AirBnB Listings: An in depth dive into the world of short term sublets

Armandas Bartas, Alex Romanus, Braeden Norman, Gabriel Lanzaro

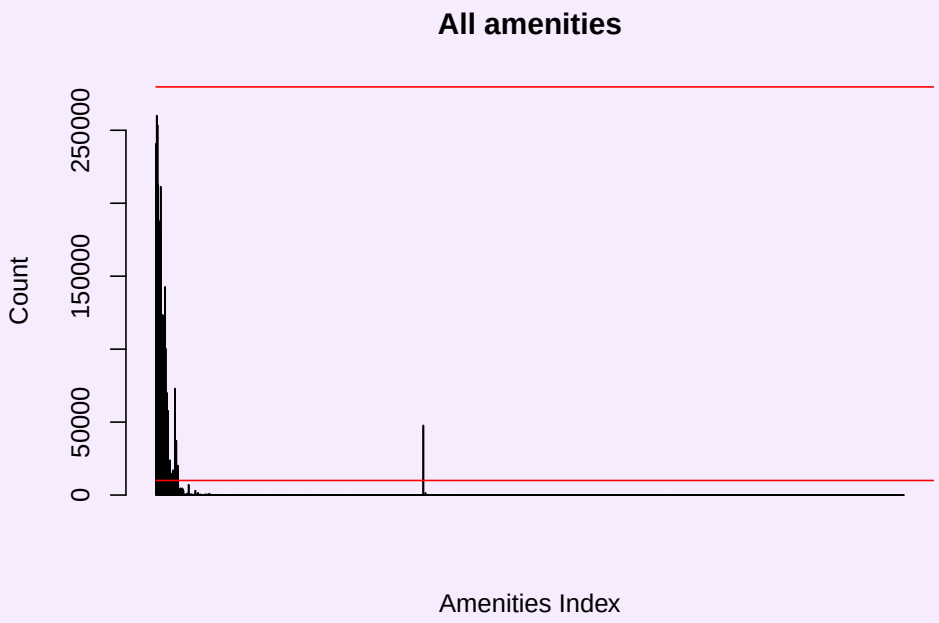
2021-11-09

```
library(tidyverse)
library(testthat)
listings <- read.csv("data/Listings_updated.csv")
amenitiesCount <- read.csv("data/amenities_count.csv")
```

