Step 1_Data Preparation & Exploration

Trista

2023-08-13

```r
library(tidyverse)
```

```
## ── Attaching packages ─────────────────────────────────────── tidyverse 1.3.2 ──
## ✓ ggplot2 3.4.0     ✓ purrr   1.0.1
## ✓ tibble  3.2.1     ✓ dplyr   1.1.2
## ✓ tidyr   1.3.0     ✓ stringr 1.5.0
## ✓ readr   2.1.3     ✓ forcats 1.0.0
```

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

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

```
## ── Conflicts ────────────────────────────────────────── tidyverse_conflicts() ──
## ✕ dplyr::filter() masks stats::filter()
## ✕ dplyr::lag()    masks stats::lag()
```

```r
library(caret)
```

```
## Warning: package 'caret' was built under R version 4.2.3
## Loading required package: lattice
##
## Attaching package: 'caret'
##
## The following object is masked from 'package:purrr':
##
##     lift
```

```
#1. set the working directory and Read CSV files
setwd("C:/Users/Trista Hu/OneDrive/BU ABA Study Summer/AD699 Data Mining Summer/Team Project")

# Import the cleaned csv file for further analysis
df <- read_csv("hong_kong_cleaned.csv")
```

```
## New names:
## Rows: 1424 Columns: 50
## ── Column specification
## ──────────────────────────────────────────────────────── Delimiter: "," chr
## (10): description, neighborhood_overview, host_location, host_response_... dbl
## (34): ...1, host_id, host_response_rate, host_acceptance_rate, host_lis... lgl
## (5): host_is_superhost, host_has_profile_pic, host_identity_verified, ... date
## (1): host_since
## 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.
## • `` -> `...1`
```

```
#II. Summary Statistics
str(df)
```

```
## spc_tbl_ [1,424 × 50] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
##  $ ...1                                        : num [1:1424] 1 2 3 4 5 6 7 8 9 10 ...
##  $ description                                 : chr [1:1424] "Flat is very quiet. A/C in each room and living room." "*
共居時光*<br /><br />關於我們：<br />共居時光是一家現代化的服務式公寓公司，為您提供不同類型的舒適房間，從合租"| __truncated__ "near
MTR<br />8 rooms sharing apartment with a kitchen and washing machine<br />with private bathroom<br /><br "| __truncated__
"This listing is for 1 private room in a shared apartment.<br />If you like to meet new friends, expand your Net"| __truncat
ed__ ...
##  $ neighborhood_overview                       : chr [1:1424] "None" "None" "None" "None" ...
##  $ host_id                                     : num [1:1424] 1.25e+08 3.81e+07 7.52e+06 7.52e+06 1.20e+08 ...
##  $ host_since                                  : Date[1:1424], format: "2017-04-10" "2015-07-10" ...
##  $ host_location                               : chr [1:1424] "Hong Kong" "Hong Kong" "Hong Kong" "Hong Kong" ...
##  $ host_response_time                          : chr [1:1424] "within a few hours" "within a few hours" "within a few hou
rs" "within a few hours" ...
##  $ host_response_rate                          : num [1:1424] 1 1 0.98 0.98 0.33 0.94 0.94 1 1 1 ...
##  $ host_acceptance_rate                        : num [1:1424] 0.57 0.57 0.57 0.57 0 0.2 0.2 0.26 0.2 1 ...
##  $ host_is_superhost                           : logi [1:1424] FALSE FALSE FALSE FALSE FALSE FALSE ...
##  $ host_neighbourhood                          : chr [1:1424] "Wan Chai" "Causeway Bay" "Wan Chai" "Wan Chai" ...
##  $ host_listings_count                         : num [1:1424] 2 1 365 365 9 441 441 367 304 7 ...
##  $ host_total_listings_count                   : num [1:1424] 2 1 396 396 11 505 505 379 323 9 ...
##  $ host_verifications                          : chr [1:1424] "['email', 'phone', 'work_email']" "['email', 'phone']" "
['email', 'phone']" "['email', 'phone']" ...
##  $ host_has_profile_pic                        : logi [1:1424] TRUE TRUE TRUE TRUE TRUE TRUE ...
##  $ host_identity_verified                      : logi [1:1424] FALSE TRUE TRUE TRUE FALSE TRUE ...
##  $ latitude                                    : num [1:1424] 22.3 22.3 22.3 22.3 22.3 ...
##  $ longitude                                   : num [1:1424] 114 114 114 114 114 ...
##  $ property_type                               : chr [1:1424] "Entire rental unit" "Shared room in serviced apartment" "P
rivate room in rental unit" "Private room in rental unit" ...
##  $ room_type                                   : chr [1:1424] "Entire home/apt" "Shared room" "Private room" "Private roo
m" ...
##  $ accommodates                                : num [1:1424] 2 7 2 2 3 1 1 1 2 2 ...
##  $ bathroom_nb                                 : num [1:1424] 1.5 2 1 1 1 1 1 1 1 1 ...
##  $ bathroom_type                               : chr [1:1424] "normal" "shared" "normal" "private" ...
##  $ bedrooms                                    : num [1:1424] 2 7 1 1 1 1 1 1 1 1 ...
##  $ beds                                        : num [1:1424] 2 7 1 1 1 1 1 1 1 1 ...
##  $ amenities                                   : chr [1:1424] "[\"Wifi\", \"Kitchen\", \"Washer\", \"Air conditioning\",
\"Smoke alarm\"]" "[\"Hangers\", \"Wifi\", \"Hot water\", \"TV\", \"Shampoo\", \"Kitchen\", \"Washer\", \"Air conditioning
\", \"Elevator\"]" "[\"Iron\", \"Air conditioning\", \"Wifi\", \"Kitchen\"]" "[\"Iron\", \"Hangers\", \"Wifi\", \"Lock on be
droom door\", \"Kitchen\", \"Washer\", \"Air conditioning\"]" ...
##  $ price                                       : num [1:1424] 470 500 217 160 1150 180 180 180 160 950 ...
##  $ minimum_nights                              : num [1:1424] 30 30 29 29 30 30 30 29 29 28 ...
##  $ maximum_nights                              : num [1:1424] 43 365 1125 1125 366 ...
##  $ has_availability                            : logi [1:1424] FALSE TRUE TRUE TRUE TRUE TRUE ...
##  $ availability_30                             : num [1:1424] 0 23 29 29 0 30 30 29 29 0 ...
##  $ availability_60                             : num [1:1424] 0 53 59 59 0 60 60 59 59 0 ...
##  $ availability_90                             : num [1:1424] 0 83 89 89 0 90 90 89 89 0 ...
##  $ availability_365                            : num [1:1424] 0 83 364 364 0 365 365 364 364 0 ...
##  $ number_of_reviews                           : num [1:1424] 1 0 0 0 0 1 0 0 0 0 ...
##  $ number_of_reviews_ltm                       : num [1:1424] 0 0 0 0 0 0 0 0 0 0 ...
##  $ number_of_reviews_l30d                      : num [1:1424] 0 0 0 0 0 0 0 0 0 0 ...
##  $ review_scores_rating                        : num [1:1424] 5 0 0 0 0 4 0 0 0 0 ...
##  $ review_scores_accuracy                      : num [1:1424] 5 0 0 0 0 4 0 0 0 0 ...
##  $ review_scores_cleanliness                   : num [1:1424] 5 0 0 0 0 3 0 0 0 0 ...
##  $ review_scores_checkin                       : num [1:1424] 5 0 0 0 0 3 0 0 0 0 ...
##  $ review_scores_communication                 : num [1:1424] 5 0 0 0 0 5 0 0 0 0 ...
##  $ review_scores_location                      : num [1:1424] 5 0 0 0 0 5 0 0 0 0 ...
##  $ review_scores_value                         : num [1:1424] 5 0 0 0 0 3 0 0 0 0 ...
##  $ instant_bookable                            : logi [1:1424] FALSE FALSE FALSE FALSE FALSE FALSE ...
##  $ calculated_host_listings_count_entire_homes : num [1:1424] 1 0 15 15 5 13 13 18 26 6 ...
##  $ calculated_host_listings_count_private_rooms: num [1:1424] 0 0 322 322 2 384 384 332 265 0 ...
##  $ calculated_host_listings_count_shared_rooms : num [1:1424] 0 1 28 28 0 8 8 16 12 0 ...
##  $ reviews_per_month                           : num [1:1424] 0.05 0 0 0 0 0.01 0 0 0 0 ...
##  $ got_reviewed                                : num [1:1424] 1 0 0 0 0 1 0 0 0 0 ...
##  - attr(*, "spec")=
##   .. cols(
##   ..   ...1 = col_double(),
##   ..   description = col_character(),
##   ..   neighborhood_overview = col_character(),
##   ..   host_id = col_double(),
##   ..   host_since = col_date(format = ""),
##   ..   host_location = col_character(),
##   ..   host_response_time = col_character(),
##   ..   host_response_rate = col_double(),
##   ..   host_acceptance_rate = col_double(),
##   ..   host_is_superhost = col_logical(),
##   ..   host_neighbourhood = col_character(),
##   ..   host_listings_count = col_double(),
##   ..   host_total_listings_count = col_double(),
##   ..   host_verifications = col_character(),
##   ..   host_has_profile_pic = col_logical(),
##   ..   host_identity_verified = col_logical(),
##   ..   latitude = col_double(),
##   ..   longitude = col_double(),
##   ..   property_type = col_character(),
##   ..   room_type = col_character(),
##   ..   accommodates = col_double(),
##   ..   bathroom_nb = col_double(),
##   ..   bathroom_type = col_character(),
##   ..   bedrooms = col_double(),
##   ..   beds = col_double(),
##   ..   amenities = col_character(),
##   ..   price = col_double(),
##   ..   minimum_nights = col_double(),
##   ..   maximum_nights = col_double(),
##   ..   has_availability = col_logical(),
##   ..   availability_30 = col_double(),
##   ..   availability_60 = col_double(),
##   ..   availability_90 = col_double(),
##   ..   availability_365 = col_double(),
##   ..   number_of_reviews = col_double(),
##   ..   number_of_reviews_ltm = col_double(),
##   ..   number_of_reviews_l30d = col_double(),
##   ..   review_scores_rating = col_double(),
##   ..   review_scores_accuracy = col_double(),
##   ..   review_scores_cleanliness = col_double(),
##   ..   review_scores_checkin = col_double(),
##   ..   review_scores_communication = col_double(),
##   ..   review_scores_location = col_double(),
##   ..   review_scores_value = col_double(),
##   ..   instant_bookable = col_logical(),
##   ..   calculated_host_listings_count_entire_homes = col_double(),
##   ..   calculated_host_listings_count_private_rooms = col_double(),
##   ..   calculated_host_listings_count_shared_rooms = col_double(),
##   ..   reviews_per_month = col_double(),
##   ..   got_reviewed = col_double()
##   .. )
##  - attr(*, "problems")=<externalptr>
```

