NYC Airbnb Data Assignment

Data Mine'R's

8/26/2020

Contents

1. Introduction

1.1. What is Airbnb?

Airbnb is an online marketplace since 2008, which connects people who want to rent their homes with people who are looking for accommodations in a particular location. It covers more than 81,000 cities and 191 countries worldwide. The company ,which is based in San Francisco, California, does not own any of the property listings, but it receives commissions from each booking like a broker. The name "Airbnb" comes from "air mattress Bed and Breakfast." The Airbnb logo is called the Bélo, which is a short version for saying 'Belong Anywhere'. Airbnb hosts list many different kinds of properties such as private rooms, apartments, shared rooms, houseboats, entire houses, etc.

1.2. Airbnb Dataset

This dataset describes the listing activity and metrics in NYC for 2019. It includes all the necessary information in order to find out more about hosts, prices, geographical availability, and necessary information to make predictions and draw conclusions for NYC. The explanation of the variables in our data, which consists of 16 columns and 48,895 rows, will be made in the next part. The data used in this assignment is called **New York City Airbnb Open Data** which is downloaded from Kaggle. This public dataset is a part of Airbnb, and the original source can be found on this website.

1.3. Objectives

In this assignment, we will perform an exploratory data analysis(EDA) in order to investigate each of the variables and also come up with a conclusion for the relationship between variables. The main purpose is to identify which variables affect the price mostly. In addition to these, we will explore which neighborhood groups and room types are the most popular ones among the guests, and which hosts are the most preferred ones. The processes during the assignment can be listed as below:

1. Data Preprocessing
2. Data Manipulation
3. Data Visualization
4. Interactive Shiny App

2. Data Explanation

2.1. Used Libraries

We have used several packages during the analysis of the historical data of Airbnb in NYC in order to make data manipulation and visualization. The list of packages used in this assignment can be seen below:

1. tidyverse
2. lubridate
3. tinytex
4. wordcloud
5. shiny
6. knitr
7. data.table
8. tm
9. SnowballC
10. corpus

```r
pti <- c("tidyverse", "lubridate", "tinytex", "wordcloud", "shiny", "knitr", "data.table", "tm", "Snowba
pti <- pti[!(pti %in% installed.packages())]
if(length(pti)>0){
    install.packages(pti)
}

library(tidyverse)
library(lubridate)
library(tinytex)
library(wordcloud)
library(shiny)
library(knitr)
library(data.table)
library(tm)
library(SnowballC)
library(corpus)
```

2.2. Data

Import Data After the importing data, to investigate variables in the data frame,i.e., *airbnb* data set, we use `glimpse()` function.

```r
file <- if(file.exists("AB_NYC_2019.csv")) {
  "AB_NYC_2019.csv"
} else {
  url('https://raw.githubusercontent.com/pjournal/boun01g-data-mine-r-s/gh-pages/Assignment/AB_NYC_2019
}
airbnb = read_csv(file)
airbnb$last_review<-as.POSIXct(airbnb$last_review,format="%Y-%m-%d")
airbnb %>% glimpse()
```

```
## Rows: 48,895
## Columns: 16
## $ id                             <dbl> 2539, 2595, 3647, 3831, 5022, 5099, ...
## $ name                           <chr> "Clean & quiet apt home by the park"...
## $ host_id                        <dbl> 2787, 2845, 4632, 4869, 7192, 7322, ...
## $ host_name                      <chr> "John", "Jennifer", "Elisabeth", "Li...
## $ neighbourhood_group            <chr> "Brooklyn", "Manhattan", "Manhattan"...
## $ neighbourhood                  <chr> "Kensington", "Midtown", "Harlem", "...
## $ latitude                       <dbl> 40.64749, 40.75362, 40.80902, 40.685...
## $ longitude                      <dbl> -73.97237, -73.98377, -73.94190, -73...
## $ room_type                      <chr> "Private room", "Entire home/apt", "...
## $ price                          <dbl> 149, 225, 150, 89, 80, 200, 60, 79, ...
## $ minimum_nights                 <dbl> 1, 1, 3, 1, 10, 3, 45, 2, 2, 1, 5, 2...
## $ number_of_reviews              <dbl> 9, 45, 0, 270, 9, 74, 49, 430, 118, ...
## $ last_review                    <dttm> 2018-10-19 03:00:00, 2019-05-21 03:...
## $ reviews_per_month              <dbl> 0.21, 0.38, NA, 4.64, 0.10, 0.59, 0....
## $ calculated_host_listings_count <dbl> 6, 2, 1, 1, 1, 1, 1, 1, 1, 4, 1, 1, ...
## $ availability_365               <dbl> 365, 355, 365, 194, 0, 129, 0, 220, ...
```

The `glimpse()` is a function of the `dplyr()`. If you do not use the dplyr() package, you can use `str()` function in the base R as an alternative. These two functions give the same results.

```r
airbnb %>% str()
```

```
## tibble [48,895 x 16] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
##  $ id                            : num [1:48895] 2539 2595 3647 3831 5022 ...
##  $ name                          : chr [1:48895] "Clean & quiet apt home by the park" "Skylit Midtown
##  $ host_id                       : num [1:48895] 2787 2845 4632 4869 7192 ...
##  $ host_name                     : chr [1:48895] "John" "Jennifer" "Elisabeth" "LisaRoxanne" ...
##  $ neighbourhood_group           : chr [1:48895] "Brooklyn" "Manhattan" "Manhattan" "Brooklyn" ...
##  $ neighbourhood                 : chr [1:48895] "Kensington" "Midtown" "Harlem" "Clinton Hill" ...
##  $ latitude                      : num [1:48895] 40.6 40.8 40.8 40.7 40.8 ...
##  $ longitude                     : num [1:48895] -74 -74 -73.9 -74 -73.9 ...
##  $ room_type                     : chr [1:48895] "Private room" "Entire home/apt" "Private room" "En
##  $ price                         : num [1:48895] 149 225 150 89 80 200 60 79 79 150 ...
##  $ minimum_nights                : num [1:48895] 1 1 3 1 10 3 45 2 2 1 ...
##  $ number_of_reviews             : num [1:48895] 9 45 0 270 9 74 49 430 118 160 ...
##  $ last_review                   : POSIXct[1:48895], format: "2018-10-19 03:00:00" "2019-05-21 03:00
##  $ reviews_per_month             : num [1:48895] 0.21 0.38 NA 4.64 0.1 0.59 0.4 3.47 0.99 1.33 ...
##  $ calculated_host_listings_count: num [1:48895] 6 2 1 1 1 1 1 1 1 4 ...
##  $ availability_365              : num [1:48895] 365 355 365 194 0 129 0 220 0 188 ...
##  - attr(*, "spec")=
##   .. cols(
##   ..   id = col_double(),
##   ..   name = col_character(),
##   ..   host_id = col_double(),
##   ..   host_name = col_character(),
##   ..   neighbourhood_group = col_character(),
##   ..   neighbourhood = col_character(),
##   ..   latitude = col_double(),
##   ..   longitude = col_double(),
##   ..   room_type = col_character(),
##   ..   price = col_double(),
##   ..   minimum_nights = col_double(),
##   ..   number_of_reviews = col_double(),
##   ..   last_review = col_date(format = ""),
##   ..   reviews_per_month = col_double(),
##   ..   calculated_host_listings_count = col_double(),
##   ..   availability_365 = col_double()
##   .. )
```

Variables This dataset contains 16 features/variables about Airbnb listings within New York City. Below are the features with their descriptions:

1. `id`: Listing ID (numeric variable)
2. `name`: Listing Title (categorical variable)
3. `host_id`: ID of Host (numeric variable)
4. `host_name`: Name of Host (categorical Variable)
5. `neighbourhood_group`: Neighbourhood group that contains listing (categorical variable)
6. `neighbourhood`: Neighbourhood group that contains listing (categorical variable)
7. `latitude`: Latitude of listing (numeric variable)
8. `longitude`: Longitude of listing (numeric variable)
9. `room_type`: Type of the offered property (categorical variable)
10. `price`: Price per night in USD (numeric variable)

11. `minimum_nights`: Minimum number of nights required to book listing (numeric variable)
12. `number_of_reviews`: Total number of reviews that listing has (numeric variable)
13. `last_review`: Last rent date of the listing (date variable)
14. `reviews_per_month`: Total number of reviews divided by the number of months that the listing is active (numeric variable)
15. `calculated_host_listings_count`: Amount of listing per host (numeric variable)
16. `availability_365`: Number of days per year the listing is active (numeric variable)

2.3. Dublicate and Missing Data

Our data set has almost 49.000 rows. Therefore it may include duplicate and/or missing values. To check them, we run the following codes.

```
NAValues <-
  airbnb %>% select(everything()) %>% summarise_all(funs(sum(is.na(.))))
```

There are 10052 values missing in the dataset and all of them are in the `reviews_per_month` column.

```
sum(duplicated(airbnb))
```

```
## [1] 0
```

There is no duplicated row in this dataset.

2.4. Summary of Data

The summary of the data set can be seen below.

```
airbnb %>% summary(.)
```

```
##        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
##  Min.   :2011-03-28 02:00:00   Min.   : 0.010    Min.   :  1.000
##  1st Qu.:2018-07-08 03:00:00   1st Qu.: 0.190    1st Qu.:  1.000
##  Median :2019-05-19 03:00:00   Median : 0.720    Median :  1.000
##  Mean   :2018-10-04 04:47:23   Mean   : 1.373    Mean   :  7.144
##  3rd Qu.:2019-06-23 03:00:00   3rd Qu.: 2.020    3rd Qu.:  2.000
##  Max.   :2019-07-08 03:00:00   Max.   :58.500    Max.   :327.000
##  NA's   :10052                 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
##
```

Before starting our analysis, we also want to check the outlier points in this dataset and we take the quantile 1 and 3 as references.

```r
qtl1 = quantile(airbnb$price, 0.25)
qtl3 = quantile(airbnb$price, 0.75)
iqr = qtl3 - qtl1

lower = qtl1 - iqr * 1.5
upper = qtl3 + iqr * 1.5

lower
```

```
## 25%
## -90
```

```r
upper
```

```
## 75%
## 334
```

```r
airbnb %>%
  filter(price < lower | price > upper) %>%
  top_n(10, price) %>%
  select(neighbourhood_group, neighbourhood, price) %>%
  arrange(desc(price)) %>%
  kable(col.names = c("Neighbourhood Group", "Neighbourhood", "Price"))
```

Neighbourhood Group	Neighbourhood	Price
Queens	Astoria	10000
Brooklyn	Greenpoint	10000
Manhattan	Upper West Side	10000

Neighbourhood Group	Neighbourhood	Price
Manhattan	East Harlem	9999
Manhattan	Lower East Side	9999
Manhattan	Lower East Side	9999
Manhattan	Tribeca	8500
Brooklyn	Clinton Hill	8000
Manhattan	Upper East Side	7703
Manhattan	Battery Park City	7500
Brooklyn	East Flatbush	7500

