Toronto Airbnb data research. Application of Supervised Learning Methods and Unsupervised Learning Methods

by Ketao Li, Kush Halani, Josue Romain, Juan Peña

Abstract From thousands of listings in different cities, Airbnb has become a massive sink of information. Data provided by homeowners are often big, messy, yet, extremely useful. The goal of this project is to extract knowledge from these datasets by applying techniques and methodologies common in data mining.As more homeowners put their properties on the platform, Airbnb is able to suggest appropriate prices for the listings based on machine learning models trained over large sets of data. Our team aims to predict the prices of the listings in Toronto, to allow homeowners to price their properties appropriately. Specifically, we seek to answer the following question: What prices should Airbnb suggest to their hosts given a set of features about the listing? This question is important because as more data becomes available, more intelligence can be extracted using modern machine learning tools. Therefore, it is worthwhile in exploring data driven analyses similar to those presented in this report as they are likely to improve experience for both hosts and customers, and ultimately add value to the company.Our team also try to cluster the listings in groups.

```
#> Warning: package 'ggmap' was built under R version 3.6.3
```

Background

Airbnb has seen a meteoric growth since its inception in 2008 with the number of rentals listed on its website growing exponentially each year. Airbnb has successfully disrupted the traditional hospitality industry as more and more travellers, not just the ones who are looking for a bang for their buck but also business travellers resort to Airbnb as their premier accommodation provider. Toronto has been one of the hottest markets for Airbnb in Canada, with over 26,000 listings as of Feb 2020. This means there are over 40 homes being rented out per square km in Toronto on Airbnb! One can perhaps attribute the success of Airbnb in Toronto to the high rates charged by the hotels, which are primarily driven by the exorbitant rental prices in the city.

Objective

The objective of customers segment according to their purchase history, is to turn them into loyal customers by recommending products of their choice.

This study pursues two goals. The main objective is to predict the airbnb rental rate as accurate as possible. We will employ CRISP-DM methodology (Ref: Jiawei Han (2012)) and supervised learning approach, to achieve this goal.

We are also motivated to cluster the toronto airbnb listings in groups. The questions we will try to address are:

- If there are specific groups of listings that share similar features
- If yes, how to use these group information for business.

To achieve the second goal we will use unsupervised learning approach.

Data Analysis

This research employs the data set sourced from Airbnb. This real-life data comprises 26000 observations of the listing information of Toronto on Feb 14,2020.

Data Dictionary

Column Name	Column Description
id	listing ID
name	Listing Title
host_id	ID of Host
host_name	Name of Host
neighbourhood_group	Borough that contains listing
neighbourhood	Name of neighbourhood that listing is in
latitude	latitude coordinates of listing
longitude	longitude coordinates of listing
room_type	Type of public space that is being offered
price	price per night in dollars
minimum_nights	minimum number of nights required to book listing
number_of_reviews	total number of reviews that listing has accumulated
last_review	date in which listing was last reviewed
reviews_per_month	total number of reviews divided by the number of months the listing is active
calculated_host_listings_count	amount of listing per host
availability_365	number of days per year the listing is active

Data Exploration

Statistics

Firstly we are going to load and examine content and statistics of the data set

```
data = read.csv("../data/AB_TOR_2019.csv", header = T,
                na.strings = c("NA","","#NA"),sep=",")
```

Table 2: Toronto Airbnb Dataset Summary

No	Variable	Stats / Values	Freqs (% of Valid)	Missing
1	id [integer]	Mean (sd) : 25700180.4 (11858759.3) min < med < max: 1419 < 27306796 < 42289607 IQR (CV) : 19949214.8 (0.5)	23398 distinct values	0 (0%)
2	name [factor]	1. " BEAUTIFULLY RENOVATE 2. 'NEW' Downtown Toronto 4 3. 'Treasure Box' Studio Apa [22920 others]	1 (0.0%) 1 (0.0%) 1 (0.0%) 23394 (100.0%)	1 (0%)
3	host_id [integer]	Mean (sd) : 104673805.4 (98549416.1) min < med < max: 1565 < 67279440 < 335942960 IQR (CV) : 158597247 (0.9)	15031 distinct values	0 (0%)
4	host_name [factor]	1. (Email hidden by Airbnb) 2. (Phone number hidden by A 3. (Wendy) Uyen [6243 others]	1 (0.0%) 1 (0.0%) 1 (0.0%) 23392 (100.0%)	3 (0.01%)
5	neighbourhood_group [logical]	All NA's		23398 (100%)
6	neighbourhood [factor]	1. Agincourt North 2. Agincourt South-Malvern W 3. Alderwood [137 others]	49 (0.2%) 97 (0.4%) 37 (0.2%) 23215 (99.2%)	0 (0%)
7	latitude [numeric]	Mean (sd) : 43.7 (0) min < med < max: 43.6 < 43.7 < 43.8 IQR (CV) : 0.1 (0)	10184 distinct values	0 (0%)

No	Variable	Stats / Values	Freqs (% of Valid)	Missing
8	longitude [numeric]	Mean (sd) : -79.4 (0.1) min < med < max: -79.6 < -79.4 < -79.1 IQR (CV) : 0 (0)	12918 distinct values	0 (0%)
9	room_type [factor]	1. Entire home/apt 2. Hotel room 3. Private room 4. Shared room	14991 (64.1%) 69 (0.3%) 7913 (33.8%) 425 (1.8%)	0 (0%)
10	price [integer]	Mean (sd) : 147.6 (325.8) min < med < max: 0 < 99 < 13256 IQR (CV) : 94 (2.2)	479 distinct values	0 (0%)
11	minimum_nights [integer]	Mean (sd) : 6.8 (28.6) min < med < max: 1 < 2 < 1125 IQR (CV) : 2 (4.2)	101 distinct values	0 (0%)
12	number_of_reviews [integer]	Mean (sd) : 28.5 (52.9) min < med < max: 0 < 8 < 803 IQR (CV) : 30 (1.9)	393 distinct values	0 (0%)
13	last_review [factor]	1. 2010-08-11 2. 2011-08-30 3. 2012-07-16 [1419 others]	1 (0.0%) 1 (0.0%) 1 (0.0%) 19023 (100.0%)	4372 (18.69%)
14	reviews_per_month [numeric]	Mean (sd) : 1.8 (2.1) min < med < max: 0 < 1 < 16.9 IQR (CV) : 2.2 (1.2)	1045 distinct values	4372 (18.69%)
15	calculated_host_listings_count [integer]	Mean (sd) : 5.3 (12.2) min < med < max: 1 < 1 < 119 IQR (CV) : 3 (2.3)	43 distinct values	0 (0%)
16	availability_365 [integer]	Mean (sd) : 126.2 (127.3) min < med < max: 0 < 85 < 365 IQR (CV) : 225 (1)	366 distinct values	0 (0%)

From the above summary, we can find that there are some negative values for Quantity and UnitPrice.These values don't make sense, so we'll delete them directly.

```
#head(data)
#summary(data)
#str(data)
```

Table **??** describes main statistical parameters of each column. Here is a look at the data sample.

% latex table generated in R 3.6.2 by xtable 1.8-4 package % Wed Mar 18 16:29:28 2020

id	name	host_id	host_name	neighbourhood_group	neighbourhood	latitude	longitude
1419	Beautiful home in amazing area!	1565	Alexandra		Little Portugal	43.65	-79.42
8077	Downtown Harbourfront Private Room	22795	Kathie & Larry		Waterfront Communities-The Island	43.64	-79.38
12604	Seaton Village Parlour Bedroom	48239	Rona		Annex	43.67	-79.42
23691	Queen Bedroom close to downtown	93825	Yohan & Sarah		Briar Hill-Belgravia	43.70	-79.45
26654	World Class downtown @CN Tower Theatre MTCC games!	113345	Adela		Waterfront Communities-The Island	43.65	-79.39
27423	Executive Studio Unit- Ideal for One Person	118124	Brent		Greenwood-Coxwell	43.67	-79.33
30931	Downtown Toronto - Waterview Condo	22795	Kathie & Larry		Waterfront Communities-The Island	43.64	-79.38
40456	Downtown 2 Bdr.Apt with King Size Bed and Parking	174063	Denis		South Parkdale	43.64	-79.44
41887	Great location	183071	Kyle		Oakridge	43.69	-79.29
43964	Bright entire 2-bedrm basement suite private entry	192364	Mitra		Wexford/Maryvale	43.75	-79.29
44452	Yonge & Bloor Studio Skyline	195095	Urbano		Rosedale-Moore Park	43.67	-79.39
45399	Fountain View Studio - Eaton center	195095	Urbano		Church-Yonge Corridor	43.66	-79.38

% latex table generated in R 3.6.2 by xtable 1.8-4 package % Wed Mar 18 16:29:28 2020

###Missing data

room_type	price	minimum_nights	number_of_reviews	last_review	reviews_per_month	calculated_host_listings_count	availability_365
Entire home/apt	469	4	7	2017-12-04	0.13	1	0
Private room	99	180	169	2013-08-27	1.32	2	0
Private room	66	1	0			1	0
Private room	72	1	217	2019-12-22	1.84	2	0
Entire home/apt	199	4	39	2020-01-06	0.35	5	365
Entire home/apt	54	120	26	2011-08-30	0.22	1	0
Entire home/apt	133	180	1	2010-08-11	0.01	2	365
Entire home/apt	99	30	109	2019-11-08	0.94	5	250
Entire home/apt	69	2	82	2019-09-02	2.09	2	270
Entire home/apt	90	2	30	2019-08-05	0.79	1	365
Entire home/apt	121	1	50	2019-12-24	0.44	13	363
Entire home/apt	121	1	78	2019-11-07	0.69	13	363

Table 3: Toronto Airbnb Data Sample

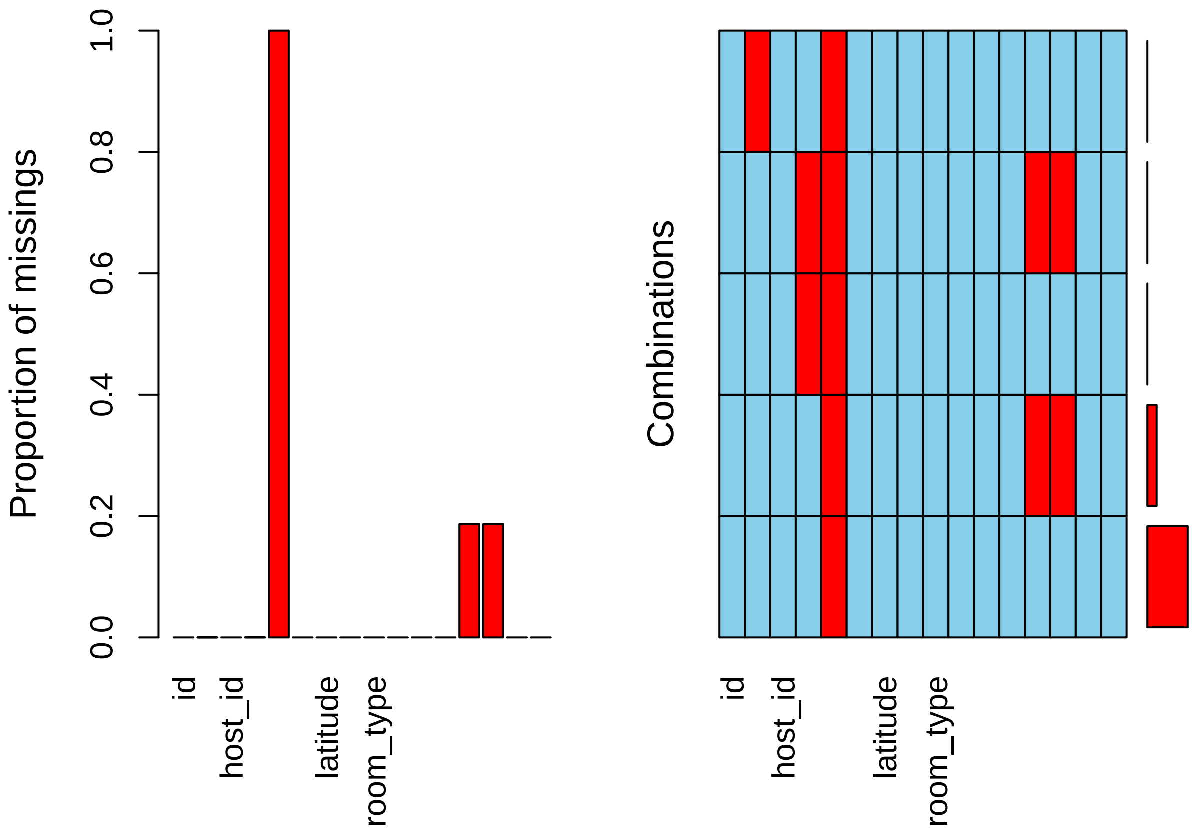


Figure 1: Missing data

```
summary(a)
```

```
 Missings per variable:
                       Variable Count
                             id     0
                           name     1
                        host_id     0
                      host_name     3
            neighbourhood_group 23398
                  neighbourhood     0
                       latitude     0
                      longitude     0
                      room_type     0
                          price     0
                 minimum_nights     0
              number_of_reviews     0
                    last_review  4372
              reviews_per_month  4372
 calculated_host_listings_count     0
```

```
              availability_365     0

 Missings in combinations of variables:
                   Combinations Count      Percent
 0:0:0:0:1:0:0:0:0:0:0:0:0:0:0 19023 81.301820668
 0:0:0:0:1:0:0:0:0:0:0:1:1:0:0  4371 18.681083853
 0:0:0:1:1:0:0:0:0:0:0:0:0:0:0     2  0.008547739
 0:0:0:1:1:0:0:0:0:0:0:1:1:0:0     1  0.004273870
 0:1:0:0:1:0:0:0:0:0:0:0:0:0:0     1  0.004273870
```

There are some missing data for neighbourhood_group,last_review and reviews_per_month.It's very strange that most of the neighbourhood_group miss in Toronto.We decide to ignore this feature.

Data Transformation

We need do some some data transformation.

```
customerData <- data %>%
  mutate( last_review= as.Date(last_review))

customerData <- customerData %>%
  mutate( last_review= as.numeric(as.Date("2020-02-15")-last_review))

customerData <- customerData %>%
  mutate( room_type= as.numeric(room_type) )

#customerData <- customerData %>%
#  mutate(last_review = as.numeric(as.Date("2020-02-15")-last_review))

#glimpse(customerData)
```

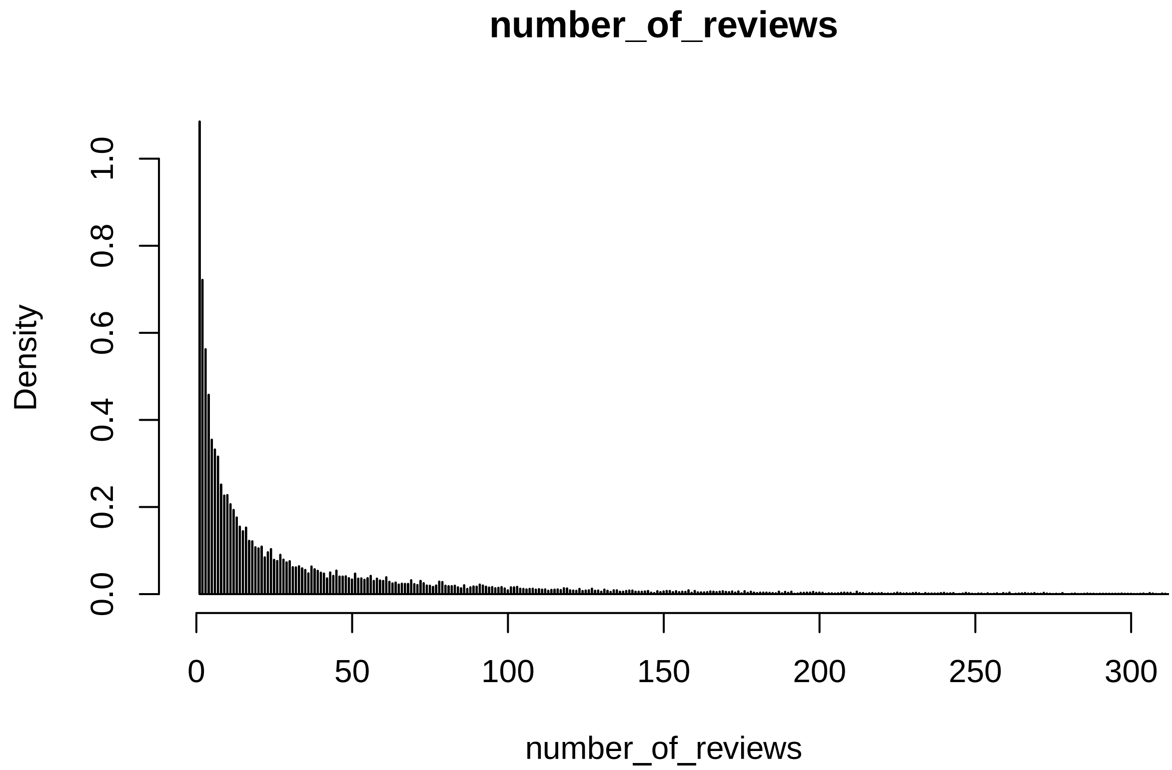
We drop neighbourhood_group and other missing data.

```
customerData1 = subset(customerData,select = -neighbourhood_group)
customerData1 = customerData1 %>%filter(complete.cases(.))
glimpse(customerData1)

Observations: 19,023
Variables: 15
$ id                             <int> 1419, 8077, 23691, 26654, 27423, 309...
$ name                           <fct> "Beautiful home in amazing area!", "...
$ host_id                        <int> 1565, 22795, 93825, 113345, 118124, ...
$ host_name                      <fct> Alexandra, Kathie & Larry, Yohan & S...
$ neighbourhood                  <fct> Little Portugal, Waterfront Communit...
$ latitude                       <dbl> 43.64617, 43.64105, 43.69602, 43.645...
$ longitude                      <dbl> -79.42451, -79.37628, -79.45468, -79...
$ room_type                      <dbl> 1, 3, 3, 1, 1, 1, 1, 1, 1, 1, 1, 3, ...
$ price                          <int> 469, 99, 72, 199, 54, 133, 99, 69, 9...
$ minimum_nights                 <int> 4, 180, 1, 4, 120, 180, 30, 2, 2, 1,...
$ number_of_reviews              <int> 7, 169, 217, 39, 26, 1, 109, 82, 30,...
$ last_review                    <dbl> 803, 2363, 55, 40, 3091, 3475, 99, 1...
$ reviews_per_month              <dbl> 0.13, 1.32, 1.84, 0.35, 0.22, 0.01, ...
$ calculated_host_listings_count <int> 1, 2, 2, 5, 1, 2, 5, 2, 1, 13, 13, 1...
$ availability_365               <int> 0, 0, 0, 365, 0, 365, 250, 270, 365,...

#b = aggr(customerData1)
#plot(b)
```

