New York City 2019 Airbnb Data Analysis Report

Karan Dassi, Marzuq Khan, Amy Lu

December 11, 2019

```
library(tidyverse)
library(tigris)
library(leaflet)
library(sp)
library(ggmap)
library(maptools)
library(broom)
library(httr)
library(rgdal)
library(pdist)
library(webshot)
```

New York Airbnb 2019

New York is one of the most popular spot for tourists around the world. Airbnb has been changing the way tourists travel. Therefore, our team would like to explore the Airbnbs in New York City in 2019. Our project will start with descriptive analysis of the data, exploratory data analysis and then a creation of a Shiny App.

By creating a Shiny dashboard, we can visually understanding where the rentals are, how much each rental is, type of rentals are and the availability of each rental within the city.

With our dashboard, the tourists can review and select the rentals that they are interested in at ease based on their needs. We would also want to incooperate the NY subway dataset to the AirBnB datase so that we can see which rentals are close to or far from the subway stop.

- `id`: listing ID
- `name`: name of the listing
- `host_id`: host ID
- `host_name`: name of the host
- `neighbourhood_group`: location
- `neighbourhood`: area
- `latitude`: latitude coordinates
- `longitude`: longitude coordinates
- `room_type`: listing space type
- `price`: price in dollars
- `minimum_nights`: amount of nights minimum
- `number_of_reviews`: number of reviews
- `last_review`: latest review
- `reviews_per_month`: number of reviews per month
- `calculated_host_listings_count`: amount of listing per host
- `availability_365`: number of days when listing is available for booking

Read Data

```
airbnb <- read_csv("../data/AB_NYC_2019.csv")
```

```
## Parsed with column specification:
## 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()
## )
```

```
head(airbnb)
```

```
## # A tibble: 6 x 16
##      id name  host_id host_name neighbourhood_g~ neighbourhood latitude
##   <dbl> <chr>   <dbl> <chr>     <chr>            <chr>            <dbl>
## 1  2539 Clea~    2787 John      Brooklyn         Kensington        40.6
## 2  2595 Skyl~    2845 Jennifer  Manhattan        Midtown          40.8
## 3  3647 THE ~    4632 Elisabeth Manhattan        Harlem           40.8
## 4  3831 Cozy~    4869 LisaRoxa~ Brooklyn         Clinton Hill     40.7
## 5  5022 Enti~    7192 Laura     Manhattan        East Harlem      40.8
## 6  5099 Larg~    7322 Chris     Manhattan        Murray Hill      40.7
## # ... with 9 more variables: longitude <dbl>, room_type <chr>,
## #   price <dbl>, minimum_nights <dbl>, number_of_reviews <dbl>,
## #   last_review <date>, reviews_per_month <dbl>,
## #   calculated_host_listings_count <dbl>, availability_365 <dbl>
```

Analysis: There are total 48,895 observations with 16 variables of rental and host id, rental name, neigourhood group, neighbourhood, longtitude, latitude, room type, price, minimum_nights, number of reviews received, most recent review date, number of reviews per month, calculated amount of listing per host and the number of day in availibility for booking in 2019.

We may not need all the variables for our dasbhoard!

```
subway <- read_csv("../data/ny_subway.csv")
```

```
## Parsed with column specification:
## cols(
##   URL = col_character(),
##   OBJECTID = col_double(),
##   NAME = col_character(),
```

```
##   the_geom = col_character(),
##   LINE = col_character(),
##   NOTES = col_character()
## )
```

```
head(subway)
```

```
## # A tibble: 6 x 6
##   URL          OBJECTID NAME     the_geom        LINE   NOTES
##   <chr>           <dbl> <chr>    <chr>           <chr>  <chr>
## 1 http://web.~        1 Astor Pl POINT (-73.991~ 4-6-6~ 4 nights, 6-all ti~
## 2 http://web.~        2 Canal St POINT (-74.000~ 4-6-6~ 4 nights, 6-all ti~
## 3 http://web.~        3 50th St  POINT (-73.983~ 1-2    1-all times, 2-nig~
## 4 http://web.~        4 Bergen ~ POINT (-73.974~ 2-3-4  4-nights, 3-all ot~
## 5 http://web.~        5 Pennsyl~ POINT (-73.894~ 3-4    4-nights, 3-all ot~
## 6 http://web.~        6 238th St POINT (-73.900~ 1      1-all times, exit ~
```

Analysis: There are 473 observations in the NY subway dataset with 6 variables of URL of each location's URL page, object ID, location name, longtitude, latitude, the lines in each location and note which includes the train schedules.

Cleaning the Subway Dataset

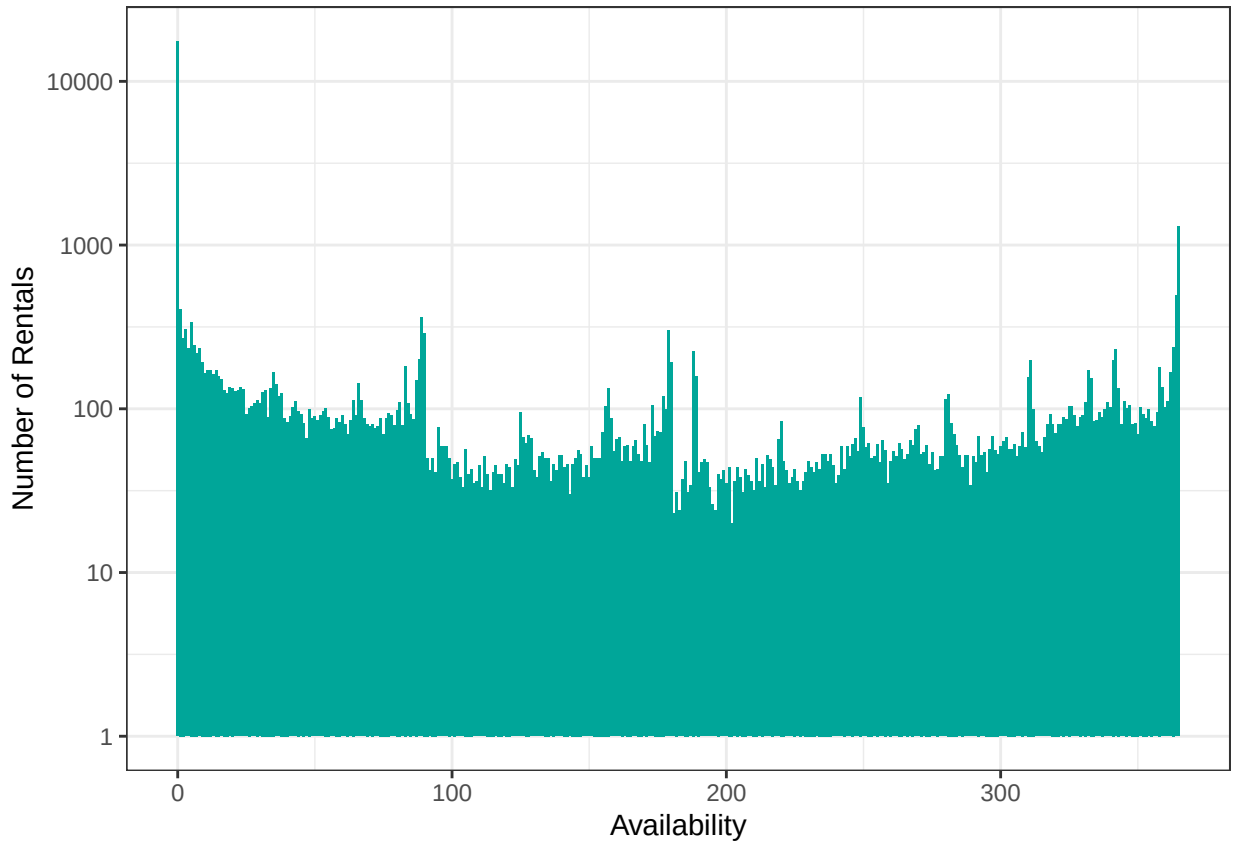
```
subway %>%
  select(-URL) %>%
  mutate(the_geom = str_remove_all(string = the_geom, pattern = "POINT ")) -> subway
subway %>%
  mutate(setup = str_remove_all(string = the_geom, pattern = "\\(")) %>%
  mutate(setup = str_remove_all(string = setup, pattern = "\\)")) %>%
  mutate(longitude = str_remove_all(string = setup, pattern = " .*$"),
         latitude = str_remove_all(string = setup, pattern = "^.* ")) %>%
  mutate(latitude = as.numeric(latitude),
         longitude = as.numeric(longitude)) %>%
  select(OBJECTID, NAME, LINE, latitude, longitude) -> cleaned_subway
#saveRDS(object = cleaned_subway, file = "../data/clean_sub.RDS") ## Only need this line the first run

nysub <- read_rds("../data/clean_sub.RDS")
head(nysub)
```

```
## # A tibble: 6 x 5
##   OBJECTID NAME              LINE          latitude longitude
##      <dbl> <chr>             <chr>            <dbl>     <dbl>
## 1        1 Astor Pl          4-6-6 Express     40.7     -74.0
## 2        2 Canal St          4-6-6 Express     40.7     -74.0
## 3        3 50th St           1-2               40.8     -74.0
## 4        4 Bergen St         2-3-4             40.7     -74.0
## 5        5 Pennsylvania Ave  3-4               40.7     -73.9
## 6        6 238th St          1                 40.9     -73.9
```

Exploratory Data Analysis

Number of days the listing is available for booking in 2019

```
airbnb%>%
  group_by(availability_365)%>%
  count()%>%
  ggplot(aes(x = availability_365, y = n)) +
  geom_col(fill = "#00A699") +
  theme_bw() +
  scale_y_log10() +
  xlab("Availability") +
  ylab("Number of Rentals")
```

