Step_0 - Data Cleaning

2023-08-14

```
library(tidyverse)
```

```
## ── Attaching core tidyverse packages ──────────────────────── tidyverse 2.0.0 ──
## ✔ dplyr     1.1.0     ✔ readr     2.1.4
## ✔ forcats   1.0.0     ✔ stringr   1.5.0
## ✔ ggplot2   3.4.1     ✔ tibble    3.2.1
## ✔ lubridate 1.9.2     ✔ tidyr     1.3.0
## ✔ purrr     1.0.1
## ── Conflicts ────────────────────────────────────────── tidyverse_conflicts() ──
## ✖ dplyr::filter() masks stats::filter()
## ✖ dplyr::lag()    masks stats::lag()
## ℹ Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflic
ts to become errors
```

```
df <- read_csv("hong_kong.csv")
```

```
## Rows: 6481 Columns: 75
## ── Column specification ────────────────────────────────────────────────────────
## Delimiter: ","
## chr  (24): listing_url, source, name, description, neighborhood_overview, pi...
## dbl  (37): id, scrape_id, host_id, host_listings_count, host_total_listings_...
## lgl   (9): host_is_superhost, host_has_profile_pic, host_identity_verified, ...
## date  (5): last_scraped, host_since, calendar_last_scraped, first_review, la...
##
## ℹ Use `spec()` to retrieve the full column specification for this data.
## ℹ Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
df <- filter(df, neighbourhood_cleansed == "Wan Chai")
```

Step I: Data Preparation & Exploration (20 points)

I. Missing Values

A.

```
colSums(is.na(df))
```

```
##                                           id
##                                            0
##                                  listing_url
##                                            0
##                                    scrape_id
##                                            0
##                                 last_scraped
##                                            0
##                                       source
##                                            0
##                                         name
##                                            0
##                                  description
##                                           14
##                        neighborhood_overview
##                                         1001
##                                  picture_url
##                                            0
##                                      host_id
##                                            0
##                                     host_url
##                                            0
##                                    host_name
##                                            0
##                                   host_since
##                                            0
##                                host_location
##                                          439
##                                   host_about
##                                          338
##                           host_response_time
##                                            0
##                           host_response_rate
##                                            0
##                         host_acceptance_rate
##                                            0
##                            host_is_superhost
##                                          871
##                           host_thumbnail_url
##                                            0
##                             host_picture_url
##                                            0
##                           host_neighbourhood
##                                           54
##                          host_listings_count
##                                            0
##                    host_total_listings_count
##                                            0
##                           host_verifications
##                                            0
##                         host_has_profile_pic
##                                            0
##                       host_identity_verified
##                                            0
##                                neighbourhood
##                                          1001
##                       neighbourhood_cleansed
##                                             0
##                 neighbourhood_group_cleansed
##                                          1424
##                                     latitude
##                                             0
##                                    longitude
##                                             0
##                                property_type
##                                             0
##                                    room_type
##                                             0
##                                 accommodates
##                                             0
##                                    bathrooms
##                                          1424
##                               bathrooms_text
##                                             4
##                                     bedrooms
##                                           964
##                                         beds
##                                           18
##                                    amenities
##                                             0
##                                        price
##                                             0
##                               minimum_nights
##                                             0
##                               maximum_nights
##                                             0
##                       minimum_minimum_nights
##                                             0
##                       maximum_minimum_nights
##                                             0
##                       minimum_maximum_nights
##                                             0
##                       maximum_maximum_nights
##                                             0
##                       minimum_nights_avg_ntm
##                                             0
##                       maximum_nights_avg_ntm
##                                             0
##                             calendar_updated
##                                          1424
##                             has_availability
##                                             0
##                              availability_30
##                                             0
##                              availability_60
##                                             0
##                              availability_90
##                                             0
##                             availability_365
##                                             0
##                        calendar_last_scraped
##                                            0
##                            number_of_reviews
##                                            0
##                        number_of_reviews_ltm
##                                            0
##                       number_of_reviews_l30d
##                                            0
##                                 first_review
##                                          962
##                                  last_review
##                                          962
##                         review_scores_rating
##                                          962
##                       review_scores_accuracy
##                                          968
##                    review_scores_cleanliness
##                                          968
##                        review_scores_checkin
##                                          968
##                  review_scores_communication
##                                          968
##                       review_scores_location
##                                          968
##                          review_scores_value
##                                          968
##                                      license
##                                         1424
##                             instant_bookable
##                                            0
##               calculated_host_listings_count
##                                            0
##  calculated_host_listings_count_entire_homes
##                                            0
## calculated_host_listings_count_private_rooms
##                                            0
##  calculated_host_listings_count_shared_rooms
##                                            0
##                            reviews_per_month
##                                          962
```

There are many missing values in many columns.

Some columns have as many missing values as the amount of rows.

This means the webscraping did not work on some of the data.

In the dataframe itself we can also see N/A values as strings not counted by is.na function that we will have to account for.

First we will remove the columns that will not be useful in our analysis.

Then we will deal with missing values differently for each column

REMOVING COLUMNS

Id is unique to each row

Every information related to the scraping process is not useful

We can remove neighbourhood_cleansed we already know it is Wan Chai

```
df <- dplyr::select( df, -c('id', 'scrape_id', 'last_scraped',
                'calendar_last_scraped', 'source', 'neighbourhood_cleansed'))
```

The urls contain no useful information

host_name can have the same name repeated for different hosts

host_about is written by the host himself and is too subjective

```
df <- dplyr::select( df, -c('listing_url', 'picture_url', 'host_url',
                            'host_thumbnail_url', 'host_name', 'host_about'
                            , 'host_thumbnail_url', 'host_picture_url'))
```

bathrooms, calendar updated, license, neighbhourhood_group_cleansed have

all their rows as NAs

```
df <- dplyr::select( df, -c( 'bathrooms', 'calendar_updated',
                             'license', 'neighbourhood_group_cleansed'))
```

neighbourhood is just neighbourhood_cleansed with useless information

added and lots of NAs, name concatenates other columns,

It is not clear what the minimum/maximum_minimum/maximum are and they are almost identical to the minimum or maximum columns

```
df <- dplyr::select( df, -c( 'neighbourhood', 'name', 'minimum_minimum_nights',
                          "maximum_minimum_nights", 'minimum_maximum_nights',
                          "maximum_maximum_nights" , "minimum_nights_avg_ntm",
                          "maximum_nights_avg_ntm"
                             ))
```