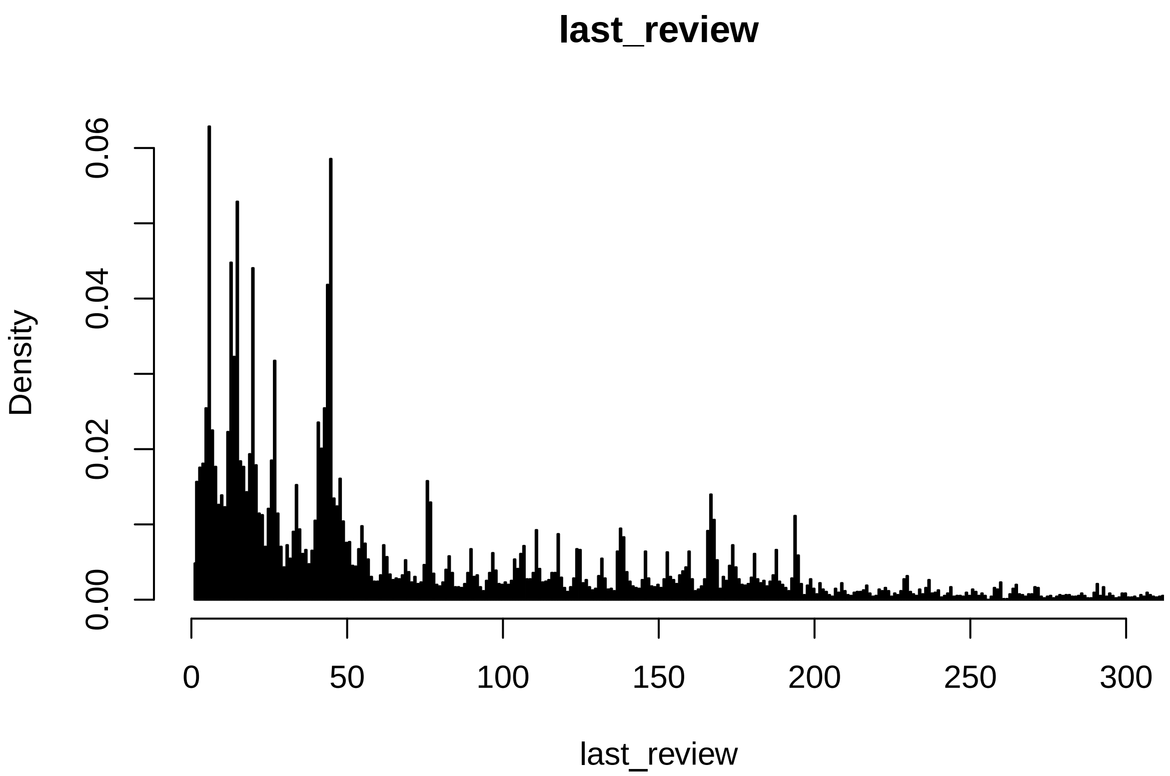
###data visualization

```
# histogram with added parameters
hist(customerData1$number_of_reviews,
main="number_of_reviews",
xlab="number_of_reviews",
breaks=10000,
```

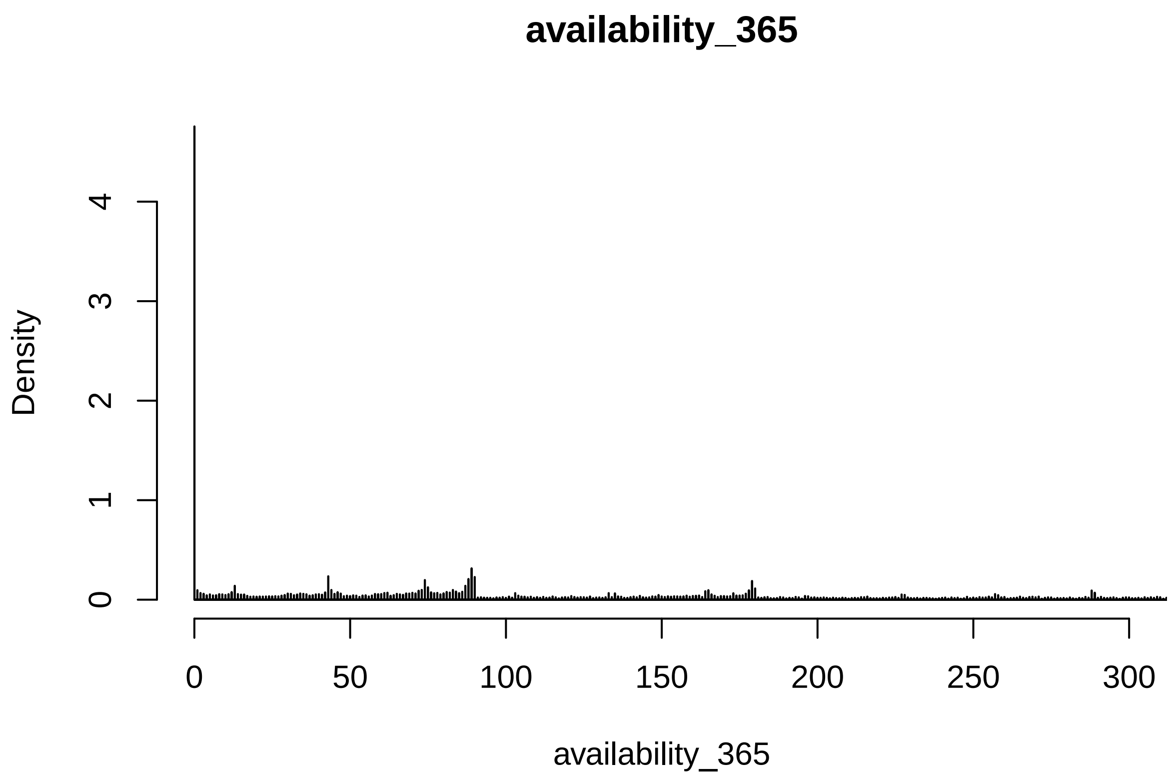
```
xlim=c(0,300),
col="darkmagenta",
freq=FALSE
)
```




```
# histogram with added parameters
hist(customerData1$last_review,
main="last_review",
xlab="last_review",
breaks=10000,
xlim=c(0,300),
col="darkmagenta",
freq=FALSE
)
```




```
# histogram with added parameters
hist(customerData1$availability_365,
main="availability_365",
xlab="availability_365",
breaks=10000,
xlim=c(0,300),
col="darkmagenta",
freq=FALSE
)
```



Hosts on Airbnb offer a wide variety of spaces, ranging from shared rooms to private islands.

All homes are grouped into the following three room types:

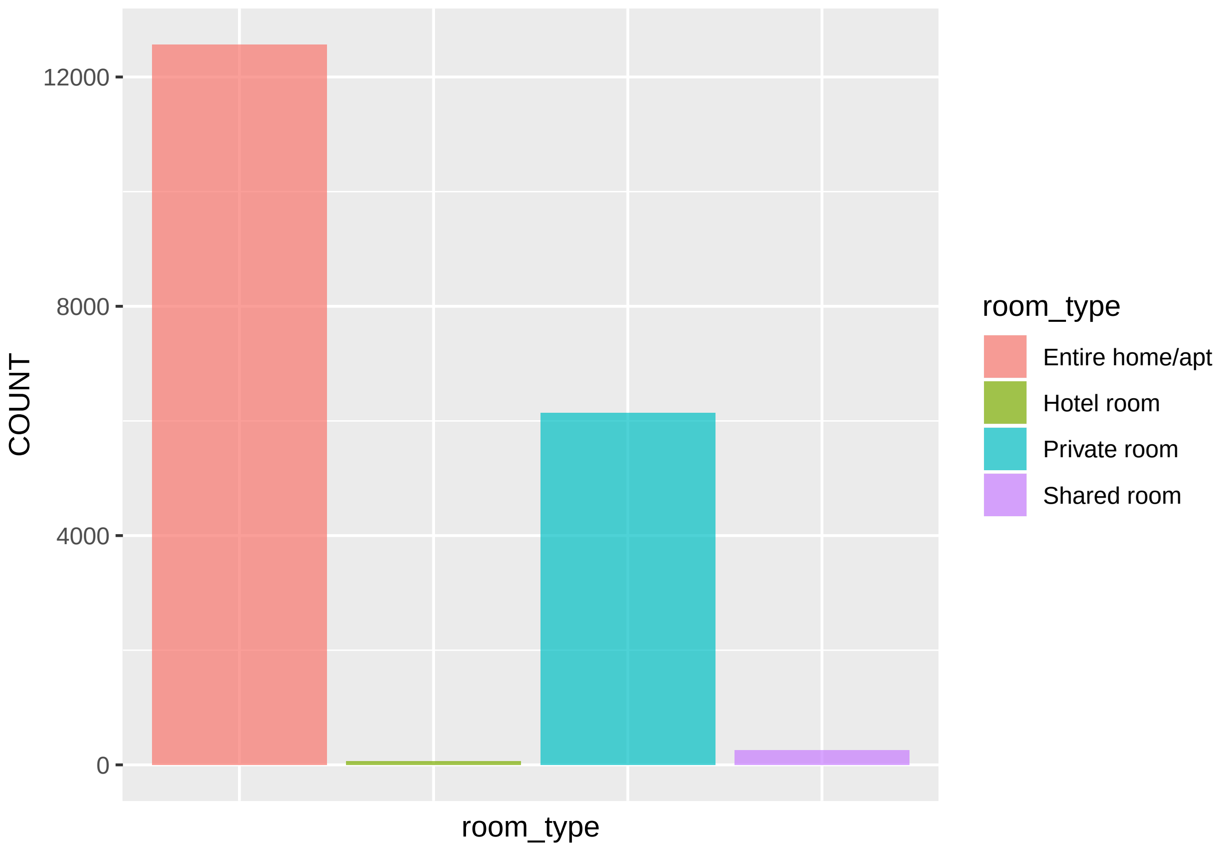
Entire place Private room Shared room hotel room

Entire place Entire places are best if you're seeking a home away from home. With an entire place, you'll have the whole space to yourself. This usually includes a bedroom, a bathroom, a kitchen, and a separate, dedicated entrance. Hosts should note in the description if they'll be on the property or not (e.g.: "Host occupies first floor of the home"), and provide further details on the listing.

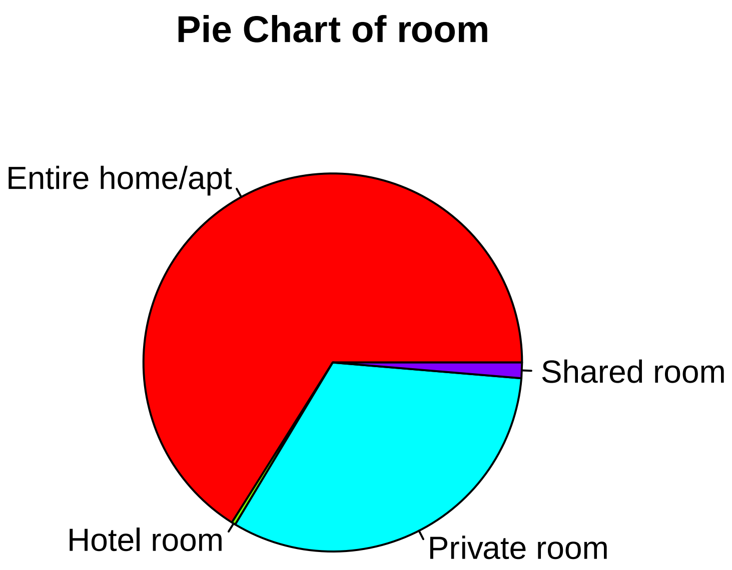
Private rooms Private rooms are great for when you prefer a little privacy, and still value a local connection. When you book a private room, you'll have your own private room for sleeping and may share some spaces with others. You might need to walk through indoor spaces that another host or guest may occupy to get to your room.

Shared rooms Shared rooms are for when you don't mind sharing a space with others. When you book a shared room, you'll be sleeping in a space that is shared with others and share the entire space with other people. Shared rooms are popular among flexible travellers looking for new friends and budget-friendly stays.

```
room = customerData1 %>%
  group_by(room_type = factor(room_type, labels = c("Entire home/apt","Hotel room","Private room","Shared roo
  summarise(COUNT = n())
p1 = ggplot(room,aes(x=room_type, y = COUNT, fill = room_type)) +
      geom_bar(stat = "identity", alpha = 0.7) +  theme(axis.text.x=element_blank(),
      axis.ticks.x=element_blank())
plot(p1)
```

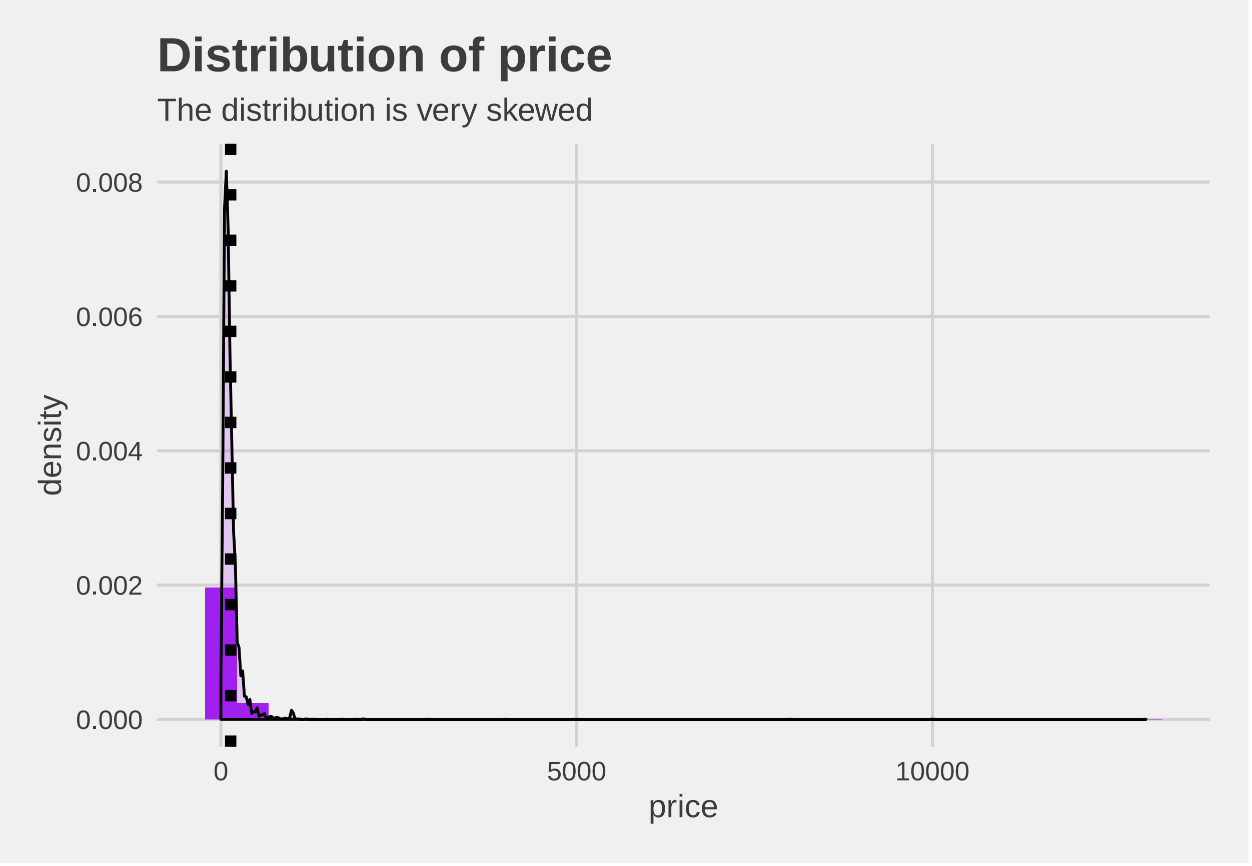



```
mytable <- table(customerData1$room_type)
lbls <-  c("Entire home/apt","Hotel room","Private room","Shared room")
pie(mytable, labels = lbls,col=rainbow(length(lbls)),
   main="Pie Chart of room")
```



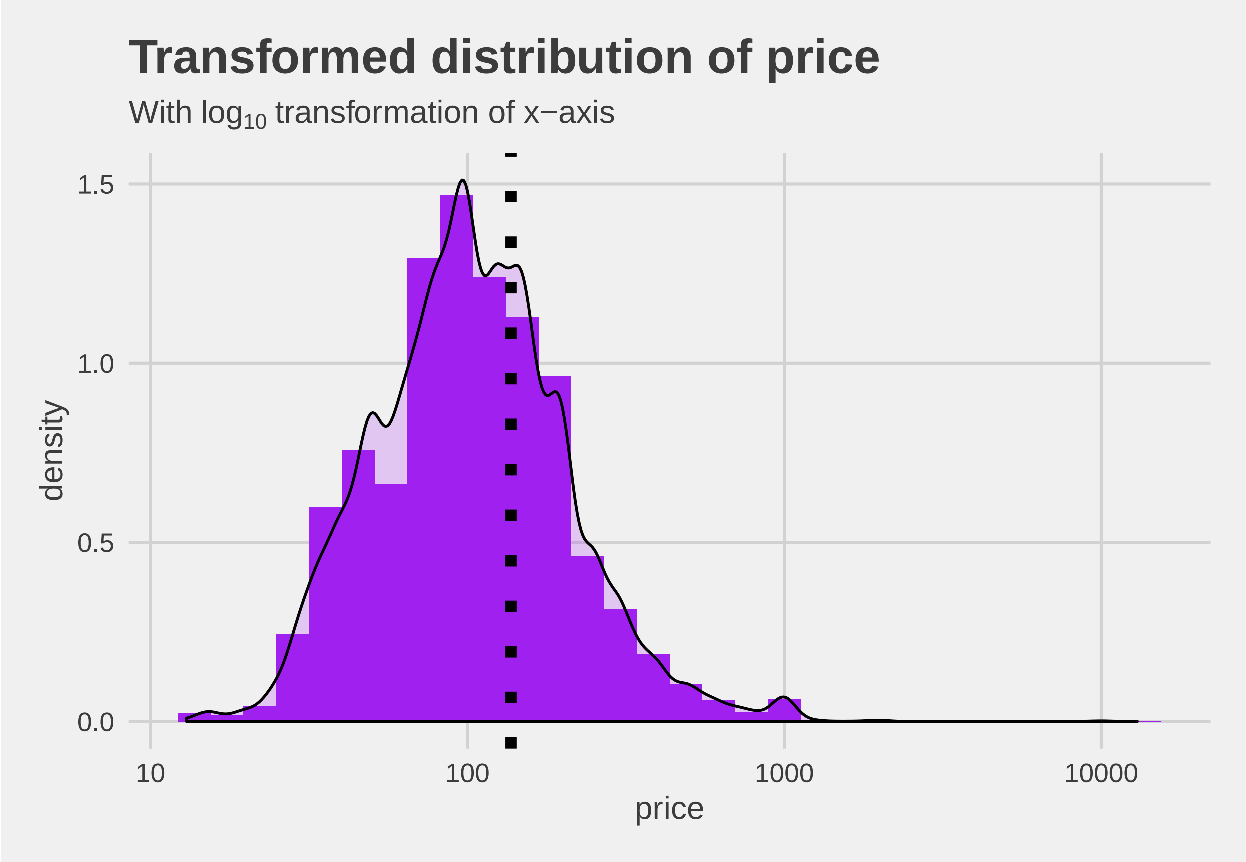
Price The most important (target) variable is price.

```
ggplot(customerData1, aes(price)) +
  geom_histogram(bins = 30, aes(y = ..density..), fill = "purple") +
  geom_density(alpha = 0.2, fill = "purple") +
  th +
  ggtitle("Distribution of price",
          subtitle = "The distribution is very skewed") +
  theme(axis.title = element_text(), axis.title.x = element_text()) +
  geom_vline(xintercept = round(mean(customerData1$price), 2), size = 2, linetype = 3)
```



Histogram & Density with log10 transformation Since the original distribution is very skewed, logarithmic transformation can be used to gain better insight into data.

