Airbnb Data Visualization

AUTHOR
Huiting Wu, Sharmeen Kapoorwala

PUBLISHED
December 8, 2025

```r
library(dplyr)
library(tidyr)
library(tidytext)
library(ggplot2)
library(wordcloud)
library(ggwordcloud)
library(plotly)
library(lubridate)
library(leaflet)
library(tidygeocoder)
library(scales)
library(hexbin)
library(viridis)
```

Read in the data & initial check

```r
# read the data
airbnb <- read.csv("Airbnb_Open_Data.csv")

# factor all the characters
airbnb <- airbnb |> mutate(across(where(is.character), as.factor))
```

```r
# checking missing values
colSums(is.na(airbnb))
```

```
                             id                            NAME 
                              0                               0 
                        host.id          host_identity_verified 
                              0                               0 
                      host.name             neighbourhood.group 
                              0                               0 
                  neighbourhood                             lat 
                              0                               8 
                           long                         country 
                              8                               0 
                   country.code                instant_bookable 
                              0                             105 
            cancellation_policy                       room.type 
                              0                               0 
              Construction.year                           price 
                            214                               0 
                    service.fee                  minimum.nights 
                              0                             409 
```

```
             number.of.reviews                    last.review 
                           183                              0 
             reviews.per.month             review.rate.number 
                         15879                            326 
calculated.host.listings.count               availability.365 
                           319                            448 
                   house_rules                        license 
                             0                              0 
```

```r
sum(duplicated(airbnb)) # there are 541 duplicated data
```

```
[1] 541
```

```r
airbnb <- airbnb[-which(duplicated(airbnb)), ] # remove the duplicated rows
dim(airbnb) # there are 102058 rows and 26 variables
```

```
[1] 102058     26
```

Tidy the dataset

```r
levels(airbnb$neighbourhood.group)
```

```
[1] ""              "Bronx"         "brookln"       "Brooklyn"     