We remove first_review and last_review, most of the rows are NA and we also have information on recent reviews in the number of reviews ltm and number of reviews l30d columns.

```
df <- dplyr::select(df, -c('first_review', 'last_review'))
```

Calculated_host_listings_count is almost the same as another column and it is

the sum of three other columns also

```
df <- dplyr::select(df, -c('calculated_host_listings_count' ))
```

DEALING WITH MISSING VALUES

NEIGHBOURHOOD_OVERVIEW

It is a description so if there are no descriptions we can replace it with “None” we can exclude that word later for our wordcloud

```
df$neighborhood_overview[is.na(df$neighborhood_overview)] <- "None"
```

DESCRIPTION

Same as before

```
df$description[is.na(df$description)] <- "None"
```

BEDS COLUMN

Number of beds has 14 values missing

```
mean(df[!is.na(df$beds),]$accommodates ==  df[!is.na(df$beds),]$beds)
```

```
## [1] 0.359175
```

```
mean(df[!is.na(df$beds),]$accommodates == 2 * df[!is.na(df$beds),]$beds)
```

```
## [1] 0.4964438
```

Most of its values are equal to two times the amount of people the apartment

can accommodate. Thus we can replace it by that value it will have a low

impact on the dataset because the amount of missing values is very low.

We use ceiling instead of floor to avoid getting 0 beds

```
df[is.na(df$beds),]$beds <-
  df[is.na(df$beds),]$accommodates/2 %>% sapply(ceiling)
```

BEDROOMS COLUMN

Number of bedrooms has most of its values missing, we consider replacing it using the number of beds column to predict it

```
cor(df[!is.na(df$bedrooms),]$bedrooms, df[!is.na(df$bedrooms),]$beds)
```

```
## [1] 0.946962
```

We create a linear model to make the prediction

```
fit <- lm(bedrooms ~ beds, data = df[complete.cases(df$bedrooms),])
```

Predict missing values based on number of beds

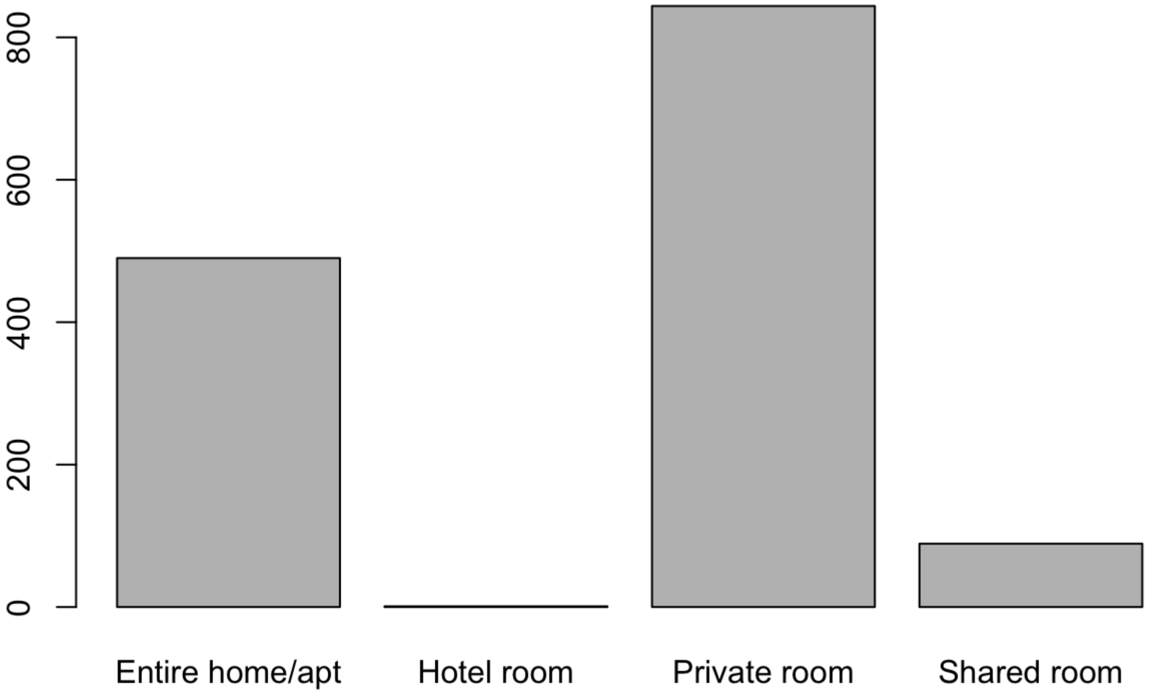
```
df$bedrooms[is.na(df$bedrooms)] <-
  predict(fit, newdata = df[is.na(df$bedrooms), "beds", drop = FALSE])
```

Use ceiling function to avoid decimals and avoid having 0 bedrooms

```
df$bedrooms <- sapply(df$bedrooms,ceiling)
```

Let us check the frequency of room_types, if most were private rooms we could consider replacing it with the median of 1

```
df$room_type <- as.factor(df$room_type)

barplot(table(df$room_type))
```