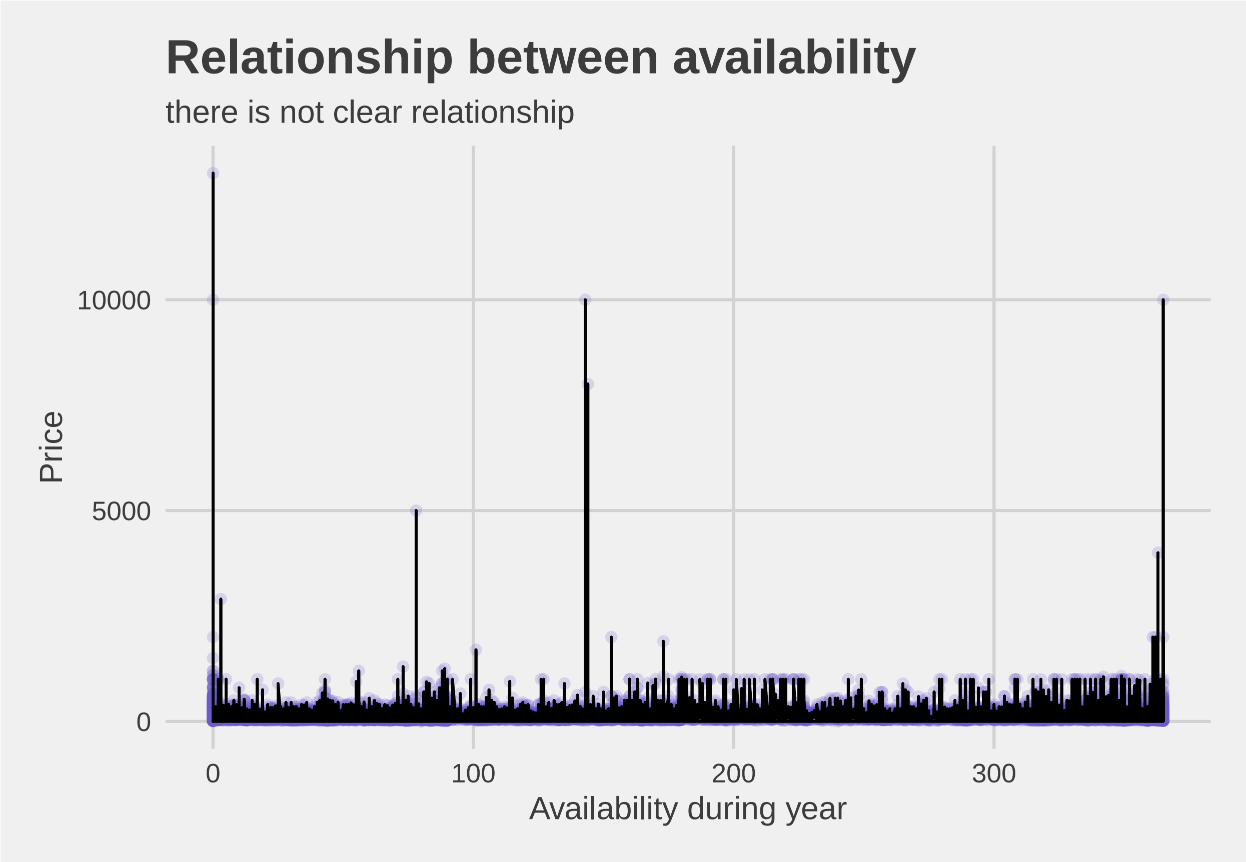
```
ggplot(customerData1, aes(price)) +
  geom_histogram(bins = 30, aes(y = ..density..), fill = "purple") +
  geom_density(alpha = 0.2, fill = "purple") +
  th +
  ggtitle("Transformed distribution of price",
          subtitle = expression("With" ~'log'[10] ~ "transformation of x-axis")) +
  #theme(axis.title = element_text(), axis.title.x = element_text()) +
  geom_vline(xintercept = round(mean(customerData1$price), 2), size = 2, linetype = 3) +
  scale_x_log10() #+
```




```
#annotate("text", x = 1800, y = 0.75,label = paste("Mean price = ", paste0(round(mean(customerData1$price),
#         color =  "#32CD32", size = 8)
```

Price & Availability

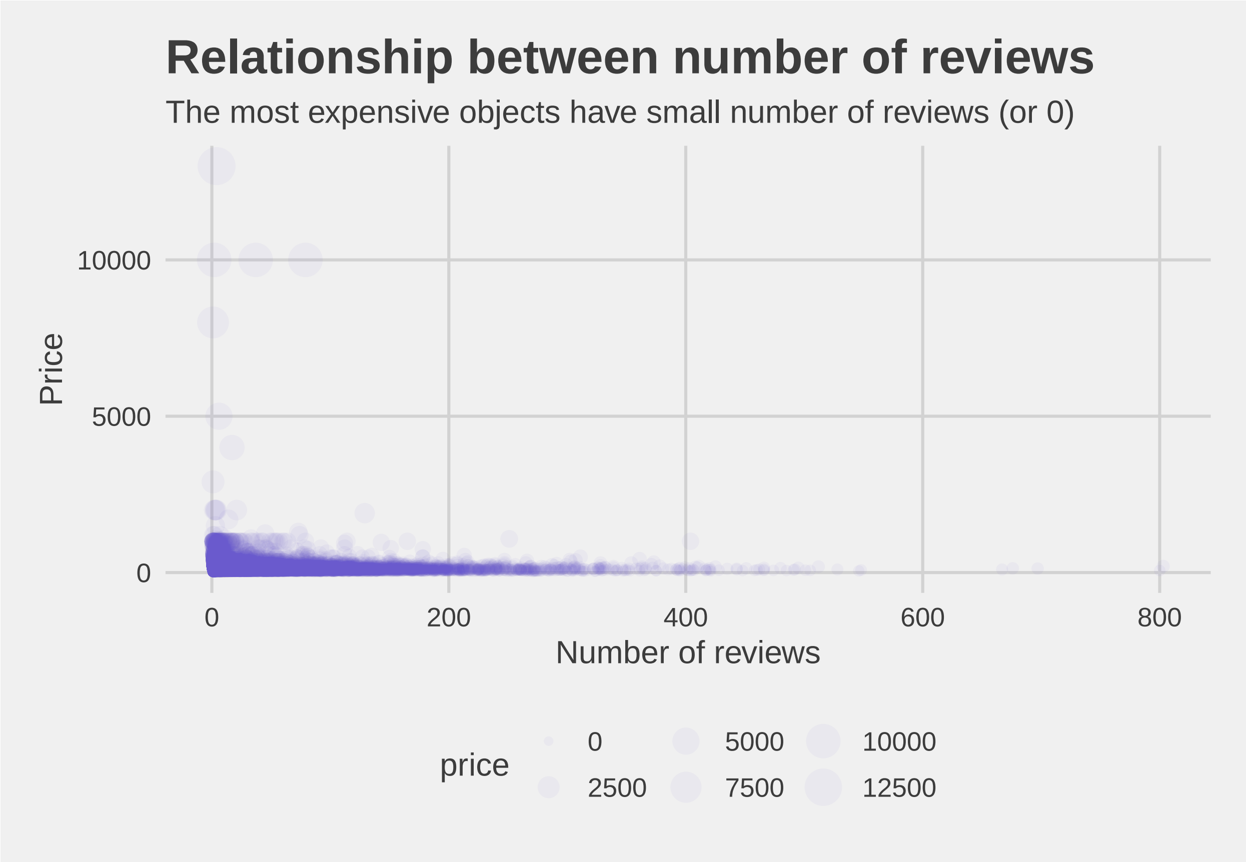
```
ggplot(customerData1, aes(availability_365, price)) +
  th +
  geom_point(alpha = 0.2, color = "slateblue") +
  geom_density(stat = "identity", alpha = 0.2) +
  xlab("Availability during year") +
  ylab("Price") +
  ggtitle("Relationship between availability",
          subtitle = "there is not clear relationship")
```



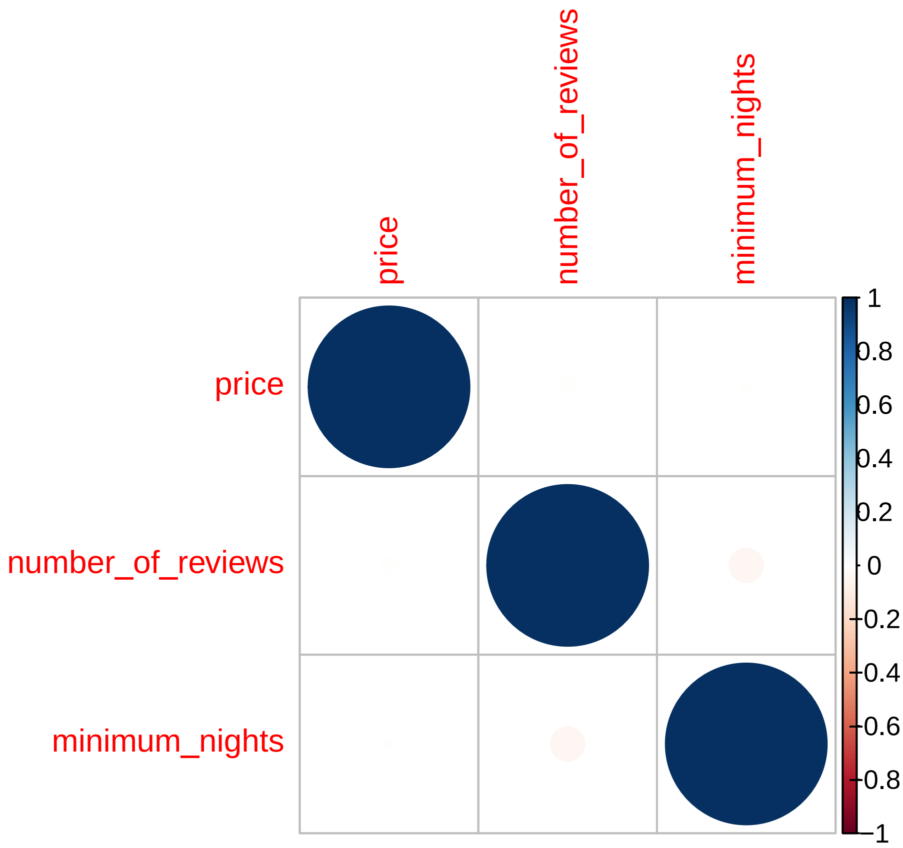
It's hard to see clear pattern, but there's a lot of expensive objects with few available days and many available days.

Price & Number Of Reviews

```
ggplot(customerData1, aes(number_of_reviews, price)) +
  th + theme(axis.title = element_text(), axis.title.x = element_text()) +
  geom_point(aes(size = price), alpha = 0.05, color = "slateblue") +
  xlab("Number of reviews") +
  ylab("Price") +
  ggtitle("Relationship between number of reviews",
          subtitle = "The most expensive objects have small number of reviews (or 0)")
```

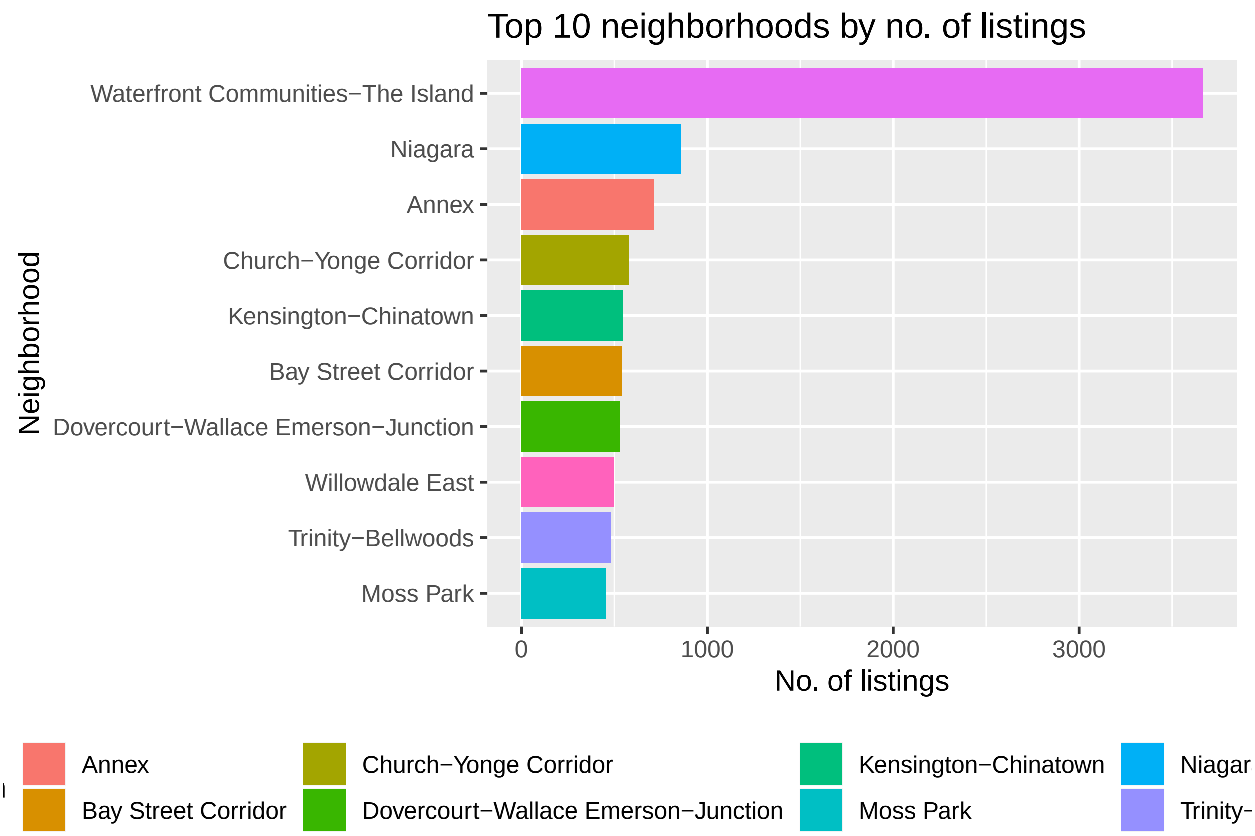



```
df  <- customerData1  %>% select("price","number_of_reviews","minimum_nights")
cor(df)

                        price number_of_reviews minimum_nights
price             1.000000000      -0.009435758   -0.003047704
number_of_reviews -0.009435758      1.000000000   -0.043245174
minimum_nights    -0.003047704      -0.043245174    1.000000000

corrplot(cor(df))
```



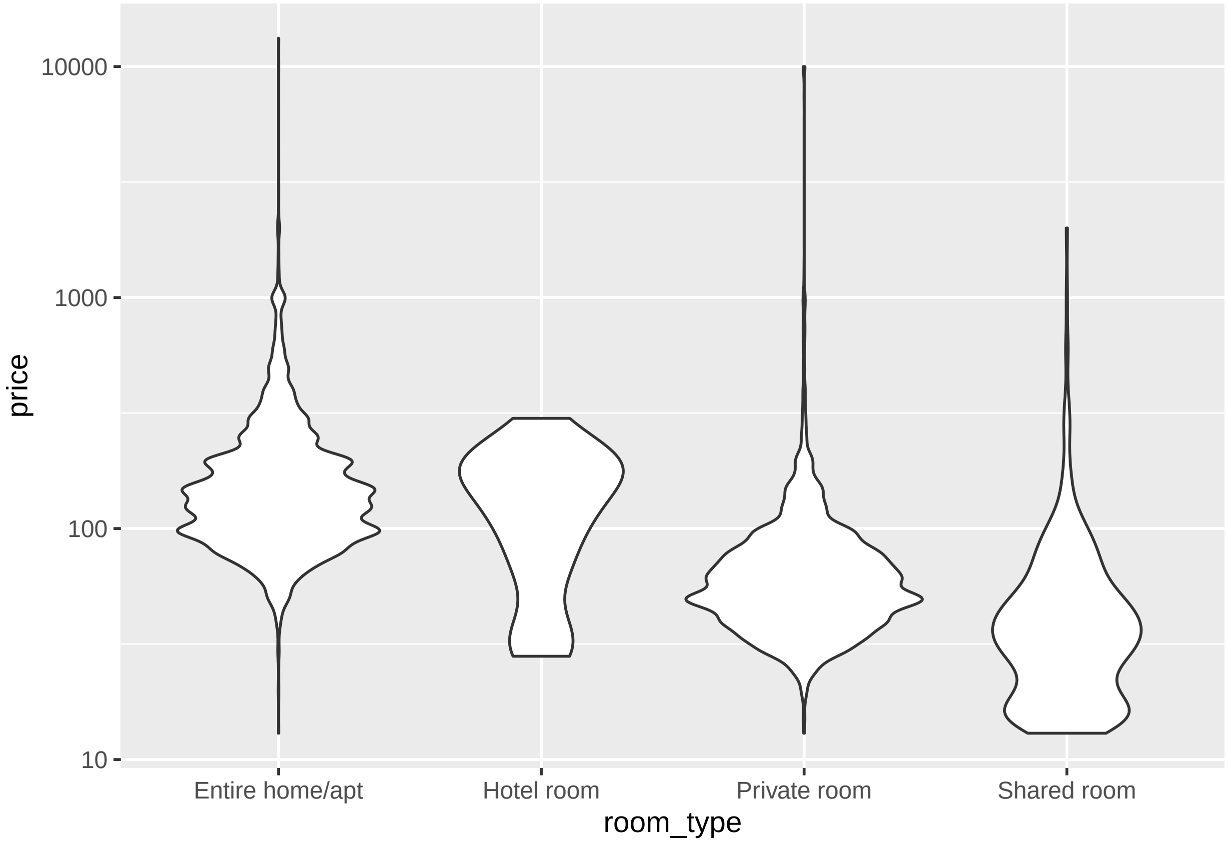
Below is a plot of the top 10 neighborhoods by number of listings.

```
customerData1 %>%
    group_by(neighbourhood) %>%
    summarize(num_listings = n(),
              borough = unique(neighbourhood)) %>%
    top_n(n = 10, wt = num_listings) %>%
    ggplot(aes(x = fct_reorder(neighbourhood, num_listings),
               y = num_listings, fill = borough)) +
    geom_col() +
    coord_flip() +
    theme(legend.position = "bottom") +
    labs(title = "Top 10 neighborhoods by no. of listings",
         x = "Neighborhood", y = "No. of listings")
```



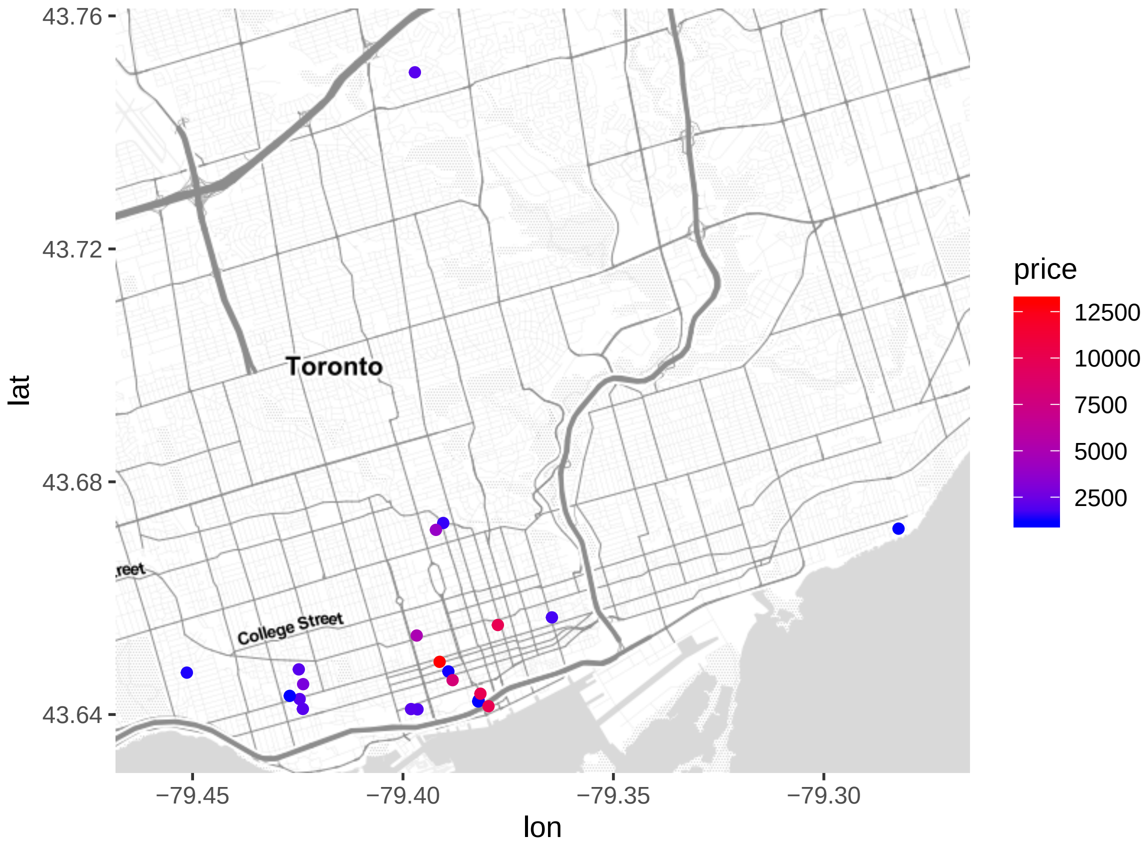
The plot below shows the distribution of price by room type. (Note that the y-axis is on a log scale.) There is much variation in price within each room type. Overall, it looks like "Entire home/apt" listings are slightly pricier than "Private room", which in turn are more expensive than "Shared room". This makes intuitive sense.

```
ggplot(data, aes(x = room_type, y = price)) +
    geom_violin() +
    scale_y_log10()
```