[5] "manhatan"      "Manhattan"     "Queens"        "Staten Island"
```

```r
# rename the levels of neighbourhood group
airbnb <- airbnb |>
  mutate(
    neighbourhood.group = case_when(
      neighbourhood.group %in% c("brookln") ~ "Brooklyn",
      neighbourhood.group %in% c("manhatan") ~ "Manhattan",
      TRUE ~ neighbourhood.group
    ),
    neighbourhood.group = factor(neighbourhood.group)
  )
levels(airbnb$neighbourhood.group)
```

```
[1] ""              "Bronx"         "Brooklyn"      "Manhattan"    
[5] "Queens"        "Staten Island"
```

```r
# The neighbourhoods with neighborhood group empty
airbnb |>
  filter(neighbourhood.group == "") |>
  pull(neighbourhood)
```

```
 [1] Washington Heights Clinton Hill       East Village       Upper East Side   
 [5] Woodside           Williamsburg       Bushwick           Prospect Heights  
 [9] East Village       Williamsburg       Clinton Hill       Chelsea
[13] Prospect Heights   East Harlem        Bushwick           Eastchester
[17] Williamsburg       Harlem             Chinatown          Williamsburg
[21] Queens Village     Harlem             Williamsburg       Bedford-
Stuyvesant
[25] East Village       East Harlem        Harlem             Bushwick
[29] Upper West Side
225 Levels:  Allerton Arden Heights Arrochar Arverne Astoria ... Woodside