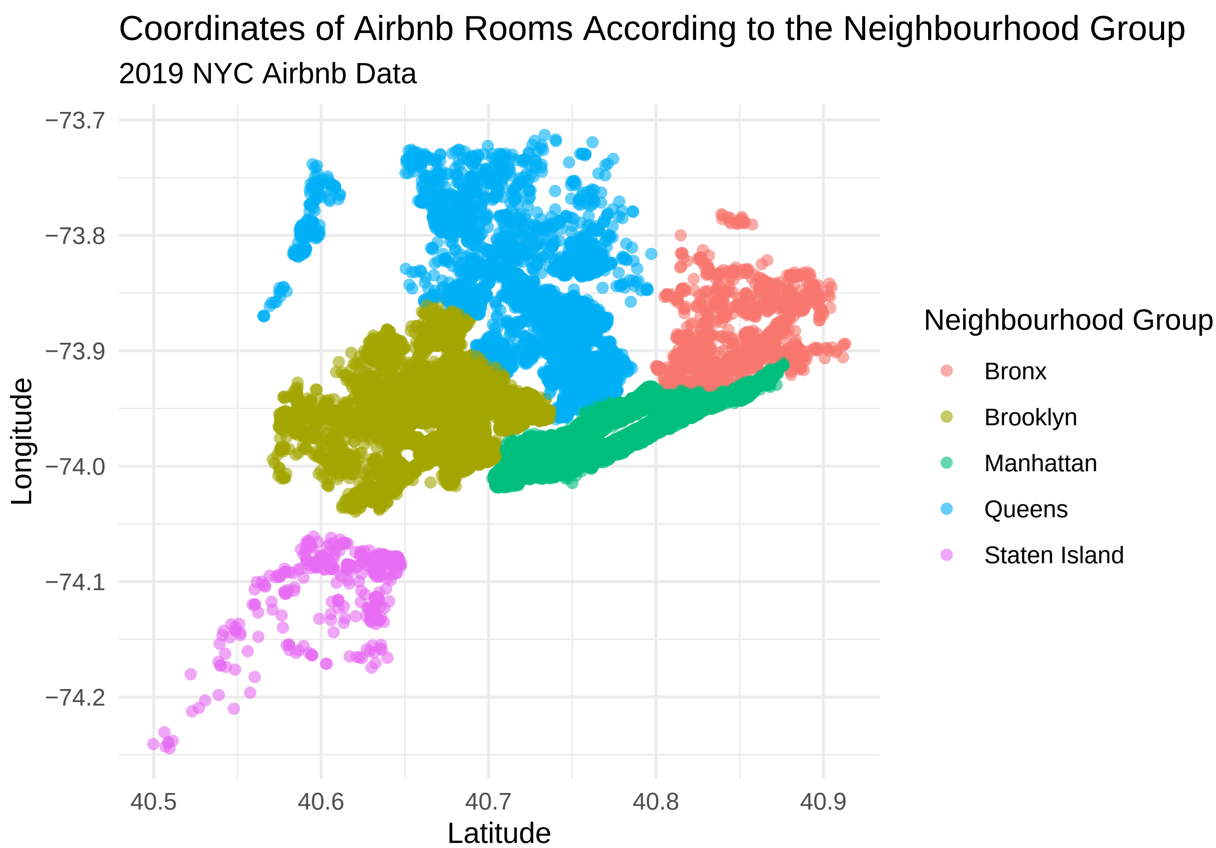
When we analyze the lower and upper bound of the non-outliers data, the lower bound is obtained as minus 90. In our data set, as we consider the price of the airbnb room, there is no negative price. For this reason, we only consider the upper bound. The upper bound address the 334. This means that, if the price value is greater than 334, it becomes an outlier value.In this data set, there are 2972 outliers and the top ten with the highest price is listed as above.

3. Exploratory Data Analysis

3.1 Coordinates of Neighborhood Groups

In order to see the location of airbnb rooms, we use coordinates (latitude and longitude) and color the neighborhood groups. Moreover, to see the density of the rooms in each neighborhood group, we use feature of the `geom_point()`, which is `alpha`.

```r
ggplot(airbnb, aes(latitude, longitude, color = neighbourhood_group)) +
  geom_point(alpha = 0.6) +
  theme_minimal() +
  labs(title = "Coordinates of Airbnb Rooms According to the Neighbourhood Group",
       subtitle = "2019 NYC Airbnb Data",
       x = "Latitude",
       y = "Longitude",
       color = "Neighbourhood Group")
```



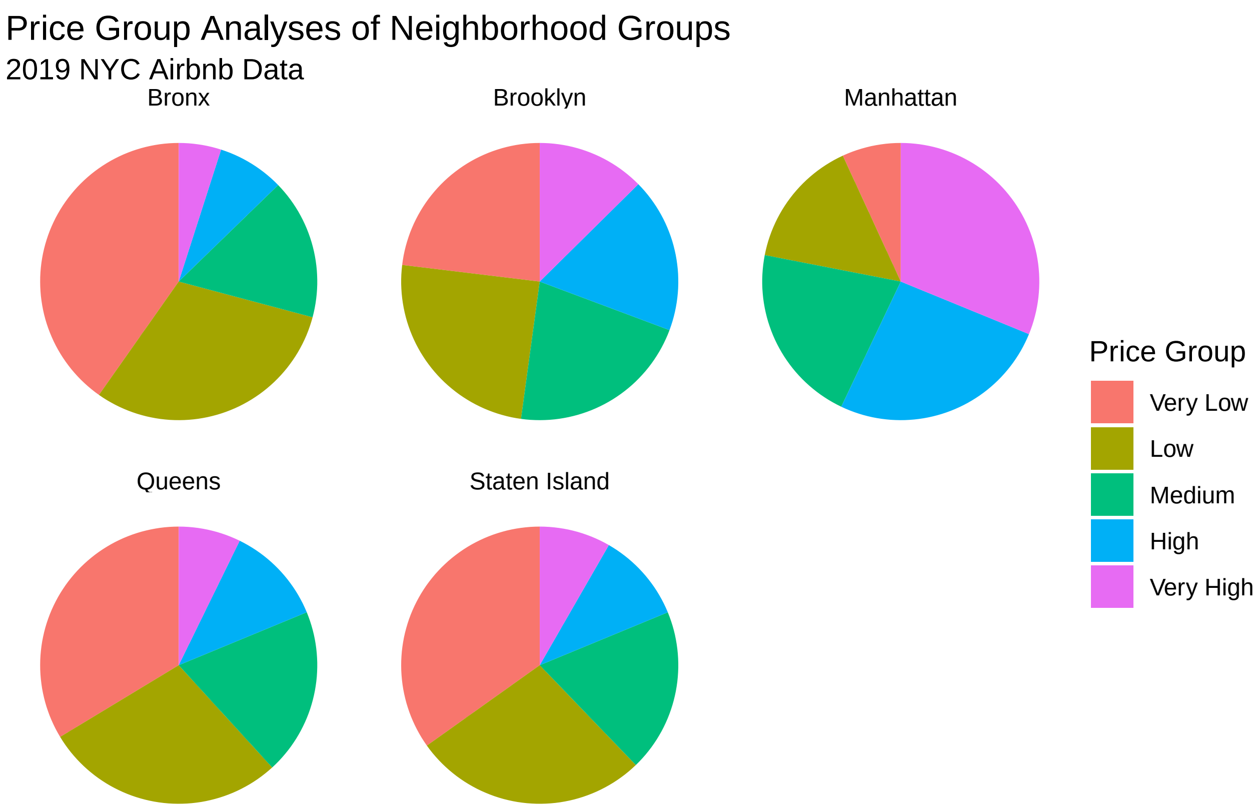
Bronx and Staten Island have less room than the others. The room densities of Brooklyn and Manhattan are distributed balanced in their regions.

3.2 Price Group Analyses of Neighborhood Groups

By using quantile function, we divide price interval into five. Then, we define values in this intervals as very low, low, medium, high, very high. Then by using this categorical value, we prepare pie chart for each neighborhood group.

```
quant = quantile(airbnb$price, seq(0, 1, 0.2))
#quant

airbnb_price_group = airbnb %>%
  mutate(price_group = case_when(
    price < quant[2] ~ "Very Low",
    price < quant[3] ~ "Low",
    price < quant[4] ~ "Medium",
    price < quant[5] ~ "High",
    TRUE ~ "Very High"
  )) %>%
  mutate(price_group = factor(price_group, levels = c("Very Low", "Low", "Medium", "High", "Very High")

airbnb_price_group %>%
  group_by(neighbourhood_group, price_group) %>%
  summarize(counter = n())  %>%
```

```
  ggplot(., aes(x = '', y = counter, fill = price_group)) +
  geom_bar(width = 1, stat = "identity", position = "fill") +
  coord_polar("y") +
  theme_void() +
  theme(plot.title = element_text(vjust = 0.5)) +
  facet_wrap(~neighbourhood_group) +
  labs(title = "Price Group Analyses of Neighborhood Groups",
       subtitle = "2019 NYC Airbnb Data",
       fill = "Price Group")
```



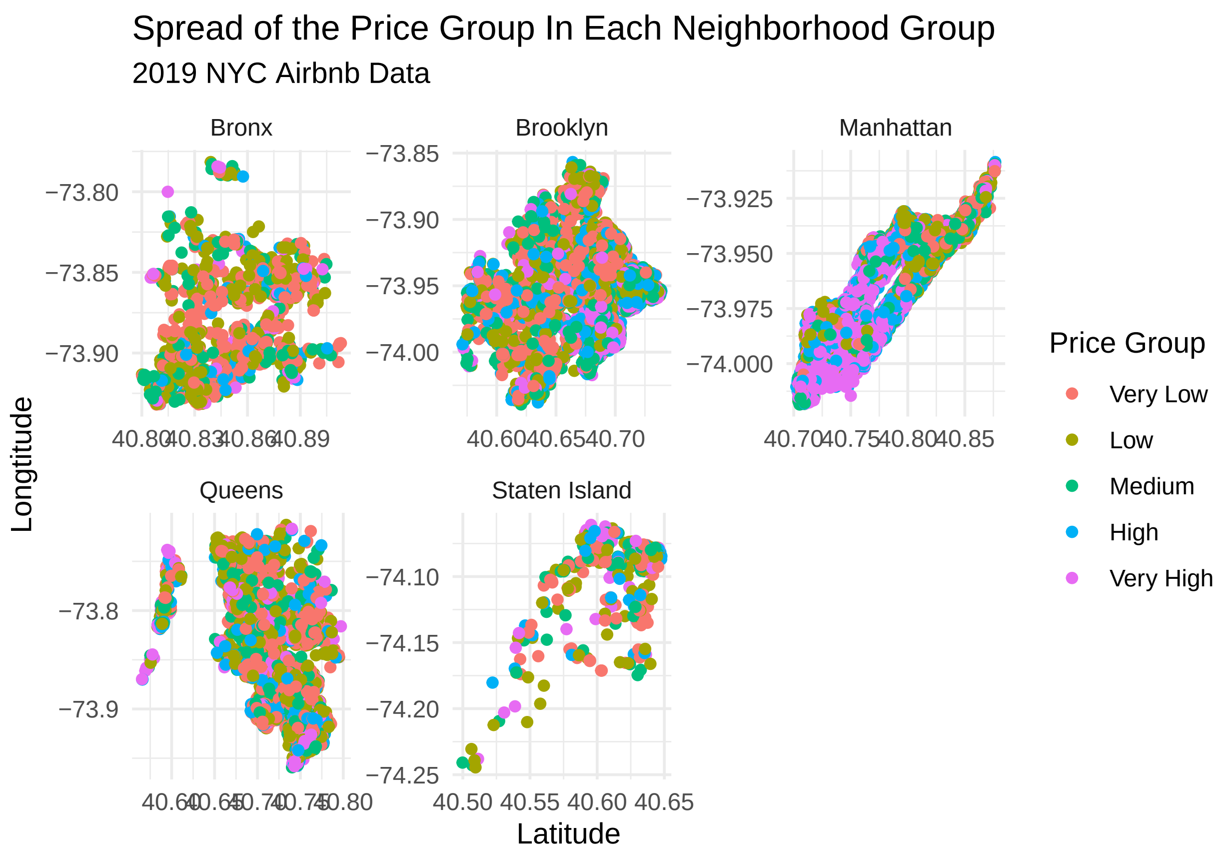
We summarize the results as follow:

- The most of the rooms in Bronx, Queens and Staten Island has very low price.
- The rooms with very low, low and medium prices in the Brooklyn are almost distributed equal percentage.
- The very high price group in Manhattan has higher percentage than the other price groups.

