BUSN 41201 Initial Final

Group 11: Yu-Ting Weng, Mengdi Hao, Elena Li, Minji Park, Sarah Lee

Load the Data

- Summary statistics for numerical attributes

```
##
## Summary statistics for numerical attributes:
```

```
##        id             log_price      accommodates      bathrooms
##  Min.   :     344   Min.   :0.000   Min.   : 1.000   Min.   :0.000
##  1st Qu.: 6261964   1st Qu.:4.317   1st Qu.: 2.000   1st Qu.:1.000
##  Median :12254147   Median :4.710   Median : 2.000   Median :1.000
##  Mean   :11266617   Mean   :4.782   Mean   : 3.155   Mean   :1.235
##  3rd Qu.:16402260   3rd Qu.:5.220   3rd Qu.: 4.000   3rd Qu.:1.000
##  Max.   :21230903   Max.   :7.600   Max.   :16.000   Max.   :8.000
##                                                      NA's   :200
##     latitude       longitude       number_of_reviews review_scores_rating
##  Min.   :33.34   Min.   :-122.51   Min.   :  0.0     Min.   : 20.00
##  1st Qu.:34.13   1st Qu.:-118.34   1st Qu.:  1.0     1st Qu.: 92.00
##  Median :40.66   Median : -77.00   Median :  6.0     Median : 96.00
##  Mean   :38.45   Mean   : -92.40   Mean   : 20.9     Mean   : 94.07
##  3rd Qu.:40.75   3rd Qu.: -73.95   3rd Qu.: 23.0     3rd Qu.:100.00
##  Max.   :42.39   Max.   : -70.99   Max.   :605.0     Max.   :100.00
##                                                      NA's   :16722
##     bedrooms           beds
##  Min.   : 0.000   Min.   : 0.000
##  1st Qu.: 1.000   1st Qu.: 1.000
##  Median : 1.000   Median : 1.000
##  Mean   : 1.266   Mean   : 1.711
##  3rd Qu.: 1.000   3rd Qu.: 2.000
##  Max.   :10.000   Max.   :18.000
##  NA's   :91       NA's   :131
```

- Convert to factors for categorical variables
- Summary statistics for factors

```
##
## Summary statistics for categorical attributes:
```

```
##      property_type             room_type              bed_type
##  Apartment  :49003   Entire home/apt:41310   Airbed       :  477
##  House      :16511   Private room   :30638   Couch        :  268
##  Condominium: 2658   Shared room    : 2163   Futon        :  753
##  Townhouse  : 1692                           Pull-out Sofa:  585
##  Loft       : 1244                           Real Bed     :72028
##  Other      :  607
##  (Other)    : 2396
```

```
##       cancellation_policy      city       instant_bookable
##  flexible       :22545    Boston : 3468   f:54660
##  moderate       :19063    Chicago: 3719   t:19451
##  strict         :32374    DC     : 5688
##  super_strict_30:  112    LA     :22453
##  super_strict_60:   17    NYC    :32349
##                           SF     : 6434
##
```

Missing Value Imputation

- Check the number of missing value

```
##                  variable missing_count
## 1      host_response_rate         18299
## 2    review_scores_rating         16722
## 3            first_review         15864
## 4             last_review         15827
## 5           thumbnail_url          8216
## 6           neighbourhood          6872
## 7                 zipcode           966
## 8              bathrooms           200
## 9    host_has_profile_pic           188
## 10 host_identity_verified           188
## 11             host_since           188
## 12                   beds           131
## 13               bedrooms            91
## 14                     id             0
## 15              log_price             0
## 16          property_type             0
## 17              room_type             0
## 18              amenities             0
## 19           accommodates             0
## 20               bed_type             0
## 21    cancellation_policy             0
## 22           cleaning_fee             0
## 23                   city             0
## 24            description             0
## 25       instant_bookable             0
## 26               latitude             0
## 27              longitude             0
## 28                   name             0
## 29      number_of_reviews             0
```

```
## Updated summary statistics for numerical attributes:
```

```
##        id             log_price      accommodates      bathrooms
##  Min.   :     344   Min.   :0.000   Min.   : 1.000   Min.   :0.000
##  1st Qu.: 6261964   1st Qu.:4.317   1st Qu.: 2.000   1st Qu.:1.000
##  Median :12254147   Median :4.710   Median : 2.000   Median :1.000
##  Mean   :11266617   Mean   :4.782   Mean   : 3.155   Mean   :1.235
##  3rd Qu.:16402260   3rd Qu.:5.220   3rd Qu.: 4.000   3rd Qu.:1.000
##  Max.   :21230903   Max.   :7.600   Max.   :16.000   Max.   :8.000
##  host_response_rate    latitude       longitude       number_of_reviews
##  Min.   :  0.00     Min.   :33.34   Min.   :-122.51   Min.   :  0.0
##  1st Qu.:100.00     1st Qu.:34.13   1st Qu.:-118.34   1st Qu.:  1.0
```

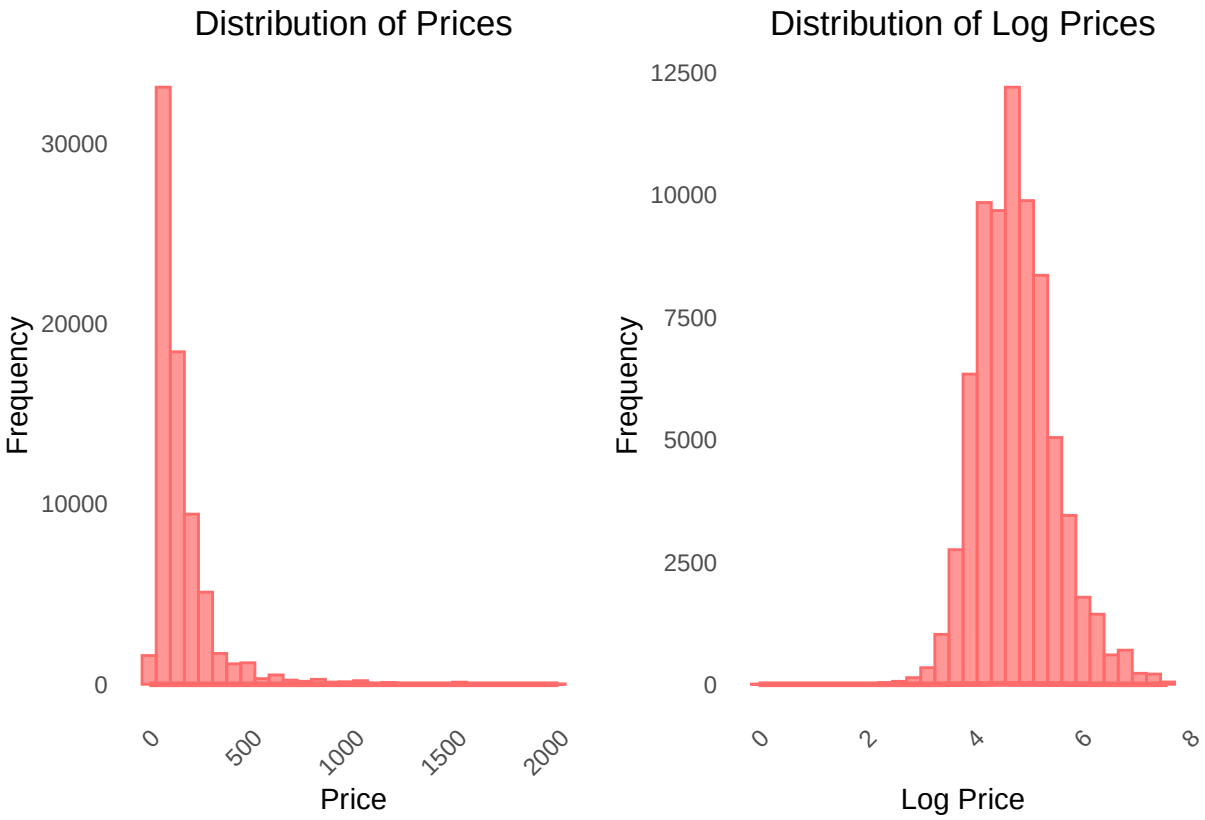
```
##  Median :100.00     Median :40.66   Median : -77.00   Median :  6.0
##  Mean   : 95.75     Mean   :38.45   Mean   : -92.40   Mean   : 20.9
##  3rd Qu.:100.00     3rd Qu.:40.75   3rd Qu.: -73.95   3rd Qu.: 23.0
##  Max.   :100.00     Max.   :42.39   Max.   : -70.99   Max.   :605.0
##  review_scores_rating    bedrooms          beds
##  Min.   : 20.0        Min.   : 0.000   Min.   : 0.00
##  1st Qu.: 93.0        1st Qu.: 1.000   1st Qu.: 1.00
##  Median : 96.0        Median : 1.000   Median : 1.00
##  Mean   : 94.5        Mean   : 1.265   Mean   : 1.71
##  3rd Qu.: 99.0        3rd Qu.: 1.000   3rd Qu.: 2.00
##  Max.   :100.0        Max.   :10.000   Max.   :18.00
```

```
##
## Updated summary statistics for categorical attributes:
```

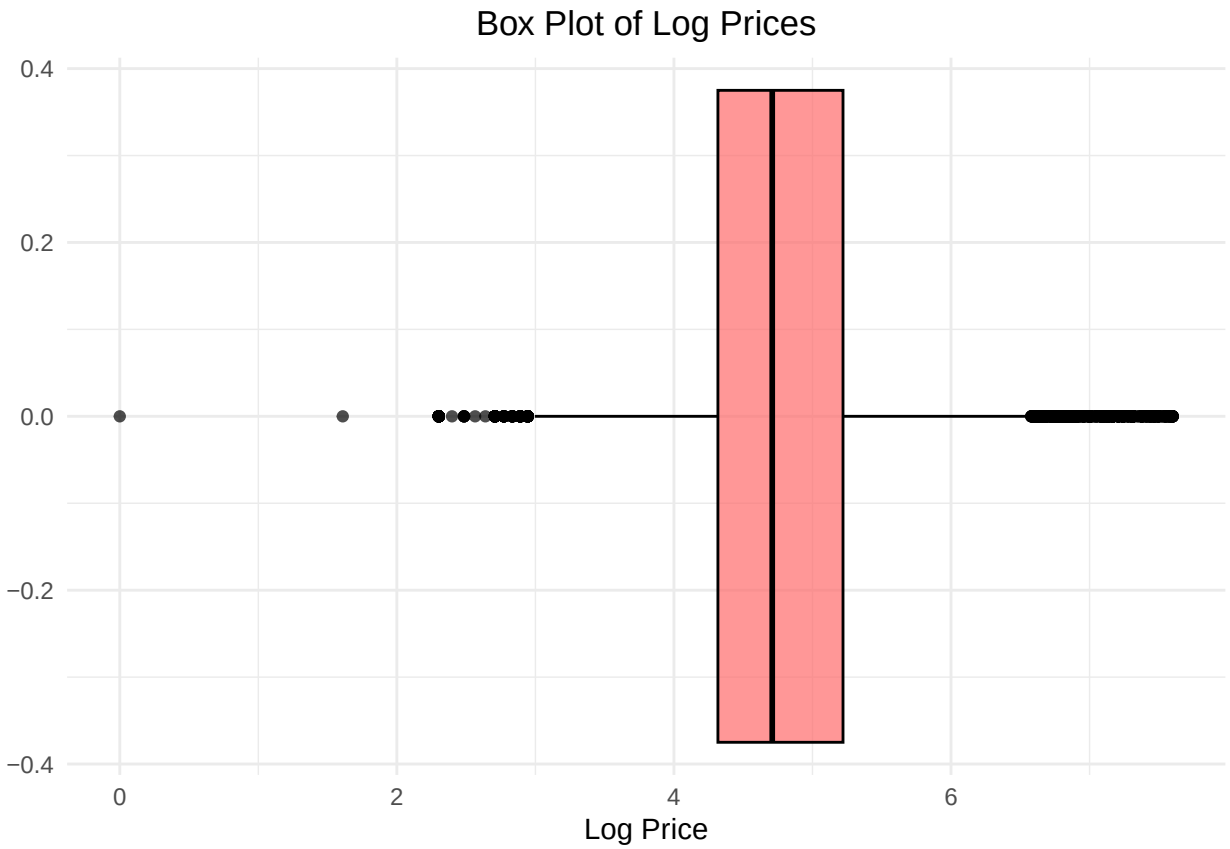
```
##      property_type             room_type              bed_type
##  Apartment  :49003   Entire home/apt:41310   Airbed       :  477
##  House      :16511   Private room   :30638   Couch        :  268
##  Condominium: 2658   Shared room    : 2163   Futon        :  753
##  Townhouse  : 1692                           Pull-out Sofa:  585
##  Loft       : 1244                           Real Bed     :72028
##  Other      :  607
##  (Other)    : 2396
##       cancellation_policy      city           first_review
##  flexible       :22545    Boston : 3468   Unknown   :15864
##  moderate       :19063    Chicago: 3719   2017-01-01:  293
##  strict         :32374    DC     : 5688   2017-01-22:  249
##  super_strict_30:  112    LA     :22453   2016-01-02:  221
##  super_strict_60:   17    NYC    :32349   2017-01-02:  211
##                           SF     : 6434   2017-09-04:  193
##                                           (Other)   :57080
##  host_has_profile_pic host_identity_verified      host_since
##  f:  414              f:24363                2015-03-30:  246
##  t:73697              t:49748                Unknown   :  188
##                                              2014-02-14:  173
##                                              2015-05-18:   83
##                                              2016-09-16:   83
##                                              2015-05-11:   82
##                                              (Other)   :73256
##  instant_bookable     last_review               neighbourhood      zipcode
##  f:54660          Unknown   :15827   Unknown           : 6872   11211.0: 1368
##  t:19451          2017-04-30: 1344   Williamsburg      : 2862   90291  : 1274
##                   2017-09-24: 1278   Bedford-Stuyvesant: 2166   11221  : 1188
##                   2017-09-17: 1215   Bushwick          : 1601   94110  :  988
##                   2017-04-23: 1025   Upper West Side   : 1396   90046  :  967
##                   2017-09-18:  832   Mid-Wilshire      : 1392   Unknown:  966
##                   (Other)   :52590   (Other)           :57822   (Other):67360
```