```

```r
# group those area into the a county
lookup <- tibble::tribble(
  ~neighbourhood,       ~borough,
  "Washington Heights", "Manhattan",
  "Clinton Hill",       "Brooklyn",
  "East Village",       "Manhattan",
  "Upper East Side",    "Manhattan",
  "Woodside",           "Queens",
  "Williamsburg",       "Brooklyn",
  "Bushwick",           "Brooklyn",
  "Prospect Heights",   "Brooklyn",
  "Chelsea",            "Manhattan",
  "East Harlem",        "Manhattan",
  "Eastchester",        "Bronx",
  "Harlem",             "Manhattan",
  "Chinatown",          "Manhattan",
  "Queens Village",     "Queens",
  "Bedford-Stuyvesant", "Brooklyn",
  "Upper West Side",    "Manhattan"
)
```

```r
airbnb <- airbnb |>
  left_join(lookup, by = "neighbourhood") |>
  mutate(
    neighbourhood.group = if_else(
      neighbourhood.group == "",
      borough,                 # fill with correct borough
      neighbourhood.group      # keep existing value
    ),
    neighbourhood.group = as.factor(neighbourhood.group)
  ) |>
  select(-borough)
levels(airbnb$neighbourhood.group)
```

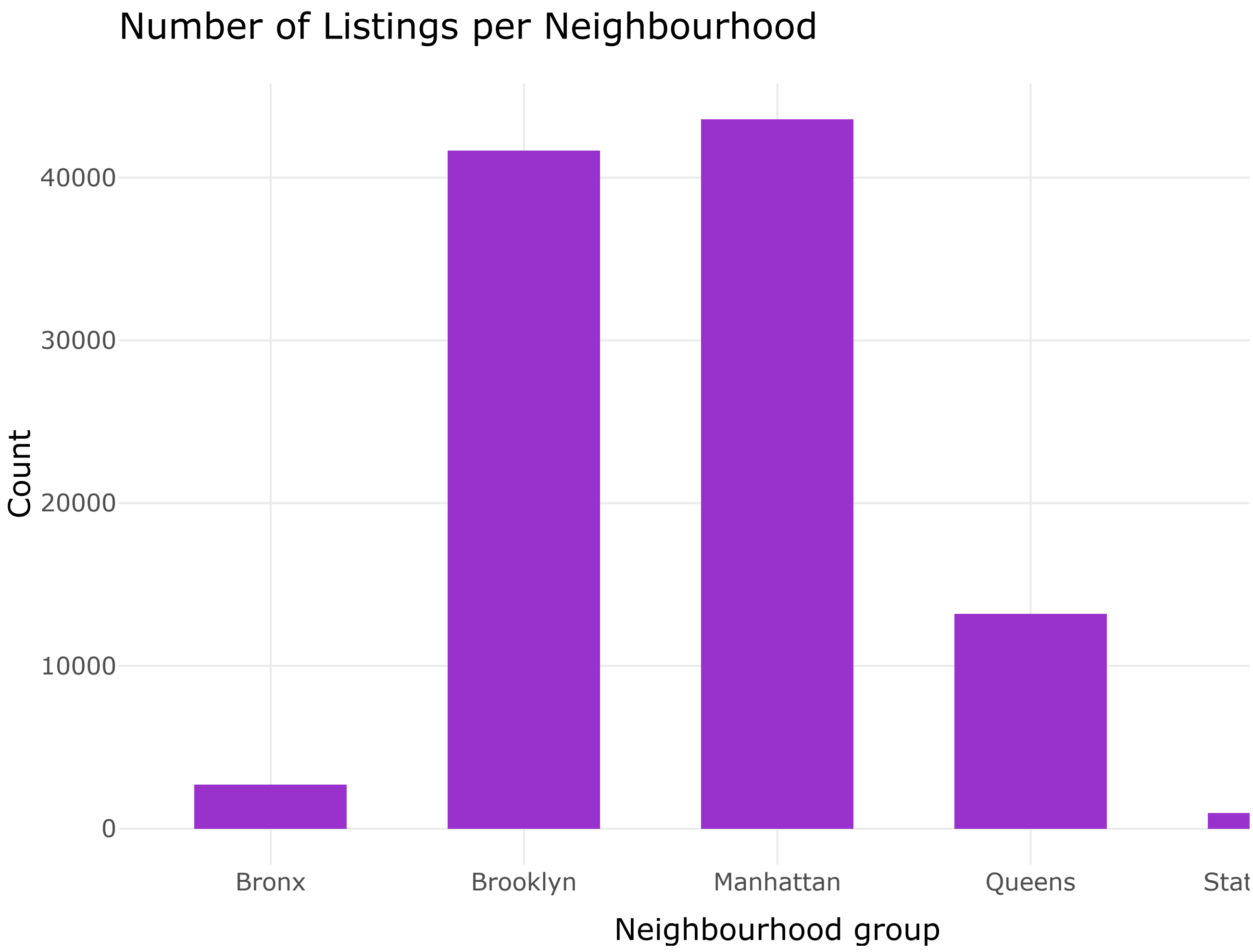
```
[1] "Bronx"         "Brooklyn"      "Manhattan"     "Queens"
[5] "Staten Island"
```

```r
# remove the $ in the price variable
airbnb$price <- gsub("\\$", "", airbnb$price)