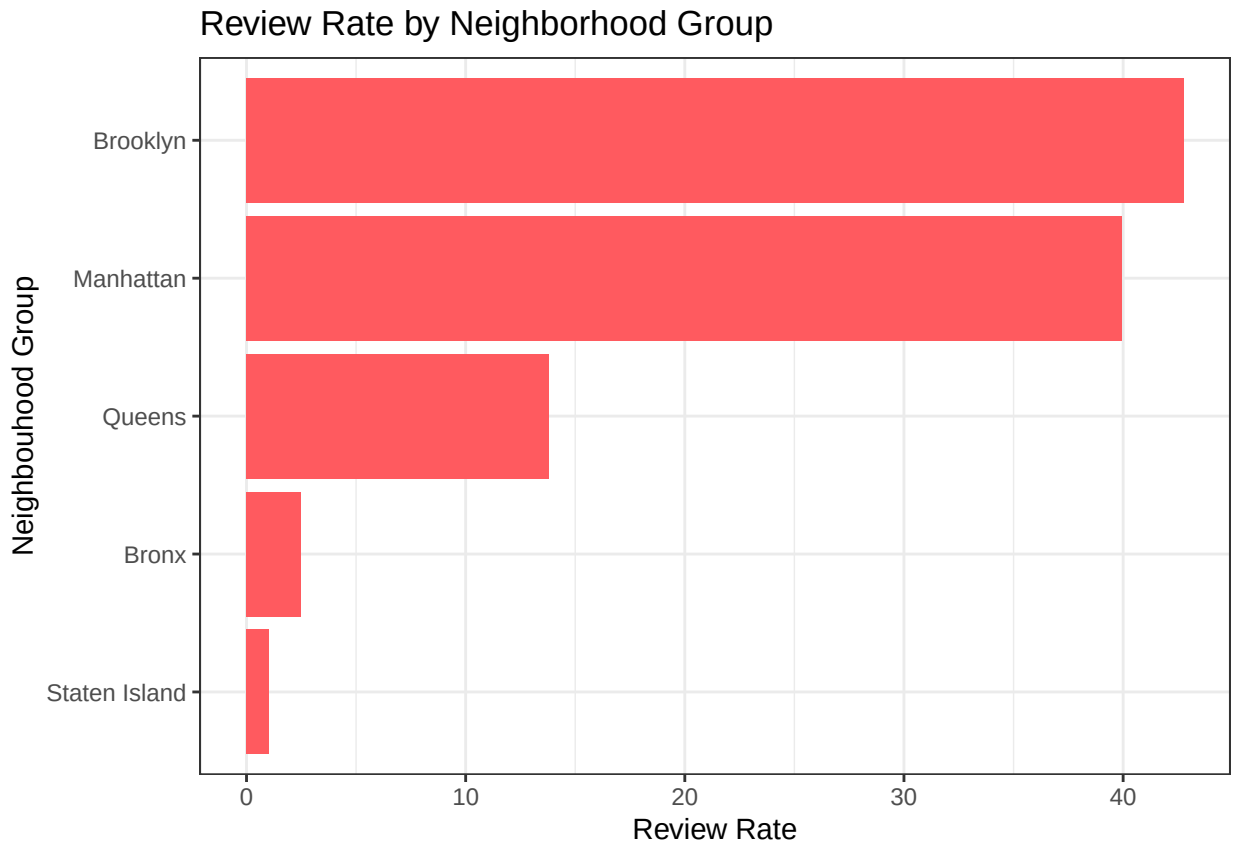


Findings: It seems there are various rentals available throughout the year. A good amount of the units seem to be booked with zero availability; however, there are still many units can be rented through out the year.

How popular Airbnb is by neighborhoods

We calculate the number of reviews across the neighborhoods to identify which location(s) have the most reviews (doesn't matter if it was positve or negative reviews, a review means a stay in the unit.)

```r
airbnb%>%
  group_by(neighbourhood_group)%>%
  summarise(total_review = sum(number_of_reviews))%>%
  mutate(percent_review = total_review/ sum(total_review)*100)%>%
  ggplot(aes(x = fct_reorder(neighbourhood_group, percent_review), y = percent_review)) +
  geom_col(fill = "#FF5A5F") +
  theme_bw() +
  ggtitle("Review Rate by Neighborhood Group") +
  xlab("Neighbouhood Group") +
  ylab("Review Rate") +
  coord_flip()
```



Analysis: Based on the plot, Brooklyn has the highest review rate for more than 40% among all 5 neighbourhood groups, following by Manhattan with approximately 40%. Both Brooklyn and Manhattan are the neighbourhood that are popular for the renters.

Listed price by neighbourhood_group

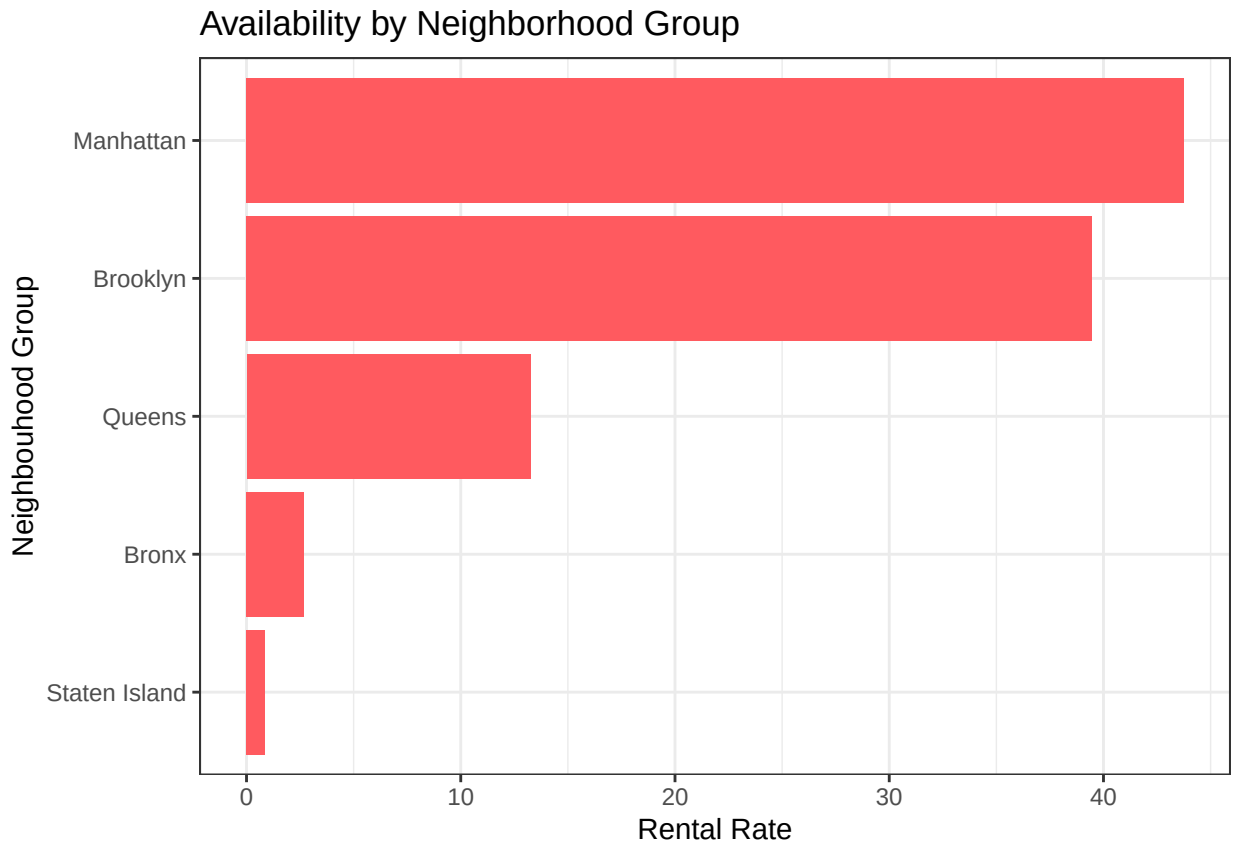
```r
airbnb%>%
  group_by(neighbourhood_group)%>%
  summarise(avg_price = mean(price))%>%
  ggplot(aes(x = fct_reorder(neighbourhood_group, avg_price), y = avg_price)) +
  geom_col(fill = "#FF5A5F") +
  theme_bw() +
```

```
  ggtitle("Rental Price by Neighborhood Group") +
  xlab("Neighbouhood Group") +
  ylab("Price") +
  coord_flip()
```

Analysis: Manhattan has the highest average rental price among all other Neighbouhood groups by almost 1/3 more. Interestingly, rentals in other area have simialr price range.

Neighbourhood vs listed rental rate

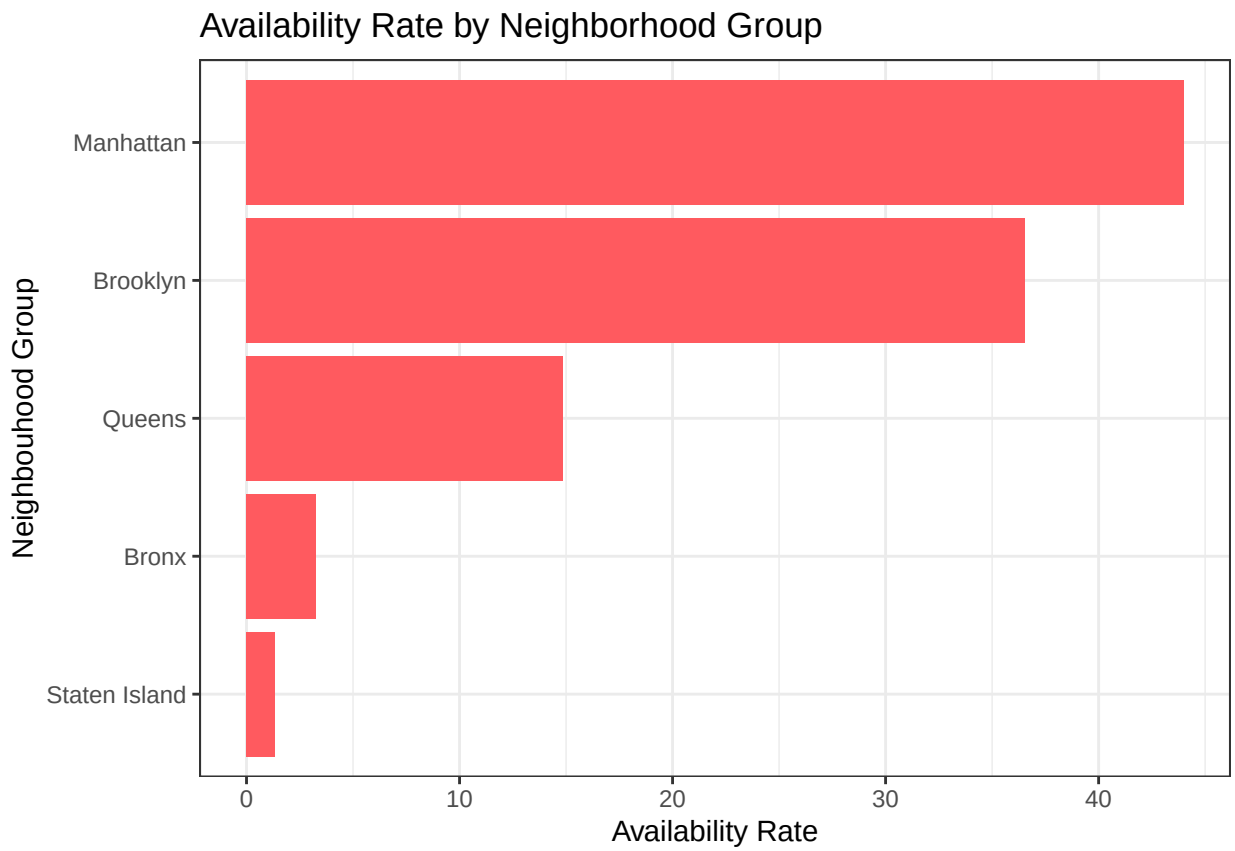
```
airbnb%>%
  group_by(neighbourhood_group)%>%
  summarise(sum_rental = sum(id))%>%
  mutate(percent_rental_count = sum_rental/ sum(sum_rental)*100)%>%
  ggplot(aes(x = fct_reorder(neighbourhood_group, percent_rental_count), y = percent_rental_count)) +ge
  theme_bw() +
  ggtitle("Availability by Neighborhood Group") +
  xlab("Neighbouhood Group") +
  ylab("Rental Rate") +
  coord_flip()
```



Analysis: Around 80% list rentals are gathering in Manhattan and Brooklyn which we believe it is expecting as these two Neighbouhood groups have the most attractions that the tourists are interested in and have the most entertainments and restaurants to explore as well.