Data Visualization

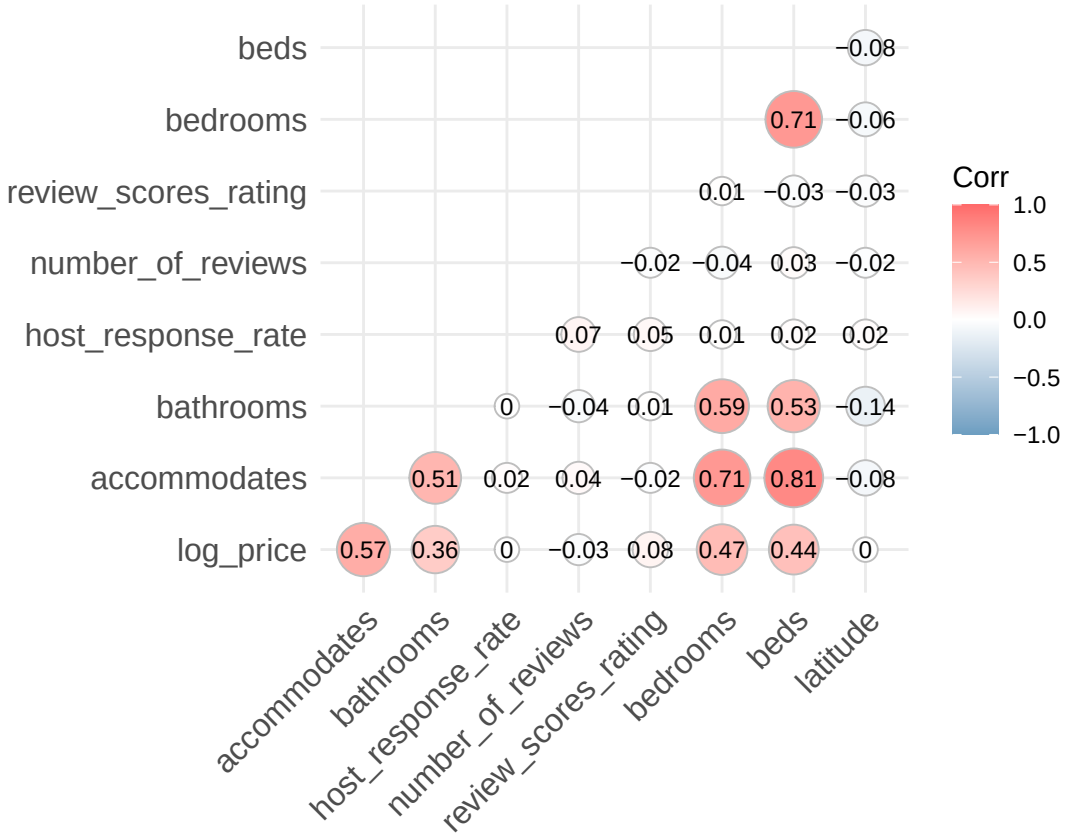
- Histogram of log prices



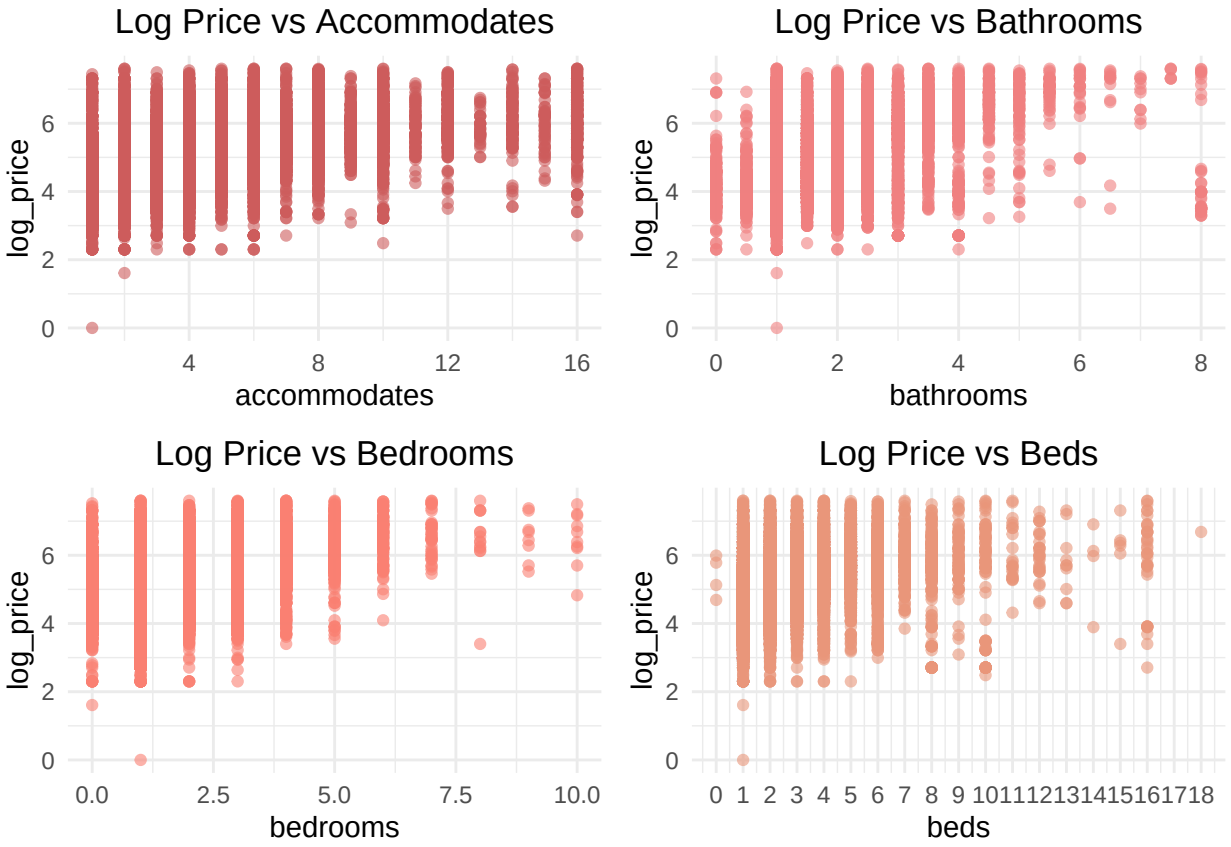
- Box plot of log price



- Correlation plot for numerical columns

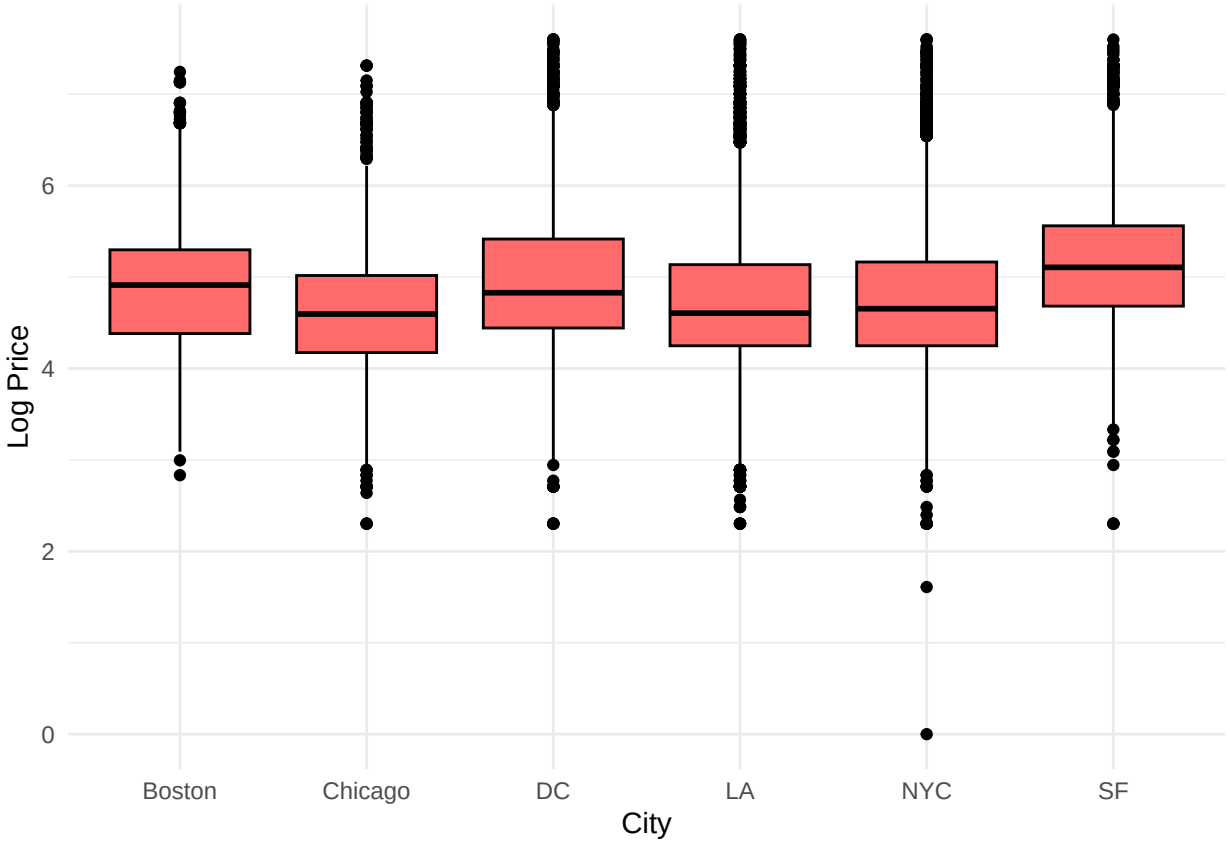


- Scatter plot for log price vs numerical property features

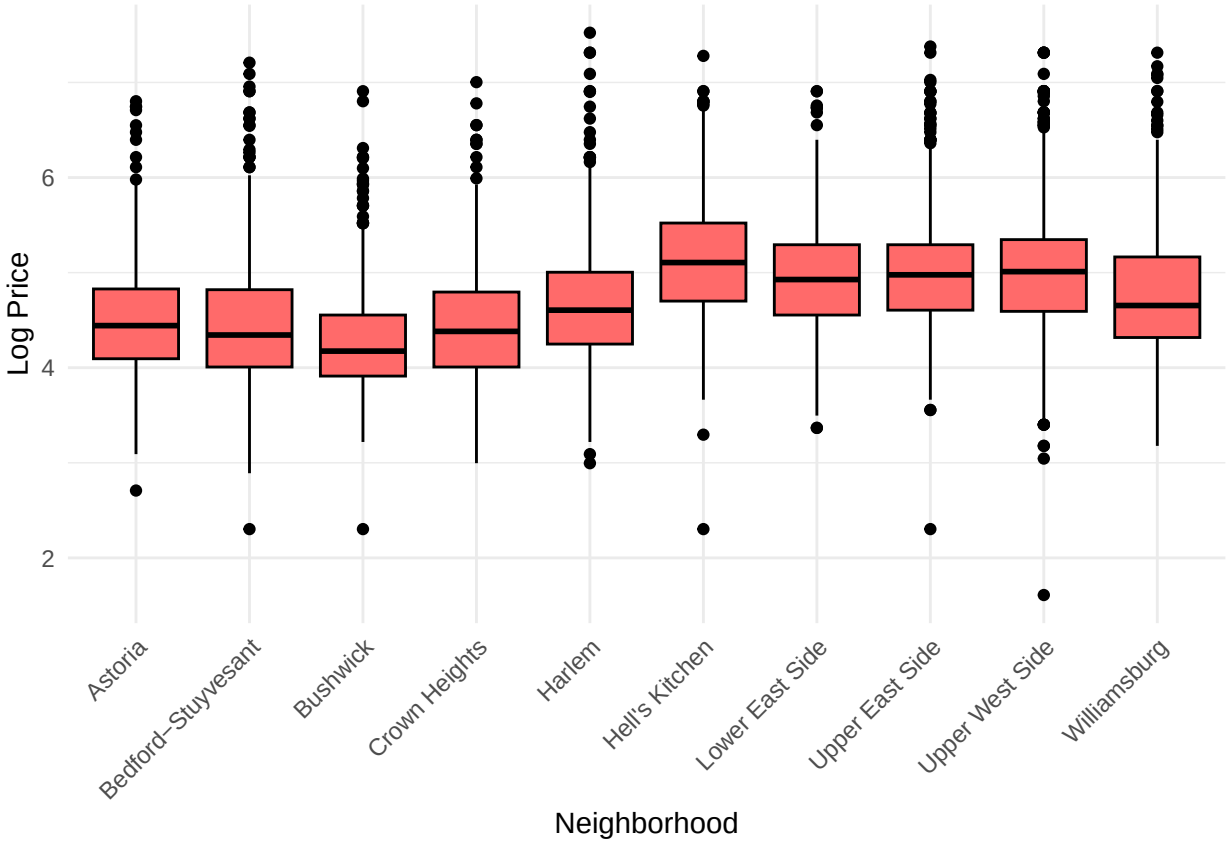



```
## TableGrob (2 x 2) "arrange": 4 grobs
##   z     cells    name           grob
## 1 1 (1-1,1-1) arrange gtable[layout]
## 2 2 (1-1,2-2) arrange gtable[layout]
## 3 3 (2-2,1-1) arrange gtable[layout]
## 4 4 (2-2,2-2) arrange gtable[layout]
```

