EDA\_Airbnb\_analysis

Ayush Singh

2023-04-09

# load required packages  
library(tidyverse)

## Warning: package 'tidyverse' was built under R version 4.2.3

## Warning: package 'tibble' was built under R version 4.2.3

## Warning: package 'tidyr' was built under R version 4.2.3

## Warning: package 'readr' was built under R version 4.2.3

## Warning: package 'purrr' was built under R version 4.2.3

## Warning: package 'dplyr' was built under R version 4.2.3

## Warning: package 'forcats' was built under R version 4.2.3

## Warning: package 'lubridate' was built under R version 4.2.3

## ── Attaching core tidyverse packages ──────────────────────── tidyverse 2.0.0 ──  
## ✔ dplyr 1.1.1 ✔ 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 ]8;;http://conflicted.r-lib.org/conflicted package]8;; to force all conflicts to become errors

library(ggplot2)  
library(dplyr)  
library(reshape2)

## Warning: package 'reshape2' was built under R version 4.2.3

##   
## Attaching package: 'reshape2'  
##   
## The following object is masked from 'package:tidyr':  
##   
## smiths

#' loading of data  
my\_data <- read.csv("C:/Users/Hp/Downloads/Airbnb NYC 2019.csv")

#' display the contents of the data frame  
head(my\_data)

## id name host\_id host\_name  
## 1 2539 Clean & quiet apt home by the park 2787 John  
## 2 2595 Skylit Midtown Castle 2845 Jennifer  
## 3 3647 THE VILLAGE OF HARLEM....NEW YORK ! 4632 Elisabeth  
## 4 3831 Cozy Entire Floor of Brownstone 4869 LisaRoxanne  
## 5 5022 Entire Apt: Spacious Studio/Loft by central park 7192 Laura  
## 6 5099 Large Cozy 1 BR Apartment In Midtown East 7322 Chris  
## neighbourhood\_group neighbourhood latitude longitude room\_type price  
## 1 Brooklyn Kensington 40.64749 -73.97237 Private room 149  
## 2 Manhattan Midtown 40.75362 -73.98377 Entire home/apt 225  
## 3 Manhattan Harlem 40.80902 -73.94190 Private room 150  
## 4 Brooklyn Clinton Hill 40.68514 -73.95976 Entire home/apt 89  
## 5 Manhattan East Harlem 40.79851 -73.94399 Entire home/apt 80  
## 6 Manhattan Murray Hill 40.74767 -73.97500 Entire home/apt 200  
## minimum\_nights number\_of\_reviews last\_review reviews\_per\_month  
## 1 1 9 2018-10-19 0.21  
## 2 1 45 2019-05-21 0.38  
## 3 3 0 NA  
## 4 1 270 2019-07-05 4.64  
## 5 10 9 2018-11-19 0.10  
## 6 3 74 2019-06-22 0.59  
## calculated\_host\_listings\_count availability\_365  
## 1 6 365  
## 2 2 355  
## 3 1 365  
## 4 1 194  
## 5 1 0  
## 6 1 129

## check structure and summary of data-set  
  
#The output shows that my\_data is a data frame with 48,895 observations and 16 variables  
str(my\_data)

## 'data.frame': 48895 obs. of 16 variables:  
## $ id : int 2539 2595 3647 3831 5022 5099 5121 5178 5203 5238 ...  
## $ name : chr "Clean & quiet apt home by the park" "Skylit Midtown Castle" "THE VILLAGE OF HARLEM....NEW YORK !" "Cozy Entire Floor of Brownstone" ...  
## $ host\_id : int 2787 2845 4632 4869 7192 7322 7356 8967 7490 7549 ...  
## $ host\_name : chr "John" "Jennifer" "Elisabeth" "LisaRoxanne" ...  
## $ neighbourhood\_group : chr "Brooklyn" "Manhattan" "Manhattan" "Brooklyn" ...  
## $ neighbourhood : chr "Kensington" "Midtown" "Harlem" "Clinton Hill" ...  
## $ latitude : num 40.6 40.8 40.8 40.7 40.8 ...  
## $ longitude : num -74 -74 -73.9 -74 -73.9 ...  
## $ room\_type : chr "Private room" "Entire home/apt" "Private room" "Entire home/apt" ...  
## $ price : int 149 225 150 89 80 200 60 79 79 150 ...  
## $ minimum\_nights : int 1 1 3 1 10 3 45 2 2 1 ...  
## $ number\_of\_reviews : int 9 45 0 270 9 74 49 430 118 160 ...  
## $ last\_review : chr "2018-10-19" "2019-05-21" "" "2019-07-05" ...  
## $ reviews\_per\_month : num 0.21 0.38 NA 4.64 0.1 0.59 0.4 3.47 0.99 1.33 ...  
## $ calculated\_host\_listings\_count: int 6 2 1 1 1 1 1 1 1 4 ...  
## $ availability\_365 : int 365 355 365 194 0 129 0 220 0 188 ...

