: Hosts with the maximum properties

```
> #Apartments
>
> #Number of Reviews (LTM)
> summary(df.ss3$number_of_reviews_ltm)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  0.000   0.000   1.000   5.145   5.000 104.000
> summary(df.ss3.1$number_of_reviews_ltm)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
   0.00    0.00    0.00    2.22    3.00   35.00
> #Review Scores Rating
> summary(df.ss3$review_scores_rating)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  20.00   95.00  100.00   96.19  100.00  100.00
> summary(df.ss3.1$review_scores_rating)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  20.00   90.00  100.00   93.67  100.00  100.00
```

```
> #Condominium, Loft & Serviced Apartment
>
> #Number of Reviews (LTM)
> summary(df.ss4$number_of_reviews_ltm)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  0.000   0.000   2.000   7.279  10.000 127.000
> summary(df.ss4.1$number_of_reviews_ltm)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  0.000   0.000   0.000   1.091   1.000  10.000
> #Review Scores Rating
> summary(df.ss4$review_scores_rating)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  20.00   95.00  100.00   95.62  100.00  100.00
> summary(df.ss4.1$review_scores_rating)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  20.00   80.00  100.00   88.18  100.00  100.00
```

```
> #Houses
>
> #Number of Reviews (LTM)
> summary(df.ss5$number_of_reviews_ltm)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  0.000   0.000   0.000   1.402   2.000  14.000
> summary(df.ss5.1$number_of_reviews_ltm)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  0.000   0.000   1.000   2.214   3.000  12.000
> #Review Scores Rating
> summary(df.ss5$review_scores_rating)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  20.00   98.50  100.00   95.78  100.00  100.00
> summary(df.ss5.1$review_scores_rating)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  20.00   83.75   98.00   87.07  100.00  100.00
```