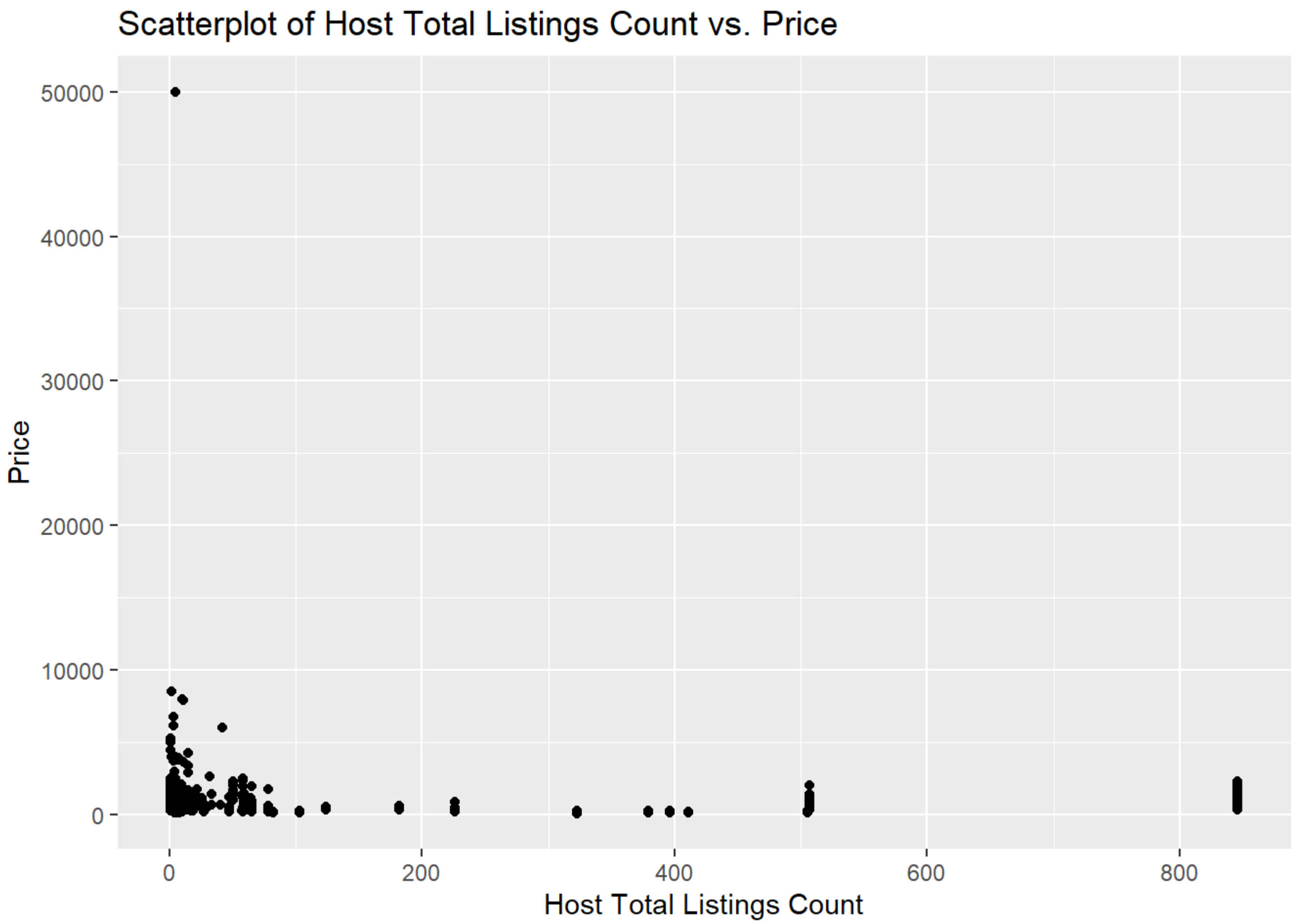
```r
# five summaries of host_total_listings_count
df.summary <- summary(df$host_total_listings_count)
df.summary
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
##     1.0    11.0   182.0   235.4   396.0   846.0
```

```r
# five summaries grouped by room type
grouped_summary <- df %>%
  group_by(room_type) %>%
  summarize(
    mean_count = mean(host_total_listings_count),
    median_count = median(host_total_listings_count),
    min_count = min(host_total_listings_count),
    max_count = max(host_total_listings_count),
    sd_count = sd(host_total_listings_count)
  )
grouped_summary
```

```
## # A tibble: 4 × 6
##   room_type       mean_count median_count min_count max_count sd_count
##   <chr>                <dbl>        <dbl>     <dbl>     <dbl>    <dbl>
## 1 Entire home/apt       159.         13.5         1       846     276.
## 2 Hotel room             65          65          65        65      NA 
## 3 Private room          289.        379           1       846     192.
## 4 Shared room           153.         65           1       505     172.
```

```r
library(ggplot2)
# discover the relationship between host_total_listings_count and price with a scatterplot
scatter_plot <- ggplot(df, aes(x = host_total_listings_count, y = price)) +
  geom_point() +
  labs(x = "Host Total Listings Count", y = "Price") +
  ggtitle("Scatterplot of Host Total Listings Count vs. Price")
print(scatter_plot)
```



```
# Check if the host_total_listings_count is the sum of individual counts
df2<-df
df2$check <- ifelse(df2$host_total_listings_count ==
                       df2$calculated_host_listings_count_entire_homes +
                       df2$calculated_host_listings_count_private_rooms +
                       df2$calculated_host_listings_count_shared_rooms,
                     "Equal", "Not Equal")
table(df2$check)
```

```
## 
##     Equal Not Equal 
##       163      1261
```

```
# check on the difference
df2$difference <- df2$host_total_listings_count-
                      (df2$calculated_host_listings_count_entire_homes +
                      df2$calculated_host_listings_count_private_rooms +
                      df2$calculated_host_listings_count_shared_rooms)
# Display the difference
table(df2$difference)
```

```
## 
##   0   1   2   3   4   5   6   7   8   9  10  11  12  13  14  16  18  19  20  21 
## 163 180  61  44  52  20  16   5   6  10   9   8   8  59  44   1   1   2 113   8 
##  24  28  29  31  38  41  45  49  62 100 158 194 418 696 
##   1   4   2 210   1   1   3 106   1 130  24  31  37  63
```

```
# Based on the calculation, The host_total_listings_count does not always equal the sum of individual calculated host counts
```

The dataset reveals a wide-ranging distribution of listings, with a minimum of 1 listing and a maximum of 846 listings. The median value of 182 signifies that a significant portion of hosts offer 182 or fewer listings, while a slightly right-skewed distribution is indicated by a mean value of 235.6, suggesting that a few hosts present a notably higher number of listings. The analysis of host_total_listings_count in relation to different room_type categories provides valuable insights into the diversity of accommodations offered by hosts. Distinct patterns emerge when considering specific room_type categories. The consistent mean count of 65 for “Hotel room” listings implies that a single host exclusively offers this type of accommodation. “Private room” listings, with a higher mean count of approximately 288.9, point toward a diverse array of private room selections. Meanwhile, the moderate mean count of about 153.5 for “Shared room” listings suggests a moderate offering of shared accommodations. The summary reveals the variety of strategies employed by hosts. While a substantial number of hosts offer a moderate number of listings - likely individual owned, a few hosts stand out by offering a notably higher count. These hosts may include property management companies, investors with a large number of properties, capitalizing on the demand for accommodations in Hong Kong.