The majority are private rooms twice as many entire homes/apartments, the two other categories have very low frequencies. All private rooms are one bedroom and a considerable amount of homes or apartments are studios with only one bedroom so we can confidently

```
df[is.na(df$bedrooms),]$bedrooms <- median(df[!is.na(df$bedrooms),]$bedrooms)

colSums(is.na(df))
```

```
##                                  description
##                                            0
##                        neighborhood_overview
##                                            0
##                                      host_id
##                                            0
##                                   host_since
##                                            0
##                                host_location
##                                          439
##                           host_response_time
##                                            0
##                           host_response_rate
##                                            0
##                         host_acceptance_rate
##                                            0
##                            host_is_superhost
##                                          871
##                           host_neighbourhood
##                                           54
##                          host_listings_count
##                                            0
##                    host_total_listings_count
##                                            0
##                           host_verifications
##                                            0
##                         host_has_profile_pic
##                                            0
##                       host_identity_verified
##                                            0
##                                     latitude
##                                            0
##                                    longitude
##                                            0
##                                property_type
##                                            0
##                                    room_type
##                                            0
##                                 accommodates
##                                            0
##                               bathrooms_text
##                                            4
##                                     bedrooms
##                                            0
##                                         beds
##                                            0
##                                    amenities
##                                            0
##                                        price
##                                            0
##                               minimum_nights
##                                            0
##                               maximum_nights
##                                            0
##                             has_availability
##                                            0
##                              availability_30
##                                            0
##                              availability_60
##                                            0
##                              availability_90
##                                            0
##                             availability_365
##                                            0
##                            number_of_reviews
##                                            0
##                        number_of_reviews_ltm
##                                            0
##                       number_of_reviews_l30d
##                                            0
##                         review_scores_rating
##                                          962
##                       review_scores_accuracy
##                                          968
##                    review_scores_cleanliness
##                                          968
##                        review_scores_checkin
##                                          968
##                  review_scores_communication
##                                          968
##                       review_scores_location
##                                          968
##                          review_scores_value
##                                          968
##                             instant_bookable
##                                            0
##  calculated_host_listings_count_entire_homes
##                                            0
## calculated_host_listings_count_private_rooms
##                                            0
##  calculated_host_listings_count_shared_rooms
##                                            0
##                            reviews_per_month
##                                          962
```

SUPERHOST COLUMN

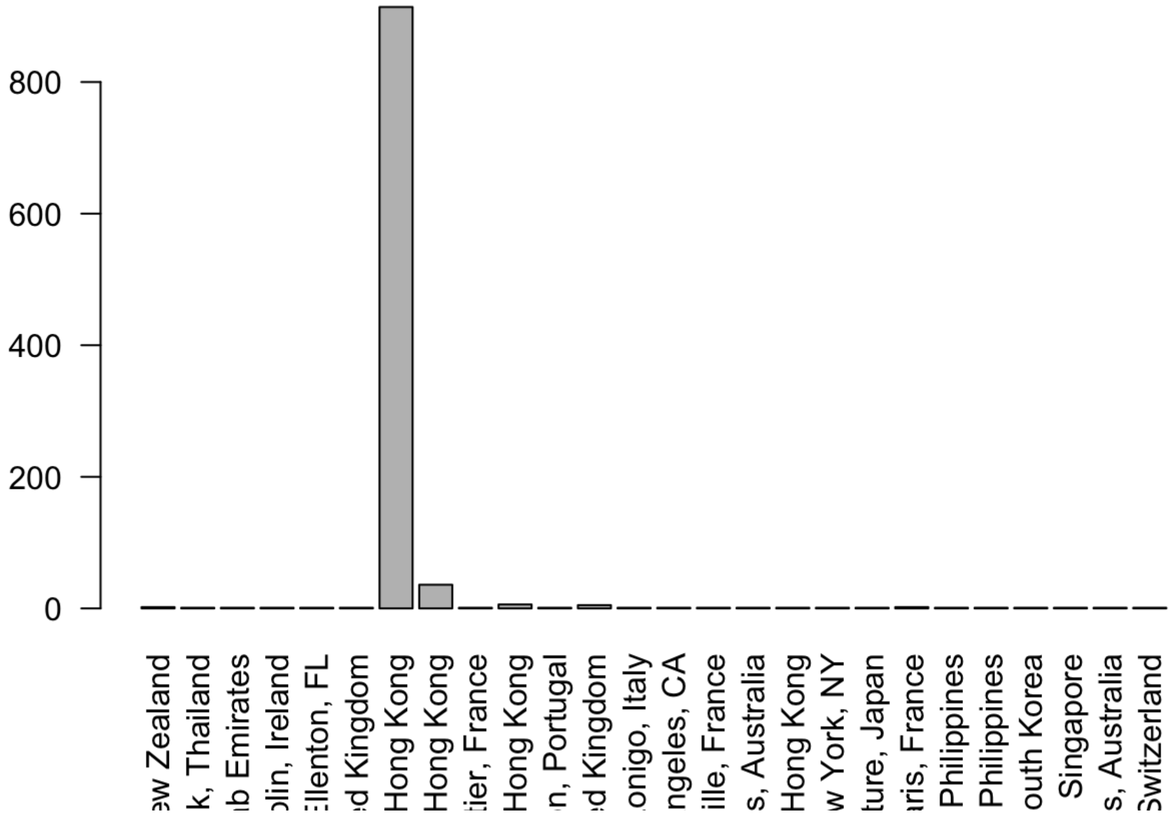
For the superhost column, the criteria for being a superhost is very exigent it asks for many good reviews consistency etc… Basically it means the person is well known and appreciated in the community if the data is missing we can safely assume the host is not a superhost

```
df[is.na(df$host_is_superhost),]$host_is_superhost <- FALSE
```

HOST_LOCATION COLUMN

For host_location :

```
barplot(table(df$host_location), las=2)
```



```
mean(df$host_location == "Hong Kong Island, Hong Kong", na.rm = TRUE)
```

```
## [1] 0.03654822
```

Almost all know host locations are Hong Kong and the amount of missing values is not that high we replace them with Hong Kong

```
df[is.na(df$host_location),]$host_location <- "Hong Kong"
```

