BUSN 41201 Initial Final

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

Load the Data

```
data <- read.csv("Airbnb_Data.csv")
```

- Summary statistics for numerical attributes

```
cat("\nSummary statistics for numerical attributes:\n")
```

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

```
data %>%
  select_if(is.numeric) %>%
  summary()
```

```
##        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

```
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)
```

- Summary statistics for factors

```
cat("\nSummary statistics for categorical attributes:\n")
```

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

```
data %>%
  select_if(is.factor) %>%
  summary()
```

```
##      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

```
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)
```

```
##                  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
```

```
# 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 =
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 == ''] <-
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")
```

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

```
data %>%
  select_if(is.numeric) %>%
  summary()
```

```
##        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
```

```
# Summarize key statistics for categorical attributes after handling missing values
cat("\nUpdated summary statistics for categorical attributes:\n")
```

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

```
data %>%
  select_if(is.factor) %>%
  summary()
```

```
##      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
```

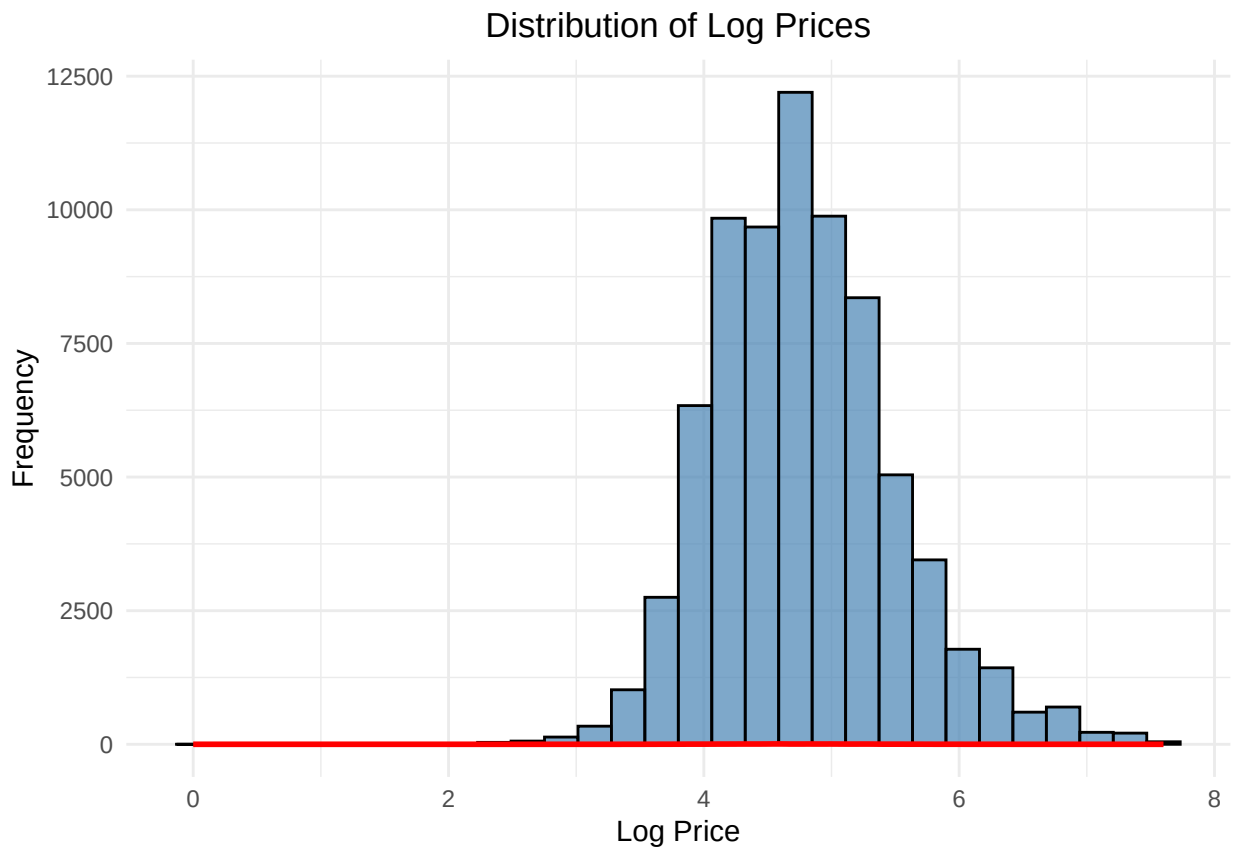
```
##  Townhouse  : 1692                          Pull-out Sofa:  585
##  Loft       : 1244                          Real Bed     :72028
##  Other      :  607
##  (Other)    : 2396
##       cancellation_policy      city           first_review
##  flexible       :22545    Boston : 3468   Unknown   :15864
##  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

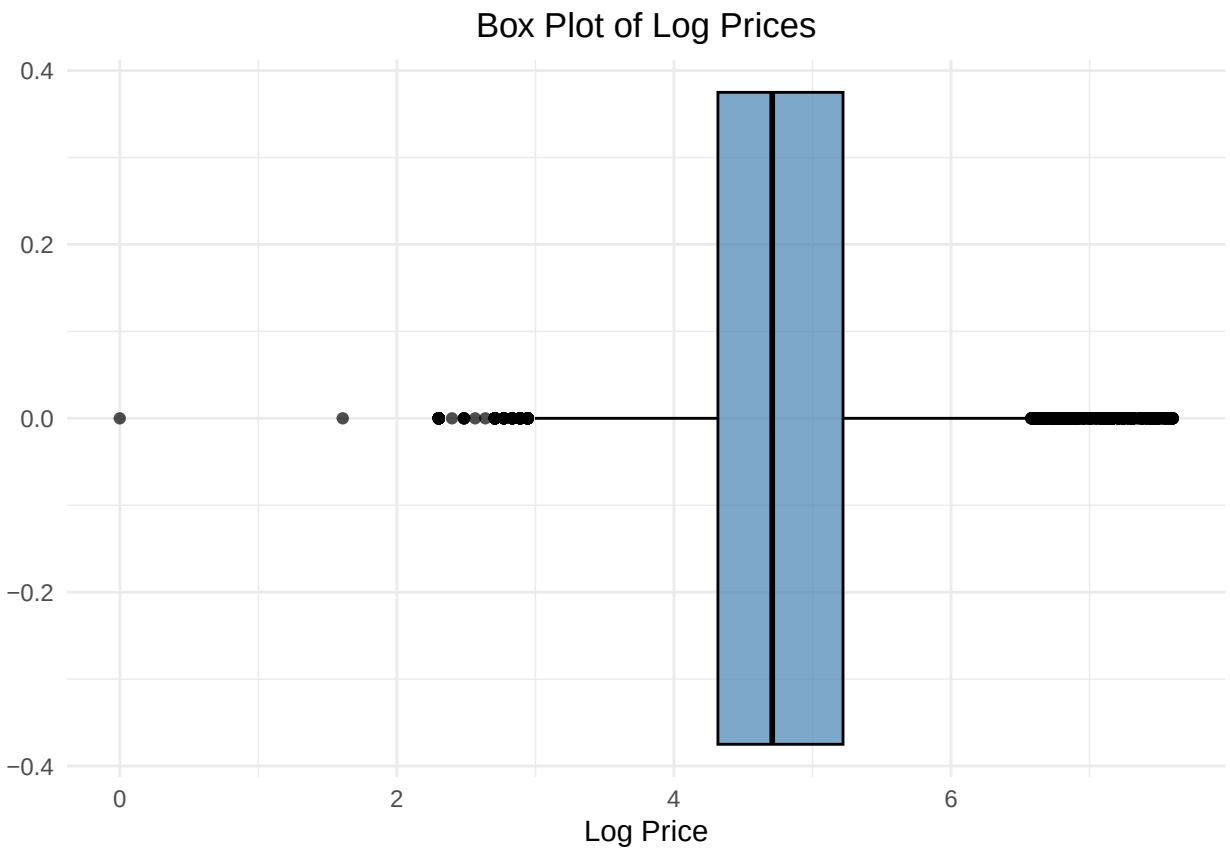
```
# Replace 'data' with your actual data frame name
log_price_data <- data$log_price