“Entire home/apt” listings exhibit the highest standard deviation (sd_count) of approximately 276.18. This elevated standard deviation suggests a notable degree of variability in the host_total_listings_count within this room type. The mean count of around 158.51 implies a substantial average number of entire homes or apartments being offered by hosts. The median count of 13.5, however, indicates that a significant proportion of hosts within this category have the relatively lower number of listings – the lowest among all types. The range from a minimum count of 1 to a maximum of 846 listings underscores the diversity in offerings, while the larger standard deviation points to a wider spread of listing counts, possibly reflecting varying levels of investment and commitment among hosts in Hong Kong.

The scatterplot of “Price” against “host_total_listings_count” reveals that there is no clear linear relationship between the two variables. Prices for single listings vary widely, ranging from 0 to 10000, with 50000 as the highest. Most listings are priced below 100, while those with more than 100 listings tend to have less price variation. This could because the competitive nature of the Hong Kong Airbnb market can drive hosts to differentiate their offerings based on investment levels. Those with higher investments may provide distinctive amenities, meticulously designed spaces, and supplementary services to cater to guests seeking a luxury experience. Conversely, hosts with a limited room offering seem to exhibit a broad price range, accommodating customers with both basic needs and more extravagant preferences. This intricate interplay between investment decisions, market competition, and diverse customer demands contributes to the observed variations in listing counts and pricing strategies among “Entire home/apt” listings in the Hong Kong Airbnb market.

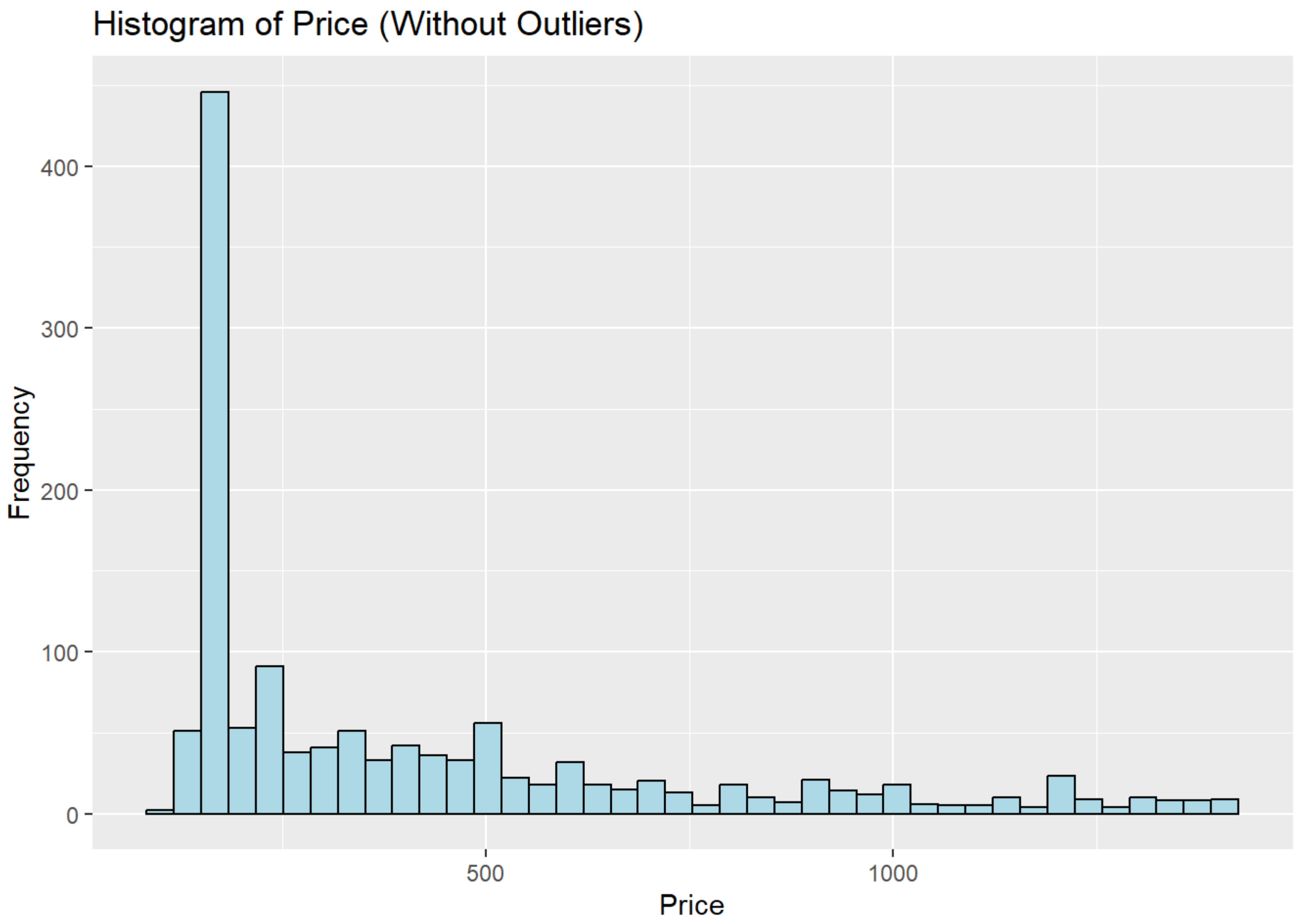
```
#III. Data Visualization
# check on the five summaries on variable "price"
summary(df$price)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
##   110.0   160.0   304.5   618.8   670.2 50000.0
```

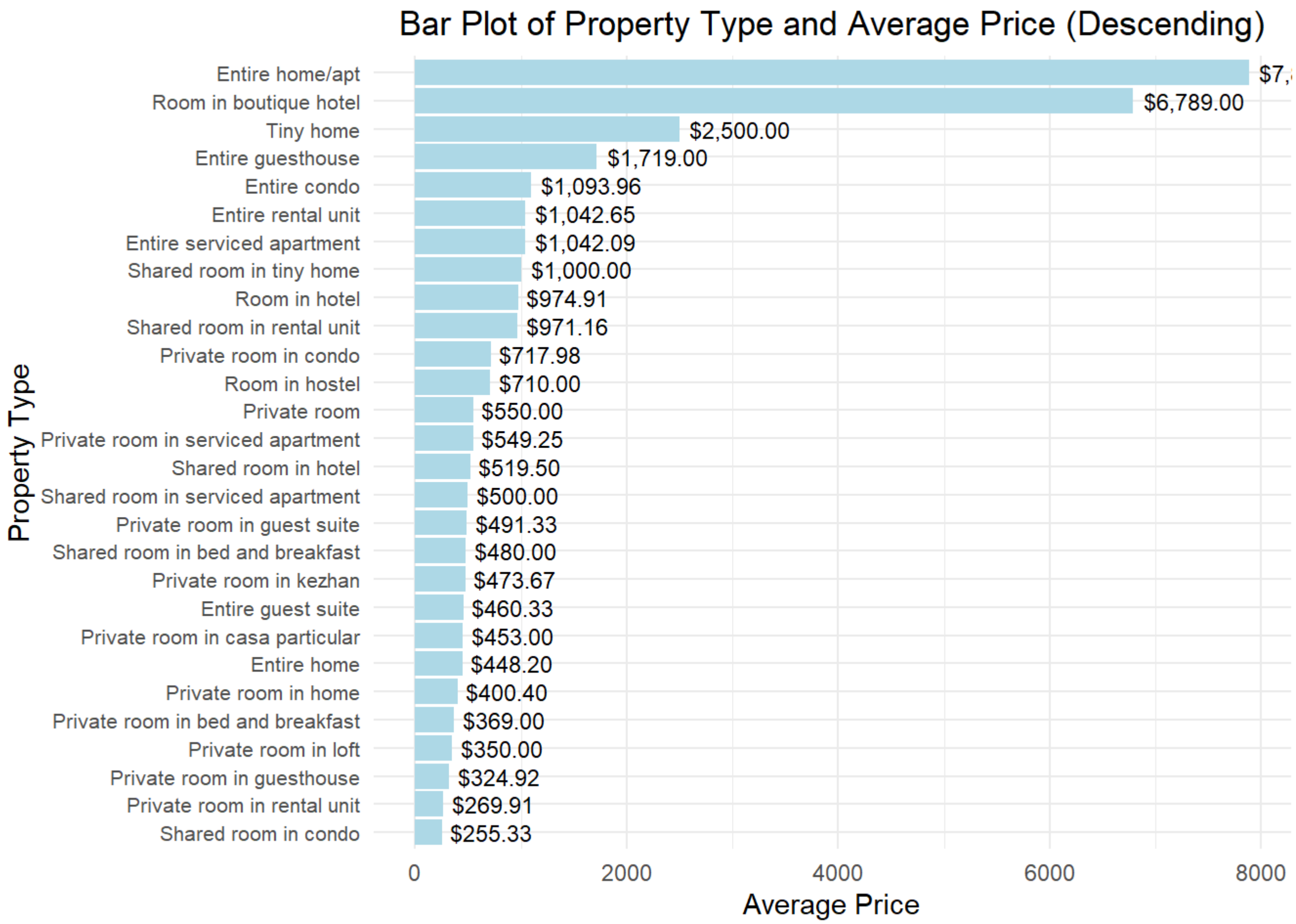
```
#understand the variable
summary_perroomtype <- df %>%
  group_by(room_type) %>%
  summarize(
    mean = mean(price),
    median = median(price),
    min = min(price),
    max = max(price),
    sd = sd(price)
  )
summary_perroomtype
```

```
## # A tibble: 4 × 6
##   room_type        mean median   min   max    sd
##   <chr>           <dbl>  <dbl> <dbl> <dbl> <dbl>
## 1 Entire home/apt 1063.    800   160  8000  948.
## 2 Hotel room       710     710   710   710   NA 
## 3 Private room     328.    180   110  6789  488.
## 4 Shared room      926.    219   120 50000 5340.
```

