Case Study: New York Airbnb 2023

Jinxiu

2025-11-23

Phase 1: Business Understanding

Background

New York City is one of Airbnb's largest and most supply-constrained markets, where location, pricing, and housing density vary significantly across neighborhoods. With gradually intensive competition for rental housing in New York, Airbnb occurs phenomena such as insufficient housing resource, rising rent, and the impact of short-term rental demands on long-term supply. Other Key stakeholders, such as house and apartment owners who care more about rent and occupancy. And regulatory institutions focus more on legal leasing and regional housing density.

Purpose of Analysis

- From the perspective of **owners**: Which pricing patterns are associated with higher occupancy?
- From the perspective of **Airbnb**: Which neighborhoods signs of supply shortage or unmet demand?
- From the perspective of **city**: Which neighborhood has highest density of short-term/long-term housing? Why does rental density matter for the city?

Business task

Organizing these considerations above, I propose the following questions:

- Which neighborhoods show significantly higher median listing prices?
- Is review volume correlated with availability or estimated occupancy?
- Which neighborhood deliver the highest price per listing or highest occupancy potential?

Phase 2: Data Structure Understanding

This section is to be quickly familiar with data, data type, and data quality. Dataset is publicly available and can be downloaded from Kaggle: https://www.kaggle.com/datasets/godofoutcasts/new-york-city-airbnb-2023-public-data/data

2.1 Data Schema Overview

```r
# load required package
library(readr)
library(skimr)
library(here)
```

```
## here() starts at D:/CaseStudy_Airbnb_NYC_2023/r_report
```

```r
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.4     v purrr     1.2.0
## v forcats   1.0.1     v stringr   1.6.0
## v ggplot2   4.0.1     v tibble    3.3.0
## v lubridate 1.9.4     v tidyr     1.3.1
```

```
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become
```

```r
library(dplyr)
library(stringr)
library(lubridate)
library(ggplot2)
library(leaflet)
library(maps)
```

```
##
## Attaching package: 'maps'
##
## The following object is masked from 'package:purrr':
##
##     map
```

```r
library(scales)
```

```
##
```

```
## Attaching package: 'scales'
##
## The following object is masked from 'package:purrr':
##
##     discard
##
## The following object is masked from 'package:readr':
##
##     col_factor
```

```r
# import data
nyc <- read_csv('../raw_data/NYC-Airbnb-2023.csv')
```

```
## Warning: One or more parsing issues, call `problems()` on your data frame for details,
## e.g.:
##   dat <- vroom(...)
##   problems(dat)

## Rows: 42931 Columns: 18
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (6): name, host_name, neighbourhood_group, neighbourhood, room_type, la...
## dbl (11): id, host_id, latitude, longitude, price, minimum_nights, number_of...
## lgl  (1): license
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```r
# check data schema
glimpse(nyc)
```

```
## Rows: 42,931
## Columns: 18
## $ id                             <dbl> 2595, 5121, 5203, 5178, 5136, 29628, 55~
## $ name                           <chr> "Skylit Midtown Castle", "BlissArtsSpac~
## $ host_id                        <dbl> 2845, 7356, 7490, 8967, 7378, 127608, 8~
## $ host_name                      <chr> "Jennifer", "Garon", "MaryEllen", "Shun~
## $ neighbourhood_group            <chr> "Manhattan", "Brooklyn", "Manhattan", "~
## $ neighbourhood                  <chr> "Midtown", "Bedford-Stuyvesant", "Upper~
## $ latitude                       <dbl> 40.75356, 40.68535, 40.80380, 40.76457,~
## $ longitude                      <dbl> -73.98559, -73.95512, -73.96751, -73.98~
## $ room_type                      <chr> "Entire home/apt", "Private room", "Pri~
## $ price                          <dbl> 150, 60, 75, 68, 275, 93, 295, 124, 200~
## $ minimum_nights                 <dbl> 30, 30, 2, 2, 60, 3, 4, 3, 1, 30, 30, 2~
## $ number_of_reviews              <dbl> 49, 50, 118, 575, 3, 350, 45, 223, 68, ~
## $ last_review                    <chr> "6/21/2022", "12/2/2019", "7/21/2017", ~
## $ reviews_per_month              <dbl> 0.30, 0.30, 0.72, 3.41, 0.03, 2.25, 0.2~
## $ calculated_host_listings_count <dbl> 3, 2, 1, 1, 1, 1, 1, 3, 4, 1, 2, 1, 1, ~
## $ availability_365               <dbl> 314, 365, 0, 106, 181, 145, 1, 164, 310~
## $ number_of_reviews_ltm          <dbl> 1, 0, 0, 52, 1, 48, 4, 17, 0, 5, 1, 9, ~
## $ license                        <lgl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA,~
```

```
# check descriptive information of fields
summary(nyc)
```

```
##        id                name               host_id            host_name
##  Min.   :2.595e+03   Length:42931       Min.   :     1678   Length:42931
##  1st Qu.:1.940e+07   Class :character   1st Qu.: 16085328   Class :character
##  Median :4.337e+07   Mode  :character   Median : 74338125   Mode  :character
##  Mean   :2.223e+17                      Mean   :151601209
##  3rd Qu.:6.305e+17                      3rd Qu.:268069240
##  Max.   :8.405e+17                      Max.   :503872891
##
##  neighbourhood_group neighbourhood         latitude       longitude
##  Length:42931        Length:42931       Min.   :40.50   Min.   :-74.25
##  Class :character    Class :character   1st Qu.:40.69   1st Qu.:-73.98
##  Mode  :character    Mode  :character   Median :40.72   Median :-73.95
##                                         Mean   :40.73   Mean   :-73.94
##                                         3rd Qu.:40.76   3rd Qu.:-73.92
##                                         Max.   :40.91   Max.   :-73.71
##
##   room_type            price         minimum_nights    number_of_reviews
##  Length:42931       Min.   :    0.0   Min.   :   1.00   Min.   :   0.00
##  Class :character   1st Qu.:   75.0   1st Qu.:   2.00   1st Qu.:   1.00
##  Mode  :character   Median :  125.0   Median :   7.00   Median :   5.00
##                     Mean   :  200.3   Mean   :  18.11   Mean   :  25.86
##                     3rd Qu.:  200.0   3rd Qu.:  30.00   3rd Qu.:  24.00
##                     Max.   :99000.0   Max.   :1250.00   Max.   :1842.00
##
##  last_review        reviews_per_month calculated_host_listings_count
##  Length:42931       Min.   : 0.010    Min.   :  1.00  
##  Class :character   1st Qu.: 0.140    1st Qu.:  1.00  
##  Mode  :character   Median : 0.520    Median :  1.00  
##                     Mean   : 1.169    Mean   : 24.05  
##                     3rd Qu.: 1.670    3rd Qu.:  4.00  
##                     Max.   :86.610    Max.   :526.00  
##                     NA's   :10304                     
##  availability_365 number_of_reviews_ltm license       
##  Min.   :  0.0    Min.   :   0.000      Mode:logical  
##  1st Qu.:  0.0    1st Qu.:   0.000      NA's:42931    
##  Median : 89.0    Median :   0.000                    
##  Mean   :140.3    Mean   :   7.737                    
##  3rd Qu.:289.0    3rd Qu.:   7.000                    
##  Max.   :365.0    Max.   :1093.000                    
## 
```

The **nyc** dataset consists of 42931 rows and 18 columns, covering listing information such as host id, room type, location (latitude and longitude), price, availability, and number of reviews.
Key categorical fields include *neighbourhood_group*, *neighbourhood*, and *room_type*. Key numerical fields include *price*, *number_of_reviews*, and *availability_365*.

2.2 Missing Value Check

```
# check columns with NA or NULL
colSums(is.na(nyc))
```

```
##                             id                           name 
##                              0                             10 
##                        host_id                      host_name 
##                              0                              5 
##            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 
##                          10304                          10304
## calculated_host_listings_count               availability_365
##                              0                              0
##            number_of_reviews_ltm                        license
##                              0                          42931
```

Among 18 variables, *reviews_per_month* contains 10304 missing values, mainly because many listing properties have never get a review from renters. All other fields show no missing values.

2.3 Outliers Check

Based on the output from data schema overview, it can be obeserved that:

Price
Price ranges from 0 to 99000, which indicates abnormal entries.

- 0 likely represents missing or invalid pricing
- Price above 1000 are unrealistic for typical Airbnb listing properties

These values will be handled in the cleaning phase.

Minimum Nights
minimum_nights displays a maximum value of 1250 nights, which is far beyond the practical booking for length of stay and should be treated as outliers.

Geographic Location
Latitude and longitude values fall within the expected New York City boundaries. No invalid coordinates are detected.