# Convert the data frame to a usable form in ggplot2
df <- data.frame(log_price = log_price_data)

ggplot(df, aes(x = log_price)) +
  geom_histogram(bins = 30, fill = "steelblue", color = "black", alpha = 0.7) +
  geom_density(aes(y = ..density.. * 10), color = "red", linewidth = 1) +  # KDE line overlay
  theme_minimal() +
  labs(title = "Distribution of Log Prices",
       x = "Log Price",
       y = "Frequency") +
  theme(plot.title = element_text(hjust = 0.5))
```

```
## Warning: The dot-dot notation (`..density..`) was deprecated in ggplot2 3.4.0.
## i Please use `after_stat(density)` instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```



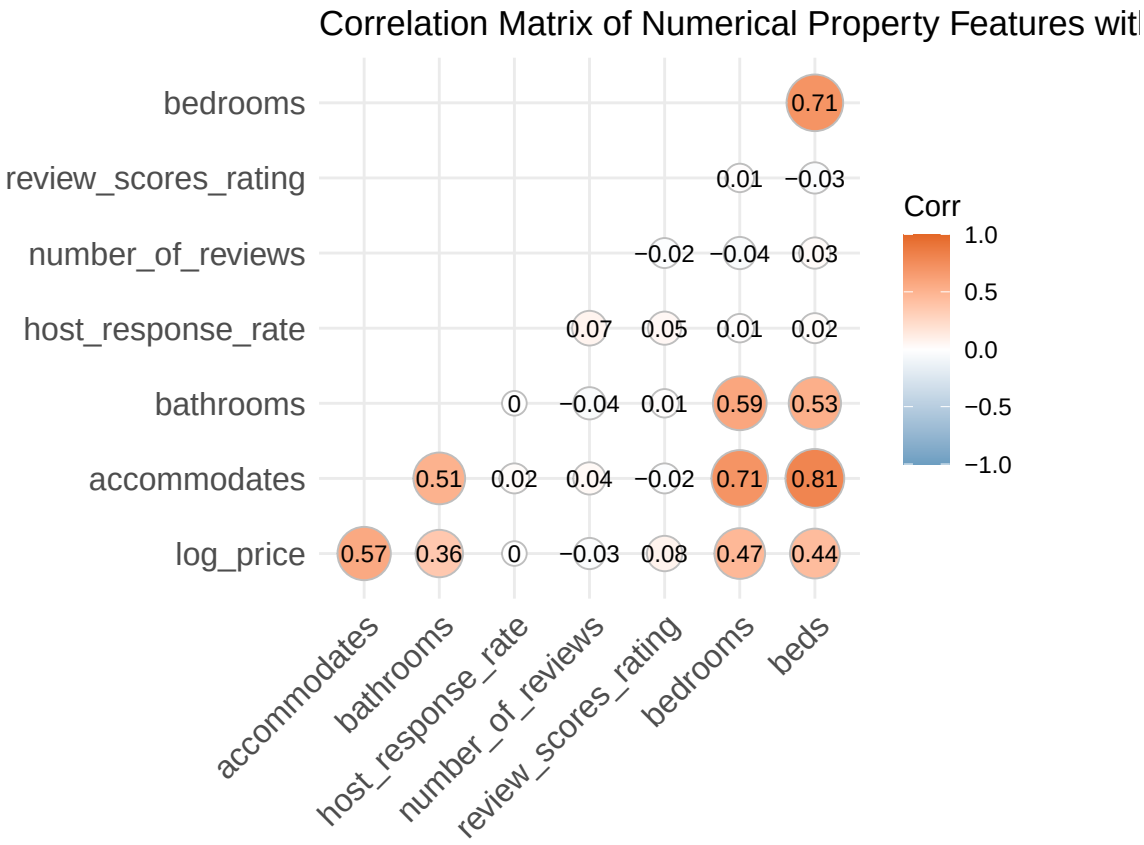
- Box plot of log price

```
ggplot(df, aes(x = log_price)) +
  geom_boxplot(fill = "steelblue", 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))
```



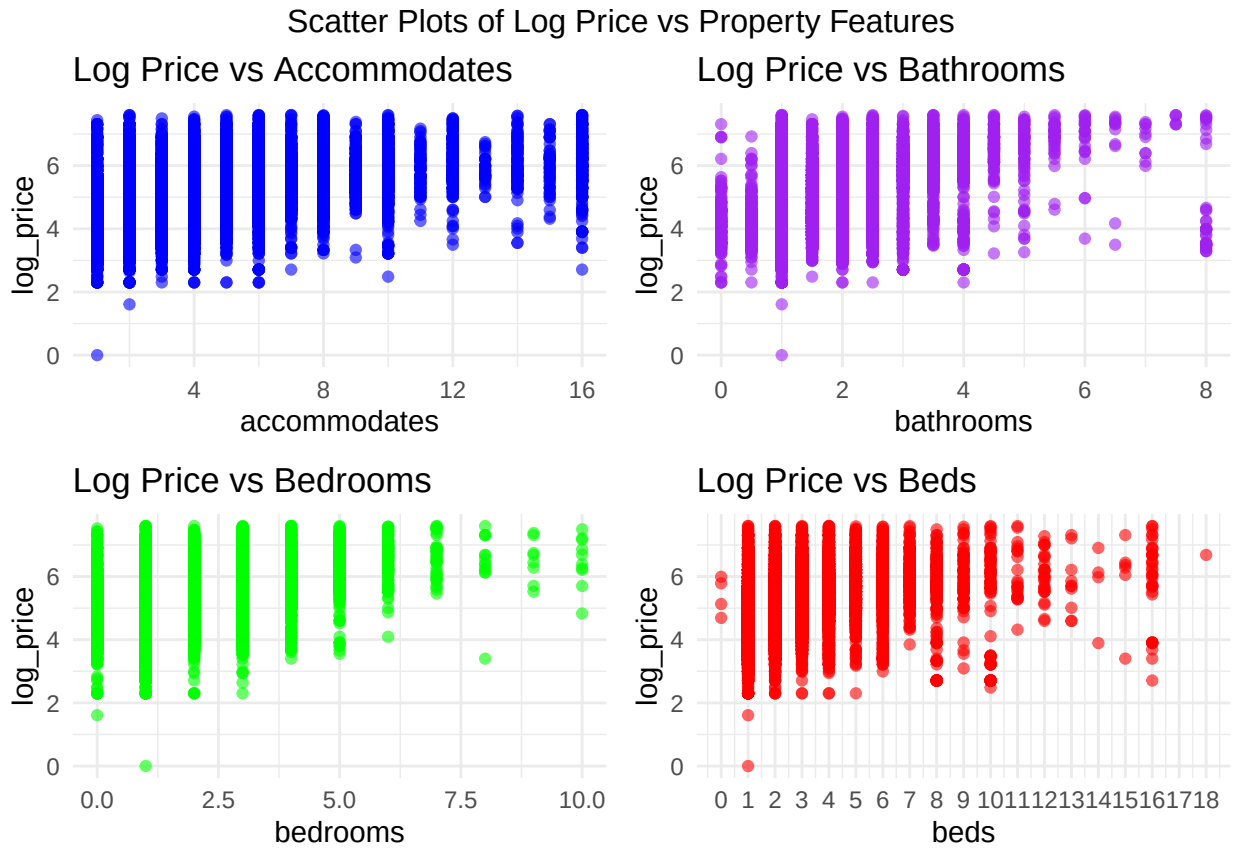
- Correlation plot for numerical columns

```
numerical_features <- data[, c('log_price', 'accommodates', 'bathrooms', 'host_response_rate', 'number_

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

ggcorrplot(correlation_matrix,
           method = "circle",
           type = "lower",
           lab = TRUE,
           ggtheme = ggplot2::theme_minimal(),
           colors = c("#6D9EC1", "white", "#E46726"),
           lab_size = 3,
           outline.color = "gray") +
  ggtitle('Correlation Matrix of Numerical Property Features with Log Price')
```