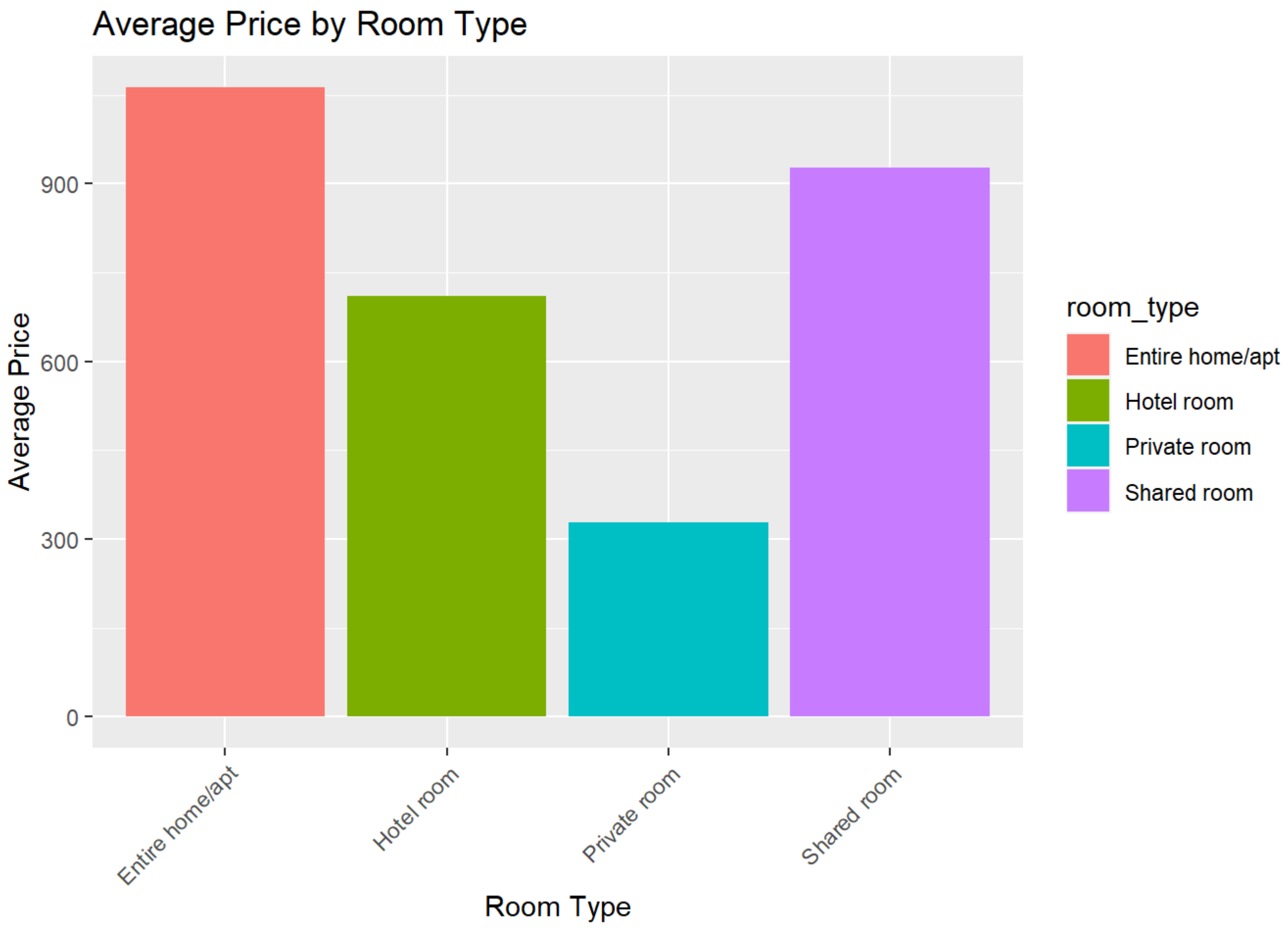
```
#1. histogram of price
# Calculate outlier boundaries using the IQR method
Q1 <- quantile(df$price, 0.25, na.rm = TRUE)
Q3 <- quantile(df$price, 0.75, na.rm = TRUE)
IQR <- Q3 - Q1
lower_bound <- Q1 - 1.5 * IQR
upper_bound <- Q3 + 1.5 * IQR

# Filter out outliers
df_filtered <- df %>%
  filter(price >= lower_bound, price <= upper_bound)

# Create the histogram plot with filtered data
df.histogram <- ggplot(df_filtered, aes(x = price)) +
  geom_histogram(bins = 40, color = "black", fill = "lightblue") +
  labs(x = "Price", y = "Frequency", title = "Histogram of Price (Without Outliers)")

# Print the plot
print(df.histogram)
```


Histogram of Price (Without Outliers)
Frequency
400
300
200
100
0
500
1000
Price

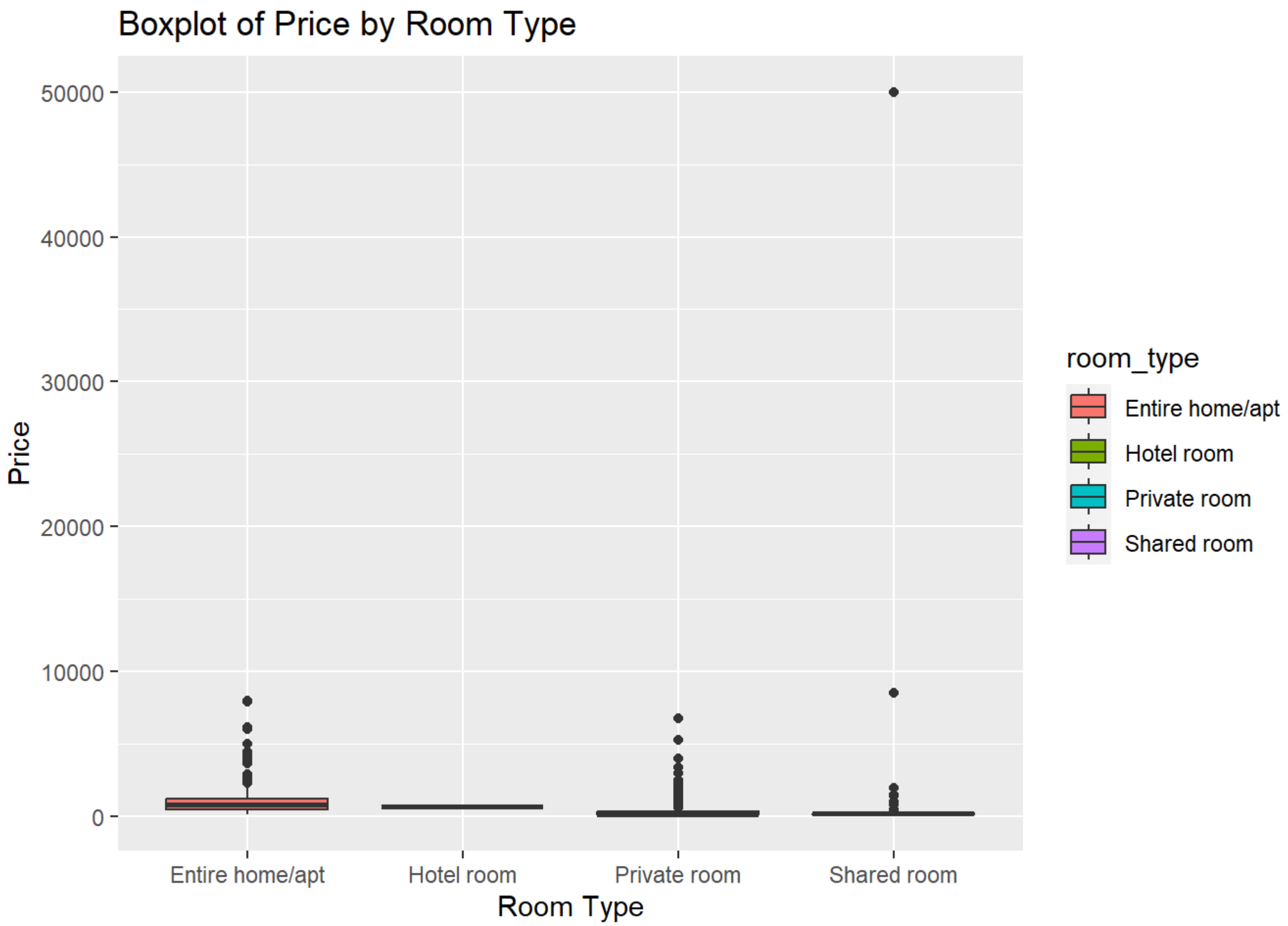
```r
# 2. Bar Plot of Property Type and Average Price (Descending)
# Filter and summarize the data
df.barplot <- df %>%
  group_by(property_type) %>%
  summarise(n = n(), avg_price = mean(price, na.rm = TRUE)) %>%
  arrange(desc(avg_price))

# Create the bar plot
df.barplot %>%
  ggplot(aes(y = reorder(property_type, avg_price), x = avg_price)) +
  geom_bar(stat = 'identity', fill = 'lightblue') +
  labs(title = "Bar Plot of Property Type and Average Price (Descending)",
       x = "Average Price",
       y = "Property Type") +
  theme_minimal() +
  theme(plot.title = element_text(hjust = 0.5),
        axis.text.y = element_text(size = 8)) +
  geom_text(aes(label = scales::dollar(avg_price, prefix = "$")), hjust = -0.1, size = 3, color = 'black')
```