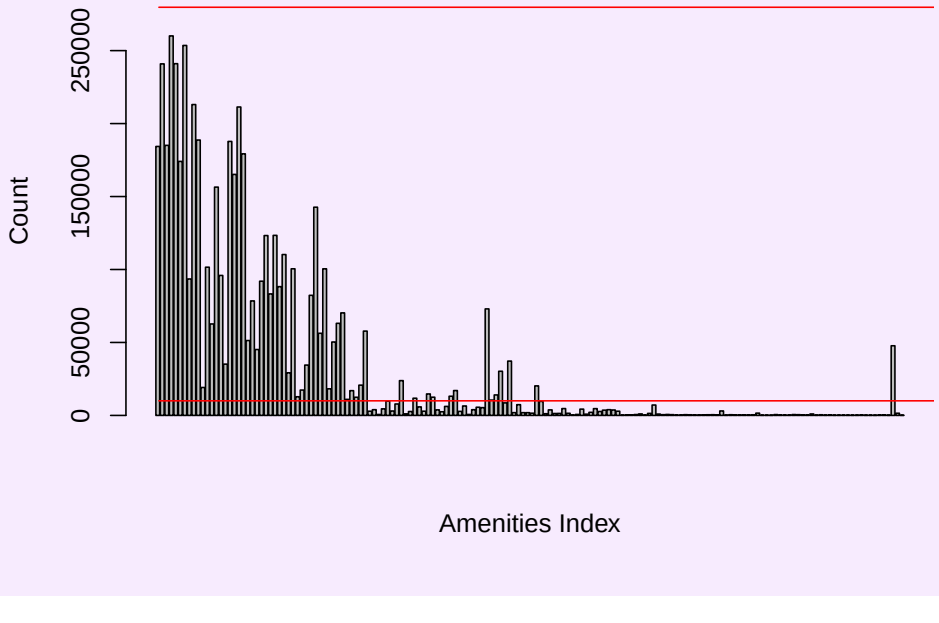
Exploring Amenities

In the original dataset there is a columns that has a list of string of all amenities a listing has. Here we explored these lists to extract useful information for our predictions.
["Heating", "Kitchen", "Washer", "Wifi", "Long term stays allowed"], [“Shampoo”, “Heating”, “Kitchen”, “Essentials”, “Washer”, “Dryer”, “Wifi”, “Long term stays allowed”], [“Heating”, “TV”, “Kitchen”, “Washer”, “Wifi”, “Long term stays allowed”], [“Heating”, “TV”, “Kitchen”, “Wifi”, “Long term stays allowed”], [“Heating”, “TV”, “Kitchen”, “Essentials”, “Hair dryer”, “Washer”, “Dryer”, “Bathtub”, “Wifi”, “Elevator”, “Long term stays allowed”, “Cable TV”]
This is the first 5 observation. We found the list of all the amenities and graphically determined which to include in update dataset. For each included amenities, we will add a new columns labeling whether this listing has this amenities or not. (1/0)

```
barplot(amenitiesCount$V2, ylim = c(0, 280000), main = "All amenities"
        , xlab = "Amenities Index", ylab = "Count")
lines((integer(279712) + 1)*279712, col = "red")
lines((integer(279712) + 1)*10000, col = "red")
```




```
noOnes <- amenitiesCount$V2[-which( amenitiesCount$V2 < 100)]
barplot(noOnes, ylim = c(0, 280000), main = "Amenities with above 100 observations"
        , xlab = "Amenities Index", ylab = "Count")
lines((integer(279712) + 1)*279712, col = "red")
lines((integer(279712) + 1)*10000, col = "red")
```



```
reducedAmenities <- amenitiesCount$V1[which(amenitiesCount$V2 > 10000)]
reducedAmenitiesCount <- amenitiesCount$V2[which(amenitiesCount$V2 > 10000)]
```

The first graph shows the count of all amenities. There is 2865 different amenities in the dataset. A lot of them only have a count of 1, so to better view the distribution we removed all amenities that had below 100 observations. The second graph gives all amenities with a count over 100. The top red line is at total observation of dataset, and the bottom line is at 10,000. 10k was decided to be a good number to remove all amenities with less observations, because this would leave us with a more reasonable size of amenities to add to our dataset as new columns (60 after reduction). All observations we updated with a new columns, 1 for has amenity and 2 for not.

