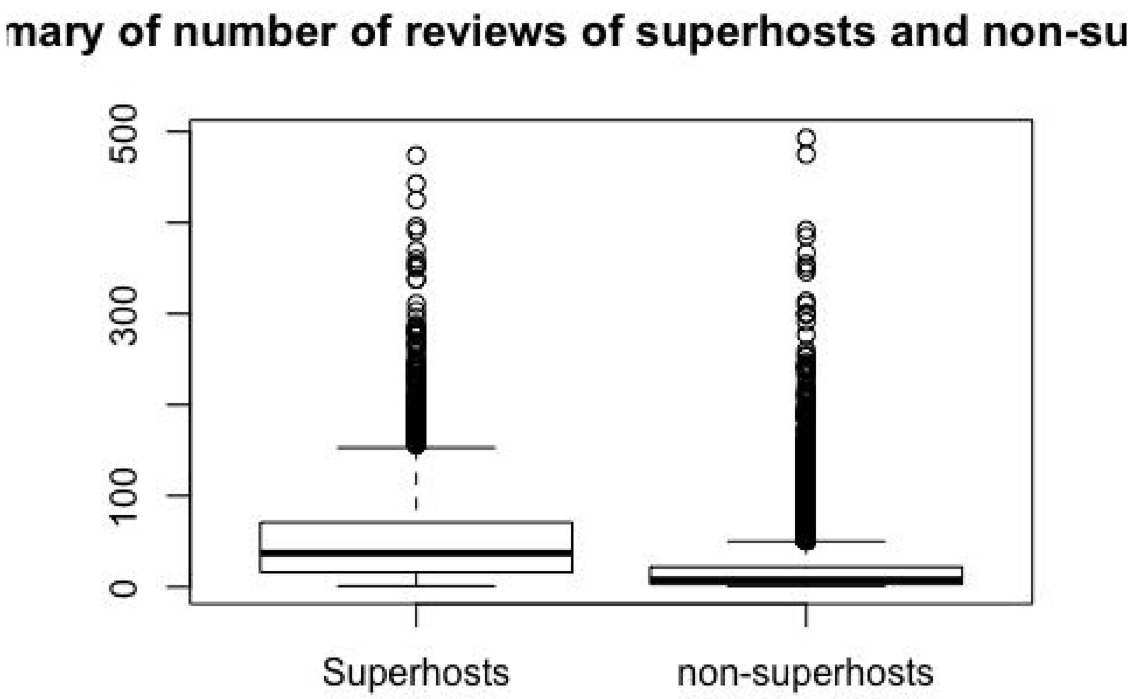
Business suggestions we can make based on our findings:

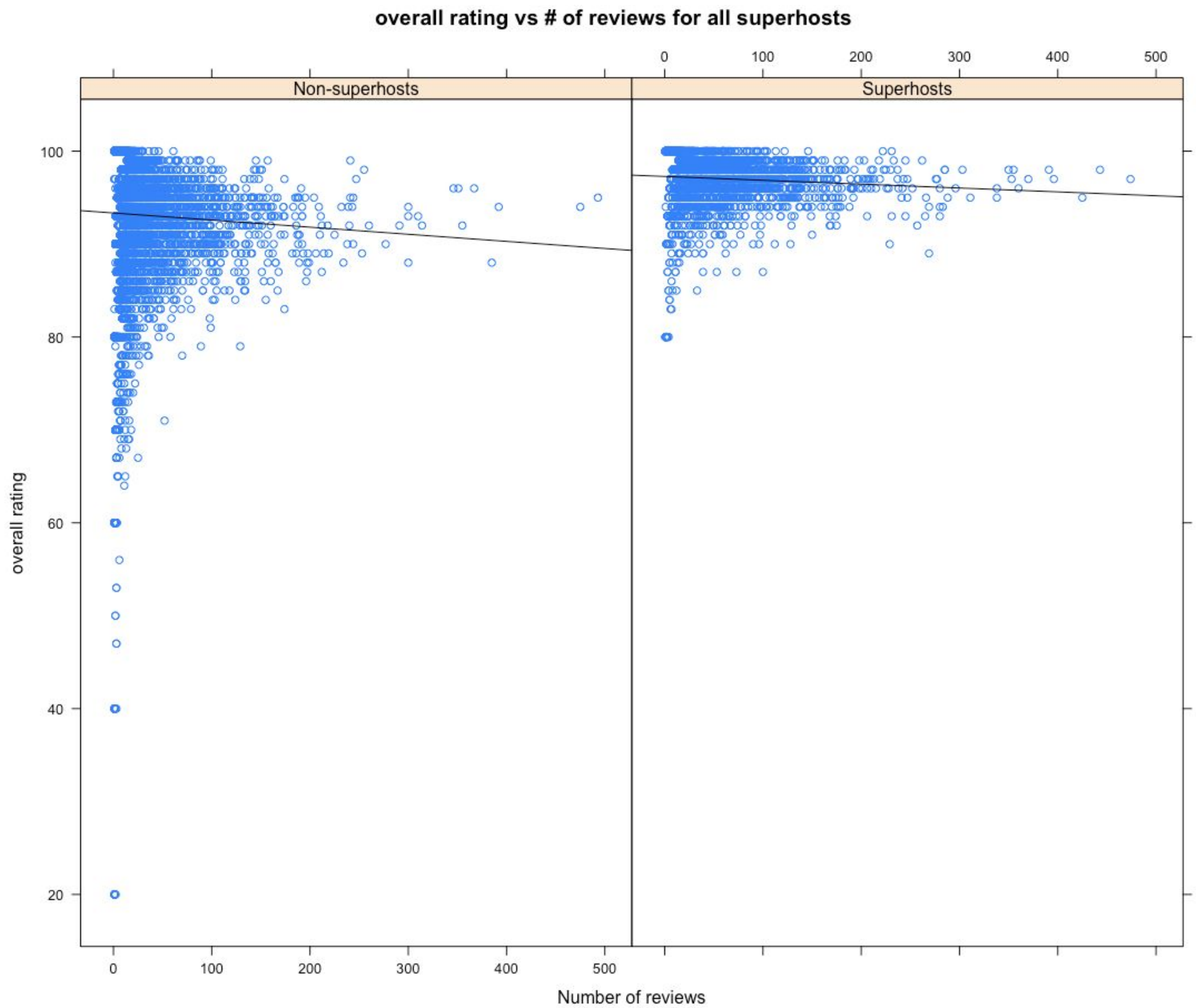
1. Being a superhost is very important.
    a. It has effect on boosting number of reviews. I choose the visualize the relationship between being a superhost or not and the overall rating by a boxplot, since there is one numeric variable and one categorical variable



We can tell that being a superhost or not does affect the number of reviews. The interquartile range and the median of number of reviews of superhosts are all higher than those of non-superhosts.

b. It has effects on overall ratings as well

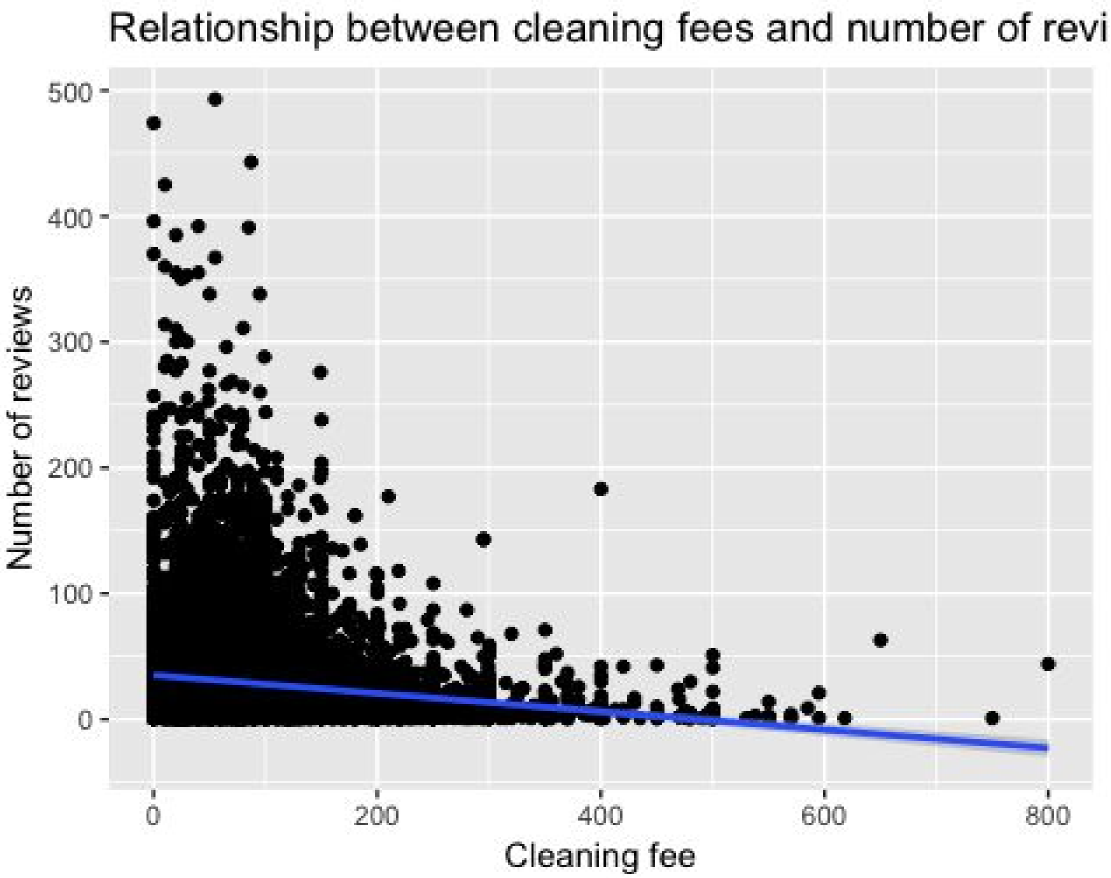
   For superhosts, the minimum rating is much lower. In terms of the regression line, we can see that the rating of superhosts' listing is higher than the rating of non-superhosts' rating for every level of number of reviews. Which means that superhosts generally get more reviews and higher ratings.



   Beside visualizing the relationships, I also discover the summary of overall ratings and being a superhost or not. It turns out the being a superhost does have a average score of 4 higher, which is 97, than not being a superhost, which is around 93.

2. Canceling or lowering the cleaning fee can generally earn you higher overall ratings

After discovering the distribution of cleaning fees, I discovered that some listings have ridiculously high cleaning fee. I started to wonder if there is a relationship between cleaning fee and number of reviews. And I chose to visualize such relationship.

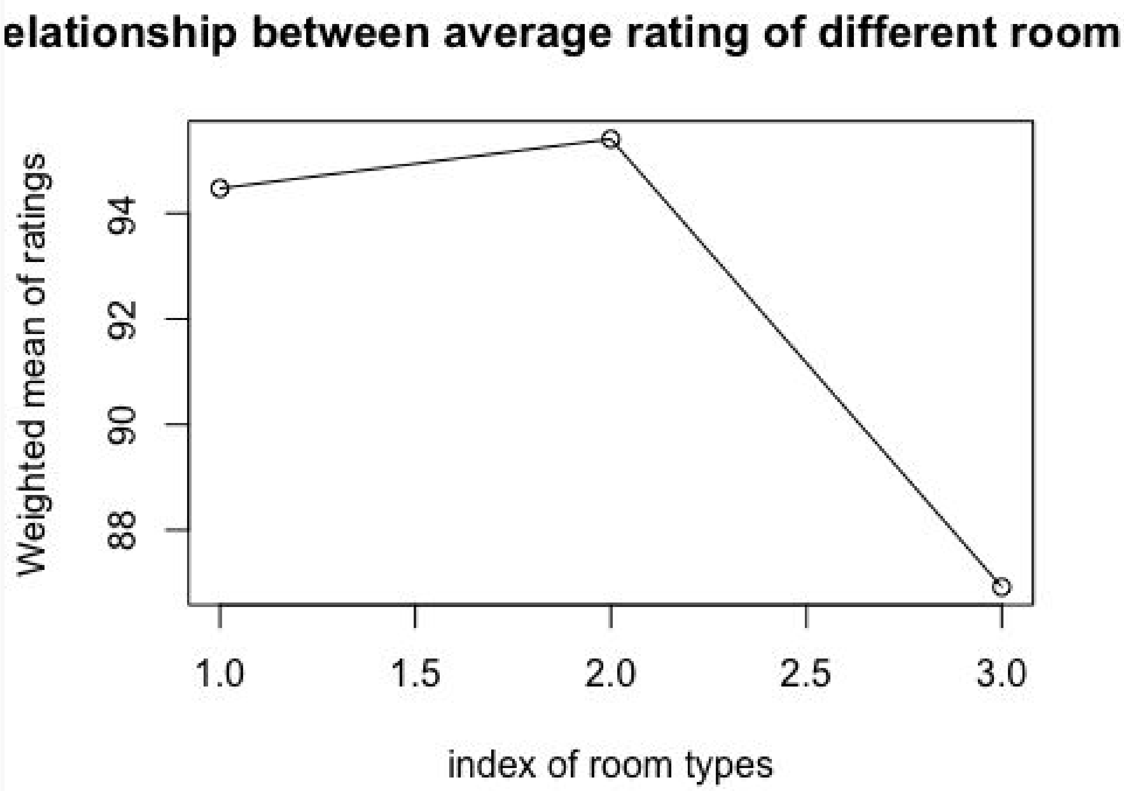


From the regression line, I can tell that once the listing has a cleaning fee, the higher the fee, the less of the number of reviews we generally get.

There are also a lot of listings without a cleaning fee. So I also compare the average number of reviews for those that don't contain the cleaning fee and those who do. It turns out that those that don't contain a cleaning fee has a 10 more reviews averagely.

3. Hosting private spaces (entire room or private rooms) can earn you higher ratings

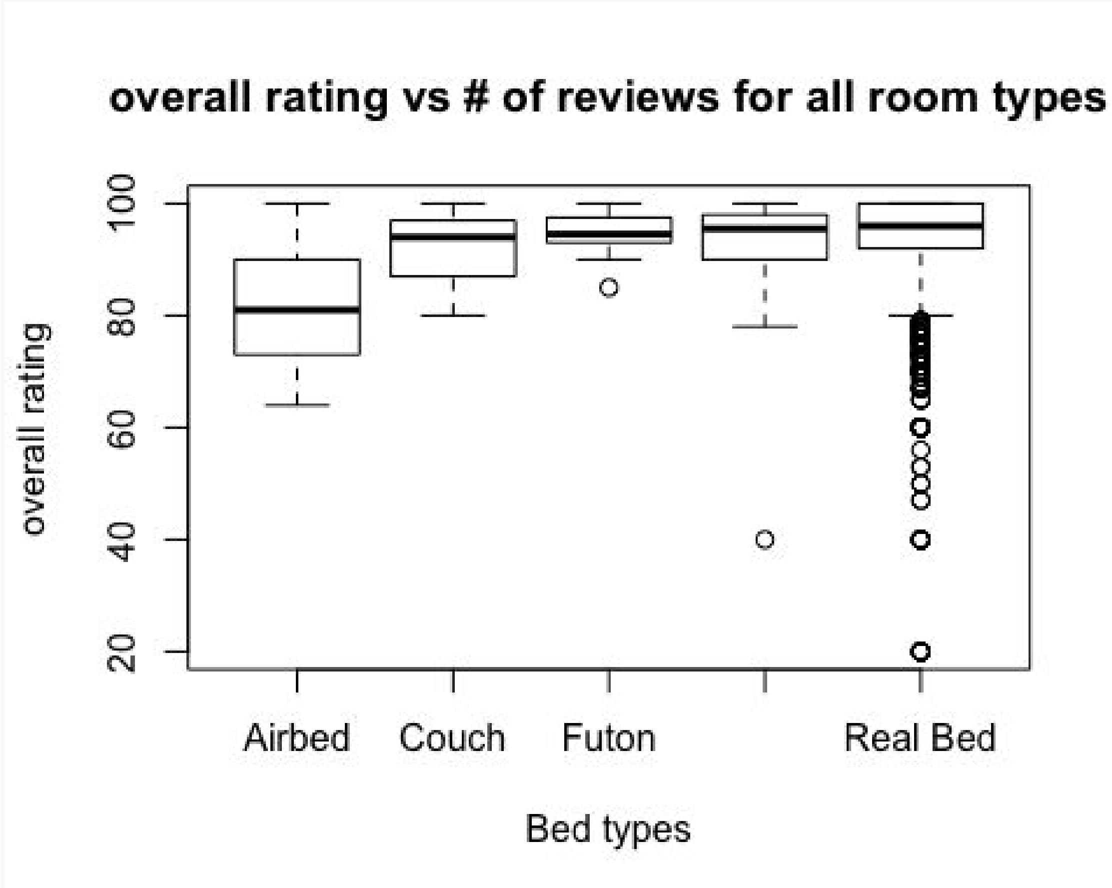
   We first discovered that containing the word "private" or not will actually affect the weighted mean of the overall rating of listings. From our observations, listings with the word "private" in their descriptions typically have 0.1 higher weighted mean. Although this doesn't seem impressive, we choose to discover the relationship between room types and average rating, since some room types provide private spaces and some don't



From this graph, each index one denotes the entire apartments, 2 means private rooms, and 3 means shared room. It is obvious to see that the weighted mean of overall ratings of private spaces are higher than the shared room's

4. Having real bed is better for overall rating

After exploring different types of beds, we realize that there is a huge difference between the numbers of different type of beds, as most of them are real beds, which means that people have preferences over beds, as the supply is correlated with the demand. We then explore the relationship between bed types and rating by a box plot since box plot is for the relationship between a numeric variable and a categorical variable.