In the previous analysis, we try to define the percentage of the price group in each neighborhood group. To illustrate the price group change by location in each neighborhood group, the following plots are conducted.

```
airbnb_price_group %>%
  ggplot(., aes(latitude, longitude, color = price_group)) +
  geom_point() +
  theme_minimal() +
```

```r
  facet_wrap(~neighbourhood_group, scales = "free") +
  labs(title = "Spread of the Price Group In Each Neighborhood Group",
       subtitle = "2019 NYC Airbnb Data",
       x = "Latitude",
       y = "Longtitude",
       color = "Price Group")
```



Very high price group in Manhattan concentrates in a particular area, although there are homogeneous spread of price groups in Bronx, Brooklyn, Queens, and Staten Island.

3.3 Minimum, Maximum and Average Price

Before the detailed explanatory data analysis, we obtain average, minimum and maximum price for each neighborhood group. Moreover, we can add the average availability and average number of reviews. These values give general information about the airbnb rooms.

```r
airbnb %>%
  group_by(neighbourhood_group) %>%
  summarise(min_price = min(price),
            mean_price = round(mean(price), digits = 2),
            max_price = max(price),
            average_availability = round(mean(availability_365), digits = 2),
            average_review = round(mean(number_of_reviews), digits = 2)) %>%
```

```r
  select(neighbourhood_group, min_price, mean_price, max_price, average_availability, average_review ) %
  arrange(desc(mean_price)) %>%
  kable(col.names = c("Neighborhood Group", "Min Price", "Mean Price", "Max Price", "Average Availabili
```

Neighborhood Group	Min Price	Mean Price	Max Price	Average Availability	Average Review
Manhattan	0	196.88	10000	111.98	20.99
Brooklyn	0	124.38	10000	100.23	24.20
Staten Island	13	114.81	5000	199.68	30.94
Queens	10	99.52	10000	144.45	27.70
Bronx	0	87.50	2500	165.76	26.00

After the neighborhood group analysis, same preparation can be made by using room types. This table presents a more comprehensive analysis.

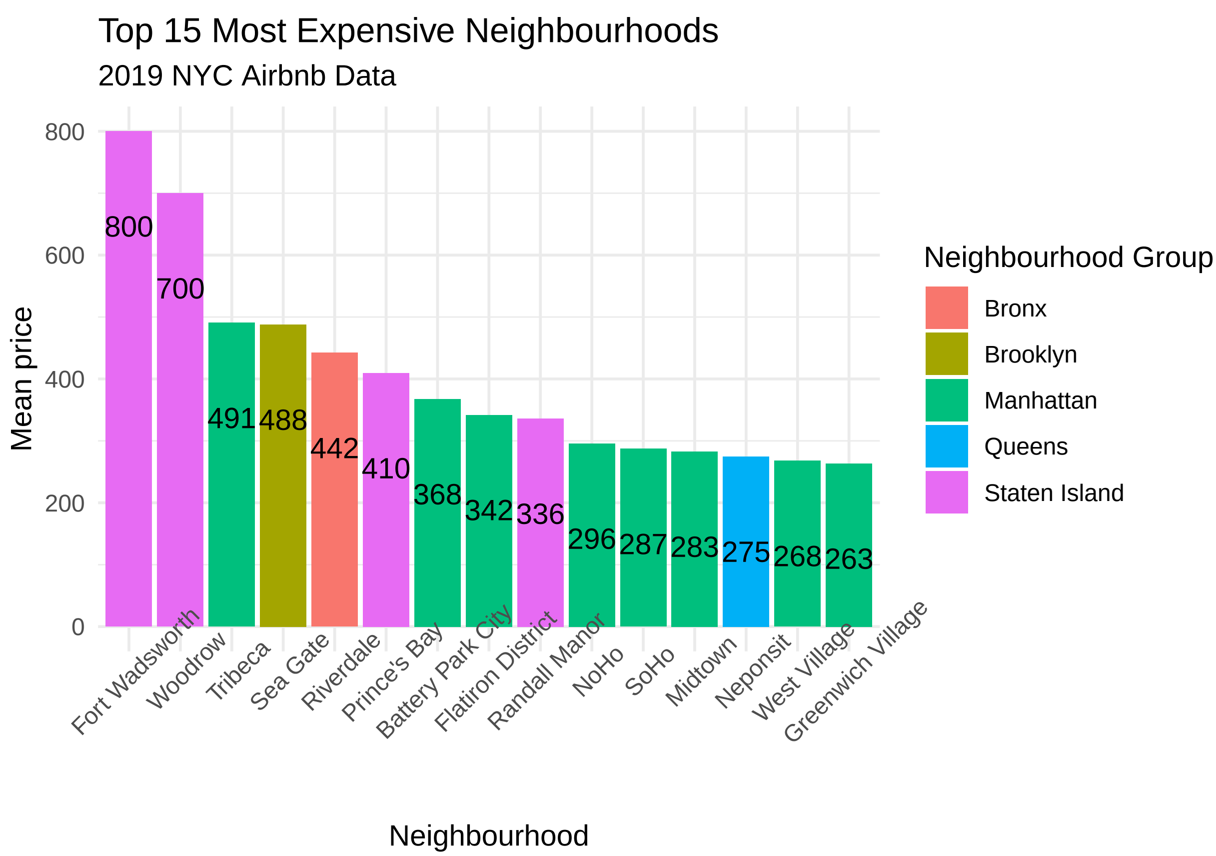
```r
airbnb %>%
  group_by(neighbourhood_group, room_type) %>%
  summarise(min_price = min(price),
            mean_price = round(mean(price), digits = 2),
            max_price = max(price),
            average_availability = round(mean(availability_365), digits = 2),
            average_review = round(mean(number_of_reviews), digits = 2)) %>%

  select(neighbourhood_group,room_type, min_price, mean_price, max_price, average_availability, average_
  kable(col.names = c("Neighborhood Group", "Room Type", "Min Price", "Mean Price", "Max Price", "Averag
```

Neighborhood Group	Room Type	Min Price	Mean Price	Max Price	Average Availability	Average Review
Bronx	Entire home/apt	28	127.51	1000	158.00	30.68
Bronx	Private room	0	66.79	2500	171.33	25.02
Bronx	Shared room	20	59.80	800	154.22	7.20
Brooklyn	Entire home/apt	0	178.33	10000	97.21	27.95
Brooklyn	Private room	0	76.50	7500	99.92	21.09
Brooklyn	Shared room	0	50.53	725	178.01	14.03
Manhattan	Entire home/apt	0	249.24	10000	117.14	17.82
Manhattan	Private room	10	116.78	9999	101.85	26.20
Manhattan	Shared room	10	88.98	1000	138.57	21.40
Queens	Entire home/apt	10	147.05	2600	132.27	28.93
Queens	Private room	10	71.76	10000	149.22	27.75
Queens	Shared room	11	69.02	1800	192.19	13.86
Staten Island	Entire home/apt	48	173.85	5000	178.07	33.28
Staten Island	Private room	20	62.29	300	226.36	30.16
Staten Island	Shared room	13	57.44	150	64.78	1.56

3.4 The Most and Least Expensive Neighborhoods

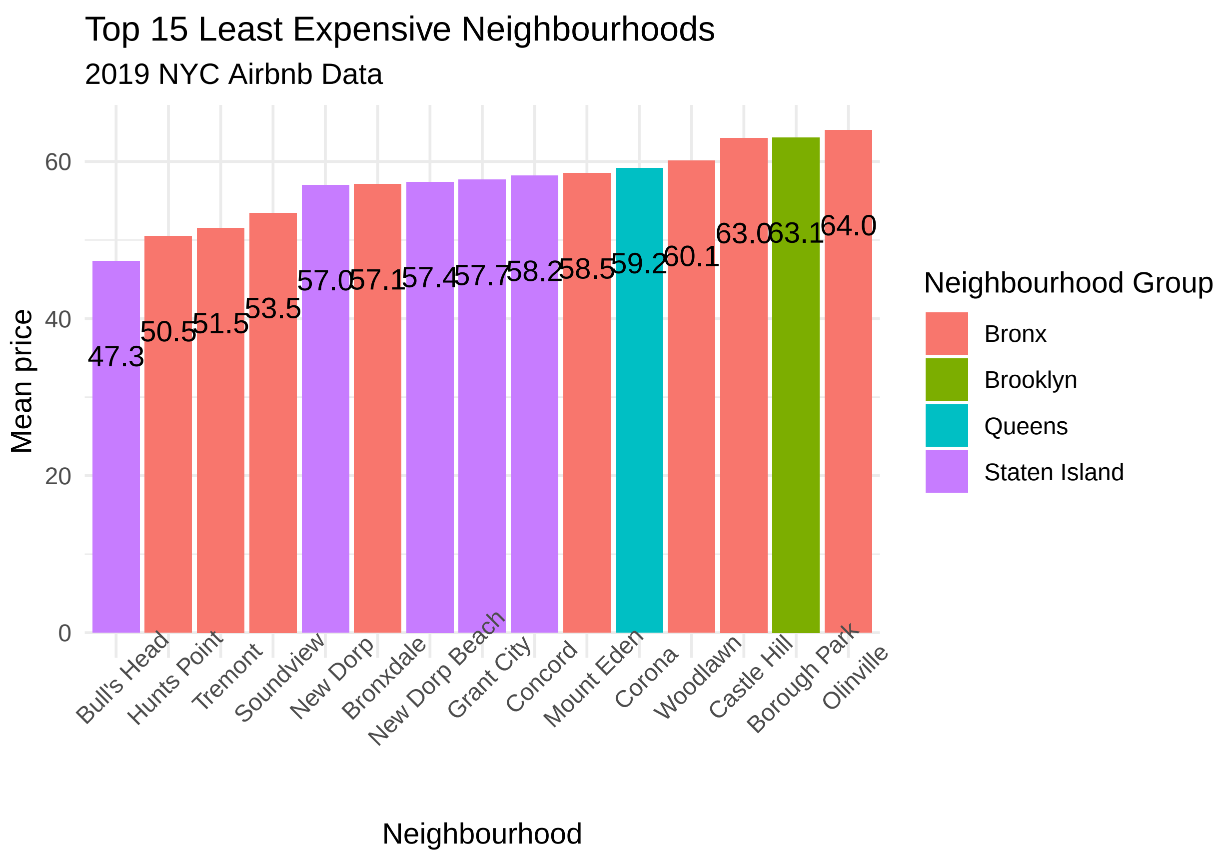
We can obtain the most and least expensive neighborhoods according to the mean price. To provide more understandable results, we use bar chart that illustrates the neighborhoods and its neighborhood groups.

```
airbnb %>%
  group_by(neighbourhood_group,neighbourhood)%>%
  summarise(mean_price = mean(price))%>%
  arrange(desc(mean_price))%>%
  head(15)%>%
  ggplot(., aes(x = reorder(neighbourhood, -mean_price) , y = mean_price, fill = neighbourhood_group))
  geom_col() +
  theme_minimal() +
  geom_text(aes(label = format(mean_price,digits=3)), size=4, position = position_dodge(0.9),vjust = 5)
  theme(axis.text.x = element_text(angle = 45), legend.position = "right") +
  labs(title = "Top 15 Most Expensive Neighbourhoods",
       subtitle ="2019 NYC Airbnb Data",
       x = "Neighbourhood",
       y = "Mean price",
       fill = "Neighbourhood Group")
```