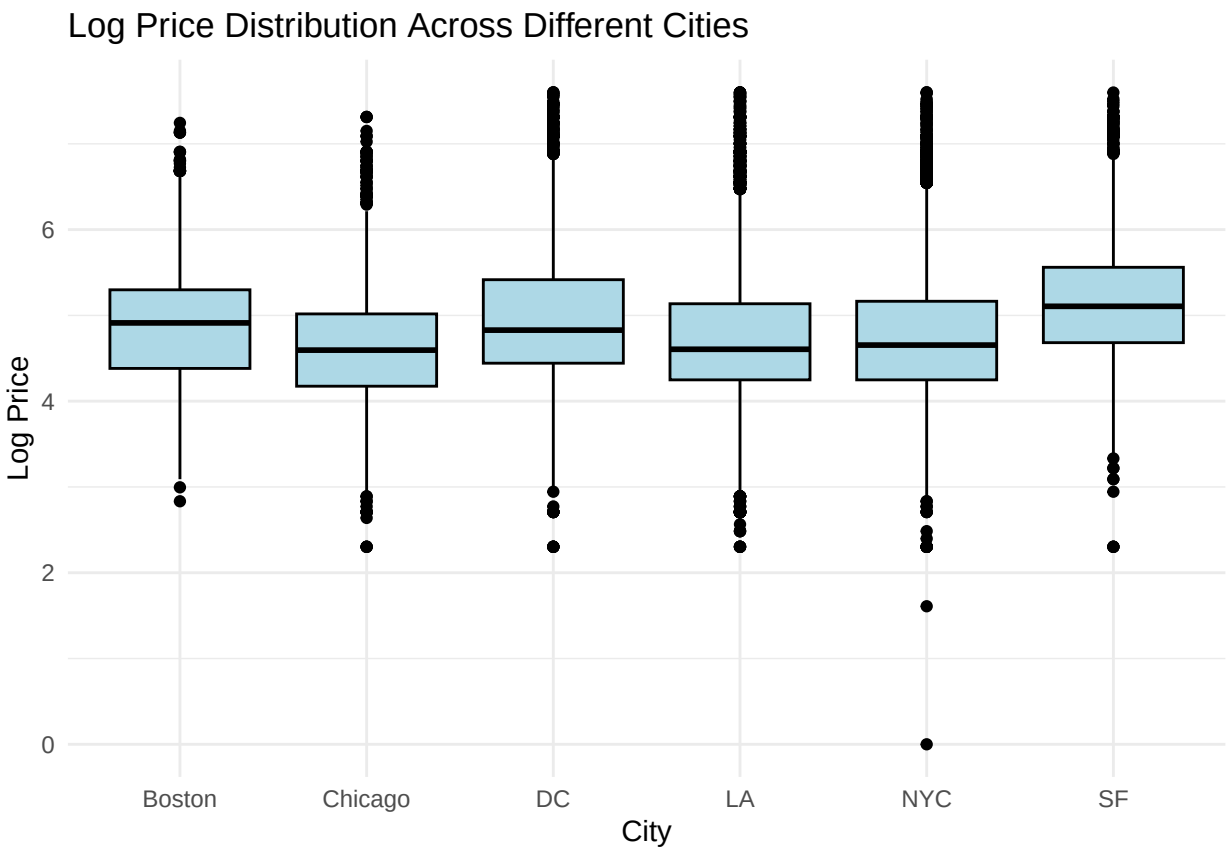
- Scatter plot for log price vs numerical property features

```
scatter_plots <- list(
  ggplot(data, aes(x = accommodates, y = log_price)) +
    geom_point(color = "blue", alpha = 0.6) +
    theme_minimal() +
    ggtitle('Log Price vs Accommodates'),

  ggplot(data, aes(x = bathrooms, y = log_price)) +
    geom_point(color = "purple", alpha = 0.6) +
    theme_minimal() +
    ggtitle('Log Price vs Bathrooms'),

  ggplot(data, aes(x = bedrooms, y = log_price)) +
    geom_point(color = "green", alpha = 0.6) +
    theme_minimal() +
    ggtitle('Log Price vs Bedrooms'),

  ggplot(data, aes(x = beds, y = log_price)) +
    geom_point(color = "red", alpha = 0.6) +
    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')
)

# Arrange all plots in a 2x2 grid layout
grid.arrange(grobs = scatter_plots, ncol = 2, top = "Scatter Plots of Log Price vs Property Features")
```