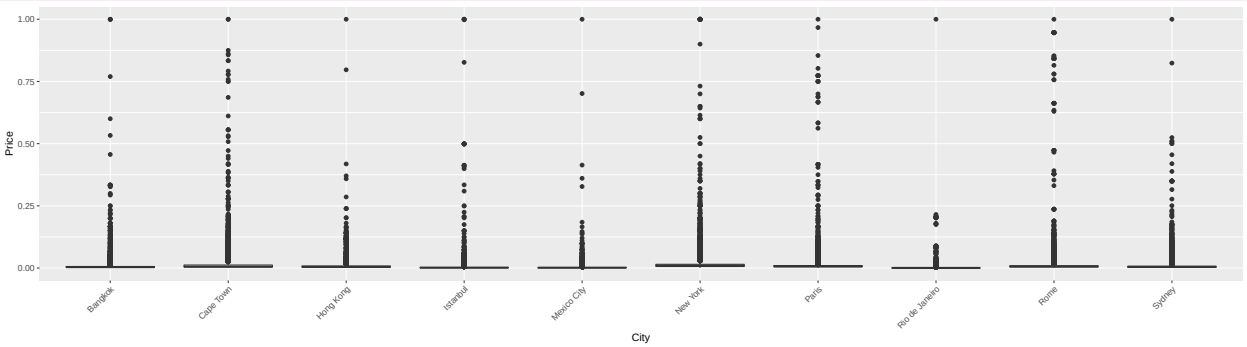
```
#listings[1:5,c(2,4,16,35:38)]
```

Quick preview of what the updated dataset looks like.
Below are the percent TRUE/FALSE values of the selected amenities for each city. Since we are trying to determine the city, based on the listings, seeing the difference of amenities by city will give us an understanding of how useful these amenities will be in our model. The graphs selected were:
Heating, Shampoo, Hair dryer, Smoke alarm, Carbon monoxide alarm, Air conditioning, Free parking on premises, Pool
The usefulness was determine by seeing the distinct difference between each city. For example, in Heating we only have 4 cities that have almost no listings with heating (Bangkok, Hong Kong, Mexico City, and Rio de Janeiro) If we also look at Air conditioning we see that only 2 maybe 3 have little to no listings with AC. (Cape Town, Mexico City, and Paris) Now if we take these two into account we can see that given no AC and Heating, we can be almost positive the listing is Mexico City.

```
selected <- c(1,6,10,14,22,36,54,58)
knitr::opts_chunk$set(fig.width=unit(18,"cm"), fig.height=unit(5,"cm"))
#for (ii in selected) {
#   plt <- ggplot(listings, aes(x = city,
#      fill = factor(listings[,str_replace_all(reducedAmenities[ii], " ", ".")],
#      levels = c(0, 1), labels = c("False", "True")))) +
#      geom_bar(position = "fill") +
#      labs(y = "Percent", fill = "", x = "City",
#      title =  paste(reducedAmenities[ii],"by City")) +
#    theme(axis.text.x = element_text(angle = 90, vjust = 0.5, hjust=1))
#    print(plt)
#}
```

Exploring Price

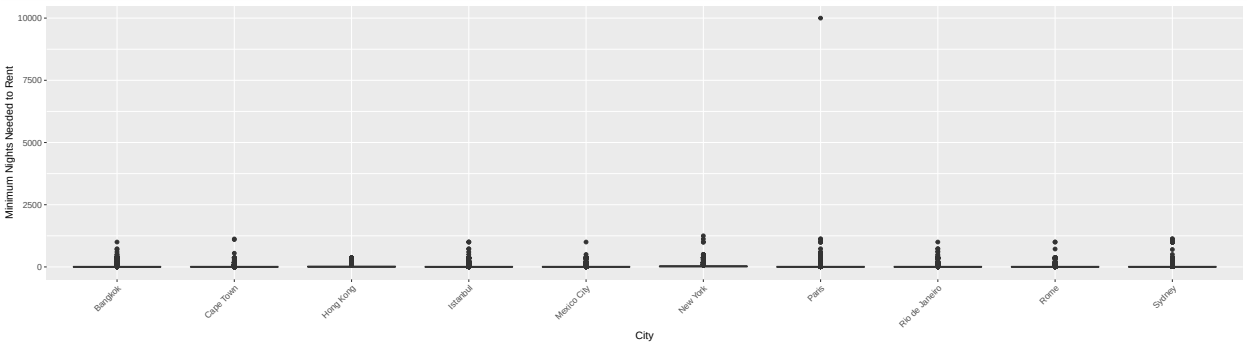
```
library(dplyr)
df <- data.frame(listings)

df <- (df %>% group_by(city) %>% mutate(price_standardized = price/max(price)))

ggplot(df, aes(city, price_standardized)) + geom_boxplot() + xlab('City') +
  ylab('Price') + theme(axis.text.x = element_text(angle = 45, vjust = 1, hjust = 1))
```