#' The summary provides information on the minimum and maximum values  
#' as well as the median and mean values  
summary(my\_data)

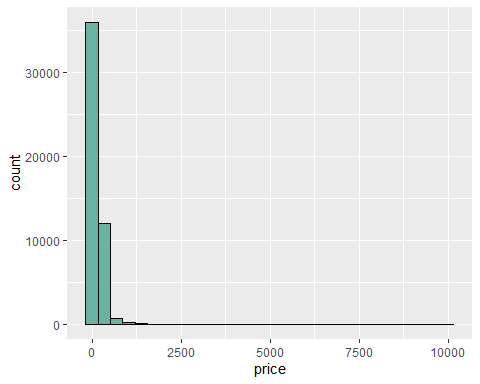
## id name host\_id host\_name   
## Min. : 2539 Length:48895 Min. : 2438 Length:48895   
## 1st Qu.: 9471945 Class :character 1st Qu.: 7822033 Class :character   
## Median :19677284 Mode :character Median : 30793816 Mode :character   
## Mean :19017143 Mean : 67620011   
## 3rd Qu.:29152178 3rd Qu.:107434423   
## Max. :36487245 Max. :274321313   
##   
## neighbourhood\_group neighbourhood latitude longitude   
## Length:48895 Length:48895 Min. :40.50 Min. :-74.24   
## Class :character Class :character 1st Qu.:40.69 1st Qu.:-73.98   
## Mode :character Mode :character Median :40.72 Median :-73.96   
## Mean :40.73 Mean :-73.95   
## 3rd Qu.:40.76 3rd Qu.:-73.94   
## Max. :40.91 Max. :-73.71   
##   
## room\_type price minimum\_nights number\_of\_reviews  
## Length:48895 Min. : 0.0 Min. : 1.00 Min. : 0.00   
## Class :character 1st Qu.: 69.0 1st Qu.: 1.00 1st Qu.: 1.00   
## Mode :character Median : 106.0 Median : 3.00 Median : 5.00   
## Mean : 152.7 Mean : 7.03 Mean : 23.27   
## 3rd Qu.: 175.0 3rd Qu.: 5.00 3rd Qu.: 24.00   
## Max. :10000.0 Max. :1250.00 Max. :629.00   
##   
## last\_review reviews\_per\_month calculated\_host\_listings\_count  
## Length:48895 Min. : 0.010 Min. : 1.000   
## Class :character 1st Qu.: 0.190 1st Qu.: 1.000   
## Mode :character Median : 0.720 Median : 1.000   
## Mean : 1.373 Mean : 7.144   
## 3rd Qu.: 2.020 3rd Qu.: 2.000   
## Max. :58.500 Max. :327.000   
## NA's :10052   
## availability\_365  
## Min. : 0.0   
## 1st Qu.: 0.0   
## Median : 45.0   
## Mean :112.8   
## 3rd Qu.:227.0   
## Max. :365.0   
##