From this graph we can tell that all the other beds except for airbeds have higher median and interquartile range. Therefore it is the best to have real beds, since it has the highest median and first quartile.

5. Hosting listings with parking lots can earn you higher ratings

We made this conclusion by first discovering how the word “parking” in description might affect the overall ratings. It does affect. The average of rating with “parking” is 94.77, and without is 94.86.

We then connect this thought to the amenities. How will the listings’ amenities containing “parking” affect the overall rating？ The result is as follows：

From this graph, the index of 1 denotes listings of having parking in its amenities, 2 means not. And we can see that people prefer listings with parkings.

6. You can demand higher prices if your listing is near the beach, or it is large, or has a garden, or at a popular (noisy) area

   By analyzing the change in average prices containing the keywords of “beach”, “large”, “garden” and “quiet”. We noticed that the listings containing keywords of beach”, “large”, “garden” have higher average prices than without, whereas “quiet” behaves the opposite.

   With or without “beach” has a difference of 59.45

   “Garden” has 27.72

   “Large” has 36.01

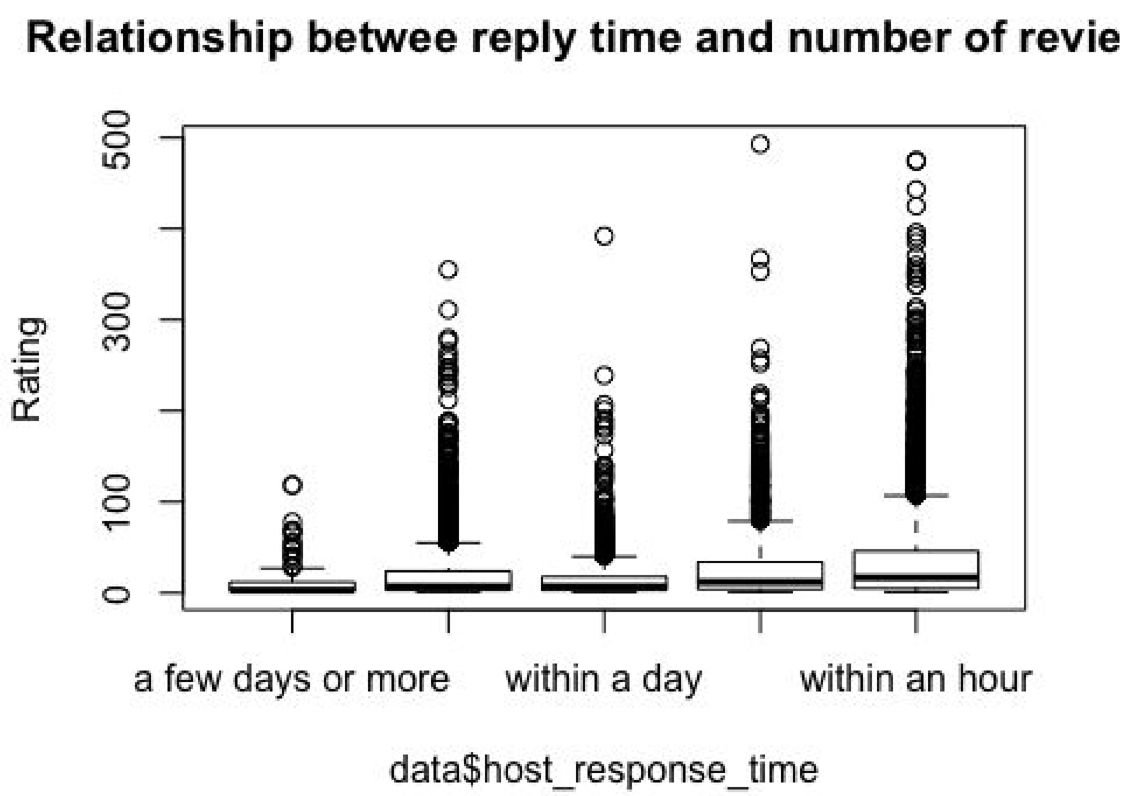
   “Quiet” has -35.35 denoting that listing at noisy(popular) area are more expensive.

7. Strict cancellation policy can give you lower ratings

   By analyzing the how the cancellations might affect the overall ratings, we choose to use a tapply on all ratings, and grouped them by the cancellation policies. It shows that the good policies (more free, like “flexible” and “moderate) has higher ratings with 94.15 and 95.00604 respectively. Whereas strict policies like super_strict_30 only has average of 80.

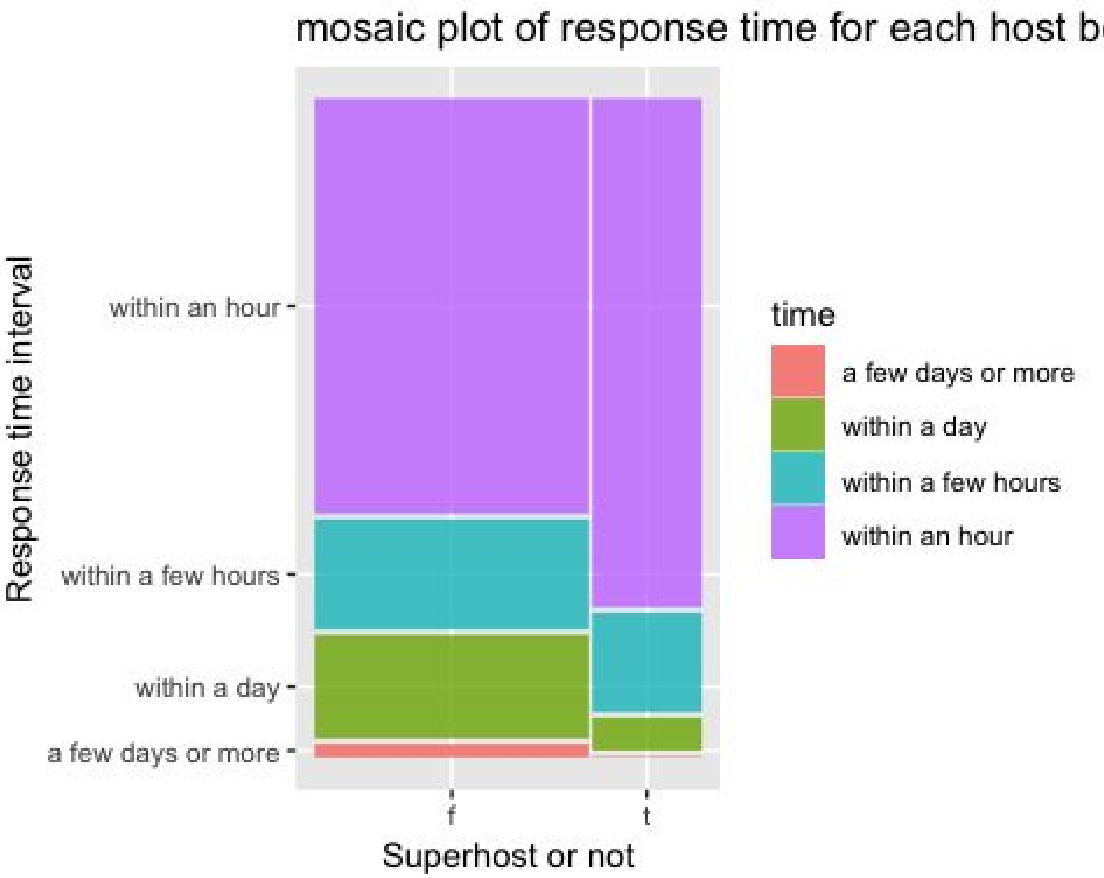
8. Replying faster can boost your popularity

For analyzing the relationship between number of reviews and the replying time(speed), we choose to use box plots to visualize.



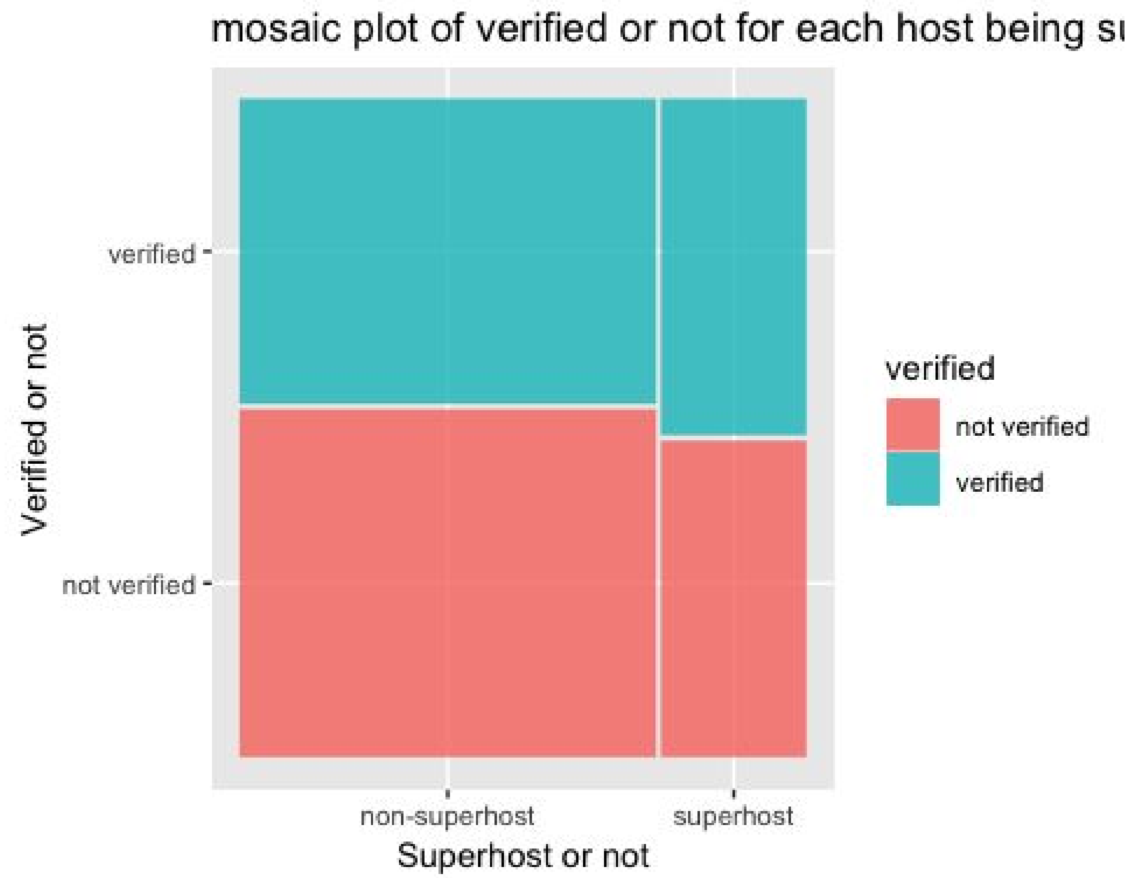
From the graph we can tell that the highest reply speed, which is within an hour, has the highest median and interquartile range.

Also, superhosts generally reply faster, and we have found that superhosts typically having higher ratings and popularities. We can use mosaic plot to visualize this:



9. It is also better to have yourself verified

   By discovering how important to have ourselves verified as a host. We discovered that more superhosts have themselves verified by a mosaic plot.



We also discovered that the average rating of verified hosts' listing is higher, which is 94.51, without verification, the average is 93.38.

10. Customer end - Best zipcodes

As we started to provide analysis for the customer ends. We first define a variable called the rating-price ratio, which is the division of rating by price. Such ratio denotes how well the price is suited for its quality. The higher the ratio, the better experience that the users will have. So out data list the top 5 zipcode, 2146, 2173, 2151, 2119, 2212, as they have rating-price ratio over 2, whereas the average rating-price ratio is only about 0.97.

The remaining part of the pdf are just printed report of the codes and the comments. I figure that it would be more convenient to include them below.

Final_project_91.R

rickypeng99

Mon Apr 29 12:02:19 2019

```
# Final Project for IS 457
# Class id: 91

#Part I: Data processing
data = read.csv("Airbnb Sydney.csv", header = TRUE,  sep=",")

# 1.1 Find missing values

#vector of variables with missing values
missing = c()
count = 1
#Variables that have blanks
for(i in (1 : ncol(data))){
  col = data[, colnames(data)[i]]
  if(class(col) == "factor" & length(col[col == ""]) >= 1){
    print(colnames(data)[i])
    missing[count] = colnames(data)[i]
    count = count + 1
  }
}

## [1] "neighborhood_overview"
## [1] "house_rules"
## [1] "city"
## [1] "zipcode"
## [1] "cleaning_fee"

#Variables that have NA
for(i in (1 : ncol(data))){
  col = data[, colnames(data)[i]]
  if(length(col[is.na(col)]) >= 1){
    print(colnames(data)[i])
    missing[count] = colnames(data)[i]
    count = count + 1

  }
}

## [1] "bathrooms"
## [1] "bedrooms"
```

```
## [1] "review_scores_rating"
## [1] "review_scores_accuracy"
## [1] "review_scores_cleanliness"
## [1] "review_scores_checkin"
## [1] "review_scores_communication"
```

```r
#Variables that have N/A
for(i in (1 : ncol(data))){
  col = data[, colnames(data)[i]]
  if(class(col) == "factor" & length(col[col == "N/A"]) >= 1){
    print(colnames(data)[i])
    missing[count] = colnames(data)[i]
    count = count + 1
  }
}
```

```
## [1] "host_response_time"
## [1] "host_response_rate"
```

```r
#Variables that have - as missing values

for(i in (1 : ncol(data))){
  col = data[, colnames(data)[i]]
  if(class(col) == "factor" & length(col[col == "-"]) >= 1){
    print(colnames(data)[i])
    missing[count] = colnames(data)[i]
    count = count + 1

  }
}
#There isn't any variables containing "-" as missing value
#Totally, there are 14 variables with missing values.
not_missing = c()
count = 1
for(i in (1:length(colnames(data)))){
  if(!colnames(data)[i] %in% missing){
    not_missing[count] = colnames(data)[i]
    count = count + 1
  }
}

#The variables that don't have missing values are:
not_missing
```

```
##  [1] "id"                      "description"
##  [3] "host_id"                 "host_since"
##  [5] "host_is_superhost"       "host_verifications"
##  [7] "host_identity_verified" "property_type"
##  [9] "room_type"               "accommodates"
```

```
## [11] "beds"                   "bed_type"
## [13] "amenities"              "price"
## [15] "guests_included"        "extra_people"
## [17] "minimum_nights"         "number_of_reviews"
## [19] "review_scores_location" "review_scores_value"
## [21] "cancellation_policy"    "reviews_per_month"
```

```
#1.2 How to deal with missing values and why?

#Factor:
#For factors, there is no need to drop the blanks for decription and neighborhood review... etc. These are
#not related to numerical data anaysis, unless we implement sentiment analysis to understand the positivity of the descriptions...etc.
#However we should drop the factor oberservations with N/A when we are dealing with the specific variables.
#For instance, if we want to deal with host_response_rate, we should delete the observations of N/A.

#Numerics:
#Number of missing values of numeric variables

length(which(is.na(data$bathrooms)))
```

```
## [1] 1
```

```
length(which(is.na(data$bedrooms)))
```

```
## [1] 1
```

```
length(which(is.na(data$review_scores_rating)))
```

```
## [1] 1
```

```
length(which(is.na(data$review_scores_accuracy)))
```

```
## [1] 1
```

```
length(which(is.na(data$review_scores_cleanliness)))
```

```
## [1] 1
```

```
length(which(is.na(data$review_scores_checkin)))
```

```
## [1] 1
```

```
length(which(is.na(data$review_scores_communication)))
```

```
## [1] 1
```

```
#For numerics and integers, we should convert all na values to the median values. Such value might be
```

```r
#different from the actual value, but it is not bad for the overall
distribution as the median is never an
#outliers. Also, as the data above showed, there are only one missing value
for every numeric variables that has missing values.
#Therefore, comibing the missing values to median won't actually affect
anything

#1.3 Effects on later data analysis

#Similar as above. Having the NA values to be converted to medians does no
harm to the overall distribution,
#considering the fact that there aren't too much numerical missing values.
Also, each observation has several
#numnerical features (such as rating, accuracy...etc.). Ditching all the data
of these features for one missing
#feature wouldn't be a great choice

#1.4 Dealing with missing values
#Making numerical NAs to be the median
for(i in (1 : ncol(data))){
  col = data[, colnames(data)[i]]
  if(class(col) != "numeric" & class(col) != "integer" ){
    next
  } else{
    col[is.na(col)] = median(na.omit(col))
    data[i] = col
  }
}

# 1.5 After dealing with missing values, show the dimensions of the data.
ncol(data)

## [1] 36

nrow(data)

## [1] 10815

#it should be the same as we are not deleting any N/A entries here.

# 1.6 Comment on and explain any other data cleaning or preparation steps you
think would be
# necessary from your inspection of the data (you do not need to carry them
out).

#I believe that we should clean the data beased on what features we want to
explore.
#Since there are many variables, and some observatins only had one missing
```