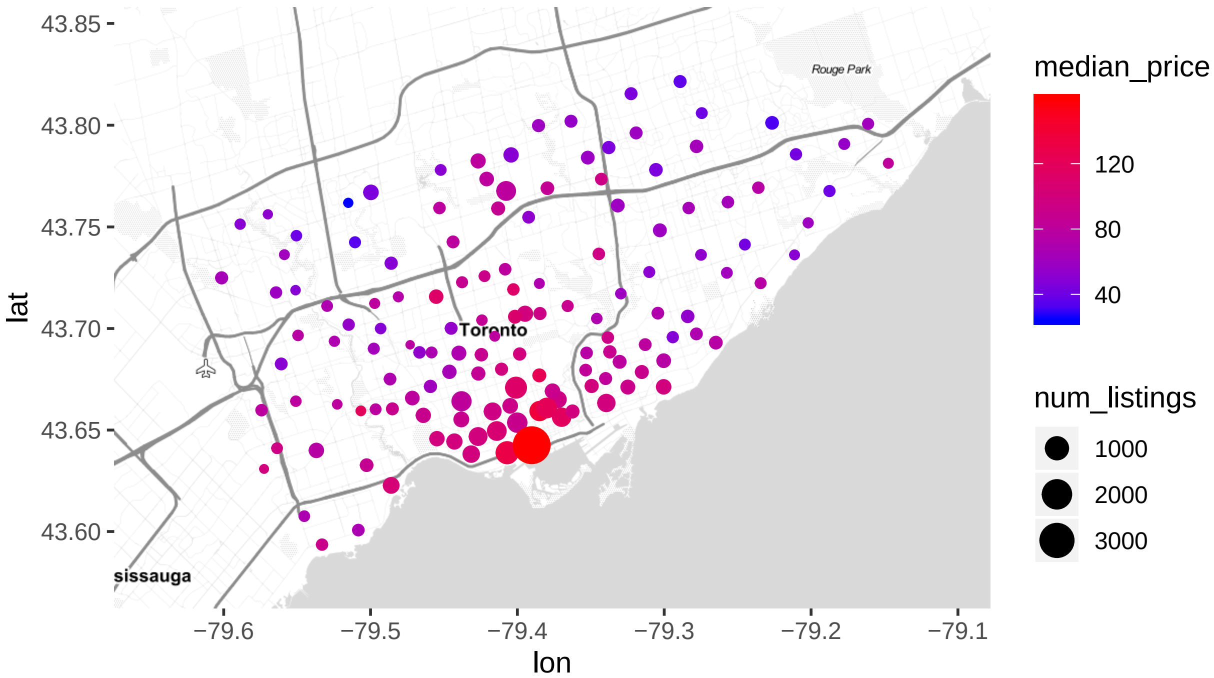
Map of the top 20 most expensive listings

```
# get top 20 listings by price
top_df <- customerData1 %>% top_n(n = 20, wt = price)

# get background map
top_height <- max(top_df$latitude) - min(top_df$latitude)
top_width <- max(top_df$longitude) - min(top_df$longitude)
top_borders <- c(bottom  = min(top_df$latitude)  - 0.1 * top_height,
                 top     = max(top_df$latitude)  + 0.1 * top_height,
                 left    = min(top_df$longitude) - 0.1 * top_width,
                 right   = max(top_df$longitude) + 0.1 * top_width)

top_map <- get_stamenmap(top_borders, zoom = 12, maptype = "toner-lite")

# map of top 50 most expensive
ggmap(top_map) +
    geom_point(data = top_df, mapping = aes(x = longitude, y = latitude,
                                    col = price)) +
    scale_color_gradient(low = "blue", high = "red")
```



Median price by neighborhood In the map below, each dot is one neighborhood. The size of the dot depends on the number of listings and the color of the dot depends on the median price in that neighborhood.

```
nhd_df <- customerData1 %>%
    group_by(neighbourhood) %>%
    summarize(num_listings = n(),
              median_price = median(price),
              long = median(longitude),
              lat = median(latitude),
              borough = unique(neighbourhood))
# map of all listings: one point per neighborhood
height <- max(customerData1$latitude) - min(customerData1$latitude)
width <- max(customerData1$longitude) - min(customerData1$longitude)
borders <- c(bottom  = min(customerData1$latitude)  - 0.1 * height,
             top     = max(customerData1$latitude)  + 0.1 * height,
             left    = min(customerData1$longitude) - 0.1 * width,
             right   = max(customerData1$longitude) + 0.1 * width)

map <- get_stamenmap(borders, zoom = 11, maptype = "toner-lite")
ggmap(map) +
    geom_point(data = nhd_df, mapping = aes(x = long, y = lat,
                                        col = median_price, size = num_listings)) +
    scale_color_gradient(low = "blue", high = "red")
```



Descriptive Features.

name is free-text features that might provide additional insights about the listing. We are going to take a close look at this feature and decide if we could utilize it.

Lets' begin with the *name*

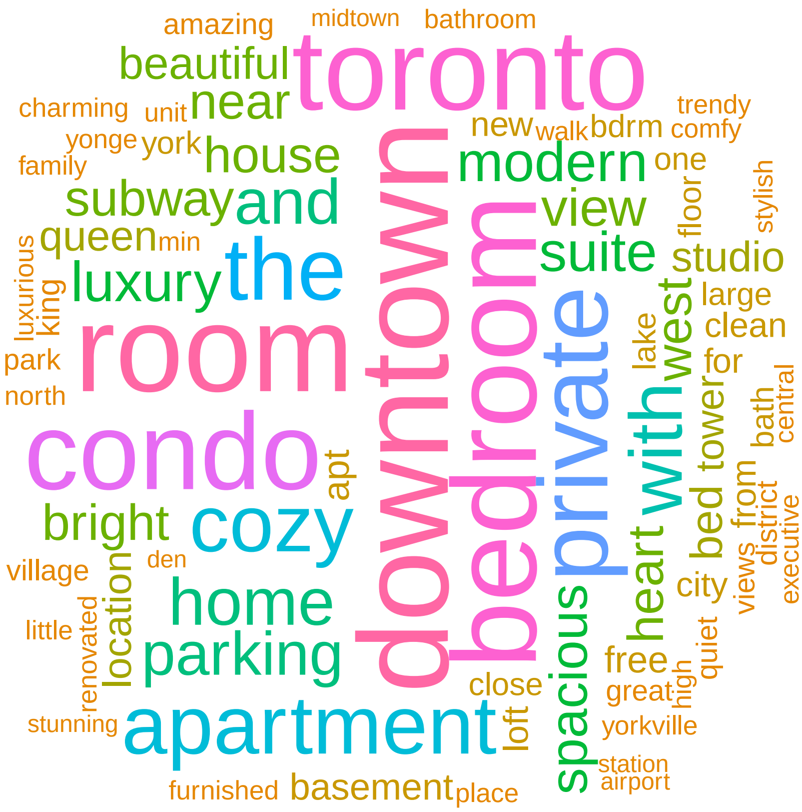


Figure 2: Most Common Words in Description

From the word cloud, we can get some highly frequently used words such as downtown,bedroom,room,condo,private,apartm

Unfortunately *name* feature does not provide more knowledge to what the others features already supply. Thus it will be dropped.

Regression Models and Model Performance Evaluation

In this section we will apply various clustering methods to cluster the customer based rfm. We will use Partitioning clustering and Hierarchical clustering approaches.

```
airbnb_train <- customerData1 %>% sample_frac(.7) %>% filter(price > 0)
airbnb_test  <- anti_join(customerData1, airbnb_train, by = 'id') %>% filter(price > 0)
```

##1st Linear Regression model

```
first_model <- train(price ~ latitude + longitude + room_type + minimum_nights  + availability_365 , data = a
summary(first_model)
```

```
Call:
lm(formula = .outcome ~ ., data = dat)

Residuals:
   Min     1Q Median     3Q    Max
-196.5  -59.2  -22.3   16.3 9879.3

Coefficients:
                   Estimate Std. Error t value Pr(>|t|)
(Intercept)       3.028e+04  3.232e+03   9.369  < 2e-16 ***
latitude         -4.218e+02  3.960e+01 -10.652  < 2e-16 ***
longitude         1.468e+02  2.862e+01   5.129 2.95e-07 ***
room_type        -4.584e+01  1.855e+00 -24.711  < 2e-16 ***
```

```
minimum_nights   -9.998e-02  6.689e-02  -1.495    0.135
availability_365  1.290e-01  1.358e-02   9.504  < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 195.4 on 13307 degrees of freedom
Multiple R-squared:  0.07377,   Adjusted R-squared:  0.07342
F-statistic:   212 on 5 and 13307 DF,  p-value: < 2.2e-16
```

This model is not so good. Median residual error is -22.9, while it should be near 0. R2=0.06 is also not so good.

Let's plot the first model.

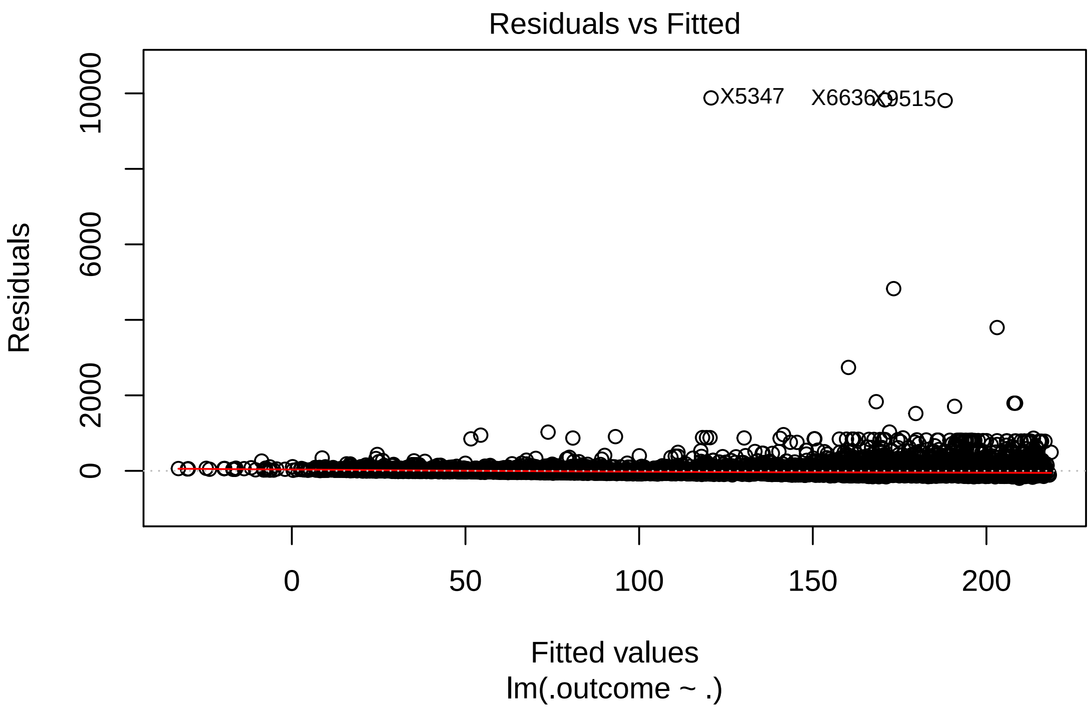


Figure 3: Number of Predictors vs Accuracy

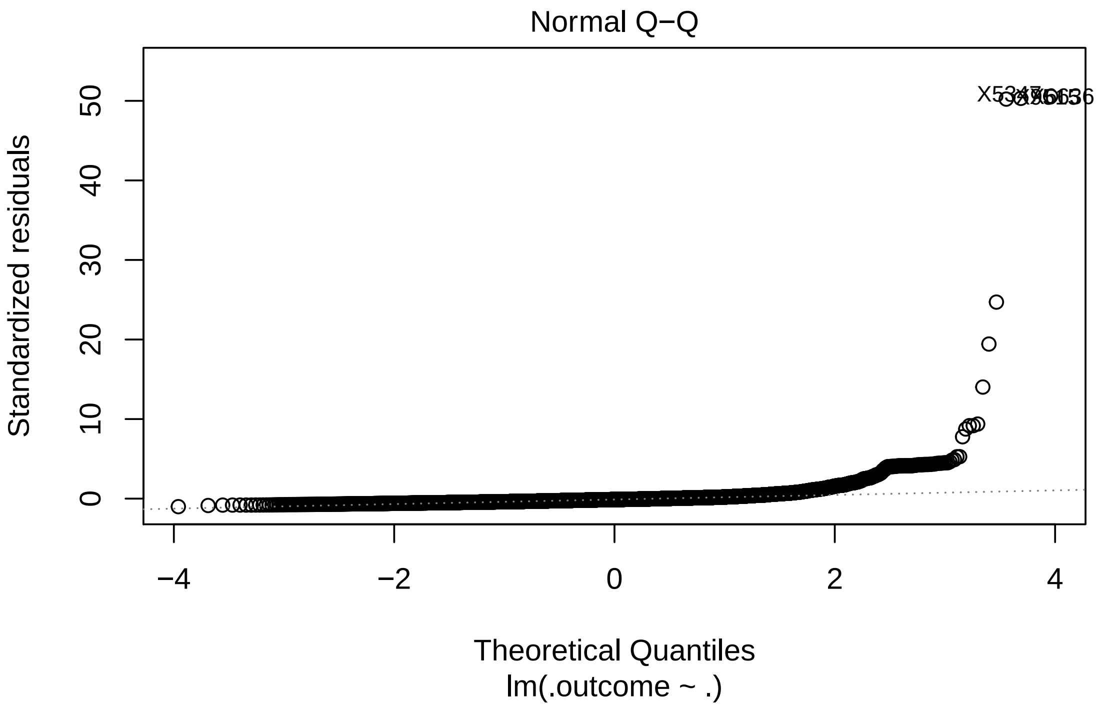


Figure 4: Number of Predictors vs Accuracy

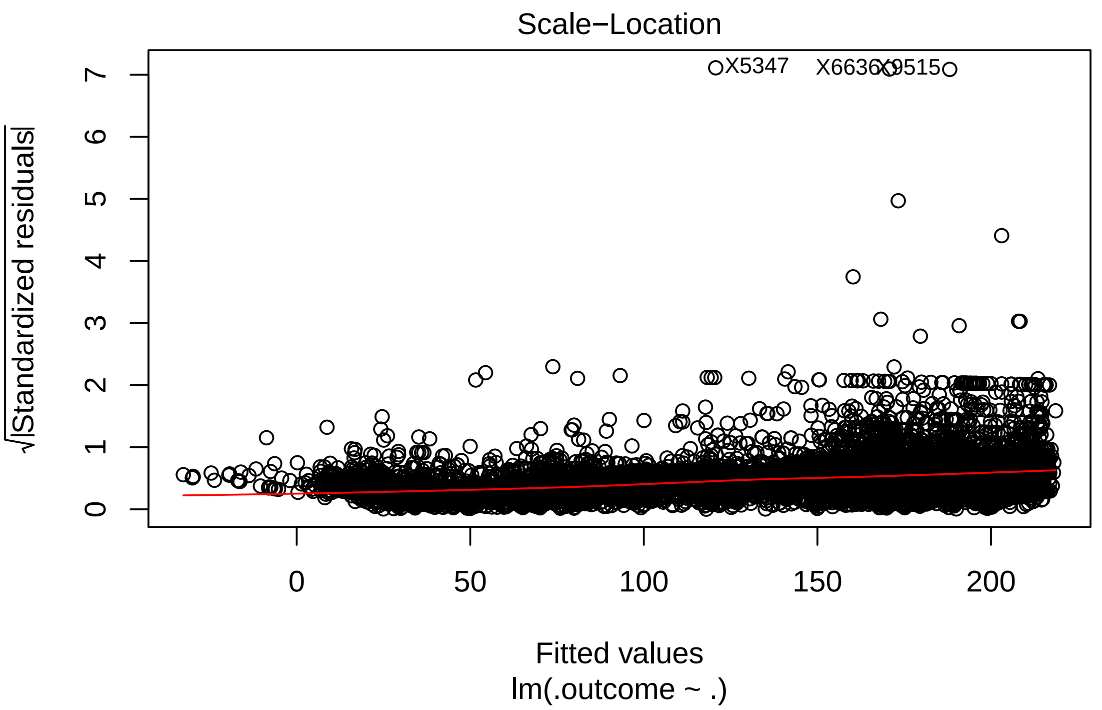


Figure 5: Number of Predictors vs Accuracy

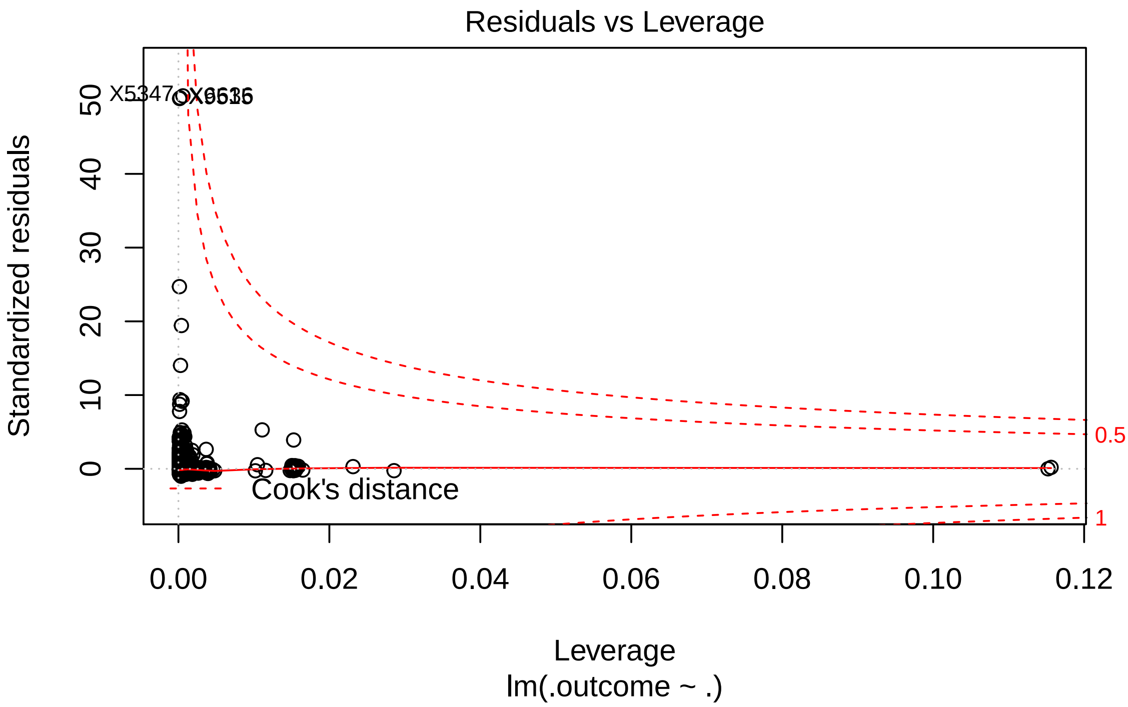


Figure 6: Number of Predictors vs Accuracy

##2nd Linear Regression model

```
learn <- airbnb_train %>% filter(price < quantile(airbnb_train$price, 0.9) & price > quantile(airbnb_train$pr
second_model <- lm(log(price) ~ room_type + latitude + longitude
                        + number_of_reviews + availability_365
                       + reviews_per_month +
                     calculated_host_listings_count + minimum_nights, data = learn)
# Summarize the results
summary(second_model)


Call:
lm(formula = log(price) ~ room_type + latitude + longitude +
    number_of_reviews + availability_365 + reviews_per_month +
    calculated_host_listings_count + minimum_nights, data = learn)

Residuals:
     Min       1Q   Median       3Q      Max
-1.10574 -0.25653 -0.01645  0.24385  1.40928

Coefficients:
                                 Estimate Std. Error t value Pr(>|t|)
(Intercept)                     1.395e+02  6.798e+00  20.516  < 2e-16 ***
room_type                      -2.599e-01  3.848e-03 -67.554  < 2e-16 ***
latitude                       -2.025e+00  8.371e-02 -24.190  < 2e-16 ***
longitude                       5.796e-01  6.064e-02   9.557  < 2e-16 ***
number_of_reviews              -4.187e-04  7.634e-05  -5.485 4.24e-08 ***
availability_365                2.080e-04  2.860e-05   7.273 3.77e-13 ***
reviews_per_month              -9.347e-04  2.136e-03  -0.438   0.6616
calculated_host_listings_count  3.435e-03  3.417e-04  10.052  < 2e-16 ***
minimum_nights                 -2.221e-04  1.272e-04  -1.746   0.0809 .
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
Residual standard error: 0.3523 on 10593 degrees of freedom
Multiple R-squared:  0.3731,    Adjusted R-squared:  0.3726
F-statistic: 787.9 on 8 and 10593 DF,  p-value: < 2.2e-16
```

This model is an improvement. Median residual error is now -0.014, which is far better than -22.9 from the first model. R2=0.371 means that this model explains about 50% variance of target variable. Obviously we choose the second model for the prediction.