HOST_NEIGHBOURHOOD COLUMN

For host_neighbourhood most values are Wan Chai and there are only a few missing values compared to the amount of rows We replace those NAs with Wan Chai

```
df[is.na(df$host_neighbourhood),]$host_neighbourhood <- "Wan Chai"

colSums(is.na(df))
```

```
##                                  description
##                                            0
##                        neighborhood_overview
##                                            0
##                                      host_id
##                                            0
##                                   host_since
##                                            0
##                                host_location
##                                            0
##                           host_response_time
##                                            0
##                           host_response_rate
##                                            0
##                         host_acceptance_rate
##                                            0
##                            host_is_superhost
##                                            0
##                           host_neighbourhood
##                                            0
##                          host_listings_count
##                                            0
##                    host_total_listings_count
##                                            0
##                           host_verifications
##                                            0
##                         host_has_profile_pic
##                                            0
##                       host_identity_verified
##                                            0
##                                     latitude
##                                            0
##                                    longitude
##                                            0
##                                property_type
##                                            0
##                                    room_type
##                                            0
##                                 accommodates
##                                            0
##                               bathrooms_text
##                                            4
##                                     bedrooms
##                                            0
##                                         beds
##                                            0
##                                    amenities
##                                            0
##                                        price
##                                            0
##                               minimum_nights
##                                            0
##                               maximum_nights
##                                            0
##                             has_availability
##                                            0
##                              availability_30
##                                            0
##                              availability_60
##                                            0
##                              availability_90
##                                            0
##                             availability_365
##                                            0
##                            number_of_reviews
##                                            0
##                        number_of_reviews_ltm
##                                            0
##                       number_of_reviews_l30d
##                                            0
##                         review_scores_rating
##                                          962
##                       review_scores_accuracy
##                                          968
##                    review_scores_cleanliness
##                                          968
##                        review_scores_checkin
##                                          968
##                  review_scores_communication
##                                          968
##                       review_scores_location
##                                          968
##                          review_scores_value
##                                          968
##                             instant_bookable
##                                            0
##  calculated_host_listings_count_entire_homes
##                                            0
## calculated_host_listings_count_private_rooms
##                                            0
##  calculated_host_listings_count_shared_rooms
##                                            0
##                            reviews_per_month
##                                          962
```

BATHROOM_TEXT COLUMN

For bathroom_text most apartments have 1 bath and there are only 4 missing values so we replace it with 1 bath except for one host_id where the it is a hotel in description and we know there are 50 beds so there should be 50 bathrooms

```
df$bathrooms_text <-
  replace(df$bathrooms_text,
          df$host_id == 491510217 & is.na(df$bathrooms_text), 50)

df[is.na(df$bathrooms_text),]$bathrooms_text <- "1 bath"
```

## ALL OF THE COLUMNS WITH REVIEW RATINGS

For reviews when there is a missing value it means it received no reviews

We know this by comparing it with the number of reviews column :

```
mean(df[is.na(df$review_scores_rating),]$number_of_reviews == 0)
```

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

Since the amount of missing values is so high it would not be advisable to replace it with the mean or median.

Instead we will just replace it with 0 so that it is easy to filter when we want to do predictions using only the apartments with real review ratings and it will also be easy to replace for a particular model.

There are six rows with review score ratings without any review scores in particular categories (cleanliness, checkin…) this doesn’t affect our process or analysis later on.

```
df[is.na(df$review_scores_rating),]$review_scores_rating <- 0
df[is.na(df$review_scores_accuracy),]$review_scores_accuracy <- 0
df[is.na(df$review_scores_checkin),]$review_scores_checkin <- -0
df[is.na(df$review_scores_cleanliness),]$review_scores_cleanliness <- 0
df[is.na(df$review_scores_communication),]$review_scores_communication <- 0
df[is.na(df$review_scores_location),]$review_scores_location <- 0
df[is.na(df$review_scores_value),]$review_scores_value <- 0
```

For reviews_per_month it is 0 if NA because there were never any reviews.

```
df[is.na(df$reviews_per_month),]$reviews_per_month <- 0
```

COLUMNS WITH “N/A” VALUES AS STRINGS/CHARACTERS

We have to locate those values first

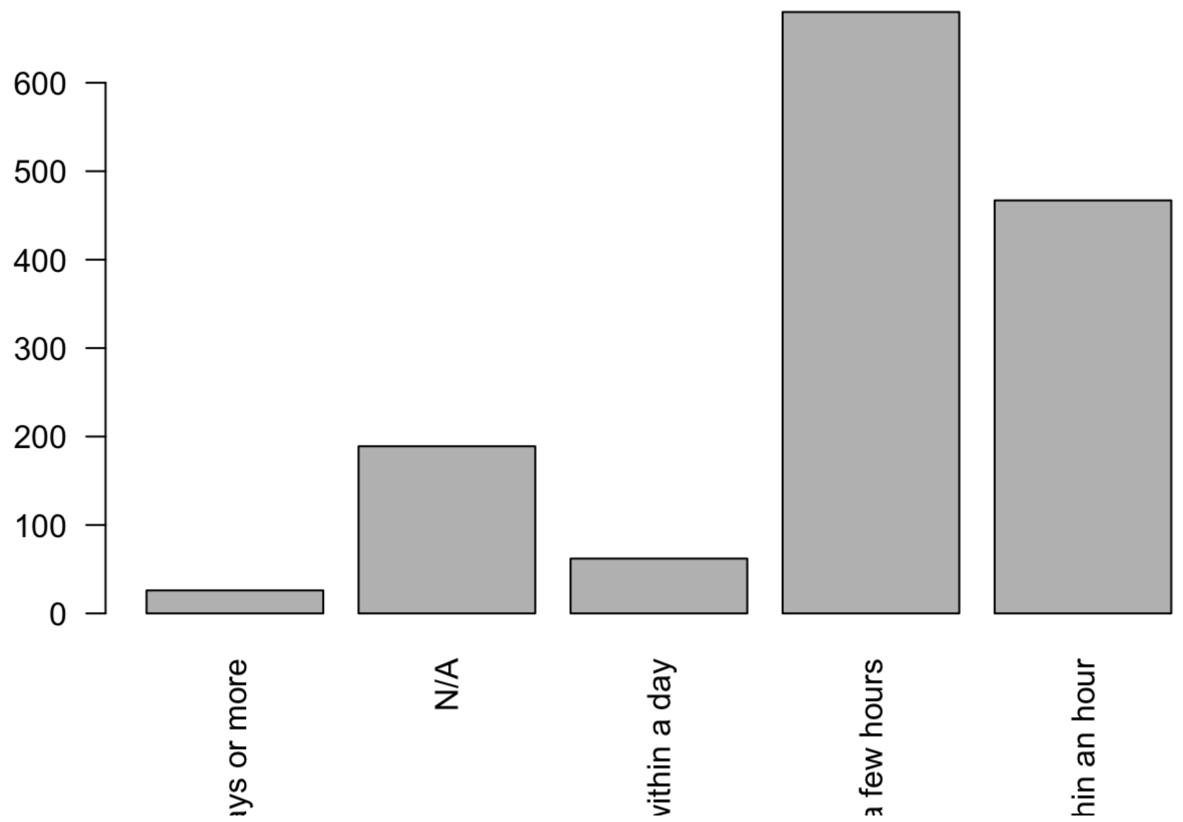
```
na_cols <- sapply(dplyr::select(df, -c('host_since')), # Date format had a bug
                  function(x) any(x == "N/A"))
na_cols <- names(na_cols)[na_cols]
na_cols
```