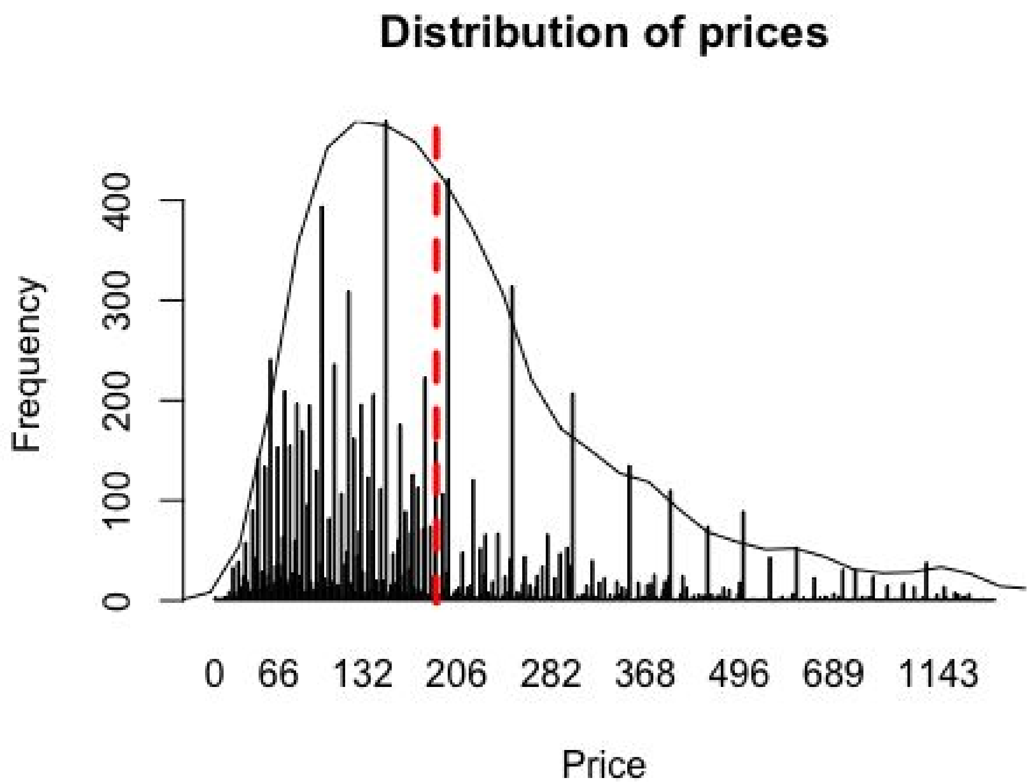
```
variable, which won't affect
#our analysis if we are analyzing other variables. Therefore, we shouldn't
consider deleting any oberservation at the data exploration phase

#Some variables are supposed to have numerical values, such as the
host_response_rate and price. We later should
#convert them to numerics instead of factors.

#2 Preliminary exploration - Exploring some variables comprehensively
#Summary of all varibales related to prices
#Removing the $ signs from prices
data$price = as.numeric(gsub("[$,]", "", data$price))
data$cleaning_fee = as.numeric(gsub("[$,]", "", data$cleaning_fee))
data$extra_people = as.numeric(gsub("[$,]", "", data$extra_people))
#prices
summary(data$price)

##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##     0.0    96.0   150.0   203.2   230.0 10001.0

#It is unexpected that there are houses that are free to live
#cleaning fee
summary(data$cleaning_fee)

##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##    0.00   40.00   80.00   94.47  125.00  800.00     621

#a lot of airbnb actually doesn't have a cleaning fee policy, we should
remove those if we want to do analysis regaring cleaning fee
#fee for extra people
summary(data$extra_people)

##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    0.00    0.00   10.00   17.07   25.00  410.00

#price of listings with highest cleaning fee
head(data[order(-data$cleaning_fee),]$price)

## [1]  999 2353  689 2578  595 1200

#Summary of all ratings
rating_index = c(28, 29, 30, 31, 32, 33, 34)
for(i in (rating_index)){
  print(summary(data[i]))
}

##  review_scores_rating
##  Min.   : 20.00
##  1st Qu.: 92.00
##  Median : 96.00
```

```
##  Mean   : 94.19
##  3rd Qu.:100.00
##  Max.   :100.00
##  review_scores_accuracy
##  Min.   : 2.00
##  1st Qu.: 9.00
##  Median :10.00
##  Mean   : 9.64
##  3rd Qu.:10.00
##  Max.   :10.00
##  review_scores_cleanliness
##  Min.   : 2.000
##  1st Qu.: 9.000
##  Median :10.000
##  Mean   : 9.398
##  3rd Qu.:10.000
##  Max.   :10.000
##  review_scores_checkin
##  Min.   : 2.000
##  1st Qu.:10.000
##  Median :10.000
##  Mean   : 9.782
##  3rd Qu.:10.000
##  Max.   :10.000
##  review_scores_communication
##  Min.   : 2.000
##  1st Qu.:10.000
##  Median :10.000
##  Mean   : 9.802
##  3rd Qu.:10.000
##  Max.   :10.000
##  review_scores_location
##  Min.   : 2.000
##  1st Qu.:10.000
##  Median :10.000
##  Mean   : 9.737
##  3rd Qu.:10.000
##  Max.   :10.000
##  review_scores_value
##  Min.   : 2.000
##  1st Qu.: 9.000
##  Median :10.000
##  Mean   : 9.385
##  3rd Qu.:10.000
##  Max.   :10.000
```

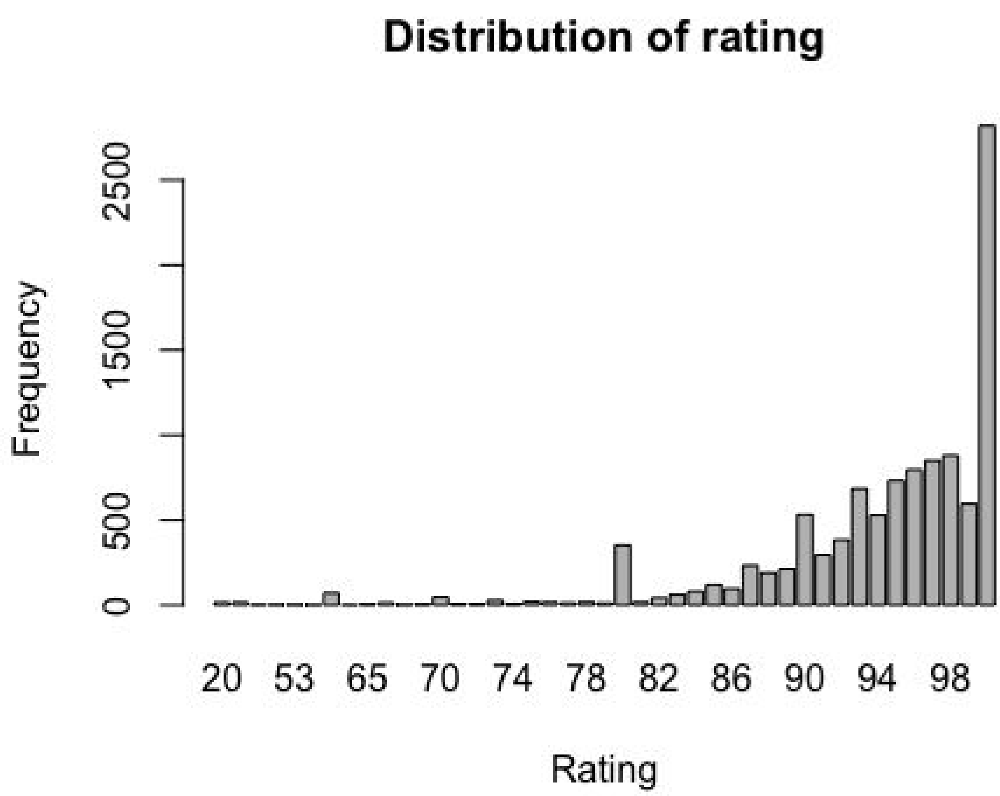
```r
#Summary of other numeric varibles
# # of bedrooms
summary(data$bedrooms)

##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
##   0.000   1.000   1.000   1.629   2.000  14.000

# # of bathrooms
summary(data$bathrooms)

##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
##   0.000   1.000   1.000   1.349   1.500  10.000

# # of minimum nights
summary(data$minimum_nights)

##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
##   1.000   1.000   2.000   4.078   3.000 500.000

# # of beds
summary(data$beds)

##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
##   0.000   1.000   2.000   2.188   3.000  29.000

# # of number of reviews
summary(data$number_of_reviews)

##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
##    1.00    4.00   12.00   28.94   36.00  493.00

# # of reviews per month
summary(data$reviews_per_month)

##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
##   0.020   0.270   0.950   1.572   2.310  15.180

#Explorations of other factors
#area that has the most amount of airbnbs
summary(data$city)[1]

## Bondi Beach 
##         555

#number of superhosts and non superhosts
summary(data$host_is_superhost)

##    f    t 
## 8020 2795
```

```r
#most popular room type
summary(data$room_type)[1]

## Entire home/apt
##            7922

#most popular property type
summary(data$property_type, maxsum = 2)[1]

## Apartment
##      6222

#most popular bed type
summary(data$bed_type, maxsum = 2)[1]

## Real Bed
##    10738

#Mean and min rating for superhosts's airbnb and non-superhosts's airbnb
mean(data[data$host_is_superhost == "t", ]$review_scores_rating)

## [1] 97.06118

min(data[data$host_is_superhost == "t", ]$review_scores_rating)

## [1] 80

mean(data[data$host_is_superhost == "f", ]$review_scores_rating)

## [1] 93.19065

min(data[data$host_is_superhost == "f", ]$review_scores_rating)

## [1] 20

#This is interesting, although the average score of airbnb with superhost is
quite similar to the
#airbnbs that don't have a superhost, (97 vs 93). The minimum bar of
superhost's airbnb is much better
#(80 vs 20). Therefore, it is mostly likely better to find airbnb with a
superhost.
```

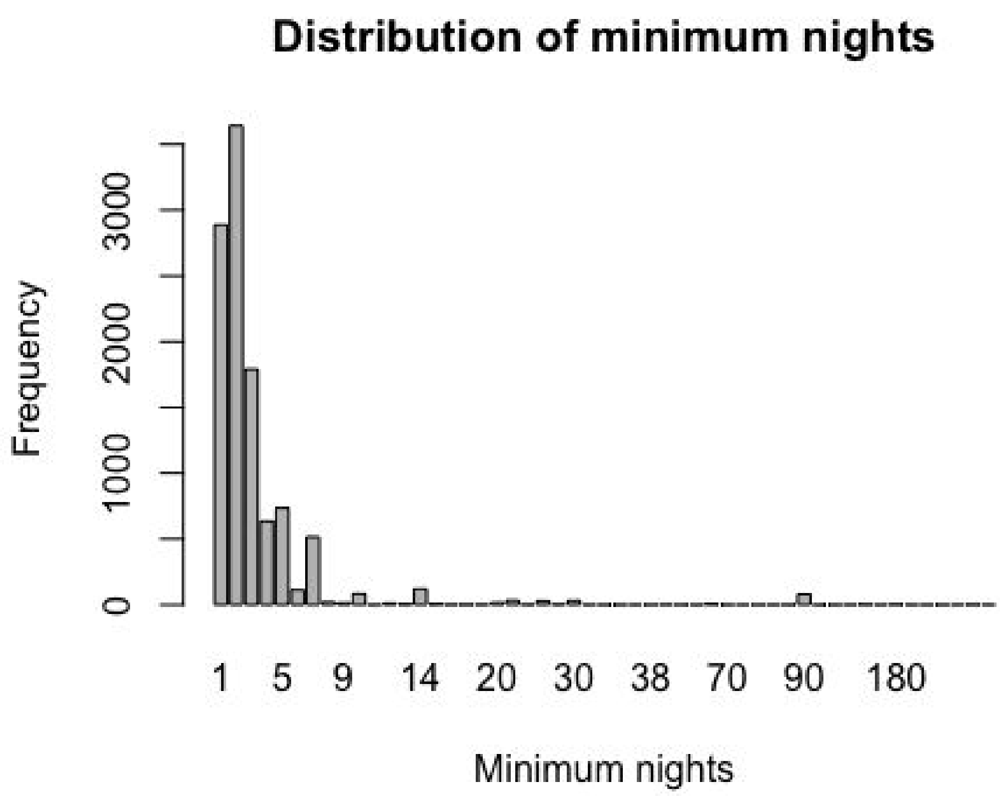
```
#3 visulizations of varibles
#distribution of prices
barplot(table(data$price), ylab = "Frequency", xlab = "Price", main =
"Distribution of prices")
weighted_density = density(data$price)
weighted_density$y = density(data$price)$y * (max(table(data$price)) /
max(density(data$price)$y))
lines(weighted_density)
abline(v = median(data$price), col="red", lwd=3, lty=2)
```



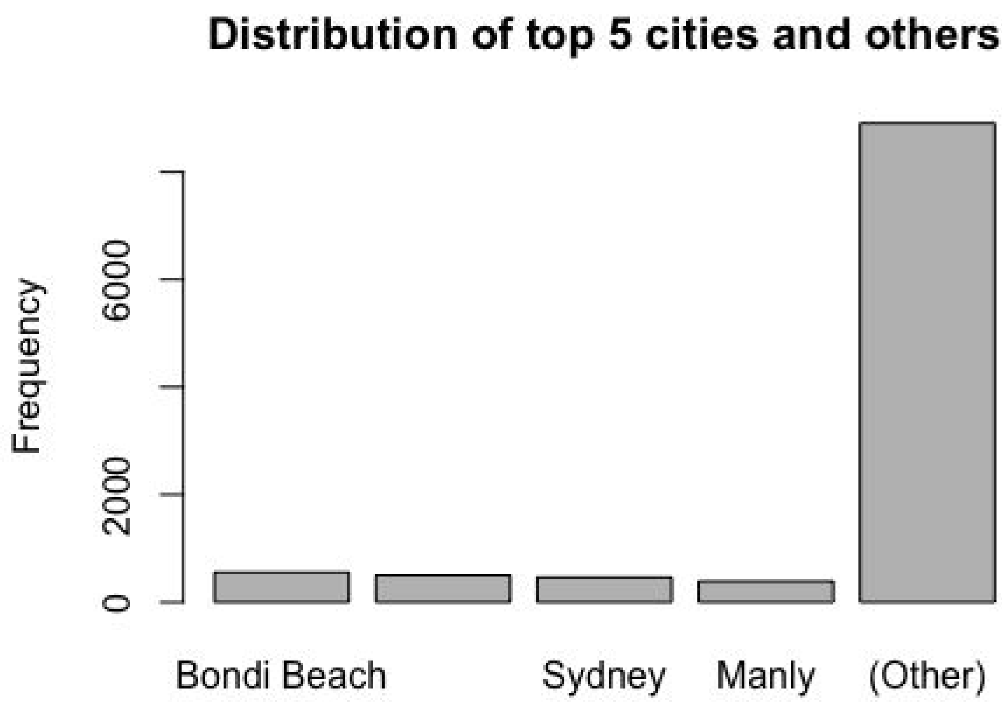
```r
#distribution of rating
barplot(table(data$review_scores_rating), ylab = "Frequency", xlab =
"Rating", main = "Distribution of rating")
```



```
# minimum night distribution
barplot(table(data$minimum_nights), ylab = "Frequency", xlab = "Minimum nights", main = "Distribution of minimum nights")
```

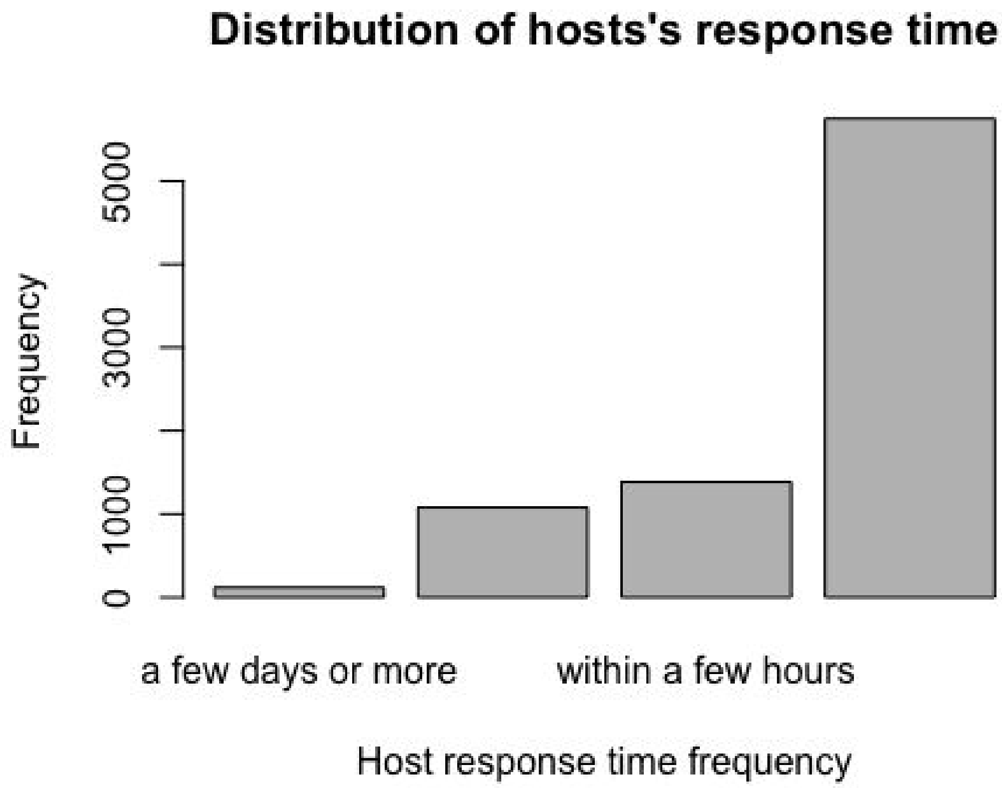


```r
# # of airbnbs in area of the city
#we use bar chart to show top 5 cities and other cities
barplot(summary(data$city, maxsum = 5), ylab = "Frequency", main =
"Distribution of top 5 cities and others")
```