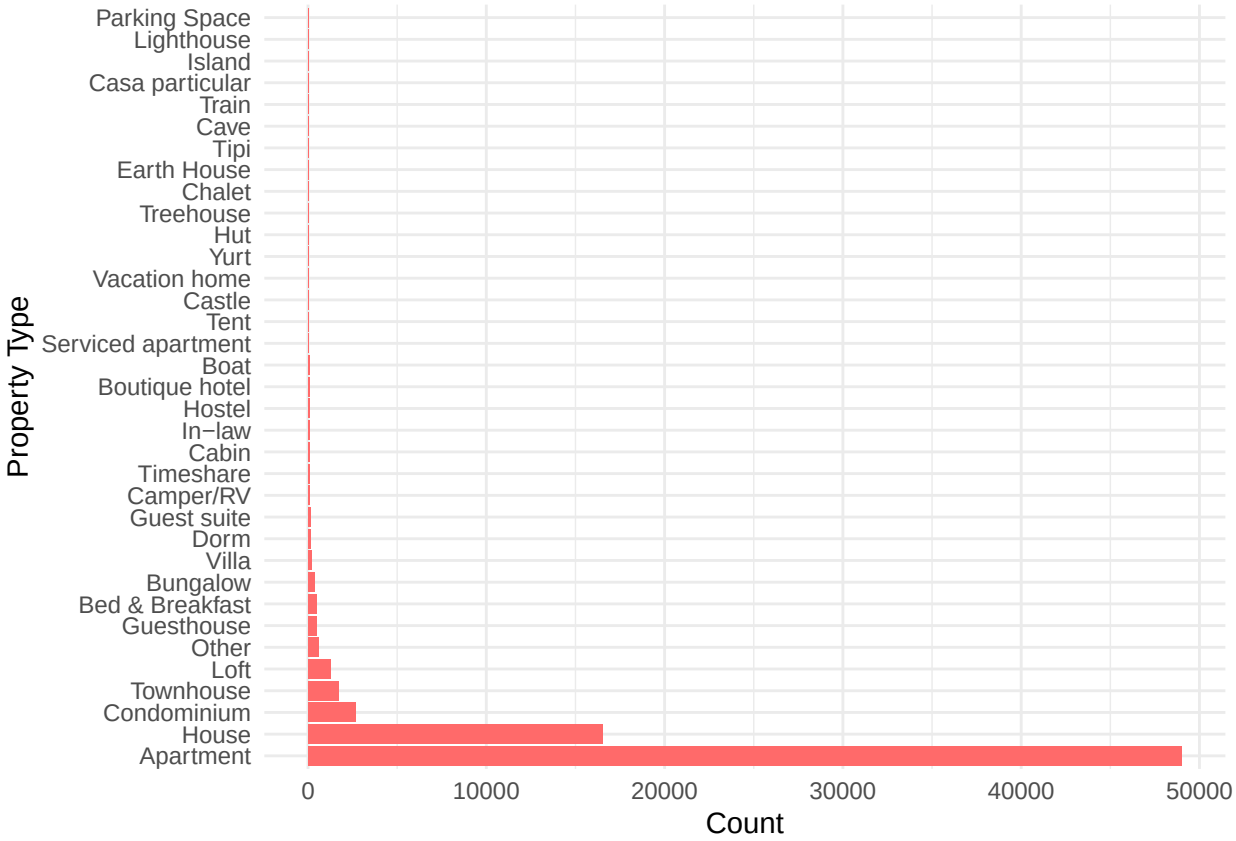
- Log Price Distribution Across Different 6 Cities



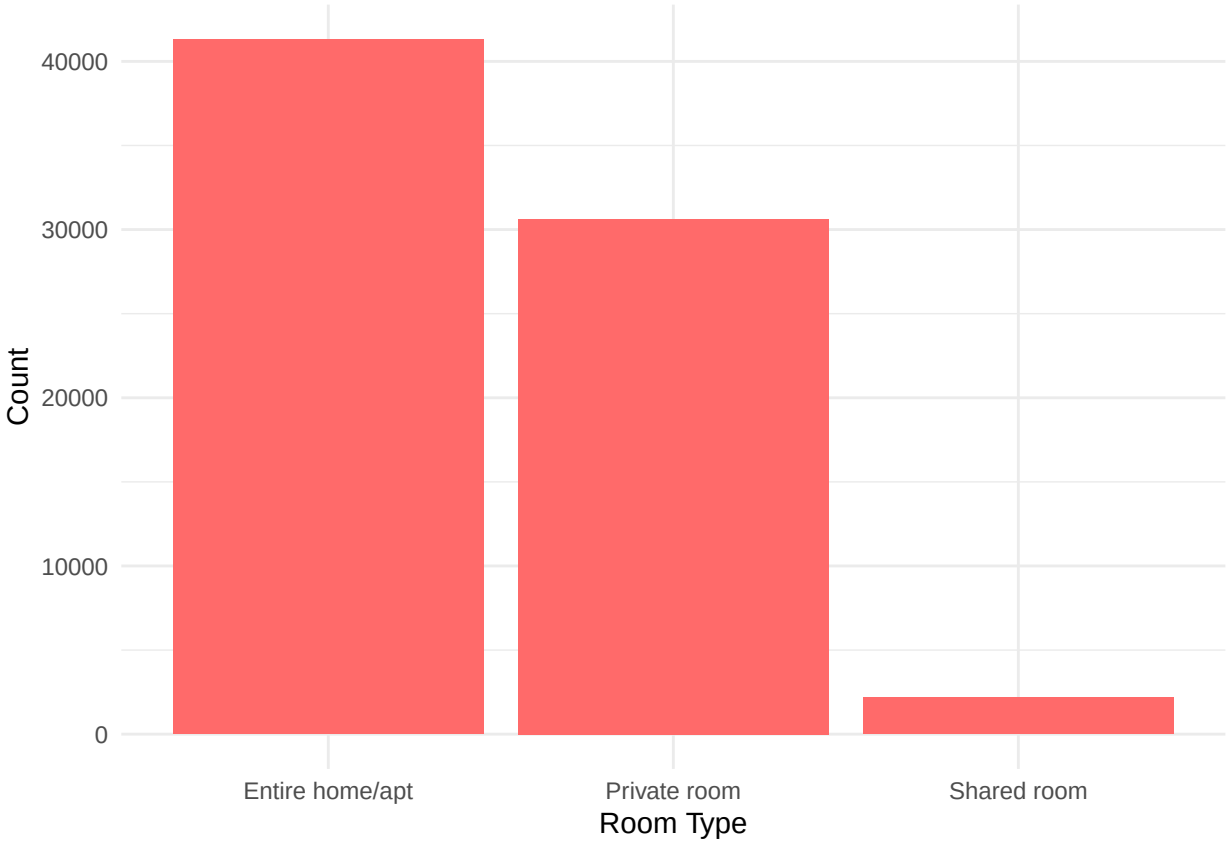
- Log Price Distribution Across Top 10 Neighborhoods in NYC