Figure 2: Summary Statistics

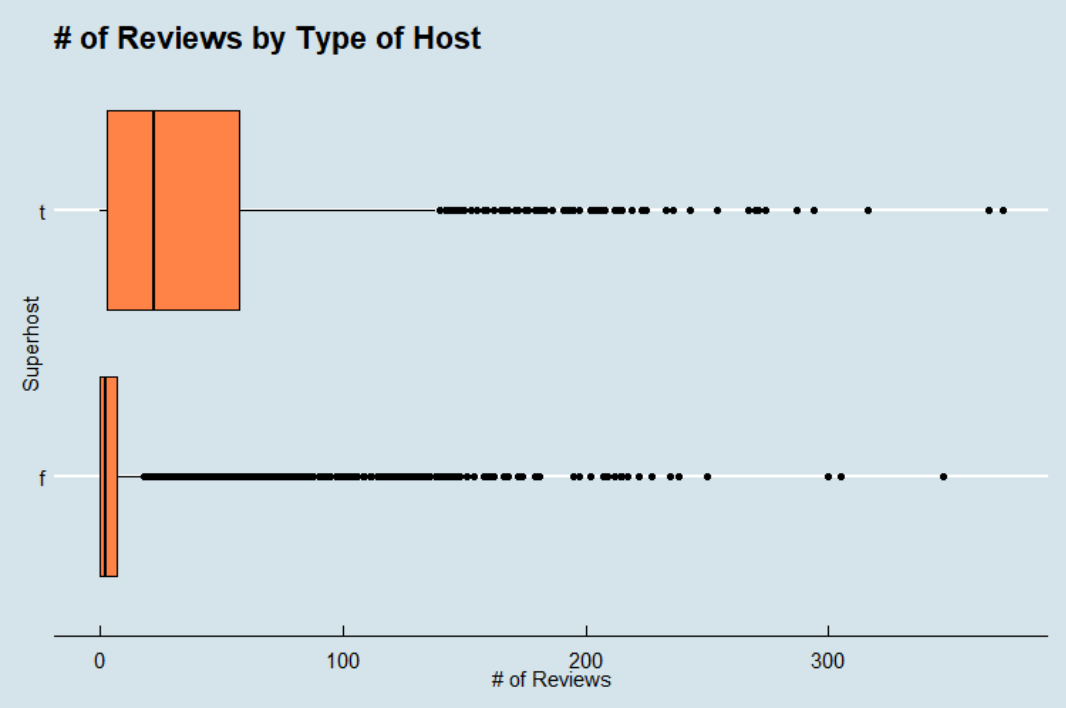

Figure 3: Number of Reviews by Type of Host

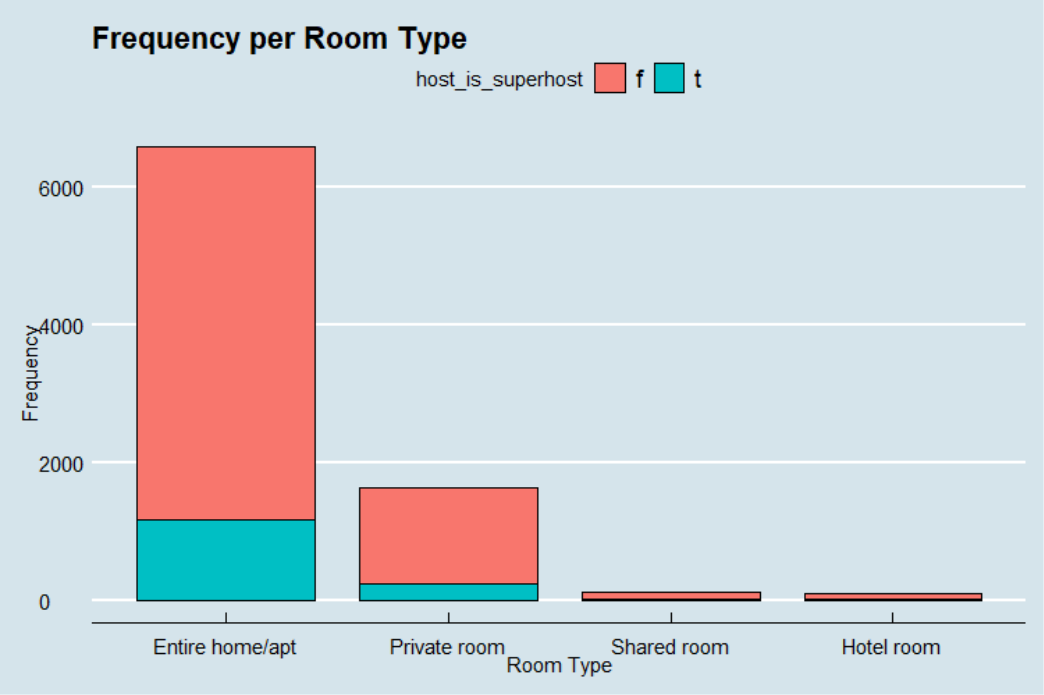

Figure 4: Frequency per Room Type and Superhost

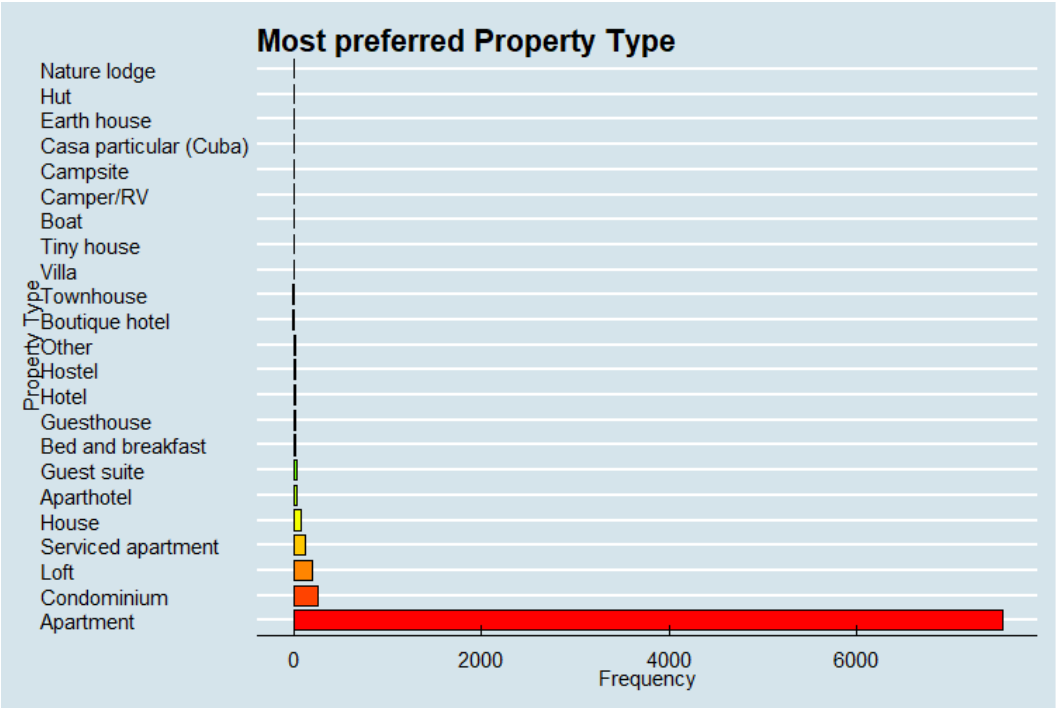

Figure 5: Frequency per Property Type

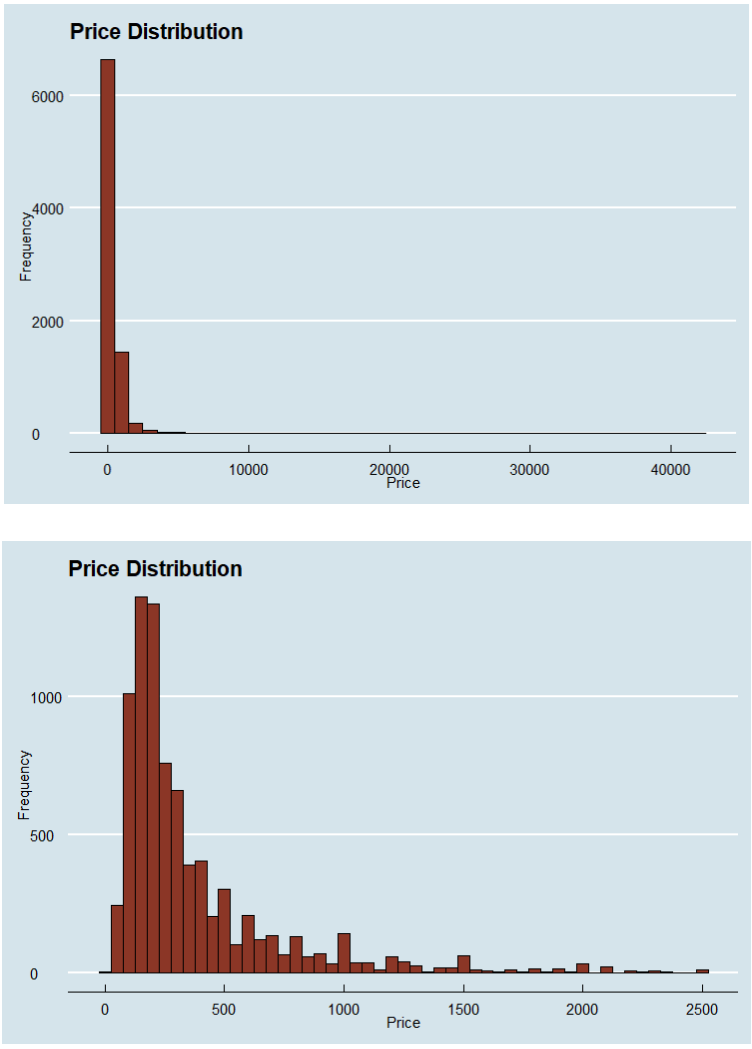

Figure 6: Price Distribution

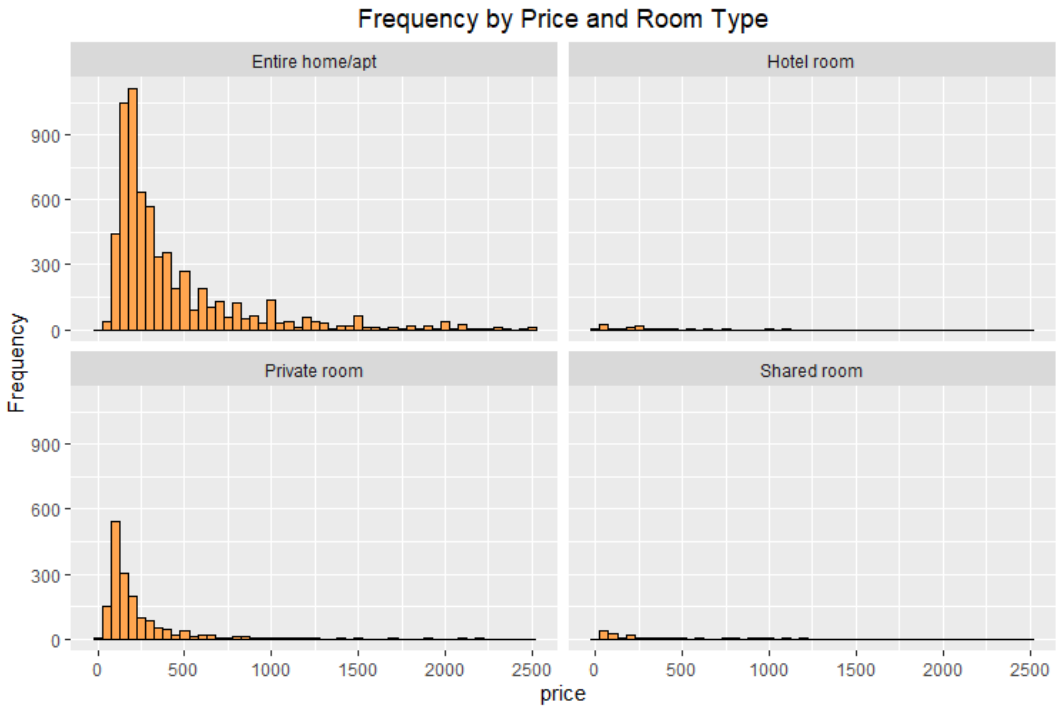

Figure 7: Price Distribution by Room Type

Figure 8: Word Cloud highlighting most required amenities by customers of Airbnb

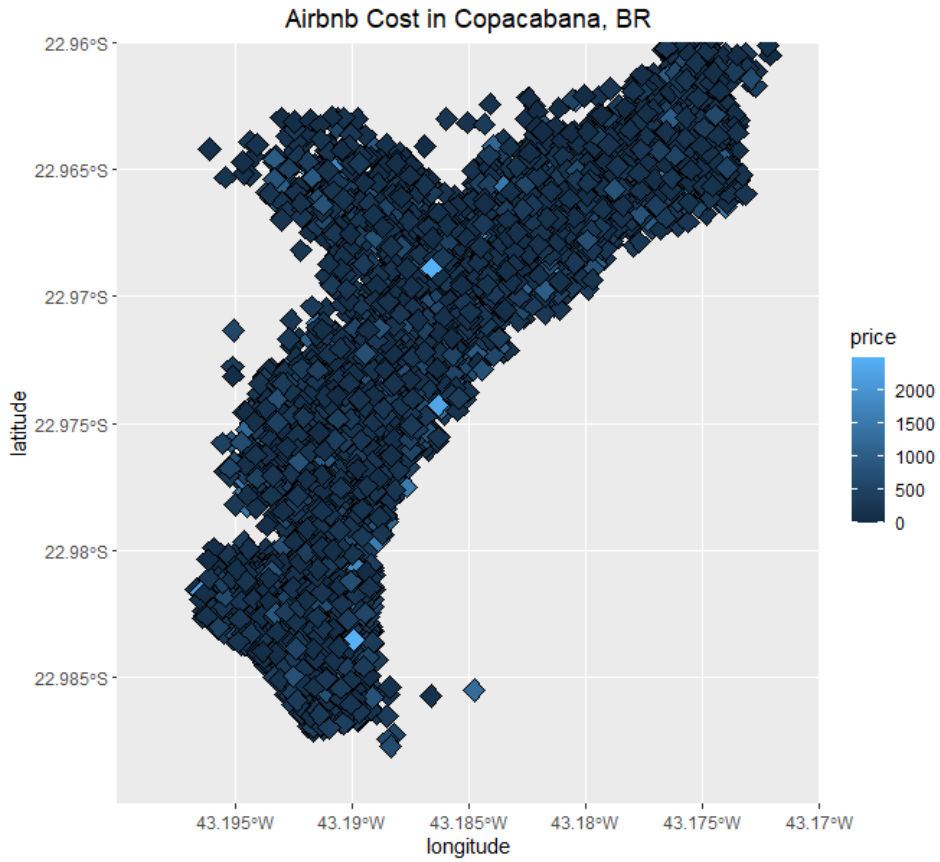

Figure 9: Airbnb Cost in Copacabana

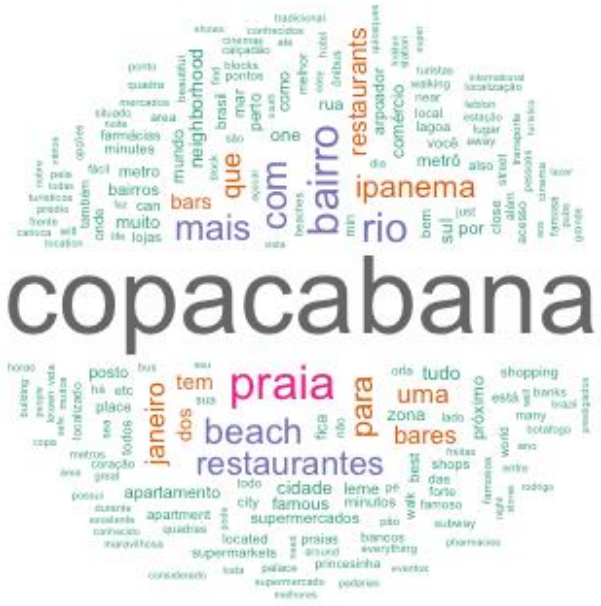

Figure 10: Word Cloud highlighting the neighborhood overview

	host_response_rate	latitude	longitude	accommodates	bathrooms	bedrooms	beds	guests_included	extra_people	minimum_nights	maximum_nights
host_response_rate	1.000000000	-0.040057746	0.002476904	-0.025080120	-0.033714790	-0.043423868	-0.03486225	0.064736886	-0.019495363	-0.0116658366	-0.006658391
latitude	-0.040057746	1.000000000	0.793532375	-0.026104708	-0.075422183	-0.061201268	-0.02144745	-0.056428894	0.002858293	-0.0165322951	0.016557768