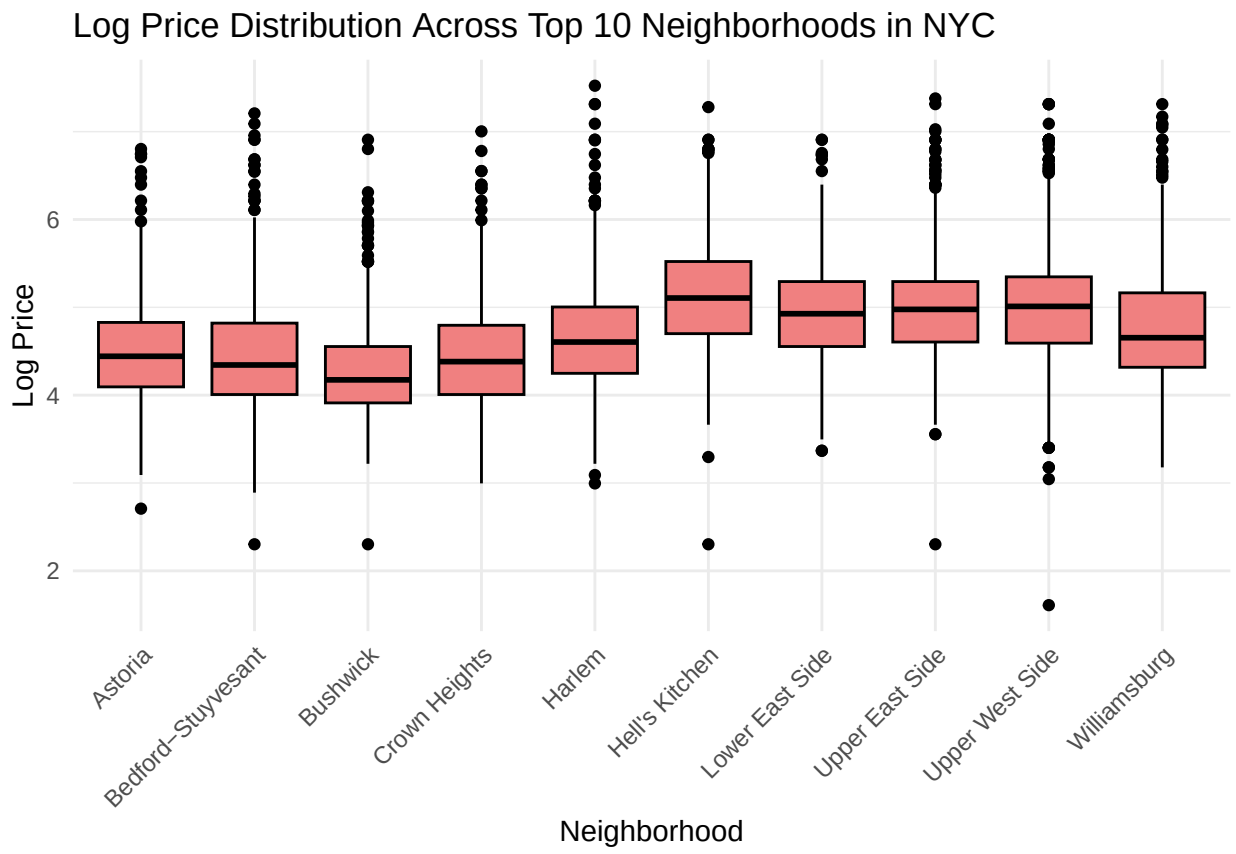
- Log Price Distribution Across Different 6 Cities

```
ggplot(data, aes(x = city, y = log_price)) +
  geom_boxplot(fill = "lightblue", color = "black") +
  theme_minimal() +
  labs(title = "Log Price Distribution Across Different Cities",
       x = "City",
       y = "Log Price")
```



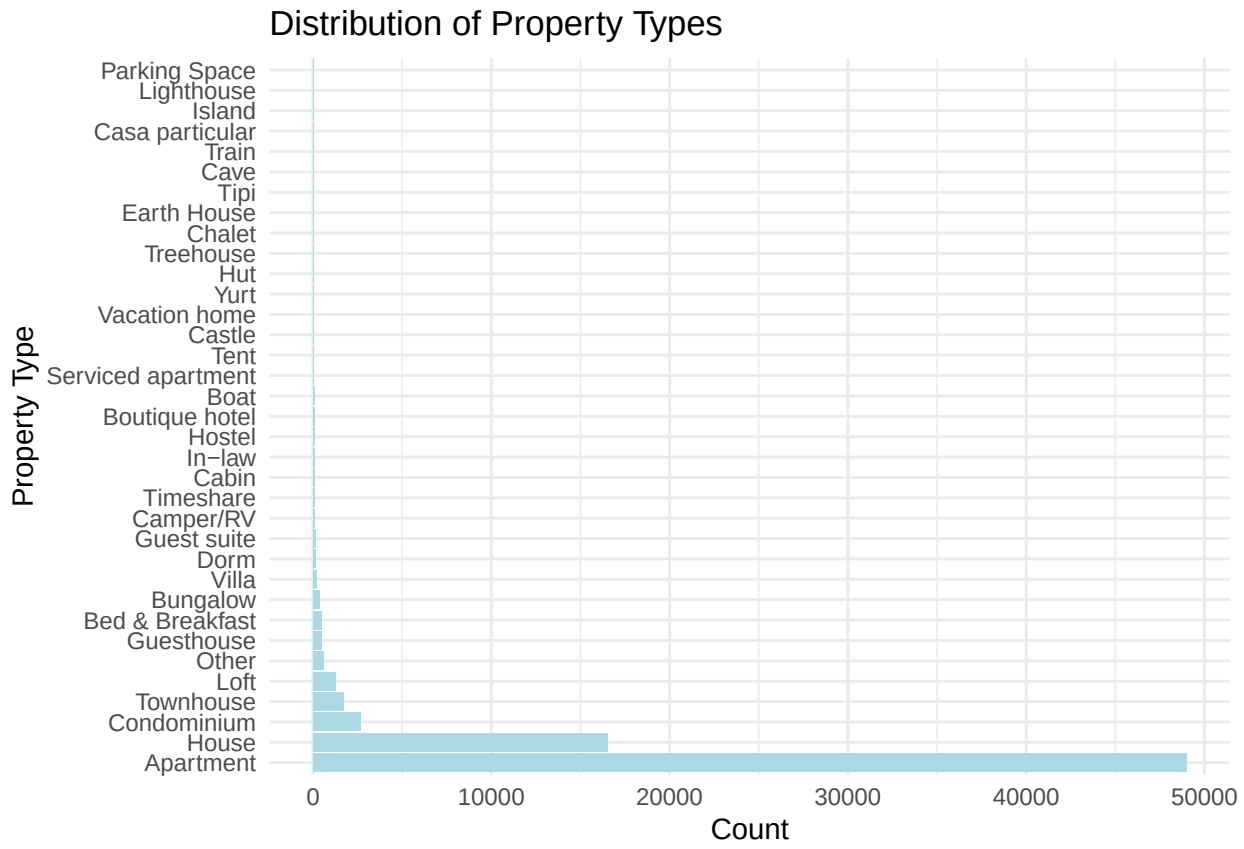
- Log Price Distribution Across Top 10 Neighborhoods in NYC

```
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
ggplot(nyc_top_neighbourhoods_data, aes(x = neighbourhood, y = log_price)) +
  geom_boxplot(fill = "lightcoral", color = "black") +
  theme_minimal() +
  labs(title = "Log Price Distribution Across Top 10 Neighborhoods in NYC",
       x = "Neighborhood",
       y = "Log Price") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```



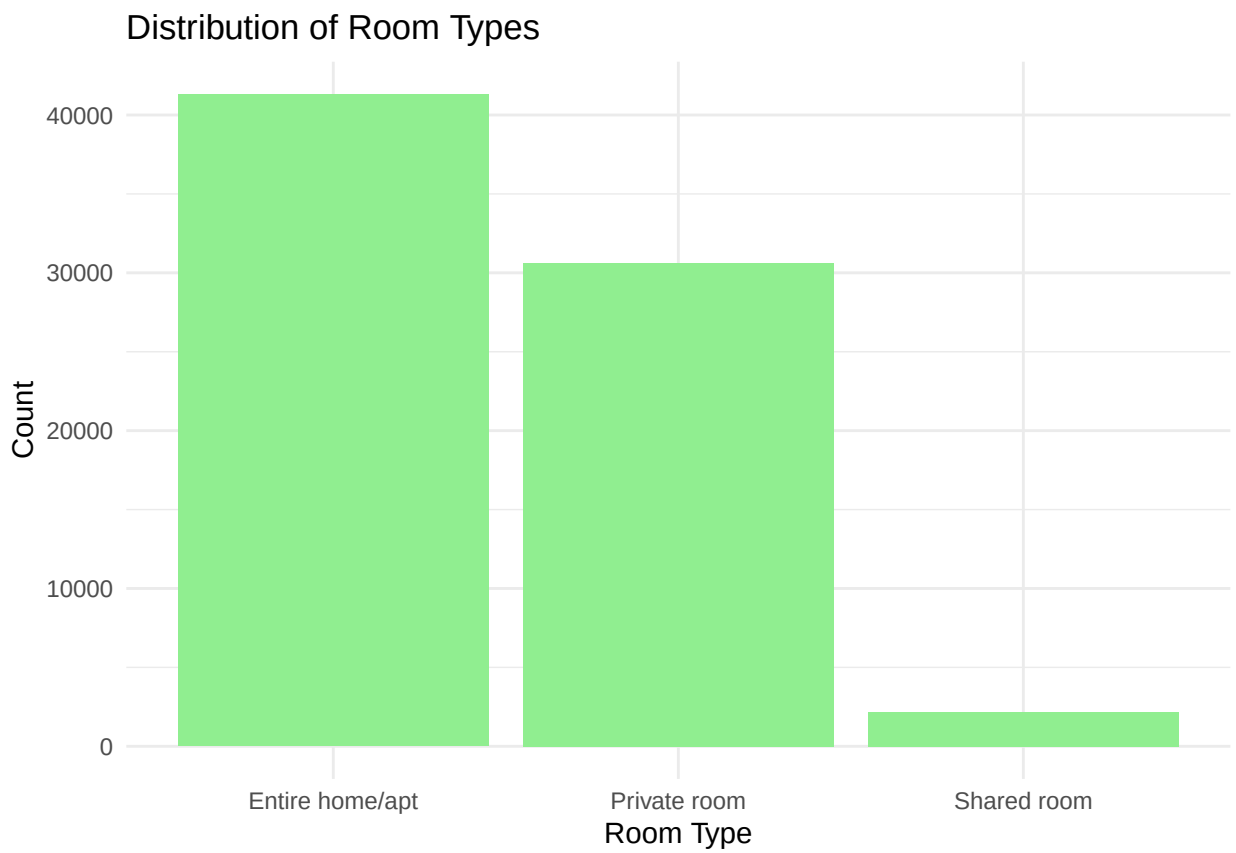
- Distribution of Property Types

