**Code, 15th October:**

* **Explanatory:** Pinar, deadline: 13th October
* **Mapping:** Rishabh, deadline: 15th October
* **Regression:** Huilun +Rishabh: 15th October

**Memo: Deadline: 17th October**

1. **Executive Summary, NIKO**: Based on your best model, indicate the factors that influence `price\_4\_nights`. This should be written for an intelligent but non-technical audience. All other sections can include technical writing.
2. **Data Exploration and Feature Selection, HEE**: Present key elements of the data, including tables and graphs that help the reader understand the important variables in the dataset. Describe how the data was cleaned and prepared, including feature selection, transformations, interactions, and other approaches you considered.
3. **Model Selection and Validation, JONATHAN**: Describe the model fitting and validation process used. State the model you selected and why they are preferable to other choices.
4. **Findings and Recommendations, WILL:** Interpret the results of the selected model and discuss additional steps that might improve the analysis

**Presentation, NIKO, 17th October**

host\_response\_rate, number

neighbourhood\_cleansed

property\_type

room\_type

bathrooms, number

price, number

host\_is\_superhost

host\_has\_profile\_pic

host\_identity\_verified

has\_availability

instant\_bookable

accommodates >1

bedrooms

beds

minimum\_nights

maximum\_nights

availability\_30

availability\_60

availability\_90

availability\_365

number\_of\_reviews

review\_scores\_rating

review\_scores\_accuracy

review\_scores\_cleanliness

review\_scores\_checkin

review\_scores\_communication

review\_scores\_location

review\_scores\_value

reviews\_per\_month

\* Looking at the raw values.

\* `dplyr::glimpse()`

\* Computing summary statistics of the variables of interest, or finding NAs

\* `mosaic::favstats()`

\* `skimr::skim()`

\* Creating informative visualizations.

\* `ggplot2::ggplot()`

\* `geom\_histogram()` or `geom\_density()` for numeric continuous variables

\* `geom\_bar()` or `geom\_col()` for categorical variables

\* `GGally::ggpairs()` for scaterrlot/correlation matrix

\* Note that you can add transparency to points/density plots in the `aes` call, for example: `aes(colour = gender, alpha = 0.4)`

\*How many variables/columns? How many rows/observations?\*

\*Which variables are numbers?\*

- Which are categorical or \*factor\* variables (numeric or character variables with variables that have a fixed and known set of possible values?

- What are the correlations between variables? Does each scatterplot support a linear relationship between variables? Do any of the correlations appear to be conditional on the value of a categorical variable?

---

title: "Final Group Project: AirBnB analytics"

date: "12 Oct 2021"

author: "Reading Time: About 8 minutes"

output:

html\_document:

highlight: zenburn

theme: flatly

toc: yes

toc\_float: yes

number\_sections: yes

code\_folding: show

---

```{r setup, include=FALSE}

# leave this chunk alone

options(knitr.table.format = "html")

knitr::opts\_chunk$set(warning = FALSE, message = FALSE,

comment = NA, dpi = 300)

```

```{r load-libraries, echo=FALSE}

library(tidyverse) # the usual stuff: dplyr, readr, and other goodies

library(lubridate) # to handle dates

library(GGally) # for correlation-scatter plot matrix

library(ggfortify) # to produce residual diagnostic plots

library(rsample) # to split dataframe in training- & testing sets

library(janitor) # clean\_names()

library(broom) # use broom:augment() to get tidy table with regression output, residuals, etc

library(huxtable) # to get summary table of all models produced

library(kableExtra) # for formatting tables

library(moderndive) # for getting regression tables

library(skimr) # for skim

library(mosaic)

library(leaflet) # for interactive HTML maps

library(tidytext)

library(viridis)

library(vroom)

```

In your final group assignment you have to analyse data about Airbnb listings and fit a model to predict the total cost for two people staying 4 nights in an AirBnB in a city. You can download AirBnB data from [insideairbnb.com](http://insideairbnb.com/get-the-data.html){target="\_blank"}; it was originally scraped from airbnb.com.

The following [Google sheet](https://docs.google.com/spreadsheets/d/1QrR-0PUGVWvDiVQL4LOk7w-xXwiDnM3dDtW6k15Hc7s/edit?usp=sharing) shows which cities you can use; please choose one of them and add your group name next to it, e.g., A7, B13. No city can have more than 2 groups per stream working on it; if this happens, I will allocate study groups to cities with the help of R's sampling.