# shows the data type of each column, the first few values of each column  
glimpse(my\_data)

## Rows: 48,895  
## Columns: 16  
## $ id <int> 2539, 2595, 3647, 3831, 5022, 5099, 512…  
## $ name <chr> "Clean & quiet apt home by the park", "…  
## $ host\_id <int> 2787, 2845, 4632, 4869, 7192, 7322, 735…  
## $ host\_name <chr> "John", "Jennifer", "Elisabeth", "LisaR…  
## $ neighbourhood\_group <chr> "Brooklyn", "Manhattan", "Manhattan", "…  
## $ neighbourhood <chr> "Kensington", "Midtown", "Harlem", "Cli…  
## $ latitude <dbl> 40.64749, 40.75362, 40.80902, 40.68514,…  
## $ longitude <dbl> -73.97237, -73.98377, -73.94190, -73.95…  
## $ room\_type <chr> "Private room", "Entire home/apt", "Pri…  
## $ price <int> 149, 225, 150, 89, 80, 200, 60, 79, 79,…  
## $ minimum\_nights <int> 1, 1, 3, 1, 10, 3, 45, 2, 2, 1, 5, 2, 4…  
## $ number\_of\_reviews <int> 9, 45, 0, 270, 9, 74, 49, 430, 118, 160…  
## $ last\_review <chr> "2018-10-19", "2019-05-21", "", "2019-0…  
## $ reviews\_per\_month <dbl> 0.21, 0.38, NA, 4.64, 0.10, 0.59, 0.40,…  
## $ calculated\_host\_listings\_count <int> 6, 2, 1, 1, 1, 1, 1, 1, 1, 4, 1, 1, 3, …  
## $ availability\_365 <int> 365, 355, 365, 194, 0, 129, 0, 220, 0, …

# checked the distribution of the "price" variable using a histogram.  
ggplot(my\_data, aes(x = price)) +  
 geom\_histogram(fill = "#69b3a2", color = "black")

## `stat\_bin()` using `bins = 30`. Pick better value with `binwidth`.



### data cleaning\_\_\_\_\_\_\_\_\_\_

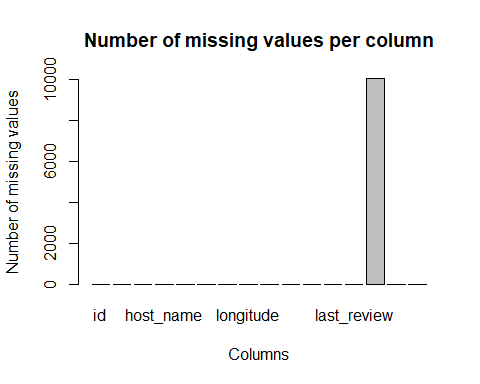
# for finding the sum of total null values  
sum(is.na(my\_data))

## [1] 10052

# Compute the number of missing values for each column  
sum\_null\_values <- colSums(is.na(my\_data))  
print(sum\_null\_values)

## id name   
## 0 0   
## host\_id host\_name   
## 0 0   
## neighbourhood\_group neighbourhood   
## 0 0   
## latitude longitude   
## 0 0   
## room\_type price   
## 0 0   
## minimum\_nights number\_of\_reviews   
## 0 0   
## last\_review reviews\_per\_month   
## 0 10052   
## calculated\_host\_listings\_count availability\_365   
## 0 0

# Plot the number of missing values for each column  
barplot(sum\_null\_values, main = "Number of missing values per column",   
 xlab = "Columns", ylab = "Number of missing values")



#from the plot we understand that the null values are appear in the column “reviews\_per\_month” only. #“so in next step we drop all the null values in the data-set

## Drop rows with Null values in the "reviews\_per\_month" column  
my\_data <- my\_data[complete.cases(my\_data$reviews\_per\_month), ]

sum(is.na(my\_data))

## [1] 0

the null values is eliminated

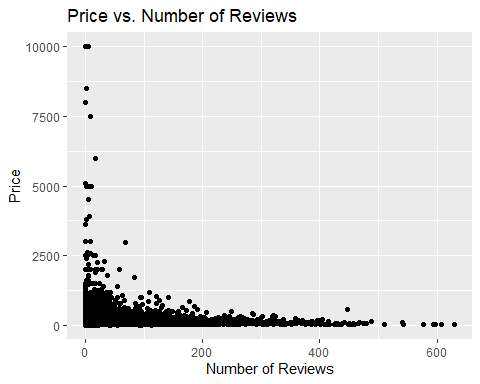
#for cheaking the duplicates in the data-set  
duplicates <- my\_data[duplicated(my\_data),]