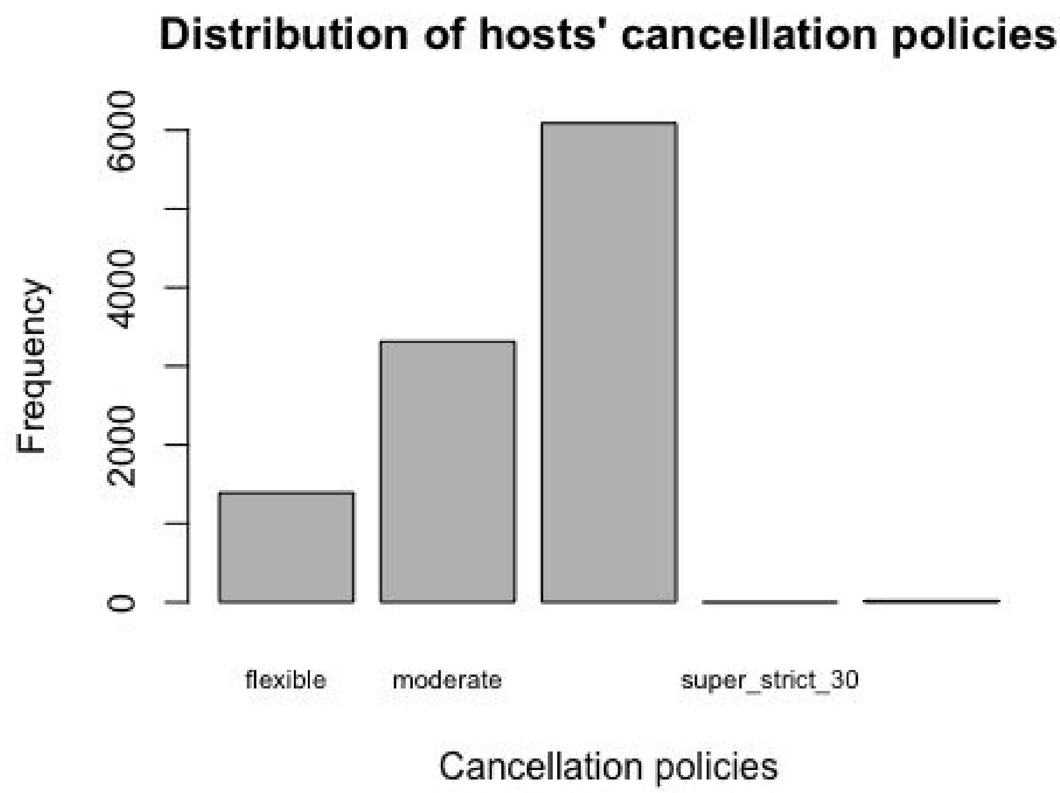

```r
#This means that there are a lot cities levels and even the top 5 cities only
compose a very little amount of it
```

```
#distribution of host_repsonse_time
#remove N/A from host_reponse time
withoutNA = table(droplevels(data$host_response_time[data$host_response_time
!= "N/A"]))
barplot((withoutNA), ylab = "Frequency", xlab = "Host response time
frequency", main = "Distribution of hosts's response time")
```

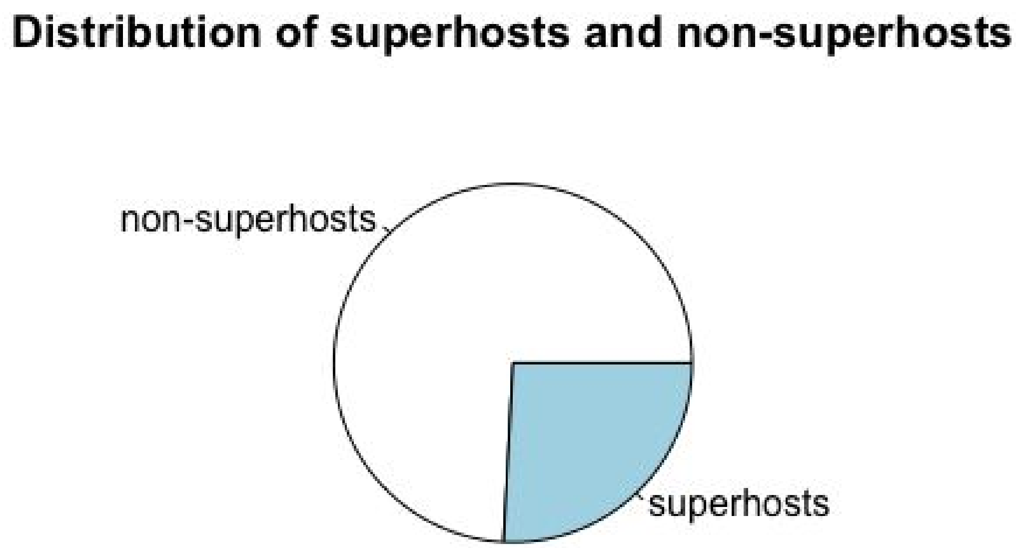



```
#Most of the hosts (had recorded responses) response within an hour
```

```r
#distribution of cancellation policies
barplot(table(data$cancellation_policy), cex.names=.7, ylab = "Frequency", xlab = "Cancellation policies", main = "Distribution of hosts' cancellation policies")
```

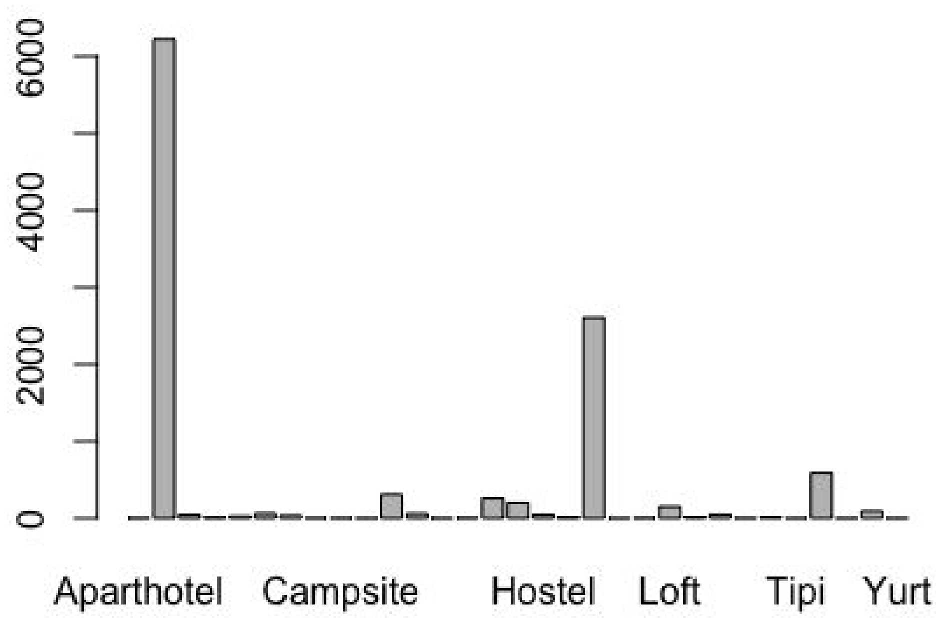


```
#percentage of super and non-super hosts
#We use pie chart here because superhost or not is a binary variable, and a
pie chart is pretty clear for it.
pie(table(data$host_is_superhost), labels = c("non-superhosts",
"superhosts"), main = ("Distribution of superhosts and non-superhosts"))
```

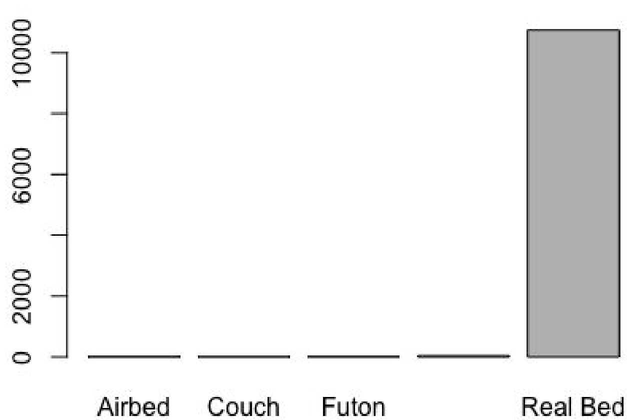


```
#distribution of room_types, property types, bed types
plot(data$room_type)
```

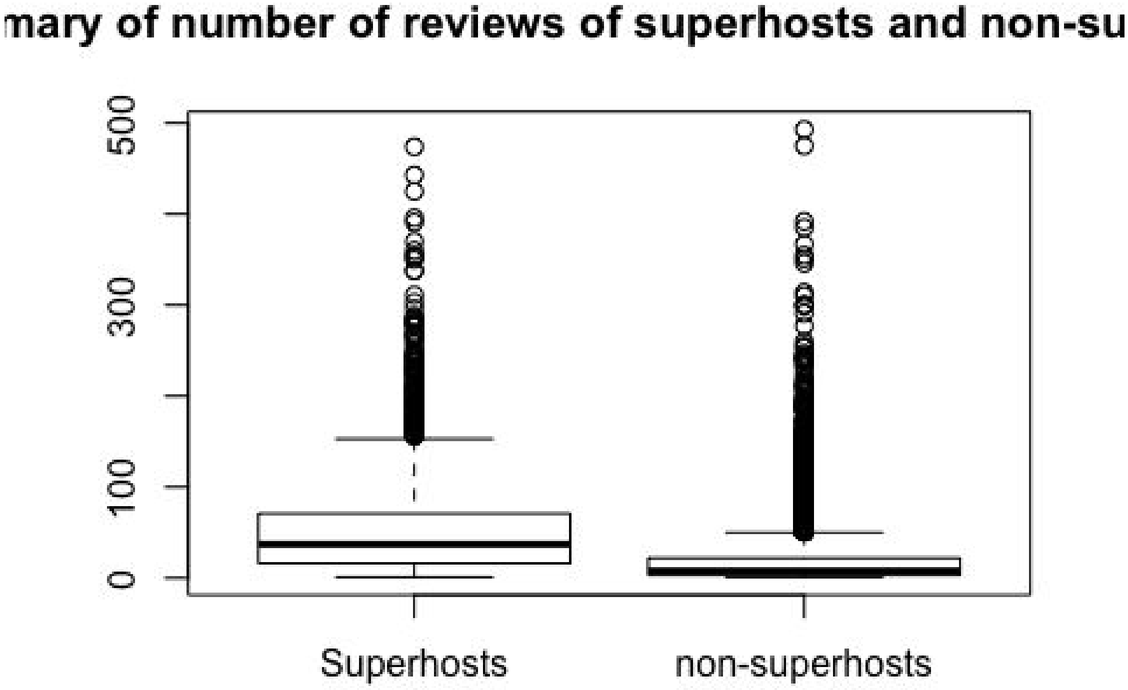
```
plot(data$property_type)
```



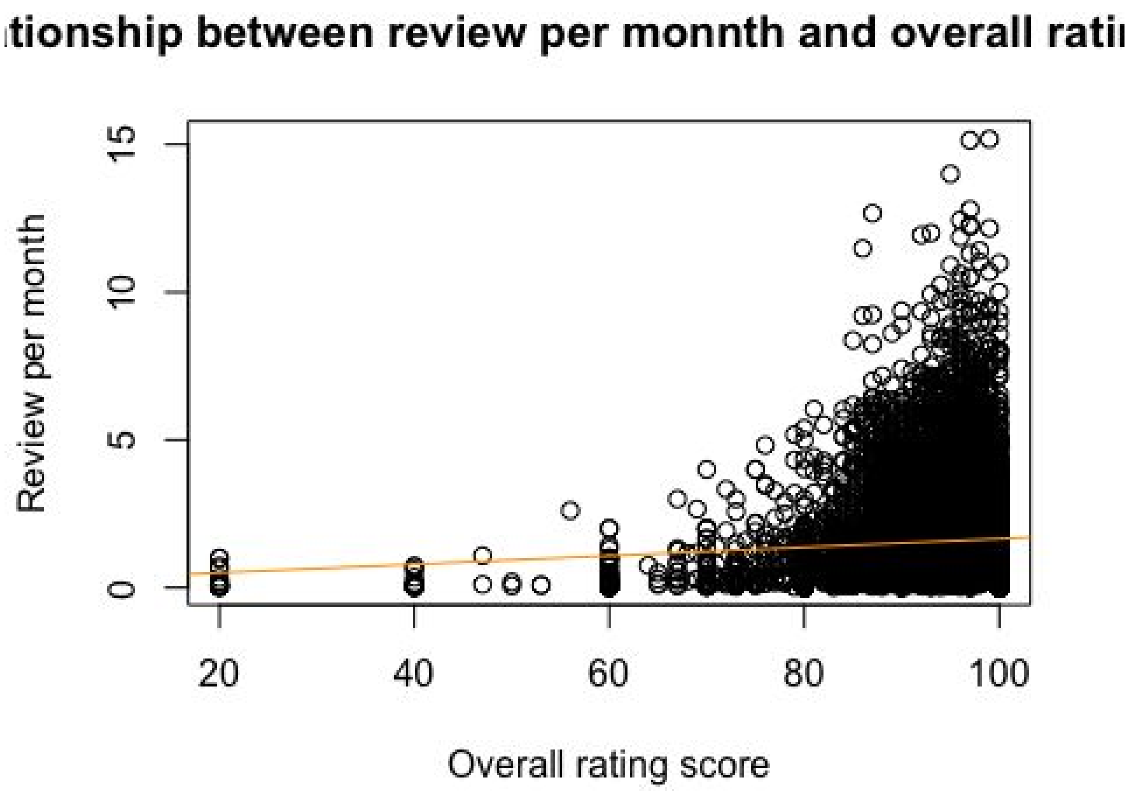

```
plot(data$bed_type)
```



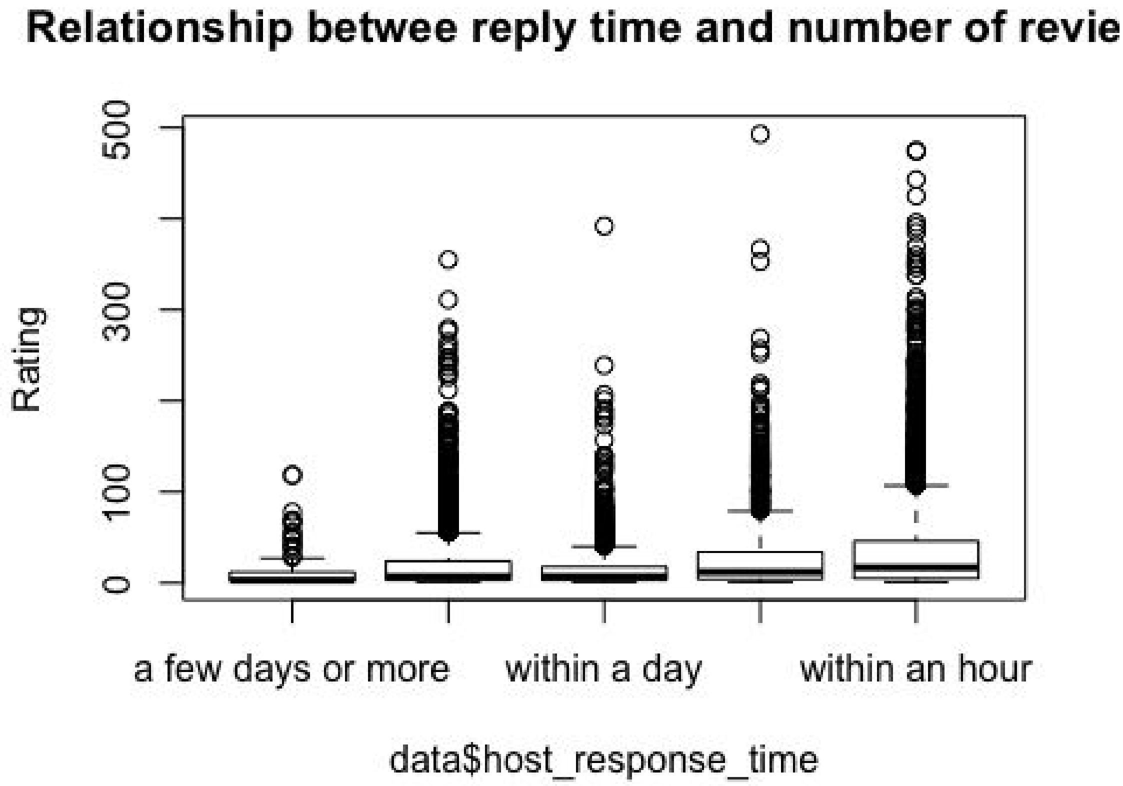
```
#4 Relationships
#4.1 Review_per_month vs number_of_reviews
month = head(data[order(-data$reviews_per_month), ]$id, 100)
number = head(data[order(-data$number_of_reviews), ]$id, 100)

similarity = length(intersect(month, number)) / 100
similarity

## [1] 0.22

#There isn't a huge overlap of host_ids between these two variables (only 22%). Meaning that only
#22 top host_ids reach the top 100 for both variables.
#The host_ids in the number_of_reviews has some ids with low-digits id, that means that older hosts'
#probably have more number of reviews because they had their listings for a longer amount of time. Whereas
#the review_per_month denotes the most recent popularity of a listing.

#In overall, the two variables are related if the hosted_since for all the listings are the same, since the information that
#they convey are both indications of popularities. But since the hosted_since isn't the same for all listings, there are some noises interrupting the relations.

#4.2 Relationships of other three groups of variables
#number of reviews get as a superhost or not
boxplot(data[data$host_is_superhost == "t", ]$number_of_reviews,
data[data$host_is_superhost == "f", ]$number_of_reviews, names =
c("Superhosts", "non-superhosts"), main = "Summary of number of reviews of
superhosts and non-superhosts")
```




```
#We can tell that being a superhost or not does affect the number of reviews.
The interquatile range and the median of number of reviews of superhosts are
all higher than those of non-superhosts.

#relationship between overall rating rating and # of reviews per month.
plot(data$reviews_per_month ~ data$review_scores_rating, ylab = "Review per
month", xlab = "Overall rating score", main = "Relationship between review
per monnth and overall rating score")
abline(lm(data$reviews_per_month ~ data$review_scores_rating), col =
"orange")
```