Neighbourhood vs availability

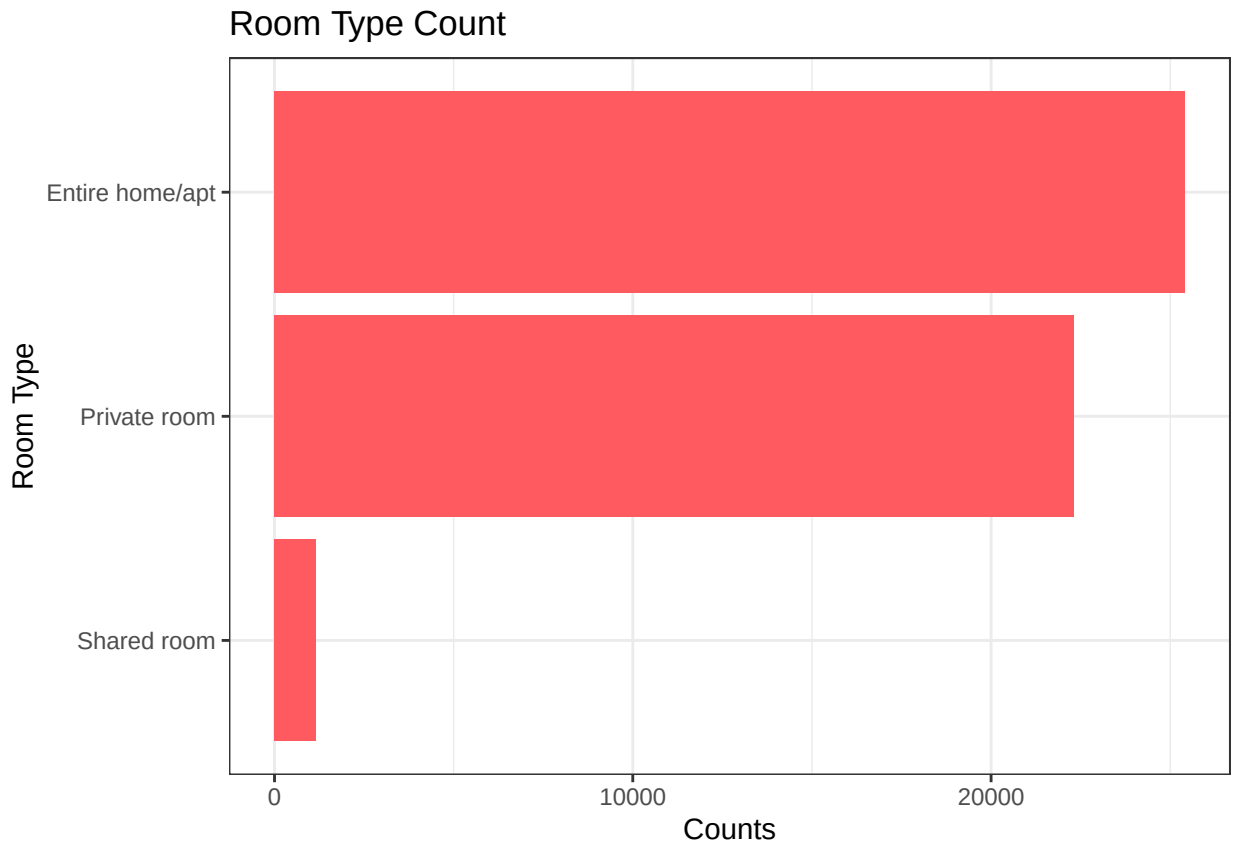
```
airbnb%>%
  group_by(neighbourhood_group)%>%
  summarise(sum_avalibility = sum(availability_365))%>%
  mutate(percent_avai = sum_avalibility/ sum(sum_avalibility)*100)%>%
  ggplot(aes(x = fct_reorder(neighbourhood_group, percent_avai), y = percent_avai)) +
  geom_col(fill = "#FF5A5F") +
  theme_bw() +
  ggtitle("Availability Rate by Neighborhood Group") +
  xlab("Neighbouhood Group") +
  ylab("Availability Rate") +
  coord_flip()
```



Analysis: Similar to the previous plot, most available rentals are also gathering in Manhattan and Brooklyn.

Type of rental

```
airbnb%>%
  group_by(room_type)%>%
  count(name = "type_count")%>%
  ggplot(aes(x = fct_reorder(room_type, type_count), y = type_count)) +
  geom_col(fill = "#FF5A5F") +
  theme_bw() +
  ggtitle("Room Type Count") +
  xlab("Room Type") +
  ylab("Counts") +
  coord_flip()
```

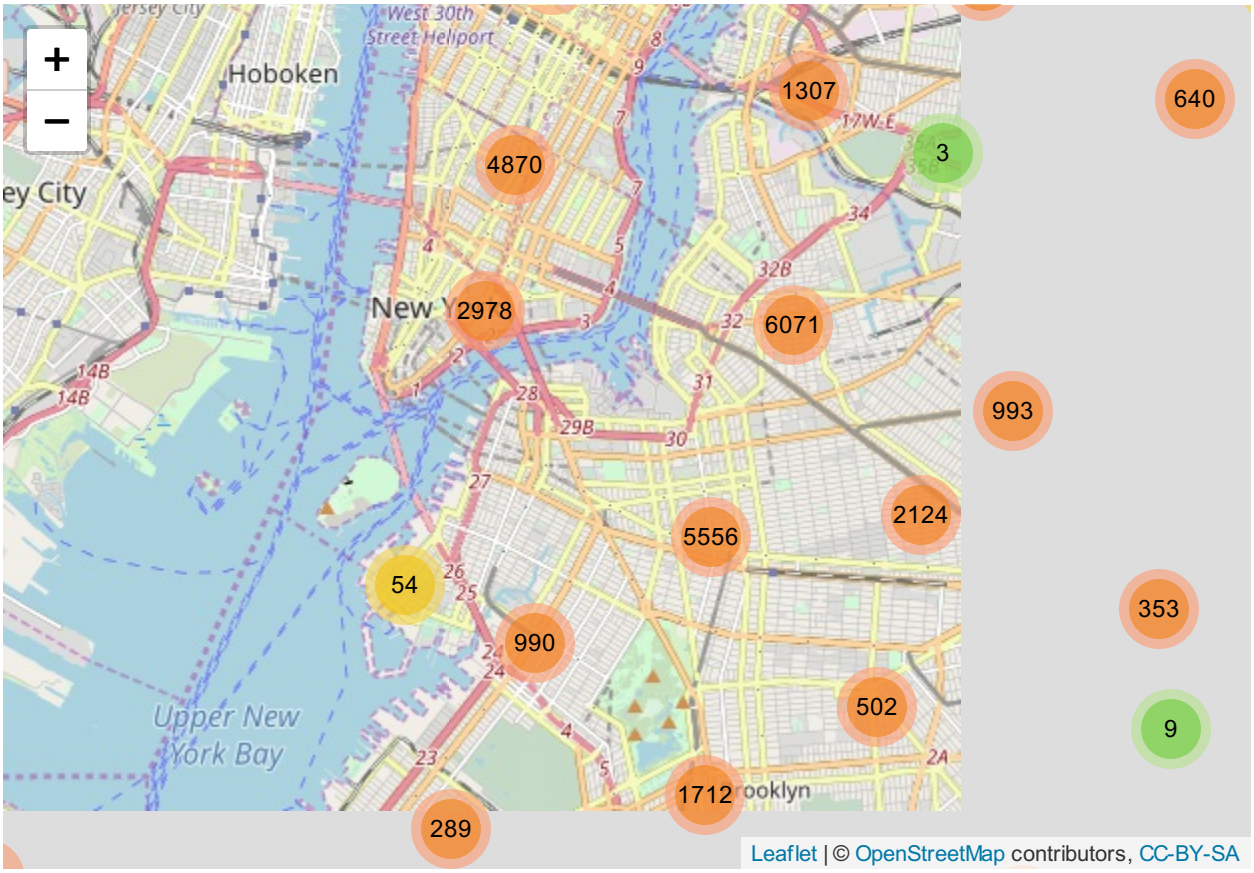


Analysis: Most rental types are either entire unit or private room. While shared room has the less within the city.

NYC map vs Airbnb rentals

```
airbnb%>%
  group_by(neighbourhood_group)%>%
  mutate(sum_rental = n())%>%
  select(name, neighbourhood_group, latitude, longitude, sum_rental) -> airbnb_count



leaflet(airbnb_count) %>%
  addTiles() %>%
  setView(-74.00, 40.71, zoom = 12)%>%
  addMarkers(clusterOptions = markerClusterOptions(), label = ~as.character(name))
```

```
## Assuming "longitude" and "latitude" are longitude and latitude, respectively
```



Analysis: This is a quick glance of what we would like to explore on our Shiny App. We would like to incooperate the location of each Airbnb rental with the closest subway stations to each unit.