longitude	0.002476904	0.793532375	1.000000000	0.016129772	-0.064639174	-0.052121912	0.01870066	-0.013409380	-0.020012749	-0.0199321152	0.029007112
accommodates	-0.025080120	-0.026104708	0.016129772	1.000000000	0.582166451	0.731826600	0.78617700	0.463822530	0.055565748	0.0185116373	0.036515908
bathrooms	-0.033714790	-0.075422183	-0.064639174	0.582166451	1.000000000	0.714561940	0.52751505	0.265023970	0.053859853	0.0248982226	0.018724871
bedrooms	-0.043423868	-0.061201268	-0.052121912	0.731826600	0.714561940	1.000000000	0.63686704	0.351744790	0.073778734	0.0204972072	0.022302706
beds	-0.034862250	-0.021447455	0.018700656	0.786176999	0.527515045	0.636867041	1.00000000	0.384095366	0.031773279	0.0260510039	-0.011967207
guests_included	0.064736886	-0.056428894	-0.013409380	0.463822530	0.265023970	0.351744790	0.38409537	1.000000000	0.230350641	-0.0042598104	-0.008597671
extra_people	-0.019495363	0.002858293	-0.020012749	0.055565748	0.053859853	0.073778734	0.03177328	0.230350641	1.000000000	0.0140471434	-0.006974331
minimum_nights	-0.011665837	-0.016532295	-0.019932115	0.018511637	0.024898223	0.020497207	0.02605100	-0.004259810	0.014047143	1.0000000000	-0.001136906