Review Metrics
reviews_per_month shows a maximum of 86.6, which is unrealistic for Airbnb listings. This value likely results from inconsistencies in the period of activity and aggregated historical reviews.

Availability
availability_365 ranges from 0 to 365, which is consistent with Airbnb's definition that represents the number of days in the upcoming year during which the properties is available for booking. Lower values indicate higher occupancy or demand, while higher values mean low booking frequency or availability.

Summary

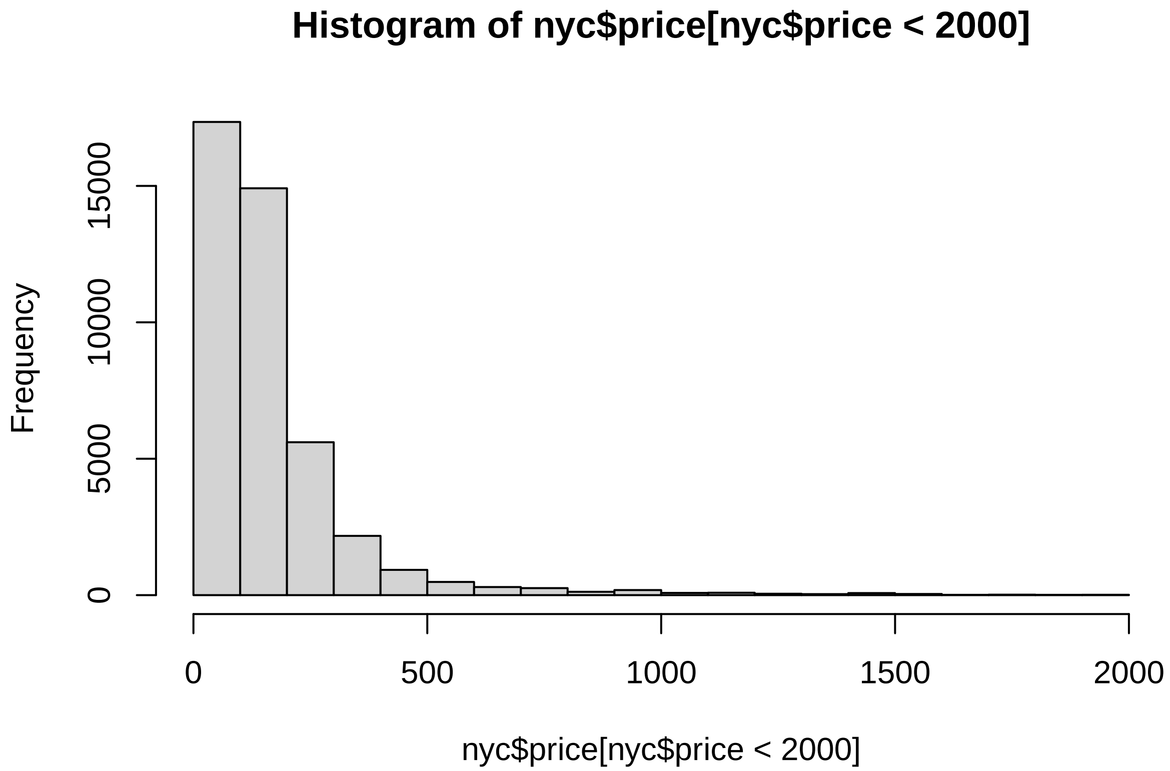
To sum up, the dataset is complete with minimal missing values. However, several fields contain abnormal values, including 0 or extremely high prices, unrealistic minimum night requirements, and overmuch review counts. These issues will be dealt with in the following data cleaning phase to ensure the accuracy of subsequent analysis.

Phase 3: Data Cleaning

```
# step 1: clean price
nyc %>%
  filter(price == 0)
```

```
## # A tibble: 27 x 18
##          id name    host_id host_name neighbourhood_group neighbourhood latitude
##       <dbl> <chr>     <dbl> <chr>     <chr>               <chr>            <dbl>
##  1 40560656 The Ho~  2.73e8 The Hoxt~ Brooklyn            Williamsburg      40.7
##  2 41740615 The Ja~  2.68e8 The Jame~ Manhattan           Midtown           40.7
##  3 41740622 Garden~  2.69e8 Gardens ~ Manhattan           Upper East S~     40.8
##  4 41792753 Mint H~  1.97e8 Mint Hou~ Manhattan           Financial Di~     40.7
##  5 42279171 Leon H~  2.65e8 Leon Hot~ Manhattan           Chinatown         40.7
##  6 42065545 Carvi ~  3.10e8 Carvi Ho~ Manhattan           Midtown           40.8
##  7 42065547 Hotel ~  3.09e8 Hotel Fi~ Manhattan           Hell's Kitch~     40.8
##  8 42065563 Opera ~  3.10e8 Opera Ho~ Bronx               Mott Haven        40.8
##  9 42065564 The Wa~  3.14e8 The Wall~ Manhattan           Financial Di~     40.7
## 10 42228997 Sister~  3.14e8 Sister C~ Manhattan           Lower East S~     40.7
## # i 17 more rows
## # i 11 more variables: longitude <dbl>, room_type <chr>, price <dbl>,
## #   minimum_nights <dbl>, number_of_reviews <dbl>, last_review <chr>,
## #   reviews_per_month <dbl>, calculated_host_listings_count <dbl>,
## #   availability_365 <dbl>, number_of_reviews_ltm <dbl>, license <lgl>
```

```
## check price distribution
hist(nyc$price[nyc$price < 2000])
```



I realize that records with 0 price have Hotel Room as room type and many have Manhattan as neighbourhood group. Meanwhile, the distribution of price shows strongly positive skewness. Therefore, I intend to fill the price with the corresponding median values.

```
# fill price by using median price of various neighbourhood group
nyc <- nyc %>%
  filter(price <= 2000) %>%
  group_by(neighbourhood_group) %>%
  mutate(
    median_price_group = median(price[price > 0], na.rm = TRUE),
    price = if_else(price == 0, median_price_group, price)
  ) %>%
  ungroup() %>%
  select(-median_price_group)
```

```
# step 2: clean minimum nights
## using IQR, Q1, Q3 to filter outlier
nyc <- nyc %>%
  mutate(
    q1 = quantile(minimum_nights, 0.25, na.rm = TRUE),
    q3 = quantile(minimum_nights, 0.75, na.rm = FALSE),
    iqr = q3 - q1,
    lower = q1 - 1.5 * iqr,
    upper = q3 + 1.5 * iqr
  ) %>%
  filter(
    minimum_nights >= lower & minimum_nights <= upper) %>%
  select(-q1, -q3, -iqr, -lower, -upper)
```

```
nyc_clean <- nyc %>%
  # step 3: clean reviews
  mutate(
    reviews_per_month = ifelse(reviews_per_month >= 20, NA, reviews_per_month),
    reviews_per_month = replace_na(reviews_per_month, 0)
  ) %>%
  # step 4: fix dates
  mutate(
        last_review = na_if(last_review, ""),
        last_review = mdy(last_review)
  ) %>%
  # step 5: remove duplicates
  distinct() %>%
  # step 6: remove license
  select(-license)
```

For accuracy and reliable of analysis, several data cleaning steps are applied to correct abnormal values and standardize key variables.

1. Price Correction

Records with invalid pricing are removed or corrected.
Listing properties with extremely high values above 2000 are determined as unrealistic and cleaned accordingly.
For properties priced at 0, group-level median prices by *neighbourhood_group* are used to replace them.

2. Minimum Nights

The distribution of *minimum_nights* displays extreme values up to 1250 nights, which are unlikely for short-term leasing.
Outliers are removed by using the IQR and quantile method to reserve realistic length of stay.

3. Review Count Handling

reviews_per_month contains unrealistic values above 20 and many missing values for listing properties without receiving or organizing reviews.

Values above 20 are set to NA and all missing values are replaced with 0 to indicate no reviews.

4. Date Standardization

last_review includes empty strings and inconsistent formats.

These are converted into valid date objects through *mdy()*, and empty values are identified as missing.

5. Duplicate Removal

Duplicate listing properties, if any, are removed to avoid repeated records.

6. Useless Field Removal

All values of *license* are NA and unimportant for subsequent analysis, therefore the whole column needed to be removed.

After applying all cleaning steps, the dataset contains 41982 valid records.

This cleaned dataset will be used for the following analysis.