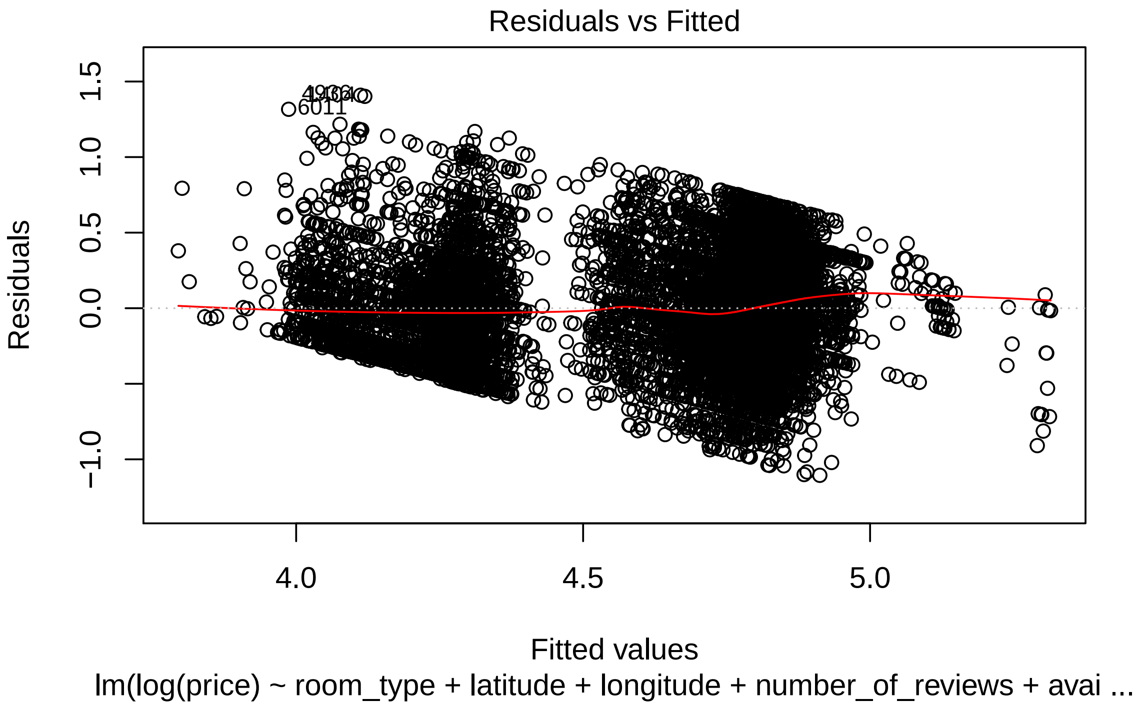


Figure 7: Number of Predictors vs Accuracy

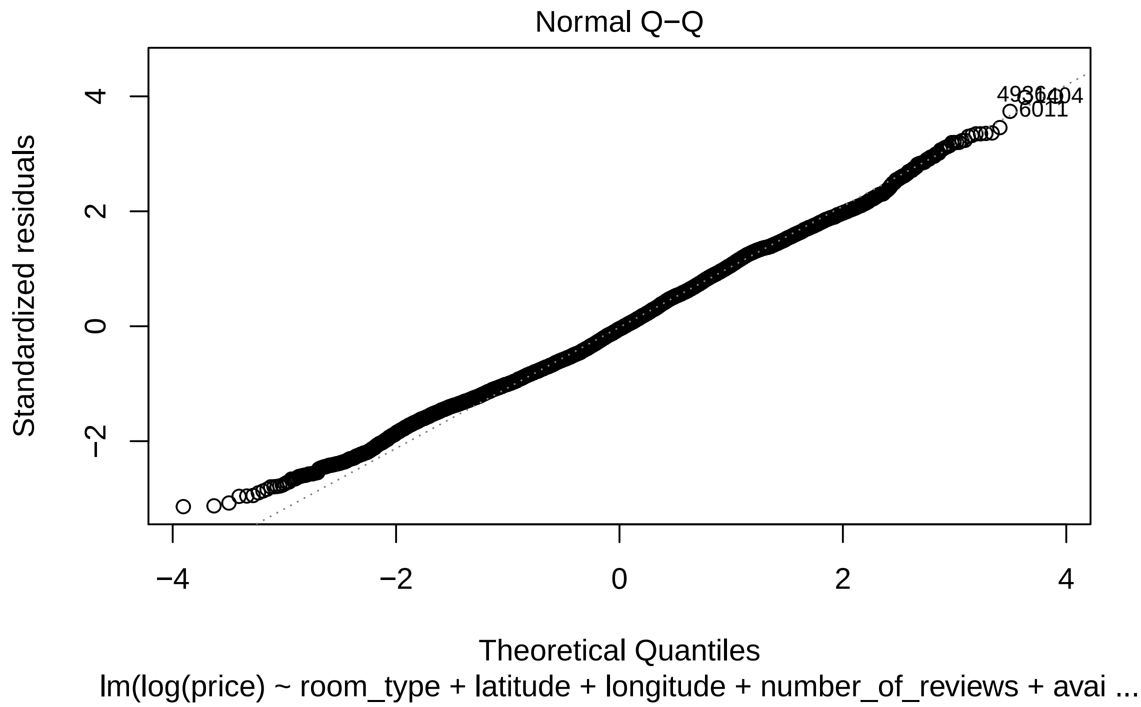


Figure 8: Number of Predictors vs Accuracy

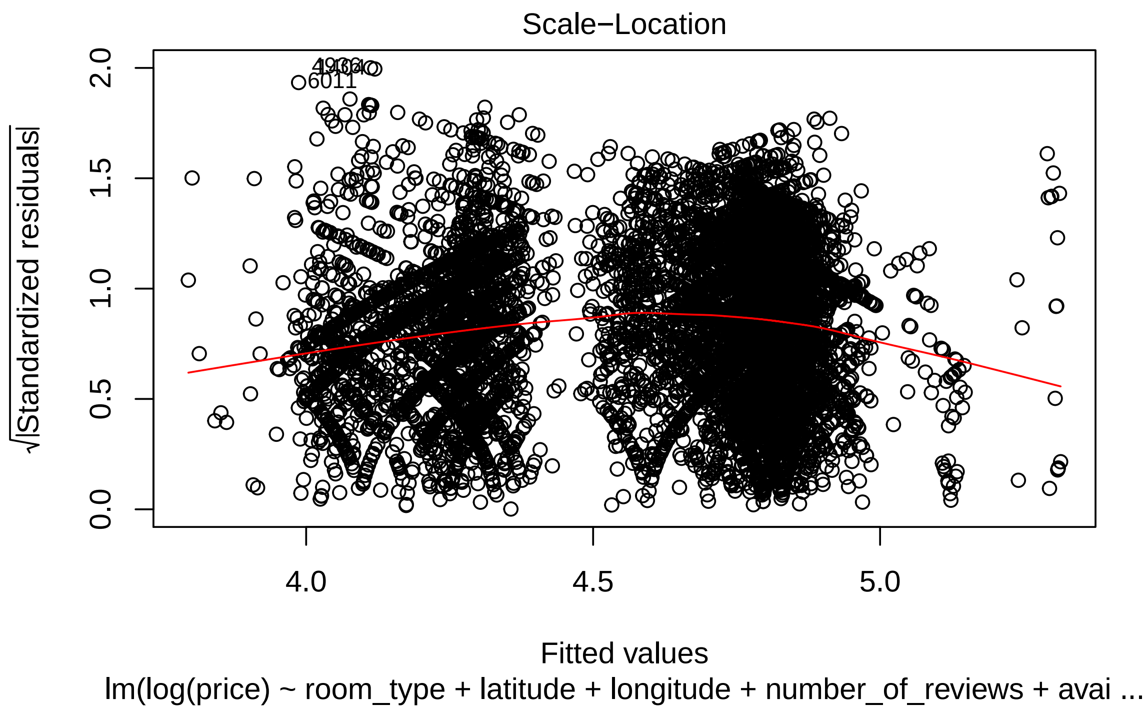


Figure 9: Number of Predictors vs Accuracy

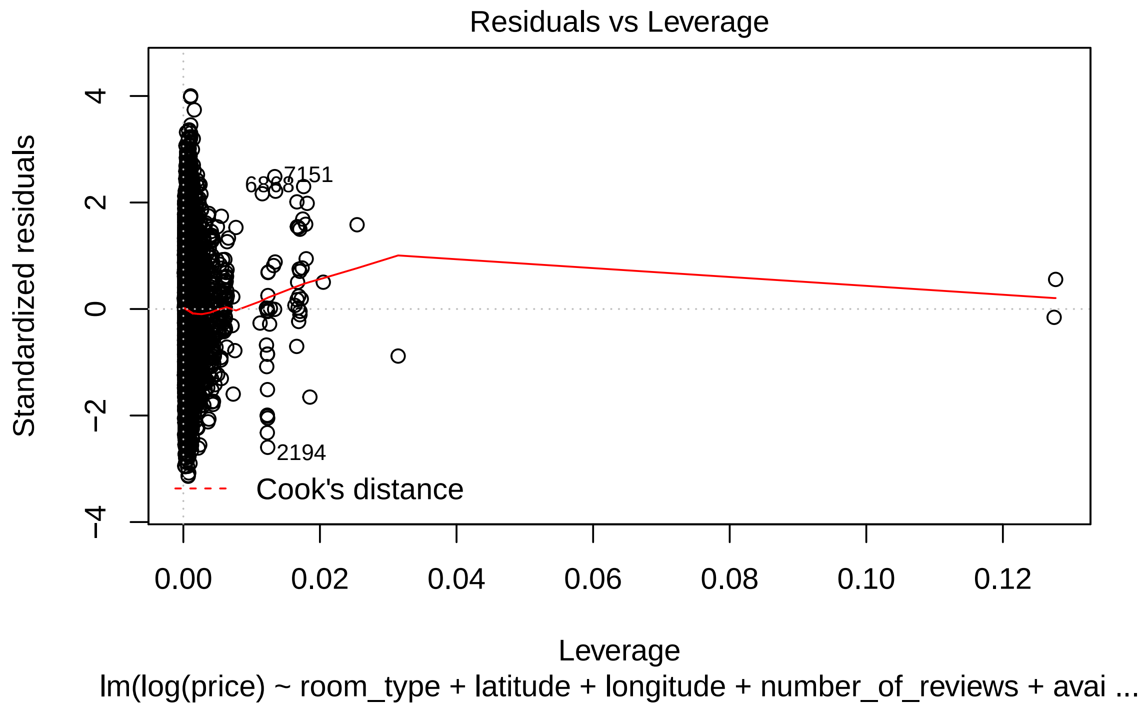


Figure 10: Number of Predictors vs Accuracy

##Predict prices for training set

```
[1] 44.81829
```

```
[1] 0.2648741
```

Undertanding the listing Employing Unsupervised Learning

In this section we will apply various clustering methods to cluster the listings in several groups. We will use Partitioning clustering and Hierarchical clustering approaches.

Before we apply clustering models to the dataset we should assess clustering tendency. In order to do so we will employ **Hopkins** statistics.

Hopkins Statistics

Hopkins statistic is used to assess the clustering tendency of a dataset by measuring the probability that a given dataset is generated by a uniform data distribution.(Ref: Jiawei Han (2012)). Let's calculate Hopkins (**H**) statistics for customerData1:

The **H** value close to one indicates very good clustering tendency. The **H** value around or greater than 0.5 denotes poor clustering tendency(Ref: Alboukadel Kassambara).

```
customerData2 = subset(customerData1,select = -id)
customerData2 = subset(customerData2,select = -name)
customerData2 = subset(customerData2,select = -host_id)
customerData2 = subset(customerData2,select = -host_name)
customerData2 = subset(customerData2,select = -neighbourhood)

customerData2 <- customerData2 %>%
mutate(
       latitude = scale(latitude),
```

```
        longitude = scale(longitude),
        price = scale(price),
        minimum_nights = scale(minimum_nights),
        number_of_reviews = scale(number_of_reviews),
        last_review = scale(last_review),
        reviews_per_month  = scale(reviews_per_month ),
        calculated_host_listings_count = scale(calculated_host_listings_count),
        availability_365 = scale(availability_365)
)
#summary(customerData2)
```

```
H =  get_clust_tendency(customerData2,n = 100, graph = F, seed = 6709)
print(H[["hopkins_stat"]])
```

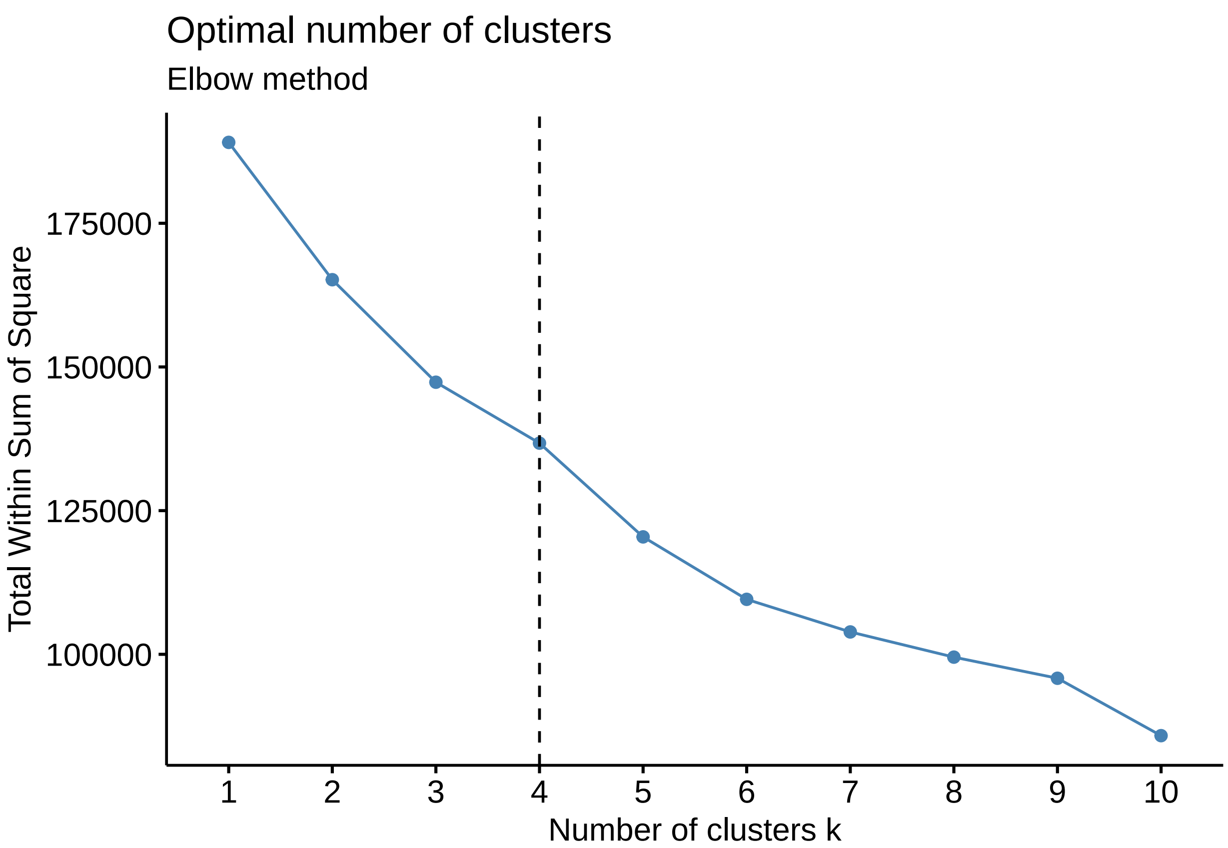
```
[1] 0.9809828
```

Perfect! H value is very close to 1. The dataset is clustrable.

Partitioning Clustering Approach

At first, we use Elbow method to get optimal number of clusters for k-means clustering:

```
set.seed(123)
# Elbow method
fviz_nbclust(customerData2, kmeans, method = "wss") +
    geom_vline(xintercept = 4, linetype = 2)+
  labs(subtitle = "Elbow method")
```

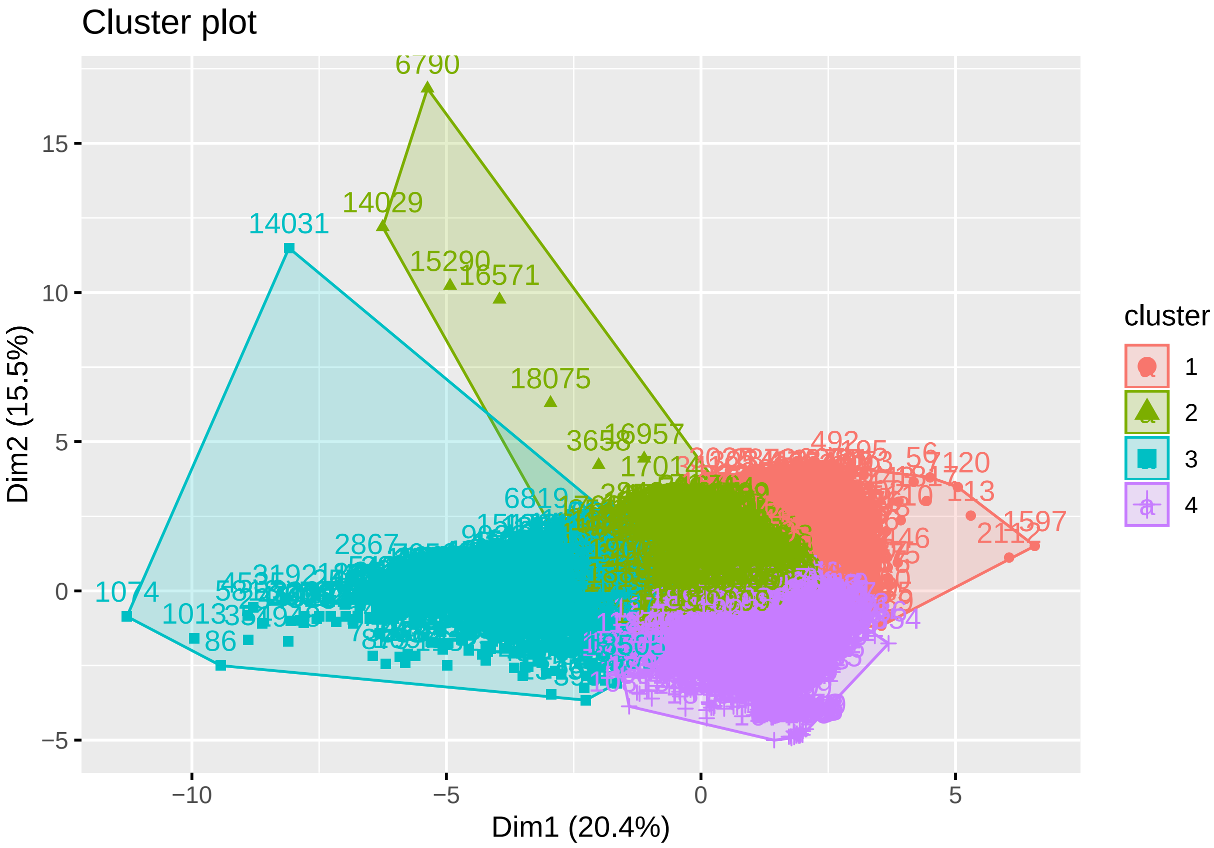


It seems that the optimal number of clusters is 4.