Price versus Distance from Subway Stations

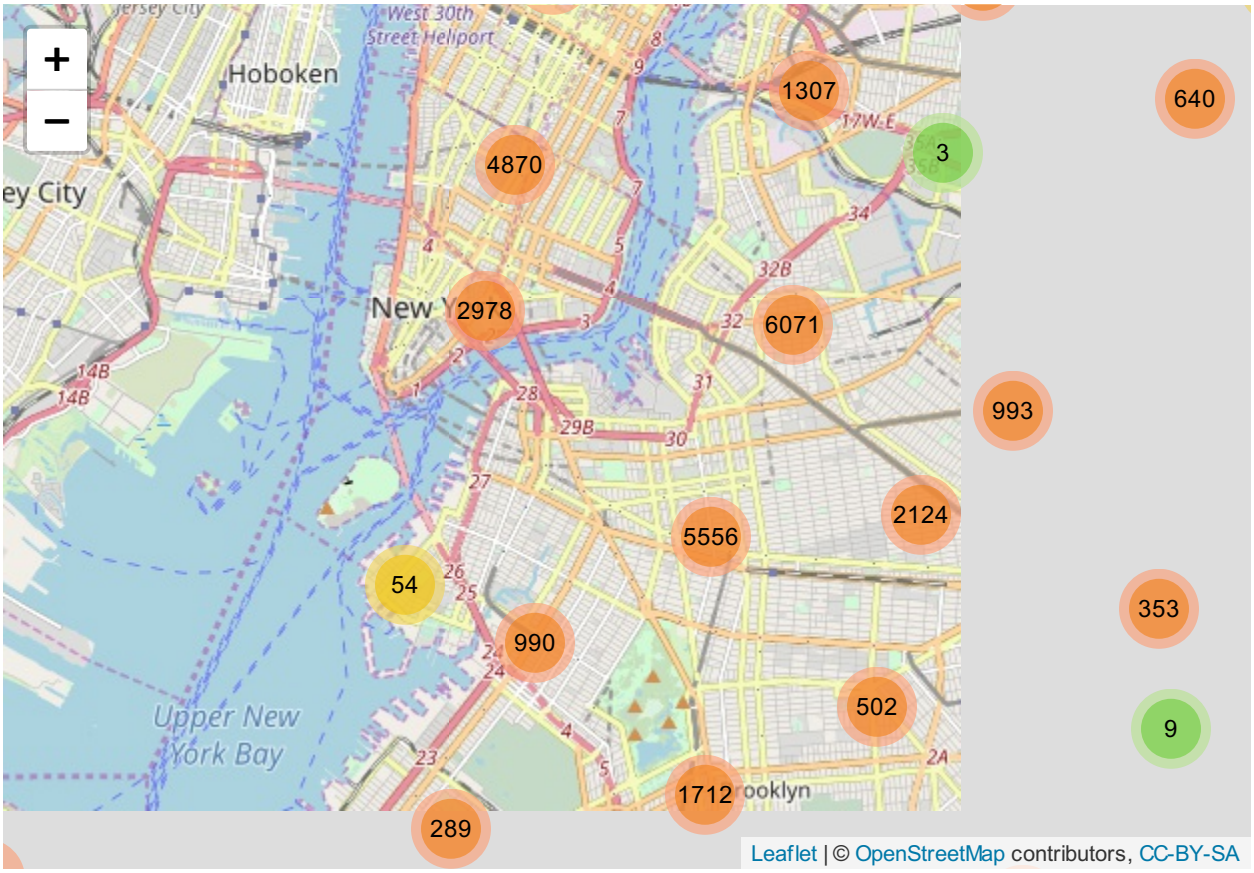
```
airbnb %>%
  select(name, neighbourhood_group, latitude, longitude, price) -> airbnb_price

airbnb_loc <- tibble("lat" = airbnb_price$latitude, "long" = airbnb_price$longitude)

sub_loc <- tibble("lat" = nysub$latitude, "long" = nysub$longitude)

distmat <- as.matrix(pdist(X = airbnb_loc, Y = sub_loc))

airbnb_loc$min_dist_from_sub <- apply(distmat, 1, min)
bind_cols(airbnb_price, "near_sub" = (airbnb_loc$min_dist_from_sub*87)) -> airbnb_price
#saveRDS(airbnb_price, "../data/airbnb_price.RDS") ## Only need this line the first run

airbnb_price <- read_rds("../data/airbnb_price.RDS")
glimpse(airbnb_price)
```

```
## Observations: 48,895
## Variables: 6
## $ name                <chr> "Clean & quiet apt home by the park", "Sky...
```

```
## $ neighbourhood_group <chr> "Brooklyn", "Manhattan", "Manhattan", "Bro...
## $ latitude            <dbl> 40.64749, 40.75362, 40.80902, 40.68514, 40...
## $ longitude           <dbl> -73.97237, -73.98377, -73.94190, -73.95976...
## $ price               <dbl> 149, 225, 150, 89, 80, 200, 60, 79, 79, 15...
## $ near_sub            <dbl> 0.41238577, 0.08635225, 0.33147818, 0.3268...
```

```
leaflet(airbnb_price) %>%
  addTiles() %>%
  setView(-74.00, 40.71, zoom = 12)%>%
  addMarkers(clusterOptions = markerClusterOptions(), label = ~as.character(price))
```

```
## Assuming "longitude" and "latitude" are longitude and latitude, respectively
```

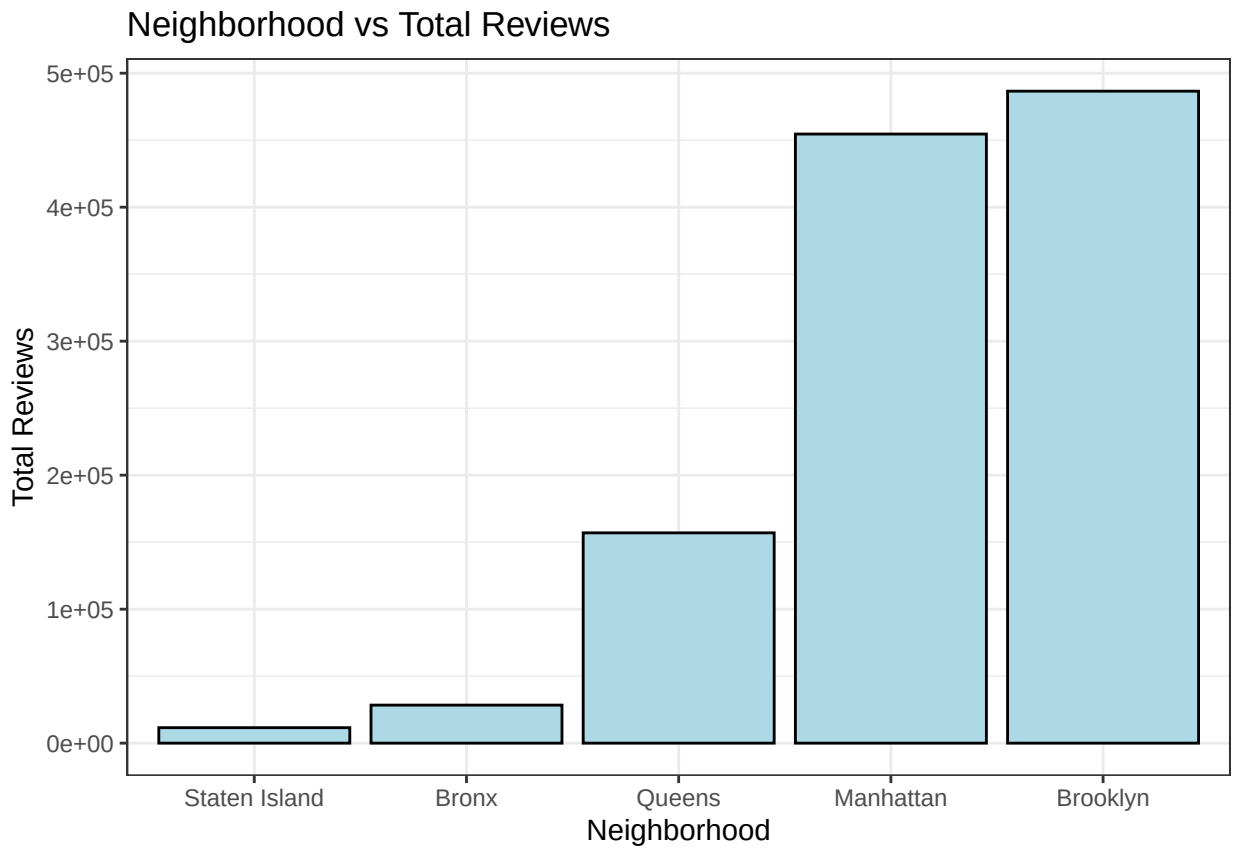


Analysis: Set up and another look at what we try to improve on with the app.

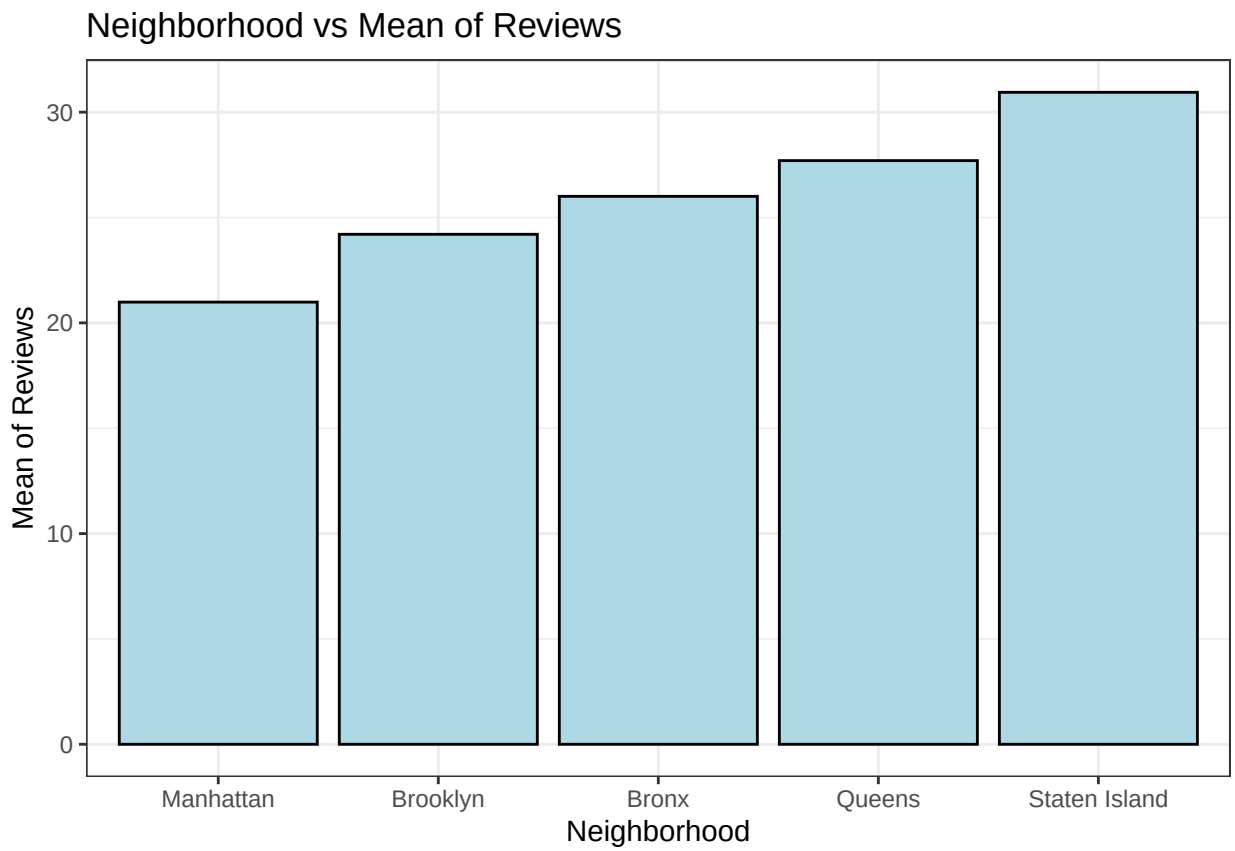
Exploring the reviews per neighborhood group to get popularity

```
airbnb %>%
  group_by(neighbourhood_group) %>%
  summarise(sumReview = sum(number_of_reviews)) %>%
  ggplot(aes(x=fct_reorder(neighbourhood_group, sumReview), y=sumReview)) +
  geom_col(fill = "light blue", color = "black") +
  ylab("Total Reviews") +
```