#there are no duplicated in the data-set

#let’s explore the relationships between variables using scatter-plots and correlation matrices

#scatter-plot of price vs. number of reviews

ggplot(my\_data, aes(x = number\_of\_reviews, y = price)) +  
 geom\_point() +  
 labs(x = "Number of Reviews", y = "Price",   
 title = "Price vs. Number of Reviews")

 #Based on the above data, we can conclude that as the price increases, # the number of reviews decreases. Additionally, # the majority of reviews come from the price range of 0-2500. # Therefore, we can infer that price is a significant factor when booking a room.

# Compute correlation matrix of numerical variables

#’in the data-set to see which variables are most strongly correlated with “price”.

num\_cols <- sapply(my\_data, is.numeric)  
 num\_data <- my\_data[, num\_cols]  
 corr\_matrix <- cor(num\_data)  
 print(corr\_matrix)

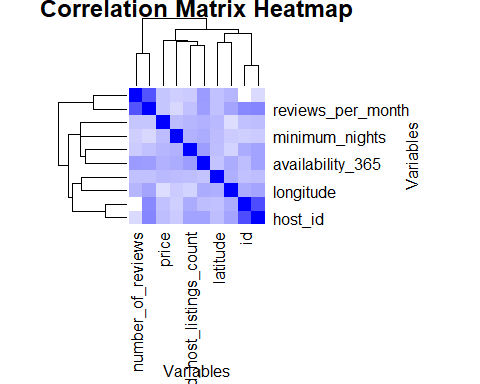
## id host\_id latitude  
## id 1.000000000 0.591528876 -0.010013040  
## host\_id 0.591528876 1.000000000 0.012946249  
## latitude -0.010013040 0.012946249 1.000000000  
## longitude 0.103149693 0.141094653 0.088151639  
## price -0.006646538 0.006269474 0.031317773  
## minimum\_nights -0.073901751 -0.051692700 0.024807912  
## number\_of\_reviews -0.329864763 -0.141819190 -0.008746500  
## reviews\_per\_month 0.291827896 0.296416584 -0.010141595  
## calculated\_host\_listings\_count 0.098482270 0.149411818 0.004325651  
## availability\_365 0.006429969 0.155081719 -0.022228087  
## longitude price minimum\_nights  
## id 0.10314969 -0.006646538 -0.07390175  
## host\_id 0.14109465 0.006269474 -0.05169270  
## latitude 0.08815164 0.031317773 0.02480791  
## longitude 1.00000000 -0.155360501 -0.05541832  
## price -0.15536050 1.000000000 0.02550577  
## minimum\_nights -0.05541832 0.025505770 1.00000000  
## number\_of\_reviews 0.05468116 -0.035938148 -0.06936781  
## reviews\_per\_month 0.14594803 -0.030608349 -0.12170220  
## calculated\_host\_listings\_count -0.09332451 0.052903171 0.07347894  
## availability\_365 0.10254019 0.078233616 0.10168573  
## number\_of\_reviews reviews\_per\_month  
## id -0.32986476 0.291827896  
## host\_id -0.14181919 0.296416584  
## latitude -0.00874650 -0.010141595  
## longitude 0.05468116 0.145948027  
## price -0.03593815 -0.030608349  
## minimum\_nights -0.06936781 -0.121702201  
## number\_of\_reviews 1.00000000 0.549867506  
## reviews\_per\_month 0.54986751 1.000000000  
## calculated\_host\_listings\_count -0.05978440 -0.009421162  
## availability\_365 0.19355663 0.185790961  
## calculated\_host\_listings\_count availability\_365  
## id 0.098482270 0.006429969  
## host\_id 0.149411818 0.155081719  
## latitude 0.004325651 -0.022228087  
## longitude -0.093324508 0.102540192  
## price 0.052903171 0.078233616  
## minimum\_nights 0.073478943 0.101685731  
## number\_of\_reviews -0.059784397 0.193556633  
## reviews\_per\_month -0.009421162 0.185790961  
## calculated\_host\_listings\_count 1.000000000 0.182910972  
## availability\_365 0.182910972 1.000000000