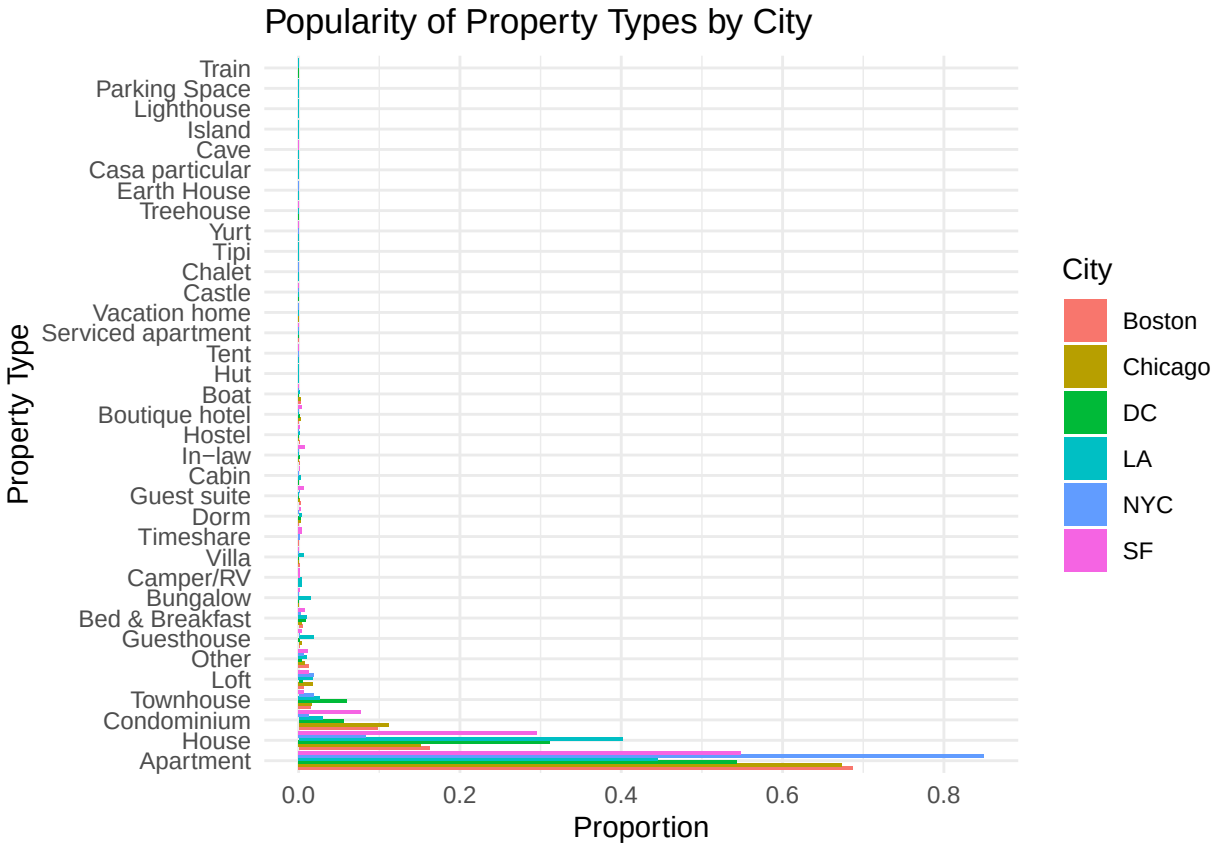
- Distribution of Property Types



- Distribution of Room Types

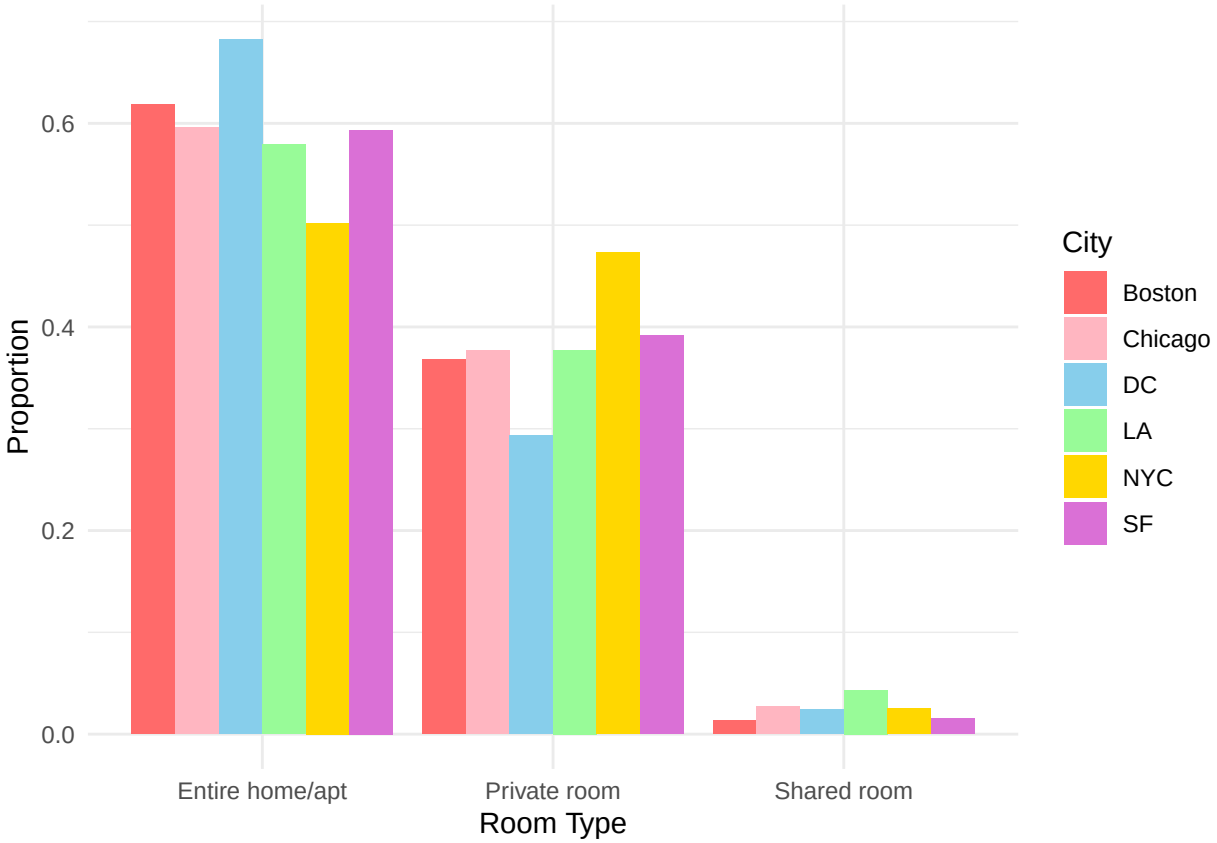


- Popularity of Property Types by City

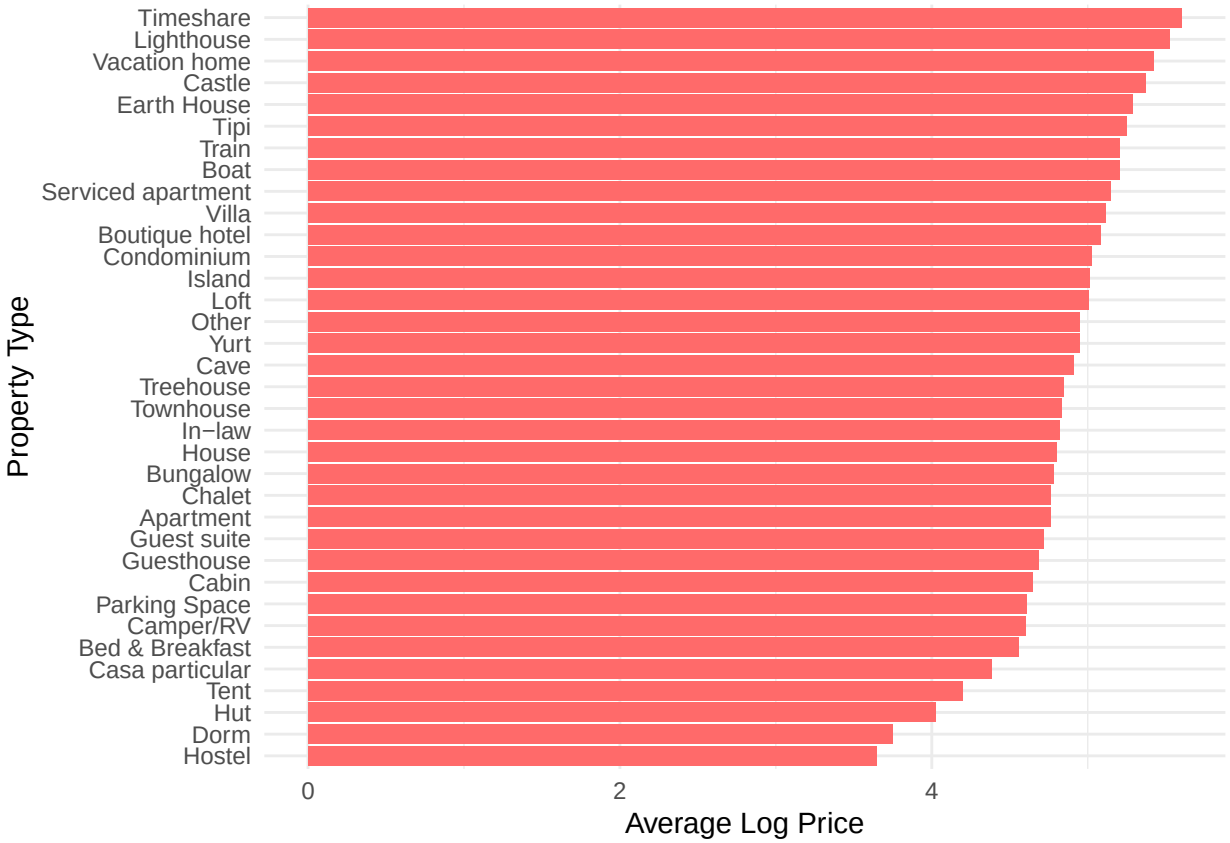


- Popularity of Room Types by City

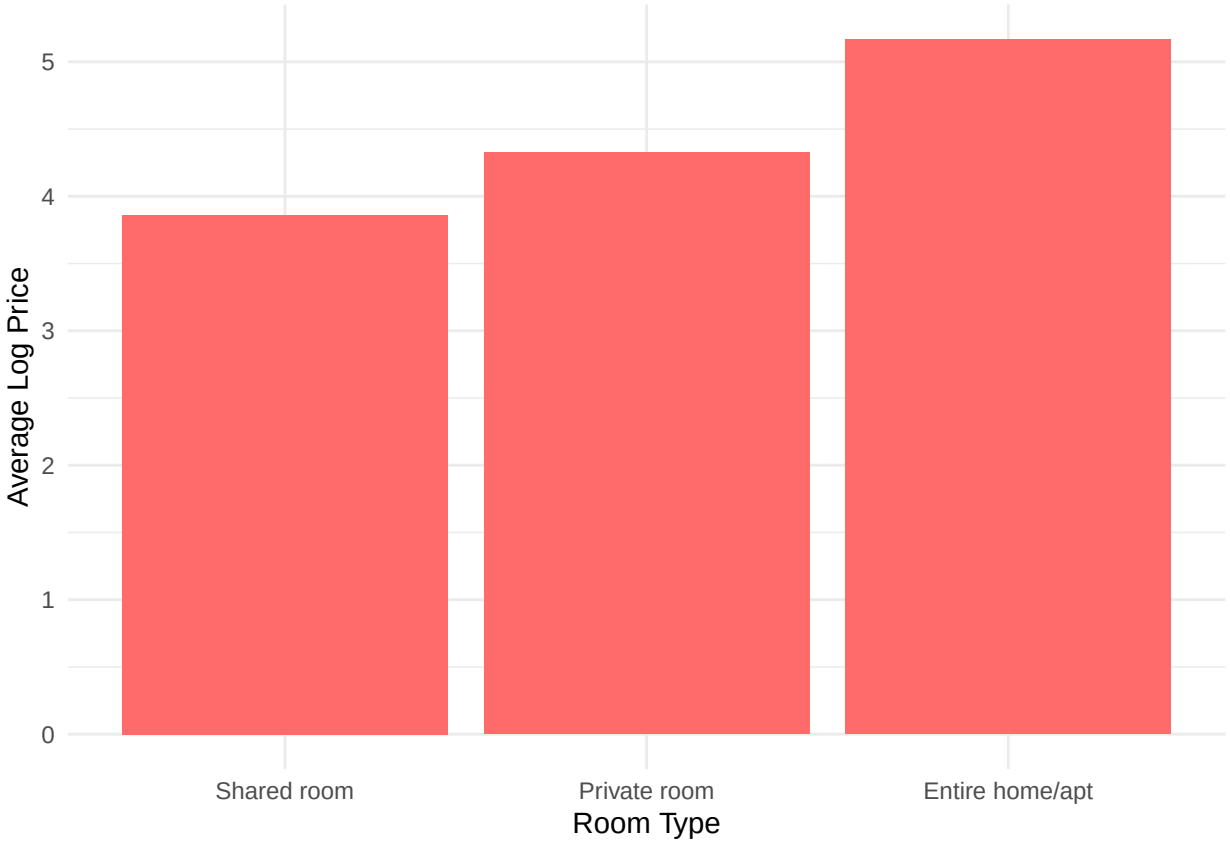
```
## `summarise()` has grouped output by 'city'. You can override using the
## `.groups` argument.
```



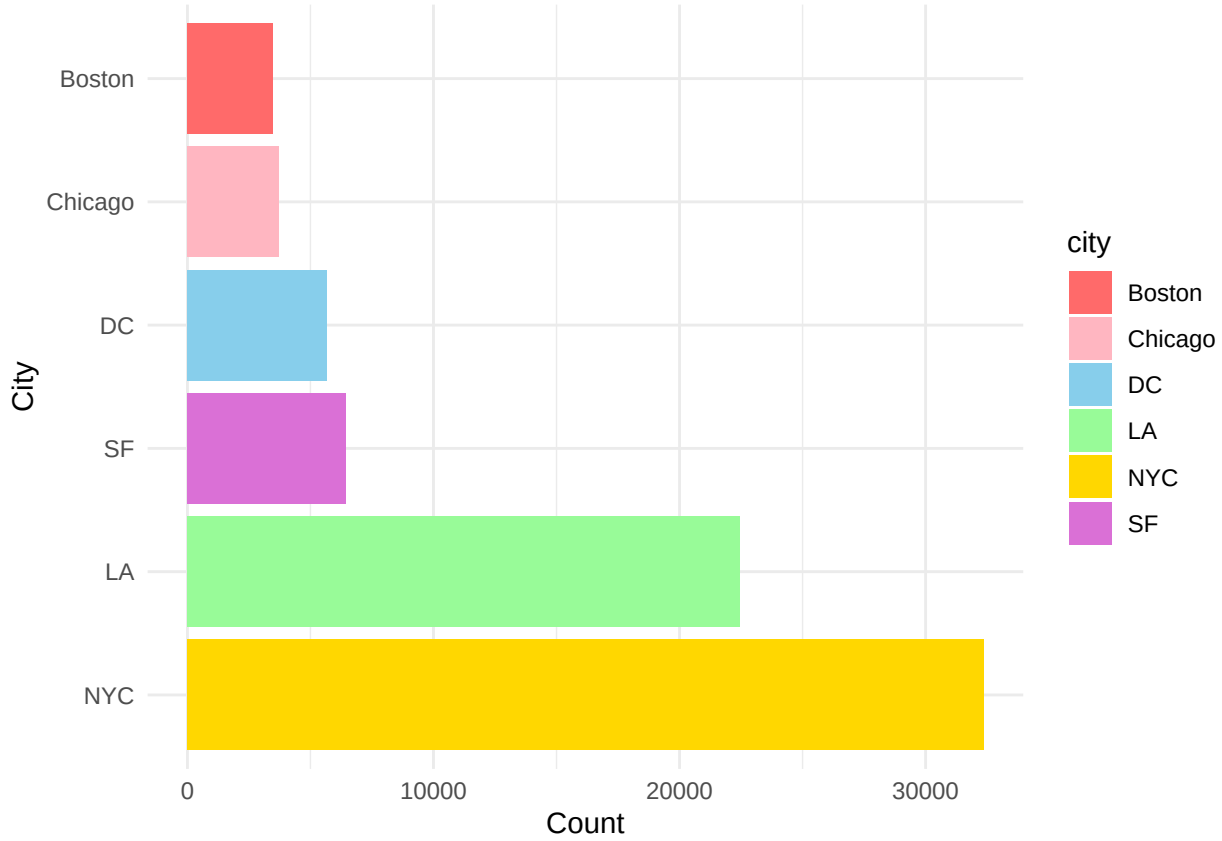
- Average Prices for Different Property Types



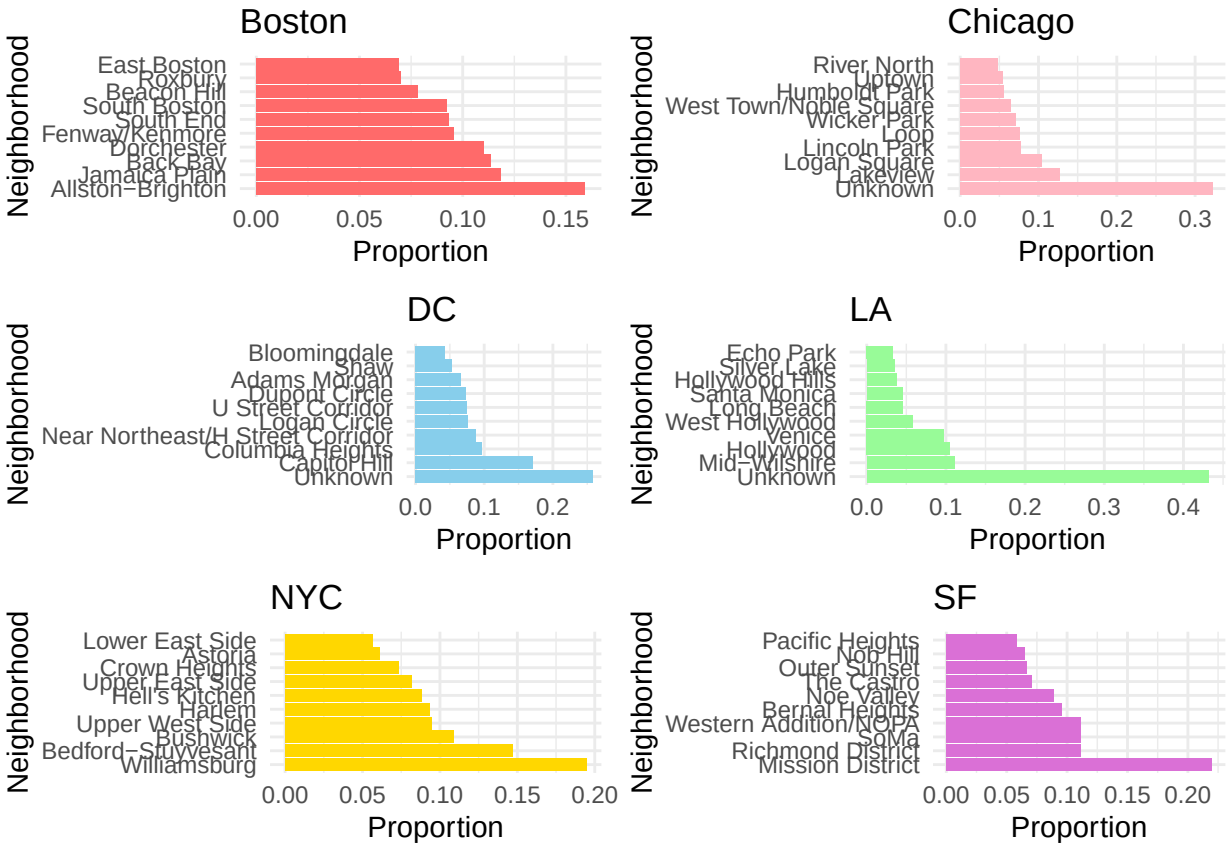
- Average Prices for Different Room Types