Phase 4: Exploratory Data Analysis

4.1 Geographical Dimension

```r
# step 1: neighbourhood group vs price
price_ng <- nyc_clean %>%
  select(price, neighbourhood_group) %>%
  group_by(neighbourhood_group) %>%
  summarise(
    cnt_property = n(),
    min_price = min(price),
    avg_price = mean(price),
    median_price = median(price),
    max_price = max(price)
  )
price_ng
```

```
## # A tibble: 5 x 6
##   neighbourhood_group cnt_property min_price avg_price median_price max_price
##   <chr>                      <int>     <dbl>     <dbl>        <dbl>     <dbl>
## 1 Bronx                       1681        10      112.           87      2000
## 2 Brooklyn                   15817        10      148.          111      2000
## 3 Manhattan                  17231        10      235.          165      2000
## 4 Queens                      6829        10      118.           90      2000
## 5 Staten Island                424        26      130.           99     1200
```

```
# step 2: check density of listing property
## only available in HTML report

# map <- leaflet(nyc_clean) %>%
#   addTiles() %>%
#   addMarkers(
#     lng = ~longitude,
#     lat = ~latitude,
#     clusterOptions = markerClusterOptions()
#   )
# map
```

```
# step 3: review Top 10 popular neighborhoods
neighbour_top10 <- nyc_clean %>%
  group_by(neighbourhood) %>%
  summarise(cnt_property = n()) %>%
  slice_max(cnt_property, n = 10)
neighbour_top10
```

```
## # A tibble: 10 x 2
##    neighbourhood      cnt_property
##    <chr>                     <int>
##  1 Bedford-Stuyvesant         3000
##  2 Williamsburg               2554
##  3 Midtown                    2132
##  4 Harlem                     2045
##  5 Bushwick                   1725
##  6 Upper West Side            1498
##  7 Hell's Kitchen             1496
##  8 Upper East Side            1405
##  9 Crown Heights              1296
## 10 East Village               1118
```

Listing properties of Airbnb show a significant characteristic of spatial clustering. Counting the number of properties based on *neighbourhood_group*, **Brooklyn** and **Manhattan** account for above 70% of the total supply, which are highly competitive areas. Among them, **Bedford-Stuyvesant**, **Williamsburg**, and **Midtown** rank top three with the most intensive housing supply, which highly align with NYC's short-term rentals market in 2023. This indicates that the demands of owners and renters concentrate on main urban areas, while **Staten Island** has the least supply and less active market.

4.2 Room Type Dimension

```
# step 1: room type vs price
price_rt <- nyc_clean %>%
  select(price, room_type) %>%
  group_by(room_type) %>%
  summarise(min_price = min(price),
            avg_price = mean(price),
            median_price = median(price),
            max_price = max(price)
  )
price_rt
```

```
## # A tibble: 4 x 5
##   room_type       min_price avg_price median_price max_price
##   <chr>               <dbl>     <dbl>        <dbl>     <dbl>
## 1 Entire home/apt        10      228.          174      2000
## 2 Hotel room             87      332.          240      1592
## 3 Private room           10      109.           74      1999
## 4 Shared room            15       97            56      2000
```

The entire distribution of price shows highly positive skewness. Most properties are priced as from 50 to 300 dollars. Further stratification by *neighbourhood_group* cab be found that the median price of Manhattan is significantly higher than other regions and Brooklyn is at middle level. Although Staten Island has limited supply, its price is relatively stable.
Stratification by *room_type* is found that average price of hotel rooms is highest, followed by entire homes / apartments. the price of shared and private rooms is relatively lower. Price level reflects market positioning clearly: The entire homes and hotels target higher consumption needs, while multi-person rooms mainly cover renters with limited budget.

```
# step 2: room type vs minimum nights
min_night_rt <- nyc_clean %>%
  select(minimum_nights, room_type) %>%
  group_by(room_type) %>%
  summarise(min_nights = min(minimum_nights),
            avg_nights = mean(minimum_nights),
            median_nights = median(minimum_nights),
            max_nights = max(minimum_nights)
  )
min_night_rt
```

```
## # A tibble: 4 x 5
##   room_type       min_nights avg_nights median_nights max_nights
##   <chr>                <dbl>      <dbl>         <dbl>      <dbl>
## 1 Entire home/apt          1      16.0              7         70
## 2 Hotel room               1       6.54             1         60
## 3 Private room             1      15.6              7         64
## 4 Shared room              1      15.5              7         62
```

There are extreme outliers in the raw data of *minimum_nights*. After cleaning: The minimum length of stay of hotel rooms is the least (mostly 1 to 2 days). Other room types are concentrated on 7 days. Entire homes / apartments requirement is relatively higher with an average of 16 days. This reflects that the entire property is more inclined towards long-term rental demands or owners setting length of stay to reduce the frequency of renter changes.

4.3 Owners Structure Dimension

```r
# step 1: owner vs property count
owner <- nyc_clean %>%
  group_by(host_id) %>%
  summarise(cnt_property = n(), .groups = 'drop')
## distribution of property count grouped by owners
owner_summary <- owner %>%
  summarise(
    total_owner = sum(ifelse(owner$cnt_property, 1, 0)),
    avg_property = mean(cnt_property),
    median_property = median(cnt_property),
    max_property = max(cnt_property)
  )
owner_summary
```

```
## # A tibble: 1 x 4
##   total_owner avg_property median_property max_property
##         <dbl>        <dbl>           <dbl>        <int>
## 1       27048         1.55               1          523
```

```r
# step 2: calculate proportion of owners with multiple properties and only one property
big_owner_ratio <- round(
  sum(owner$cnt_property > 1) * 100
  / owner_summary$total_owner
  , 2)
small_owner_ratio <- 100 - big_owner_ratio

paste0('Proportion of owners with multiple properties: ', big_owner_ratio, '%')
```

```
## [1] "Proportion of owners with multiple properties: 17.98%"
```

```
paste0('Proportion of owners with only 1 properties: ', small_owner_ratio, '%')
```

```
## [1] "Proportion of owners with only 1 properties: 82.02%"
```

Most owners have only one property (82%), but still about 18% of them are “multi-property owners” or even “hotel-like operators”. The biggest owner have over 500 properties, who is signigicantly higher than other owners. The entire market displays typical long-tail feature: Large amount of owners provides fragmented supply and few super owners contribute to substantial properties. This gives an important significance for Airbnb’s platform advantages and supervision strategies.

4.4 Reviews Activity Dimension

```
# step 1: review per month vs neighbourhood group
review_ng <- nyc_clean %>%
  group_by(neighbourhood_group) %>%
  summarise(
    avg_review = mean(reviews_per_month),
    median_review = median(reviews_per_month),
    max_review = max(reviews_per_month)
  )
review_ng
```

```
## # A tibble: 5 x 4
##   neighbourhood_group avg_review median_review max_review
##   <chr>                    <dbl>         <dbl>      <dbl>
## 1 Bronx                    1.22           0.7        11  
## 2 Brooklyn                 0.917          0.3        17.2
## 3 Manhattan                0.674          0.14       19.9
## 4 Queens                   1.25           0.56       17.0
## 5 Staten Island            1.29           0.77       10.2
```

Cleaned *reviews_per_month* shows that most listing properties obtain less than 1 review each month. The difference between various areas is very slightly. This means that the entire review behavior of NYC Airbnb's renters is lower, and distribution of review count shows obviously positive skewness. Few popular listing properties generate most of reviews.

4.5 Suppy-demand Dimension

```
# step 1: availability
available <- nyc_clean %>%
  summarise(
    min_avb = min(availability_365),
    cnt_0_ratio = round(
      sum(ifelse(availability_365 == 0, 1, 0))
      / sum(ifelse(availability_365, 1, 0))
      , 3),
    avg_avb = mean(availability_365),
    median_avb = median(availability_365),
    max_avb = max(availability_365),
    cnt_365_ratio = round(
      sum(ifelse(availability_365 == 365, 1, 0))
      / sum(ifelse(availability_365, 1, 0))
      , 3)
  )
available
```

```
## # A tibble: 1 x 6
##   min_avb cnt_0_ratio avg_avb median_avb max_avb cnt_365_ratio
##     <dbl>       <dbl>   <dbl>      <dbl>   <dbl>         <dbl>
## 1       0       0.487    140.         88     365         0.081
```

```
# step 2: availability vs neighbourhood group
available_ng <- nyc_clean %>%
  group_by(neighbourhood_group) %>%
  summarise(
    avg_avb = mean(availability_365),
    median_avb = median(availability_365)
  )
available_ng
```

```
## # A tibble: 5 x 3
##   neighbourhood_group avg_avb median_avb
##   <chr>                 <dbl>      <dbl>
## 1 Bronx                  209.       248
## 2 Brooklyn               125.        65
## 3 Manhattan              132.        74
## 4 Queens                 173.       156
## 5 Staten Island          210.       210.
```

availability_365 indicates that nearly half of NYC listing properties (48.7%) show zero availability for the upcoming year.
This does not simply mean that properties are fully booked. Instead, it reflects a combination of factors such as seasonal hosting, calendar closure, regulatory restrictions, and long-term rental usage.