All of the listings are a GZ file, namely they are archive files compressed by the standard GNU zip (gzip) compression algorithm. You can download, save and extract the file if you wanted, but `vroom::vroom()` or `readr::read\_csv()` can immediately read and extract this kind of a file. You should prefer `vroom()` as it is faster, but if vroom() is limited by a firewall, please use `read\_csv()` instead.

`vroom` will download the \*.gz zipped file, unzip, and provide you with the dataframe.

```{r load\_data, echo=FALSE, message=FALSE, warning=FALSE, cache=TRUE}

# use cache=TRUE so you dont donwload the data everytime you knit

listings <- vroom("http://data.insideairbnb.com/turkey/marmara/istanbul/2021-09-30/data/listings.csv.gz") %>%

clean\_names()

```

Even though there are many variables in the dataframe, here is a quick description of some of the variables collected, and you can find a [data dictionary here](https://docs.google.com/spreadsheets/d/1iWCNJcSutYqpULSQHlNyGInUvHg2BoUGoNRIGa6Szc4/edit#gid=982310896)

- `price` = cost per night

- `property\_type`: type of accommodation (House, Apartment, etc.)

- `room\_type`:

- Entire home/apt (guests have entire place to themselves)

- Private room (Guests have private room to sleep, all other rooms shared)

- Shared room (Guests sleep in room shared with others)

- `number\_of\_reviews`: Total number of reviews for the listing

- `review\_scores\_rating`: Average review score (0 - 100)

- `longitude` , `latitude`: geographical coordinates to help us locate the listing

- `neighbourhood\*`: three variables on a few major neighbourhoods in each city

# Exploratory Data Analysis (EDA)

##Raw Value Inspection

There are 23,019 entries and 74 characteristics for each entry.

```{r}

glimpse(listings)

```

##Summary Statistics & Finding Missing Entries

Through skimming the data, we see that there are 23 character, 5 date, 9 logical, and 37 numeric variables. Numeric variables include but not limited to number of accommodates that a property can host, bedrooms, minimum & maximum nights for stay, review scores, and availability within upcoming days. Categoric variables include but not limited to property and room types, verification status of the host, number of bathrooms in a property, and neighborhood.

For certain variables like bathrooms and license, all the entries are missing. Therefore, before dropping off the missing values immediately, first we need to eliminate the variables that we do not think will be useful for our analysis. Then, we can drop the missing values for the variables of interest. Additionally, some of the important variables like price is stored as a character. So, we need to convert it into numeric variable before we start building our model (see data wrangling).

Computing the summary statistics for number of reviews, minimum/maximum nights, and availability of the property for the next 30 and 365 days, we learn about the distribution of values (min, max, Q1, Q3, median) as well as the basic explanatory statistics like mean, standard deviation, count and number of missing values. Since we base our analysis on a travel data, we should eliminate the properties that are available only for the long-term rent. When we check the explanatory statistics and the box plot for the minimum night for stay (see data wrangling), the maximum value is 730. This means, there is a property who is only available if the guest is going to stay for minimum 730 days. Therefore, we filter our dataframe according to travel conditions.

###Finding Missing Values & Variable Types

```{r}

skimr::skim(listings)

```

###Summary Statistics for Certain Numeric Variables

```{r}

round(mosaic::favstats(unlist(listings ["number\_of\_reviews"])), digits=2)

```

```{r}

round(mosaic::favstats(unlist(listings ["minimum\_nights"])), digits = 2)

```

```{r}

round(mosaic::favstats(unlist(listings ["maximum\_nights"])), digits = 2)

```

```{r}

round(mosaic::favstats(unlist(listings ["availability\_30"])), digits = 2)

```

```{r}

round(mosaic::favstats(unlist(listings ["availability\_365"])), digits = 2)

```

```{r}

round(mosaic::favstats(unlist(listings ["review\_scores\_rating"])), digits = 2)

```

##Data Wrangling

###Changing the Types of Variables of Interest

```{r}

#Change the type of Price to Numeric

listings <- listings %>%

mutate(price = parse\_number(price))

#Change the type of Host Response Rate to Numeric

listings$host\_response\_rate <- round((as.numeric(gsub('.{1}$', '', listings$host\_response\_rate)))/100, digits=2)

#Change the type of Bathroom to Numeric

listings$bathrooms\_text <- as.numeric(sub("^(.{1}).\*", "\\1",listings$bathrooms\_text ) )

listings <- listings %>%

mutate(bathrooms=bathrooms\_text ) #changed the name of bathroom\_text to bathrooms

listings %>%

select(bathrooms)

```

###Filtering & Choosing Variables of Interest & Dropping Missing Values

```{r}

#Box plot for Minimum Nights & Filtering for Travel Conditions

ggplot(listings, aes(y= minimum\_nights)) +

geom\_boxplot()+

theme\_bw() +

labs(title= "Box Plot for Minimum Nights",

y="Minimum Nights")

#Most common values for minimum nights is 1. As discussed above, there are properties that are only available for the long-term rent. For example, there is a property with minimum night stay of 730 days. Since we consider travel conditions, we should eliminate those by filtering minimum\_nights <= 4 below.