The most expensive neighborhood is Fort Wadsworth with the average price $800. The other inference obtained from the bar chart is that the most expensive rooms are located in Manhattan and Staten Island. Same analysis can be made for the least expensive neighborhoods.

```
airbnb %>%
  group_by(neighbourhood_group,neighbourhood)%>%
  summarise(mean_price = mean(price))%>%
  arrange(mean_price) %>%
  head(15)%>%
  ggplot(., aes(x = reorder(neighbourhood, mean_price) , y = mean_price, fill = neighbourhood_group)) +
  geom_col() +
  theme_minimal() +
  geom_text(aes(label = format(mean_price,digits=3)), size=4, position = position_dodge(0.9),vjust = 5)
  theme(axis.text.x = element_text(angle = 45), legend.position = "right") +
  labs(title = "Top 15 Least Expensive Neighbourhoods",
       subtitle ="2019 NYC Airbnb Data",
       x = "Neighbourhood",
       y = "Mean price",
       fill = "Neighbourhood Group")
```



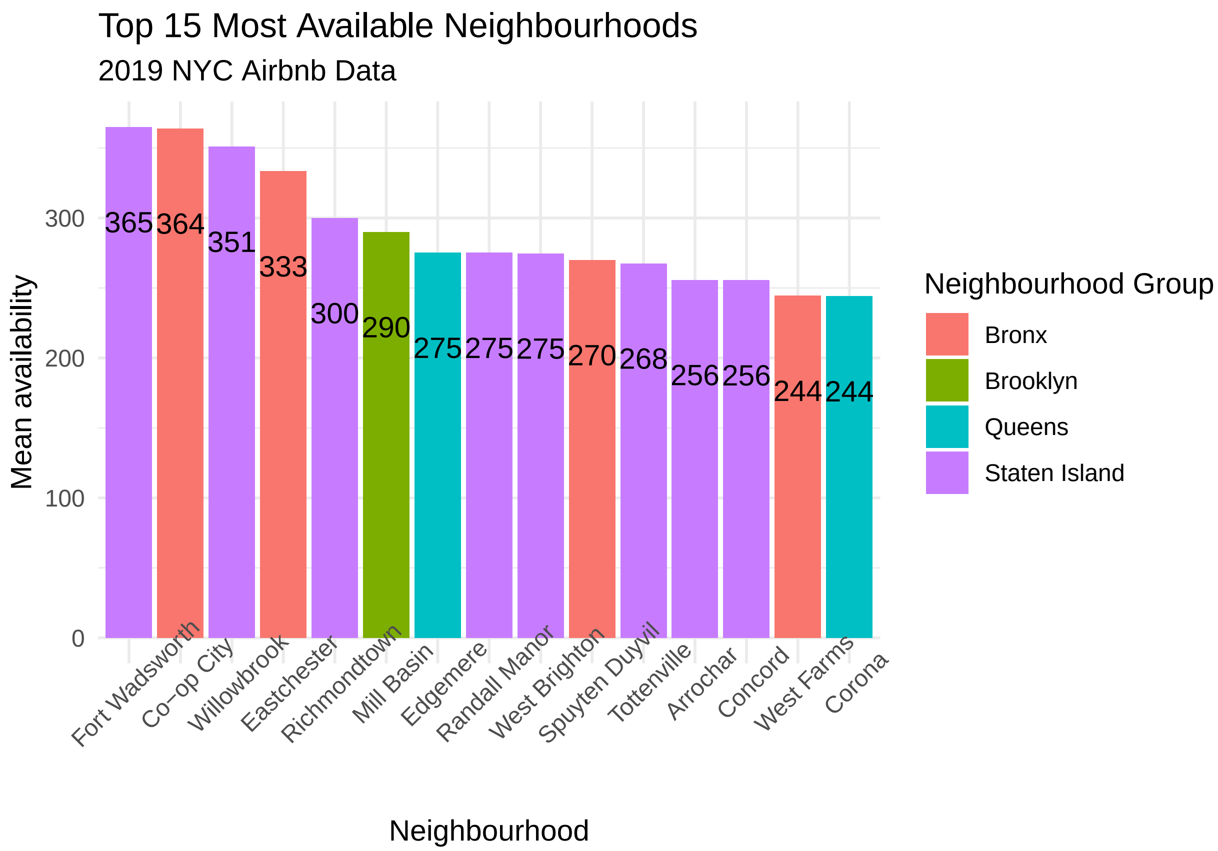
The least expensive neighborhood is Bull's Head with the average price $47.3. The other inference obtained from the bar chart is that the least expensive rooms are located in Bronx and Staten Island. Moreover, there is no room belongs to Manhattan in the least expensive neighborhoods.

These results show that, the price of rooms in Staten Island has a wide range.

3.5 The Most and Least Available Neighborhoods

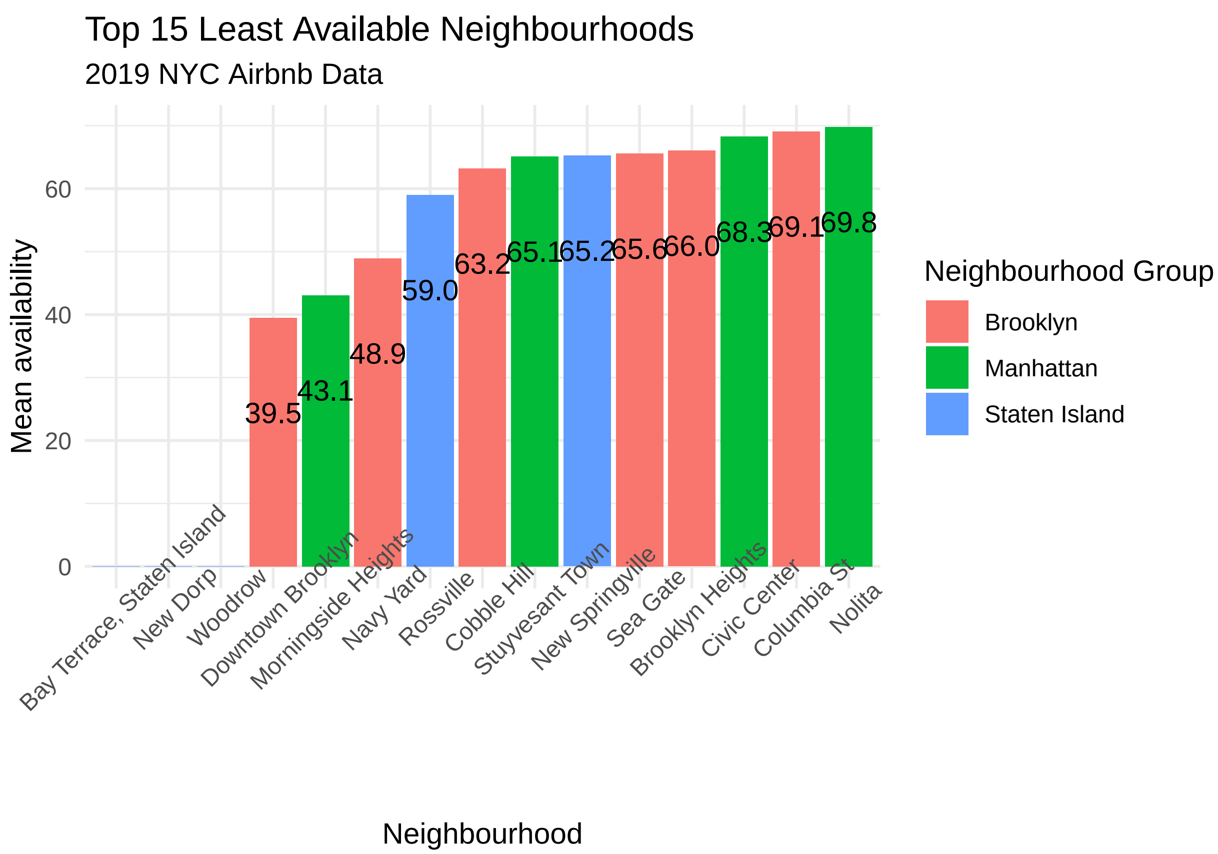
The neighborhoods are also investigated by using average room availability. In this part of the report, we give the most and least available neighborhoods according to the neighborhood groups.

```
airbnb %>%
  group_by(neighbourhood, neighbourhood_group)%>%
  summarise(mean_availability = mean(availability_365))%>%
  arrange(desc(mean_availability))%>%
  head(15)%>%
  ggplot(., aes(x = reorder(neighbourhood,-mean_availability) , y = mean_availability, fill = neighbour
  geom_col() +
  theme_minimal() +
  geom_text(aes(label = format(mean_availability, digits = 3)), size=4, position = position_dodge(0.9),
  theme(axis.text.x = element_text(angle = 45), legend.position = "right") +
  labs(title = "Top 15 Most Available Neighbourhoods",
       subtitle ="2019 NYC Airbnb Data",
       x = "Neighbourhood",
       y = "Mean availability",
       fill = "Neighbourhood Group")
```



By using average availability of the rooms, the graph shows that Staten Island is the most available neighborhood group in the top 15. Manhattan, on the other hand, does not have any neighborhood in the top 15.

```r
airbnb %>%
  group_by(neighbourhood, neighbourhood_group)%>%
  summarise(mean_availability = mean(availability_365))%>%
  arrange(mean_availability)%>%
  head(15)%>%
  ggplot(., aes(x = reorder(neighbourhood,mean_availability) , y = mean_availability, fill = neighbourh
  geom_col() +
  theme_minimal() +
  geom_text(aes(label = format(mean_availability, digits = 3)), size=4, position = position_dodge(0.9),
  theme(axis.text.x = element_text(angle = 45), legend.position = "right") +
  labs(title = "Top 15 Least Available Neighbourhoods",
       subtitle ="2019 NYC Airbnb Data",
       x = "Neighbourhood",
       y = "Mean availability",
       fill = "Neighbourhood Group")
```