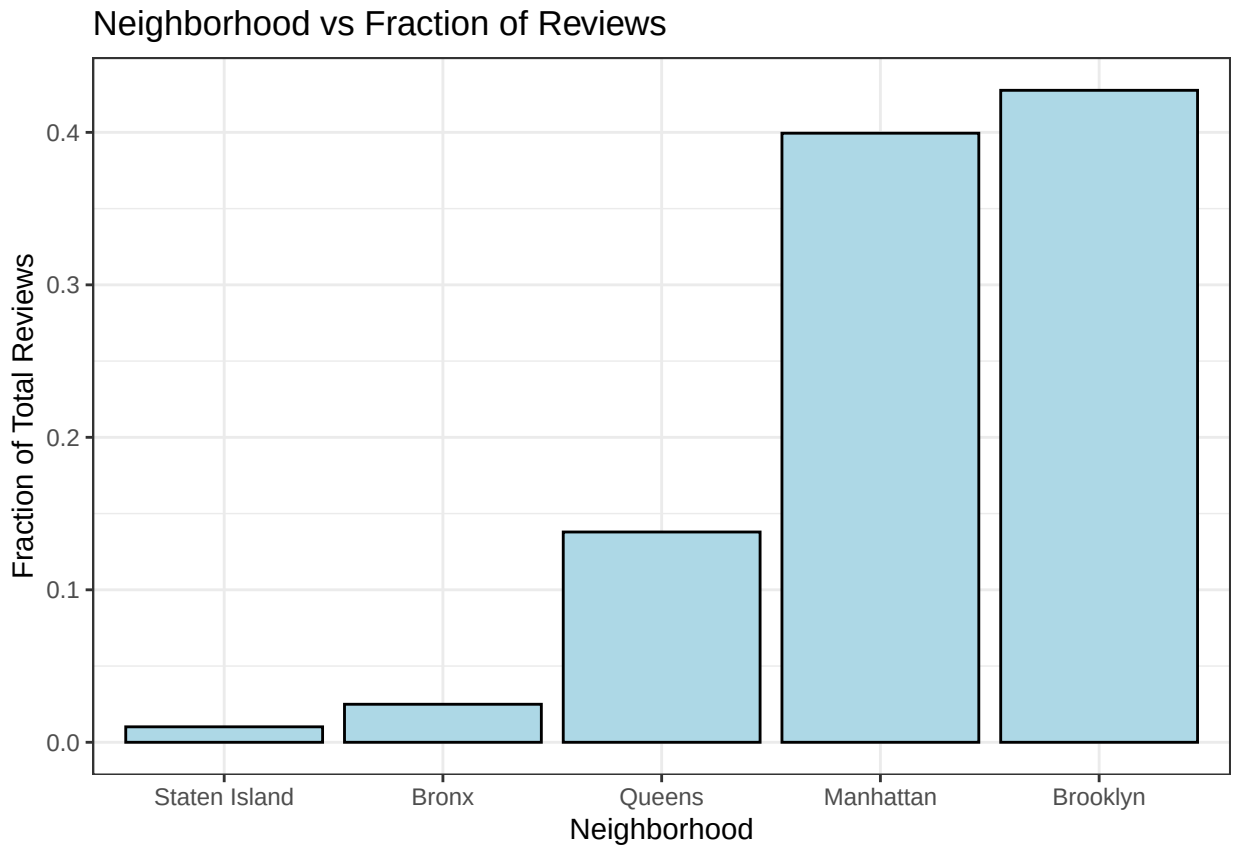
```
  xlab("Neighborhood") +
  ggtitle("Neighborhood vs Total Reviews") +
  theme_bw()
```




```
airbnb %>%
  group_by(neighbourhood_group) %>%
  summarise(meanReview = mean(number_of_reviews)) %>%
  ggplot(aes(x=fct_reorder(neighbourhood_group, meanReview), y=meanReview)) +
  geom_col(fill = "light blue", color = "black") +
  ylab("Mean of Reviews") +
  xlab("Neighborhood") +
  ggtitle("Neighborhood vs Mean of Reviews") +
  theme_bw()
```




```
airbnb %>%
  group_by(neighbourhood_group) %>%
  summarise(sumReview = sum(number_of_reviews)) %>%
  mutate(fracReview = sumReview/sum(sumReview)) %>%
  ggplot(aes(x=fct_reorder(neighbourhood_group, fracReview), y=fracReview)) +
  geom_col(fill = "light blue", color = "black") +
  ylab("Fraction of Total Reviews") +
  xlab("Neighborhood") +
  ggtitle("Neighborhood vs Fraction of Reviews") +
  theme_bw()
```




```
airbnb %>%
  select(neighbourhood_group, number_of_reviews) %>%
  group_by(neighbourhood_group) %>%
  count()
```

```
## # A tibble: 5 x 2
## # Groups:   neighbourhood_group [5]
##   neighbourhood_group     n
##   <chr>               <int>
## 1 Bronx                1091
## 2 Brooklyn            20104
## 3 Manhattan           21661
## 4 Queens               5666
## 5 Staten Island         373
```

```
airbnb %>%
  select(neighbourhood_group, number_of_reviews) %>%
  group_by(neighbourhood_group) %>%
  summarise(sumofReviews = sum(number_of_reviews))
```

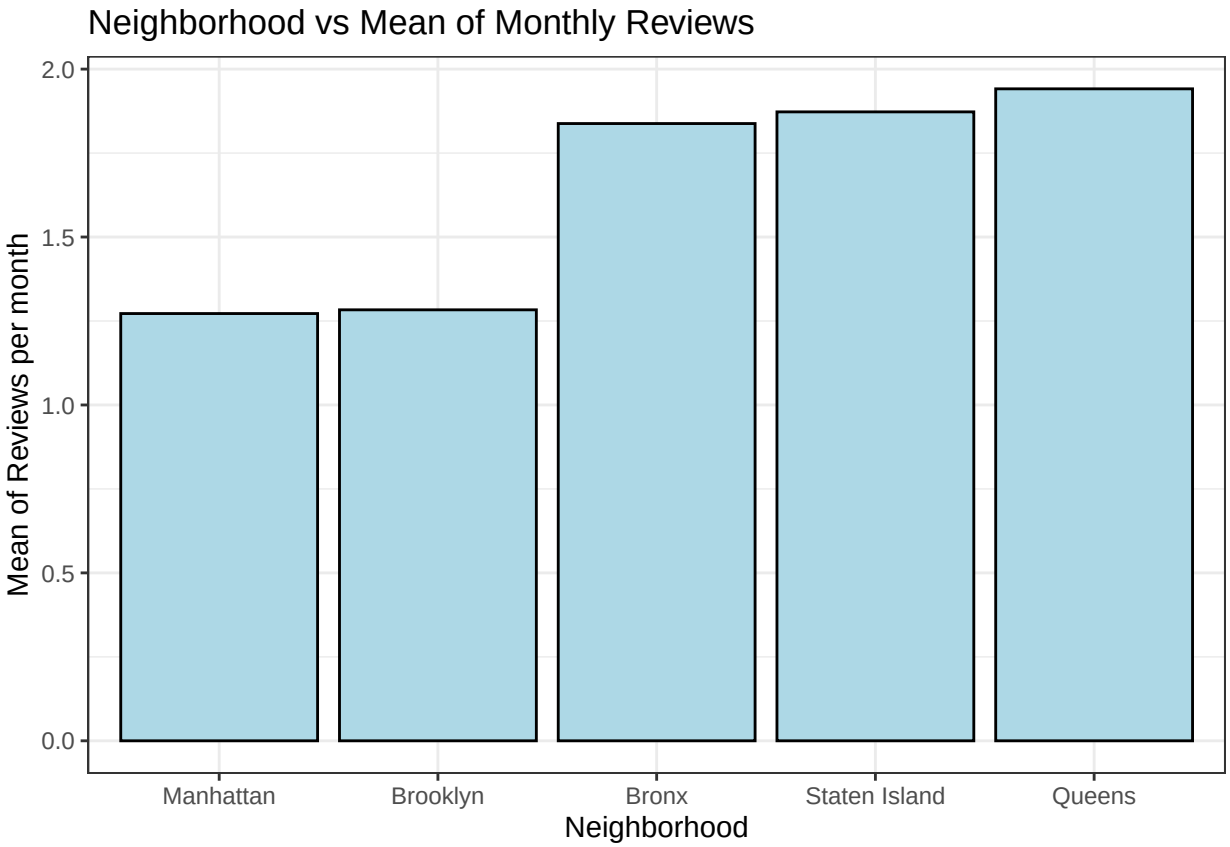
```
## # A tibble: 5 x 2
##   neighbourhood_group sumofReviews
##   <chr>                     <dbl>
## 1 Bronx                     28371
## 2 Brooklyn                 486574
```

```
## 3 Manhattan                454569
## 4 Queens                   156950
## 5 Staten Island             11541
```

Analysis: The above plots clearly show that Brooklyn and Manhattam have the maximum number of reviews in total but the mean of reviews is greater for staten island and Queens which may be because they just have less reviews recordings as they not as popular as manhattan and brooklyn. This disparity may be explained by the fewer number of observations recorderd for the neighborhood groups with higher mean.

More exploration using reviews per month:

```
airbnb %>%
  group_by(neighbourhood_group) %>%
  summarise(meanReview = mean(reviews_per_month, na.rm = T)) %>%
  ggplot(aes(x=fct_reorder(neighbourhood_group, meanReview), y=meanReview)) +
  geom_col(color = "black", fill = "light blue") +
  ylab("Mean of Reviews per month") +
  xlab("Neighborhood") +
  ggtitle("Neighborhood vs Mean of Monthly Reviews") +
  theme_bw()
```



Analysis: We believe the average of reviews per month is a better way of known which neighborhood groups have more reviews per month, here Queens, Staten Island and Bronx have the maximum amount of reviews per month.

Explore what role does minimum_nights play?