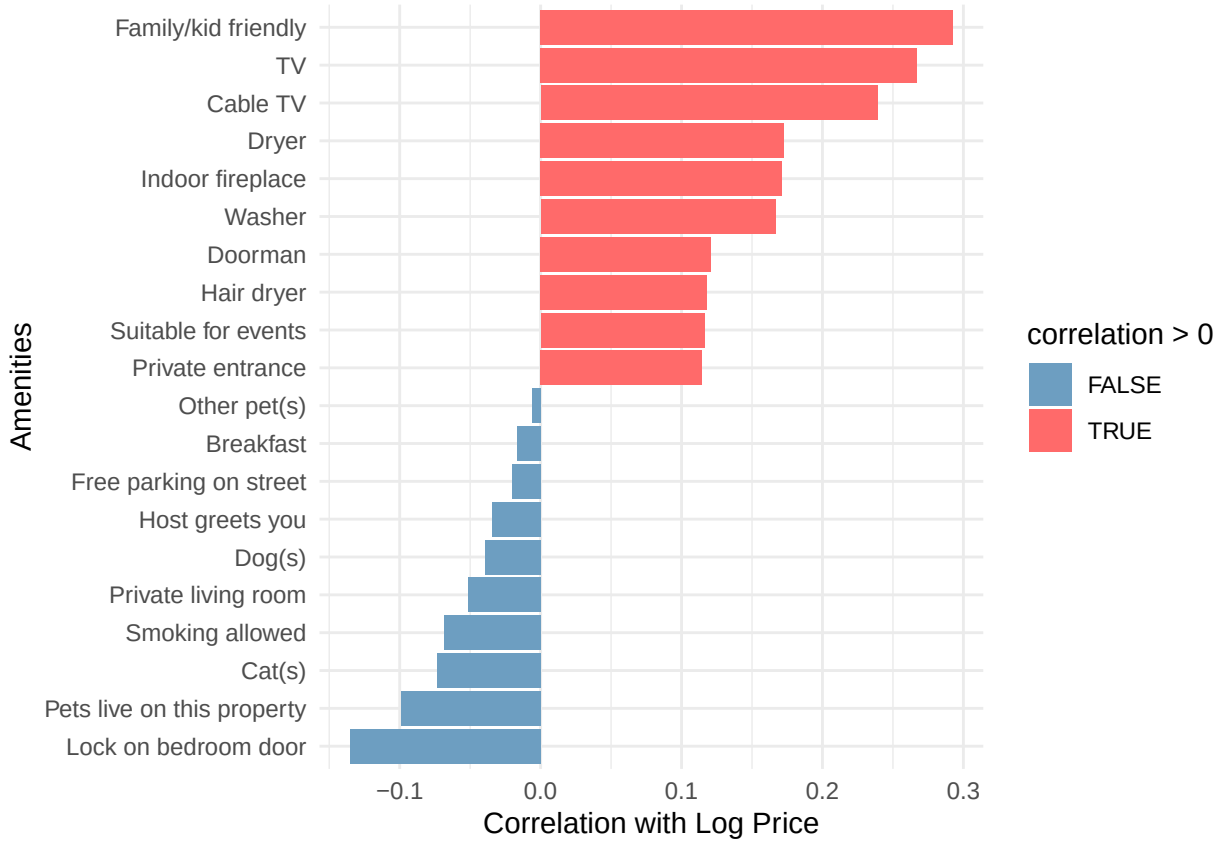
- Distribution of Listings Across Different Cities

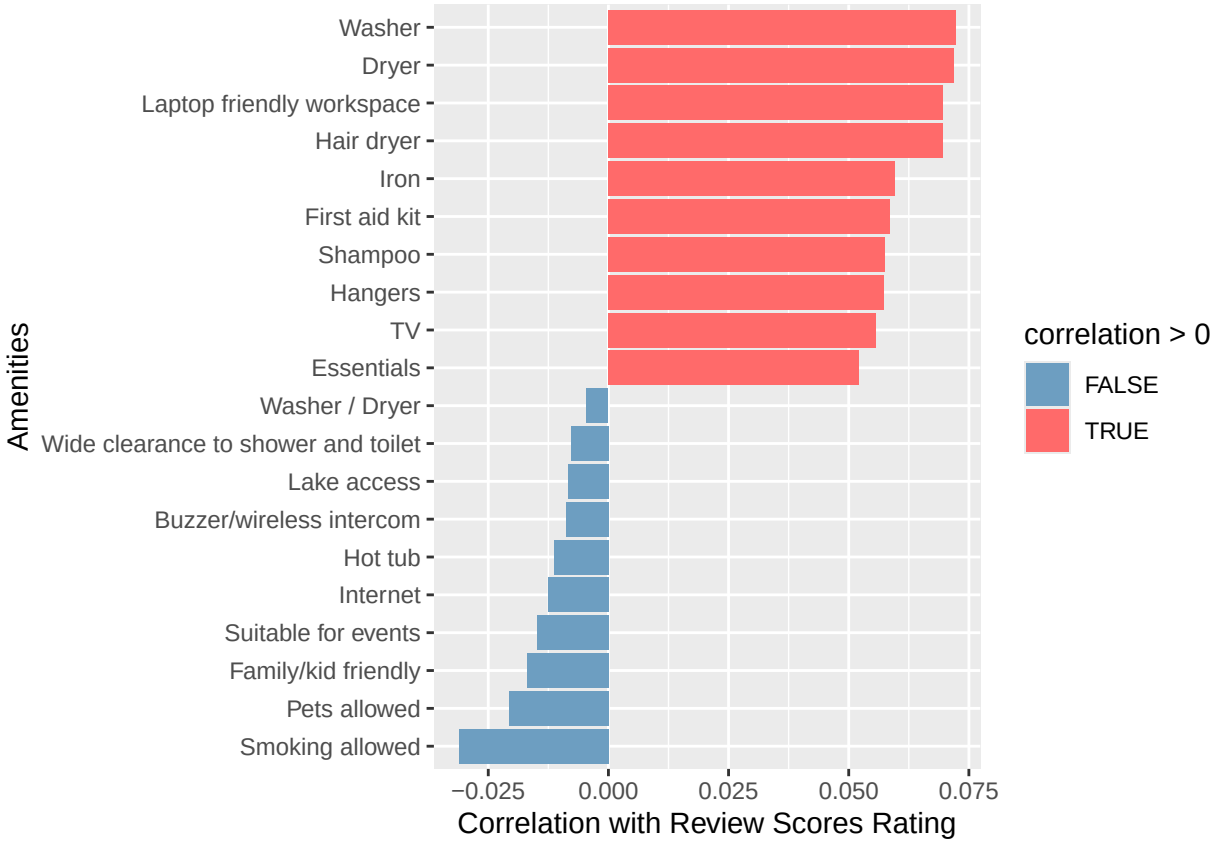


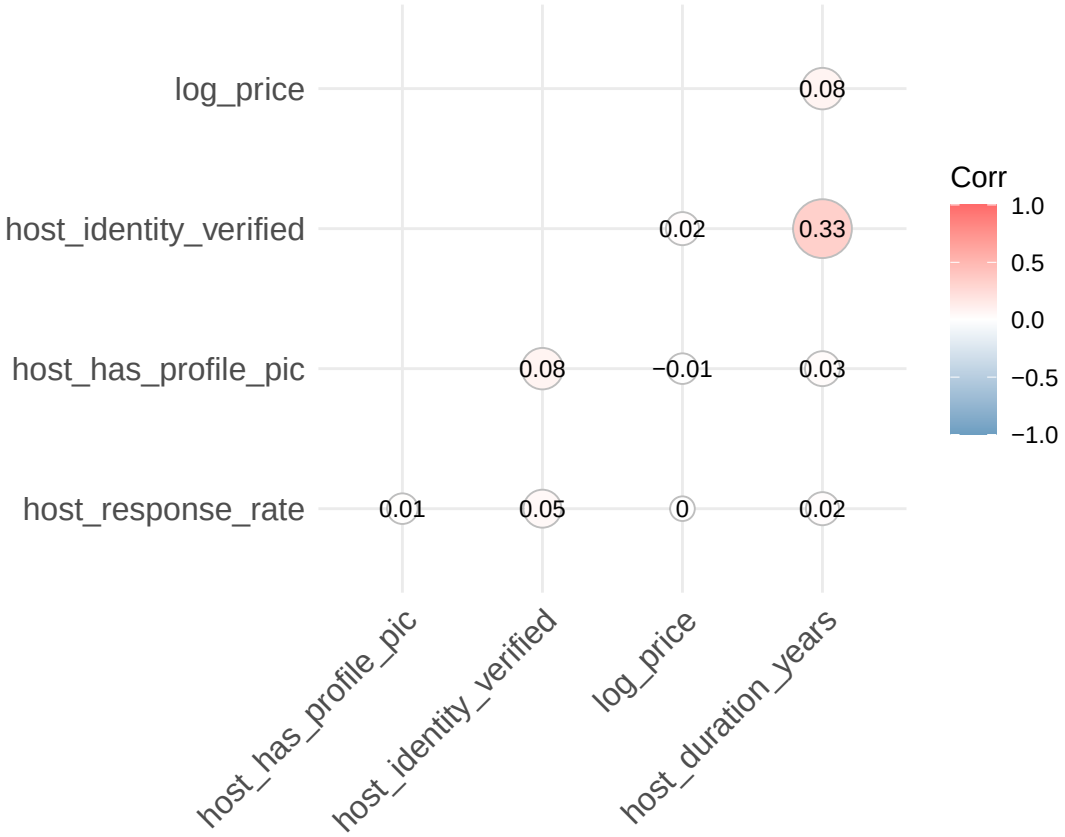
- Distribution of Listings in Neighborhoods for the Top Cities

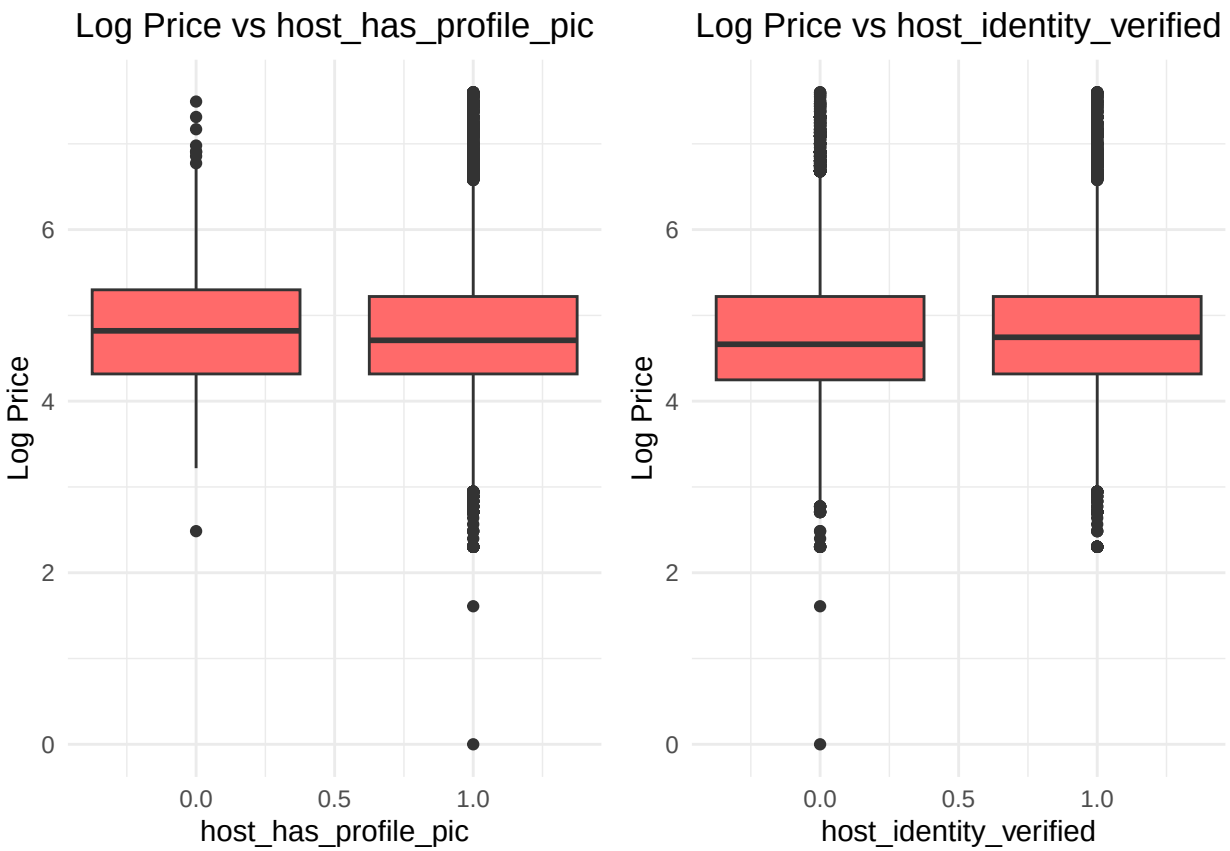



```
## TableGrob (3 x 2) "arrange": 6 grobs
##   z     cells    name           grob
## 1 1 (1-1,1-1) arrange gtable[layout]
## 2 2 (1-1,2-2) arrange gtable[layout]
## 3 3 (2-2,1-1) arrange gtable[layout]
## 4 4 (2-2,2-2) arrange gtable[layout]
## 5 5 (3-3,1-1) arrange gtable[layout]
## 6 6 (3-3,2-2) arrange gtable[layout]
```