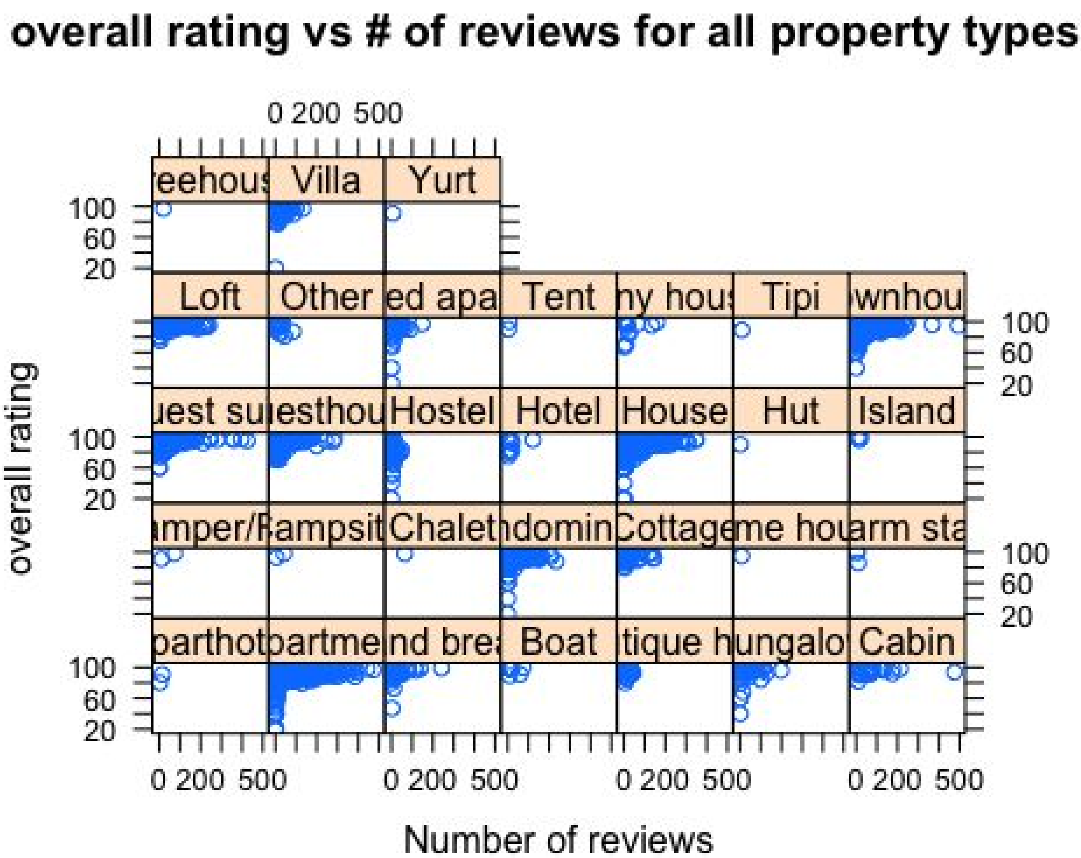

```
#The linear regression of the relationship between the reviews per month and
the overall rating does have a weak postive correalation.
#Also the maximum value of review per month of each level of overall rating
is steadily increasing

#boxplot of comparison between reply time and number of reviews
plot(data$number_of_reviews ~data$host_response_time, ylab = "Rating", main =
"Relationship betwee reply time and number of reviews")
#we can tell that the hosts that reply faster typically have a higher median
of number of reviews. Alos, their interqutile range is higher.
```




```
#5. Hypothesis

#5.1 Being a superhost or not will affect the overall rating of the listing
and the number of reviews.
#I will explore the relationship between number of reviews and overall rating
for each category of being a superhost or not.
#From the explorations, I have discovered the distribution of superhosts or
not and I discovered the relationship between superhosts and number of
reviews to be a postive correlation

#5.2 Having a cleaning fee or not will affect the number of reviews.
#I will divide the listings into the listings with cleaning fee or not. And
explore the relationship between amount of cleaning fee and the overall
rating.
#I have discovered the distribution of cleaning fee. And i noticed that the
```

```
price of highest cleaning fee listings are overwhelmingly high, which
probably have lower number of reviews since people won't want to go there.

#5.3 Different bed types will affect the overall rating of the listing.
#I will explore the relationship between overalll rating of the listing for
each category of bed_types.
#From the explorations I have discovered the distributions of different bed
types, and the most popular bed type. I figure that people would have
preferences over bed_types as the distribution of bed_types is vastly
different.


#II. Data analysis
#6.1 overall rating vs # of reviews for all property types
library("lattice")

## Warning: package 'lattice' was built under R version 3.4.4
```

```
xyplot(data$review_scores_rating ~ data$number_of_reviews |
data$property_type, data = data,
       main = "overall rating vs # of reviews for all property types", xlab =
"Number of reviews",
       ylab = "overall rating")
```



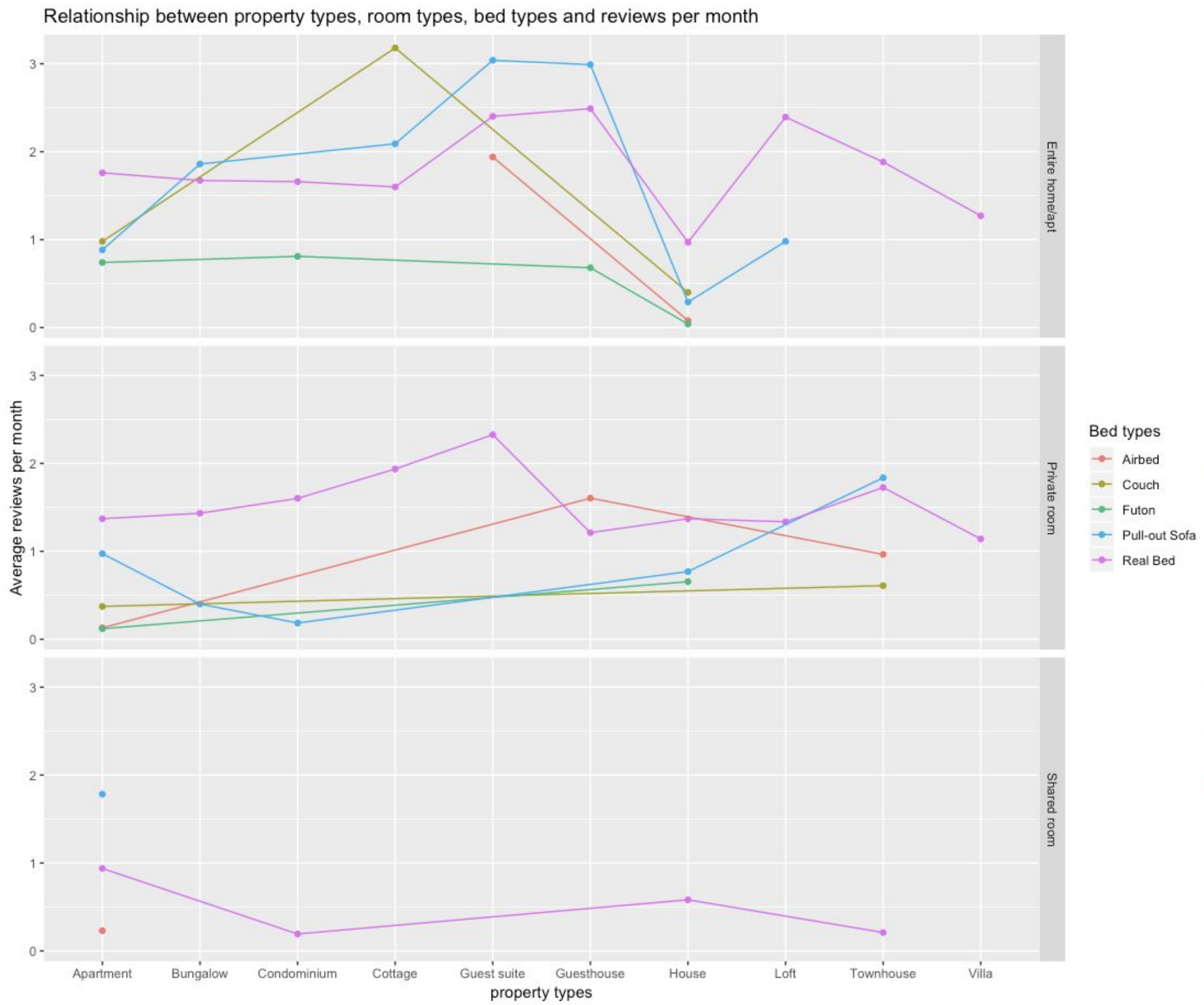

```
#This graph plotted the ralationship between number of reviews and overall
ratings for all types of property types. we can see that for different
property types, there isn't a correlation between number of reviews and
overall ratings, which means that the different propertypes probably won't
affect the overall rating and number of reviews of a listing

#6.2 Relationship betweem multiple variables of property types, room types,
bed types and reviews per month.
library("ggplot2")

## Warning: package 'ggplot2' was built under R version 3.4.4

# # of listings for all categories:
propertyTypes = unique(data$property_type)
sumArray = rep(0, length(propertyTypes))
names(sumArray) = propertyTypes
for(i in (1 : length(data$property_type))){
  for(j in (1 : length(propertyTypes))){
    if(data$property_type[i] == propertyTypes[j]){
      sumArray[j] = sumArray[j] + 1
      break
    }
```

```
  }
}

sumArray = sort(sumArray, decreasing = TRUE)
top_10_property = head(names(sumArray), 10)
#clean data with only top 10 properties
top_10_prop_data = data[data$property_type %in% top_10_property,]

#dropping used levels
top_10_prop_data$property_type = droplevels(top_10_prop_data$property_type)


require("ggrepel")
```

```
## Loading required package: ggrepel
```

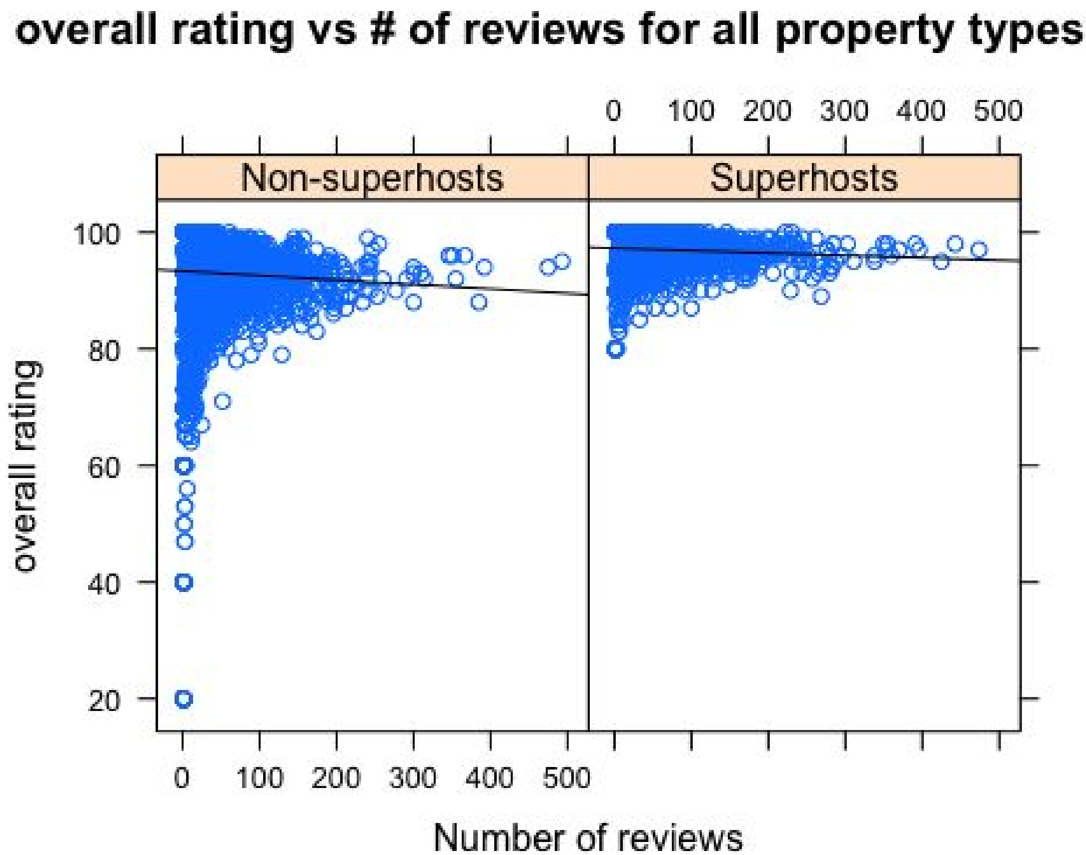
```
## Warning: package 'ggrepel' was built under R version 3.4.4
```

```
ggplot(top_10_prop_data, aes(x=top_10_prop_data$property_type,
y=top_10_prop_data$reviews_per_month, color = top_10_prop_data$bed_type,
group = top_10_prop_data$bed_type)) +
  stat_summary(fun.y=mean, geom = "point") +
  stat_summary(fun.y=mean, geom = "line")  +
facet_grid(top_10_prop_data$room_type~., scale="free_x")
+ggtitle("Relationship between property types, room types, bed types and
reviews per month") + xlab("property types") +ylab("Average reviews per
month")+labs(color = "Bed types")
```



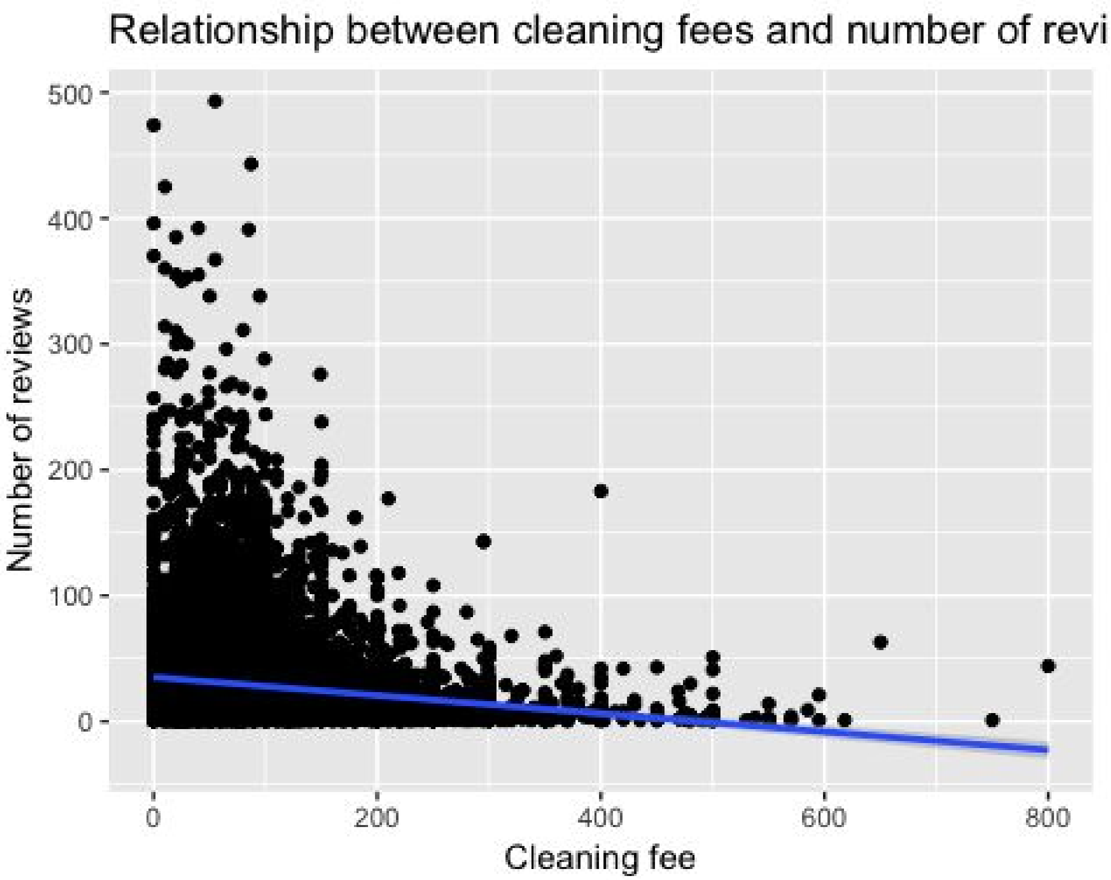

```
#In this graph, since we are finding the relationship among four varibles, we
can tell that Cottages with entire-room and couch has the most amount of
average reviews per month. We could also tell that entire home usually have
more reviews per month, the popularity of shared rooms is very low.

#6.3 plots for hypothesis in q5

#6.3.1 Being a superhost or not will affect the overall rating of the listing
and the number of reviews.
host_superhost = data$host_is_superhost
levels(host_superhost) = c("Non-superhosts", "Superhosts")
xyplot(data$review_scores_rating ~ data$number_of_reviews | host_superhost,
data = data,
       main = "overall rating vs # of reviews for all property types", xlab =
"Number of reviews",
       ylab = "overall rating", panel = function(x, y){
         panel.xyplot(x, y)
         panel.lmline(x, y)
```

```
    })
```