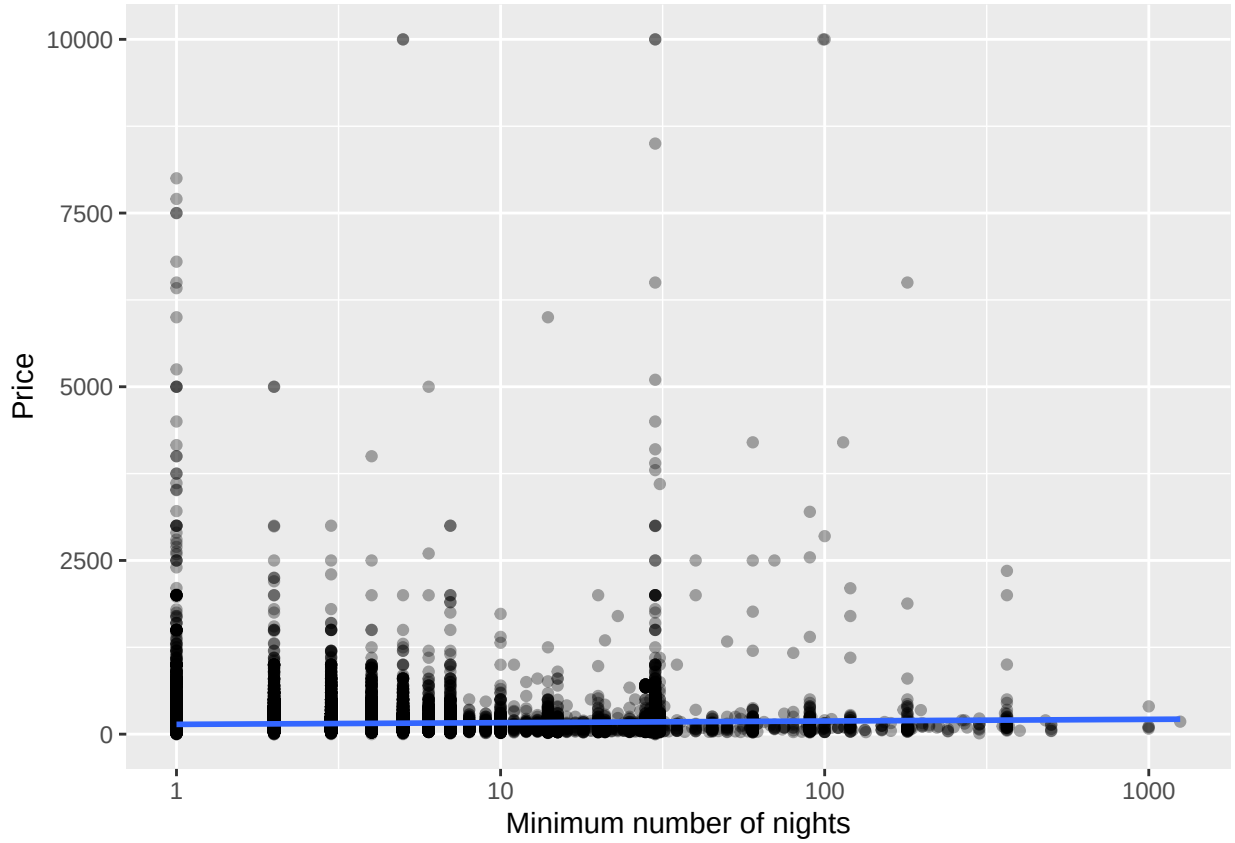
```
airbnb %>%
  group_by(id) %>%
  count() %>%
  filter(n>1)
```

```
## # A tibble: 0 x 2
## # Groups:   id [0]
## # ... with 2 variables: id <dbl>, n <int>
```

```
# So id is what is making the rows unique only

glimpse(airbnb)
```

```
## Observations: 48,895
## Variables: 16
## $ id                              <dbl> 2539, 2595, 3647, 3831, 5022, 5...
## $ name                            <chr> "Clean & quiet apt home by the ...
## $ host_id                         <dbl> 2787, 2845, 4632, 4869, 7192, 7...
## $ host_name                       <chr> "John", "Jennifer", "Elisabeth"...
## $ neighbourhood_group             <chr> "Brooklyn", "Manhattan", "Manha...
## $ neighbourhood                   <chr> "Kensington", "Midtown", "Harle...
## $ latitude                        <dbl> 40.64749, 40.75362, 40.80902, 4...
## $ longitude                       <dbl> -73.97237, -73.98377, -73.94190...
## $ room_type                       <chr> "Private room", "Entire home/ap...
## $ price                           <dbl> 149, 225, 150, 89, 80, 200, 60,...
## $ minimum_nights                  <dbl> 1, 1, 3, 1, 10, 3, 45, 2, 2, 1,...
## $ number_of_reviews               <dbl> 9, 45, 0, 270, 9, 74, 49, 430, ...
## $ last_review                     <date> 2018-10-19, 2019-05-21, NA, 20...
## $ reviews_per_month               <dbl> 0.21, 0.38, NA, 4.64, 0.10, 0.5...
## $ calculated_host_listings_count <dbl> 6, 2, 1, 1, 1, 1, 1, 1, 1, 4, 1...
## $ availability_365                <dbl> 365, 355, 365, 194, 0, 129, 0, ...
```

```
# Relation between minimum nights and price

airbnb %>%
  ggplot(aes(y=price, x=minimum_nights)) +
  geom_point(alpha = 1/3) +
  scale_x_log10() +
  xlab("Minimum number of nights") +
  ylab("Price") +
  geom_smooth(method = lm, se = F)
```

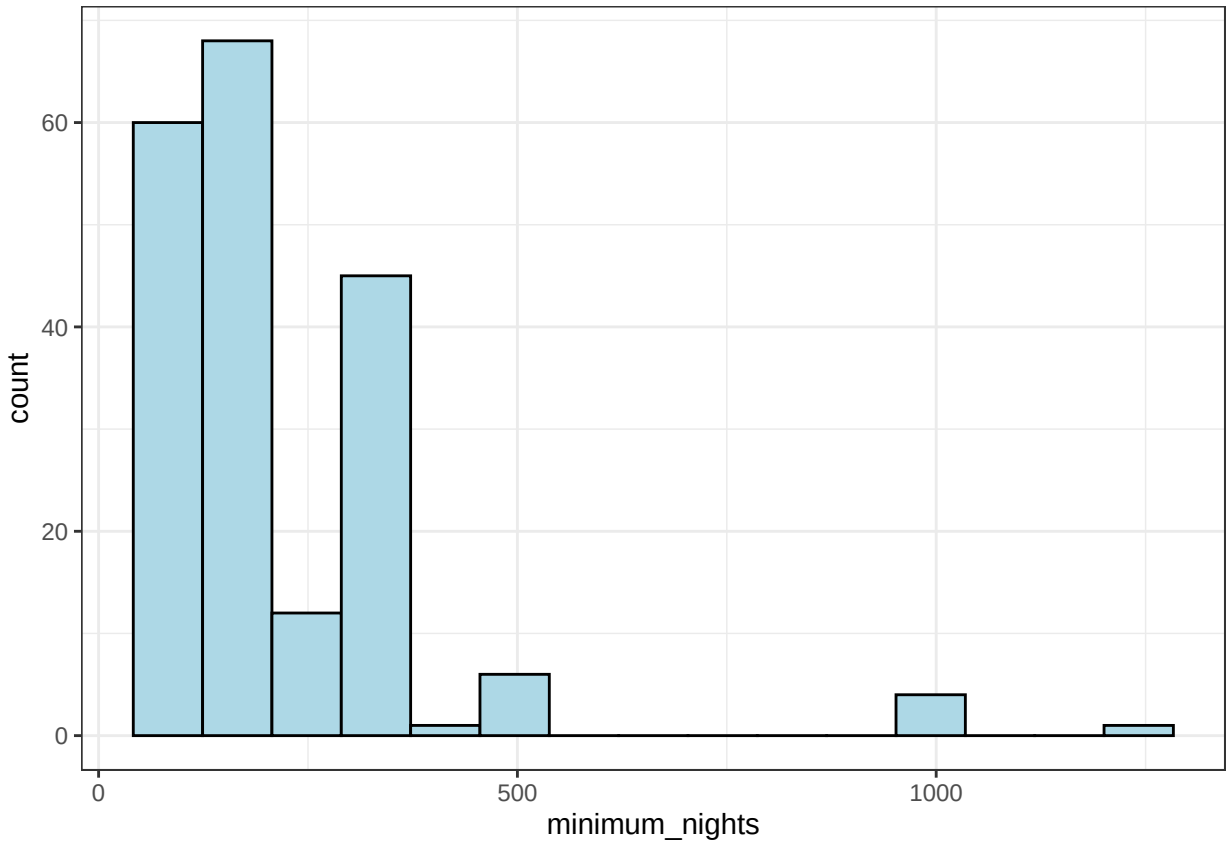



```
# Checking minimum nights

airbnb %>%
  arrange(-minimum_nights) %>%
  head(n=50) %>%
  select(name, host_name, price, minimum_nights)
```

```
## # A tibble: 50 x 4
##    name                                host_name price minimum_nights
##    <chr>                               <chr>     <dbl>          <dbl>
##  1 Prime W. Village location 1 bdrm    Genevieve   180           1250
##  2 <NA>                                Peter       400           1000
##  3 Historic Designer 2 Bed. Apartment  Glenn H.     99            999
##  4 Beautiful place in Brooklyn! #2     Angie        79            999
##  5 Shared Studio (females only)        Meg         110            999
##  6 Beautiful Fully Furnished 1 bed/bth Aliya       134            500
##  7 Wonderful Large 1 bedroom           John         75            500
##  8 Zen Room in Crown Heights Brooklyn  Laura        50            500
##  9 Peaceful apartment close to F/G     Amanda       45            500
## 10 Williamsburg Apartment              Meg         140            500
## # ... with 40 more rows
```

```
airbnb %>%
  filter(minimum_nights > 90) %>%
  ggplot(aes(x=minimum_nights)) +
```

```
  geom_histogram(fill = "light blue", color = "black", bins = 15) +
  theme_bw()
```



Analysis: It is interesting to note that many airbnb listings require over 3 months of minimum nights. Quite a few even require a full year's stay (365 minimum nights). One listing even requires 1250 minimum nights (~3.42 years), which is hard to believe. It is understandable that a host would like to have a steady income through renting out their property, however it seems unlikely many people would actually agree to stay for a minimum of over 1 year. Generally it does not seems to be the purpose of Airbnb to help people find long term housing, but it seems as though it can be used for that as well.