maximum_nights	-0.006658391	0.016557768	0.029007112	0.036515908	0.018724871	0.022302706	-0.01196721	-0.008597671	-0.006974331	-0.0011369056	1.000000000
number_of_reviews	0.188589234	-0.032158324	0.012877235	-0.053805342	-0.097131176	-0.102813839	-0.03549149	0.102325292	0.008083992	-0.0376312481	-0.007940194
review_scores_rating	-0.038656262	0.046560192	0.005147343	-0.030067334	0.024270299	0.031229628	-0.02907705	-0.075900097	-0.009701388	-0.0010308164	0.008277615
review_scores_cleanliness	-0.011840298	0.007184925	-0.020884102	-0.034665336	0.017323658	0.009807777	-0.02994954	-0.067366467	-0.010306237	0.0040477493	0.006848315
review_scores_communication	0.040281139	0.003737505	0.001642066	-0.006538447	0.006681294	0.003700688	-0.00849447	-0.024371853	-0.008115852	-0.0018097236	0.004119111
review_scores_location	-0.001299498	0.003082369	0.029092130	-0.018796193	-0.008398251	0.003298267	-0.01158413	-0.031875396	-0.040139471	0.0006218667	0.003663708
review_scores_value	-0.034732675	0.026090354	0.004068310	-0.045621972	0.008093094	0.008301466	-0.03338418	-0.092859988	-0.013728460	0.0108931248	0.008568366