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

Only three columns have those values

HOST_RESPONSE_TIME COLUMN

```
barplot(table(df$host_response_time), las =2)
```



Most response_time are within a few hours we will use that as replacement because the frequency of N/A is low overall

```
df$host_response_time[df$host_response_time == "N/A"] <- "within a few hours"
```

HOST_RESPONSE_RATE AND HOST_ACCEPTANCE_RATE COLUMNS

```
mean(df$host_response_rate == "N/A")
```

```
## [1] 0.1327247
```

```
mean(df$host_acceptance_rate == "N/A")
```

```
## [1] 0.1264045
```

Around 13% of values are N/A it is low enough for us to consider replacing those values with the median or the mean

We choose the median to add less bias through the outliers

Those two columns are not numerical even though they represent numerical variables.

To deal with that first we change the N/A to the same format with an impossible number then we extract all numbers and reassign to the impossible number the median through filtering

```
df$host_response_rate[df$host_response_rate == "N/A"] <- "-1%"
df$host_acceptance_rate[df$host_acceptance_rate == "N/A"] <- "-1%"
```

We divide by 100 to represent the percentages

```
df$host_response_rate <- (as.numeric(sub("%", "", df$host_response_rate))) /100
df$host_acceptance_rate <- (as.numeric(sub("%", "", df$host_acceptance_rate))) /100

df$host_response_rate[df$host_response_rate < 0] <-
  median(df$host_response_rate[df$host_response_rate >= 0])

df$host_acceptance_rate[df$host_acceptance_rate < 0] <-
  median(df$host_acceptance_rate[df$host_acceptance_rate >= 0])
```

OUTLIERS

We will look at boxplots of numerical variables

First we have to make sure each column has the correct data structure

Price is a character while it should be numeric :

```
df$price <- parse_number(df$price)
```

bathrooms_text contains a numeric information in the number of bathrooms and a useless information in the word bath or baths let’s divide it into multiple columns and keep the important information.

Splitting :

```
df <- separate(df, bathrooms_text, sep = ' ',
                      into = c("bathroom_nb", "bathroom_type", "bath"),
               fill = "right")
```

Removing unecessary information

```
df$bath <- NULL
```

When the bath type is not mentionned we can consider that it is not in a shared apartment so there is no need to precise.

In this case we replace it with normal and convert to a factor for future use

```
df$bathroom_type[df$bathroom_type == "bath"] <- "normal"
df$bathroom_type[df$bathroom_type == "baths"] <- "normal"
df$bathroom_type[is.na(df$bathroom_type)] <- "normal"
df$bathroom_type <- as.factor(df$bathroom_type)

df$bathroom_nb <- as.numeric(df$bathroom_nb)
```

```
## Warning: NAs introduced by coercion
```

```
sum(is.na(df$bathroom_nb))
```

```
## [1] 5
```

5 NAs were created in bathroom_nb after this we replace it with the median

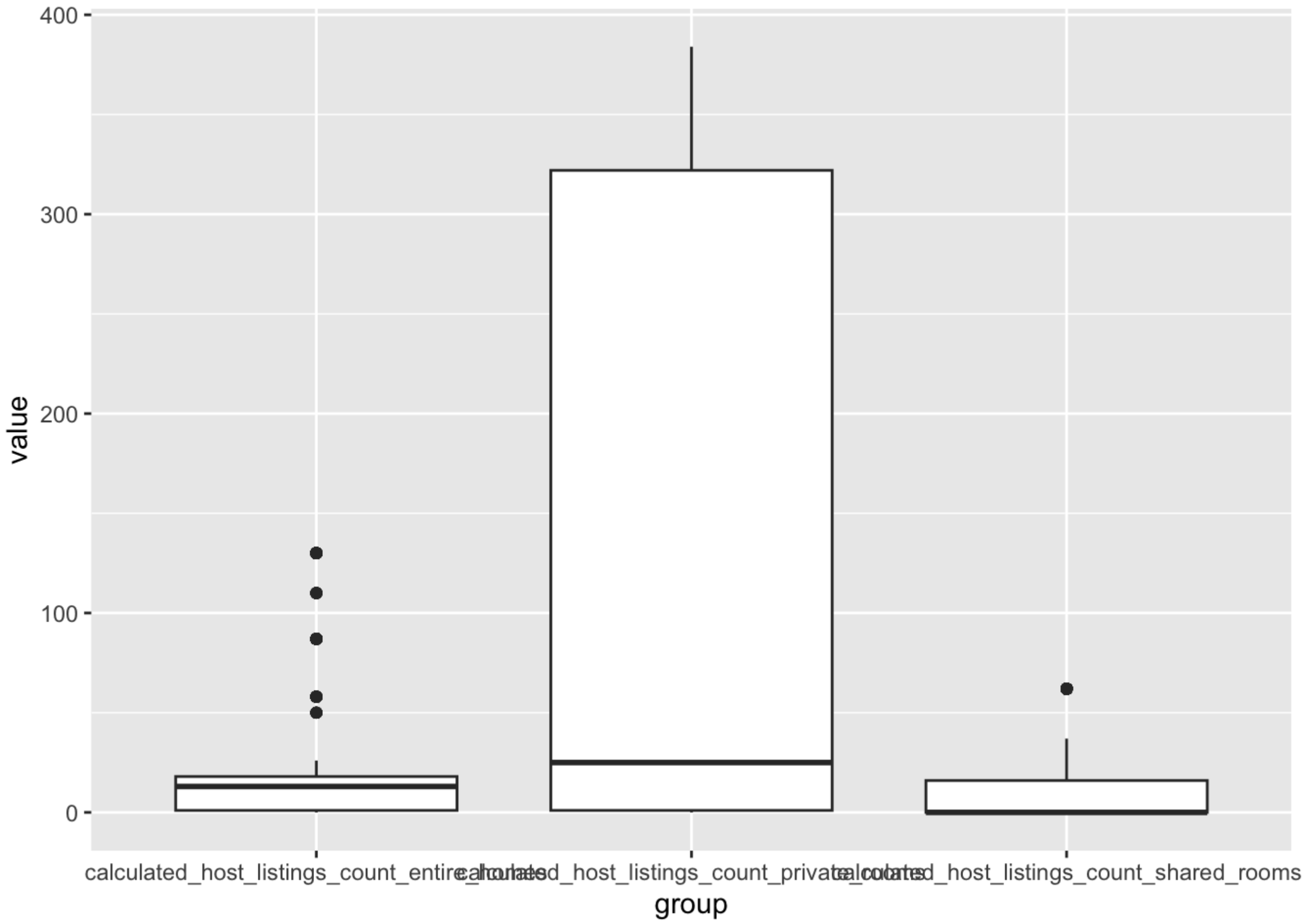
```
df$bathroom_nb[is.na(df$bathroom_nb)] <-
  median(df$bathroom_nb[!is.na(df$bathroom_nb)], na.rm = TRUE)
```