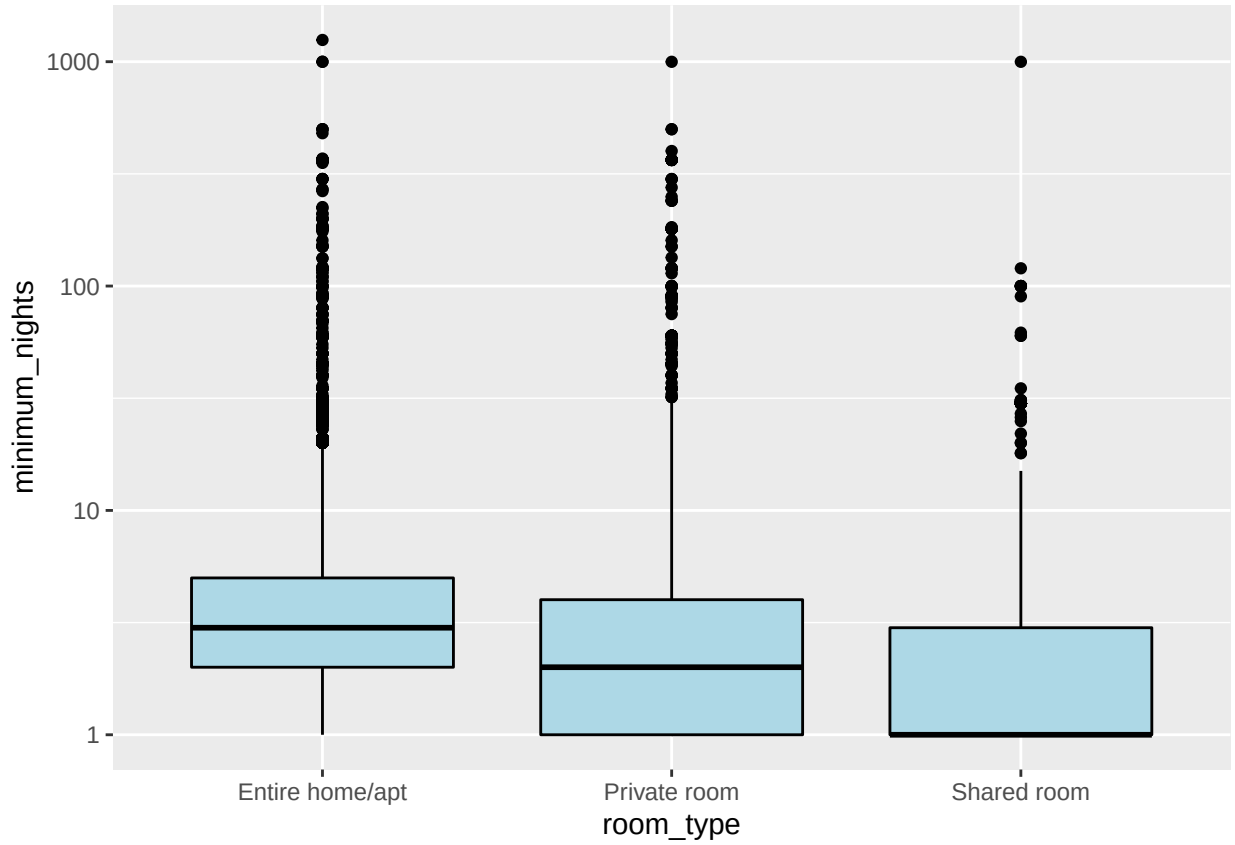
There is not a clear relationship betwwen minimum number of nights required and price. Still we can say that the price ddecreases as the number of nights increases

```
# minimum nights vs room type
glimpse(airbnb)
```

```
## Observations: 48,895
## Variables: 16
## $ id                            <dbl> 2539, 2595, 3647, 3831, 5022, 5...
## $ name                          <chr> "Clean & quiet apt home by the ...
## $ host_id                       <dbl> 2787, 2845, 4632, 4869, 7192, 7...
## $ host_name                     <chr> "John", "Jennifer", "Elisabeth"...
## $ neighbourhood_group           <chr> "Brooklyn", "Manhattan", "Manha...
## $ neighbourhood                 <chr> "Kensington", "Midtown", "Harle...
## $ latitude                      <dbl> 40.64749, 40.75362, 40.80902, 4...
## $ longitude                     <dbl> -73.97237, -73.98377, -73.94190...
```

```
## $ room_type                       <chr> "Private room", "Entire home/ap...
## $ price                           <dbl> 149, 225, 150, 89, 80, 200, 60,...
## $ minimum_nights                  <dbl> 1, 1, 3, 1, 10, 3, 45, 2, 2, 1,...
## $ number_of_reviews               <dbl> 9, 45, 0, 270, 9, 74, 49, 430, ...
## $ last_review                     <date> 2018-10-19, 2019-05-21, NA, 20...
## $ reviews_per_month               <dbl> 0.21, 0.38, NA, 4.64, 0.10, 0.5...
## $ calculated_host_listings_count <dbl> 6, 2, 1, 1, 1, 1, 1, 1, 1, 4, 1...
## $ availability_365                <dbl> 365, 355, 365, 194, 0, 129, 0, ...
```

```
airbnb %>%
  ggplot(aes(x=room_type, y=minimum_nights)) +
  geom_boxplot(fill = "light blue", color = "black") +
  scale_y_log10()
```



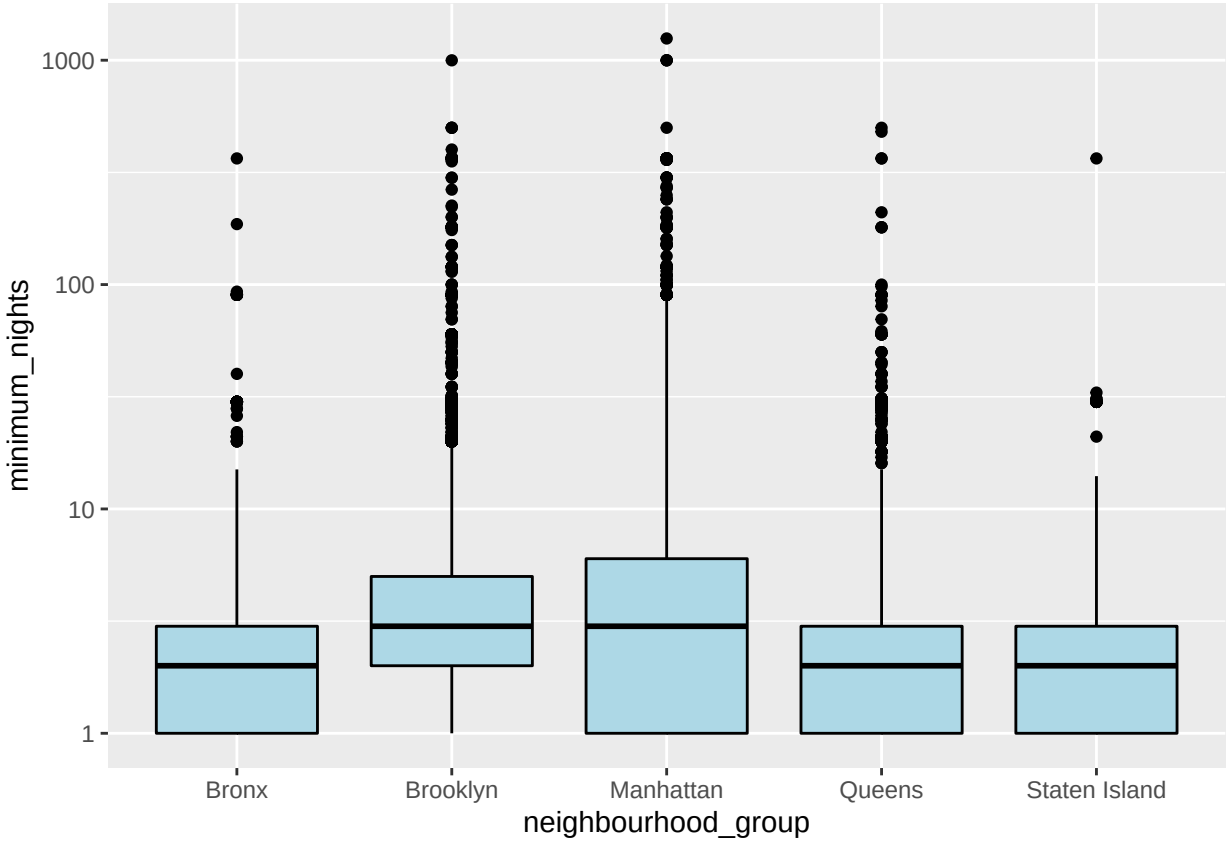
Analysis: We can see that room type does affect minimum number of nights spent which is the lowest for shared rooms but highest for entire home/apt which is also logically correct

```
glimpse(airbnb)
```

```
## Observations: 48,895
## Variables: 16
## $ id                              <dbl> 2539, 2595, 3647, 3831, 5022, 5...
## $ name                            <chr> "Clean & quiet apt home by the ...
## $ host_id                         <dbl> 2787, 2845, 4632, 4869, 7192, 7...
## $ host_name                       <chr> "John", "Jennifer", "Elisabeth"...
```

```
## $ neighbourhood_group            <chr> "Brooklyn", "Manhattan", "Manha...
## $ neighbourhood                  <chr> "Kensington", "Midtown", "Harle...
## $ latitude                       <dbl> 40.64749, 40.75362, 40.80902, 4...
## $ longitude                      <dbl> -73.97237, -73.98377, -73.94190...
## $ room_type                      <chr> "Private room", "Entire home/ap...
## $ price                          <dbl> 149, 225, 150, 89, 80, 200, 60,...
## $ minimum_nights                 <dbl> 1, 1, 3, 1, 10, 3, 45, 2, 2, 1,...
## $ number_of_reviews              <dbl> 9, 45, 0, 270, 9, 74, 49, 430, ...
## $ last_review                    <date> 2018-10-19, 2019-05-21, NA, 20...
## $ reviews_per_month              <dbl> 0.21, 0.38, NA, 4.64, 0.10, 0.5...
## $ calculated_host_listings_count <dbl> 6, 2, 1, 1, 1, 1, 1, 1, 1, 4, 1...
## $ availability_365               <dbl> 365, 355, 365, 194, 0, 129, 0, ...
```

```
airbnb %>%
  ggplot(aes(x=neighbourhood_group, y=minimum_nights)) +
  geom_boxplot(fill = "light blue", color = "black") +
  scale_y_log10()
```