```

``` {r}

#Defining 5 Categories for Property Type

listings %>%

group\_by(property\_type) %>%

summarise(count= count (property\_type)) %>%

arrange(desc(count)) %>%

mutate(percentage= count/(sum(count)))

listings <- listings %>%

mutate(prop\_type\_simplified = case\_when(

property\_type %in% c("Entire rental unit","Entire serviced apartment", "Entire condominium (condo)","Entire residential home") ~ property\_type,

TRUE ~ "Other"

))

listings %>%

count(property\_type, prop\_type\_simplified) %>%

arrange(desc(n))

```

```{r}

#Creating a new dataframe with variables of interest

new\_listings<- listings %>%

select(price,

prop\_type\_simplified,

room\_type,

neighbourhood\_cleansed,

host\_response\_rate,

accommodates,

bathrooms,

bedrooms,

beds,

minimum\_nights,

maximum\_nights,

availability\_30,

availability\_60,

availability\_90,

availability\_365,

number\_of\_reviews,

review\_scores\_rating,

review\_scores\_accuracy,

review\_scores\_cleanliness,

review\_scores\_checkin,

review\_scores\_communication,

review\_scores\_location,

review\_scores\_value,

instant\_bookable,

reviews\_per\_month,

has\_availability,

host\_identity\_verified,

host\_has\_profile\_pic,

host\_is\_superhost) %>%

filter(accommodates >1,minimum\_nights <= 4)

#Since we are interested in properties for 2 people, we need to eliminate the properties for only one person

#Our analysis is based on the travel data, so we filter the entries with minimum\_nights <= 4

#Dropping missing values of the new dataframe

new\_listings <-drop\_na(new\_listings)

head(new\_listings)

```

##Informative Visualization

As we see from the graph, the distribution of price is positively skewed with most of the values <1,000 and some very large values like +100,000. Computing the explanatory statistics, we see that man price is 650.96 while the maximum price is 103,133.

```{r fig.width = 12, fig.height = 12}

#Density Plot for Price

ggplot(new\_listings, aes(x= price)) +

geom\_density(alpha=0.2) +

theme\_bw() +

labs (

title = "Density Plot for Price",

y = "Density",

x= "Price")

ggplot(new\_listings, aes(y= price)) +

geom\_boxplot() +

theme\_bw() +

labs (

title = "Box Plot for Price",

y = "Price")

round(mosaic::favstats(unlist(listings ["price"])), digits = 2)

```

Plotting the histogram for availability within x= 30, 60, 90 and 365 days, we see that in each case, most of the properties are either not available or available +(x-5) days. Exceptionally, in the case of availability within the next 365, there is a seasonal pattern in every 100 days. So, some of the properties are available within next 100 days, 200 days, or +300 days.

```{r fig.width = 12, fig.height = 12}

#Histogram for Availability within next 30 days

ggplot(new\_listings, aes(x= availability\_30)) +

geom\_histogram(alpha=0.2) +

theme\_bw() +

labs (

title = "Histogram for Availability Within Next 30 Days",

x= "Availability within next 30 days",

y= "Number of Properties")

```

```{r fig.width = 12, fig.height = 12}

#Histogram for Availability within next 60 days

ggplot(new\_listings, aes(x= availability\_60)) +

geom\_histogram(alpha=0.2) +

theme\_bw() +

labs (

title = "Histogram for Availability Within Next 60 Days",

x= "Availability within next 60 days",

y= "Number of Properties")

```

```{r fig.width = 12, fig.height = 12}

#Histogram for Availability within next 90 days

ggplot(new\_listings, aes(x= availability\_90)) +

geom\_histogram(alpha=0.2) +

theme\_bw() +

labs (

title = "Histogram for Availability Within Next 90 Days",

x= "Availability within next 90 days",

y= "Number of Properties")