On average, listing properties are available for only 140 days per year, with a median of 88 days. Further stratification by *neighbourhood_group* is found that high-demand markets (**Brooklyn**, **Manhattan**) tend to lower availability (mostly 60 to 80 days), while low-demand markets (**Bronx**, **Staten Island**) have more avaible days.

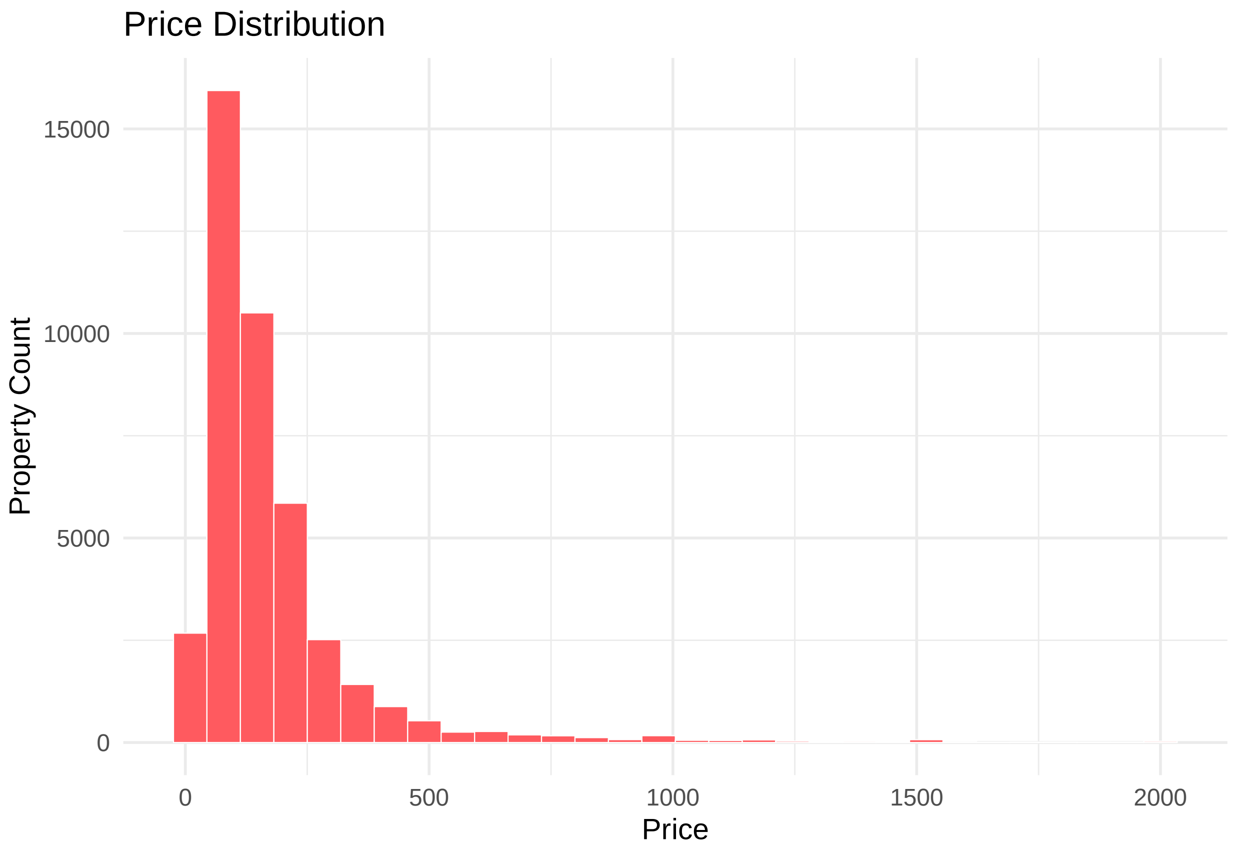
Phase 5: Visualization

This section provides a visual examination of pricing, geographical distribution, supply-demand patterns, owner structure, room type characteristics, and review activities.
Charts are organized to directly support the business questions defined in Phase 1: pricing patterns, supply shortages or unmet demand, and occupancy / review activity.

5.1 Price Analysis

(1) Price Distribution

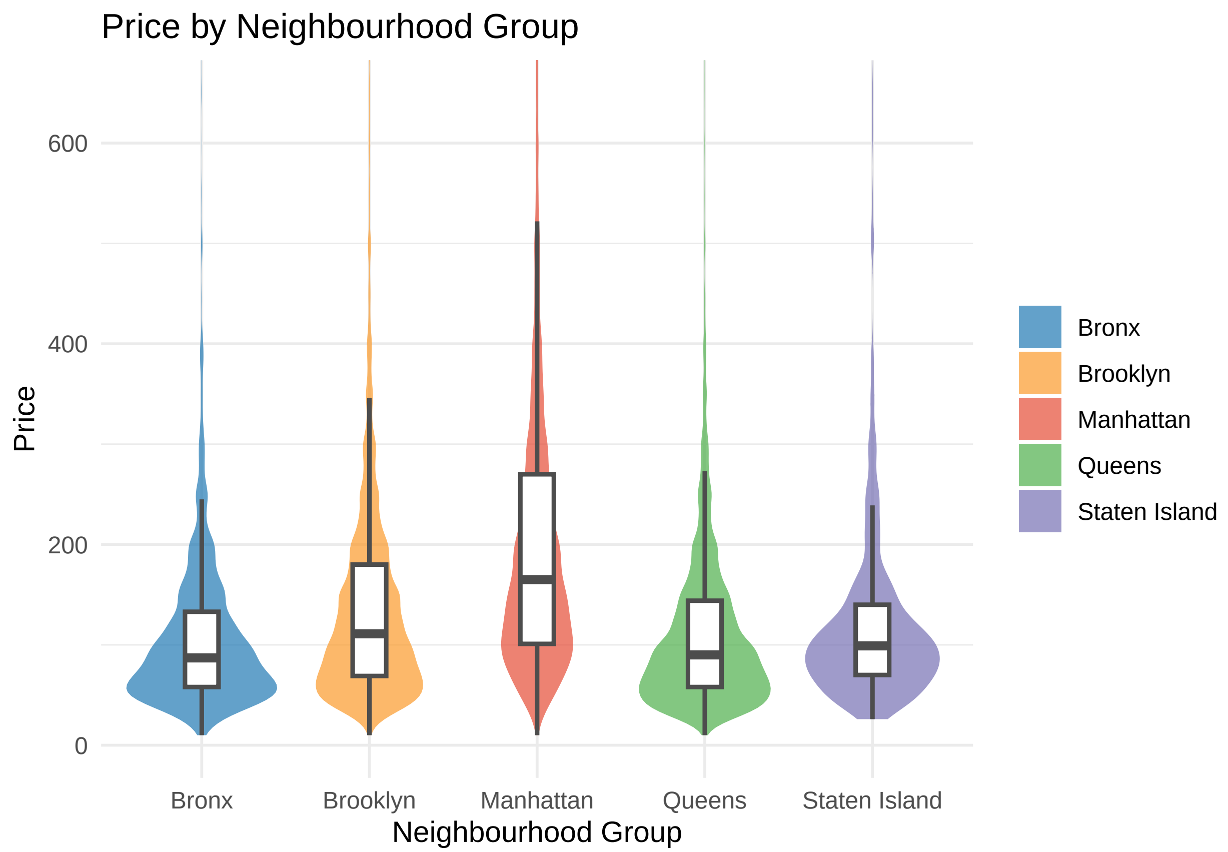
```
p1 <- nyc_clean %>%
  ggplot(aes(x = price)) +
  geom_histogram(
    linewidth =  0.2,
    bins = 30,
    fill = '#FF5A5F',
    color = 'white'
    ) +
  labs(
    title = 'Price Distribution',
    x = 'Price',
    y = 'Property Count'
    ) +
  theme(
    plot.title = element_text(face = 'bold', size = 14),
    axis.title = element_text(face = 'bold'),
    legend.title = element_blank()) +
  theme_minimal(base_size = 11)
p1
```



The overall price distribution is positively skewed, with most listing properties between $50-300, confirming a highly concentrated mid-market segment.
A long tail of higher-priced properties exists but represents only a small proportion.