Now let us find the numerical variables and create boxplots to see outliers.

The numerical variables where we can consider outliers are :

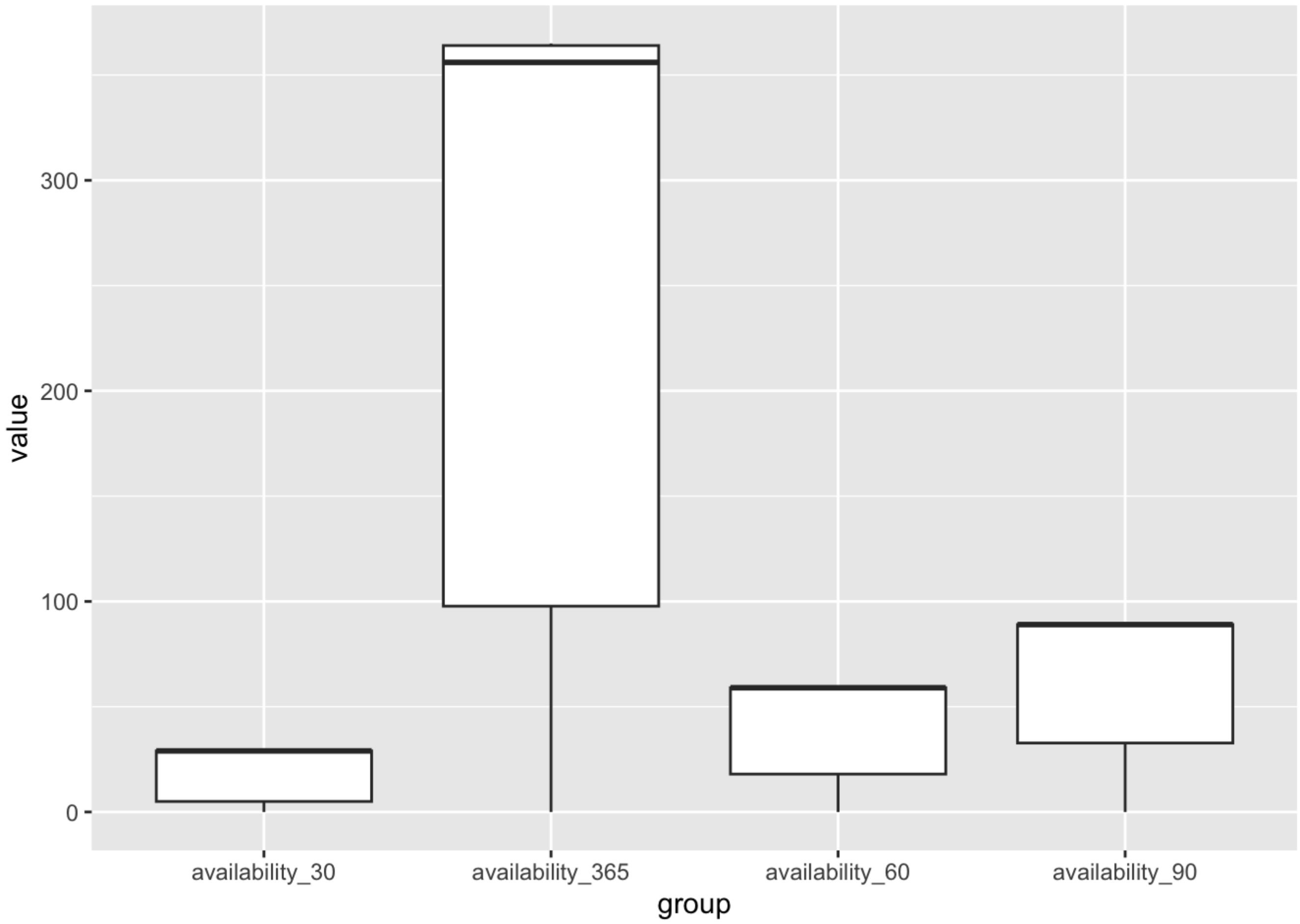
The listing counts, the accommodates, number of bathrooms/bedrooms/beds, price, maximum_nights, availability for the next x amount of days, number of reviews and calculated host listings per category

```
library(ggplot2)

df_long <- gather(df, "group", "value",
                  calculated_host_listings_count_shared_rooms,
                  calculated_host_listings_count_entire_homes,
                  calculated_host_listings_count_private_rooms)

ggplot(df_long, aes(x=group, y=value)) +
  geom_boxplot()
```