Bar Plot of Property Type and Average Price (Descending)
Property Type
Average Price
Entire home/apt
$7,
Room in boutique hotel
$6,789.00
Tiny home
$2,500.00
Entire guesthouse
$1,719.00
Entire condo
$1,093.96
Entire rental unit
$1,042.65
Entire serviced apartment
$1,042.09
Shared room in tiny home
$1,000.00
Room in hotel
$974.91
Shared room in rental unit
$971.16
Private room in condo
$717.98
Room in hostel
$710.00
Private room
$550.00
Private room in serviced apartment
$549.25
Shared room in hotel
$519.50
Shared room in serviced apartment
$500.00
Private room in guest suite
$491.33
Shared room in bed and breakfast
$480.00
Private room in kezhan
$473.67
Entire guest suite
$460.33
Private room in casa particular
$453.00
Entire home
$448.20
Private room in home
$400.40
Private room in bed and breakfast
$369.00
Private room in loft
$350.00
Private room in guesthouse
$324.92
Private room in rental unit
$269.91
Shared room in condo
$255.33
0
2000
4000
6000
8000

```
# 3. bar chart of average prices by room type
# Calculate the average cost for each room type
avg_price_by_room <- df %>%
  group_by(room_type) %>%
  summarize(avg_price = mean(price, na.rm = TRUE))

# Create a grouped bar chart of average prices by room type
df.bar.roomtype <- ggplot(avg_price_by_room, aes(x = room_type, y = avg_price, fill = room_type)) +
  geom_bar(stat = "identity", position = "dodge") +
  labs(x = "Room Type", y = "Average Price", title = "Average Price by Room Type") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))  # Rotate x-axis labels for better readability

# Print the plot
print(df.bar.roomtype)
```



```
# 4. Boxplot of Price by Room Type (with outliers)
df.p_boxplot <- ggplot(df, aes(x = room_type, y = price, fill = room_type)) +
  geom_boxplot() +
  labs(x = "Room Type", y = "Price", title = "Boxplot of Price by Room Type")

# Print the plot
print(df.p_boxplot)
```



```
# seeing a clear outlier, need to be removed
```

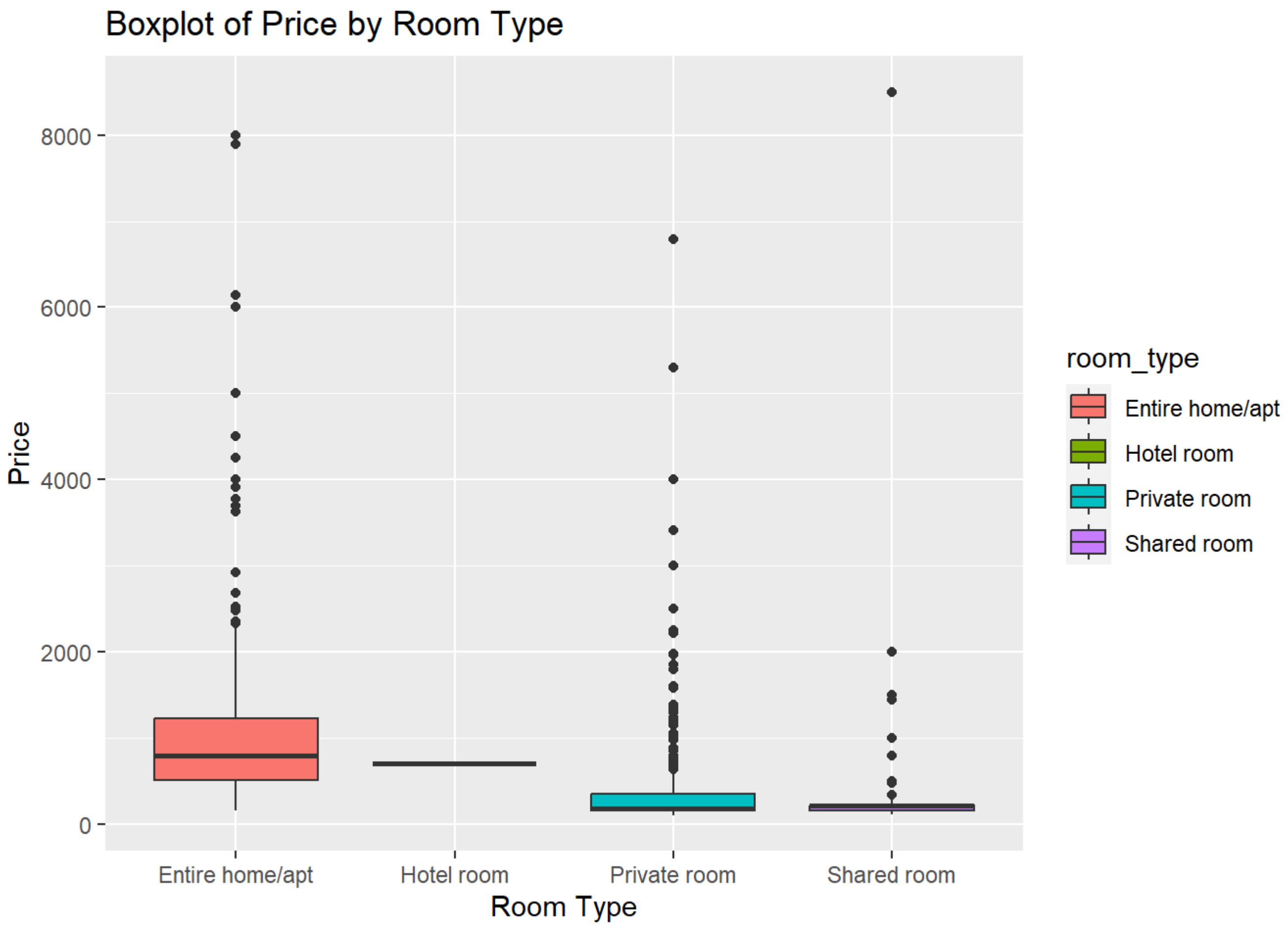
```
# Find the index of the outlier in the "shared_room" group
outlier_index <- which(df$room_type == "Shared room" & df$price > 40000)
outlier_index
```

```
## [1] 925
```

```r
# Remove the outlier row from the original dataset
df <- df[-outlier_index, ]

str(df) # only one observation is removed
```