(2) Price by Neighbourhood Group

```
p2 <- nyc_clean %>%
  ggplot(
    aes(
      x = neighbourhood_group,
      y = price,
      fill = neighbourhood_group
      )
    ) +
  geom_violin(trim = TRUE, alpha = 0.7, color = NA) +
  geom_boxplot(
    width = 0.2, color = 'grey30', fill = 'white',
    outlier.shape = NA, linewidth = 0.8
  ) +
  coord_cartesian(ylim = c(0, 650)) +
  scale_fill_manual(
    values = c(
      'Manhattan' = '#e64b35',
      'Brooklyn' = '#fb9a29',
      'Queens' = '#4daf4a',
      'Bronx' = '#1f78b4',
      'Staten Island' = '#7570b3'
    )
  ) +
  labs(
    title = 'Price by Neighbourhood Group',
    x = 'Neighbourhood Group',
    y = 'Price'
  ) +
  theme(
    plot.title = element_text(face = 'bold', size = 14),
    axis.title = element_text(face = 'bold'),
    ) +
  theme_minimal(base_size = 11) +
  guides(fill = guide_legend(title = NULL))
p2
```



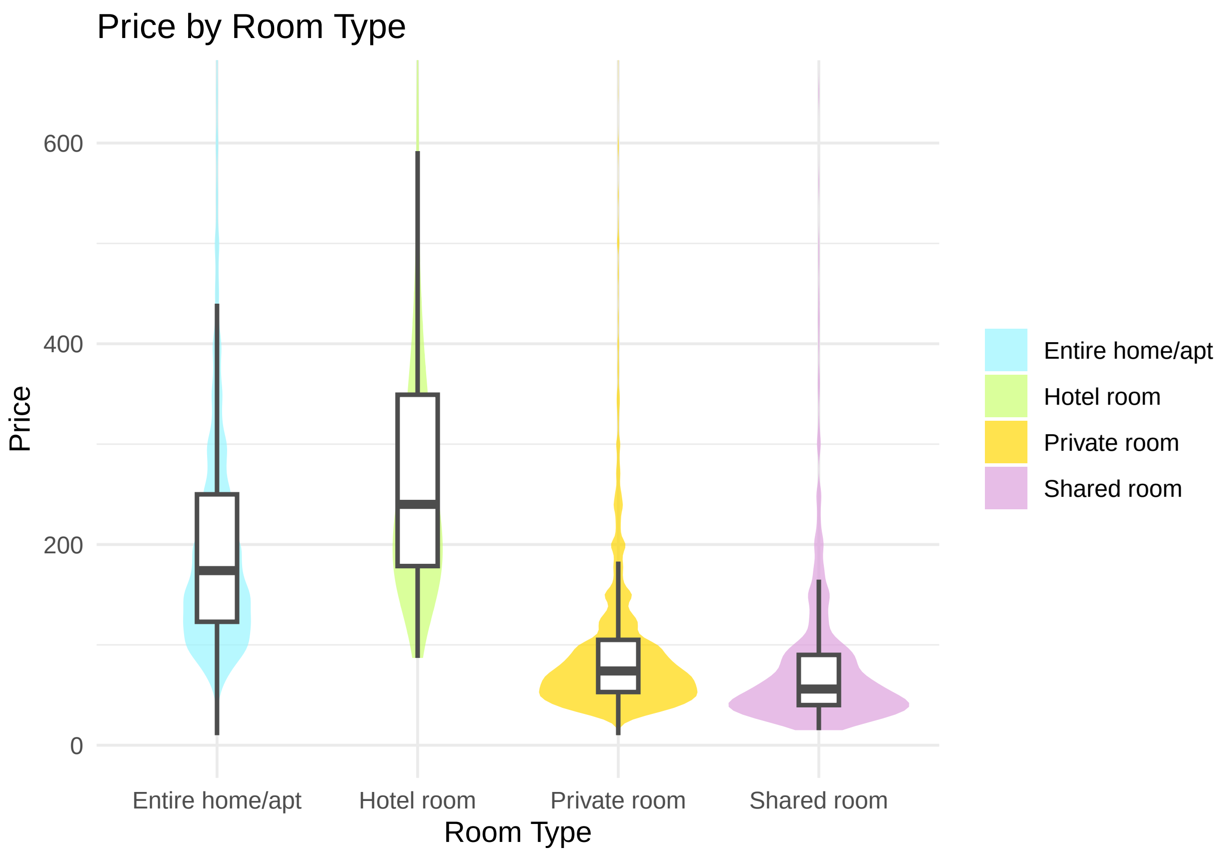
Clear spatial price gradients appear across New York:

- **Manhattan** shows the highest median prices, consistent with its central location and limited rental supply.
- **Brooklyn** ranks second but with wider variability.
- **Queens, Bronx, and Staten Island** display significantly lower prices, aligning with lower demand and less touristic concentration.

(3) Price by Room Type

```
p3 <- nyc_clean %>%
  ggplot(
    aes(
      x = room_type,
      y = price,
      fill = room_type
      )
    ) +
  geom_violin(trim = TRUE, alpha = 0.7, color = NA) +
  geom_boxplot(
```

```r
    width = 0.2, color = 'grey30', fill = 'white',
    outlier.shape = NA, linewidth = 0.8
  ) +
  coord_cartesian(ylim = c(0, 650)) +
  scale_fill_manual(
    values = c(
      'Entire home/apt' = '#98f5ff',
      'Hotel room' = '#caff70',
      'Private room' = '#ffd700',
      'Shared room' = '#dda0dd'
    )
  ) +
  labs(
    title = 'Price by Room Type',
    x = 'Room Type',
    y = 'Price'
  ) +
  theme(
    plot.title = element_text(face = 'bold', size = 14),
    axis.title = element_text(face = 'bold'),
    ) +
  theme_minimal(base_size = 11) +
  guides(fill = guide_legend(title = NULL))
p3
```



Price differences strongly reflect product positioning:

- **Hotel rooms** and **entire homes/apartments** have premium pricing.
- **Private and shared rooms** target budget-conscious renters.

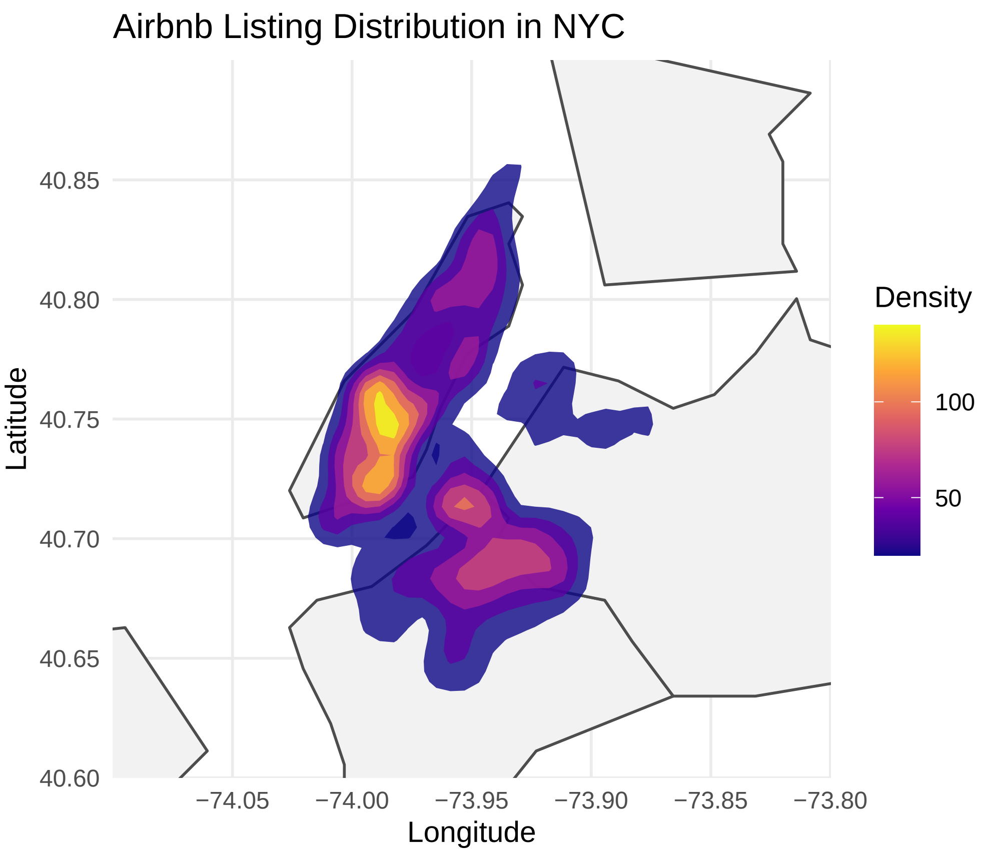
This indicates that NYC Airbnb market provides hierarchical price structures aligned with renter demands and owner pricing strategies.