Airbnb Shiny App

We would recommend running the R file in the app folder for the best result as well as looking over the README file in that same folder for some insight on its applications

```
library(shiny)
library(DT)
library(tidyverse)
library(ggstance)
library(broom)
library(ggthemes)
library(leaflet)
library(shinythemes)
library(tigris)
library(sp)
library(maptools)
library(httr)
library(rgdal)
library(ui)
library(rsconnect)
library(plotly)

# Karan
airbnb <- read_csv("../data/AB_NYC_2019.csv")
airbnbd <- airbnb
airbnb %>%
  dplyr::select(-latitude, -longitude) ->
  airbnbR


# Marzuq
airbnb_price <- read_rds("../data/airbnb_price.RDS")
glimpse(airbnb_price)
nysub <- read_rds("../data/clean_sub.RDS")
glimpse(nysub)

# Amy
head(airbnb)
names(airbnb)[5] <- "borough"
airbnb <- airbnb%>%
  select(id, name, host_id, borough, latitude, longitude, room_type, price, number_of_reviews)%>%
  mutate(id = as.factor(id),
         host_id = as.factor(host_id))
borough <- c("Brooklyn", "Manhattan", "Queens", "Staten Island", "Bronx")
room_type <- c("Private room", "Entire home/apt", "Shared room")
pal <- colorFactor(c("#FF5A5F", "#00A699","#767676"), domain = c("Entire home/apt", "Private room","Sha
max(airbnb$price)
min(airbnb$price)
max(airbnb$number_of_reviews)
min(airbnb$number_of_reviews)

ui <- fluidPage(shinythemes::themeSelector(),
                  fluidRow(
                  column(4,
                         titlePanel("New York City Airbnb")
                  ),
                  column(4
                  ),
```

```
      column(4,
             tags$img(src = "airbnb.png",  height = "60")
      )
    ),
    tabsetPanel(
      tabPanel("Dataset",
               dataTableOutput("dt")
      ),
      tabPanel("Histograms",
               sidebarLayout(
                 sidebarPanel(
                   varSelectInput("univar", "Variable to Plot", data = airbnbR, selected =
                   checkboxInput("unilog", "Log X"),
                   sliderInput("unibins", "Bins", min = 1, max = 100, value = 20),
                   numericInput("uninull", "Null Value", value = 0),
                   tableOutput("unitest_results")
                 ),
                 mainPanel(
                   plotOutput("hist")
                 )
               )
      ),
      tabPanel("Plots",
               sidebarLayout(
                 sidebarPanel(
                   varSelectInput("var1", "Variable X", data = airbnbR, selected = "neighbou
                   checkboxInput("var1log", "Log X"),
                   varSelectInput("var2", "Variable Y", data = airbnbR, selected = "price")
                   checkboxInput("var2log", "Log Y"),
                   checkboxInput("ols", "OLS Line")
                 ),
                 mainPanel(
                   plotOutput("scatter")
                 )
               )
      ),
      tabPanel(
        "Price/Subway Map",
        sidebarLayout(
          sidebarPanel(
            selectInput("var",
                        label = "Airbnb or Subway?",
                        choices = list("Airbnb", "Subway"),
                        selected = "Airbnb")
          ),
          mainPanel(
            leafletOutput("PriceMap")
          )
        )
      ),
      tabPanel(
        "Price/Distance Relationship",
        sidebarLayout(
```

```
                    sidebarPanel(
                      checkboxInput("logx", "Log the Distance Variable"),
                      checkboxInput("logy", "Log the Price Variable"),
                      tableOutput("lmt"),
                      tableOutput("minT"),
                      tableOutput("maxT")
                    ),
                    mainPanel(
                      plotOutput("PDplot")
                    )
                  )
                ),
                tabPanel("Rental Finder",
                         div(class="outer",
                             leafletOutput("map"),
                             absolutePanel(
                               column(3, checkboxGroupInput("borough","Neighborhood:",
                                                            choices = borough,
                                                            selected = borough),
                                      checkboxGroupInput("room_type","Room Type:",
                                                         choices = room_type,
                                                         selected = room_type),
                                      sliderInput("price", "Budget:", min = 0,
                                                  max = 10000, value = c(0, 3000), step = 50),
                                      sliderInput("review", "Number of Reviews:", min = 0,
                                                  max = 629, value = c(0,200), step = 10)),
                               column(5, plotOutput("plot1")),
                               column(4, plotOutput("plot2")))
                         )
                ),
                tabPanel("Reference", textOutput("ref1"),
                         textOutput("ref2"),
                         textOutput("ref3")
                         )
                )
              )


server <- function(input, output, session) {
  output$hist <- renderPlot({
    pl <- ggplot(airbnbR, aes(x = !!input$univar)) +
      theme_bw()

    if (is.numeric(airbnbR[[input$univar]])) {
      pl <- pl + geom_histogram(bins = input$unibins, fill = "light blue", color = "black")
      if (input$unilog) {
        pl <- pl + scale_x_log10()
      }
    } else {
      pl <- pl + geom_bar(fill = "light blue", color = "black")
    }

    pl
```

```
})

output$unitest_results <- renderTable({
  if (input$unilog & is.numeric(airbnbR[[input$univar]])) {
    airbnbR %>%
      mutate(logvar = log2(!!input$univar + 0.5)) -> temp
    t.test(temp[["logvar"]], mu = input$uninull) %>%
      tidy() %>%
      select(`P-value` = p.value,
             Lower = conf.low,
             Upper = conf.high)
  } else if (is.numeric(airbnbR[[input$univar]])) {
    t.test(airbnbR[[input$univar]], mu = input$uninull) %>%
      tidy() %>%
      select(`P-value` = p.value,
             Lower = conf.low,
             Upper = conf.high)
  } else {
    "Not a numeric"
  }
})

output$scatter <- renderPlot({
  airbnbR %>%
    ggplot(aes(x = !!input$var1, y = !!input$var2)) +
    theme_bw() ->
    pl
  if (is.numeric(airbnbR[[input$var1]]) & is.numeric(airbnbR[[input$var2]])) {
    pl <- pl + geom_point(color = "#FF5A5F")
  } else if (!is.numeric(airbnbR[[input$var1]]) & is.numeric(airbnbR[[input$var2]])) {
    pl <- pl + geom_boxplot(fill = "#00A699")
  } else if (is.numeric(airbnbR[[input$var1]]) & !is.numeric(airbnbR[[input$var2]])) {
    pl <- pl + geom_boxploth(fill = "#00A699")
  } else {
    pl <- pl + geom_jitter()
  }

  if (input$var1log & is.numeric(airbnbR[[input$var1]])) {
    pl <- pl + scale_x_log10()
  }

  if (input$var2log & is.numeric(airbnbR[[input$var2]])) {
    pl <- pl + scale_y_log10()
  }

  if (input$ols & is.numeric(airbnbR[[input$var1]]) & is.numeric(airbnbR[[input$var2]])) {
    pl <- pl + geom_smooth(se = FALSE, method = "lm")
  }

  pl
})

output$dt <- renderDataTable({
```

```
  airbnbd
},
options = list(pageLength = 10)
)

output$PriceMap <- renderLeaflet({
  if (input$var == "Airbnb") {
    leaflet(airbnb_price) %>%
      addTiles() %>%
      setView(-74.00, 40.71, zoom = 12)%>%
      addMarkers(clusterOptions = markerClusterOptions(),
                 popup = ~paste("-Listing: ", name,
                                "-Subway Distance (miles): ", near_sub,
                                sep = "<br/>"),
                 label = ~paste("Price: $",price))
  } else if (input$var == "Subway") {
    leaflet(nysub) %>%
      addTiles() %>%
      setView(-74.00, 40.71, zoom = 12)%>%
      addMarkers(clusterOptions = markerClusterOptions(),
                 label = ~as.character(str_c("Subway Station: ", NAME)),
                 popup = ~as.character(str_c("Subway Line: ", LINE)))
  }
})

output$lmt <- renderTable({
  if (input$logx == TRUE & input$logy == FALSE) {
    newlm <- lm(price ~ log(near_sub), airbnb_price)
  } else if (input$logx == FALSE & input$logy == TRUE) {
    newlm <- lm(log(price + 1 - min(price)) ~ near_sub, airbnb_price)
  } else if (input$logx == TRUE & input$logy == TRUE) {
    newlm <- lm(log(price + 1 - min(price)) ~ log(near_sub), airbnb_price)
  } else {
    newlm <- lm(price ~ near_sub, airbnb_price)
  }
  tidy(newlm, conf.int = TRUE) %>%
    select(term, estimate, p.value)
})

output$minT <- renderTable({
  airbnb_price %>%
    select("Min Price" = price) %>%
    arrange(`Min Price`) %>%
    head(n = 5) -> c1
  airbnb_price %>%
    select("Min Distance" = near_sub) %>%
    arrange(`Min Distance`) %>%
    head(n = 5) -> c2
  airbnb_price %>%
    mutate(pricel = log10(price + 1 - min(price))) %>%
    select("Min Price" = pricel) %>%
    arrange(`Min Price`) %>%
    head(n = 5) -> c3
```

```
  airbnb_price %>%
    mutate(near_subl = log10(near_sub)) %>%
    select("Min Distance" = near_subl) %>%
    arrange(`Min Distance`) %>%
    head(n = 5) -> c4

  if (input$logx == TRUE & input$logy == FALSE) {
    bind_cols(c4, c1)
  } else if (input$logx == FALSE & input$logy == TRUE) {
    bind_cols(c2, c3)
  } else if (input$logx == TRUE & input$logy == TRUE) {
    bind_cols(c4, c3)
  } else {
    bind_cols(c2, c1)
  }
})

output$maxT <- renderTable({
  airbnb_price %>%
    select("Max Price" = price) %>%
    arrange(-`Max Price`) %>%
    head(n = 5) -> c_1
  airbnb_price %>%
    select("Max Distance" = near_sub) %>%
    arrange(-`Max Distance`) %>%
    head(n = 5) -> c_2
  airbnb_price %>%
    mutate(pricel = log10(price + 1 - min(price))) %>%
    select("Max Price" = pricel) %>%
    arrange(-`Max Price`) %>%
    head(n = 5) -> c_3
  airbnb_price %>%
    mutate(near_subl = log10(near_sub)) %>%
    select("Max Distance" = near_subl) %>%
    arrange(-`Max Distance`) %>%
    head(n = 5) -> c_4

  if (input$logx == TRUE & input$logy == FALSE) {
    bind_cols(c_4, c_1)
  } else if (input$logx == FALSE & input$logy == TRUE) {
    bind_cols(c_2, c_3)
  } else if (input$logx == TRUE & input$logy == TRUE) {
    bind_cols(c_4, c_3)
  } else {
    bind_cols(c_2, c_1)
  }
})

output$PDplot <- renderPlot({
  airbnb_price %>%
    ggplot(aes(x=near_sub, y=price)) +
    geom_smooth(method = "lm", se = FALSE, color = "black") +
    geom_point(aes(color = neighbourhood_group)) +
```

```
    ylab("Rental Price (USD)") +
    xlab("Distance from Nearest Subway Station (Miles)") +
    labs(color = "Neighborhood") +
    theme_bw() -> pl

  airbnb_price %>%
    ggplot(aes(x=near_sub, y=log10(price + 1 - min(price)))) +
    geom_smooth(method = "lm", se = FALSE, color = "black") +
    geom_point(aes(color = neighbourhood_group)) +
    ylab("Rental Price (USD)") +
    xlab("Distance from Nearest Subway Station (Miles)") +
    labs(color = "Neighborhood") +
    theme_bw() -> nl

  airbnb_price %>%
    ggplot(aes(x=log10(near_sub), y=price)) +
    geom_smooth(method = "lm", se = FALSE, color = "black") +
    geom_point(aes(color = neighbourhood_group)) +
    ylab("Rental Price (USD)") +
    xlab("Distance from Nearest Subway Station (Miles)") +
    labs(color = "Neighborhood") +
    theme_bw() -> ol

  airbnb_price %>%
    ggplot(aes(x=log10(near_sub), y=log10(price + 1 - min(price)))) +
    geom_smooth(method = "lm", se = FALSE, color = "black") +
    geom_point(aes(color = neighbourhood_group)) +
    ylab("Rental Price (USD)") +
    xlab("Distance from Nearest Subway Station (Miles)") +
    labs(color = "Neighborhood") +
    theme_bw() -> ql

  if (input$logx == TRUE & input$logy == FALSE) {
    ol
  } else if (input$logx == FALSE & input$logy == TRUE) {
    nl
  } else if (input$logx == TRUE & input$logy == TRUE) {
    ql
  } else {
    pl
  }
})

mapdata <- reactive({
  airbnb %>%
    filter(borough %in% input$borough,
           room_type %in% input$room_type,
           price >= input$price[1],
           price<= input$price[2],
           number_of_reviews >=input$review[1],
           number_of_reviews <=input$review[2])
})
```

```
  output$map <- renderLeaflet({
    leaflet(mapdata()) %>%
      setView(lng = -73.94197, lat = 40.73638, zoom = 12) %>%
      addProviderTiles(providers$CartoDB.Positron) %>%
      addTiles()%>%
      addMarkers(clusterOptions = markerClusterOptions(),
                 popup = ~paste("Neighborhood:", borough,
                                "Room Type:", room_type,
                                "Budget:", price,
                                "Number of Reviews:", number_of_reviews,
                                sep = "<br/>"))
  })

  output$plot1 <- renderPlot({
    mapdata()%>%
      ggplot(aes(x = borough, y = price)) +
      geom_boxplot(fill = "#FF5A5F") +
      theme_bw() +
      xlab("Neighborhood") +
      ylab("Price") +
      scale_y_log10()
  })

  output$plot2 <- renderPlot({
    mapdata()%>%
      ggplot(aes(x = room_type, y = price)) +
      geom_boxplot(fill = "#00A699") +
      theme_bw() +
      xlab("Room Type") +
      ylab("Price") +
      scale_y_log10()
  })

  output$ref1 <- renderText("https://nycdatascience.com/blog/student-works/how-airbnb-is-in-nyc-interac
  output$ref2 <- renderText("https://rstudio.github.io/leaflet/markers.html")
  output$ref3 <- renderText("https://usbrandcolors.com/airbnb-colors/")
}

shinyApp(ui, server)
```