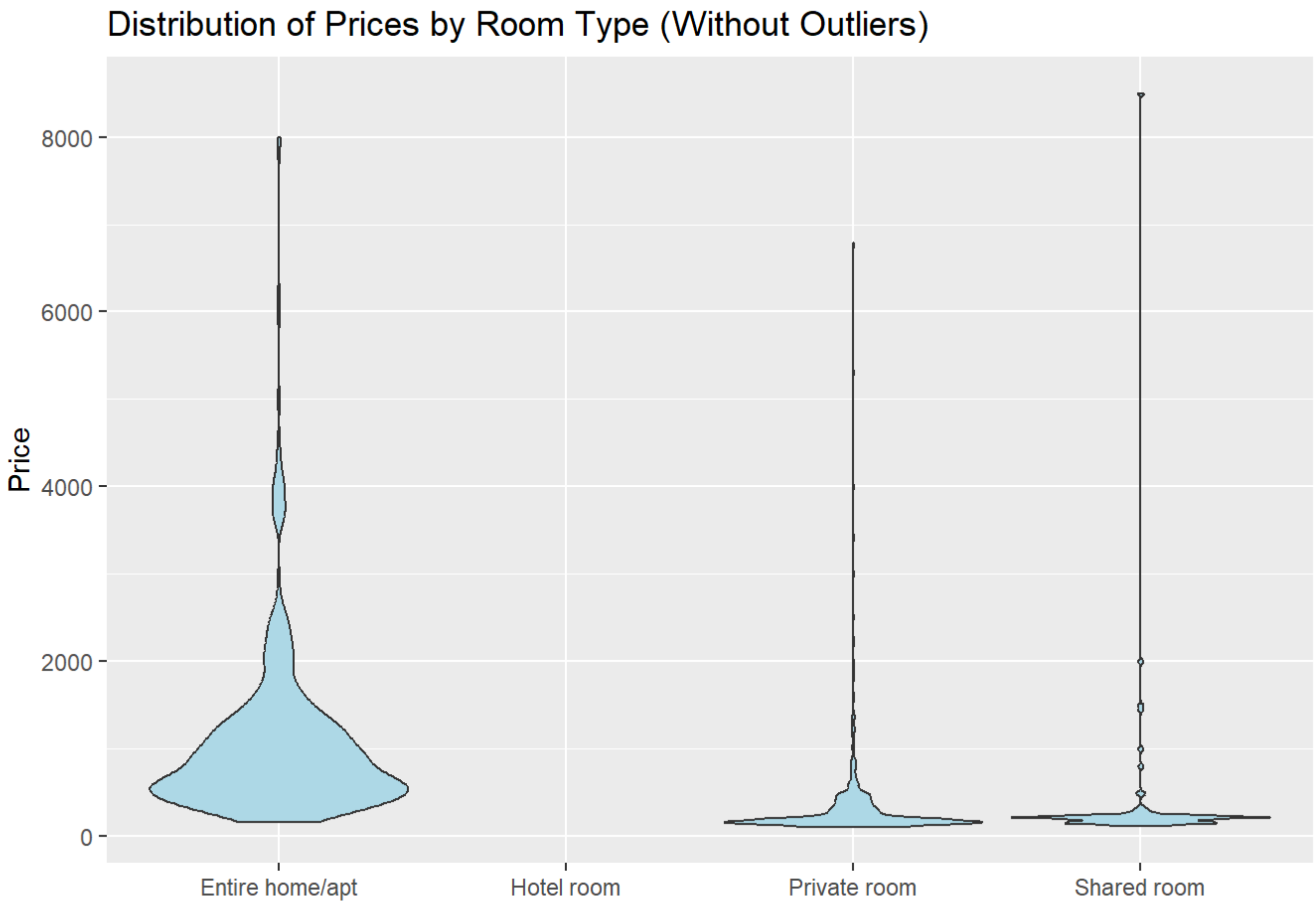
```
## tibble [1,423 × 50] (S3: tbl_df/tbl/data.frame)
##  $ ...1                                        : num [1:1423] 1 2 3 4 5 6 7 8 9 10 ...
##  $ description                                 : chr [1:1423] "Flat is very quiet. A/C in each room and living room." "*
共居時光*<br /><br />關於我們：<br />共居時光是一家現代化的服務式公寓公司，為您提供不同類型的舒適房間，從合租"| __truncated__ "near
MTR<br />8 rooms sharing apartment with a kitchen and washing machine<br />with private bathroom<br /><br "| __truncated__
"This listing is for 1 private room in a shared apartment.<br />If you like to meet new friends, expand your Net"| __truncat
ed__ ...
##  $ neighborhood_overview                       : chr [1:1423] "None" "None" "None" "None" ...
##  $ host_id                                     : num [1:1423] 1.25e+08 3.81e+07 7.52e+06 7.52e+06 1.20e+08 ...
##  $ host_since                                  : Date[1:1423], format: "2017-04-10" "2015-07-10" ...
##  $ host_location                               : chr [1:1423] "Hong Kong" "Hong Kong" "Hong Kong" "Hong Kong" ...
##  $ host_response_time                          : chr [1:1423] "within a few hours" "within a few hours" "within a few hou
rs" "within a few hours" ...
##  $ host_response_rate                          : num [1:1423] 1 1 0.98 0.98 0.33 0.94 0.94 1 1 1 ...
##  $ host_acceptance_rate                        : num [1:1423] 0.57 0.57 0.57 0.57 0 0.2 0.2 0.26 0.2 1 ...
##  $ host_is_superhost                           : logi [1:1423] FALSE FALSE FALSE FALSE FALSE FALSE ...
##  $ host_neighbourhood                          : chr [1:1423] "Wan Chai" "Causeway Bay" "Wan Chai" "Wan Chai" ...
##  $ host_listings_count                         : num [1:1423] 2 1 365 365 9 441 441 367 304 7 ...
##  $ host_total_listings_count                   : num [1:1423] 2 1 396 396 11 505 505 379 323 9 ...
##  $ host_verifications                          : chr [1:1423] "['email', 'phone', 'work_email']" "['email', 'phone']" "
['email', 'phone']" "['email', 'phone']" ...
##  $ host_has_profile_pic                        : logi [1:1423] TRUE TRUE TRUE TRUE TRUE TRUE ...
##  $ host_identity_verified                      : logi [1:1423] FALSE TRUE TRUE TRUE FALSE TRUE ...
##  $ latitude                                    : num [1:1423] 22.3 22.3 22.3 22.3 22.3 ...
##  $ longitude                                   : num [1:1423] 114 114 114 114 114 ...
##  $ property_type                               : chr [1:1423] "Entire rental unit" "Shared room in serviced apartment" "P
rivate room in rental unit" "Private room in rental unit" ...
##  $ room_type                                   : chr [1:1423] "Entire home/apt" "Shared room" "Private room" "Private roo
m" ...
##  $ accommodates                                : num [1:1423] 2 7 2 2 3 1 1 1 2 2 ...
##  $ bathroom_nb                                 : num [1:1423] 1.5 2 1 1 1 1 1 1 1 1 ...
##  $ bathroom_type                               : chr [1:1423] "normal" "shared" "normal" "private" ...
##  $ bedrooms                                    : num [1:1423] 2 7 1 1 1 1 1 1 1 1 ...
##  $ beds                                        : num [1:1423] 2 7 1 1 1 1 1 1 1 1 ...
##  $ amenities                                   : chr [1:1423] "[\"Wifi\", \"Kitchen\", \"Washer\", \"Air conditioning\",
\"Smoke alarm\"]" "[\"Hangers\", \"Wifi\", \"Hot water\", \"TV\", \"Shampoo\", \"Kitchen\", \"Washer\", \"Air conditioning
\", \"Elevator\"]" "[\"Iron\", \"Air conditioning\", \"Wifi\", \"Kitchen\"]" "[\"Iron\", \"Hangers\", \"Wifi\", \"Lock on be
droom door\", \"Kitchen\", \"Washer\", \"Air conditioning\"]" ...
##  $ price                                       : num [1:1423] 470 500 217 160 1150 180 180 180 160 950 ...
##  $ minimum_nights                              : num [1:1423] 30 30 29 29 30 30 30 29 29 28 ...
##  $ maximum_nights                              : num [1:1423] 43 365 1125 1125 366 ...
##  $ has_availability                            : logi [1:1423] FALSE TRUE TRUE TRUE TRUE TRUE ...
##  $ availability_30                             : num [1:1423] 0 23 29 29 0 30 30 29 29 0 ...
##  $ availability_60                             : num [1:1423] 0 53 59 59 0 60 60 59 59 0 ...
##  $ availability_90                             : num [1:1423] 0 83 89 89 0 90 90 89 89 0 ...
##  $ availability_365                            : num [1:1423] 0 83 364 364 0 365 365 364 364 0 ...
##  $ number_of_reviews                           : num [1:1423] 1 0 0 0 0 1 0 0 0 0 ...
##  $ number_of_reviews_ltm                       : num [1:1423] 0 0 0 0 0 0 0 0 0 0 ...
##  $ number_of_reviews_l30d                      : num [1:1423] 0 0 0 0 0 0 0 0 0 0 ...
##  $ review_scores_rating                        : num [1:1423] 5 0 0 0 0 4 0 0 0 0 ...
##  $ review_scores_accuracy                      : num [1:1423] 5 0 0 0 0 4 0 0 0 0 ...
##  $ review_scores_cleanliness                   : num [1:1423] 5 0 0 0 0 3 0 0 0 0 ...
##  $ review_scores_checkin                       : num [1:1423] 5 0 0 0 0 3 0 0 0 0 ...
##  $ review_scores_communication                 : num [1:1423] 5 0 0 0 0 5 0 0 0 0 ...
##  $ review_scores_location                      : num [1:1423] 5 0 0 0 0 5 0 0 0 0 ...
##  $ review_scores_value                         : num [1:1423] 5 0 0 0 0 3 0 0 0 0 ...
##  $ instant_bookable                            : logi [1:1423] FALSE FALSE FALSE FALSE FALSE FALSE ...
##  $ calculated_host_listings_count_entire_homes : num [1:1423] 1 0 15 15 5 13 13 18 26 6 ...
##  $ calculated_host_listings_count_private_rooms: num [1:1423] 0 0 322 322 2 384 384 332 265 0 ...
##  $ calculated_host_listings_count_shared_rooms : num [1:1423] 0 1 28 28 0 8 8 16 12 0 ...
##  $ reviews_per_month                           : num [1:1423] 0.05 0 0 0 0 0.01 0 0 0 0 ...
##  $ got_reviewed                                : num [1:1423] 1 0 0 0 0 1 0 0 0 0 ...
```

```
# update Boxplot of Price by Room Type (without outliers)
df.p_boxplot2 <- ggplot(df, aes(x = room_type, y = price, fill = room_type)) +
  geom_boxplot() +
  labs(x = "Room Type", y = "Price", title = "Boxplot of Price by Room Type")

# Print the plot
print(df.p_boxplot2)
```