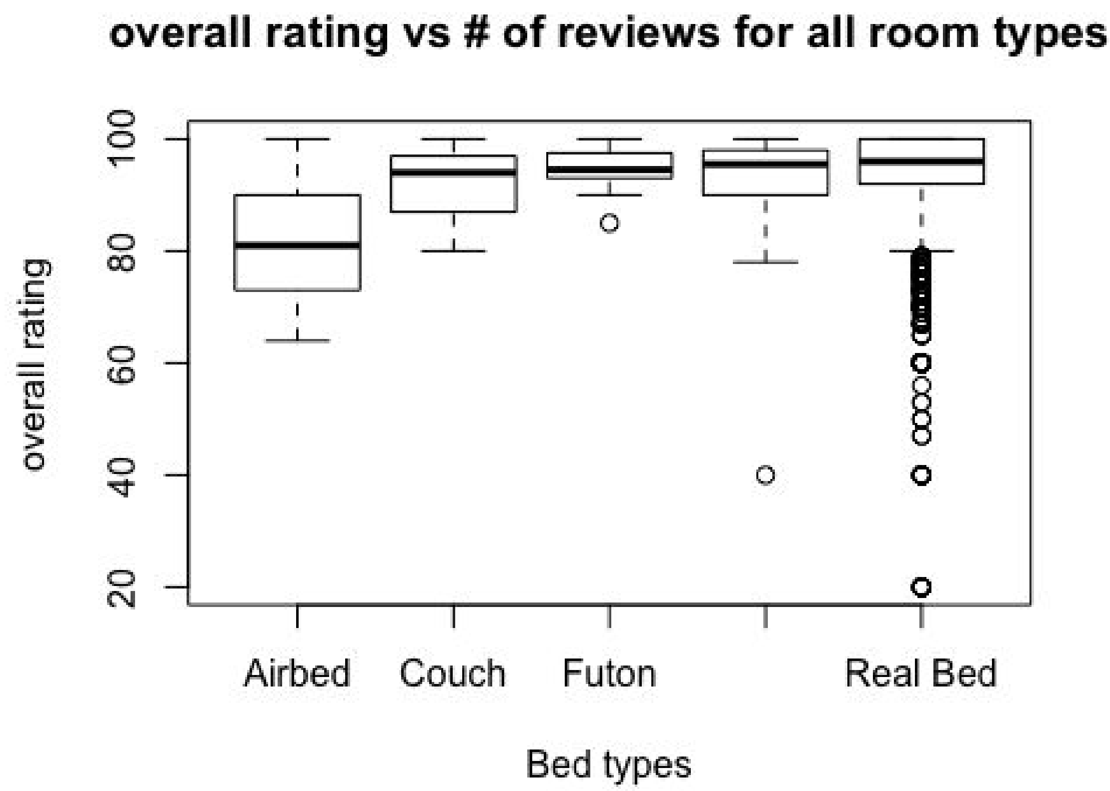



```
#I chose scatter plot with regreesion line, because we are discovering the relationship between two numeric varibales of each category of being a superhost or not. The scatterplot includes all pairs of datas of overall rating and number of reviews.
#For superhosts, the minimum rating is much lower. In terms of the regression line, we can see that the rating of superhosts' listing is higher than the rating of non-superhosts' rating for every level of number of reviews. Which means that superhosts generally get more reviews and higher ratings.

#6.3.2 Having a cleaning fee or not will affect the number of reviews
having_cleaning = data$cleaning_fee
having_cleaning[!is.na(data$cleaning_fee)] = "Having cleaning fee"
having_cleaning[is.na(data$cleaning_fee)] = "Not having cleaning fee"
data = as.data.frame(cbind(data, having_cleaning))
ggplot(data[!is.na(data$cleaning_fee),], aes(x =
data$cleaning_fee[!is.na(data$cleaning_fee)], y =
data$number_of_reviews[!is.na(data$cleaning_fee)])) + geom_point() +
geom_smooth(method='lm',formula=y~x) + ggtitle("Relationship between cleaning
```

```
fees and number of reviews") +ylab("Number of reviews") + xlab("Cleaning
fee")
```




```
#I chose this plot because there are two numeric variables to relate, so from
the regression line, I can tell that once the listing has a cleaning fee, the
higher the fee, the less of the number of reviews we generally get.

#6.3.3 Different room types will affect the overall rating of the listing
plot(data$review_scores_rating ~ data$bed_type ,
       main = "overall rating vs # of reviews for all room types", xlab =
"Bed types",
       ylab = "overall rating")
```



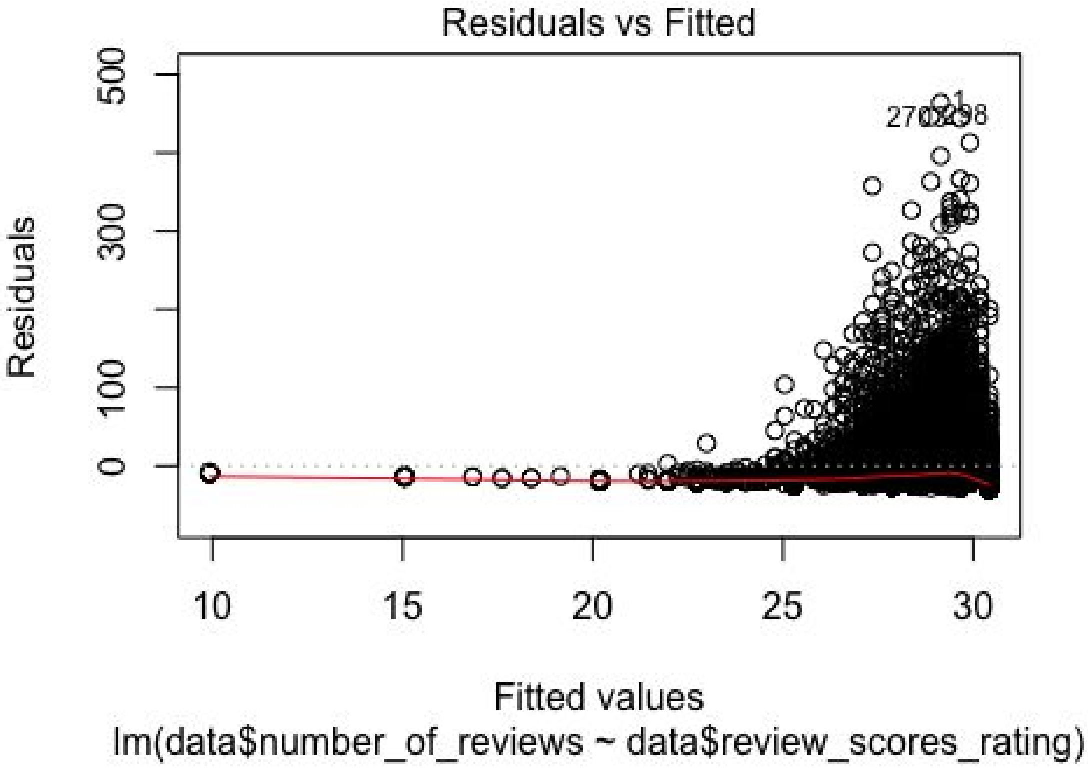

```
#I chose box plot because there are three groups and one numeric value, we
can see that all the other beds except airbed have higher medians and
interquatile range, significantly than the air beds. also the Real bed has
the highest median. From these explorations we can know that people do have a
preference over bed types.

#7 data manipulation
#7.1 remove $ in prices and convert to numerics
data$price = as.numeric(gsub("[$,]", "", data$price))
#7.2 number of amenities column
num_amenities = rep(0, length(data$amenities))
for(i in (1 : length(data$amenities))){
  amenity = data$amenities[i]
  amenity = gsub("[{}]", "", amenity)
  amenity = strsplit(amenity, ",")
  num_amenities[i] = length(amenity[[1]])

}

data = as.data.frame(cbind(data, num_amenities))
```

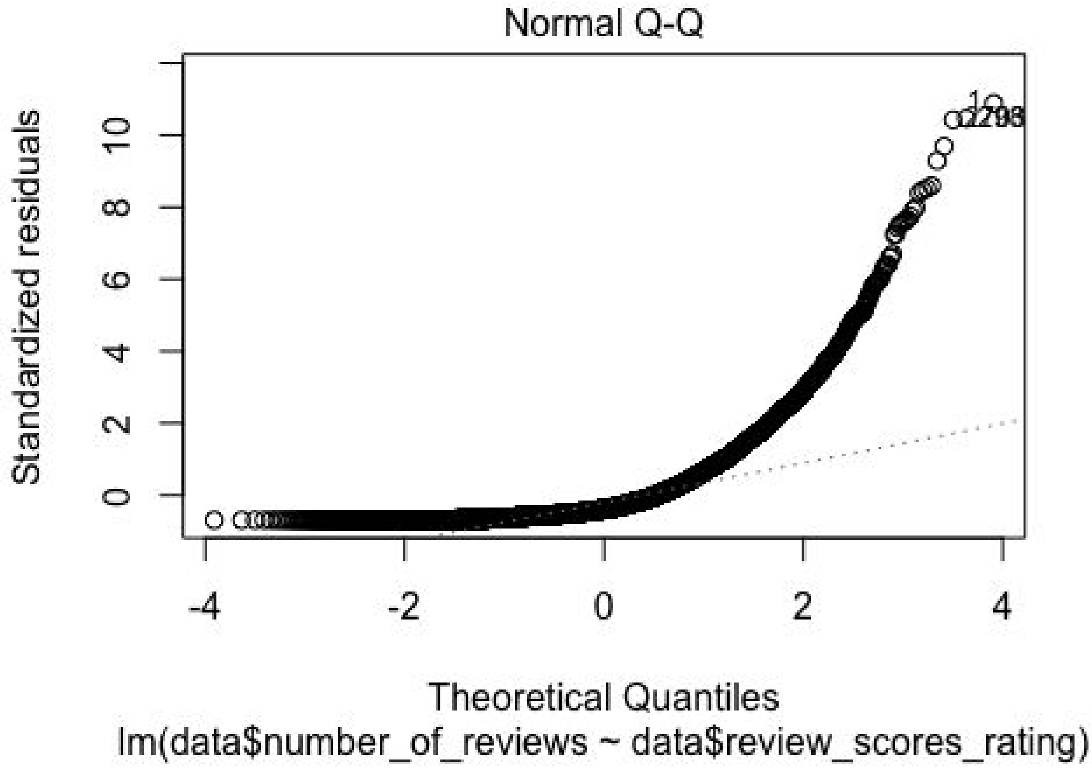
```
#7.3
tapply(data$review_scores_rating, data$cancellation_policy, mean)

##                    flexible                    moderate
##                    94.15888                    95.00604
## strict_14_with_grace_period             super_strict_30
##                    93.77139                    80.00000
##             super_strict_60
##                    89.80000

#strict cancellation policies generally have lower rating

#7.4 more manipulations
#add a column of numers of verifications to the data frame
num_verifications = rep(0, length(data$host_verifications))
for(i in (1 : length(data$host_verifications))){
  verify = data$host_verifications[i]
  verify = gsub("[\\[\\]]", "", verify)
  verify = strsplit(verify, ",")
  num_verifications[i] = length(verify[[1]])

}
data = as.data.frame(cbind(data, num_verifications))

#8 fit simple linear model
#8.1 review_per_month vs number_of_reviews
#I would choose number_of_reviews. Because it means that how many people have
visitied this lisiting,
#also it is more attractive to new customers, as higher number of reviews
means that more people have came
#and the listing might be more safe in some sense.

#Therefore there are several candidate variables that affect the choice of
primary indicator
#1. city, position might affect popularity
#2. reviw_scores_rating, higher score, more popularity
#3. room_type people might prefer private room
#4. host_since, older listing might be more popular
#5. cleaning_fee, people might prefer rooms with lower cleaning fee
#6. cancellation_policy, people might prefer better cancellation policy
#7. host_is_superhost, superhost might be having more popularity
#8. accomodates, people might prefer to bring more people with
#9. bathrooms, more bathrooms might be more popular
#10. minimum nights, shorter minimum nights might be more popular

#I choose the overall ratings, as the higher the rating, the more likely it
become really popular.
```

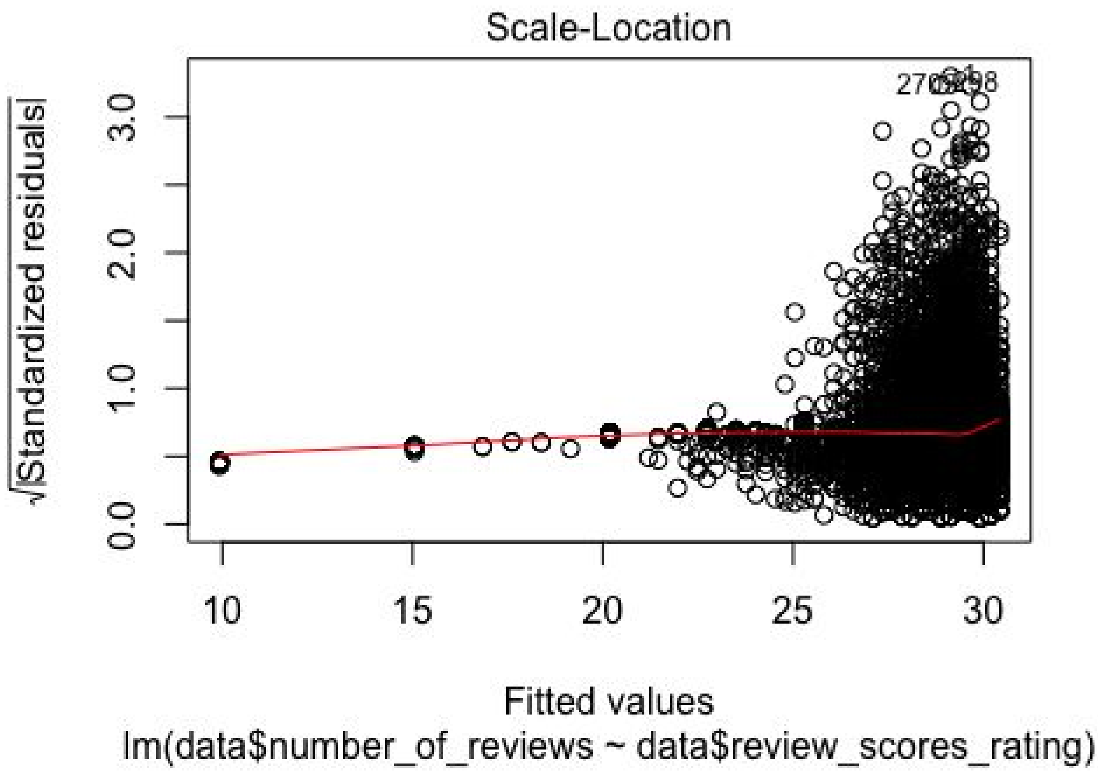
```r
model = lm(data$number_of_reviews ~ data$review_scores_rating, data = data)
summary(model)
```