```

```{r fig.width = 12, fig.height = 12}

#Histogram for Availability within next 365 days

ggplot(new\_listings, aes(x= availability\_365)) +

geom\_histogram(alpha=0.2) +

theme\_bw() +

labs (

title = "Histogram for Availability Within Next 365 Days",

x= "Availability within next 365 days",

y= "Number of Properties")

```

Since we filtered data based on travel conditions (minimum\_nights<=4), we have 4 values for this variable; 1, 2, 3 and 4. Majority of the properties have a minimum 1-night stay.

```{r fig.width = 12, fig.height = 12}

#Histogram for Minimum Nights

ggplot(new\_listings, aes(x= minimum\_nights)) +

geom\_histogram(alpha=0.2) +

theme\_bw() +

labs (

title = "Histogram for Minimum Nights",

x= "Minimum Nights",

y= "Number of Properties")

```

On average, guests make around 10 reviews per property. However, there are some properties with significant number of reviews (e.g., 606). Considering the density plot for number of reviews, data is positively skewed with majority of data points being around 0 to 100 reviews per property.

```{r fig.width = 12, fig.height = 12}

#Density plot for Number of Reviews

ggplot(new\_listings, aes(x= number\_of\_reviews)) +

geom\_density(alpha=0.2) +

theme\_bw() +

labs (

title = "Density Plot for Number of Reviews",

y = "Density",

x="Number of Reviews")

round(mosaic::favstats(unlist(listings ["number\_of\_reviews"])), digits = 2)

```

Majority of the reviews are centered around 5 with a mean 4,46 and median 4,8. There are 11,983 entries for review scores rating but we have almost two times more listings. Therefore, we can assume that, guests provide rating when they are satisfied with the property/host/service, etc. If they think that the service is on average or below average, they do not provide as much as feedback as they do when they are satisfied.

We would expect higher concentration around rating=1 & 2, because people tend to give feedback when they are not satisfied with the service they get. Alternatively, there is not much dissatisfaction and guests are happy with the service.

```{r fig.width = 12, fig.height = 12}

#Density Plot for Review Scores Rating

ggplot(new\_listings, aes(x= review\_scores\_rating)) +

geom\_density(alpha=0.2) +

theme\_bw() +

labs (

title = "Density Plot for Review Scores Rating",

y = "Density",

x= "Review Scores Rating")

round(mosaic::favstats(unlist(listings ["review\_scores\_rating"])), digits = 2)

```

37% of properties listed accommodate 2 people. The percentage properties with +6 accommodates is very low approximately 6%.

```{r fig.width = 12, fig.height = 12}

#Histogram for Number of Accommodates

ggplot(new\_listings, aes(x= accommodates )) +

geom\_histogram(alpha=0.2) +

theme\_bw() +

labs (

title = "Number of Accommodates in a Property",

x= "Accomodates",

y= "Number of Properties")

new\_listings %>%

group\_by(accommodates) %>%

summarise(count=count(accommodates)) %>%

mutate(percentage= count/sum(count) \*100) %>%

arrange(desc(percentage))

```

There are 5 property types (simplified version). Within those 5 units, "Entire Rental Unit" has the majority of the weight, 48,5% followed by "Other" 37,3%.

```{r fig.width = 12, fig.height = 12}

#Bar chart for Property Type

ggplot(new\_listings, aes(y= prop\_type\_simplified )) +

geom\_bar(alpha=0.2) +

theme\_bw() +

labs (

title = "Property Type",

x="Number of Properties",

y="Property Type")

new\_listings %>%

group\_by(prop\_type\_simplified) %>%

summarise(count=count(prop\_type\_simplified)) %>%

mutate(percentage= count/sum(count) \*100) %>%

arrange(desc(percentage))

```

There are 4 types of rooms, "Entire home", "Private room", "Hotel room", and "Shared room". Approximately 70% of listings' room type is "Entire home".

```{r fig.width = 12, fig.height = 12}

#Bar chart for Room Type

ggplot(new\_listings, aes(y= room\_type )) +

geom\_bar(alpha=0.2) +

theme\_bw() +

labs (

title = "Room Type",

x= "Number of Properties",

y= "Room Type")

new\_listings %>%

group\_by(room\_type) %>%

summarise(count=count(room\_type)) %>%

mutate(percentage= count/sum(count) \*100) %>%

arrange(desc(percentage))