airbnb$price <-  as.numeric(airbnb$price)
```

Visualizations

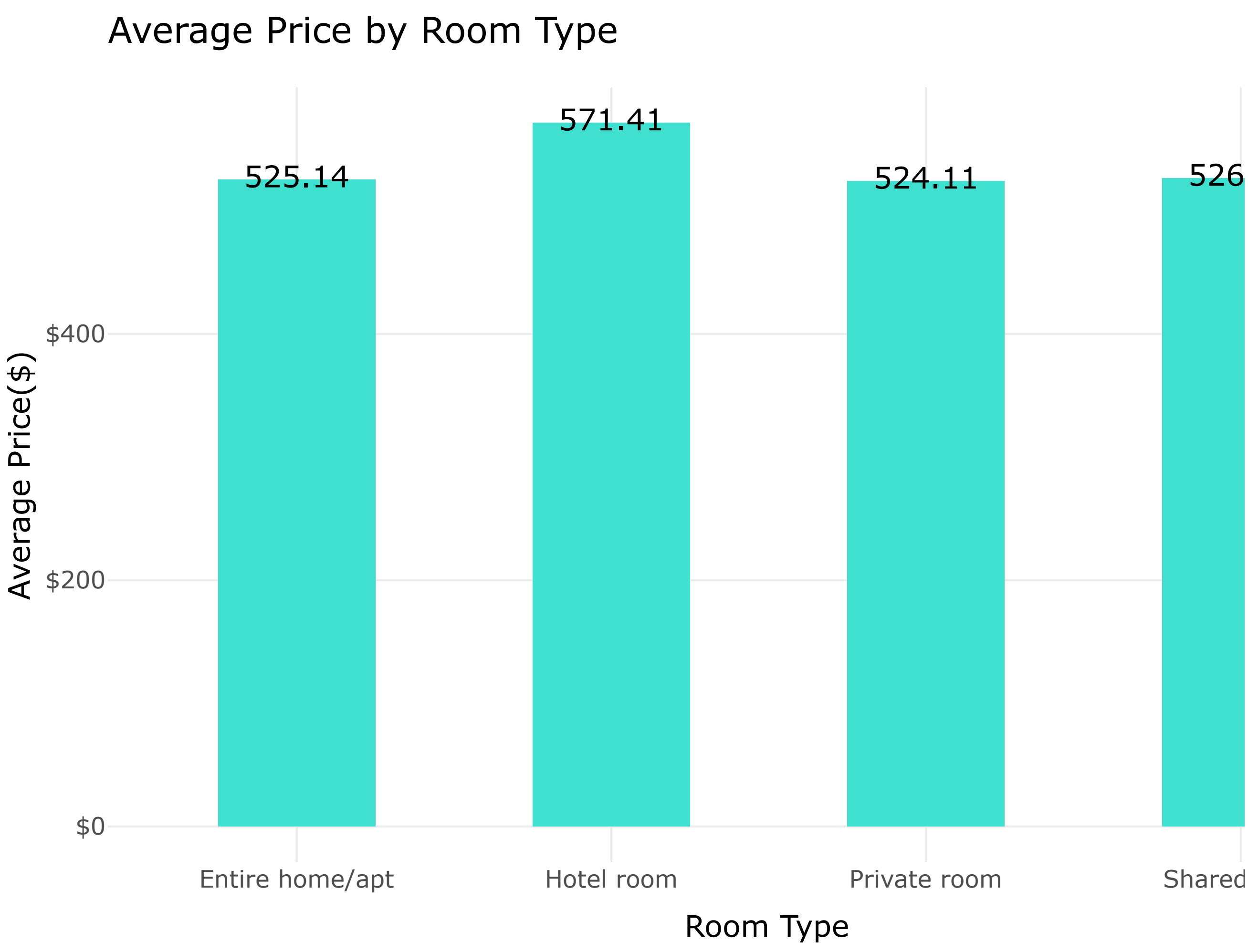
Count of Listings by Neighborhood Group

```
neighbour_list <- ggplot(airbnb, aes(x = neighbourhood.group))+
  geom_bar(fill = "darkorchid", width = 0.6)+
  labs(
    title = "Number of Listings per Neighbourhood",
    x = "Neighbourhood group",
    y = "Count"
  )+
  theme_minimal()

ggplotly(neighbour_list)
```



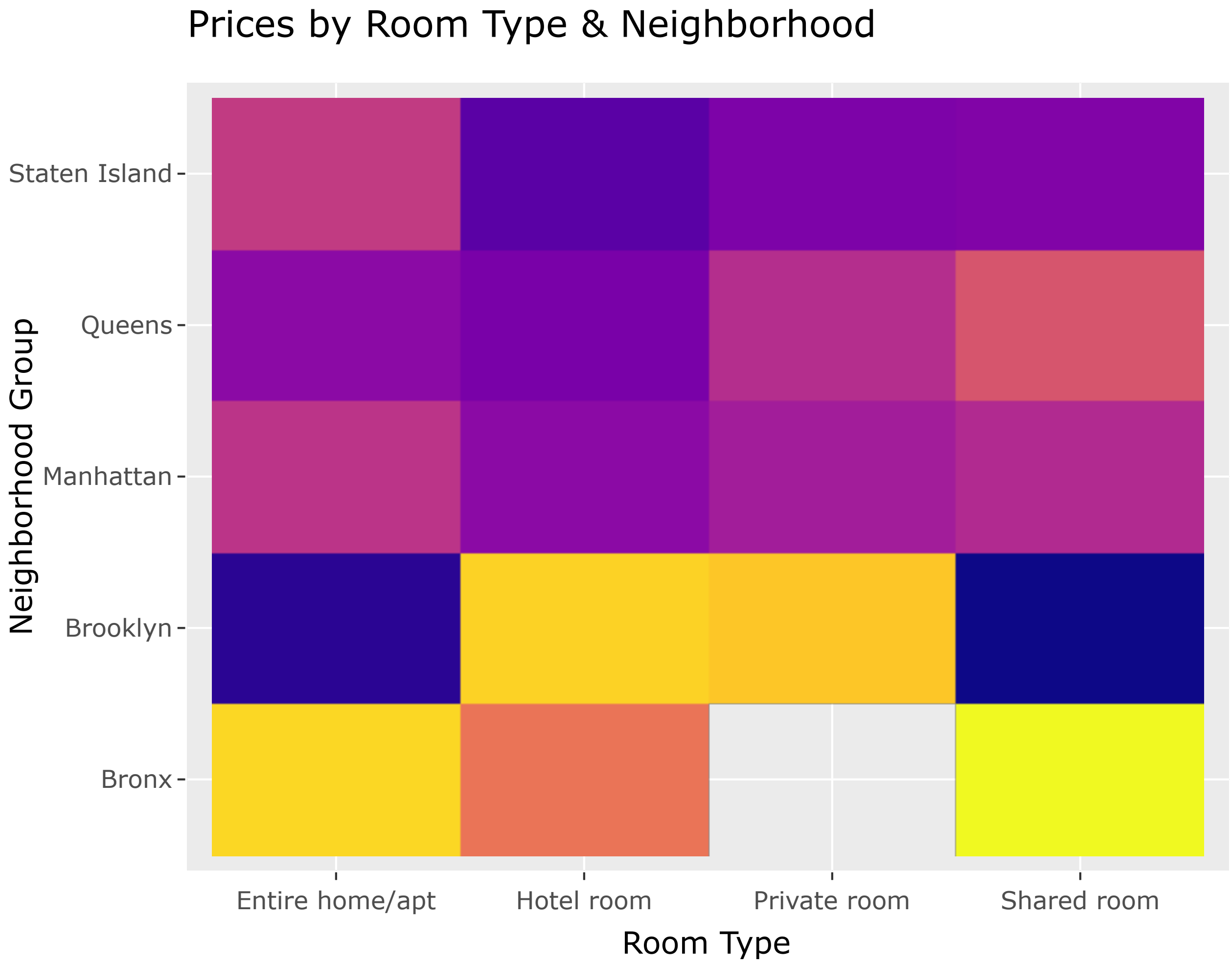
Average Price by Room Type

```
price_per_room <- airbnb |>
  group_by(room.type) |>
  summarise(avg_price= mean(price, na.rm = T))

price_roomtype <- ggplot(price_per_room, aes(x = room.type, y = avg_price)) +
  geom_bar(stat = "identity", fill = "turquoise", width = 0.5)+
  geom_text(aes(label = round(avg_price,2)))+
  scale_y_continuous(labels = dollar_format())+
  labs(
    title = "Average Price by Room Type",
    x = "Room Type",
    y = "Average Price($)"
  ) +
  theme_minimal()

ggplotly(price_roomtype)
```



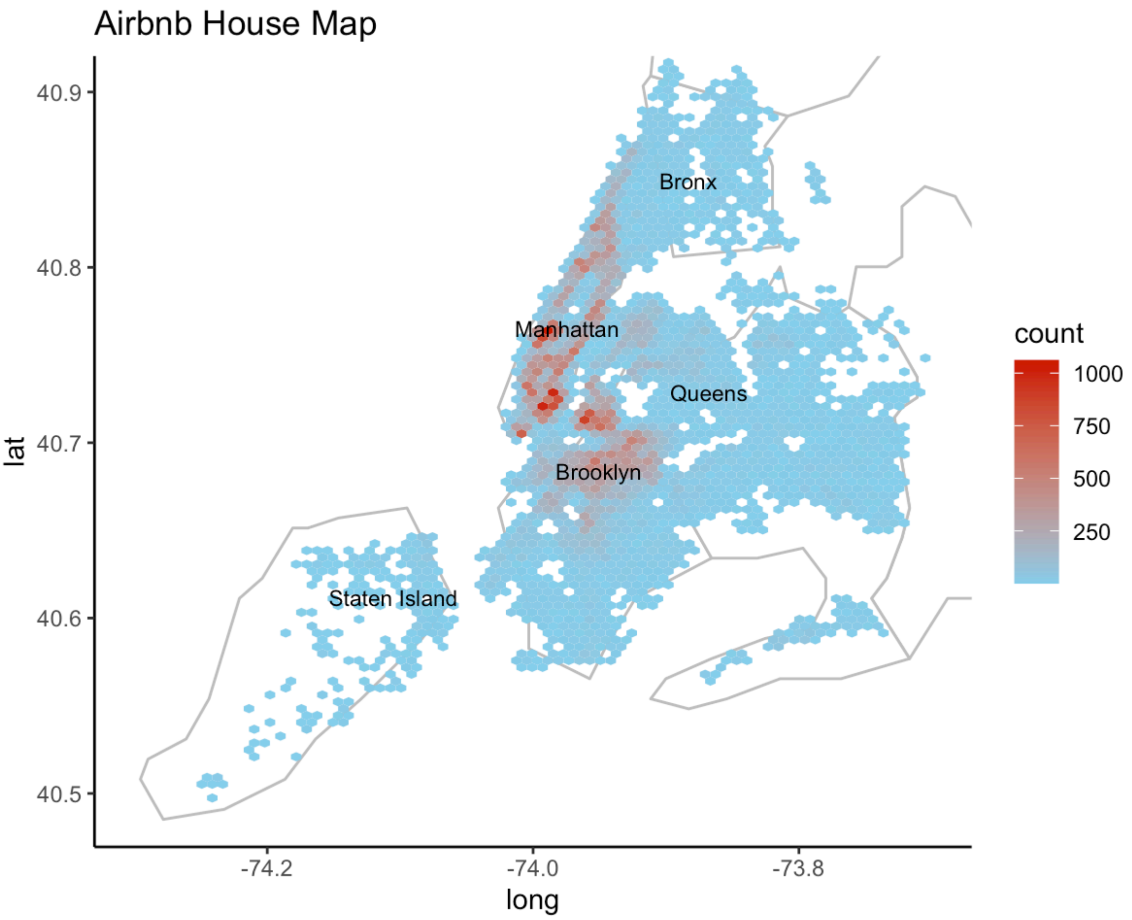
Listings by Price and Room Type

```
list_room_price <- ggplot(airbnb, aes(x = room.type,
                                      y = neighbourhood.group,
                                      fill= price
    )) +
  geom_tile()+
  scale_fill_viridis(option = "plasma",