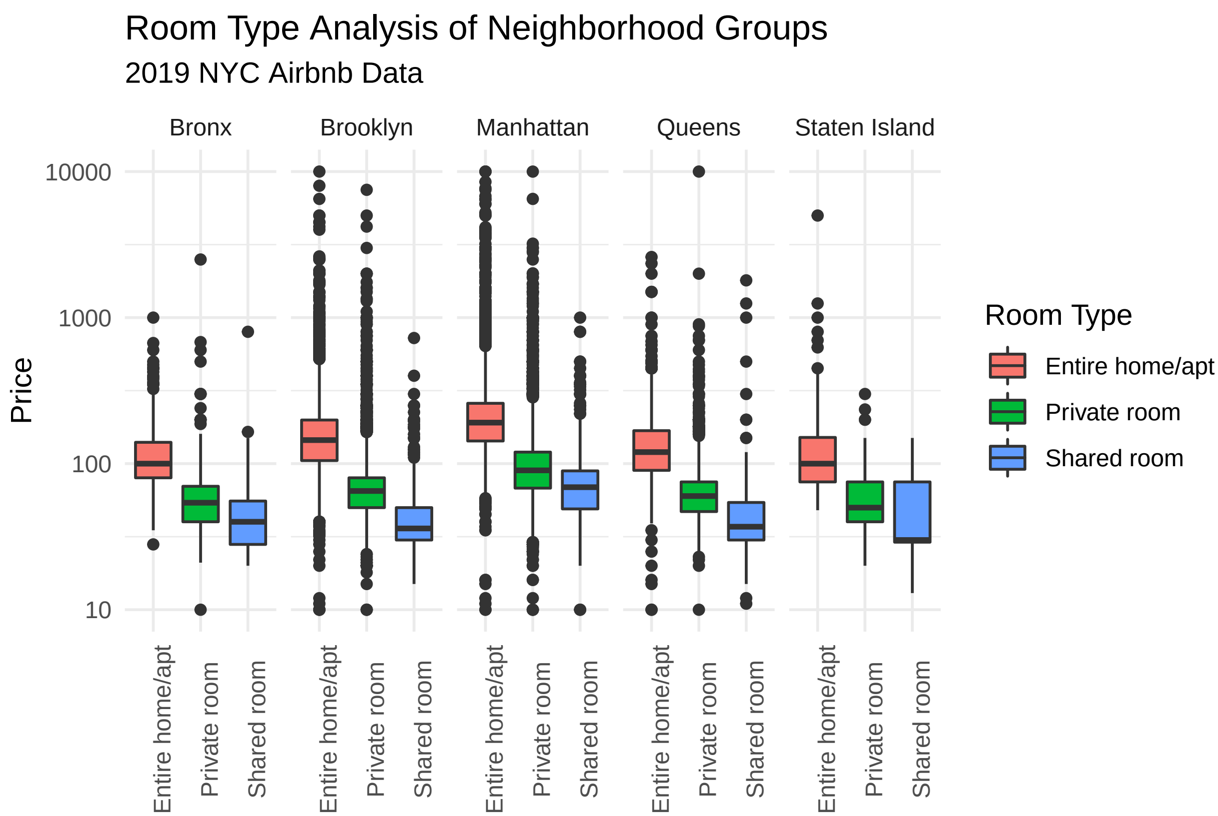


There are many neighborhoods in the data set with zero availability. Bay Terrace (Staten Island) and New Dorp (Staten Island) do not have availability.

3.6 Room Type Analysis of Neighborhood Groups

We use box plot to illustrate the log(price) of the different room types for each neighborhood group.

```
ggplot(airbnb, aes(x = room_type, y = price, fill = room_type)) + scale_y_log10() +
  geom_boxplot() +
  theme_minimal() +
  labs (x="", y= "Price") +
  facet_wrap(~neighbourhood_group) +
  facet_grid(.~ neighbourhood_group) +
  theme(axis.text.x = element_text(angle = 90), legend.position = "right") +
  labs(title = "Room Type Analysis of Neighborhood Groups",
       subtitle = "2019 NYC Airbnb Data",
       fill = "Room Type")
```



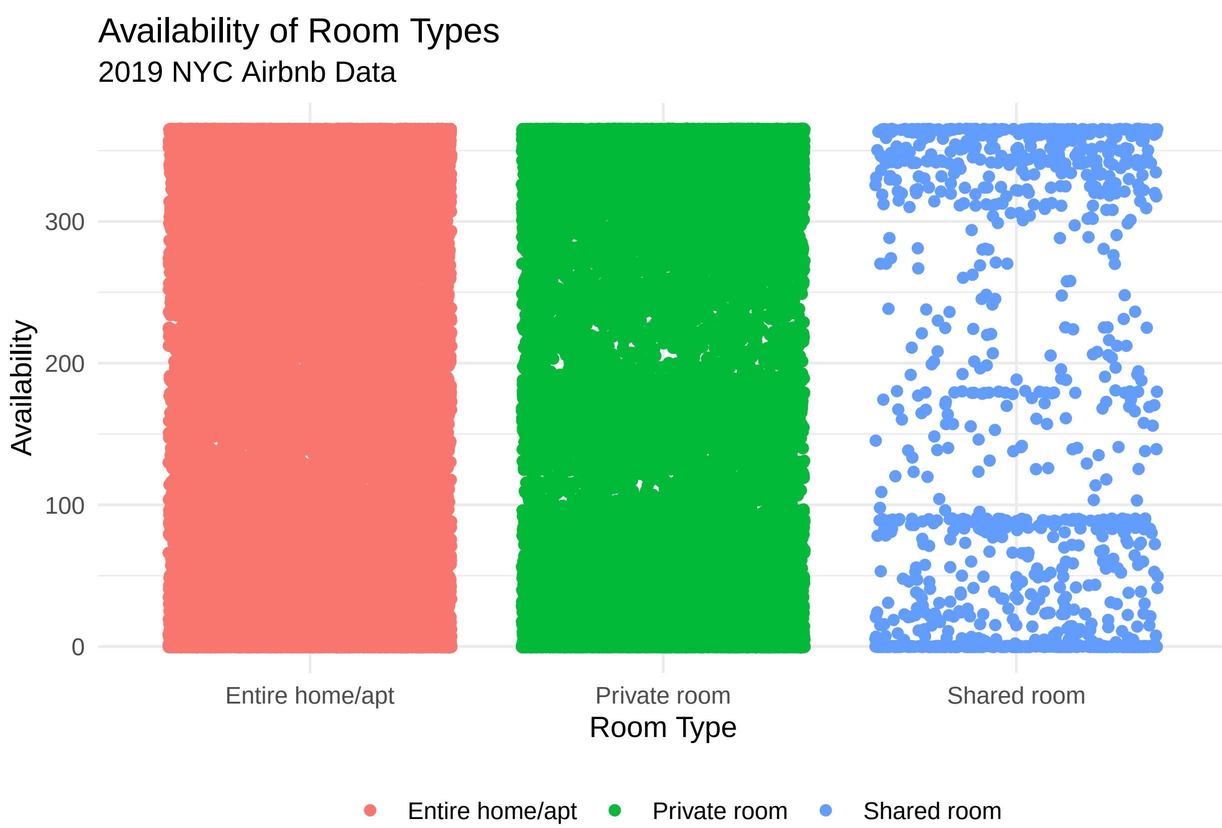
Results show that:

- For each neighborhood group, the order of the rooms according to descending price is entire home, private room, and shared room.
- In each room type, Manhattan has highest average price. However, the price structure is similar among Brooklyn, Manhattan, and Queens.
- The outliers in Brooklyn and Manhattan are more than the others.

3.7 Availability of Room Types According to the Neighborhood Groups

To see the availability of different room types, we use `geom_jitter()` function and also check the density of each room type.

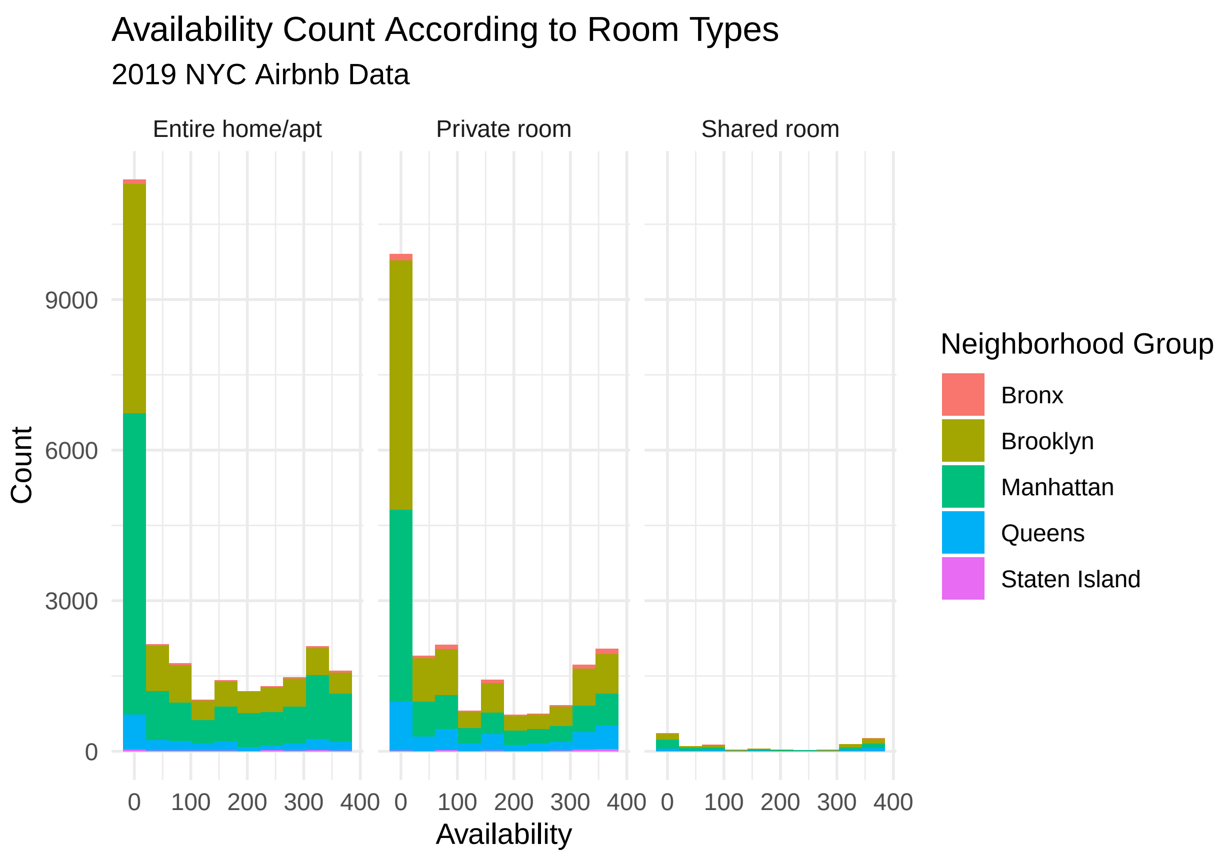
```r
airbnb %>%
  ggplot(., aes(x = room_type, y = availability_365, color = room_type)) +
  geom_jitter() +
  theme_minimal() +
  theme(legend.position="bottom", plot.title = element_text(vjust = 0.5)) +
  labs(title = "Availability of Room Types",
       subtitle = "2019 NYC Airbnb Data",
       x = "Room Type",
       y = "Availability",
       color = " ")
```