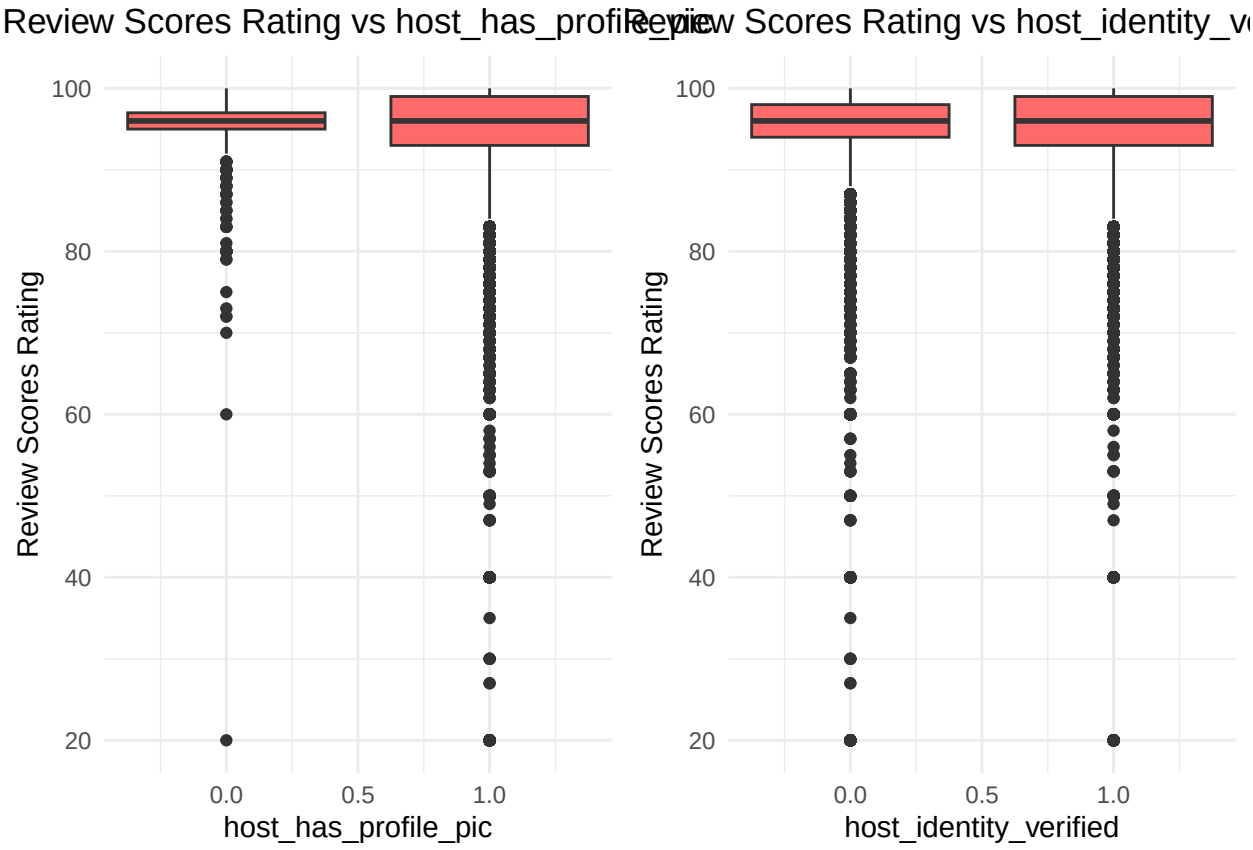

Family/kid friendly
TV
Cable TV
Dryer
Indoor fireplace
Washer
Doorman
Hair dryer
Suitable for events
Private entrance
Other pet(s)
Breakfast
Free parking on street
Host greets you
Dog(s)
Private living room
Smoking allowed
Cat(s)
Pets live on this property
Lock on bedroom door
Amenities
−0.1
0.0
0.1
0.2
0.3
Correlation with Log Price
correlation > 0
FALSE
TRUE


Washer
Dryer
Laptop friendly workspace
Hair dryer
Iron
First aid kit
Shampoo
Hangers
TV
Essentials
Washer / Dryer
Wide clearance to shower and toilet
Lake access
Buzzer/wireless intercom
Hot tub
Internet
Suitable for events
Family/kid friendly
Pets allowed
Smoking allowed
Amenities
−0.025
0.000
0.025
0.050
0.075
Correlation with Review Scores Rating
correlation > 0
FALSE
TRUE


log_price
host_identity_verified
host_has_profile_pic
host_response_rate
0.08
0.02
0.33
0.08
−0.01
0.03
0.01
0.05
0
0.02
host_has_profile_pic
host_identity_verified
log_price
host_duration_years
Corr
1.0
0.5
0.0
−0.5
−1.0



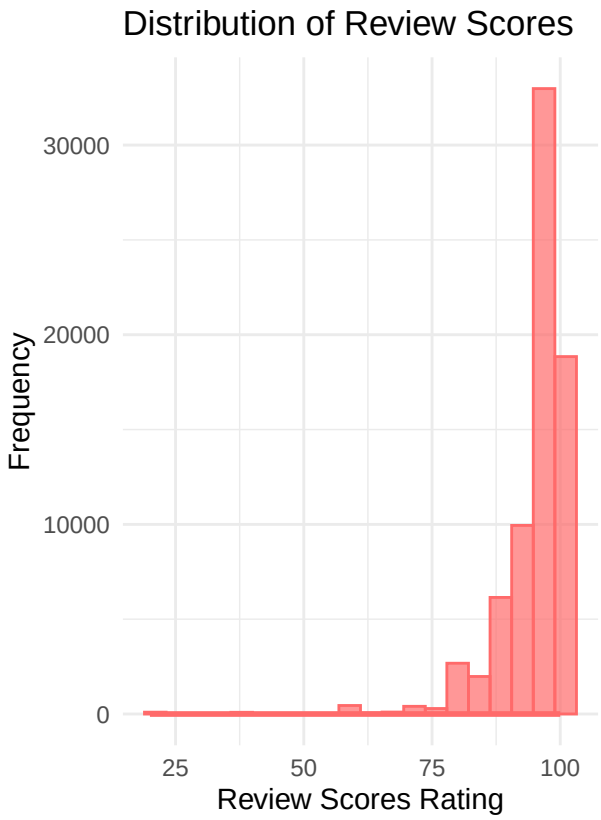

```
## TableGrob (1 x 2) "arrange": 2 grobs
##   z     cells    name           grob
## 1 1 (1-1,1-1) arrange gtable[layout]
## 2 2 (1-1,2-2) arrange gtable[layout]
```

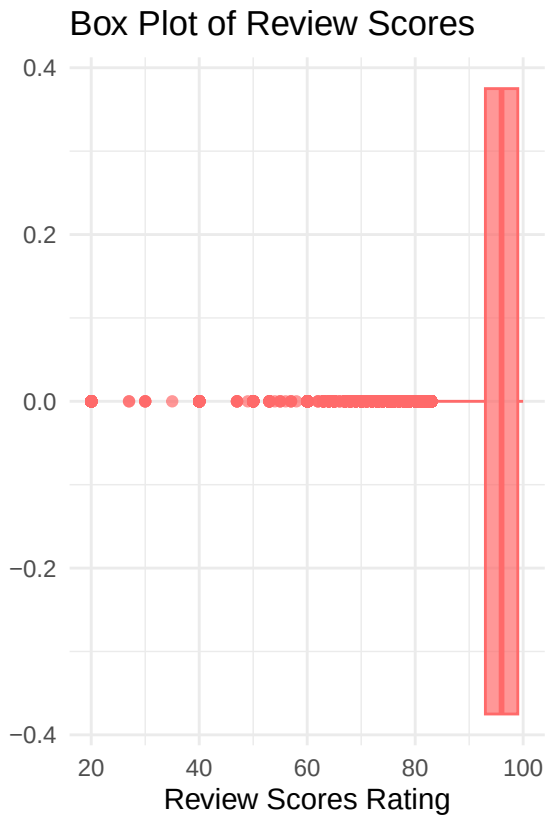



```
## TableGrob (1 x 2) "arrange": 2 grobs
##   z     cells    name           grob
## 1 1 (1-1,1-1) arrange gtable[layout]
## 2 2 (1-1,2-2) arrange gtable[layout]
```

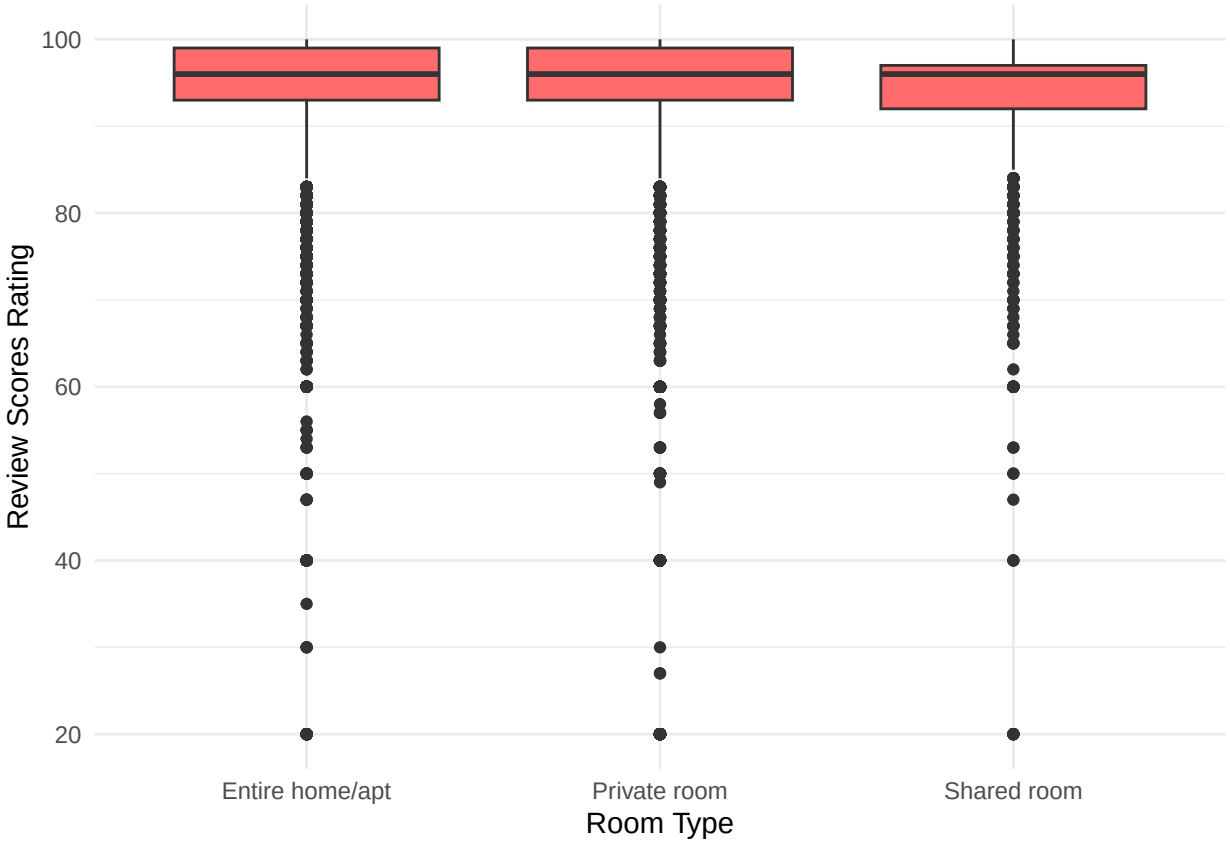
- Distribution of Review Scores

```
## Warning: Using `size` aesthetic for lines was deprecated in ggplot2 3.4.0.
## i Please use `linewidth` instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```

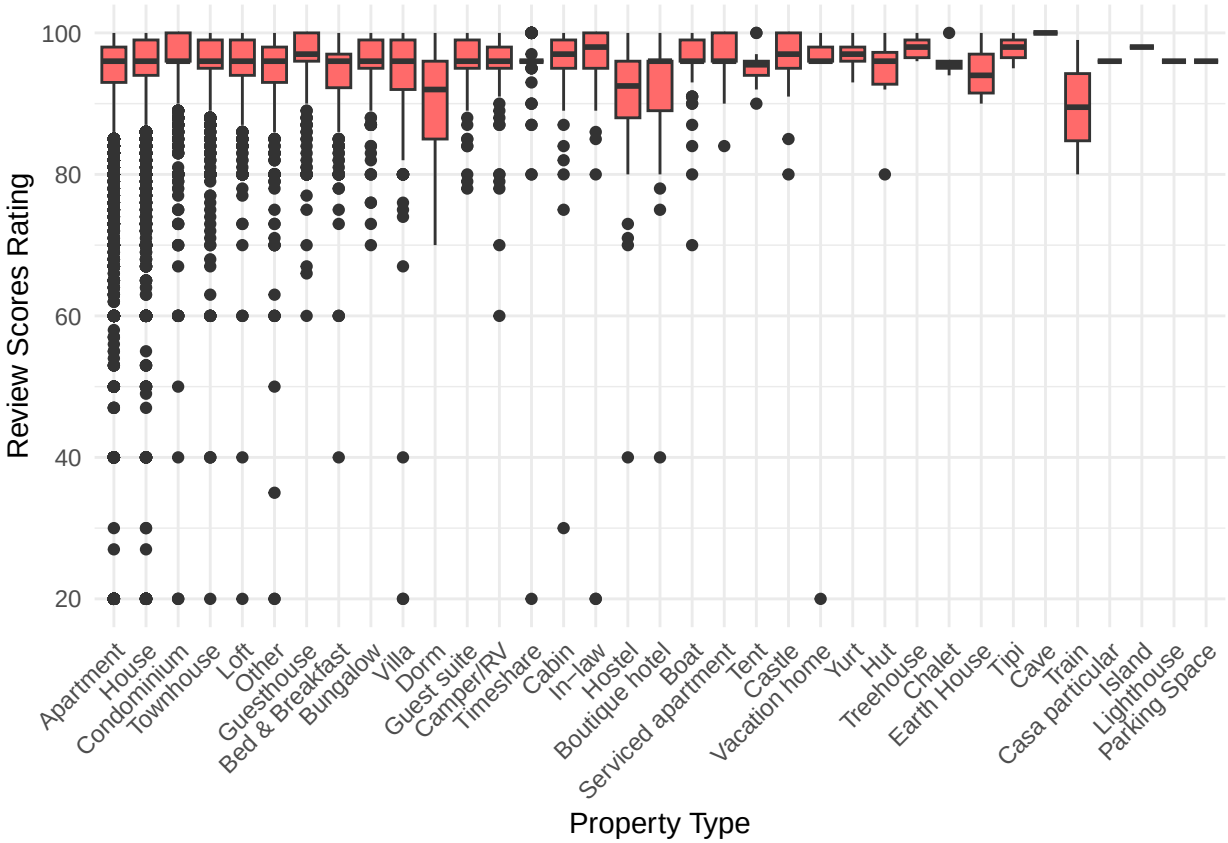




- Box plot for Room Type



- Box plot for Property Type


Review Scores Rating
100
80
60
40
20
Apartment
House
Condominium
Townhouse
Loft
Other
Guesthouse
Bed & Breakfast
Bungalow
Villa
Dorm
Guest suite
Camper/RV
Timeshare
Cabin
In-law
Hostel
Boutique hotel
Boat
Serviced apartment
Tent
Castle
Vacation home
Yurt
Hut
Treehouse
Chalet
Earth House
Tipi
Cave
Train
Casa particular
Island
Lighthouse
Parking Space
Property Type

5. Appendix