Let's use kmeans to cluster the dataset.

```
set.seed(123)
k2 <- kmeans(customerData2, centers = 4, nstart = 25)
```

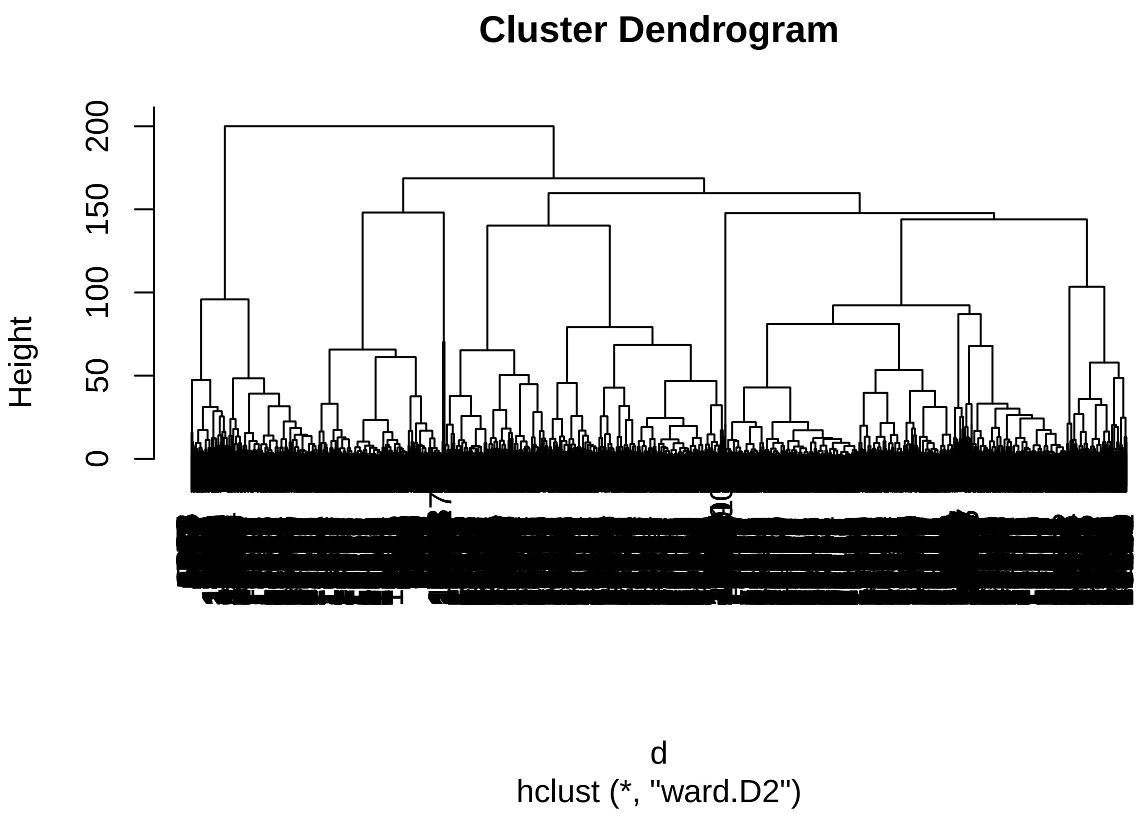
```
#k2
fviz_cluster(k2, data = customerData2)
```




```
group <- k2$cluster
customerData3 <- cbind(customerData1, group)
#write.csv(customerData3, "../shiny/www/mydata1.csv")
```

Hierarchical clustering approache

```
set.seed(123)
d <- dist(customerData2)
c <- hclust(d, method = 'ward.D2')

plot(c)
```



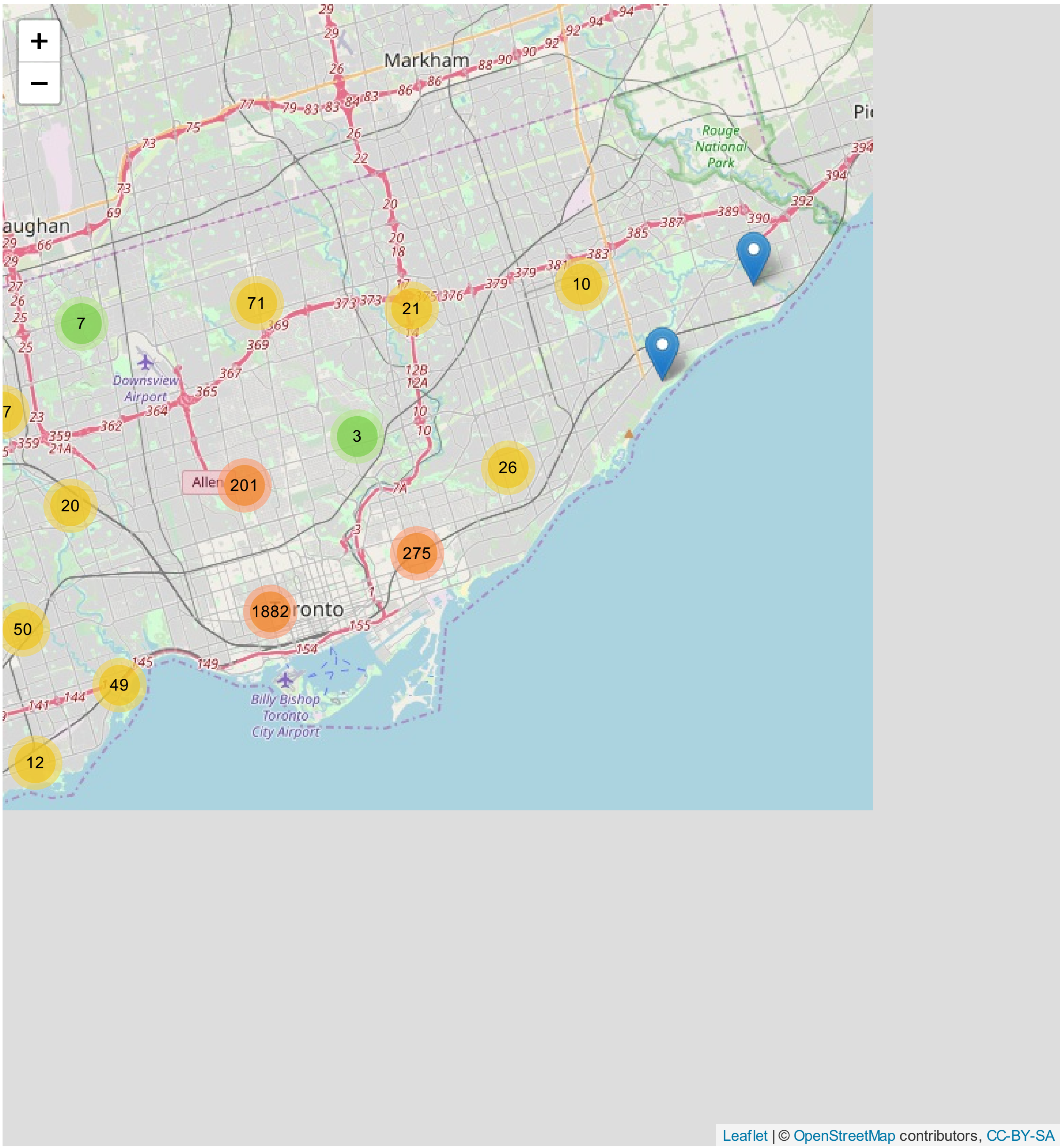

```
members <- cutree(c,k = 4)

table(members)
#members

members
   1    2    3    4
2572 8164 2579 5708
```

We decide to accept the kmeans cluster result. K-means clustering with 4 clusters of sizes group1 = 2649, group2 = 9925, group3 = 2800, group4 = 3649

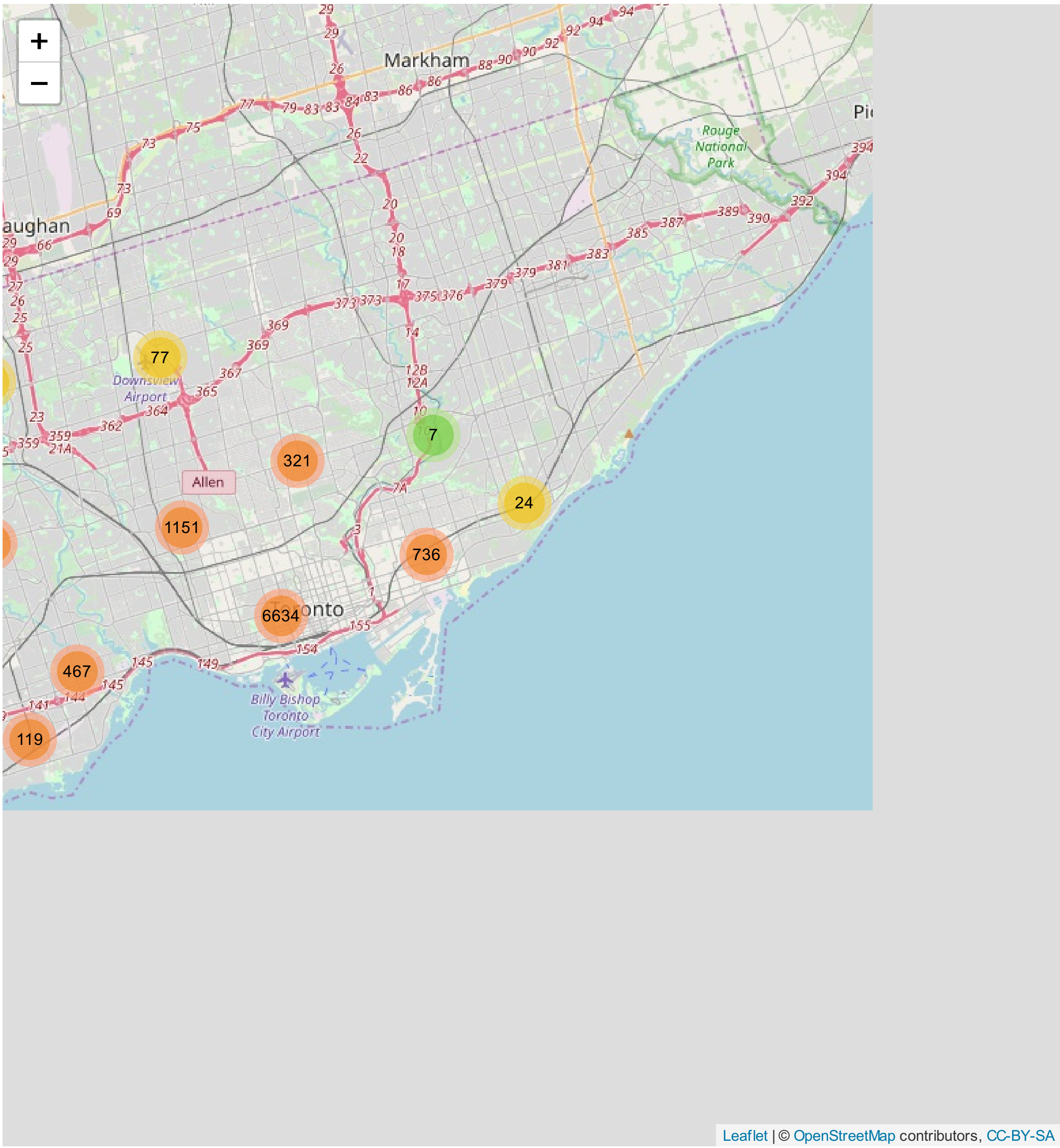
```
group1 <- customerData3 %>%
  filter(group == 1)

group2 <- customerData3 %>%
  filter(group == 2)

group3 <- customerData3 %>%
  filter(group == 3)

group4 <- customerData3 %>%
  filter(group == 4)
```

Let's dig out why the listings are clustered to 4 groups. At first, let's observe every group's geographic distribution.

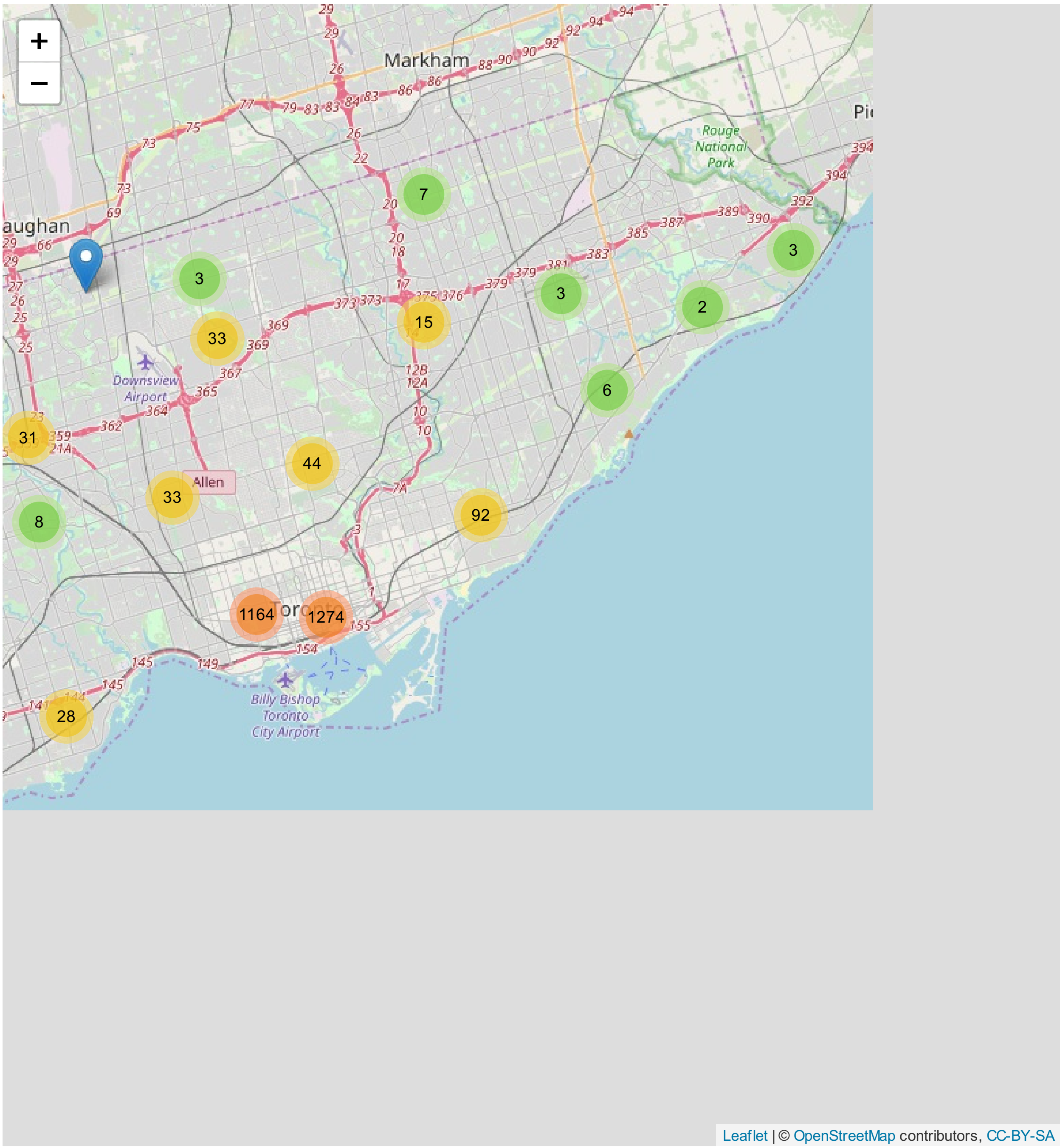
```
leaflet(group1 %>% select(longitude,latitude)) %>%
  setView(lng = -79.38, lat = 43.75,zoom = 11) %>%
   addTiles() %>%
  addMarkers(
  clusterOptions = markerClusterOptions())
```



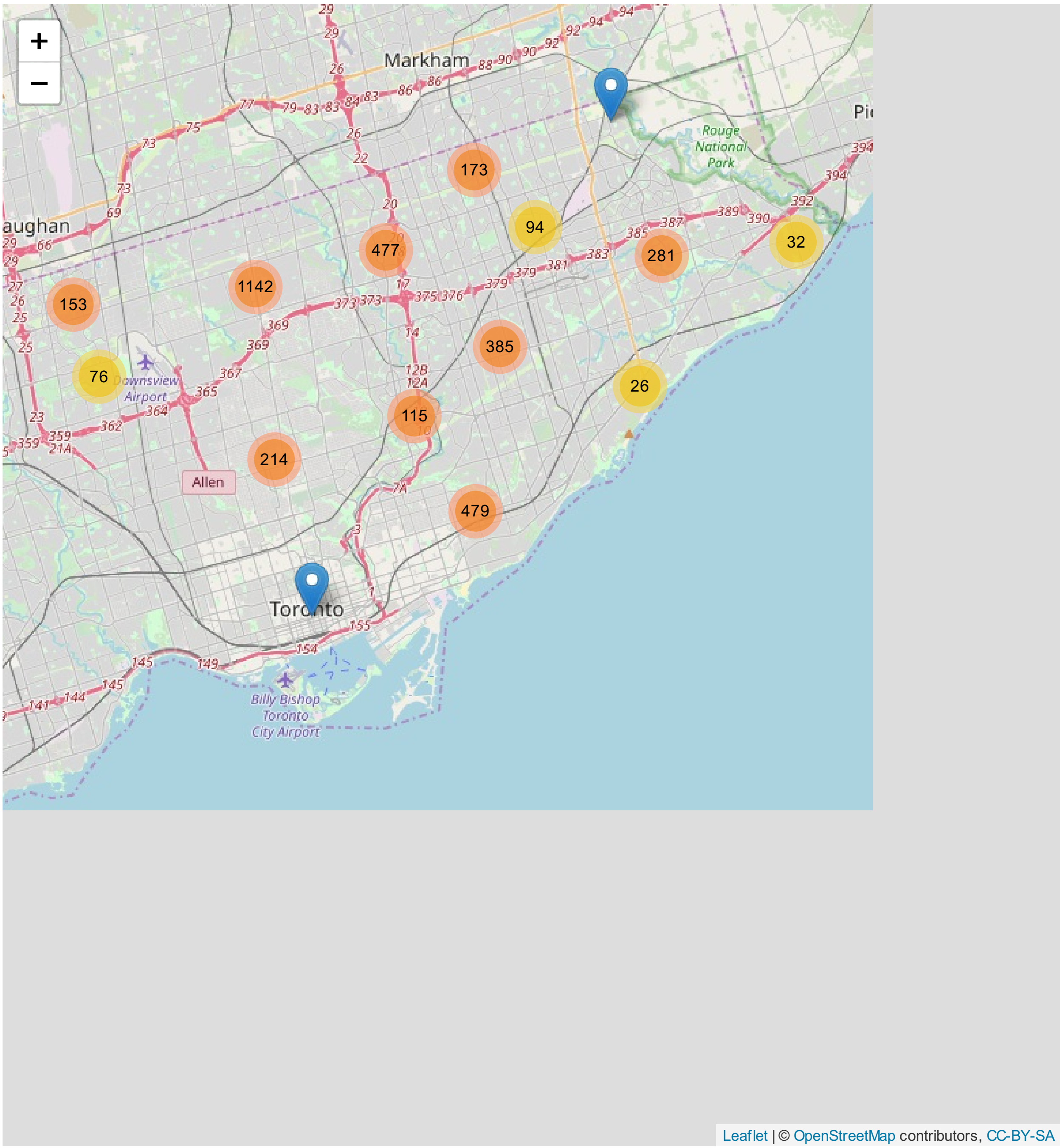

```
leaflet(group2 %>% select(longitude,latitude)) %>%
  setView(lng = -79.38, lat = 43.75,zoom = 11) %>%
   addTiles() %>%
  addMarkers(
  clusterOptions = markerClusterOptions())
```




```
leaflet(group3 %>% select(longitude,latitude)) %>%
  setView(lng = -79.38, lat = 43.75,zoom = 11) %>%
   addTiles() %>%
  addMarkers(
  clusterOptions = markerClusterOptions())
```