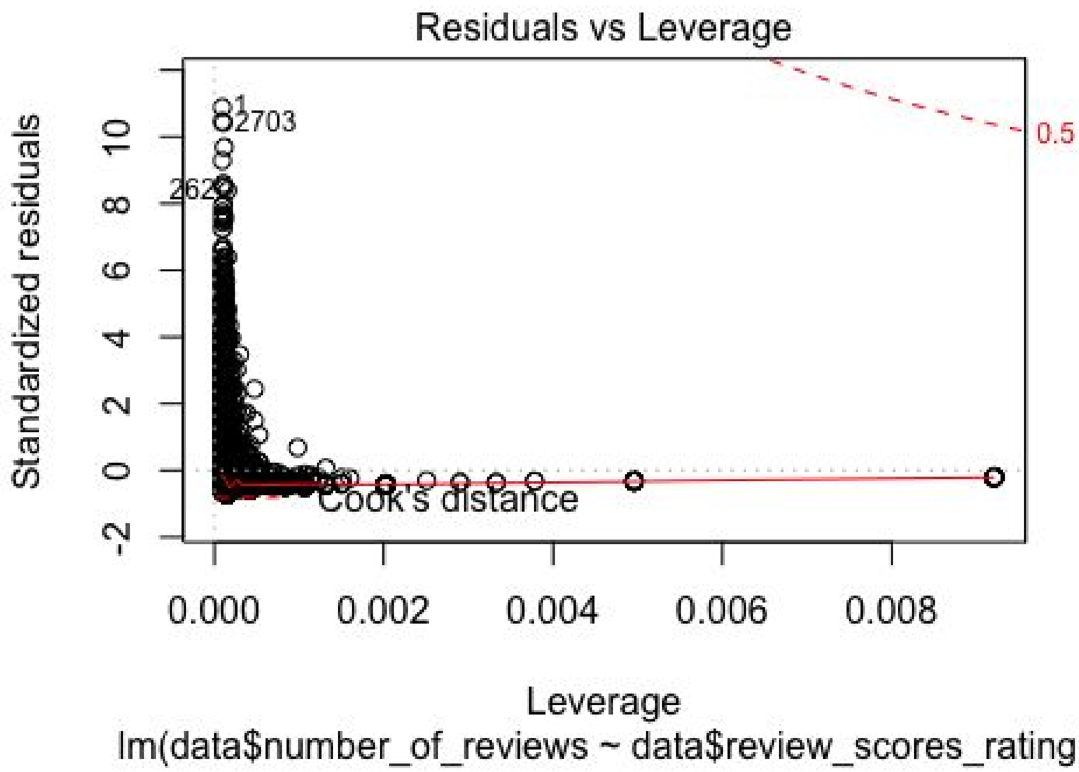
```
##
## Call:
## lm(formula = data$number_of_reviews ~ data$review_scores_rating,
##     data = data)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -29.42 -24.40 -16.63   7.14 463.86
##
## Coefficients:
##                           Estimate Std. Error t value Pr(>|t|)
## (Intercept)                4.80003    5.18337   0.926    0.354
## data$review_scores_rating  0.25624    0.05486   4.671 3.04e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 42.62 on 10813 degrees of freedom
## Multiple R-squared:  0.002014,   Adjusted R-squared:  0.001921
## F-statistic: 21.82 on 1 and 10813 DF,  p-value: 3.035e-06
```

```r
plot(model)
```


Residuals vs Fitted
Residuals
500
300
100
0
10
15
20
25
30
Fitted values
lm(data$number_of_reviews ~ data$review_scores_rating)

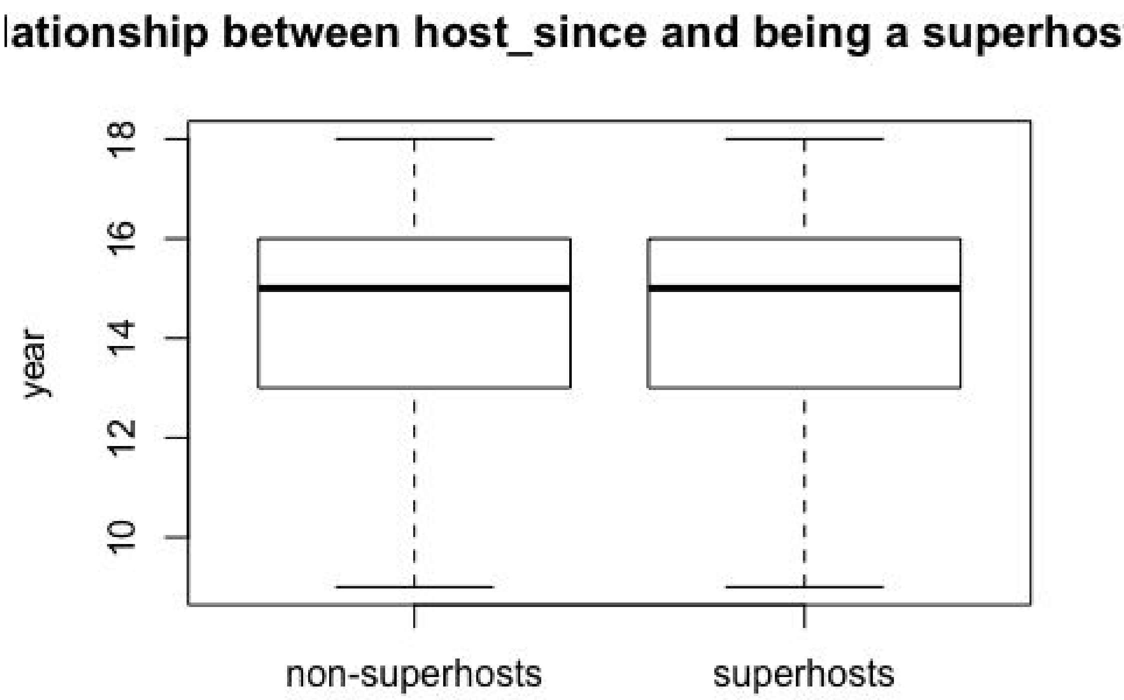

Normal Q-Q
Standardized residuals
0
2
4
6
8
10
-4
-2
0
2
4
Theoretical Quantiles
lm(data$number_of_reviews ~ data$review_scores_rating)


Scale-Location
√|Standardized residuals|
0.0
1.0
2.0
3.0
10
15
20
25
30
Fitted values
lm(data$number_of_reviews ~ data$review_scores_rating)




```
err = summary(model)$coefficients[2,2]
beta = model$coefficients[2]
c(beta-1.96*err,beta+1.96*err)

## data$review_scores_rating data$review_scores_rating
##                 0.1487145                 0.3637586

#it is statisically significant, as the 95% confidence interval doesn't
contain 0, which means that review_scores_rating
#has an effct on number_of_reviews.

#The model seems far from the margin of scatterplot, also, the residual are
far from the zero lines. The standardised
#residual also doesn't lie on the line, denoting that the random errors are
not from the theoritical distributions.
#Hoever, all the data points are far from the cook's distance, denoting that
there aren't too much influential points.Therefore, I wouldn't say that this
model is a great model to suit the data.



#Part III Further analysis
```

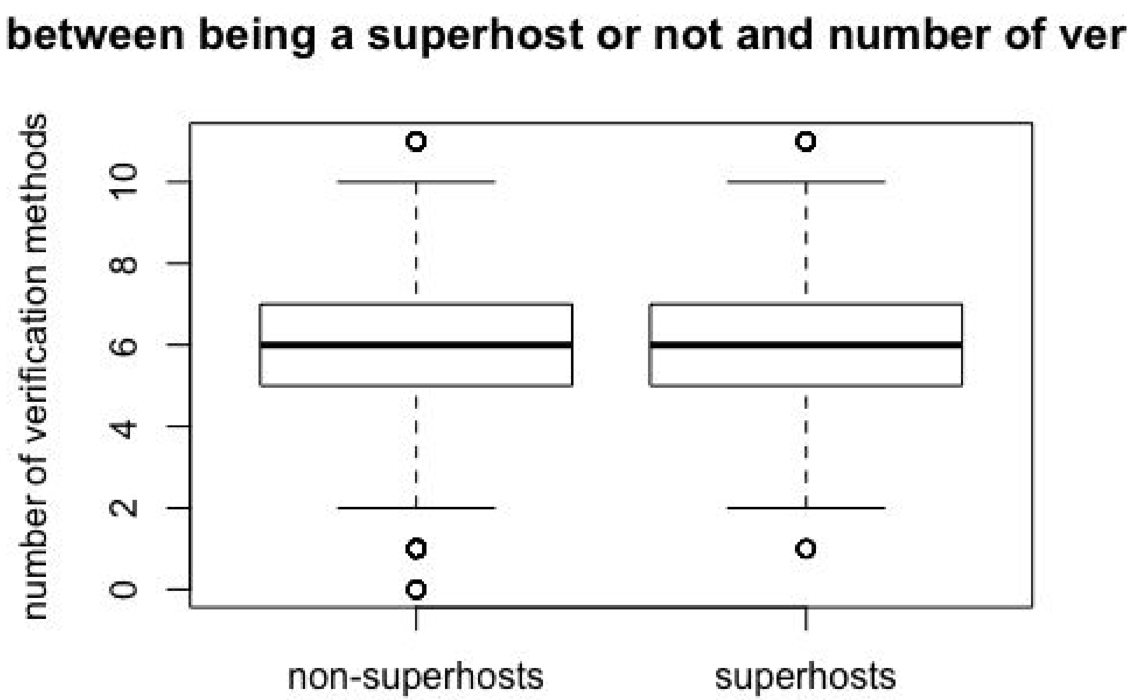
```r
#9.1
#superhost or not vs host_response_rate
host_response_rate_super = data$host_response_rate[data$host_response_rate !=
"N/A" & data$host_is_superhost == "t"]
host_response_rate_not = data$host_response_rate[data$host_response_rate !=
"N/A" & data$host_is_superhost == "f"]
host_response_rate_super = as.numeric(as.character(gsub("%", "",
host_response_rate_super)))
host_response_rate_not = as.numeric(as.character(gsub("%", "",
host_response_rate_not)))
mean(host_response_rate_super)

## [1] 99.14207

mean(host_response_rate_not)

## [1] 95.31711

#being a superhost has a higher average response rate.


#superhost or not vs host since
host_since =as.numeric((sapply(data$host_since, function(x){
  return(strsplit(as.character(x), "/")[[1]][3])
})))
superhost = data$host_is_superhost
levels(superhost) = c("non-superhosts", "superhosts")
tapply(host_since, superhost, summary)

## $`non-superhosts`
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    9.00   13.00   15.00   14.56   16.00   18.00
##
## $superhosts
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    9.00   13.00   15.00   14.63   16.00   18.00

boxplot(host_since ~ superhost, main = "Relationship between host_since and
being a superhost or not", ylab = "year")
```




```
#being a superhost or not doesn't really have any effect on host_since

#superhost or not vs host_verification
#comparing superhost or not against the number of verifications
num_verifications = rep(0, length(data$host_verifications))
for(i in (1 : length(data$host_verifications))){
  verify = data$host_verifications[i]
  verify = gsub("[\\[\\]]", "", verify)
  verify = strsplit(verify, ",")
  num_verifications[i] = length(verify[[1]])

}
superhost = data$host_is_superhost
levels(superhost) = c("non-superhosts", "superhosts")
tapply(num_verifications, superhost, summary)

## $`non-superhosts`
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   0.000   5.000   6.000   5.685   7.000  11.000
##
## $superhosts
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   1.000   5.000   6.000   6.211   7.000  11.000
```

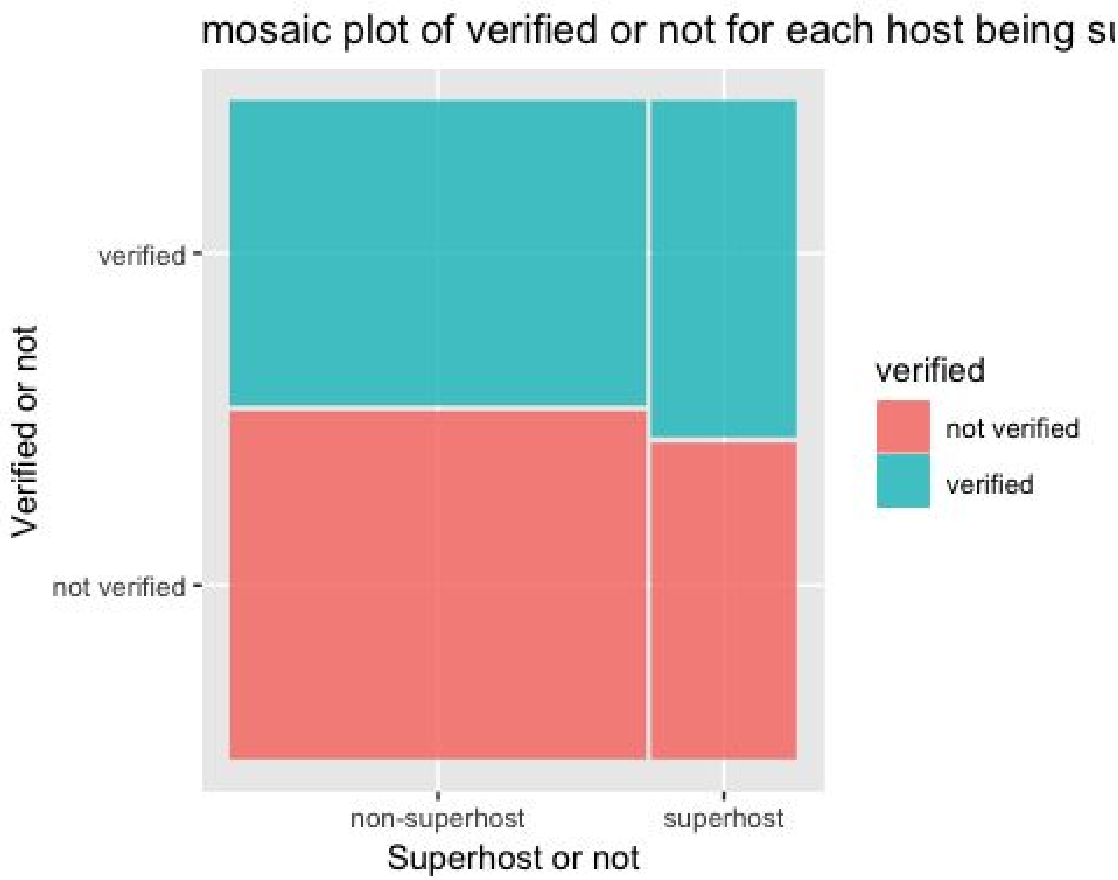
```
boxplot(num_verifications ~ superhost, main = "Relationship between being a
superhost or not and number of verification methods", ylab = "number of
verification methods")
```



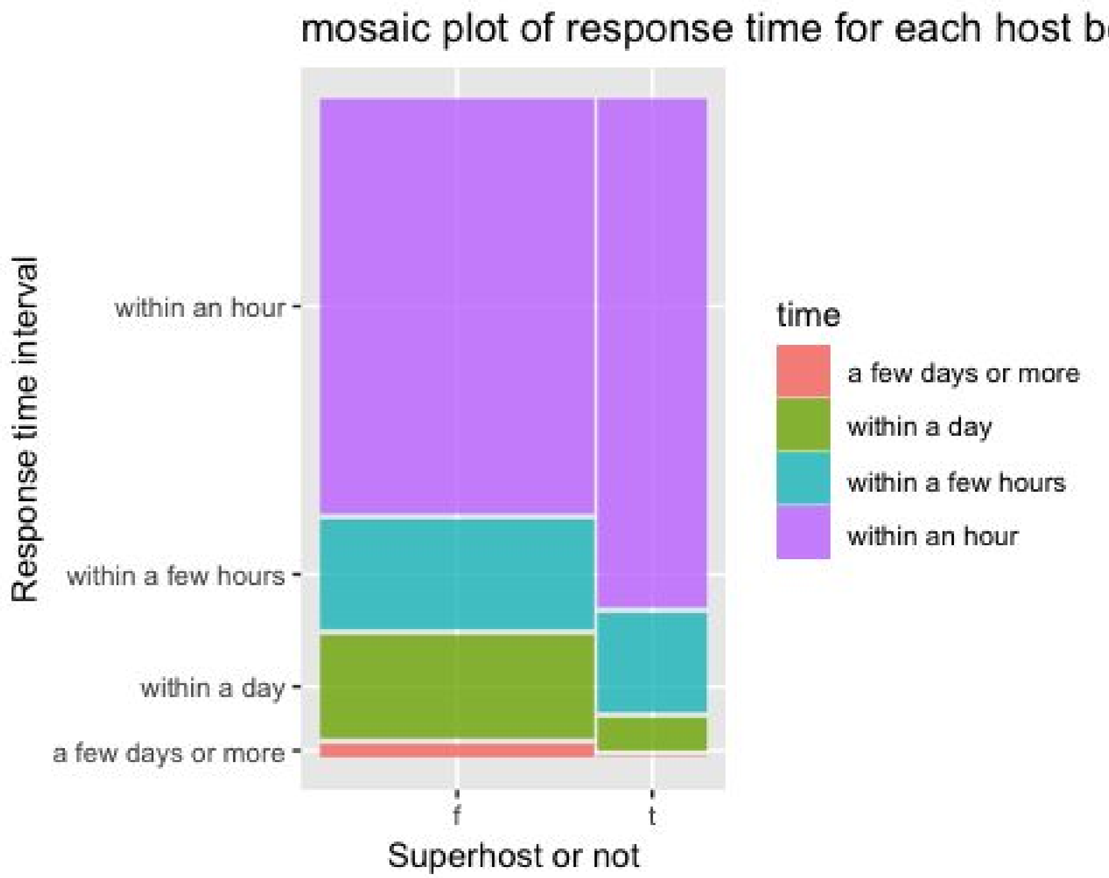

```
#being a superhost or not doesn't really have any effect on
host_verifications

#superhost or not vs host_identity_verified
verified = as.vector(data$host_identity_verified)
superhost = as.vector(data$host_is_superhost)
verified[verified == "f"] = "not verified"
verified[verified == "t"] = "verified"

superhost[superhost == "f"] = "non-superhost"
superhost[superhost == "t"] = "superhost"
require(ggmosaic)

## Loading required package: ggmosaic
```

```
## Warning: package 'ggmosaic' was built under R version 3.4.4
```

```
ggplot() + geom_mosaic(aes(x = product(verified, superhost), fill =
verified)) + xlab("Superhost or not") +ylab("Verified or not")
+ggtitle("mosaic plot of verified or not for each host being superhost or
not")
```




```
#more superhosts have their identities verified.

#superhost or not vs host_response_time (analyzed with mosaic plot in 9.2)
#9.2 mosaic plot of host_response_time vs host_is_superhost
#removing N/A in host_response_time
require(ggmosaic)
time = as.vector(data$host_response_time[data$host_response_time != "N/A"])
superhost = as.vector(data$host_is_superhost[data$host_response_time !=
"N/A"])
ggplot(data = data[data$host_response_time != "N/A", ]) + geom_mosaic(aes(x =
product(time, superhost), fill = time))+ xlab("Superhost or not")
+ylab("Response time interval") +ggtitle("mosaic plot of response time for
each host being superhost or not")
```




```
#superhosts generally reply faster, the percentage of superhosts who reply
within an hour is higher than non-superhosts

#10
#import stop words
stop_word = "a, able, about, across, after, all, almost, also, am, among, an,
and, any, are, as,
              at, be, because, been, but, by, can, cannot, could, dear, did,
do, does, either, else, ever, every, for,
              from, get, got, had, has, have, he, her, hers, him, his, how,
however, i, if, in, into, is, it, its, just,
              least, let, like, likely, may, me, might, most, must, my,
neither, no, nor, not, of, off, often, on,
              only, or, other, our, own, rather, said, say, says, she,
should, since, so, some, than, that, the, their,
              them, then, there, these, they, this, is, to, too, was, us,
wants, was, we, were, what, when, where,
              which, while, who, whom, why, will, with, would, yet, you,
your"
stop_word = strsplit(gsub(" ", "", stop_word), ",")
all_description = gsub("-", " ", data$description)
all_description = gsub("[^[:alpha:] ]", "", all_description)
```

```r
all_description = tolower(all_description)
all_description = strsplit(all_description, " ")
#splitedDescription = all_description
all_description = unlist(all_description)
all_description = all_description[all_description != ""]
all_description = all_description[!all_description %in% gsub("\n",
"",(stop_word[[1]]))]
#all words splited
temp = all_description
#unique words
all_description = unique(all_description)



#calculating word frequncy
temp.freq = table(temp)
class(temp.freq)
```

```
## [1] "table"
```

```r
class(as.integer(temp.freq))
```

```
## [1] "integer"
```