```
knitr::opts_chunk$set(echo = TRUE)
library(dplyr)
library(ggplot2)
library(caret)
library(cluster)
library(factoextra)
library(corrplot)
library(GGally)
library(mice)
library(tidyverse)
library(corrplot)
library(reshape2)  # To reshape data
library(ggcorrplot)
library(gridExtra)
library(stringr)
library(mltools)
library(data.table)
library(patchwork)
data <- read.csv("Airbnb_Data.csv")
cat("\nSummary statistics for numerical attributes:\n")
data %>%
  select_if(is.numeric) %>%
  summary()
data$room_type <- as.factor(data$room_type)
data$property_type <- as.factor(data$property_type)
data$bed_type <- as.factor(data$bed_type)
data$cancellation_policy <- as.factor(data$cancellation_policy)
data$city <- as.factor(data$city)
#data$amenities <- as.factor(data$amenities)
#data$description <- as.factor(data$description)
#data$first_review <- as.factor(data$first_review)
#data$host_has_profile_pic <- as.factor(data$host_has_profile_pic)
#data$first_review <- as.factor(data$first_review)
#data$host_identity_verified <- as.factor(data$host_identity_verified)
#data$host_response_rate <- as.factor(data$host_response_rate)
#data$host_since <- as.factor(data$host_since)
data$instant_bookable <- as.factor(data$instant_bookable)
#data$last_review <- as.factor(data$last_review)
#data$neighbourhood <- as.factor(data$neighbourhood)
#data$thumbnail_url <- as.factor(data$thumbnail_url)
#data$zipcode <- as.factor(data$zipcode)
cat("\nSummary statistics for categorical attributes:\n")
data %>%
  select_if(is.factor) %>%
  summary()
count_missing <- function(x) {
  sum(is.na(x) | x == "")
}
# Create a summary of missing values (NA and empty strings) for each column
missing_values_summary <- data %>%
  summarise_all(count_missing) %>%
  gather(key = "variable", value = "missing_count") %>%
```

```
  arrange(desc(missing_count))

print(missing_values_summary)
# Convert host_response_rate to numeric by removing the '%' sign
data$host_response_rate <- as.numeric(gsub("%", "", data$host_response_rate))

# Replace missing values with the median for numeric columns
data$bathrooms[is.na(data$bathrooms)] <- median(data$bathrooms, na.rm = TRUE)
data$bedrooms[is.na(data$bedrooms)] <- median(data$bedrooms, na.rm = TRUE)
data$beds[is.na(data$beds)] <- median(data$beds, na.rm = TRUE)
data$review_scores_rating[is.na(data$review_scores_rating)] <- median(data$review_scores_rating,
                                                                     na.rm = TRUE)
data$host_response_rate[is.na(data$host_response_rate)] <- median(data$host_response_rate,
                                                                 na.rm = TRUE)

# Replace missing values with specific values for categorical columns
data$host_has_profile_pic[is.na(data$host_has_profile_pic) | data$host_has_profile_pic == ''] <- 'f'
data$host_identity_verified[is.na(data$host_identity_verified)
                            | data$host_identity_verified == ''] <- 'f'
data$first_review[is.na(data$first_review) | data$first_review == ''] <- 'Unknown'
data$host_since[is.na(data$host_since) | data$host_since == ''] <- 'Unknown'
data$last_review[is.na(data$last_review) | data$last_review == ''] <- 'Unknown'
data$neighbourhood[is.na(data$neighbourhood) | data$neighbourhood == ''] <- 'Unknown'
data$thumbnail_url[is.na(data$thumbnail_url) | data$thumbnail_url == ''] <- 'No URL'
data$zipcode[is.na(data$zipcode) | data$zipcode == ''] <- 'Unknown'


# Convert categorical columns to factors
# data$thumbnail_url <- factor(data$thumbnail_url)
data$zipcode <- factor(data$zipcode)
data$host_has_profile_pic <- factor(data$host_has_profile_pic)
data$first_review <- factor(data$first_review)
data$last_review <- factor(data$last_review)
data$host_since <- factor(data$host_since)
data$host_identity_verified <- factor(data$host_identity_verified)
data$neighbourhood <- factor(data$neighbourhood)


# Summarize key statistics for numerical attributes after handling missing values
cat("Updated summary statistics for numerical attributes:\n")
data %>%
  select_if(is.numeric) %>%
  summary()

# Summarize key statistics for categorical attributes after handling missing values
cat("\nUpdated summary statistics for categorical attributes:\n")
data %>%
  select_if(is.factor) %>%
  summary()
# Calculate price from log_price assuming log_price is the natural log of price
df <- data.frame(price = exp(data$log_price), log_price = data$log_price)

# Plot for 'price' using indianred1 color
```

```r
price_plot <- ggplot(df, aes(x = price)) +
  geom_histogram(bins = 30, fill = "indianred1", color = "indianred1", alpha = 0.7) +
  geom_density(aes(y = ..density.. * 10), color = "indianred1", linewidth = 1) +
  theme_minimal() +
  labs(title = "Distribution of Prices",
       x = "Price",
       y = "Frequency") +
  theme(plot.title = element_text(hjust = 0.5),
        axis.text.x = element_text(angle = 45, hjust = 1),
        panel.grid.major = element_blank(),  # Removes major grid lines
        panel.grid.minor = element_blank())  # Removes minor grid lines

# Plot for 'log_price' using indianred1 color
log_price_plot <- ggplot(df, aes(x = log_price)) +
  geom_histogram(bins = 30, fill = "indianred1", color = "indianred1", alpha = 0.7) +
  geom_density(aes(y = ..density.. * 10), color = "indianred1", linewidth = 1) +
  theme_minimal() +
  labs(title = "Distribution of Log Prices",
       x = "Log Price",
       y = "Frequency") +
  theme(plot.title = element_text(hjust = 0.5),
        axis.text.x = element_text(angle = 45, hjust = 1),
        panel.grid.major = element_blank(),  # Removes major grid lines
        panel.grid.minor = element_blank())  # Removes minor grid lines

# Combine the plots using patchwork
combined_plot <- price_plot + log_price_plot +
  plot_layout(ncol = 2)

# Print the combined plot
print(combined_plot)
ggsave("price_plot.png", plot = combined_plot, width = 10, height = 6, dpi = 300)
box_plot <- ggplot(df, aes(x = log_price)) +
  geom_boxplot(fill = "indianred1", color = "black", alpha = 0.7) +
  theme_minimal() +
  labs(title = "Box Plot of Log Prices",
       x = "Log Price") +
  theme(plot.title = element_text(hjust = 0.5))
print(box_plot)
ggsave("box_plot.png", plot = box_plot, width = 10, height = 6, dpi = 300)
numerical_features <- data[, c('log_price', 'accommodates', 'bathrooms', 'host_response_rate',
                               'number_of_reviews', 'review_scores_rating', 'bedrooms', 'beds',
                               'latitude')]

correlation_matrix <- cor(numerical_features, use = "complete.obs")

corr_plot <- ggcorrplot(correlation_matrix,
           method = "circle",
           type = "lower",
           lab = TRUE,
           ggtheme = ggplot2::theme_minimal(),
           colors = c("#6D9EC1", "white", "indianred1"),
           lab_size = 3,
```

```
          outline.color = "gray")
print(corr_plot)
ggsave("corr_plot.png", plot = corr_plot, width = 10, height = 6, dpi = 300)
scatter_plots <- list(
  ggplot(data, aes(x = accommodates, y = log_price)) +
    geom_point(color = "#CD5C5C", alpha = 0.6) +  # IndianRed
    theme_minimal() +
    ggtitle('Log Price vs Accommodates') +
    theme(plot.title = element_text(hjust = 0.5)),

  ggplot(data, aes(x = bathrooms, y = log_price)) +
    geom_point(color = "#F08080", alpha = 0.6) +  # LightCoral
    theme_minimal() +
    ggtitle('Log Price vs Bathrooms') +
    theme(plot.title = element_text(hjust = 0.5)),

  ggplot(data, aes(x = bedrooms, y = log_price)) +
    geom_point(color = "#FA8072", alpha = 0.6) +  # Salmon
    theme_minimal() +
    ggtitle('Log Price vs Bedrooms') +
    theme(plot.title = element_text(hjust = 0.5)),

  ggplot(data, aes(x = beds, y = log_price)) +
    geom_point(color = "#E9967A", alpha = 0.6) +  # DarkSalmon
    theme_minimal() +
    scale_x_continuous(breaks = seq(min(data$beds, na.rm = TRUE), max(data$beds, na.rm = TRUE),
                                    by = 1)) +
    ggtitle('Log Price vs Beds') +
    theme(plot.title = element_text(hjust = 0.5))
)

# Arrange all plots in a 2x2 grid layout
two_two_plot <- grid.arrange(grobs = scatter_plots, ncol = 2)
print(two_two_plot)
ggsave("scatter_plot.png", plot = two_two_plot, width = 10, height = 8, dpi = 300)
city <- ggplot(data, aes(x = city, y = log_price)) +
  geom_boxplot(fill = "indianred1", color = "black") +
  theme_minimal() +
  labs(
       x = "City",
       y = "Log Price")
print(city)
ggsave("city_plot.png", plot = city, width = 10, height = 6, dpi = 300)
nyc_data <- subset(data, city == "NYC")

# Identify the top 10 neighborhoods based on frequency
top_neighbourhoods <- names(sort(table(nyc_data$neighbourhood), decreasing = TRUE)[1:10])

# Filter NYC data to only include the top 10 neighborhoods
nyc_top_neighbourhoods_data <- subset(nyc_data, neighbourhood %in% top_neighbourhoods)

# Log Price Distribution Across Top 10 Neighborhoods in NYC
nyc <- ggplot(nyc_top_neighbourhoods_data, aes(x = neighbourhood, y = log_price)) +
```