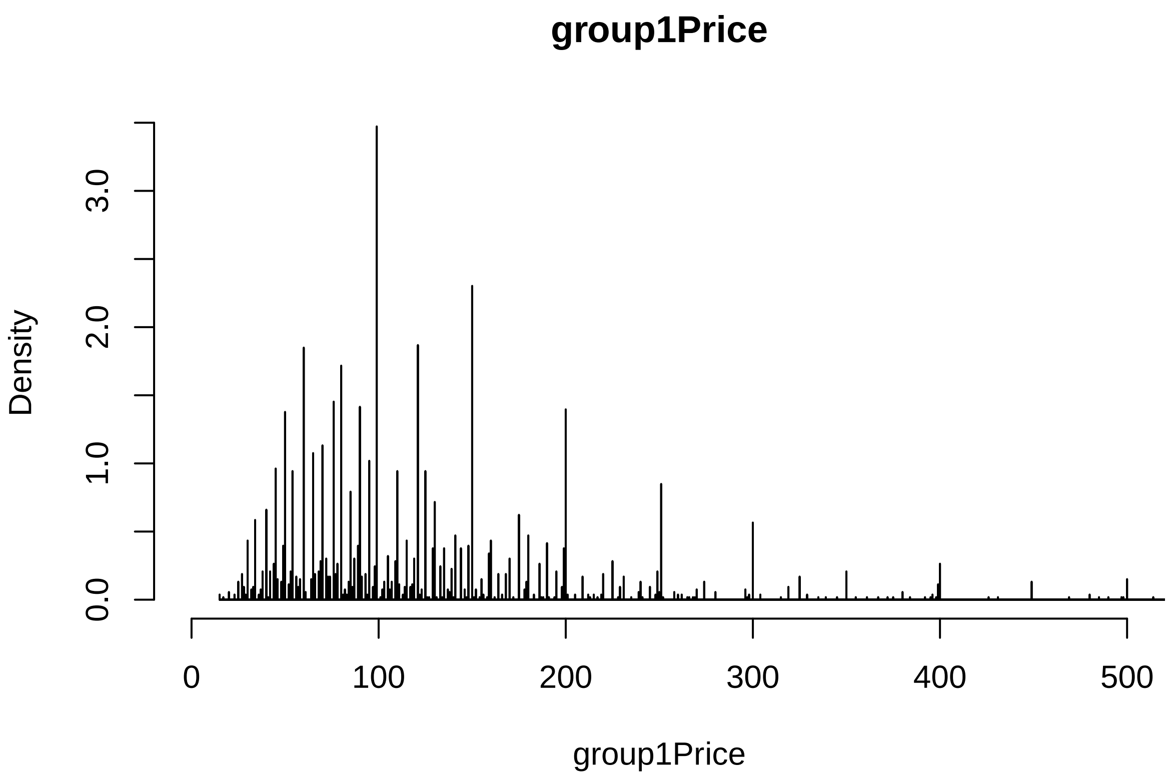

```
leaflet(group4 %>% select(longitude,latitude)) %>%
  setView(lng = -79.38, lat = 43.75,zoom = 11) %>%
   addTiles() %>%
  addMarkers(
  clusterOptions = markerClusterOptions())
```



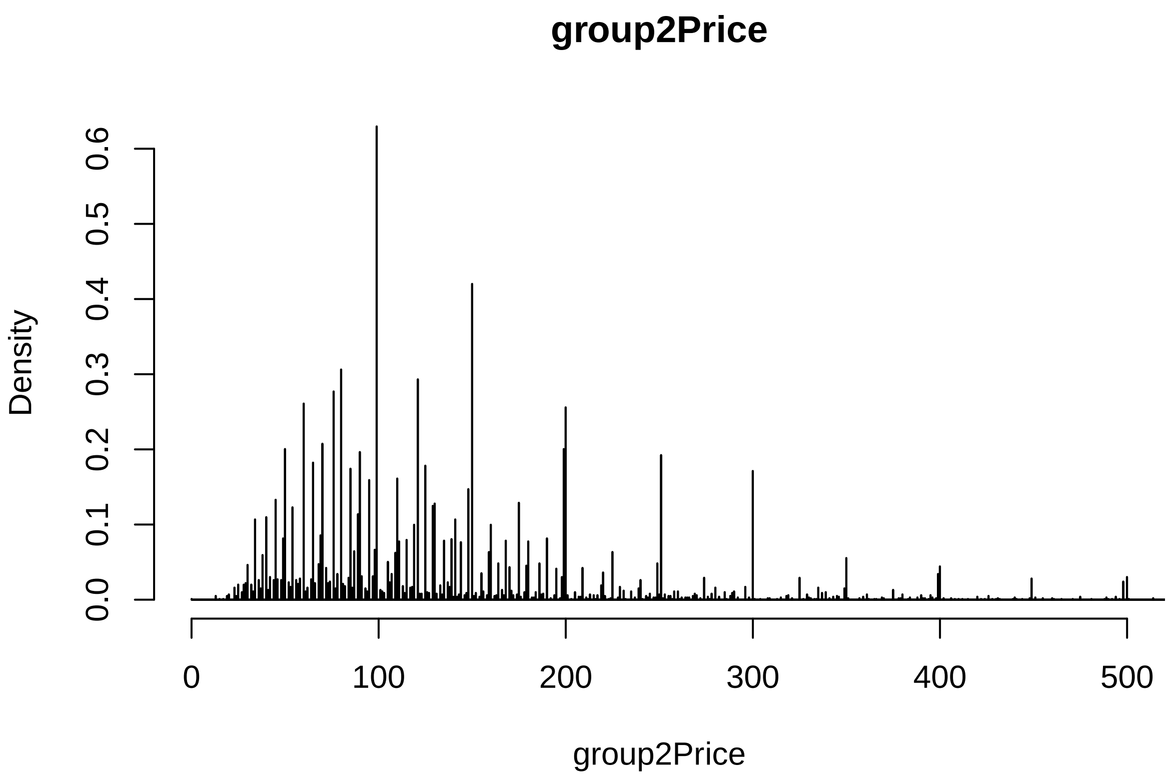
From the geographic distribution, we can observe that group4, most of the listings are far from downtown.

Let's observe every group's price distribution.

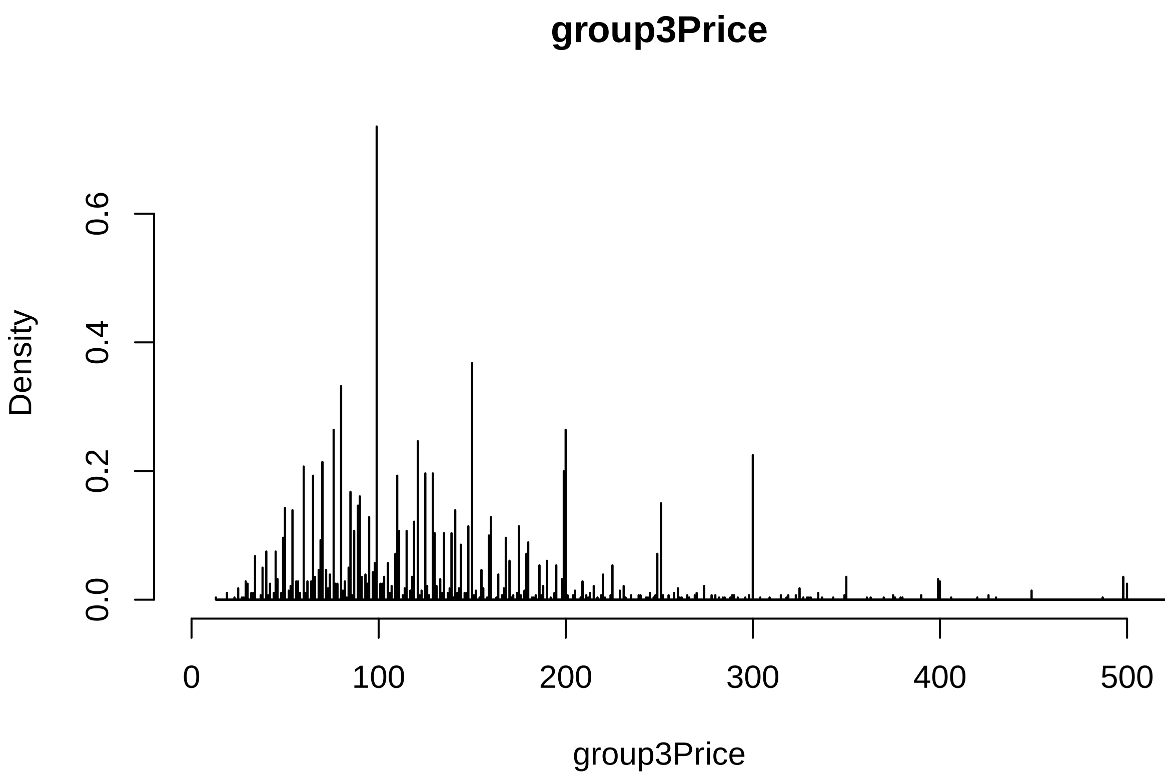
```
# histogram with added parameters
hist(group1$price,
main="group1Price",
xlab="group1Price",
breaks=100000,
xlim=c(0,500),
col="darkmagenta",
freq=FALSE
)
```



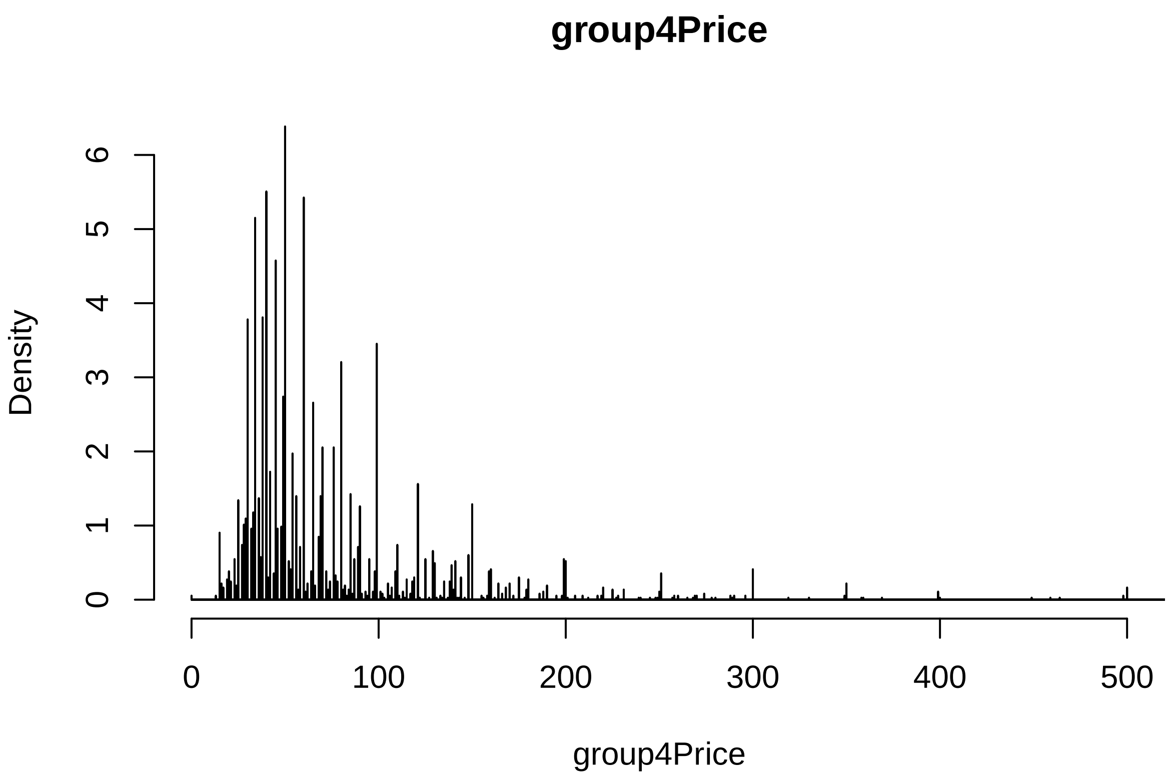

```
# histogram with added parameters
hist(group2$price,
main="group2Price",
xlab="group2Price",
breaks=100000,
xlim=c(0,500),
col="darkmagenta",
freq=FALSE
)
```




```
# histogram with added parameters
hist(group3$price,
main="group3Price",
xlab="group3Price",
breaks=100000,
xlim=c(0,500),
col="darkmagenta",
freq=FALSE
)
```




```
# histogram with added parameters
hist(group4$price,
main="group4Price",
xlab="group4Price",
breaks=100000,
xlim=c(0,500),
col="darkmagenta",
freq=FALSE
)
```



Obviously in group4, The vast majority of rental prices are concentrated below 100$.

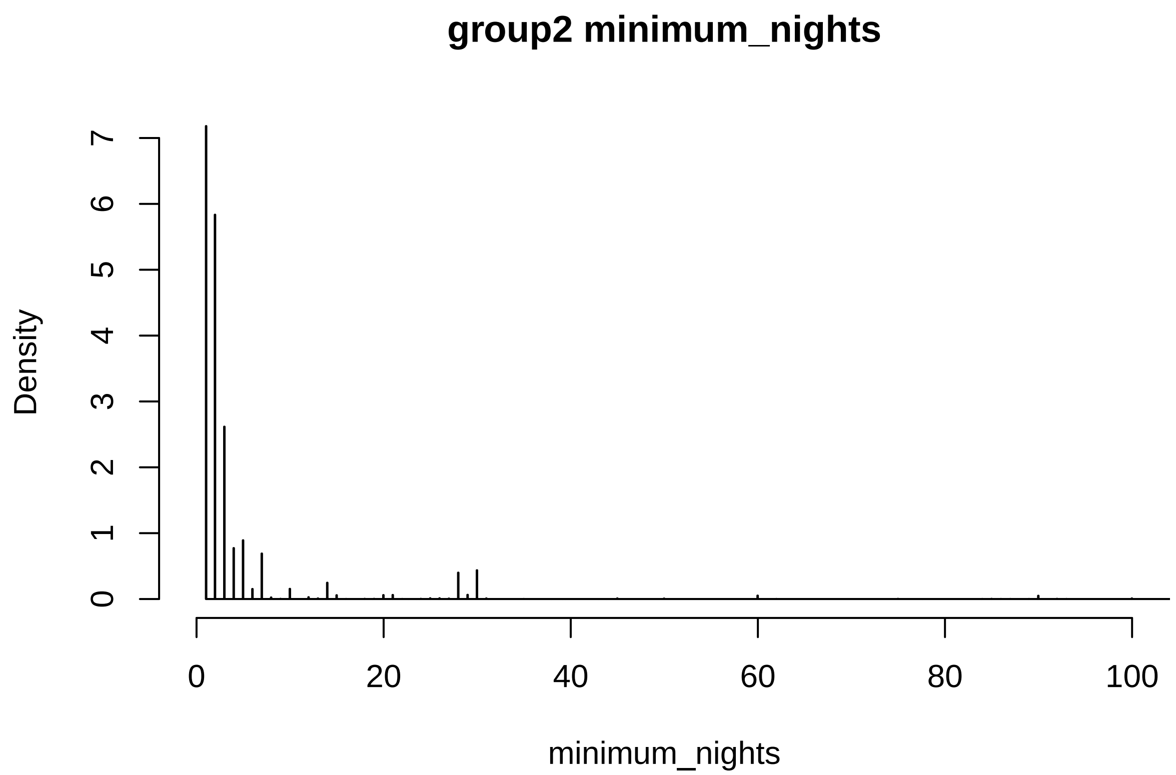
Let's list every group's median price.

```
median(group1$price)

[1] 99

median(group2$price)

[1] 117

median(group3$price)

[1] 117

median(group4$price)

[1] 54
```

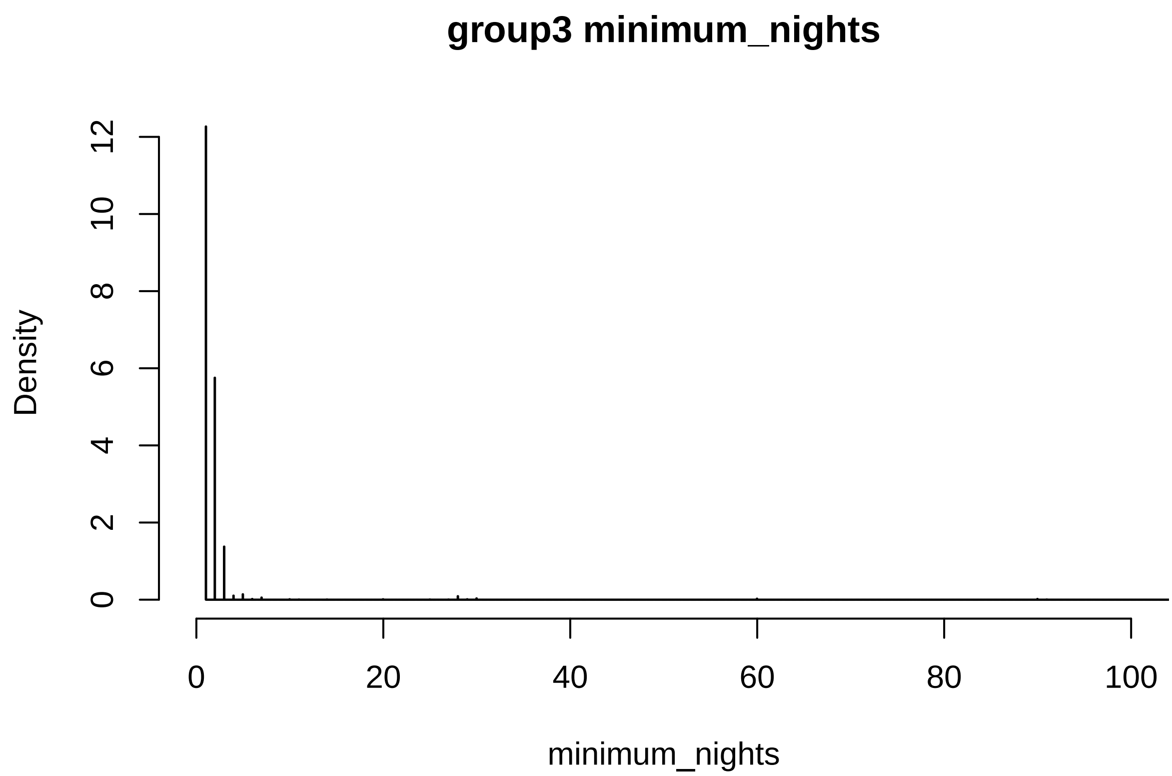
In group1, the rental price is median. Group4 is the cheapest. Group2 & Group3 has the same median price.

We're interested in what's the difference between group2 and group3. Let's dig it further.