Entire home and private room have homogeneous distribution of availability, while the shared room accumulates on the edge of the intervals. To make more analysis, we also plot histogram. In this histogram, we want to analyze the availability of room types according to the neighborhood groups. It can be said that entire home/apt and private room can be reached every day in a year, whereas, shared room is not always accessible.

```r
airbnb %>%
  ggplot(., aes(availability_365, fill = neighbourhood_group)) +
  geom_histogram(bins = 10) +
  facet_wrap(~room_type)+
  theme_minimal() +
  labs(title = "Availability Count According to Room Types",
       subtitle = "2019 NYC Airbnb Data",
       x = "Availability",
```

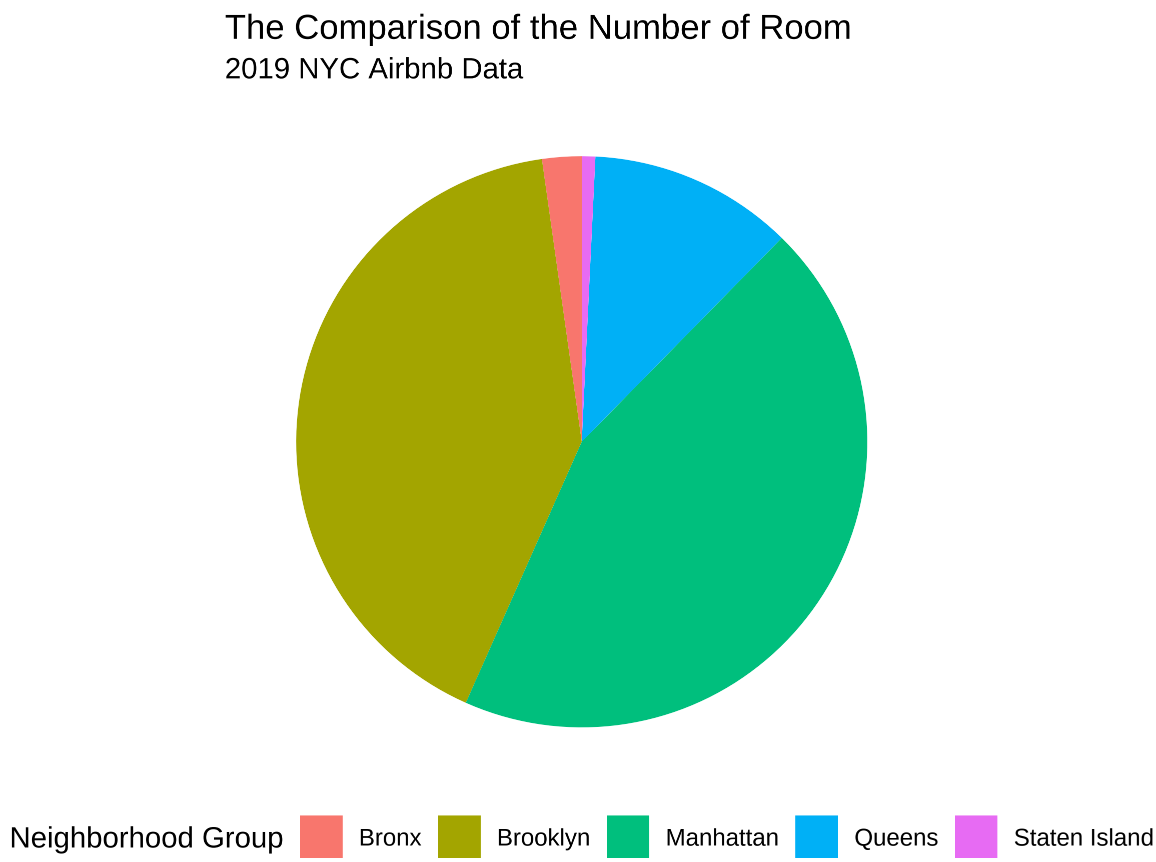
```
      y = "Count",
      fill = "Neighborhood Group")
```



3.8 The Number of Rooms in Each Neighborhood Group

There are almost 50000 rooms in our data set. As we want to find the number of rooms and compare with each other, first we draw a pie chart and then we summarize in the table to provide clear difference.

```
airbnb %>%
  group_by(neighbourhood_group) %>%
  summarise(count = n(), percentage = n()/nrow(airbnb)) %>%
  ggplot(., aes(x = '', y = count, fill = neighbourhood_group)) +
  geom_bar(width = 1, stat = "identity") +
  coord_polar("y") +
  theme_void() +
  #geom_text(aes(label = scales::percent(round(percentage,2))), position = position_stack(vjust = 0.5))
  theme(legend.position="bottom", plot.title = element_text(vjust = 0.5)) +
  labs(title = "The Comparison of the Number of Room",
       subtitle = "2019 NYC Airbnb Data",
       fill = "Neighborhood Group")
```




```
airbnb %>%
  group_by(neighbourhood_group) %>%
  summarise(count = n())%>%
  transmute(neighbourhood_group,count, percentage = round(100*(count/nrow(airbnb)),digits = 2)) %>%
  kable(col.names = c("Neighborhood Group", "Number", "Percentage"))
```

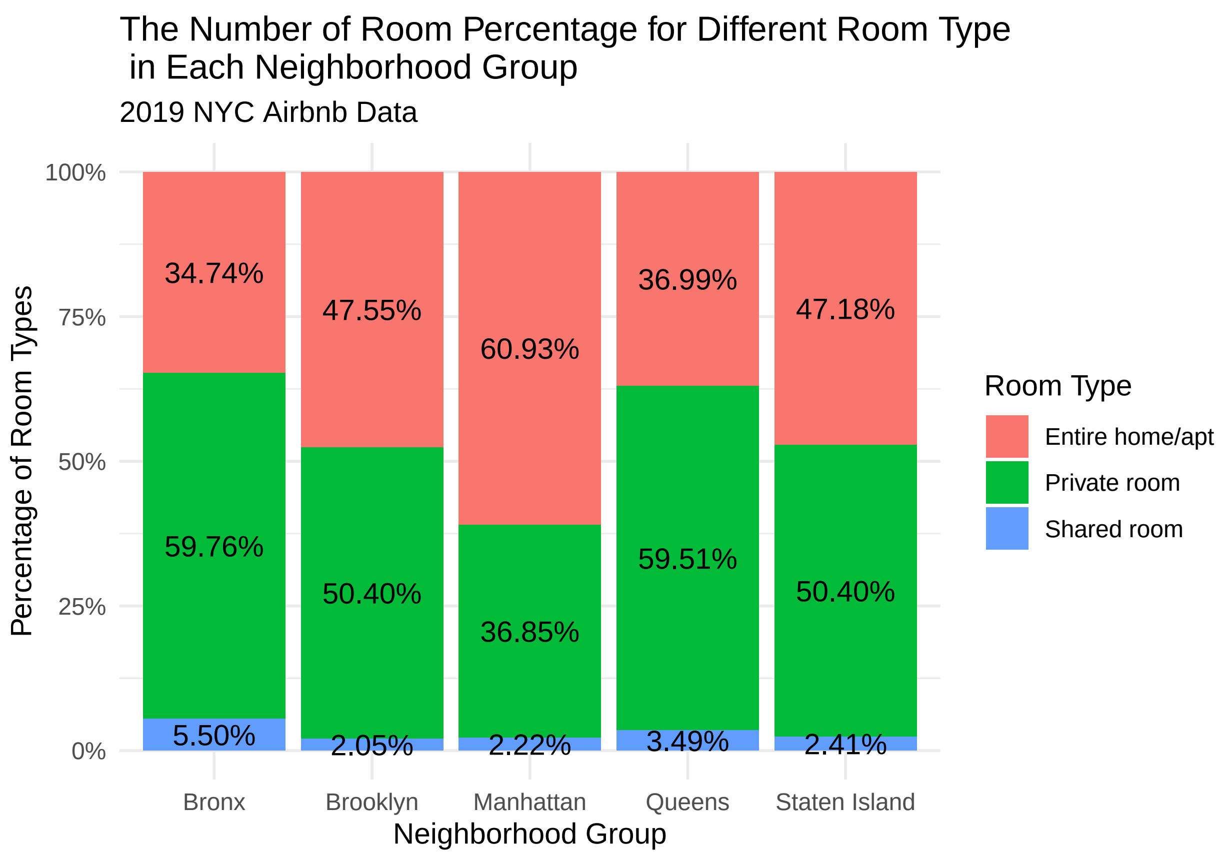
Neighborhood Group	Number	Percentage
Bronx	1091	2.23
Brooklyn	20104	41.12
Manhattan	21661	44.30
Queens	5666	11.59
Staten Island	373	0.76

The results illustrate that the rooms in Manhattan and Brooklyn constitute the huge majority, i.e., the sum of these two percentage is equal to 85.42.

3.9 The Number of Rooms in Each Neighborhood Group By Using Room Type

We analyze the number of room in each neighborhood group in previous graph. We can enlarge this analysis by using room type.

```r
airbnb %>%
  group_by(neighbourhood_group, room_type) %>%
  summarize(room_type_count = n())  %>%
  mutate(room_type_percentage = room_type_count / sum(room_type_count)) %>%
  ggplot(., aes(x = neighbourhood_group, y = room_type_percentage, fill = room_type)) +
  geom_bar(position = "fill",stat = "identity") +
  scale_y_continuous(labels = scales::percent_format()) +
  geom_text(aes(label = scales::percent(round(room_type_percentage, 4))),
            position = position_stack(vjust = .5)) +
  theme_minimal() +
  labs(title = "The Number of Room Percentage for Different Room Type \n in Each Neighborhood Group",
       subtitle = "2019 NYC Airbnb Data",
       x = "Neighborhood Group",
       y = "Percentage of Room Types",
       fill = "Room Type ")
```



Following results can be obtained:

- Private room has the largest percentage for the room type in NYC except Manhattan where the entire home is more preferred.
- In every neighborhood group, shared room type is the least preferable. When we compare the percentages belong to shared room, Bronx is on the top.

3.10 Wordcloud

Like the numerical values, airbnb data includes verbal information such as `name`. By using this information, we can obtain the most used words in name column which describes the room features.