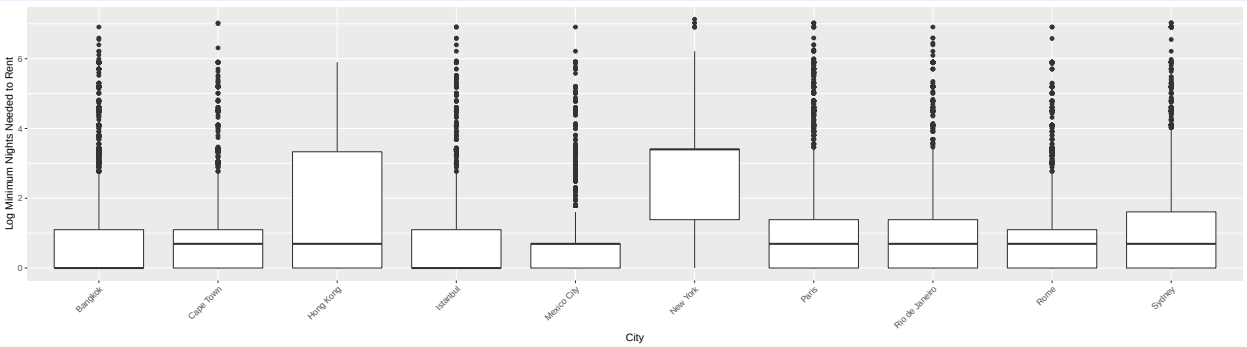
Since prices of the Airbnb's are recorded in each city's country's own currency, to make the prices comparable, it makes sense to standardize the variable. The best way to make all listings' prices comparable is to divide each observation's price by it's city's costliest Airbnb. That way, the spreads of the data separated by city are preserved while making the price variable unitless to allow for comparison of observations between cities.

Exploring Minimum Nights

```
ggplot(df, aes(city, minimum_nights)) + geom_boxplot() +
xlab('City') + ylab('Minimum Nights Needed to Rent') +
theme(axis.text.x = element_text(angle = 45, vjust = 1, hjust = 1))
```




```
index_to_remove <- df[df$minimum_nights > 9000,]$listing_id
df <- subset(df, listing_id != index_to_remove)

ggplot(df, aes(city, log(minimum_nights))) + geom_boxplot() +
xlab('City') + ylab('Log Minimum Nights Needed to Rent') +
theme(axis.text.x = element_text(angle = 45, vjust = 1, hjust = 1))
```

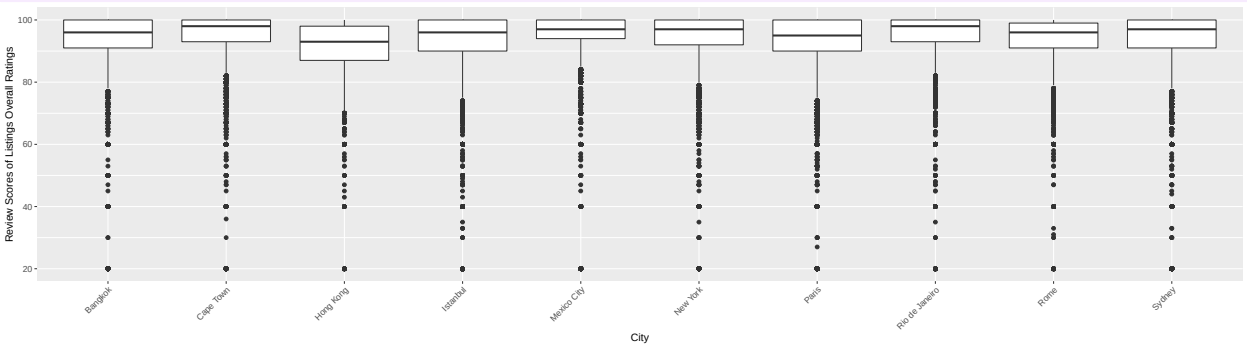


Since no other observations has anywhere near 10000 minimum nights required to rent, which no person would realistically want to rent anyway, it is reasonable to remove this observation from the data set. Furthermore, there are still many outliers, i.e values above the whiskers, preventing the spread from being reasonably assessed, so we apply a log transformation to minimum_nights to make it more readable.

Upon transforming the data, it is clear that most cities have many listings with very few minimum nights required to rent, as many of their boxes show Q1's hovering right at 0. New York is the only exception, with a Q1 value of about 1.5 log days, or about 4.5 days, and a median of about 12 days. Conversely, Istanbul and Bangkok have medians hovering just above 0. These distinct spreads of values of minimum days may make Istanbul, Bangkok, and especially New York much easier to predict if this variable is significant in our model.