```
ggplot(data, aes(y = reorder(property_type, -table(property_type)[property_type]))) +
  geom_bar(fill = "lightblue") +
  theme_minimal() +
  labs(title = "Distribution of Property Types",
       x = "Count",
       y = "Property Type")
```



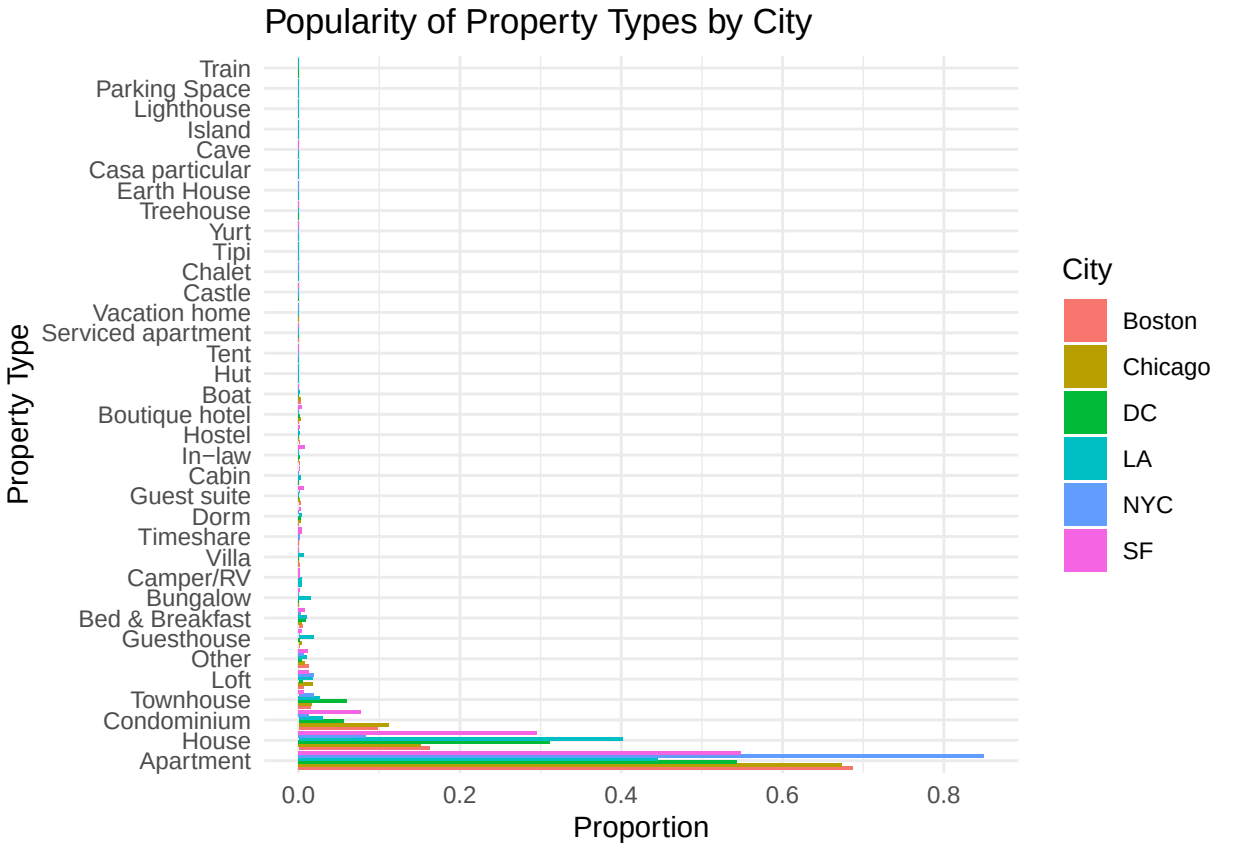
- Distribution of Room Types