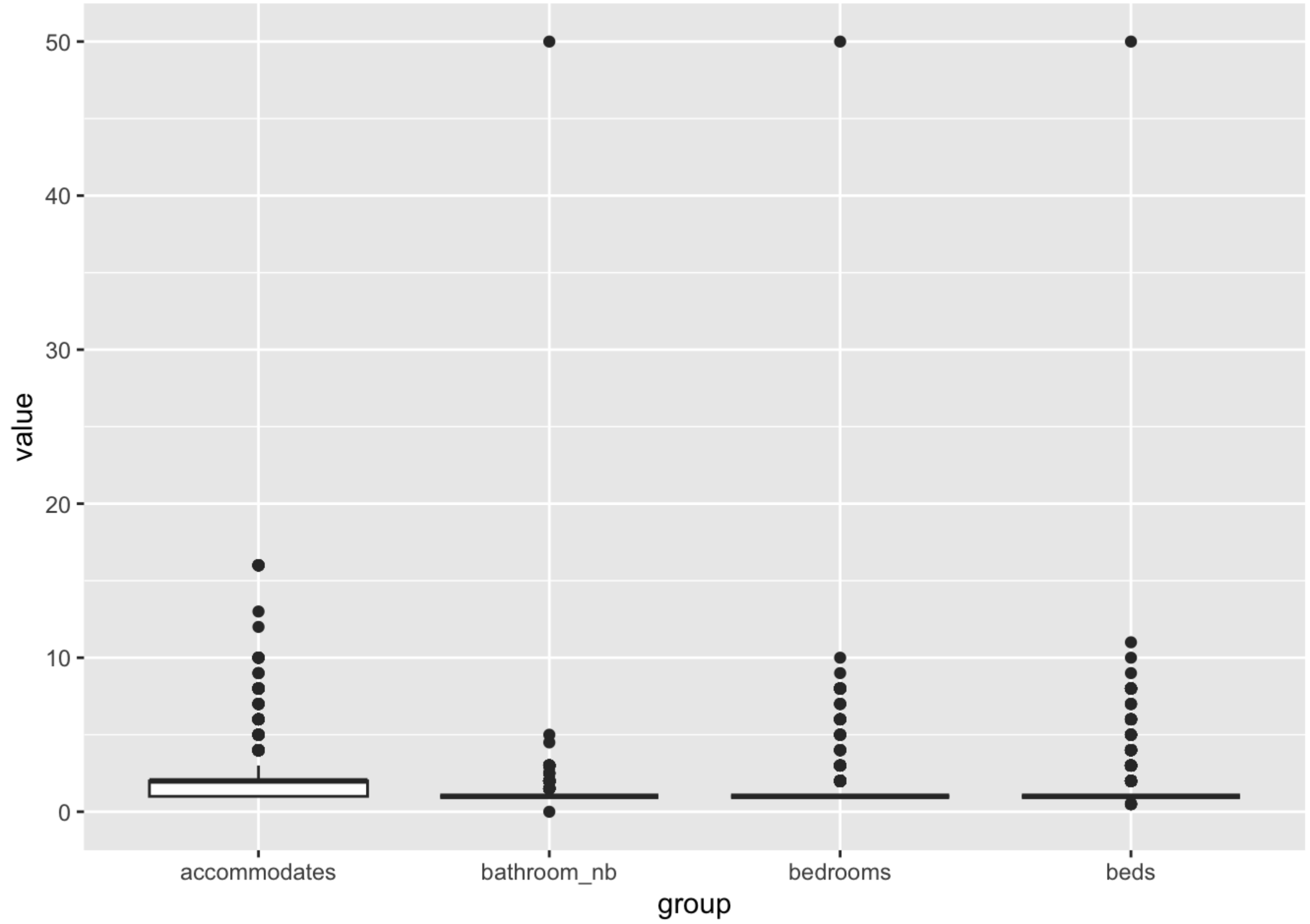
First we can see multiple outliers in the entire_home category for the count of listings

```
df_long <- gather(df, "group", "value",
                  availability_30, availability_60, availability_90,
                  availability_365
                  )

ggplot(df_long, aes(x=group, y=value)) +
  geom_boxplot()
```



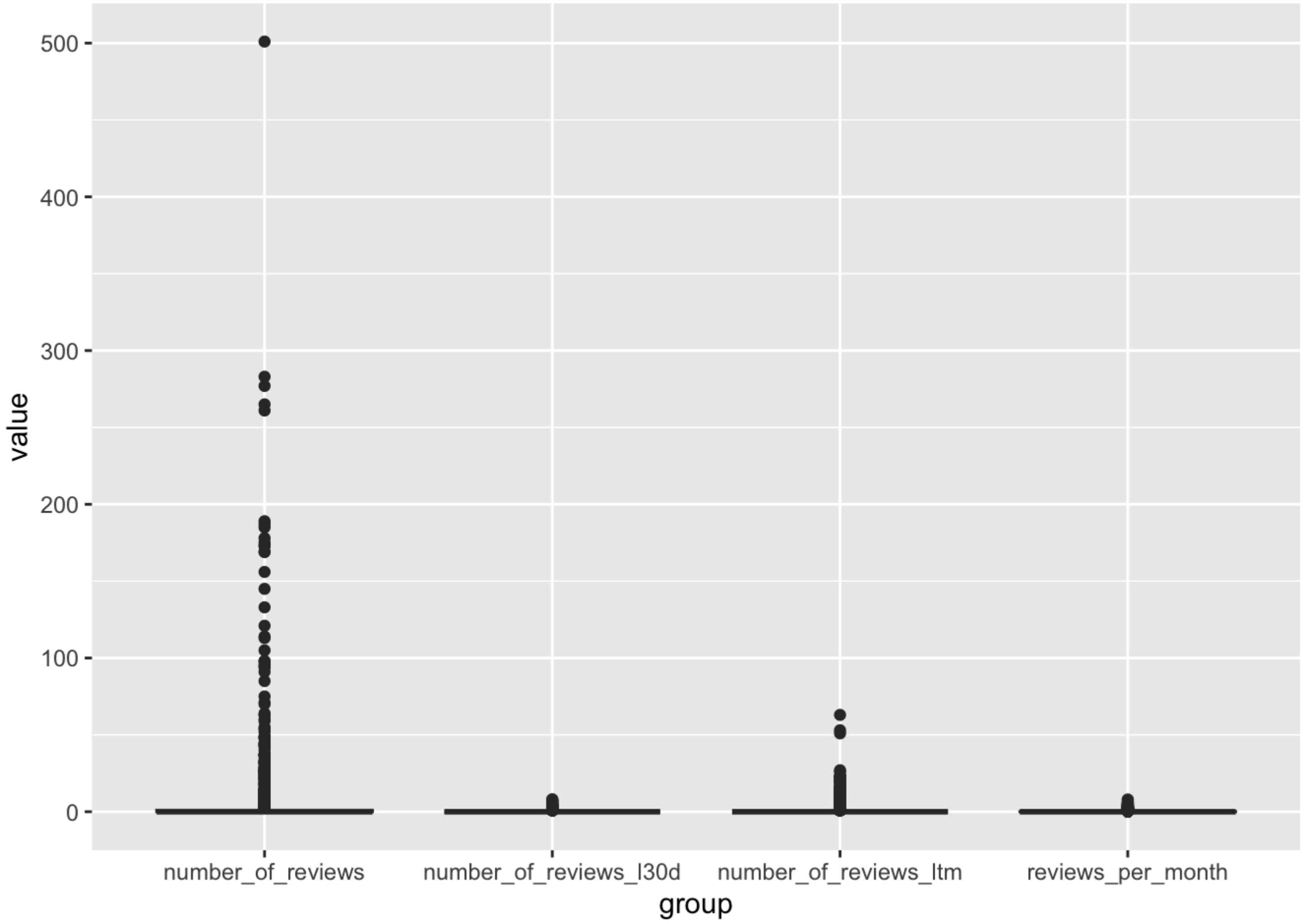
No outliers for availability

```
df_long <- gather(df, "group", "value",
                  accommodates, bathroom_nb, bedrooms, beds )

ggplot(df_long, aes(x=group, y=value)) +
  geom_boxplot()
```