```

As expected, neighborhood is an important factor determining the price of the property. Majority of the properties are in Beyoglu & Fatih which is the most touristic region of Istanbul.

```{r fig.width = 12, fig.height = 12}

#Bar chart for Neighborhood

ggplot(new\_listings, aes(y= neighbourhood\_cleansed )) +

geom\_bar(alpha=0.2) +

theme\_bw() +

labs (

title = "Neighborhood",

x= "Number of Properties",

y= "Neighborhood")

```

There are approximately 4000 instant bookable properties and 3000 not instantly bookable ones.

```{r fig.width = 12, fig.height = 12}

#Bar chart for Instant Bookable

ggplot(new\_listings, aes(x= instant\_bookable)) +

geom\_bar(alpha=0.2) +

theme\_bw() +

labs (

title = "Instant Bookability",

x= "Status of Instant Bookability",

y= "Number of Properties")

new\_listings %>%

count(instant\_bookable)

```

Identity verification might be a significant factor in determining the price of a property. But, most of the properties', 83%, host is verified.

```{r fig.width = 12, fig.height = 12}

#Bar chart for Host Identity Verification

ggplot(new\_listings, aes(x= host\_identity\_verified )) +

geom\_bar(alpha=0.2) +

theme\_bw() +

labs (

title = "Host Identity Verification",

y= "Number of Identifications",

x= "Identity Verification Status")

new\_listings %>%

count(host\_identity\_verified)

```

Does each scatterplot support a linear relationship between variables? Do any of the correlations appear to be conditional on the value of a categorical variable?

##Correlations

Computing the correlation between the numeric variables, we see that review scores are strongly positively correlated with each other. For example, the correlation between review\_scores\_accuracy and review\_scores\_rating is 0,875 and the correlation between review\_scores\_communication and review\_scores\_cleanliness is 0,833. To prevent collinearity, we shoudl not include all review scores in our regression models, but rathr just include review\_scores\_rating.

Moreover, the correlation between the availability of 30, 60, 90, and 365 days is high, so similar exclusion to prevent collinearity holds for these variables. For example, the correlation between the availabilty within 60 and 30 days is 0,917 which is almost 1.

Interestingly, we see that the number of reviews has relatively low correlation with other numeric variables whşch is around 10% (except number of reviews per month).

The strongest correlation with price is number of accommodates which is 0,244.

```{r fig.width = 20, fig.height = 20}

correlation\_data <- new\_listings %>%

dplyr::select(where(is.numeric)) %>%

ggpairs()

correlation\_data

```

# Mapping

Visualisations of feature distributions and their relations are key to understanding a data set, and they can open up new lines of exploration. While we do not have time to go into all the wonderful geospatial visualisations one can do with R, you can use the following code to start with a map of your city, and overlay all AirBnB coordinates to get an overview of the spatial distribution of AirBnB rentals. For this visualisation we use the `leaflet` package, which includes a variety of tools for interactive maps, so you can easily zoom in-out, click on a point to get the actual AirBnB listing for that specific point, etc.

The following code, having downloaded a dataframe `listings` with all AirbnB listings in Milan, will plot on the map all AirBnBs where `minimum\_nights` is less than equal to four (4). You could learn more about `leaflet`, by following [the relevant Datacamp course on mapping with leaflet](https://www.datacamp.com/courses/interactive-maps-with-leaflet-in-r)

```{r, out.width = '80%'}

leaflet(data = filter(listings, minimum\_nights <= 4)) %>%

addProviderTiles("OpenStreetMap.Mapnik") %>%

addCircleMarkers(lng = ~longitude,

lat = ~latitude,

radius = 1,

fillColor = "blue",

fillOpacity = 0.4,

popup = ~listing\_url,

label = ~property\_type)

```

# Regression Analysis

For the target variable $Y$, we will use the cost for two people to stay at an Airbnb location for four (4) nights.

Create a new variable called `price\_4\_nights` that uses `price`, and `accomodates` to calculate the total cost for two people to stay at the Airbnb property for 4 nights. This is the variable $Y$ we want to explain.

Use histograms or density plots to examine the distributions of `price\_4\_nights` and `log(price\_4\_nights)`. Which variable should you use for the regression model? Why?

Fit a regression model called `model1` with the following explanatory variables: `prop\_type\_simplified`, `number\_of\_reviews`, and `review\_scores\_rating`.

- Interpret the coefficient `review\_scores\_rating` in terms of `price\_4\_nights`.

- Interpret the coefficient of `prop\_type\_simplified` in terms of `price\_4\_nights`.