                     direction = -1,
                     )+
  labs(
    title = "Prices by Room Type & Neighborhood",
    x = "Room Type",
    y = "Neighborhood Group",
    fill = "Price ($)"
  )

ggplotly(list_room_price, tooltip = "text")
```



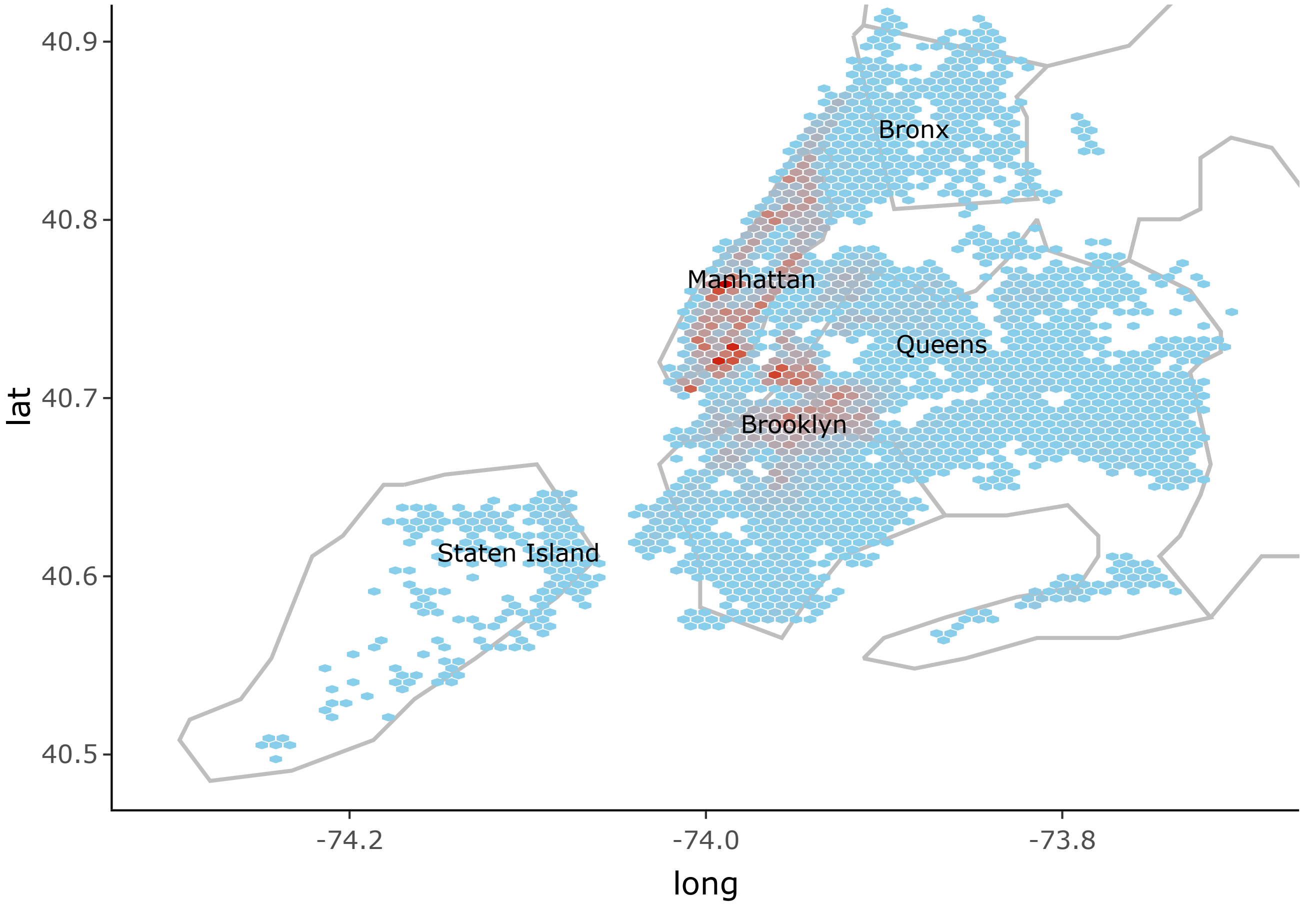
Map

```
nyc <- map_data("county") |> filter(region == "new york")
label_df <- airbnb |>
  group_by(neighbourhood.group)  |>
  summarise(
    long = mean(long, na.rm = TRUE),
    lat  = mean(lat, na.rm = TRUE)
  )

label_df <- airbnb |>
  group_by(neighbourhood.group)  |>
  summarise(
    long = mean(long, na.rm = TRUE),
    lat  = mean(lat, na.rm = TRUE),
    n_listings = n(),
    avg_price = mean(price, na.rm = TRUE)
  )

map <- ggplot() +
  geom_polygon(data = nyc, aes(x = long, y = lat, group = group),
               fill = "white", color = "gray") +
  coord_quickmap(xlim = c(-74.3, -73.7), ylim = c(40.49,40.9)) +
  geom_hex(data = airbnb, aes(x = long, y = lat), bins = 1000) +
  scale_fill_gradient(low = "skyblue", high = "red3") +
  geom_text(data = label_df,
            aes(x = long, y = lat,
                label = neighbourhood.group,
                text = paste(
        "Total listings:", n_listings,
        "\n Avg price: $", round(avg_price, 1)
      )),
            color = "black",
            size = 3) +