```
ggplot(data, aes(x = reorder(room_type, -table(room_type)[room_type]))) +
  geom_bar(fill = "lightgreen") +
  theme_minimal() +
  labs(title = "Distribution of Room Types",
       x = "Room Type",
       y = "Count")
```



- Popularity of Property Types by City

```
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"))
```

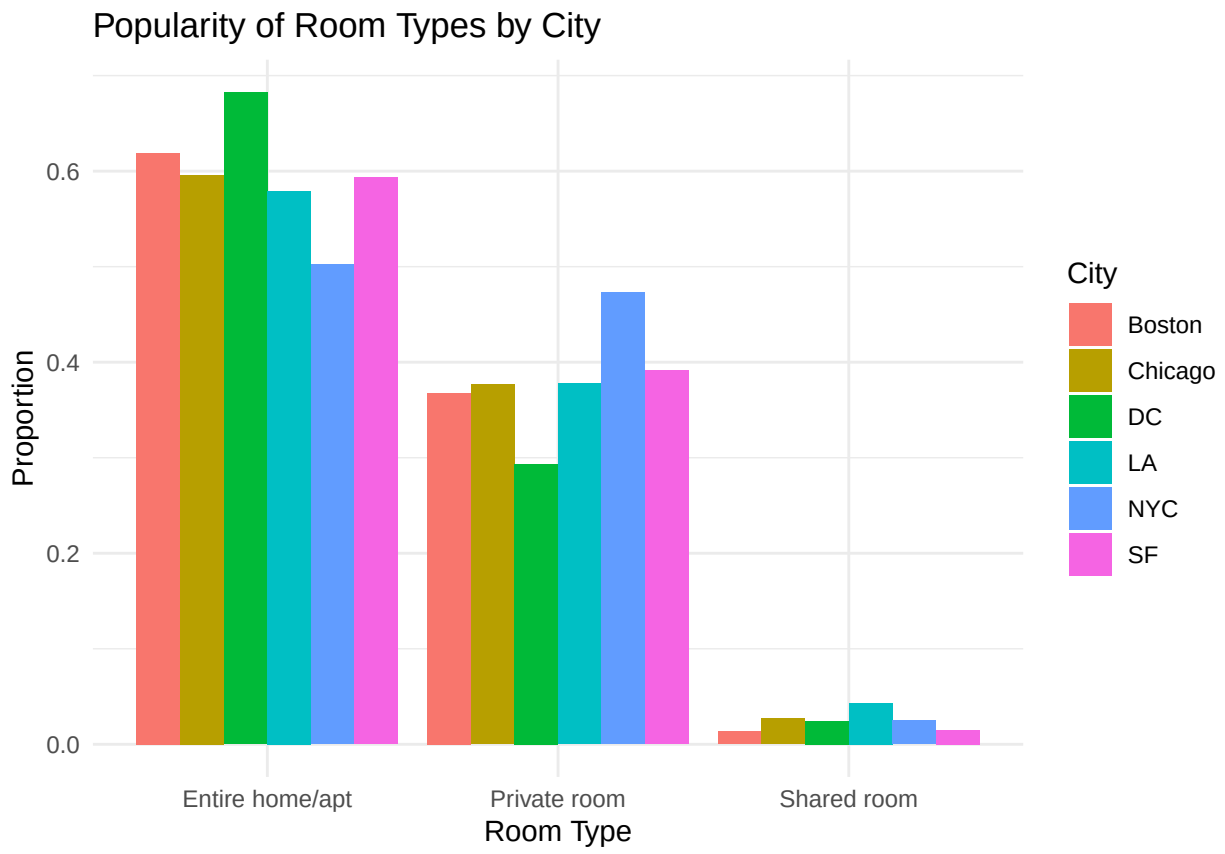


- Popularity of Room Types by City

```
# 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))
```

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

```
# Plot the proportions
ggplot(room_type_proportion, aes(x = reorder(room_type, -count), y = proportion, fill = city)) +
  geom_bar(stat = "identity", position = "dodge") +
  theme_minimal() +
  labs(title = "Popularity of Room Types by City",
       x = "Room Type",
       y = "Proportion") +
  guides(fill = guide_legend(title = "City"))
```



- Average Prices for Different Property Types

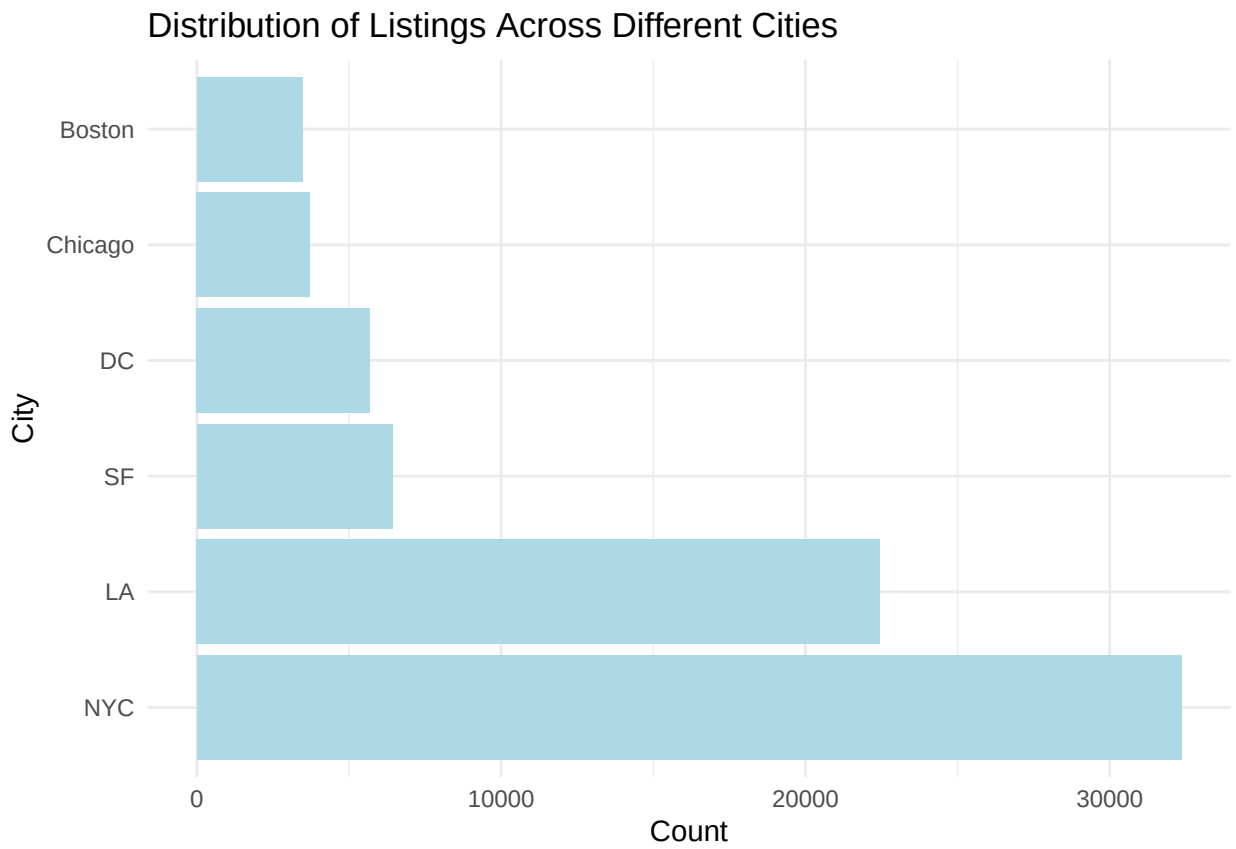
```
property_avg <- data %>%
  group_by(property_type) %>%
  summarize(mean_log_price = mean(log_price, na.rm = TRUE)) %>%
  arrange(desc(mean_log_price))