	number_of_reviews	review_scores_rating	review_scores_cleanliness	review_scores_communication	review_scores_location	review_scores_value
host_response_rate	0.188589234	-0.038656262	-0.011840298	0.040281139	-0.0012994983	-0.034732675
latitude	-0.032158324	0.046560192	0.007184925	0.003737505	0.0030823695	0.026090354
longitude	0.012877235	0.005147343	-0.020884102	0.001642066	0.0290921299	0.004068310
accommodates	-0.053805342	-0.030067334	-0.034665336	-0.006538447	-0.0187961927	-0.045621972
bathrooms	-0.097131176	0.024270299	0.017323658	0.006681294	-0.0083982508	0.008093094
bedrooms	-0.102813839	0.031229628	0.009807777	0.003700688	0.0032982665	0.008301466
beds	-0.035491491	-0.029077052	-0.029949545	-0.008494470	-0.0115841309	-0.033384177
guests_included	0.102325292	-0.075900097	-0.067366467	-0.024371853	-0.0318753965	-0.092859988
extra_people	0.008083992	-0.009701388	-0.010306237	-0.008115852	-0.0401394715	-0.013728460
minimum_nights	-0.037631248	-0.001030816	0.004047749	-0.001809724	0.0006218667	0.010893125
maximum_nights	-0.007940194	0.008277615	0.006848315	0.004119111	0.0036637084	0.008568366
number_of_reviews	1.000000000	-0.053169021	-0.030343067	0.032809976	0.0539746249	-0.006233028
review_scores_rating	-0.053169021	1.000000000	0.688017333	0.583230505	0.4263924297	0.710013683
review_scores_cleanliness	-0.030343067	0.688017333	1.000000000	0.423588956	0.3127858066	0.604378987
review_scores_communication	0.032809976	0.583230505	0.423588956	1.000000000	0.3451110915	0.446586891
review_scores_location	0.053974625	0.426392430	0.312785807	0.345111092	1.0000000000	0.444982229
review_scores_value	-0.006233028	0.710013683	0.604378987	0.446586891	0.4449822289	1.000000000

Figure 11: Correlations among variables for Multiple Linear Regression

```
Coefficients:
                                                Estimate Std. Error t value Pr(>|t|)
(Intercept)                                    3.088e+04  3.850e+04   0.802 0.422441
host_response_rate                            -6.535e+03  2.594e+03  -2.519 0.011801 *
host_is_superhostTRUE                         -5.966e+01  1.331e+01  -4.483 7.53e-06 ***
latitude                                      -3.468e+03  1.034e+03  -3.353 0.000805 ***
longitude                                      2.566e+03  1.266e+03   2.028 0.042663 *
room_typeHotel room                           -2.111e+02  5.672e+01  -3.722 0.000200 ***
room_typePrivate room                         -1.707e+02  1.219e+01 -14.002  < 2e-16 ***
room_typeShared room                          -1.311e+02  4.980e+01  -2.633 0.008480 **
bathrooms                                      1.109e+02  8.545e+00  12.983  < 2e-16 ***
bedrooms                                       1.327e+02  7.611e+00  17.433  < 2e-16 ***
beds                                          -7.849e+00  3.322e+00  -2.363 0.018179 *
bed_typeCouch                                  1.168e+01  2.607e+02   0.045 0.964260
bed_typeFuton                                  1.479e+02  1.659e+02   0.892 0.372669
bed_typePull-out Sofa                          7.587e+01  1.383e+02   0.549 0.583368
bed_typeReal Bed                               6.751e+01  1.321e+02   0.511 0.609222
guests_included                               -2.273e+01  3.200e+00  -7.103 1.39e-12 ***
extra_people                                   3.394e-01  4.952e-02   6.854 8.06e-12 ***
minimum_nights                                 4.231e-02  2.122e-01   0.199 0.841939
maximum_nights                                -1.153e-06  3.182e-05  -0.036 0.971090
number_of_reviews                             -1.262e+00  1.301e-01  -9.703  < 2e-16 ***
review_scores_rating                           4.545e+00  1.067e+00   4.260 2.08e-05 ***
review_scores_cleanliness                      1.469e+01  7.530e+00   1.951 0.051070 .
review_scores_communication                    1.326e+01  9.653e+00   1.374 0.169572
review_scores_location                        -2.346e+01  1.153e+01  -2.034 0.042027 *
review_scores_value                            2.852e+00  8.162e+00   0.349 0.726789
cancellation_policymoderate                   -1.256e+02  1.462e+01  -8.588  < 2e-16 ***
cancellation_policystrict_14_with_grace_period -1.064e+02  1.162e+01  -9.158  < 2e-16 ***
cancellation_policysuper_strict_30             1.107e+01  5.983e+01   0.185 0.853292
cancellation_policysuper_strict_60             2.570e+01  3.925e+01   0.655 0.512650
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 317.6 on 4930 degrees of freedom
Multiple R-squared:  0.3584,    Adjusted R-squared:  0.3548
F-statistic: 98.35 on 28 and 4930 DF,  p-value: < 2.2e-16
```