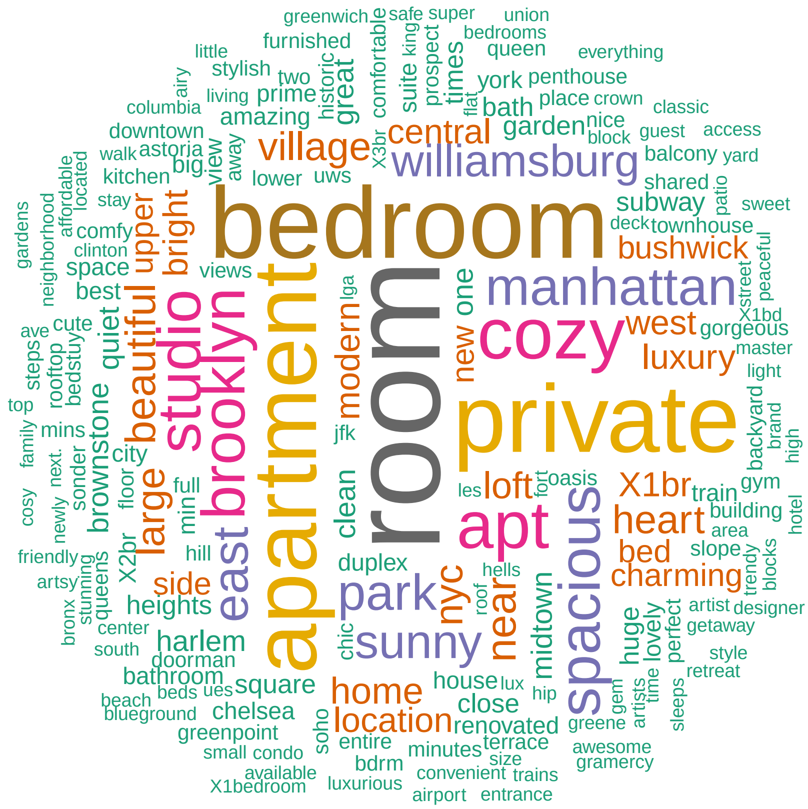
```
wordcloudfunction = function(namesSparse, seed = 123){
  set.seed(seed)
  m2 <- as.matrix(namesSparse)
  v2 <- sort(colSums(m2),decreasing=TRUE)
  d2 <- data.frame(word = names(v2),freq=v2)

  wordcloud(words = d2$word, freq = d2$freq, min.freq = 1,
            max.words=200, random.order=FALSE, rot.per=0.35,
            colors=brewer.pal(8, "Dark2"))
}
```

With the `wordcloudfunction`, we tried to make the wordcloud process reproducible. After geting the data frame of the frequencies, it shows the wordcloud plot. There will be some randomness in plotting. So, we set the seed before plotting.

To be able to plot the wordcloud, we need to prepare the data. Like in the Unit 5 of Analytics Edge, we need to apply these steps:

- Create a Corpus of the column (to be able to apply following processes)
- Convert all words to lower case (so that Airbnb and airbnb will be the same word)
- Remove all punctuations
- Remowe stopping words (like 'a, an, the' which are not any valuable words) ** In this part, we can remove additional words like 'airbnb', 'room' etc.
- Create a document term matrix (which has the unique words in the column and all observations in the row. The values are the number of occurence of that word in that sentence.)
- Stem the words (removing the suffix from the word. For example it turns "universal, university, universe" to "univers" or "apple, apples" to "appl"). We applied the first process without the stemming and applied the second process with stemming.
- Reduce the sparsity of the matrix (we can remove a word that only occured in one sentence. With this step, we can apply the plotting process faster. But, we didn't choose to do that. To be able to remove sparse words, we can use the function `removeSparseTerms(frequencies, 0.995)`. 0.995 means the ratio of the word occured in all sentence.)

```
corpus = VCorpus(VectorSource(airbnb$name))
corpus = tm_map(corpus, content_transformer(tolower))
corpus = tm_map(corpus, removePunctuation)
corpus = tm_map(corpus, removeWords, c("airbnb", stopwords("english")))
frequencies = DocumentTermMatrix(corpus)

namesSparse = as.data.frame(as.matrix(frequencies))
colnames(namesSparse) = make.names(colnames(namesSparse))

wordcloudfunction(namesSparse)
```




```
rm(frequencies, namesSparse)
```

As you can see in the plot, "bedroom", "room" and "private" words are the most common words in the `name` column. It means that most of the customers of Airbnb looks for the private rooms, so that these listings have these words in their names. As you can see from the plot that "brooklyn" and "manhattan" words are common in the name of the listings. We can infer that Brooklyn and Manhattan would have more listing than the others.

```
corpus = tm_map(corpus, stemDocument)
frequencies = DocumentTermMatrix(corpus)

namesSparse = as.data.frame(as.matrix(frequencies))
colnames(namesSparse) = make.names(colnames(namesSparse))

wordcloudfunction(namesSparse)
```

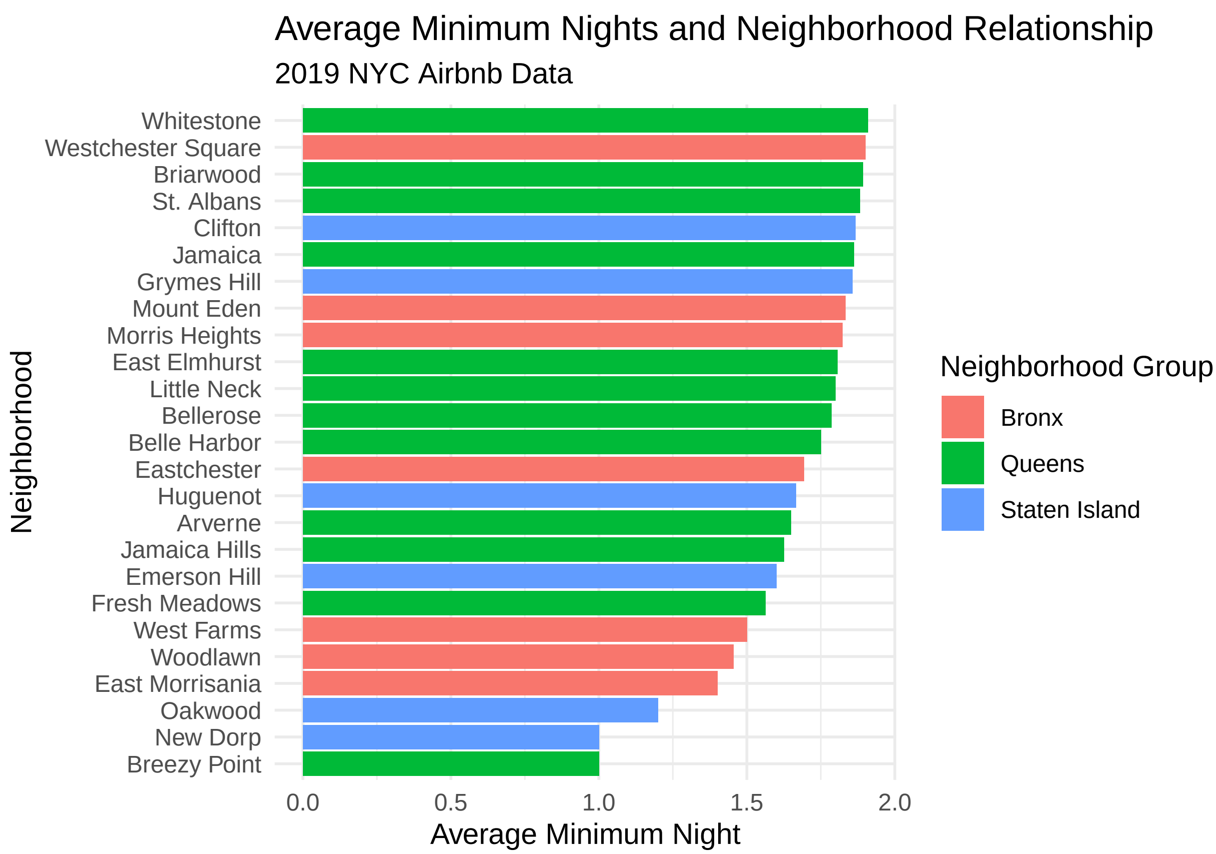

```r
rm(frequencies, namesSparse)
```

In this plot, we applied the stemming operation and then plot the words.As you can see, the word "apartment" is turned into "apart" and its occurence in the data becomes approximately the same with "privat" which stands for the "private" or "privates" words.

3.11 Minimum Nights and Neighborhood Relationship

In this part, we want to analyze the neighborhoods according to the average night of the `minimum_nights` variable.

```r
airbnb %>%
  group_by(neighbourhood, neighbourhood_group)%>%
  summarise(average_night = mean(minimum_nights))%>%
  arrange(average_night) %>%
  head(25)%>%

ggplot(., aes(y=reorder(neighbourhood, average_night), x= average_night, fill = neighbourhood_group)) +
  geom_col() +
  theme_minimal() +
  labs(title = "Average Minimum Nights and Neighborhood Relationship",
       subtitle = "2019 NYC Airbnb Data",
       x = "Average Minimum Night",
       y = "Neighborhood",
       fill = "Neighborhood Group")
```



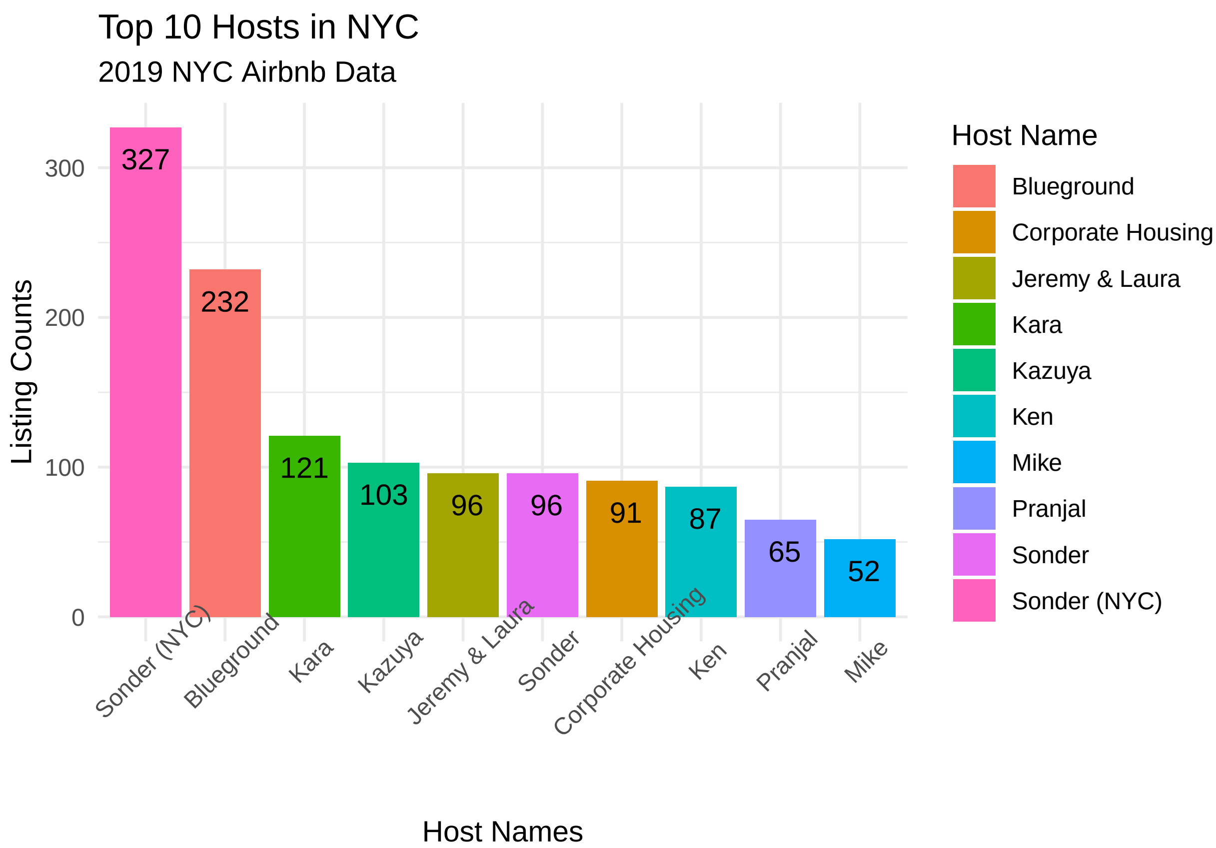
When we search information about NYC to make more clear analysis by reading this link, we realize that the most of the landmarks are located in Manhattan. Thus, we expect the more staying in this place. To check the assumption, we order the neighborhoods according to their average minimum nights. The above plot shows that neighborhoods in Manhattan are not included daily hosting.