5.2 Geographical Visualization

Airbnb Listing Density Map

```
nyc_county_map <- map_data('county') %>%
  filter(
    region == 'new york',
    subregion %in% c('bronx', 'new york', 'kings', 'queens', 'richmond')
  )
```

```
p4 <- ggplot() +
  geom_polygon(
    data = nyc_county_map,
    aes(x = long, y = lat, group = group),
    fill = "grey95", color = "grey30"
  ) +
  stat_density_2d(
    data = nyc_clean,
    aes(x = longitude, y = latitude, fill = after_stat(level)),
    geom = 'polygon',
    alpha = 0.8
  ) +
  scale_fill_viridis_c(option = 'C') +
  coord_sf(
    xlim = c(-74.1, -73.8),
    ylim = c(40.6, 40.9),
    expand = FALSE
  ) +
  labs(
    title = 'Airbnb Listing Distribution in NYC',
    fill = 'Density',
    x = 'Longitude',
    y = 'Latitude'
  ) +
  theme(
    plot.title = element_text(face = 'bold', size = 14),
    axis.title = element_text(face = 'bold'),
    ) +
  theme_minimal()
p4
```



The density map highlights obvious spatial clustering:

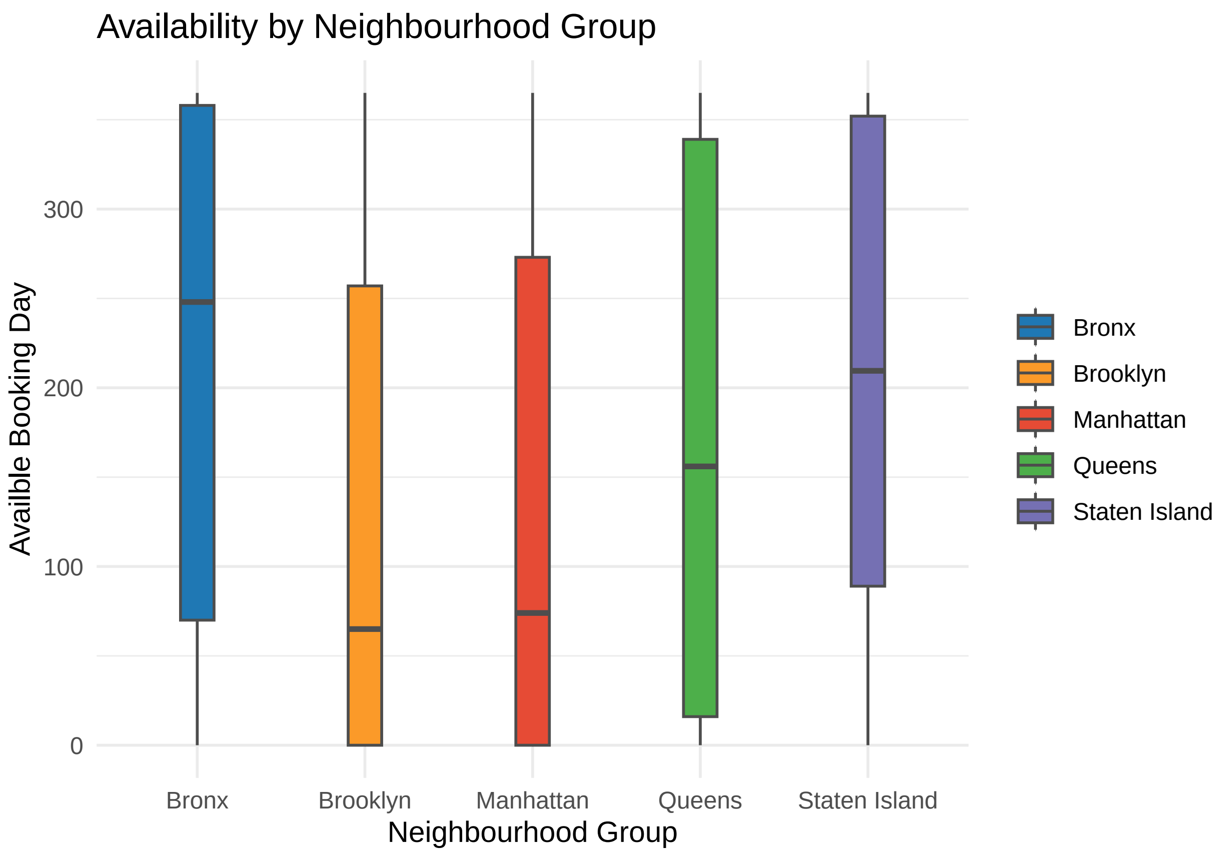
- Highest concentrations appear in **Midtown, Lower Manhattan, Williamsburg, and Bedford-Stuyvesant**, determining them as Airbnb hotspots.
- Outer neighbourhoods such as Staten Island show sparse activity.

Intensive areas may indicate competitive pricing pressure, while sparse areas may represent growth opportunites or regulatory constraints.

5.3 Supply & Demand

Availability vs Neighbourhood Group

```
p5 <- nyc_clean %>%
  ggplot(
    aes(
      x = neighbourhood_group,
      y = availability_365,
      fill = neighbourhood_group
      )
    ) +
  geom_boxplot(
    width = 0.2, color = 'grey30',
    outlier.shape = NA, linewidth = 0.5
    ) +
  coord_cartesian(ylim = c(0, 365)) +
  scale_fill_manual(
    values = c(
      'Manhattan' = '#e64b35',
      'Brooklyn' = '#fb9a29',
      'Queens' = '#4daf4a',
      'Bronx' = '#1f78b4',
      'Staten Island' = '#7570b3'
    )
  ) +
  labs(
    title = 'Availability by Neighbourhood Group',
    x = 'Neighbourhood Group',
    y = 'Availble Booking Day'
  ) +
  theme(
    plot.title = element_text(face = 'bold', size = 14),
    axis.title = element_text(face = 'bold')
    ) +
  theme_minimal(base_size = 11) +
  guides(fill = guide_legend(title = NULL))
p5
```



availability_365 shows strong demand patterns:

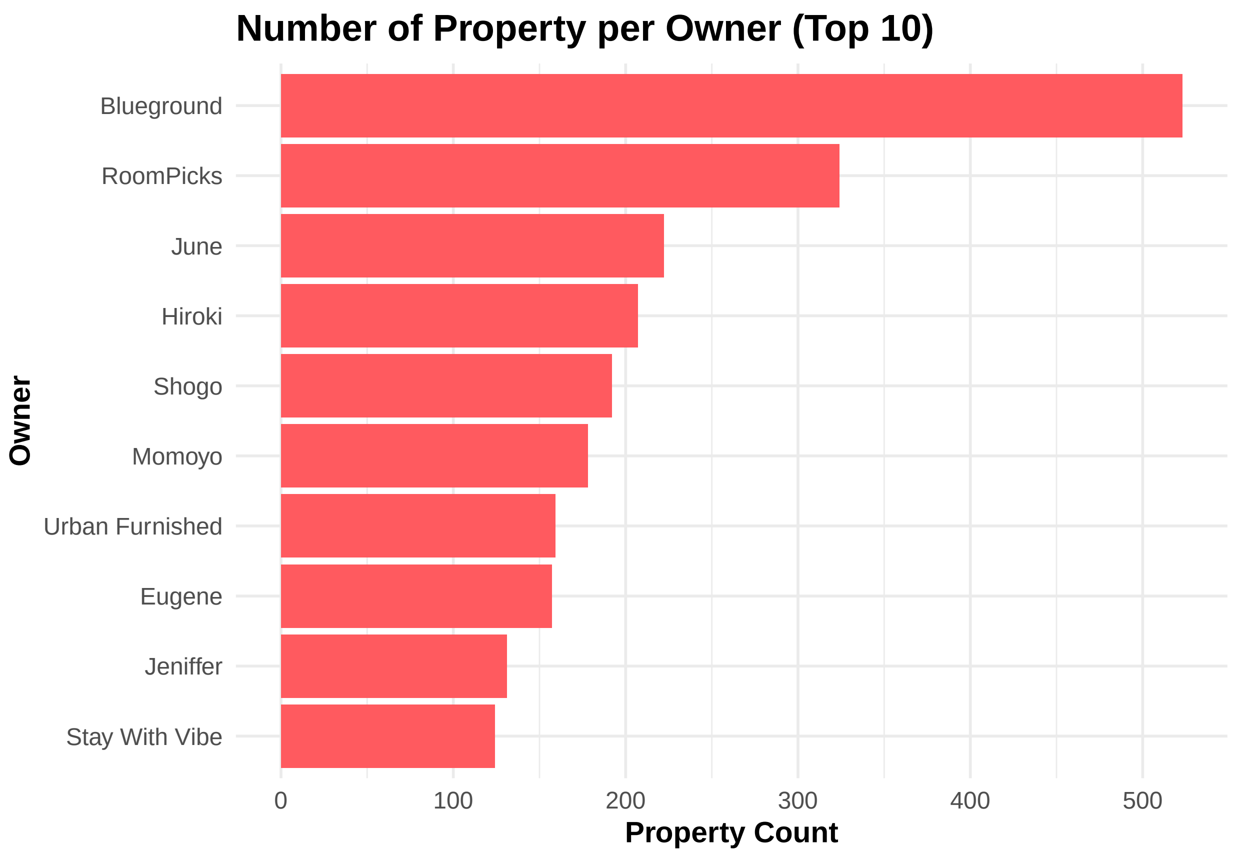
- Nearly 50% of listing properties display zero availability, suggesting either consistently high booking demand or restricted calendars.
- **Brooklyn and Manhattan** reflects the lowest availability, indicating supply-constrained and high-demand markets.
- **Bronx and Staten Island** have high availability, implying slower demand and more useable capacity.