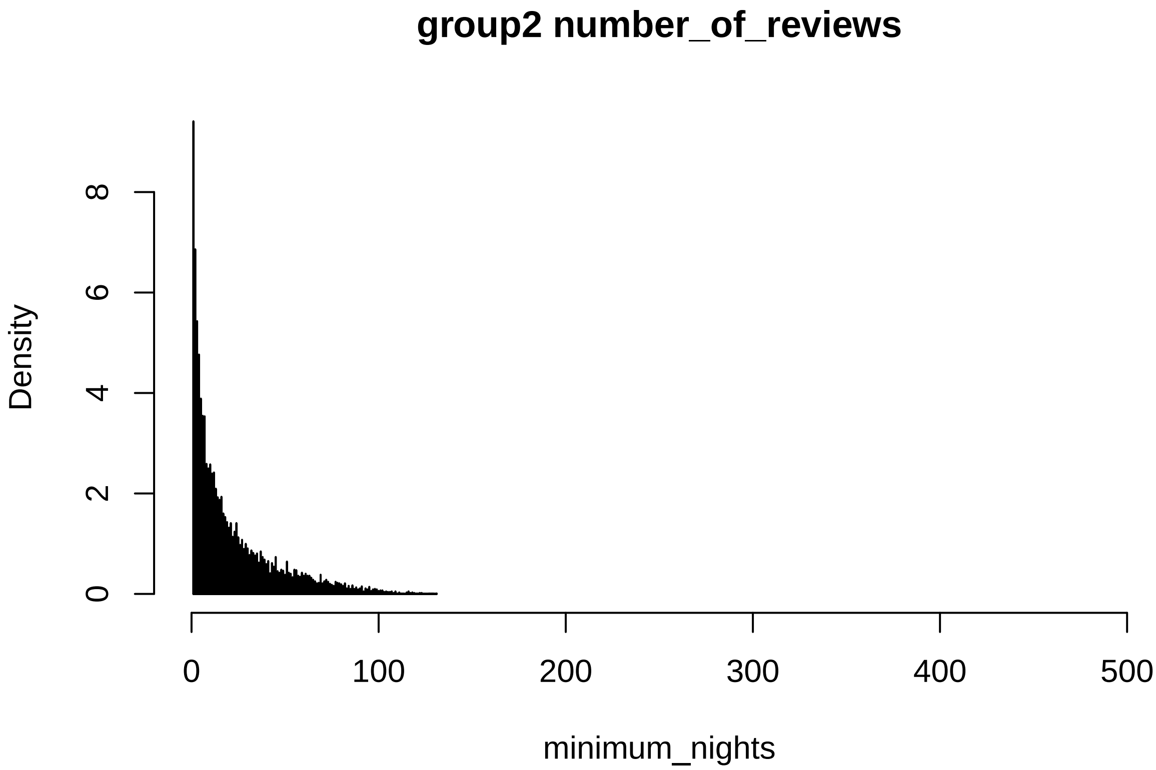
```
# histogram with added parameters
hist(group2$minimum_nights,
main="group2 minimum_nights",
xlab="minimum_nights",
breaks=10000,
xlim=c(0,100),
col="darkmagenta",
freq=FALSE
)
```



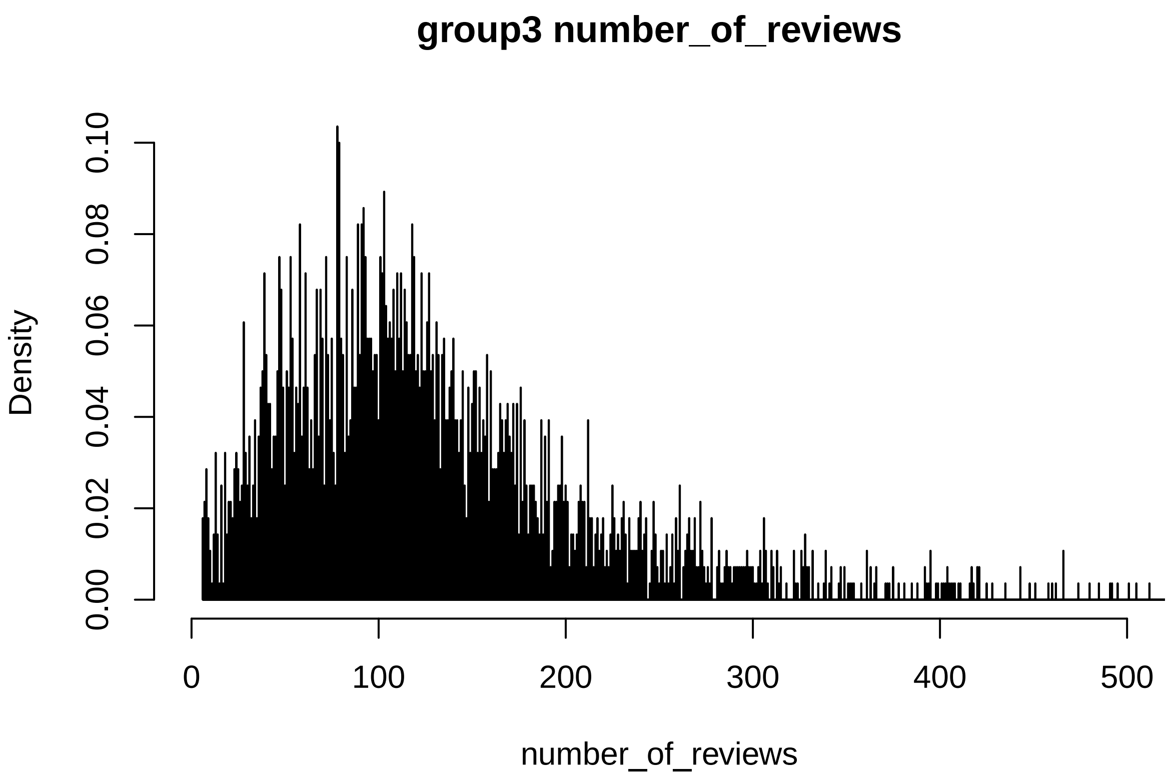

```
# histogram with added parameters
hist(group3$minimum_nights,
main="group3 minimum_nights",
xlab="minimum_nights",
breaks=10000,
xlim=c(0,100),
col="darkmagenta",
freq=FALSE
)
```




```
# histogram with added parameters
hist(group2$number_of_reviews,
main="group2 number_of_reviews",
xlab="minimum_nights",
breaks=10000,
xlim=c(0,500),
col="darkmagenta",
freq=FALSE
)
```

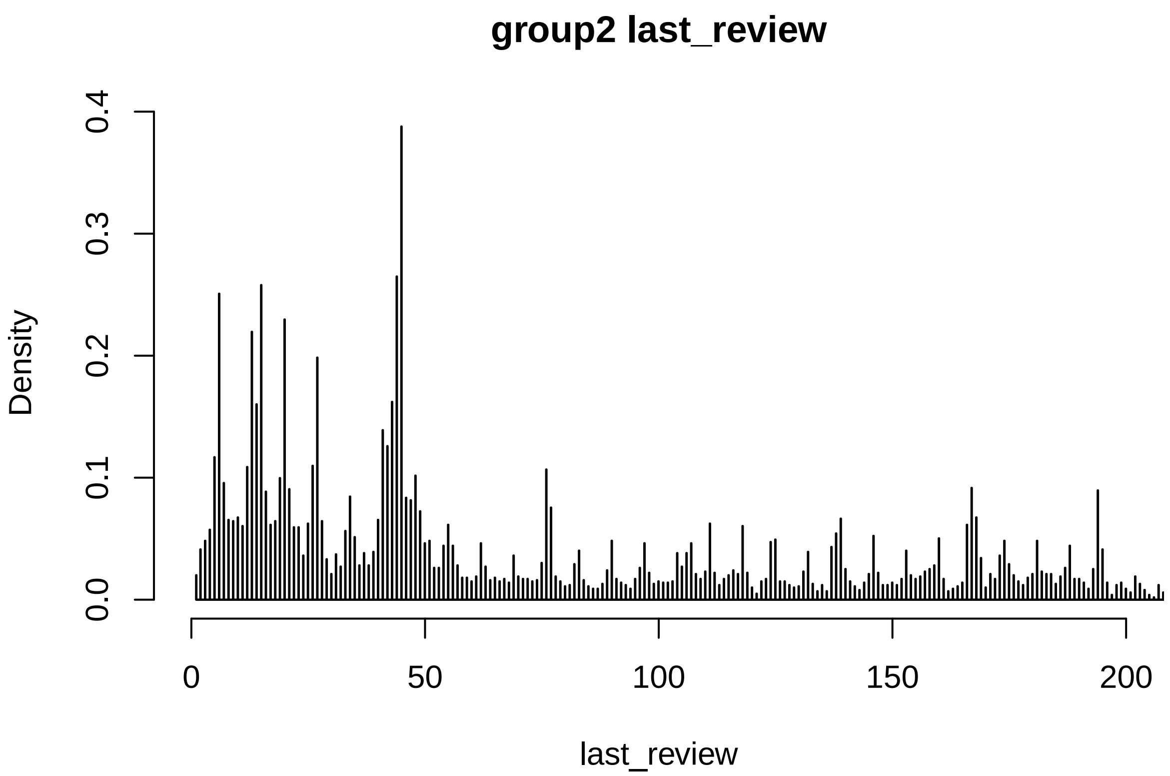



```
# histogram with added parameters
hist(group3$number_of_reviews,
main="group3 number_of_reviews",
xlab="number_of_reviews",
breaks=10000,
xlim=c(0,500),
col="darkmagenta",
freq=FALSE
)
```

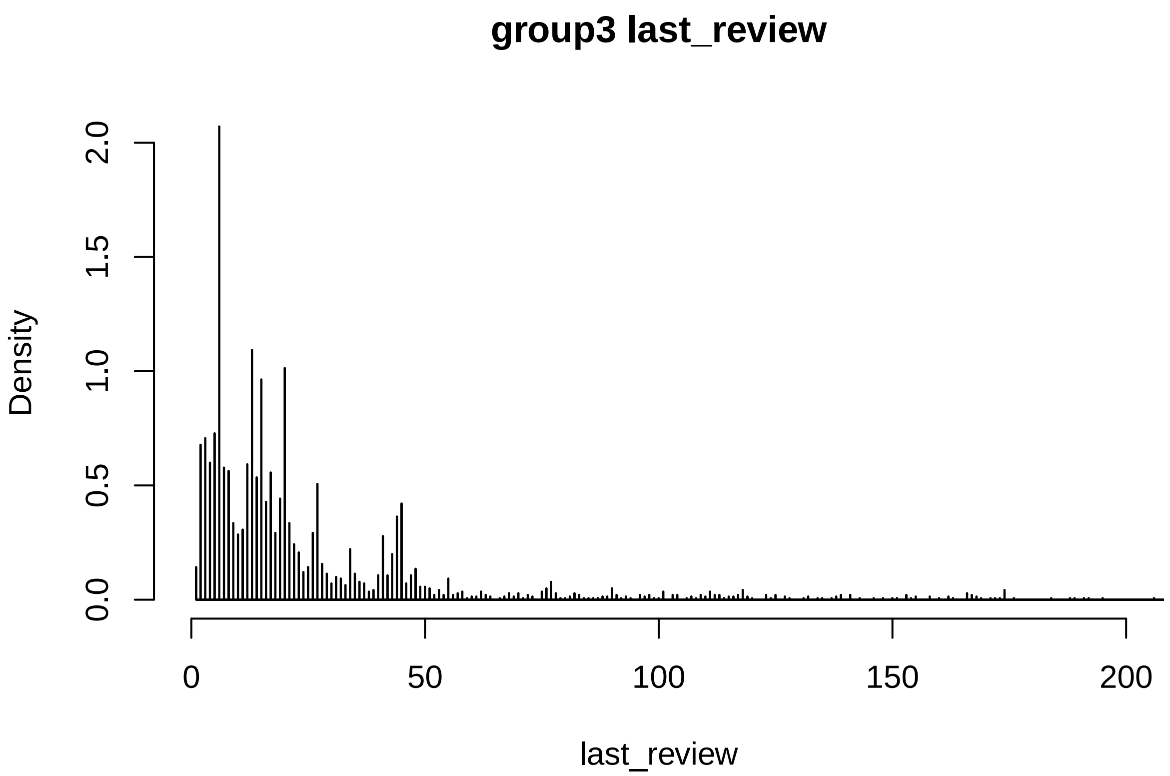


It seems that the group3 has more reviews that group2.

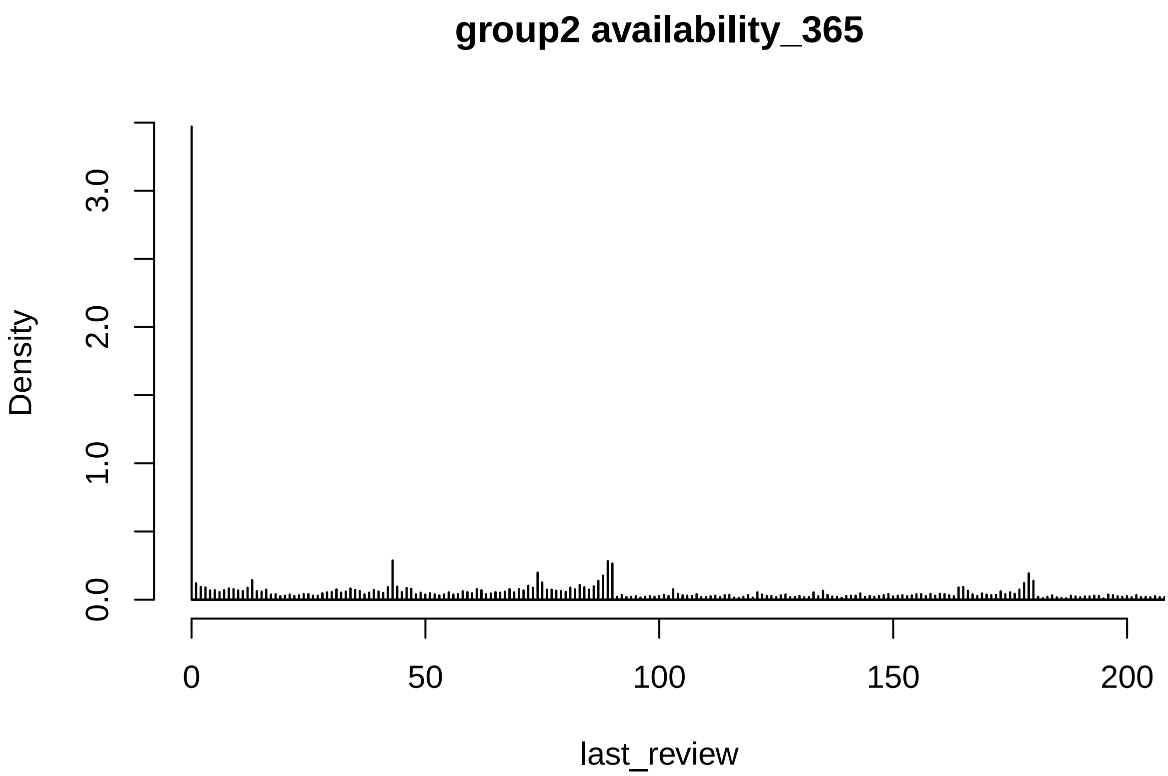
```
# histogram with added parameters
hist(group2$last_review,
main="group2 last_review",
xlab="last_review",
breaks=10000,
xlim=c(0,200),
col="darkmagenta",
freq=FALSE
)
```



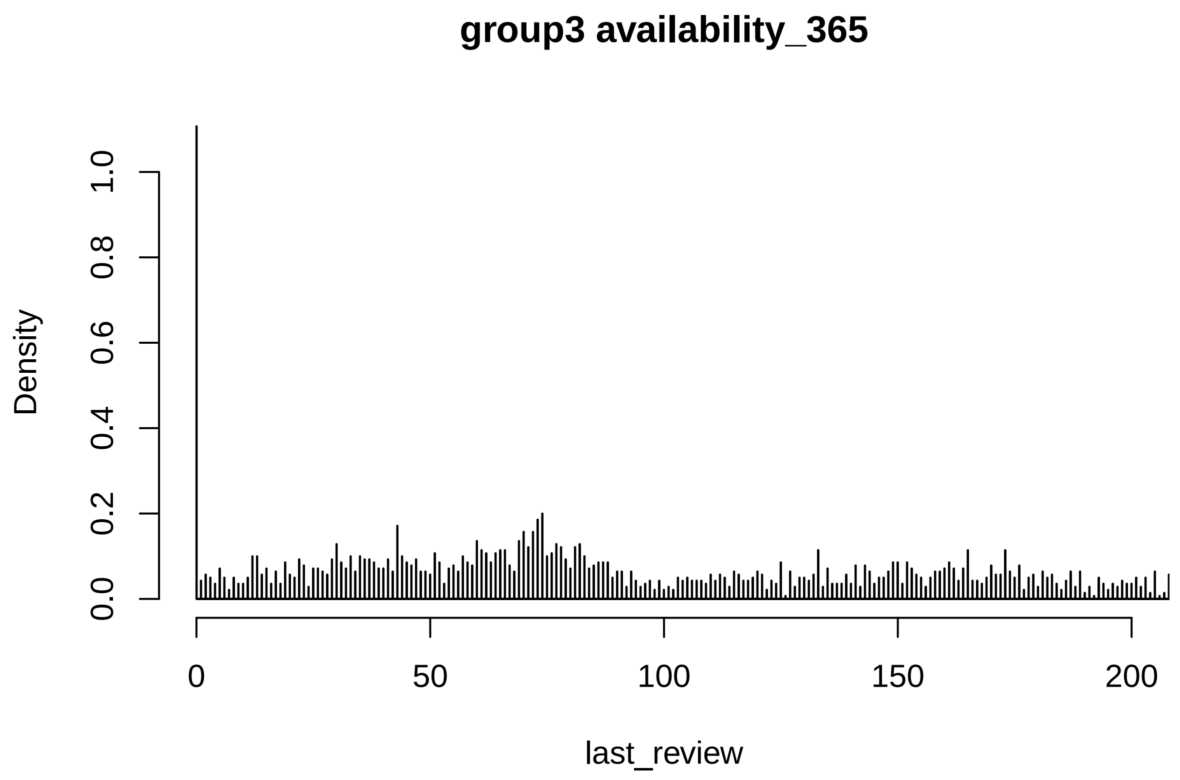

```
# histogram with added parameters
hist(group3$last_review,
main="group3 last_review",
xlab="last_review",
breaks=10000,
xlim=c(0,200),
col="darkmagenta",
freq=FALSE
)
```




```
# histogram with added parameters
hist(group2$availability_365,
main="group2 availability_365",
xlab="last_review",
breaks=10000,
xlim=c(0,200),
col="darkmagenta",
freq=FALSE
)
```




```
# histogram with added parameters
hist(group3$availability_365,
main="group3 availability_365",
xlab="last_review",
breaks=10000,
xlim=c(0,200),
col="darkmagenta",
freq=FALSE
)
```



It seems that the group3 has more availability_365 that group2. Let's summarize every group's characteristic.

Group1: the rental rate is median. Group2: the rental rate is high, and the attention is low because there are less reviews in this group.Since most of listings in this group are located in downtown,they have Convenient transportation. Group3: the rental rate is high, and the attention is high because there are more reviews in this group.Since most of listings in this group are located in downtown,they have Convenient transportation. Group4: the rental rate is low.

Some potential business value: Airbnb can recommeded the listings in group 4 for the customers that have high price sensetivity. The listings in group2 can be recommended to the customers that prefer the convenient transportation and don't mind the attention. The listings in group3 can be recommended to the customers that prefer the convenient transportation and the high attention. The listings in group4 can be recommended to other customers.

Model Deployment

In our second regression model, metrics for testing set: R2=0.285 and RMSE=42.18. Is it good enough? It is hard to tell. We feel that the model meets our first objective, which is prediction of toronto airbnb rental rate. The second logistic regression model we picked is more accurate. The model is fast and easy to deploy. Due to the nature of the business the model does require frequent data updates and re-training. We are also very satisfied with accomplishment of our second goal, which is understanding of the listing. We believe that clustering make sense for business.

Conclusion

Through exploring toronto airbnb listings dataset collected in 2020 we were able to come up with two models. One is a linear regression model,that predicts the property owner's rental rate. The second model provides in-depth view of the listings.

The study started with thorough analysis of the data set. At this phase we were able to identified many interesting patterns that insured the success of the whole project. We commenced our research providing descriptive stats on all available features of the data set.

Then we applied and evaluated two supervised learning algorithms: two kinds of Logistic Regression.

Lastly we applied unsupervised learning to understand how to cluster the listings in groups and find the potential business value.

As a result of this study we fully understood the data we dealt with. We designed reasonably accurate rental rate prediction model. We managed to group the listings into meaningful, highly interpretable clusters that explain the property's characteristics well.

Overall we believe we have achieved all our goals.

Bibliography

F. M. Alboukadel Kassambara. Extract and visualize the results of multivariate data analyses. URL https://cran.r-project.org/web/packages/factoextra/factoextra.pdf. [p27]

J. P. Jiawei Han, Micheline Kamber. *Data Mining. Concepts and Techniques*. The Morgan Kaufmann Series in Data Management Systems, 225 Wyman Street, Waltham, MA 02451, USA, 2012. ISBN 978-0-12-381479-1. [p1, 27]

Note from the Authors

This file was generated using *The R Journal* style article template, additional information on how to prepare articles for submission is here - Instructions for Authors. The article itself is an executable R Markdown file that could be downloaded from Github with all the necessary artifacts.

Ketao Li
York University

liketao@yahoo.com

Kush Halani
York University

kush.halani@ontariotechu.net

Josue Romain
York University

josue.rolland.romain@gmail.com

Juan Peña
York University

jppena62@my.yorku.ca