# we can see that there are some variables that are positively correlated with each other

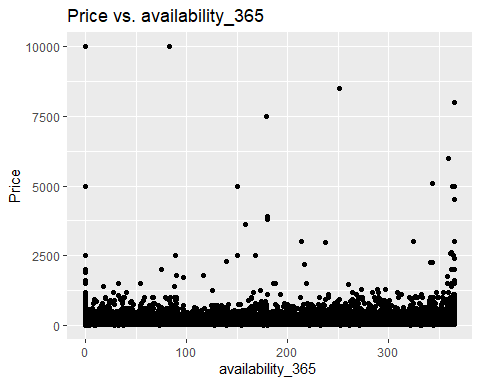
#such as host\_id and calculated\_host\_listings\_count, which have a correlation coefficient of 0.149.

#there are variables that are negatively correlated with each other, such as number\_of\_reviews and #availability\_365, which have a correlation coefficient of -0.194.

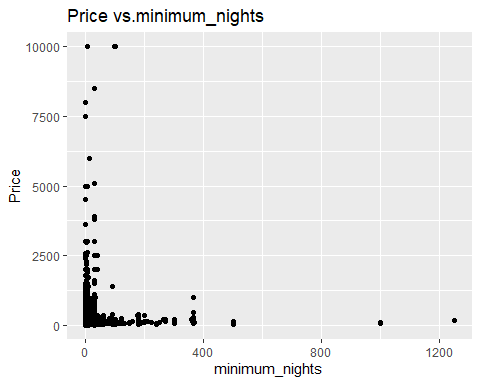
# plot a heatmap of the correlation matrix  
 heatmap(corr\_matrix,   
 col = colorRampPalette(c("white", "blue"))(100),   
 scale = "none",   
 symm = TRUE,  
 margins = c(10,10),  
 main = "Correlation Matrix Heatmap",  
 xlab = "Variables",   
 ylab = "Variables")

 #to find the outliers

# Scatter-plot of price vs. availability\_365   
 ggplot(my\_data, aes(x = availability\_365 , y = price)) +  
 geom\_point() +  
 labs(x = "availability\_365 ", y = "Price",   
 title = "Price vs. availability\_365 ")

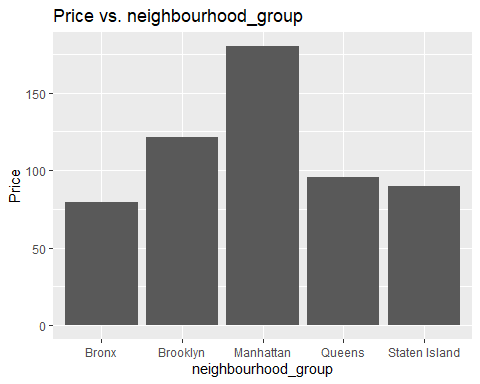
 # from the scatter plot we able to understand that 0-2500 price range most are availability # all around 360 days

# Scatter-plot of price vs minimum\_nights   
 ggplot(my\_data, aes(x =minimum\_nights, y = price)) +  
 geom\_point() +  
 labs(x = "minimum\_nights ", y = "Price",   
 title = "Price vs.minimum\_nights")

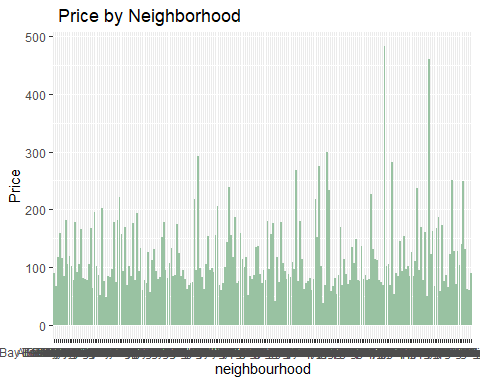
 #from the scatter plot we conclude that the price range 0-2500 is the price range which have minimum night spends where as you see as price increases the minimum night stays was decreasing as this indicates the price factor is the most important thing when comes to hotel choosing.

#now we have to check whether there is any relationship between price and categorical variable or not.