Figure 12: Feature selection for Multiple Linear Regression

```
Coefficients:
                                                Estimate Std. Error t value Pr(>|t|)
(Intercept)                                    3.134e+04  3.841e+04   0.816 0.414647
host_response_rate                            -6.350e+03  2.588e+03  -2.454 0.014153 *
host_is_superhostTRUE                         -5.936e+01  1.328e+01  -4.471 7.97e-06 ***
latitude                                      -3.503e+03  1.032e+03  -3.393 0.000698 ***
longitude                                      2.592e+03  1.263e+03   2.052 0.040197 *
room_typeHotel room                           -2.113e+02  5.668e+01  -3.729 0.000194 ***
room_typePrivate room                         -1.709e+02  1.217e+01 -14.037  < 2e-16 ***
room_typeShared room                          -1.334e+02  4.934e+01  -2.704 0.006866 **
bathrooms                                      1.111e+02  8.539e+00  13.008  < 2e-16 ***
bedrooms                                       1.322e+02  7.589e+00  17.421  < 2e-16 ***
beds                                          -7.728e+00  3.314e+00  -2.332 0.019746 *
guests_included                               -2.277e+01  3.195e+00  -7.129 1.16e-12 ***
extra_people                                   3.397e-01  4.947e-02   6.867 7.36e-12 ***
number_of_reviews                             -1.256e+00  1.299e-01  -9.667  < 2e-16 ***
review_scores_rating                           5.237e+00  8.894e-01   5.889 4.15e-09 ***
review_scores_cleanliness                      1.576e+01  7.330e+00   2.150 0.031600 *
review_scores_location                        -2.082e+01  1.117e+01  -1.864 0.062349 .
cancellation_policymoderate                   -1.249e+02  1.459e+01  -8.564  < 2e-16 ***
cancellation_policystrict_14_with_grace_period -1.058e+02  1.158e+01  -9.135  < 2e-16 ***
cancellation_policysuper_strict_30             1.045e+01  5.980e+01   0.175 0.861242
cancellation_policysuper_strict_60             2.532e+01  3.922e+01   0.645 0.518677
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 317.5 on 4938 degrees of freedom
Multiple R-squared:  0.358,     Adjusted R-squared:  0.3554
F-statistic: 137.7 on 20 and 4938 DF,  p-value: < 2.2e-16
```

Figure 13: Final Multiple Regression Model

```
#STEP III. CLASSIFICATION - KNN

str(alpha)
names(alpha)
dfclass<- alpha[-c(2,3,5:7,10:12,17,34,36,37)]
str(dfclass)


#nrow(dfclass)
set.seed(699)
samplerclass <- sample_n(dfclass, 8415)
train.df.class <- slice(samplerclass, 1:5049)
valid.df.class <- slice(samplerclass, 5050:8415)

norm.values<-preProcess(train.df.class, method = c("center", "scale"))
train.norm.df<-predict(norm.values, train.df.class)
valid.norm.df<-predict(norm.values, valid.df.class)

names(train.norm.df)
#cancellation_policy [25]

accuracy<-data.frame(k = seq(1, 30, 1), accuracy = rep(0,30))
for (i in 1:30) {
  knn.pred<-knn(train.norm.df[,c(1:24,26)], valid.norm.df[,c(1:24,26)],
                cl = train.norm.df$cancellation_policy, k = i)
  accuracy[i, 2]<-confusionMatrix(knn.pred,valid.norm.df$cancellation_policy)$overall[1]
}
accuracy

#K=22
```

Figure 14: K-Nearest Neighbors Model

```
Naive Bayes Classifier for Discrete Predictors

Call:
naiveBayes.default(x = X, y = Y, laplace = laplace)

A-priori probabilities:
Y
        f         t
0.6091286 0.3908714

Conditional probabilities:
   bedrooms
Y            0           1           2           3           4           5           6           7           9
  f 0.092643052 0.508174387 0.219346049 0.144414169 0.023160763 0.008174387 0.000000000 0.002724796 0.001362398
  t 0.080679406 0.581740977 0.199575372 0.112526539 0.023354565 0.002123142 0.000000000 0.000000000 0.000000000

   review_scores_location
Y            2           4           5           6           7           8           9          10
  f 0.002724796 0.000000000 0.000000000 0.000000000 0.000000000 0.014986376 0.049046322 0.933242507
  t 0.000000000 0.000000000 0.002123142 0.000000000 0.002123142 0.014861996 0.053078556 0.927813163

   review_scores_cleanliness
Y            2           4           5           6           7           8           9          10
  f 0.002724796 0.002724796 0.000000000 0.012261580 0.009536785 0.039509537 0.136239782 0.797002725
  t 0.002123142 0.002123142 0.000000000 0.016985138 0.021231423 0.063694268 0.208067941 0.685774947

   property_type
Y    Aparthotel   Apartment        Barn Bed and breakfast        Boat Boutique hotel    Bungalow       Cabin   Camper/RV
  f 0.001362398 0.949591281 0.000000000       0.000000000 0.000000000    0.000000000 0.000000000 0.000000000 0.000000000
  t 0.006369427 0.938428875 0.000000000       0.000000000 0.000000000    0.000000000 0.000000000 0.000000000 0.000000000
   property_type
   room_type
Y   Entire home/apt  Hotel room Private room Shared room
  f     0.799727520 0.000000000  0.186648501 0.013623978
  t     0.798301486 0.006369427  0.182590234 0.012738854

   host_is_superhost
Y           f         t
  f 0.8637602 0.1362398
  t 0.7855626 0.2144374

   require_guest_phone_verification
Y            f          t
  f 0.97411444 0.02588556
  t 0.98301486 0.01698514

   cancellation_policy
Y      flexible    moderate      strict strict_14_with_grace_period super_strict_30 super_strict_60
  f 0.279291553 0.167574932 0.000000000                 0.547683924     0.001362398     0.004087193
  t 0.303609342 0.163481953 0.000000000                 0.450106157     0.027600849     0.055201699

>
```

Figure 15: Naïve Bayes Classification Model

```
> predict(nb_model, newdata = fictionalapartment)
[1] f
Levels: f t
> predict(nb_model, newdata = fictionalapartment, type = "raw") #55%,44%
             f         t
[1,] 0.5518521 0.4481479
>
```