ggplot(property_avg, aes(y = reorder(property_type, mean_log_price), x = mean_log_price)) +
  geom_bar(stat = "identity", fill = "lightblue") +
  theme_minimal() +
  labs(title = "Average Log Prices for Different Property Types",
       x = "Average Log Price",
       y = "Property Type")
```

- Average Prices for Different Room Types

```
room_avg <- data %>%
  group_by(room_type) %>%
  summarize(mean_log_price = mean(log_price, na.rm = TRUE)) %>%
  arrange(desc(mean_log_price))

ggplot(room_avg, aes(x = reorder(room_type, mean_log_price), y = mean_log_price)) +
  geom_bar(stat = "identity", fill = "lightgreen") +
  theme_minimal() +
  labs(title = "Average Log Prices for Different Room Types",
       x = "Room Type",
       y = "Average Log Price")
```

- Distribution of Listings Across Different Cities

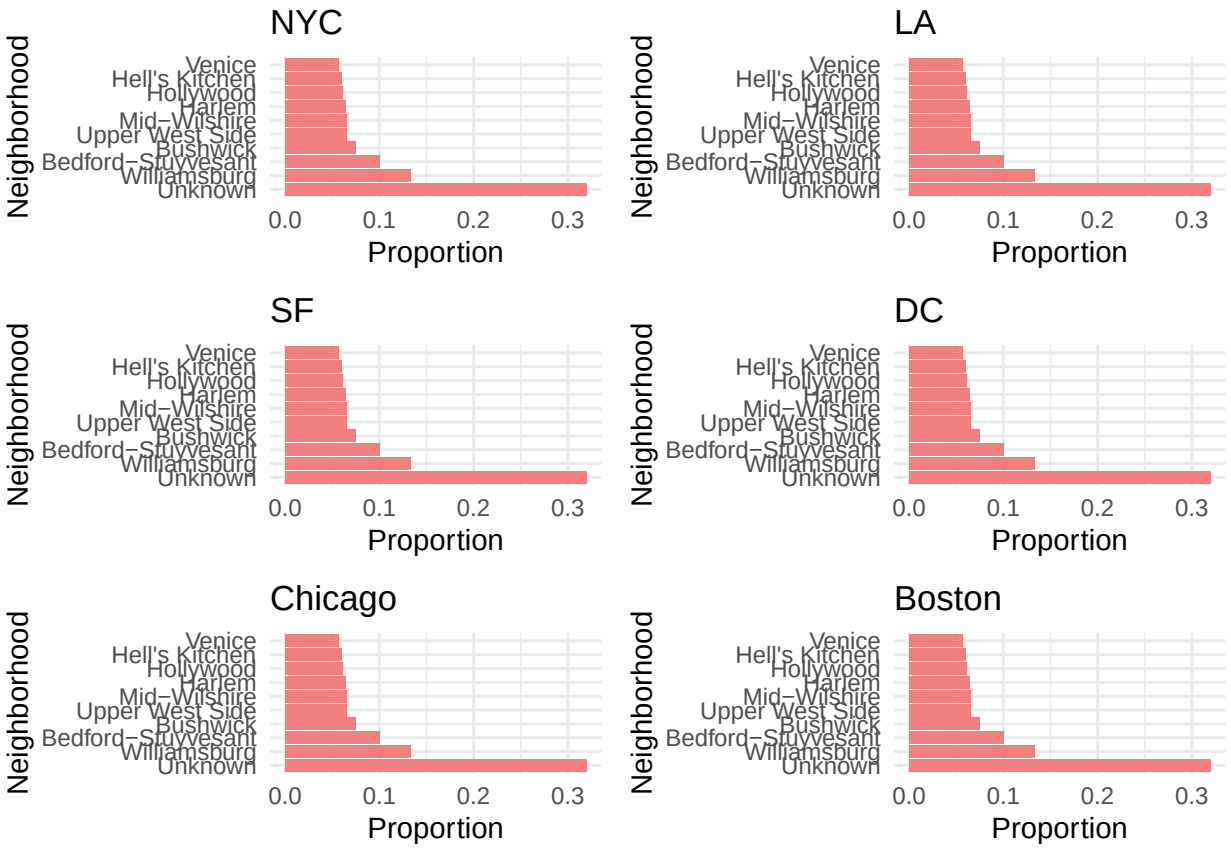
```
ggplot(data, aes(y = reorder(city, -table(city)[city]))) +
  geom_bar(fill = "lightblue") +
  theme_minimal() +
  labs(title = "Distribution of Listings Across Different Cities",
       x = "Count",
       y = "City")
```



- Distribution of Listings in Neighborhoods for the Top Cities

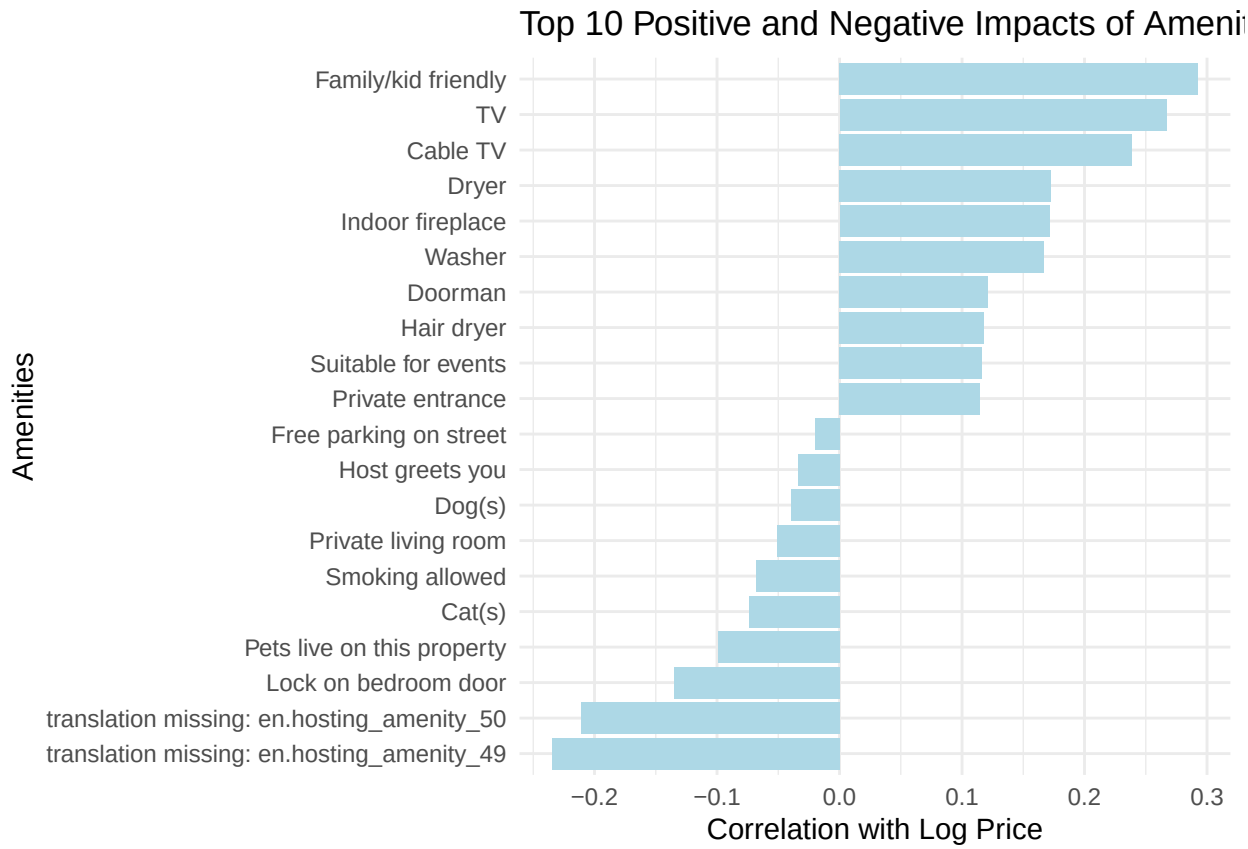
```
# Analyze the distribution of listings in neighborhoods for the top cities
top_cities <- data %>%
  count(city, sort = TRUE) %>%
  top_n(6, n) %>%  # Adjust this number to match the number of cities you want to plot
  pull(city)