We want to determine if `room\_type` is a significant predictor of the cost for 4 nights, given everything else in the model. Fit a regression model called model2 that includes all of the explananatory variables in `model1` plus `room\_type`.

## Further variables/questions to explore on our own

Our dataset has many more variables, so here are some ideas on how you can extend your analysis

1. Are the number of `bathrooms`, `bedrooms`, `beds`, or size of the house (`accomodates`) significant predictors of `price\_4\_nights`? Or might these be co-linear variables?

1. Do superhosts `(host\_is\_superhost`) command a pricing premium, after controlling for other variables?

1. Some hosts allow you to immediately book their listing (`instant\_bookable == TRUE`), while a non-trivial proportion don't. After controlling for other variables, is `instant\_bookable` a significant predictor of `price\_4\_nights`?

1. For all cities, there are 3 variables that relate to neighbourhoods: `neighbourhood`, `neighbourhood\_cleansed`, and `neighbourhood\_group\_cleansed`. There are typically more than 20 neighbourhoods in each city, and it wouldn't make sense to include them all in your model. Use your city knowledge, or ask someone with city knowledge, and see whether you can group neighbourhoods together so the majority of listings falls in fewer (5-6 max) geographical areas. You would thus need to create a new categorical variabale `neighbourhood\_simplified` and determine whether location is a predictor of `price\_4\_nights`

1. What is the effect of `avalability\_30` or `reviews\_per\_month` on `price\_4\_nights`, after we control for other variables?

## Diagnostics, collinearity, summary tables

As you keep building your models, it makes sense to:

1. Check the residuals, using `autoplot(model\_x)`

1. As you start building models with more explanatory variables, make sure you use `car::vif(model\_x)`` to calculate the \*\*Variance Inflation Factor (VIF)\*\* for your predictors and determine whether you have colinear variables. A general guideline is that a VIF larger than 5 or 10 is large, and your model may suffer from collinearity. Remove the variable in question and run your model again without it.

1. Create a summary table, using `huxtable` (https://mfa2022.netlify.app/example/modelling\_side\_by\_side\_tables/) that shows which models you worked on, which predictors are significant, the adjusted $R^2$, and the Residual Standard Error.

1. Finally, you must use the best model you came up with for prediction. Suppose you are planning to visit the city you have been assigned to over reading week, and you want to stay in an Airbnb. Find Airbnb's in your destination city that are apartments with a private room, have at least 10 reviews, and an average rating of at least 90. Use your best model to predict the total cost to stay at this Airbnb for 4 nights. Include the appropriate 95% interval with your prediction. Report the point prediction and interval in terms of `price\_4\_nights`.

- if you used a log(price\_4\_nights) model, make sure you anti-log to convert the value in $. You can read more about [hot to interpret a regression model when some variables are log transformed here](https://stats.idre.ucla.edu/other/mult-pkg/faq/general/faqhow-do-i-interpret-a-regression-model-when-some-variables-are-log-transformed/)

# Deliverables

- By midnight on Monday 18 Oct 2021, you must upload on Canvas a short presentation (max 4-5 slides) with your findings, as some groups will be asked to present in class. You should present your Exploratory Data Analysis, as well as your best model. In addition, you must upload on Canvas your final report, written using R Markdown to introduce, frame, and describe your story and findings. You should include the following in the memo:

1. Executive Summary: Based on your best model, indicate the factors that influence `price\_4\_nights`.

This should be written for an intelligent but non-technical audience. All

other sections can include technical writing.

2. Data Exploration and Feature Selection: Present key elements of the data, including tables and

graphs that help the reader understand the important variables in the dataset. Describe how the

data was cleaned and prepared, including feature selection, transformations, interactions, and

other approaches you considered.

3. Model Selection and Validation: Describe the model fitting and validation process used. State

the model you selected and why they are preferable to other choices.

4. Findings and Recommendations: Interpret the results of the selected model and discuss

additional steps that might improve the analysis

Remember to follow R Markdown etiquette rules and style; don't have the Rmd output extraneous messages or warnings, include summary tables in nice tables (use `kableExtra`), and remove any placeholder texts from past Rmd templates; in other words, (i.e. I don't want to see stuff I wrote in your final report.)

# Rubric

Your work will be assessed on a rubric which you can find here

```{r rubric, echo=FALSE, out.width="100%"}

knitr::include\_graphics(here::here("images", "rubric.png"), error = FALSE)

```

# Acknowledgements

- The data for this project is from [insideairbnb.com](insideairbnb.com)