  labs(title = "Airbnb House Map") +
  theme_classic()

map
```



```
ggplotly(map, tooltip = "text")
```

Airbnb House Map



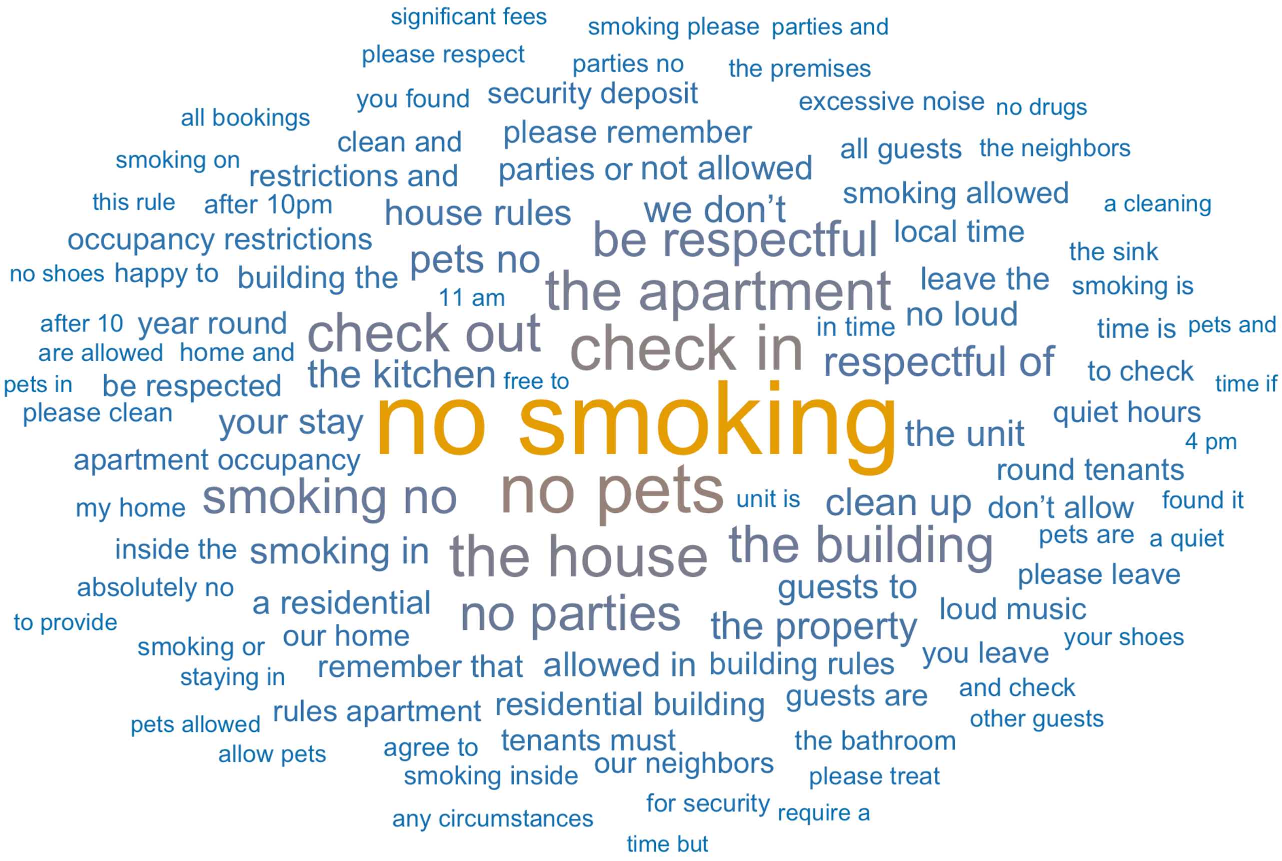
Word Cloud

```
airbnb <- airbnb |>
  mutate(house_rules = as.character(house_rules))

# word frequency table
word_freq <- airbnb |>
  filter(house_rules != "") |> # remove missing house rules
  unnest_tokens(bigram, house_rules, token = "ngrams", n = 2) |>
  # lowercasing all; token the sentences into bigram(2 words for a phase)
  separate(bigram, c("word1", "word2"), sep = " ")  |> # separate the bigram
  filter(
    !is.na(word1), !is.na(word2), # remove NA
    !word1 %in% stop_words$word | !word2 %in% stop_words$word
    # remove the bigram with both words are stop words
  )  |>
  unite(bigram, word1, word2, sep = " ")  |>  # recombine the bigram
  count(bigram, sort = TRUE)
head(word_freq, 10)
```

```
          bigram     n
1     no smoking 26067
2        no pets 11741
3       check in 10188
4      the house  8626
5  the apartment  7754
```

```
6       check out  7395
7    the building  6653
8      no parties  6311
9   be respectful  6142
10     smoking no  6137
```

```
word_freq |> filter(n > 1300) |>
  ggplot(aes(label = bigram, size = n, color = n)) +
  geom_text_wordcloud() +
  scale_size_area(max_size = 20) +