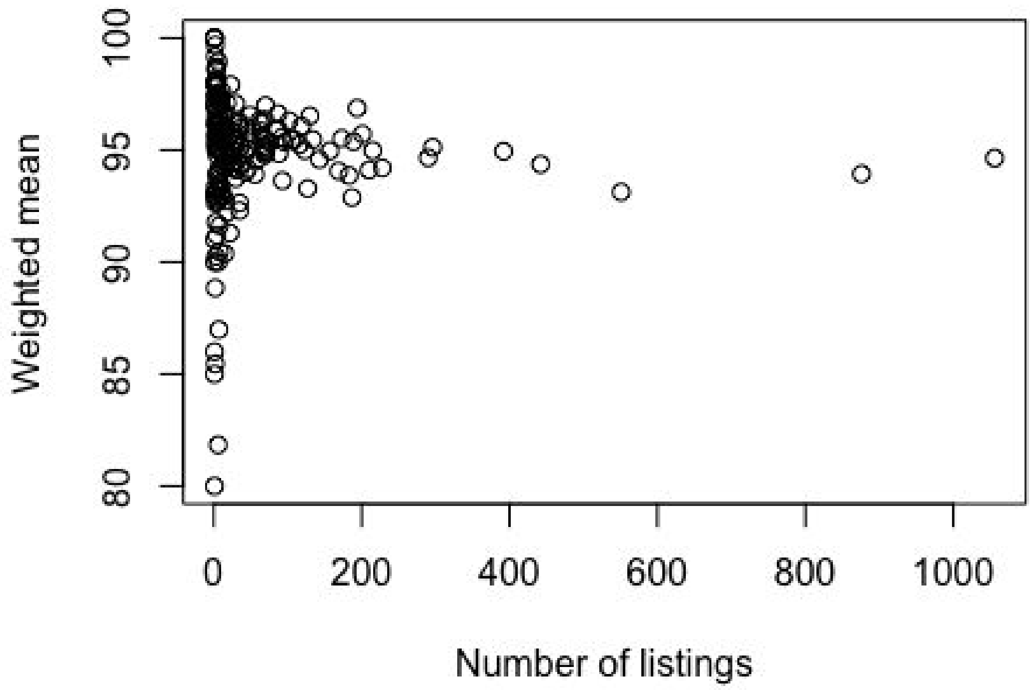
```r
df = as.data.frame(cbind(names(temp.freq), as.integer(temp.freq)))
names(df) = c("Word", "freq")
df$freq = as.numeric(as.character(df$freq))
df = df[order(-(df$freq)),]
head(df)
```

```
##            Word  freq
## 714   apartment 14632
## 1873    bedroom 10651
## 20425      walk 10644
## 18560    sydney  9685
## 15995      room  9637
## 10431   kitchen  9091
```

```r
#word frequency function
wfreq = function(word, text){
  text = tolower(text)
  text = gsub("-", " ", text)
  text = gsub("[^[:alpha:] ]", "", text)
  temp = strsplit(text, " ")
  temp = unlist(temp)
  count = 0
  if(length(temp) >= 1){
    for(i in (1 : length(temp))){
      if(temp[i] == word){
```

```r
        count = count + 1
      }
    }
  }
  return(count)
}

#beach & beaches
#Average price of listings with descriptions containing substring "beach"
("beaches" as well)

booleanArray = sapply(data$description, function(x){
  return(wfreq("beach", x) > 0 | wfreq("beaches", x) > 0)
})

withBeach = mean(data[booleanArray, ]$price)
withBeach

## [1] 237.1627

#no beach
withoutBeach = mean(data[!booleanArray, ]$price)
withoutBeach

## [1] 177.7102

#The difference is
withBeach - withoutBeach

## [1] 59.45252

#3 other words
#contains "quiet"
booleanArray = sapply(data$description, function(x){
  return(wfreq("quiet", x) > 0)
})
withQuiet = mean(data[booleanArray, ]$price)
withoutQuiet = mean(data[!booleanArray, ]$price)
withQuiet

## [1] 176.9059

withoutQuiet

## [1] 212.4427

#Quieter places usually mean that the house is not at a popular area, thus
the price is usually lower

#contains "large"
```

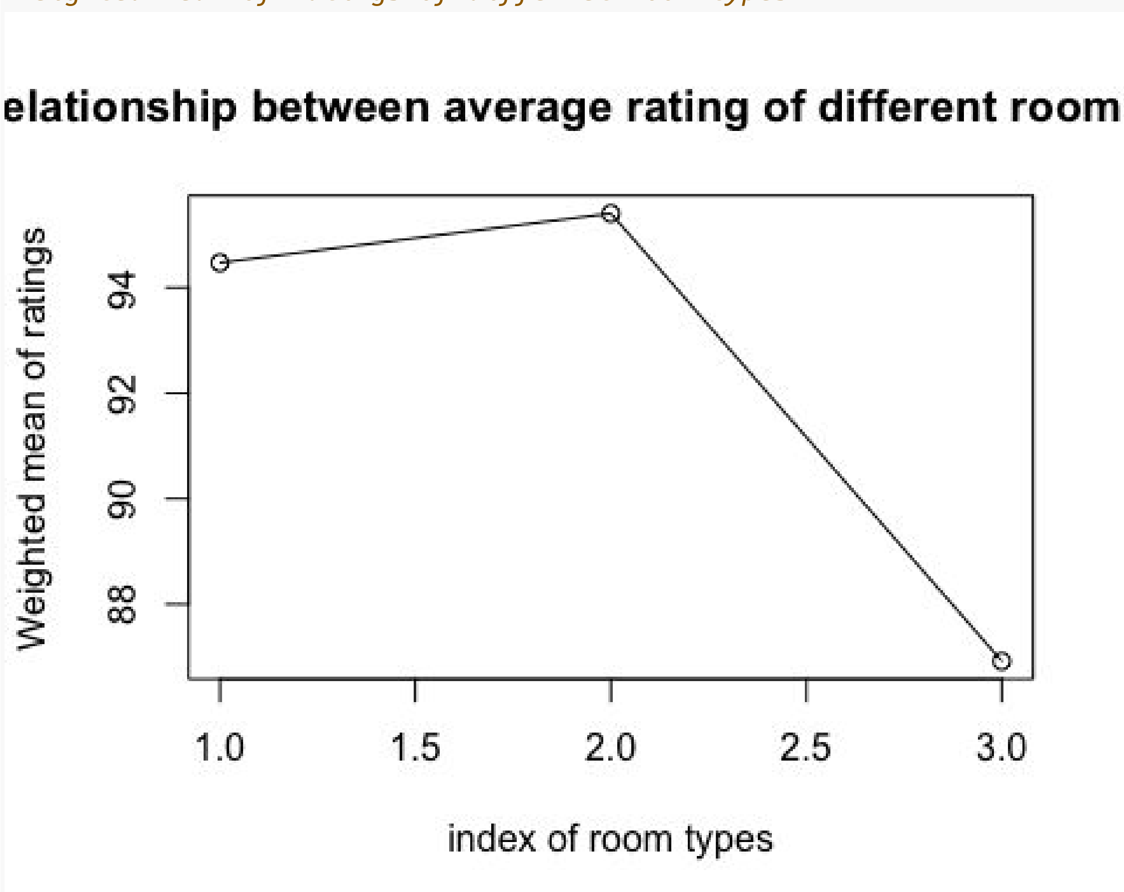
```
booleanArray = sapply(data$description, function(x){
  return(wfreq("large", x) > 0)
})
withLarge = mean(data[booleanArray, ]$price)
withoutLarge = mean(data[!booleanArray, ]$price)
withLarge

## [1] 226.683

withoutLarge

## [1] 190.6644

#larger room often has more price, as the original house price is usually
higher

#contains "garden"
booleanArray = sapply(data$description, function(x){
  return(wfreq("garden", x) > 0)
})
withGarden = mean(data[booleanArray, ]$price)
withoutGarden = mean(data[!booleanArray, ]$price)
withGarden

## [1] 226.3475

withoutGarden

## [1] 198.6187

#listing with garden has more price, as the original house price is usually
higher

#10.2
#(1) top 100 zipcode
zipcode = data$zipcode[data$zipcode != ""]
top100zip = names(summary(zipcode, maxsum = 200))
weightedMean = function(location){
  total_rating =weighted.mean(data[data$zipcode == location,
]$review_scores_rating, data[data$zipcode == location, ]$number_of_reviews)
  return(total_rating)
}
top100zipmean = sapply(top100zip, function(x){
  return(weightedMean(x))
})
plot(top100zipmean ~ summary(zipcode, maxsum = 200), ylab = "Weighted mean",
xlab = "Number of listings")
```




```
#the location (being popular or not for listings) doesn't have any
relationship with the rating

#(2) two other aspects from descriptions
# the word "private"
booleanArray = sapply(data$description, function(x){
  return(wfreq("private", x) > 0)
})
withPrivate = weighted.mean(data[booleanArray, ]$review_scores_rating ,
data[booleanArray, ]$number_of_reviews)
withoutPrivate = weighted.mean(data[!booleanArray, ]$review_scores_rating ,
data[!booleanArray, ]$number_of_reviews)
withPrivate

## [1] 95.40816

withoutPrivate

## [1] 94.31359

#Private could affect the weighted mean, which means that we should look at
the room types
weightedMean = function(type){
```

```
  total_rating =weighted.mean(data[data$room_type == type,
]$review_scores_rating, data[data$room_type == type, ]$number_of_reviews)
  return(total_rating)
}
room_types = levels(data$room_type)


roomtypemean = sapply(room_types, function(x){
  return(weightedMean(x))
})

plot(roomtypemean ~ seq(1:3), main = "Relationship between average rating of
different room types", ylab = "Weighted mean of ratings", xlab = "index of
room types")
lines(seq(1:3), y = roomtypemean, type = "l")
```

```
#Weighted mean of ratings of differnet room types
```




```
entire_apt_weighted_mean = weightedMean(room_types[1])
entire_apt_weighted_mean
```

```
## [1] 94.47359

private_weighted_mean = weightedMean(room_types[2])
private_weighted_mean

## [1] 95.40821

shared_weighted_mean = weightedMean(room_types[3])
shared_weighted_mean

## [1] 86.91732

#We can tell that that the weighted mean of ratings of private spaces (entire
apt and private room) are much higher.

#the word space
booleanArray = sapply(data$description, function(x){
  return(wfreq("parking", x) > 0)
})
with = weighted.mean(data[booleanArray, ]$review_scores_rating ,
data[booleanArray, ]$number_of_reviews)
without = weighted.mean(data[!booleanArray, ]$review_scores_rating ,
data[!booleanArray, ]$number_of_reviews)
with

## [1] 94.77798

without

## [1] 94.65875

#We can see that if the description contains the word "parking", the weighted
mean has a slight increment.
#we can therefore look at ratings of houses with "paking" as amenties and
house that aren't

has_parking = function(amenties){
  return(length(grep("parking", amenties)) >= 1)
}
booleanArray = sapply(data$amenities, function(x){
  x = gsub("[{}]", "", x)
  x = strsplit(x, ",")
  return(has_parking(x))
})
has_parking = data[booleanArray,]
no_parking = data[!booleanArray,]
weighted_mean_parking = c(weighted.mean(has_parking$review_scores_rating,
has_parking$number_of_reviews),
weighted.mean(no_parking$review_scores_rating, no_parking$number_of_reviews))
plot(weighted_mean_parking~ seq(1:2), main = "Comparison between weighted
```

```
mean of having parking or not", xlab = "index of having parking(1 means
having, 2 means no)", ylab = "Weighted mean")
lines(seq(1:2), y = weighted_mean_parking, type = "l")
```


```
#The listings that have parking lots as one of the amentities have higher
weighted mean.

#IV More analysis on customer end
#Define a rating of "Rating - Price ratio", we will find where to live have
higher Rating - Price ratio.
#Make a column of Rating - Price ratio to the dataframe

rating_price_ratio = c()
for(i in (1 : nrow(data))){
  price = data$price[i]
  rating = data$review_scores_rating[i]
  if(price == 0){
    price = mean(data$price)
    #print(price)
  }
  ratio = rating / price
  if(ratio == Inf){
```

```r
    print(price)
    print(rating)
  }

  rating_price_ratio[i] = ratio

}
data = as.data.frame(cbind(data, rating_price_ratio))

#We could explore on what zipcodes averagely have higher rating-price ratio
plot(data$rating_price_ratio ~ data$host_is_superhost)

zipcode = data$zipcode[data$zipcode != ""]
allzip = names(summary(zipcode, maxsum = length(zipcode)))
allzip_mean = sapply(allzip, function(x){
  return(mean(data[data$zipcode == x,]$rating_price_ratio))
})

allzip_mean[order(-allzip_mean)[1:5]]

##     2146     2173     2151     2119     2212
## 2.574134 2.210136 2.155070 2.120623 2.115833

#Living in 2146, 2173, 2151, 2119, 2212 zipcodes usually have the best
experiences.


#V. Conclusion
#In overall, we have explored a lot of variables from this dataset, below are
the suggestions that we can make based on the analysis
#For business end:
#Being a superhost generally can boost your number of reviews, which is
popularity, and the overall rating score.
#Canceling or lowering the cleaning fee can generally earn you higher overall
ratings.
#Hosting private spaces (entire room or private rooms) can earn you higher
ratings and more popularity than shared rooms
#Having real bed, pull out sofa, futon and couch can be better than having an
airbed. People like the real beds the most. so it is the best to not be using
other beds by real beds for your overall rating.
#Hosting listings with parking lots can earn you higher ratings
#You can demand higher prices if your listing is near the beach, or it is
large, or has a garden, or at a popular (noisy) area
#Your good location doesn't necessarily mean the higher rating, the location
has no correlation with its weighted average mean.
#Strict cancellation policy can give you lower ratings
#Replying faster can boost your popularity
#It is also better to have yourself verified, as more of the superhosts
```

```
verify themselves and they averagely having higher ratings and higher numbers
of reviews.

#For Customer end:
#Superhosts usually reply faster
#it is better living in zipcodes of 2146, 2173, 2151, 2119 and 2212

#VI.Exaplanations of data science life cycles (for my project).
# 1. Data Processing
#During this phase, we comprehensively explore all the attributes of
different variables, such that we observe the missing values, we observe the
general distribution of different varibales (single feature) and we deal with
the missing values accordingly. We also observe the potential trend and
relationship between two obvious variables (direct relations), such that we
have a general idea of how variables are related with each other, and how the
data looks like in general.
#We should also analyze how variabels could potentially affect our later
analysis.
# 2. Basic data analysis
#During this phase, we firstly define several hypothesis that gives us
objectives to discover possible business insights, since we have done some
baisc explorations. We then further explore the relationship between multiple
varibales, which will give us a better overall idea of the data, and how
several variables might affect our business sugeestions.
# 3. Data manipulation
#During this phase, we started to notice that in order to do a more
comprehensive analysis, it is necessary to manipulate more of the data such
that it can be better utilized for analysis. For instance, we can transform
the amenities to the number of amenities, such that we could explore the
relationship between other variables and the number of amenities.
#4. Further analysis
#During this phase, we could do more advanced and detailed analysis since we
have the transformed dataset which can be better utilized. For instance, we
can go into the details of description by noticing how the words in
descrition could possibly affect other variables. We can also make
predictions of how the ratings will go if we increase the cleaning price
based on the trends that we discovered.
#5 Make conclusions
#We should make conclusions and provide suggestions based on the trends and
calculations that we discover.
```