```
  geom_boxplot(fill = "indianred1", color = "black") +
  theme_minimal() +
  labs(
       x = "Neighborhood",
       y = "Log Price") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
print(nyc)
ggsave("nyc_plot.png", plot = nyc, width = 10, height = 6, dpi = 300)
property <- ggplot(data, aes(y = reorder(property_type, -table(property_type)[property_type]))) +
  geom_bar(fill = "indianred1") +
  theme_minimal() +
  labs(
       x = "Count",
       y = "Property Type")
print(property)
ggsave("property_plot.png", plot = property, width = 10, height = 8, dpi = 300)
room <- ggplot(data, aes(x = reorder(room_type, -table(room_type)[room_type]))) +
  geom_bar(fill = "indianred1") +
  theme_minimal() +
  labs(
       x = "Room Type",
       y = "Count")
print(room)
ggsave("room_plot.png", plot = room, width = 10, height = 8, dpi = 300)
property_type_proportion <- data %>%
  group_by(city, property_type) %>%
  summarize(count = n(), .groups = 'drop') %>%
  group_by(city) %>%
  mutate(proportion = count / sum(count)) %>%
  filter(count > 0)  # Exclude property types with a count of zero

# Plot the proportions
ggplot(property_type_proportion, aes(y = reorder(property_type, -count), x = proportion
                                     , fill = city)) +
  geom_bar(stat = "identity", position = "dodge") +
  theme_minimal() +
  labs(title = "Popularity of Property Types by City",
       x = "Proportion",
       y = "Property Type") +
  guides(fill = guide_legend(title = "City"))
# Calculate proportions of room types by city
room_type_proportion <- data %>%
  group_by(city, room_type) %>%
  summarize(count = n()) %>%
  group_by(city) %>%
  mutate(proportion = count / sum(count))

city_colors <- c("indianred1", "#FFB6C1", "#87CEEB", "#98FB98", "#FFD700", "#DA70D6")

# Plot the proportions
city_portion <- ggplot(room_type_proportion, aes(x = reorder(room_type, -count), y = proportion,
                                                 fill = city)) +
  geom_bar(stat = "identity", position = "dodge") +
```

```
  theme_minimal() +
  labs(x = "Room Type",
       y = "Proportion") +
  scale_fill_manual(values = city_colors) +
  guides(fill = guide_legend(title = "City"))
print(city_portion)
ggsave("city_room_portion_plot.png", plot = city_portion, width = 10, height = 6, dpi = 300)
property_avg <- data %>%
  group_by(property_type) %>%
  summarize(mean_log_price = mean(log_price, na.rm = TRUE)) %>%
  arrange(desc(mean_log_price))

property <- ggplot(property_avg, aes(y = reorder(property_type, mean_log_price),
                                     x = mean_log_price)) +
  geom_bar(stat = "identity", fill = "indianred1") +
  theme_minimal() +
  labs(x = "Average Log Price",
       y = "Property Type")
print(property)
ggsave("average_price_plot.png", plot = property, width = 10, height = 10, dpi = 300)
room_avg <- data %>%
  group_by(room_type) %>%
  summarize(mean_log_price = mean(log_price, na.rm = TRUE)) %>%
  arrange(desc(mean_log_price))

room <- ggplot(room_avg, aes(x = reorder(room_type, mean_log_price), y = mean_log_price)) +
  geom_bar(stat = "identity", fill = "indianred1") +
  theme_minimal() +
  labs(x = "Room Type",
       y = "Average Log Price")
print(room)
ggsave("average_price_room_plot.png", plot = room, width = 6, height = 6, dpi = 300)
city_colors <- c("indianred1", "#FFB6C1", "#87CEEB", "#98FB98", "#FFD700", "#DA70D6")

# Create the bar plot with city-specific colors
city_count <- ggplot(data, aes(y = reorder(city, -table(city)[city]), fill = city)) +
  geom_bar() +
  theme_minimal() +
  labs(x = "Count",
       y = "City") +
  scale_fill_manual(values = city_colors)

print(city_count)
ggsave("city_count_plot.png", plot = city_count, width = 10, height = 6, dpi = 300)
city_colors <- c("Boston" = "indianred1",
                 "Chicago" = "#FFB6C1",
                 "DC" = "#87CEEB",
                 "LA" = "#98FB98",
                 "NYC" = "#FFD700",
                 "SF" = "#DA70D6")

# Analyze the distribution of listings in neighborhoods for the top cities
top_cities <- data %>%
```

```
  count(city, sort = TRUE) %>%
  top_n(6, n) %>%
  pull(city)

# Create a list to hold the plots
plots <- list()

# Loop through each city and create a plot
for (i in 1:length(top_cities)) {
  current_city <- top_cities[i]
  city_data <- data %>%
    filter(city == current_city)

  # Select the top 10 neighborhoods for each city
  top_neighbourhoods <- city_data %>%
    count(neighbourhood, sort = TRUE) %>%
    top_n(10, n) %>%
    pull(neighbourhood)

  city_data_top_neighbourhoods <- city_data %>%
    filter(neighbourhood %in% top_neighbourhoods) %>%
    group_by(neighbourhood) %>%
    summarize(count = n(), .groups = 'drop') %>%
    mutate(proportion = count / sum(count))

  p <- ggplot(city_data_top_neighbourhoods, aes(y = reorder(neighbourhood, -proportion),
                                                x = proportion)) +
    geom_bar(stat = "identity", fill = city_colors[[current_city]]) +
    theme_minimal() +
    labs(title = current_city,
         x = "Proportion",
         y = "Neighborhood")

  plots[[current_city]] <- p
}

# Arrange the plots in a multi-column layout
neigh <- do.call(grid.arrange, c(plots, ncol = 2))
print(neigh)
ggsave("neigh_plot.png", plot = neigh, width = 10, height = 10, dpi = 300)
if (is.character(data$amenities[1])) {
  data <- data %>%
    mutate(amenities = str_remove_all(amenities, '[{}"]')) %>%
    mutate(amenities = str_split(amenities, ','))
}

# Unnest the amenities list column into multiple binary columns
data_unnested <- data %>%
  unnest(amenities) %>%
  mutate(amenities = str_trim(amenities)) %>%  # Trim white spaces
  mutate(amenities = ifelse(amenities == "", NA, amenities)) %>%  # Handle empty strings
  filter(!is.na(amenities)) %>%  # Remove NA amenities
  distinct() %>%
```

```r
  mutate(dummy = 1) %>%
  pivot_wider(names_from = amenities, values_from = dummy, values_fill = list(dummy = 0))

# Select only the amenities columns for correlation analysis
amenities_cols <- data_unnested %>%
  select(-id, -log_price, -review_scores_rating, -accommodates, -bathrooms, -number_of_reviews,
         -latitude, -longitude, -host_response_rate, -bedrooms, -beds, -host_has_profile_pic) %>%
  select(where(is.numeric))

# Analyze the impact of amenities on log_price
amenities_price_impact <- amenities_cols %>%
  summarise(across(everything(), ~ cor(.x, data_unnested$log_price, use = "complete.obs"))) %>%
  pivot_longer(everything(), names_to = "amenity", values_to = "correlation") %>%
  filter(!str_detect(amenity, "translation missing")) %>%  # Exclude translation missing amenities
  arrange(desc(correlation))

# Select top 10 positive and top 10 negative correlations for log_price
top_positive_price <- amenities_price_impact %>% top_n(10, correlation)
top_negative_price <- amenities_price_impact %>% top_n(-10, correlation)
top_price_impact <- bind_rows(top_positive_price, top_negative_price)

# Plot the top 10 positive and top 10 negative correlations for log_price
amenities_price <- ggplot(top_price_impact, aes(x = correlation, y = reorder(amenity, correlation),
                                                fill = correlation > 0)) +
  geom_bar(stat = "identity") +
  scale_fill_manual(values = c("TRUE" = "indianred1", "FALSE" = "#6D9EC1")) +
  theme_minimal() +
  labs(x = "Correlation with Log Price",
       y = "Amenities")

# Analyze the impact of amenities on review_scores_rating
amenities_review_impact <- amenities_cols %>%
  summarise(across(everything(), ~ cor(.x, data_unnested$review_scores_rating, use = "complete.obs"))) %
  pivot_longer(everything(), names_to = "amenity", values_to = "correlation") %>%
  filter(!str_detect(amenity, "translation missing")) %>%  # Exclude translation missing amenities
  arrange(desc(correlation))

# Select top 10 positive and top 10 negative correlations for review_scores_rating
top_positive_review <- amenities_review_impact %>% top_n(10, correlation)
top_negative_review <- amenities_review_impact %>% top_n(-10, correlation)
top_review_impact <- bind_rows(top_positive_review, top_negative_review)

# Plot the top 10 positive and top 10 negative correlations for review_scores_rating
amenities_review <- ggplot(top_review_impact, aes(x = correlation,
                                                  y = reorder(amenity, correlation),
                                                  fill = correlation > 0)) +
  geom_bar(stat = "identity") +
  scale_fill_manual(values = c("TRUE" = "indianred1", "FALSE" = "#6D9EC1")) +
  labs(x = "Correlation with Review Scores Rating",
       y = "Amenities")
print(amenities_price)
print(amenities_review)
ggsave("amenities_price_plot.png", plot = amenities_price, width = 10, height = 10, dpi = 300)
```

```r
ggsave("amenities_review_plot.png", plot = amenities_review, width = 10, height = 10, dpi = 300)
# Convert 'host_since' to Date and create a new feature 'host_duration_years' to see
# how long they have been a host
data$host_since <- as.Date(data$host_since)
data$host_duration_years <- as.numeric((Sys.Date() - data$host_since) / 365.25)

# Convert 'host_has_profile_pic' and 'host_identity_verified' to numeric for correlation analysis
data$host_has_profile_pic <- ifelse(data$host_has_profile_pic == "t", 1, 0)
data$host_identity_verified <- ifelse(data$host_identity_verified == "t", 1, 0)
# Select relevant columns for analysis
host_data <- data %>%
  select(host_response_rate, host_has_profile_pic, host_identity_verified,
         log_price, host_duration_years)

# Calculate the correlation matrix for host_data
correlation_matrix <- cor(host_data, use = "complete.obs")

# Plot the correlation matrix using ggcorrplot
corr_plot <- ggcorrplot(correlation_matrix,
           method = "circle",
           type = "lower",
           lab = TRUE, # Show correlation coefficients
           ggtheme = ggplot2::theme_minimal(),
           colors = c("#6D9EC1", "white", "indianred1"),
           lab_size = 3,
           outline.color = "gray")
print(corr_plot)
ggsave("corr_host_plot.png", plot = corr_plot, width = 10, height = 8, dpi = 300)
# Define host features and output variables
host_features <- c("host_has_profile_pic", "host_identity_verified")

# Create lists to store plots separately for log_price and review_scores_rating
plot_list_log_price <- vector("list", length = length(host_features))
plot_list_review_scores <- vector("list", length = length(host_features))

# Create plots and store them in the respective lists
for (i in seq_along(host_features)) {
  host_feature <- host_features[i]

  plot_list_log_price[[i]] <- ggplot(data, aes(x = .data[[host_feature]], y = log_price)) +
    geom_boxplot(aes(group = .data[[host_feature]]), fill = "indianred1") +
    theme_minimal() +
    labs(title = paste("Log Price vs", host_feature), x = host_feature, y = "Log Price") +
    theme(plot.title = element_text(hjust = 0.5))

  plot_list_review_scores[[i]] <- ggplot(data, aes(x = .data[[host_feature]],
                                                   y = review_scores_rating)) +
    geom_boxplot(aes(group = .data[[host_feature]]), fill = "indianred1") +
    theme_minimal() +
    labs(title = paste("Review Scores Rating vs", host_feature), x = host_feature,
         y = "Review Scores Rating") +
    theme(plot.title = element_text(hjust = 0.5))
}
```

```r
# Display log_price plots
log_price_host <- grid.arrange(grobs = plot_list_log_price, ncol = 2)
print(log_price_host)

# Display review_scores_rating plots
review_scores_host <- grid.arrange(grobs = plot_list_review_scores, ncol = 2)
print(review_scores_host)
ggsave("log_price_host_plot.png", plot = log_price_host, width = 10, height = 6, dpi = 300)
ggsave("review_host_plot.png", plot = review_scores_host, width = 10, height = 6, dpi = 300)
histogram_plot <- ggplot(data, aes(x = review_scores_rating)) +
  geom_histogram(bins = 20, fill = "indianred1", color = "indianred1", alpha = 0.7) +
  geom_density(color = "indianred1", size = 1) +
  labs(title = "Distribution of Review Scores",
       x = "Review Scores Rating",
       y = "Frequency") +
  theme_minimal()

boxplot_plot <- ggplot(data, aes(y = review_scores_rating)) +
  geom_boxplot(fill = "indianred1", color = "indianred1", alpha = 0.7) +
  coord_flip() +  #
  labs(title = "Box Plot of Review Scores",
       x = "",
       y = "Review Scores Rating") +
  theme_minimal()

combined_plot <- histogram_plot + boxplot_plot + plot_layout(ncol = 2)
print(combined_plot)
ggsave("review_scores_plot.png", plot = combined_plot, width = 10, height = 6, dpi = 300)
review_room <- ggplot(data, aes(x = room_type, y = review_scores_rating)) +
  geom_boxplot(fill = "indianred1") +
  labs(x = "Room Type",
       y = "Review Scores Rating") +
  theme_minimal()
print(review_room)
ggsave("review_room_plot.png", plot = review_room, width = 10, height = 6, dpi = 300)
review_property <- ggplot(data, aes(x = reorder(property_type, -table(property_type)[property_type]),
                                    y = review_scores_rating)) +
  geom_boxplot(fill = "indianred1") +
  labs(
       x = "Property Type",
       y = "Review Scores Rating") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
print(review_property)
ggsave("review_property_plot.png", plot = review_property, width = 10, height = 6, dpi = 300)
```