  scale_color_gradient(low = "#0072B2", high = "#E69F00") +
    theme_minimal()
```


significant fees
smoking please
parties and
please respect
parties no
the premises
you found
security deposit
excessive noise
no drugs
all bookings
clean and
please remember
all guests
the neighbors
smoking on
restrictions and
parties or not allowed
smoking allowed
this rule
after 10pm
house rules
we don’t
a cleaning
occupancy restrictions
be respectful
local time
the sink
no shoes
happy to
building the
pets no
the apartment
leave the
smoking is
11 am
after 10
year round
check out
check in
in time
no loud
time is
pets and
are allowed
home and
respectful of
the kitchen
free to
pets in
be respected
to check
time if
please clean
no smoking
your stay
the unit
quiet hours
4 pm
apartment occupancy
round tenants
my home
smoking no
no pets
unit is
clean up
don’t allow
found it
inside the
smoking in
the house
the building
pets are
a quiet
absolutely no
guests to
please leave
a residential
no parties
to provide
loud music
the property
our home
smoking or
your shoes
you leave
remember that
allowed in
building rules
staying in
and check
residential building
guests are
rules apartment
other guests
pets allowed
tenants must
the bathroom
agree to
allow pets
our neighbors
smoking inside
please treat
for security
require a
any circumstances
time but