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

# Loop through each city and create a plot
for (city in top_cities) {
  city_data <- data %>%
    filter(city == 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 = "lightcoral") +
    theme_minimal() +
    labs(title = city,
         x = "Proportion",
         y = "Neighborhood")

  plots[[city]] <- p
}

# Arrange the plots in a multi-column layout
do.call(grid.arrange, c(plots, ncol = 2))
```

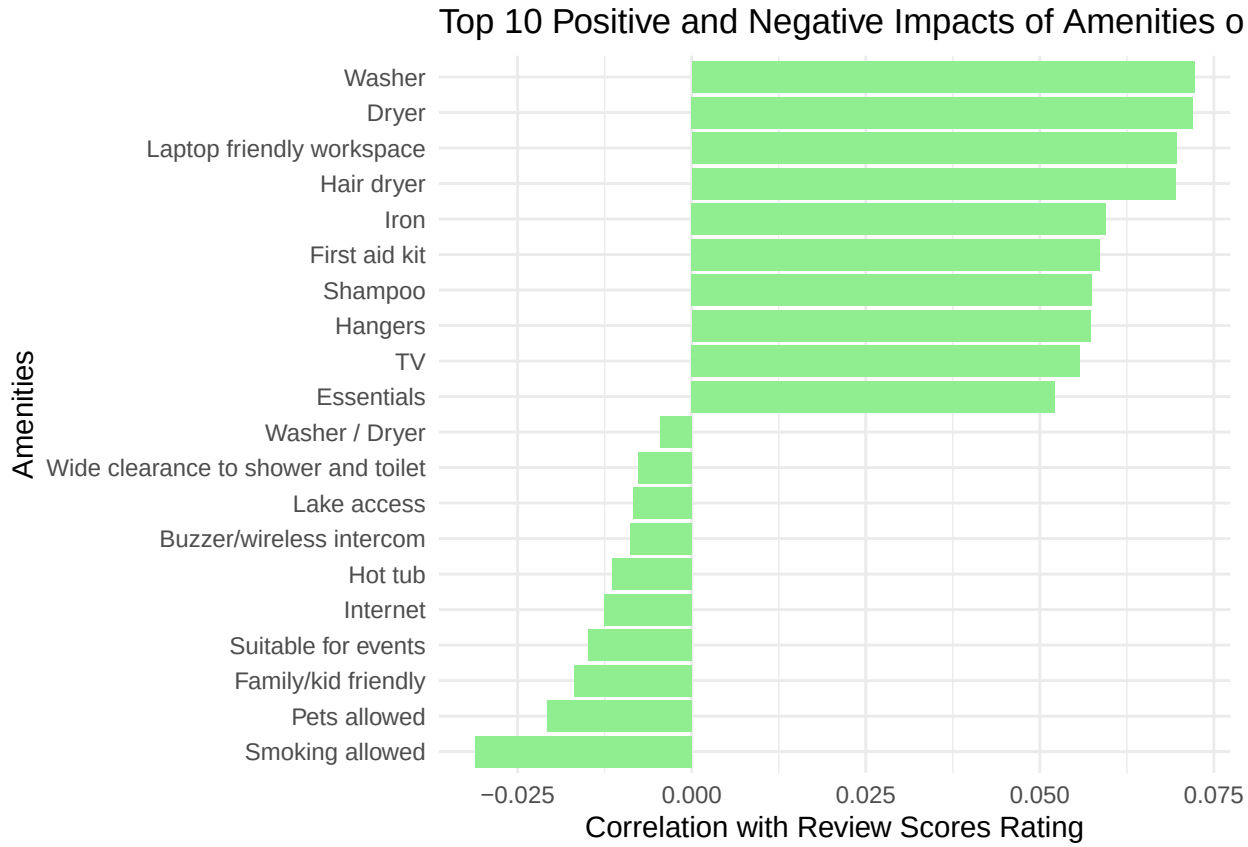



```
# Check if amenities are already processed into lists, if not, convert them
if (is.character(data$amenities[1])) {
  data <- data %>%
    mutate(amenities = str_remove_all(amenities, '[{}"]')) %>%
    mutate(amenities = str_split(amenities, ','))
}

# 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
```

```r
  filter(!is.na(amenities)) %>%  # Remove NA amenities
  distinct() %>%
  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, -latitu
  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") %>%
  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
ggplot(top_price_impact, aes(x = correlation, y = reorder(amenity, correlation))) +
  geom_bar(stat = "identity", fill = "lightblue") +
  theme_minimal() +
  labs(title = "Top 10 Positive and Negative Impacts of Amenities on Log Price",
       x = "Correlation with Log Price",
       y = "Amenities")
```




```r
# 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") %>%
  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
ggplot(top_review_impact, aes(x = correlation, y = reorder(amenity, correlation))) +
  geom_bar(stat = "identity", fill = "lightgreen") +
  theme_minimal() +
  labs(title = "Top 10 Positive and Negative Impacts of Amenities on Review Scores Rating",
       x = "Correlation with Review Scores Rating",
       y = "Amenities")
```


Top 10 Positive and Negative Impacts of Amenities o
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