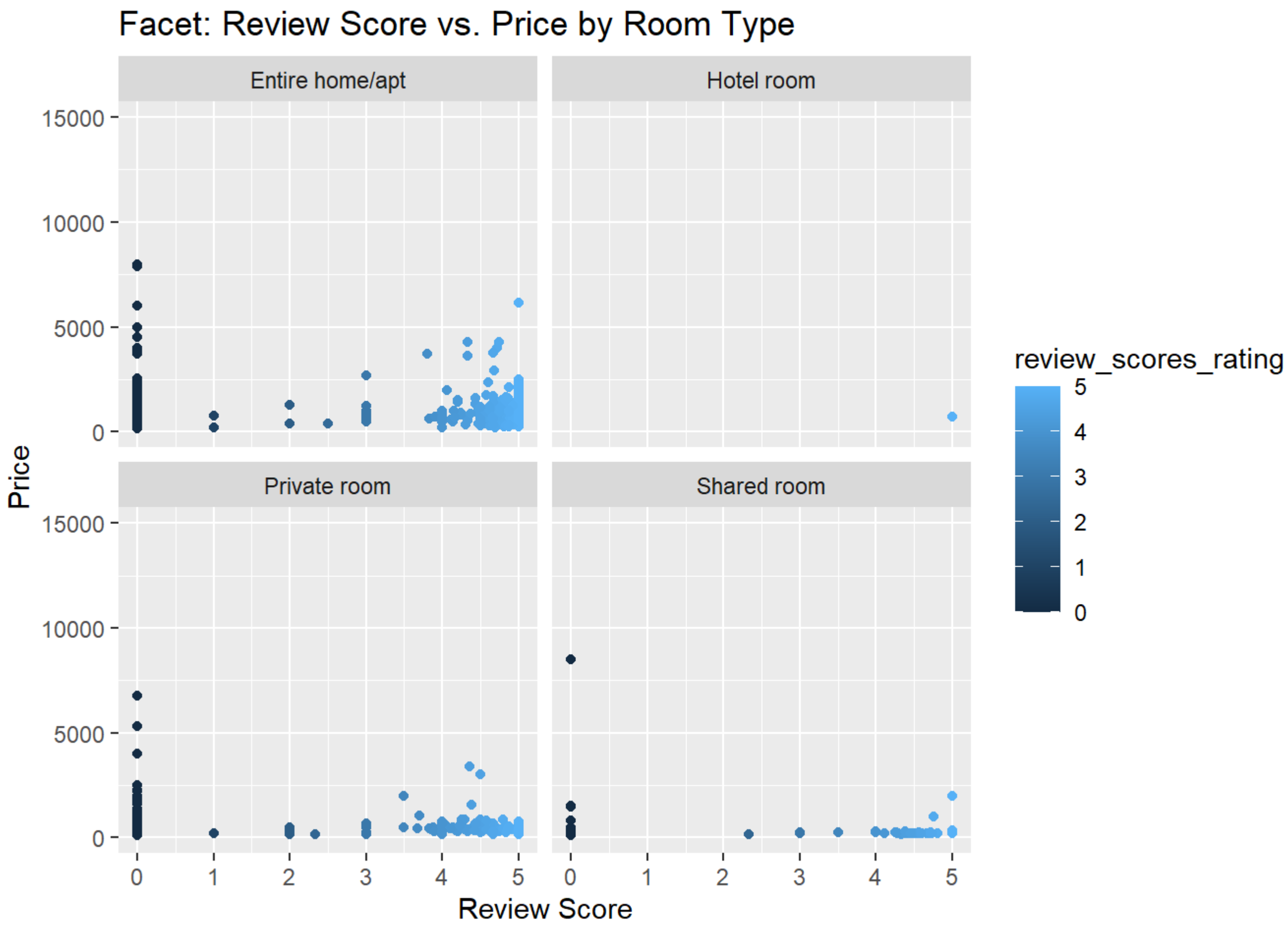


```
#5. Distribution of Prices by Room Type (Without Outliers)
# Create a violin plot of price by room type without outliers
plot_violin <- ggplot(df, aes(x = room_type, y = price)) +
  geom_violin(scale = "width", fill = "lightblue") +
  labs(x = "", y = "Price", title = "Distribution of Prices by Room Type (Without Outliers)")

# Print the violin plot
print(plot_violin)
```

```
## Warning: Groups with fewer than two data points have been dropped.
```



```
#6. facet-Scatter Plot: Review Score vs. Price by Room Type"
plot_facet <- ggplot(df, aes(x = review_scores_rating, y = price, color = review_scores_rating)) +
  geom_point() +
  labs(x = "Review Score", y = "Price", title = "Facet: Review Score vs. Price by Room Type") +
  coord_cartesian(ylim = c(0, 15000)) +  # Set x and y-axis limits
  facet_wrap(~ room_type, ncol = 2)  # Facet by room type, 2 columns

# Print the faceted scatter plot
print(plot_facet)
```



Upon analyzing the Histogram of Price, it becomes evident that a significant concentration of prices lies below 250. Across all property types, “Entire home/apt” listings command the highest average price, exceeding 7,000, followed by “Room in boutique hotel” at 6,700. The considerable difference between the top two and the subsequent “Tiny home” highlights a substantial pricing variation, with “Entire home/apt” prices approximately triple those of “Tiny home”. This fit our understanding of the HK market. The concentration of prices below 250 in the Histogram of Price reflects a common price range that attracts a significant portion of potential guests. The higher average price commanded by “Entire home/apt” and “boutique hotel room” listings suggests that these accommodations offer a premium experience and are positioned as a luxurious option. The detailed categorization of property types indicates the diversity in property types, and the different level of average price highlights the range of options available to travelers, from opulent entire homes to more budget-friendly tiny homes.

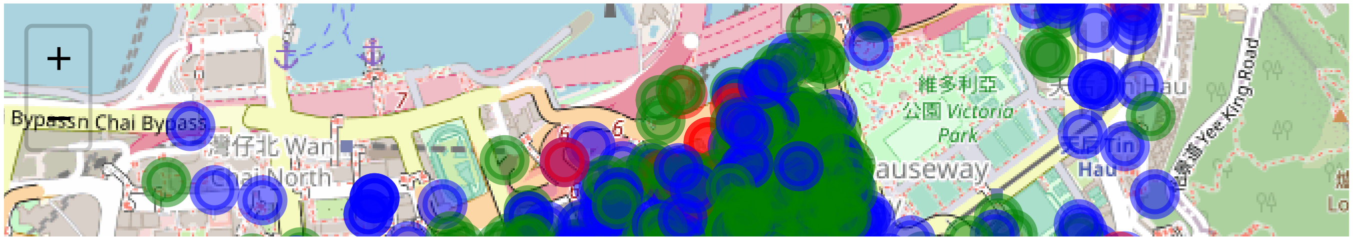
Moreover, the box plots and violin distribution charts reveal distinctions in price quartiles and outlier presence across various room types. Notably, “Entire home/apt” listings exhibit both the highest average price and the widest price range, with the lower quartile of “Entire home/apt” surpassing the upper quartiles of “Private room” and “Shared room.” Private and shared rooms exhibit smaller price ranges, with the presence of notable outliers, particularly in private rooms. The shape of the violin distribution appears similar between “Private room” and “Shared room,” with shared rooms displaying a more centralized range alongside more extreme outliers. We’d imagine the wide price range and presence of outliers in “Entire home/apt” listings could be attributed to the uniqueness and luxury associated with these accommodations.