Figure 16.1: Assessing the model through predict function

```
> instantbookablefalse / (instantbookabletrue+instantbookablefalse) # 54%
[1] 0.5438624
> instantbookabletrue / (instantbookabletrue+instantbookablefalse) # 45%
[1] 0.4561376
```

Figure 16.2: Assessing the model through calculating probabilities

```
> # Training data
> pred1 <- predict(nb_model, newdata = Train_nb)
> confusionMatrix(pred1, Train_nb$instant_bookable)
Confusion Matrix and Statistics

          Reference
Prediction   f   t
         f 681 379
         t  53  92

               Accuracy : 0.6415
                 95% CI : (0.6137, 0.6686)
    No Information Rate : 0.6091
    P-Value [Acc > NIR] : 0.01122

                  Kappa : 0.1406

 Mcnemar's Test P-Value : < 2e-16

            Sensitivity : 0.9278
            Specificity : 0.1953
         Pos Pred Value : 0.6425
         Neg Pred Value : 0.6345
             Prevalence : 0.6091
         Detection Rate : 0.5651
   Detection Prevalence : 0.8797
      Balanced Accuracy : 0.5616

       'Positive' Class : f
```

```
> # Validation data
> pred2 <- predict(nb_model, newdata = Valid_nb)
> confusionMatrix(pred2, Valid_nb$instant_bookable)
Confusion Matrix and Statistics

          Reference
Prediction   f   t
         f 435 267
         t  44  58

               Accuracy : 0.6132
                 95% CI : (0.5785, 0.647)
    No Information Rate : 0.5958
    P-Value [Acc > NIR] : 0.166

                  Kappa : 0.0973

 Mcnemar's Test P-Value : <2e-16

            Sensitivity : 0.9081
            Specificity : 0.1785
         Pos Pred Value : 0.6197
         Neg Pred Value : 0.5686
             Prevalence : 0.5958
         Detection Rate : 0.5410
   Detection Prevalence : 0.8731
      Balanced Accuracy : 0.5433

       'Positive' Class : f
```