Exploring Review Score for Overall Rating

```
df_no_NAN <- df[!is.na(df$review_scores_rating),]

ggplot(df_no_NAN, aes(city, review_scores_rating)) + geom_boxplot() +
xlab('City') + ylab('Review Scores of Listings Overall Ratings') +
theme(axis.text.x = element_text(angle = 45, vjust = 1, hjust = 1))
```



Many observations had to be removed from the data to accommodate the large amount of NANs in the review_score_ratings variable.

The high medians of and Q1 values of Cape Town and Rio suggest these cities may be easier to predict than the rest. Hong Kong also seems to have a significantly lower rating spread of ratings than the rest of the cities, suggesting it may also be easier to predict than the rest.

However, unless the model used is sensitive to subtle differences in the values of these ratings, it like won't be very useful for classification. This is because the spreads of the data for these reviews are all very similar,

with almost all quantilies above a score of 90. This contradicts what we originally thought, as we predicted that overall reviews may differ significantly from city to city. Generally, it seems Airbnb customers seem to give high reviews.

Exploring Review Scores of Location

```
df_no_NAN <- df[!is.na(df$review_scores_location),]
library(janitor)

tabyl(df_no_NAN, city, review_scores_location)
```

```
##            city    10   2 3  4  5   6   7    8    9
##         Bangkok  5598  62 0 33 15 181 217 1230 3832
##       Cape Town 10900  50 1 10  9  94  59  426 1847
##       Hong Kong  2698  29 0 12  4  33  24  175  804
##        Istanbul  7886 147 2 52 28 212 137  713 2042
##     Mexico City 12674  70 0  8  2  78  42  233 1336
##        New York 18819  64 0 27 14 200 151 1245 6220
##           Paris 36662  76 1 22 12 243 239 1882 8833
##  Rio de Janeiro 13657  46 1 20  7 114  63  497 1701
##            Rome 13842  29 1 27 21 138 130 1076 5548
##          Sydney 17759  62 1 23  8 187  89  988 3186
```

```
tabyl(df_no_NAN, city, review_scores_location)%>%
adorn_percentages("col") %>%
adorn_pct_formatting(digits = 2)
```

```
##            city     10      2      3      4      5      6      7      8      9
##         Bangkok  3.98%  9.76%  0.00% 14.10% 12.50% 12.23% 18.85% 14.53% 10.84%
##       Cape Town  7.76%  7.87% 14.29%  4.27%  7.50%  6.35%  5.13%  5.03%  5.23%
##       Hong Kong  1.92%  4.57%  0.00%  5.13%  3.33%  2.23%  2.09%  2.07%  2.27%
##        Istanbul  5.61% 23.15% 28.57% 22.22% 23.33% 14.32% 11.90%  8.42%  5.78%
##     Mexico City  9.02% 11.02%  0.00%  3.42%  1.67%  5.27%  3.65%  2.75%  3.78%
##        New York 13.39% 10.08%  0.00% 11.54% 11.67% 13.51% 13.12% 14.71% 17.60%
##           Paris 26.09% 11.97% 14.29%  9.40% 10.00% 16.42% 20.76% 22.23% 24.99%
##  Rio de Janeiro  9.72%  7.24% 14.29%  8.55%  5.83%  7.70%  5.47%  5.87%  4.81%
##            Rome  9.85%  4.57% 14.29% 11.54% 17.50%  9.32% 11.29% 12.71% 15.69%
##          Sydney 12.64%  9.76% 14.29%  9.83%  6.67% 12.64%  7.73% 11.67%  9.01%
```

Similar to overall review scores, many observations had to be removed to analyze review_scores_location due to the high volume of NANs. Some cities, such as Hong Kong, were significantly effected, as it is far fewer location reviews than, say, Paris.

The largest proportion of high reviews, i.e 9s and 10s, belong to Paris at 24.99% and 26.09% respectively. Also, Istanbul has large shares of low reviews, with 23.15%, 28.57%, and 22.22% of 2s, 3s, and 4s respectively, compared to th other cities. These heavy tails may make it easier to classify these two cities if review scores based on location is a significant predictor in our model.