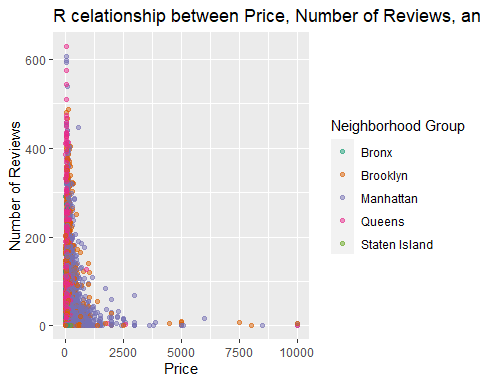
#price vs neighbourhood\_group  
 ggplot(my\_data, aes(x = neighbourhood\_group, y = price)) +  
 geom\_bar(stat = "summary", fun = "mean") +  
 labs(x = "neighbourhood\_group", y = "Price",   
 title = "Price vs. neighbourhood\_group")

 #from above graph we easily conclude that ‘Manhattan’ is the most expensive place where price is very high followed by ‘Brooklyn’ & ‘Queens’ the least expensive is ‘Bronx’ and ‘Staten island’.

#price vs neighborhood  
   
 ggplot(data = my\_data, aes(x =price , y = neighbourhood)) +  
 stat\_summary(fun = "mean", geom = "bar", fill = "#99c2a2") +  
 coord\_flip() +  
 labs(x = "Price", y = "neighbourhood",   
 title = " Price by Neighborhood")

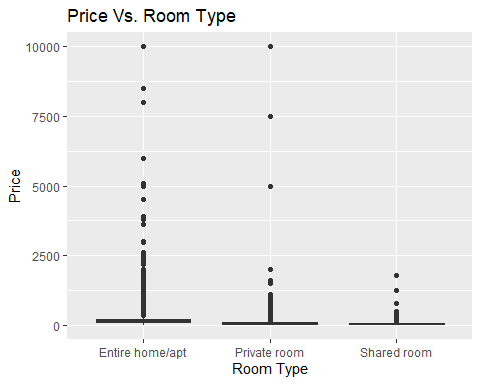
 # from the above graph you can easily see the ‘Manhattan’ have highest prices across all the places

#price Vs number of reviews Vs neighborhood\_group   
   
 ggplot(data = my\_data, aes(x = price, y = number\_of\_reviews, color = neighbourhood\_group)) +  
 geom\_point(alpha = 0.5) +  
 scale\_color\_manual(values = c("#1b9e77", "#d95f02", "#7570b3", "#e7298a", "#66a61e")) +  
 labs(x = "Price", y = "Number of Reviews", color = "Neighborhood Group",  
 title = "R celationship between Price, Number of Reviews, and Neighborhood Group")



#from this its is clear that Queens and Manhattan have highest number of reviews so its conclude that the more expensive the hotel the reviews are also increases

# Create a box plot of Price Vs room\_type  
 ggplot(data = my\_data, aes(x = room\_type, y = price)) +  
 geom\_boxplot() +  
 labs(title = "Price Vs. Room Type", x = "Room Type", y = "Price")



#so by the above Analysis we conclude that\_\_\_\_\_\_\_\_\_.

#(1)= Price is a significant factor when booking a room on Airbnb. The majority of reviews come from the price range of 0-2500, and as the price increases, the number of reviews decreases. #(2)= The most expensive place to stay in NYC is Manhattan, followed by Brooklyn and Queens. The least expensive is the Bronx and Staten Island. #(3)= There are variables that are positively correlated with each other, such as host\_id and calculated\_host\_listings\_count. Additionally, there are variables that are negatively correlated with each other, such as number\_of\_reviews and availability\_365. #(4)= Neighborhood is also an essential factor when booking a room on Airbnb. Some neighborhoods are more expensive than others, and certain neighborhoods are more popular than others. #(5)= Based on the scatter plots, the price range of 0-2500 is the most common range that most people book a room, and minimum nights between 0-400 are the most popular.

#we can use these conclusions and recommendations to gain insights into the Airbnb market in NYC and make data-driven decisions on pricing, marketing, and location.