Figure 17: Testing the training set against the Validation Set

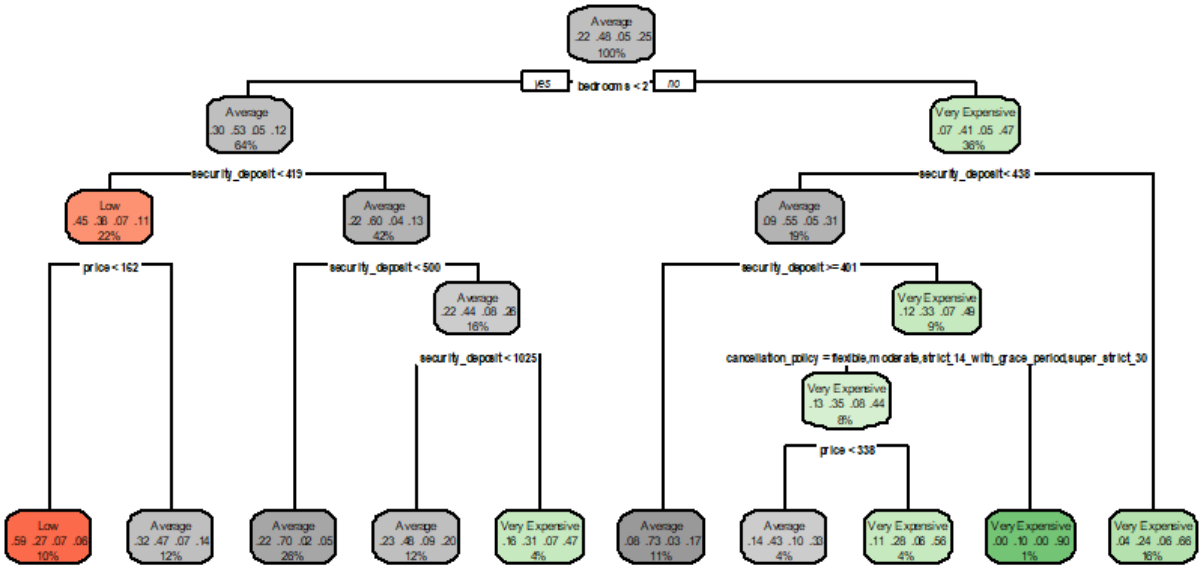

Figure 18: Classification Tree

```r
alpha <- read.csv("alpha.csv") #Load clean Copacabana rentals dataset
alpha$id <- 1:nrow(alpha)

range(alpha$accommodates)

CopaCluster <- subset(alpha, select = c(id,accommodates,guests_included,
                                        bathrooms,bedrooms,beds,price,
                                        minimum_nights,security_deposit,
                                        cleaning_fee,extra_people))

#Cost if maxing out guests accommodated into custom variable
CopaCluster$Max.Cost <- (CopaCluster$price*CopaCluster$minimum_nights)+
  ((CopaCluster$accommodates-CopaCluster$guests_included)*CopaCluster$extra_people)+
  CopaCluster$security_deposit+
  CopaCluster$cleaning_fee

fivenum(CopaCluster$accommodates)
fivenum(CopaCluster$beds) #So general rule is beds*2=accommodates

#Just renaming but this is our max guest custom variable
CopaCluster$Max.Guests <- CopaCluster$accommodates

row.names(CopaCluster) <- CopaCluster[,1] #Overwrites row num with last name
CopaCluster <- CopaCluster[,-1] #Drops row containing last name

#Normalizes to z-scores and add to new df
CopaCluster.norm <- sapply(CopaCluster, scale)
#Adds the last names back to df by overwrite row num
row.names(CopaCluster.norm) <- row.names(CopaCluster)

#new df with custom variables and last names over row num
NewClusterScores <- data.frame(subset(CopaCluster.norm, select =
                                        c(Max.Guests,Max.Cost)))
class(NewClusterScores)

set.seed(110)
#kmeans(dataframe, count of centers, random sets chosen)
Copakm <- kmeans(NewClusterScores, 4, nstart=15)
Copakm$cluster #Show which rows belong to which clusters
Copakm$centers #Matrix showing centers by cluster and variable
dist(Copakm$centers) #Distances between cluster centers
#Adds cluster assignments to df containing custom variables
NewClusterScores <- cbind(NewClusterScores, Copakm$cluster)

class(NewClusterScores) #df proceed to ggplot

#Renames cluster column so more comprehensive
colnames(NewClusterScores)[colnames(NewClusterScores) ==
                             'Copakm$cluster'] <- 'cluster'

ggplot(data=NewClusterScores, aes(x=Max.Guests, y=Max.Cost, color=cluster)) +
  geom_point(size=3)

#Coerce from factor to character so can name clusters meaningfully, not random num
NewClusterScores$cluster <- as.character(NewClusterScores$cluster)

NewClusterScores$cluster[NewClusterScores$cluster == "4"]  <- "Average Value"
NewClusterScores$cluster[NewClusterScores$cluster == "3"]  <- "Crash Pad"
NewClusterScores$cluster[NewClusterScores$cluster == "2"]  <- "Overpriced"
NewClusterScores$cluster[NewClusterScores$cluster == "1"]  <- "Best Value"

ggplot(data=NewClusterScores, aes(x=Max.Guests, y=Max.Cost, color=cluster)) +
  geom_point(size=3) +
  ggtitle("Best Value of Airbnbs in Copacabana, BR")+
  theme(plot.title = element_text(hjust = 0.5))
```

Figure 19: Clustering Analysis using k-means

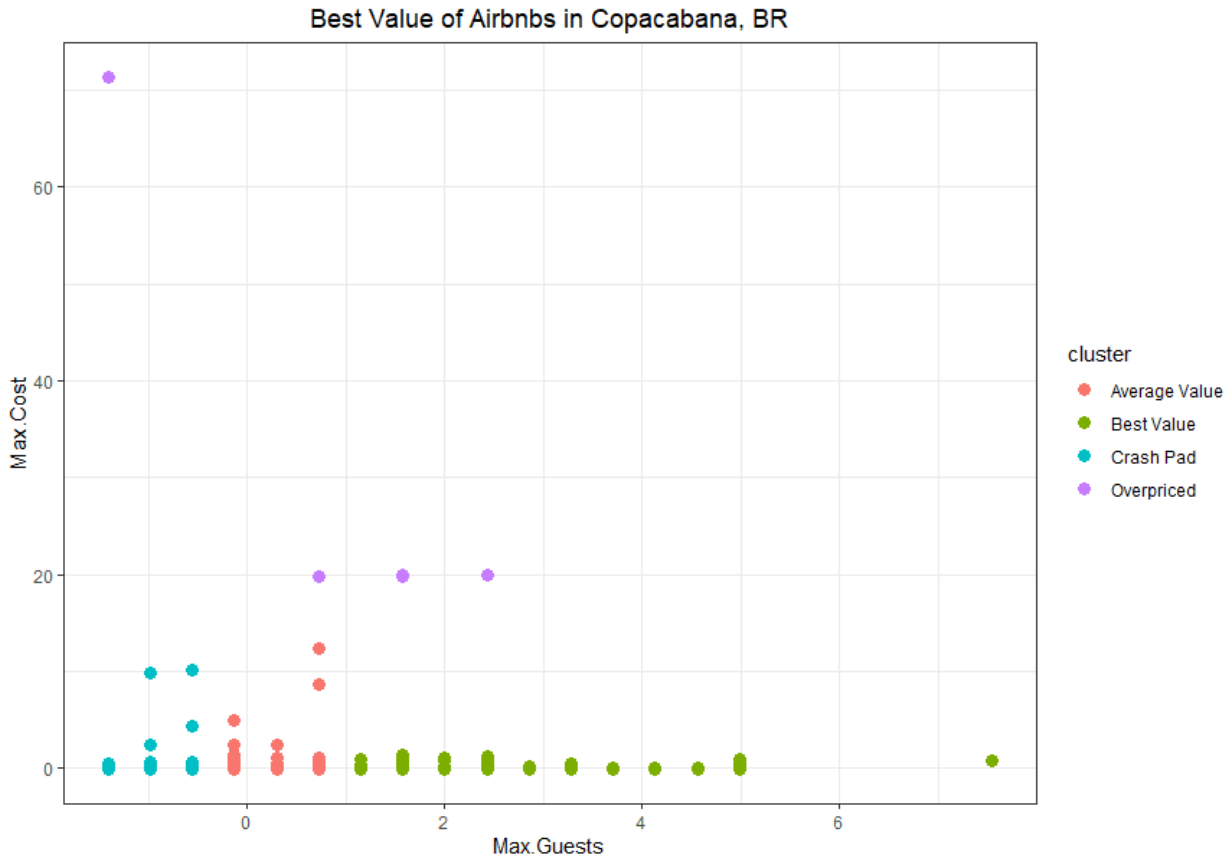

Figure 20: Clusters formed by k-means