Here because most apartments have the same amount of rooms the few values that goes above look like outliers for the function. Those outliers should not affect the models much.

```
df_long <- gather(df, "group", "value",
                  number_of_reviews, number_of_reviews_l30d,
                  number_of_reviews_ltm, reviews_per_month)

ggplot(df_long, aes(x=group, y=value)) +
  geom_boxplot()
```

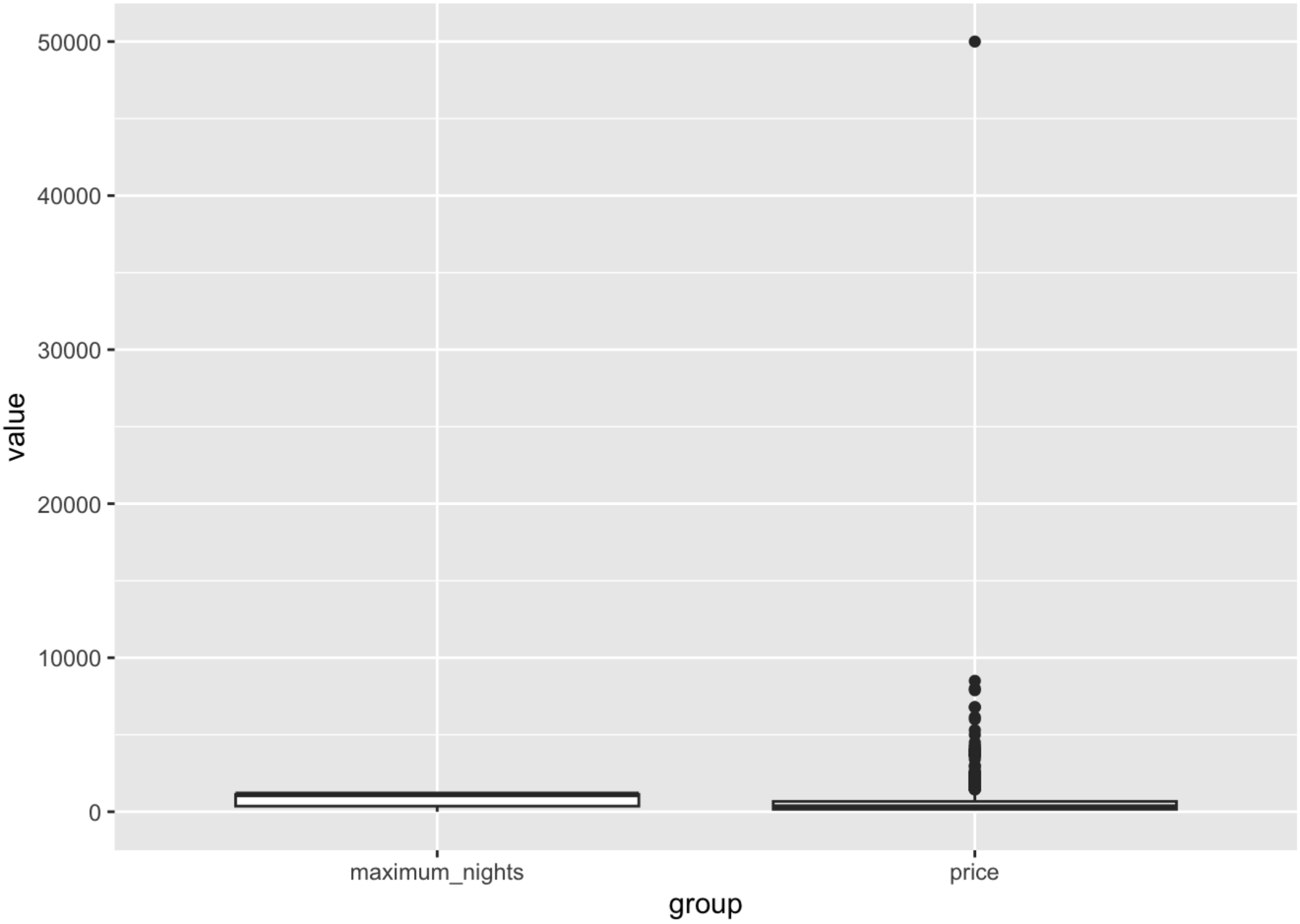


A lot of apartments differentiate themselves by having many reviews.

Maybe those hosts know how to get attention and their apartment becomes massively popular or the apartment itself became popular.

Those outliers should be dealt with later on when building models

```
df_long <- gather(df, "group", "value",
                  price, maximum_nights)

ggplot(df_long, aes(x=group, y=value)) +
  geom_boxplot()
```



Price has many outliers that could influence models that we will consider for future models.

B

In summary we first removed all unnecessary columns, then replaced missing values using the median or most common category if the frequency of NAs is low

In specific cases we replaced missing values by using another variable or replacing with a value that would prevent errors from occurring and would clearly indicate that the value is missing.

Finally for outliers we looked at boxplots to see if we it was necessary to deal with them and consider which variables had problematic outliers for future models.

One last step before going in we create a new dummy column that would indicate whether the apartment received a review or not this seems more important than the review rating itself since most were missing.