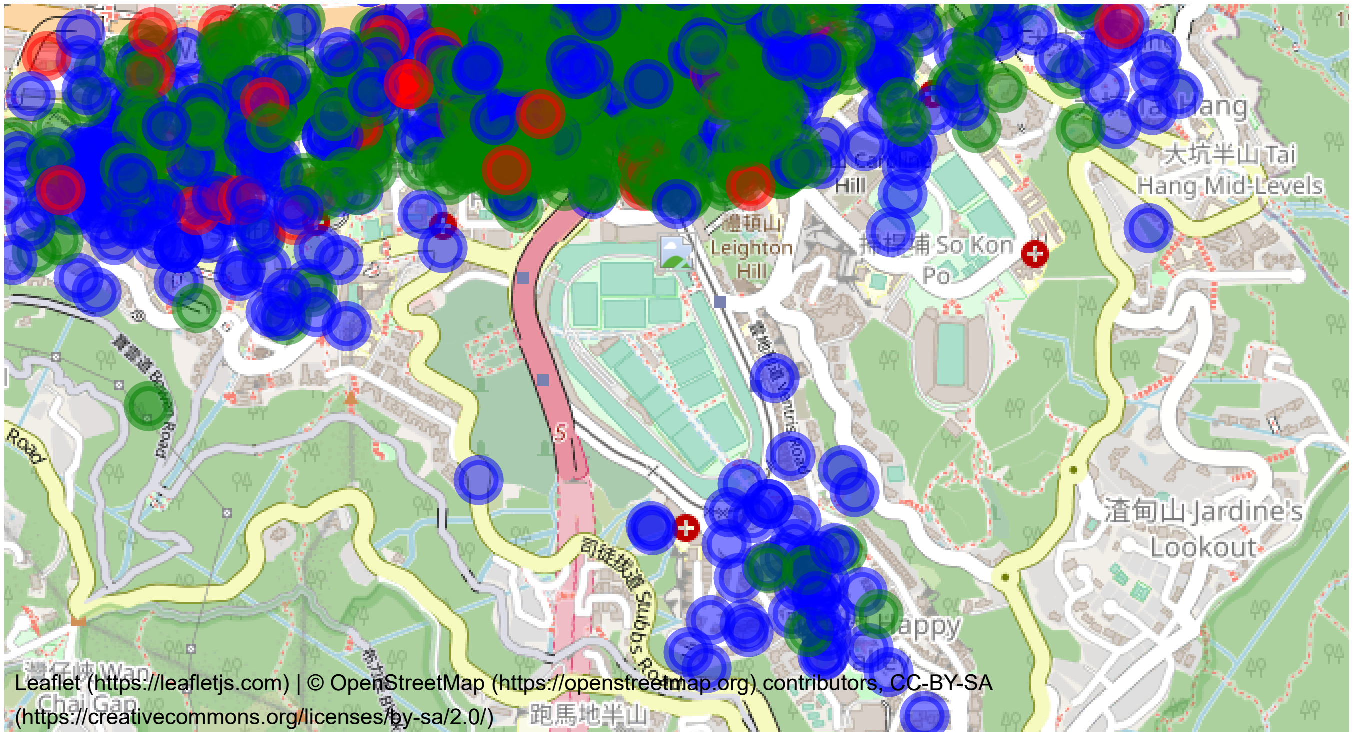
Interestingly, based on the facet plot, review scores do not exhibit a strong correlation with high prices, as lower-priced properties can still achieve high guest satisfaction. This observation suggests that factors beyond pricing significantly influence guest experiences, underscoring the intricate interplay between pricing and guest sentiment. This could lead to consideration on the significance of factors such as cleanliness, communication, and overall experience in shaping guest reviews in the HK market.

```
#IV. Mapping
library(leaflet)
```

```
## Warning: package 'leaflet' was built under R version 4.2.3
```

```
# Create a custom icon
custom_icon <- makeIcon(iconUrl = "custom_icon.png", iconWidth = 20, iconHeight = 20)

# Create a map with colored circle markers based on room types and custom icon
neighborhood_map2 <- leaflet(data = df) %>%
  addTiles() %>%
  setView(lng = mean(df$longitude), lat = mean(df$latitude), zoom = 15) %>%
  addCircleMarkers(lng = df$longitude, lat = df$latitude,
                   color = ~ifelse(room_type == "Entire home/apt", "blue",
                                   ifelse(room_type == "Private room", "green", "red")),
                   fillOpacity = 0.5,
                   popup = paste("Room Type: ", df$room_type)) %>%
  addMarkers(lng = mean(df$longitude), lat = mean(df$latitude),
             icon = custom_icon)

# Print the map
neighborhood_map2
```





Leaflet (https://leafletjs.com) | © OpenStreetMap (https://openstreetmap.org) contributors, CC-BY-SA (https://creativecommons.org/licenses/by-sa/2.0/)

Wan Chai is a district situated on the northern shore of Hong Kong Island, nestled between the Central and Causeway Bay districts. It features a blend of commercial, residential, and entertainment zones, with accommodations predominantly clustered in specific regions. The color-coded room type distribution indicates that Private rooms are concentrated in the northern expanse of Wan Chai, with a few near the HK Cricket Club, offering convenient access to both the northern and southern parts of the district, including local attractions and activities. This arrangement is likely influenced by the proximity to renowned attractions and activities. On the other hand, Entire house/apartment accommodations are positioned more towards the outskirts of Wan Chai, possibly due to their spatial requirements, while still maintaining proximity to the town center. The distribution of Shared room types disperse throughout the area.

```
#V. Wordcloud
library(tidytext)
```

```
## Warning: package 'tidytext' was built under R version 4.2.3
```

```
library(tm)
```

```
## Warning: package 'tm' was built under R version 4.2.3
```

```
## Loading required package: NLP
```

```
##
## Attaching package: 'NLP'
```

```
## The following object is masked from 'package:ggplot2':
##
##     annotate
```

```
neigh_over <- df %>% select(neighborhood_overview)
custom_stop_words <- bind_rows(stop_words,
                               data_frame(word = tm::stopwords("english"),
                                          lexicon = "custom"))
```

```
## Warning: `data_frame()` was deprecated in tibble 1.1.0.
## ℹ Please use `tibble()` instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```

```
tidy_neigh <- neigh_over %>% unnest_tokens(word, neighborhood_overview)
View(tidy_neigh)
tidy_neigh <- tidy_neigh %>% anti_join(custom_stop_words)
```

```
## Joining with `by = join_by(word)`
```

```
num_tidy <- tidy_neigh %>% count(word, sort = TRUE) %>%
  top_n(10)
```

```
## Selecting by n
```

```
num_tidy
```

```
## # A tibble: 11 × 2
##    word            n
##    <chr>       <int>
##  1 br            791
##  2 wan           230
##  3 restaurants   225
##  4 chai          218
##  5 causeway      189
##  6 bay           173
##  7 hong          157
##  8 local         146
##  9 shopping      145
## 10 shops         143
## 11 street        143
```

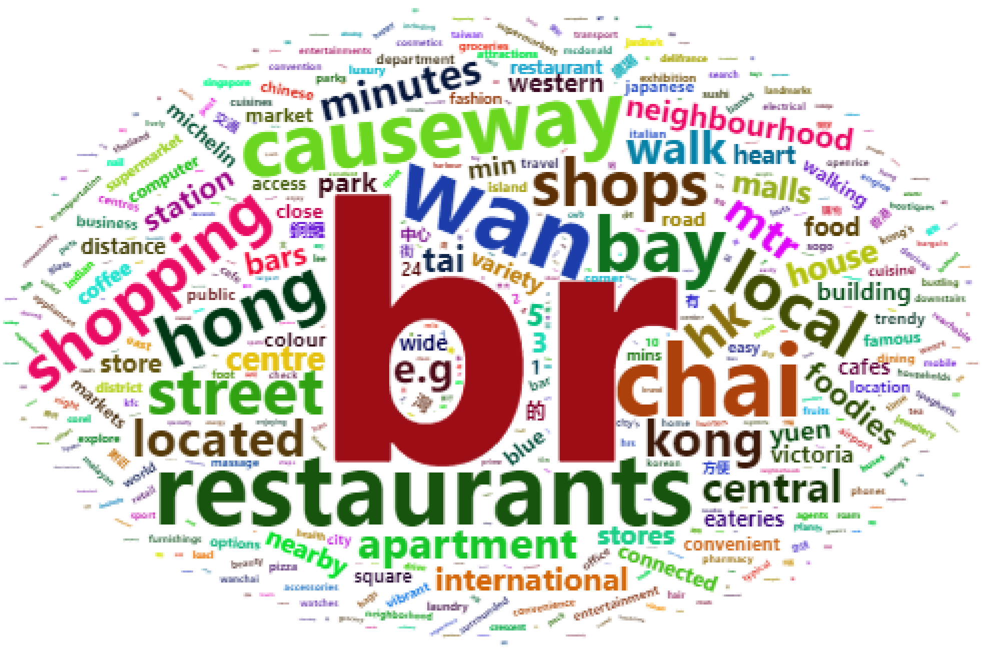
```
tidy_text <- neigh_over %>% unnest_tokens(bigram, neighborhood_overview,
                                          token = "ngrams", n = 2)
tidy_neigh_bigrams <- tidy_text %>%
  separate(bigram,c("word1","word2"), sep = " ") %>%
  filter(!word1 %in% custom_stop_words$word) %>%
  filter(!word2 %in% custom_stop_words$word) %>%
  count(word1, word2, sort = TRUE) %>%
  unite(bigram, word1, word2, sep = " ")
tidy_neigh_bigrams
```

```
## # A tibble: 2,856 × 2
##    bigram               n
##    <chr>            <int>
##  1 NA NA             1000
##  2 wan chai           216
##  3 br br              188
##  4 causeway bay       171
##  5 hong kong          110
##  6 tai yuen            59
##  7 yuen street         59
##  8 central causeway    50
##  9 blue house          48
## 10 colour e.g          46
## # ℹ 2,846 more rows
```

```r
library(wordcloud2)
```

```
## Warning: package 'wordcloud2' was built under R version 4.2.3
```

```r
#word cloud for unigrams
wordcloud2(tidy_neigh %>% count(word, sort = TRUE) %>% filter(n>5))
```



```
#word cloud for bigrams
wordcloud2(tidy_neigh_bigrams %>% filter(n<240))
```

In the first diagram, several words are visually prominent due to varying sizes. Notably, “br”, “Chai”, “Wan,” “restaurants,” “shopping” “centre”,“Causeway,” and “foodies” are highlighted. The larger appearance of “restaurants” implies its heightened significance within the context. The frequent mentions of “restaurants” and “shops” underscore the commercial dimension of Wan Chai, signifying a diverse range of dining and shopping options.

The second diagram showcases keywords such as “wan chai”, “hong kong,” “br br”, references to attractions in Chinese, “Michelin restaurants”, and specific street names like “Yuen Street”. The repetition of “br br” likely results from formatting or parsing issues from the web scrapping. The prominence of “Wan chai” and “hong kong” accentuates its central role - location indication. The allusions to attractions in Chinese point to cultural and tourist highlights that contribute to the area’s allure. Furthermore, the inclusion of specific streets and stations emphasizes the convenience and accessibility of Wan Chai, portraying it as a well-connected district with a focus on local attractions and its close proximity to transportation hubs.