5.4 Owner Structure

(1) Top 10 Owners by Listing Property Count

```
# calculate the property count owned by each owner (top 10)
owner_top10 <- nyc_clean %>%
  count(host_name, host_id, name = "property_count") %>%
  arrange(desc(property_count)) %>%
  slice_head(n = 10)
```

```r
# plot
p6 <- owner_top10 %>%
  ggplot(
    aes(x = property_count,
        y = reorder(host_name, property_count))) +
  geom_bar(stat = 'identity', fill = '#FF5A5F') +
  labs(
    title = 'Number of Property per Owner (Top 10)',
    x = 'Property Count',
    y = 'Owner'
  ) +
  theme_minimal(base_size = 11) +
  theme(
    plot.title = element_text(face = 'bold', size = 14),
    axis.title = element_text(face = 'bold')
  )
p6
```



The top owners control disproportionately large inventories, some exceeding hundreds of properties.

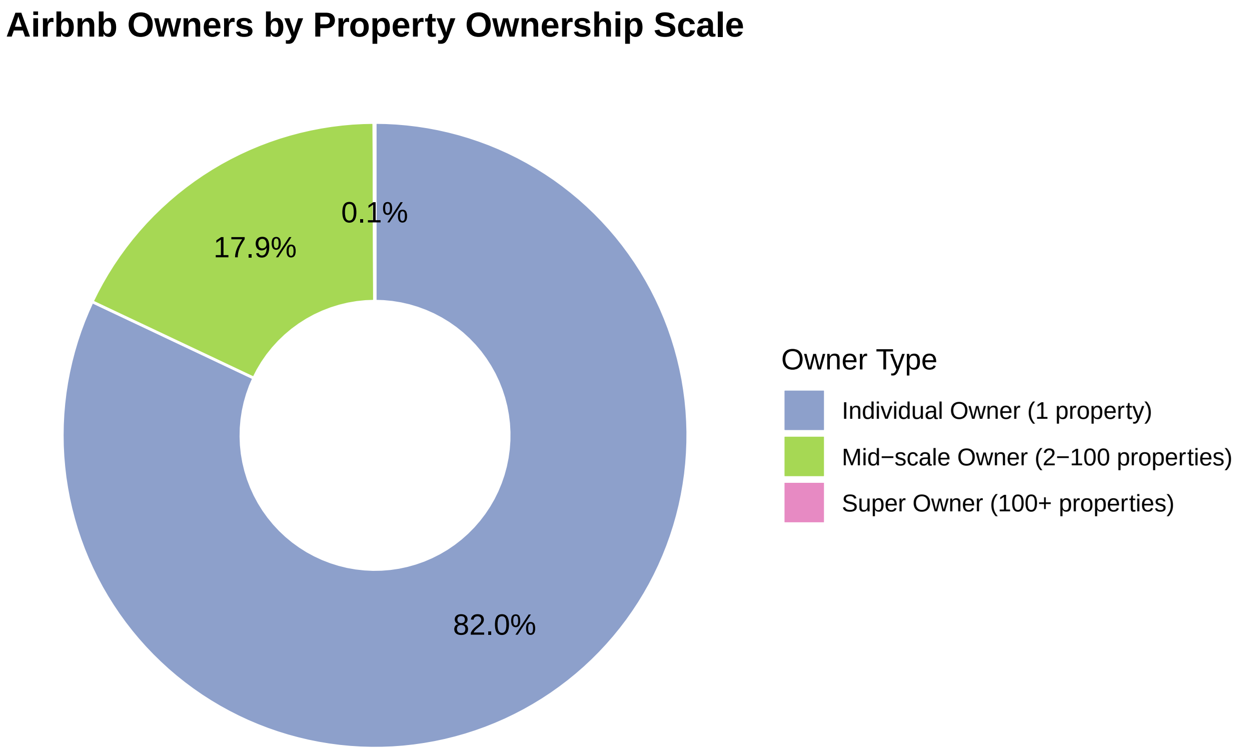
This reflects a “hotel-like operator” mode and shows a long-tail market structure, where most owners remain small-scale, but a few dominate supply volume.

(2) Owner Type Distribution

```r
# calculate the owner count
owner_class <- nyc_clean %>%
  count(host_id, name = 'property_count') %>%
  mutate(
    owner_type = case_when(
      property_count == 1 ~ 'Individual Owner (1 property)',
      property_count > 1 & property_count <= 100 ~ 'Mid-scale Owner (2-100 properties)',
      property_count > 100 ~ 'Super Owner (100+ properties)'
    )
  ) %>%
  count(owner_type, name = 'n')
owner_class
```

```
## # A tibble: 3 x 2
##   owner_type                             n
##   <chr>                              <int>
## 1 Individual Owner (1 property)      22186
## 2 Mid-scale Owner (2-100 properties)  4847
## 3 Super Owner (100+ properties)         15
```

```r
# prepare data
df <- owner_class %>%
  mutate(
    ratio = n / sum(n),
    ymax = cumsum(ratio),
    ymin = c(0, head(ymax, n = -1)),
    label_position = (ymax + ymin) / 2,
    label = paste0(percent(ratio, accuracy = 0.1))
  )
# plot
p7 <- df %>%
  ggplot(
    aes(ymax = ymax, ymin = ymin,
        xmax = 4, xmin = 2,
        fill = owner_type)
  ) +
  geom_rect(color = 'white') +
  coord_polar(theta = 'y') +
  xlim(c(0.5, 4)) +
  theme_void() +
  scale_fill_manual(
    values = c(
      'Individual Owner (1 property)' = '#8da0cb',
      'Mid-scale Owner (2-100 properties)' = '#a6d854',
      'Super Owner (100+ properties)' = '#e78ac3'
    )
  ) +
  geom_text(
    aes(x = 3, y = label_position, label = label),
    size = 4
  ) +
  labs(
    title = 'Airbnb Owners by Property Ownership Scale',
    fill = 'Owner Type'
  ) +
  theme(
    plot.title = element_text(face = 'bold', size = 13, hjust = 0.5)
  )
p7
```



The ownership landscape is:

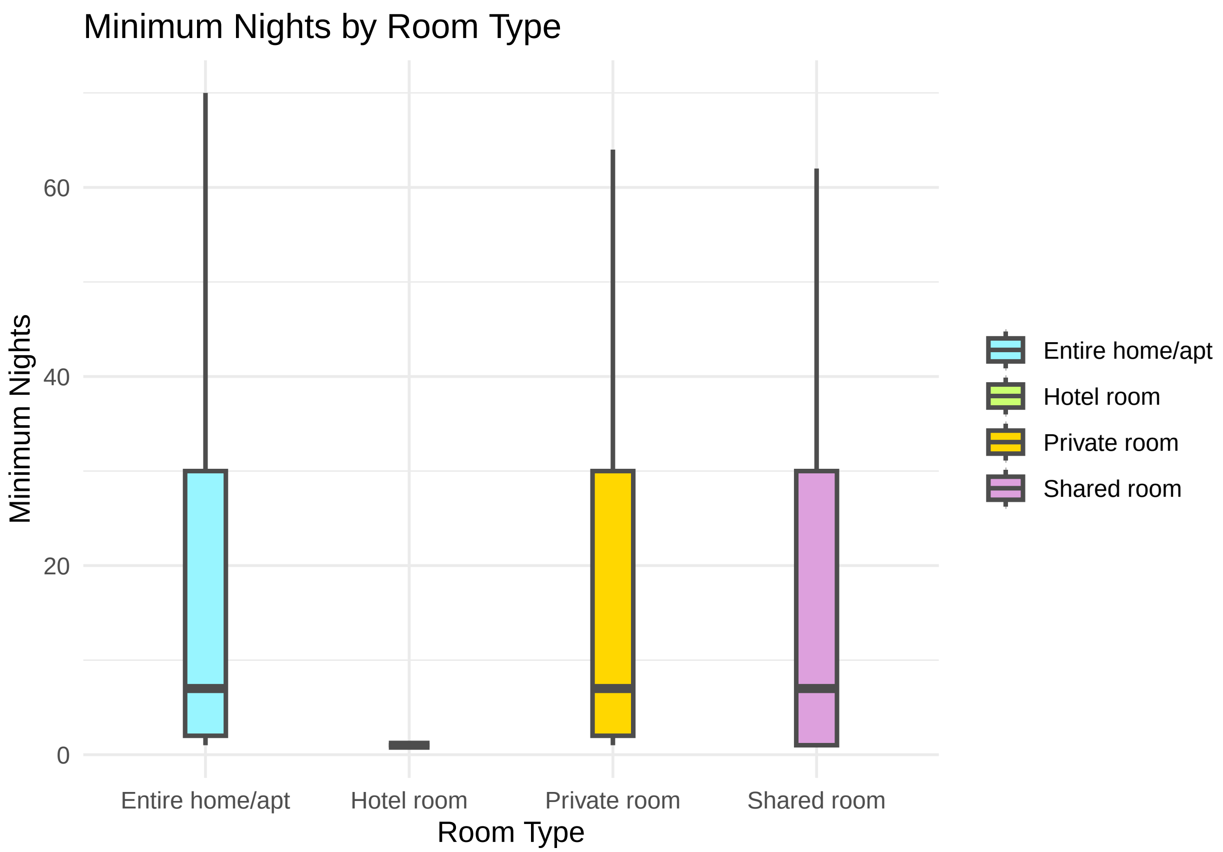
- 82% individual owners with 1 property
- ~18% mid-scale owners with 2 to 100 properties
- <1% super owners with 100+ properties

The big proportion of individual owners means that Airbnb remains largely a point-to-point market, but the existing of large operators may affect pricing and regional competition.

5.5 Room Type Comparison

Minimum Nights by Room Type

```r
p8 <- nyc_clean %>%
  ggplot(
    aes(
      x = room_type,
      y = minimum_nights,
      fill = room_type
      )
    ) +
  geom_boxplot(
    width = 0.2, color = 'grey30',
    outlier.shape = NA, linewidth = 0.8
  ) +
  scale_fill_manual(
    values = c(
      'Entire home/apt' = '#98f5ff',
      'Hotel room' = '#caff70',
      'Private room' = '#ffd700',
      'Shared room' = '#dda0dd'
    )
  ) +
  labs(
    title = 'Minimum Nights by Room Type',
    x = 'Room Type',
    y = 'Minimum Nights'
  ) +
  theme(
    plot.title = element_text(face = 'bold', size = 14),
    axis.title = element_text(face = 'bold'),
    ) +
  theme_minimal(base_size = 11) +
  guides(fill = guide_legend(title = NULL))
p8
```



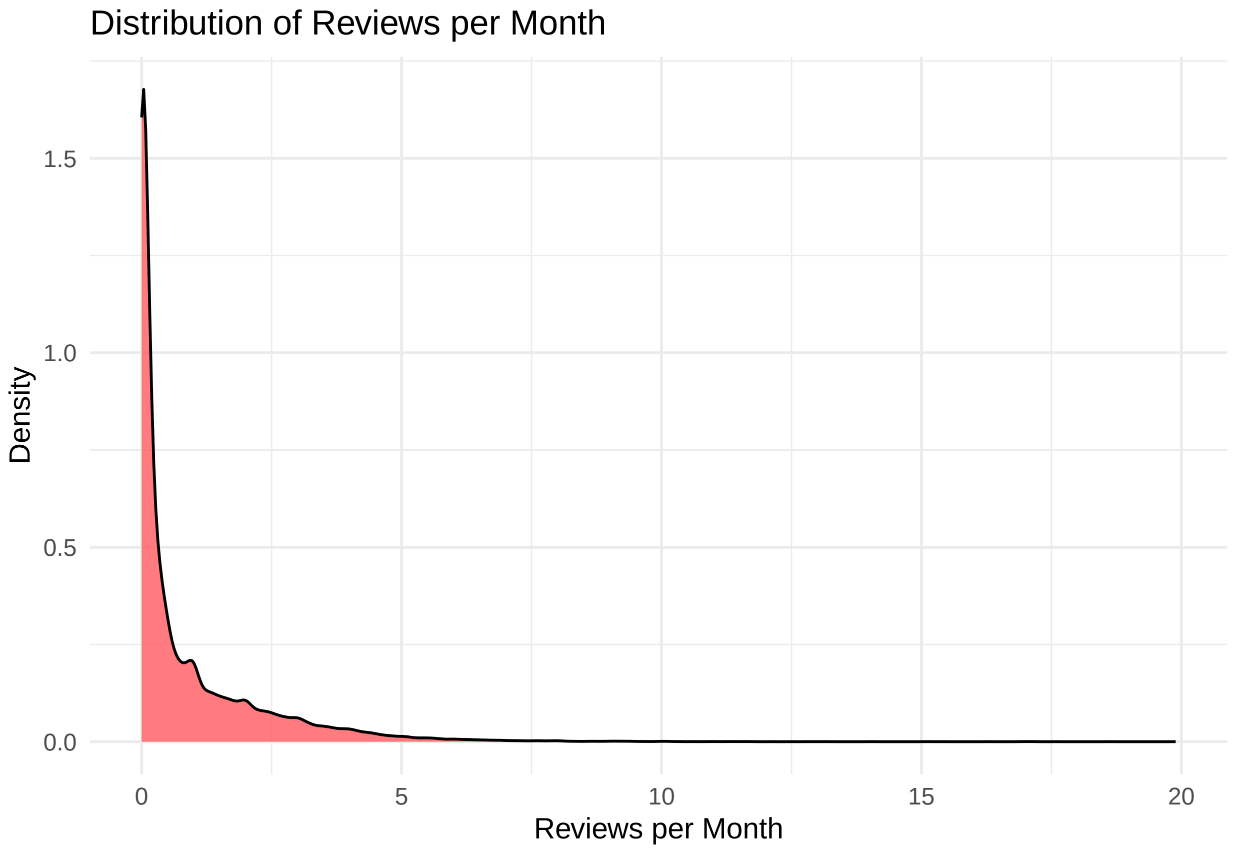
Minimum night requirements differ by room type:

- **Hotel rooms** allow the shortes stays (often 1 night).
- **Entire homes and private rooms** usually require 7+ nights, indicating owners' preference for lowering renter changes or targeting medium-term renters.

5.6 Review Activity

Distribution of Reviews per Month

```r
p9 <- nyc_clean %>%
  ggplot(aes(x = reviews_per_month)) +
  geom_density(fill = '#FF5A5F', alpha = 0.8) +
  labs(
    title = 'Distribution of Reviews per Month',
    x = 'Reviews per Month',
    y = 'Density'
  ) +
  theme(
    plot.title = element_text(face = 'bold', size = 14),
    axis.title = element_text(face = 'bold'),
    ) +
  theme_minimal(base_size = 11) +
  guides(fill = guide_legend(title = NULL))
p9
```



Reviews per month display a highly skewed pattern:

- Most listing properties receive 0-1 review monthly, which means that renters are not enthusiastic about providing feedback overall.
- Only a small proportion have high review volume, indicating a minority of high-performing, consistently booked listing properties.

Review activity (a proxy for occupancy) is unevenly distributed and concentrated among popular listing properties.

Phase 6: Key Insights & Business Recommendations

A. Key Market Insights

1. **Airbnb NYC market shows a long-tail owner distribution**, with 82% individual owners and <1% commercial-scale operators, yet the latter dominate the supply.
2. **Strong spatial concentration** occurs in Manhattan, Williamsburg, and Bedford-Stuyvesant, forming the primary short-term rental clusters in NYC.
3. **Pricing closely correlates with location**, with Manhattan consistently ranking highest.
4. **Demand is uneven**, as indicated by skewed review activity. Only a small proportion of listing properties capture the majority of renter traffic.
5. **Availability patterns indicate supply-demand imbalance**, especially in Manhattan and Brooklyn.

B. Recommendations

1. **Increase supply or encourage owners in high-demand regions** such as Manhattan and Brooklyn.
2. **Enhance visibility for low-performing listing properties** (less reviews) via platform recommendation algorithms.
3. **Support individual owners** with pricing and occupancy optimization tools, countering the advantage of large operators.
4. **Strategically promote listing properties in emerging areas** (Queens/Bronx) to balance spatial distribution and improve platform coverage.