3.12 The Most Popular Hosts in NYC Airbnb

Like we find the most popular neighborhoods, we can also determine the most popular host in NYC according to listing counts.

```
top_10_listing_counts = airbnb %>%
  group_by(host_id) %>%
  summarise(listing_count = n()) %>%
  arrange(desc(listing_count))

id_name = distinct(airbnb[, c("host_id", "host_name")])

top_10_listing_counts[1:10, ] %>%
  left_join(., id_name, by = "host_id") %>%
  ggplot(., aes(x = reorder(host_name, -listing_count) , y = listing_count, fill = host_name)) +
  geom_col() +
  theme_minimal() +
  geom_text(aes(label = format(listing_count,digits=3)), size=4, position = position_dodge(0.9),vjust =
  theme(axis.text.x = element_text(angle = 45), legend.position = "right") +
  labs(title = "Top 10 Hosts in NYC",
```

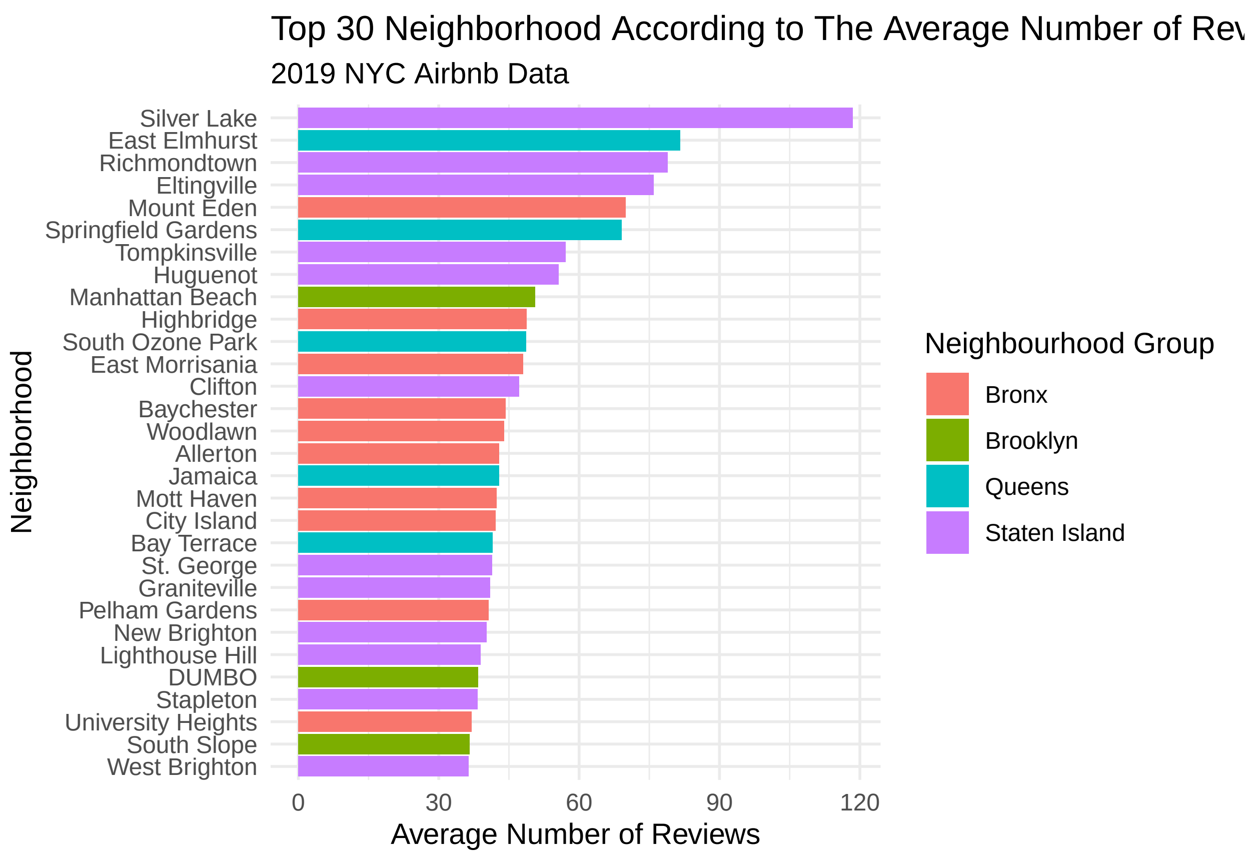
```
       subtitle = "2019 NYC Airbnb Data",
       x = "Host Names",
       y = "Listing Counts",
       fill = "Host Name")
```



3.13 The Average Number of Reviews in Each Neighboorhood

The another analysis can be made by using number of reviews.

```
airbnb %>%
  group_by(neighbourhood, neighbourhood_group) %>%
  summarise(mean_review = mean(number_of_reviews)) %>%
  arrange(desc(mean_review)) %>%
  head(30) %>%

  ggplot(aes(x=mean_review, y = reorder(neighbourhood,mean_review), fill = neighbourhood_group)) +
    geom_col() +
  theme_minimal() +
  labs(title = "Top 30 Neighborhood According to The Average Number of Reviews",
       subtitle = "2019 NYC Airbnb Data",
       x = "Average Number of Reviews",
       y = "Neighborhood",
       fill = "Neighbourhood Group")
```

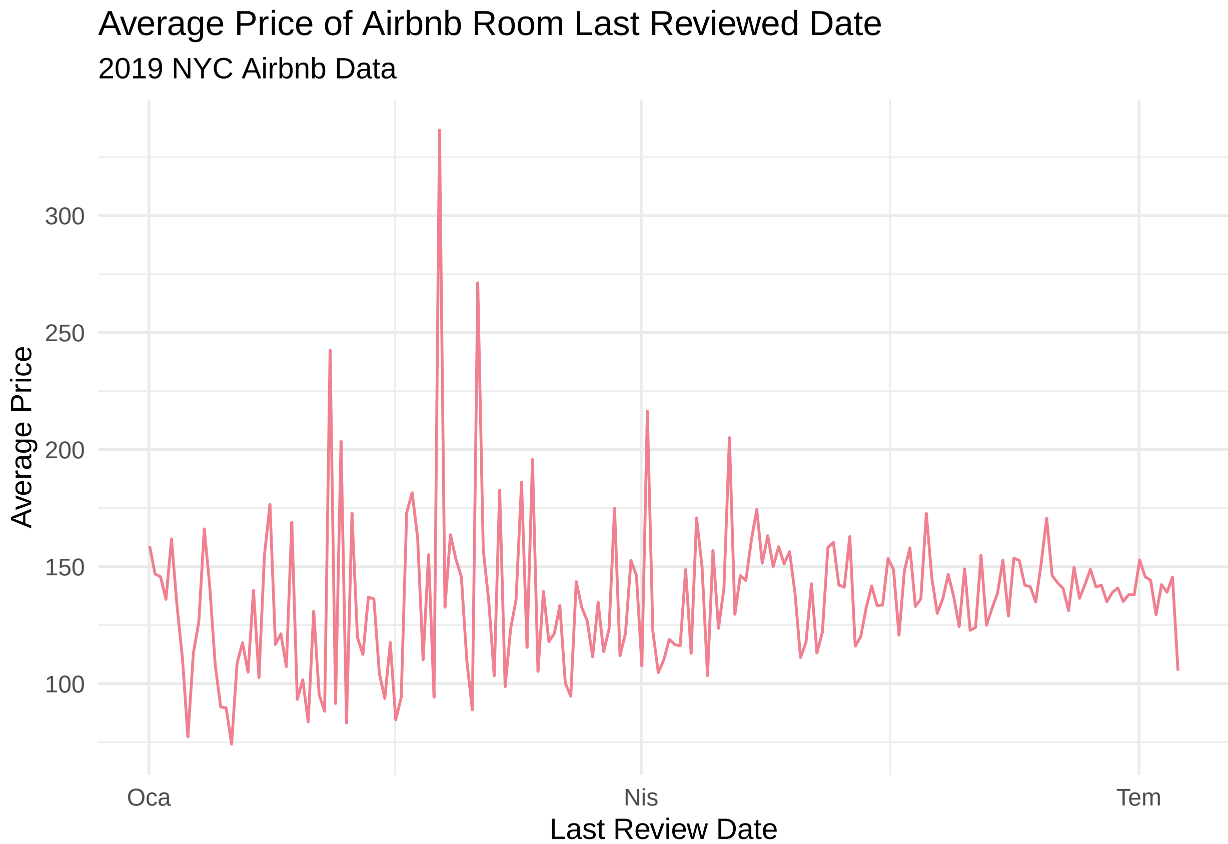


The results show that Bronx and Staten Island take the most of the reviews. On the other hand, there is no neighborhood from Manhattan in the top 30.

3.14 Last Review Analysis

In 2019, the fluctuation of the average price is getting smaller after April.

```
airbnb1 <- na.omit(airbnb)

airbnb1 %>%
  group_by(last_review) %>%
  transmute(last_review, average_price = mean(price)) %>%
  filter(lubridate::year(last_review) > 2018) %>%

  ggplot(aes(x=last_review, y = average_price)) +
  geom_line(color = "#F08090") +
  theme_minimal() +
  labs(title = "Average Price of Airbnb Room Last Reviewed Date",
       subtitle = "2019 NYC Airbnb Data",
       x = "Last Review Date",
       y = "Average Price")
```



4. Conclusion

In this study, we address the explanatory analysis of the airbnb data with several key features such as price, neighborhood, neighborhood group, room type, number of reviews, etc. By using these data,

- We obtain price and neighborhood relationship, i.e., Manhattan is the most expensive airbnb region when we compare the other neighborhood groups. On the other hand, the least expensive region is Bronx.
- Another analysis is conducted by using room type. The results show that the entire home/apt type is more preferable and the others are private room and shared room, respectively.
- To make a different analysis instead of numerical analysis, we use Wordcloud which makes text mining.
- Number of reviews are also investigated to find which neighborhoods take the most review according to the neighborhood group.

The other analysis made to calculate following relationships:

- The minimum nights and neighborhood relationship,
- The most popular hosts in airbnb in 2019,
- The average price and last review relationship in 2019,

References

To prepare this report we use some directive notes, reports, and web pages that are listed below:

- Lecture Notes
- Kaggle Data Set

The Extra Notebooks in Kaggle

- Exploratory Data Analysis(EDA) of NYC Airb
- NYC Airbnb EDA
- Analytics Edge